



**Instituto Tecnológico y de Estudios Superiores de Monterrey**

**Campus Estado de México**

Implementación de Internet de las cosas (Grupo 501)

***“Reporte Final”***

Andrea Elizabeth Roman Varela      A01749760

Ignacio Solís Montes      A01751213

Arturo Alejandro Paniagua      A01749736

**Profesores:** José Reyes Eslava Zavaleta

David Higuera Rosales

Roberto Martínez Román

30 de noviembre de 2023

## *Índice*

Índice	1
Introducción	2
Administración del proyecto	2
Antecedentes	4
Propuesta de valor	5
KPIs dentro de nuestro proyecto	5
Hardware	6
Software	8
Base de datos	11
Aplicación móvil	15
Video de pruebas	16
Resultados	16
Conclusiones grupales	17
Conclusiones individuales	17
Referencias:	18

## *Introducción*

Internet de las cosas se refiere a un sistema tecnológico de soluciones digitales e inteligentes con la intención de que los objetos se conecten al internet y entre sí, con el fin de facilitar un ecosistema interconectado de dispositivos y máquinas para que los usuarios accedan al control de sus dispositivos desde cualquier lugar. Está compuesto de computación integrada, sistemas ciber físicos, aplicaciones y servicios.

Dentro del bloque de implementación de internet de las cosas el proyecto tiene el objetivo principal de desarrollar un sistema de Internet de las Cosas (IoT) que permita a las personas monitorear su salud personal en tiempo real. Esto se logrará mediante la implementación de sensores de temperatura y frecuencia cardíaca, junto con una plataforma de software que recopilará, procesará y presentará los datos de manera accesible.

## *Administración del proyecto*

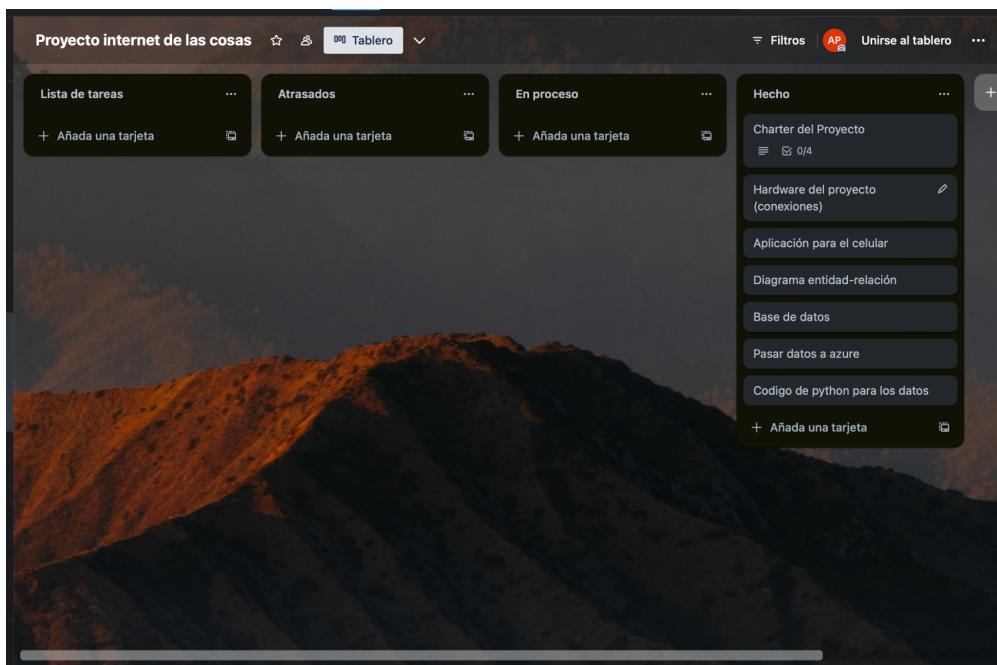


Imagen 1: Plataforma de Trello



## Gantt

### Project

Timeline By Priority All projects

#### Projects



Imagen 2: Diagrama de Gantt

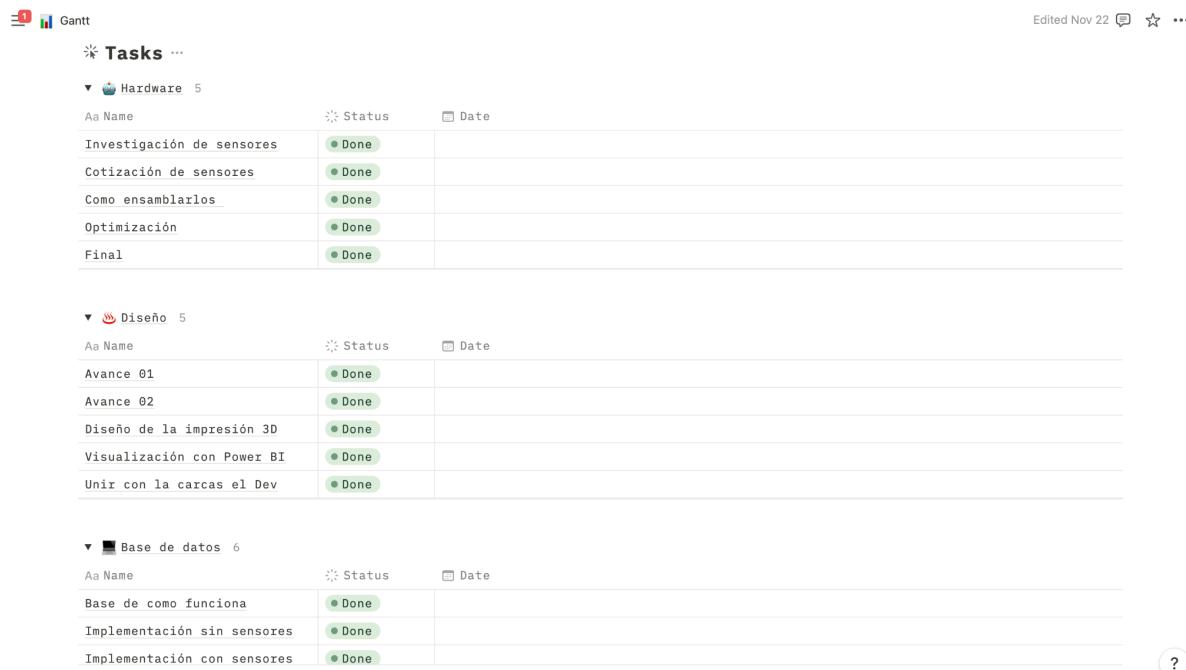


Imagen 3: Diagrama de Gantt

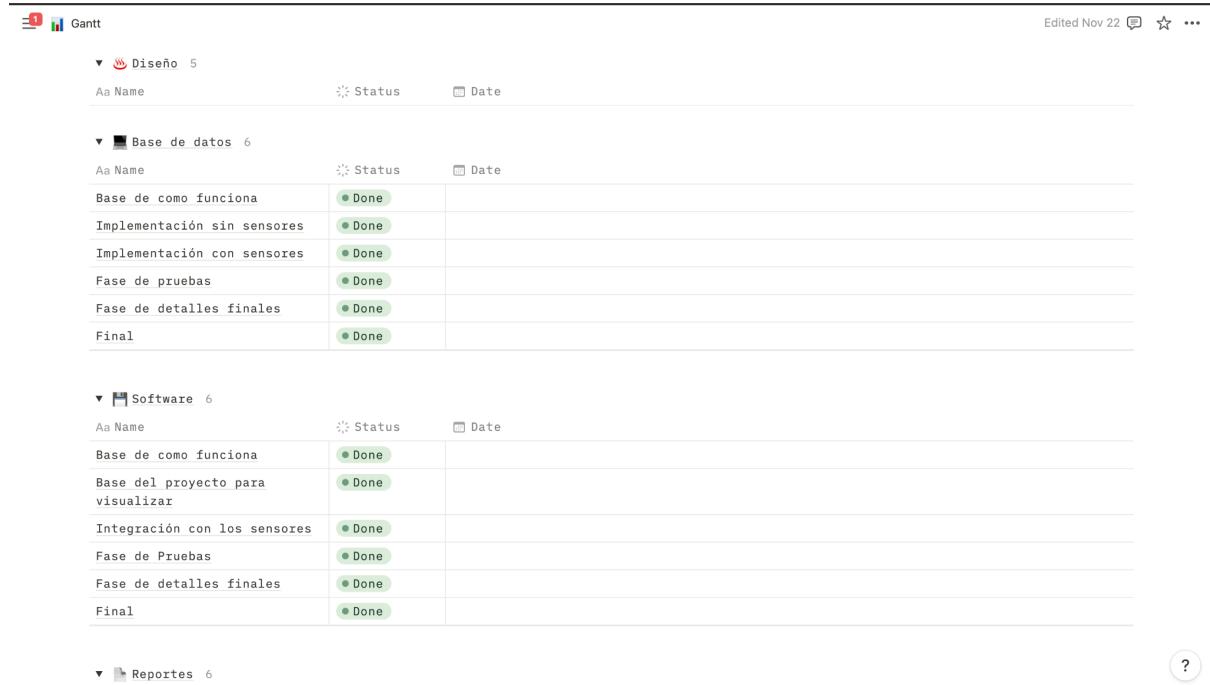


Imagen 4: Diagrama de Gantt



Imagen 5: Diagrama de Gantt

### Antecedentes

El primer avance con los relojes de pulsera se produjo con Pulsar, el primer reloj digital lanzado en el año 1972 de la compañía japonesa Seiko. De aquí a los relojes inteligentes solo pasaron unos años. Comenzamos con el Hewlett Packard HP-01. Un reloj inteligente de 1977 con calculadora, y no muy accesible, por cierto. Para apretar los botones había que utilizar un mini lápiz con punta, porque si no era imposible marcar el número. Además de esto también tenía calendario. Los relojes inteligentes eran todos similares hasta el año 2000. En junio de ese mismo año, IBM mostró un prototipo con sistema operativo de Linux con 6 horas de batería y 8MB de memoria. Este fue mejorado posteriormente con la incorporación de un acelerómetro, un sistema de vibración y un sensor de huella digital.

El crecimiento exponencial en la adopción de dispositivos IoT se ha convertido en una tendencia destacada en los últimos años. Este aumento se debe a la capacidad de estos dispositivos para recopilar y transmitir datos en tiempo real. Además, la preocupación por la

salud personal se ha intensificado a raíz de la pandemia, la cual dejó muchos miedos y una necesidad constante de monitorear la salud personal. El proyecto nace de la necesidad de proporcionar principalmente a las personas que padecen de asma o de problema de frecuencia cardiaca un dispositivo que mida la humedad, la temperatura corporal y la frecuencia cardiaca de manera constante para poder prevenir problemas en el futuro. La detección temprana de problemas de salud como fiebre o cambios en el ritmo cardíaco puede ser crucial para evitar que el problema llegue a un estado más grave.

### *Propuesta de valor*

Nosotros proponemos crear una pulsera inteligente que cuente con 3 sensores que midan la temperatura corporal, la frecuencia cardiaca y la humedad, tomando en cuenta que una detección temprana de enfermedades puede evitar incidentes mayores y que las personas que padecen de asma o enfermedades crónicas requieren monitoreo constante. Lo que nos diferencia del resto es nuestra aplicación porque sabemos que muchas veces un adulto quiere monitorear a un miembro de la familia y nosotros lo permitimos ya que dentro de nuestra aplicación pueden conectar varias pulseras a la misma aplicación con la intención de que un amigo o familiar también pueda recibir las alertas.

### *KPIs dentro de nuestro proyecto*

La importancia del uso de KPIs para este proyecto es poder medir el éxito y rendimiento del proyecto, optimizar los recursos ya que en este caso el tiempo para realizar el dispositivo era limitado, ayudan a la identificación de problemas y oportunidades para el dispositivo, algunos de los KPIs dentro de nuestro proyecto son:

- Precisión de las mediciones: Precisión en las mediciones de frecuencia cardiaca, temperatura y humedad en comparación con otros métodos
- Confiabilidad del dispositivo: Tasa de fallos para evaluar qué tan confiable es el dispositivo
- Facilidad de uso: Número de usuarios que encuentran fácil usar el dispositivo y la aplicación
- Conectividad: Porcentaje de tiempo que permanece conectada a la aplicación
- Adopción del usuario: Usuarios que han adoptado y utilizan activamente el dispositivo
- Seguridad: Nivel de seguridad implementado para proteger los datos

### *Hardware*

Para poder realizar este proyecto ocupamos un DevKit 2.0 proporcionado por el profesor David Higuera, un sensor de humedad (DHT11), un sensor de temperatura corporal (LM35) y un sensor de frecuencia cardíaca (XD58C) con la intención de medir la temperatura y frecuencia cardíaca de una persona, además de la humedad. Realizamos las conexiones del DevKit y los sensores a una protoboard con la ayuda de jumpers, aunque para poder realizar las conexiones de los sensores se llevó a cabo una práctica para conectar 3 sensores proporcionados por el profesor David Higuera al DevKit, también diseñamos en Tinkercad una caja la cual mandamos a imprimir en 3D para contener tanto el DevKit como el protoboard y los sensores.

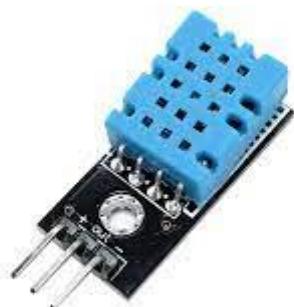


Imagen 6: Sensor de humedad DHT11

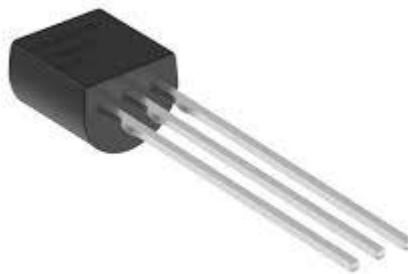


Imagen 7: Sensor de temperatura LM35



Imagen 8: Sensor de frecuencia cardíaca XD58C

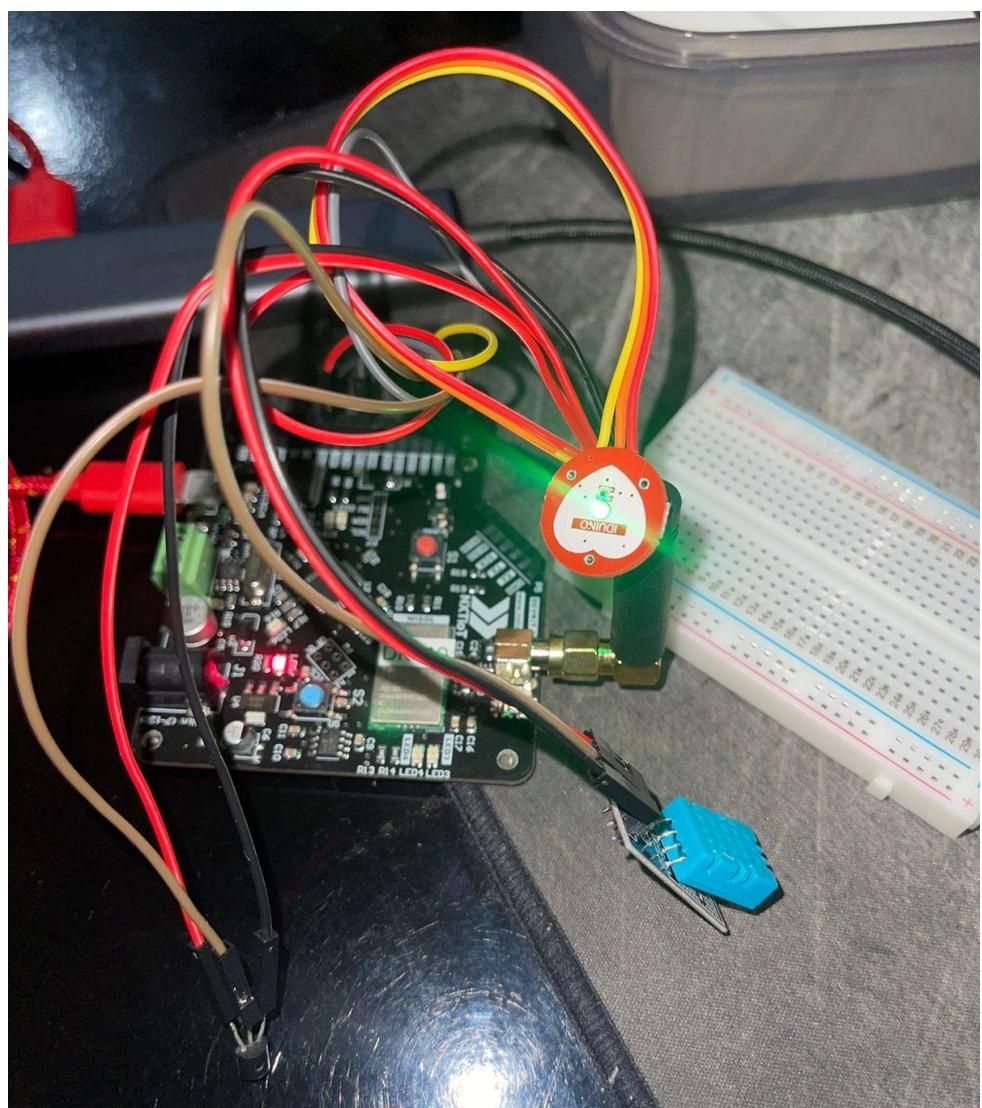


Imagen 9: Prototipo terminado

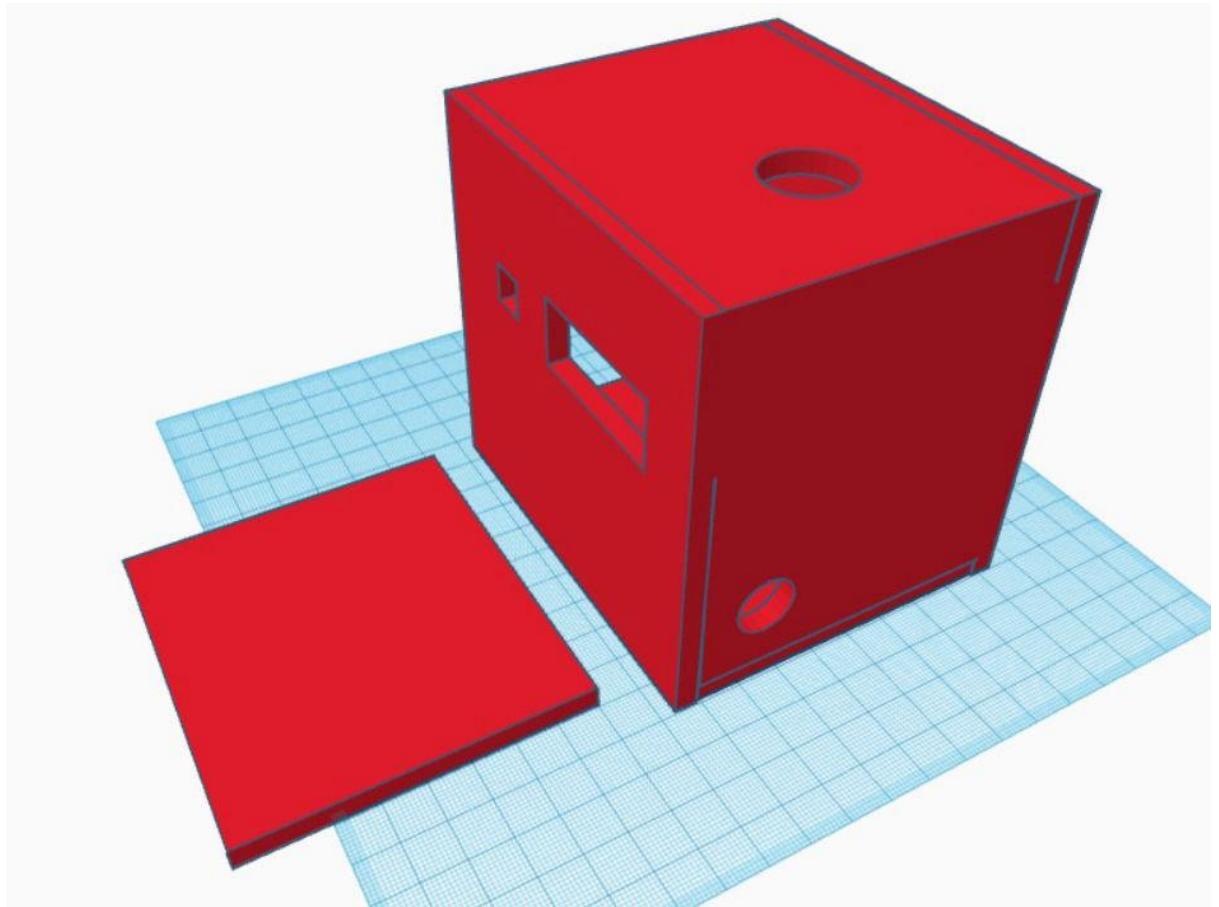


Imagen 10: Modelación 3D de la caja

### *Software*

Posteriormente con el prototipo terminado se creó el código en arduino para que los sensores proporcionarán los datos tanto en el monitor serial como en la plataforma de Sigfox. Para que los datos se mostrarán de la manera deseada en la plataforma realizamos una práctica con el objetivo de aprender a configurar los callbacks para que al momento de mostrarlos en la plataforma fuera en el formato correcto para facilitar la lectura. A continuación se programó un código de Python con la finalidad de pasar los callbacks de la plataforma de Sigfox a la plataforma de Microsoft Azure.

```

hc-sr04.ino    NXTIoT_dev.h
1 #include <NXTIoT_dev.h>
2 #include "DHT.h"
3
4 NXTIoT_dev mysigfox;
5 const int boton = 6;
6 const int sensorPin1 = A0; // Pin del primer sensor analógico
7 const int sensorPin2 = A1; // Pin del segundo sensor analógico
8 const int sensorPin3 = A2; // Pin del sensor DHT11
9
10 DHT dht(sensorPin3, DHT11);
11
12 float temp; // Variable para almacenar la temperatura del primer sensor
13 float humedad; // Variable para almacenar la humedad del sensor DHT
14 int promedioFrecuencia; // Variable para almacenar el promedio de frecuencia cardíaca
15
16 void setup()
17 {
18     Serial.begin(9600);
19     pinMode(boton, INPUT);
20     dht.begin();
21 }
22
23 void leer_temperatura()
24 {
25     // Código para el primer sensor de temperatura
26     int sensorVal = analogRead(sensorPin1);
27     float voltaje = (sensorVal / 1023.0) * 5;
28     Serial.print("Voltaje: ");
29     Serial.println(voltaje);
30     Serial.print("Grados Cº: ");
31     temp = (voltaje) * 100;
32     Serial.println(temp);
33 }
34
35 void leer_temperatura_dht()
36 {
37     // Código para el sensor de temperatura DHT
38     humedad = dht.readHumidity();
39     float temperatura = dht.readTemperature();

```

Imagen 11: Código arduino

Output    Serial Monitor X

Message (Enter to send message to 'Arduino Uno' on '/dev/cu.usbserial-AL028DNF')

```

17:51:54.332 -> Heart rate: 90
17:51:54.364 -> Heart rate: 91
17:51:54.364 -> Heart rate: 91
17:51:54.364 -> Heart rate: 93
17:51:54.396 -> Promedio de las 5 mediciones: 91
17:51:56.371 -> AT$RC
17:51:56.371 -> AT$SF=e0804341000080425b
17:52:01.970 -> Voltaje: 0.34
17:52:01.970 -> Grados Cº: 33.72
17:52:02.492 -> Humedad: 65%, Temperatura: 17.00°C
17:52:02.985 -> Heart rate: 95
17:52:03.018 -> Heart rate: 97
17:52:03.018 -> Heart rate: 96
17:52:03.050 -> Heart rate: 97
17:52:03.084 -> Heart rate: 97
17:52:03.084 -> Promedio de las 5 mediciones: 96
17:52:05.061 -> AT$RC
17:52:05.061 -> AT$SF=bae506420000824260

```

Imagen 12: Monitor serial con mediciones de sensores

				Base station reception attributes					
Time	Delay (s)	Seq Num	Data / Decoding	Station	RSSI (dBm)	Freq (MHz)	LQI	Callbacks	Location
2023-11-29 18:23:38	< 1	123	1f7e9841000078424d temperatura: 19.061583 humedad: 62.0 frecuencia: 77	79AE	-99.00	902.2689			
			30C2	-121.00	902.1935				
			3187	-116.00	902.1934				
2023-11-29 18:23:29	< 1	122	da69c741000080424d temperatura: 24.926685 humedad: 64.0 frecuencia: 77	79AE	-103.00	902.2569			
			3173	-130.00	902.1963				
			7ADE	-109.00	902.2570				
2023-11-29 18:23:08	< 1	121	1f7e984100007c425a temperatura: 19.061583 humedad: 63.0 frecuencia: 90	79AE	-104.00	902.2683			
			31F2	-125.00	902.1847				
			7ADE	-108.00	902.2684				
2023-11-29 18:22:52	< 1	120	a69a6a41000078424f temperatura: 14.662756 humedad: 62.0 frecuencia: 79	3173	-127.00	902.1819			
			79AE	-102.00	902.1813				
			7ADE	-116.00	902.1814				

Imagen 13: Callbacks en plataforma sigfox

```

final.py > on_event
1 EventHubConsumerClient
2
3
4 s://iotfinal.servicebus.windows.net/;SharedAccessKeyName=RootManageSharedAccessKey;SharedAccessKey=5CdhojINV68zosztpxLqhhBnTiiw5s10B+AEhCAMgSY=
5
6 eventos
7 ext, event):
8 mensaje
9 rom partition: {}."format(partition_context.partition_id))
10 .format(event.body_as_str())
11
12 _checkpoint(event)
13
14 idor de Event Hub
15 ent.from_connection_string(CONNECTION_STR, consumer_group="$Default", eventhub_name=EVENTHUB_NAME)
16
17
18 manera asincrónica
19
20
21 it,
22 on=-1, # "-1" es desde el inicio del stream
23
24 el usuario")
25
-

```

PROBLEMS OUTPUT TERMINAL JUPYTER

DEBUG CONSOLE TERMINAL

```

"device": "device",
"data": "data",
"seq_number": 122,
"device_type_id": "61645b12853343523f5b49a3",
"temperatura": 24.926685,
"humedad": 64.0,
"frecuencia": 77,
"date": 1701303809
}
Received event from partition: 0.
Message body:
{
"device": "device",
"data": "data",
"seq_number": 123,
"device_type_id": "61645b12853343523f5b49a3",
"temperatura": 24.061583,
"humedad": 62.0,
"frecuencia": 77,
"date": 1701303818
>

```

Imagen 14: Código en Python para mandar eventos



Imagen 15: Datos en la plataforma Microsoft Azure

### Base de datos

Se realizó un diagrama entidad- relación para poder crear la base de datos de una manera acertada, con la ayuda constante del profesor Roberto Martínez para obtener el mejor resultado posible. Dentro de la base de datos encontramos 4 tablas, la primera es la tabla de frecuencia cardiaca donde encontramos los atributos de idSensor, noMedicion, valor, fecha, hora, de las cuales noMedicion es la llave primaria y la llave foránea es idSensor. En las tablas de humedad y temperatura encontramos los mismos atributos con las mismas

características. Por último, encontramos la tabla de sensor con los atributos de idSensor siendo este la llave primaria y el atributo de tipo.

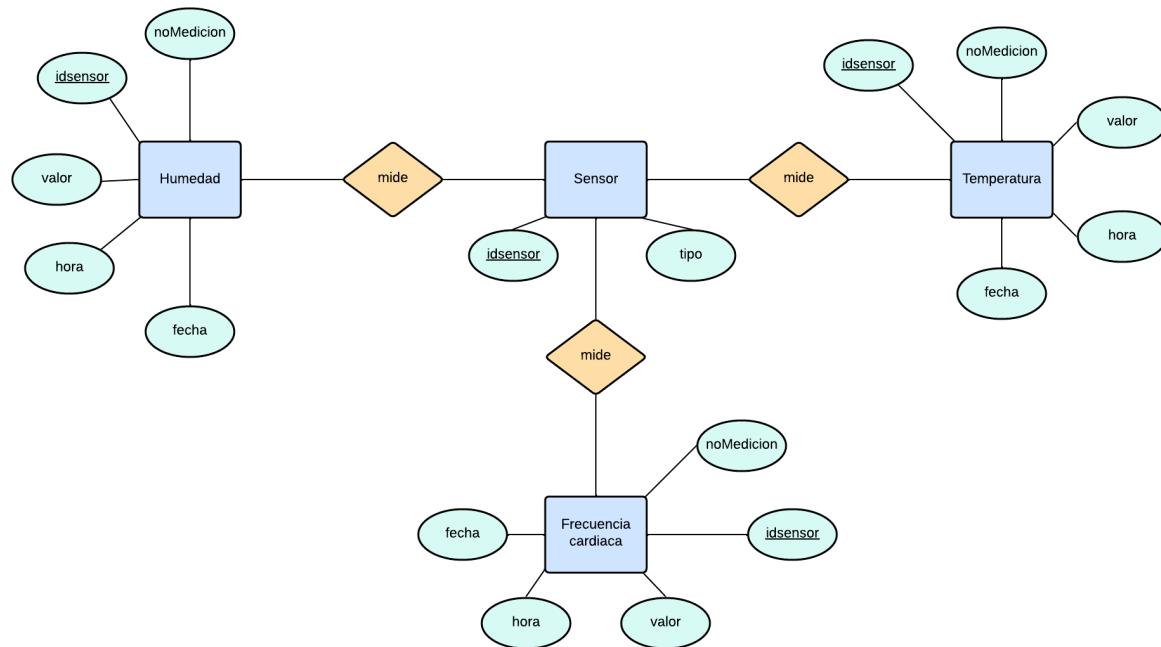


Imagen 16: Diagrama entidad-relación

La captura de pantalla muestra la interfaz de MySQL Workbench para la base de datos iotdatabase. Se observa la siguiente información:

- Barra superior:** Muestra el servidor (localhost), la base de datos (iotdatabase) y un menú con opciones como Estructura, SQL, Buscar, Generar una consulta, Exportar, Importar, Operaciones, Privilegios, Rutinas, Eventos y Más.
- Panel central:** Muestra la lista de tablas existentes:
 

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
frecuencia_cardiaca	Examinar, Estructura, Buscar, Insertar, Vaciar, Eliminar	0	InnoDB	utf8mb4_general_ci	32.0 KB	-
humedad	Examinar, Estructura, Buscar, Insertar, Vaciar, Eliminar	0	InnoDB	utf8mb4_general_ci	48.0 KB	-
sensor	Examinar, Estructura, Buscar, Insertar, Vaciar, Eliminar	3	InnoDB	utf8mb4_general_ci	16.0 KB	-
temperatura	Examinar, Estructura, Buscar, Insertar, Vaciar, Eliminar	0	InnoDB	utf8mb4_general_ci	32.0 KB	-
- Botones y herramientas:** Incluiron Imprimir, Diccionario de datos, Crear nueva tabla, Consola y otras opciones de administración.

Imagen 17: Base de datos

Estructura de tabla

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	idsensor	int(11)			No	Ninguna			Cambiar  Eliminar  Más
2	valor	decimal(5,3)			No	Ninguna			Cambiar  Eliminar  Más
3	fecha	date			No	curdate()			Cambiar  Eliminar  Más
4	hora	time			No	curtime()			Cambiar  Eliminar  Más
5	noMedicion	int(11)			No	Ninguna			Cambiar  Eliminar  Más

↑ Seleccionar todo Para los elementos que están marcados: Examinar Cambiar Eliminar Primaria Único Índice Índice Espacial

Texto completo Agregar a columnas centrales Eliminar de las columnas centrales

Imprimir Planteamiento de la estructura de tabla Hacer seguimiento a la tabla Mover columnas Normalizar

Agregar 1 columna(s) después de noMedicion Continuar

Índices

Acción	Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
Editar  Renombrar  Eliminar	PRIMARY	BTREE	Sí	No	noMedicion	0	A	No	
Editar  Renombrar  Eliminar	idsensor	BTREE	No	No	idsensor	0	A	No	

Crear un índice en 1 columna(s) Continuar

Particiones

⚠ ¡Ninguna partición definida!

Tabla de divisiones

Consola

Imagen 18: Tabla de frecuencia

Estructura de tabla

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	idsensor	int(11)			No	Ninguna			Cambiar  Eliminar  Más
2	valor	float(5,3)			No	Ninguna			Cambiar  Eliminar  Más
3	fecha	date			No	curdate()			Cambiar  Eliminar  Más
4	hora	time			No	curtime()			Cambiar  Eliminar  Más
5	noMedicion	int(11)			No	Ninguna			Cambiar  Eliminar  Más

↑ Seleccionar todo Para los elementos que están marcados: Examinar Cambiar Eliminar Primaria Único Índice Índice Espacial

Texto completo Agregar a columnas centrales Eliminar de las columnas centrales

Imprimir Planteamiento de la estructura de tabla Hacer seguimiento a la tabla Mover columnas Normalizar

Agregar 1 columna(s) después de noMedicion Continuar

Índices

⚠ Los índices idsensor y idsensor\_2 parecen ser iguales y posiblemente se puede eliminar uno.

Acción	Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
Editar  Renombrar  Eliminar	PRIMARY	BTREE	Sí	No	noMedicion	0	A	No	
Editar  Renombrar  Eliminar	idsensor	BTREE	No	No	idsensor	0	A	No	
Editar  Renombrar  Eliminar	idsensor_2	BTREE	No	No	idsensor	0	A	No	

Crear un índice en 1 columna(s) Continuar

Particiones

⚠ ¡Ninguna partición definida!

Consola

Imagen 19: Tabla de humedad

**Table: temperatura**

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	idsensor	int(11)		No	Ninguna				<a href="#">Cambiar</a> <a href="#">Eliminar</a> <a href="#">Más</a>
2	valor	decimal(5,3)		No	Ninguna				<a href="#">Cambiar</a> <a href="#">Eliminar</a> <a href="#">Más</a>
3	fecha	date		No	curdate()				<a href="#">Cambiar</a> <a href="#">Eliminar</a> <a href="#">Más</a>
4	hora	time		No	curtime()				<a href="#">Cambiar</a> <a href="#">Eliminar</a> <a href="#">Más</a>
5	noMedicion	int(11)		No	Ninguna				<a href="#">Cambiar</a> <a href="#">Eliminar</a> <a href="#">Más</a>

**Índices**

Acción	Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
<a href="#">Editar</a> <a href="#">Renombrar</a> <a href="#">Eliminar</a>	PRIMARY	BTREE	Sí	No	noMedicion	0	A	No	
<a href="#">Editar</a> <a href="#">Renombrar</a> <a href="#">Eliminar</a>	idsensor	BTREE	No	No	idsensor	0	A	No	

**Particiones**

!Ninguna partición definida!

**Tabla de divisiones**

**Consola**

Imagen 20: Tabla de temperatura

**Table: sensor**

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	idsensor	int(11)		No	Ninguna				<a href="#">Cambiar</a> <a href="#">Eliminar</a> <a href="#">Más</a>
2	tipo	text	utf8mb4_general_ci	No	Ninguna				<a href="#">Cambiar</a> <a href="#">Eliminar</a> <a href="#">Más</a>

**Índices**

Acción	Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
<a href="#">Editar</a> <a href="#">Renombrar</a> <a href="#">Eliminar</a>	PRIMARY	BTREE	Sí	No	idsensor	3	A	No	

**Particiones**

!Ninguna partición definida!

**Tabla de divisiones**

**Información**

Comentarios de la tabla: Contiene información del sensor

Espacio utilizado	Estadísticas de la fila
16.0 KB	dinámico/a

Imagen 21: Tabla de sensor

Una vez los datos proporcionados por los sensores están en la base de datos, se ejecutó un código en Dart para que pase los datos de la base a la visualización en la aplicación. a continuación se muestran algunas consultas que hicimos.

```

var conn = await MySqlConnection.connect(settings);
var results = await conn.query(
  'SELECT valor FROM humedad ORDER BY fecha DESC, hora DESC LIMIT 1');
var humedad = results.first.fields['valor'];

results = await conn.query(
  'SELECT valor FROM temperatura ORDER BY fecha DESC, hora DESC LIMIT 1');
var temperatura = results.first.fields['valor'];

results = await conn.query(
  'SELECT valor FROM frecuencia_cardiaca ORDER BY fecha DESC, hora DESC LIMIT 1');
var frecuenciaCardiaca = results.first.fields['valor'];

```

Imagen 22: Código en Dart para pasar datos a la aplicación

```

var conn = await MySqlConnection.connect(settings);
var results = await conn.query(
  'SELECT valor, fecha, hora FROM $tableName ORDER BY fecha DESC, hora DESC LIMIT 15');

```

Imagen 23: Código en Dart para pasar datos a la aplicación

### *Aplicación móvil*

La aplicación se desarrolló en Flutter que es un conjunto de herramientas y su lenguaje de programación es dart. Usamos este método porque fue fácil de entender y muy rápido y efectivo a la hora de realizar la aplicación, ya que es orientado a objetos pudimos hacer clases como por ejemplo de las páginas que mostraba y de ahí ir editando y construyendo nuestra aplicación. A continuación se muestra un ejemplo de cómo usamos las clases para crear las páginas.

```

222 //Pantalla del dashboard principal
223 > class DashboardScreen extends StatefulWidget { ...
229
230 > class _DashboardScreenState extends State<DashboardScreen> { ...
372
373 > Widget _buildInfoCard(String title, String value) { ...
398 //Pantalla de Familia
399 > class FamiliaScreen extends StatelessWidget { ...
483
484 //Pantalla de Alertas
485 > class AlertasScreen extends StatelessWidget { ...
543

```

Imagen 24: Clases y Widgets de la aplicación

Como mencionamos anteriormente, los datos de la base se pasan a la aplicación móvil, para la información actual que aparece al inicio de la aplicación se recopilan los datos

más recientes de las tablas de frecuencia cardiaca, humedad y temperatura, aunque para las gráficas de los 3 se recopilan los últimos 15 datos.

Estas son algunas de las librerías que usamos en la programación de la aplicación:

```
import 'package:flutter/material.dart';
import 'package:fl_chart/fl_chart.dart';
import 'sql.dart';
import 'dart:math';
```

Imagen 25: Librerías de dart

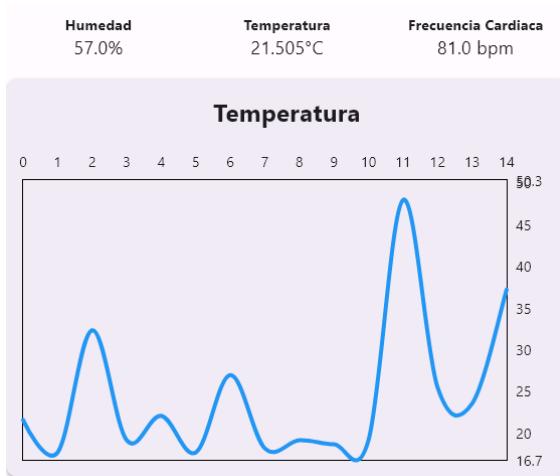


Imagen 26: Datos en la aplicación móvil

Asimismo, la aplicación permite el uso de perfiles o usuarios con la finalidad de poder conectar varios dispositivos a la misma aplicación, esto pensando en los menores o adultos mayores que utilicen la pulsera, de esta manera otro miembro de la familia podrá revisar constantemente los datos recopilados y así poder evitar incidentes futuros.

#### *Video de pruebas*

[https://youtu.be/s\\_mflJHPMZY](https://youtu.be/s_mflJHPMZY)

#### *Resultados*

Se logró nuestro objetivo de crear la pulsera con los 3 sensores, aunque respecto a hardware sentimos que hay muchos aspectos a mejorar para un segundo prototipo como usar sensores de mejor calidad y la caja modelada en 3D realizarla más portátil junto al prototipo. A pesar de esto, nuestra pulsera tiene un diseño con durabilidad y reparabilidad extendiendo la vida útil de la pulsera. No logramos al final tener la aplicación completa que quisieramos mostrar por falta de tiempo pero tenemos un buen avance. Tuvimos varios problemas con los

sensores, especialmente el Max1030 ya que usaba mucha memoria al aplicar oximetría al arduino y sus mediciones de frecuencia cardiaca no eran muy confiables por lo que decidimos cambiarlo por un sensor solo de frecuencia cardiaca. En un futuro esperamos que nuestro dispositivo pueda evolucionar y comercializarse para que la aplicación se pueda usar en todo su potencial leyendo datos de varios usuarios.

#### *Conclusiones grupales*

A pesar de las dificultades iniciales que enfrentamos con la plataforma de Sigfox y el DevKit, hemos logrado culminar un proyecto notable que no solo nos ha permitido desarrollar habilidades en el ámbito del hardware y la creación de bases de datos, sino que también ha reforzado nuestras capacidades preexistentes en programación. La creación de nuestra pulsera inteligente va más allá de un simple ejercicio académico; creemos que esta iniciativa tiene el potencial de generar un impacto social significativo.

Creemos que con nuestra pulsera no solo desarrollamos habilidades sino que también podemos crear un impacto social pensando en las consecuencias en el área de la salud a raíz de la pandemia, sino también en la vida de las personas que por alguna razón ya sea enfermedades crónicas o medidas preventivas requieren un constante monitoreo con la intención de responder a las demandas actuales y anticipar necesidades futuras.

El éxito de este proyecto trasciende la culminación de una materia académica. Más bien, representa el inicio de una aplicación práctica y significativa de nuestros conocimientos en el ámbito de Internet de las cosas (IoT). La experiencia adquirida durante el desarrollo de la pulsera inteligente nos ha proporcionado una comprensión más profunda de cómo aplicar conceptos teóricos a proyectos tangibles y del mundo real. Estamos convencidos de que esta experiencia será invaluable en nuestra trayectoria profesional y nos brindará una ventaja significativa en la industria.

#### *Conclusiones individuales*

Andrea: Este proyecto me permitió recordar los conocimientos previos que poseía acerca de hardware, de la misma manera me obligó a retarme no solo organizándonos como equipo sino también individualmente para que el proyecto pudiera prosperar en el tiempo requerido. A pesar de que se requieren mejoras para un siguiente prototipo, personalmente quedo satisfecha con el resultado final entregado ya que implica que como equipo logramos

nuestros objetivos. Estos aprendizajes obtenidos en la materia me serán útiles en un futuro para mi carrera profesional y hasta vida laboral.

Arturo: En el proyecto aprendí muchas cosas, desde organizar y administrar un proyecto hasta la creación de bases de datos y aplicaciones. Fue un proyecto bastante complejo y con muchas dificultades que fueron surgiendo a lo largo del trabajo. Me gustó que pude tener la oportunidad de trabajar con un devkit y de usar la creatividad para crear una aplicación de forma libre y añadiendo y quitando sensores para que fuera mejorando nuestro proyecto final. Espero poder seguir haciendo este tipo de proyectos en el futuro con más tiempo y dedicando más de mis esfuerzos ya que hay muchas áreas de oportunidad en el trabajo.

Ignacio:Este proyecto fue muy enriquecedor, dado a que pudimos poner en práctica lo aprendido, habilidades individuales y en equipo, aprendimos como se debe gestionar un proyecto que era a corto plazo con áreas escalables y aplicables en la vida laboral, además; conocimos el gran potencial que tienen estos dispositivos, lenguajes de programación y base de datos como campo de desarrollo de competencias no solo dándonos conocimiento en bruto sino también experiencia.

En resumen, IoT es una bloque con un alto valor e impacto para nuestro desarrollo como ingenieros, con una enorme evolución y cambios que requieren de una adaptación continua para no estancarnos a largo plazo.

#### *Referencias:*

*Internet de las cosas: definición y ejemplos | Universitat Carlemany.* (2023, 20 noviembre). UCMA.

<https://www.universitatcarlemany.com/actualidad/blog/internet-de-las-cosas-definicion-y-ejemplos/>