



Universidad Carlos III  
Curso Ingeniería del Software 2022-23  
Práctica

*diabetesFREE*

Fecha: **20/10/2022** - ENTREGA: **20/10/PRÁCTICA 1**

GRUPO: **80** EQUIPO: **03**

Alumnos: **Miguel Avella Miravet / Rubén de Arriba Viejo**  
**Juan Marín Olloqui/ Ana María Ortega Mateo/ David Sierra Fernández**

## PÁGINA DE ESTADO DE DOCUMENTO

### Registro de Cambios

Versión	Fecha	Autor	Cambios en la versión
0	13/10/2022	Ana Ortega	Primer borrador.
0.1	14/10/2022	Ana Ortega	Primera versión de los requisitos
0.1	14/10/2022	Rubén de Arriba	Primera versión de los requisitos
0.2	18/10/2022	Juan Marín	Vocabulario y tabla de acoplamientos
0.2	18/10/2022	David Sierra	Vocabulario y tabla de acoplamientos
0.3	19/10/2022	Ana Ortega	Revisión y perfeccionamiento del documento
0.3	19/10/2022	Miguel Abella	Revisión y perfeccionamiento del documento
0.3	19/10/2022	Rubén de Arriba	Revisión y perfeccionamiento del documento

### Horas invertidas en el proyecto

NOMBRE	INDIVIDUAL	EQUIPO	TOTAL
Miguel Abella Miravet	3	3	<b>6</b>
Rubén de Arriba Viejo	3	6	<b>9</b>
Juan Marín Olloqui	2	3	<b>5</b>
Ana María Ortega Mateo	4	6	<b>10</b>
David Sierra Fernández	2	3	<b>5</b>
<b>TOTAL</b>	<b>14</b>	<b>21</b>	<b>35</b>

### Distribución de responsabilidades

NOMBRE	Descripción de sus principales contribuciones al proyecto
Miguel Abella Miravet	Revisión y perfeccionamiento del documento.
Rubén de Arriba Viejo	Requisitos funcionales y revisión y perfeccionamiento del documento.
Juan Marín Olloqui	Vocabulario y tabla de acoplamiento
Ana María Ortega Mateo	Primera versión requisitos no funcionales y revisión y perfeccionamiento del documento.
David Sierra Fernández	Vocabulario y tabla de acoplamiento

## TABLA DE CONTENIDO

<b>1</b>	<b>INTRODUCCIÓN</b>	<b>4</b>
1.1	PROPÓSITO DEL DOCUMENTO	4
1.2	VISIÓN GENERAL DEL DOCUMENTO	4
1.3	REFERENCIAS	5
<b>2</b>	<b>DESCRIPCIÓN GENERAL</b>	<b>6</b>
2.1	PERSPECTIVA DEL PRODUCTO	6
2.2	ALCANCE DEL SOFTWARE	6
2.3	CAPACIDADES GENERALES	7
2.4	RESTRICCIONES GENERALES	7
2.5	CARACTERÍSTICAS DE LOS USUARIOS: ROLES Y CAPACIDADES	8
2.6	ENTORNO OPERACIONAL	10
<b>3</b>	<b>REQUISITOS</b>	<b>11</b>
3.1	JUSTIFICACIÓN DE LA CLASIFICACIÓN DE REQUISITOS	11
3.2	JUSTIFICACIÓN DE LA PLANTILLA DE REQUISITOS	11
3.3	REQUISITOS FUNCIONALES	12
3.4	REQUISITOS NO FUNCIONALES	17
3.5	VOCABULARIO DEL DOMINIO	23
	3.5.1 VOCABULARIO DEL MODELO CONCEPTUAL	23
	3.5.2 VOCABULARIO TÉCNICO	24
<b>5</b>	<b>LISTAS Y TABLAS</b>	<b>26</b>
5.1	CONSISTENCIA ENTRE REQUISITOS: CONFLICTOS, REDUNDANCIAS, ACOPLAMIENTOS	26

## **1. INTRODUCCIÓN**

### **1.1. Propósito del documento**

Es un documento de especificación de la implementación del sistema. El objetivo principal, es servir como medio de comunicación entre usuarios, empresas y desarrolladores. Deberá recoger las necesidades de clientes y cumplir con las condiciones y capacidades del sistema deseado. El documento servirá de guía para establecer los requisitos del software a desarrollar con la finalidad de satisfacer los requisitos del propio usuario. Será accesible para todo el personal involucrado en el proyecto.

Este documento es una guía común para todas las partes involucradas con carácter contractual. Deberá ser validado con el cliente para asegurarse de haber cumplido con sus expectativas y verificado en caso de complejidad con el equipo desarrollador.

### **1.2. Visión general del documento**

Este documento se divide en diversas partes: la introducción, la descripción general, los requisitos, la arquitectura y las listas y tablas.

La introducción aportará al lector una visión general del documento, especificando los propósitos y las referencias utilizadas. La descripción general servirá para dar perspectiva del producto, indicar el alcance, las capacidades y restricciones del sistema. También, especificará las características del usuario y el entorno operacional de funcionamiento.

La parte de los requisitos consta con la justificación de la clasificación y del conjunto de atributos elegidos para la configuración de la plantilla. Los requisitos se dividirán en funcionales y no funcionales teniendo en cuenta las capacidades y restricciones establecidos. También incluirá un vocabulario de dominio en el que se aclararán los términos tanto conceptuales como técnicos.

La arquitectura es una sección en la que se establecerán los distintos modelos conceptuales y de arquitectura del sistema y se especificarán las propiedades de cada uno de los componentes identificados.

Por último, las listas y tablas, que proporcionarán de manera visual los conflictos, redundancias y acoplamientos de los requisitos y su trazabilidad con respecto a la información e implementación.

También consta de una portada, un índice y un registro de las horas invertidas, la distribución y los cambios en el proyecto por cada uno de los participantes.

### 1.3. Referencias

Diabetes, Medlineplus (Biblioteca Nacional de Medicina),  
<https://medlineplus.gov/spanish/ency/article/001214.htm#:~:text=Ex%C3%A1menes%20de%20sangre%3A,glucosa%20en%20ayunas%20o%20prediabetes.>

Tratamiento con Sistema Integrado, Medtronic,  
<https://www.medtronic-diabetes.com/es-ES/sobre-la-diabetes/terapia-con-bomba-de-insulina>

Jeringas de insulina, Cigna,  
<https://www.cigna.com/es-us/knowledge-center/hw/jeringas-de-insulina-hw39086>

URL fichero Google Drive:  
<https://docs.google.com/document/d/1HNRRTaKK4quP5agk7zPvg097Rx9P4ZcC/edit#>

## **2. DESCRIPCIÓN GENERAL**

### **2.1. Perspectiva del producto**

El sistema *diabetesFREE* llega para ser un sustituto de las inyecciones de insulina y de los glucómetros. Los usuarios podrán de esta manera mantener un día a día normal sin tener que preocuparse de medir su nivel de glucosa en sangre o tener que inyectarse la insulina.

Nuestro producto contra la diabetes no es el único, existen instrumentos parecidos que realizan funciones similares a las del producto final. Existe un sistema muy parecido a nuestro sistema *diabetesFREE* desarrollado por la empresa Medtronic, MiniMed, que se encarga de medir la glucosa en sangre e inyectar dosis de insulina en el organismo del usuario. Otro instrumento que cumple con la funcionalidad de medir la concentración de glucosa en sangre, al igual que nuestro sistema, es el glucómetro.

### **2.2. Alcance del software**

Se generarán dos productos principalmente: una aplicación para móvil, *diabetesFREE APP*, y un sistema integrado, *diabetesFREE system*. Además, se utilizará un servidor para almacenar información y facilitar la comunicación entre usuarios del sistema general.

El objetivo de nuestro sistema es ayudar a los pacientes de diabetes a llevar una vida normal. El sistema deberá medir el nivel de azúcar en sangre del usuario y calcular e inyectar la dosis de insulina correspondiente en base al nivel de azúcar en sangre. Logrando estos objetivos, los pacientes de diabetes podrán llevar a cabo una vida normal sin tener que preocuparse tanto por su patología.

Otro de los objetivos de nuestro sistema será la administración de datos. El sistema deberá ser capaz de almacenar la información y llevar a cabo un registro de los tratamientos del paciente. Independientemente de su estado.

Nuestro sistema estará formado por una bomba miniaturizada, que incorporará una aguja. Habrá que tener en cuenta que la aguja es frágil y se podrá producir su rotura, suponiendo un grave problema. Se deberá acudir a un centro sanitario y se reajustará el sistema para que continúe con su funcionamiento. Al implicar negativamente en la salud del usuario, los desarrolladores deberán prestar especial atención.

Con este sistema se pretende ayudar a mejorar la salud de nuestros pacientes y facilitar el trabajo de los servicios sanitarios. Reduciendo el nivel de riesgo de los pacientes con diabetes y bajo un control constante.

### **2.3. Capacidades generales**

El sistema ofrecerá una nueva forma de administrar las inyecciones de insulina a los usuarios. No tendrán que preocuparse de las cantidades ni de los horarios porque nuestro sistema se encargará de ello. El microprocesador se encargará de medir la conductividad eléctrica de la sangre y la bomba calculará el nivel de azúcar y la cantidad de insulina exacta que el cliente necesite.

El paciente no tendrá que preocuparse de las inyecciones. DiabetesFREE proporciona en su sistema una bomba miniaturizada que se encargará de suministrar la insulina por medio de una aguja.

El objetivo primordial de este sistema es no poner en riesgo la salud del cliente. Cuando el sistema se esté quedando sin insulina, el controlador avisará al usuario su reemplazo con suficiente antelación para prevenir riesgos.

También se ha pensado en la comodidad del cliente. Para facilitar el uso del sistema, se creará una aplicación móvil como complemento. En ella se proporcionará información constante sobre su historial de tratamientos y recomendaciones e indicaciones de especialistas, que aportarán seguridad y confianza. Existirá un modo manual por el cual el propio paciente tendrá la libertad de realizar un tratamiento específico según su historial de tratamiento. El cliente estará en contacto directo con el tratamiento, pero nunca se pondrá en riesgo su salud.

La diabetes es una enfermedad compleja que requiere atención continua. DiabetesFREE proporcionará seguridad y facilidad al cliente. Estará continuamente en funcionamiento y libre de preocupaciones en relación con las dosis de insulina, los horarios, las cantidades e inyecciones.

### **2.4. Restricciones generales**

El sistema contará con varias restricciones con el fin de no poner en riesgo la salud del cliente y facilitar su vida diaria. El sistema no podrá presentar ningún tipo de error.

La portabilidad es un factor importante para aportar comodidad al usuario. El sistema no deberá pesar más de 0,5 gramos. La diabetes es una enfermedad de gran riesgo que pone en peligro la vida de las personas, por ello, se tendrá en cuenta la rapidez de nuestro sistema. Deberá calcular e inyectar la dosis correspondiente de insulina en un intervalo de tiempo de menos de cinco segundos. La constancia es un factor importante para el continuo registro del estado del cliente, con el objetivo de un seguimiento continuo. DiabetesFREE deberá medir el nivel de azúcar en sangre cada hora. La disponibilidad del sistema también será continua. Se deberá mostrar al usuario los datos

almacenados en el sistema a cualquier hora del día (nivel de azúcar en sangre, cantidad de insulina restante o tiempo hasta la próxima medición).

En el caso de que algún error sea detectado. El sistema se pondrá en contacto directo con el paciente y dependiendo de la gravedad, con los servicios sanitarios. La exactitud y precisión son imprescindibles. DiabetesFREE proporcionará la dosis exacta de insulina necesaria en el momento adecuado.

La privacidad del cliente también es un aspecto a tener en cuenta. Solo el y su centro sanitario correspondiente o especialista podrá tener acceso a toda la información sobre su estado. El historial de tratamientos del paciente quedará registrado en la aplicación. De esta forma, en caso de riesgo, se podrán analizar sus resultados.

El sistema deberá de tener en cuenta los aspectos legales. Cumplirá con el Real Decreto 1591/2009 de 16 de octubre por el que se regulan los productos sanitarios y con la recomendación europea Directiva 93/42/CE de 14 de junio de 1993. Todas estas restricciones son las que deberá tener el sistema predeterminadamente, pero posteriormente se podrán modificar según el tratamiento específico de cada usuario.

## 2.5. Características de los usuarios: roles y capacidades

<b>Roles</b>	<b>Funciones/Servicios</b>	<b>Estimación</b>
Usuarios(expertos)	Hacer uso del sistema. Son clientes que se encuentran en un estado muy grave de diabetes y necesitan ayuda diaria.	59%
Usuarios(ocasionales)	Hacer uso del sistema. Se encuentran en las fases iniciales de la diabetes.	26%
Especialistas/ Servicios sanitarios	Analizar los historiales de los pacientes. Tienen el acceso total de los registros diarios y deberán estar alertas de la situación ante un posible mal funcionamiento.	15%

Los roles que este servicio tiene son muy variados y un rol podrá estar compuesto por muchas personas. Cuando nos referimos a su nivel educativo, dominio del idioma, experiencia y conocimiento técnico, son las expectativas que se esperan de las personas que desempeñan dicho rol.



<b>Roles</b>	<b>Nivel educativo</b>	<b>Dominio del idioma</b>	<b>Experiencia</b>	<b>Conocimiento técnico</b>
Usuarios(expertos)	Medio	Alto	Alta	Alto
Usuarios(ocasionales)	Medio	Alto	Baja	Bajo
Especialistas/ Servicios sanitarios	Alto	Alto	Alta	Alto

En cuanto a la interfaz del sistema. Este producto consta de dos partes; una aplicación y un controlador, que en su conjunto, y apoyados por un servidor, realizan todas las funcionalidades del sistema.

El sistema integrado es un dispositivo miniaturizado de administración de insulina portátil y automatizado. Ha sido creado con el fin de estar permanentemente en contacto con el paciente. Tendrá unos botones para poder controlar el sistema. Cada botón tendrá un dibujo o palabra intuitiva para conocer su función. Un botón será el modo manual, cuando este sea presionado podrá suministrar la dosis de insulina con otro botón que tendrá el símbolo +. También tendrá un botón de volumen, que, al ser presionado, se oirá en voz alta el nivel de azúcar en sangre, la batería y la hora de la última inyección. Por último, tendrá un botón de brillo para poder mejorar la visibilidad de la pantalla. En caso de que el cliente no sepa manejarlo, la función de todos los botones quedará registrado en el manual de instrucciones. También tendrá una pantalla no táctil, en la cual se mostrará la batería, la cantidad de insulina disponible en los cartuchos, el nivel de azúcar en sangre, la hora y la última hora en la que se ha inyectado la insulina.

La aplicación, aparte de complementar al sistema, facilita su uso. Se podrá utilizar sólo en dispositivos móviles. Dispondrá de las mismas funciones que los controles situados en la propia bomba de insulina. Sin embargo, proporcionará un apartado de información que indicará el funcionamiento de todos los controles. Hemos incorporado a esta app la accesibilidad al historial de tratamientos, que será accesible cuando el usuario pulse un apartado adicional llamado *estado*. También habrá otro apartado adicional llamado recomendaciones, en el que podrá ver las indicaciones y sugerencias de los especialistas.

En cuanto a la base de datos, se podrá acceder a ella tanto desde la app como por una web disponible tanto para usuarios como para especialistas y médicos. La funcionalidad será similar a la de la app, pero sólo mostrará el historial de tratamientos, las recomendaciones y el historial de mediciones. Utilizando esta web, los especialistas y

médicos accederán a su lista de pacientes y podrán enviar las recomendaciones que consideren.

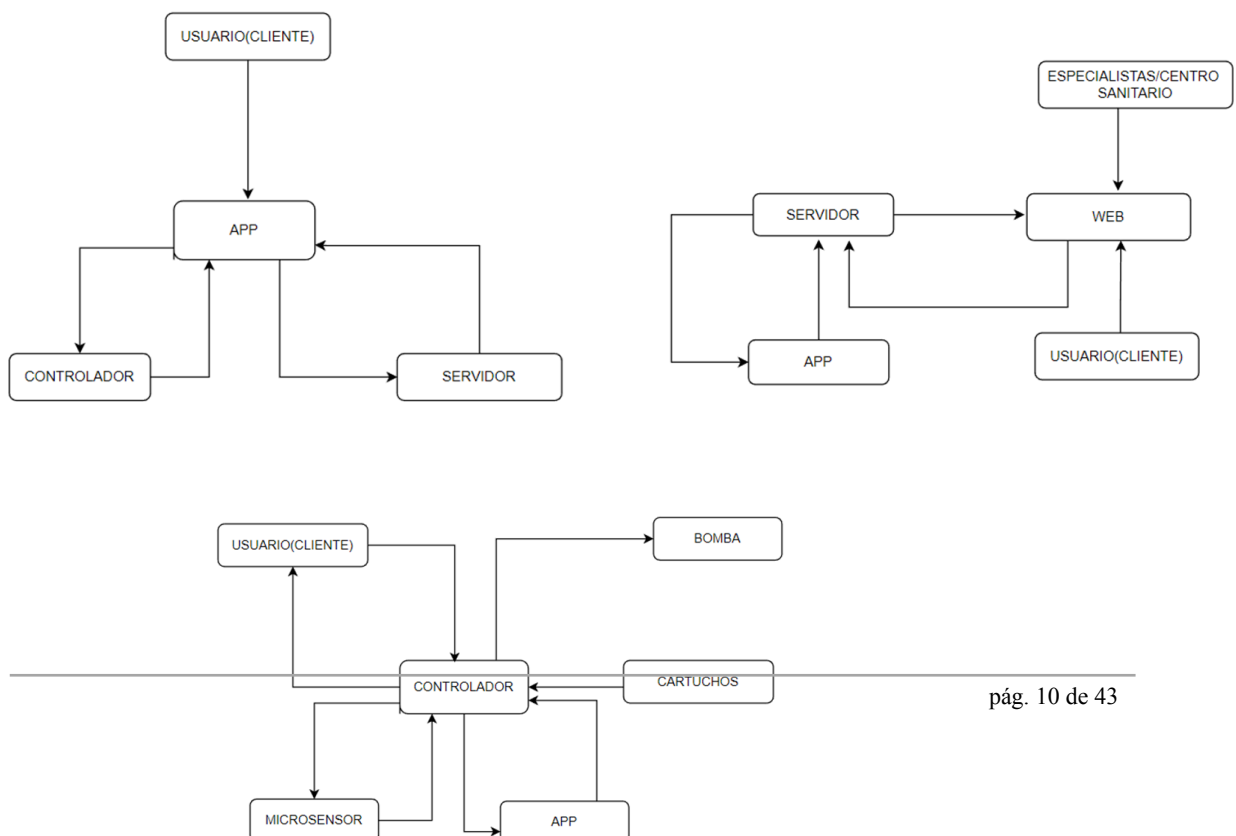
## 2.6. Entorno operacional

El producto *diabetesFREE* se compondrá de dos sistemas principales apoyados por un servidor, un aparato que medirá el nivel de azúcar en sangre y se encargará de proporcionar al organismo la insulina suficiente (llamado *diabetesFREE system*) y una aplicación que recibirá información de *diabetesFREE system* y se la proporcionará al usuario (llamado *diabetesFREE APP*).

*DiabetesFREE system* interacciona con el cuerpo del usuario. Debe estar en contacto continuo con el paciente para poder realizar su función. Deberá de ser capaz de soportar todos los movimientos que el usuario realice.

*DiabetesFREE APP* se comunicará con el usuario mostrando alertas en el teléfono móvil. Hay tres tipos de alertas. La primera es un aviso con sonido para alertar al usuario de un posible peligro, se deberá responder al aviso y tomar precauciones al respecto. Por último, en caso de que el cliente no responda ante el segundo aviso, el sistema automáticamente llamará a emergencias o al hospital de referencia del paciente. Esta aplicación será compatible con cualquier dispositivo móvil y se podrá descargar en cualquier app store.

Para el diagrama de contexto hemos decidido dividirlo en 3 partes, la base de datos, el controlador y la aplicación. Los diagramas serían los siguientes:



### **3. REQUISITOS**

#### **3.1. Justificación de la clasificación de requisitos**

La clasificación que hemos elegido es separar los requisitos por funcionales y no funcionales. Dentro de esta clasificación hemos decidido ordenarlos de manera que vayan primero los que consideramos más importantes para la función final del producto. Primero se pondrán los requisitos que intervengan en la seguridad del usuario, como la forma en la que el sistema inyecta la insulina o la forma de alertar al usuario de su nivel de azúcar en sangre ya sea alto o bajo.

Los requisitos funcionales se clasifican en disponibilidad de información, disponibilidad de medicina y modificación de datos. Esta clasificación se debe a que primero se toman los datos y la información está disponible, luego se administra la medicina en base a los datos obtenidos y la información disponible y finalmente, si se desea, se pueden modificar los datos de las dosis o el tiempo entre pruebas.

Los requisitos no funcionales, son los que imponen restricciones al producto. Los hemos clasificado en consumo de recursos, rendimiento, portabilidad, fiabilidad y disponibilidad, manejo de errores, requisitos de interfaz, restricciones y seguridad.

En cuanto al rendimiento, se establecerá la velocidad y el tiempo de respuesta tanto del propio sistema como de la aplicación.

La portabilidad, en la que indicaremos el tamaño del sistema. La fiabilidad en la que se establecerá la cuantificación de fallos permitidos y la disponibilidad.

Para no poner en riesgo al cliente, se priorizará su seguridad del cliente frente a las amenazas externas y a los fallos del sistema. También se tendrá en cuenta el manejo de errores, en el cual se establecerán los distintos tipos de respuestas ante la posible aparición de un error. Por último, se deberá respetar la privacidad del cliente imponiendo limitaciones en el uso de sus datos.

#### **3.2. Justificación de la plantilla de requisitos**

Cada uno de los requisitos será representado mediante el uso de tablas con una plantilla determinada. Se especificarán por separado los distintos atributos de cada requisito, que servirán para explicar y distinguir los requisitos.

Para configurar esta plantilla se han elegido los atributos más significativos. En primer lugar se incluirá un identificador(ID), para poder enumerar y clasificar cada requisito. El tipo de requisito para separarlos en distintas clases y misma temática, respetando el

orden de prioridad. La fecha, para mantener un orden cronológico de los requisitos. Un título y una descripción breve, que facilitará su entendimiento. Por último, el plan de pruebas de verificación, para asegurar la implementación y funcionalidad de dicho requisito.

No hemos tenido en cuenta la necesidad, consideramos que todos los requisitos son imprescindibles. En cuanto al creador y la fuente, se considerarán datos muy específicos que no aportarán demasiada distinción entre los requisitos y no se tendrán en cuenta en la plantilla.

ID		Fecha	
Título			
Tipo			
Descripción			
Pruebas			

### 3.3. Requisitos funcionales

## 3.3. Requisitos funcionales

### § Disponibilidad de información

ID	01	Fecha	14/10/2022
Título	Toma de muestras.		
Tipo	Funcional / Disponibilidad de información		
Descripción	El sistema deberá recolectar muestras de sangre del usuario.		
Pruebas	<ul style="list-style-type: none"><li>● Se probará si el sistema es capaz de recolectar líquidos.</li><li>● Si se satisface la prueba anterior, se probará en usuarios reales.</li></ul>		

ID	02	Fecha	14/10/2022
Título	Cálculo de datos.		
Tipo	Funcional / Disponibilidad de información		
Descripción	El sistema deberá calcular el nivel de azúcar en sangre.		
Pruebas	<ul style="list-style-type: none"><li>● Se tomarán dos muestras de sangre. El nivel de azúcar en sangre de una de ellas se calculará a mano y el nivel de azúcar en sangre de la otra lo calculará el sistema.</li><li>● La diferencia entre los dos cálculos deberá estar entre -1 y 1 mg/dL para que el cálculo del sistema se considere correcto.</li></ul>		

ID	03	Fecha	14/10/2022
Título	Comunicación sistema-aplicación.		
Tipo	Funcional / Disponibilidad de información		
Descripción	El sistema deberá comunicarse con la aplicación móvil.		
Pruebas	<ul style="list-style-type: none"><li>● Se meterán datos en el sistema que tendrán que ser visibles desde la aplicación móvil.</li></ul>		

ID	04	Fecha	04/12/2022
Título	Comunicación aplicación-base de datos.		
Tipo	Funcional / Disponibilidad de información		
Descripción	La aplicación deberá enviar datos (mediciones del nivel de azúcar en sangre, tiempo hasta la próxima medición...) a la base de datos.		

Pruebas	<ul style="list-style-type: none"> <li>● Se comprobará si la aplicación puede enviar datos a la base de datos.</li> <li>● Se comprobará si la información mostrada en la base de datos es correcta.</li> </ul>
---------	--

ID	05	Fecha	04/12/2022
Título	Comunicación base de datos-aplicación.		
Tipo	Funcional / Disponibilidad de información		
Descripción	La base de datos deberá enviar datos (historial de tratamientos, recomendaciones de expertos...) a la aplicación.		
Pruebas	<ul style="list-style-type: none"> <li>● Se comprobará si la base de datos puede enviar datos a la aplicación.</li> <li>● Se comprobará si la información mostrada en la base de datos es correcta.</li> </ul>		

ID	06	Fecha	17/10/2022
Título	Comunicación sistema-pantalla		
Tipo	Funcional / Disponibilidad de la información.		
Descripción	El sistema deberá mostrar a través de la pantalla la batería, la cantidad de insulina disponible en los cartuchos, el nivel de azúcar en sangre, la hora y la última hora en la que se ha inyectado la insulina		
Pruebas	<ul style="list-style-type: none"> <li>● Comprobar que se muestra toda la información indicada.</li> <li>● Comprobar que la información que se muestra por pantalla es la correcta.</li> </ul>		

ID	07	Fecha	17/10/2022
Título	Gestión de usuario.		

Tipo	Funcional / Disponibilidad de la información.
Descripción	El sistema deberá de ser capaz de distinguir a los usuarios y conocer qué función pueden desempeñar en la aplicación, a través de su perfil.
Pruebas	<ul style="list-style-type: none"><li>● Intentar entrar a una misma cuenta desde varios usuarios para ver qué capacidades puede desempeñar cada usuario.</li><li>● Modificar datos y comprobar si el sistema lo permite en función del rol establecido.</li></ul>

### § Disponibilidad de medicina

ID	08	Fecha	14/10/2022
Título	Disponibilidad de insulina		
Tipo	Funcional / Disponibilidad de medicina		
Descripción	El sistema deberá disponer de 1000 UI de insulina (equivalente a 5 cartuchos de 200 UI cada uno).		
Pruebas	<ul style="list-style-type: none"><li>● Se comprobarán posibles fugas de líquido.</li><li>● Se considerará que se ha cumplido este requisito si no se producen fugas y se conservan las 1000 UI.</li></ul>		

ID	09	Fecha	14/10/2022
Título	Uso de la insulina.		
Tipo	Funcional / Disponibilidad de medicina		
Descripción	El sistema deberá administrar las dosis de insulina correspondiente con el nivel de azúcar del paciente.		
Pruebas	<ul style="list-style-type: none"><li>● Se comprobará si el sistema es capaz de administrar insulina en el cuerpo del usuario.</li></ul>		

ID	10	Fecha	14/10/2022
Título	Gestión de la insulina.		
Tipo	Funcional / Disponibilidad de medicina		
Descripción	El sistema deberá informar al usuario de sustituir el vial de 1000 UI de insulina cuando en este queden 25 UI de insulina.		
Pruebas	<ul style="list-style-type: none"><li>Se utilizará un vial de 1000 UI de insulina hasta que queden 25 y se comprobará si se avisa al usuario por el controlador.</li></ul>		

#### § Modificación de datos

ID	11	Fecha	14/10/2022
Título	Comunicación sistema-controles		
Tipo	Funcional / Modificación de datos		
Descripción	El sistema deberá de responder ante la pulsación del botón de brillo. Se podrá subir o bajar tanto a través de la aplicación como del controlador.		
Pruebas	<ul style="list-style-type: none"><li>Comprobar que al pulsar los botones se modifique el brillo</li></ul>		

ID	12	Fecha	14/10/2022
Título	Comunicación sistema-controles		
Tipo	Funcional / Modificación de datos		



Descripción	El sistema deberá de responder ante la pulsación del botón de volumen. Se podrá aumentar o disminuir el volumen tanto a través de la aplicación como del controlador.
Pruebas	<ul style="list-style-type: none"> <li>Comprobar que al pulsar los botones se modifique el volumen</li> </ul>

ID	13	Fecha	17/10/2022
Título	Sincronización sistema-controles-aplicación		
Tipo	Funcional / Modificación de datos		
Descripción	El sistema deberá de inyectar insulina cuando el botón + sea pulsado. Cada pulsación a través de la app o del controlador será equivalente a una dosis.		
Pruebas	<ul style="list-style-type: none"> <li>Comprobar que al pulsar los botones se inyecte la insulina, según las pulsaciones dadas por el usuario.</li> </ul>		

ID	14	Fecha	14/10/2022
Título	Modificación de tiempo entre pruebas.		
Tipo	Funcional / Modificación de datos		
Descripción	Los usuarios expertos podrán modificar el intervalo de tiempos en los que se realizan las pruebas de nivel de azúcar en sangre.		
Pruebas	<ul style="list-style-type: none"> <li>Se comprobará si se puede modificar el tiempo entre pruebas.</li> <li>Se comprobará que, al cambiar el tiempo entre pruebas, las pruebas se realizan en el intervalo determinado.</li> </ul>		

ID	15	Fecha	14/10/2022
Título	Modificación de dosis.		
Tipo	Funcional / Modificación de datos		
Descripción	Los usuarios expertos podrán modificar las dosis de insulina administrada.		
Pruebas	<ul style="list-style-type: none"><li>• Se comprobará si se pueden modificar las dosis de insulina.</li><li>• Se comprobará que, las dosis de insulina, se administra la dosis que se ha determinado.</li></ul>		

ID	16	Fecha	17/10/2022
Título	Introducción de recomendaciones e indicaciones.		
Tipo	Funcional / Modificación de datos		
Descripción	Los especialistas o el centro médico asociado podrán dejar recomendaciones e indicaciones tras haber accedido al historial de tratamientos del paciente.		
Pruebas	<ul style="list-style-type: none"><li>• Añadir recomendaciones con un usuario experto en una cuenta de simulación.</li><li>• Comprobar que la información que se añade, quede registrada en la aplicación.</li></ul>		

### ○ 3.4. Requisitos no funcionales

#### § Seguridad

ID	17	Fecha	16/10/2022
Título	Amenazas externas		
Tipo	No funcional/ Seguridad		

Descripción	El sistema proporcionará un elevado nivel de asistencia monitorizada. Si se realiza una pulsación no esperada por el sistema, como puede ser una cantidad de insulina excesiva en el modo manual, el sistema no realizará ningún cambio.
Pruebas	<ul style="list-style-type: none"> <li>● Introducir más insulina de la debida, para comprobar que el sistema lo detecta como acción peligrosa y no realizará nada.</li> </ul>

ID	18	Fecha	16/10/2022
Título	Amenazas externas		
Tipo	No funcional/ Seguridad		
Descripción	Todas las partes del sistema realizan autodiagnósticos periódicos alertando de la situación de posibles mal funcionamiento. Alertará tanto al usuario como al centro médico asociado.		
Pruebas	<ul style="list-style-type: none"> <li>● Realizar autodiagnósticos con un paciente que se encuentra con problemas renales debida a la diabetes.</li> <li>● Comprobar que las alertas sean producidas en caso de riesgo.</li> </ul>		

ID	19	Fecha	16/10/2022
Título	Fallos del sistema		
Tipo	No funcional/ Seguridad		
Descripción	En el caso de que un fallo sea detectado en el controlador, este avisará al usuario.		
Pruebas	<ul style="list-style-type: none"> <li>● Comprobar en caso de caída del sistema, este al minuto enviará una notificación y avisará al usuario por el controlador con una vibración prolongada.</li> </ul>		

ID	20	Fecha	17/10/2022
Título	Inicio de sesión.		

Tipo	Funcional / Modificación de datos
Descripción	<ul style="list-style-type: none"> <li>• Para poder entrar en la aplicación o web cualquier usuario se deberá de registrar con un usuario y una contraseña. De esta forma el sistema será capaz de distinguir los distintos usuarios.</li> </ul>
Pruebas	<ul style="list-style-type: none"> <li>• Introducir varios perfiles de distintos usuarios y comprobar que permite el acceso correspondiente.</li> <li>• Comprobar que el sistema registra el perfil y permita acceso a las funcionalidades del usuario.</li> <li>• Intentar entrar a otros perfiles y comprobar que no es posible.</li> </ul>

ID	21	Fecha	17/10/2022
Título	Distancia mínima.		
Tipo	Funcional / Modificación de datos		
Descripción	Si el controlador se aleja a más de 2 metros del teléfono móvil en el que esté registrado el usuario, no aceptará ninguna instrucción de la aplicación.		
Pruebas	<ul style="list-style-type: none"> <li>• Comprobar si alejando el teléfono móvil el controlador deja de recibir instrucciones de la aplicación.</li> </ul>		

#### § Consumo de recursos

ID	22	Fecha	19/10/2022
Título	Dosis de insulina		
Tipo	No funcional/ Consumo de recursos		

Descripción	<p>La insulina se deberá administrar, controlado por pulsos, con una bomba miniaturizada a través de una aguja de la siguiente manera:</p> <table> <tr> <th><i>Glucemia (mg/dL)</i></th><th><i>Insulina regular (UI)</i></th></tr> <tr> <td>&lt; 150</td><td>0</td></tr> <tr> <td>150-200</td><td>2</td></tr> <tr> <td>200-250</td><td>4</td></tr> <tr> <td>250-300</td><td>6</td></tr> <tr> <td>&gt; 300</td><td>8</td></tr> </table>	<i>Glucemia (mg/dL)</i>	<i>Insulina regular (UI)</i>	< 150	0	150-200	2	200-250	4	250-300	6	> 300	8
<i>Glucemia (mg/dL)</i>	<i>Insulina regular (UI)</i>												
< 150	0												
150-200	2												
200-250	4												
250-300	6												
> 300	8												
Pruebas	<ul style="list-style-type: none"> <li>• Se pondrán diferentes muestras de sangre en el sistema y se verá si la dosis administrada es la correcta.</li> <li>• Se comprobará que la aguja se mantiene colocada durante el uso del usuario.</li> <li>• Se medirá que cada pulso sea 1 UI.</li> </ul>												

#### § Fiabilidad

ID	23	Fecha	16/10/2022
Título	Frecuencia		
Tipo	No funcional/ Fiabilidad		
Descripción	Ajusta la administración de insulina automáticamente y corrige las subidas y bajadas cada 30 minutos.		
Pruebas	<ul style="list-style-type: none"> <li>• Comprobar el nivel de insulina cada cinco minutos.</li> <li>• Comprobar el registro del tratamiento durante un día entero. Se tendrá en cuenta las subidas y bajadas de la glucosa en sangre y como el sistema responde ante ello.</li> </ul>		

ID	24	Fecha	16/10/2022
Título	Disponibilidad		
Tipo	No funcional/ Fiabilidad		

Descripción	Las funciones del controlador deberán estar disponibles durante todos los días del año.
Pruebas	<ul style="list-style-type: none"><li>● Comprobar que se cumple la garantía de acceso.</li><li>● Mantener el sistema en funcionamiento durante veinticuatro horas.</li></ul>

#### § Rendimiento

ID	25	Fecha	16/10/2022
Título	Velocidad		
Tipo	No funcional/ Rendimiento		
Descripción	El tiempo de respuesta del sistema no deberá de ser menor que dos segundos.		
Pruebas	<ul style="list-style-type: none"><li>● Calcular el tiempo que tarda en reaccionar tras pulsar cada botón.</li><li>● Calcular el tiempo para cargar el historial de tratamientos.</li></ul>		

#### § Manejo de errores

ID	26	Fecha	16/10/2022
Título	Notificación con sonido.		
Tipo	No funcional/ Manejo de errores		
Descripción	Cuando el controlador detecta niveles peligrosos de azúcar en sangre, activará una alerta sonora en la aplicación.		
Pruebas	<ul style="list-style-type: none"><li>● Realizar pruebas para asegurar que cuando el usuario se encuentre con niveles de azúcar en sangre menores de 70 mg/dL y mayores que 200 mg/dL, se produzca esta notificación.</li></ul>		

ID	27	Fecha	16/10/2022
Título	Llamada al servicio sanitario.		
Tipo	No funcional/ Manejo de errores		
Descripción	Cuando el controlador detecte niveles menores de 70 mg/dL o mayores de 200 mg/dL de azúcar en sangre y no se haya producido ningún cambio en un minuto, se realizará una llamada al centro médico a través de la aplicación.		
Pruebas	<ul style="list-style-type: none"><li>● Comprobar que, si el usuario no responde ante la notificación con sonido, un minuto después se producirá la llamada.</li><li>● Tras haber introducido la insulina suficiente, el paciente sigue en riesgo, verificar que la llamada se produce.</li></ul>		

#### § Privacidad

ID	28	Fecha	16/10/2022
Título	Limitación de uso de datos		
Tipo	No funcional/ Privacidad		
Descripción	El diagnóstico de tratamiento y el historial, almacenados en la base de datos, sólo estarán disponibles para el propio usuario y el centro médico correspondiente.		
Pruebas	<ul style="list-style-type: none"><li>● Comprobar que los datos del paciente son inaccesibles; información cifrada.</li><li>● Comprobar que el acceso a la información no es posible sin autenticación</li></ul>		

### 3.5 Vocabulario del dominio

#### 3.5.1 Vocabulario del modelo conceptual

---

**Servidor:** es un programa que ofrece un servicio especial que otros programas denominados clientes pueden usar a nivel local o a través de una red.

**Datos:** Información sobre algo concreto que permite su conocimiento exacto o sirve para deducir las consecuencias derivadas de un hecho. Ejemplo: *A este problema le faltan datos numéricos.*

**Microsensor:** Dispositivo o instrumento de tamaño relativamente pequeño, que tienen el objetivo de medir alguna variable dentro del sistema en el cual se encuentra.

**Perfil:** Contienen la información que el sistema necesita para permitir a los usuarios iniciar una sesión en el sistema, para acceder a su propia sesión personalizada.

**Especialista:** Profesional de la salud con los conocimientos y habilidades necesarias para desarrollar actividades de tratamiento o cuidado de individuos.

**Pulsación:** Incrementos o disminuciones rápidos y rítmicos de la presión del medio.

**Almacenamiento:** Acción y efecto de almacenar.

**Llamada:** Es el mecanismo usado por una aplicación para solicitar un servicio de ayuda.

**Notificación:** alertas que emiten ciertos programas o servicios para advertir algo al usuario.

**Historial:** Reseña circunstanciada de los antecedentes de algo o de alguien.

**Web:** un subconjunto de Internet que consta de las páginas a las que se puede acceder mediante un navegador web.

**Botón:** Pieza que se pulsa o se hace girar para activar, interrumpir o regular el funcionamiento de un aparato o de un mecanismo, especialmente eléctrico.

**Paciente:** Dicho de un sujeto que recibe o padece la acción del agente.

**Alerta:** Aviso para que se extremen las precauciones o se incremente la vigilancia.

**Usuario:** Individuo que utiliza de manera habitual un sistema o servicio determinado.

**Cartucho:** Frasco pequeño destinado a contener un medicamento inyectable, del cual se van extrayendo las dosis convenientes.

**Bomba:** pequeño aparato que se lleva puesto y que algunas personas con diabetes usan para recibir insulina



### 3.5.2 Vocabulario técnico

---

**Comunicación sistema-aplicación:** Interacción que se produce entre el sistema y la aplicación telefónica que permite un funcionamiento adecuado del producto general.

**Cálculo:** Cómputo que se hace de algo por medio de operaciones matemáticas.

**Muestra:** Parte representativa de un conjunto que se usa para ser analizada.

**Autenticación:** Acción y efecto de autenticar.

**Manualmente:** De manera manual o con las manos.

**Diferencia:** Cualidad o accidente por el cual algo se distingue de otra cosa.

**Glucosa:** Aldohexosa de seis átomos de carbono, que constituye un sólido blanco, muy soluble en agua, de sabor muy dulce y presente en muchos frutos maduros.

**Interfaz:** Conexión, física o lógica, entre una computadora y el usuario, un dispositivo periférico o un enlace de comunicaciones.

**Modificación:** Cambio que por influencia del medio se produce en los caracteres anatómicos o fisiológicos de un ser vivo y que no se transmite por herencia a los descendientes.

**Dosis:** Cantidad o porción de algo, material o inmaterial.

**Disolución:** Mezcla que resulta de disolver una sustancia sólida en un líquido.

**Concentración:** Magnitud que expresa la cantidad de una sustancia por unidad de volumen.

**Glucómetros:** Instrumentos de medida que se utilizan para obtener la concentración de glucosa en sangre.

**Glucemia:** Medida de concentración de glucosa libre en la sangre, suero o plasma sanguíneo.

**Insulina:** Hormona segregada por los islotes de Langerhans en el páncreas, que regula la cantidad de glucosa existente en la sangre.

**Fuga:** Salida accidental de gas o de líquido por un orificio o una abertura producidos en su contenedor.

**Intervalo:** Conjunto de los valores que toma una magnitud entre dos límites dados.

**Rendimiento:** Proporción entre el producto y el resultado obtenido en base a los medios utilizados.

**Registro:** conjunto de datos que han sido obtenidos.

**Asistencia monitorizada:** Ayudar en base a la observación mediante aparatos especiales el curso de uno o varios parámetros fisiológicos o de otra naturaleza para detectar posibles anomalías.

**Autodiagnóstico:** Prueba que se realiza una persona o entidad para comprobar su propio estado o funcionamiento.

**Hiperglucemia:** Nivel de glucosa en la sangre superior al normal.

**Hipoglucemia:** Nivel de glucosa en la sangre inferior al normal.

**Información cifrada:** Transcribir en letras o símbolos, de acuerdo con una clave, un mensaje o texto cuyo contenido se quiere proteger.

Unidades de medida:

**μ:** micro.

**mg:** miligramo.

**dL:** decilitro.

**UI:** unidad internacional. En este contexto, corresponde a 0,035 mg de insulina humana anhidra.

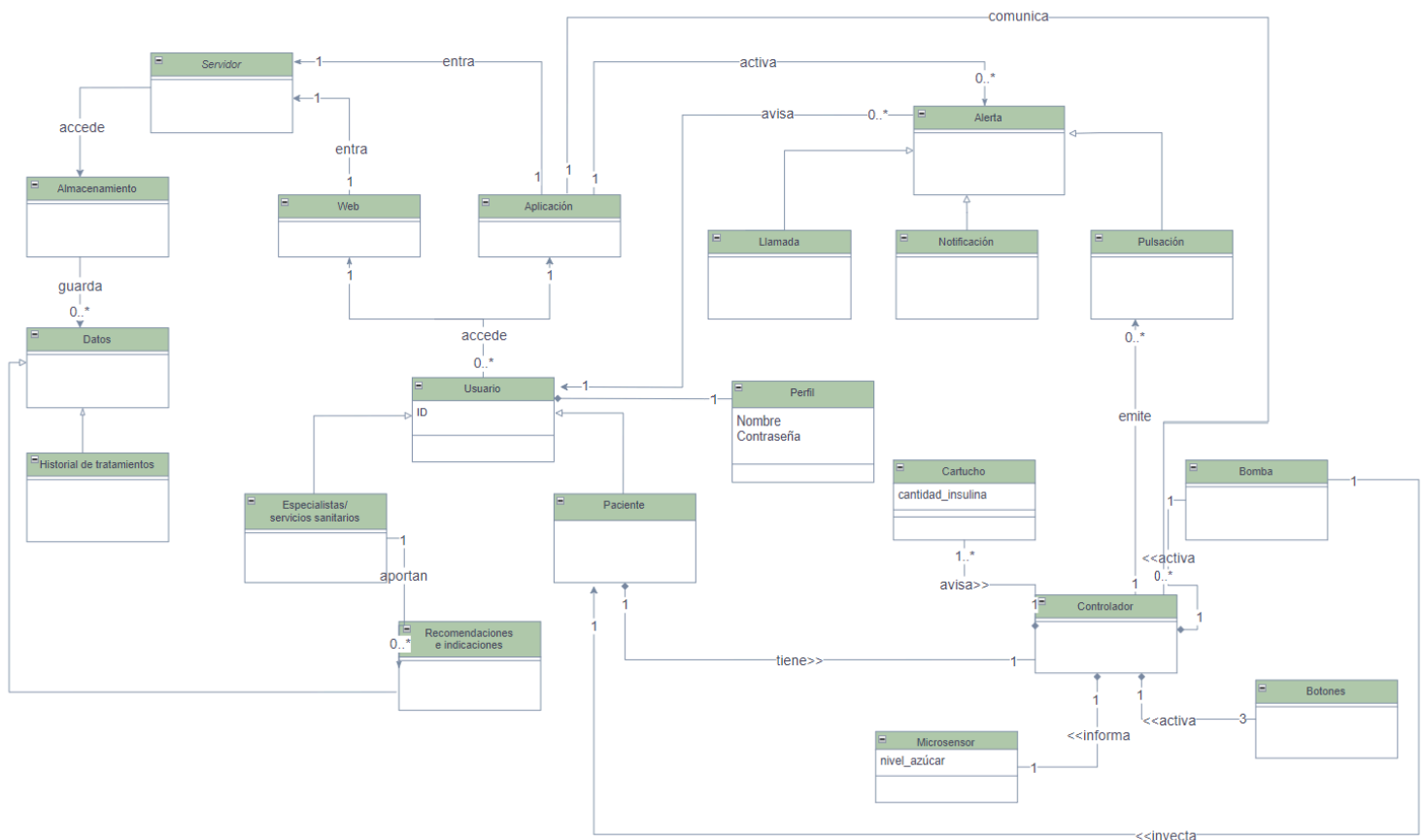
**mL:** mililitro.

**cm:** centímetro.

# Arquitectura

## Modelo conceptual

El modelo conceptual se basa en los distintos requisitos y funcionalidades del sistema. Teniendo en cuenta que nuestro sistema se divide en tres subsistemas; el controlador, la aplicación y el servidor, se ha estructurado las clases que forman cada componente y su relación entre ellas y entre los subsistemas, creando un modelo conceptual a partir de su unión.



El servidor, accede a su almacenamiento donde se encuentran guardados los datos (0...\*); el historial de tratamientos y las recomendaciones e indicaciones. Las recomendaciones e indicaciones solo las puede mandar un usuario que sea especialista o servicio sanitario, que al igual que los usuarios pacientes pueden acceder al servidor y consultar estos datos a través de la web y de la aplicación.

La aplicación se comunica con el controlador y con el servidor, a través del servidor puede acceder a los datos y con el controlador, los datos que la aplicación aporta son accesibles gracias a esta conexión. La aplicación se comunica con 0...\* controladores ya que tiene que aportar los datos de todos los usuarios que pueden ser o bien pacientes o bien especialistas. Para que cada usuario pueda entrar a la aplicación se necesita un perfil, de este modo la app diferencia los distintos usuarios. La aplicación en caso de peligro activa una alerta; o llamadas o notificaciones o si no hay peligro ninguna, por ello 0..\*, y alerta al usuario.

Por último, el controlador que está formado por tres clases (de ahí la composición); bomba, cartucho y microsensor. Estas tres clases son las que permiten al controlador realizar sus funcionalidades. El cartucho avisa al controlador cuando se está quedando vacío, y mandará o no (0...\*), una pulsación al usuario. El controlador activa la bomba en caso de que haya que inyectar insulina al paciente, el cual lleva el controlador (composición). El microsensor recoge los datos y es el que informa al controlador el estado del paciente. Todas estas clases excepto cartuchos, ya que hay más de uno, tiene multiplicidad 1 porque un controlador solo tiene una bomba y un microsensor. Finalmente, se comunica con la aplicación por si alguna instrucción es realizada, el controlador deberá hacerla.

## **Modelo de arquitectura**

Debido a que el sistema general lo hemos dividido en 3 subsistemas, hemos creado 3 arquitecturas diferentes según las funciones que desempeña cada subsistema.

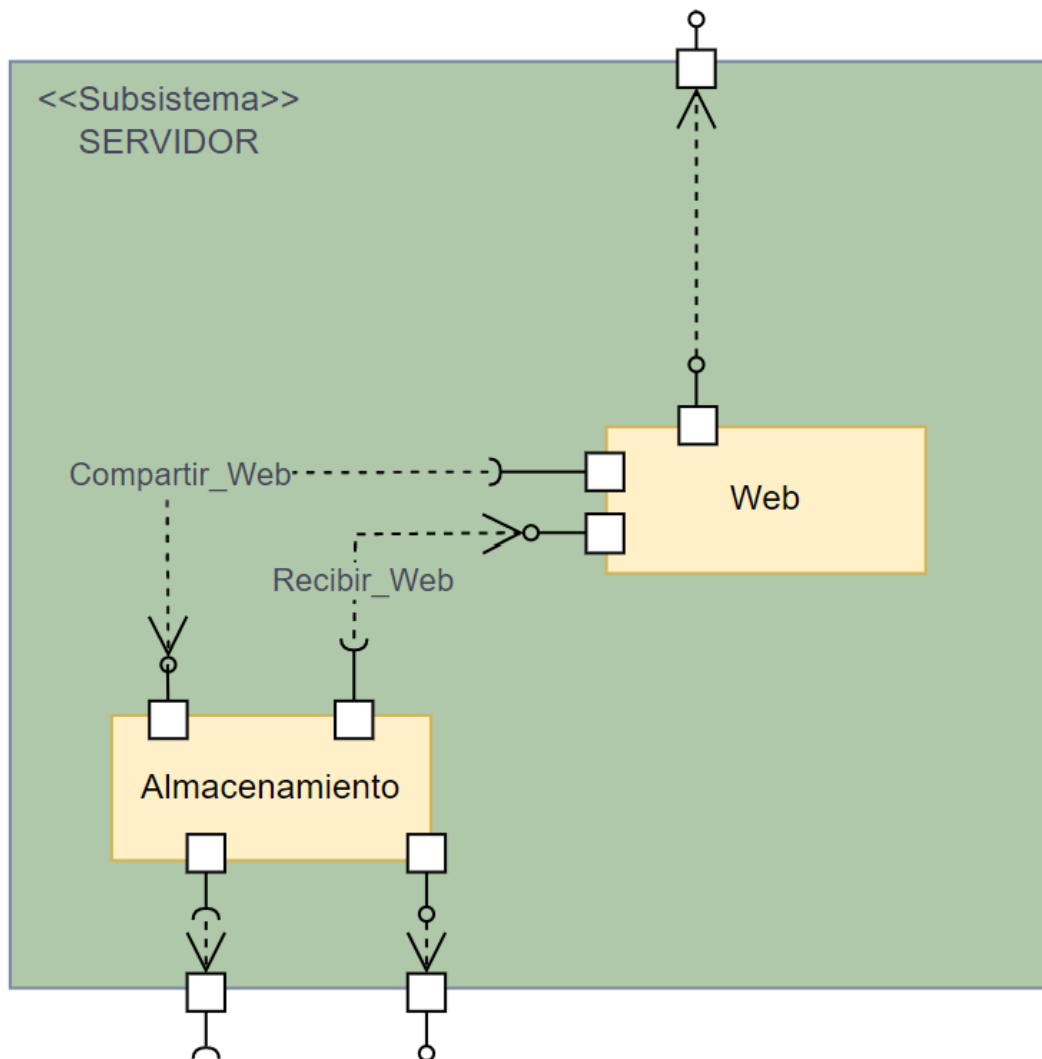
Por un lado, hemos determinado que la arquitectura del servidor. El servidor tiene una arquitectura en forma de cebolla, teniendo en el centro el almacenamiento con el historial y las listas de pacientes de cada especialista. Este centro está rodeado completamente por una página web que le aporta seguridad ya que los usuarios que se conecten a ella tendrán que registrarse con un nombre de usuario y contraseña. Gracias a la arquitectura en forma de cebolla, es más complicado que entren desde el exterior hasta el centro de la cebolla, aportando seguridad a los datos del sistema.

El interior del servidor está formado por el almacenamiento, que tiene un sistema de almacenamiento como el de una memoria de un ordenador convencional. Para almacenar los datos, los separa en historiales y listas de pacientes. Las listas de pacientes son listas creadas a partir de listas simplemente enlazadas, de tal manera que

el especialista obtendría la lista de pacientes completa únicamente conociendo a uno de sus pacientes. Gracias a este tipo de almacenamiento, es posible ahorrar espacio en los discos duros por los que se forma el almacenamiento. Sin embargo, los historiales tienen un espacio delimitado cada uno y, cuando se necesita almacenar un dato nuevo en el historial y este está completo, se sobrescribe la zona más antigua del espacio del historial, pudiendo almacenar una cierta cantidad de días de tratamiento.

También aparece la página web, a la que los usuarios tienen acceso. Los usuarios están diferenciados porque tienen labores diferentes, el especialista/centro sanitario es quien tiene la lista de pacientes con el historial de cada uno y puede proporcionar recomendaciones, mientras que el usuario sólo puede acceder al historial y puede leer recomendaciones de su especialista. La web se encarga de comprobar que el usuario sea válido y de solicitar el historial, lista de pacientes o recomendaciones, acorde al usuario, al almacenamiento del servidor.

Ofrece cuatro interfaces, una que necesita información de la aplicación, otra que ofrece información a la aplicación y las otras dos sirven para que los usuarios accedan al servidor a través de la página web.

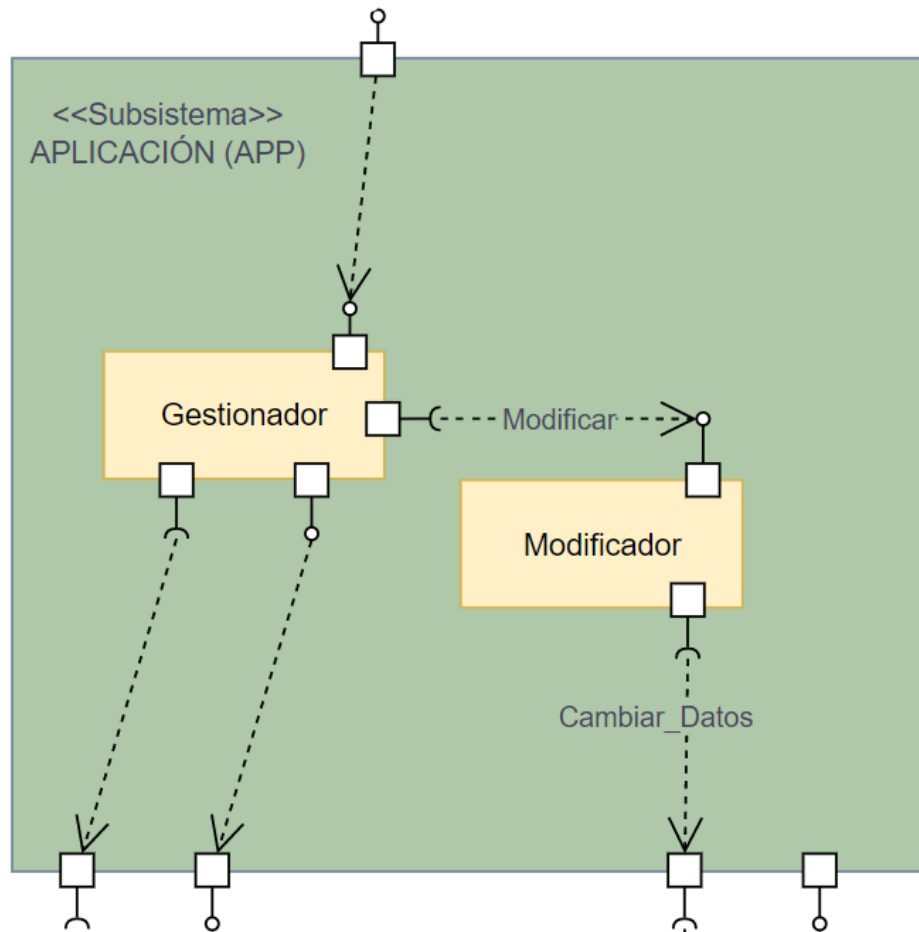


Por otro lado tenemos la aplicación. Esta sirve básicamente de conexión entre el controlador y el servidor, pero en ella se puede ver información tanto del controlador como del servidor.

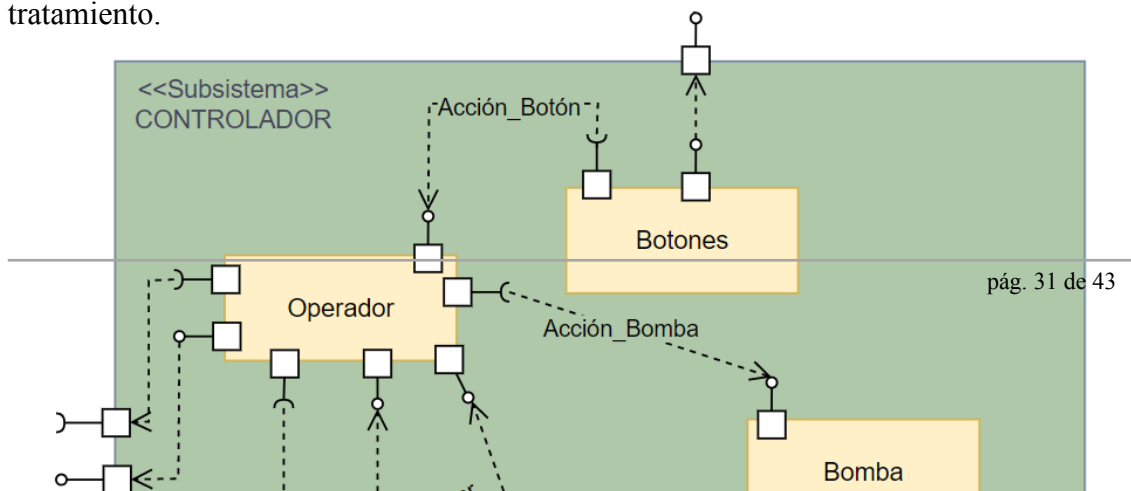
Del controlador se puede ver el tiempo hasta la próxima medición de azúcar en sangre, el valor de la última medición, la cantidad de insulina que queda en los cartuchos y el nivel de brillo, a su vez, se puede modificar la insulina por dosis necesaria para el paciente, el brillo de la pantalla del controlador ó inyectar directamente la insulina en modo manual.

Del servidor recibe el historial de tratamientos y las recomendaciones del especialista/centro sanitario. Para que el servidor cree la base de datos, la aplicación

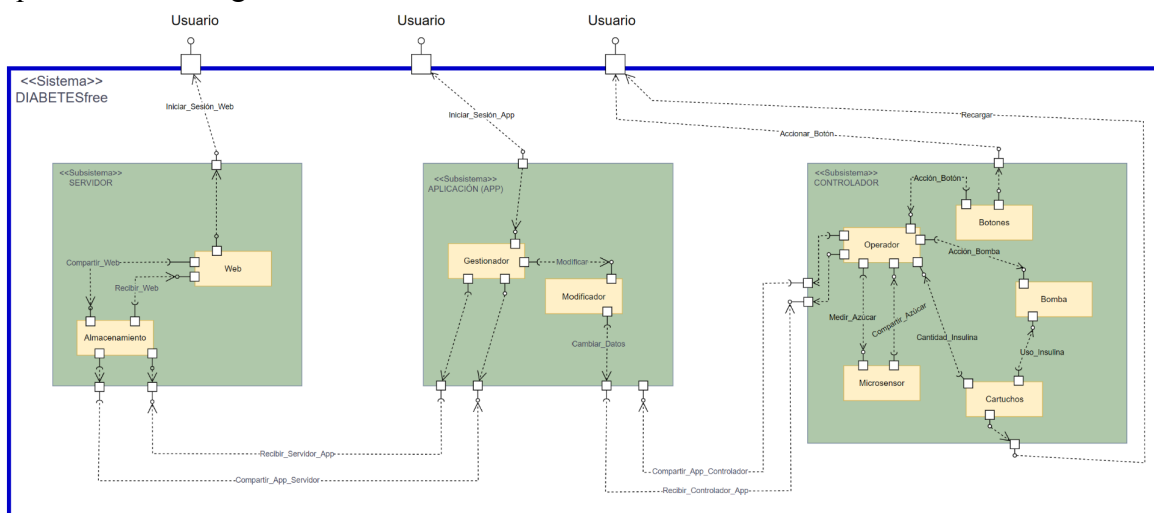
comparte con el servidor la información sobre la última medición y la cantidad de insulina inyectada como resultado, y el servidor se encarga de gestionar esa información. Por todo lo que realiza la aplicación, se podría decir que es un microservicio que apoya al resto del sistema.



Finalmente tenemos el controlador. La arquitectura del ordenador podemos considerarla monolítica, ya que este tipo de arquitectura puede soportar las labores que tiene que realizar el controlador. Estas labores son las explicadas en los requisitos y se pueden resumir en: medir el nivel de azúcar en sangre del paciente e inyectar insulina en base al azúcar calculado, mostrar datos por una pantalla en el controlador y compartir datos con la aplicación para que el paciente pueda mantener un seguimiento efectivo de su tratamiento.



Gracias a estos tres subsistemas, se puede hacer la vista general de nuestro sistema, quedando de la siguiente manera:



componente, siendo a la vez breve y significativo. Si es necesario utilizar abreviaturas, deben utilizarse coherentemente y sin ambigüedad, y deben quedar documentadas. Los componentes deberán tener identificadores mutuamente consistentes: por ejemplo, si un componente se llama Publicador\_Registros, entonces es posible que exista uno que se



llame `Editor_Registros`, pero no `Edición_Registros`, ni `Editor-Registros`, ni `EditorRegistros`.

El elemento esencial de la especificación de cada componente son sus interfaces, que obviamente deben coincidir perfectamente con las interfaces identificadas en el modelo de implementación (sección 4.2).

## **Componente A (repetir para cada componente)**

### **Componente Servidor**

**Tipo:** No ejecutable.

**Propósito [Función, Procesamiento, Datos, Recursos]:** El componente servidor es una agrupación de componentes. Sirve para que el usuario tenga acceso a sus datos(historial de tratamientos y recomendaciones e indicaciones) y se queden guardados.

**Dependencias (interfaces requeridas):** Necesita la interfaz de *Inicio\_Sesión\_Web*, se encuentra conectada a la interfaz procedente de Usuario. Se utiliza para que cuando el Usuario quiera acceder al servidor para que pueda tener acceso a todos sus datos deberá de registrarse, de manera que se podrá distinguir al usuario dentro del servidor. Como no sólo se puede acceder desde una web sino que también desde la aplicación requiere la interfaz de *Recibir\_Servidor\_App*, en la que el controlador recibe los datos a través de la app. Como precondition el cliente deberá utilizar una aplicación como modo de acceso, teniendo que iniciar sesión tanto por la web como la app.

**Interfaces:** La interfaz, *Compartir\_App\_Servidor* sirve para el traspaso de datos entre el servidor y la aplicación.

### **Componente Web**

**Tipo:** No ejecutable.

**Propósito [Función, Procesamiento, Datos, Recursos]:** La web es el medio por el cual los usuarios pueden acceder a sus datos. Es un componente propio del servidor.

**Dependencias (interfaces requeridas):** Requiere la interfaz del servidor del *Inicio\_Sesión\_Web*, de este modo la web podrá distinguir al usuario a través del inicio de sesión que el usuario proporciona al servidor y mostrará los datos correspondientes del usuario a través de la web. También requiere de la interfaz *Recibir\_Web*, en la que recibirá los datos guardados en almacenamiento para poder mostrarlas a través de este componente al usuario.

**Interfaces:** La interfaz que proporciona este componente es la de *Compartir\_Web*, para poder acceder al Almacenamiento(componente del servidor), se necesita compartir los datos web, por ello accede al almacenamiento según los datos del usuario web. Como precondition necesita que el cliente haya iniciado sesión para poder acceder a su almacenamiento. Como postcondición deberá de cumplir con su función de mostrar los datos web de cada usuario.

### **Componente Almacenamiento**

**Tipo:** Ejecutable.

**Propósito [Función, Procesamiento, Datos, Recursos]:** El almacenamiento es un componente que se encuentra dentro del servidor. Su propósito es almacenar todos los datos del usuario, dentro del servidor.

**Dependencias (interfaces requeridas):** Requiere la interfaz de *Compartir\_Web*, ya explicado en el componente Web. También requiere de la interfaz *Recibir\_App\_Servidor* es una interfaz que se conecta a través de los componentes aplicación y servidor, y el componente servidor le proporciona los datos compartidos de la aplicación al almacenamiento para que se pueda extraer los datos guardados en el.

**Interfaces:** Proporciona la interfaz de *Recibir\_Web*, en la que proporciona los datos en almacenamiento a la web. También proporciona la interfaz *Compartir\_App\_Servidor*, la interfaz está conectada con el servidor en la que le proporciona al servidor los datos almacenados del usuario que después serán recibidos en la aplicación. Como precondition el usuario no tiene ninguna obligación ya que si utiliza el sistema por primera vez no tendrá ningún tipo de dato que almacenar. Como postcondición todos los datos que lleguen a través del servidor o la web deberán quedar almacenados.

### **Componente Aplicación**

**Tipo:** No ejecutable.

**Propósito [Función, Procesamiento, Datos, Recursos]:** El componente aplicación es una agrupación de componentes. Su propósito es que el usuario pueda acceder a la información o realizar instrucciones como son la de aumentar el brillo o el modo manual a través de este componente que facilita el uso del sistema.

**Dependencias (interfaces requeridas):** El componente aplicación requiere de la interfaz *Iniciar\_Sesión\_App*, el usuario deberá de iniciar sesión(precondición) y creando un perfil en la propia aplicación de modo que la aplicación podrá diferenciar a los usuarios. La aplicación proporciona *Compartir\_App\_Servidor*, le comparte al servidor el perfil del usuario del cual queremos conocer sus datos. *Compartir\_App\_Controlador*;

la app se comparte con el controlador. Esto sirve por si el usuario indica una acción a través de la app para que la aplicación comparta esta acción al controlador.

**Interfaces:** La interfaz *Recibir\_Servidor\_App* en la que el servidor le proporciona la información de la aplicación y la recibe. Por último, requiere de la interfaz *Recibir\_Controlador\_App* necesita recibir todos los datos del controlador para que la app contenga toda la información que el controlador le proporciona. La precondition es que el cliente deberá de presionar algún botón para que el controlador reciba dicha acción. La postcondición es que el controlador realice la instrucción dada a través de la aplicación.

### **Componente Gestionador**

**Tipo:** No ejecutable.

**Propósito [Función, Procesamiento, Datos, Recursos]:** El componente gestionador se encargará de gestionar los distintos perfiles que se crean al iniciar sesión a través de la aplicación por el usuario.

**Dependencias (interfaces requeridas):** *Compartir\_App\_Servidor*, en la que requiere de la aplicación poder compartir los datos obtenidos del servidor al usuario. Requiere de la interfaz *Inicio\_Sesión\_App*, en la que necesita que el usuario haya iniciado sesión para poder gestionar su perfil.

**Interfaces:** El gestionador proporciona a la aplicación a través de la interfaz *Recibir\_Servidor\_App* la información que necesita la aplicación para que el servidor la reciba tras gestionar el usuario. *Modificar*, en la que proporciona al Modificador el perfil del usuario que hay que cambiar. Como precondition necesita que el usuario inicie sesión y como postcondición que sepa distinguir los distintos usuarios y gestionarlos.

### **Componente Modificador**

**Tipo:** Ejecutable.

**Propósito [Función, Procesamiento, Datos, Recursos]:** El propósito de este componente es poder realizar a través de la app alguna modificación.

**Dependencias (interfaces requeridas):** El Modificador requiere de la interfaz *Modificar* los datos procedentes de un usuario específico proveniente del Gestionador para poder modificarlos.

**Interfaces:** La interfaz *Cambiar\_Datos* proporciona a la aplicación los datos que han sido modificados. Como precondition necesita que el cliente realice un cambio desde la aplicación y como postcondición que los datos sean modificados.

### **Componente Controlador**

**Tipo:** No ejecutable.

**Propósito [Función, Procesamiento, Datos, Recursos]:** El componente controlador es una agrupación de componentes. Sirve para medir el nivel de azúcar en sangre, inyectar insulina en el usuario y para pasar información sobre el tratamiento a la aplicación. También es posible utilizarlo de manera manual con los botones que tiene.

**Dependencias (interfaces requeridas):** Necesita la interfaz *Recibir\_Controlador\_App*, para recibir la información necesaria de la aplicación para realizar su función, *Accionar\_Botón*, que necesita que el usuario accione botones para realizar diferentes funciones, y la interfaz *Recargar*, para recargar los cartuchos de insulina cuando se terminan.

**Interfaces:** Tiene una interfaz que sirve para el traspaso de información entre el controlador y la aplicación, *Compartir\_App\_Controlador*. Como precondition el usuario es el responsable de la recarga de cartuchos y como postcondición el controlador deberá de realizar todas sus funciones y mandar la información a la app.

### **Componente Operador**

**Tipo:** Ejecutable.

**Propósito [Función, Procesamiento, Datos, Recursos]:** El componente operador se encarga de gestionar todos los componentes dentro del controlador y hacer que funcione de manera adecuada. Además, calcula la cantidad de insulina que hay que inyectar en el usuario teniendo en cuenta el nivel de azúcar en sangre y de alertar si es necesario a la aplicación.

**Dependencias (interfaces requeridas):** Necesita las interfaces *Recibir\_Controlador\_App*, *Compartir\_Azúcar*, *Cantidad\_Insulina* y *Acción\_Botón*. *Recibir\_Controlador\_App* se encarga de recibir información de la aplicación, cómo inyectar insulina ó modificar la dosis según el nivel de azúcar. *Compartir\_Azúcar* se encarga de recibir el nivel de azúcar en sangre. *Cantidad\_Insulina* se encarga de recibir la cantidad de insulina disponible en los cartuchos. *Acción\_Botón* se encarga de recibir las pulsaciones de los botones cuando estas se realizan.

**Interfaces:** El Operador ofrece 3 interfaces: *Compartir\_App\_Controlador*, *Medir\_Azúcar* y *Acción\_Bomba*. La primera se encarga de compartir la información del controlador a la aplicación. *Medir\_Azúcar* se encarga de accionar el microsensor para que mida el nivel de azúcar en sangre del usuario. *Acción\_Bomba* se ocupa de accionar la bomba para inyectar la insulina necesaria. La precondition es que el cliente lleve el controlador para que el operador pueda realizar sus cálculos. La postcondición sería que las operaciones que esta componente realiza deben de ser las correctas a partir de los datos recibidos y que tendrá que mandar.

### **Componente Botones**

**Tipo:** No ejecutable.

**Propósito [Función, Procesamiento, Datos, Recursos]:** El componente Botones sirve para poder inyectar insulina en modo manual desde el controlador y para modificar el brillo de la pantalla.

**Dependencias (interfaces requeridas):** Necesita la interfaz *Accionar\_Botón*, que sirve para que el usuario pueda activar los botones del controlador.

**Interfaces:** Proporciona la interfaz *Acción\_Botón*, que envía al Operador el botón que ha sido pulsado para que el Operador gestione esa información. La postcondición será que se realice la acción correspondiente con el botón pulsado.

### **Componente Microsensor**

**Tipo:** No ejecutable.

**Propósito [Función, Procesamiento, Datos, Recursos]:** El componente Microsensor sirve para medir el nivel de azúcar en sangre y enviar esa información al Operador.

**Dependencias (interfaces requeridas):** Necesita la interfaz *Medir\_Azúcar*, que recibe una orden desde el Operador para hacer una medición de azúcar en sangre en el usuario.

**Interfaces:** Proporciona una única interfaz, *Compartir\_Azúcar*, que se encarga de enviar la información de la medición de azúcar al Operador. La precondition será que el cliente lleve el dispositivo. La postcondición es que deberá de medir el nivel de azúcar del usuario y compartirlo.

### **Componente Cartuchos**

**Tipo:** Ejecutable.

**Propósito [Función, Procesamiento, Datos, Recursos]:** Los Cartuchos se encargan de mantener la insulina y tenerla preparada para su uso. Es ejecutable ya que el usuario debe recargar la insulina cuando ésta se acabe.

**Dependencias (interfaces requeridas):** Necesita la interfaz *Recargar* para que el usuario recargue los cartuchos de insulina cuando estos se terminen.

**Interfaces:** Ofrece dos interfaces *Cantidad\_Insulina* y *Uso\_Insulina*. *Cantidad\_Insulina* comparte la cantidad de insulina en los cartuchos al Operador y *Uso\_Insulina* se encarga de mantener la insulina disponible para que la bomba la utilice cuando tenga que inyectar. La precondition es que los cartuchos serán recargados por el cliente. La postcondición será que los cartuchos tengan la capacidad de guardar la cantidad de insulina prometida al cliente.

### **Componente Bomba**

**Tipo:** No ejecutable.

**Propósito [Función, Procesamiento, Datos, Recursos]:** El componente Microsensor sirve para medir el nivel de azúcar en sangre y enviar esa información al Operador.

**Dependencias (interfaces requeridas):** Necesita las interfaces *Uso\_Insulina* y *Acción\_Bomba*.

**Interfaces:** La postcondición será que deberá de inyectar insulina(accionar), cuando el operador lo indique dependiendo del nivel de azúcar del usuario.

## 5 LISTAS Y TABLAS

### 5.1 Consistencia entre requisitos: conflictos, redundancias, acoplamientos

	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17	R18	R19	R20	R21	R22	R23	R24	R25
R1																									
R2																									
R3																									
R4									+																
R5																									
R6								+																	
R7																									
R8						+																			
R9				+																					
R10																									
R11																									
R12																									
R13																									
R14																									
R15																									
R16																									
R17																									
R18																									
R19																									
R20																									
R21																									
R22																									
R23																						+			
R24																									
R25																									

### 5.2 Trazabilidad requisitos - modelo conceptual (clases)

Vamos a ver cómo se relacionan los requisitos funcionales de nuestra aplicación con los elementos más importantes que tenemos en nuestros diagramas.

	RF 01	RF 02	RF 03	RF 04	RF 05	RF 06	RF 07	RF 08
Bomba	+		+				+	+
Cartucho	+			+		+		
Microsensor	+			+			+	

Datos		+	+	+	+		+	
Perfil	+				+	+		+

	RF 09	RF 10	RF 11	RF 12	RF 13	RF 14	RF 15	RF 16
Bomba								
Cartucho	+		+			+		
Microsensor	+							
Datos		+	+	+	+		+	
Perfil	+		+		+			+

### 5.3 Trazabilidad requisitos – modelo de implementación (componentes)

	RF 01	RF 02	RF 03	RF 04	RF 05	RF 06	RF 07	RF 08
Servidor			+	+	+	+		
Botones	+			+		+		+
Almacenamiento								
Aplicación		+	+	+	+		+	
Gestionador			+					+
Modificador	+				+		+	
Controlador	+		+					+



Operador	+		+		+		+	
----------	---	--	---	--	---	--	---	--

	RF 09	RF 10	RF 11	RF 12	RF 13	RF 14	RF 15	RF 16
Servidor		+	+				+	
Botones	+		+			+		
Almacenamiento								+
Aplicación		+	+	+	+		+	
Gestionador	+	+						+
Modificador	+					+		
Controlador			+			+		+
Operador	+				+		+	

## Anexo 1

Tras la primera entrega de este proyecto, se recibió la siguiente corrección por parte de nuestros compañeros del grupo 4:

### 1. Aspectos positivos que aporta el documento

El documento está estructurado de forma correcta, siguiendo el índice establecido por la plantilla. El contenido es correcto en general, usa un formato válido de IEEE y las referencias usadas en el documento son correctas, indicando las fuentes de información utilizadas en el documento. Describe de una forma clara el producto que presenta y los productos a lo que sustituye están referenciados adecuadamente.

### 2. Consideraciones generales sobre el formato del documento

Desde el punto de vista ortográfico no hemos encontrado ningún error relevante. Cabe destacar que está redactado en lenguaje coloquial en lugar de lenguaje formal como es habitual en este tipo de documentos.

La presentación del documento es adecuada y la información es de calidad y está expuesta correctamente. En cambio, la organización del documento sería mejorable, deberían organizar correctamente las tablas de requisitos porque en ocasiones se cortan a la mitad.

La legibilidad del documento es buena con carácter general, pero existen una serie de apartados en los que hace falta una segunda lectura para la correcta comprensión del mensaje. Por ejemplo, en la página 4 el punto 1.1 incide dos veces en lo mismo de forma innecesaria o en la página 7 punto 2.4 presenta dos frases que parecen contradictorias “El sistema no podrá presentar ningún error” y “En caso de que algún error sea detectado”.

### **3. Consideraciones individuales sobre la adecuación al índice de la práctica.**

#### **3.1. Estructura del documento: ¿sigue los apartados del índice de la práctica?**

El documento sigue el orden establecido por el índice de la plantilla, no hemos encontrado descripciones en apartados erróneos o fuera de lugar. Tampoco se han encontrado alteraciones que añadan o quiten algún apartado del índice.

#### **3.2. Contenido del documento: ¿se corresponde cada uno de los apartados con la finalidad que el índice propone?**

Si se corresponde el contenido de los apartados con la finalidad que el índice propone, a excepción de la interfaz en el 2.6. Es necesario revisar si en esta sección entra en mucho detalle con los diferentes botones. En esta sección no se trata de detallar completamente la interfaz, eso será una tarea posterior en diseño, se trata de dar una idea general sobre el tipo de software a desarrollar.

Además hay un error en el contenido, se indica que el documento es “una guía común para todas las partes involucradas con carácter contractual”, lo cual es incorrecto, el carácter del documento es más bien informativo dirigido a todas las partes implicadas en el proyecto incluyendo clientes y desarrolladores.

En cuanto a la plantilla de requisitos, se podría añadir un apartado en el que se indique el ESTADO del requisito. A pesar de que en esta primera entrega todos los requisitos son “Propuestos”, en etapas posteriores del proyecto unos requisitos estarán en un estado más avanzado que otros.

#### **3.3. Explicar con más detalle los errores principales: inconsistencias en los requisitos, incoherencias entre requisitos y modelos, diseño de la solución mal explicado, etc.**

La mayoría de los errores se encuentran en la redacción de los diferentes apartados del documento: por ejemplo, el apartado 1.1 está estructurado en forma de lista, con frases excesivamente sencillas.

Algunas restricciones generales no explican la existencia de ningún requisito como deberían, como por ejemplo el peso máximo.

En el apartado 2.5 donde explica las características de los usuarios hay un aspecto a mejorar. Deberían fusionar o aclarar qué capacidades diferentes tienen los usuarios expertos y los usuarios ocasionales. Además no queda claro cuál es la función del usuario empresa. Por lo demás la tabla de roles es correcta. En la segunda tabla, las expectativas de los usuarios expertos y ocasionales son demasiado elevadas.

Cuando hacen referencia a la interacción entre sistema no mencionan al sistema informático de los servicios de emergencia, al centro médico o especialista.

En el tercer párrafo del apartado 2.6, donde hablan del entorno operacional, habla de tres tipos de alertas pero solo hace mención a una de ellas.

En el requisito 13 no funcional se debe mejorar el lenguaje de la prueba para evitar ambigüedad en el plan de pruebas. En el requisito 21, se debería actualizar el conjunto de pruebas para no solo calcular el tiempo de respuesta del sistema, si no comprobar que este es el adecuado en cada escenario propuesto.

La tabla de consistencia entre requisitos indica acoplamiento de los requisitos 4 y 9 (que tratan sobre la sincronización del sistema con distintos elementos) con el 6, que trata sobre la disponibilidad de insulina, temas demasiado separados entre sí como para ser un acoplamiento

En esta segunda versión de nuestro documento hemos resuelto varias de las correcciones propuestas en el texto anterior. Por ejemplo, todos los requisitos que hablaban sobre el tamaño y peso del sistema.

A su vez hemos separado el sistema en tres partes, lo que hace que se resuelvan la mayoría de las correcciones, como el acoplamiento de los requisitos, que es menor ya que cada uno de los 3 sistemas tiene sus propios requisitos que hacen posible que el sistema sea coherente y funcional.