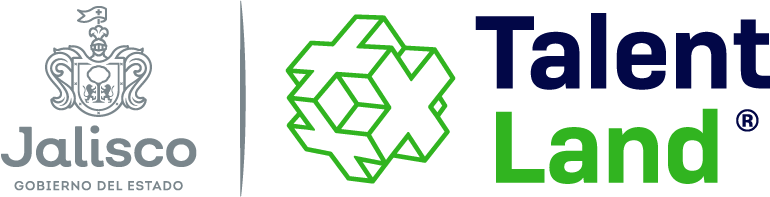
"Por un mundo menos contaminado"

Talent Land 2019

[](https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&ved=2ahUKEwifz-v8qcjhAhWIJjQIHUsJA_sQjRx6BAgBEAU&url=https://talent-network.org/&psig=AOvVaw2tTdyTpibclUcm8BhbjBgl&ust=1555081986087836)

[](https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwiJmuzAqcjhAhWkJjQIHV-MArUQjRx6BAgBEAU&url=https://www.globalspec.com/supplier/profile/Newarkelement14&psig=AOvVaw21ALV2Vwyk-_URWbyYE1jh&ust=1555081854249486)

Idea de proyecto

Alejandro García Arana

Carlos Uriel García Villalobos

Tabla de  
Contenido

Tabla de contenido 1

**Registro de equipos y capitán** 2

Integrantes del equipo 3

Capitán designado 3

Motivación para participar en el reto 3 y 4

**Etapa 1. – 12 de abril** 5

Introducción 6

Idea del proyecto 6

Marco teórico 6-10

Evalucación de impacto ambiental 10-11

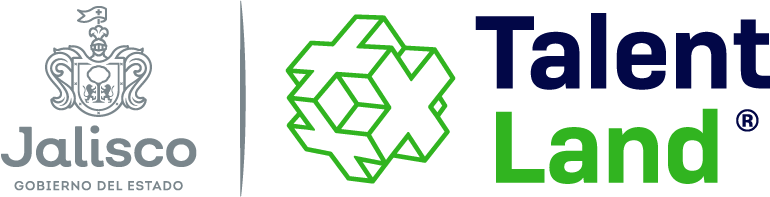
**Etapa 2. – 19 de abril** 12

Documentación del desarrollo 13

Diagramas de flujo, esquematicos, de conexión 13

Fritzings videos 13

**Registro de equipos y capitán**

[](https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&ved=2ahUKEwifz-v8qcjhAhWIJjQIHUsJA_sQjRx6BAgBEAU&url=https://talent-network.org/&psig=AOvVaw2tTdyTpibclUcm8BhbjBgl&ust=1555081986087836)

## Integrantes del equipo:

- Alejando García Arana

Número telefónico (y WhatsApp): +52 33 2669 2294

Correo electrónico: alex.grc.18@gmail.com

Facebook: https://www.facebook.com/alejandroGA0519

- Carlos Uriel García Villalobos

Número telefónico (y WhatsApp): +52 33 3947 9885

Correo electrónico: carlos111199@gmail.com

Facebook: https://www.facebook.com/Te.para3.1999

## capitan designado:

- Carlos Uriel García Villalobos

## Motivacion para participar en el reto:

- Alejandro García Arana

La tecnología en la última década ha estado en gran desarrollo y ha tenido un gran crecimiento, al igual que los grandes problemas ambientales que tenemos actualmente, se necesita actuar en dichos problemas ya que están generando grandes consecuencias al ser humano, en el clima, en el ambiente, en la flora y fauna, y que si no se actúa rápido las consecuencias serán irreversibles, por ello, me complace participar en este reto que ayuda a disminuir el problema haciendo uso de las herramientas tecnológicas que tenemos hoy en día.

- Carlos Uriel García Villalobos

El hecho de participar en un evento de enfoque tecnológico como lo es Talent Land presenta una oportunidad de ponerme a mí mismo a prueba y demostrar las habilidades que he adquirido en este campo, a su vez, me motiva la aplicación de las tecnologías requeridas para el reto, como lo es el uso de placas de desarrollo tipo Raspberry o el uso de IoT y poder darles un enfoque en un campo tan importante como lo es el cuidado del ambiente. De igual forma, creo que puedo dar orgullo y reconocimiento a mi institución, que me ha brindado la mayor parte de mis conocimientos.

**Etapa 1. – 12 de abril**

## [Resultado de imagen para talent land](https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&ved=2ahUKEwifz-v8qcjhAhWIJjQIHUsJA_sQjRx6BAgBEAU&url=https://talent-network.org/&psig=AOvVaw2tTdyTpibclUcm8BhbjBgl&ust=1555081986087836)

## Introducción

En los últimos años, la tecnología ha ido incrementado enormemente creando nuevas soluciones a diversos problemas, por ello, preocupados por el planeta y el cuidado del medio ambiente, junto con la compañía Newark, hemos iniciado el desarrollo de soluciones tecnológicas que ayude al cuidado del medio ambiente para así mejorar la calidad de vida de las personas en todo el mundo.

## Idea del proyecto

Diseñar e implementar un prototipo de un sistema informático que detecte e interprete la calidad del aire mediante el uso del sensor MQ-135, Arduino y Raspberry, y que mediante un gestor de base de datos en la nube se almacenen dichos datos recabados para sincronizarlos con una aplicación móvil y que esta pueda ser consultada por cualquier usuario con conexión a internet y un dispositivo Android, asimismo, mandar alertas a dicho dispositivo móvil si se sobrepasa el rango establecido de la calidad del aire.

## Marco teorico

Por lo general, se considera contaminación del aire a cualquier sustancia, introducida en la atmósfera por las personas, que tenga un efecto perjudicial sobre los seres vivos y el medio ambiente.

El dióxido de carbono, un gas de efecto invernadero, es el contaminante que está causando en mayor medida el calentamiento de la Tierra. Si bien todos los seres vivos emiten dióxido de carbono al respirar, éste se considera por lo general contaminante cuando se asocia con actividades humanas que requieren el uso de combustibles fósiles. Durante los últimos 150 años, estas actividades han enviado a la atmósfera una cantidad de dióxido de carbono suficiente para aumentar los niveles de éste por encima de donde habían estado durante cientos de miles de años.

El monóxido de carbono es considerado uno de los mayores contaminantes de la atmósfera terrestre. Sus principales fuentes productoras responsables de aproximadamente 80% de las emisiones, son los vehículos automotores que utilizan como combustible gasolina o Diesel y los procesos industriales que utilizan compuestos del carbono. Esta sustancia es bien conocida por su toxicidad para el ser humano.

La contaminación del aire representa un importante riesgo medioambiental para la salud. Mediante la disminución de los niveles de contaminación del aire los países pueden reducir la carga de morbilidad derivada de accidentes cerebrovasculares, cánceres de pulmón y neumopatías crónicas y agudas, entre ellas el asma.

La mayor parte de la gente está de acuerdo en que, para luchar contra el calentamiento global, se deben tomar una serie de medidas. A nivel individual, un menor uso de coches y aviones, el reciclaje y la protección del medio ambiente son medidas que reducen la huella de carbono de una persona, es decir, la cantidad de dióxido de carbono liberada a la atmósfera debido a las acciones de cada individuo.

Cuanto más bajos sean los niveles de contaminación del aire mejor será la salud cardiovascular y respiratoria de la población, tanto a largo como a corto plazo.

Es necesario evaluar de manera cuantitativa su calidad, mediante la medición de la concentración de los contaminantes que se presentan.

Los indicadores de la calidad del aire permiten, entre otras cosas, evaluar el estado de la contaminación atmosférica y comunicar al público cuál es la calidad del aire que respira.

Una de las formas para evaluar la calidad del aire es comparando las concentraciones de los contaminantes obtenidas de las redes de monitoreo con los límites máximos permisibles establecidos en las Normas Oficiales Mexicanas (NOM) publicadas por la Secretaría de Salud.

A través de este sistema se encuentran disponibles la consulta de seis contaminantes que funcionan como criterio los cuales son

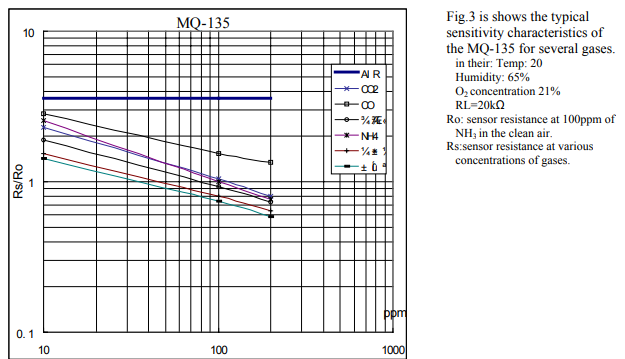
* Partículas suspendidas PM10 y PM2.5,
* Ozono (O3),
* Dióxido de azufre (SO2),
* Dióxido de nitrógeno (NO2).
* Monóxido de carbono (CO)

Para realizar un monitoreo de las condiciones del aire de una zona determinada, hemos decidido utilizar un sensor de la serie MQ ya que estos tienen la característica de detectar la presencia de distintos componentes químicos en el aire. Estos dispositivos están compuestos por un sensor electroquímico que varía su resistencia al estar en contacto con las sustancias.

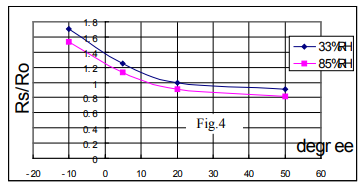
Los sensores de gases son dispositivos con alta inercia, es decir, la respuesta necesita tiempos largos para estabilizarse tras un cambio de concentración de los gases medidos. Ello es debido a la necesidad física de que el gas abandone el material sensible, lo cual es un proceso lento.

Todos los modelos MQ disponen de un calentador necesario para elevar la temperatura del sensor, y que sus materiales adquieran la sensibilidad. Mientras el calentador no alcance la temperatura de funcionamiento, la lectura del sensor no será fiable.

El sensor MQ-135 se utilizan en equipos de control de calidad del aire, tienen una alta sensibilidad al Amoníaco (NH3), Óxidos de nitrógeno (NOx), Alcohol, Sulfuros, Benceno (C6H6), Monóxido de carbono (CO), dióxido de carbono (CO2)y otros gases nocivos.



El sensor realiza una medición más precisa en temperaturas de 20°C (con rangos entre -10°C y 50°C) con humedades relativas en un rango de 33% y 85%



Los datos obtenidos del sensor serán procesador por la placa de desarrollo Rasberry Pi 3B+ el cual es un ordenador de bajo coste y tamaño reducido.

Raspberry es la placa de un ordenador simple compuesto por un SoC, CPU, memoria RAM, puertos de entrada y salida de audio y vídeo, conectividad de red, ranura SD para almacenamiento, reloj, una toma para la alimentación, conexiones para periféricos de bajo nivel, etc.

Raspberry permite mediante Python la conexión a Google Firebase.

Python es un lenguaje el cual es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible.

Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma. Es de código abierto.

### Firebase es una plataforma para el desarrollo de páginas web y aplicaciones. Cuenta con funciones de analítica y con funciones de Desarrollo como Firebase Cloud Messaging, Auth​, Storage, Firestore y Realtime Database. Esta última proporciona una base de datos en tiempo real y back-end.

## Evaluacion de impacto ambiental

Para entender un poco el contexto, se responden a las preguntas, ¿Qué? ¿Cómo? ¿Cuándo?

¿Qué? Diseñar e implementar un prototipo de un sistema informático que detecte e interprete la calidad del aire y en un gestor de base de datos en la nube se almacenen dichos datos recabados para sincronizarlos con una aplicación móvil.

¿Cómo? Mediante el uso del sensor MQ-135, Arduino y Raspberry, y algún servicio de base de datos en la nube (AWS o Firebase, pendiente por decidir) y mostrando los datos en los dispositivos Android con la aplicación instalada y conexión a internet.

¿Cuándo? Activo durante todo el día en un horario determinado, en el cual no exista mucha variación en la temperatura para mantener el sensor estable (Un ejemplo sería de 8:00am a 8:00pm). También podría ser cuando la empresa/negocio lo requiera en su negocio en un intervalo de horario establecido por la empresa/negocio.

Matriz de impacto para la evaluación de la acción a tomar cuando se tenga una calidad en el aire mala

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sistema para la detección de aire contaminado  (Alertas para cuando se detecte un alto nivel de aire contaminado) | | | | | |
| Componente ambiental | **Signo** | **Magnitud** | **Efecto** | **Persistencia** | **Alcance** |
| *Calidad mala del aire* | - | Alta | Indirecto | Media | Local |
|  | 3 | 1 | 2 | 2 |

Donde:

ALTO=3; MEDIO=2 y BAJO=1

SIGNO: positivo o negativo

MAGNITUD: alta:3, media:2 o baja:1

ALCANCE: global:3, local:2 o restringido:1

PERSISTENCIA: alta:3, media:2 o baja:1

EFECTO: directo:2 o indirecto:1

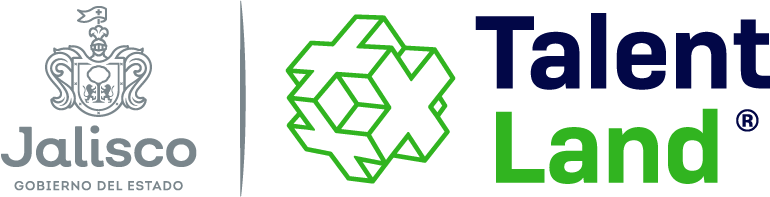
Esto quiere decir que el impacto es negativo, que posee una magnitud alta de intervención sobre la calidad mala del aire, que presenta un alcance local, que cuando se envié la alerta cuando se detecte un alto nivel de aire contaminado se espera un efecto indirecto, y dicha alerta persistirá en el tiempo por un periodo no muy largo, hasta que la calidad del aire mejore, por lo que estimamos media su persistencia.

Tenemos entonces que la ACCIÓN (Alertas para cuando se detecte un alto nivel de aire contaminado) sobre el COMPONENTE AMBIENTAL (calidad mala del aire), el impacto producido es caracterizarle como:

+AIML (las primeras letras de cada cualificación) o que numéricamente tenemos un valor de -3122 (reemplazando las letras por los números)

Ese valor -3122 no es un número entero, sino que es una ASOCIACIÓN de los valores cualitativos individuales

**Etapa 2. – 19 de abril**

[](https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&ved=2ahUKEwifz-v8qcjhAhWIJjQIHUsJA_sQjRx6BAgBEAU&url=https://talent-network.org/&psig=AOvVaw2tTdyTpibclUcm8BhbjBgl&ust=1555081986087836)

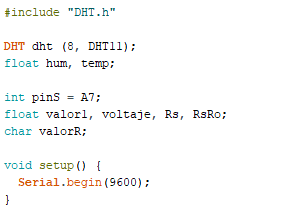
## Documentacion del desarrollo

Nuestro sistema consta de 2 partes fundamentales: la lectura y manejo de los datos arrojados por los sensores, y la interpretación de los datos por medio de una aplicación móvil.

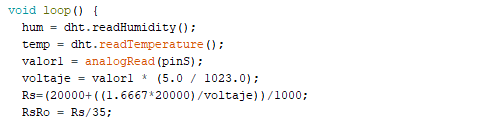
A continuación, se presenta la información recabada y la estructura de estas 2 partes.

**Lectura de los sensores MQ135 y DHT11 mediante Arduino**

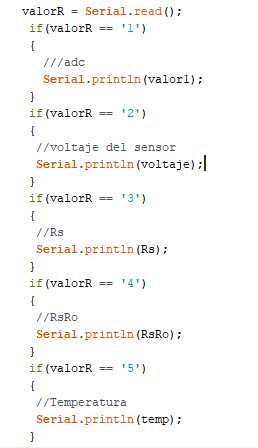
La primera etapa para la obtención de datos es un código para Arduino que realice lecturas constantes de los sensores. Se declaran las librerías necesarias, así como las variables y los pines a utilizar y se configura el cable serial a 9600 baudios.

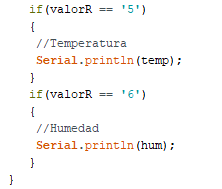


Dentro del loop, el primer paso es leer los valores que entreguen los sensores y almacenarlos en variables. Rs y RsRo son calculadas dependiendo el voltaje así como RL y Ro con valores de 20 y 35 KΩ



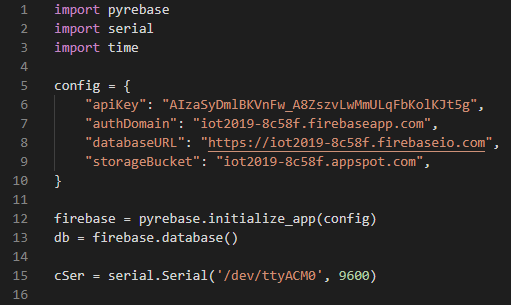
La función de valorR es almacenar una solicitud enviada por el cable serial, dependiendo del dato, se envía un valor determinado.



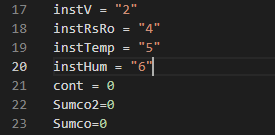


**Procesamiento de datos mediante Raspberry y carga a la nube**

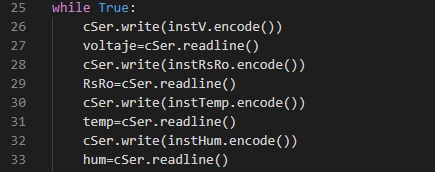
En la segunda etapa se ve involucrada la Raspberry Pi, se conecta la placa Arduino al puerto USB de Raspberry y se ejecuta el siguiente script. Se declaran las librerías necesarias y se configura la conexión a la base de datos de Firebase así como el puerto usado por Arduino.



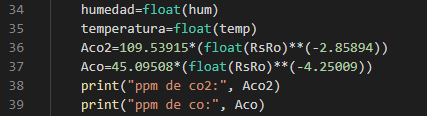
Se declaran constantes y variables necesarias

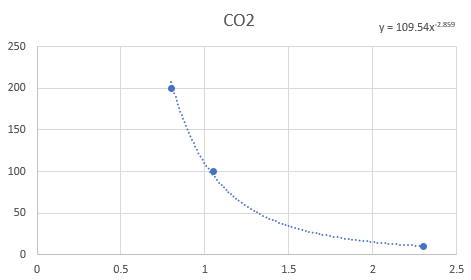


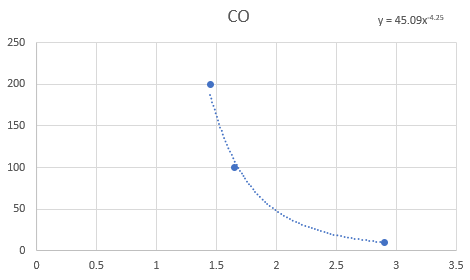
Se inicializa un bucle, se envía al puerto serial un dato que sirve de solicitud para Arduino y almacena el dato obtenido.



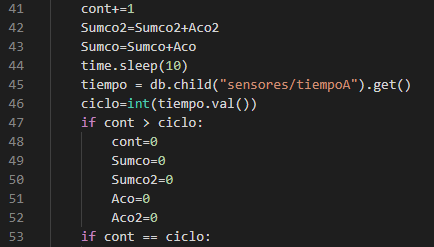
Las variables son convertidas a float con el fin de manipularlos con mayor facilidad, la RsRo que fue enviada por Arduino se convierten a ppm de CO₂ y CO según la ecuación de las siguientes gráficas.



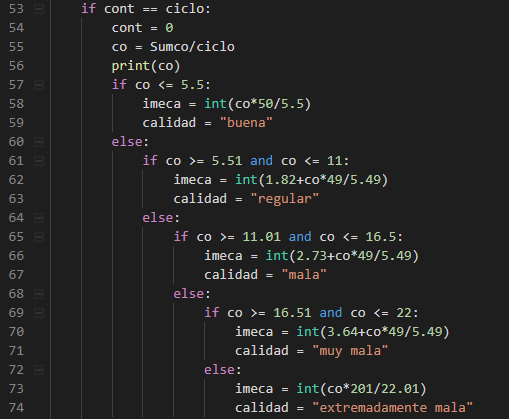




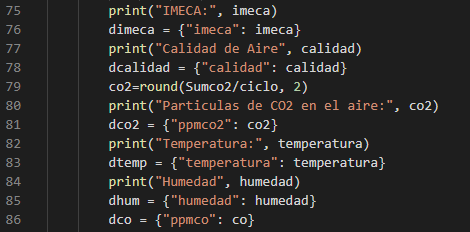
El contador de lecturas incrementa, los datos del CO₂ y CO se acumular para luego promediarlos y se obtiene el dato del tiempo de actualización de Firebase, si el contador es mayor al tiempo, este se reinicia y cuando se igualan entra en la condición.



Cuando el contador se iguala al tiempo de actualización el contador se reinicia y se promedia el CO para obtener la media en el tiempo determinado. Dependiendo la cantidad media de CO se obtiene el índice IMECA y la calidad de aire, cada uno con diferente rango y distinta ecuación.

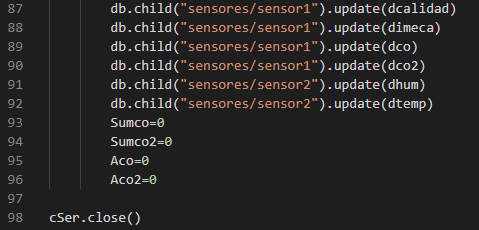


Se imprimen los datos para su control, así como se prepara cada dato para su actualización en Firebase.



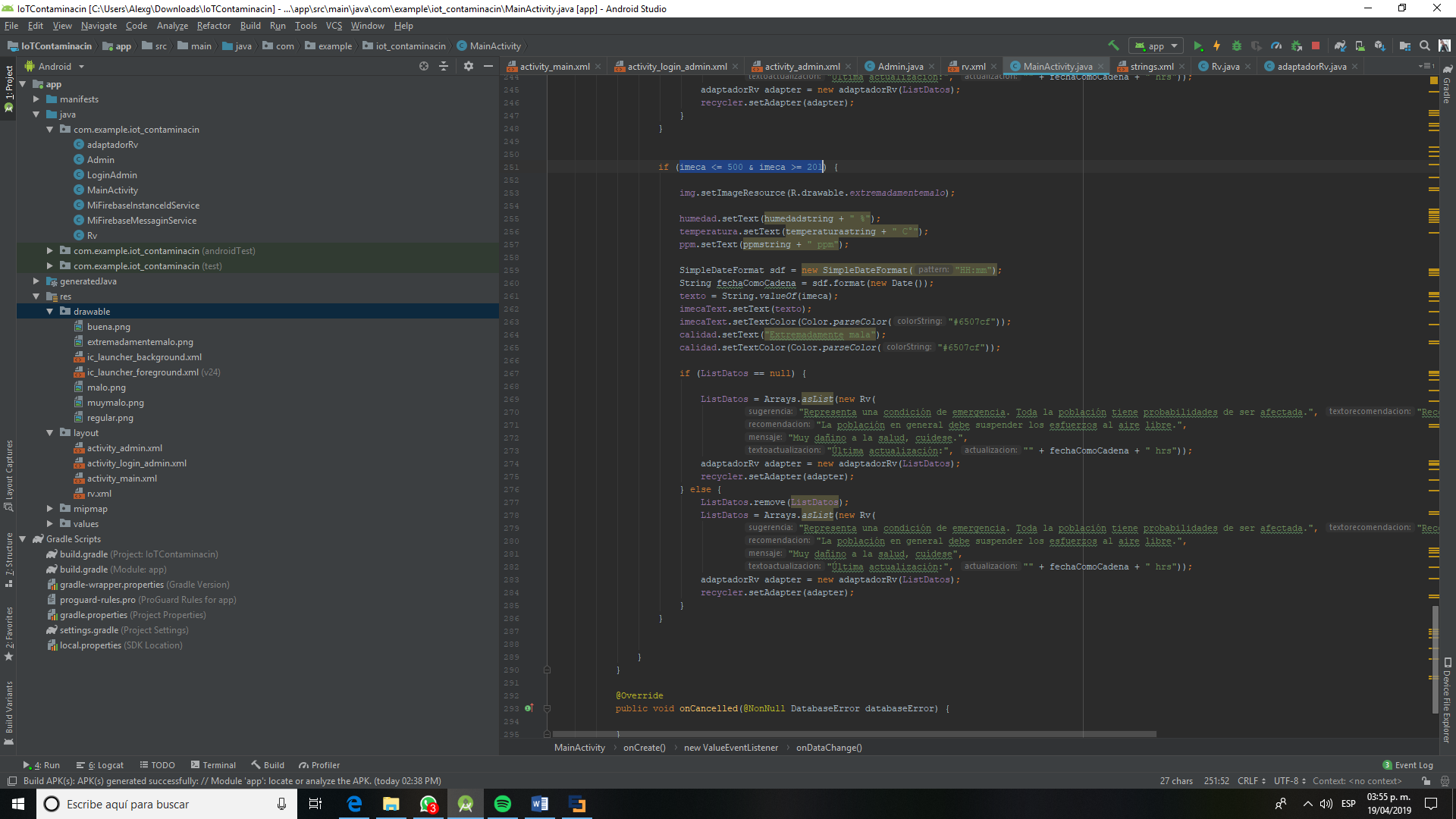
Los datos son subidos a Firebase mediante un update y se reinician las variables para iniciar nuevamente el proceso.

En caso de que se cierre el ciclo, se cierra la conexión con Arduino.

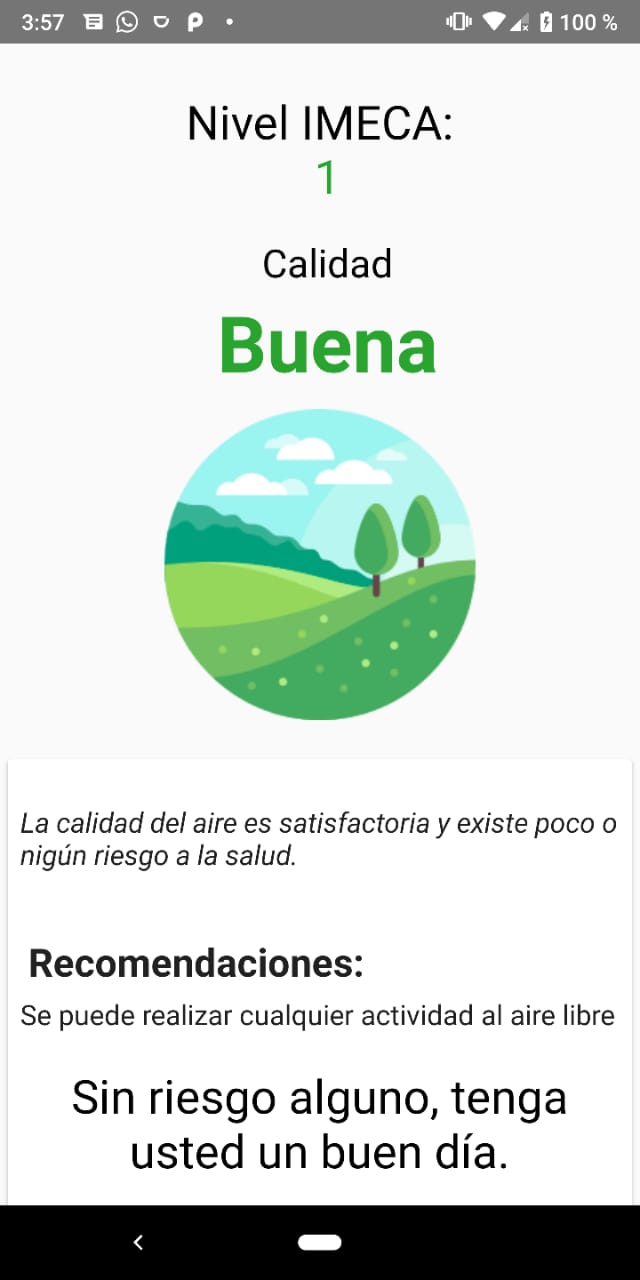
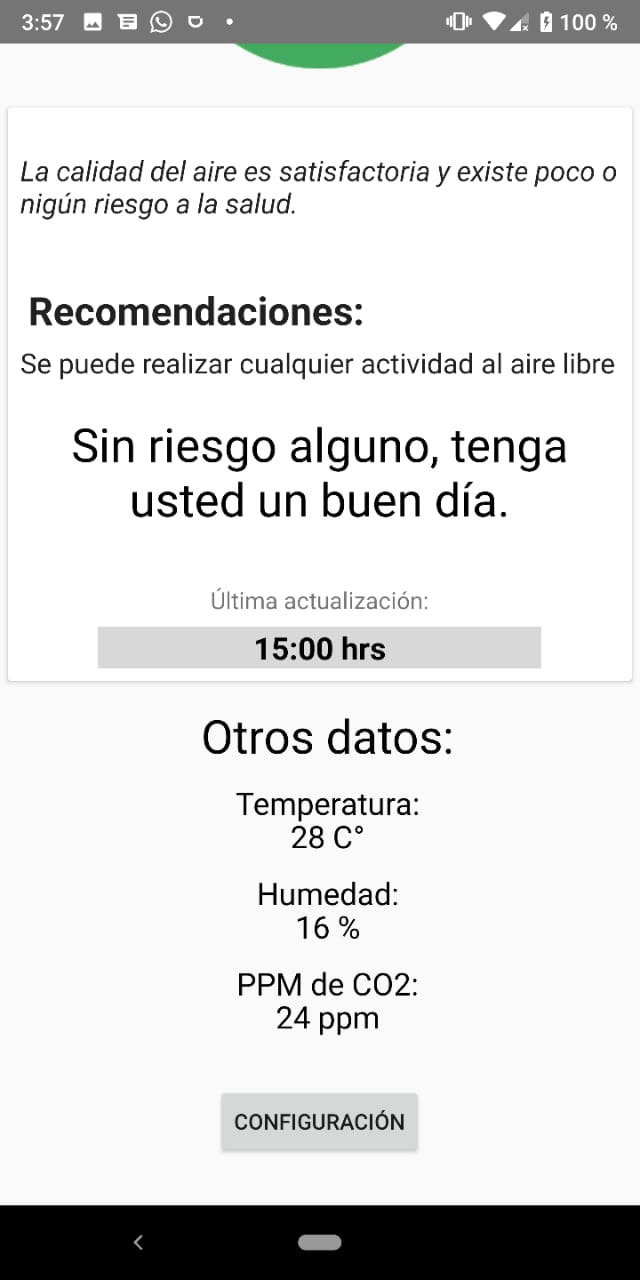


**Uso de los datos recibidos en una aplicación móvil en Android Studio**

Para la realización de la aplicación móvil, utilizamos el programa Android Studio. La aplicación está estructurada de la siguiente manera:



Principalmente tenemos la clase MainActivity, que es la que se muestra al iniciar la aplicación. Se ve de la siguiente manera:

El código de dicha clase está estructurado de la siguiente manera:

Al principio tenemos todos los imports necesarios para poder hacer las referencias correctas.

import android.content.Intent;  
import android.graphics.Color;  
import android.support.annotation.NonNull;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.support.v7.widget.LinearLayoutManager;  
import android.support.v7.widget.RecyclerView;  
import android.view.View;  
import android.widget.Button;  
import android.widget.ImageView;  
import android.widget.LinearLayout;  
import android.widget.TextView;  
import com.google.firebase.database.DataSnapshot;  
import com.google.firebase.database.DatabaseError;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.database.ValueEventListener;  
import java.text.SimpleDateFormat;  
import java.util.Arrays;  
import java.util.Date;  
import java.util.List;

Después tenemos la declaración de variables, las que se usan para referenciar a los Textview, ImageView, RecyclerView, a una lista de datos, y una para guardar texto.

private List<Rv> ListDatos;  
private RecyclerView recycler;  
private TextView imecaText, calidad, humedad, temperatura, ppm;  
private String texto;  
private ImageView img;

Enseguida referenciamos las variables a los objetos correctos e inicializamos la forma en la que queremos mostrar el recyclerview.

@Override  
protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 imecaText = (TextView) findViewById(R.id.*imeca*);  
 calidad = (TextView) findViewById(R.id.*calidad*);  
 humedad = (TextView) findViewById(R.id.*humedad*);  
 temperatura = (TextView) findViewById(R.id.*temperatura*);  
 ppm = (TextView) findViewById(R.id.*ppmco2*);  
 recycler = (RecyclerView) findViewById(R.id.*RecyclerID*);  
 img = (ImageView) findViewById(R.id.*img*);

recycler.setLayoutManager(new LinearLayoutManager(this, LinearLayoutManager.*HORIZONTAL*, false));

Ahora tenemos el evento que pasa cuando pulsamos el botón de “CONFIGURACIÓN”, que llama a un Intent para iniciar el activity del login del administrador.

Button btn = (Button) findViewById(R.id.*btn*);  
btn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
  
 Intent intent = new Intent(MainActivity.this, LoginAdmin.class);  
 startActivity(intent);  
 }  
});

Después iniciamos una referencia a la base de datos para que cuando detecte un cambio en la rama sensores, empiece a leer en tiempo real dichos datos

final DatabaseReference ref = FirebaseDatabase.*getInstance*().getReference();  
ref.child("sensores").addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {

Al iniciar dicha lectura, por cuestión de errores iniciamos una comparación que valida que existan datos en dicha referencia iniciada, y si sí hay datos empieza a guardar en las variables los datos de imeca, ppm del CO2, humedad y temperatura, datos leídos y almacenados en Firebase por medio de la Raspberry.

if (dataSnapshot.exists()) {  
  
 int imeca = dataSnapshot.child("sensor1").child("imeca").getValue(Integer.class);  
 int ppmc = dataSnapshot.child("sensor1").child("ppmco2").getValue(Integer.class);  
 int hhumedad = dataSnapshot.child("sensor2").child("humedad").getValue(Integer.class);  
 int tempp = dataSnapshot.child("sensor2").child("temperatura").getValue(Integer.class);  
 String humedadstring = String.*valueOf*(hhumedad);  
 String temperaturastring = String.*valueOf*(tempp);  
 String ppmstring = String.*valueOf*(ppmc);

Y ahora sí, tenemos principalmente 5 comparaciones para interpretar lo puntos IMECA y mandar las sugerencias y recomendaciones correctas dependiendo el nivel de IMECA: Vamos a explicar la primera ya que todas son repetitivas y sólo cambia la comparación principal:

if (imeca <= 50) {

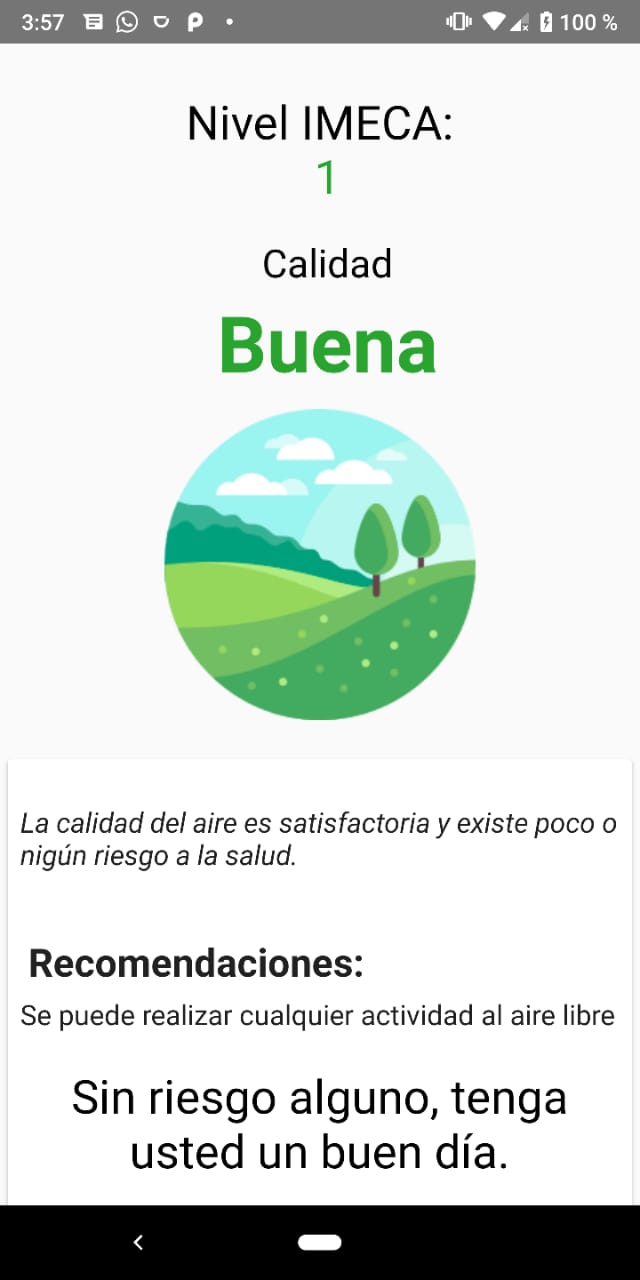
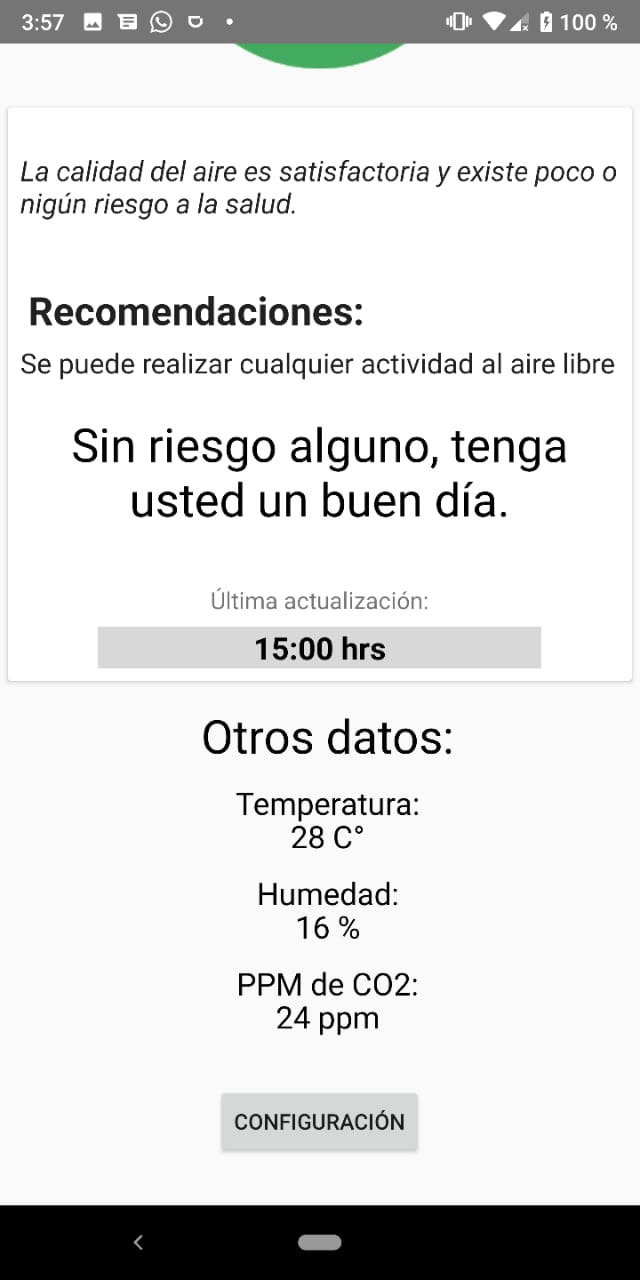
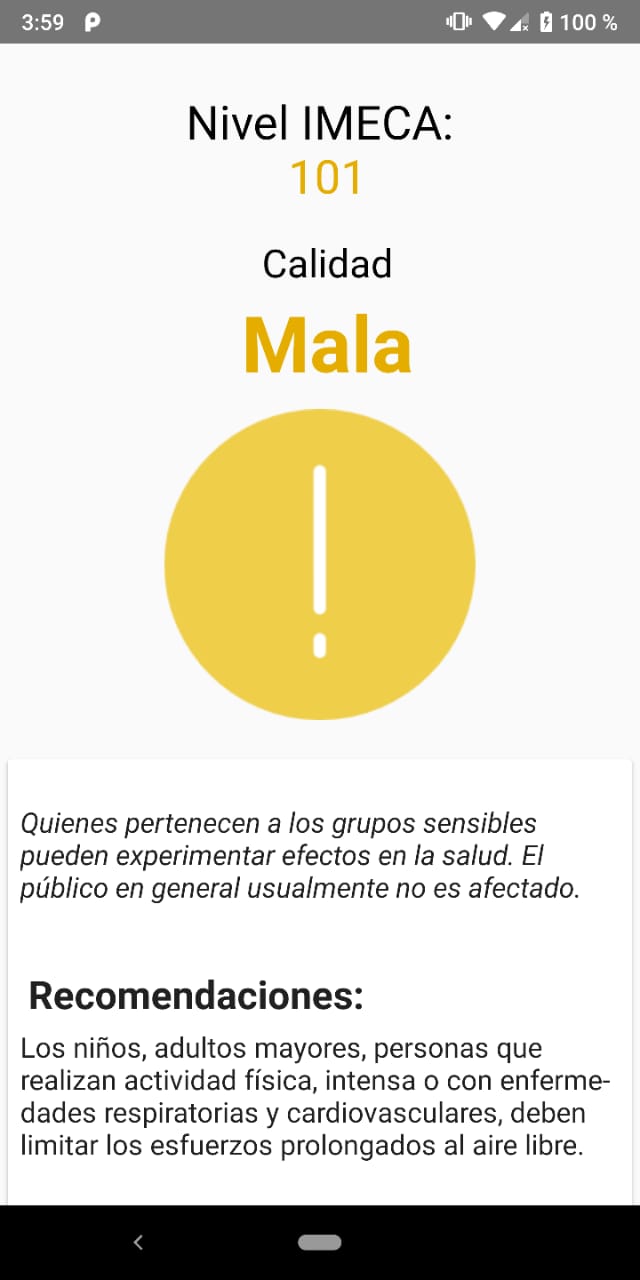
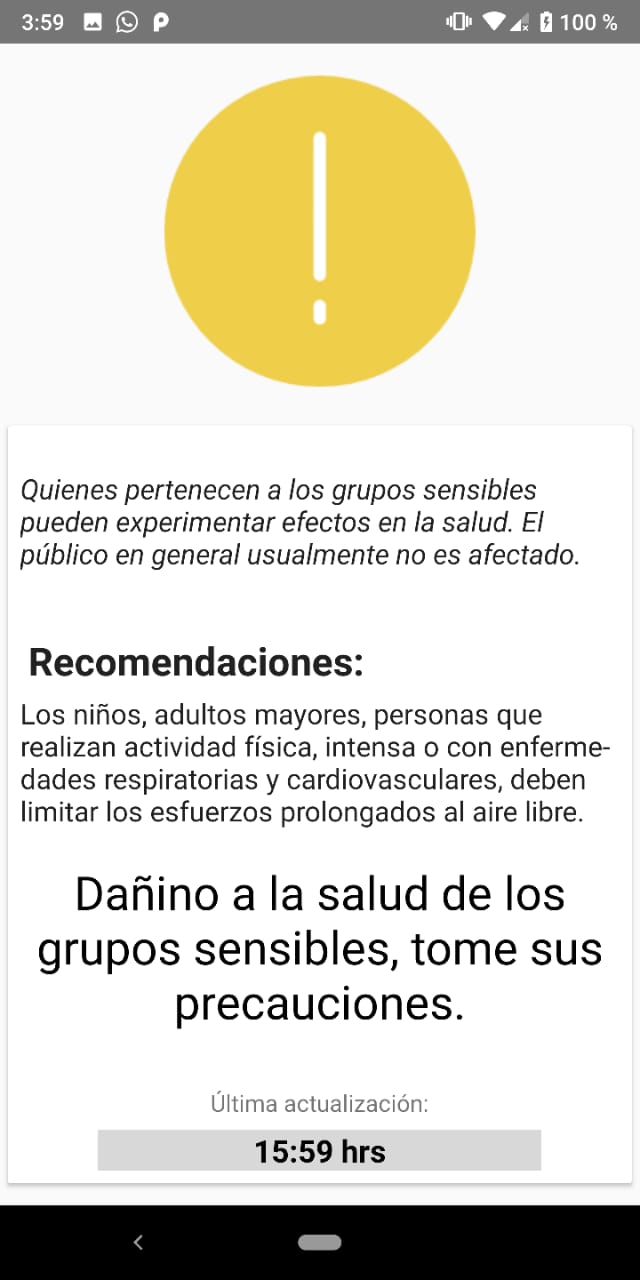
Si los puntos IMECA leídos son igual o inferiores a 50, se envía una imagen con un significado de calidad del aire buena, se muestra los puntos IMECA, la humedad, la temperatura y las ppm del CO2, después se obtiene la hora y los minutos exactos en que se realizó la lectura para mostrarla como última actualización, y se envía un color verde al texto calidad y puntos IMECA:

img.setImageResource(R.drawable.*buena*);  
humedad.setText(humedadstring + " %");  
temperatura.setText(temperaturastring + " C°");  
ppm.setText(ppmstring + " ppm");  
  
SimpleDateFormat sdf = new SimpleDateFormat("HH:mm");  
String fechaComoCadena = sdf.format(new Date());  
texto = String.*valueOf*(imeca);  
imecaText.setText(texto);  
imecaText.setTextColor(Color.*parseColor*("#29a42e"));  
calidad.setText("Buena");  
calidad.setTextColor(Color.*parseColor*("#29a42e"));

Ahora se verifica que no se haya creado ninguna lista en la variable list que creamos al inicio del programa, si sí está creada se elimina y se crea una nueva, ya que si no se elimina se crea otra lista y se van amontonando todas las sugerencias enviadas.

if (ListDatos == null) {  
  
 ListDatos = Arrays.*asList*(new Rv(  
 "La calidad del aire es satisfactoria y existe poco o nigún riesgo a la salud.", "Recomendaciones:",  
 "Se puede realizar cualquier actividad al aire libre", "Sin riesgo alguno, tenga usted un buen día.",  
 "Última actualización:", "" + fechaComoCadena + " hrs"));  
 adaptadorRv adapter = new adaptadorRv(ListDatos);  
 recycler.setAdapter(adapter);  
} else {  
  
 ListDatos.remove(ListDatos);  
 ListDatos = Arrays.*asList*(new Rv(  
 "La calidad del aire es satisfactoria y existe poco o nigún riesgo a la salud.", "Recomendaciones:",  
 "Se puede realizar cualquier actividad al aire libre", "Sin riesgo alguno, tenga usted un buen día.",  
 "Última actualización:", "" + fechaComoCadena + " hrs"));  
 adaptadorRv adapter = new adaptadorRv(ListDatos);  
 recycler.setAdapter(adapter);  
}

Y así se hace con las otras 4 comparaciones, las cuales muestran las siguientes sugerencias:



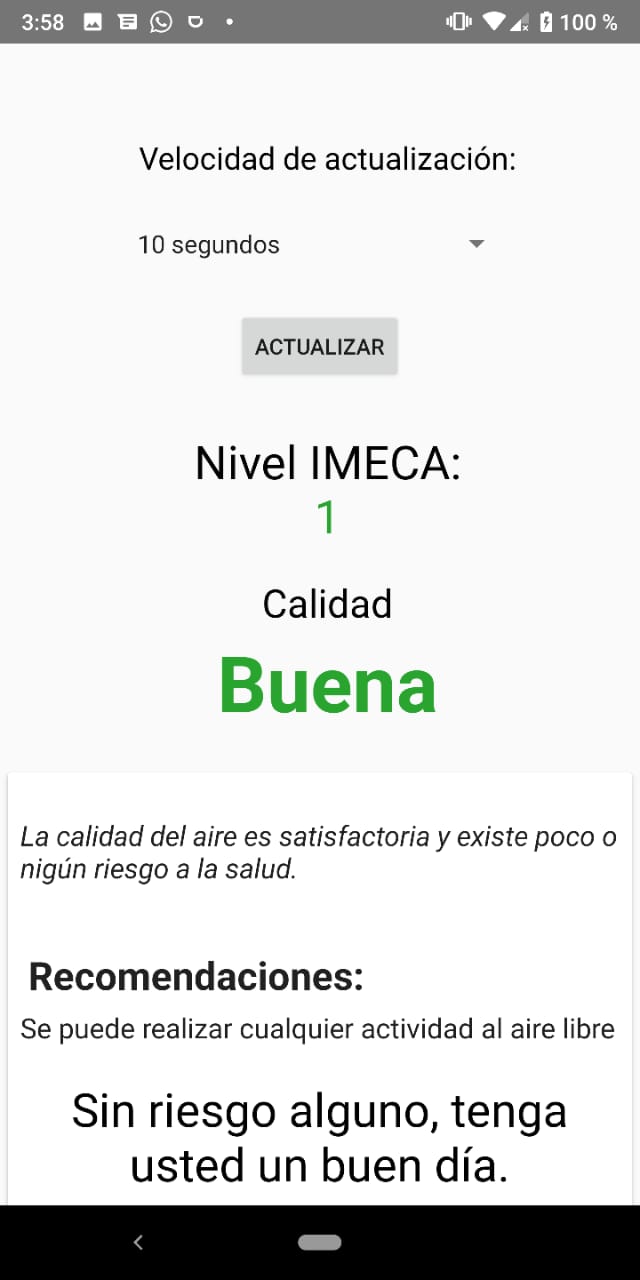
Cuando se presione el botón de “CONFIGURACIÓN” se abre el siguiente Activity:



El cual es un login sencillo, te pide usuario y contraseña y en caso de ser incorrecto manda un mensaje diciendo “Usuario o contraseña incorrectas”.

Dicha validación se hace de la siguiente manera cuando se presione el botón “ENTRAR”:

public void login (View view)  
{  
 EditText usuario = (EditText) findViewById(R.id.*etusu*);  
 EditText contra = (EditText) findViewById(R.id.*et2*);  
 if (usuario.getText().toString().trim().equalsIgnoreCase(""))  
 usuario.setError("Ingresa usuario");  
 if (contra.getText().toString().trim().equalsIgnoreCase(""))  
 contra.setError("Ingrese contraseña");  
  
 if (usuario.getText().toString().equals("admin") && contra.getText().toString().equals("admin1")) {  
 Intent intent = new Intent(this, Admin.class);  
 startActivity(intent);  
 } else {  
 Toast toast1 = Toast.*makeText*(this, "Usuario o contraseña incorrectos", Toast.*LENGTH\_SHORT*);  
 toast1.show();  
 }  
  
  
}

Si los datos son correctos se manda al Activity del admin, dicho activity se ve así:

A diferencia del primer Activity, este tiene un spinner que muestra las 6 distintas posibilidades de la velocidad de actualización de los datos.

## 

Se tiene que activar dicha función cuando se presione el botón. Dicha actualización se hace de la siguiente manera:

Button btn = (Button) findViewById(R.id.*btn*);  
btn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
  
 ObtenerSpinner();  
 }  
});

El botón llama a la función ObtenerSpinner(); la cual dependiendo la selección en el spinner manda a Firebase un valor, dicho valor es leído por Arduino y modifica su velocidad de lectura y envió de datos.

private void ObtenerSpinner() {  
 Spinner spinnertipo = (Spinner) findViewById(R.id.*spiner*);  
 String mesa = spinnertipo.getSelectedItem().toString();  
  
 FirebaseDatabase database;  
 database = FirebaseDatabase.*getInstance*();  
 DatabaseReference inventerio = database.getReference().child("sensores").child("tiempoA");  
  
  
 if (mesa.equals("10 segundos")){  
 inventerio.setValue(1);  
 Toast.*makeText*(getApplicationContext(),"Lectura cambiada a cada 10 segundos",Toast.*LENGTH\_SHORT*).show();  
 }  
  
 if (mesa.equals("30 segundos")){  
 inventerio.setValue(3);  
 Toast.*makeText*(getApplicationContext(),"Lectura cambiada a cada 30 segundos",Toast.*LENGTH\_SHORT*).show();  
 }  
 if (mesa.equals("1 minuto")){  
 inventerio.setValue(6);  
 Toast.*makeText*(getApplicationContext(),"Lectura cambiada a cada minuto",Toast.*LENGTH\_SHORT*).show();  
 }  
 if (mesa.equals("5 minutos")){  
 inventerio.setValue(30);  
 Toast.*makeText*(getApplicationContext(),"Lectura cambiada a cada 5 minutos",Toast.*LENGTH\_SHORT*).show();  
 }  
 if (mesa.equals("15 minutos")){  
 inventerio.setValue(90);  
 Toast.*makeText*(getApplicationContext(),"Lectura cambiada a cada 15 minutos",Toast.*LENGTH\_SHORT*).show();  
 }  
 if (mesa.equals("30 minutos")){  
 inventerio.setValue(180);  
 Toast.*makeText*(getApplicationContext(),"Lectura cambiada a cada 30 minutos",Toast.*LENGTH\_SHORT*).show();  
 }  
  
  
}

## diagramas de flujo, esquematicos, de conexion

A continuación, se muestra un diagrama de flujo del funcionamiento del sistema.

Diagrama para Raspberry

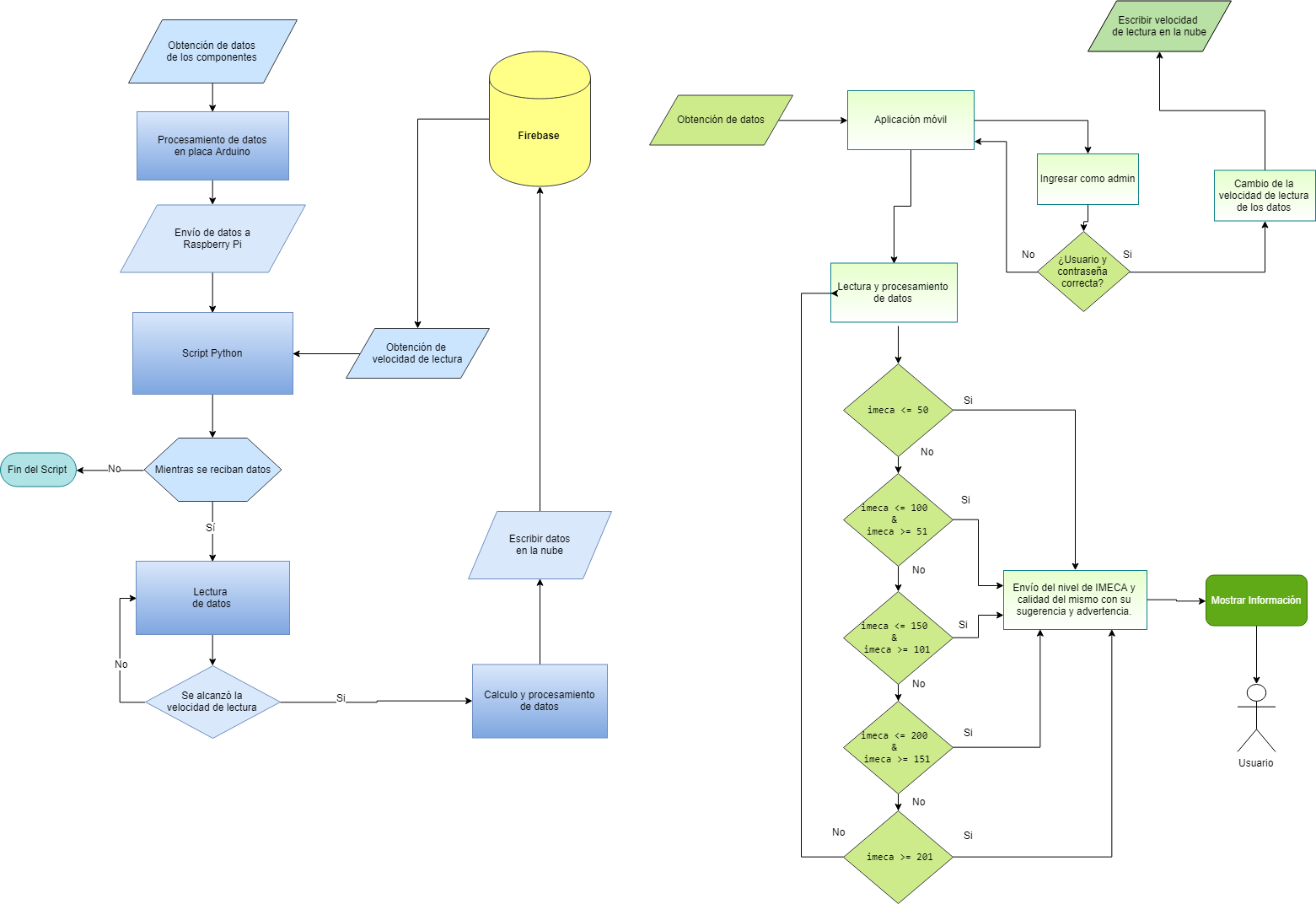
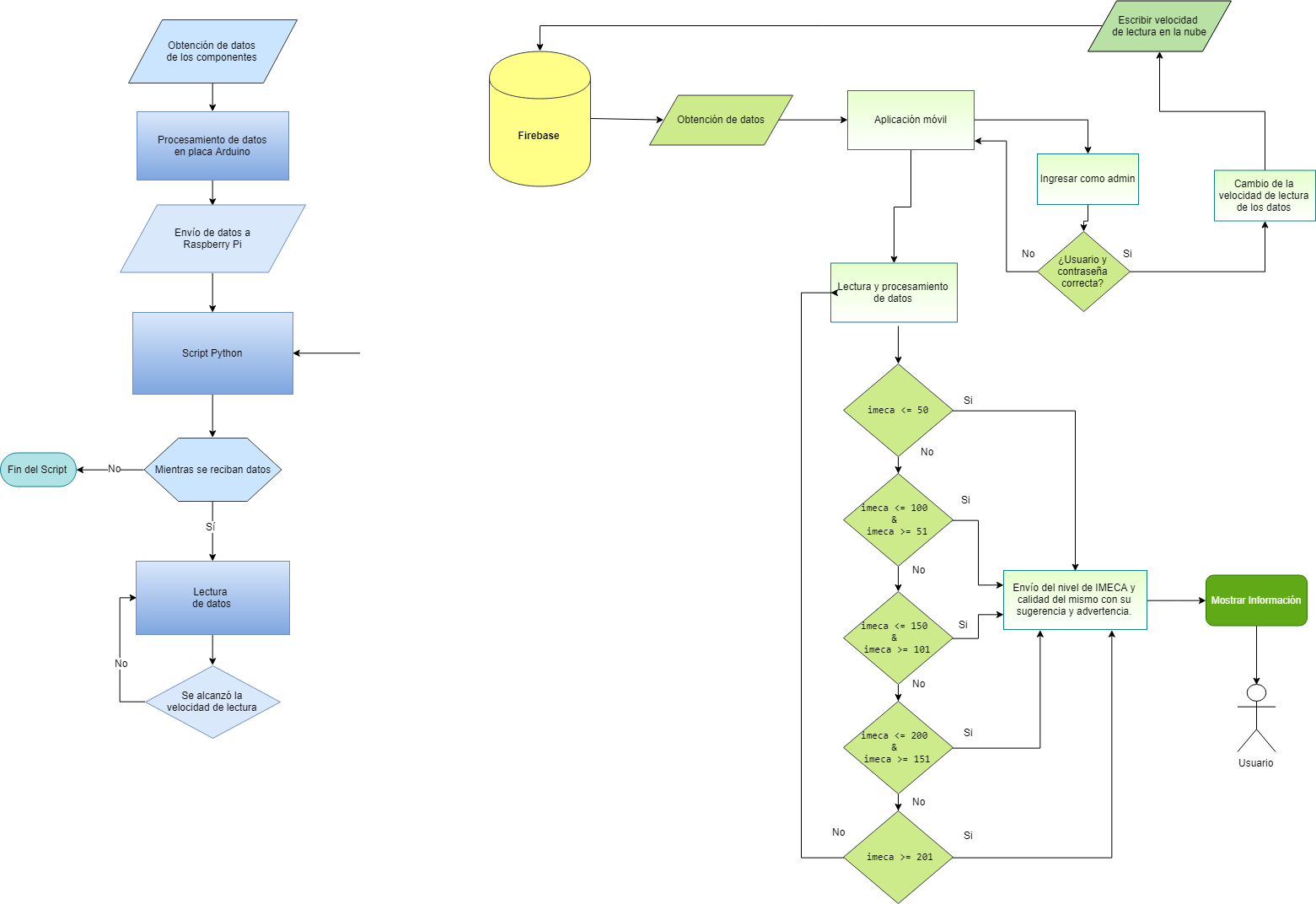
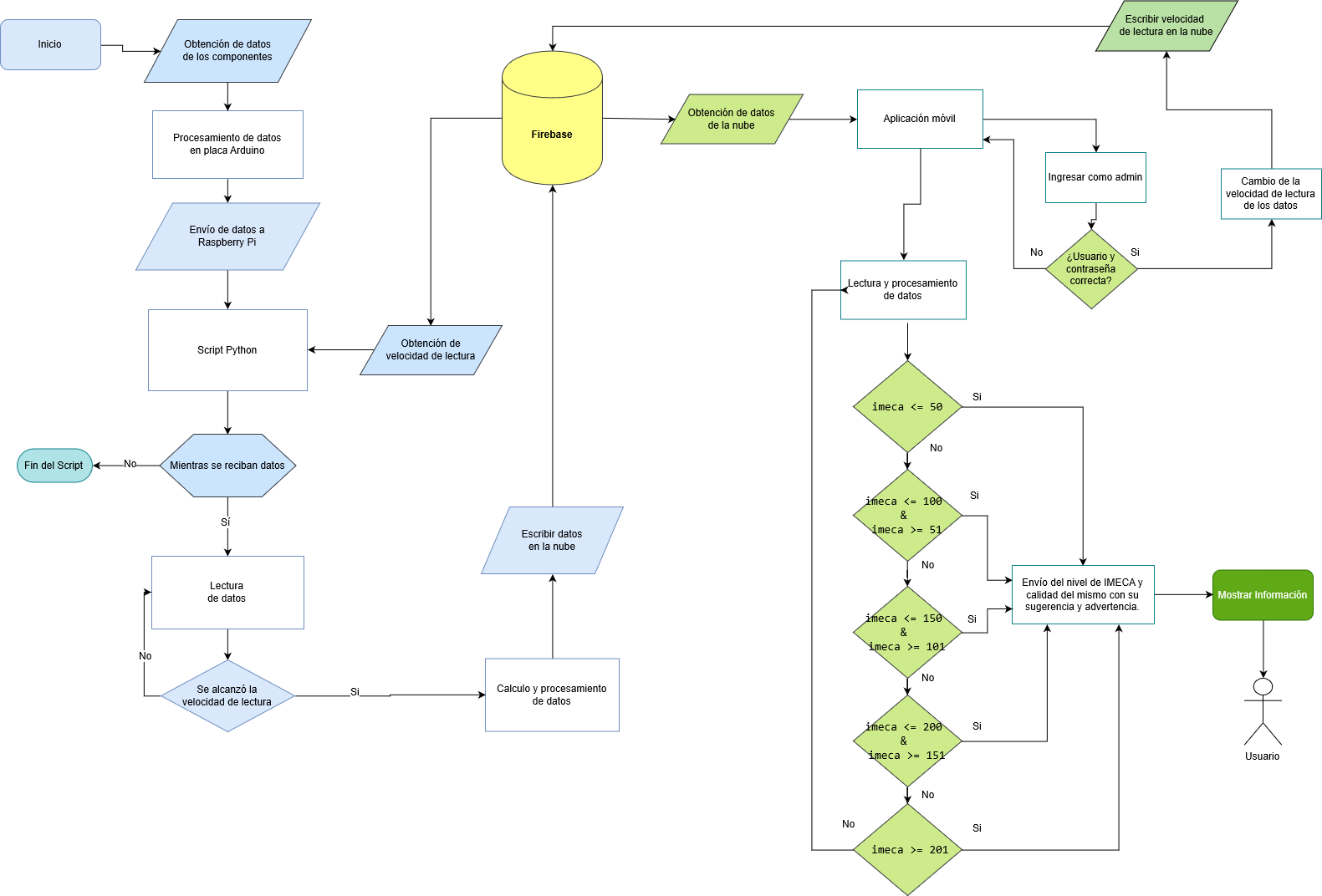


Diagrama para App de Android





## fritzings

Diseño del circuito para Arduino

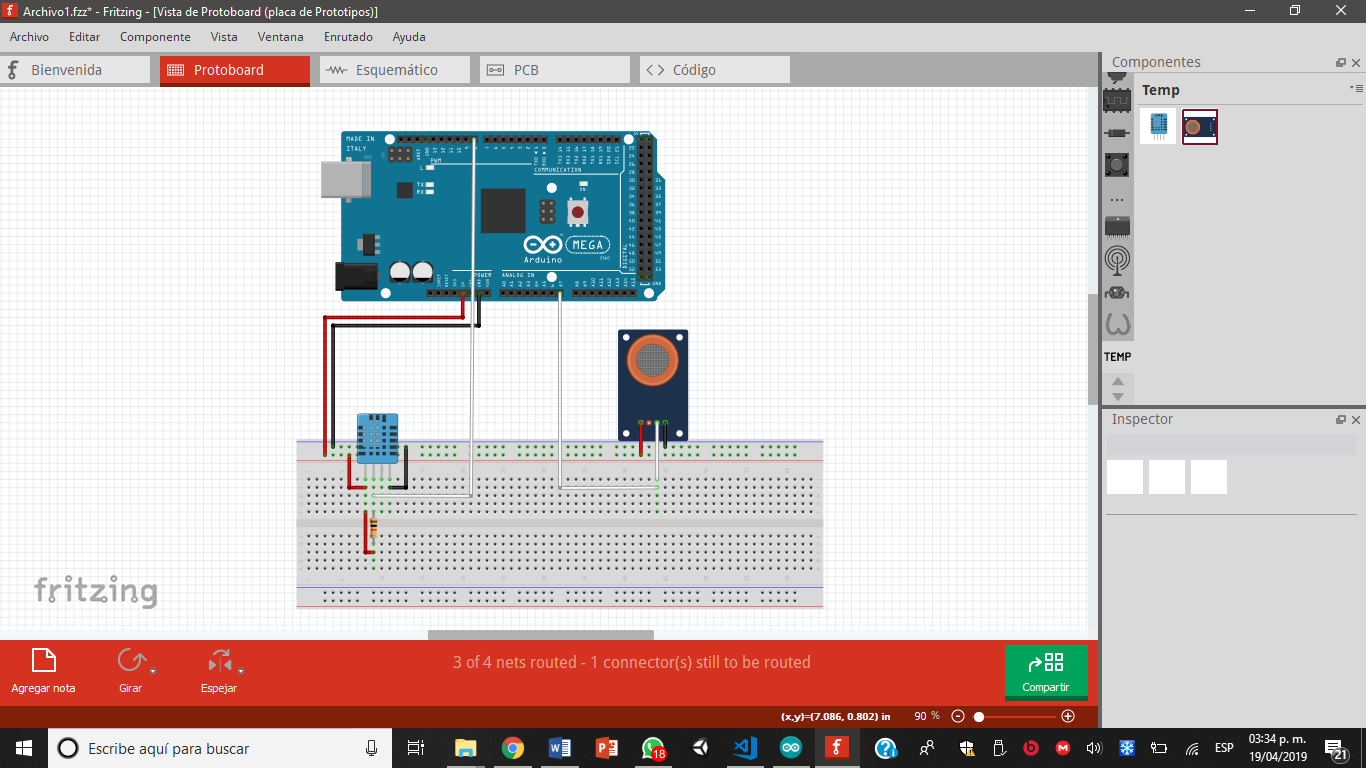


Diagrama esquemático de la conexión a Arduino

