



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Aplicación Web de planificación nutricional de usuarios mediante preferencias personalizadas

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Daniel Aibar Giménez

Tutor: Victoria Torres Bosch
Manoli Albert Albiol

2019/2020

Resumen

En este trabajo final de grado se busca desarrollar un sistema de planificación nutricional para que cualquier profesional dedicado al mundo de la nutrición pueda llevar una gestión automatizada de sus clientes, indistintamente del tipo de nutrición que lleve el cliente. Esta aplicación permitirá a los clientes de los nutricionistas generar una plantilla personalizada con sus preferencias nutricionales.

Para desarrollar esta aplicación, se va a crear una solución web que se adapte a diferentes dispositivos, donde el cliente podrá consultar su planificación nutricional además de permitir gestionar otras funcionalidades como la selección de recetas.

El nutricionista accederá a la gestión en un apartado diferente donde tendrá acceso a la cartera de clientes y el listado de recetas que haya creado para los usuarios.

Para la elaboración de esta aplicación, se partirá de un análisis de requisitos, seguido de una planificación y un análisis de la metodología a utilizar.

Esta aplicación se adaptará a las necesidades de diferentes perfiles de clientes con diversos estilos de nutrición, ya sean vegetarianos, intolerantes o alérgicos a diferentes alimentos, etc... Obteniendo finalmente un sistema que ayude a los nutricionistas a realizar su trabajo.

Palabras clave: planificación, nutricional, automatizada, aplicación, web.

Abstract

In this final degree project, it seeks to develop a nutritional planning system so that any professional dedicated to the world of nutrition can carry out automated management of their clients, regardless of the type of nutrition that the client carries. This application will allow clients of nutritionists to generate a personalized template with their nutritional preferences.

To develop this application, a web solution will be created that adapts to different devices, where the client will be able to consult their nutritional planning in addition to allowing them to manage other functionalities such as the selection of recipes.

The nutritionist will access management in a different section where he will have access to the client portfolio and the list of recipes that he has created for users.

For the preparation of this application, it will start from an analysis of requirements, followed by planning and an analysis of the methodology to be used.

This application will adapt to the needs of different customer profiles with different styles of nutrition, whether vegetarian, intolerant or allergic to different foods, etc ... Finally obtaining a system that helps nutritionists to do their job.

Keywords: planning, nutritional, automated, application, web.

Resum

En aquest treball final de grau es busca desenvolupar un sistema de planificació nutricional per a que qualsevol professional dedicat al món de la nutrició pugui portar una gestió automatitzada dels seus clients, indiferentment de la nutrició que porte el client. Aquesta aplicació permetrà als clients dels nutricionistes generar una plantilla personalitzada amb les seues preferències nutricionals.

Per a desenvolupar aquesta aplicació, es crearà una solució web que s'adapte a diferents dispositius, on el client podrà consultar la seua planificació nutricional a més de permetre gestionar altres funcionalitats com la selecció de receptes.

El nutricionista accedirà a la gestió en un apartat diferent on tindrà accés a la cartera de clients i el llistat de receptes que haja creat per als usuaris.

Per a l'elaboració d'aquesta aplicació, es partirà d'un anàlisi de requisits, seguit d'una planificació i un anàlisi de la metodologia a utilitzar.

Aquesta aplicació s'adaptarà a les necessitats de diferents perfils de clients amb diversos estils de nutrició, ja siguin vegetarians, intolerants o al·lèrgics a diferents aliments, etc... Obtenint finalment un sistema que ajude als nutricionistes a fer el seu treball.

Paraules clau: planificació, nutricional, automatitzada, aplicació, web.

Agradecimientos

Mis primeras palabras de agradecimiento quiero dárselas a mi abuela Carmen, ella fue la primera persona que creyó en mí tras decidir entrar a estudiar este Grado de Ingeniería Informática y estuvo apoyándome cada año a dar la mejor versión de mí, creo que sin su apoyo no hubiera estado ahora mismo entregando este documento, por ello, quiero que sepa que estaré eternamente agradecido de ser su nieto.

Como no, a mis padres, que me han aguantado mis épocas de exámenes durante estos años, y viéndolo con perspectiva creo que no me aguantaría ni yo mismo.

También quiero agradecer a mis dos tutoras Manoli y Victoria por el apoyo dado durante las últimas semanas, guiándome en todo momento para la correcta elaboración de este documento.

Por último, pero no menos importante, quería agradecer a todos mis amigos y compañeros de trabajo que me han aportado su granito de arena en los momentos que más los necesitaba, ya sea ayudándome con ideas para este proyecto, como sacándome a tomar una cerveza cuando empezaba a salir humo de mi cabeza.

Tabla de contenidos

1. Introducción	11
1.1 Motivación.....	11
1.2 Objetivos	12
1.3 Distribución de la memoria.....	12
2. Estado del arte	13
2.1 MiDietario.....	14
2.2 anamatedietista.com	15
2.3 YUKA.....	15
2.4 YAZIO: Contador de Calorías para Adelgazar y Dieta.....	16
2.5 Resumen del Arte	17
3. Fase de análisis	19
3.1 Requisitos funcionales.....	19
3.1.1 Inicio de sesión / Gestión de usuarios	19
3.1.2 Cerrar sesión	20
3.1.3 Perfil del usuario.....	20
3.1.4 Configuración del perfil de usuario	20
3.1.5 Plan nutricional.....	21
3.1.6 Selector Nutricional	21
3.1.7 Panel de Administración.....	22
3.1.8 Panel de gestión de usuarios	22
3.1.9 Panel de registro de usuarios.....	22
3.1.10 Panel de gestión de recetas	22
3.2 Requisitos no funcionales	23
3.2.1 RNF1 Acceso Multiplataforma.....	23
3.2.2 RNF2 Seguridad	23
3.2.3 RNF3 Usabilidad	23
3.2.4 RNF4 Rendimiento	23
3.3 Casos de uso	24
3.3.1 Usuario / Cliente.....	24
3.3.2 Usuario Administrador / Nutricionista	25
3.4 Metodología.....	25
3.5 Identificación y análisis de posibles soluciones.....	27

3.5.1 Servidor Web.....	28
3.5.2 Alojamiento Web	28
3.6 Buenas prácticas de codificación	28
3.6.1 Universalizar las variables	29
3.6.2 Comentar el código.....	29
3.6.4 Indentaciones	29
3.6.5 Control de versiones.....	30
4. Fase de diseño	31
4.1 Arquitectura.....	31
4.2 Diseño detallado.....	31
4.2.1 Modelo conceptual.....	31
4.2.3 Diseño de la interfaz de usuario.....	34
5. Fase de Implementación.....	41
5.1 Tecnologías	41
5.2 Configuración del entorno	44
5.2.1 Organización de los ficheros	44
5.2.3 XAMPP – Servidor web local.	46
5.2.2 Configuración MySQL.	47
5.3 Ficheros de funcionalidad de la aplicación.....	48
5.3.1 Fichero de conexión conn.php.....	48
5.3.2 Fichero de inicio de sesión login.php.....	48
5.3.3 Capa usuario	50
5.3.4 Capa Administrador	64
6. Conclusiones.....	69
7. Futuras mejoras.....	71
8. Bibliografía.....	73

Índice de figuras

Fig. 1: Panel de MiDietario	14
Fig. 2: Dieta Generada por MiDietario	14
Fig. 3: Panel de Inicio de anamatedietista	15
Fig. 4: Interfaz de la aplicación YUKA.....	15
Fig. 5: Interfaz de la aplicación YAZIO.....	16
Fig. 6: Casos de Uso del usuario cliente.	24
Fig. 7: Casos de Uso del usuario administrador.....	25
Fig. 8: Tareas para el desarrollo de la aplicación.	26
Fig. 9: Desglose de horas por cada tarea	26
Fig. 10: Ejemplo de Indentaciones	29
Fig. 11: Repositorio GIT donde se encuentra la aplicación.	30
Fig. 12: Arquitectura de la aplicación.	31
Fig. 13: Estructura de la Base de Datos	32
Fig. 14: Wireframe de la pantalla de inicio.....	35
Fig. 15: Wireframe de la pantalla de Inicio de Sesión	35
Fig. 16: Wireframe del panel de usuario.....	36
Fig. 17.: Wireframe del panel de actualizar perfil	36
Fig. 18: Wireframe del panel de creación de la Dieta.....	37
Fig. 19: Wireframe del panel de Consulta de la Planificación.....	37
Fig. 20: Wireframe del panel de Selector.	38
Fig. 21: Wireframe del panel de inicio de Administrador.	38
Fig. 22: Wireframe del panel de edición de perfil del Administrador.	39
Fig. 23: Wireframe del panel de Recetas	39
Fig. 24: Wireframe del panel de creación de Recetas.	40
Fig. 25: Imagen conceptual de la funcionalidad de online.visual.paradigm.	41
Fig. 26: Logotipo de PHP	41
Fig. 27: Logotipo de JavaScript.	42
Fig. 28: Logotipo de jQuery	42
Fig. 29: Logotipo de MySQL.....	43
Fig. 30: Logotipo de Git.	43
Fig. 31: Panel principal de Git Extensions.....	44
Fig. 32: Organización de lo ficheros.	44
Fig. 33: Ficheros dentro del directorio app	45
Fig. 34: Ficheros del directorio security	45
Fig. 35: Resultado de buscar XAMPP en Google.....	46
Fig. 36: XAMPP en funcionamiento.....	46
Fig. 37: Creación de una nueva tabla de Base de Datos	47
Fig. 38: Asignación del nombre de la Base de Datos.	47
Fig. 39: Fichero de conexión de la Base de Datos	48
Fig. 40: Interfaz de Inicio de sesión.	48
Fig. 41: Código ejecutado para el Inicio de Sesión.	49
Fig. 42.: Interfaz de creación de un Plan Nutricional.	50
Fig. 43: Código del funcionamiento de creación de un Plan Nutricional.....	51
Fig. 44: Código del funcionamiento de creación de un Plan Nutricional 2	52

Fig. 45: Código del funcionamiento de creación de un Plan Nutricional 3.	53
Fig. 46: Código del funcionamiento de creación de un Plan Nutricional 4.	54
Fig. 47: Código del funcionamiento de creación de un Plan Nutricional 5.	55
Fig. 48: Interfaz de consulta del Plan Nutricional.	56
Fig. 49: Interfaz de Edición del perfil del usuario.	57
Fig. 50: Código de edición del perfil del usuario.	58
Fig. 51: Interfaz de consulta del perfil de usuario.	58
Fig. 52: Interfaz de edición de seguridad del usuario.	59
Fig. 53: Código de edición de la seguridad del usuario.	59
Fig. 54: Interfaz del selector de recetas.	60
Fig. 55: Código HTML del selector de recetas.	60
Fig. 56: Código jQuery para el selector de recetas.	61
Fig. 57: Código de inicio del selector de recetas.	62
Fig. 58: Código del selector de recetas 2.	62
Fig. 59: Interfaz del selector de recetas al realizar el código.	62
Fig. 60: Código del selector de recetas al dar like.	63
Fig. 61: Código PHP del selector de recetas al dar like	64
Fig. 62: Interfaz de inicio del Administrador.	64
Fig. 63: Interfaz de consulta de usuarios.	65
Fig. 64: Interfaz de registro de usuario.	65
Fig. 65: Código de registro de usuario.	66
Fig. 66: Interfaz de consulta de recetas.	67
Fig. 67: Código de creación de receta.	67
Fig. 68: Código de creación de receta 2.	68

1. Introducción

Llega un momento en la vida de muchas personas donde se cuestionan si su alimentación es lo suficientemente saludable. Muchas empiezan sin darse cuenta al entrar al supermercado y ver la información nutricional de cada alimento que vaya a comprar, buscar en internet recetas “saludables” (alias quema grasas, depurativas, con fibra) para llegar al verano con un tipo fantástico o incluso el “no ceno que luego engordo”. Cuando esto sucede; esa persona ha pasado a una fase en la que da más importancia a mantener su línea que a otras cosas.

Actualmente vivimos en la sociedad de la información, donde tenemos acceso a cualquier tipo de contenido. No obstante, cuando hablamos con una persona sobre la posibilidad de acudir a un nutricionista con el objetivo de mejorar sus hábitos alimenticios, suelen pensar que se trata de un derroche económico, infiriendo en los errores más comunes como: “Tengo una amiga que entiende de esto” o “No quiero arroz y pollo todos los días”.

Cuando sucede esto, la probabilidad de que un usuario se salte su planificación nutricional suele ser bastante alta, dado que él mismo se impone sus reglas, pero no sabe ajustarse a las necesidades reales de su cuerpo. Es cuando la labor del nutricionista cobra sentido, dado que este analizará las necesidades de esta persona e intentará adaptar esta nutrición para que pueda conseguir los objetivos marcados.

Con la tecnología actual, la labor del nutricionista se ha podido facilitar en muchos aspectos, pero en muchas ocasiones aún no se ha podido facilitar esa labor. Por ello, en este Trabajo Final de Grado nace con la finalidad de facilitar la labor de estos profesionales.

1.1 Motivación

Actualmente en el campo de la nutrición existen diferentes tipos de alimentación. Esto hace que muchas veces la labor del nutricionista tenga que abarcar muchos conocimientos extensos, como norma general, o se dedica a un campo en específico o puede enfrentarse a la pérdida de muchos clientes por este desconocimiento.

La aplicación desarrollada en este Trabajo Final de Grado pretende ser una herramienta de ayuda a la labor del nutricionista, ayudándole a gestionar diferentes planificaciones. Además, será una solución útil para el usuario donde podrá gestionar su planificación nutricional en base a sus preferencias.

1.2 Objetivos

Como objetivo principal, se desarrollará una aplicación que permita a los usuarios de un profesional de la nutrición poder gestionar la planificación alimentaria en base a sus preferencias y objetivos, ofreciendo recetas específicas a estos parámetros, además de poder llevar un control diario de su alimentación para lograr sus metas.

Además, para el nutricionista contará con una herramienta que le permita tener un banco de recetas con diferentes configuraciones para adaptarse a las diversas preferencias alimentarias, ofreciendo a sus clientes un trato personalizado.

Para agilizar todo esto, esta aplicación deberá adaptarse a cualquier plataforma, para que el usuario pueda acceder desde cualquier dispositivo además de contar con una interfaz amigable, que facilitará el acceso a los puntos clave de la aplicación.

1.3 Distribución de la memoria

A continuación, se detallarán los capítulos de la presente memoria

- Capítulo 1: *Introducción* - Es el capítulo en el que nos encontramos, donde se tendrá una introducción del proyecto, sus objetivos y una información de la estructura de la memoria.
- Capítulo 2: *Estado del Arte* - En este capítulo se analizará el estado del arte de diferentes aplicaciones que se tendrán de referencia, analizando sus puntos fuertes, y sus carencias, para poder identificar las posibles necesidades a incluir en la aplicación.
- Capítulo 3: *Fase de Análisis* - A partir del estudio realizado sobre el estado del arte, en esta sección se empezará con el proceso de desarrollo siguiendo una metodología *top-down*. En particular, aquí se abordará la fase de identificación de requisitos mediante la utilización de casos de uso.
- Capítulo 4: *Fase de Diseño* - En este capítulo se detalla la fase de análisis y diseño donde se lleva a cabo una descripción más detallada del sistema a partir de los casos de uso recogidos durante la fase de requisitos.
- Capítulo 5: *Fase de Implementación* - En este capítulo se detallará la implementación de la aplicación, donde se mostrará el funcionamiento de los diferentes componentes de la aplicación.
- Capítulo 6: *Conclusiones* - Se realizará un desarrollo mediante un informe de pruebas de accesibilidad de la aplicación además de comprobar las buenas prácticas aplicadas.

2. Estado del arte

En la actualidad, se pueden encontrar diferentes aplicaciones de nutrición online, ofreciendo distintos servicios al usuario pudiendo diferenciar entre dos tipos:

- 1) Aplicaciones o páginas web con una suscripción mensual, en el cual un nutricionista te pondrá una dieta, o dará acceso a la generación de una. Generalmente estos servicios tienen su propia biblioteca de información nutricional y cuentan en algún caso con una consulta online con un profesional, como por ejemplo: **MiDietario**, **DietaPlan**.
- 2) Aplicaciones de gestión nutricional, actualmente cuentan con una gran cuota de mercado, dado que tienen una gran fuente de datos nutricionales. Dan acceso gratuitamente a un usuario a la autogestión de su nutrición, pudiendo así introducir lo que va a consumir, así mismo, realizan el cálculo de las calorías que lleva y de las que necesita en su día a día, como por ejemplo: **YUCA**, entre otras.

La aplicación contará con una solución combinada entre los dos tipos, teniendo como público objetivo el sector profesional nutricionista, pero, dando el control alimentario al usuario pudiendo configurar su alimentación dadas sus preferencias, además de su predilección a los platos recomendados por el nutricionista.

Se evaluarán diferentes aplicaciones dentro de estos dos tipos para analizar qué ventajas y desventajas se observan en cada solución examinada.

2.1 MiDietario

MiDietario es una de las aplicaciones más conocidas en el ámbito de los profesionales de la nutrición. Esta aplicación está en el mercado desde hace más de 10 años sin recibir actualizaciones, esta aplicación otorga al nutricionista un programa donde gestionar perfiles de usuario y generar una dieta personalizada a cada usuario gracias a la gran base de datos que tiene almacenada, además de poder agregar nuevos alimentos como se muestra en la figura 1. Una de sus características principales es que puedes descargar estas dietas en *EXCEL* o en *PDF*.

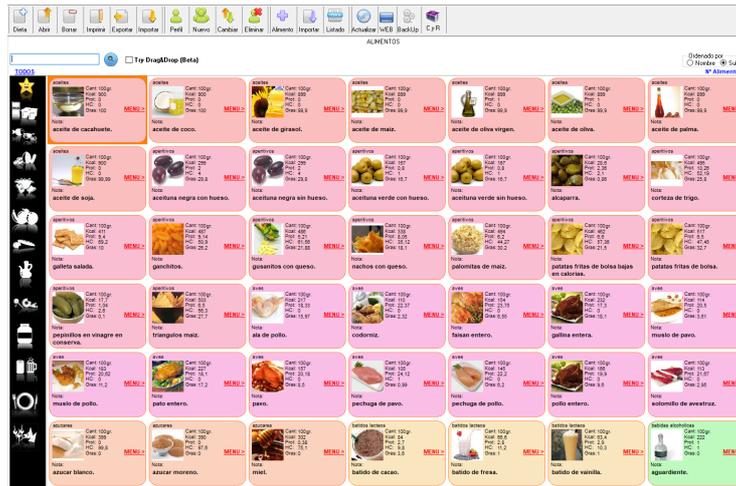


Fig. 1: Panel de MiDietario

Como puntos negativos, al tratarse de una aplicación con bastantes años a sus espaldas, únicamente la podemos encontrar para la plataforma de Windows, impidiendo su acceso desde otros sistemas operativos o plataformas. Además, limita al nutricionista a tener que agregar todos los alimentos que vayan a asociarse a una receta uno a uno como se muestra en la figura 2, dificultando la tarea de crear un plan específico a cada usuario dado que tiene que ir de dieta en dieta.

Dieta 1	Dieta 2	Dieta 3	Dieta 4	Dieta 5	Dieta 6	Dieta 7
Título Descripción 20190329-2 Dieta generada automáticamente el 29/03/2019						
COMIDA 1 10:00						
100 gr. pan blanco tipo baguett	Kcal	Prot	Gras	HC	IG	
5 gr. zumo de tomate comercial	258	9,65	1,10	50,60	0	
30 gr. jamon serrano.	1	0,03	0,00	0,13	0	
	41	6,41	1,68	0,00	0	
	300	16,09	2,78	50,73		
Comida 14:00						
300 gr. muslo de pollo.	Kcal	Prot	Gras	HC	IG	
500 gr. yogur desnatado natura	549	61,86	33,60	0,00	0	
	224	21,25	1,60	31,50	0	
	773	83,11	35,20	31,50		
Merienda 17:00						
500 gr. manzana.	Kcal	Prot	Gras	HC	IG	
	270	1,55	1,80	57,00	0	
	270	1,55	1,80	57,00		
Cena 22:00						
500 gr. ternera magra.	Kcal	Prot	Gras	HC	IG	
	655	103,50	27,00	5,00	0	
	655	103,50	27,00	5,00		
TOTAL	1998	204	67	144		

Fig. 2: Dieta Generada por MiDietario

2.2 anamatedietista.com

Como un ejemplo de plataforma online se puede indicar **anamatedietista**, esta plataforma web como se muestra en la figura 3, que concretamente es un generador de dietas online, puede verse muy parecido a lo que queremos crear con la aplicación que queremos desarrollar, pero a lo largo de este trabajo explicaremos las grandes diferencias que tenemos sobre otras aplicaciones.



Fig. 3: Panel de Inicio de anamatedietista

Esta aplicación web, viene a dar un acceso por suscripción a un cliente para que introduzca sus datos y pueda generarse una dieta para perder o ganar peso, pero obviando ciertos fundamentos como la predilección del cliente a ciertos alimentos, o el estilo de dieta que quiere el cliente, como por ejemplo un usuario que sea vegano. Si un cliente de esta aplicación quiere un análisis específico, deberá adquirir una suscripción con un coste superior.

2.3 YUKA

YUKA se ha convertido sin duda en una de las aplicaciones con más descargas en la Play Store para el cálculo de calorías de los alimentos.

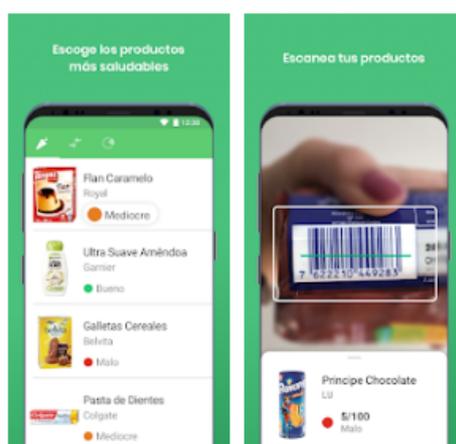


Fig. 4: Interfaz de la aplicación YUKA

Esta aplicación cuenta con toda la información nutricional de cada alimento que se busque ya sea por el nombre o por el código de barras como se muestra en la figura 4. Además, indica si se trata de un alimento saludable o no según las opiniones de los usuarios de este producto.

Como parte negativa, esta aplicación solo informa al usuario de la composición del artículo, y no da más detalles que las calorías que deba consumir el usuario, haciendo con ello que el usuario tenga que realizar la autogestión de su consumo calórico diario sin ningún otro control.

2.4 YAZIO: Contador de Calorías para Adelgazar y Dieta

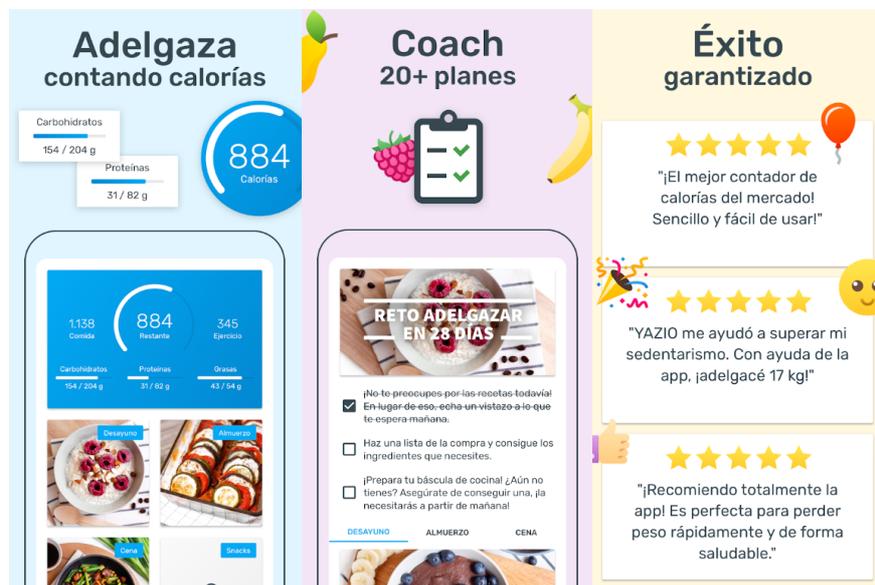


Fig. 5: Interfaz de la aplicación YAZIO

Como último análisis, otro triunfo en la Play Store, **YAZIO** se ha convertido en uno de los contadores de calorías más completos, ofreciendo un contador de calorías sencillo y muy visual, además de ofrecer al usuario recetas informando de toda su composición.

Como parte negativa, sus planes nutricionales se basan en sistemas generalizados para un gran número de usuarios, pero sin analizar ningún estilo de vida ni estilo de dieta del usuario, en el caso de querer otra receta el usuario deberá buscarla por sí mismo.

2.5 Resumen del Arte

Actualmente, los usuarios que quieren cambiar sus hábitos alimenticios recurren principalmente al uso de aplicaciones móviles para poder encontrar métodos o dietas que los puedan llevar a lograr su objetivo, que es principalmente, perder peso.

	MiNutricionista	anamatedietista	Yuka	Yazio
Suscripción	NO	SI	NO	NO
Multiplataforma	NO	SI	SI	SI
Plan nutricional	Personalizado	Personalizado	Genérico	Genérico
Predilección de alimentos	SI	NO (Solicitud)	NO	NO (Puedes eliminarlos)
Control Calórico	SI	SI	NO	SI

“En una mano el producto y en la otra el móvil que escanea de forma compulsiva” estas aplicaciones nuevas que han surgido muchas de ellas han sido planteadas por nutricionistas que tienen una gran actividad en las redes sociales, lo único premisa será dar una correcta información al consumidor

Como puntos negativos a estas aplicaciones, se pueden indicar los siguientes:

- 1) Consumidor indeciso. Dada la influencia de estas aplicaciones, un usuario puede perder el poder de decidir si un alimento es bueno o malo y sin saber el porqué.
- 2) Falsa ilusión de que cumple correctamente con su alimentación. Muchas veces un usuario busca en estas aplicaciones un motivo para poder consumir ciertos alimentos como por ejemplo un helado. Una frase de la nutricionista Raquel Bernácer que comentó en una noticia sobre ello¹: “El problema no es el propio helado, que se puede consumir esporádicamente, sino lo que se deja de comer cuando se consume un producto de forma habitual. Es decir, dejamos de comer frutas, frutos secos o vegetales”, argumenta.
- 3) Uso compulsivo de la aplicación. En muchas ocasiones, el usuario empieza a desconfiar mucho de la información nutricional que se les indica en cada producto y tiende a escanear todos los productos que compra para asegurar de que es cierto todo lo que indican.
- 4) Indecisión de la aplicación adecuada. Cada aplicación se va a regir por distintos tipos de criterios, pudiendo con ellos confundir al usuario de si cierto alimento es sano o no. **Yuka** por ejemplo, valora más si los productos son ecológicos y eso no le da un mejor valor nutricional de otro que no sea ecológico.

Después de haber analizado en detalle algunas aplicaciones, se pasará a presentar la metodología que se empleará para desarrollar esta aplicación.

¹ https://cadenaser.com/ser/2019/10/16/gastro/1571202007_085107.html

3. Fase de análisis

Para realizar este proyecto, se debe realizar un análisis que permita encontrar todos los requisitos que debe tener la aplicación. Primero se analizarán los requisitos, luego los casos de uso de cada perfil de usuario, posibles soluciones y el uso de la susodicha metodología.

3.1 Requisitos funcionales

En este punto se comentarán los requisitos funcionales de la aplicación, se contará con una descripción del comportamiento y del alcance de esta.

Para desarrollar estos requisitos se contará con la colaboración de diferentes compañeros que actualmente están realizando la carrera de Nutrición, además de otros profesionales que ya están ejerciendo esta profesión.

Uno de los requisitos más comúnmente remarcados es que les gustaría un portal privado donde poder gestionar a cada cliente y ver sus progresiones, además de poder darle cierta independencia en la gestión de sus planificaciones (basándose en sus preferencias alimentarias), y solo acudir a las sesiones concertadas con ellos.

Existe un pensamiento común entre los usuarios que “cuanto más acudas a un nutricionista más dinero ganan” se trata de un pensamiento erróneo, ya que por norma el beneficio no depende de la cantidad de citas que puedan concertar, sino de la cantidad de personas que puedan gestionar correctamente al mismo tiempo.

3.1.1 Inicio de sesión / Gestión de usuarios

- Para acceder a la aplicación se debe introducir el nombre de usuario y una contraseña, previamente creadas por el nutricionista en el proceso de registro.
- Si se produce algún error, la aplicación informará con un mensaje.
- Si todo es correcto, la aplicación deberá dejar acceder a dos tipos de perfiles de usuario, y dependiendo de su rol configurado accederá al panel de usuario o al panel del nutricionista.



3.1.2 Cerrar sesión

- Cuando el usuario decida, podrá cerrar sesión mediante el botón destinado para esa tarea.
- Este botón debe ser accesible desde cualquier pantalla de la aplicación además de cualquier dispositivo en la que se use.
- Si se ha cerrado correctamente la sesión, se redirigirá a la página de inicio.

3.1.3 Perfil del usuario

- Si un usuario tiene la sesión iniciada, tendrá un acceso a la información de su perfil, donde encontrará toda la información personal, además de cierta información relacionada con el estado corporal a partir de su peso y medidas.
- También se le debe ofrecer al usuario la posibilidad de editar la información de su perfil.
- Si el usuario actualiza su perfil, la aplicación validará todos los datos nuevos introducidos por el usuario.
- Si todo es correcto se le mostrará al usuario su información actualizada además de los nuevos datos calculados a partir de sus nuevas medidas introducidas.
- Si no ha sido correcta, se le mostrará un mensaje informando del error al usuario.

3.1.4 Configuración del perfil de usuario

- Si un usuario tiene la sesión iniciada, se le ofrecerá el acceso para poder configurar su acceso a la aplicación.
- La aplicación mostrará un formulario para que el usuario pueda modificar su contraseña de acceso a la aplicación.
- Cuando el usuario introduce los datos, la aplicación deberá validar la información introducida por el usuario.
- Si el usuario introduce correctamente los datos, se le mostrará un mensaje informando que su contraseña ha sido cambiada correctamente, manteniendo la sesión actual.
- Si sucede algún error, se le tendrá que mostrar al usuario un mensaje que no se ha podido cambiar la contraseña correctamente.

3.1.5 Plan nutricional

- Si el usuario tiene sesión iniciada, se le ofrecerá acceso al panel “*Plan nutricional*”, donde podrá acceder a planificación.
- La aplicación debe proporcionar al usuario una tabla informativa donde se le mostrará cada día toda la información nutricional para conseguir el objetivo seleccionado.
 - Se debe ofrecer al usuario la posibilidad de acceder al día seleccionado y poder visualizar toda la información de los menús.
 - También debe dar la opción al usuario de poder modificar la alimentación de ese mismo día por otras de la lista de recetas que el nutricionista tenga en su base de datos, esas recetas deben ser la seleccionadas con las preferencias del usuario. Esta funcionalidad se comentará en el siguiente apartado.
- También se ofrecerá al usuario un formulario en otra pestaña dentro de la página donde podrá gestionar una nueva planificación en base al objetivo y al tipo de alimentación seleccionada.
- También dispondrá de otra pestaña donde podrá volver a visualizar sus anteriores planificaciones.

3.1.6 Selector Nutricional

- Si el usuario tiene sesión iniciada, se le ofrecerá acceso al selector nutricional.
- La aplicación debe ofrecer al usuario un selector alimentario en el que se le ofrecerán todas las recetas de la base de datos que se adapten a su perfil de alimentación.
- Si al usuario le gusta la receta podrá seleccionarlo en el botón de “*Me gusta*”, cuando este botón es pulsado se almacenará su receta en su recetario, si todo es correcto se le mostrará la siguiente receta.
- Si el usuario no le gusta la receta se le ofrecerá un botón de “*No me gusta*”, cuando este botón es pulsado, se almacenará la receta en el recetario de recetas rechazadas, si la asignación es correcta se le mostrará la siguiente receta.

3.1.7 Panel de Administración

- Si el usuario tiene sesión iniciada y su role es Administrador (Nutricionista), se le ofrecerá acceso al panel de administración.
- La aplicación ofrecerá al usuario administrador un panel en el cual pueda observar todas las tareas que puede gestionar en la aplicación.
- Si el usuario quiere realizar una tarea únicamente deberá hacer clic en una de las opciones para enviarlo al panel de gestión de dicha tarea.

3.1.8 Panel de gestión de usuarios

- Si el usuario tiene sesión iniciada y su usuario tiene el rol de administrador, se le ofrecerá acceso al panel de gestión de Usuarios.
- La aplicación le deberá ofrecer al usuario un panel donde muestre un listado de todos los usuarios de la aplicación donde podrá gestionar las operaciones de editar el perfil, restablecer la contraseña, eliminar el usuario.
- Si todo es correcto se le notificara al usuario mediante una notificación que la operación ha sido realizada correctamente.
- Si sucede algún error se le notificara al usuario mediante una notificación que la operación no ha sido realizada correctamente.

3.1.9 Panel de registro de usuarios

- Si el usuario tiene sesión iniciada y su usuario tiene el role de administrador, se le ofrecerá acceso al panel de gestión de Usuarios.
- La aplicación le deberá ofrecer al usuario un formulario donde podrá insertar toda la información necesaria para registrar un usuario.
- Si todo es correcto se le notificara al usuario mediante una notificación que la operación ha sido realizada correctamente.
- Si sucede algún error se le notificara al usuario mediante una notificación que la operación no ha sido realizada correctamente.

3.1.10 Panel de gestión de recetas

- Si el usuario tiene sesión iniciada y su usuario tiene el role de administrador, se le ofrecerá acceso al panel de gestión de Usuarios.
- La aplicación le deberá ofrecer al usuario un panel que le muestre un listado de todas las recetas, pudiendo gestionar las operaciones de editar la receta, eliminar la receta.

- Si en la tarea que se ejecute todo es correcto, se le notificara al usuario mediante una notificación que la operación ha sido realizada correctamente.
- Si sucede algún error se le notificara al usuario mediante una notificación que la operación no ha sido realizada correctamente.

3.2 Requisitos no funcionales

3.2.1 RNF1 Acceso Multiplataforma

- 1- El usuario debe poder conectarse a la aplicación a través de un navegador desde diferentes dispositivos (móvil, tableta, ordenador...)

3.2.2 RNF2 Seguridad

- 1- Para poder acceder a la aplicación el usuario deberá estar autenticado.
- 2- La sesión deberá tener un límite de tiempo iniciada sin uso, cuando pase el tiempo configurado se cerrará automáticamente.

3.2.3 RNF3 Usabilidad

- 1- El usuario debe tener acceso a todas las funcionalidades en menos de 5 *clicks* para asegurarse que un usuario básico le sea atractiva.
- 2- La gran mayoría de peticiones *CRUD* se deberán de realizar en peticiones *Ajax* para no ralentizar la interacción con el usuario.
- 3- El diseño de la interfaz debe ser de colores planos para ser amigable con el usuario, donde el color base será blanco, dando sensación de sencillez y de iluminación, también predominar en los colores base el color azul.
- 4- El usuario podrá acceder a paginas concretas a modo de atajos gracias a las migas de pan que estarán dentro del *navbar* de la aplicación.
- 5- Cuando suceda algún error o confirmación de algún suceso dentro de la aplicación, el sistema le informará mediante notificaciones de lo sucedido.

3.2.4 RNF4 Rendimiento

- 1- El sistema debe ser lo más rápido posible en modo respuesta.

3.3 Casos de uso

Un diagrama de casos de uso representará la forma en como un actor opera en la aplicación, además de la forma, tipo, y orden en como los elementos interactúan (operaciones y casos)

En este apartado se indicará en las figuras 6 y 7 a modo de diagrama de casos de uso, todos los requisitos nombrados anteriormente, con todas las acciones que puede realizar el usuario.

3.3.1 Usuario / Cliente

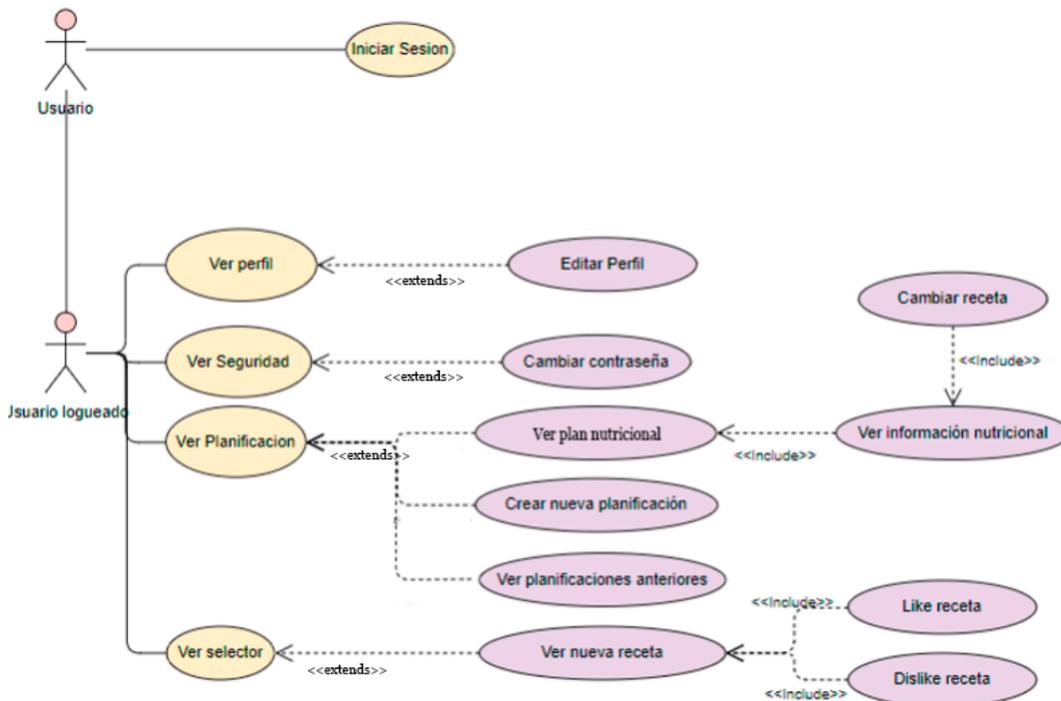


Fig. 6: Casos de Uso del usuario cliente.

3.3.2 Usuario Administrador / Nutricionista

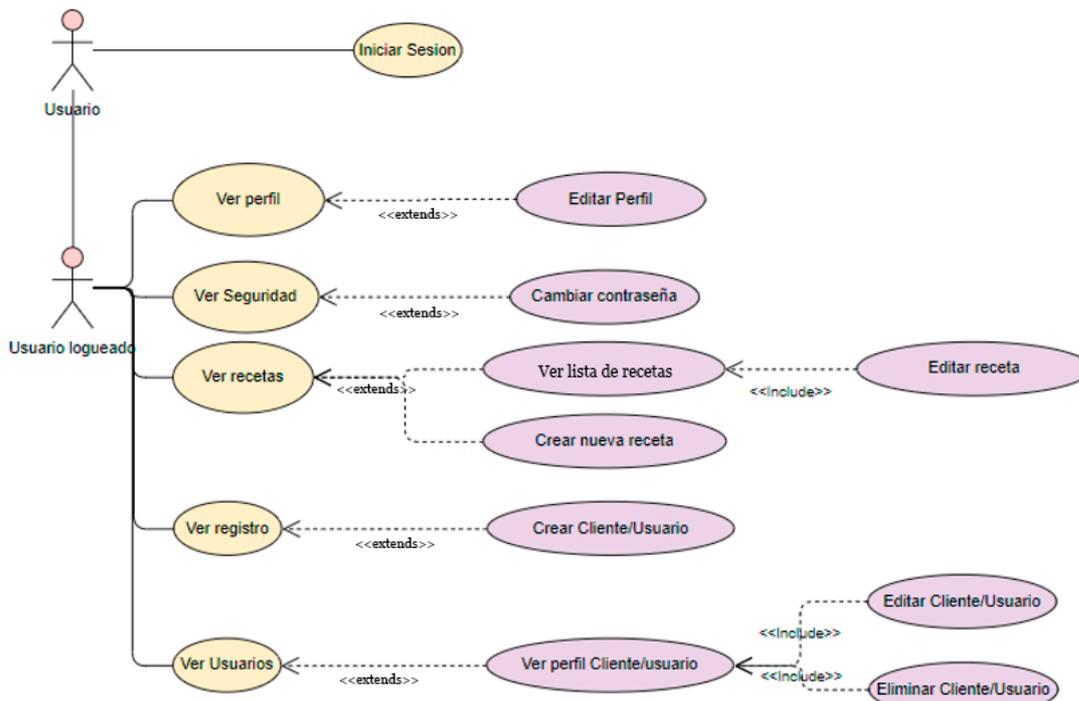


Fig. 7: Casos de Uso del usuario administrador.

3.4 Metodología.

Esta aplicación al tratarse de un desarrollo de una única persona las metodologías ágiles tienen que ser tratadas de forma más específica. Por ello se a elegido el enfoque de la estrategia de desarrollo *Top-Down* [1].

La estrategia de desarrollo *Top-Down* tiene como principales características el énfasis de tener una planificación y conocimiento del sistema completo, es decir, no se podrá empezar a realizar ningún desarrollo de la aplicación hasta que se obtenga un nivel de detalle suficiente para realizar el desarrollo.

Para este proyecto, se desarrollará en base a tareas, las cuales tendrán una duración de tiempo estimado de una jornada laboral. Estas tareas se separarán en 4 secciones:

- **Global:** Componentes necesarios para todo el proyecto.
- **Admin:** Componentes para el usuario Administrador.
- **User:** Componentes para el usuario Cliente.
- **Documentación:** Todo el desarrollo de este documento.

Aplicación Web de planificación nutricional de usuarios mediante preferencias personalizadas

ACTIVIDAD	INICIO DEL PLAN	DURACIÓN DEL PLAN	INICIO REAL	DURACIÓN REAL	PORCENTAJE COMPLETADO
Planteamiento Inicial	1	1	1	1	100%
Instalacion de los recursos para el proyecto	1	1	1	1	100%
Instalacion Base de Datos	1	1	1	1	100%
Global: Login	2	1	2	2	100%
Global: Registro	2	1	2	2	100%
Admin: Pagina Principal	3	2	3	2	100%
Admin: CRUD Recetas	3	4	3	4	100%
Admin: Listar Usuario	6	1	6	1	100%
User: Pagina Principal	7	1	7	1	100%
User: Perfil (Mostrar+Editar)	7	2	7	2	100%
User: Cambiar contraseña	7	2	8	2	100%
User: Formulario Planificador	9	2	9	2	100%
User: Crear planificacion	11	6	11	10	100%
User: Mostrar planificacion	20	5	21	6	100%
User: Ver planificaciones anteriores	27	4	27	3	100%
User: Selector Pagina principal	30	3	30	3	100%
User: Configuracion Like-Dislike	33	5	33	6	100%
User: Conexión Plan-Selector	39	4	39	4	100%
Global: Tets Funcionalidad	44	1	44	1	100%
Global: Solucion de fallos	45	2	45	5	100%
Global: Subida a Produccion	50	3	50	3	100%
Global: Comprobacion de errores.	53	2	53	2	100%
Desarrollo del Documento	7	70	7	73	100%
Desarrollo de la Presentacion	80	10	80	10	100%

Fig. 8: Tareas para el desarrollo de la aplicación.

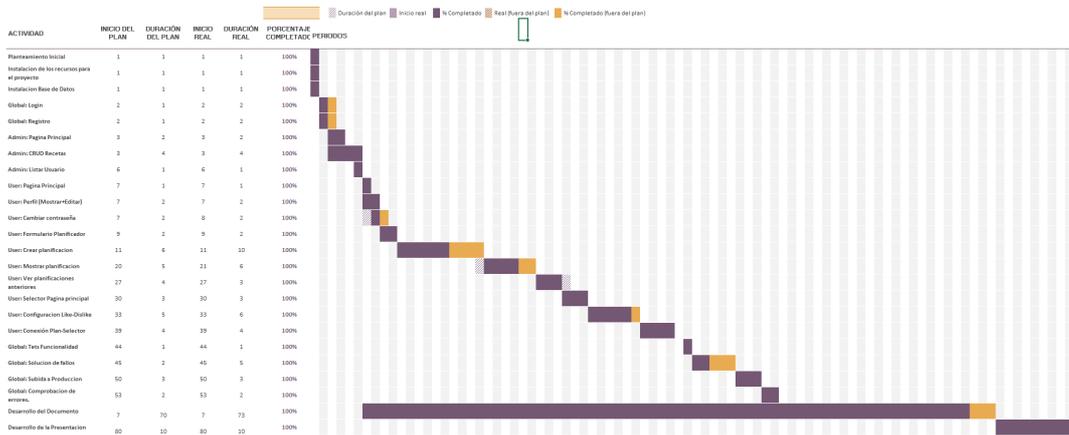


Fig. 9: Desglose de horas por cada tarea

Como se muestra en las figuras 8 y 9, hay un total de 24 tareas para el desarrollo de proyecto. Al tratarse de un desarrollo individual casi no existen tareas simultaneas cosa que con un proyecto con diferentes integrantes sí que seria posible, así como el solapado de tareas y una mejor planificación. Lo que se realiza en este caso es una planificación secuencial de las tareas a llevar a cabo.

En la figura 9 el color morado indica cual ha sido la estimación inicial de la tarea, y las zonas marcadas en amarillo corresponden con el desvío del tiempo planificado de dicha tarea. Como se muestra, hay diferencia entre el tiempo estimado y el real en 8 tareas, las cuales por ciertos inconvenientes han tenido que ser ampliadas.

Además, para el desarrollo de esta aplicación, ciertas tareas han tenido que ser realizadas en paralelo, como por ejemplo la redacción de este documento, donde a medida que se realizaban las tareas se iba completando con más información y cambios.

3.5 Identificación y análisis de posibles soluciones.

En los puntos anteriores se ha analizado los requisitos, pasando a continuación al estudio de las tecnologías que se emplearán para el desarrollo de la aplicación con sus ventajas e inconvenientes.

La primera decisión que se tomará será aplicar un desarrollo “*responsive design*”, con ello la aplicación podrá ser visualizada perfectamente en distintos dispositivos. Además, al tratarse de una aplicación web, se garantizará el acceso desde cualquier sistema operativo que tenga un navegador.

Se ha valorado la posibilidad de desarrollar una solución nativa que garantizaría, como en el caso de **MiDietario** (apartado 2.1), un funcionamiento *offline*. Pero esta solución no aportaba una clara ventaja, ya que existen diversas plataformas y cada una cuenta con tecnologías y lenguajes distintos, lo que pese al *handicap* que salvaría, el tiempo de desarrollo sería mayor al previsto.

Por otro lado, toca determinar que tecnologías se van a utilizar para el desarrollo de esta aplicación:

- Para el desarrollo de la parte *Backend* de la aplicación, se ha escogido el lenguaje de programación *PHP* dado que era una tecnología conocida y con ello poder realizar tareas con mayor complejidad.
- Para la parte *Frontend* se ha optado por utilizar la librería de estilos *Bootstrap*, para poder obtener una base de diseño que pueda adaptarse a los requisitos de la aplicación. También se han utilizado otros *frameworks* orientados al navegador como *JavaScript* y *Jquery* dando ciertas funcionalidades en la parte cliente, además de utilizar *Ajax* para realizar de peticiones asíncronas.
- Como entorno de trabajo se ha optado por *XAMPP*, software que contiene diversas herramientas para gestionar servicios en local. Para el servidor de producción se ha establecido un servidor web ubicado en un *NAS*.

3.5.1 Servidor Web

En el momento de desarrollar una aplicación web, se debe tener en cuenta donde se desarrollará y alojará.

Para el desarrollo de la aplicación es común emplear un entorno local que cuente con un servidor web, ofreciendo una solución que permite realizar distintas pruebas y las adaptaciones necesarias.

Existen distintos servidores web, uno de los más extendidos y conocidos es el servidor web *Apache*.

3.5.1.1 Servidor Apache

Un servidor Apache se trata de un servidor desarrollado en código abierto, es uno de los servidores más utilizados en la actualidad [2].

Apache es determinado como un software que trabaja dentro de un servidor web, que establece una comunicación entre un servidor y el navegador del cliente. Esta comunicación está recibiendo y enviando información indefinidamente hasta que finalice la comunicación entre ellos.

Además, cuenta con una gran capacidad de personalización gracias a su estructura basada en módulos que permite poder activar nuevas funcionalidades, como por ejemplo módulos de seguridad.

3.5.2 Alojamiento Web

Se trata de un servicio en el que un proveedor de hosting “alquila” un servidor conectado a internet donde se alojará la aplicación permitiendo el acceso de manera remota [3].

3.6 Buenas prácticas de codificación

Para un buen desarrollo, se debe tener en cuenta la ejecución de las buenas prácticas de codificación. Se puede aprender a programar buscando en internet tutoriales que permitirán al lector adquirir los conocimientos necesarios, pero lo que realmente apremia es aprender una metodología estructurada, precisa y rigurosa.

Las buenas prácticas, es un conjunto formal o informal de reglas, que se imponen con la finalidad de mejorar la calidad del software, con beneficios como:

- Ayudar en el desarrollo del programador.
- Mejora en la legibilidad y mantenimiento del código.
- Evitar errores que suelen ser muy comunes en los desarrollos, por ejemplo, equivocarse en una iteración en una posición.

En todo el desarrollo de este proyecto se van a aplicar buenas prácticas para poder garantizar un código lo más limpio posible. A continuación, se describen algunas de las buenas prácticas aplicadas en el desarrollo de la aplicación.

3.6.1 Universalizar las variables

El código debe ser lo más universal posible, para ello se debe tener en cuenta que no haya por ejemplo variables con la letra ñ e incluso evitar las tildes. Una de las normas más usadas es declarar las variables en inglés con una nomenclatura descriptiva del contenido de la variable para obtener un código con una mejor legibilidad global.

3.6.2 Comentar el código

Puede resultar una de las partes más tediosas del trabajo, pero esto facilitará que diferentes personas puedan entender el código, después de un tiempo hasta al mismo programador le puede resultar difícil entender lo que hizo en ese momento si no tiene comentado su código.

Para este proyecto en ciertas imágenes de código, se indicarán comentarios en español para facilitar al lector de este proyecto el entendimiento de cada parte del código, no obstante, el código definitivo debería estar comentado en inglés.

3.6.4 Indentaciones

Para garantizar una buena lectura del código se recomienda el uso de indentaciones, de forma que el código como se muestra en la figura 10 quede estructurado mediante tabulaciones al inicio de cada línea, acotando al mismo nivel jerárquico las líneas que están al mismo nivel.

```
<form class="comment-form" method="post">
  <div class="row">
    <div class="col-md-10">
      <div class="col-md-5">
        <label style="float:left; width: 100%;">Objetivo:</label>
        <select name="goal" style="width: 87%; float: left; margin-left: 20px; margin-top: 0px; height: 50px; font-size: 15px;">
          <option>Ganar Peso</option>
          <option>Perder Peso</option>
          <option>Mantenimiento</option>
        </select>
      </div>
      <div class="col-md-5">
        <label style="float:left; width: 100%;">Altura (cm):</label>
        <input type="number" name="height" placeholder="" style="width: 90%; float: left; margin-left: 20px;" required />
      </div>
      <div class="col-md-5">
        <label style="float:left; width: 100%;">Peso Actual(Kg):</label>
        <input type="number" name="current_weight" placeholder="" style="width: 90%; float: left; margin-left: 20px;" required />
      </div>
      <div class="col-md-5">
        <label style="float:left; width: 100%;">Peso ideal (Kg):</label>
        <input type="number" name="goal_weight" placeholder="" style="width: 90%; float: left; margin-left: 20px;" required />
      </div>
    </div>
  </div>
</form>
```

Fig. 10: Ejemplo de Indentaciones

3.6.5 Control de versiones.

Para garantizar la calidad del proyecto es conveniente el uso de un sistema de control de versiones.

Un control de versiones es un sistema que registra los cambios realizados en un archivo o conjunto de archivos a lo largo del tiempo, de modo que se pueden recuperar versiones específicas más adelante.

Para el desarrollo de esta aplicación se usará *Git* para gestionar las versiones, en el siguiente apartado se detallará mejor esta herramienta [4].

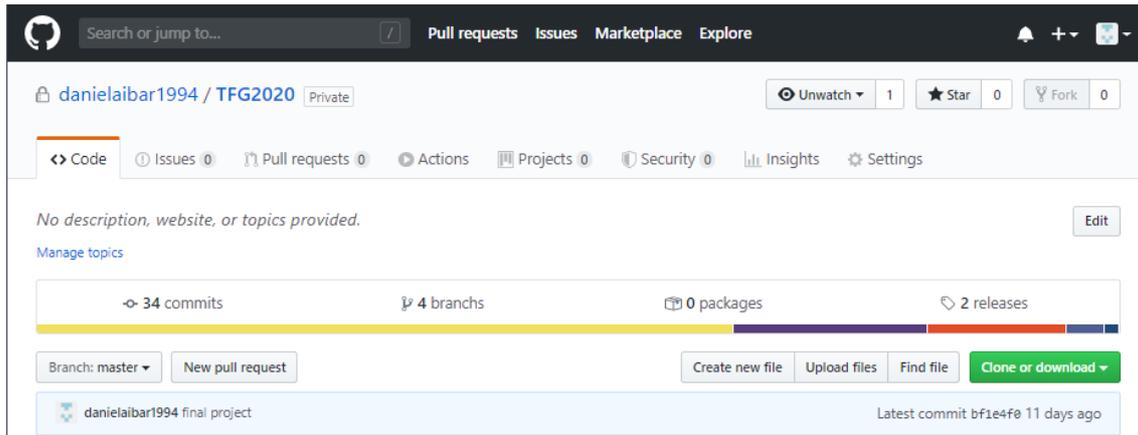


Fig. 11: Repositorio GIT donde se encuentra la aplicación.

4. Fase de diseño

En este capítulo se presentará el diseño de la aplicación.

4.1 Arquitectura

En este apartado se analizará la arquitectura de la aplicación. Ésta se basa en servicios web, es decir, estará formada por distintos servicios que se comunican entre sí, donde cada uno tiene una funcionalidad y un ámbito específico.

En la siguiente figura se puede observar la arquitectura con la que contará la aplicación, donde el cliente/usuario accederá a la misma de forma segura para gestionar la información dentro de la aplicación. Para tal fin, se empleará un navegador (o cliente web) realizando el acceso al servidor web con la información alojada en la base de datos.

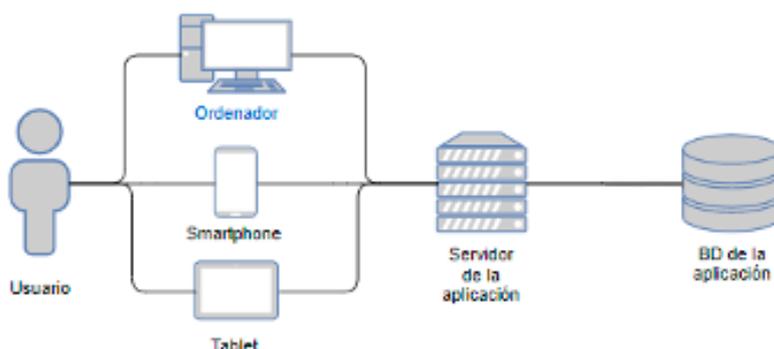


Fig. 12: Arquitectura de la aplicación.

4.2 Diseño detallado

4.2.1 Modelo conceptual

En este apartado se mostrará el diagrama de base de datos que se ha utilizado para la integración de la aplicación como muestra en la figura 13.

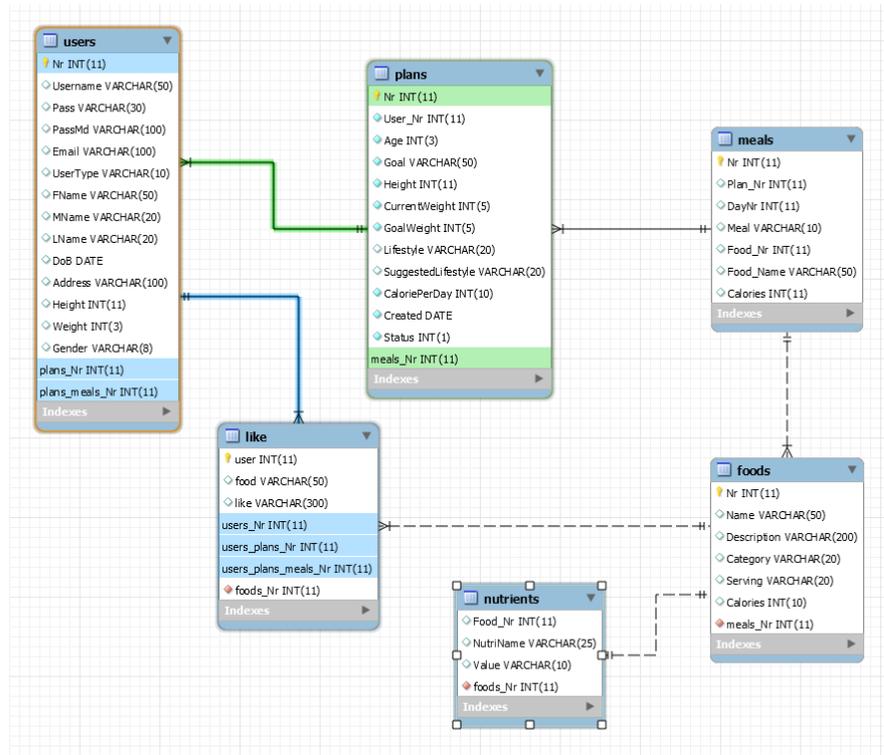


Fig. 13: Estructura de la Base de Datos

A continuación, se detallan las entidades del diagrama:

1. Entidad *Users*: aquí se almacenarán los usuarios creados en la aplicación, así como los datos necesarios.

Atributos	Descripción	Tipo
Nr	Identificador del plan	Int(11)
Username	Identificador del usuario	Varchar(20)
Pass	Contraseña	Varchar(30)
PassMd	Contraseña encriptada por MD5	Varchar(100)
Email	Correo asociado a la cuenta del usuario	Varchar(100)
UserType	Tipo de usuario	Varchar(10)
FName	Nombre del usuario	Varchar(50)
MName	Primer apellido del usuario	Varchar(50)
LName	Segundo apellido del usuario.	Varchar(50)
DoB	Fecha de registro en BBDD	Date
Address	Dirección del usuario	Varchar(100)
Height	Altura del usuario	Int (11)
Weight	Peso actual del usuario	Int (11)
Gender	Genero del usuario	Varchar(20)

2. Entidad *Plans*: aquí se almacenarán las planificaciones de los usuarios.

Atributos	Descripción	Tipo
Nr	Identificador del plan	Int(11)
User_Nr	Identificador del usuario asociado a ese plan	Int(11)
Age	Edad del Usuario en ese plan	Int(3)
Goal	Objetivo del plan	Varchar(50)
Dieta	Tipo de dieta del plan	Varchar(20)
Height	Altura del usuario en el momento en que genero el plan	Int(11)
CurrentWeight	Peso del usuario en el momento en que genero el plan	Int(5)
GoalWeight	Peso Objetivo para conseguir del usuario en el momento en que genero el plan.	Int(5)
LifeStyle	Estilo de vida del usuario en el momento que generó el plan	Varchar(20)
SuggestesLifeStyle	Estilo de vida sugerido del usuario en el momento que generó el plan	Varchar(20)
CaloriesPerDay	Calorías Diarias del usuario en el momento que generó el plan	Int(10)
Created	Fecha de creación del plan	Date
Status	Estado del plan, (0 No activo, 1 Activo)	Int(1)

3. Entidad *Meals*: esta tabla almacenará los menús diarios que tendrá cada usuario

Atributo	Descripción	Tipo
Nr	Identificador de la comida	Int(11)
Plan_nr	Identificador del plan asociado	Int(11)
DayNr	Identificador del día (posición)	Int(11)
Meal	Identificador de la posición asociada en la comida (Desayuno (1), comida (2) o cena (3))	Int(11)
Food_Nr	Identificador de la receta asociada	Int(11)
Food_Name	Nombre de la receta asociada	Varchar(50)
Calories	Calorías de la receta asociada	Varchar(20)

4. Entidad *Foods*: En esta tabla se almacenarán todas las recetas que el nutricionista/administrador de la aplicación vaya creando.

Atributo	Descripción	Tipo
Nr	Identificador de la receta	Int(11)
Name	Nombre de la receta	Varchar(50)
Description	Breve descripción de la receta	Varchar(200)
Category	Categoría a la que pertenece esta receta.	Varchar(20)
Serving	Tipo de ración de la receta	Varchar(20)
Calories	Valor calórico de la receta	Int(10)
photo	Nombre de la imagen correspondiente a la receta.	Varchar(255)

Normal	Indica si la receta se adapta a este tipo de dieta con un 1 o un 0 si no se adapta	Int(1)
Vegan	Indica si la receta se adapta a este tipo de dieta con un 1 o un 0 si no se adapta	Int(1)
Vegetarian	Indica si la receta se adapta a este tipo de dieta con un 1 o un 0 si no se adapta	Int(1)
Intolerant	Indica si la receta se adapta a este tipo de dieta con un 1 o un 0 si no se adapta	Int(1)
Paleo	Indica si la receta se adapta a este tipo de dieta con un 1 o un 0 si no se adapta	Int(1)

5. Entidad *Nutrients*: esta tabla contiene la información nutricional que tendrá cada una de las recetas creadas en la entidad *Food*

Atributos	Descripción	Tipo
Food_Nr	Identificador de la receta asociada	Int(11)
NutrientName	Nombre del nutriente	Varchar(50)
Value	Valor del nutriente	Varchar(20)

6. Entidad *Likes*: esta tabla almacenará las opiniones que tienen los usuarios sobre una receta, indicando con un 0 o 1 si es de su agrado o no le gusta.

Atributo	Descripción	Tipo
user	Identificador del usuario	Int(11)
meal	Identificador de la receta	Int(11)
like	Valor 0 si no le gusta, valor 1 si le gusta la receta.	Int(11)

4.2.3 Diseño de la interfaz de usuario

En la fase de análisis se han comentado los diferentes requisitos que debe cumplir la aplicación para que sea accesible desde cualquier plataforma, además de aportar una interfaz sencilla y fácil de usar.

El concepto de *wireframe* se trata de un tipo boceto que puede representar visualmente un esquema sencillo de interfaz de aplicación web [5].

Los *wireframe*, son utilizados antes de realizar el diseño web, en donde antes de realizar estos bocetos, se deberán revisar por diferentes personas especializadas en diseño y usabilidad web.

Una de las características de los *wireframe* es que carecen de estilo, imágenes, colores, ni ningún tipo de tipografía, dado que principalmente están diseñados para plasmar la funcionalidad y contenidos de tendrá la aplicación.

A continuación, se mostrarán las diferentes ventanas que ofrecerá la aplicación y una base de su diseño.

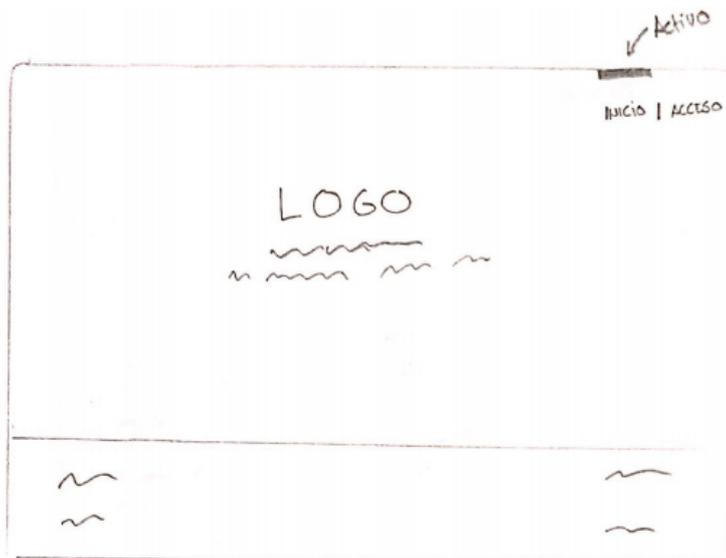


Fig. 14: Wireframe de la pantalla de inicio.

En la figura 14 se le muestra al usuario una Página Principal que tiene como función presentar una breve información de la funcionalidad de la aplicación, además de incorporar imágenes corporativas, donde contará con un diseño simple al lado de dos botones en la parte superior, Inicio y Acceso, al presionar en Acceso, le redirigirá a la siguiente ventana, Inicio de Sesión tal y como se muestra en la figura 15.

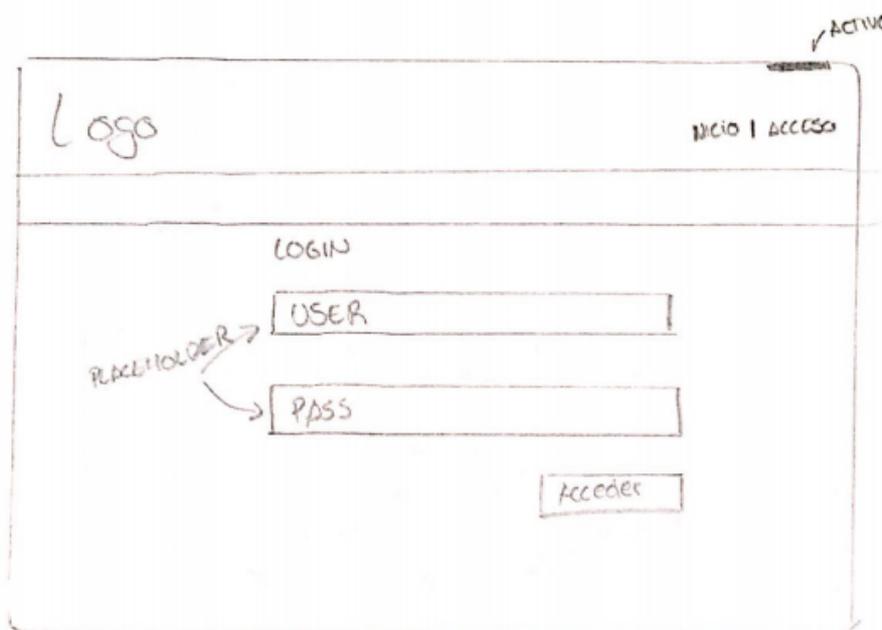


Fig. 15: Wireframe de la pantalla de Inicio de Sesión

La interfaz de Inicio de sesión es un diseño bastante simple, donde se encuentra un formulario donde se deberá introducir el usuario y la contraseña.

Una vez se haya iniciado sesión, aparecerá la página principal de usuario, en la cual se mostrará al usuario su perfil, entre los que destaca su información personal, sus medidas, y su situación de peso actual tal y como muestra la figura 16.

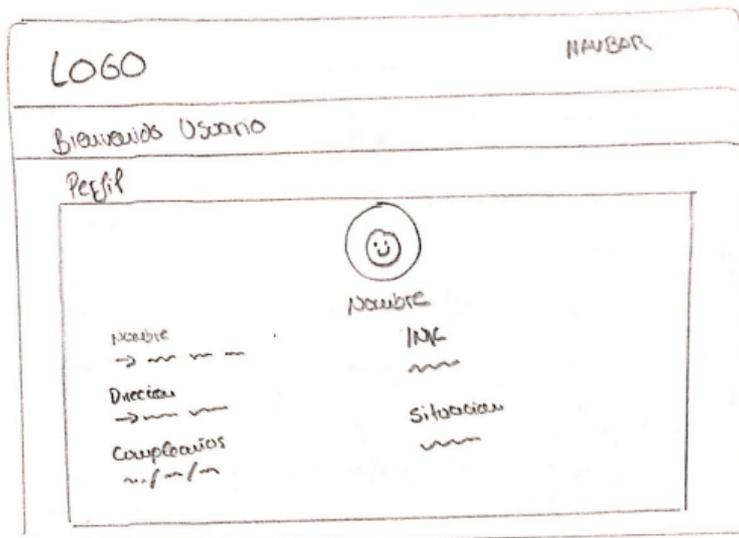


Fig. 16: Wireframe del panel de usuario

Esta misma ventana ofrece también el acceso directo para actualizar la información personal del usuario, donde aparecerá un formulario tal y como muestra la figura 17, donde cada input contendrá la información anteriormente guardada y podrá editarla.

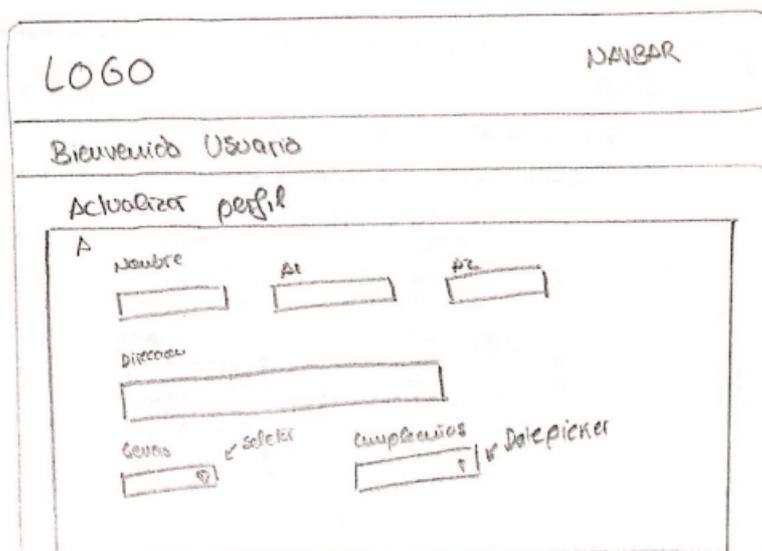


Fig. 17.: Wireframe del panel de actualizar perfil

Si el usuario entra al apartado de crear una dieta, le aparecerá un panel como se muestra en la figura 18 donde rellenar todos los parámetros para así generar la dieta personalizada.

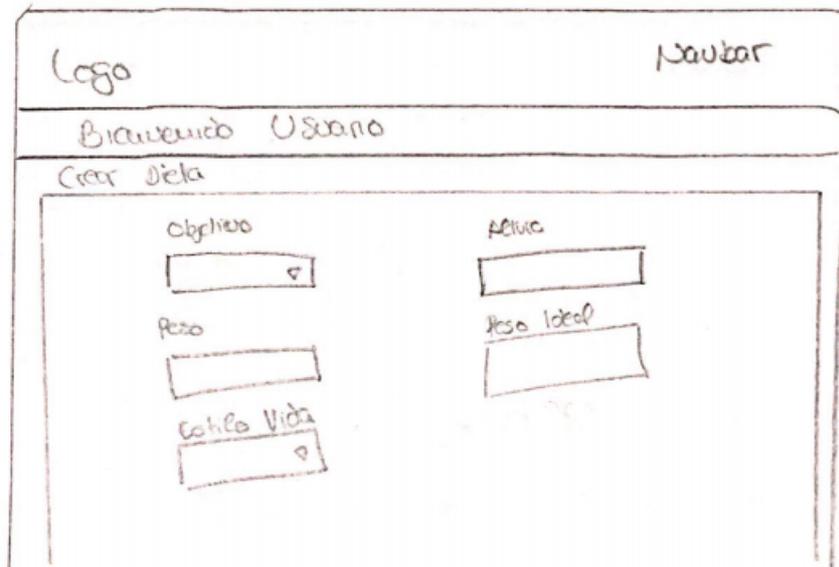


Fig. 18: Wireframe del panel de creación de la Dieta.

Una vez insertada la información, la ventana cambiará y aparecerá un panel con una tabla mensual con todas las recetas generadas por la aplicación, estando separadas en 3 apartados, Almuerzo, Comida y Cena, y en ellas se le indicará cada una de la que debe consumir tal y como se muestra en la figura 19.

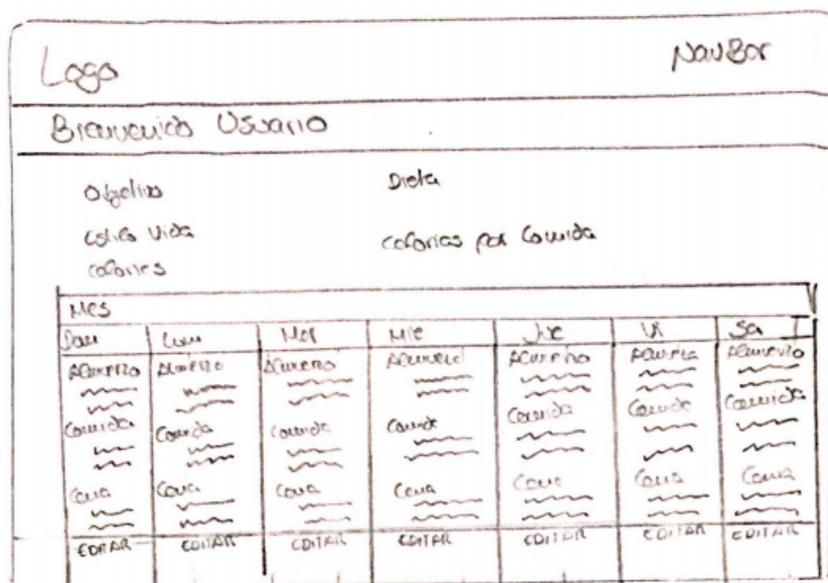


Fig. 19: Wireframe del panel de Consulta de la Planificación.

Si el usuario quiere configurar sus preferencias alimentarias, podrá acceder al apartado selector, donde como en la figura 20 a modo de “Tinder” se le mostrará una serie de recetas donde podrá elegir si son de su agrado o no de manera sencilla.

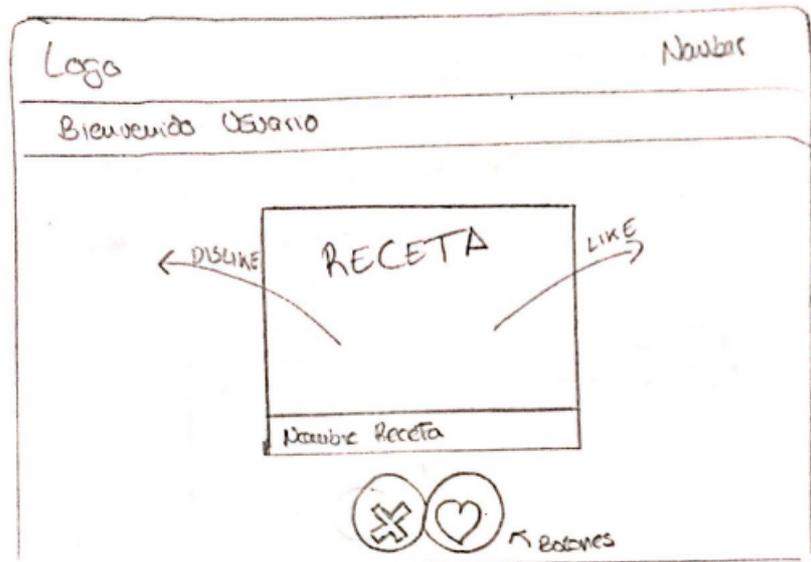


Fig. 20: Wireframe del panel de Selector.

La siguiente zona esta solamente reservada para un usuario que sea administrador, esta interfaz le ofrecerá a ese usuario ciertas configuraciones de cara a la aplicación.

Lo primero que le aparecerá al usuario cuando acceda será su perfil e información tal y como se muestra en la figura 21.

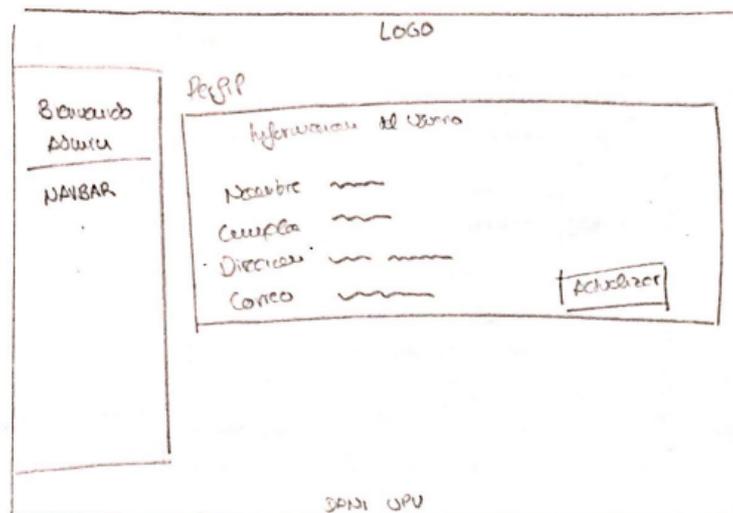


Fig. 21: Wireframe del panel de inicio de Administrador.

Si el usuario pulsa sobre Editar perfil, le aparecerá un panel en donde podrá editar todos los parámetros de su cuenta como en la figura 22.

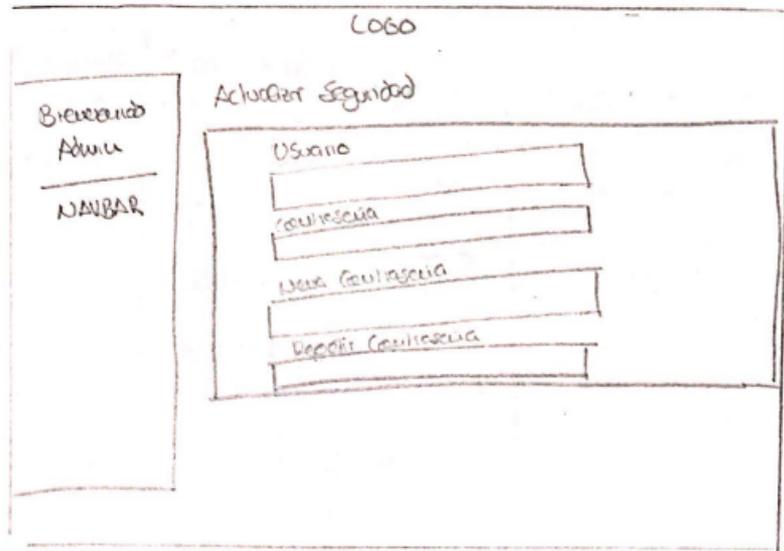


Fig. 22: Wireframe del panel de edición de perfil del Administrador.

El usuario administrador, podrá acceder al buscador de recetas tal y como se muestra en la figura 23, donde le aparecerán todas las recetas creadas a modo de tabla, pudiendo también realizar las operaciones *CRUD*.

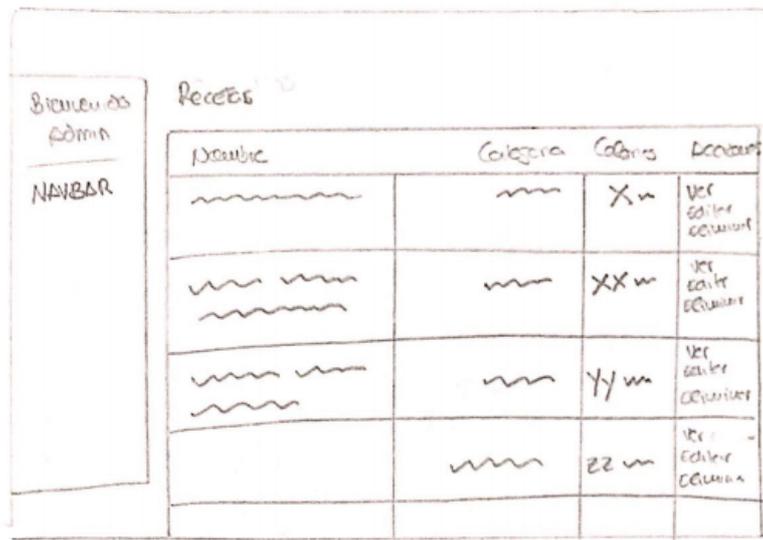


Fig. 23: Wireframe del panel de Recetas

Si el usuario presiona en crear una nueva receta, le aparecerá un panel como en la figura 24 donde podrá insertar toda la información necesaria para crear una receta además de tener un subpanel donde puede indicar los nutrientes que tiene la receta, ya sea la cantidad de proteínas, hidratos, grasas, hierro, entre otros.

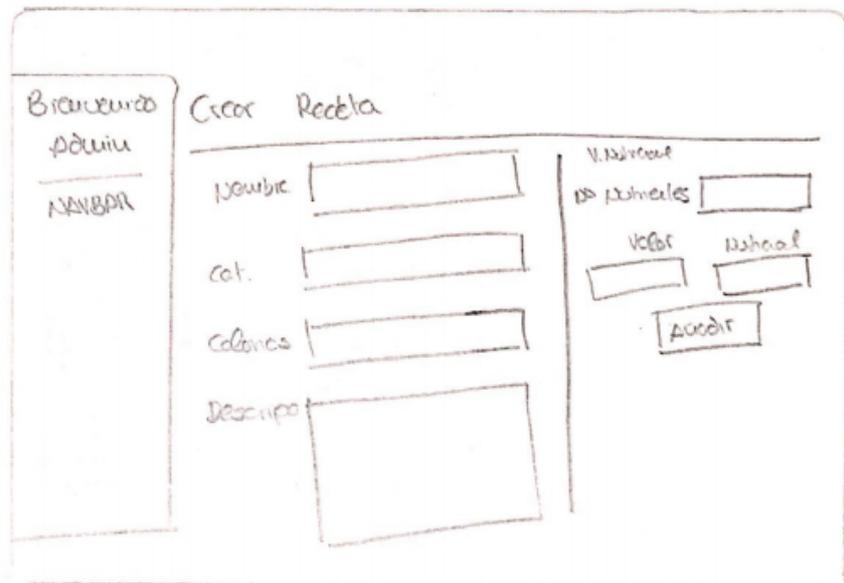


Fig. 24: Wireframe del panel de creación de Recetas.

5. Fase de Implementación

En este apartado se definirá cómo se ha implementado la aplicación. Para ello, se detallarán las tecnologías utilizadas, se mostrará la configuración del entorno de desarrollo y por último se presentarán algunas partes de código de la aplicación.

5.1 Tecnologías

En este apartado se van a resumir las tecnologías utilizadas para el desarrollo de la aplicación.

5.1.2.1 *Online.visual-paradigm*

Se trata de una potente herramienta web que permite crear una gran variedad de diagramas técnicos con el uso de plantillas ya diseñadas. Durante todo el proyecto se utilizarán para escenificar la arquitectura de la aplicación. [6]

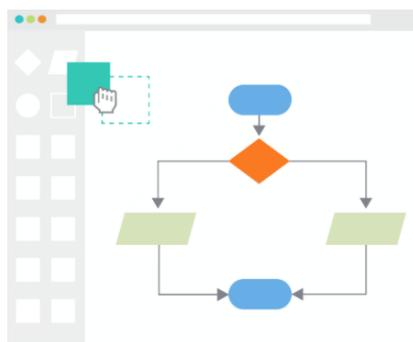


Fig. 25: Imagen conceptual de la funcionalidad de *online.visual-paradigm*.

5.1.2.2 *PHP*

PHP según su web, es el acrónimo recursivo de *Hypertext Proprocessor*, se trata de un lenguaje de código abierto muy popular, adecuado para el desarrollo web en donde puede incrustarse junto al *HTML* [7].

El código es interpretado por un servidor web con módulo de procesamiento de *PHP* que genera un texto plano en formato *UTF-8*, que es reconocida por el formato *HTML*, pudiendo dar una salida entendible en los navegadores.



Fig. 26: Logotipo de *PHP*

Como principales ventajas del uso del lenguaje PHP son:

- Lenguaje abierto.
- Es fácil de aprender.
- Sus entornos son de fácil configuración.
- Su facilidad para poder integrarse con las bases de datos
- Tiene una comunidad muy extensa.

5.1.2.3 Java Script

Es uno de los lenguajes de programación cliente más utilizado en el mundo, JavaScript permite la realización de actividades complejas dentro de una pagina web. [8]

Tiene un núcleo de características comunes de programación que nos permiten realizar tareas como:

- Almacenar valores en variables.
- Operaciones escritas en formato texto
- Funcionamiento de respuestas de eventos.



Fig. 27: Logotipo de JavaScript.

5.1.2.4 JQuery

Se trata de una librería basada en *Javascript*, la cual simplifica ciertas tareas a la hora de programar en JavaScript permitiendo además agregar interacciones en un sitio web [9] [10].

Para trabajar con *JQuery* únicamente tendremos de incluir el fichero de *JQuery* en el proyecto y ya podremos trabajar con él.



Fig. 28: Logotipo de jQuery

5.1.2.5 MySQL

Se trata de un gestor de Bases de Datos de los más utilizados y reconocidos en la actualidad para el desarrollo web [11].

Es altamente empleado en conjunto con servidores web relacionadas con aplicaciones web o *CMS* de sitios online. Esta especialmente ligado con desarrollos en *PHP*.

MySQL está basado en *SQL* siendo un sistema de bases de datos relacional, es decir, archiva datos en tablas separadas en lugar de guardar todo en un gran archivo, permitiéndole obtener una mayor velocidad y flexibilidad. Las tablas están relacionadas de formas definidas, por lo que se posibilita la combinación de destinitos datos en varias tablas para su conexión.



Fig. 29: Logotipo de MySQL

5.1.2.6 Git

Es un software para la gestión de versiones pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente [12].

Su función principal es llevar un registro de los cambios en los archivos y coordinar el trabajo que puedan desarrollar una o varias personas sobre archivos compartidos.



Fig. 30: Logotipo de Git.

5.1.2.7 Git Extensions

Con este programa se puede examinar el historial completo de todos los *commits* del repositorio *git* con una interfaz gráfica [13].

Esto agilizará la tarea de interacción con las diferentes versiones de la aplicación. Para garantizar una mayor calidad en el proyecto, cada tarea se separará en una rama diferente para poder desarrollar sobre ella y no dañar así nada del código que funcione en ese momento.

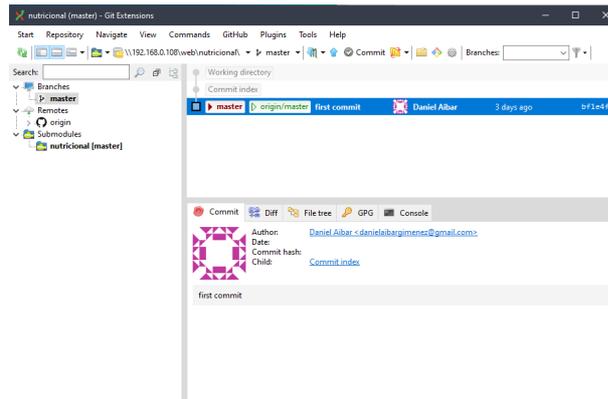


Fig. 31: Panel principal de Git Extensions

5.2 Configuración del entorno

Para empezar, se tendrá que realizar una organización de la aplicación seguido de la instalación del entorno.

5.2.1 Organización de los ficheros

Para la organización de los ficheros como se muestra en la figura 32, se ha establecido una estructura lo más básica y simplificada posible separando cada apartado de la aplicación:

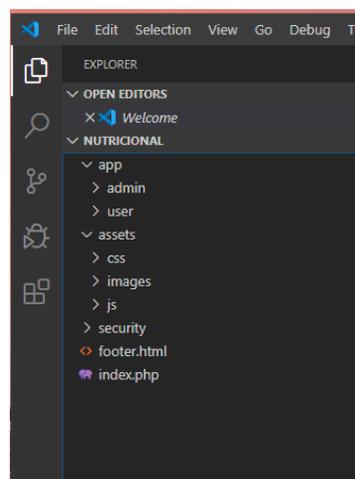


Fig. 32: Organización de lo ficheros.

En primer lugar, existirá un fichero *index.php* que será la página de inicio.

La carpeta *app* como se muestra en la figura 33 será donde se alojarán los diferentes tipos de usuario de la aplicación, en este caso un usuario normal (*user*) y un usuario administrador / nutricionista (*admin*).

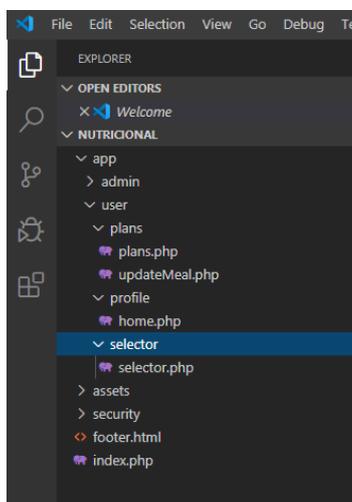


Fig. 33: Ficheros dentro del directorio *app*

La carpeta *assets*, almacenará los recursos compartidos de estilos, funciones en *JavaScript* e imágenes.

La carpeta *security* como se muestra en la figura 34 se encargará de todas las operaciones compartidas dentro de la aplicación.

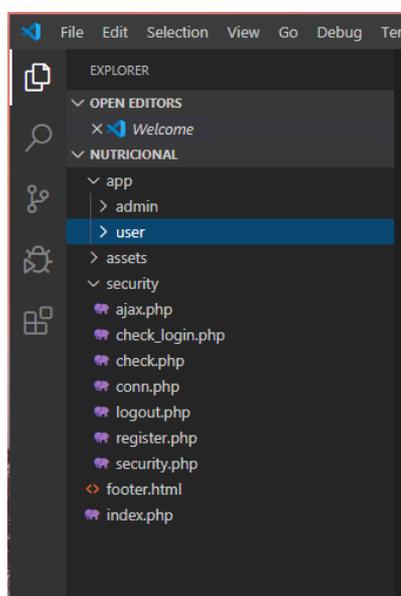


Fig. 34: Ficheros del directorio *security*

5.2.3 XAMPP - Servidor web local.

Para la instalación de XAMPP bastará con buscarla en Google como se muestra en la figura 35 y acceder a su página principal.

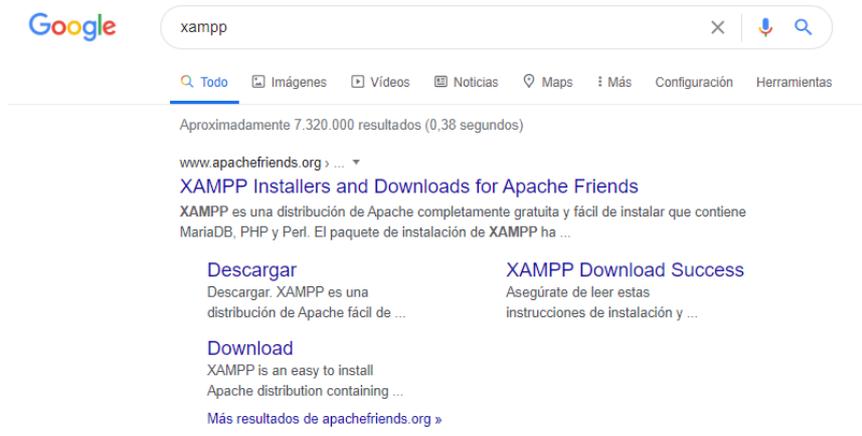


Fig. 35: Resultado de buscar XAMPP en Google

Una vez descargado el ejecutable e instalado en el equipo de desarrollo ya estará instalado un servidor apache y la base de datos, estando preconfiguradas en ciertos puertos. Como únicamente existirá un proyecto, no será necesario modificar estos parámetros.

Una vez levantado XAMPP se procederá a levantar el modulo Apache y el módulo MySQL como se muestra en la figura 36.

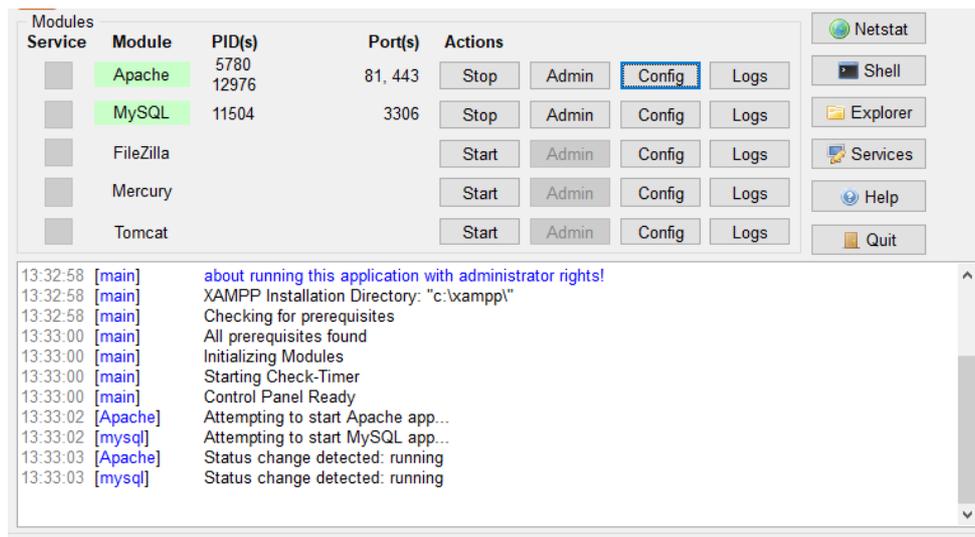


Fig. 36: XAMPP en funcionamiento.

5.2.2 Configuración MySQL.

Se procederá a la creación de la base de datos para la aplicación.

Para ello en la fila del módulo de *MySQL* se presionará en el botón *Admin*, donde se accederá a una ventada del navegador mostrando el acceso con *PHPMYAdmin* que hará de aplicación de gestión de la BD.

Una vez dentro se debe crear una nueva base de datos, para ello, se presionará sobre el botón “Nueva” como se muestra en la figura 37.



Fig. 37: Creación de una nueva tabla de Base de Datos

Aparecerá una nueva ventana como en la figura 38, donde solicitará introducir el nombre de la nueva base de datos:



Fig. 38: Asignación del nombre de la Base de Datos.

Posteriormente a la creación de la base de datos, se deberán crear todas las tablas que compongan esta base de datos mencionadas en el apartado *4.2.1 Modelo conceptual*.

5.3 Ficheros de funcionalidad de la aplicación

5.3.1 Fichero de conexión conn.php

Uno de los ficheros fundamentales será el fichero que establezca la comunicación entre la aplicación y la base de datos. Para ello se deberá configurar la conexión con las credenciales de la base de datos para una perfecta conexión como se muestra en la figura 39.

```
security > conn.php > ...
1  <?php
2  session_start();
3  localtime();
4  $hostname="localhost";
5  $user="daniAibar";
6  $password="TFG2020";
7  $database="nutcal2";
8
9  if (!$con_db=mysqli_connect($hostname,$user,$password)){
10 die("Can't connect to database!");
11 }
12 else{
13     if(! mysqli_select_db("$database",$con_db)){
14 die("Can't connect to database!");
15     }
16 }
17 }
```

Fig. 39: Fichero de conexión de la Base de Datos

Una de las recomendaciones que se indican para la base de datos es deshabilitar el usuario *root* de *MySQL* y crear otros usuarios para acceder únicamente a las tablas que se quieran utilizar.

5.3.2 Fichero de inicio de sesión login.php

Para que el usuario pueda acceder a la aplicación contará con un formulario como el de la figura 40 donde indicará su usuario y contraseña.



Fig. 40: Interfaz de Inicio de sesión.

Cuando se presione el botón Acceder, se validarán los valores introducidos en los inputs.

Para realizar este proceso, se ejecutarán las instrucciones mostradas en la figura 41 de la siguiente forma:

1. Se comprobará que existe la variable `$_POST['submit']`. Esta es la variable encargada de saber si se ha hecho un *submit* en el formulario. Si se ha realizado esta operación se procederá a los siguientes pasos.
2. Se recogerá los datos que tenemos en el formulario, como son el usuario y la contraseña.
3. Se procederá a realizar la consulta *SQL* de comprobación que el usuario existe en la base de datos.

```
1 <?php
2 include "conn.php";
3 include "check.php";
4
5 $message = "";
6 $msg = 0;
7 if (isset($_POST['submit'])) {
8     $uname = $_POST['user'];
9     $pword = md5($_POST['pass']);
10
11     $q = mysqli_query(
12         $conn,
13         "SELECT UserType FROM users WHERE Username = '$uname' AND PassMd = '$pword'"
14     ) or die(mysqli_connect_errno());
15     $num = mysqli_num_rows($q);
16
17     $msg = 1;
18     if ($num == 1) {
19         $rec = mysqli_fetch_array($q);
20         $type = $rec['UserType'];
21
22         $_SESSION['user'] = $uname;
23         $_SESSION['pass'] = $pword;
24         $_SESSION['type'] = $type;
25
26         if ($type == "admin") {
27             header("Location: admin/index.php");
28         } else {
29             header("Location: user/home.php");
30         }
31     } else {
32         session_destroy();
33         $msg = 1;
34         $message = "Credenciales no validas.";
35     }
```

Fig. 41: Código ejecutado para el Inicio de Sesión.

4. Si el resultado de la *SQL* es de 1 resultado se agregarán a la sesión creada las credenciales del usuario. También se comprobará de que tipo es el usuario, si es un usuario corriente o es un usuario administrador, dado que tendrán diferentes zonas de acceso.
5. Si la variable `'num'` no es 1 se procederá a mostrar al usuario un mensaje de credenciales no válidas.

5.3.3 Capa usuario

5.3.3.1 Fichero plans.php

En este fichero, se mostrará toda la planificación mensual de manera que de un simple vistazo pueda ver que recetas le van a tocar hoy.

Si se requiere crear un nuevo plan nutricional, el usuario debe acceder al apartado de Crear Dieta, donde le aparecerá un formulario como se muestra en la figura 42 que debe rellenar con las características que desea.

Crear Dieta

Objetivo:

Altura (cm):

Peso Actual(Kg):

Peso ideal (Kg):

Estilo de vida:

Información:

Sedentario (Realiza ejercicio 1 o menos veces/semana)

Ligeramente Activo (Ejercicio ligero 1-3 dias/semana)

Moderadamente Activo (Ejercicio Moderado 3-5 dias/semana)

Activo (Practica deporte 6-7 dias/semana)

Muy Activo (Practica mucho deporte mas de 1 vez por dias - +7dias/semana)

Fig. 42.: Interfaz de creación de un Plan Nutricional.

Para realizar este proceso, se ejecutarán las instrucciones mostradas en las figuras 43, 44, 45, 46 y 47 de la siguiente forma:

- 1- Si se accede en crear una nueva planificación, lo primero que se debe hacer es deshabilitar el plan que tenía actualmente el usuario
- 2- Se almacenará en variables todos los datos que se consideren necesarias para el buen funcionamiento de la aplicación.
- 3- Como medidas de cálculo se utilizarán la Tasa de Metabolismo Basal (TMB en adelante), donde primero se analizará el sexo del usuario, y se calculará el TMB. Los casos serán dos:
 - a. Hombre: $TMB = (10 * \$Peso_actual) + (6.25 * \$altura) - (5 * \$edad) + 5$
 - b. Mujer: $TMB = (10 * \$Peso_actual) + (6.25 * \$altura) - (5 * \$edad) - 161$
- 4- Sobre el cálculo que se han obtenido en el anterior punto se va a analizar el estilo de vida que ha seleccionado el usuario, a partir de esa selección dependerá la variable con la cual se multiplique:
 - a. Sedentario: $\$tmb * 1,2$
 - b. Ligeramente Activo: $\$tmb * 1,375$
 - c. Moderadamente Activo: $\$tmb * 1,55$
 - d. Activo: $\$tmb * 1,725$
 - e. Por defecto*: $\$tmb * 1,9$

```

if (isset($_POST['sign'])) {
    // EL PLAN ACTUAL SE DESHABILITA
    mysql_query($conn, $conn, "UPDATE plans SET Status = 0 WHERE User_Nr = '$id' AND Status = 1");

    //proposito
    $goal = $_POST['goal'];

    //peso actual
    $current_weight = $_POST['current_weight'];
    //peso al que quiero llegar
    $goal_weight = $_POST['goal_weight'];
    //estilo de vida
    $lifestyle = $_POST['lifestyle'];
    //peso
    $height = $_POST['height'];
    //fecha
    $date = date("Y-m-d");

    $dieta = $_POST['dieta'];

    $birthdate = new DateTime($dob);
    $today = new DateTime("today");
    $age = $birthdate->diff($today->y);

    if ($gender == "Hombre") {
        $bmr = (10 * $current_weight) + (6.25 * $height) - (5 * $age) + 5;
    } else {
        $bmr = (10 * $current_weight) + (6.25 * $height) - (5 * $age) - 161;
    }

    switch ($lifestyle) {
        case 'Sedentario':
            $bmr *= 1.2;
            break;
        case 'Ligeramente Activo':
            $bmr *= 1.375;
            break;
        case 'Moderadamente Activo':
            $bmr *= 1.55;
            break;
        case 'Activo':
            $bmr *= 1.725;
            break;
        default:
            $bmr *= 1.9;
            break;
    }
}

```

Fig. 43: Código del funcionamiento de creación de un Plan Nutricional.

- 5- Después de obtener los datos anteriores, se debe insertar una nueva condición, que será el objetivo del usuario en esta nueva planificación, dependiendo de esta, se incrementará o disminuirá la cantidad de la variable \$TMB:
 - a. Ganar peso: \$TMB + 500 (calorías)
 - b. Perder peso: \$TMB - 500 (calorías)
 - c. Por defecto (Mantenimiento): \$TMB

- 6- El cálculo realizado en el punto anterior se almacenará en la variable \$CaloriasPorDia, donde se analizará si es inferior a los mínimos calóricos establecidos para una persona, si fuera el caso, se determinaría el valor de esta variable en el mínimo.

- 7- Se establecerá una media de calorías por comida. En este caso la aplicación está diseñada para tres comidas al día, por ello se distribuirán las calorías uniformemente entre ellas, este calculo se almacenará en la variable \$CaloriasPorComida

```
// si queremos ganar peso se incrementaran 500calorias sino -500 o sino se mantienen las calorías
$updteLifestyle = 0;
switch ($goal) {
  case 'Ganar Peso':
    $calorieDay = $bmr + 500;
    if ($gender == "Hombre") {
      if ($calorieDay < 1750) {
        $calorieDay = 1800;
        $updteLifestyle = 1;
      }
    } else {
      if ($calorieDay < 1150) {
        $calorieDay = 1200;
        $updteLifestyle = 1;
      }
    }
    $calorieMeal = $calorieDay / 3;
    break;
  case 'Perder Peso':
    $calorieDay = $bmr - 500;
    if ($gender == "Hombre") {
      if ($calorieDay < 1750) {
        $calorieDay = 1800;
        $updteLifestyle = 1;
      }
    } else {
      if ($calorieDay < 1150) {
        $calorieDay = 1200;
        $updteLifestyle = 1;
      }
    }
    // aqui es el calculo de 3 comidas por día
    $calorieMeal = $calorieDay / 3;
    break;
  default:
    $calorieDay = $bmr;
    $calorieMeal = $calorieDay / 3;
    break;
}
```

Fig. 44: Código del funcionamiento de creación de un Plan Nutricional 2

- 8- Tras establecer los datos para calcular las calorías se pasará a guardar la información. Por ello se establecerá una conexión con la base de datos y se procederá a realizar un UPDATE de los datos personales del usuario, y finalmente insertar la planificación en la tabla de 'plans' con todos los datos obtenidos en los puntos anteriores.
- 9- Se tendrá que recoger el plan que se ha insertado para averiguar su identificador, por ello se realizará una SELECT para averiguarlo.
- 10- Se inicializarán varios *arrays* que almacenarán la información necesaria para poder establecer el plan. En este caso son 3, la de comida, la del nombre de la comida, y el *array* de calorías de cada comida, con ellos cada posición de cada *array* proporcionaran los datos de una receta en cuestión.
- 11- Por defecto al insertar una receta/alimento, la aplicación incluirá el Agua como un artículo obligatorio en cada comida, para establecer unos hábitos de hidratación correctos, por ello, se buscará con una SELECT la información de "Agua" y se incluirá en variables.
- 12- Se consultarán todas las recetas almacenadas en la aplicación, con la excepción de la receta Agua, que ya está almacenada en el punto anterior.
- 13- Luego de realizar la consulta, se incluirán todos los alimentos en los *arrays* del punto 10. Almacenando en una variable el número de comidas que se han insertado en el *array*, además de un nuevo *array* que informará de las comidas por semana.

- 14- Se deberá realizar el cálculo para llegar al peso objetivo de la mejor manera posible, la cantidad establecida máxima recomendable es de 0,5Kg por semana. Con esta cantidad establecida deberemos de ver el peso actual y el peso objetivo, restarlo y la diferencia multiplicarla x2, para así saber el número de semanas.
- 15- Una vez averiguado en número de semanas, se pasará a saber cuántos días son esas semanas, se almacenará en una variable el número de semanas x7, con ello se sabrá cuantos días de planificación se deben generar más adelante.

```

mysql_query($conn, "UPDATE users SET Height = '$height', Weight = '$current_weight'");
mysql_query($conn,
"INSERT INTO plans (User_Nr, Age, Goal, Height, CurrentWeight, GoalWeight, Lifestyle, SuggestedLifestyle, CaloriesPerDay, Created, Status, Dieta)
VALUES ('$id', '$age', '$goal', '$height', '$current_weight', '$goal_weight', '$lifestyle', '$sgstedlifestyle', '$bmr', '$date', 1, '$dieta')");

$q_plan = mysql_query($conn, "SELECT Nr FROM plans WHERE User_Nr = '$id' AND Status = '1'" or die(mysql_connect_errno());
$f_plan = mysql_fetch_array($q_plan);

$nr_plan = $f_plan['Nr'];

$queueFoods = array();
$queueFoodName = array();
$queueCalories = array();

$q_rice = mysql_query($conn, "SELECT Nr, Calories FROM foods WHERE Name = 'Agua'" or die(mysql_connect_errno());
$q_r = mysql_fetch_array($q_rice);
$nr_rice = $q_r['Nr'];
$cal_rice = $q_r['Calories'];

$q_foods = mysql_query($conn, "SELECT Nr, Name, Calories FROM foods WHERE Name != 'Agua' ORDER BY Calories ASC" or die(mysql_connect_errno());

while ($f = mysql_fetch_array($q_foods)) {
    $queueFoods[] = $f['Nr'];
    $queueFoodName[] = $f['Name'];
    $queueCalories[] = $f['Calories'];
}

$nr_q_foods = count($queueFoods);

$queueFoodsPerWeek = array();

// calculo de la mejor manera en semanas para perder el peso adecuadamente
switch ($goal) {
    case 'Ganar Peso':
        $nrWeeks = ($goal_weight - $current_weight) * 2;
        break;
    case 'Perder Peso':
        $nrWeeks = ($current_weight - $goal_weight) * 2;
        break;
    default:
        $nrWeeks = 3;
        break;
}

$nrDays = $nrWeeks * 7;

```

Fig. 45: Código del funcionamiento de creación de un Plan Nutricional 3.

- 16- A partir de este punto, se tendrá que recorrer todos los días contabilizados por la variable anterior y asignar a cada día las comidas que también estarán organizadas en 3 posiciones numéricas:
 - a. 1: Almuerzo
 - b. 2: Comida
 - c. 3: Cena
- 17- Por defecto la primera receta que le vamos a insertar será la del 'Agua', que es obligatoria en el código. Para ello se realizará el INSERT en la base de datos en la tabla *meals* con la información del plan, además de los datos de la receta Agua.
- 18- Se seleccionará aleatoriamente entre el *array* de recetas que tenemos, donde seleccionaremos una posición entre el 0 y el totalRecetas -1.



- 19- Una vez seleccionado el identificador de la receta aleatoriamente, lo insertaremos en la cola de recetas, en el *array* de nombres, y en el *array* de calorías.

```
// empezamos, de aqui recorremos todos los días que necesitamos para perder o ganar el peso que hemos indicado
for ($i = 1; $i <= $nrDays; $i++) {
    for ($nrMeal = 1; $nrMeal < 4; $nrMeal++) {
        $queueFoodsPerWeek[] = $nr_rice;
        $accuCaloriePerMeal = $cal_rice;

        mysqli_query($conn,
            "INSERT INTO meals (Plan_Nr, DayNr, Meal, Food_Nr, Food_Name, Calories) VALUES ('$nr_plan', '$i', '$nrMeal', '$nr_rice', 'Agua', '$cal_rice')")
            or die(mysqli_connect_errno());

        if (isset($queueFoodsPerWeek)) {
            do {
                $random = rand(0, count($queueFoods) - 1);
                $nrFood = $queueFoods[$random];
            } while (in_array($nrFood, $queueFoodsPerWeek));
        } else {
            $random = rand(0, count($queueFoods) - 1);
            $nrFood = $queueFoods[$random];
        }

        $queueFoodsPerWeek[] = $nrFood;
        $foodName = $queueFoodName[$random];
        $foodCalorie = $queueCalories[$random];

        mysqli_query($conn, "INSERT INTO meals (Plan_Nr, DayNr, Meal, Food_Nr, Food_Name, Calories)
            VALUES ('$nr_plan', '$i', '$nrMeal', '$nrFood', '$foodName', '$foodCalorie')")
            or die(mysqli_connect_errno());

        $accuCaloriePerMeal += $queueCalories[$random];
        $lackCalories = $calorieMeal - $accuCaloriePerMeal;

        for ($y = 0, $x = 0; $y < count($queueCalories); $y++) {
            $x = $y;
            if ($lackCalories <= $queueCalories[$y]) {
                break;
            }
        }

        switch ($goal) {
            case 'Ganar Peso':
                $option1 = abs($queueCalories[$x] - $lackCalories);
                $option2 = abs($queueCalories[$x - 1] - $lackCalories);
                break;
            case 'Perder Peso':
                $option1 = abs($queueCalories[$x] - $lackCalories);
                $option2 = abs($queueCalories[$x + 1] - $lackCalories);
                break;
            default:
                $option1 = abs($queueCalories[$x] - $lackCalories);
                $option2 = abs($queueCalories[$x - 1] - $lackCalories);
                break;
        }
    }
}
```

Fig. 46: Código del funcionamiento de creación de un Plan Nutricional 4.

- 20- Se realizará un INSERT en la base de datos con la información de la planificación, añadiendo el: Identificador de la receta, el día asignado, el momento de la comida (1, 2 o 3 son Desayuno, comida, y cena respectivamente), el nombre de la receta, y las calorías de la receta.
- 21- Una vez obtenida la segunda receta, se procederá a insertar la última de todas, para ello se debe calcular cuantas calorías quedan por sumar al cálculo de calorías por comida. Para ello se sumarán en el *array* de calorías las que se tengan contadas de momento, y luego restarlas a las calorías que tocan por comida. \rightarrow $caloriasPorComida - caloriasContabilizadas = caloriasQueFaltan$.
- 22- Con el conocimiento de las calorías que faltan por consumir, es necesario buscar en el *array* de comidas, alguna receta que tengas aproximadamente esa cantidad de calorías, por ello se realizará una búsqueda con un bucle *for*, donde se comparará la cantidad de calorías que falta, con la cantidad de calorías de una receta en la posición \$Y. Luego cuando se encuentre una posición que sea acorde a los criterios establecidos, se almacenará en la posición \$X.

23- Se analizará la opción que ha elegido el usuario de objetivo, y se decidirá entre dos posiciones diferentes en el *array* de recetas siendo diferente dependiendo de el objetivo configurado por el usuario.

```

$r = 0;
do {
    if (($option1 < $option2) && (isset($queueFoods[$x - $r]))) {
        $nrFood = $queueFoods[$x - $r];

        if (in_array($nrFood, $queueFoodsPerWeek)) {
            if (isset($queueFoods[$x + 1 + $r])) {
                $nrFood = $queueFoods[$x + 1 + $r];
                $foodName = $queueFoodName[$x + 1 + $r];
                $foodCalorie = $queueCalories[$x + 1 + $r];
            }
            else {
                $foodName = $queueFoodName[$x - $r];
                $foodCalorie = $queueCalories[$x - $r];
            }
        }
        elseif (isset($queueFoods[$x + 1 + $r])) {
            $nrFood = $queueFoods[$x + 1 + $r];

            if (in_array($nrFood, $queueFoodsPerWeek)) {
                if (isset($queueFoods[$x - $r])) {
                    $nrFood = $queueFoods[$x - $r];
                    $foodName = $queueFoodName[$x - $r];
                    $foodCalorie = $queueCalories[$x - $r];
                }
                else {
                    $foodName = $queueFoodName[$x + 1 + $r];
                    $foodCalorie = $queueCalories[$x + 1 + $r];
                }
            }
        }
    }

    if (!isset($queueFoods[$x - $r]) && !isset($queueFoods[$x + 1 - $r])) {
        break;
    }

    $r++;
} while (in_array($nrFood, $queueFoodsPerWeek));

$queueFoodsPerWeek[] = $nrFood;
mysqli_query($conn, "INSERT INTO meals (Plan_Nr, DayNr, Meal, Food_Nr, Food_Name, Calories)
VALUES ('$nr_plan', '$i', '$nrMeal', '$nrFood', '$foodName', '$foodCalorie');");
}

if ($i % 4 == 0) {
    unset($queueFoodsPerWeek);
}
}

```

Fig. 47: Código del funcionamiento de creación de un Plan Nutricional 5.

24- Habrá que comprobar las diferentes opciones seleccionadas, para ello se comprobará mediante iteraciones hasta encontrar la correcta y dentro de ella con un if la siguiente comprobación:

- a. Si(opcion1 < opcion2) Y existe(colaRecetas(\$X+\$r)
 - i. (\$r será la variable donde se incrementará en cada iteración hasta que se encuentre una receta adecuada).
- b. Si no existe en colaRecetas, se comprobaría si existe una posición más en ella: existe(colaRecetas(\$X + 1 + \$r))

25- Se comprobará además si la receta existe ya en la cola de recetas de la semana, para comprobar que no se repite ninguna receta durante la semana y así no aburrir al usuario.

26- Una vez seleccionada la tercera receta, únicamente faltará insertarla en la base de datos.



Con todo esto se redirigirá a la consulta de la dieta y se verá que se obtendrá una nueva planificación a partir de los datos introducidos por el usuario como se muestra en la figura 48.

Calculadora **Nutricional** PERFIL CONFIGURACION **DIETARIO** CONFIGURADOR ALIMENTARIO CERRAR SESION

Bienvenido Daniel Planificación Alimentaria

Dietario

Dieta Actual	Crear Dieta	Dietas Anteriores	
<p>Dieta Actual</p> <p>Objetivo: Dieta: Perder Peso Normal</p> <p>Estilo de vida marcado: Moderadamente Activo</p> <p>Calorías por día: Calorías por comida: 2437 812.33333333333</p>			
April 2020			
Domingo	Lunes	Martes	Miércoles
			1
			2
			3
4	5	6	7
	Almuerzo Agua Brochetas de calamar y salmón Carpaccio temera Comida Agua Tallarines de calabacín con queso fresco Tallarines de calabacín con pesto rojo Cena Agua Tortilla de patatas Parrillada de verduras Cambiar	Almuerzo Agua Pizza ligera de verduras Brochetas de pollo con salteado de verduras Comida Agua Crema de calabaza Tallarines de calabacín con pesto rojo Cena Agua Arroz Blanco Verduras salteadas con langostinos Cambiar	Almuerzo Agua Pollo al horno con patata asada Vichyssoise super ligera Comida Agua Pechuga de pollo al limon Pollo con crema de calabacín Cena Agua Merluza al vapor con verduritas Dorada con ajitos Cambiar
			8
			9
			10
			11
12	13	14	15
Almuerzo Agua Arroz Blanco Verduras salteadas con langostinos	Almuerzo Agua Sepia con alcachofas y champiñones	Almuerzo Agua Pizza ligera de verduras Brochetas de	Almuerzo Agua Tallarines de calabacín con queso fresco
			16
			17
			18
			19

Fig. 48: Interfaz de consulta del Plan Nutricional.

5.3.3.2 Fichero home.php

En este apartado el usuario podrá acceder a toda su información personal que esté relacionada con su cuenta.

Si el usuario quiere modificar algún aspecto de este perfil bastará con presionar sobre la pestaña Actualizar Datos donde aparecerán los inputs con la información del usuario para que sean modificados como en la figura 49.

Perfil de Usuario

Información Actualizar Datos

Actualizar información personal

Nombre:

Nombre Apellido Apellido2

Dirección:

Genero: Complejion:

Hombre [input]

Altura (cm): Peso (kg):

180 80

Tipo de Uñas:

Normal

Cancelar ACTUALIZAR

Fig. 49: Interfaz de Edición del perfil del usuario.

Cuando el usuario actualiza su perfil, se procederá a realizar ciertas comprobaciones y asignaciones de variables para recoger todos los datos relacionados con el usuario como se muestra en la figura 50:

1. Primero se accederá a los archivos de conexión de la base de datos para realizar las operaciones que se argumentarán a continuación.
2. Si se verifica que la operación que se ha realizado es de tipo SUBMIT se procederá a llamar al proceso de actualización de datos, para comprobar esta condición se tendrá de consultar la variable `$_POST["submit"]`.
3. Se almacenarán en variables la información actualizada del usuario que ha introducido en el formulario.
4. Se realizará una sentencia SQL donde se realizará un UPDATE con la nueva información del usuario.

```
<?php
include 'conn.php';
include 'check_login.php';

$user = $_SESSION['user'];
$pass = $_SESSION['pass'];
$msg = 0;

if (isset($_POST['submit'])) {
    $efname = $_POST['fname'];
    $emname = $_POST['mname'];
    $elname = $_POST['lname'];
    $eaddress = $_POST['address'];
    $egender = $_POST['gender'];
    $edob = $_POST['dob'];
    $eheight = $_POST['height'];
    $eweight = $_POST['weight'];

    mysqli_query($conn, "UPDATE users
        SET FName = '$efname', MName = '$emname', LName = '$elname', Address = '$eaddress',
            DoB = '$edob', Gender = '$egender', Height = '$eheight', Weight = '$eweight'
        WHERE Username = '$user' AND PassMd = '$pass'");

    if (mysqli_connect_errno()) {
        $msg = 1;
        $message = mysqli_connect_errno();
    } else {
        $msg = 2;
        $message = "Tu información personal ha sido actualizada";
    }
}

$q = mysqli_query($conn, "SELECT * FROM users WHERE Username = '$user' AND PassMd = '$pass'");
$r = mysqli_fetch_array($q);

$dob = $r['DoB'];
$height = $r['Height'];
$weight = $r['Weight'];
?>
```

Fig. 50: Código de edición del perfil del usuario.

5. Si todo se ha realizado correctamente, se redirigirá al usuario a la página de información personal como se muestra en la figura 51.

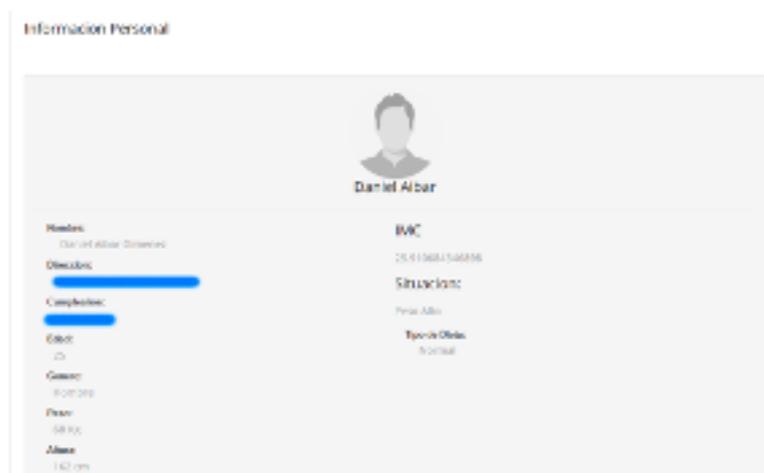


Fig. 51: Interfaz de consulta del perfil de usuario.

5.3.3.3 Fichero profile.php

Este apartado ofrecerá la capacidad de cambiar la contraseña al usuario como se muestra en la figura 52, para ello introducirá el correo y la anterior contraseña para verificar su identidad.

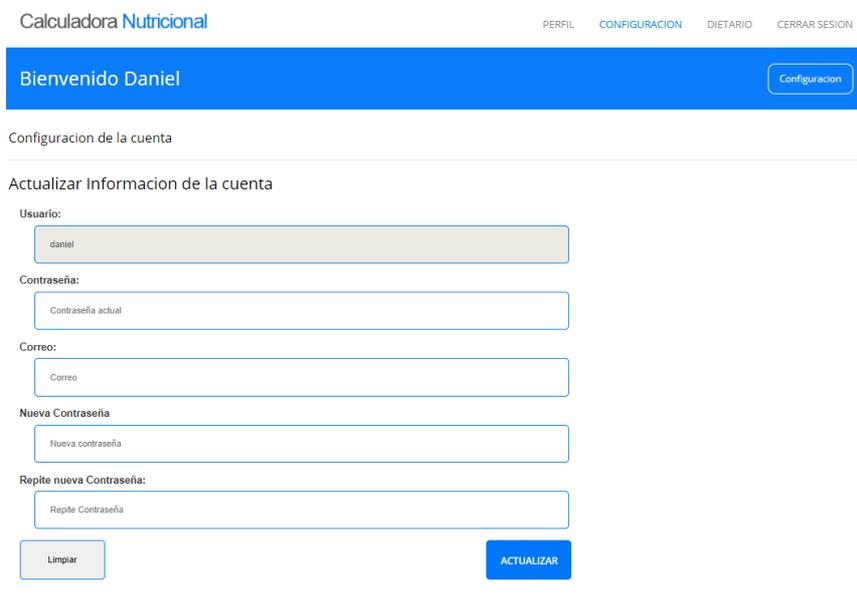


Fig. 52: Interfaz de edición de seguridad del usuario.

```
1 <?php
2 include 'conn.php';
3 include 'check_login.php';
4
5 $user = $_SESSION['user'];
6 $pass = $_SESSION['pass'];
7 $msg = 0;
8
9 if (isset($_POST['submit'])) {
10     $pass = md5($_POST['pass']);
11
12     if ($pass != $pass) {
13         $msg = 1;
14         $message = "La contraseña introducida no es correcta.";
15     } else {
16         $pass1 = $_POST['pass1'];
17         $pass2 = $_POST['pass2'];
18
19         if ($pass1 != $_POST['pass2']) {
20             $msg = 1;
21             $message = "The new passwords you provided did not match.";
22         } else {
23             $email = $_POST['email'];
24             $pass2 = md5($pass2);
25
26             mysqli_query($conn, "UPDATE users SET Pass = '$pass1', PassMd = '$pass2', Email = '$email' WHERE Username = '$user' AND PassMd = '$pass'");
27
28             if (mysqli_connect_errno()) {
29                 $msg = 1;
30                 $message = mysqli_connect_errno();
31             } else {
32                 $msg = 2;
33                 $message = "Tu informacion ha sido actualizada correctamente";
34             }
35         }
36     }
37 }
38
39 $q = mysqli_query($conn, "SELECT FName, LName, Email FROM users WHERE Username = '$user' AND PassMd = '$pass'");
40 $r = mysqli_fetch_array($q);
41
42 ?>
```

Fig. 53: Código de edición de la seguridad del usuario.

En la figura 53 se puede visualizar el código utilizado para realizar la operación de actualización de la contraseña, si el funcionamiento es correcto se le indicará al usuario mediante un mensaje que la información ha sido actualizada con éxito.

5.3.3.4 Fichero selector.php

En este apartado el usuario podrá elegir como si fuese la famosa red social *Tinder* las recetas que más le gusten tal y como se muestra en la figura 54, además de poder obtener gracias a sus elecciones, una configuración personalizada del plan nutricional que quiera crear.

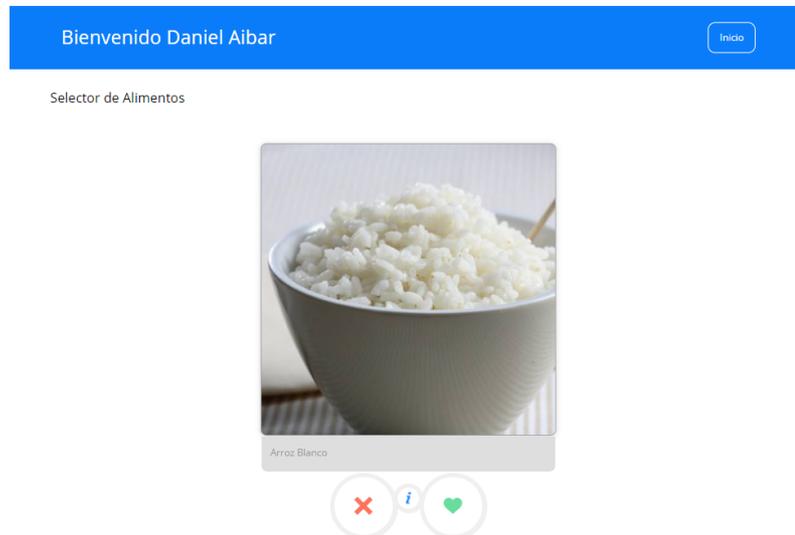


Fig. 54: Interfaz del selector de recetas.

De manera sencilla el usuario podrá escoger con un *like* o un *dislike* las recetas que se le muestren en esta sección.

```
<div class="app">
  <div class="content">
  </div>
  <div class="footer">
    <div id="swipe_dislike" class="rate"></div>
    <div class="info"></div>
    <div id="swipe_like" class="rate"></div>
  </div>
</div>
```

Fig. 55: Código HTML del selector de recetas.

Una vez esté preparada la página como se muestra en la figura 55, se iniciarán 3 funcionalidades indicadas en la figura 56:

1. Se añadirá al div con id *swipe_like* la propiedad *onclick* cuando se presione sobre la capa se procederá a llamar al método *swipeLike()*.
2. Se añadirá al div con id *swipe_dislike* la propiedad *onclick* cuando se presione sobre la capa se procederá a llamar al método *swipeLike()*.
3. Se llamará después a la función *addNewProfile()* que empezará todo el proceso para visualizar las recetas.

```

$(document).ready(function(event) {

    $("#div#swipe_like").on("click", function() {
        swipeLike();
    });

    $("#div#swipe_dislike").on("click", function() {
        swipeDislike();
    });

    addNewProfile();
});

```

Fig. 56: Código jQuery para el selector de recetas.

Cuando se llama a la función *addNewProfile()* mostrada en la figura 56, se procede a realizar los procesos mostrados en las figuras 57 y 58:

1. Se establece una conexión con los archivos de configuración de la base de datos desde la función.
2. Se crea un array llamada *names* donde se almacenan los nombres de las recetas, dentro de ella se llama a la base de datos para insertar mediante un recorrido todos los elementos que se han recibido en el *array*.

En esta *SQL* únicamente se recibirán las recetas que no haya seleccionado el usuario aún, es decir, las recetas que tiene pendiente de dar *LIKE* o *DISLIKE* y que se adapten al estilo nutricional que haya seleccionado el usuario en su perfil.

3. Se almacenará en una variable la receta que se vaya a mostrar.
4. Luego se creará un nuevo *array* para almacenar esta vez el id de la receta para que luego se pueda identificar bien la receta, este recorrido será parecido al del punto 2 pero con el identificador de la receta.
5. Se añadirá el identificador de receta en la variable *idReceipt*.

```

</php
include 'conn.php';
include 'check_login.php';
$username=$_SESSION['user'];
$password=$_SESSION['pass'];
$$=mysql_query("SELECT UserType FROM users WHERE Username = '$username' AND PassMd = '$password'");
$count=mysql_num_rows($$);

if($count==1){
    $q = mysql_fetch_array($$);
    $diet = $q['diet'];
}
?>

var names = [
</php
    $q_n = mysql_query($conn, "SELECT Name, Nr FROM foods WHERE $diet=1 AND Nr NOT IN
        (SELECT meal FROM likes WHERE user=$user GROUP BY meal) LIMIT 1 ASC");
    while ($n = mysql_fetch_array($q_n)) {
        $nombreReceta = $n['Name'];
    }
?>
    <?php echo $n['Name']; ?>,
</php } ?>
][cont];

nombreReceta = names;
var idFood = [
</php
    $q_n = mysql_query($conn, "SELECT * FROM foods WHERE $diet=1 AND WHERE Nr NOT IN
        (SELECT meal FROM Likes WHERE user=$user GROUP BY meal) LIMIT 1 ASC");
    while ($n = mysql_fetch_array($q_n)) { ?>
        <?php echo $n['Nr']; ?>,
    </php } ?>
][cont];

idReceta = idFood;

var idPhoto = [
</php
    $q_n = mysql_query($conn, "SELECT * FROM foods WHERE $diet=1 AND Nr NOT IN
        (SELECT meal FROM Likes WHERE user=$user GROUP BY meal) LIMIT 1 ASC");
    while ($n = mysql_fetch_array($q_n)) {
    ?>
        <?php echo $n['photo']; ?>,
    </php } ?>
][cont];

idReceta = idFood;

```

Fig. 57: Código de inicio del selector de recetas.

6. Luego se comprobará que las variables contengan datos, donde si todo es correcto se le añadirá al div *content* ciertos elementos visuales para insertar el contenido visual y ver la receta.

Si no fuese así y no contiene ningún dato, significaría que no tiene ninguna receta que verificar, entonces se le añadirá una capa visual indicándole que no tiene más recetas que seleccionar.

```

} else {
    $( "div.content" ).prepend( '<div class="photo" id="photo" style="background-image:url(/nutricional/imagesSelector/' + foto + '.jpg)'> +
        '<span class="meta">' +
        '<p>' + names + '</p>' +
        +
        '</span>' +
        '</div>' );
    foto++;
    cont++;
}

```

Fig. 58: Código del selector de recetas 2.

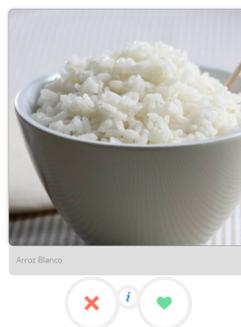


Fig. 59: Interfaz del selector de recetas al realizar el código.

Cuando el usuario presione el botón *LIKE* se realizarán las funcionalidades mostradas en las figuras 60 y 61:

1. Se guardará en una variable la capa *div* que tenga como identificador *photo*.
2. Se llamará a la librería de *JavaScript TimelineMax* [14] que es la responsable de poder desarrollar los movimientos de la capa. En esta llamada se le configurará los parámetros que se quieren realizar y se indicará cual es la capa que debe realizar este movimiento, en este caso es *\$photo*.
3. Se creará una variable *params* donde se almacenará la información de la receta a la que el usuario le ha dado *Like*, exactamente se guardará el *id* de la receta, y que opción ha seleccionado, que será para *Like* un 1 y para *Dislike* un 0.
4. Se procederá a realizar una llamada *Ajax* al archivo donde será procesada la petición, en este caso se llamará *recommendedFood.php*.

```
function swipeLike() {  
  
    var $photo = $("div.content").find('#photo');  
  
    var swipe = new TimelineMax({  
        repeat: 0,  
        yoyo: false,  
        repeatDelay: 0,  
        onComplete: remove,  
        onCompleteParams: [$photo]  
    });  
    swipe.staggerTo($photo, 0.8, {  
        bezier: [{  
            left: "+=400",  
            top: "+=300",  
            rotation: "60"  
        }],  
        ease: Power1.easeInOut  
    });  
  
    var params = {  
        "data" : idReceta,  
        "like" : 1  
    };  
    $.ajax({  
        data: params,  
        url: 'recommendedFood.php',  
        type: 'post',  
        beforeSend: function () {  
        },  
        success: function (response) {  
            addNewProfile();  
        }  
    });  
}
```

Fig. 60: Código del selector de recetas al dar like.

5. Cuando se accede al archivo *recommendedFoods.php* el primer paso será realizar las conexiones con la base de datos para ejecutar las operaciones que se comentarán a continuación.
6. Se almacenará en variables la información que se han recibido de los parámetros enviados en la solicitud *Ajax*.

7. Se realizará un INSERT en la base de datos en la tabla *likes* donde se indicará el usuario, la receta y la opción que haya marcado el usuario de la receta.

```
recommendedFood.php > ...
1 <?php
2 include 'conn.php';
3 include 'check_login.php';
4
5 $user = $_SESSION['user'];
6 $pass = $_SESSION['pass'];
7
8 $q = mysqli_query($conn, "SELECT * FROM users WHERE Username = '$user' AND PassMd = '$pass'");
9 $r = mysqli_fetch_array($q);
10 $id = $r['Nr'];
11
12 $response = $_POST['data'];
13 $like = $_POST['like'];
14
15 mysqli_query($conn, "INSERT INTO likes (user, meal, gusta) VALUES ($id, $response, $like)");
16
17 if (mysqli_connect_errno()) {
18     $msg = 1;
19     $message = mysqli_connect_errno();
20 } else {
21     $msg = 2;
22     $message = "Receta subida!";
23 }
24
25 echo $message;
26 >>
```

Fig. 61: Código PHP del selector de recetas al dar like

Una vez comentados los apartados que componen la sección de usuario, se pasará a explicar los ficheros y las funcionalidades que tendrán los administradores de la aplicación.

5.3.4 Capa Administrador

5.3.4.1 Fichero *index.php*

Una vez que se haya iniciado sesión, si el usuario se registra con el role *Admin* se le redirigirá hacia la sección de administración de la aplicación. La primera página a la que accede es *index.php*, en ella se encuentra la información personal del usuario y en un panel lateral las opciones que tendrá disponibles como en la figura 62.

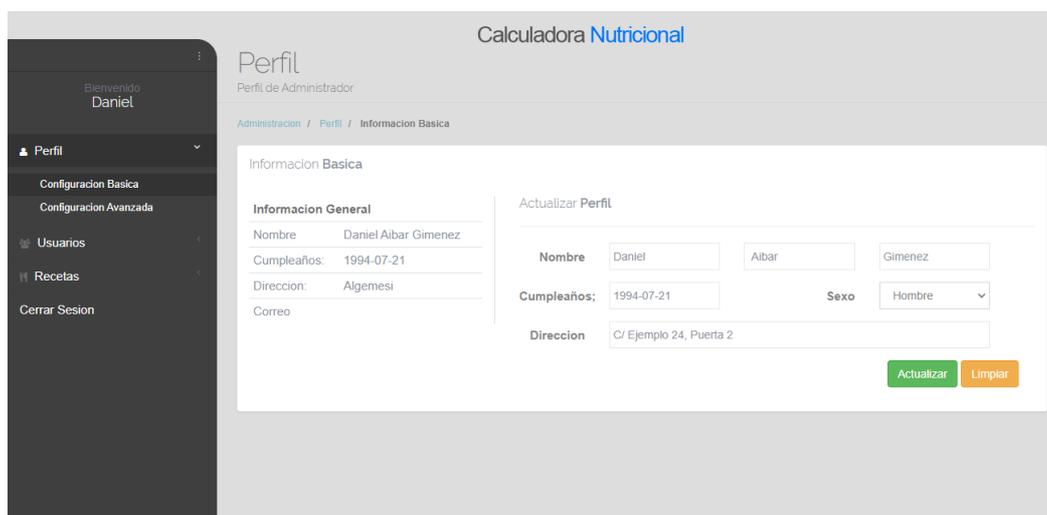


Fig. 62: Interfaz de inicio del Administrador.

Entre las opciones que dispone un usuario de tipo administrador se encuentran las siguientes:

- Consulta y edición de perfil de usuario
- Registro de nuevos usuarios
- Consulta y edición de usuarios existentes
- Registro de recetas.
- Consulta y edición de recetas existentes.

5.3.4.2 Users.php

Si el usuario accede al panel de Usuarios, se le mostrará un listado de todos los usuarios que están creados en la aplicación a modo de tabla como en la figura 63, donde le aparecerán diferentes opciones, como *Consultar*, *Editar* y *Eliminar*.

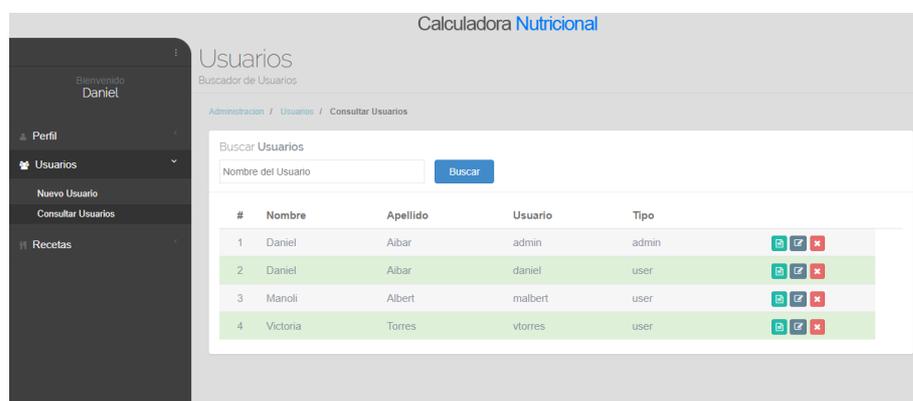
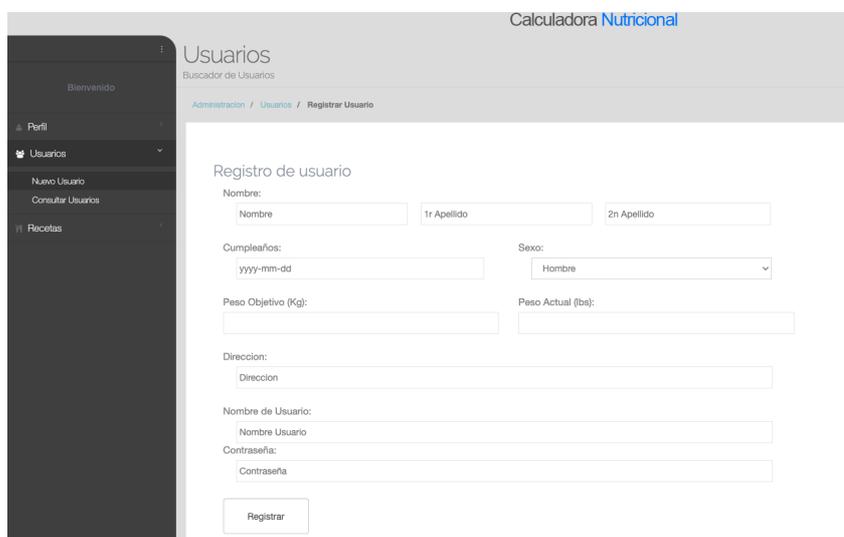


Fig. 63: Interfaz de consulta de usuarios.

Si por el contrario, el Administrador quiere registrar un usuario, accederá a la sección “Nuevo Usuario”, donde aparecerá un formulario como en la figura 64 que rellenará con los datos necesarios para crear el nuevo usuario.

The screenshot shows the 'Registro de usuario' form in the 'Calculadora Nutricional' application. The sidebar is the same as in Figure 63. The main content area is titled 'Usuarios' and 'Buscador de Usuarios', with a breadcrumb trail 'Administración / Usuarios / Registrar Usuario'. The form fields are:

- Nombre: Three input fields for 'Nombre', '1r Apellido', and '2n Apellido'.
- Cumpleaños: A date input field with the format 'yyyy-mm-dd'.
- Sexo: A dropdown menu with 'Hombre' selected.
- Peso Objetivo (Kg): An input field.
- Peso Actual (lbs): An input field.
- Dirección: An input field.
- Nombre de Usuario: An input field.
- Contraseña: An input field.

A 'Registrar' button is located at the bottom of the form.

Fig. 64: Interfaz de registro de usuario.

Para registrar un usuario en la aplicación se seguirán las instrucciones mostradas en la figura 65:

1. Una vez que el Administrador pulse el botón de registrar, se recogerá en la variable `$_POST['reg_username']` el nombre de usuario con el que quiere ser registrado.
2. Mediante una consulta *SQL*, se revisará si en la base de datos existe un usuario con el mismo nombre, si no es así el resultado de la consulta deberá ser 0, por el contrario, debería de registrar un valor 1, indicando que existe un usuario con el mismo nombre.
3. Una vez comprobado que el usuario no existe en la aplicación, se almacenará en variables toda la información rellena en el formulario.
4. Con todos los datos almacenados, se procede a realizar una operación *INSERT* en la tabla *Users* de la base de datos, donde se guardará toda la información.
5. Si todo funciona correctamente se almacenarán las credenciales de usuario en las variables de sesión, y se redirigirá al inicio del panel de Administración.

```
$reg_user = $_POST['reg_username'];

$sql_user = mysqli_query($conn, "SELECT Nr FROM users WHERE Username = '$reg_user'");

$n = mysqli_num_rows($sql_user);

if ($n > 0) {
    $msg = 1;
    $message = "El nombre de usuario no esta disponible, seleccione otro." . mysqli_connect_errno();
} else {
    $fname = $_POST['fname'];
    $mname = $_POST['mname'];
    $lname = $_POST['lname'];
    $dob = $_POST['dob'];
    $gender = $_POST['gender'];
    $height = $_POST['height'];
    $weight = $_POST['weight'];
    $address = $_POST['address'];
    $reg_pass = $_POST['reg_pass'];
    $md_pass = md5($reg_pass);

    mysqli_query($conn, "INSERT INTO users (FName, MName, LName, DoB, Gender, Height, Weight, Address, UserType, Username, Pass, PassMd) VALUES ('$fname', '$mname', '$lname', '$dob', '$gender', '$height', '$weight', '$address', 'user', '$reg_user', '$reg_pass', '$md_pass')");

    if (mysqli_connect_errno()) {
        $msg = 1;
        $message = mysqli_connect_errno();
    } else {
        $_SESSION['user'] = $reg_user;
        $_SESSION['pass'] = $md_pass;
        $_SESSION['type'] = "user";
        header("Location: home.php");
    }
}
```

Fig. 65: Código de registro de usuario.

5.3.4.3 Fichero *recipes.php*

Si el usuario accede al panel de “Recetas”, se le mostrará un listado de todas las recetas que están creadas en la aplicación en forma tabla como en la figura 66, donde aparecerán diferentes opciones, como *Consultar*, *Editar* y *Eliminar*.

#	Nombre	Categoria	Calorias			
1	Agua	Bebida	0			
2	Alcachofas con ternera picada	Receta	60			
3	Anillas de calamar al vino con alcachofas	Receta	305			
4	Arroz Blanco	Receta	204			
5	Brochetas de calamar y salmón	Receta	330			
6	Brochetas de langostino con calabacín	Fruta	130			
7	Brochetas de pollo con salteado de verduras	Receta	543			
8	Brochetas de pollo y verduras	Receta	653			
9	Carne de caballo con ajos tiernos	Bebida	350			

Fig. 66: Interfaz de consulta de recetas.

Si el usuario quiere crear una nueva receta deberá acceder al apartado de “Añadir Recetas” dentro del apartado “Recetas”. En ella, se encontrará con un formulario para rellenar los datos necesarios para crear la nueva receta, además de añadir los nutrientes correspondientes a la nueva receta.

Para registrar una nueva receta se realizarán las instrucciones mostradas en la figura 67:

1. Al pulsar sobre el botón de añadir, se almacenarán los datos recogidos en el formulario en variables.
Para los nutrientes al tratarse de inserciones opcionales se realizará un *array* para almacenar todos los que se hayan añadido.

```
$("#submit").click(function(event){
    var nam = $("#name").val();
    var cate = $("#category").val();
    var calo = $("#calories").val();
    var serv = $("#serving").val();
    var desc = $("#description").val();
    var num = $("#num").val();

    var nut = new Array();
    var nutVal = new Array();
    var msgType = "";
    var msg = "";

    for (var i = 1; i <= num; i++) {
        nut.push($("#nutrient" + i).val());
        nutVal.push($("#value" + i).val());
    };

    $.post(
        "ajax.php",
        {
            action: 'addfood',
            name: nam,
            cate: cate,
            calo: calo,
            serv: serv,
            desc: desc,
            num: num,
            nut: nut,
            nutVal: nutVal
        },
        function(data) {
            if (data != 1) {
                msgType = "danger";
                msg = "<strong>Error</strong><br>" + data;
            } else {
                msgType = "success";
                msg = "<strong>Completado! </strong><br>Se ha registrado la receta correctamente.";
            }
        });
});
```

Fig. 67: Código de creación de receta.

2. Se realizará una llamada *Post* en *JQuery*, donde se enviarán los datos por parámetros y se insertarán todas las variables utilizadas en el punto 1.

Para realización de las siguientes operaciones, se accederá al fichero *ajax.php*, el cual ejecutará la operación con el *action* “*addFood*” indicado como parámetro en el código de la figura 68.

```
switch ($action) {
    case 'addFood':
        $name = $_POST['name'];
        $cate = $_POST['cate'];
        $calo = $_POST['calo'];
        $serv = $_POST['serv'];
        $desc = $_POST['desc'];
        $num = $_POST['num'];
        $msg = "1";

        mysqli_query($conn, "INSERT INTO foods (Name, Description, Category, Serving, Calories) VALUES ('$name', '$desc', '$cate', '$serv', '$calo')");

        if (mysqli_connect_errno()) {
            $msg = mysqli_connect_errno();
        } else {
            if ($num > 0) {
                $q = mysqli_query($conn, "SELECT Nr FROM foods WHERE Name = '$name' AND Category = '$cate' AND Description = '$desc' AND Serving = '$serv' AND Calories = '$calo'");
                $n = mysqli_fetch_array($q);

                $id = $n['Nr'];
                $nut = $_POST['nut'];
                $nutVal = $_POST['nutVal'];

                for ($i=0; $i < $num; $i++) {
                    mysqli_query($conn, "INSERT INTO nutrients (Food_Nr, NutriName, Value) VALUES ('$id', '$nut[$i]', '$nutVal[$i]');");
                    if (mysqli_connect_errno()) {
                        $msg = mysqli_connect_errno();
                        break;
                    }
                }
            }
        }
    }
}
```

Fig. 68: Código de creación de receta 2.

3. Una vez dentro del archivo *ajax.php* se accederá al caso *addfood* en donde guardaremos las variables que se han obtenido.
4. Se realizará una operación *INSERT* en *Sql* donde se guardarán las variables relacionadas con las recetas.
5. Si la operación ha sido ejecutada correctamente se procederá a realizar una nueva operación *Sql* de *CONSULT* para encontrar el identificador que ha obtenido la nueva receta al crearse en la base de datos.
6. Con el identificador ya obtenido, se realizará una nueva operación de *INSERT* para almacenar en la tabla *nutrients* toda la información de los nutrientes que se ha introducido en el formulario.
7. Si todo ha funcionado correctamente se notificará al administrador que la operación ha sido realizada con éxito, de lo contrario se le mostrará un mensaje indicándole el error producido.

6. Conclusiones

Además de cumplir con su finalidad, esta aplicación aporta al usuario cierta participación en las tareas de selección de recetas, invitándole a ser más proactivo dentro de la aplicación.

Para llevar a cabo toda esta aplicación, se tuvo que plantear las necesidades reales de los usuarios que iban a utilizar esta aplicación, detallando cada característica y viendo la funcionalidad que debían de obtener. Tras este planteamiento se detalló una estructura de como iba a ser desarrollado este proyecto, utilizando conceptos obtenidos durante los años en el Grado de Ingeniería Informática como los Diseños de la Interfaz y la estructura de la Base de Datos para después realizar el Diseño de la aplicación y la Programación.

La finalidad de esta aplicación era el desarrollar un sistema de planificación nutricional donde un nutricionista pueda gestionar fácilmente a sus usuarios. Con todo el desarrollo que se ha mostrado en los apartados anteriores, la aplicación esta en un momento que podría publicarse para su uso, dado que durante todo el proceso se ha intentado que cumpliera con todos los requisitos necesarios.

Durante este proceso hemos marcado los objetivos que debe cumplir la aplicación para poder publicarse:

- Los usuarios, pueden acceder a una aplicación en la cual pueden crear nuevas planificaciones nutricionales y personalizar sus elecciones como más le guste.
- Los nutricionistas (administradores), tienen una aplicación en la cual pueden crear y modificar nuevos usuarios y nuevas recetas.

Para mí, la realización de esta aplicación ha sido una tarea verdaderamente divertida con la que he podido mejorar mis conocimientos de programación poniendo en práctica muchos conceptos adquiridos en el Grado de Ingeniería Informática.

7. Futuras mejoras

La aplicación desarrollada como se ha comentado en el punto 6 de conclusiones cumpliría los requisitos para ser publicada, pero actualmente se le puede agregar muchas más funcionalidades que se han quedado fuera de la planificación inicial para convertirla en una aplicación más completa.

Una de las funcionalidades que podría mejorar el uso de esta aplicación sería el gestor de citas con el nutricionista. Con esta mejora, el usuario podría solicitar con un simple *click* una cita de revisión con su nutricionista, el uso sería sencillo, el usuario podría visualizar los días libres que tiene su nutricionista en un calendario y seleccionará el día que le vaya mejor.

Cuando el usuario haya creado la cita, se le notificaría al nutricionista mediante un mensaje. Además, el nutricionista que accederá a la aplicación podrá visualizar y gestionar todas las citas que tiene en el calendario con sus clientes.

Otra posible mejora, podría ser el poder otorgar al nutricionista de un sistema de diagnóstico, donde cuando visualice a un usuario, pueda obtener mediante estadísticas, gráficos y otra información todos los datos registrados del usuario, en donde poder analizar sus cambios de peso, la nutrición que ha llevado a cabo, etc.

Otra posible nueva funcionalidad sería la de poder compartir diferentes recetas entre usuarios además de poder realizar valoraciones de cada una de ellas y que los usuarios puedan comentarlas, y solicitar que se incluyan en su plan nutricional.

Como mejoras internas de la aplicación podríamos indicar lo siguiente:

- Mejorar la carga del contenido cuando un usuario quiere crear o visualizar el plan nutricional.
- Mejora en la usabilidad del panel de administración del nutricionista.

8. Bibliografía

[1] Wikipedia - Top-down y bottom-up. [Consulta: 2 de julio de 2020]:

https://es.wikipedia.org/wiki/Top-down_y_bottom-up

[2] Wikipedia - Servidor HTTP Apache. [Consulta: 2 de julio de 2020]:

https://es.wikipedia.org/wiki/Servidor_HTTP_Apache

[3] Wikipedia - Alojamiento web. [Consulta: 2 de julio de 2020]:

https://es.wikipedia.org/wiki/Alojamiento_web

[4] Git - Acerca del Control de Versiones. [Consulta: 2 de julio de 2020]:

<https://git-scm.com/book/es/v2/Inicio---Sobre-el-Control-de-Versiones-Acerca-del-Control-de-Versiones>

[5] Lara Perez - Como crear un boceto web o Wireframe. [Consulta: 2 de julio de 2020]:

<https://www.palbin.com/es/blog/p1027-como-crear-un-boceto-web-o-wireframe.html>

[6] Online Visual Paradigm. [Consulta: 2 de julio de 2020]:

<https://online.visual-paradigm.com/es/>

[7] PHP - ¿Qué es PHP? [Consulta: 2 de julio de 2020]:

<https://www.php.net/manual/es/intro-what-is.php>

[8] Developers Mozilla - ¿Qué es JavaScript? [Consulta: 2 de julio de 2020]:

https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/Qu%C3%A9_es_JavaScript

[9] Laura Chuburu - Qué es jQuery y como implementarlo. [Consulta: 2 de julio de 2020]:

<https://www.laurachuburu.com.ar/tutoriales/que-es-jquery-y-como-implementarlo.php>

[10] Openclassrooms - ¿Qué es jQuery? [Consulta: 2 de julio de 2020]:

<https://openclassrooms.com/en/courses/4309491-simplifica-tus-proyectos-con-jquery/4309498-que-es-jquery>

[11] Wikipedia - MySQL. [Consulta: 2 de julio de 2020]:

<https://es.wikipedia.org/wiki/MySQL>

[12] Wikipedia - Git [Consulta: 2 de julio de 2020]:

<https://es.wikipedia.org/wiki/Git>

[13] GitExtensions [Consulta: 2 de julio de 2020]:

<http://gitextensions.github.io/>

[14] GreenSock. TimelineMax [Consulta: 2 de julio de 2020]:

<https://greensock.com/docs/v2/TimelineMax>