



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica  
Universitat Politècnica de València

Estudio de la dinámica de los niveles de glucosa  
de un paciente sano mediante el modelo de  
Dalla Man

Trabajo Fin de Grado

**Grado en Ingeniería Informática**

**Autor:** Jorge Ibáñez Felipe

**Tutor:** Antonio Hervás Jorge

**Cotutor:** Rafael J. Villanueva Micó

**2017/2018**

PÁGINA EN BLANCO

# Resumen

---

En un momento en que el grado de incidencia de la diabetes no hace sino aumentar año a año, el objeto de este trabajo es observar y analizar el modelado del comportamiento del organismo de una persona sana, de modo que podamos evaluar la utilidad del modelo metabólico de Dalla Mann et al. (que describe la asimilación y procesado de la glucosa en el cuerpo) sobre pacientes reales, esto es, que se pueda personalizar el modelo, con la esperanza de contribuir así a la prevención y tratamiento de esta enfermedad.

**Palabras clave:** Niveles de glucosa, modelo de Dalla Man, insulina, modelo computacional, programación Java

# Abstract

---

At a time when the diabetes incidence degree increases year to year, the purpose of this work is to observe and analyze the metabolism behavior model of a healthy person, so we can see the usefulness of the mathematical-physiological model of Dalla Man et al. (which describes the assimilation and processing of glucose in the human body) in real patients, that is, if it is possible to customize the model, with the aim of contributing to prevention and treatment of this disease.

**Keywords:** Glucose levels, Dalla Man model, insulin, computational model, Java programming.

PÁGINA EN BLANCO

# Índice de contenidos

---

Índice de figuras y tablas.....	7
Figuras.....	7
Tablas.....	8
1. Introducción.....	9
1.1. Contexto histórico.....	10
1.2. Motivación y alcance.....	11
2. Diabetes.....	13
2.1. Conceptos generales.....	14
2.1.1. Insulina.....	14
2.1.2. Glucosa.....	16
2.2. Tipos.....	17
2.3. Consecuencias y complicaciones.....	17
2.4. Costes de la enfermedad.....	19
2.5. Diagnóstico y tratamiento.....	20
3. Modelo interacción glucosa-insulina.....	22
3.1. Subsistema de glucosa.....	24
3.2. Subsistema de insulina.....	27
3.3. Subsistema de interacción glucosa – insulina.....	29
4. Implementación.....	32
4.1. Herramientas empleadas.....	32
4.1.1. Eclipse y Java 7.....	32
4.1.2. MATLAB.....	33
4.2. MatlabControl.....	34
4.3. OpenCSV.....	35
4.4. GlucoJ.....	35
4.5. Subsistemas de glucosa e insulina.....	36
4.6. FileManager.....	37
4.7. Ajuste – <i>Particle Swarm Optimization</i> (PSO).....	37
4.8. Validación – Matrices de error de Clarke y Parkes.....	38
4.9. Flujo de trabajo.....	39
4.10. Análisis de las salidas.....	43
5. Resultados obtenidos.....	45
6. Mejoras o ampliaciones.....	50



6.1.	Múltiples instancias de clientes y servidores.....	50
6.2.	Base de datos .....	51
6.3.	Compilar funciones MATLAB en Java.....	52
6.4.	Uso autónomo – app móvil .....	52
6.5.	Entradas auxiliares para el modelo (estrés, ejercicio, etc.) .....	53
7.	Conclusiones .....	54
8.	Bibliografía.....	55

# Índice de figuras y tablas

---

## Figuras

Figura 1 – Distribución de la prevalencia de diabetes por raza. [25] .....	9
Figura 2 – Papiro Ebers, columna 37, sobre desórdenes gastrointestinales. [26].....	10
Figura 3 – Código QR de [8]. Infografía “La Diabetes en España” .....	12
Figura 4 – Función de la insulina en el organismo. [9] .....	14
Figura 5 – Curvas descritas por cada tipo de insulina. [28] .....	15
Figura 6 – Principales síntomas de la diabetes mellitus, asociados a los órganos afectados. [27] .....	18
Figura 7 – Costes directos relacionados con la diabetes mellitus y sus complicaciones (% del total). AGMC: automonitorización de glucemia capilar; AP: atención primaria; DM: diabetes mellitus; IRC: insuficiencia renal crónica. [12] .....	20
Figura 8 – Esquema del sistema de control de glucosa – insulina. [14].....	22
Figura 9 – Modelo del metabolismo de la glucosa. Elaboración propia a partir de [7] .....	23
Figura 10 – Detalle del modelo fisiológico que describe la aparición de glucosa en plasma sanguíneo. Elaboración propia a partir de [7] .....	24
Figura 11 – Esquema de compartimentos del modelo para el subsistema de glucosa. Elaboración propia a partir de [7].....	25
Figura 12 – Modelo de farmacocinética de la insulina. Elaboración propia a partir de [7] .....	27
Figura 13 – Fragmento del fichero CSV empleado en nuestros experimentos. Elaboración propia. .....	40
Figura 14 – Esquema general de funcionamiento del sistema. En azul contenido y/o datos de entrada, en amarillo unidades funcionales, en gris flujo de información. La parte de la simulación en MATLAB queda contenida en el bloque funcional GlucoJ. Elaboración propia.....	41
Figura 15 – Detalle de funcionamiento perspectiva GlucoJ – MATLAB. Elaboración propia. .	42
Figura 16 – Ejemplo de aumento de niveles glucosa en paciente tras las respectivas ingestas de carbohidratos. Elaboración propia.....	43
Figura 17 – Ejemplo de aumento de niveles de insulina en plasma sanguíneo tras la infusión subcutánea de insulina. Elaboración propia. ....	44
Figura 18 – Ingesta de carbohidratos distribuida en el tiempo. Elaboración propia a partir de los datos disponibles de un paciente sano real. ....	45
Figura 19 – Salidas obtenidas por los subsistemas del modelo tras procesar los datos del paciente real. Elaboración propia. ....	46
Figura 20 – Comparación entre la glucosa obtenida por el modelo y la medida en el paciente real. Elaboración propia. ....	46
Figura 21 – Diagrama de error de Clarke (“Clarke Error Grid”). Elaboración propia.....	47
Figura 22 – Diagrama de error de Parkes (“Parkes Error Grid”). Elaboración propia.....	48
Figura 23 – Predicción de niveles de glucosa para las siguientes cuatro horas. Elaboración propia. .....	49

## Tablas

Tabla 1 – Tipos de insulina y variaciones temporales en inicio, pico y duración del efecto. [10] .....	15
Tabla 2 – Valores típicos para la población promedio de los parámetros del modelo del subsistema de glucosa. [13] [14] .....	26
Tabla 3 – Valores típicos para población promedio de los parámetros del modelo de subsistema de insulina. [14].....	29



# 1. Introducción

---

La diabetes mellitus es una pandemia global que afecta a casi cuatrocientos millones de personas en todo el mundo, con una incidencia ligeramente mayor en la región del Pacífico asiático (APAC), siendo causa directa o indirecta del fallecimiento de alrededor de cuatro millones de personas al año en todo el mundo. [1]

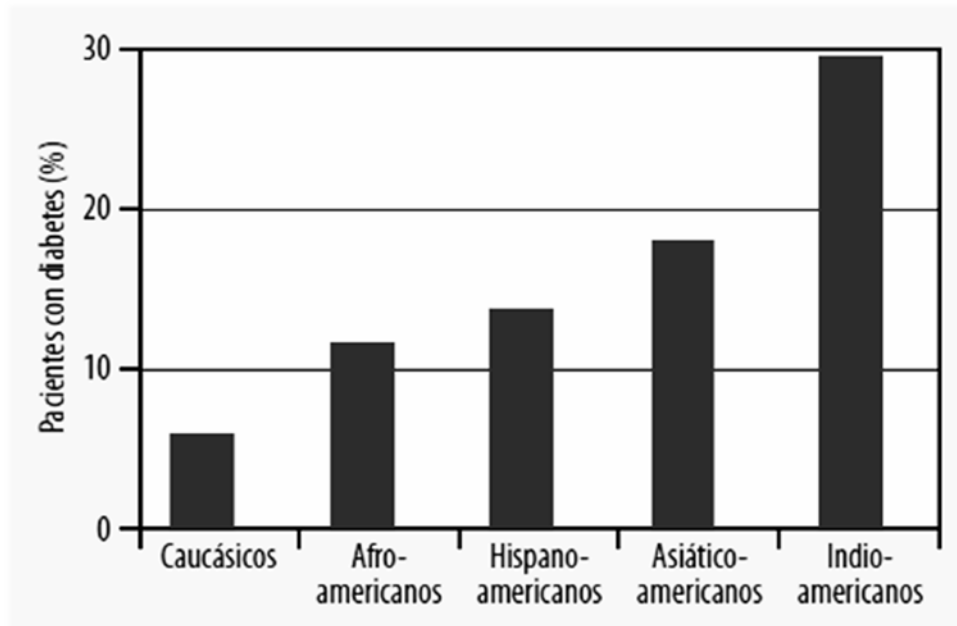


Figura 1 – Distribución de la prevalencia de diabetes por raza. [25]

Tomando como ejemplo España y según el estudio “di@bet.es” [2], realizado por la necesidad de actualizar los datos referentes a la prevalencia de la enfermedad, casi el catorce por ciento de los mayores de dieciocho años tiene diabetes tipo 2. De ellos, casi la mitad (el cuarenta y tres por ciento) desconocían que la padecían.

Esta enfermedad perjudica gravemente el normal funcionamiento del cuerpo, dañando diferentes órganos y tejidos tan importantes como el corazón, la vista, los riñones o el sistema nervioso. No es exagerado decir que la diabetes es una de las principales causas de ceguera, amputaciones y enfermedades renales en las sociedades desarrolladas.

Su principal síntoma es la incapacidad de regular la concentración de glucosa en el flujo sanguíneo, ya sea por un mal aprovechamiento de la insulina o la nula producción de la misma.

Aunque el objetivo de este trabajo es la implementación y análisis de un modelo metabólico que sea capaz de predecir los flujos de los indicadores relevantes, previamente analizaremos varias perspectivas relevantes, partiendo desde un breve resumen de la progresión en el conocimiento sobre la afección, tras lo cual profundizaremos en la época actual.

A pesar de la antigüedad de los primeros registros históricos sobre la afección, es recientemente cuando se han venido realizando los avances más significativos en controlar y paliar los problemas derivados de la misma, buscando minimizar el impacto que supone en la vida diaria del paciente.

### 1.1. Contexto histórico

Las primeras referencias a síntomas relacionados con la diabetes que podemos hallar está fechada más de diez siglos antes del inicio de la era cristiana, en el papiro Ebers, encontrado en la actual Luxor y conservado hoy día en la Biblioteca Universitaria de Leipzig. [3] [4]

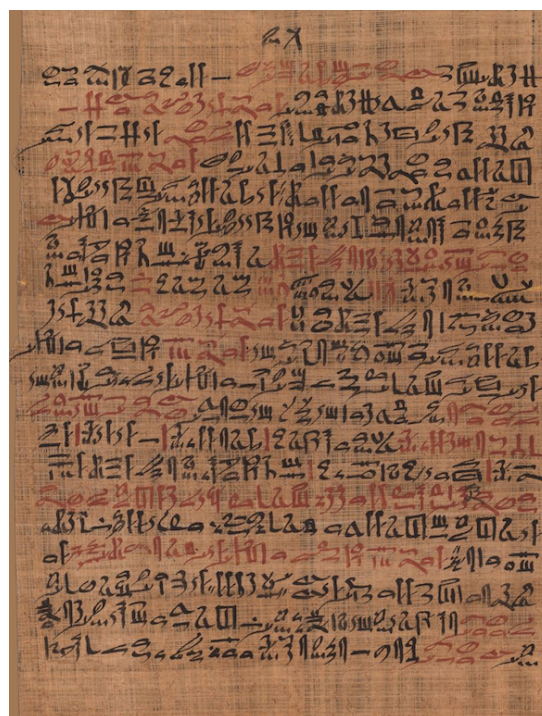


Figura 2 – Papiro Ebers, columna 37, sobre desórdenes gastrointestinales. [26]

En sus más de veinte metros de longitud se narra, entre diferentes temas médicos, la evolución y sintomatología de enfermos que pierden peso, miccionan muchas veces al día, y tienen un hambre y sed constantes.

Casi mil años más tarde, en el continente asiático, se asocian también la obesidad, la vida acomodada y la frecuencia de comidas dulces como factores característicos de estos pacientes, de los cuales se describe una orina pegajosa y dulce que atrae a los insectos. [4]

Si bien más adelante profundizaremos en la vertiente médica de la enfermedad sobre la que orbita este trabajo, resulta notable que a pesar de existir numerosos registros medievales, como el Canon de Avicena, sobre los síntomas y tratamientos, no es hasta bien entrado el siglo XVI, cuando se hacen avances significativos en la observación y tratamiento de la misma, apreciando la similitud

entre los síntomas documentados hasta la fecha y los que mostraban algunos animales de laboratorio al extirparles el páncreas.

Con la llegada del microscopio se mejoró considerablemente el conocimiento sobre la estructura de los tejidos. Ya en el siglo XIX, gracias a las sucesivas mejoras en la óptica de estos aparatos, Paul Langerhans descubrió durante su tesis doctoral los grupúsculos celulares pancreáticos que llevan su nombre (“islotos de Langerhans”), y que más tarde revelaron su principal función: la producción de insulina.

Durante los primeros compases del s. XX, en 1923 se otorga el premio Nobel de Fisiología y Medicina a F. G. Banting y J. J. R. Macleod [5] por el descubrimiento y aislamiento de la insulina, hormona anabólica secretada por el páncreas que regula el metabolismo de la glucosa en el cuerpo. Pocos años más tarde, F. Sanger [6] es galardonado con el Nobel de Química por revelar la estructura de la molécula de insulina bovina, contribuyendo notablemente a la posterior obtención artificial de la misma.

Dejando atrás la experimentación con perros original, en los experimentos de Banting y Macleod se observó que, al inocular ciertos extractos refinados del páncreas a pacientes humanos, disminuía la concentración de glucosa en sangre a niveles muy bajos (no necesariamente normales) y los síntomas de la diabetes remitían.

Tras el hallazgo se emprende y extiende el uso de la insulina como tratamiento para los diabéticos, cambiando ya en tiempos relativamente recientes su origen porcino por el sintético, siendo fabricada a partir de bacterias modificadas con técnicas de ingeniería genética para que produzcan insulina humana.

Podemos considerar este último avance como el principio de la época actual, en la que soluciones como las bombas de insulina y los páncreas artificiales (similares a bombas de insulina autorreguladas) buscan ser el remedio para dejar atrás los perjuicios de la enfermedad.

## 1.2. Motivación y alcance

En los últimos años la modelización de los niveles de glucosa se ha convertido en un campo muy prolífico, persiguiendo el objetivo de explicar y predecir la concentración en plasma sanguíneo de la misma. Disponer de un modelo válido para explicar el metabolismo de la glucosa permitiría prevenir situaciones de riesgo en personas con resistencia a la insulina.

Este estudio se basa en el modelo desarrollado por Dalla Man, Rizza y Cobelli [7], que explicaremos posteriormente. Este sistema está conformado por varios subsistemas conectados, dependiendo cada uno de ellos de los demás en el tiempo.

En el trabajo original se obtienen valores de los parámetros significativos mediante pacientes “in silico”, pero nuestro objetivo es comprobar que este modelo es también aplicable a pacientes reales. De esta forma, lograremos valores de los parámetros personalizados para cada caso, y podremos utilizar este modelo en clínica diaria.

En este trabajo aplicaremos el modelo a datos de un paciente sano, desarrollando una aplicación Java que implemente el comportamiento del metabolismo del sujeto, y comprobaremos que el modelo explica las variaciones de los niveles de glucosa del organismo, lo que nos debe permitir hacer una predicción fiable a cuatro horas vista.

Debemos subrayar que el trabajo se aplica a un paciente sano, pero si se probara su validez aplicando la técnica propuesta a varios pacientes, se plantea el posible uso con afectados por diabetes tipo 1, que necesitan un aporte total de insulina de fuentes externas.

Hoy en día la calidad de vida de un diabético ha mejorado considerablemente, siendo similar a la de cualquier persona sin deficiencias en el metabolismo de la glucosa.

Sin embargo, su tratamiento y control sigue sujeto a herramientas (plumas, bombas, etc.) y un estricto control diario, lo que supone una cota de dependencia que por el momento ha sido muy complicada de resolver, por lo que dista todavía de ser una enfermedad que podamos considerar como superada.

Además del perjuicio para el paciente, si tenemos en cuenta los costes directos (tratamiento y posibles hospitalizaciones) e indirectos (absentismo laboral, jubilación anticipada, etc.) de la afección, que superan ampliamente los diez mil millones de euros anuales en nuestro país [8], nos encontramos ante una razón más para espolear la investigación y la búsqueda de nuevas soluciones.

La labor realizada parte con la esperanza a largo plazo de que la informática contribuya a la mejora de la vida diaria de los pacientes de diabetes y el alivio del estrés originado por un continuo autocontrol, contrastando el potencial presente en la aplicación de nuestra disciplina a otros campos, en este caso la medicina o biología humana.

En un horizonte más próximo, dar a conocer los avances más recientes en parametrización y simulación del metabolismo humano, así como el proceso de implementación y el funcionamiento general del entrenamiento y comportamiento de un modelo predictivo, junto a la infraestructura de datos generada para mantener el bucle que realimenta esta función, y el papel de MATLAB como herramienta en la que realizar parte de las simulaciones pertinentes.



Figura 3 – Código QR de [8].  
Infografía “La Diabetes en España”

## 2. Diabetes

---

Como ya se ha comentado, la diabetes es un trastorno metabólico-endocrino crónico que se presenta cuando el organismo no secreta suficiente insulina o es incapaz de usarla eficazmente, con lo que los niveles de glucosa en sangre se vuelven muy elevados de forma persistente.

Los síntomas más característicos son conocidos como las “cuatro P”:

- Polifagia o incremento del hambre.
- Polidipsia o aumento de la sed.
- Poliuria, exceso en la producción de orina.
- Pérdida de peso sin explicación aparente.

Estos suelen formar el cuadro básico que debe alertar de la posibilidad de padecer diabetes.

Por otro lado, la Sociedad Española de Diabetes señala los siguientes factores de riesgo para el tipo dos de la enfermedad a partir del estudio “di@bet.es” [2]:

- La obesidad y la hipertensión arterial duplican la probabilidad de padecerla.
- El nivel de estudios resulta determinante, un bajo nivel de estudios duplica el riesgo tanto de obesidad, como de hipertensión y diabetes.
- Las personas que no realizan ejercicio físico en su tiempo libre tienen aproximadamente un cincuenta por ciento más de posibilidades de ser obesas, y las personas con diabetes hacen la mitad de ejercicio que las personas delgadas o sin diabetes.
- Hay una presencia significativa de personas que padecen diabetes y no lo saben (alrededor de un cuatro por ciento de la población).

Durante este capítulo tendremos ocasión de profundizar en la propia dolencia, introduciendo los conceptos generales necesarios para la comprensión general del trabajo y acotando la sintomatología, características principales y efectos en la salud de cada una de las dos variantes de la afección.

Como curiosidad, el estudio confirma los efectos positivos sobre la salud del aceite de oliva. Frente a las personas que consumen aceite de girasol o de semillas, las personas que toman aceite de oliva tienen niveles significativamente más elevados de HDLc (colesterol bueno), aproximadamente dos miligramos por decilitro.

## 2.1. Conceptos generales

### 2.1.1. Insulina

Empezaremos ampliando lo dicho sobre la insulina, que recordemos es una hormona secretada por las células beta del páncreas que juega un papel fundamental en la metabolización de la glucosa y su transformación en energía.

En concreto, se libera ante ciertos estímulos, como la ingestión de proteínas y carbohidratos, y se encarga de que las células del organismo puedan disponer de la glucosa (una gran parte de los carbohidratos que comemos dan lugar a glúcidos al ser procesados) necesaria para realizar sus funciones, transformándola mediante un proceso llamado glucólisis en “trifosfato de adenosina” o ATP, la fuente de energía para la mayoría de trabajos que llevan a cabo.

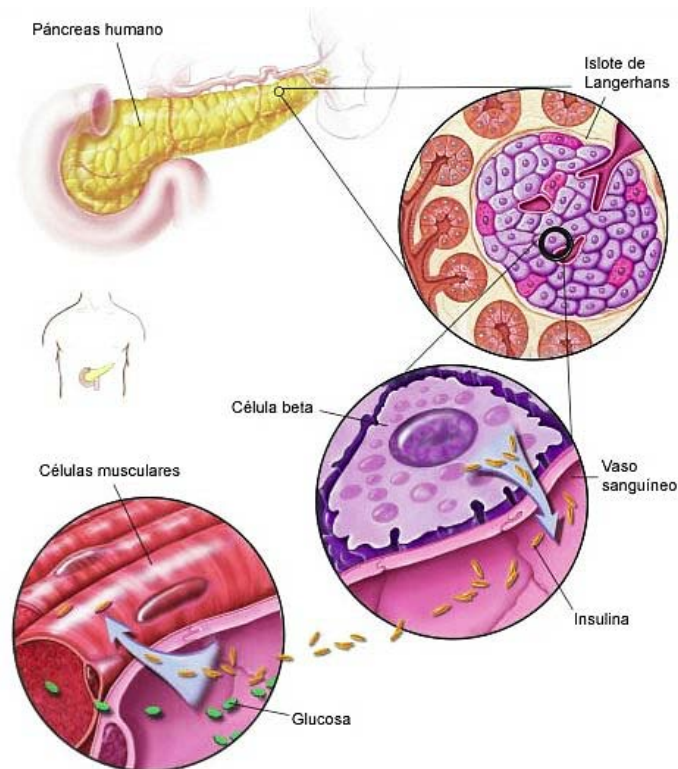


Figura 4 – Función de la insulina en el organismo. [27]

Cabe señalar que la liberación de la insulina al torrente sanguíneo no es un proceso homogéneo, sino que se produce de diferente manera en función del estímulo que provoca su secreción. Más concretamente, hablaremos de una acción “rápida”, impulsada por una ingesta de carbohidratos y la presencia de glucosa en sangre que de ésta se deriva, y otra de acción “lenta”, no tan aguda y prolongada en el tiempo.

Veremos cómo esta diferencia se aprecia en el modelo que emplearemos más adelante, para el que utilizaremos en todo momento insulina de acción rápida, que es la inyectada de forma subcutánea.

Aunque el anterior es el efecto más importante, la insulina también interviene directamente en otros factores del funcionamiento general del metabolismo, principalmente poniendo en marcha otras reacciones llamadas glucogénesis (almacenamiento de glucosa en el hígado como reserva energética, en una forma llamada glucógeno) y glucogenólisis, el proceso contrario al anterior.

Además, también aumenta la retención de sodio por parte del sistema renal y estimula la síntesis de proteínas, ácidos grasos y triglicéridos.

De todo lo mencionado se observa la importancia de la hormona y se sientan las bases para las complicaciones y problemas que derivan de la enfermedad, que explicaremos más adelante.

Siendo el tratamiento principal para la afección, cabe resaltar que existen diferentes tipos de insulina. Además del posible origen, se distinguen por:

- Rapidez con que se absorben por el organismo.
- Tiempo que tardan en desaparecer totalmente.
- Tiempo que tardan en alcanzar el pico de máxima concentración.

Tipo de insulina	Inicio	Pico	Duración
<b>Acción ultrarrápida (lispro, aspart)</b>	10 a 15 minutos	30 minutos a 2 horas	2 a 4 horas
<b>Acción rápida (humana)</b>	25 a 40 minutos	2 a 4 horas	5 a 8 horas
<b>Acción intermedia (NPH)</b>	2 a 4 horas	4 a 8 horas	10 a 16 horas
<b>Acción lenta (glargina, detemir)</b>	1 a 4 horas según concentración	Sin pico, constante.	16 a 24 horas, según concentración
<b>Mixta</b> (mezcla de diferentes insulinas)	Depende de la proporción de las diferentes insulinas en la mezcla.		10 a 16 horas

Tabla 1 – Tipos de insulina y variaciones temporales en inicio, pico y duración del efecto. [9]

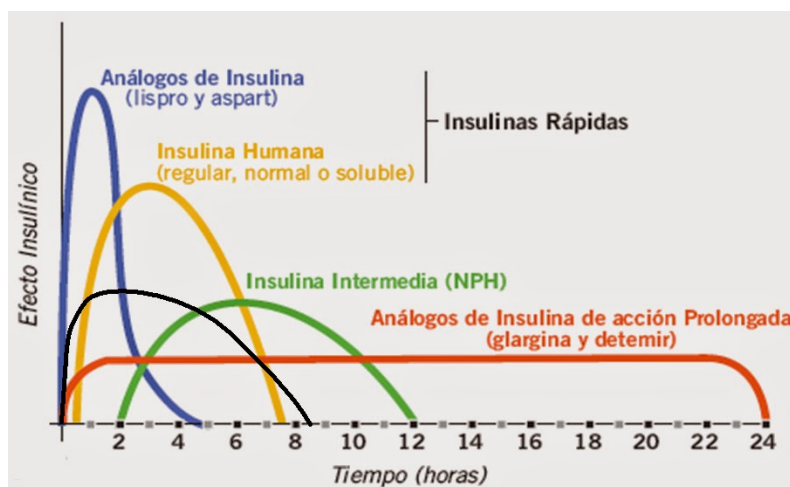


Figura 5 – Curvas descritas por cada tipo de insulina. [29]

### 2.1.2. Glucosa

A continuación, hablaremos sobre la glucosa, un azúcar simple o monosacárido (no se descompone en otros más simples) que representa el compuesto orgánico más común en la naturaleza.

Esta molécula representa el principal combustible de las células, que la emplean para obtener la energía que necesitan, y el principal constituyente de los elementos de almacenamiento energético que emplean los seres vivos, como el almidón (en plantas) o el glucógeno, que hemos mencionado en el apartado anterior.

Como se ha comentado, para su asimilación y metabolización necesita de la insulina, que se encarga de ayudar a que los tejidos y órganos del cuerpo puedan absorberla y utilizarla.

Durante el proceso de digestión de los alimentos ingeridos, el cuerpo descompone los carbohidratos en glucosa, que se absorbe directamente hacia el torrente sanguíneo, facilitando la disponibilidad inmediata y su correspondiente aprovechamiento a lo largo del organismo.

Es por esto que resulta equivalente hablar de “índice glucémico” o “nivel de azúcar en sangre”. Así mismo, hablaremos de “hipoglucemia” para referirnos a niveles anormalmente bajos de concentración en sangre, o “hiperglucemia” en caso de niveles demasiado altos.

También resulta interesante hablar sobre la fructosa [10], otro monosacárido natural, con mucha presencia en las frutas y verduras. No requiere de insulina para ser procesada, por lo que resulta una opción interesante para los pacientes de diabetes desde la perspectiva del control glucémico.

Sin embargo, hay que tener en cuenta que no estimula la secreción de insulina ni de grelina (hormona que provoca sensación de saciedad), por lo que no se puede sustituir totalmente la glucosa por fructosa sin padecer efectos secundarios, como el aumento de colesterol en sangre, la obesidad o la resistencia a la insulina, lo que terminaría redundando en un mayor riesgo o gravedad de diabetes tipo dos.

Además, la fructosa se procesa únicamente en el hígado, a diferencia de la glucosa que es procesada por cualquier órgano o tejido sobre el que pueda actuar la insulina. Este hecho podría tener, a largo plazo, consecuencias tóxicas para el tejido hepático, aunque no está determinado de forma concluyente.



## 2.2. Tipos

Existen dos tipos diferenciados de diabetes, en función de la problemática que enfrenta el cuerpo relacionada con la diabetes. Obviamos intencionadamente en este análisis la diabetes gestacional, que ocurre en el feto cuando las hormonas del embarazo bloquean la acción de la insulina. Muchas veces, ésta última desaparece con el parto.

En primer lugar, la diabetes llamada de “tipo uno” o insulino dependiente es la que se origina como un proceso en el cual el sistema inmunitario ataca a las células que se encargan de la generación de insulina, los ya mencionados “islotes de Langerhans”, sitios en el páncreas.

La irreversible destrucción de las mismas conduce irrevocablemente a la disminución en la producción de la hormona, lo que se traduce en deficiencia grave de la misma.

De este modo, la glucosa se acumula en el torrente sanguíneo sin remedio, provocando los efectos que venimos comentando (y que ampliaremos en el siguiente apartado) además de obligar al paciente a inyectarse insulina y monitorizar los niveles de la misma durante todos los días de su vida.

En segundo lugar, la diabetes de “tipo dos” o no insulino dependiente, que es la más común, tiene lugar cuando el cuerpo no es capaz de producir suficiente insulina, o en su defecto las células generan “resistencia” a la hormona, impidiendo que realice su función de aportación de glucosa a los tejidos de forma eficaz.

Mientras que la primera suele contraerse durante la juventud o infancia y habitualmente se desarrolla totalmente en un periodo corto y tiene una incidencia menor (entre el diez y el quince por ciento de los enfermos), el segundo tipo afecta de forma más gradual, disminuye la secreción de insulina y va aumentando el nivel de producción de glucosa y la resistencia a la misma de un modo paulatino.

## 2.3. Consecuencias y complicaciones

Junto a los síntomas identificativos que ya hemos mencionado (las “cuatro P”), existe toda una plétora de perjuicios a la salud. Algunos son producto de la hiperglucemia que suele acompañar a la diabetes, en tanto en cuanto la presencia de altos niveles de glucosa de forma persistente resulta tóxica para los tejidos y órganos con los que entra en contacto.

En función de su gravedad y duración, los podemos agrupar en crónicos y agudos.

De entre los primeros, destacan los problemas relacionados con la circulación, siendo de frecuente aparición el menoscabo del sistema nervioso periférico, del cerebro y cardíaco. Huelga decir que las consecuencias últimas de estos daños críticos tienen efectos devastadores.

Algunos de ellos son la ceguera originada por el deterioro de los vasos sanguíneos que llegan a la retina (retinopatía diabética), el pie diabético (que puede conllevar amputación del miembro gangrenado) originado por la insensibilización en los nervios de las extremidades, en

combinación con la mayor dificultad para cicatrizar las heridas, el accidente cardiovascular (infarto de miocardio, angina de pecho) o la enfermedad renal crónica.

También son significativos el deterioro del corazón y la hipertensión arterial, la impotencia en los hombres y la desaparición de la menstruación en mujeres, náuseas, vómitos y dolor abdominal, falta de energía o debilidad (letargo ocasional), convulsiones y el coma diabético.



Figura 6 – Principales síntomas de la diabetes mellitus, asociados a los órganos afectados. [28]

Por otro lado, las posibles complicaciones agudas que se pueden presentar son la “Cetoacidosis Diabética” (CAD), en caso de diabetes tipo 1, y el “Estado Hiper glucémico Hiperosmolar” o “Coma Hiperosmolar” (CH), que puede presentarse en ancianos con diabetes de tipo 2.

En el caso de la CAD, debido a un déficit de insulina, el cuerpo se muestra incapaz de utilizar la glucosa como fuente de energía, por lo que recurre a la grasa como solución. El proceso químico mediante la cual se descomponen los lípidos produce unos compuestos ácidos, tóxicos en concentraciones altas, llamados “cetonas”, que se acumulan en la sangre y en la orina.

Para compensar esta situación, el organismo libera glucosa desde el hígado. Al seguir faltando insulina, las células y tejidos no pueden absorberla, lo que redundará en un incremento desmesurado de la glucemia, lo que a su vez realimenta la reacción anterior.

La duración del proceso es de pocas horas, y pone en peligro la vida del paciente, que debe hidratarse y disminuir sus niveles glucémicos urgentemente. Es una de las primeras señales de alarma para enfermos de diabetes tipo 1 que no saben que la padecen.

Por otra parte, el CH se origina también por una elevada proporción de glucosa en sangre, que puede tener lugar cuando el cuerpo tiene suficiente insulina para evitar la cetosis, pero no la suficiente como para compensar los altos niveles glucémicos.

El sistema renal trata de eliminar el exceso de glucosa mediante la orina, pero si el paciente no está suficientemente hidratado, este proceso no tiene lugar o al menos no es eficiente, por lo que el índice glucémico se eleva.

Esto induce una situación de diuresis osmótica (aumenta la micción por el incremento de sodio y potasio en el líquido filtrado por los riñones), que lleva a una pérdida de agua y sales minerales de diferentes partes del cuerpo, incluyendo el cerebro.

De este modo, se genera un círculo vicioso de deshidratación celular y diuresis osmótica que sólo se puede corregir con un aporte correcto de líquidos.

Ambas son patologías agudas de mucha gravedad, al presentarse deshidratación grave, convulsiones, insuficiencia renal, disminución del flujo sanguíneo y riesgo de coma y fallo multiorgánico, lo que en la mayoría de los casos acarrea el fallecimiento.

## 2.4. Costes de la enfermedad

Hemos citado como parte de la motivación de este trabajo los costes económicos para el sistema sanitario, que no es sino otra manera de definir el impacto social de la diabetes. [8]

Dada la gravedad de la situación actual, en que la cantidad de afectados se incrementa a priori de forma irreversible en los países del primer mundo y en desarrollo, se realizan periódicamente estudios para evaluar el peso creciente que la enfermedad abarca en el presupuesto del Sistema Nacional de Salud (SNS).

En nuestro caso tomaremos como referencia los resultados del estudio elaborado por CIBERDEM (Centro de Investigación Biomédica en Red de Diabetes y Enfermedades Metabólicas Asociadas) y CABIMER (Centro Andaluz de Biología Molecular y Medicina Regenerativa) en el año 2013, de los más recientes, publicado en la revista “Diabetología” de la Asociación Europea para el Estudio de la Diabetes. [11]

Sin ánimo de extendernos más allá del propósito explícito de este punto, obviaremos las metodologías empleadas y pasaremos directamente a los resultados, que revisaremos sucintamente para reforzar la motivación de este trabajo.

En ellos se detalla el astronómico coste que supone la diabetes para el sistema sanitario español, que ascendió a los 5809 millones de euros en el año 2013, de los cuales la parte concerniente a las complicaciones fue de 2143 millones de euros.

Esto supone más de un 8% del total presupuestario, y es un cálculo realizado sin tener en cuenta el elevado infradiagnóstico de la DM. Según el estudio, cuantificar el gasto asumiendo que fueran tratados todos los enfermos que, se estima, la padecen elevaría la cifra hasta los 8787 millones de euros, el 12,5% del presupuesto sanitario.

En el siguiente gráfico podemos encontrar el detalle de los costes según su origen:

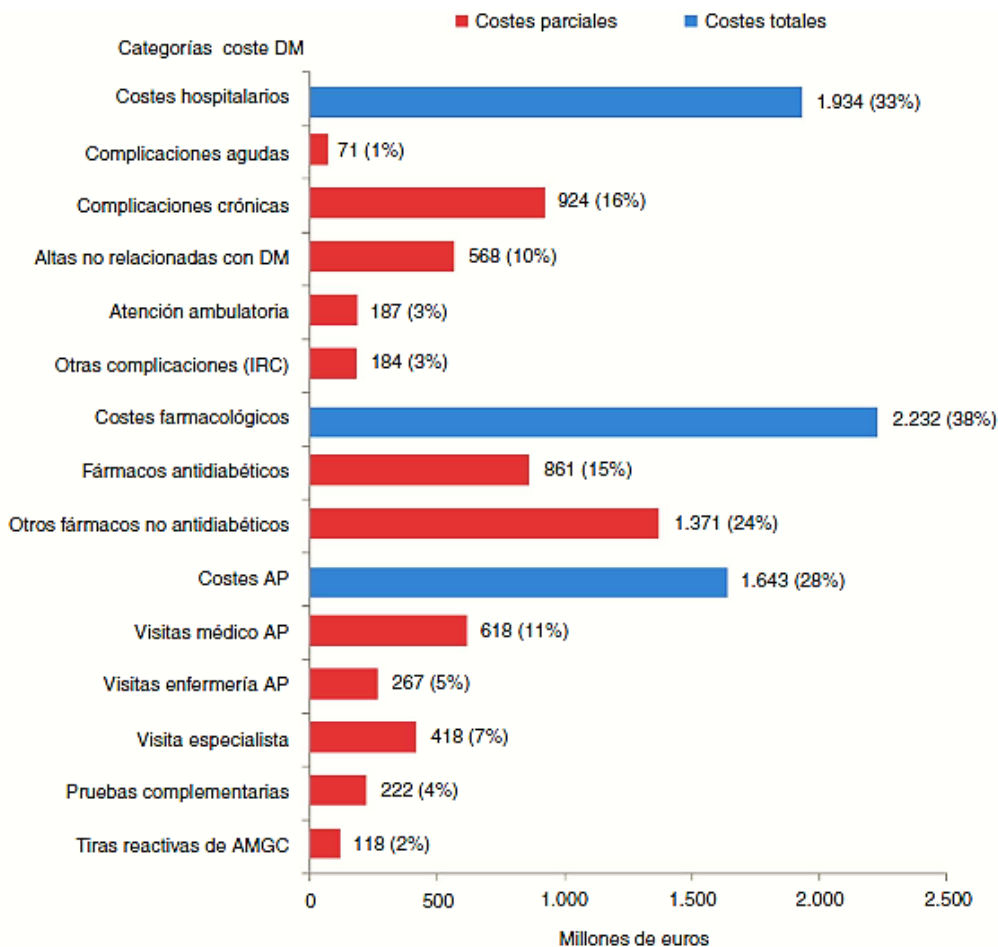


Figura 7 – Costes directos relacionados con la diabetes mellitus y sus complicaciones (% del total). AGMC: automonitorización de glucemia capilar; AP: atención primaria; DM: diabetes mellitus; IRC: insuficiencia renal crónica. [11]

## 2.5. Diagnóstico y tratamiento

Según establece la Organización Mundial de la Salud, el procedimiento de diagnóstico de la diabetes se basa en la detección de niveles anormalmente altos de glucosa en sangre, además de los síntomas característicos mencionados anteriormente: polifagia, poliuria y polidipsia, aparejados a pérdida de peso.

Como nota orientativa, el nivel considerado “normal” de glucosa basal en plasma sanguíneo es de alrededor de noventa mg/dl (oscila entre ochenta y ciento veinte mg/dl). Dos horas después de comer, la medición debe arrojar una concentración menor de entre ciento cuarenta y ciento sesenta mg/dl.

Más precisamente, se consideran niveles glucémicos anómalos más de doscientos miligramos por decilitro (mg/dl) en una medición aleatoria, o pasadas dos horas tras la ingesta de setenta y cinco gramos de azúcar en trescientos setenta y cinco mililitros de agua.

Si la prueba se realiza en ayunas, la alerta se fija en ciento veintiséis miligramos por decilitro.

En cualquiera de los tipos de diabetes, el tratamiento se centra en la recuperación de niveles normales de glucosa en plasma sanguíneo, aunque existen diferencias a la hora de enfrentar una variante u otra de la enfermedad.

Respecto al primer tipo, trasplantes de páncreas serían un posible remedio, pero no es una solución idónea por la escasez de donantes.

Por tanto, habitualmente la única opción viable es la administración de insulina, ya sea a través de inyecciones, el uso de una bomba de insulina o similares, acompañado de un estricto control de la dieta y ejercicio regular.

Además, es conveniente, cuando no necesaria, la educación del paciente para que pueda llevar el control constante de su glucemia, conocer las dietas y comportamientos a seguir, etc.

Con el segundo tipo, dependiendo de la patología concreta y su gravedad, será suficiente con mantener hábitos de vida saludables (dieta controlada, ejercicio regular), o habrá que administrar también medicación antidiabética, en posible combinación con insulina.

### 3. Modelo interacción glucosa-insulina

A partir de este punto, habiendo sentado las bases teóricas sobre las que se vertebra el trabajo, presentaremos el modelo de interacción metabólica para la glucosa planteado por Dalla Man, Rizza y Cobelli [7].

Éste consta de un conjunto de ecuaciones diferenciales que describen la variación de distintos procesos orgánicos, que discretizaremos y convertiremos en ecuaciones en diferencias, lo que facilitará los posteriores esfuerzos de programación y computación.

Deberemos calibrar el modelo para determinar los valores de los parámetros que reflejen lo más fielmente posible los niveles o evolución de glucosa de un paciente sano del que poseemos una medición continua de sus niveles de glucosa en un periodo de tiempo. Como punto de partida para estos parámetros se obtienen valores de pacientes *in silico*, es decir, un intervalo de confianza dentro del cual se parte de un valor medio, que podremos adaptar a cada paciente.

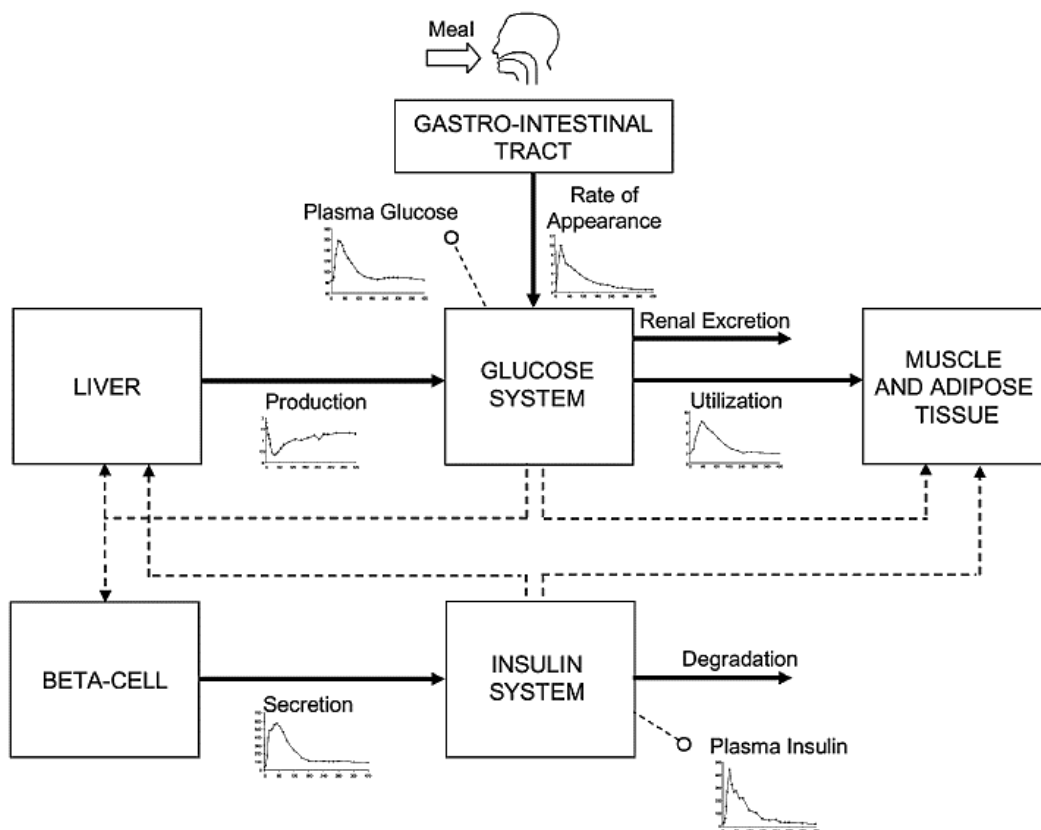


Figura 8 – Esquema del sistema de control de glucosa – insulina. [13]

En este orden de cosas, introduciremos en primer lugar el diseño general, para transitar posteriormente hacia el detalle de sus componentes internos. Lo que se explica aquí es una simplificación del modelo descrito en [7]. En este trabajo, no obstante, aparecen muchos más subsistemas.

Como se ilustra en la siguiente figura, el funcionamiento del metabolismo de la glucosa se puede esquematizar como un sistema con dos o más entradas, y una salida resultante.

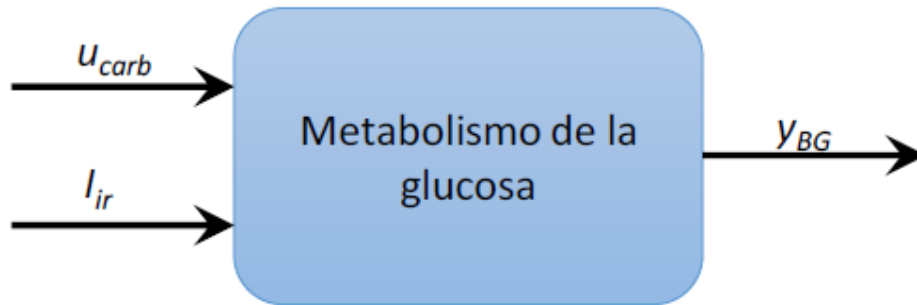


Figura 9 – Modelo del metabolismo de la glucosa. Elaboración propia a partir de [7]

Estas dos entradas, la ingesta de carbohidratos ( $u_{carb}$ ) e insulina suministrada ( $I_{ir}$ ), pueden ser complementadas con otras, como por ejemplo estrés, ejercicios, etc. Sin embargo, por motivos de simplicidad, en este caso se consideran únicamente las dos entradas primarias, en tanto en cuanto son las que tenemos modeladas, acotadas y cuantificadas.

La salida es, para cada instante de tiempo de las entradas, el nivel de glucosa en sangre del paciente, que hemos llamado  $y_{BG}$ .

Teniendo en cuenta los modelos de vanguardia estudiados, se presentan tres subsistemas:

- Subsistema de glucosa (GS, del inglés Glucose Subsystem)
- Subsistema de insulina (IS, del inglés Insulin Subsystem)
- Subsistema de interacción glucosa-insulina (GIIS, del inglés Glucose-Insulin Subsystem)

En el siguiente esquema podemos observar la composición interna, con los tres subsistemas. Se incluye también, sombreado para resaltar que no se utiliza, la hipotética entrada alternativa que hemos mencionado anteriormente.

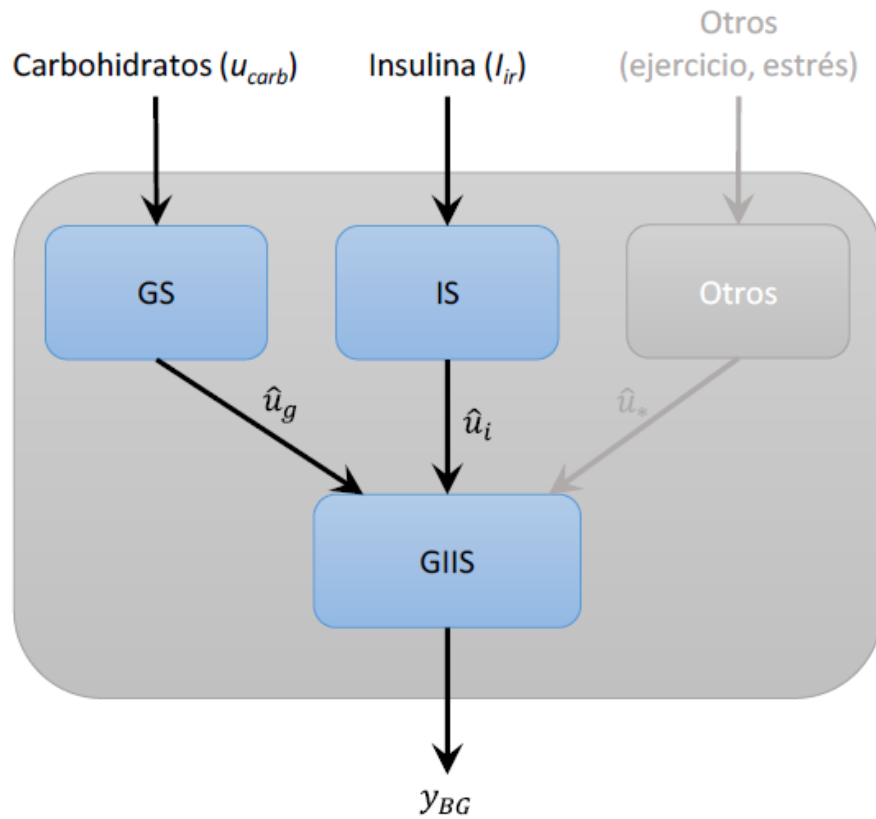


Figura 10 – Detalle del modelo fisiológico que describe la aparición de glucosa en plasma sanguíneo. Elaboración propia a partir de [7]

Pasaremos ahora a describir las ecuaciones que rigen los modelos más significativos encontrados en la literatura, para cada uno de los subsistemas definidos.

### 3.1. Subsistema de glucosa

En la definición del subsistema de glucosa, se ha decidido implementar el modelo propuesto por Chiara Dalla Man, Rizza y Cobelli en [7]. Este se puede esquematizar en tres compartimentos, como se muestra en la siguiente ilustración:



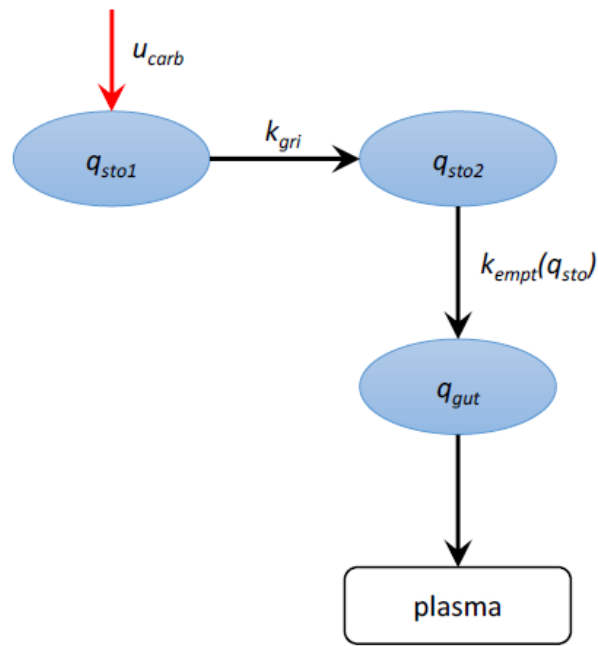


Figura 11 – Esquema de compartimentos del modelo para el subsistema de glucosa. Elaboración propia a partir de [7]

En este se representa el tránsito de la glucosa a través del estómago en sus fases sólida y líquida, para pasar luego al intestino delgado y diluirse en el plasma sanguíneo. El conjunto de ecuaciones que describe este proceso es el siguiente:

$$q_{sto}(t) = q_{sto1}(t) + q_{sto2}(t)$$

$$q_{sto1}(t + 1) = -k_{gri} \cdot q_{sto1}(t) + u_{carb}(t) \cdot \delta(t) \quad q_{sto1}(0) = 0$$

$$q_{sto2}(t + 1) = -k_{empt} \cdot q_{sto2}(t) + k_{gri} \cdot q_{sto1}(t) \quad q_{sto2}(0) = 0$$

$$q_{gut}(t + 1) = -k_{abs} \cdot q_{gut}(t) + k_{empt} \cdot q_{sto2}(t) \quad q_{gut}(0) = 0$$

$$\hat{u}_g(t) = \frac{f \cdot k_{abs} \cdot q_{gut}(t)}{m_b} \quad \hat{u}_g(0) = 0$$

donde:

- $\hat{u}_g$  [mg/kg/min] – Tasa de aparición de glucosa en sangre.
- $q_{sto1}$  y  $q_{sto2}$  [mg] – Cantidades de carbohidratos en el estómago, en las fases sólida y líquida, respectivamente.
- $u_{carb}$  [mg] – Cantidad de carbohidratos ingeridos.  $D$  es la misma cantidad, en gramos.

- $q_{gut}$  [mg] – Masa de carbohidratos en el intestino.
- $k_{gri}$  – Tasa de molienda.
- $k_{empt}$  – Tasa de vaciado del jugo gástrico.
- $m_b$  [kg] – Masa corporal.
- $k_{abs}$  – Tasa de absorción de glucosa en el intestino.
- $f$  – Fracción de absorción intestinal que aparece en plasma.

$k_{empt}$  viene definida por una ecuación no lineal de la siguiente forma:

$$k_{empt}(q_{sto}) = k_{min} + \frac{k_{max} - k_{min}}{2} \cdot \{\tanh[\alpha(q_{sto} - b \cdot D)] - \tanh[\beta(q_{sto} - c \cdot D)] + 2\}$$

Siendo  $\alpha$  y  $\beta$ :

$$\alpha = \frac{5}{2 \cdot D \cdot (1-b)} \qquad \beta = \frac{5}{2 \cdot D \cdot c}$$

En cuanto al valor de las constantes indicadas, se han recopilado parámetros del modelo para una población “media”, obteniendo información de varias publicaciones. [12] [13]

Estos valores se han utilizado en la literatura para realizar simulaciones sobre sujetos in-silico (pacientes sintéticos creados por un simulador reconocido) o sujetos in-vivo (reales).

Parámetro	Valor
$k_{gri}$	0.0558 [min <sup>-1</sup> ]
$k_{max}$	0.0558 [min <sup>-1</sup> ]
$k_{min}$	0.0080 [min <sup>-1</sup> ]
$k_{max}$	0.0568 [min <sup>-1</sup> ]
$B$	0.82
$C$	0.01
$F$	0.9

Tabla 2 – Valores típicos para la población promedio de los parámetros del modelo del subsistema de glucosa. [12] [13]

### 3.2. Subsistema de insulina

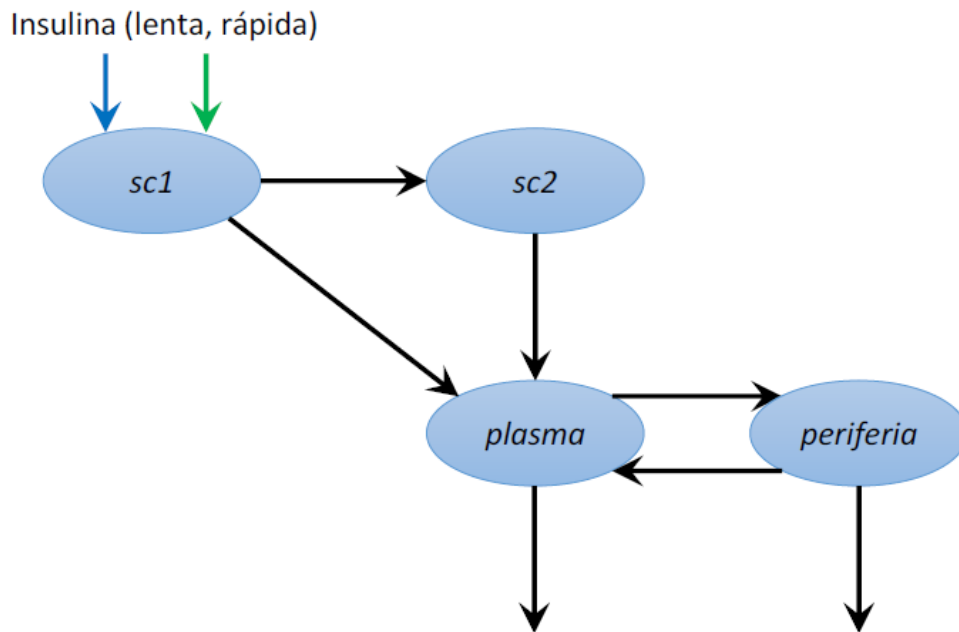


Figura 12 – Modelo de farmacocinética de la insulina. Elaboración propia a partir de [7]

De un modo similar al subsistema anterior, el de insulina se estructura en varios compartimentos, como se muestra en la figura siguiente:

En este caso, el modelo consta de insulinas de acción rápida y lenta de forma conjunta. Los compartimentos *sc1* y *sc2* representan los módulos de infusión de insulina subcutánea. La salida del modelo es la concentración de insulina en plasma.

Las ecuaciones del modelo son las que siguen:

$$I_{sc1}(t + 1) = I_{ir}(t) - (k_d + k_{a1}) \cdot I_{sc1}(t) \qquad I_{sc1}(0) = I_{sc1b}$$

$$I_{sc2}(t + 1) = k_d \cdot I_{sc1}(t) - k_{a2} \cdot I_{sc2}(t) \qquad I_{sc2}(0) = I_{sc2b}$$

$$s(t) = k_{a1} \cdot I_{sc1}(t) + k_{a2} \cdot I_{sc2}(t)$$

$$I_p(t + 1) = s(t) + m_1 \cdot I_l(t) - (m_2 + m_4) \cdot I_p(t) \qquad I_p(0) = I_{pb}$$

$$I_l(t + 1) = m_2 \cdot I_p(t) - (m_1 + m_3) \cdot I_l(t) \qquad I_l(0) = I_{lb}$$

$$\hat{u}_i(t) = \frac{I_p(t)}{V_i}$$

donde:

- $I_{sc1}$  [pmol/kg] – Cantidad de insulina no monomérica en el espacio subcutáneo.
- $I_{sc2}$  [pmol/kg] – Cantidad de insulina monomérica en el espacio subcutáneo.
- $k_d$  [ $\text{min}^{-1}$ ] – Ratio de disociación de la insulina.
- $k_{a1}$  [ $\text{min}^{-1}$ ] – Ratio de absorción de insulina no monomérica.
- $k_{a2}$  [ $\text{min}^{-1}$ ] – Ratio de absorción de insulina monomérica.
- $I_{ir}$  [pmol/kg/min] – Ratio de infusión de insulina exógena.
- $s(t)$  – Flujo de insulina que entra al torrente sanguíneo y se degrada según las ecuaciones  $I_p$  y  $I_l$ .
- $I_p$  [pmol/kg] – Masa de insulina en plasma.
- $I_l$  [pmol/kg] – Masa de insulina en el riñón.
- $V_i$  [litro/kg] – Volumen de distribución de insulina
- $\hat{u}_i$  [pmol/litro] – Concentración total de insulina en plasma
- $m_1$  – Ratio de depuración hepática
- $m_2, m_3, m_4$  [ $\text{min}^{-1}$ ] – Ratios que siguen las siguientes ecuaciones:

$$m_2 = \frac{3}{5} \cdot \frac{I_{CL}}{5 \cdot HE_b \cdot (V_i \cdot m_b)} \quad m_3 = m_1 \cdot \frac{HE_b}{1 - HE_b} \quad m_4 = \frac{2}{5} \cdot \frac{I_{CL}}{V_i \cdot m_b}$$

En las ecuaciones anteriores,  $HE_b$  es la extracción de insulina hepática basal, mientras que  $I_{CL}$ [litros/minuto] es la depuración de insulina. En estado estacionario, se cumple:

$$0 = I_{irb} - (k_d + k_{a1}) \cdot I_{sc1b}$$

$$0 = k_d \cdot I_{sc1b} - k_{a2} \cdot I_{sc2b}$$

$$s_b = k_{a1} \cdot I_{sc1b} + k_{a2} \cdot I_{sc2b}$$

De donde se obtiene que el valor basal de la insulina para los compartimentos subcutáneos es:

$$I_{sc1b} = \frac{I_{irb}}{k_d + k_{a1}}$$

$$I_{sc2b} = \frac{k_d}{k_{a2}} \cdot I_{sc1b}$$

$$s_b = I_{irb}$$

Además, aplicando el estado estacionario a las ecuaciones de  $I_p$  y  $I_l$  resulta:

$$0 = -(m_2 + m_4) \cdot I_{pb} + m_1 \cdot I_{lb} + s_b$$

$$0 = -(m_1 + m_3) \cdot I_{lb} + m_2 \cdot I_{pb}$$

De este modo, podemos deducir las expresiones que indican las cantidades de insulina en el hígado y en plasma, respectivamente, en un estado basal:

$$I_{lb} = I_{pb} \cdot \frac{m_2}{m_1 + m_3}$$

$$I_{pb} = \frac{I_{irb}}{m_2 + m_4 - \frac{m_1 \cdot m_2}{m_1 + m_3}}$$

Igual que para el caso del subsistema de glucosa, los valores promedio de los parámetros del subsistema de insulina para una población promedio, obtenidos tras ensayos clínicos asociados al proyecto europeo DIAdvisor™ [14], son los siguientes:

Parámetro	Insulina rápida	Insulina lenta	Unidades
$k_{a1}$	0.004	0.0002	min <sup>-1</sup>
$k_{a2}$	0.0182	0.00091	min <sup>-1</sup>
$k_d$	0.0164	0.00164	min <sup>-1</sup>
$m_1$	0.1766	0.1766	min <sup>-1</sup>
$V_1$	0.05	0.05	Litros / Kg
$I_{CL}$	1.1069	1.1069	Litros / min
$HE_b$	0.6	0.6	Sin dimensiones

Tabla 3 – Valores típicos para población promedio de los parámetros del modelo de subsistema de insulina. [13]

### 3.3. Subsistema de interacción glucosa – insulina

Existen dos tendencias en cuanto a modelar esta etapa del metabolismo de la glucosa. La primera es implementar el subsistema de interacción como un modelo fisiológico de bucle abierto, como el publicado originalmente por Dalla Man en [7].

Posteriormente se emplean técnicas de identificación para adaptar los parámetros del modelo a cada sujeto de estudio.

La segunda tendencia es la identificación de un modelo predictivo que sea capaz de realizar una estimación a futuro teniendo en cuenta un histórico de datos.

Contrastando ambas opciones, observamos que un modelo fisiológico es, sobre todo, cualitativo. De este modo, el comportamiento de las funciones que logremos como resultado serán bastante similares a la realidad (la ingestión de carbohidratos tendrá un efecto de subida en la curva de glucosa, la administración de insulina resultará en una disminución, etc.)

Por otra parte, un modelo predictivo se caracteriza por ser cuantitativo. En otras palabras, las curvas de respuesta se aproximarán a su valor numérico mejor que las del modelo fisiológico. No obstante, dado que el modelo predictivo basa sus estimaciones en lo ocurrido en el pasado, corre el riesgo de errar ante una entrada anómala o imprevista.

Sin embargo, el modelo predictivo permite flexibilidad de actuación a corto o medio plazo, y el proceso de identificación de parámetros es relativamente sencillo, como hemos mencionado previamente durante el apartado correspondiente al subsistema de insulina.

Habitualmente, los modelos fisiológicos se emplean para implementar simuladores, mientras que los modelos predictivos son la elección acostumbrada para controladores en situaciones reales.

Dado que nos centramos en el segundo punto, hemos optado por la implementación y prueba de modelos predictivos en este trabajo. Para la identificación de modelos válidos se ha recopilado datos de pacientes *in silico* e *in vivo*, a partir de simuladores, de la literatura disponible ya mencionada.

La identificación de los subsistemas de los modelos predictivos se ha llevado a cabo mediante un método de regresión múltiple basado en el espacio de estado, la función N4SID del “System Toolbox” (un conjunto de herramientas y funciones para facilitar el trabajo con sistemas de ecuaciones) de MATLAB.

En concreto, el método N4SID estima un modelo en el espacio de estados de orden  $n$ , usando para ello medidas experimentales de entrenamiento. En resumen, estima las matrices  $A$ ,  $B$ ,  $C$ ,  $D$  y  $K$  del siguiente sistema de ecuaciones:

$$\dot{x}(t) = Ax(t) + Bu(t) + Ke(t)$$

$$y(t) = Cx(t) + Du(t) + e(t)$$

donde:

- $u(t)$  es el vector de entradas al sistema.
- $x(t)$  es el vector de variables de estado.
- $y(t)$  es el vector de salida.
- $e(t)$  es un vector de perturbación.

La medida que hemos empleado para evaluar la calidad del modelo es el Error Cuadrático Medio normalizado (en adelante FIT), que se define del siguiente modo:

$$FIT = 100 \cdot \left(1 - \frac{\|y - \hat{y}\|}{\|y - \bar{y}\|}\right)$$

donde:

- $y$  es la medida real.
- $\bar{y}$  es el promedio de la medida real.
- $\hat{y}$  es la medida estimada por el modelo.

En nuestro caso, el vector de entrada  $u$  corresponde al ratio de aparición de glucosa en sangre y la concentración de insulina  $[\hat{u}_g, \hat{u}_i]$ , mientras que el vector  $y$  se corresponde con la cantidad de glucosa en sangre [mg/dl], denotado anteriormente como  $y_{BG}$ .

De este modo, usaremos como datos de entrada medidas de glucosa reales, ingesta de carbohidratos e infusión de insulina de un paciente para alimentar los modelos GS e IS, para posteriormente llamar a N4SID y estimar la interacción, devolviendo una salida con una medida del error estimado (FIT).

Pasaremos, en el apartado siguiente, a detallar la implementación del sistema general y del proceso de alimentado de fichas de datos, creando un ciclo simulación – ajuste hasta alcanzar valores óptimos para los parámetros. Este proceso emplea la medida de FIT obtenida en la simulación para modificar los parámetros hasta encontrar los más adecuados para el paciente analizado.

## 4. Implementación

---

En este apartado se profundiza en los detalles relativos tanto al proceso como a la organización de la implementación.

Esto incluye desde las herramientas utilizadas, pasando por la motivación que llevó al uso de las mismas, hasta la organización y especificación de las clases internas y el proceso general de funcionamiento o flujo de trabajo, donde describiremos el modo en que transita una iteración, pasando por todas sus etapas.

En este sentido, y dado que simplemente incluyen las ecuaciones que definen el comportamiento del modelo que ya se ha comentado anteriormente, las clases “Subsistema\_glucosa” y “Subsistema\_insulina” tan solo requieren de una presentación breve.

### 4.1. Herramientas empleadas

#### 4.1.1. Eclipse y Java 7

El entorno de desarrollo (en adelante IDE, del inglés “Integrated Development Environment”) elegido para llevar a cabo la implementación del modelo en código Java ha sido Eclipse, en su última versión estable –“Photon”, v4.8-. [15]

Este software fue creado originariamente por IBM para la creación de un sucesor de la familia de herramientas “VisualAge” [16]. Sin embargo, en 2003 la Fundación Eclipse, de creación exprofeso para tal menester, se hizo cargo del proyecto, liberándolo primero bajo la licencia “Common Public License” y más tarde bajo la “Eclipse Public License” [17], ambas de software libre.

El leitmotiv del proyecto es “Una especie de herramienta universal, un IDE abierto y extensible para todo y nada en particular”.

Estando concebido como una Plataforma de Cliente Enriquecido, se caracteriza por basarse en la integración de componentes independientes. En concreto, el entorno es ampliable con módulos que permiten extender su funcionalidad en gran variedad de sentidos, aportando ligereza y posibilidades de personalización a la base, en contraste con otros IDE donde se incorporan numerosas características que no siempre resultan de utilidad al usuario final.

Algunos ejemplos conocidos de estas extensiones son las que permiten utilizar lenguajes como Python, JavaScript o C/C++, o trabajar directamente con ciertos Sistemas de Gestión de Bases de Datos, como MariaDB o PostgreSQL.

Cabe señalar que desde la web de Eclipse se ofrecen diferentes bases prefabricadas, que ya integran ciertos módulos, de forma que están orientadas al desarrollo en ciertos lenguajes y con características concretas.



La empleada en nuestro caso es aquella que ya incorpora el SDK de Java, editor de texto inteligente, compilador en tiempo real, asistentes para creación de proyectos y modelos completos de las clases base del lenguaje, e integración con el Sistema de Control de Versiones “Git”, que hemos empleado durante el desarrollo del proyecto para mantener el control en los sucesivos cambios operados en el software, apoyándonos en el repositorio online de Atlassian, BitBucket.

La elección de Java 7 como lenguaje general de trabajo ha contribuido a asentar los conocimientos adquiridos durante el grado, durante el cual ha sido el principal vehículo en casi todas las asignaturas dedicadas a la programación o la algoritmia. Dado que el trabajo de investigación en el que se participa versa sobre disciplinas (medicina, biología, endocrinología) totalmente desconocidas, es pertinente partir de una base sólida como la aportada durante los años de estudio.

En esta dirección, elegir este lenguaje ha facilitado el desarrollo y ha resultado de utilidad para enfrentar la gestión de diferentes entornos de trabajo al ser multiplataforma, en tanto en cuanto siendo este desarrollo un prototipo, ha sido utilizado y probado en entornos Windows, Linux y macOS, con lo que no hemos padecido ningún perjuicio al respecto, sino que ha eliminado este tipo de limitaciones.

Además, como parte de la motivación para la elección de esta versión del lenguaje, hemos podido beneficiarnos del empleo del “try” con recursos, evitando tener que utilizar un bloque “finally” para cerrar recursos como *buffers* o ficheros abiertos; y de la simplificación en el uso de genéricos gracias a la inferencia de tipos, de modo que la declaración de los mismos es más sencilla y rápida (operador “<>”):

```
List<String> list=new ArrayList<>();  
Map<String, File> map=new HashMap<>();
```

También resulta reseñable la posibilidad de usar estructuras “switch” con String, útil para filtrar argumentos de entrada, o las estructuras “multicatch” (evitando así la repetición de código que supone escribir múltiples bloques “catch” para capturar diferentes excepciones para las que se va a realizar el mismo procedimiento u operaciones), con lo que el tratamiento de estas casuísticas se ha visto simplificado.

#### 4.1.2. MATLAB

En un apartado anterior hacíamos mención a cierta función de MATLAB. En este punto comentaremos brevemente los usos y características principales de este software, con la versión R2018a (v9.4.0) del cual hemos interactuado desde nuestro programa Java, enviando los cálculos a realizar y recibiendo su resultado, para después tratarlo de la forma correspondiente.

MATLAB es una herramienta propietaria de desarrollo para software matemático que incorpora un IDE y un lenguaje de programación propios, el conocido como lenguaje M. Es muy conocido y utilizado en los entornos universitario y de investigación.

Habitualmente es el candidato elegido para:

- Trabajo con matrices
- Implementación de algoritmos
- Visualización de datos
- Simulación de sistemas
- Funciones de cálculo para programas externos en un esquema de trabajo cliente-servidor (nuestro caso de uso).

En nuestro caso, tras invocar una instancia del programa o conectar con una existente, haremos uso de la función N4SID del “System Identification Toolbox”, que nos devolverá, ante la entrada de varios vectores con la glucosa, insulina y carbohidratos calculados por los subsistemas del modelo, el resultado de la interacción de los mismos, en forma de vector, indicando la ratio de aparición de glucosa en plasma sanguíneo, y una estimación de la corrección del mismo.

A pesar de que la función mencionada existe también en el software libre Octave [18], que se promociona como un equivalente de código abierto de MATLAB, se ha encontrado más productivo utilizar la licencia de éste último que la Universidad pone a nuestra disposición, en tanto en cuanto la interfaz entre el mismo y Java está más depurada y mantenida que la de su contraparte gratuita. Además, en la literatura se citan numerosos ejemplos de uso de este software, por lo que se ha considerado adecuado su empleo por ser una opción de solvencia demostrada. [13]

## 4.2. MatlabControl

El hecho emplear funciones propias de MATLAB como N4SID desde una capa externa o superior que facilitase la gestión de archivos nos llevó a utilizar la interfaz Java ofrecida por el propio MATLAB (JMI, del inglés “Java Matlab Interface”).

No obstante, el empleo directo de dicha interfaz requiere de un intensivo trabajo de gestión de estructuras de datos de MATLAB, porque no son directamente compatibles con las disponibles en Java, de modo que, con el objetivo de disminuir la posible problemática o incorrección en los datos originada por este punto, decidimos emplear una capa intermedia para interactuar con la interfaz mencionada, de modo que estas conversiones resultaran transparentes en la medida de lo posible.

Por ello elegimos una implementación de las funcionalidades que necesitábamos, que permite el uso relativamente directo de las instrucciones de MATLAB “eval” (evaluación), “feval” (evaluación de función) o “returningEval” y “returningFeval”, que a su vez invocan directamente la ejecución de ciertos cálculos, con la posibilidad de devolver el resultado al programa Java que los solicita –“returning”–.

Dichas instrucciones están empaquetadas en la librería MatlabControl [19] (la versión empleada ha sido la última disponible, 4.1.0), cuyo autor, Joshua Kaplan, pone a disposición pública bajo licencia BSD, lo que permite su uso incluso en proyectos comerciales so mención de la renuncia correspondiente.

Además, nos permite también gestionar el control de conexiones a las sesiones de MATLAB, de modo que se puede coordinar la conexión a, o la apertura de una sesión ya abierta del mismo, lo que a su vez redundaría en facilitar el trabajo general con la herramienta y, en nuestro caso particular, el proceso de pruebas en general, evitándonos tener que abrir sesiones nuevas o reiniciar la aplicación en cada ocasión.

El uso que ofrece esta librería pasa por instanciar la clase “MatlabProxyFactory” con las opciones deseadas, en nuestro caso “setUsePreviouslyControlledSession(true)” para conectar a una sesión de MATLAB abierta si la hubiera, o “hidden(true)” si hemos configurado la ejecución con el atributo “-c” o “—console”.

A esta clase factoría podemos solicitar un objeto “MatlabProxy”. Sobre este elemento podemos invocar los métodos “eval”, “feval” y las variedades “returning” de cada uno, para realizar las operaciones indicadas en las cadenas suministradas.

Dicho de otro modo, en la práctica permite el acceso a una línea de comandos MATLAB desde el código Java, volviendo transparente la gestión de los elementos intermedios y permitiendo que el trabajo de desarrollo se centre en el envío, recepción y proceso de resultados.

### 4.3. OpenCSV

OpenCSV es una librería Java de código abierto cuyo objetivo principal es la interacción sencilla con archivos separados por comas, en inglés Comma-Separated-Values (CSV).

De este modo, en la versión que estamos utilizando, la 3.10, se dispone de las clases “CSVReader” y “CSVWriter”, que ofrecen métodos para leer un CSV como una lista de vectores de tipo “String”, y para escribir a un fichero con el formato correspondiente, respectivamente.

Su utilización es por tanto relativamente simple, y queda embebida en el elemento gestor de archivos (“FileManager”) para realizar las operaciones necesarias en cada caso, que describiremos en el apartado pertinente.

### 4.4. GlucoJ

Hablaremos ahora de la clase principal del programa Java desarrollado, que realiza las funciones de punto de entrada para el usuario, y desde la invocación de la cual se gestionan las opciones para su ejecución, a saber:

- Estipular el directorio de trabajo - Atributo “-d” o “—folder” en la llamada de línea de comandos). Por defecto, el valor es el directorio actual, “.”
- Indicar el tiempo de espera entre lecturas del directorio de trabajo – Atributo “-t” o “—timeout”.

Desde que arranca su ejecución, se buscan continuamente nuevas fichas con parámetros de los subsistemas para volver a lanzar los cálculos correspondientes. Con esta opción, se permite indicar el tiempo a esperar antes de volver a procesar el directorio, de modo que es posible, por ejemplo, ajustar la comunicación entre el simulador y el ajustador. Por defecto, esperamos tres segundos.

- Configurar la ejecución en modo consola – Atributo “-c” o “—console”.  
Invoca la ejecución de MATLAB sin la interfaz gráfica completa, lo que disminuye el tiempo completo de ejecución. Cabe señalar que, si existe ya un MATLAB abierto, se conectará al mismo ignorando esta opción. Por defecto, empleamos la interfaz gráfica completa.
- Mostrar instrucciones – Atributo “-h” o “—help”.  
Muestra un breve texto explicativo acerca del uso de la herramienta que contiene los diferentes atributos y su función.

Además de los diferentes argumentos de invocación desde línea de comandos, esta clase, que como hemos visto hace las veces de dirección y control del flujo de ejecución, es la encargada de interactuar con las unidades funcionales que representan los diferentes subsistemas, así como con la clase encargada del trabajo con el sistema de archivos (“FileManager”), para realizar el proceso de alimentación, cálculo y repetición que detallaremos en posteriores apartados.

#### 4.5. Subsistemas de glucosa e insulina

En este caso necesitábamos disponer de sendas clases Java en las que encapsular las ecuaciones y cálculos a realizar en cada subsistema. El objetivo del desarrollo de estos elementos era obtener un comportamiento similar al descrito en la presentación del modelo, una caja negra que representa cada subsistema, y a la que proporcionar las entradas para obtener las salidas correspondientes.

De este modo, en el “Subsistema\_glucosa” hemos plasmado las ecuaciones del trabajo de Dalla Man en [12], de manera que desde el “FileManager” se pasan los parámetros contenidos en la ficha (archivo .pro) que se esté procesando en ese instante, así como la entrada pertinente en este caso, el vector que contiene los carbohidratos ingeridos por el paciente, indicados en el CSV que acompaña a estas fichas.

Por otra parte, y de forma similar, nuestra clase “Subsistema\_insulina” contendrá las ecuaciones que modelan el comportamiento del metabolismo de la hormona que ya hemos descrito en puntos anteriores, por lo que necesitará la entrada de otra parte de los parámetros de la ficha, y los vectores que contienen la insulina inyectada (“Rápida”) y la basal obtenidas del fichero CSV con las mediciones del paciente.

## 4.6. FileManager

Esta clase fue creada con el objetivo de actuar como capa intermedia entre los ficheros necesarios para lanzar las diferentes iteraciones de cálculo, y el proceso de llenado y preparación de las estructuras de datos necesarias.

De este modo, y como veremos en el apartado donde se describe el flujo de trabajo, desde la clase principal se instancia esta otra, indicando como parámetro la ruta del directorio de trabajo.

Una vez disponible el objeto con la información correspondiente hará las veces de interfaz, para lo que existen una serie de métodos “get” desarrollados para solicitar la lectura e incorporación de la información de los archivos “.pro” (parámetros del paciente para una iteración del modelo) y “.csv” (mediciones de los niveles del paciente real) a las estructuras de datos, que contendrán entonces lo necesario para generar el modelo y lanzar los cálculos pertinentes en los respectivos subsistemas.

## 4.7. Ajuste – Particle Swarm Optimization (PSO)

Tras realizar la primera iteración de las simulaciones, dispondremos de unos primeros resultados del modelo. Será entonces necesario un proceso de ajustado de los parámetros de entrada, para que la salida alcanzada sea lo más parecida posible a los valores reales medidos del paciente.

Para realizar este afinado, se contrastan los resultados calculados contra los datos reales del paciente. Veremos una retrospectiva general de todo el funcionamiento en el siguiente apartado, pero ahora profundizaremos en el propio ajustado.

En función de la calidad de la comparación, es decir, el porcentaje de error o similitud entre lo obtenido por nuestro programa y los datos suministrados por el paciente, se generan nuevas “fichas” (nuevos archivos “.pro”) que alimentarán la siguiente iteración del entrenamiento del modelo con ligeras modificaciones de los valores de los parámetros, hasta alcanzar el mejor porcentaje de acierto posible con los datos disponibles.

Este error se calcula a partir de:

$$\sum_{i=0}^{minutes} ||Glucosa_{real_i} - Glucosa_{calculada_i}||$$

Donde  $|| \cdot ||$  denota la norma euclidiana, de modo que se considera el error cuadrático medio.

El algoritmo que empleamos para acomodar estos parámetros de los que vamos incrementando o disminuyendo ligeramente los valores en cada ciclo, es la optimización por enjambres de partículas, conocido en inglés como *Particle Swarm Optimization* (PSO) [20], utilizado con frecuencia en problemas de optimización.

Del mismo modo que los enjambres de abejas tienen en cuenta las zonas conocidas con mayor cantidad de flores para buscar nuevas fuentes de polen, este cálculo metaheurístico (esto es, que no asume hipótesis o reglas sobre el problema a optimizar) trata de, emulando el movimiento de

un enjambre de abejas o pájaros, orientar las soluciones obtenidas teniendo en cuenta las mejores hasta el momento.

En nuestro caso, cada abeja, pájaro o partícula hace las veces de un vector de parámetros, y la posición de la misma en un instante de tiempo sería una de nuestras soluciones. El desplazamiento del insecto, o la función que dibujan las diferentes soluciones que conseguimos, será aleatorio dentro de su dominio de búsqueda, pero tendrá en cuenta las mejores soluciones conseguidas hasta ahora para orientar su movimiento. Este dominio de búsqueda está acotado hasta en el 500% del valor inicial de cada parámetro.

A medida que encontremos nuevas soluciones que mejoren las anteriores, se anotarán para tratar de mejorar las sucesivas iteraciones de búsqueda de las siguientes.

Este proceso lo haremos gran cantidad de veces (sobre quinientas) hasta encontrar los valores personalizados más adecuados. Cabe notar que el algoritmo utilizado, como toda metaheurística, no asegura una solución óptima.

#### **4.8. Validación – Matrices de error de Clarke y Parkes**

Existen variedad de formas de determinar la bondad del ajuste. Las más utilizadas en la literatura son los “Error Grid” de Clarke [21] y de Parkes [22], que se consideran “gold-standard” (pruebas de diagnóstico de referencia con la máxima fiabilidad, si bien no absoluta certeza). Por ello, han sido estas las herramientas elegidas en nuestro caso.

Estos diagramas fueron desarrollados en 1987 y 2000 respectivamente, con la idea de cuantificar el error entre los niveles de glucosa medidos por el paciente y las medidas obtenidas por los aparatos de referencia (sensores, etc.).

Por extensión los podremos utilizar para estimar el error entre los niveles reales de glucosa en sangre y la estimación aportada por el modelo.

A nivel funcional, consisten en dibujar, mediante diagramas de dispersión, puntos que representan los niveles de glucosa en sangre del paciente junto con valores devueltos por el modelo, en cinco regiones:

- La región A son aquellos valores que están por debajo de un 20% de la referencia del sensor.
- La región B contiene puntos que están fuera del 20% pero no conducen a tratamientos inapropiados.
- La región C corresponde a los puntos que conducen a un tratamiento innecesario.
- La región D contiene los puntos que indican un fallo potencialmente peligroso en la detección de hiperglucemia o hipoglucemia.
- La región E son aquellos puntos que confundirían el tratamiento de hipoglucemia e hiperglucemia, y viceversa.

Por lo tanto, un ajuste mejor es aquel en que todos los puntos se encuentren en la región A. Más adelante, en el capítulo dedicado a la presentación de resultados, podemos encontrar los diagramas asociados a nuestra experimentación.

La ventaja añadida de utilizar estos modelos de error probados es que, si el calibrado es bueno, podríamos aventurarnos en predecir la evolución del nivel de glucosa del paciente en las próximas horas, de modo que un médico endocrino podría dar al paciente pautas de tratamiento o comprobar si padece alguna patología mediante nuestro modelo.

## 4.9. Flujo de trabajo

Hemos hablado sobre fichas, parámetros, vectores de entrada y salida, resultados obtenidos y diferentes mecanismos (ajuste, validación...), en el detalle de cada uno de los procesos que componen el sistema desarrollado.

Sin embargo, con el objetivo de facilitar la visión global de todo el conjunto, es conveniente disponer de una perspectiva esquemática de todo el proceso.

Partiremos de este modo de una situación inicial en la que tenemos uno o más ficheros con extensión “.pro” (en adelante fichas), que contendrán los valores de los parámetros del paciente.

Estos parámetros son las variables que hemos comentado durante la presentación de los diferentes subsistemas del modelo, como  $k_d$ ,  $k_{a1}$  o  $k_{a2}$ , para el caso del de insulina, o  $k_{gri}$  y  $k_{empt}$  para el de glucosa.

En el archivo inicial que empleamos, podemos observar varios datos como el nombre del paciente, el peso y valores de los parámetros mencionados, entre otros. Estos serán los parámetros utilizados para la prueba de concepto de la que mostraremos más adelante los resultados en el capítulo correspondiente. El resto de valores son los promedios presentados en cada subsistema.

Por otra parte, tendremos también un único fichero con extensión “.csv”, en cuyo interior podremos encontrar las mediciones de insulina suministrada, carbohidratos ingeridos y glucosa en sangre del paciente, es decir, los datos que alimentarán el modelo para observar su reacción. Seguidamente podemos un fragmento del fichero, donde podemos apreciar su formato y parte del contenido:

Hour	Glucose	Ch.	Ins. Short Units	Ins. Long Units
0:01:00	80.0	0.0	0.0125	0.0
0:02:00	80.0	0.0	0.0125	0.0
0:03:00	80.0	0.0	0.0125	0.0
0:04:00	80.0	0.0	0.0125	0.0
0:05:00	80.0	0.0	0.0125	0.0
0:06:00	72.0	0.0	0.0125	0.0
0:07:00	72.0	0.0	0.0125	0.0

Figura 13 – Fragmento del fichero CSV empleado en nuestros experimentos.  
Elaboración propia.

Es importante dirigir la atención a los parámetros del archivo PRO llamados “turnos\_ini” y “turnos\_fin”, con los que se acota el periodo temporal (turnos o filas) que elegiremos del CSV para entrenar el modelo, mientras que el total del mismo será empleado para comparar la corrección de la predicción.

Así pues, el funcionamiento general del sistema consiste en estar en continua ejecución, revisando si en el directorio de trabajo (indicado para el arranque) existen fichas no procesadas todavía.

Cabe apuntar que durante la misma ejecución no se usará dos veces la misma ficha para lanzar el modelo, puesto que existe un registro de las fichas utilizadas hasta el momento que se va comprobando para evitarlo, con persistencia en el archivo “procesados.txt”. Este fichero se genera automáticamente en el directorio de trabajo, una vez se inicia el sistema.

De este modo, se toman esas fichas y se inicializan los subsistemas, calculando las ecuaciones que hemos comentado en cada caso.

En resumen, se lanza el modelo con los valores contenidos en las mismas, siguiendo el funcionamiento descrito en la siguiente ilustración:



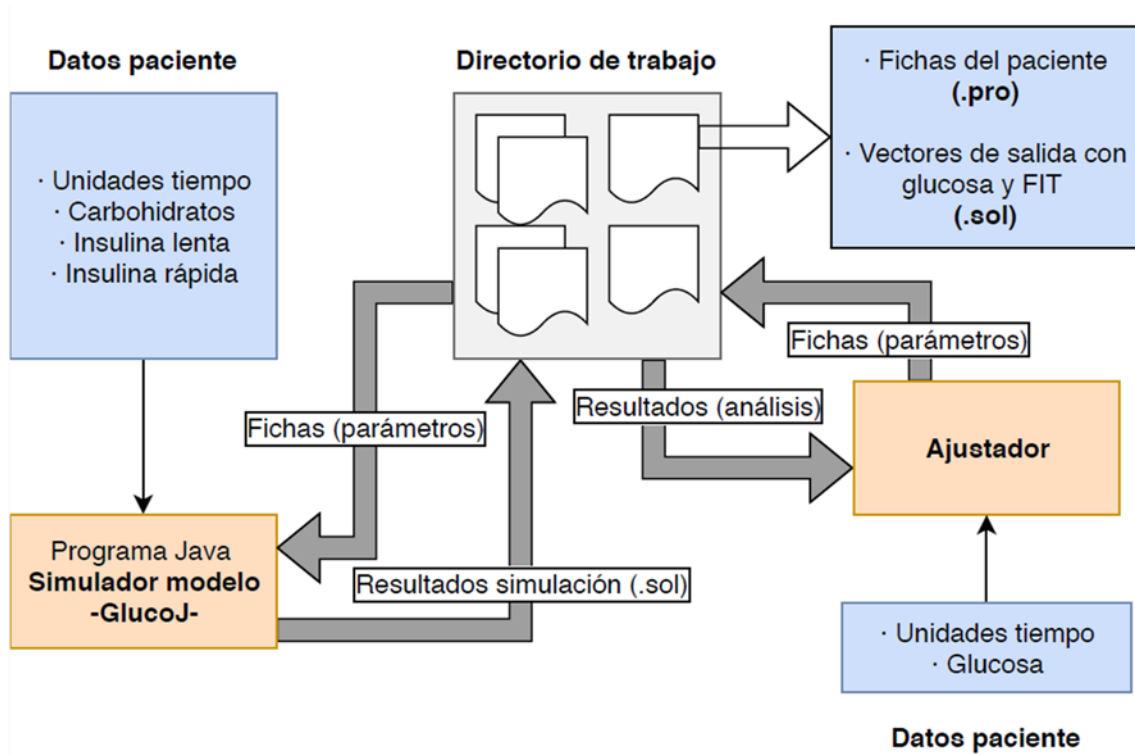


Figura 14 – Esquema general de funcionamiento del sistema. En azul contenido y/o datos de entrada, en amarillo unidades funcionales, en gris flujo de información. La parte de la simulación en MATLAB queda contenida en el bloque funcional GlucoJ. Elaboración propia.

Describiremos ahora el detalle de una iteración completa de la simulación, desde que se suministran las primeras fichas y la medición de niveles de glucosa, carbohidratos e insulina reales del paciente.

En primer lugar, generamos una instancia de “MatlabProxyFactory” para solicitar un objeto del tipo “MatlabProxy”. Este procedimiento se traduce en lanzar MATLAB, que quedará preparado para la recepción de información, que tendrá lugar una vez se hayan realizado los cálculos preliminares en Java.

Mientras esta inicialización tiene lugar, creamos otra instancia, esta vez de la clase “FileManager”, cuyo atributo es la ruta del directorio de trabajo.

Con los ficheros que se encuentran en esa carpeta rellenamos los vectores de entrada para los respectivos subsistemas, tomando en consideración el intervalo de tiempo o “turnos” del CSV (es decir, las filas, que representan unidades de tiempo) que se indican como parámetro en la ficha que estemos procesando.

El siguiente paso es lanzar a MATLAB los vectores de salida de cada subsistema (la cantidad de glucosa e insulina que aparece en sangre según los cálculos realizados). El trabajo realizado hasta este punto se esquema en la siguiente ilustración:

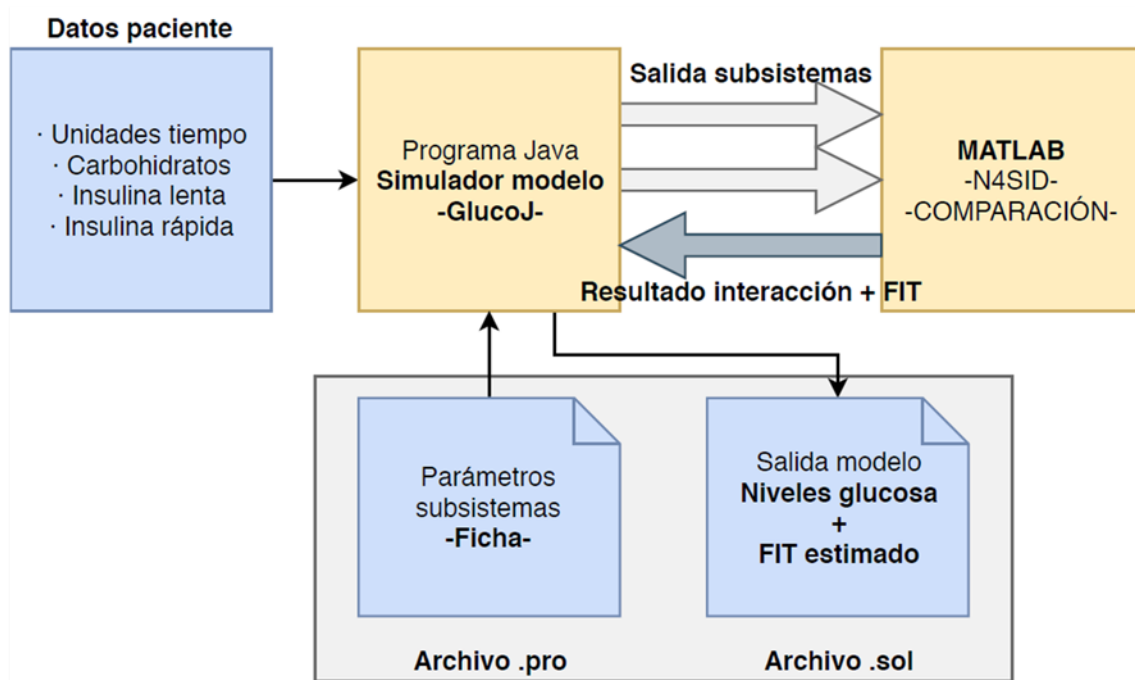


Figura 15 – Detalle de funcionamiento perspectiva GlucoJ – MATLAB. Elaboración propia.

El siguiente paso es llamar a la función N4SID. El programa Java quedará pendiente del resultado de las operaciones que hemos solicitado, que recibirá una vez hayan terminado.

Acto seguido, escribiremos el vector que indica la aparición de glucosa en sangre según nuestra estimación, así como el FIT o porcentaje de error estimado, en un archivo “.sol” (SOL), lo que representa el producto de GlucoJ para una ficha o iteración.

A partir de la existencia de este archivo SOL, puede empezar el siguiente paso del procedimiento: el ajuste o calibrado del modelo. Como se indica en el apartado pertinente, el ajustador contrasta lo que hemos dejado en el SOL contra los datos reales de que disponemos para comprobar el margen de error.

En función de esta diferencia, modifica levemente los parámetros de la ficha a la que corresponde esa solución del modelo siguiendo el algoritmo PSO comentado, para volverlo a lanzar en busca de la optimización del resultado.

Cuando ya se considere que estamos obteniendo un resultado aceptable o adecuado, es decir, que hayamos realizado tantas repeticiones del proceso como consideremos necesarias hasta alcanzar un error mínimo, podremos avanzar a la siguiente etapa: validar el modelo.

Con la validación cerramos el esquema general y concluye el trabajo para un único paciente, puesto que si al emplear los diagramas de Clarke y Parkes, los resultados son óptimos, dispondremos de un modelo personalizado para el paciente objeto de análisis.

Por tanto, deberá poder explicar y predecir las variaciones en sus niveles de glucosa con las entradas proporcionadas, por lo que será de gran ayuda para un endocrino a la hora de concretar decisiones o delimitar tratamientos para dicho paciente.

#### 4.10. Análisis de las salidas

Quedan por revisar los resultados que produce cada etapa de nuestro programa tras completar su ejecución.

Analizaremos la salida de cada una de las fases para comprobar las funciones dibujadas y el comportamiento de las mismas, es decir, estudiaremos el efecto en el nivel de glucosa o insulina que provoca una ingesta o una infusión subcutánea de la hormona, según sean procesadas por sus respectivos subsistemas.

Cabe recordar que vamos a emplear para la prueba los parámetros mostrados durante la explicación del flujo de ejecución, que son valores promedio de los que partimos para ajustarlos con nuestro procedimiento.

En las gráficas siguientes observaremos las funciones que dibujan los vectores resultantes de cada subsistema: la aparición de glucosa en sangre a partir de los carbohidratos ingeridos y la aparición en plasma de la insulina suministrada.

En primer lugar, analizaremos el vector que nos devuelve el subsistema de glucosa después de procesar la entrada de carbohidratos contenida en el CSV. Como se puede apreciar, cada ingesta se traduce en un aumento en los niveles glucémicos.

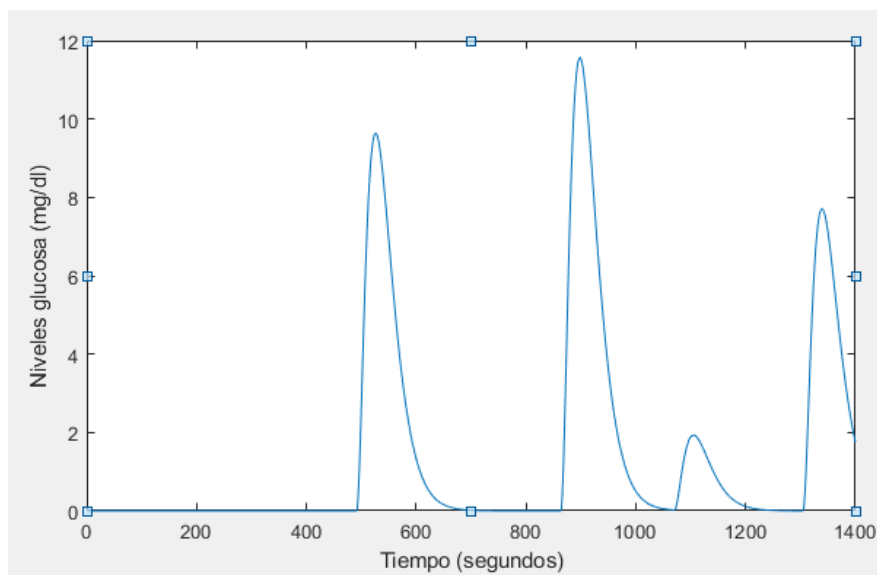


Figura 16 – Ejemplo de aumento de niveles glucosa en paciente tras las respectivas ingestas de carbohidratos. Elaboración propia.

Paralelamente obtendremos la salida del subsistema de insulina, que de forma similar al de glucosa dibuja una función, en este caso mostrando el pico en los niveles de la hormona, correspondiente a la inyección que el paciente se administra aproximadamente en el instante mil ochenta del CSV.

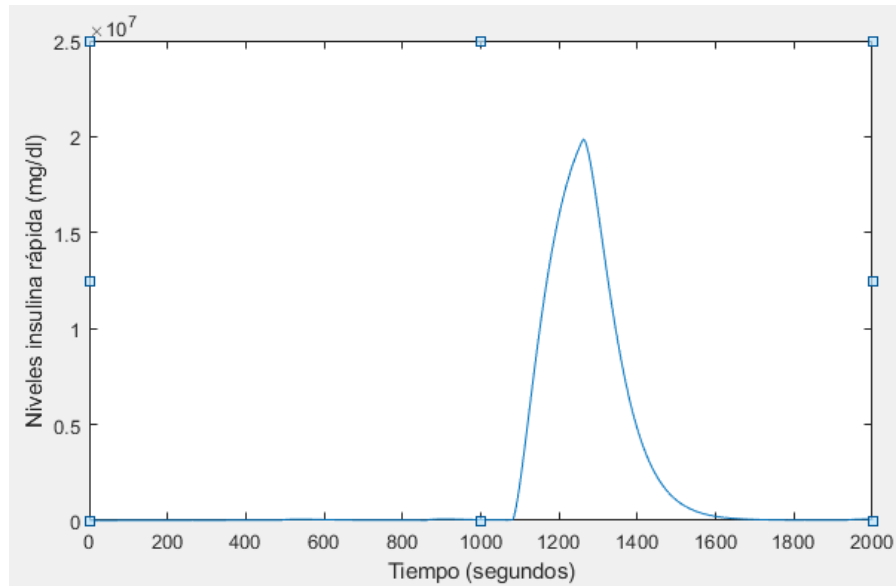


Figura 17 – Ejemplo de aumento de niveles de insulina en plasma sanguíneo tras la infusión subcutánea de insulina. Elaboración propia.

A continuación, una vez obtenidas las entradas necesarias, calcularemos la interacción de glucosa e insulina en un horizonte temporal mediante la función N4SID de MATLAB.

Del uso de esta función resultará el producto final para esta iteración, devolviendo el vector ya comentado con los niveles definitivos de glucosa en plasma, así como la medida “FIT”, la estimación del error de la predicción.

A partir de este punto tendrá lugar el ajuste mencionado anteriormente, con el que buscaremos acercar los parámetros de la siguiente ficha que alimentará el modelo a los valores óptimos para adaptarse al paciente.

Realizadas las iteraciones pertinentes de este proceso de simulación, entrenamiento y ajuste, conseguiremos como resultado un modelo entrenado para un paciente específico, que se tendrá que validar con los diagramas de Clarke y Parkes, con lo que terminará todo un ciclo de ejecución.

## 5. Resultados obtenidos

En este capítulo comentaremos los resultados de los ensayos realizados tanto para el entrenamiento del modelo, como las variaciones en los valores de los diferentes parámetros del sistema, así como las conclusiones respecto a la validez del mismo.

Para poder obtener datos de un paciente a los que adaptar o adecuar el modelo, uno de los colaboradores en el mismo tomó las medidas de sus niveles de glucosa en plasma durante veinte horas y anotó las diferentes ingestas de carbohidratos que realizó en este periodo. El peso del paciente es de 95 kilos, y la actividad física realizada durante el tiempo que dura la muestra se puede considerar como media, con periodos de ejercicio y de descanso.

En primer lugar, con el propósito de facilitar la comprensión de los resultados a presentar, hemos trasladado las ingestas de carbohidratos presentes en el CSV a un soporte visual:

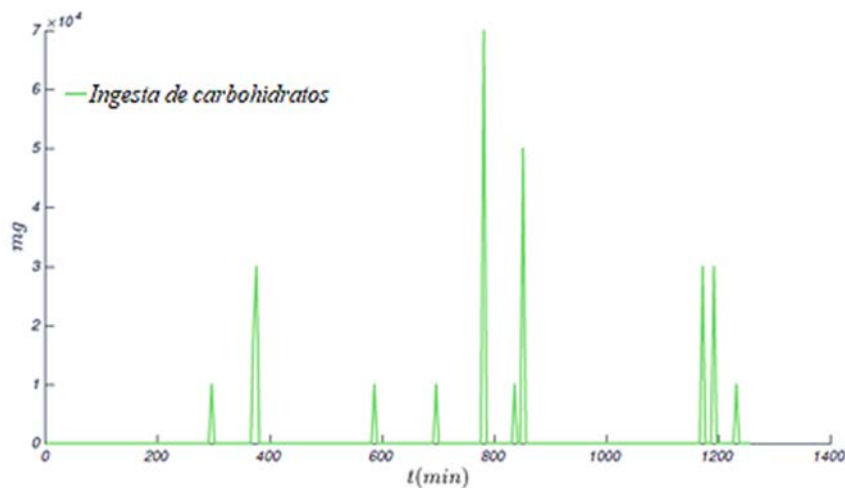


Figura 18 – Ingesta de carbohidratos distribuida en el tiempo. Elaboración propia a partir de los datos disponibles de un paciente sano real.

El primer paso para considerar los parámetros obtenidos tras el ajuste PSO como válidos será comparar la salida del modelo resultante con la encontrada en la literatura [7], de modo que podamos validar que las funciones obtenidas en nuestro modelo siguen las mismas tendencias que las generadas por los autores originales.

Estos resultados que hemos calculado se muestran en la Imagen 21, donde podemos observar el comportamiento de cada uno. Todas las salidas están en el mismo orden de magnitud que los de [7].

En la Imagen 22 podemos comprobar la comparación entre el nivel de glucosa predicho y el real. Comparando la ingesta de hidratos, la ratio de aparición en plasma y el nivel de glucosa en plasma, se observa que existe una clara correlación, si bien hay un ligero retraso entre los dos valores del modelo.

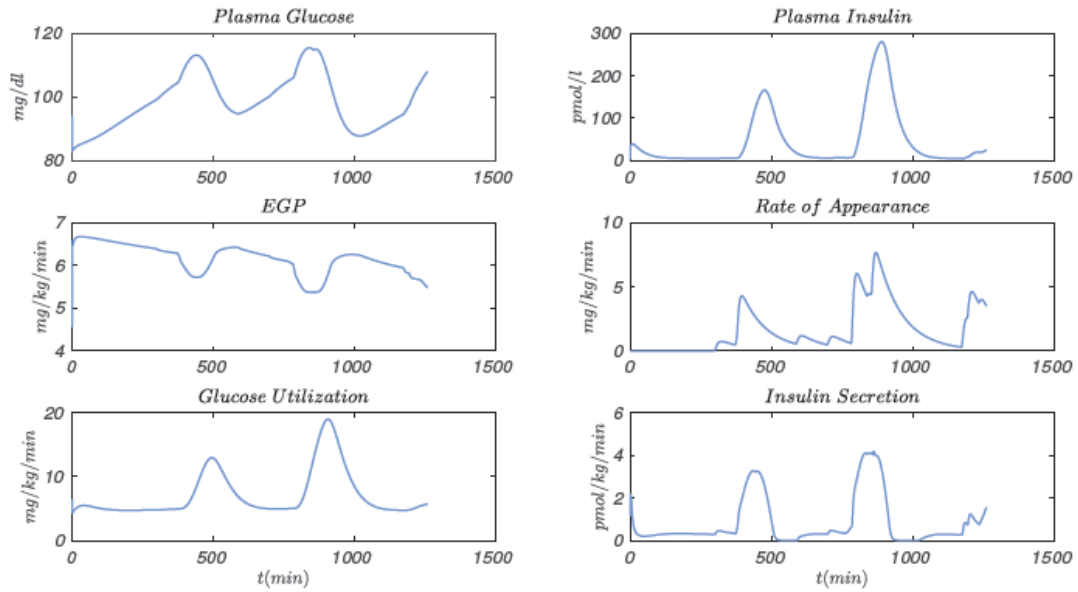


Figura 19 – Salidas obtenidas por los subsistemas del modelo tras procesar los datos del paciente real. Elaboración propia.

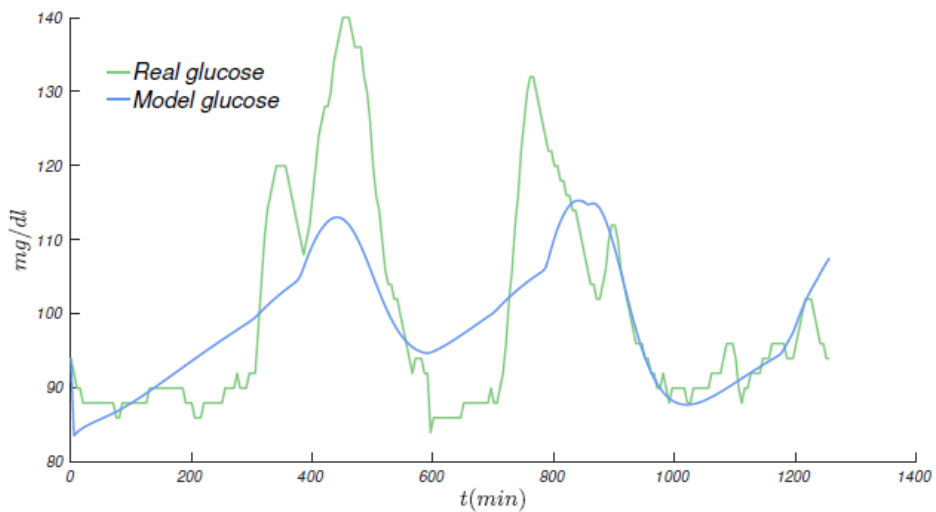


Figura 20 – Comparación entre la glucosa obtenida por el modelo y la medida en el paciente real. Elaboración propia.

Sin embargo, solamente con las funciones dibujadas en las imágenes que hemos presentado, la validez del modelo no se puede garantizar.

Es por ello que entran en juego los diagramas de Clarke [21] y Parkes [22], creados originalmente para cuantificar el error de medidores de glucosa en comparación con los valores reales, obtenidos con métodos más fiables.

Dichos diagramas funcionan comparando la medida real con la de referencia, de modo que la real se dibuja en el eje horizontal, y la medida en el eje vertical. Si no hay diferencia entre ambas, se dibuja un punto en el diagrama donde se cruzan ambas.

Además, recordando lo que hemos mencionado en el apartado correspondiente, los gráficos están divididos en áreas, representando cada un margen de error específico (A, B, C, D y E, de menor a mayor error). Como podemos ver en los siguientes diagramas, las mediciones obtenidas estaban todas dentro de la zona A, lo que implica que el error es menor que el que cometería un medidor de referencia [23].

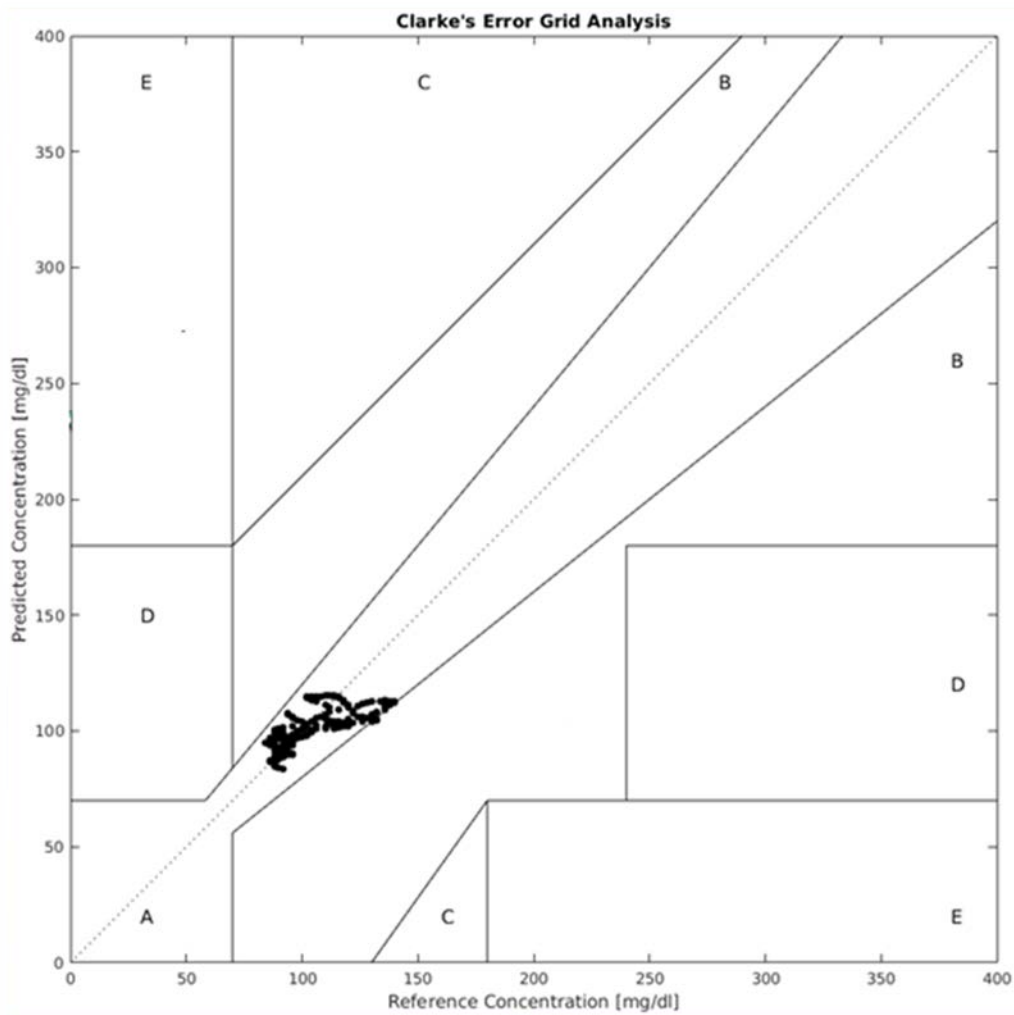


Figura 21 – Diagrama de error de Clarke (“Clarke Error Grid”). Elaboración propia.

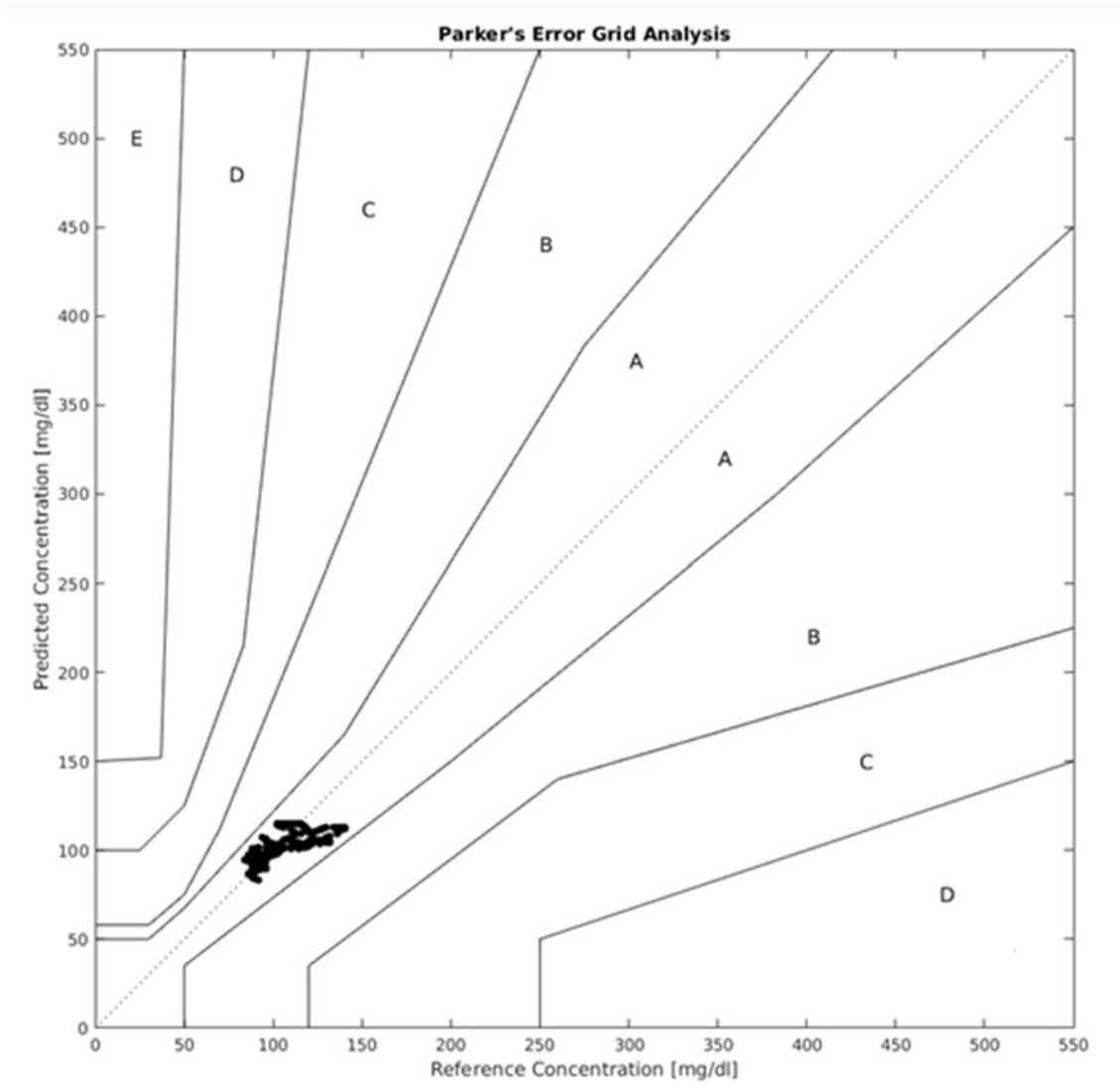


Figura 22 – Diagrama de error de Parkes (“Parkes Error Grid”). Elaboración propia.

Una vez el modelo ha sido validado, se continúa la simulación para cuatro horas más de las que hemos utilizado para calibrarlo, para comprobar que el modelo es estable pasado el periodo de validación. La glucosa ingerida se considera nula durante este periodo.

Tanto la glucosa calculada como la real se dibujan en la siguiente figura. Pasada la línea roja, el gráfico muestra los valores que devuelve la predicción del modelo, pasados 240 minutos.

La predicción muestra la misma tendencia que los datos anteriores, y tras disiparse la glucosa que se estaba procesando en el tracto gastrointestinal, los niveles de glucosa decrecen lentamente. Como se esperaba, no se observa ningún comportamiento irregular.



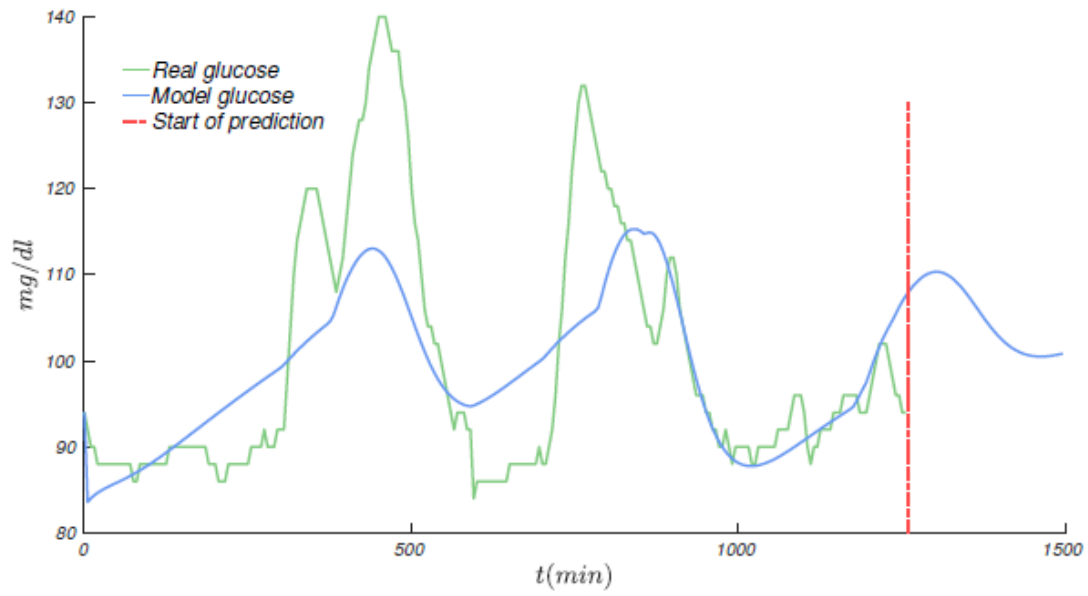


Figura 23 – Predicción de niveles de glucosa para las siguientes cuatro horas. Elaboración propia.

## 6. Mejoras o ampliaciones

---

Este es el capítulo que dedicaremos a las posibles funcionalidades adicionales que tendría sentido añadir a la aplicación, ya sea para mejorar su potencia, su flexibilidad o adaptarla a escenarios de uso reales, donde se necesitará poder procesar el modelo para varios pacientes de forma simultánea, o será conveniente poner al alcance del enfermo el sistema para que pueda utilizarlo por sí mismo.

### 6.1. Múltiples instancias de clientes y servidores

En un escenario de trabajo en que se necesiten calcular varios modelos de forma simultánea, se plantea modificar la arquitectura de funcionamiento, añadiendo un balanceador de carga que pivote entre diferentes instancias de MATLAB y de clientes de nuestro software, repartiendo los datos de cada uno de los pacientes respectivamente, de modo que varios hilos ejecutando el cliente (es decir, “GlucoJ”) puedan emplear el balanceador como “proxy” para interactuar con los diferentes servidores MATLAB en función, por ejemplo, del nombre del paciente o el criterio de distribución pertinente.

Generar esta nueva arquitectura sería un primer paso para minimizar los tiempos de proceso en caso de que la carga de trabajo se volviera masiva, por ejemplo, en un hospital, donde se tuviera que procesar simultáneamente gran cantidad de pacientes en un plazo de tiempo que permita capacidad de respuesta.

Cabe tener en cuenta que para generar una arquitectura como la que se propone habría que modificar el uso que se realiza de la librería “MatlabControl”, de modo que se instancie la clase “RemoteMatlabProxyFactory” para invocar un objeto del tipo “RemoteMatlabProxy”, que será necesario para interactuar con una instancia de MATLAB externa al equipo local donde se ejecuta nuestro software.

Antes de la invocación de dicho proxy remoto se establecerá el protocolo para contactar con el balanceador de carga y generar el hilo que interactuará con cada instancia remota de MATLAB, en función del criterio de distribución elegido (al principio hemos propuesto que fuera el nombre del paciente).

De este modo cada uno de los hilos tendría una instancia del “RemoteMatlabProxy” y por tanto trabajaría con una instancia de MATLAB, como podemos ver en el diagrama.

## 6.2. Base de datos

Otra posible ampliación en la funcionalidad de la aplicación sería la incorporación de un nivel de persistencia asentado en una base de datos, de modo que podamos guardar los registros asociados a cada paciente (sus fichas finales u otra información).

Esta opción sería interesante en combinación con la anterior por retener los resultados de cada enfermo en una base de datos consultable, lo que permitiría disponer de un histórico para cada uno, información que podría realimentar el proceso general.

Un ejemplo de base de datos de sencilla implementación sería una basada en “SQLite”, uno de los Sistemas de Gestión de Bases de Datos más utilizados universalmente. Los principales beneficios de este motor son:

- Totalmente gratuito
- Auto contenida – Toda la librería “SQLite” se encapsula en un solo archivo, sin apenas ninguna dependencia, y funciona en cualquier sistema operativo
- Sin servidor – La mayoría de bases de datos se implementan en un proceso separado, y los programas trabajan con las mismas solicitando escrituras y lecturas. Sin embargo, en este caso el proceso que quiere escribir o leer realiza directamente las operaciones sobre los ficheros, sin tener que tratar con un servidor o proceso externo.
- Cero configuraciones – Relacionada con el punto anterior, al estar embebida en la propia aplicación no necesita de instalación, configuración o depuración de funcionamiento.
- Peso mínimo – La librería pesa menos de 1 MB, lo que facilita su empleo en aplicaciones ligeras.
- Velocidad – Al ser la interacción tan directa entre programa y base de datos, la velocidad de entrada / salida es realmente rápida.

Evidentemente no es el único motor de base de datos que podríamos adaptar para funcionar con nuestro software, pero se propone uno embebido en la aplicación para facilitar en lo posible un hipotético despliegue, dado que el uso que se haría de la base de datos no sería tan intensivo como para necesitar de otros motores más potentes, como PostgreSQL o MariaDB (otras alternativas gratuitas), que necesitan de una infraestructura y configuración más compleja a diferentes niveles.

En función de la arquitectura de la aplicación, la versión más básica de una base de datos podría contener datos de más de un paciente (los valores de los parámetros calculados para cada uno), o un histórico de mediciones y predicciones para un mismo paciente, o combinar ambas opciones, existe gran variedad de posibilidades en función del objetivo final.

### 6.3. Compilar funciones MATLAB en Java

Existe la posibilidad de empaquetar la funcionalidad necesaria de MATLAB, en nuestro caso la función N4SID y la comparación posterior entre el resto de datos reales y los resultantes del modelo.

Con ello podríamos prescindir de la librería que hace las veces de interfaz entre Java y MATLAB, lo que podría aligerar el peso y mejorar el rendimiento de la aplicación, además de facilitar futuros desarrollos aportando autonomía al sistema.

Según los análisis realizados durante el desarrollo, una gran parte del tiempo de ejecución es invertido en la apertura de la interfaz de MATLAB, que se realiza en esta primera iteración si no está abierta previamente.

Teniendo en cuenta que, una vez abierto, el tiempo de transmisión, cálculo y recepción de la información pasa a suponer el grueso del tiempo de ejecución, y que comparativamente para una iteración resulta despreciable, sería una mejora significativa en velocidad y facilidad para futuros desarrollos o cambios en la arquitectura de la aplicación.

### 6.4. Uso autónomo – app móvil

Otra de las propuestas de interés sería un paso posterior al empaquetado de las funciones de MATLAB dentro de la misma aplicación, con la posible combinación de la base de datos, de modo que el paciente pudiera utilizar la aplicación por su cuenta.

De este modo, anotando las ingestas de carbohidratos y las dosis de insulina inyectadas tendríamos unas entradas similares a las que proporcionamos en el CSV a nuestra simulación.

Una vez obtenidos los parámetros adecuados para el enfermo sería posible, con un modelo válido, predecir los niveles de glucosa en sangre con un buen margen de acierto. Si el enfermo es capaz de llevar este control, que a priori parece considerablemente menos invasivo y fácil de mantener que el habitual al que está acostumbrado, podría obviar una gran parte del peso de la afección, acudiendo a su dispositivo móvil o usándolo para poder llevar un seguimiento preciso con su médico especialista.

Además, el hecho de que la aplicación esté desarrollada en Java resulta interesante, puesto que el sistema operativo móvil mayoritario a nivel mundial es Android, que está basado en este mismo lenguaje. Por tanto, sería relativamente sencillo portar el desarrollo al mismo.

Así mismo, el empleo de una base de datos SQLite está ampliamente cubierto en la literatura como elección habitual en aplicaciones para este sistema operativo, lo que facilitaría la implementación de ésta minimizando el esfuerzo de desarrollo.

## **6.5. Entradas auxiliares para el modelo (estrés, ejercicio, etc.)**

Como se comentó al inicio, en la presentación de los subsistemas, resultaría de interés poder modelar el comportamiento y efecto que otras entradas tendrían en el sistema.

Por ejemplo, si tenemos en cuenta que una recomendación habitual para los pacientes de diabetes es que realicen deporte, puesto que mejora la presión arterial y la circulación, y facilita el control del peso, es fácil pensar en algunas fuentes de información muy interesantes, como las calorías consumidas durante el día.

El principal inconveniente para añadir entradas adicionales es que algunas métricas, como el estrés, podrían resultar muy difíciles de modelar o cuantificar, aunque es evidente que tienen un efecto determinante sobre el funcionamiento del metabolismo humano y serían por ello realmente provechosas para mejorar la calidad de las predicciones.

## 7. Conclusiones

---

En los últimos años se han propuesto diferentes modelos [7] [24] para explicar e idealmente predecir la concentración de glucosa en plasma sanguíneo.

El acierto en este modelado permitiría adelantarse a las situaciones de riesgo a que están expuestos los pacientes de diabetes o resistencia a la insulina, además de constituir una herramienta para la conformación de dietas personalizadas para pacientes con estas afecciones, permitiendo un mejor control de las reacciones metabólicas relacionadas.

Hemos constatado fehacientemente la importancia de las posibles investigaciones en este campo, no sólo desde el punto de vista económico, sino desde la perspectiva de la salud y la prevención de una enfermedad con una incidencia y gravedad tan profunda como la diabetes.

En este estudio en particular, que se basa en el modelo presentado por Dalla Man en [7], se pretendía dilucidar si este modelo se podía aplicar, no sólo a pacientes *in silico* como los demostrados en el trabajo original, sino a pacientes sanos reales, de modo que pudiéramos adivinar o predecir sus niveles de glucosa en plasma.

Así pues, el valor diferencial de este trabajo viene dado en gran parte por la calibración de los parámetros de los subsistemas del modelo, adaptándolos o personalizándolos a un paciente particular mediante el proceso descrito en apartados anteriores. De esta manera, se ha podido realizar una predicción a cuatro horas con éxito.

Desde la perspectiva del estudiante, tener la oportunidad de participar en un trabajo de investigación como este ha permitido tener una visión más clara acerca de cómo la informática puede jugar un papel fundamental en combinación con otras disciplinas para obtener hallazgos de considerable magnitud.

Queda para futuros trabajos trasladar la implementación de este modelo a una aplicación completa que permita a los profesionales de la salud trabajar con ella con facilidad. Para ello, se podrían tomar en consideración algunos de los puntos de mejora o ampliación sugeridos en el capítulo destinado a tal fin.

Para concluir, podemos considerar los resultados conseguidos como satisfactorios, sentando así las bases para una nueva línea de investigación en la que se apliquen las conclusiones obtenidas en este trabajo a la clínica diaria.

## 8. Bibliografía

---

- [1] S. Wild, G. Roglic, A. Green, R. Sicree y H. King, «Global Prevalence of Diabetes, Estimates for the year 2000 and projections for 2030,» *Diabetes Care*, American Diabetes Association, 2004.
- [2] F. Soriguer, A. Goday, A. Bosch-Comas, E. Bordiú, A. Calle-Pascual, R. Carmena, R. Casamitjana, L. Castaño, C. Castell, M. Catalá, E. Delgado, J. Franch, S. Gaztambide, J. Girbés, R. Gomis, G. Gutiérrez, A. López-Alba, M. Martínez-Larrad, E. Menéndez, I. Mora-Peces, E. Ortega, G. Pascual-Manich, G. Rojo-Martínez, M. Serrano-Rios, S. Valdés, J. A. Vázquez y J. Vendrell, «Prevalence of diabetes mellitus and impaired glucose regulation in Spain: the Di@bet.es Study,» 2011.
- [3] S. Carpenter, M. Rigaud, M. Barile, T. J. Priest, L. Perez and J. B. Ferguson, "An interlinear transliteration and English translation of portions of The Ebers Papyrus, possibly having to do with diabetes mellitus," Bard College, Nueva York, 1998.
- [4] L. Poretsky, *Principles of diabetes mellitus*, New York: Springer, 2009.
- [5] Nobel Foundation, «The Nobel Prize in Physiology or Medicine,» 1923. [En línea]. Available: [http://nobelprize.org/nobel\\_prizes/medicine/laureates/1923/index.html](http://nobelprize.org/nobel_prizes/medicine/laureates/1923/index.html). [Último acceso: 1 Agosto 2018].
- [6] Nobel Foundation, «The Nobel Prize in Chemistry,» 1958. [En línea]. Available: <https://www.nobelprize.org/prizes/chemistry/1958/sanger/facts/>. [Último acceso: 1 Agosto 2018].
- [7] C. Dalla Man, R. A. Rizza y C. Cobelli, «Meal Simulation Model of the Glucose-Insulin System,» vol. 54, *IEEE Transactions on Biomedical Engineering*, 2007.
- [8] Federación de diabéticos españoles, «Infografía "La Diabetes en España",» 2014. [En línea]. Available: [https://www.fedesp.es/bddocumentos/1/La-diabetes-en-esp%C3%B1a-infografia\\_def.pdf](https://www.fedesp.es/bddocumentos/1/La-diabetes-en-esp%C3%B1a-infografia_def.pdf). [Último acceso: 1 Agosto 2018].
- [9] M. Mata Cases, «Fundación redGDPS,» 2017. [En línea]. Available: <http://www.redgdps.org/tipos-de-insulina>. [Último acceso: 9 Junio 2018].
- [10] K. L. Stanhope, J. M. Schwarz, N. L. Keim, S. C. Griffen, A. A. Bremer, J. L. Graham, B. Hatcher, C. L. Cox, A. Dyachenko, W. Zhang, J. P. McGahan, A. Seibert, R. M. Krauss, S. Chiu, E. J. Schaefer, M. Ai, S. Otokozawa, K. Nakajima, T. Nakano, C. Beysen, M. K. Hellerstein, L. Berglund y P. J. Havel, «Consuming fructose-sweetened, not glucose-sweetened, beverages increases visceral adiposity and lipids and decreases insulin sensitivity in overweight/obese humans,» *JCI - The Journal of Clinical Investigation*, 2009.

- [11] C. Crespo, M. Brosab, A. Soria-Juan, A. Lopez-Alba, N. López-Martínez y B. Soria, «Costes directos de la diabetes mellitus y de sus complicaciones en España (Estudio SECCAID: Spain estimated cost CIBERDEM - CABIMER in Diabetes),» *Diabetología*, n° 29(6), pp. 182-189, 2013.
- [12] C. Dalla Man, M. Camilleri y C. Cobelli, «A System Model of Oral Glucose Absorption: Validation on Gold Standard Data,» *IEEE Transactions on Biomedical Engineering*, vol. 53, n° 12, pp. 2472-2478, 2006.
- [13] C. Dalla Man, D. M. Raimondo, R. A. Rizza y C. Cobelli, «GIM, Simulation Software of Meal Glucose–Insulin Model,» *Journal of Diabetes Science and Technology*, vol. 1, n° 3, pp. 323-330, 2007.
- [14] M. Cescon, F. Ståhl, M. Stemmann, R. Johansson y D. Tilburry, «DIAdvisor™ - Personal glucose predictive diabetes advisor,» 15 Septiembre 2009. [En línea]. Available: <https://cordis.europa.eu/docs/projects/cnect/2/216592/080/reports/001-FinalPublishableSummaryReportfinal20120915.pdf>. [Último acceso: 10 Junio 2018].
- [15] Eclipse Foundation, «Eclipse Photon | The Eclipse Foundation,» Eclipse Foundation, 27 Junio 2018. [En línea]. Available: <http://www.eclipse.org/photon/>. [Último acceso: 2 Septiembre 2018].
- [16] Eclipse Foundation, «Eclipse Wiki,» 2004. [En línea]. Available: [http://wiki.eclipse.org/FAQ\\_Where\\_did\\_Eclipse\\_come\\_from%3F](http://wiki.eclipse.org/FAQ_Where_did_Eclipse_come_from%3F). [Último acceso: 10 Agosto 2018].
- [17] Eclipse Foundation, «Eclipse Public License,» Eclipse Foundation, Febrero 2004. [En línea]. Available: <http://www.eclipse.org/legal/epl-v10.html>. [Último acceso: 2 Septiembre 2018].
- [18] J. W. Eaton y e. al, «Octave,» 1992. [En línea]. Available: <https://www.gnu.org/software/octave/>. [Último acceso: 13 Agosto 2018].
- [19] J. Kaplan, *MatlabControl [Software]*, 2015.
- [20] C. Jacob y N. Khemka, «Particle Swarm Optimization. An Exploratory kit for Evolutionary Optimization,» de *6th International Mathematica Symposium*, Banff, Canadá, 2004.
- [21] W. L. Clarke, D. Cox, L. A. Gonder-Frederick, W. Carter y S. L. Pohl, «Evaluating clinical accuracy of systems for self-monitoring of blood glucose,» *Diabetes Care*, n° 10, pp. 622-628, 1987.
- [22] J. L. Parkes, S. L. Slatin, S. Pardo y B. H. Ginsberg, «A new consensus error grid to evaluate the clinical significance of inaccuracies in the measurement of blood glucose,» *Diabetes Care*, n° 23(8), pp. 1143-1148, 2000.



- [23] M. L. Casas Oñate y D. Montoya Martínez, «¿Son fiables los medidores de glucemia capilar? Are capillary blood glucose meters reliable?,» *Diabetología*, vol. 28, n° 5, 2012.
- [24] A. Bock, G. François y D. Gillet, «A therapy parameter-based model for predicting blood glucose concentrations in patients with type 1 diabetes,» *Computer methods and programs in biomedicine*, vol. 118, pp. 107-123, 2015.
- [25] D. Dabelea y R. F. Hamman, *Epidemiology of type 2 diabetes mellitus*, Philadelphia: Lippincot Williams & Wilkins, 2004.
- [26] Desconocido, *Papiro Ebers, columna 37*, Cerca del 1550 A.C..
- [27] R. J. Marcano Pasquier, «Medicina Preventiva Santa Fe,» 2015. [En línea]. Available:  
[http://www.medicinapreventiva.com.ve/articulos/diabetes\\_mellitus.htm](http://www.medicinapreventiva.com.ve/articulos/diabetes_mellitus.htm).  
[Último acceso: 8 Junio 2018].
- [28] M. Häggström, «Wikipedia Commons,» 19 Marzo 2015. [En línea]. Available:  
[https://commons.wikimedia.org/wiki/File:Main\\_symptoms\\_of\\_diabetes\\_es.svg](https://commons.wikimedia.org/wiki/File:Main_symptoms_of_diabetes_es.svg)  
. [Último acceso: 9 Agosto 2018].
- [29] Á. P. Álvarez, «Que pase el siguiente,» 14 Septiembre 2014. [En línea]. Available:  
<http://qpaseelsiguiente.blogspot.com/2014/09/insulinoterapia-como-se-conserva-y.html>. [Último acceso: 9 Agosto 2018].
- [30] C. Dalla man, A. Caumo y C. Cobelli, «The oral glucose minimal model: Estimation of insulin sensitivirt from a meal test,» *IEEE Transactions on Biomedical Engineering*, vol. 49, n° 5, pp. 419-429, 2002.