



**Diseño de un sistema para la visualización del nivel de glucosa de un grupo de
pacientes con diabetes**

Anderson Mosquera Ruiz

Universidad EAN

Facultad de Ingeniería

Ingeniería de Sistemas

Bogotá, Colombia

2023

Contenido

	Pág.
Objetivos	5
Objetivo general	5
Objetivos específicos	5
Planteamiento del Problema	6
Justificación	8
Marco de Referencia	9
La diabetes	9
Herramienta de Visualización (Power BI)	14
Predicciones usando técnicas de machine learning	16
Plan de implementación	23
Conclusiones	29
Referencias	30

Lista de Figuras

	Pág.
Figura 1. Arquitectura de una red neuronal.	19
Figura 2. Pestaña de selección de pacientes del Dashboard de Power BI.....	24
Figura 3. Pestaña de visualización del nivel de glucosa.	25
Figura 4. Pestaña de visualización de la predicción del nivel de glucosa.....	25
Figura 5. Código de creación de la red neural usando Python.....	27

Introducción

La diabetes es una enfermedad en la que el cuerpo no disuelve adecuadamente la glucosa (Jain, Joshi, Mohanty, 2021, p. 1). Se han documentado 3 tipos de diabetes: Diabetes tipo 1, el cuerpo no genera insulina para la regulación de la glucosa. Diabetes tipo 2, el cuerpo genera cantidades de insulina que no son suficientes para la disolución de la glucosa. Diabetes gestacional, el cuerpo de la madre embarazada no produce insulina (Jain et al., 2021, p. 4).

De acuerdo con la Federación Internacional de Diabetes 537 millones de personas en el mundo padecen diabetes (IDF ATLAS 2021), lo que representa un gasto público de 966 mil millones de dólares americanos solo en 2021. Los tratamientos para la diabetes se basan en el ajuste de la terapia con medicamentos, la rutina y/o las tendencias dietarias del paciente, para lo cual el personal médico debe tener información actualizada y de calidad sobre los niveles de glucosa históricos de sus pacientes.

En este trabajo se propone diseñar un sistema para la visualización de los niveles de glucosa de los pacientes con diabetes que permita ver los niveles de glucosa reales y predecir los niveles de glucosa futuros mediante el uso de técnicas de *machine learning* incluidas en el sistema de visualización. Para ello se propone realizar una revisión bibliográfica de la diabetes como enfermedad, construir un *dashboard* en Power BI que muestre la evolución en el tiempo del nivel de glucosa de los pacientes y diseñar un código de Python dentro del *dashboard* de Power BI que utilice técnicas de *machine learning* para predecir el nivel de glucosa de los pacientes.

Objetivos

Objetivo general

Diseñar un sistema para la visualización del nivel glucosa en el tiempo de un grupo de pacientes con diabetes que incluya datos reales y predicciones realizadas usando técnicas de *machine learning*.

Objetivos específicos

- Realizar una revisión bibliográfica de la diabetes, los mecanismos de medición de la glucosa en sangre y los métodos de predicción usando técnicas de *machine learning*.
- Construir un *dashboard* en Power BI que muestre la evolución en el tiempo del nivel de glucosa de los pacientes.
- Diseñar un código de Python dentro del *dashboard* de Power BI que utilice técnicas de *machine learning* para predecir el nivel de glucosa de los pacientes.

Planteamiento del Problema

En Colombia más de 3 millones de personas padecen diabetes (IDF, 2021), lo que convierte al país en el segundo más afectado por esta enfermedad en Suramérica y convierte también a esta enfermedad en una de las primeras cinco causas conocidas de muerte médica en el país (Mosquera-Ruiz y García-Jaramillo, 2022). Adicionalmente, la población con diabetes fue una de las más afectadas durante la pandemia del COVID-19, generando una tasa de muerte del 82% mayor a la de la población sin esta enfermedad (Saha et al. 2021). En el mundo existen más de 530 millones de personas entre 20 y 79 años con diabetes y se estima que para el 2030 este número ascenderá en más de 100 millones (IDF ATLAS 2021); una cifra alarmante es la cantidad de muertes asociadas, en 2021 se tuvo un equivalente a 1 fallecimiento cada 5 segundos por esta enfermedad (Mosquera-Ruiz y García-Jaramillo, 2022) para un total de 6.7 millones de personas.

Se ha logrado evidenciar que además de un diagnóstico temprano de la enfermedad es necesario un adecuado control de la diabetes para fin reducir la progresión de complicaciones crónicas, y en línea con esto, la creciente carga financiera para los sistemas de atención de la salud (Castillo et al. 2019). Para lograr el control de los niveles de glucosa se requiere el uso de fármacos y en algunos casos terapias de insulina en combinación con dietas balanceadas, ejercicio y un monitoreo (mediante glucometría capilar, monitorización continua de glucosa o una combinación de ambas).

La monitorización de los niveles de glucosa se realiza a través de dos grupos principales de dispositivos médicos:

- El sistema de monitoreo capilar requiere de una tira radioactiva en la que se coloca una gota de sangre para determinar el nivel de glucosa mediante su reactividad. Este tipo de glucómetros se usa en los tiempos entre comidas y antes de dormir o después de despertar.
- El sistema de monitoreo continuo requiere un sensor que se adhiere a la piel (antebrazo, abdomen, pierna) y toma muestras de glucosa intersticial en intervalos definidos de tiempo, los datos del sensor se visualizan en una pantalla y algunos de estos dispositivos cuentan con su propio software para descarga y visualización de los datos (Gómez et al., 2019).

“El paciente que cuenta con Sistemas de monitorización continua de glucosa generalmente acude a su cita médica con el dispositivo y allí se realiza la descarga de datos y análisis por parte del especialista, lo que toma un tiempo considerable de la consulta” (Mosquera-Ruiz y García-Jaramillo, 2022).

Justificación

En este trabajo se propone el diseño de un sistema de visualización del nivel de glucosa de pacientes con diabetes usando Power BI. Esta es una herramienta innovadora, fácil de usar, de distribuir y que permite el uso de pestañas, diversas visualizaciones, el uso de colores, ventanas de descripción detallada, múltiples pestañas, links, etc. Los reportes y *dashboards* creados usando Power BI pueden ser vistos en la web mediante computadores, *tablets*, teléfonos celulares y otros equipos de comunicaciones.

Diseñar el sistema de visualización usando Power BI permitirá que los pacientes y personal médico puedan acceder a la información sin necesidad de comprar licencias o hacer cursos especializados para entender cómo usar e interactuar con el sistema de visualización. Por consiguiente, los beneficiarios serán el personal médico y los pacientes con diabetes que usen la herramienta para observar sus niveles de glucosa en el tiempo y puedan generar alertas a partir de las predicciones realizadas mediante la inclusión de pronósticos usando modelos de *machine learning*.

Reducir los tiempos de descarga de datos e introducir un sistema de visualización de la información que incluya tendencias por días, horas y permita visualizar rápidamente los datos más relevantes en los periodos solicitados, permitirá que el personal clínico pueda tomar más tiempo en determinar el avance de la terapia y formular nuevas estrategias para el control de los niveles de glucosa de sus pacientes de acuerdo a su criterio, lo que resultará en una mayor eficacia de estos tratamientos y una mejor calidad de vida para los pacientes.

Marco de Referencia

En esta revisión bibliográfica se abordará la diabetes como enfermedad haciendo una introducción el descubrimiento de la enfermedad y los trabajos previos que la acompañaron, el mecanismo de visualización de datos (Power BI) y las técnicas de *machine learning* que pueden ser usadas para realizar predicciones

La diabetes

La diabetes mellitus es una enfermedad que ha sido reconocida desde la antigüedad, el primer registro de un síntoma característico de la diabetes, se encuentra en un papiro egipcio que data de alrededor del 1550 a.C. y describe un estado en el que el cuerpo elimina grandes cantidades de orina. En el siglo II d.C., Aretaeus de Capadocia describió un estado de salud en el que se elimina mucha orina, se tiene mucha sed y hay poca supervivencia usando por primera vez el termino diabetes (Holt, R. I. G. et al., 2017).

En Europa, la diabetes fue descuidada hasta que Thomas Willis (1621-1675) escribió una publicación en la que se refiere a la diabetes como una enfermedad rara entre los antiguos practicantes de medicina, pero que se ha vuelto más común en la época actual debido a la buena comida y el consumo de vino sin medida. Describió la orina de los pacientes con diabetes como casi tan dulce como el azúcar o la miel (Holt, R. I. G. et al., 2017); en 1776 Matthew Dobson concluyó que la contenía azúcar y que esta se excretaba en la orina a través de los riñones. Hacia inicios del 1800 John Rollo aplicó el adjetivo "mellitus" (de la palabra latina que significa "miel") a la diabetes.

John Rollo pensó que el azúcar se formaba en el estómago a partir de vegetales, y concluyó que la solución obvia era una dieta de alimentos de origen animal. En 1815 Michel Chevreul demostró que el azúcar en la orina diabética era en realidad glucosa y Claude Bernard hacia mediados del siglo XIX aclaró el mecanismo de metabolismo de la glucosa: hasta ese momento se pensaba que el azúcar solo podía ser sintetizado por las plantas y que el metabolismo animal descomponía estas sustancias al entrar en contacto con ellas a través de la alimentación; Bernard descubrió que la glucosa estaba presente en la sangre de todos los animales aun en condiciones normales y que el hígado contenía una sustancia similar al almidón que podía convertirse fácilmente en glucosa; de estos descubrimientos derivó la teoría glucogénica (Holt, R. I. G. et al., 2017) en la que enunciaba que el azúcar absorbido por el intestino se convertía en el hígado y luego era liberado constantemente en la sangre.

Hacia finales del siglo XIX Oskar Minkowski y Josef von Mering propusieron que el páncreas eliminó una toxina diabetogénica o produjo una secreción interna que controlaba el metabolismo de los carbohidratos (incluida la glucosa); Gustave Laguesse apoyado en el trabajo de Paul Langerhans sugirió que esta secreción interna era producida por islas de células y en 1909 Jean de Meyer llamó esta secreción interna insulina (derivado del vocablo latín para isla).

Hacia finales del siglo XIX Etienne Lancereaux identificó claramente dos grandes grupos de pacientes con diabetes delgada y diabetes grasa (respectivamente del francés *Le Diabète maigre et le diabète gras*) (Holt, R. I. G. et al., 2017). Según lo investigado por Lancereaux la Diabetes delgada era más común en personas jóvenes y tenía un inicio agudo, mientras que la diabetes grasa se presentaba en personas de mediana edad con sobrepeso, lo que representaba menores tasas de mortalidad y mayores complicaciones microvasculares (Levy, D., 2016). Los trabajos y publicaciones de Harold Himsworth a mediados del siglo XX permitieron distinguir entre diabetes tipo 1 (sensible a la insulina) y diabetes tipo 2 (resistente a la insulina) (Levy, D., 2016).

Hoy se sabe que la homeostasis de la glucosa es un estado de equilibrio entre comidas y se mantiene por varias hormonas interrelacionadas, especialmente la insulina, el glucagón, la adrenalina, el cortisol y las incretinas (Dunning, T., 2012), también juegan un papel importante el tipo de alimentos ingeridos, el ejercicio y los hábitos diarios.

También se sabe que la insulina se secreta en 2 fases, la primera ocurre justo después de la ingestión de comidas y continua durante los siguientes 10 a 15 minutos; en la segunda fase la secreción de insulina se mantiene hasta que se restablece los niveles normales de glucosa en la sangre (Dunning, T., 2012). En los pacientes con diabetes tipo 1 la primera fase de secreción de insulina no es tan eficiente, con lo cual el paciente tiene altos niveles de glucosa en sangre que pueden llevar a trastornos tales como la aterosclerosis, hipertrigliceridemia, coagulopatías, disfunción endotelial, hipertensión y la hiperglucemia crónica (Dunning, T., 2012).

Hacia mediados del siglo XX se estableció la insulina como un tratamiento efectivo y seguro de la diabetes por parte de Frederick Banting y Charles Herbert Best, esto alentó la producción industrial de insulina, lo cual redujo la mortalidad asociada a la diabetes. El trabajo combinado de Dorothy Hodgkin y continuado Frederick Sanger permitió la producción de la insulina biosintética basada en la secuencia del gen de insulina, este avance ha permitido mejoras en la tecnología del tratamiento como el uso de la bomba de insulina, el trasplante islotes y páncreas, pruebas de insulina mínimamente invasivas, entre otras (Levy, D., 2016).

Existen diferentes tipos de insulina (Holt, T. y Kumar, S., 2015):

- Insulinas animales: fabricadas hasta la década de 1980 usando extractos purificados del páncreas de vacas y cerdos.
- Insulinas de acción corta: también se conocida como insulina soluble, tiene una duración de acción de entre 6 y 8 horas y debe inyectarse unos 30 minutos antes de las comidas.

- Análogos de insulina de acción muy rápida: se pueden tomar con la una comida porque se absorben casi inmediatamente.
- Insulinas de acción intermedia: generalmente tienen una proteína básica añadida a ellas para retrasar su acción; tienden a tener una duración de acción de entre 8 y 10 horas después de la inyección.
- Suspensiones de insulina zinc de acción prolongada: se preparan agregando exceso de iones de zinc a la insulina; generalmente se administran a la hora de acostarse.
- Análogos de insulina de acción prolongada: proporcionan insulina de hasta 24 horas de duración, tiene un bajo riesgo de hipoglucemia y pueden aplicados en las noches.

La selección de insulina para un paciente se realiza teniendo en cuenta las circunstancias particulares del paciente, su dieta y sus preferencias en términos de elección del dispositivo y frecuencia de inyección (Holt, T. y Kumar, S., 2015)

La insulina también puede ser inyectada por varios mecanismos (Holt, T. y Kumar, S., 2015):

- Insulina administrada con una jeringa: El paciente utiliza una jeringa para extraer la dosis adecuada de un frasco y luego inyectarla en su piel. Con estas se corre el riesgo de administrar cantidades incorrectas de insulina, también puede llegarse a necesitar la ayuda de un experto para su administración.
- Lapiceros de insulina: son dispositivos sofisticados que administran dosis medidas de insulina desde un cartucho precargado y que se desecha una vez se ha utilizado.
- Bombas de insulina: administran insulina por vía subcutánea durante 24 horas a intervalos regulares definidos por el paciente y el personal médico.

Otros mecanismos de tratamiento de la diabetes son citados por Dods, R. F. (2013):

Medicamentos orales antidiabéticos; Inhibidores basados en incretina; derivados de amilina (Pramlintide); Activadores de la glucoquinasa (GKA): Compuestos antidiabéticos potenciales; Inhibidores de la α -glucosidasa; Inhibidores de la 11 β -hidroxiesteroide deshidrogenasa tipo 1.

Otro método de tratamiento es la dieta y los hábitos del paciente. Este es uno de los más difíciles de conseguir ya que implica cambios en el estilo de vida del paciente y la incorporación y eliminación

de algunos hábitos y acciones en la vida de los pacientes (Fonseca, V., 2010); los objetivos principales de esta dimensión del tratamiento son (Levy, D., 2016):

- Equilibrar las recomendaciones dietéticas específicas con las restricciones de la terapia con insulina
- Minimizar la hipoglucemia
- Reducir los efectos metabólicos y cardiovasculares adversos de la resistencia adquirida a la insulina (diabetes tipo 1)
- Reducir los riesgos de complicaciones microvasculares manteniendo los niveles más bajos practicables y seguros de A1C y glucosa.

Entre las complicaciones a la salud asociadas a la diabetes se encuentra la hipoglucemia, que es un descenso de los niveles de glucosa que puede conducir a deterioro cognitivo, confusión, coma. Y si se prolongan durante horas pueden causar la muerte o daño cerebral permanente; los episodios de hipoglucemia pueden ocurrir después de la administración de grandes cantidades de insulina, lo que provoca una disminución de los niveles de glucosa en la sangre y en el cerebro (Krentz, A., 2004); debido a que el cerebro consume el 20% de la glucosa total del cuerpo y no tiene un mecanismo para sintetizarla se hace esencial que exista un mínimo de concentración de glucosa en la sangre y que el cerebro continúe con su actividad normal (Holt, R. I. G. et al.2017). Los eventos de hipoglucemia se observan con mayor frecuencia en diabetes tipo 1 y pueden deberse, además del exceso de aplicación de insulina a comidas perdidas o inadecuadas, ejercicio inesperado, cambio en la dosis o tipo de insulina suministrada (Holt, T. y Kumar, S., 2015).

Otras complicaciones asociadas a la diabetes pueden incluir la etiopatogenia de las lesiones del pie diabético; úlceras del pie; neuroartropatía de Charcot; disfunción eréctil masculina; disfunción sexual femenina; contracepción; terapia de reemplazo hormonal; patofisiología; fisiopatología de la enteropatía diabética en humanos; NAFLD y el síndrome metabólico; EHNA; cambios vasculares; infecciones; yatrogénico; enfermedad musculoesquelética en la diabetes; riesgo de cáncer;

hiperglucemia versus hiperinsulinemia; exposición a agentes hipoglucemiantes; infecciones del tracto respiratorio y tuberculosis; infecciones del tracto urinario; infecciones intraabdominales; infecciones superficiales de la piel y de los tejidos blandos (Holt, R. I. G. et al. 2017), (Shaw, K. M. y Cummings, M. H., 2012)

Herramienta de Visualización (Power BI)

“Power BI es una colección de servicios de software, aplicaciones y conectores que trabajan juntos para convertir los orígenes de datos no relacionados en información coherente, visualmente envolvente e interactiva” (Microsoft, 2023d). Power BI nos permite conectar datos almacenados en una o varias hojas de cálculo de Excel con tablas almacenadas en una base de datos y archivos planos csv o txt, para que sean modelados (conectados entre sí, transformados, agrupados, etc.) y luego sean mostrados en visualizaciones como gráficos de barras, líneas de tiempo, mapas, gráficos de tortas y hasta gráficos creados por el usuario a través de códigos de Python o R.

Power BI tiene tres elementos principales usados en todo el proceso de creación de reportes, visualizaciones, *dashboards* o *apps*:

- Una aplicación de escritorio de Windows denominada Power BI Desktop: usada para diseñar el flujo de datos, desde su obtención, transformación y carga, hasta el diseño de la visualizaciones.
- Un servicio de software como servicio (SaaS) en línea denominado servicio Power BI: utilizado para compartir y observar las visualizaciones de los datos.
- Aplicaciones móviles de Power BI para dispositivos Windows, iOS y Android: usados de manera similar al servicio de Power BI, para compartir y visualizar los datos.

Normalmente un usuario de Power BI comienza un flujo de trabajo conectándose a orígenes de datos usando Power BI Desktop y creando un visualizaciones en un informe. Después publica ese informe desde Power BI Desktop en el servicio Power BI, le hace ediciones finales y comparte este informe con los usuarios finales para que lo puedan ver en el servicio Power BI y/o en dispositivos móviles (Microsoft, 2023b).

Utilizando la integración de Python en Power BI un usuario puede aprovechar las librerías visuales de Python como matplotlib, seaborn o librerías de *machine learning* como tensorflow (Absent Data., s.f.).

Para usar visualizaciones de Python, Power BI hace un proceso de transformación de los datos antes de mostrarlos (Zaczyński B., 2022), este proceso incluye la utilización de archivos temporales así:

- Power BI a CSV: en este paso Power BI genera un archivo plano que será entregado al intérprete de Python para procesar de acuerdo con el código que el usuario escribió en la visualización
- CSV a Python: en este paso el archivo plano se le pasa al código de Python para producir una salida que será visualizada
- Python a CSV: en este paso el código de Python genera la salida que será visualizada en Power BI.
- CSV a Power BI: en este paso el csv de salida del código de Python se entrega a Power BI para visualizarlo.

Las transformaciones anteriormente señaladas generan una serie de limitaciones sobre todo en tiempos de espera y tamaño de datos: las visualizaciones están limitadas a tiempos de procesamiento inferior a 5 minutos, solo se pueden graficas las primeras 150.000 filas de un conjunto de datos, el conjunto de datos de entrada debe tener un tamaño inferior a 256 megabytes y las cadenas de caracteres inferiores a 32.766 caracteres serán cortadas, las visualizaciones Python no pueden filtrar otras visualizaciones, cada vez que se genera un filtro o interacción desde otro objeto visual el código

de Python que genera la visualización debe ser procesado nuevamente siguiendo el flujo descrito anteriormente (Microsoft, 2023a).

Para sobreponerse a estas limitaciones se puede considerar que hay tres maneras de usar código de Python desde Power BI (Zaczyński B., 2022):

- Importar un datos: mediante código de Python se puede realizar la obtención de un conjunto de datos en cualquier formato y mediante el código de Python suministrado se pueden realizar algunas transformaciones a los datos originales antes de que se carguen al modelo de datos de Power BI.
- Transformar y cargar datos: se pueden realizar transformaciones basadas en código de Python antes de realizar visualizaciones sobre datos cargados desde cualquier fuente.
- Visualizar datos: se pueden crear visualizaciones usando código de Python una vez los datos han sido cargados y transformados. Esta forma de usar código de Python en Power BI tiene las limitaciones descritas en el párrafo precedente.

Predicciones usando técnicas de *machine learning*

Una serie temporal se define de acuerdo con Chou, J. S. y Tran, D. S. (2018) como una secuencia ordenada de valores que se registran a intervalos iguales de tiempo; estos valores pueden tener una distribución interna que es aleatoria o que responde a la relación de unas o varias variables que determinan los valores de la serie temporal; de acuerdo con Comito, C. y Pizzuti, C. (2022) y Srihari, A. et al. (2020) una serie temporal se puede descomponer en:

- la tendencia, que considera el patrón armónico de la serie
- la estacionalidad, es el cambio cíclico de las variables internas de la serie temporal
- el residual o ruido, que es la parte inexplicable de la serie temporal

Usando estos componentes se puede modelar el mecanismo estocástico que genera la serie temporal y hacer un análisis para generar pronósticos o predicciones (Srihari, A. et al., 2020).

Predecir el futuro de una serie temporal es una tarea muy importante en muchos campos como el clima, consumo de energía, índices financieros, ventas minoristas, monitoreo médico, detección de anomalías, predicción de tráfico, estadística aplicada, procesamiento de señales, reconocimiento de patrones, econometría, predicción de terremotos, astronomía y en cualquier dominio de la ciencia aplicada y la ingeniería que involucre mediciones temporales (Chou, J. S. y Tran, D. S., 2018), (Lara-Benítez, P. et al., 2021). Para realizar la predicción se usa la premisa de que la información se repetirá en un futuro próximo y se apoya en los componentes descritos anteriormente.

Para realizar predicciones en series temporales se siguen cuatro pasos: (1) la recopilación de datos, (2) el preprocesamiento de datos, (3) el entrenamiento de modelos y (4) las pruebas de modelos (Chou, J. S. y Tran, D. S., 2018); en el preprocesamiento se eliminan anomalías y se seleccionan características de la serie temporal que podrían afectar el rendimiento de los modelos (Sharma, E., 2018); durante el entrenamiento de los modelos se obtiene la estructura y el patrón subyacente de los datos observados con el fin de realizar las predicciones; en la evaluación de modelos se utilizan técnicas estadísticas para comparar la salida de los modelos con los datos reales medidos para determinar la efectividad de los mismos.

Se ha encontrado que los datos de entrada, las variables y el horizonte de predicción afectan la precisión de los modelos (Sharma, E., 2018); por ejemplo, entre más largo el horizonte de predicción (que tanto en el futuro se quiere predecir) se tienen errores de predicción más grandes (Oh, S. et al., 2015).

En ocasiones se usa una o la combinación de algunas de las técnicas descritas a continuación para aumentar la precisión de la predicción.

La regresión lineal es una técnica de aprendizaje supervisada basada en conceptos estadísticos que modela matemáticamente la relación entre una variable dependiente y una o más variables

independientes (Chou, J. S. y Tran, D. S., 2018), esta relación entre variables puede usarse para predecir valores futuros de las variables independientes.

Dentro de las técnicas de regresión también se encuentran los modelos de media móvil integrada autorregresiva (ARIMA por sus siglas en inglés) (Chou, J. S. y Tran, D. S., 2018), en estos modelos se obtiene el valor predicho de la serie utilizando sus valores pasados y se tiene en cuenta la variación del promedio debido a la estacionalidad o multivariabilidad de la serie temporal (Comito, C. y Pizzuti, C., 2022); debido a sus estructuras relativamente simples, los investigadores confirman la alta eficiencia del método en términos de cálculos (Oh, S. et al., 2015).

Las máquinas de soporte vectorial utilizan datos de entrenamiento para clasificar binariamente cada punto, utilizando las máquinas de soporte vectorial se pueden generar regresiones no lineales (Chou, J. S. y Tran, D. S., 2018).

La inteligencia artificial es un campo de las ciencias de la computación cuyo objetivo es construir computadoras que imiten la inteligencia humana al realizar tareas, el aprendizaje automático (*Machine Learning*) y el aprendizaje profundo (*Deep Learning*) son dos áreas del campo de la inteligencia artificial en la cuales se utilizan datos, algoritmos y modelos estadísticos para extraer inferencias y patrones que permitan a las computadoras aprender y adaptarse automáticamente a nuevos estímulos (Comito, C. y Pizzuti, C., 2022).

Los algoritmos de aprendizaje profundo incluyen redes neuronales profundas, aprendizaje de refuerzo profundo, redes neuronales recurrentes y redes neuronales convolucionales. (Comito, C. y Pizzuti, C., 2022), estas pueden modelar problemas con múltiples variables de entrada (Srihari, A. et al., 2020) por lo cual son útiles en el campo de la predicción de series temporales.

Las Redes neuronales recurrentes (RNN por sus siglas en inglés) son aptas para analizar datos secuenciales, están organizadas en capas o nodos a través de los cuales se transmite la información de manera simultánea en pasos discretos de tiempo. Una variación de las redes neuronales recurrentes son las redes de larga duración de termino corto (LSTM por sus siglas en inglés) las

cuales tienen nodos ocultos que permiten retener la salida de iteraciones anteriores para así mejorar su precisión de predicción (Comito, C. y Pizzuti, C., 2022), esta característica hace que las redes de larga duración de termino corto sean las redes neuronales más comúnmente utilizadas en la predicción de series temporales (Lara-Benítez, P. et al., 2021), en la siguiente figura se puede apreciar la arquitectura de una de estas.

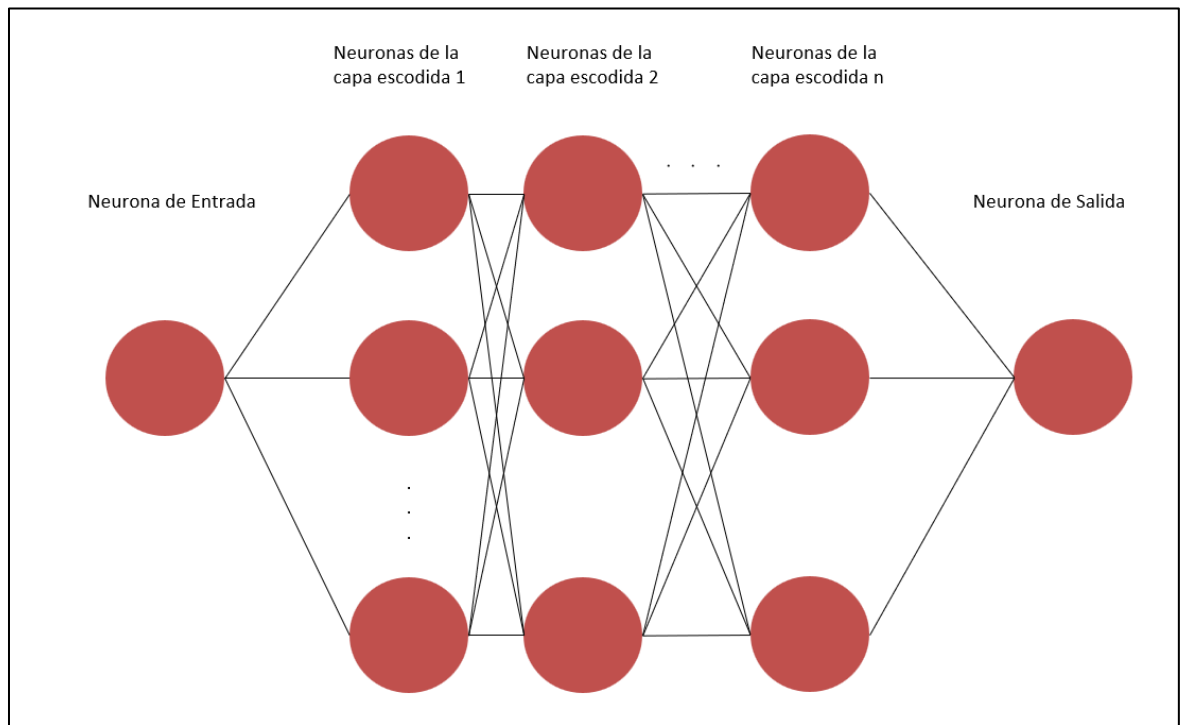


Figura 1. Arquitectura de una red neuronal.

Fuente: El Autor.

Las redes neuronales convolucionales (CNN por sus siglas en inglés) son redes de alimentación hacia adelante que utilizan tres capas principales: la capa convolucional, la capa de agrupación y la capa conectada a la salida. El nombre de este tipo de redes proviene del hecho de que se realiza una convolución matemática usando dos funciones dadas para producir la salida (Comito, C. y Pizzuti, C., 2022).

Las redes neuronales generativas adversariales (GAN por sus siglas en inglés) están compuestas por la red generativa que genera soluciones candidatas y la red discriminativa que las evalúa (Comito, C. y Pizzuti, C., 2022).

De acuerdo con Islam, S. et al. (2020) las Redes Neuronales son las más utilizadas en la predicción de series temporales y serán utilizadas en este proyecto en la medida en que Power BI presente las características técnicas para soportarlo.

Análisis de restricciones

Para el desarrollo de la solución se usará una herramienta visualización que puede ser Power BI, Tableau, Cognos o Google Data Studio; de esta manera se le brindará dinamismo a los datos y las visualizaciones generadas.

Para el desarrollo de la solución también deberá generarse un código en algún lenguaje de programación que permita predecir futuros niveles de glucosa basados en los puntos de glucosa anteriores; este código deberá ser integrable con la herramienta de visualización seleccionada.

La herramienta seleccionada deberá ser publicada en un repositorio privado para ser presentado como parte integral de este trabajo.

Metodología para la selección y desarrollo de la solución

Se selecciona Power BI como herramienta de visualización ya que permite ser publicada en repositorios privados con acceso restringido a personas o grupos específicos. También tiene una funcionalidad para integrar código de lenguajes Python o R para generar gráficos diseñados por el usuario que tengan funciones avanzadas no disponibles en Power BI, como la predicción usando *machine learning*.

Con base en lo anterior, se selecciona Python como lenguaje de programación para el desarrollo de la predicción de los valores de glucosa futuro basando en los valores pasados de glucosa por paciente.

En cuanto a la metodología se seguirán los siguientes pasos para el desarrollo de la solución:

1. Diseñar la interfaz en Power BI
2. Desarrollo del código de Python para predecir valores futuros del nivel de glucosa
3. Integración de la interfaz en Power BI con el código de Python

Plan de implementación

Se obtuvieron los perfiles glucémicos de un estudio previo publicado por Mosquera-Ruiz, A & García-Jaramillo, M. (2022). Los datos corresponden a los perfiles glicémicos de un grupo de 30 pacientes en un horizonte de 1 año recogidos mediante la bomba de insulina Paradigm® Veo™ que y la bomba de insulina Minimed® 640G. En los archivos recopilados los datos de pacientes estaban anonimizados y se usaron las variables fecha, hora, nivel de glucosa en mg/dL y tipo de equipo usado en la medición.

Para el desarrollo se dividieron los pasos descritos anteriormente en dos etapas:

Etapas 1

Se utilizó Power BI Desktop para el diseño de visualizaciones que permitieran observar las variables recopiladas por paciente y otras añadidas en el procesamiento de los datos con Python, entre las características de diseño se incluyó:

- Una pestaña de entrada donde se puede seleccionar el paciente a analizar, que mediante botones permite filtrar la información de ese paciente.

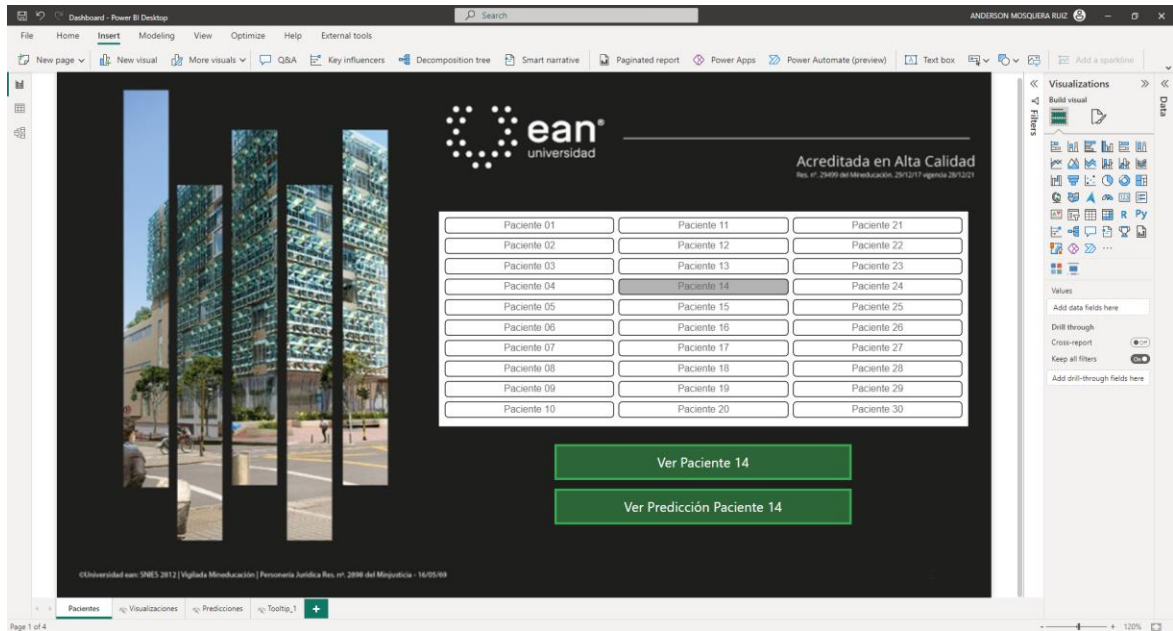


Figura 2. Pestaña de selección de pacientes del Dashboard de Power BI.

Fuente: El Autor.

- Una pestaña con la información completa del paciente que mediante un gráfico de serie de tiempo permite observar la media del nivel de glucosa por día.
- Un gráfico de herramientas adicionales (*Tooltips*) que permite observar cada uno de los puntos de mediciones en un día y detalles como el promedio del nivel de glucosa, máximo nivel de glucosa en el día, mínimo nivel de glucosa en el día, total de mediciones realizadas en el día.
- Un tabla que permite analizar el promedio del nivel de glucosa por horas y por rango medico realizadas en el periodo seleccionado.
- Un gráfico de dona que permite observar la distribución de la cantidad de mediciones realizadas por cada rango medico en el periodo seleccionado.
- Un gráfico de barras apiladas con la distribución por horas del periodo seleccionado que muestra en la leyenda el rango médico del nivel de glucosa.
- Filtro de tiempo que permite seleccionar uno o varios años, meses o días para filtrar los datos y que sean observados en todos los gráficos y herramientas adicionales (*Tooltips*) de la pestaña Paciente.

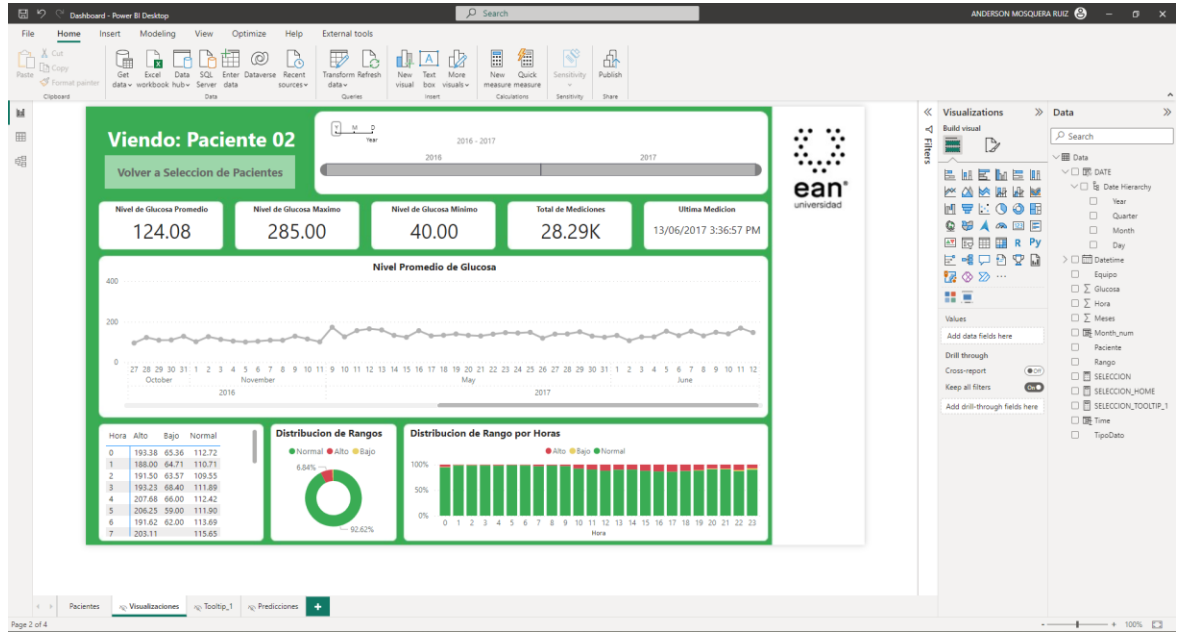


Figura 3. Pestaña de visualización del nivel de glucosa.

Fuente: El Autor.

- Una pestaña con una gráfica que permite visualizar los datos reales y las predicciones utilizando código de Python.

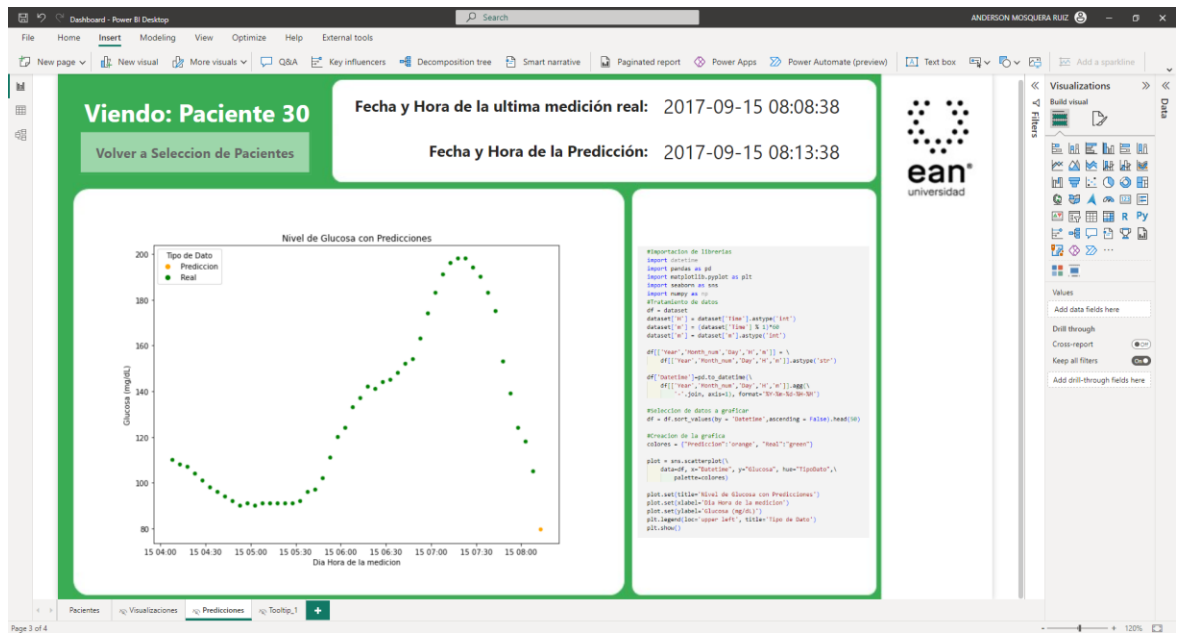


Figura 4. Pestaña de visualización de la predicción del nivel de glucosa.

Fuente: El Autor.

- Se usaron los colores institucionales de la universidad EAN y la primera diapositiva del formato de presentaciones de la EAN.

Es importante mencionar que el *dashboard* está publicado en la versión online de Power BI y puede ser consumida por cualquiera con el enlace, así mismo la visualización creada con Python solo puede ser consumida en la versión online si el *dashboard* es publicado por usuarios con licencia Pro de Power BI, para lo cual se inició un periodo de prueba de 60 días desde el 1 de junio de 2023.

El link de la solución en un sistema de visualización online se encuentra en el siguiente link: [Solución](#), de la misma manera el archivo original de Power BI se encuentra en los anexos y es denominado: Dashboard.pbix

Etapa 2

Se desarrolló un código de Python versión 3.10 usando las librerías numpy, pandas, tensorflow, keras, matplotlib, pickle, datetime, timedelta para importar y transformar los perfiles glucémicos de los pacientes con diabetes. Durante la importación y transformación de los datos se realizó la creación de un objeto tipo Dataframe de la librería pandas utilizando lógica iterativa y condicional para agregar columnas adicionales a las originalmente provistas. En este los niveles de glucosa nulos fueron reemplazados usando la interpolación lineal hacia adelante por paciente. Se eliminaron del Dataframe los puntos cuya distancia con respecto al anterior fuera mayor a 5 minutos para minimizar los errores en la predicción. Se creó la columna Rango (Rango medico) en la cual niveles de glucosa menores a 70 mg/dL fueron marcados como Bajo, niveles superiores a 180 mg/dL fueron marcados como Alto y los datos entre 70-180 mg/dL fueron marcados como Normal.

Python también se usó para la creación de una red neural recurrente de larga duración de termino corto (LSTM por sus siglas en inglés) con las siguientes características:

- 10 puntos de entrada al modelo para ser usados en la generación de la predicción
- La métrica de error es la media del error cuadrado de los datos predichos con respecto a los datos de entramiento
- 5 épocas de entrenamiento (es decir el modelo se reescribe 5 veces utilizando las iteraciones previas para reducir el error de predicción con la métrica de error definida)
- 1 capa de entrada, 2 capas escondidas y 1 capa de salida
- 64 neuronas de tipo LSTM en la primera capa escondida
- 8 neuronas con el método de activación ReLu (del inglés Rectified Linear Unit) en la segunda capa escondida
- 1 neurona con el método de activación lineal en la capa de salida
- 700,000 puntos para entramiento del modelo y 30,000 puntos para evaluación del modelo

```

def Red_Neural(window_size):
    #Esta toma solo los registros consecutivos tomados cada 5 minutos
    df = MasterDF
    df = df.sort_values(['Paciente', 'Datetime'], ascending = [True, True])
    #Vamos a crear la diferencia en minutos entre la fila actual y la anterior
    df['Delta_T_m'] = df['Datetime'].diff().apply(lambda x: x/np.timedelta64(1, 'm')).fillna(0).astype('int64')
    # df = df.sort_values(by='Delta_T_m', ascending = True)
    # df = df.iloc[-1 : :]
    # df = df.sort_values(by='Delta_T_m', ascending = False)
    # df = df.iloc[-1 : :]
    df = df[['Delta_T_m']]
    df = df.reset_index(drop=True)
    df = df.reset_index()
    df=MasterDF['Sensor_Glucose']

    #Esta parte crea lineas de 10 puntos de X, valores reales y Y valores a predecir por la red neuronal, necesita como parametro el window size
    df_as_np = df.to_numpy()
    X = []
    y = []

    for i in range(len(df_as_np)-window_size):
        row = []
        for a in df_as_np[i:i+window_size]:
            X.append(row)
            label = df_as_np[i+window_size]
            y.append(label)

    #Divide los datos en entrenamiento y pruebas
    X_train, y_train = X[:700000], y[:700000]
    X_val, y_val = X[700000:730000], y[700000:730000]

    #Define el tipo de red y los layers
    model = Sequential()
    model.add(LSTM(window_size, 64))
    model.add(Dense(8, 'relu'))
    model.add(Dense(1, 'linear'))
    #define parametros para la compilacion del modelo
    ra = ModelCheckpoint('model1/...save.h5', save_best_only=True)
    
```

Figura 5. Código de creación de la red neural usando Python.

Fuente: El Autor.

Con la red neuronal de estas características se realizaron predicciones para los 5 minutos posteriores a la última medición de cada paciente y se obtuvieron los resultados que pueden ser visualizados en la pestaña Predicciones del *Dashboard* en Power BI.

Se encontró que la red neuronal predice de una mejor manera el nivel de glucosa para el Paciente 30, por una razón probablemente asociada al momento de entrenamiento. También cabe mencionar que entrenar una red neuronal es costoso computacionalmente; la red neuronal diseñada en este proyecto es pequeña en términos de su cantidad de neuronas y tiene pocas épocas de entrenamiento, aun así, se toma más de dos horas para ser construida.

El código de limpieza y carga de los datos, la creación e implementación de la red neuronal y las visualizaciones usadas en Power BI se encuentran en los anexos en el cuaderno de jupyter llamado `Codigo.ipynb`

Conclusiones

Se realizó una revisión bibliográfica de la diabetes, su historia, sus tipos, las formas de tratamiento y las complicaciones que podrían existir de no ser bien tratada. También se incluyó en la revisión bibliográfica algunos métodos de predicción usando técnicas estadísticas y de *machine learning*. Finalmente se incluyó dentro de la revisión bibliográfica un apartado para describir a Power BI y las capacidades que puede generar cuando se combina con el lenguaje de programación Python. Esta revisión sirvió como insumo para las etapas de implementación del proyecto, desde el entendimiento de los datos disponibles hasta las fases de desarrollo del sistema de visualización y el código de predicción.

Se construyó un *dashboard* en Power BI el cual se publicó en un repositorio online para que pueda ser visualizado como parte integral de este trabajo. En este *dashboard* se incluyeron características calculadas a partir de la revisión bibliográfica y las necesidades propias del sistema de visualización, con el fin de ofrecer una experiencia de usuario que permite establecer inferencias y determinar rápidamente cuáles son los momentos problemáticos para cada paciente. Desde el *dashboard* también se pueden realizar filtros y obtener información adicional de periodos específicos de tiempo que el usuario quiera conocer.

Haciendo uso del lenguaje de programación Python se diseñó una red neuronal recurrente de tipo LSTM, que permite predecir el nivel de glucosa de los pacientes a los 5 minutos siguientes de la última medición. Para esto se tuvieron que hacer sucesivas transformaciones y consideraciones sobre los datos las cuales fueron realizadas utilizando librerías especializadas de Python. Los datos predichos se incorporaron con los datos originales y se cargaron al *dashboard* de Power BI. Estos pueden ser visualizados en una gráfica dentro el *dashboard* creada usando Python.

Referencias

- [1] Absent Data. (s.f.). Using Python Visuals in Power BI. <https://absentdata.com/how-to-user-python-and-power-bi/>
- [2] Bando, H.; Ebe, K.; Kato, Y.; Bando, M. y Yonei, Y. (2018). Investigation of blood glucose profile by continuous glucose monitoring (CGM). *Endocrinology Research and Metabolism*. Vol. 2 Mo. 1:10. 1-4.
- [3] Castillo, J. M. et al. (2019). Impacto del alto costo relacionado con la diabetes Mellitus en el sistema de salud en Colombia. *REDIIS / Revista De Investigación E Innovación En Salud*, 2, 82-91.
- [4] Chou, J. S. y Tran, D. S. (2018). Forecasting energy consumption time series using machine learning techniques based on usage patterns of residential householders. *Energy*. Volume 165. 709-726. DOI: 10.1016/j.energy.2018.09.144
- [5] Comito, C. y Pizzuti, C. (2022). Artificial intelligence for forecasting and diagnosing COVID-19 pandemic: A focused review. *Artificial Intelligence in Medicine*. Volume 128. DOI: 10.1016/j.artmed.2022.102286
- [6] Dods, R. F. (2013). *Understanding diabetes: A biochemical perspective*. John Wiley y Sons, Incorporated.
- [7] Dunning, T. (2012). *Diabetes education: Art, science and evidence*. John Wiley y Sons, Incorporated.
- [8] Fonseca, V. (2010). *Diabetes: Improving patient care*. Oxford University Press, Incorporated.
- [9] Gomez, A. M et al. (2019). Impact of sensor-augmented pump therapy with predictive low-glucose management on hypoglycemia and glycemic control in patients with type 1 diabetes mellitus: 1-year follow-up. DOI: <https://doi.org/10.1016/j.dsx.2019.07.024>
- [10] Holt, R. I. G.; Cockram, C.; Flyvbjerg, A. y Goldstein, B. J. (2017). *Textbook of diabetes*. John Wiley y Sons, Incorporated.
- [11] Holt, T. y Kumar, S. (2015). *Abc of diabetes*. John Wiley y Sons, Incorporated.
- [12] International Diabetes Federation (2021). *IDF Diabetes Atlas 10th edition*. https://diabetesatlas.org/idfawp/resource-files/2021/07/IDF_Atlas_10th_Edition_2021.pdf

- [13] Islam, S.; Hossain, E.; Rahman, A.; Hossain, M. S. y Andersson, K. (2020). A review on recent advancements in FOREX currency prediction. *Multidisciplinary Digital Publishing Institute (MDPI) – Algorithms*. 13. DOI: 10.3390/A13080186
- [14] Jain, P.; Joshi, A.; Mohanty, S. P. (2021). Everything you wanted to know about continuous glucose monitoring. pp. 1-51.
- [15] Krentz, A. (2004). *Emergencies in diabetes: Diagnosis, management and prevention*. John Wiley y Sons, Incorporated.
- [16] Lara-Benítez, P.; Carranza-García, M. y Riquelme, J. C. (2021). An Experimental Review on Deep Learning Architectures for Time Series Forecasting. *International Journal of Neural Systems*. Volume 31. DOI: 10.1142/S0129065721300011
- [17] Levy, D. (2016). *Type 1 diabetes*. Oxford University Press, Incorporated.
- [18] Microsoft. (2023a). *Create Power BI visuals with Python*. <https://learn.microsoft.com/en-us/power-bi/connect-data/desktop-python-visuals>
- [19] Microsoft. (2023b). Publish datasets and reports from Power BI Desktop. <https://learn.microsoft.com/en-us/power-bi/create-reports/desktop-upload-desktop-files>
- [20] Microsoft. (2023c). QuickStart: Connect to data in Power BI Desktop. <https://learn.microsoft.com/en-us/power-bi/connect-data/desktop-quickstart-connect-to-data>
- [21] Microsoft. (2023d). What is Power BI. <https://learn.microsoft.com/en-us/power-bi/fundamentals/power-bi-overview>
- [22] Mosquera-Ruiz, A y García-Jaramillo, M. (2022). Diseño de un prototipo para la generación de alarmas a partir del análisis de tendencias del perfil glucémico de un grupo de pacientes con diabetes. *ACOFI EIEI 2022*, pp. 1-11. <https://acofipapers.org/index.php/eiei/article/view/2246/1880>
- [23] Oh, S.; Byon, Y. J.; Jang, K. y Yeo, H. (2015). Short-term Travel-time Prediction on Highway: A Review of the Data-driven Approach. *Transport Reviews*. Volume 35. 4-32. DOI: 10.1080/01441647.2014.992496
- [24] Ramadevi, B. y Bingi, K. (2022). Chaotic Time Series Forecasting Approaches Using Machine Learning Techniques: A Review. *Multidisciplinary Digital Publishing Institute (MDPI) - Symmetry*. Volume 14. DOI: 10.3390/sym14050955
- [25] Saha, S.; Al-Rifai, R. H. y Saha, S. (2021). Diabetes prevalence and mortality in COVID-19 patients: a systematic review, meta-analysis, and meta-regression. *Journal of Diabetes Metabolic Disorders*. 2021 Mar 31; 20 (1):939-950. DOI: 10.1007/s40200-021-00779-2.
- [26] Shah, S. (2020). How to use Python Visuals in Power BI. <https://medium.com/analytics-vidhya/how-to-use-python-visuals-in-power-bi-c1ec92206781>

- [27] Sharma, E. (2018). Energy forecasting based on predictive data mining techniques in smart energy grids. *Energy Informatics*. 367-428. DOI: 10.1186/s42162-018-0048-9
- [28] Shaw, K. M. y Cummings, M. H. (2012). *Diabetes: Chronic complications*. John Wiley y Sons, Incorporated.
- [29] Srihari, A.; Mousumi, P. y Srivatsa, K. (2020). A Comparative Study and Analysis of Time Series Forecasting Techniques. *SN Computer Science*. 7. DOI: 10.1007/s42979-020-00180-5
- [30] Zaczyński, B. (2022). Microsoft Power BI and Python: Two Superpowers Combined. <https://realpython.com/power-bi-python/>