



UNIVERSIDAD PERUANA  
**CAYETANO HEREDIA**  
ESCUELA DE POSGRADO

**DISEÑO Y EVALUACIÓN DE UN  
SISTEMA DE AYUDA AL  
DIAGNÓSTICO PARA NEUROPATÍA  
DIABÉTICA BASADO EN LA LECTURA  
DE PUNTOS DE PRESIÓN PLANTAR Y  
MACHINE LEARNING**

TESIS PARA OPTAR EL GRADO DE  
MAESTRO EN INFORMÁTICA  
BIOMÉDICA EN SALUD GLOBAL CON  
MENCIÓN EN INFORMÁTICA EN SALUD

JEAN PIERRE TINCOPA FLORES

LIMA-PERÚ

2019



**Asesor**

Dr. Germán Málaga

## **Dedicatoria**

*A todas aquellas personas que trabajan cada día buscando desarrollar nuevas tecnologías en pro del bienestar de nuestra sociedad, que lo agreste del panorama no les quite la esperanza de un futuro mejor.*

## **Agradecimientos**

Agradezco al Dr. Germán Málaga por su asesoría y guía durante el desarrollo de esta tesis y su confianza en mi trabajo.

Al Dr. Alejandro Dioses por su importante aporte en el diagnóstico de neuropatía mediante la electromiografía

A la Dra. Elsa Neira, por su apoyo en el reclutamiento en las dos fases del estudio.

A los investigadores de la Unidad de Conocimiento y Evidencia (CONEVID) por su apoyo y por brindarme un espacio para poder llevar a cabo este proyecto.

A la Magister bióloga Dulce Esperanza Alarcón Yaquetto por su acompañamiento en todas las etapas de desarrollo de este trabajo y su invaluable aporte en mi desarrollo profesional y personal.

## **Fuentes de Financiamiento**

Esta investigación fue financiada con un aporte de FONDECYT.

## Resumen

La neuropatía diabética es la complicación más común en pacientes que sufren diabetes. La pérdida en la sensibilidad del pie hace que estos pacientes tengan un mayor riesgo de ulceraciones que finalmente devienen en amputaciones. Se conoce que la presión plantar aumenta en los pacientes diabéticos como resultado de una deformación del pie a causa de la enfermedad. El *gold standard* del diagnóstico de neuropatía diabética son pruebas de conducción nerviosa costosas e invasivas y difíciles de implementar en los primeros niveles de atención. Este proyecto desarrolló y evaluó un sistema de apoyo al diagnóstico de neuropatía diabética a través de la creación de una base de datos de puntos de presión plantar usando una plantilla con sensores electrónicos y algoritmos de *machine learning*. El algoritmo diseñado logró tener una precisión promedio de 0.94 y un *recall* promedio de 0.93. El sistema de apoyo al diagnóstico tuvo gran aceptación por parte de endocrinólogos quienes afirmaron les gustaría usar con más frecuencia el sistema por ser fácil de usar, cómodo y confiable. Este es el primer estudio donde se usa la medición de presiones plantares en tiempo real como base para un sistema de apoyo al diagnóstico.

Palabras clave: presión plantar, sistema de apoyo al diagnóstico, sensores, *machine learning*, *support vector machine*, neuropatía diabética

## **Abstract**

Diabetic neuropathy is the most frequent complication that diabetic patients have to face. The loss of sensitivity in the lower limbs increases the risk of foot ulcerations and eventually amputations. It is known that plantar pressure is increased in certain areas in diabetic patients due to foot deformations as a result of the disease. The gold standard to diagnose diabetic neuropathy is nervous conducting tests that are expensive, invasive and hard to implement in the first level of attention. This project developed a decision support for diabetic neuropathy system based on *machine learning* through the creation of a database of plantar pressure using an insole with electronic sensors. The developed algorithm had an average precision of 0.94 and an average recall of 0.93. The developed decision support system had wide acceptance by the endocrinologists that tested it that stated they would like to use the system more frequently because it is easy, comfortable and reliable. This is the first study where the real time assessment of plantar pressure is used for a decision support system

Keywords: plantar pressure, decision support system, electronic sensors, *machine learning*, support vector system, diabetic neuropathy.



## Tabla de contenidos

1. Introducción.....	1
2. Marco teórico.....	3
2.1. La pandemia de la diabetes mellitus.....	3
2.2. La neuropatía diabética.....	4
2.2.1 Métodos de diagnóstico de neuropatía diabética.....	5
2.3 Presión plantar y su importancia clínica.....	6
2.3.1 Medición de la presión plantar .....	7
2.4 La cuarta Revolución Industrial y la inteligencia artificial .....	10
2.4.1 El Aprendizaje automático o <i>Machine Learning</i> .....	12
2.4.2. Lenguajes de entorno y algoritmos de búsqueda.....	14
2.4.3. Aplicaciones de la inteligencia artificial .....	15
2.4.4. La inteligencia artificial en salud.....	17
2.4.5. Sistemas de apoyo al diagnóstico .....	18
2.4.6. Tecnologías de comunicación inalámbrica .....	19
3. Justificación.....	21
4. Objetivos.....	22
5. Materiales y métodos.....	22
5.1. Diseño del Estudio.....	22
5.2. Población y muestra .....	22
5.3. Procedimientos y técnicas .....	23

5.3.1. Diseño de sistema de adquisición de datos.....	23
5.3.2. Recojo de datos sobre presión plantar .....	36
5.3.3. Elaboración del algoritmo .....	38
5.4 Consideraciones éticas.....	53
6. Resultados.....	54
6.1. Estudio observacional.....	54
6.2. Desarrollo del Modelo Predictivo .....	54
6.3. Desarrollo del Sistema de Apoyo al Diagnostico.....	56
6.4. Interfaz Gráfica de Usuario (GUI).....	56
6.5. Estudio de Usabilidad.....	59
7. Discusión .....	60
8. Conclusiones.....	63
8.1. Limitaciones .....	63
8.2. Sugerencias.....	63
9. Referencias .....	64

## Índice de tablas

Tabla 1. Descripción de parámetros de la base de datos .....	38
Tabla 2. Tabla de evaluaciones del algoritmo “Decision Tree” .....	41
Tabla 3. Tabla de evaluaciones del algoritmo Random Forest .....	44
Tabla 4. Tabla de evaluación del algoritmo k – nearest neighbors .....	48
Tabla 5. Tabla de evaluaciones del algoritmo Support Vector Machine.....	52
Tabla 6. Características generales de los pacientes reclutados según diagnóstico de neuropatía .....	54
Tabla 7. Resultados del mejor desempeño de los algoritmos.....	54

## Índice de figuras

Figura 1. Partes de un sensor <i>Force sensitive resistive</i> (FSR).....	8
Figura 2. Representación del efecto piezoeléctrico .....	9
Figura 3. Galgas Extensiométricas usadas en el proyecto.....	10
Figura 4. Enfoques de la inteligencia artificial.....	12
Figura 5. Matriz de confusión para la evaluación de un sistema de aprendizaje automático .....	13
Figura 6. Componentes del sistema de adquisición.....	24
Figura 7. Arduino Nano.....	25
Figura 8. Sensor de presión FSR-402.....	25
Figura 9. Módulo ADS1115 .....	26
Figura 10. Módulos Bluetooth.....	27
Figura 11. Amplificador operacional LM358.....	28
Figura 12. Diagrama de configuración para lectura de sensor .....	29
Figura 13. Diagrama de la adquisición de datos de dos sensores.....	30
Figura 14. Conexión entre Arduino Nano y ADS1115 .....	30
Figura 15. Diseño de la placa de circuito impreso .....	31
Figura 16. Batería de polímero de litio.....	32
Figura 17. Distribución de los sensores en la planta del pie.....	33
Figura 18. Módulo de adquisición colocado en el pie derecho .....	34
Figura 19. Módulo de recepción de datos.....	34
Figura 20. Circuito de adquisición colocado en los pies del paciente.....	37
Figura 21. Tasa de error vs. Valor de K. ....	47
Figura 22. Matriz de valores de C y gamma.....	51

Figura 23. Diagrama de comportamiento del algoritmo KNN con las variables del estudio.....	
Figura 24. Componentes del sistema de apoyo al diagnóstico desarrollado .....	56
Figura 25. Primer módulo de la interfaz gráfica.....	57
Figura 26. Segundo módulo de la interfaz gráfica.....	57
Figura 27. Tercer módulo de la interfaz gráfica .....	58
Figura 28. Cuarto módulo de la interfaz gráfica.....	58
Figura 29. Quinto módulo de la interfaz gráfica .....	58
Figura 30. Resultados de la prueba de usabilidad del sistema de apoyo al diagnóstico.	59

## **1. Introducción**

Una de las complicaciones más frecuentes que presentan los pacientes diabéticos es la neuropatía diabética (1) que se caracteriza por una pérdida progresiva de las fibras nerviosas produciendo sensaciones de hormigueo, ardor y dolor (2). Se estima que el 50% de pacientes diabéticos desarrollará algún tipo de neuropatía diabética en algún momento de su vida (3) siendo la poli neuropatía distal sensitivo-motora (DSPN) la forma más frecuente (4).

En sus inicios estas neuropatías suelen ser asintomáticas pudiendo estar por años presentes sin ser detectadas (5), por lo que es necesario un diagnóstico temprano para poder dar un seguimiento adecuado del paciente con DSPN. De lo contrario el riesgo de desarrollar ulceraciones en los pies que suelen derivar en amputaciones incrementa significativamente.

En años recientes, se ha buscado mejorar el diagnóstico de DSPN con un consenso que establece la sintomatología mínima para diagnosticar DSPN y su gravedad (6), así mismo, se han desarrollado dispositivos electrónicos que permiten evaluar la conducción nerviosa, pues su anormalidad es síntoma de DSPN, entre estos dispositivos están aquellos que evalúan la innervación colinérgica a través de la sudoración del pie (7) y la disfunción de las fibras nerviosas evaluando la sensación de frío (8), sin embargo estos instrumentos son costosos e imprácticos por la cantidad de tiempo que toman (9).

La prueba de Semmes-Weinstein se usa para determinar si un paciente ha perdido sensibilidad en los pies, consiste en ejercer presión a través de un monofilamento sobre determinadas zonas del pie (10). La practicidad y el bajo costo además de la buena asociación que ha tenido su uso con el elevado riesgo de úlceras en el pie y

amputaciones la ha hecho una prueba ampliamente usada. Lamentablemente, su reproducibilidad ha sido cuestionada pues no existe un protocolo estándar y ciertos detalles de su aplicación varían considerablemente entre el personal clínico (11).

De tal manera que la identificación temprana de un paciente con mayor riesgo de sufrir ulceraciones continúa siendo un desafío. En este contexto, técnicas de aprendizaje automático (*Machine Learning*) permiten elaborar modelos predictivos que ayuden a determinar la probabilidad de desarrollarse un evento en base a diferentes parámetros vienen siendo muy usadas en el campo de la medicina, especialmente en sistemas de apoyo al diagnóstico.

En este proyecto se desarrolló y evaluó un sistema de apoyo al diagnóstico en base a sensores en plantillas que permitan medir la presión plantar, es decir, la carga que en una persona debe soportar en la planta de los pies mientras camina. . Esta información se envía de manera local a una *laptop* donde se determinan las zonas con la presión plantar más elevada. Se ha determinado que las zonas donde se presentan ulceraciones son aquellas con las presiones plantares más elevadas y por este motivo se ha usado su medición como una herramienta para evaluar el riesgo de ulceraciones en pacientes diabéticos (12).

La información recolectada usando estos sensores en plantillas servirá para construir modelos predictivos en base a *Machine Learning* para poder identificar tempranamente aquellos pacientes que tienen mayor riesgo de padecer neuropatías.

## **2. Marco teórico**

### **2.1. La pandemia de la diabetes mellitus**

La diabetes mellitus (DM) es una desregulación del metabolismo de carbohidratos, lípidos y proteínas caracterizada por hiperglicemia y producida por una inadecuada liberación de insulina, resistencia a la insulina o una combinación de ambos (13,14).

El número de personas que padece diabetes se ha duplicado en los últimos 20 años (15). En la actualidad unas 400 millones de personas padecen la enfermedad a nivel global y se estima que estos números llegaran a los 600 millones en los próximos 15 a 20 años (14), con un preocupante aumento en la prevalencia en países de bajo y mediano ingreso (16).

La DM es un desorden incapacitante dado que la hiperglicemia crónica está asociada a un daño a largo plazo de múltiples órganos. Es la causa principal de amputación en los miembros, ceguera, insuficiencia renal y enfermedad cardiovascular (17). Así mismo, el padecer diabetes, aumenta el riesgo de morbilidad y mortalidad por enfermedades infecciosas como tuberculosis (16) y otras enfermedades respiratorias agudas (18).

Ampliamente, se puede clasificar a la diabetes en 2 tipos, la diabetes mellitus tipo 2 (DM2) es la más prevalente pues explica un 80 a 90% de los casos de diabetes. Se caracteriza por insulino-resistencia y una poca secreción de insulina. Se diagnostica con niveles de plasmáticos de glucosa mayores a 200 mg/dL, mayores a 126 mg/ dL tras 8 horas de ayuno y hemoglobina glicosilada (HbA1c) mayor a 6.5% (14).

Por otro lado, la diabetes mellitus tipo I (DMI) es una enfermedad autoinmune caracterizada por la deficiencia de insulina causada por la pérdida de células  $\beta$



pancreáticas (células especializadas en producir insulina) (19). Es la forma de diabetes más prevalente en niños.

La diabetes, en cualquiera de sus dos tipos, tiene complicaciones que incrementan la discapacidad y mortalidad constituyendo una amenaza para la economía de los países. Se estima que el 50% de los pacientes diabéticos sufrirá de neuropatía diabética a lo largo de su vida (20).

## **2.2. La neuropatía diabética**

La neuropatía es el daño o disfunción de las fibras nerviosas, es la complicación más frecuente en pacientes que sufren de diabetes (21) y se presenta en diferentes formas, de las cuales la más común es la polineuropatía sensitivomotora (DSPN) cuyo factor de riesgo más importante es la exposición total a la hiperglicemia. Se ha demostrado que puede ser estabilizada con un control estricto de la glicemia (22).

Lamentablemente, las neuropatías son altamente heterogéneas y complejas en cuanto a la sintomatología, alteración patológica y los mecanismos detrás de ellas (6) por lo que su clasificación es complicada. Se puede clasificarlas según su posición anatómica (proximal o distal; focal o multifocal); según su curso clínico (crónica o aguda) y según ocurrencia (típica o atípica) (23).

Esta última clasificación es la más útil desde el punto de vista clínico y es que maneja el consenso de expertos de Toronto (6). La neuropatía diabética periférica típica es la más común y se desarrolla en personas con hiperglicemia de larga data asociada a desbalances metabólicos como acumulación de productos de la glicosilación, estrés oxidativo y alteraciones lipídicas (6). La neuropatía diabética típica es crónica, distal, simétrica, predominantemente sensitiva y generalmente

dolorosa (23). Cualquier variación en estas características constituye atipicidad y se denomina neuropatía diabética atípica (24).

### **2.2.1 Métodos de diagnóstico de neuropatía diabética**

En el 2010, un grupo de expertos llegó a establecer una serie de criterios mínimos de diagnóstico para la DSPN basándose en los siguientes síntomas: sensibilidad disminuida, síntomas neuropáticos sensoriales como dolor punzante y adormecimiento, generalmente en pies, dedos del pie y piernas; o signos de disminución de la sensación distal o disminución o pérdida de los reflejos de la rodilla (25).

Una DSPN posible se presenta con la presencia de los anteriores síntomas, mientras que una DSPN probable se da con por lo menos dos o más de los síntomas descritos; la DSPN confirmada necesita la presencia de conducción nerviosa anormal y uno o más de los anteriores síntomas. También se define la DSPN subclínica cuando el paciente no presenta ninguno de los signos o síntomas descritos pero tiene una conducción nerviosa anormal.

El *gold standard* para el diagnóstico de neuropatía diabética son los estudios de conducción nerviosa en los que se cuantifica la disfunción nerviosa periférica y el grado de déficit sensitivo motor (26). Entre estos estudios, se tiene a la electromiografía de aguja (EMG) que usa un electrodo en forma de aguja que se inserta en el músculo. La EMG muestra en tiempo real y almacena la forma, amplitud y periodo de la onda y las tasas de disparo (*firing rate*) de una unidad motora simple o múltiple son las que proporcionan la información diagnóstica (26).

La EMG es una prueba complementaria a la observación clínica del paciente con DM. A veces, en el ámbito clínico, la neuropatía se diagnostica tras establecer la sintomatología, especialmente en casos de neuropatía dolorosa (27).

Otro método usado para el diagnóstico de neuropatía diabética es la prueba con el monofilamento Semmes Weinstein (SWME). Este método fue desarrollado por Florence Semmes y Sidney Weinstein en 1960 quienes a través de un set de monofilamentos de nailon podían medir la pérdida sensitiva en los pacientes. Este test consiste en aplicar el monofilamento perpendicularmente hasta que se doblen por un Segundo. Se instruye a los pacientes a responder si sienten el monofilamento. Si el paciente no logra sentir el monofilamento ese punto es considerado insensible. Esta prueba goza de popularidad por ser no invasiva y fácil de aplicar (28).

El set de monofilamentos desarrollado por Semmes & Weinstein consiste en diversos filamentos con diferentes números, sin embargo la sensibilidad y especificidad de detección varía dependiendo de qué número de filamento se usaba para la prueba. Actualmente, el número de filamento recomendado es el 5.07. Este número representa el “logaritmo de la fuerza aplicada (en décimas de miligramo) en un mismo lugar de la superficie de la piel” (29). La pérdida de la sensación protectora se define como la incapacidad para sentir el filamento 5.07/10 g.

### **2.3 Presión plantar y su importancia clínica**

La presión plantar del pie es “el campo de presión que actúa entre el pie y la superficie de soporte durante las actividades locomotoras diarias” (30). La importancia biomédica de estas mediciones reside en su aplicación al diseño de zapatos ortopédicos, prevención de heridas, diagnóstico de enfermedades de los miembros inferiores, etc.

En el caso de los pacientes diabéticos, las presiones plantares se usan para evaluar riesgo de ulceración (31). En personas sanas, el tejido blando de la planta del pie tiene una estructura especializada que consiste en cámaras llenas de grasa cuya función es disipar el estrés y absorber el impacto de una pisada (32). Sin embargo en personas diabéticas, se cree que la hiperglicemia que resulta en la glicosilación<sup>1</sup> de proteínas, especialmente colágeno, es responsable de cambios en los tejidos que comprometen su funcionalidad e integridad mecánica (32,33). Se han observado fibras de colágeno distorsionadas y rotas en los talones de pacientes diabéticos (34). A consecuencia de esto, hay un aumento en las presiones plantares de una manera poco uniforme. Este aumento impide el suministro de oxígeno y aumenta el riesgo de ulceración (32). Así mismo, se ha demostrado que el tejido blando plantar debajo de las cabezas de los metatarsos de pacientes diabéticos debe soportar mayor energía, lo que conduciría a ruptura tisular y ulceración (35).

### **2.3.1 Medición de la presión plantar**

Existe una gran variedad de sensores que pueden usarse para medir la presión dentro de una plantilla, entre ellos destacan los *Force Sensitive Resistor* (FSR) que son dispositivos a base de polímeros que tienen una variación en la resistencia interna cuando se ejerce diferentes magnitudes de fuerza en su superficie. Estos sensores poseen partes flexibles con semiconductores impresos en conjunto con electrodos interdigitales que consisten en dos tiras de matriz de microelectrodos direccionables individualmente (Figura 1) en su composición por lo que suelen ser más fáciles de acoplar a diferentes superficies.

---

<sup>1</sup> La glicación o glicosilación no enzimática es la reacción química espontánea de reducción de azúcares en la que una molécula de azúcar (glucosa o fructuosa) se une mediante un enlace covalente a una proteína o lípido (3).

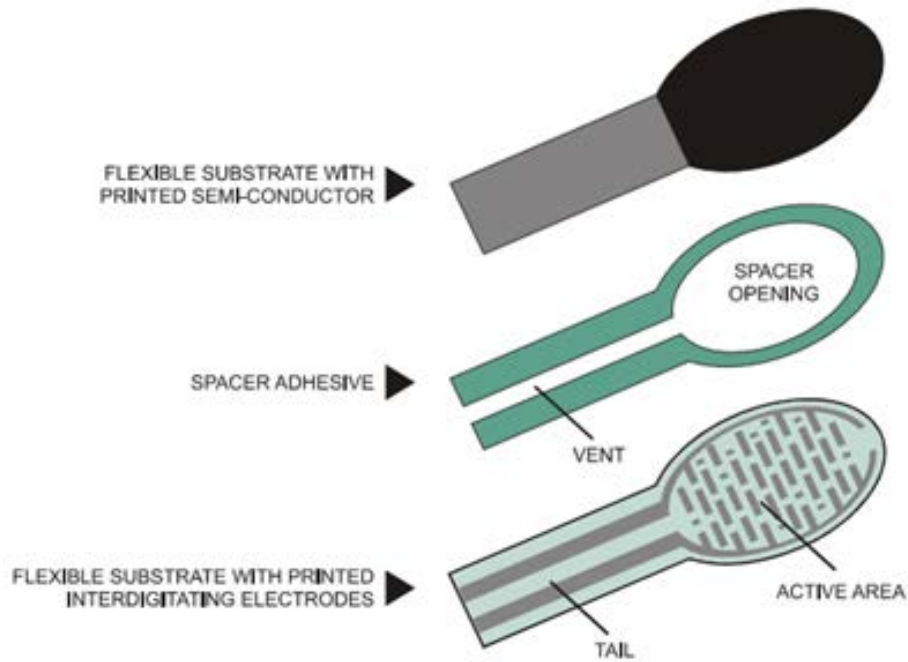


Figura 1. Partes de un sensor *Force sensitive resistive* (FSR). Se observa el sustrato flexible en el semiconductor impreso junto a electrodos interdigitales. Al aplicar fuerza de diferente magnitud, hay una variación en la resistencia interna de los FSRs.  
Adaptado de (36)

Otra opción son los sensores piezoeléctricos que funcionan en base al efecto del mismo nombre, al tener una variación en alguna de sus dimensiones se producen pequeñas corrientes eléctricas las cuales pueden transformarse en picos de voltaje a través de una resistencia o al inducirle potencial eléctrico puede hacer que se alteren sus dimensiones (Figura 2).

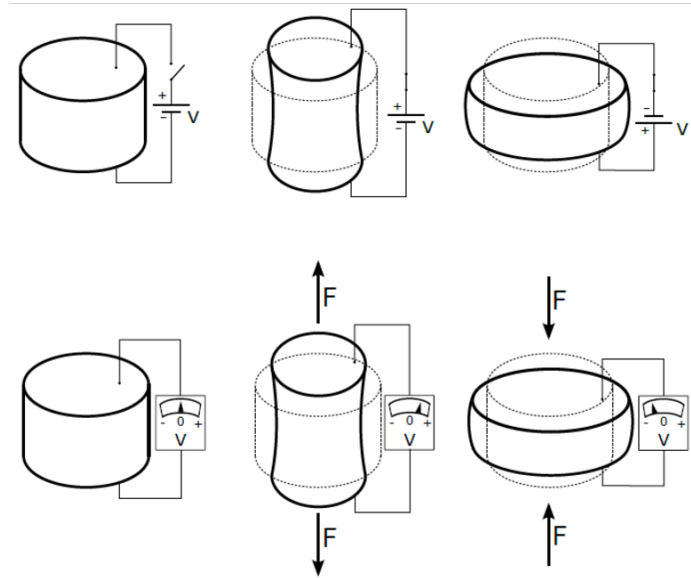
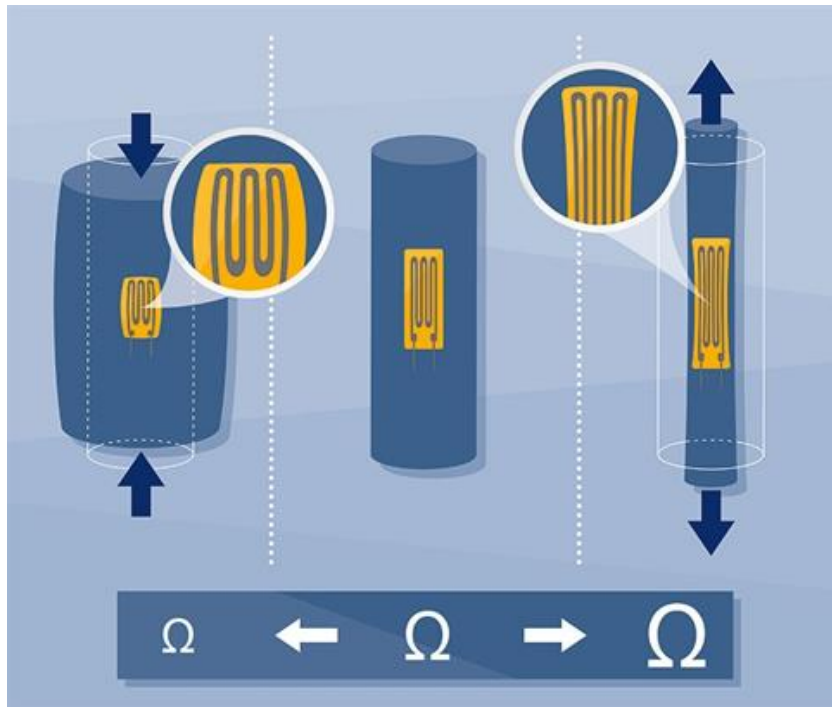


Figura 2. Representación del efecto piezoeléctrico. En sensores con este efecto se producen un pico de voltaje al ser variada su dimensión.

F: Fuerza, V: Voltaje, Adaptado de (37)

Como última alternativa también existen las galgas extensiométricas también llamadas extensómetros los cuales pueden usarse no solo para medir presión sino también deformación y carga estos funcionan en base al efecto piezorresistivo la cual es la propiedad de los materiales de variar su resistencia interna a causa de deformaciones y esfuerzos, esta resistencia variable puede ser medida a través de distintos circuitos de acondicionamiento.



*Figura 3. Galgas Extensiométricas usadas en el proyecto. Sirven para medir presión, deformación y carga.*  
Adaptado de (38)

## **2.4 La cuarta Revolución Industrial y la inteligencia artificial**

A pesar que la inteligencia artificial (IA) tuvo sus inicios en la década de los 50's, es recién en los últimos años que se ha dejado en manifiesto la real dimensión de la importancia de la aplicación de la IA. Se ha llegado a decir que la IA es la “nueva electricidad” por el potencial impacto revolucionario en diferentes aspectos en la sociedad (39).

Históricamente, estamos viviendo la era de la 4ta revolución industrial en la que la IA juega un rol preponderante (40). La primera estuvo caracterizada por las máquinas de vapor que incrementaron exponencialmente la productividad de los seres humanos. La segunda se caracterizó por la electricidad y la producción en

masa; mientras que la tercera tuvo como motor a la industria de la computación y el uso de computadores personales. En esta nueva revolución que vivimos conocida como “revolución digital” combina la capacidad tecnológica y humana a través de algoritmos capaces de aprender de sí mismos.

En líneas generales, la inteligencia artificial (IA) es la inteligencia que poseen las máquinas; sin embargo, desde que fuera por primera vez acuñado el término, ha habido muchos intentos de definir IA. Estas definiciones se presentan en la Figura 4. Se puede apreciar que las definiciones caen en dos dimensiones, una enfocada en los procesos de pensamiento y razonamiento y otras que se enfocan en cómo actúan estos sistemas, es decir, en el comportamiento (41).

Estas diferencias son importantes pues el éxito de las definiciones de la izquierda es medido en comparación con el performance de humanos, mientras que las de la derecha definen éxito de una manera más racional, pues la comparan con un performance ideal (41). Es así que tenemos cuatro categorías de inteligencia artificial: Sistemas que piensan como humanos, sistemas que piensan racionalmente, sistemas que actúan como humanos y sistemas que actúan racionalmente.



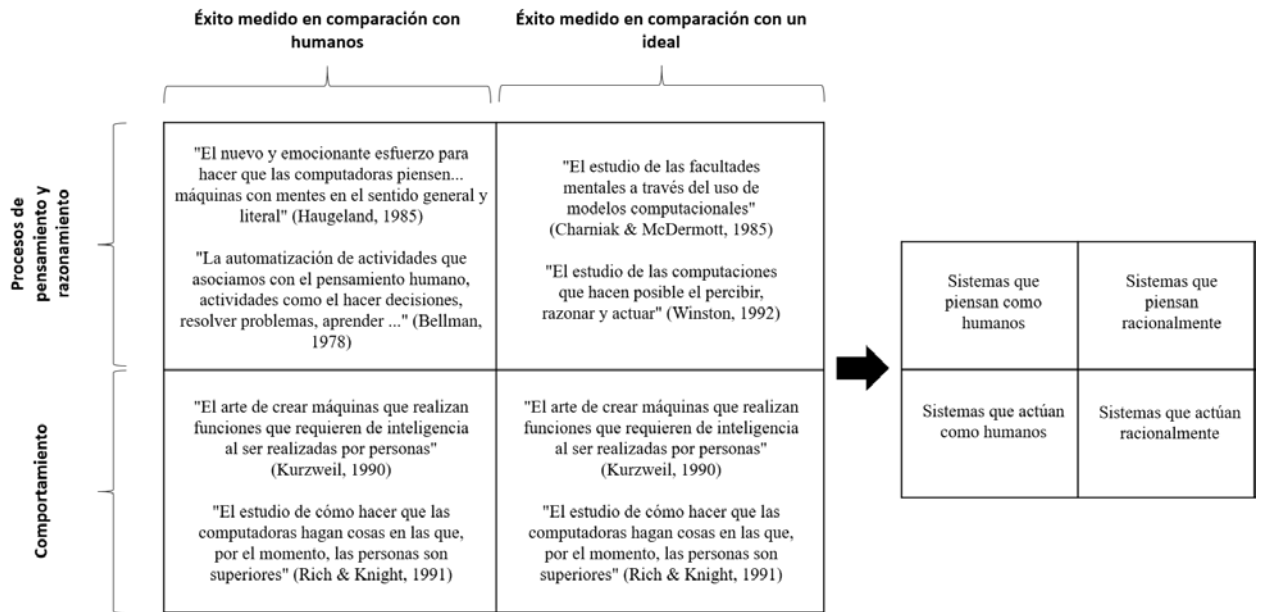


Figura 4. Enfoques de la inteligencia artificial. Se basan en las distintas definiciones de IA aceptadas que dan lugar a los cuatro enfoques del cuadro de la izquierda. Adaptado de Russell & Norvig (41)

### 2.4.1 El Aprendizaje automático o *Machine Learning*

*Machine Learning* o Aprendizaje Automático es una de las ramas de la Inteligencia Artificial (IA) más usadas en la actualidad y su objetivo es lograr que los sistemas automáticos aprendan automáticamente a través de diferentes técnicas (42). “Aprender” en este contexto se refiere a identificar patrones o parámetros y predecir eventos o comportamientos.

*Machine Learning* tiene un amplio espectro de usos entre las que destacan los diagnósticos médicos, motores de búsqueda, detección de fraudes y riesgos financieros.

Entre los modelos más utilizados de *Machine Learning* se encuentran los “árboles de decisiones (*Decision Tree*), bosques aleatorios (*Random Forest*), vecinos más cercanos (*k-Nearest Neighbors*), máquinas de soporte vectorial (*Support Vector Machine*) y las redes neuronales artificiales (*Artificial Neural Networks*)”.

Para poder evaluar un sistema de aprendizaje automático se utilizan diferentes parámetros calculados en base a su matriz de confusión, la cual es una matriz que representa los números predichos y reales entre positivos y negativos como se muestra en la Figura 5.

		<b>Predicción</b>	
		<b>Positivos</b>	<b>Negativos</b>
<b>Observación</b>	<b>Positivos</b>	Verdaderos Positivos (VP)	Falsos Negativos (FN)
	<b>Negativos</b>	Falsos Positivos (FP)	Verdaderos Negativos (VN)

Figura 5. Matriz de confusión para la evaluación de un sistema de aprendizaje automático. Adaptado de (19)

1) Precisión:

$$\frac{\text{Verdaderos Positivos (VP)}}{\text{Total de Positivos}}$$

2) Recall:

$$\frac{\text{Verdaderos Positivos (VP)} + \text{Verdaderos Negativos (VN)}}{\text{Total}}$$

Se puede dividir *machine learning* en dos, según el tipo de aprendizaje: supervisado, donde los algoritmos usan bases de datos en la cual se conoce la variable de salida, para encontrar una función que, dadas las variables de entrada, les asigne la etiqueta de salida adecuada. El algoritmo se entrena con un conjunto

de datos históricos y asigna la etiqueta de salida adecuada a una nueva entrada. Se usa el aprendizaje supervisado cuando se busca predecir clases es decir problemas de clasificación o también de regresión, algunos ejemplos de esto son el diagnóstico de enfermedades, predicción de quiebra en empresas, reconocimiento de caracteres en imágenes, etc.

En el aprendizaje no supervisado se trabaja de forma parecida a los supervisados, con la diferencia de que aquí no se tiene una variable de salida establecida, los datos ajustan su modelo predictivo tomando en cuenta solo los datos de entrada los cuales no están clasificados ni etiquetados para entrenarse, entro de este tipo de algoritmos comúnmente se usa el agrupamiento ya que divide los datos de entrada en grupos que tengan similitudes entre sí.

#### **2.4.2. Lenguajes de entorno y algoritmos de búsqueda**

El entorno de programación para desarrollar IA está muy asociado a los lenguajes que usa. ¿Cuál es el lenguaje ideal para desarrollar programas en IA? Pues el programa ideal sería aquel que fuera capaz de resolver cualquier tarea dentro de ese lenguaje. Y como lo menciona Corlett, un programa así no existe pues diferentes lenguajes son más apropiados que otros en ciertas tareas por lo que la elección de que lenguaje usar depende del tipo de problema que se quiere abordar (43).

Sin embargo, durante mucho tiempo el programa más usado para desarrollar IA fue LISP. Este programa creado a fines de los años 50's para el procesamiento de listas le daba al desarrollador mucha flexibilidad. Las estructuras de los datos se crean dinámicamente sin necesidad de alocar memoria de manera explícita; las declaraciones de datos no eran necesarias y dado que el programa estaba generalmente acompañado de un programa intérprete, la compilación no solía ser necesaria. (44). Otros lenguajes

también usados para IA son Prolog y C (43). Aunque en años recientes se ha extendido el uso de Python por su eficiencia (45).

Los algoritmos de búsqueda son muy importantes dentro del desarrollo de la IA pues el uso de un correcto algoritmo de búsqueda es el factor crítico en la performance de un sistema inteligente (46) . Se pueden clasificar en aquellos que ejecutan una búsqueda informada y aquellos que ejecutan una búsqueda no informada o ciega. Dentro de los algoritmos que realizan una búsqueda informada es importante el término heurística.

La heurística es una función que estima cuán cerca está un estado del estado target o estado objetivo. Los métodos heurísticos son preferidos en problemas complejos pues de hacerse de un modo no informado, el tiempo que tomaría la búsqueda sería muy grande (47). Entre estos algoritmos tenemos al algoritmo *greedy* o algoritmo voraz donde la heurística es hacer que la opción local sea la óptima esperando encontrar un óptimo global y a A\* donde, a partir de un nodo de partida, se pretende encontrar un camino que tenga la menor distancia de recorrido o el menor costo, El algoritmo elabora un árbol de caminos que se originan en el nodo de partida y se extienden hasta que el criterio de terminación haya sido satisfecho.

### **2.4.3. Aplicaciones de la inteligencia artificial**

Los algoritmos capaces de aprender de sí mismos reseñados anteriormente tienen una gran cantidad de aplicaciones. Por ejemplo, la minería de datos que permite “examinar, visualizar y descubrir patrones en datos que facilitan el proceso de toma de decisiones” (48). Estas decisiones generalmente están relacionadas al campo

financiero, donde la gran cantidad de datos crudos tiene un gran potencial comercializable, sin embargo también se ha aplicado en la salud (49).

Otra aplicación de la inteligencia artificial (IA) es la “*Big data*”, que son bases de datos tan largas o complejas que los métodos de procesamiento tradicionales son inadecuados. Esta clase de base de datos es encontrada con mucha facilidad dentro del campo de la salud. Las nuevas tecnologías como la genómica, proteómica y metabolómica pueden producir inmensas cantidades de datos de un mismo individuo. *Big data* en salud se viene usando en investigación. Las grandes cantidades de datos de los que se habló son estudiados para saber más sobre una condición o enfermedad (50). Los motores de búsqueda de gigantes tecnológicos como Google o Amazon también usan inteligencia artificial. Permiten predecir qué es lo que buscará el usuario o qué es lo que tiene más probabilidades de comprar de acuerdo a su historial de compras (51,52). La IA también forma parte del mundo de la robótica. La co-creación robot-IA predomina en la industria de servicios donde se diseñan robots que puedan relacionarse con el público. Una revisión sistemática reciente concluyó que la mayoría de robots con IA se usan en servicio al cliente, seguido de robots que, sin reemplazar al individuo que proporciona el servicio al cliente, lo apoya en su labor. También hay robots “inteligentes” capaces de promover la interacción entre trabajadores como una forma de mejorar el ambiente laboral. Finalmente, se han descrito robots con IA capaces de interactuar de manera empática con las personas para mejorar su estado de ánimo (53).

#### **2.4.4. La inteligencia artificial en salud**

La inteligencia artificial es una rama de la ciencia de computación que pretende entender y construir entidades inteligentes, generalmente en la forma de *software* (54). El aprendizaje automatizado o *Machine learning* (ML) es una aplicación de la inteligencia artificial en la que a través de algoritmos se logra identificar patrones en un grupo de datos. Las variables de entrada de un algoritmo de ML usado en salud vienen a ser características propias del paciente como edad, sexo e historia clínica y datos específicos de la enfermedad como una imagen diagnóstica resultados de exámenes de laboratorio o exámenes físicos etc. A esto se suman resultados clínicos de interés (55).

Los algoritmos de ML no supervisados usados en salud usan principalmente características similares de los pacientes para agruparlos por clúster. El *clustering* es el proceso por el cual se organizan los datos de alguna manera (56). Hay técnicas clásicas de *clustering* clásicas como los particionales (k-means) que descomponen la base de datos en una partición plana que contiene k clústeres desunidos. Un algoritmo particional de este tipo “emplea un esquema de relocalización iterativa para producir clústeres donde se minimice la diferencia dentro los objetos de un mismo clúster” (56). Recientemente han surgido nuevas técnicas avanzadas de *clustering* como el *clustering* espectral. Estos algoritmos maximizan la similaridad de las características de pacientes dentro de un grupo y la minimizan entre clústeres. Sin embargo, como no se usa el *outcome* clínico, por lo general no se usa este tipo de aprendizaje en las aplicaciones de IA en salud, aunque por lo general suele ser el primer paso o el pre-procesamiento del uso de un algoritmo supervisado (55,57).

Entre las técnicas más usadas de ML supervisado en salud están *support vector machines* y las redes neuronales.

#### **2.4.5. Sistemas de apoyo al diagnóstico**

Los sistemas de apoyo al diagnóstico (MDDS) se vienen usando con mayor frecuencia en años recientes. En el Reino Unido, por ejemplo, se ha iniciado a adoptarlos en el primer nivel de atención (58) y existe evidencia de una mejora en la performance del médico tratante cuando usa estos sistemas (59).

La historia de los MDDS es larga pues ya en el año 1959 se empieza a hablar de la posibilidad de apoyar el diagnóstico médico con sistemas inteligentes, esto era ya visto como factible pues la probabilística y el razonamiento lógico eran la base de los diagnósticos clínicos. Se empezó a sugerir las probabilidades de Bayes para facilitar el diagnóstico y los fundamentos para los MDDS fueron establecidos (60). Actualmente, el campo donde la AI ha logrado calar más es el diagnóstico automático por imágenes. Las imágenes son esenciales para el diagnóstico de muchas patologías en diversas especialidades médicas como por ejemplo dermatología, donde hay sistemas de apoyo basado en imágenes para la temprana identificación de melanoma (61) o para diferenciar lesiones benignas de carcinomas de piel (62).

Estos sistemas se basan en el reconocimiento de patrones, “una aplicación de *machine learning* que tiene como fundamento el entrenar con nuevos datos usando un enfoque adaptativo como una red neuronal pero sin la necesidad de programación explícita previa” (63). Por ejemplo, para el entrenamiento se usa una gran cantidad imágenes ya identificadas y la discriminación se basa en usar la

imagen entera como patrón seguida de la validación de imágenes prueba de una base de datos. Como las bases de datos de entrenamiento son muy grandes y contienen muchas imágenes, se compensa la variabilidad entre imágenes sin la necesidad de normalizaciones de alguna aberración geométrica que se pudiera presentar (63).

En años recientes, la aplicación de la inteligencia artificial en el diagnóstico no se ha circunscrito únicamente a identificar patrones pues gracias a avances en *Deep Learning* ahora se usan algoritmos que también identifican, clasifican, y hasta cuantifican patrones en imágenes médicas (64).

Las aplicaciones de ML y AI en medicina ya son una realidad, pero hay problemas en su implementación que nacen por la falta de estudios de aceptabilidad y usabilidad que consideren el punto de vista del médico.

#### **2.4.6. Tecnologías de comunicación inalámbrica**

Existen en el mercado diversos dispositivos que transmiten datos de forma inalámbrica, las principales diferencias que tienen son la máxima distancia de separación entre dispositivos, la frecuencia de transmisión y el protocolo de comunicación que usan. Uno de los más usados en dispositivos móviles son los *bluetooth* los cuales son de corto alcance (Máximo 10 metros de separación entre dispositivos), en la versión 2.0 tenemos los módulos HC que se comunican entre sí configurados como maestro y esclavo, en la versión 4.0 tenemos los BLE (*Bluetooth Low Energy*) de bajo consumo que tienen una mayor distancia de comunicación, estos BLE en conjunto con sensores forman los llamados *Beacons*



usados para distintas aplicaciones como control de acceso, marketing, geolocalización en interiores e *IoT(Internet of things)*. Otra conocida gama de módulos de comunicación inalámbrica son los *Zigbee* un estándar de comunicación diseñado para largas distancias y baja tasa de datos, otra de sus cualidades es poder usar diferentes topologías de red como estrella, malla y árbol, entre sus módulos más usados están los *Xbee* y *Xbee Pro*. En los últimos años surgieron los dispositivos *Narrowband* (Ancho de banda estrecho), la peculiaridad de estos es que usa un canal pequeño tiene mayor sensibilidad, menos ruido y un bajo consumo de energía, características que lo hacen una gran opción para aplicaciones de IoT.

### **3. Justificación**

El número de personas con diabetes a nivel mundial en 2014 fue de 108 millones, en 30 años la prevalencia se duplicó pasando de 4.7% en 1980 a 8.5% en 2014. En países de ingresos medianos y bajos el aumento en la prevalencia fue mayor que en países desarrollados (65) y en el caso particular de las Américas, se estima que 30 a 40% de personas que sufren de diabetes no han sido diagnosticadas por lo que no reciben tratamiento y/o control adecuado aumentando así el riesgo de neuropatía (66).

La DSPN es altamente incapacitante pues sus complicaciones incluyen heridas y úlceras en los pies que pueden culminar en la amputación del miembro y dada su naturaleza crónica está relacionada a depresión y ansiedad (67).

El diagnóstico de DSPN continúa siendo un desafío. En el 2010 se establecieron por consenso criterios mínimos de diagnóstico para DSPN probable, DSPN posible, DSPN subclínica y DSPN confirmada y también se han desarrollado herramientas tecnológicas de apoyo al diagnóstico. Sin embargo y tal como concluyen Papanas & Ziegler (68) hay mucho por hacer en el desarrollo de técnicas que “monitoreen la progresión y regresión del DSPN y sobre todo aquellas capaces de predecir complicaciones como úlceras y amputaciones” (27).

## **4. Objetivos**

### **4.1. Objetivo General**

- Diseñar y evaluar un sistema de ayuda al diagnóstico de neuropatía diabética

### **4.2. Objetivos Específicos**

- Diseñar un algoritmo de aprendizaje automático (*Machine Learning*) que permita detectar el porcentaje de posibilidad de padecer una neuropatía diabética.
- Implementar el algoritmo dentro de un *software* de ayuda al diagnóstico.
- Evaluar la usabilidad del *software* de ayuda al diagnóstico.

## **5. Materiales y métodos**

### **5.1. Diseño del Estudio**

Este proyecto contó con dos partes:

- Estudio observacional, donde se reclutó pacientes diabéticos con y sin diagnóstico de neuropatía. El diagnóstico de neuropatía lo realizó el médico clínico tratante a través de la electromiografía. Los pacientes usaron las plantillas con sensores de presión desarrolladas para almacenar los valores pico de la presión plantar mientras hacían una caminata simple.
- Estudio de usabilidad: Después de realizado el algoritmo en base a los datos obtenidos del estudio observacional se desarrolló un sistema de apoyo al diagnóstico y se evaluó su usabilidad por el personal clínico (Anexo 1).

### **5.2. Población y muestra**

Pacientes con diagnóstico de diabetes con y sin neuropatía diabética. El diagnóstico de neuropatía se hizo con el método de electromiografía.

### Criterios de Inclusión

- Hombres y mujeres con diagnóstico de diabetes Tipo II
- Mayores de 40 años
- Consentimiento informado firmado

### Criterios de Exclusión

- Pacientes que presenten úlceras en los pies al momento del reclutamiento
- Mujeres gestantes
- Otras enfermedades no asociadas a síndrome metabólico (hipertensión, obesidad e hipercolesterolemia)
- Discapacidad motora

## **5.3. Procedimientos y técnicas**

### **5.3.1. Diseño de sistema de adquisición de datos**

#### **Descripción General del Sistema**

Este sistema contiene 3 elementos principales:

- **Módulo de Adquisición Derecho**

Este elemento almacena los valores obtenidos por los sensores de presión convertidas previamente mediante conversores análogo digitales (ADC) y los envía de forma inalámbrica a través de un módulo *Bluetooth HC-05* al módulo de recepción con su respectivo HC-05, todo esto dentro de la plantilla que corresponde al pie derecho.

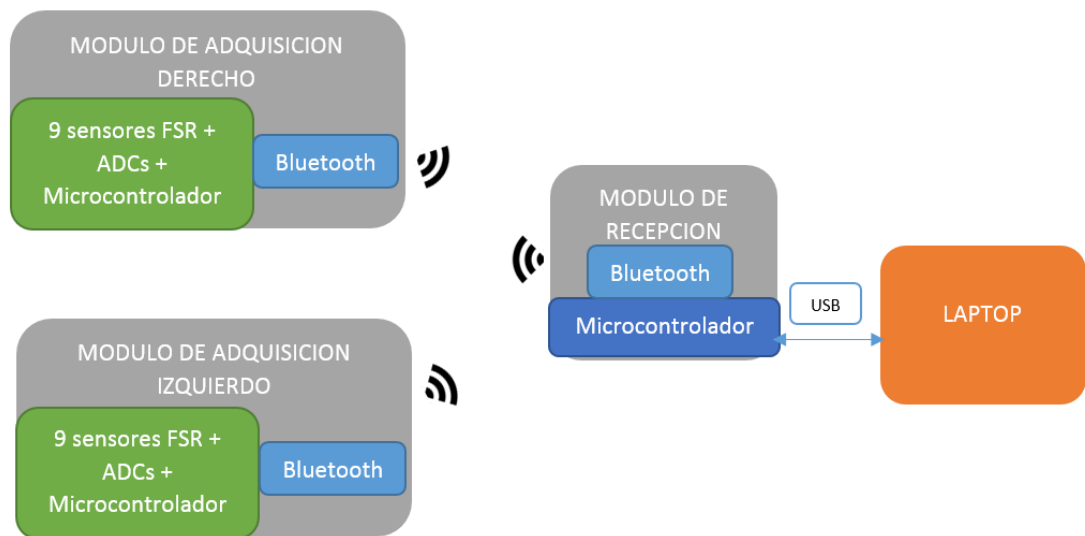
- **Módulo de Adquisición Izquierdo**

Este elemento almacena los valores obtenidos por los sensores de presión convertidas previamente mediante conversores análogo digitales (ADC) y los envía de forma inalámbrica a través de un módulo *Bluetooth HC-05* al módulo de

recepción con su respectivo HC-05, todo esto dentro de la plantilla que corresponde al pie izquierdo.

- **Módulo de Recepción**

Este elemento recibe de forma inalámbrica los datos almacenados por los módulos de adquisición y los une en una sola trama de datos para enviarlos vía USB hacia la *laptop* o PC.



*Figura 6. Componentes del sistema de adquisición*

## **Componentes electrónicos de los módulos**

### **Arduino Nano**

Para poder adquirir los datos de presión es necesario usar un microcontrolador, actualmente existen muchos módulos con microcontroladores integrados que facilitan este trabajo, la plataforma Arduino Nano es uno de los más versátiles por su tamaño y su conexión directa a USB para programar y adquirir datos. Sus especificaciones son las siguientes:

- Microcontrolador: ATmega328

- Pines Digitales (D0-D13): 14
- Pines Analógicos (A0-A7): 8
- Soporta comunicación USART, SPI y I2C.



*Figura 7. Arduino Nano*

### **Sensores de Presión**

Para la medición de presión plantar se usaron los sensores FSR-402, estos elementos pasivos presentan una disminución de su resistencia a medida que la fuerza se incrementa sobre la superficie del sensor, sus características son las siguientes:



*Figura 8. Sensor de presión FSR-402*

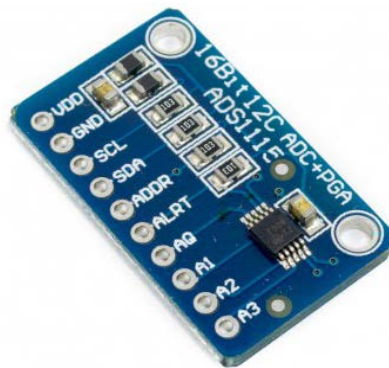
- Diámetro del área sensible: 10 mm
- Rango de trabajo: 0.1-100 N (0.01-10 kgf)

- Tipo de sensor: Resistivo (varía inversamente con la fuerza aplicada)
- Resistencia máxima: 100 KOhm (sin carga aplicada)
- Resistencia mínima: 200 Ohm (con una fuerza de 100 N aplicada)

Dado que necesitamos 9 sensores de presión para medir los puntos principales de la planta del pie y el microcontrolador solo tiene 8 canales, usaremos un módulo para aumentar la cantidad de canales ADC.

### **Módulo ADS1115**

Este módulo tiene 4 canales ADC de 16 bits de resolución, y también se pueden leer como 2 canales diferenciales, los datos digitales son leídos mediante bus I2C y puede alcanzar hasta 860 muestras por segundo, las especificaciones son las siguientes:



*Figura 9. Módulo ADS1115*

- Voltaje de Alimentación: 2.0V- 5.5V DC
- Interface con microcontrolador: I2C
- 4 canales en modo individual, 2 canales en modo comparador
- Circuito de acondicionamiento con divisor de voltaje y filtro RC
- Rango de ganancia programable: 1-128

## Módulos Bluetooth

Estos módulos usan un protocolo de comunicaciones diseñado especialmente para dispositivos de bajo consumo, son de corto alcance dado que tienen un máximo de distancia de 10 metros y están basados en transceptores de bajo costo.

Estos dispositivos se comunican entre sí automáticamente cuando son configurados como maestro y esclavo, son previamente emparejados y se encuentran dentro de su alcance. Este enlace se realiza vía radiofrecuencia por lo que no tienen que estar alineados. Para esta aplicación se hizo uso de los HC-05 los cuales tienen las siguientes especificaciones:

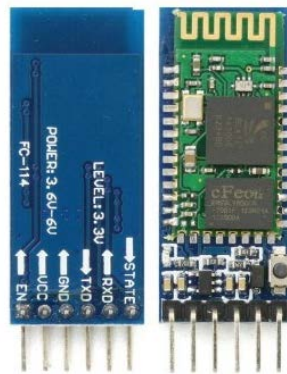


Figura 10. Módulos Bluetooth

- Protocolo: *Bluetooth* v 2.0
- Frecuencia: 2.4 GHz
- Potencia de transmisión: 4dBm, Clase 2
- Modos: puerto serie *Bluetooth* (maestro y esclavo)
- Alimentación: + 3.3VDC 50mA



### **Amplificador Operacional (Op-Amp)**

En este desarrollo necesitamos un circuito para acondicionar los sensores de presión ya que son resistencias variables necesitamos convertirlas a voltaje para ser leídos por los ADC, para ello se usaron los dispositivos LM358 que consisten de dos amplificadores operacionales independientes, de gran ganancia y diseñados para operar una sola alimentación, sus características las siguientes:



*Figura 11. Amplificador operacional LM358*

- Fuente única: 3 V a 32 V
- Drenaje de corriente de suministro baja independiente del voltaje de suministro:  
0.7 mA típica
- Ancho de banda de ganancia de unidad: 0.7 MHz
- Ganancia de voltaje diferencial de lazo abierto: 100 dB típica

### **Diseño de los módulos de adquisición**

Para una conversión simple de fuerza a voltaje, el FSR está conectado a una resistencia de medición en una configuración de divisor de voltaje (Figura 12), el voltaje de salida se describe mediante la ecuación:

$$V_{out} = \frac{R_M \cdot V}{R_M + R_{FSR}}$$

Donde:

$V_{out}$  = Voltaje de salida

$R_M$  = Resistencia de medicion

$R_{FSR}$  = Resistencia del sensor FSR

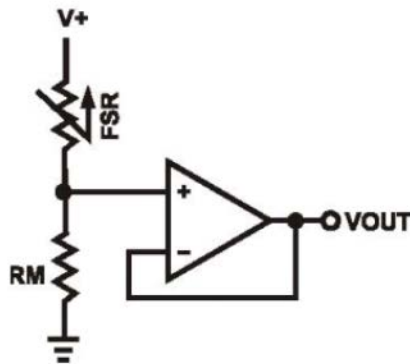


Figura 12. Diagrama de configuración para lectura de sensor

$R_M$  se elige para maximizar el rango de sensibilidad de fuerza deseado y para limitar la corriente, para esta aplicación se eligió 10Kohm. La corriente a través del FSR debe limitarse a menos de 1 mA / cm cuadrado de fuerza aplicada. Los amplificadores operacionales sugeridos en la documentación del sensor para estos diseños de son los *op-amp* LM358, por lo que el diseño de adquisición de 2 sensores por un *op-amp* sería como se muestra en la Figura 13.

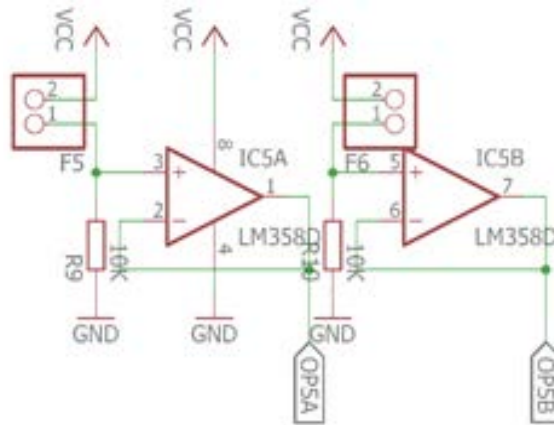


Figura 13. Diagrama de la adquisición de datos de dos sensores

Para el caso del ADS1115 ya que usa comunicación I2C se debe conectar en los pines A4 y A5 del Arduino Nano a través de esa comunicación se enviarán los valores leídos por 4 canales de este módulo, en el caso del Arduino se quedan los 6 pines (A1-A3 y A6-A7) libres para leer las salidas de los *op-amp* por lo que en total tendríamos 10 canales para leer valores de voltaje analógico (Figura 14).

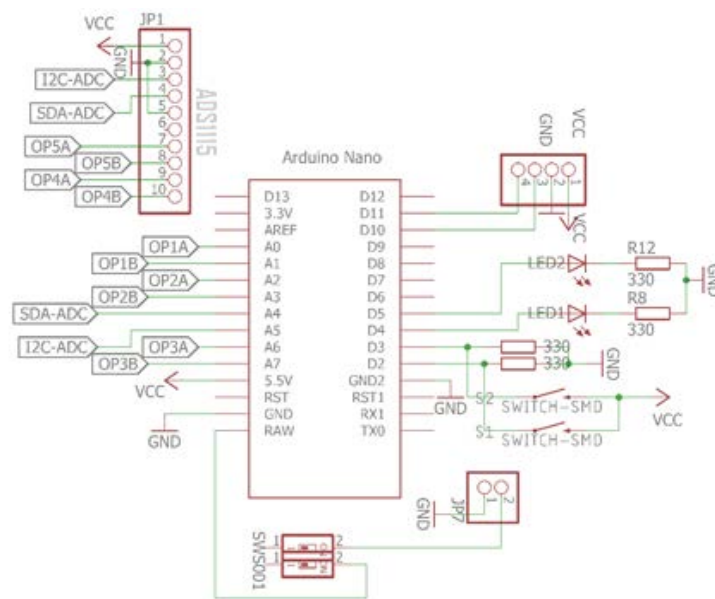


Figura 14. Conexión entre Arduino Nano y ADS1115

Para la comunicación de estos módulos de adquisición se configuraron los módulos *Bluetooth* en modo esclavo ambos emparejados con su par en el módulo de recepción.

El esquemático final del circuito se muestra en el Anexo 2.

### Diseño PCB

Para un mejor desempeño del diseño se diseñó una placa de circuito impreso (PCB). El diseño se hizo usando los *Op-Amp*, resistencias, *switch* y *leds* en empaquetadura superficial (SMD) para reducir espacio, los demás componentes usaron una empaquetadura *true hole*. El diseño de la placa fue de una sola capa (Figura 15).

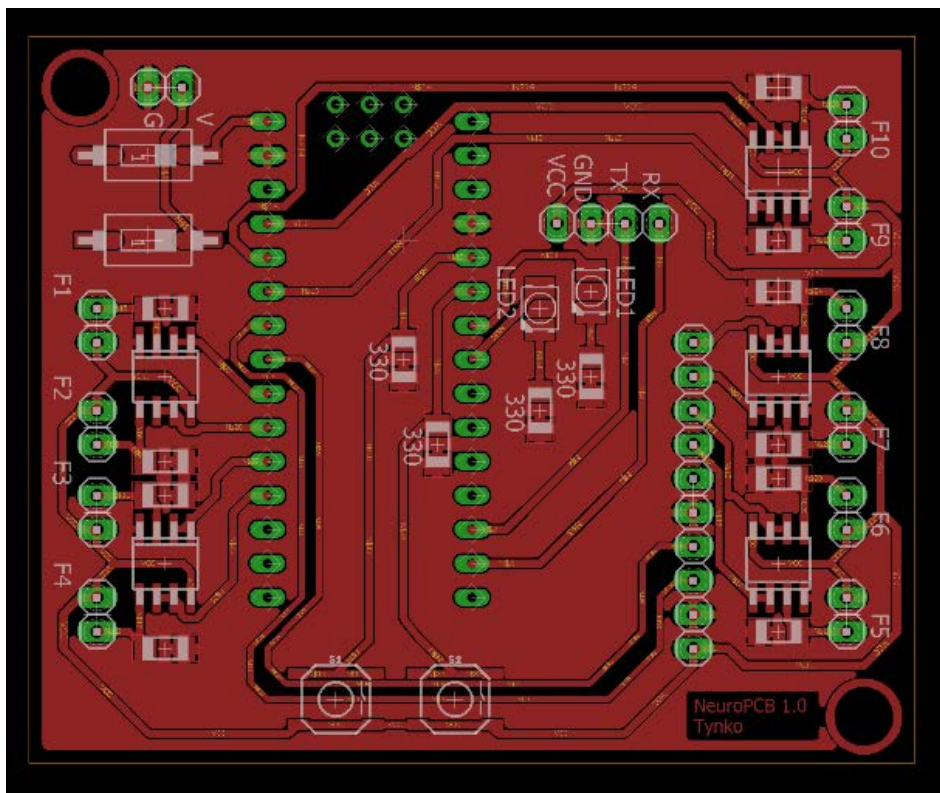


Figura 15. Diseño de la placa de circuito impreso

### Diseño de estructura de soporte

Dado que este circuito estará usándose en movimiento (paciente caminando), fue necesario diseñar e imprimir en 3D un *case* o estructura de soporte que pueda mantener en buenas condiciones el PCB e conjunto con una batería de alimentación

del tipo LiPo (Figura 16). Esta estructura fue impresa usando filamento del tipo ABS (*Acrylonitrile Butadiene Styrene*). Se puede encontrar este diseño en el siguiente enlace:

[https://github.com/jptincopa/Sistema\\_Apoyo\\_Diagnostico\\_Neuropatia\\_ML/blob/master/Hardware/3D\\_Files/case\\_FSR.stl](https://github.com/jptincopa/Sistema_Apoyo_Diagnostico_Neuropatia_ML/blob/master/Hardware/3D_Files/case_FSR.stl)

Para la alimentación del circuito se utilizaron dos baterías de LiPo en serie, en conjunto estas serían sus características



*Figura 16. Batería de polímero de litio*

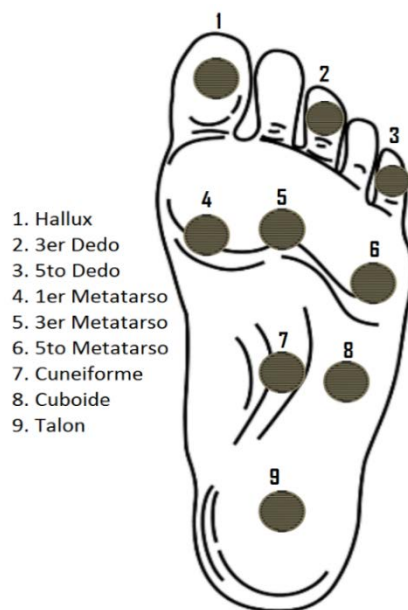
- Voltaje: 7.4v
- Capacidad: 1000 mAh

## Programación de los módulos de adquisición

La programación de estos módulos está hecha en lenguaje arduino, el *script* completo está en [https://github.com/jptincopa/Sistema\\_Apoyo\\_Diagnostico\\_Neuropatia\\_ML/tree/master/Software](https://github.com/jptincopa/Sistema_Apoyo_Diagnostico_Neuropatia_ML/tree/master/Software)

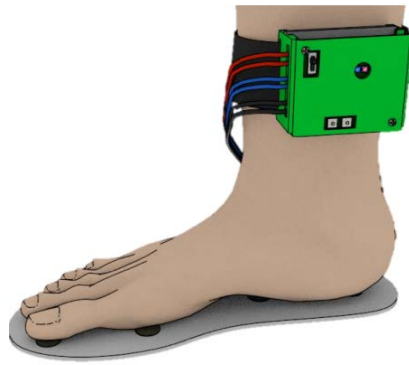
## Plantillas con sensores

Para la medición de presión plantar se hizo uso de plantillas de zapato según la talla del paciente con 9 sensores previamente detallados y distribuidos de como se muestra en la Figura 17.



*Figura 17. Distribución de los sensores en la planta del pie.*

En la Figura 18 podemos observar todos los componentes que forman parte del módulo de adquisición tanto derecho como izquierdo:



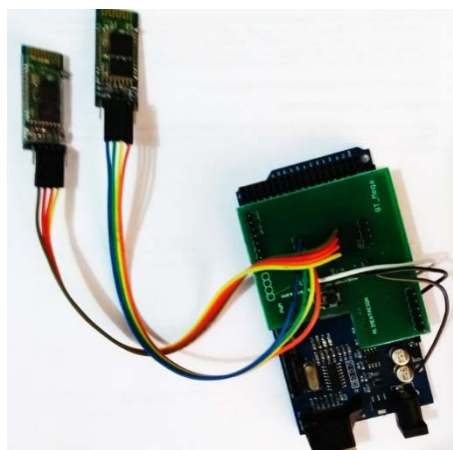
*Figura 18. Módulo de adquisición colocado en el pie derecho*

### **Diseño del módulo de recepción**

Para el diseño este módulo se hizo uso de un Arduino Mega por poseer más de un puerto serial de comunicación en los cuales se pueden colocar los módulos *Bluetooth* en modo maestro, uno para cada módulo de adquisición (Derecho e izquierdo).

Este Arduino (Figura 19) está conectado directamente a la PC o *laptop* el cual tiene un *script* hecho en lenguaje de programación Python, que se detalla más adelante.

Este módulo se encarga de juntar la trama de datos obtenidos por los módulos de adquisición y enviarlos hacia la PC mediante el puerto USB.



*Figura 19. Módulo de recepción de datos*

## Programación del módulo de recepción

La programación de este módulo está hecha en lenguaje arduino, el *script* completo está en

[https://github.com/jptincopa/Sistema\\_Apoyo\\_Diagnostico\\_Neuropatia\\_ML/tree/master/Software](https://github.com/jptincopa/Sistema_Apoyo_Diagnostico_Neuropatia_ML/tree/master/Software)

## Software para recolección de datos

Para recolectar y almacenar los datos obtenidos por los sensores se desarrolló un *script* en Python, descrito de la siguiente manera

- Importación de librerías y apertura del puerto serial donde está conectado el módulo de recepción, estableciendo la velocidad de transmisión en 9600baud

```
import serial
import csv

ser = serial.Serial("/dev/ttyUSB0", 9600, timeout=0.1)
ser.flushInput()
```

- Creación de un bucle donde se lee cada trama obtenida en el puerto, la cual se separa en comas y se guarda en un *array* de datos llamado “data”

```
while True:
    ser_bytes = ser.readline()
    data=(ser_bytes).split(",")
    if(len(data)>0):
        print(ser_bytes)
    print(a[1])
```

- Dentro del mismo bucle se crea un archivo de valores separados por coma (CSV) dentro del cual en cada fila se almacenan los valores obtenidos por el puerto separando cada celda con el delimitador “,”



```
with open("Adquisicion_P01.csv", "a") as f:
    writer = csv.writer(f, delimiter=",")
    writer.writerow([time.time(), ser_bytes])
```

En el momento que se ejecuta el *script* el bucle comienza a recolectar datos hasta que el usuario determine cuando se deba detener.

### 5.3.2. Recojo de datos sobre presión plantar

Para iniciar la recolección se ajustó el zapato que contiene la plantilla con *velcro* para que coincidan con la posición de los sensores con los 9 puntos a medir como se observa en la Figura 20.

- Una vez colocadas las plantillas en ambos pies se encienden los módulos de adquisición con el *switch* disponible cada uno de ellos. Se espera a que los módulos *Bluetooth* se emparejen, esto se confirma al momento que la rapidez de parpadeo disminuye.
- Una vez se confirma que todo está correctamente colocado y encendido se ejecuta el *software* para recolección de datos, los valores obtenidos aparecerán en la terminal del programa.
- Se le pide al paciente que camine por aproximadamente 1 minuto y luego se presiona la tecla *ENTER* para detener la recolección.



*Figura 20. Circuito de adquisición colocado en los pies del paciente*

### 5.3.3. Elaboración del algoritmo

#### Descripción de la base de datos

La base de datos consta de 39 entradas la cual contiene datos personales de los pacientes dentro de las siguientes variables:

*Tabla 1. Descripción de parámetros de la base de datos*

<b>Variable</b>	<b>Definición Operacional</b>	<b>Valores Posibles</b>	<b>Criterios de Medición</b>	<b>Tipo de Variable</b>
Edad	La edad reportada que figura en su documento de identidad	>40 años	Cuantitativa, Discreta Escala: Razón	<i>Int</i> (Entero)
Peso	Peso medido al momento del reclutamiento	40-150 Kg	Cuantitativa, Discreta Escala: Razón	<i>Float</i> (Valor con coma flotante)
Estatura	Estatura medida al momento del reclutamiento	120-180 cm.	Cuantitativa, Discreta Escala: Razón	<i>Float</i> (Valor con coma flotante)
Zapato	Talla del zapato usado por el paciente	35 - 44	Cuantitativa, Discreta Escala: Razón	<i>Int</i> (Entero)
Sexo	El sexo que figura en su documento de identidad	- Masculino (1) - Femenino (0)	Cualitativa, Dicotómica Escala: Nominal	<i>Int</i> (Entero)
Tiempo DM	Años transcurridos desde que fue diagnosticado con diabetes mellitus	1 – 30 años	Cuantitativa, Discreta Escala: Razón	<i>Int</i> (Entero)
Diagnóstico de Neuropatía	Si fue diagnosticado con neuropatía diabética	- Si (1) - No (0)	Cualitativa, Dicotómica Escala: Nominal	<i>Int</i> (Entero)
D1 - D9	Valores medidos por los sensores de presión en el pie derecho	0 – 5 Voltios	Cuantitativa, Discreta Escala: Razón	<i>Float</i> (Valor con coma flotante)
I1 - I9	Valores medidos por los sensores de presión en el pie izquierdo	0 – 5 Voltios	Cuantitativa, Discreta Escala: Razón	<i>Float</i> (Valor con coma flotante)

## **Herramientas utilizadas**

Para el desarrollo del modelo predictivo se hizo uso del lenguaje de programación Python versión 3.5.2 dentro de la interfaz web *Jupyter Notebook* con el siguiente conjunto de librerías:

- *Scikit-learn* versión 0.20.3
- *Pandas* versión 0.24.2
- *Numpy* versión 1.16.2
- *SciPy* versión 0.17.0
- *Matplotlib* versión 3.0.3
- *Seaborn* versión 0.9.0
- *Mglearn* versión 0.1.7

## **Algoritmos de *Machine Learning* Desarrollados**

Por las características de la base de datos se decidió usar los siguientes algoritmos ya que tienen un mejor desempeño con *datasets* con una cantidad limitada de entradas

### **1. Decision Tree**

*Decision Tree* o árboles de decisión en español es un algoritmo de clasificación y regresión para el modelado de predicción de atributos discretos y continuos, este algoritmo identifica las columnas de entrada que se correlacionan con la columna de predicción.

Los pasos para ejecutar este algoritmo son los siguientes:

Obtención de los datos:

- Importación de librerías

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.model_selection import train_test_split
```

- Importación de la base de datos

```
df = pd.read_csv("DB_Pmax.csv")
```

### División en Conjunto de entrenamiento y de prueba

- Se separa del mismo la variable que contiene el Diagnóstico de Neuropatía ('Dx') ya que es la variable de salida. Además se divide la base de datos en 2 partes, la de entrenamiento ( $X_{train}$  y  $y_{train}$ ) y la de evaluación ( $X_{test}$  y  $y_{test}$ )

```
X = df.drop('Dx', axis=1)
y = df['Dx']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.50, random_state=0)
```

- De la librería SciKit-Learn se obtiene el clasificador correspondiente al algoritmo *Decision Tree* llamado "*DecisionTreeClassifier*", se configura para que use el criterio de entropía y se establece una semilla (*random\_state*), este valor es esencial para poder repetir este proceso y obtener los mismos resultados.

Se ajusta el clasificador "*dtree*" con los datos establecidos para el entrenamiento del algoritmo ( $X_{train}$  y  $y_{train}$ )

```
from sklearn.tree import DecisionTreeClassifier
dtree = DecisionTreeClassifier(criterion='entropy', max_depth=9, random_state=0)
dtree.fit(X_train, y_train)
```

## Predicción y Evaluación

Se usa el clasificador para realizar las predicciones con la parte de la base de datos establecida para la evaluación ( $X_{test}$ )

```
predictions = dtree.predict(X_test)
from sklearn.metrics import classification_report, confusion_matrix
```

- Luego se consiguen las matrices de confusión con los valores obtenidos por las predicciones y los valores de  $Y_{test}$

```
print(confusion_matrix(y_test, predictions))
```

- Este proceso se realiza de forma iterativa cambiando los porcentajes de la base de datos dedicada al entrenamiento y evaluación del algoritmo, además de cambiando los valores de la semilla de 0 a 2.

- Los reportes de clasificación obtenidos se detallan en la siguiente tabla:

*Tabla 2. Tabla de evaluaciones del algoritmo "Decision Tree"*

<b>RS</b>	<b>Train Size (%)</b>	<b>Test Size (%)</b>	<b>Decision Tree (Average Precision/Recall)</b>
<b>RandomState= 0</b>	85	15	0.42 / 0.33
	80	20	0.62 / 0.50
	75	25	0.55 / 0.40
	70	30	0.62 / 0.50
	65	35	0.62 / 0.50
	60	40	0.67 / 0.56
	55	45	0.81 / 0.78
<b>RandomState= 1</b>	50	50	0.73 / 0.70
	85	15	0.00 / 0.00
	80	20	0.41 / 0.38
	75	25	0.43 / 0.40
	70	30	0.45 / 0.42
	65	35	0.50 / 0.50
60	40	0.57 / 0.56	

<i>RS</i>	<i>Train Size (%)</i>	<i>Test Size (%)</i>	<i>Decision Tree (Average Precision/Recall)</i>
	55	45	0.61 / 0.61
	50	50	0.64 / 0.65
<i>RandomState = 2</i>	85	15	0.62 / 0.50
	80	20	0.62 / 0.50
	75	25	0.48 / 0.50
	70	30	0.46 / 0.50
	65	35	0.53 / 0.57
	60	40	0.60 / 0.62
	55	45	0.50 / 0.56
	50	50	0.44 / 0.55

## 2. Random Forest

*Random Forest* o Árboles Aleatorios es un algoritmo predictivo que usa la técnica de *Bagging*, la cual consiste en crear diferentes modelos usando muestras aleatorias con reemplazo. Con esto se combinan diferentes árboles, donde cada árbol es construido con observaciones y variables aleatorias.

### Obtención de los datos

#### - Importación de librerías

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.model_selection import train_test_split
```

#### - Importación de la base de datos

```
df = pd.read_csv("DB_Pmax.csv")
```

### División en Conjunto de entrenamiento y de prueba

- Se separa del mismo la variable que contiene el Diagnóstico de Neuropatía ('Dx') ya que es la variable de salida. Además, se divide la base de datos en 2 partes, la de entrenamiento ( $X_{train}$  y  $y_{train}$ ) y la de evaluación ( $X_{test}$  y  $y_{test}$ )

```
X = df.drop('Dx', axis=1)
y = df['Dx']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.50, random_state=0)
```

- De la librería *SciKit-Learn* se obtiene el clasificador correspondiente al algoritmo de *Random Forest* llamado "*RandomForestClassifier*", se configura para que use el criterio de entropía y con un número de estimadores de 100, se establece una semilla (*random\_state*).

Se ajusta el clasificador "rfc" con los datos establecidos para el entrenamiento del algoritmo ( $X_{train}$  y  $y_{train}$ )

```
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier( criterion='entropy', n_estimators=100, random_state=0)
rfc.fit(X_train, y_train)
```

### Predicción y Evaluación

Se usa el clasificador para realizar las predicciones con la parte de la base de datos establecida para la evaluación ( $X_{test}$ )

```
rfc_pred = rfc.predict(X_test)
```

- Luego se consiguen las matrices de confusión con los valores obtenidos por las predicciones y los valores de  $Y_{test}$

```
print(confusion_matrix(y_test, predictions))
```

- Este proceso se realiza de forma iterativa cambiando los porcentajes de la base de datos dedicada al entrenamiento y evaluación del algoritmo, además de cambiar los valores de la semilla de 0 a 2.



Los reportes de clasificación obtenidos se detallan en la Tabla 3

Tabla 3. Tabla de evaluaciones del algoritmo Random Forest

<b>RS</b>	<b>Train Size (%)</b>	<b>Test Size (%)</b>	<b>Random Forest (Average Precision/Recall)</b>
<i>RandomState= 0</i>	85	15	0.44 / 0.67
	80	20	0.56 / 0.75
	75	25	<b>0.84 / 0.80</b>
	70	30	0.82 / 0.75
	65	35	0.63 / 0.64
	60	40	0.75 / 0.75
	55	45	0.77 / 0.78
	50	50	0.69/ 0.70
<i>RandomState= 1</i>	85	15	0.00 / 0.00
	80	20	0.14 / 0.38
	75	25	0.57 / 0.50
	70	30	0.77 / 0.50
	65	35	0.77 / 0.57
	60	40	0.77 / 0.58
	55	45	<b>0.79 / 0.67</b>
	50	50	0.78 / 0.65
<i>RandomState= 2</i>	85	15	0.88 / 0.50
	80	20	<b>0.85 / 0.62</b>
	75	25	0.48 / 0.50
	70	30	0.43 / 0.42
	65	35	0.47 / 0.50
	60	40	0.51 / 0.56
	55	45	0.50 / 0.60
	50	50	0.54 / 0.60

### 3. K-Nearest Neighbours

Vecinos más Cercanos o *K-Nearest Neighbours* es un algoritmo que clasifica cada dato nuevo en el grupo que corresponda, según tenga  $k$  vecinos más cerca de un grupo o de otro. Es decir, calcula la distancia del elemento nuevo a cada uno de los existentes, y ordena dichas distancias de menor a mayor para ir seleccionando el grupo al que pertenecen.

## Obtención de los datos

- Importación de librerías

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.model_selection import train_test_split
```

- Importación de la base de datos

```
df = pd.read_csv("DB_Pmax.csv")
```

## Escalado de valores

La escala de las variables es importante por lo que se importa *StandardScaler* para ajustar los parámetros a medida de la variable que queremos medir “Dx”.

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
```

Para ello se ajusta los valores y se almacenan en “scaled\_features” y se excluye la variable “Dx”

```
scaler.fit(df.drop('Dx',axis=1))
scaled_features = scaler.transform(df.drop('Dx',axis=1))
df_feat = pd.DataFrame(scaled_features,columns=df.columns[:-1])
```

## División en Conjunto de entrenamiento y de prueba

- Se divide la base de datos en 2 partes, la de entrenamiento ( $X_{train}$  y  $y_{train}$ ) y la de evaluación ( $X_{test}$  y  $y_{test}$ ) y se establece una semilla (*random\_state*).

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(scaled_features,df['Dx'],|
                                                    test_size=0.35,random_state=1)
```

- Se obtiene el clasificador correspondiente al algoritmo de KNN llamado “*KNeighborsClassifier*”, se configura para que haga las predicciones con solo **1** vecino.

Se ajusta el clasificador “knn” con los datos establecidos para el entrenamiento del algoritmo (*X\_train* y *y\_train*)

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=1)
knn.fit(X_train,y_train)
```

- Se usa el clasificador para realizar las predicciones con la parte de la base de datos establecida para la evaluación (*X\_test*)

```
pred = knn.predict(X_test)
```

### Optimización del valor de K

Usaremos un bucle para medir el error con diferentes valores de K (Numero de vecinos) para elegir el valor más óptimo de K:

```
tasa_error = []

# Tomará algún tiempo
for i in range(1,25):

    knn = KNeighborsClassifier(n_neighbors=i)
    knn.fit(X_train,y_train)
    pred_i = knn.predict(X_test)
    tasa_error.append(np.mean(pred_i != y_test))
```

Ahora visualizaremos gráficamente cómo se comportan los valores de K con la configuración inicial

```
plt.figure(figsize=(10,6))
plt.plot(range(1,25),tasa_error,color='blue', linestyle='dashed', marker='o',
         markerfacecolor='red', markersize=10)
plt.title('Tasa de Error vs. Valor de K')
plt.xlabel('K')
plt.ylabel('Tasa de Error')
```

Con ello obtenemos la siguiente gráfica:

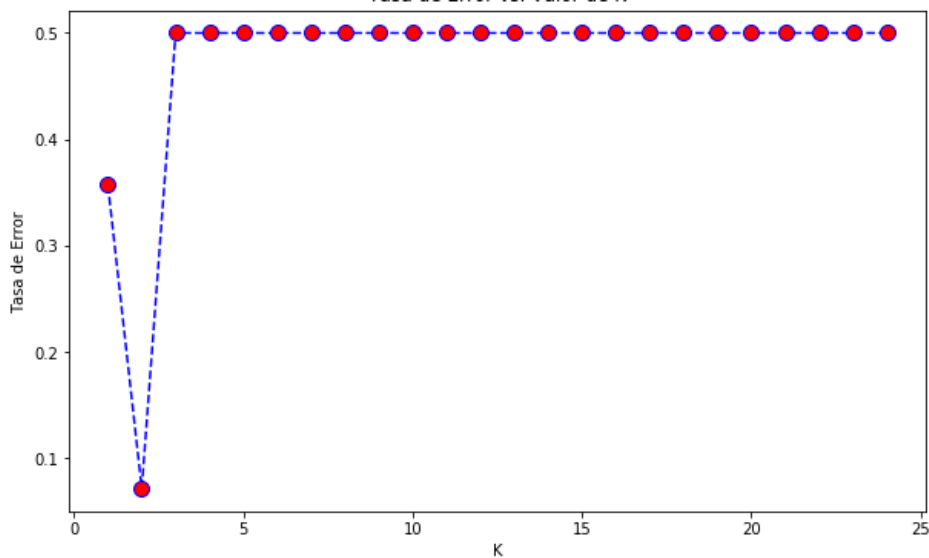


Figura 21. Tasa de error vs. Valor de K.

Como observamos en la Figura 21, el valor óptimo de k es igual 2, es decir, elegirá los 2 vecinos más cercanos para evaluar su clasificación por lo que corregimos el valor de K y reajustamos el entrenamiento del algoritmo y hacemos nuevamente las predicciones.

```
knn = KNeighborsClassifier(n_neighbors=2)
knn.fit(X_train,y_train)
pred = knn.predict(X_test)
```

- Luego se consiguen las matrices de confusión con los valores obtenidos por las predicciones y los valores de  $Y_{test}$

```
print(classification_report(y_test,pred))
```

- Este proceso se realiza de forma iterativa cambiando los porcentajes de la base de datos dedicada al entrenamiento y evaluación del algoritmo, además de cambiando los valores de la semilla de 0 a 2.

- Los reportes de clasificación obtenidos se detallan en la Tabla 4.

Tabla 4. Tabla de evaluación del algoritmo  $k$  – nearest neighbors

RS	Train Size (%)	Test Size (%)	Average Precision/Recall (Con $K=1$ )	Mejor $K$	Average Precision/Recall (Con el mejor $K$ )
RandomState= 0	85	15	0.67 / 0.67	2	<b>0.89 / 0.83</b>
	80	20	0.68 / 0.62	10	0.75 / 0.75
	75	25	0.85 / 0.70	7	0.84 / 0.80
	70	30	0.61 / 0.58	2	0.83 / 0.67
	65	35	0.56 / 0.50	9	0.80 / 0.71
	60	40	0.53 / 0.50	5	0.68 / 0.69
	55	45	0.52 / 0.44	5	0.74 / 0.72
	50	50	0.44 / 0.45	6	0.77 / 0.70
RandomState= 1	85	15	0.88 / 0.50	2	0.89 / 0.67
	80	20	0.81 / 0.50	2	0.85 / 0.75
	75	25	0.80 / 0.50	2	0.87 / 0.67
	70	30	0.79 / 0.58	2	0.84 / 0.75
	65	35	0.74 / 0.64	2	<b>0.94 / 0.93</b>
	60	40	0.79 / 0.62	2	0.83 / 0.75
	55	45	0.79 / 0.67	2	0.67 / 0.67
	50	50	0.65 / 0.65	2	0.64 / 0.65
RandomState= 2	85	15	0.89 / 0.67	5	<b>0.92 / 0.83</b>
	80	20	0.62 / 0.50	5	0.75 / 0.75
	75	25	0.48 / 0.50	4	0.65 / 0.60
	70	30	0.40 / 0.42	4	0.60 / 0.58
	65	35	0.47 / 0.50	4	0.66 / 0.64
	60	40	0.50 / 0.50	3	0.70 / 0.69
	55	45	0.56 / 0.56	3	0.78 / 0.78
	50	50	0.63 / 0.60	3	0.80 / 0.80

#### 4. Support Vector Machines

Una Máquina de Soporte Vectorial o *Support Vector Machine* (SVM) es un algoritmo en el cual los datos son mapeados mediante un *kernel* Gaussiano en base a un espacio de donde se busca la máxima separación entre clases.

Obtención de los datos

- Importación de librerías

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.model_selection import train_test_split
```

- Importación de la base de datos

```
df_feat = pd.read_csv("DB_Pmax.csv")
```

### División en Conjunto de entrenamiento y de prueba

Se divide la base de datos en 2 partes, la de entrenamiento ( $X_{train}$  y  $y_{train}$ ) y la de evaluación ( $X_{test}$  y  $y_{test}$ ) y se establece una semilla ( $random\_state$ ).

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(df_feat, df_feat['Dx'],
                                                  test_size=0.35, random_state=3)
```

- Se obtiene el clasificador correspondiente al algoritmo de KNN llamado “SVC”, se ajusta el clasificador “model” con los datos establecidos para el entrenamiento del algoritmo ( $X_{train}$  y  $y_{train}$ )

```
from sklearn.svm import SVC
model = SVC()
model.fit(X_train, y_train)
```

- Se usa el clasificador para realizar las predicciones con la parte de la base de datos establecida para la evaluación ( $X_{test}$ )

```
predictions = model.predict(X_test)
```

### Gridsearch

- Para saber que valores de C o gamma usar se crea una "cuadrícula" de parámetros y se prueba todas las combinaciones, a este procedimiento se le llama *Gridsearch* y cuando a este se le suma una validación cruzada, se llama a esto *GridSearchCV*.

- Se establecen los posibles valores para C y Gamma

```
param_grid = {'C': [0.0000001,0.000001,0.00001,0.0001,0.001,0.01,0.1,1, 10, 100, 1000,10000],
              'gamma': [1000000,100000,10000,1000,100,10,1,0.1,0.01,0.001,0.0001],
              'kernel': ['rbf']}
```

- Se importa *GridSearchCV* y se establecen los parámetros por defecto.

```
from sklearn.model_selection import GridSearchCV
grid = GridSearchCV(SVC(),param_grid,refit=True,verbose=3)
```

- Luego se hace el ajuste con los datos de entrenamiento (*X\_train* y *y\_train*) para encontrar la mejor combinación de parámetros, este proceso puede tardar bastante tiempo dependiendo del tamaño de la base de datos.

```
grid.fit(X_train,y_train)
```

- Luego se obtienen los mejores parámetros con los siguientes comandos

```
grid.best_params_
grid.best_estimator_
```

- Para visualizar los valores obtenidos de la ejecución de *GridSearchCV* podemos usar la librería *mglearn*

```
import mglearn
plt.figure(figsize=(8, 8))
results = pd.DataFrame(grid.cv_results_)
scores = np.array(results.mean_test_score).reshape(12, 11)
mglearn.tools.heatmap(scores, xlabel='gamma', xticklabels=param_grid['gamma']
                      ,ylabel='C', yticklabels=param_grid['C'], cmap="viridis")
```

- Se obtiene la Figura 22.

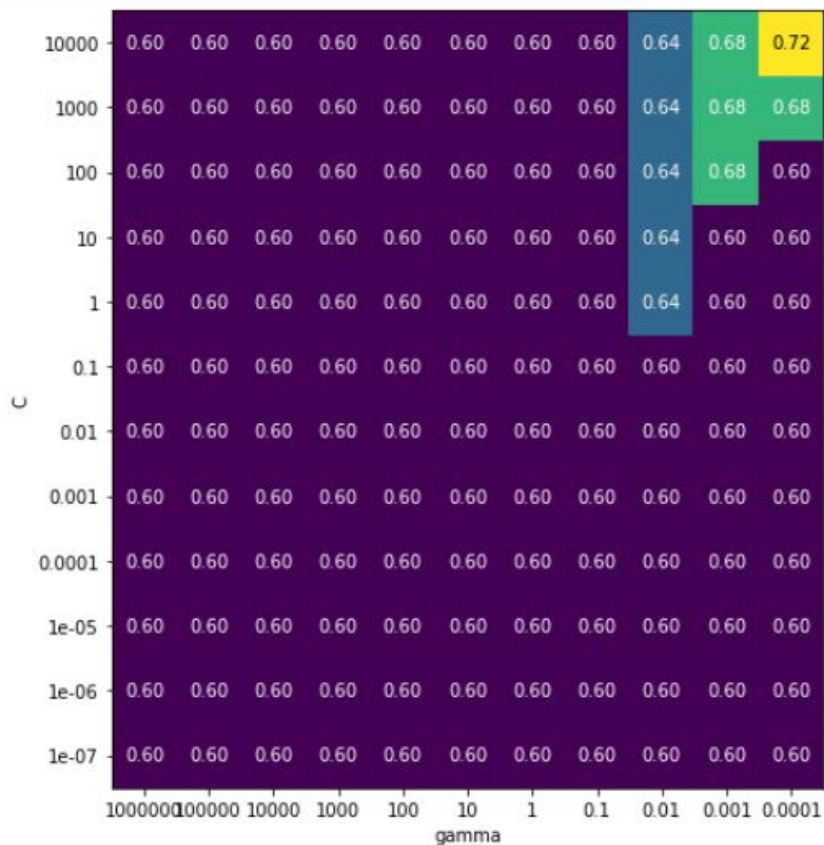


Figura 22. Matriz de valores de C y gamma.

- Se hacen las predicciones nuevamente con los parámetros obtenidos:

```
grid_predictions = grid.predict(X_test)
```

- Luego se consiguen las matrices de confusión con las predicciones obtenidas y los valores de *y\_test*

```
print(confusion_matrix(y_test, grid_predictions))
```

- Este proceso se realiza de forma iterativa cambiando los porcentajes de la base de datos dedicada al entrenamiento y evaluación del algoritmo, además de cambiando los valores de la semilla de 0 a 2.
- Los reportes de clasificación obtenidos se detallan en Tabla 5.



Tabla 5. Tabla de evaluaciones del algoritmo Support Vector Machine

RS	Train Size (%)	Test Size (%)	SVM (Average Precision/Recall)	C, Gamma	SVM + GridSearch (Average Precision/Recall)
RandomState= 0	85	15	0.44 / 0.67	1e-7,1000000	0.44 / 0.67
	80	20	0.56 / 0.75	1, 0.01	0.56 / 0.75
	75	25	0.49 / 0.70	1e-7,1000000	0.49 / 0.70
	70	30	0.44 / 0.67	1e-7,1000000	0.44 / 0.67
	65	35	0.41 / 0.64	1e-7,1000000	0.41 / 0.64
	60	40	0.39 / 0.62	100, 0.0001	0.53 / 0.50
	55	45	0.44 / 0.67	100, 0.0001	<b>0.72 / 0.61</b>
	50	50	0.42 / 0.65	1e-7,1000000	0.42 / 0.65
RandomState= 1	85	15	0.03 / 0.17	10, 0.001	0.87 / 0.33
	80	20	0.14 / 0.38	10, 0.001	0.79 / 0.50
	75	25	0.16 / 0.40	10, 0.0001	0.16 / 0.40
	70	30	0.17 / 0.42	10, 0.0001	0.17 / 0.42
	65	35	0.25 / 0.50	10, 0.0001	0.25 / 0.50
	60	40	0.25 / 0.50	10, 0.0001	0.25 / 0.50
	55	45	0.31 / 0.56	10, 0.0001	<b>0.77 / 0.61</b>
	50	50	0.36 / 0.60	10, 0.001	0.57 / 0.60
RandomState= 2	85	15	0.69 / 0.83	1000, 0.0001	0.69 / 0.83
	80	20	0.56 / 0.75	1000, 0.0001	<b>0.89 / 0.88</b>
	75	25	0.36 / 0.60	1000, 0.001	0.48 / 0.50
	70	30	0.34 / 0.58	1000, 0.0001	0.74 / 0.67
	65	35	0.41 / 0.64	1000, 0.0001	0.80 / 0.71
	60	40	0.34 / 0.62	1000, 0.0001	0.82 / 0.75
	55	45	0.37 / 0.61	1000, 0.0001	0.84 / 0.78
	50	50	0.42 / 0.65	10, 0.001	0.64 / 0.65

El código fuente para implementar los algoritmos descritos en esta sección pueden ser encontrados en el siguiente enlace:

[https://github.com/jptincopa/Sistema\\_Apoyo\\_Diagnostico\\_Neuropatia\\_ML/tree/master/Software/Algoritmos](https://github.com/jptincopa/Sistema_Apoyo_Diagnostico_Neuropatia_ML/tree/master/Software/Algoritmos)

#### **5.4 Consideraciones éticas**

El estudio fue aprobado por el Comité de ética de la Universidad Peruana Cayetano Heredia y del Hospital Cayetano Heredia. Tanto los pacientes (estudio observacional) como los médicos endocrinólogos (Estudio de usabilidad) firmaron consentimiento informado (Anexo 3).

## 6. Resultados

### 6.1. Estudio observacional

En total, se reclutaron 39 pacientes diabéticos. En la Tabla 6 se presentan las características generales de los pacientes.

Tabla 6. Características generales de los pacientes reclutados según diagnóstico de neuropatía

	No Neuropático	Neuropático	p
n	14	25	-
Sexo (Femenino=23)	12	11	0.01
Edad (años)	59.285 ± 11.309	57.720 ± 6.834	0.59
Peso (Kg)	64.200 ± 11.330	69.904 ± 13.475	0.19
Talla (cm)	153.857 ± 8.760	161.560 ± 6.420	0.003
Tiempo de Diabetes (años)	8.350 ± 7.180	11.280 ± 6.773	0.21

El diagnóstico fue establecido mediante la prueba de electromiografía. Prueba de t de dos colas. Chi cuadrado para frecuencia según sexo

### 6.2. Desarrollo del Modelo Predictivo

Para la creación del modelo predictivo se eligió el algoritmo de clasificación con mejor desempeño para predecir neuropatía que se muestra en la Tabla 7.

Tabla 7. Resultados del mejor desempeño de los algoritmos

Algoritmo Clasificador	Avg Precision	Avg Recall	Avg F1-Score
DT	0.81	0.78	0.78
RF	0.84	0.80	0.76
KNN	<b>0.94</b>	<b>0.93</b>	<b>0.93</b>
SVM + GridSearchCV	0.89	0.88	0.86

El mejor desempeño se obtuvo con el algoritmo *k nearest neighbours* (KNN), razón por la que fue el elegido para ser usado en el modelo predictivo. Tanto la precisión y el *recall* son parámetros importantes pero el concentrarse solo en uno de los no permitirá ver el panorama completo. El *Score F* (Avg F1 Score) que es

la media armónica de la precisión y el *recall* es una forma de resumir ambos parámetros de una manera más eficiente (42) siendo el mejor de ellos el 0.93 correspondiente al algoritmo KNN.

En la Figura 23 podemos ver cómo se comporta este algoritmo con las variables de Edad vs Tiempo de Diabetes y Edad vs Presión en *Hallux*.

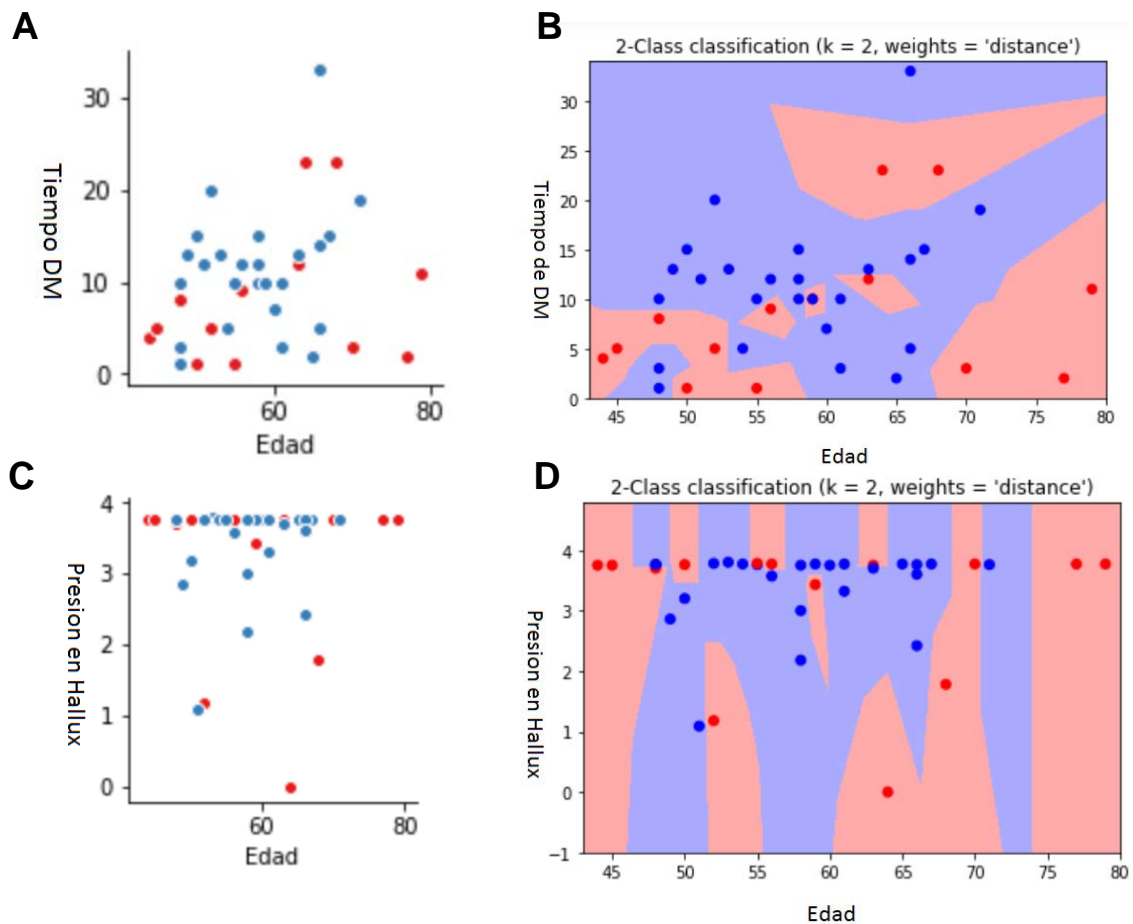


Figura 23. Diagrama de comportamiento del algoritmo KNN con las A: Edad vs tiempo de diabético B: Límites de decisión del clasificador KNN de variable edad vs. Tiempo de diabético C: Presión en el hallux vs edad D: Límites de decisión del clasificador KNN de variables presión en hallux y edad. Color azul: neuropático, color rojo: no neuropático

### 6.3. Desarrollo del Sistema de Apoyo al Diagnóstico

El sistema de apoyo al diagnóstico desarrollado cuenta con dos partes, el modelo predictivo generado con la recolección de datos de la fase observacional del estudio y una interfaz gráfica de usuario.

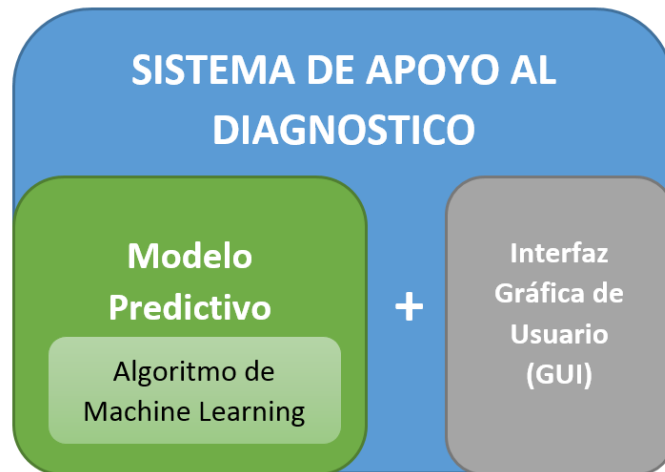


Figura 24. Componentes del sistema de apoyo al diagnóstico desarrollado

### 6.4. Interfaz Gráfica de Usuario (GUI)

La interfaz gráfica de usuario fue desarrollada en lenguaje Python en conjunto con las mismas herramientas usadas para el algoritmo (Página 32) adicionando la librería *PyQt* en su versión 5.0. Esta interfaz gráfica está dividida en módulos los cuales deben ser ejecutados en un orden específico para obtener la predicción del sistema.

- El primer módulo ( Figura 25) consiste en ingresar los datos personales del paciente como nombre y apellidos estos no entran dentro del modelo predictivo. Variables como sexo, edad, años de diabetes, peso, altura y calzado si son variables que se usan para realizar la predicción.

### 1. INGRESE LOS DATOS PERSONALES DEL PACIENTE

Nombres

Apellido Paterno

Apellido Materno

Sexo

Edad

Años de Db

Peso (Kg)

Altura (cm)

Talla Calzado

Figura 25. Primer módulo de la interfaz gráfica

El segundo módulo (Figura 26) determina el inicio y parada de la adquisición de datos de presión plantar, así mismo, la imagen de los pies con los puntos de medición va modificándose en tiempo real de acuerdo a la pisada del paciente y esos valores se almacenan en un archivo de CSV, si no se para la adquisición de datos esta tiene una barra de progreso que apaga automáticamente el almacenamiento de datos cuando llega al 100%. Se establecieron 4 niveles de presión cuyos valores y simbología en colores se muestran en la Figura 26.

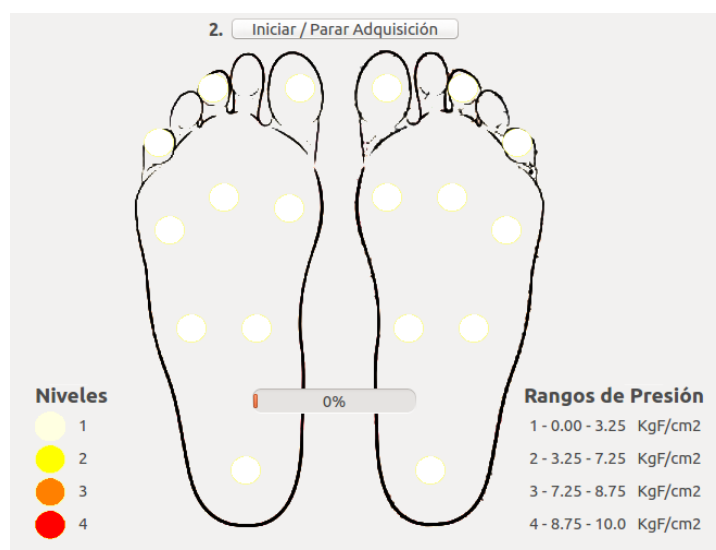


Figura 26. Segundo módulo de la interfaz gráfica.

El tercer módulo (Figura 27) muestra un resumen de los valores pico de presiones obtenidas del segundo módulo mostrando los resultados de cada pie.



3.

Pie Izquierdo		Pie Derecho	
Hallux:	0	Hallux:	0
3° Dedo:	0	3° Dedo:	0
5° Dedo:	0	5° Dedo:	0
1° Metatarso:	0	1° Metatarso:	0
3° Metatarso:	0	3° Metatarso:	0
5° Metatarso:	0	5° Metatarso:	0
Cuneiforme:	0	Cuneiforme:	0
Cuboide:	0	Cuboide:	0
Talon:	0	Talon:	0

Figura 27. Tercer módulo de la interfaz gráfica

El cuarto módulo (Figura 28) inicia el funcionamiento del modelo predictivo, del cual obtenemos 2 posibles resultados: **Normal** y **Neuropático**

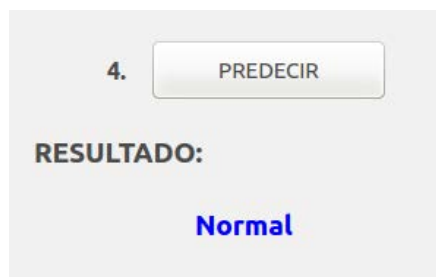


Figura 28. Cuarto módulo de la interfaz gráfica

El quinto módulo (Figura 29) nos permite exportar los datos ingresados y obtenidos en un archivo en formato PDF, el nombre del archivo está en base a los nombres y apellidos ingresados al sistema.

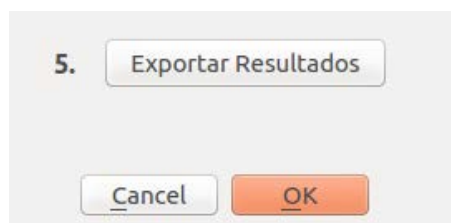


Figura 29. Quinto módulo de la interfaz gráfica

En el Anexo 4 podemos ver el Sistema de Apoyo al Diagnóstico completo.

## 6.5. Estudio de Usabilidad

Este sistema de apoyo al diagnóstico fue evaluado por 5 endocrinólogos del Hospital Cayetano Heredia (edad:  $39 \pm 5.62$ , media  $\pm$  DS) con un promedio de años de especialidad de  $8.2 \pm 3.25$ . Los doctores tuvieron opiniones favorables del sistema. Los resultados de la encuesta de usabilidad se muestran en la Figura 30. Además de la prueba los doctores aseveraron que el sistema puede ser de ayuda para visualizar que partes del pie necesitan alivianar la presión para poder recomendar plantillas especiales con el fin de evitar ulceraciones. Otra característica del sistema que causó buenos comentarios de parte de los endocrinólogos es el tener mapeadas las zonas por niveles ya que se puede analizar la pisada del paciente.

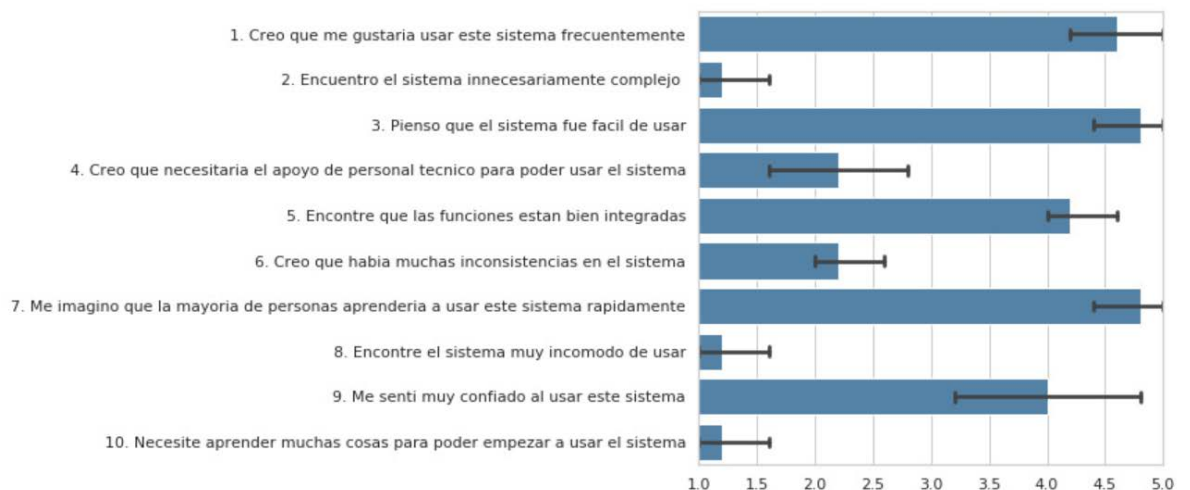


Figura 30. Resultados de la prueba de usabilidad del sistema de apoyo al diagnóstico.



## 7. Discusión

Este proyecto tuvo por objetivo desarrollar un sistema de apoyo al diagnóstico de neuropatía diabética basado en la lectura de puntos de presión plantar. Para esto, se logró desarrollar el sistema de adquisición de estos parámetros en base a sensores de presión colocados en plantillas, este sistema de adquisición se realizó usando *hardware* de código abierto, si bien existen desarrollos similares que usan el mismo tipo de sensores (69) éstos son usados para la recolección de data y posterior análisis, en esta tesis el mismo *hardware* usado para la recolección de datos se usa para trabajar en conjunto con un *software*, los cuales trabajan en tiempo real buscando tener una predicción inmediata al momento de usar las plantillas.

Se creó una base de datos con los valores de presión obtenidos del sistema de adquisición, con la cual se usaron diferentes algoritmos de *machine learning* para crear un modelo predictivo, existen antecedentes del uso de aprendizaje automático en valores de presión plantar (70) pero estos han sido del tipo No Supervisado buscando agrupar y clasificar los valores picos por zonas, en esta tesis se hizo uso de algoritmos de aprendizaje supervisado ya que tenemos una variable de salida conocida que es el diagnostico de neuropatía.

En *machine learning* se usa un porcentaje de los datos para entrenamiento del modelo y otro porcentaje para la evaluación del mismo. En la tablas de evaluación de los algoritmos utilizados vemos que usar igual cantidad de datos de entrenamiento y validación (50%-50%) nos otorgan desempeños bastante bajos. De igual manera, configuraciones donde se destina casi todos los datos a entrenar el modelo hacen que el desempeño no sea el mejor. Para el algoritmo elegido el porcentaje de entrenamiento con mejor desempeño fue de 65%, la cantidad de datos

usados para el entrenamiento fueron 25 y se dejó 35% para la evaluación (14 datos), así se obtuvo un *F1-Score* de 0.92.

Las clases de la base de datos creada no está distribuida de forma homogénea, es decir, no hay una separación visible de los valores respectivos a cada clase (Neuropático y No Neuropático) como se puede observar en la Figura 23 (A) y (B), *machine learning* permite encontrar patrones que la estadística convencional no puede hallar, en este caso vemos en la Figura 23 (C) y (D) como el algoritmo KNN logra clasificar de manera eficiente a pesar de que la distribución de los datos es heterogénea.

Cabe recalcar que este sistema no pretende ser una herramienta de diagnóstico *per se* sino un medio de ayuda para predecir neuropatía y/o alteraciones en la distribución de presión del pie que permitan al médico tratante hacer recomendaciones en el momento de la consulta.

La identificación temprana del paciente con neuropatía es esencial para evitar ulceraciones y amputaciones que terminen por socavar la calidad de vida del mismo. Lamentablemente, el diagnóstico de neuropatía es complejo. Si bien es cierto, el *gold standard* son los estudios de conducción nerviosa, estos son caros, altamente invasivos y en nuestro país están solo disponibles en grandes hospitales de referencia. El sistema desarrollado tiene costos bajos de producción, es de fácil implementación y podría ser llevado al primer nivel de atención. Además, la capacidad de medir en tiempo real la presión plantar permite al médico hacer recomendaciones en torno a qué calzado usar, qué plantillas escoger, etc. que finalmente tendrán un favorable impacto en la calidad de vida del paciente.

La evaluación de la aceptabilidad del sistema mostró que al total de médicos que lograron usar el sistema les gustaría usarlo de una manera más frecuente. Esto es notable pues los sistemas de apoyo al diagnóstico por lo general no son adoptados en clínica por una reticencia del médico a usarlos (58) y esto se da porque no se suele dar la importancia necesaria a la usabilidad del sistema ni se consideran las observaciones de los usuarios finales, en este caso, los médicos.

## 8. Conclusiones

- Se logró desarrollar un modelo predictivo basado en los valores de presión plantar obtenido de los pacientes reclutados, probando diferentes algoritmos de *machine learning* y eligiendo el que posee mayor desempeño.
- Se creó un sistema de apoyo al diagnóstico basado en este modelo predictivo y una interfaz gráfica que permite su uso.
- La usabilidad fue evaluada por médicos endocrinólogos, del mismo podemos concluir que es un sistema fácil de usar, cómodo y confiable el cual no necesita una capacitación extensa para ejecutarlo además no necesitar personal técnico de apoyo.

### 8.1. Limitaciones

Dentro del modelo predictivo no se tomó en cuenta la glucosa de los pacientes ya que tienden a ser valores volátiles. Una variable fisiológica que puede ser de mayor ayuda es la hemoglobina glicosilada ya que muestra los valores promedio de los últimos 3 meses, pero por limitaciones presupuestales no se pudo incluir en este estudio.

### 8.2. Sugerencias

El tamaño de muestra de pacientes con y sin diagnóstico de neuropatía es reducido en comparación a la mayoría de estudios donde se evalúan algoritmos de *machine learning* por lo que se requiere mayores estudios con mayor tamaño de muestra.

## 9. Referencias

1. Schreiber AK, Nones CF, Reis RC, Chichorro JG, Cunha JM. Diabetic neuropathic pain: Physiopathology and treatment. *World J Diabetes*. 2015;6(3):432–44.
2. Tesfaye S, Boulton AJM, Dickenson AH. Mechanisms and management of diabetic painful distal symmetrical polyneuropathy. *Diabetes Care*. 2013;36(9):2456–65.
3. Yagihashi S, Mizukami H, Sugimoto K. Mechanism of diabetic neuropathy: Where are we now and where to go? *J Diabetes Investig*. 2011;2(1):18–32.
4. Tesfaye S, Boulton AJM, Dickenson AH. Mechanisms and management of diabetic painful distal symmetrical polyneuropathy. *Diabetes Care*. 2013;36(9):2456–65.
5. Dobretsov M, Romanovsky D, Stimers JR. Early diabetic neuropathy: triggers and mechanisms. *World J Gastroenterol*. 2007;13(2):175–91.
6. Tesfaye S, Boulton AJM, Dyck PJ, Freeman R, Horowitz M, Kempner P, et al. Diabetic neuropathies: update on definitions, diagnostic criteria, estimation of severity, and treatments. *Diabetes Care*. 2010;33(10):2285–93.
7. Papanas N, Boulton AJM, Malik RA, Manes C, Schnell O, Spallone V, et al. A simple new non-invasive sweat indicator test for the diagnosis of diabetic neuropathy. *Diabet Med*. 2013;30(5):525–34.
8. Ziegler D, Siekierka-Kleiser E, Meyer B, Schweers M. Validation of a novel screening device (NeuroQuick) for quantitative assessment of small nerve fiber dysfunction as an early feature of diabetic polyneuropathy. *Diabetes Care*. 2005;28(5):1169–74.
9. Wang F, Zhang J, Yu J, Liu S, Zhang R, Ma X, et al. Diagnostic Accuracy of Monofilament Tests for Detecting Diabetic Peripheral Neuropathy: A Systematic Review and Meta-Analysis. *J Diabetes Res*. 2017;2017:8787261.
10. Wang F, Zhang J, Yu J, Liu S, Zhang R, Ma X, et al. Diagnostic Accuracy of Monofilament Tests for Detecting Diabetic Peripheral Neuropathy: A Systematic Review and Meta-Analysis. *J Diabetes Res*. 2017;2017:8787261.
11. Tan LS. The clinical use of the 10g monofilament and its limitations: A review. *Diabetes Res Clin Pract*. 2010;90(1):1–7.
12. Fawzy OA, Arafa AI, El Wakeel MA, Abdul Kareem SH. Plantar pressure as a risk assessment tool for diabetic foot ulceration in egyptian patients with diabetes. *Clin Med Insights Endocrinol Diabetes*. 2014;7:31–9.
13. Alam U, Asghar O, Azmi S, Malik RA. General aspects of diabetes mellitus. In: *Handbook of clinical neurology*. 2014. p. 211–22.
14. DeFronzo RA, Ferrannini E, Groop L, Henry RR, Herman WH, Holst JJ, et al. Type 2 diabetes mellitus. *Nat Rev Dis Prim*. 2015;1:15019.
15. Maffi P, Secchi A. The Burden of Diabetes: Emerging Data. In: *Developments in ophthalmology*. 2017. p. 1–5.

16. Ugarte-Gil C, Moore DAJ. [Tuberculosis and diabetes co-morbidity: an unresolved problem]. *Rev Peru Med Exp Salud Publica*. 2014;31(1):137–42.
17. The Emerging Risk Factors Collaboration. Diabetes Mellitus, Fasting Glucose, and Risk of Cause-Specific Death. *N Engl J Med*. 2011;364(9):829–41.
18. Hulme KD, Gallo LA, Short KR. Influenza Virus and Glycemic Variability in Diabetes: A Killer Combination? *Front Microbiol*. 2017;8:861.
19. Katsarou A, Gudbjörnsdóttir S, Rawshani A, Dabelea D, Bonifacio E, Anderson BJ, et al. Type 1 diabetes mellitus. *Nat Rev Dis Prim*. 2017;3:17016.
20. Kaur S, Pandhi P, Dutta P. Painful diabetic neuropathy: an update. *Ann Neurosci*. 2011;18(4):168–75.
21. Juster-Switlyk K, Smith AG. Updates in diabetic peripheral neuropathy. *F1000Research*. 2016;5.
22. Ang L, Jaiswal M, Martin C, Pop-Busui R. Glucose control and diabetic neuropathy: lessons from recent large clinical trials. *Curr Diab Rep*. 2014;14(9):528.
23. Albers JW, Pop-Busui R. Diabetic neuropathy: mechanisms, emerging treatments, and subtypes. *Curr Neurol Neurosci Rep*. 2014;14(8):473.
24. Smith AG, Singleton JR. Diabetic Neuropathy. *Contin Lifelong Learn Neurol*. 2012;18(1):60–84.
25. Tesfaye S, Boulton AJM, Dyck PJ, Freeman R, Horowitz M, Kempler P, et al. Diabetic neuropathies: update on definitions, diagnostic criteria, estimation of severity, and treatments. *Diabetes Care*. 2010;33(10):2285–93.
26. Mills KR. The basics of electromyography. *J Neurol Neurosurg Psychiatry*. 2005;76 Suppl 2(suppl 2):ii32-5.
27. Papanas N, Ziegler D. New vistas in the diagnosis of diabetic polyneuropathy. *Endocrine*. 2014;47(3):690–8.
28. Kumar S, Fernando DJ, Veves A, Knowles EA, Young MJ, Boulton AJ. Semmes-Weinstein monofilaments: a simple, effective and inexpensive screening device for identifying diabetic patients at risk of foot ulceration. *Diabetes Res Clin Pract*. 1991;13(1–2):63–7.
29. Barber MA, Conolley J, Spaulding CM, Dellon AL. Evaluation of Pressure Threshold Prior to Foot Ulceration. *J Am Podiatr Med Assoc*. 2001;91(10):508–14.
30. Fawzy OA, Arafa AI, El Wakeel MA, Abdul Kareem SH. Plantar pressure as a risk assessment tool for diabetic foot ulceration in egyptian patients with diabetes. *Clin Med Insights Endocrinol Diabetes*. 2014;7:31–9.
31. Cavanagh PR, Ulbrecht JS, Caputo GM. New developments in the biomechanics of the diabetic foot. *Diabetes Metab Res Rev*. 16 Suppl 1:S6–10.
32. Pai S, Ledoux WR. The compressive mechanical properties of diabetic and non-diabetic plantar soft tissue. *J Biomech*. 2010;43(9):1754–60.
33. Yu X, Yu G-R, Chen Y-X, Liu X-C. The Characteristics and Clinical Significance of Plantar Pressure Distribution in Patients with Diabetic Toe Deformity: A Dynamic Plantar Pressure Analysis. *J Int Med Res*.

- 2011;39(6):2352–9.
34. Buschmann WR, Jahss MH, Kummer F, Desai P, Gee RO, Ricci JL. Histology and Histomorphometric Analysis of the Normal and Atrophic Heel Fat Pad. *Foot Ankle Int.* 1995;16(5):254–8.
  35. Hsu C-C, Tsai W-C, Shau Y-W, Lee K-L, Hu C-F. Altered energy dissipation ratio of the plantar soft tissues under the metatarsal heads in patients with type 2 diabetes mellitus: A pilot study. *Clin Biomech.* 2007;22(1):67–73.
  36. Peña Quintal AE. Sensores de Fuerza Resistivos [Internet]. p. <http://sensoresyactuadorese7.blogspot.com/2013/07/>.
  37. Lema C. M. Diseño de microsensores piezoeléctricos mediante formulación robusta. 2016.
  38. Instrumentos M. Que son y para que sirven las galgas extensiométricas [Internet].
  39. Nascimento AM, Bellini CGP, Nascimento AM, Bellini CGP. Artificial intelligence and industry 4.0: The next frontier in organizations. *BAR - Brazilian Adm Rev.* 2018;15(4).
  40. Schäfer M. The fourth industrial revolution: How the EU can lead it. *Eur View.* 2018;17(1):5–12.
  41. Russell SJ, Norvig P. *Artificial Intelligence: A modern approach.* Prentice Hall Inc.; 1995. 965 p.
  42. Müller AC, Guido S. *Introduction to Machine Learning with Python* [Internet]. 2017. 394 p.
  43. Corlett RA. Features of artificial intelligence languages and their environments. *Softw Eng J.* 1986;1(4):159.
  44. Tanimoto S. *The elements of Artificial Intelligence. An Introduction using LISP.* Computer Science Press; 1986.
  45. Poole D, Mackworth A. *Python code for Artificial Intelligence: Foundations of computational agents* [Internet]. 2019. 221 p.
  46. Zhang W, Dechter R, Korf RE. Heuristic search in artificial intelligence. *Artif Intell.* 2001;129(1–2):1–4.
  47. Webster RW. Useful AI tools-a review of heuristic search methods. *IEEE Potentials.* 1991;10(3):51–4.
  48. Chamatkar A j. *An Artificial Intelligence for Data Mining.* IOSR J Comput Sci. 2014;53–7.
  49. Jothi N, Rashid NA, Husain W. Data Mining in Healthcare – A Review. *Procedia Comput Sci.* 2015;72:306–13.
  50. McCue ME, McCoy AM. The Scope of Big Data in One Medicine: Unprecedented Opportunities and Challenges. *Front Vet Sci.* 2017;4:194.
  51. Omar M, Mehmood A, Choi GS, Park HW. Global mapping of artificial intelligence in Google and Google Scholar. *Scientometrics.* 2017;113(3):1269–305.
  52. Shultz M. Comparing test searches in PubMed and Google Scholar. *J Med Libr Assoc.* 2007;95(4):442–5.
  53. Kaartemo V, Helkkula A. A Systematic Review of Artificial Intelligence and Robots in Value Co-creation: Current Status and Future Research Avenues. *J Creat Value.* 2018;4(2):211–28.
  54. Yu K-H, Beam AL, Kohane IS. Artificial intelligence in healthcare. *Nat*

- Biomed Eng. 2018;2(10):719–31.
55. Jiang F, Jiang Y, Zhi H, Dong Y, Li H, Ma S, et al. Artificial intelligence in healthcare: past, present and future. *Stroke Vasc Neurol.* 2017;2(4):230–43.
  56. Greene D, Cunningham P, Mayer R. Unsupervised Learning and Clustering. In: *Machine Learning Techniques for Multimedia.* Springer Berlin Heidelberg; p. 51–90.
  57. Bose E, Radhakrishnan K. Using Unsupervised Machine Learning to Identify Subgroups Among Home Health Patients With Heart Failure Using Telehealth. *CIN Comput Informatics, Nurs.* 2018;36(5):242–8.
  58. Porat T, Delaney B, Kostopoulou O. The impact of a diagnostic decision support system on the consultation: perceptions of GPs and patients. *BMC Med Inform Decis Mak.* 2017;17(1):79.
  59. Ramnarayan P, Kapoor RR, Coren M, Nanduri V, Tomlinson AL, Taylor PM, et al. Measuring the Impact of Diagnostic Decision Support on the Quality of Clinical Decision Making: Development of a Reliable and Valid Composite Score. *J Am Med Informatics Assoc.* 2003;10(6):563–72.
  60. Miller RA. Medical Diagnostic Decision Support Systems Review n Medical Diagnostic Decision Support Systems-Past, Present, and Future. *J Am Med Informatics Assoc.* 1994;1(1).
  61. Rigel DS, Friedman RJ, Kopf AW, Polsky D. ABCDE—An Evolving Concept in the Early Detection of Melanoma. *Arch Dermatol.* 2005;141(8):1032–4.
  62. Ercal F, Chawla A, Stoecker WV, Hsi-Chieh Lee, Moss RH. Neural network diagnosis of malignant melanoma from color images. *IEEE Trans Biomed Eng.* 1994;41(9):837–45.
  63. Benke K, Benke G. Artificial Intelligence and Big Data in Public Health. *Int J Environ Res Public Health.* 2018;15(12).
  64. Shen D, Wu G, Suk H-I. Deep Learning in Medical Image Analysis. *Annu Rev Biomed Eng.* 2017;19:221–48.
  65. OMS. Informe mundial sobre la diabetes [Internet]. 2016.
  66. OMS. Informe mundial sobre la diabetes. 2016.
  67. Kaku M, Vinik A, Simpson DM. Pathways in the diagnosis and management of diabetic polyneuropathy. *Curr Diab Rep.* 2015;15(6):609.
  68. Papanas N, Ziegler D. New vistas in the diagnosis of diabetic polyneuropathy. *Endocrine.* 2014;47(3):690–8.
  69. Razak AHA, Zayegh A, Begg RK, Wahab Y. Foot plantar pressure measurement system: a review. *Sensors (Basel).* 2012;12(7):9884–912.
  70. Bennetts CJ, Owings TM, Erdemir A, Botek G, Cavanagh PR. Clustering and classification of regional peak plantar pressures of diabetic feet. *J Biomech.* 2013;46(1):19–25.



## Anexos

### Anexo 1. Escala de Usabilidad

	<b>Totalmente en Desacuerdo</b>				<b>Totalmente de Acuerdo</b>
1. Creo que me gustaría usar este sistema frecuentemente	1	2	3	4	5
2. Encuentro el sistema innecesariamente complejo	1	2	3	4	5
3. Pienso que el sistema fue fácil de usar	1	2	3	4	5
4. Creo que necesitaría el apoyo de personal técnico para poder usar el sistema	1	2	3	4	5
5. Encontré que las funciones de este sistema estaban bien integradas	1	2	3	4	5
6. Creo que había muchas inconsistencias en el sistema	1	2	3	4	5
7. Me imagino que la mayoría de personas aprendería a usar este sistema rápidamente	1	2	3	4	5
8. Encontré el sistema muy incómodo de usar	1	2	3	4	5
9. Me sentí muy confiado al usar este sistema	1	2	3	4	5
10. Necesité aprender muchas cosas para poder empezar a usar este sistema	1	2	3	4	5

Adaptado de Brooke J. In: SUS: A "quick and dirty" usability scale. Jordan PW, Thomas B, Werdmeester BA, McClelland AL, editor. Usability Evaluation in Industry. London: Taylor and Francis; 1996.



**Anexo 4. Vista completa de la interfaz gráfica del sistema de apoyo al diagnóstico**

### SISTEMA DE APOYO AL DIAGNOSTICO DE NEUROPATIA DIABETICA

2. Iniciar / Parar Adquisición



**Niveles**

- 1
- 2
- 3
- 4

**Rangos de Presión**

1 - 0.00 - 3.25 KgF/cm2

2 - 3.25 - 7.25 KgF/cm2

3 - 7.25 - 8.75 KgF/cm2

4 - 8.75 - 10.0 KgF/cm2

3. Mostrar Informe

	Pie Izquierdo	Pie Derecho
Hallux:	-	-
3° Dedo:	-	-
5° Dedo:	-	-
1° Metatarso:	-	-
3° Metatarso:	-	-
5° Metatarso:	-	-
Cuneiforme:	-	-
Cuboide:	-	-
Talon:	-	-

**1. INGRESE LOS DATOS PERSONALES DEL PACIENTE**

Nombres

Apellido Paterno

Apellido Materno

Sexo

Edad

Años de Db

Peso (Kg)

Altura (cm)

Talla Calzado

4. PREDECIR

**RESULTADO:**

---

5. Exportar Resultados

Cancel
OK