



**APLICACIONES MÓVILES Y SISTEMAS TELEMÁTICOS**

**MANUAL TECNICO DEL “FTANK”**



**PROFESORA:** MSIG. ADRIANA COLLAGUAZO

**GRUPO Nº7**

**INTEGRANTES:**

KLEBER JOSE AVELINO MOSQUERA

VICTOR BRYAN RODRIGUEZ AGUAS

CARLOS BRYAN TOMALÁ REYES

LUIS ADRIAN ZAMORA VELIZ

**2s-2019**

**Introducción**

El presente manual tiene como finalidad explicar los aspectos técnicos de hardware y software que posee nuestro prototipo de aplicativo móvil más conocido como TANK. El servicio que presentamos tiene la función principal de presentar a nuestros usuarios (chefs, dueños de restaurants) el nivel de los tanquen de gas dentro de su establecimiento y que puedan tener el control de dichos bienes en tiempo real.

El no conocer esta información de manera oportuna puede ocasionar problemas a nuestros usuarios, ya que el encender la hornilla sin éxito retrasaría las ordenes dentro del establecimiento causando inconformidad de los comensales hasta lograr pérdidas económicas en muchos casos.

Por eso le presentamos el prototipo de FTANK que es un sistema IOT que recolecta información cada 10 segundos con la ayuda de un sensor de presión y los administra para poder informarle a los usuarios el estado su tanque, estos pueden ser LLENO o VACIO los que son visualizados dentro de nuestra aplicación mediante una interfaz gráfica agradable.

**Hardware**

**Velostat**

Es un material hecho de un polímero, el cual permite variar la conductividad del material bajo presión o torsión. Este material puede ser cortado y adaptado para distintas aplicaciones conservando sus propiedades físicas en cualquiera de los casos.



Ilustración 1 Velostat

Este material es utilizado en nuestro proyecto para poder censar los niveles que tiene el tanque de gas, con la ayuda de un divisor de voltaje, ya que este material su resistencia varia y con esto podemos conseguir los datos necesarios para nuestro servicio.

**Arduino UNO**

Es una placa electrónica basada en un microcontrolador ATmega328, posee 14 entradas digitales de las cuales 6 sirven como entradas PWM(Modulación por ancho de pulsos), y 6 entradas analógicas, además de un resonador cerámico de 16 MHz, un conector USB, un conector de alimentación, una cabecera ICSP y un botón de reseteado.

|  |  |
| --- | --- |
| Microcontrolador | ATmega328 |
| Voltaje | 5V[V] |
| Voltaje entrada(recomendado) | 7-12[V] |
| Voltaje limites | 6-20[V] |
| Digital I/O Pins | 14 (6PWM y 6 analógicas) |
| Corriente DC por I/O Pin | 40[mA] |
| Corriente DC para 3.3V Pin | 50[mA] |
| Memoria Flash | 32KB |
| SRAM | 2KB |
| EEPROM | 1KB |
| Velocidad del CLOCK | 16[Mhz] |

A circuit board

Description automatically generated

Ilustración 2 Arduino UNO



Ilustración 3 Ibutton

**I Button**

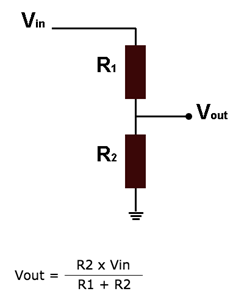
Es un componente de Hardware el cual nos ayuda a mantener objetivos localizados en posición, ya con el ibutton se puede obtener ubicaciones en tiempo real.

**Sigfox Kit**

Es una tecnología de bajo costo, basada para sistemas M2M de bajas velocidades, es una alternativa muy económica ya que no usa internet para transmitir, sino que usa su propia tecnología para enviar sus mensajes a su plataforma. La única limitación que podemos encontrar de esta solución IOT es que solo se pueden enviar hasta 140 mensajes diarios. Es una tecnología de bajo uso energético ya que puede producir su propia energía para trabajar y evita el uso de baterías.



Ilustración 4 Equipo Sigfox

**Divisor de Voltaje**

Es un circuito básico, el cual nos ayuda a medir los valores de voltaje en los extremos de un elemento pasivo, este depende de la impedancia que poseen los elementos pasivos conectados al sistema.

La importancia de este proceso matemático dentro del proyecto es que el valor de la resistencia del velostat varía dependiendo de la presión que perciba de parte del tanque de gas, haciendo que los voltajes en las terminales varíen y estos valores sean enviados a la red de sigfox para ser administrados.

Ilustración 5 Divisor de Voltaje

**Elementos Pasivos**

* Resistencia de 1[KΩ]
* Cables conectores (Jumpers)
* Protoboard

SOFTWARE

**Android Studio**

Es un Entorno de Desarrollo Integrado para implementar aplicaciones móviles oficial para Android. Es un IDE para desarrollo gratuito y que admite los lenguajes de programación Java, C++ y Kotlin.

|  |  |  |
| --- | --- | --- |
|  | Windows | OS X/macOS |
| OS version | Windows 10/8/7 (32- o 64-bit) | Mac OS X 10.10 (Yosemite) o superior, hasta 10.13 ( macOS High Sierra) |



Ilustración 6 Logo de Android Studio

**Firebase RealTime Data Base**

Es una plataforma administrada por Google para el desarrollo de aplicaciones móviles, la cual almacena y sincroniza datos en tiempo real además de que se mantiene funcionando inclusive si la aplicación no se encuentra funcionando.

Esta plataforma aloja en la Nube una base de datos NoSQL, los cuales son almacenados en formato JSON para uso de todos los clientes conectados a la red. Además de que Firebase es multiplataformas es decir puedes desarrollado con SDK de iOS, Android y JavaScript. Todos los clientes se encuentran conectados a una instancia RealTime Data Base la cual hace actualizaciones periódicamente.



Ilustración 7 Logo de Firebase

**Prototipo**

**A picture containing text

Description automatically generated**

Ilustración 8 Diagrama eléctrico del sistema

Este prototipo presentamos una plataforma a base de madera, donde se encuentra el velostat, limitamos sus extremos con un material conductor conocido como baquelita, este nos ayuda a definir los extremos del velostat además de ayudarlo con su conductividad. Esto extremos van a ser conectados como la resistencia variable del divisor de voltaje antes mencionado.



Ilustración 9 Prototipo

|  |  |  |
| --- | --- | --- |
| **Tabla 1. Características del Prototipo** | | |
| **Item** | **Cantidad** | **Dimensiones** |
| **Base de Madera** | 2 | 10 x 11 x 0.6[cm] |
| **Velostat** | 1 | 5x5 [cm] |
| **Baquelita** | 2 | 5x4 [cm] |

**Diseño**

**Diagrama de casos de uso**

El diagrama de casos de uso nos proporciona el primer paso para el diseño del sistema IOT analizando los posibles escenarios que los usuarios pueden encontrar, y brindando soluciones optimas a cada uno de los problemas que ellos tienen.

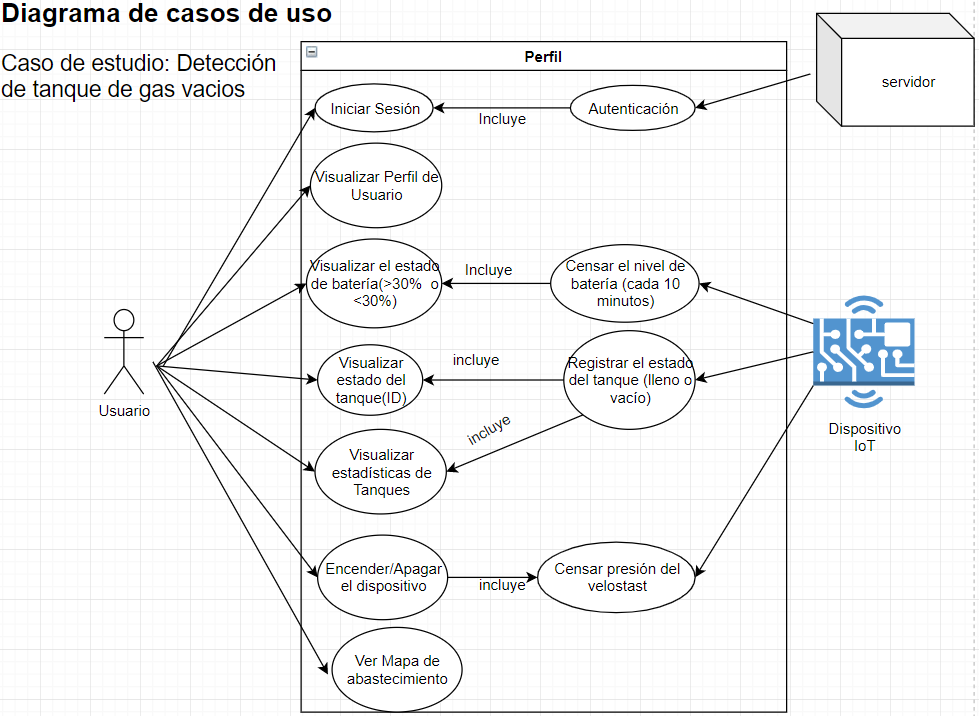


Ilustración 10 Diagrama de Casos de Uso

**Diagrama de Clases**

Luego de haber realizado el análisis de los problemas y las posibles soluciones del proyecto, se procede a crear las clases que posee el aplicativo y así poder ordenar de una manera mas estructurada todos los objetos que este va a tener.

En este caso el sistema posee los siguientes objetos:

* **Usuario** (se define todas las características y acciones que va a tener la persona que utilice la aplicación).
* **Notificación** (mensaje donde se informa el estado actual del tanque si este está lleno o vacío).
* **InterfazUsuario** (se definen las características y acciones que va realizar la aplicación con la información de los usuarios).
* **Mapa** (muestra la información de ubicación y contacto al usuario de lugares donde se pueden abastecer de tanques de gas).
* **TanqueGas** (se muestra el objeto a censar, se detecta si se encuentra o no en la plataforma y su estado).

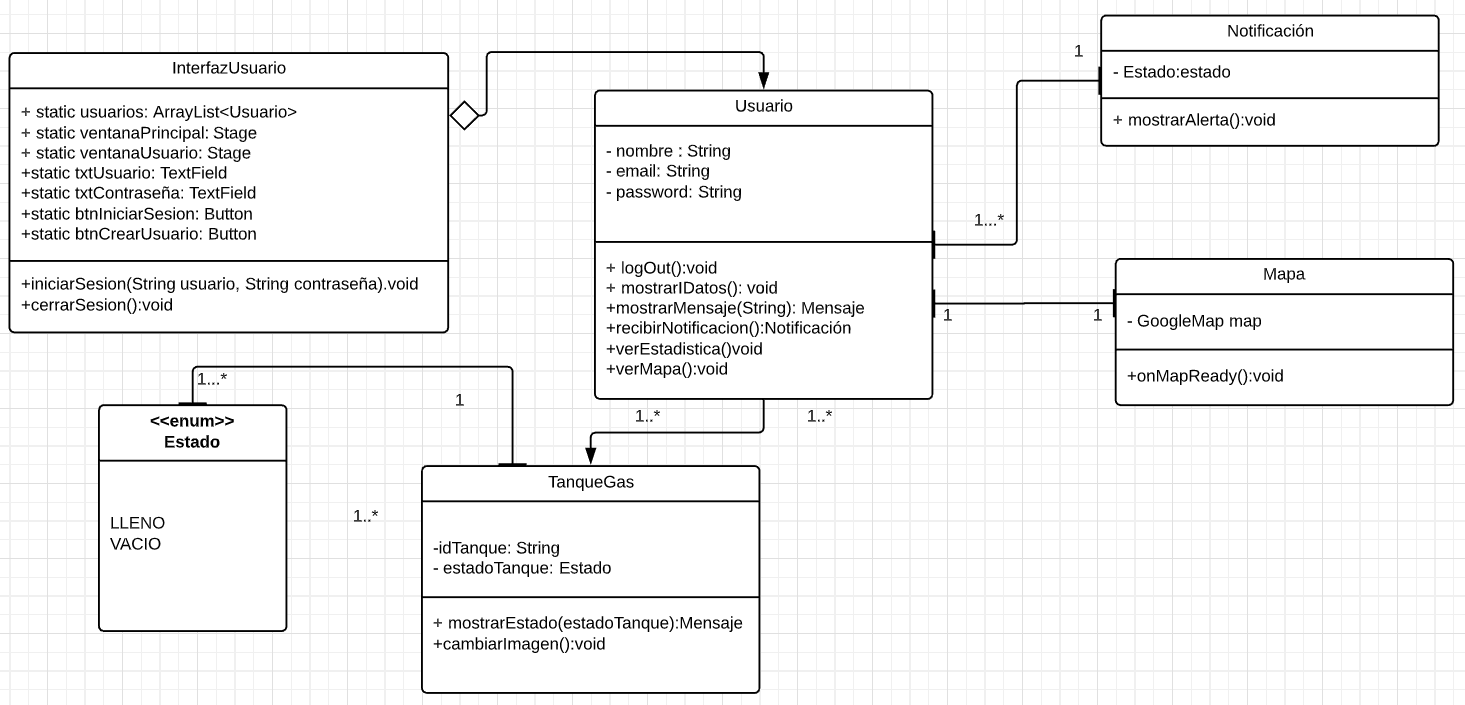


Ilustración 11 Diagrama de Clases

**Diagrama de Despliegue**

En este diagrama se muestra un poco mas a detalle la relación entre el aplicativo móvil y los distintos componentes que existen para la comunicación y administración de la aplicación, como son:

* **Dispositivo Android** (Aplicación Móvil)
* **Servidor Web** (Contiene la base de datos no relacional, que va a ser administrada)
* **Sigfox** (Canal que envía los datos obtenidos por el dispositivo IOT hacia el servidor web)
* **Dispositivo IOT** (Sistema de Hardware que obtiene los datos)

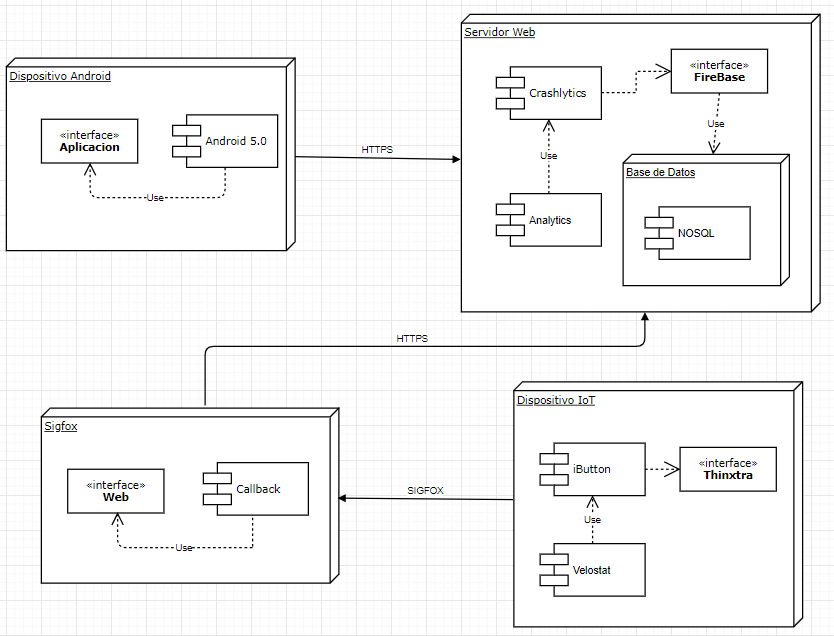


Ilustración 12 Diagrama de Despliegue

**Diagrama de Base de Datos**

Se muestran las distintas bases de datos que se van a generar en el Firebase y sus características internas de cada uno.

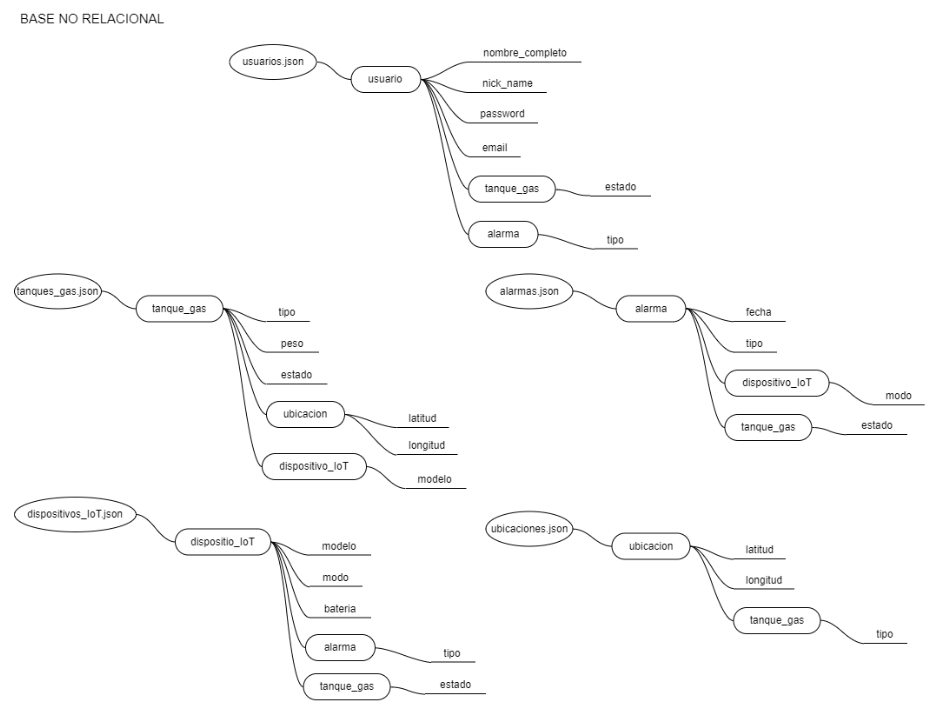


Ilustración 13 Diagrama de Base de Datos

**Presupuesto**

|  |  |  |  |
| --- | --- | --- | --- |
| Tabla 2. Presupuesto del sistema | | | |
| Articulo | **Cantidad** | **Precio** | **Precio Total** |
| Jumpers | 20 | $0.15 | $3 |
| Arduino UNO | 1 | $8.50 | $8.50 |
| Protoboard | 1 | $9 | $9 |
| Cajita | 1 | $26 | $26 |
| Opam | 1 | $0.75 | $0.75 |
| Base de Madera | 1 | $4 | $4 |
| Android Studio | 1 | GRATUITO | $0 |
| Total | | | $51.25 |

**Código Fuente**

|  |
| --- |
| **Pantalla de Bienvenida** |
| Es la primera pantalla que es presentada al abrir la aplicación por uno cuantos segundos, donde es presentado el nombre del aplicativo “FTANK” y nuestro logotipo. |
| **Java** |
| **package** com.example.detecciontanquegas;  **import** androidx.appcompat.app.AppCompatActivity;  **import** android.content.Intent; **import** android.os.Bundle; **import** android.os.Handler;  **public class** PantallaDeBienvenida **extends** AppCompatActivity {    @Override  **protected void** onCreate( Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_pantalla\_de\_bienvenida***);   **new** Handler().postDelayed(**new** Runnable() {  @Override  **public void** run() {  Intent intent = **new** Intent(PantallaDeBienvenida.**this**, MainActivity.**class**);  startActivity(intent);  finish();   }  },3000);  } } |
| **XML** |
| *<?***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/linearLayout"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:background="@drawable/bg"  android:gravity="center"**>   <**ImageView  android:id="@+id/imageView2"  android:layout\_width="374dp"  android:layout\_height="573dp"  android:layout\_marginEnd="16dp"  android:layout\_marginBottom="110dp"   android:scaleType="centerInside"  android:src="@drawable/logo"  app:layout\_constraintBottom\_toBottomOf="parent"  app:layout\_constraintEnd\_toEndOf="parent"** />   <**androidx.constraintlayout.widget.Guideline  android:id="@+id/guideline"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:orientation="horizontal"  app:layout\_constraintGuide\_begin="20dp"** />   <**androidx.constraintlayout.widget.Guideline  android:id="@+id/guideline2"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:orientation="horizontal"  app:layout\_constraintGuide\_begin="20dp"** />  </**androidx.constraintlayout.widget.ConstraintLayout**> |
| **Diseño** |
| Ilustración 14 Pantalla de Bienvenida |

|  |
| --- |
| **Pantalla Principal – MainActivity** |
| En esta actividad inicial toda la aplicación móvil y es donde se presenta el panel de inicio de sesión para que el usuario sea capaz de iniciar sesión mediante Google o de un perfil previamente creado. |
| **Java** |
| **package** com.example.detecciontanquegas;  **import** android.content.Intent; **import** android.media.MediaPlayer; **import** android.net.Uri; **import** android.os.Bundle;  **import** android.text.TextUtils; **import** android.util.Log; **import** android.util.Patterns; **import** android.view.View; **import** android.widget.Button; **import** android.widget.EditText; **import** android.widget.ImageView; **import** android.widget.ProgressBar; **import** android.widget.TextView; **import** android.widget.Toast; **import** android.widget.VideoView;  **import** androidx.annotation.NonNull; **import** androidx.appcompat.app.AppCompatActivity;  **import** com.google.android.gms.auth.api.signin.GoogleSignIn; **import** com.google.android.gms.auth.api.signin.GoogleSignInAccount; **import** com.google.android.gms.auth.api.signin.GoogleSignInClient; **import** com.google.android.gms.auth.api.signin.GoogleSignInOptions; **import** com.google.android.gms.common.api.ApiException; **import** com.google.android.gms.tasks.OnCompleteListener; **import** com.google.android.gms.tasks.OnFailureListener; **import** com.google.android.gms.tasks.Task; **import** com.google.firebase.auth.AuthCredential; **import** com.google.firebase.auth.AuthResult; **import** com.google.firebase.auth.FirebaseAuth; **import** com.google.firebase.auth.FirebaseUser; **import** com.google.firebase.auth.GoogleAuthProvider;  **import** java.util.regex.Pattern;   **public class** MainActivity **extends** AppCompatActivity {  EditText **txtUsuario**, **txtPasswd**;  Button **btnLogin**, **btnRegistro**;  TextView **registrobtn**;  GoogleSignInClient **mGoogleSignInClient**;  GoogleSignInOptions **gso**;  **static final int *GOOGLE\_SIGN\_IN*** = 123;  ImageView **googleicon**;  *//datos para iniciar sesion* String **email**;  String **password**;  *//obtener informacion del usuario* FirebaseAuth **nAuth**;  *// atributos para el video* **private** VideoView **videoBG**;  MediaPlayer **mMediaPlayer**;  **int mCurrentVideoPosition**;    @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);  *//Referencias a los controles* **nAuth** = FirebaseAuth.*getInstance*();    Videothing(); *// SOBRE EL VIDEO* **gso** = **new** GoogleSignInOptions.Builder(GoogleSignInOptions.***DEFAULT\_SIGN\_IN***)  .requestIdToken(getString(R.string.***google\_app\_id***)).requestEmail()  .build();  **mGoogleSignInClient** = GoogleSignIn.*getClient*(**this**, **gso**);    **googleicon** = (ImageView) findViewById(R.id.***googleicon***);  **txtUsuario** = (EditText) findViewById(R.id.***txtUsuario***);  **txtPasswd** = (EditText) findViewById(R.id.***txtPasswd***);  **btnLogin** = (Button) findViewById(R.id.***btnLogin***);  **registrobtn** = (TextView) findViewById(R.id.***signIn\_text***);  **registrobtn**.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  startActivity(**new** Intent(MainActivity.**this**, formulario\_registro.**class**));  }  });   **googleicon**.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  Intent signInIntent = **mGoogleSignInClient**.getSignInIntent();  startActivityForResult(signInIntent, ***GOOGLE\_SIGN\_IN***);  }  });  **btnLogin**.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {   String email = **txtUsuario**.getText().toString();  String password=**txtPasswd**.getText().toString();  **if**(!Patterns.***EMAIL\_ADDRESS***.matcher(email).matches()){  *//invalid email patter set error* **txtUsuario**.setError(**"Formato Incorrecto"**);  **txtUsuario**.setFocusable(**true**);  }  **else if**(TextUtils.*isEmpty*(email)){  **txtUsuario**.setError(**"SE REQUIERE CORREO ELECTRONICO"**);  **txtUsuario**.setFocusable(**true**);  }  **else if**(TextUtils.*isEmpty*(password)){  **txtPasswd**.setError(**"SE REQUIERE SU CONTRASEÑA"**);  **txtPasswd**.setFocusable(**true**);   }  **else**{  loginUser(email,password);  }      }  });   }   **private void** loginUser(String email,String password){  **nAuth**.signInWithEmailAndPassword(email, password)  .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* FirebaseUser user = **nAuth**.getCurrentUser();  startActivity(**new** Intent(MainActivity.**this**,PantallaPrincipal.**class**));  finish();   } **else** {  *// If sign in fails, display a message to the user.* Toast.*makeText*(MainActivity.**this**, **"Authentication failed."**,  Toast.***LENGTH\_SHORT***).show();   }  }  }).addOnFailureListener(**new** OnFailureListener() {  @Override  **public void** onFailure(@NonNull Exception e) {  Toast.*makeText*(MainActivity.**this**, e.getMessage(),  Toast.***LENGTH\_SHORT***).show();  }  });  }  **private void** Videothing(){  **videoBG** = (VideoView) findViewById(R.id.***videoView***);  Uri uri = Uri.*parse*(**"android.resource://"** *// First start with this,* + getPackageName() *// then retrieve your package name,* + **"/"** *// add a slash,* + R.raw.***fire***); *// and then finally add your video resource. Make sure it is stored  // in the raw folder.* **videoBG**.setVideoURI(uri);  *// Start the VideoView* **videoBG**.start();  **videoBG**.setOnPreparedListener(**new** MediaPlayer.OnPreparedListener() {  @Override  **public void** onPrepared(MediaPlayer mediaPlayer) {  **mMediaPlayer** = mediaPlayer;  *// We want our video to play over and over so we set looping to true.* **mMediaPlayer**.setLooping(**true**);  *// We then seek to the current posistion if it has been set and play the video.* **if** (**mCurrentVideoPosition** != 0) {  **mMediaPlayer**.seekTo(**mCurrentVideoPosition**);  **mMediaPlayer**.start();  }  }  });  }  **protected void** onActivityResult(**int** requestCode, **int** resultCode, Intent data) {  **super**.onActivityResult(requestCode, resultCode, data);  **if** (requestCode == ***GOOGLE\_SIGN\_IN***) {  Task<GoogleSignInAccount> task = GoogleSignIn.*getSignedInAccountFromIntent*(  data);  **try** {  GoogleSignInAccount account =task.getResult(ApiException.**class**);  **if** (account != **null**)  firebaseAuthWithGoogle(account);   } **catch** (ApiException e) {  Log.*w*(**"TAG"**, **"Fallo el inicio de sesión con google."**, e);  }  }  }  **private void** firebaseAuthWithGoogle(GoogleSignInAccount acct) {  Log.*d*(**"TAG"**, **"firebaseAuthWithGoogle:"** + acct.getId());  AuthCredential credential = GoogleAuthProvider.*getCredential*(acct.getIdToken(),**null**);  **nAuth**.signInWithCredential(credential).addOnCompleteListener(**this**, **new** OnCompleteListener<AuthResult>() {  @Override  **public void** onComplete(@NonNull Task<AuthResult> task) {  **if** (task.isSuccessful()) {  FirebaseUser user = **nAuth**.getCurrentUser();  startActivity(**new** Intent(MainActivity.**this**,PantallaPrincipal.**class**));  finish();  } **else** {  System.***out***.println(**"error"**);  *//MainActivity.this.updateUI(null);* }  }  });  }  @Override  **protected void** onPause() {  **super**.onPause();  *// Capture the current video position and pause the video.* **mCurrentVideoPosition** = **mMediaPlayer**.getCurrentPosition();  **videoBG**.pause();  }   @Override  **protected void** onResume() {  **super**.onResume();  *// Restart the video when resuming the Activity* **videoBG**.start();  }   @Override  **protected void** onDestroy() {  **super**.onDestroy();  *// When the Activity is destroyed, release our MediaPlayer and set it to null.* **mMediaPlayer**.release();  **mMediaPlayer** = **null**;  }   @Override  **public boolean** onSupportNavigateUp() {  onBackPressed();  **return super**.onSupportNavigateUp();  }   @Override  **protected void** onStart() {  **super**.onStart();  *//si el usuario ha iniciado sesion* FirebaseUser currentUser = **nAuth**.getCurrentUser();  **if**(currentUser!=**null**){  startActivity(**new** Intent(getApplicationContext(), PantallaPrincipal.**class**));   }  }    } |
| **XML** |
| <**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:orientation="vertical"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"**>   <**VideoView  android:id="@+id/videoView"  android:layout\_width="0dp"  android:layout\_height="0dp"  app:layout\_constraintBottom\_toBottomOf="parent"  app:layout\_constraintHorizontal\_bias="0.0"  app:layout\_constraintLeft\_toLeftOf="parent"  app:layout\_constraintRight\_toRightOf="parent"  app:layout\_constraintTop\_toTopOf="parent"  app:layout\_constraintVertical\_bias="0.0"** />   <**ImageView  android:id="@+id/imageView"  android:layout\_width="337dp"  android:layout\_height="328dp"  android:layout\_marginTop="8dp"  android:layout\_marginBottom="212dp"  android:contentDescription="AppLogo"  app:layout\_constraintBottom\_toBottomOf="@+id/videoView"  app:layout\_constraintLeft\_toLeftOf="parent"  app:layout\_constraintRight\_toRightOf="parent"  app:layout\_constraintTop\_toTopOf="parent"  app:layout\_constraintVertical\_bias="0.0"  app:srcCompat="@drawable/logo"** />   <**LinearLayout  android:id="@+id/input\_group"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_alignParentBottom="true"  android:layout\_marginLeft="25dp"  android:layout\_marginRight="25dp"  android:layout\_marginBottom="16dp"  android:gravity="center"  android:orientation="vertical"  app:layout\_constraintBottom\_toBottomOf="parent"  app:layout\_constraintHorizontal\_bias="1.0"  app:layout\_constraintLeft\_toLeftOf="parent"  app:layout\_constraintRight\_toRightOf="parent"**>   <**EditText  android:id="@+id/txtUsuario"  android:layout\_width="fill\_parent"  android:layout\_height="wrap\_content"  android:background="@drawable/edit\_text\_top"  android:hint="email"  android:inputType="textPersonName"  android:padding="10dp"** />   <**EditText  android:id="@+id/txtPasswd"  android:layout\_width="fill\_parent"  android:layout\_height="wrap\_content"  android:background="@drawable/edit\_text\_bottom"  android:hint="Password"  android:inputType="textPassword"  android:padding="10dp"** />   <**Button  android:id="@+id/btnLogin"  android:layout\_width="150dp"  android:layout\_height="30dp"  android:layout\_gravity="center\_horizontal"  android:layout\_marginTop="80dp"  android:background="@drawable/button"  android:onClick="login"  android:paddingLeft="15dip"  android:paddingRight="15dip"  android:text="Login"  android:textAllCaps="false"  android:textColor="#2196F3"  android:textSize="17sp"** />   <**TextView  android:id="@+id/signIn\_text"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginTop="3dp"  android:fontFamily="sans-serif"  android:padding="10dp"  android:text="Don't have an Account? Sign Up"  android:textAllCaps="true"  android:textColor="#2196F3"  android:textSize="13dp"  android:textStyle="bold"** />   <**ImageView  android:id="@+id/googleicon"  android:layout\_width="60dp"  android:layout\_height="60dp"  android:layout\_marginTop="15dp"  app:srcCompat="@drawable/google"** />  </**LinearLayout**>   </**androidx.constraintlayout.widget.ConstraintLayout**> |
| **Diseño** |
| Ilustración 15 Main Activity |

|  |
| --- |
| **Formulario de Registro** |
| Dentro de este panel se crea el formulario de registro para un nuevo usuario, en el cual se piden variables como nombre, apellido y correo. Estos datos son guardados en la base de datos en la nube de FireBase. |
| **Java** |
| **package** com.example.detecciontanquegas;  **import** androidx.annotation.NonNull; **import** androidx.appcompat.app.AppCompatActivity; **import** androidx.core.app.NotificationCompat;  **import** android.app.NotificationManager; **import** android.app.PendingIntent; **import** android.content.Context; **import** android.content.Intent; **import** android.os.Bundle; **import** android.text.TextUtils; **import** android.util.Log; **import** android.util.Patterns; **import** android.view.View; **import** android.widget.Button; **import** android.widget.EditText; **import** android.widget.ProgressBar; **import** android.widget.Toast;  **import** com.google.android.gms.tasks.OnCompleteListener; **import** com.google.android.gms.tasks.OnFailureListener; **import** com.google.android.gms.tasks.Task; **import** com.google.firebase.auth.AuthResult; **import** com.google.firebase.auth.FirebaseAuth; **import** com.google.firebase.auth.FirebaseUser; **import** com.google.firebase.database.DatabaseReference; **import** com.google.firebase.database.FirebaseDatabase;  **import** java.util.HashMap; **import** java.util.Map; **import** java.util.Objects; **import** java.util.regex.Pattern;  **public class** formulario\_registro **extends** AppCompatActivity {   EditText **editTextNAME**, **editTextEMAIL**, **editTextUSER**, **editTextPASSWORD**;  *//variables de los datos a registrar* String **email**,**nombre**,**password**;  Button **buttonRegistrar**;  *//variable para registro de usuario  //ProgressBar progressBar;* FirebaseAuth **nAuth**;  FirebaseDatabase **nDatabase**;  DatabaseReference **myref**;   @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_formulario\_registro***);    **nAuth** = FirebaseAuth.*getInstance*();  *//progressBar = findViewById(R.id.progressBar);* **nDatabase** = FirebaseDatabase.*getInstance*();  **myref** = **nDatabase**.getReference();   **editTextNAME** = (EditText) findViewById(R.id.***editNombre***);  **editTextEMAIL** = (EditText) findViewById(R.id.***editEmail***);   **editTextPASSWORD** = (EditText) findViewById(R.id.***editClave***);  **buttonRegistrar** = (Button) findViewById(R.id.***btnGrabaregistro***);    **buttonRegistrar**.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  **email** = **editTextEMAIL**.getText().toString();  **password**=**editTextPASSWORD**.getText().toString();  **nombre** = **editTextNAME**.getText().toString();  **if**(TextUtils.*isEmpty*(**email**)){  **editTextEMAIL**.setError(**"SE REQUIERE CORREO ELECTRONICO"**);  **editTextEMAIL**.setFocusable(**true**);  }  **else if** (!Patterns.***EMAIL\_ADDRESS***.matcher(**email**).matches()){  **editTextEMAIL**.setError(**"Email Invalido"**);  **editTextEMAIL**.setFocusable(**true**);  }  **else if**(TextUtils.*isEmpty*(**password**)){  **editTextPASSWORD**.setError(**"SE REQUIERE SU CONTRASEÑA"**);  **editTextPASSWORD**.setFocusable(**true**);  }  **else if**(**password**.length()<6){  **editTextPASSWORD**.setError(**"Se necesita contraseña >= 6 caracteres"**);  **editTextPASSWORD**.setFocusable(**true**);  }  *//progressBar.setVisibility(View.VISIBLE);   //INICIO DE SESION CON FIREBASE* **else**{  registerUser();   }  }  });     }  **private void** registerUser(){  **nAuth**.createUserWithEmailAndPassword(**email**, **password**)  .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* Map<String,Object> map=**new** HashMap<>();  map.put(**"name"**,**nombre** );  map.put(**"email"**,**email**);  map.put(**"password"**,**password**);  String id= Objects.*requireNonNull*(**nAuth**.getCurrentUser()).getUid();  **myref**.child(**"Users"**).child(id).setValue(map).addOnCompleteListener(**new** OnCompleteListener<Void>() {  @Override  **public void** onComplete(@NonNull Task<Void> task2) {  **if** (!task2.isSuccessful()){  Toast.*makeText*(formulario\_registro.**this**,**"ERROR !"**+ task2.getException().getMessage(),Toast.***LENGTH\_SHORT***).show();  }  }   });  FirebaseUser user = **nAuth**.getCurrentUser();  Toast.*makeText*(formulario\_registro.**this**, **"Registro exitoso"**,  Toast.***LENGTH\_SHORT***).show();  startActivity(**new** Intent(formulario\_registro.**this**,PantallaPrincipal.**class**));   finish();  } **else** {  *// If sign in fails, display a message to the user.* Toast.*makeText*(formulario\_registro.**this**, **"Authentication failed."**,  Toast.***LENGTH\_SHORT***).show();   }   *// ...* }  }).addOnFailureListener(**new** OnFailureListener() {  @Override  **public void** onFailure(@NonNull Exception e) {  Toast.*makeText*(formulario\_registro.**this**, **""**+e.getMessage(),  Toast.***LENGTH\_SHORT***).show();  }  });  }  @Override  **public boolean** onSupportNavigateUp() {  **return super**.onSupportNavigateUp();  } } |
| **XML** |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:background="@drawable/bg"**>    <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="vertical"  android:layout\_marginLeft="25dp"  android:layout\_marginRight="25dp"  android:gravity="center"  android:layout\_alignParentTop="true"  android:layout\_above="@id/input\_group"**>  <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:fontFamily="sans-serif"  android:text="SIGN UP"  android:textColor="#FFFFFF"  android:textSize="20sp"**/>  <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:fontFamily="sans-serif"  android:text="Restaurantes online"  android:textColor="#FFFFFF"  android:textSize="16sp"**/>     </**LinearLayout**>    <**LinearLayout  android:id="@+id/input\_group"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="vertical"  android:layout\_marginLeft="25dp"  android:layout\_marginRight="25dp"  android:layout\_marginBottom="50dp"  android:layout\_alignParentBottom="true"  android:gravity="center"**>   <**EditText  android:id="@+id/editNombre"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:background="@drawable/edit\_text\_top"  android:hint="Nombre"  android:inputType="textPersonName"  android:padding="15dp"** />   <**EditText  android:id="@+id/editEmail"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:background="@drawable/edit\_text\_middle"  android:hint="Email"  android:inputType="textEmailAddress"  android:padding="15dp"** />   <**EditText  android:id="@+id/editClave"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:background="@drawable/edit\_text\_bottom"  android:hint="Password"  android:inputType="textPassword"  android:padding="15dp"**/>    <**Button  android:layout\_width="match\_parent"  android:id="@+id/btnGrabaregistro"  android:layout\_height="wrap\_content"  android:layout\_marginTop="10dp"  android:background="@drawable/button"  android:foreground="?android:attr/selectableItemBackground"  android:text="SIGN UP"  android:textAllCaps="false"  android:textColor="#2196F3"  android:textSize="17sp"**/>   <**TextView  android:id="@+id/signIn\_text"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginTop="3dp"  android:fontFamily="sans-serif"  android:padding="10dp"  android:text="Already Have an Account? Sign in"  android:textAllCaps="true"   android:textColor="#FAD0E1"  android:textSize="13dp"  android:textStyle="bold"** />   android:textSize="13dp"  android:textStyle="bold"  android:textColor="#B2FAD0D0"  android:fontFamily="sans-serif"/>   </**LinearLayout**>  </**RelativeLayout**> |
| **Diseño** |
| Ilustración 16 Pantalla de Registro |

|  |
| --- |
| **Pantalla Principal** |
| Después de iniciar sesión dentro de la aplicación es presentado el panel de usuario de fondo blanco con 4 botones en la base de la pantalla, donde se podrá seleccionar entre 4 opciones:   * Información de Usuario * Estado de los Tanques * Mapa de contactos * Estadísticas |
| **Java** |
| **package** com.example.detecciontanquegas;  **import** androidx.annotation.NonNull; **import** androidx.appcompat.app.AppCompatActivity; **import** androidx.fragment.app.Fragment;  **import** android.app.NotificationChannel; **import** android.app.NotificationManager; **import** android.content.Intent; **import** android.os.Build; **import** android.os.Bundle; **import** android.view.MenuItem;  **import** com.example.detecciontanquegas.aplicacion.CellphoneFrag; **import** com.example.detecciontanquegas.aplicacion.GasFrag; **import** com.example.detecciontanquegas.aplicacion.StaticFrag; **import** com.example.detecciontanquegas.aplicacion.UserFrag; **import** com.google.android.material.bottomnavigation.BottomNavigationView; **import** com.google.firebase.auth.FirebaseAuth; **import** com.google.firebase.auth.FirebaseUser; **import** com.google.firebase.database.FirebaseDatabase;  **public class** PantallaPrincipal **extends** AppCompatActivity {  FirebaseAuth **firebaseAuth**;  FirebaseUser **user**;  @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_pantalla\_principal***);  BottomNavigationView bottomNav = findViewById(R.id.***bottom\_navigation***);  bottomNav.setOnNavigationItemSelectedListener(**navlistener**);  **firebaseAuth** = FirebaseAuth.*getInstance*();   **if** (Build.VERSION.***SDK\_INT*** >= Build.VERSION\_CODES.***O***) {  **int** importance = NotificationManager.***IMPORTANCE\_DEFAULT***;  NotificationChannel channel = **new** NotificationChannel(**"10"**, **"Nuevo estado"**, importance);  channel.setDescription(**"Alertas de nuevo dato"**);  NotificationManager notificationManager = getSystemService(NotificationManager.**class**);  notificationManager.createNotificationChannel(channel);  }  }  **private** BottomNavigationView.OnNavigationItemSelectedListener **navlistener** =  **new** BottomNavigationView.OnNavigationItemSelectedListener() {  @Override  **public boolean** onNavigationItemSelected(@NonNull MenuItem menuItem) {  Fragment selectedfragmente = **new** UserFrag(**user**);*// para que empiece en USER* **switch** (menuItem.getItemId()){  **case** R.id.***nav\_user***:  selectedfragmente = **new** UserFrag(**user**);  **break**;  **case** R.id.***nav\_gas***:  selectedfragmente = **new** GasFrag(**user**, getBaseContext());    **break**;  **case** R.id.***nav\_estadistic***:  selectedfragmente = **new** StaticFrag();  **break**;  **case** R.id.***nav\_phone***:  selectedfragmente = **new** CellphoneFrag();  **break**;  }  getSupportFragmentManager().  beginTransaction().  replace(R.id.***fragment\_container***,selectedfragmente).commit();  **return true**;  }  };  **private void** checkUserStatus(){  **user**= **firebaseAuth**.getCurrentUser();  **if** (**user**!= **null**){  *// el usuario ha iniciado sesion* }  **else**{  *// no ha iniciado sesion* startActivity(**new** Intent(PantallaPrincipal.**this**,MainActivity.**class**));  finish();  }  }   @Override  **protected void** onStart() { *// chequea el inicio de sesion* checkUserStatus();  **super**.onStart();  } } |
| **XML** |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**RelativeLayout 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"  tools:context=".PantallaPrincipal"**>  <**FrameLayout  android:id="@+id/fragment\_container"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_above="@+id/bottom\_navigation"**/>  <**com.google.android.material.bottomnavigation.BottomNavigationView  android:id="@+id/bottom\_navigation"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_alignParentBottom="true"  app:menu="@menu/bottom\_navigation"  android:background="?android:attr/windowBackground"  app:itemIconTint="@color/colorBotttomView"** /> |
| **Diseño** |
| Ilustración 17 Pantalla Principal |

|  |
| --- |
| **Información de usuario – fragment\_user.xml** |
| Se presenta la información del usuario que inicio sesión, se muestra una pantalla acorde a la aplicación y el correo del usuario ingresado, y la opción LogOut |
| **XML** |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:background="@drawable/kitchen"** >    <**TextView  android:id="@+id/bienvenida"  style="@android:style/Widget.Holo.Light.TextView"  android:layout\_width="261dp"  android:layout\_height="51dp"  android:layout\_alignParentStart="true"  android:layout\_alignParentTop="true"  android:layout\_marginStart="52dp"  android:layout\_marginTop="198dp"  android:gravity="center"  android:text="BIENVENIDO"  android:textAlignment="center"  android:textAllCaps="false"  android:textAppearance="@style/TextAppearance.AppCompat.Display2"  android:textColor="#3F4A8A"  android:textColorLink="#3F51B5"  android:textSize="25dp"  android:textStyle="normal"** />   <**TextView  android:id="@+id/username"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_alignParentStart="true"  android:layout\_alignParentTop="true"   android:layout\_marginStart="82dp"  android:layout\_marginTop="253dp"  android:text="user fragment"  android:textAlignment="center"  android:textColor="#3F51B5"  android:textColorLink="#FF5722"  android:textSize="20dp"** />   <**pl.droidsonroids.gif.GifImageView  android:layout\_width="162dp"  android:layout\_height="128dp"  android:layout\_alignParentTop="true"  android:layout\_alignParentEnd="true"  android:layout\_marginTop="396dp"  android:layout\_marginEnd="94dp"  android:scaleType="centerInside"  android:src="@drawable/cocinagif"** />   <**Button  android:id="@+id/logout\_b"  android:layout\_width="239dp"  android:layout\_height="60dp"  android:layout\_alignStart="@+id/bienvenida"  android:layout\_alignParentTop="true"  android:layout\_marginStart="73dp"  android:layout\_marginTop="546dp"  android:layout\_marginEnd="-256dp"  android:layout\_marginBottom="456dp"  android:layout\_toStartOf="@+id/username"  android:text="log out"** /> </**RelativeLayout**> |
| **Diseño** |
| Ilustración 18 Pantalla de Usuario |

|  |
| --- |
| **Estados de los Tanques – fragment\_gas.xml** |
| Este panel se muestran todos los tanques de gas que posee el usuario en su cocina conectado al sistema, y estos se van a presentar de 2 formas como un tanque color rojo cuando este VACIO o no se encuentre el tanque en la estación o de color azul cuando este se encuentre LLENO. |
| **Java** |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  xmlns:tools="http://schemas.android.com/tools"  android:background="@drawable/patallatanque"** >  <**GridView  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:id="@+id/grid\_view"  android:numColumns="2"  android:verticalSpacing="20dp"  android:horizontalSpacing="20dp"  android:padding="40dp"**/> </**RelativeLayout**> |
| **Diseño** |
| Ilustración 19 Visualizacion de los Tanques |

|  |
| --- |
| **Mapa de contacto** |
| En este panel se al ser seleccionado se va a visualizar un mapa de Google Maps donde se van a encontrar seteados las ubicaciones y números de los distribuidores de los tanques de gas. |
| **Java** |
| **package** com.example.detecciontanquegas;  **import** androidx.fragment.app.FragmentActivity;  **import** android.os.Bundle;  **import** com.google.android.gms.maps.CameraUpdateFactory; **import** com.google.android.gms.maps.GoogleMap; **import** com.google.android.gms.maps.OnMapReadyCallback; **import** com.google.android.gms.maps.SupportMapFragment; **import** com.google.android.gms.maps.model.LatLng; **import** com.google.android.gms.maps.model.MarkerOptions;  **public class** MapsActivity **extends** FragmentActivity **implements** OnMapReadyCallback {   **private** GoogleMap **mMap**;   @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_maps***);  *// Obtain the SupportMapFragment and get notified when the map is ready to be used.* SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()  .findFragmentById(R.id.***map***);  mapFragment.getMapAsync(**this**);  }    */\*\*  \* Manipulates the map once available.  \* This callback is triggered when the map is ready to be used.  \* This is where we can add markers or lines, add listeners or move the camera. In this case,  \* we just add a marker near Sydney, Australia.  \* If Google Play services is not installed on the device, the user will be prompted to install  \* it inside the SupportMapFragment. This method will only be triggered once the user has  \* installed Google Play services and returned to the app.  \*/* @Override  **public void** onMapReady(GoogleMap googleMap) {  **mMap** = googleMap;   *// Add a marker in Sydney and move the camera* LatLng sydney = **new** LatLng(-34, 151);  **mMap**.addMarker(**new** MarkerOptions().position(sydney).title(**"Marker in Sydney"**));  **mMap**.moveCamera(CameraUpdateFactory.*newLatLng*(sydney));  } } |
| **XML** |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**fragment xmlns:android="http://schemas.android.com/apk/res/android" xmlns:map="http://schemas.android.com/apk/res-auto" xmlns:tools="http://schemas.android.com/tools" android:id="@+id/map" android:name="com.google.android.gms.maps.SupportMapFragment" android:layout\_width="match\_parent" android:layout\_height="match\_parent" tools:context=".MapsActivity"** /> |
| **Diseño** |
| Ilustración 20 Vizualizacion de Mapa |

|  |
| --- |
| **Estadísticas** |
| Este panel tomas los datos del historial que se encuentra en el FireBase para poder mostrar estadísticas del nivel del tanque en el tiempo que ha estado censando. |
| **XML** |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:background="@drawable/stadistic"** >  <**com.github.mikephil.charting.charts.LineChart  android:id="@+id/chart"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"** /> </**RelativeLayout**> |
| **Diseño** |
| Ilustración 21 Vizualizacion de las Estadisticas |

|  |
| --- |
| **Código Arduino** |
|  |
| int pin = 8;  int minimo = 650;  int maximo = 670;  int contador\_inferior=0;  int contador\_superior=0;  int contador\_bateria=0;  int azul = 4;  int rojo = 5;  float voltaje =0 ;  void setup(){  pinMode(azul,OUTPUT);  pinMode(rojo,OUTPUT);  pinMode(pin, OUTPUT);  Serial.begin(9600);  }  void loop() {  int presion = analogRead(A0);  delay(1000);  contador\_bateria++;  Serial.println(presion);  if (contador\_inferior ==0){  if(presion <= minimo){  enviar();  contador\_inferior=1;  contador\_superior=0;  }  }  if (contador\_superior ==0){  if(presion >= maximo){  enviar();  contador\_inferior=0;  contador\_superior=1;  }  }  if(contador\_bateria>600){ // 600 segundos igual a 10 minutos  voltaje = analogRead(A1);  if(voltaje < 307){  digitalWrite(rojo,HIGH);  digitalWrite(azul,LOW);  }  else{  digitalWrite(azul,HIGH);  digitalWrite(rojo,LOW);  }  contador\_bateria =0; // se vuelve a encerar el contador  }  }  void enviar(){  digitalWrite(pin, HIGH);  delay(3000);  digitalWrite(pin, LOW);  Serial.println("Se activo el ibutton");  } |
|  |

**Bibliografía**

* [1] FireBase – Online : <https://firebase.google.com/docs/database/?hl=es-419>
* [2] Developer – Online <https://developer.android.com/studio>