**NutriGestión**

Iván Canas Ramos

**Trabajo de fin de grado en ingeniería Informática, facultad de informática**

**Universidad Complutense de Madrid**

**Imagen que contiene objeto

Descripción generada automáticamente**

Madrid, 20 de abril de 2019

**Director:** Antonio Sarasa Cabezuelo

# Índice

[Índice 2](#_Toc7890613)

[1. Introducción 3](#_Toc7890614)

[1.1. Ámbito de trabajo 3](#_Toc7890615)

[1.2. Estructura de la memoria 4](#_Toc7890616)

[2.Motivación 6](#_Toc7890617)

[3. Estado del Arte 7](#_Toc7890618)

[3.1. nutrium.io 7](#_Toc7890619)

[3.2. nutritioapp 9](#_Toc7890620)

[4. Tecnología Empleada 11](#_Toc7890621)

[4.1. Front-end 11](#_Toc7890622)

[4.1.1. Angular 11](#_Toc7890623)

[4.1.2. Typescript 12](#_Toc7890624)

[4.1.3. Bootstrap 12](#_Toc7890625)

[4.1.4. Node.js 12](#_Toc7890626)

[4.2. Back-end 13](#_Toc7890627)

[4.2.1. Apache HTTP Server 13](#_Toc7890628)

[4.2.2. PHP 13](#_Toc7890629)

[4.2.1. MariaDB 13](#_Toc7890630)

[4.2.3. Xampp 14](#_Toc7890631)

[4.3. Librerías 14](#_Toc7890632)

[4.3.1. ngx-bootstrap 14](#_Toc7890633)

[4.3.2. Chart.js 15](#_Toc7890634)

[4.3.3. Font Awesome 15](#_Toc7890635)

[4.4. Herramientas 15](#_Toc7890636)

[4.4.1. Visual Studio Code 15](#_Toc7890637)

[4.4.2. Git 16](#_Toc7890638)

[4.4.3. Postman 16](#_Toc7890639)

[5. Casos de uso 17](#_Toc7890640)

[5.1. Definición de términos 17](#_Toc7890641)

[5.1. Casos de uso (índice) 18](#_Toc7890642)

[5.1. Casos de uso (Tablas) 19](#_Toc7890643)

[6. Modelo de datos 32](#_Toc7890644)

# **Introducción**

En plena revolución digital, la informática ha entrado de lleno en todos los campos, industrias y en todas las profesiones del mundo. Nos encontramos en plena transición de lo analógico a lo digital, del bolígrafo y el papel, a los datos almacenados digitalmente. Profesionales de todos los campos, se ven obligados a subirse al carro de las nuevas tecnologías para ser competentes en su campo, poder crecer y diferenciarse de la competencia, en un mundo donde sus competidores, cada día, encuentra en la informática nuevas herramientas con la que diferenciarse del resto, haciéndoles más productivos, eficientes, eficaces y mejores que sus competidores.

Mediante el proceso de desarrollo de software, es posible crear herramientas informáticas que permitan a empresas y profesionales llevar a cabo su actividad de una forma más eficiente, ordenada y les otorga nuevas formas de desempeñar sus funciones, las cuales proporcionan valor añadido.

Hoy en día el ideal para un profesional interesado en informatizar su trabajo o negocio sería el desarrollo de una herramienta informática a medida de sus necesidades, que le permita realizar todas las acciones de su día a día y que le permitiese automatizar, o facilitar tareas periódicas.

## **1.1. Ámbito de trabajo**

En este documento, se describirá detalladamente el desarrollo de una aplicación enfocada a las ciencias de la salud, en concreto, se trata de una aplicación hecha a medida para profesionales de la nutrición.

Esta aplicación es herramienta que acompaña al nutricionista durante el ejercicio de sus funciones en una consulta.

La aplicación será la encargada de gestionar todos los pacientes que un nutricionista pueda tener, así como los datos de los pacientes, sus citas, sus medidas y métricas, sus dietas, su progreso y sus patologías.

Esta aplicación permite también, a un paciente acceder a ella y consultar sus datos, su progreso, su próxima cita y su dieta.

Se detallará todo el proceso de desarrollo de software a medida para nutricionistas, así como la estructura del modelo de datos, la arquitectura de la aplicación, tecnologías y herramientas utilizadas, funcionalidades de la aplicación, evaluación de la aplicación, así como conclusiones y trabajo futuro.

Todo el trabajo, se ha llevado a cabo teniendo en cuenta en todo momento las necesidades de los profesionales de nutrición, por lo que se considera un desarrollo hecho a medida que facilite, mejore y cubra las necesidades de un profesional de la nutrición a la hora de gestionar a sus pacientes en una consulta.

Para el desarrollo de la aplicación, se ha consultado cada decisión y funcionalidad con un profesional de este campo con el objetivo de que el producto final, cubra sus necesidades y le permita mejorar en su trabajo.

## **1.2. Estructura de la memoria**

1. **Introducción**

Se explica el contexto actual de la situación del uso de la informática como herramienta de trabajo y cómo el uso de estas herramientas se ha convertido en imprescindibles contextualizando la situación actual. Se explicará también el ámbito de la aplicación.

1. **Motivación**

Contará cual es el motivo y el porqué de la decisión de llevar a cabo el desarrollo de la herramienta. Qué beneficios aporta respecto de no utilizarla y cuales son sus aportaciones a los profesionales que la utilicen.

1. **Estado del arte**

Contiene una visión global de diferentes aplicaciones, funcionalidades y beneficios de aplicaciones similares a la aquí descrita, presentes en el mercado y disponibles a cualquiera que desde este momento quieran utilizarlas.

1. **Tecnología empleada**

Detalla cuáles son las herramientas empleadas para la construcción, el diseño y la implementación de la aplicación, así como la plataforma sobra la cual se ha construido y lenguajes de programación utilizados.

1. **Casos de uso**

Muestra todas las funcionalidades que se han implementado en la aplicación. En este apartado muestra detalles técnicos de cada funcionalidad como los datos de entrada, salida y secuencias de funcionamiento.

1. **Modelo de datos**

Explica detalladamente la estructura de la base de datos y la dependencia de los datos en las diferentes tablas creadas, su funcionalidad y significado de cada uno.

1. **Arquitectura**

Este apartado contiene los detalles de la implementación, modelos y patrones utilizados y de que manera se adaptan a la aplicación.

1. **Diseño de la aplicación**

Contiene la explicación de cada funcionalidad detallando partes del código junto a su explicación.

1. **Evaluación**

Detalla las conclusiones de la evaluación realizada de la aplicación por parte de profesionales de la nutrición y diferentes campos.

1. **Conclusiones y trabajo futuro**

Relatará las conclusiones de haber llevado a cabo la aplicación, lo aprendido durante su desarrollo, así como trabajo por hacer en un futuro

1. **Guía de uso**

Contiene las instrucciones lo mas detalladamente posible para poder utilizar la aplicación por parte de usuarios con bajos conocimientos de informática. Contendrá capturas de la aplicación junto con instrucciones de uso.

1. **Guía de instalación de la aplicación**

Explica con todo lujo de detalles, desde lo más básico, todo lo necesario para instalar la aplicación y ponerla en marcha. Contendrá una guía que abarca desde la puesta en marcha, hasta el montaje del entorno de desarrollo necesario para realizar cambios en la misma.

# **2.Motivación**

Actualmente, profesionales de la salud como los nutricionistas, poseen los conocimientos necesarios para mejorar nuestra salud a través de hábitos alimenticios saludables. Para ello, como cualquier profesional que gestione pacientes, necesita de herramientas que le permitan llevar un control de todos sus pacientes.

El control sobre los datos de estos pacientes es imprescindible, ya que en la actividad del día a día en una consulta de nutrición, se extraen multitud de datos, como pueden ser sus medidas antropométricas, hábitos alimenticios y enfermedades diagnosticadas.

Estos datos y su veracidad son fundamentales para que el nutricionista desempeñe su trabajo mejor y de forma más precisa.

El nutricionista, en su trabajo, no solo debe obtener datos de sus pacientes, su trabajo, va más allá, y debe interpretar estos datos obtenidos. Por ello, uno de los requisitos fundamentales es el de mantener un histórico de los datos de cada paciente. De esta forma, es capaz de realizar un estudio a lo largo del tiempo de la evolución de sus pacientes, y así, tomar decisiones en base a sus conocimientos e información de los datos recopilados. De esta manera, se asegura el poder llevar a cabo un tratamiento adecuado a cada paciente, habiendo estudiado la evolución de los datos y tendrá la seguridad de que sus conclusiones son fiables debido a que cubren un largo periodo de tiempo, y por ello, le otorga un margen de error mínimo.

Con la ayuda de una herramienta informática hecha a medida, el nutricionista, se asegura disponer de una lista de pacientes, con unas características determinadas y de todos sus datos y cálculos sobre los mismos en el menor tiempo posible, mostrarlos de una manera clara e inequívoca, ayudándole a ser competente y más productivo frente a sus competidores.

El objetivo de este trabajo es el de proporcionar a los profesionales de la nutrición, una herramienta enfocada a su día a día, capaz de cubrir todas las necesidades profesionales que puedan surgir y acercar al profesional y al paciente, ya que permite ser utilizada también por el paciente, para consultar sus datos, progreso, dietas y para mantenerse en contacto con el profesional. De esta manera, el paciente puede obtener realimentación por parte de la aplicación de sus consultas al visualizar de una forma sencilla su progreso manteniendo así la motivación para lograr sus objetivos.

Los datos almacenados, tratados y generados por esta herramienta, facilitan el trabajo diario del nutricionista al crearse una única fuente estandarizada de datos y herramientas, que antes se encontraban en lugares diferentes como folios, hojas de Excel o documentos de Word.

De esta forma, ahora los datos se guardarán en un único lugar siendo accesibles en todo momento desde la aplicación. Todos los cálculos necesarios los llevará a cabo la herramienta, las tareas comunes y cotidianas de un nutricionista se han simplificado dentro de la herramienta permitiéndole ahorrar tiempo. El histórico de datos de un paciente, es ahora visible de una manera muy sencilla, así como fácilmente legibles a través de gráficas.

# **3. Estado del Arte**

Actualmente existen varias herramientas capaces de gestionar pacientes, muchas de ellas son muy completas y contienen multitud de campos para poblar con datos.

Todas estas aplicaciones tienen un denominador común, que es su precio. Todas ellas tienen unos precios bastante elevados y para un nutricionista que busca hacerse un hueco en el mundo laboral, puede suponer un gran problema.

Algunas de estas aplicaciones, se desvían de su objetivo principal y terminan por abarcar muchos campos convirtiéndose así en aplicaciones muy grandes, que generan mucha confusión a la hora de ser utilizadas por un usuario novato e inexperto interesado en este tipo de aplicaciones. Esto conlleva una curva de aprendizaje elevada que sin duda proporciona funcionalidades útiles, pero se desvían del trabajo diario de una consulta de nutrición.

A modo de ejemplo, alguna de las aplicaciones de las que voy a hablar a continuación, contienen recetas de cocina, es un valor añadido, pero a efectos prácticos de una consulta de nutrición y gestión de pacientes, son prescindibles.

A continuación, mencionaré alguna de estas aplicaciones y las describiré brevemente:

## **3.1. nutrium.io**

nutrium.io[[1]](#footnote-1) es una aplicación web encargada de gestionar pacientes, citas, dietas, recetas, alimentos, tablas de equivalencias y comunicación directa con el profesional.

Imagen que contiene captura de pantalla

Descripción generada automáticamente

Figura 1 - nutrium.io principal

nutrium.io proporciona una interfaz bastante limpia, cuidada y amigable. A primera vista, permite encontrar las funcionalidades que buscas de un vistazo.

Posee una base de datos con gran cantidad de alimentos desglosando todas las propiedades de cada alimento como vitaminas, grasas, proteínas, calorías, colesterol, etc.

La aplicación nos permite añadir gran cantidad de datos a la ficha del paciente, muchos de ellos innecesarios para llevar a cabo una consulta y un seguimiento nutricional.

Imagen que contiene captura de pantalla

Descripción generada automáticamente

Figura 2- Representación ficha de paciente

En la Figura 3, podemos ver la sección se recetas de nutrium.io. Esta funcionalidad es la encargada de añadir nuestros alimentos a la base de datos, proporcionando todos los valores nutricionales de los mismos. Una vez añadidos los alimentos, se pueden combinar para crear recetas. Es una funcionalidad muy potente e inteligente, pero tiene el inconveniente de tener que introducir todos los alimentos de cada receta y sus valores nutricionales a mano. Existen alimentos ya completos con sus valores nutricionales ya introducidos en su base de datos, pero el nutricionista no puede confiar en que esos datos sean correctos, debe cerciorarse de que así es, y eso conlleva trabajo y tiempo por parte del profesional.

Por todo ello, lo considero una funcionalidad muy potente, pero necesaria del respaldo de unos datos confirmados y verídicos.

El precio de nutrium.io es de 50€ mensuales para un máximo de 10 pacientes, 68€ para un máximo de 25 pacientes, y 104€ mensuales para pacientes ilimitados.

Imagen que contiene captura de pantalla

Descripción generada automáticamente

Figura 3 - Recetas nutrium.io

## **3.2. nutritioapp**

La aplicación web de nutritioapp[[2]](#footnote-2) cuenta con una interfaz muy limpia y es muy intuitiva de utilizar. Esto se debe a que presenta unas funcionalidades muy especificas y no se desvía de su principal objetivo. Los datos que solicita al profesional respecto del paciente son los necesarios para llevar a cabo el trabajo profesional.

La aplicación permite crear múltiples pacientes y asignarles una cita, añadirle datos médicos, medidas y dietas.

Imagen que contiene captura de pantalla

Descripción generada automáticamente

Figura 4 - Principal nutritioapp

Esta aplicación, nos permite introducir alguna de las medidas del paciente, las cuales en un principio aparentan ser escasas para llevar un seguimiento adecuado de su evolución.

La aplicación, no permite ver una evolución de forma gráfica, solo nos permite ver el histórico de las pocas medidas que hemos introducido de forma numérica.

Esto puede resultar de ayuda al profesional, pero necesitaría de algunas medidas más para poder llevar un correcto seguimiento, ya que la principal funcionalidad de estas herramientas es la de aglutinar las herramientas que un nutricionista utiliza en su día a día.

Al prescindir de estos datos, el profesional deberá buscar la manera de guardar estos datos en otro lugar, haciendo que su trabajo no esté centralizado en un único lugar.

Imagen que contiene captura de pantalla

Descripción generada automáticamente

Figura 5 - nutritioapp medidas e histórico

En lo referente a obtener métricas y datos a partir de esos datos, la aplicación solo facilita el índice de masa corporal, su índice metabólico y su ingesta de calorías diarias recomendadas. Estos datos calculados por la aplicación en base a las medidas introducidas pueden ser escasos y poco fiables. El nutricionista, necesita de mucha más información para poder realizar su trabajo correctamente, por lo que le tocaría utilizar otras herramientas para la obtención de estos datos, o calcularlos a mano.

La aplicación, no dispone de una base de datos de alimentos, y a su vez, permite introducir alimentos a su base de datos y rellenar todos los valores nutricionales que pudiese tener. Estos alimentos introducidos, pueden ser utilizados más adelante para realizar una dieta a cada paciente.

El hecho de que te obligue a introducir los valores nutricionales de cada alimento genera la confianza necesaria para estar seguro de que el profesional ha introducido unos datos fiables.

Una de las funcionalidades de esta aplicación es que posee un chat en directo entre el nutricionista y el paciente. Sin duda es una funcionalidad muy útil, ya que acerca al profesional y al paciente. Al mismo tiempo, puede ser contraproducente para el nutricionista y el paciente ya que el nutricionista está trabajando, y si su bolsa de pacientes es muy grande, puede estar gran cantidad de su tiempo respondiendo mensajes de sus pacientes; y por el otro lado, el paciente puede sentirse abandonado si el nutricionista no responde a sus mensajes.

Por todo esto, puede ser una funcionalidad no aplicable a una herramienta de estas características.

El precio de nutritioapp es de 44€ al mes con máximo de 25 pacientes, 74€ al mes con máximo de 75 pacientes, no siendo posible añadir ni un paciente más.

Su página web, no indica precio para más pacientes, lo incluye en su plan de clínica del cual no ofrecen información de precios.

# **4. Tecnología Empleada**

La herramienta NutriGestión es una aplicación web que utiliza multitud de tecnologías para su construcción y correcta ejecución, las cuales se describirán y detallarán a continuación.

Al tratarse de una aplicación web, implícitamente quedarán claramente diferenciadas dos partes: la parte del cliente (Front-end) y la parte del servidor (Back-end).

Al mismo tiempo, se detallarán las herramientas y tecnologías que se han utilizado para construir cada parte, así como todas las librerías utilizadas en el desarrollo de la aplicación.

## **4.1. Front-end**

### 4.1.1. Angular

Angular[[3]](#footnote-3) es un framework de desarrollo de aplicaciones web, desarrollado y mantenido por Google[[4]](#footnote-4), cuya finalidad es el desarrollo de aplicaciones web SPA[[5]](#footnote-5), esto quiere decir que aplicación web desarrollada con Angular, constaría de una sola página, en donde la navegación y las interacciones sobre esa página, se hacen en la misma página, de manera que no es necesario recargar la página en cada cambio de sección, por lo que todas las acciones, son dinámicas, asíncronas, reactivas e instantáneas.

Al considerarse Angular un framework, nos proporciona multitud de herramientas necesarias para facilitarnos la tarea de construir cualquier aplicación web y construirla de una manera mucho más optimizada.

Se ha utilizado Angular, como parte del front-end, y por ello, su ejecución se llevará a cabo en la parte del cliente, el cual estará consumiendo un servicio de parte del servidor alojado en el back-end.

Este framework sigue una filosofía de desarrollo por componentes. Cada componente es independiente de los demás y cada uno tiene asignada una funcionalidad concreta con el objetivo de construir aplicaciones web de manera modular. Un componente tiene tres archivos diferenciados: archivo HTML, archivo Typescript, y archivo CSS.

* HTML: Representa la vista y los elementos de un componente que aparecen en la máquina del cliente.
* Typescript: Representa el modelo de datos con los que el componente va a trabajar.
* CSS: Son las hojas de estilo. Los estilos definidos en cada componente afectan únicamente a dicho componente.

En Angular, existen también los componentes denominados “servicios” que son los encargados de comunicarse con los demás componentes y librerías y proporcionar intercambios de datos.

La versión de Angular utilizada para desarrollar la aplicación ha sido la versión 7.3

### 4.1.2. Typescript

Es un lenguaje de programación de código abierto, desarrollado y mantenido por Microsoft[[6]](#footnote-6), el cual se puede considerar como un lenguaje de programación construido sobre JavaScript al que se le añaden funcionalidades como la programación orientada a objetos y tipado estático del lenguaje.

Typescript y JavaScript, son completamente compatibles debido a que todos los archivos con extensión Typescript (.ts) se compilan y se convierten en archivos de JavaScript (.js).

Por todo esto, nos encontramos con un lenguaje de programación, que aprovecha todas las ventajas y librerías de JavaScript, y funcionalidades de un lenguaje de programación sólido y estructurado, que nos permite detectar errores en tiempo de compilación.

Se ha utilizado el lenguaje Typescript a lo largo de todo el desarrollo front-end en el framework de Angular para realizar transformaciones sobre el modelo de datos.

La versión de Typescript utilizada para desarrollar la aplicación ha sido la versión 3.2.4

### 4.1.3. Bootstrap

Se trata de un framework CSS desarrollado inicialmente por el equipo de Twitter[[7]](#footnote-7) cuyo principal objetivo es el de facilitar el desarrollo de una página web, a través de las hojas de estilo CSS.

Bootstrap nos proporciona los componentes necesarios para realizar la vista de una aplicación web de manera adaptativa. Esto quiere decir, que el desarrollador, se despreocupa en parte de conocer las características del navegador web del cliente, y utilizando las herramientas de Bootstrap, se asegura que tendrá un diseño similar independientemente del navegador web que se utilice.

La filosofía de Bootstrap se basa en un diseño de filas con doce columnas cada fila. Podremos indicar a cada componente como mediante sus atributos, cuantas columnas va a ocupar y Bootstrap se encargará de que el componente ocupe el lugar deseado.

Este framework, nos proporciona multitud de estilos y efectos para muchos de los componentes de HTML, y las propiedades de todos ellos, son fácilmente modificables para adaptarlos a nuestro diseño.

La versión de Bootstrap utilizada para definir los estilos de las vistas de la aplicación ha sido la versión 4.3.1

### 4.1.4. Node.js

Node.js se define como un entorno de ejecución para JavaScript en el lado del servidor, fuera de un navegador web, construido sobre el motor JavaScript V8 de Chrome[[8]](#footnote-8).

Angular, está construido en torno a Node.js por lo tanto, para poder instalar el framework de Angular, primero debemos tener instalado Node.js, y descargarnos, a través del gestor de paquetes de Node.js (npm), el módulo de Angular, que incluye todas las herramientas en este paquete para comenzar a desarrollar la aplicación.

Node.js está construido de manera que permite ejecuciones asíncronas y orientada a eventos en tiempo real.

A pesar de la definición de Node.js respecto a trabajar del lado del servidor, Angular utiliza muchas de las funcionalidades de Node.js otorgando multitud de herramientas para poder construir la parte front-end.

La versión utilizada de Node.js para el desarrollo de la aplicación es la v10.15.1 LTS[[9]](#footnote-9)

### 

## **4.2. Back-end**

### 4.2.1. Apache HTTP Server

Apache HTTP Server es un servidor web desarrollado y mantenido por *Apache Foundation* encargado de recibir, procesar información en la parte del servidor y de renderizar la información para enviar a un cliente que ha realizado una petición.

Se utilizará Apache como el servidor de alojamiento de todos los archivos necesarios para la ejecución de la aplicación.

Para que un cliente, pueda utilizar y acceder a la aplicación, debe realizar una petición el servidor Apache. El servidor entonces le entregará al cliente los archivos necesarios tras procesar la petición y la preparación de los datos a enviar.

La versión Apache utilizada para el desarrollo de la herramienta es la 2.4.34 (Unix)

### 4.2.2. PHP

PHP son los acrónimos en inglés de “preprocesador de hipertexto”. Esto quiere decir que es un lenguaje interpretado y no se compila.

El código PHP se procesa en un servidor, en este caso Apache.

Para el desarrollo de esta Aplicación, se ha construido una herramienta en PHP en el lado del servidor con el propósito de realizar llamadas internas a las bases de datos dentro del mismo, motivadas por la necesidad de enviar esta información a la parte front-end y poblarla de los datos que sean necesarios.

Se ha utilizado también PHP para validar las credenciales introducidas por el cliente a través del front-end y de esta manera, notificar si puede acceder a la aplicación o si no.

La versión PHP utilizada para el desarrollo de la aplicación es la versión 7.1.23 (cli)

### 4.2.1. MariaDB

MariaDB es un sistema de base de datos procedente de MySQL. Este sistema está alojado en el servidor y será el encargado de almacenar en el servidor los datos generados por el cliente. Este sistema nos proporciona una persistencia y disponibilidad de los datos.

La petición de estos datos se realiza mediante la herramienta creada en PHP y únicamente podrá acceder a estos datos los clientes que realicen peticiones a través de la herramienta creada para tal fin.

La herramienta en PHP utiliza sentencias SQL para acceder a los datos.

### 4.2.3. Xampp

Xampp es una recopilación de herramientas de software libre destinada principalmente a crear un entorno de desarrollo local para aplicaciones y páginas web.

Contiene numerosas herramientas preparadas para comenzar a utilizarse justo después de la instalación sin la necesidad de tener que configurar ningún parámetro o ajuste.

Se trata de un programa multiplataforma disponible tanto para Linux, Windows o macOS.

Está enfocada como una herramienta de desarrollo por lo que la seguridad es inexistente a favor de disponer de un entorno de desarrollo dinámico y multifuncional.

Xampp contiene también un pequeño gestor de la base de datos MariaDB a través de una interfaz web llamada *phpMyAdmin* capaz de realizar gran cantidad de operaciones de gestión, administración y mantenimiento de las bases de datos

Entre sus herramientas nos encontramos:

* Servidor Apache
* Base de datos MariaDB
* PHP
* Perl

Al contener todas estas herramientas, las cuales se han descrito anteriormente, se han utilizado las mismas que contiene Xampp en su paquete. Por ello, con una simple instalación de Xampp, tendremos todo lo necesario para comenzar a construir la aplicación.

La versión utilizada para construir la aplicación es la versión 7.3.1 que contiene las versiones anteriormente mencionadas de las diferentes herramientas.

## **4.3. Librerías**

### 4.3.1. ngx-bootstrap

Se trata de una librería desarrollada y mantenida por *Valor Software* [[10]](#footnote-10) creada para trabajar con Angular la cual nos da posibilidad de crear distintos componentes con múltiples funcionalidades con el estilo de las hojas de estilo Bootstrap.

Como ejemplo de estos componentes están:

* Botones
* Alertas
* Selectores de fecha
* Modales
* Globos de consejos
* Barras de progreso

Esta librería se encuentra en el repositorio de Node.js y a través de el gestor de paquetes npm, puede ser descargada e instalada en el proyecto de Angular listo para utilizar.

Contiene una documentación muy extensa, bien presentada y sencilla de entender, con ejemplos para facilitar el uso de todos los componentes que ofrece.

Esta librería, en su versión 4.2 se encuentra bajo licencia MIT.

### 4.3.2. Chart.js

Se ha utilizado los componentes de la librería Chart.js [[11]](#footnote-11) para realizar la representación gráfica del progreso de los pacientes.

Chart.js contiene una serie de herramientas de código libre, destinadas a la creación de gráficas utilizando tecnología JavaScript y CSS. Contiene gran variedad de gráficas y permite modificar de una forma sencilla cualquiera de sus parámetros de datos y representación.

En su página web, existe una documentación muy cuidada, fácil de entender y con multitud de ejemplos.

Todas estas herramientas se encuentran en el repositorio de Node.js y a través de el gestor de paquetes npm, puede ser descargada e instalada en el proyecto de Angular.

En la aplicación se han utilizado muchas de estas gráficas y se han modificado muchos de sus parámetros para obtener la representación deseada.

Esta librería se encuentra bajo licencia MIT

### 4.3.3. Font Awesome

Algunos de los iconos que contiene la aplicación, han sido enlazados mediante la aplicación de la hoja de estilos CSS que proporciona *Font Awesome*[[12]](#footnote-12)*.*

Se han utilizado única y exclusivamente los iconos con una etiqueta *free*, los cuales se encuentran bajo licencia CC BY 4.0

## **4.4. Herramientas**

### 4.4.1. Visual Studio Code

Para la tarea de llevar a cabo la construcción de la aplicación se ha utilizado la herramienta creada por Microsoft, Visual Studio Code.

Se trata de un editor de código fuente capaz de integrar multitud de plug-ins entre los cuales se encuentran:

* Git
* Pistas, anotaciones y consejos para Typescript
* Pistas, anotaciones y consejos para PHP
* Pistas, anotaciones y consejos para SQL
* Pistas, anotaciones y consejos para CSS

Este editor de código facilita mucho el desarrollo ya que reconoce que se trata de un proyecto de Angular y organiza su estructura, iconos y sugerencias de código para este lenguaje de programación. Permite también abrir varios archivos y colocarlos en diferentes localizaciones, consiguiendo de esta manera que el proceso de codificación sea sencillo, rápido y te avisa de todos los errores sintácticos o de programación que se puedan dar.

Contiene al mismo tiempo unas normas de estilo que fuerzan a que el código sea limpio y uniforme.

Se ha utilizado Visual Studio Code para codificar todas las herramientas que contiene la aplicación, así como para el HTML o las hojas de estilo CSS.

### 4.4.2. Git

Durante todo el proceso de creación de la aplicación, se ha utilizado un control de versiones. En este caso el control de versiones ha sido Git sobre los servidores pertenecientes a *GitHub* [[13]](#footnote-13). Esto ha permitido llevar un control de los cambios realizados en el código de la aplicación, así como deshacer multitud de cambios y volver a versiones anteriores cuando ha sido necesario.

### 4.4.3. Postman

Debido a la creación de una herramienta de servicios en el lado del servidor, es necesario realizar peticiones al mismo para llevar un control sobre el funcionamiento de esta herramienta. La herramienta en la parte del servidor nos devuelve información ante diferentes peticiones del tipo *post*. Por ello necesitamos de otra herramienta capaz de enviar peticiones *post* a nuestro servidor para visualizar los datos devueltos y comprobar que funciona correctamente o depurar los datos.

Esta herramienta utilizada para tal fin se denomina *Postman*[[14]](#footnote-14)y es multiplataforma.

Postman ha sido muy utilizada, y una pieza clave a la hora de proporcionar agilidad y fiabilidad de los datos al desarrollo del proyecto.

# **5. Casos de uso**

El este apartado se enumeran los casos de uso para posteriormente, explicarlos detalladamente en tablas.

## **5.1. Definición de términos**

**Espacio** – Se considera espacio a la zona restringida a un profesional. Es absolutamente privado y únicamente accesible a cada profesional.

Este espacio contiene todos los datos, y herramientas para permitir al profesional trabajar con la aplicación.

**Profesional** – Se considera profesional al encargado de utilizar la aplicación, el profesional es el administrador único de su espacio en la aplicación y será el usuario principal de la aplicación.

**Paciente** - Se considera paciente a la entidad formada por un conjunto de datos que definen al paciente, cada paciente es único y pertenece a un espacio.

Los datos que conforman al paciente son generados únicamente por el profesional.

El paciente tiene múltiples atributos y objetos asociados a él.

**Código de registro** – Código único aleatorio generado por el administrador del sitio que permite dar de alta a un nuevo profesional.

**Dieta** - Se considera dieta a la entidad que define lo que debe ingerir un paciente.

La dieta es generada por un profesional y va asociada a un paciente.

Esta dieta se genera a partir de los datos que contiene un paciente.

**Datos** - Se consideran datos a todas aquellas características que un profesional ha asignado a cada paciente.

Los datos, se tienen en cuenta a la hora de crear dietas por parte del profesional.

**Restricciones de paciente** - Las restricciones de paciente son características del paciente que lo definen al igual que los datos y deben de ser siempre considerados a la hora de crear una dieta por parte del profesional. Por ejemplo, estas restricciones, pueden ser alergias, patologías, etc.

**Cita** – Evento en un calendario que define los detalles de una reunión entre un paciente y su profesional asociado.

**Medidas** – Hacen referencia a los datos antropométricos de cada paciente. Son el resultado de una consulta con el profesional el cual obtendrá estas medidas al realizar distintas mediciones a los pacientes.

**Métricas** – Son el resultado de los cálculos obtenidos a partir de las medidas y datos asociados a un paciente.

## **5.1. Casos de uso (índice)**

1. Registro profesional
2. Log-in profesional
3. Log-in paciente
4. Alta Paciente
5. Baja paciente
6. Añadir cita a paciente
7. Cancelar cita paciente
8. Buscar paciente
9. Añadir medidas a paciente
10. Consular histórico de medidas y métricas
11. Añadir patología a paciente
12. Eliminar patología a paciente
13. Añadir nueva patología a la lista de patologías
14. Ver progreso en graficas del paciente
15. Añadir alimentos a la dieta
16. Eliminar alimentos de la dieta
17. Crear dieta para paciente
18. Crear dieta predefinida de profesional
19. Cargar dieta predefinida a paciente
20. Editar dieta
21. Visualizar dieta
22. Paciente consulta su progreso
23. Paciente consulta su dieta
24. Paciente consulta su próxima cita
25. Contactar con el profesional

## **5.1. Casos de uso (Tablas)**

|  |  |  |
| --- | --- | --- |
| **CU – 01** | **Registro profesional** | |
| **Objetivos asociados** | Registrar en el sistema a un usuario con el rol de *profesional* | |
| **Entradas** | * Nombre * Correo electrónico * Contraseña | |
| **Salidas** | Mensaje de éxito | |
| **Precondición** | * Poseer código de registro * Código de registro registrado en la base de datos * Correo electrónico no registrado en base de datos | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| 1 | Seleccionar “Tengo un Código” |
| 2 | Introducir el código de registro único |
| 3 | Rellenar los datos |
| 4 | Validar |
| **Postcondición** | Nuevo profesional registrado en el sistema | |
| **Excepciones** | **Paso** | **Acción** |
| S-1 | Correo electrónico ya registrado en el sistema |
| **Comentarios** |  | |

|  |  |  |
| --- | --- | --- |
| **CU – 02** | **Log-in Profesional** | |
| **Objetivos asociados** | Acceder al sistema con el rol de profesional | |
| **Entradas** | * Correo electrónico * Contraseña | |
| **Salidas** | Vista principal de profesional | |
| **Precondición** | * Haber creado previamente una cuenta con rol profesional | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| 1 | Acceder a la pagina de log-in |
| 2 | Introducir credenciales |
| 3 | Validar |
| **Postcondición** | Usuario accede al sistema con rol de profesional teniendo disponibles todas las funcionalidades asociadas a este rol | |
| **Excepciones** | **Paso** | **Acción** |
| S-1 | Credenciales introducidas incorrectas |
| S-2 | Profesional no registrado en el sistema |
| **Comentarios** | Se cargará la vista correspondiente al profesional en la aplicación | |

|  |  |  |
| --- | --- | --- |
| **CU - 03** | **Log-in Paciente** | |
| **Objetivos asociados** | Acceder al sistema con el rol de paciente | |
| **Entradas** | * Correo electrónico * Contraseña | |
| **Salidas** | Vista principal paciente | |
| **Precondición** | * El profesional ha debido registrar en el sistema al paciente | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| 1 | Acceder a la pagina de log-in |
| 2 | Introducir credenciales |
| 3 | Validar |
| **Postcondición** | Usuario accede al sistema con rol de paciente teniendo disponibles todas las funcionalidades asociadas a este rol | |
| **Excepciones** | **Paso** | **Acción** |
| S-1 | Credenciales introducidas incorrectas |
| S-2 | Paciente no registrado en el sistema |
| **Comentarios** | Se cargará la vista del paciente en la aplicación | |

|  |  |  |
| --- | --- | --- |
| **CU - 04** | **Alta paciente** | |
| **Objetivos asociados** | Crea un nuevo paciente en la aplicación y se asociará al profesional que lo ha creado  Esta acción creará una cuenta con la que posteriormente el paciente podrá realizar log-in en la aplicación y acceder a su espacio.  Si el email del paciente ya existía, se volverá a activar su cuenta | |
| **Entradas** | * Nombre * Apellido 1 * Apellido 2 * Edad * Sexo * Email * Contraseña | |
| **Salidas** | Aparecerá la página del paciente creado | |
| **Precondición** | * Estar registrado como profesional | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| 1 | Seleccionar nuevo paciente |
| 2 | Rellenar campos disponibles de paciente |
| 3 | Validar |
| **Postcondición** | Paciente registrado en el sistema | |
| **Excepciones** | **Paso** | **Acción** |
| S-1 | Paciente ya existe |
| S-2 | Datos necesarios no proporcionados |
| **Comentarios** |  | |

|  |  |  |
| --- | --- | --- |
| **CU - 05** | **Baja paciente** | |
| **Objetivos asociados** | Desactiva un paciente | |
| **Entradas** | * Confirmación explicita de la eliminación | |
| **Salidas** | Confirmación de la desactivación del paciente | |
| **Precondición** | * Estar registrado como profesional * Acceder a ficha de paciente | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| 1 | Seleccionar paciente y acceder a su ficha |
| 2 | Seleccionar “Desactivar paciente” |
| 3 | Confirmar escribiendo “si” en el cuadro de texto proporcionado |
| 4 | Validar |
| **Postcondición** | Paciente marcado como baja de la base de datos | |
| **Excepciones** | **Paso** | **Acción** |
| S-1 | Profesional, introduce en cuadro de confirmación texto distinto de “si” |
| **Comentarios** |  | |

|  |  |  |
| --- | --- | --- |
| **CU - 06** | **Añadir cita a paciente** | |
| **Objetivos asociados** | Asocia una cita a un paciente | |
| **Entradas** | * Fecha * Hora | |
| **Salidas** | Mostrará su cita frente a su nombre en la lista de pacientes | |
| **Precondición** | * Estar registrado como profesional * Tener al menos un paciente registrado * El paciente está activo | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| 1 | Seleccionar paciente |
| 2 | Seleccionar “añadir cita” |
| 3 | Rellenar los campos de la cita |
| 4 | Validar |
| **Postcondición** | Nueva cita asociada a un paciente | |
| **Excepciones** | **Paso** | **Acción** |
| S-1 | No existe ningún paciente asociado |
| S-2 | Fecha y día no validos |
| **Comentarios** | Si se el paciente ya tiene asignado una cita, y se le asigna otra en otra fecha u hora diferente, la cita anterior se sobrescribirá | |

|  |  |  |
| --- | --- | --- |
| **CU - 07** | **Cancelar cita a paciente** | |
| **Objetivos asociados** | Elimina la cita de un paciente | |
| **Entradas** | Ninguna | |
| **Salidas** | Desaparecerá la fecha de cita asociada de la visualización | |
| **Precondición** | * Estar registrado como profesional * Tener al menos un paciente registrado * El paciente registrado está activo * El paciente registrado tiene una cita activa y asociada | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| 1 | Seleccionar paciente |
| 2.1 | Pinchar botón “Atendido” |
| 2.2 | Pinchar cruz roja al lado de la cita |
| **Postcondición** | Cita marcada como no activa en la base de datos | |
| **Excepciones** | **Paso** | **Acción** |
| S-1 | No existe ninguna cita asociada a un paciente |
| **Comentarios** |  | |

|  |  |  |
| --- | --- | --- |
| **CU - 08** | **Buscar paciente** | |
| **Objetivos asociados** | Buscar a un paciente por nombre o apellido en la lista de pacientes de un profesional | |
| **Entradas** | Nombre o apellido en el cuadro de búsqueda | |
| **Salidas** | Aparecerá en la lista de pacientes los nombres y apellidos que coincidan con los términos de búsqueda. | |
| **Precondición** | * Estar registrado como profesional * Tener al menos un paciente registrado | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| 1 | Escribir en el cuadro de búsqueda |
| **Postcondición** | La lista de pacientes mostrará los coincidentes | |
| **Excepciones** | **Paso** | **Acción** |
| S-1 | No existe ningún paciente registrado |
| **Comentarios** |  | |

|  |  |  |
| --- | --- | --- |
| **CU - 09** | **Añadir medidas a paciente** | |
| **Objetivos asociados** | Permite rellenar los datos de medidas pertenecientes a un paciente | |
| **Entradas** | * Peso * Altura * Pliegue tríceps * Pliegue cresta iliaca * Pliegue subescapular * Pliegue bíceps * Pliegue supraspina * Pliegue abdominal * Pliegue muslo * Pliegue pierna * Perímetro brazo relajado * Perímetro brazo flexionado * Perímetro cintura * Perímetro cadera * Perímetro pierna * Diámetro muñeca * Diámetro humero * Diámetro biepicondilar fémur | |
| **Salidas** | Métricas actualizadas en tiempo real | |
| **Precondición** | * Estar registrado como profesional * Tener un paciente registrado y asignado al profesional * Acceder a la ficha del paciente al cual se quieren modificar sus medidas | |
| **Secuencia**  **Normal** | **Paso** | **Paso** |
| 1 | Seleccionar al paciente |
| 2 | Modificar los diferentes campos |
| **Postcondición** | Nuevas medidas asociadas al paciente y añadidas a su histórico de medidas | |
| **Comentarios** |  | |

|  |  |  |
| --- | --- | --- |
| **CU - 10** | **Consultar histórico de medidas y métricas** | |
| **Objetivos asociados** | Consulta el histórico en forma de datos numéricos de medidas y métricas de un paciente | |
| **Entradas** | Seleccionar en la ficha del paciente la fecha del histórico que se desea consultar | |
| **Salidas** | Se rellenarán las medidas y métricas de acuerdo con los datos consultados | |
| **Precondición** | Estar registrado como profesional  Paciente activo  Tener un paciente asociado con al menos dos tomas de medidas en la base de datos asociadas a ese paciente | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| 1 | Seleccionar paciente a consultar |
| 2 | Seleccionar del histórico unas medidas pertenecientes a una fecha |
| **Postcondición** | Ninguna | |
| **Excepciones** | **Paso** | **Acción** |
| S-1 | No existen medidas asociadas al paciente |
| **Comentarios** | Las métricas se recalcularán en función de las medidas seleccionadas y se mostrará el cálculo en tiempo real al mismo tiempo que aparecen las medidas selecionadas | |

|  |  |  |
| --- | --- | --- |
| **CU - 11** | **Añadir patología a paciente** | |
| **Objetivos asociados** | Añadir una o más patologías a un paciente seleccionado | |
| **Entradas** | Selección de patologías deseadas | |
| **Salidas** | Se mostrará en la ficha del paciente las patologías asociadas | |
| **Precondición** | Estar registrado como profesional  Paciente activo  Existencia de patologías registradas en la base de datos | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| 1 | Seleccionar paciente |
| 2 | Pulsar en patologías |
| 3 | Asociar tantas patologías como se desee |
| 4 | Pulsar en “Guardar” para validar |
| **Postcondición** | Ninguna | |
| **Excepciones** | **Paso** | **Acción** |
| S-1 | Lista de patologías vacía |
| **Comentarios** |  | |

|  |  |  |
| --- | --- | --- |
| **CU - 12** | **Eliminar patología a paciente** | |
| **Objetivos asociados** | Eliminar una o más patologías a un paciente seleccionado | |
| **Entradas** | Selección de patologías deseadas | |
| **Salidas** | Se mostrará en la ficha del paciente las patologías asociadas | |
| **Precondición** | Estar registrado como profesional  Paciente activo  Existencia de patologías registradas en la base de datos  Existencia de patologías asociadas al paciente | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| 1 | Seleccionar paciente |
| 2 | Pulsar en patologías |
| 3 | Desmarcar tantas patologías como se desee |
| 4 | Pulsar en “Guardar” para validar |
| **Postcondición** | Ninguna | |
| **Excepciones** | **Paso** | **Acción** |
| S-1 | Lista de patologías vacía |
| **Comentarios** |  | |

|  |  |  |
| --- | --- | --- |
| **CU – 13** | **Añadir nueva patología a la lista de patologías** | |
| **Objetivos asociados** | Añadir una o más patologías a la base de datos | |
| **Entradas** | Nombre de la patología | |
| **Salidas** | Aparecerá en la lista de patologías | |
| **Precondición** | Estar registrado como profesional  Paciente activo | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| 1 | Seleccionar paciente |
| 2 | Pulsar en patologías |
| 3 | Escribir el nombre de la nueva patología en el cadro de texto |
| 4 | Pulsar en “+” para validar |
| **Postcondición** | Nueva patología en la base de datos  Nueva patología asociada al profesional que la ha creado | |
| **Excepciones** | **Paso** | **Acción** |
| S-1 | La patología ya existe |
| **Comentarios** | La nueva patología se asociará al profesional que la ha creado y únicamente será visible para el profesional en concreto.  Existen también patologías comunes a todos los profesionales que podrán ser introducidas por un técnico. | |

|  |  |  |
| --- | --- | --- |
| **CU – 14** | **Ver progreso en graficas del paciente** | |
| **Objetivos asociados** | Visualizar mediante gráficas el histórico de medidas y métricas del paciente | |
| **Entradas** | Ninguna | |
| **Salidas** | Se mostrará las gráficas del histórico | |
| **Precondición** | Estar registrado como profesional  Paciente activo  Medidas y métricas asociadas al paciente selecionado | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| 1 | Seleccionar paciente |
| 2 | Pulsar en progreso |
| 3 | Moverse por las diferentes pestañas |
| **Postcondición** | Ninguna | |
| **Excepciones** | **Paso** | **Acción** |
| S-1 | Paciente no tiene medidas asociadas |
| **Comentarios** |  | |

|  |  |  |
| --- | --- | --- |
| **CU – 15** | **Añadir alimentos a la dieta** | |
| **Objetivos asociados** | Añadir múltiples alimentos a una dieta | |
| **Entradas** | * Nombre del alimento * Cantidad * Unidades de la cantidad | |
| **Salidas** | Aparecerá el alimento añadido | |
| **Precondición** | Estar registrado como profesional | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| 1 | Pulsar sobre el botón “+” |
| 2 | Rellenar los datos de entrada |
| **Postcondición** | Alimento añadido a la dieta | |
| **Comentarios** | Si anteriormente se ha introducido un alimento en la base de datos, al escribir su nombre, si existe, se autocompletarán las unidades con las asociadas a ese alimento en la base de datos | |

|  |  |  |
| --- | --- | --- |
| **CU – 16** | **Eliminar alimentos a la dieta** | |
| **Objetivos asociados** | Eliminar un alimento que se ha añadido a nuestra dieta | |
| **Entradas** | Pulsar botón “-“ | |
| **Salidas** | Desaparecerá el alimento | |
| **Precondición** | Estar registrado como profesional  Alimento existente en la dieta | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| 1 | Pulsar sobre el botón “-” |
| **Postcondición** | Alimento eliminado de la dieta | |
| **Comentarios** |  | |

|  |  |  |
| --- | --- | --- |
| **CU – 17** | **Crear dieta para paciente** | |
| **Objetivos asociados** | Asociar una dieta a un paciente | |
| **Entradas** | * Dieta con al menos un alimento en una franja horaria * Nombre de la dieta (Opcional) | |
| **Salidas** | Visualización de la dieta en la lista de dietas del paciente | |
| **Precondición** | Estar registrado como profesional  Paciente seleccionado activo | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| 1 | Seleccionar el paciente deseado |
| 2 | Pulsar en dietas |
| 3 | Añadir/Eliminar alimentos en las franjas deseadas |
| 4 | Añadir un nombre (Opcional) |
| 5 | Pulsar en guardar |
| **Postcondición** | Nueva dieta registrada en la base de datos y asociada al paciente seleccionado | |
| **Comentarios** |  | |

|  |  |  |
| --- | --- | --- |
| **CU – 18** | **Crear dieta predefinida de profesional** | |
| **Objetivos asociados** | Asociar una dieta a un profesional | |
| **Entradas** | * Dieta con al menos un alimento en una franja horaria * Nombre de la dieta (Opcional) | |
| **Salidas** | Visualización de la dieta en la lista de dietas predefinas en la vista “Dietas de paciente” | |
| **Precondición** | Estar registrado como profesional | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| 1 | Seleccionar “Dietas predefinas” en la vista principal del profesional |
| 3 | Añadir/Eliminar alimentos en las franjas deseadas |
| 4 | Añadir un nombre (Opcional) |
| 5 | Pulsar en guardar |
| **Postcondición** | Nueva dieta registrada en la base de datos y asociada al profesional | |
| **Comentarios** | Esta dieta podrá ser cargada posteriormente en la acción “Crear dieta para paciente” con la ventaja de que ya posee multitud de alimentos precargados en franjas horarias | |

|  |  |  |
| --- | --- | --- |
| **CU – 19** | **Cargar dieta predefinida a paciente** | |
| **Objetivos asociados** | Rellenar los alimentos de una nueva dieta en sus franjas horarias correspondientes, los cuales pertenecen a una dieta predefinida por el profesional. | |
| **Entradas** | Seleccionar dieta de un histórico | |
| **Salidas** | Se rellenará la nueva dieta con los mismos datos que contiene la dieta predefinida | |
| **Precondición** | Estar registrado como profesional  Existencia de dietas predefinidas | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| 1 | Seleccionar de la lista de dietas predeterminadas la dieta |
| **Postcondición** | Nueva dieta poblada con datos de la dieta predefinida | |
| **Comentarios** | Se pueden añadir o eliminar nuevos alimentos a esta nueva dieta, así como darle un nuevo nombre | |

|  |  |  |
| --- | --- | --- |
| **CU – 20** | **Editar dieta** | |
| **Objetivos asociados** | Editar cualquier dieta creada anteriormente | |
| **Entradas** | Seleccionar la dieta deseada a editar | |
| **Salidas** | Nueva dieta poblada con datos de la dieta a editar seleccionada | |
| **Precondición** | Estar registrado como profesional  Existencia de dietas creadas asociadas al profesional o al paciente | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| 1 | Seleccionar dietas paciente o dietas predefinidas |
| 2 | Pulsar “editar” sobre la dieta deseada en la lista de dietas |
| 3 | Añadir o eliminar alimentos deseados |
| 4 | Pulsar en “Guardar” para validar y guardar la dieta editada |
| **Postcondición** | Ninguna | |
| **Comentarios** | La edición de una dieta, supone la creación de una nueva dieta con el objetivo de mantener un histórico de dietas | |

|  |  |  |
| --- | --- | --- |
| **CU – 21** | **Visualizar dieta** | |
| **Objetivos asociados** | Visualizar en un formato limpio cualquier dieta | |
| **Entradas** | * Dieta predefinida o dieta asociada a paciente desde la lista de dietas | |
| **Salidas** | Visualización de la dieta en un formato adecuado | |
| **Precondición** | Estar registrado como profesional o paciente | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| N1-1 | Como profesional, seleccionar el paciente deseado |
| N1-2 | Pulsar en dietas |
| N1-3 | Pulsar sobre cualquier dieta deseada |
| N2-1 | Estar registrado como paciente |
| N2-2 | Pulsar sobre mi dieta |
| **Excepciones** | **Paso** | **Acción** |
| S-1 | No existen dietas asociadas a profesional o a paciente |
| **Postcondición** | Nueva dieta registrada en la base de datos y asociada al paciente seleccionado | |
| **Comentarios** |  | |

|  |  |  |
| --- | --- | --- |
| **CU – 22** | **Paciente consulta su progreso** | |
| **Objetivos asociados** | Visualizar gráficamente el progreso de sus medidas y métricas asociadas a su histórico | |
| **Entradas** | Ninguna | |
| **Salidas** | Gráficas del histórico de su progreso | |
| **Precondición** | Estar registrado como paciente  Paciente activo  Existencia de medidas y métricas asociadas al paciente en la base de datos | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| 1 | Log-in como paciente |
| 2 | Navegar por las diferentes pestañas |
| **Postcondición** | Ninguna | |
| **Excepciones** | **Paso** | **Acción** |
| S-1 | No existen medidas y métricas asociadas al paciente |
| **Comentarios** |  | |

|  |  |  |
| --- | --- | --- |
| **CU – 23** | **Paciente consulta su dieta** | |
| **Objetivos asociados** | Visualizar en un formato limpio la dieta asociada al paciente | |
| **Entradas** | Pulsar sobre “Mi Dieta” | |
| **Salidas** | Dieta en bien formateada asociada al paciente | |
| **Precondición** | Estar registrado como paciente  Paciente activo  Existencia de dieta asociada al paciente en la base de datos | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| 1 | Log-in como paciente |
| 2 | Pulsar en “Mi Dieta” |
| **Postcondición** | Ninguna | |
| **Excepciones** | **Paso** | **Acción** |
| S-1 | No existe dieta asociada al paciente |
| **Comentarios** |  | |

|  |  |  |
| --- | --- | --- |
| **CU – 24** | **Paciente consulta su próxima cita** | |
| **Objetivos asociados** | Visualizar la próxima cita activa del paciente | |
| **Entradas** | Pulsar en “Mi Próxima Cita” | |
| **Salidas** | Visualización de los datos de la próxima cita asociada al paciente | |
| **Precondición** | Estar registrado como paciente  Paciente activo | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| 1 | Log-in como paciente |
| 2 | Pulsar en “Mi Próxima Cita” |
| **Postcondición** | Ninguna | |
| **Excepciones** | **Paso** | **Acción** |
| S-1 | No existe cita asociada al paciente |
| **Comentarios** | Si no existe una cita asociada al paciente, se mostrará un mensaje, anunciando que no tiene próximas citas | |

|  |  |  |
| --- | --- | --- |
| **CU – 25** | **Contactar con el profesional** | |
| **Objetivos asociados** | Escribir un correo electrónico al profesional asociado al paciente | |
| **Entradas** | Pulsar en “Contactar Nutricionista” | |
| **Salidas** | Se abrirá el administrador de correo electrónico disponible en la máquina del paciente con la dirección de destino correspondiente a su profesional | |
| **Precondición** | Estar registrado como profesional  Paciente activo  Existencia de un gestor de correo electrónico por defecto | |
| **Secuencia**  **Normal** | **Paso** | **Acción** |
| 1 | Log-in como paciente |
| 2 | Pulsar en “Contactar Nutricionista” |
| 3 | Escribir correo electrónico deseado |
| 4 | Enviar correo electrónico |
| **Postcondición** | Correo electrónico enviado | |
| **Excepciones** | **Paso** | **Acción** |
| S-1 | Gestor de correo electrónico no configurado correctamente |
| **Comentarios** |  | |

# **6. Modelo de datos**

La aplicación guarda todos los datos generados por el usuario o por la propia aplicación en una base de datos SQL. Todos estos datos son accesibles en todo momento a medida que la aplicación los requiera.

Los datos se encuentran separados en diferentes tablas pertenecientes a una única base de datos denominada *nutriGestion.* Las diferentes tablas, a su vez, contienen elementos que permiten relacionar las diferentes tablas entre sí*.* De esta manera, conseguimos que información contenida en una tabla, se complemente con la información contenida en otras tablas, y así, generar una estructura y un modelo de datos que represente la información deseada de manera completa.

A continuación, se describirá la estructura de la base de datos, el propósito y significado de las tablas creadas, el significado y la utilidad de sus campos, y se explicará mediante qué elementos se relacionan entre sí las diferentes tablas.

## **6.1. Estructura**

La estructura de la base de datos creada para la aplicación es la siguiente:

Imagen que contiene captura de pantalla

Descripción generada automáticamente

Figura 6 - Estructura de la base de datos

Tal y como se puede ver en la *figura 6*, nuestra aplicación está formada por doce tablas, cada una de estas tablas, contiene información muy concreta, la cual podemos ampliar mediante las relaciones marcadas con flechas en la *figura 6*. Estas flechas, que salen de los campos de las tablas, y van hasta otros campos en otras tablas, marcan mediante que elemento se relaciona con las demás.

La tabla **codigoregistro** no se relaciona con ninguna tabla debido a que es la encargada de almacenar un código de registro único. Teniendo nada más que la finalidad de almacenar un código de registro generado aleatoriamente, y ese código, será destruido una vez el profesional efectúe su registro en el sistema.

La tabla **paciente** podemos considerarla como la tabla principal del sistema, ya que, a través de un paciente, podemos conocer todos los datos de las demás tablas. Las claves principales de esta tabla son el *emailprofesional*, encargada de relacionar a un paciente con su **profesional**. Un paciente tiene asociado un único profesional a través del email y un profesional puede tener múltiples pacientes.

El *email* de la tabla paciente es la clave primaria que nos permite acceder al resto de información. Mediante esta clave email, podemos conocer información de la tabla **cita** que contiene el histórico de citas del paciente asociado al un email único de la tabla paciente. Una cita contiene un solo paciente mientras que un paciente puede tener asociado múltiples citas.

Podemos conocer toda la información acerca de **anatomía** y **métrica** de un paciente a través de la clave primaria email, ya que las tablas de anatomía y métrica contienen un histórico de datos asociados al email de la tabla paciente.

La tabla de **patología** se relaciona con el email de la tabla paciente a través de una tabla intermedia denominada **patología\_paciente**. De esta manera, la tabla patología\_paciente contiene un email que relaciona las tablas de paciente y patología\_paciente, y a su vez, contiene un *id* que asocia la tabla patología\_paciente con la tabla patología. De este modo podemos relacionar a un paciente con múltiples patologías y varias patologías podrán pertenecer a varios pacientes diferentes.

La tabla **dieta** mantiene una relación con el paciente mediante el campo *email*. El email de dieta corresponderá a un paciente. Un paciente puede tener múltiples dietas, pero a una dieta le corresponde únicamente un paciente.

La tabla dieta contiene siete campos representado los días de la semana, en estos campos, se insertará la clave primaria de la tabla **día**. La tabla día, contiene cinco campos los cuales representan las diferentes franjas horarias y contendrán las calves primarias que hacen referencia a la tabla **comida.** La tabla comida contiene un i*d* como clave primaria para ser referenciada, y un idAlimento que será el nexo con la tabla **alimento**. La tabla alimento, contiene como clave primaria un *id* para ser referenciada por la tabla de comida.

Debido a las características de la aplicación, esta es la estructura mediante la cual se relacionan las diferentes tablas de la base de datos, generando una coherencia y una estructura que facilita el añadir, y obtener información de la evolución de los datos almacenados mediante un histórico de datos.

## **6.2. Tablas**

### 6.2.1. codigoregistro

Esta tabla contiene los códigos de registro necesarios para permitir el registro de un profesional. Estos códigos son únicos y serán introducidos por el administrador de sistemas.

|  |  |  |
| --- | --- | --- |
| **Columna** | **Significado** | **Tipo de datos** |
| Código (PK) | Código único | varchar(10) |

### 6.2.2. profesional

Contiene la información que representa al profesional en la aplicación.

|  |  |  |
| --- | --- | --- |
| **Columna** | **Significado** | **Tipo de datos** |
| id | Código numérico del profesional | decimal(10,0) |
| nombre | Nombre del profesional | varchar(20) |
| apellido | Apellido del profesional | varchar(20) |
| email (PK) | Email del profesional | varchar(60) |
| password | Clave hash generada a partir de una contraseña introducida y concatenada con el campo *salt* y *pepper* | varchar(64) |
| salt | Cadena hash generada aleatoriamente que permite el cifrado y descifrado de la contraseña | varchar(60) |
| activo | Bit que indica si el profesional está activo o no | Bit(1) |
| token | Clave hash de tipo sha1 generada al concatenar el email, contraseña e email.  Tiene funciones a la hora de la autentificación | varchar(60) |

### 6.2.3. paciente

Contiene la información que representa al paciente en la aplicación.

La clave primaria *email* será la que nos permite conectar con todas las demás tablas respecto de un paciente.

|  |  |  |
| --- | --- | --- |
| **Columna** | **Significado** | **Tipo de datos** |
| id | Código numérico del paciente | decimal(10,0) |
| nombre | Nombre del paciente | varchar(20) |
| apellido | Apellido del paciente | varchar(20) |
| apellido2 | Segundo apellido del paciente | varchar(20) |
| edad | Edad del paciente | Decimal(4,0) |
| sexo | Sexo del paciente.  h-> hombre  m-> mujer | varchar(1) |
| email (PK) | Email del paciente | varchar(60) |
| password | Clave hash generada a partir de una contraseña introducida y concatenada con el campo *salt* y *pepper* | varchar(64) |
| activo | Bit que indica si el profesional está activo o no | Bit(1) |
| emailProfesional | Email del profesional que ha generado este paciente | varchar(60) |
| salt | Cadena hash generada aleatoriamente que permite el cifrado y descifrado de la contraseña | varchar(60) |
| token | Clave hash de tipo sha1 generada al concatenar el email, contraseña e email.  Tiene funciones a la hora de la autentificación | varchar(60) |

### 6.2.4. cita

Contiene la información que representa las citas de un paciente

|  |  |  |
| --- | --- | --- |
| **Columna** | **Significado** | **Tipo de datos** |
| email | Email del paciente | varchar(60) |
| fecha | Contiene la fecha de la cita en formato UTC | datetime |
| activo | Bit que indica si la cita está activa o no | Bit(1) |

### 6.2.5. patología\_paciente

Contiene información acerca de la relación de las tablas *6.2.3 paciente* y *6.2.6 patología*

|  |  |  |
| --- | --- | --- |
| **Columna** | **Significado** | **Tipo de datos** |
| id | Referencia al id de la tabla 6.2.6 patología | Decimal(10,0) |
| email | Email del paciente | varchar(60) |

### 6.2.6. patología

Contiene el nombre de una patología y el email del profesional que la ha creado.

Un profesional puede añadir distintas patologías. Al almacenar el email del profesional que la ha creado, podemos hacer accesibles esas patologías únicamente al profesional que las ha creado.

Si contiene por email *“all”* o el campo nombre está vacío, las patologías serán comunes a todos los profesionales.

|  |  |  |
| --- | --- | --- |
| **Columna** | **Significado** | **Tipo de datos** |
| Id(PK) | Identificador único de una patología | Decimal(10,0) |
| email | Email del paciente creador de la patología | varchar(60) |
| nombre | Nombre de la patología | varchar(50) |

### 6.2.7. anatomía

Representa las medidas de un paciente y forma parte de su histórico al almacenar la fecha y está marcada como contemporánea si tiene marcado el bit de activo a 1.

Estas medidas de la anatomía de un paciente, son introducidas por un profesional a través de la aplicación.

|  |  |  |
| --- | --- | --- |
| **Columna** | **Significado** | **Tipo de datos** |
| email | Email del paciente al que pertenecen los datos | varchar(60) |
| peso | peso | Decimal(8,3) |
| altura | altura | Decimal(8,3) |
| PLtriceps | Pliegue de tríceps | Decimal(8,3) |
| PLcrestailiaca | Pliegue de creta iliaca | Decimal(8,3) |
| PLsubescapular | Pliegue subescapular | Decimal(8,3) |
| PLbiceps | Pliegue de bíceps | Decimal(8,3) |
| PLsupraespinal | Pliegue supraespinal | Decimal(8,3) |
| PLabdominal | Pliegue abdominal | Decimal(8,3) |
| PLmuslo | Pliegue muslo | Decimal(8,3) |
| PLpierna | Pliegue de pierna | Decimal(8,3) |
| PRbrazoRelajado | Perímetro brazo relajado | Decimal(8,3) |
| PRbrazoFlexionado | Perímetro brazo flexionado | Decimal(8,3) |
| PRcintura | Perímetro de la cintura | Decimal(8,3) |
| PRcadera | Perímetro de la cadera | Decimal(8,3) |
| PRpierna | Perímetro de la pierna | Decimal(8,3) |
| Dmuneca | Diámetro de la muñeca | Decimal(8,3) |
| Dhumero | Diámetro del húmero | Decimal(8,3) |
| DbiepicondilarFemur | Diámetro biepicondilar del Fémur | Decimal(8,3) |
| fechaModificacion | Fecha de la introducción de la anatomía | datetime |
| activo | La medida es la más reciente para el paciente | bit |

### 6.2.8. métrica

Representa las métricas de un paciente y forma parte de su histórico al almacenar la fecha y está marcada como contemporánea si tiene marcado el bit de activo a 1.

Estas métricas son automáticamente calculadas por la aplicación en función de los demás datos del paciente.

|  |  |  |
| --- | --- | --- |
| **Columna** | **Significado** | **Tipo de datos** |
| email | Email del paciente al que pertenecen los datos | varchar(60) |
| Imc | Índice de masa corporal | Decimal(8,3) |
| RatioCinturaCadera | Cociente entre la cintura y la cadera | Decimal(8,3) |
| Suma6Pliegues | Suma de los 6 pliegues | Decimal(8,3) |
| Suma8Pliegues | Suma de los 8 pliegues | Decimal(8,3) |
| PorcentGrasa | Porcentaje de grasa | Decimal(8,3) |
| PorcentOsea | Porcentaje óseo | Decimal(8,3) |
| PorcentMuscular | Porcentaje muscular | Decimal(8,3) |
| PorcentResidual | Porcentaje residual | Decimal(8,3) |
| MasaGrasa | Masa grasa | Decimal(8,3) |
| MasaOsea | Masa ósea | Decimal(8,3) |
| MasaMuscular | Masa muscular | Decimal(8,3) |
| MasaResidual | Masa residual | Decimal(8,3) |
| Somatotipo | Somatotipo corporal | Decimal(8,3) |
| Endomorfo | Índice de endomorfo | Decimal(8,3) |
| Mesomorfo | Índice de mesomorfo | Decimal(8,3) |
| Ectomorfo | Índice de ectomorfo | Decimal(8,3) |
| fechaModificacion | Fecha de la introducción de la métrica | datetime |
| activo | La métrica es la más reciente para el paciente | bit |

### 6.2.9. dieta

Representación de una dieta en nuestra base de datos.

|  |  |  |
| --- | --- | --- |
| **Columna** | **Significado** | **Tipo de datos** |
| Id(PK) | Identificador único de una dieta | Decimal(30,0) |
| emailPaciente | Email del paciente al que pertenecen la dieta | varchar(60) |
| lunes | Contendrá un *id* referente al *id* de la tabla *6.2.10 día* | Decimal(30,0) |
| martes | Contendrá un *id* referente al *id* de la tabla *6.2.10 día* | Decimal(30,0) |
| miercoles | Contendrá un *id* referente al *id* de la tabla *6.2.10 día* | Decimal(30,0) |
| jueves | Contendrá un *id* referente al *id* de la tabla *6.2.10 día* | Decimal(30,0) |
| viernes | Contendrá un *id* referente al *id* de la tabla *6.2.10 día* | Decimal(30,0) |
| sabado | Contendrá un *id* referente al *id* de la tabla *6.2.10 día* | Decimal(30,0) |
| domingo | Contendrá un *id* referente al *id* de la tabla *6.2.10 día* | Decimal(30,0) |
| nombre | Nombre perteneciente a dicha dieta | varchar(35) |
| fecha | Fecha de creación de la dieta | datetime |
| activo | La métrica es la más reciente para el paciente | bit |

### 6.2.10. día

Representación de un día en nuestra base de datos.

Cada franja horaria, contendrá un id correspondiente a una comida y a su vez es referenciado a través del id por la tabla *6.2.9 dieta.*

|  |  |  |
| --- | --- | --- |
| **Columna** | **Significado** | **Tipo de datos** |
| Id(PK) | Identificador único de un día | Decimal(30,0) |
| desayuno | Contendrá un *id* referente al *id* de la tabla *6.2.11 comida* | Decimal(30,0) |
| postdesayuno | Contendrá un *id* referente al *id* de la tabla *6.2.11 comida* | Decimal(30,0) |
| comida | Contendrá un *id* referente al *id* de la tabla *6.2.11 comida* | Decimal(30,0) |
| merienda | Contendrá un *id* referente al *id* de la tabla *6.2.11 comida* | Decimal(30,0) |
| cena | Contendrá un *id* referente al *id* de la tabla *6.2.11 comida* | Decimal(30,0) |

### 6.2.11. comida

Representación de una comida en nuestra base de datos.

Es la encargada de establecer las referencias entre una comida y los alimentos que la contienen.

Una comida puede tener varios alimento, y un alimento puede ser referenciado en varias comidas.

Contiene un id al que referencia una comida y se identifica un la tabla *6.2.12 alimento* a través de idAlimeto.

|  |  |  |
| --- | --- | --- |
| **Columna** | **Significado** | **Tipo de datos** |
| Id(PK) | Identificador único de una comida | Decimal(30,0) |
| idAlimento | Contendrá un *id* referente al *id* de la tabla *6.2.12 alimento* | Decimal(30,0) |
| postdesayuno | Contiene la cantidad en forma numérica del alimento referenciado en la tabla *6.2.12 alimento* | Decimal(8,2) |

1. http://nutrium.io [↑](#footnote-ref-1)
2. https://nutritioapp.com [↑](#footnote-ref-2)
3. www.angular.io [↑](#footnote-ref-3)
4. www.Google.com [↑](#footnote-ref-4)
5. Single page application [↑](#footnote-ref-5)
6. www.microsoft.com [↑](#footnote-ref-6)
7. www.twitter.com [↑](#footnote-ref-7)
8. Navegador web desarrollado y mantenido por Google [↑](#footnote-ref-8)
9. Long Time Support [↑](#footnote-ref-9)
10. www.valor-software.com [↑](#footnote-ref-10)
11. www.chartjs.org [↑](#footnote-ref-11)
12. www.fontawesome.com [↑](#footnote-ref-12)
13. www.github.com/icanas/nutrigestion [↑](#footnote-ref-13)
14. www.getpostman.com [↑](#footnote-ref-14)