



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

IMPLEMENTACIÓN DE UN INTEGRADOR DE MODELOS DE PACIENTES VIRTUALES DIABÉTICOS TIPO 1

AUTOR: ANDREA ANAYA OLIVER

TUTOR: JOSÉ LUÍS DIEZ RUANO

COTUTOR: JORGE BONDÍA COMPANYY

Curso Académico: 2013-14

Prefacio

Es necesario justificar la estructura del presente Trabajo Fin de Grado puesto que no se ajusta a la estructura convencional. El trabajo forma parte de un proyecto de investigación en el que participa el Departamento de Ingeniería de Sistemas y Automática de la Universidad Politécnica de Valencia.

El proyecto de investigación va dirigido al tratamiento de los pacientes con Diabetes Mellitus y el trabajo realizado está enfocado de forma que complemente el trabajo desarrollado por el departamento.

Dentro de este contexto, el presente trabajo toma la estructura de un proyecto de programación y por ello no aparecen determinadas partes habituales en un Trabajo Fin de Grado como son los planos o el pliego de condiciones ya que carecen de sentido en este caso particular.

Documentos del proyecto

1. MEMORIA
2. PRESUPUESTO



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

IMPLEMENTACIÓN DE UN INTEGRADOR DE MODELOS DE PACIENTES VIRTUALES DIABÉTICOS TIPO 1

DOCUMENTO 1

MEMORIA

Contenido

1. Planteamiento del problema.....	3
1.1. Objeto del proyecto	4
1.2. Características del proyecto	4
2. Motivación y justificación.....	7
2.1. Motivación del proyecto.....	7
2.2. Justificación del proyecto	8
2.2.1. Interés académico	8
2.2.2. Interés tecnológico.....	8
3. Diabetes.....	11
3.1. Metabolismo de la glucosa	11
3.1.1. Insulina	12
3.1.2. Glucagón.....	13
3.1.3. Epinefrina	13
3.1.4. Hormonas de acción lenta	14
3.2. Diabetes Mellitus	14
3.2.1. Diabetes tipo 1	16
3.2.2. Diabetes tipo 2	16
3.2.3. Tratamiento.....	17
3.2.4. Complicaciones de la enfermedad.....	19
3.3. Páncreas artificial.....	20
1.3.1. El problema del control.....	21
4. Control postprandial.....	25
4.1. Algoritmos de control	27

4.2. Situaciones de control	28
5. Modelado del sistema glucorregulador	29
5.1. Introducción.....	29
5.2. Procesos fisiológicos del sistema insulina-glucosa	30
5.3. Selección del modelo para el simulador	30
5.4. Modelo de Cambridge	31
5.4.1. Modelo de absorción de glucosa	31
5.4.2. Modelo de absorción subcutánea de insulina	32
5.4.3. Modelo del sistema glucorregulador	32
6. Alternativas de solución	37
6.1. Lenguaje de programación	37
6.2. Introducción de datos	38
6.3. Comunicación.....	39
7. Metodología	41
7.1. Diseño del simulador	41
7.2. Diseño de la interfaz gráfica	46
7.3. Diseño de la comunicación	50
8. Resultados	55
9. Conclusiones.....	69
10. Trabajo futuro.....	73
Referencias	75
Anexos	77
Anexo I. Manual de usuario	77
Anexo II. Manual del programador.....	101

Capítulo 1

Planteamiento del problema

La diabetes es una enfermedad que se caracteriza por la presencia de altos niveles de glucosa en sangre. Esto causa graves consecuencias en el organismo y sin tratamiento se convierte en una enfermedad mortal. El tratamiento habitual consiste en múltiples inyecciones diarias de insulina, lo que permite disminuir los niveles de glucosa hasta valores no perjudiciales para el paciente. Esta enfermedad cada día afecta a más personas y es por ello por lo que se están realizando diversas investigaciones para mejorar el tratamiento de forma que sea más eficaz y que mejore la calidad de vida de las personas diabéticas.

Como se explicará más adelante, se pueden distinguir principalmente dos tipos de pacientes: los diabéticos tipo 1 y los diabéticos tipo 2. Puesto que se trata de patologías distintas el tratamiento será distinto en cada caso. Por tanto, las líneas de investigación para mejorar el tratamiento se dividen. En unos casos están orientadas a tratar a los pacientes tipo 1 y en otros a los tipo 2.

La línea de investigación del Departamento de Ingeniería de Sistemas y Automática en la que se sitúa el presente proyecto está orientada a los pacientes del primer tipo. El objetivo principal de esta investigación es desarrollar un dispositivo llamado páncreas artificial que permita que el paciente participe lo menos posible en el tratamiento y pueda llevar una vida lo más similar posible a la de una persona sana.

En el proceso de desarrollo del páncreas artificial los ensayos clínicos en seres humanos son una parte fundamental para evaluar la seguridad y eficacia de los dispositivos. Deben ser evaluados en un amplio rango de escenarios para garantizar la seguridad y demostrar la capacidad de realizar el tratamiento de forma eficaz. Los ensayos preclínicos para los sistemas de administración de insulina a menudo se realizan con animales, mientras que las pruebas de optimización se realizan en seres humanos.

Las pruebas con pacientes diabéticos y, en menor medida, con animales consumen mucho tiempo, requieren muchos recursos, son muy costosas y por cuestiones éticas dificultan el desarrollo acelerado y los esfuerzos que conducen a la comercialización de los dispositivos.

1.1. Objeto del proyecto

En lugar de realizar pruebas clínicas en pacientes reales se busca realizarlas en ordenadores que dispongan de sistemas que simulen el comportamiento del metabolismo de los pacientes con diabetes tipo 1. El entorno de simulación se diseña para dar soporte al desarrollo y evaluación de los sistemas de liberación de insulina. Refleja la configuración de los estudios y replica el flujo de datos entre un sujeto con diabetes tipo 1, un dispositivo de medida continua de glucosa, un algoritmo de control y una bomba de insulina.

El principal objetivo es obtener un paciente virtual que permita conocer los niveles de glucosa e insulina en plasma tras la perturbación que origina la ingesta de carbohidratos y la inyección de insulina subcutánea. El paciente virtual se representa con un modelo de regulación de glucosa y sus parámetros. En este caso el modelo matemático utilizado para realizar la simulación de un paciente es el de Cambridge, que se explicará más adelante.

Actualmente existen diversos simuladores que pueden ser utilizados en el desarrollo y evaluación de los dispositivos. Sin embargo, se busca disponer de un simulador que permita la comunicación entre dos ordenadores. De este modo un ordenador se encarga de realizar los cálculos necesarios para conocer cuál sería el comportamiento del paciente y el otro permite introducir los datos de entrada y visualizar los resultados.

1.2. Características del proyecto

La aplicación desarrollada va a ser utilizada en el Departamento de Ingeniería de Sistemas y Automática de la Universidad Politécnica de Valencia con el objetivo de facilitar la investigación para obtener en un futuro el páncreas artificial. Por este motivo la definición del proyecto y sus objetivos están condicionados por sus necesidades.

Como la línea de investigación permanece abierta surgirán nuevos enfoques y se obtendrán nuevas soluciones. La aplicación debe ser capaz de adaptarse de modo que en un futuro puedan utilizarse nuevos modelos matemáticos para realizar la simulación y puedan utilizarse nuevos sistemas de comunicación que faciliten el uso del simulador.

Al diseñar el paciente virtual se tienen en cuenta tres eventos:

- Las entradas del modelo: la ingesta de carbohidratos en las comidas, los bolos de insulina que se administrarán de forma subcutánea y la insulina basal.
- El modelo: el comportamiento del metabolismo del paciente con diabetes tipo 1 a la perturbación originada por la introducción de carbohidratos e insulina en el organismo.
- El análisis de resultados: visualización de la evolución temporal de glucosa e insulina en plasma.

Los tres eventos mencionados se agrupan en dos programas. Por un lado, se realiza la simulación del comportamiento del paciente que consiste en la resolución del sistema de ecuaciones del modelo y, por otro, la introducción de los datos de entrada y la visualización de los resultados obtenidos. El programa utilizado para implementar el paciente virtual es Matlab, que ofrece múltiples posibilidades para la resolución del modelo de ecuaciones diferenciales ordinarias que se plantea.

El paciente virtual se diseña de modo que cada uno de los programas se encuentre disponible en un ordenador. Los ordenadores se comunican entre ellos para realizar la simulación del paciente virtual. En este caso la comunicación se realiza a través del puerto serie con un cable USB. Para que los ordenadores compartan e intercambien datos es necesario establecer una comunicación entre los mismos. Para poder realizar esto es necesario establecer un protocolo de comunicación. Esto es, una serie de normas que utilizan los equipos informáticos para gestionar sus diálogos en los intercambios de información. Dos equipos de diferentes marcas se pueden comunicar sin problemas en el caso de que usen el mismo protocolo de comunicación.

Para introducir los datos de entrada se diseña una interfaz gráfica que permita indicar fácilmente en qué instante se realiza la ingesta de carbohidratos, los gramos ingeridos, la dosis de insulina administrada y la insulina basal del paciente. En el momento en el que se completan los datos, la información se enviará al otro ordenador para que se realice la simulación. Una vez obtenidos los resultados, se enviarán de nuevo al primer ordenador para que se pueda observar la evolución temporal de los carbohidratos ingeridos y la glucosa e insulina en plasma. En la Figura 1.1 se muestra un esquema del flujo de información del proyecto.

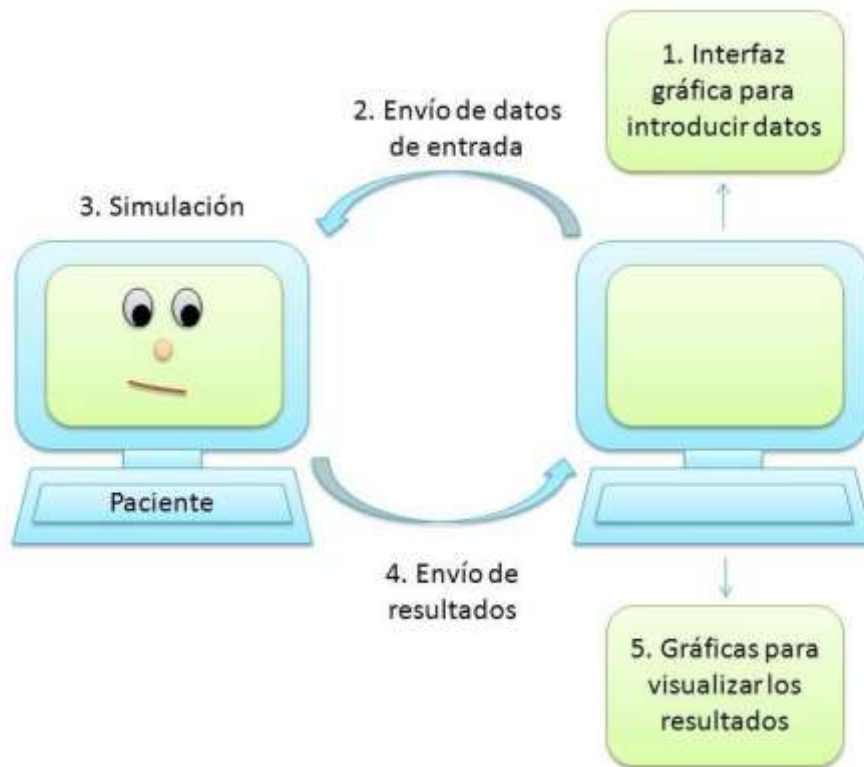


Figura 1.1. Esquema del flujo de información del proyecto

Capítulo 2

Motivación y justificación

2.1. Motivación del proyecto

Igual que con el diseño de cualquier sistema complejo de ingeniería, la simulación por ordenador puede proporcionar información muy valiosa acerca de la seguridad y las limitaciones de un algoritmo de control, así como acelerar significativamente su desarrollo. El diseño de controladores, basándose puramente en los resultados empíricos de los ensayos clínicos, puede requerir numerosas iteraciones antes de que se obtengan resultados satisfactorios. Además, la realización de ensayos, sin conocimiento previo del comportamiento que se espera del controlador, puede introducir riesgos innecesarios para el paciente. Por otro lado, los ensayos clínicos requieren de la aprobación de un comité ético.

La posibilidad de realizar pruebas preclínicas en un ordenador que disponga de un simulador acelera el desarrollo al tiempo que facilita el ahorro de recursos, la optimización preclínica y reemplaza las pruebas con animales. Además, se puede probar su estabilidad, detectar escenarios donde se comporta de forma ineficiente y evaluar situaciones extremas que no se pueden realizar con pacientes debido a razones éticas como fallos del hardware, mal funcionamiento de los sensores, oclusión de la bomba, ausencia o excesivos bolos de insulina y significativos errores en la cuenta de carbohidratos ingeridos.

Sin embargo, es importante remarcar que la validación por medio de simuladores no garantiza el rendimiento en el proceso real. Por ello, la validación en un ordenador es sólo un prerrequisito para la realización de los ensayos clínicos, pero no un sustituto.

2.2. Justificación del proyecto

La realización del presente proyecto se justifica desde dos perspectivas distintas: académica y tecnológica.

2.2.1. Interés académico

El proyecto se desarrolla bajo el contexto de un Trabajo Final de Grado, requisito necesario para la obtención del Grado en Ingeniería en Tecnologías Industriales impartido por la Escuela Técnica Superior de Ingenieros Industriales de la Universidad Politécnica de Valencia.

Desde este enfoque se distinguen varios motivos que justifican la realización:

- Modelización y simulación de sistemas fisiológicos. En primer lugar, se toma contacto con el área de la biomedicina y su industria, poniendo de manifiesto el papel de la ingeniería de control. Se estudian conceptos de modelado de procesos reales más complejos a los vistos a lo largo del grado, como es la modelización de un paciente.
- Desarrollo de Software. La realización del proyecto permite ampliar y aplicar los conocimientos de programación aprendidos. Además, se estudian y aplican técnicas y herramientas para el diseño y el desarrollo de aplicaciones de software.

2.2.2. Interés tecnológico

Desde el punto de vista tecnológico hay dos motivos principales que justifican la realización del proyecto:

- Laboratorio virtual remoto. Durante los últimos años el uso de las nuevas tecnologías en la educación e investigación ha experimentado un extraordinario crecimiento. El acceso a través de internet a cualquier tipo de documentación ha facilitado el aprendizaje y la investigación.

Desde el punto de vista de la investigación este proyecto facilita su desarrollo puesto que permite que cualquier persona del departamento pueda realizar las pruebas sin necesidad de disponer del simulador en el propio equipo, pudiendo acceder fácilmente al mismo.

Si se facilita el acceso al simulador a personas ajenas al departamento por ejemplo a través de internet, éste puede ser utilizado como herramienta de aprendizaje del comportamiento del metabolismo de un paciente diabético. Esto es interesante para los estudiantes de endocrinología que pueden ver los efectos de los distintos tratamientos que estudian sin necesidad de acceder a datos de estudios reales. Además, es interesante para los pacientes y familiares que pueden ver los efectos que tendrían ante distintas situaciones, pudiendo prevenir aquellas que resulten perjudiciales.

- Simulación Hardware in the loop (HIL). En este tipo de simulación la única diferencia existente con la implementación clínica del sistema es la sustitución del sujeto por un modelo matemático, en este caso de la dinámica del sistema insulina-glucosa de un paciente con diabetes mellitus tipo 1. De este modo se obtiene un simulador lo más similar posible al caso real, de modo que se puede comprobar el correcto funcionamiento de todos los dispositivos que intervienen en el proceso de control del sistema glucorregulador.

En el presente proyecto se ha buscado aproximarse lo máximo posible a este tipo de simulación y por ello se ha implementado el simulador en un ordenador diferente. Sin embargo, no se dispone de una bomba de insulina o un medidor continuo de glucosa por lo que la sustitución del ordenador por un paciente no será suficiente para realizar un ensayo clínico.

Capítulo 3

Diabetes

3.1. Metabolismo de la glucosa

La glucosa es la principal fuente de energía para el metabolismo de las células. En particular, el metabolismo del cerebro es muy dependiente del aporte de glucosa desde el torrente sanguíneo. Por este motivo, en el cuerpo humano existen diferentes sistemas de regulación del metabolismo de la glucosa en los que intervienen varias hormonas y órganos. Las hormonas regulan la producción y absorción a través de los tejidos de glucosa para mantener las concentraciones de glucosa en plasma en el rango adecuado. Sin embargo, existen algunas condiciones patológicas que afectan al metabolismo de la glucosa. La más conocida y frecuente es la Diabetes Mellitus.

El cuerpo humano suele mantener los niveles de glucosa en plasma en un rango muy estrecho (aproximadamente 65-140 mg/dl) [1]. Esto se debe al equilibrio existente entre la glucosa disponible y su utilización por los tejidos (músculos, tejido adiposo y cerebro). La glucosa presente en el organismo proviene de dos fuentes distintas. Por una parte, la glucosa exógena (se genera en el exterior del organismo) que se obtiene de la absorción intestinal tras la ingesta de carbohidratos y, por otra, la producción endógena (proviene del interior del organismo) que se produce en el hígado y, en menor medida, en el riñón.

Cuando se realiza una ingesta comienza el proceso de digestión que permite transformar la comida en nutrientes asimilables por el organismo. De todos los nutrientes, los más importantes en este caso son los carbohidratos. Una gran cantidad de glucosa se absorbe y se almacena en las células del hígado en forma de glucógeno. Este glucógeno se convertirá en glucosa sólo cuando el cuerpo lo necesite, por ejemplo, entre comidas o durante la noche. El resto de la glucosa que no se almacena en el hígado se distribuye, a través del sistema circulatorio, a todas las células del cuerpo para que obtengan la energía necesaria para realizar sus funciones [2].

Mantener los niveles de glucosa en el rango adecuado es crítico para la salud del paciente. Un incremento excesivo de la concentración de glucosa en plasma (hiperglucemia) es responsable de un aumento en la morbilidad y mortalidad cardiovascular en personas diabéticas en comparación con la población general. Por otra parte, evitar un descenso excesivo de la concentración de glucosa en plasma (hipoglucemia) es crítico para la supervivencia de todo el cuerpo. Esto se debe a que la glucosa provee más del 90% de la energía necesaria para el funcionamiento del cerebro y, debido a que el cerebro no es capaz de sintetizar la glucosa y sus reservas sólo son suficientes para unos pocos minutos, su función es casi totalmente dependiente del continuo suministro de glucosa desde el torrente sanguíneo [1].

Las hormonas que intervienen en el metabolismo de la glucosa (como insulina, glucagón, epinefrina o cortisol) están diseñadas para mantener la concentración de glucosa en plasma en el rango adecuado. Cuando no se realizan ingestas los niveles de glucosa disminuyen y la producción endógena de glucosa se hace necesaria [3]. Los mecanismos de control de la glucosa son complejos y los órganos que liberan glucosa en el torrente sanguíneo juegan un papel fundamental para mantener el funcionamiento del cerebro.

3.1.1. Insulina

La insulina es la principal hormona reguladora para reducir la concentración de glucosa en plasma. Su principal función es permitir la absorción de la glucosa de la sangre por los músculos, el tejido adiposo y el hígado. Además esta hormona tiene efecto en el metabolismo de muchos nutrientes, como los carbohidratos y las proteínas [4].

La insulina es secretada por el páncreas, en las células β de los islotes de Langerhans. Estos islotes son unas regiones del páncreas donde se producen las hormonas relacionadas con la homeostasis de la glucosa y el metabolismo. Hay cuatro tipos de células: células α que producen el glucagón, células β que producen la insulina y amilina, células δ que producen somatostatina y células PP que producen polipéptido pancreático [5].

En circunstancias normales, un aumento de la concentración de glucosa en plasma, como tras la ingesta de carbohidratos, está acompañado de un incremento de la secreción de insulina. Esto provoca un aumento de la concentración de la insulina en plasma que acelera la utilización de la glucosa y suprime su producción. Lo contrario ocurre cuando se produce un descenso de la concentración de glucosa en plasma, que va acompañado de un descenso de la secreción de insulina, provocando la disminución de la concentración de insulina y aumentando la producción endógena de glucosa.

Los pacientes diabéticos tienen una deficiencia de insulina por lo que no regulan adecuadamente los niveles de glucosa. Esto hace que la insulina administrada de forma exógena tenga una gran importancia en el tratamiento de la diabetes. Habitualmente, se administra de manera subcutánea, lo que retrasa el efecto de la insulina en comparación con las personas sanas que secretan la insulina directamente en el sistema circulatorio. Este retraso se debe a la lenta absorción de insulina puesto que tiene que disociarse para poder pasar a través de los tejidos al torrente sanguíneo.

3.1.2. Glucagón

El glucagón se caracteriza por oponerse a los efectos de la insulina. Su labor principal es estimular la producción de glucosa por el hígado, lo que permite mantener los niveles de glucosa en sangre en los periodos de ayuno. También compensa el aumento de absorción de glucosa por los músculos durante períodos de actividad física [3]. Por tanto, previene la disminución de la concentración de glucosa en plasma, es decir, la hipoglucemia.

Cuando la concentración de glucosa en plasma se encuentra por debajo del rango normal, la secreción de glucagón aumenta, resultando en una producción hepática de glucosa y volviendo a los niveles normales de glucosa. Esta fuente de glucosa endógena no es necesaria durante e inmediatamente después de las comidas, por lo que la secreción de glucagón se suprime.

La secreción de glucagón junto con la supresión de la producción endógena de insulina es la primera respuesta fisiológica del cuerpo para impedir la inminente hipoglucemia. Sin embargo, este sistema de regulación puede estar dañado en personas diabéticas. Esta disfunción en la homeostasis del paciente, junto con administraciones excesivas de insulina, puede causar hipoglucemias.

3.1.3. Epinefrina

La epinefrina, también conocida como adrenalina, es una hormona secretada por las glándulas suprarrenales. Un bloqueo de sus efectos provoca una hipoglucemia severa aunque otras hormonas, particularmente el glucagón, aumenten su concentración en sangre.

A diferencia del glucagón, la epinefrina tiene efecto en múltiples órganos. Inhibe la producción endógena de insulina, estimula la producción endógena de glucosa en el hígado y en el riñón y suprime la utilización periférica de glucosa.

3.1.4. Hormonas de acción lenta

La hormona del crecimiento y el cortisol son las llamadas hormonas de acción lenta. Esto se debe a que su efecto se hace evidente cuando han pasado varias horas del incremento de su concentración en la sangre. El efecto de estas hormonas sólo se aprecia pasadas tres horas de la hipoglucemia. Ambas hormonas son críticas pues una ausencia de cualquiera de las dos no se compensa con respuestas aun mayores de hormonas de acción rápida como el glucagón o la adrenalina.

3.2. Diabetes Mellitus

La Diabetes Mellitus abarca un grupo de enfermedades metabólicas cuya principal característica es la presencia de altos niveles de glucosa en la sangre. Esto se debe a la deficiencia relativa o absoluta de insulina.

Antiguamente eran capaces de reconocer los síntomas de la diabetes pero carecían de medios para tratarla efectivamente. Hasta principios del siglo XIX el método de diagnosticar la diabetes era probando la orina de los pacientes pues tiene un gusto dulce, o exponer la orina a la intemperie para ver si atraía a insectos debido a la alta concentración de glucosa [4]. No es hasta finales de siglo y principios del siglo XX cuando empiezan a desarrollarse pruebas de glucosa en sangre, pero sin tener aplicación clínica.

Hasta la aparición de la insulina algunos médicos prescribieron dietas de bajas calorías y muy pocos carbohidratos que alargaban la vida de sus pacientes diabéticos, pero sufrían de inanición. Desde el descubrimiento de la insulina en 1921 por Banting y Best [6] los avances médicos continuaron prolongando y facilitando la vida de las personas con diabetes. Otro descubrimiento importante es la existencia de dos tipos de diabetes: sensibles a la insulina (diabetes tipo 1) e insensible a la insulina (diabetes tipo 2). Los dos tipos de diabetes están caracterizados por procesos patológicos distintos lo que resulta en diferentes tratamientos.

En las últimas décadas la diabetes se ha convertido en un importante problema de salud en todo el mundo. Su impacto, especialmente en países desarrollados, ha alarmado a las organizaciones de salud en muchos países y ha provocado que se realicen mayores esfuerzos en el tratamiento de esta enfermedad. El aumento de la población combinado con el envejecimiento de la misma hace que el problema de la diabetes aumente notablemente cada año, alcanzando unas proporciones epidémicas en las próximas décadas.

Además, se debe tener en cuenta el impacto en el gasto sanitario. Este gasto no comprende sólo los tratamientos de diabetes, también todas las complicaciones causadas directamente por la enfermedad. El gasto sanitario medio de una persona diabética al menos duplica al de una persona de la misma edad y sexo sin diabetes. Este gasto varía entre países y con el tiempo.

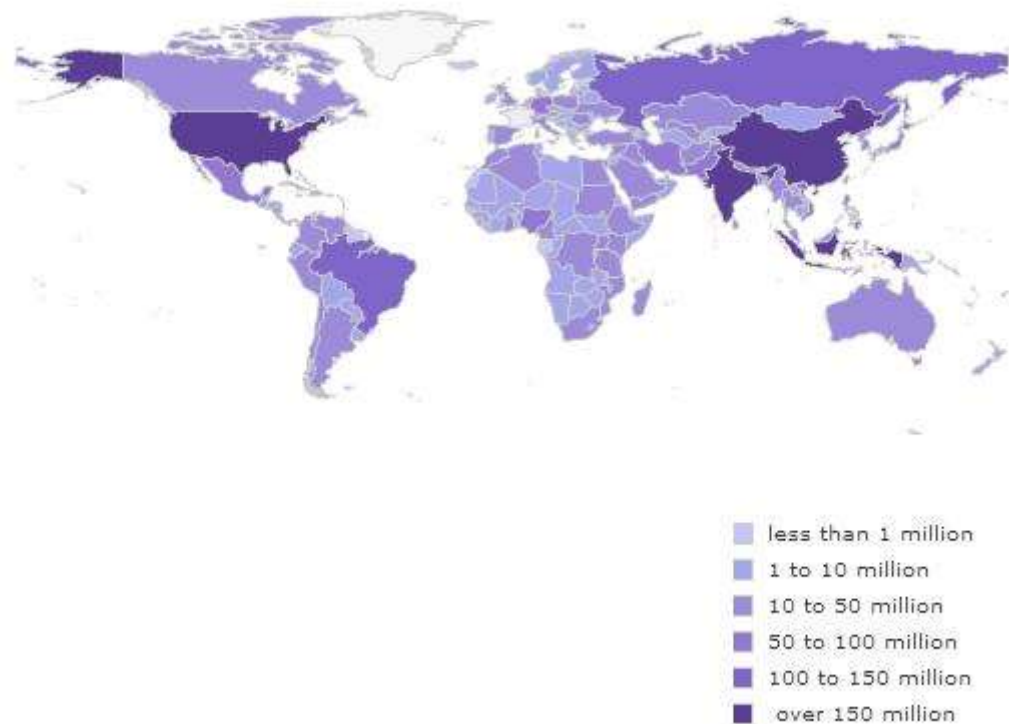


Figura 3.1. Estimaciones de población con diabetes entre 20 y 79 años en 2013 [7]

Hay dos tipos principales de diabetes mellitus. La diabetes tipo 1 se caracteriza por la destrucción autoinmune de las células β del páncreas en individuos presuntamente predispuestos, lo que produce una ausencia de secreción de insulina [8]. El tratamiento para las personas con diabetes tipo 1 consiste en administrar insulina exógena para suplir la falta de producción de insulina endógena. La diabetes tipo 2 es el resultado de dos alteraciones. Por una parte, la resistencia a la insulina, esto significa que las células tienen menor sensibilidad de lo normal a la acción de la insulina. Por otra, las células β pueden estar dañadas y la respuesta fisiológica de estas células a las comidas se ha perdido, así como su habilidad para compensar la resistencia a la insulina.

Hay otros tipos de diabetes, como las patologías provocadas por infecciones o endocrinopatías. Pueden ser de carácter temporal, como la diabetes gestacional que tiene lugar en algunas embarazadas, o permanecer de forma crónica.

3.2.1. Diabetes tipo 1

La diabetes tipo 1 está relacionada con un nivel de insulina en plasma por debajo de lo normal, lo que produce que la concentración de glucosa en sangre esté constantemente en niveles altos. Esto puede causar severos síntomas, como cetoacidosis, retinopatías o nefropatías.

Las personas sanas suelen mantener la concentración de glucosa en el estado de ayuno por debajo de los 100 mg/dl y por debajo de 140mg/dl en el periodo postprandial [5]. Durante la comida es posible mantener la concentración de glucosa gracias a la producción de insulina por el páncreas, previniendo así la hiperglucemia. Normalmente, el páncreas segrega insulina muy rápido en respuesta a la ingesta de comidas. Sin embargo, también es muy importante el rápido descenso de la misma pasados 60-90 minutos de la ingesta para prevenir la hipoglucemia en el periodo postprandial.

3.2.2. Diabetes tipo 2

La diabetes mellitus tipo 2 es la más extendida y representa el 85% de los casos en los países desarrollados. Está relacionada con diversos factores como el aumento de edad, de peso y características raciales y geográficas. Es la diabetes tipo 2 la que amenaza con convertirse en una epidemia mundial. El pico de inicio está alrededor de los 50 años y el 80% de los diabéticos tipo 2 son obesos [5].

Hay dos defectos principales que caracterizan la diabetes tipo 2:

- Disfunción de las células β . Habitualmente, el deterioro afecta inicialmente la secreción de insulina durante la comida. De hecho, la primera fase de la enfermedad es la hiperglucemia en el estado postprandial. Sin embargo, el deterioro de las células β aumenta progresivamente, causando finalmente una deficiencia absoluta de insulina
- Resistencia a la insulina. La resistencia hepática a la inulina causa una excesiva producción de glucosa, contribuyendo a acelerar la aparición de hiperglucemia. La resistencia de los músculos a la insulina conlleva una utilización ineficiente de la glucosa. Está relacionada con la obesidad y la inactividad física, ambas

relacionadas con la acumulación hepática e intramuscular de lípidos. Sin embargo, la resistencia a la insulina no provoca por sí misma la diabetes, las células β pueden ser capaces de compensar esa resistencia, por ello no todos los obesos son diabéticos. La diabetes se desarrolla cuando las células β no son capaces de compensar la resistencia a la insulina.

Existen diferentes métodos para tratar la diabetes tipo 2. Sin embargo, ningún tratamiento es capaz de impedir el deterioro de las células β por lo que cualquier paciente diabético tipo 2 se convertirá, con el tiempo, en insulino-dependiente, al que se le tratará administrándole insulina, como en el caso de los diabéticos tipo 1.

3.2.3. Tratamiento

Inicialmente se comienza a tratar a los pacientes diabéticos tipo 1 con insulina extraída de vacas y cerdos, la cual no es idéntica a la insulina humana y presenta impurezas que con el tiempo crearán anticuerpos. La única tecnología de diagnóstico para estos pacientes era la prueba de orina, pero la glucosa empieza a aparecer en la sangre únicamente cuando los niveles son muy elevados lo que hace que este procedimiento no provea información para ajustar su tratamiento. Además las jeringuillas utilizadas eran de vidrio, se esterilizaban en casa y, de vez en cuando, era necesario afilar las agujas.

En los años 70 se consiguió mejorar las técnicas de purificación de la insulina, reduciendo radicalmente los problemas debidos a las impurezas que se encontraban hasta el momento. Posteriormente se logró sintetizar en los laboratorios insulina humana utilizando la bacteria *E. Coli* con modificaciones genéticas. A principios de los 80 la insulina humana fue introducida en el mercado sustituyendo a la insulina animal, aunque inicialmente no se encontraron mejoras en el control de la glucosa, el aspecto positivo fue la tecnología utilizada en su producción. Se consiguió sintetizar insulina humana con técnicas genéticas, favoreciendo el desarrollo de análogos sintéticos de insulina en años posteriores. Actualmente se dispone de insulina de acción rápida y lenta. Las insulinas de acción lenta permanecen en sangre hasta 24 horas y se utilizan para sustituir los niveles de insulina en los periodos entre comidas. Las insulinas de acción rápida tiene un pico de absorción entre 45-50 minutos y su permanencia en sangre es de entre una y dos horas, se utilizan para replicar la insulina en el periodo postprandial. Esto ha permitido que los pacientes diabéticos tengan mayor flexibilidad, ya no necesitan suministrarse insulina horas antes de cada comida, pueden hacerlo justo antes.

Independientemente del sistema de medición de glucosa o la forma en que esta se administre el objetivo básico de todas las técnicas de control es replicar el comportamiento del metabolismo de una persona sana en un paciente con diabetes

mellitus tipo 1. En el estado de ayuno y tras el estado de absorción, la tasa de secreción de insulina está regulada por los niveles de glucosa en plasma. Esto se conoce como glucosa basal o estado basal, que modula la producción de insulina hepática para compensar exactamente la utilización de glucosa en las células y mantener los niveles de glucosa en un estrecho rango. Para administrar la insulina lo mejor es basarse en los niveles de glucosa en sangre y administrar la insulina directamente en el torrente sanguíneo. Sin embargo, por razones prácticas se suele administrar de manera subcutánea. Esto introduce un retraso desde que se realiza la inyección hasta que se aprecia el efecto, que representa el tiempo necesario para la absorción de la insulina desde el tejido subcutáneo hasta la sangre.

Las bombas de infusión continua de insulina subcutánea se han convertido en un dispositivo tecnológico que libera a los pacientes diabéticos de las múltiples inyecciones diarias de insulina. Estas bombas son más cómodas y precisas, en cuanto a la dosificación, y previenen mejor los niveles de glucosa cuando se utilizan correctamente.

El principal inconveniente de estas bombas es que un fallo en el suministro puede causar una hiperglucemia en las horas siguientes debido a que utilizan insulina de acción rápida. Los fallos que pueden provocar este problema son obstrucciones en la infusión, fugas, pérdida de potencia a la insulina, o mal funcionamiento de la bomba.

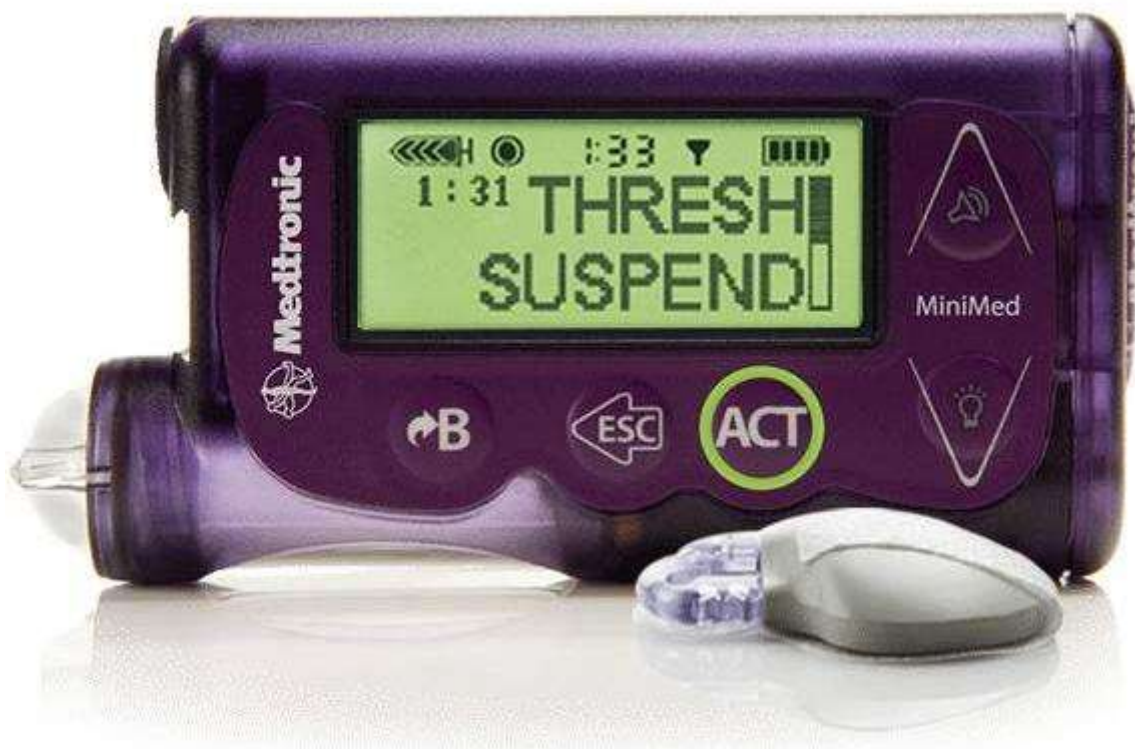


Figura 3.2. Bomba de insulina comercial MiniMed 530G de Medtronic [9]

Este tipo de tratamientos requieren también una medida de la glucosa más eficiente. A principios de la década de los 80 comienza a utilizarse un aparato que utiliza colorimetría para medir la glucosa en sangre. En 1986 aparece el primer medidor de glucosa con un biosensor. Hoy en día estos aparatos para medir la concentración de glucosa en sangre son indispensables para los pacientes diabéticos. Posteriormente aparecen los sistemas de medida continua de glucosa, CGMS (*Continuous Glucose Measurement System*) que almacenan los datos de glucosa en sangre (por medio de un sensor subcutáneo) cada cinco minutos para su posterior análisis clínico [4]. En 2005 Medtronic introduce el sistema Guardian que da la oportunidad al paciente de conocer en tiempo real los niveles de glucosa y actuar sobre ello con una acción de control, que es la infusión de insulina.

Esto define un sistema de control y hay varias leyes que pueden ser aplicadas para obtener un control de la glucosa en sangre más estricto tratando de simular lo más fielmente posible la producción de insulina de una persona sana. Este comportamiento no se puede lograr actualmente mediante la infusión subcutánea de insulina. Sin embargo, la presencia de datos de monitorización continua de glucosa, incluso con la inexactitud que presentan, abre la posibilidad de diseñar nuevas estrategias de control que estabilicen la glucemia en pacientes diabéticos [5].

3.2.4. Complicaciones de la enfermedad

Si se registran niveles elevados de glucosa durante periodos largos pueden aparecer complicaciones severas. Por este motivo, ambos tipos de diabetes deben ser controladas para evitar complicaciones en otros órganos.

Para los pacientes diabéticos tipo 1 las dos principales complicaciones son las siguientes [2]:

- Cetoacidosis. La cetoacidosis diabética es un estado de una diabetes severa no controlada, causada por una deficiencia de insulina y requiere un tratamiento de emergencia con insulina y fluidos. Se suele encontrar en pacientes jóvenes a los que todavía no se les ha diagnosticado la diabetes tipo 1 o en pacientes diabéticos tipo 2 durante graves infecciones o enfermedades.
- Hipoglucemia. Es un efecto secundario común debido al tratamiento con insulina. La hipoglucemia aparece cuando se registran niveles elevados de insulina en plasma o cuando se incrementan los efectos de la misma. El cerebro depende del aporte continuo de glucosa y una interrupción en el suministro durante unos minutos puede causar daños en el sistema nervioso central, pérdida de conciencia o incluso el coma.

Existen otras complicaciones causadas por un elevado nivel de glucosa en sangre durante largos periodos de tiempo (hiperglucemia) tanto en los diabéticos tipo 1 como 2. Las más frecuentes son:

- Problemas cardiovasculares. Los diabéticos tiene el mismo riesgo de mortalidad por un ataque al corazón como las personas no diabéticas que ya han sufrido uno. Este riesgo se multiplica por tres cuando el paciente ya ha sufrido un ataque. Los altos niveles de glucosa combinados con el colesterol favorecen la aparición de arteriosclerosis que es la base para el desarrollo de problemas de corazón.
- Complicaciones oculares. Afectan principalmente a los vasos oculares de la retina. Los pacientes diabéticos aceleran la aparición de cataratas o glaucoma. Alrededor del 5% de los pacientes están totalmente ciegos, siendo la diabetes la principal causa de la ceguera en muchos países.
- Complicaciones de riñón. La nefropatía clínica ocurre en el 30-40% de los casos, provoca la presencia de proteínas en la orina que con el tiempo afectan a las funciones del riñón y puede terminar en una enfermedad renal. La diabetes mellitus es la tercera causa de enfermedad renal terminal.
- Neuropatía. Es una complicación muy común. Los daños en los nervios están causados por unos niveles elevados de glucosa en sangre y una disminución del flujo sanguíneo. Los riesgos aumentan si los niveles de glucosa no se controlan adecuadamente.
- Disfunción gastrointestinal. La más común de estas enfermedades es la gastroparesia. Se trata de un desorden en el estómago que provoca que éste tarde mucho en vaciarse. Esto causa problemas como el excesivo crecimiento de bacterias.

3.3. Páncreas artificial

A pesar de la disponibilidad de nuevas preparaciones de insulina con mejores resultados en el control de la glucosa, las terapias actuales basadas en el suministro de insulina exógena no consiguen normalizar los niveles de glucosa de forma eficiente. Las nuevas tecnologías desarrolladas, como la monitorización continua de glucosa (CGM: *Continuous Glucose Monitoring*), y la infusión continua de insulina, han favorecido el desarrollo del páncreas artificial. Esto es, un dispositivo capaz de medir en tiempo real los niveles de glucosa en plasma y en función de los mismos realizar un control automático de infusión de insulina, sin necesidad de que el paciente intervenga en el proceso (control en lazo cerrado).

En las últimas décadas ha habido un gran progreso en el desarrollo de sensores continuos de glucosa. Los estudios se han centrado en la piel como un candidato apropiado para la medición directa de glucosa. De hecho, el tejido subcutáneo es fácilmente accesible para la implantación del sensor y la medida de glucosa en el fluido intersticial, con menos problemas en comparación a los que aparecen al realizar la medición intravenosa. Puesto que el resultado obtenido en caso de alojar el sensor en el tejido subcutáneo es de la glucosa intersticial y se desea la glucosa en plasma, tras el filtrado, se incorporan algoritmos de estimación de la glucosa en plasma calibrados mediante medidas capilares. Existe un retraso fisiológico entre ambas señales que es lo que se identifica como una de las fuentes de inexactitud en los monitores [4].

Las bombas de infusión de insulina subcutánea incluyen cada vez funciones más avanzadas para ayudar al paciente diabético en la dosificación óptima de la dosis. Ya existen en el mercado sistemas integrados bomba de insulina – monitor continuo de glucosa, incluso con funciones como la desconexión automática de la bomba en caso de previsión de hipoglucemia, considerado como el primer paso hacia el páncreas artificial. Sin embargo, aunque la tecnología de las bombas está bastante desarrollada es necesario mejorar su fiabilidad para ser integrada en un páncreas artificial.

Se han desarrollado varios prototipos de páncreas artificial pero todavía existen varios desafíos:

- Hacer frente a grandes perturbaciones que afectan al sistema, como las comidas, el estrés o el ejercicio.
- Robustez de cara a la gran variabilidad de comportamiento fisiológico del paciente.
- Mejorar la precisión y fiabilidad de los monitores continuos de glucosa.
- Mejorar la seguridad de las bombas de insulina y la detección de fallos.
- Corregir la respuesta lenta de los controladores debido a retrasos en el lazo de control.

1.3.1. El problema del control

El primer experimento conducido a cerrar el lazo de control en un paciente con diabetes fue en 1964 [10]. Fue el primer ensayo para controlar la glucosa de la sangre con señales continuas de glucosa en plasma. El algoritmo de control se trataba de un sistema de apagado y encendido con una supervisión constante de los doctores y el control de la

acción (insulina) se administraba de forma intravenosa, que minimiza el retraso relacionado con el transporte de insulina en los tejidos.

Diez años más tarde Albisser [11] y Pfeiffer [12] desarrollan el primer dispositivo comercial que emula un páncreas artificial. El dispositivo, conocido como el Biostator, disponía de un algoritmo de control complejo y estaba diseñado para prevenir la hiperglucemia en el periodo postprandial. Sin embargo, se trataba de una máquina de grandes dimensiones que precisaba de constante supervisión y no estaba diseñada para un uso independiente.

El uso de monitores continuos de glucosa junto con la infusión continua de insulina subcutánea solucionan el principal problema del anterior dispositivo, pero añade nuevos retrasos en el lazo de control, lo que hace que sean necesarios algoritmos de control más complejos.

En función del tipo de instrumentación empleada se puede hablar de varias rutas para cerrar el lazo [4], [13]:

- Medición de glucosa intravenosa (i.v.) e infusión de insulina intravenosa. Se trata de la opción más invasiva pero también la que representa el menor retraso entre la medición de glucosa y su respuesta (aproximadamente 30 minutos). Sólo es factible en entornos como la unidad de cuidados intensivos debido al acceso vascular que requiere. Un ejemplo de este tipo de instrumentación es el Biostator.
- Medición de glucosa intravenosa e infusión de insulina intraperitoneal (i.p.). Se utilizan bombas de insulina implantables en la cavidad abdominal, que tiene muchas de las ventajas de una ruta intravenosa total. Sin embargo, sigue siendo invasiva, requiere cirugía para la implantación de la bomba e introduce un retraso en la absorción de insulina.
- Medición de glucosa subcutánea (s.c.) e infusión de insulina intraperitoneal. El monitor es mínimamente invasivo. La bomba es implantable, requiere cirugía e introduce un retraso en la absorción de insulina.
- Medición de glucosa subcutánea e infusión de insulina subcutánea. Este sistema es el más adecuado para tratamientos largos puesto que es el menos invasivo de los mencionados. El inconveniente es el retraso, que es el mayor de todos a pesar de la utilización de insulinas de acción rápida. En este caso hay retrasos debidos a la absorción de insulina, la acción de la insulina, la cinética de la glucosa intersticial (G.I.) y el tiempo de transporte debido a los sistemas de monitorización.

En la figura 3.3 se muestra el retraso presente en las posibles rutas para el páncreas artificial.

El retraso fisiológico supone un desafío importante para el control, ya que una sobreactuación excesiva llevaría al paciente a hipoglucemia debido a la acumulación de insulina, ya que no existe una forma aplicable en la práctica de subir la glucemia más allá de la eliminación natural de la insulina, el consumo de alimentos o el glucagón (su uso es controvertido, entre otras razones, por el riesgo de agotamiento de las reservas de glucosa en el hígado). El problema de la sobreactuación es especialmente significativo en la compensación de ingestas, donde los errores son grandes.

Otro de los problemas a los que debe hacer frente el control es la variabilidad. No sólo se trata de la variabilidad en el comportamiento entre pacientes diabéticos, si no también en un solo paciente. Los requerimientos de insulina varían en función de la hora del día y también influyen factores como el estrés o el ejercicio y el controlador debe ser capaz de compensar estos cambios.

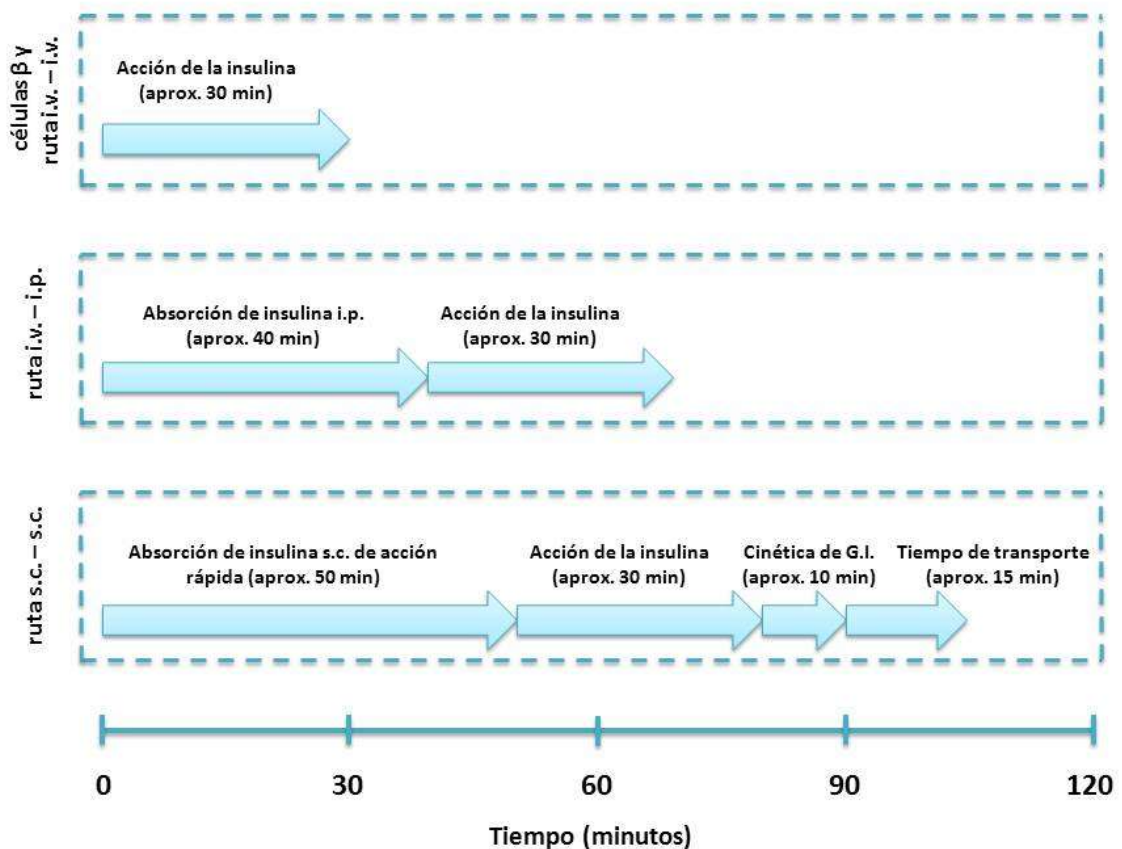


Figura 3.3. Retrasos relacionados con los distintos sistemas del páncreas artificial [13]

Capítulo 4

Control postprandial

Los métodos para el control postprandial se pueden clasificar en tres bloques dependiendo de la naturaleza del lazo de control:

- Infusión de insulina sin anuncio de comidas. Este es la forma de cerrar el lazo de control supone que no haya ninguna intervención del paciente. Algunas variaciones de este método pueden incluir predictores o alarmas.
- Infusión de insulina con anuncio cualitativo de comida. En este caso el paciente informa al controlador de que va a realizar una comida pero sin darle más información acerca del evento.
- Infusión de insulina con anuncio cuantitativo de comida. En este caso el paciente informa al controlador de que va a realizar una comida y, además, da cierta información como la cantidad de carbohidratos de la comida.

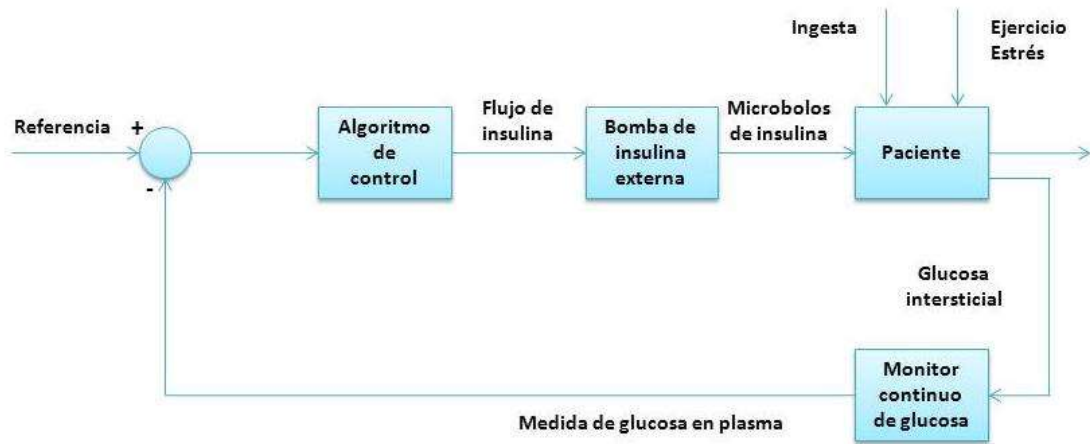


Figura 4.1. Bucle de control básico [4]

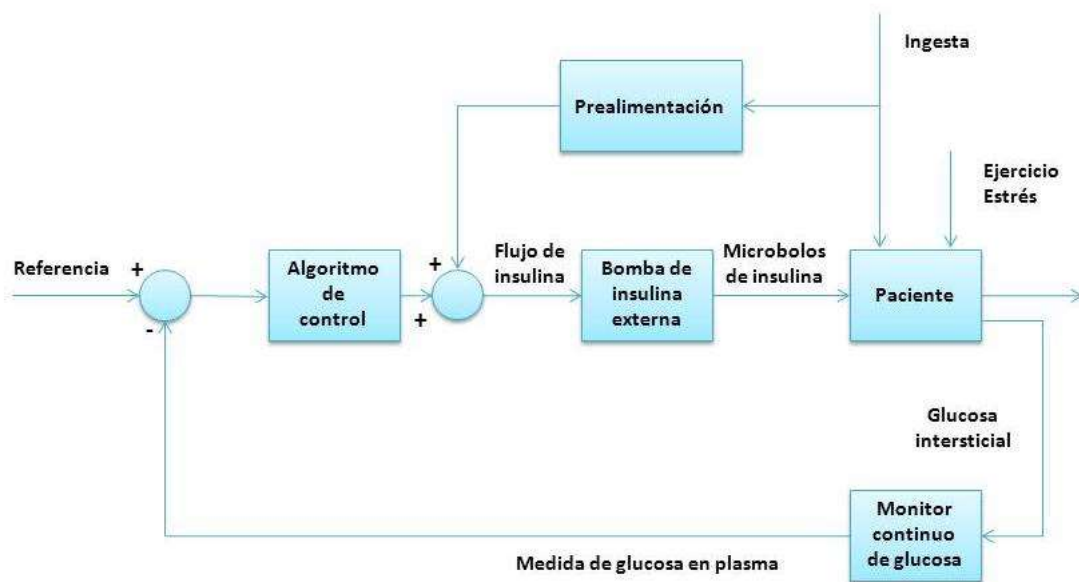


Figura 4.2. Bucle de control con anuncio de comidas [4]

4.1. Algoritmos de control

El controlador ideal para un paciente diabético debería limitar la infusión de insulina, minimizar los eventos de hipoglucemia y ser robusto frente a la variabilidad intrapaciente y fallos en la medida [4]. Debe operar adecuadamente tanto en el estado de ayuno como en el estado postprandial y adaptarse a la actividad diaria del paciente. El controlador no puede ser excesivamente complejo y debe poder ajustarse de acuerdo con la práctica clínica.

Durante los últimos años se han propuesto una gran variedad de algoritmos de control que atacan el problema con diferentes enfoques. A continuación se presentan los principales algoritmos de control [4], [5]:

- Control PID. El controlador Proporcional-Integral-Derivativo (PID) es el clásico enfoque de control. Es el algoritmo de control más ampliamente implementado en aplicaciones industriales debido a su simplicidad y el hecho de que los algoritmos para el controlador se han desarrollado durante décadas. La ecuación del algoritmo es la siguiente:

$$IRR = K_p \cdot (G - G_t) + K_i \int (G - G_t) + K_d \frac{\partial G}{\partial t} \quad (4.1)$$

Donde IRR representa la tasa de infusión de insulina (*Insulin Infusion Rate*), K_p , K_d y K_i son los parámetros del controlador, G es la medida de glucosa y G_t es la glucosa deseada.

Debido a su simplicidad el PID sirve de estructura base de un controlador que, con otros métodos, puede convertirse en uno que emule de forma más exacta el suministro de insulina de las células β del páncreas.

- Control predictivo. Esta metodología está basada en los modelos de control predictivos (MPC: *Model Predictive Control*). Se trata de una filosofía más compleja basada en métodos de optimización para encontrar la entrada óptima que permite que la variable de control alcance la referencia deseada en un determinado horizonte de predicción.

El sistema se reduce a una sola salida medida, la concentración de glucosa, y una sola variable manipulada, el flujo de infusión de insulina. Una de las principales ventajas del esquema de control MPC basado en optimización es la facilidad para incorporar de manera natural las restricciones en la infusión de insulina que recomienda la práctica clínica.

4.2. Situaciones de control

Se puede realizar una clasificación del páncreas artificial de acuerdo a la situación del paciente durante el control o a la población a la que el sistema pretende controlar:

- Diabetes temporal en mujeres embarazadas. La diabetes gestacional ha sido estudiada desde el punto de vista del páncreas artificial en diversas ocasiones. Se ha demostrado la efectividad de los monitores continuos de glucosa en mujeres embarazadas con diabetes [14] y se han usado algoritmos de control basados en modelos diabéticos para planear el tratamiento de insulina [15]. Sin embargo, no se han desarrollado controladores específicos para este caso particular.
- Control de pacientes críticos. Los pacientes críticos son monitorizados con detalle, no sólo la glucosa, también otras medidas importantes de funciones del cuerpo, que pretenden ser tan controladas como sea posible. El controlador diseñado por el grupo de Cambridge es un ejemplo de controlador utilizado para mantener la homeostasis de la glucosa en la unidad de cuidados intensivos [16].
- Control de ayuno durante la noche. El periodo de la noche normalmente está relacionado con una reducción natural de la sensibilidad a la insulina, esto se traduce como un incremento en la concentración de glucosa en plasma. El grupo de Cambridge, por ejemplo, ha logrado un gran éxito controlando este tipo de comportamiento con monitorización continua de glucosa y controladores MPC, ambos tanto de manera subcutánea como intravenosa [17].
- Control postprandial. La principal perturbación del control de la glucosa en sangre es la absorción de glucosa en el intestino. El control postprandial busca controlar el fuerte incremento de la glucosa en sangre tras la comida sin poner en riesgo la salud del paciente.

Capítulo 5

Modelado del sistema glucorregulador

5.1. Introducción

Los modelos matemáticos para describir la fisiología del sistema glucorregulador no son lo suficientemente exactos lo cual supone un gran problema para el control de la glucosa.

El principal objetivo de realizar modelos matemáticos de pacientes con diabetes tipo 1 es ser capaces de reproducir su metabolismo desde un punto de vista clínico. Para el caso del desarrollo del páncreas artificial, los modelos deben ser útiles desde el punto de vista del control. La primera distinción que se debe hacer es si se trata de un modelo fenomenológico o de comportamiento [18].

- Modelo fenomenológico: basado en un conocimiento físico. En el caso del páncreas artificial, se trata de un modelo basado en principios fisiológicos. En este tipo de modelo sólo se utilizan los datos para ajustar los parámetros, mientras que la estructura viene determinada por el proceso en sí mismo. Casi todos los modelos propuestos para modelar el metabolismo de los pacientes diabéticos son fenomenológicos, incluso los más simples.
- Modelo de comportamiento: basado en el comportamiento observado sin un conocimiento previo del proceso. Este tipo de modelos se centran en la reproducción de los datos ajustados, sin importar el proceso real que tratan de imitar.

5.2. Procesos fisiológicos del sistema insulina-glucosa

Los modelos fisiológicos del sistema de insulina-glucosa para la diabetes tipo 1 suponen tres subprocesos principales [18]:

- Modelo de absorción de insulina. Este modelo representa la forma en que la insulina exógena se introduce en el organismo. La absorción de insulina depende del tipo de insulina que se esté utilizando en la terapia y de la forma seleccionada para la administración de la misma. La insulina inyectada en el tejido subcutáneo presenta un retraso hasta que llega a la sangre en comparación a la secreción de insulina generada por el páncreas.
- Modelo de absorción de glucosa. Este modelo incluye los procesos de ingestión, digestión y absorción de glucosa y otros nutrientes desde el intestino hasta la sangre. La composición nutricional de la comida influye en el proceso de digestión y puede ralentizar el flujo de carbohidratos a través del intestino.
- Modelo de regulación de glucosa. Este modelo representa la regulación endógena de glucosa. Incluye todos los efectos que pueden afectar a la concentración de glucosa del organismo, como la transformación que tiene lugar en el hígado de glucógeno a glucosa, el consumo de glucosa por los tejidos o la influencia de las diferentes hormonas en la glucosa en plasma.

Los modelos de absorción de insulina y glucosa son considerados como los modelos de entrada para el sistema de regulación de glucosa ya que se trata de entradas exógenas. Las inyecciones subcutáneas de insulina o la insulina liberada por la bomba son los únicos métodos de control que se utilizan contra la perturbación que genera la ingesta de carbohidratos.

5.3. Selección del modelo para el simulador

A lo largo de los años se han desarrollado complejos modelos de absorción de glucosa incrementado la exactitud para describir el comportamiento fisiológico de absorción digestiva e intestinal. En el presente proyecto sólo se va a tener en cuenta uno de los modelos desarrollados hasta ahora, el modelo endógeno de Cambridge. Se ha seleccionado este modelo por los siguientes motivos:

- El modelo se basa fisiológicamente en los pacientes diabéticos y describe muchas de las dinámicas de los subsistemas de insulina y glucosa
- La validación de las predicciones de este modelo han sido demostradas por comparación con un estudio clínico con pacientes diabéticos tipo 1.
- Fue el primer modelo utilizado para un estudio en lazo cerrado utilizando monitorización continua de glucosa, proporcionando los cimientos del primer algoritmo para el páncreas artificial
- Ha sido probado para simular la variabilidad inter e intrapaciente, considerándolo en los parámetros publicados.

5.4. Modelo de Cambridge

A continuación se explican los tres subprocesos del modelo fisiológico del sistema insulina-glucosa propuesto por el grupo de Cambridge [18], incluyendo las ecuaciones que definen el modelo.

5.4.1. Modelo de absorción de glucosa

El grupo de Cambridge propone un modelo muy simple para la absorción de glucosa, donde el sistema gastrointestinal está modelado con dos compartimentos idénticos con la misma velocidad de transferencia. Las ecuaciones del subsistema son:

$$\dot{g}_1 = a_g \cdot u_{cho} - \frac{g_1}{t_{maxG}} \quad (5.1)$$

$$\dot{g}_2 = \frac{g_1}{t_{maxG}} - \frac{g_2}{t_{maxG}} \quad (5.2)$$

Donde:

- g_1 y g_2 son los compartimentos de transición para la glucosa en el proceso de digestión.
- a_g es la efectividad de la absorción de carbohidratos ingeridos, es decir, la porción de carbohidratos de los que se han ingerido que llegan al sistema circulatorio.
- u_{cho} es la cantidad de carbohidratos ingeridos en mmol min^{-1} .
- t_{maxG} es el tiempo máximo de absorción de los carbohidratos. Este parámetro regula la velocidad de transferencia entre compartimentos.

5.4.2. Modelo de absorción subcutánea de insulina

Este modelo muestra un flujo simple de insulina a través de dos compartimentos en el torrente sanguíneo y sólo considera la eliminación de insulina de la sangre. Las constantes de flujo entre compartimentos son las mismas. Las ecuaciones del modelo se muestran a continuación.

$$\dot{s}_1 = u_{ins} - \frac{s_1}{t_{maxI}} \quad (5.3)$$

$$\dot{s}_2 = \frac{s_1}{t_{maxI}} - \frac{s_2}{t_{maxI}} \quad (5.4)$$

$$\dot{I} = \frac{u_i}{v_i} - k_e \cdot I \quad (5.5)$$

Donde:

- s_1 y s_2 son dos compartimentos para la absorción de insulina subcutánea.
- I es la concentración de insulina en plasma (mU L^{-1}).
- u_{ins} es el flujo de la bomba (mU min^{-1}).
- k_e es la tasa de eliminación de insulina (min^{-1}).
- u_i es la tasa de infusión de insulina y se define como:

$$u_i = \frac{s_2}{t_{maxI}} \quad (5.6)$$

5.4.3. Modelo del sistema glucorregulador [8]

El modelo endógeno de Cambridge incluye un compartimento por cada acción de la insulina y hace referencia a cada una de estas acciones sobre diferentes fenómenos con su efecto final sobre la glucosa en plasma. Hay tres influencias consideradas:

- La insulina incrementa el flujo de glucosa de la sangre a los tejidos.
- La insulina aumenta la captación de glucosa por los músculos y el tejido adiposo.
- La insulina inhibe la producción de glucosa en el hígado.

Estas influencias se representan en el modelo como compartimentos virtuales. En la Figura 5.1 se muestra la relación entre los compartimentos virtuales que representan las acciones de la insulina, la insulina en plasma actual y los compartimentos de la glucosa. x_1 , x_2 y x_3 representan las acciones de la insulina, Q_1 es la masa de glucosa en el compartimento accesible y Q_2 es la glucosa presente en el compartimento no accesible.

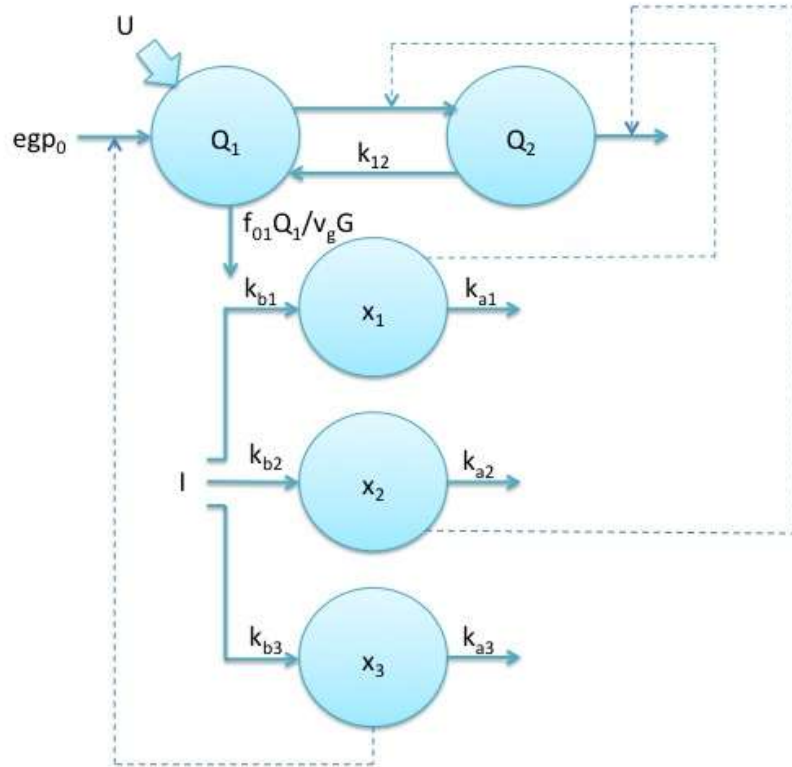


Figura 5.1. Estructura por compartimentos del modelo endógeno de Cambridge

Las ecuaciones que representan el modelo son las siguientes:

$$\dot{x}_1 = -k_{a1} \cdot x_1 + k_{b1} \cdot I \quad (5.7)$$

$$\dot{x}_2 = -k_{a2} \cdot x_2 + k_{b2} \cdot I \quad (5.8)$$

$$\dot{x}_3 = -k_{a3} \cdot x_3 + k_{b3} \cdot I \quad (5.9)$$

$$\dot{Q}_1 = -\left(\frac{f_{02}}{v_g \cdot G} + x_1\right) \cdot Q_1 + k_{12} \cdot Q_2 - F_r + u_g + e_{gp} \quad (5.10)$$

$$\dot{Q}_2 = x_1 \cdot Q_1 - (k_{12} + x_2) \cdot Q_2 \quad (5.11)$$

Donde:

- x_1 es la acción de la insulina en el transporte/distribución de glucosa.
- x_2 es la acción de la eliminación de glucosa.
- x_3 es la acción de la producción de glucosa endógena.
- Q_1 es la masa de glucosa en el compartimento accesible (directamente proporcional a la concentración de glucosa, que es la variable a controlar)
- Q_2 es la masa de glucosa presente en el compartimento no accesible.
- u_g es el flujo de absorción intestinal de glucosa en mmol min^{-1} y se define como:

$$u_g = \frac{g_2}{t_{maxG}} \quad (5.12)$$

- G es la concentración de glucosa en mmol L^{-1} y se define como:

$$G = \frac{Q_1}{v_g} \quad (5.13)$$

- v_g es el volumen de distribución de glucosa en L kg^{-1} .
- f_{02} es el flujo de glucosa independiente de insulina en $\text{mmol kg}^{-1} \text{min}^{-1}$ y se define como:

$$f_{02} = \begin{cases} f_{01} & \text{si } G \geq 4,5 \text{ mmol L}^{-1} \\ f_{01} \frac{G}{4,5} & \text{en otro caso} \end{cases} \quad (5.14)$$

- F_r es la glucosa renal en mmol kg^{-1} y se define como:

$$F_r = \begin{cases} 0,003 (G - 9) v_g & \text{si } G \geq 9 \text{ mmol L}^{-1} \\ 0 & \text{en otro caso} \end{cases} \quad (5.15)$$

- e_{gp} es la producción endógena de glucosa en $\text{mmol kg}^{-1} \text{min}^{-1}$ y se define como:

$$e_{gp} = \begin{cases} e_{gp_0} (1 - x_3) & \text{si } e_{gp} \geq 0 \\ 0 & \text{en otro caso} \end{cases} \quad (5.16)$$

El modelo de Cambridge tiene muchos parámetros que deben ser identificados, especialmente en la parte de acciones de la insulina, donde hay dos parámetros por cada acción correspondientes a los flujos de entrada y salida de cada compartimento. Habitualmente estos parámetros se reformulan en las llamadas sensibilidades a la insulina, debido a su significado fisiológico ya que se corresponde con el decrecimiento de la glucosa por unidad de insulina dada. La reformulación es entonces:

- $s_{it} = \frac{k_{b1}}{k_{a1}}$ donde s_{it} es la sensibilidad a la insulina para el transporte de glucosa.
- $s_{id} = \frac{k_{b2}}{k_{a2}}$ donde s_{id} es la sensibilidad a la insulina para la distribución de glucosa.
- $s_{ie} = \frac{k_{b3}}{k_{a3}}$ donde s_{ie} es la sensibilidad a la insulina para la producción endógena de glucosa.

Después de estas transformaciones, las ecuaciones (5.7), (5.8) y (5.9) resultan en:

$$\dot{x}_1 = -k_{a1} \cdot x_1 + k_{a1} \cdot s_{it} \cdot I \quad (5.17)$$

$$\dot{x}_2 = -k_{a2} \cdot x_2 + k_{a2} \cdot s_{id} \cdot I \quad (5.18)$$

$$\dot{x}_3 = -k_{a3} \cdot x_3 + k_{a3} \cdot s_{ie} \cdot I \quad (5.19)$$

El modelo de Cambridge se ha utilizado tanto para fines de simulación como de control, en muchos escenarios, desde pacientes críticos a experimentos nocturnos con resultados exitosos.

Los valores nominales de los parámetros del modelo se muestran en la Tabla 5.1. Los parámetros mostrados son valores medios de los diferentes conjuntos de parámetros publicados.

Parámetro	Valor	Unidades
k_{12}	0,066	min^{-1}
v_g	0,16	L kg^{-1}
egp_0	0,0161	$\text{mmol kg}^{-1} \text{min}^{-1}$
f_{01}	0,0097	$\text{mmol kg}^{-1} \text{min}^{-1}$
k_e	0,138	min^{-1}
v_i	0,12	L kg^{-1}
k_{a1}	0,006	min^{-1}
k_{a2}	0,06	min^{-1}
k_{a3}	0,03	min^{-1}
S_{it}	$51,2 \cdot 10^{-4}$	$\text{mU L}^{-1} \text{min}^{-1}$
S_{id}	$8,2 \cdot 10^{-4}$	$\text{mU L}^{-1} \text{min}^{-1}$
S_{ie}	$520 \cdot 10^{-4}$	$\text{mU L}^{-1} \text{min}^{-1}$
a_g	0,8	sin unidades
t_{maxG}	40	min
t_{maxI}	55	min

Tabla 5.1. Valores nominales de los parámetros del modelo de Cambridge

Capítulo 6

Alternativas de solución

En el presente proyecto hay tres puntos a tener en cuenta a la hora de seleccionar el mejor modo para solucionar el problema planteado:

- El lenguaje de programación para realizar la simulación.
- La forma de introducir los datos en el simulador.
- El tipo de comunicación entre ordenadores.

6.1. Lenguaje de programación

Los lenguajes de alto nivel y propósito general como C, C++ o Java son robustos, altamente flexibles y potentes. Una aplicación implementada en un lenguaje de estas características puede resolver problemas de cualquier índole y presenta muy pocas limitaciones. Sin embargo, a pesar de las ventajas que proporciona existen ciertos inconvenientes:

- Requiere más esfuerzo de desarrollo. Las funciones matemáticas no están disponibles por lo que se hace necesario implementarlas.
- Dificulta el diseño de interfaces visuales. La representación de señales y gráficas se hace más complicada que en otro tipo de lenguajes de propósito no general.
- No es adecuado para sistemas en tiempo real.

Tradicionalmente estos impedimentos eran excesivos y requerían un esfuerzo en el desarrollo muy grande. Sólo proyectos muy especiales o que precisaban de altas cotas de calidad podían justificar el trabajo a realizar. Sin embargo, actualmente existen herramientas que salvan dichas limitaciones. Existen librerías que ofrecen componentes software reutilizables que proporcionan a las aplicaciones desarrolladas en lenguajes de alto nivel capacidades matemáticas. Estos componentes incluyen elementos gráficos para la monitorización de señales y mecanismos para la adquisición de datos. Estas librerías consiguen reducir los tiempos de desarrollo hasta hacerlos comparables con los obtenidos por medio de otras soluciones.

Una alternativa a los lenguajes tradicionales es Matlab de Mathworks. Matlab es un lenguaje de programación de alto nivel y un entorno interactivo para el cálculo numérico, la visualización y la programación. Mediante Matlab es posible realizar el análisis de datos, el desarrollo de algoritmos, la creación de interfaces de usuario o la comunicación con otros programas en otros lenguajes y con otros dispositivos hardware. El lenguaje, las herramientas y las funciones matemáticas incorporadas permiten explorar diversos enfoques y llegar a una solución antes que con hojas de cálculo o lenguajes de programación tradicionales.

Matlab es capaz de resolver todos los problemas asociados al objeto del proyecto. Por un lado, permite implementar el sistema de ecuaciones diferenciales que se plantea y encontrar la solución en función de los datos de entrada al modelo. Por otro, ofrece la posibilidad de crear una interfaz gráfica para facilitar la introducción de los datos y permite visualizar los resultados en gráficas temporales que facilitan el análisis de resultados. Además, permite la comunicación entre dispositivos por lo que es posible establecer una comunicación entre ordenadores. Por tanto, Matlab se convierte en un lenguaje adecuado para la realización del presente proyecto, eliminando los inconvenientes que pueden aparecer con el uso de lenguajes tradicionales.

6.2. Introducción de datos

Como se ha indicado anteriormente es necesario introducir determinados datos para que sea posible realizar la simulación de un paciente. Estos datos se encuentran en tres matrices distintas que utilizará el simulador para realizar los cálculos oportunos.

Desde el punto de vista del programador, la forma más sencilla de introducir los datos es al inicio del programa, escribiendo las tres matrices. Sin embargo, esto supone un problema puesto que se hace necesario tener conocimientos de programación para poder hacer uso del simulador. Por este motivo se desarrolla una interfaz gráfica en la que se pueden introducir fácilmente todos los datos necesarios.

Para la realización de la interfaz se utiliza Guide, se encuentra disponible en Matlab y se trata de un entorno de programación visual que permite realizar programas que necesitan un ingreso continuo de datos. De este modo, al ejecutar el programa se abrirá una ventana que permitirá introducir los datos sin necesidad de acceder al código.

Así se consigue desarrollar un simulador que permite introducir los datos de forma sencilla y visual. Además, de este modo puede ser utilizado por cualquier persona puesto que no es necesario tener conocimientos de programación.

6.3. Comunicación

Para realizar la comunicación entre los dos ordenadores se deben tener en cuenta unos criterios:

- Debe ser una comunicación fiable y robusta.
- Debe ser bidireccional, es decir, debe permitir enviar y recibir datos.
- Debe poderse realizar de forma sencilla.

A continuación se describen diferentes formas de comunicación que se pueden escoger para la comunicación entre los ordenadores cumpliendo los criterios establecidos.

- Comunicación a través del puerto serie. La conexión se realiza mediante un cable que se conecta al puerto serie. El puerto serie transfiere información enviando o recibiendo un bit. Actualmente en la mayoría de los periféricos serie, la interfaz USB ha reemplazado al puerto serie por ser más rápida. La mayor parte de ordenadores están conectados a dispositivos externos a través de USB y, a menudo, ni siquiera llegan a tener puerto serie.
- Comunicación WLAN (Wireless Local Area Network). Es un tipo de comunicación inalámbrica muy utilizada como alternativa a las redes de área local cableadas o como extensión de éstas. Está pensada para utilizarse al aire libre, entre diferentes habitaciones o entre casas. Un ejemplo de este tipo de comunicación es la Wi-Fi.
- Comunicación WPAN (Wireless Personal Area Network). Es un tipo de comunicación inalámbrica que permite la comunicación entre distintos dispositivos cercanos al punto de acceso. Estas redes normalmente son de unos pocos metros y para uso personal. Un ejemplo de este tipo de comunicación es el Bluetooth.

De las tres alternativas comentadas la ideal sería la comunicación a través de una red Wi-Fi, de modo que se pueda acceder al simulador sin necesidad de cables y sin necesidad de estar muy próximo. Sin embargo, por su mayor sencillez a la hora de realizar la comunicación se ha optado por el puerto serie de modo que sólo sea necesario un cable para poder hacer uso del simulador.

Capítulo 7

Metodología

7.1. Diseño del simulador

Para realizar la implementación de un simulador de un paciente con diabetes mellitus tipo 1 hay que replicar, lo más fielmente posible, el comportamiento del metabolismo de la glucosa. Para ello se utiliza un modelo matemático que representa las variables objeto de estudio. Como se ha indicado en el Capítulo 5 el modelo seleccionado es el propuesto por el grupo de Cambridge.

Para poder realizar la simulación es necesario programar en Matlab el sistema de ecuaciones diferenciales ordinarias y los diferentes parámetros y ecuaciones que lo definen. Este sistema es el que se debe resolver para conocer los valores de las variables objeto de estudio. Para que se pueda resolver es necesario indicar las entradas al modelo, que se introducen a través de la interfaz gráfica.

En primer lugar es importante conocer cuáles son las entradas al modelo:

- Carbohidratos ingeridos. Instante en el que se realiza la ingesta y cantidad ingerida.
- Bolos de insulina administrados de forma subcutánea. Instante en el que se realiza la inyección y dosis administrada.
- Insulina basal. Valores de insulina basal en los diferentes momentos del día.

Puesto que se busca una simulación HIL (*Hardware in the Loop*) es necesario recibir los datos de entrada de otro ordenador y procesarlos una vez recibidos de modo que posteriormente se pueda calcular la solución del sistema de ecuaciones.

Una vez obtenidos los resultados se enviarán al otro ordenador que será el que se encargue de realizar la representación gráfica para el análisis de resultados. Además, se guardarán los datos en dos archivos de modo que se pueda acceder a ellos posteriormente. El primer archivo contiene el vector de tiempos de la simulación. El segundo archivo contiene la matriz de resultados. Esta matriz tiene una columna por cada variable del sistema de ecuaciones diferenciales ($g_1, g_2, s_1, s_2, I, x_1, x_2, x_3, Q_1$ y Q_2) y cada fila representa el valor de cada una de ellas para el instante de tiempo definido por el vector de tiempos.

Las salidas del modelo objeto de estudio se representan en gráficas temporales de modo que se facilite el análisis de las mismas. Las variables a representar son las siguientes:

- Insulina en plasma. Permite visualizar los niveles de insulina en sangre a lo largo del periodo de simulación, pudiéndose observar los picos que se corresponden con las inyecciones de insulina.
- Tasa de absorción de carbohidratos. Representa los instantes en los que se realizan las ingestas.
- Efecto de la insulina. Se observa la evolución de la acción de la insulina en el transporte/distribución de glucosa (x_1), la eliminación de glucosa (x_2) y la producción de glucosa exógena (x_3).
- Glucosa en plasma. Permite visualizar los niveles de glucosa en sangre a lo largo del periodo de simulación, pudiéndose observar los picos que se corresponden con la ingesta de carbohidratos.

A continuación se muestran un ejemplo de las gráficas que se obtienen para una simulación de tres días con tres comidas diarias.

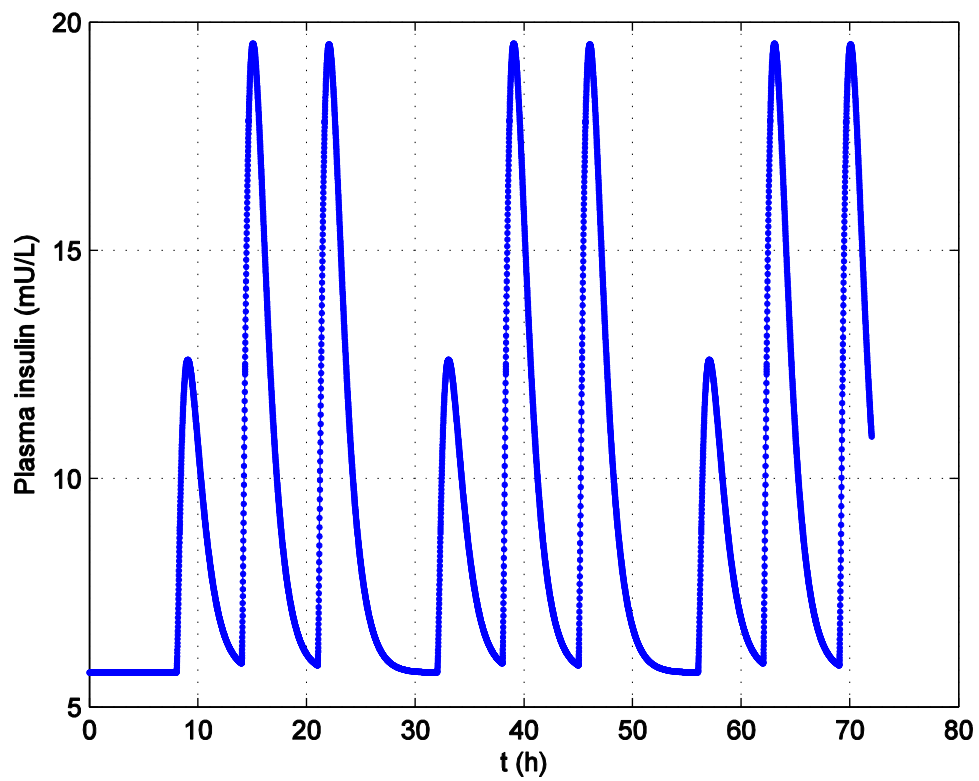


Figura 7.1. Evolución temporal de la insulina en plasma

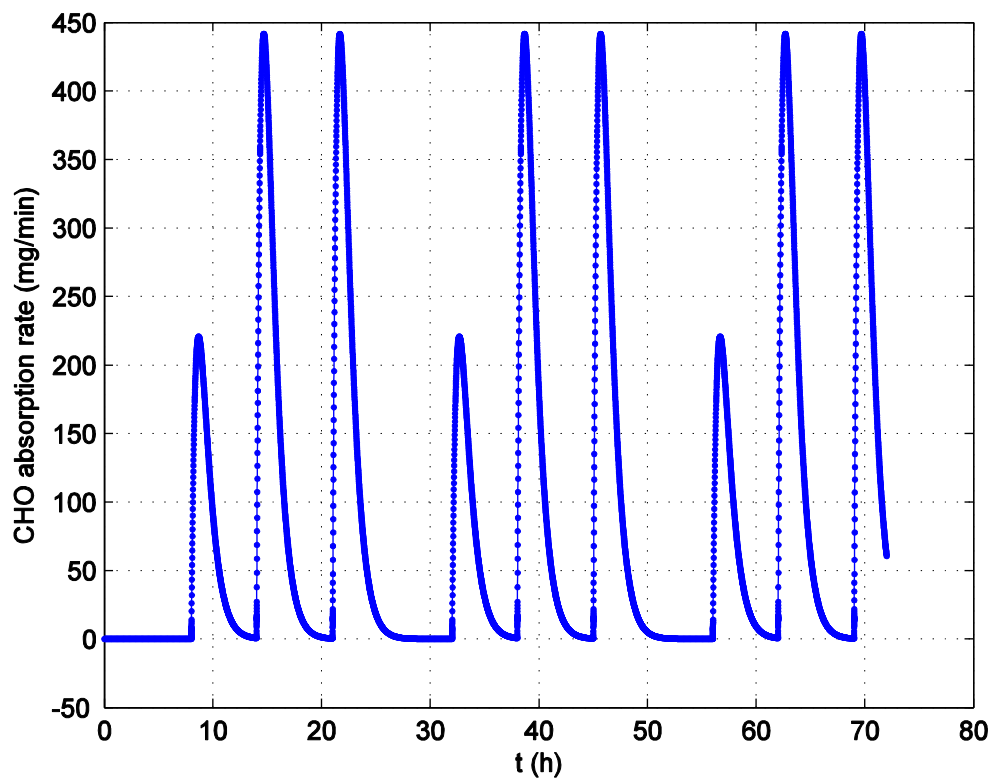


Figura 7.2. Tasa de absorción de carbohidratos

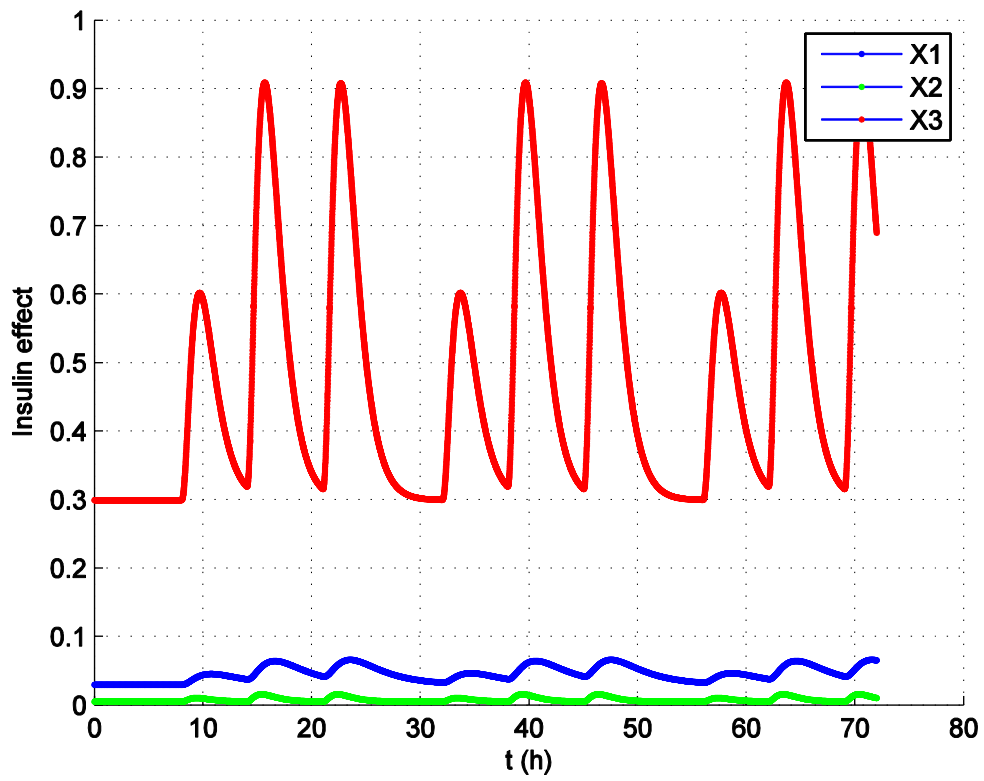


Figura 7.3. Evolución temporal del efecto de la insulina

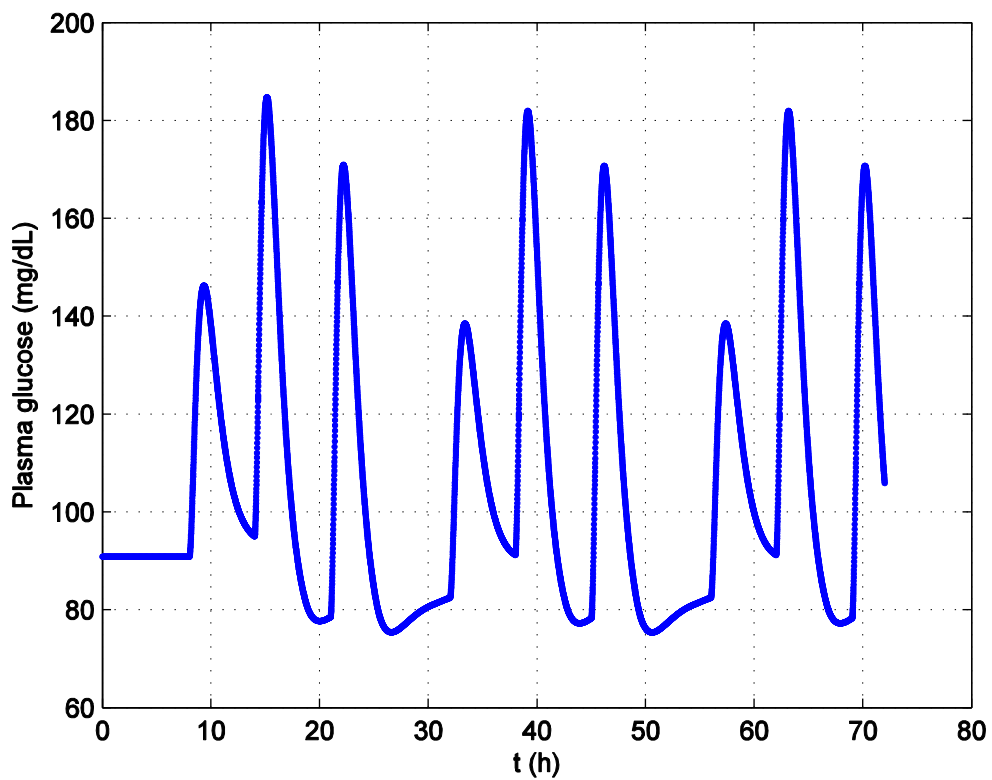


Figura 7.4. Evolución temporal de la glucosa en plasma

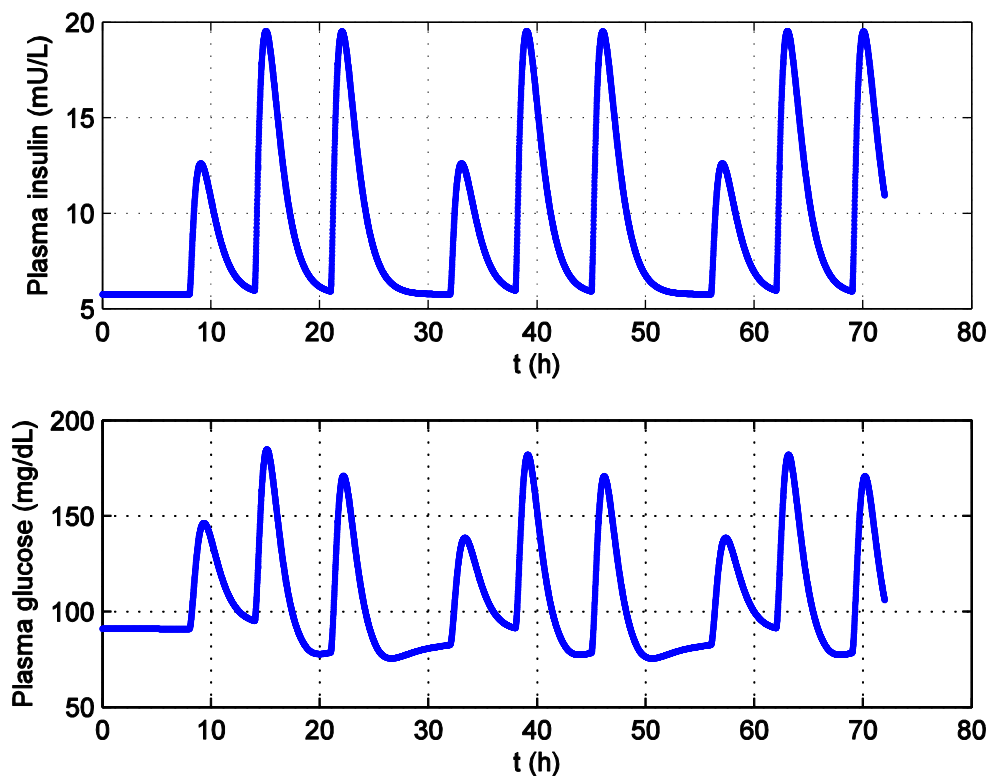


Figura 7.5. Comparación de la evolución temporal de la insulina y glucosa en plasma

El simulador se diseña de modo que sea posible realizar la simulación del paciente de tres modos distintos. El tipo de simulación se indica en la interfaz gráfica junto con los datos de entrada, seleccionando una de las siguientes opciones:

- Simulación en bloque. Con este modo de simulación se introducen todos los datos, se realiza la simulación y una vez finalizada se muestran los resultados en las gráficas correspondientes.
- Simulación en tiempo real. Este tipo de simulación se tarda el mismo tiempo en conocer los datos que en el caso real. Se muestran los resultados cada cierto tiempo definido por el usuario previamente al inicio de la simulación.
- Simulación en tiempo real acelerado. Se trata del mismo caso que el anterior pero se añade un factor de aceleración de modo que una simulación de varios días se pueda realizar en unas pocas horas.

7.2. Diseño de la interfaz gráfica

Se diseña una interfaz gráfica para introducir los datos de entrada y el modo de simulación de forma que sea sencillo y visual. Puesto que introducir los datos puede llevar un cierto tiempo se incluye la opción de guardar de modo que sea posible reutilizar unos datos ya introducidos.

Al ejecutar el programa se abre una pantalla en la que en primer lugar se observa una lista en la que se encuentran todos los escenarios ya guardados y al lado las opciones de simulación. A continuación se observa la parte de introducción de datos. En primer lugar aparece un cuadro de texto en el que se puede introducir el nombre del escenario que se desea simular. A continuación aparecen tres partes muy similares para introducir los datos de entrada. En la Figura 7.6 se puede ver el aspecto de la interfaz.

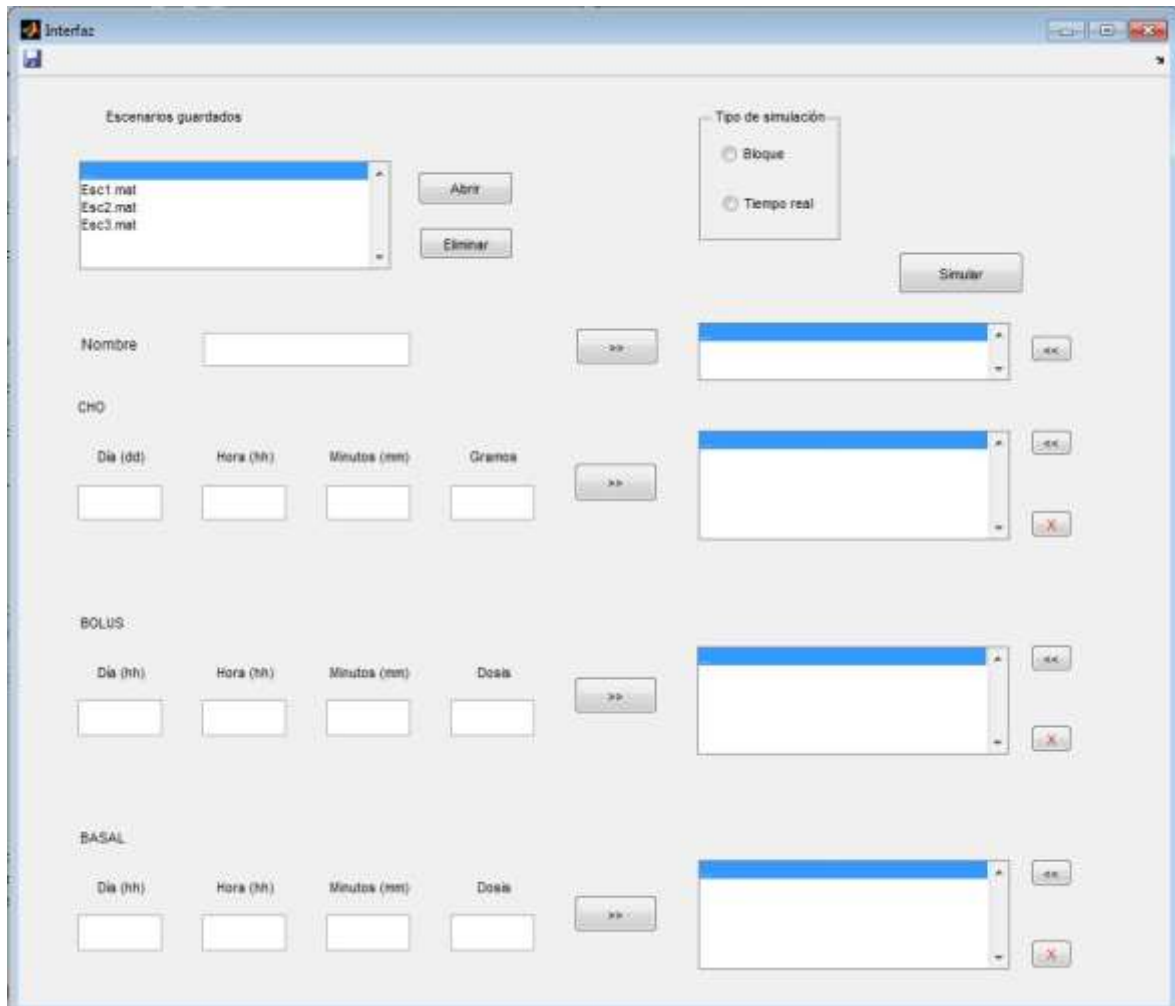


Figura 7.6. Interfaz para introducir los datos de entrada

Para introducir los datos de las ingestas se completan los cuatro cuadros de texto donde hay que introducir el día, la hora, los minutos y los gramos de carbohidratos ingeridos en la comida. Una vez introducidos los datos de una comida se añaden al listado que aparece

a la derecha, donde se pueden observar todas las comidas ya introducidas. A continuación se realiza lo mismo para las dosis de insulina administrada y para la insulina basal. Una vez introducidos todos los datos se puede observar en la parte derecha cuatro listas con el nombre, las comidas, los bolos de insulina y la insulina basal.

En caso de que se desee guardar el escenario creado, se pulsa el botón guardar. Se puede comprobar que en la lista de escenarios guardados aparece el que se acaba de crear. En caso de que no se desee guardar el escenario creado, no es necesario asignarle un nombre. En la Figura 7.7 se puede ver como queda la interfaz una vez se han introducido los datos y se ha guardado el escenario creado.

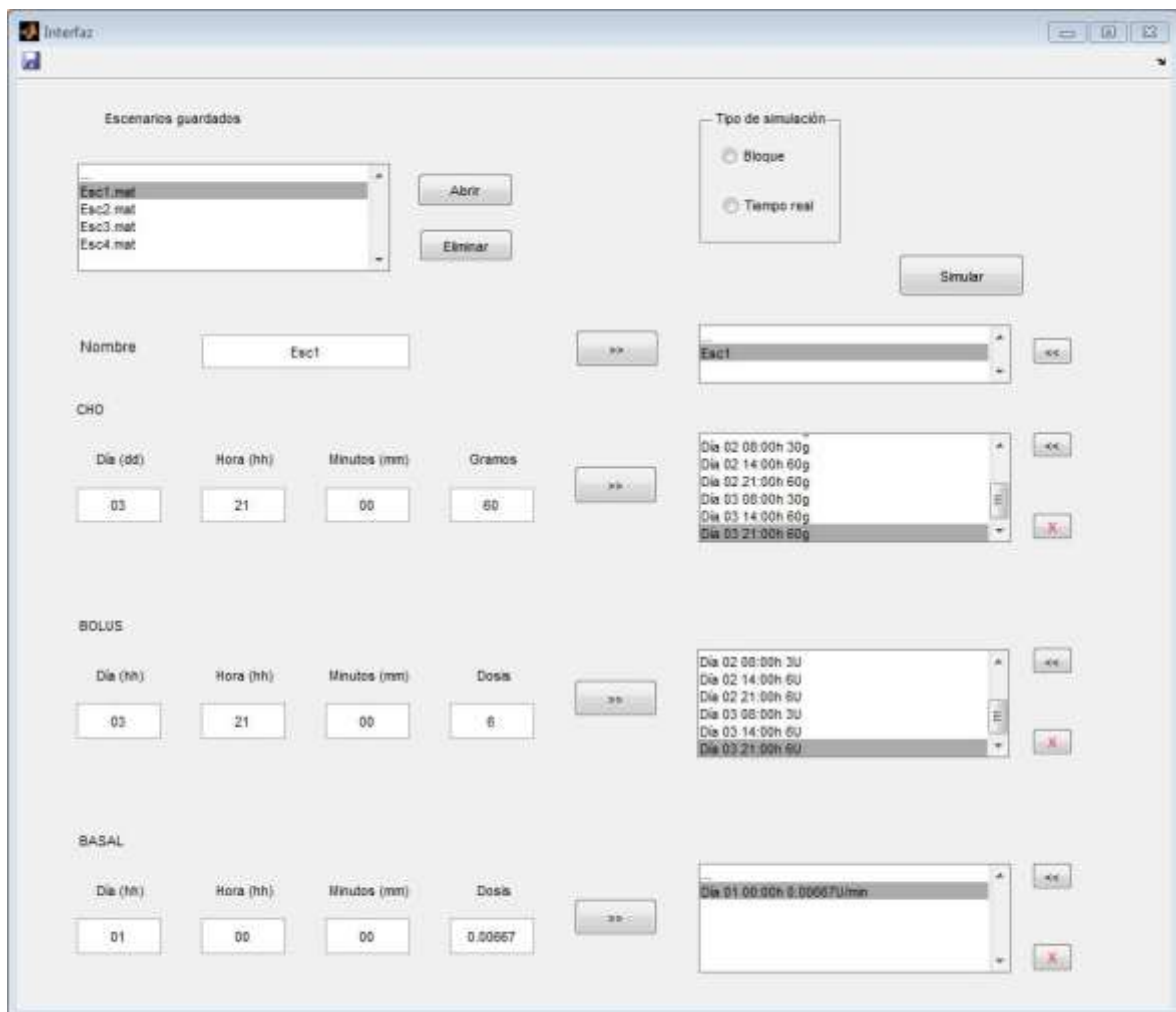


Figura 7.7. Interfaz con los datos del escenario que se va a simular una vez guardado

En caso de que se quiera utilizar un escenario creado previamente se abre el que se desee de los ya guardados, pudiéndose observar como se completan las listas.

Una vez están completas las listas con los datos se puede realizar la simulación. En caso de que se desee una simulación en bloque (opción por defecto) se pulsa el botón de simular y en el momento en el que la simulación finalice y se hayan recibido los resultados se mostrarán las gráficas temporales que se han indicado en el apartado anterior.

En caso de querer realizar una simulación en tiempo real se debe cambiar la opción e incluir en los cuadros que aparecen cada cuánto tiempo se desean recibir datos y el factor de aceleración deseado. En caso de no querer acelerar el proceso se escribe un 1. En la Figura 7.8 se puede observar la interfaz para una simulación en tiempo real.

Para facilitar el uso de la interfaz se han añadido botones que permiten eliminar escenarios guardados, ingestas de carbohidratos, inyecciones de insulina o insulina basal. Además, se pueden editar los datos introducidos en un escenario de modo que puedan corregirse errores o realizar modificaciones en escenarios ya creados.

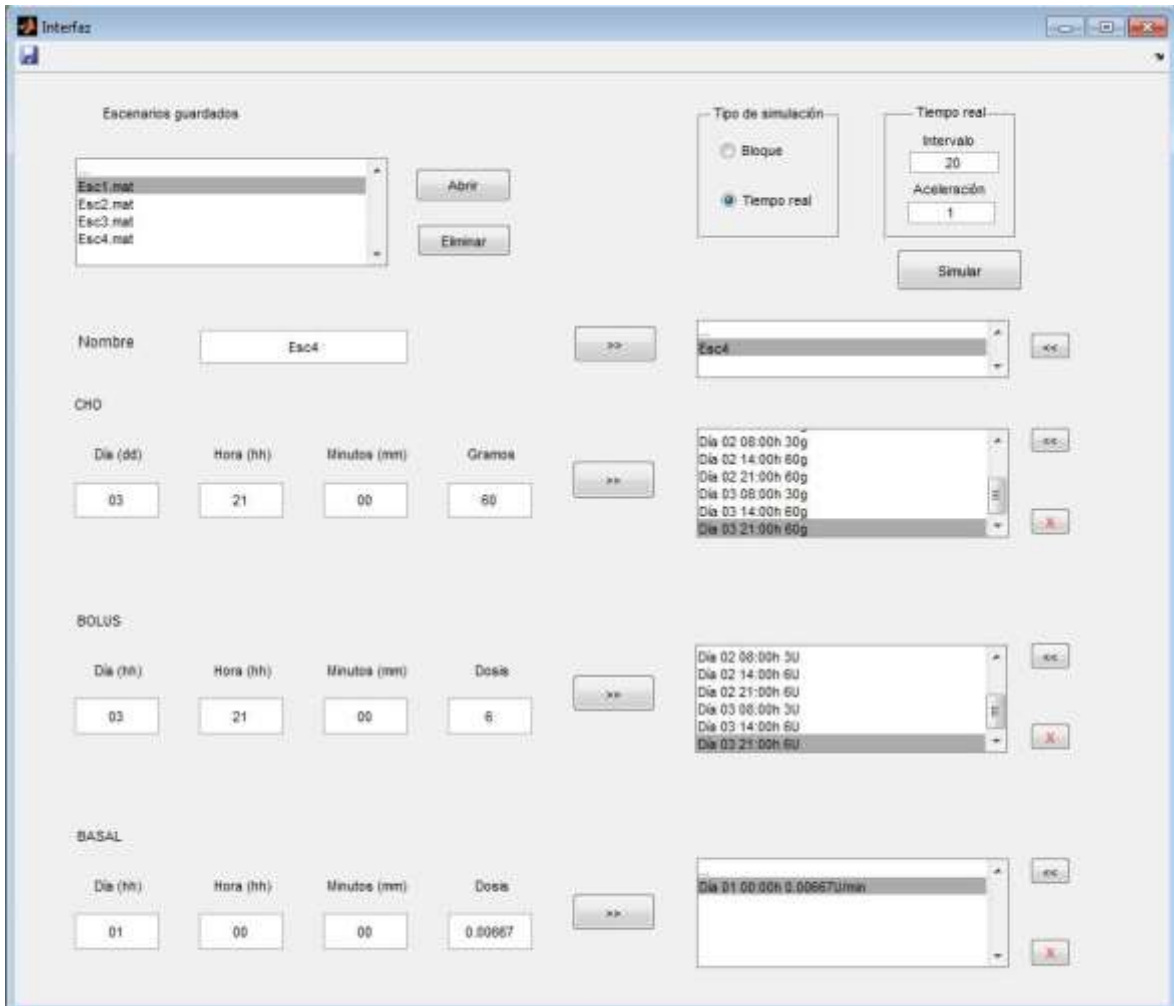


Figura 7.8. Interfaz para simulación en tiempo real

7.3. Diseño de la comunicación

El tipo de comunicación que se va a emplear es a través del puerto serie, como ya se había comentado previamente. Para ello es necesario disponer de dos ordenadores y de un cable que permita la conexión a través del puerto serie.

Para evitar la necesidad de disponer de dos ordenadores conectados a través del cable se decide realizar la comunicación a través de un puerto serie virtual, de modo que sólo sea necesario un ordenador.

Para poder realizar la implementación del programa de este modo es necesario crear los puertos series virtuales. Para ello hay que instalar una aplicación que permita crear este tipo de puertos. Se ha seleccionado Virtual Serial Port Emulator de Eterlogic por tratarse de una aplicación gratuita que permite crear diferentes tipos de conectores de puerto serie.

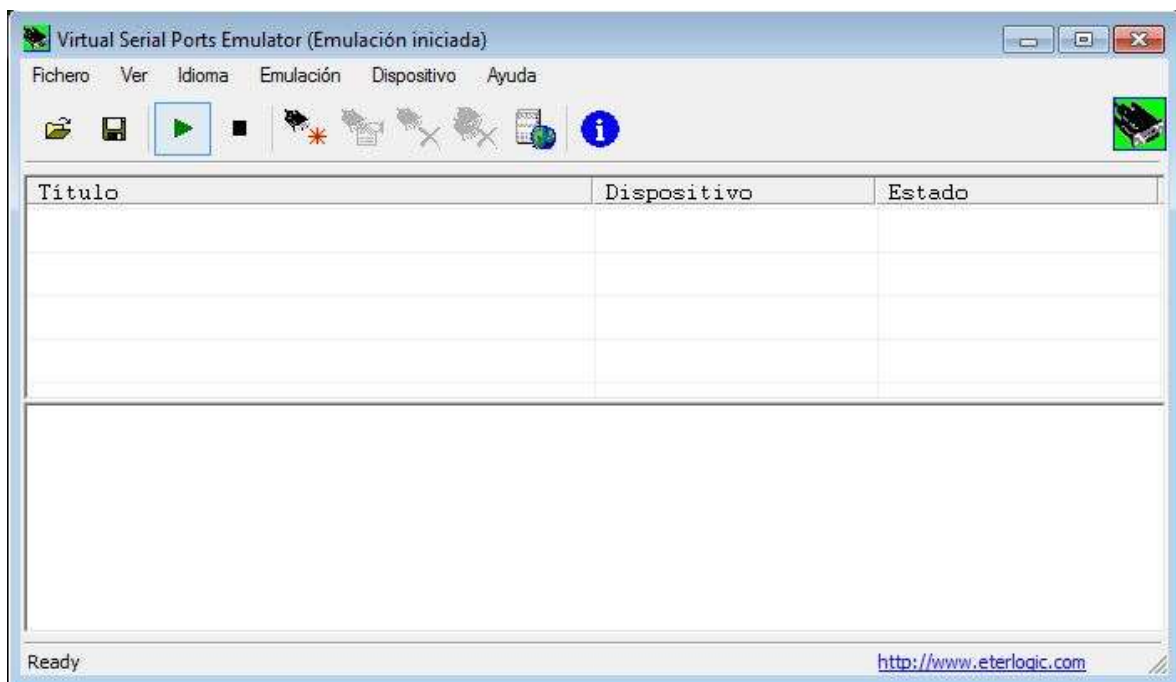


Figura 7.9. Pantalla de inicio del simulador de puerto serie virtual

Una vez instalada la aplicación se debe crear un nuevo puerto serie virtual. Para ello se selecciona la opción crear puerto serie virtual. Posteriormente se indica que clase de dispositivo se desea. En este caso se busca simular dos puertos series conectados por medio de un cable. Para crear este tipo de conexión se abre el desplegable y se selecciona Pair.

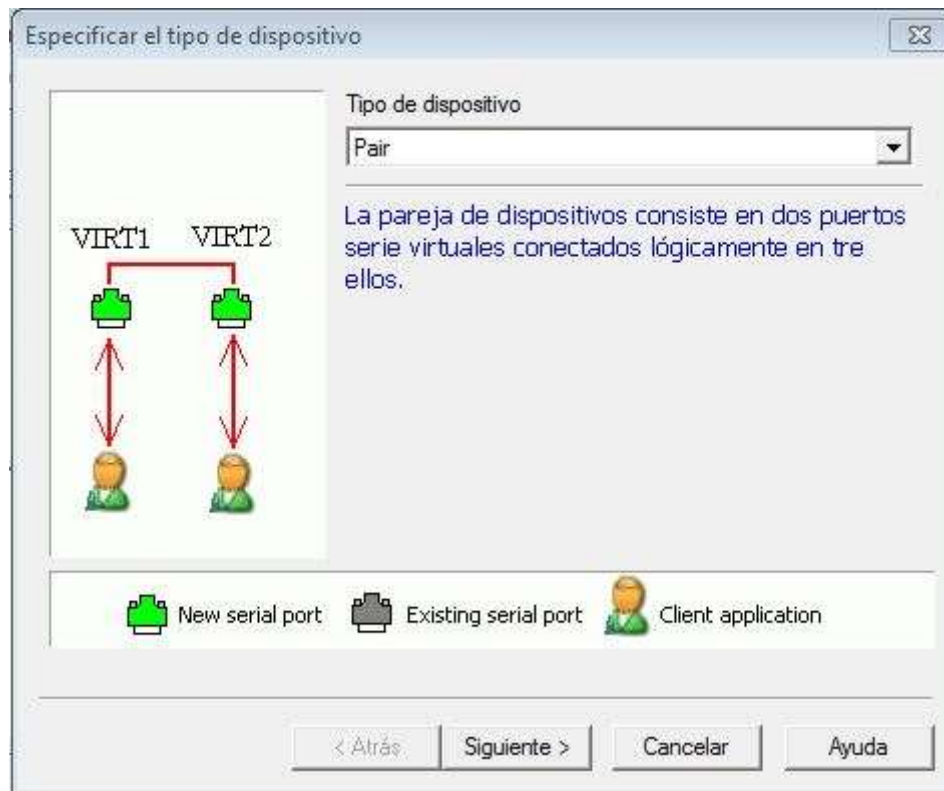


Figura 7.10. Pantalla para especificar el tipo de dispositivo

Se pasa a la ventana siguiente donde se debe indicar el nombre de los puertos. Estos nombres son los que se utilizarán posteriormente en Matlab para establecer la comunicación. Una vez creados los puertos se inicia la simulación del puerto serie. En el momento en que comienza la simulación ya se puede establecer la comunicación.

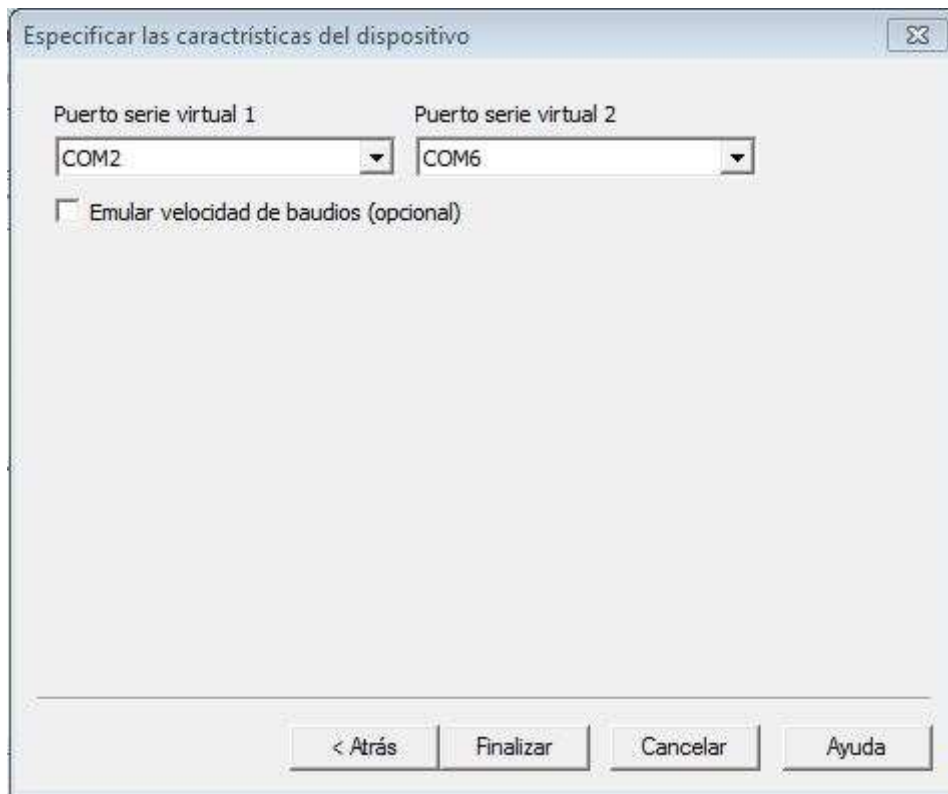


Figura 7.11. Pantalla para especificar el nombre de los puertos serie virtuales

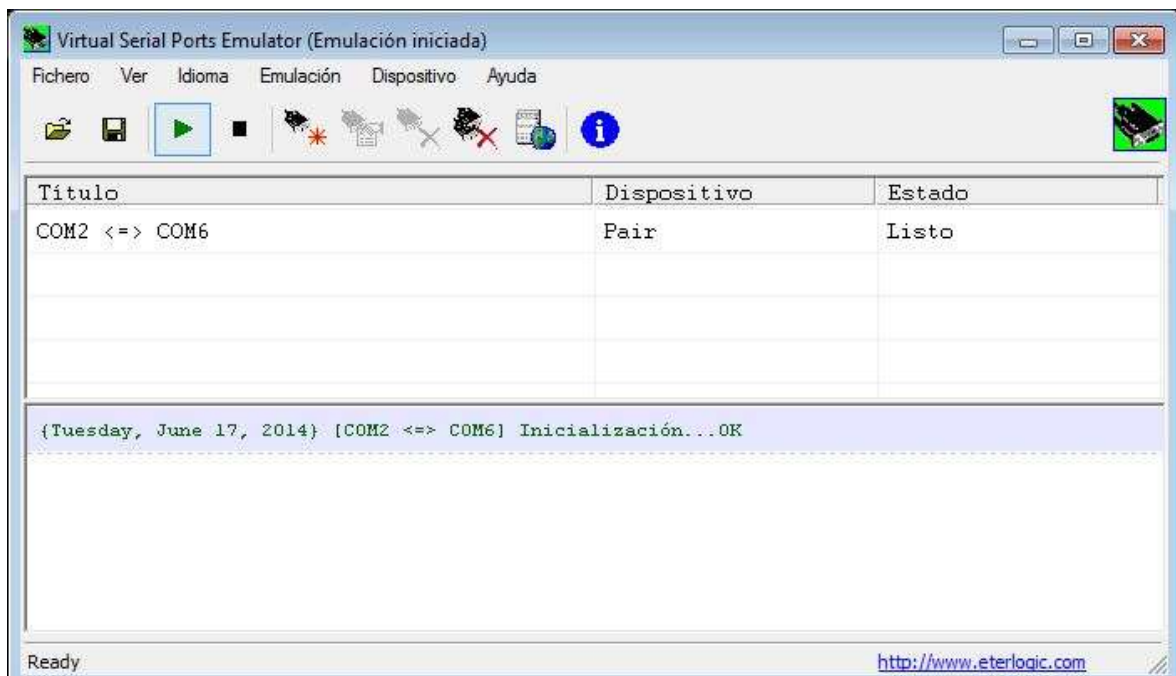


Figura 7.12. Simulación iniciada

Para simular la existencia de dos ordenadores se abrirá el programa Matlab dos veces. Cada uno representa un ordenador, por lo que en uno estará disponible la interfaz gráfica y en el otro el simulador. Para que ambos programas se puedan comunicar es necesario abrir los puertos serie creados previamente. Para ello se escribe en Matlab el código que permite establecer la comunicación entre dispositivos a partir de este tipo de puertos.

Para realizar la comunicación se debe establecer un protocolo de comunicación. Tanto cuando se envían las entradas como los resultados, se están enviando matrices de datos. Cuando se envían los datos, el primer dato indica el número de filas de la primera matriz. De este modo el que recibe los datos ya sabe cuántos datos va a incluir en la primera matriz. Los siguientes datos son las filas de la matriz que se van almacenando conforme se van recibiendo, creándose finalmente una matriz igual a la existente en el programa que ha enviado los datos. Una vez enviada la primera matriz, envía la longitud de la segunda, comenzando así de nuevo el proceso, hasta que se hayan enviado todos los datos.

Cada vez que se envía un dato se espera a recibir una confirmación de que el dato ha sido recibido y, hasta que no se recibe la confirmación, no se envía el dato siguiente. De este modo se evita que se pierdan datos por superar el buffer del puerto serie. Esto implica que al recibir un dato se debe guardar el dato leído y, a continuación, enviar la confirmación de que el dato se ha recibido. De lo contrario no se recibirían más datos. En la Figura 7.13 se puede observar un esquema del protocolo de comunicación.

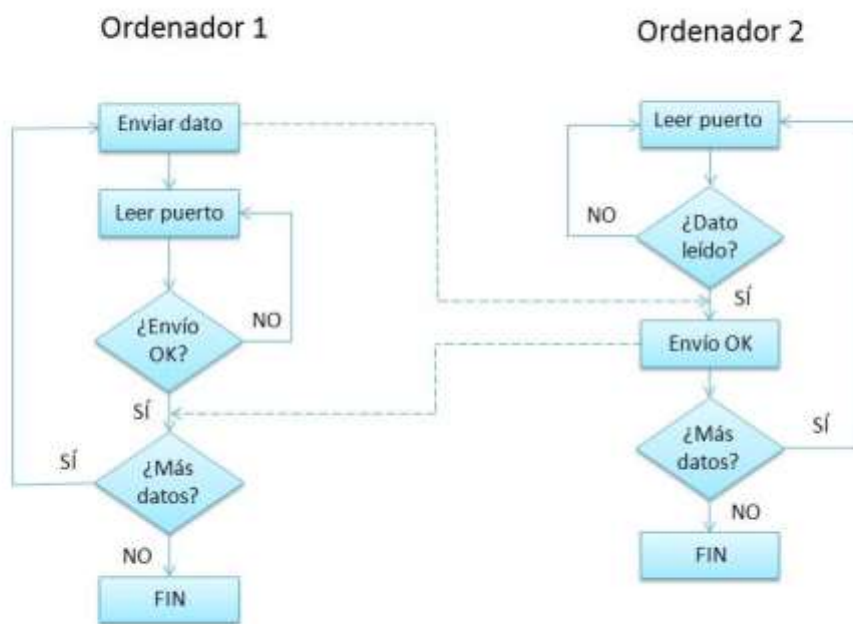


Figura 7.13. Esquema del protocolo de comunicación

Capítulo 8

Resultados

En este apartado se van a mostrar los resultados que se obtienen al simular diferentes situaciones. En primer lugar se muestra una simulación de un día y, a continuación, la misma simulación aumentando y disminuyendo la dosis de insulina. De este modo se puede observar el cambio en el comportamiento del metabolismo del paciente ante diferentes dosis de insulina. Por último, se muestra una simulación de cuatro días para observar el comportamiento en un intervalo de tiempo mayor.

- Caso 1. Simulación de un día con cinco comidas.
 - Desayuno: 8:00h, 20g de carbohidratos, 0.8U de insulina
 - Almuerzo: 10:00h, 10 g de carbohidratos, 0.4U de insulina
 - Comida: 14:30h, 60g de carbohidratos, 2.4U de insulina
 - Merienda: 18:00h, 20g de carbohidratos, 0.8U de insulina
 - Cena 22:00h, 40g de carbohidratos, 1.6U de insulina

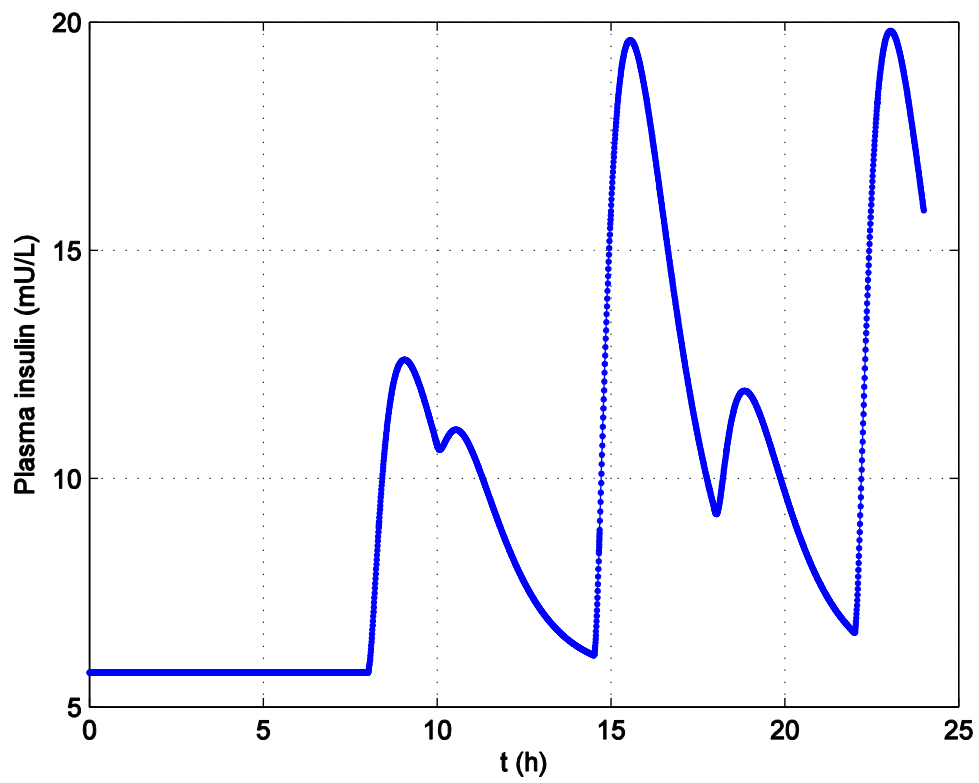


Figura 8.1. Evolución temporal de la insulina en plasma (caso 1)

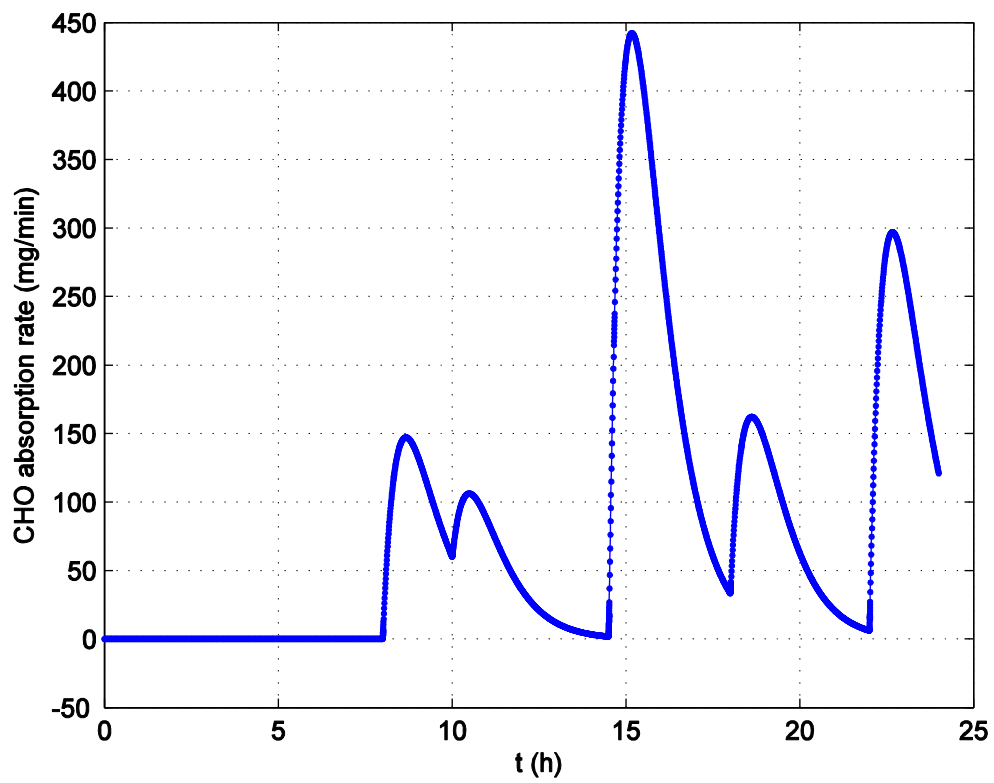


Figura 8.2. Tasa de absorción de carbohidratos (caso 1)

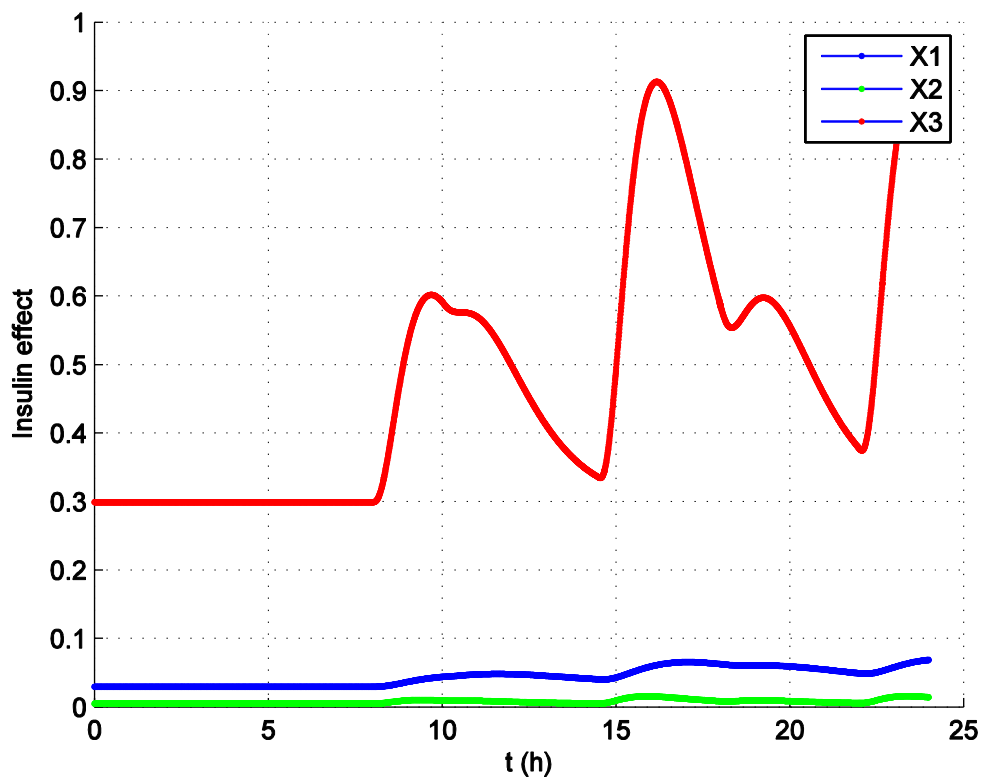


Figura 8.3. Evolución temporal del efecto de la insulina (caso 1)

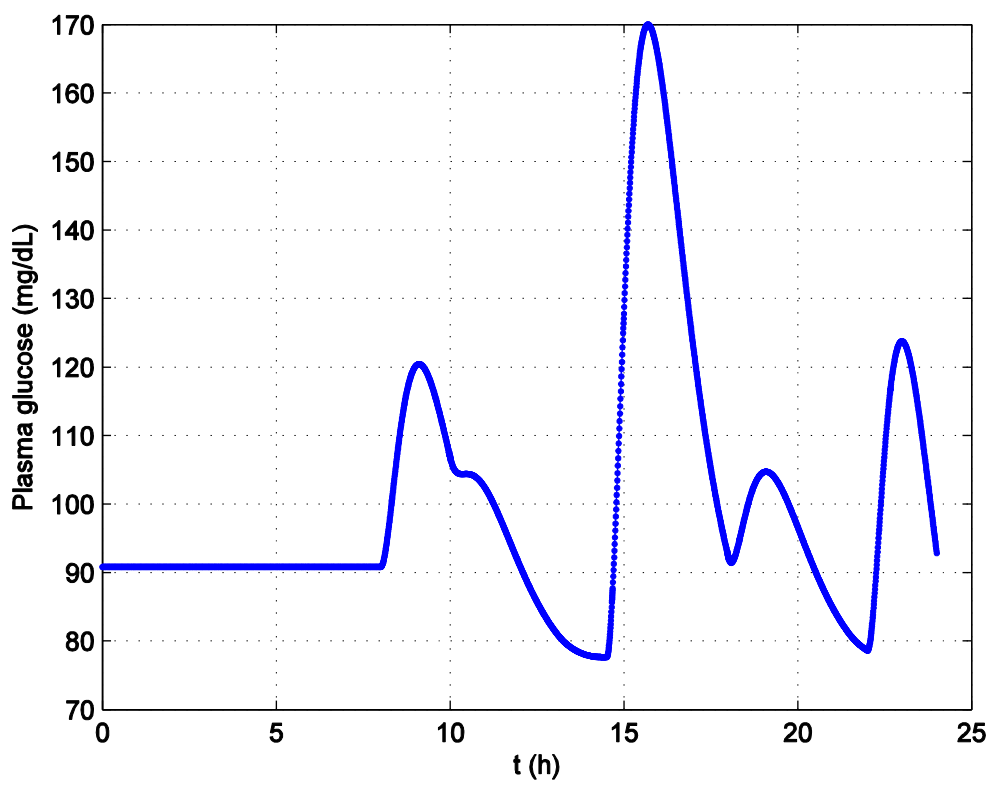


Figura 8.4. Evolución temporal de la glucosa en plasma (caso 1)

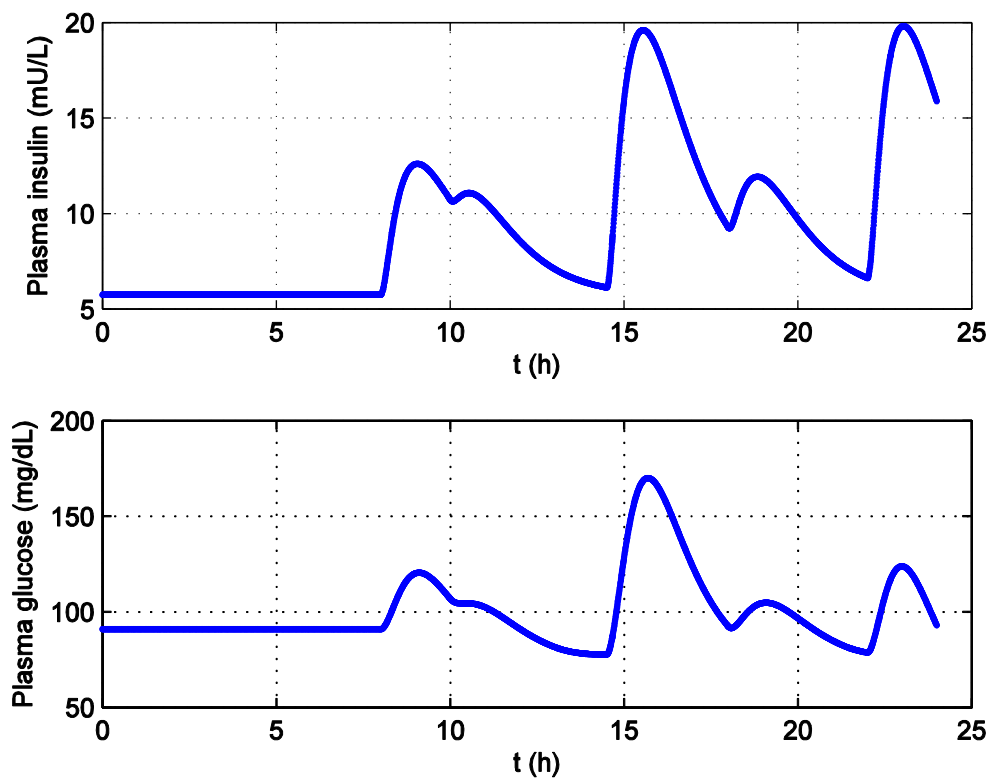


Figura 8.5. Comparación de la evolución de la insulina y glucosa en plasma (caso 1)

- Caso 2. Simulación de un día con cinco comidas. Misma situación que en el caso anterior disminuyendo la dosis de insulina administrada.
 - Desayuno: 8:00h, 20g de carbohidratos, 0.5U de insulina
 - Almuerzo: 10:00h, 10 g de carbohidratos, 0.25U de insulina
 - Comida: 14:30h, 60g de carbohidratos, 1.5U de insulina
 - Merienda: 18:00h, 20g de carbohidratos, 0.5U de insulina
 - Cena 22:00h, 40g de carbohidratos, 1U de insulina

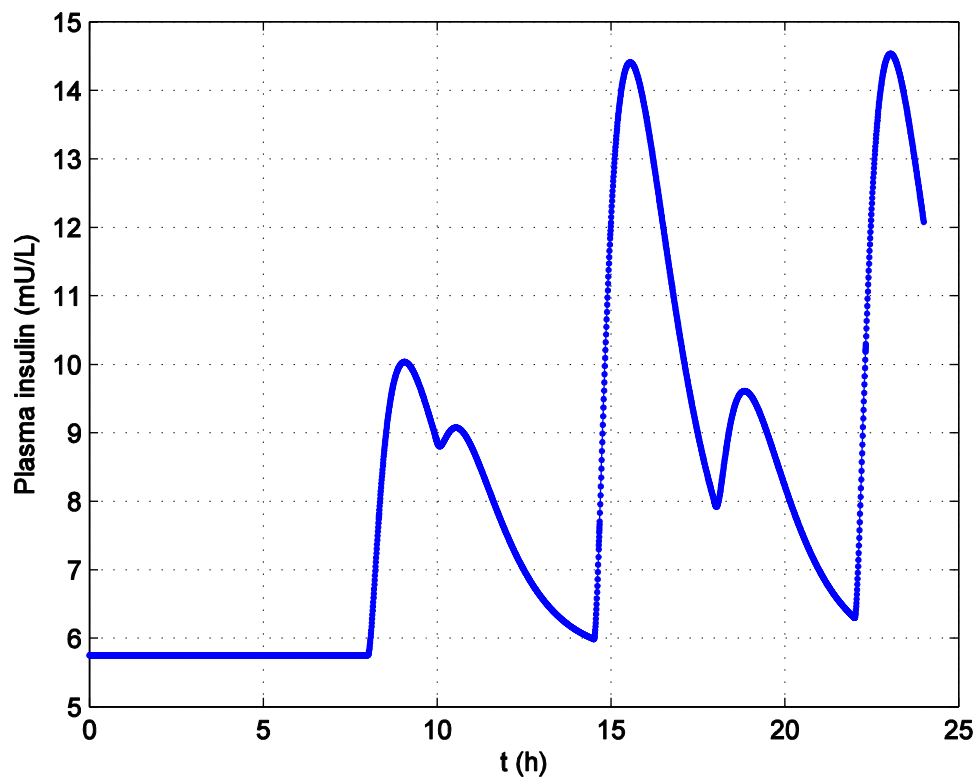


Figura 8.6. Evolución temporal de la insulina en plasma (caso 2)

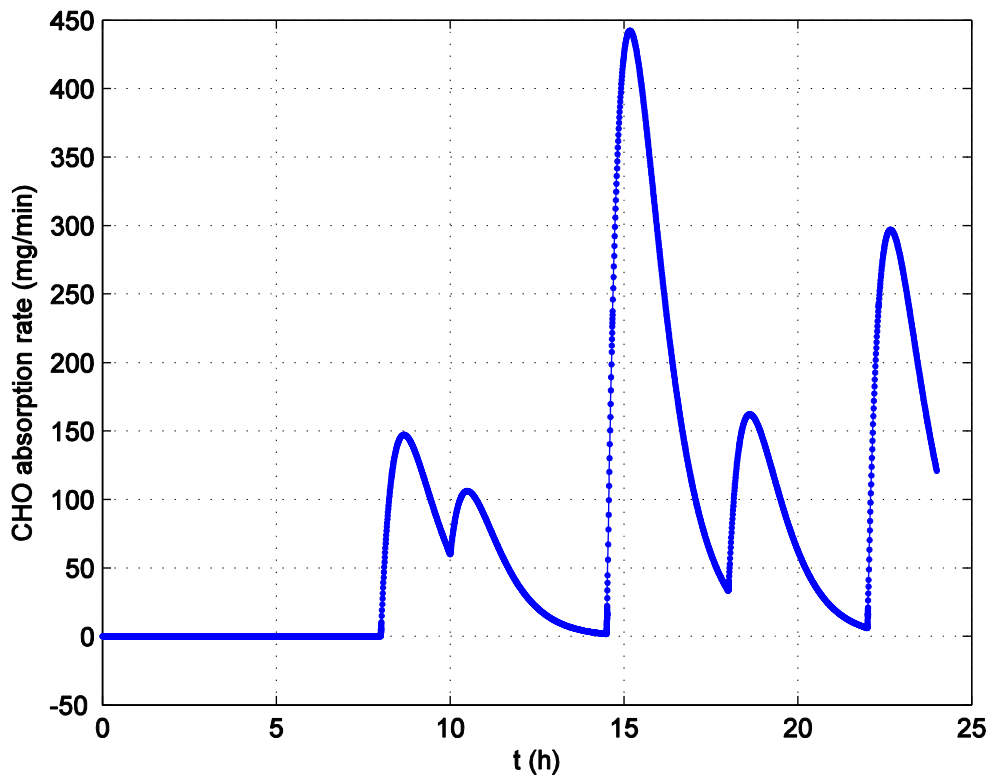


Figura 8.7. Tasa de absorción de carbohidratos (caso 2)

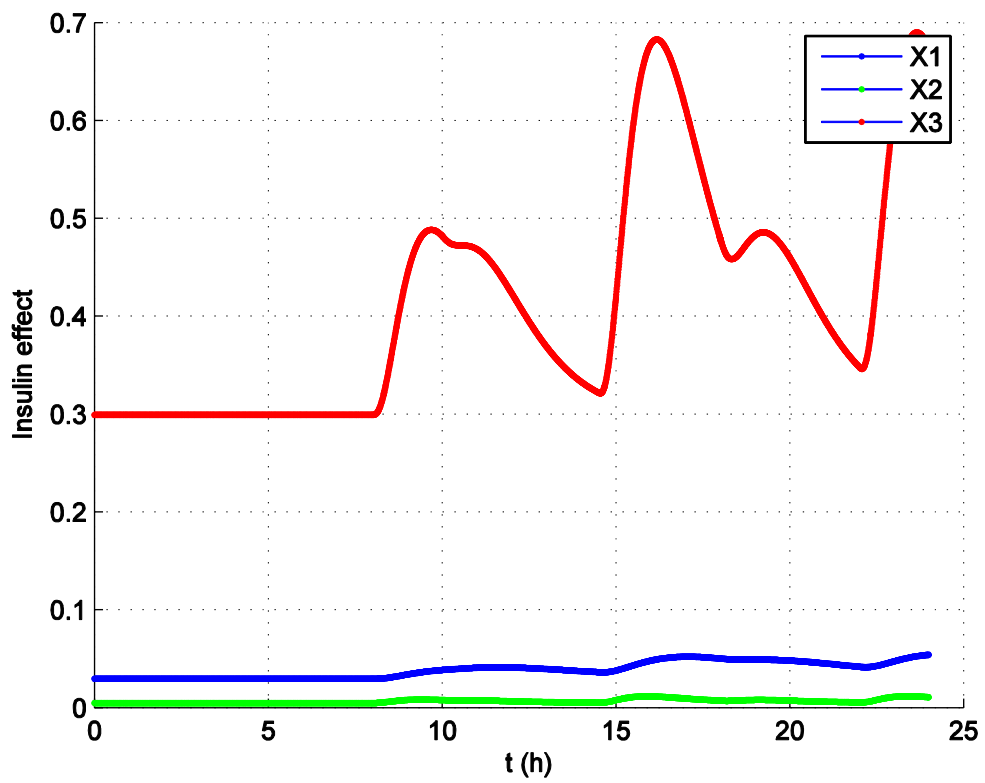


Figura 8.8. Evolución temporal del efecto de la insulina (caso 2)

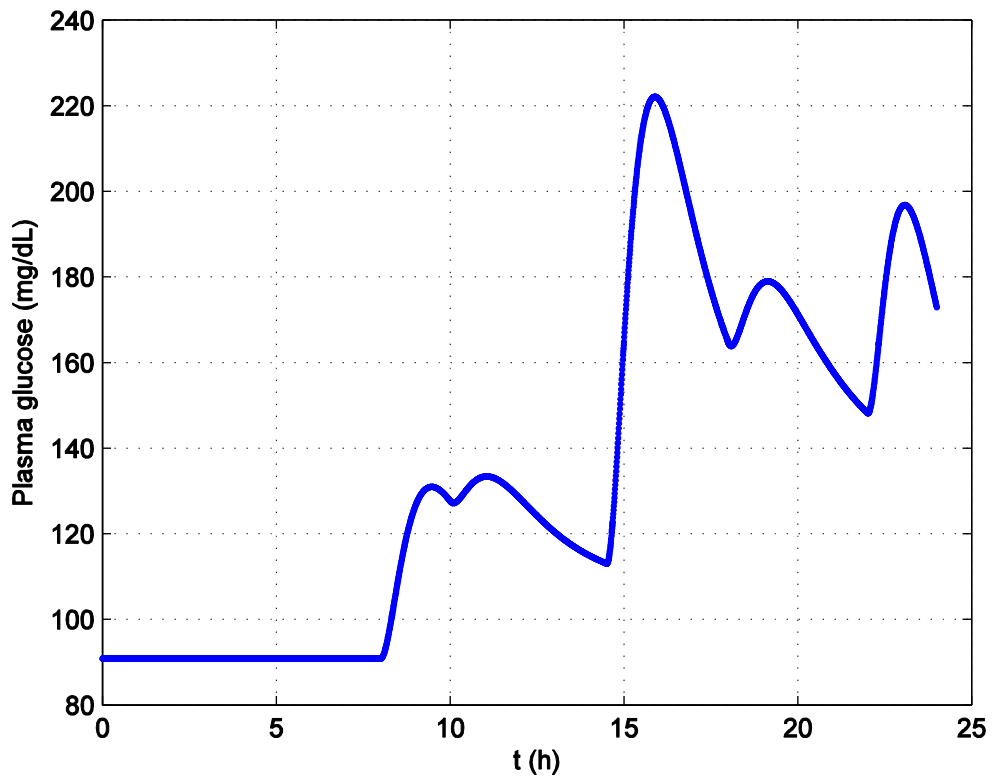


Figura 8.9. Evolución temporal de la glucosa en plasma (caso 2)

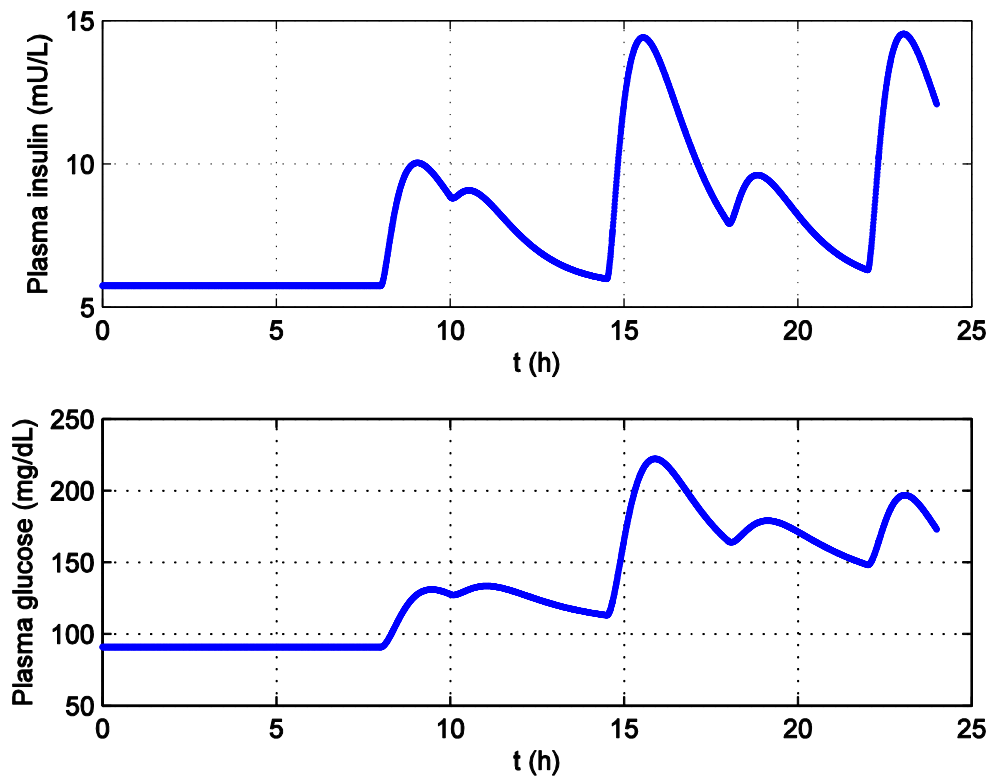


Figura 8.10. Comparación de la evolución de la insulina y glucosa en plasma (caso 2)

- Caso 3. Simulación de un día con cinco comidas. Misma situación que en el caso anterior aumentando la dosis de insulina administrada.
 - Desayuno: 8:00h, 20g de carbohidratos, 2U de insulina
 - Almuerzo: 10:00h, 10 g de carbohidratos, 1U de insulina
 - Comida: 14:30h, 60g de carbohidratos, 6U de insulina
 - Merienda: 18:00h, 20g de carbohidratos, 2U de insulina
 - Cena 22:00h, 40g de carbohidratos, 4U de insulina

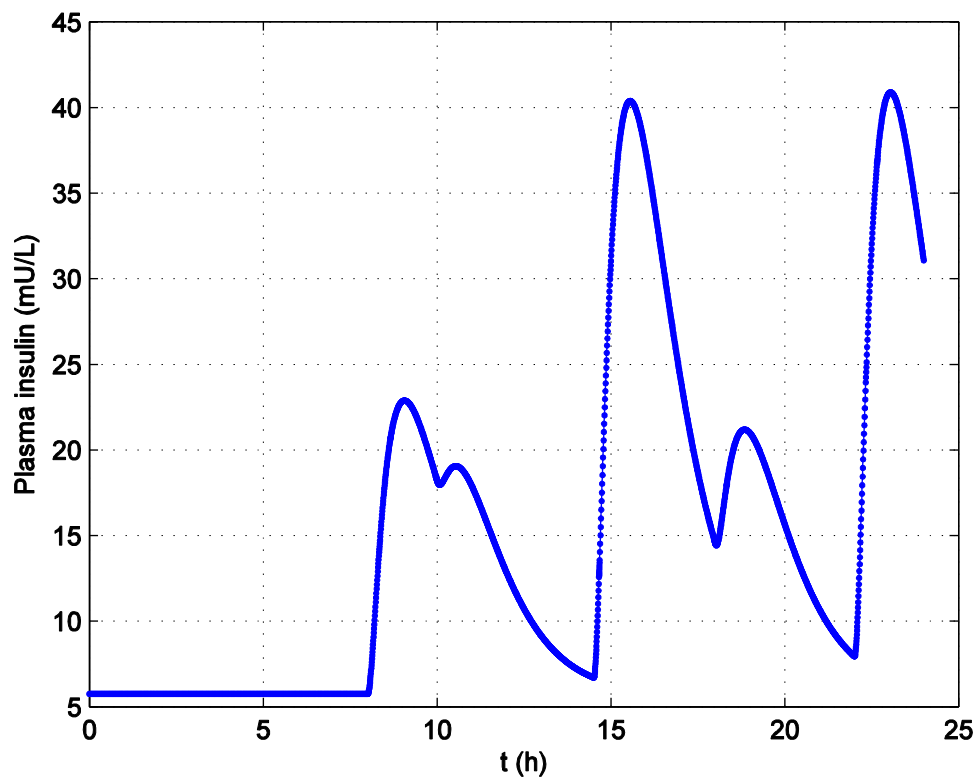


Figura 8.11. Evolución temporal de la insulina en plasma (caso 3)

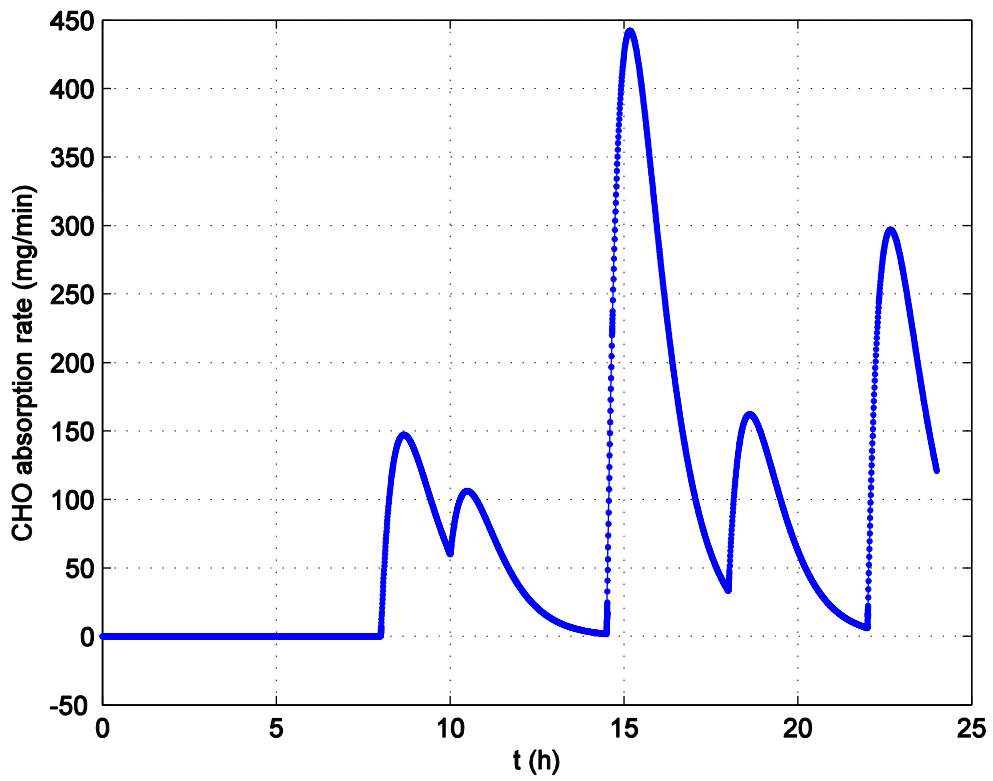


Figura 8.12. Tasa de absorción de carbohidratos (caso 3)

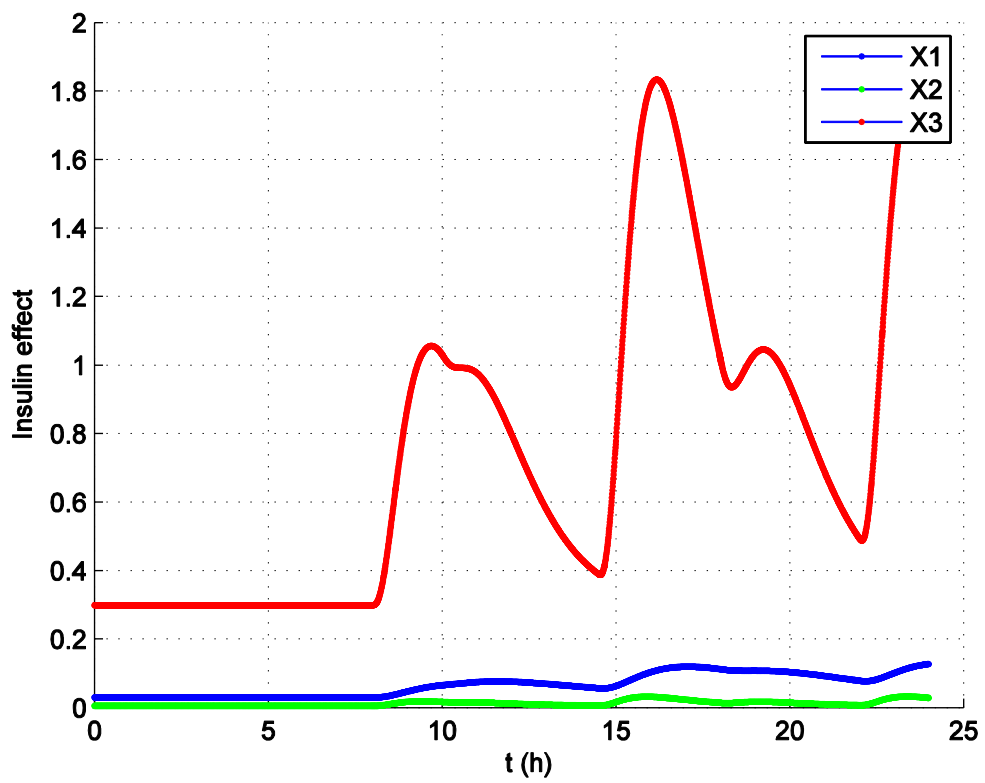


Figura 8.13. Evolución temporal del efecto de la insulina (caso 3)

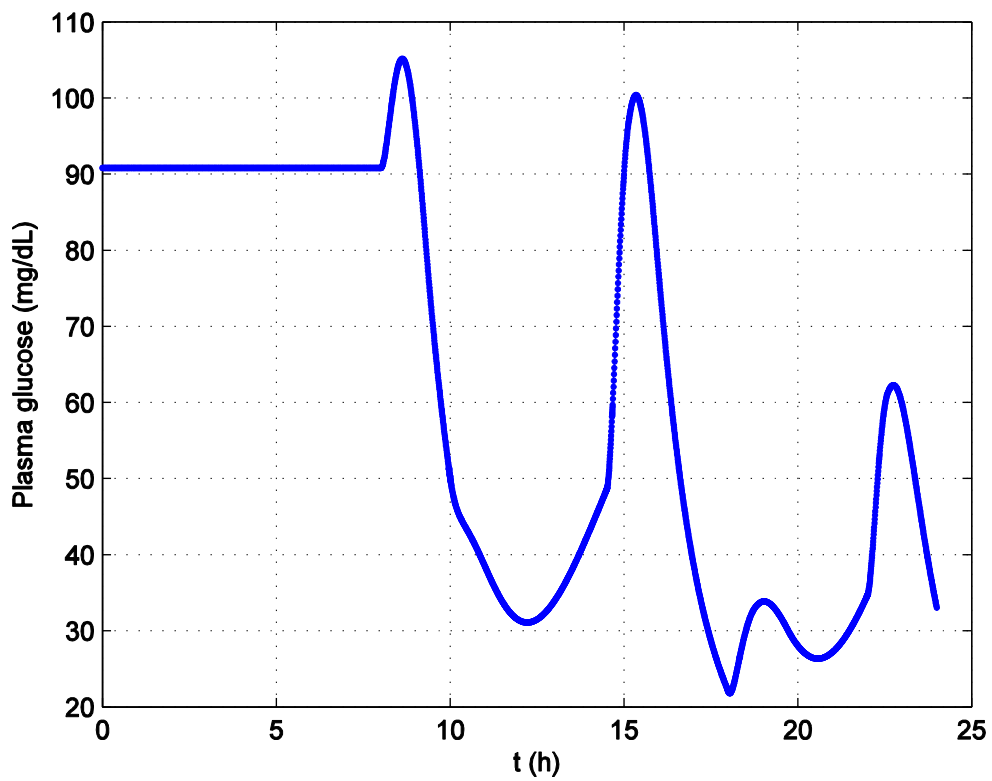


Figura 8.14. Evolución temporal de la glucosa en plasma (caso 3)

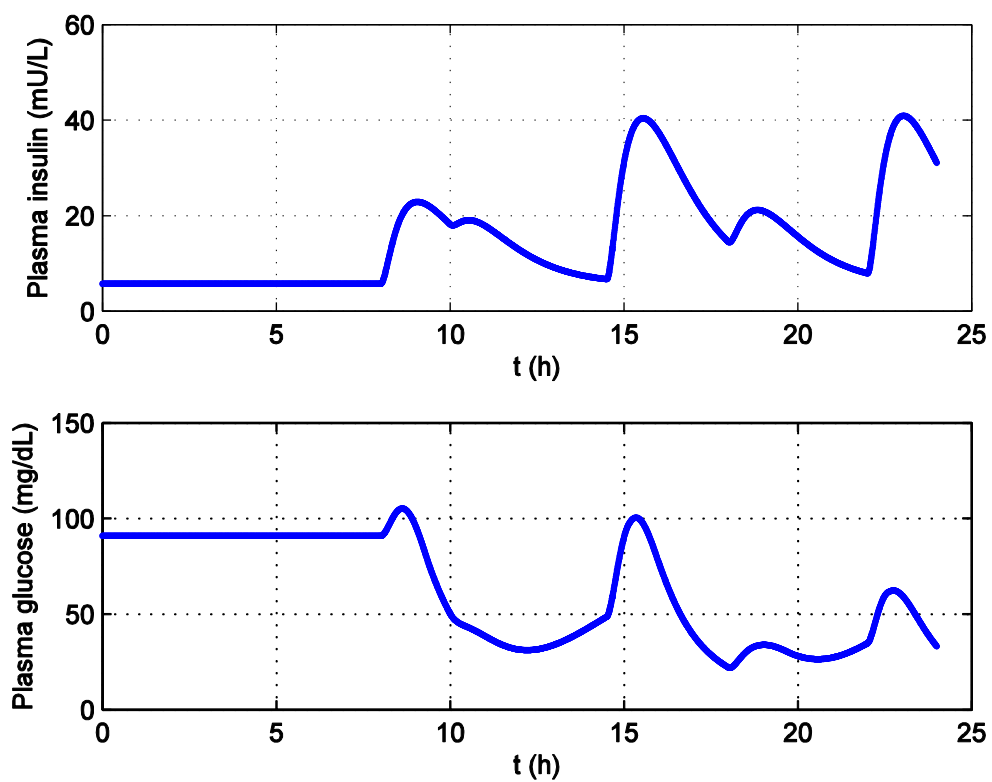


Figura 8.15. Comparación de la evolución de la insulina y glucosa en plasma (caso 3)

- Caso 4. Simulación de cuatro días con tres comidas diarias. Todos los días se realizan las mismas ingestas de carbohidratos y se administran las mismas inyecciones de insulina. A continuación se muestran los datos de uno de los días:
 - Desayuno: 8:00h, 30g de carbohidratos, 1.2U de insulina
 - Comida: 14:00h, 60g de carbohidratos, 2.4U de insulina
 - Cena 21:00h, 60g de carbohidratos, 2.4U de insulina

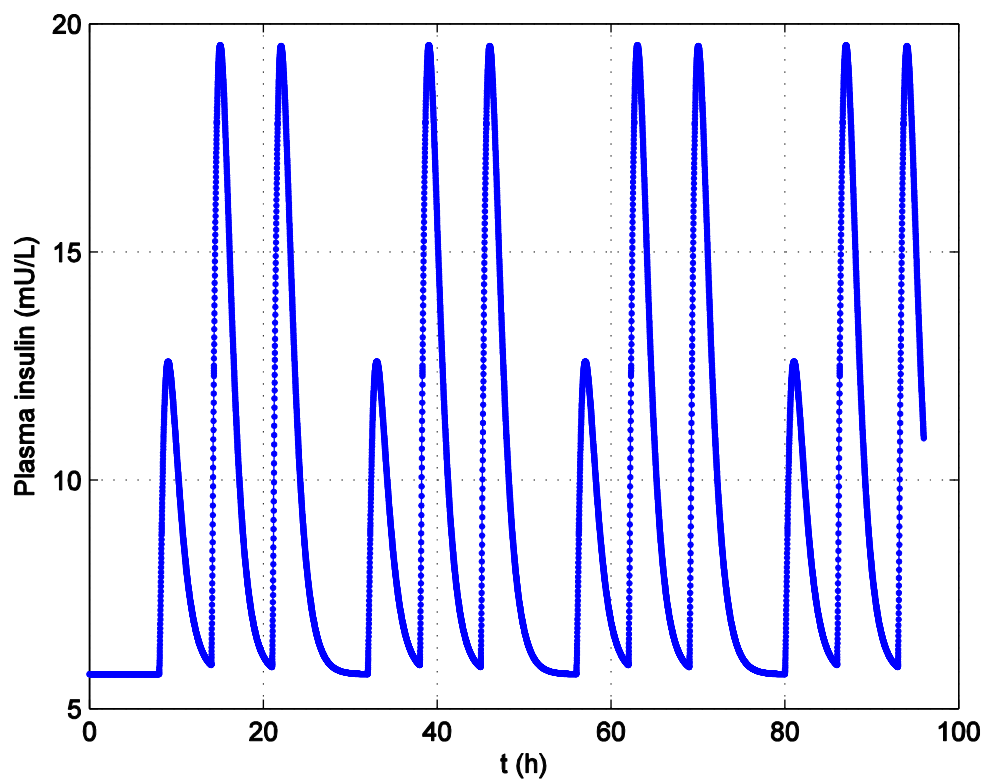


Figura 8.16. Evolución temporal de la insulina en plasma (caso 4)

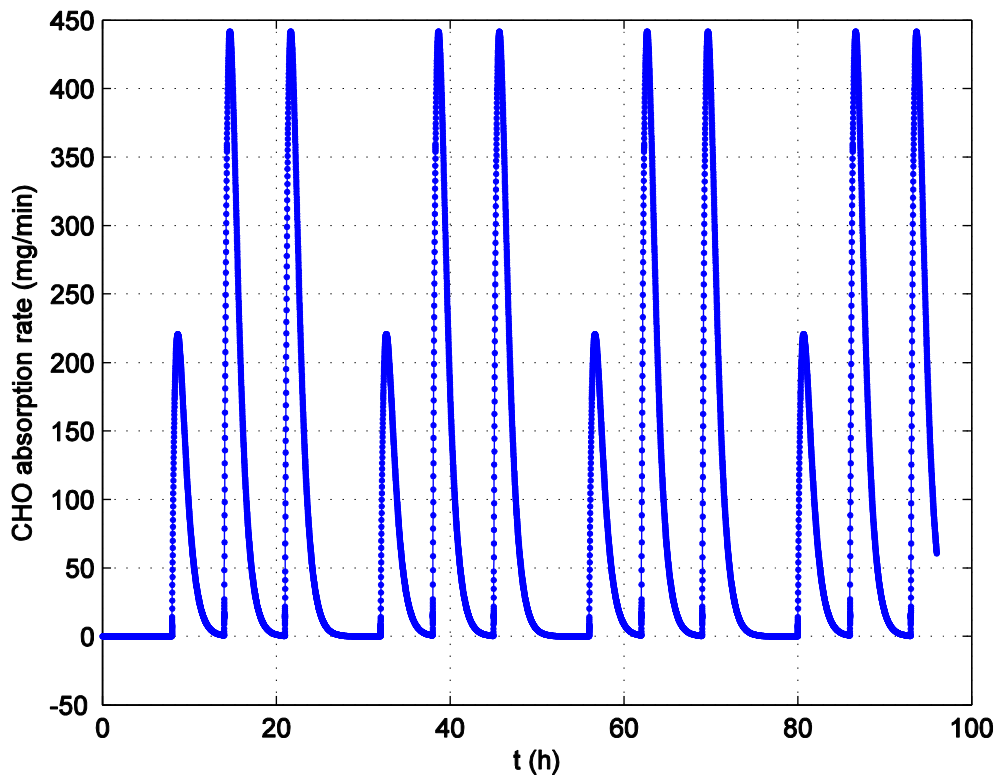


Figura 8.17. Tasa de absorción de carbohidratos (caso 4)

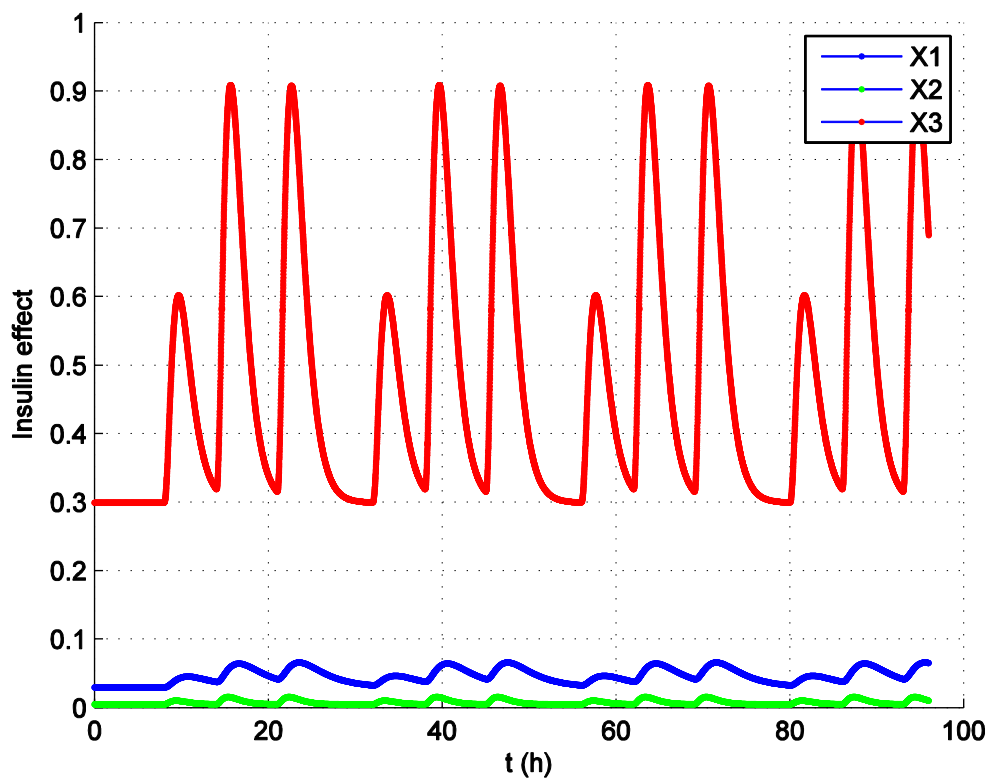


Figura 8.18. Evolución temporal del efecto de la insulina (caso 4)

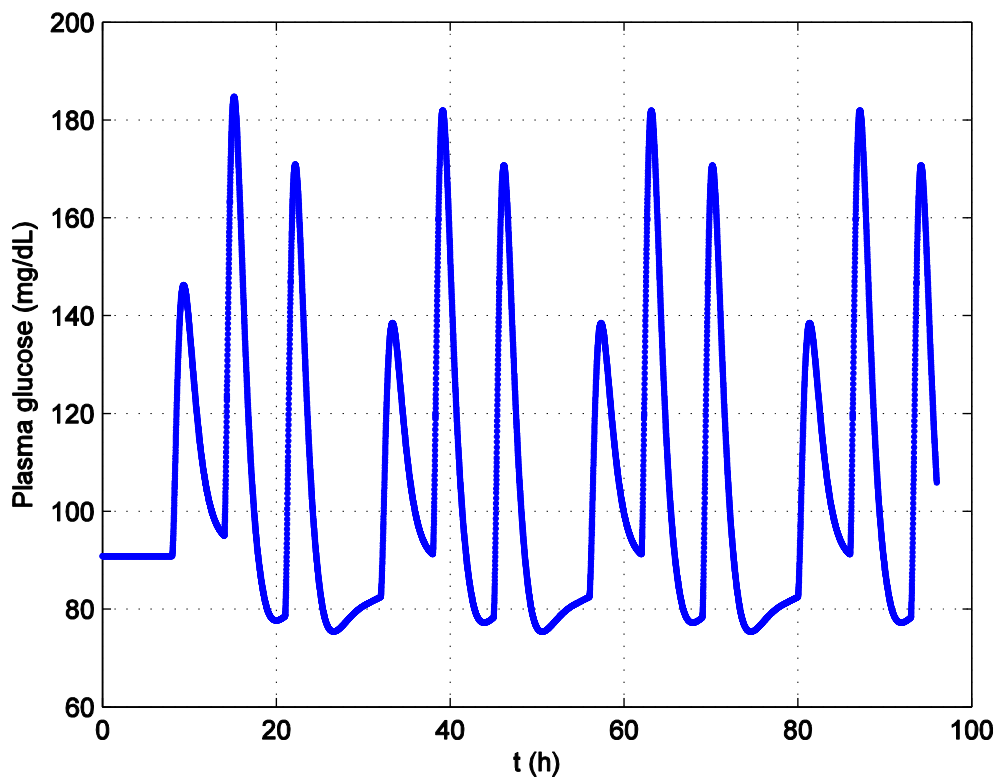


Figura 8.19. Evolución temporal de la glucosa en plasma (caso 4)

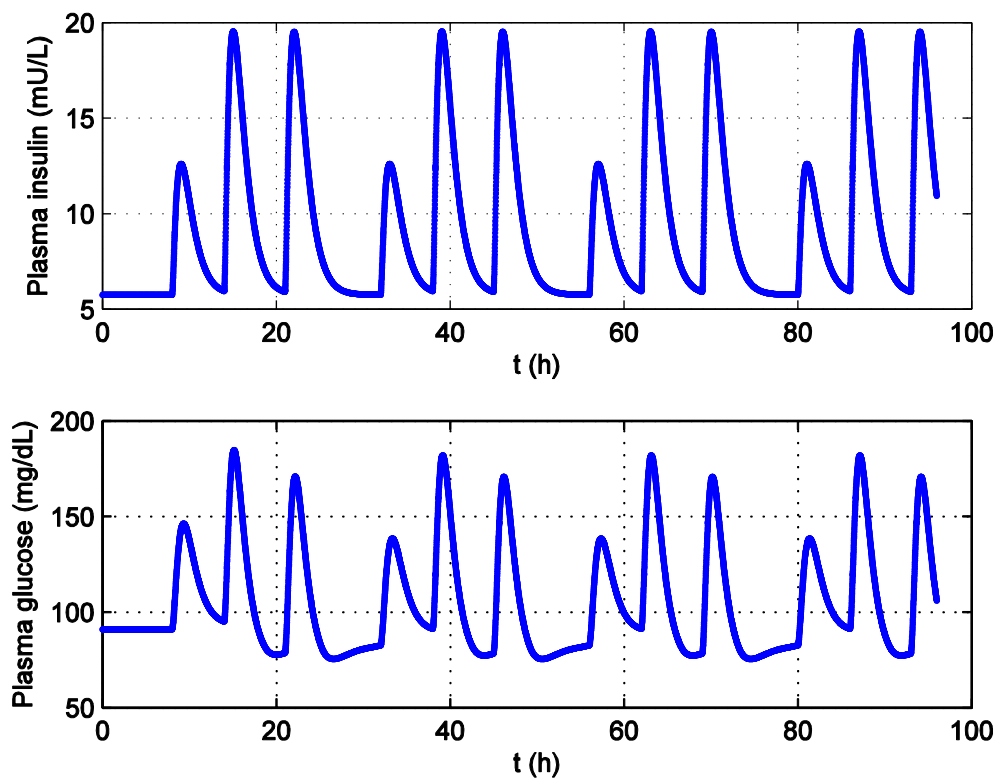


Figura 8.20. Comparación de la evolución de la insulina y glucosa en plasma (caso 4)

Capítulo 9

Conclusiones

En el capítulo anterior se han mostrado los resultados para diferentes situaciones de simulación. Analizando brevemente estas situaciones se puede comprobar que el simulador reacciona de la forma esperada ante los cambios realizados en las entradas del modelo.

Como se ha explicado en capítulos anteriores la diabetes se caracteriza por la presencia de altos niveles de glucosa en plasma. El tratamiento de la diabetes consiste principalmente en la administración de insulina exógena que permite disminuir los niveles elevados de glucosa. Por tanto, ante una disminución de la insulina administrada se espera una menor disminución de la glucosa en plasma. Por el contrario, ante un incremento de la dosis de insulina administrada se espera una mayor disminución de la glucosa en plasma.

Si el simulador funciona adecuadamente estos efectos deben ser visibles al realizar la simulación de un día aumentando o disminuyendo la dosis de insulina en cada una de las comidas. A continuación se muestran las tres gráficas que representan la evolución temporal de la glucosa en plasma, ordenadas de menor a mayor dosis de insulina administrada.

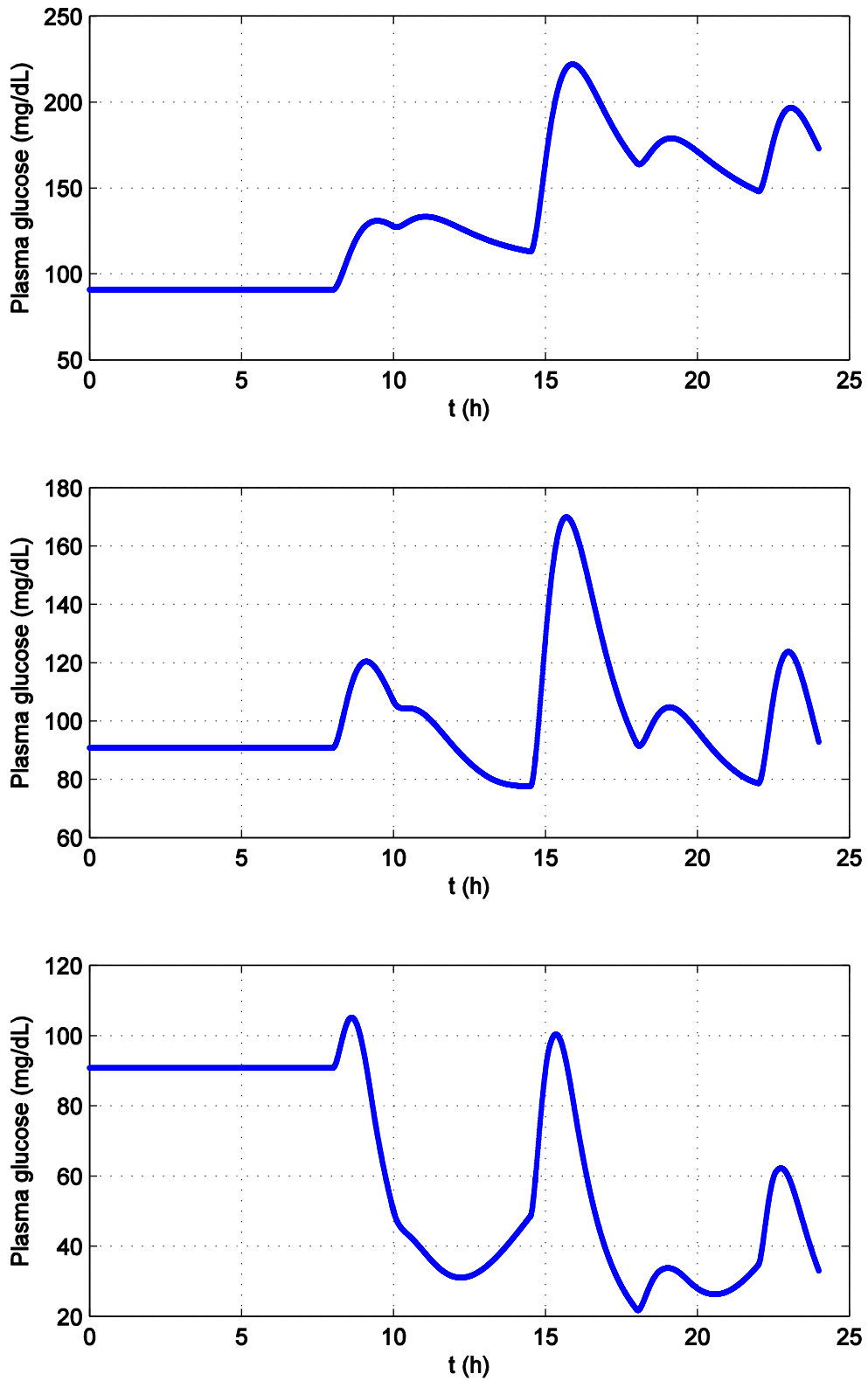


Figura 9.1. Cambios en la evolución temporal de la glucosa en plasma al aumentar las dosis de insulina administradas

En las gráficas anteriores se puede observar como los niveles de glucosa en plasma alcanzados son, en cada gráfica, menores que en la anterior. Este es el comportamiento esperado pues, como se ha indicado previamente, se ha aumentado la dosis de insulina en cada caso.

Puesto que el objetivo del presente proyecto no es realizar el análisis de los resultados de la simulación, no se va a explicar cuál de los casos anteriores es el más o menos favorable y cómo se debe reaccionar ante las diferentes situaciones para realizar un control adecuado del sistema glucorregulador.

Sin embargo, con los resultados obtenidos se puede confirmar que el simulador reacciona de la manera esperada, cumpliéndose así el objetivo del proyecto.

Capítulo 10

Trabajo futuro

Principalmente, hay dos mejoras que se pueden realizar en el simulador en un futuro para ampliar el uso del mismo.

- Selección del modelo matemático para la simulación del paciente. Actualmente el simulador está implementado con el modelo propuesto por el grupo de Cambridge. Esto limita las posibilidades de uso puesto que si se desea realizar alguna prueba con otro modelo matemático de los propuestos habría que acudir a otro simulador ya creado, modificar el realizado en el presente proyecto o implementar uno nuevo.

Además, la posibilidad de seleccionar el tipo de modelo utilizado permitiría comparar los resultados obtenidos para un mismo caso con diferentes modelos de forma sencilla, pudiendo seleccionar el que más interese finalmente para las pruebas que se vayan a realizar.

La elección del modelo debería incluirse en la interfaz ya creada. Junto al tipo de simulación podría aparecer un desplegable que permita seleccionar el modelo deseado de los disponibles. De este modo, sin necesidad de ejecutar el programa varias veces, una vez finalizada una simulación se puede cambiar el modelo e iniciar una nueva.

- Comunicación inalámbrica. Actualmente la comunicación se realiza a través del puerto serie, que facilita el desarrollo del simulador pero limita el uso del mismo, pues es necesario estar cerca y conectar un cable para poder utilizarlo. La posibilidad de utilizarlo de forma inalámbrica ampliaría las posibilidades de uso y acercaría más el proyecto a la creación del laboratorio virtual remoto, que permitiría acceder a un ordenador para realizar las pruebas sin necesidad de estar cerca del mismo.

Referencias

- [1] G. Williams, J. C. Pickup. *Handbook of Diabetes*. Blackwell Publishing, 1992.
- [2] F. Barceló Rico. *Multimodel Approaches for Plasma Glucose Estimation in Continuous Glucose Monitoring. Development of New Calibration Algorithms*. Universidad Politécnica de Valencia, 2012.
- [3] S. L. Aronoff, K. Berkowitz, B. Shreiner, L. Want. Glucose metabolism and regulation: Beyond insulin and glucagon. *Diabetes Spectrum*, vol. 17, Num. 3, pp 183-190, 2004.
- [4] J. Bondia, J. Vehí, C.C. Palerm, P.Herrero. El páncreas artificial: control automático de infusión de insulina en diabetes mellitus tipo 1. *Revista Iberoamericana de Automática e Informática Industrial*, vol.7, Núm.2, pp 5-20, 2010.
- [5] A. J. Laguna Sanz. *Model identification from ambulatory data for post-pandrial glucose control in type 1 diabetes*. Master dissertation. Universidad Politécnica de Valencia. Departamento de ingeniería de sistemas y automática, 2010.
- [6] F. G. Banting, C. H. Best. The internal secretion of the pancreas. *The Journal of Laboratory and Clinical Medicine*, vol. 7, no.5, pp – 251-276, 1992.
- [7] IDF. <http://www.idf.org/atlasmap/atlasmap>
- [8] A. Haidar, M. E. Wilinska, J. A. Graveston, R. Hovorka. Stochastic virtual population of subjects with type 1 diabetes for de assessment of closed-loop glucose contollers. *IEEE Transaction on Biomedical Engineering*, vol. 60, no. 12, pp 3524-3533, 2013.
- [9] Medtronic. <http://www.medtronic.com/>
- [10] A. H. Kadish. Automation control blood sugar a servomechanism for glucose monitoring and control. *ASAOIO Journal*, vol. 9, pp 363-367, 1963.
- [11] A. M. Albisser, B.S. Leibel, T. G. Ewart, Z. Davidovac, C. K. Botz, W. Zingg. An artificial endocrine pancreas. *Diabetes*, vol. 23, no. 5, pp 389- 396, 1974.
- [12] E. F. Pfeiffer, C. Thum, A.H. Clemens. The artificial beta cell – A continuous control of blood sugar by external regulation of insulin infusion (glucose controlled insulin infusion system). *Hormone and Metabolic Research*, vol. 6, no. 5, pp 339-342, 2002.

- [13] R. Hovorka. Continuous glucose monitoring and closed-loop systems. *Diabetic medicine*. Vol. 23, no. 1, pp 1-12, 2006.
- [14] H.R. Murphy, G. Rayman, K. Lewis, S. Kelly, B. Johal, K. Duffield, D. Fowler, P. J. Campbell, R. C. Temple. Effectiveness of continuous glucose monitoring in pregnant women with diabetes: randomized clinical trial. *British Medical Journal*, vol. 337, 2008.
- [15] M. E. Hernando, E. J. Gómez, F. Del Pozo, R. Corcoy. Diabnet: A qualitative model-based advisory system for therapy planning in gestacional diabetes. *Informatics for Health and Social Care*, vol. 21, no. 4, pp 359-374, 1996.
- [16] J. Plank, J. Blaha, J. Cordingley, M. E. Wilinska, L.J. Chassin, C. Morgan, S. Squire, M. Haluzik, J. Kremen, S. Svacina. Multicentric, randomized, controlled trial to evaluate blood glucose control by the model predictive control algorithm versus routine glucose management protocols in intensive care unit patients. *Diabetes Care*, vol. 29, no. 2, 2006.
- [17] H. C. Schaller, L. Schaupp, M. Bodelenz, M. E. Wilinska, L. J. Chassin, P. Wach, T. Vering, R. Hovorka. T. R. Pieber. On-line adaptative algorithm with glucose prediction capacity for subcutaneous closed loop control of glucose: evaluation under fasting conditions in patients with type 1 diabetes. *Diabetic Medicine*, vol.23, no. 1, pp 90-93, 2006.
- [18] A. J. Laguna Sanz. *Uncertainty in postprandial model identification in type 1 diabetes*. PhD dissertation. Universidad Politécnica de Valencia. Departamento de ingeniería de sistemas y automática, 2014.

Anexo I

Manual de usuario

1. Introducción.....	79
1.1. Organización del manual	79
1.2. Descripción de la aplicación.....	80
1.3. ¿A quién va dirigido?	80
2. Lo que se debe conocer	83
3. Especificaciones técnicas.....	85
3.1. Requisitos hardware	85
3.2. Requisitos software	86
4. Manual.....	87
4.1. Configurar por primera vez la aplicación	87
4.2. Inicio de la aplicación	90
4.3. Crear nuevo escenario	91
4.4. Guardar escenario.....	94
4.5. Modificar datos de un escenario	95
4.6. Abrir escenario guardado	96
4.7. Eliminar escenario guardado	97
4.8. Realizar la simulación.....	98
4.9. Guardar gráficas de resultados	100

Capítulo 1

Introducción

El manual de usuario es un documento con carácter formativo que trata de orientar al usuario en el uso de la aplicación. En el manual se intenta dar cobertura a todas las posibles dudas que surjan en el uso, aunque al ser una aplicación sencilla tanto en configuración como en el uso, no precisará de gran cantidad de información para hacer uso de la misma.

1.1. Organización del manual

En primer lugar se hace una breve introducción en la que se describe la aplicación para que el usuario se haga una idea del funcionamiento de la misma. A continuación se describe la clase de usuarios a los que está dirigida la aplicación y, en consecuencia, el manual.

Posteriormente se indica lo que se debe conocer para hacer uso de la aplicación. Cabe comentar que no son requisitos necesarios, pero si no se conocen determinados puntos carece de sentido la utilización del simulador.

En el siguiente apartado se describen los requerimientos técnicos necesarios para hacer uso de la aplicación, tanto a nivel de hardware como de software.

Por último, se explica paso a paso como se utiliza la aplicación. Se describen todos los elementos de la interfaz y se indica cómo guardar los datos para el análisis de resultados.

1.2. Descripción de la aplicación

A continuación se hace una breve descripción de la aplicación y se resumen las principales funciones que ofrece. No se entrará en detalles ya que la finalidad es presentar de forma breve y concisa la aplicación, permitiendo al usuario saber que puede esperar del producto.

La aplicación diseñada consta de dos programas con los que se busca simular el comportamiento de un paciente con diabetes mellitus tipo 1. Cuando se habla de simular el comportamiento de un paciente diabético se busca averiguar los niveles de glucosa e insulina en plasma tras una perturbación como es la ingesta de carbohidratos o las inyecciones de insulina.

Por una parte, se dispone del programa que permite introducir los datos necesarios para realizar la simulación a través de una interfaz gráfica. Estos datos son los carbohidratos ingeridos en las comidas, los bolos de insulina administrados y la insulina basal.

Por otra parte, se dispone del programa que realiza la simulación del metabolismo del paciente y que proporciona datos como la insulina en plasma, la tasa de absorción de carbohidratos, el efecto de la insulina o la glucosa en plasma que serán utilizados para realizar el control del sistema de regulación de glucosa.

Cada uno de los programas está disponible en un ordenador y se comunican por medio del puerto serie. De este modo los datos introducidos se envían al simulador que, una vez los haya procesado y haya obtenido los resultados, los enviará de nuevo.

Cuando finaliza la simulación se pueden observar los datos de interés en gráficas temporales que permiten el estudio del escenario simulado.

1.3. ¿A quién va dirigido?

La aplicación está diseñada para conocer el comportamiento del metabolismo de un paciente diabético ante diferentes entradas. El objetivo principal de obtener estos datos es para realizar las pruebas necesarias de los dispositivos de control destinados a crear el páncreas artificial, por tanto, la aplicación va destinada a los ingenieros que participan en el proyecto de investigación del páncreas artificial.

Sin embargo, hacer pruebas para conocer el comportamiento del paciente ante determinadas situaciones, tanto normales como extremas, puede ser muy interesante para estudiantes de endocrinología que no pueden hacer esa clase de pruebas con pacientes reales. Incluso puede ser interesante su uso para los propios pacientes y

familiares de modo que puedan conocer mejor los efectos que tienen en el organismo determinadas situaciones pudiendo evitar aquellas que supongan un peligro para la salud del paciente.

Por tanto, no está dirigido a un usuario concreto, puede tratarse tanto de personas con conocimientos de programación que sean capaces de comprender todo el código, como puede ser un ingeniero, o personas que carezcan de esos conocimientos como un médico o un paciente o familiar. Por este motivo, se ha diseñado la aplicación de forma que sea sencilla de utilizar y se adjunta el manual para que no suponga ningún tipo de dificultad el uso de la misma.

Capítulo 2

Lo que se debe conocer

La aplicación simula el comportamiento de un paciente diabético proporcionando unas gráficas en las que se puede observar la evolución de la insulina y la glucosa en plasma. Sin embargo, no está diseñada para interpretar esos resultados por lo que debe ser el propio usuario el que los interprete.

Para que la aplicación sea de utilidad el usuario debe tener ciertos conocimientos que faciliten la interpretación de los resultados. A continuación se indican esos conocimientos que se deben adquirir para dar un uso adecuado al simulador:

- Diabetes. Puesto que la aplicación simula el comportamiento de un paciente diabético es necesario conocer en qué consiste la enfermedad para entender qué ocurre en caso de que no se trate al paciente.
- Metabolismo de la glucosa. Es importante conocer las hormonas que intervienen en el metabolismo de la glucosa y cómo lo hacen.
- Niveles normales de glucosa. Se deben conocer cuáles son los niveles adecuados de glucosa en plasma para poder identificar situaciones peligrosas en las que esos se alcanzan niveles que ponen en riesgo la salud del paciente.
- Tratamiento de la diabetes. Debe conocerse en qué consiste el tratamiento y qué debe hacerse cuando se dan situaciones peligrosas.

Conociendo los puntos indicados anteriormente es posible interpretar adecuadamente los resultados para poder utilizarlos posteriormente para comprobar la efectividad y seguridad de los dispositivos de control así como las situaciones en las que se alcanzan niveles peligrosos para prevenirlas o, en caso de darse, disponer de los medios para contrarrestarlas.

Capítulo 3

Especificaciones técnicas

Las especificaciones técnicas definen las normas, exigencias y procedimientos a ser empleados y que se deben tener en cuenta durante el uso de la aplicación. Para poder instalar y utilizar correctamente la aplicación se deben cumplir unos requisitos. Dichos requisitos se dividen en requisitos de hardware que son los que el equipo debe satisfacer y en requisitos software que son los programas que deben estar instalados en el equipo.

3.1. Requisitos hardware

Los requisitos hardware son las características que debe tener el ordenador para poder soportar y ejecutar una aplicación concreta o un dispositivo específico. En este caso particular los requisitos de hardware que deben cumplir los dos ordenadores necesarios vienen impuestos por el programa en el que se ha desarrollado la aplicación y son los siguientes:

- Dos ordenadores. Estos deben contar con las siguientes características:
 - Procesador: cualquier procesador Intel o AMD x68.
 - Disco duro: se requiere al menos 1 GB de espacio disponible.
 - Memoria RAM: al menos 1 GB, preferiblemente de 2 GB a 3 GB.
- Cable serie. En caso de que el ordenador no disponga de puerto serie se puede utilizar un adaptador que permita realizar este tipo de conexión a través de otro puerto, por ejemplo, el USB.

3.2. Requisitos software

Los requisitos software son los programas requeridos por la aplicación para poder realizar su ejecución. El ordenador en el que se ejecute la aplicación debe disponer de:

- Sistema Operativo Windows XP o superior. Cabe señalar que podría utilizarse otro sistema operativo como Mac OS o Linux pero para ello sería necesario modificar algunas partes del código para poder establecer la comunicación.
- Matlab R2009 o superior. Puesto que la programación de la aplicación se ha desarrollado en Matlab es imprescindible disponer en ambos ordenadores del programa para poder ejecutarla.

Capítulo 4

Manual

4.1. Configurar por primera vez la aplicación

Para poder hacer uso del simulador en primer lugar es necesario conectar ambos ordenadores a través del cable para poder establecer la comunicación entre los mismos a través del puerto serie.

Actualmente, en la mayoría de los ordenadores, los puertos USB han reemplazado al puerto serie por su mayor velocidad. La mayor parte de ordenadores están conectados a dispositivos externos a través de USB y, a menudo, ni siquiera llegan a tener puerto serie. Sin embargo, aunque físicamente no se disponga de un puerto serie, es posible realizar este tipo de conexión a través del puerto USB con un adaptador adecuado.

A continuación se abre el archivo Interfaz en el ordenador en el que se va a realizar la introducción de datos y el análisis de resultados y el archivo Simulador en el ordenador que simula al paciente. Para poder ejecutar ambos programas es necesario comprobar que el nombre del puerto serie que aparece en el código es el mismo que el del puerto serie al que se ha realizado la conexión. Esto sólo será necesario realizarlo la primera vez que se ejecute la aplicación.

Para realizar esta comprobación se abre el Administrador de Dispositivos de Windows al que se puede acceder a través del Panel de Control. Ahí se podrán observar los puertos serie disponibles en el ordenador y el nombre de los mismos que será del tipo COMx, siendo x el número del puerto. En la figura 4.1 se puede observar la pantalla del administrador de dispositivos y un puerto serie llamado COM1.

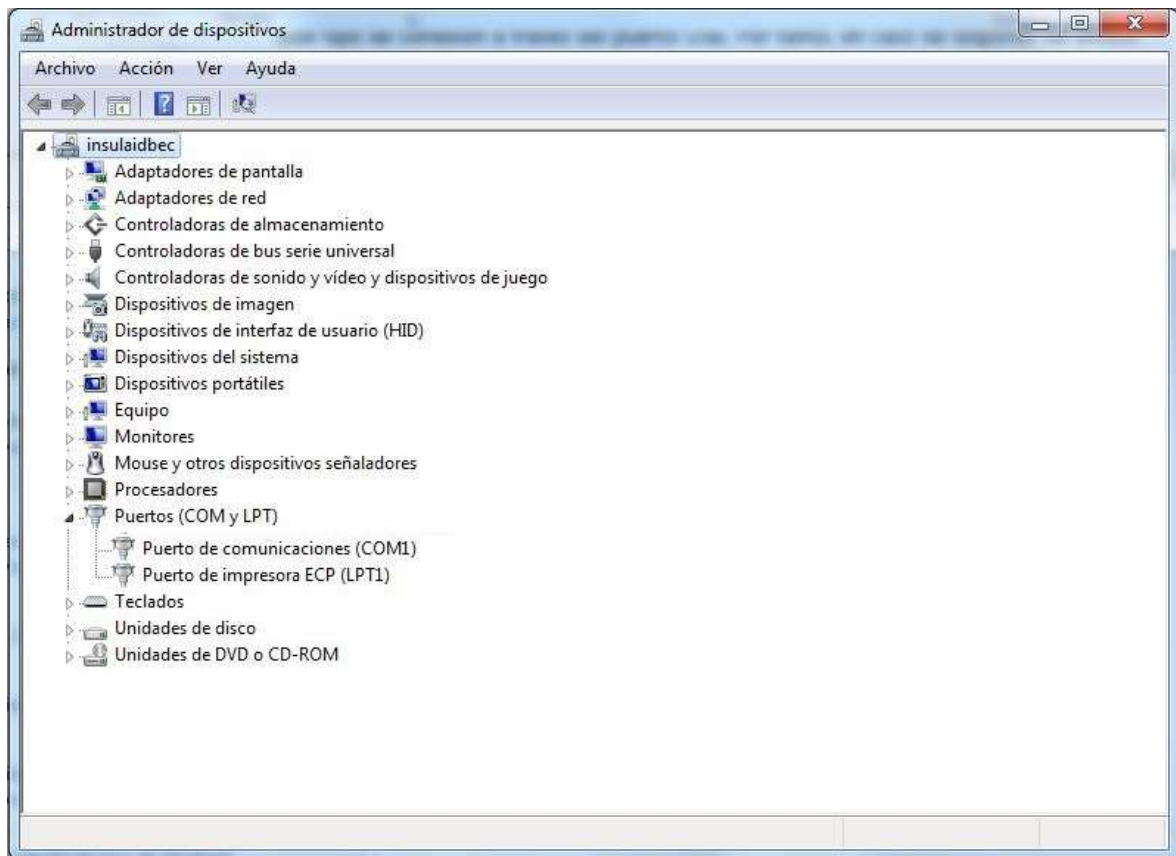


Figura 4.1. Administrador de dispositivos de Windows

Una vez conocido el nombre de ambos puertos es necesario acceder al código de ambos programas para comprobar el nombre. En primer lugar se explica cómo comprobar el nombre del puerto serie del ordenador que contiene el programa Interfaz. Para ello se busca en el código la línea 1472, en esta línea se establece el nombre del puerto y es la que, en caso necesario debe modificarse. El resto del código no es necesario modificarlo. En la figura 4.2 se puede ver la parte del código correspondiente, cabe señalar que lo que aparece en verde son comentarios y no forman parte del código que se ejecuta. En la línea indicada se puede observar en morado el nombre del puerto, en caso de no ser el mismo nombre que el del ordenador se debe modificar por el nombre correspondiente. En el caso de ejemplo habría que cambiar COM2 por COM1.

```

1472 - s1=serial('COM2'); %Cambiar COM2 por el puerto serie correspondiente del ordenador
1473 - set(s1,'Timeout',0.01);
1474 - fopen(s1); %abrir puerto serie

```

Figura 4.2. Modificar nombre puerto serie del programa Interfaz

A continuación es necesario guardar los datos, para ello se pulsa el botón “Save” de la parte superior izquierda de la pantalla.



Figura 4.3. Guardar nombre puerto serie Interfaz

Una vez hecho esto, se hace la misma comprobación para el otro ordenador que contiene el programa Simulador. En este caso podremos observar las mismas líneas de código al principio del todo, por lo que no será necesario buscar ninguna línea.

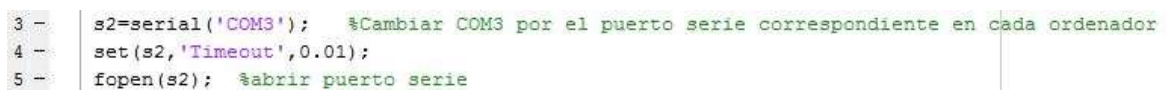


Figura 4.4. Modificar nombre puerto serie del programa Simulador

Una vez modificado el nombre por el puerto correspondiente en este caso, se guarda del mismo modo que en el caso anterior.

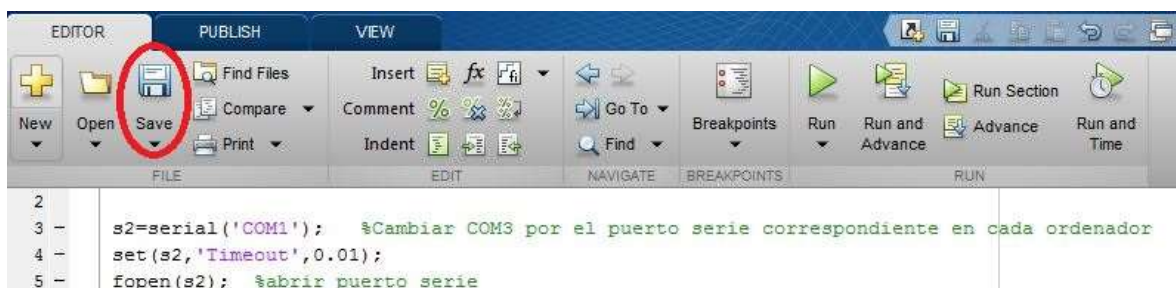


Figura 4.5. Guardar nombre puerto serie Simulador

4.2. Inicio de la aplicación

Para realizar una simulación es necesario abrir el programa Interfaz en el ordenador en el que se vaya a realizar la introducción de datos y el programa Simulador en el ordenador en el que se vaya a realizar la simulación.

Una vez abiertos ambos programas se podrá comenzar a utilizar la aplicación. En primer lugar se ejecuta el Simulador y, a continuación, la Interfaz. Para ello se pulsa el botón Run en ambos ordenadores.

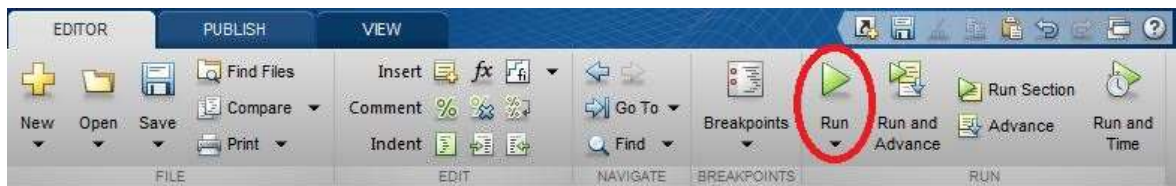


Figura 4.6. Ejecutar aplicación

Una vez ha comenzado la ejecución del programa Simulador no es necesario volver a tocar nada en el mismo. Al ejecutarse el programa Interfaz aparecerá una ventana como la que se muestra en la Figura 4.7. En caso de que ya se haya utilizado previamente la aplicación y se hayan guardado escenarios, aparecerán los nombres de los mismos en la lista “Escenarios guardados”.

Un escenario es el conjunto de datos necesarios para realizar la simulación del paciente con diabetes mellitus tipo 1. Los datos que se introducen al crear un escenario son las comidas que realiza el paciente, las inyecciones de insulina y la insulina basal.

En caso de que nunca se haya utilizado la aplicación, que no haya escenarios guardados o que se desee crear un escenario nuevo ir al punto 4.3. Crear nuevo escenario.

En caso de que se disponga de escenarios guardados y se desee abrir uno de los mismos ir al punto 4.6. Abrir escenario guardado.

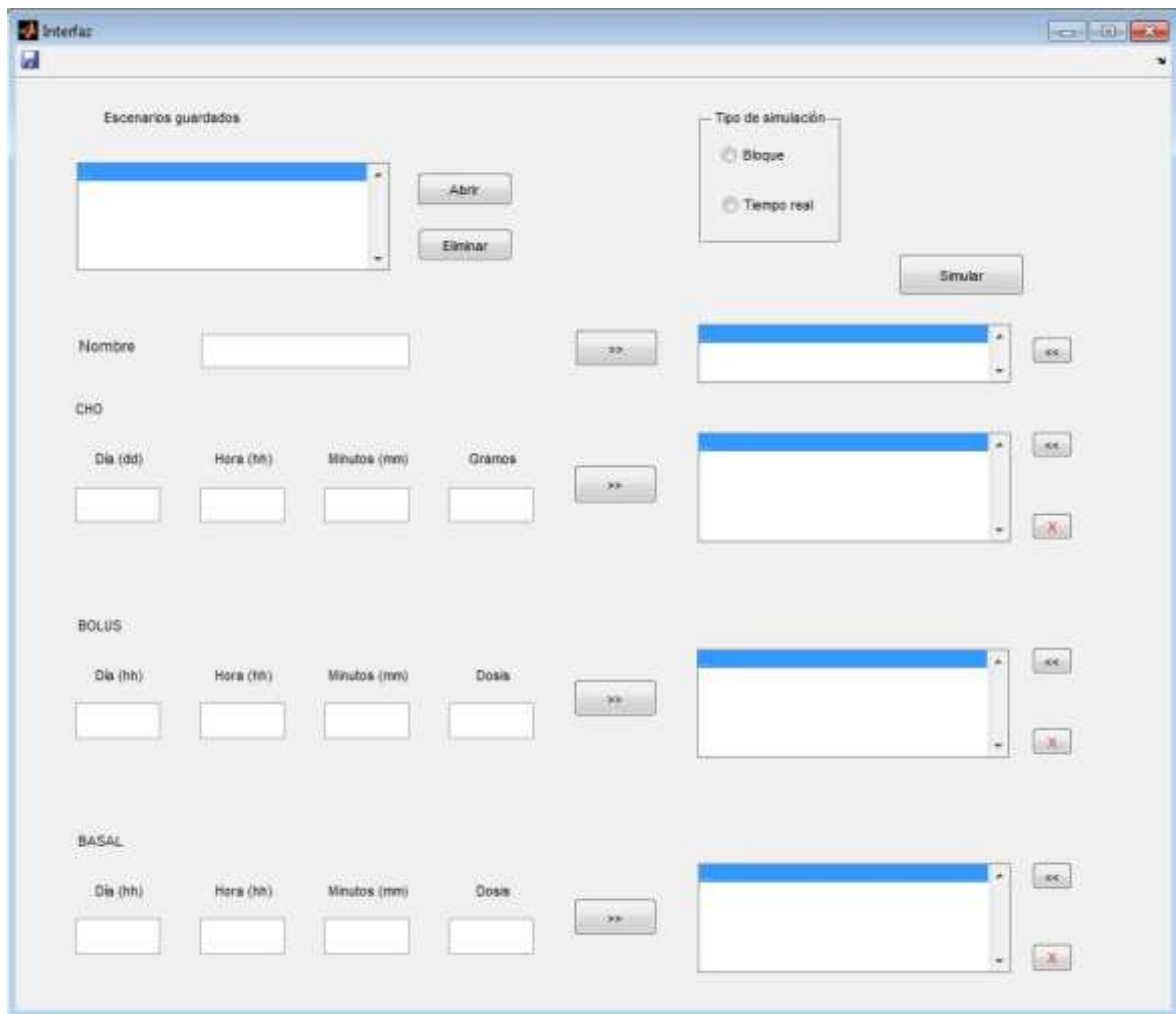


Figura 4.7. Pantalla de introducción de datos

4.3. Crear nuevo escenario

Crear un nuevo escenario significa introducir nuevos datos para realizar la simulación de un paciente diabético. Para ello es necesario completar la interfaz.

En primer lugar se pide el nombre del escenario que se desea crear. Si se desea guardar el escenario guardado se escribe el nombre en el cuadro de texto. Una vez escrito el nombre se pulsa el botón añadir. En la Figura 4.8 se puede observar cuál es este botón. Al pulsar el botón se observa como aparece el nombre del escenario en la lista.



Figura 4.8. Añadir nombre del escenario



Figura 4.9. Nombre del escenario

A continuación se deben completar los datos de las comidas que realiza el paciente durante los días en los que se realiza la simulación. Para ello es necesario indicar para cada comida el día, la hora, los minutos (todos ellos con dos dígitos) y los gramos de carbohidratos ingeridos (en este caso no es importante el número de dígitos). Por ejemplo, si se realiza el desayuno del primer día a las ocho de la mañana y se ingieren 30 gramos de carbohidratos habría que completar los cuadros como indica la Figura 4.10.

CHO			
Día (dd)	Hora (hh)	Minutos (mm)	Gramos
01	08	00	30

Figura 4.10. Ingreso de datos de una comida

Una vez introducidos los datos de una comida se pulsa el botón de añadir y se observa como la comida se añade a la lista. Cuando ya se han introducido varias comidas se observa una lista como la de la Figura 4.11.

CHO

Día (dd)	Hora (hh)	Minutos (mm)	Gramos	
03	21	00	60	>>

Día 02 08:00h 30g
 Día 02 14:00h 60g
 Día 02 21:00h 60g
 Día 03 08:00h 30g
 Día 03 14:00h 60g
 Día 03 21:00h 60g

Figura 4.11. Comidas añadidas al escenario

Una vez se han completado los datos de las ingestas se completan los datos de las inyecciones de insulina subcutánea. Para ello es necesario indicar para cada inyección el día, la hora, los minutos (todos ellos con dos dígitos) y la dosis de insulina administrada (en este caso no es importante el número de dígitos). Por ejemplo, si se realiza una inyección el primer día a las ocho de la mañana y se administra una dosis de 3 U habría que completar los cuadros como indica la Figura 4.12.

BOLUS

Día (hh)	Hora (hh)	Minutos (mm)	Dosis
01	08	00	3

Figura 4.12. Ingreso de datos de una inyección de insulina

Una vez introducidos los datos de una inyección se pulsa el botón de añadir y se observa como se añade a la lista. Cuando ya se han introducido varios bolos de insulina se observa una lista como la de la Figura 4.13.

BOLUS

Día (hh)	Hora (hh)	Minutos (mm)	Dosis	
03	21	00	6	>>

Día 02 08:00h 3U
 Día 02 14:00h 6U
 Día 02 21:00h 6U
 Día 03 08:00h 3U
 Día 03 14:00h 6U
 Día 03 21:00h 6U

Figura 4.13. Dosis de insulina añadidas al escenario

Por último, se deben añadir los diferentes valores de insulina basal, del mismo modo que en los dos casos anteriores. Si por ejemplo tuviéramos una insulina basal constante de $0,00667 \text{ U min}^{-1}$ habría que completar los datos como se muestra en la Figura 4.14.



The image shows a software interface titled "BASAL" with four input fields. The first field is labeled "Día (hh)" and contains the value "01". The second field is labeled "Hora (hh)" and contains "00". The third field is labeled "Minutos (mm)" and contains "00". The fourth field is labeled "Dosis" and contains "0.00667".

Figura 4.14. Ingreso de datos de insulina basal

Una vez introducidos los datos se pulsa el botón añadir y se observa la dosis de insulina añadida como se muestra en la Figura 4.15.



The image shows the same "BASAL" input form as in Figure 4.14, but with an additional button labeled ">>" to the right of the "Dosis" field. To the right of the form is a list box containing a single entry: "Día 01 00:00h 0.00667U/min".

Figura 4.15. Datos de insulina basal añadidos al escenario

Una vez completos los datos del escenario se puede guardar (ir al punto 4.4 Guardar escenario guardado) o realizar la simulación (ir al punto 4.8 Realizar la simulación).

4.4. Guardar escenario


Una vez se ha completado un escenario es posible guardarlo para utilizarlo en otra ocasión sin necesidad de añadir volver a añadir todos los datos a través de la interfaz. Para ello hay que pulsar el botón de guardar  de la barra de herramientas de la pantalla de la interfaz. Una vez pulsado el botón de guardar se observa como se incluye el nombre del escenario guardado en la lista de escenarios guardados.



Figura 4.16. Escenario guardado

4.5. Modificar datos de un escenario

En caso de que se desee eliminar alguno de los datos añadidos a las listas se debe seleccionar el elemento que se desee eliminar. Una vez seleccionado se pulsa el botón eliminar y se observa como el elemento desaparece de la lista.

Por ejemplo, si en el escenario creado en el apartado anterior se deseara eliminar la última comida se seleccionaría esa comida y se pulsaría el botón marcado en la Figura 4.17.



Figura 4.17. Eliminar elemento seleccionado

En caso de que se desee modificar alguno de los datos añadidos a las listas se debe seleccionar el elemento que se desee editar. Una vez seleccionado se pulsa el botón editar y se observa como los datos del elemento seleccionado aparecen en los cuadros de introducción de datos.

Por ejemplo, si se desea modificar la última comida del escenario creado en el apartado anterior se selecciona la comida y se pulsa el botón editar. Una vez aparezcan los datos en

el cuadro de texto se puede modificar los datos que se deseen. En este caso se van a modificar los gramos, en lugar de 60g se va a poner que se ingieren 30g.

CHO

Día (dd)	Hora (hh)	Minutos (mm)	Gramos
03	21	00	30

>>

- Día 02 08:00h 30g
- Día 02 14:00h 60g
- Día 02 21:00h 60g
- Día 03 08:00h 30g
- Día 03 14:00h 60g
- Día 03 21:00h 60g

<< X

Figura 4.18. Editar elemento

Una vez se han modificado los datos se añade a la lista del mismo modo que al realizar la introducción de datos para crear un nuevo escenario. Una vez se ha añadido a la lista se observa como se sustituye el elemento seleccionado por el nuevo.

CHO

Día (dd)	Hora (hh)	Minutos (mm)	Gramos
03	21	00	30

>>

- Día 02 08:00h 30g
- Día 02 14:00h 60g
- Día 02 21:00h 60g
- Día 03 08:00h 30g
- Día 03 14:00h 60g
- Día 03 21:00h 30g
- Día 03 21:00h 60g

<< X

Figura 4.19. Elemento editado

4.6. Abrir escenario guardado

Para abrir un escenario guardado se selecciona el escenario que se desea abrir y se pulsa el botón "Abrir" que aparece al lado de la lista de escenarios. En el momento en el que se pulse el botón de abrir se rellenarán todas las listas con los datos del escenario como se muestra en la Figura 4.20.

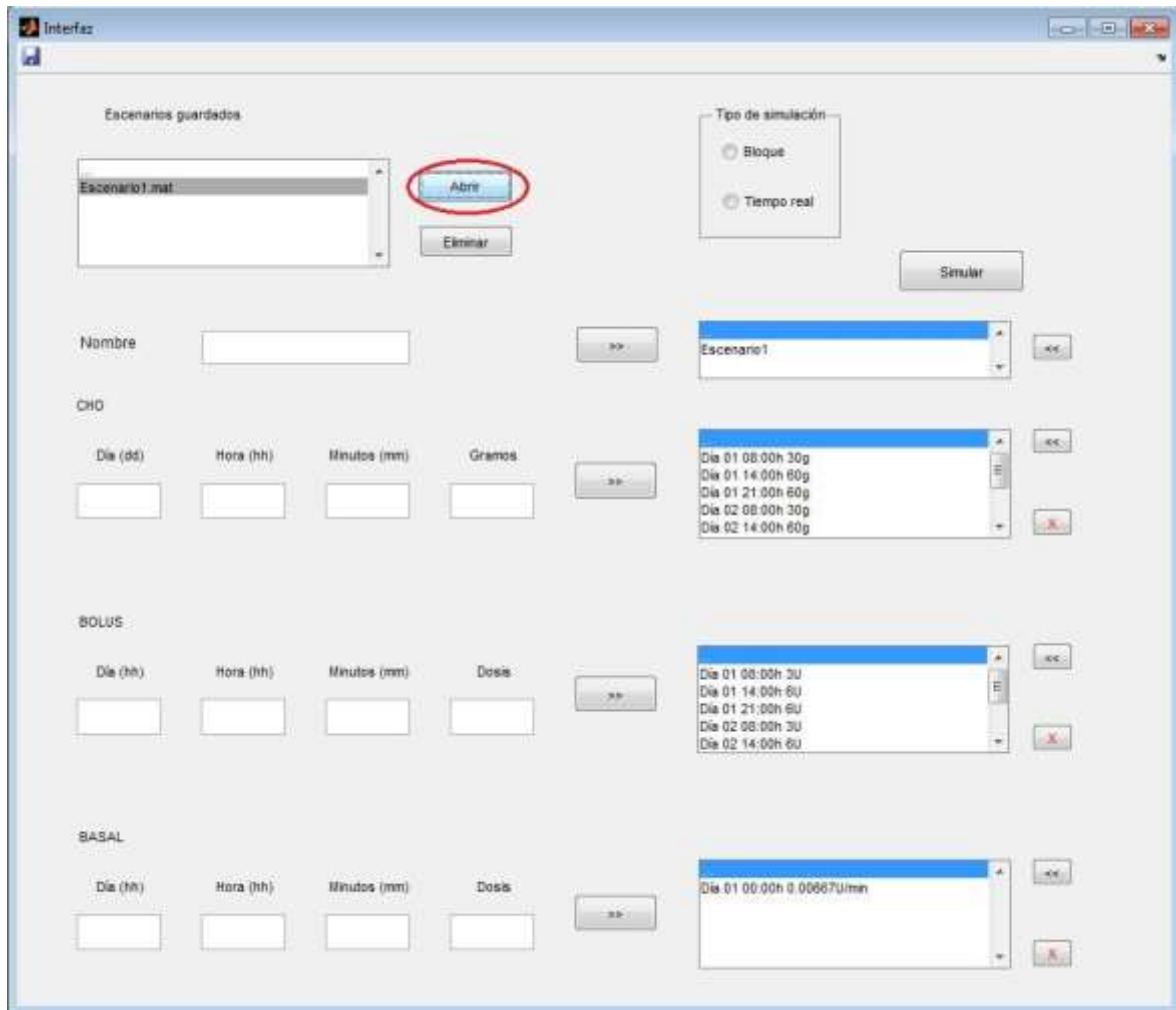


Figura 4.20. Abrir escenario guardado

4.7. Eliminar escenario guardado

Para eliminar un escenario guardado se selecciona el escenario que se desea eliminar y se pulsa el botón “Eliminar” que aparece al lado de la lista de escenarios. En el momento en el que se pulse el botón desaparecerá el escenario de la lista.

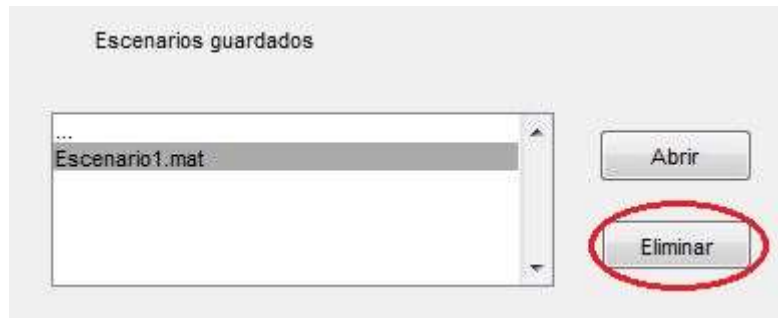


Figura 4.21. Eliminar escenario seleccionado

4.8. Realizar la simulación

Una vez introducidos todos los datos de entrada es posible realizar la simulación. Para ello se debe decidir qué tipo de simulación se desea realizar en el panel de la parte superior izquierda que se muestra en la Figura 4.22.

- Simulación en bloque. Si se selecciona este tipo se realizará la simulación y, una vez se hayan obtenido todos los datos, se mostrarán automáticamente las gráficas con los resultados obtenidos.
- Simulación en tiempo real. Si se selecciona este tipo de simulación se mostrará un segundo panel, donde se introducirá cada cuántos minutos se quieren recibir los datos de la simulación y la aceleración. En caso de que no se desee acelerar la simulación se introducirá un 1. Si se desea acelerar el proceso, para que se realice por ejemplo, en la mitad de tiempo se introducirá un 2. En la Figura 4.23 se muestran los datos para realizar una simulación en tiempo real sin aceleración recibiendo datos cada 20 minutos.

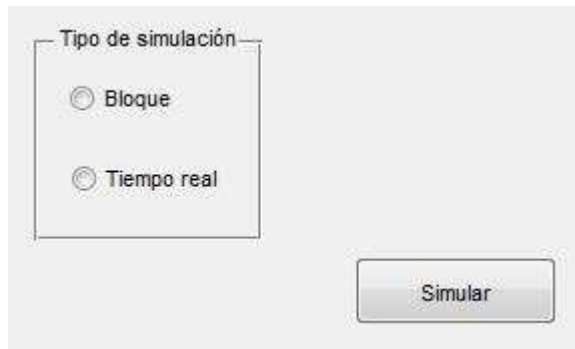


Figura 4.22. Selección del tipo de simulación

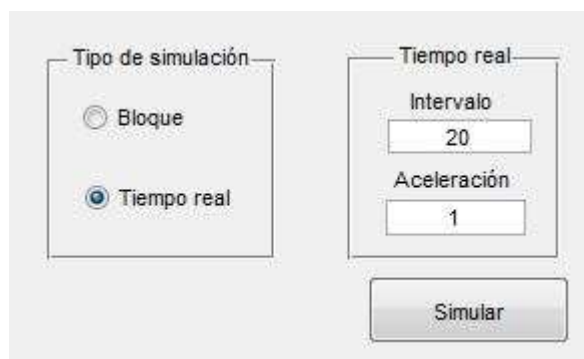



Figura 4.23. Simulación en tiempo real sin aceleración recibiendo datos cada 20 minutos

Una vez seleccionado el tipo de simulación se pulsa el botón “Simular”. En ese momento comienza a enviarse los datos al ordenador que realiza la simulación. Cuando finalice la simulación, o pase el tiempo seleccionado se mostrarán las gráficas que muestran la evolución temporal de la insulina en plasma, la tasa de absorción de carbohidratos, el efecto de la insulina y la glucosa en plasma.

Una vez finalizada la simulación se podrán consultar los resultados obtenidos en dos archivos que se crean en la carpeta en la que esté guardado el programa Interfaz. Estos archivos llevan el nombre del escenario y a continuación “_Tiempo” en el caso del archivo que contiene el vector de tiempos de la simulación y “_Resultados” en el caso del archivo que contiene la matriz de resultados obtenidos. Cada columna del segundo archivo representa las variables objeto de estudio (g_1 , g_2 , s_1 , s_2 , l , x_1 , x_2 , x_3 , Q_1 y Q_2) y cada fila el valor de las variables para el instante de tiempo definido por el vector guardado en el primer archivo. En caso de que se desee guardar las gráficas obtenidas consultar el punto 4.9. Guardar gráficas de resultados.

4.9. Guardar gráficas de resultados

Si se desea guardar la figura con las gráficas obtenidas al realizar la simulación se debe pulsar el botón guardar  de la barra de herramientas de la Figura. Al pulsar el botón se abrirá una ventana de Windows que permitirá seleccionar la dirección en la que se desea guardar el archivo y el nombre que se desea dar al mismo. Por defecto el archivo se guarda con el formato de figura de Matlab. Si se desea modificar el tipo de archivo se abre el desplegable de “Guardar como tipo” y se selecciona el formato deseado.

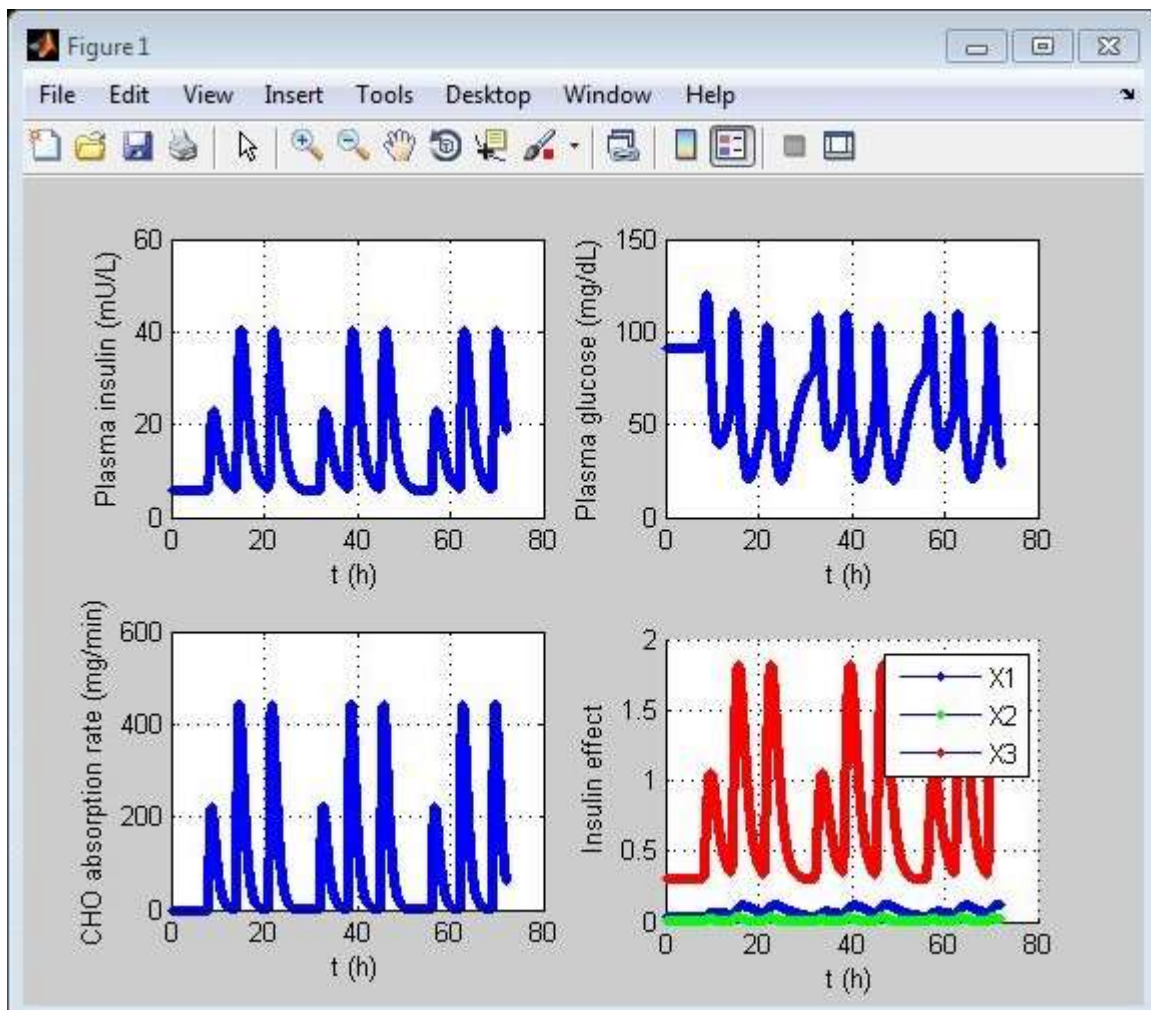


Figura 4.24. Guardar gráficas de resultados

Anexo II

Manual del programador

1. Introducción.....	103
1.1. Organización del manual	103
1.2. ¿A quién va dirigido?	103
2. Implementación del simulador.....	105
2.1. Función principal.....	105
2.2. Función Modelo Paciente	116
2.3. Obtención de los valores iniciales.....	117
3. Desarrollo de la interfaz gráfica	127
3.1. Diseño de la interfaz	128
3.2. Programación de la interfaz.....	134
3.2.1. Editores de texto y listas	134
3.2.2. Botones de añadir datos	134
3.2.3. Botón de guardar escenario.....	135
3.2.4. Botones de eliminar, editar y abrir	135
3.2.5. Selector del tipo de simulación y botón simular.....	136

4. Desarrollo de la comunicación	137
4.1. Programa Interfaz	137
4.2. Programa Simulador	137
4.3. Comunicación.....	138

Capítulo 1

Introducción

El manual del programador es un documento con carácter formativo que trata de informar al programador de la estructura interna del software. El manual intenta dar cobertura a todas las posibles dudas que surjan al desarrollar o modificar el software.

1.1. Organización del manual

En el manual se distinguen principalmente tres apartados en los cuales se intenta explicar el trabajo realizado durante el proceso de desarrollo de la aplicación. En primer lugar se explica el proceso seguido para obtener el código para realizar la simulación del paciente con diabetes tipo 1. A continuación se detalla el proceso seguido para la obtención de la interfaz para facilitar la introducción de datos. En último lugar, se explica como se ha realizado la comunicación de ambos ordenadores para obtener finalmente el simulador deseado.

1.2. ¿A quién va dirigido?

El presente manual va dirigido para todo aquel que desee desarrollar una aplicación similar a la implementada o que desee modificar la existente para mejorar algunas de sus características.

Para ser capaz de comprender sin problema alguno todo el código y ser capaz de realizar modificaciones en el mismo es necesario tener conocimientos de programación y, preferiblemente, haber trabajado previamente con Matlab para estar familiarizado con el lenguaje de programación.

Capítulo 2

Implementación del simulador

Para obtener el paciente virtual en primer lugar se implementa el código del modelo matemático que describe el comportamiento del metabolismo del paciente con diabetes tipo 1. Una vez se ha definido el modelo matemático y cómo se va a resolver, se implementa el código que permite la resolución. A partir de ahí se modifica el programa para facilitar la introducción de datos y realizar la comunicación entre ordenadores.

El programa utilizado para implementar el paciente virtual es Matlab, que ofrece múltiples posibilidades para la resolución del modelo de ecuaciones diferenciales ordinarias que se plantea.

Una vez seleccionado el modelo matemático con el que se va a realizar la simulación se debe crear una función que describa al paciente. Esta función, llamada Modelo Paciente, es la que contiene las ecuaciones del modelo de Cambridge, que es el que se ha seleccionado en este caso. Por otra parte, se crea una función que constituye el código principal, esto es, la función en la que se indican todos los parámetros del modelo y los datos de entrada, donde se realiza la simulación y donde se visualizan posteriormente los resultados.

2.1. Función principal

Inicialmente se va a diseñar el programa para que simule el comportamiento del metabolismo de un paciente con diabetes tipo 1 durante tres días haciendo tres comidas diarias: desayuno, comida y cena. Para ello, en primer lugar se debe crear una función en la que se implementará el código para realizar la simulación. Cuando se crea una función se debe tener en cuenta que se puede crear de dos formas distintas.

- Función sin argumentos de entrada y sin argumentos de salida. Ejecuta el código y se cierra al finalizar la ejecución.
- Función con argumentos de entrada y salida. Es una función que se utiliza dentro de otra función para obtener unos determinados datos. Se deben pasar los argumentos que precise en el orden adecuado y, tras su ejecución, devolverá un valor, un vector o una matriz con el resultado de la ejecución.

En este caso, puesto que se está implementando el código principal se creará una función con el primer formato, es decir, no se introducen argumentos de entrada y no devuelve ningún dato.

El primer paso para crear el código es definir cada parámetro del modelo de Cambridge. Para ello, se escriben todos los parámetros con su valor correspondiente, teniendo en cuenta que se deben introducir con las unidades adecuadas para su posterior utilización.

Una vez definidos los parámetros, se definen los datos de entrada, es decir, los datos que se modificarán para obtener las distintas respuestas en la simulación. Estos datos se introducen en tres matrices:

- Matriz de carbohidratos ingeridos (cho): se trata de una matriz en la que se indica en una columna el momento en el que se realiza la comida y, en la otra, los gramos de carbohidratos ingeridos. Hay que tener en cuenta que la simulación se hace en minutos, por tanto, hay que multiplicar cada hora de ingesta por 60 y, en los casos en los que se haya cambiado de día sumar los minutos de todos los días anteriores.
- Matriz de bolos de insulina (bolus): en esta matriz se debe indicar, en la primera columna, cuando se realiza la infusión de insulina, que será en el mismo momento en que se realice la ingesta, por tanto, se completa igual que la matriz anterior. En la segunda columna se debe indicar la dosis de insulina administrada. Se añade una tercera columna en la que se indica la duración del bolo de insulina, en este caso se va a suponer que todas son instantáneas, por tanto, se trata de una columna de unos.
- Matriz de insulina basal (basal): en este caso se introduce en la primera columna los instantes en los que se modifica la insulina basal (en minutos) y, en la segunda columna el valor de la insulina basal en ese instante $U \text{ min}^{-1}$. En caso de que sea constante sólo es necesario poner el instante inicial, es decir, cero y el valor de la insulina basal que corresponda.

El siguiente paso es definir el vector de valores iniciales. Para ello se define un vector y se le asigna un valor inicial a cada variable del modelo de Cambridge. Los valores iniciales son aquellos que se obtienen en caso de que no se realicen ingestas, como en este caso no se conocen se ponen todos a cero a excepción de la variable Q_1 .

El motivo de que esta variable no pueda ser nula inicialmente es que aparece en el numerador de la ecuación que permite obtener la concentración de glucosa, es decir, si Q_1 es nula, la concentración de glucosa también será nula. Esta concentración aparece en el denominador de la ecuación que representa la masa de glucosa del compartimento accesible del sistema de ecuaciones diferenciales, por lo tanto, no puede ser nulo puesto que no se podría obtener la solución. Se da un valor inicial de 90mg/dl a la concentración de glucosa, que se multiplicará por v_g y se dividirá por 18 para obtener Q_1 en mmol Kg^{-1} .

El siguiente paso es resolver el sistema de ecuaciones diferenciales ordinarias que se define en la función Modelo Paciente, que se explica en detalle más adelante. Para resolver este tipo de problemas Matlab dispone de diferentes funciones, que se muestran en la tabla 2.1.

Función	Método
ode45	Runge-Kutta
ode23	Runge-Kutta
ode113	Adams
ode15s	NDFs (BDFs)
ode23s	Rosenbrock
ode23t	Trapezoidal rule
ode23tb	TR-BDF2
ode15i	BDFs

Tabla 2.1. Funciones de resolución de ecuaciones diferenciales ordinarias en Matlab

En este caso se realiza la resolución del problema con dos de estas funciones (ode45 y ode15s), en ambos casos los argumentos de entrada y salida tienen el mismo formato y se explican a continuación.

Argumentos de entrada:

- odefun. Una función que evalúa el lado derecho del sistema de ecuaciones diferenciales, resuelve sistemas de ecuaciones de la forma $x' = f(t,x)$ o problemas que incluyen matrices $M(t,x) x' = f(t,x)$.

En este caso la función que contiene el sistema de ecuaciones diferenciales es, como se ha indicado anteriormente, Modelo Paciente.

- tspan. Un vector que especifica el intervalo de integración, $[t_0, t_f]$. Inicialmente se toma como condiciones iniciales la primera posición del vector, e integra desde esa posición hasta la última. Para obtener la solución en intervalos específicos se define de la forma tspan = $[t_0, t_1, \dots, t_f]$.

Inicialmente se va a llamar a la función cinco veces por cada día para hacer que coincida con las comidas, para ello se realiza un bucle que vaya cambiando de día y en cada iteración del mismo se hacen cinco llamadas cada una con un intervalo de simulación, de forma que se cubran las ingestas de un día. Los intervalos de simulación son los siguientes:

$[0.00 - 8.00]$, $[8.00 - 12.00]$, $[12.00 - 14.00]$, $[14.00 - 21.00]$, $[21.00, 24.00]$

- x0. Un vector con las condiciones iniciales del problema.

En la primera llamada a la función este vector es el que se ha definido previamente, posteriormente, se modifica cada vez que finaliza una llamada, tomando el último valor de la solución obtenida para cada variable.

- Options. Parámetros opcionales que permiten cambiar las propiedades de integración que se utilizan por defecto.

En este caso se define una tolerancia de 10^{-6} , esta tolerancia es una medida del error relativo al tamaño de cada componente de la solución.

Argumentos de salida:

- t. Vector columna de tiempos.

Tras cada llamada a la función se guarda un vector t_i . Al finalizar las cinco llamadas de un día se añaden al vector t, que se declara inicialmente vacío. Por tanto, tras cada llamada a la función el vector t aumenta de tamaño.

- X. Matriz de soluciones, cada columna se corresponde con una variable del sistema y cada fila con el tiempo definido en el vector t.

En este caso ocurre lo mismo que con el vector t de tiempo, tras cada llamada se guarda una matriz X_i con la solución al problema en el intervalo definido. Al finalizar las cinco llamadas de un día se añaden a la matriz X, inicialmente vacía.

Una vez definida la función para la resolución del sistema de ecuaciones diferenciales ya está creado el simulador, pues ya se obtienen los resultados que se desean obtener. Para poder interpretar estos resultados se incluye el código para realizar diferentes gráficas, que se explican a continuación.

- Gráfica 1. En primer lugar se muestra la evolución temporal de la insulina en plasma (mU min^{-1}). Para ello se debe realizar una gráfica con el vector de tiempos en el eje X y en el eje Y la columna 5 de la matriz X (I), pues es la que se corresponde con la variable I. Para que en la gráfica se pueda ver el tiempo en horas se divide el vector de tiempos entre 60.
- Gráfica 2. Muestra la evolución temporal de la tasa de absorción de carbohidratos (mg min^{-1}). Igual que en el caso anterior se toma en el eje X el vector de tiempos (en horas), en el eje Y se toma la columna 2 de la matriz X (g_2).
- Gráfica 3. Muestra la evolución temporal del efecto de la insulina. En este caso se busca representar tres variables en una sola gráfica x_1 , x_2 y x_3 que se corresponden con las columnas 6, 7 y 8 de la matriz X. Para facilitar la interpretación de la gráfica cada una de las variables se muestra en un color: x_1 en azul, x_2 en verde y x_3 en rojo.
- Gráfica 4. Muestra la evolución temporal de la glucosa en plasma (mg dL^{-1}). Igual que en los casos anteriores en el eje X se muestra el vector de tiempo y en el eje Y la columna 9 de la matriz X (Q_1). Se realiza un cambio de unidades para obtener los datos en mg dL^{-1} en lugar de en mmol kg^{-1} .
- Gráfica 5. Muestra dos gráficas, en una la evolución temporal de la insulina en plasma y, en la otra, la evolución temporal de la glucosa en plasma. Permite visualizar ambas variables a la vez para ver como evolucionan una respecto a la otra.

A continuación se muestran las gráficas con los resultados obtenidos al realizar la simulación con la función ode45 y ode15s.

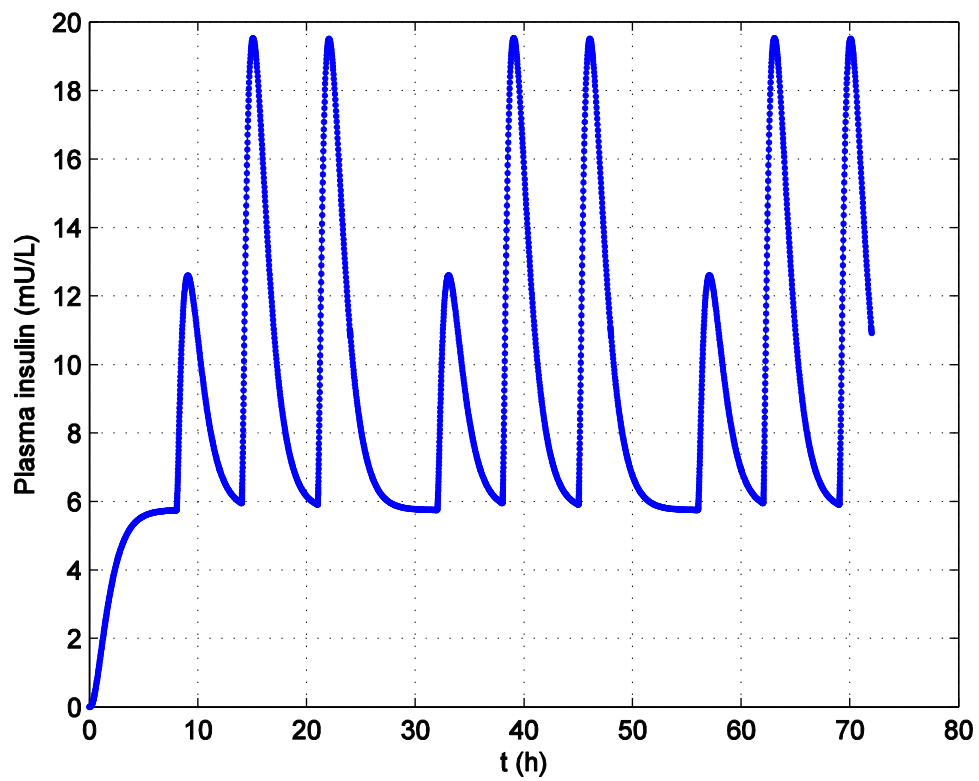


Figura 2.1. Evolución temporal de la insulina en plasma (ode45)

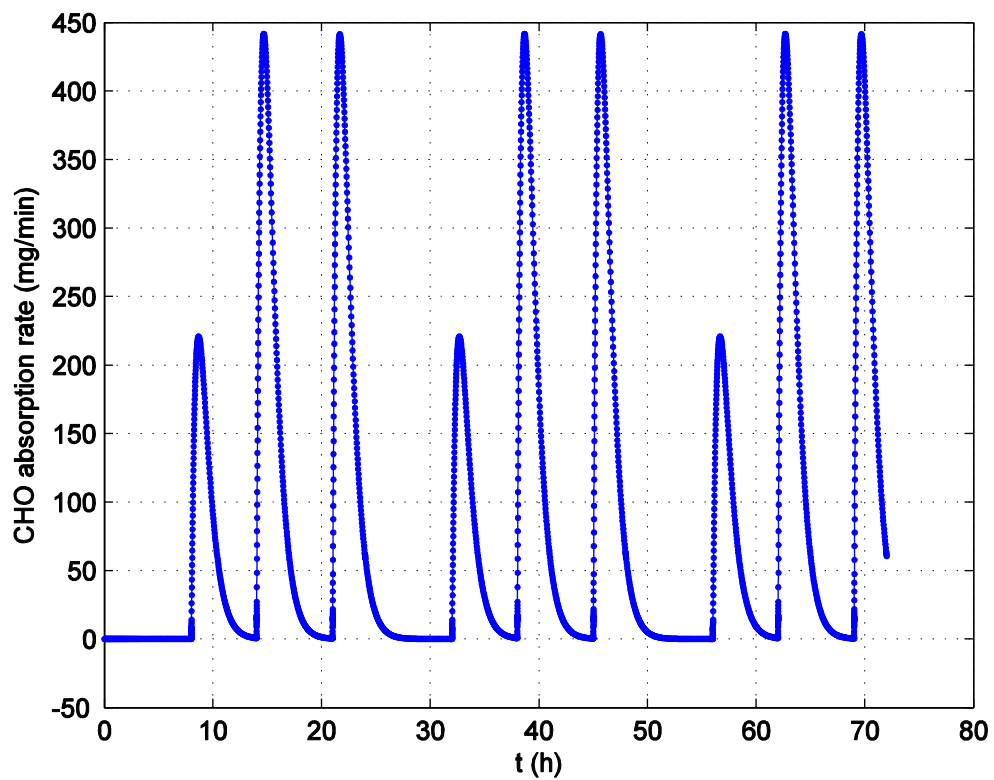


Figura 2.2. Tasa de absorción de carbohidratos (ode45)

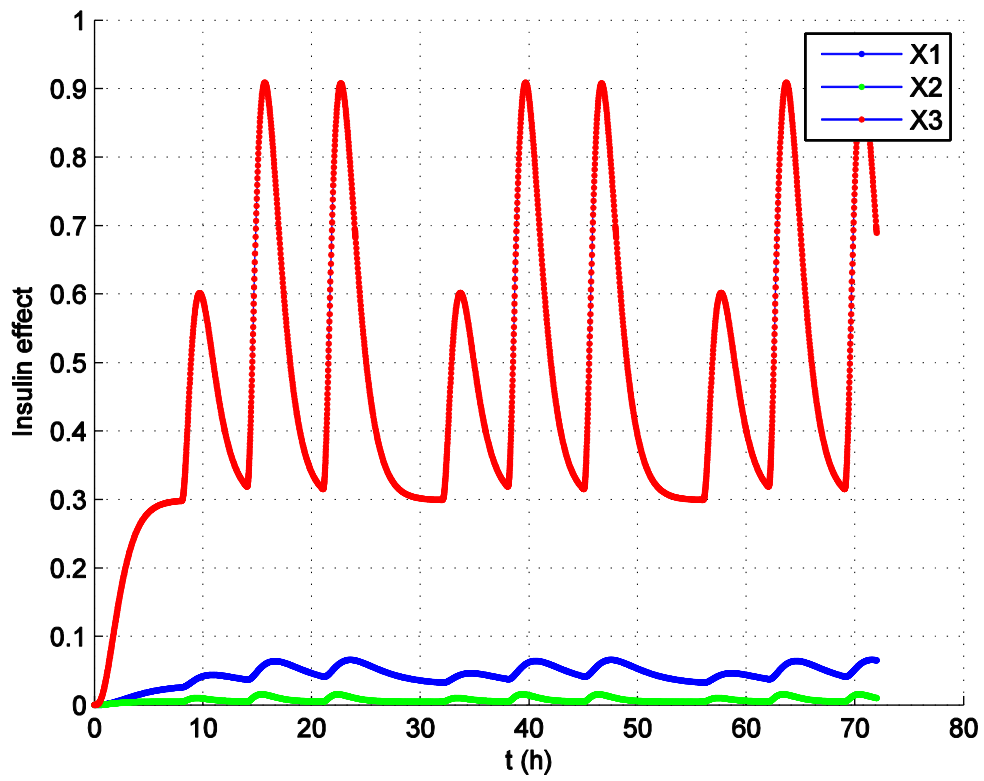


Figura 2.3. Evolución temporal del efecto de la insulina (ode45)

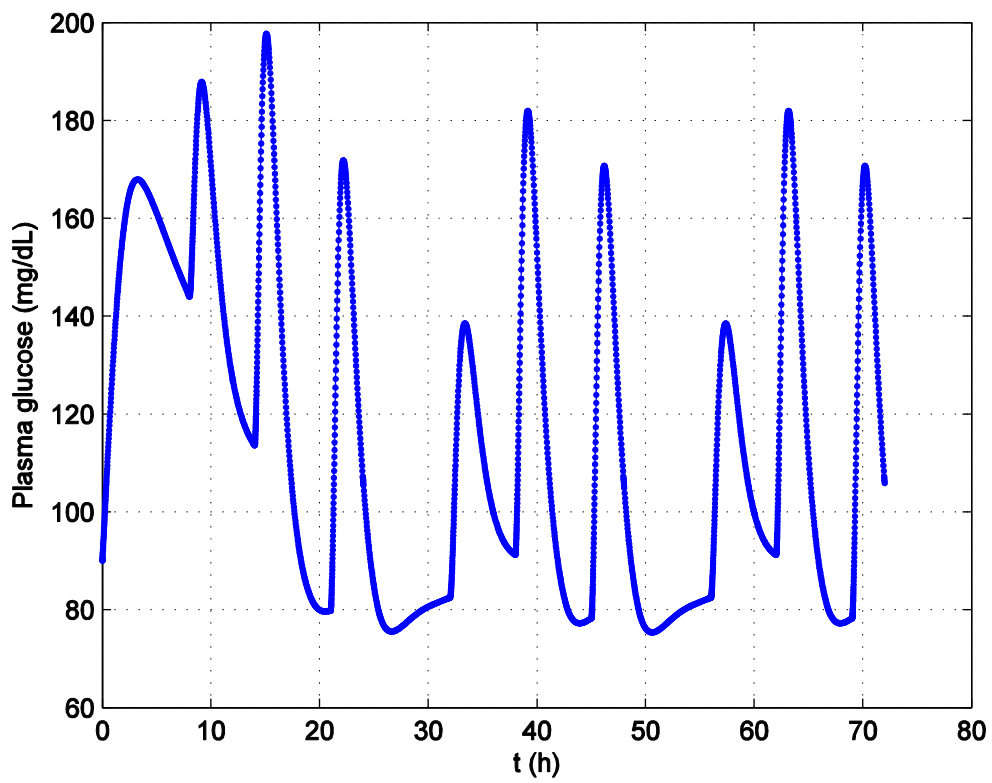


Figura 2.4. Evolución temporal de la glucosa en plasma (ode45)

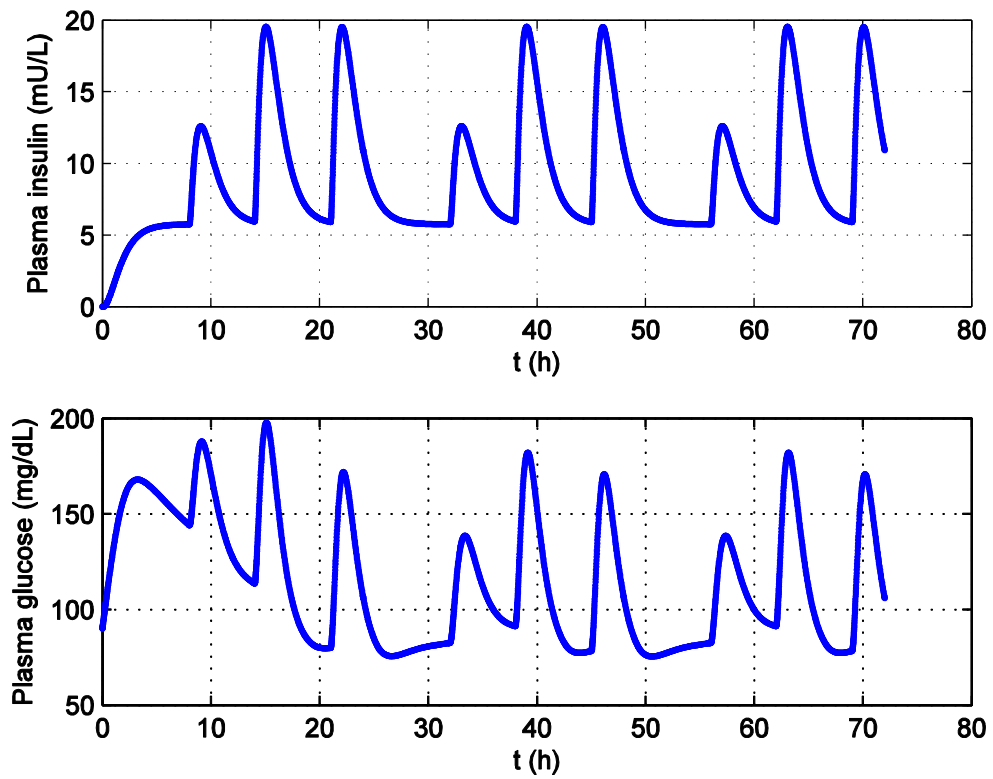


Figura 2.5. Comparación de la evolución temporal de insulina y glucosa en plasma (ode45)

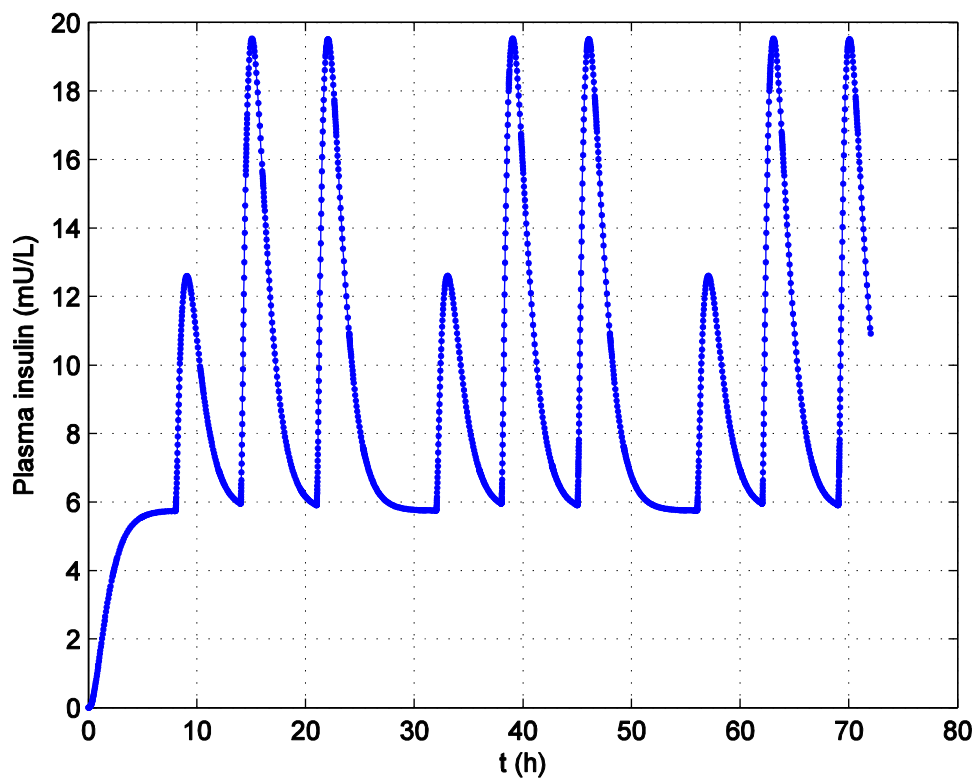


Figura 2.6. Evolución temporal de la insulina en plasma (ode15s)

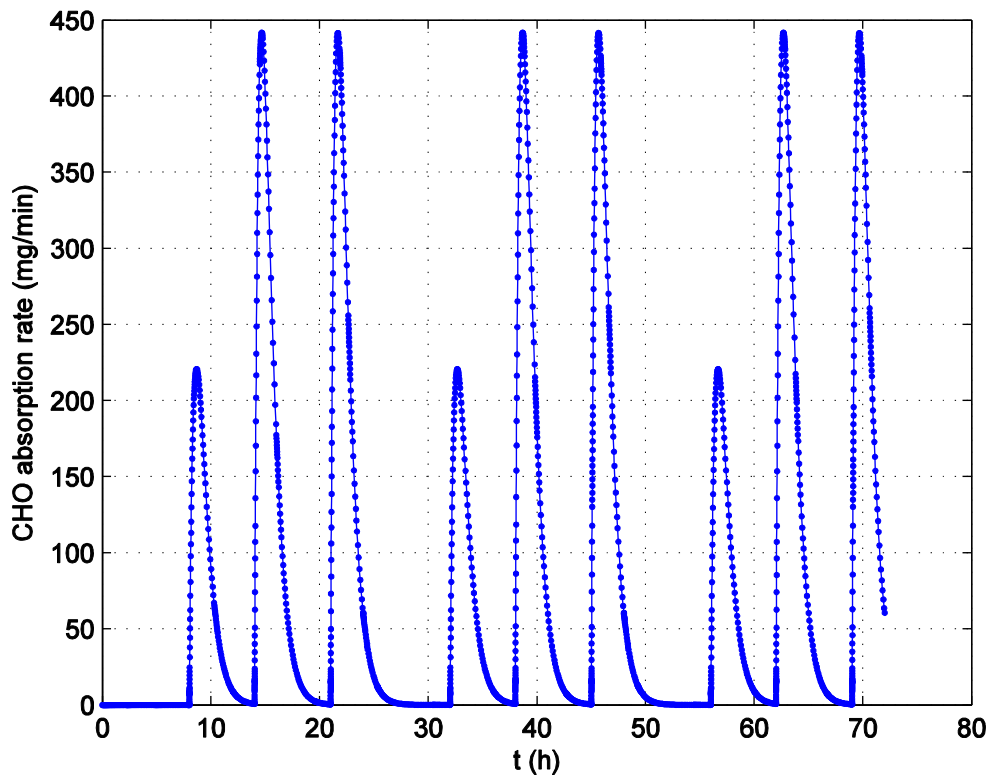


Figura 2.7. Tasa de absorción de carbohidratos (ode15s)

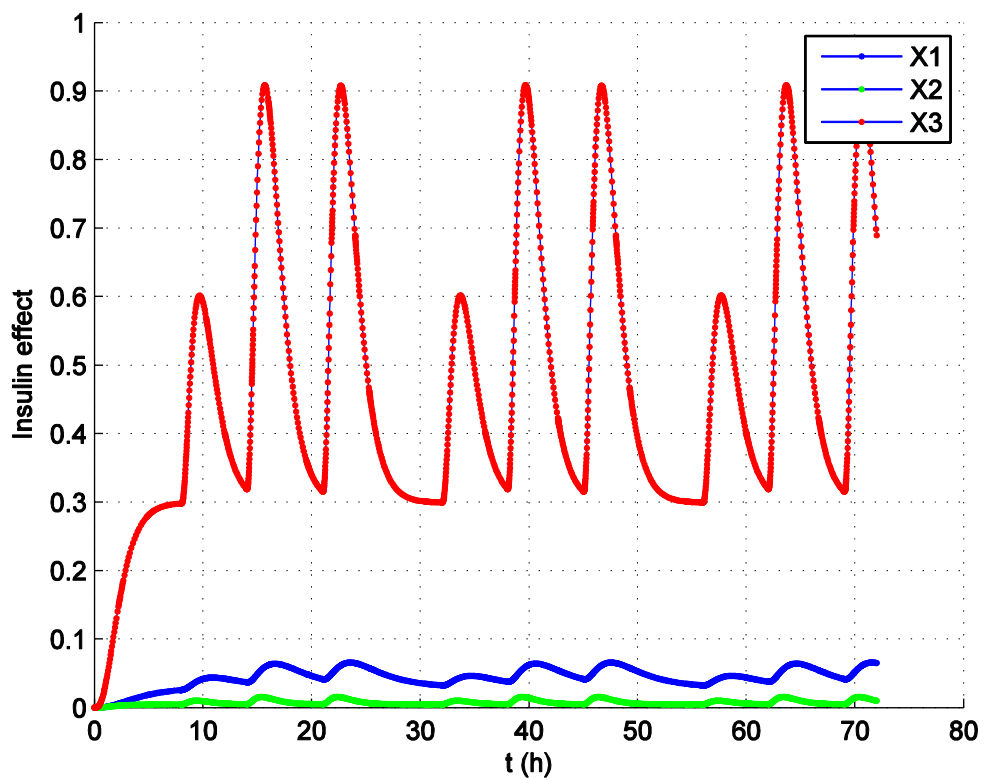


Figura 2.8. Evolución temporal del efecto de la insulina (ode15s)

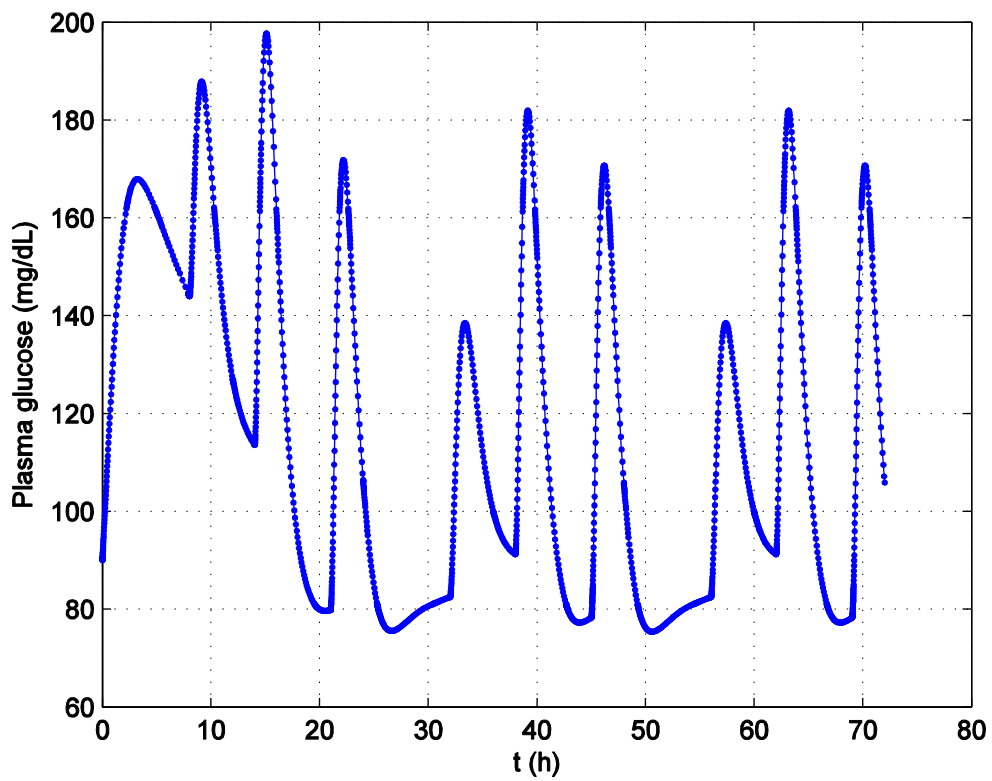


Figura 2.9. Evolución temporal de la glucosa en plasma (ode15s)

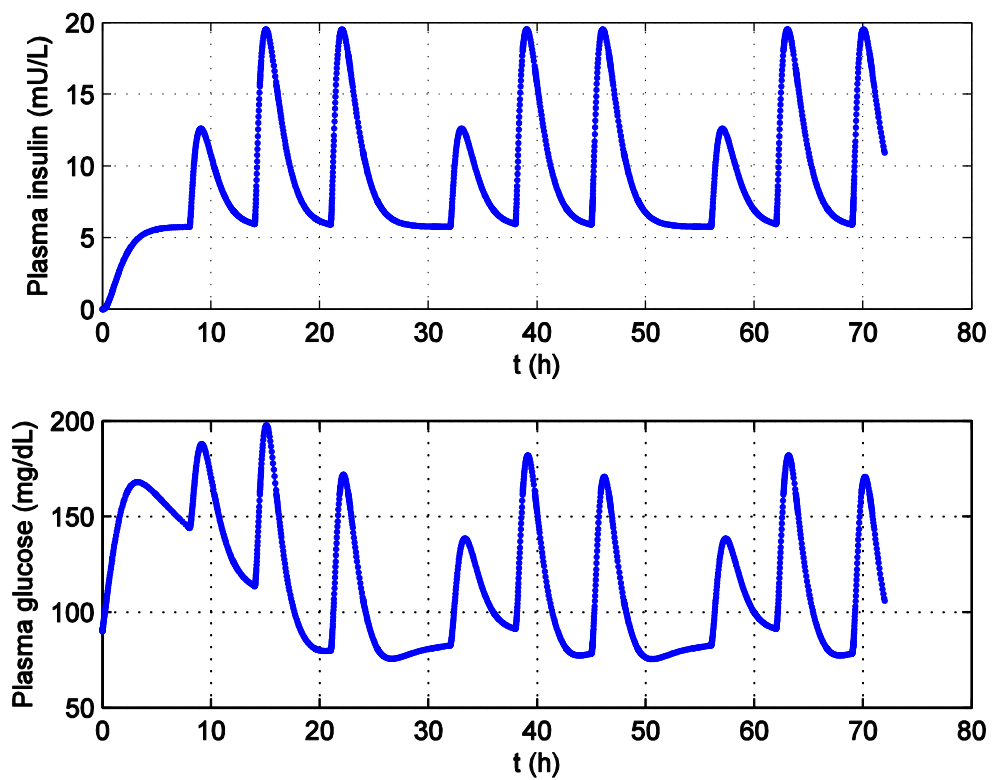


Figura 2.10. Comparación de la evolución de insulina y glucosa en plasma (ode15s)

Como se puede observar no hay ninguna diferencia entre usar la función ode45 u ode15s, por lo que de aquí en adelante sólo se utilizará ode45.

Puesto que se busca que la simulación se pueda realizar de dos modos distintos en bloque o en tiempo real, se modifica la llamada a la función ode45 para que se haga de un modo u otro en función del tipo de simulación.

En caso de que se desee realizar la simulación en bloque sólo se necesita obtener los resultados una vez se han simulado todos los días. En lugar de llamarla cinco veces cada día, adaptándola a las horas a las que se realizan las ingestas, se hace una llamada y se toma un intervalo de integración que va desde el instante inicial hasta el 4320, que se corresponde con las doce de la noche del tercer día, cada cinco minutos. De este modo se obtienen los niveles de glucosa en plasma cada cinco minutos del mismo modo que ocurriría si se tuviera un paciente conectado a un monitor continuo de glucosa.

En caso de que se desee realizar la simulación en tiempo real se deben obtener datos de resultados cada cierto tiempo n definido por el usuario en la cabecera del código. Se da la opción de acelerar el proceso, para ello se añade un factor de aceleración que valdrá uno si no se acelera el proceso. Para obtener este tipo de simulación se realiza un bucle en el que se va a realizar la llamada a la función cada I minutos, siendo I el tiempo que transcurre desde que se realiza una llamada hasta la siguiente. Se toma un intervalo de integración desde el instante inicial 0 , si ya se ha realizado alguna llamada, desde el último instante simulado hasta I minutos después cada 5 minutos. En cada caso se modificará el vector de valores iniciales por el último valor de cada variable de la matriz de resultados obtenida anteriormente.

Para que se muestren los datos en tiempo real es necesario añadir unas pausas para que los datos se muestren cada n minutos. Puesto que se añade la posibilidad de acelerar el proceso cuando se realiza en tiempo real, el tiempo de las pausas se divide por el factor de aceleración.

Si se ejecuta la función con estas modificaciones se observa que se obtiene el mismo resultado que en los casos anteriores independientemente del tipo de simulación realizada.

2 2. Función Modelo Paciente

La función Modelo Paciente tiene unos argumentos de entrada, necesarios para que se ejecute, y devuelve una salida. Los argumentos de entrada de Modelo Paciente son los parámetros del modelo, el vector de tiempos (t), inicialmente vacío, el vector con los valores iniciales de las variables del modelo (x), la matriz de carbohidratos ingeridos (cho), la matriz de bolos de insulina administrados (bolus) y la matriz de insulina basal (basal).

Lo primero que hace la función cuando se ejecuta es guardar cada valor inicial que recibe del vector x en una nueva variable, cuyo nombre se corresponde con la variable que representa ese valor inicial. De este modo no se necesita llamar a una determinada posición del vector x al definir las ecuaciones del modelo facilitando la implementación del código y la posterior lectura del mismo.

En segundo lugar calcula el flujo de la ingesta (u_{cho}) y el flujo de la bomba (u_{ins}), a partir de las matrices de ingestas, de bolos de insulina e insulina basal que se han definido previamente en el código principal. Inicialmente los flujos se consideran nulos y sólo se modificará el valor cuando se realicen las comidas. Por tanto, la función necesita averiguar en qué minutos se realizan las comidas para poder calcular, cuando sea necesario, los flujos de ingesta y de la bomba.

Se dispone de tres bucles que recorren las matrices de datos de modo que sólo se realizan cambios en caso de que se hayan realizado comidas. Si se detecta que se ha realizado una comida se deben hacer los cambios de unidades pertinentes para que se pueda realizar el cálculo de los flujos de ingesta y de la bomba. Para ello se debe considerar cada matriz por separado:

- Matriz de carbohidratos: los datos introducidos en la segunda columna, es decir, los carbohidratos ingeridos, se encuentran en gramos, por lo que se precisa realizar el cambio a mmol. Para realizar el cambio se multiplican los carbohidratos ingeridos en ese instante por 1000 para convertirlo en mg y se divide entre 180 para obtener el valor en mmol. El resultado obtenido es el flujo de la ingesta que se almacena como u_{cho} y se encuentra en mmol min^{-1} , pues se considera que la ingesta dura un minuto.
- Matriz de bolos de insulina: por una parte se introduce, en la segunda columna, la dosis de insulina y por otra la duración de la infusión, en la tercera columna. Se necesita obtener el valor en U min^{-1} , para ello se divide la dosis de insulina administrada entre la duración de la infusión, obteniéndose así que flujo del bolo de insulina administrado que se almacena en la variable u_{ins_bolus} .

- Matriz de insulina basal: en este caso no es necesario realizar un cambio de unidades pues ya se encuentra en $U \text{ min}^{-1}$, únicamente se almacena el valor obtenido en la variable $u_{\text{ins_basal}}$.

Una vez obtenidas las variables $u_{\text{ins_bolus}}$ y $u_{\text{ins_basal}}$ para que se calcule el flujo de la bomba sólo se debe realizar la suma de ambas. Una vez obtenido, se multiplica el valor por 1000 para obtenerlo en $mU \text{ min}^{-1}$.

En último lugar se deben definir las ecuaciones del modelo matemático de Cambridge. En primer lugar se definen las ecuaciones de los términos que aparecen en el sistema de ecuaciones diferenciales y, por último el propio sistema. A la hora de definir las funciones se debe tener en cuenta que hay que adaptarse al lenguaje de programación de Matlab.

Las ecuaciones definidas a trozos se deben realizar con condicionales de modo que en caso de que se cumpla una condición la función adquiera un valor u otro. Al definir el sistema de ecuaciones diferenciales se añaden las variables que representan a las derivadas y, finalmente, se incluyen en un vector columna, de modo que posteriormente se pueda resolver el sistema almacenado.

2.3. Obtención de los valores iniciales

Una vez conseguida la simulación es necesario obtener el vector de valores iniciales de forma que la simulación comience de forma estable. Para ello se crea una nueva función muy similar a la función principal explicada anteriormente. Para crear esta nueva función se hace una copia de la principal y se hacen algunos cambios.

En esta función se busca ver en qué valor se estabiliza cada una de las variables del modelo cuando no se produce ninguna perturbación. Por este motivo, las matrices de datos de entrada, es decir, la de ingesta de carbohidratos, bolos de insulina e insulina basal se encuentran vacías.

El resto del código no es necesario modificarlo y al ejecutarlo se obtienen los resultados que se muestran a continuación.

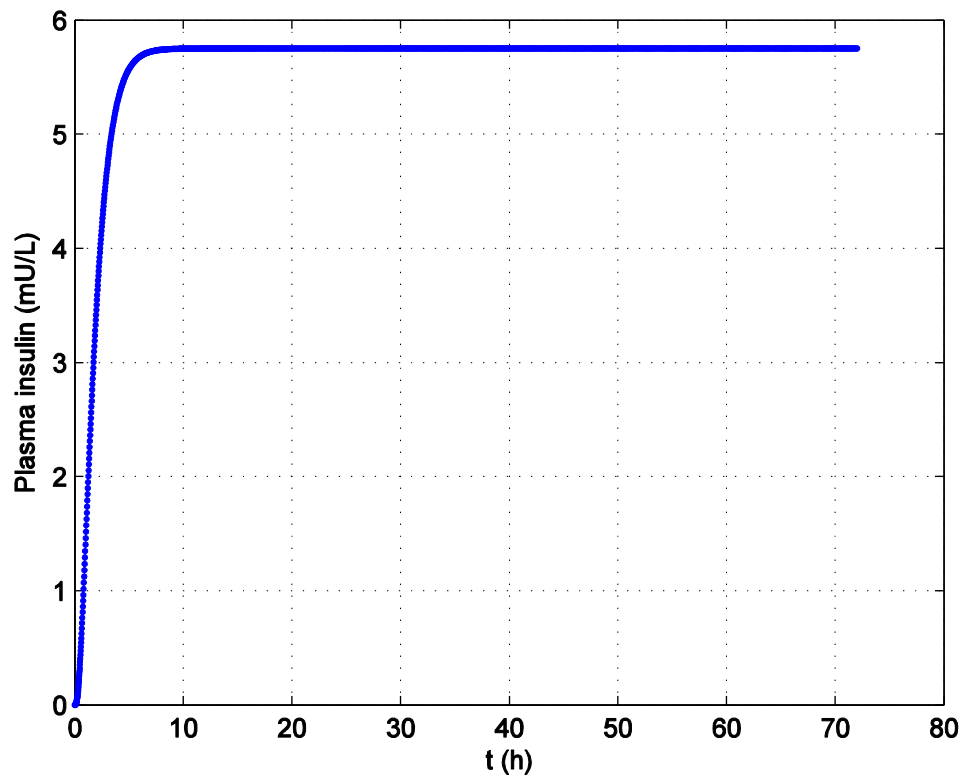


Figura 2.11. Evolución temporal de la insulina en plasma sin ingestas

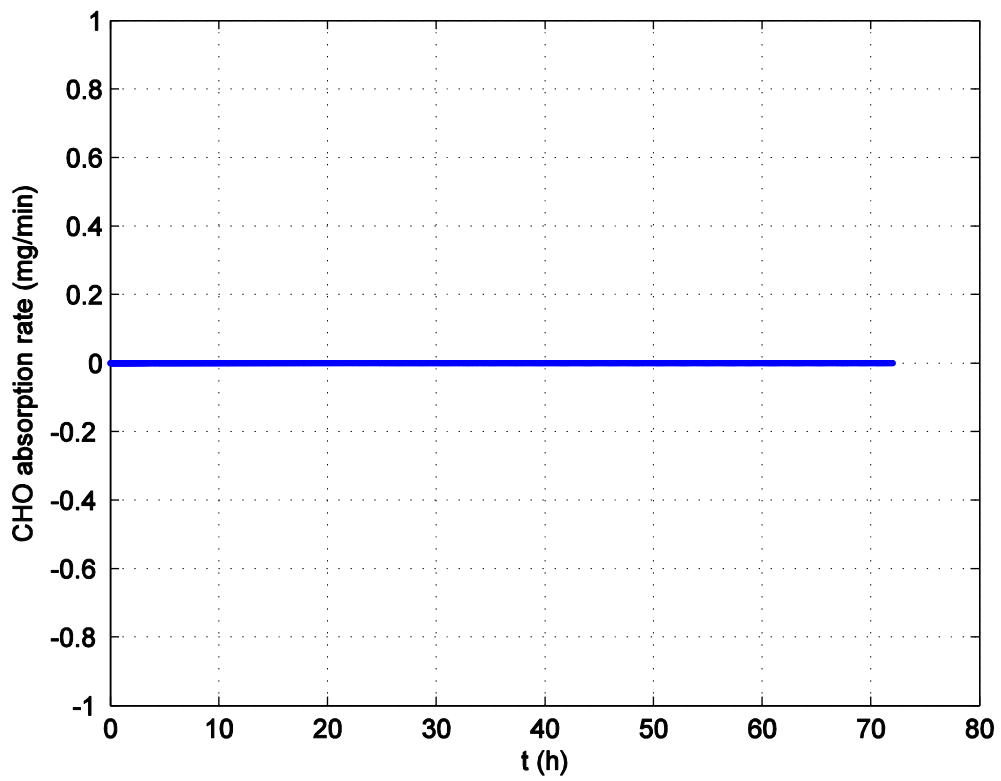


Figura 2.12. Tasa de absorción de carbohidratos sin ingestas

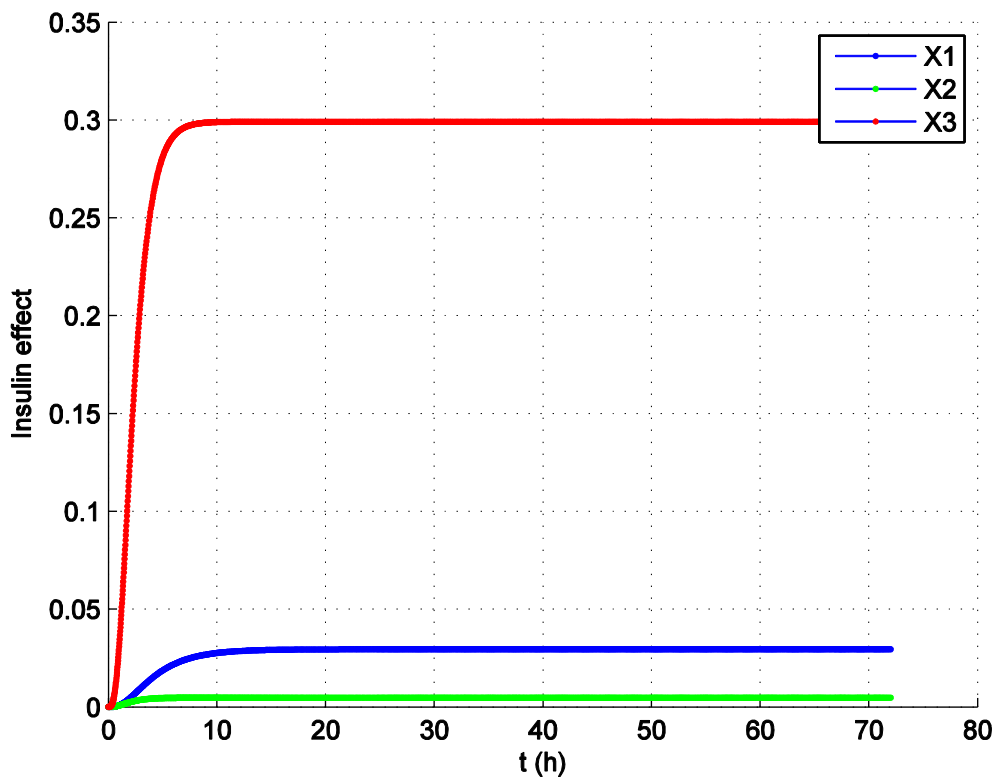


Figura 2.13. Evolución temporal del efecto de la insulina sin ingestas

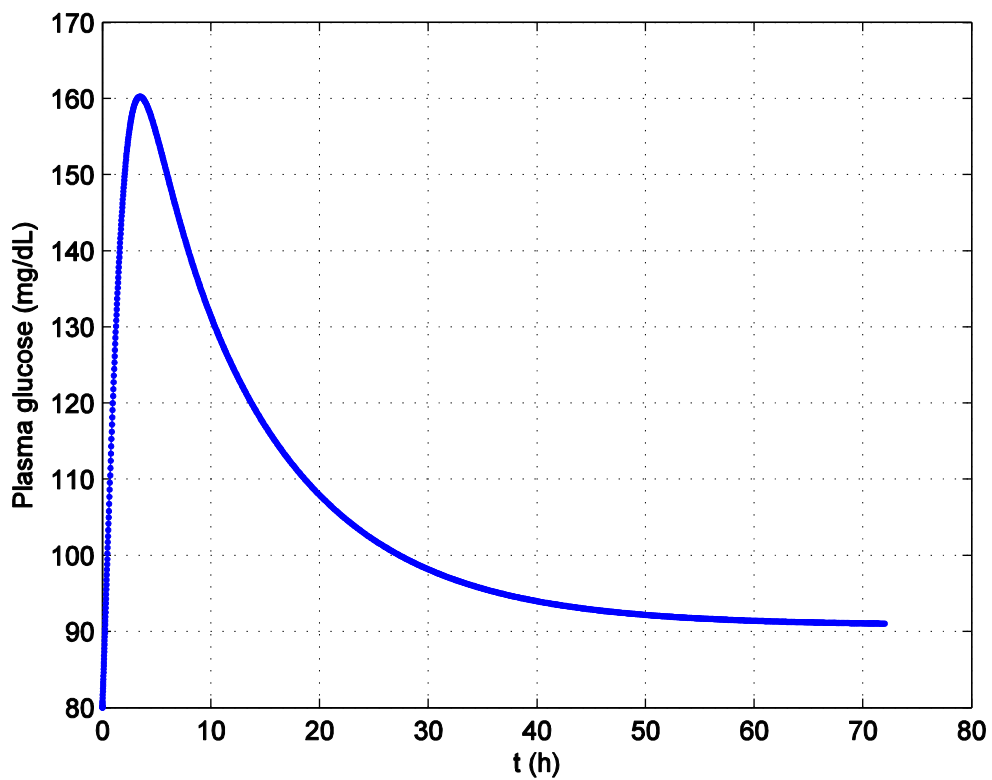


Figura 2.14. Evolución temporal de la glucosa en plasma sin ingestas

Si se realiza la simulación durante tres días se observa que la glucosa en plasma no se llega a estabilizar. Por este motivo, se aumenta el tiempo de simulación a cinco días, de este modo todas las variables se han estabilizado y no hay posibilidad de error al tomar los valores iniciales. Con la simulación de cinco días se obtienen los resultados que se muestran a continuación.

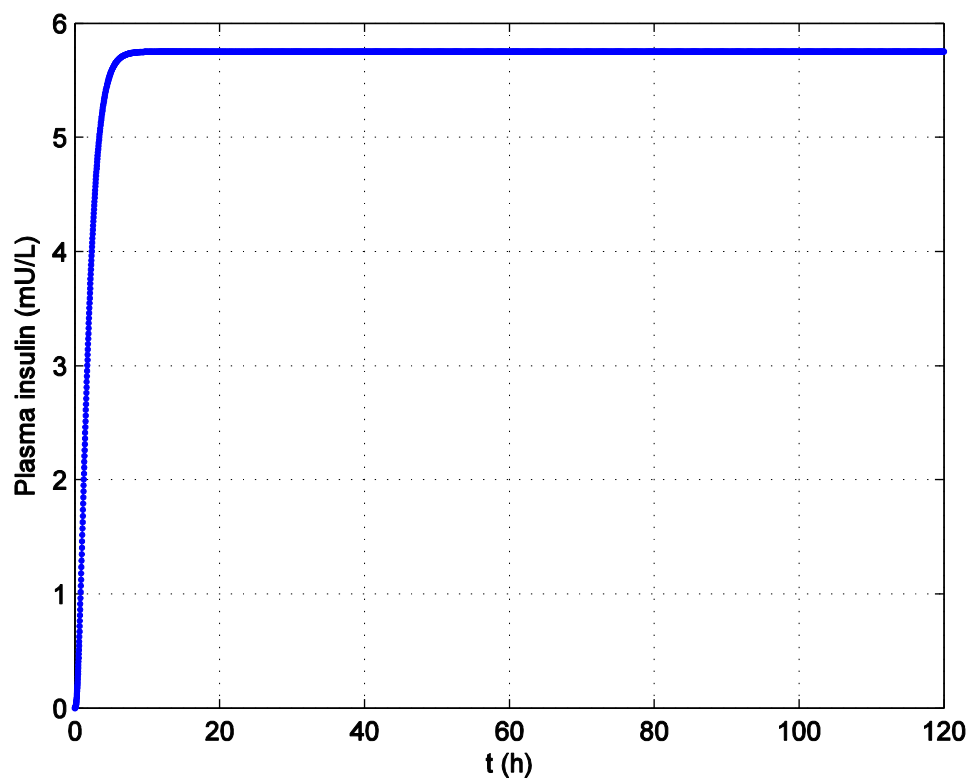


Figura 2.15. Evolución temporal de la insulina en plasma sin ingestas (5 días)

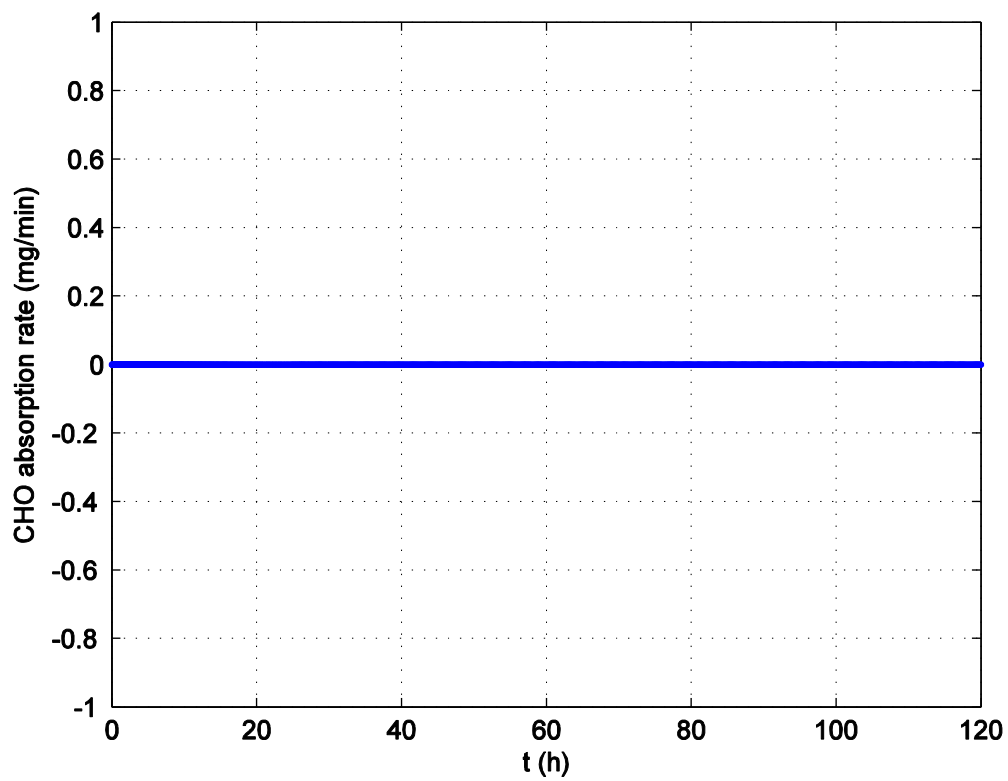


Figura 2.16. Tasa de absorción de carbohidratos sin ingestas (5 días)

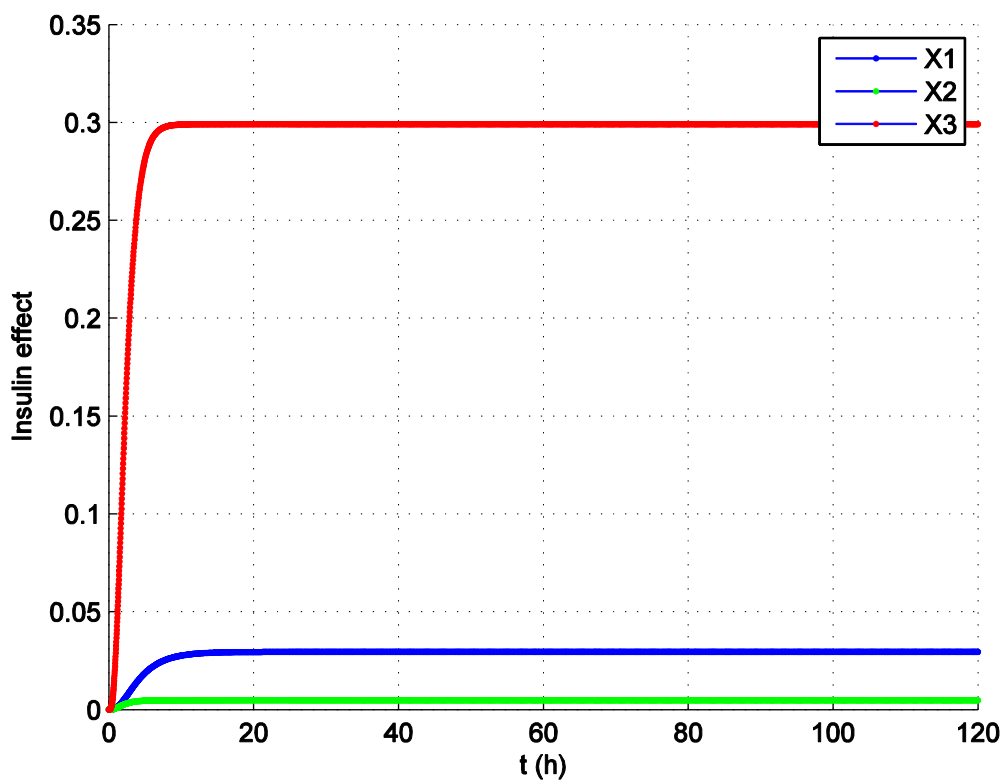


Figura 2.17. Evolución temporal del efecto de la insulina sin ingestas (5 días)

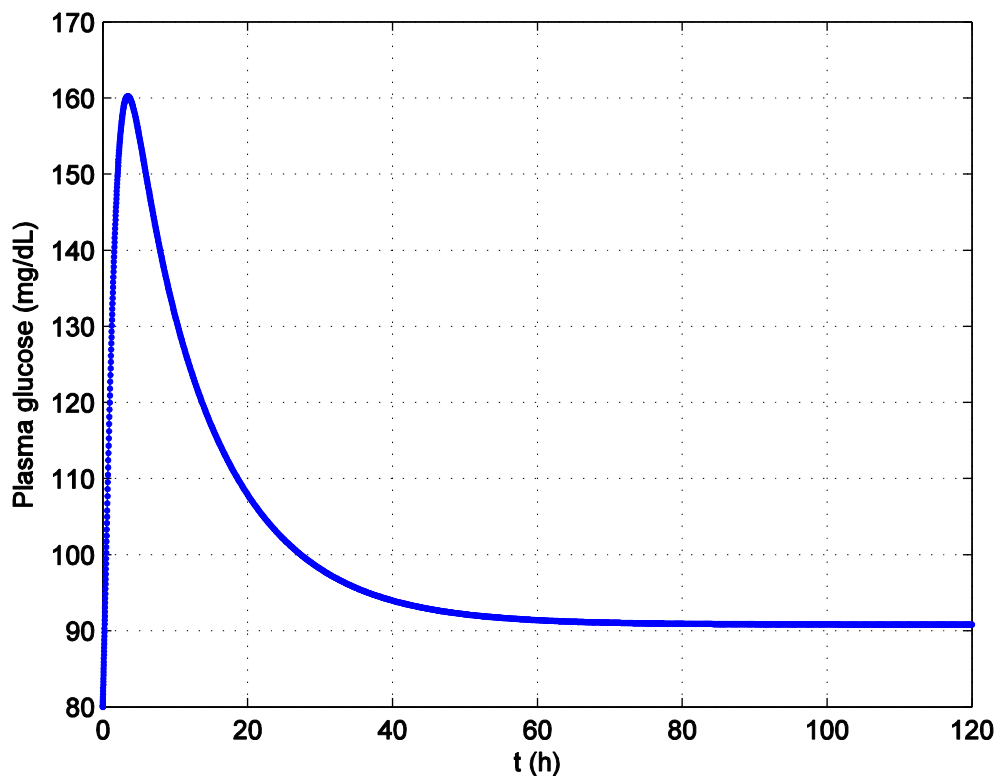


Figura 2.18. Evolución temporal de la glucosa en plasma sin ingestas (5 días)

Una vez obtenidos los valores en los que se estabiliza cada variable se modifica la función para que tenga unos argumentos de entrada y devuelva una salida. Los argumentos de entrada son los parámetros del modelo de Cambridge definidos en el código principal. La salida es el vector de valores iniciales que se utilizará posteriormente en el código principal cuando se haga la primera llamada a la función ode45. Este vector se obtiene tomando el último valor de cada columna de la matriz X que se obtiene como resultado de la simulación.

Una vez hecho esto, se modifica el código principal. En lugar de declarar un vector de valores iniciales x_0 , éste se obtiene automáticamente al hacer una llamada a la función Valores Iniciales que se acaba de crear.

De este modo se ha conseguido obtener una simulación paso a paso del comportamiento de un paciente durante tres días realizando tres comidas diarias. Los resultados que se obtienen una vez modificado el vector de valores iniciales se muestran a continuación.

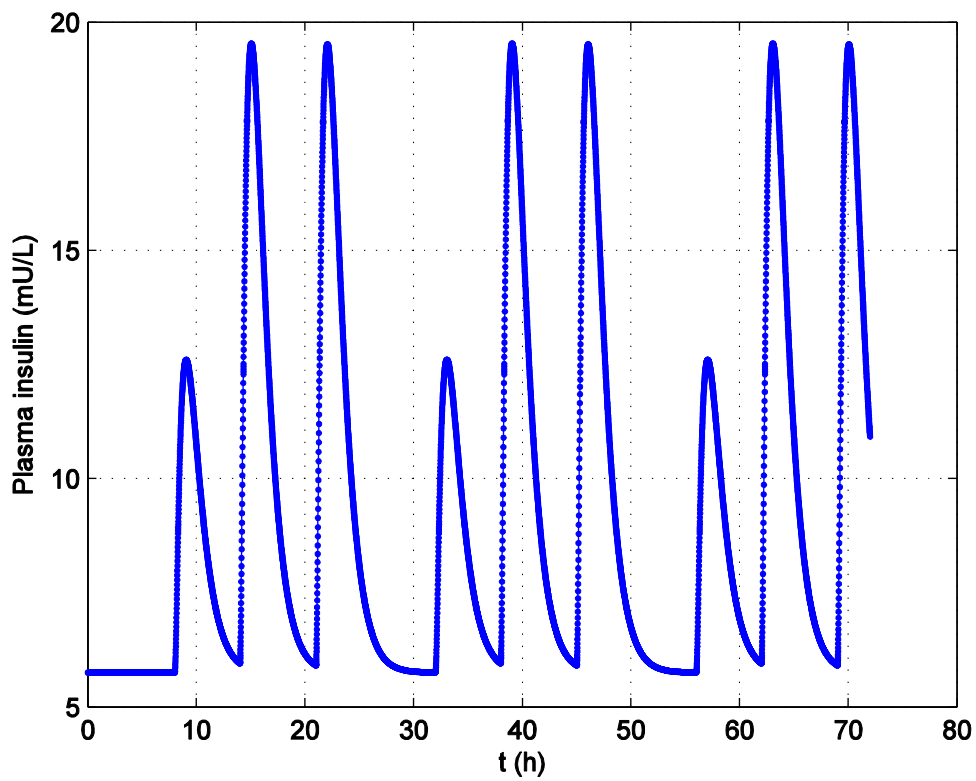


Figura 2.19. Evolución temporal de la insulina en plasma

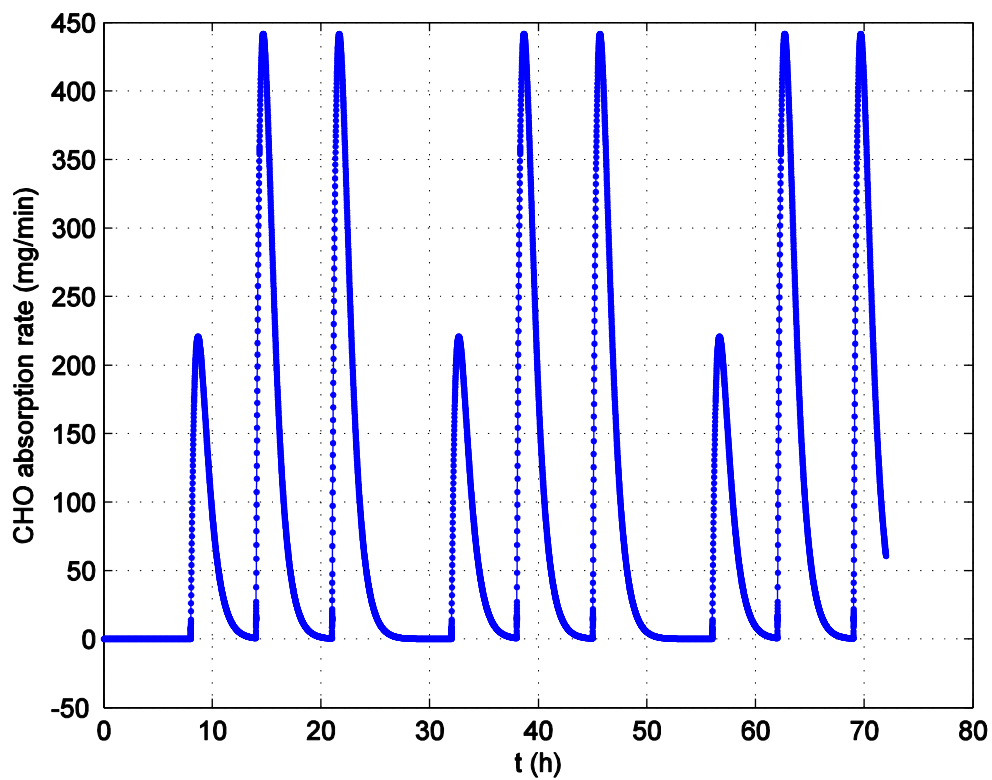


Figura 2.20. Tasa de absorción de carbohidratos

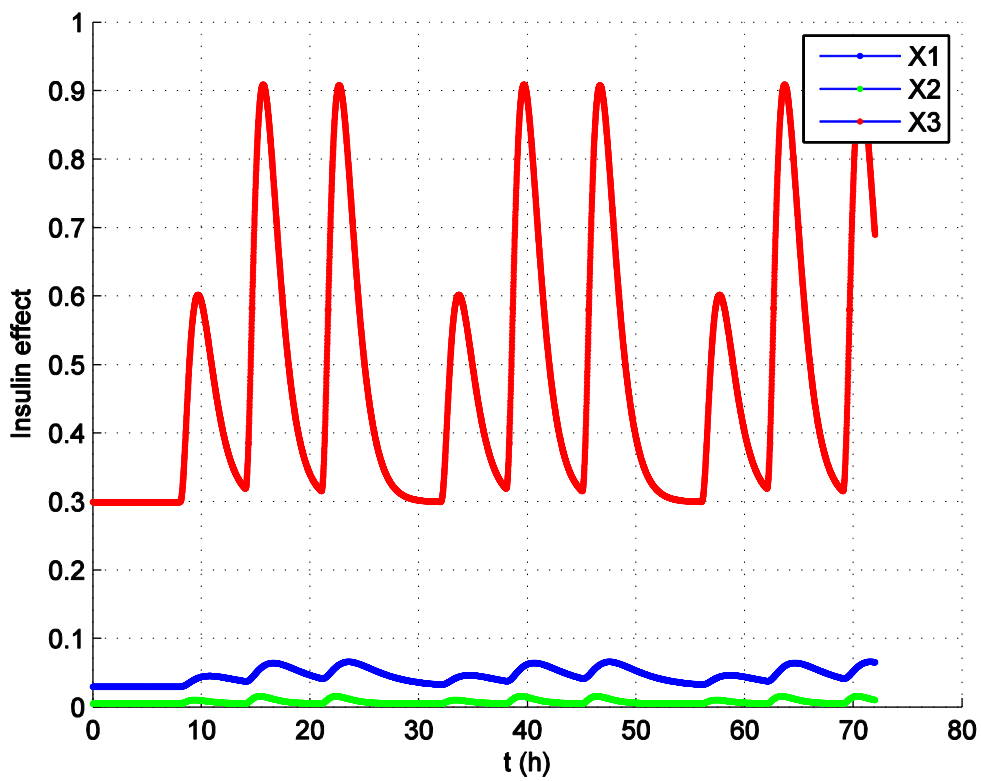


Figura 2.21. Evolución temporal del efecto de la insulina

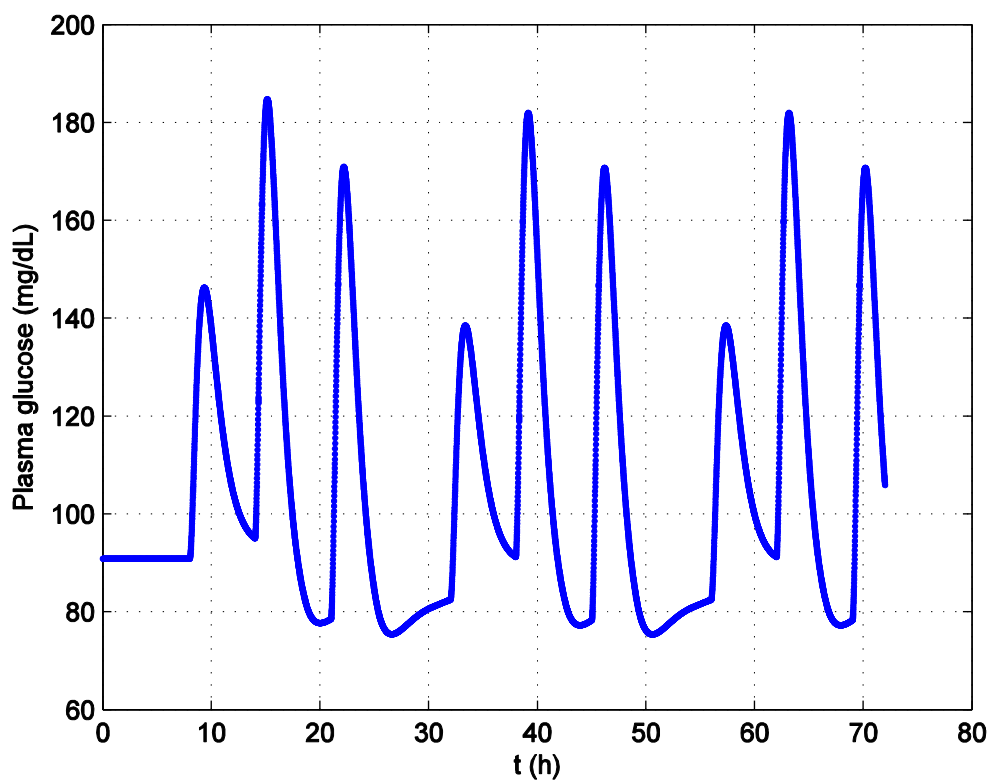


Figura 2.22. Evolución temporal de la glucosa en plasma

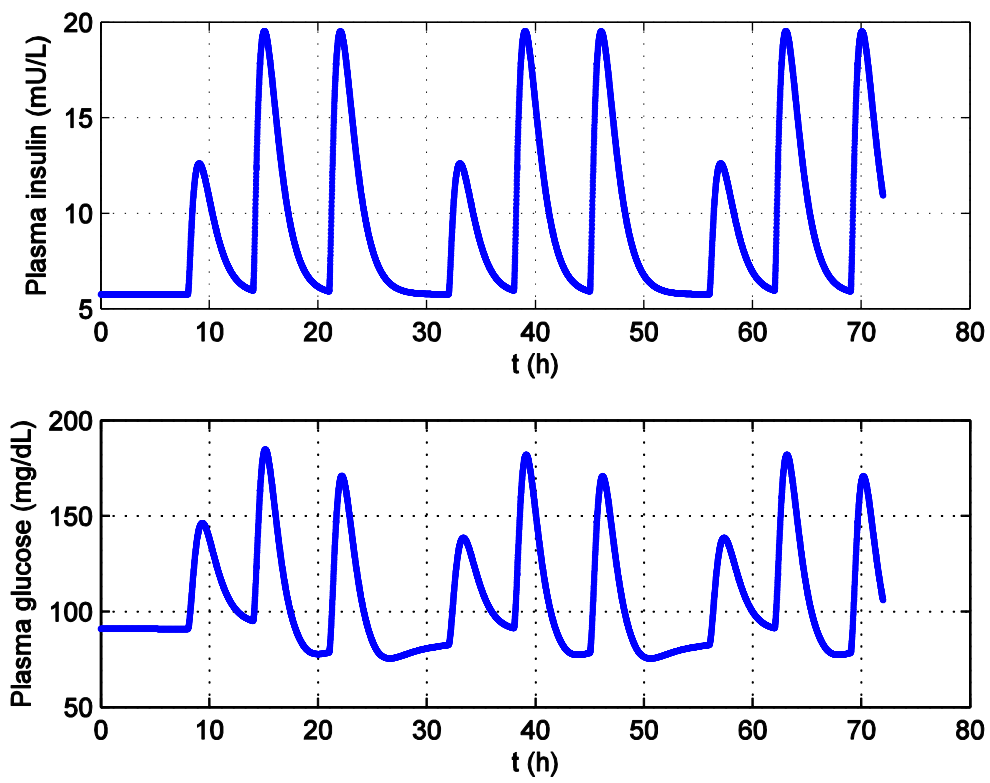


Figura 2.23. Comparación de la evolución temporal de la insulina y glucosa en plasma

Capítulo 3

Desarrollo de la interfaz gráfica

Creando una interfaz gráfica para introducir los datos de las matrices de carbohidratos, bolos e insulina basal se consigue que éstos sean más visuales y, además, ya no se necesita tener nociones de programación para poder realizar la simulación, ya que no es necesario escribir la matriz en el formato adecuado para Matlab.

Para la realización de la interfaz se utiliza Guide, se encuentra disponible en Matlab y se trata de un entorno de programación visual que permite realizar programas que necesitan un ingreso continuo de datos. De este modo, al ejecutar el programa se abrirá una ventana que permitirá introducir los datos del modo que se haya elegido previamente, sin necesidad de acceder al código.

Cuando se crea un nuevo GUI se abre una ventana como la que se muestra en la Figura 3.1 donde se muestra el entorno de diseño, se pueden observar las herramientas para incluir los diferentes componentes que ofrece la aplicación.

Cuando se inserta cualquier elemento se crea automáticamente el código básico de programación del componente. Para cada componente se deberá añadir el código que se desee ejecutar cuando el elemento se utilice.

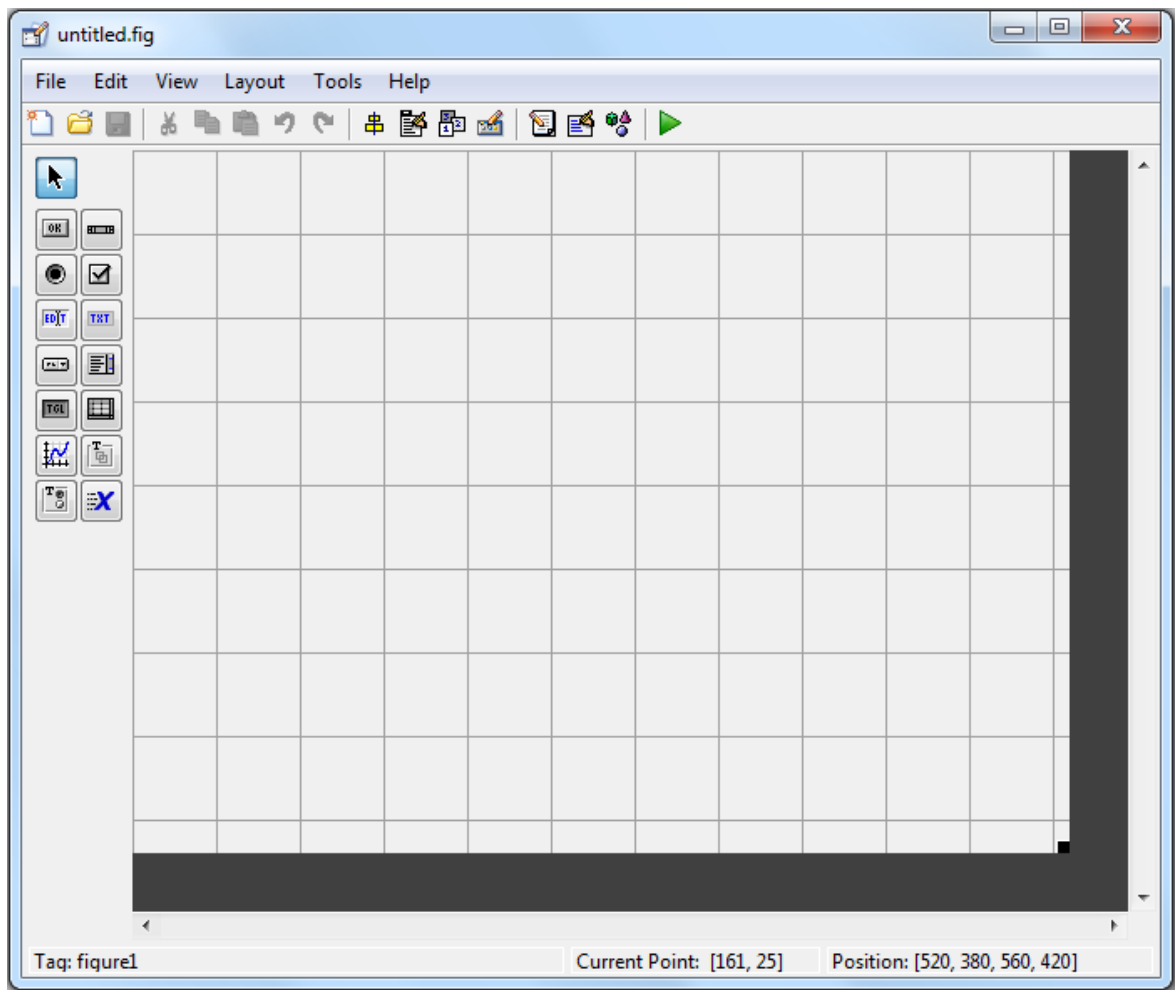


Figura 3.1. Entorno de diseño de GUI

3.1. Diseño de la interfaz

El método más sencillo de introducción de datos es a través de editores de texto. Éstos permiten que el usuario escriba los datos que se guardarán en variables que pueden ser utilizadas posteriormente. Para que se guarden los datos introducidos es necesario añadir las instrucciones para almacenarlos en variables en la función de cada uno de los editores.

Para introducir los datos se añaden varios editores de texto. Por una parte el destinado al nombre del escenario que se va a simular. A continuación se añaden cuatro editores de texto por cada tipo de datos de entrada. De este modo por cada comida, inyección de insulina o insulina basal se indicará el día, la hora, los minutos y los gramos o la dosis de insulina.

Además, se añaden textos estáticos que permitirán al usuario identificar dónde debe escribir cada dato.

Para que se puedan guardar el nombre, los datos de cada comida, inyección de insulina e insulina basal se crean cuatro listas en las que se añadirán los datos que se vayan introduciendo en la interfaz.

Para añadir los datos de los editores de texto a los listados se añaden botones. En la Figura 3.2 se puede observar el aspecto que tiene la ventana de diseño de la interfaz una vez añadidos todos estos elementos. Los botones que se han incluido en la interfaz permiten que los datos colocados en los editores previos aparezcan en el listado correspondiente.

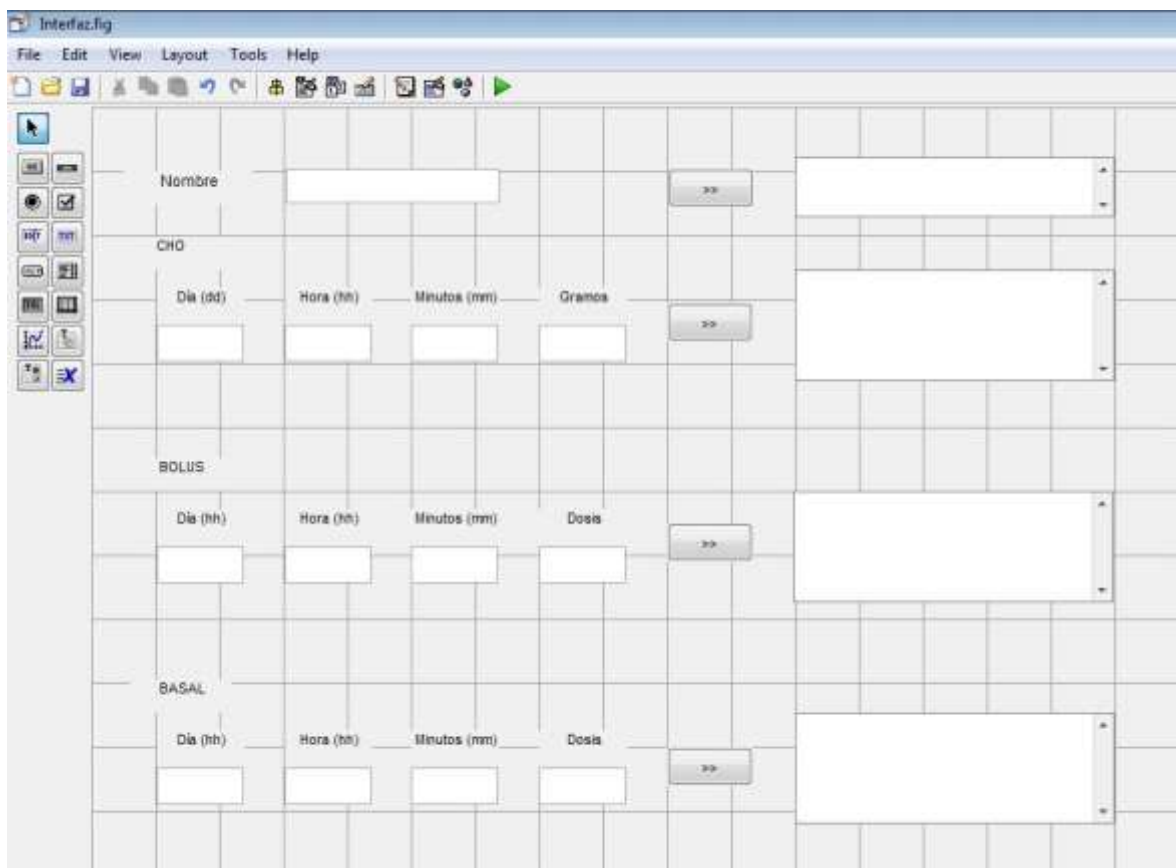


Figura 3.2. Ventana de diseño de la interfaz para introducir los datos

Para poder editar los datos introducidos se añade un botón al lado de cada listado que permite añadir los datos en los editores de texto, de modo que posteriormente se puedan añadir de nuevo al listado con las modificaciones realizadas. También se incluye un botón para eliminar elementos de las listas de modo que al pulsarlo desaparezca el elemento seleccionado.

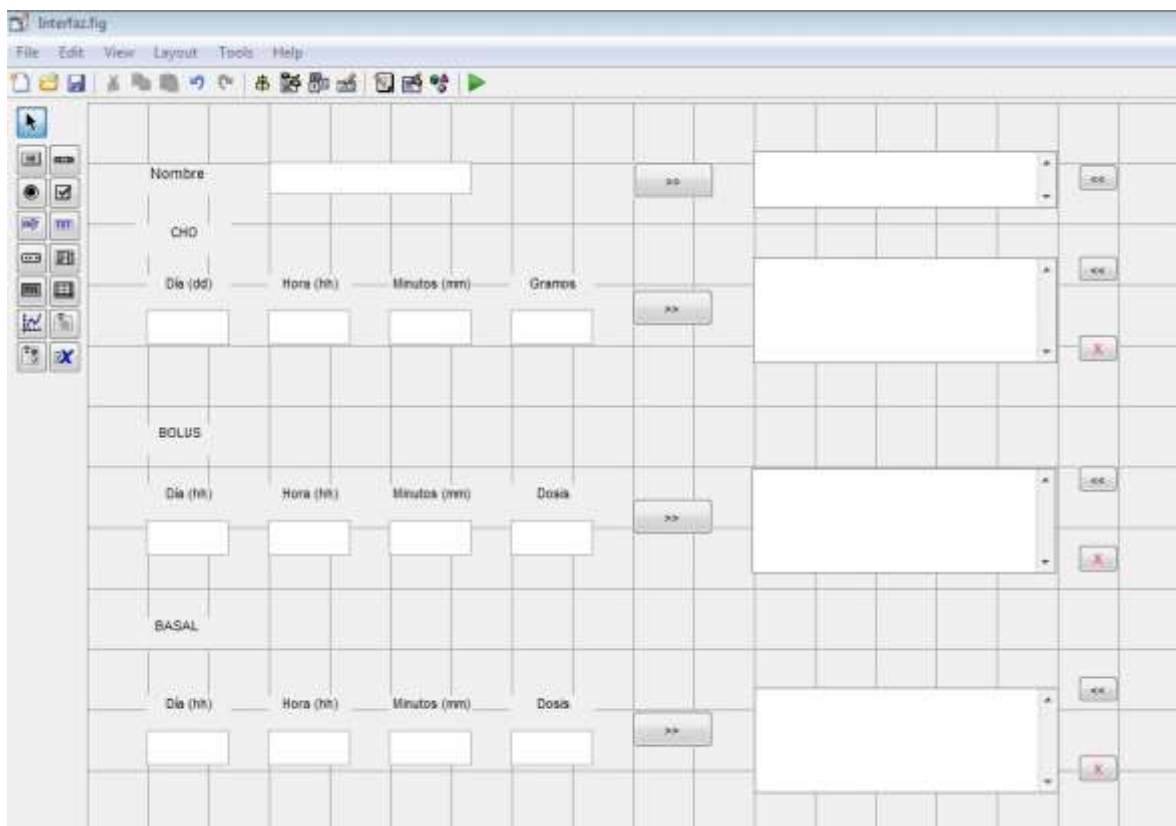


Figura 3.3. Ventana de diseño de la interfaz para modificar datos introducidos

Para que sea posible guardar el escenario que se cree para la simulación se añade en la cabecera una barra de herramientas con el botón de guardar. En el momento en el que se pulse se guardarán los datos de las listas para que posteriormente puedan ser utilizados.

Para que puedan utilizarse los datos creados previamente se incluye una lista en la que se muestran todos los escenarios ya guardados y dos botones que permiten abrir o eliminar un escenario. Cuando se selecciona abrir todos los datos se muestran en las listas de datos de la parte inferior.

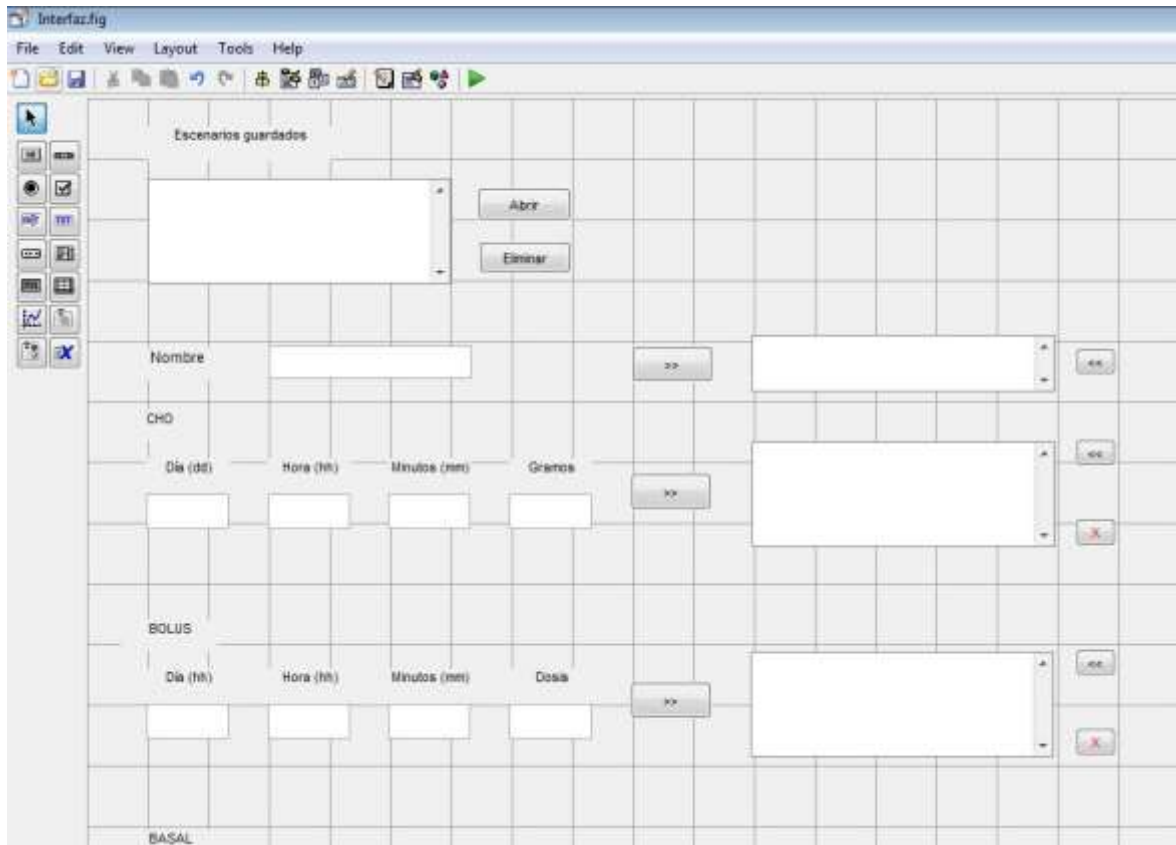


Figura 3.4. Ventana de diseño de la interfaz que permite guardar y abrir escenarios creados

Por último, se incluyen dos paneles para seleccionar el tipo de simulación. El primer panel permite seleccionar el tipo de simulación, en bloque o en tiempo real. El segundo panel no será visible al ejecutar la interfaz salvo que se seleccione la opción de simulación en tiempo real. Este panel contendrá dos editores de texto, en el primero de ellos se indicará cada cuántos minutos se quiere recibir datos de la simulación. En el segundo el factor de aceleración para la simulación en tiempo real.

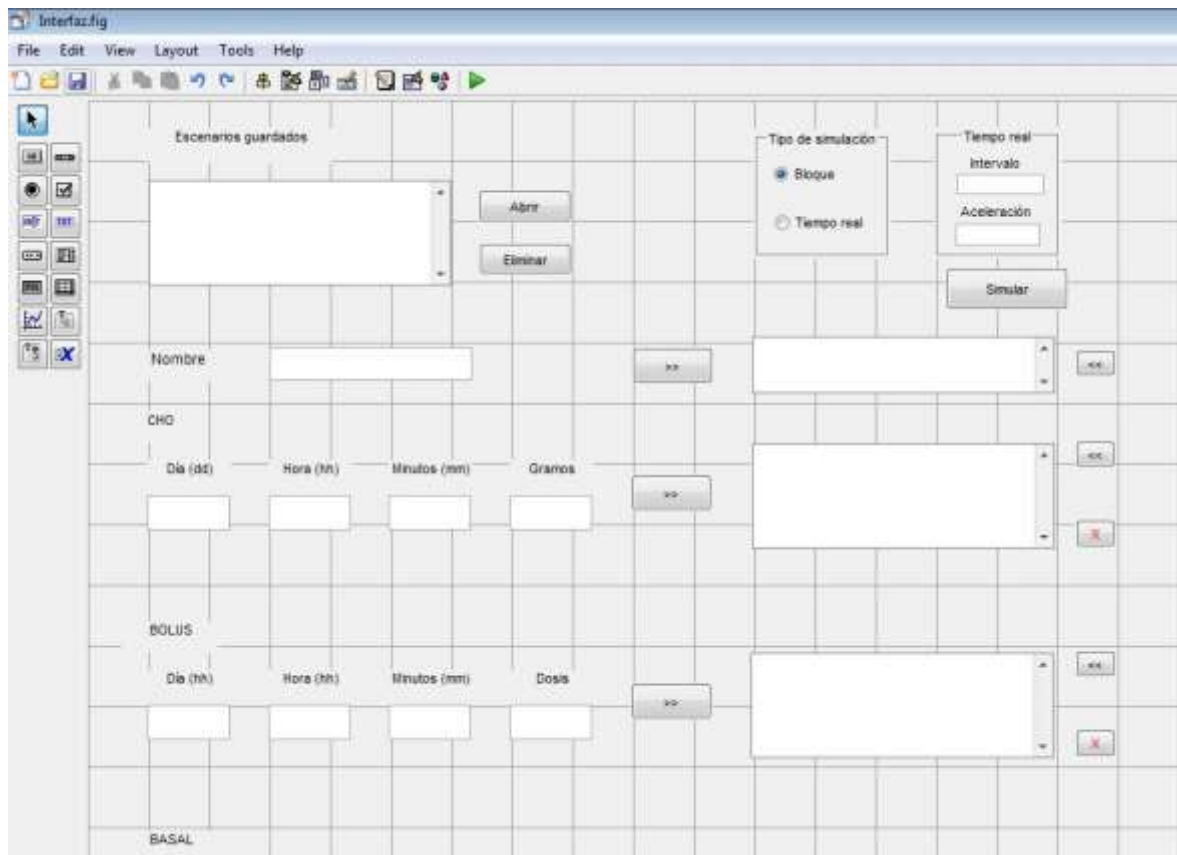


Figura 3.5. Ventana final de diseño de la interfaz

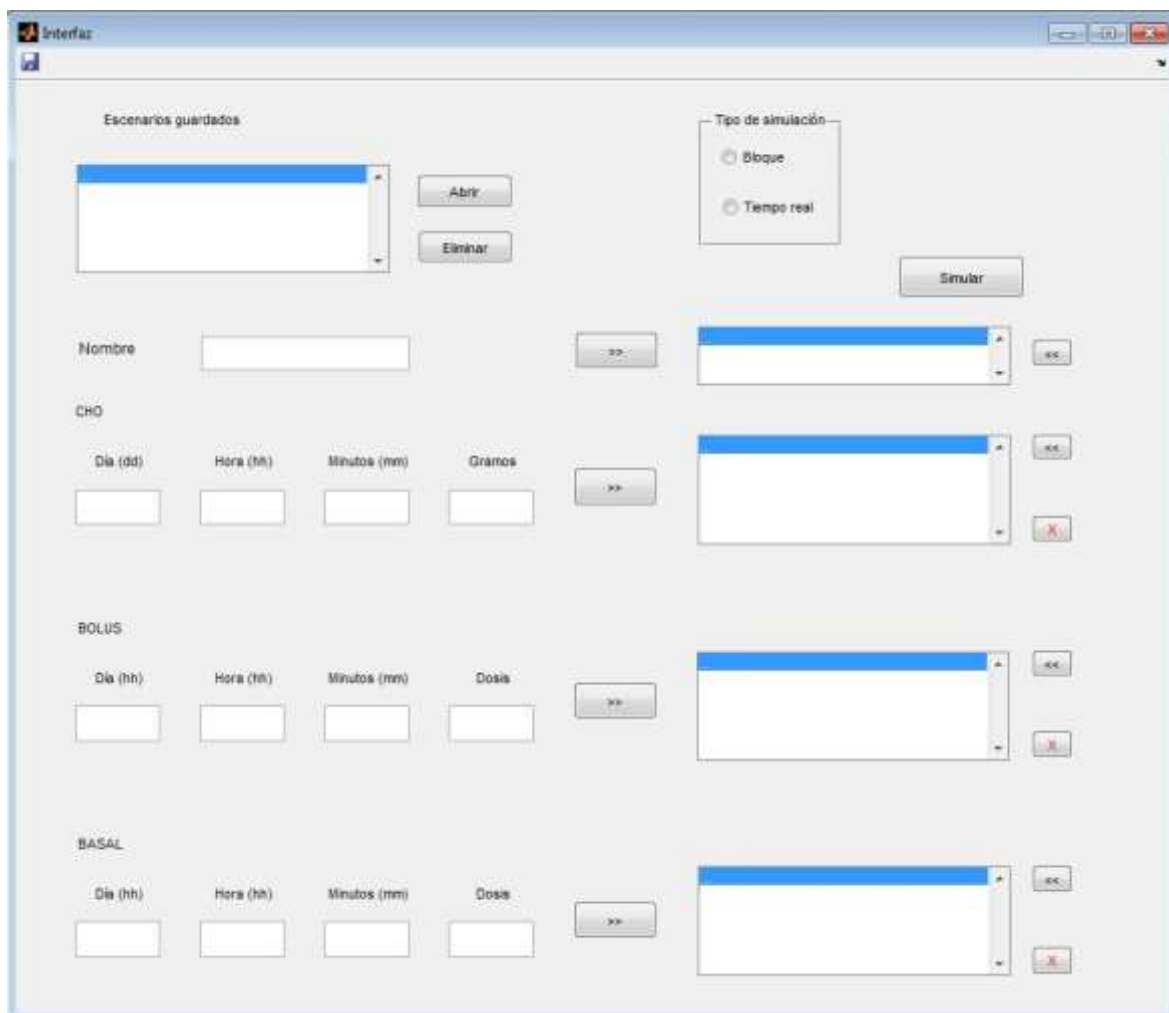


Figura 3.6. Ventana de la interfaz

3.2. Programación de la interfaz

Una vez diseñada la interfaz hay que realizar la programación de la misma para que funcione como se desea. Para ello se abre el código que se crea automáticamente al incluir los elementos en la ventana de diseño de la interfaz. Se puede observar que se han creado funciones para cada uno de los elementos en las que se debe escribir lo que debe hacer el elemento al ser utilizado.

3.2.1. Editores de texto y listas

En primer lugar hay que incluir en las funciones de todos los editores de texto y listas las líneas de código que permiten que se guarden los datos introducidos en la interfaz en variables que pueden ser utilizadas.

Para conocer el elemento seleccionado de una lista se debe añadir, en cada una de ellas, la instrucción que devuelve el valor del elemento seleccionado de la misma.

A continuación se incluye en la función de inicio de la interfaz la instrucción que permite que se muestren todos los datos de la carpeta de tipo “.mat” en la primera lista. De este modo, al iniciar la interfaz se podrán observar los nombres de todos los elementos guardados.

3.2.2. Botones de añadir datos

Para mostrar los datos introducidos en los editores de texto en las listas se deben programar los botones de añadir. En cada uno de los botones de añadir se deben leer los datos introducidos en los editores de texto y se deben cambiar a formato celda para mostrarlos en las listas. De este modo al leer los datos de las listas se obtendrá un vector columna en el que cada fila se corresponde con un dato. Esto es importante, pues de no ser así se dificulta la tarea de seleccionar un elemento y editarlo o eliminarlo.

En el caso de los botones de añadir datos de carbohidratos o insulina, antes de cambiar a formato celda, se crea una línea de texto con los datos leídos de forma que se facilite la lectura de datos introducidos. Por ejemplo, en el caso de una comida que se realiza el primer día a las dos y media de 60g de carbohidratos observaremos en la lista la siguiente línea: “Día: 01 14:30h 60g”.

Para que los datos se vayan añadiendo y no sustituyan los ya añadidos, se lee el contenido de la lista y se incluye al final la línea que se acaba de crear. Una vez hecho esto se escribe la sentencia que muestra el vector de datos en la lista.

3.2.3. Botón de guardar escenario

Para que se puedan guardar los datos hay que programar el botón de la barra de herramientas. Se va a guardar un archivo con el nombre que se haya introducido en la segunda lista de la interfaz. Este archivo va a ser una matriz en la que se van a guardar todos los datos de las tres listas.

En la primera línea de la matriz se va a indicar el número de filas de cada una de las listas. De este modo cuando se quiera leer el archivo se podrá saber en que momento acaban los datos de una lista y comienza la siguiente. El resto de filas serán los datos empezando por los carbohidratos, posteriormente los bolos de insulina y, por último, la insulina basal. La matriz contendrá cuatro columnas, una por cada dato que se almacena de cada elemento, es decir, el día, la hora, los minutos y los gramos o la dosis de insulina, según corresponda.

Para almacenar los datos como se desea se debe leer en primer lugar la lista que contiene el nombre con el que se va a guardar el archivo. Puesto que la extensión del archivo no se escribe en la interfaz es necesario añadirlo en el código. A continuación se leen los datos de las listas. Es necesario cambiar el formato de cada línea para que se pueda acceder a cada dato introducido. Una vez hecho esto es posible guardar los datos y almacenarlo en una matriz. Cuando se han obtenido las tres matrices se averigua la longitud de cada una de ellas y se crea la matriz que contiene las longitudes y las tres matrices que se han obtenido previamente. Por último se crea el archivo con el nombre correspondiente y se guarda la matriz.

3.2.4. Botones de eliminar, editar y abrir

En las funciones de los botones de eliminar se leen los elementos de la lista que corresponda y se elimina el contenido de la fila seleccionada. A continuación se añade la sentencia que permite mostrar los elementos que se habían leído y se han modificado. De este modo se observa como la fila seleccionada desaparece. En el caso del botón de eliminar escenarios se añade la sentencia que permite eliminar archivos guardados en un directorio.

En los botones de editar se leen los elementos de la lista correspondiente y se guarda en una nueva variable el contenido de la fila seleccionada. Se guardarán en distintas variables los datos de la fila y se añadirán las sentencias adecuadas para mostrar los datos en los editores de texto. Para que al editar los datos aparezcan de nuevo en la fila adecuada se añade un variable que indica si se ha pulsado el botón editar o no. En las funciones de los botones de añadir se lee esa variable y la que guarda el elemento que se

había seleccionado previamente a pulsar el botón editar. En caso de que se haya pulsado el botón editar se guardarán los datos leídos en la fila correspondiente. Si no se había pulsado, se añaden al final como ya se hacía previamente.

Cuando se pulsa el botón de abrir un nuevo escenario se deben leer todos los datos de la matriz seleccionada y escribirlos en las listas. Para ello en primer lugar se lee el nombre del archivo seleccionado y se carga, de modo que se pueda acceder a los datos que contiene. Una vez hecho esto se lee la primera fila y se guardan los datos de la longitud de las tres matrices que se escribirán en cada lista. Conocida la longitud se guardan las tres matrices correspondientes y, por último, se muestran con el formato adecuado (del mismo modo que al añadir los datos de los editores de texto) en las listas.

3.2.5. Selector del tipo de simulación y botón simular

En el panel en el que se selecciona el tipo de simulación se guarda una variable que indicará que opción se ha marcado. El segundo panel se configura de modo que sólo sea visible si se ha seleccionado la opción en tiempo real. De este panel se guardará el valor que indica el intervalo de tiempo que pasará para que se reciban datos y el valor del factor de aceleración.

Cuando se pulse el botón simular se deben leer todos los datos de las listas, a excepción del nombre, del mismo modo que al guardar un escenario. Una vez se hayan leído los datos y guardado en matrices se leerán los datos del tipo de simulación y, en caso de ser en tiempo real, los parámetros que definen esa simulación. A partir de ahí se incluye el código que ya se había implementado previamente para el simulador, sin incluir la cabecera del código en la que se definían los datos pues ahora se obtienen de la interfaz.

Capítulo 4

Desarrollo de la comunicación

Una vez desarrollado el simulador es necesario separar la parte de introducción de datos de la parte de resolución de las ecuaciones diferenciales. Este es un paso previo para, posteriormente, poder realizar la comunicación entre dos ordenadores, de modo que uno realice la simulación (el paciente virtual) y el otro permita introducir los datos y visualizar los resultados.

4.1. Programa Interfaz

El primer paso para separar la función principal en dos programas es crear una nueva función, llamada Interfaz. Esta función es del mismo tipo que la función principal, es decir, no tiene argumentos de entrada y tampoco de salida.

Para que en esta función sólo se realice la introducción de datos de entrada y la visualización de resultados se elimina la parte del código en la que se encuentran todos los parámetros del modelo y las llamadas a las funciones Valores Iniciales y ode45. De este modo sólo queda el código en el que se obtienen los datos a través de la interfaz y el código que permite la visualización de los resultados en gráficas. Si se intenta ejecutar el programa Matlab dará un error al intentar realizar las gráficas puesto que todavía no tiene los datos que necesita para ello.

4.2. Programa Simulador

Se debe crear una nueva función que no tendrá argumentos de entrada ni de salida. En esta función se debe escribir el código que se ha eliminado de la anterior, es decir, todos los parámetros del modelo y las llamadas a las funciones Valores Iniciales y ode45.

Esta función se va a ejecutar en un ordenador distinto por lo que se debe copiar al otro ordenador, incluyendo las funciones Modelo Paciente y Valores Iniciales que son necesarias para ejecutar estas líneas de código.

4.3. Comunicación

Para que se pueda ejecutar el programa es necesario establecer la comunicación entre los dos ordenadores. En primer lugar se conectan a través del puerto serie y se abre en cada uno el programa correspondiente.

Para establecer la comunicación se debe escribir en ambos programas el nombre del puerto serie correspondiente y abrir el mismo, de modo que se puedan enviar y recibir datos. Se debe tener en cuenta que en función del Sistema Operativo utilizado la instrucción para acceder al puerto serie es distinta. En este caso se ha realizado la comunicación para un ordenador con Windows, en caso de que se ejecute la misma con Mac OS o Linux se debe modificar la sentencia, de lo contrario el programa no funcionaría adecuadamente.

Al enviar datos a través del puerto se debe tener en cuenta que aunque se escriban varias líneas, como en el caso de una matriz, al hacer la lectura de los datos sólo se va a leer la primera fila, será necesario volver a leer el puerto para leer la segunda y así sucesivamente hasta finalizar con todas las líneas enviadas. Por ello, en caso de enviar matrices o vectores columna, los datos se enviarán de fila en fila.

Cada vez que se envíe un dato se entrará en un bucle de lectura del que sólo se saldrá cuando se lea la señal de Recibido. En el momento en el que reciba la confirmación, enviará el siguiente y así, sucesivamente, hasta que se hayan enviado todos los datos.

Cuando se espere recibir un dato se entrará en un bucle de lectura del que se saldrá en el momento en que se detecte un dato. Cuando esto ocurra ese dato se guardará en la variable que corresponda y se enviará la señal de Recibido. Si se van a recibir más datos comenzará de nuevo un bucle de lectura, si no se esperan más datos continuará ejecutándose el código.

Cuando se vayan a enviar matrices se enviará, en primer lugar, la longitud de la matriz y posteriormente se enviarán las filas de la matriz. De este modo el programa que recibe los datos sabe en qué momento termina la matriz y debe dejar de guardar los datos en esa variable.

En el programa Interfaz se debe añadir el código necesario para enviar los datos de la simulación justo después de la obtención de los mismos a través de la interfaz. A continuación aparecerán las líneas de código necesarias para la lectura de los resultados.

En el programa simulación se debe añadir, previo a las llamadas a las funciones de Valores Iniciales y ode45 el código que permite leer los datos de la simulación que se han enviado desde Interfaz.

A continuación se añadirá una condición que dependerá del tipo de simulación:

- Simulación en bloque. En este caso se hará la llamada a la función ode45 y se obtendrá el resultado de toda la simulación. Una vez obtenidos los resultados se enviarán al programa Interfaz que se encargará de recibir los resultados y realizar las gráficas.
- Simulación en tiempo real. En este caso se harán varias llamadas a la función ode45 con intervalos de simulación con una duración igual a la indicada por el usuario. Una vez se haya realizado una llamada se enviarán los resultados. De este modo es posible representar en las gráficas los resultados cada tantos minutos como el usuario desee. Para que la simulación se realice en tiempo real, se añade una pausa de modo que no continúe la simulación hasta que haya pasado el tiempo correspondiente. Esta pausa viene determinada por el usuario y se dividirá por el factor de aceleración para que se pueda realizar la simulación en menos tiempo.

Puesto que el modo de enviar los resultados no será el mismo en un caso que en el otro, puesto que en el primero se enviarán todos los datos en bloque y, en el segundo, se irán enviando en bloques más pequeños cada cierto tiempo se debe añadir una condición igual que en el programa de simulación. En el caso de simulación en bloque se leerán todos los datos enviados y se representarán las gráficas, finalizando así la ejecución del programa. En el caso de simulación en tiempo real se recibirá un bloque de datos, se hará la pausa correspondiente y se mostrarán los resultados obtenidos hasta el momento. Cuando haya pasado de nuevo el tiempo marcado por el usuario se añadirán los nuevos datos obtenidos y así, sucesivamente, hasta que finalice la simulación.



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

IMPLEMENTACIÓN DE UN INTEGRADOR DE MODELOS DE PACIENTES VIRTUALES DIABÉTICOS TIPO 1

DOCUMENTO 2

PRESUPUESTO

Contenido

1. Mano de obra y materiales	3
2. Precios unitarios.....	5
3. Precios unitarios descompuestos.....	7
4. Presupuesto de ejecución por contrata	11

Capítulo 1

Mano de obra y materiales

En el precio de la mano de obra se tiene en cuenta el coste por hora de un Graduado en Ingeniería en Tecnologías Industriales. Puesto que se desconoce el coste por hora debido a que actualmente aún no existe este tipo de graduado se toma el coste del Ingeniero Técnico Industrial que es el equivalente. Se sitúa este coste en 20 €/hora.

A continuación se detalla el precio de cada uno de los materiales empleados en la realización del proyecto. Estos materiales pueden agruparse en dos tipos: hardware empleado y software requerido en los equipos.

- Hardware

- o Ordenador personal de sobremesa con las siguientes características:

Procesador Intel Xeon de 2,67 Ghz	800€
Memoria RAM de 3 GB	
Disco duro de 512 GB	

- Software

o Sistema Operativo Windows 7	125 €
o Microsoft Office 2013	368 €
o Matlab versión 2014	5000 €/año
o Virtual Serial Port Emulator	0€

Capítulo 2

Precios unitarios

A lo largo de este capítulo se van a tratar de agrupar todas las tareas y operaciones realizadas en el proyecto en unidades de obra, agrupando los precios de cada producto para obtener el precio total de cada unidad de obra.

- Unidad de obra 1.

PC de sobremesa con Sistema Operativo Windows 7 preinstalado. Instalación Microsoft Office	57,468 €
--	----------

- Unidad de obra 2

Instalación del software requerido para desarrollar el proyecto: Matlab versión 2014, Virtual Serial Port Emulator	879,002 €
--	-----------

- Unidad de obra 3

Documentación y desarrollo de un simulador para un paciente virtual con diabetes tipo 1	1.260,72 €
---	------------

- Unidad de obra 4

Documentación y desarrollo de la interfaz gráfica para el simulador de paciente virtual	2.521,44 €
---	------------

- Unidad de obra 5

Documentación y desarrollo del protocolo de comunicación para la implementación final de la aplicación	2.521,44 €
--	------------

Capítulo 3

Precios unitarios descompuestos

- Unidad de obra 1

PC de sobremesa con Sistema Operativo Windows 7 preinstalado. Instalación Microsoft Office

	Importe (€)	Total (€)
1 PC de sobremesa	800	800
1 Sistema Operativo Windows 7	125	125
1 Microsoft Office 2013	368	368
1h Ingeniero Industrial	20	20
0,02 medios auxiliares	1.313	26,26
0,03 costes indirectos	1.339,26	40,178
Total		1.379,438
Coste mensual durante su vida útil (4 años)		28,734
TOTAL (2 meses)		57,468

- Unidad de obra 2

Instalación del software requerido para desarrollar el proyecto: Matlab versión 2014, Virtual Serial Port Emulator

	Importe (€)	Total (€)
1 Matlab versión 2014	5.000	5.000
1 Virtual Serial Port Emulator	0	0
1h Ingeniero Industrial	20	20
0,02 medios auxiliares	5.020	100,4
0,03 costes indirectos	5.120,4	153,612
Total		5.274,012
Coste mensual durante su vida útil (1 año)		439,501
TOTAL (2 meses)		879,002

- Unidad de obra 3

Documentación y desarrollo de un simulador para un paciente virtual con diabetes tipo 1

	Importe (€)	Total (€)
60h ingeniero industrial	20	1.200
0,02 medios auxiliares	1.200	24
0,03 costes indirectos	1.224	36,72
TOTAL		1.260,72

- Unidad de obra 4

Documentación y desarrollo de la interfaz gráfica para el simulador de paciente virtual

	Importe (€)	Total (€)
120h ingeniero industrial	20	2.400
0,02 medios auxiliares	2.400	48
0,03 costes indirectos	2.448	73,44
TOTAL		2.521,44

- Unidad de obra 5

Documentación y desarrollo del protocolo de comunicación para la implementación final de la aplicación

	Importe (€)	Total (€)
120h ingeniero industrial	20	2.400
0,02 medios auxiliares	2.400	48
0,03 costes indirectos	2.448	73,44
TOTAL		2.521,44

Capítulo 4

Presupuesto de ejecución por contrata

Descripción	Importe
1. PC de sobremesa.....	57,468 €
2. Instalación de software.....	879,002 €
3. Documentación y desarrollo del simulador.....	1.260,72 €
4. Documentación y desarrollo de la interfaz gráfica.....	2.521,44 €
5. Documentación y desarrollo del protocolo de comunicación.....	2.521,44 €
Presupuesto de ejecución material.....	7.240,070 €
Gastos Generales 13%.....	941,209 €
Beneficio Industrial 6%.....	434,404 €
Presupuesto total.....	8.615,683 €
I.V.A. 21%.....	1.809,293 €
Presupuesto de Ejecución por Contrata.....	10.424,976€

Asciende el presente presupuesto a la expresada cantidad de:

DIEZ MIL CUATROCIENTOS VEINTICUATRO EUROS CON NOVENTA Y OCHO CÉNTIMOS