

GOODDIABETICLIFE, UNA APLICACIÓN WEB
PARA EL SEGUIMIENTO DE PACIENTES
DIABÉTICOS

GOODDIABETICLIFE, A WEB APPLICATION FOR
MONITORING DIABETIC PATIENTS



TRABAJO FIN DE GRADO
CURSO 2023-2024

AUTOR
BEATRIZ ÁLVAREZ DE ARRIBA

DIRECTOR
RAMÓN GONZÁLEZ DEL CAMPO RODRÍGUEZ BARBERO

GRADO EN INGENIERÍA DE SOFTWARE
FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID

GOODDIABETICLIFE, UNA APLICACIÓN WEB
PARA EL SEGUIMIENTO DE PACIENTES
DIABÉTICOS

GOODDIABETICLIFE, A WEB APPLICATION FOR
MONITORING DIABETIC PATIENTS

TRABAJO DE FIN DE GRADO EN INGENIERÍA DE SOFTWARE

AUTOR

BEATRIZ ÁLVAREZ DE ARRIBA

DIRECTOR

RAMÓN GONZÁLEZ DEL CAMPO RODRÍGUEZ BARBERO

CONVOCATORIA: JUNIO 2024

GRADO EN INGENIERÍA DE SOFTWARE
FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID

27 DE MAYO DE 2024

RESUMEN

GoodDiabeticLife, una aplicación web para el seguimiento de pacientes diabéticos.

La diabetes es una enfermedad bastante común que requiere especial atención en el día a día de los pacientes. Por esta razón, este trabajo consiste en el desarrollo de una aplicación web para ayudar a las personas que sufren esta enfermedad.

La aplicación permite a los pacientes registrar sus comidas, niveles de glucosa y actividad física. Con estos datos, la aplicación orientará a los pacientes basándose en la información registrada y ayudándose de las pautas acordadas con sus médicos. Además, la aplicación facilitará la comunicación entre pacientes y médicos, permitiendo que los médicos accedan a la información registrada por sus pacientes y puedan añadir comentarios para que el paciente mejore. Tanto pacientes como médicos podrán visualizar gráficas que muestran la evolución de la glucemia en un periodo de tiempo.

Con esta aplicación, se busca que los pacientes tengan una herramienta fácil de usar que les ayude en su día a día y mejore la relación médico-paciente.

Palabras clave

Aplicación web, diabetes, glucosa, paciente, médico.

ABSTRACT

GoodDiabeticLife, a web application for monitoring diabetic patients.

Diabetes is a fairly common disease that requires special attention in the daily lives of patients. For this reason, this project consists of developing a web application to help people who suffer from this disease.

The web application allows patients to record their meals, glucose levels, and physical activity. With this data, the application will guide patients based on the information recorded and the guidelines agreed upon with their doctors. Additionally, the application will facilitate communication between patients and doctors, allowing doctors to access the information recorded by their patients and add comments to help improve the patient's condition. Both patients and doctors will be able to view graphs that show the evolution of blood glucose over a period of time.

With this application, the goal is to provide patients with an easy-to-use tool that helps them in their daily lives and improves the doctor-patient relationship.

Keywords

Web application, diabetes, glucose, patient, doctor.

ÍNDICE DE CONTENIDOS

Capítulo 1 - Introducción	1
1.1 Motivación	1
1.2 Objetivos.....	2
1.3 Plan de trabajo.....	2
1.4 Estructura de la memoria.....	3
Introduction	5
Capítulo 2 - Estado de la cuestión.....	9
2.1 Aplicaciones similares.....	9
2.1.1 mySugr	9
2.1.2 SocialDiabetes	10
2.1.3 Diabetes:M	10
2.1.4 FreeStyle LibreLink.....	11
2.1.5 Conclusiones	11
Capítulo 3 - Desarrollo de la aplicación	13
3.1 Tecnologías utilizadas	13
3.1.1 Entorno de desarrollo.....	13
3.1.2 Desarrollo Front-end y Back-end.....	14
3.1.3 Dependencias	15
3.2 Tipos de usuarios de la aplicación	16
3.2.1 Pacientes	16
3.2.2 Médicos	17
3.3 Base de datos	18
3.3.1 Tabla patient.....	18
3.3.2 Tabla doctor.....	19

3.3.3 Tabla patient_doctor	19
3.3.4 Tabla information	20
3.3.5 Tabla patient_information.....	20
3.3.6 Tabla food	20
3.3.7 Tabla information_food	21
3.3.8 Tabla blood_sugar.....	21
3.3.9 Tabla information_blood_sugar	21
3.3.10 Tabla sport	22
3.3.11 Tabla information_sport.....	22
3.3.12 Tabla ingredient.....	22
3.3.13 Tabla patient_ingredient.....	23
3.3.14 Tabla ration_table	23
3.3.15 Tabla sessions	23
3.4 Funcionalidad.....	24
3.4.1 Registro de usuario	24
3.4.2 Inicio de sesión.....	26
3.4.3 Página principal	27
3.5 Estructura del proyecto.....	37
Capítulo 4 - Conclusiones y trabajo futuro	41
4.1 Conclusiones.....	41
4.2 Trabajo futuro.....	41
Conclusions and future work	43
Apéndice A - Casos de uso	49

ÍNDICE DE FIGURAS

Figura 1 – 3.3 Diagrama entidad-relación de la base de datos	18
Figura 2 – 3.4 Registro de usuario	24
Figura 3 – 3.4 Ejemplo de error en el registro de usuario	25
Figura 4 – 3.4 Inicio de sesión.....	26
Figura 5 – 3.4 Error en el inicio de sesión.....	26
Figura 6 – 3.4 Página principal del paciente.....	27
Figura 7 – 3.4 Página principal del médico	27
Figura 8 – 3.4 Perfil del paciente, datos personales	27
Figura 9 – 3.4 Perfil del médico, datos personales	27
Figura 10 – 3.4 Perfil del paciente, datos diabéticos.....	28
Figura 11 – 3.4 Opciones del registro de comida	28
Figura 12 – 3.4 Registro de comida.....	29
Figura 13 – 3.4 Modal selección comida	29
Figura 14 – 3.4 Registro de comida.....	30
Figura 15 – 3.4 Registrar una nueva comida	30
Figura 16 – 3.4 Modal carbohidratos e insulina	31
Figura 17 – 3.4 Opciones del registro de glucemia	31
Figura 18 – 3.4 Registrar glucemia.....	31
Figura 19 – 3.4 Modal glucemia	32
Figura 20 – 3.4 Modal glucemia con hiperglucemia	32
Figura 21 – 3.4 Registro de deporte	33
Figura 22 – 3.4 Selección de fechas para el gráfico.....	33
Figura 23 – 3.4 Gráfico por fechas	33
Figura 24 – 3.4 Tabla de raciones [28]	34

Figura 25 – 3.4 Registros de un día	34
Figura 26 – 3.4 Gráfica de un día	35
Figura 27 – 3.4 Buscar pacientes	35
Figura 28 – 3.4 Registrar comentario del médico	36
Figura 29 – 3.4 Comentario del médico.....	36
Figura 30 – 3.5 Estructura de carpetas	37
Figura 31 – 3.5 Ficheros de la carpeta dao	37
Figura 32 – 3.5 Subcarpetas de la carpeta public	38
Figura 33 – 3.5 Ficheros de la carpeta images.....	38
Figura 34 – 3.5 Fichero de la carpeta stylesheets	38
Figura 35 – 3.5 Ficheros de la carpeta routes.....	38
Figura 36 – 3.5 Ficheros de la carpeta views.....	39

ÍNDICE DE TABLAS

Tabla 1 – Caso de uso registrar usuario.....	50
Tabla 2 – Caso de uso iniciar sesión.....	51
Tabla 3 – Caso de uso ver perfil	51
Tabla 4 – Caso de uso registrar o modificar los datos diabéticos del paciente.....	52
Tabla 5 – Caso de uso registrar comida.....	54
Tabla 6 – Caso de uso registrar glucemia.....	55
Tabla 7 – Caso de uso registrar actividad física	56
Tabla 8 – Caso de uso mostrar gráfico por fechas	57
Tabla 9 – Caso de uso mostrar tabla de raciones.....	57
Tabla 10 – Caso de uso ver los registros de un día	58
Tabla 11 – Caso de uso mostrar gráfico.....	59
Tabla 12 – Caso de uso buscar paciente	59
Tabla 13 – Caso de uso guardar paciente.....	60
Tabla 14 – Caso de uso registrar comentario del médico	61

Capítulo 1 - Introducción

1.1 Motivación

La diabetes es una enfermedad crónica que se caracteriza por mostrar niveles elevados de glucosa en sangre. A largo plazo, esto puede provocar daños graves en el corazón, los vasos sanguíneos, los ojos, los riñones y los nervios [1]. La glucosa es la principal fuente de energía que tenemos. El organismo la produce, pero también proviene de los alimentos que consumimos [2]. La insulina es una hormona que se produce en el páncreas y ayuda a que la glucosa entre en las células y así, ser utilizada como energía. Si no se produce suficiente insulina, la glucosa permanece en la sangre [2].

De los tipos de diabetes que hay, el más común es la diabetes tipo 2 [1]. Esta se da principalmente en adultos y ocurre cuando el cuerpo se vuelve resistente a la insulina o no produce la suficiente [1]. La diabetes tipo 1, también conocida como diabetes juvenil o diabetes insulino dependiente, es una enfermedad crónica en la que el páncreas produce poca o ninguna insulina por sí solo [1]. La diabetes gestacional se desarrolla durante el embarazo y suele desaparecer después del nacimiento del bebé, aunque las personas que la padecen tienen más probabilidades de desarrollar diabetes tipo 2 [3]. La prediabetes afecta a las personas que tienen niveles de glucosa en sangre elevados, pero no lo suficientemente altos como para diagnosticarles diabetes tipo 2 [3].

Los factores de riesgo de esta enfermedad varían según el tipo de diabetes [4]. Los antecedentes familiares pueden ser un factor, por eso se les realizan pruebas clínicas para detectar la presencia de células del sistema inmunitario relacionadas con la diabetes a los familiares de diabéticos tipo 1. A este tipo de diabetes, también afectan los factores ambientales y la geografía [4]. Aunque no se sabe el motivo, la raza o el origen étnico puede aumentar el riesgo de diabetes tipo 2. La prediabetes, diabetes tipo 2 y diabetes gestacional son más comunes en personas con sobrepeso u obesidad [4].

Alrededor de 442 millones de personas en el mundo tienen diabetes, y 1,5 millones de muertes cada año se deben a la diabetes [1]. Por ello, es importante tener un estilo

de vida saludable para tratar de prevenir esta enfermedad. Se recomienda consumir alimentos saludables, realizar actividad física y perder el exceso de peso [4].

1.2 Objetivos

El principal objetivo de este trabajo es ayudar a los pacientes diabéticos en el seguimiento de su enfermedad y facilitar la comunicación médico-paciente.

Los objetivos de manera más específica son los siguientes:

- Permitir el registro y la visualización de los datos personales relacionados con la diabetes del paciente que sean útiles para el correcto funcionamiento de la aplicación.
- Permitir el registro de las comidas y ayudar al paciente a determinar la cantidad de insulina que tiene que administrarse.
- Permitir el registro de la glucemia e indicar si está sufriendo hiperglucemia, hipoglucemia o si los niveles de glucemia son adecuados, y además ayudarlo a saber cómo solucionarlo.
- Permitir el registro de la actividad física.
- Generar gráficas con los registros de las glucemias para que se puedan visualizar los datos de manera más clara.
- Permitir a los médicos acceder a toda la información de los pacientes y añadir comentarios.
- Permitir la visualización de la tabla de raciones.

1.3 Plan de trabajo

Para el desarrollo de este proyecto, inicialmente se llevó a cabo una investigación de la materia. Esta consistió en recopilar información para determinar que funcionalidades eran importantes que estuviesen en la aplicación.

Una vez las funcionalidades estuvieron claras, se empezó con el desarrollo de la aplicación web. En primer lugar, se diseñó el esquema de la base de datos y después se comenzó el desarrollo de la funcionalidad.

Cuando la aplicación funcionaba correctamente, se empezó a desarrollar la memoria, al mismo tiempo que se iba mejorando la aplicación a nivel visual y concretando los mensajes de error que debía mostrar la aplicación.

1.4 Estructura de la memoria

La memoria sigue la siguiente estructura:

- En el “Capítulo 1 – Introducción”, se explica la motivación y los objetivos del trabajo, el plan que se ha seguido para la realización de este trabajo y la estructura de la memoria.
- En el “Capítulo 2 – Estado de la cuestión”, se muestran las aplicaciones disponibles actualmente, se explican sus funcionalidades y se describen las mejoras que aporta este trabajo a dichas aplicaciones.
- En el “Capítulo 3 – Desarrollo de la aplicación”, en este apartado se explican las tecnologías utilizadas para el desarrollo de la aplicación, los tipos de usuarios que pueden utilizar la aplicación, la estructura de la base de datos, las funcionalidades que tiene la aplicación y la estructura de carpetas utilizada para el desarrollo de la aplicación.
- En el “Capítulo 4 – Conclusiones y trabajo futuro”, en este apartado se exponen las conclusiones alcanzadas con la realización de este trabajo y se comentan las posibles funcionalidades que se podrían desarrollar en un futuro en la aplicación.
- En “Bibliografía”, se muestran las referencias de la información utilizada.
- En “Apéndice”, se explican los casos de uso de manera detallada a través de tablas.

Introduction

Motivation

Diabetes is a chronic disease characterized by high blood glucose levels. In the long term, this can lead to serious damage to the heart, blood vessels, eyes, kidneys, and nerves [1]. Glucose is the main source of energy we have. It is produced by the body, but it also comes from the food we eat [2]. Insulin is a hormone that is produced in the pancreas and helps glucose enter the cells and be used as energy. If not enough insulin is produced, glucose remains in the blood [2].

Of the types of diabetes that exist, the most common is type 2 diabetes [1]. It occurs mainly in adults and occurs when the body becomes resistant to insulin or does not produce enough insulin [1]. Type 1 diabetes, also known as juvenile diabetes or insulin-dependent diabetes, is a chronic disease in which the pancreas produces little or no insulin on its own [1]. Gestational diabetes develops during pregnancy and usually goes away after the baby is born, although people who suffer from it are more likely to develop type 2 diabetes [3]. Prediabetes affects people who have elevated blood glucose levels, but not high enough to be diagnosed with type 2 diabetes [3].

Risk factors for this disease vary depending on the type of diabetes [4]. Family history can be a factor, which is why family members of type 1 diabetics are tested for diabetes-related immune system cells for the presence of diabetes. This type of diabetes is also affected by environmental factors and geography [4]. Although the reason is not known, race or ethnicity can increase the risk of type 2 diabetes. Prediabetes, type 2 diabetes, and gestational diabetes are more common in people who are overweight or obese [4].

Around 442 million people in the world have diabetes, and 1.5 million deaths each year are due to diabetes [1]. Therefore, it is important to have a healthy lifestyle to try to prevent this disease. It is recommended to eat healthy foods, engage in physical activity, and lose excess weight [4].

Goals

The main objective of this work is to help diabetic patients in the monitoring of their disease and facilitate doctor-patient communication.

More specifically, the objectives are as follows:

- Allow the recording and display of the patient's diabetes-related personal data that is useful for the proper functioning of the application.
- Allow for the recording of meals and help the patient determine the amount of insulin to be administered.
- Allow the recording of blood glucose and indicate if you are suffering from hyperglycemia, hypoglycemia or if the blood glucose levels are adequate, and also help you know how to solve it.
- Allow physical activity to be recorded.
- Generate graphs with blood glucose records so that the data can be visualized more clearly.
- Allow clinicians to access all patient information and add comments.
- Allow the display of the ration table.

Work plan

For the development of this project, an investigation of the subject was initially carried out. This consisted of gathering information to determine which functionalities were important to be in the application.

Once the functionalities were clear, the development of the web application began. First, the database schema was designed, and then the development of the functionality began.

When the application was working properly, the memory began to be developed, while the application was improved visually and the error messages that the application should display were specified.

Memory structure

The report follows the following structure:

- In "Chapter 1 - Introduction," the motivation and objectives of the work are explained, along with the plan followed for its completion and the structure of the report.
- In "Chapter 2 - State of the Art," the current applications are presented, explaining their functionalities and the improvements brought to them by this work.
- In "Chapter 3 - Application Development," the technologies used for the application development are explained, along with the types of users who can use the application, the database structure and the application's functionalities.
- In "Chapter 4 - Conclusions and Future Work," the conclusions drawn from the completion of this work are presented, along with discussions on potential functionalities that could be developed in the application in the future.
- In the "Bibliography" section, the references from which the data were gathered are listed.
- In "Appendix," the use cases are explained in detail through tables.

Capítulo 2 - Estado de la cuestión

2.1 Aplicaciones similares

En este capítulo se van a presentar algunas aplicaciones para pacientes diabéticos que existen actualmente en el mercado. Se van a describir las funcionalidades de dichas aplicaciones y, finalmente, se incluirá una conclusión para mostrar lo que la aplicación desarrollada aporta a este mercado.

2.1.1 mySugr

mySugr [5] [6] es una aplicación móvil desarrollada por personas diabéticas. En la aplicación se pueden distinguir dos planes: el plan básico y el plan PRO. Para acceder a este último, debes tener el dispositivo Accu-Chek o pagar.

El plan básico de la aplicación incluye las siguientes funcionalidades:

- Registro personalizado: permite registrar datos como azúcar en sangre, comidas, actividad física, etc.
- Informe de progreso general: permite descargar informes, pero solo en CSV.
- HbA1c estimada.
- Desafíos básicos y etiquetas.
- Copia de seguridad de datos segura.

El plan PRO de la aplicación incluye todas las funcionalidades del plan básico y además añade las siguientes:

- Sincronización de música de fondo múltiple.
- Calculadora de bolo.
- Informe en PDF y Excel.
- Tarifa basal para usuarios de bombas.
- Fotos de comidas.
- Recordatorio de azúcar en sangre.

- Desafíos empoderadores.

2.1.2 SocialDiabetes

SocialDiabetes [7] [8] es una aplicación multiplataforma.

La plataforma médica incluye las siguientes funcionalidades:

- Permite configurar alertas y recibir notificaciones.
- Permite de manera remota modificar la pauta del paciente.
- Permite configurar el calculador de bolo y activar recomendaciones automáticas.
- Permite comunicarse con los pacientes y realizar el seguimiento online.

La plataforma para el paciente incluye las siguientes funcionalidades:

- Permite configurar el calculador de bolo y recibir recomendaciones de insulina antes de cada comida.
- Permite sincronizar el glucómetro y así enviar de forma automática los datos de la glucemia.
- Permite monitorear el progreso y visualizar los datos de forma fácil y gráfica.
- Permite compartir la información con el médico, permitiendo que pueda acceder a los datos del paciente.

2.1.3 Diabetes:M

Diabetes:M [9] es una aplicación móvil que incluye las siguientes funcionalidades:

- Permite ingresar las lecturas de glucosa, las inyecciones de insulina y la cantidad de carbohidratos.
- Asesor de bolo: permite calcular las unidades de insulina para las comidas según los carbohidratos, las grasas y las proteínas. Se pueden indicar los alimentos y especificar las cantidades. Permite ingresar la glucosa en sangre y determinar la dosis de insulina correcta.

- Base de datos de alimentos: ofrece una lista de alimentos.
- Permite ver gráficas detalladas y analíticas.
- Permite generar informes.

2.1.4 FreeStyle LibreLink

FreeStyle LibreLink [10] es una aplicación móvil que incluye las siguientes funcionalidades:

- Las lecturas de la glucosa se realizan de forma automática.
- Ofrece gráficas donde se pueden visualizar las lecturas de la glucosa del sensor en un día.
- Ofrece un gráfico que muestra los porcentajes de tiempo en los que las lecturas de glucosa del sensor estuvieron por encima, por debajo o dentro del rango de glucosa.
- Permite compartir datos con profesionales sanitarios.

2.1.5 Conclusiones

Teniendo en cuenta estas aplicaciones, la aplicación desarrollada en este proyecto será similar en algunos aspectos, pero también aportará funcionalidades nuevas.

El registro del día a día será similar, ya que son funcionalidades básicas que debe tener cualquier aplicación que sirva para el seguimiento de pacientes diabéticos. Sin embargo, esta aplicación aportará al registro de la comida una guía más detallada. De cada alimento que el paciente vaya a consumir calculará automáticamente los carbohidratos, simplemente indicando la cantidad y los valores de la etiqueta del producto. Además, gracias a la base de datos de alimentos, estos datos solo serán necesarios aportarlos la primera vez que los consuma o por el contrario usar los datos de la tabla de raciones que hay por defecto. La mayoría de las aplicaciones no incluyen una tabla de raciones, de las mencionadas anteriormente solo "Diabetes:M" tiene una base de datos de alimentos. Además, en la mayoría de las aplicaciones, el paciente

tiene que indicar los carbohidratos que va a consumir directamente en vez de realizar ese cálculo la propia aplicación.

En cuanto a la parte de las gráficas, la mayoría de las aplicaciones tienen gráficas determinadas, mensuales, quincenales, etc. En esta aplicación, se ofrecen gráficas diarias y de un periodo de tiempo, el cual el propio usuario podrá especificar a través del intervalo de fechas. En "mySugr", por ejemplo, las gráficas solo permiten visualizar hasta 90 días.

En cuanto a la comunicación médico-paciente, la mayoría de las aplicaciones permiten descargar informes en los que el médico puede ver la evolución del paciente cuando este acude a la consulta, el trato médico-paciente es totalmente nulo antes de dichas consultas. La única aplicación que ofrece un trato médico-paciente similar es "SocialDiabetes".

Capítulo 3 - Desarrollo de la aplicación

3.1 Tecnologías utilizadas

En este capítulo se van a exponer las tecnologías que se han utilizado para el desarrollo de la aplicación.

3.1.1 Entorno de desarrollo

3.1.1.1 Visual Studio Code

Visual Studio Code [11] es un editor de código fuente creado por Microsoft. Entre otras características, incluye soporte para la depuración, refactorización de código y control de versiones integrado con Git. Esta herramienta se ha utilizado para el desarrollo de la aplicación web.

3.1.1.2 XAMPP

XAMPP [12] es un paquete de software libre que incluye el sistema de gestión de base de datos MySQL. Funciona como un servidor web libre capaz de interpretar páginas dinámicas. Esta herramienta se ha utilizado para el servidor web.

3.1.1.3 phpMyAdmin

PhpMyAdmin [13] es una herramienta desarrollada en PHP que permite administrar MySQL y MariaDB a través de páginas web. Permite la creación y eliminación de bases de datos, la creación, eliminación y modificación de tablas y campos, la ejecución de cualquier sentencia SQL. Esta herramienta se ha utilizado para el desarrollo de la base de datos.

3.1.1.4 GitHub

GitHub [14] es una plataforma de desarrollo colaborativo que permite alojar proyectos utilizando el sistema de control de versiones Git. Es una de las plataformas más importantes de colaboración para proyectos de código abierto y se utiliza principalmente para la creación de código fuente de programas de ordenador. El código de estos proyectos se almacena principalmente de forma pública.

A continuación, se proporciona el enlace donde se encuentra el código de la aplicación:

<https://github.com/BeatrizAdA/TFG>

3.1.2 Desarrollo Front-end y Back-end

3.1.2.1 HTML

HTML [15] (HyperText Markup Language) es el lenguaje de marcado que se utiliza para crear las páginas web, y se considera el más importante. Define una estructura básica y un código para la presentación del contenido de una página web. Es el estándar que predomina para visualizar las páginas web y sirve para todos los navegadores actuales. Permite definir la estructura del documento mediante etiquetas.

3.1.2.2 CSS

CSS [16] (Cascading Style Sheets) es un lenguaje de diseño gráfico que sirve para definir y crear la presentación de un documento. Se utiliza para establecer el diseño visual de documentos web e interfaces de usuario escritas en HTML. Junto con HTML y JavaScript, es usado por muchos sitios web para la creación de páginas visualmente atractivas, interfaces de aplicaciones web o móviles.

3.1.2.3 Bootstrap

Bootstrap [17] es un framework multiplataforma o conjunto de herramientas de código abierto, para el diseño de aplicaciones web. Incluye plantillas de diseño con elementos basados en HTML y CSS. Solo se ocupa del desarrollo front-end.

3.1.2.4 JavaScript

JavaScript [18] es un lenguaje de programación interpretado. Se caracteriza por ser orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza del lado del cliente, como parte de un navegador web, lo que permite mejoras en la interfaz de usuario y páginas web dinámicas.

3.1.2.5 AJAX

AJAX [19] (Asynchronous JavaScript And XML) es una técnica de desarrollo web que permite la creación de aplicaciones web asíncronas. Las aplicaciones se ejecutan en el cliente, mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. Es posible interactuar con el servidor sin tener que recargar la página web, lo que mejora la interactividad, velocidad y usabilidad en la aplicaciones.

3.1.2.6 Node.js

Node.js [20] es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor, basado en JavaScript. Es asíncrono, con E/S de datos en una arquitectura orientada a eventos.

3.1.3 Dependencias

3.1.3.1 body-parser

Body-parser es un middleware utilizado en aplicaciones web para acceder a los datos de los formularios. Se encarga de decodificar la información de los formularios y almacenarla en "request.body".

3.1.3.2 ejs

EJS [21] (Embedded JavaScript templating) es un lenguaje de plantillas simple que permite generar marcado HTML con JavaScript simple.

3.1.3.3 express

Express es una infraestructura de aplicaciones web Node.js que ofrece un conjunto de características para las aplicaciones web y móviles [22]. Es el framework backend más utilizado para Node.js, diseñado para crear aplicaciones web de una sola página, multipágina e híbridas [23].

3.1.3.4 express-mysql-session

Express-mysql-session es un módulo que permite almacenar la información de la sesión en la base de datos.

3.1.3.5 express-session

Express-session es un middleware que gestiona el almacenamiento de las sesiones.

3.1.3.6 multer

Multer [24] es un middleware de Node.js utilizado para manejar datos de formularios en formato multipart/form-data. Se ha utilizado para almacenar la foto de perfil del usuario, utilizando el almacenamiento en memoria.

3.1.3.7 mysql

MySQL [25] es un sistema de gestión de bases de datos relacionales muy utilizado en aplicaciones web. En este caso, se utiliza para acceder a la base de datos.

3.2 Tipos de usuarios de la aplicación

En la aplicación se pueden distinguir dos tipos de usuarios: pacientes y médicos. A continuación, se detallan las funcionalidades que tiene la aplicación para cada tipo de usuario.

3.2.1 Pacientes

El paciente es el usuario principal de la aplicación, tiene acceso a las siguientes funcionalidades:

- Registro e inicio de sesión.
- Visualización del perfil, donde puede ver y modificar sus datos diabéticos.
- Vista principal, contiene toda la funcionalidad de la aplicación y muestra los registros de cada día.
- Registro de comida.
- Registro de glucemia.
- Registro de actividad física.
- Visualización de la tabla de raciones.

- Visualización de gráficas, tanto de un día concreto como de un periodo de tiempo.
- Visualización del historial, muestra todos los registros de lo que ha ocurrido en un día.

3.2.2 Médicos

El médico es el usuario secundario de la aplicación, tiene acceso a las siguientes funcionalidades:

- Registro e inicio de sesión.
- Visualización del perfil.
- Vista principal, contiene toda la funcionalidad de la aplicación y permite visualizar a sus pacientes.
- Búsqueda y guardado de pacientes.
- Visualización del historial, muestra todos los registros de lo que ha ocurrido en un día a un paciente en concreto.
- Visualización de gráficas, tanto de un día concreto como de un periodo de tiempo.
- Añadir comentarios a los registros del paciente.

3.3 Base de datos

A continuación se explicarán detalladamente las tablas que contiene la base de datos, junto con los campos que contienen cada una de ellas.

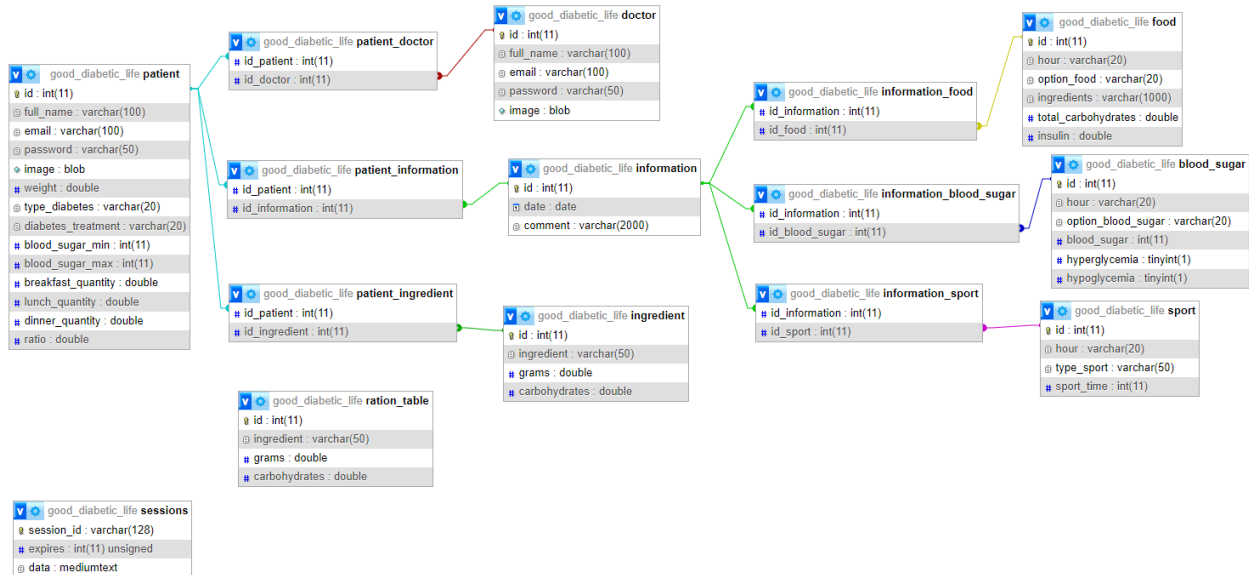


Figura 1 – 3.3 Diagrama entidad-relación de la base de datos

3.3.1 Tabla patient

La tabla patient contiene la información de los pacientes y está compuesta por los siguientes campos:

- id: entero que representa el identificador del paciente.
- full_name: varchar que representa el nombre completo del paciente.
- email: varchar que representa el correo electrónico del paciente.
- password: varchar que representa la contraseña del paciente.
- image: blob que representa la foto de perfil del paciente.
- weight: double que representa el peso del paciente.
- type_diabetes: varchar que representa el tipo de diabetes que tiene el paciente.
- diabetes_treatment: varchar que representa el tratamiento que sigue el paciente.

- blood_sugar_min: entero que representa la glucemia mínima que debe tener el paciente.
- blood_sugar_max: entero que representa la glucemia máxima que debe tener el paciente.
- breakfast_quantity: double que representa la cantidad de carbohidratos que debe consumir el paciente en el desayuno.
- lunch_quantity: double que representa la cantidad de carbohidratos que debe consumir el paciente en la comida.
- dinner_quantity: double que representa la cantidad de carbohidratos que debe consumir el paciente en la cena.
- ratio: double que representa la cantidad de insulina que el paciente debe administrarse por carbohidratos.

3.3.2 Tabla doctor

La tabla doctor contiene la información de los médicos y está compuesta por los siguientes campo:

- id: entero que representa el identificador del médico.
- full_name: varchar que representa el nombre completo del médico.
- email: varchar que representa el correo electrónico del médico.
- password: varchar que representa la contraseña del médico.
- image: blob que representa la foto de perfil del médico.

3.3.3 Tabla patient_doctor

La tabla patient_doctor guarda la relación de los pacientes y los médicos, y está compuesta por los siguientes campos:

- id_patient: entero que representa el identificador del paciente y hace referencia al identificador de la tabla patient.
- id_doctor: entero que representa el identificador del médico y hace referencia al identificador de la tabla doctor.

3.3.4 Tabla *information*

La tabla *information* contiene la información de los registros realizados por los pacientes. Sin embargo, no contiene la información completa, ya que está distribuida en diferentes tablas, como se muestra a continuación. Está compuesta por los siguientes campos:

- *id*: entero que representa el identificador de la información.
- *date*: fecha que representa la fecha de registro de la información.
- *comment*: varchar que representa el comentario del médico.

3.3.5 Tabla *patient_information*

La tabla *patient_information* guarda la relación entre los pacientes y la información, y está compuesta por los siguientes campos:

- *id_patient*: entero que representa el identificador del paciente y hace referencia al identificador de la tabla *patient*.
- *id_information*: entero que representa el identificador de la información y hace referencia al identificador de la tabla *information*.

3.3.6 Tabla *food*

La tabla *food* contiene la información de la comida registrada por los pacientes y está compuesta por los siguientes campos:

- *id*: entero que representa el identificador de la comida.
- *hour*: varchar que representa la hora de registro de la comida.
- *option_food*: varchar que representa la opción de la comida.
- *ingredients*: varchar que representa el conjunto de alimentos que va a consumir el paciente.
- *total_carbohydrates*: double que representa el total de carbohidratos que va a consumir el paciente en la comida.

- *insulin*: *double* que representa la cantidad de insulina que debe administrarse el paciente en la comida.

3.3.7 Tabla *information_food*

La tabla *information_food* guarda la relación entre la información y la comida, y está compuesta por los siguientes campos:

- *id_information*: *entero* que representa el identificador de la información y hace referencia al identificador de la tabla *information*.
- *id_food*: *entero* que representa el identificador de la comida y hace referencia al identificador de la tabla *food*.

3.3.8 Tabla *blood_sugar*

La tabla *blood_sugar* contiene la información de las glucemias registradas por el paciente y está compuesta por los siguientes campos:

- *id*: *entero* que representa el identificador de la glucemia.
- *hour*: *varchar* que representa la hora de registro de la glucemia.
- *option_blood_sugar*: *varchar* que representa la opción de la glucemia.
- *blood_sugar*: *entero* que representa la cantidad de glucemia que tiene el paciente.
- *hyperglycemia*: *booleano* que representa si el paciente tiene hiperglucemia.
- *hypoglycemia*: *booleano* que representa si el paciente tiene hipoglucemia.

3.3.9 Tabla *information_blood_sugar*

La tabla *information_blood_sugar* guarda la relación entre la información y la glucemia, y está compuesta por los siguientes campos:

- *id_information*: *entero* que representa el identificador de la información y hace referencia al identificador de la tabla *information*.

- `id_blood_sugar`: entero que representa el identificador de la glucemia y hace referencia al identificador de la tabla `blood_sugar`.

3.3.10 Tabla *sport*

La tabla `sport` contiene la información de la actividad física registrada por el paciente y está compuesta por los siguientes campos:

- `id`: entero que representa el identificador de la actividad física.
- `hour`: varchar que representa la hora de registros de la actividad física.
- `type_sport`: varchar que representa el deporte.
- `sport_time`: entero que representa el tiempo que el paciente realizada el deporte.

3.3.11 Tabla *information_sport*

La tabla `information_sport` guarda la relación entre la información y la actividad física, y está compuesta por los siguientes campos:

- `id_information`: entero que representa el identificador de la información y hace referencia al identificador de la tabla `information`.
- `id_sport`: entero que representa el identificador de la actividad física y hace referencia al identificador de la tabla `sport`.

3.3.12 Tabla *ingredient*

La tabla `ingredient` contiene la información de los ingredientes de las comidas registradas por el paciente y está compuesta por los siguientes campos:

- `id`: entero que representa el identificador del ingrediente.
- `ingredient`: varchar que representa el nombre del ingrediente.
- `grams`: double que representa la cantidad en gramos del ingrediente.
- `carbohydrates`: double que representa los carbohidratos que tiene el ingrediente.

3.3.13 Tabla *patient_ingredient*

La tabla *patient_ingredient* guarda la relación entre el paciente y los ingredientes, y está compuesta por los siguientes campos:

- *id_patient*: entero que representa el identificador del paciente y hace referencia al identificador de la tabla *patient*.
- *id_ingredient*: entero que representa el identificador del ingrediente y hace referencia al identificador de la tabla *ingredient*.

3.3.14 Tabla *ration_table*

La tabla *ration_table* contiene la información de las comidas predeterminadas y está compuesta por los siguientes campos.

- *id*: entero que representa el identificador de la comida predeterminada.
- *ingredient*: varchar que representa el nombre del ingrediente.
- *grams*: double que representa la cantidad en gramos del ingrediente.
- *carbohydrates*: double que representa los carbohidratos que tiene el ingrediente.

3.3.15 Tabla *sessions*

La tabla *sessions* contiene la información de las sesiones de los usuarios y se genera automáticamente.

3.4 Funcionalidad

A continuación se explican las funcionalidades que tiene la aplicación.

3.4.1 Registro de usuario

Para poder acceder a las funcionalidades de la aplicación, los usuarios tendrán que registrarse en ella.



El formulario de registro de usuario presenta dos opciones de rol: 'Médico' y 'Paciente'. Los campos de entrada incluyen: 'Nombre completo', 'Email', 'Contraseña' y 'Confirmar contraseña'. Una sección para 'Seleccionar archivo' muestra 'Sin archivos seleccionados'. Los botones 'Inicio' y 'Registro' están ubicados en la parte inferior.

Figura 2 – 3.4 Registro de usuario

Los usuarios tendrán que indicar:

- Si son médicos o pacientes, es una opción obligatoria.
- Su nombre completo, es un campo obligatorio. La aplicación comprobará si es un nombre válido.
- Su correo electrónico, es un campo obligatorio. La aplicación comprobará si es un correo electrónico válido.
- Su contraseña, es un campo obligatorio que debe tener al menos 8 caracteres, incluir al menos una mayúscula, una minúscula, un número y un carácter no alfanumérico.

- Confirmar la contraseña, es un campo obligatorio y debe coincidir con el campo contraseña.
- Su foto de perfil, es un campo opcional. Si el usuario no indica ninguna foto, la aplicación asignará una por defecto.

Después de realizar todas las comprobaciones, si la aplicación detecta algún error, mostrará el mensaje correspondiente. Por el contrario, se realizará el registro en la base de datos generando el identificador correspondiente y se redirigirá al usuario a la página de inicio de sesión.

El formulario de registro de usuario tiene una barra superior con dos botones: "Médico" y "Paciente". Debajo hay cinco campos de entrada:

- Nombre completo (con ícono de persona)
- Email (con ícono @)
- Contraseña (con ícono de llave)
- Confirmar contraseña (con ícono de llave)
- Selección de archivo (con ícono de foto y un botón "Sin archivos seleccionados")

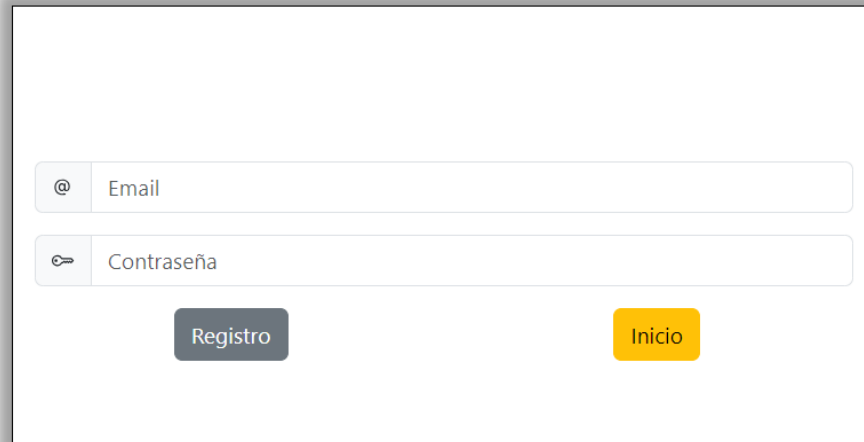
En la parte inferior hay dos botones: "Inicio" (gris) y "Registro" (amarillo).

El valor del campo nombre no representa un nombre válido

Figura 3 – 3.4 Ejemplo de error en el registro de usuario

3.4.2 Inicio de sesión

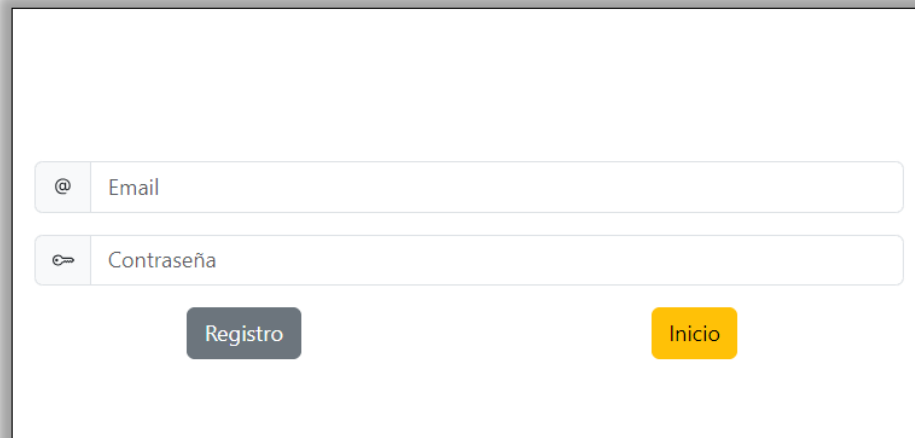
Después del registro, el usuario deberá iniciar sesión con las mismas credenciales utilizadas en el registro.



The screenshot shows a login interface with two input fields: 'Email' (with an '@' icon) and 'Contraseña' (with a key icon). Below the fields are two buttons: 'Registro' (dark grey) and 'Inicio' (yellow).

Figura 4 – 3.4 Inicio de sesión

La aplicación comprueba si los datos son válidos y coinciden con los registrados en la base de datos. Si es así, la aplicación muestra la página principal correspondiente dependiendo de si el usuario es paciente o médico. Si los datos no son válidos, la aplicación muestra el mensaje de error correspondiente.



The screenshot shows the same login interface as Figure 4, but with an error message 'Usuario y/o contraseña incorrectos' displayed below the input fields.

Usuario y/o contraseña incorrectos

Figura 5 – 3.4 Error en el inicio de sesión

3.4.3 Página principal

En la página principal se encuentra toda la funcionalidad de la aplicación.



Figura 6 – 3.4 Página principal del paciente



Figura 7 – 3.4 Página principal del médico

3.4.3.1 Ver perfil

Es una funcionalidad tanto de los pacientes como de los médicos. La parte común es visualizar los datos personales.



Figura 8 – 3.4 Perfil del paciente, datos personales



Figura 9 – 3.4 Perfil del médico, datos personales

Además, en el perfil del paciente se mostrarán y podrán modificarse los datos relacionados con la diabetes, los cuales son necesarios para el correcto funcionamiento de la aplicación. Al iniciar sesión por primera vez, el paciente debe acceder a su perfil y registrar dichos datos.



The image shows a form for entering diabetic patient data. It consists of several rows, each with a label on the left and a text input field on the right. The data entered in the fields is as follows:

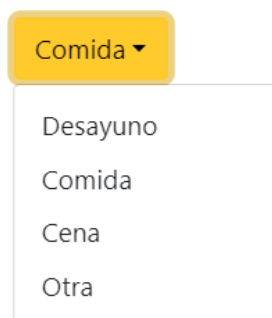
Label	Value
Peso	65
Tipo de diabetes	Tipo 1
Tratamiento de diabetes	Bomba de insulina
Glucemia mínima	70
Glucemia máxima	180
Desayuno	100
Comida	100
Cena	100
Ratio insulina (1 unidad)	9

At the bottom right of the form is a yellow button labeled "Modificar".

Figura 10 – 3.4 Perfil del paciente, datos diabéticos

3.4.3.2 Registro de comidas

Es una funcionalidad exclusiva del paciente. Lo primero que tiene que hacer es indicar la comida que va a realizar.



The image shows a dropdown menu for selecting a meal. The menu is currently open, showing the selected item "Comida" with a downward arrow, and a list of options below it: "Desayuno", "Comida", "Cena", and "Otra".

Figura 11 – 3.4 Opciones del registro de comida

Comidas

Carbohidratos totales

Figura 12 – 3.4 Registro de comida

El paciente ingresará el nombre de la comida y la cantidad que va a consumir, luego pulsará el botón “Añadir”. Esto hará que la aplicación muestre un modal donde podrá seleccionar la comida que quiere utilizar para realizar los cálculos o registrar una nueva.

Comida
×

Tabla de raciones

Comida	Gramos	Carbohidratos	
Macarrones	100	20	<input style="background-color: #FFD700; border: none; padding: 2px 5px;" type="button" value="Usar"/>
Macarrones con tomate	150	25	<input style="background-color: #FFD700; border: none; padding: 2px 5px;" type="button" value="Usar"/>

Tabla de comidas

Comida	Gramos	Carbohidratos	
Macarrones	100	20	<input style="background-color: #FFD700; border: none; padding: 2px 5px;" type="button" value="Usar"/>

Figura 13 – 3.4 Modal selección comida

Si el paciente usa una de las comidas ya registradas, ya sea de la tabla de raciones o de la tabla de comidas, se actualizará la vista de la comida, añadiendo a la lista la comida seleccionada y mostrando el cálculo de los carbohidratos totales.

Los carbohidratos totales se calculan con la fórmula:

$$\text{Carbohidratos totales} = (\text{cantidad que va a consumir} \times \text{carbohidratos}) / \text{gramos}$$

Los valores de carbohidratos y gramos se obtienen según la comida seleccionada de la tabla de raciones o de la tabla de comidas.

The screenshot shows a form titled "Comidas". At the top, there are two input fields: "Comida" and "Cantidad". To the right of "Cantidad" are two buttons labeled "g" and "ml". Further right is a yellow button labeled "Añadir". Below these is a large text area containing "90g Macarrones". Underneath the text area is a display field labeled "Carbohidratos totales" showing the value "18". At the bottom center is a yellow button labeled "Registrar comida".

Figura 14 – 3.4 Registro de comida

Si el paciente, en lugar de usar una comida ya registrada, pulsa sobre el botón "Registrar", se abrirá un nuevo modal donde tendrá que indicar los valores de la comida.

The screenshot shows a modal window titled "Nueva comida" with a close button (X) in the top right corner. The main heading inside is "Indica los datos de la etiqueta de la comida". Below this are two input fields: "Carbohidratos" and "Cantidad". To the right of "Cantidad" are two buttons labeled "g" and "ml". At the bottom right of the modal are two buttons: "Close" (grey) and "Aceptar" (yellow).

Figura 15 – 3.4 Registrar una nueva comida

El cálculo de los carbohidratos totales se realiza usando la misma fórmula, pero los valores de los carbohidratos y los gramos se obtienen de los registrados en este modal.

Una vez termina de registrar la comida completa, la aplicación le muestra un modal indicándole si ha llegado o no a los carbohidratos de dicha comida y cuánta insulina debe administrarse.

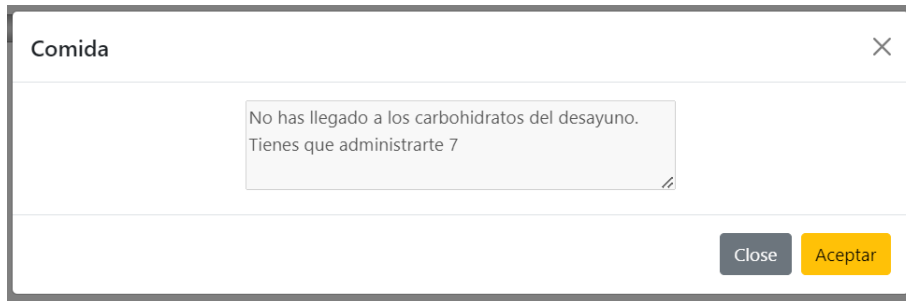


Figura 16 – 3.4 Modal carbohidratos e insulina

El cálculo de la insulina se realiza utilizando el ratio registrado en los datos diabéticos del perfil del paciente y los carbohidratos totales que el paciente va a consumir en la comida, calculados anteriormente. La fórmula utilizada es:

$$\text{Insulina} = (I \times \text{carbohidratos totales}) / \text{ratio}$$

3.4.3.3 Registro de glucemia

Es una funcionalidad exclusiva del paciente. Lo primero que tiene que hacer es indicar la glucemia que quiere registrar.

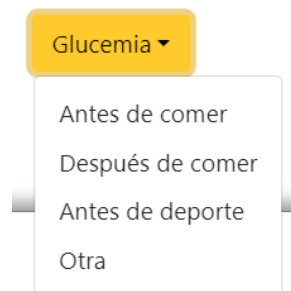


Figura 17 – 3.4 Opciones del registro de glucemia

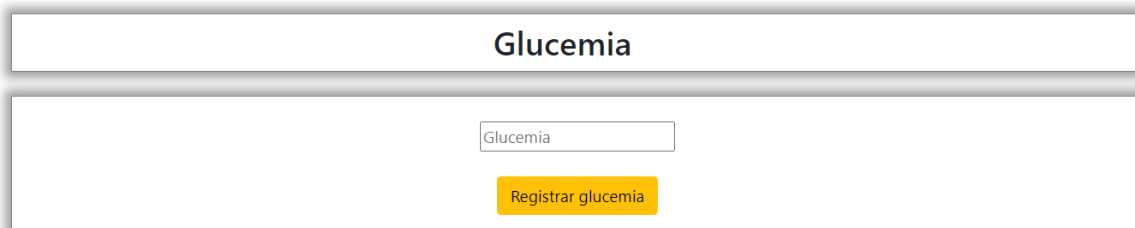


Figura 18 – 3.4 Registrar glucemia

El paciente indicará la glucemia que tiene y, al registrarla, la aplicación mostrará un modal indicando si los valores son correctos o no.

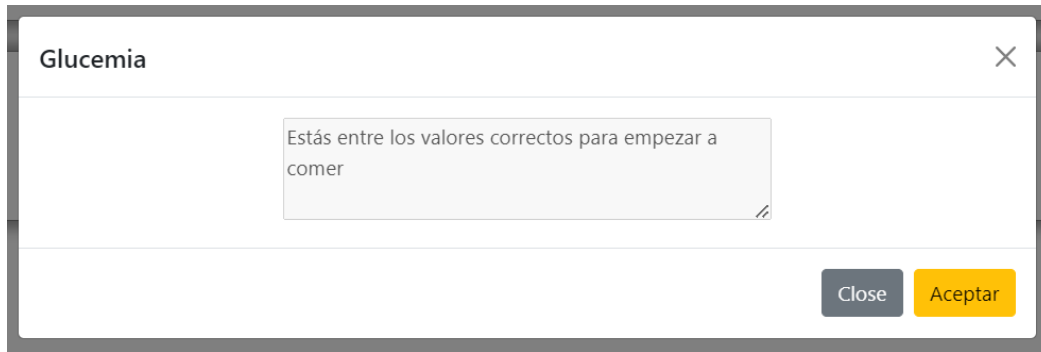


Figura 19 – 3.4 Modal glucemia

Para saber si la glucemia está entre los valores correctos o no, la aplicación tiene en cuenta la glucemia máxima y mínima que el paciente ha indicado anteriormente en los datos diabéticos del perfil. También se tienen en cuenta los siguientes parámetros [26] [27]:

- Mínimo para poder empezar a comer: 80.
- Máximo para poder empezar a comer: 130.
- Después de comer: 180.
- Mínimo para poder hacer deporte: 90.
- Máximo para poder hacer deporte: 250.

Teniendo en cuenta todos estos valores, la aplicación muestra el mensaje correspondiente y, si el paciente está sufriendo una hiperglucemia o hipoglucemia le ayuda a solucionarla.

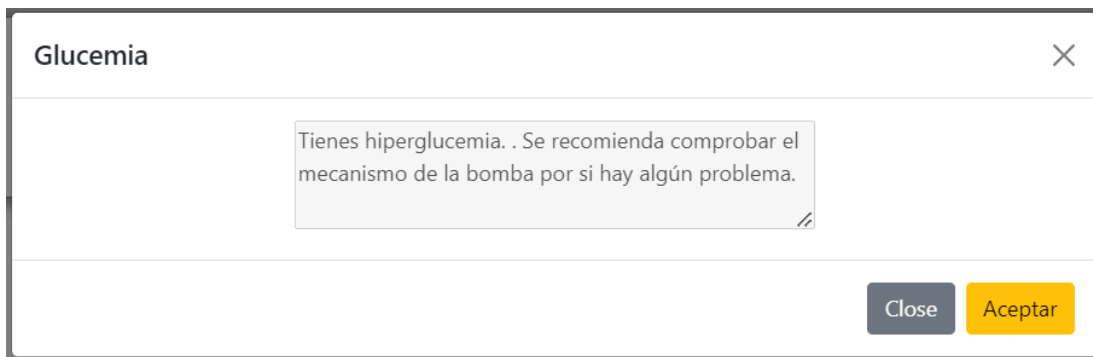



Figura 20 – 3.4 Modal glucemia con hiperglucemia

3.4.3.4 Registro de actividad física

Es una funcionalidad exclusiva del paciente.



Deporte

Ejercicio Tiempo en minutos

Registrar deporte

Figura 21 – 3.4 Registro de deporte

El paciente completará los campos necesarios y la aplicación realizará el registro.

3.4.3.5 Mostrar gráfico por fechas

Es una funcionalidad tanto de los pacientes como de los médicos.



Gráfico ▾

dd/mm/aaaa

dd/mm/aaaa Aceptar

Figura 22 – 3.4 Selección de fechas para el gráfico

El usuario tendrá que indicar las dos fechas entre las que quiere que se le muestre la gráfica de las glucemias.

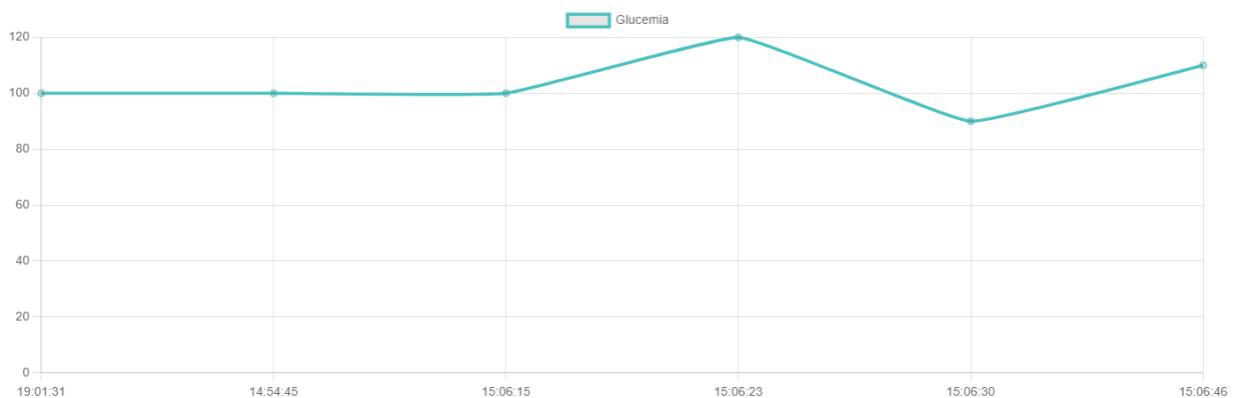


Figura 23 – 3.4 Gráfico por fechas

3.4.3.6 Mostrar tabla de raciones

Es una funcionalidad tanto de los pacientes como de los médicos.

Tabla raciones		
Comida	Gramos	Carbohidratos
Arroz	80	20
Arroz a la cubana	190	25
Macarrones	100	20
Macarrones con tomate	150	25
Espaguetis	100	20
Espaguetis con tomate	150	25
Garbanzos	100	20
Lentejas	100	20
Guisantes	200	20
Alubias	100	20
Patatas hervidas	100	20
Patatas fritas	60	20

Figura 24 – 3.4 Tabla de raciones [28]

En ella se pueden ver las comidas que tiene guardadas la aplicación por defecto.

3.4.3.7 Ver los registros de un día

Es una funcionalidad tanto de los pacientes como de los médicos.

En ella, se pueden visualizar los registros que hace un paciente a lo largo de un día. También se puede visualizar la gráfica de glucemias de ese día en concreto.



Figura 25 – 3.4 Registros de un día

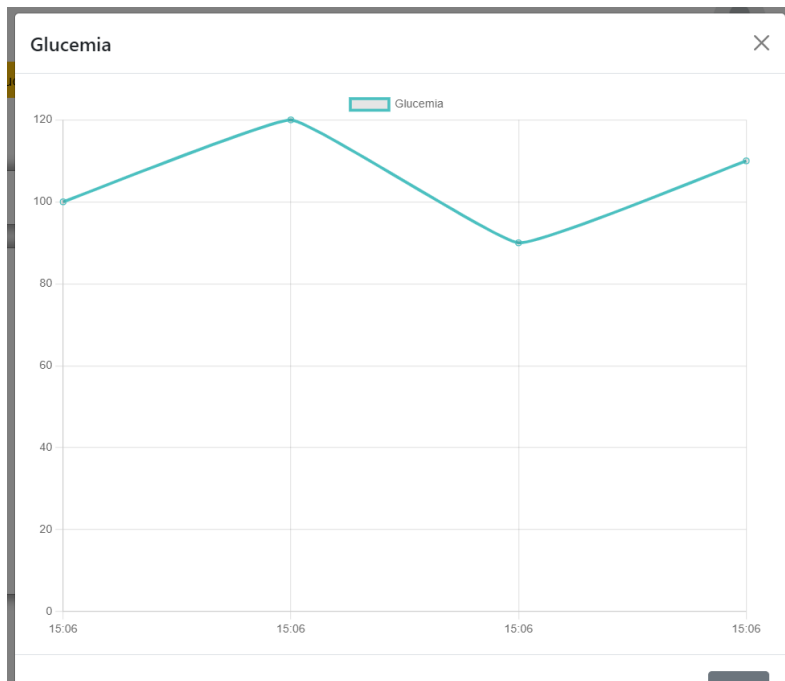


Figura 26 – 3.4 Gráfica de un día

3.4.3.8 Buscar pacientes

Es una funcionalidad exclusiva de los médicos. El médico indica el nombre del paciente en el buscador y se le mostrarán todos los pacientes que tengan dicho nombre.

GoodDiabeticLife

Tabla raciones



Daniel Álvarez de Arriba

Salir

Buscar

Beatriz Álvarez de Arriba Guardar

Figura 27 – 3.4 Buscar pacientes

El médico podrá guardarse a sus pacientes para que le aparezcan en la pantalla principal sin necesidad de buscarlos.

3.4.3.9 Registrar comentarios del médico

Es una funcionalidad exclusiva de los médicos.

GoodDiabeticLife



Daniel Álvarez de Arriba

Salir

Gráfico ▾

Tabla raciones

Buscar

12/5/2024

14:50		DESAYUNO	<div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px;">90g Macarrones 90g Pan</div> Carbohidratos totales: 63 Insulina administrada: 7.0
14:54		GLUCEMIA	100
14:58		DEPORTE	Correr Tiempo: 30 minutos

[Gráfica](#)

[Añadir comentario](#)

Figura 28 – 3.4 Registrar comentario del médico

Cuando el médico pulse sobre el botón “Añadir comentario”, se le habilitará un campo de texto donde puede escribir el comentario. Una vez registrado, tanto el paciente como el médico podrán visualizarlo.

12/5/2024

14:50		DESAYUNO	<div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px;">90g Macarrones 90g Pan</div> Carbohidratos totales: 63 Insulina administrada: 7.0
14:54		GLUCEMIA	100
14:58		DEPORTE	Correr Tiempo: 30 minutos

[Gráfica](#)

Comentario del médico

Figura 29 – 3.4 Comentario del médico

3.5 Estructura del proyecto

A continuación, se va a comentar la estructura de carpetas y ficheros utilizada para el desarrollo del proyecto.

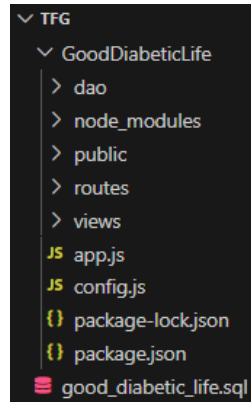


Figura 30 – 3.5 Estructura de carpetas

Dentro de la estructura del proyecto se encuentran las carpetas:

- dao: se encarga de la interacción con la base de datos.

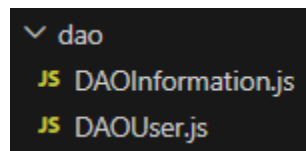


Figura 31 – 3.5 Ficheros de la carpeta dao

Esta carpeta incluye los siguientes ficheros:

- DAOInformation.js: en este fichero se realizan todas las funciones que interactúan con la base de datos relacionadas con la información de la diabetes.
- DAOUser.js: en este fichero se realizan todas las funciones que interactúan con la base de datos relacionadas con los usuarios, tanto pacientes como médicos.
- node_modules: es una carpeta generada automáticamente al ejecutar npm. Contiene las dependencias y los módulos necesarios para el correcto funcionamiento de la aplicación.

- public: esta carpeta almacena los ficheros estáticos.

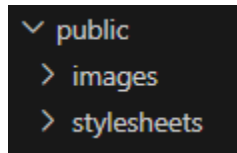


Figura 32 – 3.5 Subcarpetas de la carpeta public

Esta carpeta se divide en dos subcarpetas:

- images: es una carpeta que contiene las imágenes de la aplicación, como el favicon y la imagen predeterminada utilizada para usuarios sin foto de perfil.

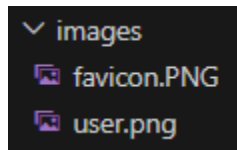


Figura 33 – 3.5 Ficheros de la carpeta images

- stylesheets: es una carpeta que contiene el fichero style.css, encargado del diseño de la aplicación.

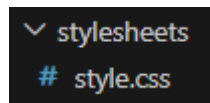


Figura 34 – 3.5 Fichero de la carpeta stylesheets

- routes: se encarga de las rutas de la aplicación.

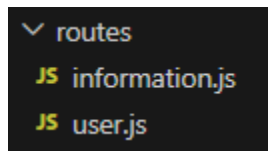


Figura 35 – 3.5 Ficheros de la carpeta routes

Esta carpeta incluye los siguientes ficheros:

- information.js: en este fichero se definen todas las funcionalidades de las rutas de la aplicación que tienen que ver con la información de la diabetes.
- user.js: en este fichero se definen todas las funcionalidades de las rutas de la aplicación que tienen que ver con los usuarios, tanto pacientes como médicos.

- views: contiene las plantillas de la aplicación.

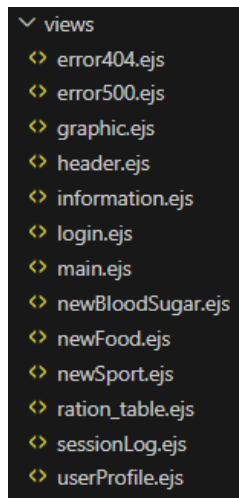


Figura 36 – 3.5 Ficheros de la carpeta views

Esta carpeta incluye los siguientes ficheros:

- error404.ejs: define la vista del error 404. El error 404 hace referencia a página no encontrada.
- error500.ejs: define la vista del error 500. El error 500 hace referencia a un error interno del servidor.
- graphic.ejs: define la vista del gráfico por fechas.
- header.ejs: es un fichero auxiliar utilizado para definir la cabecera de la aplicación, donde se encuentran todos los botones de las funcionalidades.
- information.ejs: define la vista de los registros de un día, donde se muestra la gráfica del día y la información.
- login.ejs: define la vista de inicio de sesión.
- main.ejs: define la vista de la página principal de la aplicación.
- newBloodSugar.ejs: define la vista de los registros de las glucemias. En él también se encuentra el script que muestra el modal.
- newFood.ejs: define la vista de los registros de las comidas. En él también se encuentran los scripts que muestran los modales.
- newSport.ejs: define la vista de los registros de la actividad física.

- ration_table.ejs: define la vista de la tabla de raciones.
 - sessionLog.ejs: define la vista del registro de usuario.
 - userProfile.ejs: define la vista del perfil del usuario. En él también se encuentra el script que habilita los campos del registro de los datos diabéticos.
- app.js: es el fichero principal de la aplicación. En él se inicia la aplicación.
 - config.js: en este fichero se define la configuración de la base de datos y el puerto en el que se ejecuta el servidor.
 - package-lock.json: es un fichero que se genera automáticamente y contiene las dependencias del proyecto.
 - package.json: es un fichero que contiene la información del proyecto.

Fuera de la estructura del proyecto se encuentra:

- good_diabetic_life.sql: es un fichero que contiene la definición de la estructura de la base de datos.

Capítulo 4 - Conclusiones y trabajo futuro

4.1 Conclusiones

El desarrollo de esta aplicación web se ha llevado a cabo con la finalidad de ayudar a las personas que sufren diabetes a llevar un mejor control de la enfermedad. También se ha intentado que la comunicación entre el médico y el paciente sea más directa de lo que es hoy en día.

La aplicación ofrece a los pacientes una forma fácil de registrar las comidas y les ayuda a realizar los cálculos necesarios para que el control de la diabetes sea más exhaustivo. También les proporciona beneficios en los registros de la glucemia, ayudándoles a llevar un mejor control e incluso una mejor comprensión de estos al verlos en forma de gráfica. Incluir la tabla de raciones en la aplicación les aporta facilidad, ya que tienen todo dentro de un mismo sitio.

La aplicación ofrece a los médicos una consulta directa de los datos y el tratamiento del paciente, permitiéndoles ver cómo transcurre su enfermedad.

Es por todo esto que, una vez finalizado, se puede afirmar que los objetivos propuestos al inicio de este proyecto se han cumplido.

4.2 Trabajo futuro

La aplicación cumple con los objetivos descritos, pero existen mejoras que se pueden realizar en un futuro.

- Actualmente, la aplicación solo permite registros de glucemia de forma manual. Sería interesante que la aplicación pudiera comunicarse con los medidores de glucemia que existen y usan los pacientes, permitiendo así realizar registros de glucemia de forma automática.
- En esta versión de la aplicación se ha mejorado la relación médico-paciente, pero sería interesante añadir un chat interno en la aplicación para que la comunicación sea más fluida.

Conclusions and future work

Conclusions

The development of this web application has been carried out with the aim of helping people suffering from diabetes to better control the disease. There has also been an attempt to make communication between doctor and patient more direct than it is today.

The app gives patients an easy way to log meals and helps them make the calculations needed to make diabetes management more comprehensive. It also provides them with benefits in blood glucose recordings, helping them to keep better control and even better understand them by viewing them in graph form. Including the ration table in the app makes it easier for them, as they have everything in one place.

The app gives doctors direct insight into patient data and treatment, allowing them to see how their disease is progressing.

It is for all these reasons that, once completed, it can be said that the objectives proposed at the beginning of this project have been met.

Future Work

The application fulfills the objectives described, but there are improvements that can be made in the future.

- Currently, the app only allows manual blood glucose recordings. It would be interesting if the application could communicate with the blood glucose meters that exist and are used by patients, thus allowing blood glucose recordings to be made automatically.
- In this version of the application, the doctor-patient relationship has been improved, but it would be interesting to add an internal chat in the application to make communication more fluid.

BIBLIOGRAFÍA

- [1] «World Health Organization,» [En línea]. Available: https://www.who.int/health-topics/diabetes#tab=tab_1.
- [2] «MedlinePlus,» [En línea]. Available: <https://medlineplus.gov/spanish/diabetes.html>.
- [3] «National Institute of Diabetes and Digestive and Kidney Diseases,» [En línea]. Available: <https://www.niddk.nih.gov/health-information/informacion-de-la-salud/diabetes/informacion-general/que-es>.
- [4] «Mayo Clinic,» [En línea]. Available: <https://www.mayoclinic.org/es/diseases-conditions/diabetes/symptoms-causes/syc-20371444>.
- [5] «mySugr,» [En línea]. Available: <https://www.mysugr.com/en/>.
- [6] «mySugr,» [En línea]. Available: <https://www.mysugr.com/en/diabetes-app/>.
- [7] «SocialDiabetes,» [En línea]. Available: <https://www.socialdiabetes.com/es/app>.
- [8] «SocialDiabetes,» [En línea]. Available: <https://www.socialdiabetes.com/es/platform>.
- [9] «Diabetes:M,» [En línea]. Available: <https://diabetes-m.com/features/>.
- [10] «FreeStyle,» [En línea]. Available: <https://www.freestyle.abbott/es-es/productos/conectividad/freestylelibrelinkapp.html>.
- [11] «Wikipedia Visual Studio Code,» [En línea]. Available: https://es.wikipedia.org/wiki/Visual_Studio_Code.
- [12] «Wikipedia XAMPP,» [En línea]. Available: <https://es.wikipedia.org/wiki/XAMPP>.

- [13] «Wikipedia PhpMyAdmin,» [En línea]. Available: <https://es.wikipedia.org/wiki/PhpMyAdmin>.
- [14] «Wikipedia GitHub,» [En línea]. Available: <https://es.wikipedia.org/wiki/GitHub>.
- [15] «Wikipedia HTML,» [En línea]. Available: <https://es.wikipedia.org/wiki/HTML>.
- [16] «Wikipedia CSS,» [En línea]. Available: <https://es.wikipedia.org/wiki/CSS>.
- [17] «Wikipedia Bootstrap,» [En línea]. Available: [https://es.wikipedia.org/wiki/Bootstrap_\(framework\)](https://es.wikipedia.org/wiki/Bootstrap_(framework)).
- [18] «Wikipedia JavaScript,» [En línea]. Available: <https://es.wikipedia.org/wiki/JavaScript>.
- [19] «Wikipedia AJAX,» [En línea]. Available: <https://es.wikipedia.org/wiki/AJAX>.
- [20] «Wikipedia Node.js,» [En línea]. Available: <https://es.wikipedia.org/wiki/Node.js>.
- [21] «EJS,» [En línea]. Available: <https://ejs.co/>.
- [22] «Express,» [En línea]. Available: <https://expressjs.com/es/>.
- [23] «Kinsta,» [En línea]. Available: <https://kinsta.com/es/base-de-conocimiento/que-es-express/#:~:text=Cerrar,Express.,desarrollar%20aplicaciones%20backend%20con%20Node>.
- [24] «KeepCoding,» [En línea]. Available: <https://keepcoding.io/blog/que-es-multer-en-express-y-node-js/>.
- [25] «Wikipedia MySQL,» [En línea]. Available: <https://es.wikipedia.org/wiki/MySQL>.
- [26] «FreeStyle,» [En línea]. Available: <https://www.freestyle.abbott/es-es/midiabetes/controlglucemia.html>.

[27] «FreeStyle,» [En línea]. Available: <https://www.freestyle.abbott/es-es/midiabetes/diabetes-y-deporte.html>.

[28] «Roche,» [En línea]. Available: <http://www.aprendizdediabetes.es/Guia-alimentaria.pdf>.

APÉNDICES

Apéndice A - Casos de uso

En este apéndice se van a especificar los casos de uso de las distintas funcionalidades de la aplicación.

Registrar usuario

Descripción	Añade un nuevo usuario, paciente o médico, a la base de datos.
Actor	Usuario, tanto paciente como médico.
Entrada	Paciente o médico, nombre completo, email, contraseña, confirmar contraseña y foto de perfil.
Salida	Redirección a la vista de login, en caso de error en el registro muestra mensaje de error.
Origen	Interfaz de registro de usuario.
Destino	Base de datos de paciente o médico.
Necesita	Base de datos de paciente o médico.
Acciones	<ol style="list-style-type: none">1. El usuario accede a la página de registro.2. El usuario rellena la información pedida y pulsa el botón "Registro".3. La aplicación comprueba si los datos introducidos por el usuario son válidos.<ol style="list-style-type: none">a. Si los datos son válidos, los registra en la base de datos.b. Si los datos no son válidos, muestra el mensaje de error correspondiente.

Precondición	No existe un usuario con el mismo nombre o email.
Postcondición	Se registra en la base de datos al usuario, paciente o médico.

Tabla 1 – Caso de uso registrar usuario

Iniciar sesión

Descripción	Inicia la sesión del usuario, paciente o médico, para poder acceder a todo el funcionamiento de la aplicación.
Actor	Usuario, tanto paciente como médico.
Entrada	Email y contraseña.
Salida	Redirección a la vista principal, en caso de error en el inicio muestra mensaje de error.
Origen	Interfaz de inicio de sesión de usuario.
Destino	Ninguno.
Necesita	Base de datos de paciente o médico.
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de sesión. 2. El usuario rellena la información pedida y pulsa el botón "Inicio". 3. La aplicación comprueba si los datos introducidos por el usuario son válidos. <ol style="list-style-type: none"> a. Si los datos son válidos, inicia la sesión mostrándole la vista principal. b. Si los datos no son válidos, muestra el mensaje de error correspondiente.
Precondición	El usuario existe en la base de datos de paciente o médico.

Postcondición	El usuario puede usar la aplicación.
----------------------	--------------------------------------

Tabla 2 – Caso de uso iniciar sesión

Ver perfil

Descripción	Muestra los datos personales del usuario, paciente o médico.
Actor	Usuario, tanto paciente como médico.
Entrada	Nombre del usuario, paciente o médico.
Salida	Redirección a la vista del perfil.
Origen	Interfaz de perfil de usuario.
Destino	Ninguno.
Necesita	Base de datos de paciente o médico.
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre su nombre. 2. La aplicación le redirige a la página donde se encuentra su perfil. 3. El usuario puede ver la información de su perfil.
Precondición	El usuario existe en la base de datos de paciente o médico.
Postcondición	Se muestran los datos del perfil del usuario, paciente o médico.

Tabla 3 – Caso de uso ver perfil

Registrar o modificar datos diabéticos del paciente

Descripción	Registra o modifica la información de la diabetes.
Actor	Usuario, paciente.
Entrada	Peso, tipo de diabetes, tratamiento de diabetes, glucemia mínima, glucemia máxima, carbohidratos desayuno, carbohidratos comida, carbohidratos cena y ratio de insulina.
Salida	Redirección a la vista de perfil, con los datos registrados o modificados.
Origen	Interfaz de perfil de usuario, paciente.
Destino	Base de datos del paciente.
Necesita	Base de datos del paciente.
Acciones	<ol style="list-style-type: none"> 1. El usuario, paciente, accede a la página de ver perfil. 2. Pulsa sobre el botón "Modificar". 3. El usuario, paciente, rellena la información pedida y pulsa el botón "Registrar". 4. El sistema registra o modifica los datos introducidos por el paciente en la base de datos.
Precondición	El paciente existe.
Postcondición	Se lleva a cabo el registro o la modificación de los datos en la base de datos del paciente.

Tabla 4 – Caso de uso registrar o modificar los datos diabéticos del paciente

Registrar comida

Descripción	Añade una nueva comida a la base de datos.
Actor	Usuario, paciente.
Entrada	Tipo de comida, ingrediente y cantidad.
Salida	Redirección a la vista principal.
Origen	Interfaz de registro de comida.
Destino	Base de datos de comida.
Necesita	Base de datos de comida, tabla de raciones, ingredientes e información.
Acciones	<ol style="list-style-type: none">1. El usuario accede a la página de registro de comida indicando el tipo (desayuno, comida, cena u otra).2. El usuario rellena la información pedida y pulsa el botón "Añadir".3. La aplicación abre un modal para que el usuario seleccione la comida que desea utilizar para el registro de los carbohidratos. Puede seleccionar entre una predeterminada (tabla de raciones), una guardada anteriormente por él o registrar una nueva.<ol style="list-style-type: none">a. Si el usuario decide seleccionar una predeterminada o una guardada anteriormente por él, el sistema realizará los cálculos necesarios para ir calculando los carbohidratos totales.

	<p>b. Si el usuario decide registrar una comida nueva, la aplicación abrirá otro modal en el que el paciente tiene que indicar los datos de la etiqueta del producto (carbohidratos y cantidad) y pulsar sobre el botón "Aceptar". Al igual que en el punto anterior, la aplicación realizará los cálculos necesarios.</p> <p>4. Cuando el usuario termina de rellenar la comida, pulsa el botón "Registrar comida".</p> <p>5. La aplicación le muestra si ha cumplido con la cantidad de carbohidratos que tiene que ingerir en dicha comida y la cantidad de insulina que tiene que administrarse.</p> <p>6. El usuario pulsará sobre el botón "Aceptar" y la aplicación hará el registro de la comida.</p>
Precondición	El paciente existe.
Postcondición	Si todo va bien, se registra la comida en la base de datos.

Tabla 5 – Caso de uso registrar comida

Registrar glucemia

Descripción	Añade una nueva glucemia a la base de datos.
Actor	Usuario, paciente.
Entrada	Tipo de glucemia y glucemia.
Salida	Redirección a la vista principal.
Origen	Interfaz de registro de glucemia.
Destino	Base de datos de glucemia.

Necesita	Base de datos de glucemia e información.
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la página de registro de glucemia indicando el tipo (antes de comer, después de comer, antes de deporte u otra). 2. El usuario rellena la información pedida y pulsa el botón "Registrar glucemia". 3. La aplicación le muestra el aviso de la glucemia, indicándole si está entre los valores correctos o está sufriendo hiperglucemia o hipoglucemia. En estos casos, la aplicación le indicará como solucionarlo. 4. El usuario pulsará sobre el botón "Aceptar" y la aplicación hará el registro de la glucemia.
Precondición	El paciente existe.
Postcondición	Si todo va bien, se registra la glucemia en la base de datos.

Tabla 6 – Caso de uso registrar glucemia

Registrar actividad física

Descripción	Añade un nuevo registro de deporte a la base de datos.
Actor	Usuario, paciente.
Entrada	Tipo de ejercicio y tiempo.
Salida	Redirección a la vista principal.
Origen	Interfaz de registro de deporte.
Destino	Base de datos de deporte.
Necesita	Base de datos de deporte e información.

Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la página de registro de deporte. 2. El usuario rellena la información pedida y pulsa el botón "Registrar deporte". 3. La aplicación registra el deporte y le redirige a la vista principal.
Precondición	El usuario existe.
Postcondición	Si todo va bien, se registra el deporte en la base de datos.

Tabla 7 – Caso de uso registrar actividad física

Mostrar gráfico por fechas

Descripción	Muestra la gráfica de glucemias entre dos fechas dadas.
Actor	Usuario, tanto paciente como médico.
Entrada	Fecha de inicio y fecha de fin.
Salida	Gráfica con los datos de las glucemias.
Origen	Interfaz de gráfico.
Destino	Ninguno.
Necesita	Base de datos de la glucemia e información.
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el botón "Gráfico" e indica las fechas entre las que quiere que se le muestren la gráfica y pulsa el botón "Aceptar". 2. La aplicación le redirige a la vista y muestra la gráfica.
Precondición	El usuario existe y las fechas son válidas.

Postcondición	Se muestra la gráfica con los valores de las glucemias.
----------------------	---

Tabla 8 – Caso de uso mostrar gráfico por fechas

Mostrar tabla de raciones

Descripción	Muestra la tabla de raciones, son los alimentos que tiene la base de datos guardados por defecto.
Actor	Usuario, tanto paciente como médico.
Entrada	Ninguna.
Salida	Tabla de raciones.
Origen	Interfaz de tabla de raciones.
Destino	Ninguno.
Necesita	Base de datos de tabla de raciones.
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la página de tabla de raciones. 2. La aplicación le muestra la tabla.
Precondición	La base de datos tiene comidas guardadas.
Postcondición	Se muestra la tabla de raciones.

Tabla 9 – Caso de uso mostrar tabla de raciones

Ver los registros de un día

Descripción	Muestra los registros que hace el paciente a lo largo de un día.
Actor	Usuario, tanto paciente como médico.
Entrada	Nombre del paciente y fecha.

Salida	Registros del paciente en la fecha dada.
Origen	Interfaz de información.
Destino	Ninguno.
Necesita	Base de datos de información, comida, glucemia y deporte.
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la información del día. 2. La aplicación le redirige a la vista y le muestra la información.
Precondición	La base de datos tiene registros de ese día.
Postcondición	Se muestra la información de ese día.

Tabla 10 – Caso de uso ver los registros de un día

Mostrar gráfico

Descripción	Muestra la gráfica de glucemias de un día concreto.
Actor	Usuario, tanto paciente como médico.
Entrada	Fecha.
Salida	Gráfica con los datos de las glucemias.
Origen	Interfaz de gráfico.
Destino	Ninguno.
Necesita	Base de datos de la glucemia e información.
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la página de información de un día concreto. 2. El usuario pulsa sobre el botón "Gráfica".

	3. La aplicación le muestra la gráfica.
Precondición	El usuario y la información existen.
Postcondición	Se muestra la gráfica del día.

Tabla 11 – Caso de uso mostrar gráfico

Buscar paciente

Descripción	Buscar un paciente por su nombre.
Actor	Usuario, médico.
Entrada	Nombre del paciente.
Salida	Información del paciente buscado.
Origen	Interfaz principal.
Destino	Ninguno.
Necesita	Base de datos de pacientes.
Acciones	<ol style="list-style-type: none"> 1. El médico introduce el nombre de un paciente y pulsa el botón "Buscar". 2. La aplicación actualiza la vista y le muestra todos los pacientes que coincidan con ese nombre.
Precondición	El médico y el paciente buscado tienen que existir.
Postcondición	Se muestra la lista de los pacientes que coinciden con ese nombre.

Tabla 12 – Caso de uso buscar paciente

Guardar paciente

Descripción	Guarda los pacientes para que el médico pueda verlos en su página principal.
Actor	Usuario, médico.
Entrada	Nombre del paciente.
Salida	Ninguna.
Origen	Interfaz de buscar paciente.
Destino	Base de datos de paciente_médico.
Necesita	Base de datos de paciente y médico.
Acciones	<ol style="list-style-type: none"> 1. El médico pulsa sobre el botón "Guardar". 2. La aplicación guarda al paciente entre los pacientes del médico.
Precondición	El médico y el paciente existen.
Postcondición	Guarda al paciente entre los pacientes del médico y permite verlo en la página principal del médico.

Tabla 13 – Caso de uso guardar paciente

Registrar comentario del médico

Descripción	Añade un nuevo comentario del médico a la base de datos.
Actor	Usuario, médico.
Entrada	Comentario.
Salida	Redirección a la vista de información de una día concreto.

Origen	Interfaz de información de un día concreto.
Destino	Base de datos de información.
Necesita	Base de datos de información.
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la página de información de un día concreto. 2. El usuario pulsa sobre el botón "Añadir comentario". 3. La aplicación le muestra el campo texto para que añada el comentario. 4. El usuario rellena el campo y pulsa sobre el botón "Registrar comentario". 5. La aplicación registra el comentario y le redirige a la vista de información con el comentario registrado.
Precondición	El usuario existe.
Postcondición	Muestra el comentario en la información del día.

Tabla 14 – Caso de uso registrar comentario del médico