

**FACULTAD DE INGENIERÍA ELÉCTRICA E  
INFORMÁTICA  
FIEC - ESPOL**

**ITINERARIO DE APLICACIONES MÓVILES Y  
SERVICIOS TELEMÁTICOS**

**Grupo 1**

*“Manual Técnico – Manual de la aplicación para transporte de  
alimentos”*

**MEMBERS:**

- Alex Mora
- Andrés Luna
- Debbie Donoso
- Patricio Cepeda

**CURSO: 1**

**PROFESORA:** Ing. Adriana Collaguazo

**ENLACE GITHUB:**

[https://github.com/amst-fiec/1T2022-Transporte-de-alimentos-  
v1.0](https://github.com/amst-fiec/1T2022-Transporte-de-alimentos-v1.0)

**I PAO – 2022**

# CONTENIDO

|                                                   |           |
|---------------------------------------------------|-----------|
| <b>1. INTRODUCCIÓN.....</b>                       | <b>3</b>  |
| <b>2. PRE-REQUISITOS.....</b>                     | <b>4</b>  |
| <b>2.1. HARDWARE.....</b>                         | <b>4</b>  |
| <b>2.1.1. Microcontrolador Arduino UNO .....</b>  | <b>4</b>  |
| <b>2.1.2. Módulo GSM .....</b>                    | <b>5</b>  |
| <b>2.1.3. Módulo GPS.....</b>                     | <b>5</b>  |
| <b>2.2. SOFTWARE.....</b>                         | <b>6</b>  |
| <b>2.2.1. ArduinoIDE .....</b>                    | <b>6</b>  |
| <b>2.2.2. Android Studio .....</b>                | <b>6</b>  |
| <b>2.2.3. FireBase.....</b>                       | <b>7</b>  |
| <b>3. CÓDIGOS.....</b>                            | <b>8</b>  |
| <b>3.1. ARDUINO.....</b>                          | <b>8</b>  |
| <b>3.2. ANDROID STUDIO .....</b>                  | <b>10</b> |
| <b>4. RESULTADOS.....</b>                         | <b>14</b> |
| <b>4.1. SCREENSHOT DE LA APP.....</b>             | <b>14</b> |
| <b>4.2. SCREENSHOT DE LA APP – USUARIO .....</b>  | <b>15</b> |
| <b>4.3. SCREENSHOT DE LA APP – CONDUCTOR.....</b> | <b>18</b> |
| <b>5. DIAGRAMAS.....</b>                          | <b>19</b> |
| <b>5.1. DIAGRAMA DEL PROYECTO .....</b>           | <b>19</b> |
| <b>5.2. DIAGRAMA DE CIRCUITO .....</b>            | <b>20</b> |
| <b>5.3. DIAGRAMA DE ÁRBOL .....</b>               | <b>21</b> |
| <b>5.4. DIAGRAMA DE CLASES.....</b>               | <b>22</b> |
| <b>5.5. DIAGRAMA DE CASOS DE USO .....</b>        | <b>23</b> |
| <b>5.6. DIAGRAMA DE DESPLIEGUE.....</b>           | <b>24</b> |
| <b>6. BIBLIOGRAFÍA.....</b>                       | <b>25</b> |

# 1. INTRODUCCIÓN

Una de las principales y más importantes necesidades para la población centralizada en los diferentes países es el transporte de productos alimenticios tanto primarios como elaborados, esto debido a que tienen una vida útil relativamente corta, motivo por el cual las diferentes empresas productoras de alimentos buscan que el transporte de éstos sea eficiente, rápido y seguro para los mismos. Uno de los principales problemas con el transporte de productos alimenticios es el retraso constante de conductores a los determinados puntos de entregas, esto a su vez genera otro problema derivado que es aumento de costo por refrigeración de los productos alimenticios ya que hablando en términos numéricos se estima que por cada hora de retraso en la entrega se gasta alrededor de 0.8 galones de gasolina lo que se traduce en 3.9 dólares de pérdida por hora (esto en Estados Unidos pero el costo varía dependiendo de cada país). Por este motivo proponemos una aplicación móvil para el control minuto a minuto de las rutas de los camiones de entrega mediante GPS, esta aplicación permitirá monitorear la ubicación en tiempo real de los camiones, además de que los conductores podrán conocer con exactitud cuales son las paradas que tienen asignadas y así también el administrador podrá saber cuanto tiempo demorará el camión en llegar a su destino.

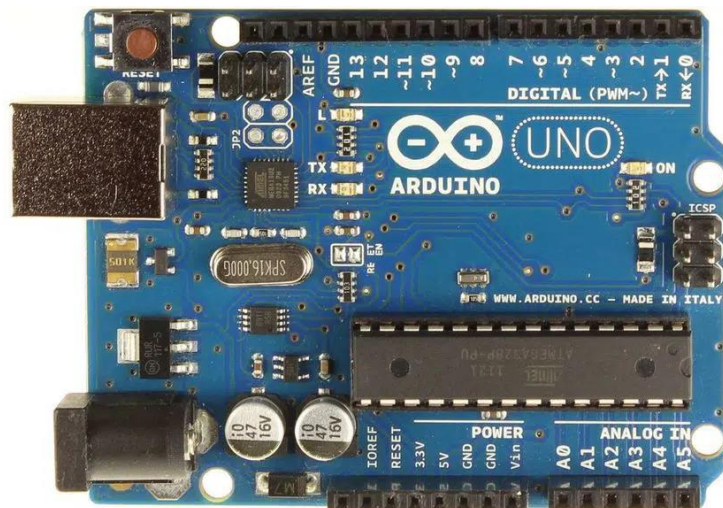
## 2. PRE-REQUISITOS

### 2.1. HARDWARE

#### 2.1.1. Microcontrolador Arduino UNO

Arduino Uno es una placa de microcontrolador basada en ATmega328P (hoja de datos). Dispone de 14 pines de entrada/salida digital (6 de los cuales pueden utilizarse como salidas PWM), 6 entradas analógicas, un resonador cerámico de 16 MHz (CSTCE16M0V53-R0), conexión USB, toma de corriente, conector ICSP y botón de reset. Contiene todo lo necesario para soportar el microcontrolador; simplemente conéctelo a la computadora con un cable USB o enciéndalo con un adaptador AC-DC o una batería para comenzar. Puedes hacer tú mismo tu Uno sin preocuparte demasiado por hacer algo mal, en el peor de los casos puedes reemplazar el chip por unos cuantos dólares y empezar de nuevo.

"Uno" significa uno en italiano y fue elegido para marcar el lanzamiento del software Arduino (IDE) 1.0. La placa Uno y el software Arduino (IDE) versión 1.0 fueron las versiones de referencia de Arduino, ahora evolucionadas a versiones más nuevas. La placa Uno es la primera de una serie de placas Arduino USB y el modelo de referencia para la plataforma Arduino; Para obtener una lista completa de placas actuales, antiguas u obsoletas, consulte el índice de placas Arduino. (Store Arduino, 2020)



### 2.1.2. Módulo GSM

Si se necesita comunicar Arduino con el mundo exterior, pero no se tiene Ethernet o Wi-Fi, se puede usar un módulo GSM / GPRS con Tarjeta SIM, capaz de comunicarse con Arduino como si fuera un teléfono móvil. Y es que esta placa basada en el módulo SIM900 permite enviar y recibir llamadas y SMS y conectarse a Internet, convirtiendo el Arduino en un móvil. (Pormetek, 2018)



### 2.1.3. Módulo GPS

El Sistema de Posicionamiento Global (GPS) utiliza señales enviadas por satélites espaciales terrestres y estaciones terrestres para determinar con precisión su posición en la Tierra. El módulo receptor GPS NEO-6M utiliza la comunicación USART para comunicarse con el microcontrolador o terminal de PC. Obtiene información como latitud, longitud, altitud, hora UTC, etc. (Descubre Arduino, 2020)



## **2.2. SOFTWARE**

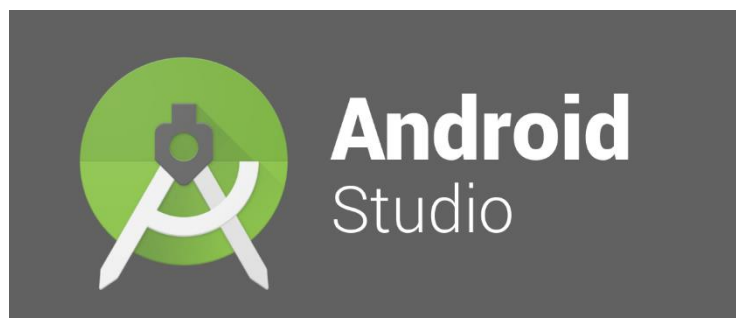
### **2.2.1. ArduinoIDE**

IDE - entorno de desarrollo integrado IDE - entorno de desarrollo integrado, llamado IDE (entorno de desarrollo integrado), es un programa informático que consta de un conjunto de herramientas de programación. Puede estar dedicado exclusivamente a un único lenguaje de programación o puede ser utilizado por varios. O, denominado IDE (siglas anglosajonas de entorno de desarrollo integrado), es un programa informático compuesto por un conjunto de herramientas de programación. Puede centrarse exclusivamente en un lenguaje de programación o utilizar varios. (Aprendiendo Arduino, 2020)



### **2.2.2. Android Studio**

Android Studio es el entorno de desarrollo integrado (IDE) oficial para desarrollar aplicaciones de Android y se basa en IntelliJ IDEA. Junto con el potente editor de código y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece aún más funciones que mejoran la productividad al desarrollar aplicaciones de Android. (Android Developer, 2022)



### **2.2.3. FireBase**

Firebase es una plataforma móvil creada por Google, cuya función principal es desarrollar y facilitar la creación de aplicaciones de alta calidad rápidamente, con el fin de aumentar la base de usuarios y ganar más dinero. La plataforma está subida a la nube y está disponible para diferentes plataformas como iOS, Android y Web. Contiene varias funciones que permiten a cualquier desarrollador mezclar y combinar la plataforma según sus necesidades.



## 3. CÓDIGOS

### 3.1. ARDUINO

```
#include <SoftwareSerial.h>

#define WIFI_SSID "NETLIFE-Luna"
#define WIFI_PASSWORD "XXXXXXXXXX"
#define rxPin 7
#define txPin 8
#include <SoftwareSerial.h>
const int pulsadorPin1 = 1;
const int pulsadorPin2 = 2;

int valorPulsador1 = 0;
int valorPulsador2 = 0;
SoftwareSerial gps(4,3);

char dato;

SoftwareSerial sim800(txPin, rxPin);

const char FIREBASE_HOST = "https://transporte-alimentos-gl-astm-default-rtdb.firebaseio.com/";
const String FIREBASE_AUTH = "lhXbLJ2XS0nphlMsyEINh2pTRDrIWeimzATT4Zap";
const String FIREBASE_PATH = "/";
const int SSL_PORT = 443;

char apn[] = "internet.claro.com.ec"; //airtel -> "airtelgprs.com"
char user[] = "afluna@espol.edu.ec";
char pass[] = "ak8ka002";

double latitud; //valor para la latitud
double longitud; //valor para la longitud

TinyGsm modem(sim800);

TinyGsmClientSecure gsm_client_secure_modem(modem, 0);
HttpClient http_client = HttpClient(gsm_client_secure_modem, FIREBASE_HOST, SSL_PORT);

unsigned long previousMillis = 0;
```

*Ilustración 1. Define librerías y parámetros necesarios para la conexión.*

```
void setup()
{
    pinMode(pulsadorPin1, INPUT);
    Serial.begin(9600);
    gps.begin(9600);

    sim800.begin(9600);
    Serial.println("SIM800L serial initialize");

    Serial.println("Initializing modem...");
    modem.restart();
    String modemInfo = modem.getModemInfo();
    Serial.print("Modem: ");
    Serial.println(modemInfo);

    bool connected = false
    while (!connected) {
        if ((gsmAccess.begin(PINNUMBER) == GSM_READY) &&
            (gprs.attachGPRS(GPRS_APN, GPRS_LOGIN, GPRS_PASSWORD) == GPRS_READY)) {
            connected = true;
        } else {
            Serial.println("Not connected");
            delay(1000);
        }
    }

    location.begin();

    http_client.setHttpResponseTimeout(10 * 1000); //^0 secs timeout
    Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH); // Se realiza la conexión con la base de datos enviando los datos del Host y la Key
}
```

*Ilustración 2. Inicializa gps y conecta con base de datos.*



```

void loop()
{
    if(gps.available())
    {
        dato=gps.read();
        Serial.print(dato);
    }
    http_client.connect(FIREBASE_HOST, SSL_PORT);

    while (true) {
        if (!http_client.connected())
        {
            Serial.println();
            http_client.stop();// Shutdown
            Serial.println("HTTP not connect");
            break;
        }

    }

    latitud = location.latitude();
    longitud = location.longitude();
    valorPulsador1 = digitalRead(pulsadorPin1);
    if (valorPulsador1 == HIGH) {
        Serial.print("Se envio la informacion correctamente");
        Firebase.set("/UsersRegis/Usuarios/aemorah/localizacion/latitud",latitud);
        Firebase.set("/UsersRegis/Usuarios/aemorah/localizacion/longitud",longitud);
    }
    valorPulsador2 = digitalRead(pulsadorPin2);
    if (valorPulsador2 == HIGH) {
        Serial.print("Babahoyo,Latitud:  -1.830809,Longitud:  -79.546698");
        Serial.print("Se envio la informacion correctamente");
        Firebase.set("/UsersRegis/Usuarios/aemorah/localizacion/latitud",-1.830809);
        Firebase.set("/UsersRegis/Usuarios/aemorah/localizacion/longitud",-79.546698);
    }
}

```

*Ilustración 3. Se lee el estado de la información del gps y se envía a firebase*

## 3.2. ANDROID STUDIO

```
//Metodo Para Borrar Parada
public void borrar(View view){
    if (!paradaSel.equals("")){
        AlertDialog.Builder builder = new AlertDialog.Builder(context: this);
        builder.setTitle("Transporte_Alimentos_AMST_61");
        builder.setIcon(R.drawable.edit_aviso);
        builder.setMessage("¿Quiere borrar la Parada "+ paradaSel+"?");
        builder.setPositiveButton(text: "Si", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id) {
                databaseReference.child("UsersRegis").child("Usuarios").child(usuarioActual).child("Paradas").child(paradaSel).removeValue();
                paradaSel="";
            }
        });
        builder.setNegativeButton(text: "No", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id) { dialog.dismiss(); }
        });
        AlertDialog alert = builder.create();
        alert.show();
    }else{
        Toast.makeText(context: this, text: "Seleccione un Dispositivo que Desea Borrar", Toast.LENGTH_SHORT).show();
    }
}
```

Ilustración 4. Se observa el método "Borrar Parada", el mismo que permite al usuario "Administrador" borrar una parada asignada a un determinado conductor.

```
//Metodo Para Guardar Informacion
public void guardado1(View view){
    latit= eLatitud.getText().toString().toLowerCase().trim();
    double dlat = Double.parseDouble(latit);
    longi=eLongitud.getText().toString().trim();
    double dlon = Double.parseDouble(longi);
    String infoSel = spinner.getSelectedItem().toString();
    System.out.println(infoSel);
    databaseReference.child("UsersRegis").child("Usuarios").child(usuarioActual).addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            System.out.println(snapshot);
            String nombre = snapshot.child("Conductores").child(infoSel).child("nombre").getValue(String.class);
            long numParadas = (long) snapshot.child("Conductores").child(infoSel).child("paradas").getValue();
            String titulo = infoSel + Long.toString(numParadas);

            Map<String, Object> taskMap = new HashMap<>();
            taskMap.put(k: "Conductor", nombre);
            taskMap.put(k: "Latitud", dlat);
            taskMap.put(k: "Longitud", dlon);
            databaseReference.child("UsersRegis").child("Usuarios").child(usuarioActual).child("Paradas").child(titulo).setValue(taskMap);
            databaseReference.child("UsersRegis").child("Usuarios").child(usuarioActual).child("Conductores").child(infoSel).child("paradas").setValue(numParadas+1);
            eLatitud.setText("");
            eLongitud.setText("");
        }

        @Override
        public void onCancelled(@NonNull DatabaseError error) {
        }
    });
}
```

Ilustración 5. Método "Guardar Información", este método permite al usuario "Administrador" agregar una nueva parada a un determinado conductor

```

//metodo de inicio de sesion normalmente
public void iniciarSesion(View view) {
    boolean resultadoValid = validarIngresos();
    if (resultadoValid) {
        databaseReference.child("UsersRegis").child("Usuarios").addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                if (snapshot.hasChild(datoUser) ) {
                    System.out.println("Correcto");
                    usuario user = snapshot.child(datoUser).child("info").getValue(usuario.class);
                    String userContra = user.getClave();
                    String userClass = user.getClase();
                    if (userContra.equals(datoPass)) {
                        usuario.setText("");
                        contra.setText("");
                        ClaseUsando.usuarioUsando = user;
                        if (userClass.equals("Administrador")){
                            Intent i = new Intent( packageContext: IngresoSesion.this, MenuPrincipalAdmin.class);
                            startActivity(i);
                        }else if(userClass.equals("Conductor")){
                            Intent i = new Intent( packageContext: IngresoSesion.this, MenuPrincipalConductor.class);
                            startActivity(i);
                        }
                    }else{
                        Toast.makeText( context: IngresoSesion.this, text: "Contraseña incorrecta",
                                Toast.LENGTH_SHORT).show();
                    }
                }else{
                    Toast.makeText( context: IngresoSesion.this, text: "No existe dicho Usuario",
                                Toast.LENGTH_SHORT).show();
                }
            }
        })
    }

    @Override
    public void onCancelled(@NonNull DatabaseError error) {

```

Ilustración 6. Método de "Inicio de Sesión", este método valida las credenciales tanto de un usuario "Administrador" como de un usuario "Conductor"

```

@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;
    databaseReference.child("UsersRegis").child("Usuarios").child(Usuario_persona).child("Paradas").addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            for (DataSnapshot objSnapshot : snapshot.getChildren()) {
                double latit = (double) objSnapshot.child("Latitud").getValue();
                double longi = (double) objSnapshot.child("Longitud").getValue();
                String nombre = objSnapshot.child("Conductor").getValue(String.class);
                // Add a marker in Sydney and move the camera
                LatLng sydney = new LatLng(latit, longi);
                mMap.addMarker(new MarkerOptions().position(sydney).title("Parada de "+nombre));
                mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney));
            }
        }

        @Override
        public void onCancelled(@NonNull DatabaseError error) {
        }
    });
}

```

Ilustración 7. Método para obtener las coordenadas a través del módulo GPS para poder ubicar el camión en el mapa

```

//método para validar el ingreso del usuario es unico y no se repite en la base de datos
private void validarUsuarioIngreso() {
    databaseReference.child("UsersRegis").addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            usuario userIngresado = new usuario();
            userIngresado.setUsuario(datoUser);
            userIngresado.setNombre(datoNombre);
            userIngresado.setApellidos(datoApellido);
            userIngresado.setCorreo(datoCorreo);
            userIngresado.setClave(datoContra);
            String infoSel = spinner.getSelectedItem().toString();
            userIngresado.setClase(infoSel);

            if (!snapshot.hasChild(userIngresado.getUsuario())) {
                databaseReference.child("UsersRegis").child("Usuarios").child(userIngresado.getUsuario()).child("info").setValue(userIngresado);
                //databaseReference.child("UsersRegis").child(per.getUsuario()).setValue(0);
                Toast.makeText(context, Registro.this, text "Agregado", Toast.LENGTH_SHORT).show();
                if (infoSel.equals("Conductor")){
                    databaseReference.child("UsersRegis").child("Usuarios").child(userIngresado.getUsuario()).child("localizacion").child("Longitud").setValue(-1.1);
                    databaseReference.child("UsersRegis").child("Usuarios").child(userIngresado.getUsuario()).child("localizacion").child("Latitud").setValue(-1.1);
                }
                vaciarCeldas();
            } else {
                user.setError("Se usuario ya en uso");
            }
        }

        @Override
        public void onCancelled(@NonNull DatabaseError error) {
        }
    });
}

```

Ilustración 8. Método para el registro de un nuevo usuario, este método además valida que el usuario ingresado ya no exista en la base de datos, caso contrario mostrará un mensaje de "Usuario ya en uso"

```

private void obtenerStatus(){
    if (validarIngresos()){
        databaseReference.child("UsersRegis").child("Usuarios").addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                if (snapshot.hasChild(datoUser) ) {
                    databaseReference.child("UsersRegis").child("Usuarios").child(datoUser).child("Conductores").child(usuario_persona).child("nombre").setValue(nombre_persona);
                    databaseReference.child("UsersRegis").child("Usuarios").child(datoUser).child("Conductores").child(usuario_persona).child("placa").setValue(datoPlaca);
                    databaseReference.child("UsersRegis").child("Usuarios").child(datoUser).child("Conductores").child(usuario_persona).child("paradas").setValue(0);
                    databaseReference.child("UsersRegis").child("Usuarios").child(usuario_persona).child("Administrador").setValue(datoUser);
                }
            }
            @Override
            public void onCancelled(@NonNull DatabaseError error) {
            }
        });
    } else{
        Toast.makeText( context: RegistroAdministrador.this, text: "Rellene los datos correctamente",
            Toast.LENGTH_SHORT).show();
    }
}
}

```

Ilustración 9. Método para poder asociar un usuario "Administrador" con un determinado conductor

```

//metodo donde se obtiene la informacion y se la agrega al listView
private void obtenerInfo() {
    databaseReference.child("UsersRegis").child("Usuarios").child(usuarioActual).child("Paradas").addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            final DecimalFormat df = new DecimalFormat( pattern: "0.00000");
            listaTit.clear();
            listaImg.clear();
            listaLat.clear();
            listaLong.clear();
            //arrayList = new ArrayList<>();
            for (DataSnapshot objSnapshot : snapshot.getChildren()){
                String etiqueta= objSnapshot.getKey();
                double latitud = (double) objSnapshot.child("Latitud").getValue();
                double longitud = (double) objSnapshot.child("Longitud").getValue();

                listaTit.add(etiqueta);
                listaLat.add(Double.toString(Double.parseDouble(df.format(latitud))));
                listaLong.add(Double.toString(Double.parseDouble(df.format(longitud))));
                listaImg.add(R.drawable.ic_locat);

                arrayTit=new String[listaTit.size()];
                arrayTit = listaTit.toArray(arrayTit);
                arrayimg=new Integer[listaImg.size()];
                arrayimg = listaImg.toArray(arrayimg);
                arrayLat=new String[listaLat.size()];
                arrayLat = listaLat.toArray(arrayLat);
                arrayLong=new String[listaLong.size()];
                arrayLong = listaLong.toArray(arrayLong);
                MostrarParadas.MyAdapter adapter =new MostrarParadas.MyAdapter( @ MostrarParadas.this,arrayTit,arrayimg,arrayLat,arrayLong);
                listaParadas.setAdapter(adapter);
            }
        }

        @Override
        public void onCancelled(@NonNull DatabaseError error) {
        }
    });
}
}

```

Ilustración 10. Método que obtiene los datos de las paradas de la base de datos y los muestra en la opción "Observar Paradas" del usuario "Administrador"



## 4. RESULTADOS

### 4.1. SCREENSHOT DE LA APP

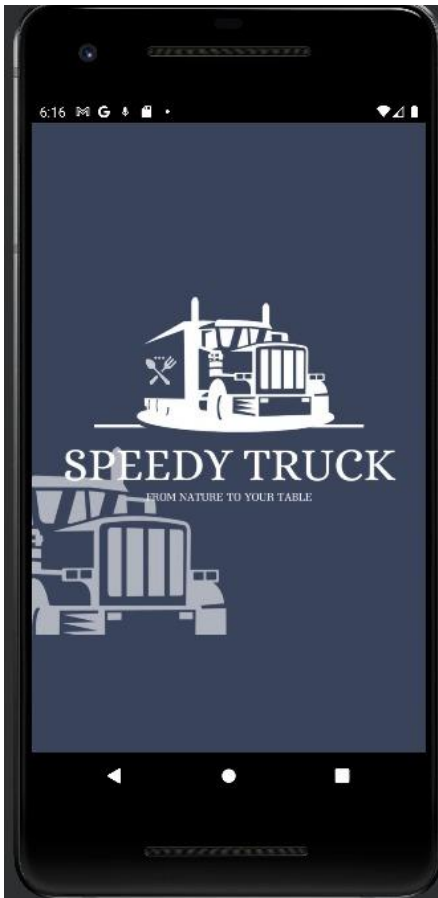


Ilustración 12. Pantalla de Splash de la aplicación



Ilustración 13. Pantalla de inicio de Sesión con Google



Ilustración 11. Pantalla de registro de "administrador" o "conductor"

## 4.2. SCREENSHOT DE LA APP – USUARIO

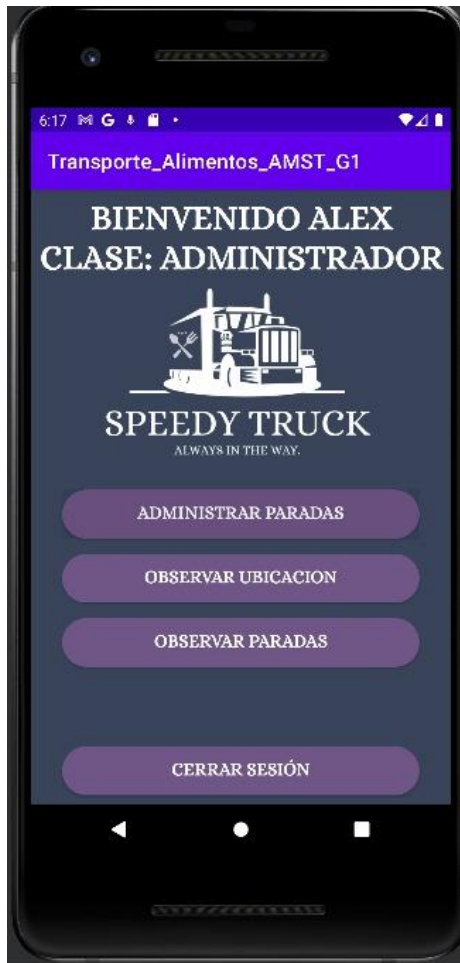


Ilustración 16. Menú de inicio de usuario "Administrador"



Ilustración 15. Opción "Administrar Paradas"



Ilustración 14. Opción para agregarle una parada a un conductor



Ilustración 18. Opción para borrar una parada



Ilustración 17. Opción "Observar Paradas"





Ilustración 19. Opción "Observar ubicación"



Ilustración 20. Opción "Observar ubicación"

#### 4.3. SCREENSHOT DE LA APP – CONDUCTOR



Ilustración 21. Menú de inicio para el usuario "Conductor"

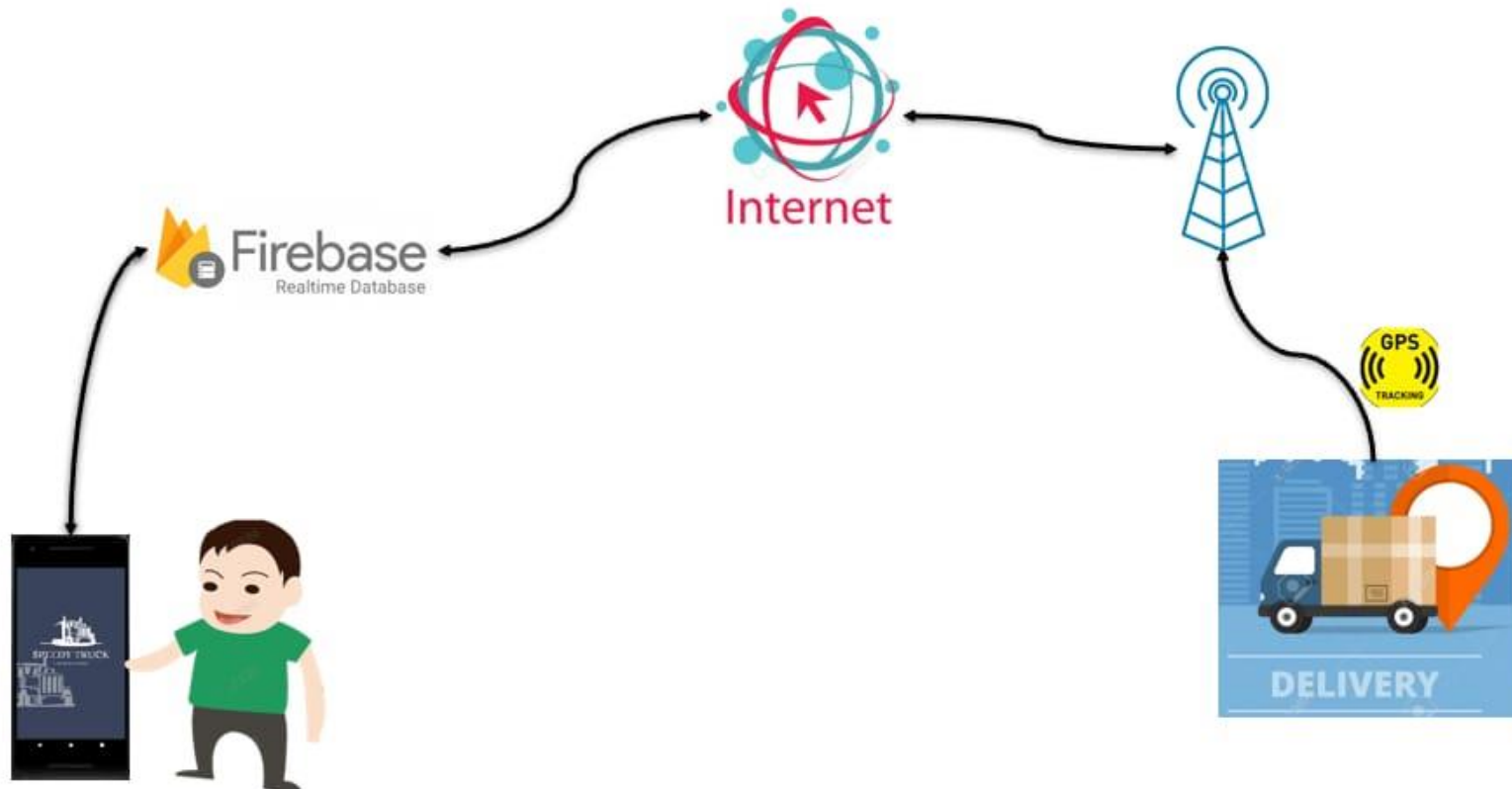


Ilustración 22. Opción "Registrar Administrador"

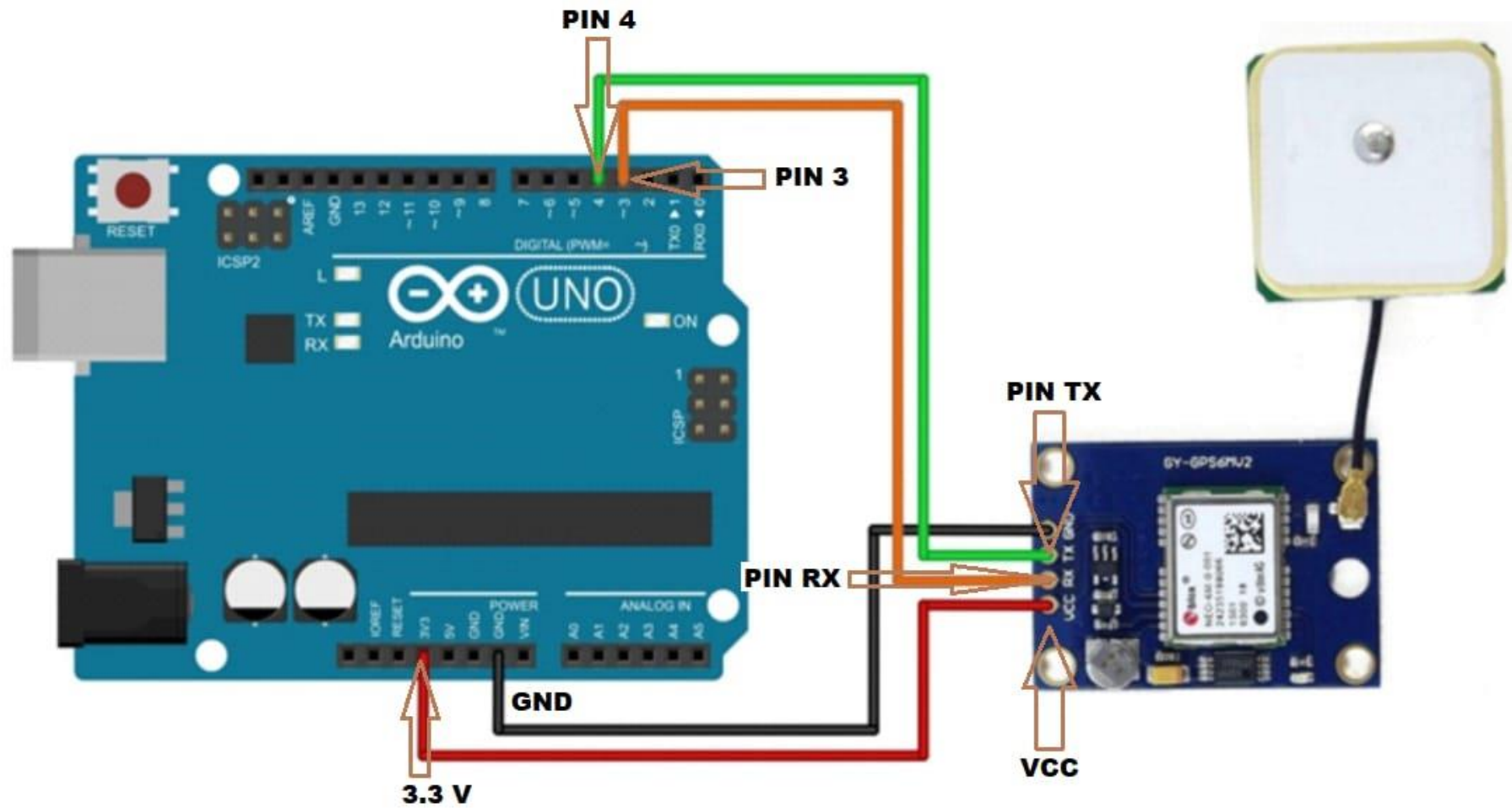
## 5. DIAGRAMAS

### 5.1. DIAGRAMA DEL PROYECTO

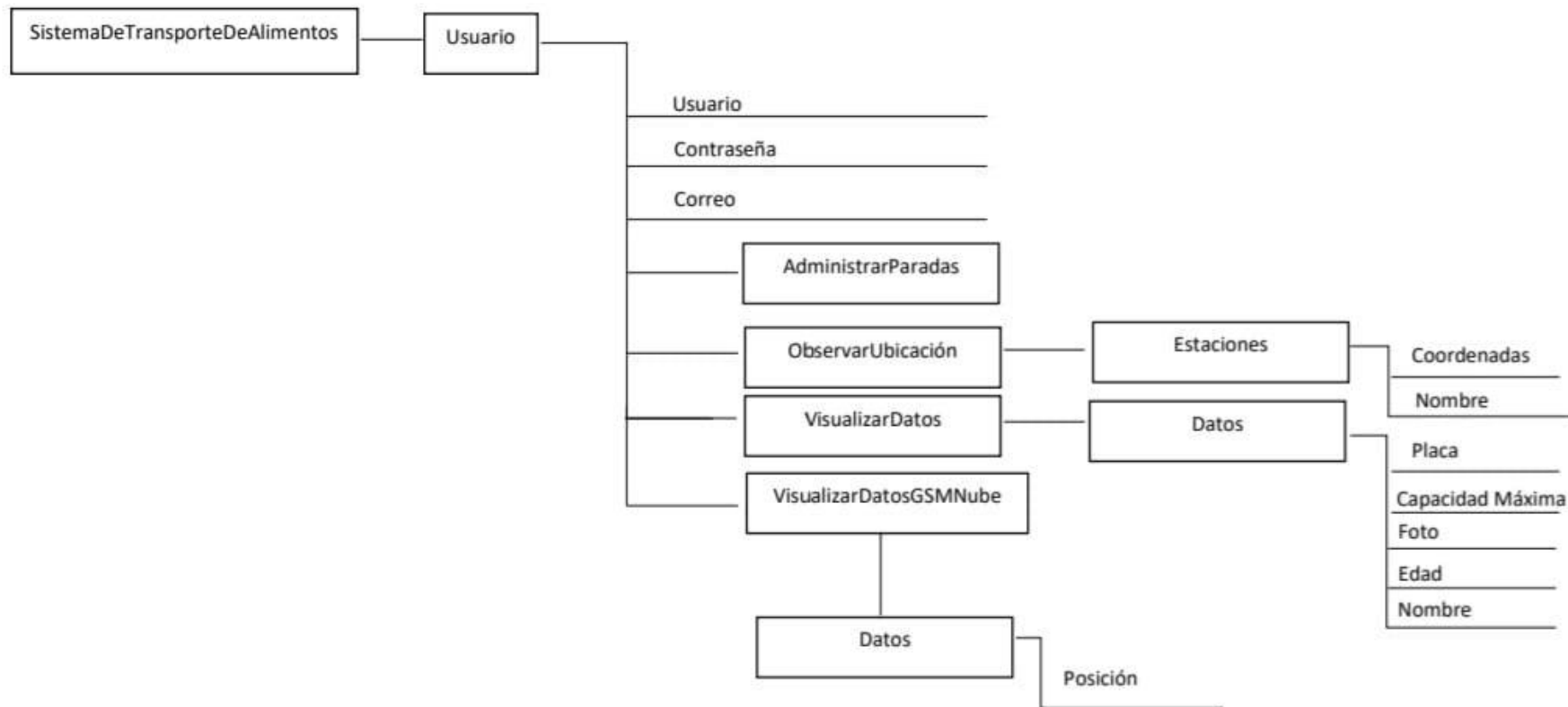
# Diagrama del proyecto



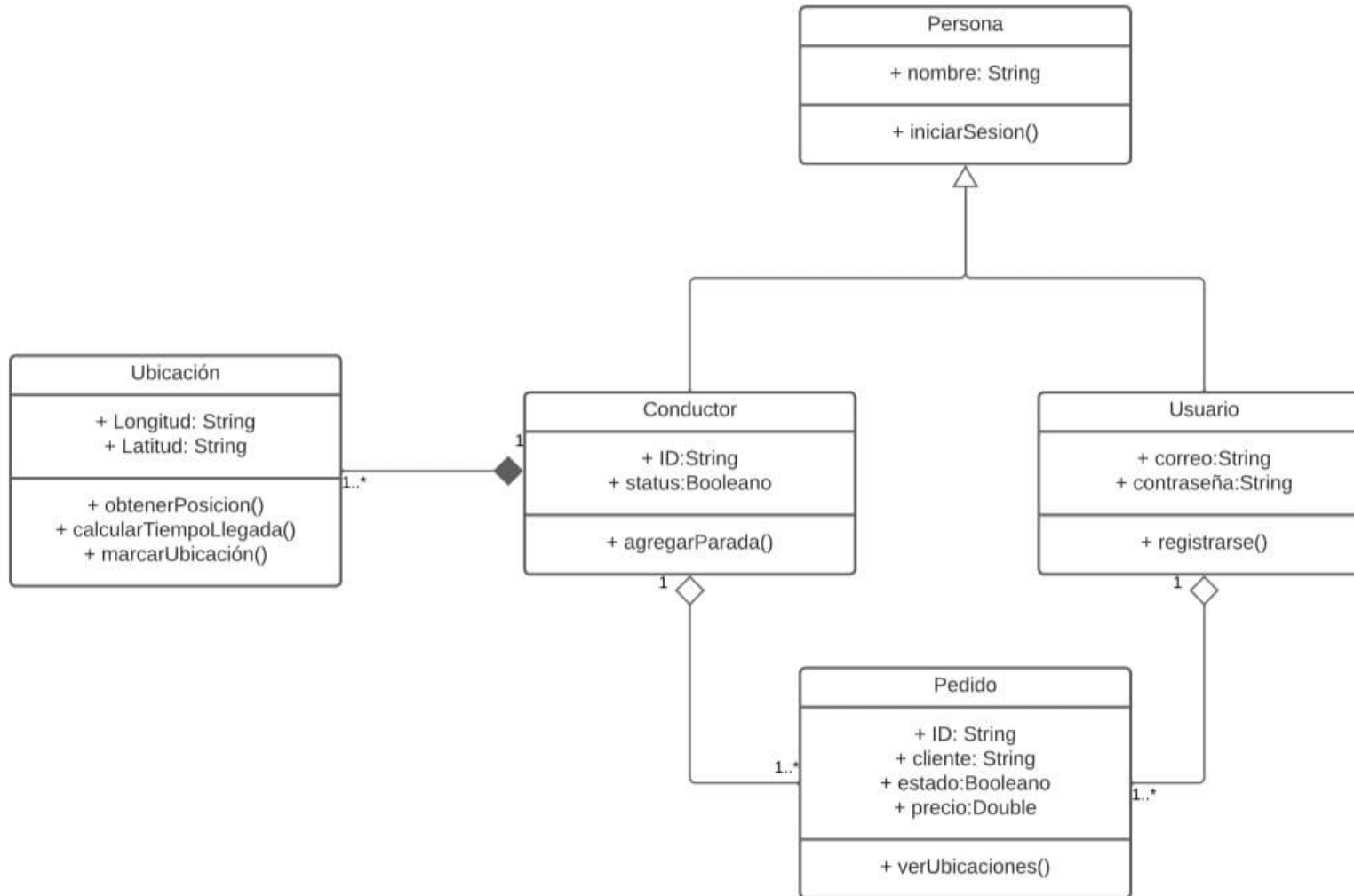
## 5.2. DIAGRAMA DE CIRCUITO



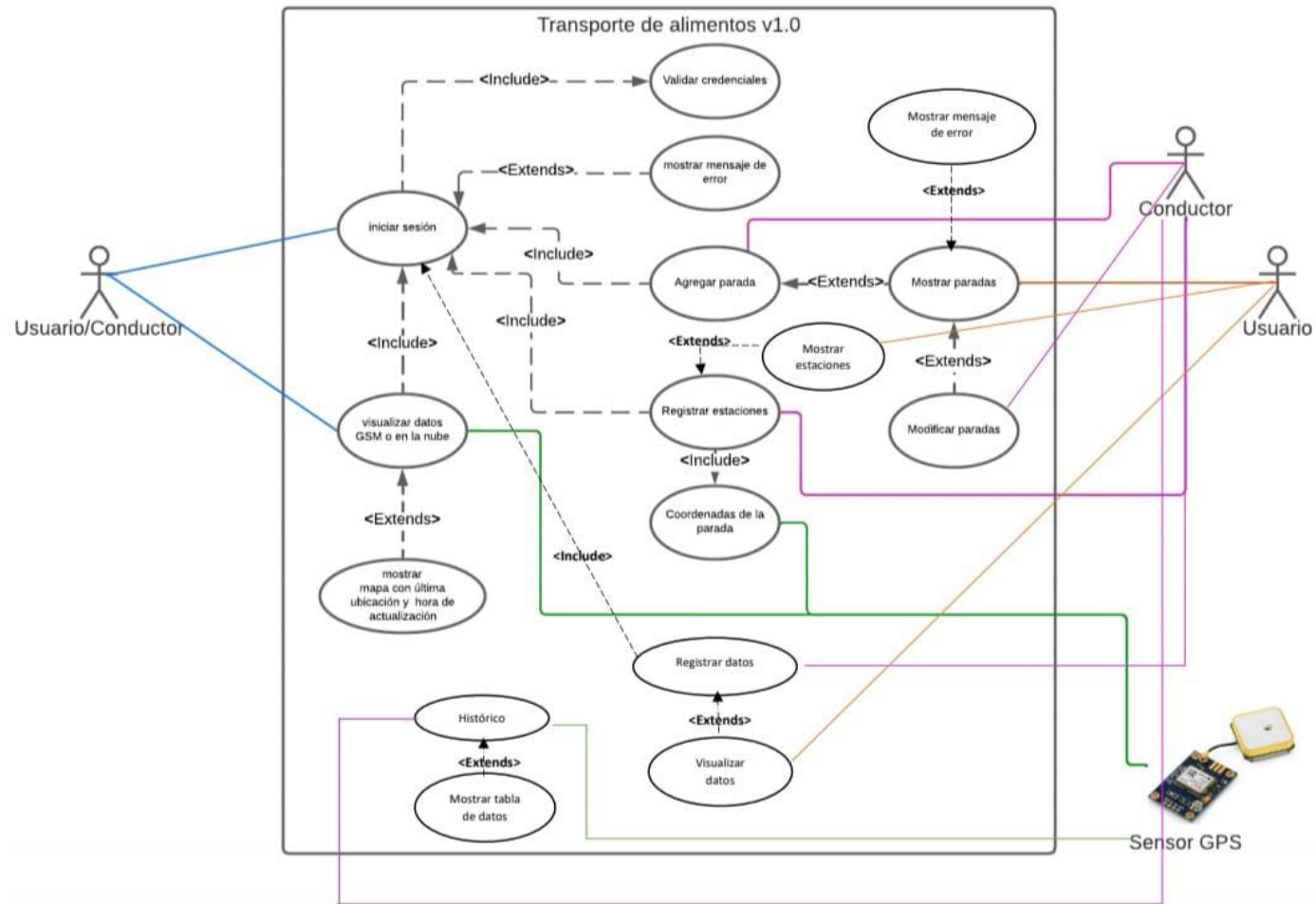
### 5.3. DIAGRAMA DE ÁRBOL



## 5.4. DIAGRAMA DE CLASES

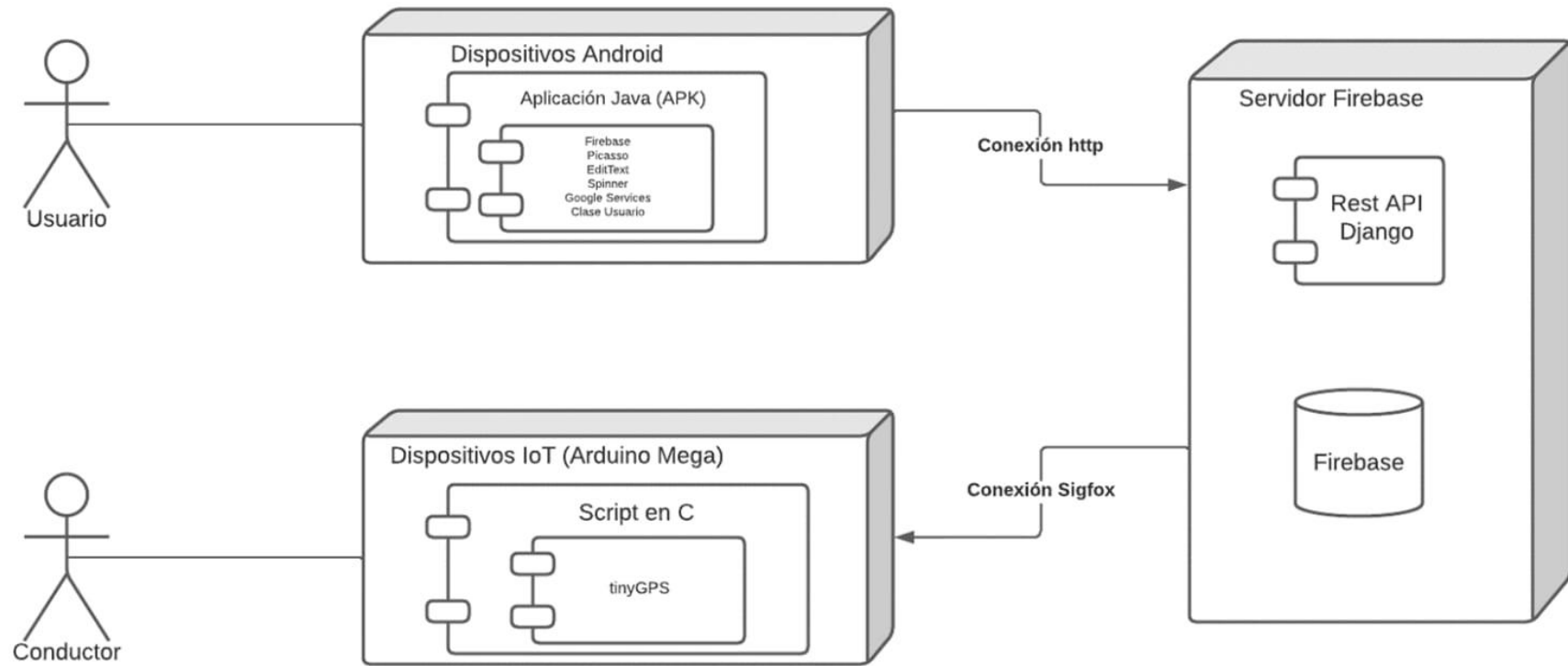


## 5.5. DIAGRAMA DE CASOS DE USO





## 5.6. DIAGRAMA DE DESPLIEGUE





## 6. BIBLIOGRAFÍA

- Android Developer. (2022). Obtenido de Introducción a Android Studio:  
<https://developer.android.com/studio/intro?hl=es-419>
- Aprendiendo Arduino. (2020). *IDE Arduino*. Obtenido de  
<https://aprendiendoarduino.wordpress.com/2016/12/11/ide-arduino/>
- Descubre Arduino. (2019). *Arduino Uno, partes, componentes, para qué sirve y donde comprar*.  
Obtenido de <https://descubrearduino.com/arduino-uno/>
- Descubre Arduino. (2020). Obtenido de Módulo GPS e Interfaz con Arduino UNO:  
<https://descubrearduino.com/modulo-gps/>
- Pormetek. (2018). Obtenido de MÓDULO GSM/GPRS: llamar y enviar SMS:  
<https://www.prometec.net/gprs-llamar-enviar-sms/>
- Store Arduino. (2020). Obtenido de Arduino Uno Rev3 :  
<https://store.arduino.cc/products/arduino-uno-rev3>
- Xataka Basics. (2019). *Qué es Arduino, cómo funciona y qué puedes hacer con uno* . Obtenido  
de <https://www.xataka.com/basics/que-arduino-como-functiona-que-puedes-hacer-uno>

