

# Back On Track

Presentación Proyecto Final



# Mia Payssé

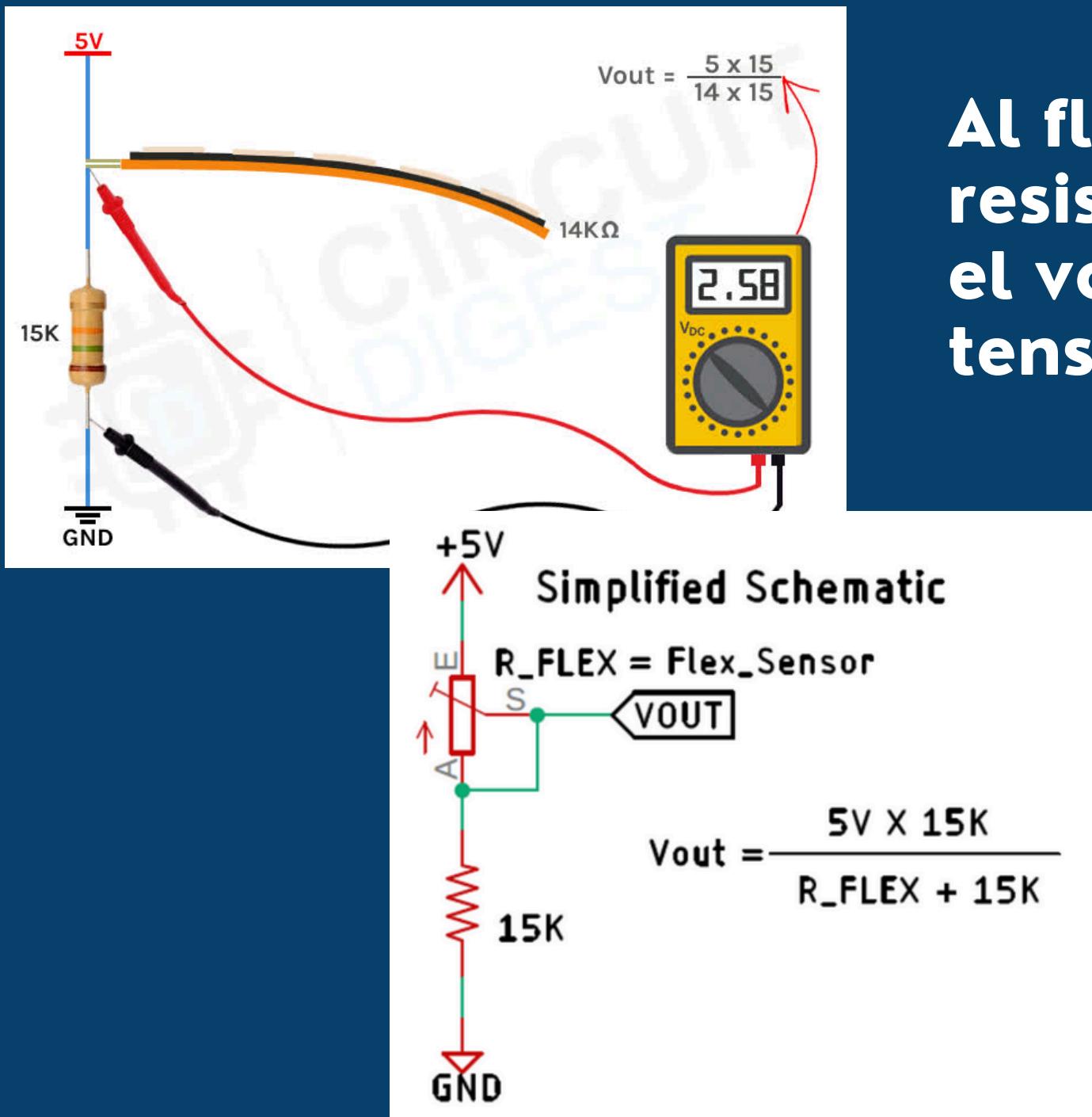
- Elección de componentes
- Programación de sensores
- Conexión con la base de datos
- Problemas
- Colaboración en el diseño del esquemático



# Elección de componentes

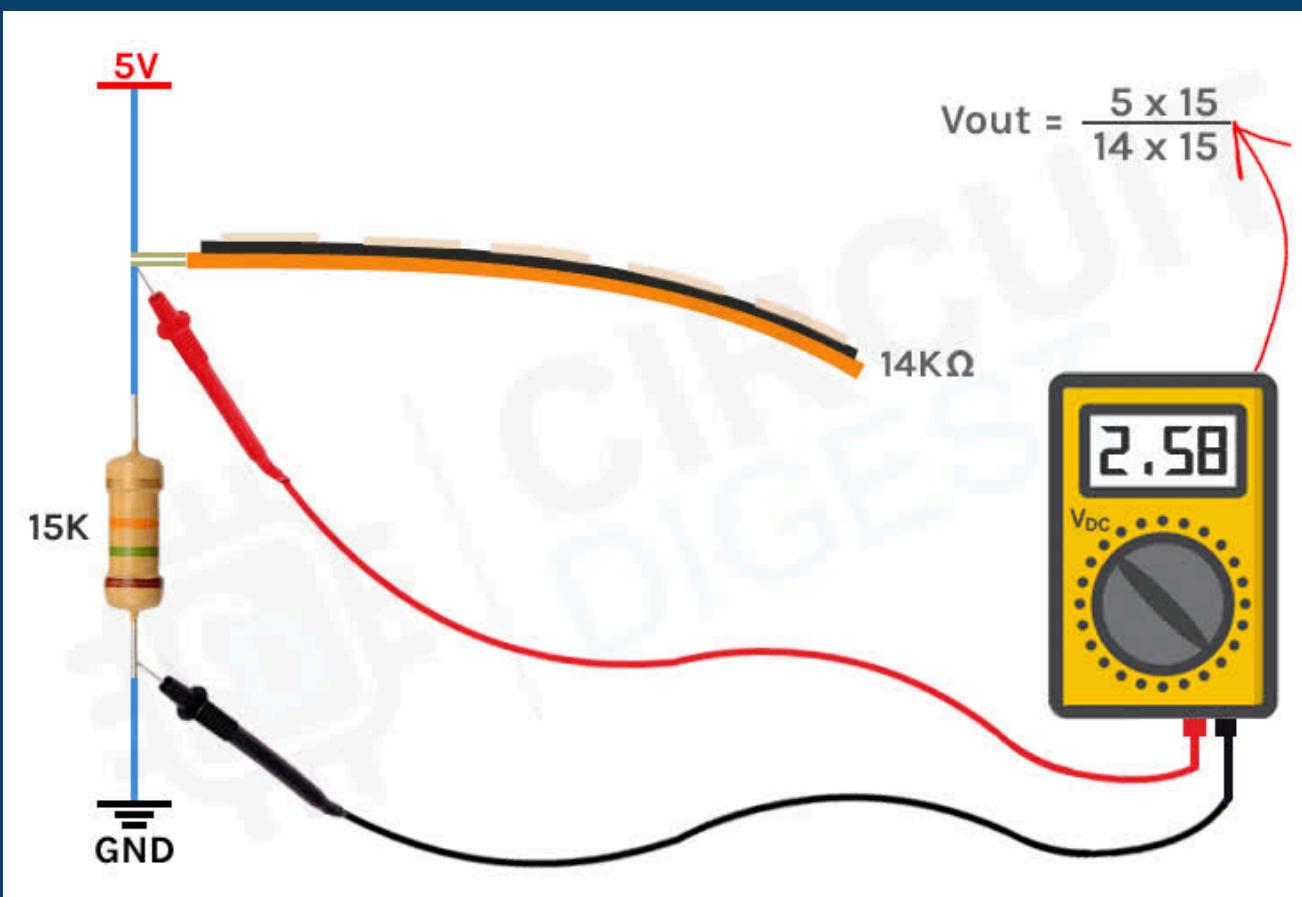


**Resistencias flexibles**

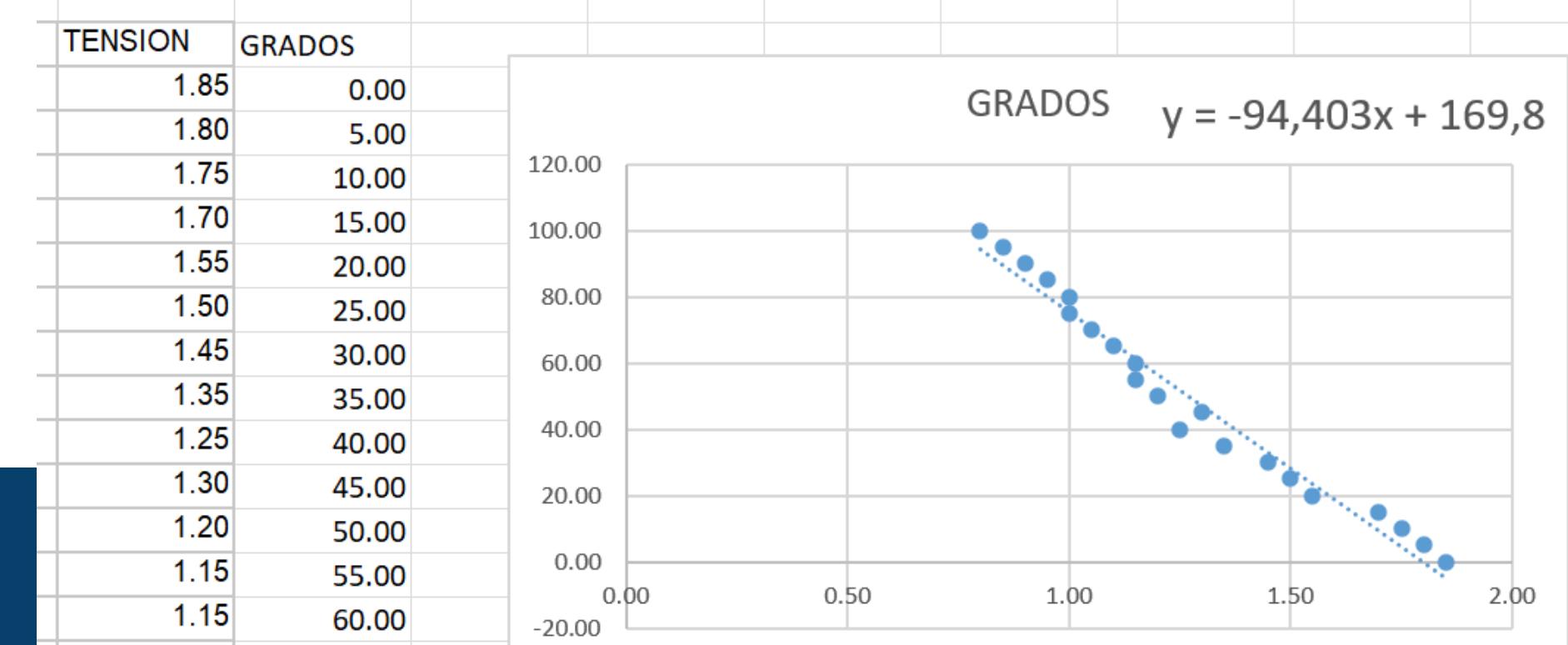
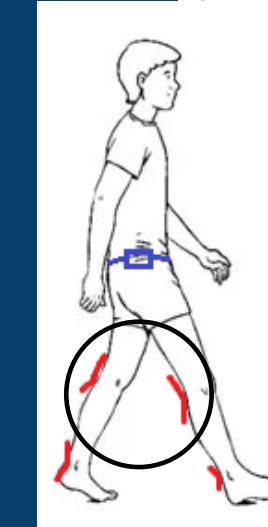


**Al flexionarla aumenta su resistencia y por lo tanto el voltaje en el divisor de tensión disminuye.**

# Elección de componentes



Nosotros utilizamos esta variación para calcular el grado de flexión de la resistencia y por lo tanto de las rodillas



# Elección de componentes



Resistencias flexibles

Funcionamiento simple



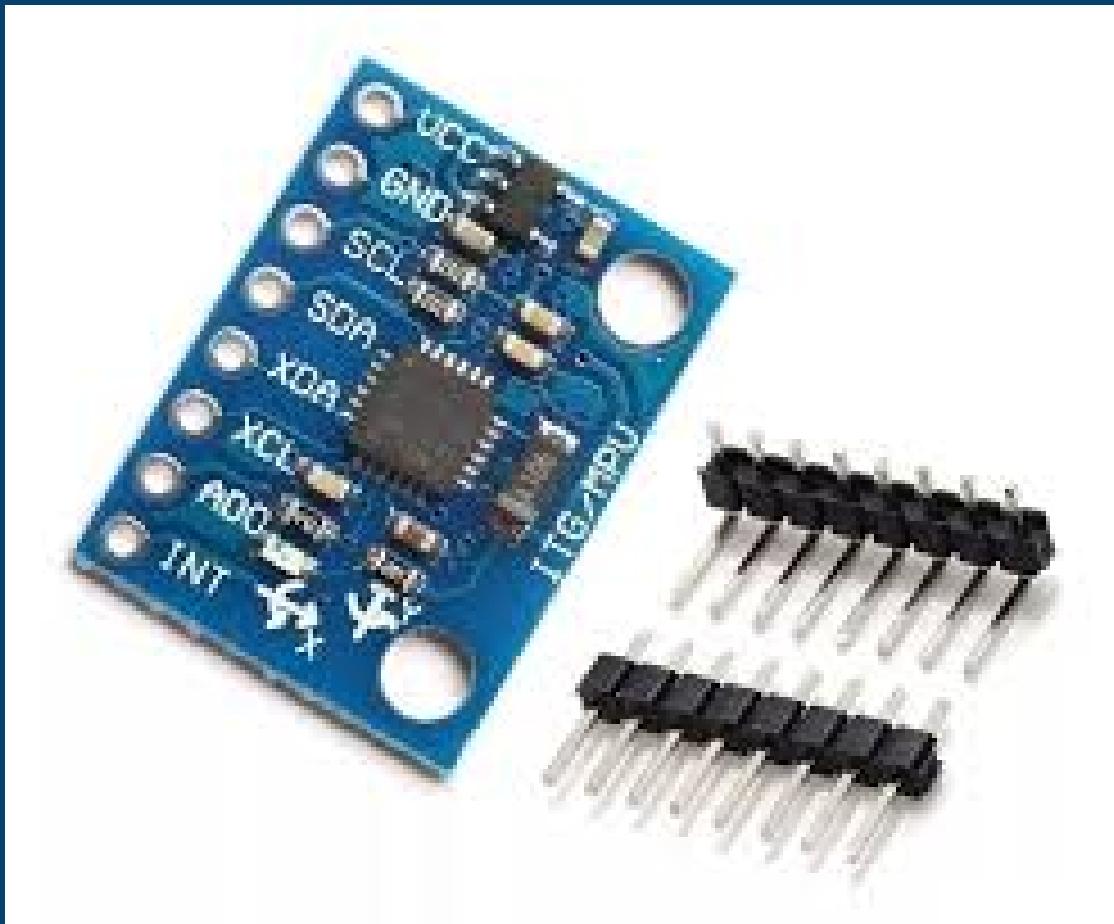
No necesita ningún tipo de librería



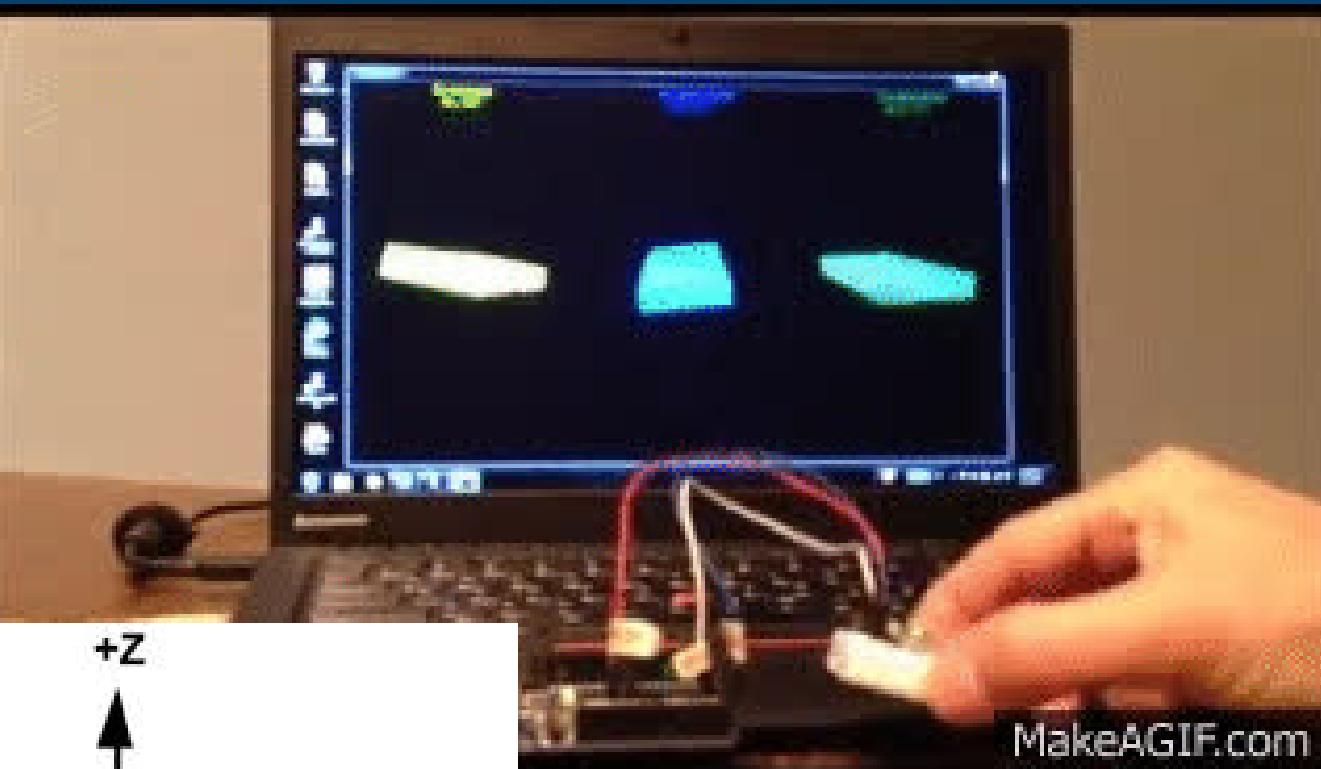
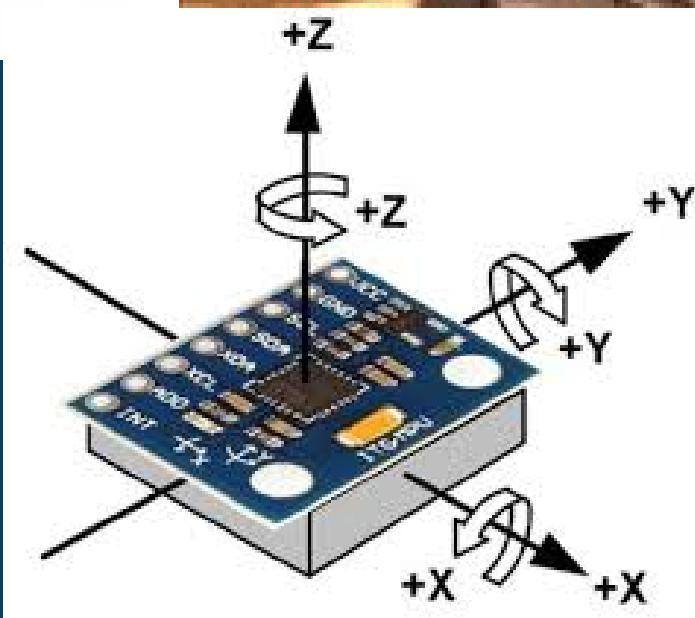
Son muy frágiles



# Elección de componentes



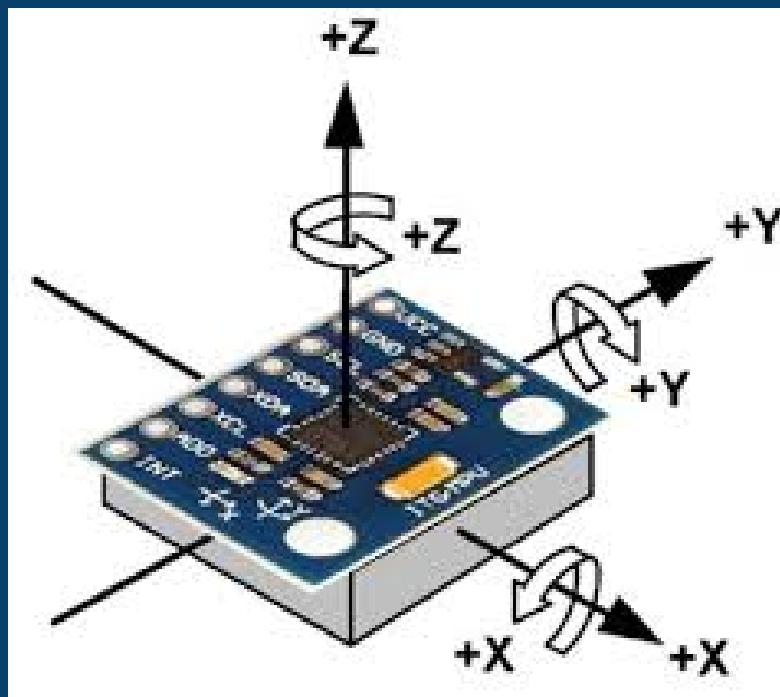
**mpu - 6050**



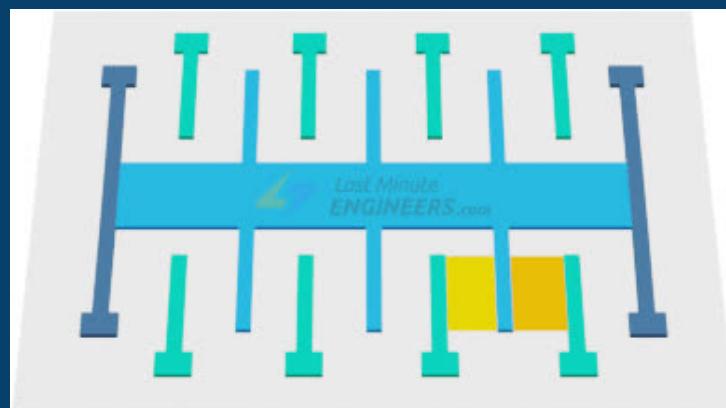
**Unidad de medición  
inercial o IMU  
(Inertial Measurement  
Units)**

**Combina un acelerómetro  
de 3 ejes y un giroscopio  
de 3 ejes**

# Elección de componentes



**mpu - 6050**



**MEMS (Micro-Electro-Mechanical System)**

**Acelerómetro de 3 ejes**

**2da ley de Newton**

$$F = m \cdot a \longrightarrow a = F/m$$

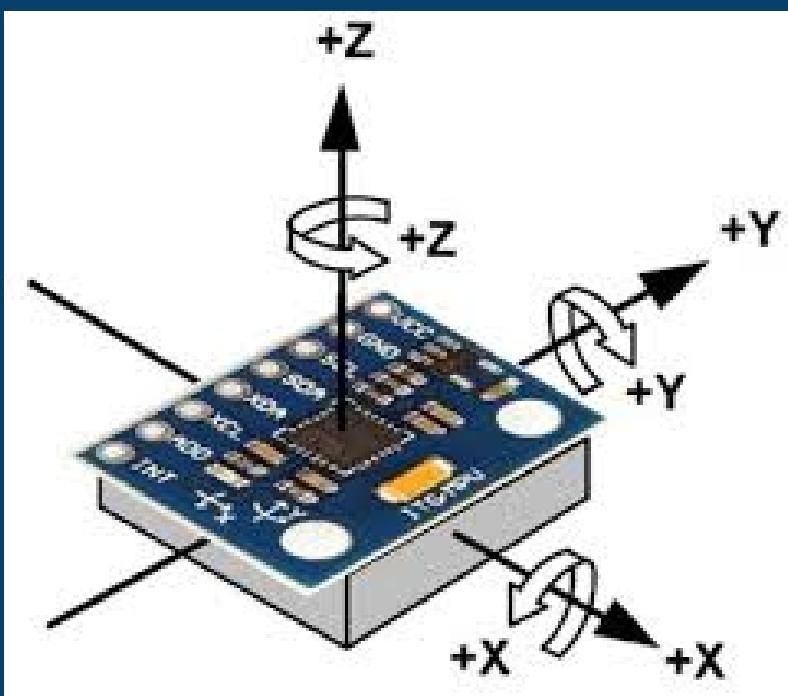


$$a = dV/dt \longrightarrow V = dX/dt$$



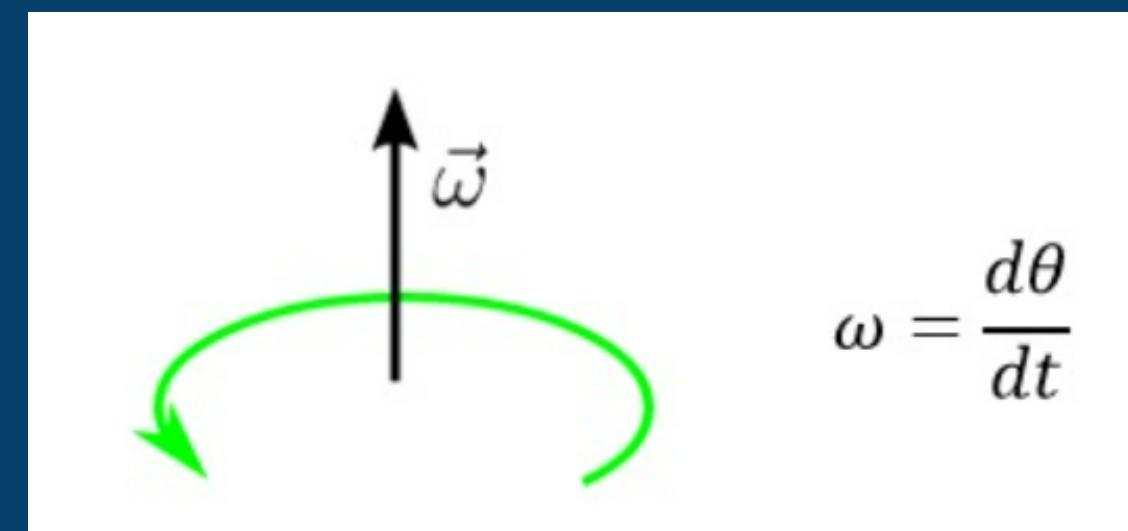
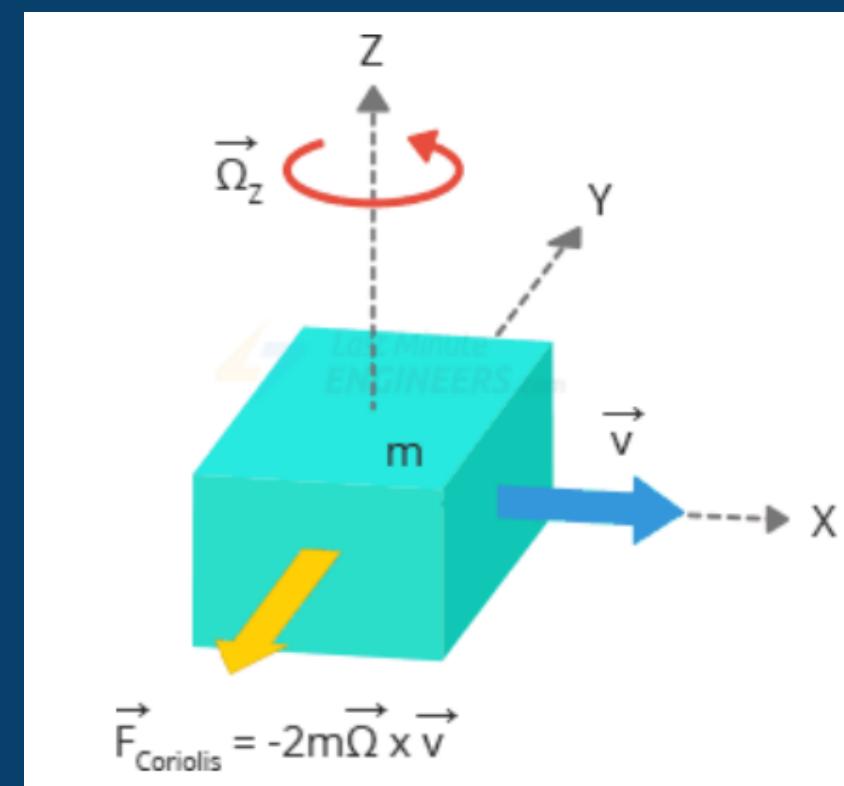
**UBICACIÓN**

# Elección de componentes

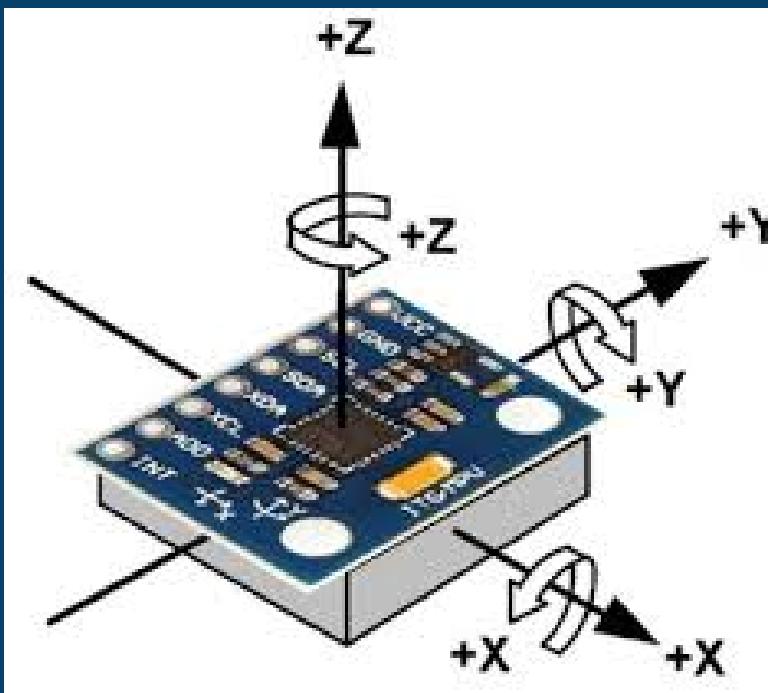


**mpu - 6050**

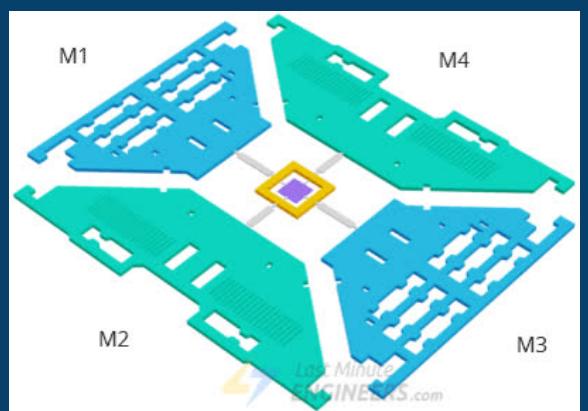
**Giroscopio de 3 ejes**  
**Efecto Coriolis → Velocidad angular**



# Elección de componentes



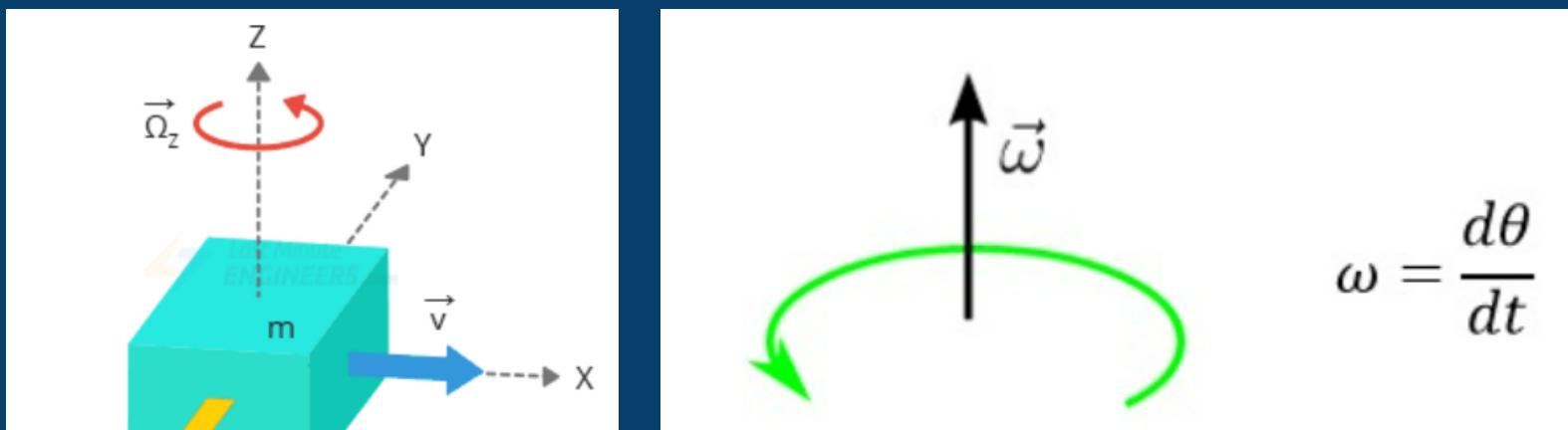
**mpu - 6050**



**MEMS (Micro-Electro-Mechanical System)  
gyroscope**

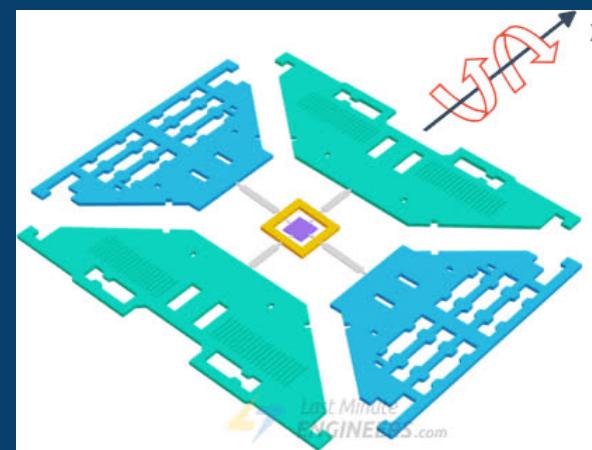
**Giroscopio de 3 ejes  
Efecto Coriolis → Velocidad angular**

**Rotación**

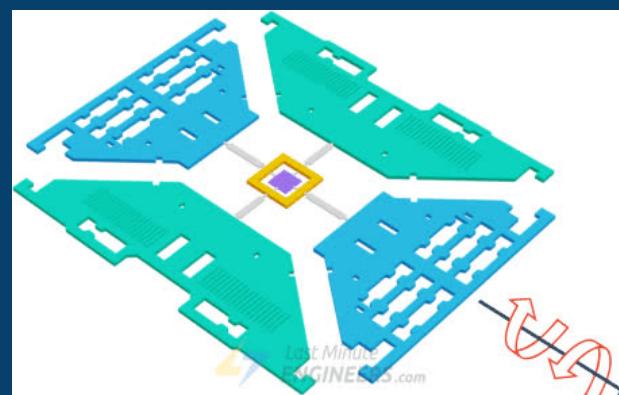


$$\omega = \frac{d\theta}{dt}$$

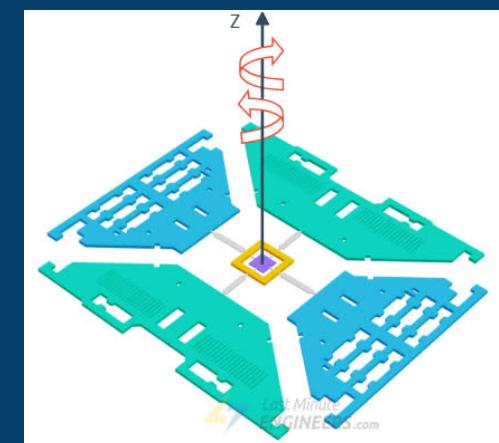
**Roll**



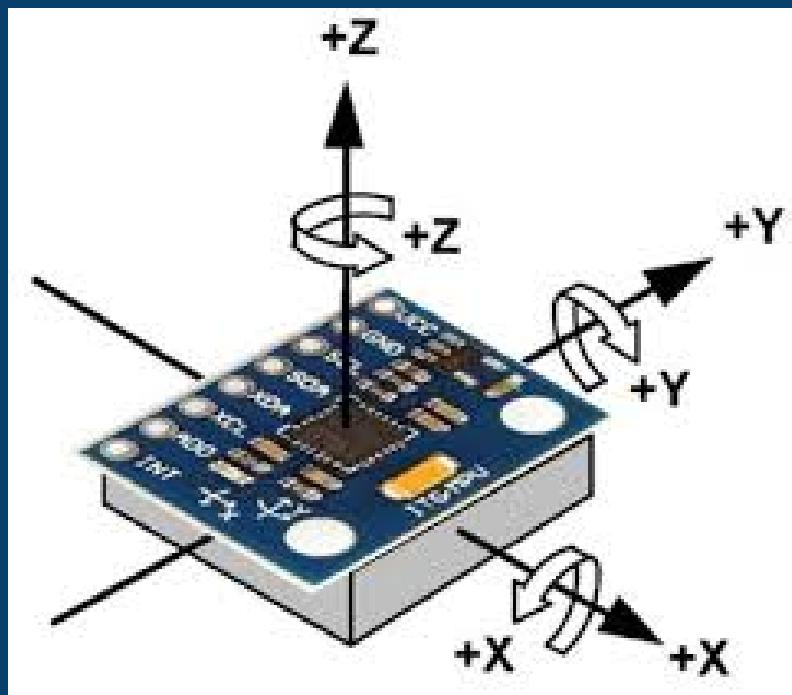
**Pitch**



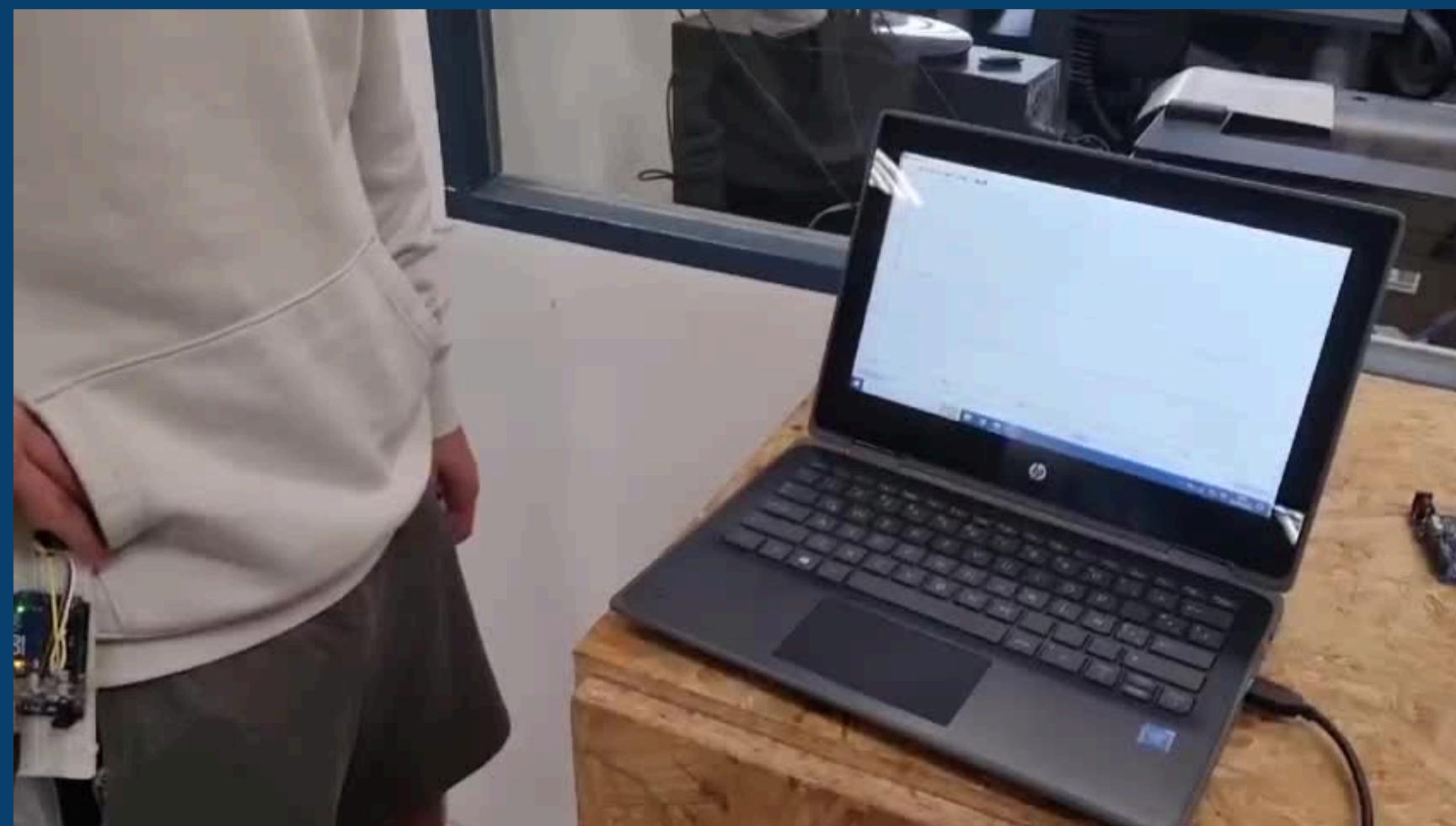
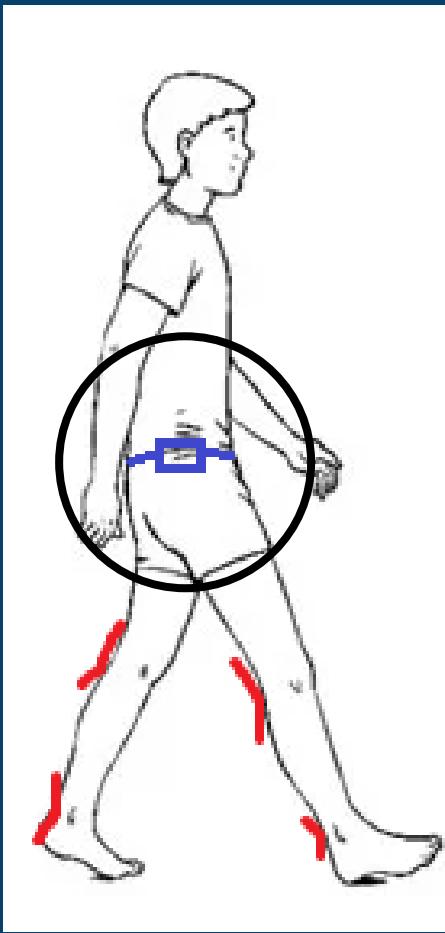
**Yaw**



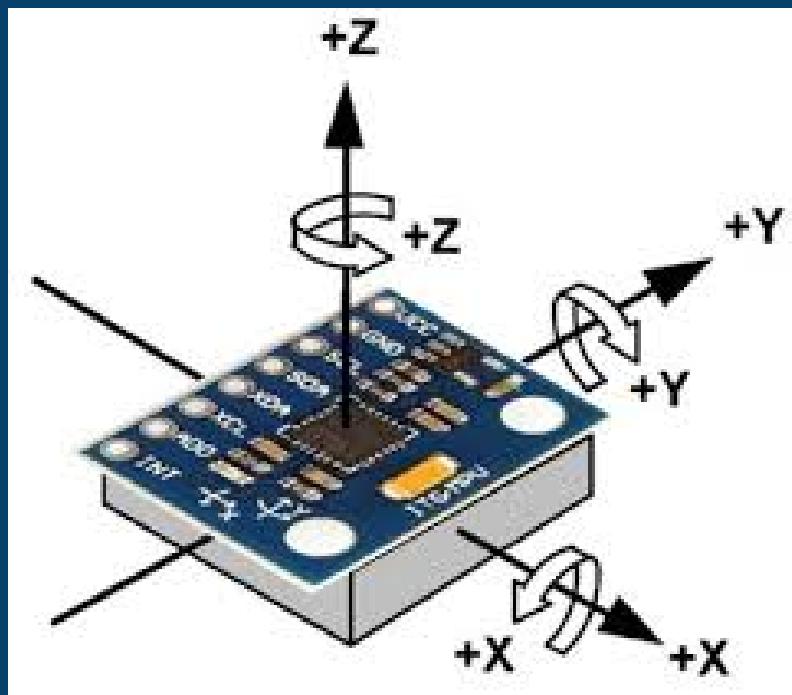
# Elección de componentes



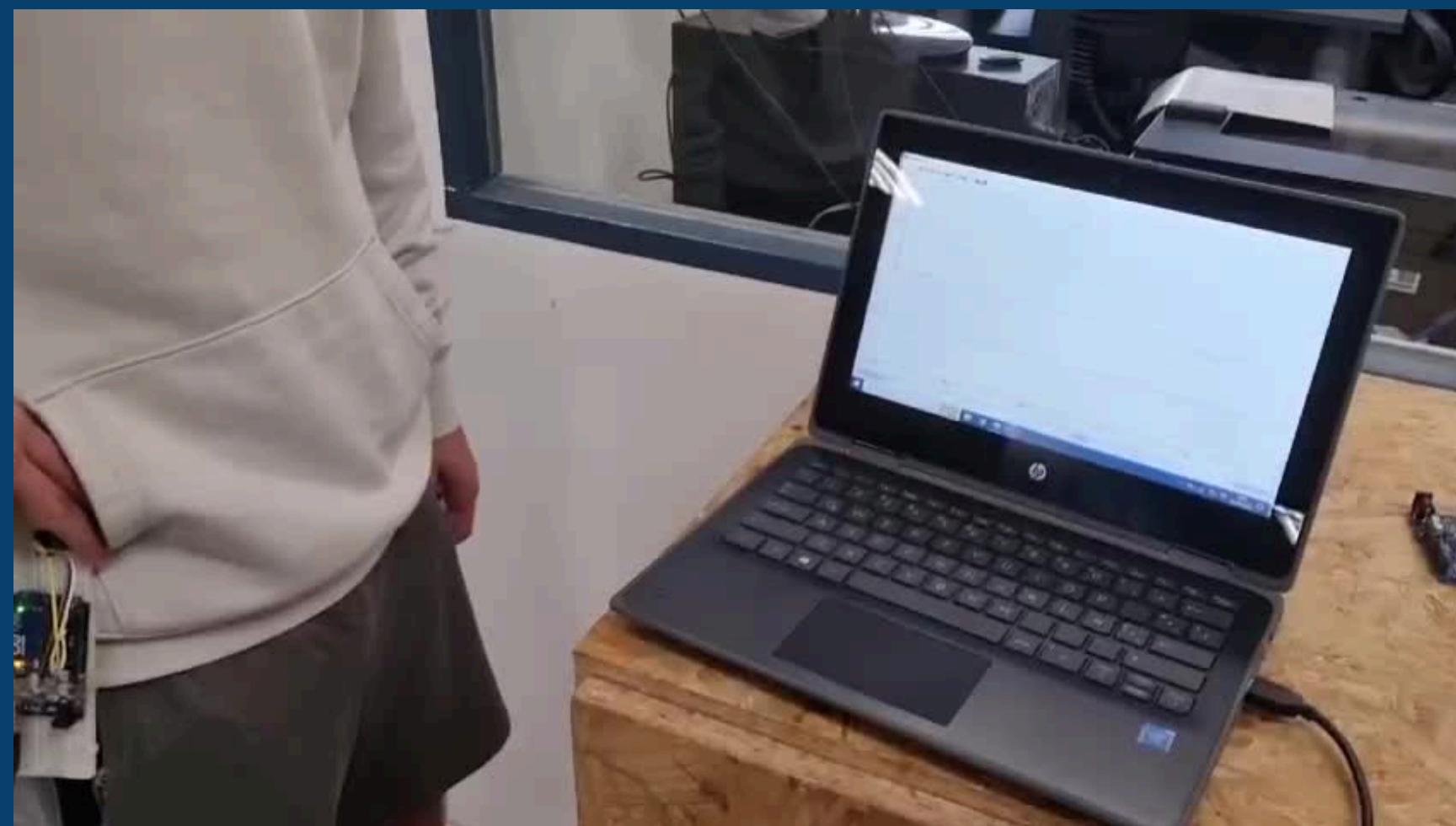
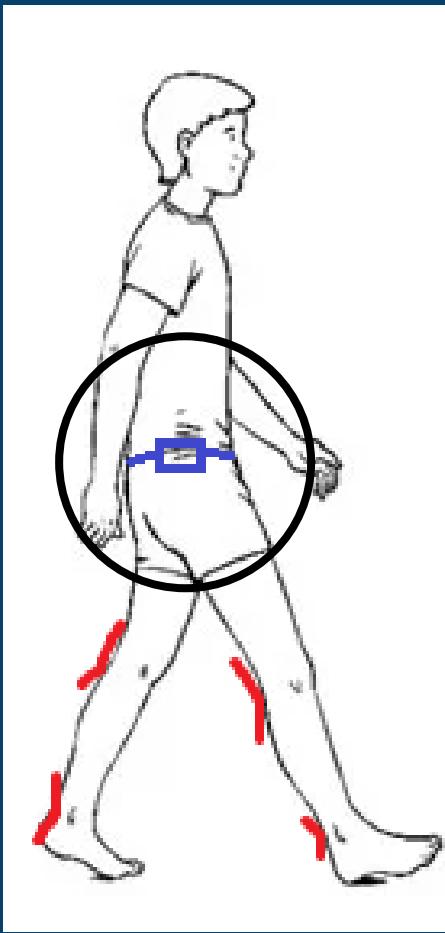
**mpu - 6050**



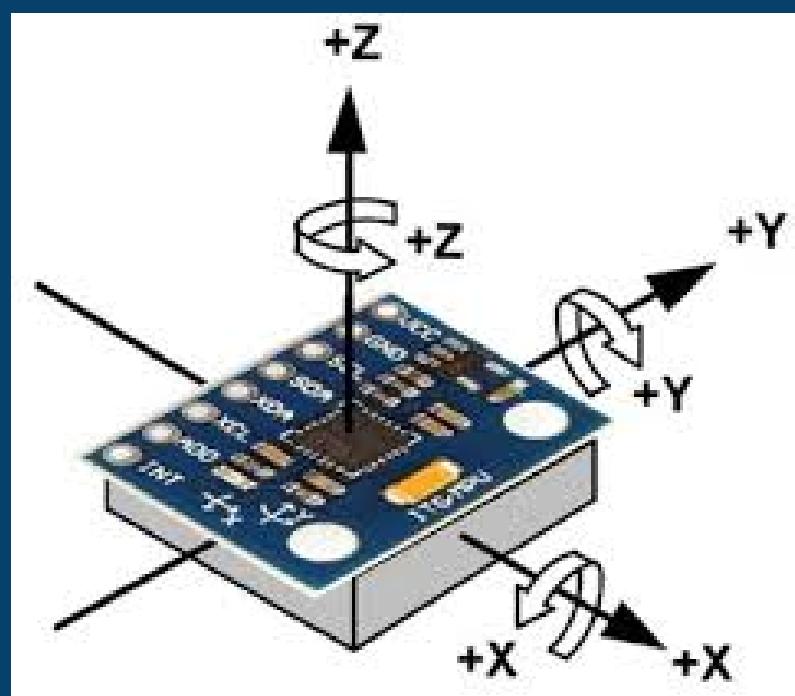
# Elección de componentes



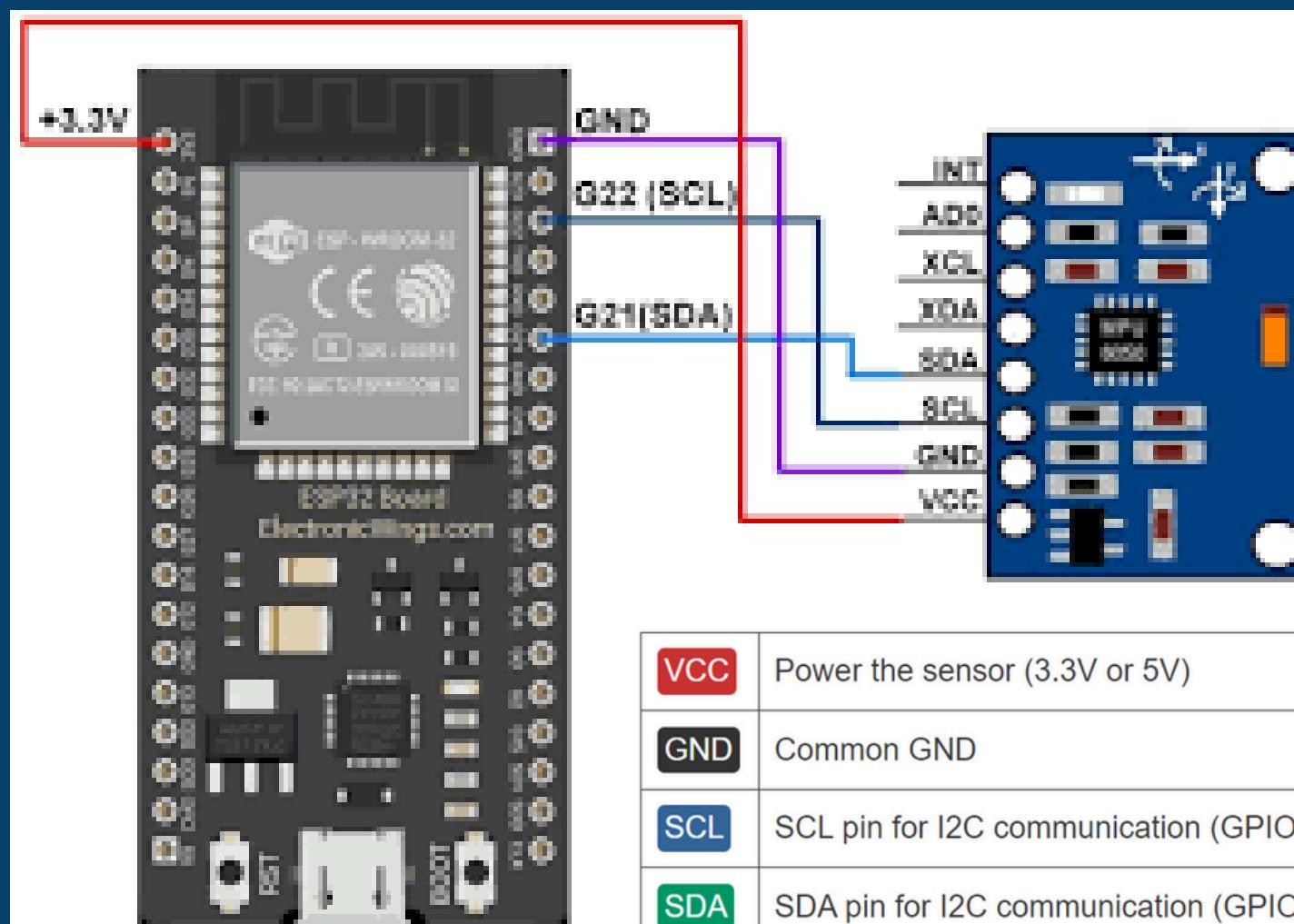
**mpu - 6050**



# Elección de componentes



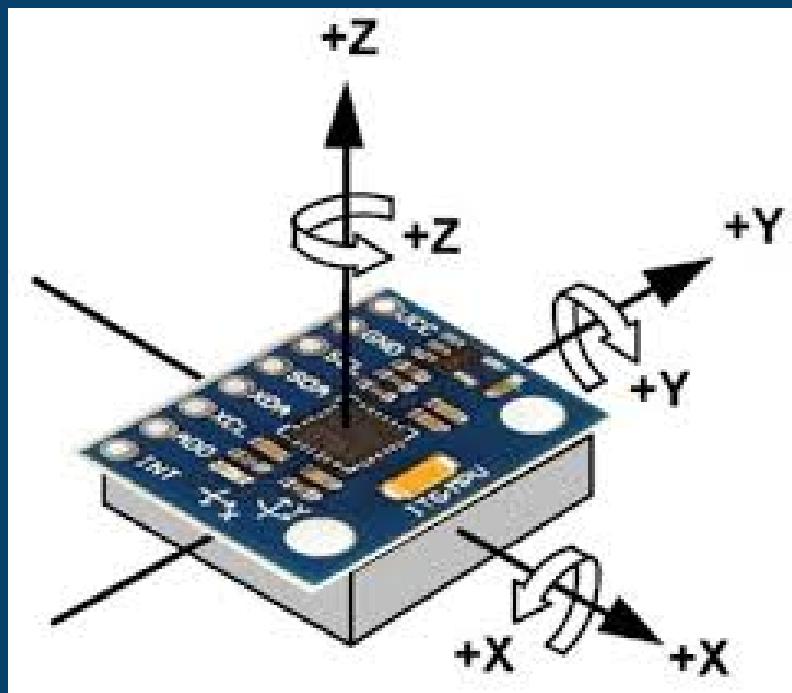
mpu - 6050



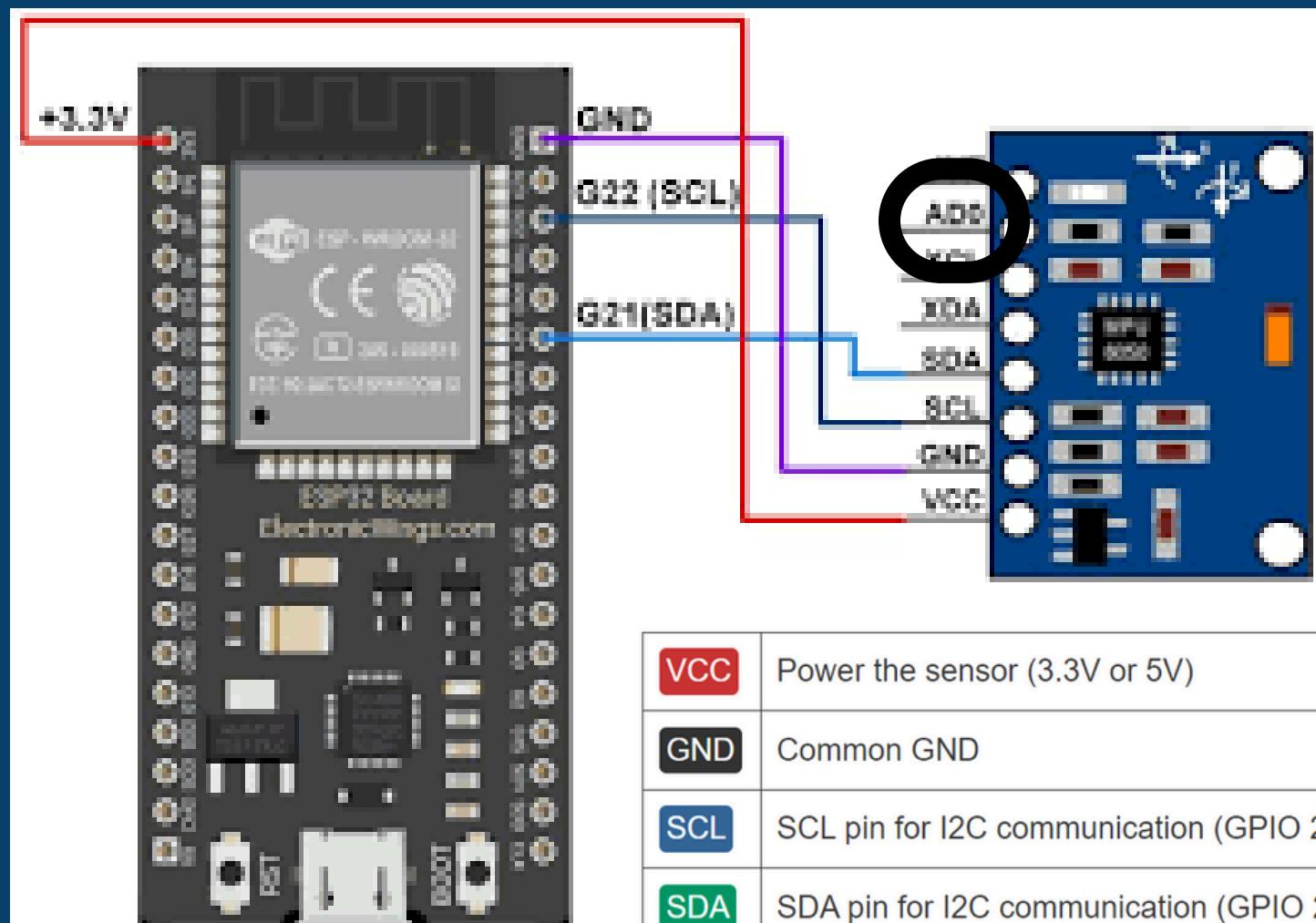
Conexión de 1 mpu

VCC	Power the sensor (3.3V or 5V)
GND	Common GND
SCL	SCL pin for I2C communication (GPIO 22)
SDA	SDA pin for I2C communication (GPIO 21)
XDA	Used to interface other I2C sensors with the MPU-6050
XCL	Used to interface other I2C sensors with the MPU-6050
AD0	Use this pin to change the I2C address
INT	Interrupt pin – can be used to indicate that new measurement data is available

# Elección de componentes



mpu - 6050



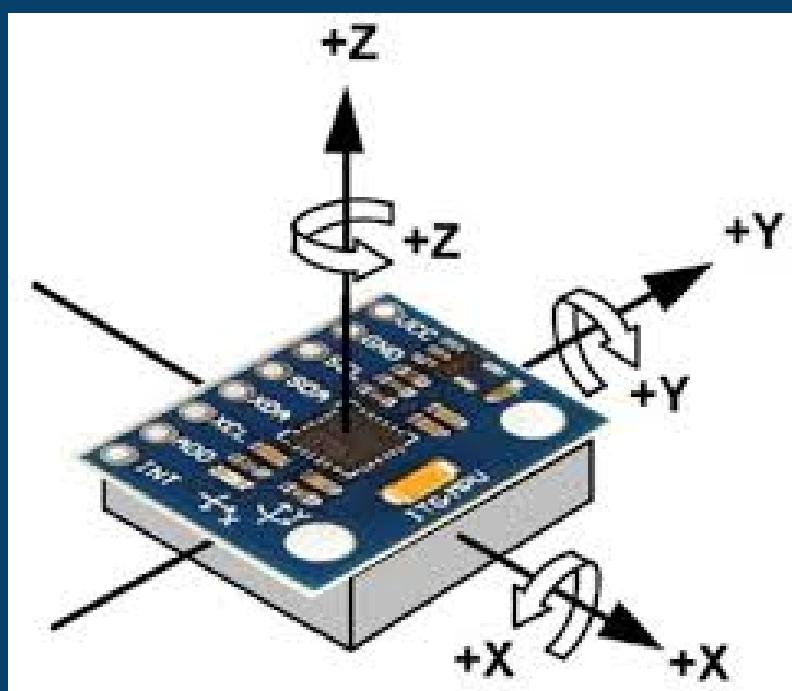
Pin AD0  
Comunicación I2C  
2 direcciones

VCC	Power the sensor (3.3V or 5V)
GND	Common GND
SCL	SCL pin for I2C communication (GPIO 22)
SDA	SDA pin for I2C communication (GPIO 21)
XDA	Used to interface other I2C sensors with the MPU-6050
XCL	Used to interface other I2C sensors with the MPU-6050
AD0	Use this pin to change the I2C address
INT	Interrupt pin – can be used to indicate that new measurement data is available

2 MPUS

# Elección de componentes

## Diferenciar direcciones I2C



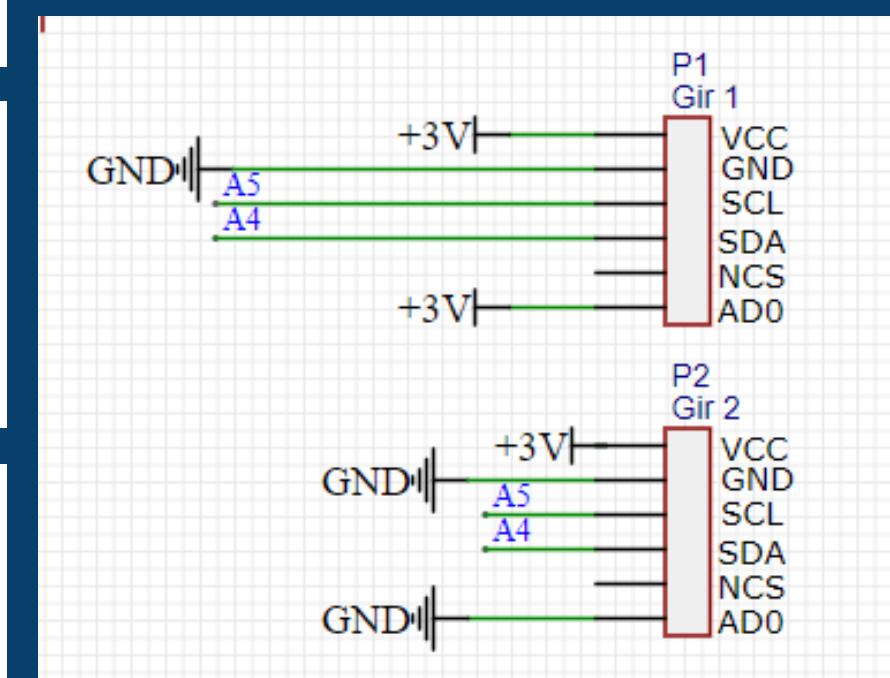
mpu - 6050

VCC	Power the sensor (3.3V or 5V)
GND	Common GND
SCL	SCL pin for I2C communication (GPIO 22)
SDA	SDA pin for I2C communication (GPIO 21)
XDA	Used to interface other I2C sensors with the MPU-6050
XCL	Used to interface other I2C sensors with the MPU-6050
<b>AD0</b>	<b>Use this pin to change the I2C address</b>
INT	Interrupt pin – can be used to indicate that new measurement data is available

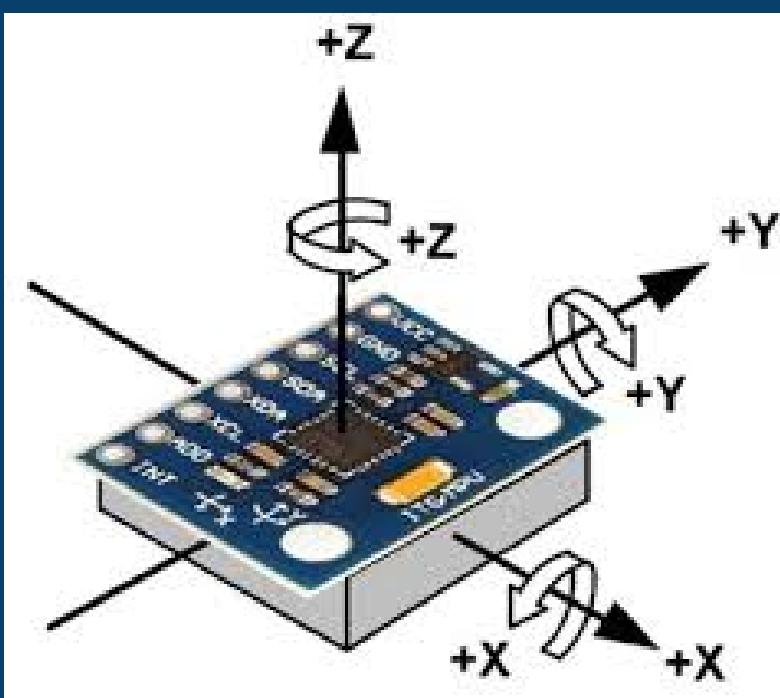
```
if (!mpu1.begin(0x69)) {  
    Serial.println("Failed to find MPU6050 chip");  
    while (1) {  
        delay(10);  
    }  
}  
  
if (!mpu2.begin(0x68)) {  
    Serial.println("Failed to find MPU6050 chip");  
    while (1) {  
        delay(10);  
    }  
}
```

```
if (!mpu1.begin(0x69)) {  
    Serial.println("Failed to find MPU6050 chip");  
    while (1) {  
        delay(10);  
    }  
}  
  
if (!mpu2.begin(0x68)) {  
    Serial.println("Failed to find MPU6050 chip");  
    while (1) {  
        delay(10);  
    }  
}
```

Pin AD0	Dirección I2C
AD0=HIGH (5V)	0x69
AD0=LOW (GND o NC)	0x68

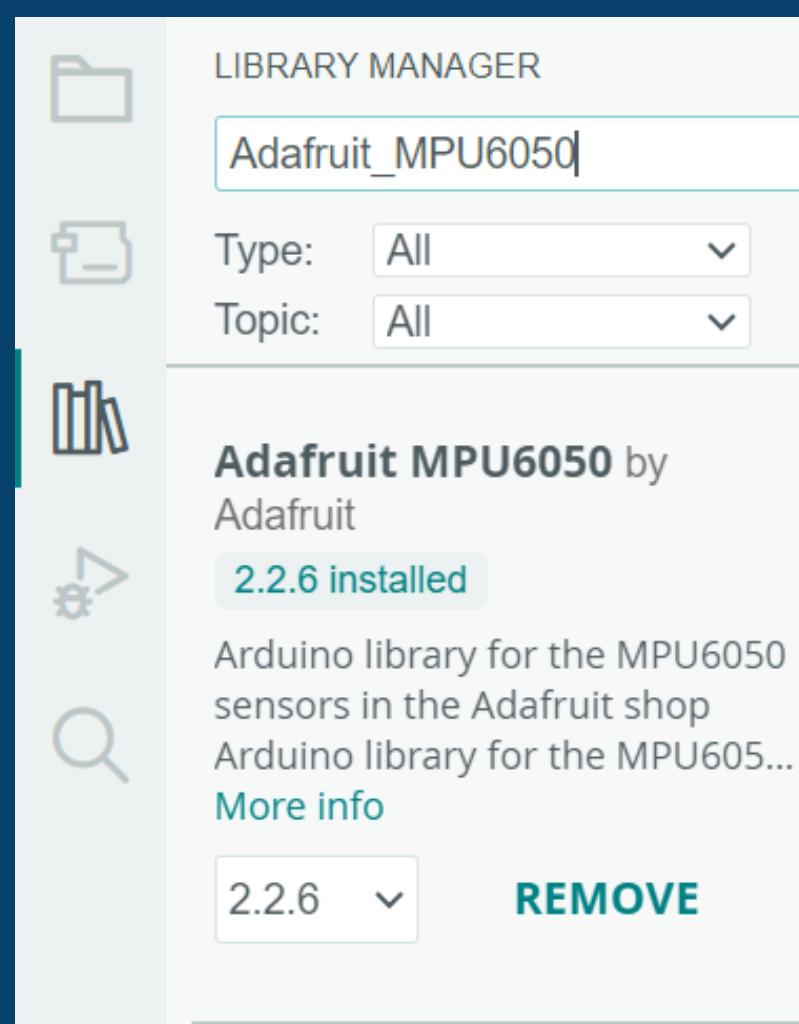


# Elección de componentes



**mpu - 6050**

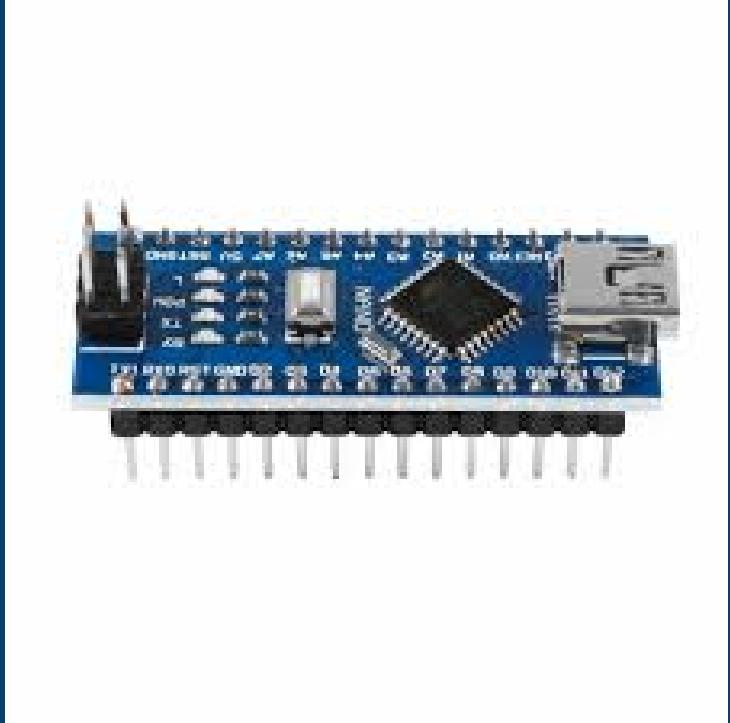
## Librería Adafruit\_MPU6050



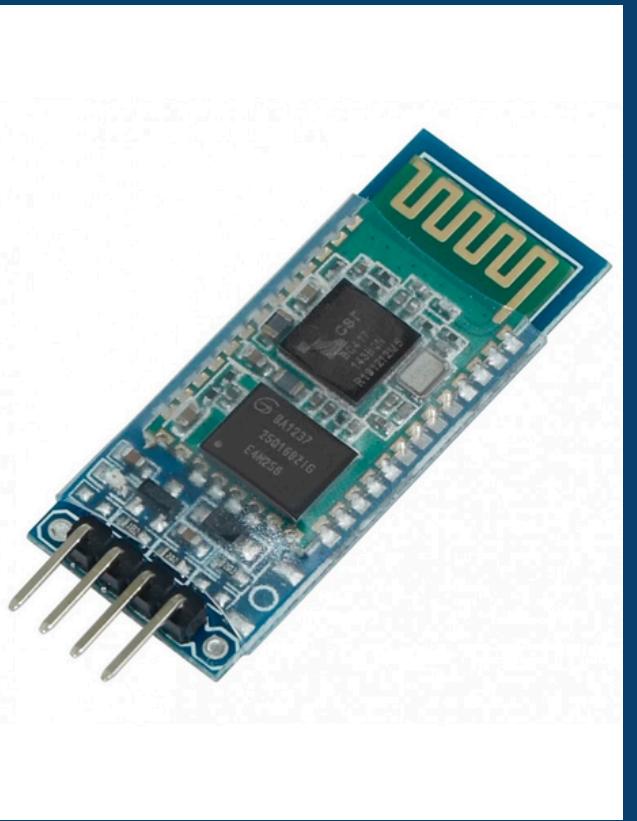
```
//LECTURA GIROSCOPIOS
sensors_event_t a, g, temp;
mpu1.getEvent(&a, &g, &temp);
x1 = g.orientation.roll
yUno = g.orientation.pitch;
z1 = g.orientation.heading;

mpu2.getEvent(&a, &g, &temp);
x2 = g.orientation.roll;
y2 = g.orientation.pitch;
z2 = g.orientation.heading;
```

# 2023

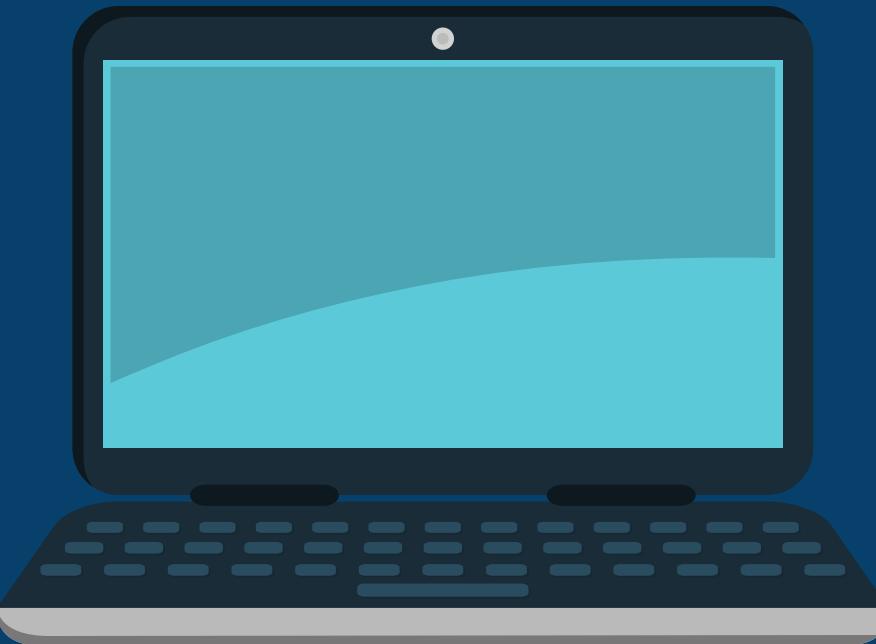


Arduino  
nano

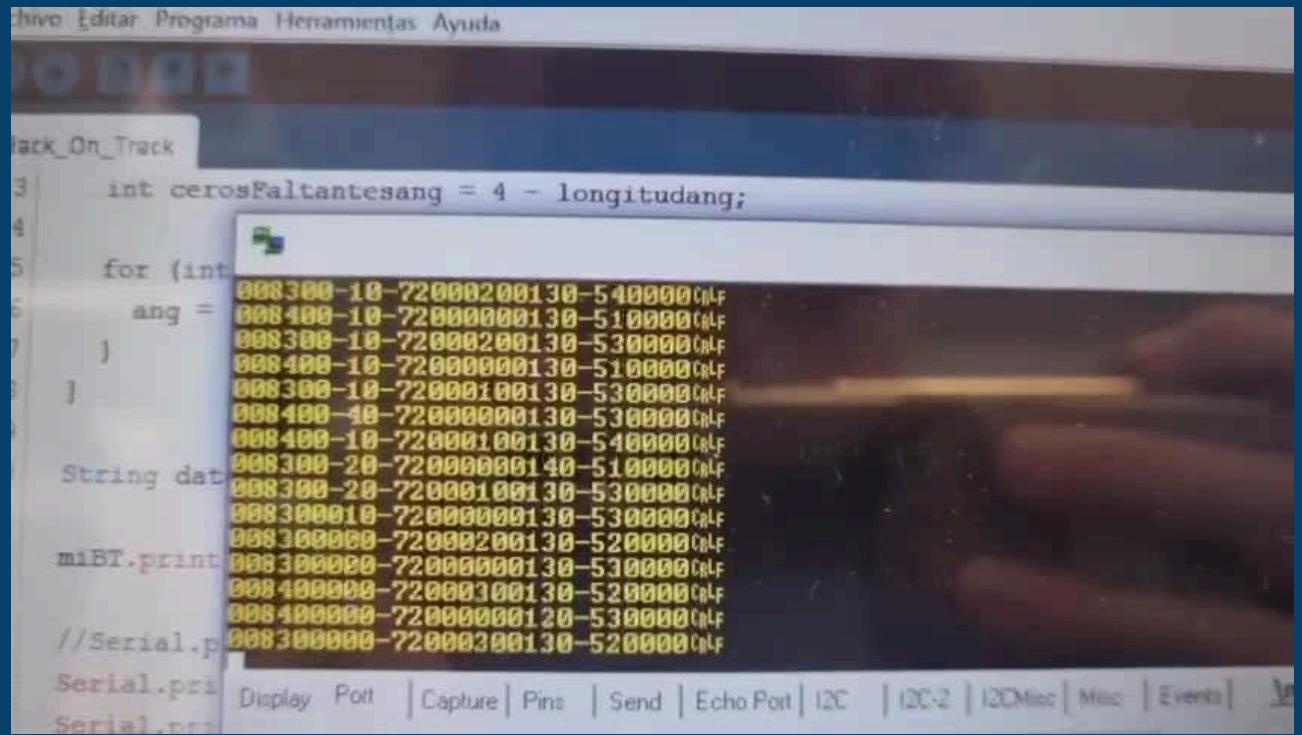


Modulo  
hc-05

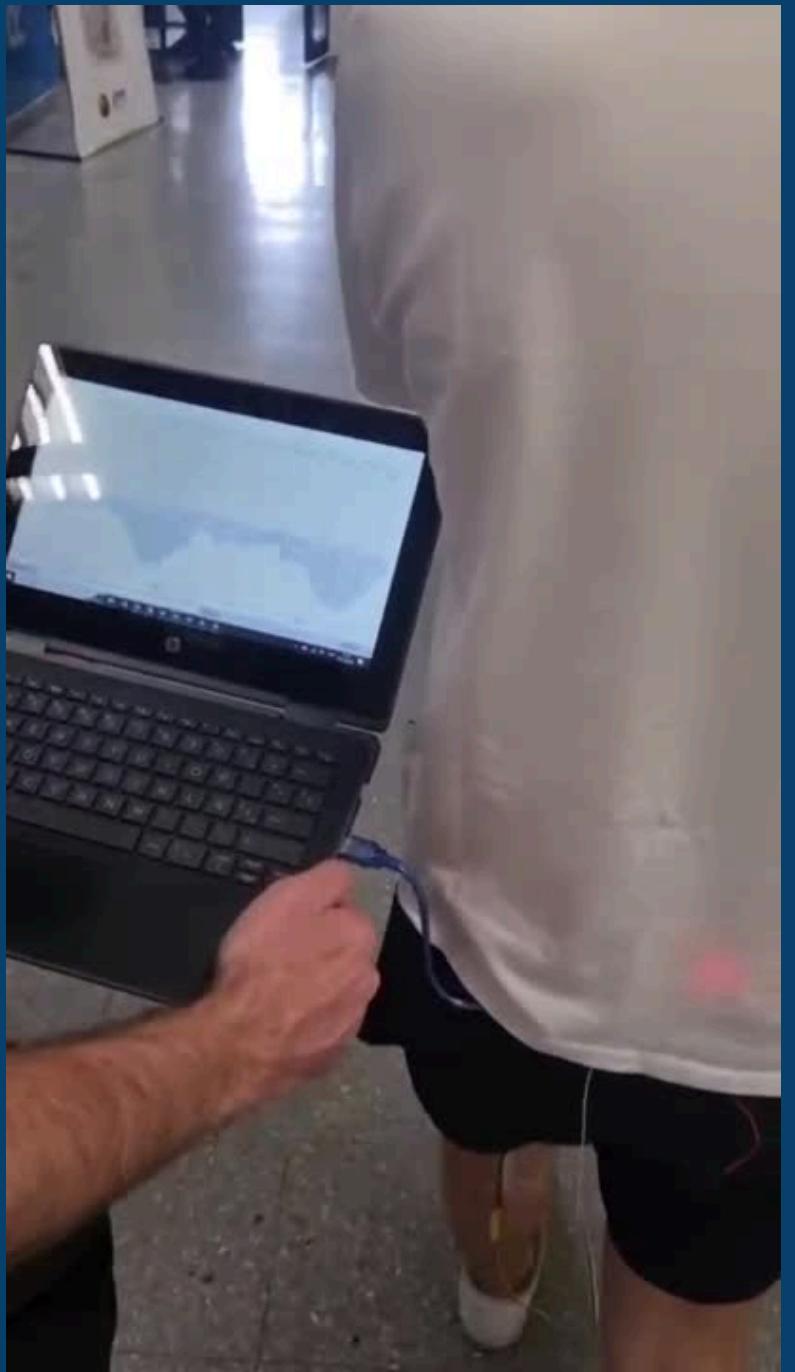
# Elección de componentes



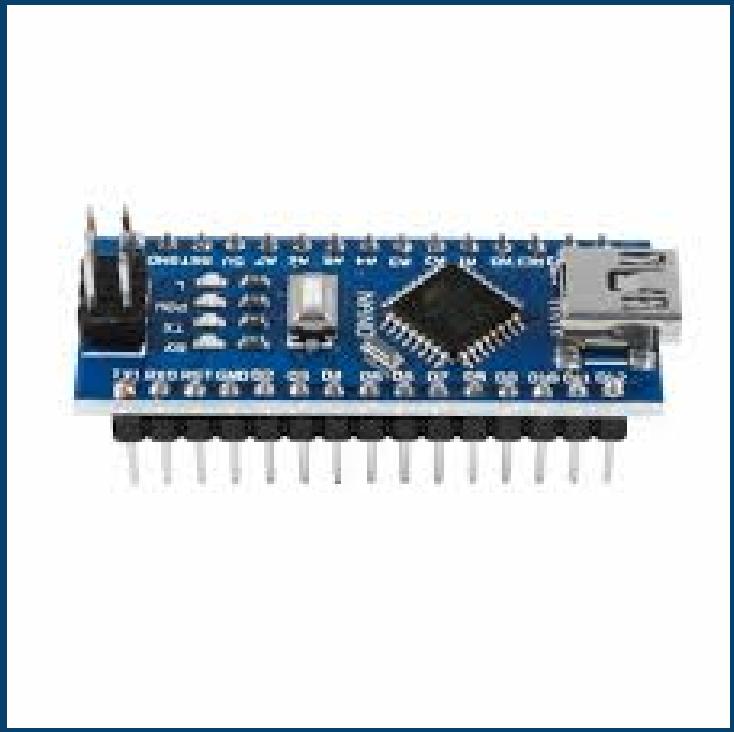
# Elección de componentes



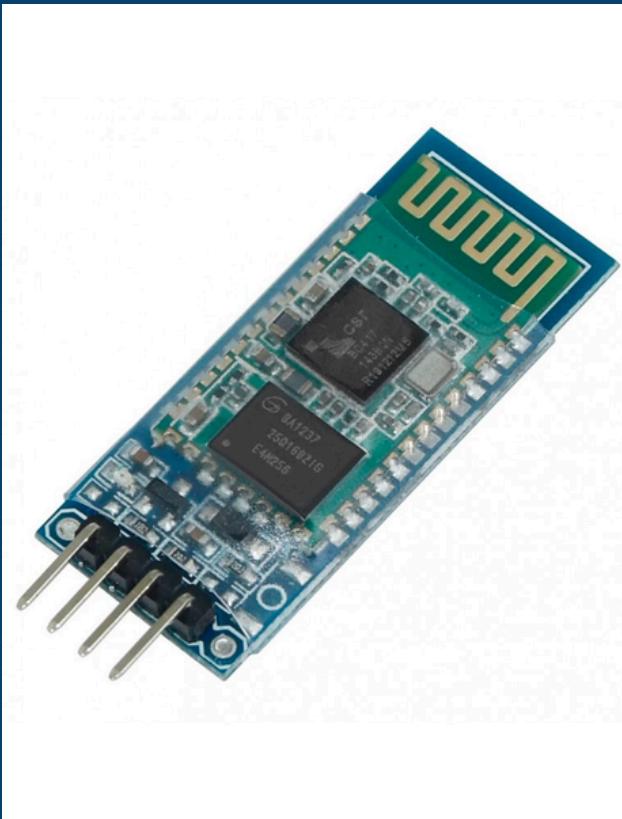
```
Back_On_Track
1
2
3 intcerosFaltantesang = 4 - longitudang;
4
5 for (int
6     ang =
7         008300-10-72000200130-5400000nLf
8         008400-10-72000000130-5100000nLf
9         008300-10-72000200130-5300000nLf
10        008400-10-72000000130-5100000nLf
11        008300-10-72000100130-5300000nLf
12        008400-10-72000000130-5300000nLf
13        008400-10-72000100130-5400000nLf
14
15 String dat
16 miBT.print
17 //Serial.p
18 Serial.pri
19 Serial.println
```



2023



Arduino  
nano

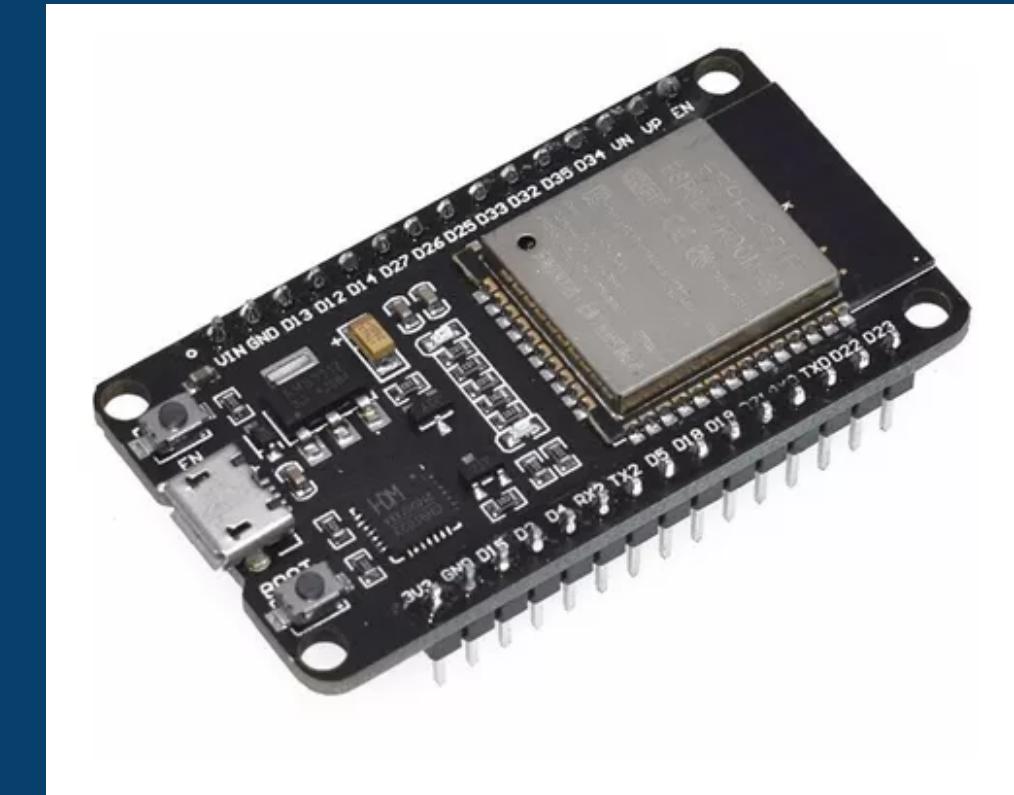


Modulo  
hc-05



# Elección de componentes

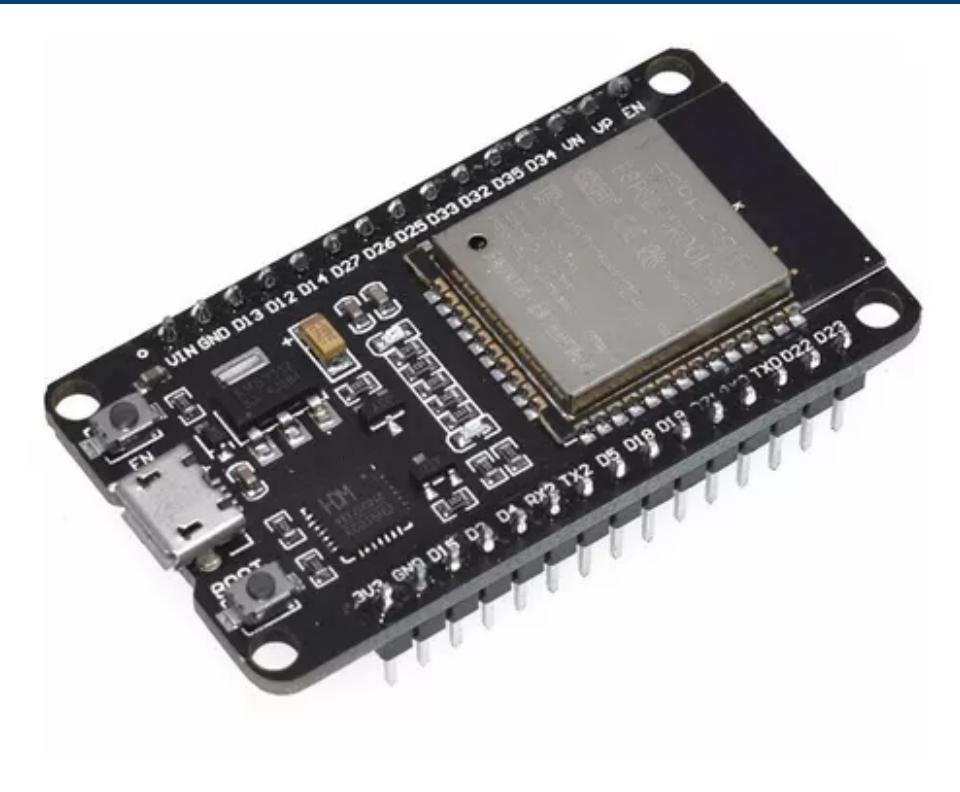
ESP32



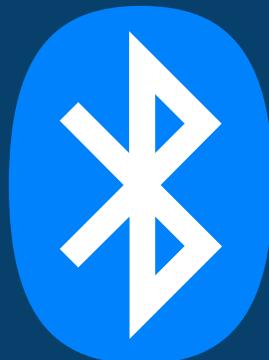
2024

# Elección de componentes

ESP32



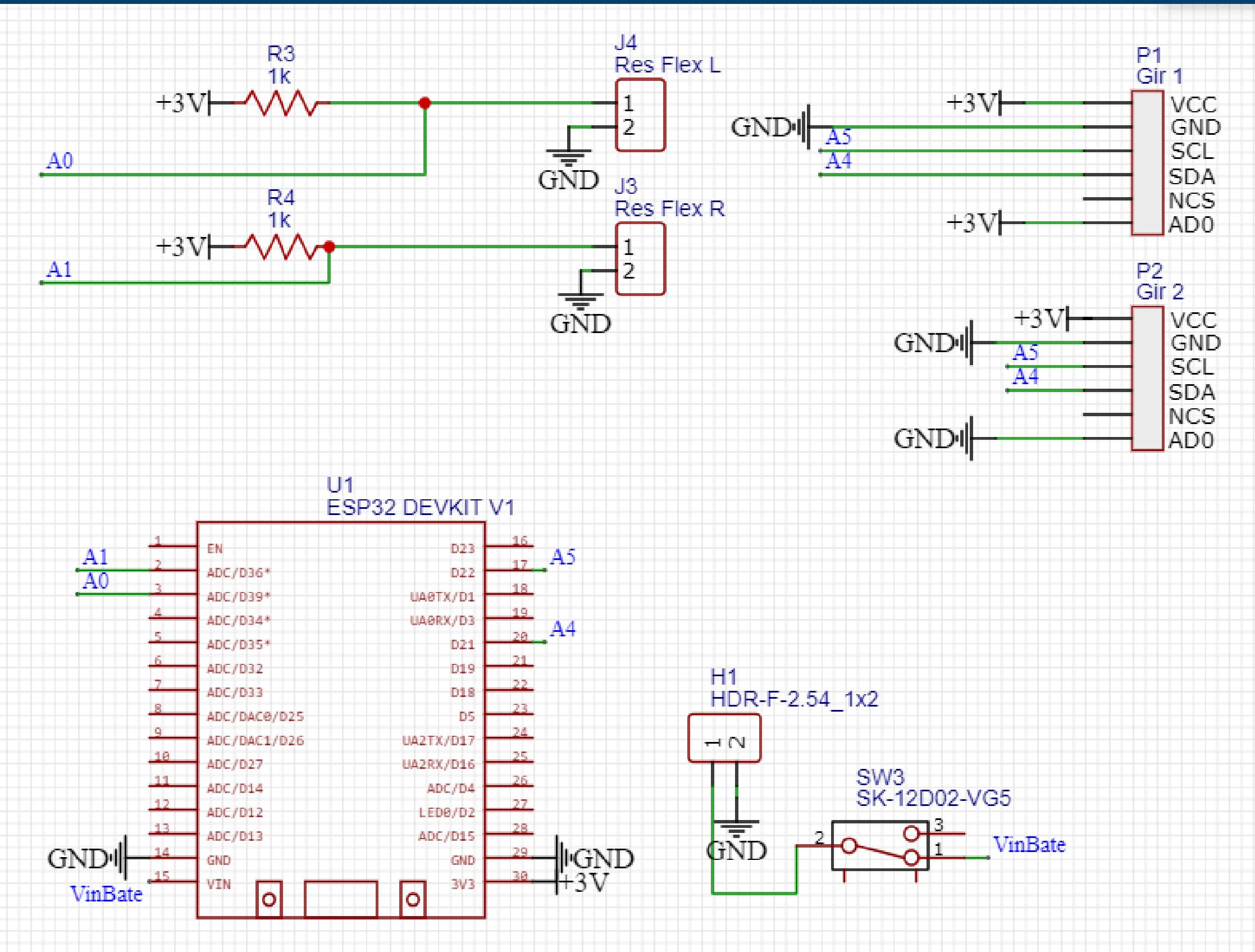
2024



Bluetooth

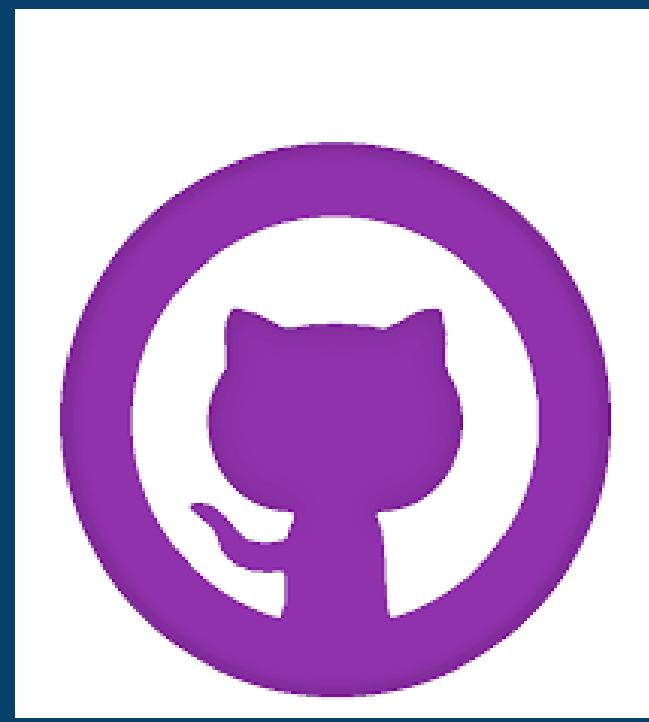


# Esquemático



# Programación de sensores

Código de prueba para  
cada sensor en particular



# Programación de sensores

A screenshot of a GitHub repository interface. At the top, there are navigation buttons for 'main' branch (with 1 Branch and 0 Tags), a search bar ('Go to file'), and buttons for 'Add file' and 'Code'. Below this is a list of recent commits by user 'Miapaysse'. The commits are:

Commit	Message	Time Ago
8c1cf5d · 1 minute ago	Miapaysse Subo codigos ordenados	3 Commits
BACK_ON_TRACK	Subo codigos ordenados	11 minutes ago
Firestore	Subo codigos ordenados	1 minute ago
Flexs	Subo codigos ordenados	1 minute ago
mpus	Subo codigos ordenados	1 minute ago

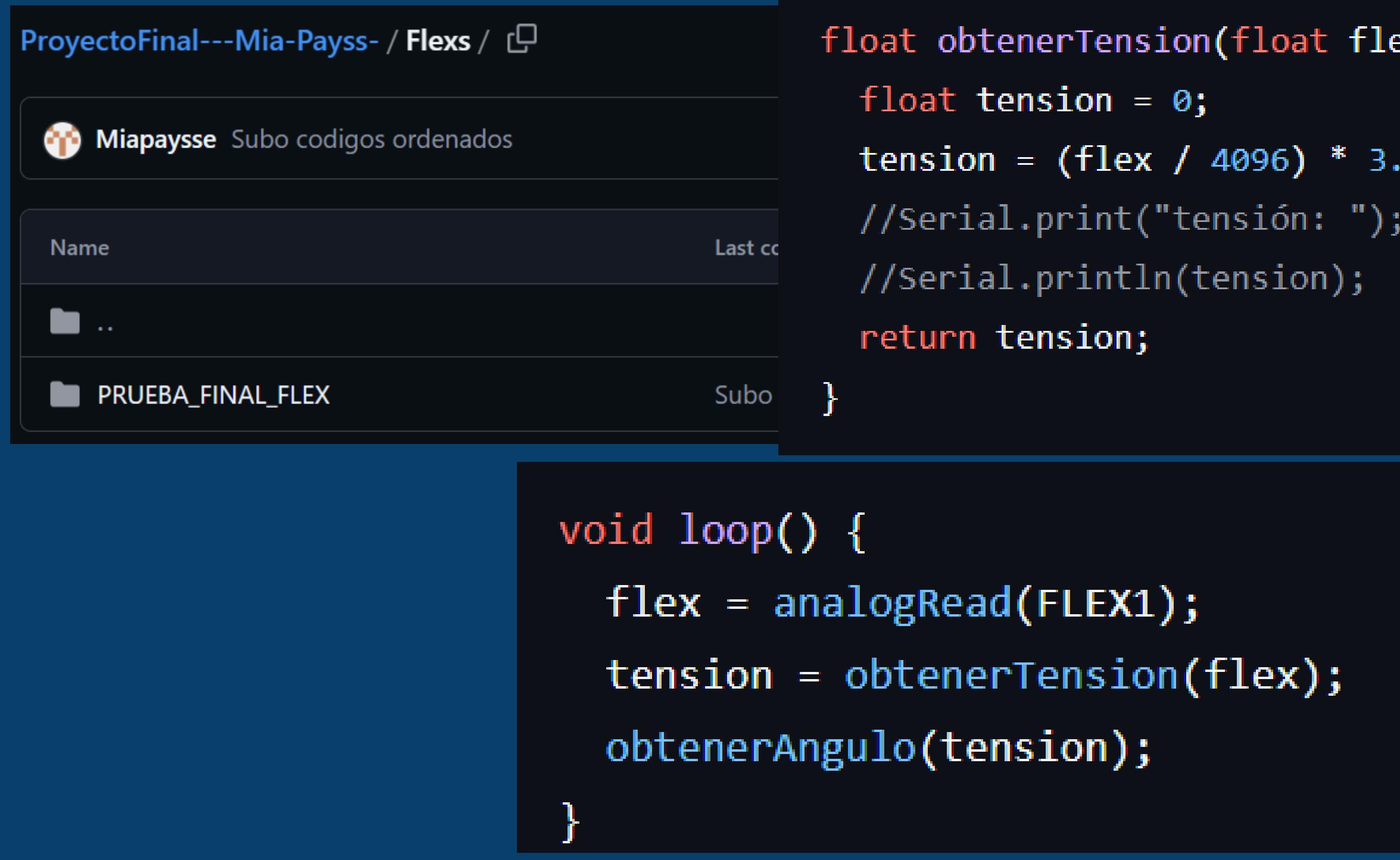
# Programación de sensores

ProyectoFinal---Mia-Payss- / Flexs / 

 **Miapaysse** Subo codigos ordenados 7605ab2 · now 

Name	Last commit message	Last commit date	
..			
PRUEBA_FINAL_FLEX	#define FLEX1 39 #define FLEX2 36	void setup() { pinMode(FLEX1, INPUT); pinMode(FLEX2, INPUT);  float tension = 0; float flex = 0; float angulo = 0;  float V_min = 0.5; float V_max = 1.25; }  void loop() { flex = analogRead(FLEX1); tension = obtenerTension(flex); obtenerAngulo(tension); }	

# Programación de sensores



ProyectoFinal---Mia-Payss- / Flexs / 

Miapaysse Subo codigos ordenados

Name	Last commit
..	
PRUEBA_FINAL_FLEX	Subo

```
float obtenerTension(float flex) {
    float tension = 0;
    tension = (flex / 4096) * 3.3;
    //Serial.print("tensión: ");
    //Serial.println(tension);
    return tension;
}

void loop() {
    flex = analogRead(FLEX1);
    tension = obtenerTension(flex);
    obtenerAngulo(tension);
}
```

file ...

History

```
float obtenerAngulo(float tension) {
    // Función sacada de excel: ((tension * (-90)) + 180);

    if (tension >= V_min && tension <= V_max) {
        // Cálculo del ángulo usando interpolación lineal
        angulo = (tension - V_min) / (V_max - V_min) * 180;
    } else {
        angulo = 0; // Valor fuera de rango o error
    }

    Serial.print("Angulo: ");
    Serial.println(angulo);

    return angulo;
}
```

# Programación de sensores

ProyectoFinal---Mia-Payss- / mpus / 

 **Miapaysse** Subo codigos ordenados 8c1cf5d · 21 minutes ago  History

Name	Last commit message	Last commit date
...		
PRUEBA_MP6050	Subo codigos ordenados	21 minutes ago
SCANER_IDI2C	Subo codigos ordenados	21 minutes ago

# Programación de sensores

The screenshot shows a GitHub repository interface for a project named "ProyectoFinal---Mia-Payss- / mpus /". The repository contains three files:

- Adafruit\_MPU6050.h**: Includes the Adafruit\_MPU6050 library and Adafruit\_Sensor library, along with the Wire library.
- Adafruit\_MPU6050.cpp**: Subo codigos ordenados (uploaded sorted codes). This file contains the setup and loop functions for the MPU6050 sensor. It initializes the sensor, prints a test message, and then enters a loop where it gets events from the sensor and prints acceleration and rotation values.
- SCANER\_IDI2C.h**: Subo codigos ordenados (uploaded sorted codes).

```
#include <Adafruit_MPU6050.h>
#include <Adafruit_Sensor.h>
#include <Wire.h>

Adafruit_MPU6050 mpu;

void setup(void) {
    Serial.begin(115200);
    while (!Serial)
        delay(10);

    Serial.println("Adafruit MPU6050 test!");

    // Try to initialize!
    if (!mpu.begin(0x68)) {
        Serial.println("Failed to find MPU6050 chip");
        while (1) {
            delay(10);
        }
    }
    Serial.println("MPU6050 Found!");
}

void loop() {
    /* Get new sensor events with the readings */
    sensors_event_t a, g, temp;
    mpu.getEvent(&a, &g, &temp);

    /* Print out the values */
    Serial.print("Acceleration X: ");
    Serial.print(a.acceleration.x);
    Serial.print(", Y: ");
    Serial.print(a.acceleration.y);
    Serial.print(", Z: ");
    Serial.print(a.acceleration.z);
    Serial.println(" m/s^2");

    Serial.print("Rotation X: ");
    Serial.print(g.gyro.x);
    Serial.print(", Y: ");
    Serial.print(g.gyro.y);
    Serial.print(", Z: ");
    Serial.print(g.gyro.z);
    Serial.println(" rad/s");
}
```

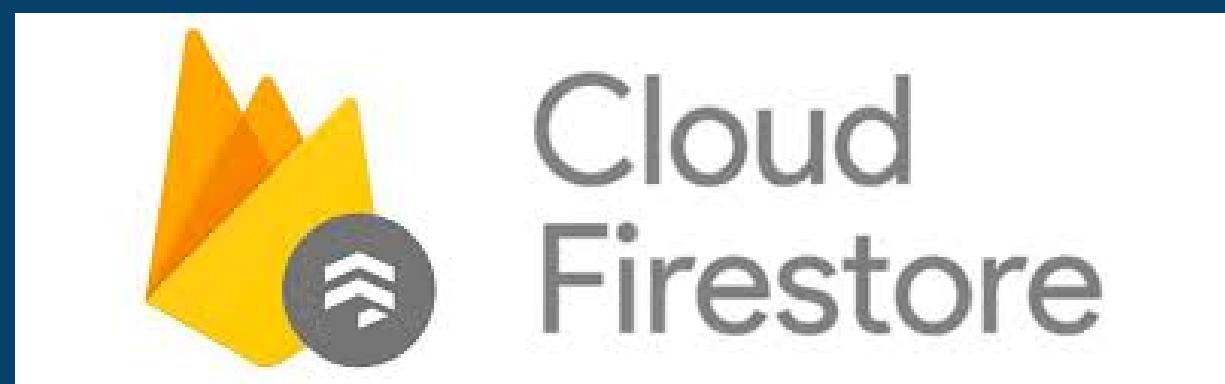
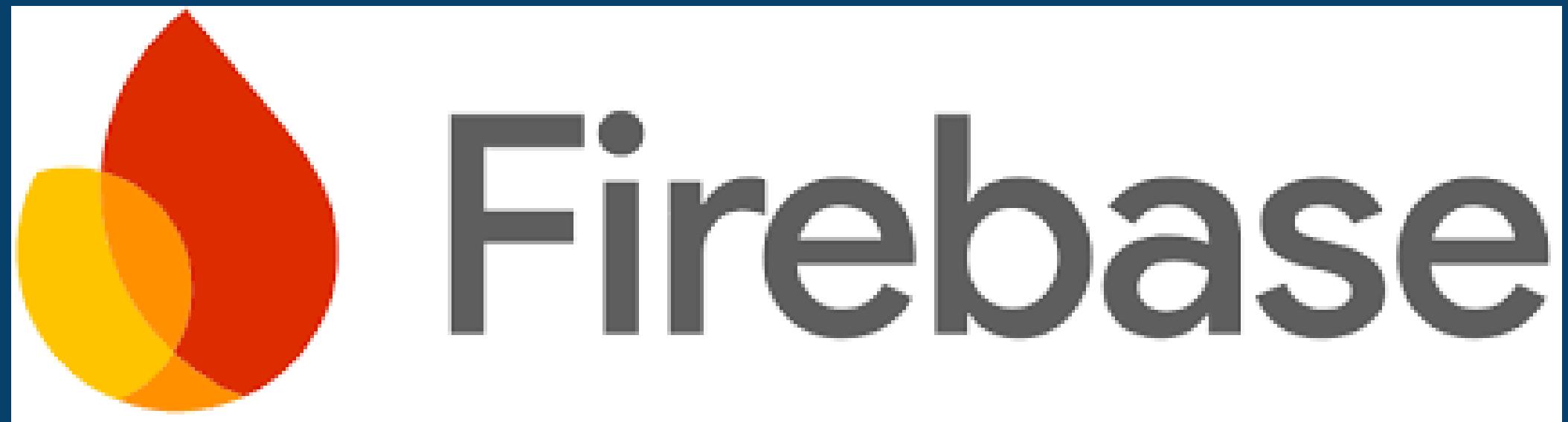
# Programación de sensores

ProyectoFinal---Mia-Payss- / mpus /

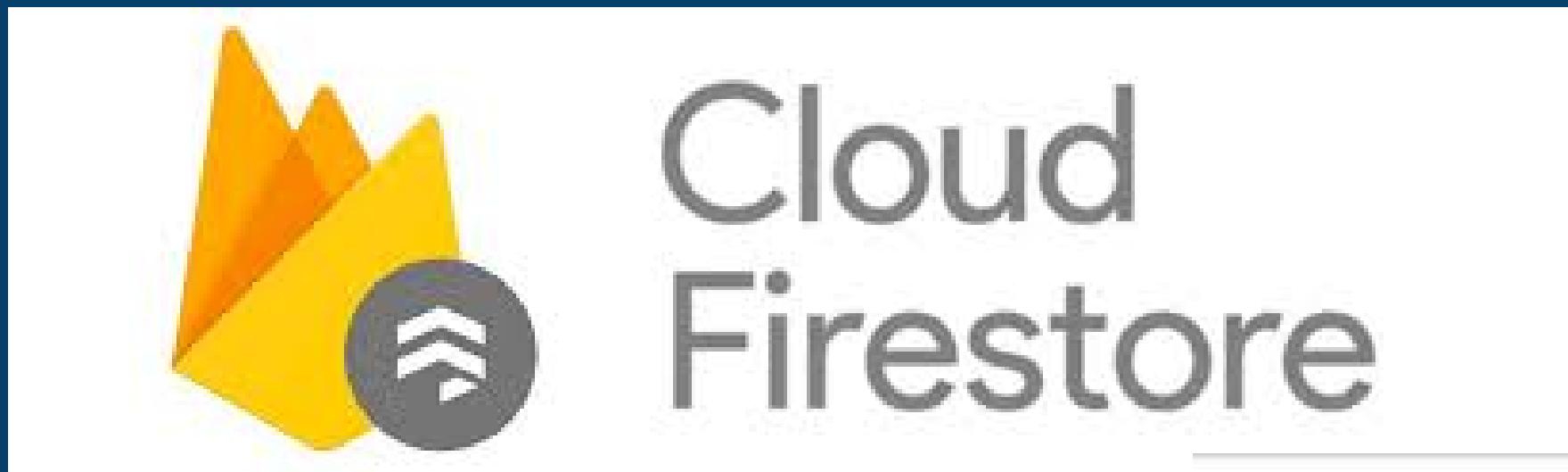
Miapaysse Subo codigos ordenados

Name	Last commit message
..	#include <Wire.h>
PRUEBA_MPU6050	void setup() { Wire.begin(); Serial.begin(115200); Serial.println("\nI2C Scanner"); }
SCANER_IDI2C	void loop() { byte error, address; int nDevices; Serial.println("Scanning..."); nDevices = 0; for(address = 1; address < 127; address++ ) { Wire.beginTransmission(address); error = Wire.endTransmission(); if (error == 0) { Serial.print("I2C device found at address 0x"); if (address<16) { Serial.print("0"); } Serial.println(address,HEX); nDevices++; } else if (error==4) { Serial.print("Unknow error at address 0x"); if (address<16) { Serial.print("0"); } Serial.println(address,HEX); } } if (nDevices == 0) { Serial.println("No I2C devices found\n"); } else { Serial.println("done\n"); } delay(5000); }

# Conexión con la base de datos



# Conexión con la base de datos



Poca documentación en relación a la interfaz con arduino



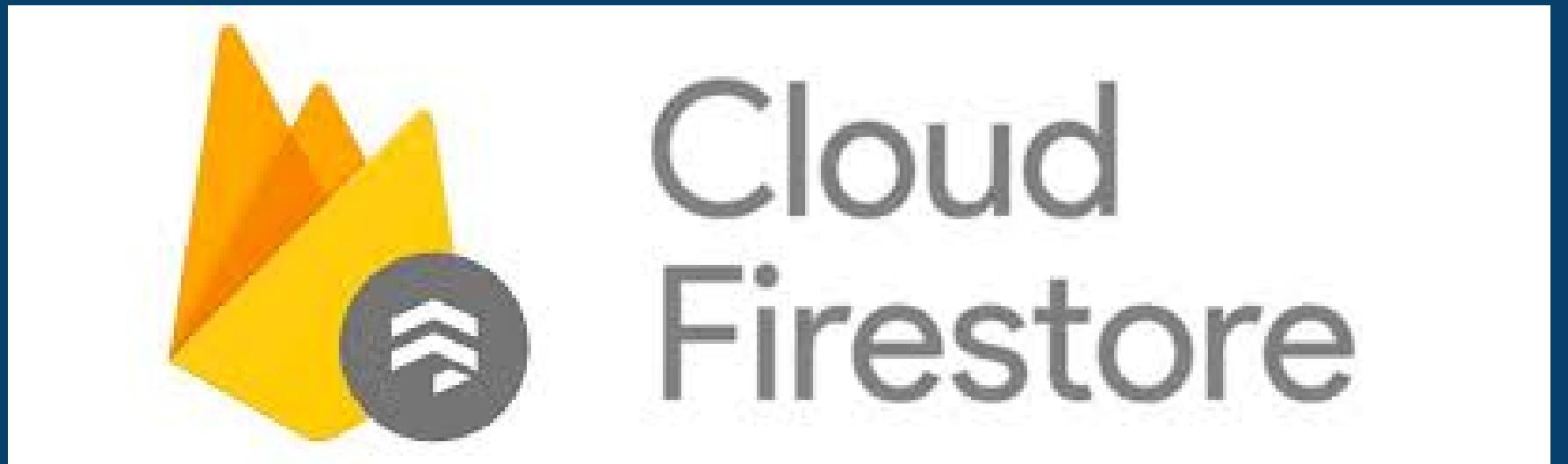
Data

Cloud Firestore in Native mode ? Database location: nam5 (us-central)

/ > order\_history > 416cda47-cb46-435c-9c9a-15527d76343f edit

Root	order_history		416cda47-cb46-435c-9c9a-15527d76343f
+ START COLLECTION	+ ADD DOCUMENT		+ START COLLECTION
: order_history	:	416cda47-cb46-435c-9c9a-15527d76343f	+ ADD FIELD
		edc06d20-e502-40ab-937b-1e66a1c94892	currency: "EUR"
			iso_time: "2022-02-10T09:21:27.718Z"
			items <span>(array)</span> + <span>edit</span>

# Conexión con la base de datos

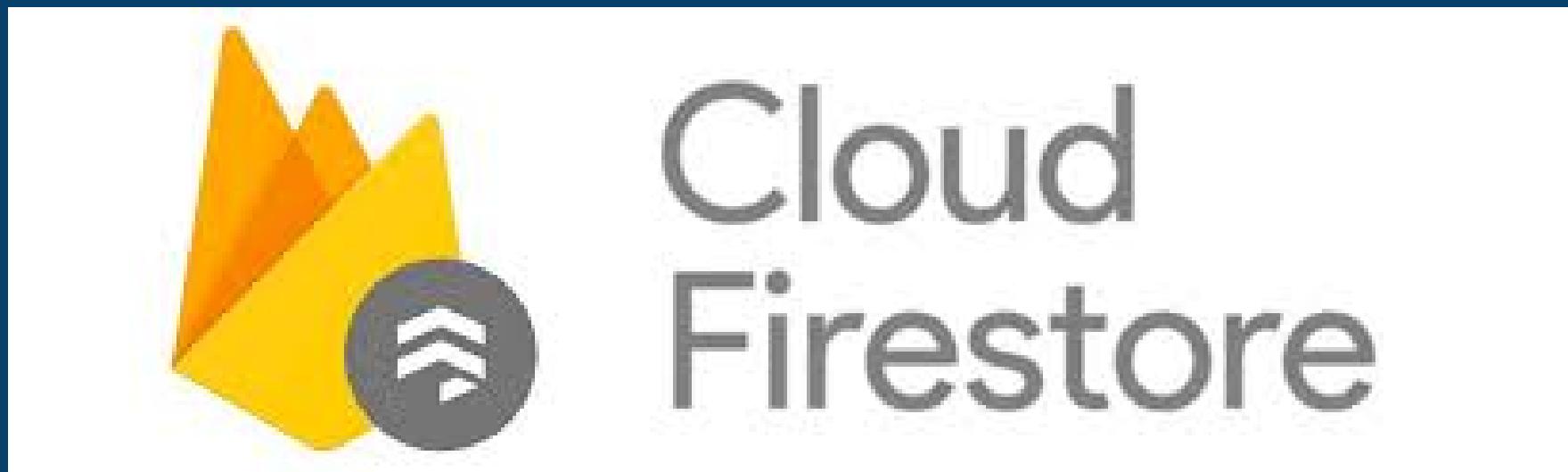


## Librería: Firebase Arduino Client Library for ESP8266 and ESP32

```
crearDoc("tracking/" + idPersona);
```

```
update("tracking/" + idPersona, "valoresFlex1", anguloFlex1, count1);
update("tracking/" + idPersona, "valoresFlex2", anguloFlex2, count2);
update("tracking/" + idPersona, "valoresX1", x1, count3);
update("tracking/" + idPersona, "valoresy1", yUno, count4);
update("tracking/" + idPersona, "valoresz1", z1, count5);
update("tracking/" + idPersona, "valoresX2", x2, count6);
update("tracking/" + idPersona, "valoresy2", y2, count7);
update("tracking/" + idPersona, "valoresz2", z2, count8);
```

# Conexión con la base de datos



Librería: Firebase Arduino Client Library for ESP8266 and ESP32

```
// For the usage of FirebaseJson, see examples/FirebaseJson/BasicUsage/Create_Edit_Parse/Create_Edit_Parse.ino
FirebaseJson content;

String documentPath = path;
// If the document path contains space e.g. "a b c/d e f"
// It should encode the space as %20 then the path will be "a%20b%20c/d%20e%20f"
content.set("fields/valoresFlex1/stringValue", "Inicio");

String doc_path = "projects/";
doc_path += FIREBASE_PROJECT_ID;
doc_path += "/databases/(default)/documents/coll_id/doc_id"; // coll_id and doc_id are your collection id and document id

Serial.print("Create a document... ");

if (Firebase.Firestore.createDocument(&fbdo, FIREBASE_PROJECT_ID, "" /* databaseId can be (default) or empty */, documentPath.c_str(), content.raw()))
| Serial.printf("ok\n%s\n\n", fbdo.payload().c_str());
else {
| Serial.println(fbdo.errorReason());
}

return;
```

```
crearDoc("tracking/" + idPersona);

void crearDoc(String path)
```

# Conexión con la base de datos

Cloud  
Firestore



```
contador = contador + 1;

// The dynamic array of write object firebase_firestore_document_write_t.
std::vector<struct firebase_firestore_document_write_t> writes;

// Write object that will be written to the document.
struct firebase_firestore_document_write_t transform_write;
transform_write.type = firebase_firestore_document_write_type_transform;

// Set the document path of document to write (transform)
transform_write.document_transform.document_path = path1;

// Set a transformation of a field of the document.
struct firebase_firestore_document_write_field_transforms_t field_transforms;
// Set field path to write.
field_transforms.fieldPath = path2;

field_transforms.transform_type = firebase_firestore_transform_type_append_missing_elements;

// For the usage of FirebaseJson, see examples/FirebaseJson/BasicUsage/Create_Edit_Parse/Create_Edit_Parse.ino
FirebaseJson content;

String muestraMedicion = String(contador) + ", " + String(medicion);
content.set("values/[1]/stringValue", muestraMedicion);
// timestamp
//content.set("values/[1]/timestampValue", "2014-10-02T15:01:23Z");

// Set the transformation content.
field_transforms.transform_content = content.raw();

// Add a field transformation object to a write object.
transform_write.document_transform.field_transforms.push_back(field_transforms);

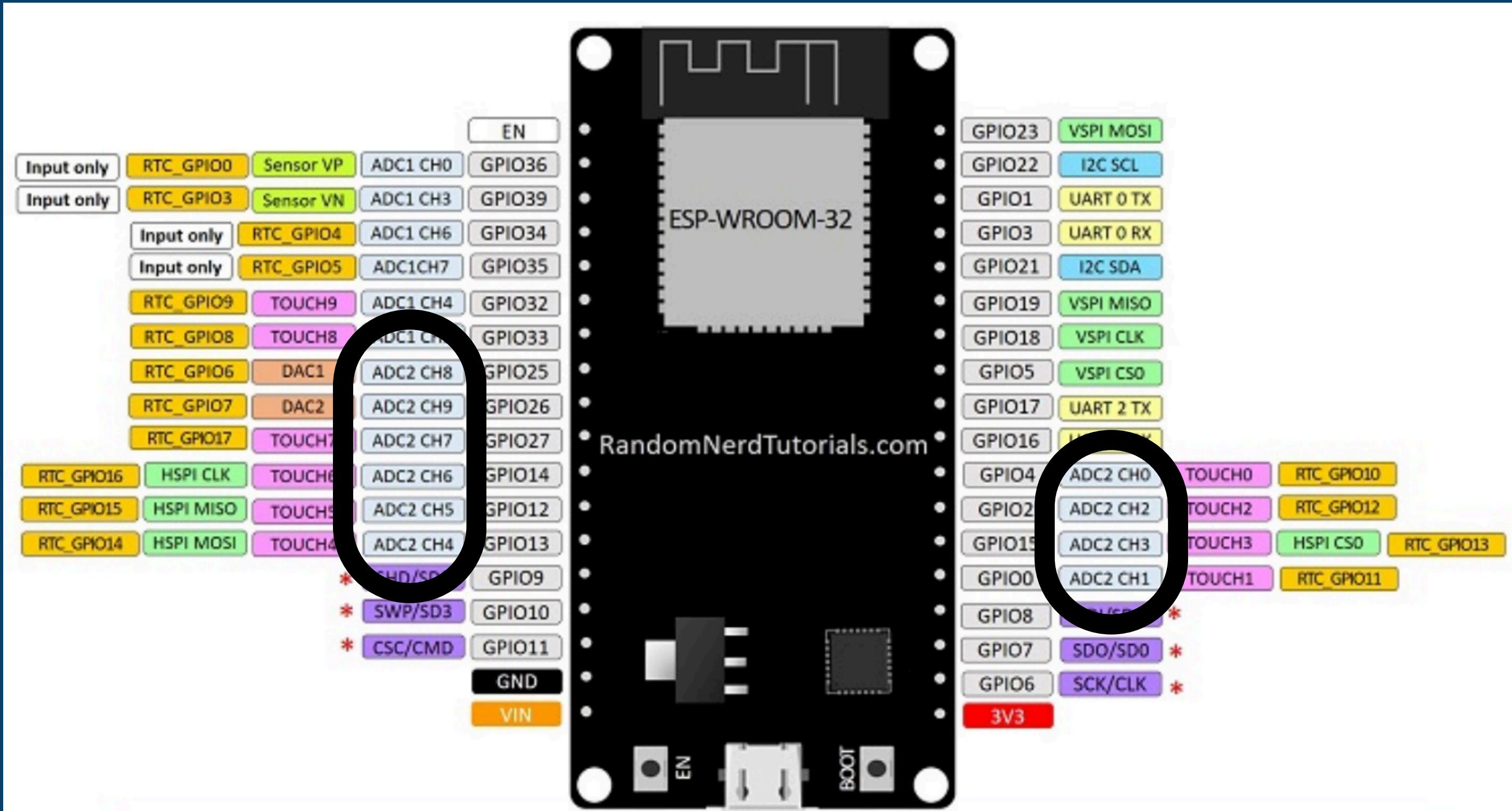
// Add a write object to a write array.
writes.push_back(transform_write);

if (Firebase.Firestore.commitDocument(&fbdo, FIREBASE_PROJECT_ID, "" /* databaseId can be (default) or empty */, writes /* dynamic array of firebase_firestore_document_write_t */, "" /* transaction */))
| Serial.printf("ok\n%s\n\n", fbdo.payload().c_str());
else
Serial.println(fbdo.errorReason());
```

void update(String path1, String path2, float medicion, int& contador)

```
update("tracking/" + idPersona, "valoresFlex1", anguloFlex1, count1);
update("tracking/" + idPersona, "valoresFlex2", anguloFlex2, count2);
update("tracking/" + idPersona, "valoresX1", x1, count3);
update("tracking/" + idPersona, "valoresy1", yUno, count4);
update("tracking/" + idPersona, "valoresz1", z1, count5);
update("tracking/" + idPersona, "valoresX2", x2, count6);
update("tracking/" + idPersona, "valoresy2", y2, count7);
update("tracking/" + idPersona, "valoresz2", z2, count8);
```

# Problemas



ADC2

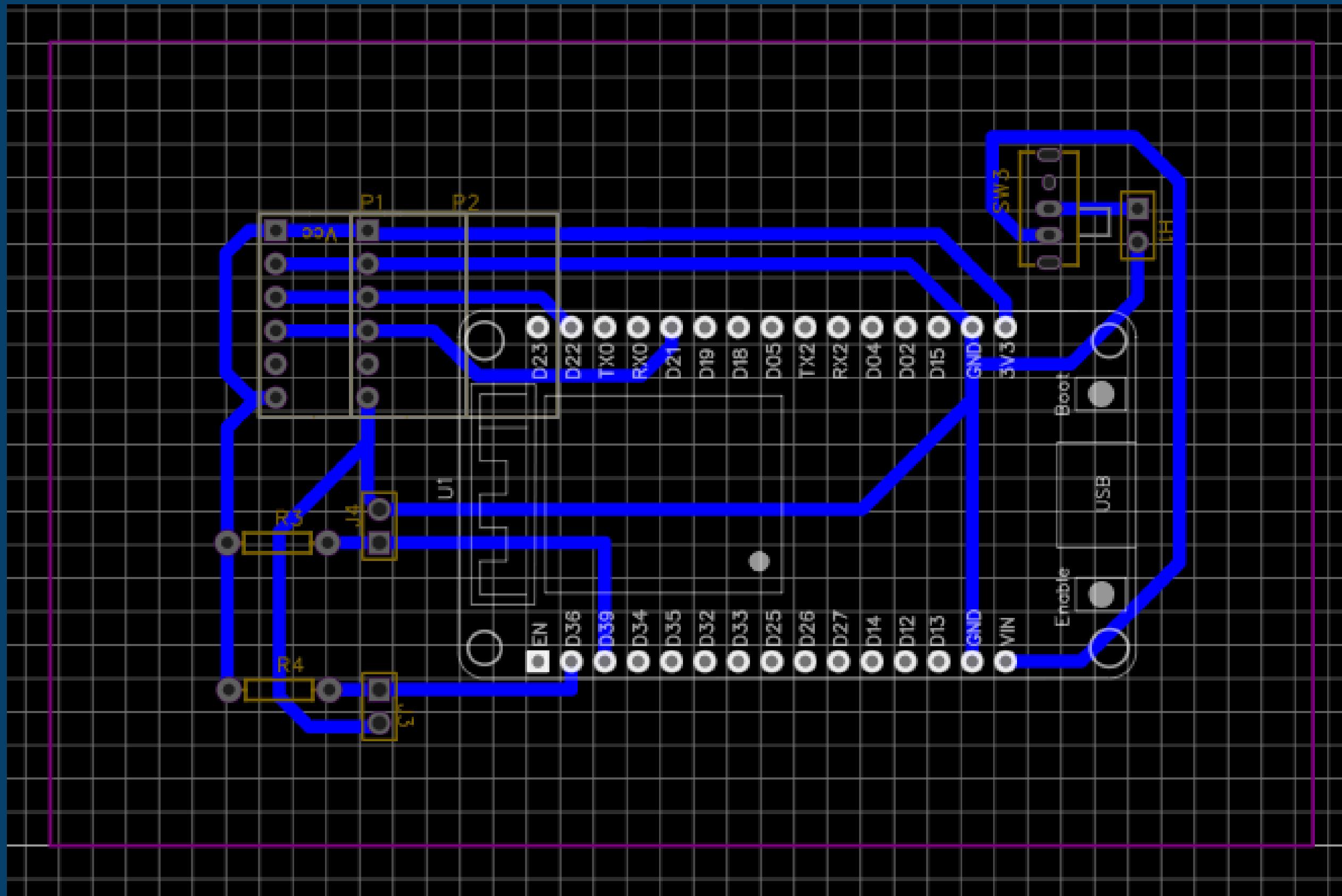
Solo se puede usar  
cuando no se  
inicializa el wifi

# Facundo Maletta

- Colaboración Esquemático
- Diseño de PCB
- Ensamblado de PCB
- Ensamblado de conectores
- Problemas



# PCB



# Sistema



# Agustín Guerra

- Elección de una identidad para el proyecto
- Diseño UX/UI de la aplicación

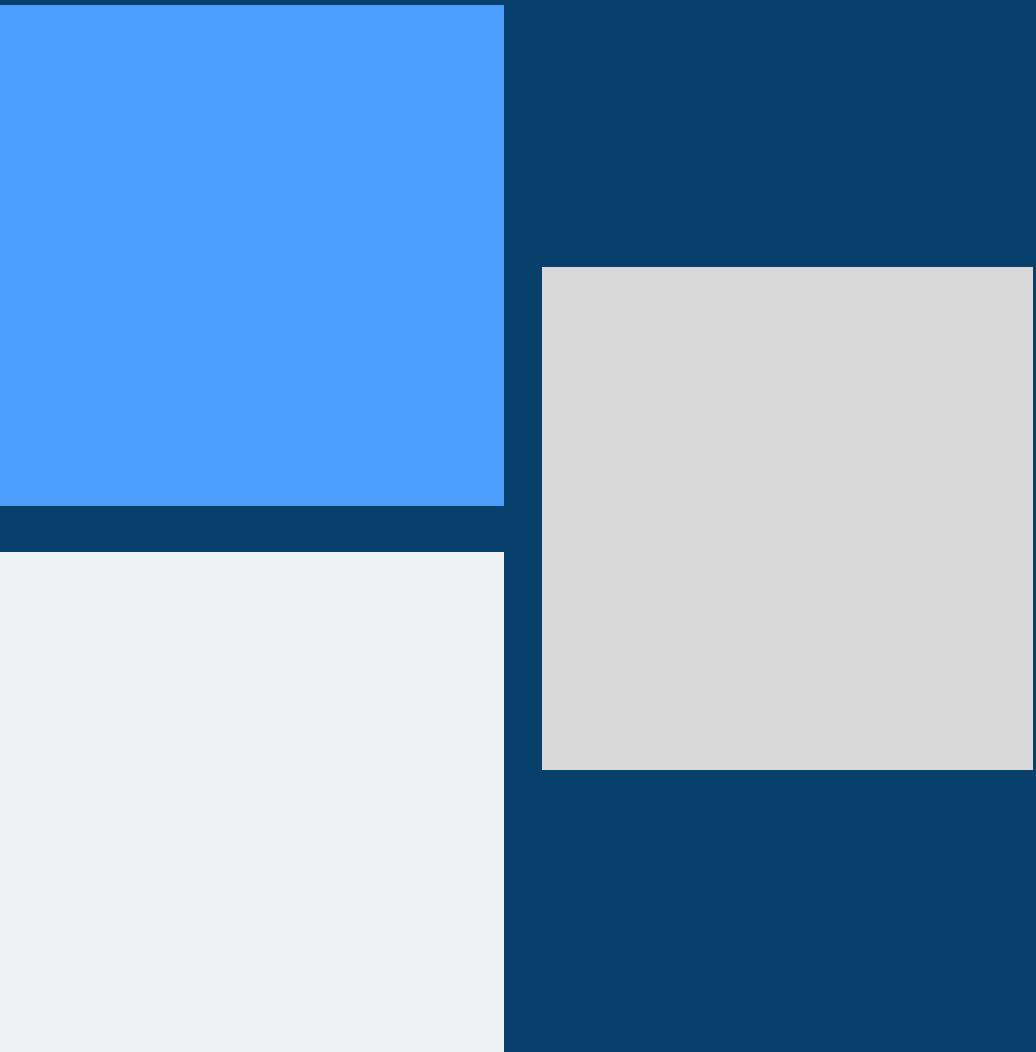


# Identidad de Back On Track

- Paleta de colores
- Creación de un logotipo
- Diseño caracterisco

# Paleta de colores

- Celeste, blanco y gris claro
- Representan y transmiten tranquilidad, paz y confianza



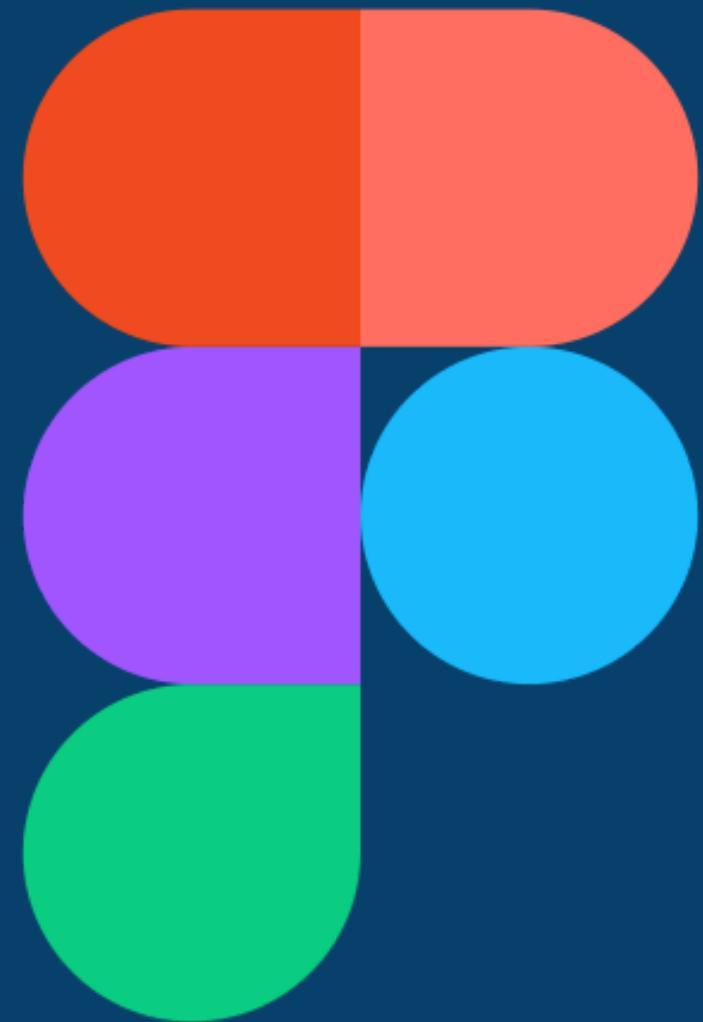
# Logotipo

- Respeta la paleta de colores
- Se identifica con el tema del proyecto



# Diseño UX/UI

- Figma
- Crear pantallas siguiendo la identidad del proyecto
- Experiencia intuitiva y atractiva



# Primer boceto

PRIMER BOCETO

LOGO

DIGITAL  
SELLING

¿QUIENES SOMOS?

FOTO 1

NOVEDADES

FOTO 2

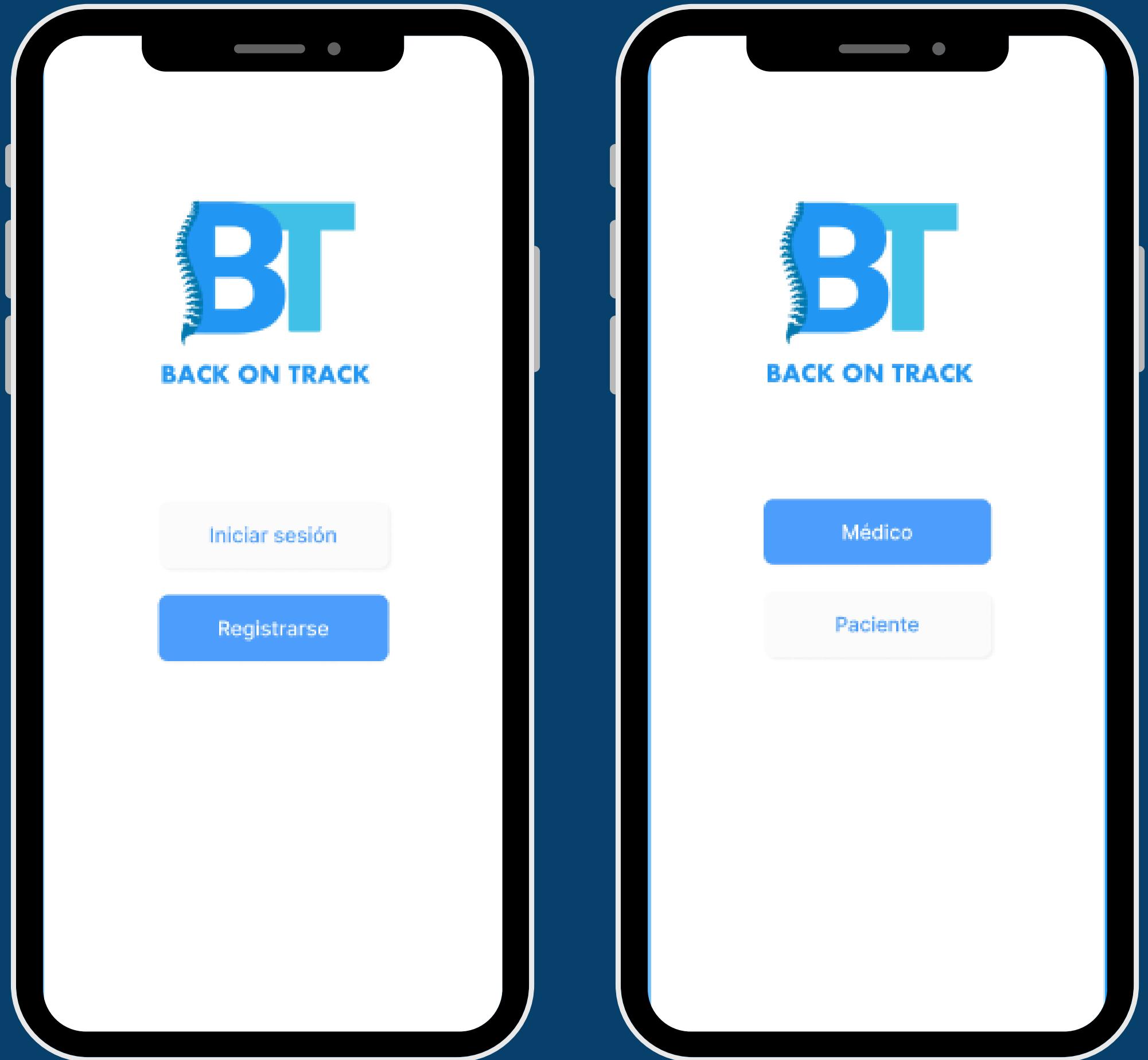
PROCESO

FOTO 3

CONTACTO

FOTO 4

# Inicio de sesión y registro





## Información personal

Nombre/s

Apellido/s

Fecha de nacimiento

xx/xx/yyyy

Número de teléfono

Siguiente



## Datos de la cuenta

Nombre de usuario

Correo electrónico

Contraseña

Confirmar contraseña

Siguiente



## Datos médicos

Altura

Peso

Antecedentes

Siguiente

# Sign Up Paciente



## Información personal

Nombre/s

Apellido/s

Fecha de nacimiento

xx/xx/yyyy

Número de teléfono

Siguiente



## Datos de la cuenta

Nombre de usuario

Correo electrónico

Contraseña

Confirmar contraseña

Siguiente

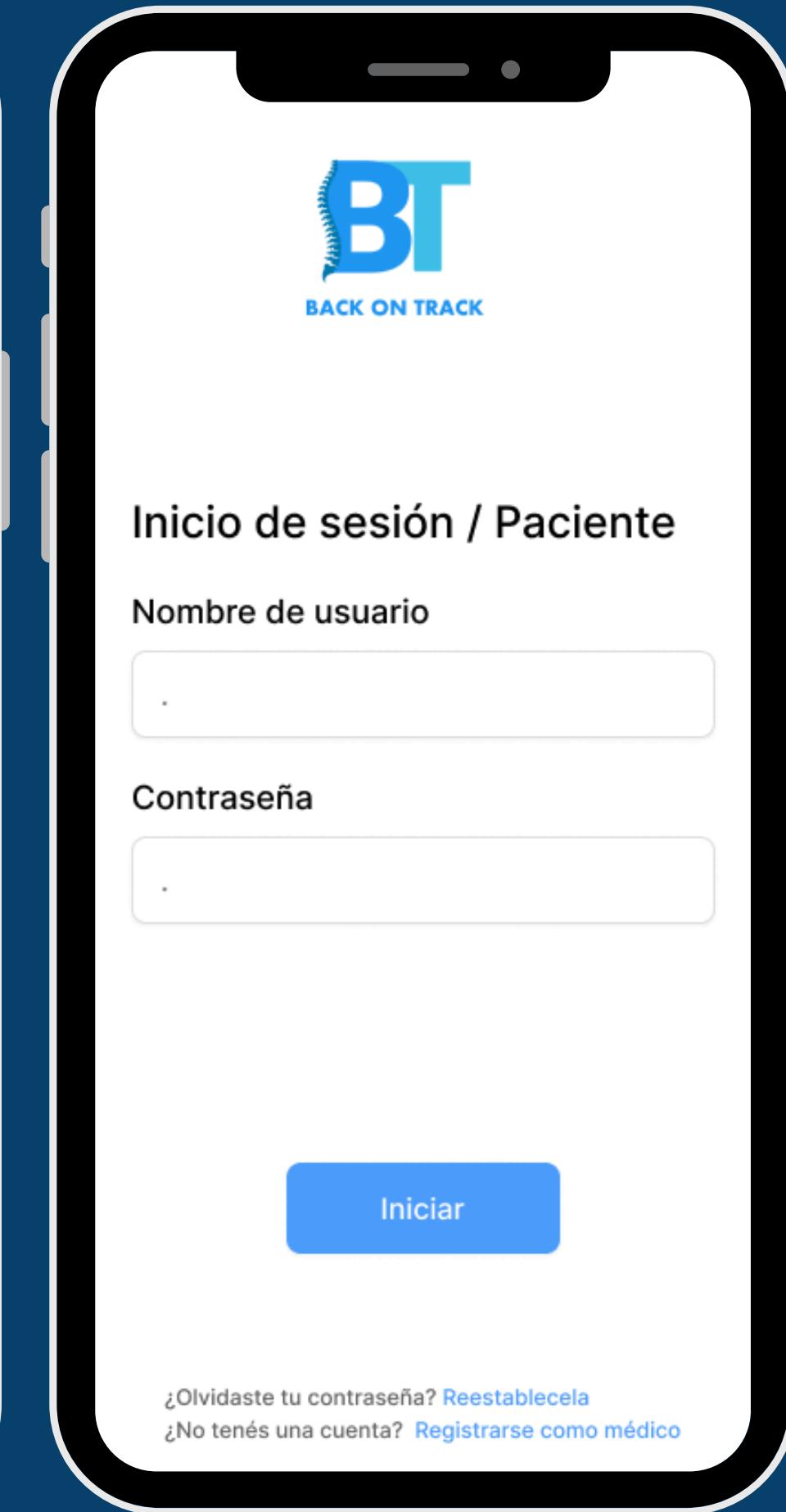
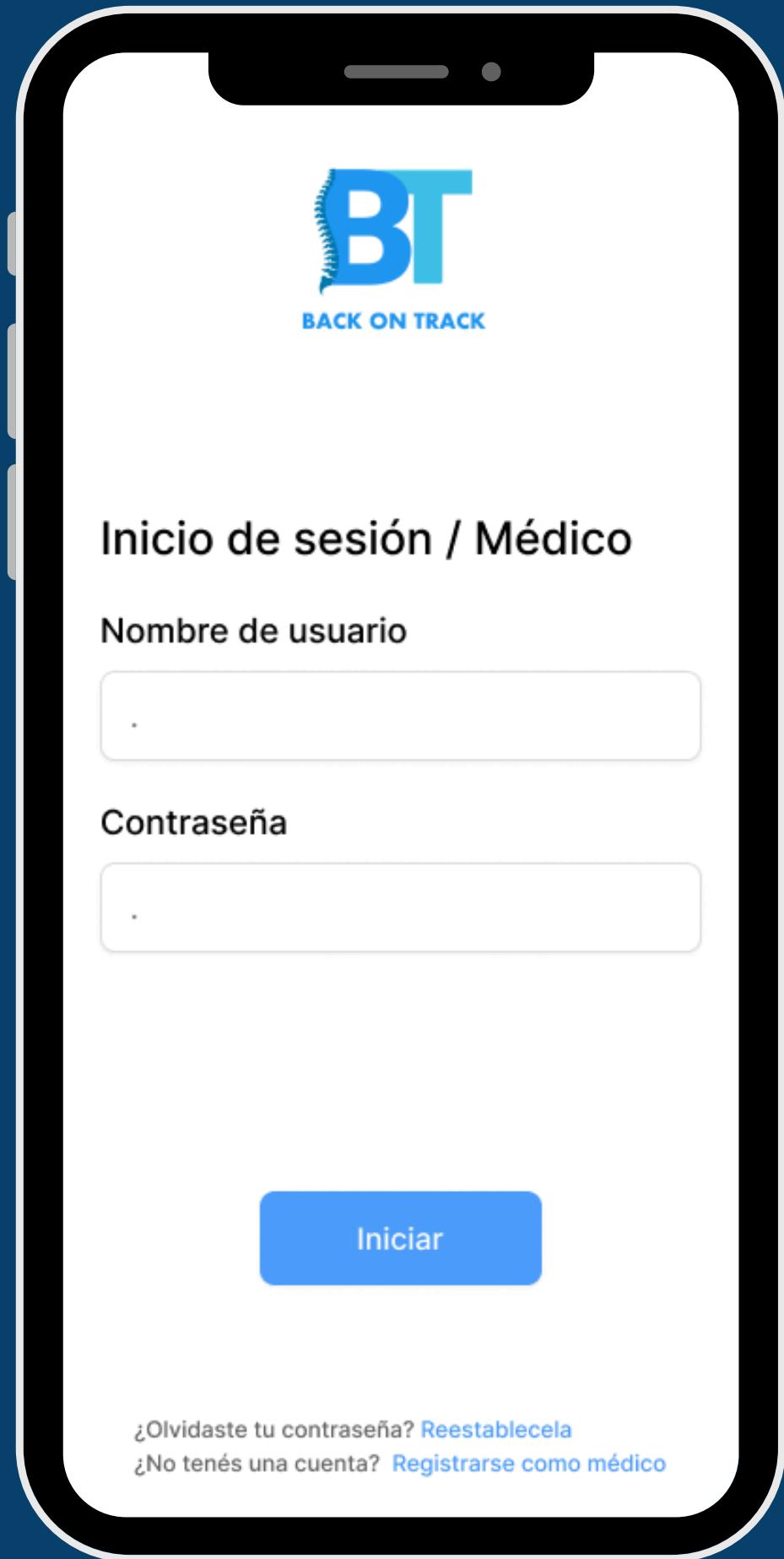


BACK ON TRACK

## Organización

¿De qué organización formás parte?

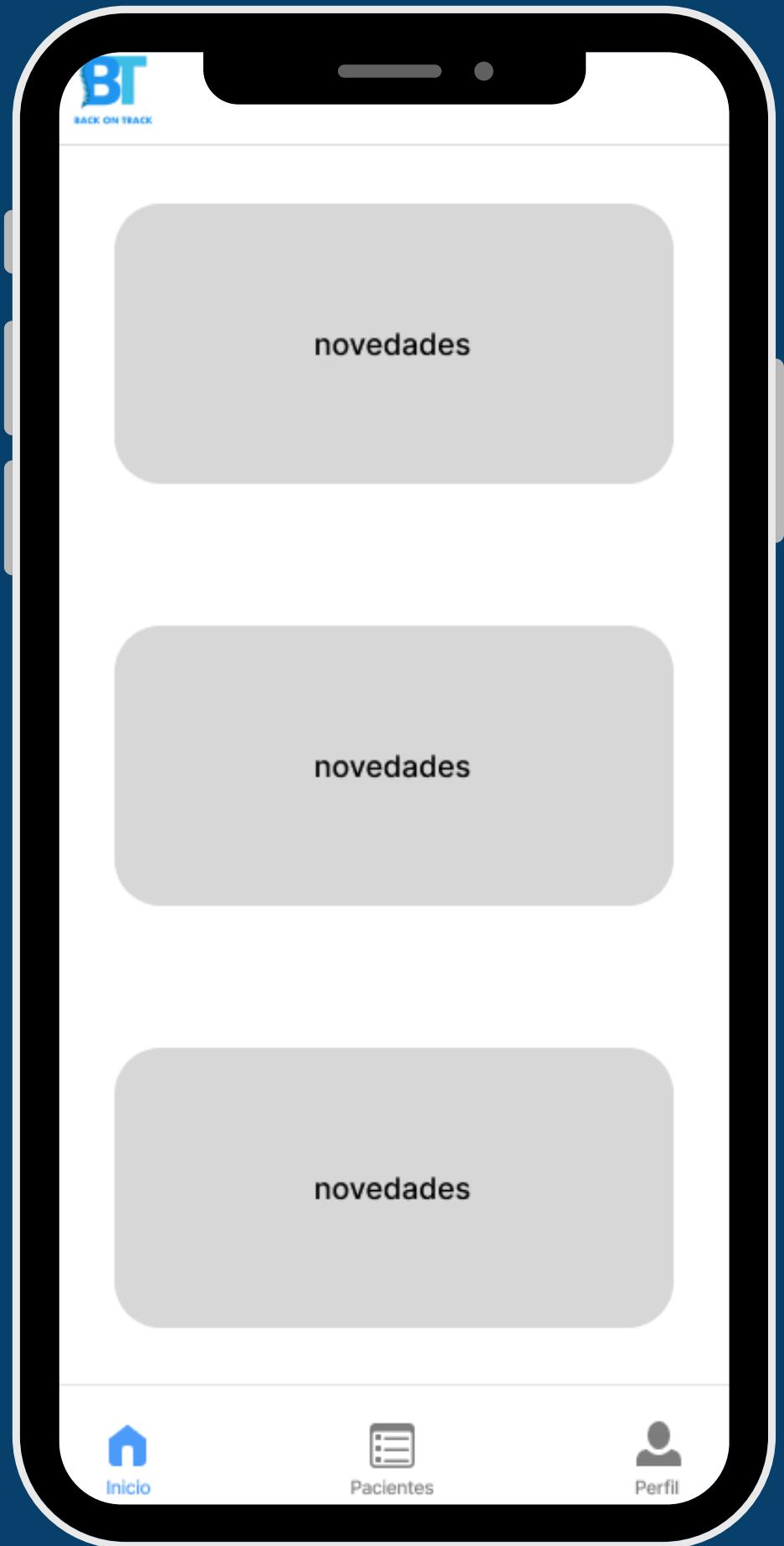
Sign Up  
Medico



# Inicios de sesión

¿Olvidaste tu contraseña? [Reestablecela](#)  
¿No tenés una cuenta? [Registrarse como médico](#)

¿Olvidaste tu contraseña? [Reestablecela](#)  
¿No tenés una cuenta? [Registrarse como médico](#)



# Home (para ambos usuarios)

# Punto de vista del médico

# Lista de pacientes y información

The image displays three sequential screenshots of a mobile application interface, likely for a healthcare provider, showing a list of patients and a detailed patient profile.

**Screenshot 1: Lista de pacientes**  
The screen shows a list of patients with their names and small profile pictures. The names listed are Sandra Casandra, Julian Vinocur, Facundo Maletta, Mia Payse, and Feliciano Miguez. The interface includes a search bar at the top right and a navigation bar at the bottom with icons for Inicio, Pacientes, and Perfil.

**Screenshot 2: Search results for Sandra Casandra**  
The screen shows a search results list for "Sandra Casandra". It lists multiple entries for Sandra Casandra, each with a profile picture and the name "Sandra Casandra". A search bar at the top and a navigation bar at the bottom are visible.

**Screenshot 3: Detalle del paciente Sandra Casandra**  
The screen shows a detailed profile for Sandra Casandra. It includes her profile picture, name, age (32 años), and a note about her medical history: "Antecedentes: ACV". Below this, there are three tabs: "Última semana" (highlighted in blue), "Último mes", and "Otro". A large "X" is overlaid on the bottom half of the screen, indicating that the information or list is currently unavailable.

# Perfil del médico



# Edición de datos del médico

The image displays three sequential screenshots from a mobile application interface, illustrating the process of editing a medical professional's profile. The app has a dark blue header with the logo "BT BACK ON TRACK".

**Screenshot 1: EDITAR PERFIL**

- Profile picture of a man with a beard.
- Name: Andres Nefario
- Age: 32 años
- Organization: CEMIC

**Screenshot 2: Datos personales**

- Nombre/s: Andres
- Apellido/s: Nefario
- Fecha de nacimiento: xx/xx/yyyy
- Organización: CEMIC

**Screenshot 3: Mi información**

- Número de teléfono: +54 11 xxxx xxxx
- Correo electrónico: nefario@gmail.com
- Contraseña: ..... (represented by dots)

Each screen includes a back arrow at the top left and a navigation bar at the bottom with icons for Inicio, Pacientes, and Perfil (the last one is highlighted in blue).

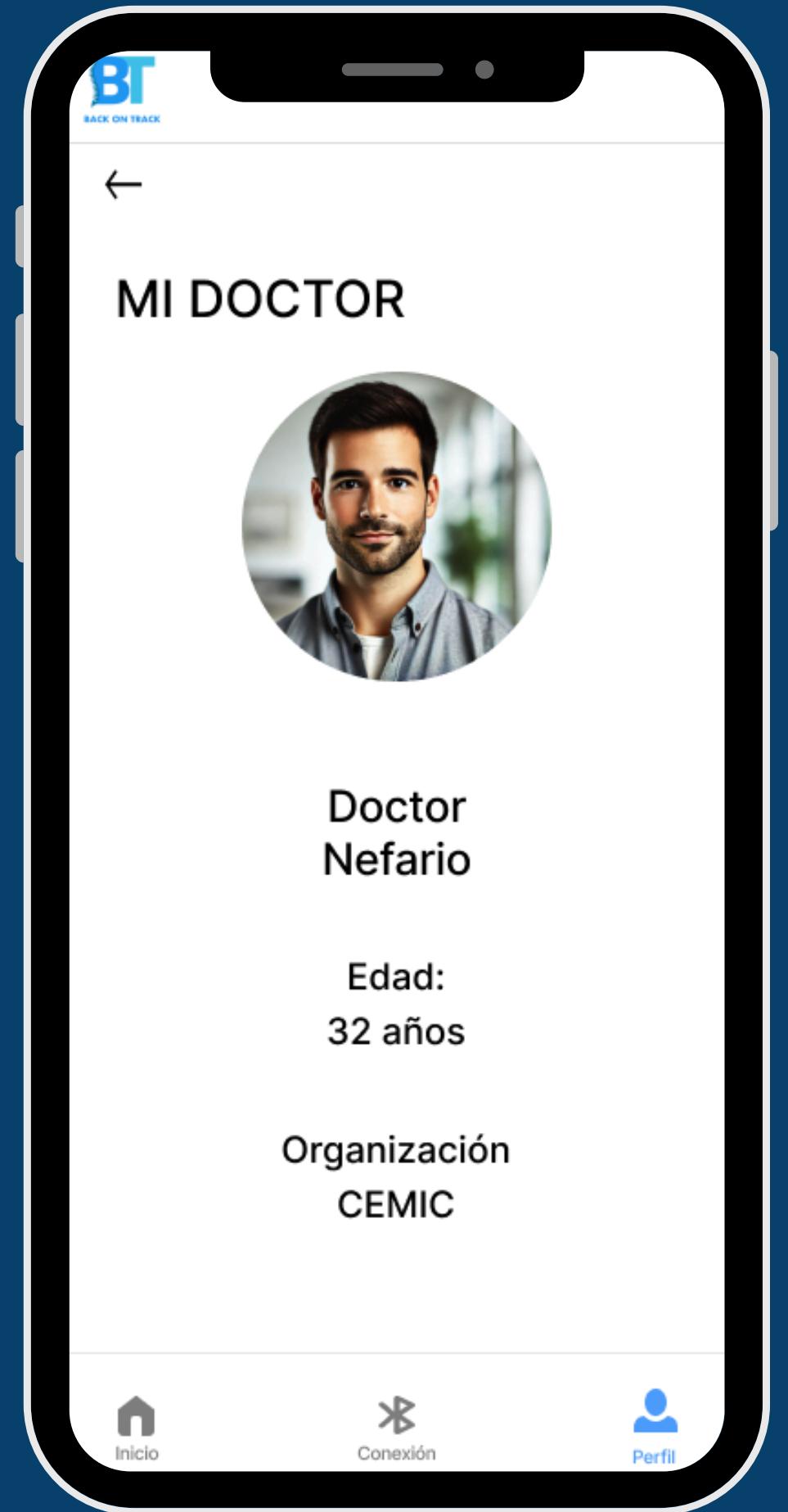
# Punto de vista del paciente



# Estado de conexión del dispositivo

# Perfil del paciente





# Mi doctor

The image displays three sequential screenshots of a mobile application interface for editing a patient's profile. The background is a dark blue gradient.

**Screenshot 1: EDITAR PERFIL**

- Top left: BT BACK ON TRACK logo.
- Top right: Back arrow.
- Section title: EDITAR PERFIL.
- User photo: Circular profile picture of a woman with long dark hair.
- Name: Sandra Casandra (with edit pencil icon).
- Age: Edad: 32 años (with edit pencil icon).
- Past Medical History: Antecedentes: ACV (with edit pencil icon).
- Bottom navigation bar: Inicio, Conexión, Perfil (the Perfil icon is highlighted in blue).

**Screenshot 2: Datos personales**

- Section title: Datos personales.
- Fields:
  - Nombre/s: Sandra (edit pencil icon)
  - Apellido/s: Casandra (edit pencil icon)
  - Fecha de nacimiento: xx/xx/xxxx (edit pencil icon)
  - Altura: 1,70 metros (edit pencil icon)
  - Peso: 55kg (edit pencil icon)

**Screenshot 3: Mi información**

- Section title: Mi información.
- Fields:
  - Número de teléfono: +54 11 xxxx xxxx (edit pencil icon)
  - Correo electrónico: sandracas@gmail.com (edit pencil icon)
  - Contraseña: ..... (edit pencil icon)

**Text on the right side:**

# Edición de datos del paciente

# Julián Vinocur

- Programación de la aplicación
- Construcción de una base de datos con todos los usuarios y su información.

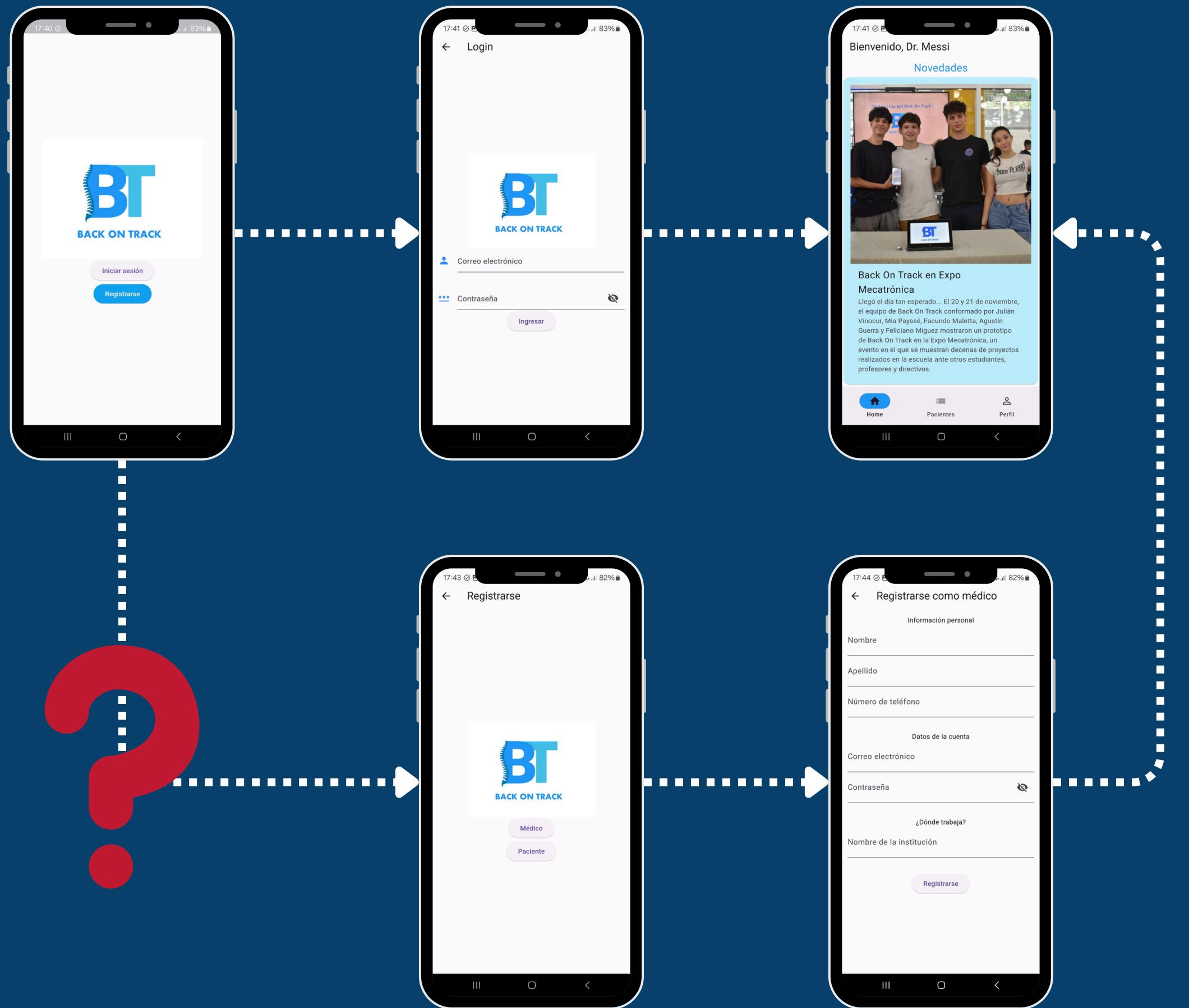


# MVP

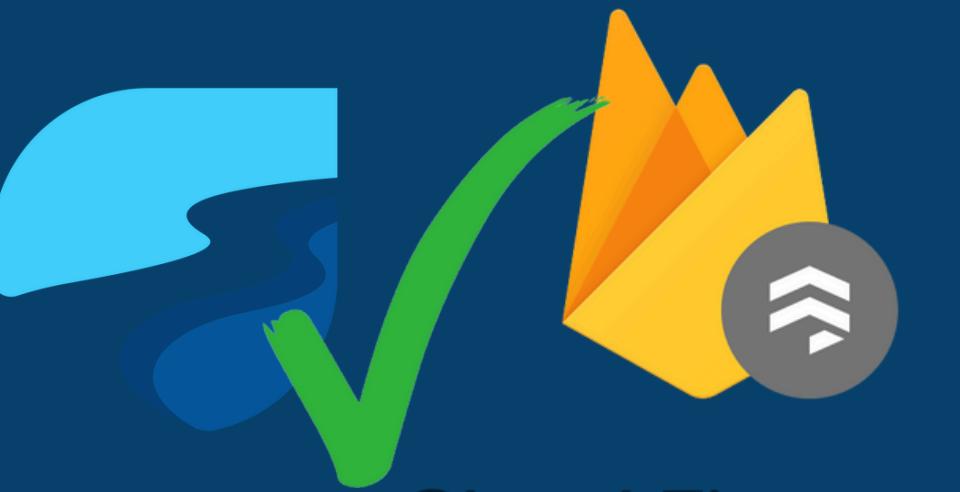
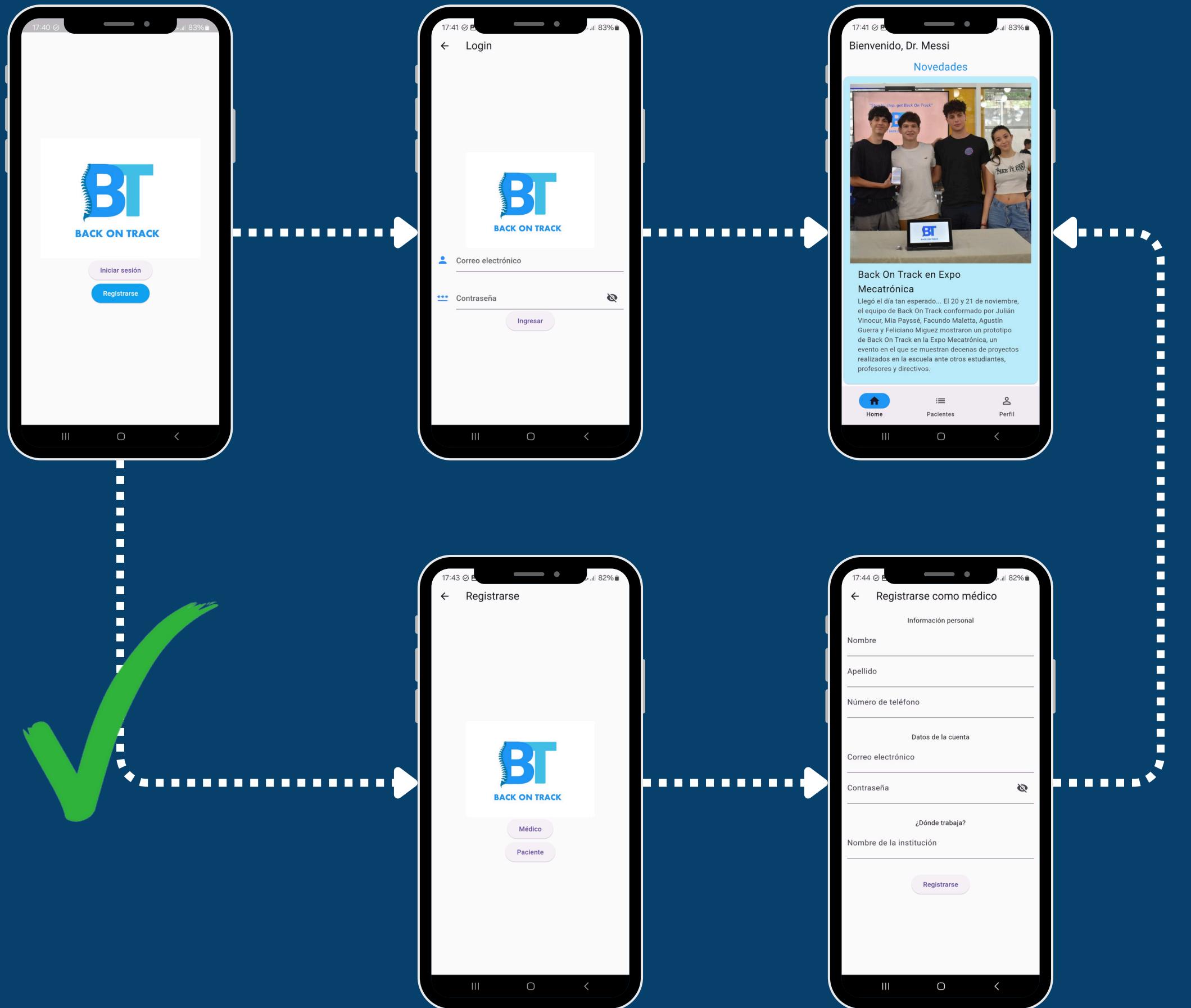
- “En una aplicación o página web, el médico podrá registrarse e iniciar sesión con su cuenta”.
- “Al ingresar, tendrá acceso a todos sus pacientes y podrá ver los datos de cada uno”.



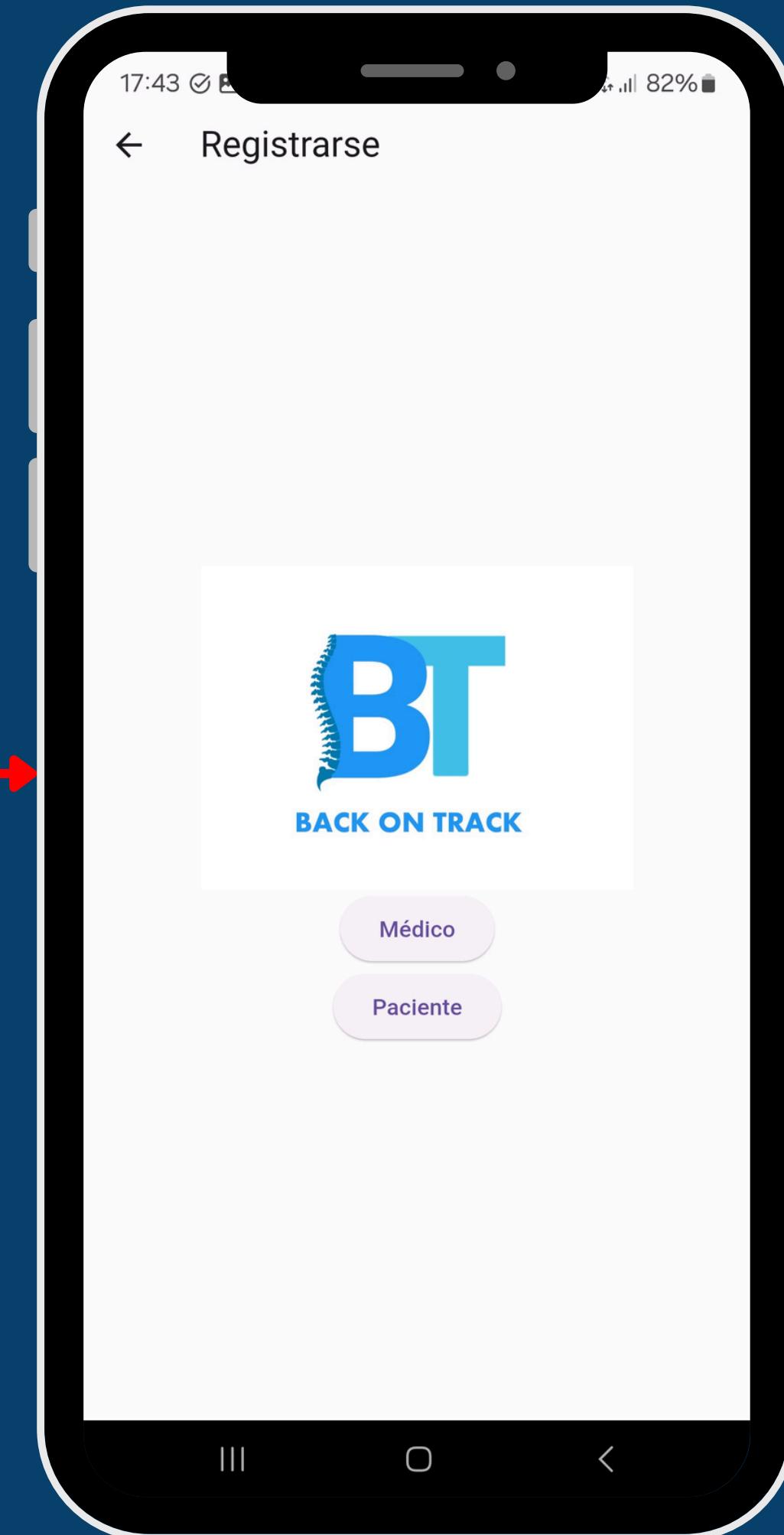
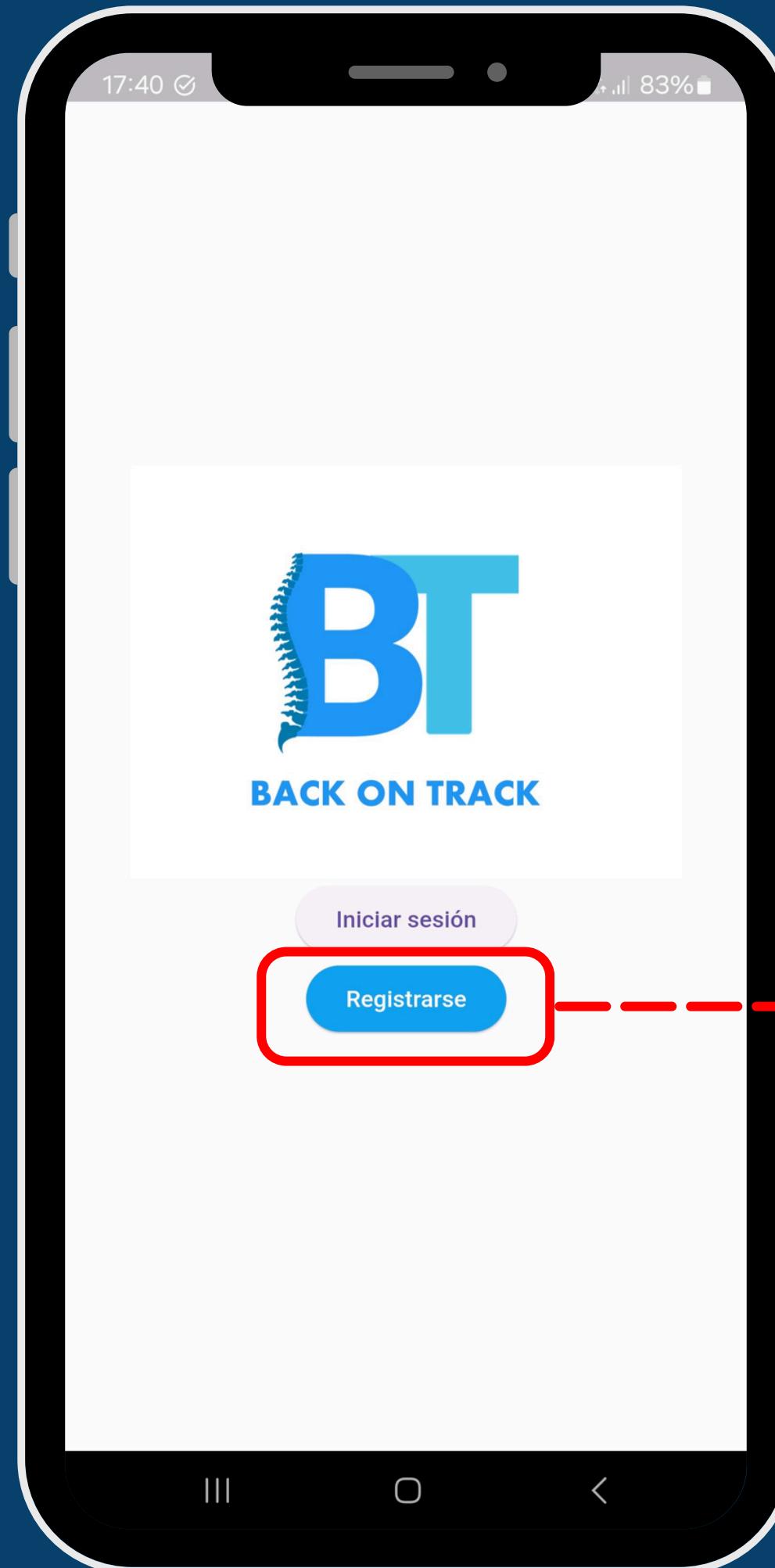
Cloud Firestore

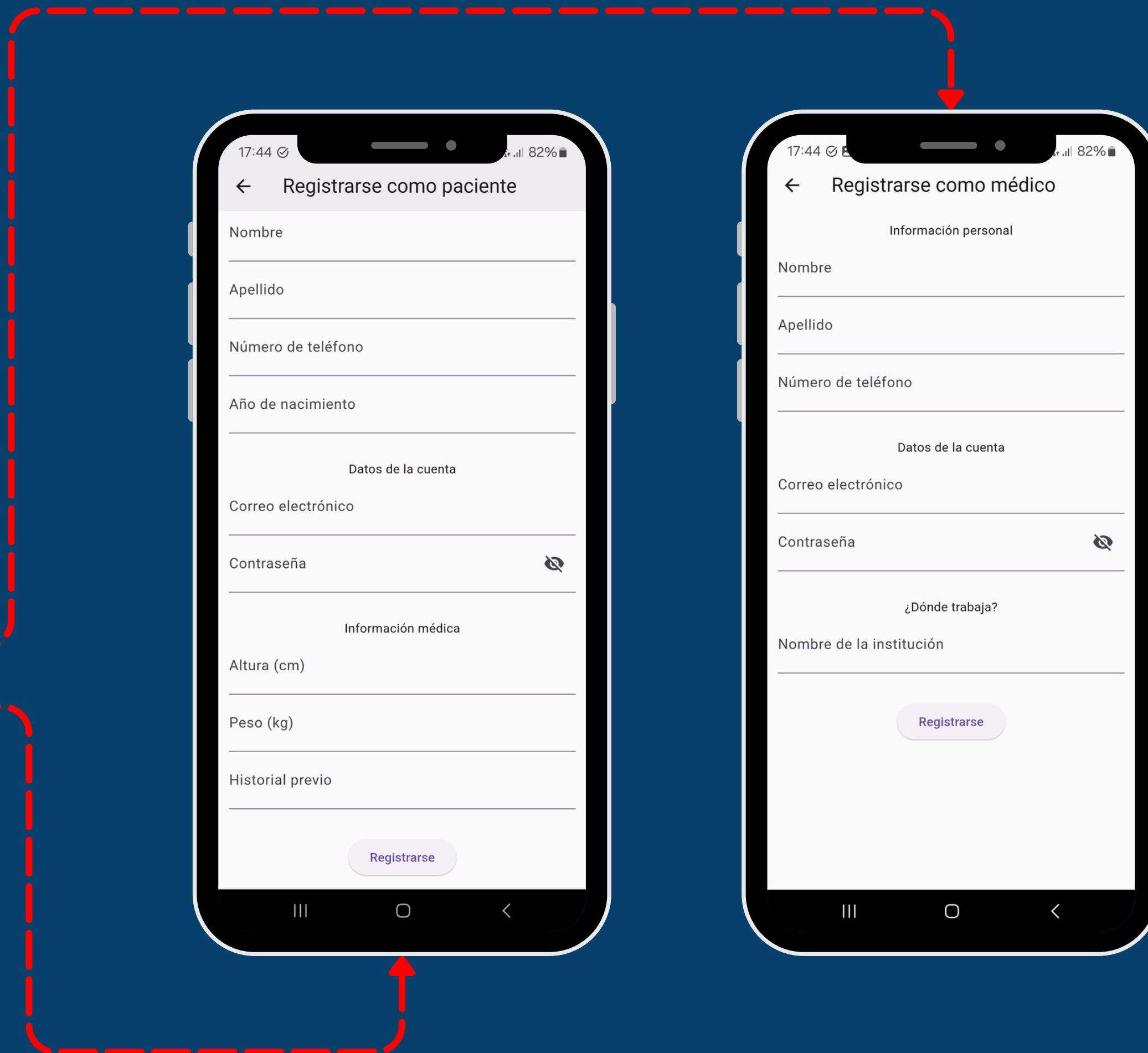
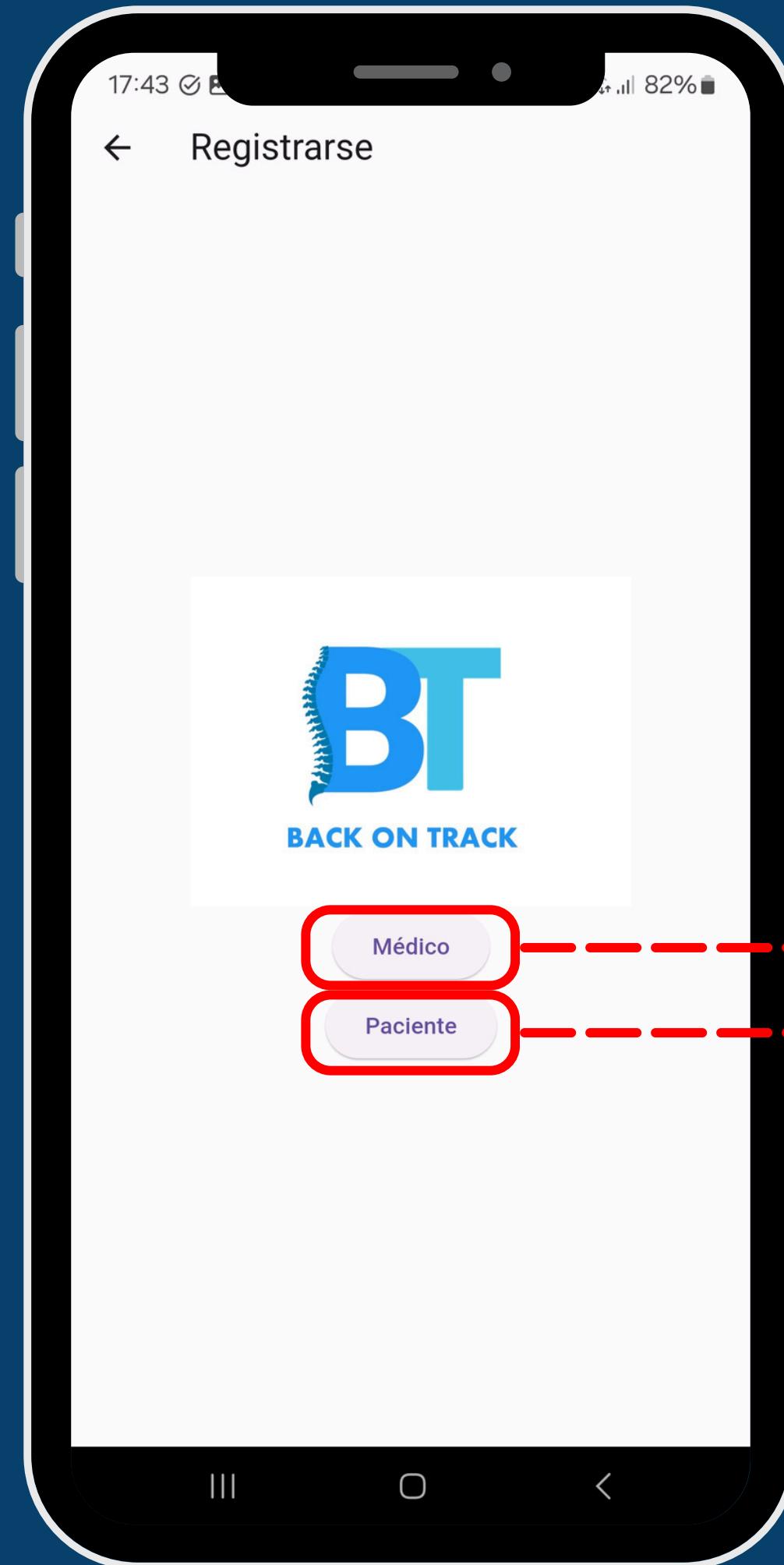


Cloud Firestore



Cloud Firestore





18:29 81%

← Registrarse como médico

Información personal

**Nombre**

Este campo no puede estar vacío

**Apellido**

Este campo no puede estar vacío

**Número de teléfono**

---

El número de teléfono no puede contener símbolos o letras

Datos de la cuenta

**Correo electrónico**

hskdksd

Ingrese un correo electrónico válido

**Contraseña**

.....

La contraseña debe contener más de 8 caracteres

¿Dónde trabaja?

**Nombre de la institución**

Este campo no puede estar vacío

**Registrarse**

17:44 82% 82%

← Registrarse como médico

Información personal

Nombre

---

Apellido

---

Número de teléfono

---

Datos de la cuenta

Correo electrónico

---

Contraseña

---

¿Dónde trabaja?

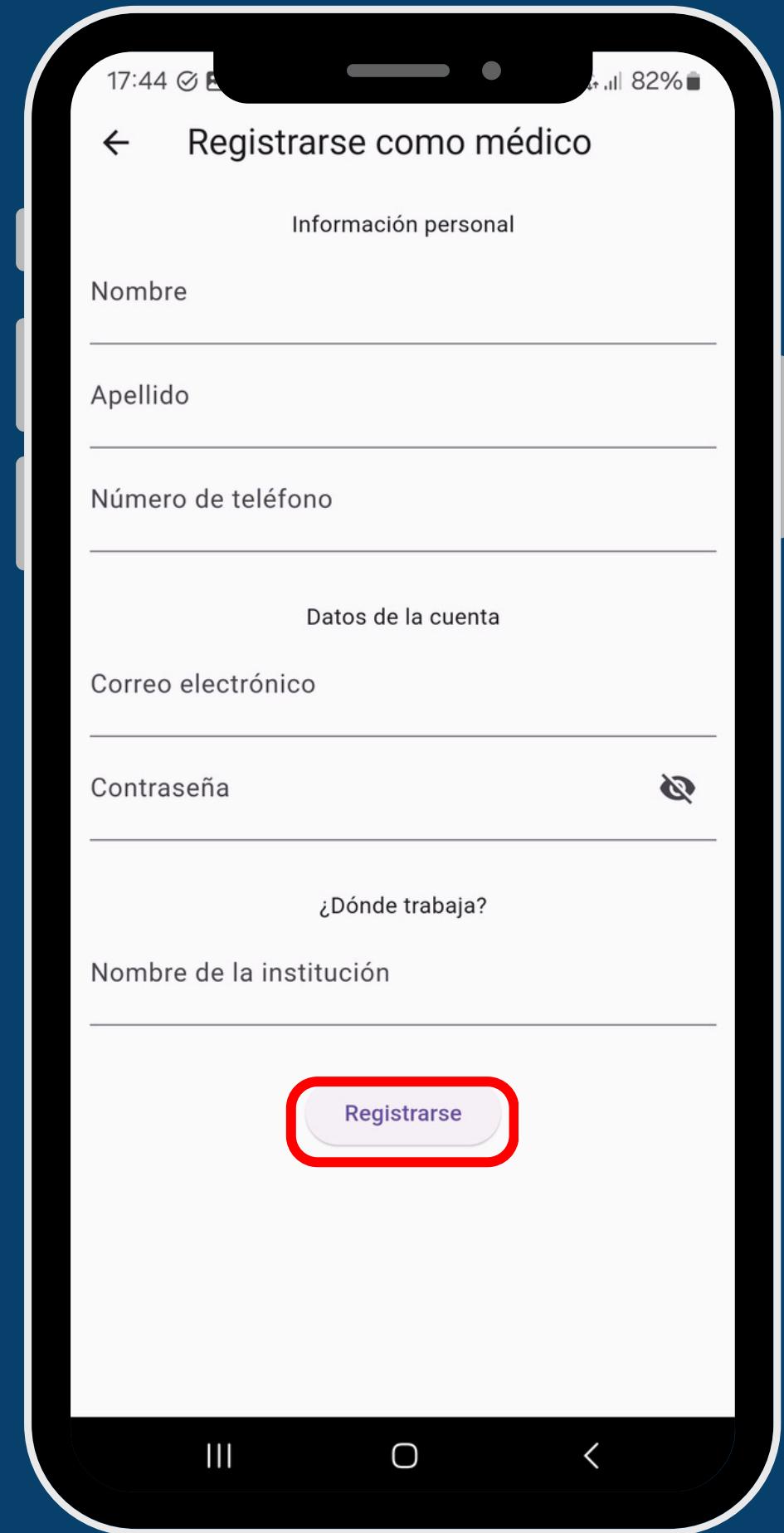
Nombre de la institución

---

Registrarse



# Cloud Firestore



```
class User {  
    String userId;  
    String name;  
    String surname;  
    String email;  
    String phone;  
    String password;  
  
    bool isDoctor;  
  
    //Patients:  
    String? assignedDoctor;  
    int? birthYear;  
    double? height;  
    double? weight;  
    String? previousInfo;  
  
    //Doctors:  
    List<String>? assignedPatients;  
    String? healthCenter;
```

17:44 82%

← Registrarse como médico

Información personal

Nombre

Apellido

Número de teléfono

Datos de la cuenta

Correo electrónico

Contraseña

¿Dónde trabaja?

Nombre de la institución

Registrarse

Más funciones en Google Cloud

(default)	users	QKP2KIFGLktyZB4QZiCS
+ Iniciar colección	+ Agregar documento	+ Iniciar colección
tracking	HSFW00LQHkrEzTBKZQl2	+ Agregar campo
ukmlaNEJSykBmvygrkKm	KHeTVh78SKnwEDCKfMlI	assignedPatients
users >	KVjAClMOjYapzqXWwWBN	0 "ukmlaNEJSykBmvygrkKm"
	N3gXrvrSbh5M7qS8bxF8	1 "9FMkZVkJnd0URiZan94r"
	QKP2KIFGLktyZB4QZiCS >	2 "X0QpdAL4BbM6ylegWPxC"
	Uq4JupU0s0EiyUG5qT3q	3 "N3gXrvrSbh5M7qS8bxF8"
	VQK8N5w4S081U6qsS1Ez	4 "KVjAClMOjYapzqXWwWBN"
	WcW3WjCnTOE7h4i3VHvJ	5 "FgmrVi204NulCPFqmrtJ"
	X0QpdAL4BbM6ylegWPxC	6 "VQK8N5w4S081U6qsS1Ez"
	fw50uRcIo2J3t6uj9gtA	7 "HSFW00LQHkrEzTBKZQl2"
	sUhJdcPKiiwWDFbkbP0j	8 "Uq4JupU0s0EiyUG5qT3q"
	ukmlaNEJSykBmvygrkKm	email: "ejemplodoctor@gmail.com"
	vywbSLB7kym8GX0hK8RZ	healthCenter: "Inter Miami"
	zSojnJlt5dZxTrj6MHzs	isDoctor: true

17:44 82% ← Registrarse como médico

Información personal

Nombre

Apellido

Número de teléfono

Datos de la cuenta

Correo electrónico

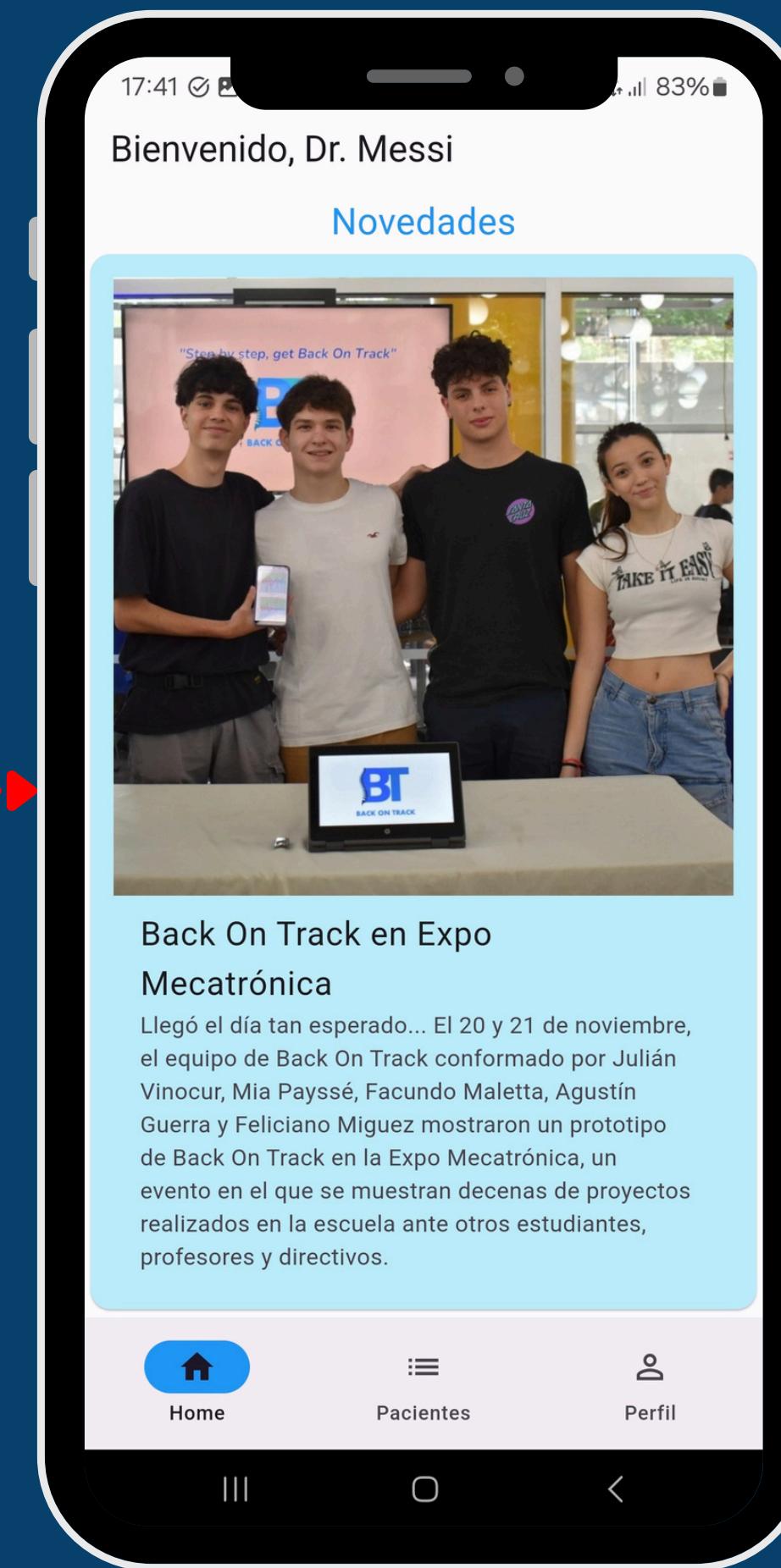
Contraseña  
 

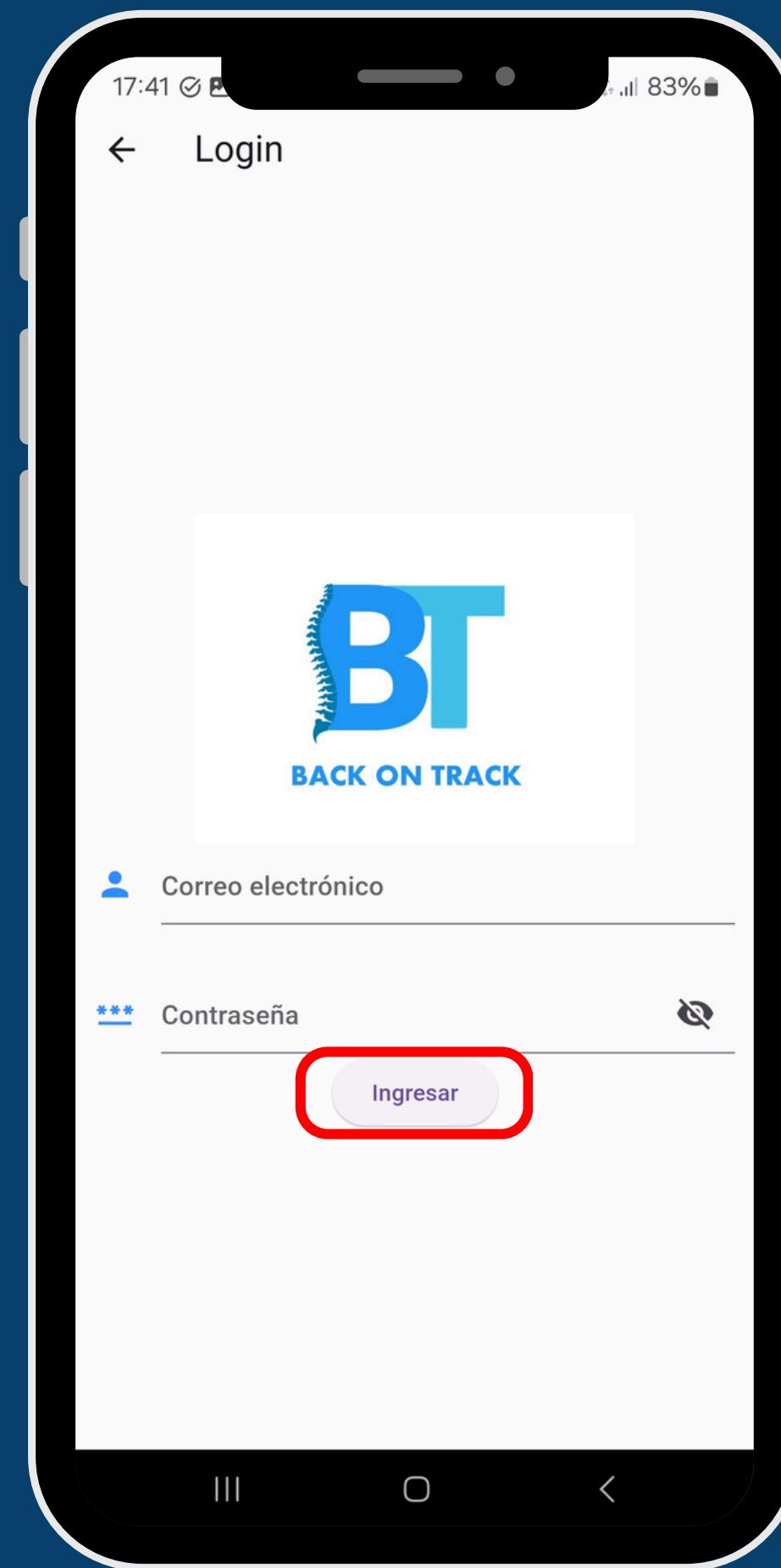
¿Dónde trabaja?

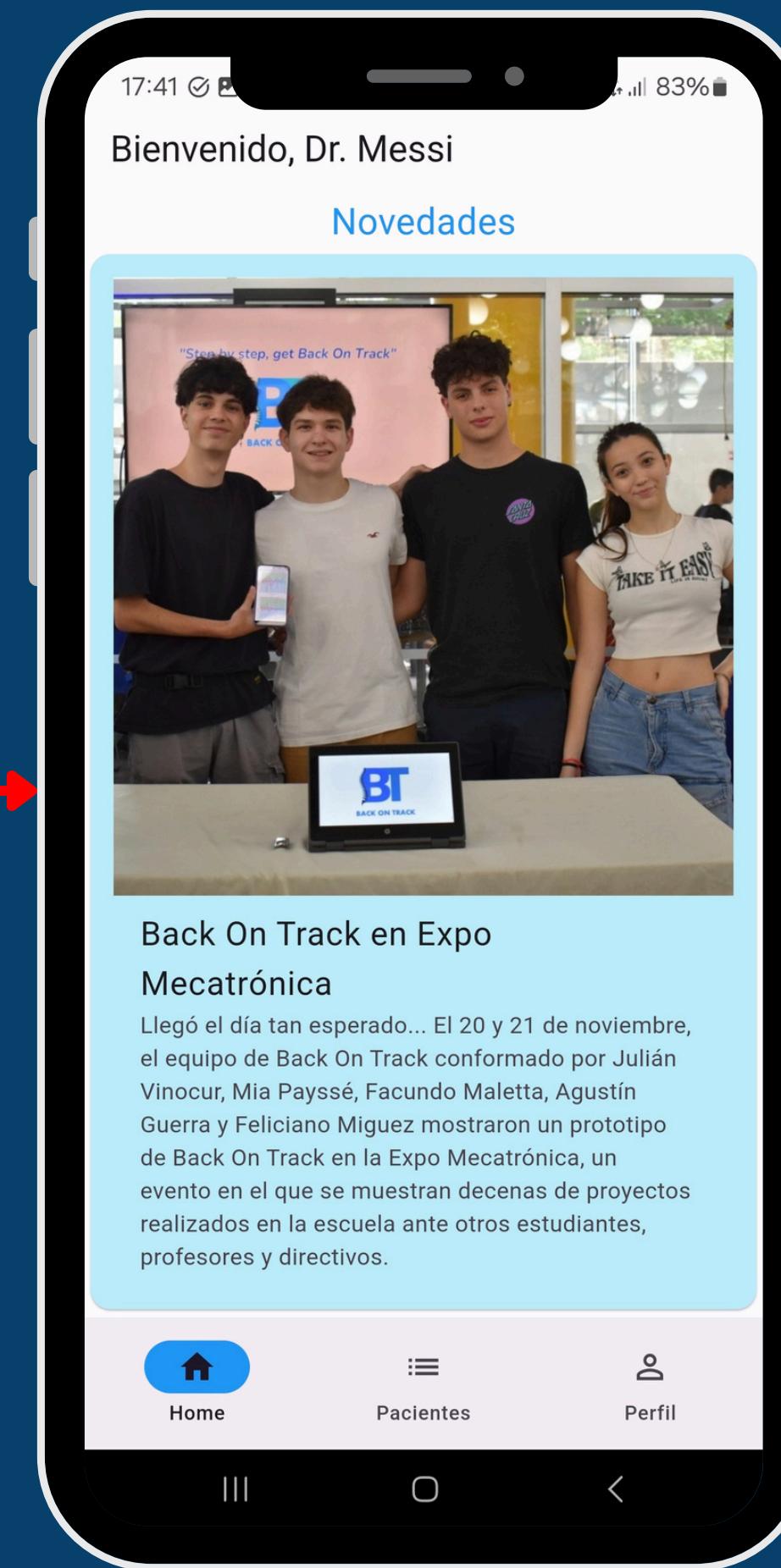
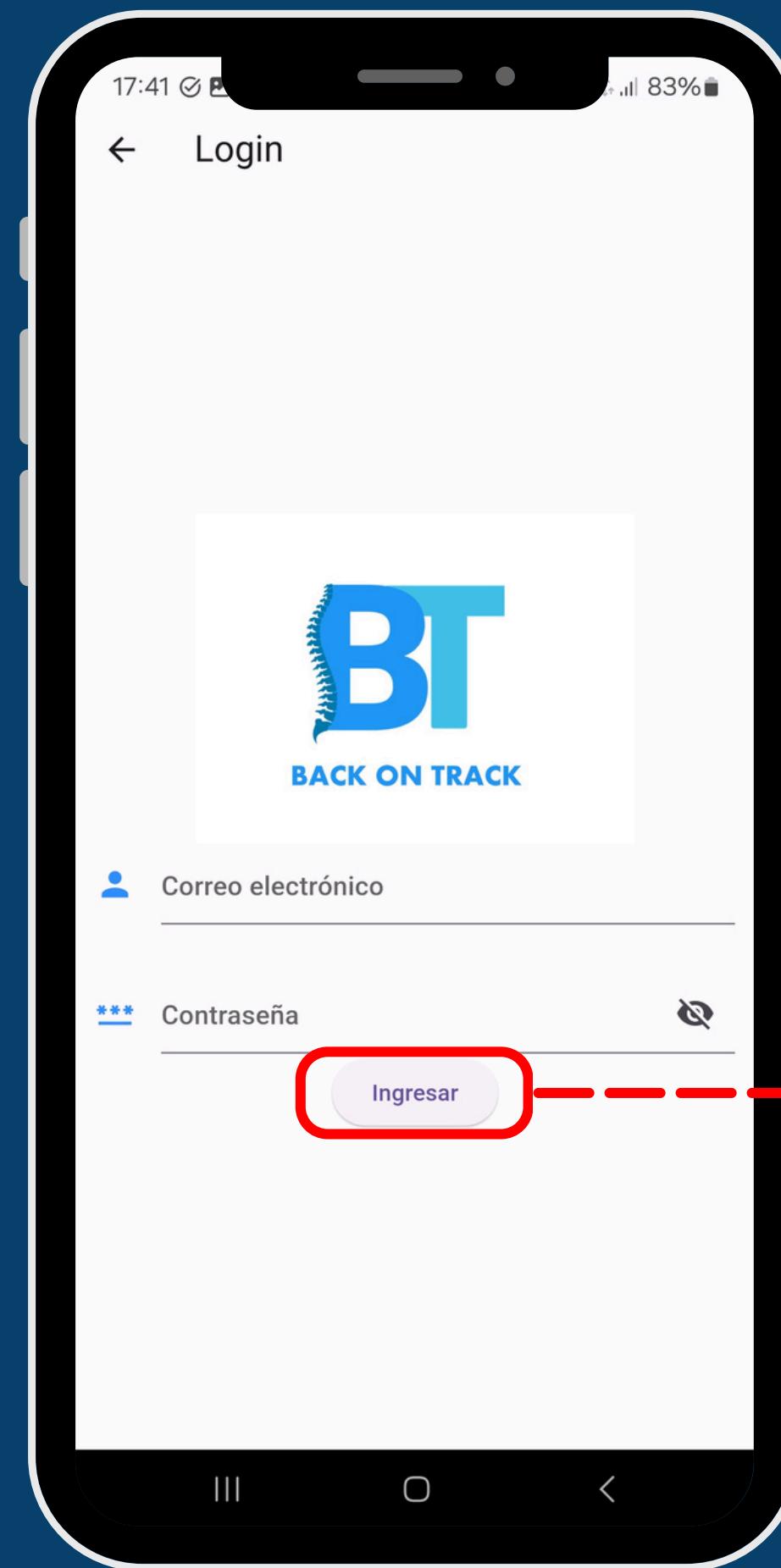
Nombre de la institución

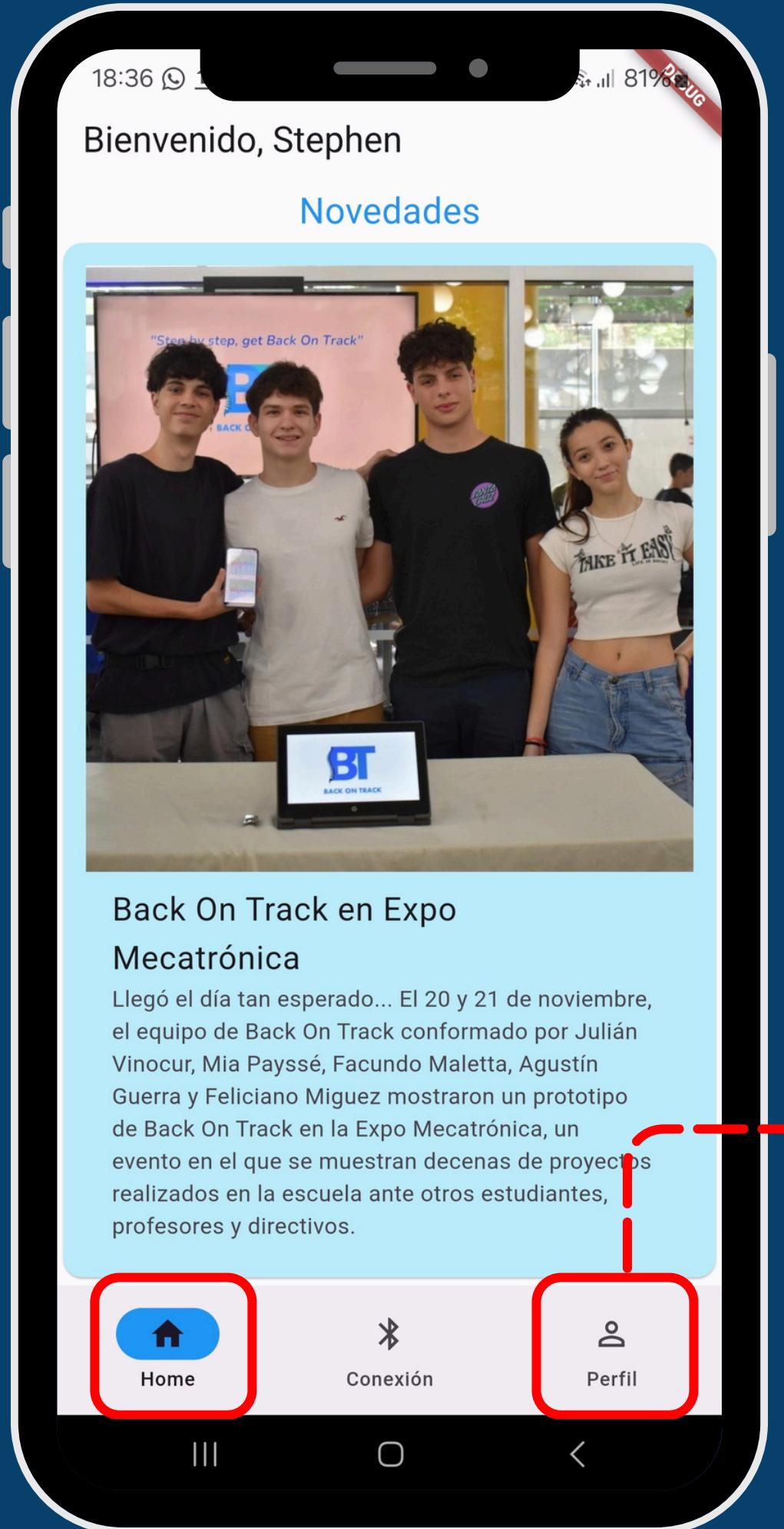
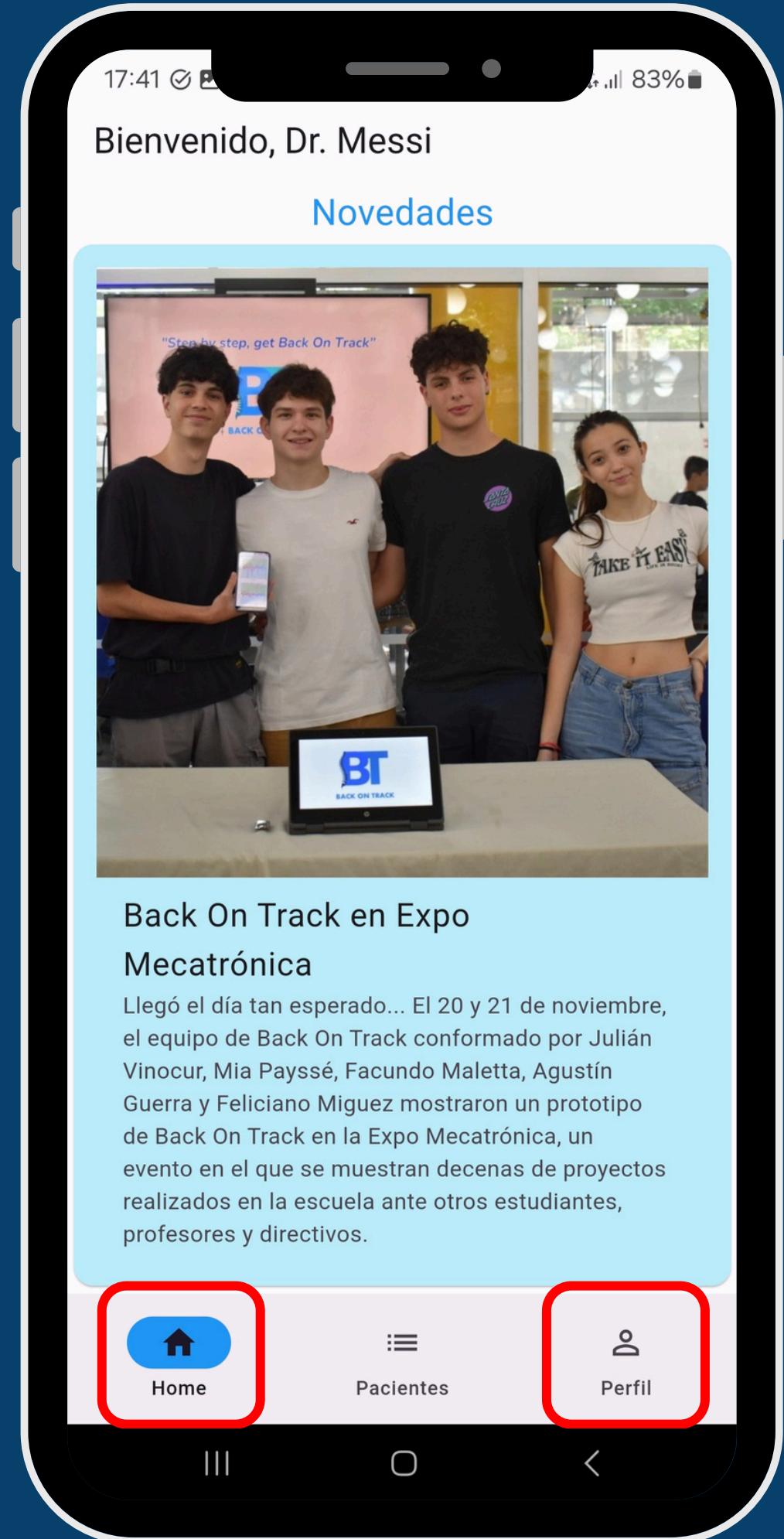
**Registrarse**

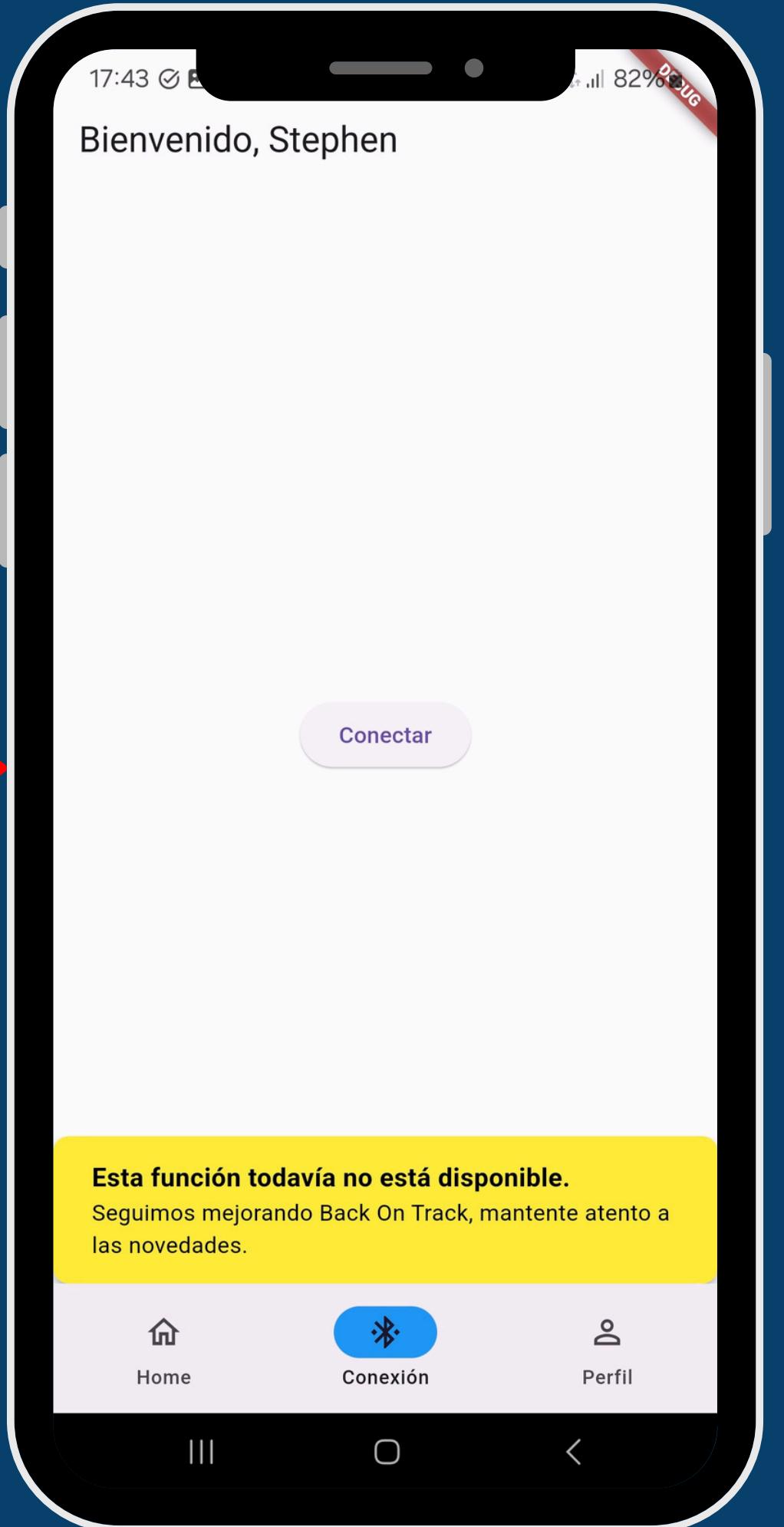
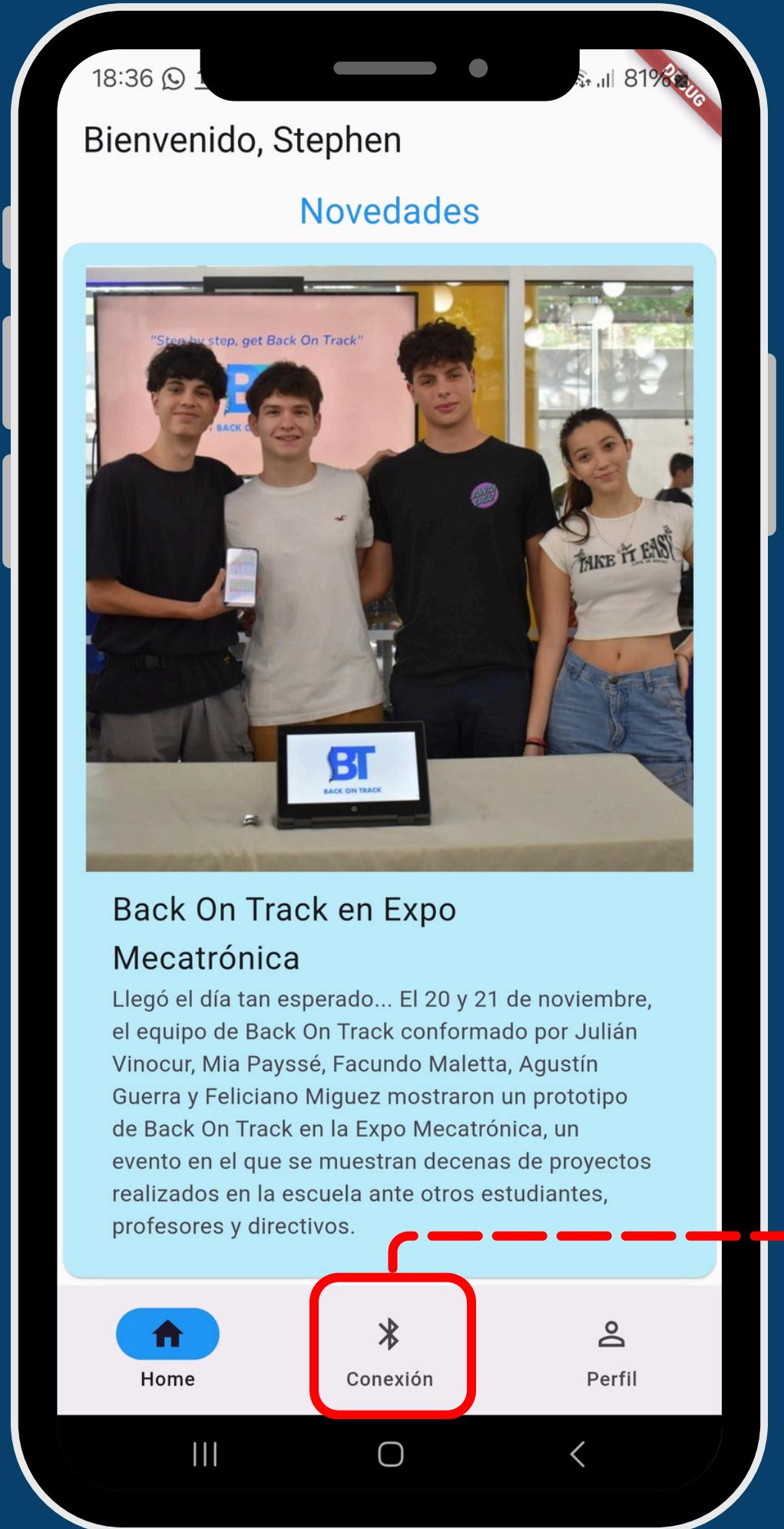
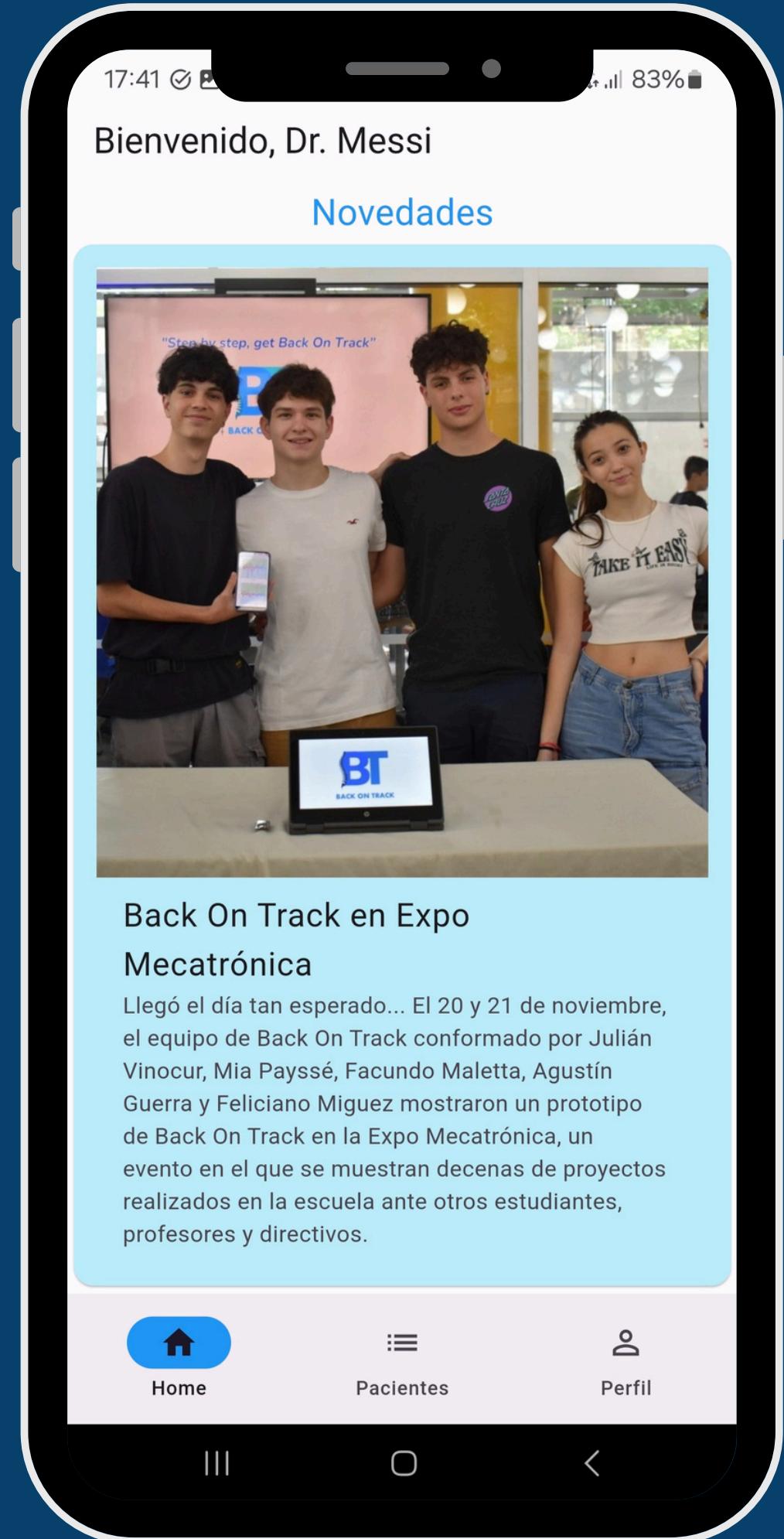
A red dashed arrow points from the "Registrarse" button on the left phone screen to the "Novedades" section on the right phone screen.

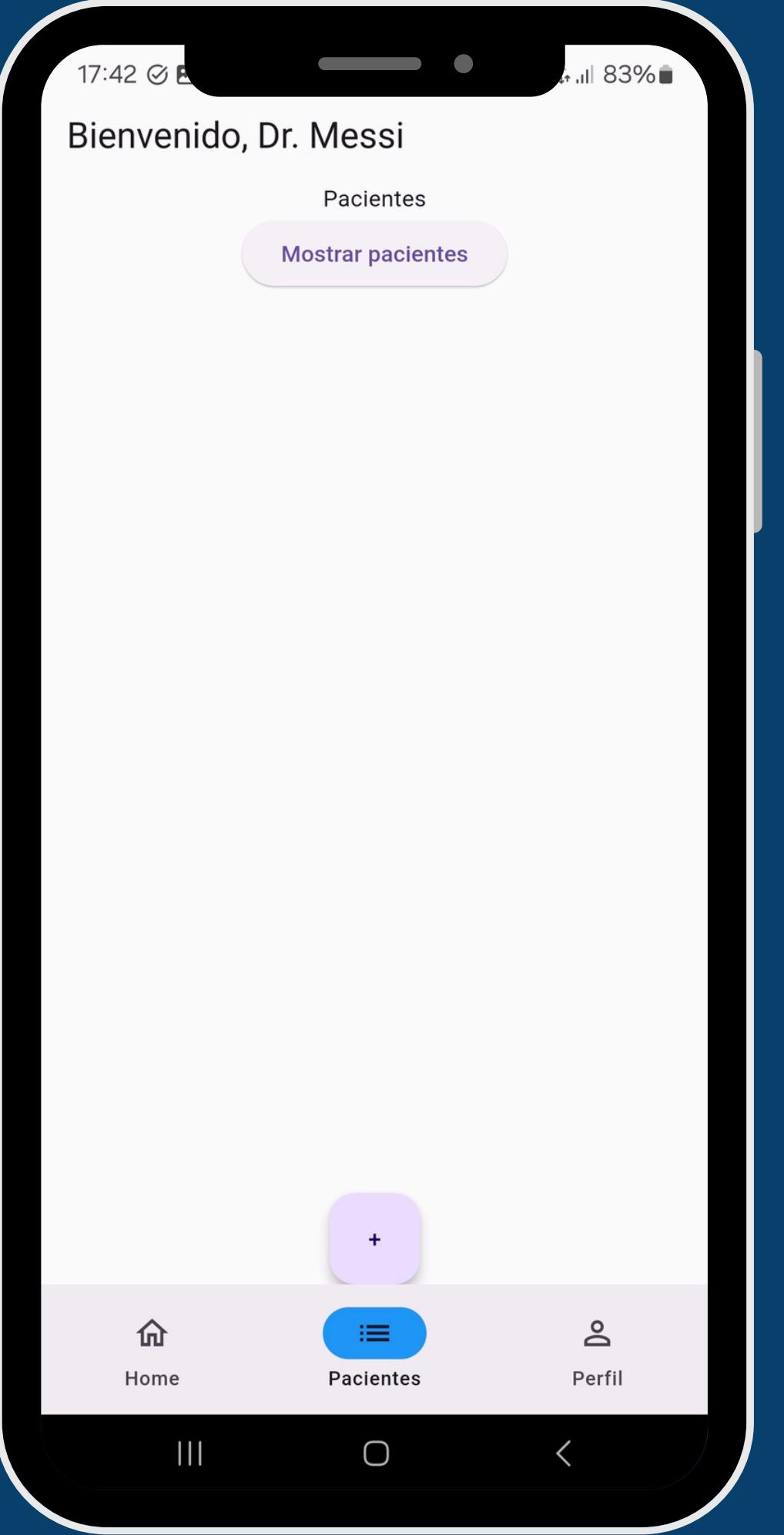
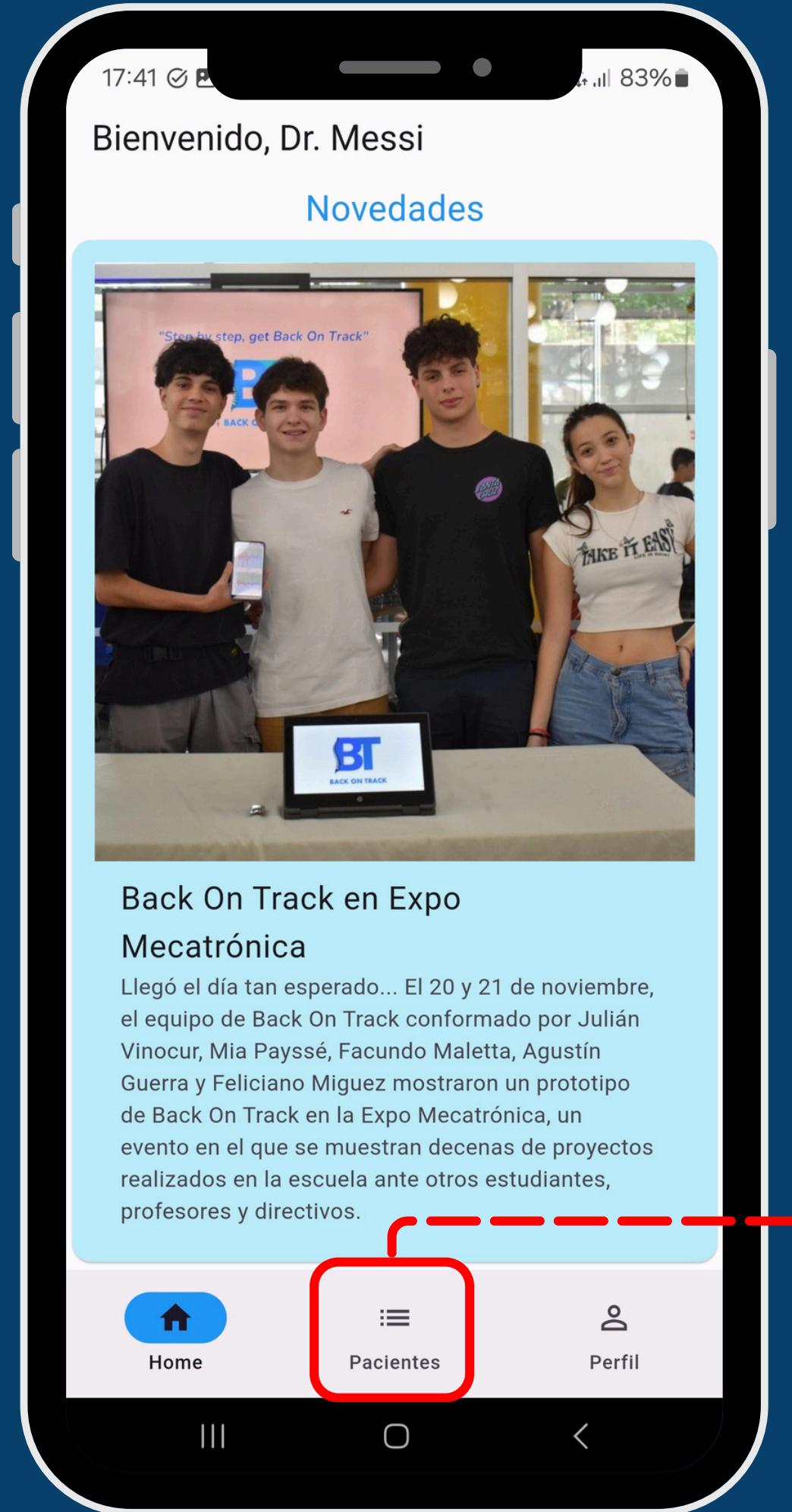












17:42 83%

Bienvenido, Dr. Messi

Pacientes

Mostrar pacientes

+



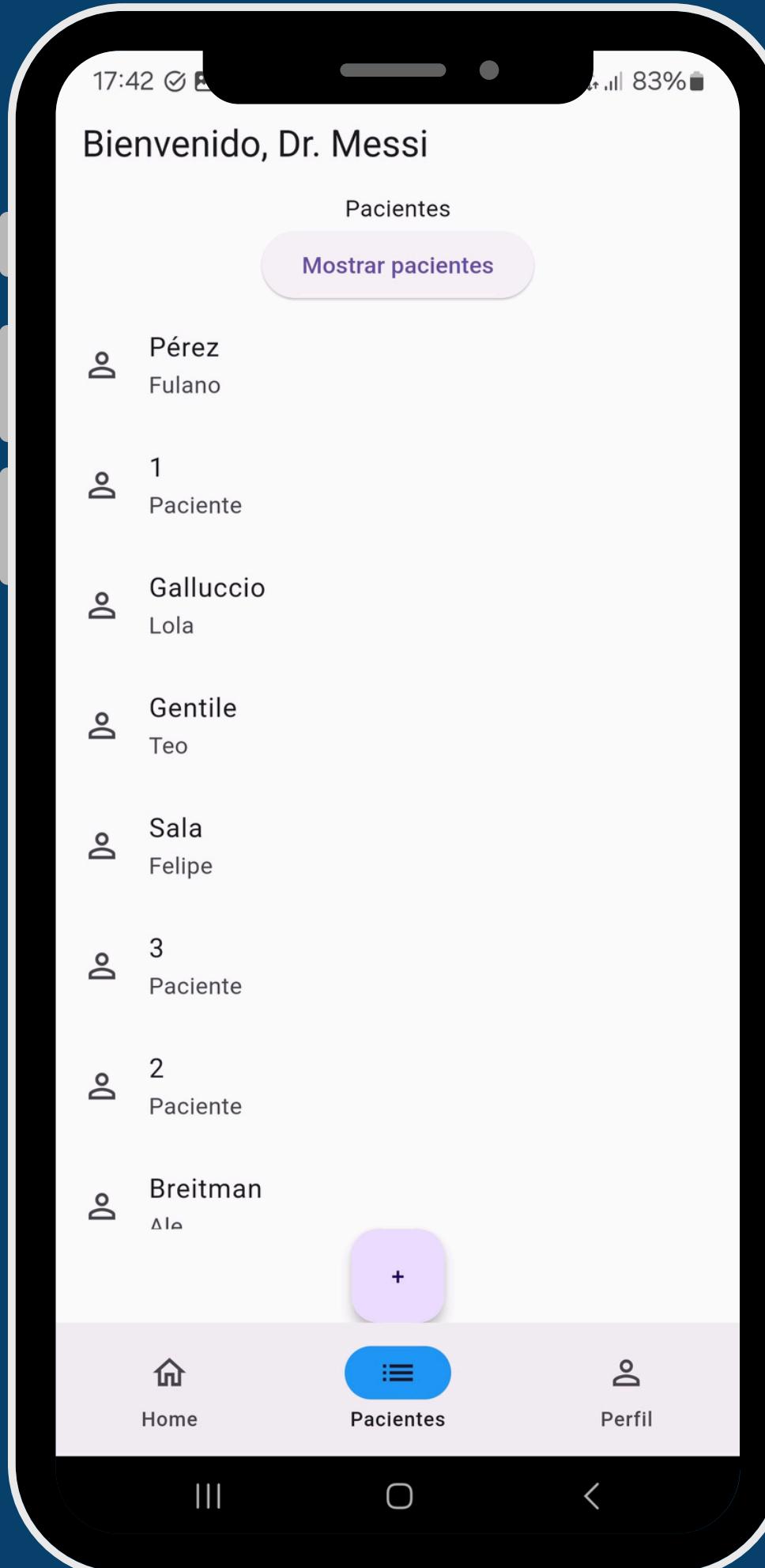
Home



Pacientes



Perfil



The screenshot shows the Google Cloud Firestore console with the path `users > QKP2KIFGLktyZB4QZiCS`. The document details are as follows:

- assignedPatients** (array):
  - 0: "ukmlaNEJSykBmvygrkKm"
  - 1: "9FMkZVkBmvygrkKm"
  - 2: "X0QpdAL4BbM6ylegWPxC"
  - 3: "N3gXrvrSbh5M7qS8bxF8"
  - 4: "KVjAC1MOjYapzqXWwWBN"
  - 5: "FgmrVi204NuICPFqmrtJ"
  - 6: "VQK8N5w4SO81U6qsS1Ez"
  - 7: "HSFW0OLQHkrEzTBKZQI2"
  - 8: "Uq4JupU0sOEiyUG5qT3q"
- email: "ejemplodoctor@gmail.com"
- healthCenter: "Inter Miami"
- isDoctor: true

17:42

83%

Bienvenido, Dr. Messi

Pacientes

Mostrar pacientes

Do Pérez  
FulanoDo 1  
PacienteDo Galluccio  
LolaDo Gentile  
TeoDo Sala  
FelipeDo 3  
PacienteDo 2  
PacienteDo Breitman  
ΔIA

+



Home

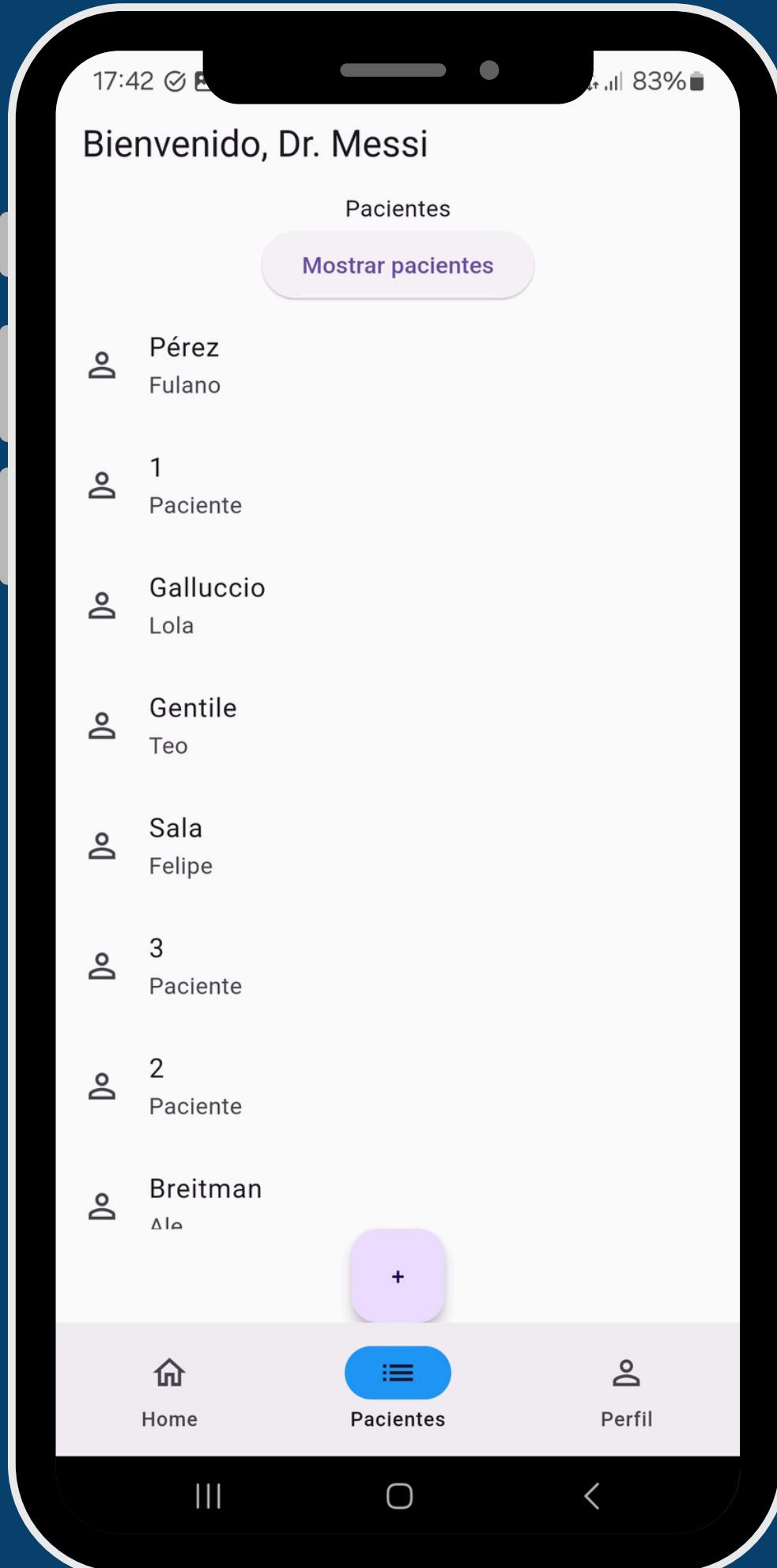


Pacientes

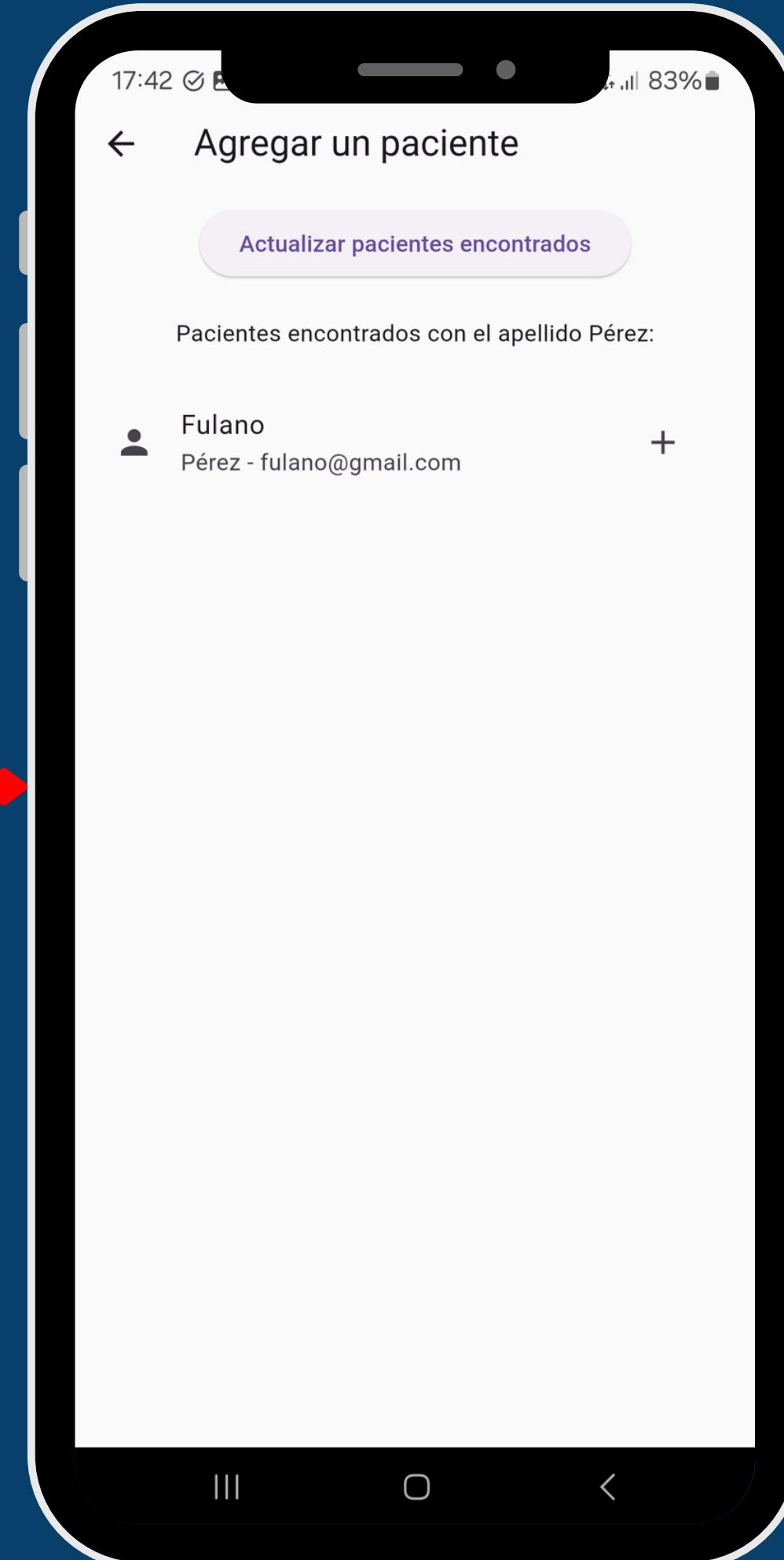
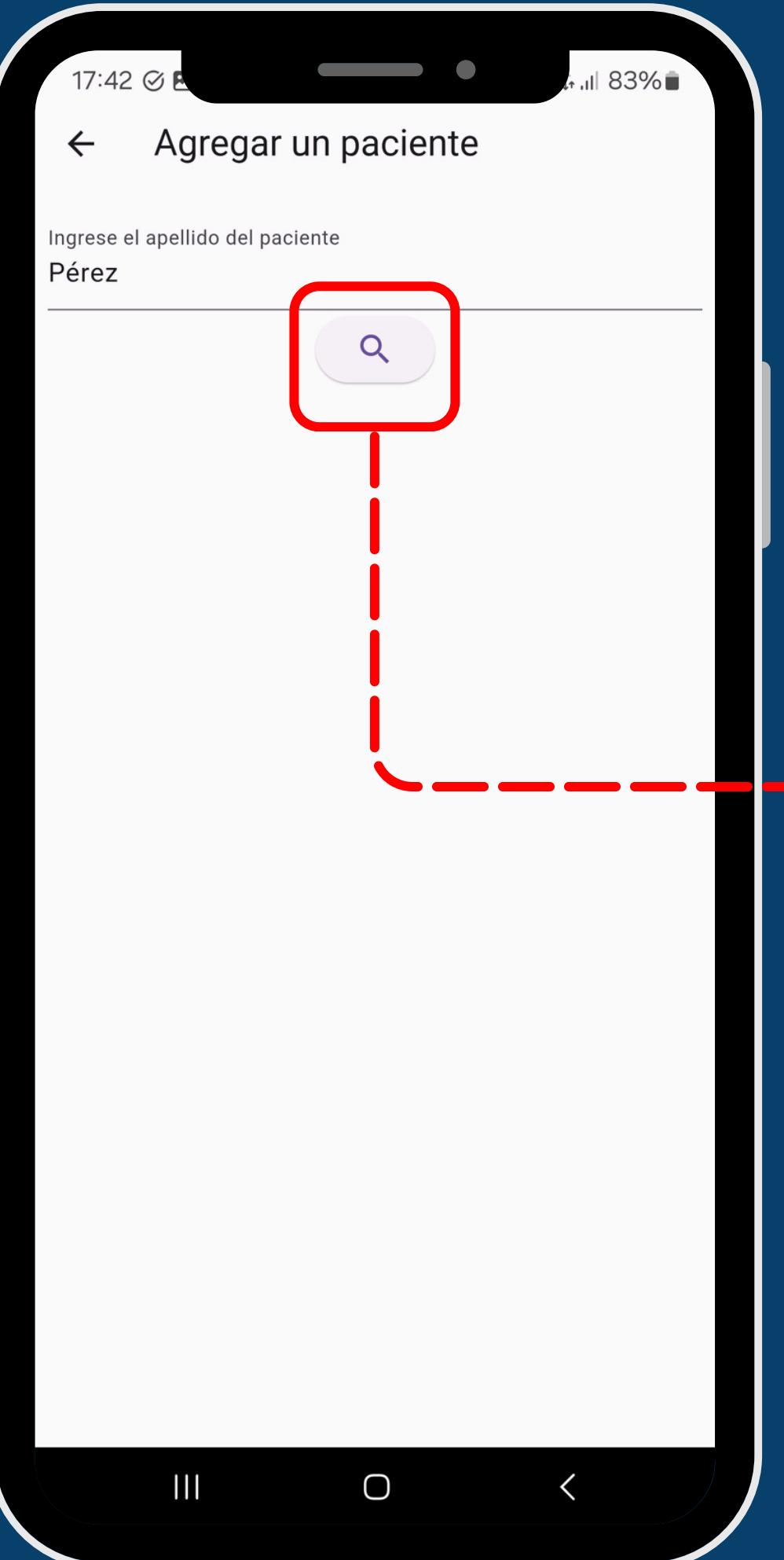
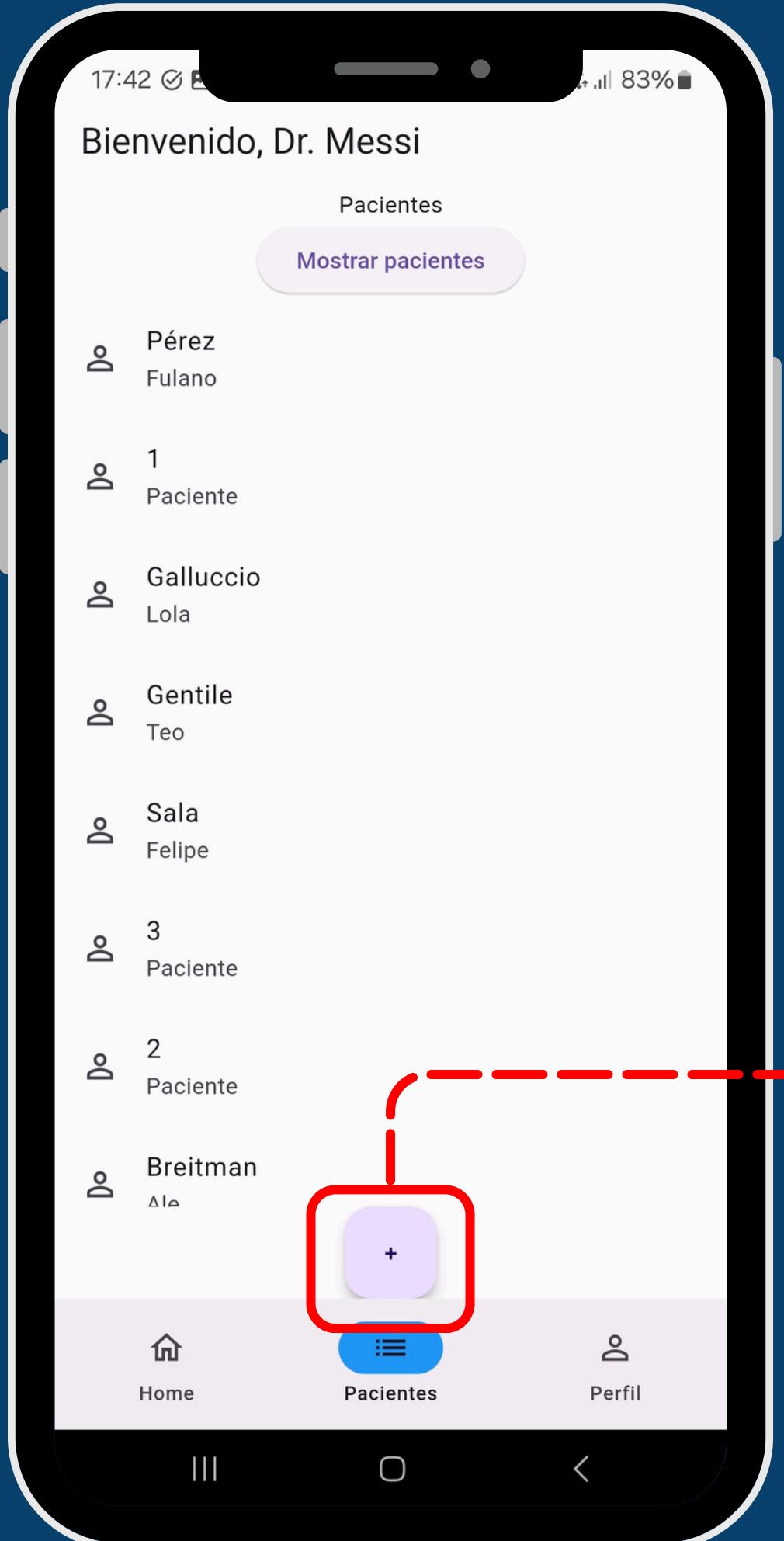


Perfil

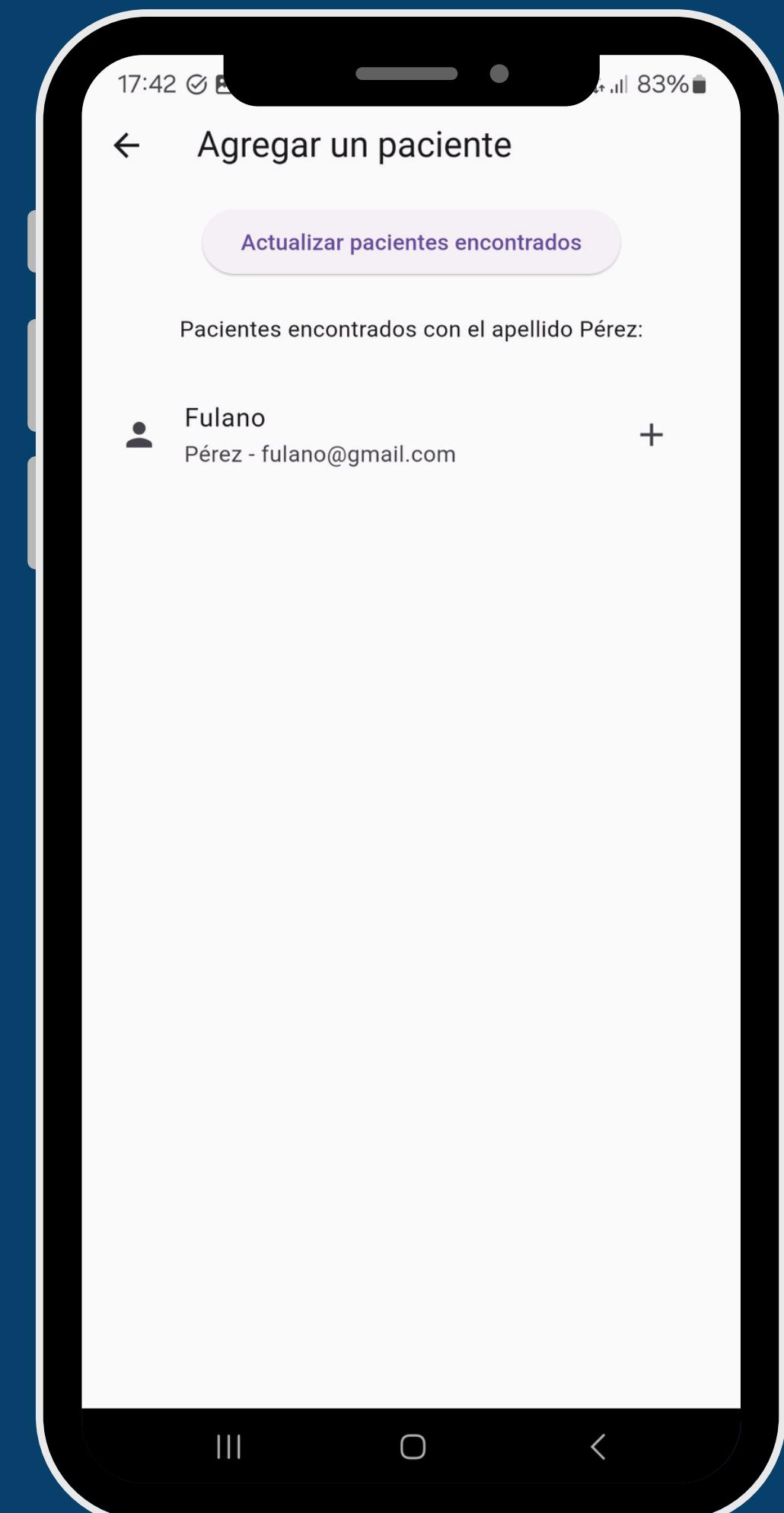
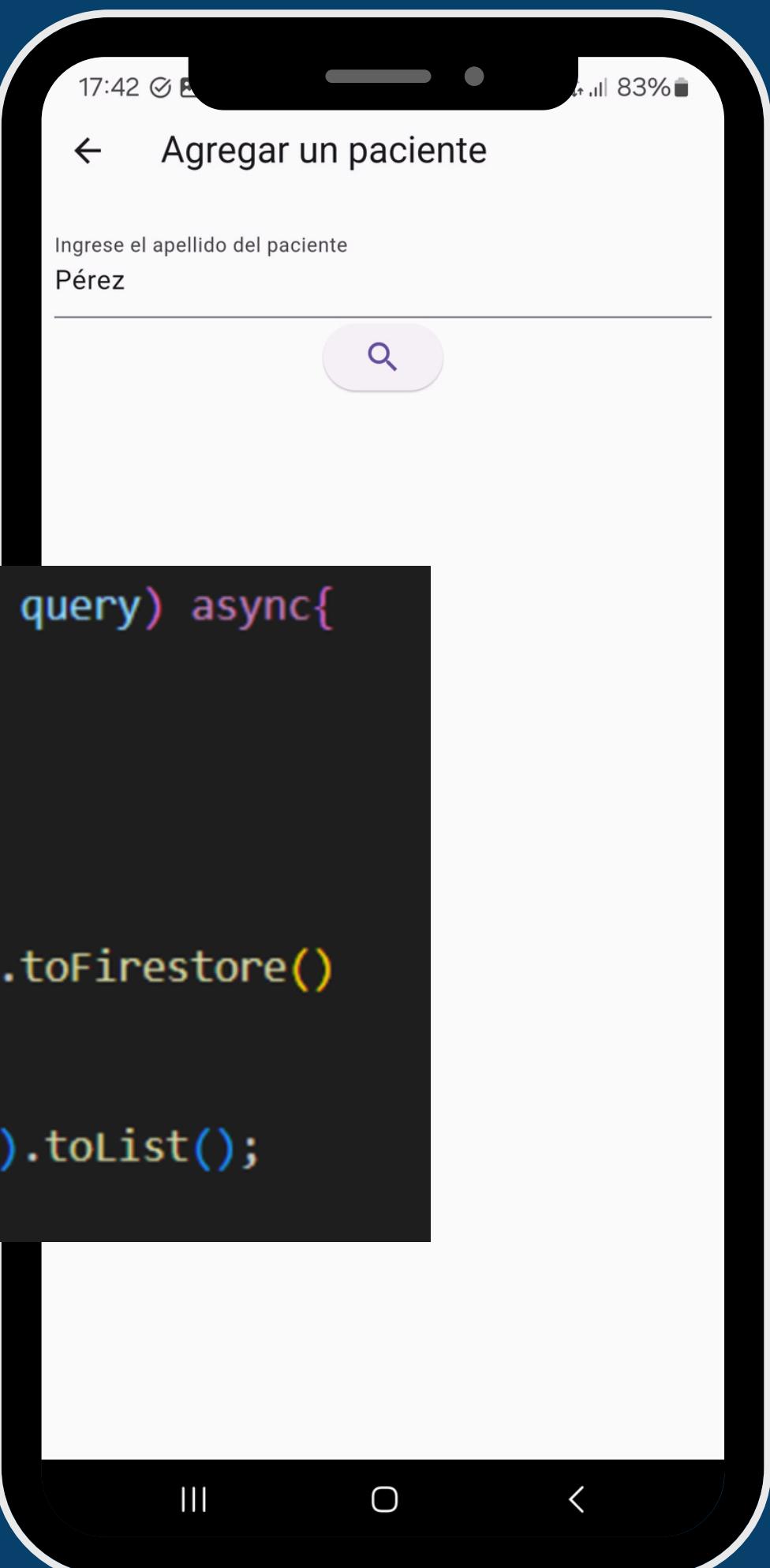
```
Future<void> getAssignedPatientsData() async {  
  
    List<String>? assignedPatientIds = userInfo.assignedPatients;  
  
    final docs = db.collection('users').withConverter(  
        fromFirestore: User.fromFirestore,  
        toFirestore: (User user, _) => user.toFirestore());  
  
    final users = await docs.get();  
  
    List<User> usersList = users.docs.map((d)=>d.data()).toList();  
  
    List<User> auxPatients = usersList.where((user) {  
        return assignedPatientIds?.contains(user.userId) ?? false;  
    }).toList();  
  
    state = auxPatients;  
}
```

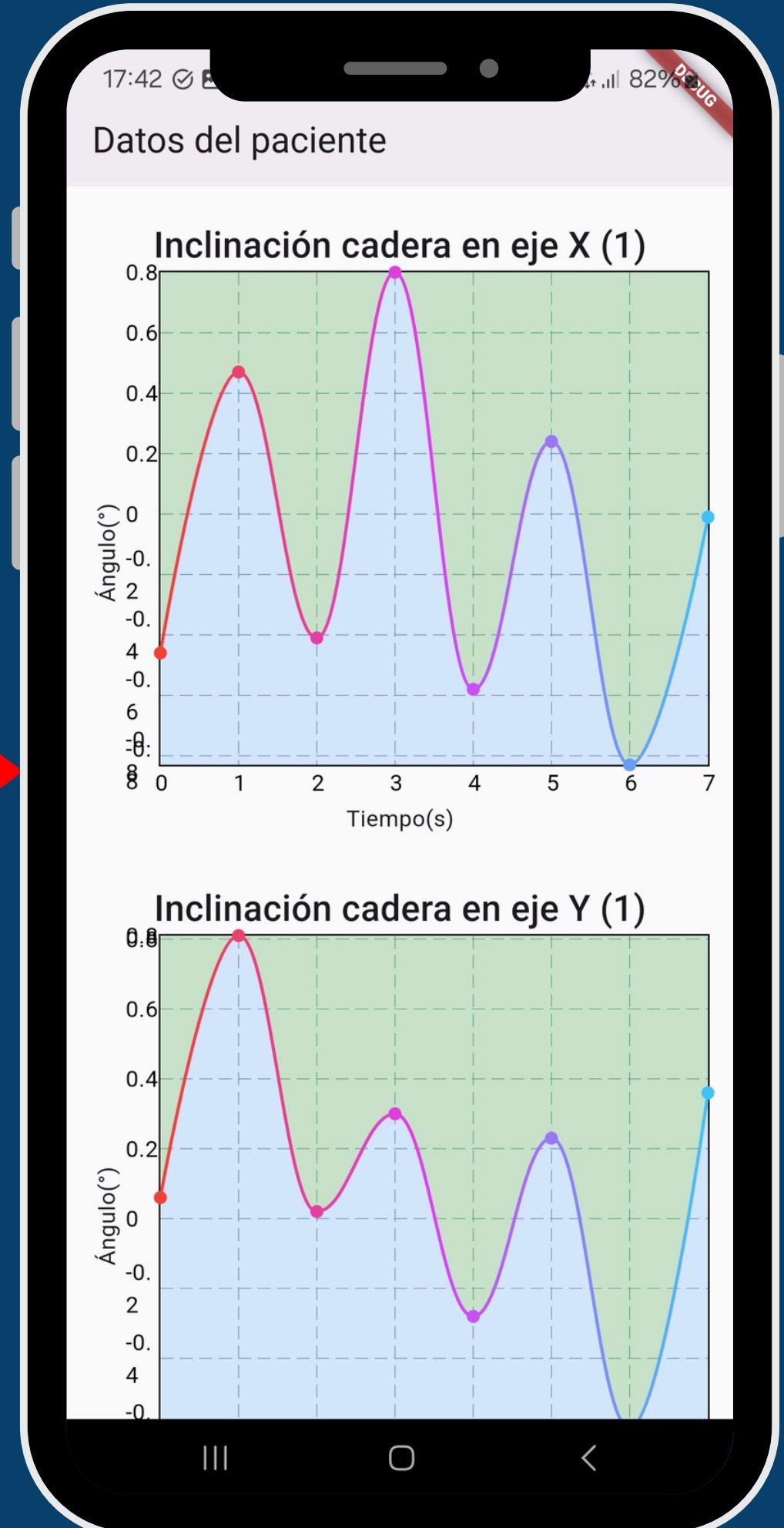
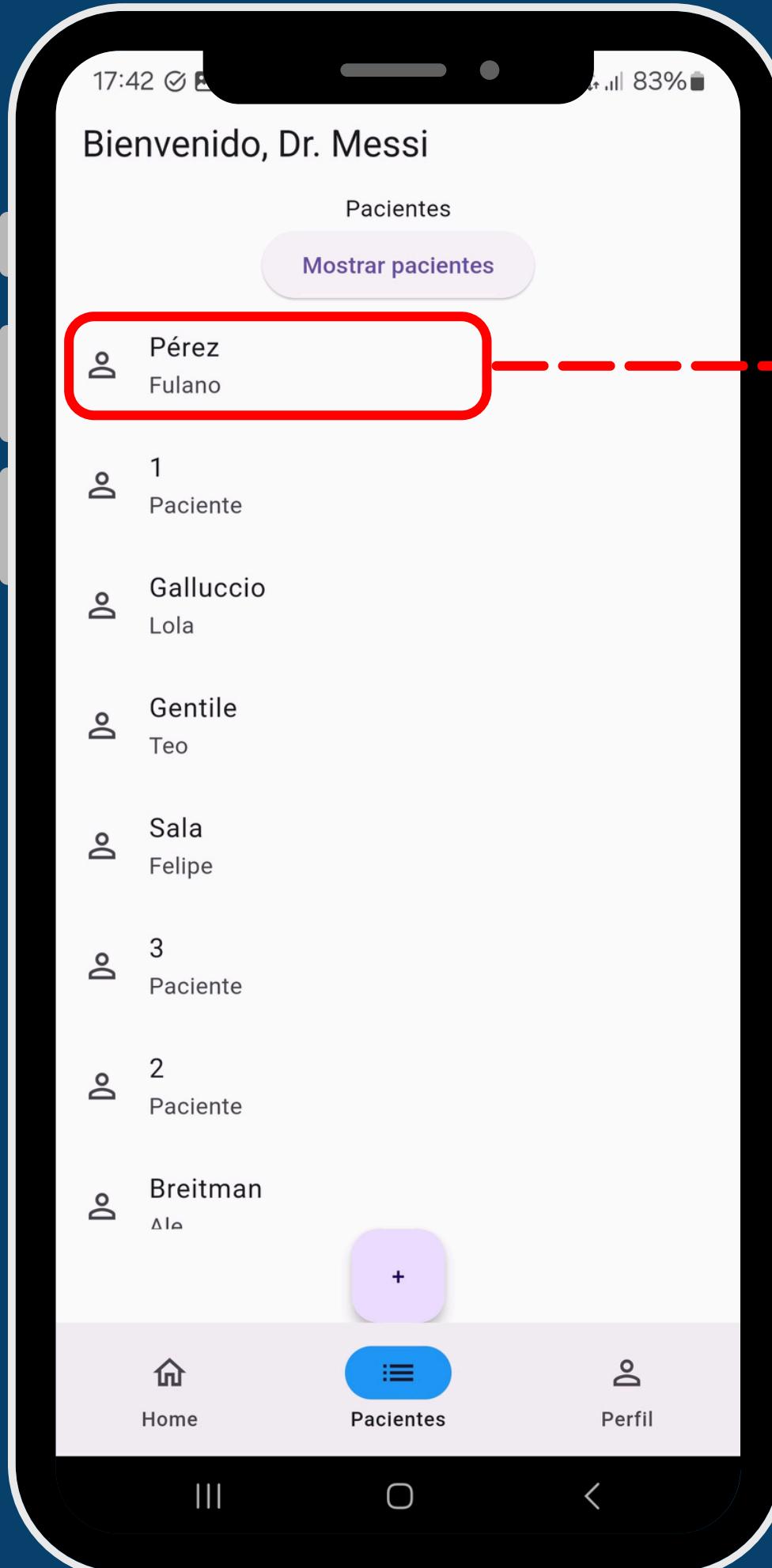


```
factory User.fromFirestore(  
    DocumentSnapshot<Map<String, dynamic>> snapshot,  
    SnapshotOptions? options,  
) {  
    final data = snapshot.data();  
    return User(  
        userId: data['userId'] ?? '', // Agregamos un valor por defecto si es null  
        isDoctor: data['isDoctor'] ?? false, // valor por defecto booleano si es null  
        name: data['name'] ?? '',  
        surname: data['surname'] ?? '',  
        email: data['email'] ?? '',  
        password: data['password'] ?? '',  
        phone: data['phone'] ?? '',  
    );  
}  
  
Map<String, dynamic> toFirestore(){  
    return{  
        "userId": userId,  
        "isDoctor": isDoctor,  
        "name": name,  
        "surname": surname,  
        "email": email,  
        "password": password,  
        "phone": phone,  
    };  
}
```



```
Future<void> getPatientsBySurname(String query) async{
  final docs = db.collection('users')
    .where('surname', isEqualTo: query)
    .where('isDoctor', isEqualTo: false)
    .withConverter(
      fromFirestore: User.fromFirestore,
      toFirestore: (User user,_) => user.toFirestore()
    );
  final users = await docs.get();
  state = users.docs.map((d) => d.data()).toList();
}
```

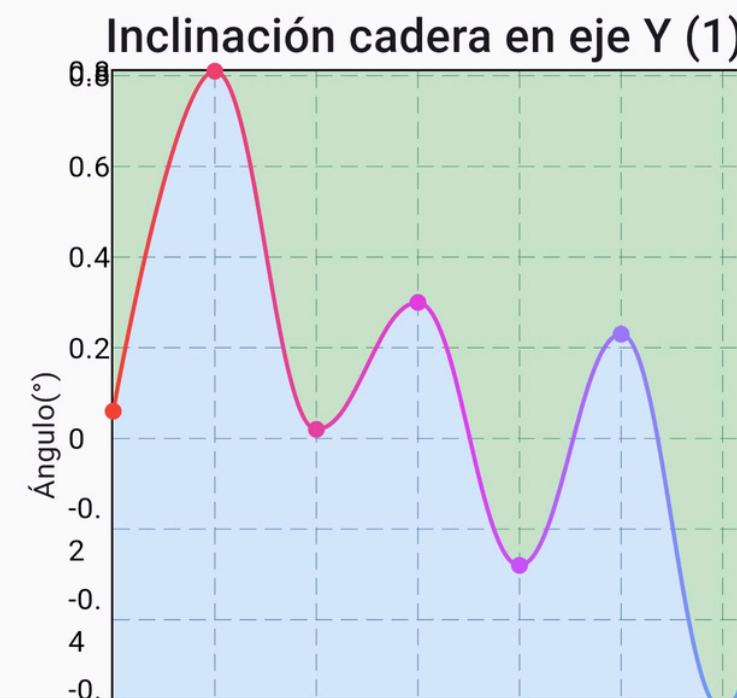
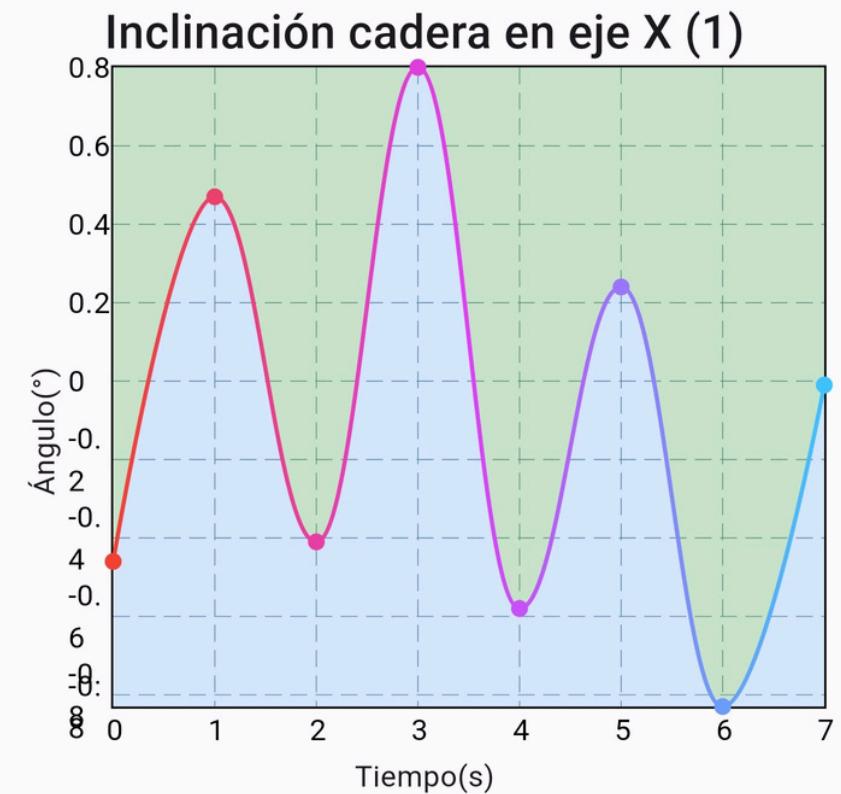




17:42

82%

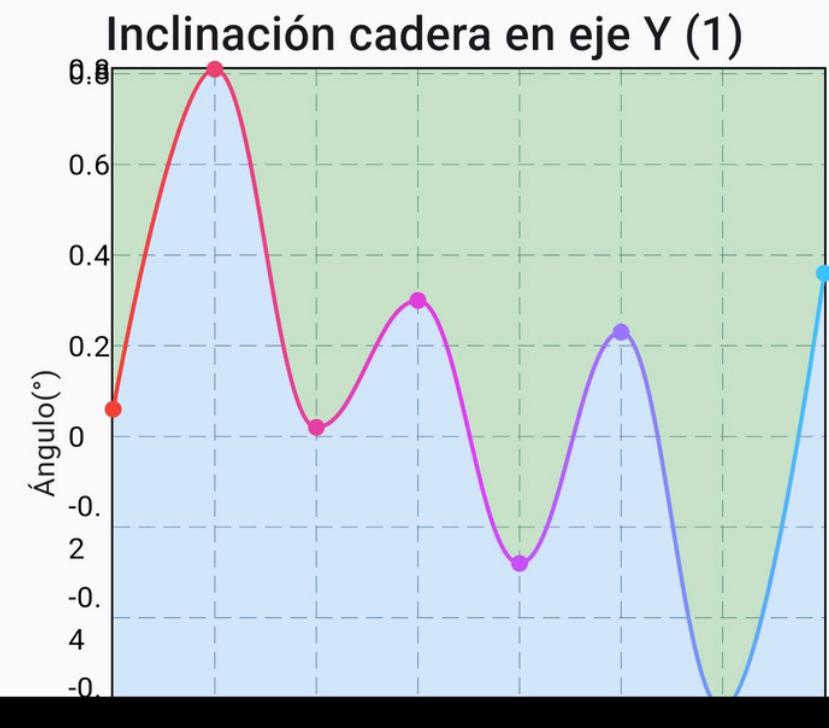
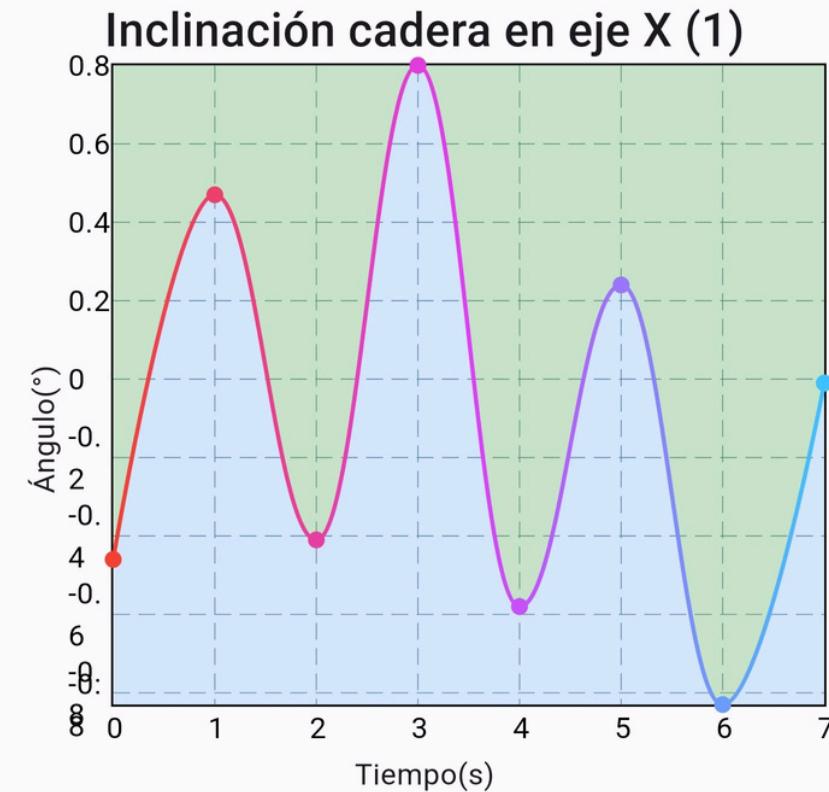
## Datos del paciente



17:42

82%

## Datos del paciente



tracking > ukmlaNEJSykBmvygrkKm

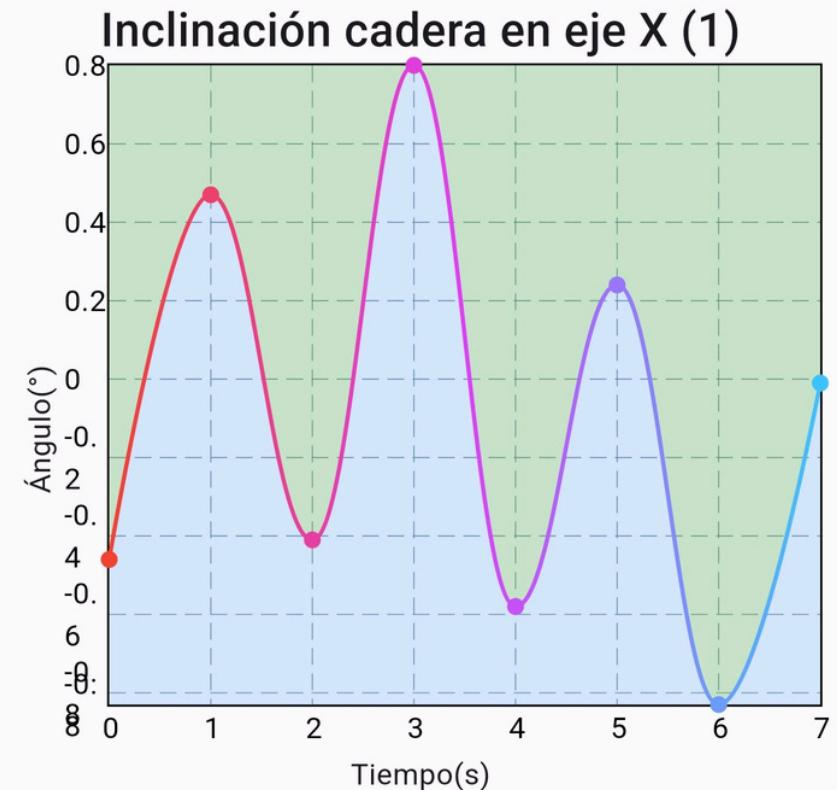
Más funciones en Google Cloud

(default)	tracking	ukmlaNEJSykBmvygrkKm
+ Iniciar colección	+ Agregar documento	+ Iniciar colección
tracking	ukmlaNEJSykBmvygrkKm	+ Agregar campo
ukmlaNEJSykBmvygrkKm		▶ patientId: ["ukmlaNEJSykBmvygrkKm"]
users		▶ timeStamp: ["0", "1732207234", "17322..."]
		▶ valoresFlex1: ["1, 0.00", "2, 0.00", "3,..."]
		▶ valoresFlex2: ["1, 0.00", "2, 0.00", "3,..."]
		▶ valoresX1: ["1, -0.46", "2, 0.47", "3..."]
		▶ valoresX2: ["1, -0.49", "2, 0.03", "3..."]
		▶ valoresy1: ["1, 0.06", "2, 0.81", "3,..."]
		▶ valoresy2: ["1, 0.28", "2, -0.40", "3..."]
		▶ valoresz1: ["1, -0.42", "2, -0.21", "..."]
		▶ valoresz2: ["1, 0.23", "2, 0.27", "3,..."]

17:42

82%

## Datos del paciente



```
class Tracking{
  List patientId;
  List timestamp;
  List z1; //roll
  List z2;
  List y1; //pitch
  List y2;
  List x1; //yaw
  List x2;
  List kneeAngle1;
  List kneeAngle2;
}

Tracking([
  required this.patientId,
  required this.timestamp,
  required this.x1,
  required this.y1,
  required this.z1,
  required this.x2,
  required this.y2,
  required this.z2,
  required this.kneeAngle1,
  required this.kneeAngle2,
]);
```

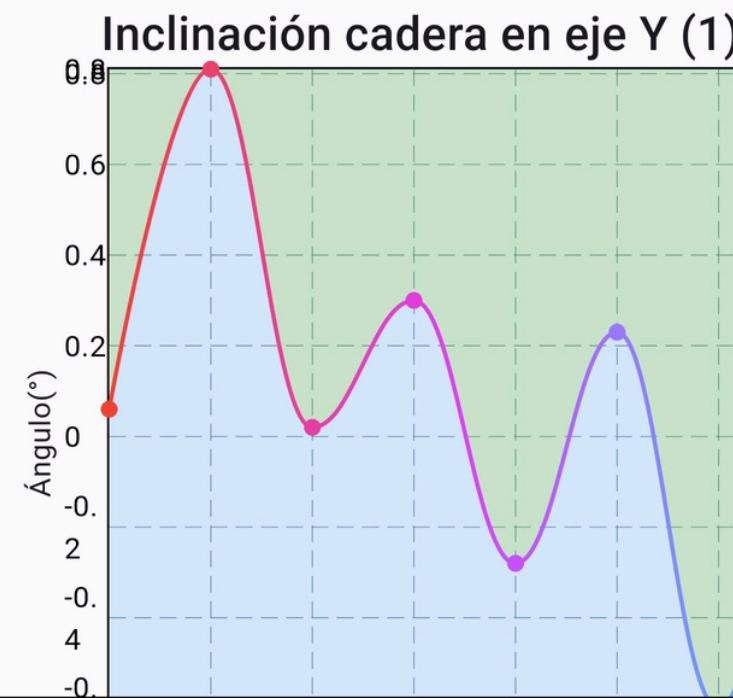
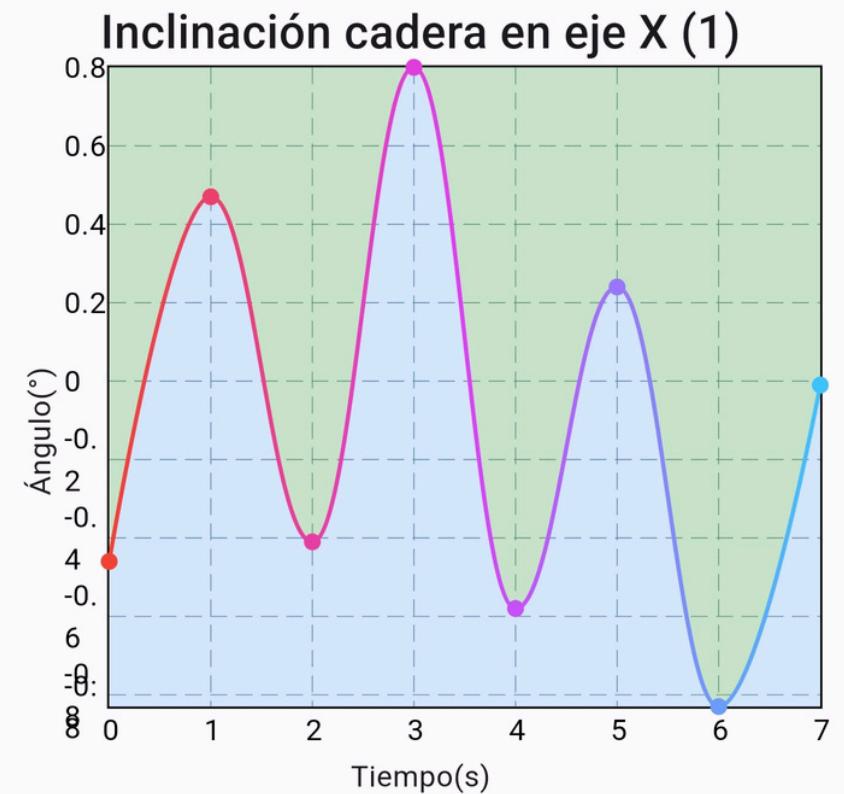
```
factory Tracking.fromFirestore(
  DocumentSnapshot<Map<String, dynamic>> snapshot,
  SnapshotOptions? options,
) {
  final data = snapshot.data();
  return Tracking(
    patientId: data?['patientId'] ?? [],
    timestamp: data?['timeStamp'] ?? [],
    x1: data?['valoresX1'] ?? [],
    y1: data?['valoresy1'] ?? [],
    z1: data?['valoresz1'] ?? [],
    x2: data?['valoresX2'] ?? [],
    y2: data?['valoresy2'] ?? [],
    z2: data?['valoresz2'] ?? [],
    kneeAngle1: data?['valoresFlex1'] ?? [],
    kneeAngle2: data?['valoresFlex2'] ?? [],
  );
}

Map<String, dynamic> toFirestore(){
  return{
    'patientId': patientId,
    'timestamp': timestamp,
    'valoresX1': x1,
    'valoresy1': y1,
    'valoresz1': z1,
    'valoresX2': x2,
    'valoresy2': y2,
    'valoresz2': z2,
    'valoresFlex1': kneeAngle1,
    'valoresFlex2': kneeAngle2,
  };
}
```

17:42

82%

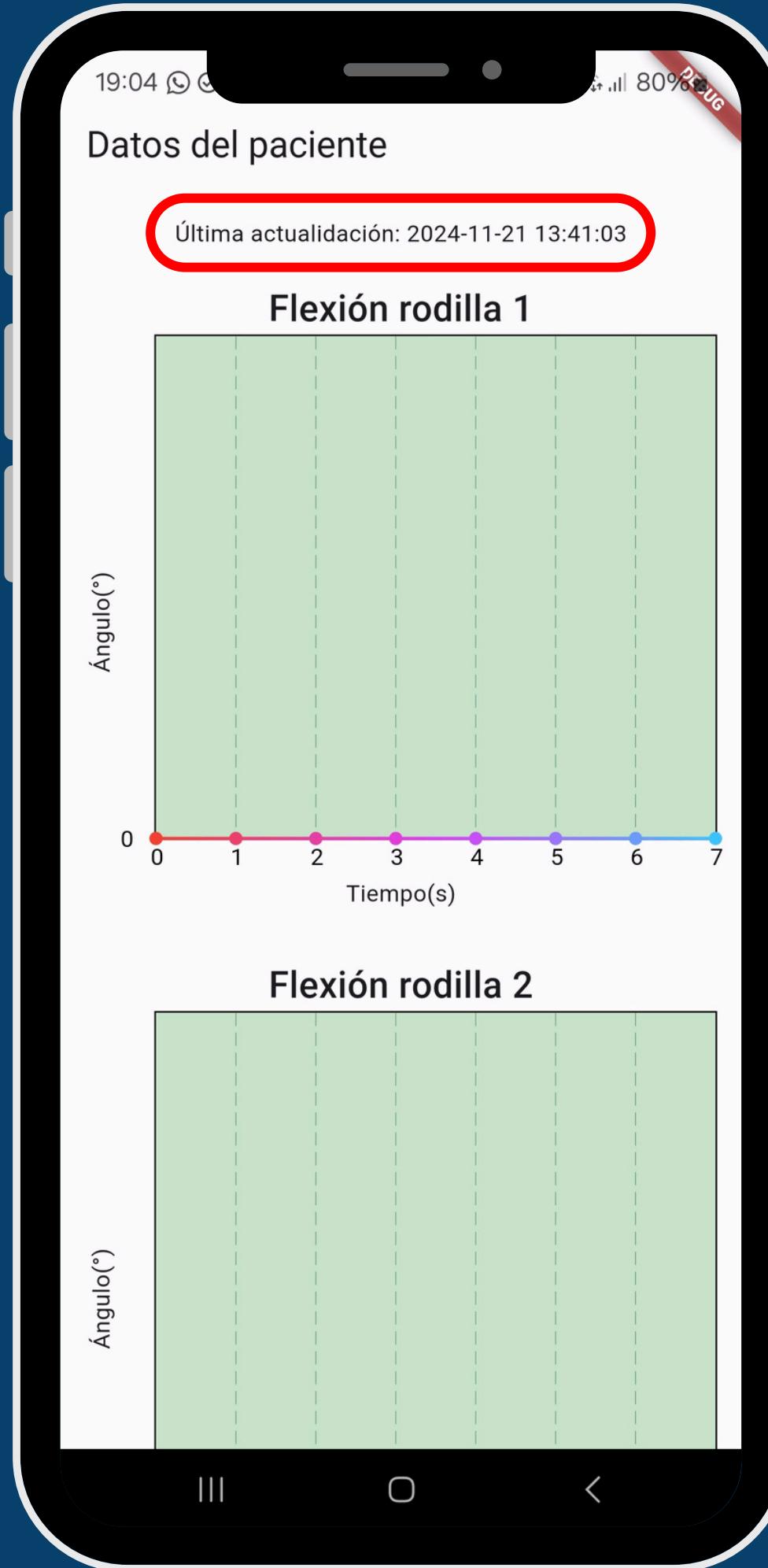
## Datos del paciente



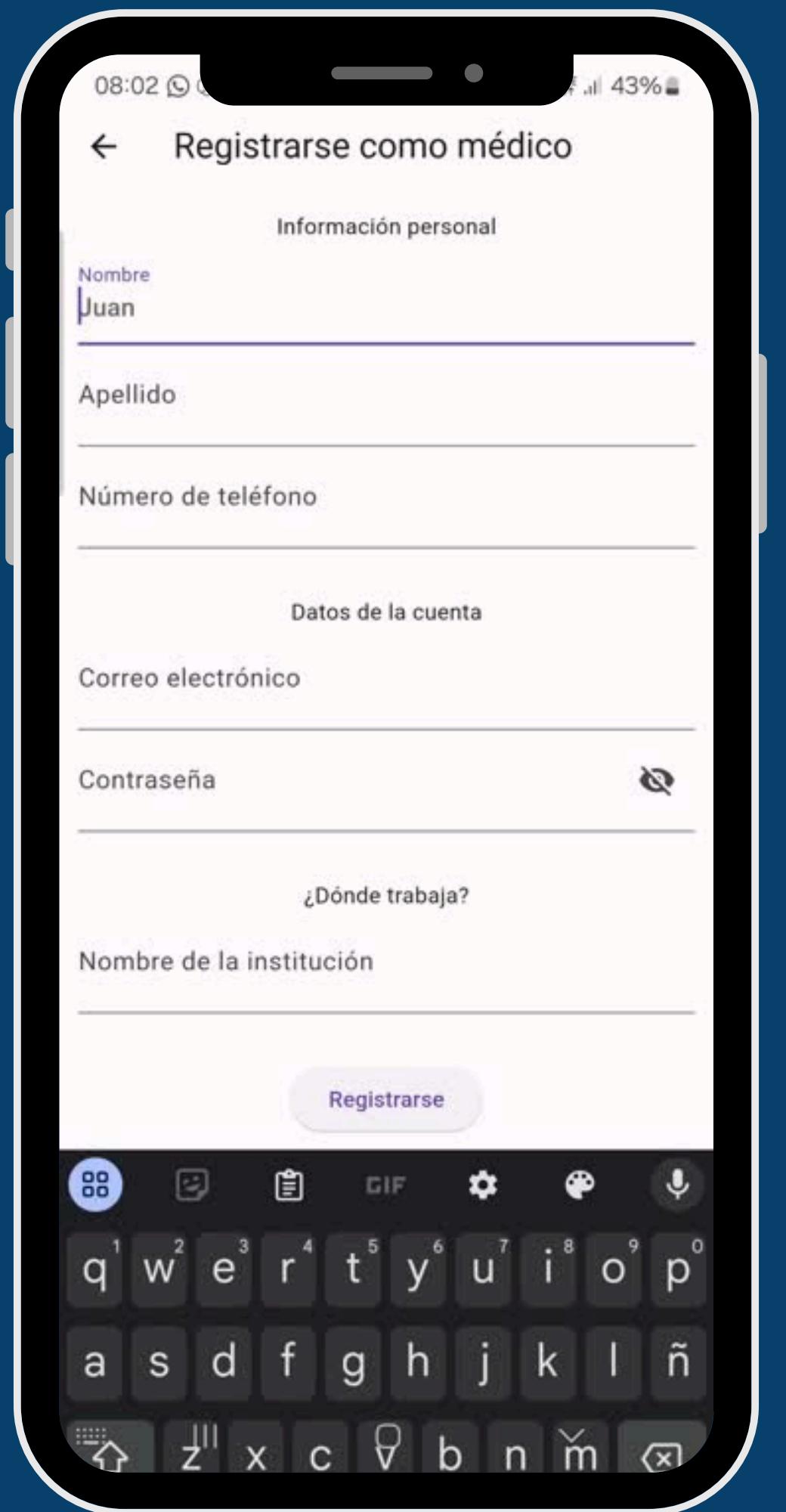
```
for (var i = 0; i < tracking.y1.length; i++) {
    int x = i;
    double y = double.parse(tracking.y1[i].toString().substring(tracking.y1[i].toString().indexOf(' ')));
    final spot = FlSpot(x.toDouble(), y.toDouble());
    spotsy1.add(spot);
}
```

▼ valoresX2 (array) + -

0	"1, -0.49"
1	"2, 0.03"
2	"3, -0.43"
3	"4, 0.71"
4	"5, -0.69"
5	"6, 0.46"
6	"7, -0.61"
7	"8, 0.01"



```
if(tracking.timestamp.isNotEmpty){  
    int timestamp = int.parse(tracking.timestamp[tracking.timestamp.length-1]);  
    date = DateTime.fromMillisecondsSinceEpoch(timestamp*1000);  
    dateFormat = DateFormat('dd-MM-yyyy HH:mm:ss');  
    formattedDate = dateFormat.format(date);
```



Back On Track

**iGracias!**

Mia Payssé, Julián Vinocur, Facundo Maletta,  
Agustín Guerra