

UNIVERSIDAD POLITÉCNICA DE MADRID

**ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN**



**MÁSTER UNIVERSITARIO EN INGENIERÍA
BIOMÉDICA**

TRABAJO FIN DE MÁSTER

**ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I**

MIRIAM PAVÓN BUENACHE

2016

UNIVERSIDAD POLITÉCNICA DE MADRID

**ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN**

Reunido el tribunal examinador el 18 de Julio de 2016, constituido por

Presidente: Dr D. Enrique J. Gómez Aguilera.

Vocal: Dra Dña. M^a Elena Hernando Pérez.

Secretario: Dr D. Álvaro Gutiérrez Martín

para juzgar el Trabajo Fin de Máster titulado:

**ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA EN LA
PLATAFORMA ARDUINO Y ANDROID PARA OBTENER
CONSTANTES VITALES ORIENTADO A PERSONAS CON
DIABETES TIPO I**

del alumno Dña. Miriam Pavón Buenache.

dirigido por D. Álvaro Gutiérrez Martín.

Acuerdan otorgar la calificación de: _____

Y, para que conste, se extiende firmada por los componentes del tribunal, la presente diligencia

Madrid, a _____ de _____ de _____

El Presidente

El Vocal

El Secretario

Fdo: _____ Fdo: _____ Fdo: _____

UNIVERSIDAD POLITÉCNICA DE MADRID

**ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN**



**MÁSTER UNIVERSITARIO EN INGENIERÍA
BIOMÉDICA**

TRABAJO FIN DE MASTER

**ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I**

MIRIAM PAVÓN BUENACHE

2016

Resumen: En este proyecto se hará un análisis de los distintos componentes y posibilidades de un dispositivo a desarrollar, mediante la plataforma Arduino, que contendrá diferentes sensores y módulos para la obtención de datos de distintas constantes vitales: módulo de comunicación Bluetooth, acelerómetro y giróscopo de tres ejes, sensor de temperatura, sensor de humedad y pulsómetro. Todo ello se integrará en un prototipo de bajo coste orientado a medir y almacenar las variables de los sensores descritos previamente. El prototipo consistirá en una pulsera inteligente. La pulsera implementa y adecúa el estándar HL7 versión 2.7, para la transmisión en tiempo real de las distintas constantes vitales. Además se ha incluido un código QR personal para cada paciente incrustado en el prototipo, con el fin de facilitar la identificación del mismo. Finalmente se desarrollará una aplicación Android para favorecer el manejo del sistema y la obtención de los datos en tiempo real.

Palabras clave: Arduino, Android, monitorización, interoperabilidad, hardware libre, código abierto

Summary: The project aims to develop an analysis of different components and options for a device, controlled through the Arduino platform, which contains different sensors and modules capable of continuous monitoring of body vitals: Bluetooth module, gyroscope and accelerometer, temperature sensor, humidity sensor and Heart rate monitor. These will be integrated in a low cost prototype oriented to measure and store all the data acquired from calibrated sensors. The prototype will consist in a smart band. The data is sendind via HL7 V2.7 messaging scheme. A personal QR code for each patient has been integrated, which allows patient identification. After that, an Android application has been done to control this device and show the results of the measures in real time.

Keywords: Arduino, Android, monitoring, interoperability, open hardware, open source.

*ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.*

**“La ciencia no conoce su deuda a la
imaginación”.**

Ralph Waldo Emerson.

ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.

Índice

1	Introducción	13
1.1	Presentación del trabajo	13
1.2	Antecedentes y definición del problema	14
1.3	Diabetes Mellitus tipo 1	15
1.3.1	Terapias con insulina en pacientes con Diabetes Mellitus tipo 1 . .	16
1.3.2	Monitorización de la glucosa en pacientes con Diabetes Mellitus tipo 1	16
1.3.3	Páncreas Artificial	17
1.4	Estructura del TFM	18
2	Justificación y objetivos	20
2.1	Justificación	20
2.2	Objetivos	20
3	Estado del arte	23
3.1	Dispositivos wearable	23
3.2	Dispositivos wearable en el ámbito de la salud	24
3.2.1	Pulseras inteligentes en el ámbito de la salud	24
3.2.2	Situación actual y futuro de los dispositivos wearable en el ámbito de la salud	27
3.3	Aplicaciones móviles en el ámbito de Diabetes	27
4	Materiales y métodos	29
4.1	Introducción	29
4.2	Análisis y elección de componentes	29
4.2.1	Hardware y componentes disponibles	30
4.2.2	Descripción del software empleado	31
4.3	Prototipo para la adquisición de constantes vitales en Arduino	33
4.4	Desarrollo aplicación Android	34
4.4.1	Modelado del sistema	34
4.4.2	Metodología UML (Lenguaje de Modelado Unificado)	36
4.4.3	Tecnologías	36
5	Resultados	38
5.1	Introducción	38
5.2	Análisis y elección de componentes	38
5.2.1	Selección de constantes vitales	38
5.2.2	Controlador	41
5.2.3	Sensor de temperatura	43
5.2.4	Sensor de acelerometría y giroscopo	55
5.2.5	Pulsioxímetro	64
5.2.6	Sensor de humedad	71

*ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.*

5.2.7	Bluetooth	74
5.3	Elaboración de un prototipo, a partir de los componentes seleccionados, para la adquisición de datos a tiempo real en Arduino	77
5.3.1	Módulo de monitorización del prototipo	77
5.3.2	Módulo de interoperabilidad del prototipo	79
5.3.3	Módulo de almacenamiento del prototipo	80
5.3.4	Módulo de comunicación del prototipo	82
5.3.5	Diseño del prototipo	84
5.4	Desarrollo aplicación Android	87
5.4.1	Introducción	87
5.4.2	Contorno del sistema. Actores	88
5.4.3	Requisitos del sistema	89
5.4.4	Diagramas	90
5.4.5	Arquitectura	101
5.4.6	Interfaz de usuario	104
6	Gestión del proyecto	107
6.1	Planificación del proyecto	107
6.2	Presupuesto	108
7	Conclusiones y líneas futuras	111
7.1	Conclusiones	111
7.2	Líneas futuras	113
8	Referencias	115
9	Apéndice I.	128
9.1	Introducción	128
9.2	Sensor DHT-11	131
9.3	Sensor LM-35	133
9.4	Sensor MPU-6050	135
9.5	Sensor ADXL-345	139
9.6	Sensor APDS-9008	141
9.7	MAX-30100	143
9.8	MATLAB y PLX-DAQ	147
10	Apéndice II.	149
10.1	Introducción	149
10.1.1	Características principales	149
10.1.2	Lista de componentes	150
10.2	Montaje Roboband	151
10.3	Módulo de monitorización	156
10.4	Módulo de comunicación	158
10.5	Módulo de almacenamiento	159

Índice de figuras

1	Componentes del Páncreas Artificial [21].	18
2	Wearable de Thorp y Shannon [30].	23
3	BodyMedia FIT [Ibídem].	25
4	Gobe (HealBe) [Ibídem].	25
5	Withings Pulse [Ibídem].	25
6	StepWatch [Ibídem].	26
7	ActiGraph wGT3X-BT Monitor [Ibídem].	26
8	Empatica E4 [Ibídem].	27
9	Procedimiento empleado para la elección final de los componentes.	30
10	Procedimiento empleado para creación del prototipo final.	33
11	Ciclo de vida en espiral [60].	35
12	Diagrama de bloques convencional del Páncreas Artificial.	39
13	Diagrama de bloques modificado.	40
14	Arduino Nano.	41
15	Arduino Nano.	43
16	Sensor de temperatura DHT11 [90].	44
17	Circuito utilizado para analizar el sensor de temperatura DHT11.	44
18	Análisis de arranque del sensor DHT11.	45
19	Análisis de estabilidad del sensor DHT11.	46
20	Sensor de temperatura LM35 [Ibídem].	48
21	Circuito utilizado para analizar el sensor de temperatura LM35.	49
22	Resultado análisis de arranque del sensor LM35.	50
23	Análisis de estabilidad del sensor LM35.	51
24	Sensor de temperatura MPU6050 [Ibídem].	51
25	Circuito utilizado para analizar el sensor de temperatura incorporado en el MPU6050.	52
26	Análisis de arranque del sensor MPU6050 de temperatura.	53
27	Análisis de estabilidad del sensor MPU6050.	54
28	Sensor MPU6050.	55
29	Circuito utilizado para analizar el sensor de acelerometría y giróscopo del MPU6050.	56
30	Funcionamiento del MPU6050.	56
31	Funcionamiento del MPU6050.	58
32	Análisis de arranque del sensor MPU-6050.	59
33	Análisis de estabilidad del sensor MPU6050.	59
34	Posición uno del análisis de estabilidad del sensor MPU6050.	60
35	Gráfica de posición uno del análisis de estabilidad del sensor MPU6050.	60
36	Posición dos del análisis de estabilidad del sensor MPU-6050.	60
37	Gráfica de posición dos del análisis de estabilidad del sensor MPU6050.	61
38	Sensor ADXL345 [Ibídem].	61
39	Circuito utilizado para analizar el sensor de acelerometría ADXL345.	62

*ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.*

40	Análisis de arranque del sensor ADXL345.	63
41	Configuraciones existentes en la fotoplestimografía. La primera es de trans- misión y la segunda de recepción.	64
42	Señal característica de la presión sanguínea.	64
43	Sensor APDS-9008.	65
44	Circuito utilizado para analizar el sensor APDS-9008.	66
45	Análisis de arranque APDS-9008.	67
46	Sensor MAX-30100.	68
47	Circuito utilizado para analizar el sensor MAX30100.	69
48	Análisis de arranque del sensor MAX30100.	70
49	Circuito utilizado para analizar el sensor de temperatura DHT11.	72
50	Análisis de arranque del sensor DHT11.	73
51	Circuito utilizado para analizar el módulo HC-06.	75
52	Sistema de monitorización del prototipo.	77
53	Circuito correspondiente a la monitorización.	78
54	Muestra de datos.	81
55	Circuito módulo almacenamiento.	82
56	Conexión microcontrolador y dispositivo móvil.	83
57	Código QR con la web optimizada para la descarga.	85
58	Ejemplo de prototipo empleado para el wearable.	86
59	Actores del sistema.	88
60	Caso de uso de Monitorización.	91
61	Caso de uso de Formación.	92
62	Caso de uso de Lista de Tareas.	93
63	Caso de uso de Cálculo de IMC.	94
64	Diagrama de clases de actores.	96
65	Diagrama de clases de Monitorización.	97
66	Diagrama de clases de Mantenimiento.	98
67	Diagrama de clases de Entretenimiento.	99
68	Diagrama de secuencia visualizar material formativo.	100
69	Diagrama de secuencia de lista de tareas.	101
70	Diagrama de la arquitectura de la aplicación móvil.	102
71	Base de datos correspondientes a los pacientes.	102
72	Diagrama del servidor.	103
73	Pantalla de inicio y menú principal.	104
74	Menú de monitorización y medida de humedad.	105
75	Menú de medida de pulso cardíaco y de temperatura.	105
76	Menú de entretenimiento y listado de tareas.	106
77	Menú de formación y de ayuda al usuario.	106
78	Diagrama de Gantt del proyecto.	107
79	Apéndice I. Instalación Arduino parte 1.	129
80	Apéndice I. Instalación Arduino parte 2.	129
81	Apéndice I. Instalación Arduino parte 3.	130

*ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.*

82	Apéndice I. Instalación Arduino parte 4.	130
83	Apéndice I. Circuito utilizado para analizar el sensor de temperatura DHT11.	131
84	Apéndice I. Circuito utilizado para analizar el sensor de temperatura LM35.	133
85	Apéndice I. Circuito utilizado para analizar el sensor MPU-6050.	135
86	Apéndice I. Circuito utilizado para analizar el sensor ADXL-345.	139
87	Apéndice I. Circuito utilizado para analizar el sensor APDS-9008.	141
88	Apéndice I. Circuito utilizado para analizar el sensor MAX-30100.	143
89	Apéndice II. Microcontrolador Arduino Nano [30].	151
90	Apéndice II. Circuito final del montaje.	152
91	Apéndice II. Programa Roboband por serial de Arduino.	152
92	Apéndice II. Foto del dispositivo que permite la monitorización, con sus elementos.	157
93	Apéndice II. Circuito correspondiente a la monitorización.	157
94	Apéndice II. Foto del dispositivo que permite la comunicación inalámbrica.	158
95	Apéndice II. Foto del dispositivo que permite el almacenamiento en una tarjeta micro SD.	159
96	Apéndice II. Circuito almacenamiento SD.	160

*ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.*

Índice de tablas

1	Tabla de características de Arduino UNO [86].	42
2	Tabla de características de consumo de Arduino UNO.	42
3	Tabla de características de consumo baja energía de Arduino UNO.	42
4	Tabla de características de consumo modo sleep de Arduino UNO.	42
5	Tabla de características de Arduino Nano [Íbidem].	43
6	Tabla de características sensor DHT11 temperatura [Íbidem].	44
7	Resultado análisis energético DHT11.	45
8	Resultado análisis estabilidad DHT11.	46
9	Tabla de características LM35 [Íbidem].	48
10	Resultado análisis energético LM35.	49
11	Resultado análisis de estabilidad del sensor LM35.	50
12	Resultado análisis energético MPU6050.	52
13	Resultado análisis de estabilidad del sensor MPU-6050 de temperatura.	53
14	Tabla de características sensor MPU6050 [Íbidem]	55
15	Resultado análisis energético MPU6050.	57
16	Tabla de características sensor ADXL345 [101]	62
17	Resultado análisis energético del sensor ADXL-345.	62
18	Tabla de características del sensor APDS-9008 [Íbidem].	65
19	Resultado análisis energético APDS-9008.	66
20	Resultado análisis estabilidad del sensor APDS-9008.	67
21	Tabla de características del sensor APDS-9008 [Íbidem].	68
22	Resultado análisis energético del sensor MAX-30100.	69
23	Resultado análisis estabilidad del sensor MAX-30100.	70
24	Tabla de características sensor DHT11 humedad [Íbidem].	71
25	Resultado análisis energético del sensor DHT11 humedad.	73
26	Tabla de características del módulo HC-06 [Íbidem].	75
27	Resultado análisis energético Bluetooth HC-06.	76
28	Características módulo Virtualbotix SD Card [117].	81
29	Correspondencia parámetros y micro SD.	81
30	Velocidades comunicación HC-06	84
31	Relación final de pines de Roboband.	151

Glosario

CAD: Computer Aided Design.

DM1: Diabetes Mellitus tipo 1.

DM2: Diabetes Mellitus tipo 2.

DMP: Movimiento Digital.

ETSI: European Telecommunications Standards Institute.

HCE: Historial Clínico Electrónico.

HDR: High-Dynamic-Range imaging.

HL7: Health Level 7.

I2C: Inter-Integrated Circuits.

IMC: Índice de Masa Corporal.

ISR: Servicio de Interrupción.

OMS: Organización Mundial de la salud.

OSI: Open System Interconnection.

PLX-DAQ: Parallax Data Acquisition tool.

PPG: Fotoplestimografía.

QR: Quick Response.

SCL: System Clock.

SDA: System Data.

SPI: Serial Peripheral Interface.

TFM: Trabajo Fin de Máster.

UML: Lenguaje de Modelado Unificado.

UPM: Universidad Politecnica de Madrid.

01/

1 Introducción

En este capítulo se describe la presentación de este Trabajo Fin de Máster (TFM), para posteriormente formular el problema y plantear los antecedentes de esta investigación, describiendo finalmente los diferentes capítulos en los que se compone este documento.

1.1 Presentación del trabajo

En este TFM se presenta el diseño y desarrollo de un prototipo para el registro y almacenamiento de diferentes constantes vitales, con un coste muy reducido, y su posterior comunicación inalámbrica con una aplicación Android, todo ello tutelado por el Prof. Álvaro Gutiérrez y desarrollado en el Laboratorio Robolabo de la Universidad Politécnica de Madrid (UPM). Las constantes vitales que se van a monitorizar son: temperatura, nivel de humedad, ritmo cardíaco y ciclos del sueño.

En este TFM, se aborda el análisis y elección de diferentes sensores y módulos, los cuales en conjunto forman un prototipo final que permite registrar y almacenar datos en una unidad de almacenamiento externa para su posterior estudio. Teniendo todos estos datos en crudo, se facilita la obtención de información para la mejora del páncreas artificial en personas con Diabetes Mellitus tipo 1 (DM1), de tal manera que a los pacientes en estos casos les resultará mucho más cómodo disponer de un sistema de transmisión inalámbrico.

La concordancia de los procesos comunicativos se consigue gracias a la utilización de estándares que, en eSalud, tratan de poner énfasis en la definición de la captura, el almacenamiento y la transmisión de información [1]. Uno de los problemas de la sanidad actual es la falta de interoperabilidad y estandarización de sistemas, por lo tanto, de esta forma, se fomenta la interoperabilidad y el potenciar los servicios de e-Salud que hoy en día existen de forma dispersa. Actualmente, Health Level 7 (HL7) [2] se ha convertido en el conjunto de estándares más reconocido para el intercambio de información médica en Estados Unidos, y su alcance a día de hoy es internacional. Su objetivo es el desarrollo de estándares globales dentro del ámbito sanitario (en los dominios clínico, asistencial, administrativo y logístico), con el fin de lograr una interoperabilidad real entre los distintos sistemas de información que existen en este sector. Hace referencia al

séptimo nivel del modelo de comunicaciones OSI (Open System Interconnection). Los datos monitorizados, se van a transmitir en este estándar.

Por último, mediante una aplicación móvil en el sistema operativo Android, se pretende dar un soporte que facilite la salud personal y la ayuda al paciente. Por un lado, contará con un apartado para monitorizar y supervisar las constantes vitales. Por otro lado, también incluye tutoriales para la formación de pacientes con DM1. Por último, incluirá un código Quick Response (QR) para que en caso de accidente, se tenga a mano información sobre el Historial Clínico Electrónico (HCE) del paciente.

1.2 Antecedentes y definición del problema

La Diabetes Mellitus es un desorden metabólico incurable que representa un gran problema de salud pública [3]. Es la enfermedad más común en los países industrializados afectando a ambos sexos y todas las razas sin respetar ningún límite de edad.

La diabetes (en sus tipos 1 y 2) afectaba en 1980 a 108 millones de personas adultas mientras que en 2014 alcanzaba ya a 422 millones, lo que supone un incremento de 314 millones [4]. Según las previsiones de la Organización Mundial de la salud (OMS), la diabetes será la séptima causa de defunción para 2030.

En las terapias con insulina actuales, el paciente debe realizar controles sanguíneos cada cierto tiempo para medir su glucosa, y en función de esta medida junto con la actividad física y las ingestas realizadas, selecciona la cantidad de insulina a administrar. Las dosis de insulina no causan siempre el mismo efecto, en general la vida diaria causa incertidumbres sobre las terapias que son difíciles de cuantificar. Cientos de investigadores continúan la búsqueda de un sistema que mejore la calidad de vida de los diabéticos. Para evitar la subjetividad y el efecto de las incertidumbres se debería diseñar un método de administración de insulina necesaria a lo largo del día, en el cual la intervención del paciente fuera mínima, ese método formaría parte de un sistema autónomo conocido como "Páncreas Endocrino Artificial"[5]. Tanto en el aspecto clínico como comercial, los dispositivos wearables actualmente ofrecen altas facilidades terapéuticas, de diagnóstico y de monitorización [6]. En la diabetes, el uso de estos dispositivos puede ayudar a mejorar el control automático de los páncreas artificiales, y por tanto, la calidad de vida de los pacientes.

En entornos médicos y hospitalarios es habitual ver pacientes conectados a equipos que monitorizan alguno de sus parámetros fisiológicos. En estos casos resultaría mucho más cómodo disponer de un sistema de transmisión sin cable que permitiese eliminar las conexiones entre el paciente y los instrumentos de medida.

1.3 Diabetes Mellitus tipo 1

La diabetes mellitus (del griego diabainein, “pasar a través” y del latín mellitus, “endulzado con miel”), es conocida desde hace muchos siglos [7]. La primera definición conocida sobre la diabetes aparece en el papiro de Ebers (1530 a.C) donde se describen diferentes remedios para combatir el exceso de orina así como dietas para tratar esta enfermedad. Areteo de Capadocia estaba seguro de que en la diabetes lo que sucedía era “la fundición de la carne hacia la orina”, debido a la notable pérdida de peso que experimentaban los individuos que la padecían. Galeno interpretó que la enfermedad era consecuencia del fallo del riñón y Celso en el Imperio Romano hizo una detallada descripción de la enfermedad, siendo el primero en aconsejar ejercicio físico y dieta [8].

Con el desarrollo progresivo de la ciencia, su tratamiento ha estado mejorándose regularmente. El descubrimiento de insulina data del año 1922 con Frederick Grant Banting y su ayudante Charles Herbert Best, cuando consiguieron detener la enfermedad en un grupo de perros con diabetes, inyectando insulina exógena y reduciendo sus glicemias a los pocos minutos. No sólo habían descubierto la insulina [9], sino que ganaron el Premio Nobel de Medicina al año siguiente. La diabetes tiene una clasificación actual de Tipo I, Tipo II y gestacional.

La diabetes tipo II (DM2) se caracteriza por una resistencia insulínica combinada con una deficiente secreción de insulina. Afecta al 90 % de personas con diabetes. Es la más frecuente en población adulta. En principio, en este tipo de diabetes, los pacientes no necesitan insulina, pero si la evolución natural de la DM2 es hacia un deterioro progresivo de la secreción de insulina por las células, por ello un porcentaje importante de los pacientes requerirá en algún momento la administración de insulina [Ibídem].

La DM1 es un desorden metabólico caracterizado por niveles de glucemia elevados, como consecuencia de una alteración en la secreción de insulina [10]. Se caracteriza por la deficiencia absoluta de insulina debido a la destrucción autoinmune de las células beta del páncreas. Las células beta sintetizan y segregan la insulina y por tanto, una persona con DM1, al carecer de estas células no puede producir insulina. Se desconoce aún la causa de la DM1, y no se puede prevenir con el conocimiento actual. Su presentación puede ser a cualquier edad, siendo su mayor incidencia en menores de quince años, con mayor frecuencia en edad preescolar y especialmente prepuberal [11].

Por este motivo, las personas con DM1 necesitan administración externa de insulina, mediante inyecciones o con una bomba de insulina, para mantener el nivel de glucosa en sangre dentro de los valores adecuados. La DM1 representa alrededor del 10 % de los casos de diabetes mellitus y es donde el control de la concentración de glucosa es más difícil, especialmente en horario nocturno o después de las comidas [12].

La diabetes gestacional empieza o se diagnostica durante el embarazo. Aunque no se conoce la causa exacta, se cree que las hormonas del embarazo pueden bloquear la acción de la insulina. Las mujeres embarazadas con diabetes gestacional suelen tener bebés con un peso superior al habitual y esto puede provocar problemas durante el parto. Este tipo

de diabetes provoca una mayor probabilidad de que el bebé desarrolle DM2 u obesidad. La diabetes gestacional afecta entre el 3 % y el 8 % de las mujeres embarazadas.

1.3.1 Terapias con insulina en pacientes con Diabetes Mellitus tipo 1

El objetivo de todas las formas de tratamiento de la DM1 es [13] mantener el nivel de glucosa en sangre dentro de los límites normales y de esta forma reducir las complicaciones. Para conseguirlo, se intenta reproducir fielmente los perfiles administrados por un páncreas no patológico.

Una terapia de insulina es el plan de inyecciones de insulina que debe seguir a diario la persona con diabetes para mantener el nivel de glucosa en sangre dentro de los límites normales y de esta forma reducir las posibilidades de sufrir alguna complicación [14]. En las personas con DM1 la secreción de insulina por parte del páncreas es nula y, por tanto, la administración exógena de insulina es imprescindible. Uno de los principales problemas es que el sistema regulador de la glucosa es afectado por un gran número de factores como son el peso, el ejercicio físico, y especialmente, la dieta y provocando que el control de los niveles de glucemia sea complicado.

En las terapias vigentes, el paciente mide su nivel de glucosa en sangre, para que en función de este valor, de las ingestas y de la cantidad de ejercicio realizado, se valore la cantidad de insulina que es necesaria inyectarse. Estas razones de imprecisión y subjetividad, es uno de los problemas que se quieren solucionar con el páncreas artificial.

Actualmente, existen dos tipos de terapias [15]:

■ **Terapia convencional mediante inyecciones:** consiste en inyectar por medio de jeringuillas o plumas diferentes tipos de insulina. Para cubrir la secreción basal de insulina se utiliza insulina de acción lenta o intermedia y para cubrir los bolos se necesita insulina de acción rápida o ultra-rápida.

■ **Terapia de infusión Subcutánea Continua de Insulina:** desaparece el concepto de inyección y aparece el de infusión periódica cada 5-15 minutos mediante un catéter. La insulina exclusivamente de acción rápida, es ahora administrada por una bomba o infusor de forma continua durante todo el día, de modo que imita a la insulina basal. Para cubrir los bolos se puede programar la bomba para que inyecte más insulina antes de una comida.

1.3.2 Monitorización de la glucosa en pacientes con Diabetes Mellitus tipo 1

En la actualidad hay diferentes dispositivos portátiles que permiten a los pacientes hacer una medida de manera inmediata para conocer y poder controlar el nivel de glucosa en sangre. A continuación se explican los diferentes tipos de sistemas de monitorización existentes.

■ **Glucómetros:** la gran mayoría de los glucómetros [16] usan métodos electroquímicos para medir la corriente que se produce cuando la glucosa de la sangre reacciona con una tira reactiva y se oxida. A pesar de la precisión y facilidad de las medidas, el gran inconveniente del glucómetro es el elevado número de pinchazos a los que debe someterse el paciente y el consiguiente rechazo que ello provoca. Además, este método proporciona solamente medidas discontinuas que no permiten seguir de manera exacta todas las excursiones reales de la glucemia, lo que dificulta su posterior procesamiento.

■ **Sistemas mínimamente invasivos:** estos sistemas [17], a diferencia de los glucómetros, penetran en la piel a menos profundidad, reduciendo drásticamente el dolor y realizando la medición del nivel de glucemia a partir del fluido intersticial u otro tipo de fluido extravascular. Una de las técnicas empleadas, es la iontoforesis inversa, que consiste en extraer el líquido de la piel aplicando un potencial, lo que origina una corriente eléctrica muy baja sobre esta, causando un movimiento de iones que arrastra las moléculas de glucosa a través de la piel hacia 2 discos de gel donde están situados los electrodos del sistema. En la actualidad no existe ningún sensor de glucosa subcutánea que cumpla las características deseadas para su uso clínico ambulatorio, es decir, mínima invasividad, fiabilidad, precisión, biocompatibilidad, facilidad de uso, larga duración, bajo coste, facilidad de integración tecnológica con otros dispositivos y posibilidad de ser ampliamente comercializado.

■ **Sistemas no invasivos:** las investigaciones en este campo se centran [18] en técnicas ópticas como la espectroscopia de infrarrojo cercano, espectroscopia fotoacústica o cambios en la polarización

1.3.3 Páncreas Artificial

El Páncreas Artificial es un dispositivo [19] que se encarga de hacer mediciones del nivel de glucosa en sangre de manera automática. A partir de un sistema de control en lazo cerrado, calcula la infusión óptima de insulina o glucagón que va a necesitar el paciente con el objetivo de conservar los niveles de glucosa en los parámetros deseables. Otro de los objetivos, y no menos importante, es liberar, tanto los pacientes como a sus supervisores, del estrés que supone el control de la enfermedad de forma continua y a lo largo de toda la vida del paciente, obligándoles a tomar medidas del nivel de glucemia e inyecciones de insulina varias veces al día.

El Páncreas Artificial se compone de tres partes [20]:

■ **Monitor continuo de glucosa:** mide los niveles de glucosa en sangre continuamente dando lecturas cada cinco o diez minutos en los modelos más extendidos.

■ **Bombas de insulina:** es un dispositivo portable que se encarga de suministrar, gracias a su programación y de manera controlada, microbolos de insulina. Este

dispositivo es capaz de administrar insulina de forma continuada durante las 24 horas del día.

■ **Algoritmo de control:** es el encargado de regular la dosis exacta de insulina o glucagón que hay que suministrar para conseguir que el nivel glucémico este en los márgenes adecuados. Es el responsable de prevenir o minimizar los episodios de hipoglucemia e hiperglicemia. Se han desarrollado muchos algoritmos de control para el controlador, pero todavía no existe ninguno que cumpla con todos los requisitos de manera efectiva.



Figura 1: Componentes del Páncreas Artificial [21].

1.4 Estructura del TFM

El siguiente proyecto está organizado en los siguientes capítulos que se muestran a continuación.

Capítulo 1: Introducción. En este capítulo se hace una presentación del trabajo, se describe el problema de esta investigación y se introducen los antecedentes de la misma, haciendo una introducción a la enfermedad de la diabetes, sus consecuencias para la salud si no se trata correctamente. Se detallan las terapias utilizadas para la DM1

Capítulo 2: Justificación y objetivos. En el capítulo segundo se justifica el tema del proyecto y los objetivos del mismo.

Capítulo 3: Estado del arte. Este capítulo comprende el concepto de estado del arte, con el estudio de los dispositivos wearables y las aplicaciones móviles en el campo de la diabetes existentes.

Capítulo 4: Materiales y Métodos. En el capítulo cuarto se describe la metodología seguida para el análisis y elección de los diferentes componentes, como para el prototipo de adquisición de constantes vitales en Arduino, como para la aplicación en el sistema operativo Android.

Capítulo 5: Resultados. El capítulo quinto se centra en el análisis y elección de los componentes en función de las metodologías detalladas en el capítulo tercero. También

*ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.*

se diseñará un prototipo de dispositivo wearable con un módulo de monitorización, interoperabilidad, almacenamiento y comunicación. Por último se mostrarán los resultados correspondientes a la aplicación móvil desarrollada.

Capítulo 6: Gestión del proyecto. El capítulo sexto explica la planificación llevada para el proyecto, así como el presupuesto total.

Capítulo 7: Conclusiones y líneas futuras. En el capítulo séptimo se exponen las conclusiones y se indican posibles trabajos futuros de investigación que podrían realizarse a partir de los resultados obtenidos.

Capítulo 8: Referencias. En el capítulo octavo se citan las fuentes bibliográficas consultadas.

Apéndice I. En este apéndice se muestra el procedimiento para realizar el análisis de los distintos sensores, con los códigos y circuitos empleados.

Apéndice II. El segundo apéndice muestra información extra de cómo se ha realizado el diseño del prototipo, explicando cómo desarrollar los diferentes módulos que lo componen.

02/

2 Justificación y objetivos

2.1 Justificación

La DM1 constituye un importante problema sanitario a nivel mundial debido a su prevalencia (crece un 3 % al año) y a su afectación multisistémica [22] que se asocia frecuentemente a la aparición de numerosas complicaciones tanto vasculares, renales, oculares como neurológicas que son responsables de un gran número de invalideces, incapacidades prematuras y muerte [23]. Se sabe que la DM1 afecta 500.000 niños menores de 15 años y 1 de cada 200 adolescentes. En promedio, 15 de cada 100.000 personas sufren de DM1, afecta entre el 10 % y el 15 % del número total de diabéticos [24]. Por lo tanto, la diabetes mellitus tipo 1 tiene una tasa de crecimiento muy prolífico. Esto genera un alto coste social y económico.

En este TFM se pretende crear un prototipo de bajo coste para monitorizar las constantes vitales del paciente ayudándose con la tecnología móvil para, con los datos obtenidos mejorar el control del páncreas artificial en personas con DM1. En el contexto sanitario, se pueden dar dos situaciones. En una mano, los sistemas de asistencia médica en países desarrollados afrontan reducciones presupuestarias, debido a la crisis financiera, y por otro lado el aumento previsible de personas diagnosticadas con diabetes [25]. Para hacerle cara a este reto ha nacido este proyecto.

Mediante la realización de este trabajo se afianzan conocimientos estudiados durante el máster en los campos de:

- Interoperabilidad en dispositivos médicos.
- Usabilidad.
- Telemedicina.
- Modelado y simulación de biosistemas.

2.2 Objetivos

El principal objetivo es el de realizar un dispositivo wearable de bajo coste para medir y almacenar las constantes vitales de manera remota e interoperativa con el fin de

obtener datos en crudo para su futuro análisis. El proyecto se va a centrar especialmente en las constantes vitales de interés de pacientes con DM1.

Objetivo 1. Análisis y elección de componentes.

El primer objetivo va a consistir en analizar diferentes sensores y módulos con el fin de seleccionar los que mejor se adecuen al proyecto. En base a este objetivo fundamental, se establecen los siguientes objetivos secundarios:

- Selección de los parámetros a medir.
- Selección de diferentes sensores en función a los parámetros a medir.
- Análisis energético, de arranque, de estabilidad y de comunicación de diferentes sensores de temperatura, de acelerometría y giróscopo, pulsioximetría y de humedad.

A partir de todos estos análisis, se evaluarán y se seleccionarán los sensores y módulos que sean más apropiados para este trabajo.

Objetivo 2. Elaboración de un prototipo de bajo coste, a partir de los componentes seleccionados, para la adquisición de datos a tiempo real en Arduino.

El segundo objetivo va a ser el de crear un prototipo de pulsera gracias al uso de una impresora 3D y a la introducción y programación de los diferentes componentes seleccionados. Los cinco alcances que se han marcado han sido:

- Módulo de monitorización: donde se programará el microcontrolador para recoger las variables fisiológicas a partir de los sensores escogidos en la fase del análisis.
- Módulo de almacenamiento: para almacenar todos los datos obtenidos.
- Módulo de interoperabilidad: se implementa y adecúa el estándar HL7 versión 2.7, para la transmisión en tiempo real de las distintas constantes vitales desde un dispositivo de adquisición de baja capacidad (Arduino).
- Módulo de comunicación: que va a permitir establecer una comunicación Bluetooth entre la aplicación móvil y un módulo Bluetooth conectado al microcontrolador, generando y extrayendo la información en el formato y protocolo adecuados.
- Diseño del prototipo: será un estudio con una serie de planos y diagramas que darán una idea de cómo va a ser y cómo se construirá el prototipo de la pulsera wearable.

Objetivo 3. Desarrollo de una aplicación Android.

Como último objetivo, en este proyecto se ha desarrollado una aplicación en el sistema operativo Android, dada su alta popularidad, que permitirá al paciente la monitorización de sus constantes vitales así como de obtener formación. Se desarrollará un código QR personal para cada paciente, de tal manera, que en caso de accidente, se le pueda identificar de manera adecuada, ya que la identificación inadecuada de pacientes es una

*ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.*

causa importante en los problemas de complicaciones asociadas a errores en la asistencia [26]. Otro de los principales objetivos de una plataforma de estas características es la forma de llegar al usuario, por lo que una aplicación Android con una interfaz amigable y de fácil de uso se hacen indispensables en este proyecto.

03/

3 Estado del arte

3.1 Dispositivos wearable

Los dispositivos wearable hacen referencia [27] al conjunto de aparatos y dispositivos electrónicos que se incorporan en alguna parte de nuestro cuerpo interactuando de forma continua con el usuario y con otros dispositivos, con la finalidad de realizar alguna función concreta. Pueden ser gafas, pulseras, relojes, ropa, zapatos y hasta calcetines. A diferencia de los teléfonos móviles y tablets, que son dispositivos complementarios a nuestra vida, los dispositivos wearables se caracterizan porque han sido desarrollados para formar parte del cuerpo o vida humana [28].

El primer dispositivo [29] de este tipo se creó en el año 1960 con el objetivo de hacer trampas en los casinos (ver figura 2). Edward Thorp y Claude Shannon, investigadores por aquel entonces del prestigioso MIT, desarrollaron el primer wearable para contar cartas y aumentar las probabilidades de ganar en la ruleta.



Figura 2: Wearable de Thorp y Shannon [30].

A finales de los años setenta, llegó al mercado el primer wearable comercial, fue un reloj calculadora algebraica HP-01 de Hewlett-Packard [31]. Durante los años ochenta y noventa, Steve Mann, el padrino de la tecnología ciborg, integró una cámara en su ojo con la que veía al mundo proyectado desde su retina. Entre otros aportes, Mann desarrolló el HDR (high-dynamic-range imaging), un procesamiento de imágenes y fotografías

que permiten mejorar el rango de luminancias entre las zonas más claras y oscuras, obteniendo una imagen muy similar a la que obtiene con los ojos.

En la actualidad, los dispositivos wearables se encuentran en una época de rápido desarrollo. En el año 2006, las empresas deportivas empiezan a desarrollar dispositivos wearable para medir monitorear el rendimiento de los atletas, a destacar el sistema desarrollado por Nike y Apple que consistía en un chip implantando en el zapato formado por un podómetro y un transmisor de Bluetooth que registraba información de movimiento, cantidad de pasos y velocidad [32].

El desarrollo de dispositivos wearables se va a convertir en un campo en auge en los próximos años. Además es una tecnología con un gran potencial, especialmente en el campo sanitario ya que permite medir y registrar variables biométricas [33].

3.2 Dispositivos wearable en el ámbito de la salud

La población está envejeciendo, las enfermedades crónicas están aumentando y los costes sanitarios están creciendo de manera insostenible. En el campo de la salud, la tecnología wearable permite al usuario mejorar tanto el control sobre su salud como su calidad de vida [34]. La mayoría de las funciones de estos dispositivos en el ámbito sanitario es la medición de constantes vitales, monitorización ciclos del sueño, avisos de medicación, cálculo de calorías consumidas con el ejercicio, etc.

Las pulseras inteligentes son uno de los dispositivos wearables que permite el registro tanto de los datos a tiempo real, como la monitorización de constantes vitales o de patrones del sueño. Estas pulseras permiten ayudar a abordar aquellas situaciones que pongan en riesgo las funciones vitales del paciente, y a abrir nuevos caminos para la investigación y campos de uso [35]. En la siguiente sección se describen los principales rasgos de las pulseras inteligentes.

3.2.1 Pulseras inteligentes en el ámbito de la salud

Se han analizado, para la realización de este proyecto, las diferentes pulseras inteligentes existentes en el mercado.

■ **BodyMedia FIT:** mide los movimientos del cuerpo, el número de pasos, la respuesta galvánica y la temperatura de la piel. Este sistema también proporciona informes de la actividad a través de una aplicación o página web. Su precio es de cien euros [36].



Figura 3: BodyMedia FIT [Ibídem].

■ **Gobe (HealBe):** esta pulsera está formada por un acelerómetro, un sensor de presión para capturar el ritmo cardíaco y la presión sanguínea, y un sensor de impedancia para medir el nivel de hidratación. Este sistema proporciona información sobre calorías quemadas con el ejercicio, ritmo cardíaco, niveles de estrés, niveles de hidratación y ciclos de sueño. Su precio es de 250 euros [37].



Figura 4: Gobe (HealBe) [Ibídem].

■ **Withings Pulse:** proporciona información sobre los pasos andados incluyendo información sobre las calorías quemadas y la distancia andada. Este dispositivo solo mide el ritmo cardíaco y el nivel de oxígeno en la sangre, aunque estas funciones solo están operativas si se pone el dedo en la pulsera. Tiene una aplicación y página web para ver los datos medidos. Su precio es de 100 euros [38].



Figura 5: Withings Pulse [Ibídem].

■ **StepWatch:** este dispositivo mide el número de pasos, es altamente usado en ámbito sanitario por su alta exactitud, para ello incorpora un acelerómetro. Se emplea para el reconocimiento de estilo de vida en pacientes con enfermedades crónicas. Su precio ronda los 475 euros [39].



Figura 6: StepWatch [Ibídem].

■ **ActiGraph wGT3X-BT Monitor:** permite obtener la aceleración prima, gasto energético, ratios MET, pasos dados, intensidad de la actividad física y posición corporal. Y ofrece mediante un software valoraciones del sueño/despertar y medidas de la actividad en función de los usuarios. Su precio es de 200 euros [40].



Figura 7: ActiGraph wGT3X-BT Monitor [Ibídem].

■ **Empatica E4:** está formado por un acelerómetro de tres ejes, un sensor de plestímetro para calcular el pulso, un sensor electrodermal para monitorizar factores relacionados con el estrés y un sensor para calcular la temperatura de la piel. Calcula con gran precisión el pulso, la conductividad de la piel, la temperatura y el movimiento. Su precio ronda los 1600 euros [41].



Figura 8: Empatica E4 [Ibídem].

3.2.2 Situación actual y futuro de los dispositivos wearable en el ámbito de la salud

Actualmente, los dispositivos wearable son capaces de recopilar una gran cantidad de datos, el siguiente paso es profundizar en el análisis de los datos para de esta manera comprender mejor las enfermedades y crear mejores dispositivos médicos [42]. Otro de los pasos es atraer a los consumidores para usar estos sistemas en su salud y vida diaria, ya que son una promesa para la vigilancia, ya sea en el hospital o a domicilio, puesto que monitorizan la salud constantemente y proporcionan datos a los médicos. Es más, incluso podrían alertar a los pacientes de que vayan al hospital antes y así prevenir ingresos repetidos en el hospital.

Los wearables tienen un alto potencial en el ámbito de la salud, tanto para ofrecer plataformas de telemedicina de mejor calidad y de menor coste para los ciudadanos, como para prevenir y predecir enfermedades.

3.3 Aplicaciones móviles en el ámbito de Diabetes

Desde que han llegado las aplicaciones móviles de salud a los smartphones, el control de una enfermedad ya no es una tarea difícil. La oferta de apps saludables es abundante y se espera que continúe creciendo durante los próximos años. A continuación se muestran las más importantes enfocadas a la diabetes. Se ha extraído información en primer lugar de la página de descarga de las aplicaciones y en segundo lugar se han bajado las aplicaciones, y se han probado en la medida de lo posible para ver qué es lo que pueden y no pueden hacer. Seguidamente se presentarán las aplicaciones estudiadas.

■ **Diabetik:** Muestra los niveles promedio de glucosa en la sangre, las horas de actividad, gramos de la ingesta de alimentos, y las desviaciones en los niveles de glucosa en la sangre. Hay sistemas basados en reglas que recuerdan al usuario una tarea específica, como añadir una entrada para un medicamento o actividad. Es una manera de ayudar a los pacientes a mejorar el cumplimiento y garantizar su control de la diabetes [43].

■ **Social Diabetes:** Permite introducir las comidas que se han ingerido a lo largo del día, y también el nivel de glucosa o la cantidad de insulina que se ha necesitado. Tiene la opción de recibir consejos para evitar hipoglucemias nocturnas o corregir el nivel de insulina. Ofrece Teleasistencia virtual, estadísticas y gráficas para ver la evolución [44].

■ **BGMonitor Diabetes:** Especialmente útil para personas que precisan de múltiples inyecciones diarias, BGMonitor Diabetes funciona introduciendo al principio unos ajustes personales, posteriormente introduciendo los datos de azúcar en sangre y de hidratos de carbono y así el programa establece la dosis necesaria, del mismo modo que una bomba de insulina. Proporciona además información añadida y análisis de los resultados [45].

■ **Glucose Buddy:** permite registrar y controlar el azúcar en sangre, así como los efectos de los medicamentos y los carbohidratos ingeridos. Esta información es exportable a PDF y mediante una plataforma online puede ser enviada al médico [46].

04/

4 Materiales y métodos

4.1 Introducción

Este capítulo tiene la finalidad de exponer la metodología seguida para el proyecto. La técnica empleada se caracteriza por ser exploratoria, descriptiva, explicativa y de intervención [47].

Es exploratoria, puesto que en primer lugar se ha investigado cuáles son las constantes vitales de interés a tener en cuenta en pacientes con DM1. Es descriptiva y explicativa, ya que se analizaron los diferentes componentes existentes en el mercado para conseguir la adquisición de los datos en crudo, y se seleccionaron los más adecuados a partir de rigurosos análisis. Y de intervención porque se ha diseñado una propuesta final formada por una pulsera inteligente y una aplicación móvil.

A continuación se presenta con detalle el método empleado para cada parte.

4.2 Análisis y elección de componentes

Como tarea específica de este trabajo, se hará un estudio de la calidad de diferentes sensores y módulos, con los siguientes alcances:

Alcance 1. Análisis energético: donde se analizará el consumo de los diferentes componentes.

Alcance 2. Análisis de arranque: de esta manera se conocerá el tiempo en el que cada componente consigue ofrecer datos fiables.

Alcance 3. Análisis de estabilidad: donde se analizará la dispersión de la información obtenida.

Alcance 4. Análisis de comunicación: este análisis permitirá obtener información acerca de la velocidad de comunicación y frecuencia de muestreo máxima permitida por cada elemento.

Para efectuar este proyecto se han adquirido ciertos componentes electrónicos necesarios para la realización de los ensayos y su posterior análisis. Teniendo en cuenta la gran

cantidad y variedad de sensores y módulos que existen, es necesario un procedimiento estructurado y una organización del mismo. Es por ello que se va a seguir la siguiente ejecución:

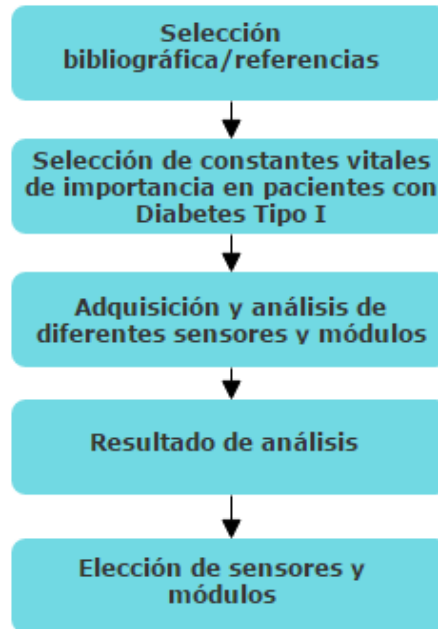


Figura 9: Procedimiento empleado para la elección final de los componentes.

Se ha empleado un método iterativo durante el trabajo, realizando un primer prototipo con el que se realizaron una serie de experimentos, a partir de los cuales se realizaron reformas de cara a un prototipo final.

4.2.1 Hardware y componentes disponibles

En este proyecto se utilizará una pulsera inteligente que contendrá diferentes sensores y módulos. Se hará un análisis de los siguientes componentes:

- **Microcontrolador:** El controlador es el dispositivo encargado de recoger los datos de los diferentes sensores de la pulsera inteligente, para de esta manera procesarlos. Un microcontrolador es un circuito integrado que aúna varios bloques funcionales. Los principales órganos que forman el microcontrolador son la Unidad Central de Procesos, la memoria, y los periféricos [48]. Se trata además, de un dispositivo de bajo costo, versátil, programable, que puede ejecutar funciones y dotar de inteligencia a sistemas más complejos.

- **Acelerómetro y Giróscopo 3 Ejes:** El acelerómetro mide la fuerza de

inercia generada cuando una masa es afectada por un cambio de velocidad. El giróscopo se encarga de medir la velocidad de rotación sobre un eje determinado. Gracias a estos dos sensores, se puede recoger datos para el análisis de los ciclos del sueño.

- **Sensor de temperatura:** Permiten monitorizar la temperatura interna del cuerpo, por sí solo no aportan demasiada información, pero combinados con otros sensores, pueden detectar diferentes cuadros médicos.

- **Pulsioxímetro:** Para conocer la frecuencia cardíaca a tiempo real. Con cada latido del corazón, hay un cambio en el volumen de la sangre dentro de todas las arterias y capilares [49]. Este cambio del volumen influye en la presión de la pared del vaso sanguíneo, causando un cambio del tamaño de su diámetro. Este cambio en la morfología de los vasos sanguíneos afecta a la reflexión, la refracción y la absorción de la luz. Iluminando la piel con la luz, y midiendo la luz reflejada y transmitida, el cambio de volumen de sangre causado por el latido del corazón puede ser obtenido. Para la medida del pulso se ha utilizado este principio.

- **Sensor de humedad:** Permite determinar el comportamiento del cuerpo con respecto a la humedad corporal.

- **Módulo de Bluetooth:** Permite dar la capacidad de transmitir y recibir información con otros dispositivos sin la necesidad de cables, es decir, por un medio inalámbrico. El objetivo del módulo Bluetooth dentro del proyecto es poder establecer una conexión con un Smartphone.

4.2.2 Descripción del software empleado

A continuación se describirá el software empleado para el correcto análisis y posterior elección de los componentes.

4.2.2.1 Fritzing

Fritzing es una aplicación para la edición de esquemas de conexión para proyectos de electrónica. La versión utilizada es la 0.5.4b. Esta aplicación está pensada principalmente para organizar proyectos con Arduino de una forma clara y mediante una interfaz muy sencilla [50]. Cada proyecto que se cree contiene tres vistas principales (protoboard, esquema y PCB) organizadas en pestañas. En este proyecto se va a emplear para realizar los diferentes circuitos.

4.2.2.2 Entorno programación Arduino

El desarrollo y programación del código se ha llevado a cabo mediante el software Arduino IDE 1.6.8. Este software, es fácil de usar y bastante intuitivo. Es de licencia con distribución y uso gratuito [51]. El entorno de desarrollo Arduino lo constituye un editor de texto, donde se plasmará el código; una consola de texto, un área de mensajes y la barra de herramientas con sus menús.

Este software permite realizar la edición de los programas fuente, la compilación, la programación del microcontrolador, la simulación en tiempo real. Arduino tiene la ventaja que no necesita ningún tipo de tarjeta de programación como pasa con otros microcontroladores. Este entorno de desarrollo se conecta al hardware de Arduino para cargar programas y mantener una comunicación serie con el ordenador usando un cable USB [52]. De esta manera, se pueden cargar los programas, sin riesgo de dañar la tarjeta debido a su protección adicional. El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing) [53]. Se pueden usar documentos con diferente extensión, archivos sin extensión, archivos C# (.c), archivos C++ (.cpp) o archivos de cabecera (.h).

Una vez grabado el programa en la memoria del controlador, éste se ejecuta sin necesidad de estar conectado a un ordenador. De esta manera se crea un proyecto hardware/software totalmente autónomo.

4.2.2.3 Entorno programación MatLab

MATLAB (abreviatura de MATrix LABoratory, "laboratorio de matrices") es una herramienta de la empresa Mathworks [54] de software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M). Está disponible para las plataformas Unix, Windows y Mac OS X. Se emplea para la adquisición de los datos de los diferentes sensores, de tal manera que se podrán representar para observar los cambios que sufre el entorno en el que está trabajando el sistema casi de manera instantánea.

Además de la visualización gráfica, se usará MATLAB para generar un archivo donde queden registradas todas las muestras que se han recibido del sistema, para analizar la calidad de todas las muestras y así evaluar que sensor es mejor.

4.2.2.4 Exportación de los datos mediante Parallax-DAQ

Se ha añadido en la programación una serie de líneas de código que permiten la exportación de los valores que llegan por el monitor serie de Arduino a un documento Excel. Esta herramienta se trata concretamente del software Parallax Data Acquisition tool (PLX-DAQ), un macro para Microsoft Excel que guarda los datos de cualquier microcontrolador a tiempo real [55]. Se empleará para la representación y adquisición de los datos de los sensores de acelerometría y giróscopo.

4.2.2.5 PuLseSensor Amped Processing Visualizer

Este software muestra todo lo que Arduino procesa. Se utilizará para representar el pulso del usuario en tiempo real. Este software no calcula el pulso, simplemente lee el puerto-serie y permite visualizar los datos que recibe [56].

4.3 Prototipo para la adquisición de constantes vitales en Arduino

Una vez decidida la tecnología a emplear, se hace el diseño del prototipo final. En este punto se describen los componentes finales empleados para la construcción del prototipo, mostrando sus principales características así como su funcionalidad y su conexionado. Para la metodología se ha hecho la división que se puede ver en la siguiente figura:

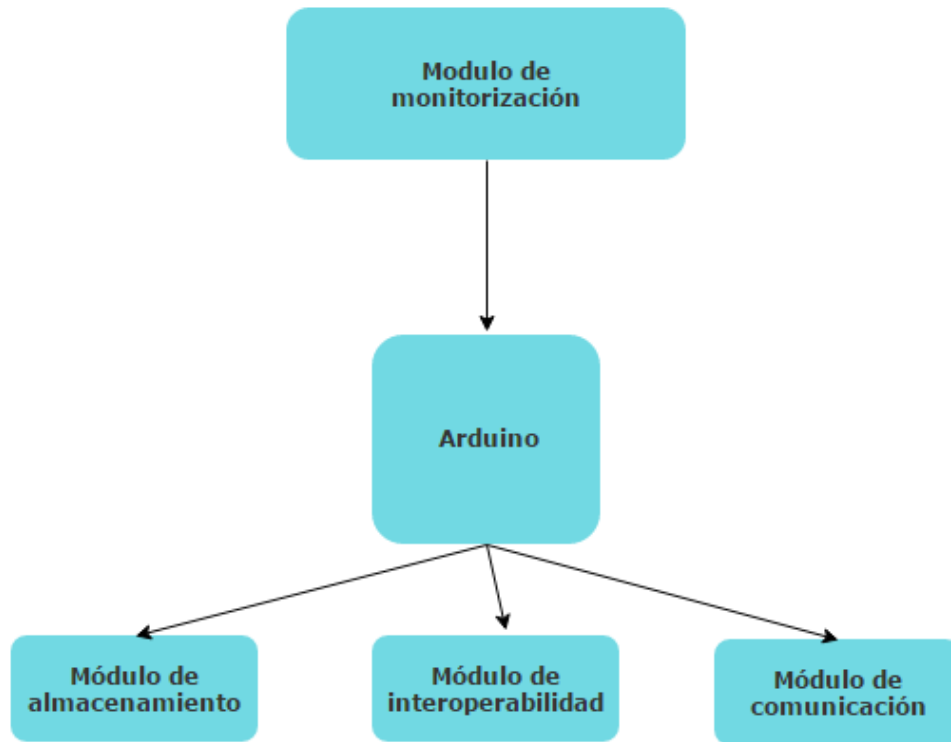


Figura 10: Procedimiento empleado para creación del prototipo final.

- **Módulo de monitorización:** Este módulo está formado por los sensores que se han escogido a partir del análisis. Permitirá la obtención de las constantes vitales.
- **Módulo de almacenamiento:** Este módulo es el encargado de almacenar los datos de la monitorización. Se incorporará un módulo de tarjetas SD-MCC compatible con el microcontrolador. Todos los datos serán almacenados antes de transmitirlos a la aplicación móvil.
- **Módulo de interoperabilidad:** El uso de estándares en la transmisión y almacenamiento de señales biomédicas fomenta la interoperabilidad. Se implementará un sistema que se va a encargar de capturar los diferentes parámetros biomédicos del paciente para procesarlos y transmitirlos en el estándar HL7 v2.7.

- **Módulo de comunicación:** Para establecer una comunicación Bluetooth entre la aplicación móvil y un módulo Bluetooth conectado al microcontrolador Arduino, generando y extrayendo la información en el formato y protocolo adecuados.

- **Diseño del prototipo:** para el diseño del prototipo se usará el programa OpenScad, Este es un programa CAD (Computer Aided Design) y de uso libre que permite la creación de objetos a través de tres comandos: esferas, paralelepípedos y conos. Estos comandos permiten, a través de ciertas instrucciones, obtener cualquier figura física que se pueda plantear. Posteriormente el diseño se imprimirá con una impresora 3D.

4.4 Desarrollo aplicación Android

A continuación se explica la metodología seguida para la realización de la aplicación móvil. Primero se explicará la metodología seguida para el modelado del sistema, y luego se analizará la metodología empleada para el desarrollo del software, hardware, la interfaz de usuario y las tecnologías.

4.4.1 Modelado del sistema

Para el modelado de este proyecto se ha empleado un desarrollo en espiral. Se ha escogido teniendo en cuenta el estándar internacional que regula el método de selección, implementación y monitoreo del ciclo de vida de un software, ISO 12207 [57]. El desarrollo en espiral fue propuesto por primera vez por Barry Bohem [58], se trata de un modelo rápido y eficiente, en el cual se hace un ciclo de prototipo iterativo. En cada iteración, las nuevas expresiones que son obtenidas transformando otras dadas son examinadas para ver si representan progresos hacia el objetivo. En total se han empleado estas fases [59]:

1. Planificación: Durante esta fase se determinan y analizan los objetivos, alternativas y restricciones de la respectiva fase.
2. Evaluación y análisis de riesgos: Se evalúan las distintas alternativas en función de los objetivos y las restricciones. Se determinan los riesgos y se ponen en marcha las actividades que reduzcan dichos riesgos.
3. Desarrollo: Se elige el modelo más apropiado para el sistema de entre diferentes alternativas.
4. Evaluación: Se revisa el proyecto y se continúa. Se planifica la siguiente fase (nueva vuelta de la espiral).

El modelo sería el representado en la siguiente figura.

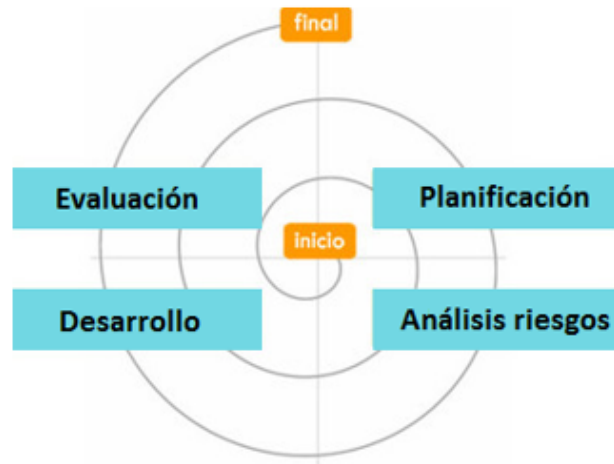


Figura 11: Ciclo de vida en espiral [60].

4.4.1.1 Extracción de requisitos

Para la extracción de requisitos hay que seguir los siguientes pasos [61]:

- **Fuentes:** sirve para obtener documentación inicial sobre el tema a tratar.
- **Entrevistas:** para obtener toda la información posible de la visión que el entrevistado tiene de los requisitos. Sirven para reunir información de personas o grupos. Primero se tiene que hacer una investigación de la situación, es decir, una preparación de tema para tener dominio del mismo, para entender al cliente. Después, se deben identificar a los entrevistados. Y por último, se debe preparar el objetivo de la entrevista y el contenido (establecer la finalidad de la entrevista con el entrevistado).

4.4.1.2 Especificación de requisitos

Una vez realizada la extracción de requisitos, se especifican los actores principales que va a tener el sistema de telemedicina así como las funcionalidades que va a tener. La especificación de los requisitos se basa en una completa descripción de sus funcionalidades. El estándar empleado para la extracción de requisitos es el IEEE 830-1998 [62]. Se van a dividir en dos tipos de requisitos, que se muestran a continuación.

- **Funcionales.** Describen el comportamiento que debe adoptar el sistema. Qué acciones ha de desarrollar, cómo las debe de llevar a cabo y qué resultado se debe dar para cada una de ellas.
- **No Funcionales.** Especifican cómo se debe comportar el sistema en términos de rendimiento, costo, escalabilidad... Son por tanto requisitos que no entran en la funcionalidad, pero son imprescindibles para construir una aplicación efectiva

4.4.2 Metodología UML (Lenguaje de Modelado Unificado)

Es un lenguaje de modelado de sistemas de software [63] que integra y unifica diferentes notaciones y lenguajes formales. Además, facilita la representación del conocimiento acerca de un sistema y la comunicación de dicho conocimiento. La versión 1.4 de UML consta de nueve notaciones gráficas (tipos de diagramas) y un lenguaje formal. En este proyecto se utilizarán los diagramas de casos de usos, los diagramas de clase y los diagramas de secuencia.

4.4.2.1 Diagramas de casos de uso

Los diagramas de casos de uso nos permiten diferenciar los actores que interactúan con nuestra aplicación, las relaciones entre ellos y las acciones que puede realizar cada uno dentro del sistema. Los diagramas se dividen en tres elementos [64]:

- **Actores:** Son los usuarios del sistema. Un actor puede ser una persona, un conjunto de personas, un sistema hardware o un sistema software. Los actores representan un rol, que puede desempeñar alguien que necesita intercambiar información con el sistema.
- **Casos de uso:** Un caso de uso describe una forma concreta de utilizar parte de la funcionalidad de un sistema. La colección de todos los casos de uso describe toda la funcionalidad del sistema.
- **Comunicación entre actores y casos de uso:** Cada actor ejecuta un número específico de casos de uso en la aplicación.

4.4.2.2 Diagramas de clase

Se caracteriza principalmente porque muestra entidades del mundo real, elementos de análisis y diseño o clases de implementación y sus relaciones.

4.4.2.3 Diagramas de secuencia

Un diagrama de secuencia [65] muestra una interacción ordenada según la secuencia temporal de eventos. En particular, muestra los objetos participantes en la interacción y los mensajes que intercambian ordenados según su secuencia en el tiempo. El eje vertical representa el tiempo, y en el eje horizontal se colocan los objetos y actores. En estos diagramas, el tiempo fluye de arriba abajo.

4.4.3 Tecnologías

Se desarrollará una aplicación móvil, con información estática. Como solución tecnológica para satisfacer las necesidades del sistema se ha optado por lo siguiente.

4.4.3.1 Hardware

El hardware necesario para el desarrollo de esta aplicación es un ordenador personal y cualquier móvil Android conectado a Internet es válido para utilizar la aplicación.

4.4.3.2 Software

El software necesario para la aplicación es el siguiente:

- **Android Studio:** Para el desarrollo de la aplicación, se empleará el entorno de desarrollo integrado oficial (IDE) de Android. Esta herramienta destaca por características como su renderización en tiempo real, refactorización específica de Android o su opción de pre visualizado en múltiples configuraciones de pantalla [66].

- **XAMPP:** en el sistema de gestión de bases de datos MySQL con los usuarios de la plataforma móvil, está formado por el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl.

- **Adobe Dreamweaver CC:** para la codificación de los documentos PHP, HTML y CSS. Se empleará para la creación de las webs necesarias para el proyecto.

- **Microsoft Office Visio:** para realizar los diferentes diagramas UML.

En cuanto a los lenguajes de programación, se empleará Java, que es un lenguaje orientado a objetos con una sintaxis similar a C/C++ pero con un modelo de objetos simplificado.

05/

5 Resultados

5.1 Introducción

En este capítulo se ha aplicado la metodología propuesta en el capítulo cuatro, por lo tanto aquí se expondrán y analizarán los distintos resultados. Se dividirá en tres partes:

- Análisis y elección de componentes.
- Elaboración de un prototipo, a partir de los componentes seleccionados, para la adquisición de datos a tiempo real en Arduino.
- Desarrollo de una aplicación Android.

5.2 Análisis y elección de componentes

En esta sección muestran los resultados de todos los elementos de los cuales se han hecho diferentes tipos de análisis. Para ello primero se hace un estudio de las constantes vitales a medir, y en función de esto se seleccionarán diferentes elementos. Después se explicará qué elementos son, cómo funcionan, cuáles son sus principales características y cómo han de ir conectados. A continuación se mostraran los resultados correspondientes al análisis energético, de arranque, de estabilidad y de comunicación de cada uno de ellos. Por último se seleccionaran los más adecuados para el proyecto.

5.2.1 Selección de constantes vitales

La diabetes mellitus tipo 1 es una enfermedad sistémica, crónica, caracterizada principalmente por hiperglicemia y que afecta a la vida de millones de personas [67]. En ella las células beta del páncreas que producen la insulina son destruidas por mecanismos autoinmunes, hecho que lleva a la pérdida total de producción de insulina; es más frecuente en niños, adolescentes y personas jóvenes, aunque puede manifestarse a cualquier edad [68].

*ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.*

La creación de un páncreas artificial totalmente funcional, está cada vez más cerca. Un sistema de lazo cerrado capaz de mantener los niveles de glucosa durante largos periodos de tiempo, provocaría una mejora drástica en calidad de vida de los pacientes con diabetes tipo 1 [69]. El concepto de control de insulina en lazo cerrado, es conocido desde hace tiempo, pero todavía no está completamente optimizado. Los sistemas existentes están formados por: una bomba de insulina, un medidor de glucosa y un algoritmo de control [70].

A continuación se muestra el diagrama de bloques que se corresponde a un Páncreas Artificial convencional.

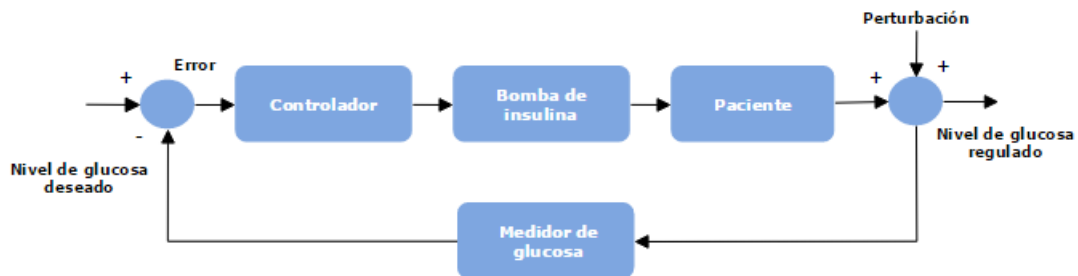


Figura 12: Diagrama de bloques convencional del Páncreas Artificial.

Uno de los mayores problemas de esta arquitectura, es que la mayoría de los pacientes con DM1 tienen una liberación irregular e insuficiente de la hormona glucagón, lo que provoca episodios de hipoglucemia que pueden ser fatales [71]. Además, debido al retraso de la acción de la insulina vía subcutánea, se pueden producir sobredosis de insulina debido a los elevados niveles de glucosa después de las comidas, lo que puede provocar al paciente hipoglucemia. Aunque los tratamientos en entornos clínicos hayan avanzado enormemente en el control de diabetes, la hipoglucemia deja un problema principal no resuelto en el control glucémico.

La monitorización en tiempo real de diferentes constantes vitales puede mejorar el control glucémico. Así que a continuación se mostrará el resultado de las constantes vitales seleccionadas.

*ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.*

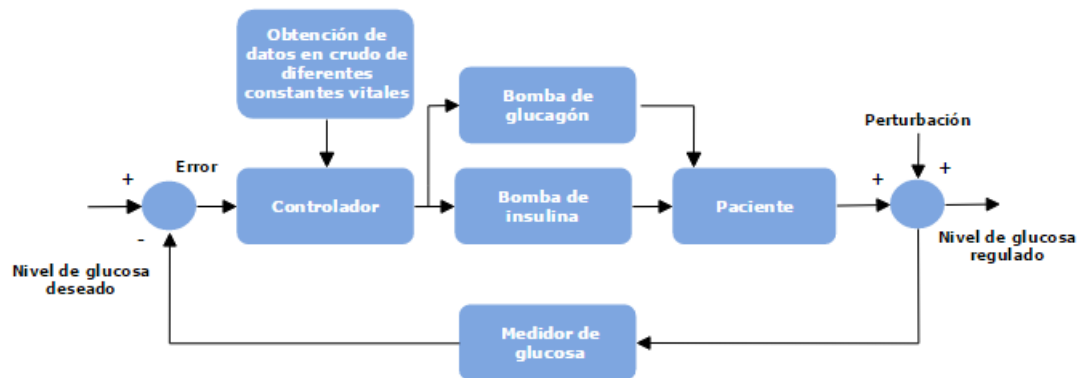


Figura 13: Diagrama de bloques modificado.

Se le denomina pulso al número de latidos, pulsos o contracciones cardiacas por minuto. El corazón late con un ritmo promedio de 75 latidos por minuto en una persona adulta normal, aunque este valor puede variar considerablemente [72]. La monitorización del pulso cardíaco sirve para la detección de casos de hipoglucemia, ya que en un episodio hipoglucémico, la frecuencia del pulso aumenta considerablemente, pudiendo provocar la muerte. Numerosos estudios [73] [74] [75] demuestran, que la muerte por hipoglucemia es cuatro veces mayor en personas con DM1, que en personas no diabéticas. Por lo tanto, una de las constantes vitales seleccionadas en el pulso.

El hipotálamo es el encargado de controlar la temperatura corporal, este recibe dos tipos de señales, una proveniente de los receptores de calor y frio que vienen de los nervios periféricos y otra de la temperatura de la sangre que irriga la región. En una temperatura ambiente neutral, el metabolismo humano genera siempre más calor de lo necesario, esto con la finalidad de mantener la temperatura corporal entre 36.5 y 37.5 °C. En los pacientes con diabetes mellitus tipo I, el aumento o disminución de la temperatura corporal, puede afectar en la absorción de insulina y en los niveles de glucosa. Por ejemplo, un aumento de la temperatura corporal acelera la absorción de la insulina [76]. La temperatura será por tanto otro signo vital de importancia para el diseño del prototipo.

El principal objetivo del sudor es regular la temperatura corporal. Sin embargo, la sudoración excesiva en los diabéticos puede ser signo de hipoglucemia, es decir, bajo nivel de azúcar, que si no recibe atención puede llevar a la pérdida del conocimiento y en algunos cuadros al coma diabético [77]. Por consiguiente, se medirán los niveles de humedad de la piel para saber cuánto sudor se ha generado.

Existe la evidencia experimental de que los desórdenes de sueño y dormir poco afecta al metabolismo de la glucosa y a la resistencia de la insulina [78] [79] [80]. Al mismo tiempo, la alteración del metabolismo de la glucosa puede afectar a la calidad del sueño. La relación entre la diabetes tipo 2 y los desórdenes del sueño ha sido ampliamente estudiada [81]. Pacientes con diabetes tipo 2 presentan una alta prevalencia de apnea

del sueño, lo que contribuye a un mal control glucémico. La relación de la diabetes mellitus tipo 1 y los ciclos del sueño, no ha sido investigada con tanta frecuencia. Es por ello, que en este proyecto, se va a considerar la calidad del sueño como un parámetro importante, de esta manera se obtendrán los datos en crudo sobre los ciclos del sueño, para en un futuro poder evaluar la relación entre la diabetes tipo 1 y la calidad del sueño, lo que facilitaría el control del páncreas artificial.

5.2.2 Controlador

Se ha escogido la plataforma Arduino, ya que se caracteriza por ser un microcontrolador de software y hardware de open-source. El hardware consiste en una placa con un microcontrolador con puertos de entrada y salida [82]. Por otro lado el software consiste en un entorno de desarrollo que implementa el lenguaje de programación Processing/Wiring [83] y el cargador de arranque que es ejecutado en la placa. Arduino se puede utilizar para desarrollar objetos interactivos autónomos o puede ser conectado a diferentes software (Adobe flash, Processing, LabView. . .) Otra de las características de Arduino es que puede adquirir datos a través de sus entradas analógicas y digitales y funcionar como una tarjeta de adquisición.

Está formado por diferentes modelos, como Arduino UNO, Nano, Mega, etc, todos ellos son compatibles entre sí. Como modelos para la realización de este proyecto se han escogido Arduino UNO y Arduino Nano. La elección se ha hecho teniendo en cuenta la búsqueda de una solución rentable, flexible y que ofrezca un buen rendimiento.

5.2.1.1 Arduino UNO

La placa Arduino Uno es un microcontrolador basado en ATmega328, es de 16 (Mhz), está formado por catorce entradas/salidas digitales (de las cuales seis pueden usarse como salidas PWM) y seis entradas analógicas.

Se puede conectar mediante USB al ordenador y dispone de puertos para la comunicación serie [84]. Por lo tanto este microcontrolador cumple con los requisitos necesarios para poder realizar este proyecto. Para el análisis de los sensores se trabajará con este modelo.



Figura 14: Arduino Nano.

El uso de esta plataforma ofrece también la ventaja de la enorme cantidad de documentación disponible en Internet para cualquier proyecto relacionado con Arduino. Ya

ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.

que es una plataforma de código abierto hay una comunidad de desarrolladores que ha generado toda esa información y resulta muy sencillo aprender a programar el micro. En la tabla 1, se pueden ver sus principales especificaciones

Microcontrolador	ATmega328
Voltaje de operación	5 (V)
Voltaje de entrada (Recomendado)	7 – 12 (V)
Voltaje de entrada (Límite)	6 – 20 (V)
Corriente continua por pin IO	40 (mA)
Corriente continua en el pin 3.3V	50 (mA)
Memoria Flash	32 KB
EEPROM	1 KB
Frecuencia máxima de reloj	16 MHz

Tabla 1: Tabla de características de Arduino UNO [86].

Las tablas de consumo se muestran a continuación, la tabla 2 corresponde al modo normal:

Dispositivo	Corriente Obtenida	Voltaje Datasheet	Voltaje Obtenido	Potencia Consumida
Arduino UNO	46.5 (mA)	Max. 5 (V)	5 (V)	232.5 (mW)

Tabla 2: Tabla de características de consumo de Arduino UNO.

La tabla 3 corresponde a baja energía:

Dispositivo	Corriente Obtenida	Voltaje Datasheet	Voltaje Obtenido	Potencia Consumida
Arduino UNO	34.4 (mA)	Max. 5 (V)	5 (V)	172 (mW)

Tabla 3: Tabla de características de consumo baja energía de Arduino UNO.

Y la tabla 4 corresponde al modo sleep:

Dispositivo	Corriente Obtenida	Voltaje Datasheet	Voltaje Obtenido	Potencia Consumida
Arduino UNO	1.5 (mA)	Max. 5 (V)	5 (V)	7.5 (mW)

Tabla 4: Tabla de características de consumo modo sleep de Arduino UNO.

5.2.1.2 Arduino Nano

Este modelo se va a utilizar para introducirlo en la pulsera inteligente. Se ha escogido debido a que a pesar de tener un tamaño mucho menor que Arduino UNO,

*ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.*

dispone de sus mismas características [87]. En conclusión, dado su pequeño tamaño y compatibilidad con el Arduino UNO, será esta versión la que utilice para el prototipo final de la pulsera.



Figura 15: Arduino Nano.

Se alimenta mediante la conexión mini USB de tipo B, a través del pin número 30 (Vin) con una fuente de alimentación no regulada o utilizando una fuente regulada en el pin número 27. La placa tiene un total de 8 entradas analógicas y 14 entradas/salidas digitales (de las cuales 6 de ellas pueden funcionar como PWM). En la tabla 5, se pueden ver sus principales especificaciones [88].

Microcontrolador	ATmega328
Voltaje de operación	5 (V)
Voltaje de entrada (Recomendado)	7 – 12 (V)
Voltaje de entrada (Límite)	6 – 20 (V)
Corriente continua por pin IO	40 (mA)
Corriente continua en el pin 3.3V	50 (mA)
Memoria Flash	32 KB
EEPROM	1 KB
Frecuencia máxima de reloj	16 MHz

Tabla 5: Tabla de características de Arduino Nano [Íbidem].

5.2.3 Sensor de temperatura

Cuando se trata de medir temperatura corporal, se necesitará un determinado grado de precisión, y será o no necesario que el sensor se encuentre a una distancia dada de la piel. Por eso, se describen a continuación un análisis con diferentes sensores de temperaturas, para la elección del sensor más adecuado.

5.2.3.1 Sensor de temperatura DHT11

Este sensor de temperatura de bajo costo, se caracteriza por tener una señal digital calibrada, por lo que no es necesario añadir ningún circuito de tratamiento de señal [89]. De esta manera, se asegura alta fiabilidad y estabilidad a largo plazo. Está formado por un sensor resistivo NTC, el cual permite medir temperaturas entre 0 y 50 (°C).



Figura 16: Sensor de temperatura DHT11 [90].

El protocolo de comunicación es a través de un único hilo, luego, sólo necesita un cable para controlar el sensor y enviar la información. Contiene un chip que transforma medidas analógicas a digitales. Requiere de una librería propia para funcionar. Sus características más importantes se detallan en la siguiente tabla:

Tensión de alimentación	3 - 5 (V)
Rango de temperatura	0 a 50 °C
Precisión temperatura	± 2 °C
Precio	2,93 €

Tabla 6: Tabla de características sensor DHT11 temperatura [Ibídem].

El circuito empleado es el siguiente:

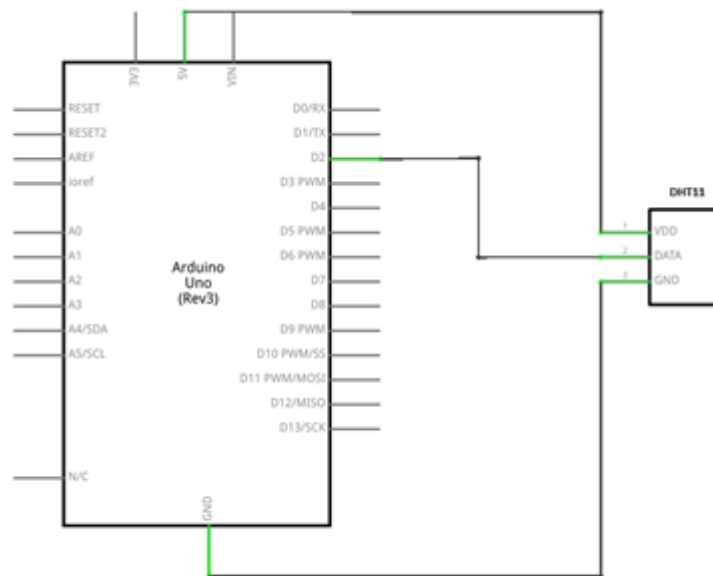


Figura 17: Circuito utilizado para analizar el sensor de temperatura DHT11.

5.2.3.1.1 Análisis energético.

El consumo energético correspondiente al sensor DHT11 se muestra en la siguiente tabla, como se puede observar este sensor consume un total de 12.1 (mW).

Dispositivo	Corriente Obtenida	Voltaje Obtenido	Potencia Consumida
DHT11	2.42 (mA)	5 (V)	12.1 (mW)

Tabla 7: Resultado análisis energético DHT11.

5.2.3.1.2 Análisis arranque.

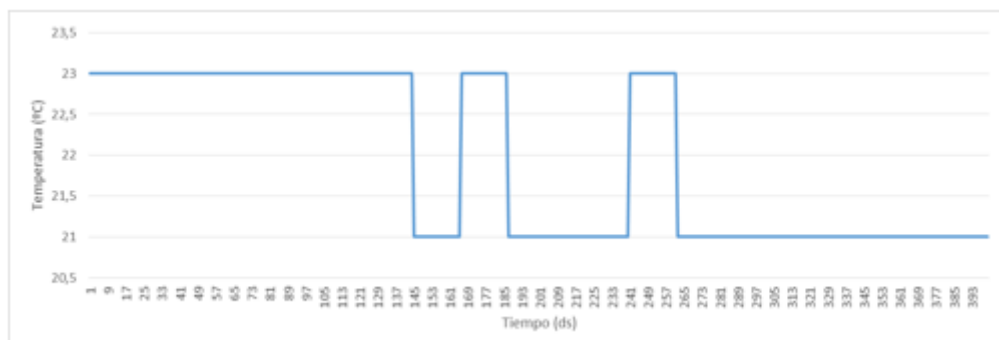


Figura 18: Análisis de arranque del sensor DHT11.

Como se puede observar en la gráfica, este sensor tarda en estabilizarse aproximadamente 25 segundos.

5.2.3.1.3 Análisis estabilidad.

En el siguiente análisis se hace una comparación entre el sensor DHT11 y un sistema real de medida de temperatura. Para el sistema real de temperatura se ha empleado el Termómetro digital TFA 30.1047.05, con una resolución de 0.1 (°C) y con un error de 0.5 °C. A continuación se muestra el % error y se hace una comparación gráfica.

*ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.*

	Manual	DHT11	Error	% Error
Primera prueba	42.3 (°C)	41 (°C)	1.3(°C)	3 %
Segunda prueba	39 (°C)	37 (°C)	2 (°C)	3.1 %
Tercera prueba	36.3 (°C)	35 (°C)	1.3 (°C)	3.5 %
Cuarta prueba	28.6 (°C)	27 (°C)	1.6 (°C)	5.6 %
Quinta prueba	27.5 (°C)	26 (°C)	1.5 (°C)	5.4 %
Sexta prueba	26.4 (°C)	24 (°C)	2.4 (°C)	9.1 %
Séptima prueba	25.6 (°C)	24 (°C)	1.6 (°C)	6.25 %

Tabla 8: Resultado análisis estabilidad DHT11.

La representación gráfica es la siguiente:

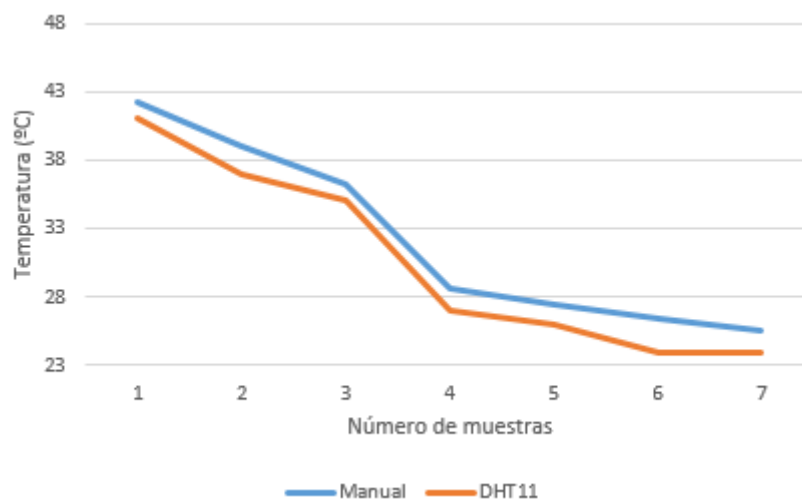


Figura 19: Análisis de estabilidad del sensor DHT11.

5.2.3.1.4 Análisis comunicación.

Su frecuencia de muestreo máximo es de 1 (Hz), es decir, cómo máximo puede tomar una lectura por segundo. Es un sensor, por tanto, muy lento. Aunque puede parecer reducida en comparación a las frecuencias de otros sensores, pero la temperatura y la humedad no son valores que suelen cambiar de forma drástica en un corto espacio de tiempo.

El DHT11 es un sensor que proporciona una salida de datos digital. Entre sus ventajas podemos mencionar el bajo coste y el despliegue de datos digitales. Esto supone una gran ventaja frente a los sensores del tipo análogo, como el LM35 por ejemplo, en los cuales las fluctuaciones en el voltaje alteran la lectura de datos. Entre las desventajas, el DHT11 solo lee enteros, por lo tanto no podemos leer

*ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.*

temperaturas con decimales, por lo que tenemos que pensarlo muy bien a la hora de utilizar este sensor para trabajos en los que se requieran lecturas precisas de temperatura y/o humedad.

5.2.3.2 Sensor de temperatura LM35

Este sensor tiene una salida de 10 (mV) por cada grado centígrado, con un comportamiento muy lineal. El rango de trabajo de este sensor va de -55 °C a +150 (°C), cubriendo de sobra el rango de temperaturas corporales [92].



Figura 20: Sensor de temperatura LM35 [Ibídem].

El LM35 es un sensor con tres pines: +Vs para alimentar el sensor con una alimentación a partir de 4 voltios, la toma de tierra GND, y la salida del sensor que devuelve 10 (mV/°C). Sus características principales se listan en la tabla 9.

Tensión de alimentación	4-10(V)
Tensión de salida	0 + 10 (mV/°C)
Rango de temperatura	-55 a 150 (°C)
Precio	1,88 €

Tabla 9: Tabla de características LM35 [Ibídem].

El voltaje de salida del sensor es interpretado por valores, de manera que 0 (V) es 0 y 5 (V) es 1023. Estos valores hay que convertirlos otra vez a voltios, de tal manera que hay que multiplicar el valor por 5(V) y dividirlo por 1023 que es el entero más grande en un rango de 10 bits. Mirando la hoja de datos del sensor, vemos que por cada °C, la tensión en el pin Vout aumenta 10 (mV), por lo que para dar los 5(V) máximos. La fórmula final sería la siguiente [93]:

$$\text{Temp}(^{\circ}\text{C}) = \frac{\text{Valor del sensor} * 5 * 100}{1023}$$

*ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.*

El circuito empleado para el análisis se muestra en la siguiente figura.

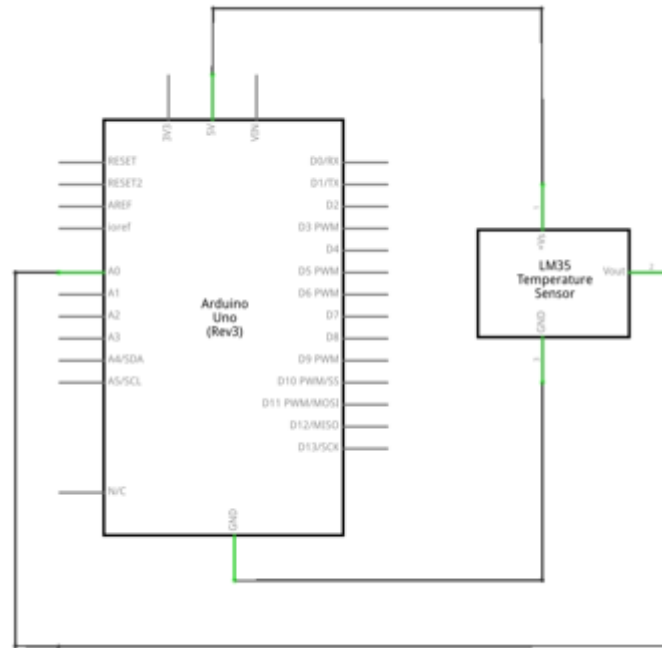


Figura 21: Circuito utilizado para analizar el sensor de temperatura LM35.

5.2.3.2.1 Análisis energético.

El consumo energético correspondiente al sensor LM35 se muestra en la siguiente tabla.

Dispositivo	Corriente Datasheet	Corriente Obtenida	Voltaje Datasheet	Voltaje Obtenido	Potencia Consumida
LM35	Max. 60 (uA)	0.053 (mA)	Max. 36 (V)	5 (V)	0.265 (mW)

Tabla 10: Resultado análisis energético LM35.

5.2.3.2.2 Análisis de arranque.

Como se puede observar en la gráfica, este sensor tarda en estabilizarse aproximadamente 23 segundos.

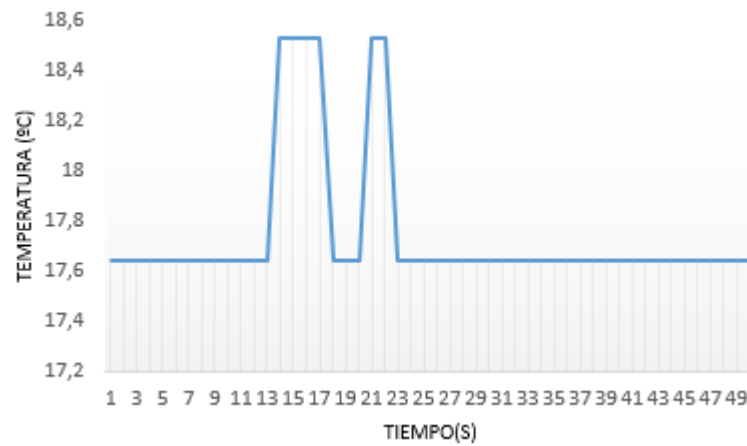


Figura 22: Resultado análisis de arranque del sensor LM35.

5.2.3.2.3 Análisis de estabilidad.

En el siguiente análisis se hace una comparación entre el sensor LM35 y un sistema real de medida de temperatura, se ha empleado el mismo sistema que con el sensor anterior.

	Manual	LM35	Error	% Error
Primera prueba	40.5 (°C)	40.1 (°C)	0.4(°C)	1 %
Segunda prueba	39.5 (°C)	39.6 (°C)	- 0.1 (°C)	0.25 %
Tercera prueba	35.8 (°C)	35.3 (°C)	0.3 (°C)	1.4 %
Cuarta prueba	29.2 (°C)	28.9 (°C)	0.3 (°C)	1.4 %
Quinta prueba	27 (°C)	26.1 (°C)	0.9 (°C)	3.3 %
Sexta prueba	26.4 (°C)	25.2 (°C)	1.2 (°C)	4.5 %
Séptima prueba	25.6 (°C)	24.8 (°C)	0.8 (°C)	3.1 %

Tabla 11: Resultado análisis de estabilidad del sensor LM35.

La representación gráfica es la siguiente:

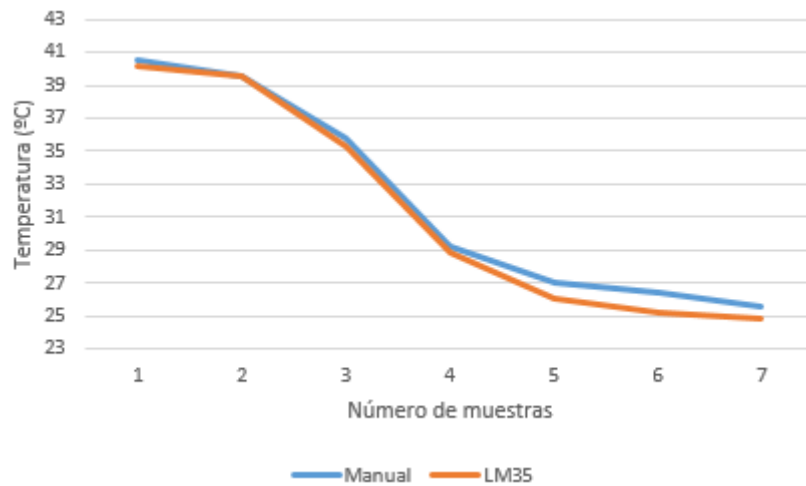


Figura 23: Análisis de estabilidad del sensor LM35.

5.2.3.3 Sensor de temperatura incorporado en MPU6050

El sensor MPU6050 se caracteriza por incluir tres acelerómetros, tres giróscopos y un sensor de temperatura digital con una oscilación de $\pm 1\%$ sobre el rango de temperaturas de operación. La comunicación con todos los registros del dispositivo se lleva a cabo usando Inter-Integrated Circuits I^2C a 400 (KHz). Este módulo funciona con un voltaje de 3.3 (V) [94].



Figura 24: Sensor de temperatura MPU6050 [Íbidem].

La metodología de comunicación de datos del bus I^2C se caracteriza por ser en serie y sincrónica [95]. Una de las señales del bus marca el tiempo, System Clock (SCL) y la otra se utiliza para intercambiar datos, System Data (SDA). Además hay una tercera línea de referencia, tierra (GND).

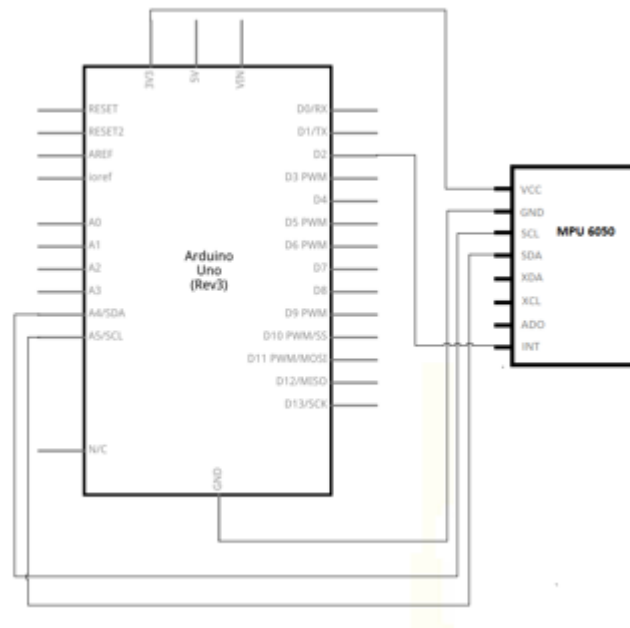


Figura 25: Circuito utilizado para analizar el sensor de temperatura incorporado en el MPU6050.

5.2.3.3.1 Análisis energético.

El consumo energético correspondiente al sensor MPU6050 se muestra en la siguiente tabla, como se puede observar este sensor consume un total de 11.88 (mW).

Dispositivo	Corriente Obtenida	Voltaje Obtenido	Potencia Consumida
MPU6050	3.6 (mA)	3.3 (V)	11.88 (mW)

Tabla 12: Resultado análisis energético MPU6050.

5.2.3.3.2 Análisis arranque.

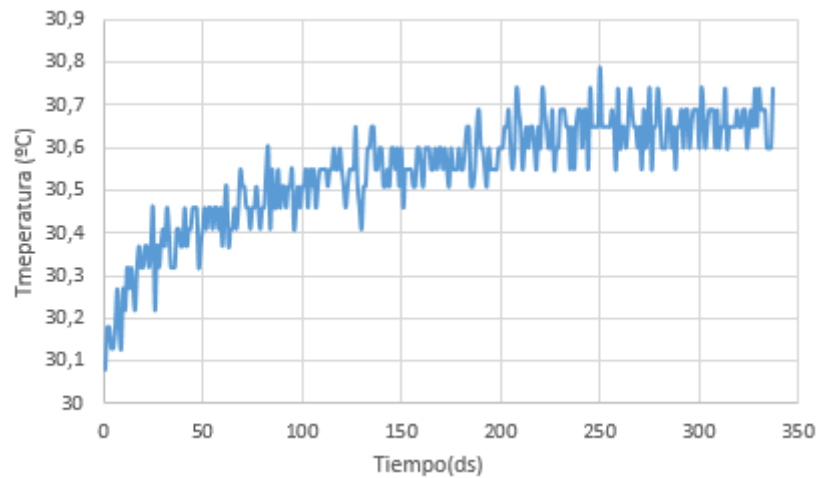


Figura 26: Análisis de arranque del sensor MPU6050 de temperatura.

Como se puede observar en la gráfica, este sensor tarda en estabilizarse aproximadamente 40 segundos. Aunque es muy ruidoso.

5.2.3.3.3 Análisis estabilidad.

En el siguiente análisis se hace una comparación entre el sensor de temperatura MPU6050 y un sistema real de medida de temperatura, para determinar el % error y hacer una comparación gráfica. El sistema real empleado es el mismo que con el sensor DHT11 del punto 5.2.3.1.3.

	Manual	MPU6050	Error	% Error
Primera prueba	40.3 (°C)	41.53 (°C)	-1.23 (°C)	3 %
Segunda prueba	39 (°C)	40.12 (°C)	- 1.12 (°C)	2.8 %
Tercera prueba	35.9 (°C)	37.33 (°C)	-1.43 (°C)	3.9 %
Cuarta prueba	28.4 (°C)	29.97 (°C)	-1.57 (°C)	5.5 %
Quinta prueba	25.6 (°C)	26.58 (°C)	-0.98 (°C)	3.8 %
Sexta prueba	24.7 (°C)	25.92 (°C)	-1.22 (°C)	4.9 %
Séptima prueba	22.6 (°C)	23.87 (°C)	-1.27 (°C)	5.6 %

Tabla 13: Resultado análisis de estabilidad del sensor MPU-6050 de temperatura.

*ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.*

La representación gráfica es la siguiente:

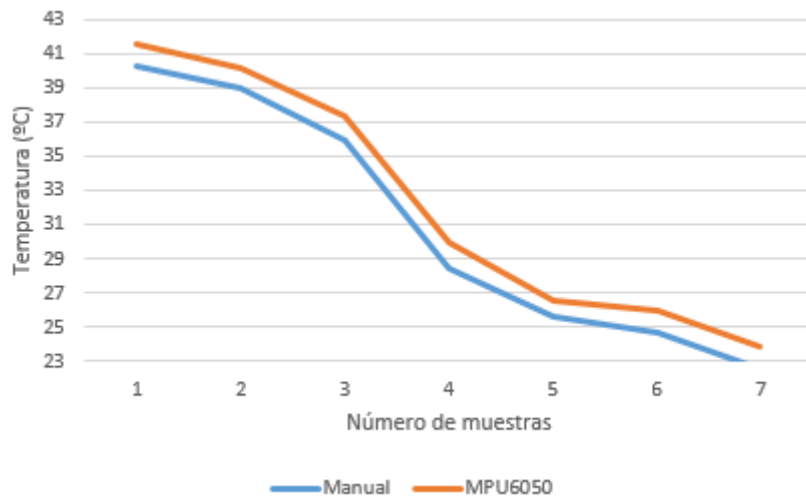


Figura 27: Análisis de estabilidad del sensor MPU6050.

5.2.3.3.1 Análisis comunicación.

La velocidad standard de comunicación es de 100 (Kbit/s). Se puede cambiar al modo de alta velocidad (400Kbit/s)

5.2.2.4 Conclusiones

Finalmente se ha escogido el sensor LM35, ya que opera a menos de 60 (A) y 5 (V). Puede detectar temperaturas entre -55 (°C) a +150 (°C). Ofrece valores muy parecidos a los obtenidos manualmente, por lo tanto es fiable. Se trata de un dispositivo de bajo coste, buena precisión, pequeñas dimensiones, capaz de operar en un amplio rango de temperatura y con un auto-calentamiento casi inexistente. El MPU6050 tiene un buen análisis energético y de arranque, además se encuentra integrado con otros sensores que se van a usar en el proyecto, sin embargo los datos que ofrece no son de la temperatura externa, mide la temperatura interna de su chip, por lo tanto los datos que ofrece no son tan fiables como el LM35, y eso se ha podido comprobar en el análisis de estabilidad. En cuanto al DHT11, ofrece un buen análisis de arranque y es económico. Su principal problema es que tan sólo proporciona medidas enteras, pues tiene una resolución de 1 (°C) y para este proyecto son necesarias temperaturas precisas.

5.2.4 Sensor de acelerometría y giróscopo

En esta sección se van a analizar distintos sensores para la monitorización de los ciclos de sueño. Para ello se usan sensores capaces de medir la aceleración y la velocidad angular. Los sensores encargados de medir estos parámetros son el acelerómetro y el giróscopo. También se explicará detalladamente el funcionamiento del giróscopo y del acelerómetro. Los sensores que se van a comparar son el MPU6050 y el ADXL345.

5.2.4.1 Sensor de acelerometría y giróscopo MPU6050

Este dispositivo es un sensor inercial que combina un giroscopio de 3 ejes y un acelerómetro de 3 ejes junto con un procesador a bordo de Movimiento Digital (DMP), en total es de seis grados de libertad. Tiene una salida digital I²C. Está formado además por un convertor digital-analógico por canal, de tal manera que se consigue asegurar medidas con una alta precisión [96].



Figura 28: Sensor MPU6050.

El giróscopo cuenta con un rango de escala de ± 250 , ± 500 , ± 1.000 y ± 2.000 °/s (dps) y el acelerómetro un rango de ± 2 (g) , ± 4 (g) , ± 8 (g) , y ± 16 (g) [97].

Tensión de alimentación	2.375-3.46(V)
Rango de Fondo de Escala acelerómetro	± 2 (g) , ± 4 (g) , ± 8 (g) , y ± 16 (g)
Rango de Fondo de Escala giróscopo	± 250 , ± 500 , ± 1.000 y ± 2.000 °/s (dps)
Precio	3,91 €

Tabla 14: Tabla de características sensor MPU6050 [Ibídem]

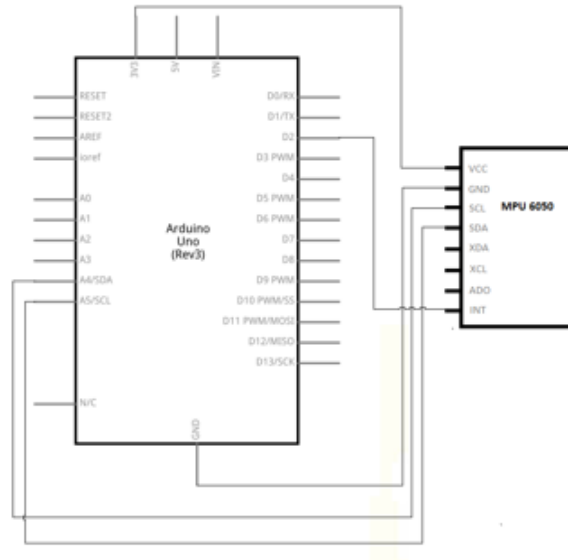


Figura 29: Circuito utilizado para analizar el sensor de acelerometría y giróscopo del MPU6050.

El acelerómetro mide la aceleración, que se expresa las tres dimensiones del espacio: X, Y, Z. Sabiendo que la gravedad de la Tierra tiene una aceleración aproximadamente de $9.8 \text{ (m/s}^2\text{)}$, perpendicular al suelo, el MPU6050 detectará esa aceleración de la gravedad terrestre, pudiendo usar esas lecturas del acelerómetro para saber cuál es el ángulo de inclinación respecto al eje X o eje Y [98].

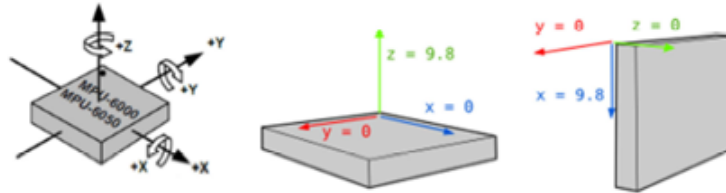


Figura 30: Funcionamiento del MPU6050.

La fórmula utilizada para calcular el ángulo es [99]:

$$\text{Ángulo } Y = \arctan \frac{x}{\sqrt{y^2 + z^2}}$$

$$\text{Ángulo } X = \arctan \frac{y}{\sqrt{x^2 + z^2}}$$

ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.

Un giroscopio mide la velocidad angular. Si se sabe el ángulo inicial del MPU6050, se puede sumar el valor que marca el giroscopio para saber el nuevo ángulo a cada momento. Si el MPU6050 se inicia en 0° y el giroscopio realiza una medida cada segundo, y marca 3 en el eje X, la fórmula para calcularlo sería la siguiente:

$$\text{Ángulo } Y = \text{Ángulo } Y_{n-1} + \text{Giroscopio}Y * \Delta t$$

Siendo Δt el tiempo cada vez que transcurre la fórmula, $\text{Ángulo } Y_{n-1}$ el ángulo Y calculado anteriormente y $\text{Giroscopio}Y$ la lectura del ángulo Y del giroscopio. El cálculo para los ejes X y Z sería igual.

5.2.4.1.1 Análisis energético.

El consumo energético correspondiente al sensor MPU6050 se muestra en la siguiente tabla, como se puede observar este sensor consume un total de 11.88 (mW).

Dispositivo	Corriente Obtenida	Voltaje Obtenido	Potencia Consumida
MPU6050	3.6 (mA)	3.3 (V)	11.88 (mW)

Tabla 15: Resultado análisis energético MPU6050.

5.2.4.1.2 Análisis de arranque.

El sensor necesita de un tiempo de estabilización al principio para dar datos precisos. Este tiempo se establece aproximadamente en torno a unos 16 segundos después de empezar a medir.

Por lo tanto cuando se quieran obtener datos fiables de la orientación, habrá que esperar en torno a 16 segundos para que el sistema empiece a presentar un funcionamiento normal y adecuado. Ya que hasta que no pasen esos segundos iniciales, los ángulos que llegan a la realimentación no son fiables.

*ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.*

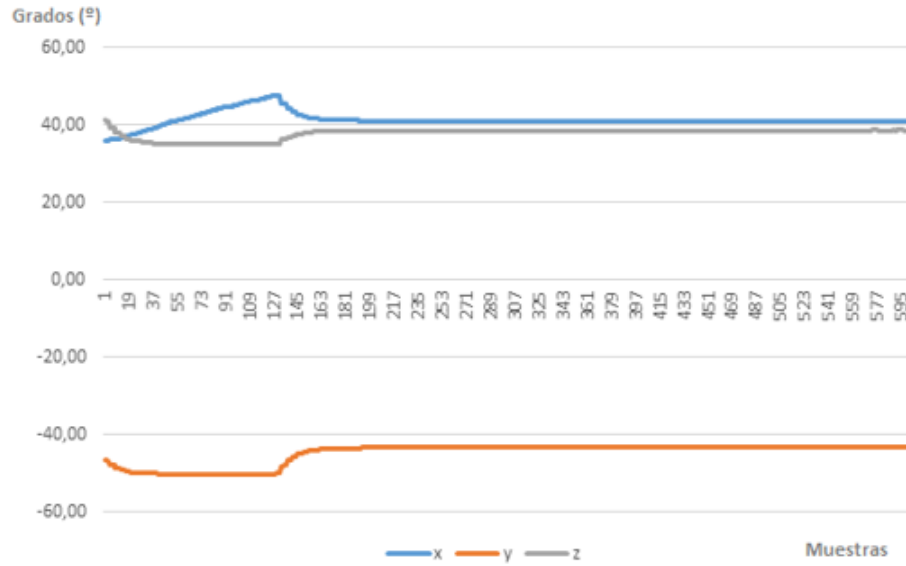


Figura 31: Funcionamiento del MPU6050.

Los datos de acelerometría calculados para el análisis no tienen unidades físicas, son cuentas del ADC, para obtener el valor en (m/s^2) hay que pasar los valores en función de los bits escogidos y del rango. En este caso se ha hecho con 16 bits (luego puede contar hasta 65526) con un rango de ± 2 g. La fórmula para obtener los datos en (m/s^2) sería la siguiente con un dato 700 cuentas del ADC:

$$Acc = \frac{700}{32768} 2(g) * 9.8(m/s^2) = 0.41(m/s^2)$$

Pasados cuatro segundos, los datos son muy fiables. Por lo tanto necesita cuatro segundos para estabilizarse.

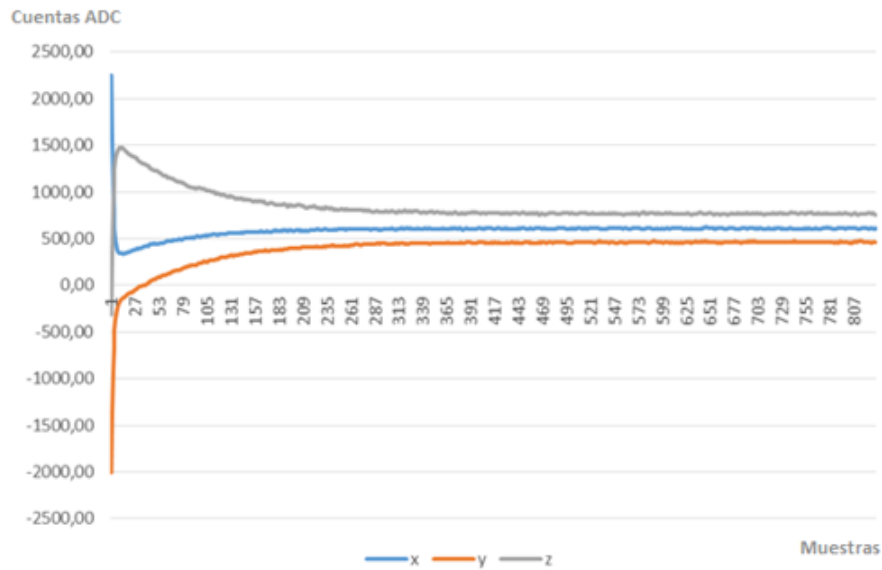


Figura 32: Análisis de arranque del sensor MPU-6050.

5.2.4.1.3 Análisis de estabilidad.

Los ángulos Yaw, Pitch y Roll son cero, en la posición y orientación mostrada en la siguiente figura:

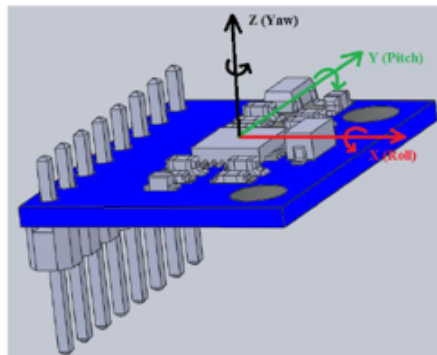


Figura 33: Análisis de estabilidad del sensor MPU6050.

Se han realizado las medidas de los ángulos en dos orientaciones estáticas, las cuales son mostradas en las figuras 34 y 36. Las gráficas correspondientes a cada una de las posiciones se han obtenido con el software Parallax-DAQ y Excel. En estas gráficas se puede apreciar la fiabilidad de los ángulos medidos por el sensor, ya que corresponden a las posiciones de las figuras.

*ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.*

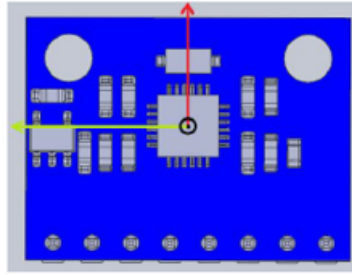


Figura 34: Posición uno del análisis de estabilidad del sensor MPU6050.

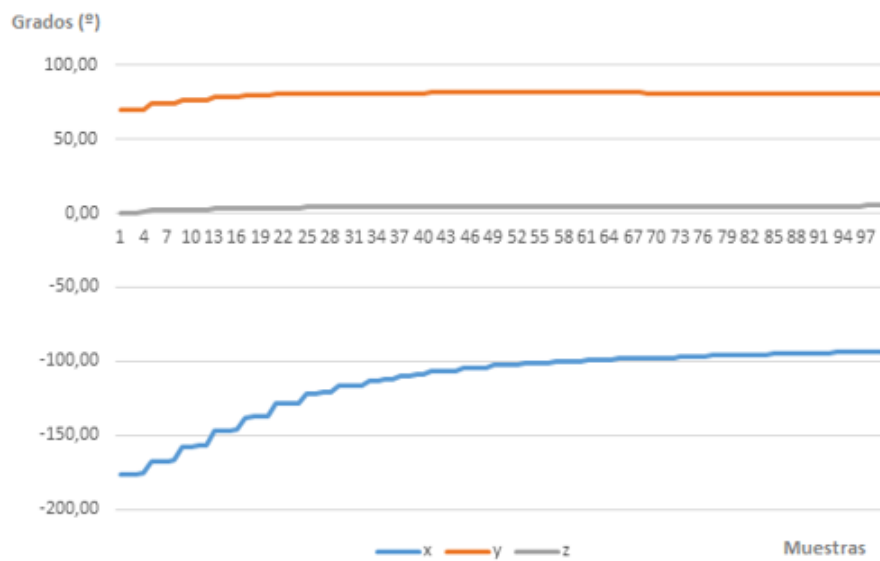


Figura 35: Gráfica de posición uno del análisis de estabilidad del sensor MPU6050.

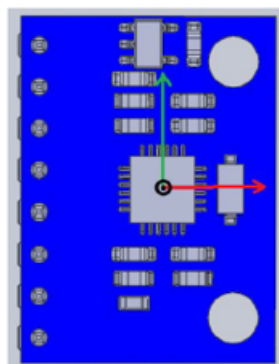


Figura 36: Posición dos del análisis de estabilidad del sensor MPU-6050.

ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.

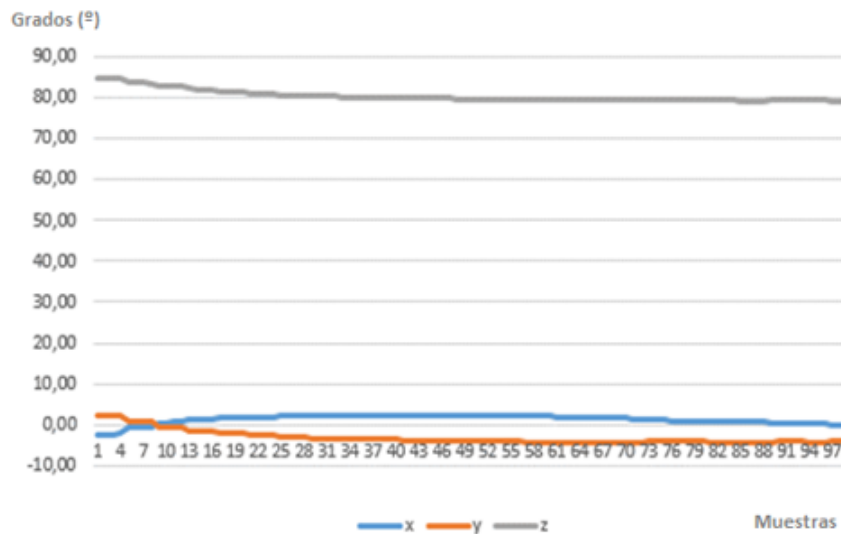


Figura 37: Gráfica de posición dos del análisis de estabilidad del sensor MPU6050.

5.2.4.1.4 Análisis de comunicación.

La velocidad standard de comunicación es de 100Kbit/s. Se puede cambiar al modo de alta velocidad (400Kbit/s) La frecuencia de muestro máxima para el giroscopo es de 1 (KHz) y para el acelerómetro es de 8 (KHz). Cada uno de sus seis grados de libertad, cuentan con registros con una resolución de 16 bits.

5.2.4.2 Sensor de acelerometría GY61 basado en ADXL345

El ADXL345 es un sistema de bajo consumo de potencia, que mide aceleraciones en los tres ejes (x, y, z), se puede programar para realizar mediciones estáticas o dinámicas. La configuración estática es usada para calcular la inclinación de un objeto donde ha sido conectado el sensor. En las mediciones de tipo dinámicas mide las aceleraciones resultantes de movimiento o impactos [100].

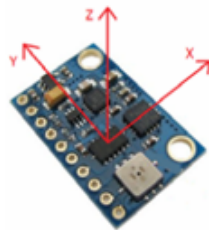


Figura 38: Sensor ADXL345 [Ibídem].

*ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.*

Posee una alta resolución, contando con 13 bits, las mediciones pueden alcanzar hasta ± 16 (g). Sus principales características son las siguientes:

Tensión de alimentación	3-5(V)
Rango de Fondo de Escala acelerómetro	± 2 (g) , ± 4 (g) , ± 8 (g) , y ± 16 (g)
Error de medición	$\pm 1^\circ$ para todos los rangos
Precio	2,32 €

Tabla 16: Tabla de características sensor ADXL345 [101]

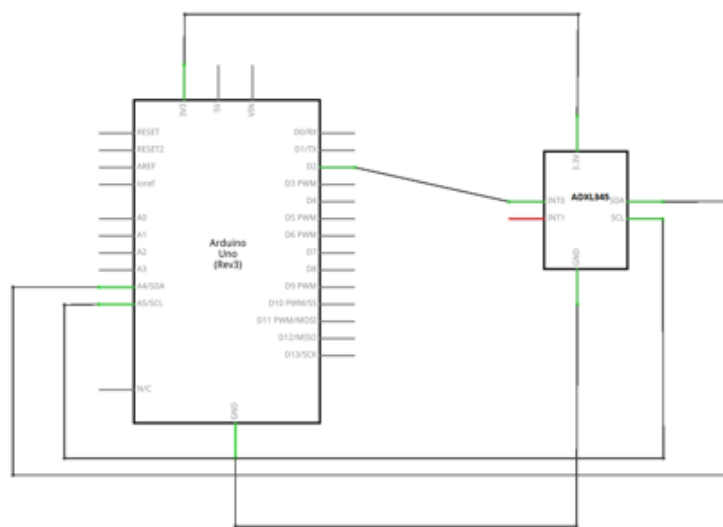


Figura 39: Circuito utilizado para analizar el sensor de acelerometría ADXL345.

5.2.4.2.1 Análisis energético.

El consumo energético correspondiente al sensor se muestra en la siguiente tabla, como se puede observar este sensor consume un total de 0.0759 (mW), y en stand-by 3.3×10^{-4} (mW).

Dispositivo	Corriente Obtenida	Voltaje Obtenido	Potencia Consumida
ADXL345	23 uA	3.3 (V)	0.0759 (mW)
Stand-by	0.1 uA	3.3 (V)	3.3×10^{-4} (mW)

Tabla 17: Resultado análisis energético del sensor ADXL-345.

5.2.4.2.2 Análisis de arranque.

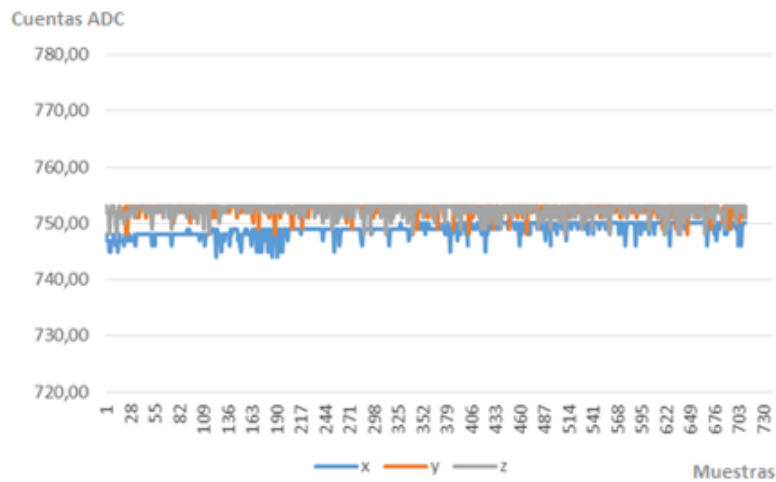


Figura 40: Análisis de arranque del sensor ADXL345.

Como se puede ver en la gráfica, el sensor tarda en estabilizarse menos de un segundo, otorgando datos que varían muy poco entre sí.

5.2.4.2.3 Análisis de comunicación.

El ADXL345 interacciona mediante una comunicación en serie. Tanto por I^2C como por SPI. En ambos casos opera como esclavo. La velocidad de lectura máxima de 5 (MHz) es posible mediante SPI. Con I^2C sólo puede trabajar a una velocidad de 400 (KHz). Y el dispositivo no puede transmitir datos a más de 400 (Hz). Por lo tanto, esta limitación se solventa usando SPI cuya salida de datos se puede realizar a una frecuencia máxima de 1600 Hz o 3200 Hz. La frecuencia de muestreo máxima es de 3,2 (KHz) usando SPI y de 400 (Hz) usando I2C.

5.2.4.3 Conclusiones

Finalmente se optó por el GY-521 (MPU6050) ya que lleva incorporado el sensor de giróscopo también, además de un procesador propio que puede descargarse al procesador principal en caso de ser necesario. En cuanto a los análisis son los dos sensores de calidad muy parecida, aunque en el energético, consume mucho menos el ADXL345.

5.2.5 Pulsioxímetro

A continuación se van a analizar dos sensores para calcular el pulso cardíaco, estos son el APDS-9008 y el MAX30100. Ambos se basan en el efecto de la fotoplestimografía. La fotoplestimografía es una técnica no invasiva que sirve para medir el pulso cardíaco a partir de la habilidad de la luz para reflejarse y penetrar en los tejidos humanos. Consiste en la emisión de luz infrarroja desde un diodo emisor y un fotodetector adyacente que recibe la luz infrarroja reflejada. A medida que aumenta el flujo de sangre cutáneo aumenta la cantidad de luz reflejada [102].

Existen dos configuraciones posibles para la fotoplestimografía (PPG) [103]: reflexión o transmisión. Los sensores por reflexión se pueden colocar sobre cualquier superficie vascular pulsátil. Los sensores por transmisión se tienen que colocar de tal manera que se encuentren enfrentados y sólo se pueden ubicar en determinadas zonas como el lóbulo de la oreja o el dedo. Los sensores empleados para el análisis se basan en la configuración de reflexión.

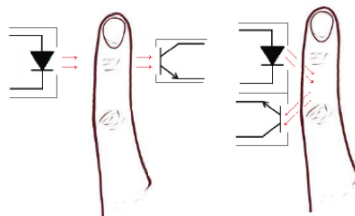


Figura 41: Configuraciones existentes en la fotoplestimografía. La primera es de transmisión y la segunda de recepción.

La señal PPG, no es la señal eléctrica del corazón como se tendría en un electrocardiograma, sino que es una fluctuación en el voltaje analógico, y tiene una forma de onda predecible como se muestra en la figura 42.



Figura 42: Señal característica de la presión sanguínea.

Como se puede observar, es una señal con dos picos en cada periodo, el pico mayor corresponde con la presión sistólica (PS) y el segundo pico, cuyo valor es el mínimo de la curva, representa la presión diastólica (PD). Estos valores dan información sobre el ritmo cardíaco, para ello, hay que encontrar los momentos sucesivos de cada latido y calcular el intervalo de tiempo que ha habido entre ellos llamado IBI (Inter-Beat Intervale), haciendo uso de la forma predecible y del patrón de onda del PPG.

Para calcular el IBI del ritmo cardíaco se emplea la siguiente fórmula [104]:

$$IBI = PD + 0,5(PS - PD)$$

Para calcular los BPM, se realiza una media de los 10 últimos valores de los intervalos entre latidos, para hacer un promediado, y este valor se lo dividiremos a 60000 para obtener los BPM o pulsos que se tienen en un minuto. La fórmula queda así [Ibídem]:

$$BPM = \frac{60000}{IBI}$$

Si el promedio de los últimos 10 latidos es de 900ms, el resultado sería el siguiente [Ibídem]:

$$BPM = \frac{60000}{900} = 66,67 \text{ pulsaciones por minuto}$$

5.2.5.1 Pulsioxímetro con sensor fotoeléctrico APDS-9008

En este sensor, el diodo led emitirá una señal que será recibida por el sensor fotoeléctrico (APDS-9008). Posteriormente, la señal de salida del sensor es acondicionada por un amplificador operacional MCP6001, de tal manera que se consigue reducir el ruido de la señal [105].

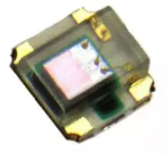


Figura 43: Sensor APDS-9008.

El principio de funcionamiento consiste en que la luz absorbida por la sangre, permite obtener el pulso cardíaco. Sus características principales se listan en la tabla 11.

Tensión de alimentación	1.6-5.5(V)
Longitud de onda	565 (nm)
Precio	2.70 €

Tabla 18: Tabla de características del sensor APDS-9008 [Ibídem].

ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA EN LA PLATAFORMA ARDUINO Y ANDROID PARA OBTENER CONSTANTES VITALES ORIENTADO A PERSONAS CON DIABETES TIPO I.

El sensor contiene un emisor infrarrojo y un receptor montado a un lado. Debe estar presionado contra la piel. Cuando el corazón bombea, la presión arterial se eleva considerablemente y lo mismo ocurre con la cantidad de luz infrarroja procedente del emisor que se refleja en el detector. Parte de la luz infrarroja atraviesa el tejido, pero con cada pulsación el paso de la sangre reduce ligeramente la cantidad de luz que es capaz de traspasarlo. Esta variación se detecta con otro sensor, un fotodiodo cuyas propiedades varían según la luz, generando una corriente eléctrica dependiendo de la cantidad de luz que recibe. El detector deja pasar más corriente cuando recibe más luz, que a su vez provoca una caída de tensión para entrar al circuito amplificador [106]. El circuito empleado es el siguiente:

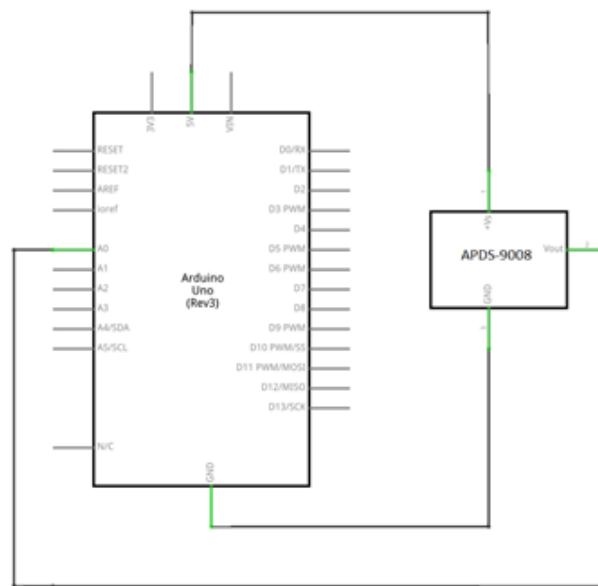


Figura 44: Circuito utilizado para analizar el sensor APDS-9008.

5.2.5.1.1 Análisis energético.

El consumo energético correspondiente al sensor APDS-9008 se muestra en la siguiente tabla, como se puede observar este sensor consume un total de 20 (mW).

Dispositivo	Corriente Obtenida	Voltaje Obtenido	Potencia Consumida
ADPS-9008	4 (mA)	5 (V)	20 (mW)

Tabla 19: Resultado análisis energético APDS-9008.

5.2.5.1.2 Análisis de arranque.

*ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.*

El ritmo cardiaco se detecta directamente conectando una de las entradas analógicas de Arduino. En la parte izquierda del visualizador se puede ver el latido del corazón. Arriba a la derecha se ve en el visualizador que el sujeto tiene alrededor de 80 latidos por minuto.



Figura 45: Análisis de arranque APDS-9008.

5.2.5.1.3 Análisis de estabilidad.

Para el análisis de estabilidad se emplea el servicio de interrupción (ISR). Por lo tanto las muestras se analizarán cada 2 (ms). Se ha empleado un visualizador creado por los desarrolladores para analizar los datos.

Se ha hecho la comprobación midiendo el pulso manualmente en la muñeca. Los resultados son los siguientes:

	Manual	Pulse Sensor
Primera prueba	76 BPM	79 BPM
Segunda prueba	77 BPM	80 BPM

Tabla 20: Resultado análisis estabilidad del sensor APDS-9008.

5.2.5.1.4 Análisis de comunicación.

Es un sensor plug-and-play. Proporciona una frecuencia de muestro para la medición del pulso adecuada. Ya que con 500 Hz es suficiente. Frecuencias de muestreo más altas, provocarían un sobremuestreo. Por lo tanto, su frecuencia de muestreo es de 2 ms, marcada por una interrupción que permite medir la sístole

y la diástole. Los datos recolectados por el sensor son enviados a un convertidor analógico digital de 10 bits.

5.2.5.2 Pulsioxímetro MAX30100

Es un biosensor óptico totalmente integrado. Se caracteriza porque combina dos LEDS y un fotodetector. Tiene un ruido analógico bajo y usa comunicación I^2C [107].



Figura 46: Sensor MAX-30100.

El chip que lleva es uno de los más pequeños del mercado, con unas medidas de 6mm x 3mm.

Tensión de alimentación	1.8-3.3(V)
Longitud de onda LED rojo	880 nm
Longitud de onda LED Infrared	660 nm
Precio	5,97 €

Tabla 21: Tabla de características del sensor APDS-9008 [Ibídem].

ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA EN LA PLATAFORMA ARDUINO Y ANDROID PARA OBTENER CONSTANTES VITALES ORIENTADO A PERSONAS CON DIABETES TIPO I.

El circuito empleado es el siguiente:

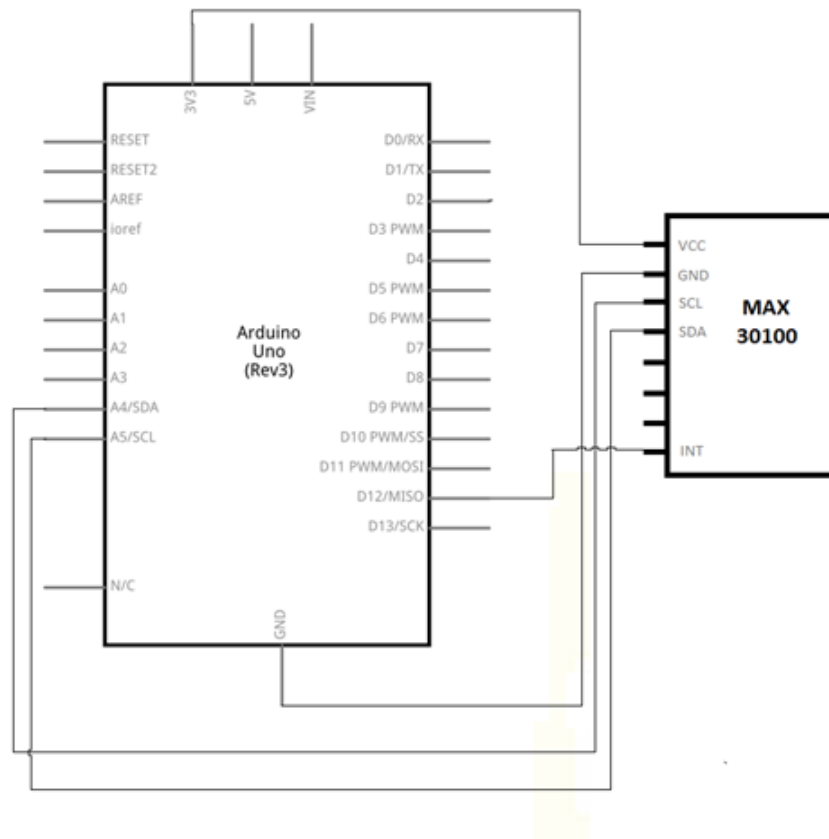


Figura 47: Circuito utilizado para analizar el sensor MAX30100.

5.2.5.2.1 Análisis energético.

El consumo energético correspondiente al sensor MAX30100 se muestra en la siguiente tabla, como se puede observar este sensor consume un total de 66 (mW), y en stand-by 2,31 (uW).

Dispositivo	Corriente Obtenida	Voltaje Obtenido	Potencia Consumida
MAX30100	20 (mA)	3.3 (V)	66 (mW)
MAX30100 stand-by	0.7(uA)	3.3 (V)	2.31 (uW)

Tabla 22: Resultado análisis energético del sensor MAX-30100.

5.2.5.2.2 Análisis de arranque.

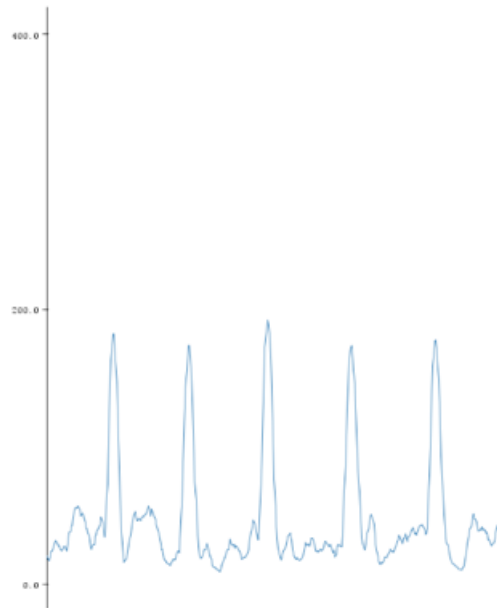


Figura 48: Análisis de arranque del sensor MAX30100.

5.2.5.2.3 Análisis de estabilidad.

Para el análisis de estabilidad se empleó el servicio de interrupción (ISR). Por lo tanto, al igual que en el anterior sensor, las muestras se analizarán cada 2 ms.

Se ha hecho la comprobación midiendo el pulso manualmente en la muñeca. Los resultados son los siguientes:

	Manual	MAX30100
Primera prueba	76 BPM	77 BPM
Segunda prueba	77 BPM	78 BPM

Tabla 23: Resultado análisis estabilidad del sensor MAX-30100.

5.2.5.2.4 Análisis de comunicación.

Su comunicación es por I^2C . Proporciona una frecuencia de muestreo para la medición del pulso que se puede programar desde 50 (Hz) a 1 (Khz). El número máximo de muestras por segundo que se ha podido obtener sin problemas es 50, es decir una muestra cada 20 (ms).

Los datos recolectados por el sensor son enviados a un convertidor analógico digital

el cual puede ser de hasta 16 bits, con el cual se conseguiría una longitud de pulso de 1.6 (ms).

5.2.5.3 Conclusiones

Se ha escogido finalmente el sensor APDS-9008, debido a su bajo coste, dimensiones y fiabilidad de los datos. Aunque el sensor MAX30100 es el que se debe usar en un futuro cuando se miniaturice el dispositivo, ya que ofrece mejores análisis.

5.2.6 Sensor de humedad

El sensor DHT11 es de bajo costo y se caracteriza por tener una señal digital calibrada, por lo que no es necesario añadir ningún circuito de tratamiento de señal. De esta manera, se asegura alta fiabilidad y estabilidad a largo plazo [108].

El protocolo de comunicación es a través de un único hilo, luego, sólo necesita un cable para controlar el sensor y enviar la información. Sus características más importantes se detallan en la siguiente tabla:

Tensión de alimentación	3 - 5 (V)
Rango de humedad	20 a 80
Precisión	Humedad
±5	2,93 €

Tabla 24: Tabla de características sensor DHT11 humedad [Ibídem].

*ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.*

El circuito empleado es el siguiente.

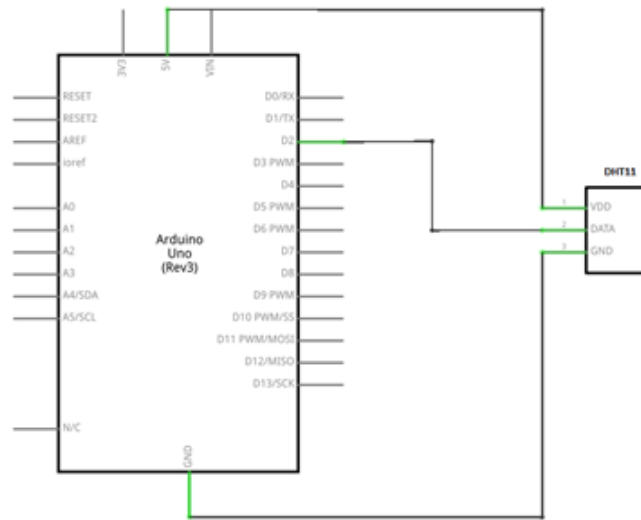


Figura 49: Circuito utilizado para analizar el sensor de temperatura DHT11.

5.2.6.1.1 Análisis energético.

El consumo energético correspondiente al sensor DHT11 se muestra en la siguiente tabla, como se puede observar este sensor consume un total de 12.1 (mW).

Dispositivo	Corriente Obtenida	Voltaje Obtenido	Potencia Consumida
DHT11	2.42 (mA)	5 (V)	12.1 (mW)

Tabla 25: Resultado análisis energético del sensor DHT11 humedad.

5.2.6.1.2 Análisis arranque.

Como se puede observar en la gráfica, en este sensor los valores de humedad tardan en estabilizarse aproximadamente 16 segundos.

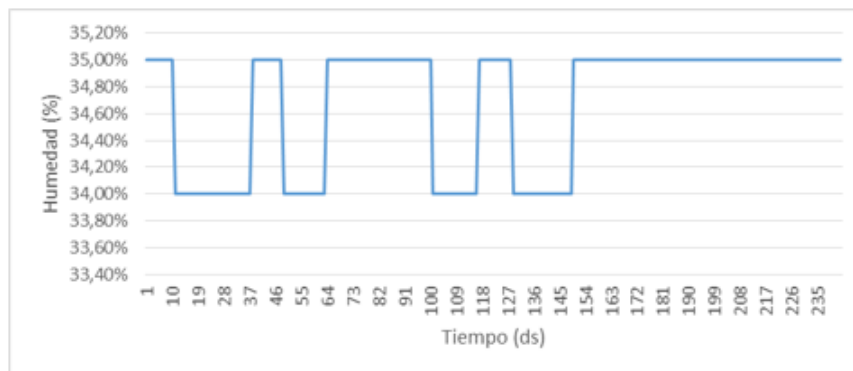


Figura 50: Análisis de arranque del sensor DHT11.

5.2.6.1.3 Análisis comunicación.

Su frecuencia de muestreo máximo es de 1 Hz, es decir, como máximo puede tomar una lectura por segundo. Es un sensor, por tanto, muy lento. Aunque puede parecer reducida en comparación a las frecuencias de otros sensores, pero la temperatura y la humedad no son valores que suelen cambiar de forma drástica en un corto espacio de tiempo.

El DHT11 es un sensor que proporciona una salida de datos digital. Entre sus ventajas podemos mencionar el bajo coste y el despliegue de datos digitales. Entre las desventajas pues, el DHT11 solo lee enteros, no podemos leer humedad con decimales, por lo cual no es muy preciso.

5.2.7 Bluetooth

El usuario conectará con el sistema mediante conexión inalámbrica bluetooth. De éste modo se le proporcionará la información necesaria acerca de las diferentes constantes vitales. El dispositivo utilizado es el modelo HC-06, se caracteriza porque solo puede actuar como esclavo, es decir, que sólo se puede conectar a un master [109]. Este módulo es de clase 2 por lo que la potencia máxima permitida es 2.5 mW y tienen un alcance máximo de 10 metros. La comunicación se realiza vía serial mediante los pines TX/RX y tiene las siguientes características:

ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA EN LA PLATAFORMA ARDUINO Y ANDROID PARA OBTENER CONSTANTES VITALES ORIENTADO A PERSONAS CON DIABETES TIPO I.

Sus principales características son las siguientes:

Tensión de alimentación	3.3 - 6(V)
Seguridad	Authentication and encryption
Velocidad Asíncrona	2.1Mbps(Max) / 160 kbps
Velocidad Síncrona	1Mbps/1Mbps
Perfiles	Bluetooth serial port
Potencia de transmisión	2,5 mW o 4 dBm
Protocolo de Bluetooth	Bluetooth Specification v2.0+EDR
Dimensiones	26.9mm x 13mm x 2.2 mm
Precio	2.93 €

Tabla 26: Tabla de características del módulo HC-06 [Ibídem].

La configuración del módulo se realiza ajustando los tres aspectos esenciales de nombre, contraseña y velocidad de transmisión mediante comandos AT que se envían mediante una tarjeta Arduino [110].

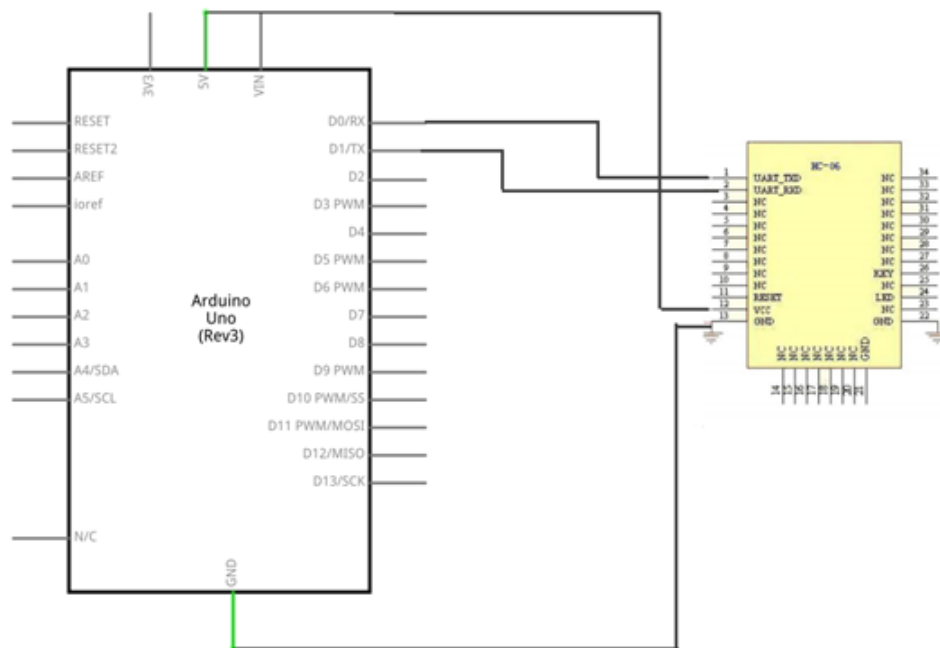


Figura 51: Circuito utilizado para analizar el módulo HC-06.

5.2.7.1 Análisis energético.

Dispositivo	Corriente Obtenida	Voltaje Obtenido	Potencia Consumida
HC-06	30 (mA)	5 (V)	150 (mW)

Tabla 27: Resultado análisis energético Bluetooth HC-06.

5.2.7.2 Conclusiones.

Se ha escogido el módulo Bluetooth HC-06, ya que es una de los dispositivos más económicos del mercado con este tipo de conectividad inalámbrica, a un precio que no alcanza la quinta parte del coste de otros dispositivos con las mismas características. Se había pensado la opción utilizar el módulo el HC-05, el cual puede funcionar como maestro/esclavo, mientras que el HC-06 solo puede actuar como esclavo. La diferencia entre maestro y esclavo es que en modo esclavo es el dispositivo quien se conecta al módulo, mientras que en modo maestro es el módulo quien se conecta con un dispositivo. Como el objetivo es que el Smartphone actúe de maestro y el módulo siempre de esclavo, se ha seleccionado el correspondiente módulo Bluetooth HC-06, ya que tiene características suficientes para poder cumplir con el objetivo propuesto.

El módulo de Bluetooth HC-06 posee buenas características de transmisión y recepción con un alcance amplio. Es de pequeño tamaño, peso ligero y de bajo consumo de corriente. Sin embargo para un futuro se deberá emplear un Bluetooth con mejores prestaciones, y que envíe de manera más síncrona los datos, ya que este tarda unos segundos en actualizarse, este debería ser el nuevo Bluetooth Low Energy, que se caracteriza por ser de bajo consumo y por tanto perfecto para dispositivos wearables.

5.3 Elaboración de un prototipo, a partir de los componentes seleccionados, para la adquisición de datos a tiempo real en Arduino

En esta sección se describe el diseño y desarrollo del software y el diseño del prototipo final, abarcando cinco puntos principales: monitorización, interoperabilidad, almacenamiento, comunicación y diseño del prototipo, incluyendo en esta parte la implementación del código QR para el reconocimiento del paciente desde la pulsera.

5.3.1 Módulo de monitorización del prototipo

A continuación se explicará el circuito y código final del prototipo. Se estudian los diferentes bloques de código que componen el sketch del microcontrolador, y se explican las clases implementadas para la monitorización de las diferentes constantes vitales.

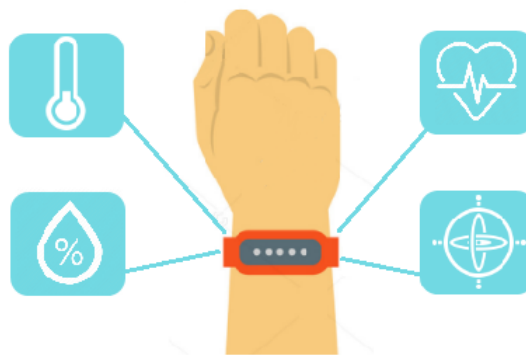


Figura 52: Sistema de monitorización del prototipo.

Los sensores escogidos van a ser por tanto los siguientes:

- Monitorización de la temperatura: sensor LM35.
- Monitorización de la humedad relativa: sensor DHT11.
- Monitorización de los ciclos del sueño: MPU6050.
- Monitorización del pulso cardíaco: APDS9008.

El programa completo se puede encontrar en el apéndice II. El programa está formado por cinco secciones: variables, void setup, void loop, librerías y funciones. A continuación se van a explicar los conceptos más importantes de cada una de ellas.

ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA EN LA PLATAFORMA ARDUINO Y ANDROID PARA OBTENER CONSTANTES VITALES ORIENTADO A PERSONAS CON DIABETES TIPO I.

El circuito final de la parte correspondiente a la monitorización es el siguiente:

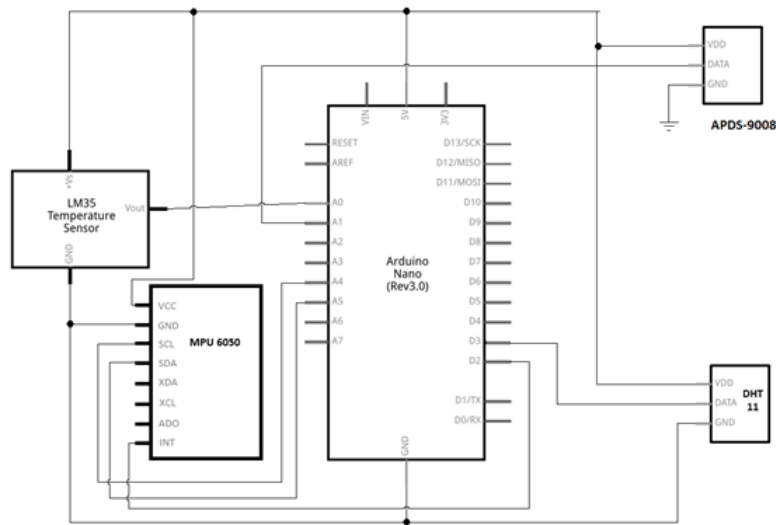


Figura 53: Circuito correspondiente a la monitorización.

5.3.1.1 Variables.

En esta parte del código se van a declarar todas las variables necesarias para el funcionamiento de todos los sensores y módulos

5.3.1.2 Void setup().

Las instrucciones que se encuentran dentro de la sección “void setup()” se ejecutan una única vez, en el momento de encender (o resetear) la placa Arduino, sirve para realizar pre-configuraciones iniciales [111].

5.3.1.3 Void loop().

Las instrucciones escritas dentro de la sección “void loop()” se ejecutan justo después de las de la sección “void setup()” infinitas veces hasta que la placa se apague (o se reinicie). Gracias a la creación de funciones, el código de la función principal se reduce, ocupando menos espacio. Tras el estudio de las posibilidades del código para Arduino y los comandos posibles a utilizar, surgió la siguiente solución al código. Se ha empleado instrucción switch, cada sección switch tiene una etiqueta case, con diferentes instrucciones del prototipo wearable. De esta manera se consigue que en función del parámetro que se reciba vía bluetooth, se ejecute una u otra instrucción.

5.3.1.3 Librerías.

Las librerías son [112] porciones de código que nos facilitan la conexión de dispositivos a Arduino, como son los diferentes sensores que se van a utilizar en este prototipo.

En el programa se hace uso de cuatro librerías. La primera librería llamada DHT.h contiene las funciones necesarias para la adquisición de valores del sensor de humedad DHT. La segunda, llamada I2Cdev.h, se utiliza para acceder a funciones que permiten conectar fácilmente módulos I^2C de otros dispositivos con la placa Arduino; la tercera, "MPU6050-6Axis-MotionApps20.h", la cual permite acceder a las funciones que a partir de datos del sensor devuelven datos de magnitudes físicas como los son la velocidad, la aceleración, la gravedad o el ángulo en cualquiera de los tres ejes; la cuarta es la librería Timer.h para la gestión de tiempos temporizados y por último es necesario incluir la biblioteca Wire.h para configurar el módulo I^2C del Arduino.

5.3.2 Módulo de interoperabilidad del prototipo

El objetivo es obtener los datos de las constantes vitales en el estándar HL7 v2.7. De tal manera que los datos recibidos en el microcontrolador generen un mensaje con este estándar implantado. HL7 (Health Level Seven) es una [113] organización sin fines de lucro que desarrolla estándares para maximizar las compatibilidades entre sistemas de información en salud, permitiendo la interacción y el intercambio productivo de datos entre aplicaciones heterogéneas, independientemente de su plataforma tecnológica o de su lenguaje de desarrollo. Por lo tanto, especifica los mensajes utilizados en cada transacción, su estructura y contenido, definiendo en este último punto, los datos que se deben enviar obligatoriamente, los datos que pueden ser enviados y como deben ser enviados. Los estados que se han configurado son los siguientes [114] [115] [116]:

- **Identificador de la cabecera (segmento MSH):** El segmento MSH forma parte de la cabecera de todo mensaje. En primer lugar se encuentra el identificador de la cabecera (MSH). Tras este campo, llega el segundo campo, donde se especifican con los caracteres especiales, que siempre son los mismos (|). El siguiente campo especifica cual es el dispositivo de envío, en este caso, SP, que identifica al manager. Como el campo de Sending Facility es opcional, no está incluido. Lo mismo ocurre con Receiving Facility. El quinto campo es el de la aplicación que recibirá el mensaje, en este caso PT (que identifica al paciente, el cliente del sistema). El séptimo campo es el de la hora a la que se ha enviado el mensaje (17:51:08 del día 16/06/2016). El siguiente campo que se especifica es el del tipo de mensaje, ADTA28ADT A05, que especifica que se va a añadir información del paciente en los siguientes segmentos que componen el mensaje. Por último, se especifica la versión del estándar, la 2.7. A continuación se puede ver un ejemplo de un mensaje recibido:

```
MSH|^~\&|SP||PT||20140616175128||ADT^A28^ADT_A05|||2.7|
```

ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.

• **Paciente (segmento PID):** Contiene la información demográfica del paciente al que se le quieren realizar las pruebas. Los datos que se introducen son DNI/ID, nombre, apellidos, fecha de nacimiento y sexo. Quedaría de la siguiente manera:

```
PID |1|87345125J |||PAVON^BUENACHE^MIRIAM ||20011029 |F|||
```

• **Pruebas solicitadas (segmento OBR):** El segmento OBR es el que contiene el detalle de las pruebas que se solicitan o que se envían. En primer lugar se encuentra el identificador del segmento (OBR), que indica que es un segmento con el detalle de la prueba clínica. El siguiente campo, 1, indica que es el primer mensaje OBR que se está enviando en el mensaje. El único campo obligatorio es el que identifica el servicio, en este caso, se especifica que la prueba que se va a tomar son la obtención de constantes vitales, cuyo identificador es 29274-8 y LN que quiere decir que el valor se refiere según codificación LOINC.

```
OBR |1|||29274-8^VITAL SIGNS MEASUREMENTS^LN
```

• **Identificador del segmento (OBX):** Para conocer la frecuencia cardíaca a tiempo real. Con cada latido del corazón, hay un cambio en el volumen de la sangre dentro de todas las arterias y capilares [49]. Este cambio del volumen influye en la presión de la pared del vaso sanguíneo, causando un cambio del tamaño de su diámetro. Este cambio en la morfología de los vasos sanguíneos afecta a la reflexión, la refracción y la absorción de la luz. Iluminando la piel con la luz, y midiendo la luz reflejada y transmitida, el cambio de volumen de sangre causado por el latido del corazón puede ser obtenido. Para la medida del pulso se ha utilizado este principio.

```
OBX |1|NM |386725007 ^Body temperature ^SNOMED-CT ||37 |C |37 |N  
|||F |||20100511220625  
OBX |2|NM |78564009 ^Pulse rate ^SNOMED-CT ||80 |bpm |60-100 |N  
|||F |||20100511220625
```

En el apéndice II se puede ver el código empleado para conseguir este objetivo.

5.3.3 Módulo de almacenamiento del prototipo

Para guardar los datos de las diferentes constantes vitales obtenidas por los distintos sensores se ha utilizado un módulo para tarjetas SD compatible con Arduino. Una tarjeta SD es una memoria no volátil, esto quiere decir que conserva los datos incluso con la pérdida de energía eléctrica. Para conseguir este objetivo se ha optado por usar el módulo Virtuabotix SD Card.

*ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.*

Sus principales características son:

Voltaje de operación	3.3-5(V)
Tamaño	A partir de 1 (Gb)
Comunicación	SPI
Número conexiones	7

Tabla 28: Características módulo Virtualbotix SD Card [117].

Este módulo utiliza una comunicación SPI, esto significa que es un protocolo de comunicación síncrona de 4 hilos, que puede alcanzar altas velocidades de transmisión, y que envía y recibe los datos a los periféricos al mismo tiempo [118]. La forma de guardar los datos en la tarjeta SD para que luego puedan ser leídos y representados es creando un archivo de texto llamado DATOS.txt que tendrá la forma del siguiente ejemplo. El código final se puede ver en el apéndice II.

1	2	3	4	5
23.44,	-24.65	78.35	3.95,	454 -692 -46
			79,	34

Figura 54: Muestra de datos.

En este caso los parámetros corresponden a:

1	Temperatura
2	Acelerómetro
3	Giróscopo
4	Humedad
5	Pulso cardíaco

Tabla 29: Correspondencia parámetros y micro SD.

ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA EN LA PLATAFORMA ARDUINO Y ANDROID PARA OBTENER CONSTANTES VITALES ORIENTADO A PERSONAS CON DIABETES TIPO I.

El circuito empleado correspondiente al módulo de almacenamiento es el siguiente:

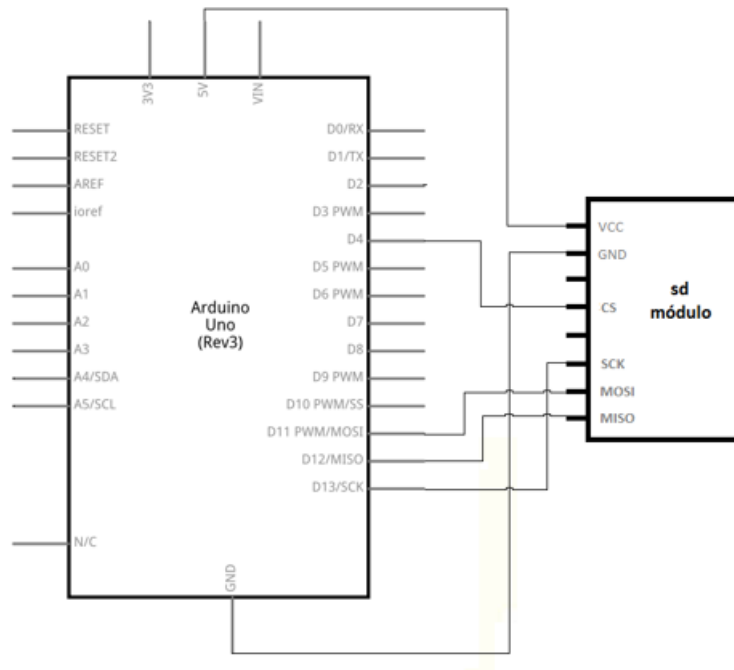


Figura 55: Circuito módulo almacenamiento.

Sin embargo, añadir esta característica provoca inestabilidad al programa debido a que la librería empleada para la SD ocupa mucho espacio y se ha decidido no dejarla en el prototipo final, ya que Arduino Nano tiene una memoria limitada.

5.3.4 Módulo de comunicación del prototipo

Se ha empleado el módulo de Bluetooth HC-06. El módulo Bluetooth HC-06 es una de las soluciones más útiles y económicas para lograr una comunicación inalámbrica de calidad vía Bluetooth con un microcontrolador como Arduino. El esquema de conexión será el siguiente:

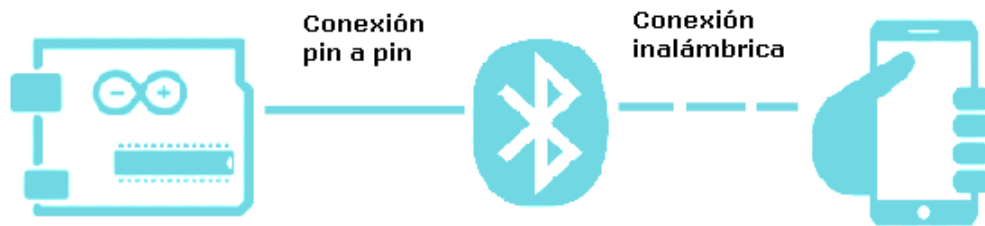


Figura 56: Conexión microcontrolador y dispositivo móvil.

El HC-06 dispone de 4 pines, de los cuales los dos primeros de la izquierda sirven para alimentar el módulo. Los dos siguientes pines, TXD y RXD, permiten enviar y recibir información a través de un puerto Serial y se conectan con los correspondientes pines TX y RX del microcontrolador de manera cruzada [119].

Este módulo viene de fábrica configurado de una manera predeterminada. Para modificar esta configuración, se hace uso de los comandos AT, los cuales son comandos en código ASCII y son enviados desde una terminal Serial al módulo [120]. Con dichos comandos es posible asignar un nombre al módulo Bluetooth, cambiar la clave de acceso y configurar la velocidad.

Primero se ha procedido a verificar la comunicación usando el monitor serial de Arduino. Para inicial la configuración hay que introducir los comandos AT.

```
Enviado: AT
```

```
Recibido: OK
```

En caso de no recibir un 'OK', significaría que no existe una respuesta del módulo. Esto se debe a que no existiría comunicación con el módulo Bluetooth.

A continuación, hay que configurar la velocidad del módulo. Para ello se ejecuta el siguiente comando:

```
Enviado: AT+BAUD4
```

```
Recibido: OK9600
```

En este caso se ha configurado para que la velocidad sea de 9600 baudios de comunicación.

*ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.*

Para configurar los parámetros hay que tener en cuenta la siguiente tabla:

1	1200
2	2400
3	4800
4	9600
5	19200
6	38400
7	57600
8	115200

Tabla 30: Velocidades comunicación HC-06

Para modificar el nombre y la contraseña, hay que introducir los siguientes comandos:

```
Enviado: AT+NAMEnombre dispositivo
Recibido: OKname
Enviado: AT+PINxxxx
Recibido: OKsetpin
```

5.3.5 Diseño del prototipo

En este apartado se presenta el diseño de las piezas 3D que integran el wearable. Estas son: la base y la tapa. Se ha empleado el programa OpenScad para su desarrollo. Los diseños se crean a partir de figuras geométricas. Las utilizadas son las siguientes [121] [122]:

cube: tiene la sintaxis `cube([x,y,z],center,fn)`. Ejemplo: `cube([45, 5, 2], center = true, fn=20)`. Permite definir un cubo en sus ejes x, y, z. La opción `center` es para indicar si el centro del objeto se encuentra en el centro de coordenadas (`true`) o si es uno de los extremos del elemento el que se encuentra en el origen de coordenadas (`false`). La opción `fn` sirve para indicar la resolución de la figura. En cuanto a las unidades, estas son en milímetros.

cylinder: la sintaxis `cylinder(h,r,center)`. Ejemplo: `cylinder(h = 2, r = 5, center = true)`. Permite definir un cilindro mediante su altura `h` y su radio `r`. La opción `center` y `fn` tienen la misma funcionalidad que la explicada en la sentencia `cube`.

sphere: tiene la sintaxis `sphere(r, center)`. Ejemplo: `sphere(30, true)`. Permite definir esferas de un radio `r`. La opción `center` y `fn` tienen la misma funcionalidad que la explicada en la sentencia `cube`. Estos elementos pueden combinarse entre ellos mediante

diversas operaciones para conseguir elementos más complejos: translate, para desplazar los objetos en los ejes x, y, z respectivamente; rotate, para rotar la figura; difference, para restar figuras entre sí; o unión, para unir figuras entre sí. Uno de los prototipos se encuentra a continuación.

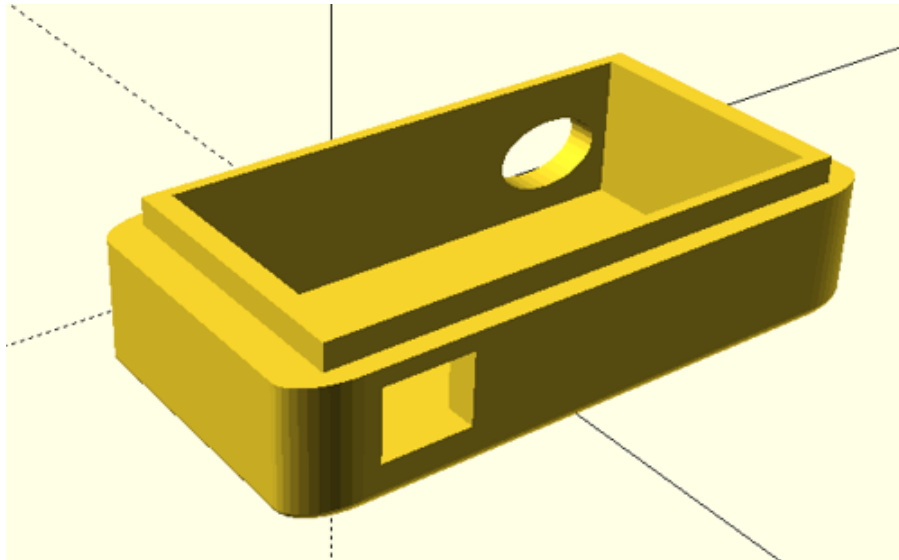


Figura 57: Código QR con la web optimizada para la descarga.

5.3.5.1 Implementación código QR.

Los códigos QR (Quick Response Barcode) son un sistema para almacenar información en un código de barras bidimensional. Fueron ideados en 1994 por la compañía DENSO Wave, de Toyota [123]. Utiliza como formato un código de barras matricial (o bidimensional), que se diferencia de los códigos de barra tradicionales por tener mayor capacidad de almacenamiento y contener información alfanumérica [124].

En este proyecto se ha implementado este proceso. Este código se encontrará en el wearable, de tal manera que en caso de accidente, cualquier persona puede identificar tanto al paciente como sus problemas médicos, reduciendo los riesgos a este. El sistema de identificación QR también facilita y optimiza los servicios médicos al personal sanitario, ya que agiliza la obtención de la información del paciente.

El código QR empleado se muestra en la figura 57, este código transfiere directamente a una web optimizada para dispositivos móviles donde el usuario podrá descargar la aplicación móvil en la cual tendrá acceso a la información de un paciente en particular.

*ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.*

Este es un código QR correspondiente a un paciente así como el direccionamiento a la web optimizada donde se puede descargar la aplicación móvil.



Figura 58: Ejemplo de prototipo empleado para el wearable.

5.4 Desarrollo aplicación Android

En esta sección se explicará los resultados de la aplicación Android realizada. Primero se seleccionarán un subconjunto de características de qué es lo que la aplicación tendrá que hacer y cómo habrá de hacerlo mediante una especificación de requisitos. Se muestra una vista global de la arquitectura pensada para el sistema. En función de estas características, por último se mostrará el resultado final de la misma.

5.4.1 Introducción

El sistema que se ha modelado cuenta con una serie de servicios que se detallan a continuación:

1.- Monitorización: El sistema va a monitorizar de manera remota las constantes vitales del paciente (temperatura, pulso, humedad y ciclos de sueño).

2.- Entretenimiento: Este servicio estará formado por una lista de tareas y una calculadora del índice de masa corporal (IMC). Con la lista de tareas el usuario podrá añadir o eliminar diferentes labores que tenga que hacer. Con la calculadora de IMC se calculará este valor en función de los datos introducidos de peso y altura.

3.- Formación: Se añadirá al sistema una opción de formación para gente con diabetes de tipo I. Se realizará mediante tutoriales en forma escrita.

4.- FAQs: En esta sección se responden dudas frecuentes acerca de la aplicación. Proporciona explicaciones de manera simple para saber utilizarla.

5.- Mantenimiento: Este servicio va a permitir que el sistema siga funcionando y adaptándose a nuevas especificaciones. Mediante este servicio se podrá añadir, eliminar, modificar usuarios y contraseñas.

5.4.2 Contorno del sistema. Actores

A continuación se muestran los diferentes actores de los que está compuesto la plataforma:

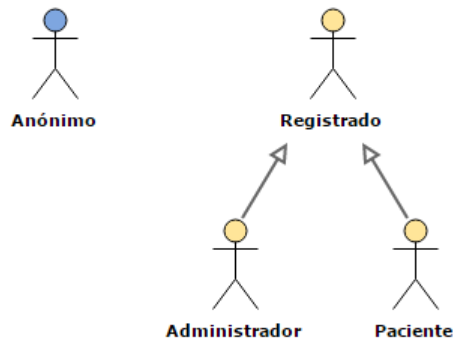


Figura 59: Actores del sistema.

■ **Usuario Anónimo:** es un usuario externo, accede a la aplicación de forma normal, sin estar registrado en la base de datos de usuarios. No se ha identificado en el sistema, por lo que sólo puede tener acceso determinadas secciones. Sus funciones son:

- Consulta datos del paciente.

■ **Usuario Registrado:** está identificado en el sistema. Tiene acceso a una parte restringida del software. Hereda todas las funciones propias del usuario anónimo. Dentro del grupo de “usuarios registrados”, hay dos tipos de perfiles: pacientes y administrador.

Pacientes: Es la persona con diabetes de tipo 1. A continuación se nombran las distintas opciones que se permiten con el usuario de tipo dependiente:

- Monitorizar sus constantes vitales.
- Acceso a tutoriales.
- Contactar con el administrador u otros usuarios vía email.
- Acceso a IMC y lista de tareas.
- Acceso a FAQ.

Administrador: Aquella con los permisos necesarios para llevar a cabo las funciones administrativas y con acceso a la base de datos para sus posibles modificaciones. Además de visualizar, puede gestionar archivos y dar de alta y baja usuarios.

5.4.3 Requisitos del sistema

La especificación de requisitos va a recoger las características que va a contener la aplicación, y son las siguientes:

5.4.3.1 Requisitos funcionales

Los requisitos funcionales son los siguientes:

RF-001: La aplicación debe autenticar al usuario mediante una pantalla de login.

RF-002: El sistema se encargará de validar y permitir o denegar el acceso a la aplicación.

RF-003: El sistema mostrará un mensaje de error en el caso de que la validación no sea correcta o falte algún dato.

RF-004: En el caso de que la validación sea correcta, se redirigirá automáticamente a la pantalla principal de la aplicación.

RF-005: El sistema deberán monitorizar las siguientes constantes vitales del paciente: temperatura, humedad, pulso cardíaco y ciclos del sueño. Para el parámetro de humedad se mostrará en %, la temperatura corporal en grados Celsius, para la presión arterial en pulsaciones por minuto.

RF-006: El sistema deberá permitir almacenar los datos.

RF-007: El sistema mostrará deberá permitir enviar datos bajo el estándar HL7 v2.7

RF-008: El usuario debe poder crear tareas.

RF-009: El usuario debe poder identificar estas tareas mediante un título.

RF-010: El usuario debe poder suprimir una tarea.

RF-011: El usuario debe poder ver las tareas en forma de lista.

RF-012: El usuario será capaz de visualizar una lista de tutoriales sobre la diabetes.

RF-013: El usuario será capaz de calcular su IMC.

RF-014: El sistema debe ser capaz de generar un consejo en función del IMC.

5.4.3.2 Requisitos no funcionales

Los requisitos no funcionales son los siguientes:

RNF-001: Alta usuario: Permite dar de alta a un usuario en el sistema generando un identificador de usuario automáticamente, comprobando que este no exista en el sistema. Esta funcionalidad solo está disponible para un usuario de tipo administrador.

RNF-002: Modificar usuario: Permite modificar la información de un usuario. La modificación estará a cargo del administrador, mediante petición del usuario.

*ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.*

RNF-003: Dar de baja a usuario: Permite dar de baja a un usuario en el sistema. Esta funcionalidad solo está disponible para un usuario de tipo administrador.

RNF-004: Para iniciar sesión el usuario deberá identificarse con su nombre de usuario y contraseña correspondiente.

RNF-005: Los datos personales y contraseñas serán cifrados con md5.

RNF-006: Manual de usuario de la aplicación.

RNF-007: Funcionará correctamente desde la versión 4.0 de Android.

RNF-008: La aplicación deberá funcionar con total fluidez en dispositivos con 1 (GB) de RAM y un procesador con un solo núcleo a 1 (GHz).

RNF-009: La app tendrá que ser capaz de mantener la información introducida por el usuario y poder recuperarse de posibles cambios sin perder esta. Estos posibles cambios incluyen entre otros el cambio de orientación del dispositivo, o llamadas entrantes durante el uso de la aplicación.

RNF-010: La interfaz deberá ser sencilla, atractiva, robusta e intuitiva. De tal manera que su uso sea accesible para todo el mundo.

RNF-011: La interfaz tendrá un diseño adaptable a distintos tamaños y tipos de pantalla, por lo que funciona en cualquier dispositivo móvil.

RNF-012: La aplicación ofrecerá usabilidad en todo dispositivo móvil que opere bajo el sistema operativo Android.

RNF-013: Uso de diálogos estéticos y diseño minimalista.

RNF-014: Uso de un lenguaje sencillo para que lo entienda cualquier tipo de usuario. Se evitara tecnicismos y frases complejas. Uso de iconos con metáforas para conseguir este requisito.

5.4.4 Diagramas

A continuación se representan los distintos tipos de diagramas usados: de casos de uso, de clases y de actividad.

5.4.4.1 Casos de uso

A continuación se muestran los diagramas de casos de uso del sistema.

5.4.4.1.1 Monitorización

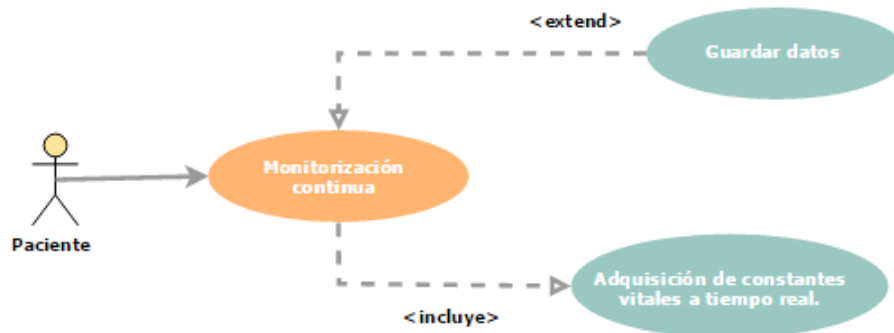


Figura 60: Caso de uso de Monitorización.

Caso de uso: Monitorización continua.

Actor: Paciente.

Precondición: Usuario registrado.

Escenario principal:

1. El caso de uso comienza cuando el usuario elige la opción de monitorización continua.
2. El sistema almacena y muestra de manera continua las constantes vitales del paciente.
3. El usuario elige la opción de finalizar la monitorización continua.

Escenario alternativo:

- 1.1 En caso de un fallo al enviar los datos de monitorización, el sistema avisa al usuario mediante la sentencia: 'Error en el envío de datos'.

5.4.4.1.2 Formación

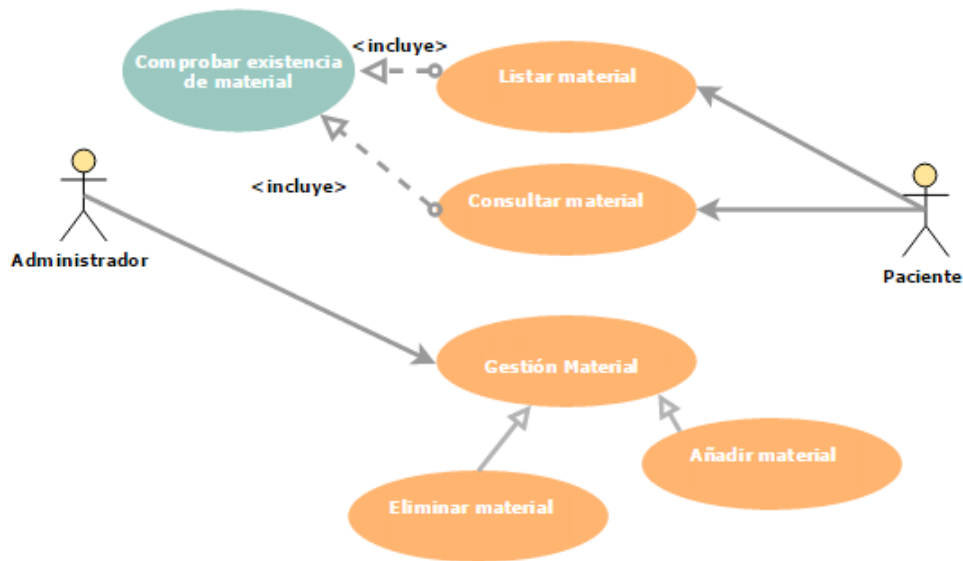


Figura 61: Caso de uso de Formación.

Caso de uso: Añadir material formativo.

Actor: Administrador.

Precondición: Usuario registrado.

Escenario principal:

1. El administrador accede a la sección de material formativo.
 2. El administrador añade los tutoriales.
 3. El administrador introduce el título, breve descripción y asigna a los usuarios que tienen permiso para acceder a ellos.
 4. La información se añade para que pueda ser usada.
-

Caso de uso: Consultar material formativo.

Actor: Paciente.

Precondición: Usuario registrado.

Escenario principal:

1. El paciente accede a la sección de material formativo.
2. El paciente selecciona el material a consultar.
3. El sistema le devuelve el material para que lo visualice.

Escenario alternativo:

3.1 No existe el material a consultar.

Caso de uso: Guardar material formativo.

Actor: Paciente.

Precondición: Usuario registrado.

Escenario principal:

1. El paciente accede a la sección de material formativo.
2. Se ofrece una lista de material formativo y el usuario lista el material que considere.

Escenario alternativo:

3.1 No existe el material que se quiere guardar.

5.4.4.1.3 Entretenimiento

5.4.4.1.3.1 Lista de tareas

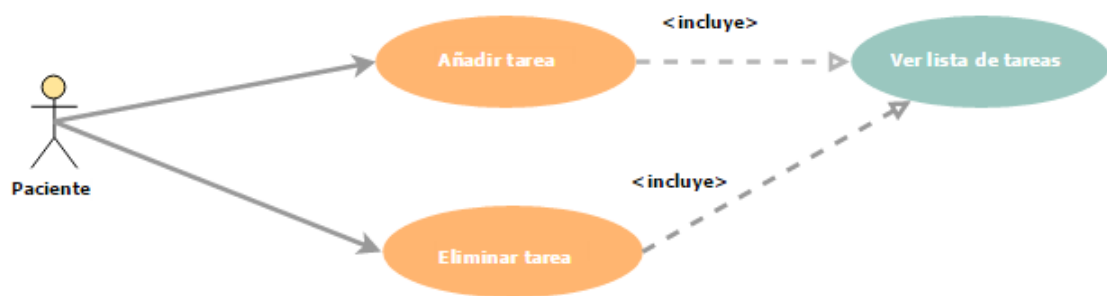


Figura 62: Caso de uso de Lista de Tareas.

Caso de uso: Añadir tarea.

Actor: Paciente.

Precondición: Usuario registrado.

Escenario principal:

1. El usuario elige la opción de añadir tarea.
2. Al pulsar el botón de aceptar, se añade la tarea en la base de datos del sistema.

Escenario alternativo:

2.1 Si el usuario solicita cancelar la operación, el sistema cancela la operación, y se vuelve a la pantalla anterior.

Caso de uso: Eliminar tarea.

Actor: Paciente.

Precondición: Usuario registrado.

Escenario principal:

1. El usuario elige la opción de eliminar tarea.
2. El sistema muestra al usuario todas las tareas existentes.
3. El usuario selecciona la tarea que quiere eliminar de la agenda.
4. Al pulsar el botón de aceptar, se elimina la tarea de la base de datos del sistema.

Escenario alternativo:

4.1 Si el usuario solicita cancelar la operación, el sistema cancela la operación, y se vuelve a la pantalla anterior.

5.4.4.1.3.2 Cálculo IMC

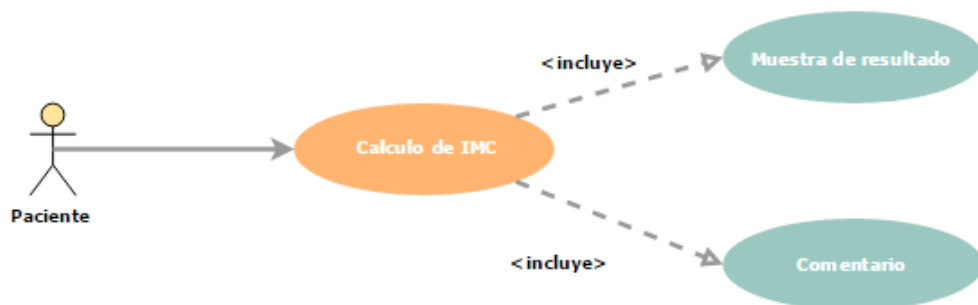


Figura 63: Caso de uso de Cálculo de IMC.

Caso de uso: Calculo de IMC.

Actor: Paciente.

Precondición: Usuario registrado.

Escenario principal:

1. El usuario elige la opción de calcular IMC.

*ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.*

2. El usuario introduce los datos de peso y altura.
3. El sistema calcula el IMC y muestra el resultado con un comentario acerca del resultado.

Escenario alternativo:

- 3.1 Si el usuario solicita cancelar la operación, el sistema cancela la operación, y se vuelve a la pantalla anterior.

5.4.4.2 Diagramas de clase

A continuación se muestran los diagramas de clase del sistema.

5.4.4.2.1 Diagrama de clases de actores

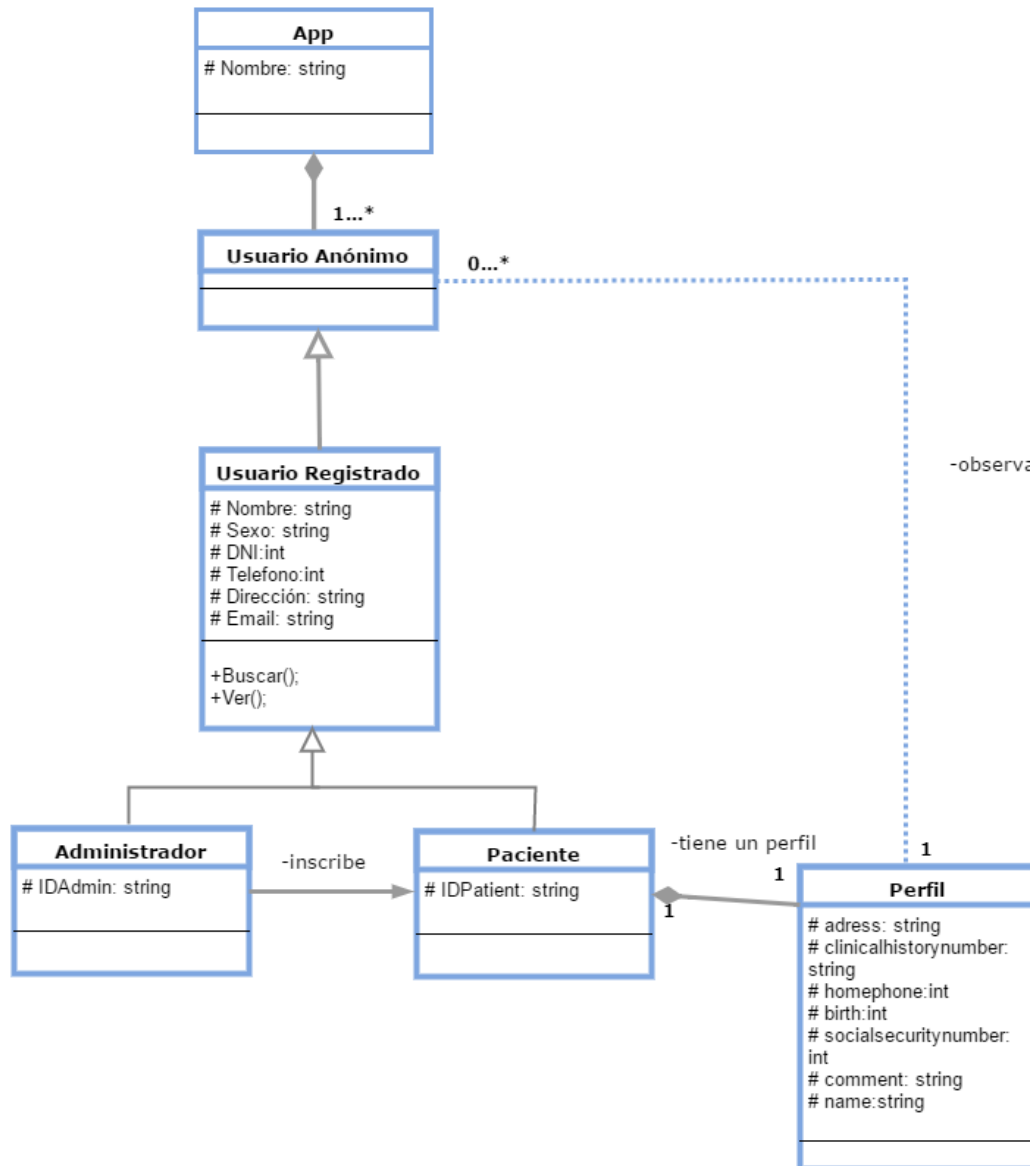


Figura 64: Diagrama de clases de actores.

5.4.4.2 Diagrama de clases de monitorización

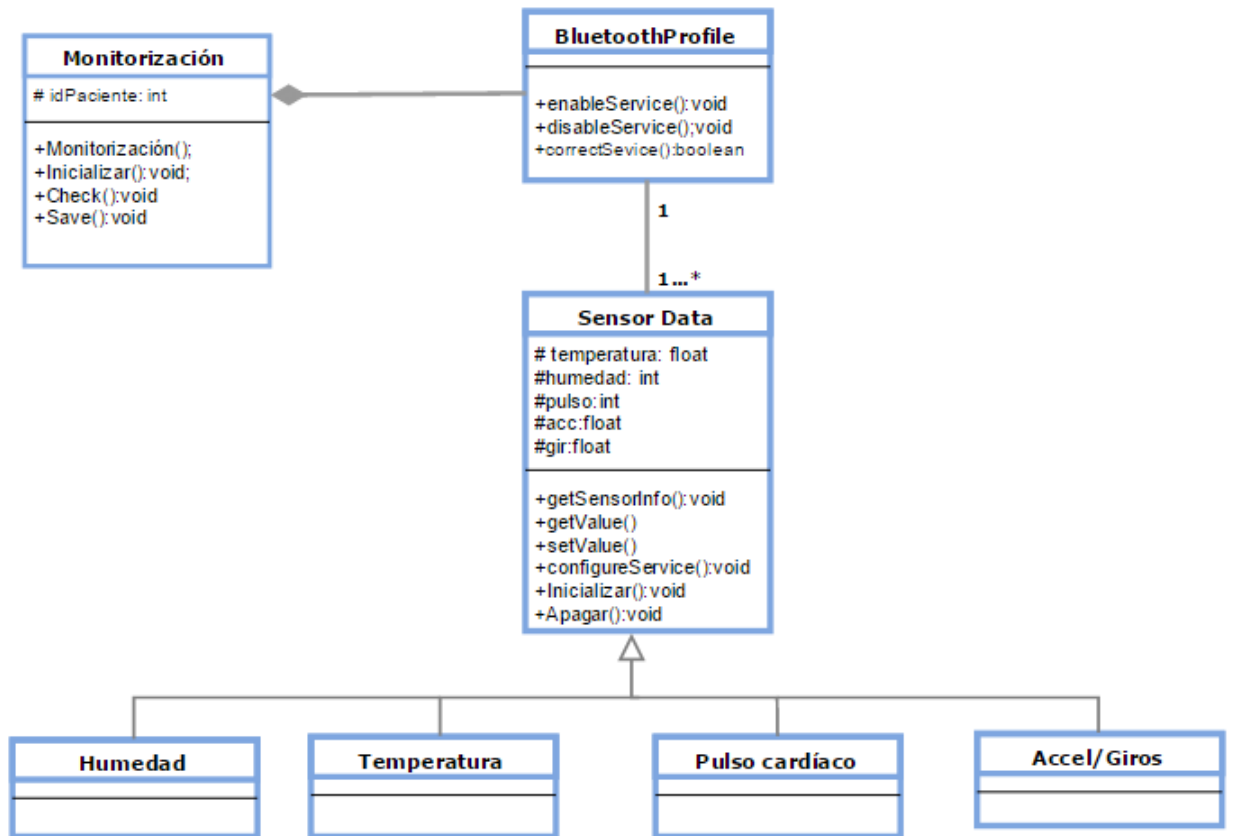


Figura 65: Diagrama de clases de Monitorización.

5.4.4.2.3 Diagrama de clases de mantenimiento del sistema

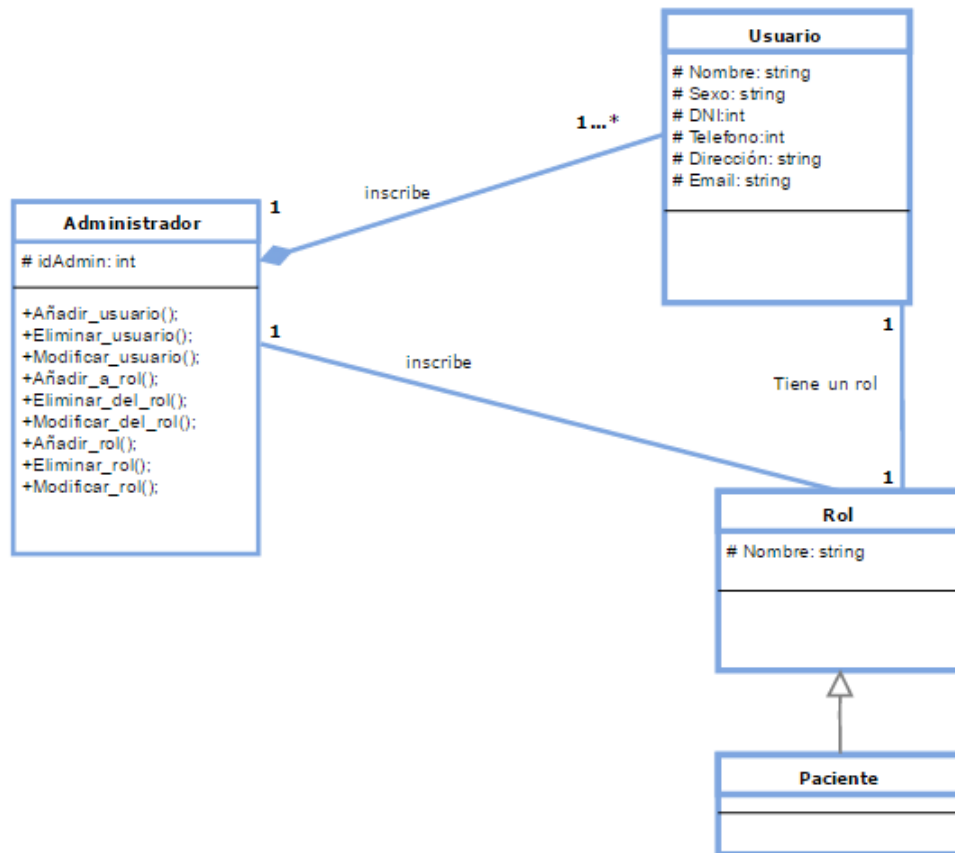


Figura 66: Diagrama de clases de Mantenimiento.

5.4.4.2.4 Diagrama de clases de entretenimiento

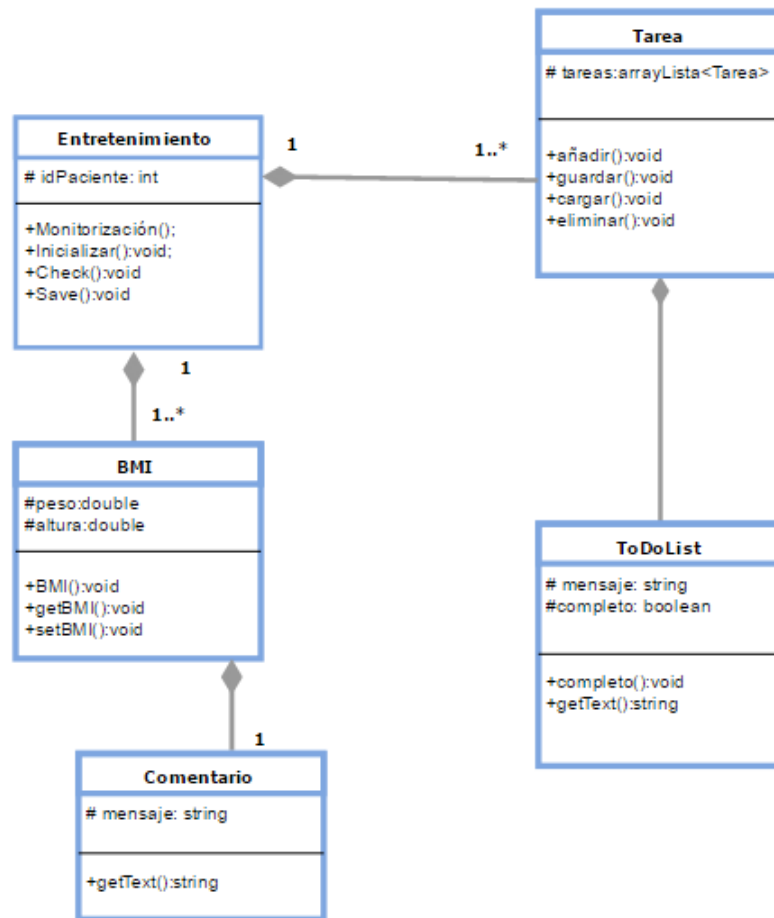


Figura 67: Diagrama de clases de Entretenimiento.

5.4.4.3 Diagramas de secuencia

A continuación se muestran los diagramas de secuencia del sistema.

5.4.4.3.1 Formación

El paciente accede a la sección de formación y selecciona un material formativo. Se comprueba la existencia de ese material, si existe, se envía el recurso.

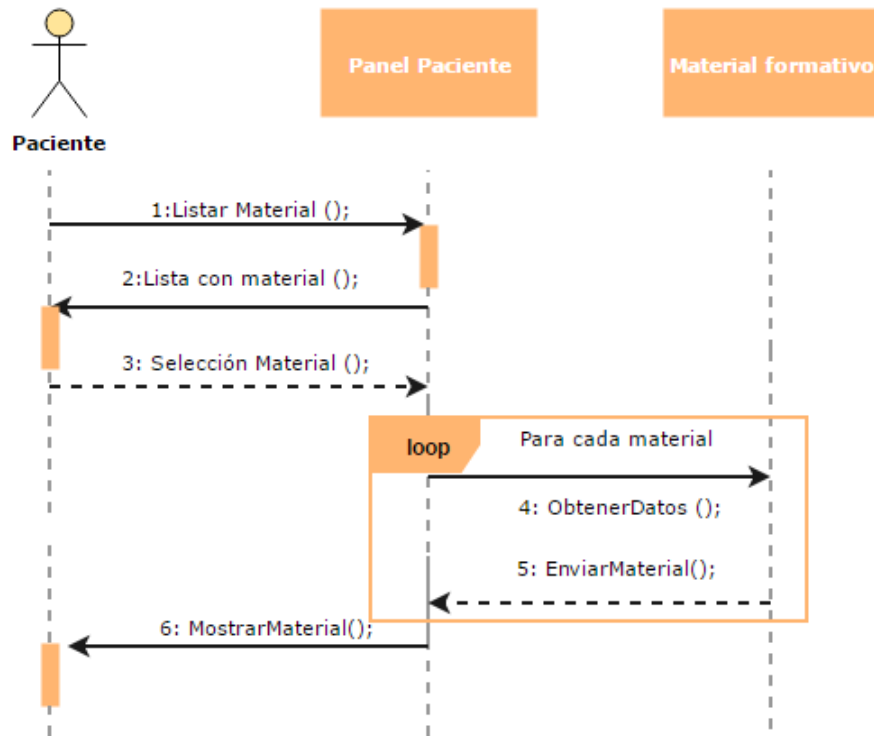


Figura 68: Diagrama de secuencia visualizar material formativo.

5.4.4.3.2 Lista de tareas

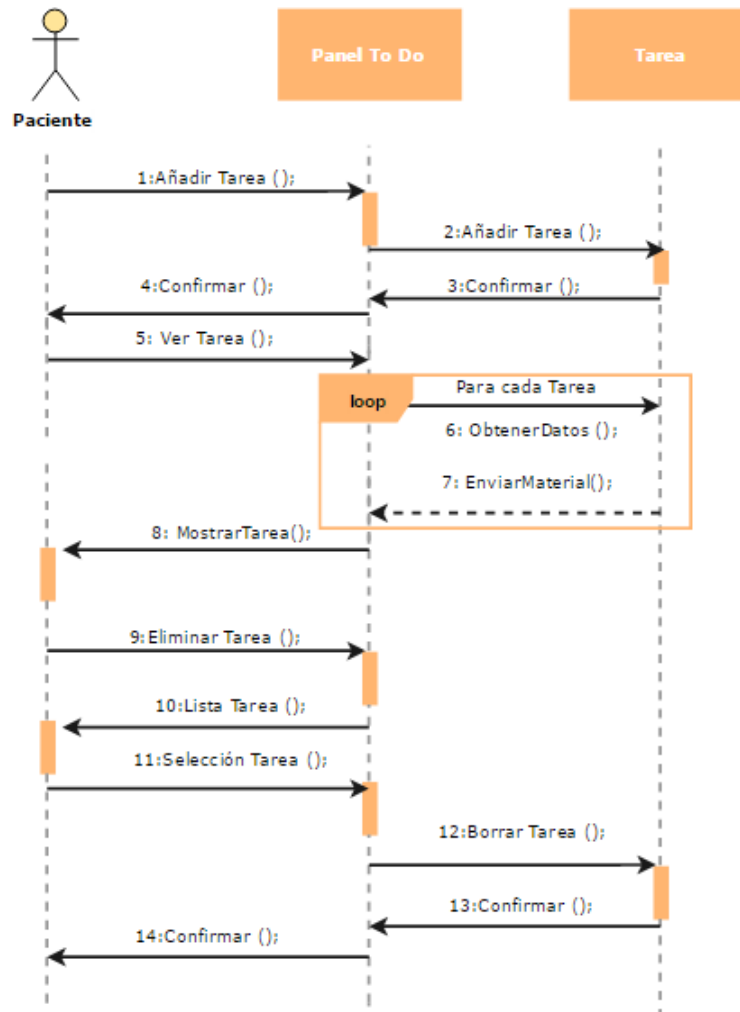


Figura 69: Diagrama de secuencia de lista de tareas.

5.4.5 Arquitectura

A continuación se explica la arquitectura seguida para la comunicación de la pulsera con la aplicación móvil, así como la arquitectura cliente-servidor empleada para loguear a los diferentes pacientes.

5.4.5.1 Arquitectura de comunicación

La arquitectura del funcionamiento del dispositivo se muestra en la figura 70. La información de la pulsera se enviará mediante Bluetooth al dispositivo móvil del paciente.

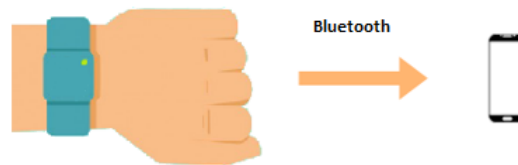


Figura 70: Diagrama de la arquitectura de la aplicación móvil.

De esta manera el Bluetooth se usará para la comunicación de la información de manera inalámbrica de los distintos sensores y módulos de la pulsera inteligente al dispositivo móvil del paciente. Se ha escogido esta opción debido a que establece conexiones con poco gasto energético, es una tecnología ampliamente extendida y su módulo tiene un bajo coste. Además, transmite los datos de forma segura.

5.4.5.2 Cliente-Servidor

La arquitectura del sistema correspondiente al login del usuario es de cliente-servidor, donde la aplicación se divide en un componente servidor, encargado de ofrecer los servicios y un conjunto de clientes que usan dichos servicios [125]. La aplicación se conecta con un servidor de base de datos. La base de datos contiene los usuarios, emails, id y contraseñas de los pacientes que vayan a usar el sistema. Se puede ver en la siguiente imagen.

+ Opciones							
			id	nombre	contra	email	
☐	 Editar	 Copiar	 Borrar	1	user	pass	ss@ss.com
☐	 Editar	 Copiar	 Borrar	2	miriam	pass	miriam@miriam.com

Figura 71: Base de datos correspondientes a los pacientes.

Estos datos se guardan en una base de datos MySQL, pero el algoritmo usado está escrito en PHP. Los scripts PHP son los que se encargan de validar a los usuarios de la aplicación. Para conseguir esto se ha usado un servidor Apache.

*ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.*

A continuación se muestra un diagrama entre la relación de la aplicación y el servidor.

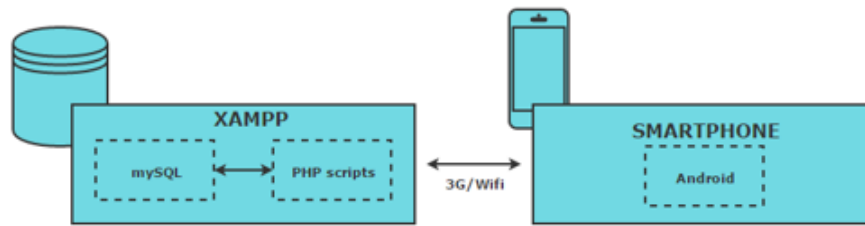


Figura 72: Diagrama del servidor.

5.4.6 Interfaz de usuario

La interfaz de usuario se ha basado en los siguientes estándares del European Telecommunications Standards Institute (ETSI): ETSI EG 202 116 [126] para el diseño de entornos, productos y servicios que sean usables para todo el mundo sin necesidad de adaptación o diseño especializado y la ETSI EG 202 416 [127] para el diseño de interfaces de usuario para terminales móviles. También se ha realizado en función del estándar ISO/IEC TR 19766 [128] que es para el diseño de iconos y símbolos accesibles para todo tipo de usuarios, incluyendo personas mayores y con discapacidades. También se ha basado en los principios de Diseño para todos. Por lo tanto, para el diseño de la aplicación se han seguido todos esos criterios, haciendo particular énfasis en:

- Multimodalidad.
- Acceso universal.
- Uso natural.

El smartphone que se ha usado para desarrollar y testear la aplicación es un Alcatel Pop C5 de cinco pulgadas, con un procesador dual-core a 1.3 (GHz). La versión de Android de este dispositivo es de 4.2 pero la aplicación es compatible a partir de Android 2.3. A continuación se presentan las distintas pantallas desarrolladas.



Figura 73: Pantalla de inicio y menú principal.

*ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.*



Figura 74: Menú de monitorización y medida de humedad.

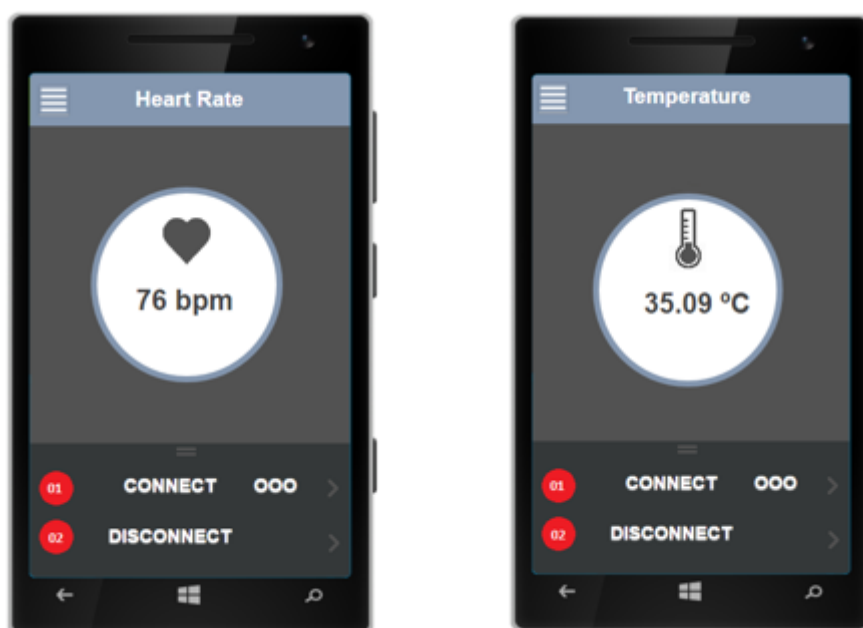


Figura 75: Menú de medida de pulso cardíaco y de temperatura.

*ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.*

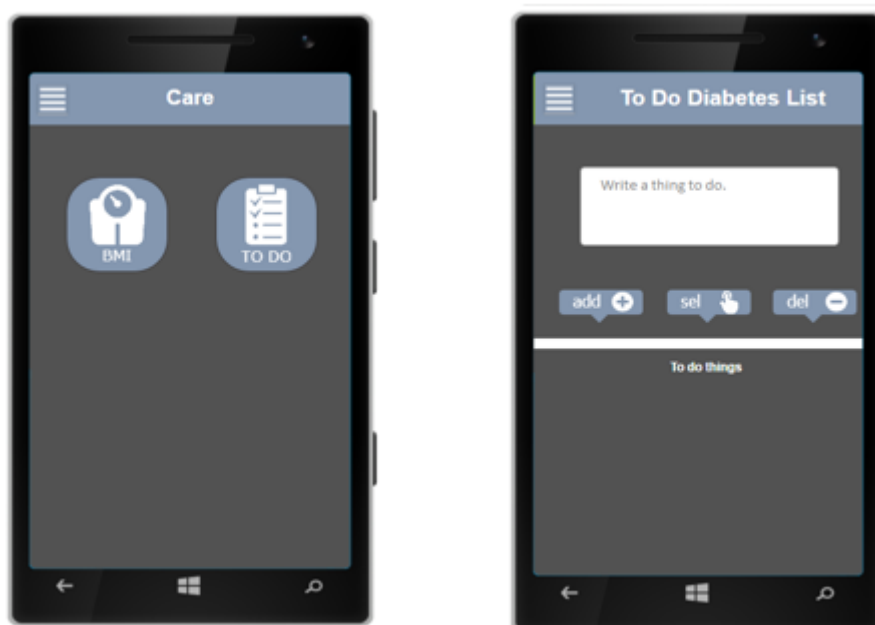


Figura 76: Menú de entretenimiento y listado de tareas.

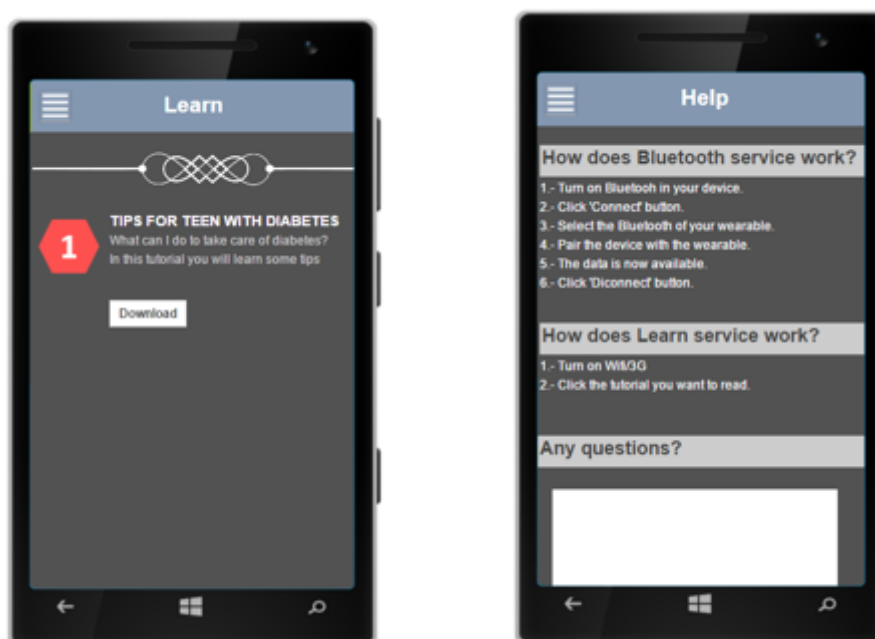


Figura 77: Menú de formación y de ayuda al usuario.

06/

6 Gestión del proyecto

6.1 Planificación del proyecto

A continuación se va a mostrar la planificación y programación de las actuaciones que se han realizado para la realización de este proyecto, se ha hecho mediante el diagrama de Gantt. Este método sirve para conocer los objetivos diarios, así como el tiempo conveniente para finalizarlos.

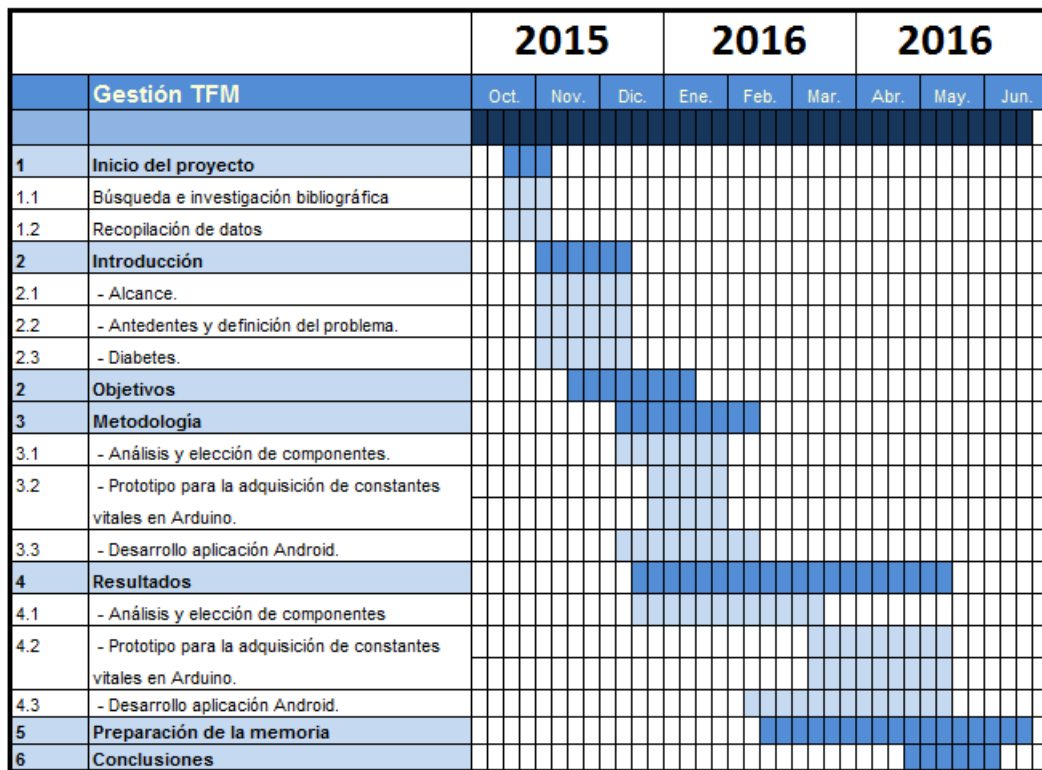


Figura 78: Diagrama de Gantt del proyecto.

*ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.*

6.2 Presupuesto

En este apartado se va a mostrar detalladamente el presupuesto total destinado al desarrollo del proyecto. Se ha dividido en ocho capítulos y un resumen final:

- Capítulo 1. Sistema de Temperatura.
- Capítulo 2. Sistema de Humedad.
- Capítulo 3. Sistema de Pulsioximetría.
- Capítulo 4. Sistema de Acelerometría y Giróscopo.
- Capítulo 5. Sistema de Comunicación.
- Capítulo 6. Sistema de Control
- Capítulo 7. Sistema de Almacenamiento.
- Capítulo 8. Diseño del Prototipo.

CÓDIGO	Ud.	RESUMEN	CANTIDAD	PRECIO	TOTAL
CAPÍTULO 01 SISTEMA DE TEMPERATURA					
01.01	Ud.	Sensor LM35	2.0	1,88	3,76
01.02	Ud.	Sensor DHT11	2.0	2,93	5,86
01.03	Ud.	Sensor MPU6050	1.0	3,91	3,91
TOTAL CAPÍTULO 01 SISTEMA TEMPERATURA.....					13,53
CAPÍTULO 02 SISTEMA DE HUMEDAD					
02.01	Ud.	Sensor DHT11	1.0	2,93	2,93
TOTAL CAPÍTULO 02 SISTEMA DE HUMEDAD.....					2,93
CAPÍTULO 03 SISTEMA DE PULSIOXIMETRÍA					
03.01	Ud.	Sensor Pulse Sensor	2.0	2,70	5,4
03.02	Ud.	Sensor <u>Heart rate click</u>	1.0	17,00	19,00
TOTAL CAPÍTULO 03 SISTEMA PULSIOXIMETRÍA.....					24,40

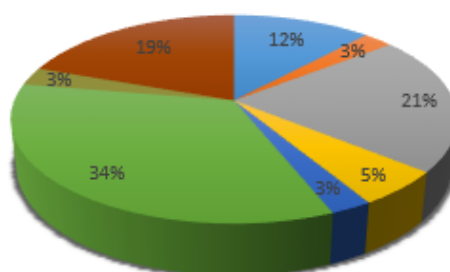
*ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.*

CÓDIGO	Ud.	RESUMEN	CANTIDAD	PRECIO	TOTAL
CAPÍTULO 04 SISTEMA DE ACELOROMETRÍA Y GIRÓSCOPO					
03.01	Ud.	Sensor MPU6050	1.0	3,91	3,91
03.02	Ud.	Sensor ADXL345	1.0	2,32	2,32
TOTAL CAPÍTULO 04 SISTEMA ACELOROMETRÍA Y GIRÓSCOPO.....					6,23
CAPÍTULO 05 SISTEMA DE COMUNICACIÓN					
03.01	Ud.	Bluetooth HC-06	1.0	2,93	2,93
TOTAL CAPÍTULO 05 SISTEMA DE COMUNICACIÓN.....					2,93
CAPÍTULO 06 SISTEMA DE CONTROL					
06.01	Ud.	Arduino UNO	1.0	14,88	14,88
06.02	Ud.	Arduino Nano	1.0	21,60	21,60
06.03	Ud.	USB	1.0	1,97	1,97
TOTAL CAPÍTULO 06 SISTEMA DE CONTROL.....					38,45
CAPÍTULO 07 SISTEMA DE ALMACENAMIENTO					
07.01	Ud.	Lector tarjeta Micro SD	1.0	3,19	3,19
TOTAL CAPÍTULO 07 SISTEMA DE ALMACENAMIENTO.....					3,19
CAPÍTULO 08 DISEÑO PROTOTIPO					
08.01	Ud.	Cables jumper	40.0	0,12	4,80
08.02	Ud.	Pegamento	1.0	2,50	2,50
08.03	Ud.	Protoboard 400 puntos	1.0	8,50	8,50
08.04	Ud.	Protoboard 120 puntos	1.0	6,50	6,50
TOTAL CAPÍTULO 08 DISEÑO PROTOTIPO.....					22,30

*ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.*

RESUMEN DEL PRESUPUESTO

CAPÍTULO	RESUMEN	IMPORTE	%
01	SISTEMA DE TEMPERATURA	13,53	12
02	SISTEMA DE HUMEDAD	2,93	3
03	SISTEMA DE PULSIOXIMETRÍA	24,40	21
04	SISTEMA DE ACELEROMETRÍA Y GIRÓSCOPO	6,23	5
05	SISTEMA DE COMUNICACIÓN	2,93	3
06	SISTEMA DE CONTROL	38,45	34
07	SISTEMA DE ALMACENAMIENTO	3,19	3
08	DISEÑO DEL PROTOTIPO	22,30	19
PRESUPUESTO TOTAL DE EJECUCIÓN DE MATERIAL (€)		113,96	100%



07/

7 Conclusiones y líneas futuras

Las conclusiones finales del proyecto se van explicar comprobando los diferentes objetivos propuestos para el mismo. Después se proponen las líneas futuras con las que se podría continuar el proyecto.

7.1 Conclusiones

A continuación, se muestran las conclusiones obtenidas a partir de los objetivos planteados en el capítulo 2.

7.1.1 Objetivo 1. ‘Análisis y elección de componentes’.

‘Selección de los parámetros a medir y elección de diferentes sensores en función de estos parámetros’.

Se ha llevado a cabo un estudio riguroso de las constantes vitales de importancia a medir en pacientes con DM1, estas finalmente han sido: temperatura, ciclos del sueño, humedad y pulso. Una vez seleccionadas, se ha podido realizar un análisis de necesidades para de esta manera seleccionar diferentes sensores y módulos que cumplieran los requerimientos de precio, dimensiones, etc, para posteriormente analizarlos.

‘Análisis energético, de arranque, de estabilidad y de comunicación de diferentes sensores’.

Para alcanzar este objetivo ha sido necesario profundizar en los lenguajes de programación que utilizan los softwares Matlab, Parallax Data Acquisition tool y Arduino IDE, y conocer las especificaciones principales y el funcionamiento del hardware y sensores empleados, explicado a lo largo del apartado 5.2. Esto ha permitido conseguir unos resultados comparables, potenciales y abundantes para analizar satisfactoriamente cada elemento utilizado.

‘Seleccionar los sensores y módulos que sean más apropiados para este trabajo’.

A través del análisis desarrollado en los diferentes sensores y módulos ha sido posible realizar la identificación y elección de los elementos que se adecuan a las necesidades de este proyecto. Estos finalmente han sido el sensor LM35 para medir la temperatura, el

ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.

sensor DHT11 para medir la humedad, el sensor APDS-9008 para medir el pulso cardíaco y el sensor MPU-6050 para medir los ciclos del sueño.

Objetivo 2. ‘Elaboración de un prototipo de bajo coste, a partir de los componentes seleccionados, para la adquisición de datos a tiempo real en Arduino’.

‘Módulo de monitorización: donde se programará el microcontrolador para recoger las variables fisiológicas a partir de los sensores escogidos en la fase del análisis’.

Esta parte está compuesta por el conexionado de todos los sensores para conformar el wearable en su forma conjunta. Tanto los diseños del hardware electrónico como del programa del software son versátiles y dejan abierta la opción de ampliar las funcionalidades de forma asequible, pudiendo reconfigurar el código de programación o incorporar accesorios adicionales.

‘Módulo de almacenamiento: para almacenar todos los datos obtenidos’.

Con este objetivo se ha dado al sistema la capacidad de almacenar los datos de los pacientes con DM1 en una tarjeta micro SD extraíble en formato de texto plano, con esto se consigue simplicidad para la descarga para la posterior manipulación de la información. Sin embargo Arduino Nano no es el mejor dispositivo para almacenarlos debido a que la librería usada para el módulo de almacenamiento hace aumentar bastante el tamaño final del programa, ya que las capacidades de memoria de este microcontrolador no son suficientes para almacenar todo el programa.

‘Módulo de interoperabilidad: se implementa y adecúa el estándar HL7 versión 2.7, para la transmisión en tiempo real de las distintas constantes vitales desde un dispositivo de adquisición de baja capacidad (Arduino)’.

Los parámetros a tiempo real correspondientes a las diferentes constantes vitales se han implementado con los mensajes formados por segmentos del estándar HL7 v2.7. De esta manera se consigue la interoperabilidad de la transmisión de las señales biomédicas.

‘Módulo de comunicación: que va a permitir establecer una comunicación Bluetooth entre la aplicación móvil y un módulo Bluetooth conectado al microcontrolador, generando y extrayendo la información en el formato y protocolo adecuados’.

Para conseguir una comunicación inalámbrica entre el wearable y la aplicación móvil se ha introducido el módulo de Bluetooth HC-06. Se ha conseguido configurar el módulo para enviar los datos a una velocidad de transmisión correcta. Gracias a este sistema se puede controlar el prototipo y ver los datos a tiempo real desde un dispositivo móvil.

‘Diseño del prototipo de bajo coste: será un estudio con una serie de planos y diagramas que darán una idea de cómo va a ser y cómo se construirá el prototipo de la pulsera wearable’.

Se ha realizado un prototipado de las piezas con OpenScad, optimizando el tamaño de las mismas, y se han impreso en el formato .stl gracias al uso de la impresora 3D. De tal manera que toda la electrónica va en su interior, consiguiendo realizar el objetivo

de crear un prototipo de pulsera inteligente de bajo coste. Además esta pulsera lleva incrustada un código QR individual para la identificación de cada paciente.

Objetivo 3. ‘Desarrollo de una aplicación Android’.

El desarrollo de una aplicación Android totalmente funcional ha permitido hacer tangible una idea, con un resultado más que satisfactorio, haciendo una mejora final del wearable. Se ha dotado al usuario de una solución que le permite controlar las constantes vitales medidas por la pulsera de manera inalámbrica. La aplicación además tiene otras funciones que permiten a las personas con DM1 mejorar su calidad de vida, como son tutoriales, lista de tareas a hacer o control de su peso.

Tal y como se comentó en el apartado 2.2 se ha introducido un código QR personal para cada paciente, de tal manera, que en caso de accidente, se le pueda identificar de manera adecuada, ya que la identificación inadecuada de pacientes es una causa importante en los problemas de complicaciones asociadas a errores en la asistencia de los pacientes. Esta sección es muy interesante ya que en caso de accidente, cualquier persona con un Smartphone es capaz de decodificar el código QR que irá implementado en la pulsera, con lo que se accedería rápidamente a su información médica a través de la aplicación desarrollada. Esta aplicación es sencilla, intuitiva y amigable. Para que sea usable se ha evitado que la información superflua y se ha minimizado todo aquello que pueda confundir, abrumar o distraer innecesariamente. Para ello, se han usado textos breves y concisos, con un menú claro y bien estructurado. Se ha mantenido una coherencia en el diseño en toda la aplicación y se han situado los contenidos donde el usuario espera que estén y hacer que las cosas funcionen como se espera que lo hagan.

7.2 Líneas futuras

Como futuras líneas de trabajo se podrían incluir las siguientes para enriquecer el modelo:

- Obtener mayor información acerca del paciente a partir de un análisis de los datos ya obtenidos.
- Conseguir que la aplicación guarde los datos monitorizados en un servidor y que esta información pueda mostrarse mediante gráficas.
- Que la aplicación sea capaz de avisar cuando se pasen unos determinados umbrales en los datos monitorizados.
- Desarrollo de una electrónica más reducida y apropiada para que el paciente la lleve sin ningún inconveniente.
- Usar dispositivos que gasten menos energía, como el Bluetooth Low Energy, que tiene un consumo muy reducido.

*ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.*

- Utilización de un microcontrolador con mejores prestaciones que Arduino para poder conseguir mejores y más eficaces resultados.

08/

8 Referencias

[1] Yuksel, M. Dogac, A (2011). Interoperability of Medical Device Information and the Clinical Applications: An HL7 RMIM based on the ISO/IEEE 11073 DIM. IEEE Transactions on Information Technology in Biomedicine, vol. 15, no. 4, pp. 557—566.

[2] Health Level 7 International. [Online]. Available at:

<http://www.hl7.org/>

[3] Williams, G. Pickup, JC (2004). The Handbook of Diabetes.

[4] World Health Organization (2016). World Health Day 2016: WHO calls for global action to halt rise in and improve care for people with diabetes. [Online]. Available at:

<http://bit.ly/1RNE0km>

[5] Rodríguez-Herrero, A (2010). Propuesta de un Algoritmo de Control en Lazo Cerrado para la Diabetes Tipo 1. Tesis Doctoral UPM. [Online]. Available at:

<http://bit.ly/29Ftm5h>

[6] PwC (2016). The Wearable Future. [Online]. Available at:

<http://pwc.to/1MrDNp2>

[7] Zajac, J. Shrestha, A. Patel, P. y Poretsky, L. The Main Events in the History of Diabetes Mellitus. [Online]. Available at:

<http://bit.ly/29XzTnU>

[8] Savona, C (2002). The History of Diabetes Mellitus. [Online]. Available at:

<http://bit.ly/29sf07w>

- [9] Bliss, M. The Discovery of Insulin. Chicago: The University of Chicago Press.
- [10] Ministerio de Salud. Guía Clínica Diabetes Mellitus Tipo 1. 1a Ed. Santiago: Minsal, 2005.
- [11] Karvonen M, Tuomilehto J, Libman I, LaPorte R. A review of the recent epidemiological data on the worldwide incidence of type 1 (insulin-dependent) diabetes mellitus. *Diabetologia* 1993 Oct; 36 (10): 883-92.
- [12] Serra, E. Guanyabens, E. y Aguilera, E (2012). Diagnóstico y manejo del paciente adulto con diabetes tipo 1 al inicio de la enfermedad. [Online]. Available at:
<http://bit.ly/29wPfm0>
- [13] Guyton, A.C (2001). Insulina, glucagón y diabetes mellitus. Tratado de Fisiología Médica.
- [14] Davis, E. Johnson, S. y Cooper, M (2013). Long-term outcome of insulin pump therapy in children with type 1 diabetes assessed in a large population-based case-control study. *Diabetes care*.
- [15] Silvio, E. Inzucchi, MD. y Sachin, K. y Majumdar, MD (2016). Terapias Actuales para el Manejo Médico de la Diabetes. [Online]. Available at:
<http://bit.ly/29u3iKI>
- [16] Vidal, M. Jansà, M. Monitorización glucémica y educación terapéutica en la diabetes. [Online]. Available at:
<http://bit.ly/29JPpY9>
- [17] Calvo Arenilla JI, Rubio López I. Iontoforesis. Aplicaciones de la Iontoforesis. Electroestimulación aplicada.
- [18] Ángel García Olaya (2004). Arquitectura Multi-Acceso de Telemedicina para el Cuidado Compartido de Pacientes Diabéticos. Tesis Doctoral. [Online]. Available at:
<http://bit.ly/29A2uBL>
- [19] Orozco López, O., Castañeda Hernández, C., Rodríguez Herrero, A., García Saéz, G. y Elena Hernando, M. (2014). Glucose-Insulin regulator for Type 1 Diabetes using high order neural networks. [Online]. Available at:

<http://bit.ly/29FsAFn>

[20] Bondia, J., Vehí, J., Palerm, C. y Herrero, P. El Páncreas Artificial: Control Automático de Infusión de Insulina en Diabetes Mellitus Tipo 1. [Online]. Available at:

<http://bit.ly/29qJsdd>

[21] Kearns, J., Merrigan, J y Simon Schork, J. Artificial Pancreas. [Online]. Available at:

<http://bit.ly/29u3qdk>

[22] FID. Atlas de la diabetes de la FID. [Online]. Available at:

<http://bit.ly/1BqjvXy>

[23] Oliva, J., Losbo, F., Molina, B. y Monereo, S. Estudio de los costes directos sanitarios de los pacientes con diabetes mellitus en España. Universidad Carlos III de Madrid.

[24] Borchers, AT., Uibo, R. y Gershwin, ME. The geoepidemiology of type 1 diabetes. Autoimmun

[25] WHO (2009). Impact of the financial crisis on health and health systems. [Online]. Available at:

<http://bit.ly/29qJ6D1>

[26] Junta de Andalucía (2009). Conserjería de Salud. Estrategia para la seguridad del paciente. [Online]. Available at:

<http://bit.ly/29ojv2S>

[27] Quees (2015). Wearable - Explicación y definición de wearable. [Online]. Available at:

<http://bit.ly/1mYdZ7M>

[28] W. Tao (2012). Gait analysis using wearable sensors. Sensors (Basel, Switzerland).

[29] Edward O. Thorp Associates. The Invention of the First Wearable Computer. [Online]. Available at:

<http://bit.ly/29v340p>

[30] Chris Davies (2013). Cracked Glass: Why wearables are the next security maelstrom. [Online]. Available at:

<http://bit.ly/29oj741>

[31] Cubeos. The history of wearable computing. [Online]. Available at:

<http://bit.ly/29LNvGH>

[32] Morgan Stanley (2014). Wearable Devices. [Online]. Available at:

<http://bit.ly/29XAZA6>

[33] KARIM (2014). Wearable Technology. KARIM Foresight Report. [Online]. Available at:

<http://bit.ly/29qvdxW>

[34] Roggen, D., Arnrich, B., Troster, G. Life Style Management using Wearable Computer. Wearable Computing Laboratory, ETH Zurich. [Online]. Available at:

<http://bit.ly/29EfdUf>

[35] Tehrani, K., Michael, A. (2014). Wearable technology and wearable devices: Everything you need to know. Wearable Devices Magazine

[36] Ayala, E. y Liu Mengxi, A. (2012). A wearable interface that helps the diagnosis of ADHD behavior based on biofeedback technology

[37] GoBe. The History and Technology Behind Healbe GoBe. [Online]. Available at:

<http://bit.ly/1R0gQFD>

[38] Barry Fass-Holmes. Withings Pulse Review. Keep a finger on your Pulse [Online]. Available at:

<http://bit.ly/29Z71LZ>

[39] Mudge S, Stott NS, Walt SE. Criterion validity of the StepWatch Activity Monitor as a measure of walking activity in patients after stroke

[40] ActiGraph. [Online]. Available at:

<http://bit.ly/29v2C2j>

[41] Empatica S.R.L (2014). Empatica E4. [Online]. Available at:

<http://bit.ly/29LNbrp>

[42] Billinghamurst, M., Starner, T. (1999). Wearable devices: new ways to manage information. Computer.

[43] UglyApps. Diabetik. [Online]. Available at:

<http://bit.ly/29FgR6C>

[44] SocialDiabetes. Diabetes APP SocialDiabetes. [Online]. Available at:

<http://bit.ly/1bOD5LD>

[45] Gordon Wong. BG Monitor Diabetes. [Online]. Available at:

<http://bit.ly/29quVzu>

[46] Tran, J. Tran, R. y White, JR (2012). Smartphone-Based Glucose Monitors and Applications in the Management of Diabetes: An Overview of 10 Salient “Apps” and a Novel Smartphone-Connected Blood Glucose Monitor. [Online]. Available at:

<http://bit.ly/29v1Sly>

[47] Serrano Rodríguez, L. (2011). Análisis técnico y su relación con el consumo energético en la construcción de pavimento flexible del paso lateral de Portoviejo. Tesis. Universidad de Manabí. [Online]. Available at:

<http://bit.ly/29w9Aqz>

[48] Martínez García, S. Gualda Gil, JA. (2006). Electrónica de potencia: componentes, topologías y equipos.

[49] Martínez González, D (2008). Sistema autónomo para la medida óptica del ritmo cardíaco. Tesis. Universidad Politécnica de Cataluña.

[50] Fritzing Fritzing. [Online]. Available at:

<http://bit.ly/1j0BcTZ>

[51] Arduino Software. [Online]. Available at:

<http://bit.ly/1R2xniM>

[52] Enríquez Herrador, R (2009). Guía de usuario de Arduino. [Online]. Available at:

<http://bit.ly/29v12W9>

[53] López Aldea, E (2016). Arduino. Guía práctica de fundamentos y simulación.

[54] Universidad de Sevilla. Programación en Matlab. [Online]. Available at:

<http://bit.ly/29w9Plr>

[55] Fluck, R. The Ultimate AndroiDAQ Guide

[57] Norma Internacional (1995). ISO/IEC 12207 Information Technology. Software Life Cycle Processes.

[58] Galo Fariño, R (2001). Modelo Espiral de un proyecto de desarrollo de software. [Online]. Available at:

<http://bit.ly/29quy7Z>

[59] El ciclo de vida del Software. Universidad de Valladolid. [Online]. Available at:

<http://bit.ly/29qjyXC>

[60] IEDGE. Ciclo de vida en el Desarrollo de Software. [Online]. Available at:

<http://bit.ly/29piiIE>

[61] Moreno Sánchez, P. A. (2014). Entornos de ayuda a la vida independiente de personas mayores sobre servicios de redes de próxima generación. Tesis doctoral. [Online]. Available at:

<http://bit.ly/29FeUqM>

[62] IEEE. 830-1998. Recommended Practice for Software Requirements Specifications. [Online]. Available at:

<http://bit.ly/29qizXb>

[63] Pierre-Alain, M (1997). Modelado de objetos con UML. Gestión 2000, Barcelona.

[64] Pérez Pérez, V. Desarrollo de un sitio web para un colegio. CEIP Montealegre de L´Eliana. [Online]. Available at:

<http://bit.ly/29v0HLu>

[65] Ruiz F, Patricia L. Ingeniería del Software. Interacciones del Sistema. [Online]. Available at:

<http://bit.ly/29LL190>

[66] Ableson F, Collins C, Sen R. Android, guía para desarrolladores. Editorial Manning.

[67] Keene JR., Kaltman SI., Kaplan HM (2002). Treatment of patients who have type 1 diabetes mellitus: physiological misconceptions and infusion pump therapy.

[68] Federación internacional de la Diabetes (2011). Plan Mundial contra la diabetes. [Online]. Available at:

<http://bit.ly/29qi1Rd>

[69] Kanderian SS, Weinzimer S, Voskanyan G, Steil GM: Identification of intraday metabolic profiles during closed-loop glucose control in individuals with type 1 diabetes.

[70] GBT-UPM (2016). Tecnologías en diabetes. [Online]. Available at:

<http://bit.ly/29wEZrn>

[71] AHTA (2010). Closed-loop insulin delivery system (‘artificial pancreas’) for management of hypoglycaemia in type 1 diabetics. [Online]. Available at:

<http://bit.ly/29vk0tk>

[72] Cruz García M, Colín Cortes O (2015). Diseño de un sistema para el registro de actividad cardíaca y temperatura corporal. [Online]. Available at:

<http://bit.ly/29Fe25k>

[73] Tu E, Twigg SM, Dufflou J, Semsarian C (2008). Causes of death in young Australians with type 1 diabetes: a review of coronial postmortem examinations.

[74] Skrivarhaug T, Bangstad HJ, Stene LC (2006). Long-term mortality in a nationwide cohort of childhood-onset type 1 diabetic patients in Norway. *Diabetologia*.

[75] Barkai L, Gurieva I, Stirban A (2012). Microvascular complications in children and adolescents with type 1 diabetes. *Diabetes Metab Heart*.

[76] Hôpital Charles LeMoyne (2010). Getting to know your diabetes. Reference guide for people with diabetes.

[77] Vive con diabetes (2013). Si suda más de lo normal, tenga mucho cuidado. [Online]. Available at:

<http://bit.ly/1bjii1mw>

[78] Leproult R, Balbo M, Spiegel K. (2010) Role of sleep duration in the regulation of glucose metabolism and appetite.

[79] Tasali E, Leproult R, Ehrmann DA, Van Cauter E (2008). Slow-wave sleep and the risk of type 1 diabetes in humans.

[80] Mallon L, Broman JE, Hetta J. High incidence of diabetes in men with sleep complaints or short sleep duration: a 12-year follow-up study of a middle-aged population. *Diabetes Care*.

[81] Meisinger C, Heier M, Loewel H, Study MKAC (2005). Sleep disturbance as a predictor of type 2 diabetes mellitus in men and women from the general population. *Diabetologia*.

[82] Ozer, J. Blemings, H (2009). *Practical Arduino: Cool Projects for Open Source Hardware* (1ª edición). Apress.

[83] Barragán, H (2011). ¿Qué es Wiring?. *DEARQ: Revista de Arquitectura de la Universidad de los Andes*.

[84] Herrero JC. Sánchez J (2015). Una mirada al mundo Arduino. [Online]. Available at:

<http://bit.ly/29FdV9Z>

[85] Forefront (2013). The Absolute Beginner's Guide to Arduino. [Online]. Available at:

<http://bit.ly/29pgVtx>

[86] Official Web Arduino (2016). Arduino UNO Genuino UNO. [Online]. Available at:

<http://bit.ly/1SB57T5>

[87] John Boxxal (2013). Arduino Workshop. [Online]. Available at:

<http://bit.ly/29EdbC7>

[88] Official Web Arduino . Arduino Nano User Manual. [Online]. Available at:

<http://bit.ly/29GnYvX>

[89] D-Robotics UK (2010). DHT11 Humidity Temperature Sensor. [Online]. Available at:

<http://bit.ly/29vZjsW>

[90] Aosong Guangzhou Electronics. DHT11 Product Manual. [Online]. Available at:

<http://bit.ly/29xuR45>

[91] Open Impulse. DHT11 Datasheet. [Online]. Available at:

<http://bit.ly/29zgL1y>

[92] Texas Instruments (2016). LM35 Datasheet. [Online]. Available at:

<http://bit.ly/1lhp1kg>

[93] Sebastiano Giannitto. Acquisizione della temperatura con LM35. [Online]. Available at:

<http://bit.ly/29rqt2p>

[94] Garret Ward (2014). Design of a Small Form-Factor Flight Control System. [Online]. Available at:

<http://bit.ly/29vZmoJ>

[95] Pastor J. y Revenga P (2013).Curso de Sensores en Plataforma Arduino. Introducción al protocolo I2C.

[96] Invensense. Mpu 6050. [Online]. Available at:

<http://bit.ly/1RbOT3x>

[97] List of Unclassified Manufacturers. MPU-6050 Datasheet. [Online]. Available at:

<http://bit.ly/29xuNRQ>

[98] Robologs (2014). Tutorial de Arduino y MPU-6050. [Online]. Available at:

<http://bit.ly/1RWELc3>

[99] Starlino (2009). Guía IMU MPU-6050 (Acelerómetro y Giroscopio. [Online]. Available at:

<http://bit.ly/1V7USna>

[100] Earn William (2014). ADXL345 Digital Accelerometer. [Online]. Available at:

<http://bit.ly/29xu0jT>

[101] Analog Devices. ADXL 345 Datasheet. [Online]. Available at:

<http://bit.ly/29Di6of>

[102] Verma, D y Bhasin, M (2014). Real time optical heart rate monitor. International Journal of Computer Science and Information Technologies.

[103] Wang, L., Lo, B. y Yang, G. (2007). Reflective photoplethysmograph earpiece sensor for ubiquitous heart rate monitoring

[104] Pulse Sensor. Technical Article Pulse Sensor Amped. [Online]. Available at:

<http://bit.ly/1DLhDqx>

[105] Avago Technologies. APDS-9008 Datasheet. [Online]. Available at:

<http://bit.ly/29Yg4gf>

[106] Chen, W., Kobayashi, T., Ichikawa, S., Takeuchi, Y., y Togawa, T. Continuous estimation of systolic blood pressure using the pulse arrival time and intermittent calibration.

[107] Maxim Integrated. MAX30100 Datasheet. Pulse Oximeter and Heart-Rate Sensor IC for Wearable Health. [Online]. Available at:

<http://bit.ly/29pwugm>

[108] Research Design Lab. Digital output temperatura and humidity sensor. [Online]. Available at:

<http://bit.ly/29v11ND>

[109] HC Information Technology. HC-06 Datasheet. [Online]. Available at:

<http://bit.ly/29pTexk>

[110] GI Electronic. Arduino Bluetooth Module HC-06. User Manual. [Online]. Available at:

<http://bit.ly/29uBS7D>

[111] Óscar Torrente Artero (2013). Arduino. Curso práctico de formación. [Online]. Available at:

<http://bit.ly/29EJ0Ty>

[112] William David Galvis (2013). Iniciando con Arduino. [Online]. Available at:

<http://bit.ly/29ud8qT>

[113] Health Level 7 Spain. [Online]. Available at:

<http://bit.ly/29xt1k0>

[114] iBit – OTIC (2013). Guía de Implementación HL7. Gestión de peticiones del Laboratorio. [Online]. Available at:

<http://bit.ly/29DgDhV>

[115] Carmen Delgado Pinillos (2014). Implementación del estándar HL7 v2.7 para electrocardiografía wireless en tiempo real. Universidad de Zaragoza. Trabajo fin de Máster.

[116] Proyecto Cornalvo. Gestión de Análisis Clínicos: Cornalvo. [Online]. Available at:

<http://bit.ly/29Yemvm>

[117] Virtuabotix. Arduino SD Card Reader For Standard SD Cards. [Online]. Available at:

<http://bit.ly/29DgHxX>

[118] Sparkfun. Serial Peripheral Interface (SPI). [Online]. Available at:

<http://bit.ly/29KGFRG>

[119] Brian W. Evans (2007). Arduino Notebook: A Beginner's Reference Written. [Online]. Available at:

<http://bit.ly/29EJ7JL>

[120] Micro for you. Comando AT Bluetooth HC-06. [Online]. Available at:

<http://bit.ly/29vkhti>

[121] Granabot. Manual breve de OpenScad. [Online]. Available at:

<http://bit.ly/29vX0pZ>

[122] Juan González. Diseño de piezas con OpenScad. [Online]. Available at:

www.iearobotics.com

[123] History of QR Code - Denso Wave Incorporated. [Online]. Available at:

<http://www.qrcode.com/en/history/>

[124] Kheder, L. S., Alvi, A. (2013). Authentication System Using Quick Response Code. International Journal of Management. IT and Engineering.

[125] Tanenbaum AS, Woodhull AS. Sistemas operativos: diseño e implementación.

[126] ETSI (2002). ETSI EG 202 116: Human Factors (HF); Guidelines for ICT products and services; "Design for All". [Online]. Available at:

<http://bit.ly/29oHKy0>

[127] ETSI (2008). ETSI EG 202 416. Human factors (HF): User Interfaces; Setup pro-

cedure design guidelines for mobile terminals and services. [Online]. Available at: :

<http://bit.ly/29zdi2Z>

[128] ISO (2007). ISO/IEC TR 19766. Guidelines for the design of icons and symbols accessible to all users, including the elderly and persons with disabilities. [Online]. Available at:

<http://bit.ly/29DfJBU>

A1/

9 Apéndice I.

9.1 Introducción

Arduino es una plataforma de electrónica abierta, centrada en la creación de prototipos, basada en hardware y software libre fáciles de usar, ya que el usuario final para el que se diseñó eran artistas, diseñadores y aficionados.

Arduino toma información del entorno a través de sus pines de entrada por los que se puede conectar una gran variedad de sensores y puede interactuar con lo que le rodea a través de actuadores tales como luces, motores, etc.

El microcontrolador existente en la placa Arduino se programa en C/C++ haciendo uso de las librerías de Arduino, aunque también es posible programar sin utilizarlas utilizando sólo las librerías estándares de avrlibc o haciéndolo directamente en ensamblador. Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de estar conectados a un ordenador, el código es grabado en la memoria flash de la placa y su ejecución comienza automáticamente al alimentar la placa. El proyecto Arduino está basado en otros proyectos de código abierto y libre tales como Wiring para las librerías del microcontrolador y Processing para el entorno de desarrollo. El propio proyecto Arduino también es de código abierto y libre, tanto para la parte de hardware como para la de software, por lo que las placas pueden ser fabricadas a mano o compradas a través de los distribuidores existentes, pudiendo descargar los ficheros de diseño con la posibilidad de modificarlos para adaptarlos a la aplicación que se desee.

Para descargarlo y continuar con el análisis, hay que dirigirse a la siguiente página e instalarlo en el ordenador:

<https://www.arduino.cc/en/Main/Software>

Una vez instalado, hay instalar los drivers adecuados para que el sistema operativo reconozca este periférico y se pueda utilizar con el proyecto. En las plataformas Linux y Mac OS X se instalarán los drivers automáticamente, por lo que no hay que realizar ninguna acción más. Para la plataforma Windows hay que realizar lo siguiente:

*ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.*

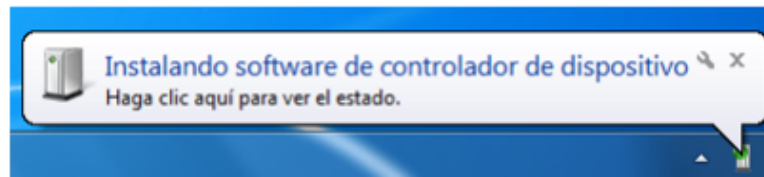


Figura 79: Apéndice I. Instalación Arduino parte 1.

Transcurridos unos segundos, nos aparecerá una nueva ventana para seleccionar la localización driver manualmente, ya que el asistente no pudo encontrarlos por si mismo. Hacemos clic en 'Buscar software de controlador en el equipo' para continuar con el proceso de instalación.

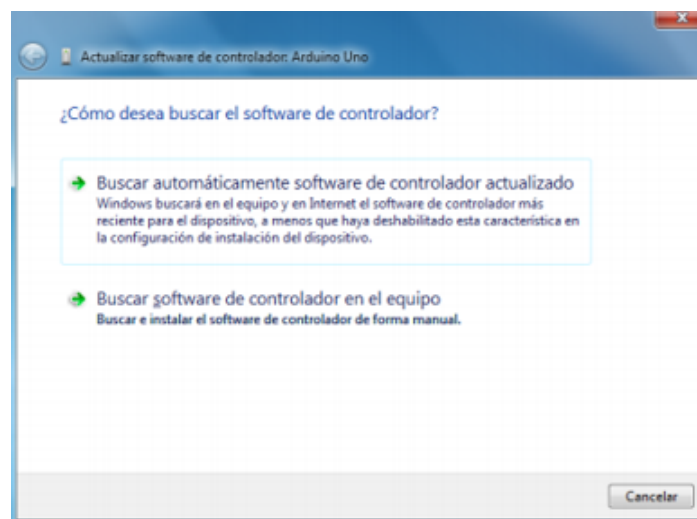


Figura 80: Apéndice I. Instalación Arduino parte 2.

Se indica la ubicación donde están los drivers de Arduino, que será la carpeta 'drivers' dentro de la carpeta del software de Arduino que nos hemos descargado anteriormente. Es importante que la opción de buscar en subcarpetas esté activada. Se da clic en 'Siguiente' para que inicie la búsqueda.

*ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.*

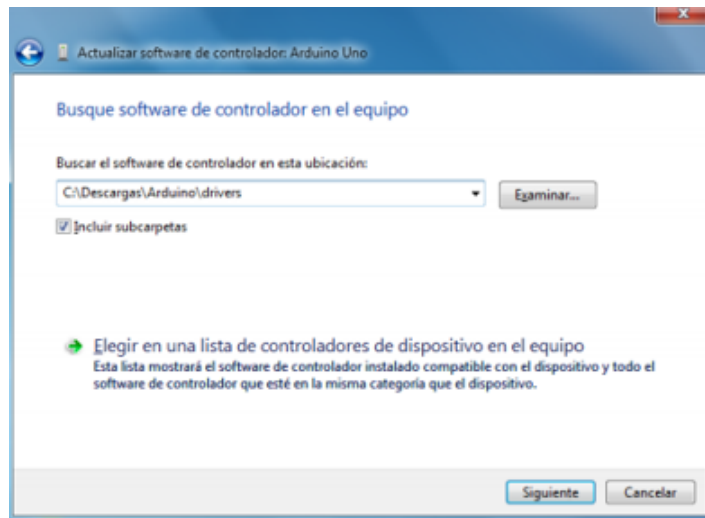


Figura 81: Apéndice I. Instalación Arduino parte 3.

El asistente comprobará si existen los drivers adecuados para el Arduino dentro de las carpetas indicadas. El puerto serie aparecerá listado como cualquier otro, siguiendo la nomenclatura de 'COMxx'.

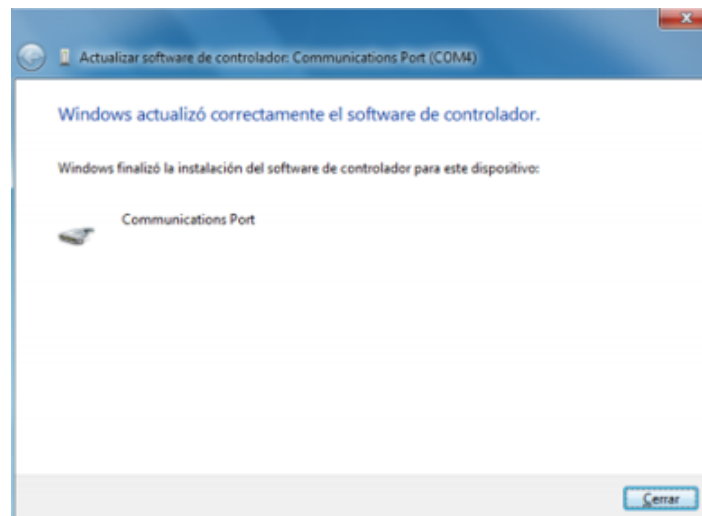


Figura 82: Apéndice I. Instalación Arduino parte 4.

9.2 Sensor DHT-11

Este sensor mide la temperatura y la humedad, para hacer el análisis solo hacen falta los siguientes materiales: placa de arduino, jumpers y el sensor DHT11. Hay que conectarlo de la siguiente manera:

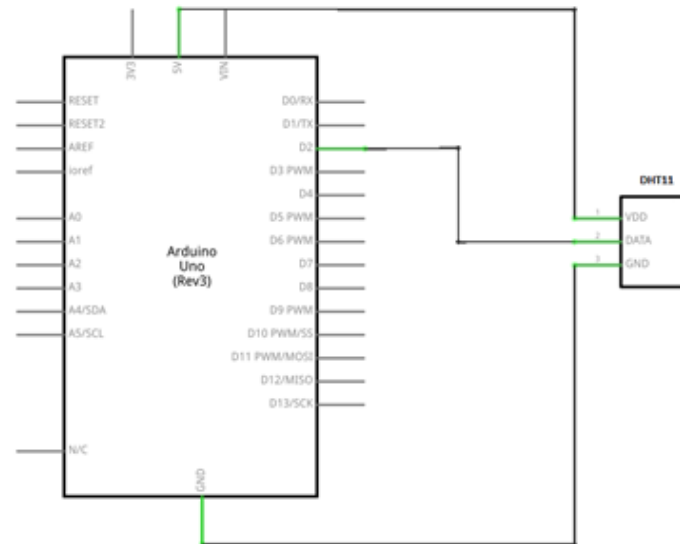


Figura 83: Apéndice I. Circuito utilizado para analizar el sensor de temperatura DHT11.

Es necesario agregar la librería DHT11 en la carpeta de librerías de Arduino, se puede descargar la librería de aquí.

El código empleado se muestra a continuación. El primero mide la temperatura y la humedad, y el segundo hace lo mismo pero con una interrupción cada 2 ms.

*ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.*

```
#include "DHT.h" //Se incluye la libreria del sensor
#define DHTPIN 2 // Pin DHT=2
#define DHTTYPE DHT11 // DHT11
DHT dht(DHTPIN, DHTTYPE);
int t,h;
void setup()
{
  Serial.begin(9600);
  dht.begin();
}
void loop() {
  t = dht.readTemperature(); //Se lee la temperatura
  h = dht.readHumidity(); //Se lee la humedad
  Serial.println(t);
  Serial.println(h);
}
```

```
#include "DHT.h" //Se incluye la libreria del sensor
#define DHTPIN 2 // Pin DHT=2
#define DHTTYPE DHT11 // DHT11
DHT dht(DHTPIN, DHTTYPE);
int t,h;
void setup()
{
  Serial.begin(9600);
  dht.begin();
  cli();
  //Interrupciones
  TCCR1A = 0x00; // Se limpian los valores por defecto
  TCCR1B = 0x00; // Timer en modo normal
  TCCR1B |= (1 << WGM12); // Se configura el Timer 1 en el modo CTC (Clear
    Timer on Compare)
  TCCR1B |= (0 << CS12); // Se fija el preescalar mediante 3 bits
  TCCR1B |= (1 << CS11); // 001=1, 010=8, 011=64, 100=256, 101=1024
  TCCR1B |= (1 << CS10);
  TIMSK1 |= (1 << OCIE1A); // Se habilita la interrupci n CTC con OCF1A
    flag en TIFR1
  OCR1A = 124; // Se fija el valor CTC , resultado de 500Hz para fI/O = 8MHz
  sei(); //activa las interrupciones
}
void loop() {
  // no se hace nada, ya que es tarea de las interrupciones
}
ISR(TIMER1_COMPA_vect) // cuando el timer hace un conteo regresivo , se
  dispara el interruptor
{
  //int val = analogRead(Analog_Input_Pin);
  //Serial.write( (val >> 2));
  t = dht.readTemperature();
  h = dht.readHumidity();
  Serial.println(t); Serial.println(h);}
}
```

9.3 Sensor LM-35

El LM35 es un sensor de temperatura digital. A diferencia de otros dispositivos como los termistores en los que la medición de temperatura se obtiene de la medición de su resistencia eléctrica, el LM35 es un integrado con su propio circuito de control, que proporciona una salida de voltaje proporcional a la temperatura. Hay que conectionarlo de la siguiente manera:

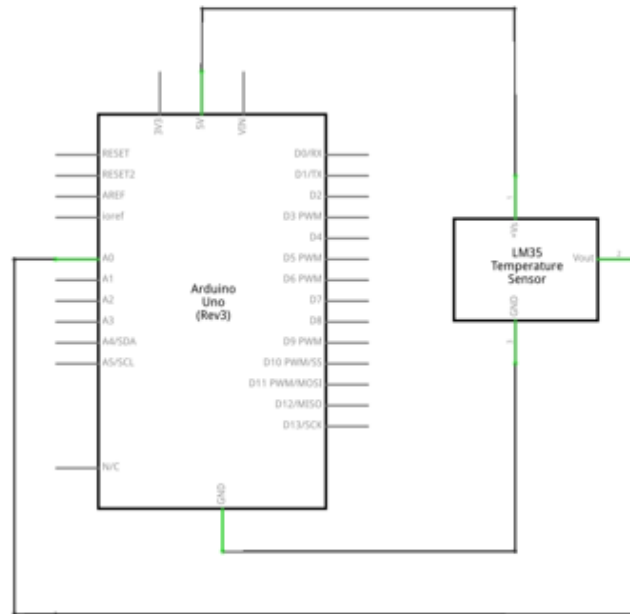


Figura 84: Apéndice I. Circuito utilizado para analizar el sensor de temperatura LM35.

El código empleado se muestra a continuación. El primero mide la temperatura, y el segundo hace lo mismo pero con una interrupción cada 2 ms.

*ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.*

```
const float voltage_usado = 5.0;           // El voltaje usado
const unsigned int velocidad = 9600;
const int Analog_Input_Pin = 0; //Pin usado, 0 anal gico
void setup()
{
  Serial.begin(velocidad); // Se envian los datos
}
void loop() {
  Serial.print(obtener_temperatura());
  Serial.println(" C");
}
{
const float obtener_temperatura() {
const int sensor_voltage = analogRead(Analog_Input_Pin);
const float voltage = sensor_voltage * voltage_usado / 1024;
return (voltage * 100);}
```

```
const float voltage_usado = 5.0;           // El voltaje usado
const unsigned int velocidad = 9600;
const int Analog_Input_Pin = 0;
void setup()
{Serial.begin(velocidad); // Se envian los datos
cli();
//Interrupci n
TCCR1A = 0x00; // Se limpian los valores por defecto
TCCR1B = 0x00; // Timer en modo normal
TCCR1B |= (1 << WGM12); // Se configura el Timer 1 en el modo CTC (Clear
Timer on Compare)
TCCR1B |= (0 << CS12); // Se fija el preescalar mediante 3 bits
TCCR1B |= (1 << CS11); // 001=1, 010=8, 011=64, 100=256, 101=1024
TCCR1B |= (1 << CS10);
TIMSK1 |= (1 << OCIE1A); // Se habilita la interrupci n CTC con OCF1A
flag en TIFR1
OCR1A = 124; // Se fija el valor CTC , resultado de 500Hz para fI/O = 8MHz
sei(); //activa las interrupciones
}
void loop() {
// no se hace nada, ya que es tarea de las interrupciones
}
ISR(TIMER1_COMPA_vect) // cuando el timer hace un conteo regresivo , se
dispara el interruptor
{int val = analogRead(Analog_Input_Pin);
//Serial.write( (val >> 2));
Serial.print(obtener_temperatura());
Serial.println(" C");
}
const float obtener_temperatura() {
const int sensor_voltage = analogRead(Analog_Input_Pin);
const float voltage = sensor_voltage * voltage_usado / 1024;
return (voltage * 100);}
```

9.4 Sensor MPU-6050

Este sensor permite medir la velocidad angular en los tres ejes, y un acelerómetro de 3 ejes con el que medimos los componentes X, Y y Z de la aceleración. Primero hay que descargar las librerías de este link de aquí. En este link, se encuentran las librerías 'I2Cdev' y 'MPU6050', las cuales hay que copiar en la carpeta Libraries dentro de la carpeta Arduino. Con esto ya se tiene todo el software necesario en Arduino para crear el sketch. El circuito y código empleados se muestran a continuación.

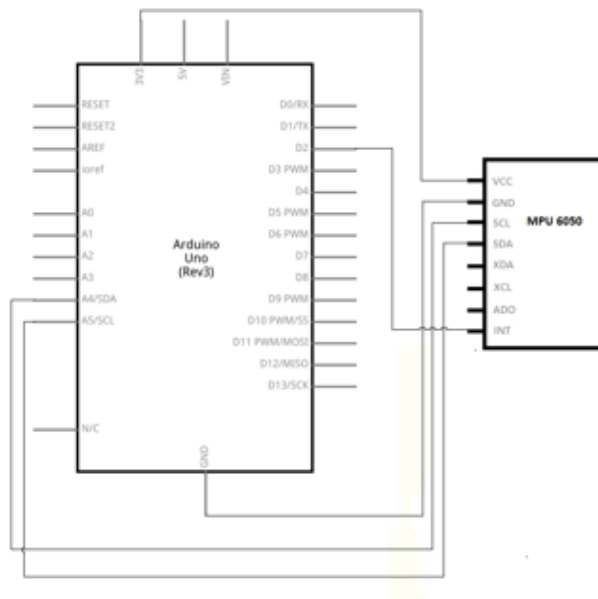


Figura 85: Apéndice I. Circuito utilizado para analizar el sensor MPU-6050.

```
//Se introducen las librerías del sensor MPU6050, de I2Cdev y de Wire
#include "I2Cdev.h"
#include "MPU6050_6Axis_MotionApps20.h"
#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
    #include "Wire.h"
#endif
MPU6050 mpu;

#define GRADOS
int row = 0;
// MPU control/status vars
bool dmpReady = false; // se pone verdadero si el DMP (Digital Motion
    Processor) se ha iniciado correctamente
uint8_t mpuIntStatus; // contiene el actual byte de la interrupci n del
    MPU
uint8_t devStatus; // devuelve el estado despu s de cadd operaci n
    (0 = correcto , !0 = error)
uint16_t packetSize; // establece el tama o del paquete de datos (
    default is 42 bytes)
uint16_t fifoCount; // cuenta todos los bytes del FIFO
uint8_t fifoBuffer[64]; // almacenamiento en el FIFO

// Acelerometr a/Gir scopo
Quaternion q; // [w, x, y, z] Cuaternio
VectorInt16 aa; // [x, y, z] Medidas de acelerometr a
VectorInt16 aaReal; // [x, y, z] gravity-free accel sensor
    measurements
VectorInt16 aaWorld; // [x, y, z] world-frame accel sensor
    measurements
VectorFloat gravity; // [x, y, z] Vector gravedad
float euler[3]; // [psi, theta, phi] Angulos de Euler
float ypr[3]; // [yaw, pitch, roll] velocidad de giro/ yaw/
    pitch/roll container and gravity vector
uint8_t teapotPacket[14] = { '$', 0x02, 0,0, 0,0, 0,0, 0,0, 0x00, 0x00, '\r
    ', '\n' };

//Interrupci n

volatile bool mpuInterrupt = false;
void dmpDataReady() {
    mpuInterrupt = true;
}

//Setup

void setup() {
    // Union bus I2C
    #if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
        Wire.begin();
        TWBR = 24; // 400kHz I2C clock (200kHz en la CPU son 8MHz)
    #elif I2CDEV_IMPLEMENTATION == I2CDEV_BUILTIN_FASTWIRE
        Fastwire::setup(400, true);
    #endif
}
```


ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.

```
// Se inicia la comunicaci n en serie. Se eligen 38400
Serial.begin(38400);
while (!Serial);
mpu.initialize();
Serial.println(F("Inicializacion DMP..."));
devStatus = mpu.dmpInitialize();
// Se ponen los offsets aqui, escalados a la minima sensibilidad
mpu.setXGyroOffset(220);
mpu.setYGyroOffset(76);
mpu.setZGyroOffset(-85);
mpu.setZAccelOffset(1788);
// Se comprueba que funciona (devuelve 0 si es verdad )
if (devStatus == 0) {
    // Se habilita el DMP al estar ya preparado
    mpu.setDMPEnabled(true);
    // Se habilitan las interrupciones
    attachInterrupt(0, dmpDataReady, RISING);
    mpuIntStatus = mpu.getIntStatus();
    dmpReady = true;
//Desde esta secci n se env an los datos al Excel mediante un macro,
// Parallax Data Aquisition Tool
//Se configura la obtenci n de datos con Parallax_DAQ
    Serial.println("CLEARDATA"); //Se limpian los datos
//En Excel se mostrar el tiempo, los grados en el eje x, y, z
    Serial.println("LABEL,TIME,x,y,z"); //Siempre se escribe LABEL, para
    // que Excel sepa que los proximos parametros son columnas
// se obtiene el tama o esperado del paquete
    // get expected DMP packet size for later comparison
    packetSize = mpu.dmpGetFIFOPacketSize();
} else {
// ERROR!
    // 1 = memoria ha fallado.
    // 2 = DMP ha fallado.
    // (si hay fallo, suele ser de tipo 1)
//    Serial.print(F("DMP ha fallado (code ") );
    Serial.print(devStatus);
    Serial.println(F(")"));
}
}

void loop() {
    // Si el programa falla, no se hace nada
    if (!dmpReady) return;
    // Se espera a la interrupcion
    while (!mpuInterrupt && fifoCount < packetSize) {
    }
    // Se resetea el flag de la interrupcion y se inicializa el byte
    // inicial
    mpuInterrupt = false;
    mpuIntStatus = mpu.getIntStatus();

    // Se obtiene la cuenta FIFO
```

*ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.*

```
fifoCount = mpu.getFIFOCount();
if ((mpuIntStatus & 0x10) || fifoCount == 1024) {
    mpu.resetFIFO();
} else if (mpuIntStatus & 0x02) {
    // Se espera al correcto tamaño de la longitud de los datos
    while (fifoCount < packetSize) fifoCount = mpu.getFIFOCount();
// Se lee el paquete FIFO
mpu.getFIFOBytes(fifoBuffer, packetSize);
fifoCount -= packetSize;
#ifdef GRADOS
    // Se muestran los angulos de Euler en grados
    mpu.dmpGetQuaternion(&q, fifoBuffer);
    mpu.dmpGetGravity(&gravity, &q);
    mpu.dmpGetYawPitchRoll(ypr, &q, &gravity);
    Serial.print(F("DATA,TIME, "));
    Serial.print(ypr[0] * 180/M_PI);
    Serial.print(F(" , "));
    Serial.print(ypr[1] * 180/M_PI);
    Serial.print(F(" , "));
    Serial.println(ypr[2] * 180/M_PI);
    //Se mandan a PRL-CAF
    row++;
    //Sobreescritura de los datos al llegar al llegar a 1000
    if (row > 1000)
    {
        row=0;
        Serial.println("ROW,SET,2");
    }
    delay(100);
    //Fin de la escritura de datos con Parallax_Daq
#endif
}}
```

9.5 Sensor ADXL-345

Lo primero que hay que hacer es descargarse la librería Wire.h de aquí. Para programar I2C hay que tener en cuenta las siguientes funciones:

- Wire.begin(): inicia el protocolo I2C.
- Wire.write(X): transmite la información X.
- Wire.endTransmission(): termina la transmisión.

El circuito y código empleado se encuentra a continuación.

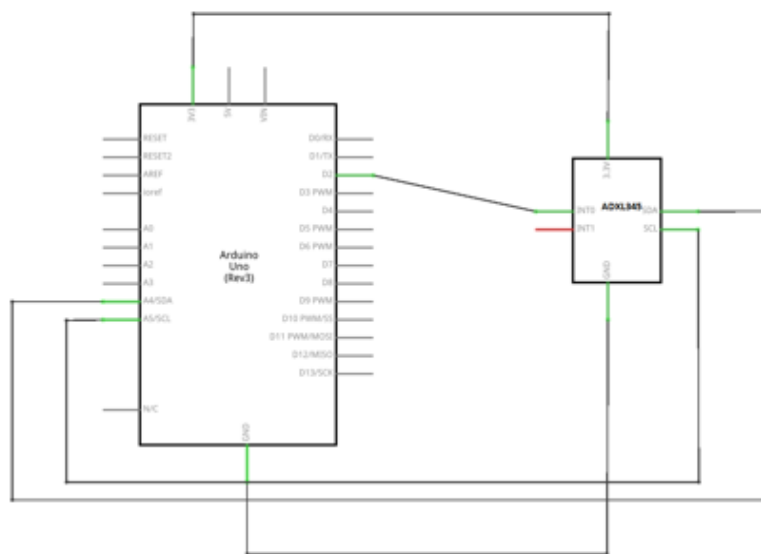


Figura 86: Apéndice I. Circuito utilizado para analizar el sensor ADXL-345.

*ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.*

```
byte values[6] ;
char output[512];
void setup(){
Wire.begin();
Serial.begin(9600);
Wire.beginTransmission(accel_module);
Wire.write(0x2D);
Wire.write(0);
Wire.endTransmission();
Wire.beginTransmission(accel_module);
Wire.write(0x2D);
Wire.write(16);
Wire.endTransmission();
Wire.beginTransmission(accel_module);
Wire.write(0x2D);
Wire.write(8);
Wire.endTransmission();
}
void loop(){
int xyzregister = 0x32;
int x, y, z;

Wire.beginTransmission(accel_module);
Wire.write(xyzregister);
Wire.endTransmission();
Wire.beginTransmission(accel_module);
Wire.requestFrom(accel_module, 6);
int i = 0;
while(Wire.available()){
values[i] = Wire.read();
i++;
}
Wire.endTransmission();
x = (((int) values[1]) << 8) | values[0];
y = (((int) values[3]) << 8) | values[2];
z = (((int) values[5]) << 8) | values[4];

sprintf(output, "%d %d %d", x, y, z);
Serial.print(output); Serial.write(10); delay(1000);}
```

9.6 Sensor APDS-9008

Este sensor sirve para medir el pulso cardíaco a través de un sensor óptico. Lo primero que hay que hacer es cargar el programa en Arduino, este se encuentra disponible en la siguiente página del apéndice. A continuación hay que montar el siguiente circuito:

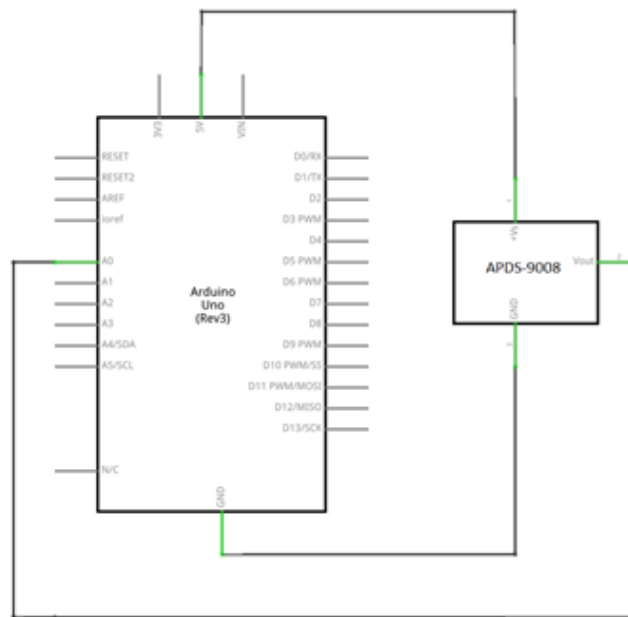


Figura 87: Apéndice I. Circuito utilizado para analizar el sensor APDS-9008.

Hay que limpiar bien el sensor para conseguir captar bien el pulso, ya que si no está limpio puede provocar ruido en la señal. Como se indica en el circuito, hay que conectar el sensor en el pin analógico 0, aunque se puede configurar en el pin analógico que se quiera. El sensor se puede alimentar tanto a 5 (V) como a 3.3 (V). Realizado esto, ya se puede utilizar el sensor en la parte del cuerpo donde se capte mejor el pulso cardíaco, para el análisis se ha realizado en el dedo. Para visualizar bien la señal, se ha utilizado los programas disponibles aquí y aquí.

ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.

```
// Variables
int pulsePin = 1;           // El sensor se conecta en ANALOG1
int fadeRate = 0;          // Fallo

//Variables volatiles para la parte del interruptor
volatile int BPM;
volatile int Signal;
volatile int IBI = 600;     // Tiempo entre pulso y pulso
volatile boolean Pulse = false; // Verdadero cuando el pulso es
                             detectado
volatile boolean QS = false; // Se convierte en verdadero cuando
                             Arduino detecta un pulso

static boolean serialVisual = true; // Monitor ASCII
void setup() {

    Serial.begin(115200);
    interruptSetup();
}
void loop() {

    serialOutput() ;

    if (QS == true){

        serialOutputWhenBeatHappens();
        QS = false;
    }

    delay(20); }
```

9.7 MAX-30100

El sensor MAX-30100 sirve para medir el pulso cardíaco. Lo primero que hay que hacer es descargar e instalar la siguiente librería de aquí.

A continuación, hay que montar el siguiente circuito alimentado a 3.3 (V) y luego usar el código que se proporciona a continuación.

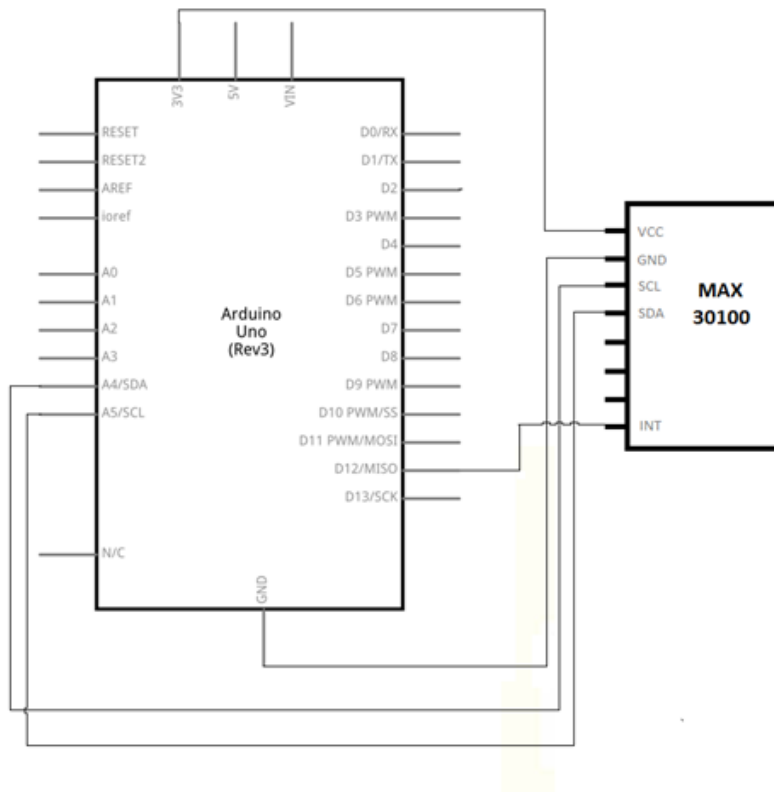


Figura 88: Apéndice I. Circuito utilizado para analizar el sensor MAX-30100.

ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.

```
#include <Wire.h>
#include "Adafruit_MAX30100.h"

Adafruit_MAX30100 pulse = Adafruit_MAX30100();
#define MAX30100_IRQPIN 12 //Se define el pin
#define SECONDS_OF_SAMPLES (5) //Muestras por segundo
#define SAMPLE_RATE (50) //Frecuencia de muestreo
#define MAX_SAMPLES (SECONDS_OF_SAMPLES * SAMPLE_RATE) //Numero
//maximo de muestras

uint32_t fifo_buffer[16];
uint32_t databufferptr = 0;
uint16_t databuffer_ir[MAX_SAMPLES];
uint16_t databuffer_red[MAX_SAMPLES];

void setup() {
  while (!Serial);
  delay(100);
  Serial.begin(115200);
  Serial.println("An lisis MAX 30100");

  if (!pulse.begin()) { //Se comprueba que se encuentra el sensor
    Serial.println("Fallo de conexi n , no se ha encontrado el sensor");
    while (1);
  }
  Serial.println("Sensor encontrado"); //Si se ha encontrado , se continua
  //el programa
  pinMode(MAX30100_IRQPIN, INPUT);

  // Se establecen los valores de oximetria , SpO2, para gererar la se al
  pulse.setSpO2SampleRate(MAX30100_SPO2SR_50HZ);

  max30100_spo2_samplerate_t rate = pulse.getSpO2SampleRate();
  uint16_t ratehz = 0;
  switch (rate) {
    case MAX30100_SPO2SR_50HZ: ratehz = 50; break;
    case MAX30100_SPO2SR_100HZ: ratehz = 100; break;
    case MAX30100_SPO2SR_167HZ: ratehz = 167; break;
    case MAX30100_SPO2SR_200HZ: ratehz = 200; break;
    case MAX30100_SPO2SR_400HZ: ratehz = 400; break;
    case MAX30100_SPO2SR_600HZ: ratehz = 600; break;
    case MAX30100_SPO2SR_800HZ: ratehz = 800; break;
    case MAX30100_SPO2SR_1000HZ: ratehz = 1000; break;
  }
  Serial.print("SpO2 ratio = "); Serial.print(ratehz); Serial.println(" Hz"
);

  //Se establecen los valores del led
  pulse.setLEDpulseWidth(MAX30100_LEDPW_1600US);

  max30100_led_pulsewidth_t width = pulse.getLEDpulseWidth();
  uint16_t pulsewidth = 0;
```


*ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.*

```
switch (width) {
  case MAX30100_LEDPW_200US: pulsewidth = 200; break;
  case MAX30100_LEDPW_400US: pulsewidth = 400; break;
  case MAX30100_LEDPW_800US: pulsewidth = 800; break;
  case MAX30100_LEDPW_1600US: pulsewidth = 1600; break;
}
Serial.print("LED longitud = "); Serial.print(pulsewidth); Serial.println
(" uS");

//Se establecen las intensidades de los leds
pulse.setRedLEDcurrent(MAX30100_LEDCURRENT_30_6MA);
pulse.setIRLEDcurrent(MAX30100_LEDCURRENT_11MA);
max30100_led_current_t red = pulse.getRedLEDcurrent();
max30100_led_current_t ir = pulse.getIRLEDcurrent();

float ledcurrent = 0;
switch (red) {
  case MAX30100_LEDCURRENT_0MA: ledcurrent = 0; break;
  case MAX30100_LEDCURRENT_4_4MA: ledcurrent = 4.4; break;
  case MAX30100_LEDCURRENT_7_6MA: ledcurrent = 7.6; break;
  case MAX30100_LEDCURRENT_11MA: ledcurrent = 11; break;
  case MAX30100_LEDCURRENT_14_2MA: ledcurrent = 14.2; break;
  case MAX30100_LEDCURRENT_17_4MA: ledcurrent = 17.4; break;
  case MAX30100_LEDCURRENT_20_8MA: ledcurrent = 20.8; break;
  case MAX30100_LEDCURRENT_24MA: ledcurrent = 24; break;
  case MAX30100_LEDCURRENT_27_1MA: ledcurrent = 27.1; break;
  case MAX30100_LEDCURRENT_30_6MA: ledcurrent = 30.6; break;
  case MAX30100_LEDCURRENT_33_8MA: ledcurrent = 33.8; break;
  case MAX30100_LEDCURRENT_37MA: ledcurrent = 37; break;
  case MAX30100_LEDCURRENT_40_2MA: ledcurrent = 40.2; break;
  case MAX30100_LEDCURRENT_43_6MA: ledcurrent = 43.6; break;
  case MAX30100_LEDCURRENT_46_8MA: ledcurrent = 46.8; break;
  case MAX30100_LEDCURRENT_50MA: ledcurrent = 50; break;
}
Serial.print("Corriente led rojo = "); Serial.print(ledcurrent); Serial.
println(" mA");

switch (ir) {
  case MAX30100_LEDCURRENT_0MA: ledcurrent = 0; break;
  case MAX30100_LEDCURRENT_4_4MA: ledcurrent = 4.4; break;
  case MAX30100_LEDCURRENT_7_6MA: ledcurrent = 7.6; break;
  case MAX30100_LEDCURRENT_11MA: ledcurrent = 11; break;
  case MAX30100_LEDCURRENT_14_2MA: ledcurrent = 14.2; break;
  case MAX30100_LEDCURRENT_17_4MA: ledcurrent = 17.4; break;
  case MAX30100_LEDCURRENT_20_8MA: ledcurrent = 20.8; break;
  case MAX30100_LEDCURRENT_24MA: ledcurrent = 24; break;
  case MAX30100_LEDCURRENT_27_1MA: ledcurrent = 27.1; break;
  case MAX30100_LEDCURRENT_30_6MA: ledcurrent = 30.6; break;
  case MAX30100_LEDCURRENT_33_8MA: ledcurrent = 33.8; break;
  case MAX30100_LEDCURRENT_37MA: ledcurrent = 37; break;
  case MAX30100_LEDCURRENT_40_2MA: ledcurrent = 40.2; break;
  case MAX30100_LEDCURRENT_43_6MA: ledcurrent = 43.6; break;
  case MAX30100_LEDCURRENT_46_8MA: ledcurrent = 46.8; break;
```

ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.

```
    case MAX30100_LEDCURRENT_50MA:    ledcurrent = 50; break;
}
Serial.print("Corriente led IR = "); Serial.print(ledcurrent); Serial.
println(" mA");

pulse.startRead();

}
void loop() {
    databufferptr = 0;
    while (databufferptr < MAX_SAMPLES) {
        if (digitalRead(MAX30100_IRQPIN)) continue;
        // some data ready!
        uint32_t fifodata = pulse.readRegister32(MAX30100_FIFO_DATA_REG);
        databuffer_ir[databufferptr] = fifodata >> 16;
        databuffer_red[databufferptr] = fifodata & 0xFFFF;
        databufferptr++;
    }
    for (uint16_t i=0; i<MAX_SAMPLES; i++) {
        Serial.println(databuffer_ir[i]);
    }
    return;

    databufferptr = 0;
    while ((databufferptr+16) < MAX_SAMPLES) {
        while (digitalRead(MAX30100_IRQPIN)); // wait until IRQ goes low!
        // read 8 bytes of fifo
        pulse.readFIFO(fifo_buffer, 8);
        for (uint8_t i=0; i<8; i++) {
            databuffer_ir[databufferptr+i] = fifo_buffer[i] >> 16;
            databuffer_red[databufferptr+i] = fifo_buffer[i] & 0xFFFF;
        }
        databufferptr += 8;
    }

    for (uint16_t i=0; i<MAX_SAMPLES; i++) {
        Serial.println(databuffer_red[i]);
    }
    delay(1000);
}
```

9.8 MATLAB y PLX-DAQ

Con Parallax Data Acquisition tool (PLX-DAQ) se pueden exportar todos los datos del monitor serie de Arduino a Excel, para ello se tienen que añadir estas líneas al código de Arduino y descarga el programa de aquí. Este programa usa los siguientes comandos:

- LABEL: que se usa para indicar los títulos de la columnas en Excel.
- DATE, TIME : envía los datos que se encuentran en el microcontrolador incorporando la hora actual.
- ROW, SET, k, Toma el de la última fila anotada en Excel.

```
void setup() {  
  //Desde esta seccion se envian los datos al Excel mediante un macro,  
  Parallax Data Aquisition Tool  
  //Se configura la obtencion de datos con Parallax_DAQ  
  Serial.println("CLEARDATA"); //Se limpian los datos  
  //En Excel se mostrar el tiempo, los grados en el eje x, y, z  
  Serial.println("LABEL,TIME,x,y,z"); //Siempre se escribe LABEL, para  
  que Excel sepa que los pr ximos par metros son columnas  
}  
void loop() {  
  //Se mandan a PRL-CAF  
  row++;  
  //Sobreescritura de los datos al llegar al llegar a 1000  
  if (row > 1000)  
  {  
    row=0;  
    Serial.println("ROW,SET,2");  
  }  
  delay(100);  
  //Fin de la escritura de datos con Parallax_DAQ  
#endif
```

En cuanto a Matlab, hay que descargarse primero un programa para Arduino, este programa se encuentra en un paquete gratuito de mathworks, que se encuentra en la siguiente página:

<http://es.mathworks.com/hardware-support/arduino-matlab.html>

Una vez instalado, se utiliza el siguiente código para la adquisición de datos:

```
clear all;  
close all;  
s1 = serial('COM4'); %define puerto serie  
s1.BaudRate=9600; %define velocidad  
display('Bienvenido a su programa de toma de medidas v1.0')  
puntos = input('Cuantos puntos de medida desea tomar?');
```

ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.

```
periodo = input('Cada cuantos segundos desea que se tome su medida?');
double matriz(puntos);
%open serial port
fopen(s1);
clear data;
for i= 1:puntos                                %adquisicion de los puntos
    data=fscanf(s1);                            %lectura del sensor
    y(i) = str2double(data);
    plot(i,y(i),'+-r')
    title('Temperatura');
    xlabel({'Ultimo dato tomado a las: '(datestr(now,13))});
    ylabel({'Temperatura en C '};y(i)});
    drawnow;
    grid on
    hold on
    matriz(i)=y(i);
    pause(periodo);
end
fclose(s1);
display('El programa ha terminado de tomar las medidas.')
guardar=input('Desea guardar esta sesi n?(si=1/no=2)');
if (guardar==1)
    display('Al guardar se crear un archivo en su carpeta MATLAB,')
    display('recuerde tener cerrado el archivo donde se va a guardar su')
    display('toma de datos, y no sobrescribir hojas del libro de excel')
    file=input('Cual es el nombre del archivo: ','s');
    hoja=input('Hoja en la que deseas trabajar: ','s');
    desc=[{'Tiempo(s) '},{ 'Centigrados '},{ 'Farenheit '}];
    [estado]=xlswrite(file ,desc ,hoja , 'A1');
    if estado==0
        disp('La creacion del archivo ha fallado. ');
    end
    muestras=[1:puntos]';
    [estado]=xlswrite(file ,(muestras*periodo),hoja , 'A2');
    if estado==0
        disp('La creacion del archivo ha fallado. ');
    end
    valor=matriz(1:puntos)';
    [estado]=xlswrite(file ,valor ,hoja , 'B2');
    if estado==0
        disp('La creacion del archivo ha fallado. ');
    end
    temp=(1.8).*valor+32;
    [estado]=xlswrite(file ,temp ,hoja , 'C2');
    if estado==0
        disp('La creacion del archivo ha fallado. ');
    end
end
end
%cierre serial port
```

A2/

10 Apéndice II.

10.1 Introducción

En este apéndice se ofrece un pequeño manual de usuario para el desarrollo de un wearable formado por diferentes sensores y módulos. Roboband es una pulsera inteligente basada en Arduino. Gracias a su comunicación tanto en serie como inalámbrica es capaz de permitir monitorizar los datos de diferentes constantes vitales: humedad corporal, temperatura, ciclos del sueño y pulso cardíaco. También se podrá ampliar con relativa facilidad.

El entorno de desarrollo Arduino (IDE) se puede obtener de aquí:

<http://arduino.cc/en/Main/Software/>

Todas las versiones del Código de Roboband se pueden obtener de aquí:

<https://github.com/MGoodche/Smart-Band-Code-Roboband>

10.1.1 Características principales

Las características de las que consta la programación de esta pulsera son las siguientes:

- Monitorización de la temperatura.
- Monitorización de la humedad.
- Monitorización del pulso cardíaco.
- Monitorización de los datos del acelerómetro y giróscopo.
- Comunicación inalámbrica mediante Bluetooth para transmitir los datos a una aplicación móvil.
- Interoperabilidad de los datos en el estándar HL7 v2.7
- Opcionalmente, almacenamiento de los datos en una tarjeta micro SD.

10.1.2 Lista de componentes

Los dispositivos finales empleados se muestran a continuación, además se ha proporcionado un link con lugares de confianza donde se ha hecho su compra. También se pueden comprar en otro lugar, para ello sigue cualquier enlace proporcionado, obtén las especificaciones de este sitio, y trata de encontrarlo en otro lugar.

- Sensor de temperatura LM35: <http://bit.ly/2951Pv2>
- Sensor de humedad DHT11: <http://bit.ly/28ZJ41D>
- Sensor de acelerometría y giróscopo MPU-6050: <http://bit.ly/29a1DYD>
- Sensor de pulso Pulse Sensor: <http://bit.ly/292hNDW>
- Módulo Bluetooth HC-06: <http://bit.ly/2927ZWb>
- Arduino Nano: <http://bit.ly/2974GAw>
- Módulo SD Card: <http://bit.ly/29a1HaF>

10.2 Montaje Roboband

Arduino Nano está formado por diferentes pines como se puede ver en la siguiente figura:



Figura 89: Apéndice II. Microcontrolador Arduino Nano [30].

En la placa, debería ser capaz de ver que cada pin está claramente identificado con un número o una breve descripción de lo que el pin es. Como se puede observar tiene 14 Entradas/Salidas digitales (6 de las cuales pueden utilizarse como salidas PWM) y 8 entradas analógicas. Funciona a 16MHz con un cristal externo al microcontrolador montado en el PCB. A continuación se muestra una tabla con la asignación de pines para el wearable Roboband:

Sensor de temperatura LM 35	PIN A3
Sensor de humedad	PIN 3
Bluetooth TX	PIN 10
Bluetooth RX	PIN 11
Sensor de Pulso Pulse Sensor	PIN A1
MPU-6050 INT	PIN 2
MPU-6050 SCL	PIN A4
MPU-6050 SDA	PIN A5

Tabla 31: Relación final de pines de Roboband.

El dispositivo wearable es capaz de medir diferentes constantes vitales como se ha mencionado anteriormente. Para conseguir este propósito hay que instalar diferentes librerías. Las librerías necesarias que hay que incluir en la carpeta libraries de Arduino para el correcto funcionamiento del wearable son las siguientes:

- SoftwareSerial.h
- I2Cdev.h
- MPU6050-6Axis-MotionApps20.h
- Wire.h
- DHT.h

*ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.*

- Timer.h

El circuito final es el siguiente:

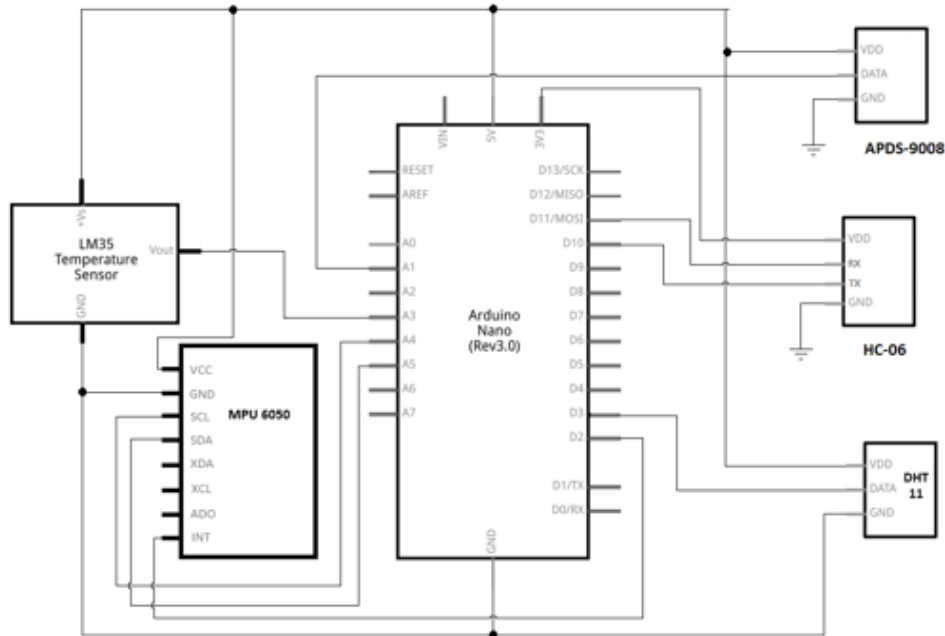


Figura 90: Apéndice II. Circuito final del montaje.

Se ha hecho un menú por el serial para poder ver los resultados sin la aplicación móvil. Debería salir algo así:

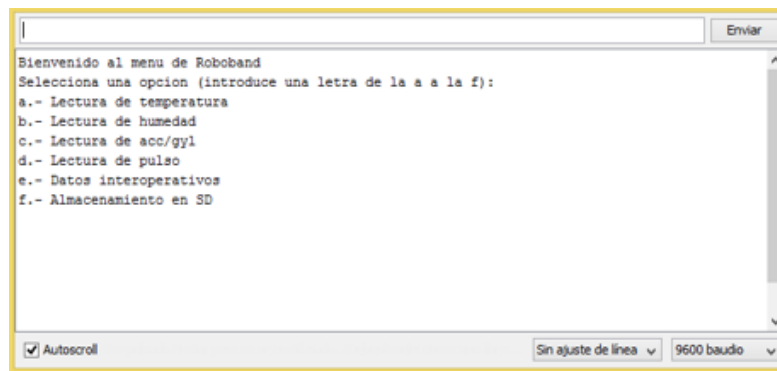


Figura 91: Apéndice II. Programa Roboband por serial de Arduino.

ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.

```
#define DHTPIN 4
#include <SoftwareSerial.h>
#include "I2Cdev.h" //Se carga la libreria de las comunicaciones con el
    sensor de acelerometria y giroscopo
#include "MPU6050_6Axis_MotionApps20.h" //Se carga la libreria del sensor
    MPU6050
#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
    #include "Wire.h"
#endif
#include "DHT.h"
#include <Timer.h>
// #include <SD.h> //Se carga la libreria del almacenamiento SD
// #include <SPI.h> //Se carga la libreria SPI
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
const int sensorPin= A3;

//Bluetooth
SoftwareSerial BT(10,11); //Se conecta el bluetooth HC-06.
//Variables MPU-6050
MPU6050 mpu;
bool dmpReady = false; // se pone verdadero si el DMP (Digital Motion
    Processor) se ha iniciado correctamente
uint8_t mpuIntStatus; // contiene el actual byte de la interrupcion del
    MPU
uint8_t devStatus; // devuelve el estado despues de cada operacion
    (0 = correcto, != 0 = error)
uint16_t packetSize; // establece el tamaño del paquete de datos (
    default is 42 bytes)
uint16_t fifoCount; // cuenta todos los bytes del FIFO
uint8_t fifoBuffer[64]; // almacenamiento en el FIFO
Quaternion q; // [w, x, y, z] Cuaternio
VectorInt16 aa; // [x, y, z] Medidas de acelerometria
VectorInt16 aaReal; // [x, y, z] gravity-free accel sensor
    measurements
VectorInt16 aaWorld; // [x, y, z] world-frame accel sensor
    measurements
VectorFloat gravity; // [x, y, z] Vector gravedad
float euler[3]; // [psi, theta, phi] Angulos de Euler
float ypr[3]; // [yaw, pitch, roll] velocidad de giro

uint8_t teapotPacket[14] = { '$', 0x02, 0,0, 0,0, 0,0, 0,0, 0x00, 0x00, '\r',
    '\n' };
// Interrupcion MPU
volatile bool mpuInterrupt = false; // Interrupcion activada,
    indicacion
void dmpDataReady() {
    mpuInterrupt = true;
}
//Variables almacenamiento SD
//File DatoSensor; //Lugar donde se escribieran los datos
//int sdpin = 11; // Pin para introducir la SD
```

ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.

```
//PULSO
// Variables
int pulsePin = 1;           // El sensor se conecta en ANALOG1
int fadeRate = 0;           // Fallo

//Variables volatiles para la parte del interruptor
volatile int BPM;
volatile int Signal;
volatile int IBI = 600;      // Tiempo entre pulso y pulso
volatile boolean Pulse = false; // Verdadero cuando el pulso es
                               detectado
volatile boolean QS = false; // Se convierte en verdadero cuando
                               Arduino detecta un pulso

static boolean serialVisual = true; // Monitor ASCII
char rec;

Timer t;

void setup()
{
  Serial.begin(9600);
  //Interrupci n 2ms
  interruptSetup();
  //DHT11 INICIALIZACI N
  dht.begin();

  //Almacenamiento SD
  //pinMode(10, OUTPUT); //Se declara la salida 10 para la SD
  //SD.begin(4); //Se inicializa la lectura de datos

  //MPU INICIALIZACI N
  #if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
    Wire.begin();
    TWBR = 24; // 400kHz I2C clock
  #elif I2CDEV_IMPLEMENTATION == I2CDEV_BUILTIN_FASTWIRE
    Fastwire::setup(400, true);
  #endif
  while (!Serial);
  mpu.initialize();
  devStatus = mpu.dmpInitialize();
  // Se ponen los offsets aqui, escalados a la minima sensibilidad
  mpu.setXGyroOffset(220);
  mpu.setYGyroOffset(76);
  mpu.setZGyroOffset(-85);
  mpu.setZAccelOffset(1788); //

  if (devStatus == 0) {
    mpu.setDMPEnabled(true);

    attachInterrupt(0, dmpDataReady, RISING);
    mpuIntStatus = mpu.getIntStatus();
```

```
dmpReady = true;

    packetSize = mpu.dmpGetFIFOPacketSize();
} else {

}
//Imprime en pantalla los siguientes mensajes
Serial.println("Bienvenido al menu de Roboband");
Serial.println("Selecciona una opcion (introduce una letra de la a al f):");
;
Serial.println("a.- Lectura de temperatura");
Serial.println("b.- Lectura de humedad");
Serial.println("c.- Lectura de acc/gyl");
Serial.println("d.- Lectura de pulso");
Serial.println("e.- Datos interoperativos");
Serial.println("f.- Almacenamiento en SD");
}
/* VOID LOOP
 * Las instrucciones escritas dentro de la seccion void loop() se
 * ejecutan justo despues de las de la seccion void setup()
 * infinitas veces hasta que la placa se apague
 * (o se resetee).
 */
void loop()
{
t.update();
//MPU configuraciones
if (!dmpReady) return;
while (!mpuInterrupt && fifoCount < packetSize) { }
mpuInterrupt = false;
mpuIntStatus = mpu.getIntStatus();
fifoCount = mpu.getFIFOCount();
if ((mpuIntStatus & 0x10) || fifoCount == 1024) {
    mpu.resetFIFO();
} else if (mpuIntStatus & 0x02) {
    while (fifoCount < packetSize) fifoCount = mpu.getFIFOCount();
    mpu.getFIFOBytes(fifoBuffer, packetSize);
    fifoCount -= packetSize;}
if (Serial.available())
{
    rec = Serial.read();

    switch(rec) //Para la lectura de las diferentes opciones
    {
        case 'a':
            t.every(1500, enviart);
            break;
        case 'b':
            t.every(1500, enviarh);
            break;
        case 'c':
            t.every(1500, enviarag);
            break;
```

```
case 'd':

for (int count =0; count <5; count++){
    serialOutput() ;

    if (QS == true){        // Se ha encontrado un latido y se
                            // determinan los valores
                                // BPM and IBI
                                // Se cuantifican en la variable QS
        fadeRate = 255;    // Quitar
        serialOutputWhenBeatHappens(); // Se llama a la función cuando
                                // se detecta un pulso.
        QS = false;          // Se resetea la variable QS para
                                // el siguiente pulso.
    }
    delay(20);
    count -= 1; }

break;

case 'e': //Datos interoperativos
    enviarop();
break;
case 'f': //Almacenamiento en SD
    enviarsd();
break;
}
}}
```

10.3 Módulo de monitorización

El módulo de monitorización consta de cuatro sensores y un microcontrolador. Los elementos seleccionados son los siguientes: Arduino Nano, DHT11 (sensor de humedad), APDS-9008 (sensor para detectar el pulso cardíaco), LM35 (sensor de humedad) y MPU-6050 (sensor para detectar los ciclos de sueño).

*ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.*

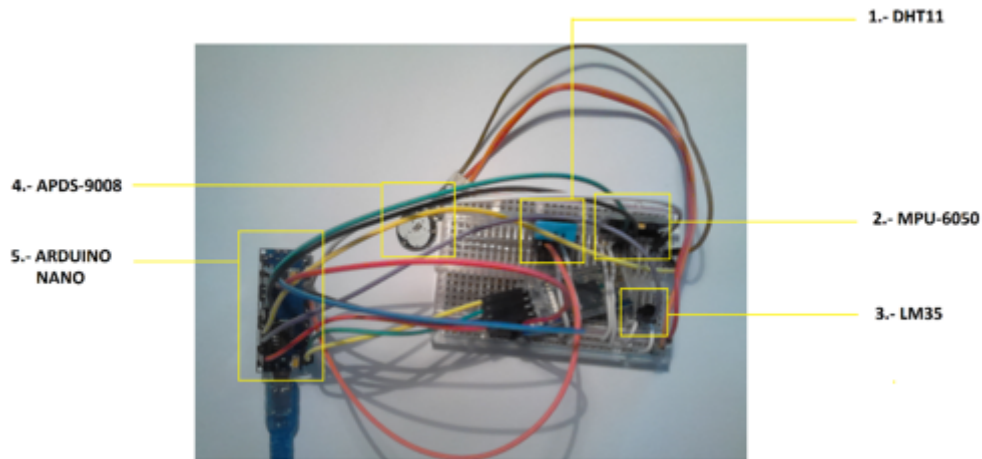


Figura 92: Apéndice II. Foto del dispositivo que permite la monitorización, con sus elementos.

El circuito final correspondiente al módulo de monitorización es el siguiente:

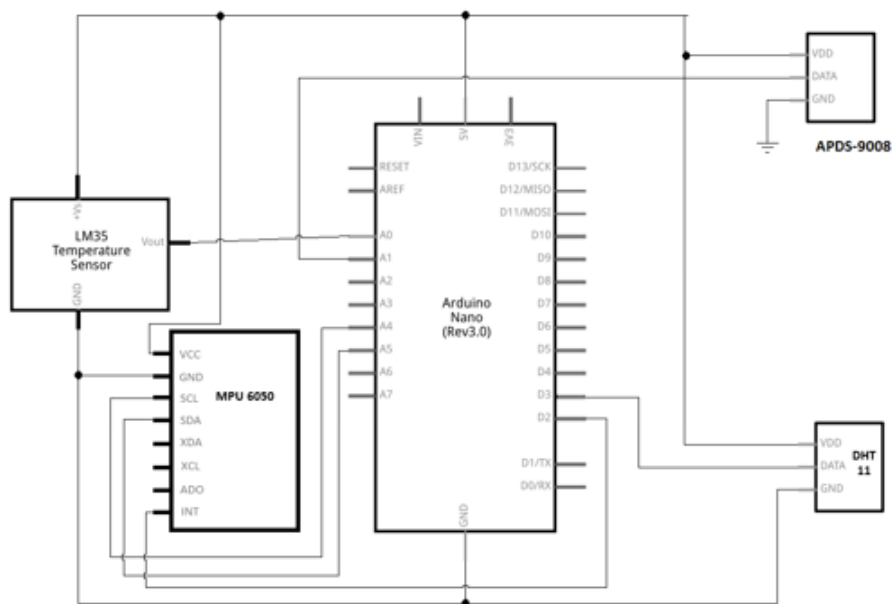


Figura 93: Apéndice II. Circuito correspondiente a la monitorización.

10.4 Módulo de comunicación

El módulo de comunicación es configurable gracias al uso de los comandos AT. A continuación se muestra el código para la configuración del HC-06.

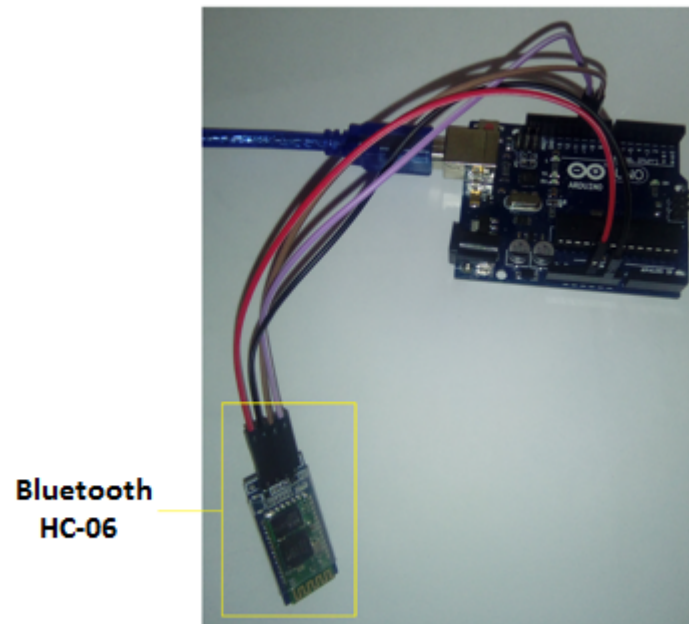


Figura 94: Apéndice II. Foto del dispositivo que permite la comunicación inalámbrica.

```
#include <SoftwareSerial.h>
  SoftwareSerial BT(10,11);
void setup(){

  Serial.begin(9600);
  Serial.println("Comando AT:");
  BT.begin(9600);

}

void loop(){

  if (BT.available()){
    Serial.write(BT.read());
  }

  if (Serial.available()){
    BT.write(Serial.read()); } }
```

Para cambiar la velocidad hay que poner el comando AT+BAUDn, n se refiere a 1,2,3,4, etc, en función de la velocidad que se quiera poner (1 ->1200 baudios, 2 ->2400 baudios, 3 ->4800 baudios, 4 ->9600 baudios, 5 ->19200 baudios, 6 ->38400 baudios, 7 ->57600 baudios, 8 ->115200 baudios). Para cambiar la contraseña hay que escribir AT+PIN y seguido de la contraseña que se quiera poner. Para cambiar el nombre se escribe AT+NAMEnombreaqui.

10.5 Módulo de almacenamiento

El módulo de almacenamiento sirve para almacenar los datos de los diferentes sensores en una tarjeta SD. No hay que añadir ninguna librería, ya que la librería para almacenamiento SD viene junto con el IDE de Arduino. El módulo usa el estándar de comunicación SPI formado por un bus de cuatro señales: MISO, MOSI, SCK (CLK) y la señal del pin seleccionado, CS. En la siguiente ilustración se puede ver el resultado del módulo.

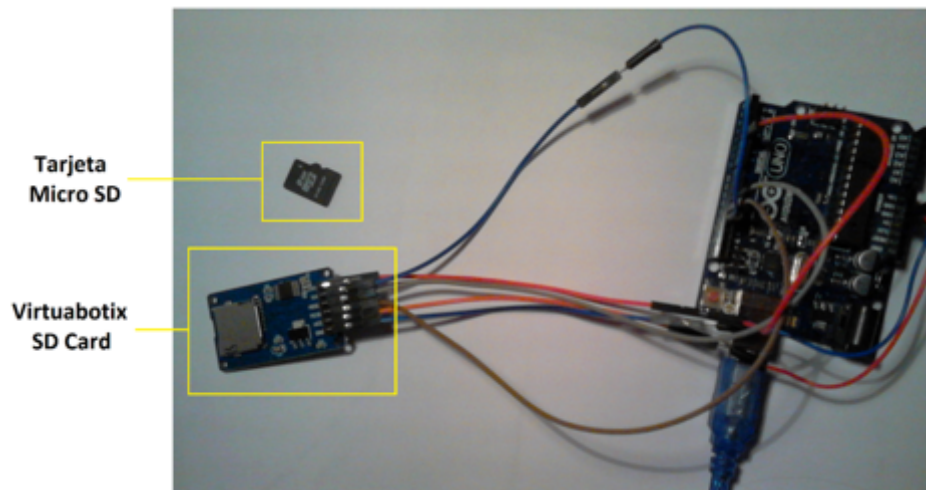


Figura 95: Apéndice II. Foto del dispositivo que permite el almacenamiento en una tarjeta micro SD.

ANÁLISIS E IMPLEMENTACIÓN DE UN SISTEMA
EN LA PLATAFORMA ARDUINO Y ANDROID PARA
OBTENER CONSTANTES VITALES ORIENTADO A
PERSONAS CON DIABETES TIPO I.

```
case '5': //Almacenamiento de datos en la microSD
{
    int v = analogRead(temperatura_pin);
    mpu.dmpGetQuaternion(&q, fifoBuffer);
    mpu.dmpGetGravity(&gravity, &q);
    mpu.dmpGetYawPitchRoll(ypr, &q, &gravity);
    mpu.dmpGetAccel(&aa, fifoBuffer);
    mpu.dmpGetGravity(&gravity, &q);
    mpu.dmpGetLinearAccel(&aaReal, &aa, &gravity);
    DatoSensor = SD.open("DATOS.txt", FILE_WRITE);
        DatoSensor.print(pulso(bpm)); //Se
            escriben los datos del pulso en la SD
    DatoSensor.print(","); //Se escribe
        una coma entre datos y dato
DatoSensor.print(lm35_convertor(v)); //Se escriben los
    datos de la temperatura en la SD
    DatoSensor.print(","); //Se escribe
        una coma entre datos y dato
    DatoSensor.print(ypr[0] * 180/M_PI); //Se
        escriben los datos de acelerometría y giroscopo en la SD
    DatoSensor.print(","); //Se escribe
        una coma entre datos y dato
    DatoSensor.print(ypr[1] * 180/M_PI);
    DatoSensor.print(",");
    DatoSensor.print(ypr[2] * 180/M_PI);
    DatoSensor.print(",");
    DatoSensor.print(aaReal.x);
    DatoSensor.print(",");
    DatoSensor.print(aaReal.y);
    DatoSensor.print(",");
    DatoSensor.print(aaReal.z);
    DatoSensor.print(",");
    DatoSensor.println(h); //Se escribe
        la humedad al final de la línea
    DatoSensor.close(); //Se cierra
        el archivo
    break;
}
}}
```