



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

TELECOM ESCUELA
TÈCNICA **VLC** SUPERIOR
DE **UPV** INGENIEROS
DE TELECOMUNICACIÓN

Escola Tècnica Superior d'Enginyeria de Telecomunicació

Universitat Politècnica de València

Dispositivo de monitorización de temperatura

TRABAJO FIN DE GRADO

Autor: Rafael Lizán Ortiz

Tutor: Vicent Miquel Rodrigo Peñarrocha

Curso académico: 2017 - 2018

Resumen

En este proyecto final de grado se describe el desarrollo de un prototipo de dispositivo cuyo objetivo tiene el de mejorar un proceso médico ya existente y aplicar la tecnología para mejorar la calidad de vida de pacientes diabéticos. Este desarrollo se realiza para que los pacientes puedan tener monitorizada su enfermedad desde cualquier lugar en el que se encuentren.

Durante el transcurso del proyecto, se examinarán diferentes tecnologías y nos enfocaremos en un desarrollo tanto de hardware como de software. Utilizaremos las tecnologías Python, .NET y WindowsForms. Por parte de los elementos finales que desarrollaremos contaremos con un ordenador de bolsillo (*Raspberry Pi*) que será el motor principal y una aplicación software creada en .NET.

Destacamos algunas características funcionales del desarrollo: el tener una base de datos con información del usuario, el envío de los mismos a través de SMTP y a un servidor FTP, la visualización de los datos mediante una aplicación y realizar medidas con un ordenador de bolsillo.

Palabras clave: .NET, Python, Raspberry Pi, SMTP, FTP.

Resum

En este projecte final de grau es descriu el desenrotllament d'un prototip de dispositiu l'objectiu del qual és el de millorar un procés mèdic ja existent i aplicar la tecnologia per a millorar la qualitat de vida de pacients diabètics. Este desenvolupament es realitza perquè els pacients puguin tindre monitoritzada la seua malaltia des de qualsevol lloc en què es troben.

Durant el transcurs del projecte, s'examinaran diferents tecnologies i ens enfocarem en un desenrotllament tant de hardware com de programari. Utilitzarem les tecnologies Python, .NET i WindowsForms. Per part dels elements finals que produïrem comptarem amb un ordinador de butxaca (*Raspberry Pi*) que serà el motor principal i una aplicació software creada en .NET.

Destaquem algunes característiques funcionals del desenvolupament: el tindre una base de dades amb informació de l'usuari, l'enviament dels mateixos a través de SMTP i a un servidor FTP, la visualització de les dades per mitjà d'una aplicació i realitzar mesures amb un ordinador de butxaca.

Paraules clau: .NET, Python, Raspberry Pi, SMTP, FTP.

Abstract

This final thesis describes the development of a prototype device whose objective is to improve an existing medical process and apply technology to improve the quality of life of diabetic patients. This development is done so that patients can have their disease monitored from anywhere they reside.

During the course of the project, different technologies will be examined and we will focus on both hardware and software development. We will use Python,.NET and WindowsForms technologies. For the final elements that we will develop, we will have a handheld computer (*Raspberry Pi*) which will be the main engine and a software application created in.NET.

We highlight some functional characteristics of the development: having a database with user information, sending them via SMTP and an FTP server, viewing the data through an application and performing measurements with a handheld computer.

Keywords: .NET, Python, Raspberry Pi, SMTP, FTP.

Índice General

1	Descripción del proyecto	6
1.1	Motivación	6
1.2	Objetivos	7
1.3	Estructura de la memoria	7
2	Introducción a los ordenadores de placa reducida (SBC)	9
2.1	Historia de los ordenadores de placa reducida	9
2.2	Mercado actual	10
2.3	Beneficios para la sociedad: ejemplos	11
3	Introducción a las enfermedades crónicas	14
3.1	Tipos de enfermedades crónicas	14
3.2	Diabetes Mellitus	14
3.3	Complicaciones en pacientes diabéticos	15
4	Planteamiento del problema a resolver	17
4.1	Problema a resolver	17
5	Elementos a emplear	20
5.1	Microcontrolador: Raspberry Pi	20
5.1.1	Hardware	21
5.1.2	Programa informático	22
5.1.3	Comparativa frente a otros modelos	23
5.2	Sensores de temperatura	23
5.2.1	Sensores infrarrojos	24
5.2.2	Sensores termopares	25
5.2.3	Sensores termorresistentes (RTD)	26
5.2.4	Sensor MLX90614	27
5.3	Elementos auxiliares necesarios	28
6	Solución propuesta	29
6.1	Equipamiento	31
6.1.1	Protocolo I2C	31
6.1.2	Montaje del sistema	34
6.1.3	Presupuesto	35
6.2	Aplicación informática	36
6.2.1	Vistas de la aplicación	37
6.2.2	Inicio	37
6.2.3	Toma de datos	39
6.2.4	Visualización de los datos	40
6.2.5	Envío de los datos	41

6.3	Lógica de la aplicación	42
6.3.1	Inicio	42
6.3.2	Toma de datos	42
6.3.3	Visualización de los datos	45
6.3.4	Envío de los datos	45
7	Manual del usuario	47
7.1	Inicio de sesión del usuario	47
7.2	Inicio	47
7.3	Toma de datos	47
7.4	Envío de datos	48
7.5	Visualización de los datos	49
8	Manual del desarrollador	50
8.1	Introducción	50
8.2	Preparación del proyecto	50
8.3	Scripts de Python	53
8.4	Aplicación en .NET	54
9	Conclusiones	55
9.1	Contratiempos/ problemas encontrados	55
9.2	Futuras vías del trabajo	56
9.3	Conclusión final	57
	Bibliografía	58

Índice de figuras

1	Ordenadores de placa reducida	9
2	Python logo	11
3	Easton Lachapelle y su brazo robótico (último modelo) (2015) [6]	12
4	Flujo de concertar cita un paciente	17
5	Flujo vista paciente al centro médico	18
6	Flujo del proceso en el centro médico	19
7	Raspberry Pi 3 Modelo B v1.2	21
8	Logos de los sistemas operativos	22
9	Esquema eléctrico del sensor infrarrojo	24
10	Ejemplo de sensor termopar	25
11	Ejemplo de sensor RTD	26
12	Sensor de temperatura infrarrojo MLX90614	27
13	Elementos auxiliares a emplear	28
14	Solución parte paciente	29
15	Solución parte especialista	29
16	Puntos de medición del pie	30
17	Esquema del bus I ² C	31
18	Ejemplo de transmisión completa del I ² C	33
19	Mecanismo de alimentación y visualización de la <i>Raspberry Pi</i>	34
20	Cableado de los sensores MLX90614	35
21	Pantalla de <i>Login</i> de la aplicación	37
22	Pantalla de <i>Inicio</i> de la aplicación	38
23	Pantalla de <i>Toma de datos</i> de la aplicación	39
24	Mensaje de confirmación tras la medida satisfactoria	40
25	Pantalla de <i>Visualización de datos</i> de la aplicación	40
26	Pantalla de <i>Envío de los datos</i> de la aplicación	41
27	Posibles salidas de la pantalla de <i>Toma de datos</i>	48
28	Mensaje de error al enviar los datos	48
29	Mensaje de confirmación al enviar los datos	49
30	Aplicación de Windows Forms	51
31	Estructura de carpetas	51
32	Vista de Spyder y archivo en blanco	52
33	Arquitectura de la aplicación	53
34	Dirección de un sensor conectado a la <i>Raspberry Pi</i>	54
35	Ejemplo de estructura y datos en base de datos	54

1 Descripción del proyecto

En este primer capítulo exponemos los motivos por los cuáles nos ha llevado a la realización de este trabajo, así como los objetivos que nos hemos propuesto para el mismo, una pequeña descripción de los capítulos que forman la memoria, los resultados que esperamos obtener con los objetivos marcados y una pequeña descripción de las futuras vías de desarrollo que pueda dejar este trabajo.

1.1 Motivación

A todo el mundo le gustaría que su vida fuera más fácil en algunos aspectos. Cada uno se imagina como sería su vida si dispusiera de algo con lo que pudiera facilitarle el día a día. En algunos de estos casos, estas personas son gente que sufre de alguna enfermedad, y que, en la medida de lo posible, le gustaría poder disfrutar de las cosas como hacía en el pasado.

Hoy en día existen infinidad de soluciones para problemas que podemos encontrar en la vida cotidiana. Muchas veces estos problemas suponen un esfuerzo insignificante para nosotros, como podría ser el tener que escoger un mando de entre varios para poder controlar un dispositivo en específico. Si tan solo tuviéramos un único mando con el que poder controlar todos los dispositivos, esta tarea nos resultaría más llevadera aunque no suponga una clara mejora de la calidad de vida. En otros casos, los problemas que nos encontramos no son tan triviales. Hay casos en los que si pudiéramos tener un dispositivo, que usándolo de manera cómoda, y diariamente se pudiera frenar o controlar la aparición de nuevas y más severas enfermedades, sería de una gran ayuda. Un ejemplo de este tipo si que significaría un salto abismal en la calidad de vida de las personas.

Muchas veces se llegan a estas soluciones a través de la tecnología. La tecnología nos brinda la oportunidad de crear nuevos inventos, o incluso reagrupar algunos que ya existen, para proporcionar una solución a un problema existente en nuestra sociedad. Muchas de estas veces, el problema es más simple de lo que nos pensamos, pero puede resultar en un aumento de la calidad de vida y la felicidad de las personas colosal. Los ingenieros entre otros, tenemos la oportunidad de ayudar a la sociedad con los conocimientos que se obtienen durante los años de estudios. Junto con esto, una posterior motivación hace que los problemas más complicados puedan llegar a solucionarse.

1.2 Objetivos

Son muchas las personas que sufren de enfermedades que, si se dispusiera de una solución tecnológica para prevenir, frenar, o controlar sus efectos o efectos secundarios que produjera, sin dudarle le darían una oportunidad para mejorar su situación actual. En este trabajo nos vamos a centrar en concreto en las personas que sufren de *diabetes mellitus*, más comúnmente conocida como diabetes. Es habitual que algunas personas que sufren de diabetes desarrollen otras enfermedades como pueden ser la de los *Pies de Charcot* o el *Pie diabético*. Según la **OMS** (Organización Mundial de la Salud) en un estudio realizado en 2016 sobre la diabetes: "La prevalencia a nivel mundial de la patología "pie diabético" varía entre el 1,3%-4,8%." [1].

Es de notar, que cada vez hay más gente que sufre de diabetes. El número de enfermos de diabetes se ha disparado en los últimos años, habiendo según la **OMS**: "422 millones de adultos en todo el mundo tenían diabetes en 2014, frente a los 108 millones de 1980" [2]. Si miramos también las enfermedades más mortíferas del mundo, encontramos a la diabetes en el puesto número 12 en el año 2008 [3] con una estimación de 1.26 millones de muertes al año.

Es por esto, que el objetivo de este trabajo es crear una solución para la prevención, monitorización y autogestión de un indicador de los *Pies de Charcot* o de los *Pies diabéticos* como es el de la temperatura de los pies. Según diversos estudios, la temperatura de los pies con respecto a la corporal puede tomarse como indicador de si una persona va a sufrir de una de estas consecuencias de la diabetes o no. Esto se conseguirá mediante un dispositivo y una aplicación informática que, juntos, podrán ayudar al paciente a autodiagnosticarse de manera diaria sin la necesidad de ser supervisados por un médico desde cualquier lugar en el que se encuentre el paciente.

1.3 Estructura de la memoria

El presente trabajo está compuesto por nueve capítulos, y en esta sección, nos disponemos a realizar una pequeña descripción del trabajo desempeñado en cada uno de ellos.

- **Capítulo 1. Descripción del proyecto:** Este capítulo trata sobre la motivación por la cual se ha llevado a cabo este trabajo, así como los objetivos que se esperan cumplimentar del mismo.
- **Capítulo 2. Introducción a los ordenadores de bolsillo:** En este capítulo se tratará el tema de los ordenadores de bolsillo. Un breve paso por la historia de estos dispositivos y algunos ejemplos de dispositivos que han supuesto una gran ayuda para la sociedad.
- **Capítulo 3. Introducción a las enfermedades crónicas:** En este capítulo se comentará acerca de las enfermedades crónicas y en especial sobre la enfermedad de la diabetes mellitus y consecuencias que puedan sufrir sus enfermos.

- **Capítulo 4. Planteamiento del problema a resolver:** En este capítulo se expondrá el problema que se pretende resolver, como se está actuando hasta el momento, y el margen de mejora que existe.
- **Capítulo 5. Elementos a emplear:** Este capítulo tiene como objetivo enumerar y explicar los distintos elementos que se van a emplear para la creación de la solución al problema expuesto en el capítulo anterior.
- **Capítulo 6. Solución propuesta:** Aquí mostraremos la solución que queremos desarrollar para poder lidiar con el problema que se ha encontrado. Se hablará a nivel técnico de la solución tanto a nivel hardware como de aplicación informática.
- **Capítulo 7. Manual de usuario:** En este capítulo nos enfocaremos en mostrar como es la navegación del usuario por la aplicación. Esto será una guía sencilla para que leyéndola y, junto con las capturas de pantalla pertinentes, cualquier tipo de usuario sea capaz de utilizar el dispositivo.
- **Capítulo 8. Manual del desarrollador:** Este capítulo tiene como fin el describir alguna de la información técnica que cualquier desarrollador necesitaría para poder realizar futuras mejoras o simplemente modificar la actual solución.
- **Capítulo 9. Conclusiones:** Este último capítulo tiene como propósito recoger las principales conclusiones obtenidas tras realizar el trabajo, futuras vías del trabajo y los objetivos que realmente se han alcanzado una vez finalizado el desarrollo del trabajo.

2 Introducción a los ordenadores de placa reducida (SBC)

A continuación se va a mostrar una introducción a los ordenadores de placa reducida, o más comúnmente llamados ordenadores de bolsillo y que posteriormente utilizaremos, junto con otros elementos, para alcanzar el objetivo de este proyecto.

2.1 Historia de los ordenadores de placa reducida

Los ordenadores de placa reducida o, *Single Board Computers* en inglés (SBC), son ordenadores completamente funcionales, pero que están creados y montados sobre una única placa o circuito, con microprocesadores, memoria, elementos de entrada y salida (E/S) y algunas otras características que son requeridas para el completo funcionamiento del ordenador. Los ordenadores de placa reducida fueron creados con propósitos educativos, o para su uso como controladores integrados. Todos los ordenadores que existen bajo esta categoría, tienen su completa funcionalidad integrada sobre una única placa de circuito impreso (PCB).

Algunas de las principales características de un SBC es que son de dimensiones reducidas, muchas veces el tamaño no supera el de una tarjeta de crédito. También son placas que tienen un precio considerado como bajo, ya que estas placas suelen ser muy económicas y normalmente tienen precios que no superan los 100€, aunque siempre se puede encontrar alguna excepción.

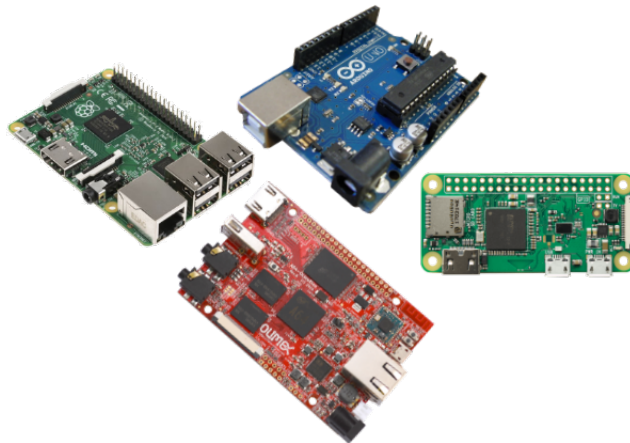


Figura 1: Ordenadores de placa reducida

Este tipo de ordenadores son relativamente jóvenes, ya que el considerado como primer SBC es el *Arduino*, cuyo primer diseño fue creado en el año 2005 [4]. Este fue inventado por, en aquel entonces estudiante del instituto IVRAE, *Massimio Banzì*, en una primera instancia ya que veía una necesidad de aprendizaje para los estudiantes de computación y electrónica. En aquel entonces adquirir una placa de micro controladores era bastante caro, además de no ofrecer un soporte adecuado. Poco tiempo después de su puesta a comercialización, esta herramienta se convertiría en el líder mundial de este tipo de ordenadores.

2.2 Mercado actual

Son muchas las empresas que fabrican y compiten en el mercado de los SBCs. Algunos de otros ejemplos que existen en la actualidad en el mercado son: Raspberry Pi (la cual usaremos en este proyecto), Olinuxino, BeagleBone, PandaBoard, y la lista continúa... En el mercado de los SBCs entran muchas empresas que compiten en distintas categorías. La principal estrategia que se busca al entrar en este mercado es, si se ofrecen prestaciones mejores, tendrá un precio más elevado. Con lo que si se requieren menos puertos USB, o si no estamos interesados en que tenga una entrada de Ethernet (RJ45), o no se necesita una entrada de audio (jack) etc, podemos conseguir dispositivos más económicos.

Arduino

Como se ha comentado anteriormente, el primer prototipo fue desarrollado en el instituto IVRAE. Este primer diseño se basó en una simple placa de circuitos eléctricos, donde se conectaban un micro controlador junto con unas resistencias. En un principio la placa únicamente permitía que se conectaran sensores simples, como pueden ser unos leds u otras resistencias, aún no existía un lenguaje de programación para manipularla.

Pasado un tiempo, se dieron cuenta de que se necesitaba un lenguaje para programar el procesador de la placa, por lo que en un inicio se usó *Wiring* [5]. Hoy en día, el *Arduino* cuenta con un IDE propio y que usa librerías del lenguaje *C++*. También es una placa a la cual se le pueden poner *add-ons* (módulos adaptables) que aportan nuevas funcionalidades a las básicas que proporciona el *Arduino*.

Raspberry Pi

La idea de la Raspberry Pi surgió durante el 2006, pero no fue lanzado al mercado hasta el febrero del 2012 [6]. En este caso, la herramienta fue desarrollada por un grupo de la Universidad de Cambridge, y el objetivo es el mismo que el de la mayoría de SBCs: el fomentar la enseñanza de la electrónica y las ciencias de la computación. El objetivo que tiene esta fundación es el de desarrollar el uso y entendimiento de los ordenadores para los más pequeños. El propósito final es que los niños puedan usar estos SBCs 'baratos' sin ningún miedo, creando nuevas aplicaciones o soluciones a problemas bajo el lema "ábrelo y mira como funciona".

Una de las principales diferencias entre la *Raspberry Pi* y el *Arduino* es que el lenguaje de programación que usa es *Python*. Este lenguaje está muy extendido en el uso de big data, y es más libre a la hora de programar que el *C++* que está más orientado a la programación de microprocesadores.



Figura 2: Python logo

En el caso de este proyecto en particular, nos decantamos por el uso de la *Raspberry Pi* por:

- El lenguaje de programación (*Python*) que usa,
- El precio de la placa por las prestaciones que ofrece,
- Tiene pines que permiten usar el protocolo I2C con el sensor que se va a usar (MLX90614)

2.3 Beneficios para la sociedad: ejemplos

Cada vez es más y más frecuente el encontrarnos con proyectos que solucionan necesidades humanas y que aportan un valor incalculable a la sociedad. No solo esto, si no que también estamos viendo que cada vez, como estas herramientas son consideradas 'asequibles' ya que su precio no es muy elevado y existen infinidad de guías, vemos como los inventores de nuevos dispositivos son muy jóvenes.

Un ejemplo que cabría remarcar es el de *Easton LaChapelle*. Un chico de Colorado, Estados Unidos, que a la edad de tan solo 14 años, hizo una mano robótica juntando piezas de LEGO, hilos de pesca y tubos eléctricos. A la edad de 17, mejoró su primera invención y creó un brazo ortopédico, totalmente funcional, y que podía controlarse con los impulsos del cerebro. El enfoque de su proyecto fue el de que cualquiera que necesitara de una mano robótica, pudiera conseguirla sin mucho esfuerzo, ya que en algunos casos estas extremidades ficticias pueden llegar a costar hasta los \$80,000.

En su proyecto, lo que hizo fue una mano robótica que pudiera ser impresa en su totalidad por una impresora 3D, y estaría controlado por un *Arduino* como motor de aquella invención. Todo esto fue, porque disponía de materiales que le permitían aprender y con los cuales podía plasmar su visión. Fue mejorando sus primeros modelos hasta conseguir la versión que está disponible hoy en día y que se puede ver en la figura 3 [6].



Figura 3: Easton Lachapelle y su brazo robótico (último modelo) (2015) [6]

Otro ejemplo de proyecto que puede ayudar a un sector de nuestra sociedad, en concreto el sector primario (que representa cerca del 4,5% del empleo en España [7]) es el de una estación meteorológica para mediciones en campos de cultivo. Incluso a un sector como es el de ganadería, agricultura y extracción de materias primas, le puede ser útil el emplear estos tipos de dispositivos para mejorar su producción.

Existe un *HAT*¹ para la *Raspberry Pi* que se llama *Raspberry Pi Weather Station*, con la que se podrá crear una estación meteorológica en cualquier lugar por menos de 100€. Esta estación cuenta con una serie de sensores:

- Calidad del aire
- Humedad
- Presión
- Anemómetro
- Temperatura de la tierra

¹Del inglés *Hardware Attached on Top*, son piezas que se pueden montar encima de la *Raspberry Pi* que aportan funcionalidades extra al dispositivo. Podemos encontrar diversos tipos de sensores ensamblados de esta manera

Contando con estos sensores se puede sacar información muy valiosa que la gente que trabaje en el campo puede utilizar en su favor. Puede saber cambios de viento para proteger plantaciones, temperatura de la tierra por si tuvieran que automatizar un proceso de enfriamiento, humedad para saber cuando es el momento óptimo para regar la tierra, la presión por si hiciera que los frutos no crecieran bien... Todo esto puede conectarse con la *Raspberry Pi* y enviar los datos a través de *3G* o *4G* (*dependiendo de la antena que se conecte*) y recibirlos en un smartphone a tiempo real. Las posibilidades de mejorar los procesos actualmente existentes son infinitas con este tipo de ordenadores.

Estos han sido algunos de los ejemplos de creaciones a partir de los ordenadores de placa reducida, pero hay un millar más de inventos que utilizan estos SBCs. Es asequible para cualquier edad tanto económicamente como a conocimientos se refiere.

3 Introducción a las enfermedades crónicas

Vamos a introducir ahora, que son las enfermedades crónicas y que tipos existen. Nos centraremos después en la *Diabetes Mellitus*, que es la comúnmente conocida como diabetes, ya que el prototipo que vamos a crear estará enfocado a mejorar un proceso para unas enfermedades como consecuencia de la diabetes en algunos pacientes.

3.1 Tipos de enfermedades crónicas

Las enfermedades crónicas son enfermedades de larga duración y por lo general de progresión lenta. Las enfermedades cardíacas, los infartos, el cáncer, las enfermedades respiratorias y la diabetes, son las principales causas de mortalidad en el mundo, siendo responsables del 63% de las muertes. En 2008, 36 millones de personas murieron de una enfermedad crónica, de las cuales la mitad era de sexo femenino y el 29% era de menos de 60 años de edad [8].

3.2 Diabetes Mellitus

La diabetes mellitus, más comúnmente conocida como diabetes, es una enfermedad que se sufre cuando el páncreas no crea suficiente insulina o cuándo este no actúa correctamente en el organismo, dado a que las células no responden como deberían. Esta enfermedad es bastante común entre los españoles, ya que [9] “el 13,8% de los españoles mayores de 18 años tiene diabetes tipo 2, lo que equivale a más de 5,3 millones de compatriotas en el 2011.” Estas son unas cifras bastante significantes, y si comparamos con el resto del mundo, en EEUU lo sufren unas 29,1 millones de personas [10] en el año 2011. El efecto de la diabetes no controlada es la hiperglicemia (aumento del azúcar en la sangre).

La diabetes de tipo 1 (anteriormente denominada diabetes insulino dependiente o juvenil) se caracteriza por la ausencia de síntesis de insulina. La diabetes de tipo 2 (llamada anteriormente diabetes no insulino dependiente o del adulto) tiene su origen en la incapacidad del cuerpo para utilizar eficazmente la insulina, lo que a menudo es consecuencia del exceso de peso o la inactividad física.

La diabetes gestacional corresponde a una hiperglicemia que se detecta por primera vez durante el embarazo.

Para todos estos tipos de diabetes existen complicaciones que se pueden dar y, muy a menudo, los pacientes pueden llegar a desarrollar otras enfermedades a causa de esta primera.

3.3 Complicaciones en pacientes diabéticos

A causa de esta enfermedad también se pueden desarrollar otro tipo de enfermedades: como son la de pies de Charcot, pies de diabetes, úlceras en los pies... Todos estos pacientes tienen probabilidades de sufrir una (o varias) de estas últimas enfermedades mencionadas. Dentro de la diabetes, la enfermedad más dura y seria [11] es el pie diabético. "Cada año, cerca de un millón de personas en el mundo que sufren de diabetes mellitus pierde una pierna a consecuencia de esta enfermedad (u alguna otra extremidad)". Algunas veces, estas amputaciones van precedidas de una úlcera en el pie. Es por eso, que la prevención y cuidado del paciente una vez instaurada la úlcera es muy importante [12].

Generalmente cuando un paciente sufre la enfermedad del pie diabético tiende a tener una disminución sensorial a la presión, a la vibración, al dolor y no es consciente de las temperaturas elevadas que puede llegar a tener el pie. En esta enfermedad hay una encrucijada, un paciente de diabetes no es consciente de que tiene pie diabético hasta que es demasiado tarde, ya que no es sensible al cambio de temperatura del pie. Se ha evidenciado que la monitorización de la temperatura de la piel de la planta del pie es un método efectivo para diagnosticar esta enfermedad, y llevar un control de la temperatura de la planta del pie de un paciente, podría reducir complicaciones posteriores en el pie [12].

Hoy en día, la solución para la prevención de esta enfermedad es, que el paciente acuda a un hospital y que haya un equipo especializado para que se realice esta monitorización. Esto claro, conlleva un posterior procesamiento y análisis de los datos, monitorización del paciente (de manera manual, sin un proceso automático)... Al acudir a un hospital se consumen muchos recursos: médicos, especialistas, ayudantes, instrumentación especializada, todo esto es un gran coste para los hospitales y por consiguiente para el estado. No solo esto, si no que también existe la posibilidad del error humano, ya que un trabajador de un centro médico podría tener un error que al paciente podría repercutirle de manera muy negativa.

En la página web del **Instituto Valenciano del Pie (IVPIE)** podemos encontrar el tratamiento que se ofrece para el *pie diabético*. El tratamiento que ofrecen es el siguiente:

"La prevención de complicación en el pie diabético se basa en educación del paciente y medidas de higiene con **revisiones periódicas por parte del podólogo**. Es para ellos que los podólogos establecen el nivel de riesgo que tiene el paciente para poder llevar un control correcto.

En el caso de que la causa de la herida sea una deficiencia de la circulación, **se derivará al paciente al cirujano para que estudie el caso y valore la posibilidad de realizar una cirugía** que modifique la circulación para que se pueda cicatrizar la herida."

Aquí vemos como claramente, si se encuentra una complicación, en el tiempo que se desarrolla y se te deriva al cirujano puede ser que ya sea demasiado tarde como para recuperarse de la enfermedad. También hemos leído que las medidas a tomar son *revisiones periódicas por parte del podólogo*. Este tratamiento es del que hemos hablado en el párrafo anterior, y este es el proceso que se pretende mejorar para que el paciente y los médicos puedan tener un control sobre la enfermedad y llevar unas pautas preventivas para ello.

4 Planteamiento del problema a resolver

En esta sección se va a introducir el problema que se pretende resolver. Sabemos que existe un proceso que tiene un margen de mejora y que tenemos como objetivo de este proyecto. A continuación, expondremos el problema que hemos detectado y que queremos a actualizar.

4.1 Problema a resolver

Como se ha mencionado anteriormente, actualmente en España, existen muy pocas soluciones para este problema, y menos aun, soluciones que el paciente conozca y pueda adoptarlas [13]. El problema es ese, el paciente no tiene suficiente información de que debe hacer ni con que solución debe ir. Una solución que encontramos es que los pacientes deben ir al hospital para que puedan tener esa toma de temperatura hecha por alguno(s) de los enfermeros. Esto quiere decir que no solo se está consumiendo tiempo del paciente al tener que acudir a un centro médico y recursos del paciente, si no que también se están consumiendo muchos recursos del centro médico (especialistas, médicos, enfermeros, gente de administración...). La propuesta es una solución en la que el paciente no tenga que depender de desplazamientos desde su domicilio y que pueda ser tramitado todo vía telemática.

Un flujo posible que podría presentar este proceso es el siguiente:

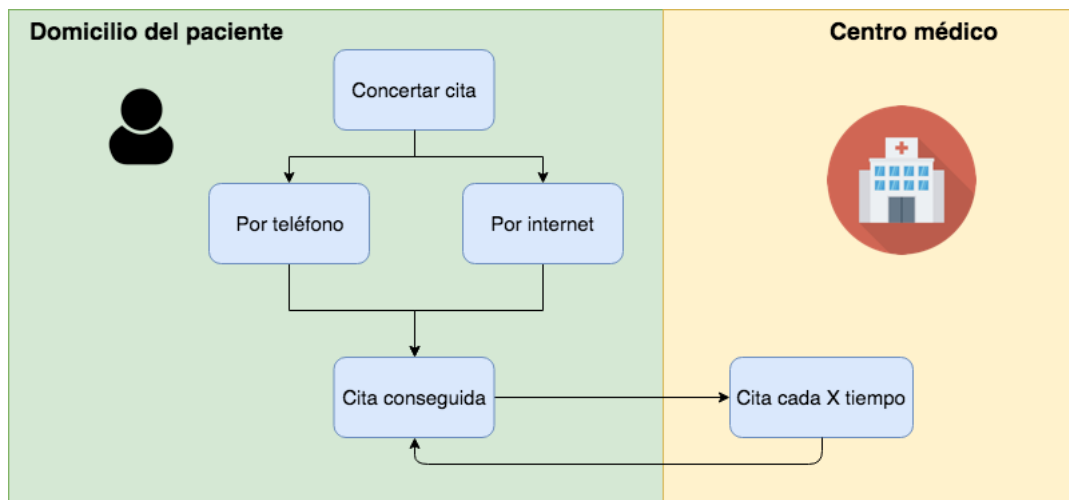


Figura 4: Flujo de concertar cita un paciente

Este proceso que vemos en la figura 4 es el que tiene que realizar un paciente hasta que consigue la primera cita, posteriormente se le asignará una cita periódica cada vez que acuda al hospital. Este proceso es sólo el de solicitar y conseguir una cita con el equipo médico. A continuación veremos otro flujo que tendrá que suceder una vez el paciente llegue al hospital y tenga que recibir la medición de temperatura por parte de los especialistas médicos, con todo lo que conlleva.

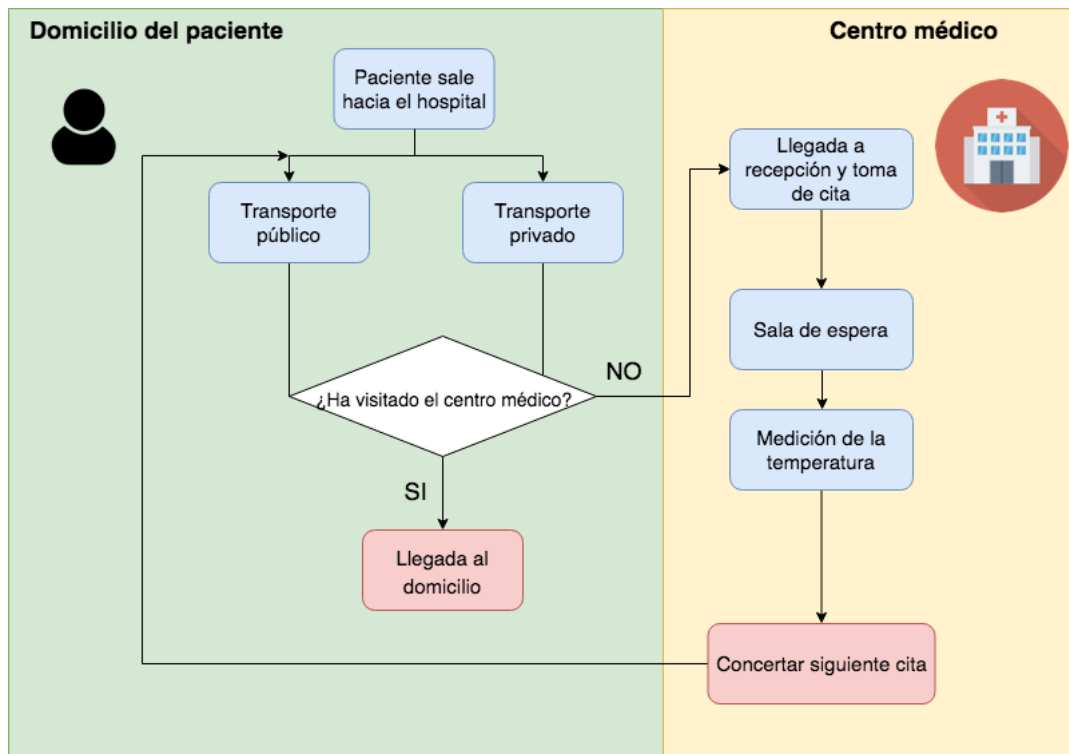


Figura 5: Flujo vista paciente al centro médico

Todo este proceso tiene que ocurrir desde el punto de vista del paciente. En el proceso de la figura 5, se tiene en cuenta tanto el coste físico como el coste de tiempo que un paciente tardaría en completar el proceso de la medición de la temperatura de los pies actualmente. A continuación vamos a analizar como sería el proceso desde el punto de vista del centro médico, sin tener en cuenta el coste físico de los aparatos, sueldo del personal etc.

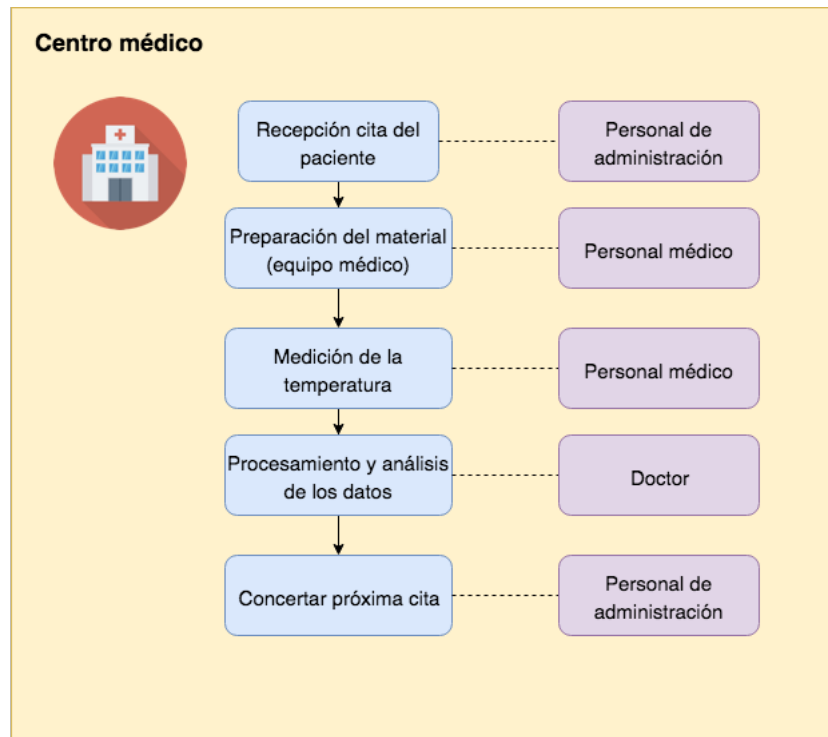


Figura 6: Flujo del proceso en el centro médico

Tras ver el proceso que ha de seguir un centro médico (figura 6) vemos como se consumen un montón de recursos al seguirlo. Vemos como efectivamente tienen que intervenir muchos activos del hospital como el personal de administración, enfermeros, especialistas... Todo esto es para un solo paciente, por lo que este proceso se multiplica por cada paciente que necesite realizar una medición de la temperatura de los pies. También hay que tener en cuenta que los pacientes necesitan de este proceso regularmente, por lo que se incrementaría aun más el coste de este proceso.

Como hemos podido analizar, existe un problema más que evidente. Hay un proceso largo y tedioso, en el cual tienen que intervenir diversas personas para que pueda completarse. Todo esto se multiplica por cada número de personal del centro médico que tenga que intervenir e invertir tiempo en este proceso.

Este es el problema que hemos visualizado, por lo que en el capítulo 6 mostraremos la solución propuesta para acabar con este.

5 Elementos a emplear

Para poder llevar a cabo la solución que se ha propuesto anteriormente, necesitaremos disponer de una serie de elementos que adaptados, crearán un prototipo o dispositivo inicial. A continuación se van a enumerar los distintos elementos necesarios.

- Microcontrolador *Raspberry Pi 3 Modelo B V1.2*
- Sensores de temperatura infrarrojos MLX90614
- Placa board
- Jumpers (M-M y H-H)
- Monitor (en este caso de 24")
- Cable HDMI
- Ratón y teclado inalámbricos
- Programa informático

Una vez sabemos que elementos se van a emplear, se indicará por que son necesarios dichos elementos, que función tienen, y por que han sido estos los escogidos de entre una amplia gama de productos donde elegir.

5.1 Microcontrolador: Raspberry Pi

Como ya se ha introducido anteriormente, la Raspberry Pi es un ordenador de bolsillo. Es un ordenador, que en algunos modelos, puede llegar a ser de dimensiones similares al de una tarjeta de crédito. Como es de esperar, la Raspberry Pi es más lenta que un ordenador portátil o uno de mesa moderno, pero sigue siendo un ordenador con sistema operativo Linux completo y que es capaz de realizar cualquier tarea que se le requiriera a un ordenador y con un bajo consumo de energía.

Este ordenador está preparado para que, con tan solo conectarlo a un monitor/televisión, un teclado y un ratón estándar pueda funcionar perfectamente y rendir (a nivel de aplicaciones) a la altura de un ordenador convencional. Es un dispositivo que en el mayor de los casos, no supera los 10cm de largo por 3cm de alto. Este dispositivo también tiene una serie de conexiones que permite que se le conecten periféricos y utilizarlos para crear casi cualquier cosa. Desde sensores (temperatura, presión, acelerómetro...), pantallas, leds, altavoces, micrófonos...

5.1.1 Hardware



Figura 7: Raspberry Pi 3 Modelo B v1.2

Hay distintos modelos de la *Raspberry Pi*. Para este desarrollo en particular, hemos escogido utilizar la *Raspberry Pi 3 Modelo B V1.2*. En este modelo encontramos prestaciones de lo más interesantes. Como núcleo de la placa nos encontramos con un chip integrado, el **Broadcom BCM2837**, el cual cuenta con un procesador 1.2GHz Quad-Core ARM Cortex-A53 por lo que tenemos hasta 1.2GHz para poder subir las frecuencias de funcionamiento sin que pierda propiedades. Este modelo también cuenta con 1GB de memoria RAM.

También podemos ver que tiene una soldadura de audio jack así como también un conector para las cámaras de tipo *DSI*. Dispone también de una toma de HDMI para poder conectarlo a cualquier monitor con calidad de vídeo en 1080p. Este modelo cuenta con la posibilidad de conectarse a la red a través de un cable (RJ45) 10/100 Ethernet y también tiene una antena Wi-Fi que cuenta con la banda de frecuencias baja que utiliza este estándar (2.4GHz). También contamos con 4 puertos USB 2.0 para conectar cualquier tipo de periféricos como pueden ser ratones, teclados... En cuanto a conexiones de red también cuenta con un módulo Bluetooth al que podemos conectarnos desde cualquier dispositivo o por el cual podemos transmitir información en la banda de frecuencia que utiliza el protocolo (2.4GHz).

En cuanto al almacenamiento de este dispositivo, no cuenta con unidades de almacenamiento de datos de serie, pero esto es porque se le puede insertar tarjetas del tipo Micro SD para poder almacenar información. Con esto podemos cambiar el sistema de almacenamiento uno por otro de manera sencilla y hace que el precio del dispositivo sea más reducido que si tuviera una unidad de almacenamiento sólida incluida. Por último, cuenta con una conexión micro USB para alimentarse con cargadores que cuenten con una corriente de al menos 750mA.

5.1.2 Programa informático

Como ya se ha mencionado anteriormente, la *Raspberry Pi* utiliza sistemas operativos que están basados en Linux (la mayoría de ellos), pero se ha de tener en cuenta que si instalamos un sistema operativo, el dispositivo tiene que poder operar de manera fluida con el GigaByte de memoria RAM que tiene. Existen diversos sistemas operativos que pueden ser empleados, los más comunes son:

- **Raspbian:** es el OS recomendado por los fabricantes de *Raspberry Pi*, es un sistema basado en Debian que ofrece un entorno igual al que podríamos tener en cualquier ordenador de mesa. Se pueden usar editores de texto, navegar por internet, jugar a juegos, crear hojas de cálculo, programar... Existen muchas aplicaciones que podemos instalar a este sistema operativo para que la experiencia de usuario sea mejor.
- **Windows IoT:** es la distribución de Windows adaptada a la *Raspberry Pi* enfocada a desarrolladores de proyectos de IoT (internet de las cosas).
- **OSMC (Open Source Media Center):** es una distribución que tiene como propósito el usar la *Raspberry Pi* como un centro multimedia en el que poder visualizar películas, imágenes, escuchar música y muchas cosas más. Está enfocado a un usuario que utilizará la *Raspberry Pi* para ocio multimedia.

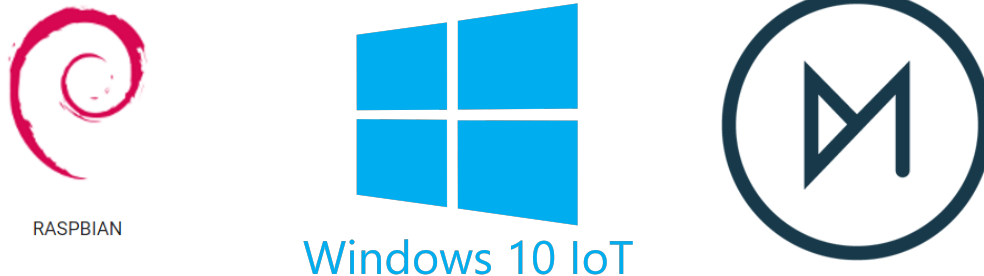


Figura 8: Logos de los sistemas operativos

En nuestro caso vamos a emplear el sistema operativo de *Raspbian* para poder ejecutar la aplicación que, posteriormente explicaremos, será la que utilice el usuario para la monitorización de la temperatura de los pies.

5.1.3 Comparativa frente a otros modelos

En este subapartado vamos a ver una comparativa de los distintos modelos y más populares que ofrece la *Raspberry Pi Foundation* que se deberían de tener en cuenta cuando se elije uno de estos dispositivos para desarrollar un proyecto. El modelo que vamos a utilizar es el más completo, se va a emplear este modelo ya que se disponía de él por algún proyecto anteriormente realizado.




	Raspberry Pi 3 B	Raspberry Pi Zero W	Raspberry Pi A+ (%)
			
Precio (\$)	35.00	20.00	10.00
Nº de cores	4	1	1
RAM	1GB DDR2	512 MB	256 MB
Bluetooth	SI	SI	NO
Wi-Fi	SI	SI	NO
USB (Tipo A)	SI	NO	SI
Tamaño (largo, ancho, alto)	85.6x56.5x17mm	65x56.5x10mm	30x65x5mm
Almacenamiento	Micro SD	Micro SD	SD

Tabla 1: Comparativa entre los modelos 3 B, Zero W y A+ de Raspberry Pi

5.2 Sensores de temperatura

Podemos encontrarnos dos tipos de sensores, los sensores de tipo analógico y los de tipo digital. La principal diferencia que existe entre los sensores digitales y analógicos es los valores de lectura. Si un sensor se activara cuando se cambia de 0V a 3V, si tuviéramos una lectura de 1.4V el sensor analógico lo interpretaría como esos 1.4V, mientras que un sensor digital suele redondear al valor más cercano (siendo 0V en este caso), por lo que solo existen dos estados, o 'alto' (high) o 'bajo' (low).

Existen muchos tipos de sensores en el mercado. Dependiendo del uso y de la aplicación que vaya a tener un sensor, se recomienda usar un tipo u otro. A continuación vamos a analizar algunos de los sensores digitales que existen y posteriormente el que vamos a emplear, el sensor digital de temperatura por infrarrojos - MLX90614.

5.2.1 Sensores infrarrojos

Los sensores infrarrojos son detectores optoelectrónicos que pueden medir la radiación electromagnética y térmica en el espectro de los infrarrojos. Las longitudes de onda de este espectro están comprendidas entre los $0.7\mu\text{m}$ hasta los $1000\mu\text{m}$. Los infrarrojos se pueden clasificar en tres categorías:

- **Infrarrojo cercano** (de 700nm a 3000nm)
- **Infrarrojo medio** (de $3\mu\text{m}$ a $60\mu\text{m}$)
- **Infrarrojo lejano** (de $60\mu\text{m}$ a $1000\mu\text{m}$)

El sensor infrarrojo está basado en el principio del cuerpo negro ideal, es decir, que la materia por su caracterización energética, emite radiación, independientemente de la longitud de onda que sea. En el esquema eléctrico del sensor, podemos observar como cuenta con un fototransistor y un diodo. Suelen tener un diodo LED como emisor (parte izquierda de la figura 9) y un fototransistor (parte derecha de la figura 9) como receptor [14].

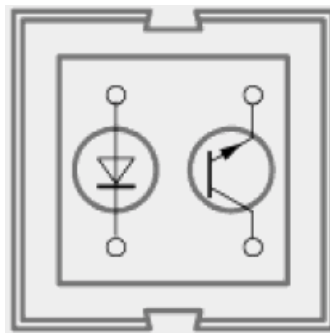


Figura 9: Esquema eléctrico del sensor infrarrojo

En los sensores infrarrojos encontramos que en la misma cara frontal están tanto el LED como el fototransistor, y debido a esta configuración, se mide la radiación que proviene del reflejo de la luz que emite el LED hacia el cuerpo que se quiere medir. Como cabe esperar la luz reflejada no suele venir de la manera que desearíamos, por lo que normalmente los dispositivos llevan incorporados circuitos de filtrado de señal para evitar errores en la lectura.

5.2.2 Sensores termopares

Los sensores de tipo termopar sirven también para medir la temperatura. Este tipo de sensores se compone normalmente de dos hilos de metales que están unidos por una extremidad. Cuando existe una variación de temperatura en presencia de este tipo de sensores, se produce una tensión proporcional a la temperatura del sistema. Existen distintos tipos de sensores termopares en el mercado, pero los más comunes son los de tipo J,K y T. Estos tipos varían según los metales que se usen para el termopar dependiendo de la aplicación [15].

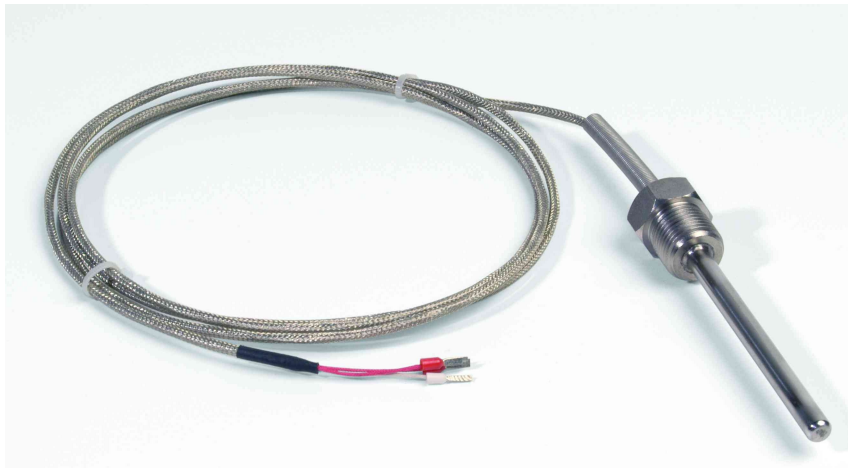


Figura 10: Ejemplo de sensor termopar

Por lo general, los termopares miden rangos de temperatura muy grandes. A la hora de elegir un tipo de termopar u otro deberíamos basarnos en los siguientes parámetros:

- Rango de temperatura en el cual se encuentre el termopar
- Resistencia química que exista en el termopar o materiales de la vaina
- Resistencia a la abrasión y a la vibración para evitar errores en la medida
- Localización donde se va a emplear el sensor

También podemos encontrar distintos tipos de sonda para los sensores termopares. Los más comunes son termopar *por cables* (la forma más simple y más utilizada), la *sonda termopar* y la *sonda termopar de superficie*.

5.2.3 Sensores termorresistentes (RTD)

Los sensores termorresistentes o RTD, son sensores que detectan la temperatura por un cambio resistivo. Este tipo de sensores presentan una cierta resistencia dependiendo de la temperatura del sistema. Está basado en la variación de la resistencia de un conductor con el cambio en temperatura. Existen distintos tipos dentro de los RTD dependiendo de la aplicación en la que se quiera utilizar, escogeremos uno u otro ya que hay algunos RTDs que tienen una resistencia de 500Ω a 0°C , pero también podemos encontrar otros que tengan resistencias de 100, 25 o 10Ω .

Este tipo de sensores tiene la siguiente fórmula para calcular la temperatura a la que se encuentra en un sistema:

$$T = \alpha \left(\frac{R}{R_0} - \alpha^{-2} \right) + T_0 \quad (5.1)$$

En la ecuación anterior tenemos la T como la temperatura final, la α es el coeficiente de temperatura del conductor a 0°C , R es la resistencia final, R_0 es la resistencia de referencia y T_0 es la temperatura de referencia.

Uno de los aspectos más comunes para este tipo de sensores es el siguiente:

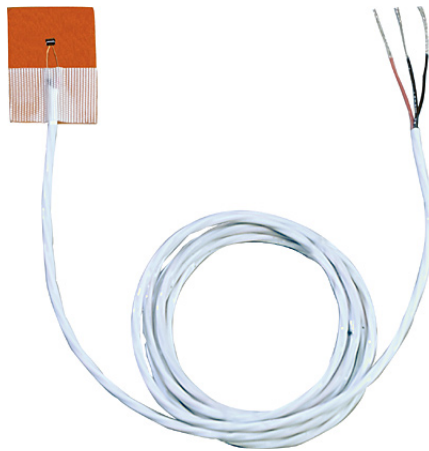


Figura 11: Ejemplo de sensor RTD

5.2.4 Sensor MLX90614

Tras haber analizado los distintos tipos de sensores de temperatura más comunes, hemos decidido utilizar un sensor de infrarrojos. En especial, hemos decidido emplear el **sensor de temperatura infrarrojo MLX90614**. Nos hemos decantado por este tipo de sensor por varios motivos, el primero es que la aplicación va a ser realizar medidas de temperatura en el pie de una persona, por lo que lo más cómodo para el paciente es este tipo de sensor. Segundo, las variaciones de temperatura que pueda tener una persona son muy pequeñas y estamos hablando de temperaturas de entre 27°C a unos 35°C. Tercero, hemos de mirar la compatibilidad de estos sensores con los ordenadores de bolsillo, por lo que este sensor en particular encajaba a la perfección.

Este sensor está diseñado para ser sensible a la radiación infrarroja emitida por un cuerpo a distancia. También cuenta con una etapa de amplificación y digitalización de la señal, y a su vez, la salida del sensor es lineal y se compensa de acuerdo con las variaciones de temperatura ambiente. Con esto nos evitamos, en la medida de lo posible, errores de lectura por parte del dispositivo.



Figura 12: Sensor de temperatura infrarrojo MLX90614

El sensor MLX90614 cuenta también con un circuito integrado que filtra el ruido, tiene un convertor analógico digital con 17 bits de resolución y un procesador digital de señales. A continuación se van a enumerar algunas de las especificaciones técnicas de este sensor:

- Alimentación a 5V.
- Rango de temperaturas ambiente de -40°C hasta +170°C.
- Rango de temperaturas del objeto de -70°C hasta +380°C.
- Precisión de $\pm 0.05^\circ\text{C}$
- Protocolo de comunicación SMBUs (I2C)

La salida de este sensor es de tipo SMBus, que utiliza el protocolo I2C que se explicará en el siguiente capítulo.

5.3 Elementos auxiliares necesarios

Se necesitan una serie de elementos auxiliares para poder crear un prototipo de dispositivo con el cual un usuario pueda realizarse tomas de medida de la temperatura de los pies. Estos materiales pueden ser sustituidos por otros similares y no causarían gran impacto en el desarrollo del prototipo (ya que dependiendo del monitor que se utilice, se empleará un cable HDMI o VGA, los periféricos del ordenador también pueden variar...).

Se requiere de un cable del tipo HDMI para poder conectar la *Raspberry Pi* a un monitor. En nuestro caso el monitor será de 24" ya que es del que disponemos. También utilizaremos un ratón y un teclado inalámbrico para poder controlar el mini-ordenador. Con esto tenemos descritos los componentes de la parte del dispositivo que el usuario va a manejar.

También necesitamos elementos a la hora del montaje, necesitamos utilizar una placa board (tablero de conexiones sin soldadura) donde tendremos las conexiones puestas y que conectaremos con nuestro dispositivo y los jumpers, que serán los cables para poder hacer las conexiones. Por lo que en su totalidad lo que necesitamos es:



Figura 13: Elementos auxiliares a emplear

Si esto fuera un producto final y no fuera un prototipo, sería una cosa a considerar el realizar un esquema eléctrico y tenerlo impreso en una PCB para su uso. También se podría crear un mini dispositivo que tuviera una pantalla de unas 12" táctil sobre la cual el usuario pudiera realizar las medidas.

6 Solución propuesta

En este capítulo se va a exponer la solución propuesta para el problema que planteamos. Se va a enfocar de varias maneras, primero un repaso sobre el hardware que se va a emplear y las conexiones que son necesarias. Posteriormente se hará un análisis al programa informático que se va a desarrollar y que se requerirá utilizar para poder realizar las medidas.

La solución que se va a proponer es la siguiente: crear un dispositivo que consista de una *Raspberry Pi* que tenga conectados unos sensores de temperatura infrarrojos para poder medir la temperatura de los pies de los pacientes. Estos datos se almacenarán en un ordenador (la misma *Raspberry Pi*) que llevará una aplicación interna en la cual los pacientes recogerán sus datos y posteriormente podrán ser visualizados, consultados y enviados a través de internet al médico. La propuesta de solución es:

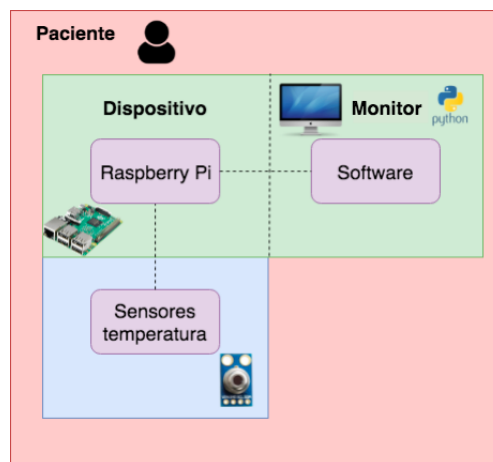


Figura 14: Solución parte paciente

El especialista únicamente debe disponer de un ordenador (que tenga instalado un programa que sea capaz de leer archivos de extensión .csv como por ejemplo el Microsoft Excel) y de una conexión a internet para poder descargarse los datos de un paciente.

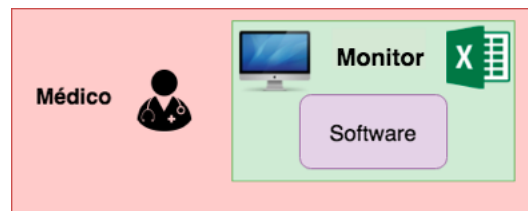


Figura 15: Solución parte especialista

Como ya se ha mencionado anteriormente, si seguimos esos estudios podemos concluir que la variación de temperatura de los pies con respecto a la corporal, es un indicativo bastante elevado de que el paciente pueda sufrir de algún tipo de enfermedad. La función que desempeñará este dispositivo es medir la temperatura de los pies (tomada en unos puntos específicos de los mismos) con respecto a la temperatura tomada en la frente como referencia de temperatura corporal. A continuación vamos a ver un diagrama que muestra en que puntos exactos del pie se precisa hacer esta medición de temperatura.



Figura 16: Puntos de medición del pie

Como se observa en la figura 16 los puntos de medición son 10, simétricos un pie con respecto al otro. Para hacer la labor de medición más sencilla de cara al usuario, lo que se va a realizar es una plataforma que contenga 12 sensores y que sea la propia plataforma (aplicación informática) la que decida que pie es al que se le está evaluando la temperatura. Se podría crear una plantilla en la que los sensores ya estén colocados previamente, para así evitarnos que el paciente tenga que insertar ningún sensor y sólo se preocupe de seguir las instrucciones del programa informático e ir subiendo los pies a la planta de nuestro dispositivo.

Según el estudio de *Monitorización de la temperatura del pie como herramienta en la neuropatía diabética* [13], los resultados muestran que las temperaturas medias del pie derecho de los pacientes que sufren de pie diabético, son de unos $2,24^{\circ}\text{C}$ mayor con respecto a la temperatura corporal. En cuanto al pie izquierdo, las temperaturas en el pie de los pacientes que sufren de la enfermedad, es de $0,86^{\circ}\text{C}$ superior a la temperatura corporal. Sabiendo esto, vemos como el error que tiene el sensor que vamos a utilizar es suficiente para poder detectar cambios de temperatura relevantes.

6.1 Equipamiento

Como hemos dicho antes, el sensor que vamos a emplear, utiliza el protocolo I2C para la transferencia de datos. A continuación vamos a ver en que consiste ese protocolo, el montaje necesario de las partes para nuestro prototipo y el presupuesto que ha tenido el desarrollo de este proyecto.

6.1.1 Protocolo I2C

I²C (inter-integrated circuit) es un protocolo de comunicación que permite la comunicación de múltiples “esclavos” (refiriéndonos a circuitos digitales integrados) a comunicarse con uno o más “maestros” en un dispositivo. Este tipo de bus se usa para conectar periféricos de baja velocidad a procesadores y micro controladores (como es el caso de la Raspberry Pi) para comunicaciones de corto alcance. Este protocolo requiere dos pines para intercambiar información (estos son el SDA y el SCL) [16].

I²C usa dos líneas bidireccionales que son colector abierto o drenador abierto con resistencias de pull-up. Este tipo de resistencias son empleadas con el fin de evitar que se hagan lecturas erróneas por ruido electrónico. Lo que se consigue es llevar la tensión a los rangos donde se considera como *HIGH* o *LOW* y no tener tensiones indeterminadas que el sistema no sabe interpretar. Estas líneas como hemos mencionado antes son las de **SDA** (serial data line) y **SCL** (serial clock line). Existe un protocolo de transferencia [17].

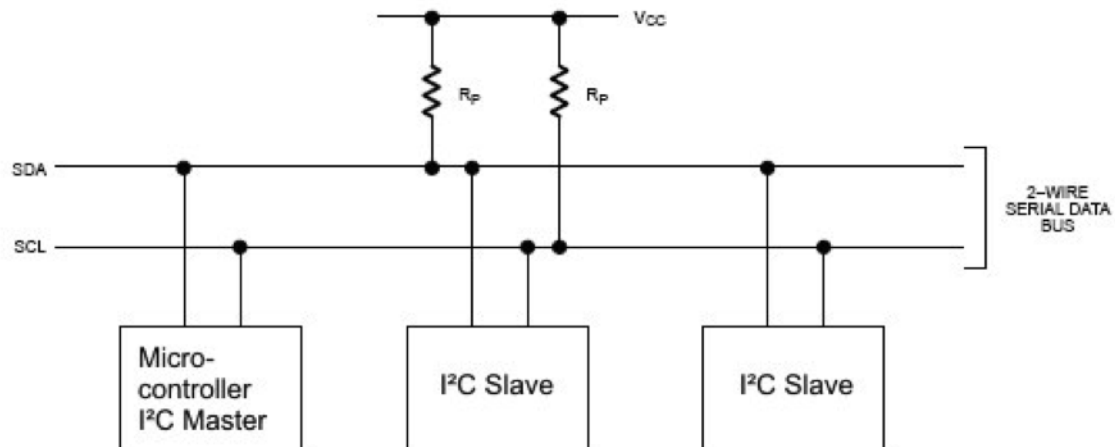


Figura 17: Esquema del bus I²C

Funcionamiento del protocolo de transferencia por I²C

En el protocolo de transferencia por medio de I²C los mensajes están separados en dos partes. Una trama de direcciones, donde el *maestro* indica al/los *esclavo(s)* a cual de ellos se le está enviando el mensaje, y uno o más tramas de datos en los que se envía un mensaje de 8 bits del *maestro* al *esclavo* o viceversa.

La información vendrá dada por la línea SDA cuando la línea de reloj vaya de alta a baja. Esta información es recogida cuando la línea del SCL se vuelve alta de nuevo. El tiempo de lectura/escritura puede variar en función del número de “esclavos” que tengamos conectados a nuestro bus.

- **Condición de inicio**

Para poder iniciar la trama de direcciones es necesario, como hemos dicho, que el *maestro* deje la señal SCL alta y la SDA baja. Con esto lo que se conseguirá es que todos los dispositivos *esclavos* que estén conectados al bus, estén atentos al inicio de la transferencia. Si hubiera dos *maestros* que quisieran iniciar comunicación a la vez, tendría prioridad el primero que baje la señal SDA.

- **Tramas de direcciones**

La trama de direcciones siempre es el primero en una nueva secuencia de comunicación. Para las direcciones de 7 bits, la dirección se marca primero con el bit más significativo (MSB), y va seguido de un bit de lectura/escritura que indica si es de lectura (1) o escritura (0).

El 9º bit de la trama es el de NACK/ACK. Esto ocurre tanto si la trama es de datos o direcciones. Cuando los primeros 8 bits de la trama son enviados, el receptor recibe el control sobre la línea de SDA. Si el dispositivo receptor no baja la línea de SDA antes del noveno pulso (del reloj), puede significar que el receptor no recibió los datos o no sabía como interpretar el mensaje. Si ocurre este último caso, el intercambio es interrumpido y, dependiendo del maestro del sistema, decide como proceder.

- **Tramas de datos**

Una vez finalizada la transmisión de la trama de direcciones, comienza la transmisión de la trama de los datos. Lo que ocurrirá es que el maestro, simplemente, continuará generando pulsos de reloj a intervalos regulares, y los datos se colocarán en la línea de SDA por el maestro o por el esclavo, dependiendo de el bit de lectura/escritura, si indica una operación de lectura o escritura. El número de tramas de datos no siempre es fijo, la mayoría de los dispositivos esclavos autoincrementarán el registro interno, lo que significa que las lecturas o escrituras posteriores vendrán en el siguiente registro de la línea. Puede haber tantas tramas de escritura/lectura como decida el que esté transmitiendo (siempre que el “maestro” lo permita y no interrumpa la comunicación).

- **Condición de parada**

Una vez ya han sido enviadas todas las tramas (tanto de direcciones como de datos), el maestro generará una condición de parada. Las condiciones de parada se definen por una transición de 0 a 1 (bajo a alto) en la línea de SDA seguida por una transición 0 a 1 (bajo a alto) en la línea SCL, con el SCL permaneciendo a nivel alto. Durante la operación normal de escritura de los datos, el valor de SDA no debe cambiar mientras SCL se mantiene en nivel alto. Esto se hace para evitar condiciones de parada falsas y no lleve a confusión.

A continuación se va a mostrar una imagen que representa un ciclo completo de la transmisión de los mensajes. Como se puede observar tendrá la **condición de inicio**, las distintas **tramas de direcciones y datos** y por último la **condición de parada**.

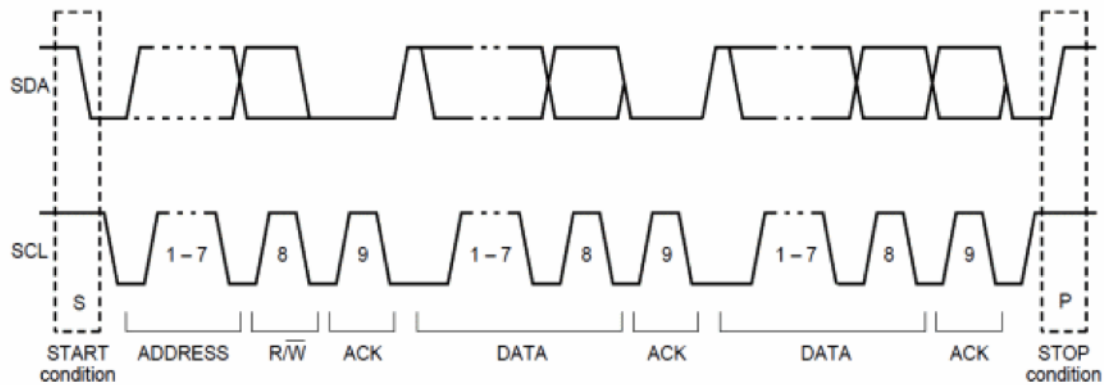


Figura 18: Ejemplo de transmisión completa del I²C

Cuando exista más de un esclavo, los distintos circuitos integrados que estén conectados, estarán obligados a generar una señal de ACK por cada trama recibida. El circuito integrado que haga el ACK, deberá poner a 0 la línea de SDA durante todo el pulso del reloj, con lo que quedará: SDA estable a nivel bajo y el periodo de ACK (señal de reloj) estará a nivel alto (se puede observar en la figura 18). Después de realizar un ACK puede darse la **condición de parar**. El *maestro* enviará una condición de inicio repetida para indicar que quiere dejar el bus en modo reposo. Con lo que si el *maestro* quiere dejar la comunicación y dejar de recibir información tendrá que enviar una señal de NACK (not ACK) después de recibir el último byte.

6.1.2 Montaje del sistema

En cuanto al montaje del sistema, podemos separarlo en dos partes, conexiones de la *Raspberry Pi* con los elementos auxiliares y alimentación necesarias, y el cableado con los sensores MLX90614.

En primer lugar vamos a ver las conexiones que son necesarias para alimentar la *Raspberry Pi* para poder visualizar el sistema operativo por pantalla.



Figura 19: Mecanismo de alimentación y visualización de la *Raspberry Pi*

Hemos visto el protocolo I2C en la sección anterior. La *Raspberry Pi* utiliza los pines **GPIO 2 y 3** para la comunicación por ese protocolo. Para conectar todos los sensores, tenemos que alinearlos en serie ya que este protocolo permite la transmisión de datos de manera secuencial. También sabemos que vamos a utilizar 12 sensores en el pie y 1 para tomar la temperatura de la cabeza como temperatura corporal de referencia.

Ahora vamos a ver como es el cableado de las conexiones entre la *Raspberry Pi* y los sensores MLX90614.

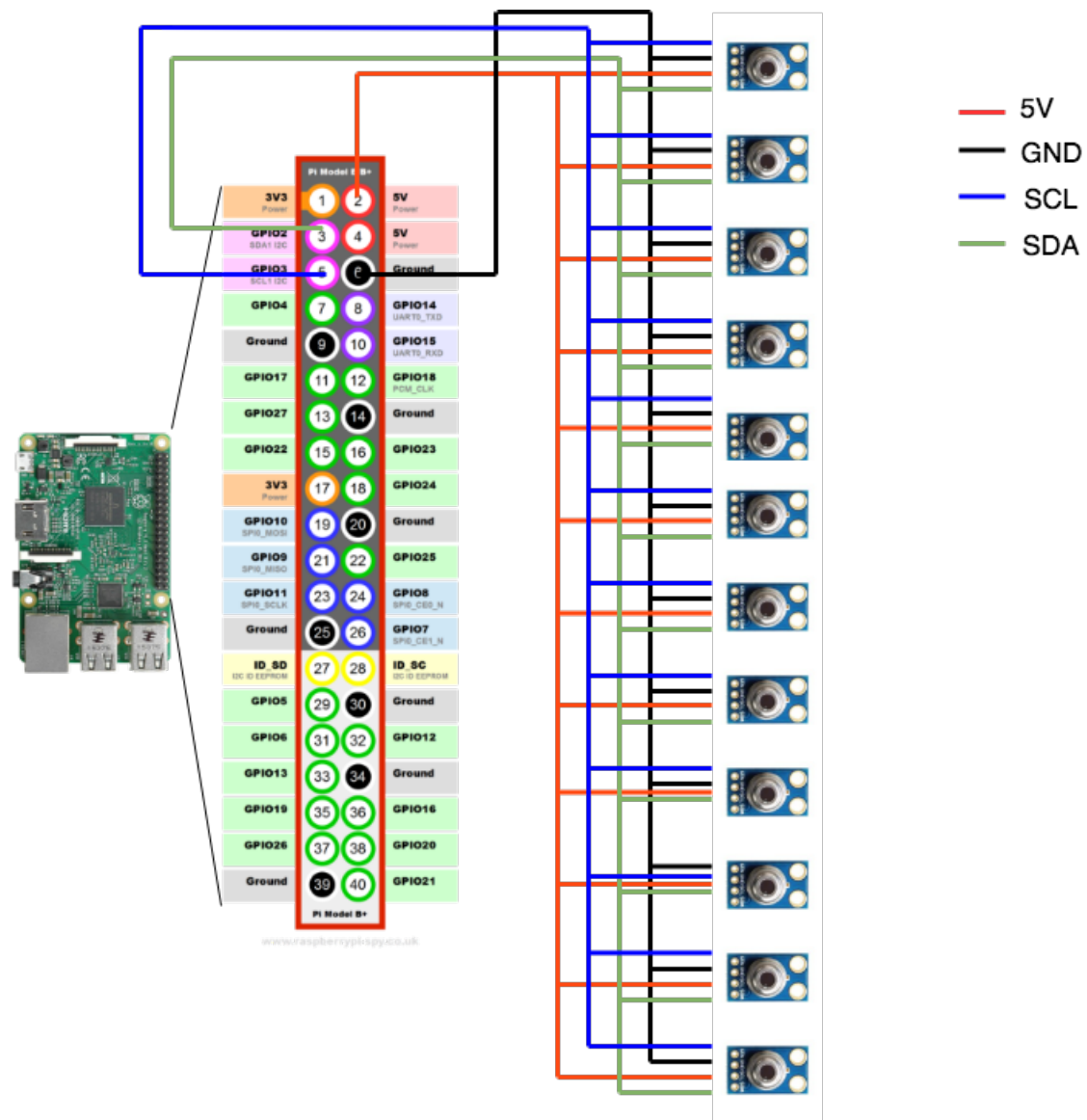


Figura 20: Cableado de los sensores MLX90614

6.1.3 Presupuesto

Se va a realizar un presupuesto de todos los componentes que vamos a requerir para poder desarrollar el prototipo del dispositivo. Ya se han comentado todos los elementos que van a ser necesarios.

Este proyecto no fue financiado por ninguna fuente externa que no fuera el propio desarrollador del trabajo. Esto hay que tenerlo en cuenta a la hora de saber por que se han elegido algunos elementos con poca repercusión, frente a otros de precio algo más elevado. También vamos a incluir los elementos de los cuales ya disponíamos pero que son necesarios para hacer que funcione el prototipo.

Artículo	Precio (por unidad) €	Total de unidades	Subtotal (€)
Raspberry Pi 3B v1.2	35.00	1	35.00
Sensor MLX90614	4.83	13	62.79
Conectores (40 jumpers)	1.00	2	2.00
Placa board	1.50	1	1.50
Monitor 24" (HDMI incluido)	99.00	1	99.00
Tarjeta Micro SD 16GB	12.00	1	12.00
Ratón y teclado inalámbricos	11.95	1	11.95
		TOTAL	224.24

Tabla 2: Presupuesto de los materiales empleados

6.2 Aplicación informática

La solución que se propone a nivel de aplicación informática es un programa que se puede usar para entornos tanto Debian (el cual es el que usa la *Raspberry Pi*) como Microsoft, y estará desarrollado con la tecnología C#. Las tecnologías que se usarán para este caso serán C# y Python. Esta aplicación será ejecutada como una aplicación de escritorio (.exe). El programa tendrá varias pantallas que se explicarán a continuación.

Podemos separar la aplicación informática en 2 partes: vistas de la aplicación y lógica de la aplicación. En primer lugar se explicarán las vistas de la aplicación(o front-end), la interfaz de usuario, que será la parte visual por la cual el usuario navegará y usará las distintas funcionalidades. Por otro lado se explicará la lógica de negocio² con la que cuenta el programa. Dentro de esta última, también haremos una distinción entre la lógica de negocio de la aplicación en si, y de la lógica que habrá que realizar con la tecnología *Python* para que la *Raspberry Pi* haga la lectura de la temperatura y podamos procesar los datos para presentarlos en la aplicación.

6.2.1 Vistas de la aplicación

La aplicación contará con una serie de pantallas que explicaremos a continuación:

6.2.2 Inicio

Esta primera pantalla tiene la apariencia de la siguiente imagen:

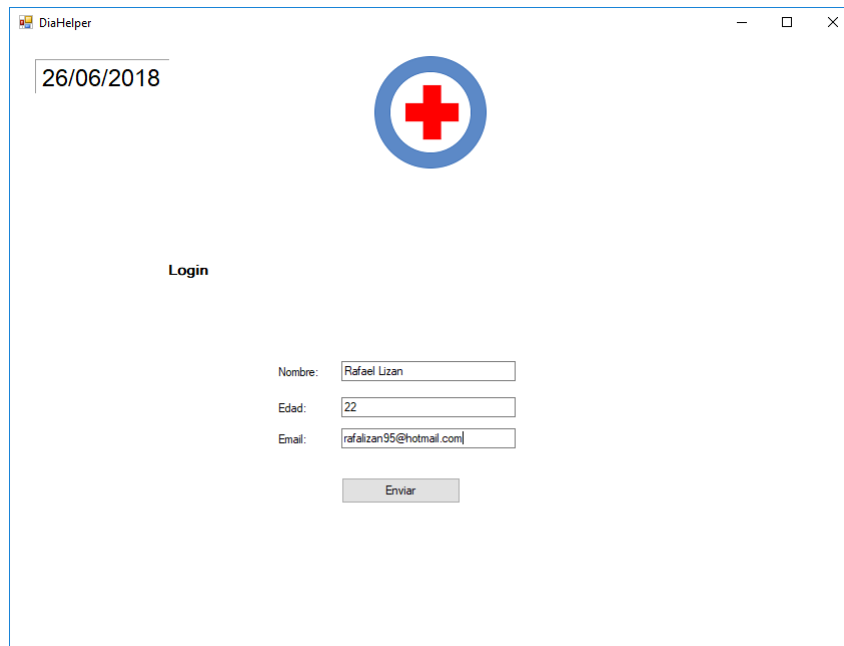


Figura 21: Pantalla de *Login* de la aplicación

Como se puede ver en la figura 21, en la pantalla de inicio el usuario tendrá que introducir sus datos personales (en este caso son: nombre, edad y email). Una vez ha cumplimentado con el formulario de inicio, el usuario procederá a navegar a la pantalla principal de la aplicación.

²El concepto de **lógica de negocio** es la funcionalidad que ofrece un programa informático. Se le llama así al procesamiento de los datos capturados por la aplicación y su posterior entrega al usuario.

Esto solo se pediría una vez, posteriormente los datos del usuario quedarían registrados y almacenados en la aplicación. Justo al completar el formulario que se pide, el usuario navegará a la pantalla de *Inicio* que será la pantalla principal por la cual se moverá el usuario.

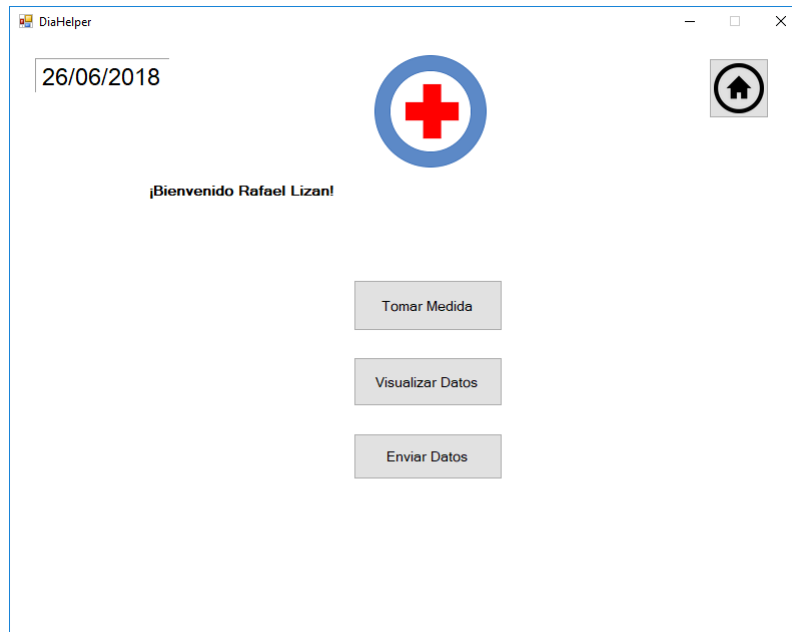


Figura 22: Pantalla de *Inicio* de la aplicación

Como se puede ver en la imagen superior, vemos que esta pantalla tiene las tres acciones principales que el usuario necesita para hacer la medida de la temperatura. Existen 3 botones: *Tomar medida*, *Visualizar datos* y *Enviar datos*. Se puede apreciar que esta pantalla tiene como propósito facilitar la navegación del usuario de cara a que sea aun más sencillo el proceso de que el paciente pueda autodiagnosticarse. A continuación vamos a ver las distintas vistas que se muestran cuando accionas uno de los tres botones principales de la aplicación.

6.2.3 Toma de datos

Esta es la pantalla desde la cual el usuario podrá realizar la toma de temperatura.

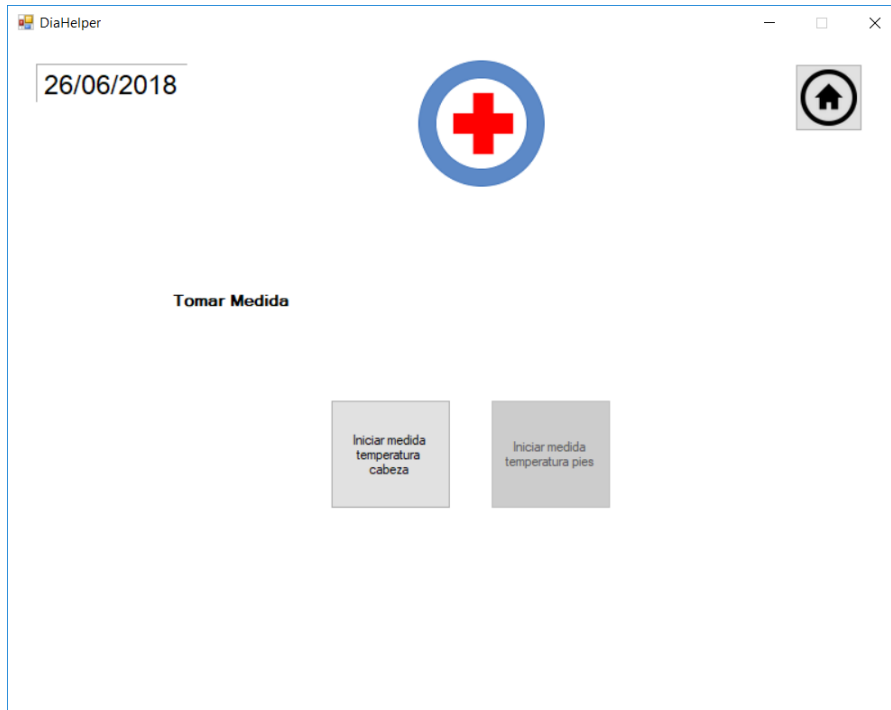


Figura 23: Pantalla de *Toma de datos* de la aplicación

Como se puede apreciar en la figura 23, el usuario deberá realizarse al menos 1 medida de temperatura de la cabeza previamente a realizar la de los pies. Una vez el usuario haya realizado una medida de temperatura de la cabeza, podrá realizar las medidas de la temperatura de los pies que quiera, para así poder tener un histórico de temperaturas del paciente y detectar anomalías si las hubiera.

Cuando el usuario accione uno de los dos botones, se mostrarán las indicaciones que debe seguir desde la misma pantalla. El usuario no deberá interactuar con la aplicación de nuevo hasta que no termine de realizarse la toma de temperatura (tardará en finalizar aproximadamente entre 3 a 4 segundos). Con esto damos indicaciones a los pacientes y evitamos posibles errores que puedan haber a la hora de la medición (conseguimos reducir el error creado por el factor humano). Una vez se haya realizado cualquier medida de la temperatura, los datos se almacenarán y al usuario le mostrará un mensaje diciendo que la toma de datos ha sido satisfactoria.

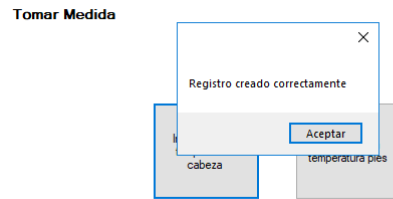


Figura 24: Mensaje de confirmación tras la medida satisfactoria

El usuario podrá realizar tantas tomas de temperatura como veces crea que son necesarias. No limitamos al usuario ya que, aunque podemos detectar si está habiendo una toma de temperatura incorrecta (considerando un rango de temperaturas válidas entre 27 y 32 °C para temperaturas de los pies y entre 35 y 40 °C para temperaturas de referencia), puede estar inseguro de que la medida haya sido correcta. Con todas las medidas validas del día se realizará una media para guardar la medida de ese día.

6.2.4 Visualización de los datos

Esta es la pantalla desde la cual el usuario podrá visualizar registros anteriores.

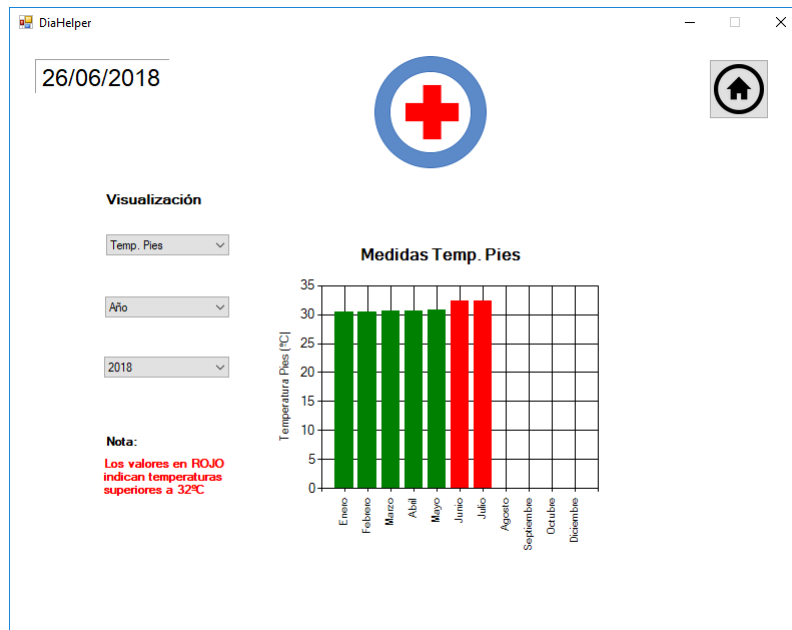


Figura 25: Pantalla de *Visualización de datos* de la aplicación

Esta pantalla contará con una manera gráfica de visualizar los datos. Es posible que los usuarios no sepan interpretar los gráficos correctamente, por lo que la visualización de ellos se hace de la manera más sencilla posible haciendo uso de colores. Los colores de gama roja indicando que podría haber algún peligro, y los de gama verde indicando estabilidad. También se muestra una nota en la que dice que los valores en rojo son las temperaturas superiores a 32°C para mayor información de los usuarios de la aplicación.

También podemos seleccionar el rango en el cual queremos mostrar los valores, si para el año 2018, o para algún mes del año en particular. Esto permite más flexibilidad a la hora de interpretar los datos.

6.2.5 Envío de los datos

Esta es la pantalla desde la cual se pueden enviar los datos al especialista y a un servidor *FTP* con el que estará conectada la aplicación (siempre y cuando exista una conexión a internet).

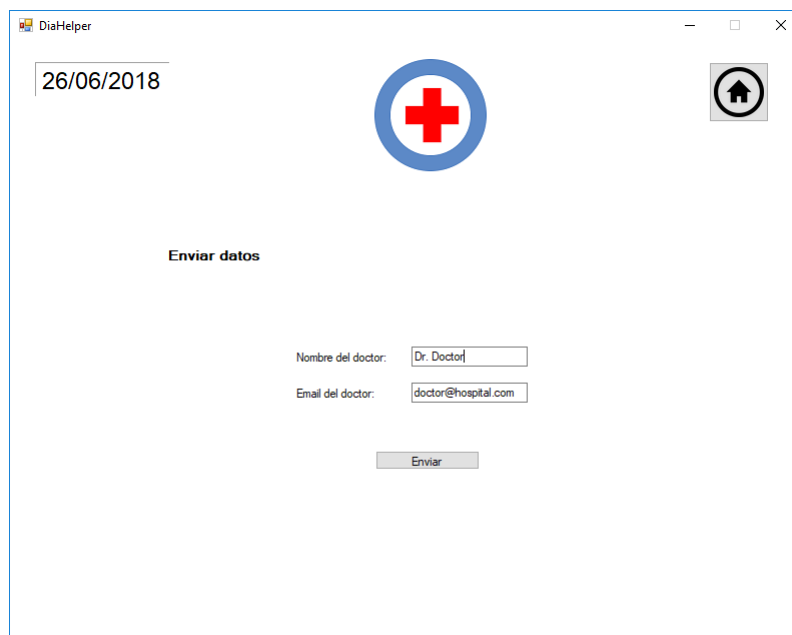


Figura 26: Pantalla de *Envío de los datos* de la aplicación

Esta aplicación no cuenta con muchos campos, como se puede observar en la imagen anterior, el usuario puede decidir a que *email* enviar los datos y el nombre del especialista al cual van dirigidos. Como se ha explicado anteriormente, el programa ya tendrá configurada la dirección del servidor FTP sobre el cual se dejarán los datos del paciente (en este caso se ha utilizado un FTP local, usando la aplicación *IIS Manager* de Windows).

6.3 Lógica de la aplicación

Hasta aquí la explicación de la parte visual de la aplicación; las vistas, pantallas y funcionalidades. A continuación se va a explicar el funcionamiento de las mismas visto desde una perspectiva de la lógica de la aplicación. Seguiremos el mismo esquema que el anterior, comentando pantalla por pantalla lo que conlleva por detrás.

6.3.1 Inicio

Esta pantalla no tiene prácticamente lógica de negocio, el usuario inserta una serie de campos que pasan una validación de formato. Las validaciones son: el campo edad tiene que ser un campo numérico de máximo 3 caracteres, y el nombre tiene que tener caracteres válidos (de la *a* a la *z*). Una vez apretamos al botón de envío del formulario lo que ocurrirá será que insertará un nuevo registro en la base de datos local. Con esto conseguimos tener un registro del usuario. Una vez este registro ya exista, por el propósito de la aplicación, recogeremos la información sin tener que pedírsela al usuario cada vez que intente acceder a la aplicación.

Cuando enviamos el formulario automáticamente aparecemos en la pantalla de inicio. Como se ha visto anteriormente, esta pantalla solo dispone de los 3 botones principales, por lo que no hay ninguna lógica aparente. Sólo existe la navegación entre las pantallas.

6.3.2 Toma de datos

Dentro de esta pantalla contamos con dos botones. Estos dos botones realizan medidas de temperatura. El botón de **tomar medida de la cabeza** es el que apretaremos para recopilar la información de la temperatura ambiente corporal del paciente. La lógica que hay detrás de este primer botón es similar a la del segundo. Aquí entra en juego los scripts de *Python* que harán que se recoja la información de uno/o varios sensores que hay conectados a la *Raspberry Pi*. Los sensores se conectan en serie a través de los pines **03** y **05** de la *Raspberry Pi*, estos se corresponden a los de **SDA** y **SCL** respectivamente. Los sensores que usamos en esta solución se conectan a través de estos pines que son del tipo I2C. Este protocolo ya se ha visto en un apartado anterior.

El segundo botón, **tomar medida del pie**, es el que apretaremos para recoger la información de la temperatura del pie del paciente. En este proceso intervendrán 12 sensores que estarán conectados en serie y que con los scripts que se ejecutan, elegiremos las direcciones de los sensores que queremos que estén activos y hagan una transferencia de datos. En el caso del primer botón con la lectura del sensor destinado para ese fin es suficiente (indicándole el programa de que sensor tiene que leer). Para este segundo botón el script detecta que pie es el que está actualmente siendo medido ya que habrá dos sensores que no estén dentro del rango del pie y tendrán el mismo valor que el de la temperatura ambiente.

Mediante el código de *Python* realizamos medidas tanto de la temperatura ambiente como la del objeto a la que está apuntando el sensor, por lo que la detección de una temperatura válida del pie se realizará con una validación de umbral que podemos desglosar en lo siguiente: Si $T_{amb} = T_{obj}$ significa que el sensor no está tapado. Con esto y sabiendo las direcciones de los sensores se averigua que pie es sobre el cual se están realizando las medidas.

Ya hemos visto la manera de recoger los datos con los sensores y el protocolo que se va a usar. Estos datos serán interpretados por la *Raspberry Pi* que, junto con nuestro código, interpretará los datos como la temperatura del objeto próximo al que estarán apuntando. Como hemos visto, este protocolo permite tener varios *esclavos* conectados en serie para la transmisión de datos. Este será el caso de nuestros sensores que serán conectados en serie por las líneas de SDA y SCL para poder transmitir los datos.

Vista ya la parte hardware, las mediciones de temperatura serán ejecutadas a través de un script de python que se ha creado usando una librería llamada ‘**micro-python**’ [18] que tiene una funcionalidad específica para usarla cuando se quieren realizar conexiones por I2C. También vamos a usar una librería que permite usar funciones preparadas para ser implementadas con los sensores MLX90614 (que serán los que vamos a usar. Hablaremos más acerca de los distintos elementos que vamos a emplear en la siguiente sección) que se llama ‘**python-MLX90614**’ [19].

Una vez se recogen los datos con el script de python, lo que se hará será almacenar la información de ese día y paciente en un archivo .csv (extensión parecida al formato .xlsx de Microsoft Excel). Con esto lo que conseguimos es tener en local una copia del registro de los datos del paciente. Esto lo tendremos tanto como para la temperatura de los pies como para la temperatura de la cabeza. El fichero será nombrado igual que el nombre y edad que proporcione el paciente, juntando los nombres, apellidos y edad por barra bajas y añadiendo al final la palabra “Temperaturas”. (Ejemplo: Rafael_Lizan_Ortiz_22_Temperaturas). El formato que tendrá el .csv es el siguiente:

Ejemplo de formato

Fecha	Temperatura Referencia	Temperatura Pies
22/04/2018	36.8	31.2
23/04/2018	36.0	31.3
24/04/2018	36.2	31.0
...

6.3.3 Visualización de los datos

En la pantalla de visualización de datos se podrá ver el registro que se ha ido tomando de todos los datos durante el tiempo. El usuario tendrá un desplegable del cual podrá seleccionar si quiere visualizar los datos para un día en particular, un mes o un año, y por otra parte tendrá un segundo desplegable que dependa de este primero donde le aparecerá las opciones según lo escogido (i.e. si elige mes el segundo desplegable tendrá las opciones de *enero, febrero, marzo, abril...*).

Lo que esto significará es, que por detrás, está habiendo una lectura de los datos del archivo *.csv* de ese paciente y, en función de lo escogido en el primer desplegable, mostraremos unas opciones u otras. Esto conllevará a mostrar los datos comprendidos entre el rango de fechas indicado por el usuario. El gráfico tendrá dos ejes: El eje de abscisas tendrá la etiqueta de la fecha, que podrá ser días o meses, mientras que el eje de ordenadas tendrá la etiqueta del valor de temperatura en grados centígrados (°C).

6.3.4 Envío de los datos

Por último queda la pantalla de envío de datos. Esta pantalla tendrá varias funcionalidades. La primera, y única que verá el usuario, será la de envío de estos datos a un médico. Este envío de datos se realizará por email usando el protocolo de SMTP (Simple Mail Transfer Protocol). El usuario proporcionará el nombre del médico así como el del hospital. Facilitará también la dirección de correo electrónico de el médico para que pueda realizarse la transferencia del archivo. Este archivo irá como adjunto en el email que se envíe.

Se montará el email para que quede de la siguiente manera (ejemplo):

De: usuario@usuario.com
Para: usuario@usuario.com
Asunto: Archivo temperatura pies paciente: *NOMBRE_PACIENTE*
Adjunto: *ARCHIVO_CSV*

A la atención de *NOMBRE_MEDICO*, profesional perteneciente al *NOMBRE_HOSPITAL*. Acaba de recibir el archivo *ARCHIVO_CSV* que contiene la información de carácter personal, acerca de la temperatura de los pies y corporal del paciente *NOMBRE_PACIENTE*.

De esta manera el médico conseguiría el archivo de una manera rápida y sencilla.

Cuando se pulse al botón *Enviar* de esta pantalla, a la vez que enviamos el email, también se está guardando una copia de la base de datos del paciente en un servidor *FTP* (File Transfer Protocol). Con esto lo que conseguimos es tener siempre una copia en la nube en caso de catástrofe y pérdida de los datos, ya que perder el histórico de datos de un paciente puede tener una gran repercusión de manera negativa. Haciendo esto evitamos pérdidas de archivos y un mantenimiento de los mismos.

7 Manual del usuario

En este capítulo explicaremos como es el flujo que sigue la aplicación y las principales funcionalidades que un usuario debe realizar por nuestra aplicación.

7.1 Inicio de sesión del usuario

La primera pantalla que verá el usuario es la de *Login* como hemos visto en la figura 21. El usuario deberá introducir su *nombre*, *edad* y *email* para poder continuar, de lo contrario se mostrará un mensaje de error informando al usuario de que no podrá continuar.

7.2 Inicio

Una vez haya realizado el proceso de *Login*, el usuario navegará automáticamente en la pantalla de *Inicio*, que tiene el aspecto de la figura 22. Una vez en esta pantalla, el usuario encontrará los tres botones principales que muestra la figura y que previamente hemos comentado.

7.3 Toma de datos

Si el usuario decide pulsar el botón de *Tomar Medida* desde la pantalla de *Inicio*, el usuario navegará a la pantalla de *Toma de datos* que es la que se ha mostrado en la figura 23. Una vez ahí el usuario podrá apretar cualquiera de los dos botones para realizar una medida de temperatura, tanto de referencia como de los pies. Teniendo en cuenta lo explicado anteriormente, el usuario deberá realizarse una medida de la temperatura de referencia previamente a la de los pies.

Una vez hechas las medidas pertinentes, el usuario solo tiene dos opciones de navegación: o bien pulsa el botón de *Home*, situado en la esquina superior derecha y representado con una casa, o pulsar el botón de salida de la aplicación.

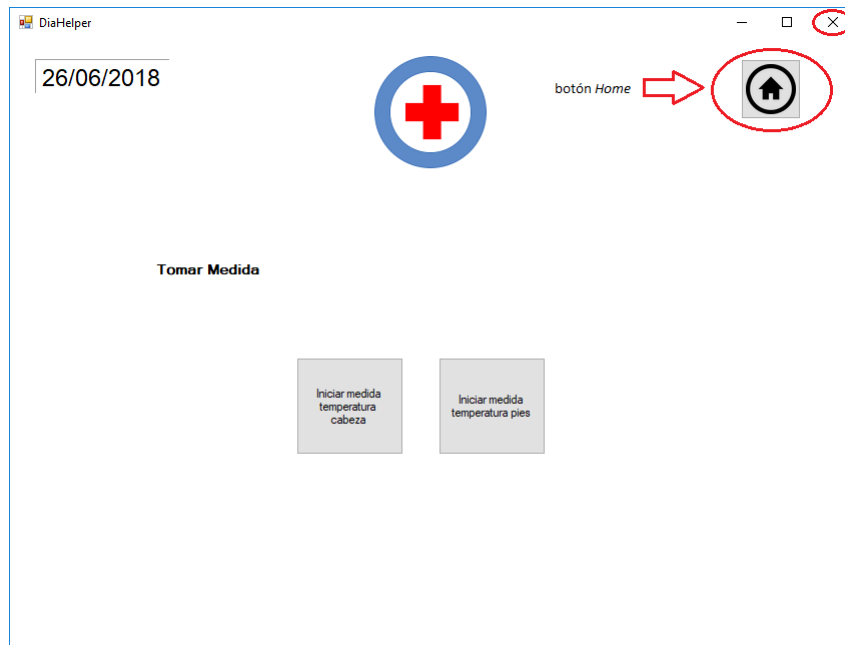


Figura 27: Posibles salidas de la pantalla de *Toma de datos*

7.4 Envío de datos

Una vez el usuario halla pulsado el botón de *Enviar Datos* desde la pantalla de inicio, navegará a la pantalla de *Envío de datos* que corresponde con la figura 26. En esta pantalla el usuario tiene la opción de escribir a quien se le van a enviar los datos así como indicar su nombre. En caso de no haber sido completada la transmisión del mensaje cuando el usuario pulse el botón de *Enviar* dentro de esta pantalla, se le mostrará un mensaje de error.

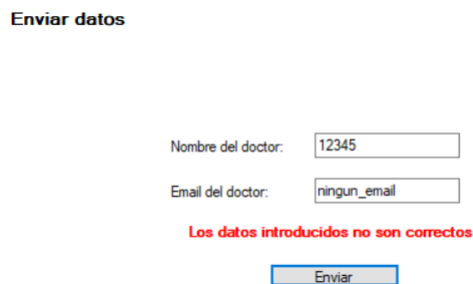


Figura 28: Mensaje de error al enviar los datos

En caso de que el envío de los dato sea satisfactorio se le mostrará un mensaje de confirmación al usuario.

8 Manual del desarrollador

En este penúltimo capítulo, vamos a describir la información necesaria para que cualquier desarrollador pueda implementar la aplicación y expandir sus funcionalidades a nivel técnico.

En este manual, se va a suponer que el desarrollador tiene unos conocimientos previos de *Python*, programación con la *Raspberry Pi* así como conocimientos de *C#* y aplicaciones de tipo *WindowsForms* de *.NET*. Trabajaremos para *.NET* con el entorno visual (IDE) de *Visual Studio 2017* y para *Python*, usaremos el IDE de *Spyder*.

8.1 Introducción

En este manual del desarrollador vamos a proporcionar la suficiente información como para que cualquiera que esté capacitado para trabajar con las herramientas y entornos mencionados, pueda hacerlo. Dividiremos el manual en 3 secciones:

- **Preparación del proyecto:** En esta primera sección mostraremos como se crea un proyecto tanto en Visual Studio y en Spyder. También se ilustrará la organización de las carpetas y la estructura que tiene nuestro sistema.
- **Scripts de Python:** En esta sección vamos a describir de manera técnica como funciona la solución y que scripts de Python estamos utilizando, así como la integración con las librerías ya mencionadas. Se explicará en un poco más de detalle que es lo que está realizando el código.
- **Aplicación en .NET:** Por último se explicará como está creada la aplicación de *.NET* y como cualquier desarrollador puede ampliar las funcionalidades (o modificar las ya existentes) en el entorno de *Visual Studio 2017*.

8.2 Preparación del proyecto

Visual Studio

Para poder crear un proyecto de *Visual Studio* que se acerque a las necesidades que tiene la solución actual, debemos crear un nuevo proyecto en el mismo IDE. Se mostrará una ventana para seleccionar el tipo de proyecto que queremos crear y seleccionamos nueva Aplicación de *Windows Forms (.NET Framework)*.

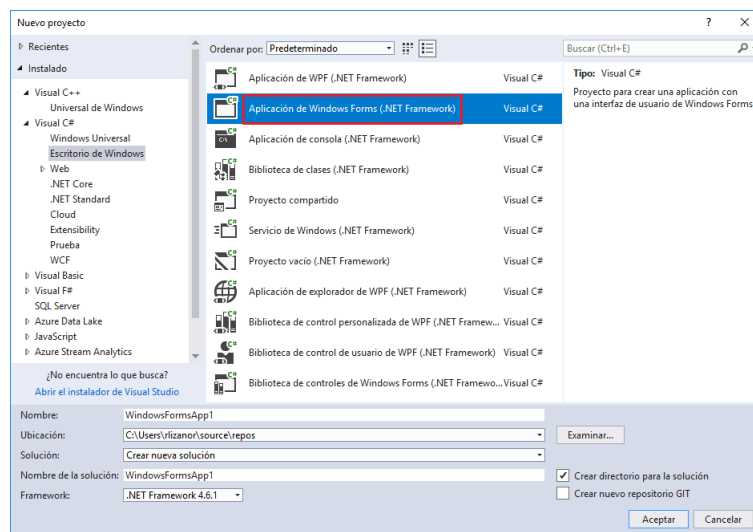


Figura 30: Aplicación de Windows Forms

Si no se quiere crear uno nuevo, también se puede modificar el ya existente. En el proyecto ya existente, nos encontramos con la siguiente estructura de ficheros.

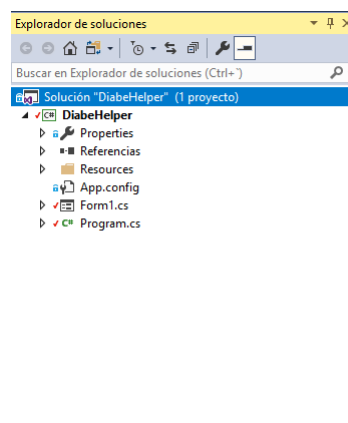


Figura 31: Estructura de carpetas

Donde tenemos un único formulario de tipo *Windows Forms* (Form1.cs) sobre el que realizaremos nuestra interfaz gráfica, y un programa principal (Program.cs) donde estará la lógica de la aplicación.

Spyder

Este es el IDE que usamos para crear archivos en *Python*, es una herramienta que se instala cuando instalamos el paquete de *conda* que sirve para editar, crear y visualizar archivos de *Python*. Para crear un nuevo fichero tan solo tenemos que darle a *crear nuevo archivo*. En *Python* no existe ningún tipo de proyecto en específico que necesitemos crear, crearemos los archivos *.py* que sean necesarios para completar el desarrollo.

Esto es toda la herramienta que necesitamos para realizar los scripts de *Python*.

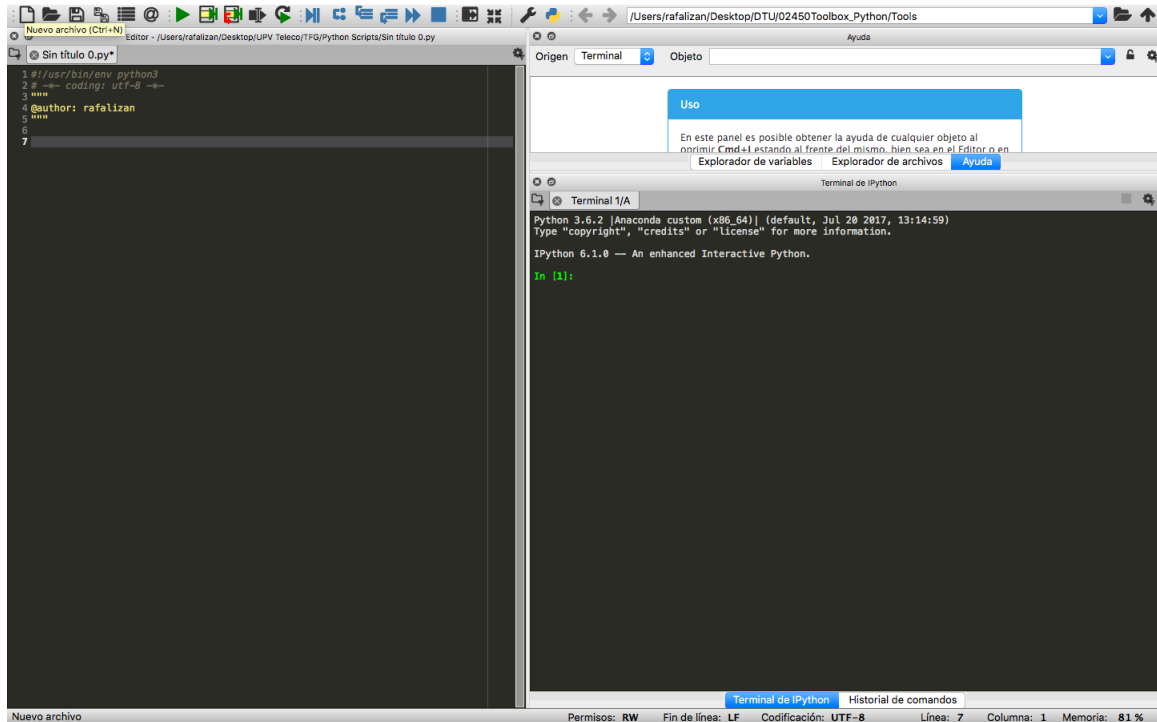


Figura 32: Vista de Spyder y archivo en blanco

Utilizamos unas librerías que ya hemos mencionado antes, las de **micro-python** [9] y la de **python-mlx90614** [10]. Las tenemos referenciadas por ruta, por lo que en la misma ruta donde tenemos nuestro *main.py* tenemos también las dos carpetas de las librerías a las que hacemos referencia desde el *main.py*.

Habiendo visto la preparación del proyecto podemos ver también la arquitectura que tiene este prototipo. Tenemos la parte de la interfaz de usuario realizada en *.NET*, el acceso a base de datos y la interpretación de los datos a través de *.NET* y la toma de los datos en *Python*. La arquitectura sería la siguiente:

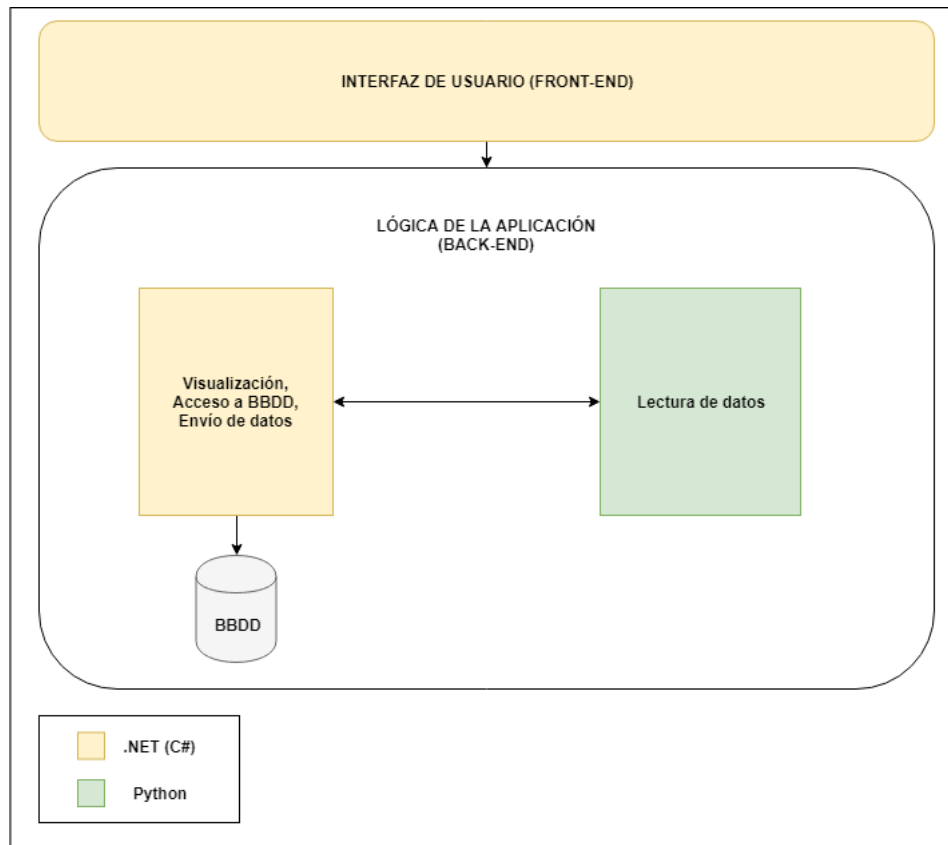


Figura 33: Arquitectura de la aplicación

8.3 Scripts de Python

Como se ha comentado anteriormente se usan librerías externas que necesitan ser referenciadas en el código de python. El código principal tiene 3 partes: importar librerías, definición de las direcciones físicas de los sensores y la lectura de temperatura por los mismos.

Para saber que direcciones tienen los sensores dentro de la *Raspberry Pi* hay que ejecutar el comando `i2cdetect -y 1` y nos aparecerá una matriz con las direcciones de los sensores que tenemos conectados a la *Raspberry Pi*.

```

      0 1 2 3 4 5 6 7 8 9 a b c d e f
00:  ---
10:  ---
20:  ---
30:  ---
40:  ---
50:  --- 5a ---
60:  ---
70:  ---

```

Figura 34: Dirección de un sensor conectado a la *Raspberry Pi*

Al definir las direcciones en el código de Python habrá que crear los objetos de las direcciones, y posteriormente habrá que llamar a la librería de **mlx90614** y crear los objetos de ese tipo. Los sensores tendrán 2 atributos, un *get_amb_temp()* y el *get_obj_temp()*. Estos serán los que usemos para comprobar que pie es el que se está midiendo y cual es la temperatura del pie dependiendo de los puntos.

8.4 Aplicación en .NET

La aplicación está construida fundamentalmente utilizando C#. Sabemos que la parte visual es controlada por las vistas de *Windows Forms* y por los controladores que se asocian a los elementos de la vista. El acceso a base de datos se hará mediante C# dentro de la lógica de la aplicación. Existen diversas aplicaciones y lenguajes para manejar bases de datos. Para la base de datos lo que utilizamos en este desarrollo es un paquete llamado *SQLite*. Este paquete permite crear bases de datos de manera local en un archivo con extensión *.db* y ejecutar sentencias de *SQL* con una sintaxis parecida.

Para poder visualizar lo que existe dentro de una tabla (que pertenezca a una base de datos) usaremos el programa de *SQLite Manager*, pero cualquier programa que pueda leer archivos *.db* y ejecutar sentencias *SQL* podría servirnos. Un ejemplo de la tabla que usaremos es el que vemos en la figura siguiente:

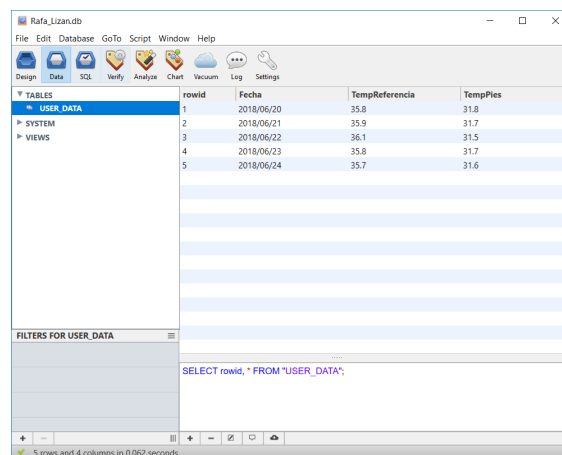


Figura 35: Ejemplo de estructura y datos en base de datos

9 Conclusiones

En este último capítulo vamos a describir los objetivos que hemos conseguido alcanzar a lo largo del trabajo, contratiempos que han surgido durante el desarrollo y futuras vías de desarrollo del proyecto. También incluiremos las conclusiones finales tras el desarrollo del proyecto.

9.1 Contratiempos/ problemas encontrados

Durante el desarrollo del proyecto no todo ha salido como se esperaba. Han surgido diversos problemas relacionados tanto con el programa informático como con el hardware del montaje. A continuación vamos a exponer algunos contratiempos que hemos encontrado y como los hemos solucionado.

- **Creación de una aplicación para utilizar el montaje de la *Raspberry Pi*:** La *Raspberry Pi* tiene instalado el sistema operativo de *Raspbian* y el prototipo requería tener una interfaz de usuario para poder interactuar con el mismo, por lo que en una primera instancia se pensaba hacer el programa informático en el lenguaje de *Python* para que pudiera ser ejecutada en un OS estilo Linux. Sin embargo esto fue bastante costoso, ya que no existen muchas librerías en Python para poder realizar interfaces gráficas, por lo que se optó por realizar una aplicación en *.NET* y exportarla en un fichero *.exe*, el cual puede ser ejecutado en la *Raspberry Pi* con unas pequeñas modificaciones.
- **Sensores a emplear en el dispositivo:** Ha sido un rompecabezas el averiguar que tipo de sensores utilizar para la *Raspberry Pi* que midieran temperatura y que fueran compatibles con lectura en serie. Al principio se pensó en usar sensores RTD, pero tras analizar los diversos sensores mencionados en el capítulo 5 escogimos los de tipo infrarrojos. Una vez con los infrarrojos, solo disponíamos de 2 pines en la *Raspberry Pi* para poder conectar los 13 sensores, por lo que se buscaron unos en particular que utilizaran el protocolo I2C para poder hacer la lectura secuencial de todos ellos.
- **Interfaz de usuario para usar el dispositivo:** Al inicio del desarrollo de este proyecto no se tenían conocimientos de *.NET* ni de la herramienta de trabajo *Visual Studio*, por lo que en un principio fue difícil adaptarse a un nuevo lenguaje y a sacarle el potencial que tiene a la herramienta de desarrollo. Aproveché que estaba realizando las prácticas en empresa empleando esta herramienta y lenguaje y, después de leer manuales y aprender a utilizarla, ya se pudo desarrollar un programa informático que pudiera ofrecer una funcionalidad mínima para poder crear el prototipo de la aplicación.

9.2 Futuras vías del trabajo

El objetivo de realizar un dispositivo que pudiera cubrir la necesidad de unos pacientes a la hora de mejorar su calidad de vida era un reto ambicioso, pero a la vez entusiasmante. En una primera instancia se quería realizar un producto final con el cual los pacientes pudiera medir la temperatura en casa, autodiagnosticarse y que los datos fueran enviados automáticamente a los doctores y servidores. El potencial y magnitud del desarrollo que tiene esta idea es enorme, por lo que tuvimos que realizar un primer paso, un prototipo, para que en el futuro, pueda desarrollarse esta idea en un producto final.

Se ha conseguido realizar un prototipo del dispositivo, siendo este funcional para los usuarios. Este proyecto tiene diversas futuras vías de trabajo, y vamos a analizar las mejoras correspondientes a dos partes, la programa informático y hardware.

- **Hardware:** Se ha utilizado una modelo de la *Raspberry Pi* muy potente, quizá más potente de lo que se requiere en este desarrollo, por lo que se puede adaptar el desarrollo a una placa más sencilla o incluso a una personalizada que contenga únicamente los componentes que sean estrictamente necesarios. El cableado, y la pantalla donde se visualizan los datos podría cambiar, ya que se podría pensar en un dispositivo que tuviera una batería extraíble que pudiera cargarse a la luz. El mismo dispositivo podría también tener una pantalla táctil por la cual el usuario interactuaría con la aplicación. Todo esto con el empaquetado de un producto final.
- **Programa informático:** La aplicación que hemos creado en .NET cumple con toda la funcionalidad que se ha querido implementar y que cubre con las necesidades que exponíamos en el apartado de objetivos del proyecto. No obstante, la aplicación podría estar mejor hecha visualmente, conectarse a una base de datos que estuviera en un servidor remoto y no en local, y hacer el envío de datos únicamente por FTP y que el doctor se conectara a una plataforma de internet para descargarse un informe.

Con esto también se puede pensar en una aplicación para móvil y que el dispositivo no disponga de ninguna pantalla y que todo el control sea a través del móvil. Este se podría conectar a un servidor para compartir los datos y contar con una plataforma de internet también para la visualización de datos por usuario.

9.3 Conclusión final

Tras la finalización de este proyecto final de grado podemos sacar diversas conclusiones:

En un primer lugar, se ha alcanzado el objetivo principal del trabajo mediante la creación de un dispositivo electrónico que consiste en: un ordenador de bolsillo (*Raspberry Pi*) y 13 sensores conectados a él. También se ha conseguido realizar una aplicación de escritorio de *Windows* que se conecta al dispositivo para transmitir información junto con una serie de funcionalidades.

Por otra parte, también ha sido realmente reconfortante el poder aplicar y asentar conocimientos adquiridos en electrónica, telemática y programación. Todo esto aplicado a un fin muy emotivo como puede ser el de mejorar un proceso para gente que tiene un problema de salud y saber que puede tener un impacto real en la sociedad.

Como se ha mencionado anteriormente era un objetivo ambicioso, el crear un prototipo de dispositivo que pudiera servir para ayudar a pacientes diabéticos con complicaciones en su enfermedad para poder llevar un seguimiento de su estado de la enfermedad mediante un indicador como es el de la temperatura de los pies. Todo esto ha sido posible gracias a modificaciones creadas sobre un dispositivo electrónico de tamaño reducido y con un muy bajo coste.

Se ha mejorado un proceso existente en la sociedad. El paciente de una complicación de la diabetes puede monitorizar su progreso desde su propia vivienda, sin necesidad de acudir y depender de un centro médico con todos los ahorros tanto de tiempo, esfuerzo y económicos que significan tanto para el paciente como para el propio centro médico, logrando así los objetivos planteados al inicio de este proyecto.

Bibliografía

- [1] Úlceras.net. (2016). Estudio de la OMS sobre Diabetes. Lugar de publicación: <https://www.ulceras.net/monograficos/83/66/pie-diabetico-epidemiologia.html>
- [2] Organización Mundial de la Salud. (2014). Informe mundial sobre la diabetes. Lugar de publicación: http://apps.who.int/iris/bitstream/handle/10665/204877/WHO_NMH_NVI_16.3_spa.pdf;jsessionid=90C6E0E210758D7D16647387525B5A68?sequence=1
- [3] Observatorio mundial de la salud - Organización Mundial de la Salud. (2008). Causes of death in 2008. Lugar de publicación: <http://www.who.int/features/qa/18/es/>
- [4] Arduino: Tecnología para todos. Historia. Lugar de publicación: <https://arduinodevics.weebly.com/historia.html>
- [5] Blog: Historia de la Informática. Museo informática. Escuela Técnica Superior de Ingeniería Informática. Universidad Politécnica de Valencia. (2013). Raspberry Pi. Lugar de publicación: <http://histinf.blogs.upv.es/2013/12/18/raspberry-pi/>
- [6] Easton LaChapelle. Unlimited Tomorrow. (2015). Lugar de publicación: <http://theroboarm.com/>
- [7] INE. Informe sobre el sector primario en España en 2017. (2017). Tabla de la estructura de la población activa española.
- [8] Organización Mundial de la Salud. (2014). Informe mundial sobre la diabetes. Lugar de publicación: http://apps.who.int/iris/bitstream/handle/10665/204877/WHO_NMH_NVI_16.3_spa.pdf;jsessionid=90C6E0E210758D7D16647387525B5A68?sequence=1
- [9] Sociedad Española de Diabetes. (2011). Estudio di@bet.es. Lugar de publicación: http://www.ciberdem.org/media/434050/estudio-diabetes_ciberdem_diapositiva_oficial.pdf
- [10] CIBERDEM (Centro de Investigación Biomédica en Red de Diabetes y Enfermedades Metabólicas Asociadas), el Instituto de Salud Carlos III (Ministerio de Ciencia e Innovación), en colaboración con la Sociedad Española de Diabetes (SED) y la Federación Española de Diabetes (FED). (2011). La Diabetes en España. Lugar de publicación: <http://www.fundaciondiabetes.org/prensa/297/la-diabetes-en-espana>. Fundación para la Diabetes.
- [11] Bakker K, Apelqvist J, Shapper NC. Practical guidelines on the management and prevention of the diabetic foot. (2011). Diabetes Metab Res Rev. 2012; 28 (suppl 1):225-231. DOI: 10.1002/dmrr.2253.

- [12] Bakker K, Schaper NC. The development of global consensus guidelines on the management and prevention of the diabetic foot. (2011). *Diabetes Metab Res Rev.* 2012; 28 (suppl 1):116-118. DOI: 10.1002/dmrr.2254.
- [13] Esther Chicharro-Luna, Ingrid Portabales Botana, Laura Ramírez Monreal, Esther Torrent Ivorra. (2014). Monitorización de la temperatura del pie como herramienta en la neuropatía diabética *Rev. 2015. Revista Internacional de Ciencias Podológicas.*
- [14] Sensor infrarrojo. (2018). Lugar de publicación: https://es.wikipedia.org/wiki/Sensor_infrarrojo
- [15] Omega a spectris company. Termopar. (2018). Lugar de publicación: <https://es.omega.com/prodinfo/termopares.html>
- [16] I2C. (2018). Lugar de publicación: <https://es.wikipedia.org/wiki/I%C2%B2C>.
- [17] Mike Hord. (2013). I2C. Lugar de publicación: <https://learn.sparkfun.com/tutorials/i2c>.
- [18] Damien P. George. (2014). Micropython. Librería de github. Lugar de publicación: <https://github.com/micropython/micropython>
- [19] Arsenijs. (2016). Python-MLX90614. Librería de github. Lugar de publicación: <https://github.com/CRImier/python-MLX90614>