

# SISTEMA DE RECOMENDACIÓN DE TRATAMIENTOS PARA PACIENTES DIABÉTICOS

Rafael Buzón Urbano

GRADO EN INGENIERÍA INFORMÁTICA  
UNIVERSIDAD COMPLUTENSE DE MADRID



TRABAJO DE FIN DE GRADO

2016-2017

Directores: Juan Antonio Recio García, Belén Díaz Agudo

# AUTORIZACIÓN DE DIFUSIÓN Y UTILIZACIÓN

Yo, Rafael Buzón Urbano, alumno matriculado en la asignatura Trabajo de Fin de Grado(TFG) en la Facultad de Informática de la Universidad Complutense de Madrid durante el curso 2016/2017, dirigido por Juan Antonio Recio García y Belén Díaz Agudo, autorizo la difusión y utilización con fines académicos, no comerciales y mencionando expresamente al autor del contenido de esta memoria, el código, la documentación adicional y la aplicación desarrollada.

*Rafael Buzón Urbano*

*A mi familia y amigos, en especial a mi padre Rafael Vladimiro Buzón Tapiador y a mi hermana Natalia Buzón Urbano.*

*Y en memoria de mi madre, María de la Sierra Urbano Benito, todo el esfuerzo y el trabajo de este proyecto va dedicado a ti, que me sigues cuidando y dando fuerzas desde el cielo.*

# ÍNDICE

<b>INDICE DE FIGURAS .....</b>	<b>V</b>
<b>RESUMEN .....</b>	<b>VII</b>
<b>ABSTRACT .....</b>	<b>VIII</b>
<b>CAPÍTULO 1. INTRODUCCIÓN .....</b>	<b>1</b>
1.1 INTRODUCCIÓN .....	1
1.2 OBJETIVOS .....	2
1.3 ORGANIZACIÓN DE LA MEMORIA .....	3
<b>CAPÍTULO 2. ESTADO DEL ARTE .....</b>	<b>4</b>
2.1 APLICACIONES MÉDICAS.....	4
2.1.1 <i>glUCModel</i> .....	4
2.1.2 <i>OnTrack Diabetes</i> .....	5
2.1.3 <i>GMate</i> .....	8
2.1.4 <i>Accu-Chek Connect Diabetes Management App</i> .....	10
2.2 IMPLANTACIÓN Y USO DE APLICACIONES MÉDICAS .....	13
2.3 SISTEMAS RECOMENDADORES.....	14
2.3.1 <i>Recomendadores colaborativos</i> .....	14
2.3.2 <i>Recomendadores basados en casos</i> .....	15
2.3.3 <i>Recomendadores basados en contenido</i> .....	15
2.3.4 <i>Recomendadores híbridos</i> .....	16
<b>CAPÍTULO 3. FUNCIONALIDAD .....</b>	<b>17</b>
3.1 FUNCIONES DE LA APLICACIÓN .....	17
3.2 FLUJO DE LA APLICACIÓN .....	18
3.3 PROTOTIPOS INICIALES .....	20
3.4 INTERFAZ DE MÉDICO.....	25
3.4.1 <i>Interfaz de inicio de sesión</i> .....	25
3.4.2 <i>Interfaz página de inicio</i> .....	27
3.4.3 <i>Interfaz gestión de pacientes</i> .....	31
3.4.3.1 <i>Listado de pacientes</i> .....	31
3.4.3.2 <i>Dar de alta a un nuevo paciente</i> .....	33
3.4.4 <i>Interfaz recomendador del paciente</i> .....	35

3.4.5 Interfaz perfil del paciente .....	40
3.4.6 Interfaz perfil médico .....	43
<b>CAPÍTULO 4. IMPLEMENTACIÓN .....</b>	<b>44</b>
4.1 ARQUITECTURA .....	44
4.1.1 Interacción entre front-end y back-end .....	45
4.2 IMPLEMENTACIÓN DEL FRONT-END .....	47
4.2.1 Entorno de desarrollo .....	47
4.2.2 Utilización de Bootstrap .....	48
4.2.3 Otras tecnologías del front-end .....	49
4.2.3.1 jQuery y AJAX .....	49
4.2.3.2 jQuery UI y modals.....	50
4.3 IMPLEMENTACIÓN DEL BACK-END .....	51
4.3.1 Arquitectura de la aplicación .....	52
4.3.2 Servicios web .....	55
4.3.3 Sistema recomendador .....	56
4.3.4 Implementación de la seguridad .....	58
4.3.4.1 Seguridad de las cuentas de usuario .....	58
4.3.4.2 Seguridad en el envío de datos .....	58
4.3.4.3 Seguridad de acceso a secciones privadas .....	58
4.3.4.4 Seguridad en acceso a la base de datos .....	58
4.3.5 Base de datos .....	59
<b>CAPÍTULO 5. CONCLUSIONES Y TRABAJO FUTURO.....</b>	<b>60</b>
5.1 CONCLUSIONES .....	60
5.2 CONCLUSIONS .....	62
5.3 TRABAJO FUTURO .....	64
<b>REFERENCIAS .....</b>	<b>66</b>

# ÍNDICE DE FIGURAS

Figura 1: Menú principal OnTrack diabetes.....	6
Figura 2: Interfaz de registro de glucosa OnTrack diabetes.....	6
Figura 3: Gráfica de glucosa OnTrack diabetes.....	7
Figura 4: Histórico de actividad OnTrack diabetes.....	7
Figura 5: Uso de GMate.....	8
Figura 6: Tiras de ensayo y punzón GMate.....	9
Figura 7: Medidor Aviva Connect.....	11
Figura 8: Interfaz medición glucosa Accu-Check app.....	11
Figura 9: Interfaz timeline de actividad.....	12
Figura 10: Interfaz recomendación de bolo Accu-Check app.....	12
Figura 11: Flujo de la aplicación.....	19
Figura 12: Prototipos a mano iniciales de login y registro de pacientes.....	20
Figura 13: Prototipos a mano del menú de paciente y versión móvil del menú.....	20
Figura 14: Prototipo de login.....	21
Figura 15: Prototipo pestañas y sección de recomendador.....	21
Figura 16: Prototipo de tarjeta de recomendación automática.....	22
Figura 17: Prototipo recomendación enviada.....	22
Figura 18: Prototipo de listado de pacientes.....	23
Figura 19: Prototipo perfil de paciente.....	23
Figura 20: Prototipos de recomendador de paciente.....	24
Figura 21: Prototipo tarjeta de recomendación manual.....	24
Figura 22: Interfaz inicio de sesión versión escritorio.....	25
Figura 23: Interfaz inicio de sesión versión móvil.....	25
Figura 24: Diagrama de actividad de inicio de sesión.....	26
Figura 25: Interfaz página de inicio versión escritorio.....	27
Figura 26: Campos habilitados y deshabilitados página de inicio.....	28
Figura 27: Listado de soluciones página de inicio.....	28
Figura 28: Vista previa tarjeta de recomendación.....	29
Figura 29: Vista completa de tarjeta de recomendación.....	29

Figura 30: Interfaz página de inicio versión móvil.....	30
Figura 31: Diagrama de actividad página de inicio.....	30
Figura 32: Listado de pacientes.....	31
Figura 33: Resultados búsqueda de paciente.....	32
Figura 34: Resultados búsqueda de paciente.....	32
Figura 35: Diagrama de actividad listado de pacientes.....	33
Figura 36: Formulario de alta nuevo paciente.....	33
Figura 37: Diagrama de actividad formulario de alta nuevo paciente.....	34
Figura 38: Interfaz gestión de pacientes versión móvil.....	34
Figura 39: Interfaz recomendador paciente versión escritorio.....	35
Figura 40: Envío de tarjetas de recomendaciones mediante arrastre.....	36
Figura 41: Vista completa tarjeta recomendación sugerida.....	37
Figura 42: Formulario recomendación manual.....	38
Figura 43: Recomendación manual añadida con éxito.....	38
Figura 44: Diagrama de actividad recomendador paciente.....	39
Figura 45: Interfaz recomendador paciente versión móvil.....	39
Figura 46: Interfaz perfil paciente versión escritorio.....	40
Figura 47: Carrusel gráficas perfil paciente.....	41
Figura 48: Carrusel gráficas perfil paciente.....	41
Figura 49: Carrusel gráficas perfil paciente.....	41
Figura 50: Diagrama de actividad perfil paciente.....	42
Figura 51: Interfaz perfil paciente versión móvil.....	42
Figura 52: Interfaz perfil médico versión escritorio.....	43
Figura 53: Interfaz perfil médico versión móvil.....	43
Figura 54: Arquitectura de la aplicación glUCModel.....	44
Figura 55: Funcionamiento llamadas Java Servlets.....	52
Figura 56: Estructura de los Java Servlets.....	53
Figura 57: Ejemplo de petición AJAX.....	53

## RESUMEN

La diabetes es una enfermedad que en la actualidad afecta en España a un 13,8% del total de la población, lo que equivale a más de 5.3 millones de personas. De ellos, casi 3 millones estaban diagnosticados, pero 2.3 millones (43% del total) desconocían que padecían esta enfermedad.

Con el desarrollo de la aplicación web glUCModel 2.0 se ha buscado el objetivo de usar la tecnología en beneficio de la salud, facilitando a los médicos la labor del tratamiento de esta enfermedad, apoyándose para ello en dos pilares.

El primero de ellos es la facilidad de uso de la aplicación, ya que la misma se ha desarrollado buscando un diseño intuitivo y siguiendo los principios de diseño adaptativo para posibilitar su uso en distintos dispositivos.

El segundo y no menos importante es el uso de un recomendador que sirva como apoyo a la siempre necesaria intervención del médico. Este se nutre de recomendaciones ya realizadas, sugiriendo nuevos tratamientos adaptados a cada paciente.

A continuación, pasa a describirse como ha ido evolucionando el proyecto desde sus primeras fases hasta la finalización del mismo.

**Palabras clave:** glUCModel 2.0, enfermedad, diabetes, sistema recomendador

## ABSTRACT

Diabetes is a disease that currently affects 13.8% of the total population in Spain, which is equivalent to more than 5.3 million people. Of these, almost 3 million were diagnosed, but 2.3 million (43% of the total) were unaware that they had this disease.

With the development of the glUCModel 2.0 web application, the objective of using technology for the benefit of health has been sought, making it easier for doctors to treat this disease, supported by two pillars.

The first one is ease of use of the application, since it has been developed looking for an intuitive design and following the principles of adaptative design to enable its use in different devices.

The second and not least is the use of a recommender that serves as support for the always necessary intervention of the doctor. This is based on recommendations already made, suggesting new treatments adapted to each patient.

It then goes on to describe how the project has evolved from its earliest stages until its completion.

**Keywords:** glUCModel 2.0, disease, diabetes, recommender system

# Capítulo 1. INTRODUCCIÓN A GLUCMODEL

## 1.1 Introducción

La tecnología ha revolucionado en la última década la manera en la interactuamos con nuestros amigos, familiares, entorno y en definitiva con el mundo. Una parte muy importante de nuestro mundo se refiere a la salud.

Multitud de tecnologías y dispositivos han centrado sus esfuerzos en ayudarnos a mejorar y controlar nuestra salud, ya sea a la hora de monitorizar nuestra actividad física con pulseras o relojes inteligentes (*smartwatches*), o de facilitarnos el seguimiento de tratamientos si padecemos alguna enfermedad mediante las mediciones de varios tipos de sensores, combinados con el uso de aplicaciones que tenemos utilizar en nuestros dispositivos como *smartphone* o *tablets*.

El objetivo de este TFG es el desarrollo de una versión nueva de la aplicación glUCModel, desarrollada originalmente por el grupo ABSys del departamento de Arquitectura de Computadores y Automática de la UCM. [1][2]. La nueva versión, glUCModel 2.0, es una aplicación web que responde a las necesidades que necesitan los pacientes diabéticos aportando mejoras y un diseño moderno respecto a la versión anterior.

En la actualidad la diabetes es una enfermedad que afecta a 387 millones de personas a nivel global, situándose en el caso de España en 5.3 millones de personas (13.8% de la población), teniendo una estimación de 592 millones de afectados en el año 2035. Más allá de lo abultado de las cifras, destaca un dato concreto: más del 40% de las personas que sufren esta enfermedad no están diagnosticadas. [3]

Gracias a glUCModel 2.0 se facilita el seguimiento y el tratamiento de la enfermedad, apostando por una aplicación web de uso intuitivo y desarrollada siguiendo los principios de diseño adaptativo, permitiendo así su uso tanto en sistemas de escritorio como en dispositivos móviles.

Además de la importancia de contar con una aplicación que nos permita un uso cómodo e intuitivo, es fundamental que el sistema facilite la labor que tiene que desempeñar el médico.

Esto se lleva a cabo mediante el uso de un sistema recomendador, presentando al médico de forma automática posibles tratamientos para cada paciente, basándose en similitudes con el resto de pacientes registrados en el sistema.

glUCModel 2.0 es una aplicación pensada tanto para pacientes como para los médicos encargados de los mismos, aunque debido a la extensión de desarrollar el proyecto completo, siendo además un TFG de realización individual, se ha desarrollado la parte de la aplicación perteneciente al médico, quedando la aplicación del lado del paciente para un posible futuro trabajo.

## 1.2 Objetivos

El objetivo general de este TFG es el desarrollo de la versión mejorada de la aplicación web glUCModel, glUCModel 2.0, la cual facilitará la labor del médico a la hora de poder realizar tratamientos para pacientes diabéticos.

Este proyecto tiene una serie de objetivos:

- Estudio del estado del arte.
- Mejora del diseño de la aplicación original glUCModel mediante el uso de tecnologías web modernas.
- Incluir nuevas funcionalidades no presentes en la versión original, como la implementación de un sistema recomendador funcional.
- Implementación completa de todas las funcionalidades de glUCModel 2.0 para su uso en casos reales.

Para ello se llevan a cabo una serie de tareas que se describen a continuación:

- Estudiar las necesidades y requisitos de la aplicación.
- Investigar sobre los distintos sistemas recomendadores y ver cual se adapta mejor al proyecto.
- Realizar bocetos iniciales de la interfaz de la aplicación.
- Elegir los lenguajes de programación que se usarán en el proyecto tanto en la parte *front-end* como en el *back-end*.
- Estudiar y desarrollar la arquitectura más adecuada para el proyecto.
- Desarrollar de forma adaptativa la interfaz de la aplicación.
- Integrar las tecnologías usadas en el *front-end* y *back-end* con el recomendador para dar forma a la aplicación que usará el médico.

- Evaluar la validez de la aplicación haciendo uso de datos médicos de pacientes reales.
- Publicar el proyecto en un servidor online de la facultad para darle la mayor visibilidad posible.

### 1.3 Organización de la memoria

La memoria está organizada de la siguiente manera:

- En el capítulo 1 se presenta una introducción que explica el contexto y las motivaciones que han llevado a la realización de este proyecto, así como los tareas y objetivos perseguidos.
- En el capítulo 2 se hace un análisis de otras aplicaciones médicas que realicen funcionalidades similares, el uso y la implantación de las aplicaciones médicas a día de hoy y se realiza un pequeño estudio sobre los sistemas recomendadores.
- En el capítulo 3 se detalla el funcionamiento de la aplicación, pasando desde los primeros prototipos hasta la versión final de la aplicación. Esta sección se centra en explicar el uso y la interacción con las distintas interfaces.
- En el capítulo 4 se detalla la estructura e implementación del *front-end* y el *back-end* de la aplicación, así como del sistema recomendador.
- En el capítulo 5 se recogen el nivel de cumplimiento de las tareas y objetivos propuestos y los problemas encontrados durante el desarrollo. Además, se recogen posibles cambios, mejoras y nuevas funcionalidades que se pudiesen añadir en una futura versión de la aplicación.

## Capítulo 2. ESTADO DEL ARTE

En este capítulo se describe qué otras aplicaciones existen actualmente o han existido en el mercado y que realicen funcionalidades iguales o parecidas a las que se propone desarrollar en el proyecto glUCModel 2.0.

Así mismo, se expone el uso y la aceptación de las aplicaciones médicas a día de hoy a lo largo del planeta y se realiza un pequeño estudio sobre los sistemas recomendadores.

### 2.1 Aplicaciones médicas

A continuación, se exponen diferentes aplicaciones médicas, mostrando su interfaz con diferentes figuras y explicando su uso.

#### 2.1.1 glUCModel

glUCModel [1][2] es una aplicación web para el control de la diabetes crónica, desarrollada por el grupo ABSys del departamento de Arquitectura de Computadores y Automática de la UCM. En glUCModel se potencia la comunicación e interacción entre paciente y médico logrando así un mejor control del seguimiento del paciente y de los tratamientos que está llevando a cabo.

Esta aplicación posee varias características propias, la primera de ellas es un sistema recomendador que utiliza razonamiento basado en casos (case based reasoning), proporcionando una ayuda extra al médico.

También incluye un espacio de e-learning mediante Moodle, este espacio facilita a los pacientes un lugar donde poder consultar información sobre la diabetes.

Como última característica presenta una funcionalidad capaz de predecir los niveles de glucosa que tendrá un paciente en el futuro, basándose para ello en algoritmos de computación evolutiva.

### 2.1.2 OnTrack Diabetes

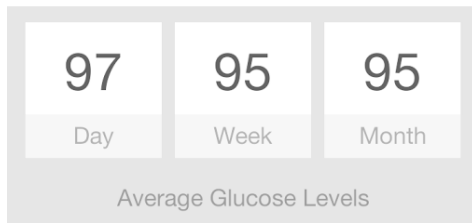
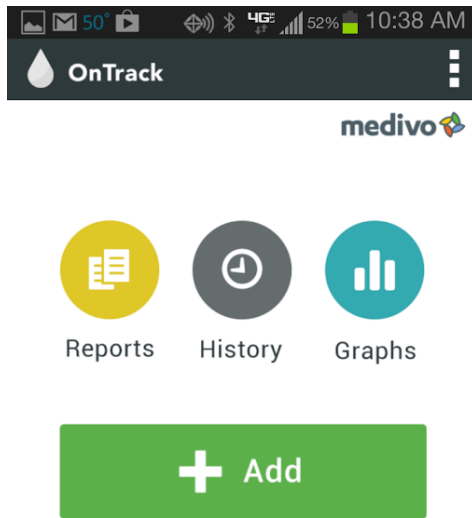
OnTrack Diabetes [4] es una aplicación desarrollada para *smartphones Android* que permite llevar un control de la glucosa que contiene nuestro cuerpo para poder tener un control de los niveles de glucosa a lo largo del día.

La aplicación permite a los usuarios el poder gestionar los datos relativos a la cantidad de glucosa en su sangre, su presión arterial o el pulso entre otros, y también llevar cuenta de los alimentos y medicamentos ingeridos. La aplicación permite añadir todo tipo de entradas a lo largo del día y etiquetarlas, para llevar un control exhaustivo. Además, se pueden añadir alertas para así recordar tomar la medicación o por ejemplo medir los niveles de glucosa dos horas después de la comida.

La aplicación permite realizar copias de seguridad de nuestros datos, dando la opción de exportarlos a archivos como CSV, HTML o XML y visualizar la evolución en gráficos para poder mostrar toda esta documentación a nuestro médico.

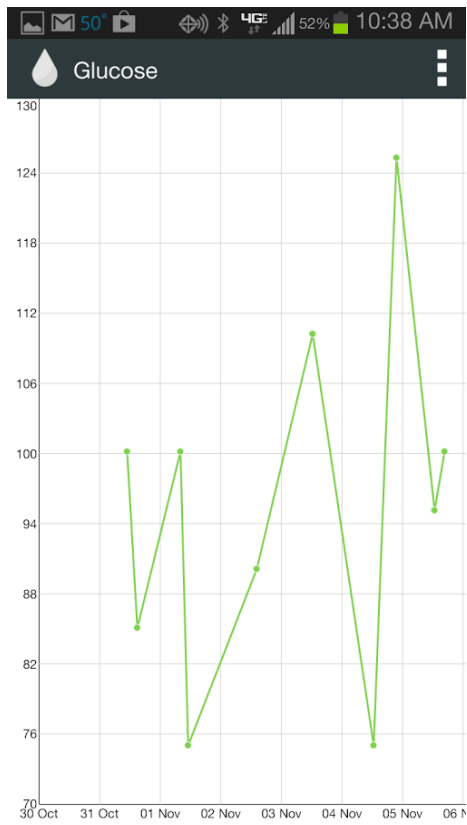
Se trata de una aplicación completamente gratuita, con un diseño sencillo y cómodo de usar. Actualmente la aplicación no se encuentra en castellano, estando solo disponible en inglés.

En las figuras 1 a la 4 se presentan capturas de la interfaz de la aplicación.



This screenshot shows the 'Add' entry form. The status bar at the top shows the time as 10:35 AM and the battery level at 54%. The form includes several input fields: a date field set to '11/05/2013', a time field set to '10:33 AM', a meal type dropdown menu currently showing 'Breakfast', and a reminder field. Below these is a text input field containing 'Granola bar'. A large section for 'Glucose' has a value of '95' and units of 'mg/dl'. Another section for 'Food' has a value of '25' and units of 'carbs'. At the bottom of this section is a button labeled '+ Add Another'. A large green button with a white checkmark and the text 'Save' is at the very bottom of the screen.

Figuras 1 y 2: Menú principal e interfaz de registro de glucosa OnTrack diabetes



Date	Activity	Glucose Value	Time
Thursday, November 14, 2013	Food	25.0	9:42am
	Glucose	125	7:39am
Wednesday, November 13, 2013	Glucose	250	2:40pm
	Glucose	75	9:41am
Tuesday, November 12, 2013	Glucose	253	2:40pm
	Glucose	108	10:41am

Figuras 3 y 4: Gráfica de glucosa e histórico de actividad OnTrack diabetes

### 2.1.3 GMate

El sistema de control de glucosa en sangre GMate SMART [5] es una *app* médica móvil disponible para *smartphones Android e iOS*.

Se compone por una parte de un medidor de glucosa en sangre pequeño y portátil que se conecta a la salida del auricular del *smartphone* y ofrece unos resultados rápidos y precisos (figura 5) y por otra parte de una *app*.

El fabricante es Philosys, una empresa surcoreana fabricante de aparatos médicos, cuyo lema es “*proporcionar aparatos y servicios médicos de diagnóstico, para que todos puedan llevar una vida saludable*”.



*Figura 5: Uso de GMate*

Su uso requiere adquirir una serie de dispositivos: un punzón Gmate SMART y las tiras Gmate para ensayos (figura 6), así como descargarse la aplicación.

Tras iniciar la *app*, se pide al usuario que introduzca el Gmate SMART y una tira. Después tiene que pincharse en la yema del dedo con el punzón y aplicar la sangre sobre la tira. El sistema analiza la muestra y presenta los resultados al paciente, teniendo la opción de poder enviárselos a su médico.



*Figura 6: Tiras de ensayo y punzón GMate*

### 2.1.4 Accu-Chek Connect Diabetes Management App

Este sistema se compone de dos piezas [6]:

- El medidor Accu-Chek Aviva Connect, un aparato médico que mide el nivel de glucosa en sangre de una muestra tomada en una tira y lo muestra en su pantalla.
- La Accu-Chek Connect Diabetes Management App, una *app* médica compatible con *smartphones*.

El fabricante describe la *app* como ayuda para el tratamiento de la diabetes. Esta *app* permite la descarga electrónica desde medidores de glucosa, la entrada manual de datos, almacenamiento, presentación, transmisión y autogestión de los datos de glucosa en sangre.

La parte diferenciadora y exclusiva de esta *app* es que está diseñada (y ya autorizada) para su uso como calculadora de insulina en el bolo, es decir, que facilita el cálculo de la insulina necesaria en cada comida.

Accu-Chek Connect Diabetes Management App fue la primera *app* médica móvil autorizada por la FDA norteamericana que ofrece recomendaciones sobre la dosis de insulina.



Figura 7 y 8: Medidor Aviva Connect e interfaz medición glucosa Accu-Check app

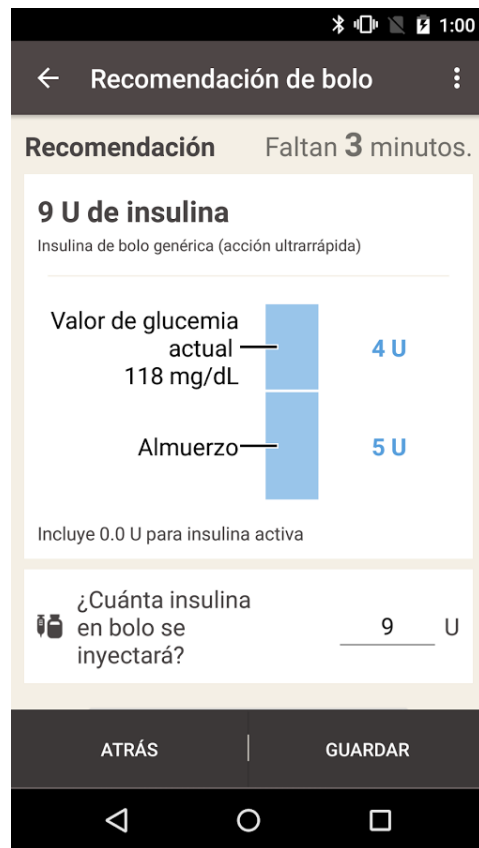
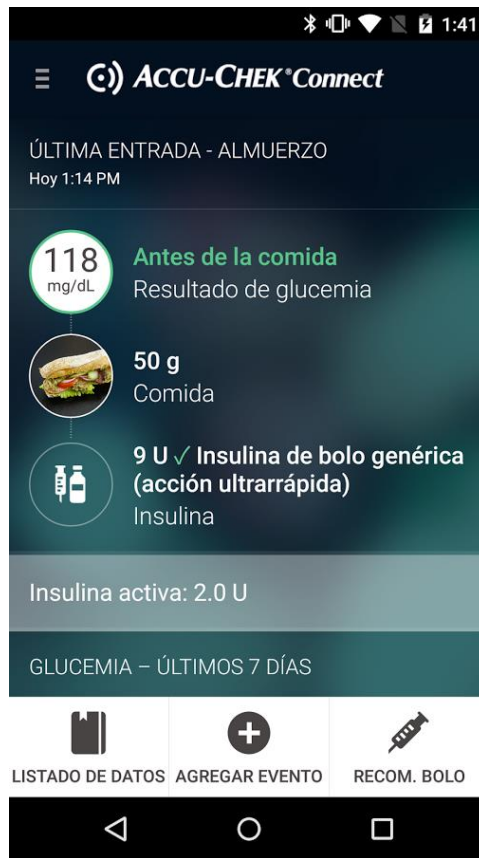


Figura 9 y 10: Interfaz timeline de actividad e interfaz recomendación de bolo Accu-Check app

## 2.2 Implantación y uso de aplicaciones médicas

El nexo de unión entre la tecnología, los pacientes y los médicos cada vez es mayor.

Una simple consulta en la red o sofisticados sistemas de monitorización (como la tensión o la respiración), que junto a información de otros usuarios forman un gran volumen de datos denominado *Big Data*, están llamados a revolucionar la investigación epidemiológica.

Una de cada 20 consultas en Google es sobre salud. Según una encuesta de iDoctus realizada en 2014 entre más de 7.400 facultativos de España, el 60% aseguraba que usa tres dispositivos para llegar a la información médica (ordenador, tableta y móvil) y casi el 95% accede por dos de ellos.

Tres de cada cuatro encuestados afirmaban que las aplicaciones médicas le ayudan a ahorrar tiempo, y el 88% concluía que incrementan la seguridad en el diagnóstico y en la prescripción. Casi la mitad de los médicos declaraba utilizar aplicaciones médicas delante de los pacientes, y el 45% creía necesario recomendárselas, siempre que cuenten con un aval científico-médico [8].

Según el último Informe Socialogues de IPSOS [9], realizado en 27 países de todo el mundo, entre los que se encuentra España, se ha obtenido que el 26% de los españoles utiliza aplicaciones de salud recomendadas por algún especialista. Esta cifra se debe en gran medida a la existencia de un gran número de apps médicas disponibles tanto para *Android* como para *iOS*.

Según indica Mercedes Abizanda [10], presidenta de la Sociedad Española de Médicos de Atención Primaria (Semergen) en Cataluña, *"siempre que venga de un entorno de prestigio, proveniente de la medicina basada en la evidencia, no tenemos ningún problema en recomendar a los pacientes que utilicen las apps. Todo lo contrario. Los médicos buscamos que los pacientes se hagan más responsables de su enfermedad. Estas herramientas son complementarias a lo que ya estamos trabajando, es decir, a un enfoque que potencie que el paciente esté formado e informado en su enfermedad. Es la medicina del futuro, aunque ya tiene mucho de presente"*.

Para la elaboración del informe de IPSOS, se han realizado entre 500 y 1.000 entrevistas en cada uno de los 27 países seleccionados, a adultos cuya edad estaba

entre los 18 y 64 años. [9]

El 49% de los indios y el 46% de los chinos reconocen que utilizan aplicaciones móviles de salud. En el caso contrario se encuentran países como Noruega (1%), Holanda (3%) o Suecia (5%), donde el uso de estas *apps* es testimonial. [9]

España encabeza el uso de estas aplicaciones con un 26%, situándose por delante de Francia (17%), Alemania (13%), Bélgica (12%) o Gran Bretaña (9%). [9]

## 2.3 Sistemas recomendadores

Dado el estado actual en la cantidad de datos que se pueden llegar a procesar sobre un campo o tema concreto, manejar de forma manual decenas de terabytes de información para sacar beneficios de la misma se antoja algo muy complicado, es aquí donde cobran importancia los sistemas recomendadores.

Un sistema recomendador es un sistema inteligente que proporciona información o sugerencias personalizadas al usuario sobre un tema en concreto (gustos de libros o películas, dietas y ejercicios personalizados, tratamientos médicos, etc.)

Dentro de los sistemas recomendadores podemos encontrar varios tipos [11] [12]

### 2.3.1 Recomendadores colaborativos

Los recomendadores colaborativos basan su funcionamiento en coleccionar y analizar gran cantidad de información sobre diferentes características del usuario. El sistema analiza estas características propias del usuario buscando otros usuarios similares a él, pudiendo en este caso predecir sugerencias sobre el campo en el que se esté trabajando.

Los recomendadores colaborativos asumen que si hemos tenido un buen grado de similitud con alguien en el pasado seguiremos teniéndolo en el futuro a la hora de dar sugerencias. Este tipo de sistemas presentan 3 problemas principales:

- Arranque en frío: este tipo de sistemas requieren una gran cantidad de datos para poder realizar recomendaciones precisas, siendo esto un problema cuando el sistema se encuentra vacío o en un estado inicial.
- Escalabilidad: En muchos de los entornos en los que se usan este tipo de sistemas podemos tener millones de productos y usuarios, siendo necesario un gran coste computacional para el cálculo de las recomendaciones.

- Escasez: Al ser usados en muchos sistemas de comercio electrónico como *Amazon*, nos encontramos con que hay un gran volumen de distintos elementos vendidos, dándose el caso en el que solo una pequeña parte de ellos han recibido valoraciones y por lo tanto solo una pequeña parte de estos elementos pasarán a ser tenidos en cuenta para las recomendaciones.

### 2.3.2 Recomendadores basados en casos

Los recomendadores basados en casos funcionan mediante la utilización de casos ya existentes en el sistema, estos casos tienen una descripción y una solución asociada.

El recomendador analiza los casos existentes, quedándose con los casos que presenten una similitud suficientemente buena, adaptándolos a la descripción del caso que lo requiere.

Uno de los puntos fuertes de este tipo de recomendador es que se evita el problema del arranque en frío, ya que no es necesaria una gran cantidad de información para poder sugerir recomendaciones de manera precisa. Por el contrario, uno de sus puntos débiles es la posibilidad de encontrarnos cuellos de botella a la hora de adquirir conocimientos (knowledge acquisition) debido a la necesidad de definir este conocimiento de manera explícita.

### 2.3.3 Recomendadores basados en contenido

Este tipo de recomendadores se basan en las descripciones de un objeto y las preferencias que encontramos en el perfil del usuario. Las palabras clave se usan para describir al objeto y el perfil del usuario se utiliza para indicar que tipo de objeto está dentro de las preferencias del usuario.

Varios objetos candidatos se comparan con otros objetos que previamente han sido valorados por el usuario, esto se hace para poder elegir cual es el objeto que mejores coincidencias presenta para poder ser recomendado.

El principal problema que presenta este tipo de recomendadores es el grado de capacidad que puedan alcanzar para aprender sobre las preferencias del usuario respecto a un solo tipo de fuente de información. Si el sistema recomendador está limitado al mismo tipo de contenido que el usuario ha consumido previamente, este estará muy limitado si quieren poder recomendarse objetos o contenidos de otros ámbitos.

### 2.3.4 Recomendadores híbridos

Los recomendadores híbridos combinan los 3 tipos de recomendadores anteriores para poder conseguir un nivel mayor de eficacia a la hora de sugerir recomendaciones.

Pueden darse recomendadores híbridos que generan recomendaciones por separado con cada uno de los tipos anteriores para posteriormente combinar los resultados o bien unificarlos a todos desde un principio. El rendimiento que puede alcanzarse en cuanto a efectividad en las sugerencias con estos sistemas puede llegar a ser superior que usando alguno de los sistemas anteriores.

Un ejemplo de recomendadores híbridos lo encontramos en la plataforma *Netflix*, la cual usa recomendadores colaborativos para comparar las búsquedas y las visualizaciones de los usuarios, así como recomendadores basados en contenido para sugerir al usuario series o películas similares a las que el usuario ha dado previamente una valoración alta.

## Capítulo 3: FUNCIONALIDAD

En este capítulo se describen todas las funcionalidades disponibles en la aplicación a modo de manual de usuario, mostrando mediante diagramas de actividad el flujo de estados de la aplicación.

### 3.1 Funciones de la Aplicación

Antes de entrar en detalle en el flujo de la aplicación y explicar detalladamente cada interfaz que posee la misma, se muestra cuáles son sus funcionalidades básicas.

La funcionalidad principal de glUCModel 2.0 es el recomendador personalizado de cada paciente, el cual automatiza la tarea de comparar la similitud con distintos pacientes facilitando al médico la tarea de proporcionar tratamientos.

Dentro del recomendador de cada paciente el médico tiene 3 columnas con las que puede interactuar: *Recomendaciones sugeridas*, *Recomendaciones manuales* y *Enviadas*, se habla con más detalle de esta sección en el análisis de la interfaz.

Fuera de esta sección, encontramos un recomendador general en el que se pueden realizar recomendaciones manuales para todos los pacientes asignados al médico que cumplan una serie de parámetros que se definen en un pequeño formulario.

También encontramos una sección para poder gestionar los pacientes de una forma fácil e intuitiva, pudiendo visualizar un listado desde el que se puede entrar en su perfil y a su recomendador personalizado.

Entrando en el perfil vemos información general del paciente, así como una serie de variables bajo seguimiento que le serán útiles al médico para el seguimiento del mismo. Lo último que vemos en esta sección es la posibilidad de dar de alta a un nuevo paciente mediante un pequeño formulario.

También tenemos la posibilidad de entrar en nuestro perfil como médico, viendo diversa información como el correo electrónico con el que estamos dados de alta o el número de colegiado.

Por último, tenemos las secciones de Gestión de documentos (no implementada en esta versión de la aplicación) y de Educación para la diabetes, la cual no está integrada directamente en la aplicación, ya que proporciona acceso a una plataforma Moodle de gIUCModel usada en versiones antiguas de esta aplicación.

### 3.2 Flujo de la Aplicación

La primera pantalla que se encuentra el médico es la de inicio sesión, teniendo que introducir su correo electrónico y contraseña con los que este dado de alta en el sistema.

Una vez validadas sus credenciales se encuentra en primer lugar con la página de inicio, el recomendador general, donde puede ver las recomendaciones de todos sus pacientes que aun estén pendientes de envío, así como un formulario con distintos requisitos para generar una recomendación nueva a todos sus pacientes que los cumplan.

En la parte superior de la pantalla tenemos acceso al resto de secciones de la aplicación, si estamos usando la aplicación desde un *smartphone* tenemos que pulsar sobre el menú de la parte superior derecha para desplegar el acceso al resto de secciones.

En la sección de *Gestión de Pacientes*, además de poder dar de alta a nuevos pacientes, si pinchamos sobre el nombre del paciente podremos acceder a su perfil médico, y pulsando sobre el botón *Ver recomendador* tendremos acceso a su recomendador personalizado.

Pulsando sobre el icono del médico en la parte superior derecha de la pantalla tendremos la opción de cerrar sesión.

En la figura 11 se muestra el flujo de interacción con la aplicación.

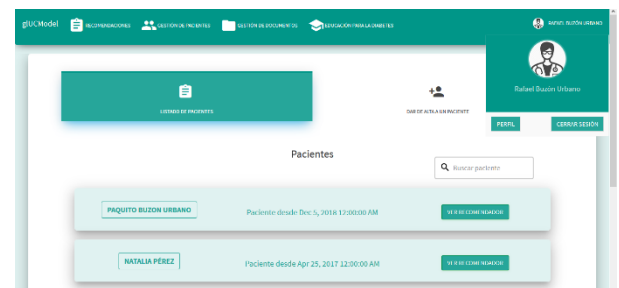
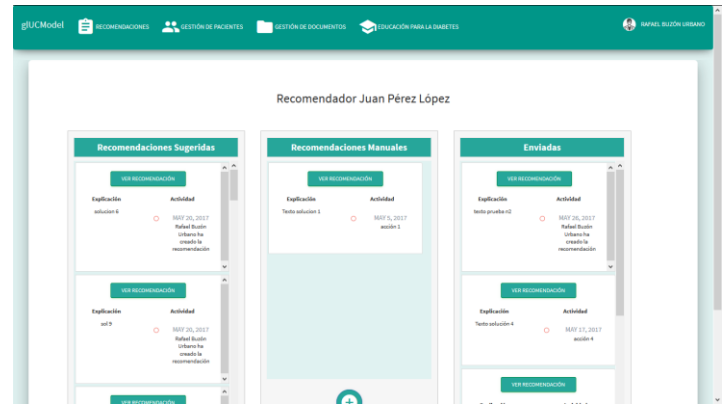
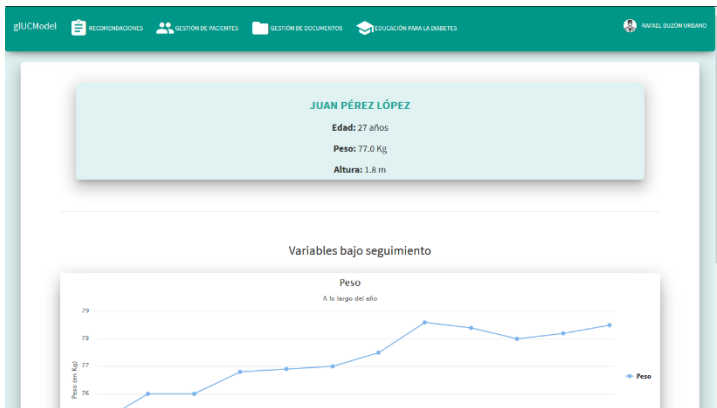
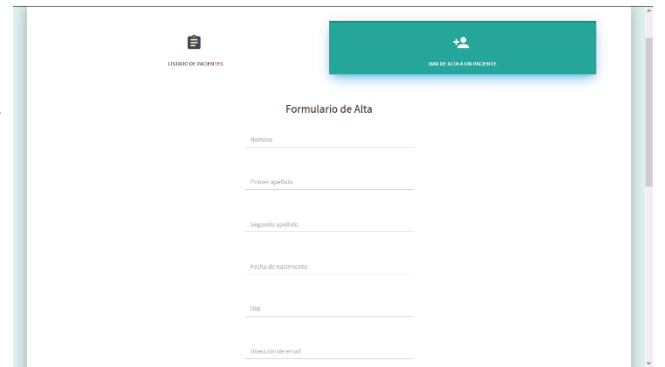
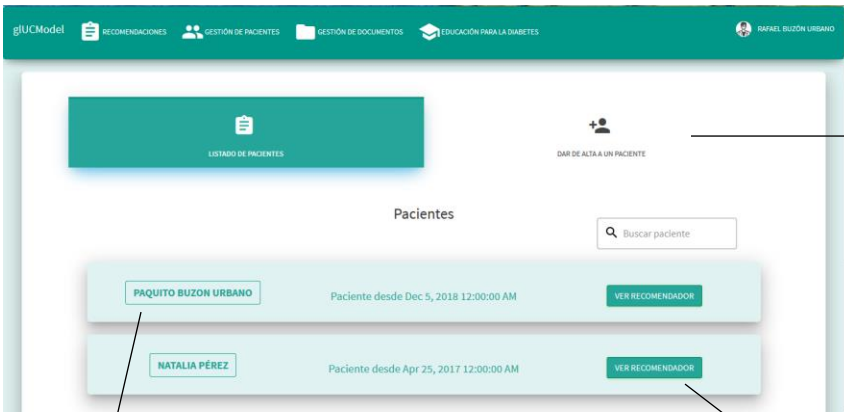
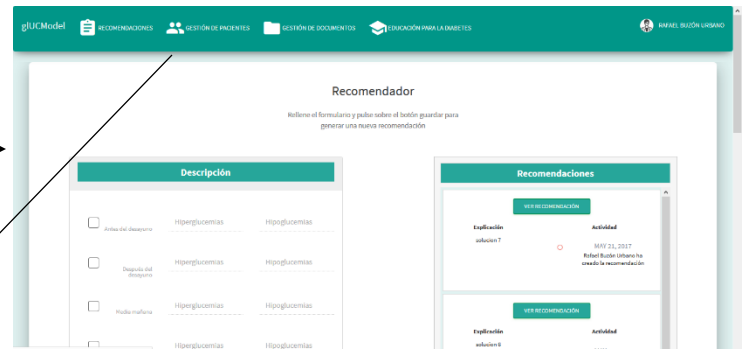
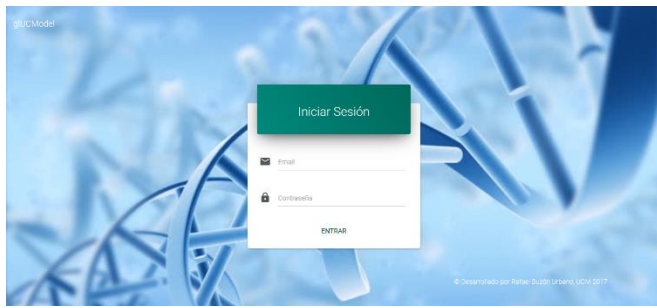


Figura 11: Flujo de la aplicación

### 3.3 Prototipos iniciales

En un principio no estaba muy claro cuál de ambas partes (paciente y medico) iba a desarrollarse, por ello los primeros bocetos en papel daban una visión más general, apareciendo interfaces que posteriormente no han sido implementadas.

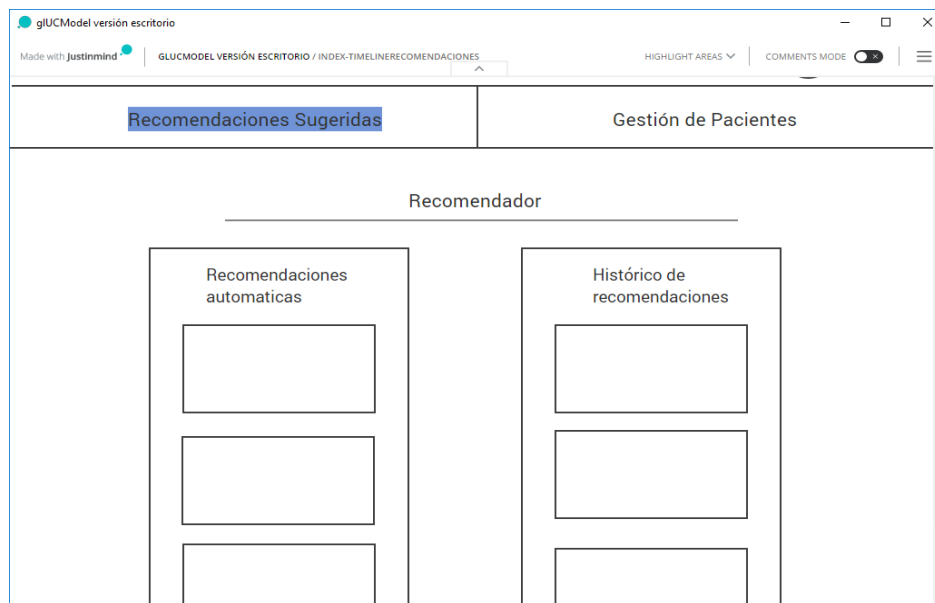
En las figuras 12 y 13 se muestran los primeros bocetos en papel



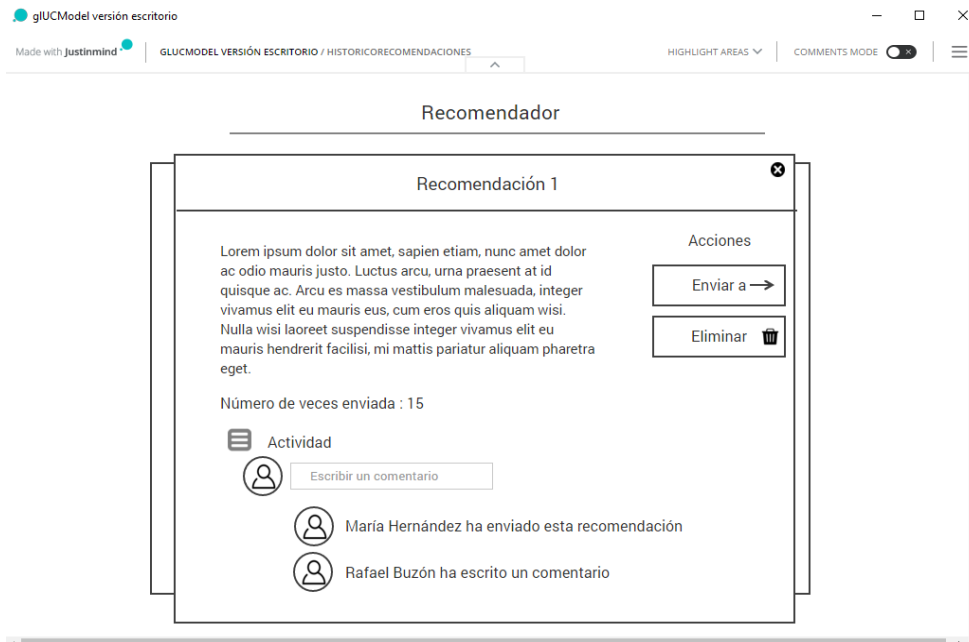
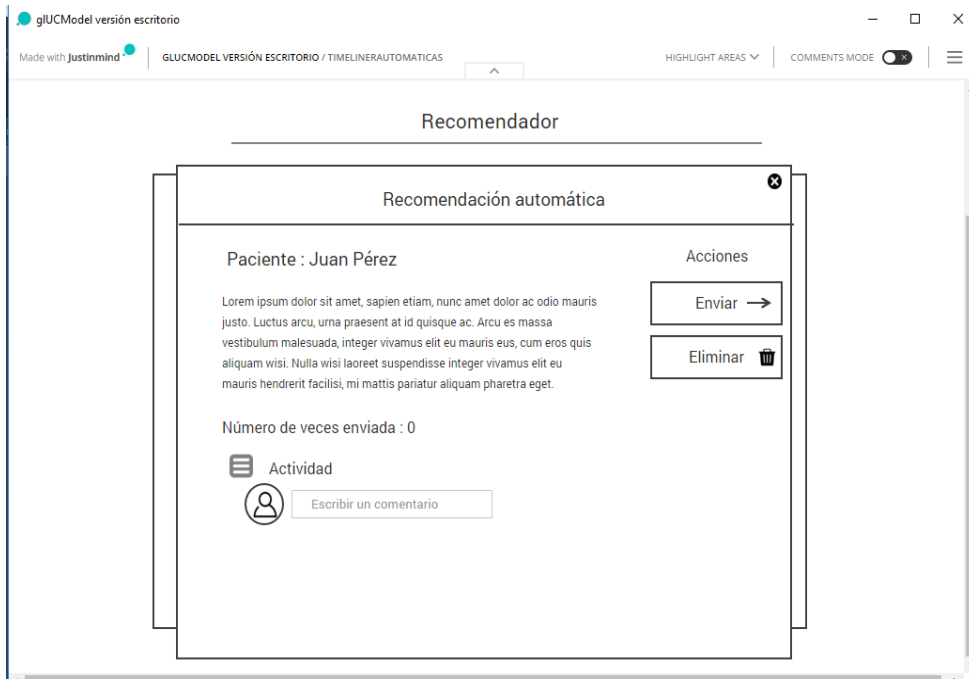
Figuras 12 y 13: Prototipos a mano iniciales de login, registro de pacientes, menú de paciente y versión móvil del menú

Según se fue avanzando en el estudio de las necesidades y los requisitos de la aplicación, se tomó la decisión de desarrollar únicamente la interfaz del médico.

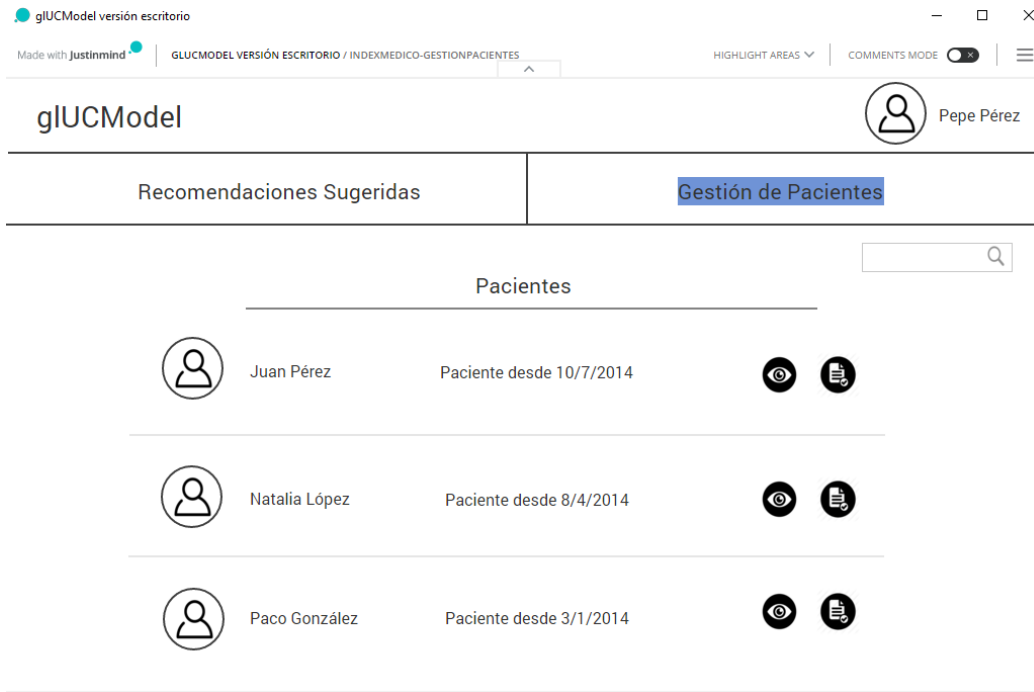
Se desarrolló un prototipo con la herramienta *Justinmind* [13] donde quedaban definidas las distintas secciones que tendría la aplicación como se muestra desde la figura 14 hasta la 21.



Figuras 14 y 15: Prototipos de login, pestañas y sección de recomendador



*Figuras 16 y 17: Prototipos de tarjeta de recomendación automática y recomendación enviada*



Figuras 18 y 19: Prototipos de listado de pacientes y perfil de paciente

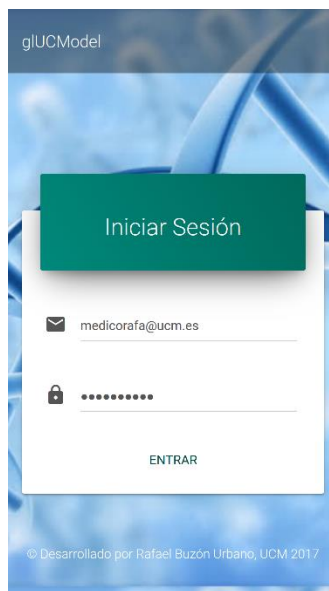
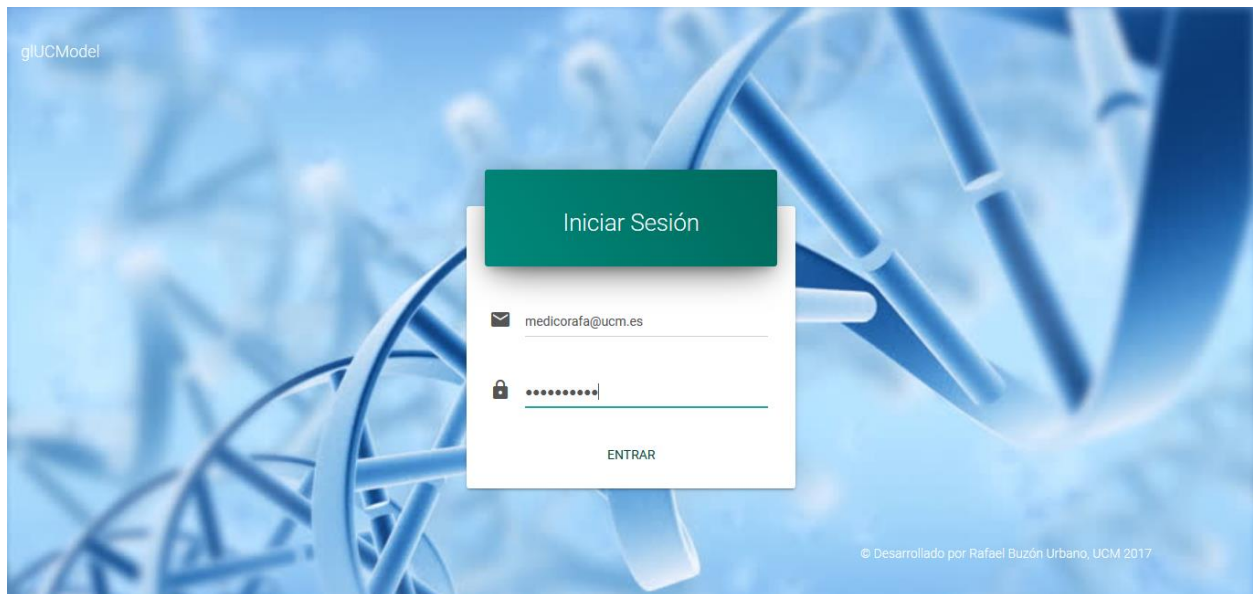


Figuras 20 y 21: Prototipos de recomendador de paciente y tarjeta de recomendación manual

## 3.4 Interfaz de médico

En esta sección se explican en detalle las distintas interfaces de la aplicación en las que el médico puede interactuar, mostrando su aspecto tanto en versión de escritorio como en versión móvil.

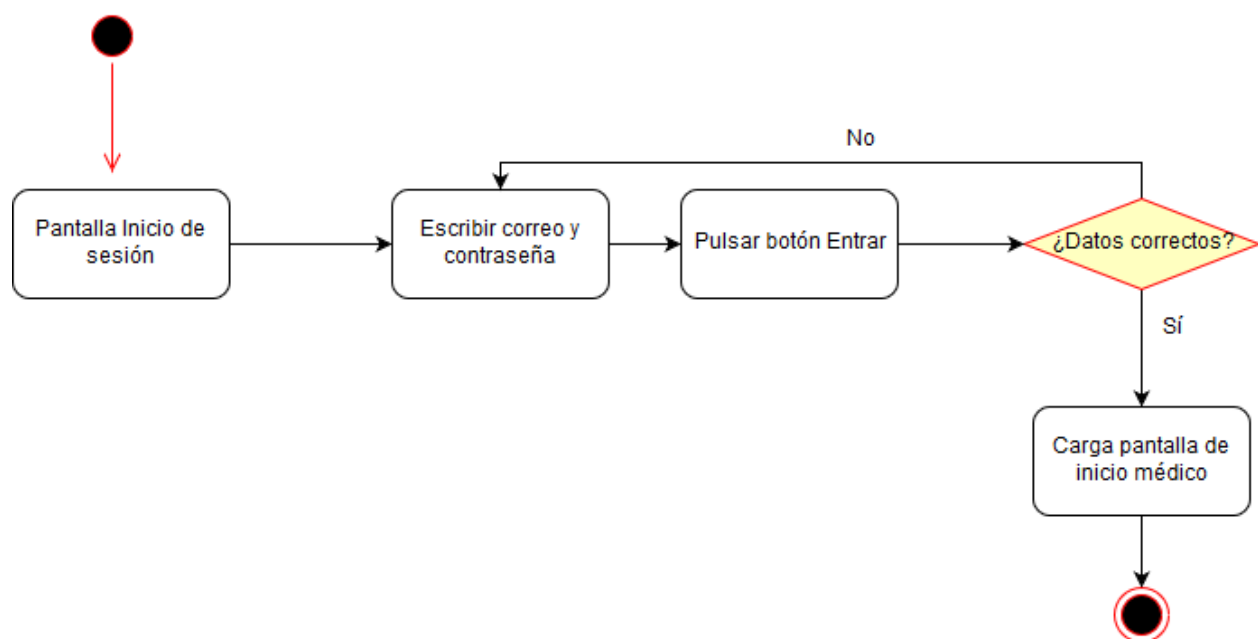
### 3.4.1 Interfaz de inicio de sesión



*Figuras 22 y 23: Interfaz inicio de sesión versión escritorio y versión móvil*

Las figuras 22 y 23 muestran la primera interfaz que encuentra el médico cuando quiere comenzar a usar la aplicación, tiene que introducir su correo electrónico y contraseña asociados de forma correcta para poder acceder al resto de funcionalidades. Pulsando sobre el botón *Entrar* se comprueba si el formato de correo electrónico es válido y posteriormente la validez de las credenciales, si hubiese algún error se le notificaría al médico para corregirlo, sino se carga la pantalla de inicio.

La figura 24 muestra el diagrama de actividad asociado a esta interfaz:



*Figura 24: Diagrama de actividad de inicio de sesión*

### 3.4.2 Interfaz página de inicio

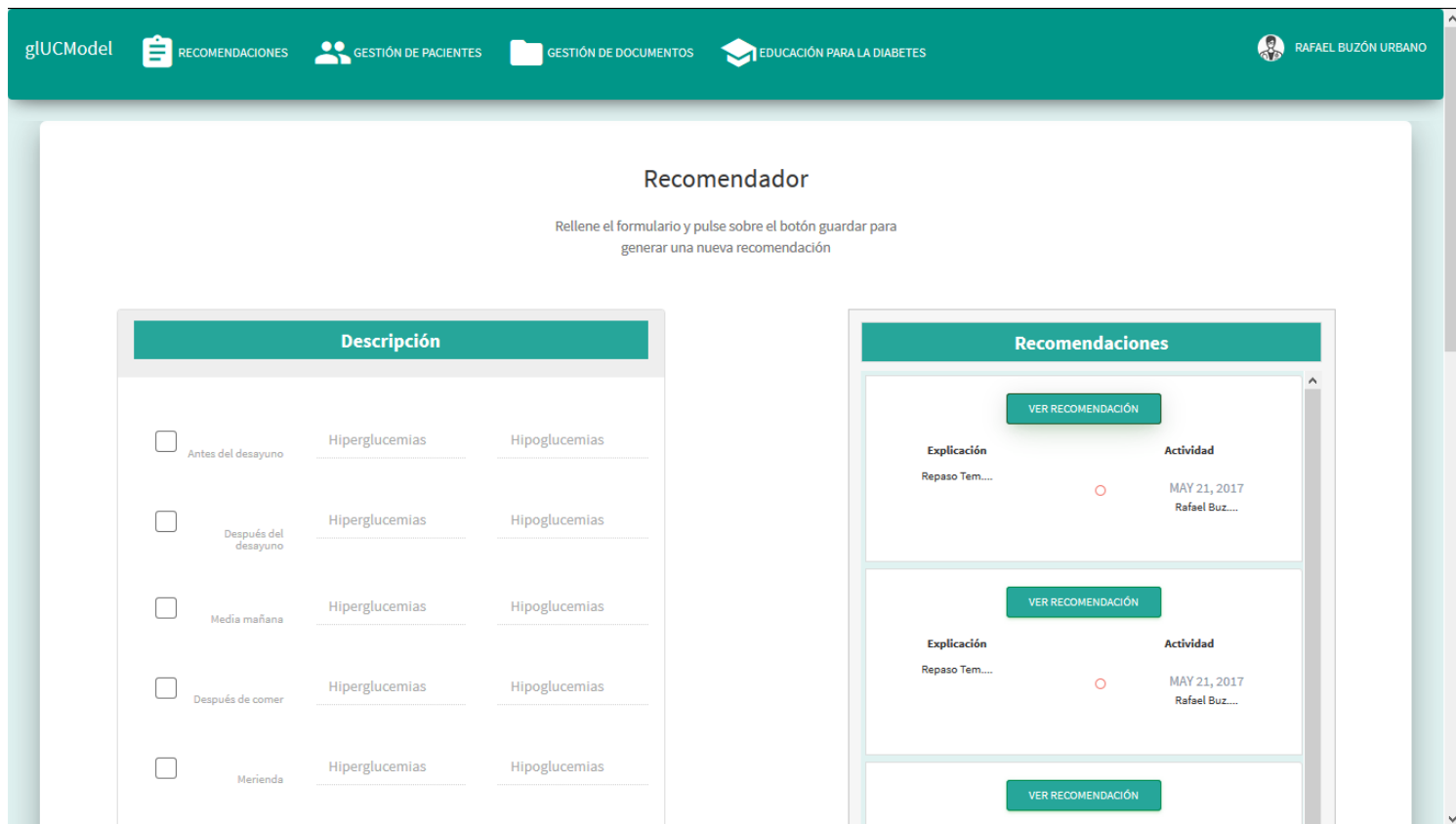


Figura 25: Interfaz página de inicio versión escritorio

En la figura 25 se muestra el recomendador general, compuesto de un formulario en la parte izquierda, y un listado de tarjetas que representan las recomendaciones pendientes de envío a todos los pacientes que tenga asignados el médico.

El formulario se compone de una descripción y una solución.

La descripción está formada por una serie de campos, como valores de Hiperglucemias e Hipoglucemias en distintas horas del día, la confirmación de la realización de un fondo de ojos en el último año o el índice de masa corporal(IMC).

La solución es un listado de acciones, en el cual el médico puede elegir la que considere oportuno o bien añadir una nueva.

Los campos referidos a las Hiperglucemias e Hipoglucemias vienen deshabilitados por defecto, siendo elección del médico (pulsando sobre el *checkbox*) cuales de estas medidas son tenidas en cuenta a la hora de generar la recomendación, este comportamiento queda reflejado en la figura 26:

Descripción			
<input type="checkbox"/>	Antes del desayuno	Hiperglucemias	Hipoglucemias
<input checked="" type="checkbox"/>	Después del desayuno	Hiperglucemias	Hipoglucemias

Figura 26: Campos habilitados y deshabilitados página de inicio

El listado de soluciones, así como el campo que permite añadir una nueva solución se muestra en la figura 27. Pulsando sobre el botón *Guardar* se genera una recomendación a los pacientes que cumplan las condiciones del formulario

### Solución

Nueva solución 

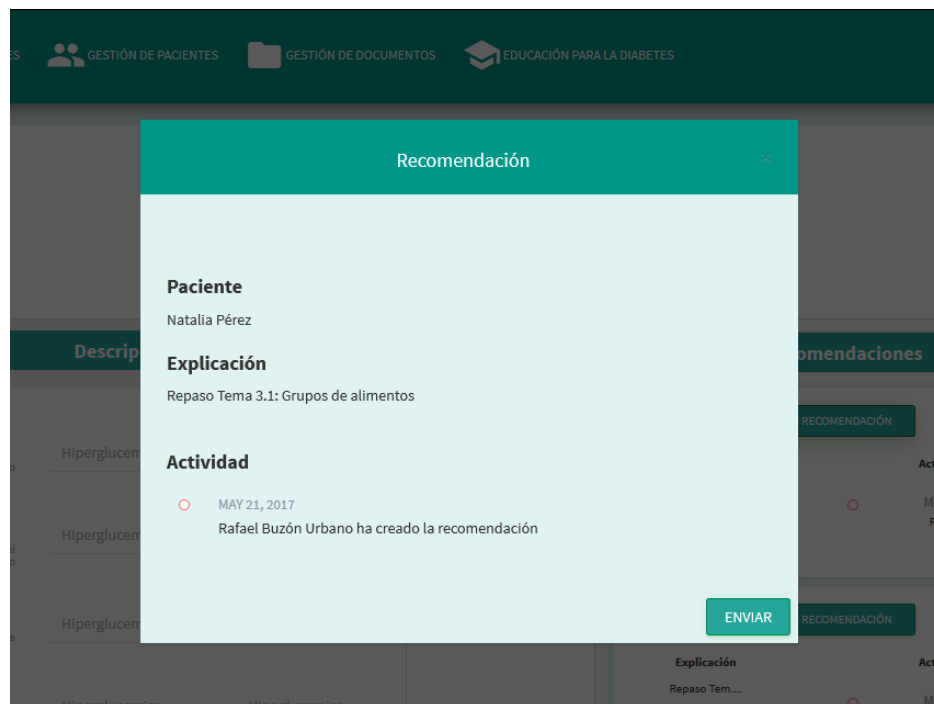
- Repaso Tema 1: Introducción
- Repaso Tema 2.1: Bases del tratamiento
- Repaso Tema 2.2: Ejercicio
- Repaso Tema 2.3: Dieta
- Repaso Tema 2.4: Antidiabéticos orales
- Repaso Tema 2.5: Insulina
- Repaso Tema 3.1: Grupos de alimentos
- Repaso Tema 3.2: Dieta por raciones. Medidas culinarias

 GUARDAR

Figura 27: Listado de soluciones página de inicio

En el listado de tarjetas encontramos las recomendaciones pendientes de envío. Cada tarjeta presenta una vista previa tanto del texto destinado al paciente como de la actividad que ha tenido esa recomendación a lo largo del tiempo, pulsando sobre el botón *Ver Recomendación* vemos una vista completa de la recomendación, dándonos la opción de cerrarla o proceder a su envío.

Este comportamiento queda reflejado en las figuras 28 y 29:



Figuras 28 y 29: Vista previa y vista completa de tarjeta de recomendación

Para terminar esta sección, se muestra en las figuras 30 y 31 el aspecto de la página de inicio en versión móvil y el diagrama de actividad asociado a esta interfaz.



Figura 30: Interfaz página de inicio versión móvil

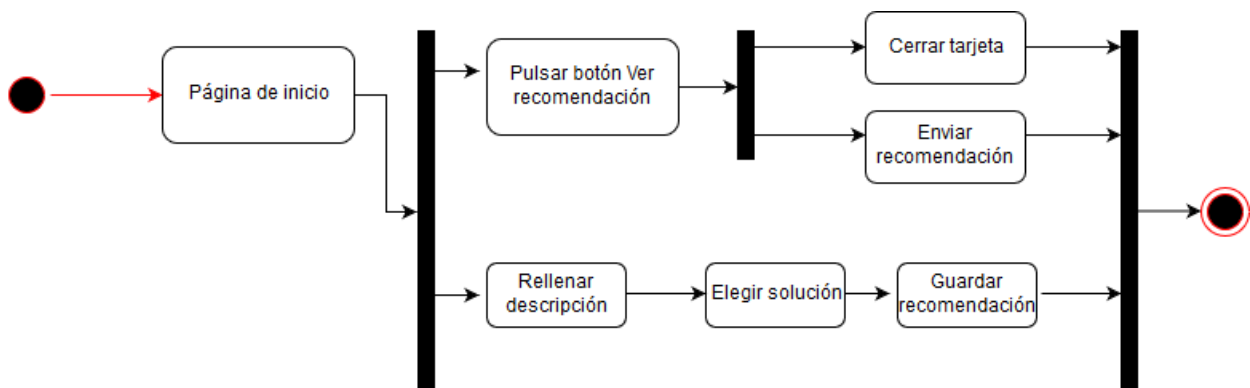
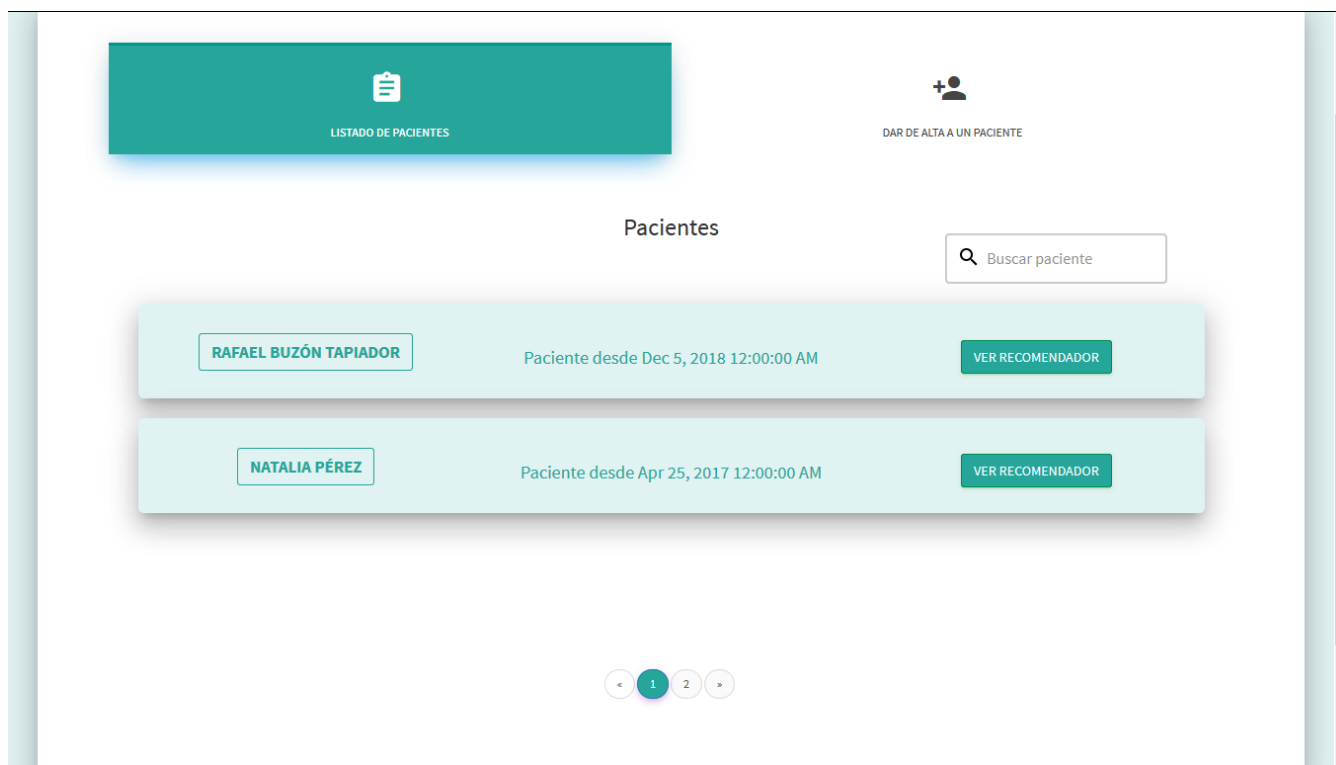


Figura 31: Diagrama de actividad página de inicio

### 3.4.3 Interfaz gestión de pacientes

En esta interfaz se va a dividir la descripción en cada una de las pestañas en las que está dividida la gestión de pacientes: El listado de pacientes y el dar de alta a un paciente nuevo.

#### 3.4.3.1 Listado de pacientes

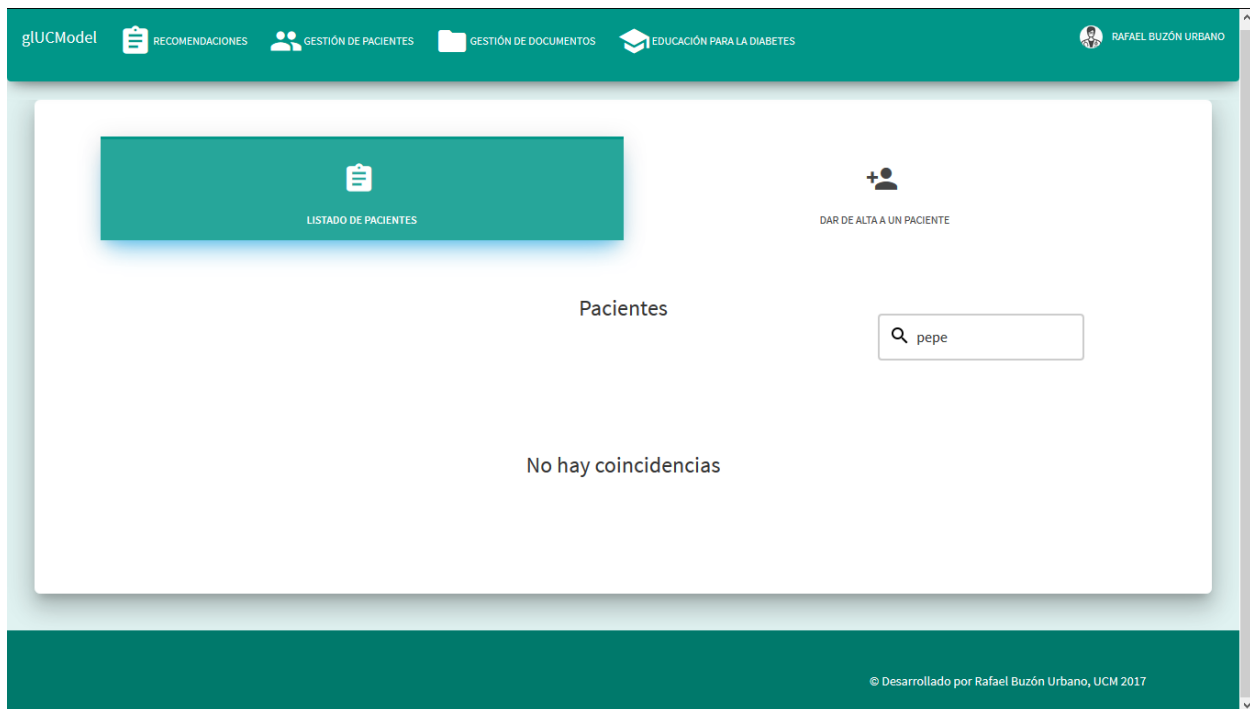
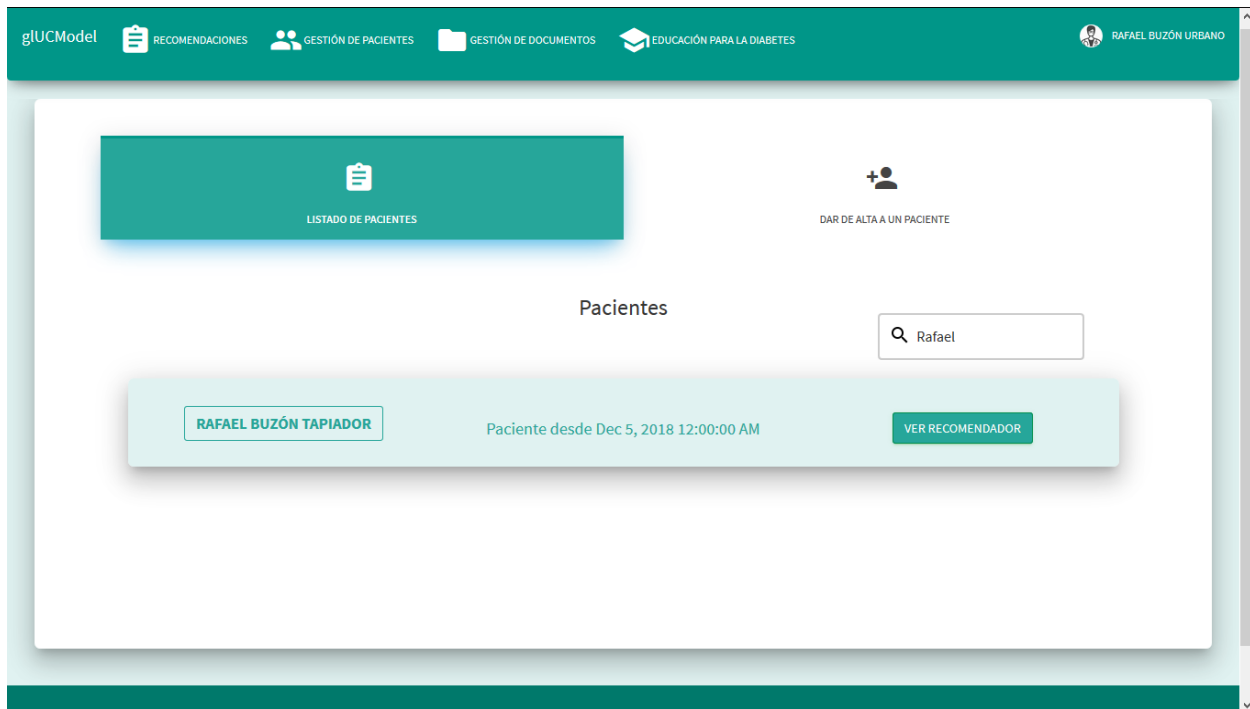


*Figura 32: Listado de pacientes*

En la figura 32 se muestra el listado de pacientes asignados al médico. El listado se presenta ordenado por fecha de alta en el sistema, mostrando una paginación que facilita la organización y claridad del mismo.

En la parte superior derecha del listado se encuentra el buscador de pacientes, tecleando el nombre del paciente el listado se actualiza en tiempo real mostrando los resultados que coincidan con la búsqueda (figura 33). En caso de no encontrar coincidencias se muestra un mensaje informando al médico de ello (figura 34).

Para volver a mostrar el listado inicial basta con borrar la búsqueda que hubiésemos realizado previamente.



*Figuras 33 y 34: Resultados búsqueda de paciente*

En cada paciente se nos muestra la siguiente información: nombre y apellidos, fecha de registro en el sistema y el botón para acceder a su recomendador personalizado. Pulsando sobre el nombre del paciente accedemos a su perfil, sección que se cuenta en detalle más adelante en esta memoria.

En la figura 35 se muestra el diagrama de actividad asociado al listado de pacientes:

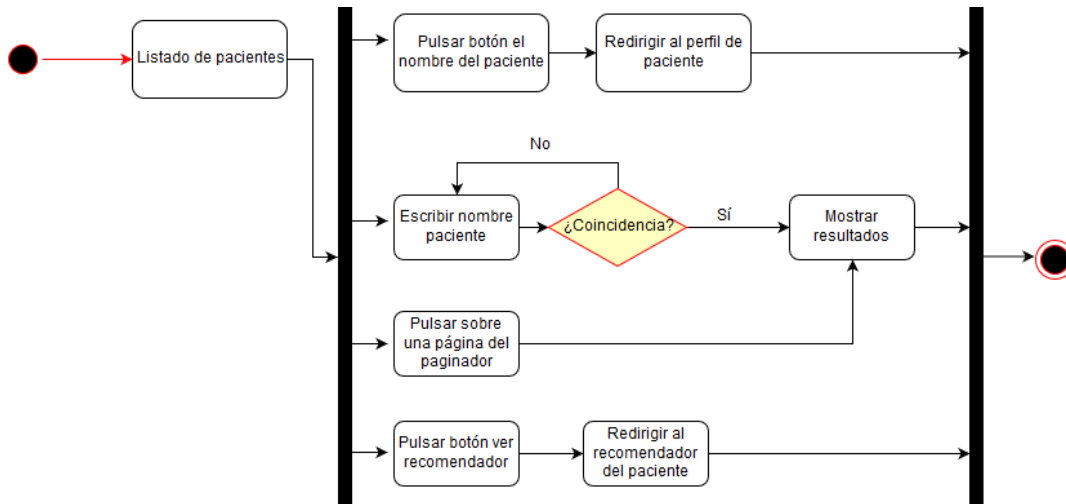


Figura 35: Diagrama de actividad listado de pacientes

### 3.4.3.2 Dar de alta a un nuevo paciente

Figura 36: Formulario de alta nuevo paciente

En la figura 36 se muestra el formulario de alta para nuevos pacientes, con datos como nombre, apellidos, fecha de nacimiento, DNI (entre otros) que debe rellenar el médico para poder dar de alta a un nuevo paciente.

Pulsando sobre el botón *Dar de Alta* se comprueba la validez de los datos, en caso de existir algún error se señala para que el médico pueda rectificarlo, en caso contrario el paciente es dado de alta con éxito y aparece en el listado de pacientes descrito en la sección 3.4.3.1.

En las figuras 37 y 38 se muestra el diagrama de actividad asociado al listado de pacientes y como es el aspecto de la interfaz de gestión de pacientes en versión móvil:

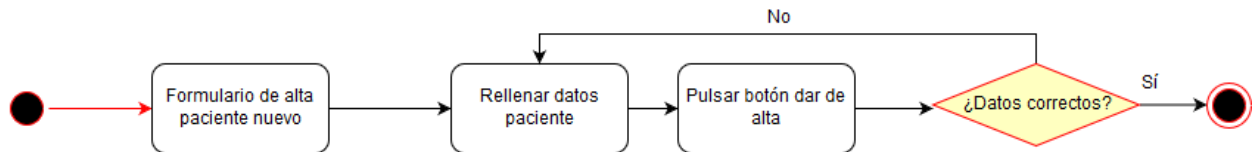


Figura 37: Diagrama de actividad formulario de alta nuevo paciente

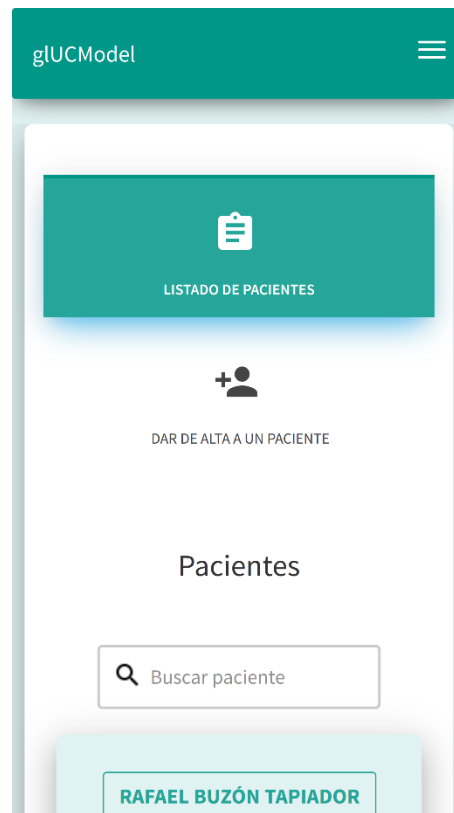


Figura 38: Interfaz gestión de pacientes versión móvil

### 3.4.4 Interfaz recomendador del paciente

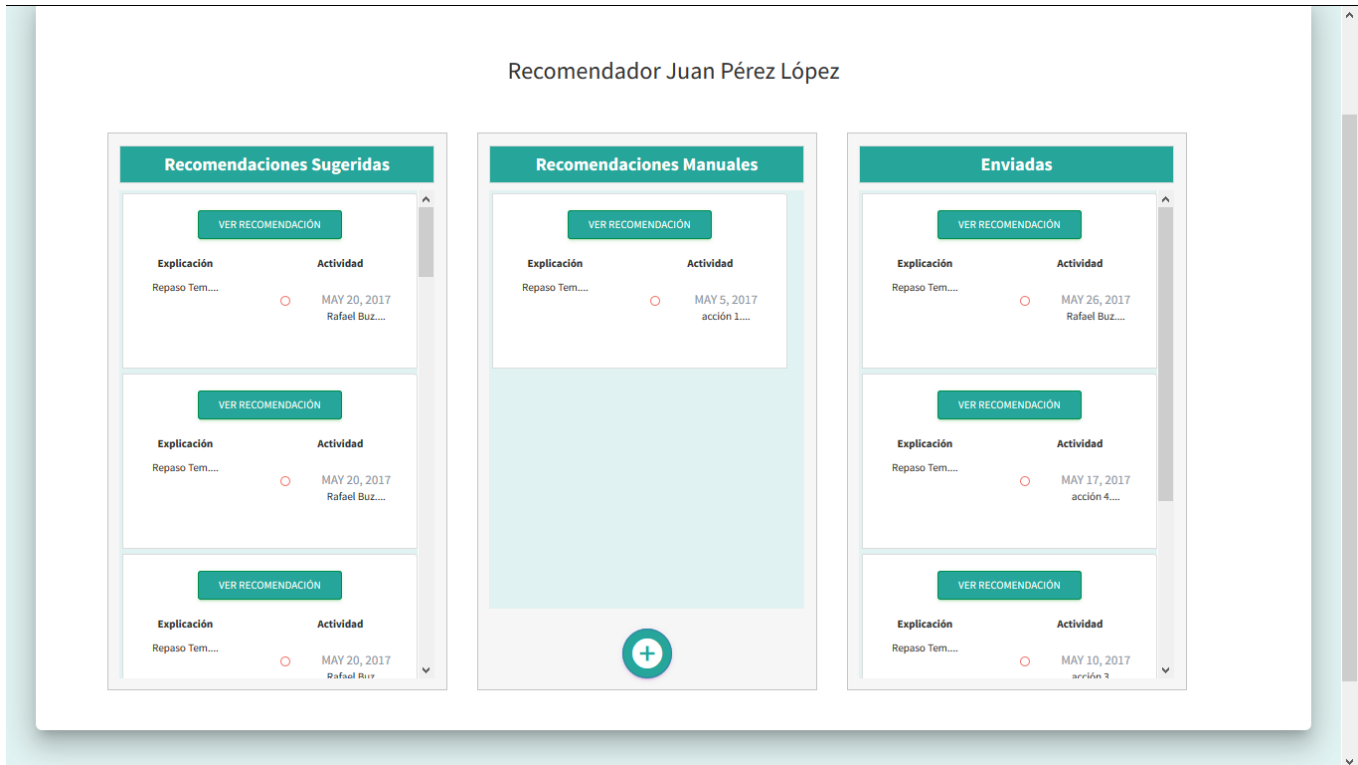


Figura 39: Interfaz recomendador paciente versión escritorio

En esta interfaz se encuentra el recomendador personalizado del paciente, el cual está dividido en 3 columnas: *Recomendaciones Sugeridas*, *Recomendaciones Manuales* y *Enviadas*.

Se ha seguido este diseño buscando una adaptación de la metodología *Kanban*, la cual es una palabra japonesa que significa “tarjetas visuales” [14].

El sistema *Kanban* como tal surgió en Toyota, el fabricante japonés de automóviles. Su objetivo era organizar mejor su producción de vehículos dividiendo el proceso en fases bien delimitadas, las cuales se tenían que cubrir correctamente para pasar a la siguiente fase pudiendo garantizar un producto de calidad. De este sistema, surgió el método *Kanban*, ideado por David J. Anderson y que adapta la filosofía original al desarrollo de software, con diferentes fases, equipos de trabajo y el objetivo de que cada pieza del programa a crear funcione correctamente y sea de la mejor calidad posible. El método *Kanban* en su versión moderna aplicada al software se usó por primera vez en Microsoft, y desde

entonces ha sido aplicado en cientos de proyectos de todo el mundo. Su propósito es ser una técnica de gestión de las tareas muy visual, gestionándolas mediante tarjetas.

Estas tarjetas contienen distinto tipo de información y permiten ser arrastradas entre las diferentes columnas para cambiar su estado.

En glUCModel 2.0 se ha utilizado para la gestión de distintos tipos de recomendaciones: las sugeridas por el sistema, las creadas manualmente por el médico y las recomendaciones ya enviadas, pudiendo arrastrar cualquiera de las dos primeras a la columna de *Enviadas*, consiguiendo un proceso intuitivo y sencillo de envío, la transición de las tarjetas queda reflejada en la figura 40:

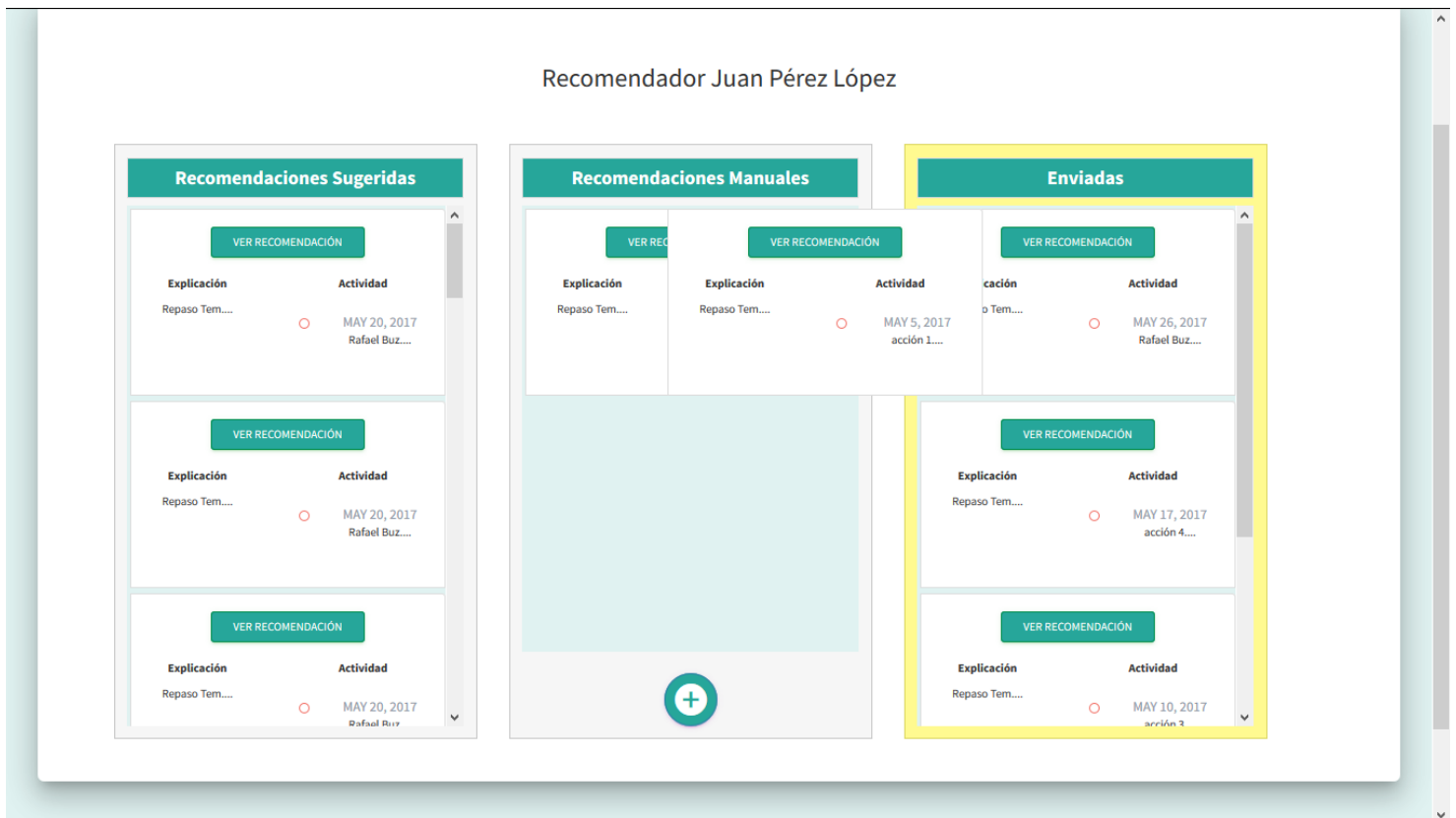


Figura 40: Envío de tarjetas de recomendaciones mediante arrastre

En cualquiera de las tarjetas de las 3 columnas, podemos pulsar sobre el botón *Ver recomendación* para obtener una vista completa de la recomendación como se ve en la figura 41, de forma similar a como sucede en la sección 3.4.2

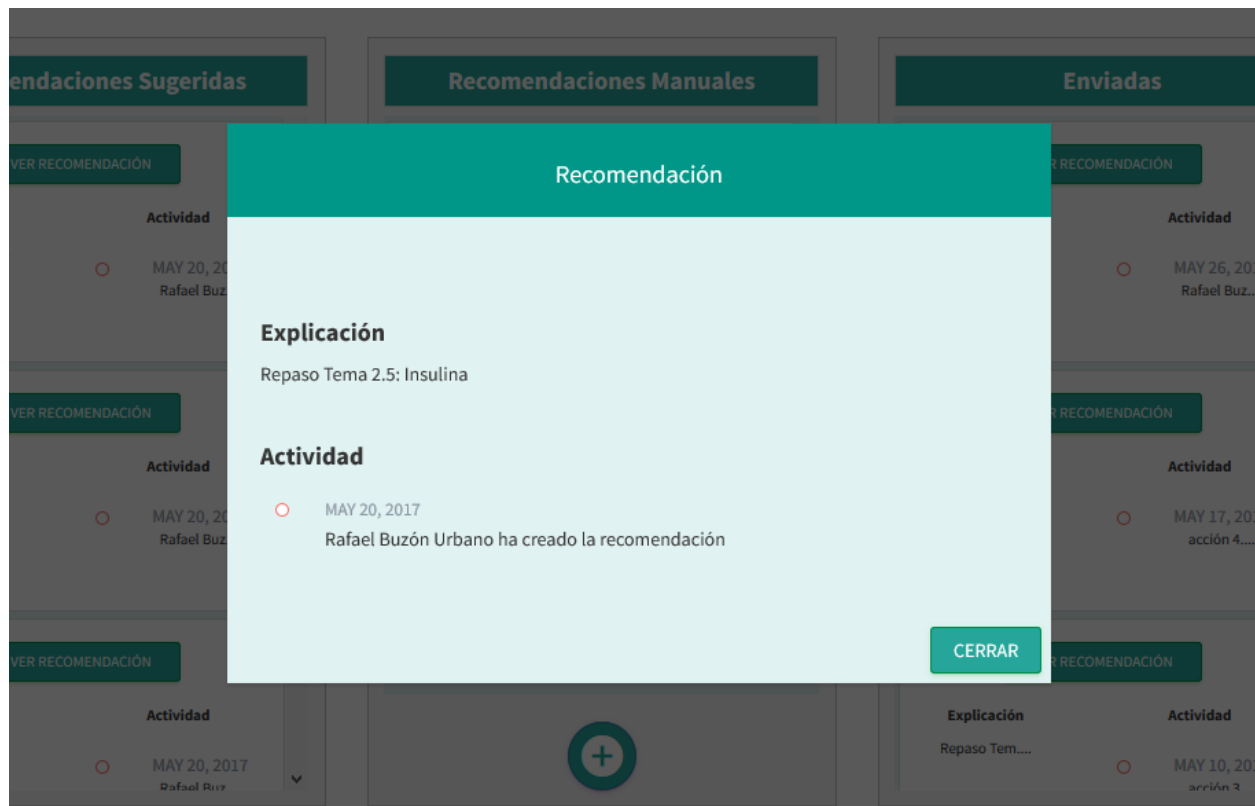


Figura 41: Vista completa tarjeta recomendación sugerida

En la columna de *Recomendaciones manuales* podemos crear una nueva recomendación pulsando sobre el botón + situado en la parte inferior de la columna, abriéndose el siguiente formulario donde se elige una solución del listado o bien se puede crear una nueva, para posteriormente pulsar sobre el botón *Añadir* y que la recomendación quede añadida con éxito, como se muestra en las figuras 42 y 43:

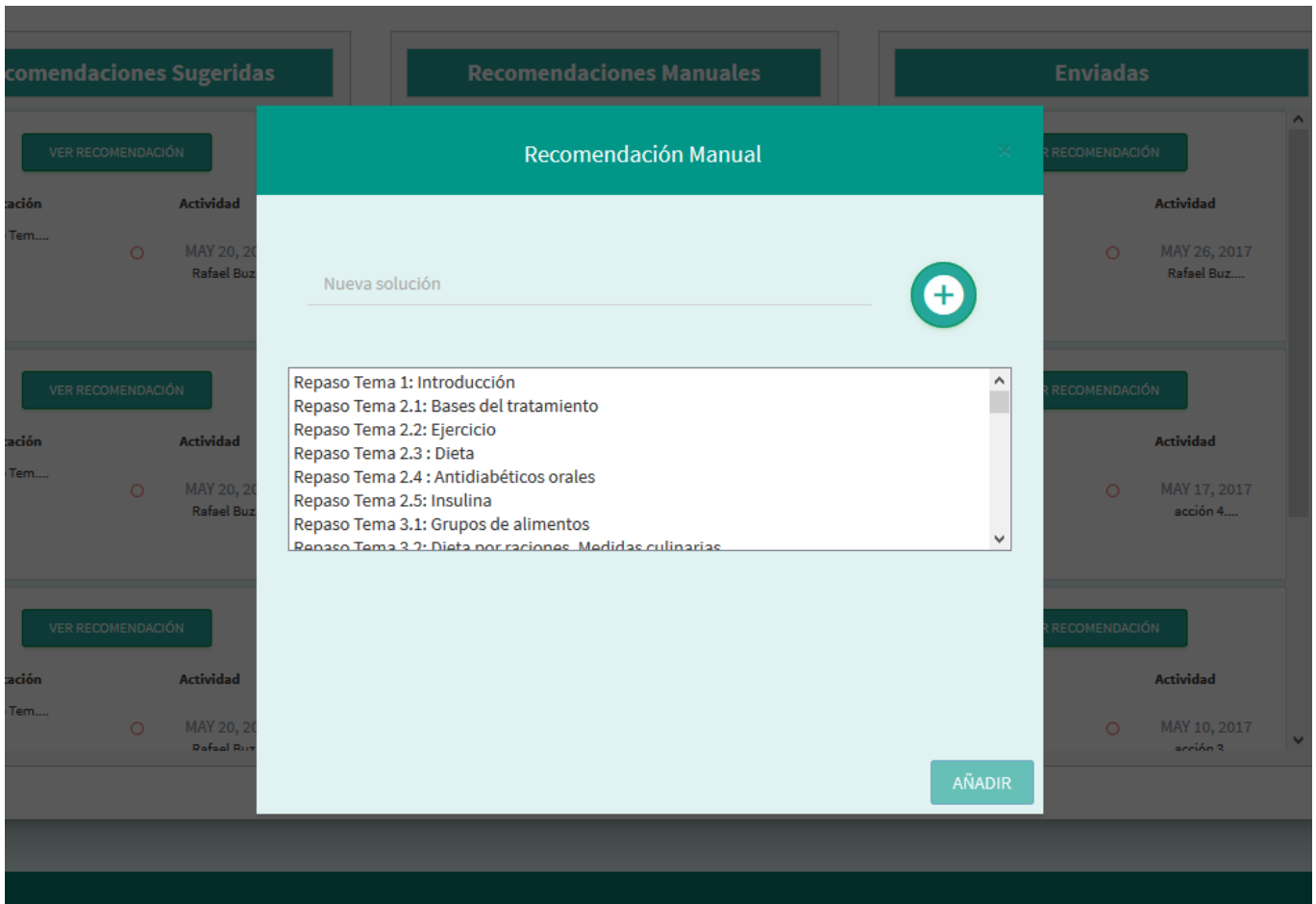


Figura 42: Formulario recomendación manual



Recomendador Juan Pérez López



Figura 43: Recomendación manual añadida con éxito

Las figuras 44 y 45 muestran el diagrama de actividad asociado a la interfaz del recomendador del paciente y como es el aspecto de la interfaz en versión móvil:

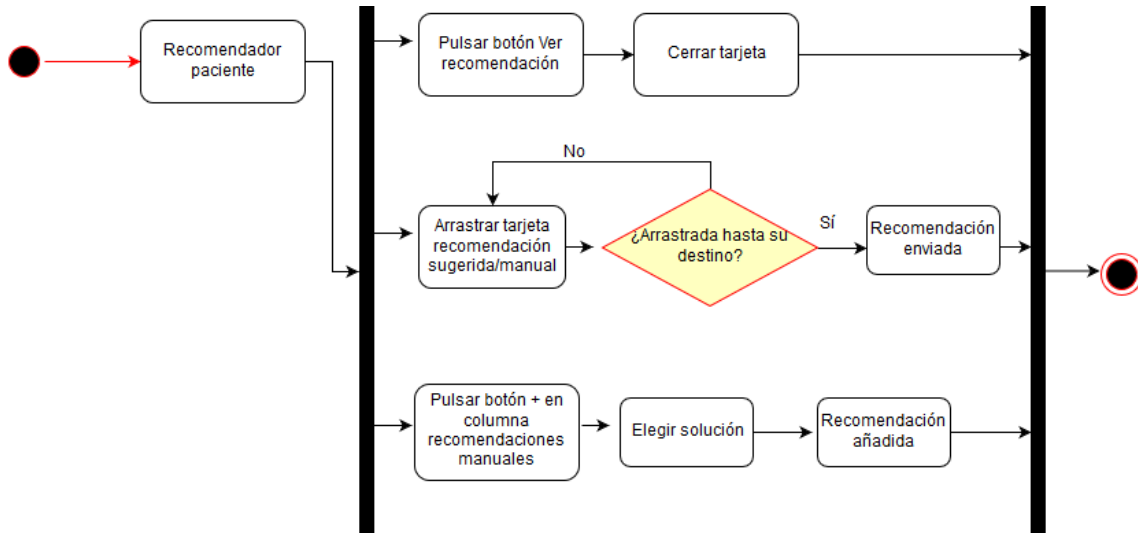


Figura 44: Diagrama de actividad recomendador paciente



Figura 45: Interfaz recomendador paciente versión móvil

### 3.4.5 Interfaz perfil del paciente

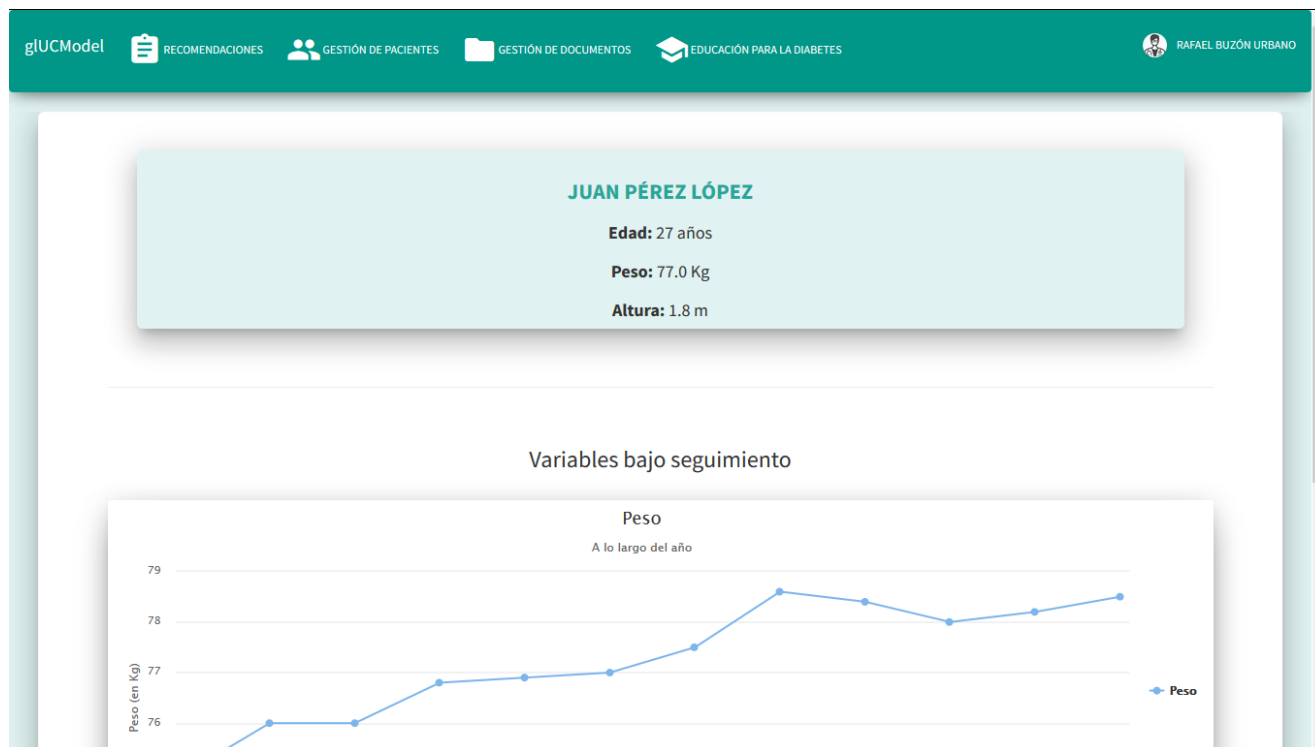


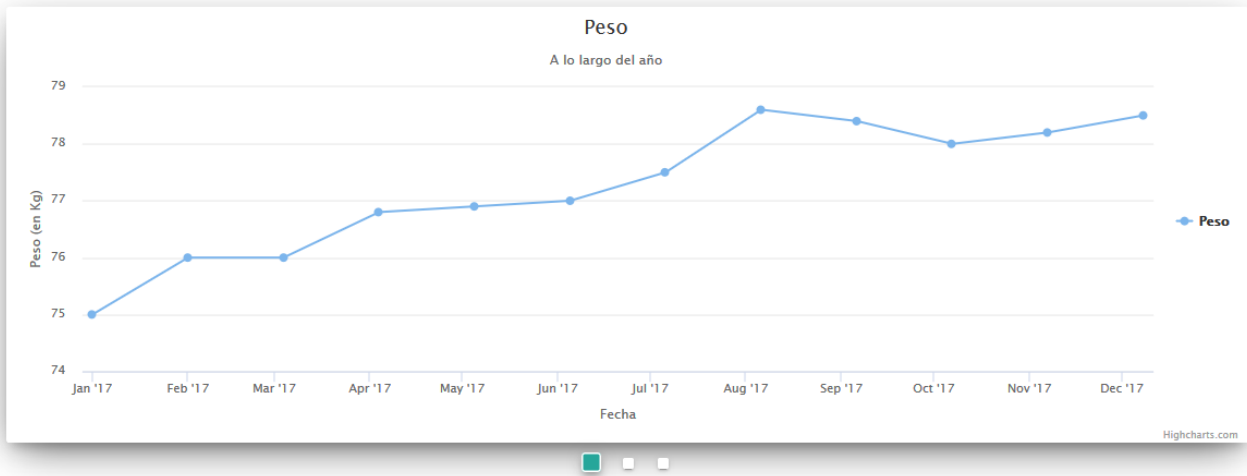
Figura 46: Interfaz perfil paciente versión escritorio

En la figura 46 se muestra el perfil del paciente, el cual consta de dos partes diferenciadas: datos básicos del paciente en la parte superior y variables bajo seguimiento en la parte inferior.

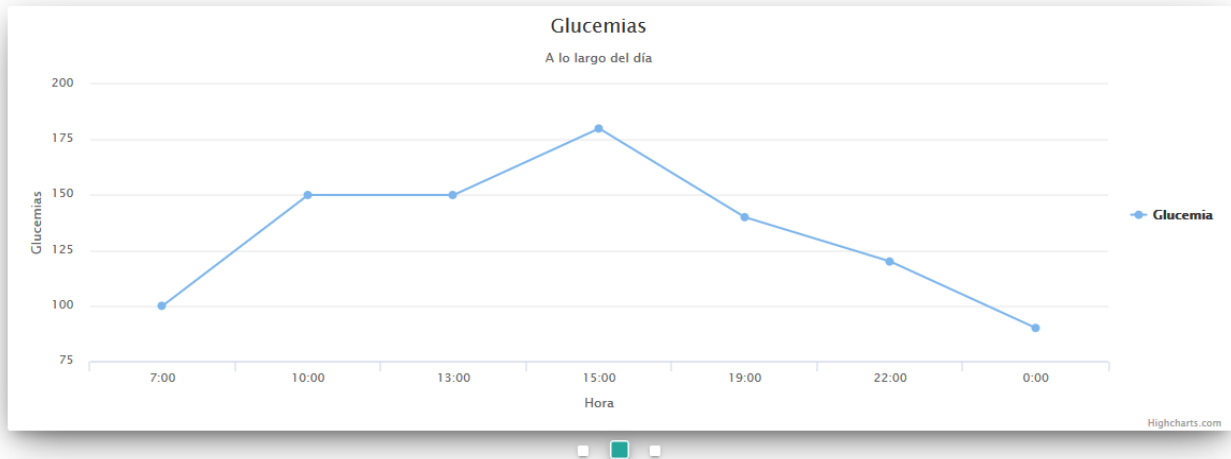
En los datos básicos se encuentra el nombre y apellidos del paciente, su edad, peso actual y altura.

En las variables bajo seguimiento se encuentra un carrusel de gráficas con distintos datos: peso a lo largo del año, glucemias a lo largo del día e insulina a lo largo del día, tal y como se muestra en las figuras 47, 48 y 49:

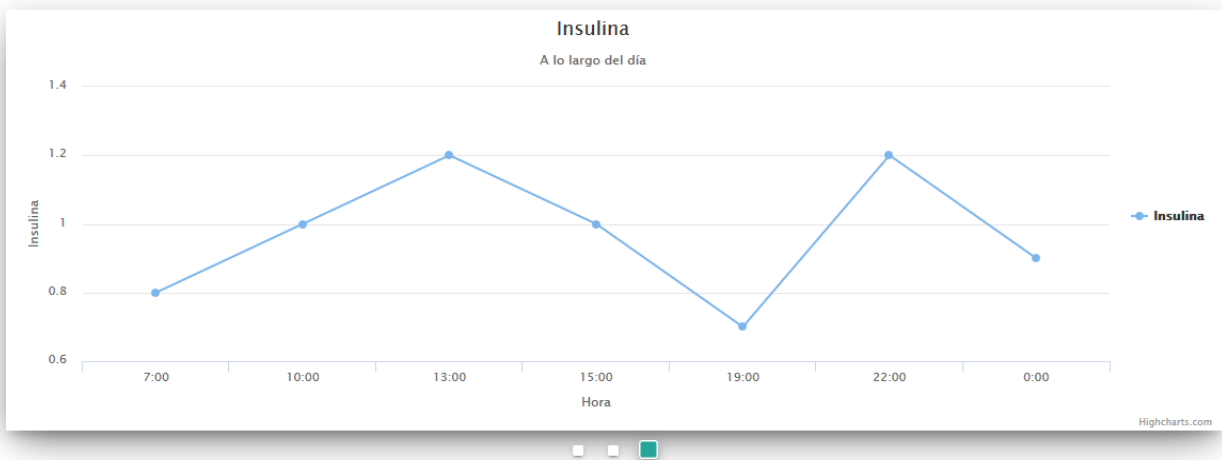
### Variables bajo seguimiento



### Variables bajo seguimiento



### Variables bajo seguimiento



Figuras 47, 48 y 49: Carrusel gráficas perfil paciente

Las figuras 50 y 51 muestran el diagrama de actividad asociado a la interfaz del perfil del paciente y como es el aspecto de la interfaz en versión móvil:

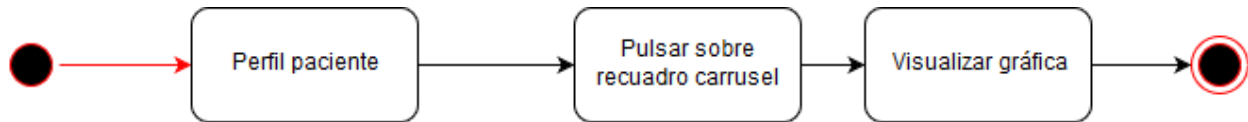


Figura 50: Diagrama de actividad perfil paciente

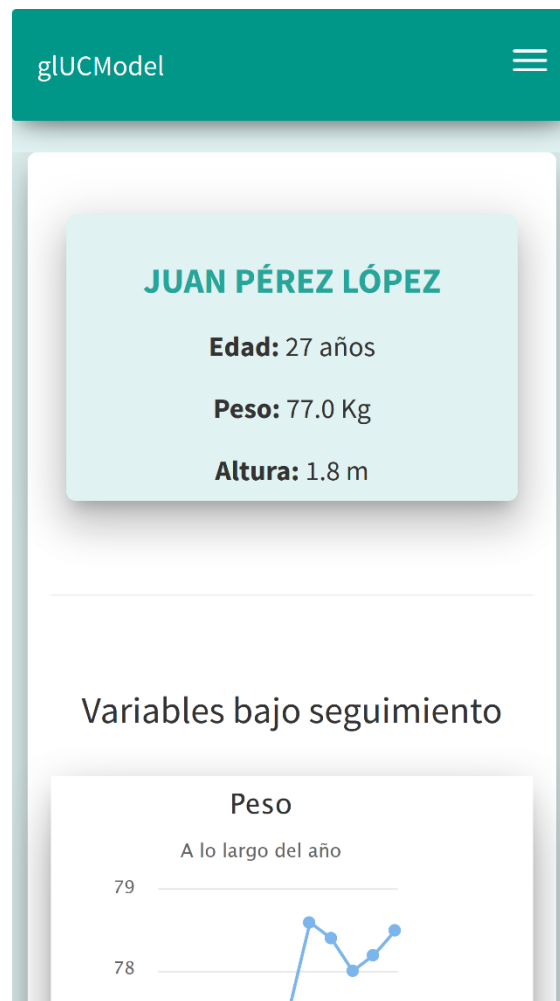
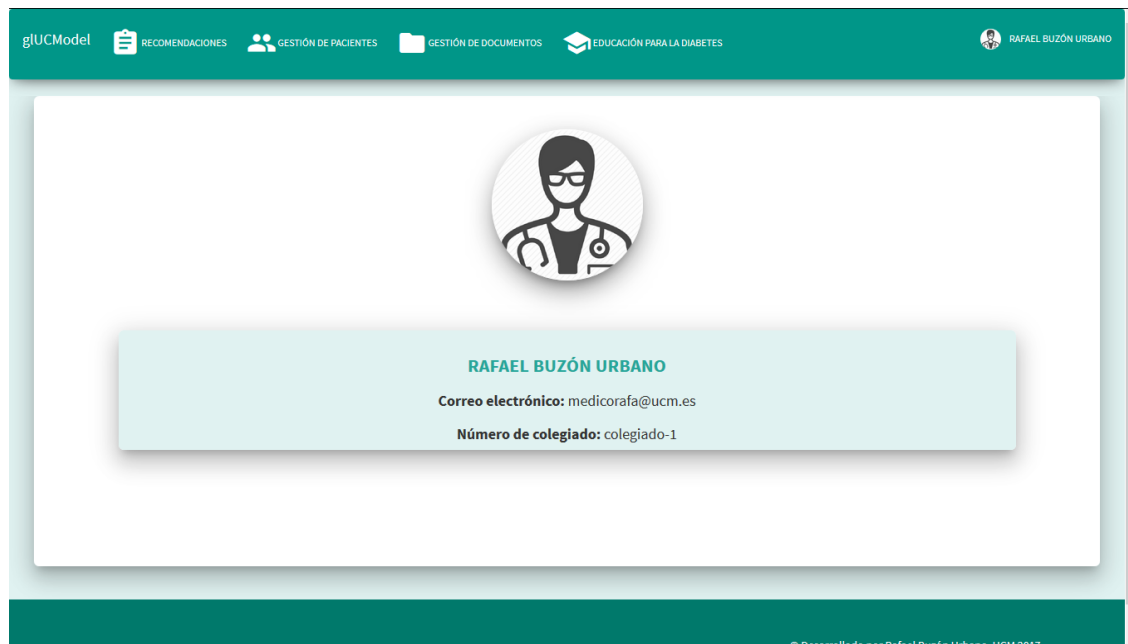


Figura 51: Interfaz perfil paciente versión móvil

### 3.4.6 Interfaz perfil médico

En las figuras 52 y 53 se muestra la interfaz del perfil médico



*Figuras 52 y 53: Interfaz perfil médico versión escritorio y versión móvil*

## Capítulo 4. IMPLEMENTACIÓN

En este capítulo se detalla la arquitectura de la aplicación y el proceso de creación de la misma.

Se describen las distintas tecnologías aplicadas, los lenguajes de programación y entorno de desarrollo utilizado.

### 4.1 Arquitectura

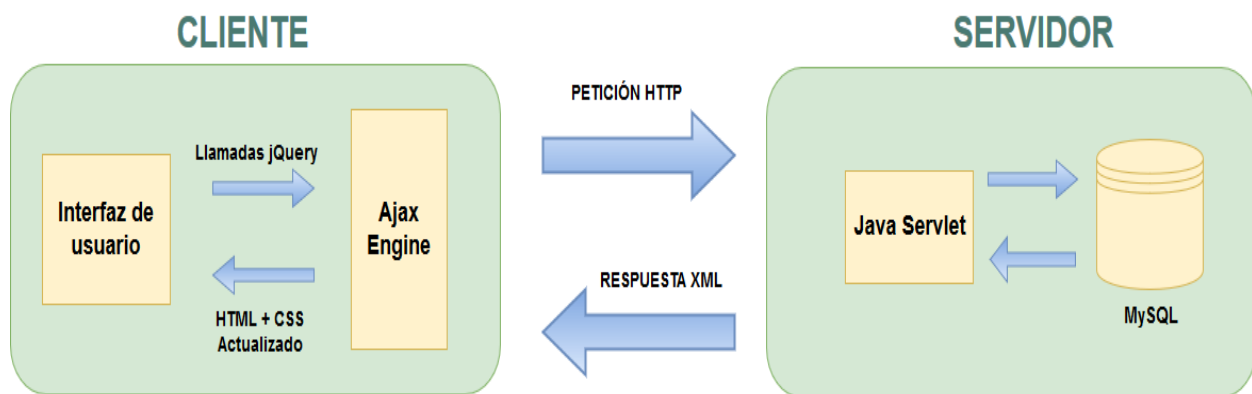


Figura 54: Arquitectura de la aplicación glUCModel 2.0

Como muestra la figura 54, la arquitectura de la aplicación es una arquitectura *cliente-servidor* basada en servicios web (*Web services*). En este sistema el *cliente* (parte *front-end*) solicita servicios al *servidor* (parte *back-end*) mediante peticiones *AJAX* (Asynchronous JavaScript And XML) a un servicio web, este procesa la petición y devuelve una respuesta al cliente utilizando el formato *JSON* (JavaScript Object Notation).

En la parte del *servidor* o *back-end* se encuentran distintos *Java Servlets*, que implementan funciones que hacen uso de los recursos del sistema, esto es, la información almacenada en la base de datos. Los datos están almacenados en una base de datos *MySQL*, en la que se entra en detalle más adelante.

La parte del *cliente* o *front-end* corresponde a glUCModel 2.0, aplicación web que está construida siguiendo los principios de diseño adaptativos, utilizando para ello varios templates de *Bootstrap 3*, así como distintas tecnologías web como *HTML 5*, *CSS 3*, *jQuery* y *AJAX*.

#### 4.1.1 Interacción entre *front-end* y *back-end*

En esta sección se explica el uso que hace glUCModel 2.0 (parte *front-end*) de los servicios proporcionados por el *back-end*.

Se le dedica este apartado en la memoria para aclarar la relación que existe entre el *front-end* y el *back-end*, es decir el funcionamiento interno de la aplicación, para en apartados posteriores explicar en profundidad cada una de las dos partes por separado.

Los servicios prestados desde las distintas interfaces de la aplicación son los siguientes:

- En la interfaz de inicio de sesión (ver sección 3.4.1) se hace uso de los servicios web que proporcionan información sobre:
  - Correo y contraseña asociadas a médico.
  - Número de colegiado.
- En la interfaz de la página de inicio (ver sección 3.4.2) se hacen uso de los servicios web que proporcionan información sobre las recomendaciones manuales sin enviar:
  - Nombre y apellidos del paciente.
  - Solución asignada a la recomendación.
  - Actividad asociada a la recomendación.

Además del acceso a estos recursos del sistema, también se requieren servicios web que proporcionen otros datos para el correcto funcionamiento del formulario inicial:

- Pacientes asignados al médico.
- Valores de Hiperglucemias e Hipoglucemias en distintos momentos

- del día de estos pacientes.
  - Índice de masa corporal(IMC) así como la fecha del último fondo de ojos.
  - Listado de soluciones almacenadas en el sistema.
- En la interfaz de gestión de pacientes (ver sección 3.4.3) se hacen uso de los servicios web que proporcionan información sobre los pacientes asignados al médico:
  - Nombre y apellidos del paciente, así como su ID asignado.
  - Fecha del registro en el sistema.
- En la interfaz recomendador del paciente (ver sección 3.4.4) se hacen uso de los servicios web que proporcionan información sobre los distintos tipos de recomendaciones.

Por un lado, para las recomendaciones manuales y para las enviadas se proporciona la siguiente información:

  - Solución asignada a la recomendación
  - Actividad asociada a la recomendación.

Para poder mostrar las recomendaciones sugeridas por el sistema, además la información descrita arriba, se hacen uso de servicios web que proporcionan información sobre:

  - Último registro de valores de Hiperglucemias e Hipoglucemias del paciente concreto.
  - Registros de valores de Hiperglucemias e Hipoglucemias de todos los pacientes dados de alta en glUCModel 2.0.
- En la interfaz perfil del paciente (ver sección 3.4.5) se hacen uso de los servicios web que proporcionan información sobre el paciente, así como de distintas variables asociadas a él:
  - Nombre y apellidos.
  - Edad y peso actuales.
  - Pesos registrados en el último año.
  - Valores de glucemias a lo largo del día.
  - Valores de insulina a lo largo del día.

- En la interfaz perfil médico (ver sección 3.4.6) se hace uso del servicio web que proporciona información sobre:
  - Nombre y apellidos del médico.
  - Correo electrónico asociado a la cuenta.
  - Número de colegiado.

## 4.2 Implementación del *front-end*

En esta sección se explica en detalle la implementación de la parte *front-end* de glUCModel 2.0, hablando de como se ha estructurado y de las tecnologías empleadas para cumplir los objetivos propuestos de conseguir una aplicación de uso intuitivo y que siga los principios de diseño adaptativo.

El *front-end* de la aplicación está dividido en 3 bloques principales: recomendador principal, gestión de pacientes y recomendador personalizado del paciente.

A pesar de que cada uno de ellos tiene sus particularidades, se han respetado las mismas líneas de diseño, las mismas líneas de estilo tanto en los colores como en las fuentes elegidas, así como la organización del contenido dentro de la sección. Cumplir estas directrices es importante para conseguir una buena experiencia de uso, consiguiendo que la transición entre distintas secciones se sienta de forma fluida y natural.

### 4.2.1 Entorno de desarrollo

Uno de los objetivos propuestos era conseguir una interfaz intuitiva y adaptable a cualquier tipo de dispositivo, glUCModel 2.0 es una aplicación web así que la elección fue utilizar el *framework* de diseño *Bootstrap* [15].

*Bootstrap* es un *framework Open Source* (de código abierto) desarrollado por *Twitter*, con su uso se consigue simplificar la tarea de maquetación y proporciona una guía de buenas prácticas a la hora de desarrollar el *front-end*. Cuenta además con un gran soporte por parte de la comunidad y una gran integración con diferentes librerías de *Javascript*, lo cual es un punto clave para el correcto funcionamiento e integración de la parte *front-end* con la parte *back-end* de glUCModel 2.0.

El uso de *Bootstrap* incluye el uso de tecnologías web, en este caso *HTML 5*, *CSS 3* y *Javascript* (en concreto *jQuery*).

El *IDE* (entorno de desarrollo integrado) elegido fue *Eclipse* [16], en concreto la versión *Eclipse Neon 3*. Existen otros entornos de desarrollo apropiados para el desarrollo de una aplicación con estas características, pero se eligió *Eclipse* por la familiaridad y el conocimiento previo que existía del mismo en otros proyectos a lo largo de los años de la carrera en entornos *Java*, lenguaje elegido para la parte *back-end* de *glUCModel 2.0*. *Eclipse* permite la creación de proyectos web, facilitando un esquema predefinido de los directorios necesarios para el correcto funcionamiento de un proyecto de esas características.

#### 4.2.2 Utilización de *Bootstrap*

A la de hora de utilizar *Bootstrap* tenemos la posibilidad de utilizar directamente el *framework* que se proporciona, o bien utilizar alguna de las muchas versiones modificadas por distintas empresas o usuarios, estas versiones modificadas y listas para su uso reciben el nombre de *templates*.

En el desarrollo de *glUCModel 2.0* se utilizan dos *templates* distintos, *AdminLTE* [17] y *Material Kit* [18].

El primero de ellos, *AdminLTE*, es un *template* centrado en sistemas de gestión o administración, su uso en *glUCModel 2.0* ha ido orientado al menú principal de la aplicación ya que proporciona tanto un diseño cómodo a la hora de desarrollar como agradable en su uso por parte del usuario.

El segundo, *Material Kit*, es el *template* que se ha empleado en el desarrollo del resto de la aplicación. *Material Kit* proporciona un diseño agradable, llamativo y moderno en todos los elementos de *Bootstrap* que modifica, añadiendo animaciones, colores o transiciones en elementos que habitualmente poseen un diseño más plano.

Recorriendo todas las interfaces puede observarse el efecto que produce en elementos como:

- Diseño de la página de iniciar sesión, botones, *checkboxes*, *inputs* de formularios.

- Pestañas de navegación (*tabs*), carrusel de gráficas en el perfil del paciente, ventanas modales (*modals*), paginación en el listado de pacientes.

Además del ya mencionado objetivo de conseguir un diseño adaptativo, se ha querido dotar a la aplicación de una navegación ágil y cómoda, es aquí donde la navegación mediante pestañas (*nav pills*) tiene un papel a destacar.

En el diseño original de glUCModel 2.0 dos de las grandes secciones (recomendador general y gestión de pacientes) no se encontraban en el menú principal como sucede en la versión final, sino que estaban repartidas en dos pestañas que proporcionaban una navegación y una transición cómoda y rápida entre ambas secciones. Sin embargo, se descartó debido al incremento de elementos principales al incluir la sección de gestión de documentos (aunque finalmente no fue implementada) y de educación para la diabetes.

Las pestañas siguen presentes en el diseño final de glUCModel 2.0, encontrándose dentro de la sección gestión de pacientes, permitiendo alternar entre el listado de pacientes y el formulario de alta para nuevos pacientes (ver sección 3.4.3) sin necesidad de cargas ni esperas adicionales.

### 4.2.3 Otras tecnologías del *front-end*

En este capítulo se describen el resto de tecnologías y/o librerías utilizadas durante el desarrollo de la parte *front-end*.

#### 4.2.3.1 *jQuery* y *AJAX*

Para poder interactuar con los elementos de la interfaz de usuario es necesario el uso de la tecnología *Javascript*, en concreto en glUCModel 2.0 se utiliza el *framework* denominado *jQuery* [19], el cual es software libre y de código abierto.

El uso de *jQuery* permite la realización de acciones que evitan sobrecargar el servidor, como es el caso de la validación de formularios antes de procesar su envío como sucede en la página de iniciar sesión y en el formulario de alta de nuevos pacientes.

El uso combinado de *jQuery* con la tecnología *AJAX* permite una interacción dinámica y una experiencia de uso agradable y fluida. Esto es posible gracias al uso de *AJAX*, el cual permite la modificación y actualización de partes de la

interfaz sin necesidad de la recarga completa de la página.

En glUCModel 2.0 pueden observarse estas ventajas en distintas interfaces:

- Buscador de pacientes.
- Paginación en el listado de pacientes.
- Añadir soluciones nuevas tanto en la página de inicio como en las recomendaciones manuales del recomendador del paciente.
- Creación de una nueva recomendación manual en el recomendador del paciente.
- Envío de recomendaciones arrastrando las tarjetas en el recomendador del paciente.

#### 4.2.3.1 *jQuery UI y modals*

Para el desarrollo de varios elementos de la parte *front-end* se ha hecho uso de *jQuery UI* [20]. *jQuery UI* es una biblioteca de componentes para el *framework jQuery* que le añaden un conjunto de *plug-ins*, *widgets* y efectos visuales para la creación de aplicaciones web.

En concreto se han utilizado distintos tipos de interacciones y *widgets*:

- *Draggable*: Permite que un elemento pueda ser arrastrado. [21]
- *Droppable*: Permite que un elemento responda a elementos arrastrables. [22]
- *Datepicker*: Representación de calendario gráfico. [23]

Las interacciones *Draggable* y *Droppable* se han utilizado en el desarrollo de las columnas de recomendaciones en la interfaz del recomendador del paciente.

Las tarjetas de recomendaciones sugeridas y manuales son los elementos *draggables*, es decir, elementos arrastrables por la interfaz del recomendador del paciente. Su movimiento está limitado a poder moverse en el eje de desplazamiento X (eje Y si estamos en la versión móvil) para evitar comportamiento indeterminados.

La columna de recomendaciones enviadas es el elemento *droppable*, es decir, admite los elementos *draggables*, añadiéndolos a su lista y produciendo la acción de enviar la recomendación al paciente. Los elementos *draggables* solo pueden ser admitidos por la columna de recomendaciones enviadas, regresando a su posición original de no ser arrastrados hacia la misma, esto al igual que la limitación de los ejes de desplazamiento, está hecho con la intención de evitar inconsistencias y

comportamientos indeterminados en la interfaz.

Su uso e integración resultan sencillos siguiendo la documentación proporcionada por la página oficial de *jQuery UI*, sin embargo, se presentaron varios problemas para usar estos componentes junto con *Bootstrap* ya que presentaban incompatibilidades.

Para ello hubo que maquetar esa interfaz prescindiendo de *Bootstrap* en algunos elementos como las columnas para que las interacciones *draggable* y *droppable* funcionasen sin problemas, teniendo que hacer uso de *media queries* en los documentos *CSS* para que se siguiese respetando el diseño adaptativo de *glUCModel 2.0*.

El *widget DatePicker* se ha utilizado en el formulario de alta para nuevos pacientes, su uso proporciona una interacción más sencilla e intuitiva en los *inputs* que requieran la introducción de fechas, como es el caso de la introducción de la fecha de nacimiento y la fecha del consentimiento para ser dado de alta en el sistema.

Para cerrar este capítulo se detalla el uso de las ventanas modales (*modals*) en *glUCModel 2.0*. Estas ventanas modales se utilizan para visualizar la información completa de las recomendaciones.

Esta forma de visualización permite presentar esta información sin tener que crear una interfaz aparte y sin tener que recargar la página cada vez que quiera accederse a una recomendación concreta. Esto último produciría una mala experiencia de uso debido a que pueden llegar a acumularse un gran número de tarjetas, convirtiendo la interacción con estas en algo tedioso y tosco.

### 4.3 Implementación del *back-end*

En esta sección se explica en detalle la implementación de la parte *back-end* de *glUCModel 2.0*, hablando de como se ha estructurado y de las tecnologías empleadas para ello.

Tal y como se ve en la figura 54, es una arquitectura *cliente-servidor* basada en servicios web (*Web services*). En este capítulo se detalla el funcionamiento interno de esta arquitectura, los distintos servicios web, el uso y funcionalidades del sistema recomendador, la implementación de la seguridad, así como el diseño de la base de datos usada en la aplicación.

### 4.3.1 Arquitectura de la aplicación

El *back-end* de glUCModel está desarrollado en el lenguaje *Java* utilizando la tecnología *JSP* (JavaServer Pages), la elección de este lenguaje no es casual, ya que responde a la necesidad de poder integrar el sistema recomendador, el cual también está desarrollado con el lenguaje *Java*.

En el *back-end* encontramos servicios proporcionados por los *Java Servlets*. Los *servlets* son clases escritas en el lenguaje *Java* utilizadas para ampliar las capacidades de respuesta a la parte cliente o *front-end*.

Las figuras 55 y 56 explican el proceso interno que se sigue a la hora de utilizar un *Java Servlet* como servicio web y como está estructurado internamente.

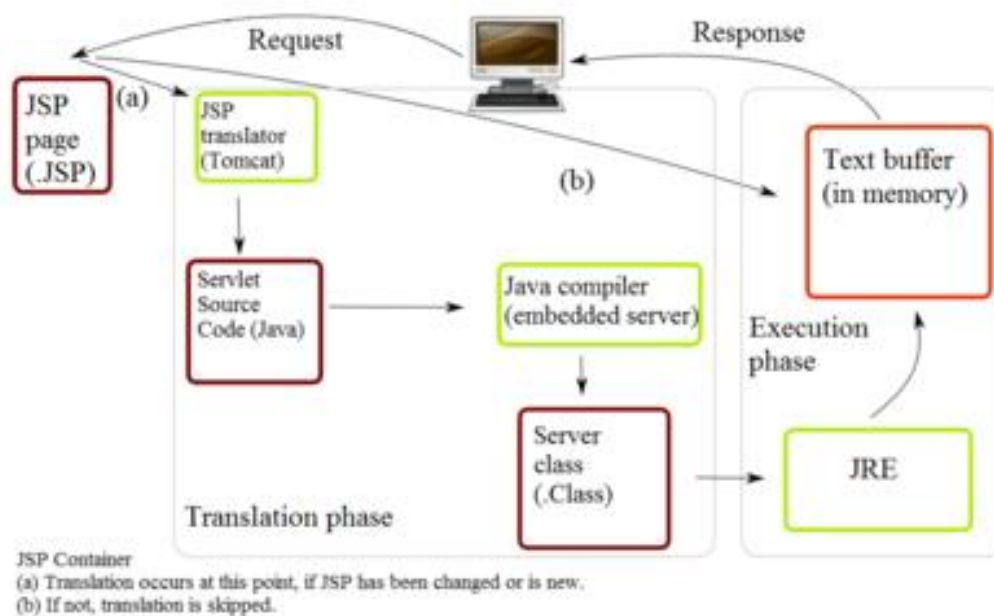


Figura 55: Funcionamiento llamadas Java Servlets

```

14 import java.io.IOException;
15 @WebServlet("/cargaSoluciones/*")
16 public class SolucionesServlet extends HttpServlet{
17
18     private static final long serialVersionUID = 1L;
19
20     private DatabaseRecieve db;
21
22     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
23
24         db = new DatabaseRecieve();
25
26         List<Solucionrec> lista = db.cargaSoluciones();
27
28         String json = new Gson().toJson(lista);
29         System.out.println(json);
30
31         response.setContentType("application/json");
32         response.setCharacterEncoding("UTF-8");
33
34         response.getWriter().write(json);
35     }
36 }
37

```

Figura 56: Estructura de los Java Servlets

```

$.get("cargaSoluciones",function(responseJson){
    rellenalistaSoluciones(responseJson);
}).fail(function () {
    ($.alert).show();
});

```

Figura 57: Ejemplo de petición AJAX

Desde las páginas JSP se hacen llamadas a los servicios implementados en los servicios web mediante el uso de peticiones *AJAX* como muestra la figura 57.

En el desarrollo de los servicios web de *glUCModel 2.0* se ha hecho uso de anotaciones que facilitan la creación de servicios web. Estas anotaciones pertenecen a *JAX-WS* [24] (Java API for XML Web Services). *JAX-WS* es una interfaz de programación de aplicaciones (*API*) de Java en Extensible Markup

Language (XML) para la creación de servicios web (*WS*).

Dentro de estos *servlets* encontramos en primer lugar una anotación *WebServlet*, esta anotación es la que hace posible que la llamada a *cargaSoluciones* desde la página *JSP* termine por ejecutar correctamente las funciones implementadas en el *servlet*.

La clase del *servlet* extiende de la clase *HttpServlet*, al extender de esta clase es necesario definir el método *doGet* o *doPost*, dependiendo de si se ha realizado una petición *HTTP GET* o *HTTP POST*, para responder la petición. Este método recibe los parámetros dados por el cliente a través de la clase *HttpServletRequest* y encapsula la respuesta que se le dará al cliente a través de la clase *HttpServletResponse*.

Desde dentro de estos métodos se hace uso de los recursos del sistema, esto es, la información almacenada en la base de datos necesaria para completar la petición *AJAX*. Una vez obtenida la información se hace uso de la librería *Gson* [25] (también conocido como Google *Gson*) para poder devolver la respuesta en formato *JSON*, se eligió este tipo de formato debido a la facilidad de uso y manipulación una vez la respuesta se recibe en la página *JSP*.

*Gson* es una biblioteca de código abierto para el lenguaje de programación *Java* que permite la serialización y deserialización entre objetos *Java* y su representación en notación *JSON*.

Para que el sistema descrito arriba pueda funcionar correctamente, es necesario la configuración de varios servicios dentro del entorno de desarrollo *Eclipse*.

Para el desarrollo de *glUCModel 2.0* se ha hecho uso del contenedor de *servlets* *Tomcat* [26] en su versión 8.5, de la integración con *Maven* [27] y de la herramienta *Hibernate* [28].

El uso de *Tomcat* era necesario para dar soporte al uso de las páginas *JSP* y de los *Java Servlet* como servicios web.

*Maven* es una herramienta para la gestión y construcción de proyectos *Java*, su uso ha agilizado mucho el desarrollo debido a la gestión y descarga automática que realiza de las dependencias de las librerías utilizadas en el proyecto, utilizando para ello el fichero *pom.xml*.

Con *Hibernate* se gestiona y se realiza el mapeo de las bases de datos, generando

de forma automática las clases *Java* y los ficheros *hbm.xml* necesarios para el funcionamiento del acceso a la base de datos desde el *back-end*. Al usar *Hibernate* las consultas a la base de datos se realizan mediante el lenguaje *HQL* (Hibernate Query Language) en lugar de consultas SQL, su particularidad es que las consultas se realizan sobre los objetos java y no sobre las tablas y columnas que forman parte de la base de datos.

### 4.3.2 Servicios web

En esta sección se enumeran los servicios web utilizados en glUCModel 2.0, detallando la funcionalidad que posee cada uno de ellos.

Los servicios web relacionados con los usuarios son los siguientes:

- Login del médico: Comprueba la validez del correo y la contraseña asociados al médico.
- Logout del médico: Cierra la sesión del médico y destruye las variables de sesión asociadas a la misma.
- Listado de pacientes: Carga el listado de todos los pacientes asociados al médico.
- Cargar número de pacientes: Carga el número de pacientes para poder implementar de manera correcta el paginador.
- Búsqueda de pacientes: Muestra los resultados de la búsqueda de pacientes por nombre.
- Cargar datos del paciente: Proporciona información del paciente: nombre y apellidos, edad, altura, peso más reciente y fecha de registro en el sistema.
- Cargar pesos: Carga los datos del peso del paciente a lo largo del último año.
- Cargar glucemias: Carga los datos de las glucemias del paciente a lo largo del día.
- Cargar insulinas: Carga los datos de las glucemias del paciente a lo largo del día.
- Cargar datos del médico: Proporciona información del médico: nombre y apellidos, correo electrónico asociado a glUCModel 2.0 y número de colegiado.
- Dar de alta a un paciente: Añade un nuevo paciente al sistema, asociado al médico que ha rellenado el formulario de alta.

Los servicios web relacionados con las recomendaciones son los siguientes:

- Cargar recomendaciones página de inicio: Carga la información de las tarjetas de las recomendaciones de la página de inicio.
- Cargar datos iniciales recomendador paciente: Carga la información inicial necesaria para mostrar el recomendador personalizado del paciente.
- Cargar recomendaciones manuales recomendador paciente: Carga la información de las tarjetas de las recomendaciones sin enviar del recomendador del paciente.
- Cargar recomendaciones enviadas recomendador paciente: Carga la información de las tarjetas de las recomendaciones enviadas del recomendador del paciente.
- Cargar recomendaciones sugeridas recomendador paciente: Haciendo uso del recomendador, carga la información de las tarjetas de las recomendaciones sugeridas del recomendador del paciente.
- Cargar soluciones: Carga las soluciones de los formularios de creación de recomendaciones, tanto en la página de inicio como en el recomendador del paciente.
- Añadir recomendación manual recomendador paciente: Genera y carga una nueva recomendación manual en el recomendador del paciente.
- Añadir recomendación manual página de inicio: Genera y carga una nueva recomendación en la página de inicio.
- Añadir nueva solución: Añade una nueva solución a los formularios de creación de recomendaciones, tanto en la página de inicio como en el recomendador del paciente.
- Enviar recomendación: Envía la recomendación al paciente, tanto en la página de inicio como en el recomendador del paciente.

### 4.3.3 Sistema recomendador

En la versión final de glUCModel 2.0 se usa un recomendador basado en casos (CBR) usando una implementación desarrollada de forma manual sin usar ningún framework o proyecto externo, desarrollando los mismos algoritmos de recomendación que poseen otros sistemas recomendadores estudiados.

El sistema recomendador tiene dos partes diferenciadas, por un lado, la descripción y por otro la solución. La descripción corresponde a valores de Hiperglucemias e

Hipoglucemias en distintos momentos del día, la solución es un texto perteneciente a un listado de tareas a realizar por el paciente, entre las que se incluyen el estudio de temas relacionados con la diabetes, distintos tipos de ejercicios o dietas a seguir.

El sistema recomendador de glUCModel 2.0 se basa en las similitudes entre los pacientes registrados en el sistema. El criterio de comparación para esta similitud usa los valores de Hiperglucemias e Hipoglucemias en distintos momentos del día (descripción de la recomendación).

Por una parte, se obtienen los valores de la tabla *mediciones* de la base de datos, de Hiperglucemias e Hipoglucemias del paciente del que quiere obtenerse las recomendaciones sugeridas, y se comparan por medio de operaciones matemáticas con los valores de la tabla *descripcionrec* de la base de datos, de Hiperglucemias e Hipoglucemias del resto de pacientes registrados en el sistema.

Estas operaciones matemáticas devuelven unos valores numéricos entre 0 y 1, las recomendaciones de otros pacientes que tengan un valor numérico mayor o igual a 0.5 se toman como recomendaciones válidas para ese paciente concreto, pudiéndose aplicar las soluciones que tuviesen asignadas.

Se descartan las recomendaciones que obtengan un valor numérico menor a 0.5 por no tener el grado de similitud óptimo, también se descartan las recomendaciones que obtengan un valor numérico igual a 1 ya que eso indica que esa recomendación es propia de ese paciente.

El sistema recomendador posee distintas fases de evolución:

- Fase de recomendador inicial: En esta fase el sistema no posee recomendaciones, por lo que es tarea del médico empezar a realizar recomendaciones manuales para cada uno de sus pacientes.
- Fase de recomendador sencillo: En esta fase el sistema posee un número limitado de recomendaciones, puede empezar a sugerir recomendaciones, aunque la cantidad y la precisión de las mismas no es la idónea.
- Fase de recomendador final: En esta última fase el sistema posee un gran número de recomendaciones, puede sugerir un gran número de recomendaciones y con una precisión muy ajustada. El sistema recomendador está listo para su funcionamiento.

### 4.3.4 Implementación de la seguridad

En el desarrollo de glUCModel 2.0 la seguridad de los datos tiene mucha importancia, dada la delicada naturaleza de los mismos al tratarse de datos personales y médicos de personas.

#### 4.3.4.1 Seguridad de las cuentas de usuario

Las contraseñas de los médicos y pacientes no se almacenan en texto plano en la base de datos, se les aplica un cifrado de tipo *SHA1* para evitar posibles accesos no autorizados. Se hace uso de la librería *Apache Commons Codec 1.10* [29], en concreto el método *sha1Hex* de la clase *DigestUtils*.

#### 4.3.4.2 Seguridad en el envío de datos

A la hora de acceder a secciones como el perfil del paciente o el recomendador del paciente, se hace uso de métodos *POST* en lugar de usar métodos *GET*.

Esto impide la exposición de datos sensibles en las *URLs*, evitando su visualización y manipulación, que podría dar lugar a accesos no permitidos a datos sensibles de los pacientes.

#### 4.3.4.3 Seguridad de acceso a secciones privadas

Cuando se quiere acceder a secciones privadas de la aplicación primero se comprueba que se haya producido un inicio de sesión válido, haciendo uso de las variables de sesión creadas cuando se produce con éxito el *login* del médico, en caso contrario se redirige a la página de inicio de sesión.

Esta medida se toma como prevención de accesos no autorizados en caso de conocerse las *URLs* que cargan páginas con contenido sensibles como los perfiles o el recomendador del paciente.

#### 4.3.4.4 Seguridad en acceso a la base de datos

Al realizarse consultas a la base de datos mediante el lenguaje *HQL* pueden producirse los ataques denominados inyección de *SQL*, los cuales tienen el objetivo de acceder a las tablas de la base de datos y poder robar la información que contienen.

Para evitar este tipo de situaciones se hace uso de los marcadores con nombre que proporciona *Hibernate*, haciendo que las consultas que reciben parámetros externos puedan realizarse en condiciones óptimas de seguridad.

### 4.3.5 Base de datos

La base de datos utilizada en glUCModel 2.0 es una base de datos relacional MySQL.

A continuación, se describen las tablas utilizadas:

- Tabla usuarios: Contiene la información personal tanto de los pacientes como de los médicos, entre la que se incluyen nombre y apellidos, DNI, fecha de nacimiento, altura...entre otros.
- Tabla pesos: Contiene información de los pesos de los pacientes en distintas fechas del año.
- Tabla glucemias: Contiene información de los valores de glucemia de los pacientes en distintos momentos del día.
- Tabla insulinas: Contiene información de los valores de insulina de los pacientes en distintos momentos del día.
- Tabla pruebas: Contiene información sobre las pruebas que se han realizado los pacientes en una fecha en concreto
- Tabla tipo pru: Contiene el listado de las posibles pruebas que ha podido realizarse un paciente.
- Tabla mediciones: Contiene los valores más recientes de Hiperglucemias e Hipoglucemias a lo largo del día de pacientes concretos, identificados mediante el id del paciente.
- Tabla descripcionrec: Contiene los valores más recientes de Hiperglucemias e Hipoglucemias a lo largo del día de todos los pacientes del sistema, sin estar identificados por el id del paciente.
- Tabla solucionrec: Contiene el listado de tareas que se les enviará a los pacientes en sus recomendaciones.
- Tabla actividadesrec: Contiene las actividades de una recomendación.
- Tabla recomendaciones: Contiene información sobre las descripciones y soluciones de las recomendaciones, así como la fecha y valores para saber si ha sido enviada al paciente y si ha sido leída por el mismo.

# Capítulo 5. CONCLUSIONES y TRABAJO FUTURO

## 5.1 Conclusiones

Al iniciarse el desarrollo de este TFG se establecieron unos objetivos y tareas principales necesarias para poder desarrollar satisfactoriamente la aplicación. Los objetivos y tareas principales que se alcanzaron al finalizar el proyecto fueron los siguientes:

- Se investigó y se estudió distintos sistemas recomendadores con el objetivo de entender su funcionamiento para una integración satisfactoria en el proyecto.
- Se estudiaron nuevas tecnologías y metodologías necesarias para la posterior implementación de las partes *back-end* y *front-end*.
- Se diseñó e implementó una arquitectura completa y funcional para el *back-end* basada en servicios web.
- Se diseñó e implementó una interfaz para la parte *front-end* usando tecnologías como *Bootstrap* haciendo posible el diseño adaptativo de la misma.
- Se integraron las tecnologías usadas en el *front-end* y *back-end* con el recomendador para dar forma a la aplicación.

Hubo dos puntos pertenecientes a los objetivos principales que no pudieron cumplirse y que forman parte del trabajo futuro de este proyecto:

- Evaluar la validez de la aplicación haciendo uso de datos médicos de pacientes reales.
- Publicar el proyecto en un servidor online de la facultad para darle la mayor visibilidad posible.

Para lograr los objetivos principales descritos arriba se fueron cumpliendo una serie de requisitos a lo largo del tiempo:

- Se estudiaron las particularidades y se aprendió a desarrollar una aplicación usando *JSP* y *Java Servlets* en la parte *back-end*.
- Se aprendió a diseñar e implementar una arquitectura basada en servicios web, como contrapartida de las arquitecturas *MVC*, que eran las únicas que se manejaban con soltura antes de empezar el proyecto.

- Se aprendió a maquetar de forma correcta una aplicación con diseño adaptativo utilizando *Bootstrap*.
- Se aprendió a utilizar elementos del *front-end* con interacciones complejas como *draggable* y *droppable* para dar forma a la interfaz del recomendador de pacientes.
- Se aprendió a poner en marcha un proyecto web con *Eclipse*, configurando *Maven* y *Tomcat* para el correcto funcionamiento del mismo.
- Se aprendió a manejar la tecnología *Hibernate* y las consultas *HQL* para el correcto mapeo y uso de la base de datos.
- Se aprendió a integrar un sistema recomendador con la parte *front-end* y la parte *back-end* para la realización exitosa del proyecto *glUCModel 2.0*.

Durante la realización del proyecto fueron surgiendo problemas y dificultades a la hora de desarrollar la aplicación.

Al comienzo del mismo se quiso aprovechar una versión antigua procedente del curso anterior, pero arreglar la configuración defectuosa de ese proyecto llevó más tiempo del deseado. Finalmente se desechó ya que además la interfaz que poseía ese proyecto no respetaba las líneas de diseño adaptativo.

Surgieron problemas a la hora de implementar los elementos *draggables* y *droppables*, ya que su configuración y puesta a punto no era familiar y no resultaba sencilla, así como sus incompatibilidades con la maquetación con *Bootstrap*.

Con el tiempo se fueron manejando con más soltura, y el problema con *Bootstrap* se solucionó haciendo la maquetación de forma manual y usando *media queries* para seguir respetando el diseño adaptativo de la aplicación.

Finalmente, se pudieron cumplir la gran mayoría de objetivos y tareas principales durante el desarrollo de *glUCModel 2.0*, logrando tener una versión estable de la aplicación con todas las funcionalidades correctamente integradas.

El desarrollo de *glUCModel 2.0* ha supuesto una satisfacción tanto personal como académica. Se pudo sacar delante de forma individual un proyecto que en varios momentos presentó muchos problemas, aprendiendo por el camino sobre nuevas tecnologías y formas de trabajar con aplicaciones web, y lo más importante, haciendo posible una aplicación médica que en el futuro pueda ayudar y mejorar la calidad de vida de las personas que sufren enfermedades como la diabetes.

## 5.2 Conclusions

At the beginning of the development of this TFG, the main objectives and tasks were established necessary to be able to successfully develop the application. The main objectives and task achieved at the end of the project were:

- Different recommender systems were investigated and studied to understand their operation for a successful integration into the project.
- New technologies and methodologies necessary for the subsequent implementation of the *back-end* and *front-end* parts were studied.
- A complete and functional architecture for the web-based *back-end* was designed and implemented.
- An interface for the *front-end* was designed and implemented using technologies such as *Bootstrap* making it possible to design adaptive interfaces.
- The technologies used in the *front-end* and *back-end* were integrated with the recommender to finalize the application.

There were two points pertaining to the main objectives that could not be fulfilled and that are part of the future work of this project:

- Evaluate the validity of the application using medical data from actual patients.
- Publish the project on an online server to give it the highest possible visibility.

To achieve the main objectives described above, a series of requirements were fulfilled over time:

- I studied the peculiarities and learned to develop an application using JSP and Java Servlets in the back end.
- I learned how to design and implement a web-services-based architecture, as a counterpart to *MVC* architectures, which were the only ones that were handled with ease before starting the project.
- I learned how to properly model a responsive application using *Bootstrap*.
- I learned how to use *front-end* elements with complex interactions such as *draggable* and *droppable* to shape the patient referral interface.

- I learned how to start a web project with *Eclipse*, configuring *Maven* and *Tomcat* for proper operation.
- I learned how to handle *Hibernate* technology and *HQL* queries for the correct mapping and database usage.
- I learned to integrate a recommender system with the *front-end* and the *back-end* for the successful completion of the glUCModel 2.0 project.

During the implementation of the project problems and difficulties arose when developing the application.

At the beginning of it was wanted to take advantage of an old version from the previous course, but fixing the defective configuration of that project took more time than desired. Finally, it was discarded since in addition the interface that owned that project did not respect the lines of adaptive design.

Problems arose when implementing *draggables* and *droppables*, since their configuration and set-up was not familiar and not simple, as well as their incompatibility with the layout with *Bootstrap*.

Over time I could handle it more easily, and the problem with *Bootstrap* was solved by doing the layout manually and using *media queries* to continue to respect the adaptive design of the application.

Finally, it was possible to fulfill most main objectives and tasks during the development of glUCModel 2.0, achieving a stable version of the application with all the functionalities correctly integrated.

The development of glUCModel 2.0 has been a personal and academic satisfaction. It was possible to take individually a project that at various times presented many problems, learning along the way on new technologies and ways of working with web applications, and most importantly, making possible a medical application that in the future can help and improve the quality of life of people suffering from diseases such as diabetes.

## 5.3 Trabajo futuro

Durante el desarrollo de este TFG se han usado distintos tipos de tecnologías, algunas como *JSP* o el tener que trabajar con una arquitectura de servicios web (*Web service*), cuando no se tenía experiencia previa en ninguna de las anteriores.

Esto supuso un tiempo extra de aprendizaje, lo cual añadido al estudio e investigación sobre sistemas recomendadores, hizo que por motivos de tiempo se fuesen descartando funcionalidades que no se consideraban imprescindibles y centrándose en los aspectos fundamentales de *glUCModel 2.0*.

En los siguientes puntos se describen posible mejoras y nuevas funcionalidades a añadir en un futuro a la aplicación *glUCModel 2.0*:

- Configuración de un servidor online donde poder publicar el *WAR* generado a partir del proyecto, dotando de mayor visibilidad al mismo.
- Implementación e integración de la interfaz del paciente.
- Añadir una descripción en la página de inicio de sesión donde se hable sobre lo que es *glUCModel 2.0*.
- Permitir que las tarjetas de recomendaciones sean más interactivas, dando la posibilidad de añadir comentarios, que se mostrarían en el *timeline* de actividades.
- Poder descartar recomendaciones sugeridas por el sistema, así como poder editarlas para ajustarlas según el criterio del médico.
- Poder eliminar recomendaciones manuales ya creadas, así como la posibilidad de poder editarlas.
- En el perfil del paciente, poder elegir cuales son los intervalos de tiempo que se quieren observar para las distintas variables bajo seguimiento.
- Implementar la interfaz de gestión de documentos, donde poder visualizar y subir nuevos documentos complementarios a las recomendaciones.
- Implementar el botón de adjuntar archivos tanto en el formulario de la página de inicio como en el recomendador del paciente.
- Integrar el Moodle de *glUCModel* dentro de la aplicación web.
- Creación de un panel de administración para poder gestionar distintos datos directamente desde la aplicación.

- Poder elegir distintos filtros a la hora de mostrar el listado de paciente, no solo por fecha de registro como está actualmente implementado, pudiendo elegir filtrado por peso, edad, etc.
- Portear glUCModel 2.0 a *Android* e *iOS*, permitiendo un mejor ajuste o rediseño de algunas interfaces centrándolas solo para dispositivos móviles, así como integrando servicios nativos de *smartphones* como notificaciones de distintos eventos (aviso de nuevas recomendaciones, aviso de análisis cercanos, etc.).
- En el perfil médico, dar la posibilidad de actualizar datos como el correo electrónico con el que está dado de alta en el sistema, así como poder modificar su contraseña.

## REFERENCIAS

- [1] gIUCModel [En línea] <http://glucmodel.dacya.ucm.es/testing/>
- [2] José Ignacio Hidalgo, Esther Maqueda, José L. Risco-Martín, Alfredo Cuesta-Infante, J. Manuel Colmenar, Javier Nobel: gIUCModel: A monitoring and modeling system for chronic diseases applied to diabetes. *Journal of Biomedical Informatics* 48: 183-192 (2014)
- [3] La diabetes en España [En línea]  
<http://www.fundaciondiabetes.org/prensa/297/la-diabetes-en-espana>
- [4] Una aplicación para medir tu glucosa [En línea] <http://www.naturarla.es/una-aplicacion-movil-para-medir-tu-nivel-de-glucosa>
- [5] Sistema de control de glucosa en sangre GMate [En línea]  
[http://www.gmv.com/blog\\_gmv/normativa-sobre-apps-medicas-moviles-parte-3-ejemplo-ii-sistema-de-control-de-glucosa-en-sangre-gmate/](http://www.gmv.com/blog_gmv/normativa-sobre-apps-medicas-moviles-parte-3-ejemplo-ii-sistema-de-control-de-glucosa-en-sangre-gmate/)
- [6] Sistema de control de glucosa en sangre Accu-Check [En línea]  
[http://www.gmv.com/blog\\_gmv/normativa-sobre-apps-medicas-moviles-parte-3-ejemplo-iii-accu-chek/](http://www.gmv.com/blog_gmv/normativa-sobre-apps-medicas-moviles-parte-3-ejemplo-iii-accu-chek/)
- [7] ¿En que puede sustituir una aplicación de móvil a su médico? [En línea]  
[http://elpais.com/elpais/2015/05/21/buenavida/1432198163\\_619344.html](http://elpais.com/elpais/2015/05/21/buenavida/1432198163_619344.html)
- [8] Encuesta iDoctus [En línea]  
[http://www.semg.es/images/stories/recursos/2014/variados/habitos\\_digitaless\\_medicos\\_2014.pdf](http://www.semg.es/images/stories/recursos/2014/variados/habitos_digitaless_medicos_2014.pdf)
- [9] Informe IPSOS [En línea] <http://www.ipsos.es/socialogue>
- [10] Los médicos españoles ya “recetan” aplicaciones de salud [En línea]  
<http://www.elmundo.es/salud/2013/12/10/52a5ff800ab740a5768b45a6.html>

[11] Dietmar Jannach, Markus Zanker, Alexander Felfernig, Gerhard Friedrich, *Recommender Systems, An Introduction*. New York: Cambridge University Press, 2012

[12] Recommender system [En línea]

[https://en.wikipedia.org/wiki/Recommender\\_system#Content-based\\_filtering](https://en.wikipedia.org/wiki/Recommender_system#Content-based_filtering)

[13] Justinmind: Prototyping tool for web and mobile apps [En línea] <https://www.justinmind.com/>

[//www.justinmind.com/](https://www.justinmind.com/)

[14] Kanban [En línea] [https://es.wikipedia.org/wiki/Kanban\\_\(desarrollo\)](https://es.wikipedia.org/wiki/Kanban_(desarrollo))

[15] Bootstrap [En línea] [getbootstrap.com](http://getbootstrap.com)

[16] Eclipse [En línea] <https://eclipse.org/>

[17] AdminLTE [En línea] <https://adminlte.io/themes/AdminLTE/index2.html>

[18] Material Kit [En línea] <http://demos.creative-tim.com/material-kit/index.html>

[19] jQuery [En línea] <https://jquery.com/>

[20] jQuery UI [En línea] <https://jqueryui.com/>

[21] jQuery UI Droppable [En línea] <http://jqueryui.com/droppable/#photo-manager>

[22] jQuery UI Draggable [En línea] <http://jqueryui.com/draggable/>

[23] jQuery UI Datepicker [En línea] <http://jqueryui.com/datepicker/>

[24] JAX-WS [En línea] <https://es.wikipedia.org/wiki/JAX-WS>

[25] Gson [En línea] <https://github.com/google/gson>

[26] Apache Tomcat [En línea] <http://tomcat.apache.org/>

[27] Maven [En línea] <https://maven.apache.org/>

[28] Hibernate [En línea] <http://hibernate.org/>

[29] Apache Commons Codec 1.10 [En línea]  
<https://mvnrepository.com/artifact/commons-codec/commons-codec/1.10>

[30] w3schools [En línea] <https://www.w3schools.com/>

[31] Tutorial Java-web Servlet y JSP [En línea]  
<http://www.edu4java.com/es/servlet/servlet.html>

[32] Sesiones en Servlets y JSP. Ejemplo de login y logout [En línea]  
<http://www.galisteocantero.com/sesiones-en-servlets-y-jsp-ejemplo-login-logout/>

[33] How to generate Hibernate mapping files & annotation with Hibernate Tools  
[En línea] <https://www.mkyong.com/hibernate/how-to-generate-code-with-hibernate-tools/>

[34] jCOLIBRI 2 [En línea] <http://gaia.fdi.ucm.es/research/colibri/jcolibri>

[35] Highcharts [En línea] <https://www.highcharts.com/>