

**ESCUELA SUPERIOR POLITECNICA DEL LITORAL**

**Materia**

Aplicaciones Móviles y Servicios Telemáticos

**Tema:**

Clasificador de objetos mediante peso.

“Manual Tecnico”

Grupo 4

**Profesora:**

Ing. Adriana Collaguazo

**Integrantes:**

* Alfredo Valenzuela
* Jorge García
* Jose Delgado

**I Término - 2020**

MANUAL TECNICO

1. **Objetivo del proyecto**

El clasificador de objetos tiene como objetivo darle al usuario una mayor eficiencia a la hora de clasificar los productos con un porcentaje de error mínimo.

Este prototipo de solución se diseñó a través de una aplicación móvil para dispositivos Android, que a través de hardware que procesa y censa el peso de un artículo para compararlo con una base de datos, lo discriminara a través de un algoritmo para seleccionar el producto adecuado y mostrárselo al usuario.

Además, la aplicación móvil ofrecerá una vista donde el usuario podrá entrenar al modelo para que la hora de elegir un artículo, lo realice con una mayor precisión.

El usuario podrá ingresar al sistema con un usuario registrado o a través de su cuenta de Google y cerrar sesión de la misma forma.

1. **Explicación de los recursos de Hardware**

|  |  |  |
| --- | --- | --- |
| Instrumento | Descripcion | Presupuesto(Costo) |
|  | **Thinkxtra**  Dispositivo que a través de redes Lpwan, interconecta el Hardware con el Software a través de mensajes de callbacks. | $ 143.87 |
|  | **Batería.( 9 Voltios)**  Suministro de energía para el funcionamiento del prototipo de manera inalámbrica. | $ 1.5 |
|  | **Celda de carga (20kg)**  Convertidor de fuerza a una señal de voltaje que pueda procesar el microcontrolador | $ 3.00 |
|  | **Modulo HX711**  Modulo que traduce de manera sencilla la señal de voltaje que emite la celda de carga, convirtiendo la lectura analógica en digital. | $ 2.00 |
|  | R**esistencia**  Componente que opone resistencia entre dos puntos, Se uso para medir el porcentaje de batería a traes de un circuito divisor de voltaje. | $0.20 |
|  | J**umpers**  Cables de interconeciones entre los elementos que conforman el hadware. | $1.00 |
|  | **Interruptor**  Componente para encender o apagar el prototipo. | $ 0.45 |
|  | **Arduino Uno**  Microcontrolador programable, procesa la información a través de sus pines. | $ 12.00 |
| Total | ------------------------------- | $ 164.02 Dólares |

1. **Recursos de sofware**

|  |  |
| --- | --- |
| Software | Descripción |
| Firebase  Versión: 7.15.12 | Firebase es la base de datos no relacional donde se almaceno los datos del sistema, artículos y en donde llegara la información envida por el Thinextra. |
| GitHub  GitHub Logos and Usage · GitHub | GitHub es una Plataforma para el control y manejo de versiones del proyecto con propósito colaborativo. |
| Arduino  Versión: 1.8.13 | El IDE de Arduino es un software donde se implementó la programación que enviara los datos de Peso y batería a través de una rutina. |
| Android Studio  Versión: 4.0 | Entorno de desarrollo donde se llevará la aplicación móvil para el sistema operativo Android, se llevó a cabo el diseño de la interfaz y lógica de la misma. |

1. **Diagrama del circuito( Incluido el diagrama esquemático )**

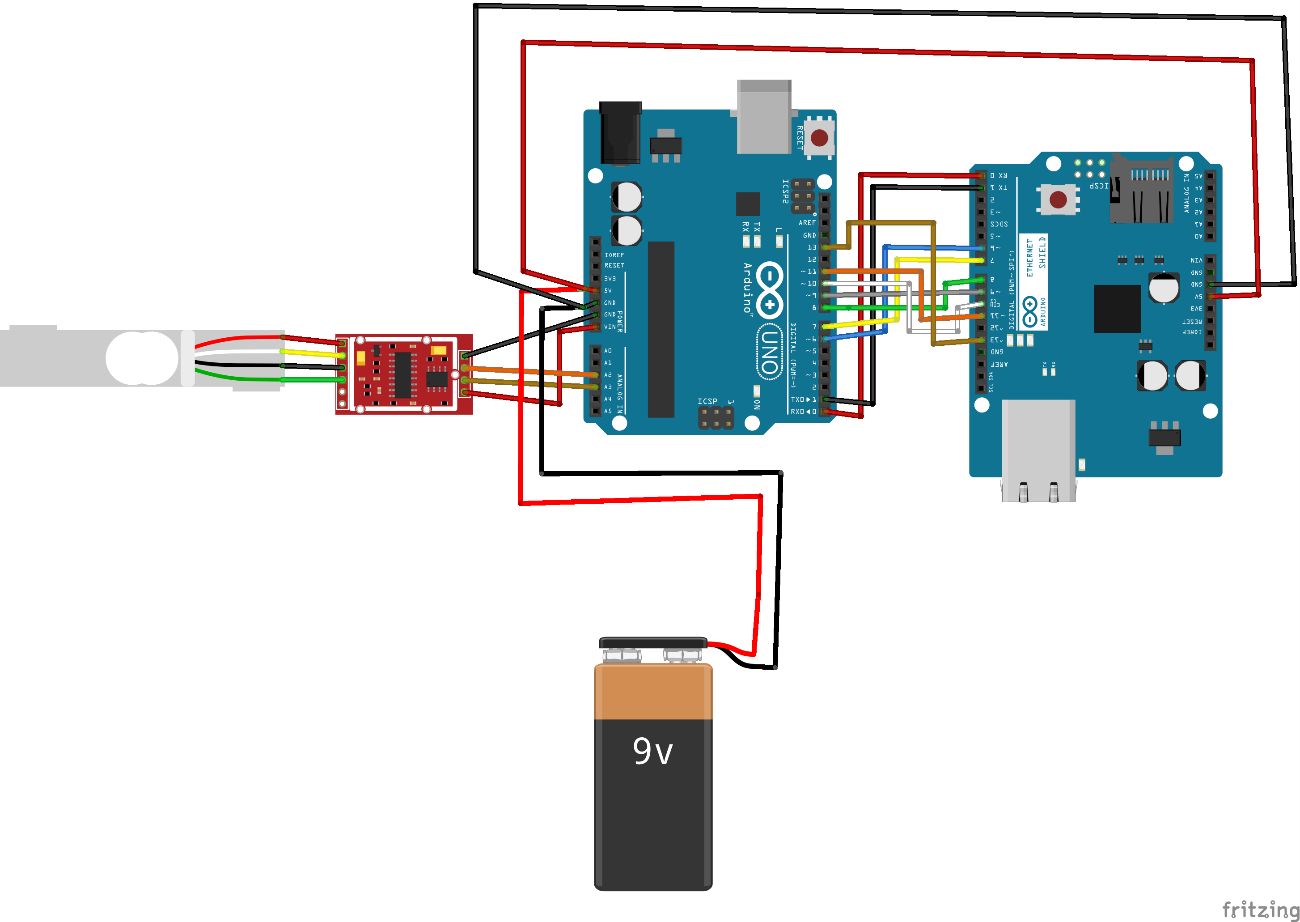
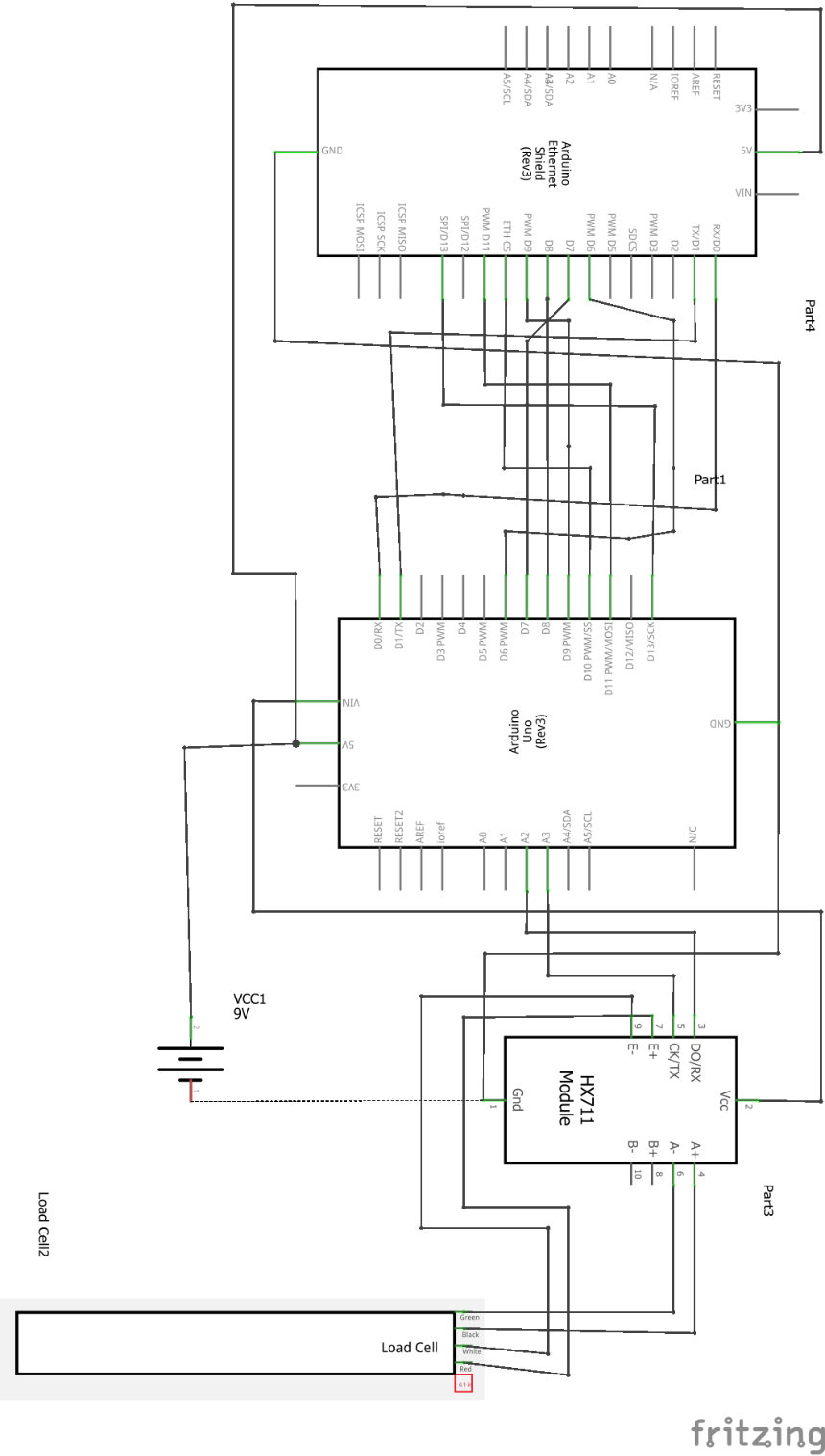
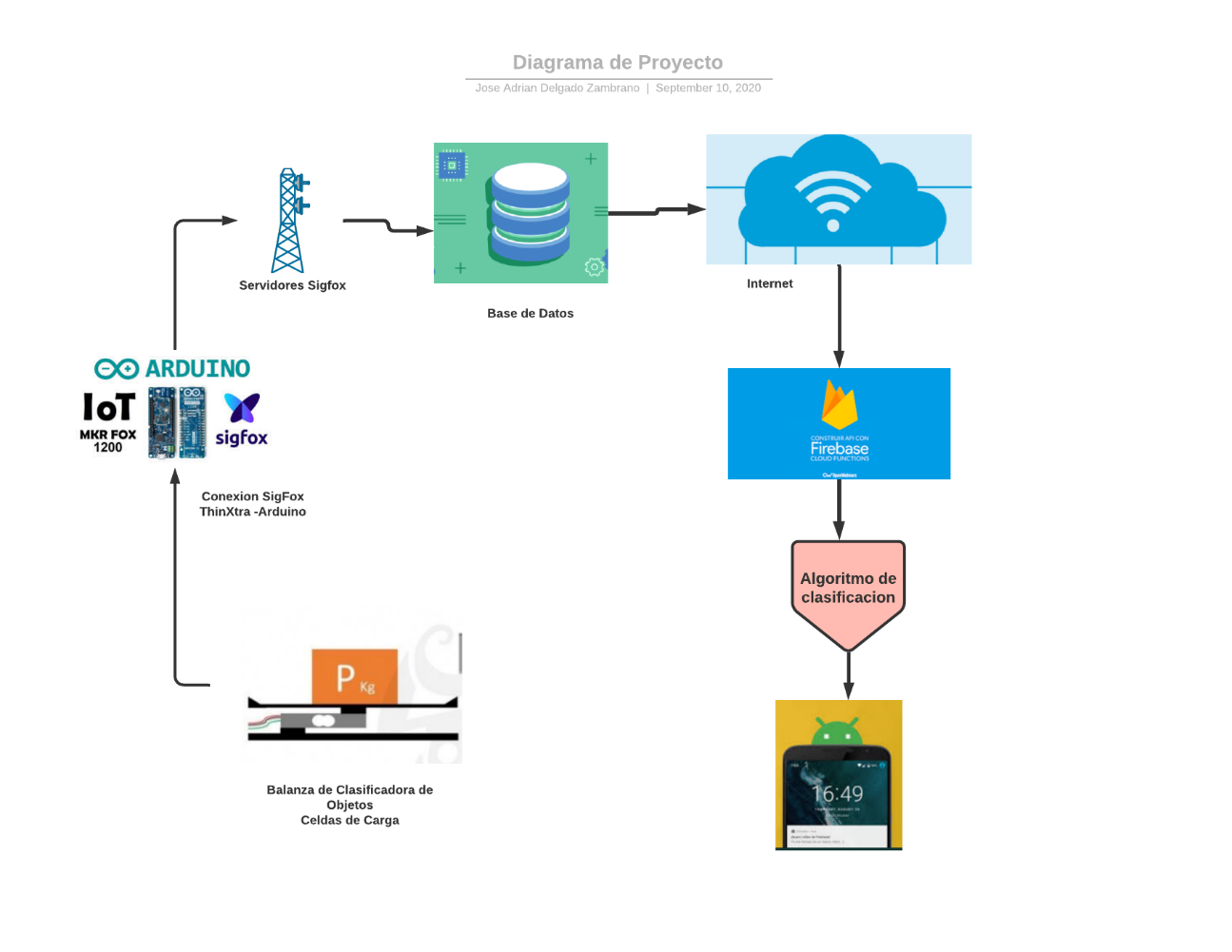


Ilustración 1 Diagrama del circuito desarrollado en Fritzing

Ilustración 2 Diagrama esquemático del circuit

1. **Diagrama de diseño del proyecto**



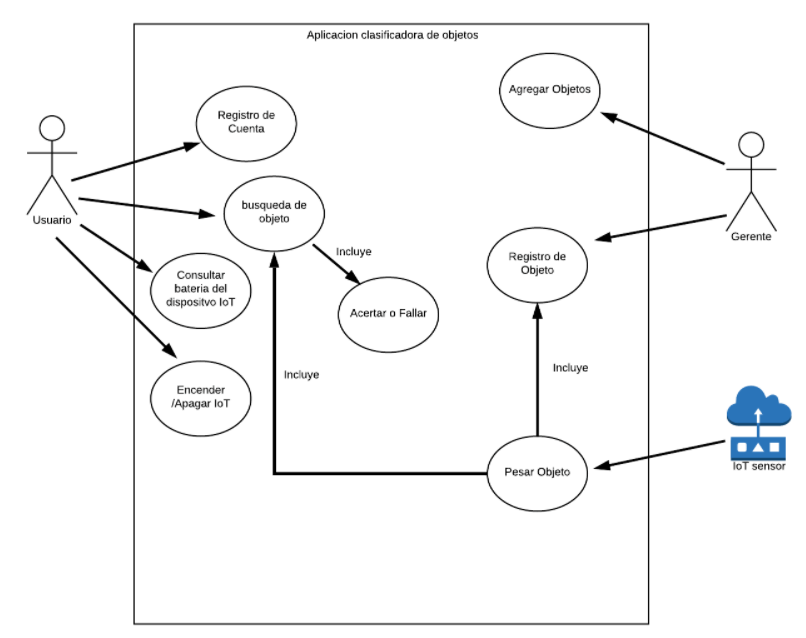
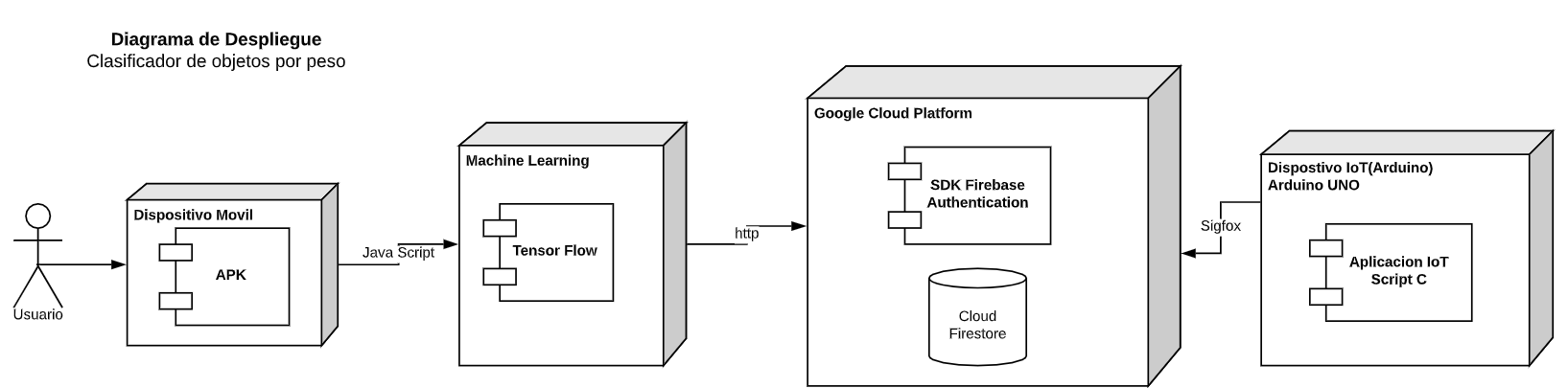
1. **Diagrama de casos de uso**
2. 
3. **Diagrama de despliegue del sistema**
4. 

Ilustración 3 Diagrama de despliegue

1. **Diagrama de árbol de la base de datos**



Ilustración 4 Diagrama de Arbol Firebase Parte 1

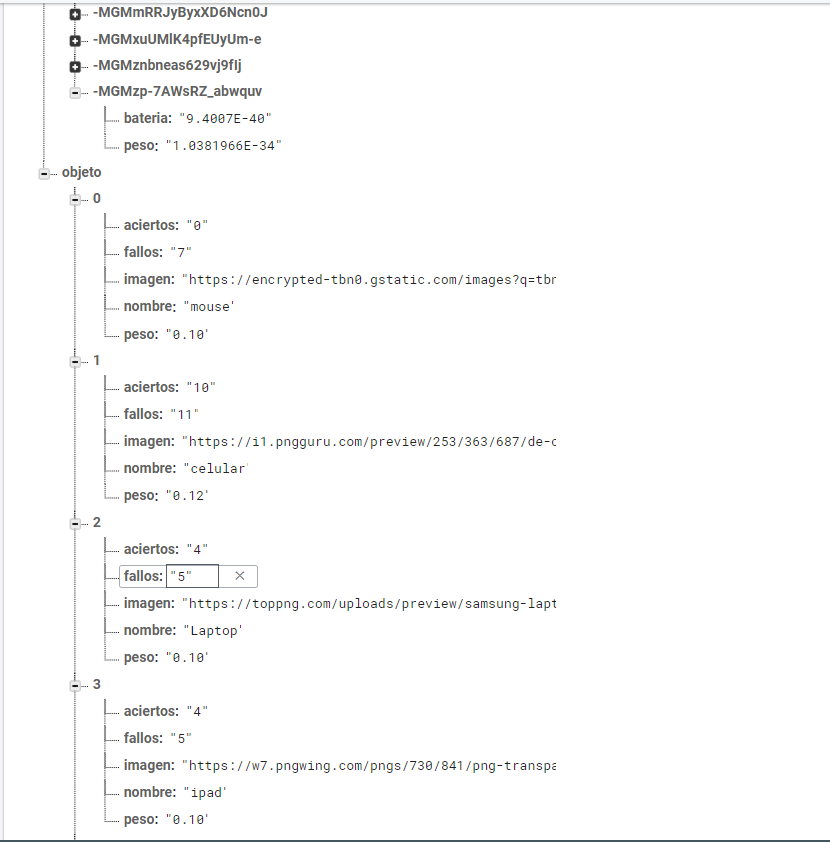


Ilustración 5 Diagrama de Arbol Firebase Parte 2

1. Diagrama de clase

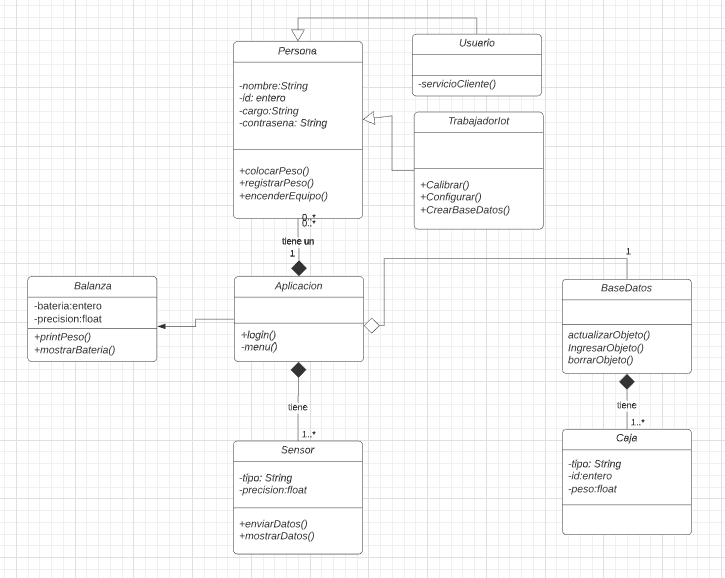


Ilustración 6 Diagrama de clases

**Importancia del Proyecto**

Como se explico anteriormente uno de los paso mas importante de la realización del proyecto fue el modelo de clasificación que se iba a seguir para poder escoger el articulo mas parecido, para esto se realización un modelo de aciertos y fallos cada ves el usuario deslice entre los artículos con un mismo peso, se tomara como referencia que el articulo que se paso se le aumentar un fallo mientras que si selecciona uno se le aumentara un acierto.

**Bibliografía**

https://app.lucidchart.com/invitations/accept/87840040-964c-4a0f-b3ae-32978e49fe5c

<https://es.wikipedia.org/wiki/Sigfox>

<https://naylampmechatronics.com/blog/25_tutorial-trasmisor-de-celda-de-carga-hx711-ba.html>

<https://es.wikipedia.org/wiki/Resistor>

Código de la aplicación

Se comienza mostrando el mainActivity donde se realizara la solicitud para iniciar sesión a través de Google.

**public class** MainActivity **extends** AppCompatActivity {  
 *//Variables para nuestro login* **private** EditText **nombre**,**pasdword**;  
  
 **private static final** String ***TAG*** = **"GoogleActivity"**;  
 **private static final int *RC\_SIGN\_IN*** = 9001;  
  
 **private** FirebaseAuth **mAuth**;  
  
 **private** GoogleSignInClient **mGoogleSignInClient**;  
  
 **private** Button **btnGoogleLogin**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 *//Hilo usado para mostrar el splahs activity, luego de esto  
 //no es recomendable dormir al hilo en el mainactivity* **try** {  
 Thread.*sleep*(500);  
 } **catch** (InterruptedException e) {  
 e.printStackTrace();  
 }  
 setTheme(R.style.***AppTheme***);  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
  
 *//isntaciamos nuestros edistext de sesion* **nombre**=(EditText)findViewById(R.id.***etUsuario***);  
 **pasdword**=(EditText)findViewById(R.id.***edContrasena***);  
  
  
 GoogleSignInOptions gso = **new** GoogleSignInOptions.Builder(GoogleSignInOptions.***DEFAULT\_SIGN\_IN***)  
 .requestIdToken(**"532856539478-8v7llo0hoopoc26gvpq0kgr31tem2i6q.apps.googleusercontent.com"**)  
 .requestEmail()  
 .build();  
  
 **btnGoogleLogin** = findViewById(R.id.***btnLoginGoogle***);  
  
 **btnGoogleLogin**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 signIn();  
 }  
 });  
  
 **mGoogleSignInClient** = GoogleSignIn.*getClient*(**this**, gso);  
 **mAuth** = FirebaseAuth.*getInstance*();  
 *//to remove "information bar" above the action bar* getWindow().setFlags(WindowManager.LayoutParams.***FLAG\_FULLSCREEN***,  
 WindowManager.LayoutParams.***FLAG\_FULLSCREEN***);  
 *//to remove the action bar (title bar)* getSupportActionBar().hide();  
 }

**public void** Acceder(View view){  
 HashMap<String,String> datos= **new** HashMap<>();  
 datos.put(**"jadriandelgado17@gmail.com"**,**"Hola123"**);  
  
 String nombreV=**nombre**.getText().toString();  
 String password=**pasdword**.getText().toString();  
  
 **if**(**nombre**.length()==0){  
 Toast.*makeText*(**this**,**"Debe Ingresar un usuaio valido"**, Toast.***LENGTH\_LONG***).show();  
 }  
  
 *//if (password != datos.get("jadriandelgado17@gmail.com")){* **if** (password.length() == 0 ){  
 Toast.*makeText*(**this**,**"Contraseña Incorrecta"**,Toast.***LENGTH\_LONG***).show();  
 } **else** {  
 Toast.*makeText*(**this**,**"Contraseña correcta"**,Toast.***LENGTH\_LONG***).show();  
 }  
}  
  
**private void** signIn() {  
 Intent signInIntent = **mGoogleSignInClient**.getSignInIntent();  
 startActivityForResult(signInIntent, ***RC\_SIGN\_IN***);  
}  
  
@Override  
**public void** onActivityResult(**int** requestCode, **int** resultCode, Intent data) {  
 **super**.onActivityResult(requestCode, resultCode, data);  
  
 *// Result returned from launching the Intent from GoogleSignInApi.getSignInIntent(...);* **if** (requestCode == ***RC\_SIGN\_IN***) {  
 Task<GoogleSignInAccount> task = GoogleSignIn.*getSignedInAccountFromIntent*(data);  
 **try** {  
 *// Google Sign In was successful, authenticate with Firebase* GoogleSignInAccount account = task.getResult(ApiException.**class**);  
 Log.*d*(***TAG***, **"firebaseAuthWithGoogle:"** + account.getId());  
 firebaseAuthWithGoogle(account.getIdToken());  
 } **catch** (ApiException e) {  
 *// Google Sign In failed, update UI appropriately* Log.*w*(***TAG***, **"Google sign in failed"**, e);  
 *// ...* }  
 }  
}

@Override  
 **public void** onStart() {  
 **super**.onStart();  
 *// Check if user is signed in (non-null) and update UI accordingly.* FirebaseUser currentUser = **mAuth**.getCurrentUser();  
 updateUI(currentUser);  
  
 }  
  
 **private void** firebaseAuthWithGoogle(String idToken) {  
 AuthCredential credential = GoogleAuthProvider.*getCredential*(idToken, **null**);  
 **mAuth**.signInWithCredential(credential)  
 .addOnCompleteListener(**this**, **new** OnCompleteListener<AuthResult>() {  
 @Override  
 **public void** onComplete(@NonNull Task<AuthResult> task) {  
 **if** (task.isSuccessful()) {  
 *// Sign in success, update UI with the signed-in user's information* Log.*d*(***TAG***, **"signInWithCredential:success"**);  
 FirebaseUser user = **mAuth**.getCurrentUser();  
 updateUI(user);  
 } **else** {  
 *// If sign in fails, display a message to the user.* Log.*w*(***TAG***, **"signInWithCredential:failure"**, task.getException());  
 updateUI(**null**);  
 }  
  
 *// ...* }  
 });  
 }  
  
 **private void** updateUI(FirebaseUser user) {  
 String nombre;  
 String correo;  
 String urlFoto;  
  
 **if**(user!= **null**){  
 nombre = user.getDisplayName();  
 correo = user.getEmail();  
 urlFoto = user.getPhotoUrl().toString();  
 Intent intent = **new** Intent(MainActivity.**this**,HomeActivity.**class**);  
 intent.putExtra(**"nombreUser"**,nombre);  
 intent.putExtra(**"correoUser"**,correo);  
 intent.putExtra(**"uriFotoUser"**,urlFoto);  
 startActivity(intent);  
 }  
 }  
  
  
}

Para el menú del usuario se tiene una clase HomeActivity donde se muestra los datos y fotos del usuario cuando inicia sesión con Google.

**public class** HomeActivity **extends** AppCompatActivity {  
  
 **private** AppBarConfiguration **mAppBarConfiguration**;  
 **private** String **fotoUser**;  
 **private** String **nombreUser**;  
 **private** String **correoUser**;  
  
 **private** FirebaseAuth **mAuth**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_home***);  
 Toolbar toolbar = findViewById(R.id.***toolbar***);  
 setSupportActionBar(toolbar);  
  
 **mAuth** = FirebaseAuth.*getInstance*();  
  
 DrawerLayout drawer = findViewById(R.id.***drawer\_layout***);  
 NavigationView navigationView = findViewById(R.id.***nav\_view***);  
  
 View headerView = navigationView.getHeaderView(0);  
  
 **nombreUser** = getIntent().getExtras().getString(**"nombreUser"**);  
 **correoUser** = getIntent().getExtras().getString(**"correoUser"**);  
 **fotoUser** = getIntent().getExtras().getString(**"uriFotoUser"**);  
  
  
 TextView navUsername = (TextView) headerView.findViewById(R.id.***tvNombreUser***);  
 navUsername.setText(**nombreUser**);  
 TextView navUsermail = (TextView) headerView.findViewById(R.id.***tvCorreoUser***);  
 navUsermail.setText(**correoUser**);  
  
 ImageView navUserPhoto = (ImageView) headerView.findViewById(R.id.***imgvwFotoUser***);  
  
 Picasso.*get*().load(**fotoUser**)  
 .placeholder(R.drawable.***ic\_menu\_camera***)  
 .error(R.drawable.***ic\_menu\_gallery***)  
 .into(navUserPhoto);  
  
 *// Passing each menu ID as a set of Ids because each  
 // menu should be considered as top level destinations.* **mAppBarConfiguration** = **new** AppBarConfiguration.Builder(  
 R.id.***nav\_home***, R.id.***nav\_ident\_objeto***)  
 .setDrawerLayout(drawer)  
 .build();  
 NavController navController = Navigation.*findNavController*(**this**, R.id.***nav\_host\_fragment***);  
 NavigationUI.*setupActionBarWithNavController*(**this**, navController, **mAppBarConfiguration**);  
 NavigationUI.*setupWithNavController*(navigationView, navController);  
 }  
  
  
  
  
 @Override  
 **public boolean** onCreateOptionsMenu(Menu menu) {  
 *// Inflate the menu; this adds items to the action bar if it is present.* getMenuInflater().inflate(R.menu.***home***, menu);  
 **return true**;  
 }  
  
 @Override  
 **public boolean** onSupportNavigateUp() {  
 NavController navController = Navigation.*findNavController*(**this**, R.id.***nav\_host\_fragment***);  
 **return** NavigationUI.*navigateUp*(navController, **mAppBarConfiguration**)  
 || **super**.onSupportNavigateUp();  
 }  
  
 **public void** salir(){  
 **mAuth**.signOut();  
 Intent intent = **new** Intent(**this**, MainActivity.**class**);  
 startActivity(intent);  
 }  
  
 @Override  
 **public boolean** onOptionsItemSelected(@NonNull MenuItem item) {  
 **switch** (item.getItemId()){  
 **case** R.id.***action\_salir***:  
 salir();  
 **return true**;  
 **default**:  
 **return super**.onOptionsItemSelected(item);  
  
 }  
  
 }  
}

* Nuestra aplicaciones esta compuesta de varios fragmentos que se explicaran en el manual de usuario.

Se puede visualizar activity HomeFragment que extiende de un fragmento, aquí es donde se mostrara el objeto cuando es pesado por la balanza, la lógica que se utilizado es que el programar compara el peso del objeto que esta en la balanza con los artículos de igual peso en la base de datos, mostrando solo el que tenga el mayor número de aciertos.

**public class** HomeFragment **extends** Fragment {  
  
 *//Variables de mapeo con la vista* **private** TextView **nombreObjeto**;  
 **private** TextView **txtPrecision**;  
 **private** TextView **pesoObjeto**;  
 **private** ImageView **imagenObjeto**;  
 **private** Button **botonAcierto**;  
  
 *//Variables de conexion con la base de Datos* **private** FirebaseInstanceService **servicioBase**;  
 **private** List<Articulo> **listaBase** = **null**;  
 **private float pesoBalanza** = 0;  
 **float acumuladorAciertos**=0;  
 **float acumuladorFallos**=0;  
  
 **public** View onCreateView(@NonNull LayoutInflater inflater,  
 ViewGroup container, Bundle savedInstanceState) {  
  
 View root = inflater.inflate(R.layout.***fragment\_home***, container, **false**);  
  
 **nombreObjeto** = root.findViewById(R.id.***text\_home***);  
 **imagenObjeto** = (ImageView) root.findViewById(R.id.***imgFotoObj***);  
 **pesoObjeto** = root.findViewById(R.id.***txt\_peso\_objeto***);  
 **botonAcierto** = root.findViewById(R.id.***btn\_sensar\_home***);  
 **txtPrecision** = root.findViewById(R.id.***txt\_precision***);  
  
 **servicioBase** = **new** FirebaseInstanceService();  
 **servicioBase**.leerBalanza();  
 **servicioBase**.leerArticulos();  
  
 **txtPrecision**.setVisibility(View.***GONE***);  
  
 **botonAcierto**.setOnClickListener(**new** View.OnClickListener() {  
 **public void** onClick(View v) {  
 actualizarPorPeso();  
 }  
 });  
  
 **nombreObjeto**.setText(**"Pese un objeto!"**);  
 Picasso.*get*().load(R.drawable.***bot\_final***)  
 .placeholder(R.drawable.***ic\_menu\_camera***)  
 .error(R.drawable.***ic\_menu\_gallery***)  
 .into(**imagenObjeto**);  
 **pesoObjeto**.setText(**"El peso de su objeto aparecera aqui!"**);  
  
 **return** root;  
 }

Además dentro del HomeFragment nos encontramos con la función Actualizar peso que al momento de presionar el botón realizara ua peticiona nuestra base de datos para verificar si es que existe un objeto nuevo en la base de datos, solo si existe un objeto diferente este leo y lo muestra al usario con la lógica ya explicada anteriormente.

**public void** actualizarPorPeso(){  
 **pesoBalanza** = **servicioBase**.getPesoBalanza();  
 **listaBase** = **servicioBase**.getListaArticulos();  
  
 **float** precision = 0;  
  
 List<Articulo> listaRepetidos = **new** ArrayList<Articulo>();  
 Articulo mayorAciertos = **null**;  
 **int** aciertosActuales=0;  
  
 **for**(Articulo actual: **listaBase**){  
 **acumuladorAciertos**= **acumuladorAciertos** + actual.getAciertos();  
 **acumuladorFallos** = **acumuladorFallos** + actual.getFallos();  
 **if**(**pesoBalanza**>=actual.getPeso()-0.01 && **pesoBalanza**<=actual.getPeso()+0.01 ){  
 listaRepetidos.add(actual);  
 }  
 }  
  
 **double** preci=0;  
  
 **if**(**acumuladorAciertos**+**acumuladorFallos**!=0){  
 **float** denominador = **acumuladorFallos** + **acumuladorAciertos**;  
 precision= **acumuladorAciertos** / denominador;  
 precision=precision\*100;  
 }  
  
 **txtPrecision**.setVisibility(View.***VISIBLE***);  
 **txtPrecision**.setText(**"La precision actual del sistema es: "**+String.*format* (**"%.2f"**, Double.*valueOf*(precision))+**"%"**);  
  
 **if**(listaRepetidos.size()==0){  
 **nombreObjeto**.setText(**"Sense un objeto!"**);  
 Picasso.*get*().load(R.drawable.***bot\_final***)  
 .placeholder(R.drawable.***ic\_menu\_camera***)  
 .error(R.drawable.***ic\_menu\_gallery***)  
 .into(**imagenObjeto**);  
 **pesoObjeto**.setText(**"El peso de su objeto aparecera aqui!"**);  
 Toast.*makeText*(getContext(),**"No existe un objeto con ese peso"**,Toast.***LENGTH\_LONG***).show();  
 }  
  
 **if**(listaRepetidos.size()==1){  
 **nombreObjeto**.setText(listaRepetidos.get(0).getCategoria());  
 Picasso.*get*().load(listaRepetidos.get(0).getUrl())  
 .placeholder(R.drawable.***ic\_menu\_camera***)  
 .error(R.drawable.***ic\_menu\_gallery***)  
 .into(**imagenObjeto**);  
 **pesoObjeto**.setText(String.*valueOf*(listaRepetidos.get(0).getPeso()));  
 }**else if**(listaRepetidos.size() >1){  
 **for**(Articulo actual: listaRepetidos){  
 **if**(actual.getAciertos()>aciertosActuales){  
 aciertosActuales=actual.getAciertos();  
 mayorAciertos = actual;  
 }  
 }  
  
 **if**(mayorAciertos!=**null**){  
 **nombreObjeto**.setText(mayorAciertos.getCategoria());  
 Picasso.*get*().load(mayorAciertos.getUrl())  
 .placeholder(R.drawable.***ic\_menu\_camera***)  
 .error(R.drawable.***ic\_menu\_gallery***)  
 .into(**imagenObjeto**);  
 **pesoObjeto**.setText(String.*valueOf*(mayorAciertos.getPeso()));  
 }  
 }  
 }  
  
  
  
  
  
}

Luego nos encontramos con la vista entrenamiento donde el usario a través de la lógica programada podrá discrimar a través de objeto según lo que se encuentre en la balanza, este actvtividad también es un fragmet.

**public class** entrenamientoFragment **extends** Fragment {  
  
 *//Variables de conexion con la base de Datos* **private** FirebaseInstanceService **servicioBase**;  
 **private** List<Articulo> **listaBase**;  
 **private float pesoBalanza**;  
  
 **private** TextView **txtNombre**;  
 **private** TextView **txtPeso**;  
 **private** TextView **txtUbicacion**;  
 **private** Button **btnSeleccionar**;  
 **private** Button **btnAnterior**;  
 **private** Button **btnSiguiente**;  
 **private** ImageView **imgFotoArticulo**;  
 List<Articulo> **listaRepetidos**;  
  
 **private** Articulo[] **articulos**;  
  
 **int posicionArticulo**;  
  
 **boolean primeraVuelta**=**true**;  
  
 **public** View onCreateView(@NonNull LayoutInflater inflater,  
 ViewGroup container, Bundle savedInstanceState) {  
  
 View root = inflater.inflate(R.layout.***fragment\_entrenamiento***, container, **false**);  
  
 **servicioBase** = **new** FirebaseInstanceService();  
 **servicioBase**.leerBalanza();  
 **servicioBase**.leerArticulos();  
  
  
 **txtNombre** = root.findViewById(R.id.***txt\_nombre***);  
 **txtPeso** = root.findViewById(R.id.***txt\_peso\_articulo***);  
 **txtUbicacion** = root.findViewById(R.id.***txt\_num\_articulo***);  
  
 **btnSeleccionar** = root.findViewById(R.id.***btn\_seleccionar***);  
 **btnAnterior** = root.findViewById(R.id.***btn\_anterior***);  
 **btnSiguiente** = root.findViewById(R.id.***btn\_siguiente***);  
  
 **imgFotoArticulo** = root.findViewById(R.id.***img\_foto\_articulo***);  
 **btnSiguiente**.setVisibility(View.***GONE***);  
 **btnAnterior**.setVisibility(View.***GONE***);  
  
 **btnSeleccionar**.setText(**"Sensar"**);  
  
 **btnSeleccionar**.setOnClickListener(**new** View.OnClickListener() {  
 **public void** onClick(View v) {  
 sensarSeleccionar();  
 }  
 });  
 **btnAnterior**.setOnClickListener(**new** View.OnClickListener() {  
 **public void** onClick(View v) {  
 moverAnterior();  
 }  
 });  
 **btnSiguiente**.setOnClickListener(**new** View.OnClickListener() {  
 **public void** onClick(View v) {  
 moverSiguiente();  
 }  
 });  
  
 **return** root;  
 }

* Además también nos encontramos con el método seleccionar donde sensara el peso del la balanza y verificara si es que existe algún objeto antes de iniciar el entrenamiento, en caso de que muestra algún objeto que ha sido sensado se procederá al entrenamiento.

**public void** sensarSeleccionar(){  
 **if** (**primeraVuelta**){  
 **posicionArticulo**=0;  
 **btnSeleccionar**.setText(**"Seleccionar"**);  
 sensar();  
 **if**(**listaRepetidos**.size()==0){  
 Toast.*makeText*(getContext(),**"No existe un objeto con ese peso"**,Toast.***LENGTH\_LONG***).show();  
 }**else**{  
 **primeraVuelta**=**false**;  
 }  
 }**else**{  
  
 **txtNombre**.setText(**"Nombre Articulo"**);  
 Picasso.*get*().load(R.drawable.***balanzaprueba***)  
 .placeholder(R.drawable.***ic\_menu\_camera***)  
 .error(R.drawable.***ic\_menu\_gallery***)  
 .into(**imgFotoArticulo**);  
 **txtPeso**.setText(**"Peso Articulo"**);  
 **txtUbicacion**.setText(**"X/X"**);  
  
 **btnSeleccionar**.setText(**"Sensar"**);  
 **btnSiguiente**.setVisibility(View.***GONE***);  
 **btnAnterior**.setVisibility(View.***GONE***);  
  
 *//Actualizar el numero de aciertos o fallos* **if**(**articulos**[**posicionArticulo**-1].isRevisado()){  
 **servicioBase**.getArticulosReference().child(**articulos**[**posicionArticulo**-1].getKey()).child(**"aciertos"**).setValue(String.*valueOf*(**articulos**[**posicionArticulo**-1].getAciertos()+1));  
 **servicioBase**.getArticulosReference().child(**articulos**[**posicionArticulo**-1].getKey()).child(**"fallos"**).setValue(String.*valueOf*(**articulos**[**posicionArticulo**-1].getFallos()-1));  
 }**else**{  
 **servicioBase**.getArticulosReference().child(**articulos**[**posicionArticulo**-1].getKey()).child(**"aciertos"**).setValue(String.*valueOf*(**articulos**[**posicionArticulo**-1].getAciertos()+1));  
 }  
  
  
 **servicioBase**.leerBalanza();  
 **servicioBase**.leerArticulos();  
 Toast.*makeText*(getContext(),**"Articulo Actualizado"**,Toast.***LENGTH\_LONG***).show();  
  
 **primeraVuelta**=**true**;  
 **posicionArticulo**=0;  
 }  
}

luego nos encontramos con el método censar el cual, a través de la instancias a la base da datos, compara y discriminar el articulo para realizar el entrenamiento, aquí en esta función se verificara el rango de peso en el que se puede encontrar un objeto, debido a que la balanza puede tener ciertos valores aberrantes, esta opción los compara y los mostrara al usuario con una exactitud de 0.01gramos.

**public void** sensar(){  
 **posicionArticulo**=0;  
 **pesoBalanza** = **servicioBase**.getPesoBalanza();  
 **listaBase** = **servicioBase**.getListaArticulos();  
  
 **listaRepetidos** = **new** ArrayList<Articulo>();  
 Articulo mayorAciertos = **null**;  
 **int** aciertosActuales=-1;  
  
  
 **int** tamanioArreglo;  
  
 **for**(Articulo actual: **listaBase**){  
 **if**(**pesoBalanza**>=actual.getPeso()-0.01 && **pesoBalanza**<=actual.getPeso()+0.01 ){  
 **listaRepetidos**.add(actual);  
 }  
 }  
  
 **if**(**listaRepetidos**.size()==0){  
 **return**;  
 }  
  
 tamanioArreglo =**listaRepetidos**.size();  
  
 **articulos** = **new** Articulo[tamanioArreglo];  
  
 **int** cont=0;  
  
 **for**(Articulo actual: **listaRepetidos**){  
 actual.setPosicion(cont);  
 **articulos**[cont]=actual;  
 cont++;  
 }  
  
 **if**(**listaRepetidos**.size()==1){  
 **txtNombre**.setText(**listaRepetidos**.get(0).getCategoria());  
 Picasso.*get*().load(**listaRepetidos**.get(0).getUrl())  
 .placeholder(R.drawable.***ic\_menu\_camera***)  
 .error(R.drawable.***ic\_menu\_gallery***)  
 .into(**imgFotoArticulo**);  
 **txtPeso**.setText(String.*valueOf*(**listaRepetidos**.get(0).getPeso()));  
 **posicionArticulo**=1;  
 }**else if**(**listaRepetidos**.size() >1){  
 **for**(Articulo actual: **listaRepetidos**){  
 **if**(actual.getAciertos()>aciertosActuales){  
 aciertosActuales=actual.getAciertos();  
 mayorAciertos = actual;  
 **posicionArticulo**++;  
 }  
 }  
  
 **if**(mayorAciertos!=**null**){  
 **txtNombre**.setText(mayorAciertos.getCategoria());  
 Picasso.*get*().load(mayorAciertos.getUrl())  
 .placeholder(R.drawable.***ic\_menu\_camera***)  
 .error(R.drawable.***ic\_menu\_gallery***)  
 .into(**imgFotoArticulo**);  
 **txtPeso**.setText(String.*valueOf*(mayorAciertos.getPeso()));  
 **posicionArticulo**=mayorAciertos.getPosicion()+1;  
 }  
 }  
  
 **txtUbicacion**.setText(**posicionArticulo**+**"/"**+**articulos**.**length**);  
 **if**(**posicionArticulo**==**articulos**.**length** && **articulos**.**length**!=1){  
 **btnSiguiente**.setVisibility(View.***GONE***);  
 **btnAnterior**.setVisibility(View.***VISIBLE***);  
 }**else if**(**articulos**.**length**==1){  
 **btnSiguiente**.setVisibility(View.***GONE***);  
 **btnAnterior**.setVisibility(View.***GONE***);  
 }**else if**(**posicionArticulo**>1){  
 **btnSiguiente**.setVisibility(View.***VISIBLE***);  
 **btnAnterior**.setVisibility(View.***VISIBLE***);  
 }**else if**(**posicionArticulo**==1){  
 **btnSiguiente**.setVisibility(View.***VISIBLE***);  
 **btnAnterior**.setVisibility(View.***GONE***);  
 }  
}

Para finalizar la vista entrenamiento para el momento de discriminar se usaron dos funciones que dependiendo del objeto que se muestre se deslizara por el menú, para el entrenamiento se mostrara el objeto inicial con mayor numero de acierto y de acuerdo la vista que se muestre esta; si se pasa al siguiente articulo con un peso semejante, se le aumentar el nuemero de acierto al siguiente y al anterior se la aumentara el numero de fallos. Y así sucesivamente hasta que se seleccione al articulo correcto, esto se lo realiza a través de una doble verificación, donde se revisa si el articulo ya se reviso, para no aumentarle el numero de fallos.

**public void** moverSiguiente(){  
 **if**(**posicionArticulo**<=**articulos**.**length**-1){  
 **txtNombre**.setText(**articulos**[**posicionArticulo**].getCategoria());  
 Picasso.*get*().load(**articulos**[**posicionArticulo**].getUrl())  
 .placeholder(R.drawable.***ic\_menu\_camera***)  
 .error(R.drawable.***ic\_menu\_gallery***)  
 .into(**imgFotoArticulo**);  
 **txtPeso**.setText(String.*valueOf*(**articulos**[**posicionArticulo**].getPeso()));  
 **txtUbicacion**.setText(**posicionArticulo**+1+**"/"**+**articulos**.**length**);  
 **if**(!**articulos**[**posicionArticulo**-1].isRevisado()){  
 **servicioBase**.getArticulosReference().child(**articulos**[**posicionArticulo**-1].getKey()).child(**"fallos"**).setValue(String.*valueOf*(**articulos**[**posicionArticulo**-1].getFallos()+1));  
 **articulos**[**posicionArticulo**-1].setFallos(**articulos**[**posicionArticulo**-1].getFallos()+1);  
 **articulos**[**posicionArticulo**-1].setRevisado(**true**);  
 }  
  
 **posicionArticulo**++;  
 **if**(**posicionArticulo**==**articulos**.**length**){  
 **btnSiguiente**.setVisibility(View.***GONE***);  
 }  
 **btnAnterior**.setVisibility(View.***VISIBLE***);  
 }  
  
 }  
  
 **public void** moverAnterior(){  
 **if**(**posicionArticulo**-2>=0){  
 **if**(!**articulos**[**posicionArticulo**-1].isRevisado()){  
 **servicioBase**.getArticulosReference().child(**articulos**[**posicionArticulo**-1].getKey()).child(**"fallos"**).setValue(String.*valueOf*(**articulos**[**posicionArticulo**-1].getFallos()+1));  
 **articulos**[**posicionArticulo**-1].setFallos(**articulos**[**posicionArticulo**-1].getFallos()+1);  
 **articulos**[**posicionArticulo**-1].setRevisado(**true**);  
 }  
 **posicionArticulo**--;  
 **txtNombre**.setText(**articulos**[**posicionArticulo**-1].getCategoria());  
 Picasso.*get*().load(**articulos**[**posicionArticulo**-1].getUrl())  
 .placeholder(R.drawable.***ic\_menu\_camera***)  
 .error(R.drawable.***ic\_menu\_gallery***)  
 .into(**imgFotoArticulo**);  
 **txtPeso**.setText(String.*valueOf*(**articulos**[**posicionArticulo**-1].getPeso()));  
 **txtUbicacion**.setText(**posicionArticulo**+**"/"**+**articulos**.**length**);  
  
 **if**(**posicionArticulo**-1==0){  
 **btnAnterior**.setVisibility(View.***GONE***);  
 }  
 **btnSiguiente**.setVisibility(View.***VISIBLE***);  
  
 }  
  
 }  
  
}

Codigo xml ACTIVITY HOME

En el activityHome, se mostrara las opciones para el entrenamiento y la vista principal. Además de los datos del usuario.

*<?***xml version="1.0" encoding="utf-8"***?>*<**androidx.drawerlayout.widget.DrawerLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/drawer\_layout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:fitsSystemWindows="true"  
 tools:openDrawer="start"**>  
  
 <**include  
 layout="@layout/app\_bar\_main"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"** />  
  
 <**com.google.android.material.navigation.NavigationView  
 android:id="@+id/nav\_view"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="start"  
 android:fitsSystemWindows="true"  
 app:headerLayout="@layout/nav\_header\_main"  
 app:menu="@menu/activity\_main\_drawer"** />  
</**androidx.drawerlayout.widget.DrawerLayout**>

MAIN ACTIVITY, aquí se muestra la pantalla principal y el inicio de sesión.

*<?***xml version="1.0" encoding="utf-8"***?>*<**androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@drawable/fondo"  
 android:orientation="vertical"  
 tools:context=".MainActivity"**>  
  
  
 <**EditText  
 android:id="@+id/etUsuario"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="100dp"  
 android:layout\_marginTop="32dp"  
 android:layout\_marginEnd="101dp"  
 android:ems="10"  
 android:hint="usuario"  
 android:inputType="textPersonName"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/imageView3"  
 tools:ignore="MissingConstraints"** />  
  
 <**Button  
 android:id="@+id/button4"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="40dp"  
 android:layout\_marginEnd="100dp"  
 android:background="@drawable/round"  
 android:text="Salir"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/edContrasena"** />  
  
 <**Button  
 android:id="@+id/button3"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="100dp"  
 android:layout\_marginTop="40dp"  
 android:background="@drawable/round"  
 android:onClick="ingresar"  
 android:text="Acceder"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/edContrasena"** />  
  
 <**ImageView  
 android:id="@+id/foto"  
 android:layout\_width="192dp"  
 android:layout\_height="173dp"  
 android:layout\_marginStart="108dp"  
 android:layout\_marginTop="28dp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/btnLoginGoogle"** />  
  
 <**EditText  
 android:id="@+id/edContrasena"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="100dp"  
 android:layout\_marginTop="16dp"  
 android:ems="10"  
 android:hint="contraseña"  
 android:inputType="textPassword"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/etUsuario"** />  
  
 <**Button  
 android:id="@+id/btnLoginGoogle"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="17dp"  
 android:layout\_marginTop="50dp"  
 android:layout\_marginEnd="100dp"  
 android:background="@drawable/round"  
 android:text="Acceder con Google"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toEndOf="@+id/imageView4"  
 app:layout\_constraintTop\_toBottomOf="@+id/button4"** />  
  
 <**ImageView  
 android:id="@+id/imageView3"  
 android:layout\_width="321dp"  
 android:layout\_height="93dp"  
 android:layout\_marginStart="45dp"  
 android:layout\_marginTop="50dp"  
 android:layout\_marginEnd="45dp"  
 android:layout\_marginBottom="32dp"  
 app:layout\_constraintBottom\_toTopOf="@+id/etUsuario"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:srcCompat="@drawable/logo3"** />  
  
 <**ImageView  
 android:id="@+id/imageView4"  
 android:layout\_width="42dp"  
 android:layout\_height="52dp"  
 android:layout\_marginStart="100dp"  
 android:layout\_marginTop="50dp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/button3"  
 app:srcCompat="@drawable/googleg\_standard\_color\_18"** />  
  
</**androidx.constraintlayout.widget.ConstraintLayout**>

Fragment Home, donde se muestra el resultado de un objeto al momento de poner un articulo en la balanza. Se compone de textview y imageview

*<?***xml version="1.0" encoding="utf-8"***?>*<**androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/constrain1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@drawable/fondo"  
 tools:context=".Entrenamiento"**>  
  
 <**ImageView  
 android:id="@+id/imageView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:src="@drawable/logo3"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"** />  
  
 <**ImageView  
 android:id="@+id/imgFotoObj"  
 android:layout\_width="324dp"  
 android:layout\_height="282dp"  
 android:layout\_marginTop="16dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.494"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/text\_home"  
 app:srcCompat="@drawable/bot\_final"** />  
  
 <**TextView  
 android:id="@+id/text\_home"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="12dp"  
 android:text="Sense un objeto!"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.5"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/imageView"** />  
  
 <**TextView  
 android:id="@+id/txt\_peso\_objeto"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="24dp"  
 android:text="Aqui aparecera el peso de su objeto!"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.498"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/imgFotoObj"** />  
  
 <**Button  
 android:id="@+id/btn\_sensar\_home"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:text="Sensar"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.498"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_peso\_objeto"** />  
  
 <**TextView  
 android:id="@+id/txt\_precision"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Precision Actual"  
 app:layout\_constraintBottom\_toTopOf="@+id/imgFotoObj"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.5"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/text\_home"** />  
  
</**androidx.constraintlayout.widget.ConstraintLayout**>

Activity Entrenamiento compuesto por textView e image view que se cargan directamente de la base de datos.

*<?***xml version="1.0" encoding="utf-8"***?>*<**androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@drawable/fondo"  
 tools:context=".IdentObjetoViewModel"**>  
  
 <**TextView  
 android:id="@+id/txt\_nombre"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Nombre Articulo"  
 app:layout\_constraintBottom\_toTopOf="@+id/img\_foto\_articulo"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/imageView2"  
 app:layout\_constraintVertical\_bias="0.0"** />  
  
 <**ImageView  
 android:id="@+id/imageView2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="4dp"  
 android:src="@drawable/logo3"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"** />  
  
 <**ImageView  
 android:id="@+id/img\_foto\_articulo"  
 android:layout\_width="330dp"  
 android:layout\_height="232dp"  
 android:layout\_marginTop="28dp"  
 android:src="@drawable/balanzaprueba"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.493"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/imageView2"** />  
  
 <**TextView  
 android:id="@+id/txt\_peso\_articulo"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="24dp"  
 android:text="Peso Articulo"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.506"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/img\_foto\_articulo"** />  
  
 <**Button  
 android:id="@+id/btn\_seleccionar"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="49dp"  
 android:layout\_marginTop="52dp"  
 android:text="Seleccionar"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.498"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_peso\_articulo"** />  
  
 <**Button  
 android:id="@+id/btn\_anterior"  
 android:layout\_width="60dp"  
 android:layout\_height="49dp"  
 android:layout\_marginEnd="8dp"  
 android:text="ant"  
 app:layout\_constraintEnd\_toStartOf="@+id/btn\_seleccionar"  
 app:layout\_constraintTop\_toTopOf="@+id/btn\_seleccionar"** />  
  
 <**TextView  
 android:id="@+id/textView3"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="4dp"  
 android:text="Seleccione si este es su objeto"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.502"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/btn\_seleccionar"** />  
  
 <**Button  
 android:id="@+id/btn\_siguiente"  
 android:layout\_width="60dp"  
 android:layout\_height="49dp"  
 android:layout\_marginStart="8dp"  
 android:text="sgt"  
 app:layout\_constraintStart\_toEndOf="@+id/btn\_seleccionar"  
 app:layout\_constraintTop\_toTopOf="@+id/btn\_seleccionar"** />  
  
 <**TextView  
 android:id="@+id/txt\_num\_articulo"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="16dp"  
 android:text="X/X"  
 app:layout\_constraintBottom\_toTopOf="@+id/btn\_seleccionar"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_peso\_articulo"** />  
</**androidx.constraintlayout.widget.ConstraintLayout**>

Código Arduino

Para el código de Arduino se tomaran los valores de la balanza a través de una librería y así mismo el valor de la batería para poder enviralas a través de un callback a sigfox

#include <WISOL.h>

#include <Wire.h>

#include <math.h>

#include "HX711.h"

// definimos variables de entrada

#define DOUT A1

#define CLK A0

#define ANALOGPILA A3

Isigfox \*Isigfox = new WISOL();

typedef union{

uint16\_t number;

uint8\_t bytes[2];

} UINT16\_t;

//se instacia el metodo de la clase

HX711 balanza(DOUT, CLK);

// se declara variables globales para obtener la bateria

int analogValor = 0; // variable para contar el tiempo de envio de bateria

float previousMillis = 0; // variable para setear el inicio del tiempo

float intervaloMaximo = 60000; // variable para setear el m�ximo tiempo

int voltaje = 0; // variable para obtener el voltaje de la bateria

int porcentaje\_voltaje=0; // variable para obtener el porcentaje de bateria

boolean tiempo = false; // variable

float volt=0;

// variables globales del proceso

float peso; // para obtener el peso

int categoria; // setear categoria

int bateria; // para obtener el porcentaje de bateria

float peso\_anterior=0;

float notificacion= 0;

// se inicializa las variables

void setup() {

Wire.begin();

Wire.setClock(100000);

Serial.begin(9600);

// WISOL modem test

Isigfox->initSigfox();

Isigfox->testComms();

//Serial.println(balanza.read());

Serial.print("Lectura del valor del ADC: ");

Serial.println(balanza.read());

Serial.println("No ponga ningun objeto sobre la balanza");

Serial.println("Destarando...");

Serial.println("...");

balanza.set\_scale(439430.25);

balanza.tare(20); //Reset the scale to 0

Serial.println("Listo para pesar");

}

void loop() {

peso = obtenerPeso();

bateria = obtener\_porcentaje\_bateria();

categoria = obtenerCategoria(peso);

Serial.print("Peso: ");

Serial.print(peso);

Serial.println(" kg");

Serial.print("Categoria: ");

Serial.println(categoria);

if ((peso <= peso\_anterior-0.01) or (peso >= peso\_anterior+0.01)){

byte \*float\_byte = (byte \*)&peso;

byte \*float\_byte2 = (byte \*)&categoria;

byte \*float\_byte3 = (byte \*)&bateria;

const uint8\_t payloadSize = 9;

uint8\_t buf\_str[payloadSize];

buf\_str[0] = float\_byte3[0];

buf\_str[1] = float\_byte3[1];

buf\_str[2] = float\_byte[0];

buf\_str[3] = float\_byte[1];

buf\_str[4] = float\_byte[2];

buf\_str[5] = float\_byte[3];

buf\_str[6] = float\_byte2[0];

buf\_str[7] = float\_byte2[1];

Send\_Pload(buf\_str, payloadSize);

Serial.println("\_Se envia");

//delay(20000);

}

peso\_anterior=peso;

tiempo = retardo();

if (tiempo==true){

envio\_porcentaje();

}

}

// funcion para obtener el peso del objeto colocado en la balanza

float obtenerPeso(){

//Serial.print("Peso: ");

float peso1 ;

peso1 = (balanza.get\_units(20)\*-1);

//Serial.println(" kg");

delay(500);

if (peso1 < 0){

peso1 = 0;

}

return peso1;

}

// funcion para obtener la categoria de acuerdo a un perfil

int getCategoria(float peso){

int categoria;

if (peso <= 0.040){

categoria = 1;//No existe articulo sobre balanza

}

else if (peso >= 0.1 and peso< 0.230){

categoria = 2;//celular o electronico del mismo tamano

}

else if (peso>= 0.370 and peso <= 0.750){

categoria = 3;// tablets y Ipads

}

else if (peso>= 0.800 and peso <= 1.800){

categoria =4;//laptop

}

else if (peso>= 0.045 and peso <= 0.099){

categoria =5;//cuaderno

}

return categoria;

}

// funcion para el envio de datos a sigfox

void Send\_Pload(uint8\_t \*sendData, int len) {

recvMsg \*RecvMsg;

RecvMsg = (recvMsg \*)malloc(sizeof(recvMsg));

Isigfox->sendPayload(sendData, len, 0, RecvMsg);

for (int i = 0; i < RecvMsg->len; i++) {

Serial.print(RecvMsg->inData[i]);

}

Serial.println("");

free(RecvMsg);

}

// funcion para cintabilizar el tiempo

boolean retardo(){

float currentMillis = millis();

if(currentMillis - previousMillis > intervaloMaximo) { //Cuando se cumplen los 10 minutos

previousMillis = currentMillis;

return true;

}else{

return false;

}

}

// funcion para obtener el porcentaje de bateria

int obtener\_porcentaje\_bateria(){

analogValor = analogRead(ANALOGPILA);

// Obtenemos el voltaje

voltaje = int(0.0048 \* analogValor);

porcentaje\_voltaje = int (voltaje \* 22.22);

Serial.print("Voltaje: ");

Serial.println(voltaje);

Serial.print("Porcentaje Voltaje: ");

Serial.println(porcentaje\_voltaje);

return porcentaje\_voltaje;

}

// funcion para enviar el porcentaje de bateria al backend de sigfox

void envio\_porcentaje(){

volt = int (obtener\_porcentaje\_bateria());

byte \*float\_byte = (byte \*)&volt;

const uint8\_t payloadSize = 3;

uint8\_t buf\_str[payloadSize];

buf\_str[0] = float\_byte[0];

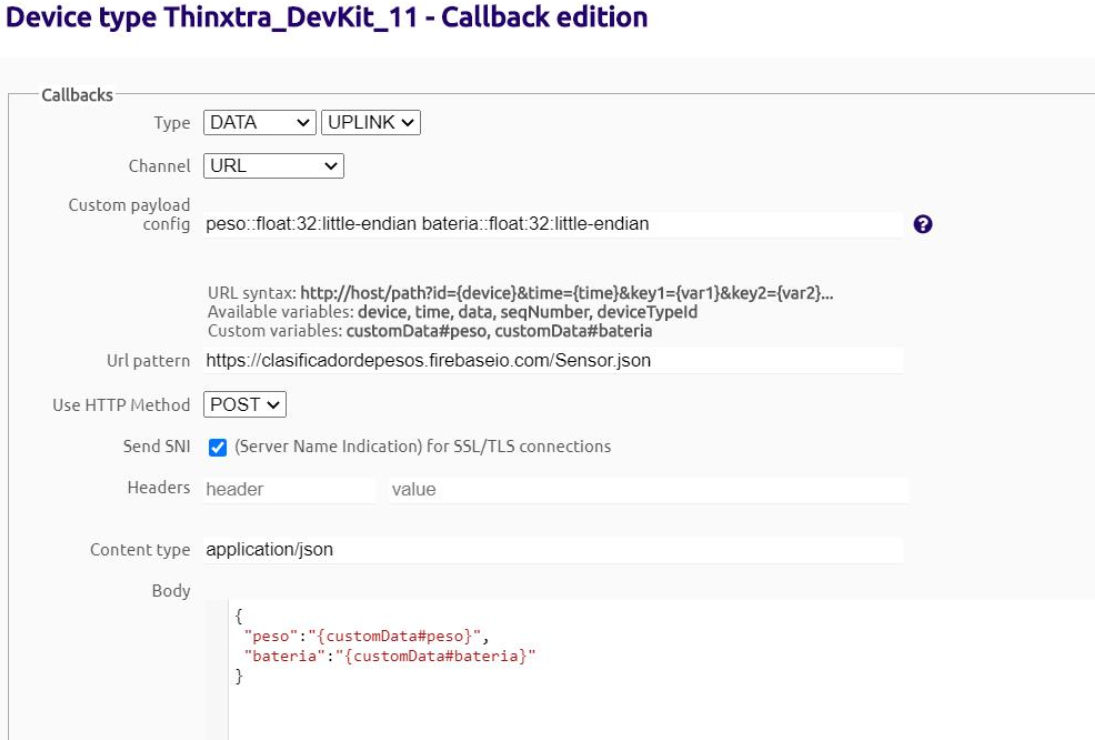
buf\_str[1] = float\_byte[1];

Send\_Pload(buf\_str, payloadSize);

Serial.println("\_Se envia bateria");

}

**configuración del callback sigfox.**



Link del repositorio de GITHUB del proyecto

<https://github.com/amst-fiec/1T2020-Clasificador-de-objetos-mediante-peso-v2.0.git>

**Bibliografía**

https://app.lucidchart.com/invitations/accept/87840040-964c-4a0f-b3ae-32978e49fe5c

<https://es.wikipedia.org/wiki/Sigfox>

<https://naylampmechatronics.com/blog/25_tutorial-trasmisor-de-celda-de-carga-hx711-ba.html>

<https://es.wikipedia.org/wiki/Resistor>