



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

---

**FACULTAD DE INGENIERÍA**

**Aplicación de un algoritmo de identificación  
paramétrica para pacientes diabéticos**

**TESIS**

Que para obtener el título de

**Ingeniero Eléctrico Electrónico**

**P R E S E N T A**

Gabriel Villanueva Aragón

**DIRECTORA DE TESIS**

Dra. Ana Gabriela Gallardo Hernández



**Ciudad Universitaria, Cd. Mx., 2023**

*Para mis papás, Carlos Villanueva y Miriam Aragón, que con esfuerzo y cariño me educaron para ser un hombre que siente orgullo de su forma de ser.*

# Agradecimientos

- A los doctores Ana Gabriela Gallardo y Marcos Ángel González, por ser tutores comprensivos, empáticos y honestos.
- A Diana Sánchez, por ser la mejor compañía que podría haber tenido durante mi carrera universitaria.
- A Julio Villanueva, por ser un gran hermano y amigo.
- A María Gómez y Julieta Cruz, porque el cariño siempre es más grande que cualquier distancia que nos separe.
- A Leonardo Morales, Ana Muñoz, y Adán Castillo, por ser amigos que me inspiran y enorgullecen.

# Índice

<b>1. Introducción</b>	<b>7</b>
1.1 Motivación . . . . .	7
1.1.1 ¿Qué es la diabetes? . . . . .	7
1.2 Antecedentes . . . . .	9
1.2.1 Modelo de Sorensen . . . . .	9
1.2.2 Modelo mínimo de Bergman . . . . .	13
1.2.3 Reconstrucción de parámetros . . . . .	14
1.2.4 El uso de aplicaciones en el tratamiento de la diabetes . . . . .	17
1.3 Objetivo . . . . .	17
<b>2. Descripción del problema</b>	<b>19</b>
2.1 Métodos de control aplicados a la diabetes . . . . .	19
2.1.1 Control PID . . . . .	20
2.1.2 Control con lógica difusa . . . . .	20
2.1.3 Control predictivo por modelo . . . . .	21
2.2 Control con modos deslizantes . . . . .	21
2.2.1 Trabajo previo . . . . .	22
<b>3. Diseño de estrategias de identificación</b>	<b>23</b>
3.1 Recolección de datos . . . . .	23
3.1.1 Bomba de insulina . . . . .	23
3.1.2 Monitoreo continuo de glucosa (CGM) . . . . .	24
3.2 Estructura de los datos . . . . .	24
3.3 Elección de datos . . . . .	25
3.4 Implementación de algoritmos de identificación de parámetros en MATLAB .	27
3.5 Implentación de métodos para la reconstrucción de parámetros . . . . .	27
<b>4. Diseño de la interfase digital</b>	<b>35</b>
4.1 App Designer . . . . .	35
4.1.1 Interfaz gráfica de usuario (GUI) . . . . .	35
4.1.2 Interfaz gráfico de usuario en Matlab (App Designer) . . . . .	36
4.2 Diseño . . . . .	40
<b>5. Resultados</b>	<b>45</b>
5.1 Guía para el uso de la interfaz de usuario . . . . .	45
5.2 Coeficiente de correlación de Pearson y medidas de desempeño de Varvel . .	47

---

5.2.1 Experimentos . . . . .	48
5.3 Análisis de resultados . . . . .	51
5.4 Contribuciones . . . . .	57
5.4.1 Presentación de póster en ATTD 2023 . . . . .	57
<b>6. Conclusiones</b>	<b>59</b>
6.1 Trabajo futuro . . . . .	60
<b>Bibliografía</b>	<b>61</b>



# Capítulo 1

## Introducción

### 1.1. Motivación

#### 1.1.1. ¿Qué es la diabetes?

La Asociación Americana de la Diabetes (ADA, por sus siglas en inglés) define este padecimiento como “un grupo de enfermedades metabólicas caracterizada por hiperglucemia causada por defectos en la secreción o en la reacción a la insulina o ambas” [2]. En este sentido, se puede entender a la diabetes como una alteración en el sistema de regulación de la glucosa a través de la insulina.

De acuerdo con el *Atlas de la Diabetes IDF* [11] la glucosa es la principal fuente de energía del cuerpo humano, la cual entra a la sangre a través de la digestión de carbohidratos. Para que las células puedan hacer uso de la glucosa, ésta tiene que atravesar la pared celular, utilizando un grupo de proteínas transportadoras denominadas GLUT; existen diferentes tipos de GLUT dependiendo del tipo de célula. Las células del músculo y grasa utilizan el GLUT 4, las cuales dependen de la insulina, la cual incrementa la expresión de GLUT 4 en la membrana celular, lo cual permite que la glucosa entre en la célula y decrete su nivel en plasma.

La insulina es una hormona producida en los islotes pancreáticos por las células  $\beta$ . Estas células se encargan de liberar la insulina en el torrente sanguíneo, lo cual les permite utilizar la glucosa como fuente de energía.

Los niveles de glucosa en sangre se miden en ayunas y luego de realizar una prueba de tolerancia a la glucosa, en la cual se evalúa su nivel después de ingerir un líquido con glucosa. Según el Centro para el Control y Prevención de Enfermedades de Estados Unidos (CDC), los niveles típicos son:

Resultado	Nivel de glucosa en ayunas	Prueba de tolerancia a la glucosa
<b>Paciente sano</b>	<99 mg/dl	<140 mg/dl
<b>Prediabetes</b>	100 - 125 mg/dl	144 - 199 mg/dl
<b>Diabetes</b>	>126 mg/dl	>200 mg/dl

Las principales afecciones de la diabetes se dividen en Hipoglucemia e Hiperglucemia.

De acuerdo con la ADA [6] la hipoglucemia es causada porque el nivel de glucosa en la sangre es menor que el rango aceptado ( $\leq 70$  mg/dl). Sus síntomas varían desde palidez, temblores, sudoración y pueden llegar a causar pérdida de conciencia o incluso provocar un estado de coma, ya que el sistema nervioso es de los principales consumidores de glucosa en el cuerpo. La hipoglucemia es el principal peligro a corto plazo para los diabéticos.

La hiperglucemia es el exceso de glucosa en la sangre, sus síntomas van desde dolores de cabeza, sed excesiva y/o disminución de peso. Normalmente estos síntomas son los que determinan un diagnóstico de diabetes y pueden causar afecciones a largo plazo.

La diabetes se presenta como tipo 1 y tipo 2 según sus características. La diabetes tipo 1 se caracteriza por la ausencia de producción de insulina, causada por una reacción autoinmune que ataca a las células beta. En el *Atlas de la Diabetes* [11] se explica que la causa de esta reacción no se entiende por completo, aunque probablemente se debe a un componente genético y/o un componente ambiental, como un virus.

Los pacientes con diabetes tipo 1 representan el 10% de los casos de esta enfermedad. Entre éstos se encuentran los casos de infantes y jóvenes adultos, y aunque la diabetes tipo 2 cada vez más se encuentra presente en niños de países con altos índices de obesidad infantil, la diabetes tipo 1 sigue siendo de las enfermedades crónicas más comunes en la infancia [11]. Los pacientes con diabetes tipo 1 necesitan inyecciones diarias de insulina, las cuales, acompañadas de monitoreo de glucosa y educación, permiten tener una vida saludable sin el deterioro y complicaciones constantes generalmente asociadas a la enfermedad.

La diabetes tipo 2 es el tipo de diabetes más común [11] donde 90% de los pacientes de diabetes la padecen. En este tipo de diabetes los niveles de glucosa en el cuerpo se elevan generando hiperglucemia, debido a que las células no responden de manera adecuada a la insulina, que se conoce como Resistencia a la Insulina. Esta resistencia incrementa la necesidad de insulina que el cuerpo tiene que generar, por lo cual esta demanda con el tiempo provoca un desgaste en las células beta y produce una falla en la producción de insulina.

Este tipo de diabetes se presenta principalmente en personas mayores, aunque cada vez se diagnostica más en gente joven. El tratamiento consiste en medicación e inyecciones de insulina, acompañados de ejercicio y una dieta saludable. Los principales medicamentos usados son Metformina, que reduce la resistencia a la insulina, y las Sulfonilureas, que incrementan la producción de insulina producida por el cuerpo. Los componentes principales que causan diabetes tipo 2 son: a) el genético, específicamente si existe un familiar de primer grado con diabetes; y b) el componente étnico, ya que algunas etnias presentan una mayor predisposición a la diabetes tipo 2.

De acuerdo con la Organización Panamericana de la Salud (OPS), la diabetes es una de las principales causas de ceguera, daño renal, ataques cardíacos, derrame cerebral y ampu-



taciones de extremidades inferiores. Considerando que, según la Federación Internacional de Diabetes [11], “En 2021, se estimó que 537 millones de personas tienen diabetes, este número se proyecta a llegar a los 643 millones en 2030 y 783 millones en 2045”. La federación concluye que es necesario realizar reformas médicas para disminuir las afecciones crónicas causadas por la diabetes, para lo cual será necesario poder personalizar el tratamiento de cada paciente con el propósito de que sea más efectivo.

## 1.2. Antecedentes

### 1.2.1. Modelo de Sorensen

Entre el grupo de modelos que describen la homeostasis de la diabetes, el modelo de Sorensen es considerado uno de los más complejos y detallados. Fue desarrollado por Thomas J. Sorensen como tema de tesis para obtener su título de Ingeniero Químico en la Universidad de Berkeley en 1978. El objetivo principal de este modelo es proporcionar una descripción exhaustiva del control de la insulina y la glucosa en el cuerpo humano, abordando su comportamiento y sus efectos en diferentes partes del organismo.

El modelo de Sorensen [12] divide el cuerpo humano en varios compartimentos clave: el cerebro, el corazón y los pulmones; el hígado y el intestino, los riñones y la periferia. A su vez, estos compartimentos se subdividen en tres submodelos principales: el modelo de insulina, el modelo de glucosa y el modelo de glucagón. Cada uno de estos submodelos tiene como objetivo capturar de manera precisa los procesos y las interacciones específicas relacionadas con la regulación de la insulina y la glucosa en el organismo humano.

Por otro lado, el modelo de glucosa se enfoca en la regulación de los niveles de glucosa en sangre; también utiliza ecuaciones diferenciales para describir la dinámica de la glucosa en el cuerpo. Estas ecuaciones tienen en cuenta la producción de glucosa por parte del hígado, la absorción de glucosa en los tejidos y la utilización de glucosa por parte de las células. El modelo de glucosa también considera la acción de la insulina en la captación de glucosa por parte de los tejidos y en la inhibición de la producción hepática de glucosa.

El tercer submodelo es el modelo de glucagón, que se centra en la regulación del glucagón, una hormona que tiene el efecto opuesto a la insulina. El modelo de glucagón describe la producción, liberación y eliminación de esta hormona, así como su influencia en la producción y utilización de glucosa en el cuerpo.

El compartimento del cerebro tiene un papel crucial en la regulación de la homeostasis de la glucosa, ya que es el responsable de detectar y responder a los cambios en los niveles de glucosa en sangre. El cerebro utiliza la glucosa como su principal fuente de energía y, por lo tanto, es sensible a las fluctuaciones en los niveles de glucosa. El modelo de Sorensen tiene en cuenta las interacciones entre el cerebro y otros compartimentos para comprender mejor cómo se regula la glucosa en el organismo.

El compartimento del corazón y los pulmones se incluye en el modelo de Sorensen debido a su participación en el transporte y la distribución de la glucosa y la insulina en el cuerpo. El corazón bombea la sangre que lleva la glucosa y la insulina a través de los vasos sanguíneos, mientras que los pulmones permiten el intercambio de gases y ayudan a mantener la concentración de oxígeno necesaria para el metabolismo adecuado de la glucosa.

El hígado desempeña un papel central en la homeostasis de la glucosa, ya que es responsable de la producción y el almacenamiento de glucosa. El modelo de Sorensen considera las diferentes funciones hepáticas, como la producción de glucosa por glucogenólisis y gluconeogénesis, así como la captación y almacenamiento de glucosa en forma de glucógeno.

Los riñones también son relevantes en el control de la homeostasis de la glucosa, ya que están involucrados en la reabsorción y excreción de glucosa en la orina. El modelo de Sorensen tiene en cuenta la filtración glomerular y la reabsorción tubular de glucosa en los riñones para comprender mejor su contribución al equilibrio de glucosa en el cuerpo.

Finalmente, el compartimento periférico abarca los tejidos y órganos fuera de los compartimentos mencionados, e incluye los músculos, el tejido adiposo y otros tejidos periféricos que son responsables de la captación y utilización de glucosa en respuesta a la insulina.

Las múltiples ecuaciones diferenciales, generalmente no lineales (Ecuación 1.1 a 1.16) y la gran cantidad de parámetros que se utilizan, hacen que utilizar el Modelo de Sorensen como herramienta para el control y la parametrización sea virtualmente imposible. Sin embargo, es indudable que es una gran herramienta para entender la dinámica de la glucosa e insulina en el cuerpo y los efectos en los distintos órganos.

### Submodelo de glucosa

Cerebro:

$$V_{BV}^G \frac{dG_{BV}}{dt} = Q_B^G (G_H - G_{BV}) - \frac{V_{BI}}{T_B} (G_{BV} - G_{BI}) \quad (1.1)$$

$$V_{BI} \frac{dG_{BI}}{dt} = \frac{V_{BI}}{T_B} (G_{BV} - G_{BI}) - r_{BGU} \quad (1.2)$$

Corazón y pulmones:

$$V_H^G \frac{dG_H}{dt} = Q_B^G G_{BV} + Q_L^G G_L + Q_K^G G_K + Q_P^G G_{PV} - Q_H^G G_H - r_{RBCU} \quad (1.3)$$

Intestino:

$$V_J^G \frac{dG_J}{dt} = Q_J^G (G_H - G_J) - r_{JGU} \quad (1.4)$$

Hígado:

$$V_L^G \frac{dG_L}{dt} = Q_A^G G_H + Q_J^G G_J - Q_L^G G_L + r_{HGP} - r_{HGU} \quad (1.5)$$

Riñones:

$$V_K^G \frac{dG_K}{dt} = Q_K^G (G_H - G_K) - r_{KGE} \quad (1.6)$$

Periferia:

$$V_{PV}^G \frac{dG_{PV}}{dt} = Q_P^G (G_H - G_{PV}) - \frac{V_{PI}}{T_P^G} (G_{PV} - G_{PI}) \quad (1.7)$$

$$V_{PI} \frac{dG_{PI}}{dt} = \frac{V_{PI}}{T_P^G} (G_{PV} - G_{PI}) - r_{PGU} \quad (1.8)$$

### Submodelo de insulina

Cerebro:

$$V_B^I \frac{dI_B}{dt} = Q_B^I (I_H - I_B) \quad (1.9)$$

Corazón y pulmones:

$$V_H^I \frac{dI_H}{dt} = Q_B^I I_B + Q_L^I + I_L + Q_K^I + I_K + Q_P^I I_{PV} - Q_H^I I_H \quad (1.10)$$

Intestino:

$$V_J^I \frac{dI_J}{dt} = Q_J^I (I_H - I_J) \quad (1.11)$$

Hígado:

$$V_L^I \frac{dI_L}{dt} = Q_A^I I_H + Q_J^I I_J - Q_L^I I_L + r_{PIR} - r_{LIC} \quad (1.12)$$

Riñones:

$$V_K^I \frac{dI_K}{dt} = Q_K^I (I_H - I_K) - r_{KIC} \quad (1.13)$$

Periferia:

$$V_{PV}^I \frac{dI_{PV}}{dt} = Q_P^I (I_H - I_{PV}) - \frac{V_{PI}}{T_P^I} (I_{PV} - I_{PI}) \quad (1.14)$$

$$V_{PI} \frac{dI_{PI}}{dt} = \frac{V_{PI}}{T_P^I} (I_{PV} - I_{PI}) r_{PIC} \quad (1.15)$$

Submodelo del glucagón

$$V^\Gamma \frac{d\Gamma}{dt} = r_{PGR} - r_{PGC} \tag{1.16}$$

Se puede observar la representación del modelo de Sorensen en forma esquemática en la Figura 1.1, donde se incluyen todos los submodelos. El modelo original no incluye al tracto gastrointestinal, este fue posteriormente añadido.

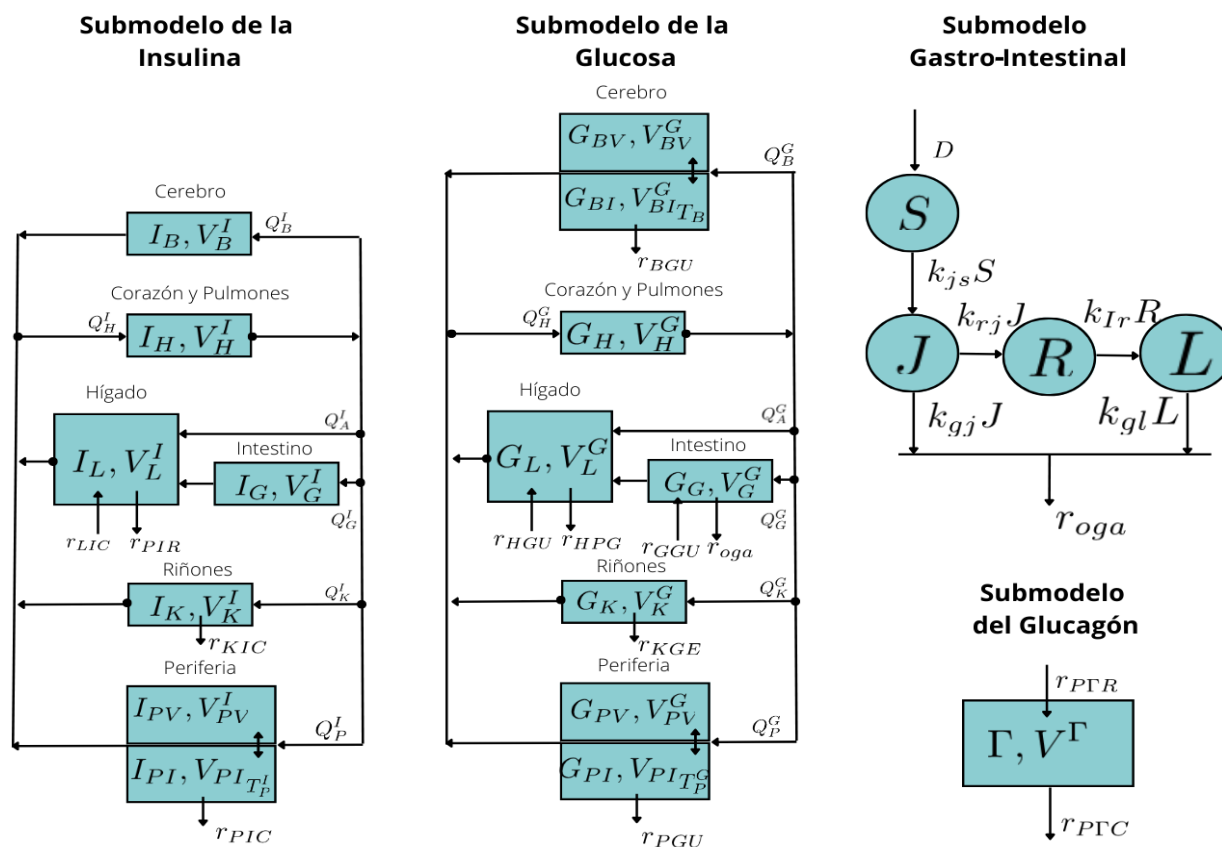


Figura 1.1: Representación esquemática del modelo de Sorensen [12].

### 1.2.2. Modelo mínimo de Bergman

El modelo mínimo de Bergman surge por la necesidad de modelar procesos endocrinos, específicamente la producción de insulina para absorber la glucosa en la sangre. Los modelos anteriores al de Bergman fueron poco aceptados por la comunidad médica pues, generalmente, los endocrinólogos no están familiarizados con el modelado y las ecuaciones diferenciales. Los modelos anteriores sólo eran propuestos por sus creadores sin que fueran probados experimentalmente. Esto motivó la creación de este modelo, el cual está basado por completo en los resultados experimentales.

Las variables clave son la insulina plasmática, la cual es la entrada; la glucosa plasmática, la salida. Estas dos variables fácilmente medibles, simplifican el modelo y evitan que sean necesarias una gran cantidad de variables de estado para poder simular las dinámicas del páncreas y de los procesos biológicos asociados. El modelo consiste en las siguientes ecuaciones.

$$\dot{G} = -p_1[G - G_b] - GX \quad (1.17)$$

$$\dot{X} = -p_2[X] + p_3[I - I_b] \quad (1.18)$$

$$\dot{I} = -n[I - I_b] + \gamma[G - h]t + u(t) \quad (1.19)$$

Como se mencionó anteriormente, las variables principales son la glucosa plasmática ( $\mathbf{G}$ ), la insulina plasmática ( $\mathbf{I}$ ) y una variable auxiliar ( $\mathbf{X}$ ) que permite describir el efecto en la insulina en la disipación de la glucosa. Las dinámicas de estas tres variables se describe respectivamente en las ecuaciones 1.17, 1.18 y 1.19.

En la primera ecuación (1.17) se describe el proceso de absorción de la insulina después de una inyección de glucosa, este proceso busca normalizar el nivel de glucosa al nivel base, denominado glucosa basal ( $\mathbf{G}_b$ ).

El cambio de la glucosa con respecto al tiempo 1.17, está dado por la suma de dos términos lineales: uno independiente de la insulina y otro dependiente de ella. En el término dependiente se utiliza la variable  $\mathbf{X}$ , la cual describe el efecto de la insulina en la absorción de glucosa. El término independiente está multiplicado por el coeficiente  $\mathbf{p}_1$ , que es la tasa de absorción de la glucosa independiente de la insulina. Este proceso se lleva a cabo por las células que no utilizan GLUT 4, como las células de la sangre, los riñones, el hígado, el páncreas y las neuronas, entre otros.

La dinámica del efecto de la insulina para la absorción de la glucosa, descrita en la ecuación 1.18, igualmente está determinada por dos términos. Uno que describe la capacidad de los tejidos para absorber glucosa, el cual es dependiente del nivel de insulina plasmática, superior a la insulina basal  $\mathbf{I}_b$  y que incrementa en una tasa denominada  $\mathbf{p}_3$ . El otro término modela la pérdida que sufren los tejidos en la capacidad para absorber glucosa, lo cual ocurre a una tasa denominada  $\mathbf{p}_2$ . Se puede ver que la capacidad de absorción de la glucosa depende de

$p_2$  y  $p_3$  y así es como con estos dos términos calculamos la sensibilidad a la insulina de un individuo como  $S_I = p_3/p_2$ .

En 1.19, se describe la dinámica de la insulina en plasma, la cual está determinada por tres términos donde dos de ellos son fuentes de insulina, mientras que el tercero,  $\mathbf{u}(\mathbf{t})$ , representa a la entrada de insulina externa, es decir, una modificación experimental. La segunda fuente de insulina es la producida por el páncreas, la cual depende de que la glucosa plasmática supere el nivel de la glucosa basal, por lo tanto, es la generada en respuesta a la ingesta de un alimento e incrementa de forma proporcional con  $\gamma$  veces. La última ecuación modela la degradación de la insulina, la cual, al entrar al torrente sanguíneo gradualmente es eliminada de éste, lo cual mantiene el nivel de insulina en el basal y ocurre a una velocidad  $\mathbf{n}$ .

### 1.2.3. Reconstrucción de parámetros

Las técnicas de localización de parámetros no sólo constituyen un campo vital en la optimización, sino que también juegan un rol crucial en la modelización y simulación de fenómenos complejos. Estas técnicas buscan determinar los valores óptimos de los parámetros de una función previamente definida, con el objetivo de minimizar la diferencia entre los resultados experimentales medidos y los generados por el modelo matemático propuesto.

Para llevar a cabo esta tarea, se emplea una función objetivo, cómo se desempeñan los parámetros en cuestión. Esta función objetivo cuantifica la discrepancia entre los valores reales medidos del fenómeno y los valores predichos por el modelo, brindando así una medida de error que se busca minimizar.

Cabe destacar que la tarea de localización de parámetros es, en esencia, un problema de búsqueda multidimensional, y no siempre es un camino sencillo hacia la solución óptima [13]. Las superficies de error asociadas pueden ser altamente no convexas, presentando múltiples mínimos locales que pueden confundir a los algoritmos de búsqueda.

Por esta razón, los métodos empleados para la localización de parámetros son variados y van más allá de las simples técnicas de búsqueda por gradiente. Entre estos métodos, los algoritmos estocásticos han ganado popularidad, ya que la introducción de aleatoriedad en la búsqueda puede ayudar a escapar de mínimos locales y, en última instancia, conducir a soluciones más cercanas al óptimo global.

### Algoritmo genético (GA)

La selección natural es un proceso fundamental en la evolución. Permite que ciertas características heredables y favorables para la supervivencia sean transmitidas a las nuevas generaciones dentro de una población de individuos. En este proceso, los individuos más aptos son aquellos que transmiten sus características a través de la reproducción.

Un algoritmo genético (GA, por sus siglas en inglés) es un método para resolver problemas de optimización inspirado en el proceso de selección natural. Consiste en generaciones de individuos, los cuales representan una posible solución a un problema. Estos son sometidos a un proceso de selección natural, en el que los individuos producen una nueva generación, la cual es igualmente evaluada.

Siguiendo el modelo de la naturaleza, en el proceso de reproducción se considera la posibilidad de mutaciones, lo cual añade un nivel de variación que permite explorar todas las soluciones posibles y alcanzar un resultado óptimo.

En la figura 1.2 se observa un diagrama de flujo de un algoritmo genético, donde se pueden observar operadores principales para este algoritmo como el entrecruzamiento y la mutación, los cuales son explicados a mayor detalle más adelante en este trabajo.

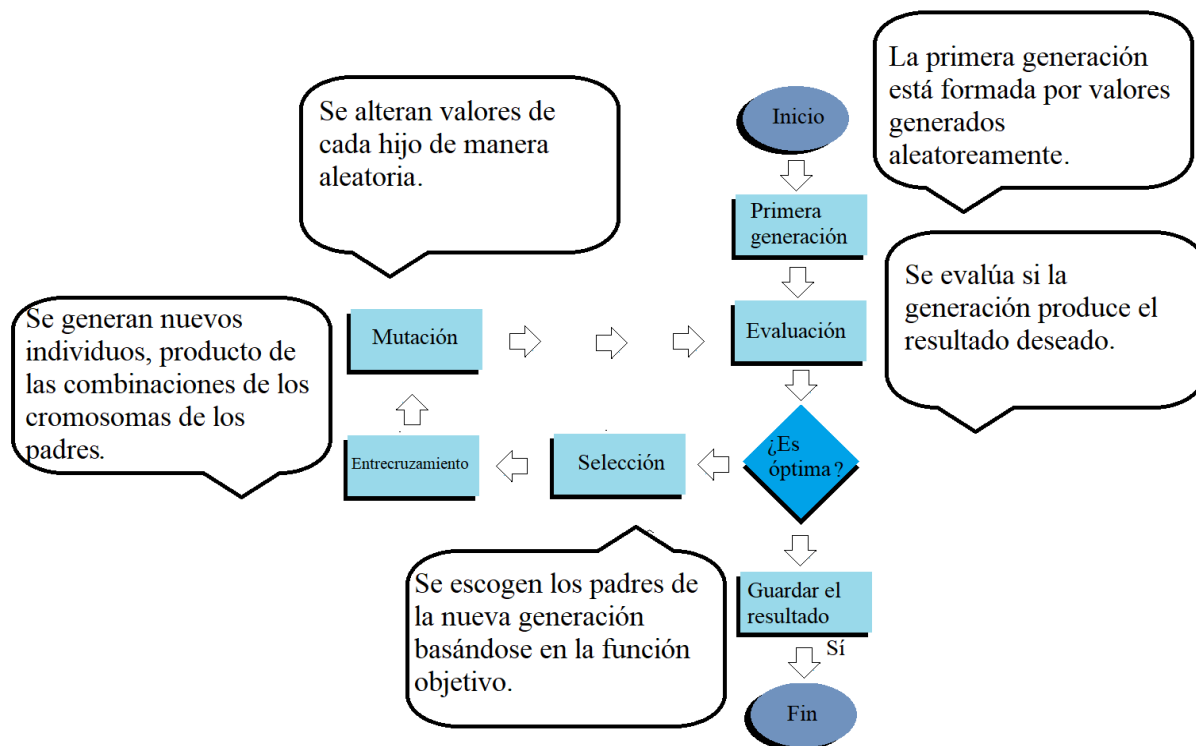


Figura 1.2: Diagrama de flujo de GA.

Los algoritmos genéticos se utilizan con éxito en problemas como: optimización global; diseño y configuración de sistemas complejos; aprendizaje automático y minería de datos; y problemas de secuenciación y planificación. En estos problemas el algoritmo demuestra su capacidad para resolver problemas complejos, con múltiples términos, de alta densidad o no convexos.

## Enjambre de partículas (PSW)

El algoritmo de enjambre de partículas (PSW, por sus siglas en inglés) es un método de optimización inspirado en el vuelo en parvada de aves [13]. Cada individuo en el enjambre se encuentra en un punto del espacio que representa los parámetros a optimizar y genera una solución en la función objetivo. Las partículas se mueven a una velocidad determinada y se aceleran en función de la mejor solución encontrada hasta el momento a nivel personal y la mejor solución global obtenida en todo el enjambre.

Al inicio, todas las partículas se encuentran en posiciones aleatorias y en cada iteración del algoritmo se actualizan tanto la mejor solución personal de cada partícula como la mejor solución global del enjambre. Estas actualizaciones permiten guiar el movimiento de las partículas hacia regiones prometedoras del espacio de búsqueda, buscando así mejorar las soluciones encontradas.

La particularidad de este método de optimización es que depende de una función (ecuación 1.20 y 1.21) con la que se actualizan la velocidad y posición de cada partícula. En éste se considera la velocidad actual y se redirige a un máximo particular y a uno global, lo que permite que se utilice la información adquirida en el pasado y la comunicación colectiva para lograr un mejor resultado.

$$V_i(t + 1) = WV_i(t) + r_1C_1(P_i - x_i(t)) + r_2C_2(G(t) - x_i(t)) \quad (1.20)$$

$$x_i(t + 1) = x_i(t) + V_i(t) \quad (1.21)$$

Podemos observar que la función de la velocidad (1.20) está formada por un término de inercia en el que se multiplica la velocidad por el peso, un componente cognitivo, en el cual se guía la partícula en dirección a un máximo particular y un componente social en el que se guía en dirección del máximo global. Posteriormente se enlistan los parámetros principales.

- $x$  = posición de la partícula.
- $V$  = velocidad de la partícula.
- $W$  = Peso. Éste forma parte del término de Inercia, un peso grande permite una buena optimización global y un peso bajo permite una buena optimización particular. Normalmente, para asegurar un buen resultado, el valor del peso decrece linealmente.
- $C_1$  y  $C_2$  = Parámetros de aprendizaje, generalmente se escogen  $C_2 > C_1 > 0$ .
- $r_1$  y  $r_2$  = Número aleatorio del 0 al 1.
- $P$  = Máximo particular.
- $G$  = Máximo global.

El proceso del algoritmo [13] comienza con la inicialización, en la cual se determinan los parámetros, el tamaño de la población y se asigna una posición y velocidad aleatorios. Posteriormente, se calcula la posición y velocidad siguiente y se evalúa en la función objetivo. Si es posible, se actualiza el mejor valor global y particular y se evalúa si se cumple la condición



de finalización, sea una precisión deseada o un máximo de iteraciones.

#### 1.2.4. El uso de aplicaciones en el tratamiento de la diabetes

Las tecnologías de la información son hoy en día una parte inherente de la vida. Podemos ver sus usos en todas las áreas de nuestra cotidianidad, desde el trabajo hasta nuestro tiempo libre. La presencia de la tecnología se encuentra igualmente en el tratamiento de enfermedades crónicas degenerativas, como la diabetes.

Para entender la presencia de las aplicaciones en el ámbito de la diabetes se utilizó el metanálisis [16], realizado para la revista científica *Advances in Nutrition* que publica la Sociedad Americana de Nutrición. Se analizaron 24 estudios sobre el uso y la efectividad de las aplicaciones móviles con fines médicos (mHealth) en el tratamiento de la diabetes y la obesidad.

El estudio encontró tres usos principales que se le dan a las aplicaciones móviles: aplicaciones móviles que utilizan mensajes de texto para la comunicación entre pacientes y personal de salud; aplicaciones que facilitan la educación y ayudan a los pacientes a manejar su condición; y aplicaciones para la recolección de datos que permiten el monitoreo del estado de salud, las cuales pueden sincronizarse con otros dispositivos de manera inalámbrica.

En este contexto [16], las aplicaciones móviles se utilizan en gran parte para realizar cambios de hábitos. Brindan consejos respecto del control de peso o control de azúcar, establecen recordatorios sobre medicación, dieta o actividad física, proporcionan apoyo social, y administran y muestran datos sobre el consumo de calorías, actividad física, niveles fisiológicos o estudios de laboratorio.

Aunque la mayoría de los estudios demuestran resultados positivos en el uso de aplicaciones móviles, [16] un trabajo de 2017 critica que no se demuestra su efectividad a largo plazo. Además, menciona la dificultad de mantener resultados positivos a largo plazo.

De este metaestudio podemos observar el uso que se le da a las aplicaciones móviles en el tratamiento de la diabetes. Asimismo, podemos ver cómo estas aplicaciones pueden permitir acumular datos del paciente, los cuales pueden ser utilizados por aplicaciones diseñadas para que los trabajadores de la salud mejoren el diagnóstico y el tratamiento

### 1.3. Objetivo

Implementar, mediante la creación de una aplicación interactiva de fácil empleo, algoritmos de identificación paramétrica para el modelo mínimo de Bergman con base en datos experimentales.



# Capítulo 2

## Descripción del problema

El desarrollo de un páncreas artificial es uno de los retos de mayor discusión entre la comunidad de control de sistemas complejos. El avance en los métodos de sensado de glucosa y la creación de bombas de insulina han permitido que la creación de un páncreas artificial de uso comercial sea cada vez más una posibilidad real. El reto pendiente es crear sistemas de control resistentes a las perturbaciones, por ello, este trabajo trata de brindar un avance en el desarrollo de un sistema de control basado en un modelo que sea comercializable y de fácil entendimiento y uso para la comunidad médica.

### 2.1. Métodos de control aplicados a la diabetes

La diabetes mellitus, vista desde la perspectiva del control, se puede entender como una falla en los elementos que permiten la realimentación responsable del mantenimiento del nivel de la glucosa en el cuerpo [4]. En una persona sana, la glucosa (variable a controlar) estimula la producción de insulina en el páncreas (controlador), donde esta interacción se puede considerar un control por realimentación efectivo. Ante la presencia de un tipo de diabetes existe una falla parcial o total del controlador, lo que obliga a introducir un elemento externo al ciclo de control.

Tradicionalmente, la tarea de regular la insulina es efectuada por el mismo paciente, el cual decide la dosis y el momento de la inyección, ayudado de las recomendaciones de un médico. Esto hace que el paciente asuma la responsabilidad de regular un sistema multivariable, donde las fallas y los errores pueden producir consecuencias que dañan la calidad de vida o incluso ponerla en riesgo.

Se han utilizado diferentes métodos de control para poder resolver este problema, dichos métodos no sólo incluyen los utilizados en el control clásico.

### 2.1.1. Control PID

El controlador PID es un controlador clásico, altamente usado por su simplicidad y eficacia. Su funcionamiento está basado en la realimentación y busca mantener una variable de control en un valor deseado. Su representación matemática 2.1 depende del error, el cual se calcula como la salida menos una referencia deseada ( $e = r - y$ ). Como lo indica su nombre, su acción está conformada por tres componentes principales: el término proporcional ( $P$ ), el término integral ( $I$ ) y el término derivativo ( $D$ ).

$$u(t) = u_0 + k_c \left[ e(t) + \frac{1}{\tau_I} \int e(t) dt + \tau_D \frac{de(t)}{dt} \right] \quad (2.1)$$

Los parámetros a ajustar son:  $k_c$ , la ganancia proporcional;  $\tau_I$ , el tiempo integral y  $\tau_D$ , el tiempo derivativo. Existen múltiples métodos para ajustar los parámetros, como el Ziegler–Nichols [3].

El uso de controladores PID para el control de insulina fue propuesto en un intento de simular la acción de las células beta, la mayoría de controladores de este tipo son PD (Proporcional y Derivativo), ya que la acción integral puede producir una sobreadministración, lo que puede causar hipoglucemia. Steil [14] propone que la dinámica de las células beta es bifásica, por lo tanto, asocia su comportamiento con un controlador PID, proponiendo su uso para reemplazar el control que realiza el páncreas. Existe un rechazo hacia esta aproximación [3] ya que la dinámica del sistema es demasiado complicada para poderse resolver con un PID.

### 2.1.2. Control con lógica difusa

La lógica difusa fue creada por Lotfi A. Zadeh a mediados del siglo pasado y ha tenido un gran éxito en el diseño de controladores, especialmente en Japón. Con la evolución en la computación el control difuso se ha transformado en la técnica de control inteligente más popular en la industria.

La lógica difusa nació por la dificultad de representar procesos complejos, como los biológicos. Según su creador, las matemáticas convencionales, las que usan números precisos y definidos, tienen problemas al describir procesos de razonamiento humano [17]. Como definición, la lógica difusa es un sistema lógico que trabaja con clases con fronteras difusas [17], dentro de esta lógica existen los conjuntos difusos en los que los elementos pueden pertenecer a un conjunto de manera parcial.

Para este tipo de control se utiliza la lógica difusa para poder representar el conocimiento humano y la capacidad de hacer inferencias. Para ello, se hace uso de ciertas reglas difusas las cuales usan el conocimiento difuso en el que están basadas para tomar decisiones de control dependientes de las observaciones de la planta. Esto es similar a un panadero subiendo y bajando la perilla del gas donde, dependiendo de la temperatura que recibe, se

regula la cantidad de calor provista al sistema usando unas reglas de inferencia y basándose en mediciones no estrictamente cuantitativas empleando su experiencia.

Se han realizado muchos trabajos que diseñan controladores que aprovechan la lógica difusa para el control de la glucosa [5]. En éstos usualmente se utiliza la lógica difusa para traducir el conocimiento de un especialista médico a un sistema de inferencia de control. Esto permite que se individualice el tratamiento. El desafío de este método de control es que en un sistema tan delicado se deben crear gran cantidad de reglas, las cuales pueden generar conflictos al estorbarse unas a otras. En este sentido, es necesario desarrollar un método para la resolución de estos conflictos en los que se determinen prioridades a ciertas reglas.

### 2.1.3. Control predictivo por modelo

Como su nombre lo indica, el control predictivo por modelo utiliza un modelo para predecir los estados futuros y permite predecir la acción de control necesaria, de esta manera, es necesario que el modelo se calcule continuamente para afrontar las perturbaciones. El primer elemento de control se calcula con la primera medición y el cálculo de un modelo, después se calcula la acción de control para el momento siguiente, posteriormente se mide y se compara el resultado del modelo con la medición y, si es necesario, se recalcula el modelo [1]. La ventaja de este tipo de modelo es que permite tener un control robusto ante perturbaciones.

En el caso del control de la glucosa se utilizará un modelo que describa la dinámica de su presencia en el cuerpo y su comportamiento ante la insulina. Se requiere un sensor de glucosa para poder sensar los datos que corregirán este modelo y una bomba de insulina para la acción de control, se puede buscar alcanzar cierto valor de glucosa en el cuerpo o mantenerlo en un rango saludable; igualmente, se puede diseñar un control que prediga eventos de hipoglucemia y de hiperglucemia.

De acuerdo con Bequette [3] el control predictivo por modelo ha tenido mejor desempeño que el clásico PID, proporcionando menor insulina y glucosa. Aún así esta estrategia de control muestra varios retos, uno de ellos es lo bien que el modelo representa la dinámica individual de un paciente, tomando en cuenta que esta dinámica cambia constantemente. Otro desafío, uno de los principales, es el rechazo de perturbaciones, teniendo que administrar rápidamente insulina ante una ingesta de alimentos altos en carbohidratos.

## 2.2. Control con modos deslizantes

El control por modos deslizantes es un control orientado a *caja negra*, es decir, sólo es necesario conocer el grado relativo del sistema y algunos rangos apropiados. Este tipo de control se usa tanto en sistemas en los que se conoce su estructura como en los que no. Una de sus principales ventajas es que ya que no depende de los parámetros de un sistema y es robusto ante los cambios que puedan sufrir estos.

En Gallardo [8] se puede observar un ejemplo de la aplicación del control de modos deslizantes para controlar la glucosa. En este ejemplo se obtuvo el mismo grado relativo del modelo de Sorensen y del modelo mínimo de Bergman, el cual fue de 3, y se diseñó un controlador por modos deslizantes. El controlador diseñado en este ejemplo dio resultados para ambos modelos, al igual que obtuvo un buen desempeño en pruebas con roedores. Este método de control permitió diseñar un controlador que funciona para cualquier modelo con el que se probó.

### 2.2.1. Trabajo previo

Este trabajo es una continuación al artículo publicado en 2022, *Minimally-Invasive and Efficient Method to Accurately Fit the Bergman Minimal Model to Diabetes Type 2* [7] en el que se utilizaron 20 ratas, a las que se les indujo diabetes y para definir los 5 parámetros del modelo mínimo de Bergman se usó un algoritmo genético.

Para inducir la diabetes en las ratas se utilizó una inyección de estreptozocina, el cual es un medicamento utilizado normalmente para tratar tumores en páncreas y para inducir diabetes mellitus en animales de laboratorio. Una vez inducida la diabetes, se colocó un sensor de glucosa en el área dorsal y utilizando una bomba de insulina se logró medir la insulina y glucosa plasmática. Con estos datos se definió a la función objetivo como la raíz cuadrada del promedio del error obtenido (RMS), es decir, se buscó que la función de glucosa calculada ( $\hat{G}(kT_s)$ ) fuera lo más parecida posible a la función de la glucosa medida ( $G(kT_s)$ ) (ecuación 2.2).

$$RMS = \sqrt{\frac{1}{N} \sum_{k=0}^{N-1} (G(kT_s) - \hat{G}(kT_s))^2} \quad (2.2)$$

Se logró conseguir un error promedio RMS de 15.27 mg/dl, lo cual muestra la cercanía que tuvo el seguimiento del modelo al comportamiento deseado, lo que permitió concluir que el método para determinar los parámetros es efectivo y práctico. De esta manera, se podrían realizar las mediciones en casa ya que sólo es necesario registrar la insulina y glucosa plasmática.

Este proyecto de tesis nace con la finalidad de facilitar el uso de un algoritmo genético para determinar el modelo mínimo de Bergman, luego de que al realizar el método de optimización para escribir el artículo [7] se utilizó un grupo de datos seleccionados priorizando el mejor resultado. Para ello se filtraron los datos, almacenados en variables de Matlab, los que obtuvieron puntos máximos. Se determinó que era necesario un sistema que leyera directamente los archivos generados por la bomba de insulina. Al mismo tiempo se buscó crear una interfaz de usuario para facilitar la obtención de parámetros, con el objetivo de hacer accesible el trabajo desarrollado y que sea útil para las personas que no conozcan el programa.

# Capítulo 3

## Diseño de estrategias de identificación

### 3.1. Recolección de datos

En el experimento original que fundamenta esta tesis, se empleó el *SOFTSensor* de Medtronic para la medición de glucosa. Dentro de este contexto, se llevó a cabo una cuidadosa selección de datos, los cuales fueron obtenidos con el consentimiento del paciente y bajo estrictos protocolos de confiabilidad. Posteriormente, estos datos se depuraron y transformaron en estructuras de fácil lectura para su manejo en Matlab. Para los fines de este proyecto, se optó por leer directamente los datos provenientes del sensor, el cual devuelve un archivo CSV con las fechas previamente definidas para el experimento.

#### 3.1.1. Bomba de insulina

De acuerdo con McAdams y Rizvi [10] una bomba de insulina es un dispositivo para suministrar insulina de manera continua. La bomba consiste en una reserva de insulina de acción rápida y un mecanismo para suministrar esta hormona de forma programada.

La bomba obedece al perfil de glucosa de cada paciente para administrar durante 24 horas insulina basal. Como explican los autores, el rango de insulina basal cambia durante el día, dependiendo del nivel de estrés, fisiología, actividad y otros factores. Es por esto que la mayoría de las bombas tiene la función para programar rangos de insulina basal para ciertas horas o ciertos eventos. Por otra parte, la bomba también administra bolos de insulina para contrarrestar la ingesta de alimentos y corregir niveles altos de glucosa.

Las ventajas de la bomba de insulina son [10]:

- La administración de insulina es más cercana a las necesidades fisiológicas.
- Utiliza sólo insulina de acción rápida, lo que minimiza los picos de glucosa.
- Reduce el riesgo de hospitalización.

- Disminuye el costo del tratamiento.

### 3.1.2. Monitoreo continuo de glucosa (CGM)

El monitoreo continuo de glucosa (CGM, por sus siglas en inglés), de acuerdo con [10], ha sido uno de los mayores avances en el tratamiento de la diabetes. Consiste en el monitoreo constante del nivel de glucosa a través de un sensor, un transmisor y un receptor. El CGM permite tener un monitoreo de manera menos invasiva y con más frecuencia que los métodos tradicionales.

El sensor mide de manera subcutánea la glucosa del fluido intestinal cada 3 a 5 minutos, donde se utiliza una enzima que reacciona a la glucosa y produce una corriente eléctrica. El sensor está conectado a un transmisor, el cual envía los datos de forma inalámbrica a un receptor o una aplicación. Los datos recibidos permiten llevar un historial del nivel de la glucosa y permite integrar alarmas ante ciertos niveles peligrosos.

La integración de CGM con la bomba de insulina permite crear un sistema de lazo cerrado, lo que permite que la administración de insulina sea lo más parecido a la función natural del páncreas. Por ello, aunque el CGM se puede utilizar de manera individual, existe la tendencia de que sea integrado a una bomba. Por ejemplo, el sensor de glucosa *EnLite* de Medtronic, el cual está diseñado específicamente para funcionar junto con la bomba de insulina Medtronic 530G[10].

## 3.2. Estructura de los datos

Todos los datos procesados en este trabajo fueron recopilados utilizando una bomba de insulina de la marca Medtronic, específicamente una bomba MiniMed. Esta bomba está diseñada para el tratamiento de la diabetes, mediante la infusión continua de insulina, índices fijos y variables, así como para determinar los niveles de glucosa del usuario. De esta bomba se recopilaron los datos utilizando un archivo CSV, en el cual se guardaron los registros de la bomba. Los datos útiles para el desempeño de la interfaz de usuario son: la fecha de recolección de datos; la hora, minuto y segundo en que se hizo la medición; la insulina suministrada (basal y bolo); y la glucosa plasmática medida.

Para leer el registro de los datos se utilizó el comando *readtable()*, el cual transforma un documento tipo CSV en un dato tipo *table* en MATLAB. El comando utilizado permite definir los límites de lectura de la tabla, así que se crearon dos tablas, una con las medidas de la glucosa y otra con las de la insulina.

*Table* es un tipo de arreglo de organización tabular, es decir, los datos se distribuyen en filas y columnas. Cada columna puede almacenar sólo elementos del mismo tipo de dato,



aunque pueden existir diferentes tipos de datos en varias columnas en una sola tabla. La tabla generada es similar a una hoja de datos en el sentido en que los datos se organizan en columnas etiquetadas. Se pueden acceder a las propiedades de un dato tipo *table* como tradicionalmente se hace en un lenguaje orientado a objetos, utilizando puntos. Las ventajas de este tipo de datos son: organización estructurada, acceso de datos, etiquetado de columna y manejo de datos faltantes utilizando *NaN*.

Utilizar una tabla nos permite extraer los datos necesarios para las fechas requeridas, es decir, en el rango de fechas seleccionadas por el usuario se pueden extraer las mediciones tomadas en ese periodo de tiempo. La bomba de insulina mide la glucosa cada 5 minutos y la insulina cada vez que se realiza una dosis, lo cual puede ser un tiempo variable. Se crean entonces dos arreglos, uno que guarda la insulina y otro la glucosa, medidas en el rango de tiempo deseado.

Definimos entonces una **muestra** como un periodo de tiempo seleccionado por el usuario, del que se extraen las medidas de glucosa e insulina plasmática. Entonces la muestra es procesada por un algoritmo de identificación de parámetros, del que se obtiene un arreglo de parámetros que luego puede ser evaluado en el modelo mínimo de Bergman y obtener un arreglo con la glucosa e insulina estimados, el cual puede ser comparado con la muestra para conocer el error de estimación. Los **resultados** se guardan en datos tipo *table*, en los que se tiene una estructura formada por: nombre del archivo, fecha de inicio, fecha de fin y los parámetros generados. Así es posible reconstruir el experimento, observar la muestra con el que se generó y ver las estimaciones obtenidas.

### 3.3. Elección de datos

Como caso ejemplo para el uso del interfaz se analizó un grupo de datos recopilados durante 14 días, periodo durante el cual se utilizó un sistema CGM junto con una bomba de insulina y se recopilaron los datos de la insulina basal, los bolos de insulina y la glucosa plasmática.

Los datos recopilados en el ejemplo pueden ser observados en la imagen 3.1, donde se puede observar un comportamiento frenético de la glucosa: dos líneas, una superior y otra inferior denotan los límites, aceptados por la Asociación Americana de la Diabetes [6], para determinar la hiperglucemia e hipoglucemia (200 mg/dl y 70 mg/dl).

Los datos del ejemplo se dividieron en múltiples muestras para analizarlas cada una por separado, empleando los dos métodos de parametrización a cada una. Para elegir estas muestras se tomaron periodos de tiempo donde la glucosa alcanza un máximo local y decrece abruptamente (ejemplo de las muestras 3.2), esto mejora el comportamiento de los métodos de parametrización, ya que garantiza que en el modelo de Bergman la glucosa sea mayor al nivel basal.

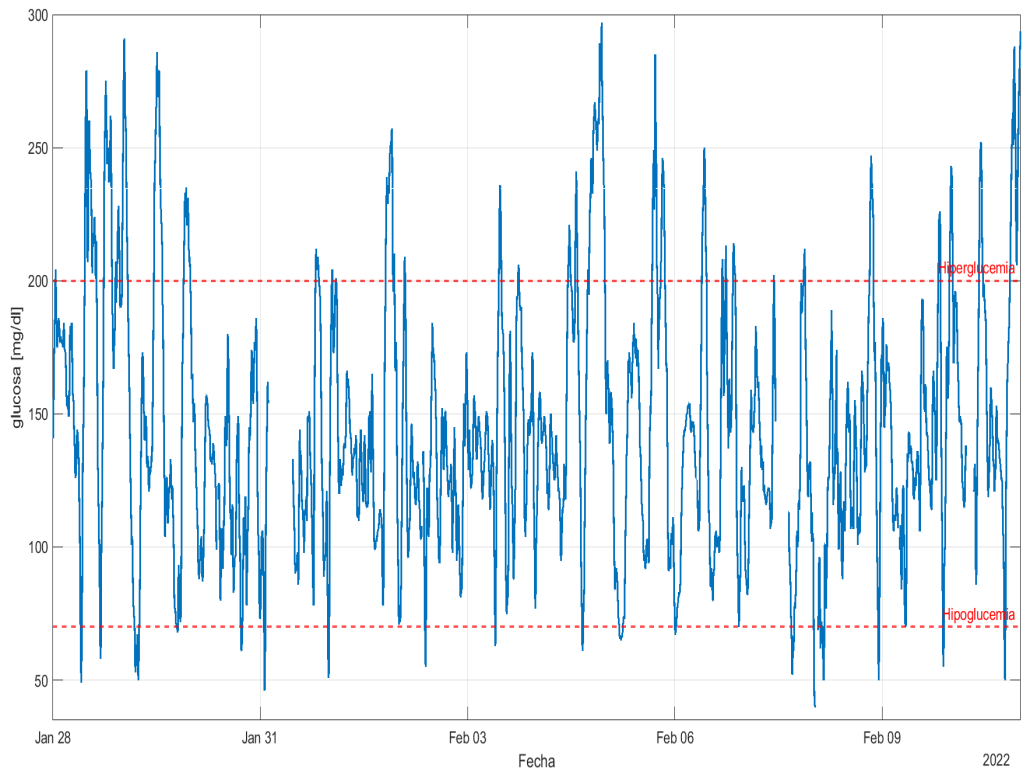


Figura 3.1: Comportamiento completo del caso ejemplo.

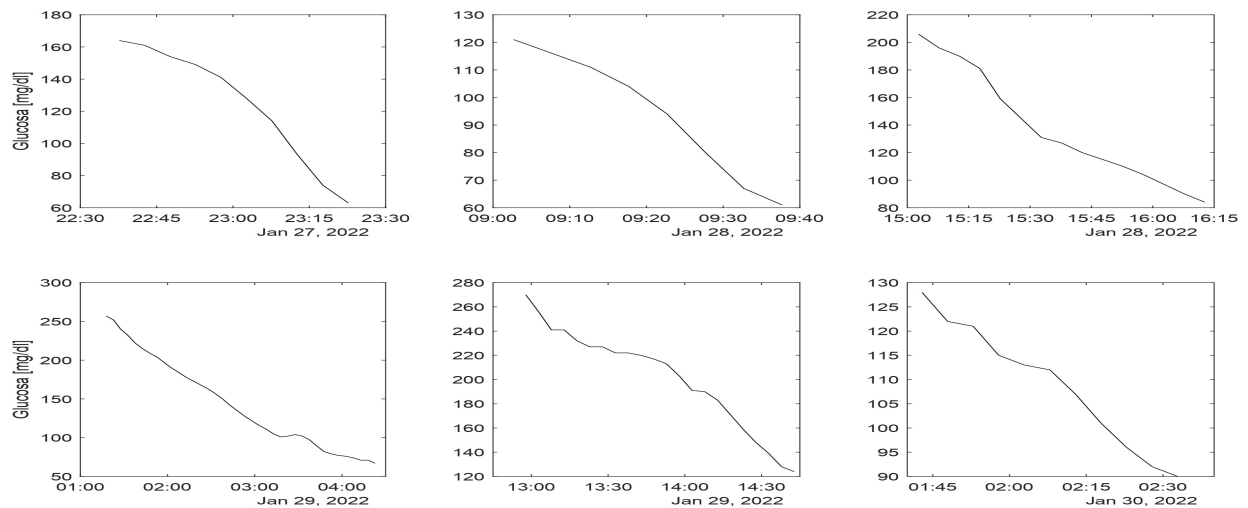


Figura 3.2: Ejemplos de muestras de tiempo utilizadas.

### 3.4. Implementación de algoritmos de identificación de parámetros en MATLAB

Para poder utilizar los algoritmos de identificación de parámetros en Matlab se utilizó el paquete Global Optimization Toolbox. Este paquete incluye funciones como: algoritmo genético, enjambre de partículas o búsqueda global y permite optimizar problemas desde su función objetivo.

### 3.5. Implementación de métodos para la reconstrucción de parámetros

Los dos métodos para reconstruir parámetros buscan definir los valores de  $p_1, p_2, p_3, n$  e  $I_b$  del modelo mínimo de Bergman (Ecuaciones 1.17, 1.18 y 1.19), utilizando la glucosa e insulina plasmática obtenidas experimentalmente. El modelo utilizado es el 3.1, en donde se usan las nuevas variables:  $\hat{G}$  Glucosa plasmática estimada,  $\hat{X}$  Insulina plasmática estimada y los parámetros estimados  $\hat{p}_1, \hat{p}_2, \hat{p}_3, \hat{I}_b$  y  $\hat{n}$ . Las variables ya conocidas quedan la glucosa basal ( $G_b$ ) y la entrada de insulina suministrada por la bomba ( $u$ ).

$$\begin{aligned}\dot{\hat{G}} &= -\hat{p}_1[\hat{G} - G_b] - \hat{G}\hat{X} \\ \dot{\hat{X}} &= -\hat{p}_2[\hat{X}] + \hat{p}_3[\hat{I} - \hat{I}_b] \\ \dot{\hat{I}} &= -\hat{n}[\hat{I} - \hat{I}_b] - \hat{h}t + u(t)\end{aligned}\tag{3.1}$$

Para ambos algoritmos de optimización el objetivo es encontrar la menor diferencia entre la glucosa medida y la obtenida a través del modelo mínimo de Bergman. Para esto se utiliza el error (RMS) dado por la ecuación:

$$RMS = \sqrt{\frac{1}{N} \sum_{k=0}^{N-1} (G(kT_s) - G(\hat{k}T_s))^2}\tag{3.2}$$

#### Algoritmo genético

Como se había mencionado en la Introducción, un algoritmo genético está inspirado en la evolución natural. Para esto se adopta el término de **Gen**, el cual se redefine como una posible solución, es decir, un posible valor que puede adquirir uno de los valores buscados. El grupo de genes forman un **Cromosoma**, el cual consiste en un vector que puede ser evaluado en la función objetivo y el resultado se conoce como **Individuo**; un grupo de individuos forman la **Población**. Los genes se representan de distintas maneras: el fenotipo es el valor que ocupan en el mundo real y el genotipo es un valor binario en el que opera en el espacio de búsqueda. Podemos observar estos elementos en la figura 3.3, donde se pueden identificar

tambien las generaciones del algoritmo genético.

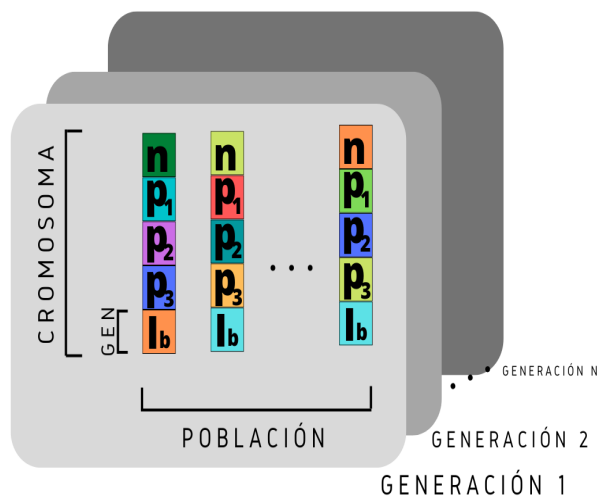


Figura 3.3: Componentes de un algoritmo genético.

Si se observa la figura 1.2 puede verse el proceso que debe de seguir el algoritmo genético. Está formado por tres operadores principales: la selección, la reproducción y la mutación.

El operador de selección forma un grupo con los individuos más aptos, generalmente conservando el mejor resultado y otorgando a un grupo la mayor probabilidad de reproducción. Se preservan un conjunto de cromosomas de élite para la siguiente generación y es necesario elegir el porcentaje de la población que tendrá el privilegio de permanecer en la siguiente generación; esto se conoce como porcentaje de elitismo.

La reproducción es un operador del algoritmo, en el que al igual que en el proceso biológico consiste en la creación de nuevos individuos a partir del intercambio de los genes de dos progenitores. En la figura 3.4 se muestra el modelo básico de reproducción de dos individuos, con cromosomas formados por dos genes cada uno. El procedimiento da lugar a dos nuevos individuos, los cuales contienen una combinación particular de los genes de los padres. Estos nuevos individuos, diferentes entre sí, pueden tener un cromosoma más apto que el de los progenitores.

El entrecruzamiento es un operador del proceso de reproducción en el que se intercambian los cromosomas de los padres. En la figura 3.4 podemos ver este proceso para individuos con sólo dos genes, en el que el modo de entrecruzamiento únicamente puede ser el mostrado en la imagen. En cromosomas formados por más de dos genes existen distintas formas en las que se puede realizar el entrecruzamiento, en la figura 3.5 se pueden observar dos métodos de entrecruzamiento en que dos padres generan un nuevo individuo de una nueva generación.

El primer método utilizado en el ejemplo es el de “por punto dado”, en éste los genes de un padre 1 (color amarillo) se transmiten hasta un punto dado, desde el cual los siguientes

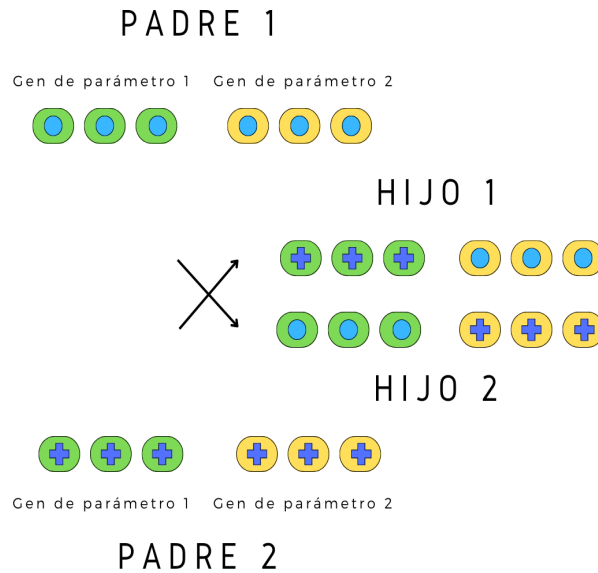


Figura 3.4: Reproducción de dos individuos padres para generar dos hijos.

genes provienen del padre 2 (color verde). Se puede observar un vector que indica este punto formado con unos y ceros, estos tienen una separación definida, normalmente este punto se define de manera aleatoria.

En el otro método, el “entrecruzamiento aleatorio”, los unos y ceros del vector que determina la herencia son formados aleatoriamente, lo cual produce un cromosoma formado por genes de los dos padres, acomodados de forma aleatoria.

La mutación es un proceso que en la biología ocurre cuando hay una variación en el copiado de los genes cuando éstos se duplican, este proceso es el que permite la evolución ya que genera nuevos genes que se traducen en nuevas características, las cuales pueden volver a un individuo más o menos capaz. En el algoritmo genético se utiliza esta misma herramienta, se cambia el valor de un gen aleatorio, esta alteración permite realizar un recorrido en el espacio de búsqueda y evita quedarse en un máximo local.

Para resolver problemas utilizando un algoritmo genético en MATLAB se utiliza la función  $x=ga(fun, nvars)$ . Sus parámetros son:  $fun$ , la función objetivo, es decir, la función a la que el algoritmo le encontrará un mínimo ( $x$ ), y  $nvars$ , el numero de variables de la función  $fun$ .

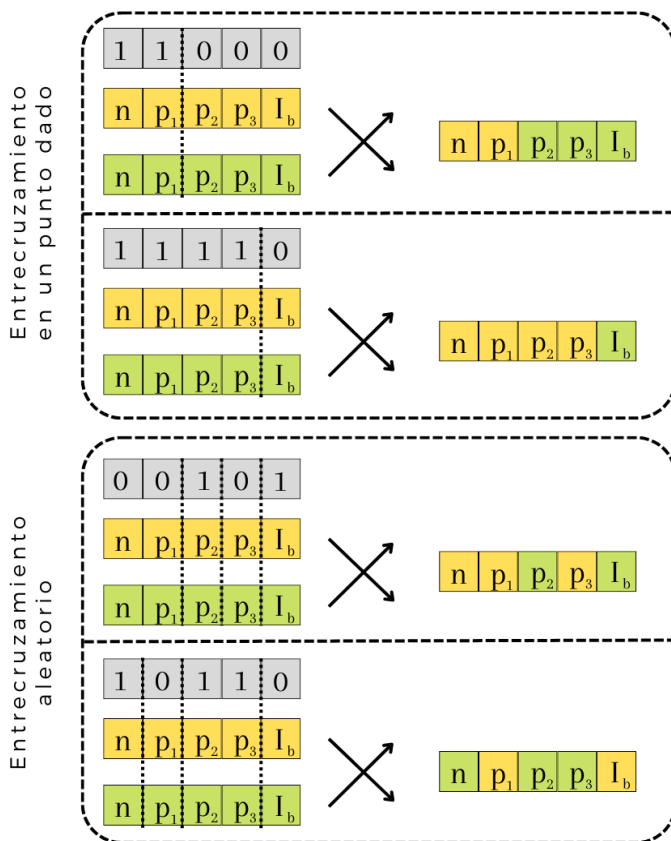


Figura 3.5: Entrecruzamiento en un punto dado y entrecruzamiento aleatorio.

### Enjambre de partículas

Como se mencionó en la Introducción, el método por enjambre de partículas es un método colaborativo, en el cual se utiliza la información de un colectivo de individuos que individualmente buscan un objetivo deseado. Los individuos son una partícula que vive en el espacio de soluciones, es decir, un vector formado por las variables que se buscan encontrar que puede moverse dentro de unos límites donde se encuentra el valor objetivo. Cada individuo puede ser evaluado en la función objetivo y se guarda el mejor resultado particular y global.

En la figura 3.6 se observa la dinámica de una partícula, la cual tiene una posición y una velocidad en el tiempo  $t$ . La partícula se acelerará buscando acercarse tanto a su máximo particular ( $\vec{p}_i(t)$ ) como, al global ( $\vec{g}(t)$ ). La velocidad de la partícula se calcula por la velocidad actual más la atracción en dirección al máximo histórico individual y al máximo global. La posición de la partícula se calcula modificando la posición actual más la velocidad calculada. Las ecuaciones de velocidad y posición se pueden observar en la introducción (1.21 y 1.20).

En la ecuación 1.20 se pueden observar los elementos  $r_1$  y  $r_2$ , los cuales toman un valor aleatorio entre cero y uno, éstos permiten que la partícula afectada por la velocidad explore un punto aleatorio en un área definida, lo cual maximiza las posibilidades de encontrar un

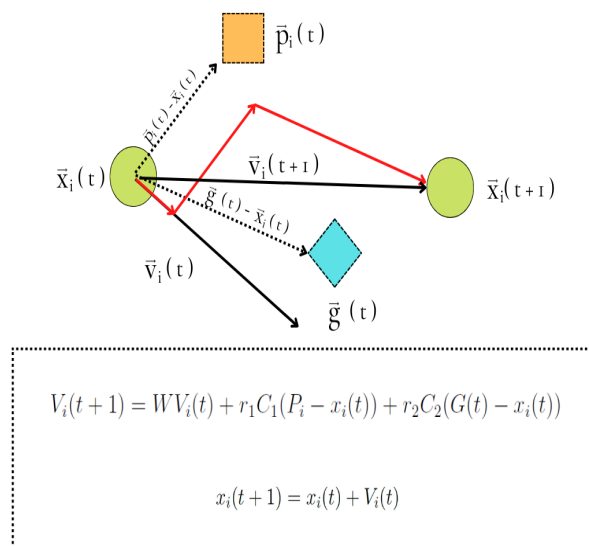


Figura 3.6: Comportamiento de una partícula en PSO.

máximo local. El parámetro relacionado con la inercia,  $W$ , permite que se explore el espacio de búsqueda. Usualmente se busca que  $W$  adopte un valor alto al inicio de las iteraciones y que mientras éstas ocurran decrezca su valor; esto asegura que exista una buena exploración en un inicio y al final los puntos puedan coincidir en el máximo buscado.

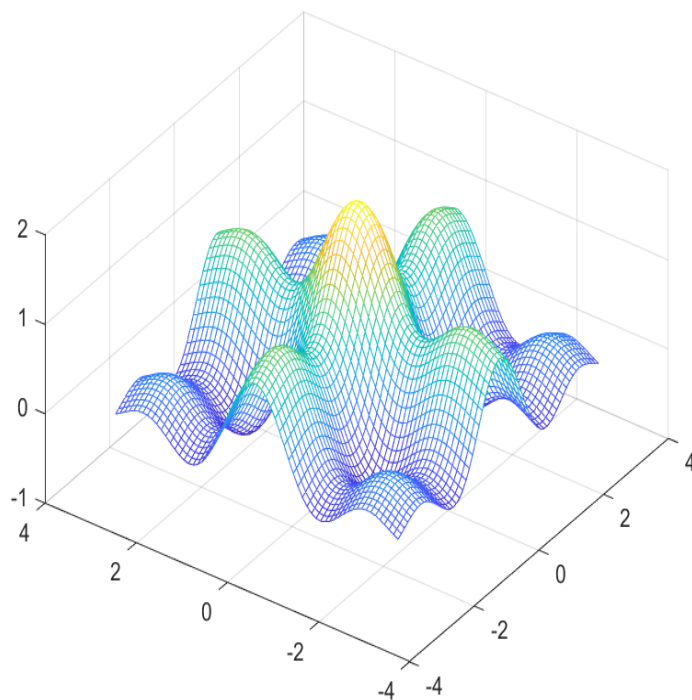


Figura 3.7: Superficie formada por la función 3.3.

$$f(x_1, x_2) = \frac{\sin(\pi x_1)}{\pi x_1} + \frac{\sin(\pi x_2)}{\pi x_2} \quad (3.3)$$

Para ejemplificar el funcionamiento de un enjambre de partículas utilizamos el algoritmo para optimizar la función 3.3 definida entre  $-\pi$  y  $\pi$ . Se utilizó, como se puede observar en la figura 3.7, ya que la función crea una superficie donde el máximo global está rodeado de múltiples máximos locales. Esto permite ver la eficiencia del algoritmo para encontrar el máximo global y evitar los máximos locales.

En la figura 3.8 se observa cómo cambia de posición una población de partículas. Estas posiciones al ser evaluadas en la función objetivo se encuentran en el mapeo observado anteriormente. En el instante  $t = 0$ , se genera un grupo de partículas con posiciones aleatorias, podemos ver que en el tiempo 0 las partículas se distribuyen por el plano; igualmente podemos observar a la partícula con mayor costo, la cual fijará el máximo global. En las imágenes posteriores se observa el proceso en el que las partículas se congregan alrededor del punto máximo. Para este problema fue necesario usar desde el inicio un valor bajo de  $W$  para evitar que las partículas cayeran en un máximo local.

El algoritmo de enjambre de partículas (PSO) tiene varias ventajas que lo hacen atractivo: simplicidad de implementación, pocos parámetros y operadores, capacidad de búsqueda global mediante múltiples partículas y alta aplicabilidad a diferentes problemas con fácil adaptabilidad. Sin embargo, también es importante tener en cuenta sus desventajas. El PSO puede ser sensible a la configuración de parámetros, lo que puede afectar su convergencia. En algunos problemas, puede tener una convergencia lenta. Además, el algoritmo es sensible a la inicialización aleatoria y puede caer en máximos locales, como ocurre con cualquier algoritmo de optimización.



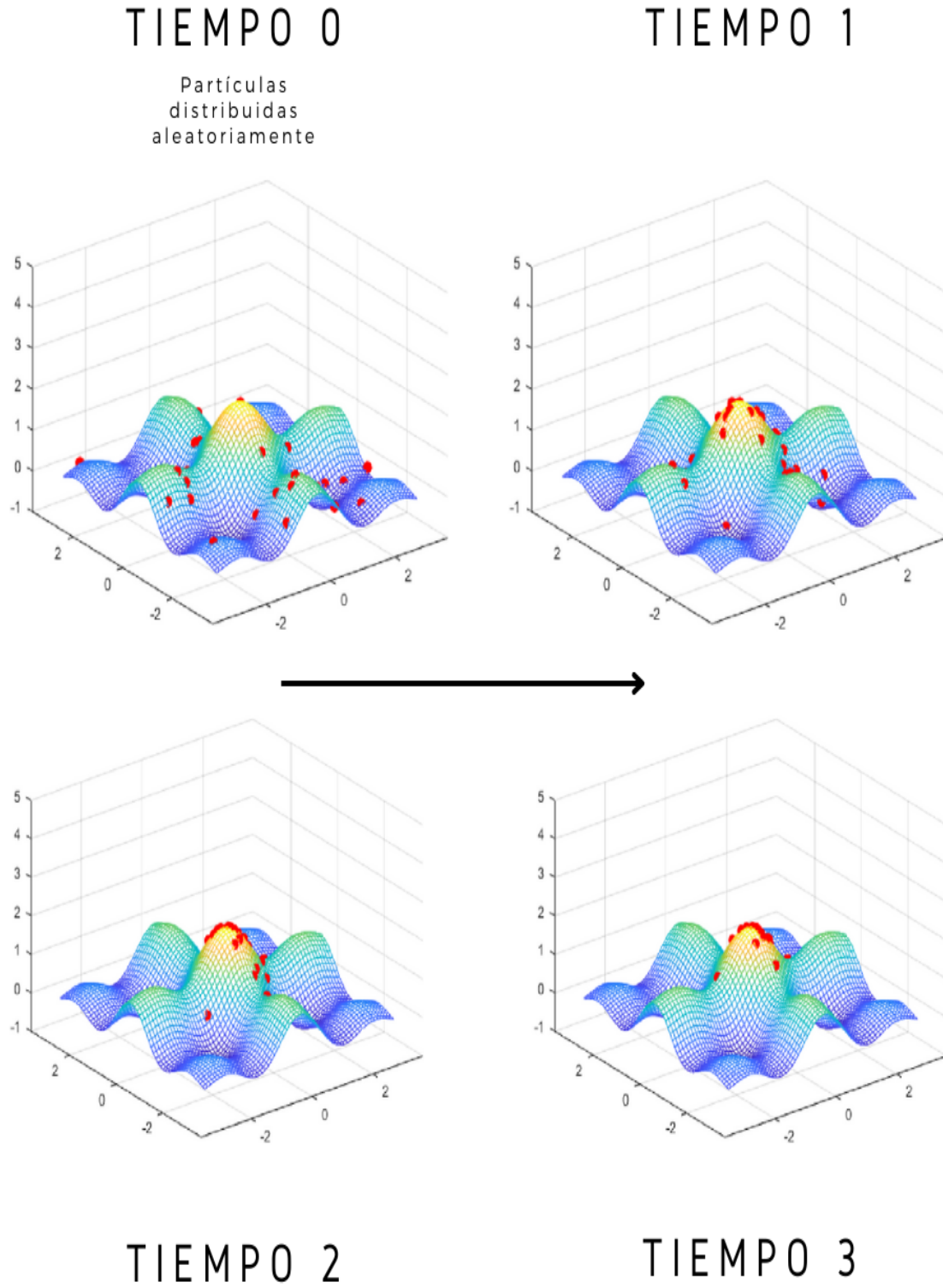


Figura 3.8: Comportamiento de las partículas en PSO.



# Capítulo 4

## Diseño de la interfase digital

### 4.1. App Designer

#### 4.1.1. Interfaz gráfica de usuario (GUI)

Una interfaz de usuario es una herramienta que facilita la comunicación entre el usuario y la máquina. Su función principal es permitir que las personas que no están familiarizadas con un programa puedan utilizarlo, intercambiar información, dar instrucciones y enviar comandos a una tecnología subyacente.

Antes de la existencia de interfaces de usuario era necesario saber programar y conocer el programa. Hoy en día una interfaz de usuario es el método con el que la mayoría de personas hacemos uso de un programa o una tecnología. En específico, las interfaces de usuario gráficas (GUI) utilizan elementos gráficos como ventanas, íconos, menús, botones y más. Los elementos gráficos permiten que se garantice el buen uso del programa, al igual que se restringe el uso según las especificaciones del desarrollador, lo cual hace más seguro el programa.

Usualmente los elementos de una GUI se activan al ser seleccionados con el cursor, esto se hace utilizando programación enfocada en eventos, para lo cual se construye el programa con bloques de código, los cuales serán activados al ocurrir cierto evento. Por ejemplo, la selección de un botón que precede a que la GUI mande un mensaje al usuario.

Una GUI no tiene una lista limitada de componentes, pues varían dependiendo del sistema operativo, la plataforma de desarrollo, el enfoque de diseño, el gusto del programador y las necesidades del usuario. A continuación, presentamos una lista de los elementos más utilizados en una GUI:

- *Ventana:* una pantalla contiene un grupo de elementos. Se puede dividir la interfaz de usuario en distintas pantallas para poder organizar el flujo de información y no sobrecargarla de elementos.
- *Menú:* un menú permite mostrar las secciones en las que está dividido el programa y

navegar por ellas.

- *Cuadro de diálogo*: espacio en el que despliega información en forma de texto.
- *Botón*: herramienta que permite que el usuario interactúe utilizando el cursor para elegir determinada opción al presionarlo.
- *Cuadro de texto*: es un espacio donde el usuario puede insertar información en forma de texto.

Estos son los principales elementos que se pueden utilizar en una GUI. Es importante mencionar también una serie de principios, que funcionan como bases para el diseño de una buena interfaz. Posteriormente veremos los principios recomendados en el artículo “Graphical User Interfaces”, por Wendy L Martinez[9].

1. *Enfocarse en el usuario, no en la tecnología*

Según la autora, éste es el principio fundamental para el diseño de una interfaz, ya que determina todos los demás elementos. En este sentido, es indispensable que el desarrollador tenga siempre en mente al usuario, debe de hacer un ejercicio constante en el que ignora su conocimiento del programa para poder facilitar su uso a alguien que puede ni siquiera pertenecer a su área de trabajo.

2. *Considerar primero la funcionalidad, luego la presentación*

La autora expone que aunque el desarrollador esté tentado a enfocarse en la parte estética de la GUI, es importante que la funcionalidad sea el principal propósito. La dimensión estética del diseño es importante, en efecto, pero si se sobrepone a la funcionalidad entonces aumentan las probabilidades de que existan errores o el usuario tenga una mala experiencia.

3. *Conformarse con la visión del usuario, no hacerlo demasiado complicado*

Al diseñar una GUI el desarrollador debe de priorizar una experiencia de usuario sencilla, es decir, evitar componentes innecesarios o distracciones, sin llegar al extremo de crear una experiencia austera.

4. *Promueve el aprendizaje y entrega información, no sólo datos*

Este principio se explica por sí sólo: la GUI no es únicamente un despliegue de datos, se debe de buscar aprovechar el diseño y crear herramientas para que el usuario obtenga fácilmente la información que requiere.

### 4.1.2. Interfaz gráfico de usuario en Matlab (App Designer)

MATLAB es un *software* creado por MathWorks, creado en 1970 como una calculadora de matrices interactiva escrita en Fortran para facilitar las operaciones matriciales con este len-

guaje. Puede definirse como una calculadora con su propio lenguaje de programación. Este lenguaje tiene una estructura simple, la cual permite a los usuarios enfocarse en el trabajo en lugar de la estructura o la sintaxis del programa. Por la sencillez del lenguaje de programación y la creación de bibliotecas especializadas en problemas del mundo real, MATLAB se ha convertido en una herramienta fundamental para todo tipo de ingenierías.

MATLAB facilita el análisis de ciertos problemas matemáticos con una alta capacidad gráfica, lo que permite observar fácilmente los valores de las variables y sus características, al igual que permite un análisis profundo de las gráficas generadas con los datos, así como para extraer valores de las mismas. Otras herramientas que forman parte de capacidad gráfica de MATLAB, y que permiten llevar el desarrollo a usuarios sin conocimientos de programación, son las tres herramientas para diseñar GUI:

1. **Código:** se puede generar una GUI desde código con funciones, se generan las figuras y los componentes. Esta es la forma más antigua para generar una GUI, pues permite el control de cada pequeña parte del desarrollo. La desventaja de este método es que es un proceso lento y poco eficiente.
2. **GUIDE:** es una herramienta para desarrollar la interfaz de usuario, comparada con el desarrollo en código facilita la creación de las interfaces. En GUIDE se puede usar una serie de componentes limitados y permite la creación de nuevos componentes que pueden ser programados, que junto con el código de funcionamiento crean una ventana de código y la parte gráfica se forma en una ventana aparte.
3. **App Designer:** este es el método más moderno para la creación de GUI en MATLAB. Publicado en la versión R2016a, se ha convertido en la principal herramienta para la creación de aplicaciones.

App Designer permite crear aplicaciones con un aspecto moderno; además, para cada componente agregado se genera un objeto en el código, al que pueden ser asignados eventos. El *software* facilita la programación brindando menús que muestran los componentes ya agregados y permite acceder a los eventos asociados a los mismos.

Otra novedad que brinda este método es que permite exportar la aplicación en forma de un ejecutable, el cual instala una aplicación que se puede utilizar sin tener que ser ejecutada a través de MATLAB. Esta herramienta puede usarse para desarrollar una aplicación web. Esta es una gran ventaja, ya que se pueden crear aplicaciones para usuarios sin que necesariamente tengan que tener MATLAB instalado.

Para el desarrollo de este trabajo era necesario construir una interfaz de usuario la cual fuera fácil de entender y, al mismo tiempo, no tuviera limitantes en su funcionamiento. En consecuencia, se decidió trabajar con App Designer por sus características. Esta plataforma permitió utilizar las herramientas de MATLAB e implementarlas fácilmente a un diseño.

Como se mencionó anteriormente, App Designer funciona con elementos colocados en un lienzo digital (véase figura 4.1), el cual muestra cómo se verá la aplicación final. Para crear el diseño de la GUI se arrastran los elementos de un menú. Cada componente arrastrado funciona como un objeto, del cual se pueden cambiar las características a través de un menú denominado “Component Browser”, en donde se pueden especificar características físicas y/o funcionales. La ventana de diseño permite que el diseñador se concentre en el el aspecto visual de la GUI.

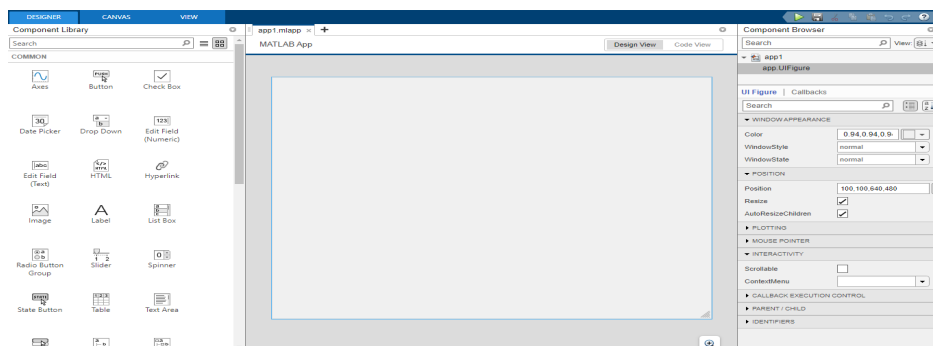


Figura 4.1: Ventana de diseño, App Designer.

Es decir, se diseña en la ventana de programación (véase la figura 4.2). Cada elemento arrastrado al espacio de trabajo en la ventana de diseño, que automáticamente será creado con código, y entonces el desarrollador tendrá solamente que asignar las llamadas, es decir, el programa que se ejecuta cuando el objeto es seleccionado en la GUI. En esta ventana se crean funciones, las cuales pueden ser llamadas en diferentes partes del código. Es posible crear variables globales, las cuales igualmente pueden ser modificadas y llamadas alrededor del código.

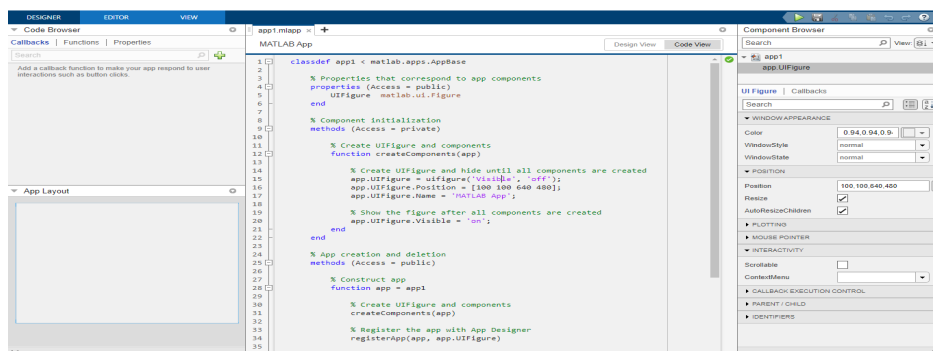



Figura 4.2: Ventana de programación, App Designer.

En la próxima sección se detalla el diseño de la interfaz digital creada para este trabajo. En la tabla (4.1) se pueden observar los elementos utilizados en el diseño y ver cuál es el uso de cada uno.

Ícono	Nombre	Descripción
 Axes	Axes	Permite crear una gráfica dentro del espacio de trabajo, se utilizan las mismas funciones que en un plot para modificar sus parámetros.
 Button	Button	Coloca un botón en el espacio de trabajo, al que se le asocia un evento, el cual se realiza cuando hacemos clic.
 Date Picker	Date-Picker	Elemento desde el cual podemos seleccionar una fecha, la cual queda guardada como un valor tipo “datetime”, se pueden modificar sus parámetros para limitar las fechas de las que se puede elegir.
 Switch	Switch	El elemento “Switch” es un elemento que demuestra un estado lógico, este puede cambiar con la interacción del usuario.
 Edit Field (Text)	Edit Field (Text)	Este es un espacio donde una variable tipo <i>string</i> puede ser o registrada por el usuario o mostrada por el programa.
 Edit Field (Numeric)	Edit Field (Numeric)	Este es un espacio donde una variable numérica puede ser o registrada por el usuario o mostrada por el programa.
 Panel	Panel	Un panel es un espacio de trabajo que puede ser colocado para ubicar distintos elementos, se puede utilizar para hacer visible o invisible secciones de la UI.
 Table	Table	Permite insertar una tabla en el espacio de trabajo; al alterar las propiedades de la tabla se puede especificar los nombres de las columnas y los valores.


	<p>List box</p>	<p>Es un espacio en donde se pueden colocar datos en forma de lista, los elementos de la lista pueden ser seleccionados por el usuario y se puede programar un evento para cuando son seleccionados.</p>
---	-----------------	--

Tabla 4.1: Elementos usados en el diseño de la UI.

## 4.2. Diseño

Para el diseño de la GUI se buscó cumplir con la estructura de datos explicada previamente. Se establecieron los siguientes objetivos basados en el flujo de datos definido (véase figura 4.3):

1. Diseñar un método para el registro de los datos del paciente y la lectura de los mismos.
2. Crear una interfaz para la selección de muestras.
3. Facilitar la identificación de parámetros a través de los dos métodos previamente mencionados.
4. Desarrollar un sistema gráfico para comparar y evaluar los resultados.
5. Por último, agregar un sistema de guardado y reproducción de resultados.

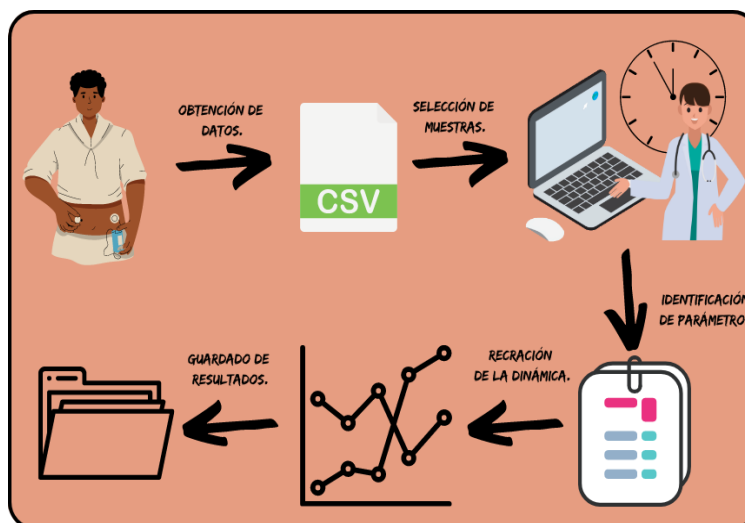


Figura 4.3: Flujo de datos necesario en la interfaz de usuario.



Se planteó un usuario objetivo, el cual es un modelo de persona para el que está diseñada la GUI. Se espera que este usuario sea un profesional de la salud, por lo tanto, no es necesario que comprenda los algoritmos utilizados. Asimismo, se evitarán menús extensos y se priorizará el uso intuitivo para que funcione para la mayor cantidad de usuarios.

Se decidió utilizar un diseño compuesto por tres partes (figura 4.4). La parte superior izquierda, donde se recopilan los siguientes datos necesarios para los cálculos: se selecciona el documento a analizar donde están guardados las mediciones; se define el rango de tiempo a analizar; y se elige el método de identificación de parámetros. Contiene dos botones, uno para tener una vista previa de la glucosa en el intervalo de tiempo seleccionado y otro para calcular los parámetros.

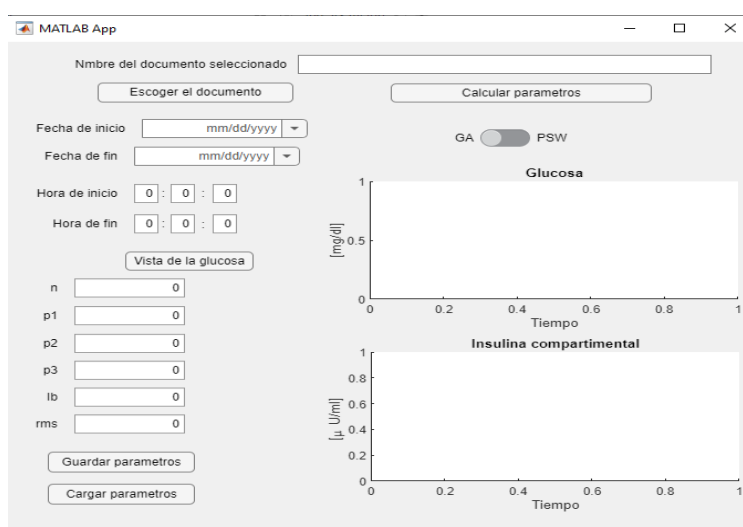


Figura 4.4: Interfaz de usuario, vista inicial.

En la segunda sección, localizada bajo la anterior, se ubica una columna de cajas numéricas donde se observan los parámetros resultantes de la parametrización. La última de estas cajas de texto no indica un parámetro reconstruido, sino que indica el RMS, el cual permite evaluar el desempeño del algoritmo al calcular los parámetros.

Otro elemento de la sección de Resultados numéricos, la segunda, son dos botones, uno para guardar resultados y otro para cargarlos. Estos botones permiten la creación de una base de datos a la cual el usuario puede añadir periódicamente los resultados que decida, lo que permite fácilmente comparar, recrear y recordar resultados generados previamente.

El botón Guardar permite agregar los siguientes datos, generados en un experimento, a una base de datos: parámetros generados, RMS del error, fechas utilizadas, algoritmo de identificación de parámetros utilizado y nombre del archivo de donde se obtuvieron los datos.

Para hacer uso de los datos de la base de datos se utiliza el botón de Cargar, que despliega una ventana donde se puede observar una lista de los datos guardados (figura 4.5). Desde

esta ventana se puede seleccionar un dato y cargar sus parámetros, con los que se recrea la dinámica sustituyéndolos en el modelo de Bergman en los límites de tiempo guardados. Estos parámetros se muestran en las cajas de texto de la segunda sección y en las gráficas. Igualmente se pueden borrar los datos de la lista, lo que permite que el usuario pueda depurar la base de datos desde la aplicación; el botón de Cancelar permite regresar a la pantalla principal.

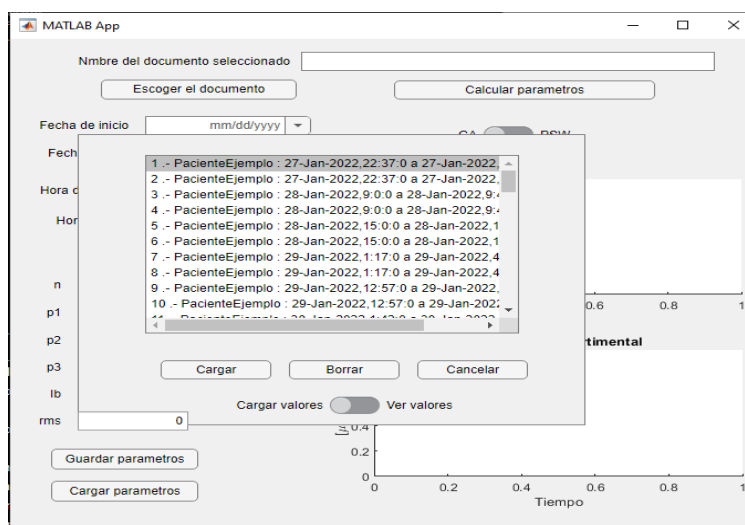


Figura 4.5: Menú para cargar datos.

En la pantalla de carga también se encuentra un *switch* en la parte inferior que permite observar la base de datos en forma de tabla (figura 4.6). Desde aquí se pueden observar cómodamente los valores guardados y es posible exportarlos a un archivo excel, lo cual permite que el usuario pueda hacer uso de los datos generados en una aplicación de alto conocimiento público. Todo esto facilita la comparación de resultados, la generación de las gráficas con estos y el compartir los datos.

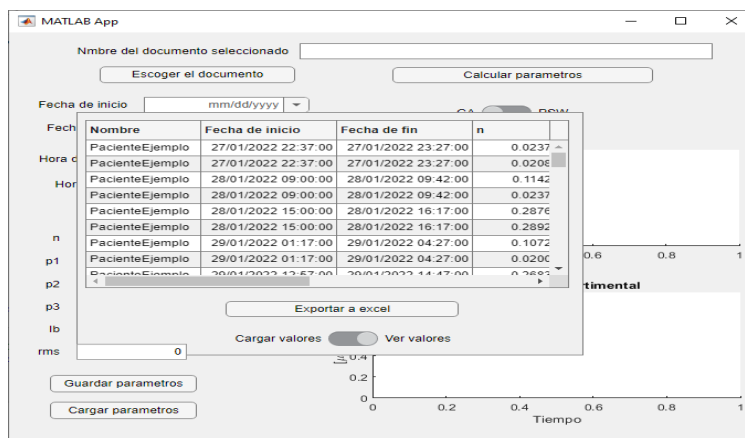


Figura 4.6: Tabla de datos guardados.

La tercera sección y la más llamativa a primera vista está formada por dos gráficas, que representan la glucosa e insulina compartimental. En estas gráficas se observa una vista previa al cálculo de los parámetros (al seleccionar el botón “Vista de la glucosa”), con lo cual el usuario puede tener una vista previa y hacer una buena elección del rango de tiempo a analizar. Las gráficas también muestran las reconstrucciones de los parámetros comparadas con los datos medidos, permiten un acercamiento a estos datos así como un análisis más detallado.

El diseño de la interfaz de usuario está enfocado en la fácil comprensión y amigable navegación, pues se buscó que el usuario pudiera aprender a utilizarla con facilidad y que fuera intuitiva. Es por esto que su uso está planeado para una lectura de arriba a abajo y según el camino de las manecillas del reloj, es decir, el usuario comienza definiendo los parámetros para realizar los cálculos en la parte superior, en la parte media de la aplicación observa el resultado en las gráficas y lo valida, para que al final, en el fondo, conociendo los parámetros generados decida si guardar o rehacer los datos (véase figura 4.7).

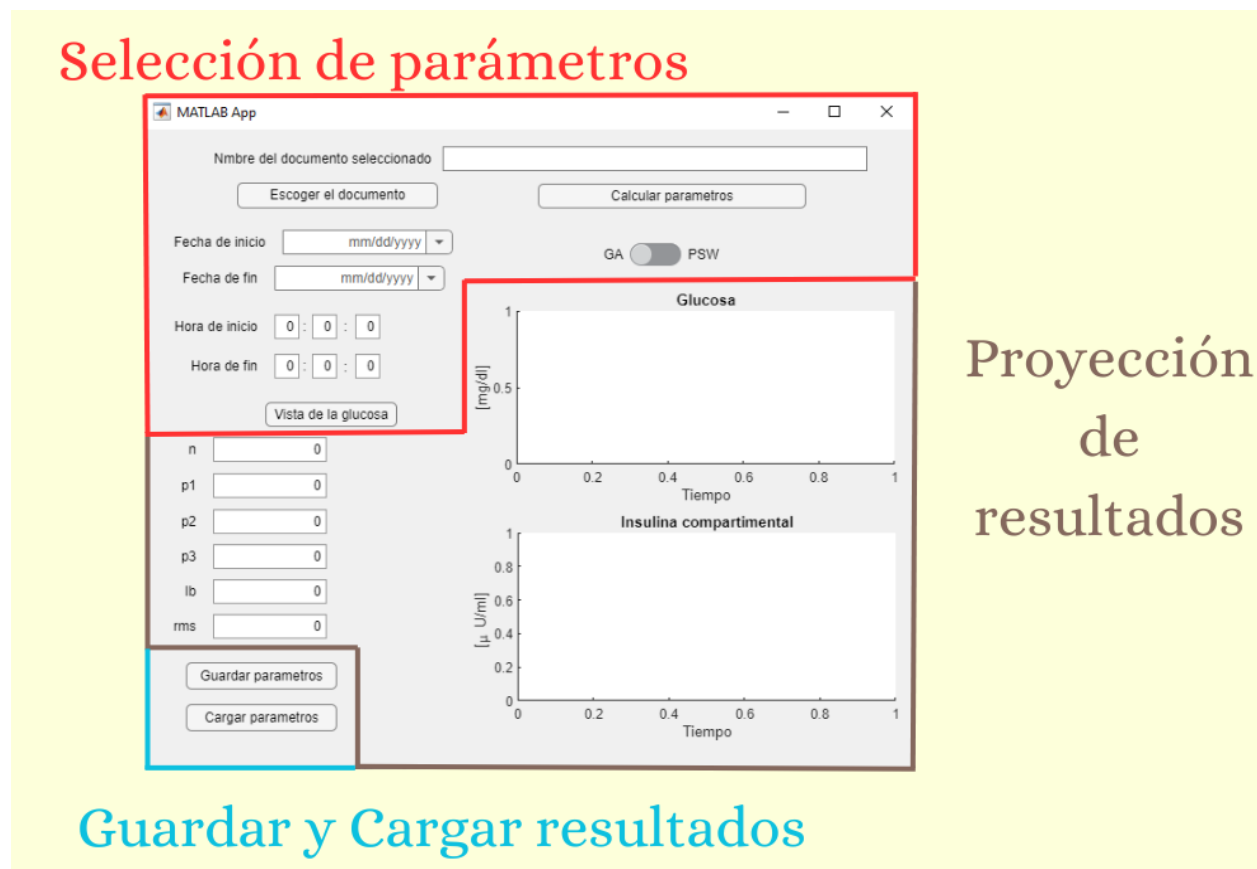


Figura 4.7: División de la interfaz gráfica.



# Capítulo 5

## Resultados

### 5.1. Guía para el uso de la interfaz de usuario

Una vez diseñada la interfaz de usuario en esta sección se presenta la manera en que se utilizó para los experimentos realizados. Para esto se muestra un ejemplo en el que se elegirán los datos a analizar, se identificarán los pámetros y se guardarán los resultados generados.

En la figura 5.1 se observa la elección del documento generado por la bomba de insulina, este se transfiere a la memoria de la computadora y se accede desde el explorador de archivos, al cual puede ser abierto desde la interfaz de usuario.

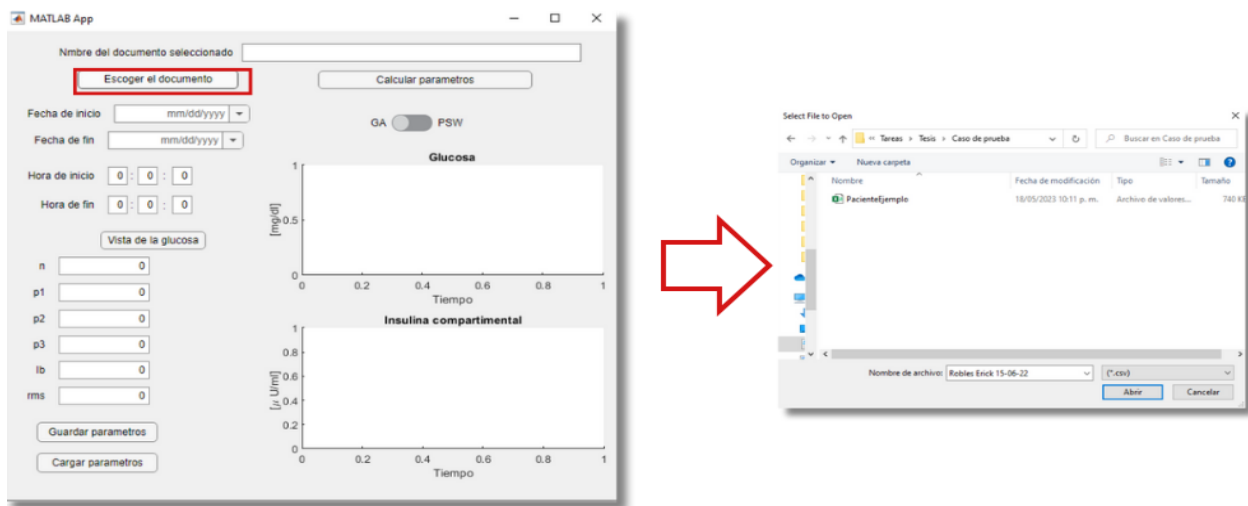


Figura 5.1: Elección del documento de donde se obtienen los datos.

Una vez se ha seleccionado el archivo con el que se trabajará, se selecciona la muestra de tiempo con la que se trabajará. Para esto se selecciona la fecha y hora en la sección 1 se la figura 5.2. El botón “Vista de la glucosa” permite observar una vista previa de los datos guardados de la glucosa, lo cual permite al usuario seleccionar la muestra de tiempo adecuada para el experimento.

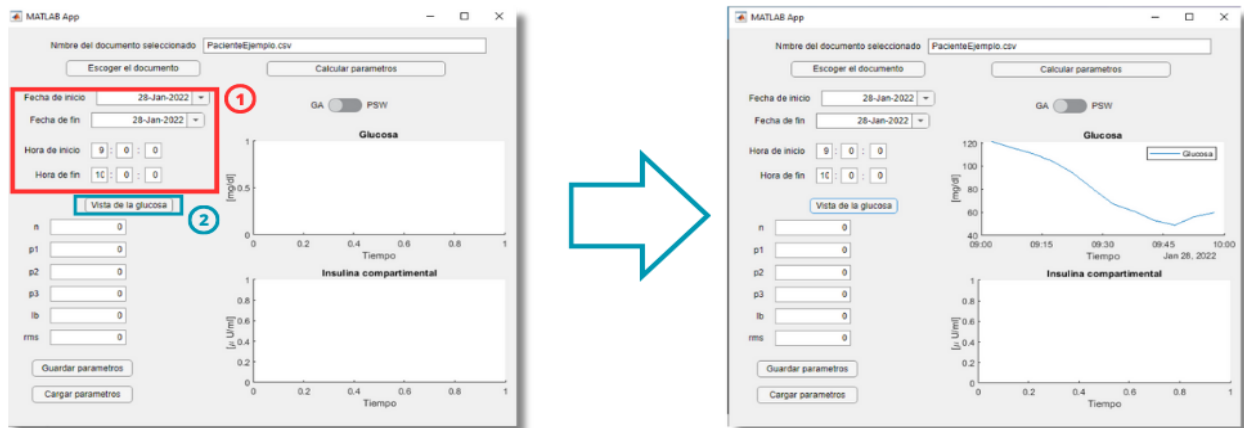


Figura 5.2: Selección de muestras de tiempo a analizar.

Una vez seleccionada una muestra se elige el método de indentificación de parámetros (figura 5.3). Una vez seleccionado el método se inicia el proceso utilizando el botón “Calcular parámetros”, una vez se realiza el cálculo se obtienen los parámetros calculados, el error RMS y la reconstrucción de la dinámica de la glucosa e insulina plasmática, lo que permite evaluar y comparar los resultados con las mediciones utilizadas.

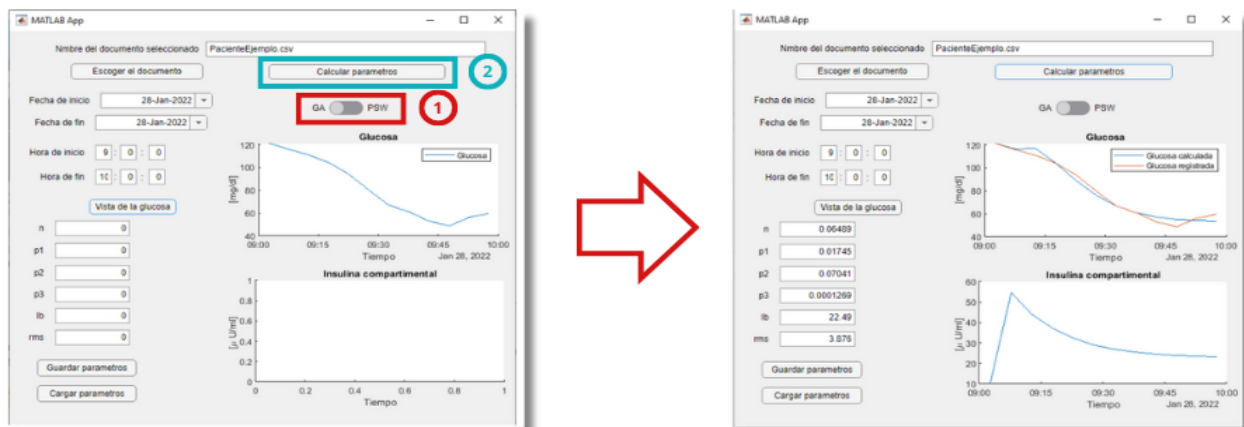


Figura 5.3: Cálculo de parámetros.

Por ultimo, si el usuario esta conforme con el resultado obtenido puede guardar los resultados obtenidos, para esto se utiliza el botón de “Guardar”. Para acceder a los resultados guardados se utiliza el botón “Cargar”, desde el cual se despliega una ventana con la que se pueden volver a cargar la dinámica generada, se pueden observar los resultados o generar un documento para exportar los resultados tipo excel (véase figura 5.4).

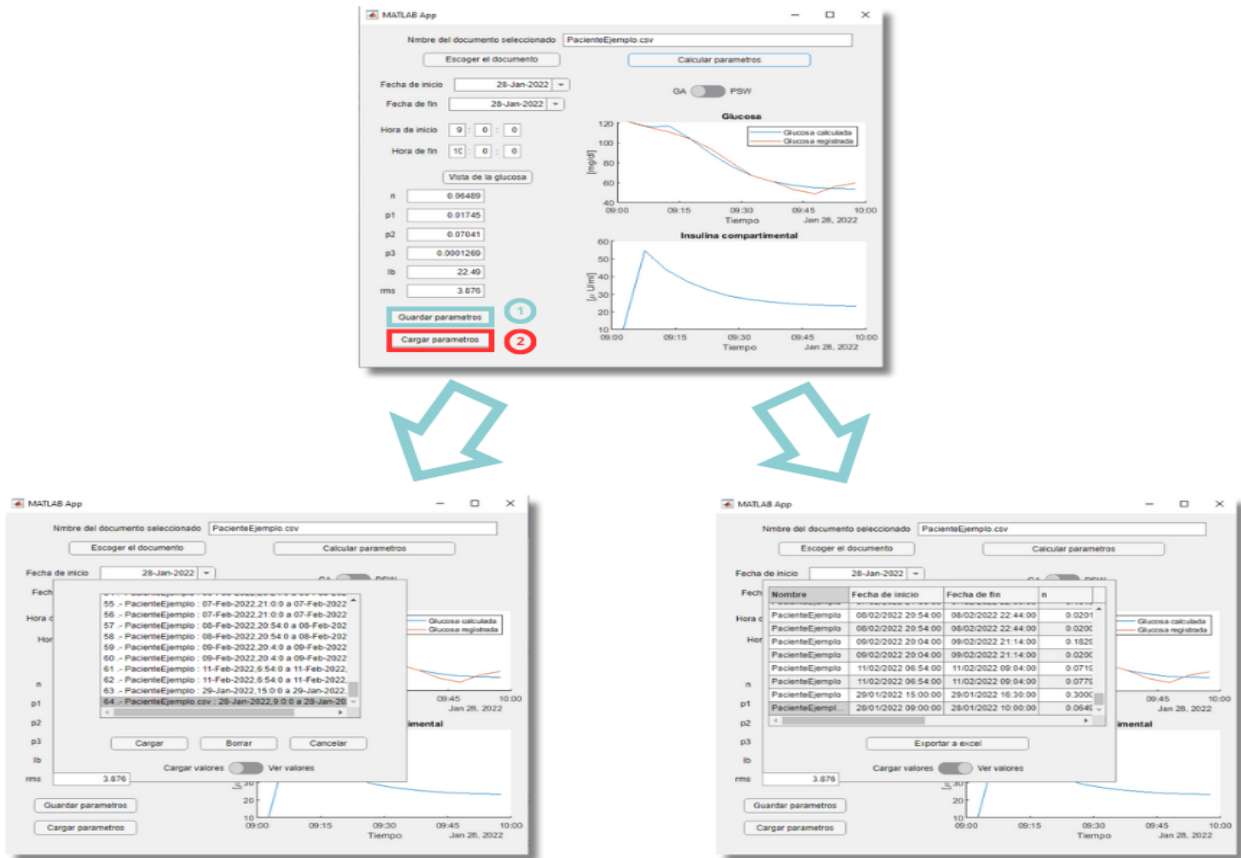


Figura 5.4: Guardado y cargado de resultados.

## 5.2. Coeficiente de correlación de Pearson y medidas de desempeño de Varvel

El coeficiente de correlación de Pearson se utiliza para medir la correlación entre dos grupos de datos del mismo tamaño. Se representa por  $r$  y puede variar entre 1 y -1. Donde 1 representa la correlación perfecta, -1, la correlación negativa perfecta y 0, que no hay correlación. Se calcula con la siguiente fórmula en la que  $x$  y  $y$  son un grupo de datos y  $n$  el número de datos.

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (5.1)$$

Los coeficientes de correlación de Varvel son tres, los cuales se calculan utilizando el error porcentual de predicción (PE). La media de error de predicción (MDPE) evalúa si existe un sesgo en los datos. La media de error de predicción absoluta (MDAPE) evalúa la inexactitud y el "Wobble" mide la variación de cada medición. Las fórmulas para calcular cada coeficiente

se obtuvieron de Gallardo *et al* [7].

$$PE[k] = 100 \frac{G[k] - \hat{G}[k]}{\hat{G}[k]} \quad (5.2)$$

$$MDPE = \text{mediana}\{PE[k], k = 0, \dot{s}, N - 1\} \quad (5.3)$$

$$MDAPE = \text{mediana}\{|PE[k]|, k = 0, \dot{s}, N - 1\} \quad (5.4)$$

$$MDPE = \text{mediana}\{|PE[k] - MDPE|, k = 0, \dot{s}, N - 1\} \quad (5.5)$$

### 5.2.1. Experimentos

Como se mencionó anteriormente, para probar el interfaz de usuario y ver el desempeño de los algoritmos (GA y PSW) se analizó un periodo de tiempo de 14 días, del que se seleccionaron 31 muestras, consideradas aptas para la parametrización del modelo mínimo de Bergman. Los dos algoritmos utilizaron puntos iniciales aleatorios y el espacio de búsqueda paramétrica se limitó a los siguientes rangos, obtenidos de [7] y que obedecen a los valores que se pueden obtener al ser medidos.

$$n \in [0,02, 0,3], p_1 \in [0,01, 0,1], p_2 \in [0,01, 0,2], \\ p_3 \in [10^{-5}, 10^{-3}], I_b \in [1, 30]$$

En las figuras 5.11 y 5.12 que presento más adelante se muestra la dinámica medida del paciente en cada muestra de tiempo, donde se muestra una caída en la glucosa como una línea sólida de color negro. La línea punteada en azul muestra la dinámica del modelo mínimo de Bergman reconstruida con los parámetros calculados con el algoritmo genético. Algo similar se realizó con la dinámica reconstruida usando un algoritmo de enjambre de partículas, la cual se representó con una línea punteada roja. Encima de cada gráfica es posible observar el número de muestra que se está representando y el error RMS obtenido con cada método.

En la tabla 5.1 se muestran los parámetros calculados. Cada uno proviene de una muestra a la que se le aplicó uno de los algoritmos de identificación de parámetros. Ya que se realizaron 31 muestras durante el experimento, de una duración promedio de una hora con 19 minutos, se calcularon 62 grupos de parámetros utilizando ambos métodos (GA y PSW). Para evaluar el desempeño de cada algoritmo se utilizó el coeficiente de correlación de Pearson y las medidas de desempeño de Varvel. Estos dos métodos comparan los datos experimentales con la dinámica reconstruida con el modelo mínimo de Bergman. En la primera columna de la tabla se observa el número de muestra con la que se obtuvieron los resultados, es posible observar cómo los parámetros generados por muestra se desempeñan al compararla con los resultados obtenidos en las gráficas 5.11 y 5.12.

El promedio del error RMS fue 4.1095 mg/dl, lo cual indica la poca diferencia entre los valores teóricos y experimentales, este error es de los ambos métodos. El promedio del error RMS de las medidas obtenidas con GA, fue 4.5495 mg/dl y el de PSW, 3.6695.



El promedio del MDPE de todos los datos es 0.1376 %. El promedio obtenido con GA es 0.2071 % y el obtenido con PSW es 0.0682 %. El promedio del MDAPE general es 2.0995 %, el obtenido con los datos generados por GA es 2.3771 %, y el de datos generados por PSW es igual a 1.8220 %. Por último, el "Wobble" general obtenido fue 1.9845 %, utilizando GA, 2.2232 % y PWS, 1.7458 %.

En las figuras 5.5, 5.6, 5.7, 5.8 y 5.9 se observan las magnitudes de cada parámetro obtenido en el eje de las ordenadas, en el de las abscisas se presenta cada una de las muestras. Es así como se observa cómo fueron cambiando los parámetros en cada una de las muestras. El título de cada gráfico de barras muestra la desviación estándar de producida por cada método.

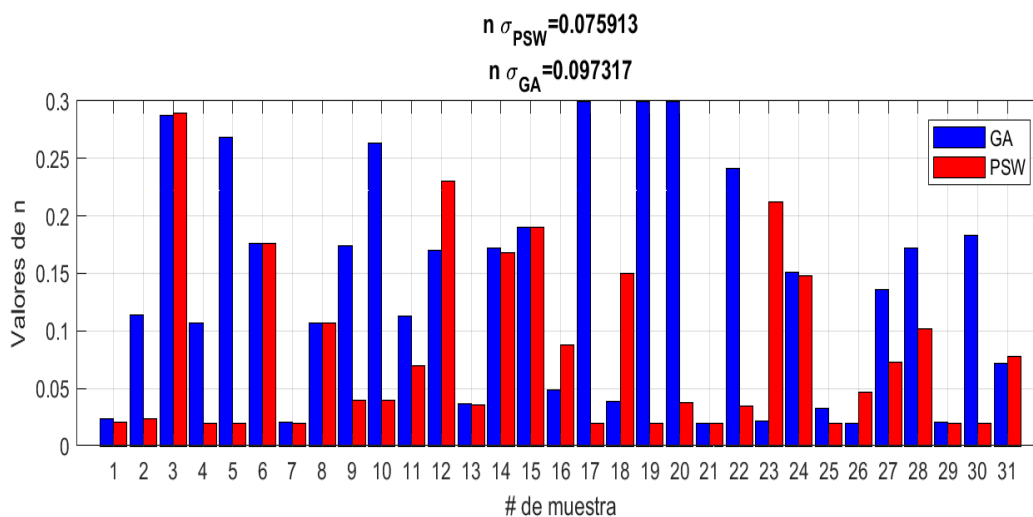


Figura 5.5: Parámetros  $n$  obtenidos durante el experimento generados por dos distintos métodos.

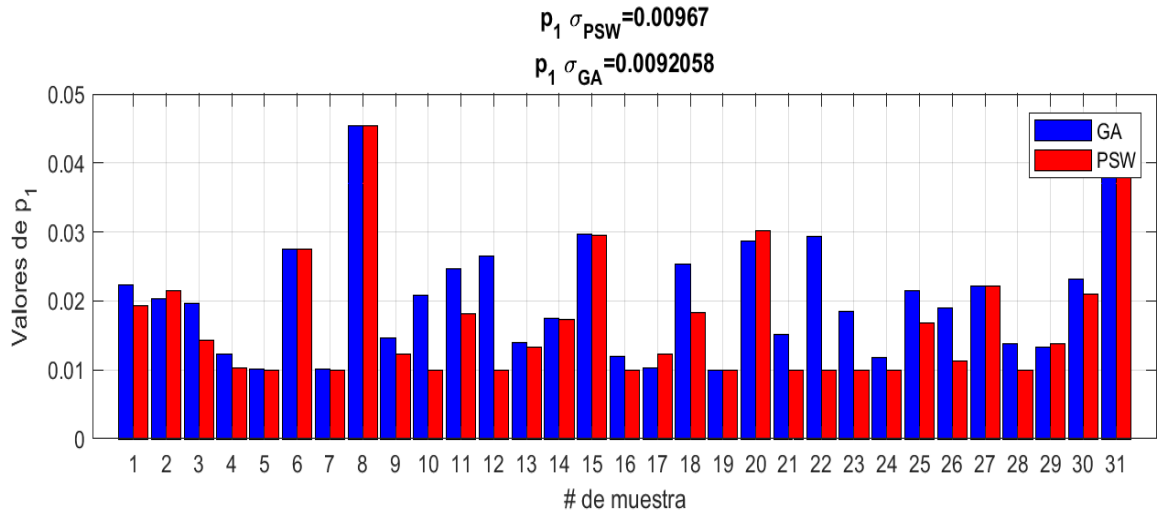


Figura 5.6: Parámetros  $p_1$  obtenidos durante el experimento generados por dos distintos métodos.

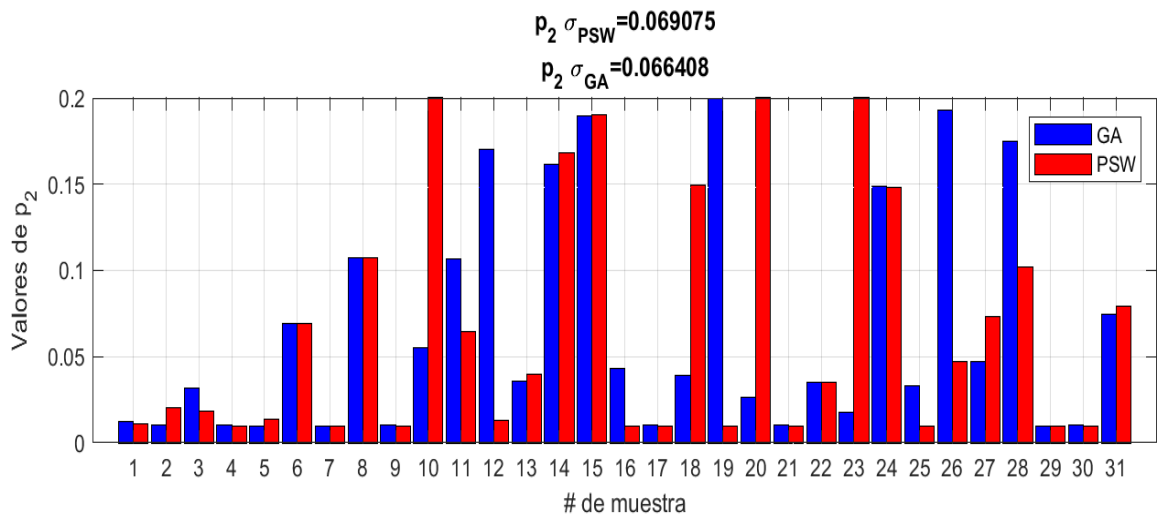


Figura 5.7: Parámetros  $p_2$  obtenidos durante el experimento generados por dos distintos métodos.

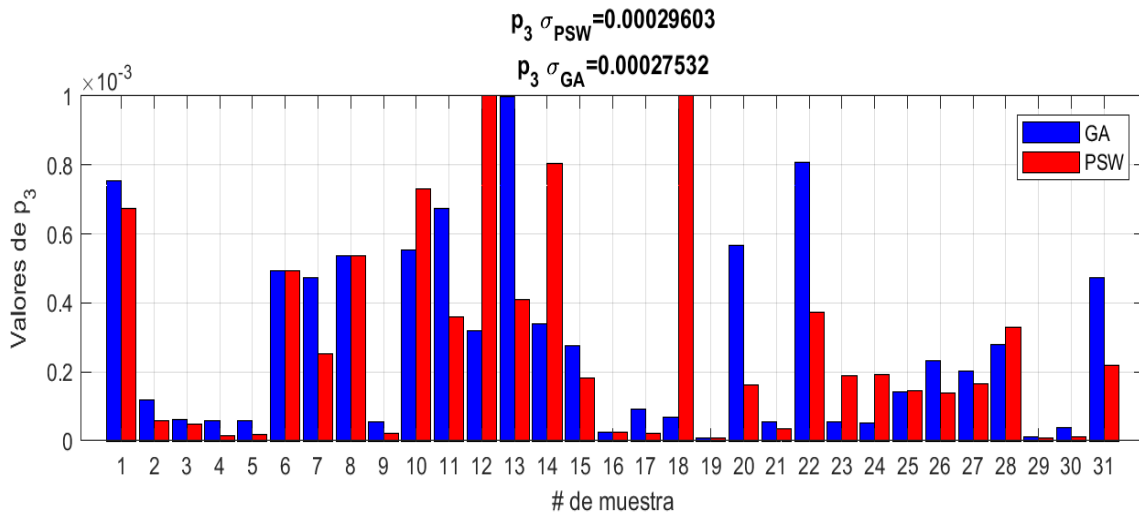


Figura 5.8: Parámetros  $p_3$  obtenidos durante el experimento generados por dos distintos métodos.

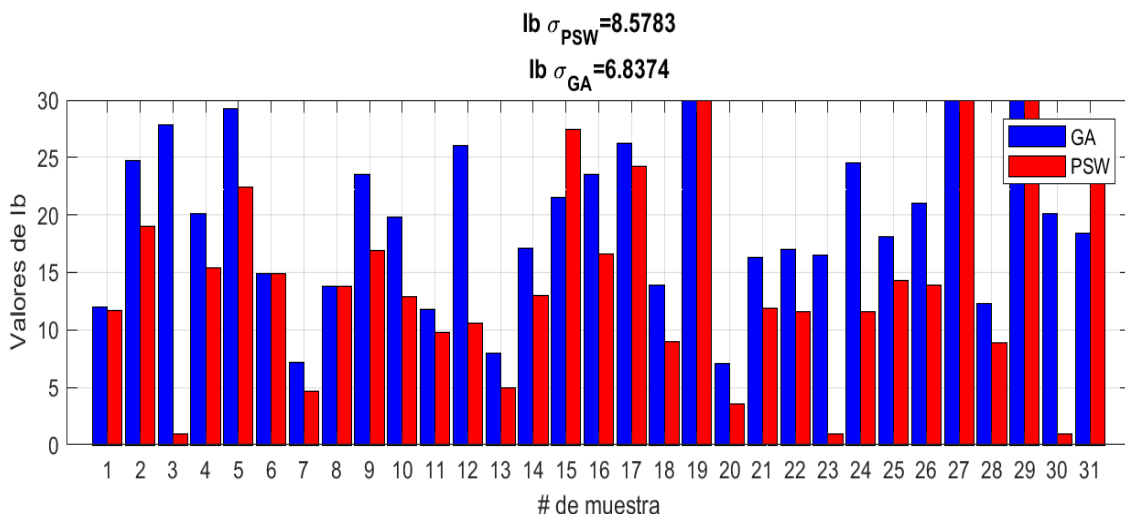


Figura 5.9: Parámetros  $I_b$  obtenidos durante el experimento generados por dos distintos métodos.

### 5.3. Análisis de resultados

En las gráficas 5.11 y 5.12 se observa detalladamente la efectividad de cada método en la recreación de la dinámica del comportamiento de la glucosa. Estas representaciones visuales proporcionan una visión clara y comparativa de cómo cada enfoque se desempeña en la simulación de la glucosa en el tiempo.

En el artículo previo [7], los autores emplearon el error RMS como métrica para validar la

efectividad del algoritmo genético desarrollado, logrando un valor promedio de 15.27 mg/dl. Este enfoque se consideró un estándar en la evaluación de la precisión de los métodos de parametrización.

Sin embargo, al utilizar la interfaz de usuario desarrollada en nuestro estudio, obtuvimos un error RMS aún más bajo para el caso de ejemplo, alcanzando solamente 4.1095 mg/dl. Esta diferencia significativa entre el error RMS obtenido mediante nuestro enfoque y el del artículo previo demuestra la efectividad y mejoría en los métodos de parametrización propuestos.

Estos resultados respaldan firmemente la afirmación de que los métodos de parametrización utilizados en nuestra interfaz de usuario son altamente efectivos en la recreación precisa de la dinámica del comportamiento de la glucosa. Esto implica que nuestra herramienta proporciona una solución confiable y prometedoras para la estimación y el monitoreo preciso de los niveles de glucosa en situaciones clínicas y experimentales.

Una vez comprobada la efectividad de ambos métodos, procedimos a analizar el comportamiento de los parámetros y su variación en cada muestra calculada. Este análisis se muestra detalladamente en las figuras 5.5, 5.6, 5.7, 5.8 y 5.9.

Al examinar las gráficas, observamos que la desviación estándar de los parámetros se mantiene en un orden de magnitud similar para ambos métodos. Esto implica que, en cada muestra medida, los parámetros calculados presentan una variación consistente en ambos métodos. Este hallazgo es importante para comprender la estabilidad y la consistencia de los resultados obtenidos con cada enfoque.

En particular, en la figura 5.10 se presenta el cálculo de la resistencia a la insulina utilizando los parámetros  $p_3$  y  $p_2$ . Es interesante observar cómo la resistencia a la insulina varía dependiendo del momento en el que se realiza la medición. Estos resultados resaltan la importancia de considerar el contexto temporal en el estudio de la resistencia a la insulina y demuestran cómo los parámetros pueden influir en su estimación.

En conclusión, el análisis detallado de los parámetros y su variación en cada muestra proporciona información valiosa sobre la dinámica y la interpretación de los resultados obtenidos con los métodos utilizados. Estos hallazgos contribuyen a mejorar nuestra comprensión de los factores que influyen en la resistencia a la insulina y respaldan la necesidad de considerar el momento de medición al interpretar y comparar los resultados en estudios clínicos y experimentales.

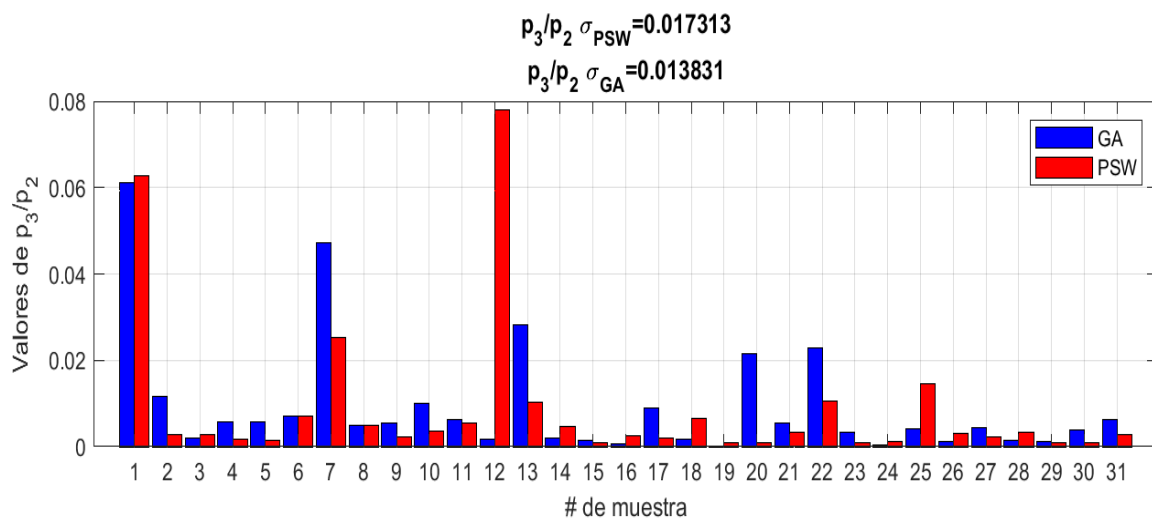


Figura 5.10: Resistencia a la insulina obtenida calculando  $p_3/p_2$  obtenidos por los dos distintos métodos

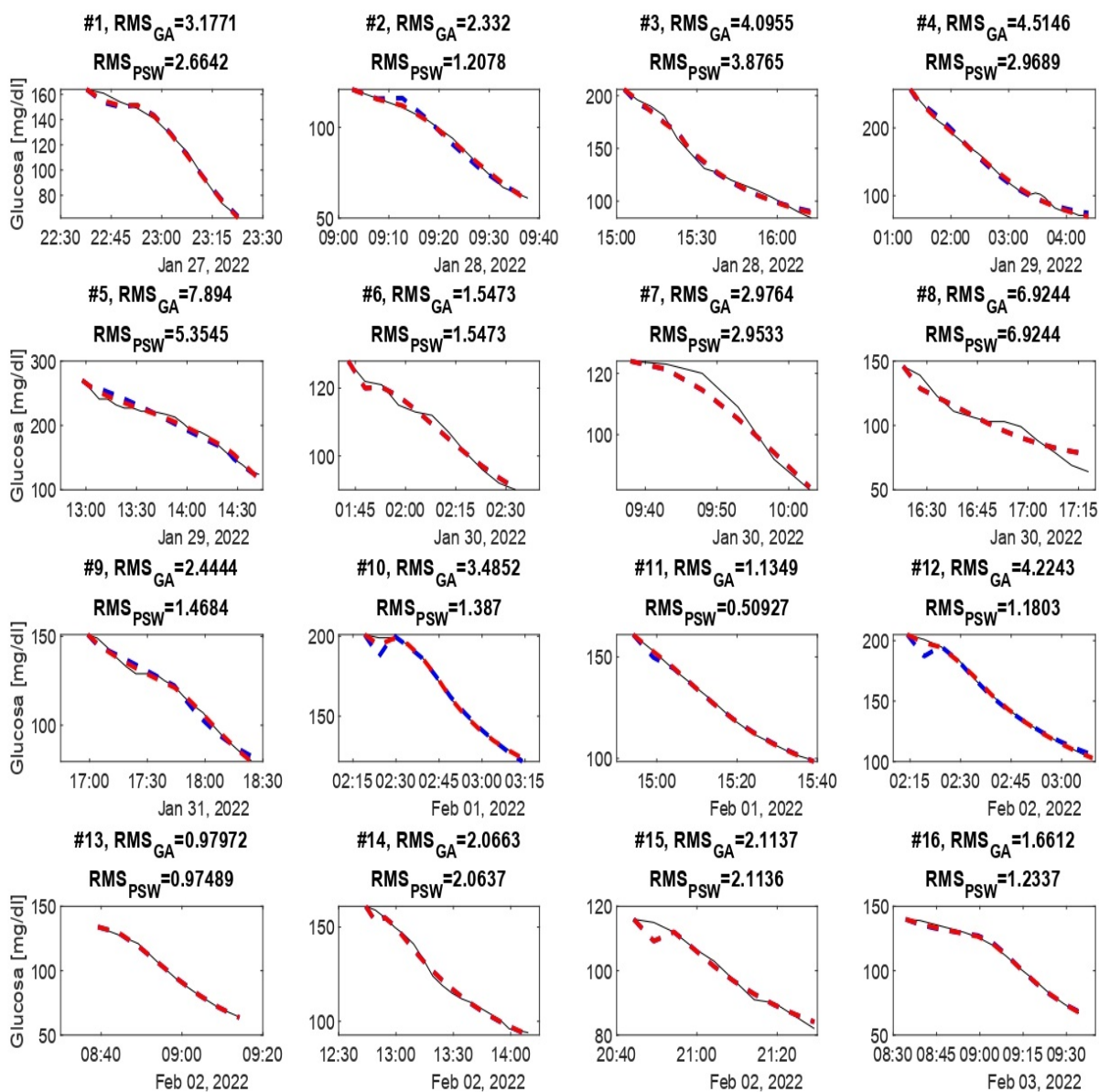


Figura 5.11: Dinámica de las primeras 16 muestras comparadas con las generadas con los diferentes métodos (GA y PSW).

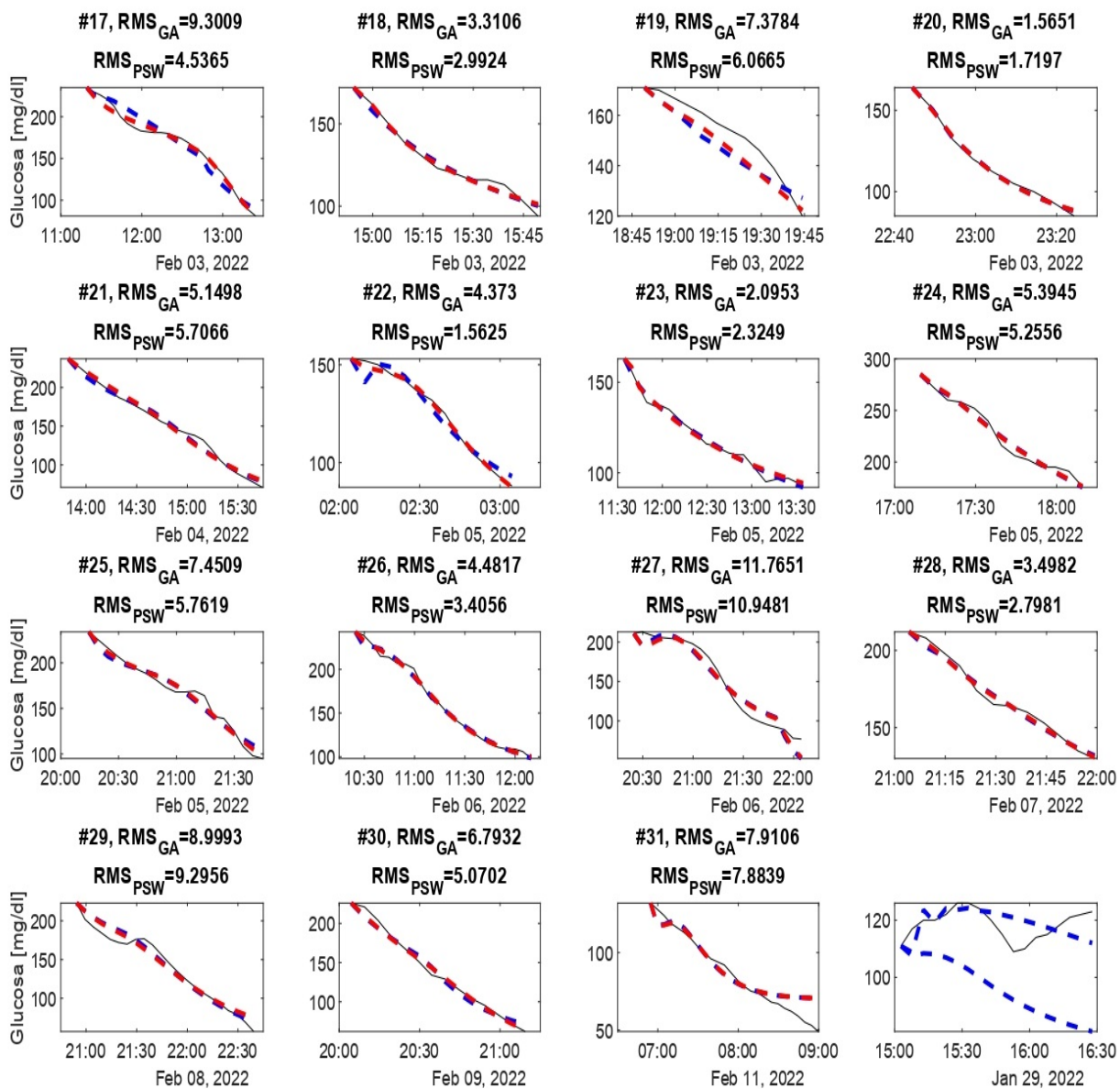


Figura 5.12: Dinámica de las utimas 15 muestras comparadas con las generadas con los diferentes métodos (GA y PSW).

M#	n	p1	p2	p3 · e <sup>3</sup>	Ib	Metodo	RMS	MDPE	MDAPE	Wobble	Corr
1	0.024	0.022	0.012	0.754	12.015	GA	3.177	-0.27	1.695	1.425	0.996
1	0.021	0.019	0.011	0.674	11.731	PSW	2.664	0.1	1.672	1.667	0.997
2	0.114	0.02	0.01	0.118	24.703	GA	2.332	0.083	1.26	1.343	0.995
2	0.024	0.022	0.021	0.058	19.02	PSW	1.208	0.192	0.758	0.95	0.998
3	0.288	0.02	0.032	0.061	27.86	GA	4.096	0.798	3.248	2.45	0.995
3	0.289	0.014	0.018	0.05	1	PSW	3.877	0	2.644	2.644	0.995
4	0.107	0.012	0.01	0.059	20.088	GA	4.515	-0.176	2.618	2.638	0.997
4	0.02	0.01	0.01	0.016	15.371	PSW	2.969	-0.404	1.388	1.44	0.999
5	0.268	0.01	0.01	0.059	29.265	GA	7.894	0.033	3.256	3.289	0.987
5	0.02	0.01	0.014	0.019	22.462	PSW	5.354	-1.089	1.798	2.256	0.992
6	0.176	0.028	0.07	0.492	14.863	GA	1.547	0	1.05	1.05	0.992
6	0.176	0.028	0.07	0.492	14.863	PSW	1.547	0	1.05	1.05	0.992
7	0.02	0.01	0.01	0.473	7.128	GA	2.976	0.713	1.877	2.361	0.989
7	0.02	0.01	0.01	0.253	4.625	PSW	2.953	0.701	1.885	2.345	0.989
8	0.107	0.045	0.107	0.538	13.775	GA	6.924	0.417	4.129	4.546	0.966
8	0.107	0.045	0.107	0.538	13.775	PSW	6.924	0.417	4.129	4.546	0.966
9	0.174	0.015	0.01	0.056	23.55	GA	2.444	-0.061	1.418	1.371	0.994
9	0.04	0.012	0.01	0.023	16.922	PSW	1.468	-0.116	0.658	0.713	0.998
10	0.263	0.021	0.055	0.554	19.812	GA	3.485	0.071	0.554	0.598	0.993
10	0.04	0.01	0.2	0.732	12.888	PSW	1.387	0.174	0.226	0.183	0.999
11	0.113	0.025	0.107	0.674	11.753	GA	1.135	-0.137	0.396	0.373	0.999
11	0.069	0.018	0.065	0.36	9.76	PSW	0.509	-0.007	0.277	0.269	1
12	0.17	0.027	0.17	0.319	25.988	GA	4.224	-0.075	0.578	0.653	0.995
12	0.23	0.01	0.013	1	10.56	PSW	1.18	0.21	0.337	0.254	1
13	0.037	0.014	0.036	1	8.022	GA	0.98	-0.089	0.68	0.593	0.999
13	0.035	0.013	0.04	0.409	4.995	PSW	0.975	-0.058	0.626	0.596	0.999
14	0.172	0.018	0.161	0.338	17.156	GA	2.066	0.373	1.109	0.848	0.996
14	0.168	0.017	0.168	0.805	12.983	PSW	2.064	0.269	1.068	0.881	0.996
15	0.19	0.03	0.19	0.277	21.475	GA	2.114	0.117	0.639	0.639	0.988
15	0.19	0.03	0.19	0.182	27.465	PSW	2.114	0.133	0.643	0.643	0.988
16	0.049	0.012	0.043	0.026	23.572	GA	1.661	0	1.112	1.112	0.998
16	0.088	0.01	0.01	0.025	16.588	PSW	1.234	0	0.834	0.834	0.999
17	0.3	0.01	0.01	0.091	26.272	GA	9.301	-0.213	4.161	4.325	0.98
17	0.02	0.012	0.01	0.021	24.202	PSW	4.536	0.088	1.904	1.904	0.995
18	0.039	0.025	0.039	0.067	13.893	GA	3.311	-0.007	1.491	1.491	0.99
18	0.15	0.018	0.15	1	8.96	PSW	2.992	-0.112	0.384	0.372	0.991
19	0.3	0.01	0.2	0.01	30	GA	7.378	4.35	5.058	2.204	0.944
19	0.02	0.01	0.01	0.01	30	PSW	6.066	3.504	3.504	1.431	0.976
20	0.3	0.029	0.026	0.568	7.116	GA	1.565	0.313	0.773	0.739	0.998
20	0.038	0.03	0.2	0.164	3.521	PSW	1.72	0.46	1.099	0.828	0.998
21	0.02	0.015	0.01	0.056	16.326	GA	5.15	-0.537	2.563	2.56	0.995
21	0.02	0.01	0.01	0.034	11.88	PSW	5.707	-1.885	2.53	1.961	0.993
22	0.241	0.029	0.035	0.807	17.036	GA	4.373	-0.295	2.191	1.896	0.983
22	0.035	0.01	0.035	0.372	11.63	PSW	1.562	0	0.687	0.687	0.998
23	0.022	0.019	0.018	0.057	16.461	GA	2.095	0.247	1.248	1	0.994
23	0.212	0.01	0.2	0.189	1	PSW	2.325	0.09	1.314	1.262	0.993
24	0.151	0.012	0.149	0.051	24.572	GA	5.394	0	2.035	2.035	0.988
24	0.148	0.01	0.148	0.193	11.577	PSW	5.256	-0.298	2.3	2.002	0.988
25	0.033	0.021	0.033	0.142	18.082	GA	7.451	0.405	3.969	3.564	0.984
25	0.02	0.017	0.01	0.146	14.3	PSW	5.762	-0.256	2.671	2.645	0.99
26	0.02	0.019	0.193	0.233	21.052	GA	4.482	-0.375	1.733	1.444	0.996
26	0.047	0.011	0.047	0.139	13.856	PSW	3.406	-0.001	1.27	1.269	0.997
27	0.136	0.022	0.047	0.203	30	GA	11.765	0.329	7.293	6.964	0.974
27	0.073	0.022	0.073	0.165	30	PSW	10.948	1.033	6.203	5.974	0.978
28	0.172	0.014	0.175	0.281	12.302	GA	3.498	0.484	1.56	1.845	0.992
28	0.102	0.01	0.102	0.331	8.903	PSW	2.798	0.391	1.445	1.231	0.995
29	0.02	0.013	0.01	0.012	29.992	GA	8.999	0.104	4.841	4.841	0.984
29	0.02	0.014	0.01	0.01	30	PSW	9.296	-0.243	5.059	4.816	0.981
30	0.183	0.023	0.01	0.039	20.156	GA	6.793	1.08	4.905	4.081	0.993
30	0.02	0.021	0.01	0.011	1	PSW	5.07	0.155	1.932	1.919	0.996
31	0.072	0.047	0.075	0.475	18.383	GA	7.911	-1.262	4.249	4.64	0.966
31	0.078	0.045	0.079	0.22	26.296	PSW	7.884	-1.334	4.186	4.547	0.969

Tabla 5.1: Tabla que presenta los parámetros calculados, el método por el que fueron calculados, las medidas de desempeño de Varvel y el coeficiente de correlación de Pearson.



## 5.4. Contribuciones

### 5.4.1. Presentación de póster en ATTD 2023

La Convención sobre el Tratamiento y Tecnologías Avanzadas para la Diabetes (ATTD, por sus siglas en inglés: *Advanced Technologies & Treatments for Diabetes*) reúne anualmente a expertos de todo el mundo en el campo de la diabetes, incluyendo médicos, científicos e investigadores. Durante este evento, se llevan a cabo presentaciones y discusiones sobre los últimos avances en tecnología para el tratamiento de la diabetes.

La ATTD aborda una amplia gama de temas, centrándose en tecnologías avanzadas y tratamientos innovadores. Entre ellos se incluyen dispositivos de monitorización continua de glucosa (CGM), sistemas de páncreas artificial, sistemas de administración de insulina, aplicaciones móviles y otras tecnologías relacionadas. Además, se exploran temas emergentes como la inteligencia artificial, el análisis de grandes volúmenes de datos (*big data*) y la telemedicina en el contexto de la diabetes.

Esta conferencia proporciona un espacio invaluable para que los investigadores compartan los resultados de estudios clínicos y científicos, presenten nuevas tecnologías y discutan temas relevantes en el campo de la diabetes. Asimismo, ofrece oportunidades de aprendizaje, *networking* y colaboración entre profesionales y expertos en diabetes, fomentando el avance continuo en el manejo y tratamiento de esta enfermedad.

En la ATTD 2023, celebrada en Berlín, se presentó un póster electrónico [15] que muestra la interfaz de usuario diseñada en el marco de esta tesis. El póster, representado en la figura 5.13, ilustra de manera visual la interfaz de usuario desarrollada y destaca las características y funcionalidades clave de la misma.

Este tipo de eventos científicos y las oportunidades que brindan para compartir conocimientos y experiencias son fundamentales para impulsar la investigación y el desarrollo de tecnologías innovadoras en el campo de la diabetes, con el objetivo de mejorar la calidad de vida de las personas afectadas por esta enfermedad crónica.

# Insulin resistance monitoring application for highly individualized diabetes treatment

Diabetes is one of the oldest recorded diseases, yet instead of having the ability to stop it, its global burden is growing. Diabetes treatment has proven to be most effective when it is highly individualized. We developed an application that automatically analyzes the glucose sensor and insulin infusion information to calculate insulin resistance in order monitor the variations.



### INTRODUCTION

The Bergman minimal model is a model designed to describe the glucose-insulin dynamics. As the model was created using experimental data, it focuses on simplicity, therefore it is popular in computer modeling and practical uses.

$$\begin{aligned} \dot{G} &= -p_1[G - G_b] - GX \\ \dot{X} &= -p_2[X] + p_3[I - I_b] \\ \dot{I} &= -n[I - I_b] + \gamma[G - h]t + u(t) \end{aligned}$$

By defining the parameters, using a glucose sensor and an insulin, an individual personal model can be obtained being able to predict the future behavior of the system.

### GENETIC ALGORITHM

A Genetic Algorithm is a global minimization algorithm. It resembles natural selection in a population of individuals formed by different values of the parameters.

The initial population is made up of a set of randomly generated individuals. Each is tested against a fitness function, where the most capable individuals are selected and a new generation, created by recombining the selected ones, replaces the unselected part of the population. This process is repeated all along several generations to reduce the RMS error.

### METHODS

Insulin resistance is calculated with a Genetic Algorithm (GA) based on the glucose-insulin regulation minimal model. We used a GA for model parameter estimation as an efficient method to search the parameters' values space to converge on a set of values that minimize an appropriately created fitness function. The model's parameters are tuned so its dynamics fit as closely as possible to the experimental data obtained from a glucose sensor and an infusion pump. The GA is embedded in the application that automatically calculates the insulin resistance for any given time frame that the physician wants to analyze in order to have a tool to individualize the diabetes treatment.

### APP DESIGN

For the design, App Designer was used, a MATLAB tool to create graphical user interfaces. The design is focused on allowing the user to easily choose a sample to analyze and for the generated parameters to be easily viewed, stored, and reviewed.

The user can choose a time frame to analyze a .csv file, that is generated by the insulin bomb (Medtronic). Then the calculated parameters are shown in the text box and the model generated is compared to the measurements in the graphs,

so the physician can review the performance of the algorithm.

The parameters can be stored in an easily readable spreadsheet database and can be reproduced in the app at any time.

### CONCLUSION

To test the app, it was tested with a database of 20 patients. Varvel's performance measurements showed that the system has high precision with low bias. Variations in insulin resistance over time were plotted to facilitate interpretation.

With these results, we conclude that the app is capable of tracking changes in insulin resistance throughout the day, therefore, creating an individualized treatment schedule to minimize glucose off-target time.

### References:



### Contact:



Authors: A.G. Gallardo-Hernández, G. Villanueva-Aragón, M.A. González-Olvera, J. Escobar, D.J. López-Mezquita, C. Nevilla-Monsalve, R. Leber, Instituto Mexicano del Seguro Social, Unidad de Investigación en Enfermedades Metabólicas, Mexico City, Mexico, 20/Universidad Nacional Autónoma de México, Facultad de Ingeniería, Mexico City/Mexico, 3/Universidad Autónoma de la Ciudad de México, Colegio de Ciencia y Tecnología, Mexico City, Mexico, 4/Instituto Politécnico Nacional, Escuela Superior de Ingeniería Mecánica y Eléctrica Zacatepec, Mexico City, Mexico, 5/Secretaría de la Defensa Nacional, Hospital Central Militar, Mexico City, Mexico.

Figura 5.13: Póster que se presentó en ATTD 2023. [15]

# Capítulo 6

## Conclusiones

Se diseñó una interfaz de usuario que permite utilizar el Algoritmo Genético para la identificación de parámetros. La GUI diseñada permite comprobar los resultados obtenidos en el artículo que procede a este trabajo [7], al permitir crear fácilmente múltiples experimentos, donde es posible ver la efectividad de este método para replicar la dinámica deseada.

Igualmente se integró en la interfaz de usuario la capacidad de utilizar el algoritmo Enjambre de Partículas para la identificación de parámetros, con el objetivo de comparar los resultados. Se construyó la GUI para poder repetir los experimentos fácilmente utilizando ambos métodos de identificación de parámetros.

Se comprobó que la GUI es óptima al realizar una serie de experimentos utilizando ambos métodos, serie de experimentos del que se pudo llegar a sacar conclusiones respecto de cómo cambia la resistencia a la insulina. Esto nos permite confirmar que el diseño de la interfaz de usuario es adecuada para realizar múltiples experimentos.

Aunque no se recibió una realimentación sobre el diseño de la interfaz gráfica, se buscó realizar un diseño que facilite el manejo del usuario.

Se desarrolló un póster en el cual se resumió el proyecto. El cual se presentó en una conferencia internacional.

Creando una interfaz cómoda y adecuada para la parametrización del Modelo Mínimo de Bergman se buscó desarrollar herramientas tecnológicas para una mejor individualización del tratamiento de la diabetes de una forma poco invasiva.

Se llevó a cabo una verificación de la efectividad de los dos algoritmos de identificación de parámetros, demostrando que ambos tienen la capacidad de reconstruir adecuadamente la dinámica deseada. Sin embargo, aún queda pendiente evaluar los recursos computacionales que requiere cada uno de los algoritmos.

## 6.1. Trabajo futuro

Con fundamento en las conclusiones obtenidas, definimos el trabajo futuro: es necesario comprobar y mejorar el diseño asesorándonos con un equipo interdisciplinario de posibles usuarios.

Ya que la interfaz de usuario fue diseñada para usuarios sin un alto conocimiento en los algoritmos de identificación de parámetros, son ellos los que tienen que comprobar que es fácil de manejar, aprender a utilizar y por qué no, mejorar, en beneficio de un mejor control de su salud.

Igualmente se propone utilizar una Red Neuronal para poder construir las muestras que en este trabajo se construyeron manualmente, esto hará los experimentos aún más rápidos y accesibles.

# Bibliografía

- [1] Abdul Afram y Farrokh Janabi-Sharifi. “Theory and applications of HVAC control systems—A review of model predictive control (MPC)”. En: *Building and Environment* 72 (2014), págs. 343-355.
- [2] American Diabetes Association. “Diagnosis and classification of diabetes mellitus”. En: *Diabetes care* 34.Supplement\_1 (2011), S62-S69.
- [3] B. Wayne Bequette. “A critical assessment of algorithms and challenges in the development of a closed-loop artificial pancreas”. En: *Diabetes technology & therapeutics* 7.1 (2005), págs. 28-47.
- [4] Ewart R. Carson y Tibor Deutsch. “A spectrum of approaches for controlling diabetes”. En: *IEEE Control Systems Magazine* 12.6 (1992), págs. 25-31.
- [5] F. Ekram et al. “A feedback glucose control strategy for type II diabetes mellitus based on fuzzy logic”. En: *The Canadian Journal of Chemical Engineering* 90.6 (2012), págs. 1411-1417.
- [6] Nuha A. ElSayed et al. “2. Classification and Diagnosis of Diabetes: Standards of Care in Diabetes—2023”. En: *Diabetes Care* 46.Supplement\_1 (2023), S19-S40.
- [7] Ana Gabriela Gallardo-Hernández et al. “Minimally-Invasive and Efficient Method to Accurately Fit the Bergman Minimal Model to Diabetes Type 2”. En: *Cellular and Molecular Bioengineering* 15.3 (2022), págs. 267-279.
- [8] Ana Gabriela Hernández-Gallardo et al. “High-order sliding-mode control for blood glucose: Practical relative degree approach”. En: *Control Engineering Practice* 21.5 (2013), págs. 747-758.
- [9] Wendy L. Martínez. “Graphical user interfaces”. En: *Wiley Interdisciplinary Reviews: Computational Statistics* 3.2 (2011), págs. 119-133.
- [10] Brooke H. McAdams y Ali A. Rizvi. “An overview of insulin pumps and glucose sensors for the generalist”. En: *Journal of clinical medicine* 5.1 (2016), pág. 5.
- [11] Katherine Ogurtsova et al. “IDF diabetes Atlas: Global estimates of undiagnosed diabetes in adults for 2021”. En: *Diabetes research and clinical practice* 183 (2022), pág. 109118.
- [12] Simona Panunzi et al. “A revised Sorensen model: Simulating glycemic and insulinemic response to oral and intra-venous glucose load”. En: *Plos one* 15.8 (2020), e0237215.
- [13] Marcio Schwaab et al. “Nonlinear parameter estimation through particle swarm optimization”. En: *Chemical Engineering Science* 63.6 (2008), págs. 1542-1552.

- [14] Garry M. Steil, Antonios E. Panteleon y Kerstin Rebrin. “Closed-loop insulin delivery—the path to physiological glucose control”. En: *Advanced drug delivery reviews* 56.2 (2004), págs. 125-144.
- [15] Gabriel Villanueva Aragón et al. *Insulin resistance monitoring application for highly individualized diabetes treatment*. Presented at: Advanced Technologies & Treatments for Diabetes 2023. Berlín, Alemania. 2023.
- [16] Youfa Wang et al. “A systematic review of application and effectiveness of mHealth interventions for obesity and diabetes treatment and self-management”. En: *Advances in Nutrition* 8.3 (2017), págs. 449-462.
- [17] Lotfi A. Zadeh. “Fuzzy logic—a personal perspective”. En: *Fuzzy sets and systems* 281 (2015), págs. 4-20.