

MANUAL TÉCNICO GETABED



OCUPACIÓN DE CAMAS EN LA UNIDAD DE CUIDADOS INTENSIVOS

GRUPO #4

- BAILÓN MIGUEL
- CARRILLO MA. EUGENIA
- MENDOZA DAVID

APLICACIONES MÓVILES Y SERVICIOS TELEMÁTICOS



Tabla de contenido

Introducción.....	3
Descripción del sistema implementado	3
Recursos	3
Diagramas	4
Diagrama de proyecto	4
Diagrama de despliegue.....	5
Diagrama de clases	5
Diagrama de modelo de base de datos no relacional	6
Diagrama de Casos de Uso.....	6
Diagrama del circuito	7
Código Fuente	9
Código Fuente Arduino	9
Código_Sensor_Nano.ino.....	9
Conexión_Wifi_Firebase_EnvioDatoa_ESP8266	16
SplashScreen.java (Pantalla de la pantalla de Inicio)	19
activity_splash_screen.xml	19
Cama.java (Data class de objeto Cama)	20
Seccion.java	21
MainActivity.java (Pantalla de login).....	22
activity_ingreso.xml	24
Menu.java (Pantalla de menú)	26
activity_menu.xml	29
EstadoCamas.java (Pantalla donde se muestran el estado de las camas).....	31
activity_estado_camas.xml.....	33
SpinAdapter.java (Seleccion de seccion en Acitivty EstadoCamas)	33
CamasAdapter.java (Creacion de objetos de Grid View donde se muestran las camas)	35
fragemnt_camas.xml (Layout que contiene el Grid View donder se van a mostrar las camas)	36
Cama_desocupada.xml (Layout de cada cama)	36
GraficoDisponibilidad.java (Pantalla con diagrama pastel de disponibilidad de cama)	37
activity_grafico_disponibilidad.xml	38



Introducción

Este manual técnico describe tanto los recursos de hardware como de software utilizados para el correcto desarrollo del proyecto.

El objetivo principal de este proyecto es conocer la disponibilidad de las camas pertenecientes a la unidad de cuidados intensivos. Por lo que con ayuda de Google se accede a la cuneta y se puede visualizar la cantidad de cama ocupadas o desocupadas. En caso de no encantar disponibilidad, se puede añadir en una lista de espera donde le llegará una notificación indicando la disponibilidad de una de las camas de esa área.

Descripción del sistema implementado

El sistema se basa en una aplicación móvil dirigida para el personal médico y enfermeros especialmente del área de cuidados intensivos, ya que actualmente debido a la pandemia es una de las áreas más colapsadas dentro de los hospitales.

El personal médico y enfermeros podrán acceder a la aplicación por medio de su cuenta de Gmail y visualizarán el estado de las camas de las del área mencionada, observaran si estas se encuentran ocupadas o desocupadas y en caso de que se desee, el usuario podrá colocarse en una lista de espera, para que una vez que se haya desocupado una cama, pueda recibir una notificación que le indique que a cama está libre,

Recursos

Hardware	Software	Extras
Arduino Nano	Android studio	Caja
Modulo de Wifi ESP8266-01	Firebase	Cautín
Jumpers		Estaño
Resistencias		Cable arduino
Protoboard		Programador ESP8266-01
Sensor de presión		Cinta
Switch		
Baterias		
Leds		



Cantidad	Nombre	Precio	Proveedor
1	Arduino Nano Atmega328p CH340	\$ 7,50	Mercado libre
1	Módulo WiFi ESP8266	\$ 7,99	Mercado libre
1	Sensor de presión FSR402	\$11,50	Mercado libre
1	Switch	\$ 1,00	Electrónica Castro
1	Baquelita	\$ 1,00	Electrónica Castro
1	Batería	\$ 1,00	Electrónica Castro
1	Conector de batería	\$ 0,25	Electrónica Castro
10	Jumpers Hembra-Macho	\$ 1,00	Electrónica Castro
10	Jumpers Macho-Macho	\$ 1,00	Electrónica Castro
2	Led Verde	\$ 0,20	Electrónica Castro
1	Led Rojo	\$ 0,10	Electrónica Castro
1	Led Amarillo	\$ 0,10	Electrónica Castro
4	Resistencias 1k [ohm]	\$ 0,40	Electrónica Castro
3	Resistencias 220 [ohm]	\$ 0,30	Electrónica Castro
1	Resistencia 470 [ohm]	\$ 0,10	Electrónica Castro
2[m]	Estaño	\$ 1,00	Electrónica Castro
Total		\$34,44	

Diagramas

Diagrama de proyecto

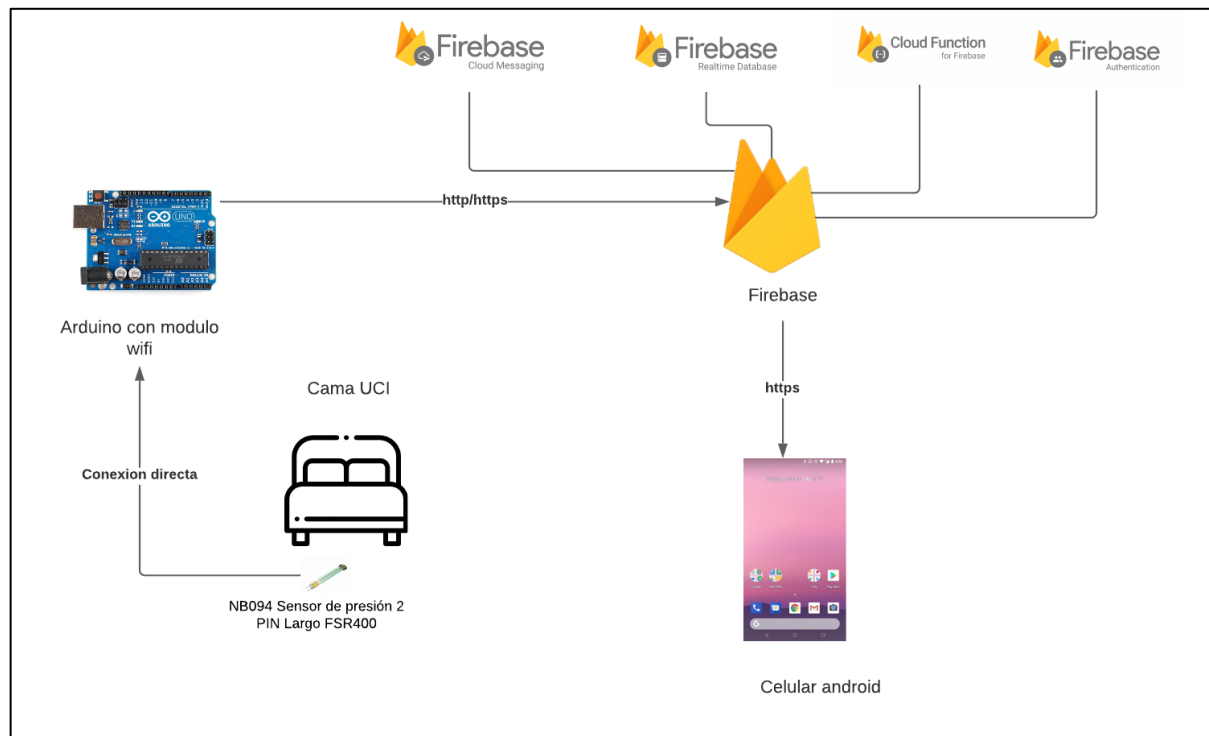


Diagrama de despliegue

La aplicación sensa y envía datos en tiempo real a través del dispositivo IoT que fue desarrollado, donde se define la presión ejercida por el paciente y el porcentaje de batería del dispositivo.

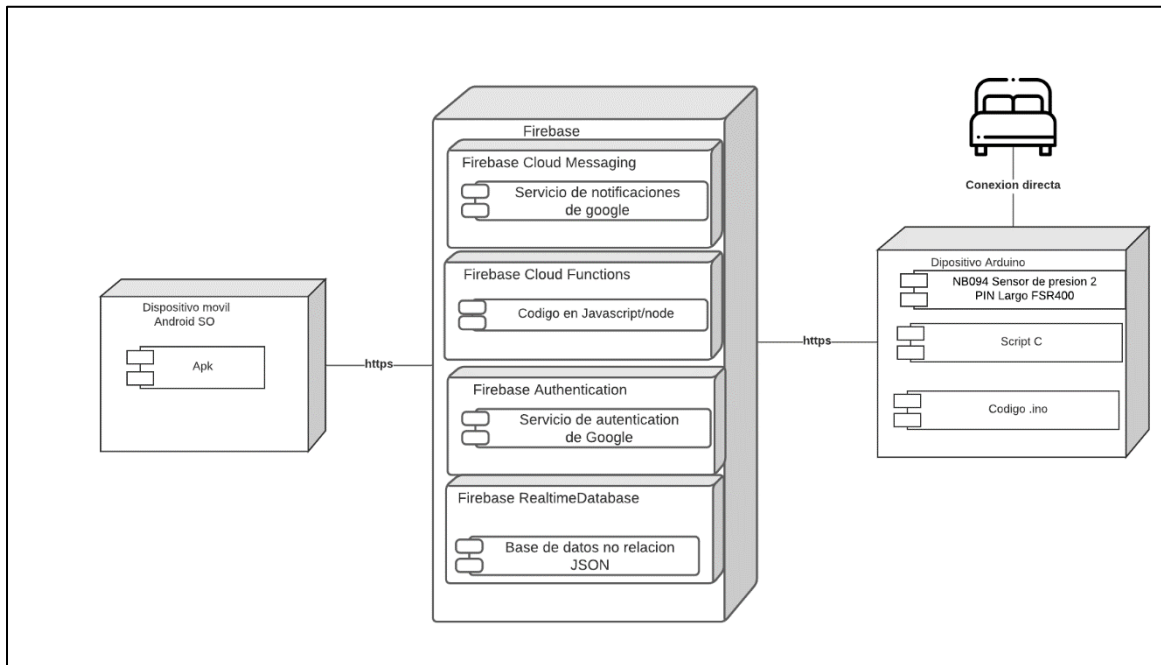


Diagrama de clases

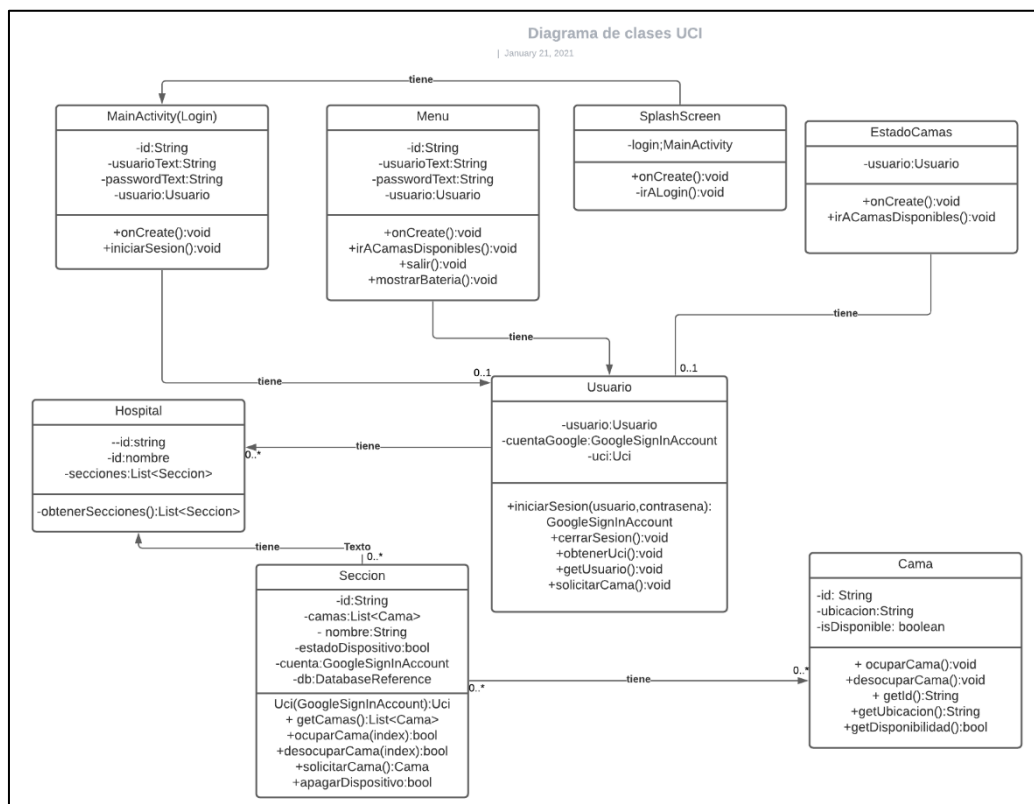


Diagrama de modelo de base de datos no relacional

Se muestra la respectiva base de datos no relacional, la cual representa datos almacenados en bases de datos Firebase.

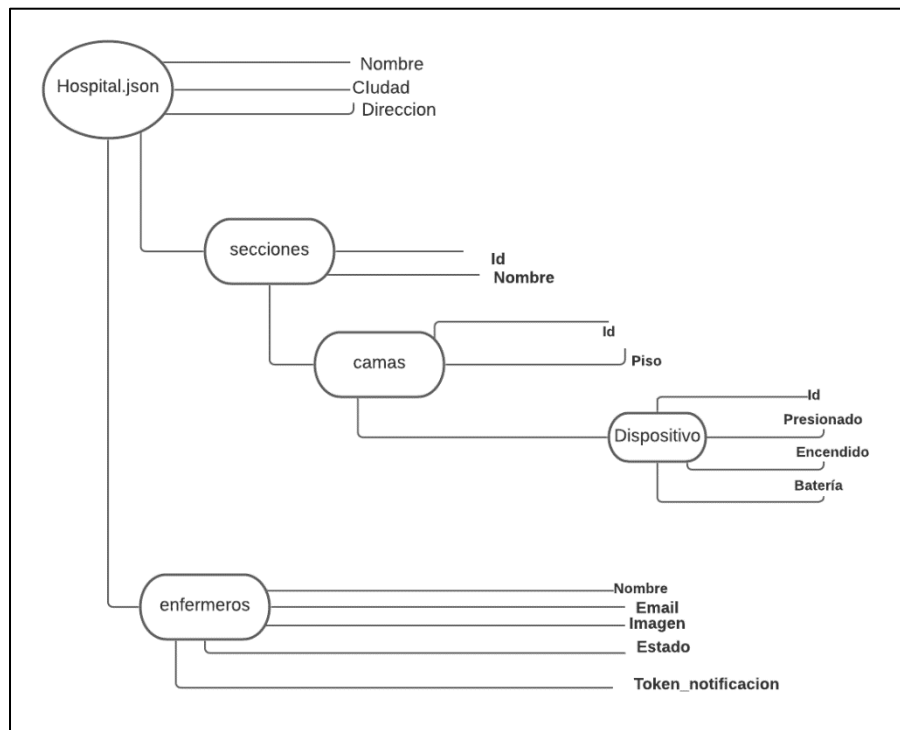


Diagrama de Casos de Uso

Se detalla los diferentes casos de la aplicación desarrollada, lo que esta ofrece y como se relaciona todo el sistema entre si.

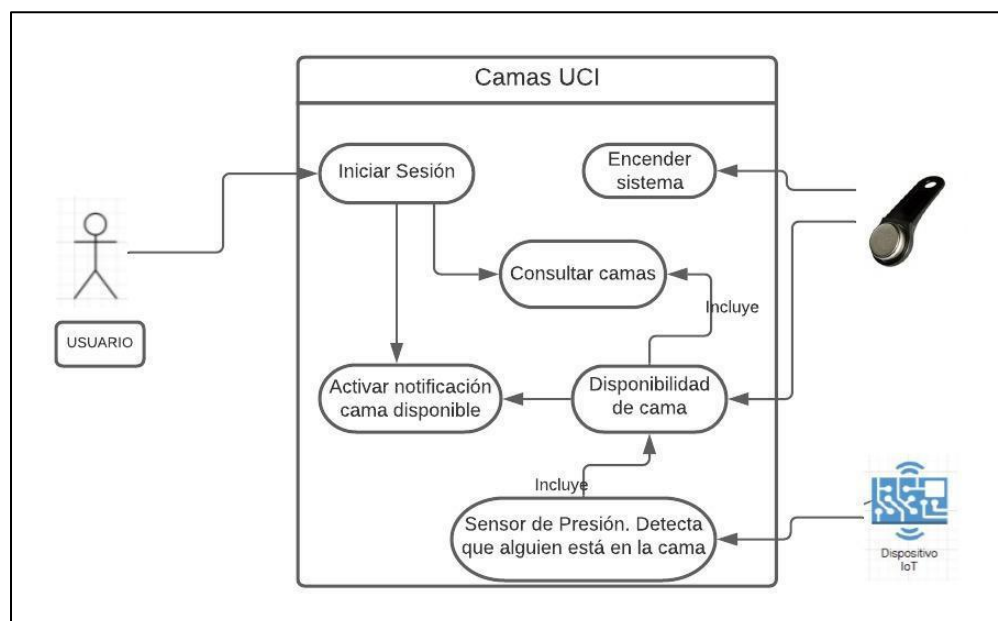


Diagrama del circuito

El circuito propuesto cuenta con un Arduino Nano que es el microcontrolador principal del sistema que a través del módulo WiFi ESP8266 envía la información a la base de datos en tiempo real Firebase. El sistema utiliza un sensor a presión, que se colocará debajo de la almohada del paciente, sentirá la presión ejercida por el mismo y marcará en la aplicación que la cama fue ocupada. Además, consta de luces leds que muestra la cantidad de batería que mantiene el dispositivo IoT y de activación del sistema. El prototipo es capaz de encenderse o apagarse según convenga a través de un switch conectado a la alimentación del sistema (batería). En adición, se utilizaron cables jumpers, macho-macho y macho-hembra, para realizar la conexión entre los componentes del proyecto. Finalmente, las resistencias fueron utilizadas para controlar la corriente que llegaba a los leds. Así mismo se crearon divisores de voltaje para la correcta comunicación serial entre el modulo WiFi y el microcontrolador y también se utilizaron para realizar el monitoreo del nivel de batería.

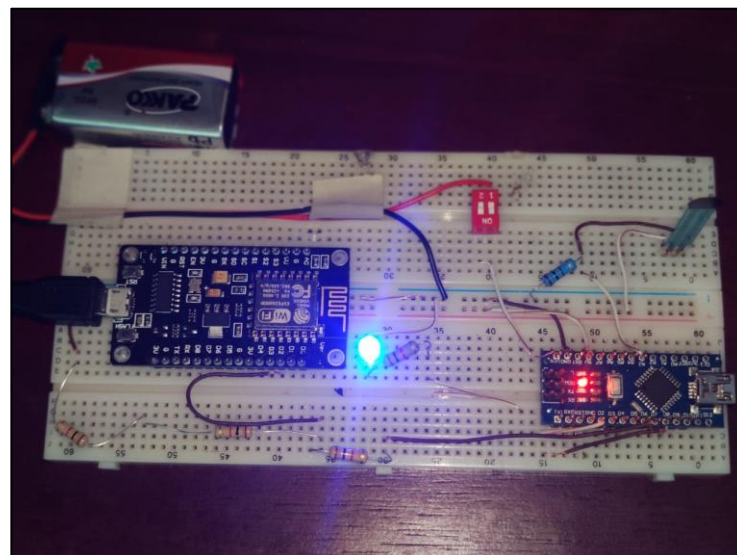
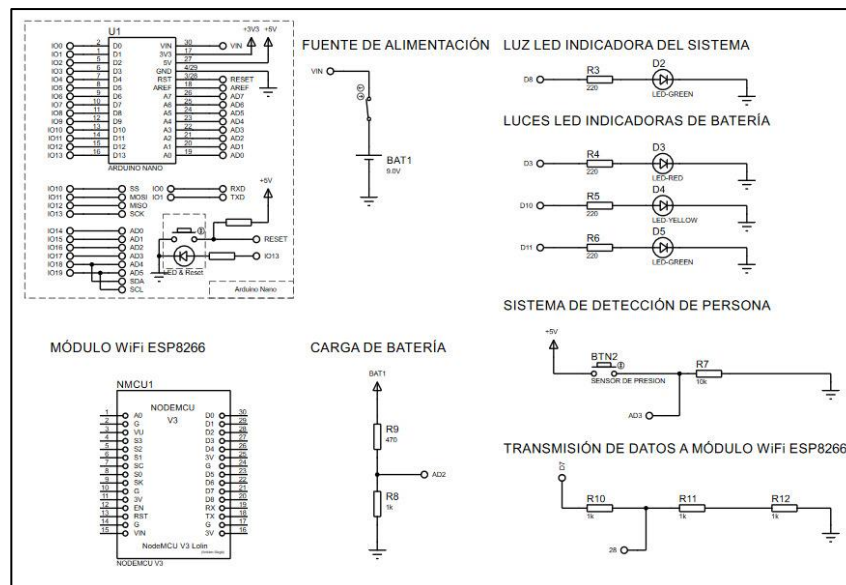


Figura 1 Prototipo del Circuito



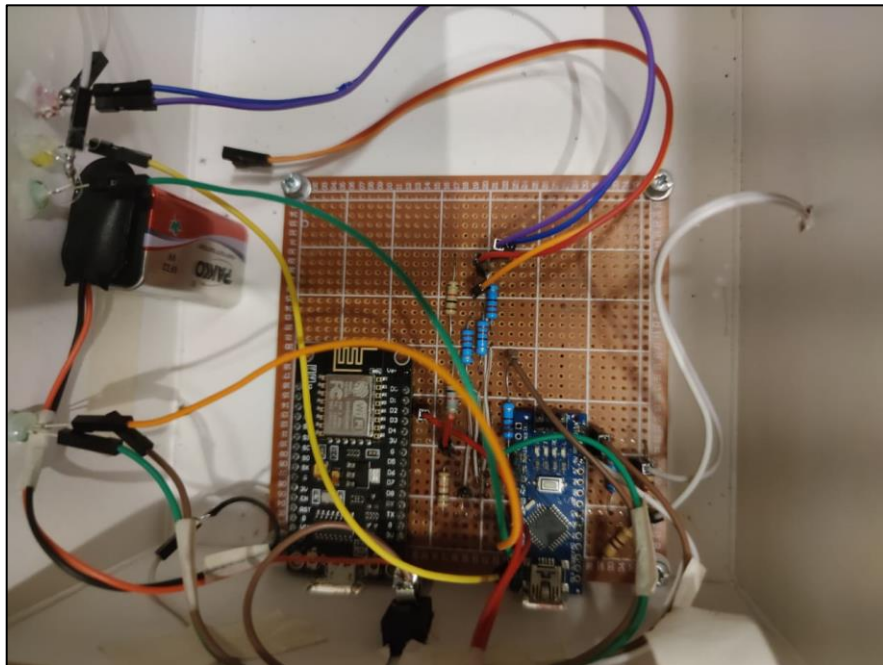


Figura 2 Circuito Final

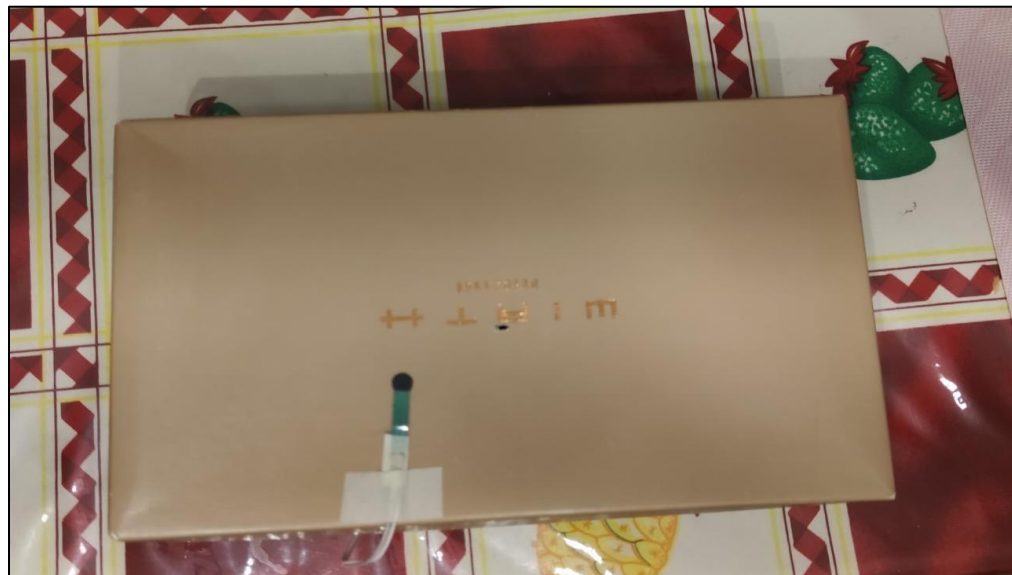


Figura 3 Prototipo Final



Código Fuente

Código Fuente Arduino

Código_Sensor_Nano.ino

```
#include <SoftwareSerial.h> //Libreria para realizar la conexión Serial

#define pinA1 2

#define LEDpin 8

#define BatAlta 10

#define BatMedia 11

#define BatBaja 3

#define AnalogPin 3

int limiteMaximoBateria;

int *bateria_Actual;

int bateria_Act;

int *bateria_Anterior;

int bateria_Ant;

String estado; //Estado del sensor de presion -> Puede ser "ocupado" o "desocupado"

int ResRead; // La Lectura de la Resistencia por División de Tensión.

int *pContador = 0; // Contador único para detectar el primer estado de la cama al iniciar el sistema.

int contador=0;

SoftwareSerial SUART(9,7); //Comunicación serial con el ESP 8266.

//----- SET UP-----

void setup()

{

    Serial.begin(115200); // Enviaremos la información de depuración a través del Monitor de Serial.

    iniciarConfiguracion();

    limiteMaximoBateria = calibrarBateria(1000, 470, 9);

    pContador = &contador;

    bateria_Anterior = &bateria_Ant;
```



```

bateria_Actual = &bateria_Act;
}

// ----- LOOP -----

void loop()
{
    //Sensor nivel de la bateria
    int porcentajeBateria = leerPorcentajeBateria(limiteMaximoBateria);
    *bateria_Actual = enviarBateria(porcentajeBateria);
    if( contador == 1){
        enviarEstadoBateria(*bateria_Anterior,*bateria_Actual);
    }
    //Lectura del sensor
    int lectura = leerSensor(AnalogPin);
    // Detección del estado inicial del sensor y envio al ESP8266 a traves del método print.
    lecturaInicial(estado,lectura,*pContador,*bateria_Actual);
    //Envio de datos unicamente cuando existe un cambio de estado.
    estado = enviarDatosSerial(lectura,estado);
    delay(500); //Cien "ms" de espera en cada lectura
}

//----- FUNCIONES -----

void iniciarConfiguracion(){
    SUART.begin(115200);    // BaudRate al que trabaja el ESP8266.
    pinMode(LEDpin, OUTPUT); // Pin de salida de LED conxi3n.
    // Luces LED indicadoras de porcentaje de batería.
    pinMode(BatBaja,OUTPUT);
    pinMode(BatMedia,OUTPUT);
}

```



```
pinMode(BatAlta,OUTPUT);

while(!Serial){      // Se asegura de realizar la conexión serial antes de avanzar al ciclo infinito.
}

// Luz LED indicadora de bateria.
digitalWrite(LEDpin, HIGH); // Enciende el LED indicador del sistema activo.;
// Mensaje de bienvenida.
Serial.println("Bienvenidos a Get A Bed v1.0");

}

int leerSensor(int analogPin){
    ResRead = analogRead(analogPin); // La Resistencia es igual a la lectura del sensor (Analog 0)
    Serial.print("Lectura Analogica = ");
    Serial.println(ResRead);
    return ResRead;
}

void lecturalInicial(String &estado, int lectura,int &contador,int &porcentaje){
    if(lectura>500 & contador == 0){
        estado = "ocupado";
        Serial.print("ocupado");
        SUART.print('o');
        delay(3000);
    } else if(lectura<500 & contador == 0) {
        estado = "desocupado";
        SUART.print('d');
        delay(3000);
    }
}
```



```

if(contador == 0){
    *bateria_Anterior = porcentaje;
    SUART.print(porcentaje);
}

// Bandera para detener la detección inicial del sensor.
if(contador == 0){
    contador = contador + 1;
}
}

```

```

String enviarDatosSerial(int lectura, String estado){
    if(lectura>500 && estado.equals("desocupado")){
        estado = "ocupado";
        SUART.print('o');
        Serial.println("Ocupado");
    } else if (ResRead<500 && estado.equals("ocupado")){
        estado = "desocupado";
        SUART.print('d');
        Serial.println("Desocupado");
    }
    return estado;
}

/**

```

@brief Función que se debe utilizar de forma obligatoria si se requiere obtener el porcentaje de batería.

Es necesario que se implemente un divisor de voltaje.

Establece el voltaje máximo de entrada que entrega la batería en el rango de [0, 1023].

Debe utilizarse en la función setup() luego de inicializar las variables de la librería.



```

@param rBajo: valor de la resistencia en Ohmios ( $\Omega$ ) que se encuentra conectada al negativo.
@param rArriba: valor de la resistencia en Ohmios ( $\Omega$ ) que está conectada al positivo.
@param VIn: voltaje de la batería, puede indicar un voltaje nominal o realizar mediciones con un
multímetro.

*/
int calibrarBateria(float rBajo, float rArriba, float vIn) {
    float vMax = (rBajo / (rBajo + rArriba)) * vIn;
    int limiteMaximoBateria = (int)(vMax * (1023 / 5));
    Serial.print("Calibrado! - El valor analogico maximo de bateria es = ");
    Serial.println(limiteMaximoBateria);
    delay(1000);
    return limiteMaximoBateria;
}

/**
 @brief Función que se encarga de mapear el nivel de la batería en el rango de [0, 100]
 @return porcentajeBateria: valor de tipo int que indica el nivel actual de carga de la batería.
 */
int leerPorcentajeBateria(int limiteMaximoBateria) {
    int bateria = analogRead(pinA1);
    delay(5); // permite que se estabilice el convertidor analógico-digital (ADC).
    int porcentajeBateria = map((int) bateria, 0, limiteMaximoBateria, 0, 100);
    return porcentajeBateria;
}

/**
 @brief Obtiene el nivel de bateria más bajo que ha sido censado.
        Si existe una caída del nivel de la batería mayor al 5% el porcentaje no cambiará.
 @param porcentajeBateria: valor que será leído continuamente en el programa.

```



```

    @return nivelBateriaMayor: devuelve el porcentaje de batería mayor que ha sido leído.
*/
int compararNivelBateria(int porcentajeBateria) {
    int nivelBateriaMayor = 100;
    if ((porcentajeBateria <= nivelBateriaMayor)) {
        nivelBateriaMayor = porcentajeBateria;
    }
    if ((porcentajeBateria > nivelBateriaMayor - 5)) {
        nivelBateriaMayor = porcentajeBateria;
    }
    return nivelBateriaMayor;
}

```

```

/**
    @brief Función que determina el envío de la batería dado cierto intervalo de tiempo.
           La batería solo se envía cuando el nivel es mayor a 30%.
    @param porcentaje:
*/
int enviarBateria(int porcentaje)
{
    int porcentaje_real = compararNivelBateria(porcentaje);
    if (porcentaje_real > 70) {
        digitalWrite(BatAlta,HIGH);
        digitalWrite(BatMedia,LOW);
        digitalWrite(BatBaja,LOW);
    } else if (30<porcentaje_real <=70){
        digitalWrite(BatAlta,LOW);
        digitalWrite(BatMedia,HIGH);
        digitalWrite(BatBaja,LOW);
    }
}

```



```

}else {
    digitalWrite(BatAlta,LOW);
    digitalWrite(BatMedia,LOW);
    digitalWrite(BatBaja,HIGH);
}
return porcentaje_real;
}

void enviarEstadoBateria(int &cargaAnterior, int &cargaActual){
    if(cargaAnterior != cargaActual){
        cargaAnterior = cargaActual;
        SUART.print(cargaActual);
    }
}

```




```
#include <ESP8266WiFi.h>    //Libreria para la conexión WiFi
#include <SoftwareSerial.h>  //Libreria para la comunicación Serial
#include <FirebaseArduino.h> //Libreria para la conexión a base de datos en tiempo real FireBase

#define FIREBASE_HOST "amst-grupo4-camas-uci.firebaseio.com"    //API KEY o Dirección del HOST de
la base de datos

#define FIREBASE_AUTH "6aMyQhLCpma4yjVPOL1eZrmi2SGMbYsO8HKSRBej" //"Secretos de la base de
datos".

const char* ssid = "NETLIFE-BAILON";    //Nombre de la red WiFi.
const char* password = "ENRIQUE092427"; //Clave de la red WiFi.

const String direccion = "hospital-prueba/secciones/1/camas/1/dispositivo/estado"; //Direccion en
donde se ubica el campo "Estado" de la cama.

const String direccion_bat = "hospital-prueba/secciones/1/camas/1/dispositivo/bateria";

String estado; //Estado de la cama.

SoftwareSerial mySerial(4,5);

// NUEVO

const byte numChars = 32;
char receivedChars[numChars];
boolean newData=false;
int dataNumber=0;

// NUEVO

void setup() {
  Serial.begin(115200); // BaudRate de 115200 para la comunicación Serial.
  mySerial.begin(115200);
  pinMode(4,INPUT);
```



```

WiFi.mode(WIFI_STA);      // Modo WiFi Station para conectarse a la red.
WiFi.begin(ssid,password); // Inicio del intento de conexión a la red.
Serial.print("Conectando a:\t");// Mensaje de conexión a la red WiFi.
Serial.println(ssid);      // Imprimir nombre de la red.

// Lazo while ejecutado hasta conectarse exitosamente a la red.
while(WiFi.status() != WL_CONNECTED){
    delay(200);
    Serial.print(".");
}
//Impresion por consola de la conexión exitosa a la red.
Serial.println();
Serial.print("Conectado a:\t");
Serial.println(WiFi.SSID());
Serial.print("IP address:\t");
Serial.println(WiFi.localIP());

Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
}

void loop() {
    if(Firebase.failed()){
        delay(500);
        Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
        Serial.println("Error");
        delay(500);
    } else{
        char bateria[32];
        int bateria_real;
    }
}

```



```

char x;          // Variable a almacenar el valor enviado.

mySerial.flush(); // Método para recibir exitosamente los datos.

if( mySerial.available() != 0){ // Valida que existan datos a recibir.

    x = mySerial.read();// Lectura del dato recibido.

    Serial.println(x);

    if( x == 'o'){

        estado = "Ocupado";    // Se establece el String de Ocupado si el datos recibido es 1.

    } else if( x == 'd'){

        estado = "Desocupado"; // Se establece el String de Desocupado si el datos recibido es 0.

    } else {

        for(int i =0; i<2;i++){

            bateria[i] = x;

            if(i!=2){

                x = mySerial.read();

            }

        }

        bateria_real = atoi(bateria);

    }

    //Serial.println("Estado: + " + estado + ".");// Impresion en consola del estado actual.

}

if(!(Firebase.getString(direccion).equals(estado)) && estado != "." && estado != ""){

    Firebase.setString(direccion,estado);    // Envio de datos a la base de datos Firebase para su
    actualización.

}

if(!(Firebase.getInt(direccion_bat) == bateria_real) && bateria_real <=100){

    Firebase.setInt(direccion_bat,bateria_real);    // Envio de datos a la base de datos Firebase para su
    actualización.

}

```



```

}

delay(1000); // Espera de un segundo durante cada iteración.

}

```

SplashScreen.java (Pantalla de la pantalla de Inicio)

```

public class Splash_Screen extends AppCompatActivity {
    private FirebaseAuth auth;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splash__screen);
        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                auth=FirebaseAuth.getInstance();
                FirebaseUser user=auth.getCurrentUser();

                if(user!=null){
                    Intent intent = new Intent(Splash_Screen.this,Menu.class);
                    intent.putExtra("nombre",user.getDisplayName());
                    intent.putExtra("imagen",String.valueOf(user.getPhotoUrl()));
                    intent.putExtra("id",user.getId());
                    startActivity(intent);
                    finish();

                }else{
                    Intent intent = new
Intent(Splash_Screen.this,MainActivity.class);
                    startActivity(intent);
                    finish();

                }
            },3000);
        }
    }
}

```

activity_splash_screen.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:layout_width="match_parent"

```



```

    android:layout_height="match_parent"
    android:layout_gravity="center"
    android:background="#80DEEA"
    tools:context=".Splash_Screen">

    <LinearLayout
        android:id="@+id/linearLayout"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        tools:ignore="MissingConstraints">

        <ImageView
            android:id="@+id/splash_screen"
            android:layout_width="200dp"
            android:layout_height="200dp"
            android:src="@drawable/logotipo_proyecto1"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintHorizontal_bias="0.535"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent"
            tools:ignore="MissingConstraints" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:text="@string/app_name"
            android:layout_marginLeft="40dp"
            android:textStyle="bold"
            android:textSize="30dip"></TextView>
    </LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

```

Cama.java (Data class de objeto Cama)

```

public class Cama {
    private String id;
    private String estado;
    private String piso;
    private String bateria;

    public Cama(String id, String estado, String piso,String bateria) {
        this.id = id;
        this.estado=estado;
        this.piso=piso;
    }
}

```



```

        this.bateria=bateria;
    }

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }

    public String getEstado() {
        return estado;
    }

    public void setEstado(String estado) {
        this.estado = estado;
    }

    public String getPiso() {
        return piso;
    }

    public void setPiso(String piso) {
        this.piso = piso;
    }

    public String getBateria() {
        return bateria;
    }

    @Override
    public String toString() {
        return "Cama{" +
            "id='" + id + '\'' +
            ", estado='" + estado + '\'' +
            ", piso='" + piso + '\'' +
            '}';
    }
}

```

Seccion.java

```

public class Seccion {
    private String id;
    private String nombre;
    private DatabaseReference db;

    Seccion(String id,String nombre){
        this.id=id;
        this.nombre=nombre;
    }

    public String getId() {

```



```

        return id;
    }

    public String getNombre() {
        return nombre;
    }

    private void fetchNombre(){
        db.child("nombre").addListenerForSingleValueEvent(
            new ValueEventListener() {
                @Override
                public void onDataChange(@NonNull DataSnapshot snapshot) {
                    nombre= (String) snapshot.getValue();
                }

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

                }
            }
        );
    }

    @Override
    public String toString() {
        return nombre;
    }
}

```

MainActivity.java (Pantalla de login)

```

public class MainActivity extends AppCompatActivity {
    static final int GOOGLE_SIGN_IN= 123;
    FirebaseAuth mAuth;
    DatabaseReference db;
    GoogleSignInClient mGoogleSignInClient;
    Button btn_login;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_ingreso);
        mAuth = FirebaseAuth.getInstance();
        Intent intent = getIntent();
        GoogleSignInOptions gso = new GoogleSignInOptions.Builder
            (GoogleSignInOptions.DEFAULT_SIGN_IN)
            .requestIdToken(getString(R.string.default_web_client_id))
            .requestEmail()
            .build();
        mGoogleSignInClient = GoogleSignIn.getClient(this,gso);
        String msg = intent.getStringExtra("msg");
        if(msg != null){
            if(msg.equals("cerrarSesion")){
                cerrarSesion();
            }
        }
    }
}

```




```

    }
}
private void cerrarSesion() {
    mGoogleSignInClient.signOut();
}

public void iniciarSesion(View view){
    Intent signInIntent = mGoogleSignInClient.getSignInIntent();
    startActivityForResult(signInIntent, GOOGLE_SIGN_IN);
}

@Override
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);
            }else {
                updateUI(null);
            }
        }catch (ApiException e){
            Context context = getApplicationContext();
            CharSequence text = "Error al iniciar sesión";
            int duration = Toast.LENGTH_SHORT;
            Toast toast = Toast.makeText(context, text, duration);
            toast.show();
        }
    }
}

private void firebaseAuthWithGoogle(GoogleSignInAccount acct){
    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 updateUI(FirebaseUser user){
    addToEnfermeros(user);
    Intent intent= new Intent(getBaseContext(), com.example.getabed.Menu.class);
    intent.putExtra("nombre",user.getDisplayName());
    intent.putExtra("imagen",String.valueOf(user.getPhotoUrl()));
}

```



```

        intent.putExtra("id",user.getUid());
        startActivity(intent);
    }

    private void addToEnfermeros(FirebaseUser user){

        FirebaseMessaging.getInstance().getToken()
            .addOnCompleteListener(new OnCompleteListener<String>() {
                @Override
                public void onComplete(@NonNull Task<String> task) {
                    if (!task.isSuccessful()) {
                        Log.w("TokenError", "Fetching FCM registration token
failed", task.getException());
                        return;
                    }

                    // Get new FCM registration token
                    String token = task.getResult();
                    db=
FirebaseDatabase.getInstance().getReference().child("hospital-
prueba").child("enfermeros");
                    DatabaseReference userDb= db.child(user.getUid());
                    userDb.child("nombre").setValue(user.getDisplayName());
                    userDb.child("email").setValue(user.getEmail());

                    userDb.child("imagen").setValue(user.getPhotoUrl().toString());
                    userDb.child("token_notification").setValue(token);

                }
            });

    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        int id = item.getItemId();
        if (id == R.id.action_settings) {
            return true;
        }
        return super.onOptionsItemSelected(item);
    }
}

```

activity_ingreso.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"

```



```

android:layout_height="match_parent"
android:background="@drawable/fondo3"
android:orientation="vertical">

<TextView
    android:id="@+id/textView"
    android:layout_width="275dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="45dp"
    android:layout_marginTop="65dp"
    android:layout_marginRight="25dp"
    android:background="#00000000"
    android:gravity="center"
    android:text="Bienvenido"
    android:textSize="32dp" />

<ImageView
    android:id="@+id/imageView"
    android:layout_width="match_parent"
    android:layout_height="150dp"
    android:layout_marginTop="40dp"
    android:maxHeight="70dp"
    app:srcCompat="@drawable/logotipo_proyecto1" />

<EditText
    android:id="@+id/editTextTextPersonName"
    android:layout_width="356dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="27dp"
    android:layout_marginTop="30dp"
    android:background="@drawable/borde_redondo"
    android:ems="10"
    android:hint="Usuario"
    android:paddingVertical="5dip"
    android:paddingHorizontal="20dip"
    android:inputType="textPersonName"
    android:textSize="25dp" />

<EditText
    android:id="@+id/editTextTextPassword"
    android:layout_width="356dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="27dp"
    android:layout_marginTop="10dp"
    android:background="@drawable/borde_redondo"
    android:ems="10"
    android:hint="Contraseña"
    android:paddingVertical="5dip"
    android:paddingHorizontal="20dip"
    android:inputType="textPassword"
    android:textSize="25dp" />

<Button
    android:id="@+id/button_ingreso"

```



```

        android:layout_width="250dp"
        android:layout_height="wrap_content"
        android:layout_marginLeft="85dp"
        android:layout_marginTop="40dp"
        android:layout_marginRight="20dp"
        android:gravity="center_horizontal"
        android:text="Ingresar"
        android:textAlignment="center"
        android:textColor="@color/white"
        android:textSize="15dp"
        android:background="@drawable/borde_redondo"
        app:backgroundTint="#FF8A65" />

```

<Button

```

        android:id="@+id/btn_login"
        android:layout_width="250dp"
        android:layout_height="wrap_content"
        android:layout_marginLeft="85dp"
        android:layout_marginTop="20dp"
        android:layout_marginRight="20dp"
        android:gravity="center_horizontal"
        android:onClick="iniciarSesion"
        android:text="Inicio sesion con google"
        android:textAlignment="center"
        android:textColor="@color/white"
        android:textSize="15dp"
        app:backgroundTint="#FF8A65"
        android:background="@drawable/borde_redondo" />

```

<ImageView

```

        android:id="@+id/image_googlelogo"
        android:layout_width="303dp"
        android:layout_height="84dp"
        android:layout_marginTop="0dp"
        android:layout_marginLeft="55dp"
        android:gravity="center_horizontal"
        app:srcCompat="@drawable/googlelogo" />

```

```
</LinearLayout>
```

Menu.java (Pantalla de menú)

```

public class Menu extends AppCompatActivity {
    private Button bsignOut;
    private ProgressBar psignOut;
    private FirebaseAuth mAuth;
    private GoogleSignInOptions gso;
    private TextView saludo;
    private ImageView imagenPerfil;
    private SwitchCompat switchNotificaciones;
    private String id;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

```



```

super.onCreate(savedInstanceState);
setContentView(R.layout.activity_menu);
mAuth= FirebaseAuth.getInstance();
bsignOut = (Button) findViewById(R.id.button_cerrarsesion);
psignOut = (ProgressBar) findViewById(R.id.progressBar2);
saludo= (TextView) findViewById(R.id.saludo);
imagenPerfil= (ImageView) findViewById(R.id.fotoPerfil);
Intent intent= getIntent();
saludo.setText(intent.getStringExtra("nombre"));
String foto= intent.getStringExtra("imagen");
id= intent.getStringExtra("id");

Picasso.with(getApplicationContext()).load(foto).into(imagenPerfil);
switchNotificaciones= findViewById(R.id.notificacionSwitch);
obtenerEstadoPerfil();
switchNotificaciones.setOnCheckedChangeListener(
    new CompoundButton.OnCheckedChangeListener() {
        @Override
        public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {
            onChecked(isChecked);
        }
    }
);

bsignOut.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        psignOut.setVisibility(View.VISIBLE);
        FirebaseAuth.getInstance().signOut();
        finish();
        Intent intent= new Intent(Menu.this,MainActivity.class);
        intent.putExtra("msg", "cerrarSesion");
        startActivity(intent);
        //EpsignOut.setVisibility(View.GONE);
        startActivity(intent);
        Toast.makeText(Menu.this, "Cierre de sesión",
Toast.LENGTH_SHORT).show();
        finish();
    }
});
}

private void obtenerEstadoPerfil(){
    FirebaseDatabase.getInstance().getReference().child("/hospital-
prueba/enfermeros/"+id).addValueEventListener(
        new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                String estado= (String) snapshot.child("estado").getValue();
                Log.d("de", "onDataChange: "+estado);
                if(estado != null && estado.equals("En espera")){
                    switchNotificaciones.setChecked(true);
                }else{

```



```

        switchNotificaciones.setChecked(false);
    }
}

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

}

);
}

public void onChecked( boolean checked){
    if(checked){
        FirebaseDatabase.getInstance().getReference().child("/hospital-
prueba/enfermeros/"+id+"/estado").setValue("En espera");
    }else{
        FirebaseDatabase.getInstance().getReference().child("/hospital-
prueba/enfermeros/"+id+"/estado").setValue("Disponible");
    }
}

public void verCamasDisponibles(View view){
    Intent intent= new Intent(getBaseContext(),EstadoCamas.class);
    startActivity(intent);
}

@Override
public boolean onCreateOptionsMenu(android.view.Menu menu) {
    getMenuInflater().inflate(R.menu.menu_main, menu);
    return super.onCreateOptionsMenu(menu);
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    if (id == R.id.button_cerrarsesion) {
        cerrarSesion();
        return true;
    }
    return super.onOptionsItemSelected(item);
}

private void cerrarSesion(){
    Intent intent= new Intent(getBaseContext(),MainActivity.class);
    intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK|Intent.FLAG_ACTIVITY_CLEAR_TASK);
    startActivity(intent);
}
}

```



activity_menu.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:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/fondo2"
>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <Space
            android:layout_width="match_parent"
            android:layout_height="70dp" />

        <TextView
            android:id="@+id/bienvenido"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:text="Bienvenido"
            android:textSize="30dp"
            android:textStyle="bold"
            android:textColor="@color/black"/>

        <TextView
            android:id="@+id/saludo"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:text=""
            android:textSize="30dp"
            android:textStyle="bold"
            android:textColor="@color/black"/>

        <Space
            android:layout_width="match_parent"
            android:layout_height="20dp" />

        <ImageView
            android:id="@+id/fotoPerfil"
            android:layout_width="match_parent"
            android:layout_height="200dp"
            android:maxHeight="100dp"
            app:srcCompat="@drawable/logotipo_proyecto1" />

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:gravity="center"
```




```

        android:layout_marginTop="20dp"
    >
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Avisar cuando se desocupen camas"
        android:textStyle="bold"
        android:textSize="20dip"
    >
</TextView>
<androidx.appcompat.widget.SwitchCompat
    android:id="@+id/notificacionSwitch"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
></androidx.appcompat.widget.SwitchCompat>

</LinearLayout>
<Button
    android:id="@+id/button_camasdisponibles"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="30dp"
    android:layout_marginRight="30dp"
    android:layout_marginTop="20dp"
    android:onClick="verCamasDisponibles"
    android:paddingVertical="10dp"
    android:text="Ver camas disponibles"
    android:textSize="15dp"
    app:backgroundTint="#2F7AC5"
    android:background="@drawable/borde_redondo"/>

<Space
    android:layout_width="match_parent"
    android:layout_height="20dp" />

<Button
    android:id="@+id/button_cerrarsesion"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="30dp"
    android:layout_marginRight="30dp"
    android:paddingVertical="10dp"
    android:text="Cerrar Sesion"
    android:textSize="15dp"
    app:backgroundTint="#49535D"
    android:background="@drawable/borde_redondo"/>

<ProgressBar
    android:id="@+id/progressBar2"
    style="?android:attr/progressBarStyle"
    android:layout_width="102dp"
    android:layout_height="83dp"
    android:layout_marginLeft="155dp"
    android:layout_marginRight="30dp"
    android:gravity="center"

```



```

        android:indeterminateDrawable="@drawable/progressbar_design"
        android:indeterminateDuration="@android:integer/config_longAnimTime"
        android:paddingTop="3pt"
        android:visibility="gone"
    />

    <ProgressBar
        android:id="@+id/progressBar2"
        style="?android:attr/progressbarStyle"
        android:layout_width="102dp"
        android:layout_height="83dp"
        android:layout_marginLeft="155dp"
        android:layout_marginRight="30dp"
        android:gravity="center"
        android:indeterminateDrawable="@drawable/progressbar_design"
        android:indeterminateDuration="@android:integer/config_longAnimTime"
        android:paddingTop="3pt"
        android:visibility="gone"
    />

</LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>

```

EstadoCamas.java (Pantalla donde se muestran el estado de las camas)

```

public class EstadoCamas extends AppCompatActivity implements
    AdapterView.OnItemClickListener {
    Spinner spinner;
    LinearLayout view;
    CamasSeccion fragment;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_estado_camas);
        spinner= (Spinner) findViewById(R.id.spinner);
        inicializarSpinner();
        spinner.setOnItemClickListener(this);
        view= (LinearLayout) findViewById(R.id.view);
        Bundle bundle= new Bundle();
        bundle.putString("idSeccion","1");
        fragment= new CamasSeccion();
        fragment.setArguments(bundle);
        FragmentTransaction ft = getSupportFragmentManager().beginTransaction();

        ft.replace(R.id.your_placeholder, fragment);
        ft.commit();
    }

    private void inicializarSpinner(){
        FirebaseDatabase.getInstance().getReference()
            .child("hospital-prueba").child("secciones").addValueEventListener(

```



```

        new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot)
        {
            ArrayList<Seccion> arrayList= new ArrayList();
            for(DataSnapshot child: snapshot.getChildren()){
                String nombre= (String)
child.child("nombre").getValue();
                Seccion seccion= new
Seccion(child.getKey(),nombre);
                arrayList.add(seccion);
            }
            int size= arrayList.size();
            Seccion[] secciones= new Seccion[size];
            for(int i =0 ; i<size;i++){
                Seccion sec= arrayList.get(i);
                secciones[i]= sec;
            }
            SpinAdapter arrayAdapter= new
SpinAdapter(getApplicationContext(), android.R.layout.simple_spinner_item,secciones);

            spinner.setAdapter(arrayAdapter);
        }

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

        }
    }

    );

}

public void irGrafico(View view){
    Intent intent= new Intent(getBaseContext(),GraficoDisponibilidad.class);
    Seccion seccion= (Seccion) spinner.getSelectedItem();
    intent.putExtra("seccionId",seccion.getId());
    startActivity(intent);
}

@Override
public void onItemClick(AdapterView<?> parent, View view, int position, long
id) {
    Seccion seccion = (Seccion) spinner.getItemAtPosition(position);
    fragment.obtenerCamas(seccion.getId());
    Log.d("Seleccion", "onItemSelected: Seccion");
}

@Override
public void onNothingSelected(AdapterView<?> parent) {

}
}

```



activity_estado_camas.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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/view"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/white"
    android:orientation="vertical"
    tools:context=".EstadoCamas">

    <Spinner
        android:id="@+id/spinner"
        android:layout_width="match_parent"
        android:layout_height="40dp"
        android:layout_marginRight="30dp"
        android:layout_marginLeft="30dp"
        android:background="@drawable/borde_redondo"
        android:layout_marginTop="15dp" />

    <TextView
        android:layout_marginTop="15dp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="20dp"
        android:text="Camas"
        android:layout_marginLeft="10dp"
        android:layout_marginBottom="10dp"
        android:layout_gravity="left"
        android:textColor="@color/black"
    ></TextView>

    <FrameLayout
        android:id="@+id/your_placeholder"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

    </FrameLayout>

    <Button
        android:layout_marginTop="30dp"
        android:id="@+id/buttonGrafico"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginHorizontal="30dp"
        android:text="Ver grafica"
        android:onClick="irGrafico"
    />
</LinearLayout>
```

SpinAdapter.java (Seleccion de seccion en Acitivy EstadoCamas)

```
public class SpinAdapter extends ArrayAdapter<Seccion> {
```



```
// Your sent context
private Context context;
// Your custom values for the spinner (User)
private Seccion[] values;

public SpinAdapter(Context context, int textViewResourceId,
                    Seccion[] values) {
    super(context, textViewResourceId, values);
    this.context = context;
    this.values = values;
}

@Override
public int getCount(){
    return values.length;
}

@Override
public Seccion getItem(int position){
    return values[position];
}

@Override
public long getItemId(int position){
    return position;
}

// And the "magic" goes here
// This is for the "passive" state of the spinner
@Override
public View getView(int position, View convertView, ViewGroup parent) {
    // I created a dynamic TextView here, but you can reference your own custom
    // layout for each spinner item
    TextView label = (TextView) super.getView(position, convertView, parent);
    label.setTextColor(Color.BLACK);
    // Then you can get the current item using the values array (Users array) and
    // the current position
    // You can NOW reference each method you has created in your bean object
    // (User class)
    label.setText(values[position].getNombre());

    // And finally return your dynamic (or custom) view for each spinner item
    return label;
}

// And here is when the "chooser" is popped up
// Normally is the same view, but you can customize it if you want
@Override
public View getDropDownView(int position, View convertView,
                             ViewGroup parent) {
    TextView label = (TextView) super.getDropDownView(position, convertView,
    parent);
    label.setTextColor(Color.BLACK);
    label.setText(values[position].getNombre());
}
```



```

        return label;
    }
}

```

CamasAdapter.java (Creacion de objetos de Grid View donde se muestran las camas)

```

public class CamasAdapter extends BaseAdapter {
    Context context;
    ArrayList<Cama> arrayList;

    public CamasAdapter(Context context, ArrayList<Cama> arrayList) {
        this.context = context;
        this.arrayList = arrayList;
    }
    @Override
    public int getCount() {
        return arrayList.size();
    }
    @Override
    public Object getItem(int position) {
        return arrayList.get(position);
    }
    @Override
    public long getItemId(int i) {
        return i;
    }
    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        Cama cama= arrayList.get(position);
        convertView = LayoutInflater.from(context).inflate(R.layout.cama_ocupada,
parent, false);
        ImageView img= (ImageView) convertView.findViewById(R.id.cama);
        TextView idView= (TextView) convertView.findViewById(R.id.idCama);
        idView.setText(idView.getText()+cama.getId());
        TextView estadoView= (TextView) convertView.findViewById(R.id.estadoCama);
        estadoView.setText(cama.getEstado());
        if ( cama.getEstado().equals("Ocupado")) {
            img.setImageResource(R.drawable.cama_ocupado);
            estadoView.setTextColor(Color.parseColor("#E6352B"));
        }else{
            img.setImageResource(R.drawable.cama_desocupada);
            estadoView.setTextColor(Color.parseColor("#04A04B"));
        }

        TextView bateriaView= (TextView) convertView.findViewById(R.id.bateriaView);
        bateriaView.setText(bateriaView.getText()+cama.getBateria()+" %");
        return convertView;
    }
}

```



fragemnt_camass.xml (Layout que contiene el Grid View donde se van a mostrar las camas)

```
<?xml version="1.0" encoding="utf-8"?>

<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="480dp"
    android:layout_marginTop="10dp"
    tools:context=".CamassSeccion">

    <GridView
        android:id="@+id/gridview"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:columnWidth="120dp"
        android:horizontalSpacing="10dp"
        android:numColumns="auto_fit"
        android:verticalSpacing="10dp" />

</FrameLayout>
```

Cama_desocupada.xml (Layout de cada cama)

```
<LinearLayout
    android:layout_height="wrap_content"
    android:layout_width="120dp"
    android:orientation="vertical"
    xmlns:android="http://schemas.android.com/apk/res/android">
    <ImageView
        android:id="@+id/cama"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:layout_gravity="center"
        android:src="@drawable/cama_desocupada">

    </ImageView>
    <TextView
        android:id="@+id/idCama"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="15dp"
        android:gravity="center"
        android:text="Id: "
    ></TextView>
    <TextView
        android:id="@+id/estadoCama"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="15dp"
```




```

        android:gravity="center"
        android:text="Estado"
        android:textStyle="italic"
    ></TextView>
    <TextView
        android:id="@+id/bateriaView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Bateria:"
        android:gravity="center"
        android:textSize="15dp"
    ></TextView>
</LinearLayout>

```

GraficoDisponibilidad.java (Pantalla con diagrama pastel de disponibilidad de cama)

```

public class GraficoDisponibilidad extends AppCompatActivity {
    String idSeccion;
    PieChart pieChart;
    PieData pieData;
    PieDataSet pieDataSet;
    List<PieEntry> pieEntries;
    ArrayList PieEntryLabels;
    TextView nombreView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_grafico_disponibilidad);
        Intent intent= getIntent();
        idSeccion= intent.getStringExtra("seccionId");
        Log.d("cama", "onCreate: "+idSeccion);
        pieChart = findViewById(R.id.pieChart);
        nombreView= findViewById(R.id.nombreView);
        obtenerNombre();
        obtenerInfo();
    }

    private void obtenerInfo(){
        FirebaseDatabase.getInstance().getReference().child("hospital-prueba")
        .child("secciones").child(idSeccion).child("camas").addValueEventListener(
            new ValueEventListener() {
                @Override
                public void onDataChange(@NonNull DataSnapshot snapshot) {
                    int disponibles=0;
                    int ocupadas=0;
                    for(DataSnapshot child: snapshot.getChildren()){
                        String estado= (String)
                        child.child("dispositivo").child("estado").getValue();
                        if(estado.equals("Ocupado")){
                            ocupadas++;
                        }else{
                            disponibles++;
                        }
                    }
                }
            }
        );
    }
}

```



```

    }
    }
    Log.d("cama", "onDataChange: "+Integer.toString(ocupadas));
    pieEntries= new ArrayList();
    pieEntries.add(new PieEntry(ocupadas,"Ocupadas"));
    pieEntries.add(new PieEntry(disponibles,"Disponibles"));
    pieDataSet = new PieDataSet(pieEntries, "Disponibilidad de
camas");

    pieData = new PieData(pieDataSet);
    pieChart.setData(pieData);
    pieDataSet.setColors(Color.RED, Color.rgb(23,169,61));
    pieDataSet.setSliceSpace(2f);
    pieDataSet.setValueTextColor(Color.WHITE);
    pieDataSet.setValueTextSize(20f);
    pieDataSet.setSliceSpace(5f);
    pieChart.invalidate();
    }

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

private void obtenerNombre(){
    FirebaseDatabase.getInstance().getReference().child("hospital-prueba")
    .child("secciones").child(idSeccion).child("nombre").addValueEventListener(
        new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                nombreView.setText((String) snapshot.getValue());
            }

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

```

activity_grafico_disponibilidad.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:layout_width="match_parent"

```



```

android:layout_height="match_parent"
tools:context=".GraficoDisponibilidad">
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    tools:ignore="MissingConstraints">
    <TextView
        android:id="@+id/nombreView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:gravity="center"
        android:textSize="20dp"
        android:text="Nombre"
    ></TextView>
    <com.github.mikephil.charting.charts.PieChart
        android:id="@+id/pieChart"
        android:layout_width="match_parent"
        android:layout_height="600dp"
        tools:ignore="MissingConstraints" />

</LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>

```

Somos el grupo #4 del material Itinerario Aplicaciones móviles y Servicios Telemáticos. Nuestro proyecto va dirigido a hospitales que debido a esta pandemia se les complica llevar un correcto manejo de las personas que ingresan a la unidad de cuidados intensivos, por lo que en muchas ocasiones se genera un caos innecesario.

GetAbed permitirá a los trabajadores de la unidad mencionada poder conocer el estado de todas las camas que se encuentran localizadas allí, además en caso de necesitar de urgencia, se podrá ubicar en la lista de espera, en la cual al desocuparse una cama se le enviará una notificación al celular por medio de la app indicando la disponibilidad de la misma.

