

# TRABAJO FIN DE MASTER EN AUTOMÁTICA E INFORMÁTICA INDUSTRIAL

Curso Académico: 2016-17

## DISEÑO Y DESARROLLO DE HERRAMIENTA DIDÁCTICA PARA EL APRENDIZAJE DE LA GESTIÓN DE DIABETES DE TIPO 1 EN NIÑOS.

AUTOR: ANTONIO NADAL GALIANA LLINARES  
TUTOR: JOSÉ LUIS DIEZ RUANO  
COTUTOR: JUAN FRANCISCO BLANES NOGUERA



A mi padre



## **RESUMEN**

La diabetes es una enfermedad en la que los niveles de glucosa (coloquialmente azúcar) en la sangre no se mantienen dentro de unos valores adecuados. Una persona tiene diabetes cuando existe un trastorno en la producción o utilización de la insulina (hormona que produce el páncreas). En la actualidad existen tratamientos efectivos que inciden sobre el nivel de insulina. Éstos requieren de un control riguroso y unos conocimientos básicos sobre la enfermedad y su gestión por parte del paciente.

En el presente proyecto se ha desarrollado un sistema educativo, mediante herramientas basadas en juegos y atractivo para los más jóvenes. Este sistema muestra la evolución de la glucosa en sangre y la acción de la insulina sobre un paciente virtual. Se expone también el funcionamiento de las bombas de insulina y de los futuros páncreas artificiales.

La herramienta didáctica diseñada se compone de una aplicación informática, localizada en un PC, donde se ejecuta el paciente virtual, y un robot tipo oruga programado para responder con diferentes patrones de movimiento y apariencia a los estados glucémicos de dicho paciente virtual.

# Contenido

1.	Introducción .....	7
1.1.	Motivación.....	9
1.2.	Objetivo.....	10
1.3.	Alcance .....	10
1.4.	Interés académico .....	11
2.	Diabetes Mellitus .....	13
2.1.	El Páncreas .....	14
2.1.1.	Glucagón.....	15
2.1.2.	Insulina .....	15
2.1.3.	Somatostatina .....	16
2.1.4.	Polipéptido Pancreático .....	16
2.2.	Diabetes.....	16
2.2.1.	Sintomatología .....	16
2.2.2.	Tipos de diabetes .....	17
2.3.	Consenso de tratamiento de la Diabetes Mellitus .....	18
2.3.1.	Nutrición.....	18
2.3.2.	Ejercicio Físico .....	18
2.3.3.	Antidiabéticos orales.....	19
2.3.4.	Tratamiento con insulinas.....	19
2.3.5.	Trasplantes .....	21
2.4.	Complicaciones de la Diabetes Mellitus.....	21
2.5.	Educación en el Paciente diabético.....	21
2.6.	Páncreas Artificial.....	23
2.6.1.	Bombas de insulina .....	23
2.6.2.	Páncreas artificial .....	24
2.6.3.	Candidatos.....	25
3.	Alternativas de Solución.....	27
3.1.	Hardware.....	27
3.2.	Software .....	29
3.3.	Comunicación .....	30
3.4.	Algoritmos de control.....	32
4.	Modelo de Paciente .....	37
4.1.	Modelo Cambridge.....	38
4.1.1.	Modelo de absorción de glucosa .....	38

4.1.2.	Modelo de absorción subcutánea de insulina .....	38
4.1.3.	Modelo del sistema glucoregulador.....	39
5.	Plataforma Didáctica .....	43
5.1.	Software PC. Matlab.....	44
5.1.1.	Interfaz de Usuario.....	44
5.1.2.	Software Paciente Virtual.....	51
5.2.	Software Robot. Arduino.....	54
5.2.1.	Modos de Operación.....	54
5.2.2.	Gestión de Movimientos.....	56
5.2.3.	Gestión Efectos Visuales .....	60
5.3.	Protocolo de Comunicación .....	62
6.	Conclusiones.....	64
7.	Trabajos futuros .....	67
8.	Presupuesto .....	68
8.1.	Mano de obra .....	68
8.2.	Materiales .....	68
8.3.	Precios unitarios descompuestos por unidad de obra.....	69
8.4.	Presupuesto de ejecución por contrata .....	70
	Bibliografía y Webgrafía.....	71
ANEXO I	Manual de Usuario.....	5
1.	Introducción .....	5
2.	Conocimientos Previos.....	6
3.	Especificaciones Técnicas.....	6
3.1.	Requisitos Hardware .....	6
3.2.	Requisitos Software.....	6
4.	Manual .....	7
4.1.	Primer uso de la Plataforma.....	7
4.2.	Uso de la Plataforma .....	9
Pantalla principal.....		9
Pantalla Paciente con Bomba de Insulina .....		10
Pantalla Siguelinea con Bomba de Insulina.....		10
Pantalla Movimiento Libre .....		12
Pantalla Evita objetos con Bomba de Insulina .....		13
Pantalla Siguelinea Alimentación Manual con Bomba de Insulina .....		14
Pantalla Paciente con Páncreas Artificial .....		15

Pantalla Siguelinea con Páncreas Artificial.....	16
Pantalla Evita objetos con Páncreas Artificial .....	18
Pantalla Siguelinea Alimentación Manual con Páncreas Artificial .....	19
ANEXO II Manual del Programador .....	20
1. Introducción .....	20
2. Estructura General del Código .....	20

# Capítulo 1

## 1. Introducción

---

La diabetes es una grave enfermedad crónica que se desencadena cuando el páncreas no produce suficiente insulina (una hormona que regula el nivel de glucosa en la sangre), o cuando el organismo no puede utilizar con eficacia la insulina que produce. Esta enfermedad es un importante problema de salud pública y una de las cuatro enfermedades no transmisibles (ENT) seleccionadas por diferentes organismos internacionales para intervenir con carácter prioritario.

Según las estimaciones de la OMS, 422 millones de adultos en todo el mundo tenían diabetes en 2014, frente a los 108 millones de 1980. La prevalencia mundial (normalizada por edades) de la diabetes casi se ha duplicado desde ese año, pues ha pasado del 4,7% al 8,5% en la población adulta. Ello supone también un incremento en los factores de riesgo conexos, como el sobrepeso o la obesidad.

En 2012, la diabetes provocó 1,5 millones de muertes. Un nivel de glucosa en la sangre superior al deseable provocó otros 2,2 millones de muertes, al incrementar los riesgos de enfermedades cardiovasculares y de otro tipo.

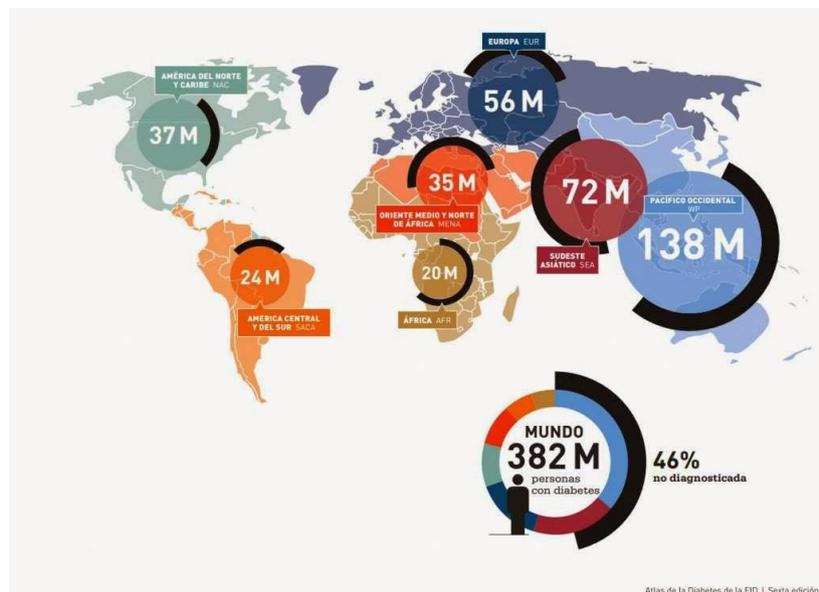


ILUSTRACIÓN 1. NÚMERO DE PERSONAS CON DIABETES POR REGIÓN. FID 2013

No hay soluciones sencillas para combatir la diabetes, pero mediante intervenciones coordinadas con múltiples componentes pueden lograrse cambios importantes.

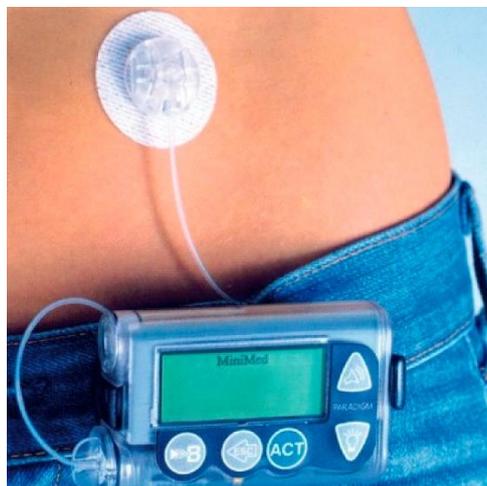
Todos tienen un papel que desempeñar en la reducción de los efectos de la diabetes en sus diferentes variantes. Los gobiernos, los profesionales de la salud, las personas con diabetes, la sociedad civil, los productores de alimentos y fabricantes de productos alimenticios y los proveedores de medicamentos: todos son partes interesadas. En conjunto pueden hacer una contribución importante para detener el aumento de la diabetes y mejorar la calidad de vida de las personas que tienen la enfermedad. (Organización Mundial de la Salud, 2016)

El tratamiento médico fundamental para el control de la diabetes es la insulina. Éste disminuye el nivel de azúcar en la sangre permitiendo que salga del torrente sanguíneo y entre en las células. Comúnmente, la insulina se inyecta de manera subcutánea mediante una jeringa, una pluma de insulina o una bomba.

Desde los comienzos de la terapia con insulina se ha intentado sustituir la función del páncreas con una aproximación lo más ajustada posible a la situación original en estado sano. Las bombas de insulina son, de momento, la mejor aproximación en este intento (ver ilustración 2).

En comparación con las múltiples inyecciones diarias, la terapia con bomba de insulina actúa de forma más parecida a un páncreas sano. El páncreas sano libera continuamente pequeñas cantidades de insulina las 24 horas del día para mantener niveles normales de glucosa en sangre entre las comidas y durante la noche.

Aunque se use una bomba de insulina, sigue siendo necesario monitorizar los niveles de glucemia a lo largo del día actualmente. Así mismo las bombas por sus características actuales no pueden reaccionar a crisis glucémicas de manera autónoma. Estas dejan el control en última instancia en el paciente y por tanto en su conocimiento sobre el nivel glucémico medido.



**ILUSTRACIÓN 2 BOMBA DE INSULINA**

El siguiente paso natural en el desarrollo de un páncreas artificial ha sido dotar a la bomba de insulina de adquisición de datos autónoma, inteligencia y poder de actuación. Se trata de un sistema capaz de detectar cuánta insulina necesita el cuerpo en cada instante y proporcionársela. Según un estudio publicado en «Diabetología» (la revista oficial de la Asociación Europea de Diabetes), un sistema como éste podrá estar disponible en 2018.

En cuanto a los posibles pacientes que se beneficiarían con una terapia con las actuales bombas de insulina o con los futuros páncreas artificiales, es fundamental que estos demuestren

unas habilidades adecuadas en el manejo de la diabetes, que sepan contar hidratos de carbono, que estén motivados, que tengan unas expectativas realistas sobre el tratamiento que van a iniciar y que estén adecuadamente formados en el funcionamiento de los dispositivos de control glucémico (bombas, sensores...). Cualquier niño con diabetes, sea cual sea su edad, es potencial candidato para llevar una bomba de insulina.

La diabetes no se puede controlar tan sólo mediante una bomba de insulina, al igual que no puede controlarse con sólo una jeringa. Los conocimientos, la motivación y el autocontrol son de máxima importancia. Por lo tanto, los niños, los adolescentes y sus padres necesitan una introducción de orientación práctica para usar la bomba día a día.

El paciente es la persona más importante a la hora de controlar la diabetes. Cuanto mayor sea el conocimiento de este sobre el funcionamiento de la enfermedad, mayores serán las posibilidades de conseguir unos niveles de glucemia adecuados y por tanto un mejor nivel de vida.

## 1.1.Motivación

La diabetes, como se ha comentado anteriormente se trata de una enfermedad global con un índice de incidencia realmente considerable en todas las franjas de edades.

Tal como se ha indicado también, las bombas de insulina representan un avance considerable en tratamiento de la enfermedad. Éstas representan mejoras en los siguientes puntos:

- Menos inyecciones de insulina para el paciente.
- Los suministros de insulina son mucho más precisos.
- Proporcionan al paciente de mayor flexibilidad.
- Son un tratamiento útil contra las hipoglucemias.
- En general, mejoran la calidad de vida del paciente.

Otro aspecto de gran importancia en la gestión de la enfermedad es la “adherencia al tratamiento” que tengan los pacientes diabéticos. Un incumplimiento parcial o total del acuerdo terapéutico al que se ha llegado con el profesional ocasiona, como es evidente, graves complicaciones. Cualquier avance en la automatización del proceso de toma de medicamentos repercutirá también en la reducción de los perjuicios de estas conductas.

La siguiente evolución en el tratamiento de la diabetes se dirige hacia la automatización del proceso, liberando al paciente de ciertos cálculos y aumentando la autonomía de las bombas (páncreas artificial)

Tanto en un sistema como en el otro es imprescindible la formación del paciente y/o sus padres en caso de que se trate de un menor. El conocimiento sobre la enfermedad y su gestión constituyen posiblemente el factor más importante en el éxito del tratamiento. El paciente debe tener conocimientos sobre aspectos como la enfermedad, el conteo de hidratos, planificación de ingestas o tratamiento para situaciones especiales (ejercicio físico, enfermedades...). También es muy importante que el paciente conozca el funcionamiento de su bomba de insulina o su futuro páncreas artificial, así como su manejo y programación.

Dentro del ámbito de la enfermedad un colectivo importante es el de los niños. Según un estudio reciente de la OMS más de 500.000 niños tienen diabetes tipo 1 en el mundo. Este

colectivo es candidato al tratamiento mediante bomba de insulina o páncreas artificial incluso dispone de ciertas ventajas con estos tratamientos como se ha comentado anteriormente.

Este escenario hace imprescindible una herramienta didáctica atractiva para los pacientes más jóvenes. Donde se interactúe con conceptos como el conteo de los carbohidratos, los efectos de estos en el nivel de glucosa en sangre, el efecto de la insulina en el nivel glucémico, el funcionamiento de las bombas de insulina o el de los páncreas artificiales. Y todo esto debe mostrarse de una forma atractiva e interactiva con el objetivo de que su aprendizaje resulte motivador para los pacientes más jóvenes.

## 1.2.Objetivo

El objetivo del actual proyecto es ofrecer una plataforma de aprendizaje sobre la diabetes y su gestión, dirigida especialmente a niños (aunque sea perfectamente aprovechable por pacientes de cualquier edad). La plataforma dispondrá de diferentes juegos que simulen los efectos de la gestión de la enfermedad en un robot tipo oruga.

El sistema desarrollado deberá también cumplir los siguientes sub-objetivos:

1. Ser sencilla de manejar y su utilización debe ser intuitiva.
2. Representar en su estructura una primera abstracción hacia el páncreas artificial.
3. Tratarse de un instrumento interactivo en el que se puedan probar distintos escenarios.
4. La información debe ser mostrada de una manera atractiva y sencilla.
5. La plataforma debe exponer, para el aprendizaje por parte del paciente, tanto aspectos sobre la propia enfermedad como sobre su gestión.
6. Debe introducir también al paciente en el funcionamiento y la operación de las bombas de insulina y los páncreas artificiales.

## 1.3.Alcance

En el siguiente proyecto se presenta una plataforma interactiva especialmente diseñada para facilitar el aprendizaje tanto de aspectos generales sobre la diabetes como sobre el funcionamiento y gestión de los nuevos páncreas artificiales.

Para conseguir este propósito se han programado una serie de juegos que introducirán al paciente en los objetivos deseados.

El sistema programado se compone de un software en Matlab que simula un paciente virtual y presenta una interfaz atractiva e intuitiva donde se puede interactuar con diferentes juegos. Y en paralelo a esto otro software en un robot tipo oruga, basado en una placa Arduino, donde se mostrarán los estados del paciente mediante diferentes patrones de comportamiento. Estos estados estarán influenciados por las ingestas de alimentos e insulina del paciente.

El paciente virtual correrá en un PC haciendo uso del modelo fisiológico del sistema insulina-glucosa propuesto por el grupo de Cambridge (Laguna Sanz, 2014), incluyendo las ecuaciones que definen el modelo.

El comportamiento del sistema será el siguiente:

Un paciente virtual será ejecutado en el PC (Matlab) con diferentes juegos con diversos patrones de ingesta de alimentos e insulina y se envía en tiempo real el nivel glucémico de dicho paciente al robot tipo oruga. Éste realiza dos tareas, la primera, realiza los cálculos necesarios para enviar al PC una señal de control adecuada (PID) y la segunda activa un patrón de comportamiento concreto en el robot según el índice glucémico recibido.

Mientras la simulación se está realizando se muestran valores de glucemia del paciente virtual en el PC. En la *ilustración 3* se muestra un esquema general de la plataforma.

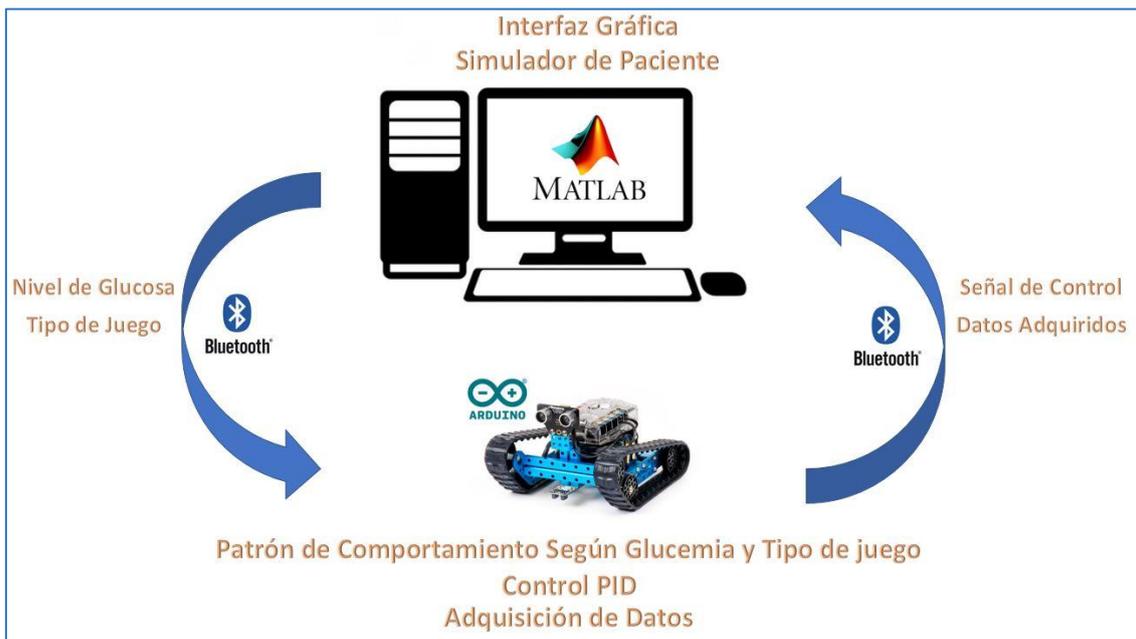


ILUSTRACIÓN 3 ESTRUCTURA DE LA PLATAFORMA EDUCATIVA

## 1.4 Interés académico

El proyecto se desarrolla bajo el contexto de un Trabajo Final de Máster, requisito necesario para la obtención del Master en Automática e Informática Industrial englobado en el programa de doctorado en Automática, Robótica e Informática Industrial de la Universidad Politécnica de Valencia.

Desde el enfoque académico se distinguen varios motivos que justifican su realización:

- **Desarrollo de software.** Dentro de este ámbito se aplican técnicas de estructura de software, así como programación enfocada a procesos en tiempo real. Ambos conceptos tratados durante el master.
- **Implementación de protocolos de comunicaciones.** Durante el proyecto resultó imprescindible el uso de protocolos de comunicaciones para obtener un sistema robusto en el que se pudieran comunicar dos procesadores. Por lo que se ha tenido que profundizar en este campo.
- **Instrumentación.** El robot tipo oruga dispone de diversos sensores montados sobre él. En las aplicaciones desarrolladas se hace uso de dichos sensores.
- **Robótica.** Se han definido también patrones de comportamiento para el robot tipo oruga.

- **Sistemas de control.** En el proyecto se ha puesto de manifiesto la importancia de la ingeniería de control en el campo de los nuevos tratamientos para la diabetes. En este campo se ha realizado una aplicación práctica y orientada a la programación de los conocimientos adquiridos en el Máster.

# Capítulo 2

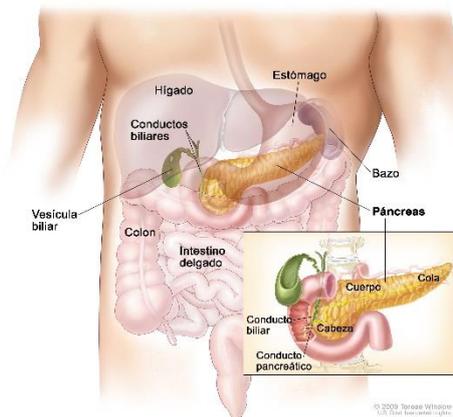
## 2. Diabetes Mellitus

---

El ser humano necesita, indispensablemente, para su supervivencia el aporte de energía. Este aporte es suministrado por los alimentos que ingiere. De ellos podemos extraer las grasas, las proteínas, los hidratos de carbono, etc.

Cada gramo de grasa aportará al organismo 10 calorías, cada gramo de proteínas 4 y cada gramo de hidratos de carbono otros cuatro.

Los alimentos ingeridos son sometidos a la digestión de los mismos. Para que el organismo pueda utilizar la energía en ellos contenida es necesario, previamente, la intervención de diversos elementos que desempeñarán varias funciones en los procesos metabólicos.



**ILUSTRACIÓN 4 APARATO DIGESTIVO CON LA LOCALIZACIÓN DEL PÁNCREAS**

En el proceso digestivo es esencial la función del páncreas ya que aportará las enzimas pancreáticas.

En el proceso de utilización de la energía -en forma de glucosa-, el páncreas con su función endocrina también desempeñará una labor indispensable.

## 2.1. El Páncreas

El páncreas es una glándula<sup>1</sup> alargada (12-15cm) que pesa hasta 100g, extendiéndose longitudinalmente detrás del estómago.

Se trata de una glándula dual:

- **Una glándula exocrina**<sup>2</sup>. Esta secreta las enzimas digestivas que se encuentran en el jugo pancreático: lipasas, amilasas, proteasas. Gracias a su actividad sobre las grasas, los hidratos de carbono y las proteínas, los enzimas pancreáticos facilitan la digestión y favorecen la absorción de alimentos.

El déficit de enzimas pancreáticas producidas por algunas patologías ocasiona que los pacientes deban ingerirlas en forma de medicamentos.



ILUSTRACIÓN 5 TRATAMIENTO TÍPICO DE INSUFICIENCIA PANCREÁTICA EXOCRINA

- **Una glándula endocrina**<sup>3</sup>. La porción endocrina está compuesta por pequeños islotes dispersos de células llamados islotes pancreáticos (solo constituyen el 2, 3% de la masa total del páncreas).

*Nos centraremos en la parte endocrina de la glándula ya que es la que influye más directamente en la diabetes de un paciente.*

Cada uno de los uno a dos millones de islotes pancreáticos del páncreas contiene una combinación de cuatro tipos primarios de células endocrinas, todas ellas unidas entre sí por uniones en hendidura. Cada tipo de célula segrega una hormona<sup>4</sup> diferente, pero las uniones permiten alguna coordinación de estas funciones en una sola unidad secretora.

La hormona glucagón, la insulina, la somatostatina y los polipéptidos pancreáticos son cuatro hormonas pancreáticas que trabajan en equipo para mantener la homeostasis<sup>5</sup> de las moléculas alimentarias (glucosa, ácidos grasos y aminoácidos).

<sup>1</sup> Célula o conjunto de células que almacenan o segregan algún tipo de sustancia.

<sup>2</sup> Una glándula exocrina es un órgano que sintetiza y libera diversas sustancias (hormonas, líquidos diversos) por fuera de la circulación sanguínea.

<sup>3</sup> Las glándulas endocrinas son un conjunto de glándulas que producen sustancias mensajeras llamadas hormonas, vertiéndolas sin conducto excretor, directamente a los capilares sanguíneos, para que realicen su función en órganos distantes del cuerpo.

<sup>4</sup> Sustancia segregada en la corriente sanguínea por una glándula endocrina que actúa sobre un tejido diana específico para producir una determinada respuesta.

<sup>5</sup> Conjunto de fenómenos de autorregulación, conducentes al mantenimiento de una relativa constancia en la composición y las propiedades del medio interno de un organismo.

Un tipo de islote pancreático es la célula alfa, que segrega la hormona glucagón. Las células beta segregan la hormona insulina, las células delta segregan la hormona somatostatina y todas las células polipéptidas pancreáticas segregan polipéptido pancreático.

Las células beta, que forman unas tres cuartas partes de todas las células de los islotes pancreáticos, suelen encontrarse cerca del centro de cada islote, en cuanto que las células de los otros tres tipos se encuentran con más frecuencia en la porción exterior.

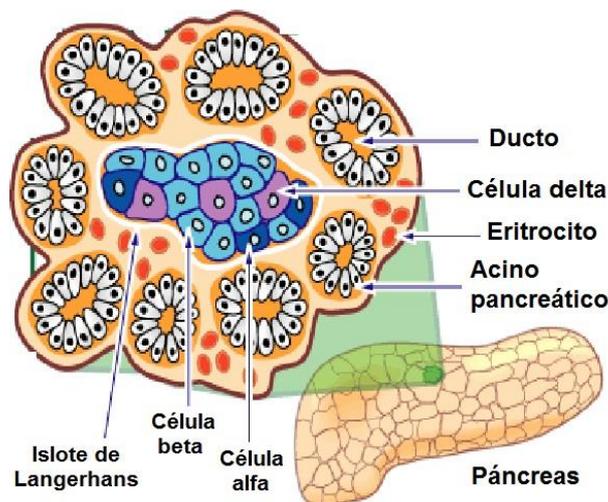


ILUSTRACIÓN 6 ANATOMÍA DEL PÁNCREAS

Las características más importantes de cada una de ellas las enunciamos a continuación.

### 2.1.1. Glucagón

El glucagón, producido por las células alfa, tiende a incrementar los niveles de glucemia, estimulando la conversión de glucógeno<sup>6</sup> en glucosa en las células hepáticas. También estimula la gluconeogénesis (transformación de ácidos grasos y aminoácidos en glucosa) en las células hepáticas. La glucosa producida en la degradación del glucógeno y por la gluconeogénesis se vierte en la corriente sanguínea, produciendo un efecto hiperglucémico.

### 2.1.2. Insulina

La insulina (este concepto se desarrollará ampliamente con posterioridad), producida por las células beta, tiende a favorecer el movimiento de glucosa, aminoácidos y ácidos grasos de la sangre a los tejidos. Por consiguiente, la insulina tiende a aminorar las concentraciones hemáticas de estas moléculas alimentarias y a favorecer su metabolismo por las células tisulares<sup>7</sup>.

<sup>6</sup> Es un polisacárido de reserva energética que se encuentra en abundancia en el hígado y en los músculos y puede transformarse en glucosa cuando el organismo lo requiere.

<sup>7</sup> Células que forman los tejidos.

### 2.1.3. Somatostatina

La somatostatina, producida por las células delta afecta a diferentes tejidos del cuerpo, pero su principal papel es regular a las otras células endocrinas de los islotes pancreáticos: la somatostatina inhibe la secreción de glucagón, insulina y polipéptido pancreático.

### 2.1.4. Polipéptido Pancreático

El polipéptido<sup>8</sup> pancreático lo producen las células PP (o F) de la periferia de los islotes pancreáticos. Aunque queda mucho por conocer acerca del polipéptido pancreático, se sabe que influye en algún grado en la digestión y en la distribución de las moléculas alimentarias.

## 2.2. Diabetes

La patología del páncreas endocrino y su función productora de insulina estándar es uno de los trastornos endocrinos más habituales. Afecta a buena parte de la población mundial como se ha comentado en la introducción de esta memoria. Esta patología conocida con el nombre de diabetes se describe como un síndrome, es decir, un conjunto de síntomas que caracterizan la enfermedad. Aunque los síntomas de la misma tienden a variar entre los individuos, así como el tipo, gravedad y duración de la enfermedad, cada signo o síntoma está relacionado de alguna manera con el metabolismo anormal de los nutrientes y con las consecuencias que acarrea.

Los diabéticos producen una menor cantidad de insulina. En otros individuos afectados, el menor número de receptores de la insulina “normales”, o de los sistemas de segundo mensajero en las células diana, hacen imposible que la glucosa entre en las células, aunque exista insulina normal y en cantidades adecuadas. Una hormona secretada por las células adiposas y llamada resistina puede interferir también con la acción de la insulina.

La presencia de cantidades adecuadas de insulina normal es la llave de entrada de esta última en las células. Si no hay problemas de insulina y los principales tejidos diana, como músculo esquelético, grasa e hígado, tienen receptores de insulina apropiados, la glucosa pasará de la sangre a las células. En la patología conocida como diabetes la glucosa no puede penetrar en las células. El resultado es uno de los síntomas más universales de la enfermedad, la elevación crónica de los niveles de glucosa en sangre, una situación denominada hiperglucemia. (Thibodeau & Patton, Kevin T., 2009)

### 2.2.1. Sintomatología

**Poliuria:** En sentido estricto, designa la producción de una gran cantidad de orina, lo que provoca la necesidad de orinar con mucha frecuencia. En el funcionamiento normal, se filtra glucosa desde la sangre y luego se reabsorbe por los túbulos renales; no obstante, en la diabetes aumentan los niveles de glucemia, de modo que la cantidad filtrada desde la sangre supera la capacidad de los túbulos renales para reabsorberla. El resultado es un rebosamiento de glucosa en la orina (denominado glucosuria) aumentando la producción de orina (poliuria), ya que se

---

<sup>8</sup> Cadena de aminoácidos que se unen entre sí mediante enlaces peptídicos.

necesita más agua para llevar la carga de azúcar. El exceso de glucosa actúa como un diurético osmótico.

**Polidipsia:** Es la sensación constante de sed intensa. Al perderse por la orina grandes cantidades de agua, el cuerpo se deshidrata. Deshidratación que empeora cuando los elevados niveles de glucemia aumentan la concentración osmótica de la sangre y sacan agua de las células.

**Polifagia:** Es la constante sensación de hambre. Dado que las células carecen de glucosa para quemarla como fuente de energía, los diabéticos suelen tener hambre intensa y continua. Su glucemia es alta, pero las células se mueren literalmente de hambre y el cuerpo reclama comida. En la exploración de la diabetes, los médicos buscan las tres polis: poliuria, polidipsia y polifagia.

**Cansancio:** El diabético no tratado es incapaz de utilizar glucosa como fuente de energía, por lo que el cuerpo se ve obligado a quemar proteína y grasa. Este cambio metabólico anormal origina fatiga.

**Adelgazamiento:** Al quemar grasas y proteínas para obtener energía se produce el adelgazamiento corporal. (Standards of Medical Care in Diabetes, 2016)

### 2.2.2. Tipos de diabetes

Existen distintos tipos de diabetes:

**Diabetes secundaria.** Hace mención a la diabetes causada por la cirugía con la extirpación total o parcial del páncreas o bien por la toma de ciertos medicamentos en dosis elevadas (corticoides o inmunosupresores), o por ciertas enfermedades que tengan como efecto un déficit en la producción de insulina [dígase la diabetes relacionada con la fibrosis quística].

**Diabetes gestacional.** La Diabetes Gestacional se caracteriza por la intolerancia a los hidratos de carbono y por el aumento de glucosa en sangre durante el embarazo.

**Diabetes Mellitus Tipo I.** La diabetes mellitus tipo 1 se denominaba antes diabetes de comienzo juvenil, ya que suele aparecer antes de los 30 años. En esta forma de la enfermedad se destruyen las células beta de los islotes pancreáticos y existe una falta absoluta de producción de insulina. Los individuos con diabetes tipo I necesitan inyecciones diarias de insulina para prevenir la cetosis<sup>9</sup> y controlar la hiperglucemia. Como consecuencia, la diabetes tipo I también se denomina diabetes mellitus insulino dependiente (DMID). En la DMID los síntomas expuestos suelen aparecer con relativa rapidez. (Jameson JL, 2015)

La causa de la destrucción de células beta en la diabetes tipo I todavía es dudosa, pero la investigación actual sugiere que la DMID es una enfermedad autoinmune que se debe probablemente a algún tipo de infección vírica en individuos genéticamente susceptibles.

Cualquiera que tenga un padre, madre, hermano o hermana con DMID tiene un 5-7% de probabilidades de contraer la enfermedad. Si un gemelo univitelino (del mismo óvulo) tiene DMID, el riesgo se eleva al 50%.

**Diabetes Mellitus Tipo II.** La diabetes mellitus tipo II o diabetes mellitus no insulino dependiente (DMNID) es la forma más frecuente de la enfermedad. Dado que su frecuencia es mayor después

---

<sup>9</sup> Cantidad elevada de cuerpos cetónicos (ácido producido durante el catabolismo de las grasas), presentes en la sangre de una persona con diabetes mellitus incontrolada.

de los 40 años, también se denomina diabetes de comienzo en la madurez. En individuos susceptibles con sobrepeso, la incidencia aumenta con la edad. En esta forma de diabetes las células beta todavía producen insulina, pero por lo general en cantidades reducidas. Además, la pérdida de receptores de insulina en las membranas de superficie de las células diana también reduce la efectividad de la absorción de glucosa de la sangre. Esta afección puede pasar desapercibida durante años, ya que el páncreas no ha cesado totalmente de producir insulina.

La denominación DMNID no es apropiada porque pueden necesitarse inyecciones de insulina, combinadas con antidiabéticos orales a menudo, para controlar la enfermedad. Sin embargo, en muchos diabéticos de este tipo la hiperglucemia suele responder a cambios en el estilo de vida, que consisten en comer una dieta equilibrada, ejercicio adecuado y mantener el peso corporal en los límites normales. La herencia y los antecedentes étnicos son importantes determinantes en la DMNID. Los que tengan antecedentes familiares de la enfermedad son especialmente susceptibles si están en sobrepeso y son sedentarios.

## 2.3. Consenso de tratamiento de la Diabetes Mellitus

Los distintos tipos de diabetes, mencionados con anterioridad, han requerido diferentes tipos de tratamiento, con un denominador común: reducir los niveles de glucosa en sangre y los factores de riesgo cardiovascular. Tradicionalmente se ha utilizado la dieta y el ejercicio físico como primeras herramientas para enfrentarse con los primeros estadios de la patología.

### 2.3.1. Nutrición

El control adecuado de la **nutrición** en la enfermedad tiene una gran relevancia, siendo una estrategia fundamental en la prevención de la resistencia a la insulina, en el entecimiento de su desarrollo, así como en la prevención primaria y secundaria de la aparición de sus complicaciones vasculares y neurológicas. Las recomendaciones nutricionales para los pacientes diabéticos se dirigen principalmente a la regulación y control de los macronutrientes, proteínas, grasas e hidratos de carbono, así como a un control de la ingesta en términos calóricos.

### 2.3.2. Ejercicio Físico

La práctica de **ejercicio físico** regular también es importante para el control metabólico y de otros factores de riesgo cardiovascular. El ejercicio físico es recomendable para los diabéticos adaptado a la edad y a su situación cardiovascular, puesto que disminuye las cifras de glucemia, aumenta la sensibilidad a la insulina, contribuye a mejorar los perfiles lipídicos y tensiones y ayuda también a controlar el peso corporal.

Cuando estos métodos resultan insuficientes, se recurre adicionalmente a las aportaciones farmacológicas como los antidiabéticos orales o las insulinas.



ILUSTRACIÓN 7 PILARES DEL TRATAMIENTO DE LA DIABETES MELLITUS

### 2.3.3. Antidiabéticos orales

Un **antidiabético oral** es un fármaco que se administra por vía oral, que actúa disminuyendo los niveles de glucemia por lo que también se llama hipoglucemiante oral. Concretamente existen tres grandes grupos: sensibilizadores a la insulina propia, estimulantes de la secreción de insulina y reductores de la absorción de la glucosa.

### 2.3.4. Tratamiento con insulinas

El objetivo del tratamiento con **insulina** es imitar la función de un páncreas de las personas sin diabetes. La insulina que se utiliza actualmente es una insulina con la misma estructura química que la humana que se fabrica por técnicas de ingeniería genética. La misma hasta la actualidad no se puede tomar en forma de pastillas.

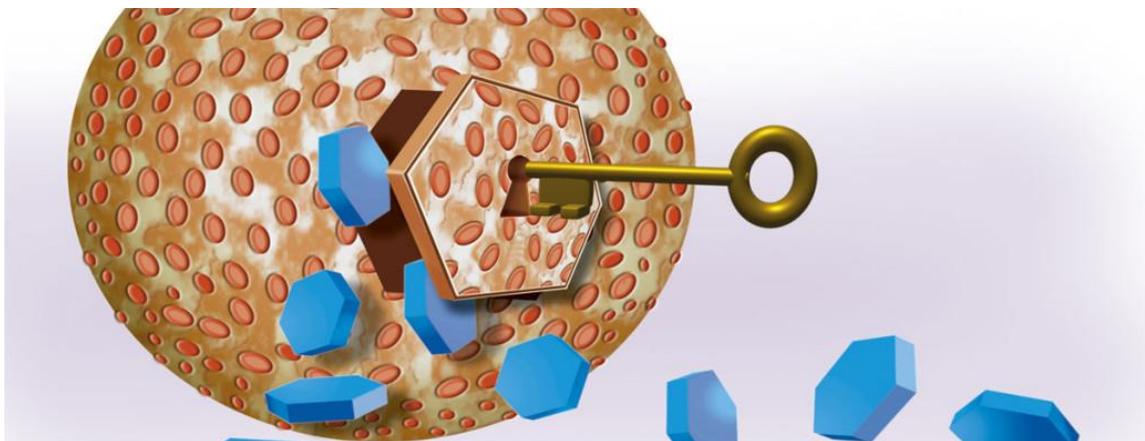


ILUSTRACIÓN 8 ABSTRACCIÓN DE LA FUNCIÓN DESARROLLADA POR LA INSULINA

El descubrimiento de la insulina en 1921 fue uno de los avances más importantes de la medicina en este siglo. Los problemas asociados a la inyección de insulina obtenida de tejido pancreático animal se resolvieron al disponer de insulina humana sintética.

En la actualidad existen diferentes tipos de insulina que pasamos a enumerar a continuación:

- **Insulinas de acción rápida.** Son un tipo de insulinas con alguna modificación molecular que permite cambiar su comienzo de acción o su duración. Comienzan a ser activas a los 10-15 minutos de haberse pinchado. Su pico de actividad es a los 30-90 minutos y duran entre 3 y 4 horas. Su perfil de acción para cubrir las comidas es actualmente el más parecido a la insulina liberada por el páncreas de una persona sin diabetes.
- **Insulina regular.** También llamada insulina soluble o cristalina. Comienza a ser activa a los 30-60 minutos de haberse pinchado. Su pico de actividad es a las 2-3 horas y su duración es de 5-7 horas. Se utiliza para cubrir las comidas, aunque su perfil de acción no se ajusta demasiado a la forma de actuar de la insulina liberada por el del páncreas de una persona sin diabetes y su utilización ha caído en desuso tras la aparición de los análogos de acción rápida.

- **Insulina de acción intermedia.** Se emplea como insulina basal. Se denomina Insulina NPH (Neutral Protamine Hagedorn). A la insulina humana se le ha añadido una proteína (protamina) para que su absorción sea más lenta, de esta manera su duración es de 10 a 13 horas. Tiene un pico de acción entre 4 y 7 horas después de haberse pinchado y comienza a ser activa a la 1-2 horas. Para utilizarla como insulina basal hay que dar, en general, tres dosis al día.
- **Insulinas de acción lenta o prolongada.** Se caracterizan por tener una menor variabilidad respecto a la NPH y una duración de la acción más alta, del orden de 24h.
- **Insulinas combinadas.** Son combinaciones de insulinas de acción rápida con insulinas de acción intermedia en diferentes proporciones. En un mismo dispositivo, y con 1 sólo pinchazo, se administran los dos tipos de insulinas, por lo que se controlan tanto las glucemias basales como las prandiales<sup>10</sup>.

En el mercado existen diferentes mezclas preestablecidas de insulinas de análogos de acción rápida o regular e intermedia. Tienen el inconveniente de que el porcentaje de cada insulina es fijo y no se pueden modificar las insulinas individualmente (por variaciones existentes en el control de glucemia, ejercicio físico, raciones de alimentos, enfermedad...).

Lo más apropiado es que cada persona realice sus propias mezclas de acuerdo a sus necesidades diarias y siguiendo las instrucciones de su equipo médico.

**Terapia basal-bolo.** El principal objetivo en el tratamiento de la diabetes es imitar lo máximo posible el funcionamiento del páncreas de una persona sin diabetes, como ya se ha comentado. Recordemos que el páncreas va a producir una pequeña cantidad de insulina continuamente para regular la salida de glucosa del hígado. A esto se le llama insulina basal. Además, cada vez que comemos, el páncreas secretará una cantidad de insulina adicional, que se conoce como bolo.

La terapia basal-bolo consiste en imitar esta forma de actuar del páncreas. Para ello disponemos hoy en día de dos alternativas. Una de ellas es el infusor continuo de insulina, y la otra forma consistiría en combinar las diferentes insulinas de acción intermedia o lenta para cubrir la secreción basal e insulinas de acción rápida para cubrir los bolos.

El principal inconveniente de este tipo de tratamiento es que se precisan, tanto para adultos como para niños, múltiples pinchazos durante el día (un mínimo de 4). Las ventajas del tratamiento basal-bolo son muchas, desde mejoría del control metabólico por ser un tratamiento más fisiológico, hasta una mayor flexibilidad horaria que suele conllevar a una mejor calidad de vida.

---

<sup>10</sup> La glucosa postprandial es aquella que se hace presente en la sangre dos horas después de haber ingerido los alimentos.

**Métodos de administración:** Existen diversos dispositivos de administración: viales, plumas con cartucho recambiable, plumas desechables, bombas para infusión continua...



**ILUSTRACIÓN 9 PLUMA DE INSULINA**

La insulina puede introducirse en el cuerpo por inyección regular con aguja y jeringa o bien implantando bombas miniaturizadas que liberan la insulina cuando se necesita.

### 2.3.5. Trasplantes

Los trasplantes de páncreas eliminan la necesidad de inyección o administración de insulina y cada vez se realizan con más éxito gracias a la introducción de nuevas técnicas quirúrgicas y fármacos más eficaces frente al rechazo. Sin embargo, los trasplantes de células de islotes, en lugar de todo el páncreas, son una de las formas de tratamiento más prometedoras para los casos más graves de diabetes de tipo I.

## 2.4. Complicaciones de la Diabetes Mellitus

Por desgracia, si no se trata, la hiperglucemia de la diabetes causa con el tiempo numerosas complicaciones que afectan a casi todas las áreas corporales. La reducción de la circulación sanguínea causada por la acumulación de materiales grasos en los vasos sanguíneos (aterosclerosis) es una de las complicaciones más graves. La enfermedad origina problemas tan diversos como ataque al corazón, apoplejía y circulación reducida en las extremidades, que produce hormigueo o entumecimiento de los pies y en los casos graves gangrena. Las alteraciones retinianas (retinopatía diabética) puede causar ceguera a algunos diabéticos que han luchado contra la enfermedad durante décadas. Las lesiones nerviosas también se producen en esta enfermedad. La escrupulosa regulación de los niveles glucémicos y la reducción de los factores de riesgo cardiovascular, como la hipertensión o el colesterol malo (LDL) son las medidas más importantes que pueden tomar los diabéticos para reducir las complicaciones a largo plazo derivadas de la enfermedad.

## 2.5. Educación en el Paciente diabético

El control óptimo de la diabetes y de las graves consecuencias que tendrá un mal control de la misma, depende de tantos factores de la vida diaria del paciente que es imposible lograrlo sin la colaboración activa del mismo. Los niveles de glucosa en sangre son fruto de un proceso

multifactorial: la ingesta de hidratos de carbono; la correlación entre los niveles de glucosa (su control), y los de insulina (incluso la inyección de la misma); el ejercicio físico, el estrés, etc.

Por todo lo dicho es imprescindible que el paciente diabético disponga de los suficientes conocimientos teórico-prácticos, protocolos de actuación, de las actitudes, ..., para enfrentarse en su vida diaria tanto con el discurrir habitual de su enfermedad como en los diferentes episodios de crisis agudas.

En la actualidad han cambiado radicalmente los principios básicos “clásicos” de la educación del paciente diabético. Tanto en las características exigibles al educador, como a los contenidos y los métodos de la educación. Educar no es informar (informar es transmitir conocimientos). En la educación se trabaja con el paciente, y no para el paciente, puesto que el objetivo, como se ha dicho, es conseguir que el paciente adquiera el control y la responsabilidad de manejo diario de su diabetes. (Sánchez Juan, 2010)

En el proceso de enseñanza aprendizaje se partirá siempre de la situación del paciente (tanto de su edad, su situación emocional, sus conocimientos...).

Entre los contenidos de la educación del paciente cabría destacar:

- **Generalidades:**  
Etiología, diagnóstico, clasificación, clínica, hipoglucemias, hiperglucemias, etc.
- **Tratamientos:**
  - Dieta, ejercicio, hipoglucemiantes orales.
  - Dispositivos para el suministro de insulinas: agujas, “bolígrafos”, bombas de infusión continua, etc.
  - El tratamiento en niveles bajos de azúcar en la sangre (hipoglucemia).
  - El tratamiento en niveles altos de azúcar en la sangre (hiperglucemia)
  - La planificación de las comidas, incluso el conteo de carbohidratos.
  - La administración de la insulina.
  - El chequeo de la glucosa en la sangre (glucómetro).



ILUSTRACIÓN 10 GLUCÓMETRO.

- El ajuste de insulina y alimentos durante el ejercicio.
- El manejo de la diabetes en las situaciones especiales.
- **Complicaciones:**  
Agudas, crónicas, pie diabético, riesgo cardiovascular.
- **Aspectos psicosociales.**
  - Desdramatización.
  - Personalidad del paciente.
  - Actividad social y situaciones extraordinarias.

## 2.6. Páncreas Artificial

Mejorar las terapias de reemplazo de la insulina hasta que el control glucémico sea prácticamente normal (evitando las complicaciones de la diabetes mellitus) y disminuir la carga de la terapia para los pacientes insulino-dependientes, están en la base de la búsqueda de nuevos sistemas para el suministro de la insulina: bomba de insulina y páncreas artificial.

Para la implementación de estos sistemas es básica la evolución en tres áreas del conocimiento: sobre la propia insulina, la forma de suministrar la insulina y la tecnología para medir la glucosa.

Se estima como diferencia esencial el tipo de control ejercido en cada sistema, siendo el de la bomba de insulina un control en lazo abierto (la medición del nivel de glucosa la realiza el paciente y éste programa la bomba según la información obtenida) y el del páncreas artificial se trata de un control en bucle cerrado (el sistema decide y actúa según los datos obtenidos por el mismo).

### 2.6.1. Bombas de insulina

Las bombas de insulina son dispositivos electromecánicos computarizados que albergan un cartucho de insulina, similar a una jeringuilla. La insulina del cartucho se dispensa a través de un sistema de infusión, el cual se inserta en el tejido subcutáneo, generalmente en el abdomen. La insulina se infunde de manera casi continua.

Estas bombas son capaces de suministrar tanto la insulina basal como los bolos. La insulina basal se suministra de manera casi continua por lo que no se llega a formar depósito de insulina en el tejido subcutáneo aumentando de forma apreciable la efectividad de la misma. La infusión basal puede variar a lo largo del día, se puede suministrar a demanda y de forma inmediata o bien diferida durante un cierto periodo de tiempo. Actualmente existen modelos capaces de infundir cantidades tan pequeñas que permiten alcanzar un grado de control muy elevado.

Para la terapia intensiva con bomba se utiliza únicamente insulina de acción rápida. Los principales fallos que se pueden producir en la infusión de la insulina son: las obstrucciones de la infusión, las fugas, la pérdida de potencia a la insulina o el mal funcionamiento de la bomba, con las consiguientes hipoglucemias o hiperglucemias (Bondia, 2010)

El funcionamiento de la bomba de insulina no es tan sencillo como parece ya que se requiere de una programación de la bomba e introducción de datos de sensibilidad, ratio insulina/hidratos de carbono, objetivos glucémicos o duración de la insulina. Con todos estos parámetros metidos dentro de la bomba ésta nos recomendará una cantidad de insulina determinada. Pero será el portador de la bomba el que decida en última instancia si se administra la cantidad recomendada o no.

Las bombas de insulina presentan ciertas carencias. Sigue siendo necesario monitorizar los niveles de glucemia a lo largo del día. Así mismo las bombas por sus características actuales no pueden reaccionar a crisis glucémicas de manera autónoma. Estas dejan el control en última instancia en el paciente y por tanto en su conocimiento sobre el nivel de glucemia medido.

Las bombas de insulina suelen impactar de manera positiva en la mejora de la calidad de vida de las personas con diabetes.

### 2.6.2. Páncreas artificial

El siguiente paso natural en el desarrollo del páncreas artificial ha sido proveer al sistema de la capacidad de adquirir datos de forma autónoma, dotarle de inteligencia y también de poder de actuación. Este paso acerca el funcionamiento de las bombas de insulina a la actuación de un páncreas de una persona sana.

Fue a principios de la década de los ochenta cuando los pacientes empezaron a utilizar los “medidores” de la glucosa en sangre. Los avances en la última década de los monitores continuos de glucosa han sido clave en el impulso dado a la idea de “cerrar el lazo” en el control de nivel de glucosa en sangre en el DM1. En un principio los monitores continuos de glucosa únicamente fueron aceptados como adjunto a las mediciones capilares mediante tiras reactivas, con posterioridad su mejora ha encaminado su eventual utilización como único elemento de medición en sistemas de lazo cerrado.

En esta evolución el páncreas artificial monitoriza continuamente los niveles de glucosa en sangre y libera insulina al cuerpo cuando es necesario, en un único proceso. El nuevo dispositivo es capaz de reemplazar el funcionamiento del órgano humano, encargado de modular la liberación de insulina, al interconectar y coordinar el funcionamiento del monitor continuo y la bomba de insulina. Esto eliminaría la necesidad en el paciente de medir continuamente el nivel de glucosa en sangre.

Con este nuevo sistema se ha cerrado el lazo de control alrededor de la glucemia. El mismo sistema que suministra la insulina dispone de información y capacidad de actuación para influir en la glucemia del paciente.

En función de la instrumentación empleada podemos hablar de varias opciones para cerrar el lazo:

- Monitorización de glucosa: subcutánea o intravenosa.
- Infusión de la insulina: subcutánea, intraperitoneal o intravenosa.

Cada una de estas configuraciones presenta características esenciales tal como se muestra en la Ilustración 11.

Medición de glucosa	Infusión de insulina	Comentario
i.v.	i.v.	Sólo factible en entornos como la unidad de cuidados intensivos debido al acceso vascular que requiere.
i.v.	i.p.	Sensor y bomba implantable. Problemas de pérdida de sensibilidad del sensor. Infusión más fisiológica. Requiere cirugía.
s.c.	i.p.	Monitor mínimamente invasivo y bomba implantable. Infusión más fisiológica. Requiere cirugía.
s.c.	s.c.	Monitor mínimamente invasivo y bomba externa. Acceso cómodo al tejido subcutáneo. Infusión no fisiológica. La absorción subcutánea introduce un retraso adicional en la acción de la insulina.

**ILUSTRACIÓN 11 RUTAS PARA EL PÁNCREAS ARTIFICIAL (BONDIA, 2010)<sup>11</sup>**

<sup>11</sup> i.v.: infusión de insulina intravenosa.  
i.p.: infusión de insulina intraperitoneal.

Entre los problemas encontrados en el desarrollo del páncreas artificial se pueden ver:

- El efecto de una ingesta en la glucemia depende de varios factores: composición nutricional, forma de cocinado, etc.
- La variabilidad de respuesta existente entre los diabéticos e incluso a lo largo del día en un mismo paciente.
- La influencia de otros factores: el estrés, el ejercicio, etc.
- La precisión del sensor para proporcionarle datos exactos al sistema.

Según un estudio publicado en «Diabetología» (la revista oficial de la Asociación Europea de Diabetes), un sistema cerrado como éste podrá estar disponible en 2018.



**ILUSTRACIÓN 12 EJEMPLO DE SISTEMA DE PÁNCREAS ARTIFICIAL**

Se trata de un sistema capaz de detectar cuánta insulina necesita el cuerpo a cada instante y proporcionársela. La idea es que, mientras la diabetes no tenga cura, aquellas personas que la padecen puedan tener una vida más fácil.

Idealmente, para replicar la función natural del páncreas lo mejor posible, un páncreas artificial debería algún día poder reemplazar todas las funciones endocrinas perdidas, incluyendo la administración de insulina, la amilina, el glucagón y el péptido C.

### 2.6.3. Candidatos

En cuanto a la candidatura a la terapia con las actuales bombas de insulina o con los futuros páncreas artificiales, es fundamental, que los candidatos demuestren unas habilidades adecuadas en el manejo de la diabetes, sepan contar hidratos de carbono, estén motivados, tengan unas expectativas realistas sobre el tratamiento que van a iniciar y estén adecuadamente formados en el funcionamiento de los dispositivos de control glucémico (bombas, sensores...). Sobre todo, los niños, los adolescentes y sus padres necesitan una introducción de orientación práctica para usar la bomba día a día, impartida por un equipo pediátrico multidisciplinar con experiencia en terapia mediante bomba de insulina. Además de los detalles técnicos y la

---

s.c.: infusión de insulina subcutánea.

dosificación adecuada de insulina, es necesario aprender cómo aplicar el dispositivo de manera eficaz y responsable.

Los niños en edad escolar suelen poder dominar las características técnicas de una bomba de insulina. Sin embargo, los niños siguen necesitando de supervisión para la compleja tarea de calcular sus requisitos de insulina, para las dosis a la hora de las comidas o, para corregir una subida de glucosa en sangre.

La diabetes no se puede controlar tan sólo mediante una bomba de insulina, al igual que no puede controlarse con sólo una jeringa. Los conocimientos, la motivación y el autocontrol son de máxima importancia.

Desde un punto de vista fisiológico, aprender un comportamiento alimentario normal sin conflictos frecuentes con los padres es especialmente importante para los niños pequeños. Una pequeña pulsación para liberar insulina cuando el niño quiere comer algo es, obviamente, menos oneroso para la relación entre padre e hijo que cinco o más inyecciones diarias de insulina.

Cualquier niño con diabetes, sea cual sea su edad y cumpliendo los requisitos mencionados, es potencial candidato para llevar una bomba de insulina. En pediatría la principal razón para recomendar la terapia con bomba de insulina es la mejora de la calidad de vida de la persona con diabetes y sus cuidadores.

En varios estudios, tras iniciar la terapia con una bomba de insulina, tan sólo un pequeño número de jóvenes ha interrumpido el tratamiento para volver a las inyecciones múltiples diarias.

El paciente es la persona más importante a la hora de controlar la diabetes. Cuanto mayor sea el conocimiento de este sobre el funcionamiento de la enfermedad, mayores serán las posibilidades de conseguir unos niveles de glucemia adecuados y por tanto un mejor nivel de vida.

# Capítulo 3

## 3. Alternativas de Solución

---

Tal como se ha descrito en apartados anteriores, la formación es una parte fundamental en el tratamiento de cualquier tipo de diabetes. También ha quedado demostrado, que se aproxima un hito clave en el tratamiento de la diabetes Tipo 1, la aparición de los nuevos páncreas artificiales (evolución de las actuales bombas de insulina).

Esta situación desemboca en la necesidad de tener herramientas didácticas potentes con las que poder educar sobre los aspectos generales de la enfermedad y al mismo tiempo introducir también al paciente en el manejo de los nuevos dispositivos que se utilizarán en su tratamiento.

Por todo esto, se ha querido desarrollar una plataforma educativa que muestre el funcionamiento básico de la diabetes (glucemias, ingestas, insulina...), así como el funcionamiento de los nuevos páncreas artificiales.

A continuación, se muestran las alternativas de solución a las diferentes secciones de la plataforma educativa planteada.

### 3.1. Hardware

En la plataforma educativa se requiere un hardware que realice dos funciones principales; la primera es que represente la función de interfaz de usuario, de forma que se pueda mostrar la información deseada de una forma atractiva e intuitiva, así como, que se pudiese interactuar con el sistema de una forma sencilla. La segunda función, es que el hardware interfaz realice una abstracción al nuevo páncreas artificial, es decir, que en este se implementaran las tareas de control propias de los nuevos páncreas artificiales.

En este punto se valoraron varias posibilidades:

- Una interfaz gráfica en PC sin hardware externo. Donde el paciente interactuase directamente sobre el PC.
- Hardware educativo como *Dash & Dot*. Sistema enfocado al aprendizaje de la programación en edad preescolar o primaria. (Wonder Workshop, 2017)
- Robots humanoides como *Mindstorms EV3* basado en sistema Lego. (Lego, 2017)

De todas las opciones valoradas la que más se ajusta a los requisitos de la plataforma es la de un doble hardware; un PC para el simulador de paciente y un robot tipo oruga basado en Arduino de Makeblock. (Makeblock, 2017)



ILUSTRACIÓN 13 ROBOT TIPO ORUGA DE MAKEBLOCK

Se ha elegido este robot por las siguientes razones:

- **Interfaz de usuario.** Este robot, como una de las ventajas respecto a los otros candidatos, dispone de múltiples accesorios montables sobre él.
  - Los leds de colores montados sobre él sumados a los movimientos del robot oruga permiten representar gran cantidad de información.
  - Los accesorios como los detectores siguelinea o los detectores de proximidad dotan al sistema de una alta capacidad de implementar juegos para facilitar el aprendizaje.
  - Por último, otro accesorio como los “Slider” ofrecen la posibilidad al usuario de interactuar con el sistema directamente en el robot.
- **Abstracción al páncreas artificial, Control.** En la plataforma se necesitaba un sistema que fuera capaz de implementar los cálculos matemáticos correspondientes a un control de proceso, en nuestro caso PID. El robot oruga Makeblock dispone de una placa basada en Arduino Mega 2560, por tanto, disponemos de la posibilidad de programar el robot con la aplicación de Arduino con todas las ventajas asociadas (estándar conocido, múltiples librerías, cálculos matemáticos).
- **Abstracción al páncreas artificial, Comunicación.** Como se ha expuesto en apartados anteriores el robot se comportará internamente como si de un páncreas artificial se tratase, es decir, ejecutará el control del proceso con los datos recibidos de un monitor continuo de glucosa virtual. Este sistema requiere de una comunicación rápida y robusta entre el robot y el PC en el que se ejecute el paciente virtual. Esta comunicación se adapta perfectamente a las posibilidades de comunicación inalámbrica vía bluetooth.

## 3.2. Software

Una vez elegido el hardware para la plataforma: robot oruga y PC, se debe escoger el software en el que se programarán todas las rutinas necesarias para desarrollar el objetivo propuesto. Serán necesarios dos programas diferentes: uno para el robot y otro para el PC.

- **Software Robot.** Nuestro robot oruga dispone de dos posibilidades para programar sus funciones. La primera, un entorno de desarrollo propio de Makeblock, mBlock basado en Scratch 2.0, que dispone de una interfaz intuitiva enfocada a la formación en programación básica.

En segundo lugar, también dispone de la opción de programar directamente sobre el entorno de desarrollo de Arduino el cual dispone de mayor potencia y muchas más posibilidades. Este último fue el entorno elegido.

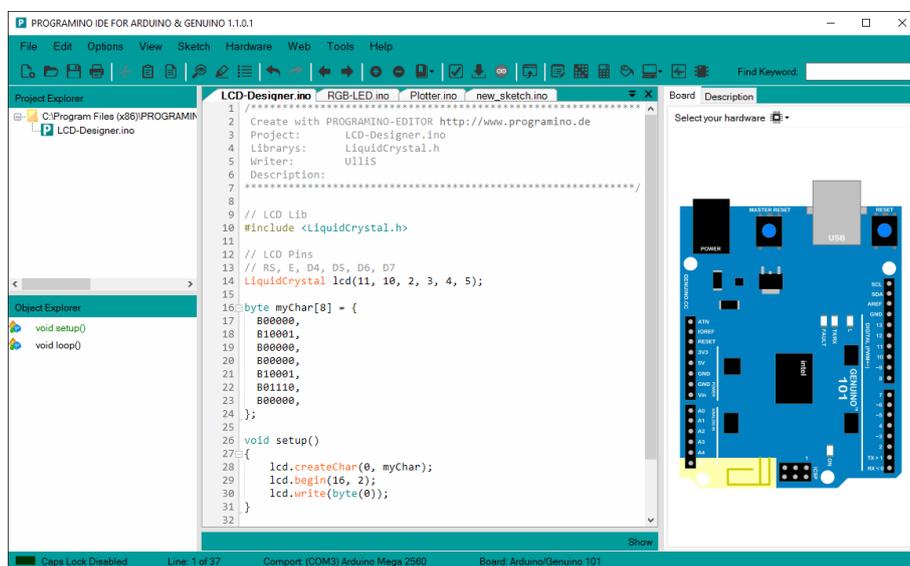


ILUSTRACIÓN 14 INTERFAZ MBLOCK

- **Software PC.** En el PC de la plataforma será implementada una interfaz de usuario donde manejar todo el sistema, así como el simulador del paciente. Para el desarrollo de esta parte del software se valoraron diferentes plataformas de desarrollo de software en Windows como *Visual Basic* o *C#* donde tendríamos una alta potencia gráfica. También se valoró programas de cálculo matemático específicos como *YaCas*, *EDPO* o *Derive* donde se dispondría de herramientas matemáticas potentes necesarias para implementar un paciente virtual (cálculos matemáticos, ecuaciones diferenciales...)

Con los requisitos planteados ninguna de las opciones representaba una solución global.

La opción finalmente elegida fue la herramienta de software matemático Matlab, que ofrece un entorno de desarrollo integrado propio y potentes herramientas para el cálculo matemático. Como ventajas, cabe destacar también su capacidad de conexión con plataformas Arduino o la posibilidad de desarrollar interfaces gráficas de usuario enlazadas con los cálculos matemáticos desarrollados (Guide).

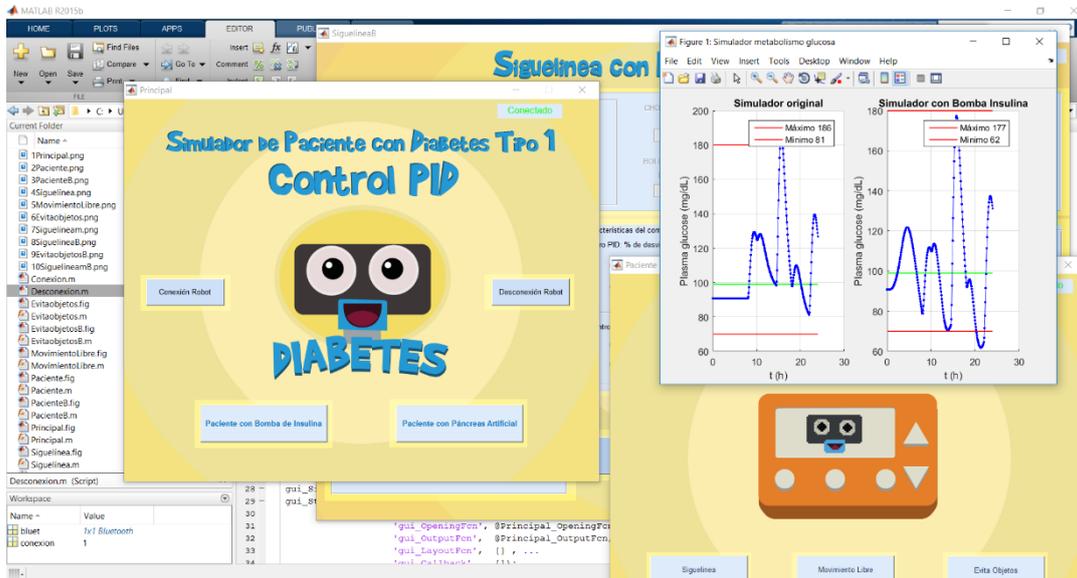


ILUSTRACIÓN 15 ENTORNO MATLAB

### 3.3. Comunicación

El sistema propuesto necesita de una comunicación fluida y estable entre el robot y el PC. Para poder implementar simulaciones atractivas, los datos intercambiados deben ser numerosos y gestionados en tiempo real.

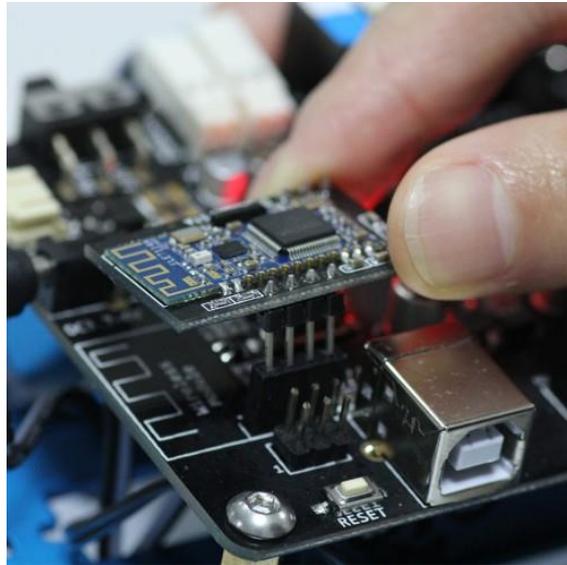
El robot dispone de diversas posibilidades de comunicación:

- **Comunicación serie.** Comunicación por cable directamente entre el robot y el PC. Descartada por la necesidad de dotar al robot de movilidad.
- **Comunicación Wireless.** Como accesorio opcional también se puede instalar en el robot un módulo Wireless que permitiría una conexión de este tipo.



ILUSTRACIÓN 16 MÓDULO WIRELESS PARA ROBOT

- **Comunicación Bluetooth.** El robot tiene instalado de serie un módulo bluetooth. Este protocolo permite movilidad e intercambio de datos de forma continua y fluida.



**ILUSTRACIÓN 17 MÓDULO BLUETOOTH PARA ROBOT ORUGA**

La opción de comunicación finalmente elegida es la comunicación mediante Bluetooth. Las razones para elegir esta tecnología son las siguientes: la movilidad que una comunicación inalámbrica dota al robot, la robustez de este sistema de comunicaciones y la disposición en el robot tipo oruga escogido de un módulo bluetooth ya instalado.

### 3.4. Algoritmos de control

El algoritmo de control es una sección fundamental en el sistema del páncreas artificial. Este algoritmo supone, junto al medidor continuo de glucosa, el paso diferenciador entre la bomba de insulina y el páncreas artificial. Esto es, porque el algoritmo de control debe decidir, por sí solo, con los datos del nivel de glucosa en sangre medidos, la acción de control a aplicar en el sistema (la insulina que se debe suministrar para conseguir el nivel de glucosa deseado).

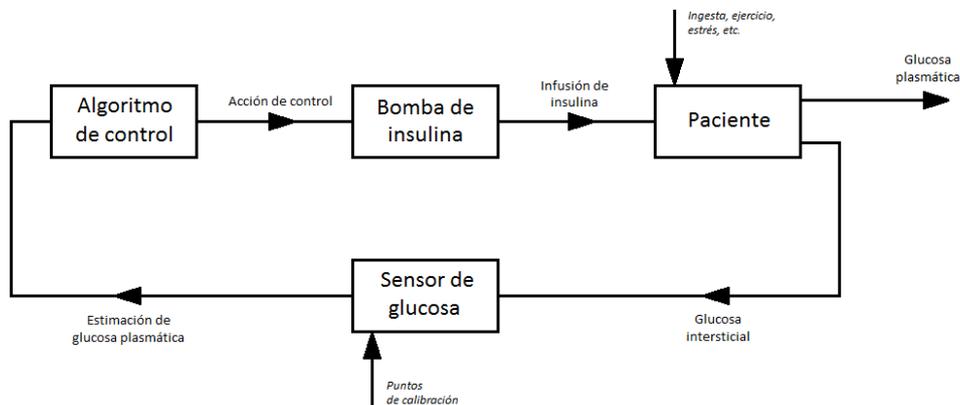


ILUSTRACIÓN 18 EJEMPLO DE SISTEMA DE CONTROL DE PÁNCREAS ARTIFICIAL

Con el fin de resolver el algoritmo de control del sistema, se han propuesto en los últimos años diferentes soluciones con características diferenciadas. A continuación, se describen algunas de las más relevantes:

- **Controlador Adaptativo.** El filtro adaptativo abarca cierto número de topologías básicas que aceptan cambios temporales en los datos de salida del filtro.

Este método por el que se generan estos resultados puede variar en el tiempo. El filtro evalúa continuamente la corrección de su actividad a través de cierta métrica, siendo capaz de alterar su propio esquema de procesado para adecuarse mejor a los criterios de éxito.

Son dos las formas en las cuales la adaptación se ha empleado para recoger el control de la glucemia: estimación de los parámetros del modelo y estimación simultánea del modelo y de sus parámetros. Ambas son variaciones sobre una 'identificación de planta', y ambas han sido integradas en un sistema que emplea el modelo 'actual' de planta para predecir los niveles de glucosa basándose en los valores presentes y pasados de inyecciones de insulina, y asignando un régimen de insulina para tratar con éste. (Ruiz, 2016)

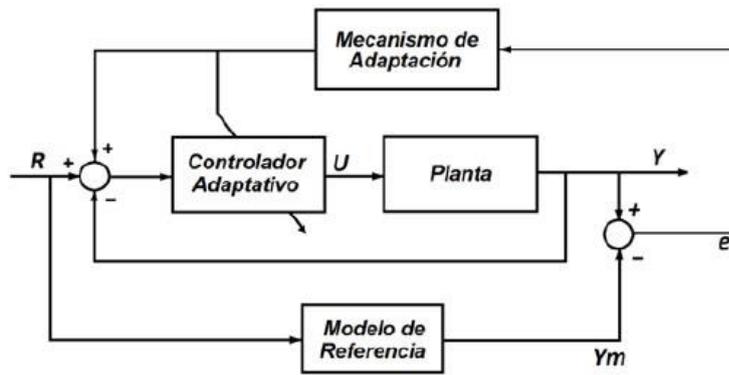


ILUSTRACIÓN 19 CONTROLADOR ADAPTATIVO POR MODELO DE REFERENCIA

- **Controlador Borroso, Redes Neuronales y Sistemas Expertos.** El campo de la lógica borrosa y la teoría de sistemas borrosos se basa en la hipótesis de que algunas relaciones entrada-salida no son rígidas.

Consideremos un proceso en el cual un cambio en el dato de entrada puede dar lugar a tres cambios de diferente magnitud en el dato de salida: bajo, medio y alto. La lógica borrosa indicaría que la salida puede ser una mezcla de medio y bajo, por ejemplo.

Frecuentemente, los sistemas biológicos son no lineales y su modelado matemático es complejo e incluso imposible. En cualquier caso, la lógica borrosa es de base empírica y libre de modelos, abriendo posibilidades para sistemas de control que serían normalmente inviables para la automatización.

Además, la lógica borrosa es muy robusta y no requiere de entradas precisas y libres de ruido para generar salidas útiles. Finalmente, puede modificarse fácilmente y sobre ella pueden realizarse ajustes finos durante la operación. Los sistemas expertos se basan en reglas proveídas por 'expertos' que tienen fácil acceso al conocimiento sobre el sistema. Este tipo de modelos se emplea frecuentemente como protocolo para la administración de insulina en el cuidado intensivo, por ejemplo. En este caso el médico aportaría las reglas, tales como 'si el nivel de glucosa se encuentra entre los valores X e Y, deben administrarse Z unidades de insulina'. Una estrategia de este tipo puede implementarse en un marco de trabajo de lógica borrosa. (Galván, 2014)

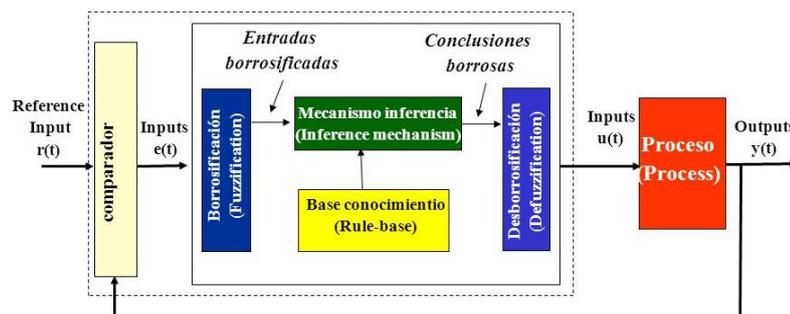


ILUSTRACIÓN 20 EJEMPLO DE CONTROLADOR BORROSO

- **Control Continua-Continua y Control Repetitivo:** El control continua-continua se ha empleado tradicionalmente en ciertos procesos de “batch”<sup>12</sup> en la industria química. Los ciclos de 24 horas de las comidas, las medidas de concentración de la glucosa y la administración del bolus correcto de insulina con el objetivo de conseguir un perfil óptimo de glucosa en sangre, puede observarse desde el mismo ángulo que ciertos procesos “batch” tradicionales.

La adaptación de un controlador continua-continua utiliza datos de pacientes recogidos durante un período predefinido y estos se emplean para actualizar los parámetros del paciente durante el siguiente período de control. (Owens, 2006)

- **Controlador Predictivo Basado en Modelo (MPC).** El control predictivo permite manipular las variables del sistema para establecer una trayectoria deseada a futuro. Un optimizador busca el mejor conjunto de señales de control presentes y futuras para mantener las salidas en los valores deseados en un determinado horizonte de predicción. La ventaja principal de este control es el hecho de que permite que la ranura de tiempo inmediata sea optimizada, mientras tiene en cuenta también las ranuras de tiempo futuras.

En el caso que nos ocupa, el sistema se reduce a una sola salida medida, la concentración de glucosa, y una sola variable manipulada, el flujo de infusión de insulina. Sin embargo, los retos principales para el controlador son la variabilidad intrapaciente, que dificulta enormemente la predicción de la concentración de glucosa y el rechazo de grandes perturbaciones como son la comida, el ejercicio o el estrés.

Una de las principales ventajas del esquema de control MPC basado en optimización es la facilidad para incorporar de manera natural las restricciones en la infusión de insulina que recomienda la práctica clínica. (Bondia, 2010)

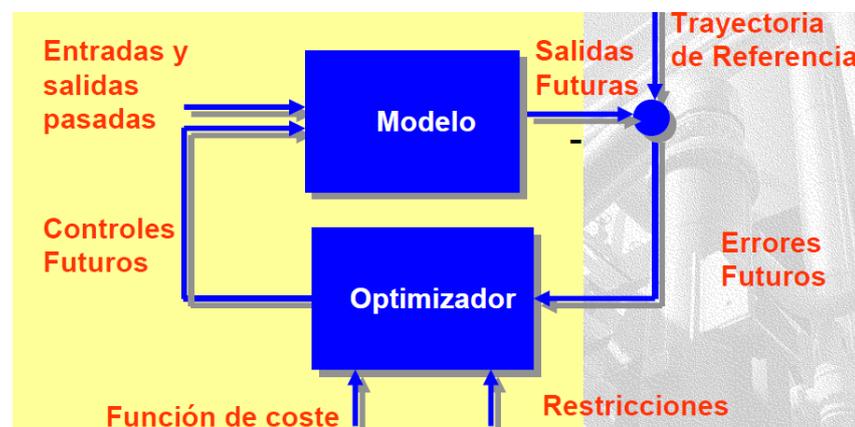


ILUSTRACIÓN 21 ESTRUCTURA BÁSICA MPC

<sup>12</sup> Se conoce como sistema por lotes (en inglés batch processing), o modo batch, a la ejecución de un programa sin el control o supervisión directa del usuario.

- **Controlador PID.** El enfoque PID ha sido representado por Medtronic Diabetes y su sistema de infusión de insulina fisiológico externo, conocido por sus siglas en inglés ePID (external Physiologic Insulin Delivery). Con el término “externo” se refiere al tejido subcutáneo tanto para la medición de la glucosa como la infusión de insulina. El algoritmo ePID es en esencia un algoritmo de control proporcional-integral-derivativo (PID). La ecuación general del algoritmo, utilizando la notación usual, es:

$$u(t) = K_P e(t) + \frac{K_P}{\tau_I} \int_0^t e(\tau) d\tau + K_P \tau_D \frac{de(t)}{dt}, \quad (4.1)$$

Donde  $u(t)$  es la velocidad de infusión de insulina y  $e(t)$  es la señal de error.

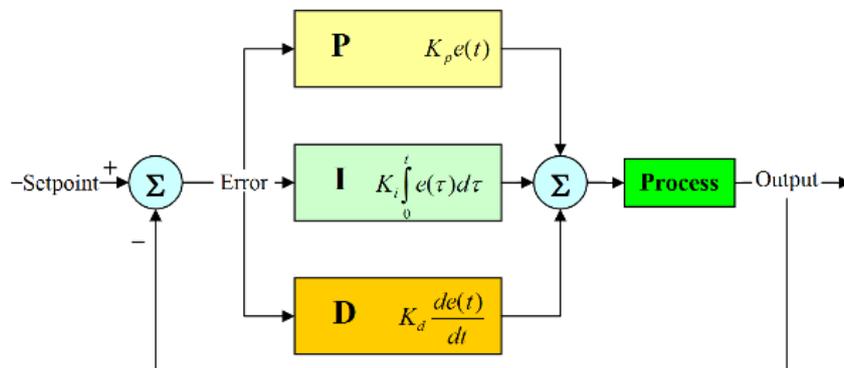


ILUSTRACIÓN 22 ESTRUCTURA BÁSICA PID

Para escoger el algoritmo de control se han tenido en cuenta tres cuestiones principales:

- **La facilidad de implementación.** El algoritmo será programado en una placa Arduino mediante código C por lo que es necesaria cierta sencillez en el desarrollo del algoritmo.
- **Ligereza de los cálculos.** La capacidad de la placa Arduino es limitada y se requiere de rapidez en los cálculos, por lo que otra condición del algoritmo es que este se comporte de manera “ligera” en cuanto a consumo de recursos.
- **Respuesta satisfactoria.** Evidentemente es necesario también que la respuesta del sistema al algoritmo de control sea suficientemente óptima para conseguir los resultados deseados

Teniendo en cuenta estas condiciones se ha seleccionado el algoritmo PID para ejecutar las tareas de control relacionadas con la plataforma. Este algoritmo es fácilmente implementable en código C de una manera “ligera” y obteniendo unos resultados suficientes para el propósito deseado.



# Capítulo 4

## 4. Modelo de Paciente

---

La plataforma educativa implementada hace uso de un modelo de paciente virtual. Este modelo de paciente simula la reacción de un paciente ante estímulos como los hidratos de carbono ingeridos o la insulina. Así mismo, permite realizar todas las pruebas deseadas, sin ningún peligro, administrando patrones de hidratos e insulina.

Este modelo de paciente permite realizar test con hidratos de carbono e insulina sobre un paciente virtual y que éste reaccione con niveles de glucosa en sangre que se aproximarían a los de un paciente real.

Aunque no sea el objetivo del presente proyecto, debido a la importancia del modelo de paciente en el sistema implementado, a continuación, se realiza una breve introducción al funcionamiento del mismo.

Los modelos fisiológicos del sistema de insulina-glucosa para la diabetes tipo 1 suponen tres subprocesos principales (Laguna Sanz, 2014):

- **Modelado de Absorción de Insulina.** Este modelo representa la forma en que la Insulina exógena se introduce en el organismo.
- **Modelo de Absorción de Glucosa.** Este modelo incluye los procesos de ingestión, digestión y absorción de glucosa y otros nutrientes desde el intestino hasta la sangre.
- **Modelo de Regulación de Glucosa.** Este modelo representa la regulación endógena de la glucosa.

El modelo de paciente utilizado se basa en el modelo endógeno propuesto por Cambridge. Un modelo endógeno enfocado en la acción de la insulina y su efecto sobre la glucosa en sangre. (Hovorka, 2002)

## 4.1. Modelo Cambridge

A continuación, se expondrá una breve definición del modelo endógeno de Cambridge en base a los tres subprocesos del modelo fisiológico que ya se ha comentado.

### 4.1.1. Modelo de absorción de glucosa

En este punto se propone un modelo donde el sistema gastrointestinal está modelado con dos comportamientos idénticos con la misma velocidad de transferencia. Las ecuaciones del subsistema son:

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

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

Donde:

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

### 4.1.2. Modelo de absorción subcutánea de insulina

El modelo muestra un flujo simple de insulina a través de dos compartimentos en el torrente sanguíneo y sólo considera la eliminación de insulina de la sangre. Las ecuaciones del modelo son las siguientes:

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

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

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

Donde:

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

$$u_i = \frac{S_2}{t_{\text{maxI}}} \quad (5.6)$$

#### 4.1.3. Modelo del sistema glucoregulador

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

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

Las ecuaciones que presentan el modelo son las siguientes:

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

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

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

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

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

Donde:

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

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

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

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

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

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

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

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

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

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

Los valores nominales de los parámetros del modelo se muestran en la Ilustración 23. Estos parámetros mostrados son valores medios de los diferentes conjuntos de parámetros publicados. (Anaya Oliver, 2014)

Parámetro	Valor	Unidades
$k_{12}$	0,066	$\text{min}^{-1}$
$v_g$	0,16	$\text{L kg}^{-1}$
$e_{gD_0}$	0,0161	$\text{mmol kg}^{-1} \text{min}^{-1}$
$f_{01}$	0,0097	$\text{mmol kg}^{-1} \text{min}^{-1}$
$k_e$	0,138	$\text{min}^{-1}$
$v_i$	0,12	$\text{L kg}^{-1}$
$k_{a1}$	0,006	$\text{min}^{-1}$
$k_{a2}$	0,06	$\text{min}^{-1}$
$k_{a3}$	0,03	$\text{min}^{-1}$
$s_{it}$	$51,2 \cdot 10^{-4}$	$\text{mU L}^{-1} \text{min}^{-1}$
$s_{id}$	$8,2 \cdot 10^{-4}$	$\text{mU L}^{-1} \text{min}^{-1}$
$s_{ie}$	$520 \cdot 10^{-4}$	$\text{mU L}^{-1} \text{min}^{-1}$
$a_g$	0,8	sin unidades
$t_{maxG}$	40	min
$t_{maxI}$	55	min

**ILUSTRACIÓN 23 VALORES NOMINALES DE LOS PARÁMETROS DEL MODELO CAMBRIDGE**



# Capítulo 5

## 5. Plataforma Didáctica

---

Como se ha comentado en apartados anteriores, con el objetivo de ofrecer una herramienta didáctica en relación con la diabetes, se ha implementado una plataforma educativa con una interfaz de usuario atractiva e intuitiva.

La plataforma se puede dividir en 3 apartados: en el primero se presentará la aplicación desarrollada en Matlab, la interfaz de usuario y los cálculos del paciente virtual, el segundo fragmento son todas las rutinas implementadas en el robot tipo oruga y por último las características del protocolo de comunicaciones implementado para el intercambio de información entre los dos elementos anteriores.

La información intercambiada, como veremos más adelante, consta principalmente de datos de nivel de glucosa del paciente para el caso de comunicación PC->Robot y acción de control para actuar sobre el nivel de glucosa en la comunicación Robot->PC.

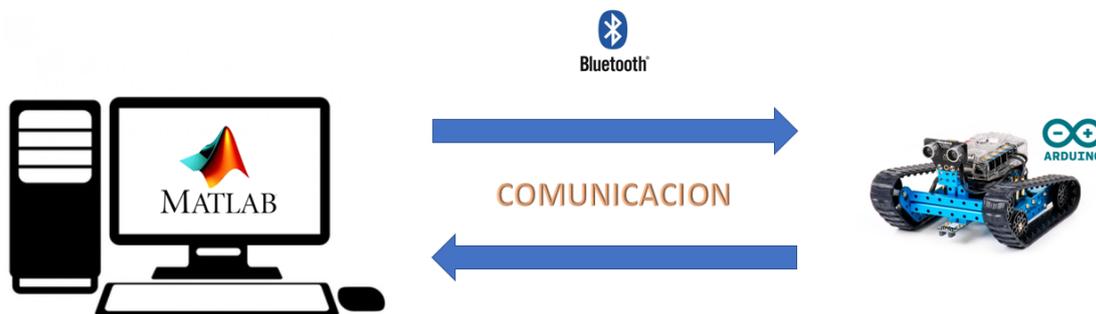


ILUSTRACIÓN 24 ESTRUCTURA PRINCIPAL DE LA PLATAFORMA

En los siguientes subapartados se detalla la estructura de dicha aplicación.

## 5.1. Software PC. Matlab

En primer lugar, se presenta la sección de la plataforma desarrollada en Matlab.

Esta sección se divide a su vez en: la interfaz de usuario implementada en el GUIDE<sup>13</sup> de Matlab y las instrucciones necesarias para desarrollar el paciente virtual.

### 5.1.1. Interfaz de Usuario

En la interfaz de usuario de la plataforma educativa se ha tratado de priorizar en la facilidad de uso para el usuario y en un diseño atractivo para los usuarios más jóvenes.

Para describir la interfaz se van a presentar las diferentes pantallas desarrolladas y sus principales características.

#### Pantalla Principal



ILUSTRACIÓN 25 PANTALLA PRINCIPAL

La pantalla principal es el primer contacto de usuario con la aplicación. Esta sirve de presentación de la plataforma.

En ella se ha implementado el acceso a las funciones de conexión y desconexión, que establecen una comunicación abierta entre PC y Robot. En el margen superior derecho se dispone de un indicativo, común a todas las pantallas, que informa del estado de esta conexión.

---

<sup>13</sup> GUIDE (entorno de desarrollo de GUI) proporciona herramientas para diseñar interfaces de usuario para Apps personalizadas. Mediante el editor de diseño de GUIDE, es posible diseñar gráficamente la interfaz de usuario.

Las GUI (también conocidas como interfaces gráficas de usuario o interfaces de usuario) permiten un control sencillo (con uso de ratón) de las aplicaciones de software, lo cual elimina la necesidad de aprender un lenguaje y escribir comandos a fin de ejecutar una aplicación.

Desde esta pantalla tenemos acceso a las pantallas de paciente.

### Pantallas de Paciente



ILUSTRACIÓN 26 PACIENTE CON BOMBA DE INSULINA Y PACIENTE CON PÁNCREAS ARTIFICIAL

Las pantallas de paciente, a las que se accede desde la pantalla principal, representan los dos modos de funcionamiento de la plataforma:

1. **Simulación de paciente con tratamiento mediante Infusión de Insulina Subcutánea Continua (CSII)**, comúnmente conocido como tratamiento con bomba de insulina. Como se ha explicado anteriormente la bomba reemplaza la necesidad de múltiples inyecciones mediante el suministro de dosis precisas de insulina de acción rápida las 24 horas del día. En nuestro caso para facilitar la comprensión de los modelos, se ha obviado el suministro de la insulina basal, por lo que tan solo son necesarios los bolos de insulina relacionados con las ingestas de hidratos de carbono (el valor de glucosa en sangre se mantiene estable si no hay ingestas de hidratos).
2. **Simulación de un paciente con tratamiento mediante Páncreas Artificial.**

Se han implementado diferentes juegos para mostrar el funcionamiento de ambos modos. Estos juegos presentan la gestión de la enfermedad de forma atractiva y sencilla para facilitar el aprendizaje de los pacientes más jóvenes.

A continuación, se presentan las pantallas de interacción de los diferentes juegos en ambos modos.

### Pantalla Siguelinea



ILUSTRACIÓN 27 SIGUELINEA CON BOMBA DE INSULINA

El primer juego presentado es el “robot siguelinea”. En él, el robot tipo oruga se desplaza siguiendo una línea. Paralelamente a la acción sigue-línea por parte del robot, se ejecuta un simulador de paciente en Matlab.

Los movimientos del robot sobre la línea se verán afectados por los resultados del simulador de paciente en “tiempo real” (se profundizará en los patrones de funcionamiento del robot en el apartado de software del robot).

En este primer juego se solicita al usuario, en primer lugar, la introducción de los valores de carbohidratos ingeridos durante el día, así como los bolus de insulina relacionados con dichas ingestas.

Con la acción “Iniciar siguelinea” se ejecuta la función “Siguelinea.m” (la cual también describiremos más adelante) y se inicia el simulador de paciente.

### Pantalla Movimiento libre

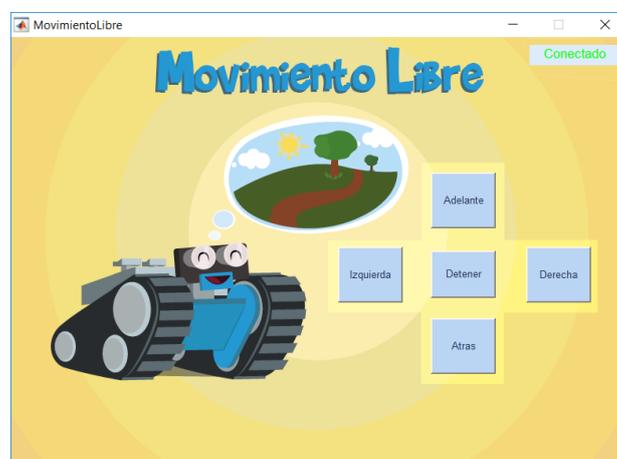


ILUSTRACIÓN 28 PANTALLA MOVIMIENTO LIBRE

El siguiente juego desarrollado ofrece la posibilidad al usuario de interactuar directamente con el robot tipo oruga.

Para acceder a la pantalla “movimiento libre” es necesario que la conexión entre PC y robot esté activa. Al iniciar esta pantalla se le envía al robot el tipo de acción que realizará, en este caso, acciones de movimiento teledirigidas. Desde la interfaz, mediante los botones adelante, derecha, atrás, izquierda y detener se mandan las acciones correspondientes al robot.

### Pantalla Evita Objetos



ILUSTRACIÓN 29 PANTALLA EVITA OBJETOS

El juego “Evita Objetos”, haciendo uso de los sensores instalados en el robot, consiste en el movimiento continuo del robot evitando los posibles objetos encontrados en su camino. Al igual que para el juego siguelinea, paralelamente a los movimientos del robot, se ejecuta en Matlab un simulador de paciente con los datos adquiridos en el interfaz.

Los movimientos del robot, como veremos en su software, están influenciados “en tiempo real” según el nivel de glucosa en sangre (resultado de la simulación del paciente).

Con la acción “Inicia Evita-Objetos” se ejecuta la función “Siguelinea.m” y por tanto se inicia el simulador de paciente y el movimiento del robot.

## Pantalla Siguelinea Alimentación Manual

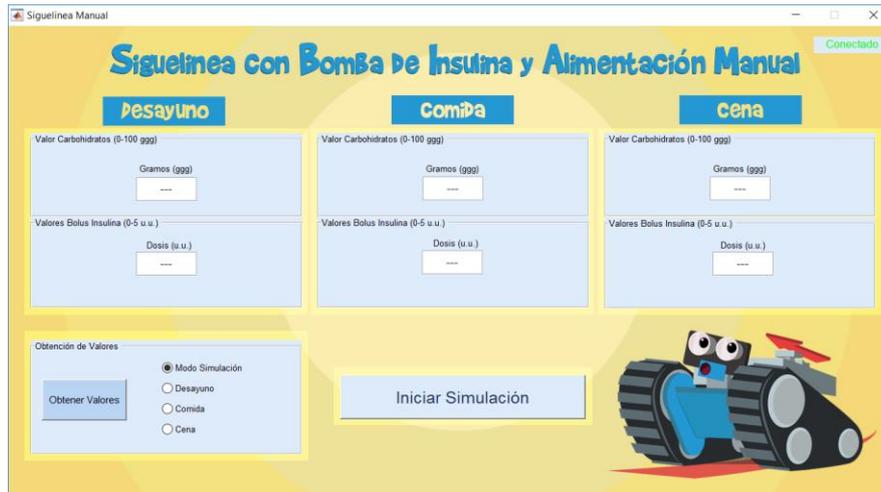


ILUSTRACIÓN 30 PANTALLA SIGUELINEA ALIMENTACIÓN MANUAL

El siguiente juego propuesto tiene el mismo comportamiento tanto del simulador como del robot que el del “Siguelinea”, la diferencia reside en la obtención de los valores de las ingestas de hidratos de carbono y de los bolus de insulina.

Los patrones de ingestas e insulina son adquiridos por el robot tipo oruga mediante los potenciómetros de los que dispone. “Slider” (Ilustración 31).



ILUSTRACIÓN 31 POTENCIÓMETRO ROBOT ORUGA

En la parte inferior izquierda de la pantalla se encuentra el botón “Obtener Valores”. Seleccionando cualquiera de las tres ingestas: Desayuno, comida o cena, y pulsando el botón “Obtener Valores” se ejecuta un bucle cada 300ms que pide los datos de ingesta de hidratos y bolus de insulina al robot. El robot los ha adquirido de sus potenciómetros.

## Pantalla Siguelinea con Páncreas Artificial

SiguelineaB

### Siguelinea con Páncreas Artificial

Conectado

Valores carbohidratos y bolus

Ejemplo 1: Cinco comidas en un día  
 Ejemplo 2: Cinco comidas en un día disminuyendo dosis insulina  
 Ejemplo 3: Cinco comidas en un día aumentando dosis insulina  
 Ejemplo 4: Cuatro días con tres comidas diarias (consumo regular)  
 Ejemplo 5: Cuatro días con tres comidas diarias (consumo irregular)  
 Introducir valores:

CHO

Día (dd) Horas (hh) Minutos (mm) Gramos (ggg) >>

BOLUS

Día (dd) Horas (hh) Minutos (mm) Dosis (u.u) >>

Parámetros de simulación

Días de simulación: 1

Intervalo de simulación: 20

Características del controlador

Filtro PID: % de desviación sobre la basal de equilibrio

Sin límites  
 Establecer límite: 0.5 / 1.5 (tanto por uno)

Control Bolo

Deficit/Exceso bolo: 100 %  
 Sin alterar

Parámetros PID

K I D

Referencia

Ref

Obtención de Parámetros PID y Referencia

Modo Simulación  
 Parámetros K, I y D  
 Referencia

ILUSTRACIÓN 32 PANTALLA SIGUELINEA CON PÁNCREAS ARTIFICIAL

Los mismos juegos desarrollados para el modo de tratamiento con bomba de insulina han sido desarrollados para el caso de que el paciente sea tratado con un “páncreas artificial”.

El simulador de paciente sigue estando en el PC (Matlab) pero se añade al sistema un controlador PID que se ejecutará en el robot tipo oruga.

El controlador tratará de ajustar el nivel de glucosa en sangre del paciente. Para ello requiere del usuario los parámetros del regulador (K, I y D) así como la referencia (Ref.) a la que se quiere que se ajuste el valor de glucosa en sangre. Requiere también por parte del simulador el valor de la glucosa en sangre en cada iteración de la simulación.

El controlador PID por su parte, proporcionará al simulador una señal de control para ajustar el nivel de insulina del paciente (este proceso será explicado con más detalle en la función “SimuladorB.m”).

Tal como se ha comentado, el usuario debe suministrar los parámetros del controlador y la referencia. Para ello, se utiliza un método similar a la introducción de datos en el juego “Siguelinea Alimentación manual”, se utilizan los “Slider” del robot.

Al pulsar sobre el botón “Iniciar Siguelinea”, se inicia la función “SimuladorB.m”, en la que se ejecuta un paciente con bomba de insulina y otro con páncreas artificial para realizar la comparación gráfica.

## Pantalla Evita Objetos con Páncreas Artificial



ILUSTRACIÓN 33 PANTALLA EVITA OBJETOS CON PÁNCREAS ARTIFICIAL

El siguiente juego tiene un comportamiento similar al de Evita Objetos con bomba de insulina, con la diferencia de que se suministran por parte del usuario los parámetros del controlador y la referencia del sistema.

Los parámetros son introducidos también mediante los “Slider” del robot. La función ejecutada también será “SimuladorB.m”.

## Pantalla Siguelinea Alimentación Manual con Páncreas Artificial



ILUSTRACIÓN 34 PANTALLA SIGUELINEA ALIMENTACIÓN MANUAL PÁNCREAS ARTIFICIAL

El último juego propuesto ofrece un comportamiento en el robot similar al “Siguelinea”. La adquisición de datos es también mediante los “Slider” y para este caso se ejecuta la función “SimuladorB.m”.

## 5.1.2. Software Paciente Virtual

Todos los juegos que hemos mencionado hacen uso de la función “Simulador.m” o “SimuladorB.m” para simular el comportamiento fisiológico de un paciente virtual. En estas funciones principalmente se ejecuta un paciente virtual, se envían los datos necesarios al robot oruga y se muestran resultados. Todo ello en “tiempo real” respecto a los patrones de movimiento del robot oruga.

A continuación, se describe brevemente el funcionamiento de ambas funciones.

### Simulador.m

La función Simulador es la correspondiente al paciente virtual cuando este es tratado mediante una bomba de insulina.

Su funcionamiento está afectado tan solo por los patrones de hidratos de carbono ingeridos y los bolus de insulina suministrados en relación con estas ingestas.

Los parámetros requeridos en esta función son, por tanto:

- Patrones de ingesta de hidratos de carbono
- Bolus de Insulina
- Parámetros relacionados con la simulación (días de ejecución de la simulación, intervalo de simulación, ...)
- Actividad que realizará el robot tipo oruga (siguelinea, evita-objetos, ...)

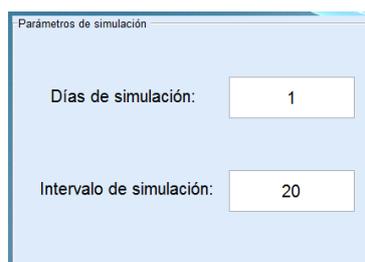
Con el fin de exponer el funcionamiento del programa, este se puede dividir en tres secciones principales:

#### 1. Cálculos matemáticos para simular el paciente virtual.

En esta sección se implementa el bucle que calcula el nivel de glucosa en sangre.

Para realizar estos cálculos se hace uso del modelo endógeno de Cambridge, sobre el que ya se ha hablado. Haciendo uso de la instrucción “ode45” se resuelven las ecuaciones diferenciales propuestas en modelo Cambridge y se obtiene entre otros el nivel de glucosa en sangre estimado.

El cálculo es realizado durante el tiempo indicado por el usuario (días de simulación) y a intervalos de tiempo también indicados por el usuario (Intervalo de simulación). Tomamos un ejemplo concreto para explicar el concepto:



Parámetros de simulación

Días de simulación:

Intervalo de simulación:

ILUSTRACIÓN 35 PARÁMETROS DE SIMULACIÓN

Para este ejemplo concreto la simulación se realizará durante 24 horas y los cálculos de glucosa en sangre serán desarrollados cada 20 minutos. Por tanto, para este caso, el bucle de cálculo de glucosa tendrá 72 iteraciones.

## 2. Envío de datos al robot oruga.

La función Simulador.m (también la SimuladorB.m en su caso) es la encargada del envío y recepción de datos del robot.

Al arrancar la función envía el tipo de juego que se ha seleccionado. El robot tendrá un comportamiento u otro dependiendo de esta selección.

La siguiente tarea que realiza la función es la de enviar el nivel de glucosa en sangre estimado al robot. En cada iteración del bucle de cálculo se envía el nivel. El robot tendrá también diferentes patrones de comportamiento según el nivel de glucosa en sangre estimado.

Para el caso concreto de la Ilustración 35, en cada una de sus 72 iteraciones será enviado el dato de glucosa al robot para que este actúe en consecuencia.

## 3. Representación de información en pantalla.

Por último, la función también representa los valores más importantes de la simulación de manera gráfica.

Mientras se realiza la simulación, se dibuja sobre una gráfica: umbrales de hipoglucemia e hiperglucemia, el valor en cada iteración de la glucosa y el máximo y mínimo valor de glucosa de la simulación. En la siguiente Ilustración se muestra un ejemplo de gráfica de valores obtenidos.

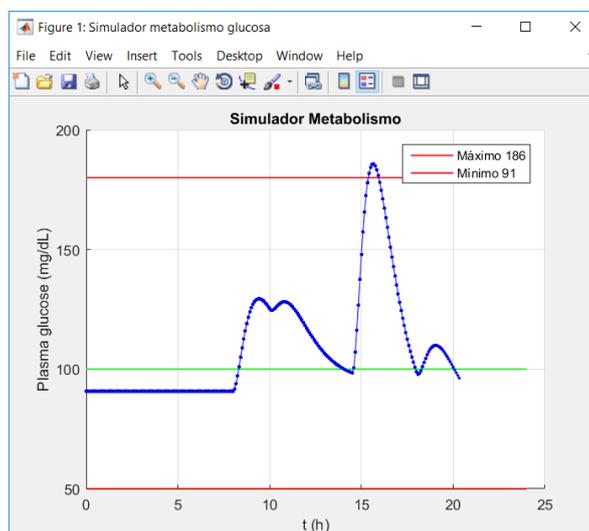


ILUSTRACIÓN 36 SIMULADOR CON BOMBA DE INSULINA

Cuando finaliza la simulación se muestra por pantalla los porcentajes de tiempos que el paciente ha sufrido hipoglucemia, hiperglucemia o normoglucemia.

En la plataforma se han escogido valores de hipoglucemia e hiperglucemia que se ajustasen a los recomendados por los organismos oficiales y que a su vez resultaran educativos en nuestra herramienta. Los umbrales seleccionados son:

El valor sobre el cual definimos que el paciente se encuentra en estado de hiperglucemia es 180 mg/dL.

El valor por debajo del cual definimos que el paciente se encuentra en hipoglucemia es 70 mg/dL.

## SimuladorB.m

La función SimuladorB es la correspondiente al paciente virtual cuando este es tratado mediante un páncreas artificial.

Su funcionamiento está afectado por: los patrones de hidratos de carbono ingeridos, los bolus de insulina suministrados en relación con estas ingestas y la acción de control propuesta por el robot.

Los parámetros requeridos en esta función son, por tanto:

- Patrones de ingesta de hidratos de carbono
- Bolus de Insulina
- Parámetros relacionados con la simulación (días de ejecución de la simulación, intervalo de simulación, ...)
- Actividad que realizará el robot tipo oruga (siguelínea, evita-objetos, ...)
- Señal de control enviada por el robot.

Esta función realiza dos simulaciones de paciente virtual.

La primera simulación, idéntica a la de la función "Simulador.m" ejecuta un paciente con bomba de insulina (con los patrones de ingesta de alimentos y de bolus prandiales suministrados por el usuario).

La segunda simulación de paciente se ejecuta con el supuesto de que el paciente es tratado mediante un páncreas artificial. El objetivo de la simulación doble es la comparación gráfica de las dos terapias.

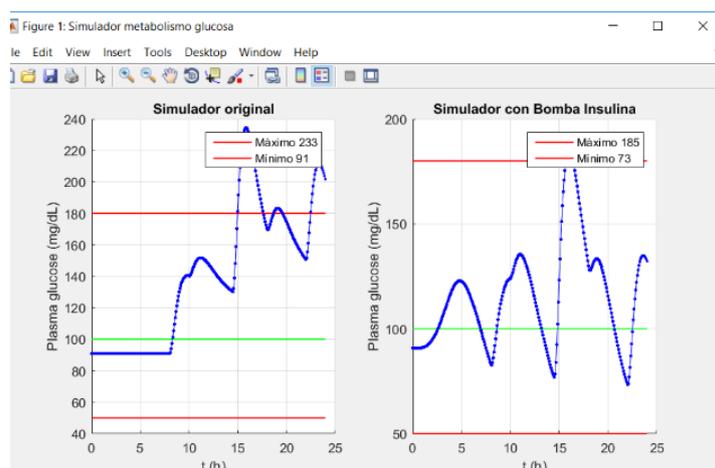


ILUSTRACIÓN 37 COMPARACIÓN PACIENTE EN AMBOS TRATAMIENTOS

En la simulación con páncreas artificial es mandado, en cada iteración, el valor de glucosa al robot. En el robot, para este caso, serán realizadas dos tareas: la gestión del nivel de glucosa, que actuará sobre los patrones de movimiento de éste, y el cálculo de la señal de control (PID) recomendada para el ajuste del nivel de glucosa a la referencia. La señal de control actuará directamente sobre el nivel de insulina en sangre en el paciente virtual (páncreas artificial).

## 5.2. Software Robot. Arduino

En la plataforma educativa presentada, el robot tipo oruga tiene un papel fundamental.

Este realiza dos papeles esenciales: la representación del nivel de glucosa en sangre del paciente virtual mediante la representación de unos efectos ficticios pero visibles y el del controlador PID que ajusta el nivel de glucosa del paciente en base a una referencia.

### 5.2.1. Modos de Operación

Como se ha comentado anteriormente, el robot se comporta de manera diferente según el juego seleccionado en la interfaz gráfica y según el nivel de glucosa en cada instante de la simulación del paciente.

Según la indicación que se reciba por parte del PC, el robot adoptará diferentes modos de operación:

#### **Operación Siguelinea.**

Si el robot se encuentra en modo siguelinea, al comenzar la simulación, este seguirá una línea negra sobre fondo blanco. Para ello hace uso de dos sensores detectores de línea. Se trata de sensores emisores y receptores de luz que diferencian una superficie clara de una oscura tal como se puede observar en la siguiente Ilustración.

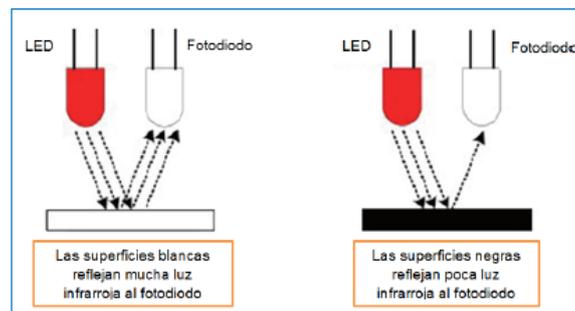


ILUSTRACIÓN 38 FUNCIONAMIENTO DEL MÓDULO DETECTOR DE LÍNEA

Según las señales recibidas en estos sensores el robot realizará los movimientos oportunos como veremos más adelante.

El modo de operación siguelinea puede requerir control PID o no, según el juego elegido en el interfaz de usuario.

Para los casos en los que es necesario el control PID (simulador de paciente con páncreas artificial), se ha implementado un algoritmo que ejecuta los cálculos de un control PID y devuelve una señal de control que será enviada al PC.

Con el fin de implementar el algoritmo de control PID se hace uso de la fórmula 4.1. Necesitaremos, para poder programar el algoritmo, discretizar la ecuación del regulador (fórmula 6.1). Una vez discretizada se calculará la acción de control para cada una de las iteraciones de la simulación.

$$u_t = K_p \cdot e_{(t)} + K_i \cdot T_s \cdot \sum_{k=0}^t e_k + K_d \cdot \frac{e_{(t)} - e_{(t-1)}}{T_s} \quad (6.1)$$

Donde  $e(t)$  es el error entre la referencia y el nivel de glucosa en el instante  $t$ ,  $T_s$  el periodo de muestreo de la señal y  $K_p$ ,  $K_i$  y  $K_d$  son la ganancia proporcional, integral y derivativa del regulador, respectivamente.

En la siguiente ilustración se muestra el algoritmo desarrollado.

```
eact= ref-glucosa;
esum+=eact;

control=(factork*eact) + factori*tmuestreo*esum - factord*(eact-eant)/tmuestreo;

desmontarfloat(control, paquetee);
```

**ILUSTRACIÓN 39 CÓDIGO PID**

Los parámetros del PID ( $K_p$ ,  $K_i$  y  $K_d$ ) son seleccionados por el usuario, así como la referencia; todos ellos mediante los “slider” montados sobre el robot.

La señal de control obtenida es finalmente enviada al simulador de paciente virtual.

### **Operación Siguelinea con Alimentación Manual**

Sigue el mismo proceso que la operación Siguelinea con el añadido que envía los datos de los “slider” de las ingestas de comida y los bolus cuando se le solicita.

### **Operación Movimiento Libre**

En este modo el robot se comporta como un vehículo teledirigido. Se encuentra a la espera de recibir las direcciones desde el PC para emprender la marcha.

### **Operación Evita-Objetos**

En el modo evita objetos, el robot se mueve a distintas velocidades según el nivel glucémico del paciente virtual. Este también posee la capacidad de evitar cualquier objeto que se encuentre (la capacidad de esquivar objetos también se verá afectada por el nivel glucémico como veremos más adelante) haciendo uso de su módulo de sensores de ultrasonidos.

Este módulo, gracias a sus dos sensores, es capaz de devolver un valor con la distancia aproximada entre el robot y un objeto próximo a él.

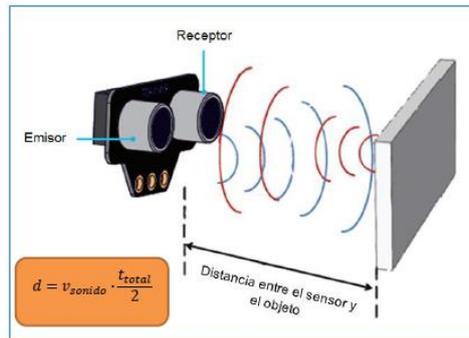


ILUSTRACIÓN 40 MÓDULO ULTRASONIDOS

### 5.2.2. Gestión de Movimientos

El robot tipo oruga realizará tres tipos de movimiento: movimientos siguelinea, movimiento libre y evita-objetos.

Estos movimientos estarán afectados por el nivel de glucosa en sangre, como ya se ha comentado. Se han seleccionado tres estados de funcionamiento: hipoglucemia (glucosa en sangre < 70 mg/dL), hiperglucemia (glucosa en sangre > 150 mg/dL) y normoglucemia (niveles de glucosa entre 70 y 150 mg/dL).

#### Movimiento siguelinea

Este movimiento lo realizarán los juegos: Siguelinea, Siguelinea con Páncreas, Siguelinea Alimentación Manual y Siguelinea Alimentación Manual con Páncreas.

La función de movimiento implementada recibirá como parámetros valores que afectarán a su funcionamiento (según el estado glucémico).

Según el nivel glucémico recibido del PC el robot seguirá la línea con unas particularidades u otras. Haciendo uso de la función "move" se discrimina el tipo de movimiento sobre la línea (Ilustración 41).

```
void move(int direction, int fvelocidad, int fretardo, int fajusterecta, int facentuacurva, int zigzag)
{
    if(direction == 1){ //Oruga inicia marcha recto
        delay(fretardo);
        motorizquierdo = fvelocidad + zigzag;
        motorderecho = fvelocidad - zigzag;
    }else if(direction == 2){ //Oruga gira a la izquierda
        delay(fretardo);
        motorizquierdo = -fvelocidad/fajusterecta;
        motorderecho = fvelocidad + facentuacurva;
    }else if(direction == 3){ //Oruga gira a la derecha
        delay(fretardo);
        motorizquierdo = fvelocidad + facentuacurva;
        motorderecho = -fvelocidad/fajusterecta;
    }else if(direction == 4){ //Oruga inicia marcha hacia atrás
        delay(fretardo);
        motorizquierdo = -fvelocidad;
        motorderecho = -fvelocidad;
    }
}
```

ILUSTRACIÓN 41 FUNCIÓN "MOVE"

En la siguiente ilustración se observan las discriminaciones sobre los parámetros de entrada de la función de movimiento según el estado glucémico del paciente.

Parámetros	Hipoglucemia	Normoglucemia	Hiperglucemia
velocidad	75	105	180
retardo	60	1	1
ajusterecta	1	-10	1
acentuacurva	25	0	0
zigzag	0	0	0

ILUSTRACIÓN 42 PARÁMETROS DE MOVIMIENTO SEGÚN SU ESTADO GLUCÉMICO

La “velocidad” del robot será diferente en cada estado glucémico. El “retardo” aplica un desfase en las acciones realizadas por el robot en estado de hipoglucemia lo que le dota de un aspecto torpe y lento a sus movimientos.

“acentúa” es un instrumento que empeora el comportamiento en curvas cerradas en estado de hipoglucemia dotando al robot de un comportamiento nervioso y errático.

El parámetro “zigzag” en el modo siguelinea está siempre a cero. Este parámetro se utiliza en el modo evitaobjetos que comentaremos más adelante.

Mención aparte merece el parámetro “ajusterecta”. Este parámetro permite realizar giros más suaves en las rectas y así dar la sensación de que el robot sigue la línea con mayor suavidad. Este caso se aplica cuando el paciente se encuentra en normoglucemia o cerca de ella. Simulando un comportamiento de un paciente con niveles de glucosa aceptables.

La problemática del parámetro “ajusterecta” radica en identificar cuando el robot se encuentra frente a una curva cerrada o frente a una recta o curva suave.

Para identificar las curvas y las rectas se hace uso de los sensores detectores de línea. Por defecto, el robot se encuentra en estado de “recta o curva suave”; solo cuando los dos detectores de línea pierden de vista la línea negra se considera que nos encontramos ante una curva cerrada (el robot se ha “salido”) y por tanto se desactiva la ayuda de recta (curvas suaves). El código utilizado se muestra en la siguiente ilustración.

```

if (sensoranterior != Sensor){
    // Si hay un cambio de estado en los sensores

    if (sensoranterior==3 || sensoranterioranterior==3 || Sensor==3){
        curvacerrada=1;
        // Curva cerrada si en el estado actual o los anteriores
        // si los 2 detectores se han salido de la línea
    }
    if ((Sensor==2 && sensoranterior==0 && sensoranterioranterior==1) || (Sensor==1 && sensoranterior==0 && sensoranterioranterior==2)){
        curvacerrada=0;
        // Fin de curva cerrada si uno de los sensores se sale de la línea
        // por un lado cuando en el estado previo estaba dentro de la línea y
        // anteriormente a ese último estado el otro sensor había salido por el otro
        // lado de la línea
    }

    sensoranterioranterior=sensoranterior;
    sensoranterior=Sensor;
}

```

ILUSTRACIÓN 43 DETECCIÓN DE CURVA O RECTA

Se considera que ha terminado el estado de curva cerrada cuando el robot sale de la línea por un lado y en el siguiente giro por el otro lado, se encuentra por tanto en una recta o curva muy suave.

Como hemos visto el comportamiento será diferente en cada estado glucémico.

1. **Normoglucemia.** En el estado de normoglucemia el robot se mueve sobre la línea con los mejores ajustes para lograr un movimiento suave, un trazado de curvas óptimo y una velocidad media.
2. **Hipoglucemia.** En este estado, y teniendo en cuenta la sintomatología en estado de hipoglucemia, se ha tratado, en la medida de lo posible, que el comportamiento del robot sea lento y torpe dando la sensación de fragilidad y desorientación (Ilustración 44).  
La velocidad de movimiento del robot será inferior a la media, se añadirá también un retardo a sus acciones, se acentuará el giro en las curvas (seguirá zigzagueando en las curvas) y no se aplicará la mejora en las rectas.
3. **Hiperoglucemia.** En estado de hiperoglucemia se ha tratado que los movimientos fueran también erráticos, pero en este caso nerviosos y acelerados. Se ha tratado también de representar en la medida de lo posible los síntomas de la hiperoglucemia (Ilustración 44).

SINTOMATOLOGÍA	
HIPOGLUCEMIA	HIPERGLUCEMIA
Debilidad	Nauseas
Inestabilidad	Orinar más
Irritación	Beber más
Confusión	Perder Peso
Mareo	Irritabilidad
Hormigueo	Dolor Abdominal
Dolores de Cabeza	Boca Seca
Fatiga	Hambre
Falta de Coordinación	Visión Borrosa
Dificultad Habla	Dolor de Cabeza
Tristeza	Dificultad Respirar

ILUSTRACIÓN 44 SINTOMATOLOGÍA HIPOGLUCEMIA E HIPERGLUCEMIA (AMERICAN DIABETES ASSOCIATION, 2017)

### Movimiento Libre

Este movimiento lo realizará el juego: Movimiento Libre.

Se recibirán desde el PC las instrucciones sobre la acción a realizar y se realizará el movimiento en consecuencia: detener, giro a izquierda, giro a derecha, marcha adelante, marcha atrás.

La función de movimiento implementada recibirá como parámetros unos valores medios de velocidad y sin ajustes adicionales.

## Movimiento Evita Objetos

Este movimiento lo realizarán los juegos: Evita-Objetos y Evita-Objetos con Páncreas.

El movimiento trata de esquivar los objetos que se encuentra en su camino el robot tipo oruga. Para esquivar los objetos se hace uso del módulo ultrasónico del robot. La distancia de detección está parametrizada según el nivel de glucemia del paciente virtual. A mayor hipoglucemia o hiperglucemia la distancia de detección es menor incluso llegando a colisionar en los casos más extremos.

Estado	Distancia de Detección (cm aprox.)
Hipoglucemia	20
Normoglucemia	$20 - (\text{glucosa} - 100) / 9$
Hiperglucemia	$15 - (100 - \text{glucosa}) / 5$

ILUSTRACIÓN 45 PARAMETRIZACIÓN DE LA DISTANCIA DE DETECCIÓN

También se ha definido el “avance en zigzag”. Este no permite al robot ir recto, le obliga a describir giros, mientras dura el estado que lo ha activado (hiperglucemia o hipoglucemia).

Cuando el robot ha detectado un objeto, este retrocede y realiza un giro para esquivarlo.

```
-----Gestión de movimientos evita-objetos-----  
  
if (objeto==0 && esquivando==0){ //No hay objeto: Inicia marcha recto hacia adelante  
  
    move(1, velocidad, retardo, ajusterecta, acentuacurva, zigzag);  
    inicio=1;  
}  
else{ //Esquiva objeto  
    if (inicio==1) {  
        tiempo=millis();  
        inicio=0;  
        tiempo2=millis();  
        controltiempo=tiempo2-tiempo; //Gestion de tiempos para el retroceso y giro  
  
        if (controltiempo<1000){ // Inicia retroceso  
            move(4, velocidad, retardo, ajusterecta, acentuacurva, zigzag);  
            esquivando=1;  
        }  
        else if (controltiempo>=1000 && controltiempo<2200){ //Inicia giro  
            move(2, velocidad, retardo, ajusterecta, acentuacurva, zigzag);  
            esquivando=1;  
        }  
        else{  
            esquivando=0;}}  
}
```

ILUSTRACIÓN 47 MOVIMIENTOS EVITA OBJETOS

La función utilizada para realizar los movimientos también es la función “move”, mostrada anteriormente.

El comportamiento del robot en este movimiento está afectado por el nivel glucémico como se ha visto, ajustándose en lo posible a los síntomas comentados en la Ilustración 44:

1. **Normoglucemia.** El robot se desplaza a una velocidad media y esquiva los objetos sin retardo y a una distancia “suficiente” (aprox. 20cm).
2. **Hipoglucemia.** La velocidad de avance del robot es más baja, se ha añadido un retardo a todas sus acciones y es incapaz de avanzar en línea recta, su movimiento de avance es en zigzag. La detección de objetos también es errante cuanto menor es el nivel de glucosa en sangre más ajustada es la detección, llegando en los casos más graves a colisionar con los objetos.
3. **Hiperoglucemia.** La velocidad de desplazamiento del robot es más alta, también tiene cierto retardo en sus acciones y su movimiento de avance también es en zigzag. Como en el caso anterior la distancia de detección de objetos está parametrizada según la gravedad de la hiperoglucemia recibida.

Parámetros	Hipoglucemia	Normoglucemia	Hiperoglucemia
velocidad	95	115	160
retardo	200	1	50
ajusterecta	1	1	1
acentuacurva	0	0	0
zigzag	60 y -60	0	60 y -60

ILUSTRACIÓN 46 PARÁMETROS DE MOVIMIENTO SEGÚN ESTADO GLUCÉMICO

### 5.2.3. Gestión Efectos Visuales

Los diferentes patrones de movimiento vinculados a los niveles glucémicos aportan notable información al usuario de cuál es el estado del paciente virtual durante la simulación. Esta información se ha completado de forma visual con el anillo de LEDs RGB montado sobre el robot.

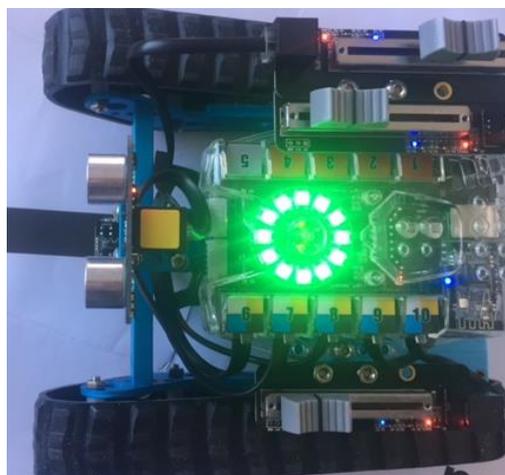


ILUSTRACIÓN 47 ANILLO DE LEDs RGB

Este anillo aporta información de fácil comprensión para los pacientes más jóvenes.

En su programación cada led adquiere un valor en cada una de las tres variables que definen su color (rojo, verde y azul).

En la elección de colores se ha optado por resaltar la gravedad de los casos más extremos de hipoglucemia con el color rojo y el azul para los casos de hiperglucemia.

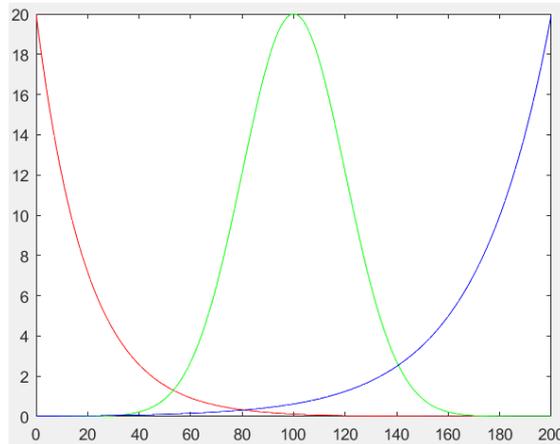


ILUSTRACIÓN 48 CURVAS DE COLORES LEDS

Como vemos en la ilustración 48 se ha tratado de realizar un paso suave entre la escala de colores; de manera que cada nivel glucémico tiene su tono de color. Para ello se ha implementado un algoritmo donde los valores de las variables de los tres colores principales son curvas. El color rojo y azul son curvas exponenciales (para remarcar los extremos glucémicos) y el verde es una campana de Gauss para enfatizar el color verde en los alrededores de 100mg/dL y minimizarlo en los extremos.

```
//-----Azul-----//  
  
if (glucosa>180){ //HIPERGLUCEMIA  
Azul=255;}  
else if (glucosa<100){  
Azul=0;}  
else{  
Azul=pow(2, ((glucosa-50)/20)); // Funcion exponencial inversa  
}  
  
//-----Verde-----//  
  
if (glucosa>180 || glucosa<65){ //NORMOGLUCEMIA  
Verde=0;  
}  
else{  
Verde=110*pow(2.72, (-1*pow((glucosa-120),2)/1000)); //Campana de Gauss  
}  
  
//-----Rojo-----//  
  
if (glucosa<65){ //HIPOGLUCEMIA  
Rojo=255;}  
else{  
Rojo=800*pow(0.95,glucosa);} //Función exponencial  
}
```

ILUSTRACIÓN 49 ALGORITMO COLORES EN LEDS

### 5.3. Protocolo de Comunicación

En el sistema desarrollado la comunicación entre el robot tipo oruga y el software Matlab adquiere un papel fundamental. La comunicación debe ser rápida y con normas que la doten de robustez.

Se requiere que la información fluya de manera rápida en ambos sentidos ya que los movimientos del robot deben estar basados en el nivel de glucemia del paciente virtual en el PC y la señal de control calculada en el robot se debe recibir con prontitud en el PC para realizar los ajustes necesarios sobre el nivel de insulina.

También se ha de tener en cuenta que los valores deben ser lo más exactos posibles ya que de esta exactitud dependerá el éxito de la regulación entre otros.

Muchos de los datos enviados serán negativos y otros valores muy grandes o muy pequeños. Por todo esto se necesita de un sistema que sea capaz de soportar valores de estos rangos sin perder exactitud.

Se debe también llegar a un compromiso entre la “calidad” (en cuanto a información) de la comunicación y los retardos que se aplicarían en el sistema.

Por otro lado, se ha de tener en cuenta también la limitación de la red elegida para las comunicaciones. Tan solo se pueden enviar datos tipo entero que disponen en nuestro caso de una capacidad de dos bytes.

Teniendo en cuenta todos los requisitos de comunicación y las limitaciones de nuestra red se ha implementado un sistema de comunicación mediante tramas de información. Esto permite elegir la precisión y magnitud de los datos enviados, así como agrupar toda la información que se quiere enviar al destino en un único paquete.

Estas tramas disponen de un flag de inicio y uno de fin, con el objetivo de dotar de robustez a la comunicación. En la trama se dispone de 4 bytes de datos para poder representar los valores numéricos que se quieren enviar. Las tramas enviadas tienen el siguiente aspecto.



ILUSTRACIÓN 50 TRAMA DE DATOS

El flag de inicio de la trama sirve como indicador de inicio de trama y como discriminador de tipo de datos que se han enviado.

Los 4 bytes de los “datos de usuario” son los destinados a enviar el valor deseado. En la codificación se han tenido en cuenta los requerimientos de precisión y magnitud.

El sistema de codificación seleccionado, que cumple con los requisitos solicitados, es la representación en coma flotante. De esta manera se pueden representar números racionales muy grandes y pequeños de una manera eficiente y compacta.

No se ha seguido el estándar para coma flotante de IEEE (IEE745), debido en parte a las características de la red de comunicaciones y para facilitar la gestión de la información recibida y enviada. No obstante, se adopta la filosofía general de la representación.

Se utiliza el primer byte como signo del valor a enviar y como signo del exponente. El segundo byte es completo para el exponente (sin signo). El tercer y el cuarto byte son para el valor (mantisa).



ILUSTRACIÓN 51 DATOS DE USUARIO TRAMA

En ambos softwares (Arduino y Matlab) se han desarrollado funciones para “montar” y “desmontar” la trama antes y después del envío respectivamente. Las funciones implementadas son idénticas en ambas plataformas (teniendo en cuenta las particularidades de cada uno de los lenguajes de programación). En las Ilustraciones 51 y 52 se muestran dos de las funciones.

```
void desmontarfloat(float dato, int paquete[]){
int exponente=0;
float absdato=0;
paquete[1]=0;
if (dato!=0){
  if (dato<0){ // Dato positivo
    paquete[1]=128;
    absdato=dato*(-1);}
  else{ // Dato negativo
    paquete[1]=0;
    absdato=dato;}
  if (absdato<1){ // Signo de exponente
    paquete[1]=paquete[1]+1;}

  if (absdato<1){
    while (absdato<1){ // Exponente
      absdato=(absdato)*10;
      exponente=exponente + 1;}
    paquete[2]=exponente;}
  else{
    while (absdato>1){
      absdato=(absdato)/10;
      exponente=exponente + 1;}
    paquete[2]=exponente-1;
    absdato=absdato*10;
  }
  dato=dato*10;
  paquete[3]=int (absdato); // Mantisa
  paquete[4]=int ((absdato - paquete[3])*100);
  paquete[0] = 'a';
}
```

```
function [float] = montarfloat(recibido)

float = recibido(4)*100;
float = float + recibido(5);
float = float/1000;
if recibido(2)== 1 || recibido(2)== 129
  recibido(3)= recibido(3)*(-1);
end
float = float*10^recibido(3);
if recibido(2)>127
  float = float*(-1);
end
return
```

ILUSTRACIÓN 52 DESMONTARFLOAT ARDUINO

ILUSTRACIÓN 53 MONTARFLOAT EN MATLAB

# Capítulo 6

## 6. Conclusiones

---

Partiendo del objetivo propuesto en el capítulo 1.2, se ha implementado una plataforma didáctica que muestra la gestión de la Diabetes Mellitus tipo I de una manera sencilla y vistosa.

Se plantearon diferentes sub-objetivos al inicio del proyecto. A continuación, enumeramos los resultados obtenidos en base a estos propósitos iniciales:

*1. Sistema sencillo de manejar e intuitivo.*

La plataforma se presenta mediante una interfaz accesible que resulta además atractiva para los pacientes más jóvenes como se puede ver en las capturas del apartado 5.1.

*2. Representar en su estructura una primera abstracción hacia el páncreas artificial.*

Por otro lado, también se ha conseguido realizar una abstracción física hacia el nuevo páncreas artificial. La plataforma puede considerarse, de manera funcional, como un paciente real que recibe un tratamiento mediante páncreas artificial. El PC sería el propio paciente, el cual dispondría además de un monitor continuo de glucosa y el robot representaría la “bomba” que suministra la insulina al paciente, haciendo uso de un algoritmo de control y del valor medido por el monitor continuo de glucosa.

*3. Tratar de un instrumento interactivo en el que se puedan probar distintos escenarios.*

Aunque la plataforma se ha diseñado dirigida a los niños, ésta puede ser útil para pacientes de todas las edades, incluso para estudiantes. Según el nivel de comprensión del paciente, éste puede profundizar más o menos en la aplicación. Puede, por ejemplo, comparar el comportamiento del paciente virtual con diferentes configuraciones o parámetros del controlador PID y analizar todos los resultados en las gráficas mostradas.

En cuanto a la utilidad para estudiantes e investigadores, éstos pueden usar la plataforma como entorno para la simulación de diferentes métodos de control y parámetros en relación con el control de la Diabetes Mellitus. El entorno en el que se ha programado el controlador (Arduino), hace que la implementación de nuevos controladores sea una tarea accesible.

*4. La información debe ser mostrada de una manera atractiva y sencilla.*

Como se muestra en el apartado 5.1 la interfaz de usuario se ha diseñado de una manera atractiva y sencilla. Por otro lado, para la representación de la información, como por ejemplo el nivel de glucosa en sangre de un paciente, se ha tratado de que esta fuera atractiva y original introduciendo patrones de movimiento y visualización en el robot tipo oruga como se puede observar en el apartado 5.2.

*5. La plataforma debe exponer, para el aprendizaje por parte del paciente, tanto aspectos sobre la propia enfermedad como sobre su gestión.*

Como se puede ver en las capturas de la plataforma y en sus funciones descritas en el apartado 5.1.2 no solo se muestra la respuesta del paciente a un escenario de ingestas de alimentos e insulina, sino que el usuario debe interactuar con la aplicación detallando patrones que le familiarizan con la respuesta de un paciente al nivel de glucosa y con la gestión de la enfermedad.



# Capítulo 7

## 7. Trabajos futuros

---

La plataforma realizada cumple con los objetivos planteados inicialmente. No obstante, sobre ésta, se pueden realizar diferentes mejoras que añadirían información o atractivo. Algunas de ellas podrían ser las siguientes:

- **Nuevos hardware de representación de la información.** Usando la estructura implementada y realizando sobre ella algunos cambios, es posible añadir dispositivos de representación de información como pantallas o nuevos robots.
- **Aplicación-Calculadora Ingesta de Comida-Hidratos de Carbono.** Como complemento a la plataforma desarrollada se puede implementar una aplicación que de manera rápida y visual convierta las ingestas de comida de un paciente en su cantidad correspondiente de hidratos de carbono.
- **Comparación entre diferentes algoritmos de control.** Gracias al entorno en el que se ha desarrollado el algoritmo de control, es posible, modificando pocas líneas de código, sustituir el control PID por otros controles o incluso añadirlos a la plataforma.
- **Comparación de modelos matemáticos.** En la misma línea del punto anterior es factible también implementar diferentes modelos matemáticos de paciente con el fin de realizar una comparación o una mejora del proceso actual.
- **Evolución de la plataforma actual.** Como mejora que requeriría de más desarrollo, podríamos contemplar la introducción de todos los cálculos en un mismo dispositivo móvil (robot o similar). De esta manera, si el dispositivo dispone de una interfaz de usuario completa, se podría eliminar el PC con todas las facilidades que ello conlleva. Para realizar esta solución se requiere que el dispositivo esté dotado de un hardware con suficiente potencia para realizar todos los cálculos de ejecución del paciente virtual.

# Capítulo 8

## 8. Presupuesto

---

En esta sección se pretende hacer un balance de los costes asociados al diseño, desarrollo y programación de la plataforma educativa descrita en los apartados anteriores. Se procederá seccionando el presupuesto del proyecto en sus unidades funcionales, de modo que el usuario sea capaz de diferenciar sus distintos bloques de carga y costes asociados a cada uno de ellos. A continuación, se desglosa cada unidad de trabajo en distintas unidades de trabajo, y sus costes correspondientes.

### 8.1. Mano de obra

Tomando como referencia una estimación de salario de un ingeniero de automatización con poca experiencia, unos 25.000€, y tomando como 223 días laborables al año podemos asumir un coste aproximado de unos 15€/h de trabajo.

### 8.2. Materiales

A continuación, se pasa a detallar los medios materiales utilizados:

- Ordenador de sobremesa	800€
- Robot tipo oruga Makeblock	160€
- Slider-Potenciómetros x4	14,85€
- Sistema operativo Windows 7	135€
- Matlab R2015 (estudiante)	125€
- IDE Arduino	0€

### 8.3. Precios unitarios descompuestos por unidad de obra

#### - Unidad de obra 1:

PC de sobremesa con sistema operativo Windows 7 y robot tipo oruga. Instalación de software requerido para el proyecto. Matlab R2015 e IDE Arduino:

PC sobremesa	800€	800€
Robot tipo oruga	160€	160€
Sistema operativo Windows 7	135€	135€
Potenciómetros	4,95€	14,85€
Matlab R2015	125€	125€
IDE Arduino	0€	0€
2h Ingeniero de Automatización	15€	30€
0,02 medios auxiliares	1264,85€	25,29€
0,03 costes indirectos	1290,14€	38,7€
<b>TOTAL</b>		<b>1328,84€</b>

#### - Unidad de obra 2:

Documentación de la plataforma educativa descrita:

60h Ingeniero de Automatización	15€	900€
0,02 medios auxiliares	900€	18€
0,03 costes indirectos	918€	18,36€
<b>TOTAL</b>		<b>936,36€</b>

#### - Unidad de obra 3:

Diseño y desarrollo en plataforma Matlab:

100h Ingeniero de Automatización	15€	1500€
0,02 medios auxiliares	1500€	30€
0,03 costes indirectos	1530€	45,9€
<b>TOTAL</b>		<b>1575,9€</b>

- Unidad de obra 4:

Diseño y desarrollo en plataforma Arduino:

80h Ingeniero de Automatización	15€	1200€
0,02 medios auxiliares	1200€	24€
0,03 costes indirectos	1224€	36,72€
<b>TOTAL</b>		<b>936,36€</b>

#### 8.4. Presupuesto de ejecución por contrata

PC sobremesa + Software	1550€
Robot tipo oruga	200€
Documentación	1250€
Presupuesto ejecución material	3125€
0.13 Gastos generales	796€
0.06 Beneficio industrial	415€
Presupuesto Total	7336€
0.21 IVA	1540€
Presupuesto de ejecución por contrata	<b>8876€</b>

## Bibliografía y Webgrafía

- American Diabetes Association. (2017). *American Diabetes Association*. Obtenido de [www.diabetes.org](http://www.diabetes.org)
- Anaya Oliver, A. (2014). *Implementacion de un Integrador de modelos de Pacientes Virtuales Diabéticos Tipo 1*.
- Bondia, J. (2010). *El Pancreas artificial: control automático de infusión de insulina en Diabetes Mellitus Tipo 1*.
- Boulton, A. (2005). *Diabetic neuropathies: a statement by American Diabetes Association*. Diabetes Care.
- Galván, J. B. (2014). *Desarrollo de un Sistema de Decisión basado en Lógica Borrosa para el uso de Bombas de Insulina*.
- Hovorka, R. (2002). *Pantitioning fluucose distribution/transport, disposal, and endogenous production during iv gtt*.
- Jameson JL, d. G. (2015). *Type 1 diabetes mellitus*. Delli AJ, Lernmark.
- Laguna Sanz, A. (2014). *Uncertainty in postprandial model identification in type 1 diabetes*. Universidad Politécnica de Valencia. Departamento de ingeniería de sistemas y automática.
- Lego. (2017). Obtenido de Lego Mindstorms: <https://www.lego.com/es-es/mindstorms>
- Makeblock. (2017). *Makeblock*. Obtenido de <https://makeblock.es/>
- OMS. (2017). *OMS*. Obtenido de [http://www.who.int/diabetes/action\\_online/basics/es/index1.html](http://www.who.int/diabetes/action_online/basics/es/index1.html)
- Organización Mundial de la Salud. (2016). *Informe Mundial sobre la Diabetes*.
- Owens, C. (2006). *Run-to-run control of blood glucose concentrations for people with Type 1 diabetes mellitus*.
- Ruiz, D. (2016). *Estrategias de control adaptativo para el páncreas artificial domiciliario en diaberes tipo 1*.
- Sánchez Juan, C. (2010). *Diabetes. Repercusión clínica, laboral y jurídica de la enfermedad*.
- Standards of Medical Care in Diabetes*. (2016). Obtenido de [www.ncbi.nlm.nih.gov](http://www.ncbi.nlm.nih.gov)
- Thibodeau, G. A., & Patton, Kevin T. (2009). *Anatomía y Fisiología*.
- Wonder Workshop. (2017). *Make Wonder*. Obtenido de <https://www.makewonder.com/>

# ANEXO I Manual de Usuario

## 1. Introducción

---

El siguiente manual ha sido concebido para guiar al usuario en la utilización de la herramienta didáctica desarrollada. Debe servir tanto para realizar una primera puesta en marcha de las aplicaciones como para resolver cualquier duda que surja durante su ejecución.

El actual documento está dirigido tanto a enfermos de diabetes Mellitus tipo 1 que usen directamente la aplicación para su formación como para educadores de cualquier tipo que quieran utilizar la plataforma en sus planes formativos.

La herramienta educativa diseñada consta esencialmente de un software en un PC, que realizará las funciones de interfaz de usuario y cálculos varios, y el robot tipo oruga que representa la información obtenida de los cálculos realizados en el PC.

En el PC se simula un paciente virtual y sus reacciones a patrones de ingestas de hidratos de carbono e insulina. Las simulaciones están divididas en dos grupos: sobre pacientes que reciben tratamiento contra la diabetes mediante una bomba de insulina tradicional y pacientes que reciben tratamiento con los nuevos páncreas artificiales (monitor continuo de glucosa y regulación automática del nivel de glucosa en sangre).

El robot reacciona en tiempo real al nivel de glucosa en sangre del paciente, ejecutando diferentes movimientos y visualizaciones para cada caso.

Las comunicaciones entre el robot y el PC se realizan mediante una conexión bluetooth que se mantiene activa durante la simulación.

Para facilitar el aprendizaje y aumentar su atractivo se han diseñado una serie de juegos con diferentes patrones de ingestas y movimientos del robot.

## 2. Conocimientos Previos

---

No son necesarios conocimientos previos extensos en ningún campo. Sin embargo, es aconsejable que el formador o el enfermo-usuario tengan conocimientos básicos en las siguientes materias:

1. Conocimientos básicos sobre la diabetes: sintomatología, tratamientos, aspectos generales...
2. Conocimientos informáticos básicos: el usuario de la herramienta deberá interactuar con ésta mediante un PC por lo que serán necesarias ciertas aptitudes muy básicas en este campo.

Para una utilización más profunda de la herramienta (análisis de resultados, pruebas con controladores y parámetros...) sí que serán necesarios conocimientos más extensos en materias como regulación automática, niveles y umbrales de glucosa en sangre o programación en Arduino.

## 3. Especificaciones Técnicas

---

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

### 3.1. Requisitos Hardware

Estas especificaciones son necesarias para instalar los programas utilizados en la herramienta, tanto la parte del PC como el robot tipo oruga.

1. PC con las siguientes características mínimas:
  - 3-4 GB de memoria en disco libre para una instalación típica de Matlab
  - Un mínimo de 2 GB de memoria RAM. 4 GB recomendables.
  - Bluetooth.
2. Robot tipo oruga Ranger + potenciómetros tipo "slider" (Makeblock, 2017).

### 3.2. Requisitos Software

Como se ha visto con anterioridad parte de la plataforma se ejecuta en el entorno de cálculo matemático Matlab, por lo que es necesaria su instalación. El diseño se ha realizado para la versión Matlab R2015b.

## 4. Manual

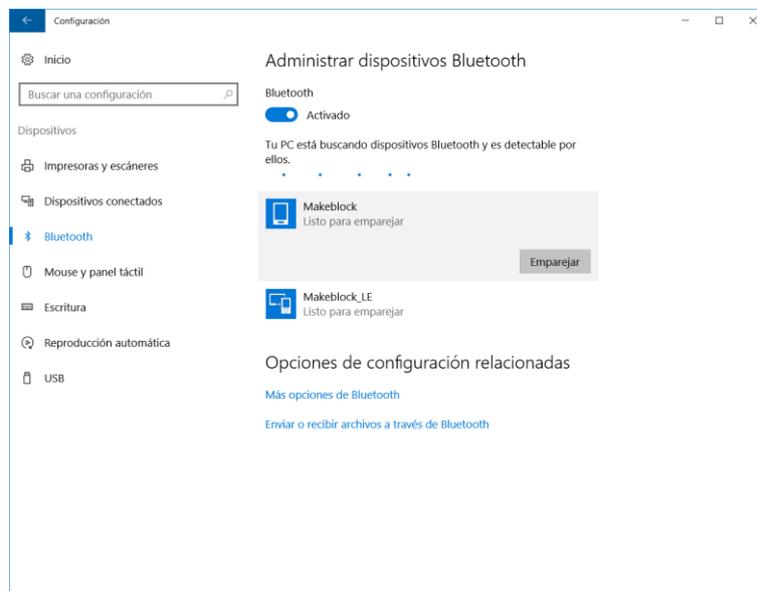
---

### 4.1. Primer uso de la Plataforma

Partiendo de la situación en la que se dispone de un PC con los requisitos necesarios y el software instalado y por otro lado el robot tipo oruga programado, se puede iniciar la ejecución de la plataforma educativa.

En primer lugar, se debe realizar la conexión entre los dos dispositivos hardware de la plataforma (esta configuración tan solo será requerida en el primer uso de la plataforma).

Para ello, tomando como ejemplo una configuración con Windows 10, se debe acceder a pantalla de administración de dispositivos bluetooth y emparejar ambos equipos: robot y PC.



**ILUSTRACIÓN 54 ADMINISTRACIÓN DE DISPOSITIVOS BLUETOOTH**

Una vez emparejados los dispositivos, se puede proceder a iniciar el programa Matlab R2015b o superior para ejecutar la plataforma.

En el primer uso se debe seleccionar la ubicación de la carpeta donde residen los archivos de la plataforma.

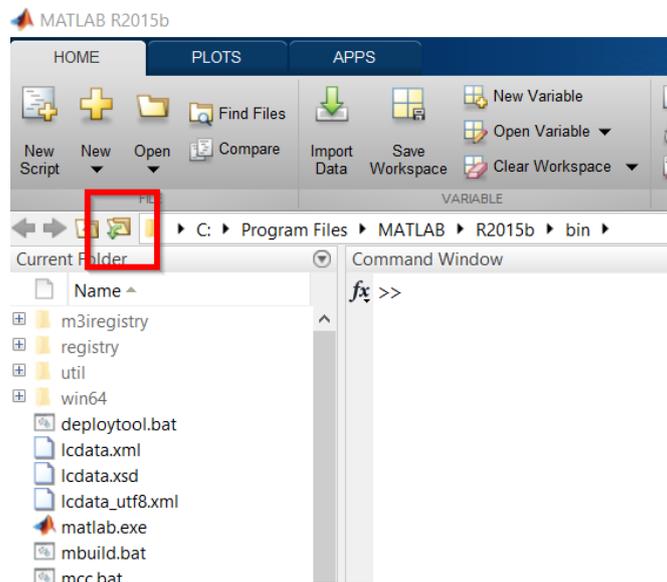


ILUSTRACIÓN 55 ABRIR ARCHIVOS

Una vez abierta la carpeta, se debe localizar el archivo “principal.m”. Este da acceso a la plataforma.

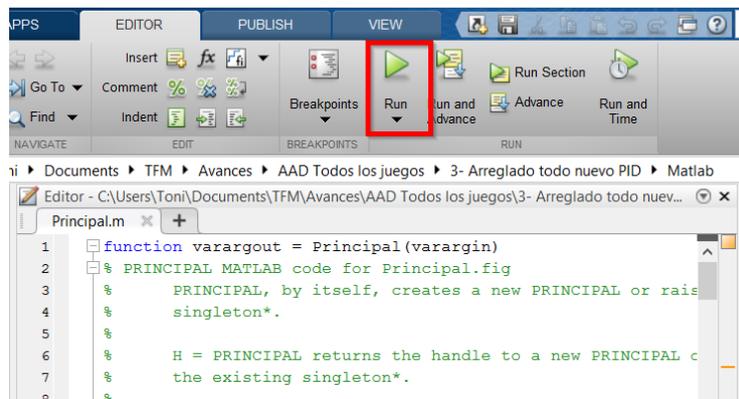


ILUSTRACIÓN 56 EJECUCIÓN DE PANTALLA PRINCIPAL

## 4.2. Uso de la Plataforma

Al ejecutar el archivo “Principal.m” se nos muestra la pantalla de inicio de la plataforma. El primer paso es la conexión mediante bluetooth al robot tipo oruga; este debe estar, evidentemente, alimentado mediante baterías y debe haber sido pulsado el botón de inicio tal como se ve en la siguiente ilustración.

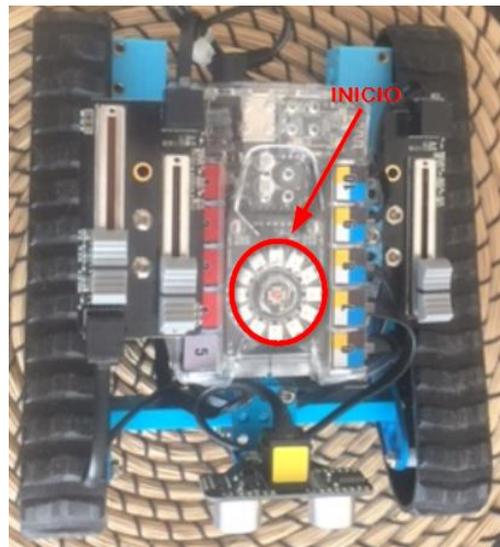


ILUSTRACIÓN 57 BOTÓN INICIO ROBOT

### Pantalla principal



ILUSTRACIÓN 58 PANTALLA PRINCIPAL

Para establecer la conexión bluetooth se debe pulsar sobre el botón “Conexión” y esperar a que en la parte superior derecha de la ventana se muestre el indicador “Conectado”.

Desde esta pantalla se accede a los dos modos de funcionamiento de la plataforma: “Paciente con Bomba de Insulina” y “Paciente con Páncreas Artificial” (pulsando en los correspondientes botones).

## Pantalla Paciente con Bomba de Insulina



ILUSTRACIÓN 59 PACIENTE CON BOMBA DE INSULINA

Desde la pantalla actual se accede a los juegos diseñados, siempre para el caso de un paciente tratado con bomba de insulina.

## Pantalla Siguelinea con Bomba de Insulina



ILUSTRACIÓN 60 SIGUELINEA CON BOMBA DE INSULINA

En este primer juego se pide al usuario que introduzca un patrón de ingesta de hidratos de carbono y otro de bolus de insulina para ejecutar una simulación de paciente virtual y mostrar los resultados sobre el robot tipo oruga y la pantalla del PC.

Para seleccionar los patrones de ingesta se puede elegir entre seis diferentes configuraciones: cinco comidas en un día, cinco comidas con disminución de dosis de insulina, cinco comidas con aumento de dosis de insulina, tres comidas diarias durante cuatro días, tres comidas diarias durante cuatro días con consumo irregular e introducción manual de patrones.

ILUSTRACIÓN 61 SELECCIÓN DE PATRONES

Tal como se ve en la ilustración anterior se selecciona el patrón deseado y se pulsa sobre efectuar cambios para guardar los datos correspondientes para la simulación.

En el último caso, en “Introducir Valores”, se debe introducir cada ingesta individualmente.

ILUSTRACIÓN 62 INTRODUCCIÓN MANUAL DE VALORES

Una vez seleccionados los patrones de ingestas e insulina se pueden modificar también algunos parámetros de la simulación como vemos en la siguiente ilustración.

ILUSTRACIÓN 63 PARÁMETROS DE LA SIMULACIÓN

En la casilla “Días de simulación” se puede modificar el periodo de simulación sobre el paciente virtual.

Por otro lado, en el “Intervalo de simulación” se ejerce control sobre el intervalo de tiempo transcurrido entre una iteración de simulación de paciente y la siguiente (el intervalo está definido en minutos). Como era de esperar, un intervalo de simulación más corto ofrecerá una regulación más ajustada para los casos con páncreas artificial.

El último paso para iniciar el juego es pulsar sobre el botón “Iniciar Siguelinea”.

El robot deberá estar situado sobre un circuito siguelinea (el diseño ha sido realizado para un circuito con fondo blanco y trazada en negro).

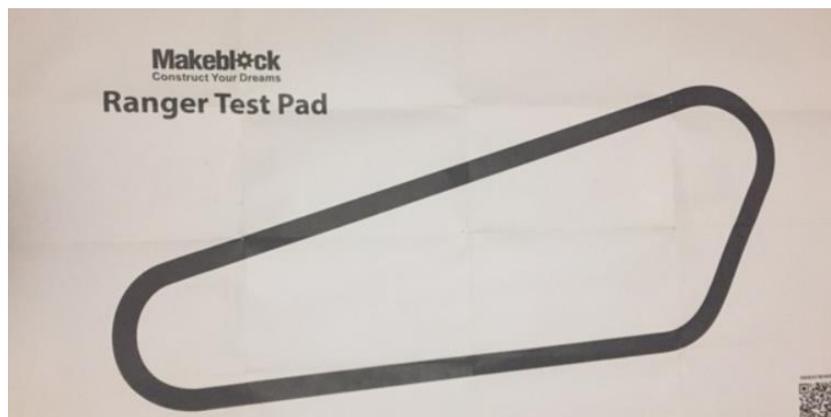


ILUSTRACIÓN 64 CIRCUITO DE EJEMPLO

El robot dibujará el circuito durante la simulación. Los movimientos como se ha visto estarán influenciados por el nivel de glucosa en sangre del paciente virtual.

## Pantalla Movimiento Libre

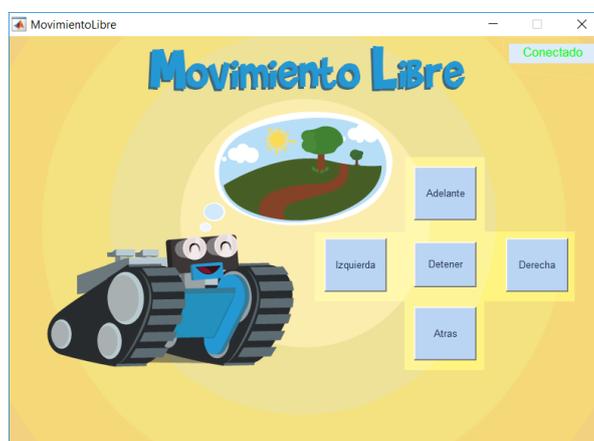


ILUSTRACIÓN 65 PANTALLA MOVIMIENTO LIBRE

El juego movimiento libre permite controlar los movimientos del robot directamente desde la interfaz en el PC. El robot se comporta como un vehículo teledirigido.

Para acceder a la pantalla la conexión bluetooth debe estar establecida. Ya en la pantalla de movimiento libre los leds del robot lucirán en verde, esto indica que el robot está preparado. Para iniciar la marcha o realizar giros solamente hay que clicar sobre los botones correspondientes.

Después de detener el robot con el botón “Detener”, para salir del juego, tan solo es necesario cerrar la ventana de movimiento libre.

Al cerrar la ventana los leds dejarán de lucir, lo que indicará que el robot está listo para ejecutar otro juego.

## Pantalla Evita objetos con Bomba de Insulina



ILUSTRACIÓN 66 PANTALLA EVITAOBJETOS

El juego evitaobjetos consiste en la simulación del paciente virtual por un lado y movimiento constante evitando objetos por parte del robot.

Tal como sucede en el juego Siguelinea la simulación se realiza en el PC y es enviado el nivel de glucosa en sangre al robot. Éste reacciona de manera diferente según el nivel recibido, siendo peor la respuesta (evitando objetos) cuanto más alejado del nivel de glucemia recomendado está el nivel recibido.

La introducción de los patrones de ingesta, insulina y parámetros de simulación es similar al del juego siguelinea.

Para iniciar el juego, después de la introducción de patrones y parámetros, se debe pulsar el botón “Iniciar Evita-Objetos”. Para un juego más vistoso colocar el robot en el centro de una habitación con diversos objetos.

El movimiento durará mientras dure la simulación en el PC.

## Pantalla Siguelinea Alimentación Manual con Bomba de Insulina



ILUSTRACIÓN 67 PANTALLA SIGUELINEA CON ALIMENTACIÓN MANUAL

El siguiente juego es una variación sobre el juego “Siguelinea”. La diferencia radica en la adquisición de datos por parte de la plataforma.

En este juego se hará uso de los “slider” (potenciómetros) instalados sobre el robot para introducir los patrones de ingestas de hidratos y de bolus de insulina. En la siguiente ilustración se muestra los slider que se utilizan para tal propósito.

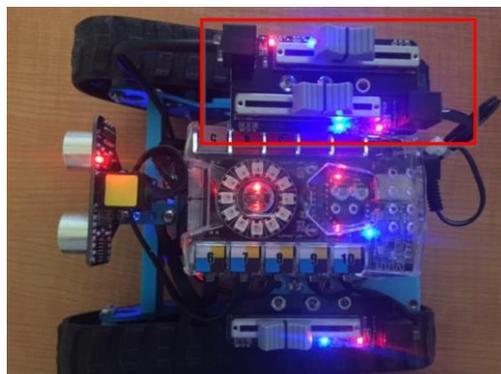


ILUSTRACIÓN 68 SLIDERS PARA PATRONES DE INGESTAS E INSULINAS

Para adquirir los datos se debe, en primer lugar, seleccionar la comida que se quiere completar: Desayuno, Comida o Cena.

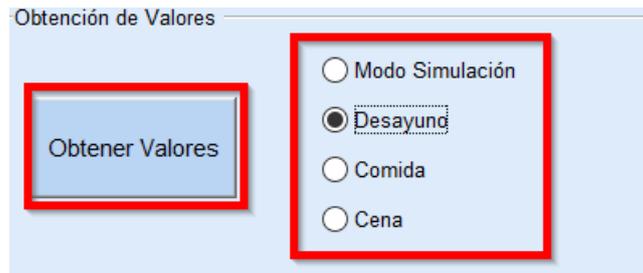


ILUSTRACIÓN 69 SELECCIÓN DE COMIDA

Una vez seleccionada la comida de la que se va a adquirir los datos tan solo es necesario pulsar el botón “Obtener Valores” para que comience la adquisición continua de datos. Los datos seleccionados se muestran en “tiempo real” en la interfaz mientras se mueven los sliders.

Se repetirá este paso para las tres ingestas propuestas.

El robot debe también situarse sobre un circuito siguelinea antes de empezar con la simulación. Para empezar con ésta se debe seleccionar el “Modo Simulación” y pulsar sobre “Iniciar simulación”.

Al igual que en el modo “Siguelinea”, el robot dibujará el circuito durante la simulación siendo influenciados sus movimientos por el nivel de glucosa en sangre simulado en el paciente virtual.

## Pantalla Paciente con Páncreas Artificial



ILUSTRACIÓN 70 PACIENTE CON PÁNCREAS ARTIFICIAL

El segundo modo de funcionamiento de los juegos es cuando el supuesto paciente es tratado mediante un páncreas. En la ilustración anterior se muestran las opciones de juegos en este modo. A esta pantalla se accede desde la pantalla principal

## Pantalla Siguelinea con Páncreas Artificial

ILUSTRACIÓN 71 SIGUELINEA CON PÁNCREAS ARTIFICIAL

El juego “Siguelinea con páncreas artificial” es similar al juego “siguelinea con bomba de insulina” explicado anteriormente. La diferencia radica en el control que se ejerce sobre el suministro de insulina al paciente. Para el caso del páncreas artificial, hay implementado un controlador PID que, mediante el valor de glucosa en sangre en cada momento y un algoritmo de control, sugiere suministros de insulina (adicionales a los bolus de insulina relacionados con las ingestas de hidratos que configura el paciente).

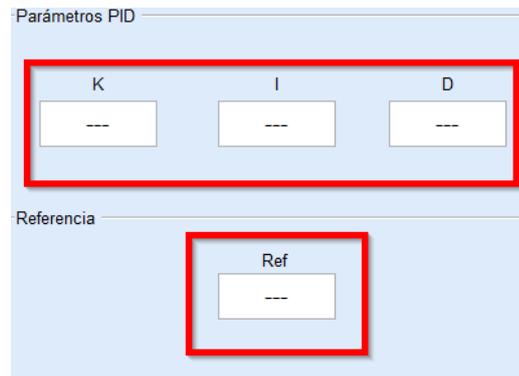
La introducción de los datos de los patrones de ingesta y de los parámetros de la simulación son idénticos a los del juego “Siguelinea con bomba de insulina” (ver ilustraciones 58, 59 y 60).

Como añadido respecto al juego con bomba de insulina, se deben introducir también algunos parámetros relacionados con el controlador PID. En primer lugar, se introducen las características del controlador. Las opciones las podemos ver en la siguiente ilustración.

ILUSTRACIÓN 72 CARACTERÍSTICAS DEL CONTROLADOR

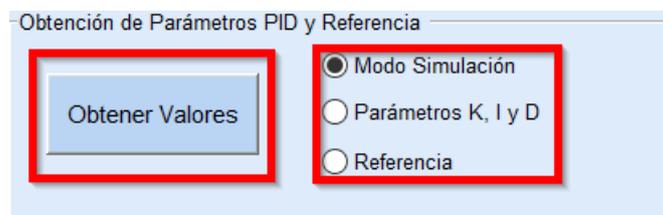
El primer parámetro se trata del filtro del controlador que limita la señal respecto a la basal de equilibrio. El caso del control de bolo se utiliza para variar la cantidad de insulina suministrada en los bolos de forma rápida. Si se selecciona la opción de 100% o sin alterar las cantidades de insulina suministradas al paciente serán las sugeridas por el controlador.

También será necesario introducir los parámetros propios del controlador PID y la referencia a la que queremos que se ajuste el nivel de glucosa en sangre. En la siguiente ilustración se muestra el interfaz para introducir estos datos.



**ILUSTRACIÓN 73 PARÁMETROS Y REFERENCIA PARA CONTROLADOR**

Estos datos son obtenidos mediante los slider montados sobre el robot. Para iniciar la adquisición de datos se selecciona la opción “Parámetros K,I y D” o “Referencia” y a continuación se pulsa “Obtener Valores”. En ese momento se ejecuta de forma cíclica una función que captura los datos de los slider del robot y los guarda para su utilización. Todos los datos se muestran en “tiempo real” en el PC mientras dura la adquisición.



**ILUSTRACIÓN 74 MODO SIMULACIÓN O MODO ADQUISICIÓN**

Para iniciar el juego, al igual que para el caso de siguelinea con bomba de insulina es necesario posicionar al robot sobre un circuito, seleccionar el “Modo Simulación” y pulsar sobre “Iniciar Simulación”.

## Pantalla Evita objetos con Páncreas Artificial

EvitaobjetosB

### Evita-objetos con Páncreas Artificial

Conectado

Valores carbohidratos y bolus

Ejemplo 1: Cinco comidas en un día  
 Ejemplo 2: Cinco comidas en un día disminuyendo dosis insulina  
 Ejemplo 3: Cinco comidas en un día aumentando dosis insulina  
 Ejemplo 4: Cuatro días con tres comidas diarias (consumo regular)  
 Ejemplo 5: Cuatro días con tres comidas diarias (consumo irregular)  
 Introducir valores:

CHO

Día (dd) Horas (hh) Minutos (mm) Gramos (ggg) >> X

BOLUS

Día (dd) Horas (hh) Minutos (mm) Dosis (u.u) >> X

Parámetros de simulación

Días de simulación: 1

Intervalo de simulación: 20

Características del controlador

Filtro PID: % de desviación sobre la basal de equilibrio

Sin límites  
 Establecer límite: 0.5 / 1.5 (tanto por uno)

Control Bolo

Déficit/Exceso bolo: 100 %  
 Sin alterar

Parámetros PID

K I D

Referencia

Ref

Obtención de Parámetros PID y Referencia

Modo Simulación  
 Parámetros K, I y D  
 Referencia

ILUSTRACIÓN 75 EVITA OBJETOS CON PÁNCREAS ARTIFICIAL

El juego evitaobjetos con páncreas artificial consiste, al igual que en el caso con bomba de insulina, en la simulación del paciente virtual por un lado y movimiento constante evitando objetos por parte del robot. La diferencia entre los dos modos es que para este caso el paciente simulado es tratado mediante un páncreas artificial (regulación automática del nivel de glucosa).

La simulación se realiza en el PC y es enviado el nivel de glucosa en sangre al robot. Éste realiza los cálculos de control necesarios y envía la sugerencia de insulina al simulador. Como ocurre en los casos con bomba de insulina, el robot reacciona de manera diferente según el nivel recibido, siendo peor la respuesta (evitando objetos) cuanto más alejado del nivel de glucemia recomendado está el nivel recibido.

La introducción de los datos por parte del usuario es similar al caso “Siguelinea con Páncreas Artificial”.

Para iniciar el juego, después de la introducción de patrones y parámetros, se debe pulsar el botón “Iniciar Evita-Objetos”. El movimiento durará mientras dure la simulación en el PC.

## Pantalla Siguelinea Alimentación Manual con Páncreas Artificial



ILUSTRACIÓN 76 SIGUELINEA MANUAL CON PÁNCREAS ARTIFICIAL

El actual juego es similar al “Siguelinea alimentación manual con bomba de insulina” en cuanto a movimientos del robot. La diferencia existe en la regulación del nivel de insulina del paciente virtual mediante un control PID.

En este juego se hará uso también de los slider instalados sobre el robot para introducir los patrones de ingestas de hidratos, los bolus de insulina relacionados con las comidas y los parámetros del controlador PID (ver Ilustración 62)

Para adquirir los datos se debe: seleccionar la comida que se quiere completar (Desayuno, Comida o Cena) o parámetros PID o referencia.

Una vez seleccionada la comida de la que se va adquirir los datos tan solo es necesario pulsar el botón “Obtener Valores” para que comience la adquisición continua de datos. Los datos seleccionados se muestran en “tiempo real” en la interfaz mientras se mueven los sliders.

El robot debe también situarse sobre un circuito siguelinea antes de empezar con la simulación. Para empezar con ésta se debe seleccionar el “Modo Simulación” y pulsar sobre “Iniciar simulación”.

Al igual que en el modo “Siguelinea”, el robot dibujará el circuito durante la simulación siendo influenciados sus movimientos por el nivel de glucosa en sangre simulado en el paciente virtual.

# ANEXO II Manual del Programador

## 1. Introducción

---

El actual manual es un documento de carácter formativo orientado a programadores. La finalidad de éste es facilitar la tarea de la comprensión del código implementado, para su modificación, mantenimiento o ampliación.

Por las características de la plataforma el manual es útil también en el caso de querer modificar el modelo de paciente, el algoritmo de control de glucosa o cualquier otro aspecto.

En el manual se presenta la estructura general de la plataforma respecto a su programación. A su vez, se muestran partes de código importantes en los distintos lenguajes de programación.

## 2. Estructura General del Código

---

El código implementado está dividido en dos grandes bloques: el software implementado en el PC (Matlab) y el software implementado en el robot tipo oruga (Arduino). Estos programas se comunican entre ellos para realizar todas las funciones descritas con anterioridad.

A continuación, se realizará una breve descripción de los aspectos más importantes del software de la plataforma.

### Software Matlab

En primer lugar, presentamos el software implementado en el PC. Este software ha sido desarrollado en Matlab y sus funciones principales son las de servir de interfaz de usuario y ejecutar un modelo de paciente virtual.

El código de Matlab lo podemos presentar siguiendo las pantallas de usuario desarrolladas como interfaz de la plataforma.

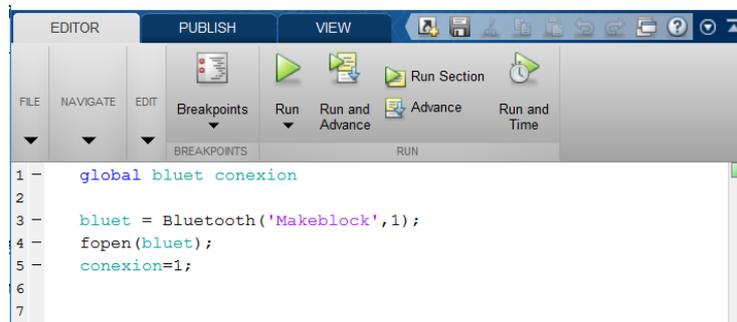
## Principal.m

“Principal” es la pantalla de acceso a la plataforma por parte del usuario. Desde esta se accede a todas las opciones y juegos.

El software asociado a la pantalla por un lado discrimina el tipo de simulación que realizará (bomba de insulina o páncreas artificial) mediante botones “típicos” y por otro lado establece la conexión con el robot tipo oruga (condición indispensable para el desarrollo de los juegos).

Para la conexión con el robot esta primera pantalla hace uso de las funciones “Conexión.m” y “Desconexión.m”.

En ellas se hace uso de las instrucciones necesarias en Matlab para establecer la conexión bluetooth. Otro aspecto importante es la definición de la variable “bluet” como variable global para poder realizar acciones de comunicación en otras funciones como veremos más adelante.



```
1 - global bluet conexion
2
3 - bluet = Bluetooth('Makeblock',1);
4 - fopen(bluet);
5 - conexion=1;
6
7
```

ILUSTRACIÓN 77 FUNCIÓN CONEXIÓN.M

## Paciente.m y PacienteB.m

Estas dos funciones a las que se accede desde la pantalla principal discriminan el tipo de juego a realizar en la plataforma. Desde el punto de vista de la programación no tienen mayor interés que el de los botones que dan acceso a las pantallas de configuración de los diferentes juegos.

## Siguelinea.m

Siguelinea es el primer juego que presentaremos. Como se ha visto anteriormente se realiza una simulación con un paciente virtual de los efectos de la ingesta de hidratos de carbono y de insulina y se muestran los resultados por pantalla y con distintos patrones de movimiento del robot oruga mientras sigue una línea en un circuito.

En la pantalla de configuración del juego (al igual que en todas las demás) en la parte superior derecha se muestra un rótulo con la indicación de si la conexión está establecida o no. En la ilustración siguiente se muestra el código correspondiente.

```
global choman bolusman salida conexion
choman=[];
bolusman=[];
salida=1;

if conexion==1
    set(handles.text,'ForegroundColor',[0 1 0]);
    set(handles.text,'String','Conectado');
else
    set(handles.text,'ForegroundColor',[1 0 0]);
    set(handles.text,'String','No Conectado');
end
```

ILUSTRACIÓN 78 RÓTULO CONEXIÓN

En la parte superior de la pantalla, en la sección “Valores carbohidratos y bolus”, como se comentó anteriormente, se introducen los valores de los patrones de ingesta de hidratos y de insulina.

El código necesario para esta función está implementado en la función “btcambios”, la cual se ejecuta al pulsar sobre el botón “Efectuar cambios”.

En esta función se graban patrones de ingestas según las opciones seleccionadas. Estos patrones son almacenados en las matrices: **chofin** para las ingestas de hidratos de carbono y **bolusfin** para las de insulina.

En la siguiente ilustración se muestra un ejemplo. Ha sido seleccionado el primer caso “Cinco comidas al día”.

```
if estado1==1
    chofin=[480.000000 20.000000; 600.000000 10.000000; 870.000000 60.000000
    bolusfin=[480.000000 0.800000 1.000000; 600.000000 0.400000 1.000000; 8
        vars = [d e d1 e h1 p m1 h e num2str(chofin(1,2)) g;
            d e d1 e h2 p m2 h e num2str(chofin(2,2)) g;
            d e d1 e h3 p m3 h e num2str(chofin(3,2)) g;
            d e d1 e h4 p m4 h e num2str(chofin(4,2)) g;
            d e d1 e h5 p m5 h e num2str(chofin(5,2)) g];
        set(handles.listboxcho,'string',vars)
        vars = [d e d1 e h1 p m1 h e num2str(bolusfin(1,2)) U;
            d e d1 e h2 p m2 h e num2str(bolusfin(2,2)) U;
            d e d1 e h3 p m3 h e num2str(bolusfin(3,2)) U;
            d e d1 e h4 p m4 h e num2str(bolusfin(4,2)) U;
            d e d1 e h5 p m5 h e num2str(bolusfin(5,2)) U];
        set(handles.listboxbolo,'string',vars)
elseif estado2==1
```

ILUSTRACIÓN 79 CINCO COMIDAS AL DÍA

En el ejemplo de la ilustración (cinco comidas al día) se introducen unos valores predefinidos a las variables “chofin” y “bolusfin”. La primera columna de las matrices corresponde al instante de tiempo en el que se realiza la acción: 480.000000, por ejemplo, corresponde a las 8 de la mañana, instante en minutos desde la hora 0:00 del día actual.

La segunda columna corresponde a la cantidad de hidratos de carbono ingeridos en el caso de “chofin” y a las unidades de insulina suministradas para la matriz “bolusfin”.

La matriz “vars” se utiliza para representar los valores en los dos cajetines del lateral.

Para el caso en el que se selecciona la última opción: “Introducir valores” se hace uso de dos funciones que recogen los datos introducidos por el usuario: “btenviar1” (para hidratos) y “btenviar2” (para insulinas).

```
function btenviar1_Callback(hObject, eventdata, handles)
global choman
d='Día';
h='h';
u='U/min';
e=' ';
p=': ';
g='g';
d_cho=(get(handles.d1, 'String'));
h_cho=(get(handles.h1, 'String'));
m_cho=(get(handles.m2, 'String'));
g_cho=(get(handles.gra1, 'String'));
dia_cho=str2double(get(handles.d1, 'String'));
hora_cho=str2double(get(handles.h1, 'String'));
minuto_cho=str2double(get(handles.m2, 'String'));
gramo_cho=str2double(get(handles.gra1, 'String'));

tiempo=(dia_cho-1)*24*60+hora_cho*60+minuto_cho;
ej6cho=[choman; tiempo gramo_cho];
choman=ej6cho;

prev_list = get(handles.listboxcho, 'string');
vars = [d e d_cho e h_cho p m_cho h e g_cho g];
new_list = [prev_list; vars];
set(handles.listboxcho, 'string', new_list)
```

#### ILUSTRACIÓN 80 SELECCIÓN MANUAL

Como se muestra en la ilustración anterior los datos son obtenidos de las casillas y almacenados en este caso en la matriz “choman”. Estos datos son almacenados ingesta a ingesta, aumentando en cada iteración el tamaño de la matriz.

Se han definido también las funciones: “bteliminar1” que elimina la última entrada de hidratos, “btenviar2” para la adquisición del patrón de insulinas y “bteliminar2” para la eliminación de la última entrada de insulinas.

Por último, con todos los patrones guardados se accede a la simulación mediante el botón “Iniciar Siguelinea”. Este botón activa la función “btejecutar”.

En esta, como vemos en su código (Ilustración siguiente) se adquieren los valores de los días e intervalo de simulación y se asignan los patrones para el caso de introducción de valores manual. Como acceso al simulador se pasan diferentes parámetros a la función “Simulador.m”, entre ellos: el tipo de actividad que realizará el robot durante la simulación, días e intervalo de simulación y las matrices de patrones de ingestas.

```
function btejecutar_Callback(hObject, eventdata, handles)
global chofin bolusfin choman bolusman bluet

d=str2double( get(handles.edias,'string'));
I=str2double( get(handles.eintervalo,'string'));

estado6 = get(handles.rej6,'Value');
if estado6==1
    chofin=choman;
    bolusfin=[bolusman [ones(5,1)]];
end

actividad = 's';           % Actividad siguelinea
Simulador(actividad,d,I,chofin,bolusfin)
```

ILUSTRACIÓN 81 EJECUCIÓN DEL SIMULADOR

## Simulador.m

La función simulador.m desempeña dos trabajos fundamentales (entre otros): realiza los cálculos matemáticos para la ejecución del paciente virtual e implementa las diferentes transacciones de datos entre el PC y el robot tipo oruga.

A continuación, se presentan las partes más importantes de esta función.

En primer lugar, en la función, se envían al robot los datos necesarios para que este conozca qué actividad ha sido seleccionada.

```
%-----Envío de byte actividad para definir actividad en Robot-----
fwrite(bluet,actividad,'uint8');
    pause(0.05);
s=fscanf(bluet,'%d');
    pause(0.05);
```

ILUSTRACIÓN 82 ENVÍO ACTIVIDAD

Como se ha comentado uno de los principales cometidos de esta función es el de ejecutar un modelo de paciente. Este modelo de paciente se ha implementado en base al modelo Hovorka04 como se ha comentado en la memoria.

Como primera acción en la implementación del modelo matemático se definen los parámetros del modelo Hovorka.

A continuación, se define un bucle “for” con tantas iteraciones como se haya definido en la configuración del juego (Minutos totales de simulación / Intervalo de simulación).

```

for i=0:I:(d*24*60-I)
    [t2,X2] = ode45(@ModeloPaciente,i:n:i+I,X4,options,...
                  cho,bolus,basal,
                  ag,tmaxG,ka1,ka2

    X4=X2(end,:);

    t3=[t3;t2];
    X3=[X3;X2];

```

**ILUSTRACIÓN 83 BUCLE PRINCIPAL MODELO DE PACIENTE**

En este bucle principal se resuelven las ecuaciones diferenciales necesarias mediante la instrucción de Matlab “ode45”.

Las ecuaciones son definidas en la función modelo paciente como se puede observar en la siguiente ilustración.

```

dg1_dt=ag*u_cho-g1/tmaxG;
dg2_dt=g1/tmaxG-g2/tmaxG;
ds1_dt=u_ins-s1/tmaxI;
ds2_dt=s1/tmaxI-s2/tmaxI;
dI_dt=ui/vi-ke*I;
dx1_dt=-ka1*x1+ka1*sit*I;
dx2_dt=-ka2*x2+ka2*sid*I;
dx3_dt=-ka3*x3+ka3*sie*I;
dQ1_dt=-(f02/(vg*G)+x1)*Q1+k12*Q2-Fr+ug+egp;
dQ2_dt=x1*Q1-(k12+x2)*Q2;

[dx_dt]=[dg1_dt;dg2_dt;ds1_dt;ds2_dt;dI_dt;dx1_dt;dx2_dt;dx3_dt;dQ1_

```

**ILUSTRACIÓN 84 ECUACIONES DEL MODELO MATEMÁTICO HOVORKA**

En la simulación es necesario también un estado inicial del paciente. Este estado es proporcionado por la función “ValoresIniciales” que tomando como entrada los parámetros del modelo Hovorka (definidos con anterioridad) devuelve un estado inicial del paciente (ver Ilustración).

```
function [x0]=ValoresIniciales(ag,tmaxG,ka1,ka2,ka3,sit,sid
% Simulation CSII therapy for type 1 diabetes

% Model: Hovorka04

%set an error
options=odeset('RelTol',1e-6,'Stats','off');

% Input
% Carbohydrates intake (min, grams). t=0 -> 0:00 a.m.

cho=[]; % no ingestas

% Bolus insulin (min, dosis). t=0 -> 0:00 a.m. day 1
i2c=1/25;
bolus=[8*60 0 1;... % day 1
];
% bolus=[]; % no bolus

% Basal insulin (min, dosis). t=0 -> 0:00 a.m. day 1
basal=[0 0.4/60;... % day 1
```

ILUSTRACIÓN 85 PARTE DEL CÓDIGO DE INICIALIZACIÓN DEL PACIENTE VIRTUAL

En cada iteración del bucle principal se realizan diversas funciones independientes al modelo de paciente:

1. **Evaluación glucémica del paciente.** Se calcula si en la iteración actual el paciente sufre de hipoglucemia, hiperglucemia o normoglucemia. Al finalizar la simulación se muestra por pantalla el tiempo que ha pasado el paciente en cada estado.
2. **Envío del nivel glucosa.** También para cada iteración la función “simulador.m” envía el dato del nivel de glucosa en sangre al robot tipo oruga para que este actúe en consecuencia modificando si fuese necesario sus patrones de movimiento. Como se explicó en la memoria del proyecto se ha definido un protocolo de comunicaciones donde los datos antes de ser enviados deben de ser tratados. Para este caso en el que queremos enviar un número real se hace uso de la función “desmontarfloat”. Esta función como podemos ver en el código siguiente trocea el valor a enviar en diferentes bytes.

```

function [envio] = desmontarfloat (dato, tipo)

exponente=0;
if dato ~= 0
if dato<0
envio(2)=128;
else
envio(2)=0;
end

if abs(dato)<1
envio(2)=envio(2)+1;
end

if abs(dato)<1
while abs(dato)<1...
envio(3)=exponente;

else
while abs(dato)>1...
envio(3)=exponente-1;
dato=dato*10;
end

dato=dato*10;
envio(4)=floor(abs(dato));
envio(4)=uint8(envio(4));

```

ILUSTRACIÓN 86 CODIFICACIÓN DE VALOR A ENVIAR

3. **Representación de la información.** Por último, se van representando los resultados de nivel de glucosa en sangre en una gráfica en el PC mostrando a su vez el máximo y mínimo alcanzados durante la simulación.

Continuando con los juegos implementados para el paciente con tratamiento de bomba de insulina, a continuación, mostraremos el “Movimiento libre”.

## MovimientoLibre.m

A esta función se accede solo si la comunicación está establecida.

Utilizando la instrucción de apertura “MovimientoLibre\_OpeningFcn” se envía al robot el tipo de actividad que se va a desarrollar.

```

function MovimientoLibre_OpeningFcn(hObject,
% This function has no output args, see Outp
% hObject handle to figure
% eventdata reserved - to be defined in a f
% handles structure with handles and user
% varargin command line arguments to Movim

global bluet

%-----Envio de byte Actividad para comuni
pause(0.2); %Pause
%comunic
fwrite(bluet,'m','uint8'); %Activ
pause(0.05);
s=fscanf(bluet,'%d');
pause(0.05);
% Choose default command line output for Mov
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

```

ILUSTRACIÓN 87 ENVÍO ACTIVIDAD MOVIMIENTO LIBRE

El procedimiento de envío de movimientos al robot es mediante los botones de la pantalla, tal como se muestra en la siguiente ilustración.

```

% --- Executes on button press in Adelante.
function Adelante_Callback(hObject, eventdata, handles)

global bluet

fwrite(bluet,2,'uint8');
pause(0.05);

% hObject handle to Adelante (see GCBO)
% eventdata reserved - to be defined in a future version of
% handles structure with handles and user data (see GUIDA

% --- Executes on button press in Derecha.
function Derecha_Callback(hObject, eventdata, handles)

global bluet

fwrite(bluet,3,'uint8');
pause(0.05);

% hObject handle to Derecha (see GCBO)
% eventdata reserved - to be defined in a future version of
% handles structure with handles and user data (see GUIDA

```

ILUSTRACIÓN 88 ENVÍO DE ACCIONES AL ROBOT ORUGA MOVIMIENTO LIBRE

El siguiente juego que presentaremos es el Evita-Objetos. En este se hace uso de la función “Evitaobjetos.m”

## Evitaobjetos.m

Este juego en cuanto a programación es idéntico al “Siguelinea” tan solo se diferencia en el tipo de actividad que será enviada al robot tipo oruga.

```
% --- Executes on button press in btejecutar.
function btejecutar_Callback(hObject, eventdata, handles)
global chofin bolusfin choman bolusman bluet

d=str2double( get(handles.edias,'string'));
I=str2double( get(handles.eintervalo,'string'));

estado6 = get(handles.rej6,'Value');
if estado6==1
    chofin=choman;
    bolusfin=[bolusman ones(5,1)];
end

actividad = 'e';           % Actividad Evitaobjetos
Simulador(actividad,d,I,chofin,bolusfin)
```

ILUSTRACIÓN 89 ACTIVIDAD EVITAOBJETOS

## Siguelineamanual.m

El siguiente juego se trata de una variación del juego siguelinea. En esta variación, la introducción de los patrones de ingestas es de forma manual por parte del usuario haciendo uso de los slider montados sobre el robot.

Para la adquisición de los datos de los slider se hace uso de la función asociada al botón “obtenervalores”. Cuando el botón es pulsado se ejecuta un código en función de la opción escogida por el usuario:

1. Modo simulación: No se realiza ninguna función concreta, tan solo se envía el fin de comunicación al robot.

```
% --- Executes on button press in obtenervalores.
function obtenervalores_Callback(hObject, eventdata, handles)
recibidodesayuno=[];           %Carbohidratos
recibidodesayuno=[];         %Dosis insulina uu
recibidocomida=[];           %Carbohidratos
recibidocomida=[];           %Dosis insulina uu
recibidocena=[];             %Carbohidratos
recibidocena=[];             %Dosis insulina uu

global bluet dosisinsulinadesayuno carbohidratosdesayuno dosis
flushinput(bluet);
flushoutput(bluet);

fwrite(bluet,'f','uint8');     % Fin de comunicación
pause(0.05);
simulacion = get(handles.rejsimu,'Value');

while simulacion==0 ...

% hObject handle to obtenervalores (see GCBO)
```

ILUSTRACIÓN 90 CAPTURA DE DATOS MODO SIMULACIÓN

2. Modo desayuno: en este modo se capturan los datos correspondientes a la ingesta de carbohidratos e insulina en la hora del desayuno.

```

if desayuno == 1                                %Capturamos los datos para el desayuno

%-----Petición y recepción de Dosis Carbohidratos-----
    fwrite(bluet,'c','uint8');
    pause(0.05);
    for j=1:1:6
        recibidodesayuno(j)=fscanf(bluet,'%d');
    end
    pause(0.1);

    carbohidratodesayuno = montarfloat(recibidodesayuno);
    pause(0.05);

    set(handles.gramosggdesay,'string',num2str(carbohidratodesayuno));
    pause(0.05);

%-----Petición y recepción de Dosis Insulina-----
    fwrite(bluet,'u','uint8');
    pause(0.05);
    for j=1:1:6
        dosisinsulinadesayuno = montarfloat(recibidodesayuno);
    end
    pause(0.05);

    set(handles.Dosisuodesay,'string',num2str(dosisinsulinadesayuno));
    pause(0.05);
end

if comida == 1                                %Capturamos los datos de la comida

```

ILUSTRACIÓN 91 CAPTURA DATOS DESAYUNO

3. Modo comida: de manera similar al modo desayuno son capturados los datos de la comida.
4. Modo cena: por último son capturados los datos para la cena.

El modo es seleccionado por el usuario en la pantalla de configuración del juego.

Por último, para empezar el juego se pulsa sobre el botón “Iniciar Simulación”. Con esta acción se ejecuta su función asociada que asigna los valores obtenidos a las matrices “chofin” y “bolusfin”

```

function btejecutar_Callback(hObject, eventdata, handles)
global chofin bolusfin dosisinsulinadesayuno carbohidratodesayuno dosisinsulinadesayuno

chofin=[480.000000 carbohidratodesayuno; 840.000000 carbohidratodesayuno];
bolusfin=[480.000000 dosisinsulinadesayuno 1.000000; 840.000000 dosisinsulinadesayuno];

permiso=0;
pause(0.5);
flushinput(bluet);
flushoutput(bluet);
pause(0.1);
fwrite(bluet,'f','uint8'); % Reseteamos comunicación
d=1;
I=20;
actividad='a';

simulador = get(handles.rejsimu,'Value');

if simulador == 1
    Simulador(actividad,d,I,chofin,bolusfin)
end

```

ILUSTRACIÓN 92 EJECUCIÓN JUEGO SIGUELINEA MANUAL

La plataforma, como hemos visto, está compuesta por dos modos de funcionamiento principales. Ya se han mostrado los juegos asociados al modo de tratamiento con bomba de insulina. A continuación, se muestran los juegos con tratamiento de páncreas artificial.

## SiguelineaB.m, EvitaobjetosB.m y SiguelineamanualB.m

Estos tres juegos pertenecientes al tratamiento con páncreas artificial tienen una estructura, en cuanto a programación, similar a sus iguales con tratamiento de bomba de insulina.

Las dos principales diferencias están relacionadas con el control PID que regula el nivel de glucosa en sangre del paciente virtual.

Por un lado, es necesaria la introducción, por parte del usuario, de los parámetros del controlador. La adquisición de estos datos se implementa, en cuanto a programación, de una manera similar a la adquisición de los patrones de ingestas en el modo de bomba de insulina.

Por otro lado, la ejecución del paciente virtual incorpora, además de la simulación con bomba de insulina, una simulación del paciente con páncreas artificial. Esto es enviando los datos de glucosa al robot y este devolviendo una señal de control como ya hemos visto en la memoria.

Por todo esto, la función a la que acceden los juegos con páncreas artificial es otra, la función “SimuladorB.m”.

## SimuladorB.m

En esta función realizan dos simulaciones de paciente independientes: una de un paciente con bomba de insulina idéntica a la de la función “Simulador.m” y otra con el paciente tratado mediante un páncreas artificial. El objetivo de realizar las dos simulaciones es poder mostrar una comparación de ambas.

Desde los juegos se accede a la función “Simulador.m” de manera similar a los juegos con bomba de insulina.

```
actividad='s'; %Actividad Siguelinea
simulacion = get(handles.rejsimu, 'Value');
if simulacion==1
SimuladorB(actividad,d,I,filtro,cbolus,chofin,bolusfin,referencia)
end
```

ILUSTRACIÓN 93 LLAMADA FUNCIÓN SIMULADOR.M

La llamada a “Simulador.m”, como podemos observar en la ilustración anterior, es similar a la de la función con bomba de insulina, tan solo se añaden variables relacionadas con el controlador: el filtro del regulador, un compensador de los bolus de insulina y la referencia de glucosa a alcanzar.

En el código de la función, en primer lugar, se ejecuta el simulador con bomba de insulina y se representa la información en un gráfico.

En segundo lugar, se ejecuta el simulador de paciente con páncreas artificial, es decir, se interactúa con el robot tipo oruga que realiza la función de un controlador PID.

El programa envía el nivel de glucosa en sangre del paciente virtual como se muestra en la ilustración siguiente.

```
glucosa=(18*X(end,9)/vg);  
  
enviado = desmontarfloat(glucosa,'g');  
for i=1:1:6  
fwrite(bluet,enviado(i),'uint8');  
pause(0.05);  
end
```

**ILUSTRACIÓN 94 ENVÍO DEL NIVEL DE GLUCOSA**

Con el nivel de glucosa en sangre actual, la referencia, y los parámetros del regulador (Los datos: referencia de glucosa en sangre y los parámetros PID están ya en posesión del robot tipo oruga ya que han sido introducidos por el usuario mediante los sliders) el robot, a petición de la función “SimuladorB.m” puede calcular y enviar de vuelta la señal de control para el sistema.

```
%-----Envío de petición de cálculo PID a Arduino  
  
fwrite(bluet,'p','uint8');  
pause(0.05);  
  
for j=1:1:6  
recibido(j)=fscanf(bluet,'%d');  
end  
factorcontr = montarfloat(recibido);
```

**ILUSTRACIÓN 95 PETICIÓN DE CÁLCULO PID**

Al recibir la señal de control desde el robot se debe tratar el paquete recibido. Para esta acción se ha implementado la función “montarfloat” que transforma el dato codificado para su envío en un valor tipo float.

```

function [float] = montarfloat(recibido)

float = recibido(4)*100;
float = float + recibido(5);
float = float/1000;
if recibido(2)== 1 || recibido(2)== 129
    recibido(3)= recibido(3)*(-1);
end
float = float*10^recibido(3);
if recibido(2)>127
    float = float*(-1);
end
return

```

**ILUSTRACIÓN 96 DECODIFICACIÓN DE DATOS RECIBIDOS**

La señal de control recibida actúa directamente en el nivel de insulina en sangre de nuestro paciente virtual, tal como se ve en la siguiente ilustración.

```

if i>3*I || i==3*I
    if basal(end,end)-factorcontr>filtro(2)*(0.4/60)
        insulina=filtro(2)*(0.4/60);

    elseif basal(end,end)-factorcontr<filtro(1)*(0.4/60)
        insulina=filtro(1)*(0.4/60);

    else
        insulina=(basal(end,end)-factorcontr);
    end
    basal=[basal; i insulina];
end

```

**ILUSTRACIÓN 97 AJUSTE DEL NIVEL DE INSULINA**

Los resultados obtenidos son presentados de manera gráfica por pantalla de PC.

## Software Arduino

El lenguaje de programación de Arduino está basado en C++ y aunque la referencia para el lenguaje de programación de Arduino lo podemos encontrar en su página web (Arduino, 2017) también es posible usar comandos estándar de C++.

El código implementado en el robot representa el papel de un “cliente”. Responde a estímulos enviados por el programa del PC.

Para presentar la aplicación describiremos las partes y funciones más significativas.

Como primera acción en el código se ha incluido las librerías “MeAuriga.h” y “SoftwareSerial.h” necesarias para la interacción con el robot y las comunicaciones respectivamente.

Para poder trabajar con todos los sensores y actuadores del robot es también necesario definir las variables mostradas en la ilustración siguiente.

```

//-----Bluetooth-----//
MeBluetooth bluetooth(PORT_5);

//-----Detectores siguelinea-----//
MeLineFollower linefollower_6(6);

//-----Ultrasonidos-----//
MeUltrasonicSensor ultrasonic_7(7);

//-----Slider-----//
MeJoystick joystick_8(8);
MeJoystick joystick_9(9);
MeJoystick joystick_10(10);

//-----LED-----//
MeRGBLed rgbled_0(0, 12);

//-----Encoder Motor-----//
MeEncoderOnBoard Encoder_1(SLOT1);
MeEncoderOnBoard Encoder_2(SLOT2);

```

**ILUSTRACIÓN 98** SENSORES Y ACTUADORES DEL ROBOT

En cuanto al código principal, como se ha comentado, el comportamiento de robot es pasivo: a la espera de recibir órdenes por parte del PC.

Cuando el PC requiere de cualquier actividad por parte del robot, se envía en primer lugar un dato con el tipo de juego al que pertenece. El robot por tanto ajusta sus patrones de cálculo y comportamiento en base a esta información recibida.

```

//-----DISCRIMINACION DE MODO DE OPERACION DE ROBOT-----

if (serial=='s' && com1==0){ //Siguelinea con bomba o sin ella
  Serial.println(1);
  glucosa=0; //Inicializamos a cero las variables qu
  control=0;
  controlant=0;
  eact=0;
  eant=0;
  eantant=0;
  esum=0;
  indice=0;
  modo=1; //Simulación de paciente
  delay(5);}

if (serial=='m' && com1==0){ //Movimiento
  Serial.println(1);
  glucosa=100; //Inicializamos a cero las variables
  control=0;
  controlant=0;
  eact=0;

```

**ILUSTRACIÓN 99** DISCRIMINACIÓN DE TIPO DE JUEGO

En base al tipo de juego seleccionado se ejecutarán unas funciones u otras. Como ejemplo en la siguiente ilustración podemos ver una sección del código donde se responde a una petición por parte del PC (enviando la letra “c” después de haber enviado la letra

correspondiente para el tipo de juego). Se solicita el dato de carbohidratos. Esta opción como se observa está disponible para los modos 1, 3 y 4 que son los siguelineas y los evitaobjetos.

```

if (modo==1 || modo==3 || modo==4){ //Comunicaciones con Matlab para modos 1 3 4
                                     //y acción de control para estos

//-----Envío Carbohidratos-----//

  if (serial=='c' && com1==0){

    temp = joystick_8.read(1);
    carbohidratos=map(temp,-490,533,0,100);

    desmontarfloat(carbohidratos, paquetec);

    for (int i = 0; i<6; i++){
      Serial.println(paquetec[i]);
      delay(5);}
      delay(5);}

```

**ILUSTRACIÓN 100 COMUNICACIONES MODOS 1, 3 Y 4**

Las funciones “montarfloat” y “desmontarfloat” son equivalentes a sus iguales en código Matlab.

De manera similar se realiza el envío de carbohidratos, los parámetros PID o la referencia.

Una vez recibido y decodificado el dato del nivel de glucosa es almacenado. Cuando se solicita un cálculo PID se hace uso de este dato para proponer una señal de control, como se muestra a continuación.

```

//-----Petición calculo PID-----//

  if (serial=='p' && com1==0){
    peticionpid=1;}

//-----Calculo PID-----//

  if (peticionpid==1 && com1==0){
    delay(0.05);
    eact=ref - glucosa;
    esum+=eact;

    control=(factork*eact) + factori*tmuestreo*esum - factord*(eact-eant)/tmuestreo;

    desmontarfloat(control, paquetee);

//-----Envío señal de control-----//

    for (int i = 0; i<6; i++){
      Serial.println(paquetee[i]);
      delay(5);}

    eantant=eant;
    eant=eact;
    controlant=control;
    peticionpid=0;}

```

**ILUSTRACIÓN 101 CÁLCULO PID**

Con este código se soluciona la primera tarea principal del robot: servir de controlador PID y de adquisición de datos.

En cuanto a la representación en forma de movimientos y colores del nivel de glucosa en sangre (segunda tarea principal del robot) se han implementado una serie de instrucciones para tal efecto. Estas instrucciones están influidas por el tipo de juego seleccionado en el PC y por el nivel de glucosa en el instante actual.

Como ejemplo se muestra el movimiento del robot en el caso de que el juego seleccionado sea algún siguelinea y el paciente se encuentre en normoglucemia ( $70 < \text{glucosa} < 150 \text{ mg/dL}$ ).

En primer lugar, se asignan los valores para los parámetros de entrada de la función “move” según el nivel de glucosa.

```
else if (glucemia==2 || glucemia==0){ //NORMOGLUCEMIA

    velocidad=105;           // Velocidad del Robot Media
    retardo =1;              // Sin retardo en la acción del robot
    acentuacurva = 0;        // No se acentúan las curvas realizadas
    zigzag=0;                // Permite al robot avanzar en zig zag (solo para evita-obtetos)
    if (curvacerrada==1){    // Discrimina el tipo de curva que se encuentra el Robot
        ajusterecta = 1;}
    else{
        ajusterecta = -10;}} // Ajuste para mejorar el comportamiento en rectas
```

#### ILUSTRACIÓN 102 MOVIMIENTOS EN NORMOGLUCEMIA

Para suavizar el movimiento en las rectas y potenciar el giro en las curvas se ha implementado un algoritmo que identifica cuando el robot se encuentra frente a una curva cerrada o frente a una curva. A continuación, se muestra su código comentado.

```
if (sensoranterior != Sensor){ // Si hay un cambio de estado en los sensores

    if (sensoranterior==3 || sensoranterioranterior==3 || Sensor==3){
        curvacerrada=1;} // Curva cerrada si en el estado actual o los anteriores
                        // si los 2 detectores se han salido de la linea
    if ((Sensor==2 && sensoranterior==0 && sensoranterioranterior==1) || (Sensor==1 && sensoranterior==0 && sensoranterioranterior==2)){
        curvacerrada=0;} // Fin de curva cerrada si uno de los sensores se sale de la linea
                        // por un lado cuando en el estado previo estaba dentro de la linea y
                        // anteriormente a ese último estado el otro sensor había salido por el otro
                        // lado de la linea

    sensoranterioranterior=sensoranterior;
    sensoranterior=Sensor;}
```

#### ILUSTRACIÓN 103 CURVA CERRADA - RECTA

En segundo lugar, ya que nos encontramos en el modo siguelinea, se debe hacer uso de los sensores siguelinea para seleccionar el tipo de movimiento a realizar: avance recto, giro izquierda, giro derecha o retroceso.

```

switch (Sensor){

case 0:    //Ningún sensor detecta, Oruga inicia marcha recto
    move(1, velocidad, retardo, ajusterecta, acentuacurva, zigzag);
    break;

case 1:    // Sensor 2 detecta, Oruga gira a la izquierda
    move(2, velocidad, retardo, ajusterecta, acentuacurva, zigzag);
    break;

case 2:    //Sensor 1 detecta, Oruga gira a la derecha
    move(3, velocidad, retardo, ajusterecta, acentuacurva, zigzag);
    break;

case 3:    //Los 2 sensores detectan, Oruga inicia marcha hacia atrás
    move(4, velocidad, retardo, ajusterecta, acentuacurva, zigzag);
    break;
}

```

#### ILUSTRACIÓN 104 MOVIMIENTOS SIGUELINEA

Por último, con los parámetros afectados por el nivel de glucosa en sangre y el tipo de movimiento indicado por los sensores siguelinea, se ejecuta la función “move” que actúa directamente sobre los motores de avance del robot tipo oruga.

```

void move(int direction, int fvelocidad, int fretardo, int fajusterecta, int facentuacurva, int zigzag)
{

    if(direction == 1){                //Oruga inicia marcha recto
        delay(fretardo);
        motorizquierdo = fvelocidad + zigzag;
        motorderecho = fvelocidad - zigzag;
    }else if(direction == 2){         //Oruga gira a la izquierda
        delay(fretardo);
        motorizquierdo = -fvelocidad/fajusterecta;
        motorderecho = fvelocidad + facentuacurva;
    }else if(direction == 3){        //Oruga gira a la derecha
        delay(fretardo);
        motorizquierdo = fvelocidad + facentuacurva;
        motorderecho = -fvelocidad/fajusterecta;
    }else if(direction == 4){       //Oruga inicia marcha hacia atrás
        delay(fretardo);
        motorizquierdo = -fvelocidad;
        motorderecho = -fvelocidad;
    }

    Encoder_1.setTarPWM((-1)*motorderecho);    //Cadena derecha
    Encoder_2.setTarPWM(motorizquierdo);      //Cadena izquierda
}

```

#### ILUSTRACIÓN 105 FUNCIÓN MOVE

Como vemos en el código de la ilustración para realizar los distintos movimientos se varían el sentido y las velocidades de los motores correspondientes a las cadenas del robot.

En cuanto a los juegos evitaobjetos, los parámetros de la función “move” son distintos, así como los movimientos a realizar.

Para conseguir la acción evita-objetos se ha hecho uso de los sensores de ultrasonidos instalados en el robot. Se ha parametrizado la distancia de detección en función del nivel de glucosa, consiguiendo de esta manera que la acción sea más torpe a niveles no deseados.

```
//Parametrización de la distancia de detección

if (glucemia==2){           //Normoglucemia
    deteccion=20;}         //distancia de detección correcta

else if (glucemia==3){     //Hiperglucemia
    deteccion = 20 - ((glucosa - 100)/9);} // Distancia de deteccion mas baja
                                     // Inversam. proporcional a hiperglucemia

else if (glucemia==1){     //Hipoglucemia
    deteccion = 15 -(100-glucosa)/5;} // Distancia de deteccion mas baja
                                     // Inversam. proporcional a hipoglucemia
```

### ILUSTRACIÓN 106 DISTANCIA DE DETECCIÓN DE OBJETOS

Una vez obtenida la distancia a la que reaccionará el robot se ha desarrollado un algoritmo que propone los movimientos concretos para evitar el objeto. Como se ve en el código siguiente al detectar un objeto retrocede durante un segundo y gira durante 1,2 segundos.

```
if (objeto==0 && esquivando==0){ //No hay objeto: Inicia marcha recto hacia adelante

    move(1, velocidad, retardo, ajusterecta, acentuacurva, zigzag);
    inicio=1;
}

else{ //Esquiva objeto
    if (inicio==1) {
        tiempo=millis();
        inicio=0;}
        tiempo2=millis();
        controltiempo=tiempo2-tiempo; //Gestion de tiempos para el retroceso y giro

    if (controltiempo<1000){ // Inicia retroceso
        move(4, velocidad, retardo, ajusterecta, acentuacurva, zigzag);
        esquivando=1;}
    else if (controltiempo>=1000 && controltiempo<2200){ //Inicia giro
        move(2, velocidad, retardo, ajusterecta, acentuacurva, zigzag);
        esquivando=1;}
    else{
        esquivando=0;}}
```

### ILUSTRACIÓN 107 MOVIMIENTOS EVITA OBJETOS

También señalar que se ha implementado una solución para que el robot avance en zigzag cuando los niveles de glucosa no son los adecuados.

La segunda parte de la representación del estado glucémico del paciente es el anillo de leds montado sobre el robot. El anillo actúa de una forma distinta para cada valor de glucosa en sangre obtenido desde el paciente virtual. A continuación, mostramos las tres funciones que definen el color de los leds (Azul, Verde y Rojo). Se trata de dos funciones exponenciales y una

campana de Gauss. Esta representación permite un color para cada nivel de glucosa y una transición suave desde el color más rojo en los casos más graves de hipoglucemia al color más verde cuando la hiperglucemia es muy elevada.

```
//-----Azul-----//  
  
if (glucosa>180){ //HIPERGLUCEMIA  
Azul=255;}  
else if (glucosa<100){  
Azul=0;}  
else{  
Azul=pow(2,((glucosa-50)/20)); // Funcion exponencial inversa  
  
//-----Verde-----//  
if (glucosa>180 || glucosa<65){ //NORMOGLUCEMIA  
Verde=0;  
}  
else{  
Verde=110*pow(2.72, (-1*pow((glucosa-120),2)/1000)); //Campana de Gauss  
  
//-----Rojo-----//  
  
if (glucosa<65){ //HIPOGLUCEMIA  
Rojo=255;}  
else{  
Rojo=800*pow(0.95,glucosa); //Función exponencial  
}
```

ILUSTRACIÓN 108 ANILLO DE LEDS