



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

**DISEÑO E IMPLEMENTACIÓN DE UNA HERRAMIENTA
EDUCATIVA SOBRE LA GESTIÓN DE DIABETES TIPO 1
EMPLEANDO HERRAMIENTAS DE SIMULACIÓN 3D
DE PROCESOS INDUSTRIALES**

AUTORA: MARÍA TORRECILLA VICENTE

TUTOR: JOSÉ LUÍS DIEZ RUANO

COTUTOR: JORGE BONDÍA COMPANYY

Curso Académico: 2018-19

“A mis padres”

RESUMEN

La diabetes es una enfermedad crónica en la que los niveles de glucosa en sangre son elevados. Esto es debido a que no se produce suficiente insulina (hormona que disminuye dichos niveles) o se utiliza de manera errónea.

Dependiendo de la causa por la cual no se pueden regular los niveles de glucosa, se distinguen distintos tipos de diabetes. Cabe destacar la diabetes tipo 1 y la diabetes tipo 2. En ambos casos, para realizar un correcto tratamiento, es necesario que el paciente tenga un amplio conocimiento de la enfermedad. Si se trata de una manera inadecuada, podría llegar a ser una enfermedad mortal.

Para enseñar de forma sencilla y entretenida la gestión de la diabetes, en este proyecto se ha desarrollado e implementado una herramienta educativa acerca de la diabetes tipo 1. Para ello, se utilizan herramientas de simulación 3D de procesos industriales a través de las cuales se puede observar la glucemia (nivel de glucosa en sangre) y el flujo de la insulina (desde que es inyectada en el cuerpo hasta que llega al plasma).

Para poder simular dichos flujos se han necesitado tres programas:

- MATLAB: ejecuta un simulador de pacientes virtuales. Con los resultados obtenidos calcula, entre otras cosas, los caudales de entrada y salida de los flujos de insulina y glucosa.
- Factory IO: simula la gestión de la diabetes dentro de una nave industrial. Los tanques representan distintos compartimentos del cuerpo y el agua simula el flujo de insulina o glucosa.
- KEPServer: sirve de puente para la conexión de los anteriores programas.

Además, como la información acerca de los pacientes se ha obtenido a través del simulador, no ha sido necesaria la involucración de ningún ser vivo.

Palabras Clave: diabetes, glucosa, insulina, simulador, virtual, herramienta y educativa



RESUM

La diabetis és una malaltia crònica en què els nivells de glucosa en sang són elevats. Això és degut al fet que no es produeix suficient insulina (hormona que disminueix aquests nivells) o s'utilitza de manera errònia.

Depenent de la causa per la qual no es poden regular els nivells de glucosa, es distingeixen diferents tipus de diabetis. Podem destacar la diabetis tipus 1 i diabetis tipus 2. En tots dos casos, per a realitzar un correcte tractament, cal que el pacient tinga un ampli coneixement de la malaltia. Si es tractés d'una manera inadequada, podria arribar a ser una malaltia mortal.

Per ensenyar de manera senzilla i entretinguda la gestió de la diabetis, en aquest projecte s'ha desenvolupat i implementat una eina educativa sobre la diabetis tipus 1. Per això, s'utilitzen eines de simulació 3D de processos industrials a través de les quals es pot observa la glucèmia (nivell de glucosa en sang) i el flux de la insulina (des que és injectada en el cos fins que arriba al plasma).

Per poder simular aquests fluxos s'han necessitat tres programes:

- MATLAB: executa un simulador de pacients virtuals. Amb els resultats obtinguts calcula, entre altres coses, els cabals d'entrada i sortida dels fluxos d'insulina i glucosa.
- Factory IO: simula la gestió de la diabetis dins d'una nau industrial. Els tancs representen diferents compartiments del cos i l'aigua simula el flux d'insulina o glucosa.
- KEPServer: serveix de pont per a la connexió del anteriors programes.

A més, com la informació sobre els pacients s'ha obtingut mitjançant el simulador, no ha sigut necessària la implicació de cap ésser viu.

Paraules clau: diabetis, glucosa, insulina, simulador, virtual, eina i educativa



ABSTRACT

Diabetes is a chronic disease in which the blood glucose levels are high. This is because the organism does not produce enough insulin (hormone which decreases said levels) or it is used in the wrong way.

Depending on the cause in which the glucose levels cannot be regulated, there are different types of diabetes. The most important are type 1 diabetes and type 2 diabetes. In both cases, in order to perform a correct treatment, it is necessary for the patient to have a broad knowledge regarding the disease. If treated inadequately, diabetes could lead to death.

In order to teach diabetes management in a simple and entertaining way, this project has developed and implemented an educational tool about type 1 diabetes. To do this, 3D simulation tools of industrial processes are used, through which, glycemia (blood glucose level) and the flow of insulin (since it is injected into the body until it reaches plasma) can be observed.

In order to simulate these flows, three programs have been needed:

- MATLAB: runs a simulator of virtual patients. The inflow and outflow of glucose and insulin are calculated among other variables.
- Factory IO: simulates the management of diabetes in an industrial warehouse environment. The tanks represent different parts of the body and the water simulates the flow of insulin or glucose.
- KEPServer: serves as a bridge for the connection of the previous programs.

In addition, as far as the information about the patients has been obtained through the simulator, it has not been necessary the participation of any living being.

Keywords: diabetes, glucose, insulin, simulator, virtual, tool and educational



DOCUMENTOS

- Memoria
- Presupuesto

ÍNDICE DE LA MEMORIA

1. Planteamiento del problema	12
1.1 Objetivo del proyecto.....	13
1.2 Características del proyecto	13
2. Motivación y justificación	15
2.1 Motivación.....	15
2.2 Interés y justificación	15
3. Diabetes Mellitus	16
3.1 Páncreas	16
3.2 Glucosa	16
3.3 Glucagón, insulina y somatostatina	16
3.4 Diabetes.....	17
3.5 Páncreas artificial	22
4. Modelado de pacientes diabéticos tipo 1	23
4.1 Modelo de Cambridge.....	23
5. Simulador de pacientes, Interfaz 3D y su conexión con el simulador	28
5.1 Herramienta de simulación	28
5.2 Interfaz 3D.....	29
5.3 Conexión con el simulador	30
6. Desarrollo e implementación de la herramienta educativa	32
6.1 Interfaz gráfica del simulador.....	32
6.2 Interfaz gráfica 3D	39
6.3 Ecuaciones para la equivalencia entre modelos dinámicos.....	48
6.4 Algoritmos de intercambio de información y comunicaciones.....	50
7. Resultados	61

8. Conclusiones, limitaciones y trabajo futuro	71
8.1 Conclusiones.....	71
8.2 Limitaciones.....	73
8.3 Trabajo futuro	73
Referencias	75
Anexo I. Selección de alternativas	77
Anexo II. Manual de instalación	79
MATLAB.....	79
KEPServerEx 5.....	79
Factory IO	79
Anexo III. Manual de usuario	81
Introducción	81
Descripción de la aplicación	81
Conocimientos previos.....	81
Especificaciones técnicas	82
Puesta en marcha de la aplicación	83
Entrada de datos en la interfaz gráfica	85
Interfaz gráfica 3D	87
Anexo IV. Manual del programador	91
Introducción	91
Conocimientos previos.....	91
Primeros pasos	91
Simulador	91
Interfaz gráfica	94
Interfaz gráfica 3D.....	100
Comunicaciones	103
Ecuaciones para la equivalencia entre modelos dinámicos.....	103
Algoritmos de intercambio de información	111

ÍNDICE DEL PRESUPUESTO

Cuadro de precios	120
Unidades de obra. Precios descompuestos	120
Precios unitarios	122
Presupuesto total	123

ÍNDICE DE FIGURAS

<i>Figura 1.</i> Flujo de información	13
<i>Figura 2.</i> Glucómetro [4]	18
<i>Figura 3.</i> Sensores de glucosa intersticial [2]	18
<i>Figura 4.</i> Bomba de insulina [14]	21
<i>Figura 5.</i> Páncreas artificial disponible en el mercado [14]	22
<i>Figura 6.</i> Esquema de absorción subcutánea de insulina del modelo de Cambridge	25
<i>Figura 7.</i> Flujo de información entre programas	31
<i>Figura 8.</i> Datos de entrada para el simulador de pacientes virtuales	33
<i>Figura 9.</i> Datos de entrada erróneos en la interfaz gráfica	34
<i>Figura 10.</i> Error al intentar guardar un nombre de escenario	35
<i>Figura 11.</i> Evolución temporal de la insulina en el plasma	37
<i>Figura 12.</i> Evolución temporal de la absorción de carbohidratos	37
<i>Figura 13.</i> Evolución temporal de la glucosa en plasma	38
<i>Figura 14.</i> Evolución temporal del efecto de la insulina	38
<i>Figura 15.</i> Comparación de la evolución temporal de la insulina y la glucosa en el plasma	39
<i>Figura 16.</i> Interfaz gráfica 3D de la nave industrial	40
<i>Figura 17.</i> Tanque con la válvula de llenado abierta	41
<i>Figura 18.</i> Tanque con los actuadores y sensores	42
<i>Figura 19.</i> Plataformas	42
<i>Figura 20.</i> Cinta transportadora de materia gris	43
<i>Figura 21.</i> Canales de transporte de insulina	44
<i>Figura 22.</i> Canales de transporte de glucosa	44
<i>Figura 23.</i> Pestaña del emisor	45
<i>Figura 24.</i> Luces y alarmas	46
<i>Figura 25.</i> Cámara de navegación	46
<i>Figura 26.</i> Elección de elementos	47
<i>Figura 27.</i> Modo de no fallo	48
<i>Figura 28.</i> Archivos de MATLAB	50
<i>Figura 29.</i> Creación del canal	51
<i>Figura 30.</i> Creación del dispositivo y el grupo de etiquetas	51

<i>Figura 31.</i> Creación de las etiquetas	52
<i>Figura 32.</i> Pestaña de direcciones	52
<i>Figura 33.</i> Corrección automática de las direcciones	53
<i>Figura 34.</i> Pestaña de direcciones completada	53
<i>Figura 35.</i> Variables.....	54
<i>Figura 36.</i> Abrir OPC.....	54
<i>Figura 37.</i> OPC con todas las variables creadas	55
<i>Figura 38.</i> OPC Client DA/UA en Factory IO	55
<i>Figura 39.</i> Configuración del OPC	56
<i>Figura 40.</i> Actuadores y sensores parte 1.....	57
<i>Figura 41.</i> Actuadores y sensores parte 2.....	58
<i>Figura 42.</i> Actuadores y sensores parte 3.....	59
<i>Figura 43.</i> Función de escritura.....	59
<i>Figura 44.</i> Función de lectura.....	60
<i>Figura 45.</i> Interfaz gráfica con el nombre del escenario 1.....	61
<i>Figura 46.</i> Interfaz gráfica con los datos de entrada del escenario 1	62
<i>Figura 47.</i> Gráficas de la insulina y la glucosa en el plasma del simulador respecto al escenario 1	62
<i>Figura 48.</i> Equilibrio del escenario 1	63
<i>Figura 49.</i> Inyección del bolo del escenario 1	63
<i>Figura 50.</i> Bolo completamente administrado del escenario 1	64
<i>Figura 51.</i> Niveles de glucosa del escenario 1.....	64
<i>Figura 52.</i> Gráficas de la insulina y la glucosa en el plasma del simulador respecto al escenario 2	65
<i>Figura 53.</i> Bolo completamente administrado del escenario 2	65
<i>Figura 54.</i> Niveles de glucosa del escenario 2	66
<i>Figura 55.</i> Gráficas de la insulina y la glucosa en el plasma del simulador respecto al escenario 3	67
<i>Figura 56.</i> Bolo completamente administrado del escenario 3	67
<i>Figura 57.</i> Niveles de glucosa del escenario 3.....	68
<i>Figura 58.</i> Gráficas de la insulina y glucosa en plasma del simulador respecto al escenario 4	69
<i>Figura 59.</i> Bolo completamente administrado del escenario 4	69
<i>Figura 60.</i> Niveles de glucosa del escenario 4.....	70
<i>Figura 61.</i> Comparación de la administración del bolo en los distintos escenarios	72
<i>Figura 62.</i> Comparación del nivel de glucosa en los distintos escenarios	72
<i>Figura 63.</i> Instalación de KEPServer	79
<i>Figura 64.</i> Conexión de KEPServer	83
<i>Figura 65.</i> Selección de los archivos de MATLAB.....	84
<i>Figura 66.</i> Selección de entrada_datos y su ejecución	84
<i>Figura 67.</i> Interfaz gráfica lista para insertar los datos deseados.....	85
<i>Figura 68.</i> Datos de entrada erróneos	87
<i>Figura 69.</i> Interfaz gráfica de Factory IO.....	88
<i>Figura 70.</i> Interfaz gráfica 3D	89
<i>Figura 71.</i> Control de glucosa	90
<i>Figura 72.</i> Control de navegación	90
<i>Figura 73.</i> Iniciación de GUIDE.....	94

<i>Figura 74.</i> Nombre de la interfaz gráfica	95
<i>Figura 75.</i> Interfaz gráfica en blanco.....	95
<i>Figura 76.</i> Elementos de la interfaz gráfica.....	96
<i>Figura 77.</i> Editor de casillas.....	97
<i>Figura 78.</i> Interfaz gráfica de GUIDE	98
<i>Figura 79.</i> Interfaz gráfica de la nave industrial 3D	100
<i>Figura 80.</i> Creación de una nueva escena 3D	100
<i>Figura 81.</i> Numeración según Factory IO.....	101
<i>Figura 82.</i> Archivos de MATLAB	111
<i>Figura 83.</i> Función trabajo 1	113
<i>Figura 84.</i> Función trabajo 2	114
<i>Figura 85.</i> Función trabajo 3	114
<i>Figura 86.</i> Función trabajo 4	115
<i>Figura 87.</i> Script prueba 1	116
<i>Figura 88.</i> Script prueba 2	117
<i>Figura 89.</i> Script prueba 3	117

ÍNDICE DE TABLAS

<i>Tabla 1.</i> Tiempos de acción de cada tipo de insulina [6]	20
<i>Tabla 2.</i> Selección de alternativas.....	30
<i>Tabla 3.</i> Caudales de entrada de la insulina.....	48
<i>Tabla 4.</i> Caudales de entrada y salida de la glucosa	49
<i>Tabla 5.</i> Caudales de salida de la insulina	49
<i>Tabla 6.</i> Ponderación de criterios a través del método indirecto.....	77
<i>Tabla 7.</i> Valoración de las alternativas para cada criterio	77
<i>Tabla 8.</i> Utilización de la valoración numérica	78
<i>Tabla 9.</i> Normalización de la valoración y resultados.....	78
<i>Tabla 10.</i> Puntos para el cálculo de la constante k.....	109
<i>Tabla 11.</i> Caudales de entrada de la insulina.....	110
<i>Tabla 12.</i> Caudales de salida de la insulina	110
<i>Tabla 13.</i> Caudales de entrada y salida de la glucosa	111

MEMORIA

Capítulo 1

Planteamiento del problema

La diabetes mellitus es una enfermedad en la que los niveles de glucosa son elevados. Esto es debido a que el páncreas no produce suficiente insulina¹ o no es utilizada correctamente en el organismo. Si sucede esto último, se daría el caso de que aunque haya suficiente insulina, las células no responderían al estímulo producido por la hormona y no se conseguiría mantener la **normogluemia** (niveles normales de glucosa en la sangre).

Las personas que padecen diabetes y no siguen ningún tratamiento, suelen alcanzar unos niveles altos de glucosa y terminan hospitalizadas.

La diabetes se divide en dos tipos: diabetes tipo 1 y diabetes tipo 2.

El tratamiento para la diabetes 1, enfermedad de la cual trata el proyecto, consiste en inyectarse insulina ya que el páncreas no produce suficiente. Dicha tarea no es sencilla, se tiene que saber la cantidad de insulina a inyectar y esta depende de la cantidad de glucosa disponible en el organismo en ese momento.

Si un paciente se inyecta más insulina de la que debería podría llegar a padecer **hipogluemia** (bajos niveles de glucosa en la sangre). Si por el contrario, se inyecta menos, podría llegar a tener **hipergluemia** (altos niveles de glucosa en la sangre). Por ello, es muy importante tener en cuenta la cantidad de insulina a administrar.

Los factores que complican el tratamiento son aquellos que alteran el nivel de glucosa, como pueden ser: los carbohidratos ingeridos, el estrés o el ejercicio. Las personas que conviven con la diabetes deben de saber los horarios de sus comidas y los carbohidratos que estas puedan contener, así como la duración y la intensidad de la actividad física a realizar. Estas tareas son imprescindibles para poder realizar un correcto tratamiento.

El principal problema de la diabetes tipo 1 es que las personas son insulino dependientes y necesitan inyectarse insulina repetidas veces al día.

Hoy en día, se están haciendo estudios para conseguir un páncreas artificial. De esta manera, los diabéticos podrían tener una vida lo más similar a la de una persona sana ya que no tendrían que preocuparse constantemente por sus niveles de glucosa.

A pesar de los avances que se están haciendo, es esencial que el paciente o sus cuidadores, en caso de ser menor de edad, estén informados acerca de la diabetes ya que sería imposible realizar un tratamiento adecuado si se carece de esta información.

¹ Hormona producida por el páncreas que sirve para controlar los niveles de glucosa en sangre.

1.1 Objetivo del proyecto

Como se ha explicado anteriormente, es muy importante que los pacientes y los padres o cuidadores, tengan conocimientos acerca de la diabetes.

Por lo tanto, el objetivo del proyecto es ofrecer una herramienta de aprendizaje donde cualquier persona con o sin conocimientos acerca de la diabetes o programación, pueda entender de una manera clara y sencilla la gestión de diabetes tipo 1. De este modo, entenderán el efecto que produce la insulina, el ejercicio y los carbohidratos en los niveles de glucosa.

Para poder comparar distintos casos, se ha incluido una plataforma interactiva en la que se pueden crear diferentes escenarios cambiando algún parámetro.

1.2 Características del proyecto

Para este proyecto se ha utilizado un simulador virtual de pacientes con diabetes tipo 1, a través del cual se han obtenido resultados y se han implantado en una interfaz gráfica 3D de una manera fácil de visualizar.

Para poder desarrollarlo se han utilizado tres programas, en la figura 1 se muestra cómo se transmiten entre ellos la información.

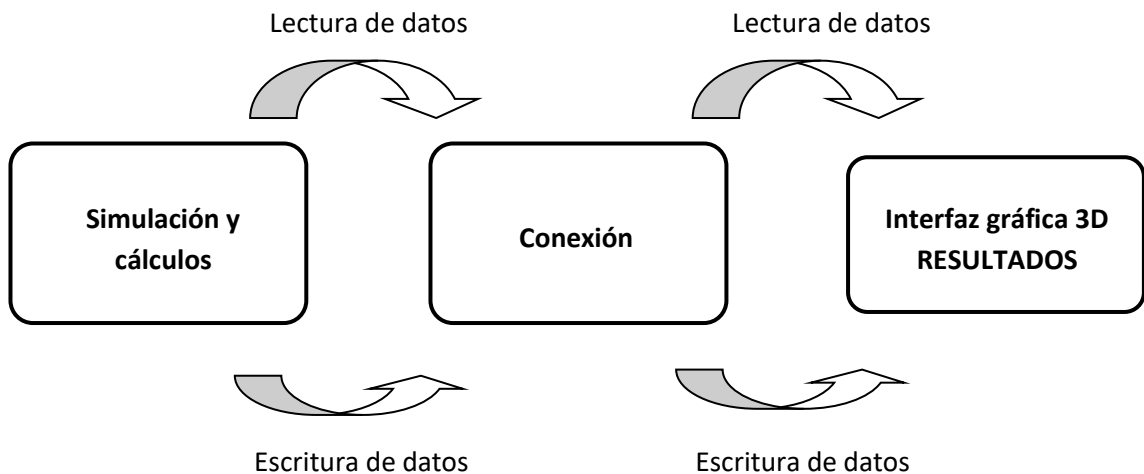


Figura 1. Flujo de información

En el primero de ellos, se introducen los datos de entrada necesarios para la simulación del paciente virtual y se obtienen los primeros resultados. Después, se realizan una serie de cálculos y se envían al programa de conexión, quien a su vez, envía los datos a la interfaz 3D. Este flujo de información se realiza en este sentido y viceversa ya que el programa de cálculos está haciendo operaciones continuamente con los resultados enviados por los sensores de la interfaz.

Como se ha explicado anteriormente, la interfaz 3D deberá de ser fácil de visualizar para cualquier persona. Además, se ha implementado otra interfaz gráfica auxiliar para poder introducir los

datos de entrada del paciente. De esta manera, no sólo será sencillo entender los resultados, sino que también lo será introducir las entradas al modelo.

Capítulo 2

Motivación y justificación

2.1 Motivación

Cada año se diagnostican más casos de diabetes por lo que más y más gente debe de estar al corriente de lo peligrosa que puede llegar a ser esta enfermedad si no se sigue un tratamiento adecuado. Para poder realizarlo, se debe de entender qué es la diabetes y cómo afecta al organismo. Por ello, es importante la implementación de herramientas educativas acerca de la gestión de la diabetes.

Los pacientes de diabetes tipo 1, son en mayoría niños por lo que debe de ser una herramienta entretenida y sencilla.

2.2 Interés y justificación

El proyecto corresponde a un Trabajo Final de Grado ya que es imprescindible para obtener el Grado en Ingeniería en Tecnologías Industriales de la Universidad Politécnica de Valencia.

Se han aprendido y puesto en práctica diferentes áreas:

- **Desarrollo de Software:** se han utilizado conocimientos adquiridos durante el grado sobre programación y se han aprendido a utilizar diferentes herramientas para el diseño e implementación de aplicaciones Software.
- **Simulación de sistemas fisiológicos:** se aprende parte del funcionamiento del organismo gracias a la simulación de pacientes virtuales.
- **Sistemas de control:** se recalca lo necesaria que puede llegar a ser la ingeniería de control y los procesos industriales.

Capítulo 3

Diabetes Mellitus

3.1 Páncreas

El páncreas es una glándula² exocrina³ situada en el abdomen, rodeada por el estómago, el intestino delgado, el hígado, el bazo y la vesícula biliar. Se encarga de producir jugo pancreático que a través de distintas enzimas, actúa en las grasas, las proteínas y los carbohidratos ingeridos durante las comidas para favorecer la digestión y absorción de alimentos [2].

Sólo un pequeño porcentaje de las células del páncreas son glándulas endocrinas⁴, las cuales forman los islotes de Langerhans que se explicarán más adelante.

3.2 Glucosa

Existe la glucosa endógena, la cual proviene del interior del organismo y la glucosa exógena, la que se forma a través de la transformación de los carbohidratos que se ingieren con los alimentos.

La glucosa se almacena en las células del hígado en forma de glucógeno y es utilizada cuando el cuerpo necesita energía.

La glucemia, nivel de glucosa en sangre, suele variar entre 70 y 145 miligramos por decilitro de sangre. En ayunas suele estar entre 70-100 mg/dl. Es muy importante tener controlado el nivel de glucemia para que el organismo pueda realizar las actividades correspondientes ya que la glucosa es quien proporciona al cerebro la mayor parte de la energía.

3.3 Glucagón, insulina y somatostatina

Los islotes de Langerhans ubicados en el páncreas están formados por conjuntos de células:

- **Células alfa (α):** producen glucagón (hormona encargada de aumentar la glucosa).
- **Células beta (β):** forman una hormona llamada insulina (disminuye la glucosa).
- **Células delta (δ):** segregan somatostatina (regula las células alfa y beta).

²Órgano compuesto por un conjunto de células que almacenan o segregan alguna sustancia.

³Glándula que libera sustancias no hormonales como las enzimas.

⁴ Glándula que produce hormonas y las vierte sin conducto a los capilares sanguíneos para que lleguen a diferentes células y realicen su función.

Si hay **hipoglucemia**, las células alfa se encargan de producir glucagón. Si por el contrario, está presente la **hiperglucemia**, las células beta segregan insulina. A través de estas dos hormonas, se consigue llegar a la **normoglucemia**.

3.4 Diabetes

La diabetes es una enfermedad crónica en la que los niveles de glucosa en sangre son elevados.

3.4.1 Tipos

Existen distintos tipos de **Diabetes Mellitus**, se van a explicar los 2 más importantes:

- **Diabetes tipo 1:**

El 15% de las enfermedades de diabetes son de este primer tipo. Aparece generalmente en niños y adolescentes aunque también puede darse el caso de estar presente en algunos adultos. Esta enfermedad aparece de forma brusca.

La diabetes tipo 1 se caracteriza por la destrucción de las células beta a través de los autoanticuerpos (es el organismo quien ataca a sus propias células). Debido a este escaso de células beta, no se llega a producir la insulina suficiente y los diabéticos tipo 1 deben inyectarse diariamente insulina para controlar la hiperglucemia [5].

Pueden aparecer los siguientes síntomas: visión borrosa, fatiga, deseo frecuente de orinar (poliuria), aumento de la sed (polidipsia), aumento del apetito (polifagia), adelgazamiento no intencionado, irritabilidad, cambios de humor y cansancio [12].

- **Diabetes tipo 2:**

El 80% son de este segundo tipo. Aparece en personas de avanzada edad quienes suelen tener problemas de obesidad. Está ligada al estilo de vida de cada persona.

En comparación con el primer tipo, la diabetes tipo 2 no destruye las células beta por lo que hay suficiente insulina. El problema es la poca eficacia de esta hormona [6].

Los síntomas que pueden aparecer son: visión borrosa, fatiga, poliuria, polidipsia, polifagia, pérdida de peso, llagas que tardan en sanar, dolores en el estómago, náuseas, vómitos y cansancio [13].

3.4.2 Complicaciones

La diabetes afecta al estilo de vida y produce complicaciones crónicas como pueden ser:

- Retinopatía: daño a los vasos sanguíneos de la retina, pueden tener fugas de líquido o cerrar el paso. Esto conlleva a una gran pérdida de visión.

- Nefropatía: impide que los riñones eliminen los productos de desecho y los líquidos adicionales del cuerpo.
- Neuropatía: se producen daños en los nervios, sobre todo en las piernas y los pies.
- Pie diabético: relacionado con la neuropatía, produce lesiones o ulceraciones en el pie.
- Enfermedad cerebrovascular y cardiovascular: el flujo de sangre en una parte del cerebro o del músculo cardíaco se detiene. Las células del cerebro pueden morir y suelen producirse infartos.

3.4.3 Medidas de glucosa

La glucosa se debe de medir frecuentemente ya sea con glucómetros o con sensores de glucosa intersticial.

- **Glucómetros** (figura 2): sirven para medir la glucosa en la sangre. Lo primero que hay que hacer es pincharse en el dedo de la mano (a través de una aguja llamada lanceta). Después, se extrae una gota de sangre y se coloca en una de las tiras del glucómetro. Por último, el glucómetro dará por pantalla la glucemia actual.
- **Sensores de glucosa intersticial** (figura 3): sirven para medir la glucosa en los tejidos. Se debe de insertar un sensor debajo de la piel, puede colocarse en el abdomen, la nalga o brazo. Su duración es entre 1 o 2 semanas. Más tarde, un transmisor enviará la señal a un receptor y mostrará el nivel de glucosa [1] [7].



Figura 2. Glucómetro [4]



Figura 3. Sensores de glucosa intersticial [2]

Ambos medidores de glucosa pueden dar valores diferentes ya que cada uno mide una glucosa distinta.

3.4.4 Diagnósticos

Existen cuatro formas de diagnosticar la diabetes:

- **Prueba de hemoglobina glucosilada (A1C):** indica la glucosa en sangre de los últimos dos o tres meses. Para ello, se mide el porcentaje de glucosa en sangre unida a la hemoglobina⁵. A mayor glucosa, mayor hemoglobina unida se detectará. Si el nivel de A1C es mayor del 6.5% es un indicador de que se padece diabetes [11].
- **Prueba de glucosa en ayunas:** mide la glucosa en sangre en ayunas, si es mayor de 126 mg/dl se tiene diabetes [11].
- **Prueba aleatoria de glucosa en sangre:** se mide la glucosa en la sangre en un momento del día, si es mayor de 200 mg/dl se considera diabetes [11].
- **Prueba oral:** primero se debe de ayunar y medirse la glucosa. Después, se bebe un líquido azucarado y a las 2 horas se vuelve a medir la glucosa. Si el valor es mayor de 200 mg/dl, se padece diabetes [11].

3.4.5 Tratamientos

Como en este proyecto se trabaja con pacientes de diabetes tipo 1, se explicarán los tratamientos correspondientes a este primer tipo:

- **Nutrición:** Una alimentación saludable ayuda a prevenir la diabetes tipo 2 y mejora los niveles de glucosa.
- **Actividad física:** Durante el ejercicio, el cuerpo libera glucosa para ser utilizada como fuente de energía.

Si no se administra insulina podría sufrirse hiperglucemia. En cambio, si se inyecta una cantidad demasiado grande, se padecerá hipoglucemia.

Si los niveles de glucosa son muy bajos, el cuerpo cogerá la grasa en vez de la glucosa como fuente de energía. Al descomponer las grasas para la obtención de energía, aparecen los cuerpos cetónicos, elementos químicos en la sangre y en la orina. Esta acción puede provocar mareos, dolores de cabeza, mal aliento, sudor fuerte, falta de apetito, náuseas, pérdida de calcio, arritmias y pérdida de músculo [3].

Por ello, los diabéticos deben de controlar el ejercicio que realizan e inyectarse la correcta dosis de insulina.

⁵ Proteína contenida en los glóbulos rojos cuya función es transportar oxígeno.

A pesar de la presencia de cuerpos cetónicos, es recomendable realizar alguna actividad física ya que regula la presión arterial, mantiene la densidad ósea, mantiene el peso corporal, etc.

- **Insulina:**

La inyección de insulina exógena sirve para compensar la falta de insulina endógena. Se hace de manera subcutánea, no directamente en el torrente sanguíneo. Esto conlleva a un retraso desde la administración hasta que se produce el efecto de dicha hormona.

Dependiendo del tiempo en el que se desea que haga efecto la insulina, existen distintos tipos:

Acción ultrarrápida y rápida: controla la glucemia durante las comidas.

Acción intermedia: controla la glucemia durante la noche, en ayunas y entre comidas.

Acción lenta y prolongada: su efecto pico es mínimo y se utiliza en los mismos casos que la intermedia.

Mixtas: es una combinación de distintas insulinas con diferentes proporciones.

Los tiempos en los que empieza a hacer efecto, el efecto máximo y la duración de cada insulina se detallan en la tabla 1.

		Inicio	Pico	Duración
ACCIÓN ULTRARRÁPIDA	Análogos rápidos	15-30 min	1-2 h	3-4 h
ACCIÓN RÁPIDA	Regular	30 min	2-4 h	6-7 h
ACCIÓN INTERMEDIA	NPH	1-2 h	6-8 h	12-16 h
ACCIÓN LENTA	Análogos basales	1-2 h	-	20-24 h
ACCIÓN PROLONGADA	Degludec	2-3 días	-	42 h

Tabla 1. Tiempos de acción de los distintos tipos de insulina [6]

Se debe diferenciar la insulina basal y los bolos de insulina. La insulina basal es aquella que hace su efecto en ayunas o en reposo y cuyo objetivo es mantener unos parámetros estables a través de una acción lenta. En cambio, los bolos de insulina son los que contrarrestan los efectos producidos por la ingesta de comida o la actividad física. Su acción es mucho más rápida.

La insulina puede ser administrada a través de diferentes métodos:

Agujas finas o plumas: el paciente debe de realizarse varios pinchazos al día (3-4 inyecciones). Debe de saber los tipos y las cantidades de insulina que necesitará, por lo que es conveniente que sepa los carbohidratos que ingerirá en las comidas.

Bombas de insulina: pequeños dispositivos programables que tienen como objetivo inyectar continuamente la cantidad de insulina necesaria en el tejido subcutáneo. Intentan imitar la función del páncreas de una persona sana, pero en ningún momento la bomba decide las cantidades de insulina a administrar. Pueden dar una recomendación, pero siempre es el paciente el que decide si dicha recomendación es aceptable o prefiere cambiarla.

Pasos a seguir para un correcto funcionamiento de la bomba:

1. Medir la glucosa con alguno de los métodos explicados anteriormente.
2. Introducir la glucosa medida en la bomba.
3. Si se van a ingerir alimentos o se va a realizar deporte se debe de indicar.
4. Aceptar o denegar la petición de la insulina a inyectar que realiza la bomba.

Aunque estos suministros son más precisos y controlan mejor la normogluemia debido a la continua administración de insulina basal, pueden llegar a ser complicados de usar. Al ingerir alimentos, se debe de programar la bomba con el ratio insulina/carbohidratos, la sensibilidad de la insulina y la duración de esta. Además, algunas personas piensan que las bombas de insulina son un método más cómodo por no tener que pincharse y que mejoran la calidad de vida del paciente. Sin embargo, estos aparatos pueden resultar incómodos ya que tienen que ser transportados en bolsillos o cinturones. Se puede observar el aspecto y funcionamiento de una bomba de insulina en la figura 4.

A) Bomba: permite al usuario introducir el nivel de glucosa. Con dicho nivel, la bomba da una recomendación sobre la insulina a suministrar y es el paciente quien decide las cantidades finales.

B) Equipo de infusión: inyecta la insulina programada por el paciente debajo de la piel.

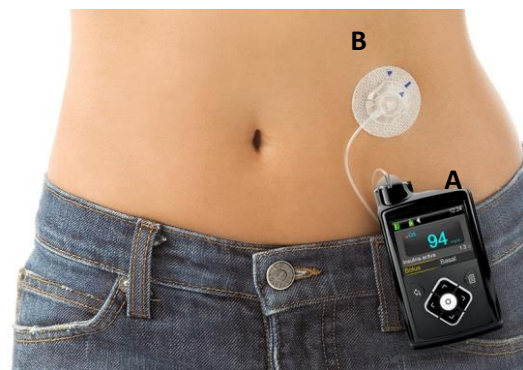


Figura 4. Bomba de insulina [14]

3.5 Páncreas artificial

Una de las funciones del páncreas es secretar al torrente sanguíneo la insulina y el glucagón necesarios. En la actualidad, se están realizando investigaciones para conseguir el páncreas artificial. El objetivo de este sistema es que las personas con diabetes puedan tener una mejor calidad de vida. Para esto, el páncreas artificial debería de realizar la función de un páncreas de una persona sana.

El sistema del páncreas artificial es completamente distinto al de las bombas de insulina. Mientras que el sistema de las bombas es un control en bucle abierto (la medición de la glucosa la realiza el paciente y es él quien indica si administra la insulina propuesta o no), el del páncreas artificial es un control en bucle cerrado (el sistema mide la glucosa y realiza lo que es más conveniente).

Estos sistemas están formados por un medidor continuo de glucosa, un algoritmo de control y un inyector de insulina.

El páncreas artificial ayudaría a los pacientes a conseguir la normoglucemia. De esta manera, no tendrían riesgo de hipoglucemia o hiperglucemia ya que se corregirían estos problemas automáticamente. Se puede observar el aspecto y funcionamiento de un páncreas artificial en la figura 5.

- A) **Monitor continuo de glucosa:** se encuentra situado debajo de la piel y transmite continuamente los niveles de glucosa a la bomba.
- B) **Bomba:** una vez se hayan indicado las comidas y el ejercicio y el monitor continuo de glucosa haya transmitido el nivel de glucemia, la bomba realizará el cálculo para determinar la insulina que se debe de administrar en el organismo.
- C) **Equipo de infusión:** inyecta la insulina programada.

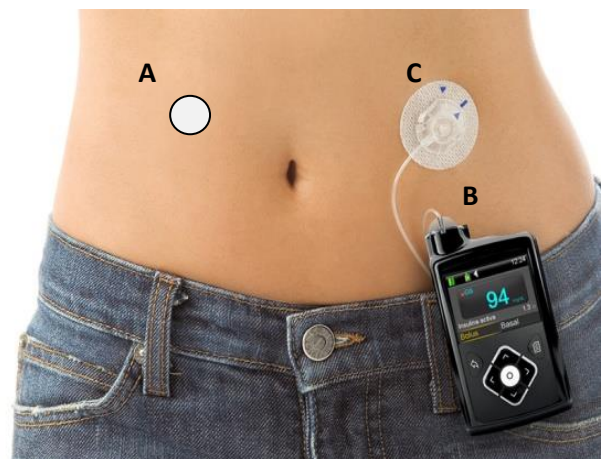


Figura 5. Páncreas artificial disponible en el mercado [14]

Capítulo 4

Modelado de pacientes diabéticos tipo 1

Como ya se ha observado en la figura 1, el primer paso fue utilizar un simulador de pacientes virtuales. Dicho simulador debía de simular la reacción de un paciente diabético ante ciertos estímulos como pueden ser los carbohidratos ingeridos, el ejercicio realizado, el peso del paciente y la insulina inyectada.

Para elegir un simulador se debe de tener en cuenta el modelo que se ha llevado a cabo para su implementación. Por ello, se empezó eligiendo el modelo.

Como primera opción, se pensó en trabajar con el modelo de UVA (Universidad de Virginia)/Padova [10] ya que es bastante completo y se obtienen resultados muy aproximados a los reales. Sin embargo, el objetivo del proyecto resultaba difícil de cumplir al utilizar un modelo tan complejo. Los niveles de insulina dependían de una gran cantidad de factores y su representación ya no era sencilla.

Se buscó una segunda opción, el modelo de Cambridge [8] [15]. Este modelo utiliza ecuaciones más sencillas para la representación de insulina, por lo que al transmitir la información a la interfaz 3D sí que se podía implementar una aplicación sin complicaciones para el usuario. Por ello, esta opción fue la elegida para el proyecto.

4.1 Modelo de Cambridge

El modelo elegido tiene distintos subprocesos los cuales se explican a continuación:

4.1.1 Subprocesos del modelo

- **Absorción subcutánea de insulina:** representa cómo la insulina exógena se introduce bajo la piel y llega al plasma. Esto sucede a través de dos compartimentos en el torrente sanguíneo. Más tarde, esta hormona llega al plasma y es consumida por el organismo. Las ecuaciones del modelo se muestran a continuación:

$$\dot{S}_1 = u - \frac{S_1}{t_{max,I}} \quad (4.1)$$

$$\dot{S}_2 = \frac{S_1}{t_{max,I}} - \frac{S_2}{t_{max,I}} \quad (4.2)$$

$$\dot{I} = \frac{U_i}{V_i} - k_e \cdot I \quad (4.3)$$

Donde:

S_1 y S_2 : son el compartimento de entrada y compartimento intermedio para la absorción de insulina subcutánea en mU (miliUnidades de insulina).

u : es el flujo de insulina que inyecta la bomba (insulina basal y bolo) en mU/min.

$t_{max,I}$: es el tiempo máximo de absorción inyectando insulina de acción corta.
 $t_{max,I} = 55$ min.

I : es la concentración de insulina en plasma en mU/L.

V_i : es el volumen de distribución de insulina. $V_i = 0.12 \text{ L} \cdot \text{kg}^{-1}$

ke : es la tasa de eliminación de insulina. $Ke = 0.138 \text{ min}^{-1}$

U_i : es la tasa de absorción de insulina en el plasma en mU/min y se define como:

$$U_i = \frac{S_2}{t_{max,I}} \quad (4.4)$$

En la figura 6 se muestra un esquema del modelo de absorción subcutánea de insulina.

- **Absorción de glucosa:** incluye la ingestión, digestión y absorción de la glucosa desde el intestino hasta la sangre. El subproceso está modelado con dos compartimentos y sus ecuaciones son las siguientes:

$$\dot{g}_1 = a_g * u_{cho} - \frac{g_1}{t_{maxG}} \quad (4.5)$$

$$\dot{g}_2 = \frac{g_1}{t_{maxG}} - \frac{g_2}{t_{maxG}} \quad (4.6)$$

Donde:

g_1 y g_2 : son los compartimentos de transición para la glucosa en el proceso de digestión en mmol.

a_g : es la porción de carbohidratos ingeridos que llegan al sistema circulatorio.

u_{cho} : es la cantidad de carbohidratos ingeridos en mmol/min.

t_{maxG} : es el tiempo máximo de absorción de los carbohidratos. Este parámetro regula la velocidad de transferencia entre compartimentos en min.

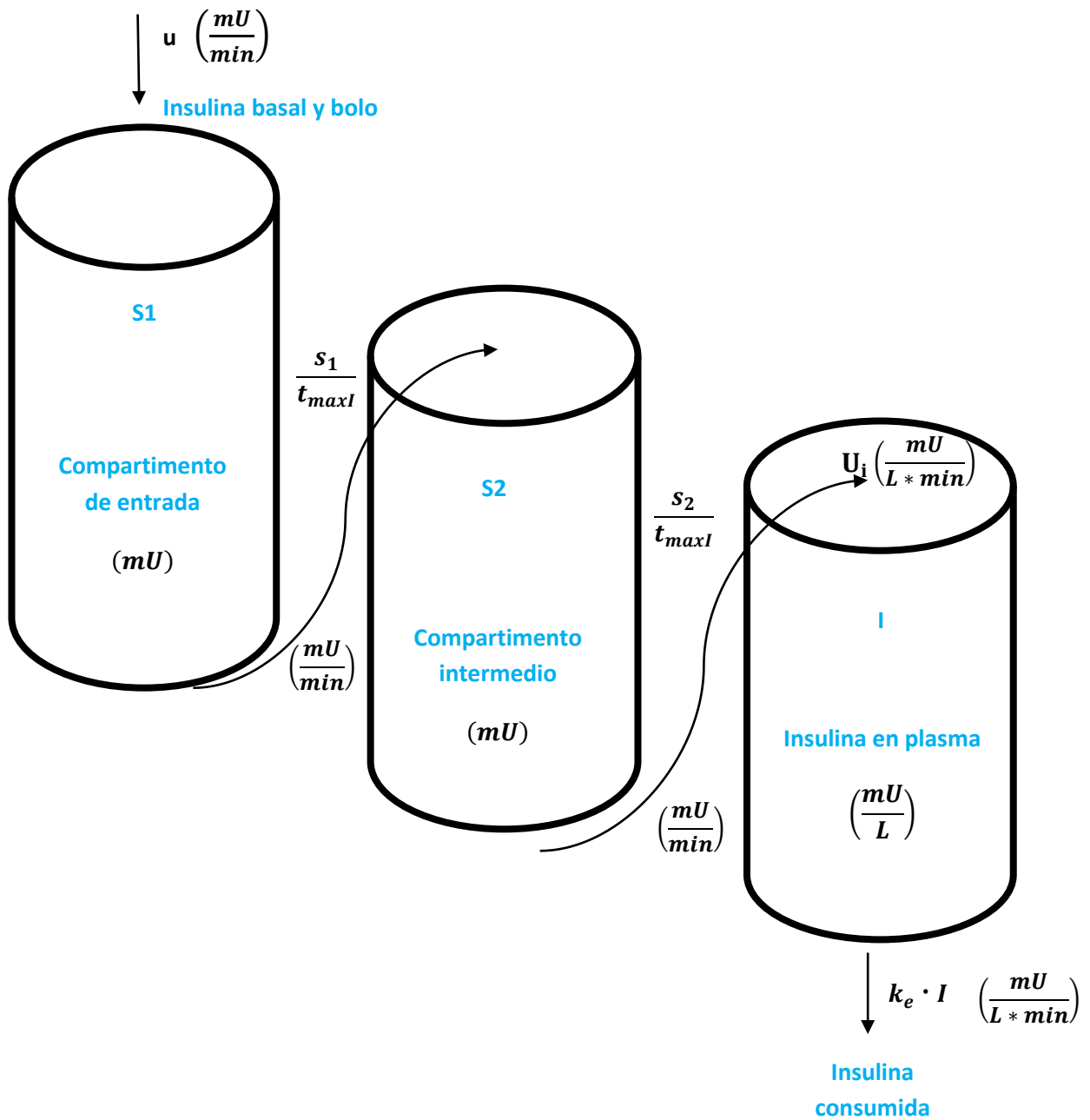


Figura 6. Esquema de absorción subcutánea de insulina del modelo de Cambridge

- **Regulación de glucosa:** representa la regulación endógena de la glucosa.

La insulina produce efectos en la glucosa de tres maneras distintas:

Incrementa el flujo de la glucosa de la sangre a los tejidos.
Aumenta la captación de glucosa en los músculos y tejidos adiposos⁶.
Inhibe la producción de glucosa en el hígado.

Dichos efectos se representan mediante las siguientes ecuaciones:

$$\dot{x}_1 = -k_{a1} * x_1 + k_{b1} * I \quad (4.7)$$

$$\dot{x}_2 = -k_{a2} * x_2 + k_{b2} * I \quad (4.8)$$

$$\dot{x}_3 = -k_{a3} * x_3 + k_{b3} * I \quad (4.9)$$

Donde:

x_1 : es la acción de la insulina en el transporte de glucosa en min^{-1} .

x_2 : es la acción de la eliminación de glucosa en min^{-1} .

x_3 : es la acción de la producción de glucosa.

k_{a1}, k_{a2} y k_{a3} : son los ratios de desactivación con los siguientes valores:

$$k_{a1} = 0.006 \text{ min}^{-1}$$

$$k_{a2} = 0.06 \text{ min}^{-1}$$

$$k_{a3} = 0.03 \text{ min}^{-1}$$

k_{b1}, k_{b2} y k_{b3} : son los ratios de activación con los siguientes valores:

$$k_{b1} = \text{Sit} * k_{a1} \quad \text{siendo } \text{Sit} = \gamma * 5.120 * 10^{-3} \frac{\text{mU}}{\text{min} * \text{L}}$$

$$k_{b2} = \text{Sid} * k_{a2} \quad \text{siendo } \text{Sid} = \gamma * 8.200 * 10^{-4} \frac{\text{mU}}{\text{min} * \text{L}}$$

$$k_{b3} = \text{Sie} * k_{a3} \quad \text{siendo } \text{Sie} = \gamma * 0.052 \frac{\text{mU}}{\text{min} * \text{L}}$$

Con γ haciendo referencia a la sensibilidad de la insulina en tanto por uno.

El modelo completo de la regulación de glucosa es el siguiente:

$$G = \frac{Q_1}{V_G} \quad (4.10)$$

⁶ Tejido formado por la asociación de células que acumulan lípidos en el citoplasma.

$$\dot{Q}_1 = - \left[\frac{F_{01}^c}{V_G * G} + x_1 \right] * Q_1 + k_{12} * Q_2 - F_R + U_G + EGP_0 * [1 - x_3] \quad (4.11)$$

$$\dot{Q}_2 = x_1 * Q_1 - [k_{12} + x_2] * Q_2 \quad (4.12)$$

Donde:

G: es la concentración de glucosa en mmol/L.

Q₁: es la masa de la glucosa en el compartimento accesible (directamente proporcional a la concentración de glucosa) en mmol.

Q₂: es la masa de la glucosa en el compartimento no accesible en mmol.

V_G: es el volumen de distribución de la glucosa. $V_G = 0.16$ L/kg

F₀₁^c: es el flujo de glucosa independiente de la insulina en mmol/min.

$$F_{01}^c \left\{ \begin{array}{l} f_{01} \left(\frac{mmol}{min} \right) \quad \text{si } G \geq 4.5 \left(\frac{mmol}{L} \right) \\ \frac{f_{01} * G}{4.5} \left(\frac{mmol}{min} \right) \quad \text{en otro caso} \end{array} \right\} \quad \text{siendo } f_{01} = 0.0097 \text{ mmol}/(\text{min} * \text{kg})$$

k₁₂: es el ratio de transferencia. $k_{12} = 0.066 \text{ min}^{-1}$

F_R: es la glucosa renal en mmol/min.

$$F_R \left\{ \begin{array}{l} 0.003 * (G - 9) * V_G \left(\frac{mmol}{min} \right) \quad \text{si } G \geq 9 \left(\frac{mmol}{L} \right) \\ 0 \left(\frac{mmol}{min} \right) \quad \text{en otro caso} \end{array} \right.$$

U_G: es el flujo de absorción intestinal de glucosa en mmol/min y se define como:

$$U_G = \frac{g_2}{t_{max,G}} \quad (4.13)$$

EGP₀: es la producción endógena de glucosa. $EGP_0 = 0.0161 \text{ mmol}/(\text{min} * \text{kg})$

Capítulo 5

Simulador de pacientes, interfaz 3D y su conexión con el simulador

Una vez se eligió el modelo, el siguiente paso fue obtener el simulador de pacientes virtuales.

5.1 Herramienta de simulación

Un simulador médico sirve para realizar diagnósticos sobre pacientes virtuales evitando utilizar seres vivos, ahorrando tiempo y usando pocos recursos.

Los integrantes del Instituto Universitario de Automática e Informática Industrial de la Universidad Politécnica de Valencia, quienes apuestan por la investigación e innovación multidisciplinar en el sector de la tecnología aplicada a la salud, tienen como objetivo ayudar a las personas que padecen diabetes y conseguir que sus vidas sean lo más parecidas a las de una persona sana. Todo ello lo logran a través de investigaciones en las áreas de robótica, informática industrial, control de procesos, visión por computador e informática gráfica y multimedia hacia las aplicaciones de salud [9].

Este grupo de investigadores dispone de varios simuladores, uno de ellos fue desarrollado utilizando el modelo de Cambridge. Como dicho modelo era el elegido para este proyecto, se optó por utilizar ese simulador.

Las simulaciones realizadas son de pacientes virtuales con diabetes tipo 1 en los que se pueden cambiar factores como el peso, la sensibilidad a la insulina, la hora a la que el paciente ingiere alimentos, la cantidad de carbohidratos, los bolos de insulina, la insulina basal y el ejercicio que realiza el paciente. Gracias a estos factores, los pacientes virtuales se comportan como una persona real.

A través del simulador se consiguió un control automático de la glucemia sin ninguna intervención de ningún paciente.

Para el desarrollo del simulador, el equipo de investigadores necesitaba un programa capaz de realizar ecuaciones diferenciales y utilizó MATLAB (MATrix LABoratory).

MATLAB, un programa de MathWorks, es un sistema de cómputo numérico que ofrece un entorno de desarrollo integrado con un lenguaje de programación propio de alto nivel (lenguaje M). Se pueden utilizar matrices, representar datos y funciones, implementar algoritmos, crear interfaces de usuario y comunicarse con programas en otros lenguajes y con otros dispositivos hardware. Además, dispone de dos herramientas adicionales: **Simulink** (entorno de programación visual) y **GUIDE** (editor de interfaces de usuario). También dispone de distintas herramientas para poder ampliar sus capacidades.

Este programa se explica durante el Grado en Ingeniería en Tecnologías Industriales en la Universidad Politécnica de Valencia por lo que se pudieron aplicar los conocimientos adquiridos.

5.2 Interfaz 3D

Tras tener el modelo y el simulador, se necesitaba implementar una interfaz para representar los resultados deseados. Se pensó en desarrollar una interfaz 3D en vez de una en 2D porque suelen ser más llamativas y de más agrado para los usuarios.

Se disponían de distintas alternativas para la interfaz:

- Factory IO
- Simulink 3D Animation
- Blender 3D
- AC3D

Simulink 3D Animation conecta los modelos de Simulink y los algoritmos de MATLAB para realizar interfaces gráficas en 3D. Sin embargo, este programa se descartó desde el inicio ya que se querían aprovechar las herramientas comerciales específicas para la representación 3D. De esta manera, se tendría la oportunidad de aprender cómo realizar la interconexión de variables entre diferentes aplicaciones.

Blender 3D es un programa para el que hacer detalles en 3D es perfecto, pero se necesita invertir bastante tiempo para aprender a usarlo. Además, no es intuitivo y esto complica el aprendizaje. En cambio, con AC3D sólo se necesitan un par de horas para poder crear un primer objeto.

Al comparar Blender 3D y AC3D con Factory IO, se puede observar que este último ya dispone de un catálogo de piezas fabricadas que pueden seleccionarse y colocarse en los lugares deseados. Gracias a dicho catálogo, se ahorran horas de trabajo. Sin embargo, se debe tener en cuenta que al tener las piezas ya disponibles, no se pueden variar sus dimensiones ni realizarles ningún cambio. Es un programa menos flexible.

La elección entre los últimos tres programas era complicada ya que no se tenían conocimientos previos de ninguno. Los pasos que se siguieron para la selección se enumeran a continuación:

1. Elegir distintas alternativas.
2. Seleccionar los criterios a evaluar.
3. Ponderación de criterios (método indirecto).
4. Valoración de las alternativas para cada criterio.
5. Ordenar las alternativas a través de la suma ponderada.
6. Elegir la mejor alternativa.

Como se puede ver en la siguiente tabla, a la conclusión que se llegó tras la selección de alternativas fue Factory IO ya que era el que mayor resultado obtuvo (número en amarillo). El procedimiento de selección se desarrolla en el Anexo I.

CARACTERÍSTICAS

PROGRAMAS	Complejidad	Tiempo	Gratuito	Limitaciones	Estética	
	0,14	0,17	0,275	0,22	0,195	
Factory IO	0,444	0,400	1,000	1,000	0,600	0,041
AC3D	0,667	0,320	0,000	0,625	0,800	-0,569
Blender 3D	1,000	1,000	1,000	0,250	1,000	-0,342
	min	min	máx	min	máx	

Tabla 2. Selección de alternativas

Factory IO es un programa de simulación 3D de naves industriales en el que el usuario puede utilizar escenas ya disponibles o crear las suyas con elementos ya existentes.

Gracias a este programa se puede adquirir un amplio y rápido conocimiento a cerca de la automatización ya que es fácil de usar. Además, Factory IO estaba disponible a través del Departamento de Ingeniería de Sistemas y Automática de la Universidad Politécnica de Valencia.

5.3 Conexión con el simulador

Para poder enviar los datos del simulador a Factory IO se necesitaba un programa que los conectara. Se eligió la tecnología OPC ya que tanto MATLAB como Factory IO podían conectarse a través de dicha comunicación. Es un modo de comunicación en el control de procesos industriales que permite que diferentes dispositivos software compartan datos. La comunicación se realiza a través de una arquitectura Cliente-Servidor. El servidor es quien ofrece un servicio y el cliente, quien pide dicho servicio y espera resultados.

Las alternativas que se escogieron fueron las siguientes:

- KEPServer
- DxpServer (DeviceXPlorer OPC Server) de Takebishi.
- MatrikonOPC de Matrikon

Como durante el grado se habían impartido clases en las que se explicó brevemente la función principal de KEPServer, ya se disponían de algunos conocimientos previos. Además, este programa también estaba disponible a través del Departamento de Ingeniería de Sistemas y Automática. Fueron estos dos motivos los que llevaron a KEPServer a ser el programa elegido para el presente proyecto.

KEPServer, producto software de la compañía KEPWARE, es un servidor de datos basado en la tecnología OPC para comunicar dispositivos. Su comunicación es bidireccional (recibe y envía datos).

Al comparar las figuras 1 y 7, se puede apreciar el papel que lleva asociado cada uno de los tres programas.

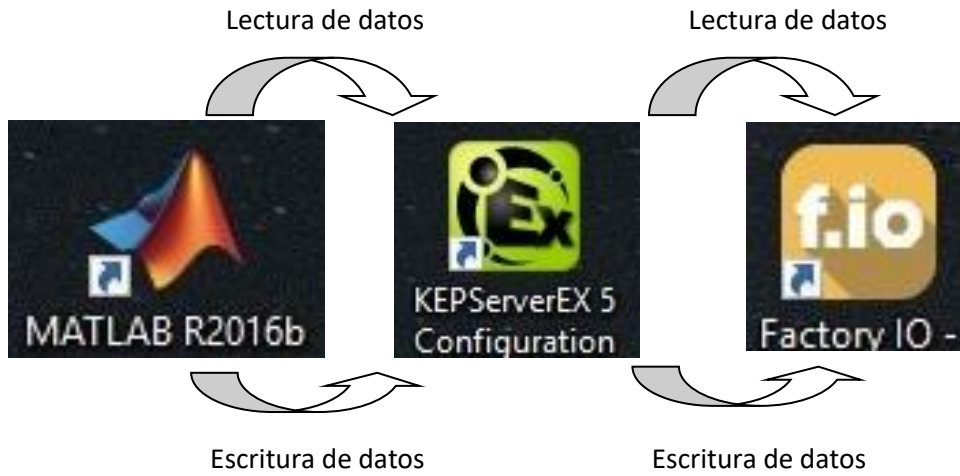


Figura 7. Flujo de información entre programas

Capítulo 6

Desarrollo e implementación de la herramienta educativa

Para poder desarrollar e implementar la herramienta educativa enunciada en el proyecto, se tuvo que crear una interfaz gráfica para la entrada de datos al simulador ya que el futuro usuario podía carecer de conocimientos de programación.

6.1 Interfaz gráfica del simulador

La interfaz gráfica desarrollada se muestra en la figura 8. Se implementó a través de GUIDE, herramienta de la que dispone MATLAB.

Los datos de entrada introducidos por el usuario son los siguientes:

- **Nombre del nuevo escenario:** nombre del escenario que se quiere crear.
- **Horas de las comidas:** se debe de poner la hora del desayuno, comida y/o cena. Se puede introducir cualquier hora entre las 0 y las 23 y cualquier minuto entre el 0 y el 59.
- **Carbohidratos ingeridos:** deben de apuntarse los gramos de carbohidratos ingeridos en cada comida.
- **Horas de los bolos de insulina:** se realiza igual que las horas de las comidas.
- **Bolos de insulina:** se apuntan las U/min de insulina que se administran.
- **Ejercicio:** si se va a realizar ejercicio se tiene que rellenar la casilla de la hora y los minutos igual que en las comidas y bolos. Además, se tiene que indicar el tiempo que dura la actividad física en minutos.
- **Peso:** es el peso del paciente en kg.
- **Sensibilidad a la insulina:** sensibilidad de la insulina en tanto por uno.
- **Basal:** insulina basal que se suministra continuamente. Se mide en U/min.

Se debe de tener en cuenta que en cualquier actividad que no se desee realizar se deberá introducir un cero.

PERSONA CON DIABETES TIPO1

NOMBRE DEL NUEVO ESCENARIO

GUARDAR ESCENARIO

MOSTRAR ESCENARIOS

MOSTRAR VARIABLES DEL ESCENARIO

ELIMINAR ESCENARIO

	Hora [0-23]	Minutos [0-59]	CHO (g)	Hora [0-23]	Minutos [0-59]	Bolo (U/min)
DESAYUNO	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
COMIDA	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
CENA	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

	Hora [0-23]	Minutos [0-59]	Duración (min)	Peso (Kg)	Sensibilidad [0-1]	Basal (U/min)
EJERCICIO	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

RESTAURAR ÚLTIMOS DATOS ELIMINAR VARIABLES **START**

Figura 8. Datos de entrada para el simulador de pacientes virtuales

Además de los datos introducidos el usuario dispone de 7 botones:

- **Guardar escenario:** guarda en un archivo de MATLAB el escenario con el nombre y los datos introducidos. El nombre del escenario creado aparecerá en la nueva lista.
- **Mostrar escenarios:** muestra en la lista el nombre de todos los escenarios creados que no hayan sido eliminados.
- **Mostrar variables del escenario:** muestra las variables del escenario seleccionado.
- **Eliminar escenario:** elimina el escenario seleccionado.
- **Restaurar últimos datos:** hace aparecer por pantalla los últimos datos correctamente introducidos.
- **Eliminar variables:** elimina las variables de la pantalla.
- **Start:** si han sido insertadas correctamente todas las variables empezará la simulación. En caso contrario, aparecerá un mensaje indicando los datos erróneos.

A la hora de utilizar la interfaz gráfica el usuario puede introducir algunos valores no válidos o dejar alguna casilla sin rellenar. Por ello, se puso un control de errores que detecta los siguientes fallos por parte del usuario:

- Si se clicca el botón START pero algún valor es erróneo o se ha dejado en blanco, el nombre de esa variable aparecerá en rojo y podrá leerse un mensaje como el de la figura 9. En dicha figura, las variables de ejercicio y peso aparecen en rojo. Esto es debido a que no se ha incluido ninguna variable en ejercicio y si no se desea realizar dicha actividad hay que escribir un cero. En cambio, la variable peso es errónea porque se ha introducido un número negativo. Se deben de cambiar estas variables y volver a presionar el botón START.
- Si se clicca el botón GUARDAR ESCENARIO sin haber introducido un nombre al escenario, aparecerá un mensaje de error como en la figura 10.

	Hora [0-23]	Minutos [0-59]	CHO (g)	Hora [0-23]	Minutos [0-59]	Bolo (U/min)
DESAYUNO	8	0	20	8	0	2
COMIDA	14	30	60	14	30	6
CENA	21	0	30	21	0	3

	Hora [0-23]	Minutos [0-59]	Duración (min)	Peso (Kg)	Sensibilidad [0-1]	Basal (U/min)
EJERCICIO				-3	0.39	0.016671

Las variables en rojo tienen un valor erróneo. Si no desea realizar alguna actividad introduzca un cero.

RESTAURAR ÚLTIMOS DATOS ELIMINAR VARIABLES **START**

Figura 9. Datos de entrada erróneos en la interfaz gráfica

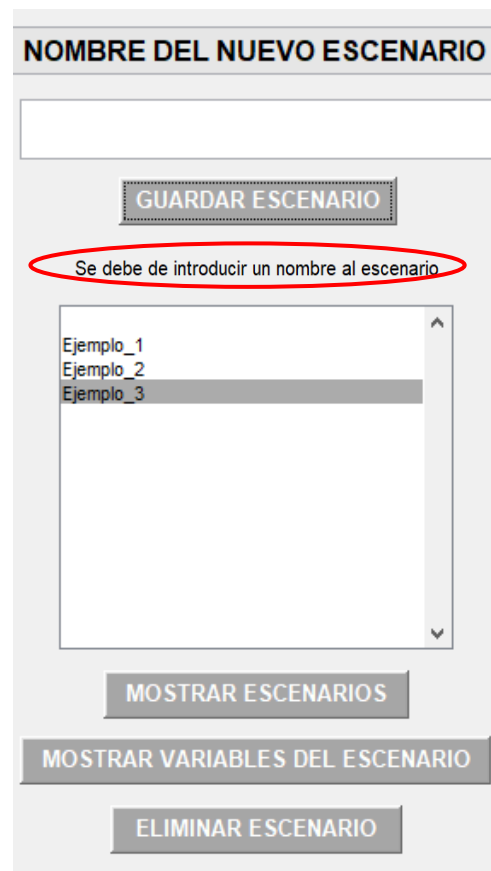


Figura 10. Error al intentar guardar un nombre de escenario

Si todos los datos son válidos y se pulsa el botón START, automáticamente se ejecuta el simulador de pacientes virtuales y se guardan los siguientes resultados en un archivo de MATLAB llamado datos_simulador:

- **Vector de tiempos de la simulación.**
- **Matriz de resultados:** formada por 10 columnas, una para 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). Cada fila representa el instante de tiempo definido en el vector tiempos.
- **Vector basal:** tiene una longitud igual a dos. Solo es interesante la segunda posición del vector en la que se indican las U/min de insulina basal a administrar.
- **Matriz bolo:** tiene 3 columnas. Solo se necesitarán las dos primeras: la que indica los minutos a los cuales se realiza la inyección de un bolo de insulina y la que indica las U/min del bolo.
- **Constantes:** Aparecen algunas constantes del modelo de Cambridge: k_e , t_{maxI} , t_{maxG} , V_i , V_g , k_{12} , $EGPO$ y f_{01} .

Una vez obtenidos todos los resultados, el simulador representa distintas gráficas según los datos de entrada. Las gráficas son las siguientes:

- **Figura 1:** representa la insulina en plasma (mU/L) a lo largo del tiempo, los picos que sobresalen son las inyecciones de insulina del paciente (figura 11).
- **Figura 2:** se observan los carbohidratos ingeridos a lo largo del tiempo, los picos son los mg de carbohidratos que el paciente tiene en cada comida (figura 12).
- **Figura 3:** muestra los efectos de la insulina, como ya se ha explicado antes x_1 hace referencia al transporte de glucosa, x_2 a la eliminación de la glucosa y x_3 a la producción de glucosa (figura 13).
- **Figura 4:** representa la glucosa en el plasma (mg/dl) a lo largo del tiempo. Los picos son las ingestas de carbohidratos que disminuyen al inyectar insulina o al realizar ejercicio (figura 14).
- **Figura 5:** se observan la Figura 1 y la Figura 4 para poder compararlas (figura 15).

Hay que tener en cuenta que el nombre de **Figura X** corresponde al título por el cual aparece cada gráfica tras la simulación en MATLAB. Siendo X un número entero entre el 1 y el 5, ambos incluidos.

A continuación se muestra un ejemplo de las gráficas al introducir los siguientes datos:

Hora del desayuno y el primer bolo: 8h	CHO en el desayuno: 20g
Hora de la comida y el segundo bolo: 14h	CHO en la comida: 60g
Hora de la cena y el tercer bolo: 21h	CHO en la cena: 40g
Bolo del desayuno: 2 U/min	Hora del ejercicio: 17h
Bolo de la comida: 6 U/min	Duración del ejercicio: 60 min
Bolo de la cena: 4 U/min	Peso: 70kg
Sensibilidad: 0.39	Insulina basal: 0.01667 U/min

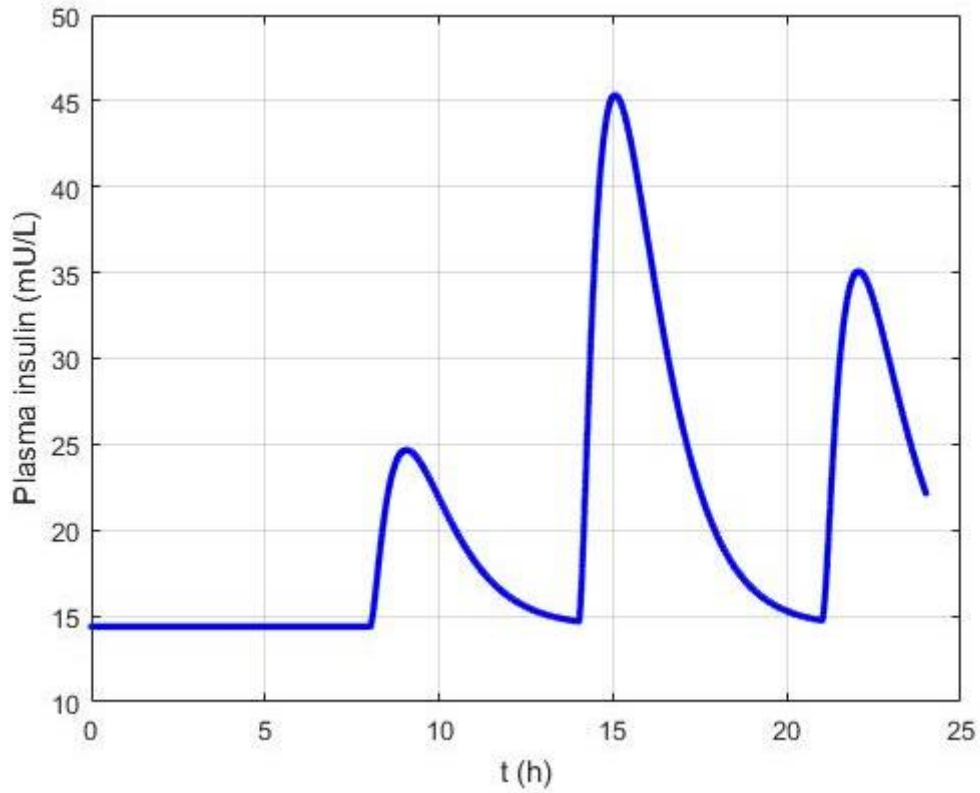


Figura 11. Evolución temporal de la insulina en el plasma

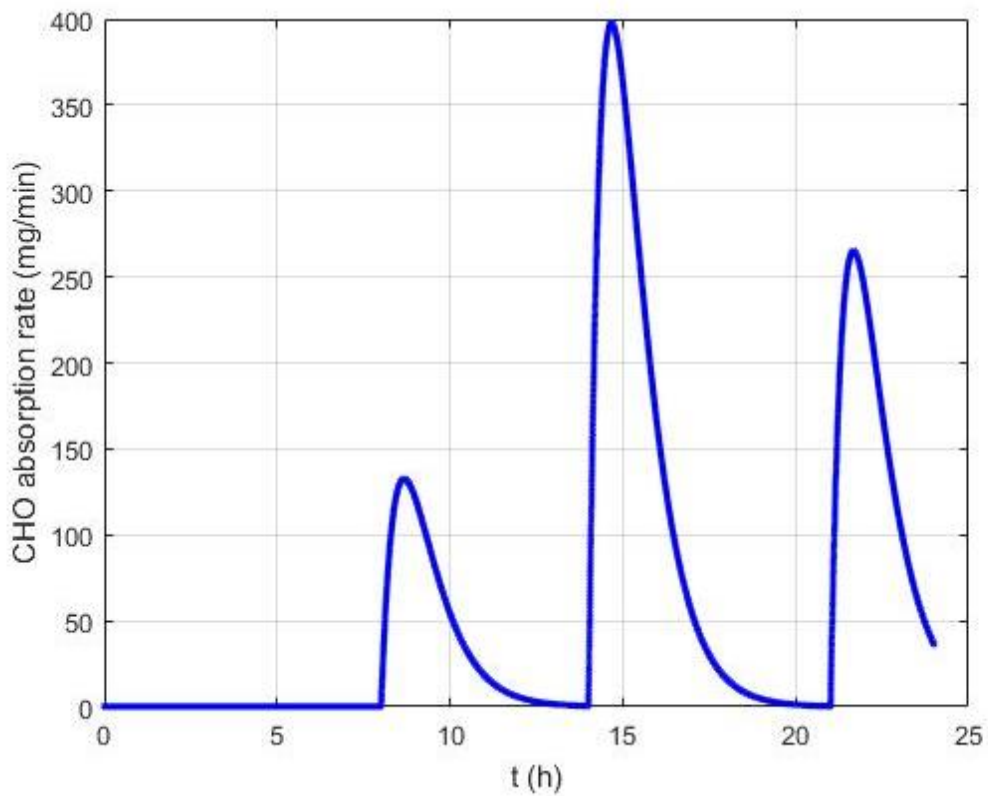


Figura 12. Evolución temporal de la absorción de carbohidratos

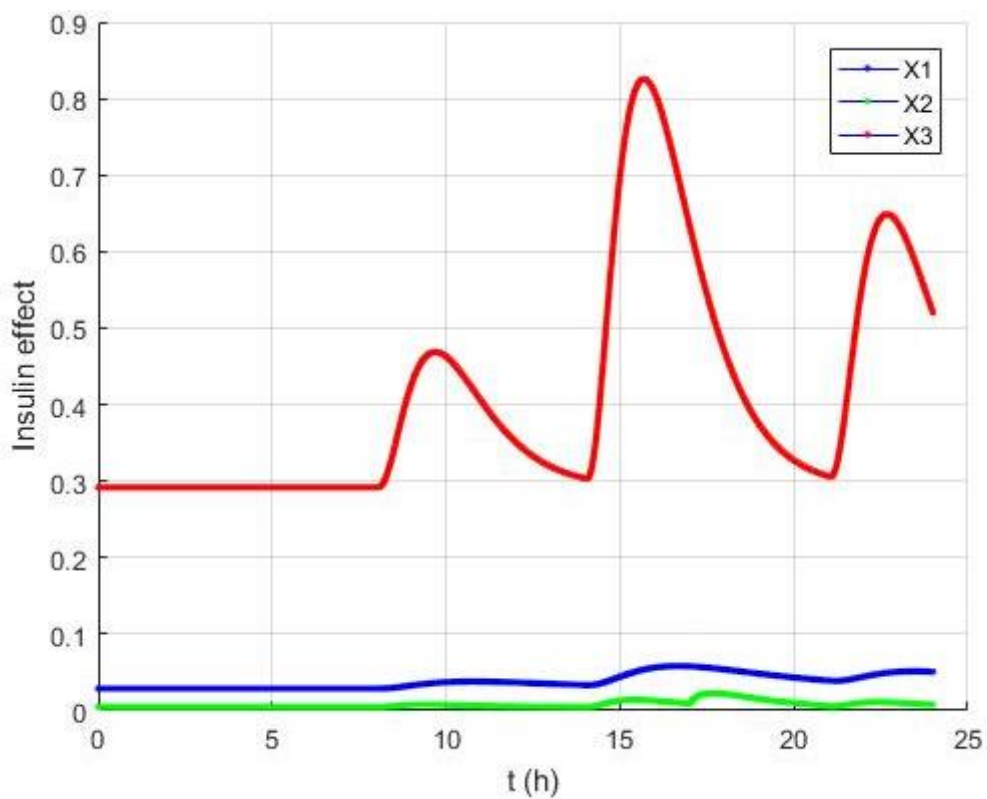


Figura 13. Evolución temporal del efecto de la insulina

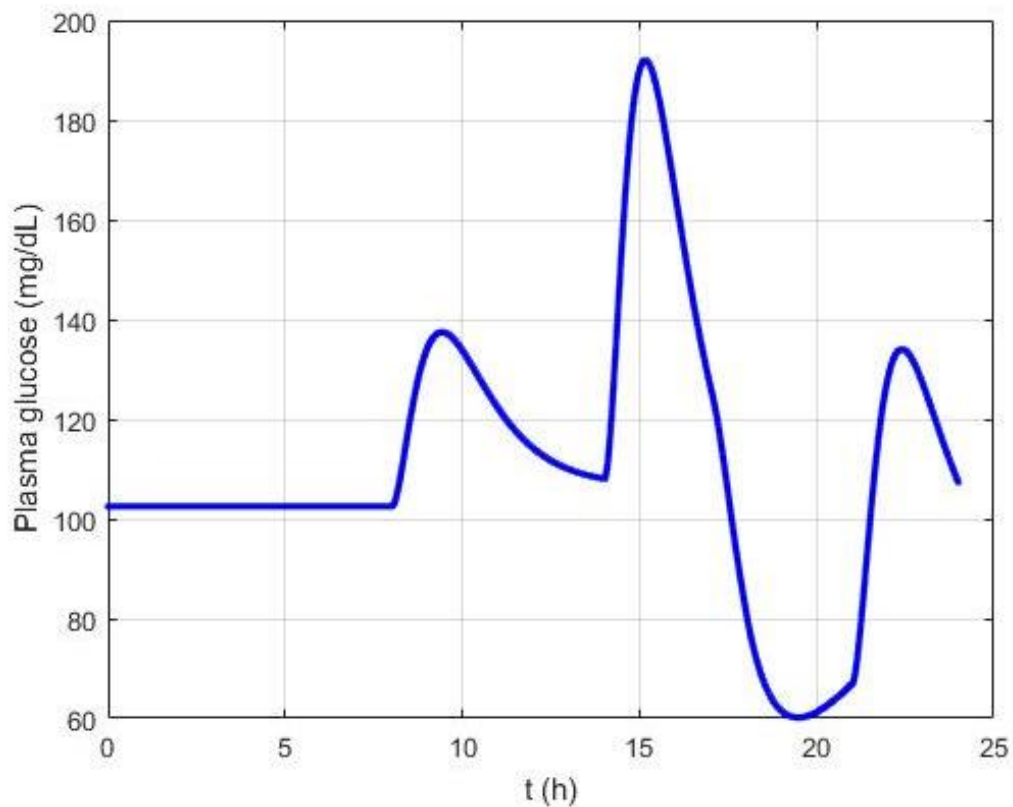


Figura 14. Evolución temporal de la glucosa en plasma

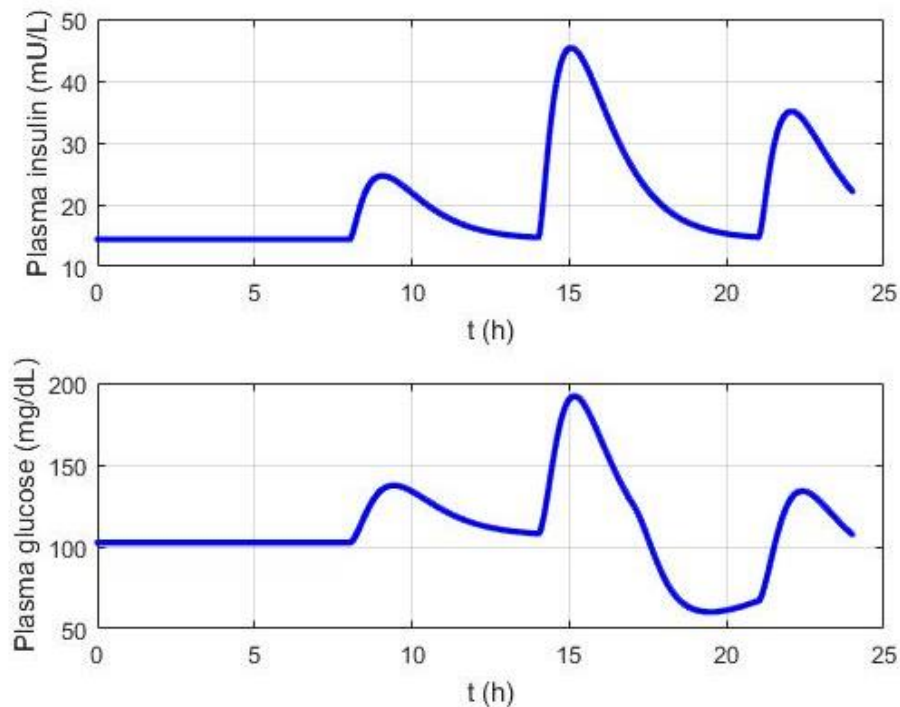


Figura 15. Comparación de la evolución temporal de la insulina y la glucosa en el plasma

Para que funcione la interfaz gráfica de MATLAB, se tuvo que transformar el script del simulador del que ya se disponía, en una función que permitiera recibir los valores de entrada de la interfaz.

6.2 Interfaz gráfica 3D

Una vez hayan sido introducidos los datos de entrada y se haya ejecutado la interfaz de GUIDE, empezará a funcionar Factory IO.

En dicho programa se visualizará el interior de una nave industrial en 3D. A modo de ejemplo, se puede apreciar la figura 16. Además, para que funcione correctamente, debido al código implementado, se deberá de acelerar la simulación al doble de velocidad. Para ello se presiona el botón señalado en la figura y se marca el número 2.

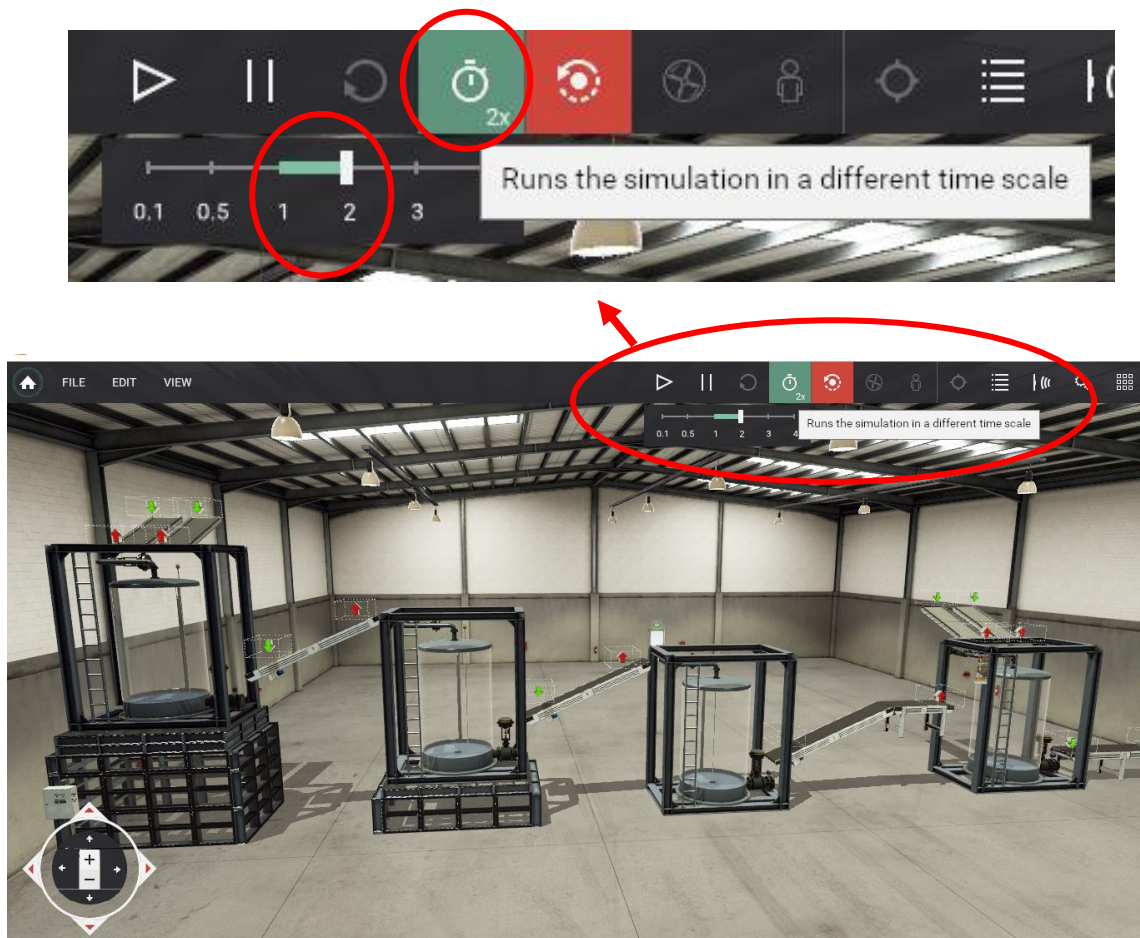


Figura 16. Interfaz gráfica 3D de la nave industrial

6.2.1 Elementos

Los elementos que componen la nave industrial de la aplicación son los siguientes:

- **Tanques:** a través del modelo de Cambridge, se sabe que desde que la insulina es inyectada en el cuerpo hasta que es consumida pasa por tres compartimentos diferentes. Por ello, se han utilizado tres tanques (los tres de la izquierda).

En el primero de ellos, el cual se llama compartimento de entrada, se introduce un caudal de agua que representa la insulina inyectada en el cuerpo. Cuando la insulina deba de pasar al siguiente compartimento, compartimento intermedio, el primer tanque se vaciará para poder llenar el segundo. Sucederá exactamente lo mismo con el tercer tanque, insulina en plasma, el cual representa la cantidad de insulina almacenada en el plasma.

Al salir de este último compartimento, el cuerpo ya es capaz de consumir la insulina. Si se aprecia una bajada del nivel de insulina, se observará más adelante una bajada del nivel

de glucosa (tanque de la derecha) ya que la insulina estará siendo consumida y hará que los niveles de glucosa disminuyan.

En la figura 17 se observa cómo se está inyectando insulina en un compartimento.



Figura 17. Tanque con la válvula de llenado abierta

Sus actuadores y sensores son los siguientes (figura 18):

Actuadores:

Válvulas de control: una de entrada del fluido y otra de salida.

Sensores:

Medidor de nivel.

Medidor de flujo.

El tanque tiene una altura de 3 m y un diámetro de 2 m. La tubería de descarga posee un radio de 0.125 m y los flujos de entrada y salida son como máximo 0.25 m³/s y 0.3543 m³/s respectivamente [16].

Clicando con el botón derecho del ratón en el tanque, a través de la configuración se pasa de digital a analógica. Como se deseó utilizar variables tipo float en los tanques, se debe de elegir la configuración analógica. De esta manera, las variables tendrán un rango entre 0 y 10 voltios.

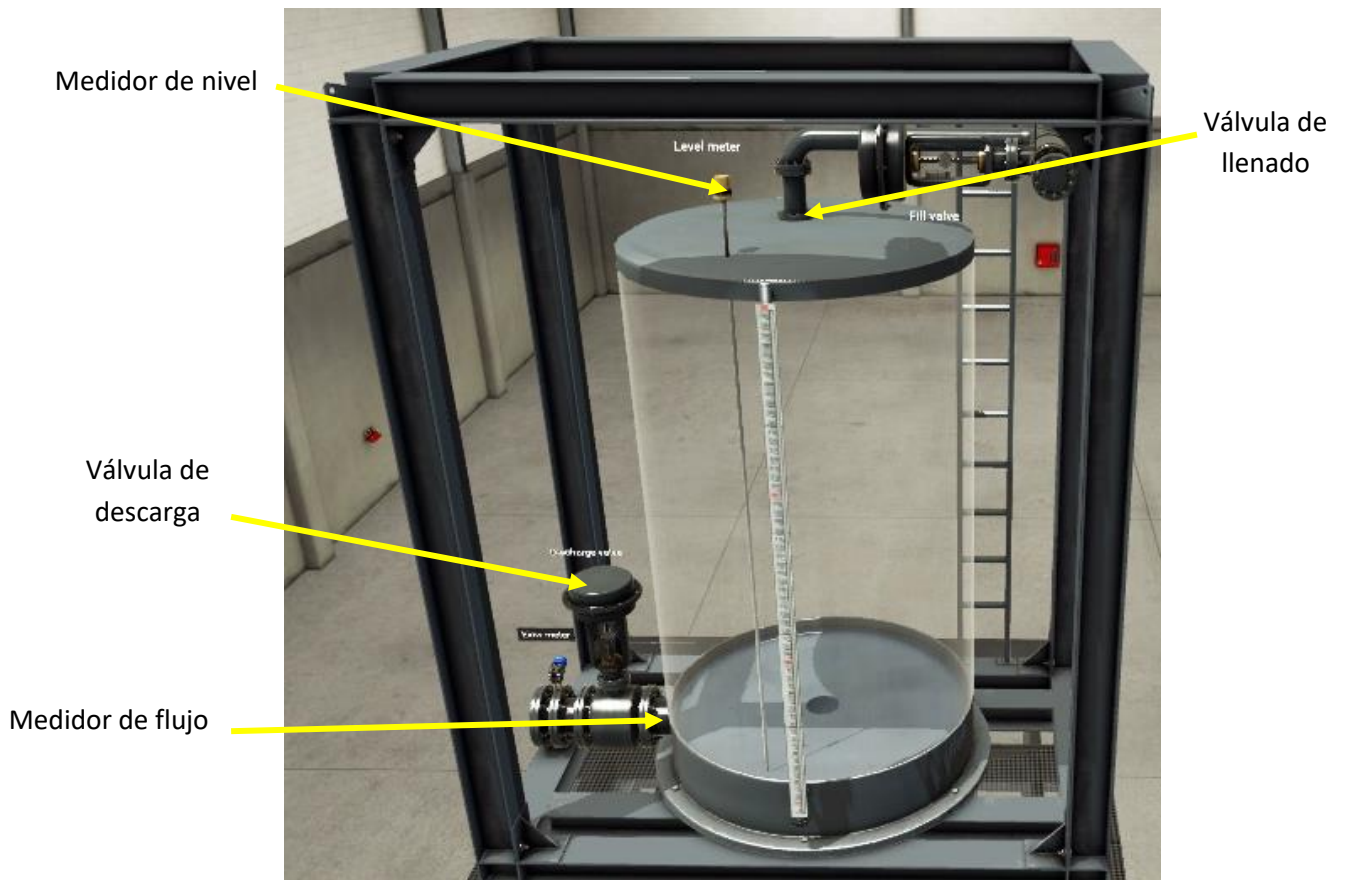


Figura 18. Tanque con los actuadores y sensores

- **Plataformas:** sirven para elevar los dos primeros tanques de insulina. Se usaron para que se viera de una manera más clara que del primer compartimento, la insulina pasa al intermedio y del intermedio, al tercero.

En un principio, se intentó colocar los tres tanques de insulina verticalmente, uno encima del otro, pero por limitaciones de espacio no fue posible ya que el techo de la nave industrial no era lo suficientemente alto.

En la figura 19 se pueden apreciar un conjunto de plataformas correspondientes al primer compartimento de insulina.



Figura 19. Plataformas

- **Cintas transportadoras:** comunican los tanques. Cuando la insulina tiene que pasar del primer compartimento al intermedio, la cinta transportadora situada entre ambos tanques se mueve y traslada cierta materia gris que hace referencia a la insulina. Al igual sucede entre el compartimento intermedio y la insulina en plasma. En cambio, entre la insulina en plasma y la glucosa hay una pequeña diferencia, la cinta transportadora llega a la mitad del tanque de la glucosa, esto es debido a que la insulina no le afecta instantáneamente.

También se puede apreciar una cinta transportadora que mueve palés a la salida del tanque de glucosa. Representa la glucosa consumida en el organismo.

En la figura 20 se puede ver una cinta inclinada y la materia gris que desplaza.

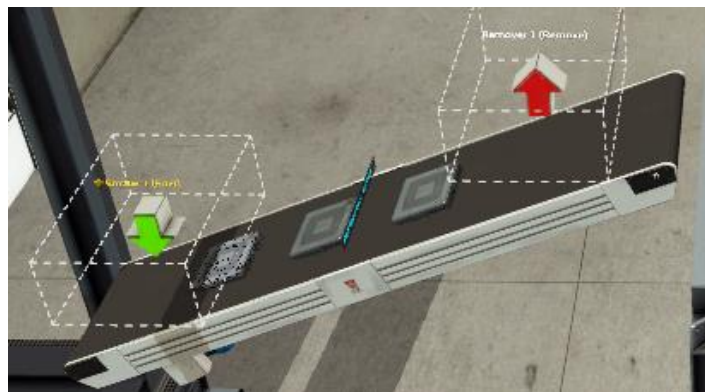


Figura 20. Cinta transportadora de materia gris

La cinta se utiliza a través de la configuración digital y solo tiene un actuador.

Actuador:

Cinta: 1 si funciona y 0 si se para.

- **Canal de transporte:** son las entradas del primer tanque de insulina y de la glucosa. En el tanque de insulina hay dos canales de transporte ya que uno envía la materia verde que representa la insulina basal y el otro, materia azul, que representa el bolo de insulina. Dichos canales se muestran en la figura 21.

Las dos entradas de la glucosa corresponden a los azúcares provenientes de la producción exógena (ingestas) y de la producción endógena (hígado). Ambas entradas son representadas con distintos tamaños de cajas para ser diferenciadas (figura 22).

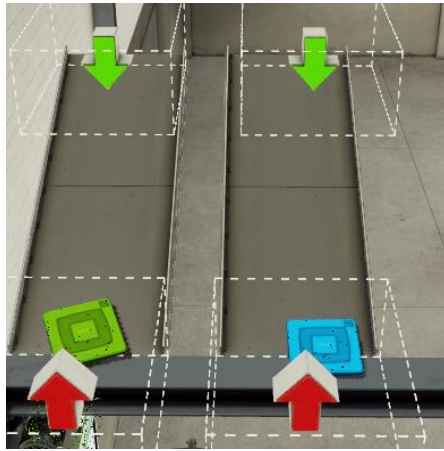


Figura 21. Canales de transporte de insulina

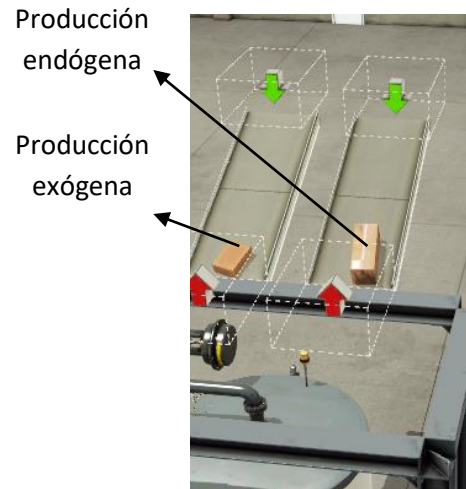


Figura 22. Canales de transporte de glucosa

- **Emisor:** emite la materia verde, azul, gris, las diferentes cajas y palés. Son las flechas verdes que aparecen en las figuras 20, 21 y 22.

Al clicar sobre los emisores con el botón derecho del ratón aparece la pestaña de la figura 23. En ella se puede elegir el máximo y mínimo tiempo que hay entre un objeto y el siguiente, la cantidad de objetos a emitir y qué objetos se desean visualizar.

Actuador:

Emisor: 1 emite y 0 no emite objeto.

- **Eliminador:** hace desaparecer lo que crea el emisor. Son las flechas rojas de las mismas figuras anteriores.

Actuador:

Eliminador: 1 elimina y 0 no elimina el objeto.

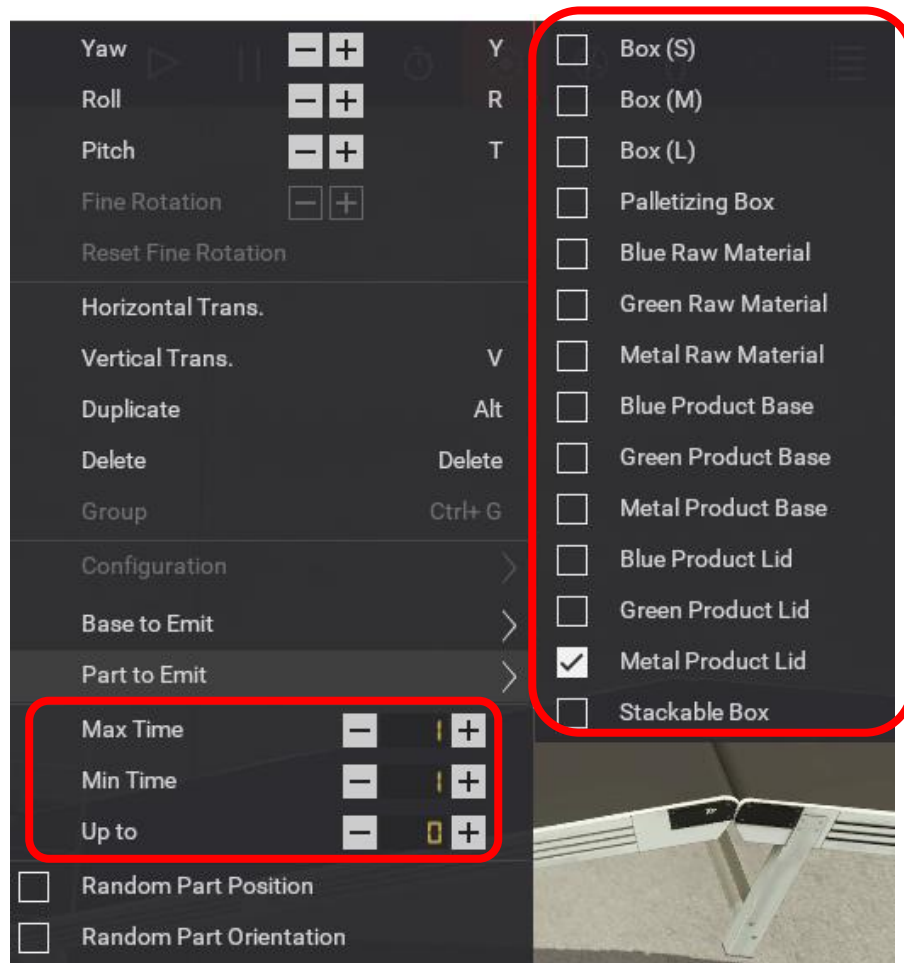


Figura 23. Pestaña del emisor

- **Luces y alarmas:** en la parte superior del tanque de la glucosa hay un total de 18 luces y 3 alarmas.

3 luces rojas en la parte superior: se encienden cuando está presente la hiperglucemia y el nivel de glucosa en el tanque supera los 250 mg/dl.

3 luces amarillas en la parte superior: se encienden cuando se detecta hiperglucemia y el nivel de glucosa en el tanque está entre los 180 mg/dl y los 250 mg/dl.

6 luces verdes en la parte intermedia: se encienden cuando hay normoglucemia, la glucosa está entre 70 mg/dl y 180 mg/dl.

3 luces amarillas en la parte inferior: se encienden cuando está presente la hipoglucemia y la glucosa está entre 54 mg/dl y 70 mg/dl.

3 luces rojas en la parte inferior: se encienden cuando hay hipoglucemia severa, la glucemia es inferior a 54 mg/dl. La hipoglucemia severa es la más peligrosa de todas, por ello a la vez que se encienden estas 3 luces rojas en la parte inferior, también se activan 3 alarmas.

Se pueden lo que se acaba de explicar en la figura 24.

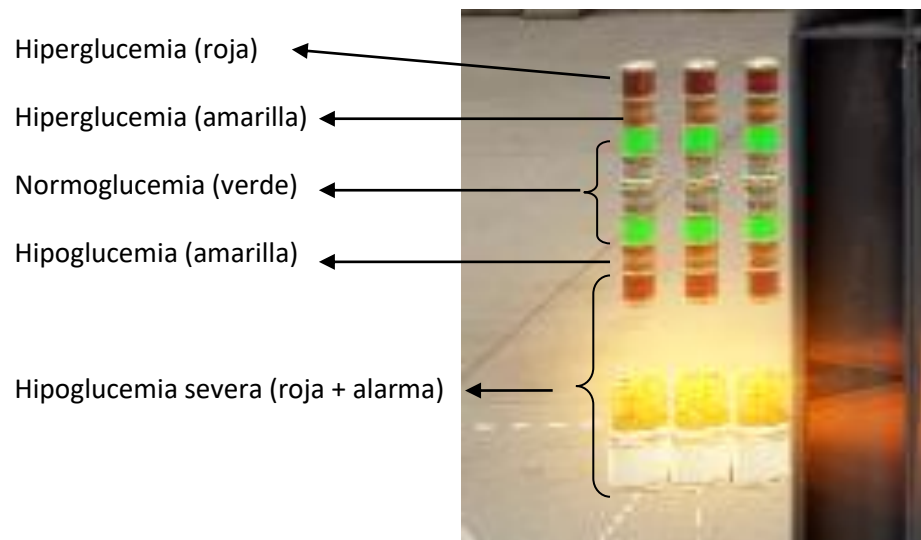


Figura 24. Luces y alarmas

Actuadores:

Luz roja, luz amarilla, luz verde, alarma: un 1 es para que se activen y un 0 para que se desactiven.

El usuario podrá crear un escenario y a través de la interfaz gráfica en 3D será capaz de entender la insulina que tiene en cada compartimento del organismo y además, podrá saber sus niveles de glucosa. Las luces y alarmas le ayudarán a interpretar dicho nivel.

Podrá desplazarse a través de la interfaz gracias a la cámara de navegación que se encuentra en la parte inferior izquierda, dicha cámara puede verse en la figura 25.



Figura 25. Cámara de navegación

6.2.3 Creación de la escena

Para crear la escena de la figura 16 se deben de realizar los siguientes pasos:

1. Crear una nueva escena.
2. Seleccionar cada elemento desde la Palette Window y colocarlo en el sitio deseado, figura 26.

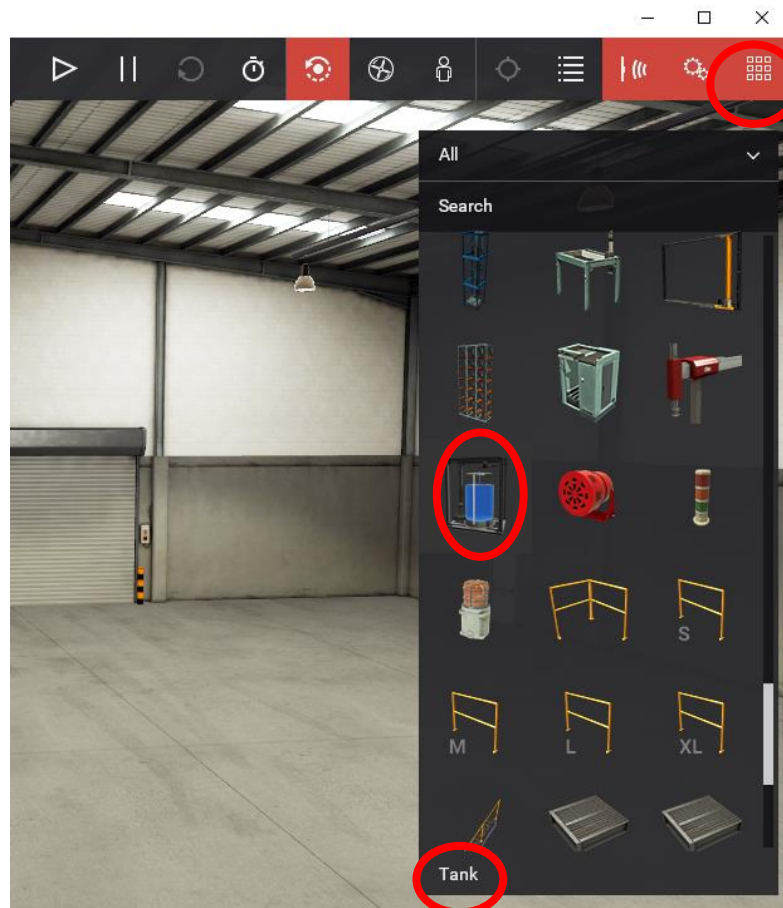


Figura 26. Elección de elementos

3. Una vez se haya colocado el elemento en el sitio deseado, se hace un clic con el botón derecho sobre el elemento para configurar las siguientes características:
 - **Tanques:** Configuración analógica.
 - **Cintas transportadoras:** Configuración digital.
 - **Emisor:**
Max Time=1 Min Time=1 Up to=0

4. Se deben de mostrar los sensores y actuadores clicando arriba a la derecha en Sensors Tags y Actuators Tags.
5. Una vez se muestren por pantalla, si se clicla en ellos debe de verse que están en modo de no fallo (no failure) tal y como aparece en la figura 27.



Figura 27. Modo de no fallo

6.3 Ecuaciones para la equivalencia entre modelos dinámicos

La acción más complicada de realizar fue la de introducir y sacar el caudal deseado de los tanques. A continuación, se van a mostrar las ecuaciones de los caudales correspondientes a cada tanque en diferentes unidades.

Tanque 1: Compartimento de entrada de insulina.

Tanque 2: Compartimento intermedio de insulina.

Tanque 3: Insulina en plasma.

Tanque 4: Nivel de glucosa.

	$\frac{U}{s}$	$\frac{m^3}{s}$	V
Q_{1entra}	$I_{Basal} \left(\frac{U}{s}\right) + I_{bolo} \left(\frac{U}{s}\right)$	$\frac{Q_{1entra} \left(\frac{U}{s}\right) * 0.25 \left(\frac{m^3}{s}\right)}{I_{Max} \left(\frac{U}{s}\right)}$	$\frac{Q_{1entra} \left(\frac{m^3}{s}\right) * 10 (V)}{0.25 \left(\frac{m^3}{s}\right)}$
Q_{2entra}	$\frac{S_1}{t_{maxi} * 60}$	$\frac{Q_{2entra} \left(\frac{U}{s}\right) * 0.25 \left(\frac{m^3}{s}\right)}{I_{Max} \left(\frac{U}{s}\right)}$	$\frac{Q_{2entra} \left(\frac{m^3}{s}\right) * 10 (V)}{0.25 \left(\frac{m^3}{s}\right)}$
Q_{3entra}	$\frac{S1}{t_{maxi} * Vi * 60}$	$\frac{Q_{3entra} \left(\frac{U}{s}\right) * 0.25 \left(\frac{m^3}{s}\right)}{I_{Max} \left(\frac{U}{s}\right)}$	$\frac{Q_{3entra} \left(\frac{m^3}{s}\right) * 10 (V)}{0.25 \left(\frac{m^3}{s}\right)}$

Tabla 3. Caudales de entrada de la insulina

Q_{1sale}	$\frac{S_1}{t_{maxI} * 60}$	$\frac{Q_{1sale} \left(\frac{U}{s}\right) * 0.25 \left(\frac{m^3}{s}\right)}{IMax \left(\frac{U}{s}\right)}$	$\frac{Q_{1sale} \left(\frac{m^3}{s}\right) * 10 (V)}{0.3543 \left(\frac{m^3}{s}\right)}$
Q_{2sale}	$\frac{S_2}{t_{maxI} * 60}$	$\frac{Q_{2sale} \left(\frac{U}{s}\right) * 0.25 \left(\frac{m^3}{s}\right)}{IMax \left(\frac{U}{s}\right)}$	$\frac{Q_{2sale} \left(\frac{m^3}{s}\right) * 10 (V)}{0.3543 \left(\frac{m^3}{s}\right)}$
Q_{3sale}	$\frac{ke * I}{60}$	$\frac{Q_{3sale} \left(\frac{U}{s}\right) * 0.25 \left(\frac{m^3}{s}\right)}{IMax \left(\frac{U}{s}\right)}$	$\frac{Q_{3sale} \left(\frac{m^3}{s}\right) * 10 (V)}{0.3543 \left(\frac{m^3}{s}\right)}$

Tabla 5. Caudales de salida de la insulina

	$\frac{mmol}{L * s}$	$\frac{mg}{dl * s}$	$\frac{m^3}{s}$	v
Q_{4entra}	$\frac{k_{12} * Q_2 + Ug + EGP_0}{Vg * 60}$	$Q_{4entra} \left(\frac{mmol}{L} * s\right) * 18$	$\frac{Q_{4entra} \left(\frac{mg}{dl * s}\right) * 0.25 \left(\frac{m^3}{s}\right)}{MaxG \left(\frac{mg}{dl * s}\right)}$	$\frac{Q_{4entra} \left(\frac{m^3}{s}\right) * 10 (V)}{0.25 \left(\frac{m^3}{s}\right)}$
Q_{4sale}	$\frac{F_{01}^C + x_1 * Q_1 + Fr + EGP_0 * x_3}{Vg * 60}$	$Q_{4sale} \left(\frac{mmol}{L} * s\right) * 18$	$\frac{Q_{4sale} \left(\frac{mg}{dl * s}\right) * 0.25 \left(\frac{m^3}{s}\right)}{MaxG \left(\frac{mg}{dl * s}\right)}$	$\frac{Q_{4sale} \left(\frac{m^3}{s}\right) * 10 (V)}{0.3543 \left(\frac{m^3}{s}\right)}$

Tabla 4. Caudales de entrada y salida de la glucosa

Si se calculan los caudales de entrada en voltios, como se corresponden a las aperturas de las válvulas, ya se tienen las variables a introducir en Factory IO. En cambio, para los caudales de salida debe de realizarse una operación extra. Esto es debido a que el caudal calculado no es igual a la apertura de la válvula porque el caudal real depende del nivel del líquido que hay en el tanque .

Por lo tanto, la apertura de la válvula de salida se puede calcular de la siguiente manera:

$$x = \frac{Q_{calculado} (V)}{3.1484 * \sqrt{H} (V)} \quad (7.1)$$

Esta ecuación dará el nivel en tanto por uno. Debe de multiplicarse por 10 para que salga un valor entre 0 y 10.

En el Manual del programador del Anexo IV, concretamente en el apartado de Caudales de entrada y salida de los tanques, están desarrolladas todas las fórmulas.

6.4 Algoritmos de intercambio de información y comunicaciones

Para que Factory IO represente de manera adecuada lo explicado anteriormente se programaron ciertos algoritmos de intercambio de datos y se realizó la comunicación entre los distintos programas.

6.4.1 Algoritmos de intercambio de información

En la figura 28 se muestran los archivos de MATLAB implementados para este proyecto. Las flechas señalan la dirección del flujo de información.

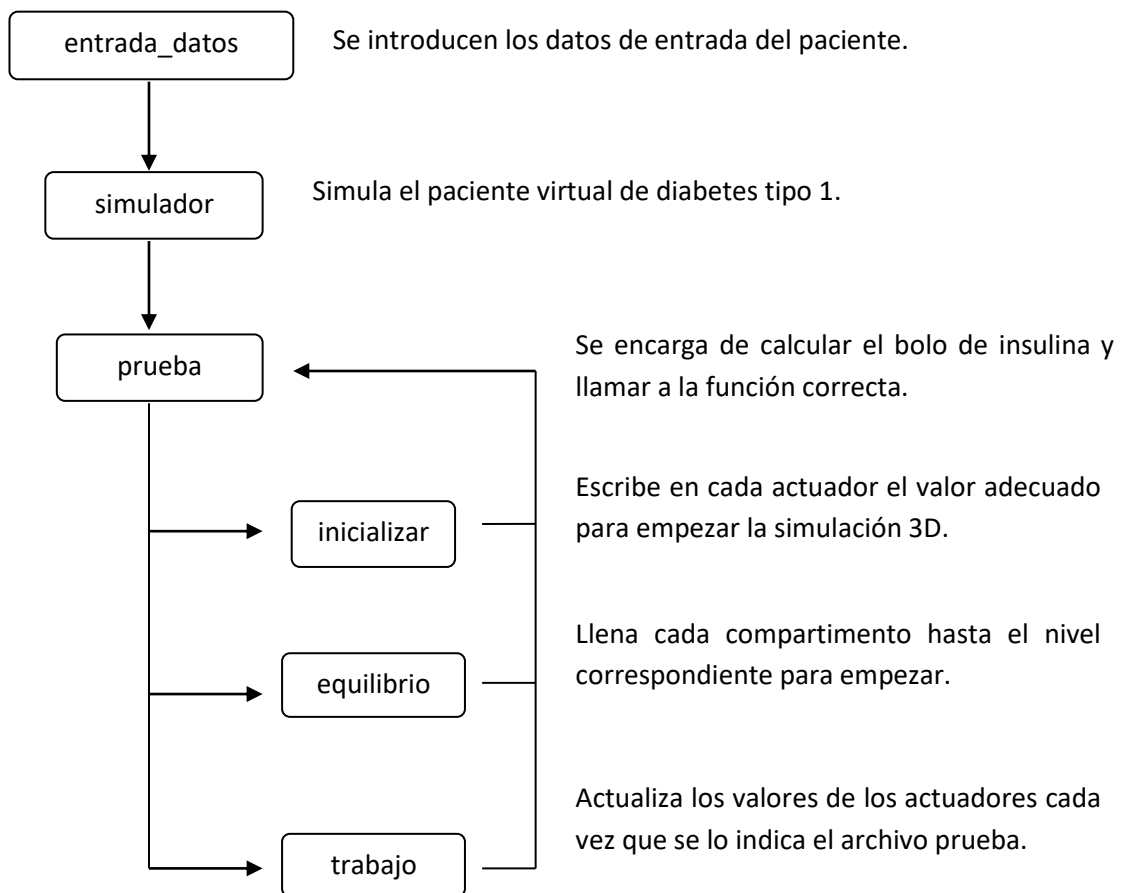


Figura 28. Archivos de MATLAB

6.4.2 Comunicaciones

La comunicación entre MATLAB y Factory IO se realizó a través de una comunicación OPC como se ha explicado anteriormente. Para un buen funcionamiento, en cada programa se debe de hacer lo siguiente:

- KEPServer

Los pasos a seguir son:

1. **Crear el canal:** clicar con el botón derecho del ratón en: Click to add a channel y seleccionar New Channel. A continuación, se abrirá una pestaña donde se puede cambiar el nombre del canal. En este proyecto se optó por el nombre Channel1. Dicho procedimiento puede verse en la figura 29.

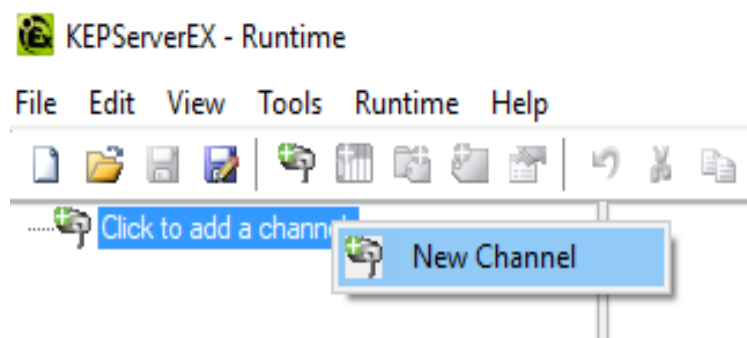


Figura 29. Creación del canal

2. **Crear el dispositivo y el grupo de etiquetas:** se realiza de la misma manera que el canal. El ejemplo está en la figura 30.

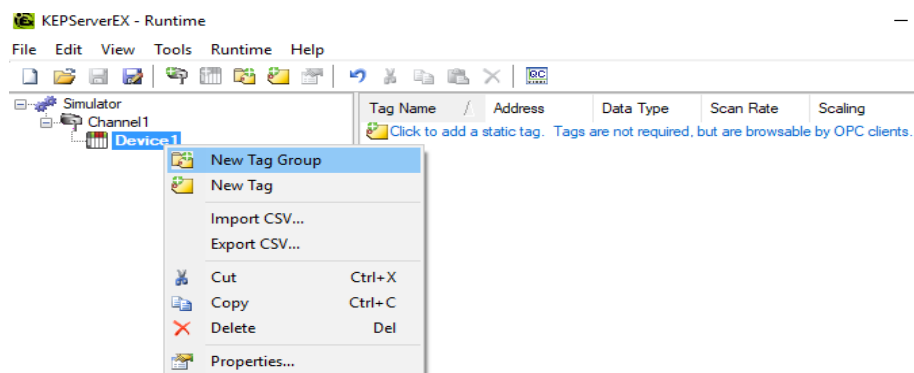


Figura 30. Creación del dispositivo y el grupo de etiquetas

3. **Crear las etiquetas:** en la pantalla de la derecha con el botón derecho del ratón se generan los New Tags, donde estarán las variables necesarias, figura 31.

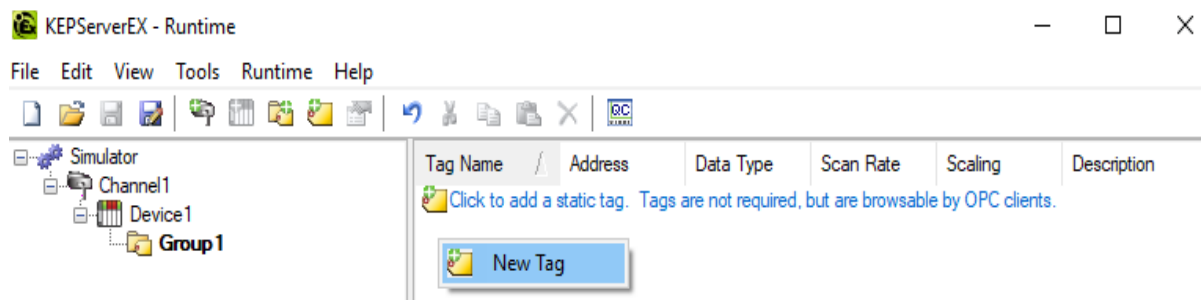


Figura 31. Creación de las etiquetas

4. **Rellenar los datos de las variables:** se va a explicar cómo se tienen que completar utilizando un ejemplo. La variable que se va a crear a continuación va a ser llenar1, que es la que llena el tanque 1, la correspondiente al primer compartimento de la insulina.

- Nombre: llenar1
- Dirección: al clicar en el interrogante de la figura 32, se despliega una pestaña en la que se puede elegir el tipo de variable a crear. En este caso, la variable llenar1 debe de ser float. A través de esa pestaña, se sabe cuál es la forma de la dirección de cada tipo de variable, por lo que bastará con introducir el principio de la dirección y un número final ya que al clicar en el tic verde se autocorregirá la dirección a una válida, se muestra un ejemplo en la figura 33.

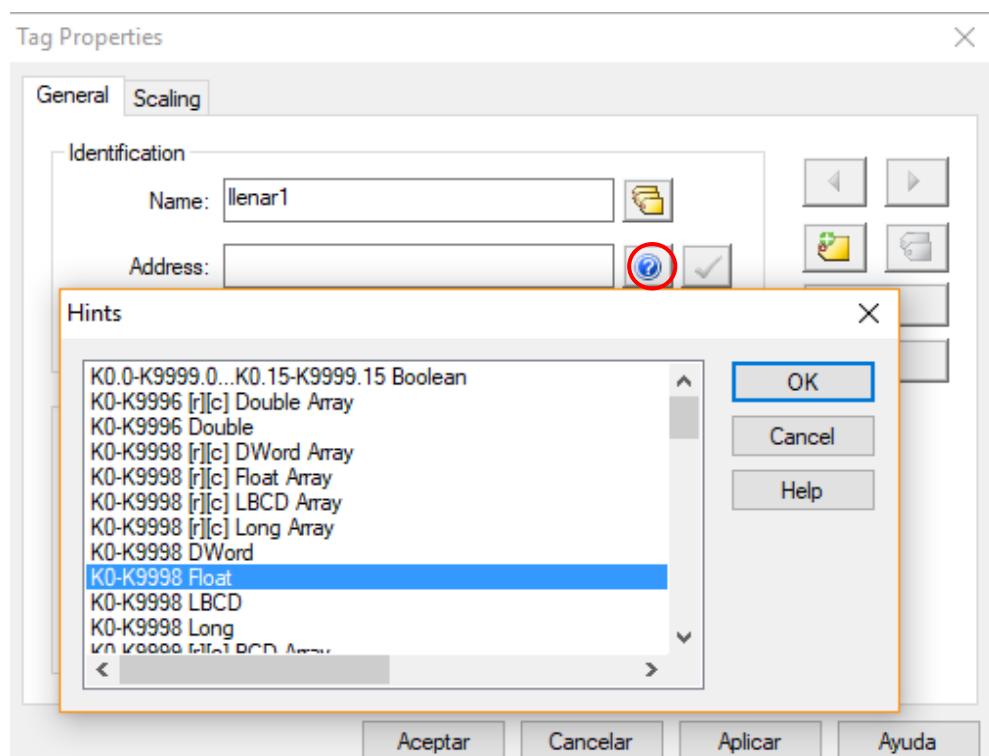


Figura 32. Pestaña de direcciones

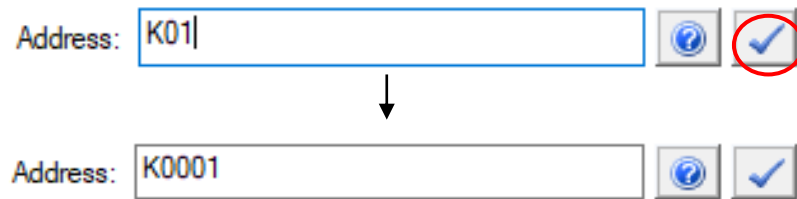


Figura 33. Corrección automática de las direcciones

5. La pestaña de la etiqueta quedará como muestra la figura 34.

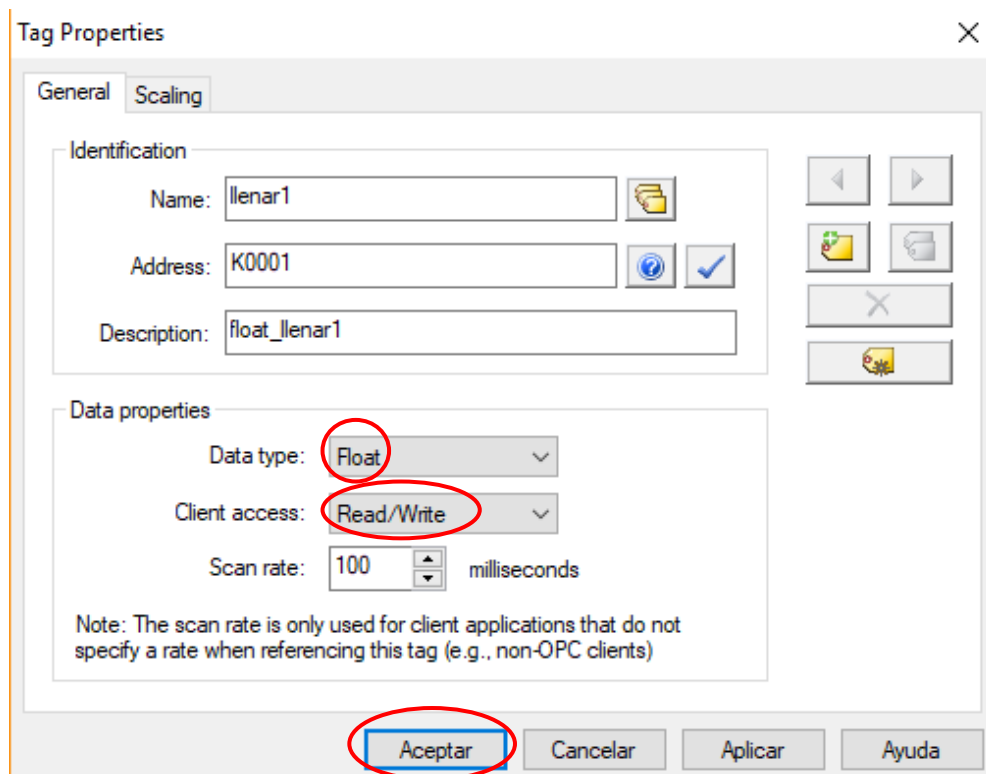


Figura 34. Pestaña de direcciones completada

6. Una vez creadas todas las variables quedará una lista como en la figura 35.
7. Clicar sobre Tools y a continuación, sobre Launch OPC Quick Client (figura 36). Aparecerá una pestaña como en la figura 37 pero con todas las variables creadas, ahí se pueden observar también los valores que toman.

DISEÑO E IMPLEMENTACIÓN DE UNA HERRAMIENTA EDUCATIVA SOBRE LA GESTIÓN DE DIABETES TIPO 1 EMPLEANDO HERRAMIENTAS DE SIMULACIÓN 3D DE PROCESOS INDUSTRIALES

Tag Name	Address	Data Type			
aluma1	K0800.02	Boolean	luzr1	K0200.07	Boolean
aluma2	K0800.02	Boolean	luzr2	K0200.07	Boolean
aluma3	K0800.02	Boolean	luzr3	K0200.07	Boolean
caudal1	K0004	Float	luzr4	K0200.10	Boolean
caudal2	K0005	Float	luzr5	K0200.10	Boolean
caudal3	K0006	Float	luzr6	K0200.10	Boolean
caudal4	K0007	Float	luzv1	K0200.13	Boolean
cinta1	K0000.08	Boolean	luzv2	K0200.13	Boolean
cinta2	K0000.08	Boolean	luzv3	K0200.13	Boolean
cinta3	K0000.08	Boolean	luzv4	K0300.01	Boolean
cintaA	K0000.08	Boolean	luzv5	K0300.01	Boolean
cintaB	K0000.08	Boolean	luzv6	K0300.01	Boolean
eliminar1	K0000.03	Boolean	nivel1	K0111	Float
eliminar2	K0000.03	Boolean	nivel2	K0222	Float
eliminar3	K0000.03	Boolean	nivel3	K0355	Float
eliminar4	K0000.03	Boolean	nivel4	K0334	Float
eliminar5	K0000.03	Boolean	producir1	K0000.11	Boolean
eliminar6	K0000.03	Boolean	producir2	K0000.12	Boolean
eliminar7	K0000.03	Boolean	producir3	K0000.13	Boolean
eliminar8	K0000.03	Boolean	producir4	K0000.14	Boolean
llenar1	K0120	Float	producir5	K0000.15	Boolean
llenar2	K0100	Float	producir6	K0000.10	Boolean
llenar3	K0444	Float	producir7	K0000.09	Boolean
llenar4	K0134	Float	producir8	K0001.02	Boolean
luza1	K0200.01	Boolean	reset	K0000.02	Boolean
luza2	K0200.01	Boolean	run	K0000.01	Boolean
luza3	K0200.01	Boolean	vaciar1	K0011	Float
luza4	K0200.04	Boolean	vaciar2	K0022	Float
luza5	K0200.04	Boolean	vaciar3	K0033	Float
luza6	K0200.04	Boolean	vaciar4	K0044	Float
luzr1	K0200.07	Boolean			
luzr2	K0200.07	Boolean			

Figura 35. Variables

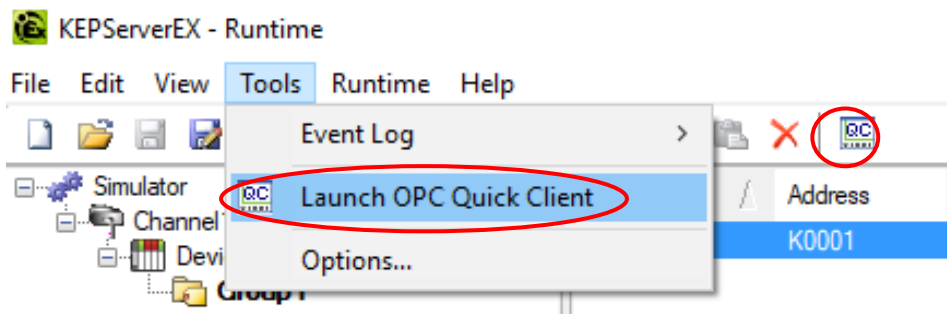


Figura 36. Abrir OPC

Item ID	Data Type	Value	Timestamp	Quality	Update
Channel1.Device1.Group1.alama1	Boolean	0	19:19:50.296	Good	1
Channel1.Device1.Group1.alama2	Boolean	0	19:19:50.296	Good	1
Channel1.Device1.Group1.alama3	Boolean	0	19:19:50.296	Good	1
Channel1.Device1.Group1.caudal1	Float	0	19:19:50.296	Good	1
Channel1.Device1.Group1.caudal2	Float	0	19:19:50.296	Good	1
Channel1.Device1.Group1.caudal3	Float	0	19:19:50.296	Good	1

Figura 37. OPC con todas las variables creadas

- Factory IO

Para que KEPServer pueda escribir y leer las variables de Factory IO, se realizaron los siguientes pasos:

1. Abrir la escena en la cual se está trabajando.
2. Pulsar la tecla f4 del ordenador.
3. Buscar OPC Client DA/UA y clicar en CONFIGURATION, como se muestra en la figura 38.
4. Se abrirá una pestaña (figura 39) que pedirá lo siguiente:
 - **Machine Name:** localhost. Clicar sobre BROWSE SERVERS.
 - **OPC Server:** Kepware.KEPServerEx. (Seleccionar de la lista desplegable).
 - **Limit:** 100 (máximas variables que se pueden introducir). Para este proyecto se han necesitado 60.
 - **Filter names that contains:** Group1. Clicar en BROWSE.

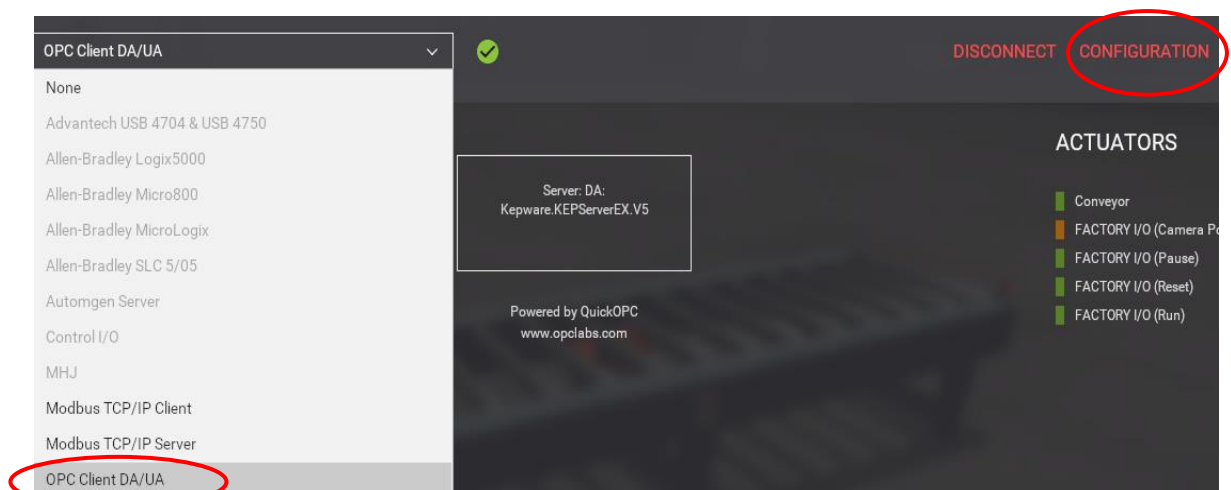


Figura 38. OPC Client DA/UA en Factory IO

The image shows a dark-themed configuration window for OPC. It is divided into two main sections: 'Server' and 'Items/Nodes'.
In the 'Server' section, there is a 'Machine Name' field with a warning icon, a 'BROWSE SERVERS' button, and an 'OPC Server' dropdown menu currently showing 'DA: Kepware.KEPServerEX.V5'.
The 'Items/Nodes' section contains a 'BROWSE' button, a 'Limit' field with the value '100', a 'Filter names that start with:' field with 'No filter', and a 'Filter names that contain:' field with 'Group1'.
At the bottom, there is a 'From Device' checkbox which is checked, accompanied by a warning icon.

Figura 39. Configuración del OPC

5. Al salir de esa pestaña, se deberán de unir los actuadores y sensores de Factory IO a las variables creadas en KEPServer como aparece en las figuras 40, 41 y 42.

Server: DA: Kepware.KEPServerEX.V5 (60)		
	alarma1	Warning Light 1
	alarma2	Warning Light 2
	alarma3	Warning Light 3
Flow meter	caudal1	
Tank 2 (Flow Meter)	caudal2	
Tank 3 (Flow Meter)	caudal3	
Tank 4 (Flow Meter)	caudal4	
	cinta1	Belt Conveyor Inclined 1
	cinta2	Belt Conveyor Inclined 2
	cinta3	Belt Conveyor Inclined 3
	cintaA	Belt Conveyor (2m) 1
	cintaB	Belt Conveyor (4m) 1
	eliminar1	Remover 1 (Remove)
	eliminar2	Remover 2 (Remove)
	eliminar3	Remover 3 (Remove)
	eliminar4	Remover 4 (Remove)
	eliminar5	Remover 5 (Remove)
	eliminar6	Remover 6 (Remove)
	eliminar7	Remover 7 (Remove)
	eliminar8	Remover 8 (Remove)
	llenar1	Fill valve
	llenar2	Tank 2 (Fill Valve)
	llenar3	Tank 3 (Fill Valve)

Figura 40. Actuadores y sensores parte 1

		llenar4	Tank 4 (Fill Valve)
		luza1	Stack Light 1 (Yellow)
		luza2	Stack Light 2 (Yellow)
		luza3	Stack Light 3 (Yellow)
		luza4	Stack Light 4 (Yellow)
		luza5	Stack Light 5 (Yellow)
		luza6	Stack Light 6 (Yellow)
		luzr1	Stack Light 1 (Red)
		luzr2	Stack Light 2 (Red)
		luzr3	Stack Light 3 (Red)
		luzr4	Stack Light 4 (Red)
		luzr5	Stack Light 5 (Red)
		luzr6	Stack Light 6 (Red)
		luzv1	Stack Light 1 (Green)
		luzv2	Stack Light 2 (Green)
		luzv3	Stack Light 3 (Green)
		luzv4	Stack Light 4 (Green)
		luzv5	Stack Light 5 (Green)
		luzv6	Stack Light 6 (Green)
Level meter		nivel1	
Tank 2 (Level Meter)		nivel2	
Tank 3 (Level Meter)		nivel3	
Tank 4 (Level Meter)		nivel4	
		producir1	Emitter 1 (Emit)
		producir2	Emitter 2 (Emit)

Figura 41. Actuadores y sensores parte 2

	producir3	Emitter 3 (Emit)
	producir4	Emitter 4 (Emit)
	producir5	Emitter 5 (Emit)
	producir6	Emitter 6 (Emit)
	producir7	Emitter 7 (Emit)
	producir8	Emitter 8 (Emit)
	reset	FACTORY I/O (Reset)
	run	FACTORY I/O (Run)
	vaciar1	Discharge valve
	vaciar2	Tank 2 (Discharge Valve)
	vaciar3	Tank 3 (Discharge Valve)
	vaciar4	Tank 4 (Discharge Valve)

Figura 42. Actuadores y sensores parte 3

- MATLAB

Para poder escribir y leer desde MATLAB a KEPServer se crearon dos funciones, una de escritura y otra de lectura, ambas pueden verse en las figuras 43 y 44.

```
1 function escribir(variable,valor)
2
3     da=opcda('localhost', 'keeware.KEPServerEX.V5');
4     connect (da);
5     grp=addgroup(da, 'Group1');
6
7     itmIDs={variable};
8     itm=additem(grp,itmIDs);
9
10    write (grp,valor)
11    disconnect (da);
12    clear da grp;
13 end
```

Figura 43. Función de escritura

```
1 function [lectura ] = leer(variable)
2
3 - da=opcda('localhost', 'keeware.KEPServerEX.V5');
4 - connect (da);
5 - grp=addgroup(da, 'Group1');
6
7 - itmIDs={variable};
8 - itm=additem(grp, itmIDs);
9 - data=read(grp);
10 - lectura=data.Value;
11 - disconnect (da);
12 - clear da grp;
13 - end
```

Figura 44. Función de lectura

Para usar la función escribir se debe de pasar la variable a la que se desea cambiar su valor y el nuevo valor a escribir. Por ejemplo, si se desea escribir un 10 en la variable llenar1, se debe de hacer lo siguiente:

```
escribir ('Channel1.Device1.Group1.llenar1',10)
```

En cambio, en la función leer, solo se necesita pasarle la variable de la cual se desea su lectura. Por ejemplo, si se desea leer la variable nivel1, se escribirá lo siguiente:

```
[nivel1]=leer ('Channel1.Device1.Group1.nivel1');
```

```
while (isempty (nivel1) ==1)
    [nivel1]=leer ('Channel1.Device1.Group1.nivel1');
end
```

El bucle while se utiliza por si hay problemas de lectura y la función leer devuelve una variable vacía. De esta manera, si no devuelve ningún valor se volverá a leer hasta encontrar un valor.

Si se desea obtener más información sobre el OPC se puede escribir en MATLAB lo siguiente: `opcserverinfo ('localhost')`.

Capítulo 7

Resultados

En este apartado se van a mostrar los distintos escenarios que fueron creados y sus resultados.

El escenario 1 tiene la siguiente información:

- Interfaz de entrada de datos
- Gráficas del simulador
- Equilibrio de la insulina y la glucosa del paciente al despertarse por la mañana
- Inyección del bolo a la hora especificada
- Bolo completamente administrado
- Niveles de glucosa
- Vídeo de la aplicación

En los demás escenarios solamente se muestra la información más relevante:

- Gráficas del simulador
- Inyección del bolo a la hora especificada
- Niveles de glucosa
- Vídeo de la aplicación

Todos los escenarios fueron creados para una persona estándar de 70 kg, con una sensibilidad a la insulina del 39% y cuya insulina basal es de 0.016671 U/min.

- **Escenario 1:**
Desayuno a las 8h con una ingesta de 30g de carbohidratos.
Bolo a las 8h con 3 U/min.

The image shows a software interface for managing scenarios. At the top, there is a header box labeled "NOMBRE DEL NUEVO ESCENARIO". Below it is a text input field containing the text "Escenario 1". Underneath the input field is a button labeled "GUARDAR ESCENARIO". Below that is a dropdown menu with "Escenario 1" selected. At the bottom of the interface are four buttons: "MOSTRAR ESCENARIOS", "MOSTRAR VARIABLES DEL ESCENARIO", and "ELIMINAR ESCENARIO".

Figura 45. Interfaz gráfica con el nombre del escenario 1

PERSONA CON DIABETES TIPO1

	Hora [0-23]	Minutos [0-59]	CHO (g)	Hora [0-23]	Minutos [0-59]	Bolo (U/min)
DESAYUNO	8	0	30	8	0	3
COMIDA	0	0	0	0	0	0
CENA	0	0	0	0	0	0

	Hora [0-23]	Minutos [0-59]	Duración (min)	Peso (Kg)	Sensibilidad [0-1]	Basal (U/min)
EJERCICIO	0	0	0	70	0.39	0.016671

La simulación empezará en breves.

RESTAURAR ÚLTIMOS DATOS ELIMINAR VARIABLES **START**

Figura 46. Interfaz gráfica con los datos de entrada del escenario 1

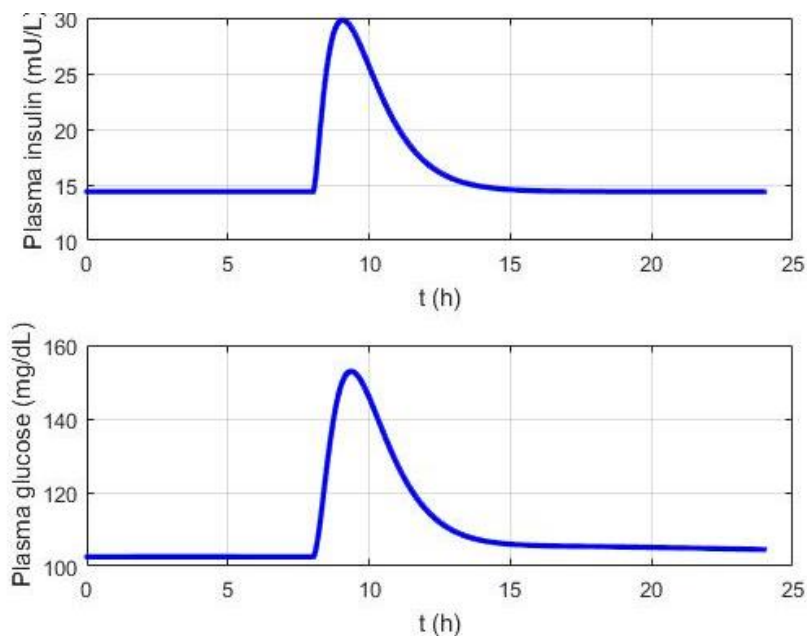
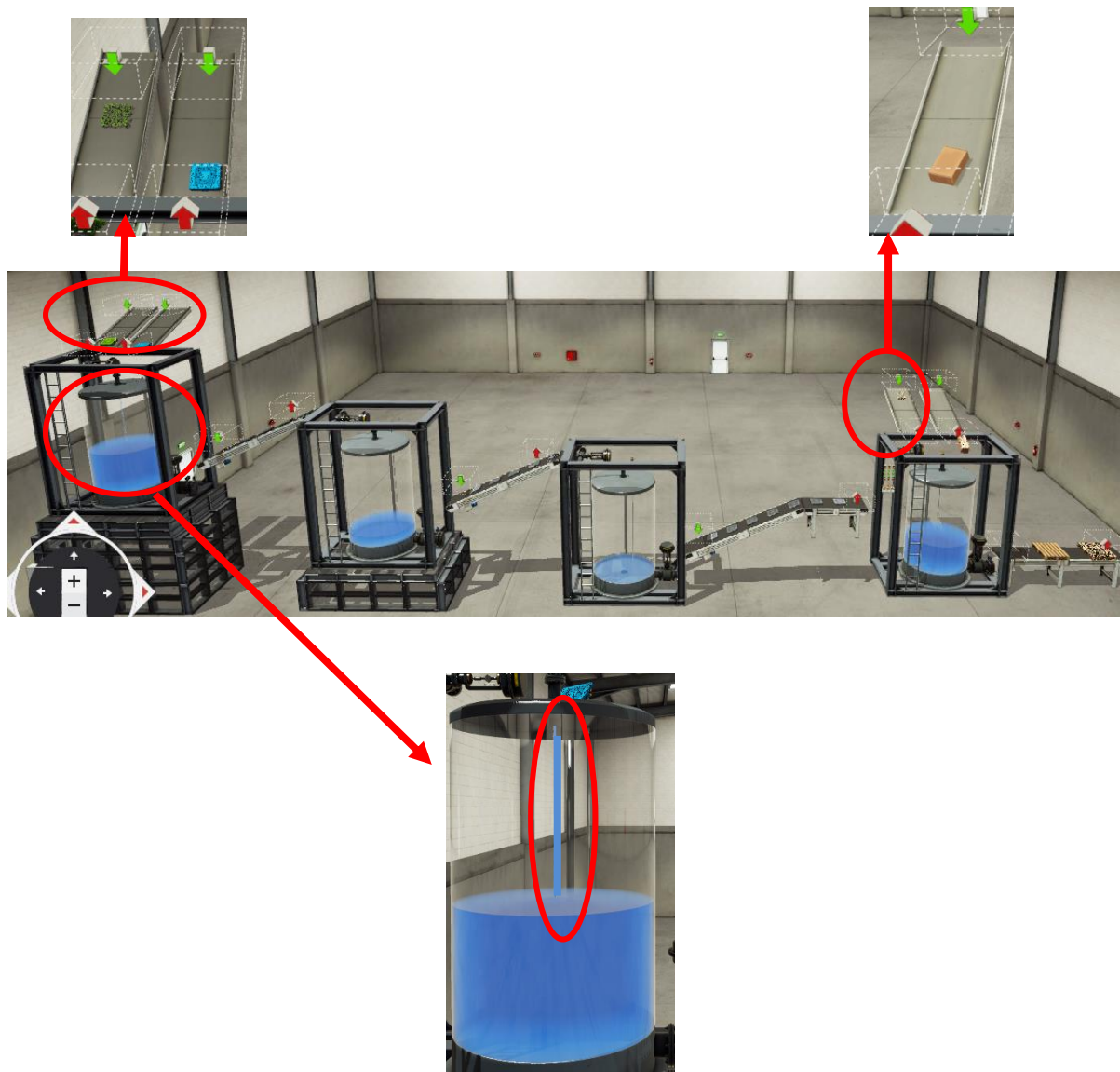


Figura 47. Gráficas de la insulina y la glucosa en el plasma del simulador respecto al escenario 1

DISEÑO E IMPLEMENTACIÓN DE UNA HERRAMIENTA EDUCATIVA SOBRE LA GESTIÓN DE DIABETES TIPO 1 EMPLEANDO HERRAMIENTAS DE SIMULACIÓN 3D DE PROCESOS INDUSTRIALES



Figura 48. Equilibrio del escenario 1



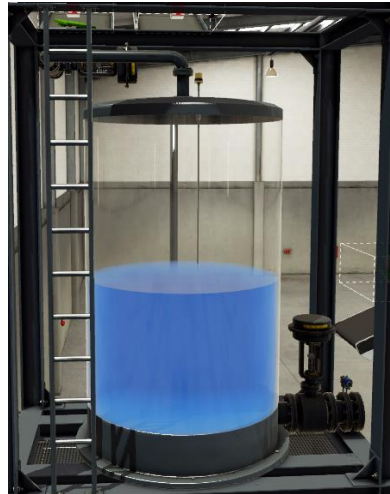


Figura 50. Bolo completamente administrado del escenario 1

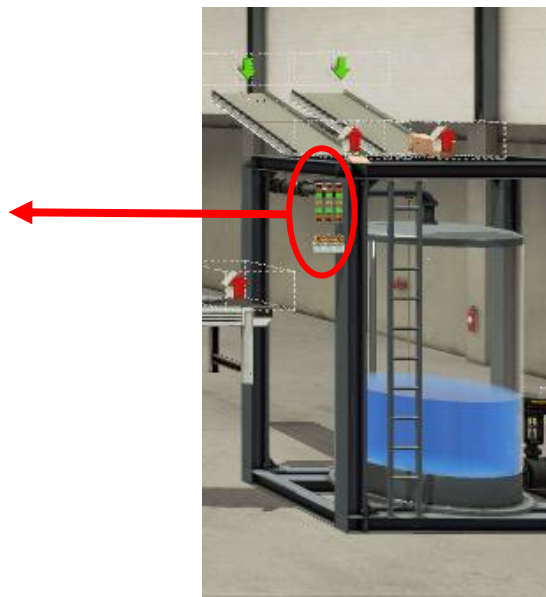


Figura 51. Niveles de glucosa del escenario 1

El vídeo del escenario 1 se encuentra disponible en:
http://personales.upv.es/jldiez/TFG_MTV/Escenario_1.mp4

- **Escenario 2:**

Desayuno a las 8h con 30g de carbohidratos.

Bolo a las 8h con 8 U/min (se ha aumentado el bolo de insulina).

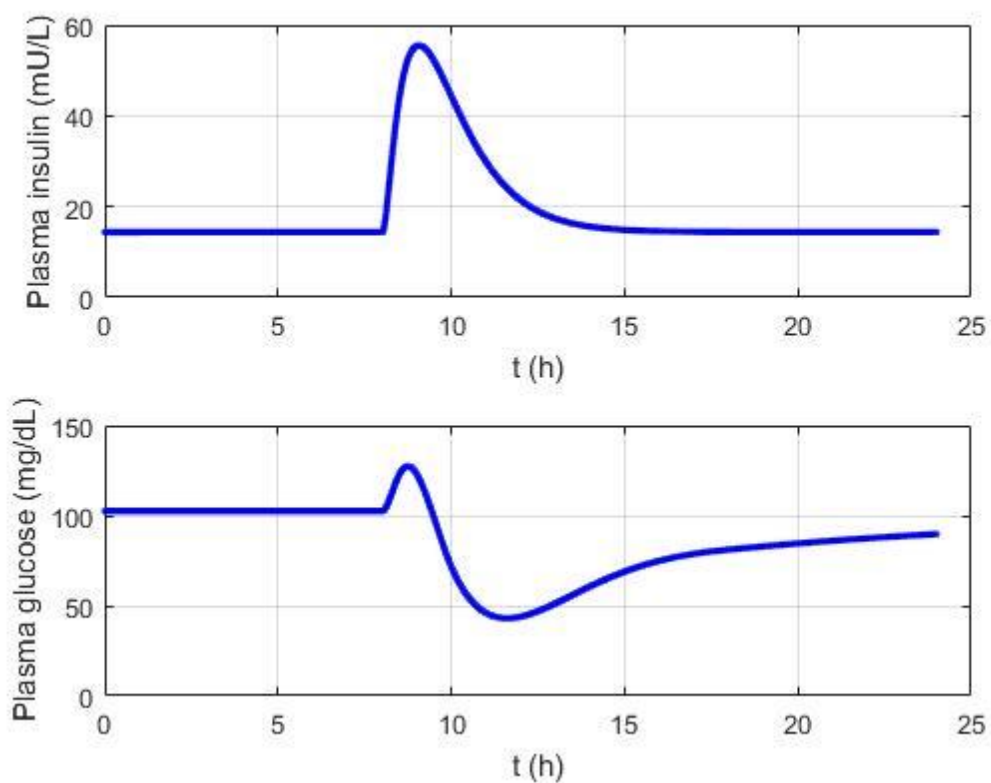


Figura 52. Gráficas de la insulina y la glucosa en el plasma del simulador respecto al escenario 2



Figura 53. Bolo completamente administrado del escenario 2

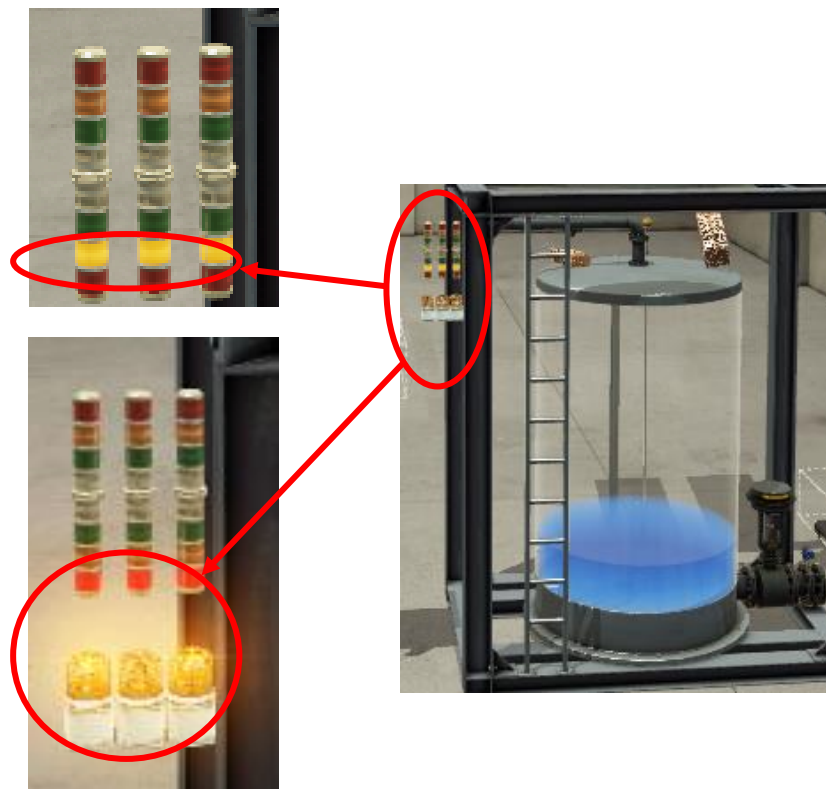


Figura 54. Niveles de glucosa del escenario 2

El vídeo del escenario 2 se encuentra disponible en:
http://personales.upv.es/jldiez/TFG_MTV/Escenario_2.mp4

- **Escenario 3:**

Desayuno a las 8h con 30g de carbohidratos.

Bolo a las 8h con 1 U/min (se ha disminuido el bolo de insulina).

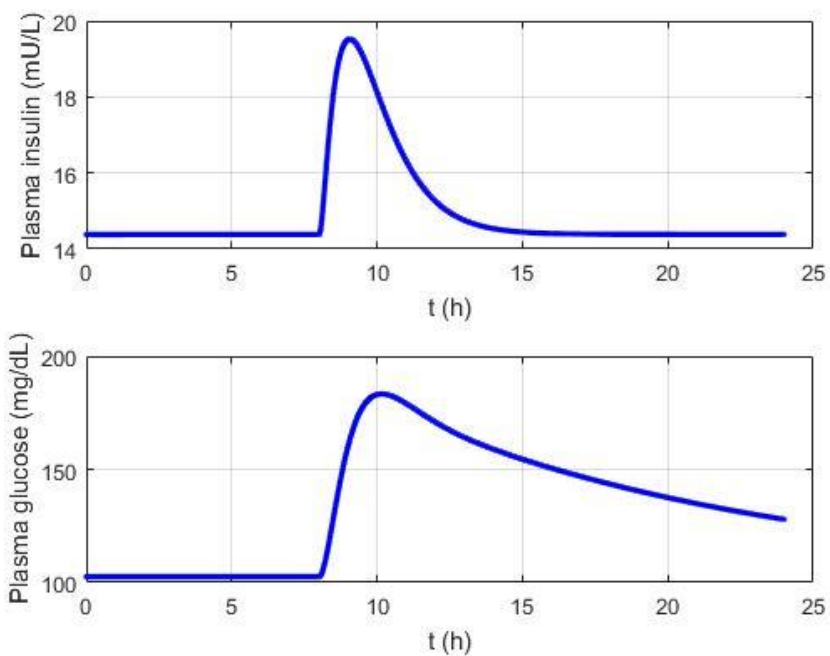


Figura 55. Gráficas de la insulina y la glucosa en el plasma del simulador respecto al escenario 3

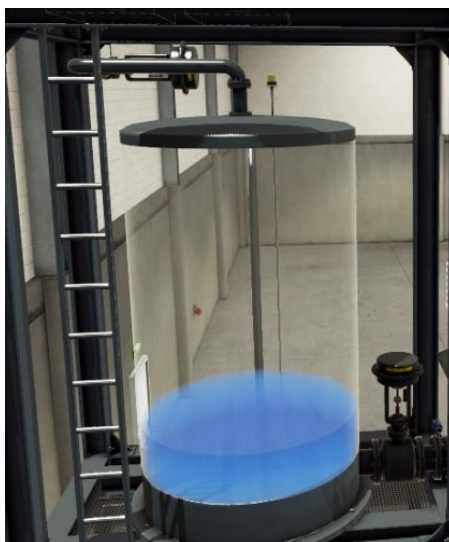


Figura 56. Bolo completamente administrado del escenario 3



Figura 57. Niveles de glucosa del escenario 3

El vídeo del escenario 3 se encuentra disponible en:
http://personales.upv.es/jldiez/TFG_MTV/Escenario_3.mp4

- **Escenario 4:**

Desayuno a las 8h con 30g de carbohidratos.

Bolo a las 8h con 3 U/min.

Ejercicio a las 17h con una duración de 45 min.

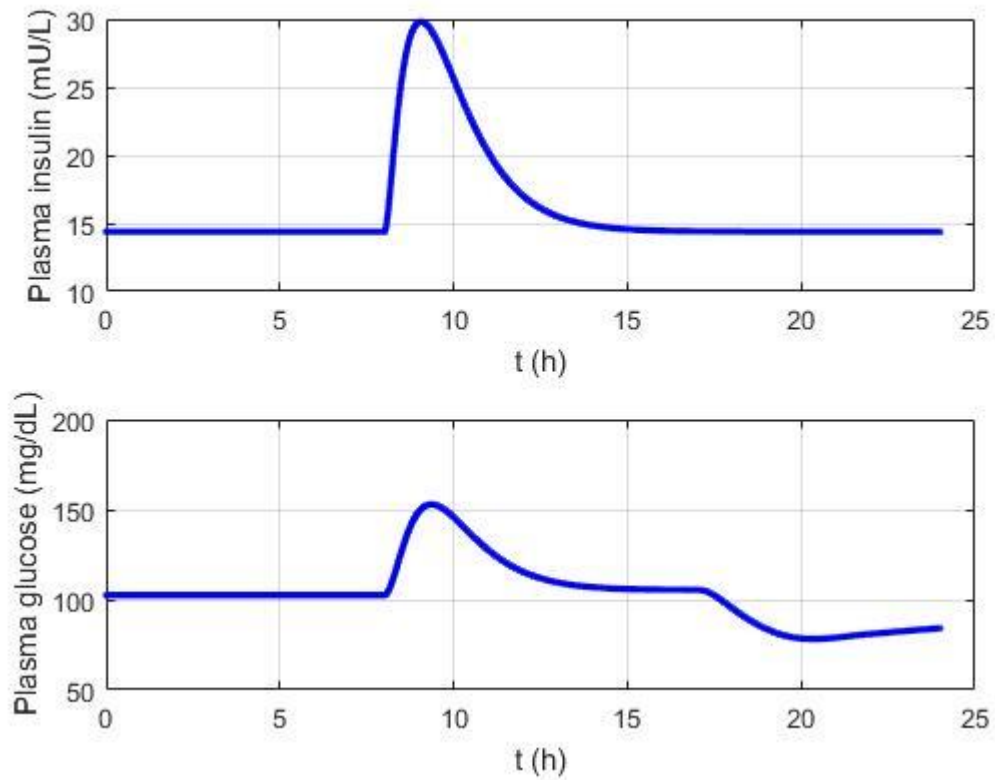


Figura 58. Gráficas de la insulina y glucosa en plasma del simulador respecto al escenario 4



Figura 59. Bolo completamente administrado del escenario 4



Figura 60. Niveles de glucosa del escenario 4

De este último escenario no se ha realizado un vídeo ya que comparando las figuras 47 y 58 se aprecia una alta similitud entre el escenario 1 y el 4. El único cambio es el ejercicio realizado que hace disminuir un poco más el nivel de glucosa.

La aplicación desarrollada no realiza vídeos. El objetivo era crear una aplicación que, manejada por personal especializado (sin necesidad de tener conocimientos de programación) como médicos o especialistas en educación diabetológica, sirviera de apoyo visual para explicar conceptos que de otro modo son difíciles de entender por parte de los pacientes.

Sin embargo, los vídeos pueden ser generados como captura de pantalla de las simulaciones realizadas mediante cualquier herramienta de edición de vídeo, bien por el personal clínico si tiene los conocimientos o con apoyo de personal técnico. Dichos vídeos, como los de los escenarios de este capítulo a modo de ejemplo, pueden ser reutilizados por el personal médico dejar a disposición de los pacientes con el material adicional que se considere.

Capítulo 8

Conclusiones, limitaciones y trabajo futuro

En este apartado se van a explicar las conclusiones de la aplicación desarrollada a través de los resultados del capítulo anterior. También se enumerarán las limitaciones encontradas a lo largo del proyecto y el trabajo futuro que se podría realizar para una mejora de la aplicación.

8.1 Conclusiones

Escenario 1 (bolo de insulina = 3 U/min):

En la figura 47 se observa que la glucosa está siempre en unos valores correctos ya que no supera los 180 mg/dl ni está por debajo de 70 mg/dl, por lo que el bolo introducido es el idóneo. Si se compara esta figura con la 51 en la que aparecen las luces del compartimento de la glucosa en verde, ambas muestran normoglucemia.

Escenario 2 (bolo de insulina = 8 U/min):

Por una parte, en la figura 52 se observa que la insulina aumenta debido a las 8 U/min de insulina administrada. Este incremento en el tratamiento conlleva a una importante bajada en los niveles de glucosa (inferior a los 54 mg/dl). Estos niveles son propios de la hipoglucemia severa.

Por otra parte, en la figura 54 se muestran las luces amarillas encendidas (hipoglucemia) y posteriormente las luces rojas y alarmas encendidas (hipoglucemia severa).

Como puede observarse, tanto en la gráfica como en las luces de la glucemia están presentes los mismos resultados: una dosis elevada de insulina es capaz de disminuir la glucemia a unos niveles altamente peligrosos.

Escenario 3 (bolo de insulina = 1 U/min):

Tanto en este caso como en los dos anteriores la gráfica del simulador y las luces de la interfaz 3D muestran los mismos resultados. En este caso, está presente la hiperglucemia. Esto es debido a la poca insulina que se ha administrado. Este escenario es todo lo contrario al anterior. Los niveles de glucosa son superiores a 180 mg/dl.

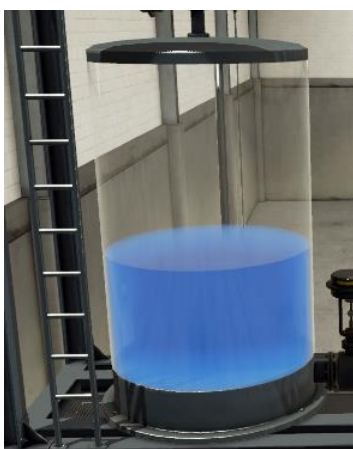
Hay que tener en cuenta que aunque sea más peligroso tener hipoglucemia severa que hiperglucemia, deben de evitarse ambos casos y conseguir mantener la glucosa en unos niveles entre 70 mg/dl y 180 mg/dl.

Escenario 4 (bolo de insulina = 3 U/min y ejercicio):

Como a través de los tres escenarios anteriores se ha podido observar que el único que administra la correcta dosis de insulina es el primero, se le ha querido añadir deporte al escenario 1.

En la figura 58, si se observa la gráfica de la glucosa en plasma se podrá ver que el nivel es el adecuado. Comparándolo con el del escenario 1, este nivel disminuye más debido al consumo de la glucosa a la hora de realizar ejercicio. Hay que tener en cuenta que un exceso de ejercicio podría consumir bastante glucosa y se podría padecer hipoglucemia. Por ello hay que tener en cuenta la dosis administrada de insulina y el ejercicio realizado.

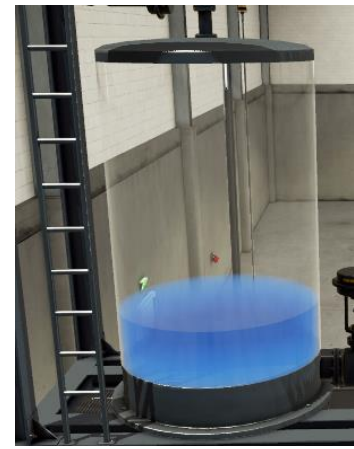
En la figura 61 se puede comparar la administración del bolo de insulina en cada uno de los tres primeros escenarios. Además, en la figura 62 se muestran los niveles de glucosa correspondientes.



Escenario 1

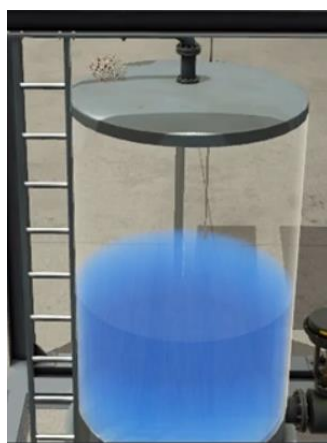


Escenario 2



Escenario 3

Figura 61. Comparación de la administración del bolo en los distintos escenarios



Escenario 1



Escenario 2



Escenario 3

Figura 62. Comparación del nivel de glucosa en los distintos escenarios

A pesar de las limitaciones explicadas en el siguiente apartado y con los cuatro escenarios previamente implementados, se apreció un buen funcionamiento de la interfaz gráfica 3D. Además, a través de las figuras añadidas en la presente memoria, se demuestra que se cumple el objetivo del proyecto: ofrecer una herramienta de aprendizaje donde cualquier persona con o sin conocimientos acerca de la diabetes y la programación, pueda entender de una manera clara y sencilla la gestión de diabetes tipo 1.

Como se ha comentado al final del capítulo 7, tanto el uso de la aplicación desarrollada como el de los vídeos generados para la educación del paciente, requiere de las adecuadas explicaciones y supervisión de especialistas en educación diabetológica.

8.2 Limitaciones

A la hora de desarrollar la aplicación, se encontraron las siguientes limitaciones:

- Se intentaron colocar los tres tanques de insulina verticalmente, esto es, uno encima del otro. Como los tanques miden 3 metros de altura y la nave no alcanza los 9 metros, fue imposible este primer planteamiento. Por ello, se utilizaron las plataformas.
- Para que el usuario final pudiera apreciar una imagen sencilla y clara se buscó la manera de colocar carteles indicativos en la interfaz gráfica 3D. Esta medida fue imposible implementarla a través de Factory IO. Sólo se pudieron poner en los vídeos.
- Para mejorar el flujo de materia, se pensó en variar la velocidad de las cintas transportadoras y la de los emisores. De esta manera, si la insulina pasaba de un tanque a otro lentamente y con poco caudal, se vería poca materia y la cinta a velocidad baja. En cambio, si la insulina fuera rápidamente, se vería mucha materia y una gran velocidad de la cinta transportadora. Sólo se consiguió variar la cinta al pasar de la configuración digital a la analógica. Como no se pudo cambiar la configuración de los emisores a través del código, se descartó la idea completa ya que carecía de sentido.

8.3 Trabajo futuro

Por un lado, para la mejora de la aplicación se podrían explorar otras alternativas a Factory IO ya que se han encontrado ciertas limitaciones.

El nuevo programa debería dejar cambiar las dimensiones de los elementos para poder colocar los tanques de insulina verticalmente. También tendría que tener más elementos creados para poder emitir un flujo de agua a través de canales abiertos en vez de usar la materia sólida y las cintas transportadoras. Daría una sensación de continuidad de flujo. Otra opción interesante sería el poder realizar explicaciones dentro de la plataforma para guiar al usuario. Además, dependiendo de las herramientas que tuviera el programa seleccionado, se podría probar con el simulador de UVA/Padova nombrado previamente ya que aunque sea más laborioso, es muy completo.

Por otro lado, si se desea seguir con Factory IO, se podría revisar el código implementado e intentar acortar los cálculos para una mayor velocidad de ejecución. Así, se podría reducir el tiempo de simulación 3D y dependiendo de la velocidad, ya no harían falta los vídeos.

Como última idea para mejorar la herramienta educativa se propone la realización de un estudio para obtener la opinión de las personas diabéticas y poder mejorar las futuras versiones de la aplicación. Esta última idea no se ha llevado a cabo por falta de tiempo pero sería interesante ponerla en práctica.

Referencias

- [1] Asociación Diabetes Madrid. Recuperado en Abril de 2019, de <https://diabetesmadrid.org/medidores-continuos-glucosa/>
- [2] Canal Diabetes. *La normoglicemia en DM1, más cerca con los sistemas de Páncreas Artificial*. Recuperado en Marzo de 2019, de <https://cuidateplus.marca.com/enfermedades/digestivas/2002/08/29/para-sirve-pancreas-5845.html>
- [3] Carbonier, D. MiAyuno. *Qué son los cuerpos cetónicos*. Recuperado en Abril de 2019, de <https://www.miayuno.es/que-son-los-cuerpos-cetonicos/>
- [4] Controlife. Recuperado en Junio de 2019, de <https://www.controlifecr.com/products/114041/glucometro-element>
- [5] Cuidate Plus. Recuperado en Abril de 2019, de <https://cuidateplus.marca.com/enfermedades/digestivas/diabetes.html>
- [6] Diabetes Teaching Center. *Tipos de insulina*. Recuperado en Junio de 2019, de <https://dtc.ucsf.edu/es/tipos-de-diabetes/diabetes-tipo-2/tratamiento-de-la-diabetes-tipo-2/medicamentos-y-terapias-2/prescripcion-de-insulina-para-diabetes-tipo-2/tipos-de-insulina/>
- [7] Hernández, R. C. (s.f.). Fundación para la Diabetes. *Sistemas de monitorización continua de glucosa*. Recuperado en Abril de 2019, de <https://www.fundaciondiabetes.org/general/articulo/173/sistemas-de-monitorizacion-continua-de-glucosa>
- [8] Hovorka, R. (2004). Physiological Measurement. Obtenido de *Nonlinear model predictive control of glucosa concentrarion in subjects with type 1 diabetes*
- [9] Instituto Universitario de Automática e Informática Industrial (Instituto ai2). Tecnodabetes. Recuperado en Junio de 2019, de <https://tecnodiabetes.ai2.upv.es/>
- [10] Man, C. D.. *The UVA/PADOVA Type 1 Diabetes Simulator*
- [11] MayoClinic. *Diabetes*. Recuperado en Abril de 2019, de <https://www.mayoclinic.org/es-es/diseases-conditions/diabetes/diagnosis-treatment/drc-20371451>
- [12] Medline Plus. Recuperado en Mayo de 2019, de <https://medlineplus.gov/spanish/diabetestype1.html>
- [13] Medline Plus. Recuperado en Mayo de 2019, de <https://medlineplus.gov/spanish/diabetestype2.html>
- [14] Medtronic. Recuperado en Mayo de 2019, de <https://www.medtronic-diabetes.com/es/terapia-con-bomba-de-insulina>

- [15] Oliver, A. A. (2013-2014) Implementación de un integrador de modelos de pacientes virtuales diabéticos tipo 1.
- [16] Real Games.Factory IO. Recuperado en Febrero de 2019, de <https://factoryio.com/docs/manual/parts/stations/#tank>

ANEXOS

Anexo I. Selección de alternativas

Como se ha explicado anteriormente, para elegir Factory IO se siguieron los siguientes pasos:

1. Elegir distintas alternativas:

- Factory IO
- Simulink 3D Animation
- AC3D
- Blender 3D

Como se ha explicado, Simulink 3D Animation se descartó porque se querían aprovechar las herramientas comerciales específicas para la representación 3D y se quería aprender la interconexión de variables entre diferentes aplicaciones.

2. Seleccionar los criterios:

Se tuvo en cuenta la complejidad del programa, el tiempo dedicado a comprender cómo funciona, el coste, las limitaciones y la estética.

3. Ponderar los criterios (método indirecto, tabla 6):

	Complejidad	Tiempo	Coste	Limitaciones	Estética		
Complejidad		0,4	0,2	0,4	0,4	1,40	0,140
Tiempo	0,6		0,4	0,4	0,3	1,70	0,170
Coste	0,8	0,6		0,6	0,75	2,75	0,275
Limitaciones	0,6	0,6	0,4		0,6	2,20	0,220
Estética	0,6	0,7	0,25	0,4		1,95	0,195
						10	1

Tabla 6. Ponderación de criterios a través del método indirecto

4. Valoración de las alternativas para cada criterio (tabla 7 y 8).

	Complejidad	Tiempo (h)	Gratuito	Limitaciones	Estética
Factory IO	baja	10	sí ⁷	4	6
AC3D	media	8	no	2,5	8
Blender 3D	alta	25	sí	1	10

Tabla 7. Valoración de las alternativas para cada criterio

⁷ Factory IO no es un programa gratuito. Sin embargo, para este proyecto ha sido cedido por el Departamento de Ingeniería de Sistemas y Automática.

	Complejidad	Tiempo	Gratuito	Limitaciones	Estética
	0,140	0,170	0,275	0,220	0,195
Factory IO	4	10	1	4	6
AC3D	6	8	0	2,5	8
Blender 3D	9	25	1	1	10
	min	min	máx	min	máx

Tabla 8. Utilización de la valoración numérica

5. Seleccionar la mejor alternativa a través de la suma ponderada, para ello se normalizan los valores de la matriz (tabla 9).

	Complejidad	Tiempo	Gratuito	Limitaciones	Estética	
	0,140	0,170	0,275	0,220	0,195	
Factory IO	0,444	0,400	1,000	1,000	0,600	0,042
AC3D	0,667	0,320	0,000	0,625	0,800	-0,569
Blender 3D	1,000	1,000	1,000	0,250	1,000	-0,343
	min	min	máx	min	máx	

Tabla 9. Normalización de la valoración y resultados

La alternativa óptima era Factory IO ya que obtuvo un mayor resultado (números en amarillo).

Anexo II. Manual de instalación

MATLAB

Los pasos a seguir para la correcta instalación de MATLAB son los siguientes:

1. Acceder a <https://software.upv.es/> e iniciar sesión con el usuario y contraseña de la Universidad Politécnica de Valencia (UPV).
2. Dirigirse a la carpeta Software para alumnos y clicar sobre MATLAB TAH donde aparecerán las instrucciones y el enlace de la instalación.
3. Si no se dispone de la OPC Toolbox (4.0.2), se deberá de instalar.

KEPServer

Para poder realizar esta segunda instalación se debe de estar conectado a la UPV. Se puede hacer desde la UPVNET (si se está dentro de la universidad) o desde la VPN (si se está fuera). Una vez conectado, los pasos a seguir son los siguientes:

1. En EQUIPO se debe de introducir: \\vmisalabos\practicass (figura 63).

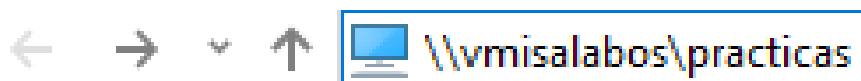


Figura 63. Instalación de KEPServer

2. Acudir a la carpeta Software para Alumnos.
3. Abrir la carpeta Kepware OPC Server.
4. Seleccionar KEPServerEX5.

Factory IO

Para la instalación y uso de Factory IO también se debe de estar conectado a la universidad. Este programa ha sido prestado por el Departamento de Ingeniería de Sistemas y Automática.

Una vez instalado, en el directorio en el que está el programa hay que añadir un fichero floating.lic para que Factory IO localice la licencia y funcione correctamente. En dicho fichero hay que introducir lo siguiente:

HOST 158.42.16.232

Anexo III. Manual de usuario

Introducción

Este manual sirve para que cualquier usuario que desee utilizar la aplicación creada pueda hacerlo sin ningún problema.

Para hacer un buen uso de la aplicación, que resulte útil y sea más sencilla de entender, se deben tener conocimientos informáticos básicos y conocer qué es la diabetes, sus síntomas y tratamientos.

Descripción de la aplicación

La aplicación ha sido diseñada para que cualquier persona pueda entender de una manera clara y sencilla la gestión de diabetes tipo 1. Por ello, el usuario puede ser un paciente interesado en saber cómo funciona su organismo, un estudiante que necesita adquirir conocimientos o un trabajador.

El usuario apreciará una interfaz en 3D donde visualizará distintos tanques que representan la insulina y la glucosa dentro del organismo. Para evitar que sea imprescindible tener conocimientos de programación, se ha implementado una interfaz gráfica para introducir los datos de entrada.

Se necesitan tres programas para el correcto funcionamiento. Primero se utilizará un programa para simular a un paciente virtual con diabetes tipo 1. De esta manera, se podrán obtener todos los datos necesarios que se podrían extraer de una persona real con diabetes pero sin que esta intervenga. Para ello, el usuario deberá de rellenar ciertas casillas en las que tendrá que introducir el peso, la insulina basal, el ejercicio que realiza, etc. Después, otro programa representará la interfaz en 3D en la cual el usuario verá los resultados. Todo esto no se podría llevar a cabo sin el programa de conexión, el cual envía y recibe datos para que los dos primeros estén conectados y funcionen de la manera adecuada.

Conectando todos los programas entre sí y con los conocimientos previos explicados a continuación, se pondrá entender cómo la insulina, los carbohidratos y el ejercicio afectan a los niveles de glucosa del organismo.

Conocimientos previos

La diabetes es una enfermedad crónica en la que los niveles de glucosa en sangre son elevados. La glucosa se almacena en las células del hígado en forma de glucógeno y más tarde, se convierte en glucosa cuando el cuerpo necesita energía.

Es importante que los niveles de glucosa estén entre 70 y 180 mg/dl. Si estos niveles aumentan se padecería hiperglucemia, en caso contrario, hipoglucemia. Hay que tener en cuenta que si el nivel está en torno a 54 mg/dl puede llegar a presentarse la hipoglucemia severa, por lo que se debe de acudir lo antes posible al hospital ya que se necesitará atención médica.

El organismo dispone de distintas hormonas para mantener unos niveles de glucosa adecuados (normoglucemia), las más importantes son:

- **Glucagón:** aumenta la glucosa.
- **Insulina:** disminuye la glucosa.

Esta aplicación sólo es válida para simular pacientes con diabetes tipo 1. Este tipo de diabetes ataca a sus propias células y no se segrega la insulina necesaria para mantener los niveles normales de glucosa.

Algunos de los síntomas de esta enfermedad son los siguientes: visión borrosa, fatiga, deseo frecuente de orinar (poliuria), aumento de la sed (polidipsia), aumento del apetito (polifagia), adelgazamiento no intencionado, irritabilidad, cambios de humor y cansancio.

Para evitar estos síntomas se debe de seguir un tratamiento a través de la nutrición, la actividad física y las inyecciones de insulina.

Especificaciones técnicas

Para poder utilizar la aplicación se deben de cumplir ciertos requisitos de hardware y software.

Requisitos de hardware

Son los conjuntos de elementos físicos que constituyen un ordenador o un sistema informático.

- **Ordenador :**
 - Memoria RAM: al menos 1 GB.
 - Procesador: cualquier procesador Intel, si se utilizan otros procesadores no se asegura su funcionamiento.
 - Disco duro: 100 GB.

Requisitos de software

Conjunto de programas que permiten realizar al ordenador determinadas tareas.

- **Sistema Operativo:** se asegura el funcionamiento si se utiliza Windows 7 o superior.

- **MATLAB**
 - **Factory IO**
 - **KEPServer**
- } La instalación se encuentra desarrollada en el Anexo I.

Además, se necesitan los siguientes archivos creados para la aplicación:

- Archivo para KEPServer: conexion.
- Archivo para Factory IO: Escena Diabetes Tipo 1.
- Archivos para MATLAB: escribir, leer, inicializar, prueba, equilibrio, trabajo, simulador y dos archivos llamados entrada_datos.

Puesta en marcha de la aplicación

Una vez que se cumplan los requisitos hardware y software, se hayan instalado los programas y se disponga de los archivos nombrados anteriormente, se puede poner en marcha la aplicación.

1. Si se va a ejecutar fuera de la Universidad Politécnica de Valencia se deberá de estar conectado a la VPN. Para que Factory IO reconozca las licencias.
2. Abrir el fichero de KEPServer, clicar sobre Tools y seguidamente en Launch OPC Quick Client o directamente en el icono de la derecha de la figura 64.

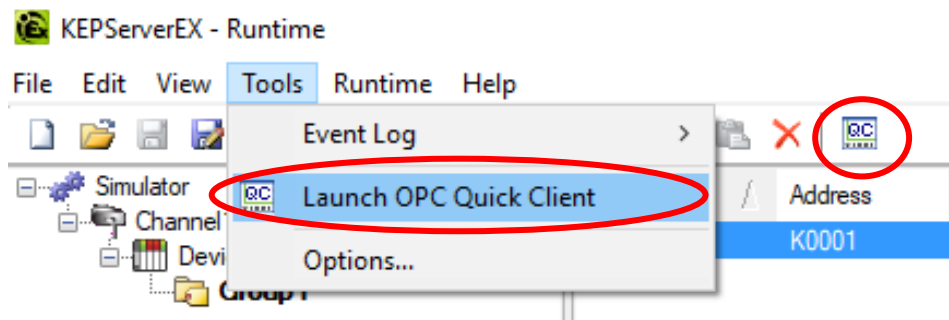


Figura 64. Conexión de KEPServer

3. Abrir la escena de Factory IO, pulsar la tecla f4 y esperar a ver una pantalla como la de las figuras 40, 41 y 42. Volver a presionar f4.
4. Abrir MATLAB y seleccionar la carpeta donde se encuentran los archivos previamente nombrados de este programa. Para ello, observar la figura 65.

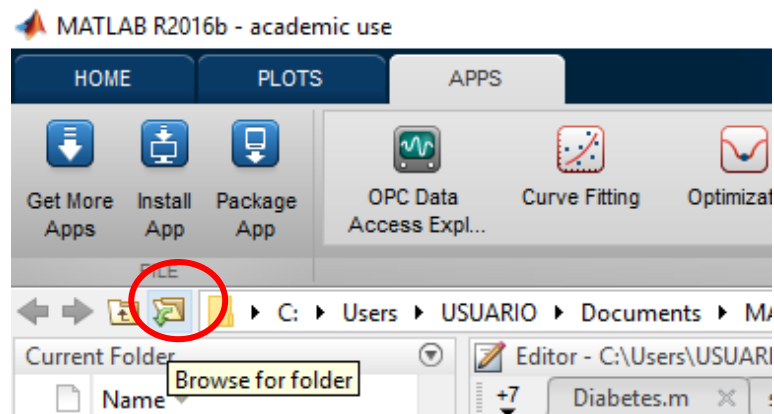


Figura 65. Selección de los archivos de MATLAB

5. Seleccionar el archivo entrada_datos y ejecutar la función, para ello hacer doble clic en entrada_datos y seguidamente pulsar el botón triangular de Run, figura 66.

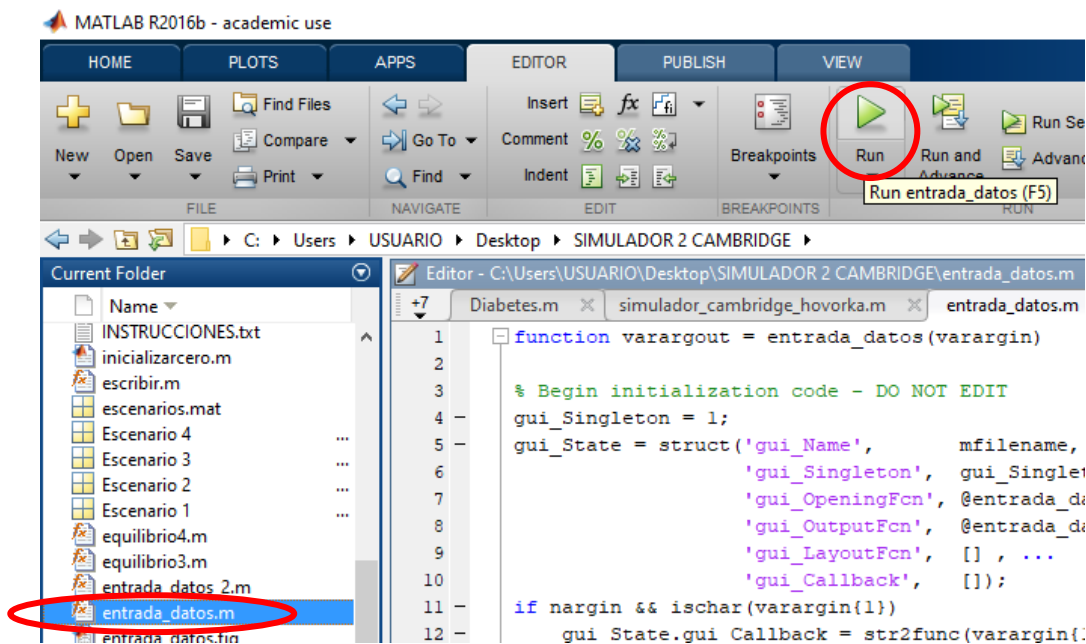


Figura 66. Selección de entrada_datos y su ejecución

6. Una vez ejecutado aparecerá la interfaz gráfica donde el usuario deberá de rellenar los datos de entrada.

Entrada de datos en la interfaz gráfica

En la interfaz gráfica aparecerá una pantalla como la de la figura 67.

The screenshot shows a software interface titled "PERSONA CON DIABETES TIPO1". On the left, there is a sidebar with a text input field for "NOMBRE DEL NUEVO ESCENARIO", a "GUARDAR ESCENARIO" button, a scrollable list area, a "MOSTRAR ESCENARIOS" button, a "MOSTRAR VARIABLES DEL ESCENARIO" button, and an "ELIMINAR ESCENARIO" button. The main area contains two tables of input fields. The first table is for meals, with columns for "Hora [0-23]", "Minutos [0-59]", and "CHO (g)", and rows for "DESAYUNO", "COMIDA", and "CENA". The second table is for exercise, with columns for "Hora [0-23]", "Minutos [0-59]", "Duración (min)", "Peso (Kg)", "Sensibilidad [0-1]", and "Basal (U/min)", and a row for "EJERCICIO". At the bottom, there are three buttons: "RESTAURAR ÚLTIMOS DATOS", "ELIMINAR VARIABLES", and a large blue "START" button.

Figura 67. Interfaz gráfica lista para insertar los datos deseados

El usuario puede introducir los datos de entrada sin guardarlos y clicar en el botón START o guardarlos a través de un escenario para poder utilizarlos en otras ocasiones.

Si el usuario elige la primera opción, los datos introducidos serán los siguientes:

- **Horas de las comidas:** se debe de poner la hora del desayuno, comida y/o cena. Se puede introducir cualquier hora entre las 0 y las 23 y cualquier minuto entre el 0 y el 59.
- **Carbohidratos ingeridos:** deben de apuntarse los gramos de carbohidratos ingeridos en cada comida.
- **Horas de los bolos de insulina:** se realiza igual que las horas de las comidas.
- **Bolos de insulina:** se apuntan las U/min de insulina que se administrarán.
- **Ejercicio:** si se va a realizar ejercicio habrá que rellenar la casilla de la hora y los minutos igual que en las comidas y bolos. Además, se tiene que indicar el tiempo que dura la actividad física en minutos.
- **Peso:** es el peso del paciente en kg.

- **Sensibilidad a la insulina:** sensibilidad de la insulina en tanto por uno.
- **Basal:** insulina basal que se suministra continuamente en el cuerpo del paciente. Se mide en U/min.

Cualquier actividad que no se desee realizar se deberá de introducir un cero. Si por el contrario se deja en blanco, aparecerá un mensaje de error.

Si se prefiere crear un escenario, aparte de los datos explicados anteriormente, se tendrá que introducir un nombre al escenario y pulsar el botón GUARDAR ESCENARIO. Si se pulsa dicho botón sin introducir el nombre, aparecerá otro mensaje de error.

Además de los datos introducidos se dispone de 7 botones:

- **Guardar escenario:** guarda en un archivo de MATLAB el escenario con el nombre y los datos introducidos. El nombre del escenario creado aparecerá en la nueva lista.
- **Mostrar escenarios:** muestra en la lista el nombre de todos los escenarios creados que no hayan sido eliminados.
- **Mostrar variables del escenario:** muestra las variables del escenario seleccionado.
- **Eliminar escenario:** elimina el escenario seleccionado.
- **Restaurar últimos datos:** hace aparecer por pantalla los últimos datos correctamente introducidos.
- **Eliminar variables:** elimina las variables de la pantalla.
- **Start:** si han sido insertadas todas las variables correctamente empezará la simulación. En caso contrario, aparecerá un mensaje indicando los datos erróneos.

Una vez se hayan introducido los datos y se pulse el botón START pueden suceder dos cosas:

1. **Algún dato es erróneo:** aparecerá un mensaje de error y las casillas que tengan un número erróneo se pondrán rojas, observar la figura 68.
2. **Todos los datos son correctos:** si los datos introducidos están en el rango determinado, aparecerá un mensaje como el siguiente: La simulación empezará en breves.

PERSONA CON DIABETES TIPO1

	Hora [0-23]	Minutos [0-59]	CHO (g)	Hora [0-23]	Minutos [0-59]	Bolo (U/min)
DESAYUNO	<input type="text" value="8"/>	<input type="text" value="0"/>	<input type="text" value="20"/>	<input type="text" value="8"/>	<input type="text" value="0"/>	<input type="text" value="2"/>
COMIDA	<input type="text" value="14"/>	<input type="text" value="30"/>	<input type="text" value="60"/>	<input type="text" value="14"/>	<input type="text" value="30"/>	<input type="text" value="6"/>
CENA	<input type="text" value="21"/>	<input type="text" value="0"/>	<input type="text" value="30"/>	<input type="text" value="21"/>	<input type="text" value="0"/>	<input type="text" value="3"/>

	Hora [0-23]	Minutos [0-59]	Duración (min)	Peso (Kg)	Sensibilidad [0-1]	Basal (U/min)
EJERCICIO	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text" value="-3"/>	<input type="text" value="0.39"/>	<input type="text" value="0.016671"/>

Las variables en rojo tienen un valor erróneo. Si no desea realizar alguna actividad introduzca un cero.

Figura 68. Datos de entrada erróneos

Interfaz gráfica 3D

Tras haber introducido los datos de entrada correctamente y haber pulsado el botón START, empezará la simulación en Factory IO. Para ello se necesitará que esté abierta la escena del programa.

El usuario visualizará en pantalla lo mostrado en la figura 69.

En la interfaz hay cuatro tanques de agua, los tres de la izquierda representan la insulina y el de la derecha, la glucosa. El tanque de la izquierda es el compartimento de entrada, en el que la insulina llega una vez ha sido inyectada en el cuerpo. El tanque que se encuentra a su lado, es el compartimento intermedio, la insulina pasa a este punto después de haber sido almacenada en el primero. El tercero, es el correspondiente a la insulina en plasma. Al salir de ahí, el cuerpo ya es capaz de consumirla.

Si se aprecia una bajada del nivel de agua en los tanques de la insulina, se observará más adelante una bajada del nivel de glucosa ya que la insulina estará siendo consumida y hará que los niveles de glucosa disminuyan.

El nivel del tanque de glucosa también variará dependiendo de los carbohidratos ingeridos y el ejercicio realizado.



Figura 69. Interfaz gráfica de Factory IO

La insulina aparte de estar representada por el agua dentro de los tanques, está representada a través de una materia de distintos colores: la materia azul corresponde a los bolos de insulina, la verde a la insulina basal y la gris a ambas mezcladas.

La producción endógena y exógena de glucosa están representadas por cajas y la glucosa que se está consumiendo, por palés.

La materia, las cajas y los palés son trasladados por las cintas transportadoras o canales de transporte. De esta manera, se puede seguir la dirección de los distintos flujos, figura 70.

Al principio los tanques estarán vacíos y poco a poco se llenarán hasta el nivel de equilibrio, nivel con el que la persona diabética empezará el día. Una vez alcanzados dichos niveles, los compartimentos variarán dependiendo de los bolos de insulina administrada, los carbohidratos ingeridos y el deporte realizado.

Para saber si el nivel de glucosa es correcto se pueden observar las luces y alarmas situadas en la parte superior izquierda del tanque de glucosa.

Dichas luces y alarmas son las siguientes:

3 luces rojas en la parte superior: se encienden cuando está presente la hiperglucemia y el nivel de glucosa en el tanque supera los 250 mg/dl.

3 luces amarillas en la parte superior: se encienden cuando hay hiperglucemia y el nivel de glucosa en el tanque está entre los 180 mg/dl y los 250 mg/dl.

6 luces verdes en la parte intermedia: se encienden cuando se detecta normoglucemia, la glucosa está entre 70 mg/dl y 180 mg/dl.

3 luces amarillas en la parte inferior: se encienden cuando está presente la hipoglucemia y la glucosa está entre 54 mg/dl y 70 mg/dl.

3 luces rojas en la parte inferior: se encienden cuando hay hipoglucemia severa, la glucemia es inferior a 54 mg/dl. La hipoglucemia severa es la más peligrosa de todas, por ello a la vez que se encienden estas 3 luces rojas en la parte inferior, también se activan 3 alarmas.

Se pueden ver las luces y alarmas en la figura 71.

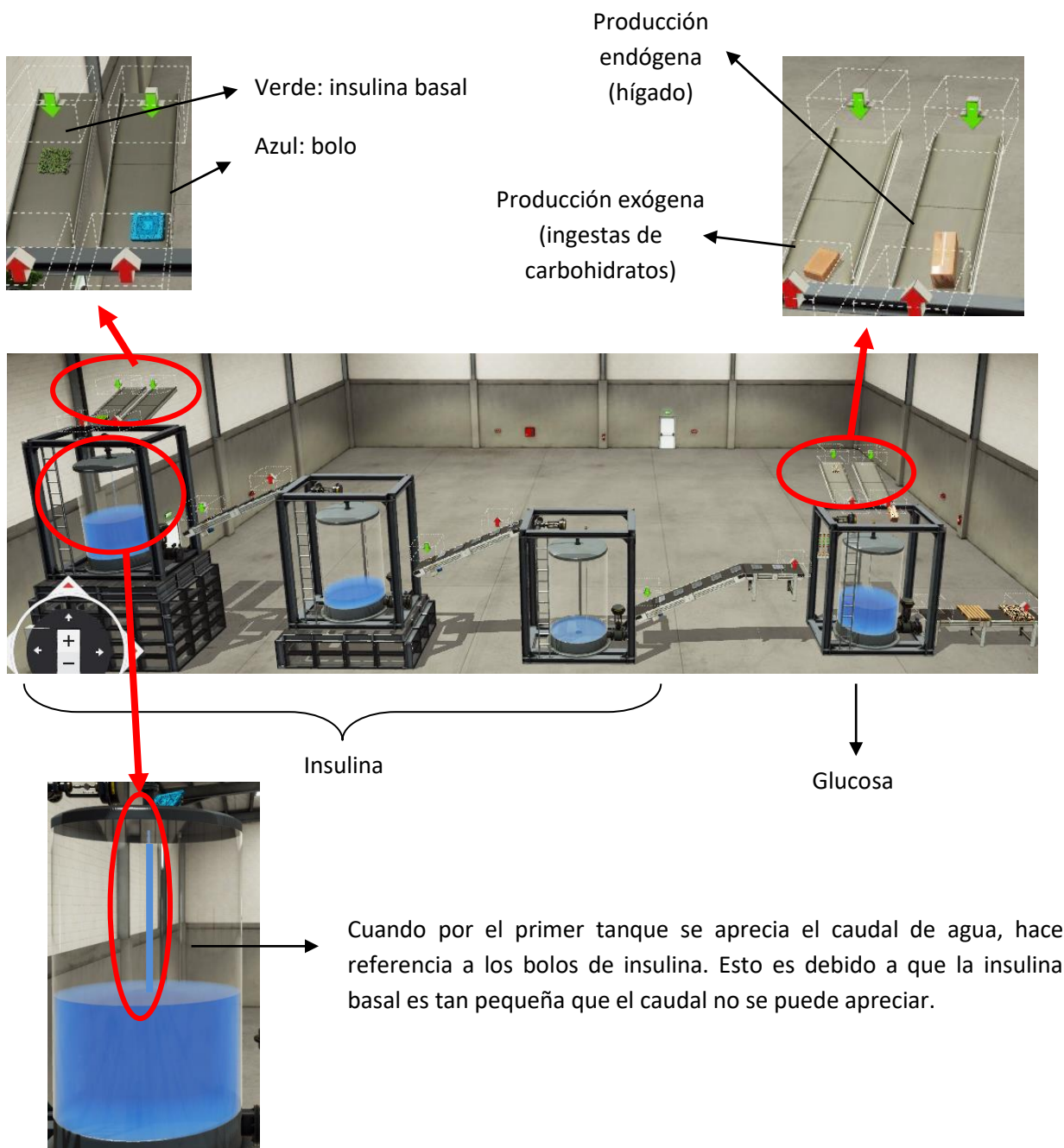


Figura 70. Interfaz gráfica 3D

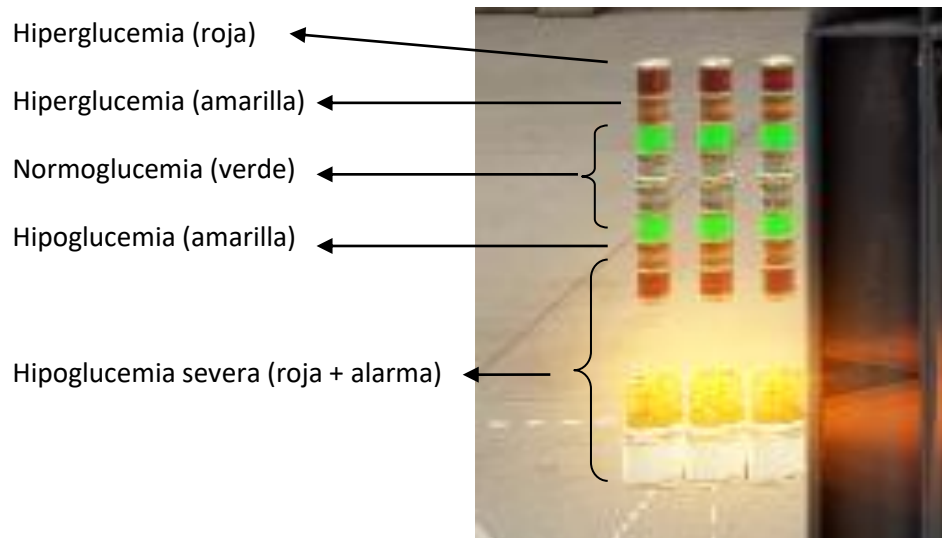


Figura 71. Control de glucosa

El usuario se puede desplazar por la interfaz a través de la cámara de navegación situada en la parte inferior izquierda de la pantalla. Si no está visible, presionar VIEW y en la pestaña que se abrirá, Camera Navigation (figura 72).

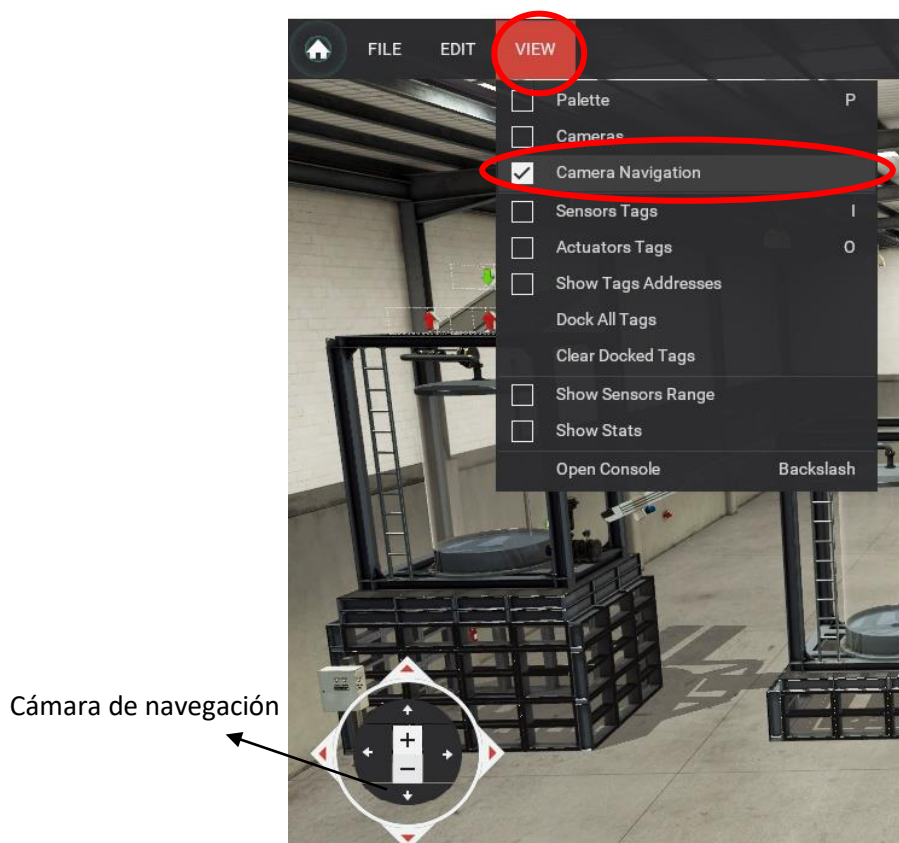


Figura 72. Control de navegación

Una vez entendido este manual, se podrá utilizar la aplicación de una manera útil y sencilla.

Anexo IV. Manual del programador

Introducción

Este manual sirve para facilitar la tarea al programador ya que informa acerca de la estructura interna del software.

A través de este manual, el programador podrá cambiar, mejorar o ampliar el código para cualquier trabajo futuro.

Conocimientos previos

Al igual que el usuario, el programador tendrá que tener unos conocimientos previos acerca de la diabetes (Anexo III. Manual de usuario, Conocimientos Previos). Además, tendrá que tener un amplio conocimiento sobre programación.

Primeros pasos

Antes de empezar a programar, se deben de instalar todos los programas que aparecen en el Anexo II. Una vez instalados, se podrá empezar a trabajar.

Simulador

Se quiere simular a un paciente con diabetes tipo 1, para ello, se ha utilizado el modelo de Cambridge, explicado en el capítulo 4 de la memoria.

El simulador fue implementado en MATLAB por los integrantes del Instituto Universitario de Automática e Informática Industrial (Instituto ai2) de la Universidad Politécnica de Valencia.

Se eligió MATLAB debido a que ofrece la resolución de ecuaciones diferenciales. Ciertos aspectos del simulador fueron cambiados debido a que se implementó una interfaz gráfica a través de GUIDE (MATLAB) para poder introducir de una manera más sencilla los valores de entrada.

Variables del código del simulador

Se cambiaron los siguientes aspectos respecto al código prestado por el Instituto ai2:

1. Todo el código se introdujo dentro de una función a la que se le pasaron las variables que el usuario puede introducir a través de la interfaz. Dichas variables se llamaron de la siguiente forma:
 - **peso**: peso en kilogramos del paciente.

- **sensibilidad**: sensibilidad a la insulina en tanto por uno.
- **th_chok**: representa la hora de la comida k. El valor introducido debe de estar entre 0 y 23, ambos incluidos.
- **tm_chok**: representa los minutos de la hora de la comida k. El valor debe de estar entre 0 y 59, ambos incluidos.
- **chok**: gramos de carbohidratos ingeridos en la comida k.
- **th_bolok**: exactamente igual que th_chok pero respecto a los bolos de insulina.
- **tm_bolok**: exactamente igual que tm_chok pero respecto a los bolos de insulina.
- **bolok**: exactamente igual que chok pero respecto a los bolos de insulina en U/ min.
- **vbasal**: insulina basal del paciente en U/min.
- **th_ejercicio**: representa la hora a la que se va a realizar ejercicio. Es un número entre 0 y 23, ambos incluidos.
- **tm_ejercicio**: los minutos de la hora a la que se hace el ejercicio. Número entre 0 y 59, ambos incluidos.
- **duracion_ejercicio**: tiempo en minutos de la duración del ejercicio a realizar.

Siendo k = 1, 2 o 3.

2. Se cambiaron las variables que hacían referencia a las modificaciones anteriores. Por ejemplo, si en el simulador original ponía bw (body weight) = 70, se cambió a bw = peso para sustituir el número por la variable introducida por el usuario.

A continuación, se muestran todos los cambios que se realizaron:

- bw=peso
- gamma=sensibilidad
- cho= [th_cho1*60+tm_cho1 cho1;...
th_cho2*60+tm_cho2 cho2;...
th_cho3*60+tm_cho3 cho3;]
- bolus= [th_bolo1*60+tm_bolo1 bolo1 1;...
th_bolo2*60+tm_bolo2 bolo2 1;...
th_bolo3*60+tm_bolo3 bolo3 1;]
- basal=[0 vbasal];

- `exercise=[th_ejercicio*60+tm_ejercicio 50 duracion_ejercicio];`

Debe de tenerse en cuenta que este simulador solo representa ejercicios con 50 VO₂ (consumo máximo de oxígeno). Por ello, la segunda casilla del vector se dejó intacta.

Si por ejemplo se desea realizar el siguiente horario de comidas:

Desayuno: a las 8 h con 20 g de carbohidratos.

Comida: a las 14:30 h con 40g de carbohidratos.

Cena: a las 21 h con 30g de carbohidratos.

La matriz a implementar tendría como primera columna el tiempo (en minutos) y como segunda, los carbohidratos (en gramos):

```
cho = [8*60      20;...
       14*60+30  40;...
       21*60     30]
```

3. Como hay datos que calcula el simulador y se necesitan más adelante, se generó un archivo de MATLAB con ciertas variables y constantes. Para ello, se escribió lo mostrado a continuación:

```
save datos_simulador t X basal bolus ke tmaxl vi k12 epg0 vg tmaxG f01;
```

Este código guarda en un archivo llamado `datos_simulador` las variables `t`, `X`, `basal` y `bolus` y las constantes `ke`, `tmaxl`, `vi`, `k12`, `epg0`, `tmaxG` y `f01`. A continuación, hay una breve explicación de cada dato guardado:

- **t**: vector de tiempos de la simulación.
 - **X**: matriz de resultados. Cada columna representa una variable del sistema de ecuaciones diferenciales. Las columnas de izquierda a derecha son las siguientes: `g1`, `g2`, `s1`, `s2`, `l`, `x1`, `x2`, `x3`, `Q1` y `Q2`. Además, cada fila hace referencia al instante de tiempo definido por el vector tiempo.
 - **basal y bolus**: contienen lo introducido previamente por el usuario.
 - **Ke, tmaxl, vi, k12, epg0, tmaxG y f01**: constantes explicadas con el modelo de Cambridge.
4. Para resolver las ecuaciones diferenciales, el simulador utiliza la función de MATLAB `ode15s`. Dicha función tiene la siguiente estructura: `[t, y] = ode15s (odefun, tspan, y0, options)`. Se cambió `tspan` para poder utilizar los valores introducidos por el usuario.

Interfaz gráfica

Una vez realizados los cambios en el código del simulador se realiza la interfaz gráfica. Se puede abrir una nueva, o bien, abrir la que ya se implementó para este proyecto.

Si se opta por la primera opción, dentro de MATLAB se debe de hacer lo correspondiente a la figura 73: clicar en HOME, abrir la pestaña de New, clicar en App y finalmente abrir GUIDE.

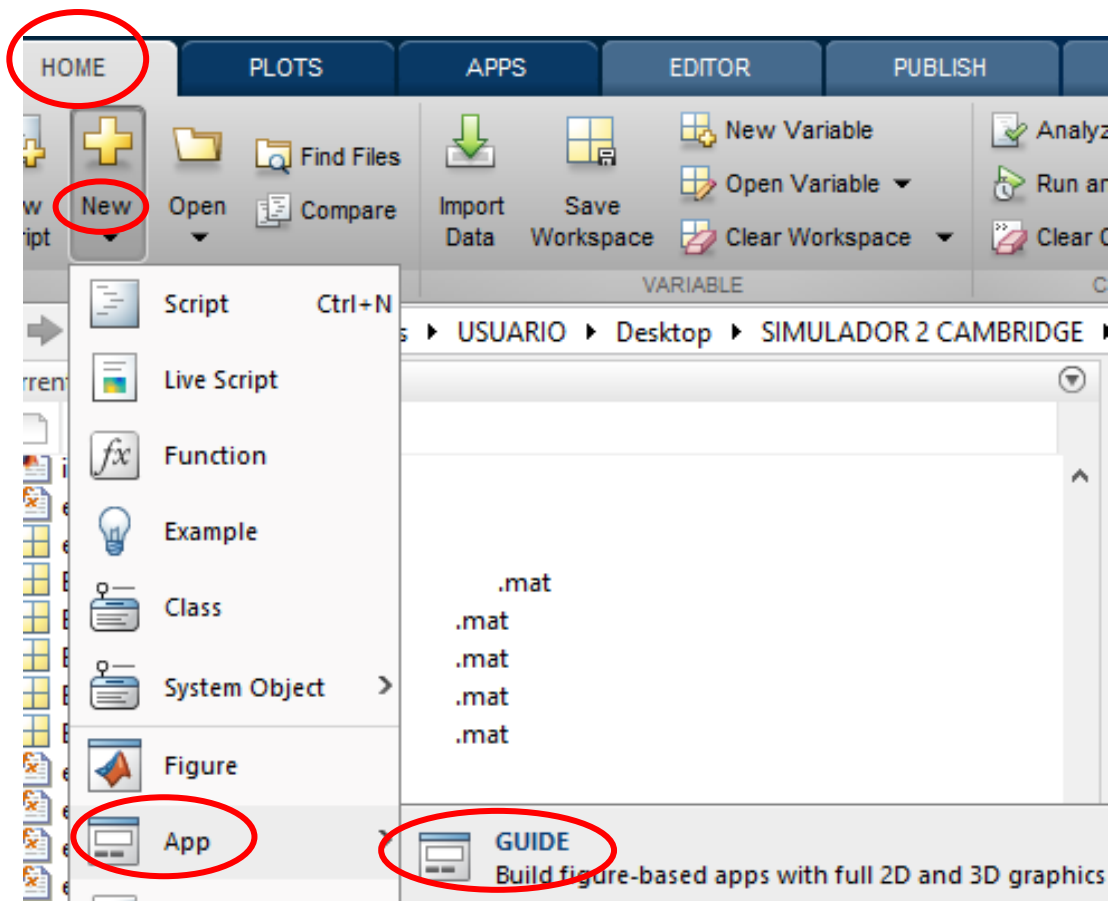


Figura 73. Iniciación de GUIDE

A continuación, aparecerá una pestaña como en la figura 74. Se debe de sustituir la palabra ejemplo, por el nombre que se le quiera asignara a la interfaz. Tras clicar OK, automáticamente se abrirá una pestaña (figura 75) y un código en MATLAB.

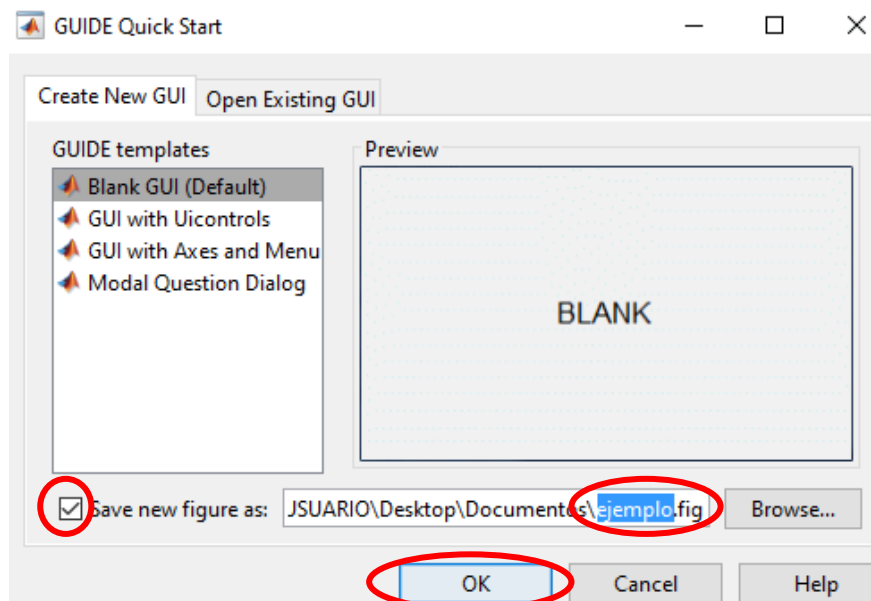


Figura 74. Nombre de la interfaz gráfica

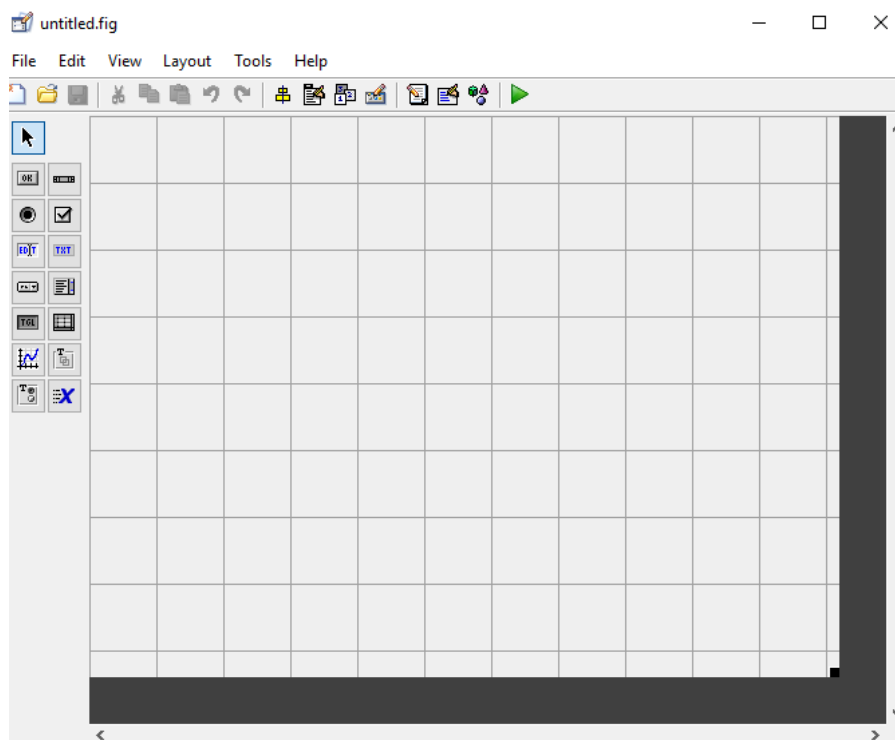


Figura 75. Interfaz gráfica en blanco

Para poder dibujar la interfaz se van a explicar los conceptos utilizados a lo largo de este proyecto (figura 76):

- **Edit text:** casilla en la que el usuario puede introducir los datos deseados.

- **Static text:** texto escrito por el programador para orientar al usuario a través de la interfaz gráfica.
- **Push button:** botón que al ser presionado por el usuario hará llamar a cierto código implementado por el programador.
- **Listbox:** lista en la que se podrán guardar y eliminar nombres a través de los botones.

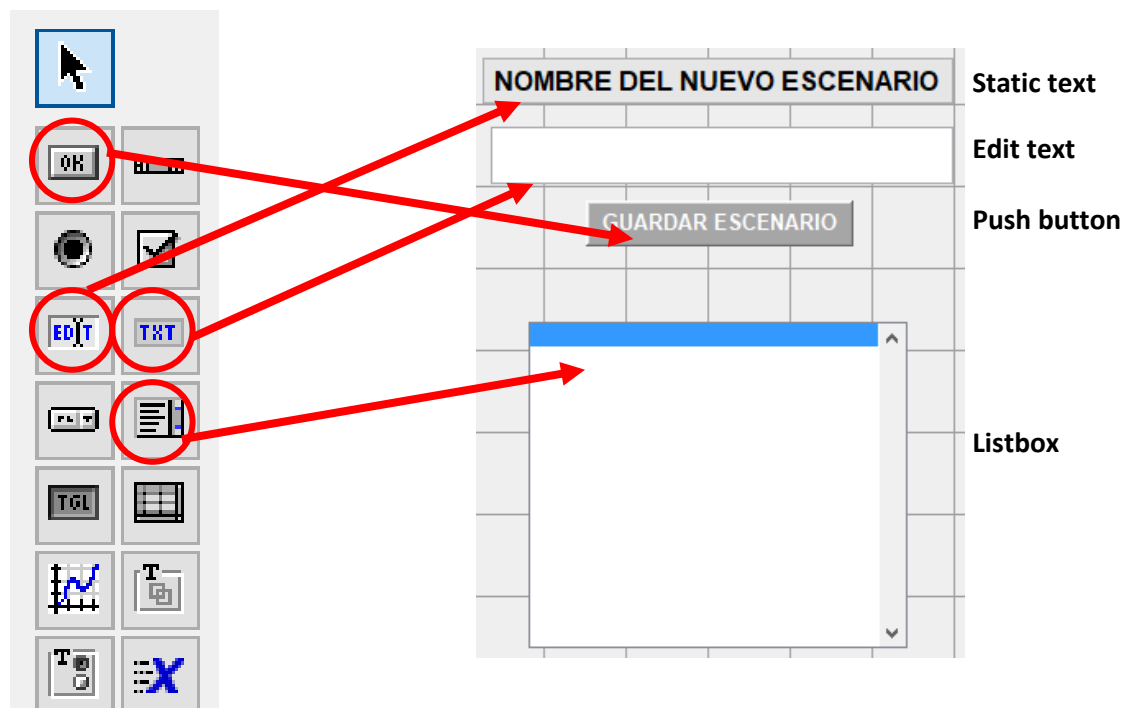


Figura 76. Elementos de la interfaz gráfica

Una vez colocados todos los elementos necesarios, se debe de clicar dos veces sobre el elemento que se quiera editar. A continuación, se abrirá una pestaña (figura 77) en la que se podrá cambiar el color, el tamaño de letra, el nombre de la casilla, etc.

La opción más importante es **Tag** ya que ahí se escribe el nombre por el cual se hará referencia a dicha casilla mediante el código.

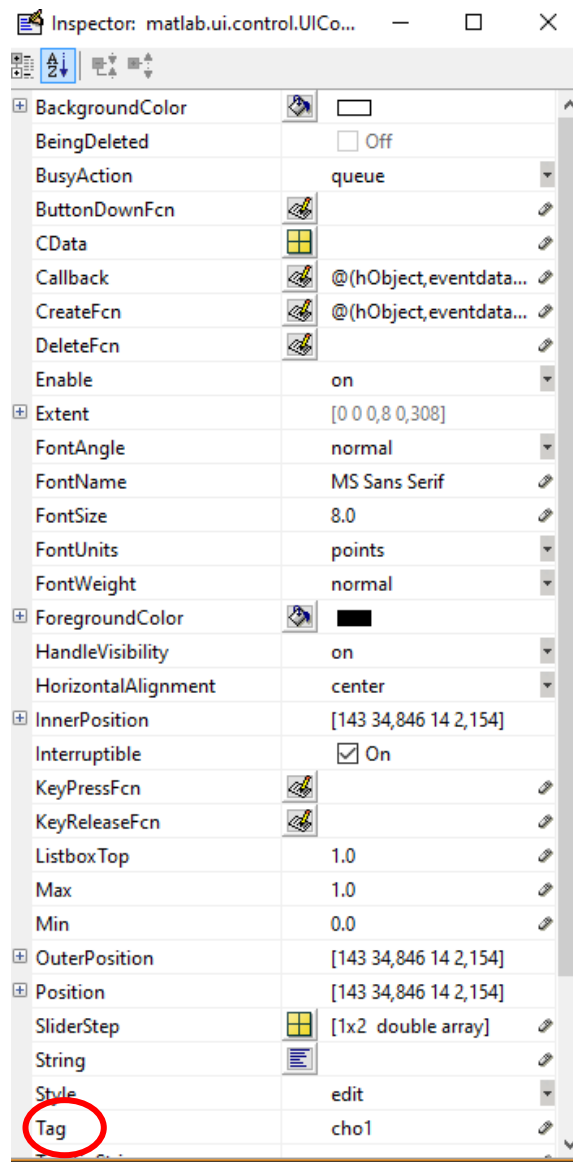


Figura 77. Editor de casillas

Tras haber dibujado la interfaz gráfica, el código se habrá rellenado casi por completo. Solo faltará ampliar la parte referida a los botones (se explica el código brevemente más adelante). Esto es debido a que el usuario solo podrá realizar acciones a través de los botones.

Si por el contrario se desea abrir la interfaz propuesta (entrada_datos.fig), se visualizará en pantalla la figura 78. Junto a entrada_datos.fig debe de abrirse el código entrada_datos.m.

NOMBRE DEL NUEVO ESCENARIO

GUARDAR ESCENARIO

MOstrar ESCENARIOS

MOstrar VARIABLES DEL ESCENARIO

ELIMINAR ESCENARIO

PERSONA CON DIABETES TIPO1

	Hora [0-23]	Minutos [0-59]	CHO (g)	Hora [0-23]	Minutos [0-59]	Bolo (U/min)
DESAYUNO	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
COMIDA	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
CENA	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	Hora [0-23]	Minutos [0-59]	Duración (min)	Peso (Kg)	Sensibilidad [0-1]	Basal (U/min)
EJERCICIO	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

RESTAURAR ÚLTIMOS DATOS **ELIMINAR VARIABLES** **START**

Figura 78. Interfaz gráfica de GUIDE

Algunas de las pautas más importantes para la realización del código son:

- **set (handles.c_peso,'string', v_peso):** muestra en la casilla c.peso el dato guardado en la variable v_peso.
- **set (handles.peso,'BackgroundColor', [0.9 0.9 0.9]):** colorea de gris el fondo de la casilla peso. Si se deseara utilizar el color rojo sería: [1 0 0].
- **v_peso=str2double (get (handles.c_peso,'string')):** guarda en la variable v_peso el dato introducido por el usuario en la casilla c_peso. Antes de guardarlo pasa el dato de string a double.
- **exist ('variables.mat'):** comprueba si existe el archivo variables.mat. Si no existe, dará como resultado un cero.
- **load ('variables'):** carga los datos guardados en el archivo llamado variables.
- Al utilizar listas, muchas veces se desea saber el elemento seleccionado por el usuario. Dicha acción se representaría de la siguiente manera:

lista = string (get (handles.lista, 'string')): guarda en la variable lista las palabras escritas en la casilla lista.

item_numero=get (handles.lista, 'value'): si el usuario ha seleccionado el segundo elemento de la lista, en ítem_numero se guardará el número 2.

item_nombre =lista (item_numero): en item_nombre se guarda la palabra escrita en la segunda casilla de la lista, es decir, la palabra seleccionada por el usuario.

item_nombre=char (item_nombre): convierte la variable item_nombre a tipo char.

- **isempty (item_nombre):** devuelve un 1 si la variable item_nombre está vacía.
- **lista_nueva=vertcat (lista_encima, lista_debajo):** concatena verticalmente los elementos guardados en lista_encima y lista_debajo y se almacena todo ello en lista_nueva.
- **delete ([item_nombre,'.mat']):** elimina el archivo item_nombre.mat.
- **save escenarios lista:** guarda en un archivo llamado escenarios con extensión .mat la variable lista.

Lo más importante es que en el botón START, debe de aparecer la función que llamará al simulador y al script programado a través de MATLAB. Este último se llama prueba y se explica más adelante.

Interfaz gráfica 3D

Una vez se tenga el simulador y la interfaz gráfica para introducir los datos de entrada, se crea la escena 3D en Factory IO. Se querrá conseguir una interfaz como la de la siguiente figura:



Figura 79. Interfaz gráfica de la nave industrial 3D

Factory IO es un programa de simulación de naves industriales. Por ello, el espacio de la interfaz y los elementos usados son propios de la industria.

En esta ocasión, también se podrá elegir entre crear una escena nueva o abrir la que ya se ha implementado a lo largo de este proyecto.

Para abrir una escena nueva, lo único que hay que hacer es acudir con el ratón del ordenador a la flecha situada junto a la palabra WELCOME, figura 80.

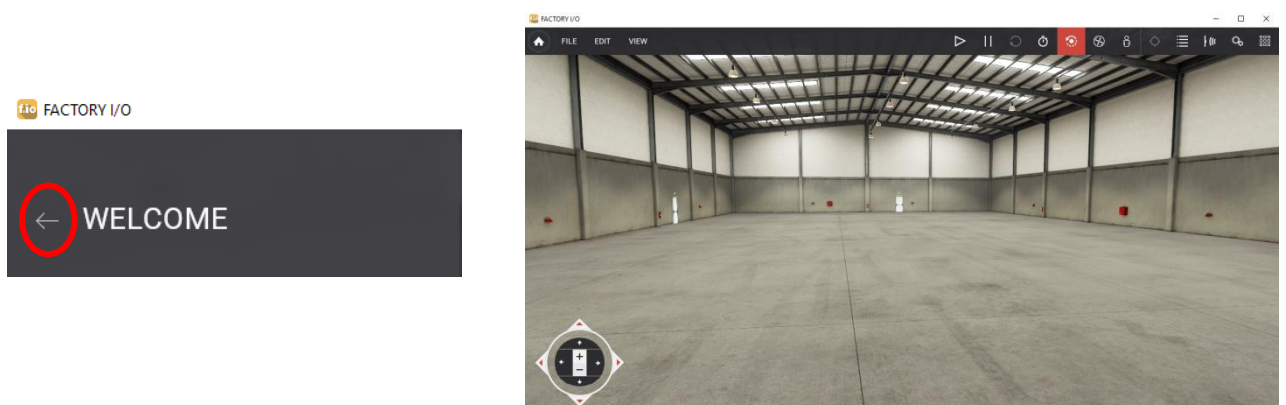


Figura 80. Creación de una nueva escena 3D

Si por el contrario se abre la interfaz disponible, se visualizarán los elementos comentados en el capítulo 6, apartado 6.2.1 Elementos.

A la hora de crear la escena se debe de tener en cuenta el nombre que le asocia Factory IO a cada elemento ya que los enumera según el orden de colocación. En la escena creada para este proyecto se siguió el siguiente orden:

- Los tanques de insulina son numerados del 1 al 3 de izquierda a derecha.
- El tanque de la glucosa es el número 4.
- Los emisores y eliminadores de la misma cinta tienen la misma numeración. Por ello, en la figura 81 solo aparecen los números de los emisores.
- Las cintas transportadoras pueden ser inclinadas u horizontales de 2 metros o de 4 metros. Las inclinadas tienen la misma numeración que los emisores. Las horizontales como solo hay una de cada tipo, ambas tienen el número 1.
- Los canales de transporte al no tener ni actuador ni sensor, es independiente el orden de colocación.
- En cuanto a la colocación de las luces y alarmas sólo importa saber las que están puestas hacia arriba y las que están hacia abajo.

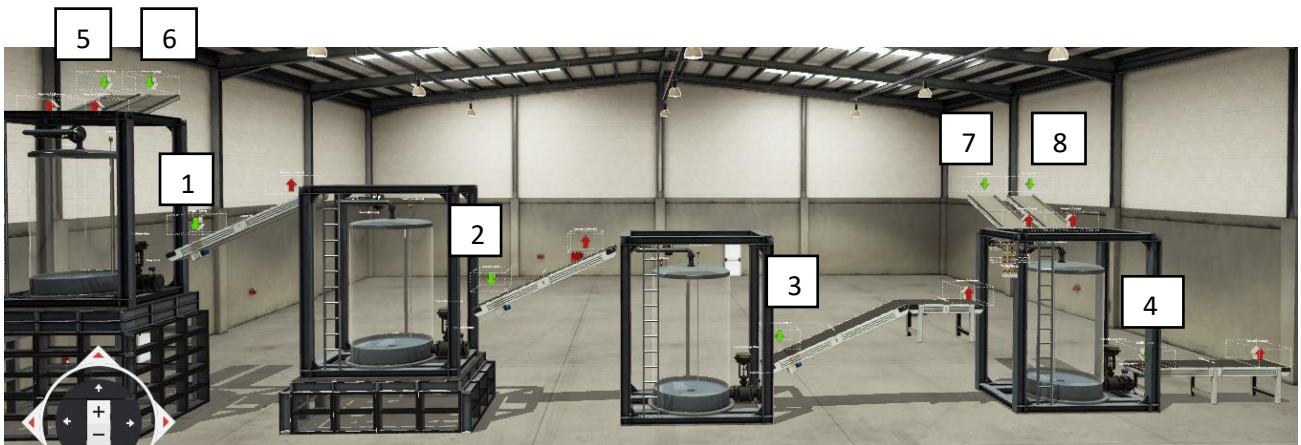


Figura 81. Numeración según Factory IO

En este proyecto se ha utilizado la siguiente nomenclatura para los sensores y actuadores:

- **Referidos a los tanques:**

Variables tipo **float**, de 0 a 10 voltios.

llenark: apertura de la válvula de llenado del tanque.

vaciark: apertura de la válvula de descarga.

nivelk: nivel del fluido en el interior del tanque.

caudalk: caudal de salida del fluido.

Siendo:

k=1, 2, 3 o 4 dependiendo de si se hace referencia al primer, segundo, tercer o cuarto tanque.

- **Referidos a las cintas transportadoras :**

Variables booleanas.

cintak: cinta situada en el número k de la figura 81.

cintaA: cinta horizontal de 2m.

cintaB: cinta horizontal de 4m.

Siendo:

k=1, 2 o 3.

- **Referidos a los emisores y eliminadores:**

Variables booleanas.

producirk: emisor situado en el número k de la figura 81.

eliminark: eliminador situado en el número k de la figura 81.

Siendo:

k: un número natural entre 1 y 8 (ambos incluidos).

- **Referidos a las luces y alarmas:**

Variables booleanas.

luzrk: luz roja.

luzak: luz amarilla.

luzvk: luz verde.

alarmaj: alarma.

Siendo:

k: un número natural entre 1 y 6 (ambos incluidos).

j = 1, 2 o 3.

- **Referidos al botón de reset y run de Factory IO:**

Variables booleanas.

reset: variable que reinicia la interfaz gráfica 3D.

run: variable que ejecuta la interfaz gráfica 3D.

Comunicaciones

Este apartado está desarrollado en el capítulo 6, apartado 6.4.2 Comunicaciones.

Ecuaciones para la equivalencia entre modelos dinámicos

En este apartado se explican algunas ecuaciones que se utilizarán en el código. Prácticamente sirven para saber cómo cambiar de unidades ya que el simulador resuelve las ecuaciones con el modelo de Cambridge y lo que se debe de introducir en Factory IO son voltios.

Caudales de los tanques de insulina:

- **Compartimento de entrada (tanque 1):**

Caudal de entrada:

El caudal que entra a este primer compartimento es la insulina basal y los bolos.
Si se inyecta un bolo:

$$Q_1entra \left(\frac{U}{s}\right) = IBasal \left(\frac{U}{s}\right) + Ibolos \left(\frac{U}{s}\right) \quad (IV.1)$$

Si no se inyecta un bolo:

$$Q_1entra \left(\frac{U}{s}\right) = IBasal \left(\frac{U}{s}\right) \quad (IV.2)$$

Para realizar los cambios de unidades se debe de hacer lo siguiente:

1. Pasar las unidades a **m³/s**:

La variable MaxU (U) es la máxima insulina que se puede introducir en el tanque. Además, si el depósito está lleno, se ocupará todo el volumen del tanque. Por lo tanto:

$$Volumen\ tanque = \pi * r^2 * h = \pi * 1^2 * 3 = 3 * \pi\ m^3 \quad (IV.3)$$

$$MaxU(U) = Volumentanque = 3 * \pi\ m^3 \quad (IV.4)$$

Si el caudal máximo de entrada en los tanques es de 0.25 m³/s, como mucho se podrán inyectar IMaxU (U/s) de insulina:

$$IMax \left(\frac{U}{s} \right) = \frac{MaxU (U) * 0.25 \left(\frac{m^3}{s} \right)}{3 * \pi (m^3)} \quad (IV.5)$$

Por lo tanto, el caudal de entrada será:

$$Q_1 entrada \left(\frac{m^3}{s} \right) = \frac{Q_1 entra \left(\frac{U}{s} \right) * 0.25 \left(\frac{m^3}{s} \right)}{IMax \left(\frac{U}{s} \right)} \quad (IV.6)$$

2. Pasar de m³/s a V:

Si se introduce el máximo caudal (0.25 m³/s) en Factory IO, se deben de escribir 10 V.

$$Q_1 entra (V) = \frac{Q_1 entra \left(\frac{m^3}{s} \right) * 10 (V)}{0.25 \left(\frac{m^3}{s} \right)} \quad (IV.7)$$

Se podrían agrupar las ecuaciones, pero se ha decidido escribirlas separadas porque de esta forma se puede ver la variable del caudal en las distintas unidades.

Caudal de salida:

$$Q_1 sale \left(\frac{U}{s} \right) = \frac{S_1}{t_{maxI} * 60} \quad (IV.8)$$

$$Q_1 sale \left(\frac{m^3}{s} \right) = \frac{Q_1 sale \left(\frac{U}{s} \right) * 0.25 \left(\frac{m^3}{s} \right)}{IMax \left(\frac{U}{s} \right)} \quad (IV.9)$$

$$Q_{1sale} (V) = \frac{Q_{1sale} \left(\frac{m^3}{s}\right) * 10 (V)}{0.3543 \left(\frac{m^3}{s}\right)} \quad (IV.10)$$

Independientemente del tanque, para cambiar de unidades se hace como se acaba de explicar. Por lo tanto, en las siguientes ecuaciones, este cambio de unidades se omite.

- **Compartimento intermedio (tanque 2):**

$$Q_{2entra} \left(\frac{U}{s}\right) = \frac{S_1}{t_{maxI} * 60} \quad (IV.11)$$

$$Q_{2sale} \left(\frac{U}{s}\right) = \frac{S_2}{t_{maxI} * 60} \quad (IV.12)$$

- **Insulina en plasma (tanque 3)**

$$Q_{3entra} \left(\frac{U}{s}\right) = \frac{S1}{t_{maxI} * Vi * 60} \quad (IV.13)$$

$$Q_{3sale} \left(\frac{U}{s}\right) = \frac{ke * I}{60} \quad (IV.14)$$

Caudales del tanque de glucosa

Caudal de entrada:

$$Q_{4entra} \left(\frac{mmol}{L * s}\right) = \frac{k_{12} * Q_2 + Ug + EGP_0}{Vg * 60} \quad (IV.15)$$

Para pasar de mmol/L a mg/dl debe de hacerse lo siguiente:

1. Pasar los mmol a mg:

$$m(g) = n(mol) * Mm\left(\frac{g}{mol}\right) \quad (IV.16)$$

$$Mm(glucosa) = Mm(C_6H_{12}O_6) = 12 * 6 + 12 * 1 + 16 * 6 = 180 \frac{g}{mol} \quad (IV.17)$$

Por lo tanto:

$$m = 10^{-3} * 180 = 0.18 g = 180 mg \quad (IV.18)$$

2. Pasar los L a dl (dividir entre 10).
3. Juntando el paso 1 y el 2:

$$Q_4 \text{ entra } \left(\frac{mg}{dl * s}\right) = Q_4 \text{ entra } \left(\frac{mmol}{L} * s\right) * 18 \quad (IV.19)$$

Factory IO no admite introducir las variables en estas unidades, por ello se debe de realizar lo que viene a continuación:

1. Pasar las unidades a m³/s:
Se supone que 300 mg/dl es la máxima cantidad de glucosa que puede haber en el tanque. Si se llena el tanque por completo, habrá 300 mg/dl y todo el volumen del tanque estará ocupado. Por lo tanto:

$$300 \left(\frac{mg}{dl}\right) = Volumen \ tanque = 3 * \pi \ m^3 \quad (IV.20)$$

Además, si el caudal máximo de entrada en los tanques es de 0.25 m³/s, como mucho se podrán inyectar MaxG mg/dl de glucosa:

$$MaxG \left(\frac{mg}{dl * s}\right) = \frac{300 \left(\frac{mg}{dl}\right) * 0.25 \left(\frac{m^3}{s}\right)}{3 * \pi \ (m^3)} \quad (IV.21)$$

Por lo tanto, el caudal de entrada será:

$$Q_{4\text{entrada}} \left(\frac{\text{m}^3}{\text{s}} \right) = \frac{Q_{4\text{entra}} \left(\frac{\text{mg}}{\text{dl} \cdot \text{s}} \right) * 0.25 \left(\frac{\text{m}^3}{\text{s}} \right)}{\text{MaxG} \left(\frac{\text{mg}}{\text{dl} \cdot \text{s}} \right)} \quad (\text{IV.22})$$

2. Pasar de m^3/s a V:

Si se introduce el máximo caudal ($0.25 \text{ m}^3/\text{s}$) en Factory IO, se deben de introducir 10 V.

Por lo tanto:

$$Q_{4\text{entra}} (V) = \frac{Q_{4\text{entra}} \left(\frac{\text{m}^3}{\text{s}} \right) * 10 (V)}{0.25 \left(\frac{\text{m}^3}{\text{s}} \right)} \quad (\text{IV.23})$$

Se podrían agrupar las ecuaciones, pero se ha decidido ponerlas separadas porque de esta forma se puede ver la variable del caudal en las distintas unidades.

Caudal de salida:

$$Q_{4\text{sale}} \left(\frac{\text{mmol}}{\text{L} \cdot \text{s}} \right) = \frac{F_{01}^c + x_1 * Q_1 + Fr + EGP_0 * x_3}{Vg * 60} \quad (\text{IV.24})$$

$$Q_{4\text{sale}} \left(\frac{\text{mg}}{\text{dl} \cdot \text{s}} \right) = Q_{4\text{sale}} \left(\frac{\text{mmol}}{\text{L}} * \text{s} \right) * 18 \quad (\text{IV.25})$$

$$Q_{4\text{sale}} \left(\frac{\text{m}^3}{\text{s}} \right) = \frac{Q_{4\text{sale}} \left(\frac{\text{mg}}{\text{dl} \cdot \text{s}} \right) * 0.25 \left(\frac{\text{m}^3}{\text{s}} \right)}{\text{MaxG} \left(\frac{\text{mg}}{\text{dl} \cdot \text{s}} \right)} \quad (\text{IV.26})$$

$$Q_{4sale} (V) = \frac{Q_{4sale} \left(\frac{m^3}{s}\right) * 10 (V)}{0.3543 \left(\frac{m^3}{s}\right)} \quad (IV.27)$$

Al cambiar de unidades, la única diferencia es que en el caudal máximo de salida, al introducir 10 V salen 0.3543 m³/s en vez de 0.25 m³/s.

Apertura de las válvulas de entrada y salida de los tanques

Para llenar o vaciar los tanques en la interfaz gráfica se necesita saber la apertura de la válvula.

- **Apertura de las válvulas de entrada:** es el número que ha salido al pasar el caudal de entrada a voltios.
- **Apertura de las válvulas de salida:** al ser proporcional con el nivel del tanque, debe de hacerse lo siguiente:

$$Q_{calculado} = x * Q_{real} \quad (IV.28)$$

Donde:

Qcalculado: caudal de salida calculado a través de las ecuaciones anteriores.

x: apertura de la válvula de salida.

Qreal: caudal de salida real.

$$Q_{real} = k * \sqrt{H} \quad (IV.29)$$

Donde:

Q: caudal de salida.

k: constante.

H: nivel del líquido en el interior del tanque en V.

Si se combinan ambas ecuaciones resulta:

$$Q_{calculado} = x * Q_{real} = x * k * \sqrt{H} \quad (IV.30)$$

Por lo tanto, la variable que se debe de introducir en la interfaz gráfica es:

$$x = \frac{Q_{calculado}}{k * \sqrt{H}} \quad (IV.31)$$

No se han mencionado las unidades porque se va a calcular el valor de k para poder introducir todos los datos en voltios. De esta manera, el resultado se podrá enviar directamente a Factory IO:

Para la obtención de la constante k se obtuvieron cuatro puntos (tabla 10):

Nivel (V)	Caudal (V)
8	8.9171
6	7.6994
4	6.3021
2	4.4503

Tabla 10. Puntos para el cálculo de la constante k

Sustituyendo los cuatro puntos en la ecuación IV.31, se obtiene un valor medio de **k = 3.1484**.

Por lo tanto, ya se puede obtener el la apertura de válvula de salida a través de la ecuación:

$$x = \frac{Q_{calculado} (V)}{3.1484 * \sqrt{H} (V)} \quad (IV.31)$$

Se deberá de multiplicar por 10 ya que da el valor en tanto por uno y en el programa debe de ser un número entre 0 y 10.

A continuación, se muestran tres tablas que resumen las ecuaciones anteriores:

	$\frac{U}{s}$	$\frac{m^3}{s}$	V
Q_{1entra}	$I_{Basal} \left(\frac{U}{s}\right) + I_{bolo} \left(\frac{U}{s}\right)$	$\frac{Q_{1entra} \left(\frac{U}{s}\right) * 0.25 \left(\frac{m^3}{s}\right)}{I_{Max} \left(\frac{U}{s}\right)}$	$\frac{Q_{1entra} \left(\frac{m^3}{s}\right) * 10 (V)}{0.25 \left(\frac{m^3}{s}\right)}$
Q_{2entra}	$\frac{S_1}{t_{maxI} * 60}$	$\frac{Q_{2entra} \left(\frac{U}{s}\right) * 0.25 \left(\frac{m^3}{s}\right)}{I_{Max} \left(\frac{U}{s}\right)}$	$\frac{Q_{2entra} \left(\frac{m^3}{s}\right) * 10 (V)}{0.25 \left(\frac{m^3}{s}\right)}$
Q_{3entra}	$\frac{S1}{t_{maxI} * Vi * 60}$	$\frac{Q_{3entra} \left(\frac{U}{s}\right) * 0.25 \left(\frac{m^3}{s}\right)}{I_{Max} \left(\frac{U}{s}\right)}$	$\frac{Q_{3entra} \left(\frac{m^3}{s}\right) * 10 (V)}{0.25 \left(\frac{m^3}{s}\right)}$

Tabla 11. Caudales de entrada de la insulina

Q_{1sale}	$\frac{S_1}{t_{maxI} * 60}$	$\frac{Q_{1sale} \left(\frac{U}{s}\right) * 0.25 \left(\frac{m^3}{s}\right)}{I_{Max} \left(\frac{U}{s}\right)}$	$\frac{Q_{1sale} \left(\frac{m^3}{s}\right) * 10 (V)}{0.3543 \left(\frac{m^3}{s}\right)}$
Q_{2sale}	$\frac{S_2}{t_{maxI} * 60}$	$\frac{Q_{2sale} \left(\frac{U}{s}\right) * 0.25 \left(\frac{m^3}{s}\right)}{I_{Max} \left(\frac{U}{s}\right)}$	$\frac{Q_{2sale} \left(\frac{m^3}{s}\right) * 10 (V)}{0.3543 \left(\frac{m^3}{s}\right)}$
Q_{3sale}	$\frac{ke * I}{60}$	$\frac{Q_{3sale} \left(\frac{U}{s}\right) * 0.25 \left(\frac{m^3}{s}\right)}{I_{Max} \left(\frac{U}{s}\right)}$	$\frac{Q_{3sale} \left(\frac{m^3}{s}\right) * 10 (V)}{0.3543 \left(\frac{m^3}{s}\right)}$

Tabla 12. Caudales de salida de la insulina

	$\frac{mmol}{L * s}$	$\frac{mg}{dl * s}$	$\frac{m^3}{s}$	V
Q_4entra	$\frac{k_{12} * Q_2 + Ug + EGP_0}{Vg * 60}$	$Q_4entra \left(\frac{mmol}{L} * s \right) * 18$	$\frac{Q_4entra \left(\frac{mg}{dl * s} \right) * 0.25 \left(\frac{m^3}{s} \right)}{MaxG \left(\frac{mg}{dl * s} \right)}$	$\frac{Q_4entra \left(\frac{m^3}{s} \right) * 10 (V)}{0.25 \left(\frac{m^3}{s} \right)}$
Q_4sale	$\frac{F_{01}^c + x_1 * Q_1 + Fr + EGP_0 * x_3}{Vg * 60}$	$Q_4sale \left(\frac{mmol}{L} * s \right) * 18$	$\frac{Q_4sale \left(\frac{mg}{dl * s} \right) * 0.25 \left(\frac{m^3}{s} \right)}{MaxG \left(\frac{mg}{dl * s} \right)}$	$\frac{Q_4sale \left(\frac{m^3}{s} \right) * 10 (V)}{0.3543 \left(\frac{m^3}{s} \right)}$

Tabla 13. Caudales de entrada y salida de la glucosa

Algoritmos de intercambio de información

MATLAB actuará como cerebro de la aplicación y será quien le indique a la interfaz 3D lo que debe de hacer. Para ello, aparte de las dos funciones realizadas para escribir y leer las variables, se necesitan los siguientes archivos:

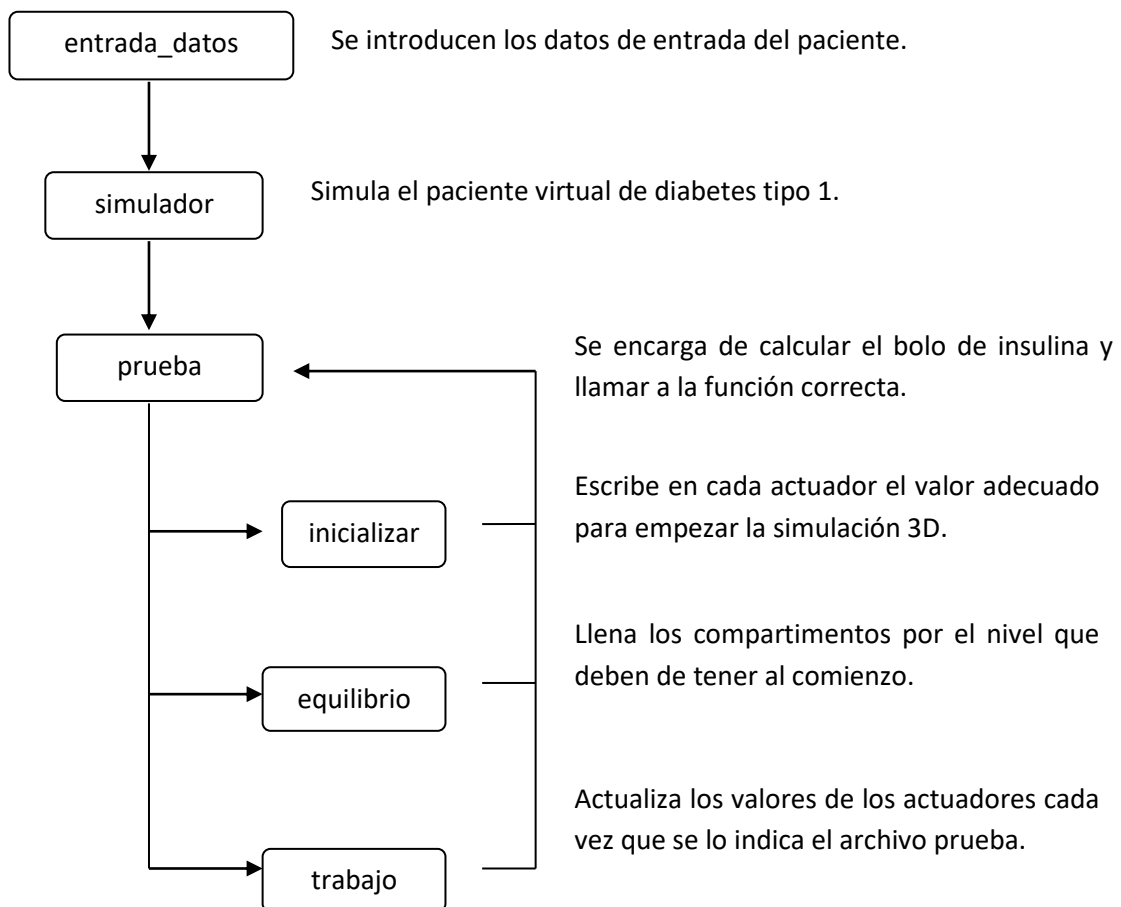


Figura 82. Archivos de MATLAB

A continuación, se explican los archivos más detalladamente:

inicializar: En cada línea se llama a la función escribir pasándole dos parámetros. La estructura es la siguiente:

escribir ('Channel1.Device1.Group1.variable', dato)



En el siguiente ejemplo, se quiere llenar lo más rápido posible el tanque 1. Como la variable encargada de realizar dicha acción es llenar1, se tiene que introducir como dato un 10.

escribir ('Channel1.Device1.Group1.llenar1 ', 10)

equilibrio: archivo llamado a través de prueba. Se le pasan cuatro variables: niveldeseadoS1, niveldeseadoS2, niveldeseadoI y niveldeseadoGlucosa. Cada uno es el nivel al que debe de llegar cada tanque nada más empezar la simulación. Es necesaria esta función ya que sino el paciente simulado empezaría con 0 U de insulina y no tendría glucosa en el organismo.

Para llegar a los niveles deseados se realiza lo siguiente:

- Se empiezan a llenar los 4 tanques a máximo caudal.
- Se lee el nivel de cada tanque.
- Si algún compartimento ha llegado al nivel deseado, se deja de llenar.

trabajo: función encargada de actualizar los valores en las variables y enviarlas a la interfaz gráfica 3D. Para ello necesita que el archivo prueba le envíe las siguientes variables:

resultado: almacena el nombre del archivo en el que el simulador guardó los resultados.

fila: fila de la matriz X que debe de leerse.

bolo: es una variable booleana. Valdrá 1 cuando se tenga que inyectar un bolo de insulina.

lboloU: las U/s que deben de inyectarse del bolo de insulina.

MaxU: máximo valor que pueden alcanzar los dos primeros tanques de insulina.

S1U, S2U e IU: son los vectores de las unidades de insulina que almacena el tanque 1, tanque 2 y tanque 3 respectivamente en cada simulación.

Los pasos más importantes a seguir son los siguientes:

- Cargar los datos del simulador y se da el nombre adecuado a cada variable (según el modelo de Cambridge).
- Calcular las variables necesarias para obtener el caudal de entrada y salida del compartimento de la glucosa, (figura 83).
- Multiplica por 18 para pasar de mmol/(L*min) a mg/(dl*min).


```
27 % Caudal de entrada y salida de la glucosa (mg/(s*dL)).
28 - G=Q1(fila)/vg;
29
30 - if G>=4.5
31 -     F01c=f01
32 - else
33 -     F01c=(f01*G)/4.5
34 - end
35
36 - if G>=9
37 -     Fr=0.003*(G-9)*vg;
38 - else
39 -     Fr=0;
40 - end
41
42 - Ug=g2(fila)/tmaxG;
43
44
45 - Q4entra=(k12*Q2(fila)+Ug+egp0)/(60*vg);
46 - Q4sale=(F01c+X1(fila)*Q1(fila)+Fr+egp0*X3(fila))/(60*vg);
47 - Q4entra=Q4entra*18;
48 - Q4sale=Q4sale*18;
```

Figura 83. Función trabajo 1

- Cambiar las unidades del caudal de entrada y salida de la glucosa a voltios.
- Con los caudales en voltios ya se puede calcular la apertura de la válvula de llenado y vaciado.
- Calcular si los emisores de las cintas transportadoras de la glucosa deben de activarse o no.
- Obtener el caudal de entrada y salida de los tres compartimentos de insulina.
- Pasar las unidades a voltios.
- Calcular la apertura de la válvula de cada tanque.
- Calcular si los emisores de las cintas transportadoras de la insulina deben de activarse o no.
- Leer el nivel de la glucosa para saber qué luces deben de encenderse y si las alarmas deben de estar apagadas o encendidas (figura 84 y 85).
- Enviar los valores calculados a KEPServer para que este los envíe a Factory IO, (figura 86).

```
193     % NIVEL DE GLUCOSA
194     [nivel4]=leer('Channell.Devicel.Group1.nivel4');
195     while (isempty(nivel4)==1)
196         [nivel4]=leer('Channell.Devicel.Group1.nivel4');
197     end
198
199     NIVEL4=(nivel4*300)/10
200
201     luz_verde=0;
202     luz_amarilla_arriba=0;
203     luz_amarilla_abajo=0;
204     luz_roja_arriba=0;
205     luz_roja_abajo=0;
206     alarma=0;
```

Figura 84. Función trabajo 2

```
207
208     if NIVEL4>=250
209         luz_roja_arriba=1;
210
211     elseif NIVEL4>=180 && NIVEL4<250
212         luz_amarilla_arriba=1;
213
214     elseif NIVEL4>=70 && NIVEL4<180
215         luz_verde=1;
216
217     elseif NIVEL4>=54 && NIVEL4<70
218         luz_amarilla_abajo=1;
219
220     elseif NIVEL4<54
221         luz_roja_abajo=1;
222         alarma=1;
223     end
224
```

Figura 85. Función trabajo 3

```
223 % Se pasan los valores al OPC:
224 - escribir('Channell.Devicel.Groupl.llenar1',valorllenar1)
225 - escribir('Channell.Devicel.Groupl.vaciar1',valorvaciar1)
226 - escribir('Channell.Devicel.Groupl.llenar2',valorllenar2)
227 - escribir('Channell.Devicel.Groupl.vaciar2',valorvaciar2)
228 - escribir('Channell.Devicel.Groupl.llenar3',valorllenar3)
229 - escribir('Channell.Devicel.Groupl.vaciar3',valorvaciar3)
230 - escribir('Channell.Devicel.Groupl.llenar4',valorllenar4)
231 - escribir('Channell.Devicel.Groupl.vaciar4',valorvaciar4)
232
233 - escribir('Channell.Devicel.Groupl.producir1',producir1)
234 - escribir('Channell.Devicel.Groupl.producir2',producir2)
235 - escribir('Channell.Devicel.Groupl.producir3',producir3)
236 - escribir('Channell.Devicel.Groupl.producir4',producir4)
237 - escribir('Channell.Devicel.Groupl.producir5',producir5)
238 - escribir('Channell.Devicel.Groupl.producir6',producir6)
239 - escribir('Channell.Devicel.Groupl.producir7',producir7)
240 - escribir('Channell.Devicel.Groupl.producir8',producir8)
241
242 - escribir('Channell.Devicel.Groupl.luzr1',luz_roja_arriba)
243 - escribir('Channell.Devicel.Groupl.luzr2',luz_roja_abajo)
244 - escribir('Channell.Devicel.Groupl.luzr3',luz_roja_arriba)
245 - escribir('Channell.Devicel.Groupl.luzr4',luz_roja_abajo)
246 - escribir('Channell.Devicel.Groupl.luzr5',luz_roja_arriba)
247 - escribir('Channell.Devicel.Groupl.luzr6',luz_roja_abajo)
248 -
249 - escribir('Channell.Devicel.Groupl.luzv1',luz_verde)
250 - escribir('Channell.Devicel.Groupl.luzv2',luz_verde)
251 - escribir('Channell.Devicel.Groupl.luzv3',luz_verde)
252 - escribir('Channell.Devicel.Groupl.luzv4',luz_verde)
253 - escribir('Channell.Devicel.Groupl.luzv5',luz_verde)
254 - escribir('Channell.Devicel.Groupl.luzv6',luz_verde)
255 - escribir('Channell.Devicel.Groupl.luza1',luz_amarilla_arriba)
256 - escribir('Channell.Devicel.Groupl.luza2',luz_amarilla_abajo)
257 - escribir('Channell.Devicel.Groupl.luza3',luz_amarilla_arriba)
258 - escribir('Channell.Devicel.Groupl.luza4',luz_amarilla_abajo)
259 - escribir('Channell.Devicel.Groupl.luza5',luz_amarilla_arriba)
260 - escribir('Channell.Devicel.Groupl.luza6',luz_amarilla_abajo)
261 - escribir('Channell.Devicel.Groupl.alarma1',alarma)
262 - escribir('Channell.Devicel.Groupl.alarma2',alarma)
263 - escribir('Channell.Devicel.Groupl.alarma3',alarma)
264 -
265 -
266 -
267 -
268 -
269 -
270 -
271 -
272 -
273 -
274 -
275 -
276 -
277 -
278 -
279 -
280 -
281 -
282 -
283 -
284 -
285 -
286 -
287 -
288 -
289 -
290 -
291 -
292 -
293 -
294 -
295 -
296 -
297 -
298 -
299 -
300 -
301 -
302 -
303 -
304 -
305 -
306 -
307 -
308 -
309 -
310 -
311 -
312 -
313 -
314 -
315 -
316 -
317 -
318 -
319 -
320 -
321 -
322 -
323 -
324 -
325 -
326 -
327 -
328 -
329 -
330 -
331 -
332 -
333 -
334 -
335 -
336 -
337 -
338 -
339 -
340 -
341 -
342 -
343 -
344 -
345 -
346 -
347 -
348 -
349 -
350 -
351 -
352 -
353 -
354 -
355 -
356 -
357 -
358 -
359 -
360 -
361 -
362 -
363 -
364 -
365 -
366 -
367 -
368 -
369 -
370 -
371 -
372 -
373 -
374 -
375 -
376 -
377 -
378 -
379 -
380 -
381 -
382 -
383 -
384 -
385 -
386 -
387 -
388 -
389 -
390 -
391 -
392 -
393 -
394 -
395 -
396 -
397 -
398 -
399 -
400 -
```

Figura 86. Función trabajo 4

prueba: archivo encargado de llamar a las funciones previamente explicadas.

- Llamar a la función inicializar y esta hace su tarea.
- Resetear y ejecutar Factory IO.
- Cargar los resultados del simulador.
- Dar el nombre adecuado a cada variable (según el modelo de Cambridge) y pasar a las unidades más apropiadas.
- Calcular las máximas unidades de insulina que se van a tener en el primer y segundo compartimento a través de la variable MaxU (figura 87 y 88).
- Calcular los niveles deseados previamente explicados y llamar a la función equilibrio.
- Si se inyecta un bolo, se llamará a la función trabajo 10 veces. Esto es debido a que el simulador ha sido programado para coger resultados cada 0.1 minuto. Debería de haber una pausa de 0.1*60 segundos para que se estuviera realizando correctamente la inyección de insulina. Sin embargo, como se Factory IO tiene la velocidad de la interfaz acelerada, no es necesaria dicha pausa.
- Si no se tuviera que administrar el bolo, la función prueba llamaría a la función trabajo enviándole los mismos parámetros pero la variable bolo en vez de ser true, sería false.
- Cada vez que se llama a la función trabajo, se aumenta en uno la variable fila para que cada vez se lea la siguiente fila de la matriz que contiene los resultados del simulador (figura 89).

```
49      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
50      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%INSULINA%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
51      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
52
53      % Se calcula el máximo valor de S1 y S2 que será el máximo nivel de
54      % los tanques de insulina.
55
56      maxS1=0;
57      for i=1:SizeX
58          S1MaxU=S1U(i);
59          if S1MaxU>maxS1
60              maxS1=S1MaxU;
61          end
62      end
63      S1MaxU=maxS1;
64
65      maxS2=0;
66      for i=1:SizeX
67          S2MaxU=S2U(i);
68          if S2MaxU>maxS2
69              maxS2=S2MaxU;
70          end
71      end
72      S2MaxU=maxS2;
```

Figura 87. Script prueba 1

```
75 % Se halla el máximo entre S1MaxU y S2MaxU para usarlo como nivel máximo en
76 % los tanques S1, S2 e I/vi
77 - if S1MaxU>S2MaxU
78 -     MaxU=S1MaxU
79 - elseif S2MaxU>S1MaxU
80 -     MaxU=S2MaxU
81 - elseif S1MaxU==S2MaxU
82 -     MaxU=S1MaxU
83 - end
84
```

Figura 88. Script prueba 2

```
128
129 % bolo: indica si es true o false (si se inyecta un bolo de insulina o no).
130 % SizeBolo: indica el tamaño del vector bolus.
131 % IboloU: indica la cantidad de insulina que se inyecta en el bolo U/s.
132
133 - if bolo==true
134 -     c=1
135 -     fila=fila-1;
136 -     while c<=10
137 -         fila=fila+1;
138 -         trabajo(resultado,fila,bolo,IboloU,MaxU, S1U, S2U, IU)
139 -
140 -         c=c+1
141 -     end
142 - else
143 -     trabajo(resultado,fila,bolo,IboloU,MaxU, S1U, S2U, IU)
144 - end
145 - fila=fila+1
146 - end
147 -
```

Figura 89. Script prueba 3

PRESUPUESTO

A través de este apartado se pretende indicar el coste para el diseño e implementación de la herramienta educativa sobre la gestión de diabetes tipo 1.

Cuadro de precios

El trabajador, los materiales y los software que han intervenido en este proyecto con sus respectivos costes se indican en la siguiente tabla:

MANO DE OBRA

Código	Unidad	Descripción	Precio (€)
MO01	h	Ingerido Industrial con poca experiencia	17,00

MATERIALES

Código	Unidad	Descripción	Precio (€/unidad) o (€/mes)
MA01	ud	Ordenador con memoria RAM de 8 GB. Procesador Intel Core i7 de 3,1 GHz. Disco duro de 930 GB y Sistema Operativo Windows 10	850,00
MA02	ud	MATLAB (versión estudiante)	150,00
MA03	mes	Factory IO	144,00
MA04	ud	KEPServer	452,00
MA05	ud	OBS	0,00
MA06	mes	Premiere Pro (versión estudiante)	16,25

Unidades de obra. Precios descompuestos

Todas las tareas se agrupan en unidades de obra con sus respectivos costes.

Código	Unidad	Descripción	Cantidad	Precio unitario	Importe (€)
UD001	h	Instalación del software			2288,92
MA01	ud	Ordenador	1	850,00	850,00
MA02	ud	MATLAB	1	150,00	150,00
MA03	mes	Factory IO	5	144,00	720,00
MA04	ud	KEPServer	1	452,00	452,00
MA05	ud	OBS	1	0,00	0,00
MA06	mes	Premiere Pro	1	16,25	16,25
MO01	h	Ingeniero Industrial	2	17,00	34,00
	%	Costes directos complementarios	0,03	2222,25	66,67

DISEÑO E IMPLEMENTACIÓN DE UNA HERRAMIENTA EDUCATIVA SOBRE LA GESTIÓN DE DIABETES
TIPO 1 EMPLEANDO HERRAMIENTAS DE SIMULACIÓN 3D DE PROCESOS INDUSTRIALES

UD002	h	Documentación del simulador			87,55
MO01	h	Ingeniero Industrial	5	17,00	85,00
	%	Costes directos complementarios	0,03	85,00	2,55
UD003	h	Implementación de la interfaz gráfica para el simulador			210,12
MO01	h	Ingeniero Industrial	12	17,00	204,00
	%	Costes directos complementarios	0,03	204,00	6,12
UD004	h	Diseño y desarrollo en MATLAB			2626,50
MO01	h	Ingeniero Industrial	150	17,00	2550,00
	%	Costes directos complementarios	0,03	2550,00	76,50
UD005	h	Diseño y desarrollo en KEPServer			140,08
MO01	h	Ingeniero Industrial	8	17,00	136,00
	%	Costes directos complementarios	0,03	136,00	4,08
UD006	h	Diseño y desarrollo en Factory IO			840,48
MO01	h	Ingeniero Industrial	48	17,00	816,00
	%	Costes directos complementarios	0,03	816,00	24,48
UD007	h	Grabación y edición de vídeos en OBS y Premiere			35,02
MO01	h	Ingeniero Industrial	2	17,00	34,00
	%	Costes directos complementarios	0,03	34,00	1,02

Precios unitarios

A las unidades creadas previamente se le deben de añadir los costes directos e indirectos:

Código	Unidad	Descripción	Cantidad	Precio unitario	Importe (€)
1	h	Instalación del software			2380,47
		Costes directos	1	2288,92	2288,92
	%	Costes indirectos	0,04	2288,92	91,56
2	h	Documentación del simulador			91,05
		Costes directos	1	87,55	87,55
	%	Costes indirectos	0,04	87,55	3,50
3	h	Implementación de la interfaz gráfica para el simulador			218,52
		Costes directos	1	210,12	210,12
	%	Costes indirectos	0,04	210,12	8,40
4	h	Diseño y desarrollo en MATLAB			2731,56
		Costes directos	1	2626,50	2626,50
	%	Costes indirectos	0,04	2626,50	105,06
5	h	Diseño y desarrollo en KEPServer			145,68
		Costes directos	1	140,08	140,08
	%	Costes indirectos	0,04	140,08	5,60
6	h	Diseño y desarrollo en Factory IO			874,10
		Costes directos	1	840,48	840,48
	%	Costes indirectos	0,04	840,48	33,62
7	h	Grabación y edición de vídeos en OBS y Premiere			36,42
		Costes directos	1	35,02	35,02
	%	Costes indirectos	0,04	35,02	1,40

Presupuesto total

En este apartado se obtiene el presupuesto total sumando todos los costes de las unidades de obra, incluyendo los gastos generales, los beneficios y el IVA.

	Importe
Instalación del software	2380,47
Documentación del simulador	91,05
Implementación de la interfaz gráfica para el simulador	218,52
Diseño y desarrollo en MATLAB	2731,56
Diseño y desarrollo en KEPServer	145,68
Diseño y desarrollo en Factory IO	874,10
Grabación y edición de vídeos en OBS y Premiere	36,42
Presupuesto total de Ejecución Material (PEM)	6477,81
Gastos Generales (13%)	842,12
Beneficios (6%)	388,67
Presupuesto total de Ejecución por Contrata (PEC)	7708,60
IVA (21%)	1618,81
Presupuesto Base de Licitación	9327,40

Asciende el presente presupuesto a la expresada cantidad de:

NUEVE MIL TRESCIENTOS VEINTISIETE EUROS CON CUARENTA CÉNTIMOS

Valencia, a 28 de Junio de 2019