

**MANUAL TÉCNICO  
  
MATERIA**Aplicaciones Móviles y Servicios Telemáticos  
 **TÍTULO**Clasificador de objetos mediante peso  
 **PROFESORA**Msig. Adriana E. Collaguazo Jaramillo  
 **INTEGRANTES**Renato Andres Jara OrdoñezPierina Paulette Pereira CampoverdeMaria Michelle Alcivar Molina

**Contenido**

[Introducción 3](#_Toc62194093)

[Objetivos 3](#_Toc62194094)

[Objetivo General: 3](#_Toc62194095)

[Objetivos Específicos: 3](#_Toc62194096)

[Solución Planteada 3](#_Toc62194097)

[Componentes de Software 4](#_Toc62194098)

[Componentes de Hardware 4](#_Toc62194099)

[Diagrama del proyecto 4](#_Toc62194100)

[Diagrama del circuito 5](#_Toc62194101)

[Diagrama Entidad-Relación 6](#_Toc62194102)

[Diagrama de casos de uso 7](#_Toc62194103)

[Diagrama de Despliegue 8](#_Toc62194104)

[Diagrama del Árbol 8](#_Toc62194105)

[Set-Up y Código Fuente 9](#_Toc62194106)

[Implementación del Software 9](#_Toc62194107)

[1. Aplicación 9](#_Toc62194108)

[1.1.1. Splash 9](#_Toc62194109)

[1.1.2. Menu 9](#_Toc62194110)

[1.1.3. Menú 16](#_Toc62194111)

[1.1.4. Tanque de oxigeno 18](#_Toc62194112)

[2. Código Arduino 21](#_Toc62194113)

[Implementación física 22](#_Toc62194114)

[Análisis de presupuesto 23](#_Toc62194115)

[Conclusiones 23](#_Toc62194116)

# 

# Introducción

La aparición de la pandemia en el mundo ha sacado a la vista múltiples problemas de en el sistema de salud, entre ellos la falta de personal, por esta razón buscar formas de reducir la carga laborar es primordial en estos tiempos, por ello se busca una solución a la gran cantidad de tanques de oxigeno que existe en los hospitales en cuidado intensivos, estos tanques deben ser llenados inmediatamente al momento que se agote ya que de ellos depende la vida de muchas personas, el personal medico debe estar constantemente revisando que los tanques estén llenos y llenarlos cuando se estén agotando, se busca entonces una solución más efectiva para conocer exactamente cuando se ha terminado el tanque de oxígeno.

# Objetivos

## Objetivo General:

* Garantizar el riego adecuado de los tanques de oxígeno en el área e cuidados intensivos de un hospital.

## Objetivos Específicos:

* Utilizar celdas de carga para hacer una balanza digital que monitoree todo el tiempo el peso del tanque.
* Crear un prototipo que genere la menor cantidad de ruido y sea de bajo costo.
* Crear una aplicación móvil que permita al usuario poder acceder en tiempo real a la información del tanque de oxigeno

# Solución Planteada

Se propone implementar un sistema de control de peso con una balanza digital elaborada con una celda de carga, al momento de ejercer presión sobre la celda se altera la resistencia de la celda, con estos cambios de resistencia se elabora un modelo para detectar variaciones y asociarlas con perfiles de peso, de esta forma saber cuando enviar los pulsos que se registran en el Backend y son interpretados por la aplicación móvil.

# Componentes de Software

* Android Studio 4.0
* Raspberry operating system
* Firebase

# Componentes de Hardware

* 1 Raspberry PI 4
* 1 Módulo sensor de peso HX711
* 1 sensor de temperatura y humedad relativa en el aire DHT11
* Sensor de peso balanza electrónica 0-5 Kg
* 4 leds
* 1 Interruptor
* 15 jumpers
* 1 Fuente de poder

# Diagrama del proyecto

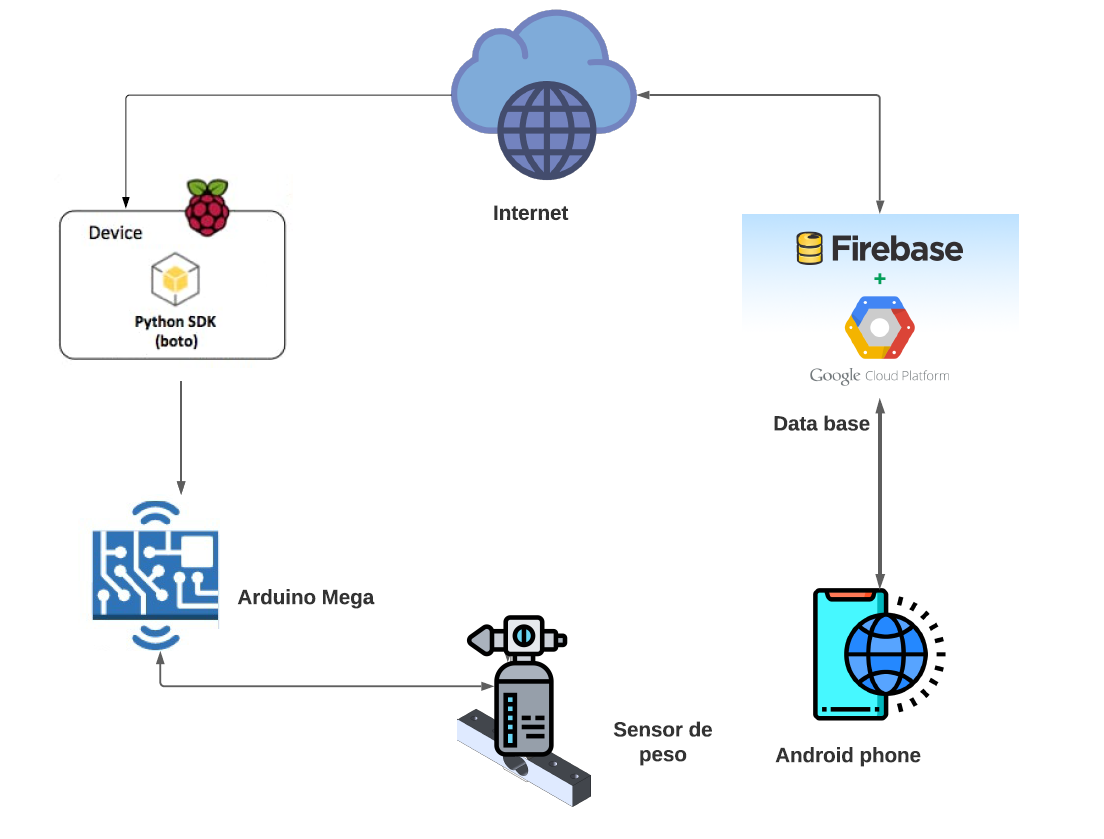


Ilustración 1 Diagrama del proyecto

# agrama del circuito

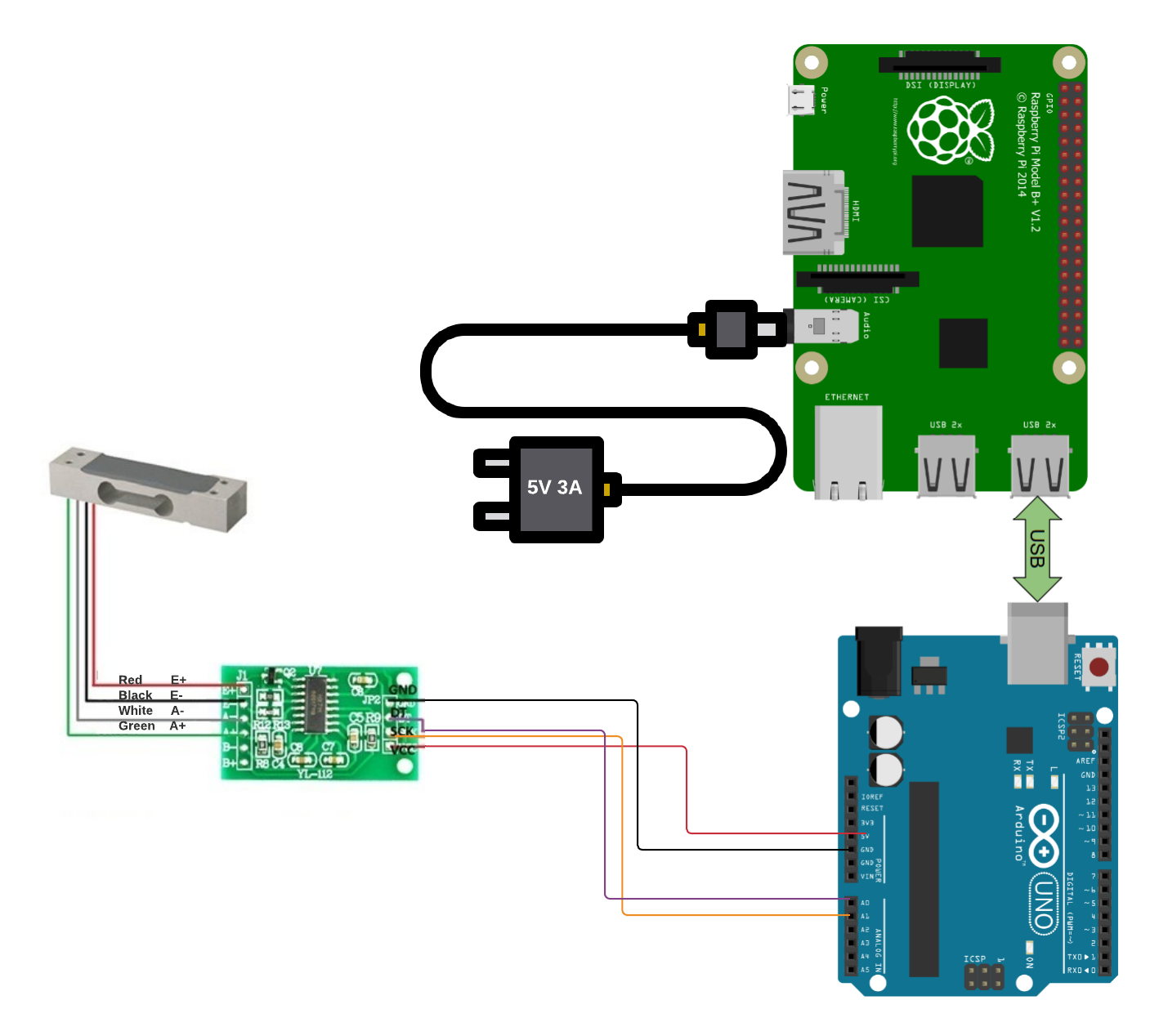


Ilustración 2 Diagrama de circuito del proyecto

# Diagrama Entidad-Relación

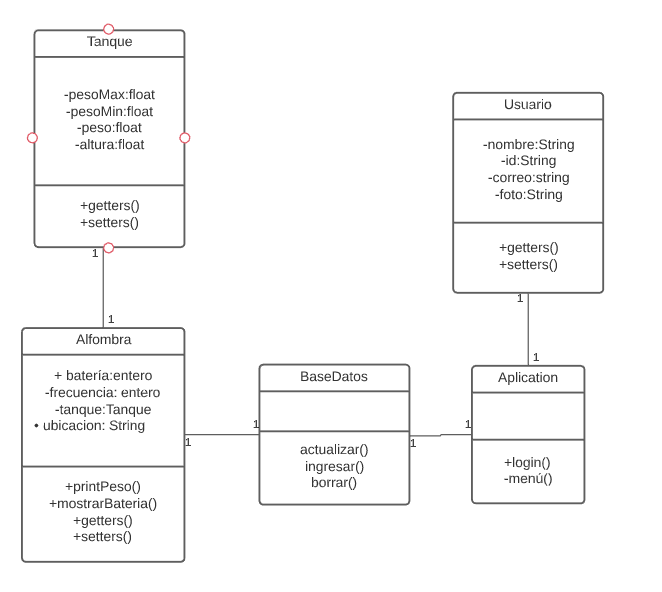


Ilustración 3 Diagrama Entidad-Relación

# Diagrama de casos de uso

Ilustración 4 Diagrama de casos de uso

# Diagrama de Despliegue

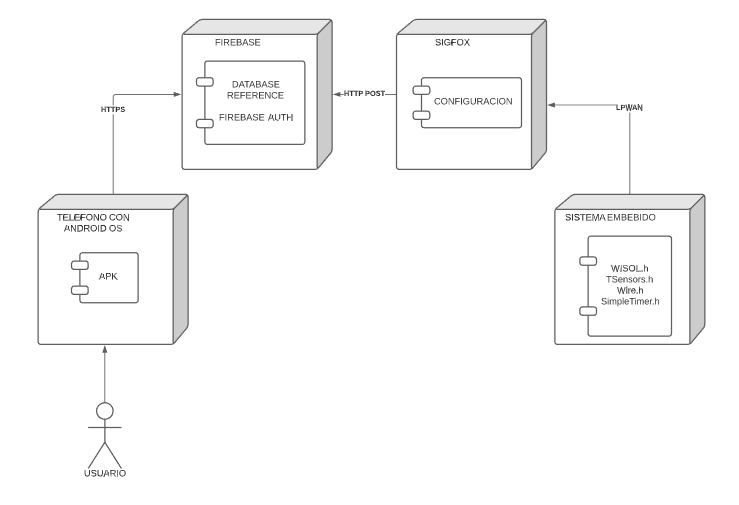


Ilustración 5 Diagrama de Despliegue

# Diagrama del Árbol

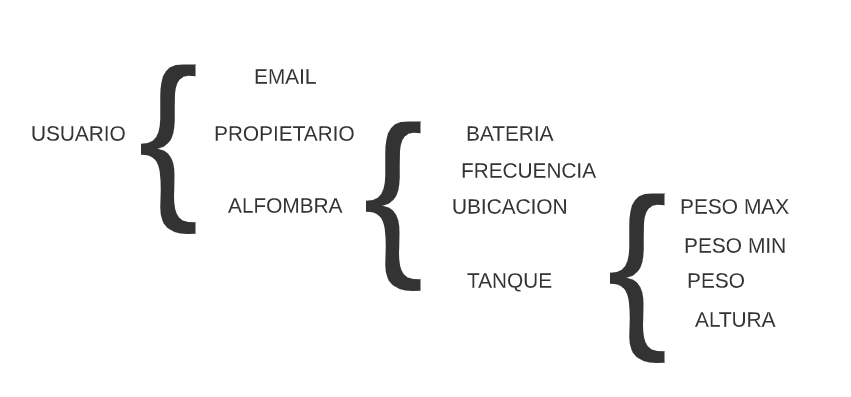


Ilustración 6 Diagrama de arbol

# Set-Up y Código Fuente

# Implementación del Software

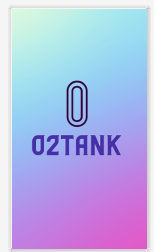
## Aplicación

### Splash

**Java**

**public class** Splash **extends** AppCompatActivity {  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_splash***);  
  
 **new** Handler().postDelayed(**new** Runnable() {  
 @Override  
 **public void** run() {  
 Intent intent = **new** Intent(Splash.**this**, MainActivity.**class**);  
 startActivity(intent);  
 finish();  
 }  
 }, 3000);  
 }  
  
  
}

**XML**

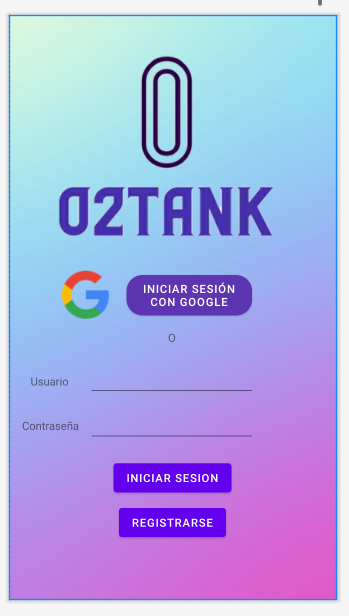


### Menu

**Java**

**public class** MainActivity **extends** AppCompatActivity {  
 **static final int *GOOGLE\_SIGN\_IN*** = 123;  
 FirebaseAuth **mAuth**;  
 GoogleSignInClient **mGoogleSignInClient**;  
 Button **btn\_login**;  
 EditText **editUser**;  
 EditText **editPass**;  
 **static boolean** *tieneAlfombraCG*;  
 **static boolean** *tieneAlfombraSG*;  
 **static boolean** *userPassCorr*;  
 **static** String *alfombra*;  
 **static** String *conGoogle*;  
 DatabaseReference **db\_reference**;  
 **public static** HashMap<String, String> getInfo\_user() {  
 **return** *info\_user*;  
 }  
  
 **public static void** setInfo\_user(HashMap<String, String> info\_user) {  
 MainActivity.*info\_user* = info\_user;  
 }  
  
 **static** HashMap<String, String> *info\_user*;  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
 **editUser** = (EditText) findViewById(R.id.***editUsuario***);  
 **editPass** = (EditText) findViewById(R.id.***editContra***);  
 **db\_reference** = FirebaseDatabase.*getInstance*().getReference();  
 **mAuth** = FirebaseAuth.*getInstance*();  
 GoogleSignInOptions gso = **new** GoogleSignInOptions.Builder(GoogleSignInOptions.***DEFAULT\_SIGN\_IN***)  
 .requestIdToken(getString(R.string.***default\_web\_client\_id***))  
 .requestEmail()  
 .build();  
 **mGoogleSignInClient** = GoogleSignIn.*getClient*(**this**, gso);  
  
 Intent intent = getIntent();  
 String msg = intent.getStringExtra(**"msg"**);  
 **if**(msg != **null**){  
 **if**(msg.equals(**"cerrarSesion"**)){  
 cerrarSesion();  
 }  
 }  
 }  
  
 **public void** iniciarSesion(View view) {  
 Intent signInIntent = **mGoogleSignInClient**.getSignInIntent();  
 startActivityForResult(signInIntent, ***GOOGLE\_SIGN\_IN***);  
 }  
  
 **public void** iniciarSesionSinGoogle(View view){  
  
 **if**(!(*isOnline*(getApplicationContext()))){  
 Toast.*makeText*(**this**, **"NO HAY CONEXION A INTERNET"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 **else** {  
 *userPassCorr* = **false**;  
 **if** (**""**.equals(**editUser**.getText().toString()) || **""**.equals(**editPass**.getText().toString())) {  
 Toast.*makeText*(MainActivity.**this**, **"POR FAVOR, INGRESE SU USUARIO Y CONTRASENA"**, Toast.***LENGTH\_SHORT***).show();  
 } **else** {  
 FirebaseDatabase.*getInstance*().getReference().child(**"Usuarios"**).addListenerForSingleValueEvent(**new** ValueEventListener() {  
 @Override  
 **public void** onDataChange(DataSnapshot dataSnapshot) {  
 **for** (DataSnapshot snapshot : dataSnapshot.getChildren()) {  
 String user = snapshot.getKey();  
 String pass = snapshot.getValue().toString();  
 **if** (user.equals(**editUser**.getText().toString()) && pass.equals(**editPass**.getText().toString())) {  
 *userPassCorr* = **true**;  
 }  
 }  
  
 **if** (*userPassCorr* == **true**) {  
 System.***out***.println(**"ANTES"**);  
 System.***out***.println(*tieneAlfombraSG*);  
 tieneAlfombraPorUsuarioSG(**editUser**.getText().toString());  
 System.***out***.println(**"AL FINAL"**);  
 System.***out***.println(*tieneAlfombraSG*);  
  
  
 }  
 **if** (*userPassCorr* == **false**) {  
 Toast.*makeText*(MainActivity.**this**, **"POR FAVOR REVISE QUE SUS CREDENCIALES SEAN CORRECTAS"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 }  
  
 @Override  
 **public void** onCancelled(DatabaseError databaseError) {  
  
 }  
 });  
 }  
 }  
 }  
  
 **protected void** onActivityResult(**int** requestCode, **int** resultCode, Intent data) {  
 System.***out***.println(**"corriendo onActivity Result"**);  
 **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) {  
 System.***out***.println(**"Corriendo firebase Auth"**);  
 Log.*d*(**"TAG"**, **"firebaseAuthWithGoogle:"** + acct.getId());  
 AuthCredential credential = GoogleAuthProvider.*getCredential*(acct.getIdToken(), **null**);  
 **mAuth**.signInWithCredential(credential)  
 .addOnCompleteListener(**this**, task -> {  
 **if** (task.isSuccessful()) {  
 FirebaseUser user = **mAuth**.getCurrentUser();  
 updateUI(user);  
 } **else** {  
 System.***out***.println(**"error"**);  
 updateUI(**null**);  
 }  
 });  
 }  
  
 **private void** cerrarSesion() {  
 **mGoogleSignInClient**.signOut().addOnCompleteListener(**this**,  
 task -> updateUI(**null**));  
 }  
  
 **private void** updateUI(FirebaseUser user) {  
 **if** (user != **null**) {  
 *conGoogle*=**"SI"**;  
 finish();  
 tieneAlfombraPorUsuarioCG( user);  
  
 } **else** {  
 System.***out***.println(**"sin registrarse"**);  
 *conGoogle*=**"NO"**;  
 }  
 }  
  
 **private void** tieneAlfombraPorUsuarioSG(String username){  
 **db\_reference** = FirebaseDatabase.*getInstance*().getReference();  
 *tieneAlfombraSG*= **false**;  
 *alfombra*=**""**;  
 FirebaseDatabase.*getInstance*().getReference().addListenerForSingleValueEvent(**new** ValueEventListener() {  
 @Override  
 **public void** onDataChange(DataSnapshot dataSnapshot) {  
 **for** (DataSnapshot snapshot : dataSnapshot.getChildren()) {  
 String rama = snapshot.getKey();  
 System.***out***.println(rama);  
 **if**(rama.startsWith(**"alfombra"**)){  
 String propietario = snapshot.child(**"propietario"**).getValue().toString();  
 **if**(**editUser**.getText().toString().equals(propietario)){  
 *tieneAlfombraSG*=**true**;  
 System.***out***.println(**"SI TIENE ALFOMBRA"**);  
 *alfombra* = rama;  
 }  
 }  
  
 }  
 **if**(*tieneAlfombraSG*==**true**){  
 Toast.*makeText*(MainActivity.**this**, **"BIENVENIDO"**, Toast.***LENGTH\_SHORT***).show();  
 *info\_user* = **new** HashMap<String, String>();  
 *info\_user*.put(**"user\_name"**, **editUser**.getText().toString());  
 *info\_user*.put(**"conGoogle"**, *conGoogle*);  
 *info\_user*.put(**"alfombraID"**,*alfombra*);  
 Intent intent2 = **new** Intent(getBaseContext(), Menu.**class**);  
 intent2.putExtra(**"info\_user"**, *info\_user*);  
 startActivity(intent2);  
 }  
 **if**(*tieneAlfombraSG*==**false**){  
 Toast.*makeText*(MainActivity.**this**, **"NO TIENE ALFOMBRAS ASOCIADAS A ESTA CUENTA"**, Toast.***LENGTH\_SHORT***).show();  
 Intent intent = **new** Intent(Intent.***ACTION\_MAIN***);  
 intent.addCategory(Intent.***CATEGORY\_HOME***);  
 intent.setFlags(Intent.***FLAG\_ACTIVITY\_NEW\_TASK***);  
 startActivity(intent);  
  
 }  
 }  
  
 @Override  
 **public void** onCancelled(DatabaseError databaseError) {  
  
 }  
 });  
 System.***out***.println(**"FINALIZANDO TODA LA FUNCION"**);  
 System.***out***.println(*tieneAlfombraSG*);  
 }  
  
 **public static boolean** isOnline(Context context) {  
  
 ConnectivityManager connectivityManager = (ConnectivityManager) context.getSystemService(Context.***CONNECTIVITY\_SERVICE***);  
  
 **if** (connectivityManager != **null**) {  
 NetworkCapabilities capabilities = **null**;  
 **if** (Build.VERSION.***SDK\_INT*** >= Build.VERSION\_CODES.***LOLLIPOP***) {  
 **if** (Build.VERSION.***SDK\_INT*** >= Build.VERSION\_CODES.***M***) {  
 capabilities = connectivityManager.getNetworkCapabilities(connectivityManager.getActiveNetwork());  
 }  
 }  
 **if** (capabilities != **null**) {  
 **if** (Build.VERSION.***SDK\_INT*** >= Build.VERSION\_CODES.***LOLLIPOP***) {  
 **if** (capabilities.hasTransport(NetworkCapabilities.***TRANSPORT\_CELLULAR***)) {  
 *// Datos* **return true**;  
 }  
 }  
 **if** (Build.VERSION.***SDK\_INT*** >= Build.VERSION\_CODES.***LOLLIPOP***) {  
 **if** (capabilities.hasTransport(NetworkCapabilities.***TRANSPORT\_WIFI***)) {  
 *// WiFi* **return true**;  
 }  
 }  
 }  
 }  
  
 **return false**;  
 }  
  
 **private void** tieneAlfombraPorUsuarioCG(FirebaseUser user){  
 *tieneAlfombraCG*= **false**;  
 *alfombra*=**""**;  
 FirebaseDatabase.*getInstance*().getReference().addListenerForSingleValueEvent(**new** ValueEventListener() {  
 @Override **public void** onDataChange(DataSnapshot dataSnapshot) {  
 **for** (DataSnapshot snapshot : dataSnapshot.getChildren()) {  
 String rama = snapshot.getKey();  
 System.***out***.println(rama);  
 **if**(rama.startsWith(**"alfombra"**)){  
 String propietario = snapshot.child(**"propietario"**).getValue().toString();  
 System.***out***.println(propietario);  
 System.***out***.println(user.getEmail());  
 **if**(user.getEmail().equals(propietario)){  
 *tieneAlfombraCG*=**true**;  
 System.***out***.println(**"SI TIENE ALFOMBRA"**);  
 *alfombra* = rama;  
 }  
 }  
  
 }  
 **if**(*tieneAlfombraCG*==**true**){  
 *info\_user* = **new** HashMap<String, String>();  
 *info\_user*.put(**"user\_name"**, user.getEmail());  
 *info\_user*.put(**"user\_photo"**, String.*valueOf*(user.getPhotoUrl()));  
 *info\_user*.put(**"conGoogle"**, *conGoogle*);  
 *info\_user*.put(**"alfombraID"**,*alfombra*);  
 Intent intent = **new** Intent(getApplicationContext(), Menu.**class**);  
 intent.putExtra(**"info\_user"**, *info\_user*);  
 startActivity(intent);  
 }  
 **else**{  
 Toast.*makeText*(getBaseContext(), **"NO TIENE ALFOMBRA ASOCIADA CON ESTA CUENTA"**, Toast.***LENGTH\_SHORT***).show();  
 cerrarSesion();  
 Intent intent = **new** Intent(Intent.***ACTION\_MAIN***);  
 intent.addCategory(Intent.***CATEGORY\_HOME***);  
 intent.setFlags(Intent.***FLAG\_ACTIVITY\_NEW\_TASK***);  
 startActivity(intent);  
  
 }  
 }  
 @Override **public void** onCancelled(DatabaseError databaseError) {  
  
 } });  
 }  
  
 **public void** registrarse(View view){  
  
 FirebaseDatabase.*getInstance*().getReference().child(**"Usuarios"**).addListenerForSingleValueEvent(**new** ValueEventListener() {  
 @Override  
 **public void** onDataChange(DataSnapshot dataSnapshot) {  
 **boolean** existe=**false**;  
 **for** (DataSnapshot snapshot : dataSnapshot.getChildren()) {  
 String user = snapshot.getKey();  
 **if** (user.equals(**editUser**.getText().toString())) {  
 existe = **true**;  
 }  
 }  
 **if**(existe==**true**){  
 Toast.*makeText*(MainActivity.**this**, **"NOMBRE DE USUARIO NO DISPONIBLE"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 **else**{  
 **if**(**editUser**.getText().toString().equals(**""**)||**editPass**.getText().toString().equals(**""**)){  
 Toast.*makeText*(MainActivity.**this**, **"LLene los campos porfavor"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 **else**{  
 Toast.*makeText*(MainActivity.**this**, **"REGISTRADO"**, Toast.***LENGTH\_SHORT***).show();  
 DatabaseReference ref = **db\_reference**.child(**"Usuarios"**);  
 ref.child(**editUser**.getText().toString()).setValue(**editPass**.getText().toString());  
  
 }  
  
  
 }  
  
 }  
  
 @Override  
 **public void** onCancelled(DatabaseError databaseError) {  
  
 }  
 });  
 }

XML



### Menú

**Java**

**public class** Menu **extends** AppCompatActivity {  
 TextView **txt\_name**;  
 ImageView **imv\_photo**;  
 ProgressBar **pb**;  
 String **alfombra**;  
 DatabaseReference **db\_reference**;  
 HashMap <String,String> **info\_user**;  
 String **conGoogle**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_menu***);  
  
  
 **txt\_name** = findViewById(R.id.***textUsername***);  
 **imv\_photo** = findViewById(R.id.***imageFotoPerfil***);  
  
 *//Intent intent = getIntent();* **info\_user** = MainActivity.*getInfo\_user*();  
  
 **txt\_name**.setText(**info\_user**.get(**"user\_name"**));  
 **alfombra**=**info\_user**.get(**"alfombraID"**);  
 **conGoogle** = **info\_user**.get(**"conGoogle"**);  
  
 **db\_reference** = FirebaseDatabase.*getInstance*().getReference().child(**alfombra**);  
  
 **if**(**conGoogle**==**"SI"**) {  
 String photo = **info\_user**.get(**"user\_photo"**);  
 Picasso.*with*(getApplicationContext()).load(photo).into(**imv\_photo**);  
 }  
  
 **pb** = (ProgressBar) findViewById(R.id.***barraBateria***);  
 leerBateria();  
 }  
  
 **public void** irEstado(View view){  
 Intent estado = **new** Intent(getBaseContext(),Estado\_de\_tanque.**class**);  
 estado.putExtra(**"info\_user"**, **info\_user**);  
 startActivity(estado);  
 }  
  
 **public void** cerrarSesion(View view){  
 FirebaseAuth.*getInstance*().signOut();  
 finish();  
 Intent intent = **new** Intent(**this**, MainActivity.**class**);  
 intent.putExtra(**"msg"**, **"cerrarSesion"**);  
 startActivity(intent);  
 }  
  
 **private void** leerBateria() {  
 **db\_reference**.addValueEventListener(**new** ValueEventListener() {  
 @RequiresApi(api = Build.VERSION\_CODES.***LOLLIPOP***)  
 @Override  
 **public void** onDataChange(@NonNull DataSnapshot snapshot) {  
 **if**(snapshot.exists()){  
 String bat = snapshot.child(**"bateria"**).getValue().toString();  
 **try** {  
 **int** porc = Integer.*parseInt*(bat);  
 **if**(porc>15 && porc<=100){  
 **pb**.setProgress(porc);  
 **pb**.setProgressDrawable(getDrawable(R.drawable.***pb\_drawable***));  
 }  
 **else**{  
 **pb**.setProgress(porc);  
 **pb**.setProgressDrawable(getDrawable(R.drawable.***pb\_drawable\_2***));  
 }  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 }  
  
 @Override  
 **public void** onCancelled(@NonNull DatabaseError error) {  
  
 }  
 });  
 }  
  
  
}

**XML**



### Tanque de oxigeno

**Java**

**public class** Estado\_de\_tanque **extends** AppCompatActivity {  
 DatabaseReference **db\_reference**;  
 TextView **txtPeso**,**txtMin**,**txtMax**,**txtAltura**;  
 ProgressBar **pbp**;  
 HashMap<String,String> **info\_user**;  
 String **alfombra**;  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_estado\_de\_tanque***);  
 **txtPeso** = (TextView) findViewById(R.id.***txtPorcentaje***);  
 **txtMax**= (TextView) findViewById(R.id.***txtMax***);  
 **txtMin**= (TextView) findViewById(R.id.***txtMin***);  
 **txtAltura**= (TextView) findViewById(R.id.***txtAltura***);  
 **db\_reference** = FirebaseDatabase.*getInstance*().getReference();  
 **pbp** = (ProgressBar)findViewById(R.id.***progressBarP***);  
 **info\_user** = MainActivity.*getInfo\_user*();  
 **alfombra**=**info\_user**.get(**"alfombraID"**);  
 leerPeso();  
 }  
  
  
 **private void** leerPeso() {  
 **db\_reference**.child(**alfombra**).child(**"Tanque"**).addValueEventListener(**new** ValueEventListener() {  
 @RequiresApi(api = Build.VERSION\_CODES.***LOLLIPOP***)  
 @Override  
 **public void** onDataChange(@NonNull DataSnapshot snapshot) {  
 **if**(snapshot.exists()){  
  
 String peso = snapshot.child(**"peso"**).getValue().toString();  
 **txtPeso**.setText(**"Peso: "**+ peso + **" Kg"**);  
 String max = snapshot.child(**"pesoMax"**).getValue().toString();  
 **txtMax**.setText(**"Peso Maximo: "**+ max + **" Kg"**);  
 String min = snapshot.child(**"pesoMin"**).getValue().toString();  
 **txtMin**.setText(**"Peso Minimo: "**+ min + **" Kg"**);  
 String alt = snapshot.child(**"altura"**).getValue().toString();  
 **txtAltura**.setText(**"Altura: "**+ alt+**" m"**);  
 **try** {  
 **float** pesomaxf = Float.*parseFloat*(max);  
 **float** pesominf = Float.*parseFloat*(min);  
 **float** pesof = Float.*parseFloat*(peso);  
 **if**(pesof>=0.0 && pesof<=0.1){  
 Toast.*makeText*(Estado\_de\_tanque.**this**, **"NO HAY NINGUN TANQUE"**, Toast.***LENGTH\_SHORT***).show();  
 }  
  
 **float** pesoNeto = pesomaxf-pesominf;  
 **float** pes = pesof-pesominf;  
  
 **int** porc = Math.*round*((pes\*100.0f)/pesoNeto);  
 **if**(porc>=0 && porc<=15){  
 Toast.*makeText*(Estado\_de\_tanque.**this**, **"EL OXIGENO ESTA POR AGOTARSE"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 System.***out***.println(porc);  
 **if**(porc>15 && porc<=100){  
 **pbp**.setProgress(porc);  
 **pbp**.setProgressDrawable(getDrawable(R.drawable.***pb\_drawable***));  
 }  
 **else**{  
 **pbp**.setProgress(porc);  
 **pbp**.setProgressDrawable(getDrawable(R.drawable.***pb\_drawable\_2***));  
 }  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 }  
  
 @Override  
 **public void** onCancelled(@NonNull DatabaseError error) {  
  
 }  
 });  
 }  
  
 **public void** irMenu(View view){  
 Intent menu = **new** Intent(**this**,Menu.**class**);  
 startActivity(menu);  
 }  
}

**Xml**



# Código Arduino

Para el código en Arduino se importó la librería “HX711h” que permita la lectura de celdas de carga, utilizando esta librería solo es necesario declarar los puertos donde esta conectada la celda y inicializarla con balanza.begin, se debe inicializar el serial también ya que con esto se va a hacer la comunicación con la Raspberry, con balanza.red se lee los datos que entrega la balanza, con balanza.set.scale se calibra la balanza y con balanza.tare se ubica el peso actual.

#include "HX711.h"

const int DOUT=A1;

const int CLK=A0;

HX711 balanza;

void setup() {

Serial.begin(9600);

balanza.begin(DOUT, CLK);

Serial.println(balanza.read());

balanza.set\_scale(439430.25); // Establecemos la escala

balanza.tare(20); //El peso actual es considerado Tara.

}

void loop() {

Serial.println(balanza.get\_units(20),3);

delay(500);

}

# Implementación física

Para la implementación física del proyecto se soldó al Módulo sensor de peso HX711 se tuvo que soldar pines para conectar jumper para así conectarlo a la celda de carga que a su vez va conectada al Arduino, además la Raspberry se conectó con un cable usb.

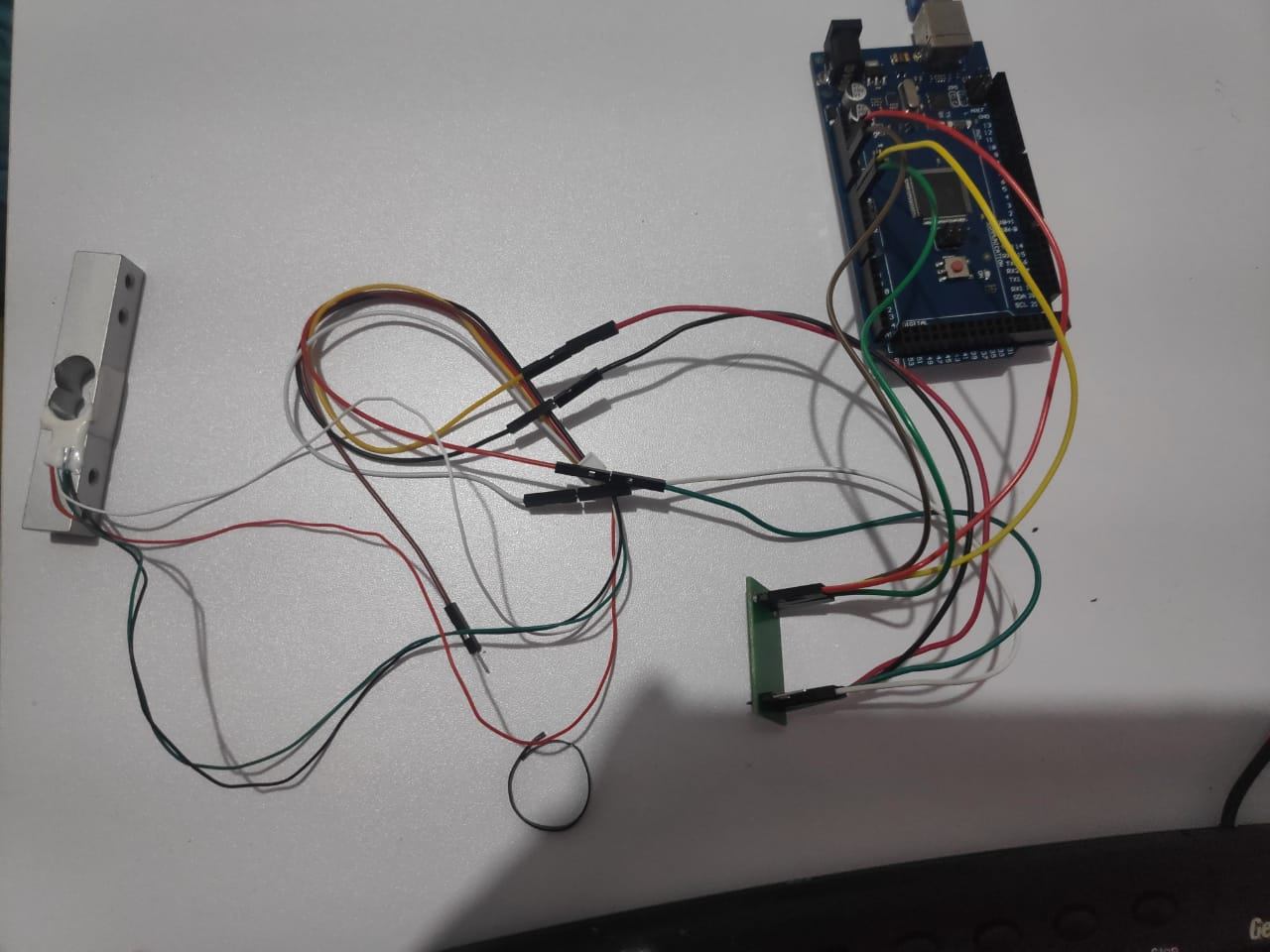


Ilustración 7 Primer prototipo

Luego para mejorar el prototipo se construyó con madera una base para pesar el tanque de oxígeno, el diseño de esta base debe se especial ya que la celda debe estar atornillada a un punto fijo y del otro lado debe estar la base sobre otro punto que puede ser elaborado con un trozo de madera más pequeño, de esta forma habrá una correcta flexión que es como una viga empotrada y al ser flexionada existe una variación de resistencia.

Ilustración 8 Prototipo final

# Análisis de presupuesto

|  |  |  |  |
| --- | --- | --- | --- |
| **Cantidad** | **Material** | **Precio Unidad** | **Precio Final** |
| **1** | Raspberry PI 3 modelo B++ | $ 60 | $ 60 |
| **1** | Celda de carga 5kg+ Modulo sensor de peso | $8.50 | $8.50 |
| **1** | Arduino Mega | $15.00 | $15.00 |
| 4 | Leds | $0.10 | $0.40 |
| 15 | Jumpers | $0.10 | $ 1.50 |
| 1 | Batería recargable | 20.00 | $20.00 |
| 2 | Madera reciclada | $0 | $0 |
|  |  | **TOTAL** | **$ 105.40** |

# Conclusiones

* Con la implementación de este proyecto, se puede concluir que un sistema de peso digital automático permite el monitoreo de tanques de oxígeno de manera fácil y sencilla, lo cual puede llegar a disminuir en gran medida la carga laborar del personal médico.
* La aplicación diseñada brinda la posibilidad al usuario de acceder en tiempo real y desde cualquier lugar, a la información relevante sobre los tanques de oxígeno, dado al uso de una base de datos la cual permite almacenar información, la mismo que será enviada a la aplicación.
* El prototipo creado permite la implementación de varias alfombras para las celdas de carga, ya que al utilizar Arduino mega sobre algunos pines para poder colocar hasta 4 alfombras.

**Recomendaciones**

* Se recomienda el uso de Firebase ya que cuenta con diversos servicios como el inicio de sesión, Realtime Database, Cloud Messsaging, todos estos servicios se introducen fácilmente a la aplicación móvil usando Android studio ya que cuenta con una interfax en tolos que de manera rápida y sencilla conecta estos servicios con la aplicación creada.
* En importante el uso del real-time ya que para aplicaciones como este de tipo Hard es de vital importancia que lo datos se estén actualizando en tiempo real y s tiempo