**MANUAL TÉCNICO DEL SISTEMA**



**SISTEMA AUTOMÁTICO PARA DESINFECCIÓN DE MANOS Y TOMA DE TEMPERATURA PARA EMPRESAS- APLICACIÓN MÓVIL**

**AUTORES**

**Jorge Ayala Chan**

**Alfredo Valenzuela**

**Guayaquil, Ecuador**

**I PAO 2020**

Tabla de contenido

[**Resumen ejecutivo** 1](#_Toc50073944)

[**Descripción del problema** 2](#_Toc50073945)

[**Objetivos específicos** 3](#_Toc50073946)

[**¿Cómo funciona la solución?** 4](#_Toc50073947)

[**¿Qué van a construir para resolver el problema?** 5](#_Toc50073948)

[**Recursos de hardware y de software** 6](#_Toc50073949)

[**Recursos de hardware:** 6](#_Toc50073950)

[**Recursos de software:** 7](#_Toc50073951)

[**Explicación paso a paso de la implementación del proyecto** 9](#_Toc50073952)

[**Diagramas de diseño del proyecto, diagrama de circuito, diagrama del modelo entidad-relación, diagrama de casos UML, diagrama de despliegue** 10](#_Toc50073953)

[**Diagramas de diseño del proyecto:** 10](#_Toc50073954)

[**Diagrama de circuito:** 10](#_Toc50073955)

[**Diagrama de modelo entidad-relación:** 11](#_Toc50073956)

[**Diagrama UML:** 11](#_Toc50073957)

[**Diagrama de despliegue:** 12](#_Toc50073958)

[**Descripción de los campos, tipos de datos creados en la base de datos, y el código SQL** 13](#_Toc50073959)

[**Código SQL:** 13](#_Toc50073960)

[**Explicación del código fuente completo desarrollado con los comentarios correspondientes** 15](#_Toc50073961)

[**Código en Java:** 15](#_Toc50073962)

[**Main Activity:** 15](#_Toc50073963)

[**Ingreso:** 17](#_Toc50073964)

[**Mostrar:** 20](#_Toc50073965)

[**CustomListView:** 23](#_Toc50073966)

[**Código en PHP:** 26](#_Toc50073967)

[**Buscar:** 26](#_Toc50073968)

[**Insertar:** 26](#_Toc50073969)

[**Mostrar:** 27](#_Toc50073970)

[**Conexión:** 27](#_Toc50073971)

[**Análisis de presupuesto** 29](#_Toc50073972)

[**Conclusiones** 30](#_Toc50073973)

[**Referencias bibliográficas** 31](#_Toc50073974)

# Resumen ejecutivo

El siguiente proyecto propone una posible solución para poder realizar un control y una prevención de los síntomas del nuevo virus COVID-19 de una manera más automatizada. Para este proyecto se ha propuesto la construcción de una máquina que dispense alcohol en gel para que el usuario tenga una correcta desinfección de sus manos, pero también se ha propuesto que la misma máquina tenga un sensor de temperatura infrarrojo para que así se pueda conocer la temperatura corporal del usuario. Dicha temperatura será enviada a través de una red wifi a una aplicación móvil, la cual a su vez la enviará a una base de datos para mantener un registro controlado de las personas (usuarios), ya sean trabajadores o clientes, que ingresen a un establecimiento. El fin de la propuesta y el prototipo a diseñar tiene como su principal objetivo contribuir a la desaceleración de la curva de contagios de la enfermedad COVID-19.

# Descripción del problema

Debido a la crisis sanitaria mundial que se está atravesando por efectos de la pandemia causada por el nuevo virus COVID-19, se está procurando mantener un control sanitario verificando los síntomas de todas las personas al momento de entrar a un restaurante, local comercial, edificio, etc… El problema es que muchas veces la temperatura tomada puede quedar olvidada y no se lleva ningún registro de la persona a la cual fue tomada. Esto podría contribuir en una desorganización y descontrol al momento de poder saber la cantidad de personas que ingresan y salen. También se tiene el ajetreo que significa mantener a una persona que constantemente tome la temperatura y desinfecte a todo aquel que ingresa por la puerta de un lugar. Esto demanda tiempo y recursos para el lugar que desea implementar esta seguridad y de vez en cuando dicha persona encargada de tomar la temperatura puede fallar, dejando al aire libre un pequeño porcentaje de error que podría ser crítico en algunas ocasiones.

# Objetivos específicos

* Desarrollar un mecanismo que dispense alcohol en gel y tome la temperatura corporal de los usuarios sin necesidad de tener contacto con su piel.
* Elaborar una aplicación móvil que muestre información detallada acerca de los registros realizados con la máquina.
* Implementar los servicios web en una aplicación móvil desarrollada en Java usando Android Studio.
* Emplear una forma de reconocer a los usuarios que presentan una temperatura corporal más elevada de lo normal (37°C) para que se pueda tener una mejor prevención sanitaria.

# ¿Cómo funciona la solución?

La solución a la problemática expuesta anteriormente consiste en el uso de las técnicas de programación de hardware con las de programación de software. Para esto se tendrá una máquina que se encarará de tomar los datos esenciales como la temperatura y mantener un control del volumen de alcohol en gel dispensado hacia los usuarios. Esta información recopilada será enviada usando los protocolos de comunicación por wifi hacia una base de datos, la cual se encargará de recopilar todos los registros de gente que ha entrado al establecimiento durante todo el día. Para poder crear un nuevo registro es necesario que el usuario ingrese su número de cédula junto con su nombre y apellido en la aplicación para que de esta forma el sistema sepa cuál es la persona a la que se le va a registrar la temperatura corporal. La aplicación móvil desarrollada para este fin posee una ventana donde muestra los datos almacenados dentro de la base de datos y también permite buscar a alguna persona específicamente según su número de cédula. La búsqueda que se realizará será del registro más reciente.

# ¿Qué van a construir para resolver el problema?

Para resolver el problema se construirán dos partes: el hardware y el software. Para el hardware se tendrá una máquina automatizada controlada por un Arduino UNO, la cual dispensará alcohol en gel usando un motor DC y tomará la temperatura mediante un sensor de temperatura infrarrojo. El software será diseñado utilizando el IDE de Android Studio y el lenguaje Java para lograr obtener un sistema automatizado que tome la temperatura y al mismo tiempo la registre dentro de la base de datos para que sea posible visualizarla en la aplicación de la forma que ya se mencionó anteriormente. La base de datos para la solución será realizada en SQL y administrada utilizando la interfaz de phpMyAdmin. El servidor para la aplicación será gestionado a través de Apache y para lograr las distintas actividades que debe realizar la aplicación (conexión, insertar datos, mostrar datos y buscar datos) se usarán scripts en lenguaje PHP.

# Recursos de hardware y de software

## **Recursos de hardware:**

* **Arduino UNO:** Microcontrolador encargado de los accionamientos y sensores del sistema.

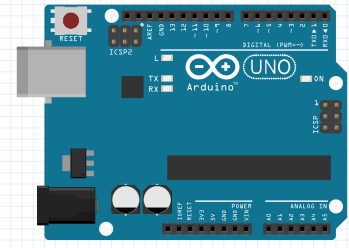
****

Ilustración 1: Arduino UNO

* **Arduino Motor Shield:** Driver que permite la comunicación entre el motor DC y el controlador.

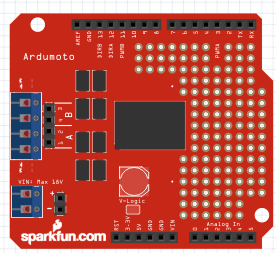
****

Ilustración 2: Arduino Motor Shield

* **Motor DC:** Actuador utilizado para dispensar el alcohol en gel.

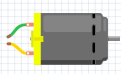
****

Ilustración 3: Motor DC

* **Sensor PIR:** Sensor utilizado para detectar cuando un usuario va a utilizar la máquina.

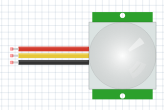
****

Ilustración 4: Sensor PIR

* **Sensor de temperatura MLX90614:** Sensor infrarrojo para detectar la temperatura corporal sin necesidad de tener contacto con la piel.

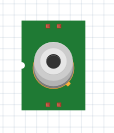
****

Ilustración 5: Sensor MLX90614

* **Módulo ESP8266:** Módulo para permitir la comunicación y envío de datos vía wifi.

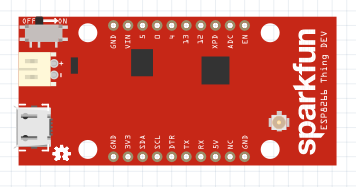
****

Ilustración 6: ESP8266

## **Recursos de software:**

* **Android Studio IDE (Java):** IDE para desarrollo de aplicaciones Android basadas en el lenguaje Java.



Ilustración 7: Android Studio y Java

* **XAMPP:** Servidor local basado en Apache para poder crear la comunicación de todo el sistema y guardar la información entre el hardware y el software.



Ilustración 8: XAMPP

* **PHP:** Lenguaje de programación utilizado para manejar los servicios web entre la aplicación móvil y la base de datos.



Ilustración 9: PHP

* **phpMyAdmin:** Administrador con interfaz gráfica de la base de datos.



Ilustración 10: phpMyAdmin

# Explicación paso a paso de la implementación del proyecto

Para la implementación del proyecto se realizó un modelo esquemático del circuito electromecánico debido a que parte del proyecto es simulado. Por tal razón, se procederá a explicar el funcionamiento del sistema de hardware.

En primer lugar, se tiene un microcontrolador (Arduino UNO), el cual fue escogido debido a su capacidad de prototipado y control por envío y recepción de señales analógicas y digitales. Junto con el Arduino se montó un motor shield (driver) debido a que el motor dc (que funcionará para activar el mecanismo dispensador de alcohol en gel) requiere de una señal analógica para poder funcionar. Así el Arduino envía sus señales de control a través del driver, el cual envía las señales analógicas necesarias para que el motor pueda funcionar correctamente.

Conectado a las salidas digitales del Arduino se va a tener al sensor PIR, el cual será fundamental para el funcionamiento ya que al detectar un movimiento activará la máquina para que esta tome la temperatura corporal del usuario y le dispense gel. También se encuentra el ESP8266 conectado a las salidas digitales del Arduino para que le lleguen los datos que necesita transmitir vía wifi a este módulo. En las entradas analógicas se encuentra conectado el sensor de temperatura infrarrojo.

Una vez se tengan los datos de la temperatura y el volumen de gel dispensado, estos se enviarán hacia la base de datos para que se agreguen a los registros. Una vez agregados los registros, estos automáticamente ya estarán disponibles dentro de la aplicación móvil para así poder observar a todas las personas que se hayan tomado la temperatura dentro del establecimiento. Los nombres cuya temperatura sobrepase los 37 °C (fiebre) se presentarán de color rojo para que se tenga una especial precaución y se actúe según las reglas de control sanitario del establecimiento. También será posible buscar los datos de algún usuario introduciendo su número de cédula en la ventana de buscar.

# Diagramas de diseño del proyecto, diagrama de circuito, diagrama del modelo entidad-relación, diagrama de casos UML, diagrama de despliegue

## **Diagramas de diseño del proyecto:**

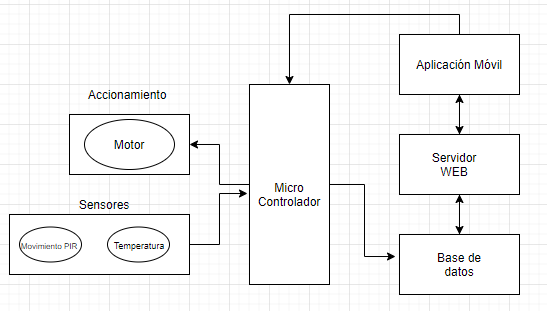


Ilustración 11: Diagrama de diseño

**Diagrama de circuito:**

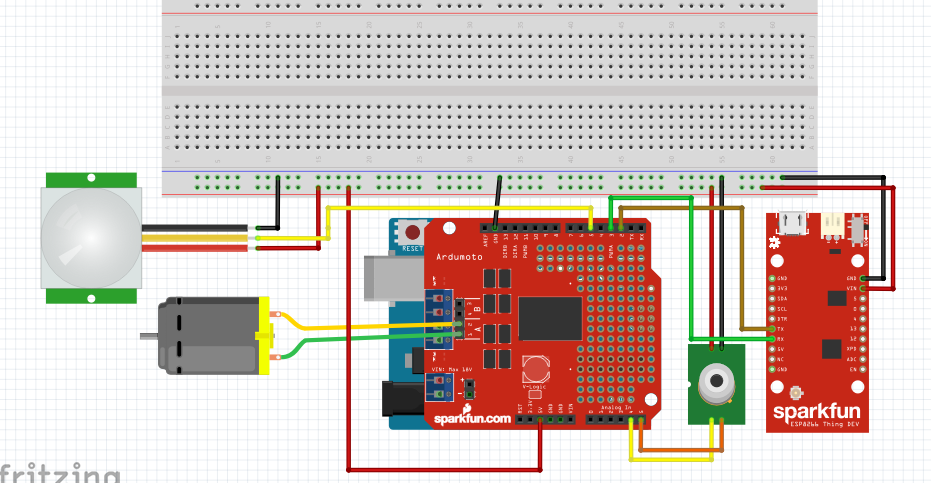


Ilustración 12: Diagrama del circuito

## **Diagrama de modelo entidad-relación:**

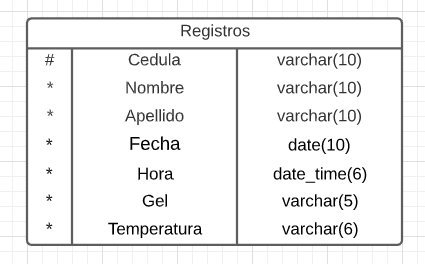


Ilustración 13: Diagrama entidad-relación

## **Diagrama UML:**

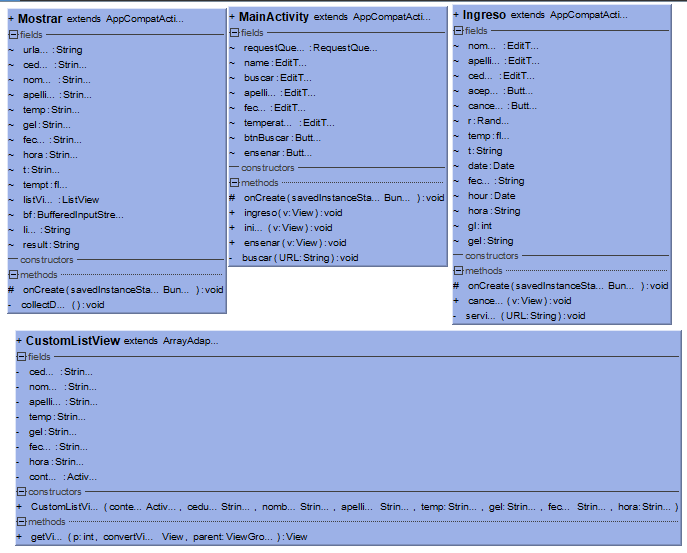


Ilustración 14: Diagrama UML

## **Diagrama de despliegue:**

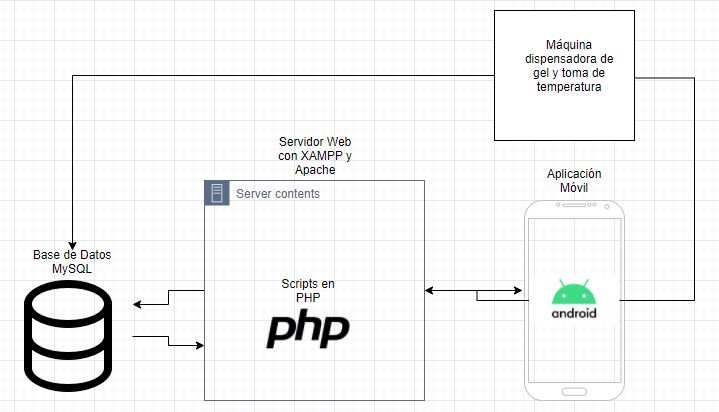
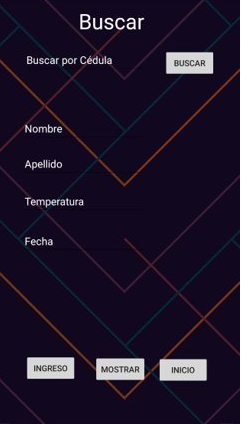


Ilustración 15: Diagrama de despliegue

**Manual de usuario**



Aplicacion movil – Buscar

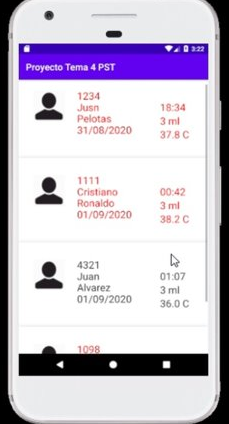


* Opción de búsqueda por número de cedula (identificador único)
* Se muestran los datos de nombre, apellido, temperatura, fecha de ingreso de la persona especifica
* Hay opción de ir al activity 2 y 3

Aplicación movil – Ingreso

* En este activity se introduce la información a la base de datos del sistema
* Los datos para meter son la cedula que es el identificador único de la persona, su nombre y su apellido
* No se podrá ingresar un numero de cedula ya existente

Aplicación movil – Mostrar datos

* Este activity muestra todos los datos guardados de los diferentes usuarios
* Los datos para mostrar por numero de cedula son: nombre, apellido, fecha de registro, hora del registro, cantidad de desinfectante en ml, temperatura registrada de la persona
* En caso de tener una temperatura superior a 37 grados centígrados, la información se mostrará en rojo para la identificación oportuna de las personas que posiblemente estén contagiadas.

# Enlace a la sección de ASANA

<https://app.asana.com/0/1192039216671572/list>

# Descripción de los campos, tipos de datos creados en la base de datos, y el código SQL

Según la ilustración 13 se puede apreciar que para la base de datos se tiene una sola tabla llamada registros que es utilizada para guardar la información de los datos recopilados en tiempo real. Los campos de dicha tabla son Cedula, como clave primaria utilizada para registrar la cédula de todos los usuarios y así poder tener un rasgo único que los identifique a todos, Nombre y Apellido, son dos campos usados para tener el registro de los nombres y el apellido de dicha persona, Fecha, es el campo donde se guarda la fecha del momento en que se realizó el registro en el sistema y es de tipo date para mantener el formato deseado de fecha, Hora es parecido al campo Fecha, solo que en este caso es de tipo date\_time para poder almacenar el valor de la hora de ingreso al sistema con el formato deseado, Temperatura es el campo donde se almacena la temperatura corporal del usuario en grados Celsius y Gel es el campo donde se almacena el volumen dispensado de gel de la máquina hacia el usuario.

## **Código SQL:**

-- phpMyAdmin SQL Dump

-- version 4.9.0.1

-- https://www.phpmyadmin.net/

--

-- Servidor: 127.0.0.1

-- Tiempo de generación: 01-09-2020 a las 05:32:28

-- Versión del servidor: 10.3.16-MariaDB

-- Versión de PHP: 7.3.7

**SET** SQL\_MODE **=** "NO\_AUTO\_VALUE\_ON\_ZERO"**;**

**SET** AUTOCOMMIT **=** 0**;**

**START** **TRANSACTION;**

**SET** time\_zone **=** "+00:00"**;**

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/**;**

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/**;**

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/**;**

/\*!40101 SET NAMES utf8mb4 \*/**;**

--

-- Base de datos: `proyecto`

--

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `registros`

--

**CREATE** **TABLE** `registros` **(**

`Cedula` varchar**(**20**)** **NOT** **NULL,**

`Nombre` varchar**(**20**)** **NOT** **NULL,**

`Apellido` varchar**(**20**)** **NOT** **NULL,**

`Fecha` varchar**(**10**)** **NOT** **NULL,**

`Hora` varchar**(**10**)** **NOT** **NULL,**

`Gel` varchar**(**5**)** **NOT** **NULL,**

`Temperatura` varchar**(**10**)** **NOT** **NULL**

**)** ENGINE**=**InnoDB **DEFAULT** CHARSET**=**latin1**;**

--

-- Volcado de datos para la tabla `registros`

--

**INSERT** **INTO** `registros` **(**`Cedula`**,** `Nombre`**,** `Apellido`**,** `Fecha`**,** `Hora`**,** `Gel`**,** `Temperatura`**)** **VALUES**

**(**'1234'**,** 'Jusn'**,** 'Pelotas'**,** '31/08/2020'**,** '18:34'**,** '3 ml'**,** '37.8 C'**),**

**(**'1111'**,** 'Cristiano'**,** 'Ronaldo'**,** '01/09/2020'**,** '00:42'**,** '3 ml'**,** '38.2 C'**),**

**(**'4321'**,** 'Juan'**,** 'Alvarez'**,** '01/09/2020'**,** '01:07'**,** '3 ml'**,** '36.0 C'**),**

**(**'1098'**,** 'Jorge'**,** 'Ayala'**,** '01/09/2020'**,** '01:08'**,** '3 ml'**,** '37.8 C'**),**

**(**'5421'**,** 'Alfredo'**,** 'Valenzuela'**,** '01/09/2020'**,** '01:08'**,** '1 ml'**,** '37.7 C'**),**

**(**'5675'**,** 'Roberto'**,** 'Gomez'**,** '01/09/2020'**,** '03:22'**,** '4 ml'**,** '35.1 C'**);**

**COMMIT;**

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/**;**

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/**;**

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/**;**

# Explicación del código fuente completo desarrollado con los comentarios correspondientes

## **Código en Java:**

### **Main Activity:**

Dentro de este activity se realizó la búsqueda de los datos más recientes de una persona según su número de cédula.

package com**.**example**.**proyectotema4pst**;**

**import** androidx**.**appcompat**.**app**.**AppCompatActivity**;**

**import** android**.**content**.**Intent**;**

**import** android**.**os**.**Bundle**;**

**import** android**.**view**.**View**;**

**import** android**.**widget**.**Button**;**

**import** android**.**widget**.**EditText**;**

**import** android**.**widget**.**ListView**;**

**import** android**.**widget**.**TextView**;**

**import** android**.**widget**.**Toast**;**

**import** com**.**android**.**volley**.**Request**;**

**import** com**.**android**.**volley**.**RequestQueue**;**

**import** com**.**android**.**volley**.**Response**;**

**import** com**.**android**.**volley**.**VolleyError**;**

**import** com**.**android**.**volley**.**toolbox**.**JsonArrayRequest**;**

**import** com**.**android**.**volley**.**toolbox**.**StringRequest**;**

**import** com**.**android**.**volley**.**toolbox**.**Volley**;**

**import** org**.**json**.**JSONArray**;**

**import** org**.**json**.**JSONException**;**

**import** org**.**json**.**JSONObject**;**

**import** org**.**w3c**.**dom**.**Text**;**

**import** java**.**util**.**ArrayList**;**

public class MainActivity **extends** AppCompatActivity **{**

RequestQueue requestQueue**;**

EditText name**,** buscar**,** apellido**,** fecha**,** temperatura**;**

Button btnBuscar**,** ensenar**;**

@Override

protected void onCreate**(**Bundle savedInstanceState**)** **{**

**super.**onCreate**(**savedInstanceState**);**

setContentView**(**R**.**layout**.**activity\_main**);**

name **=** **(**EditText**)** findViewById**(**R**.**id**.**name**);**

buscar **=** **(**EditText**)**findViewById**(**R**.**id**.**buscar**);**

apellido **=** **(**EditText**)** findViewById**(**R**.**id**.**apellido**);**

fecha **=** **(**EditText**)** findViewById**(**R**.**id**.**fecha**);**

temperatura **=** **(**EditText**)** findViewById**(**R**.**id**.**tempt**);**

btnBuscar**=(**Button**)**findViewById**(**R**.**id**.**btnBuscar**);**

btnBuscar**.**setOnClickListener**(new** View**.**OnClickListener**()** **{**

@Override

public void onClick**(**View view**)** **{**

buscar**(**"http://192.168.100.7:8008/proyecto/buscar.php?Cedula="**+**buscar**.**getText**()+**""**);**

**}**

**});**

**}**

public void ingreso**(**View v**){**

Intent intent **=** **new** Intent**(this,** Ingreso**.**class**);**

startActivity**(**intent**);**

**}**

public void inicio**(**View v**){**

Intent intent **=** **new** Intent**(this,** MainActivity**.**class**);**

startActivity**(**intent**);**

**}**

public void ensenar**(**View v**){**

Intent intent **=** **new** Intent**(this,** Mostrar**.**class**);**

startActivity**(**intent**);**

**}**

private void buscar**(**String URL**){**

JsonArrayRequest jsonArrayRequest **=** **new** JsonArrayRequest**(**URL**,** **new** Response**.**Listener**<**JSONArray**>()** **{**

@Override

public void onResponse**(**JSONArray response**)** **{**

JSONObject jsonObject **=** **null;**

**for** **(**int i **=** 0**;** i **<** response**.**length**();** i**++)** **{**

**try** **{**

jsonObject **=** response**.**getJSONObject**(**i**);**

name**.**setText**(**jsonObject**.**getString**(**"Nombre"**));**

apellido**.**setText**(**jsonObject**.**getString**(**"Apellido"**));**

fecha**.**setText**(**jsonObject**.**getString**(**"Fecha"**));**

temperatura**.**setText**(**jsonObject**.**getString**(**"Temperatura"**));**

**}** **catch** **(**JSONException e**)** **{**

Toast**.**makeText**(**getApplicationContext**(),** e**.**getMessage**(),** Toast**.**LENGTH\_SHORT**).**show**();**

**}**

**}**

**}**

**},** **new** Response**.**ErrorListener**()** **{**

@Override

public void onErrorResponse**(**VolleyError error**)** **{**

Toast**.**makeText**(**getApplicationContext**(),**"ERROR DE CONEXION"**,**Toast**.**LENGTH\_SHORT**).**show**();**

**}**

**});**

requestQueue**=**Volley**.**newRequestQueue**(this);**

requestQueue**.**add**(**jsonArrayRequest**);**

**}**

**}**

### **Ingreso:**

Dentro de este activity se tienen los campos EditText para ingresar un nuevo usuario. Los campos que se deben llenar son la cédula, el nombre y el apellido.

package com**.**example**.**proyectotema4pst**;**

**import** androidx**.**appcompat**.**app**.**AppCompatActivity**;**

**import** android**.**content**.**Intent**;**

**import** android**.**os**.**Bundle**;**

**import** android**.**text**.**format**.**DateFormat**;**

**import** android**.**text**.**format**.**Time**;**

**import** android**.**view**.**View**;**

**import** android**.**widget**.**Button**;**

**import** android**.**widget**.**EditText**;**

**import** android**.**widget**.**Toast**;**

**import** com**.**android**.**volley**.**AuthFailureError**;**

**import** com**.**android**.**volley**.**Request**;**

**import** com**.**android**.**volley**.**RequestQueue**;**

**import** com**.**android**.**volley**.**Response**;**

**import** com**.**android**.**volley**.**VolleyError**;**

**import** com**.**android**.**volley**.**toolbox**.**StringRequest**;**

**import** com**.**android**.**volley**.**toolbox**.**Volley**;**

**import** java**.**text**.**SimpleDateFormat**;**

**import** java**.**util**.**Calendar**;**

**import** java**.**util**.**Date**;**

**import** java**.**util**.**HashMap**;**

**import** java**.**util**.**Locale**;**

**import** java**.**util**.**Map**;**

**import** java**.**util**.**Random**;**

public class Ingreso **extends** AppCompatActivity **{**

EditText nombre**,** apellido**,** cedula**;**

Button aceptar**,** cancelar**;**

Random r **=** **new** Random**();**

float temp**;**

String t**;**

Date date**;**

String fecha**;**

Date hour**;**

String hora**;**

int gl**;**

String gel**;**

@Override

protected void onCreate**(**Bundle savedInstanceState**)** **{**

**super.**onCreate**(**savedInstanceState**);**

setContentView**(**R**.**layout**.**activity\_ingreso**);**

nombre **=** **(**EditText**)**findViewById**(**R**.**id**.**nombre**);**

apellido **=** **(**EditText**)**findViewById**(**R**.**id**.**apellido**);**

cedula**=(**EditText**)**findViewById**(**R**.**id**.**cdl**);**

aceptar **=** **(**Button**)**findViewById**(**R**.**id**.**aceptar**);**

cancelar **=** **(**Button**)**findViewById**(**R**.**id**.**cancelar**);**

//int i = r.nextInt(39 - 33)+33;

gl **=** r**.**nextInt**(**5**-**1**)+**1**;**

gel**=**Integer**.**toString**(**gl**);**

temp **=** r**.**nextFloat**()\*(**40f**-**35f**)+**35f**;**

t **=** String**.**format**(**Locale**.**US**,**"%.1f"**,**temp**);**

//t = Float.toString(temp);

date **=** Calendar**.**getInstance**().**getTime**();**

SimpleDateFormat dateFormat **=** **new** SimpleDateFormat**(**"dd/MM/yyyy"**);**

fecha **=** dateFormat**.**format**(**date**);**

hour **=** Calendar**.**getInstance**().**getTime**();**

SimpleDateFormat hourFormat **=** **new** SimpleDateFormat**(**"HH:mm"**);**

hora **=** hourFormat**.**format**(**hour**);**

aceptar**.**setOnClickListener**(new** View**.**OnClickListener**()** **{**

@Override

public void onClick**(**View v**)** **{**

servicio**(**"http://192.168.100.7:8008/proyecto/insertar.php"**);**

**}**

**});**

**}**

public void cancelar**(**View v**){**

Intent intent **=** **new** Intent**(this,** MainActivity**.**class**);**

startActivity**(**intent**);**

**}**

private void servicio**(**String URL**){**

StringRequest stringRequest **=** **new** StringRequest**(**Request**.**Method**.**POST**,** URL**,** **new** Response**.**Listener**<**String**>()** **{**

@Override

public void onResponse**(**String response**)** **{**

Toast**.**makeText**(**getApplicationContext**(),** "OPERACION EXITOSA"**,**Toast**.**LENGTH\_SHORT**).**show**();**

**}**

**},** **new** Response**.**ErrorListener**()** **{**

@Override

public void onErrorResponse**(**VolleyError error**)** **{**

Toast**.**makeText**(**getApplicationContext**(),**error**.**toString**(),**Toast**.**LENGTH\_SHORT**).**show**();**

**}**

**}){**

@Override

protected Map**<**String**,** String**>** getParams**()** **throws** AuthFailureError **{**

Map**<**String**,**String**>** parametros**=new** HashMap**<**String**,** String**>();**

parametros**.**put**(**"Cedula"**,**cedula**.**getText**().**toString**());**

parametros**.**put**(**"Nombre"**,**nombre**.**getText**().**toString**());**

parametros**.**put**(**"Apellido"**,**apellido**.**getText**().**toString**());**

parametros**.**put**(**"Fecha"**,**fecha**);**

parametros**.**put**(**"Hora"**,**hora**);**

parametros**.**put**(**"Gel"**,**gel**+**" ml"**);**

parametros**.**put**(**"Temperatura"**,**t**+**" C"**);**

**return** parametros**;**

**}**

**};**

RequestQueue requestQueue **=** Volley**.**newRequestQueue**(this);**

requestQueue**.**add**(**stringRequest**);**

**}**

**}**

### **Mostrar:**

En este activity se muestran todos los datos almacenados en la base utilizando un ListView.

package com**.**example**.**proyectotema4pst**;**

**import** androidx**.**appcompat**.**app**.**AppCompatActivity**;**

**import** android**.**os**.**Bundle**;**

**import** android**.**os**.**StrictMode**;**

**import** android**.**widget**.**ListView**;**

**import** org**.**json**.**JSONArray**;**

**import** org**.**json**.**JSONObject**;**

**import** java**.**io**.**BufferedInputStream**;**

**import** java**.**io**.**BufferedReader**;**

**import** java**.**io**.**InputStreamReader**;**

**import** java**.**net**.**HttpURLConnection**;**

**import** java**.**net**.**URL**;**

**import** java**.**nio**.**Buffer**;**

public class Mostrar **extends** AppCompatActivity **{**

String urladd**=**"http://192.168.100.7:8008/proyecto/mostrar.php"**;**

String **[]** cedula**;**

String **[]** nombre**;**

String **[]** apellido**;**

String **[]** temp**;**

String **[]** gel**;**

String **[]** fecha**;**

String **[]** hora**;**

String **[]** t**;**

float tempt**;**

ListView listView**;**

BufferedInputStream bf**;**

String line**=null;**

String result**=null;**

@Override

protected void onCreate**(**Bundle savedInstanceState**)** **{**

**super.**onCreate**(**savedInstanceState**);**

setContentView**(**R**.**layout**.**activity\_mostrar**);**

listView**=(**ListView**)**findViewById**(**R**.**id**.**ListView**);**

StrictMode**.**setThreadPolicy**((new** StrictMode**.**ThreadPolicy**.**Builder**().**permitNetwork**().**build**()));**

collectData**();**

CustomListView customListView**=** **new** CustomListView**(this,**cedula**,**nombre**,**apellido**,**temp**,**gel**,**fecha**,**hora**);**

listView**.**setAdapter**(**customListView**);**

**}**

private void collectData**(){**

**try{**

URL url**=new** URL**(**urladd**);**

HttpURLConnection con**=(**HttpURLConnection**)**url**.**openConnection**();**

con**.**setRequestMethod**(**"GET"**);**

bf**=new** BufferedInputStream**(**con**.**getInputStream**());**

**}catch(**Exception ex**){**

ex**.**printStackTrace**();**

**}**

**try{**

BufferedReader br **=** **new** BufferedReader**(new** InputStreamReader**(**bf**));**

StringBuilder sb **=** **new** StringBuilder**();**

**while((**line**=**br**.**readLine**())!=null){**

sb**.**append**(**line**+**"\n"**);**

**}**

bf**.**close**();**

result**=**sb**.**toString**();**

**}catch(**Exception ex**){**

ex**.**printStackTrace**();**

**}**

**try{**

JSONArray ja**=new** JSONArray**(**result**);**

JSONObject jo **=** **null;**

cedula **=** **new** String**[**ja**.**length**()];**

nombre **=** **new** String**[**ja**.**length**()];**

apellido **=** **new** String**[**ja**.**length**()];**

temp **=** **new** String**[**ja**.**length**()];**

gel **=** **new** String**[**ja**.**length**()];**

fecha **=** **new** String**[**ja**.**length**()];**

hora **=** **new** String**[**ja**.**length**()];**

**for(**int i **=**0**;**i**<=**ja**.**length**();**i**++){**

jo**=**ja**.**getJSONObject**(**i**);**

//tempt=Float.parseFloat(t[i]=jo.getString("Temperatura"));

//if(tempt>38f) {

cedula**[**i**]** **=** jo**.**getString**(**"Cedula"**);**

nombre**[**i**]** **=** jo**.**getString**(**"Nombre"**);**

apellido**[**i**]** **=** jo**.**getString**(**"Apellido"**);**

temp**[**i**]** **=** jo**.**getString**(**"Temperatura"**);**

gel**[**i**]** **=** jo**.**getString**(**"Gel"**);**

fecha**[**i**]** **=** jo**.**getString**(**"Fecha"**);**

hora**[**i**]** **=** jo**.**getString**(**"Hora"**);**

//}

**}**

**}catch** **(**Exception ex**){**

ex**.**printStackTrace**();**

**}**

**}**

**}**

### **CustomListView:**

Es una clase de Java que extiende de un ArrayAdapter para crear una colección de los datos que se deben ingresar en los TextView del activity Mostrar.

package com**.**example**.**proyectotema4pst**;**

**import** android**.**app**.**Activity**;**

**import** android**.**content**.**Context**;**

**import** android**.**view**.**LayoutInflater**;**

**import** android**.**view**.**View**;**

**import** android**.**view**.**ViewGroup**;**

**import** android**.**widget**.**ArrayAdapter**;**

**import** android**.**widget**.**ImageView**;**

**import** android**.**widget**.**TextView**;**

**import** org**.**w3c**.**dom**.**Text**;**

**import** androidx**.**annotation**.**NonNull**;**

**import** androidx**.**annotation**.**Nullable**;**

public class CustomListView **extends** ArrayAdapter**<**String**>** **{**

private String **[]** cedula**;**

private String **[]** nombre**;**

private String **[]** apellido**;**

private String **[]** temp**;**

private String **[]** gel**;**

private String **[]** fecha**;**

private String **[]** hora**;**

private Activity context**;**

public CustomListView**(**Activity context**,** String**[]** cedula**,**String**[]** nombre**,**String**[]** apellido**,**String**[]** temp**,** String**[]** gel**,**String**[]** fecha**,**String**[]** hora**)** **{**

**super(**context**,** R**.**layout**.**diseno**,**nombre**);**

**this.**context**=**context**;**

**this.**cedula**=**cedula**;**

**this.**nombre**=**nombre**;**

**this.**apellido**=**apellido**;**

**this.**temp**=**temp**;**

**this.**gel**=**gel**;**

**this.**fecha**=**fecha**;**

**this.**hora**=**hora**;**

**}**

@NonNull

@Override

public View getView**(**int p**,** @Nullable View convertView**,** @NonNull ViewGroup parent**){**

View r **=** convertView**;**

ViewHolder viewHolder**=null;**

**if(**r**==null){**

LayoutInflater layoutInflater**=**context**.**getLayoutInflater**();**

r**=**layoutInflater**.**inflate**(**R**.**layout**.**diseno**,null,true);**

viewHolder**=new** ViewHolder**(**r**);**

r**.**setTag**(**viewHolder**);**

**}**

**else** **{**

viewHolder**=(**ViewHolder**)**r**.**getTag**();**

**}**

String ing **=** **(**temp**[**p**]).**replace**(**"C"**,**""**);**

float compare **=** Float**.**parseFloat**(**ing**);**

**if(**compare**>=**37f**){**

viewHolder**.**tvw1**.**setText**(**nombre**[**p**]);**

viewHolder**.**tvw1**.**setTextColor**(**android**.**graphics**.**Color**.**RED**);**

viewHolder**.**tvw2**.**setText**(**apellido**[**p**]);**

viewHolder**.**tvw2**.**setTextColor**(**android**.**graphics**.**Color**.**RED**);**

viewHolder**.**tvw**.**setText**(**cedula**[**p**]);**

viewHolder**.**tvw**.**setTextColor**(**android**.**graphics**.**Color**.**RED**);**

viewHolder**.**tvw3**.**setText**(**fecha**[**p**]);**

viewHolder**.**tvw3**.**setTextColor**(**android**.**graphics**.**Color**.**RED**);**

viewHolder**.**tvw4**.**setText**(**hora**[**p**]);**

viewHolder**.**tvw4**.**setTextColor**(**android**.**graphics**.**Color**.**RED**);**

viewHolder**.**tvw5**.**setText**(**gel**[**p**]);**

viewHolder**.**tvw5**.**setTextColor**(**android**.**graphics**.**Color**.**RED**);**

viewHolder**.**tvw6**.**setText**(**temp**[**p**]);**

viewHolder**.**tvw6**.**setTextColor**(**android**.**graphics**.**Color**.**RED**);**

**}**

**else{**

viewHolder**.**tvw1**.**setText**(**nombre**[**p**]);**

viewHolder**.**tvw2**.**setText**(**apellido**[**p**]);**

viewHolder**.**tvw**.**setText**(**cedula**[**p**]);**

viewHolder**.**tvw3**.**setText**(**fecha**[**p**]);**

viewHolder**.**tvw4**.**setText**(**hora**[**p**]);**

viewHolder**.**tvw5**.**setText**(**gel**[**p**]);**

viewHolder**.**tvw6**.**setText**(**temp**[**p**]);**

**}**

**return** r**;**

**}**

class ViewHolder**{**

TextView tvw**;**

TextView tvw1**;**

TextView tvw2**;**

TextView tvw3**;**

TextView tvw4**;**

TextView tvw5**;**

TextView tvw6**;**

ImageView iv**;**

ViewHolder**(**View v**){**

tvw1**=(**TextView**)**v**.**findViewById**(**R**.**id**.**tvname**);**

tvw2**=(**TextView**)**v**.**findViewById**((**R**.**id**.**tvapellido**));**

tvw**=(**TextView**)**v**.**findViewById**(**R**.**id**.**tvcedula**);**

tvw3**=(**TextView**)**v**.**findViewById**(**R**.**id**.**tvfecha**);**

tvw4**=(**TextView**)**v**.**findViewById**(**R**.**id**.**tvhora**);**

tvw5**=(**TextView**)**v**.**findViewById**(**R**.**id**.**tvgel**);**

tvw6**=(**TextView**)**v**.**findViewById**(**R**.**id**.**tvtemp**);**

**}**

**}**

**}**

## **Código en PHP:**

### **Buscar:**

Es el código realizado en PHP que corre en el servidor y es usado para buscar según el valor introducido para el campo cédula. Usa la función GET para poder extraer dichos datos.

<?php

**include** 'conexion.php';

$cedula = $\_GET['Cedula'];

$consulta = "select \* from registros where Cedula='**$cedula**'";

$resultado= $conexion -> query($consulta);

**while**($fila=$resultado->fetch\_array()){

$lista[]=**array\_map**('utf8\_encode',$fila);

}

**echo** **json\_encode**($lista);

$resultado -> close();

?>

### **Insertar:**

Es el código en PHP que permite insertar los datos en la base de datos. Utiliza la función POST para poder realizar esta acción.

<?php

**include** 'conexion.php';

$cedula=$\_POST['Cedula'];

$nombre=$\_POST['Nombre'];

$apellido=$\_POST['Apellido'];

$fecha=$\_POST['Fecha'];

$hora=$\_POST['Hora'];

$gel=$\_POST['Gel'];

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

$consulta="insert into registros values('".$cedula."','".$nombre."', '".$apellido."','".$fecha."','".$hora."','".$gel."','".$temperatura."')";

**mysqli\_query**($conexion,$consulta) **or** **die** (**mysqli\_error**());

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

?>

### **Mostrar:**

Es el código en PHP utilizado para mostrar los datos de la base. Crea una lista de los datos y los codifica en una lista tipo JSON para que puedan ser leídos y recorridos en el Android Studio.

<?php

**include** 'conexion.php';

$query = **mysqli\_query**($conexion,"SELECT \* FROM registros");

**if**($query){

**while**($row = **mysqli\_fetch\_array**($query)){

$flag[]=$row;

}

**print**(**json\_encode**($flag));

}

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

?>

### **Conexión:**

Es el código en PHP que realiza la conexión con la base de datos.

<?php

$hostname = 'localhost';

$database = 'proyecto';

$username = 'root';

$password = '';

$conexion=**new** mysqli($hostname,$username,$password,$database);

**if**($conexion->connect\_errno){

**echo** "Se estan experimentando problemas";

}

?>

# Análisis de presupuesto

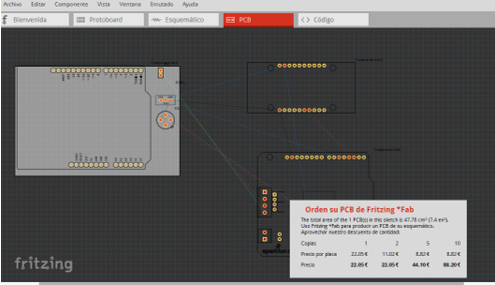


Ilustración 16: Diseño en PCB

Para poder darle un toque más profesional al proyecto se ha utilizado la herramienta generadora de PCB de Fritzing, la cual también da el precio estimado del PCB a realizar. Se puede observar en la ilustración 16 que se da un precio estimado de 22.05 €, lo cual en dólares equivale a $26.13. Si a esto se le suma la realización de la app con un entorno y herramientas más profesionales se podría tener un coste alrededor de los $500 para poder realizar el producto de forma profesional. A esto se le deben sumar los costos operativos en gastos por servidor en la nube y materia prima para hacer la carcasa del hardware y que así funcionen en conjunto. En teoría la electrónica para este proyecto no es tan cara y es muy rentable si se quisiera hacerlo comercializable a una gran escala debido a que los gastos por la aplicación móvil solo se realizan una vez y dependiendo de los cambios que se le hagan conforme pase el tiempo.

# Conclusiones

En conclusión, el desarrollo de este proyecto contribuye satisfactoriamente a la prevención y control sanitario contra el virus COVID-19. Al tener una aplicación móvil que permita ver en tiempo real el registro de personas ingresadas y sus temperaturas corporales es posible sentirse en un ambiente más seguro. Inclusive el mostrar los datos de las personas con temperatura elevada de color rojo ayuda a mejorar la prevención ya que los seres humanos son visuales y el color rojo funciona como una advertencia.

Al tener un sistema automatizado se puede administrar de una mejor forma la fuerza laboral del establecimiento y se puede prescindir de tener a una persona en la entrada tomando la temperatura constantemente. También el sistema automatizado puede llegar a ser más preciso debido a que no presenta fallas ni errores humanos.

Los servicios web dentro de las aplicaciones móviles son de gran utilidad debido a que permiten presentar y manejar datos que se encuentran en un servidor remoto. Al permitir esto se puede tener una mayor flexibilidad y control de la información para la aplicación, ya que este es un modelo que usan la mayoría de las aplicaciones móviles en la actualidad. Para lograr este funcionamiento es necesario implementar la librería Volley y permitir los permisos de acceso al internet dentro de Android Studio.

# Referencias bibliográficas

* Beltran, A. A., Clavero, A. R., De Vera, J. M. B., Lopez, P. A. P., Mueca, C. A., IX, N. A. P., & Roxas, A. Z. D. (2015). Arduino-based Food and Water Dispenser for Pets with GSM Technology Control. International Journal of Scientific Engineering and Technology, 4(4), 231-234.
* Manzaneda, E. M. (2017). ARQUITECTURA DE APLICACIONES ANDROID-WEB BASADA EN LA LIBRERÍA VOLLEY (Doctoral dissertation, UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE).
* Lucas Herrera, K. L., & Ulloa Yagual, A. E. (2018). Desarrollo de una aplicación web y móvil basada en PHP y android para automatizar y monitorear las gestiones administrativas y operativas de una Cooperativa de Taxis ubicada en la Ciudad de Guayaquil (Doctoral dissertation, Universidad de Guayaquil. Facultad de Ciencias Matemáticas y Físicas. Carrera de Ingeniería en Sistemas Computacionales).
* Olmedo Toscano, P. D. L. A., & Angamarca Rodríguez, M. E. (2016). Análisis Comparativo de Entornos de Desarrollo Android Studio y Android SDK para el Desarrollo de Aplicaciones Cliente Móvil (Bachelor's thesis, Escuela Superior Politécnica de Chimborazo).