



UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN

**Desarrollo de una aplicación Web/móvil que apoye a los profesionales y
pacientes en el área de Nutrición y Dietética.**

Trabajo Especial de Grado presentado ante la ilustre
Universidad Central de Venezuela por:

Br. Valerie Da Silva.

Br. Victoria Da Silva.

Para optar por el título de Licenciado en Computación

Tutores:

Profa. Concettina Di Vasta

Prof. José Sánchez

UCV Caracas, Julio 2019

ACTA

Quienes suscriben, miembros del Jurado designado por el Consejo de Escuela de Computación, para examinar el Trabajo Especial de Grado presentado por las Bachilleres Valerie María Da Silva De Amoreira C.I. 23.563.654 y Victoria María Da Silva De Amoreira C.I. 23.563.653, con el título "DESARROLLO DE UNA APLICACIÓN WEB/MÓVIL QUE APOYE A LOS PROFESIONALES Y PACIENTES EN EL ÁREA DE NUTRICIÓN Y DIETÉTICA" a los fines de optar al título de Licenciada en Computación, dejan constancia de lo siguiente:

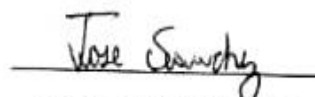
Dicho trabajo, leído por cada uno de los miembros del jurado, se fijó el día 26 de Julio de 2019, a las 12:30 p.m., para que sus autoras lo defendieran en forma pública en PA III de la Escuela de Computación, mediante una presentación oral de su contenido, luego de lo cual se respondieron las preguntas formuladas. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobarlo con la nota de 20 puntos.

En fe de lo cual se levanta la presente Acta, en Caracas a los 26 días del mes de Julio del año 2019.



Profa. Concettina Di Vasta

Tutora



Prof. José Sánchez

Tutor



Profa. Mercy Ospina

Jurado Principal



Prof. Antonio Silva

Jurado Principal

UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN

**Desarrollo de una aplicación Web/móvil que apoye a los profesionales y
pacientes en el área de Nutrición y Dietética.**

Autores: Valerie Da Silva

Victoria Da Silva

Tutores: Profa. Concettina Di Vasta

Prof. José Sánchez

Fecha: Julio de 2019

RESUMEN

El presente Trabajo Especial de Grado, describe la problemática que presentan los profesionales y pacientes del área de Nutrición y Dietética en Venezuela, para ofrecer a los nutricionistas, distintas funcionalidades para llevar el control del estado nutricional de todos sus pacientes y sus respectivos planes dieto-terapéuticos, así como también, permitir a sus pacientes llevar un seguimiento continuo de su tratamiento, y observar, los avances logrados a lo largo del mismo. Conociendo las necesidades e inconvenientes existentes, se plantea el desarrollo de una aplicación Web/móvil que apoye a profesionales y pacientes del área de Nutrición y Dietética, incorporando un módulo de soporte para los pacientes a la aplicación Web ya desarrollada, llamada SysNutr, una aplicación Web/móvil creada con tecnologías de software libre (HTML5, CSS3, JavaScript, jQuery, Bootstrap, PHP, Laravel, IONIC, y PostgreSQL), bajo una adaptación de la metodología ágil eXtreme Programming (XP), la cual permite consultar y gestionar los datos del paciente, realizar cálculos automatizados, crear, gestionar y modificar menús y citas; y además, por parte del paciente, permite observar información relevante de su tratamiento, ayudando a simplificar las actividades realizadas por parte de los nutricionistas y sus pacientes.

Palabras clave: nutrición, nutricionista, paciente, gestión de pacientes, seguimiento continuo, menús, plan dieto-terapéutico, aplicación Web/móvil.

TABLA DE CONTENIDO

INTRODUCCIÓN	1
CAPÍTULO I: PROBLEMA DE INVESTIGACIÓN	3
1.1. PLANTEAMIENTO DEL PROBLEMA	3
1.2. OBJETIVOS.....	5
1.2.1. <i>General</i>	5
1.2.2. <i>Específicos</i>	6
1.3. ARQUITECTURA DE LA SOLUCIÓN.....	6
1.4. JUSTIFICACIÓN DE LA SOLUCIÓN.....	8
CAPÍTULO II: MARCO CONCEPTUAL	9
2.1. ANTECEDENTES.....	9
2.1.1. <i>NUTRIUM</i>	10
2.1.2. <i>NUTRIMIND</i>	14
2.1.3. <i>SYSNUTR</i>	18
2.2. CONCEPTOS BÁSICOS RELACIONADOS AL ÁREA DE NUTRICIÓN	20
2.3. ARQUITECTURA CLIENTE-SERVIDOR.....	22
2.4. APLICACIONES WEB.....	25
2.4.1. <i>Definición</i>	25
2.4.2. <i>Características generales</i>	25
2.4.3. <i>Ventajas de las aplicaciones Web</i>	27
2.4.4. <i>Desventajas de las aplicaciones Web</i>	27
2.4.5. <i>Arquitectura MVC (Modelo Vista Controlador)</i>	28
2.5. APLICACIONES MÓVILES	30
2.5.1. <i>Definición</i>	30
2.5.2. <i>Aplicaciones móviles nativas</i>	31
2.5.3. <i>Aplicaciones móviles híbridas</i>	33
2.6. HERRAMIENTAS TECNOLÓGICAS PARA EL DESARROLLO DE APLICACIONES WEB... 34	
2.6.1. <i>Tecnologías del lado del cliente</i>	34
2.6.2. <i>Tecnologías del lado del servidor</i>	39
2.7. HERRAMIENTAS TECNOLÓGICAS PARA EL DESARROLLO DE APLICACIONES MÓVILES	
46	
2.7.1. <i>IONIC</i>	47
CAPÍTULO III: MARCO METODOLÓGICO	49
3.1. PROGRAMACIÓN EXTREMA (XP)	49
3.1.1. <i>Artefactos XP</i>	50
3.1.2. <i>Roles</i>	51
3.1.3. <i>Fases o Etapas</i>	52
CAPÍTULO IV: MARCO APLICATIVO	56
4.1. FASES DEL PROYECTO.....	56

4.1.1. Fase de planificación	57
4.1.2. Fase de diseño	62
4.1.3. Fase de codificación	67
4.1.4. Fase de pruebas.....	74
4.2. FLUJO DE LA APLICACIÓN WEB/MÓVIL COMO USUARIO PACIENTE	81
4.3. FLUJO DE LA APLICACIÓN WEB/MÓVIL COMO USUARIO NUTRICIONISTA	96
CONCLUSIONES	111
RECOMENDACIONES Y TRABAJOS FUTUROS.....	113
REFERENCIAS BIBLIOGRÁFICAS Y DIGITALES	114

INTRODUCCIÓN

En la actualidad, la gran mayoría de las personas no le dan la suficiente importancia al tema de la alimentación. Los seres humanos, generalmente no toman en cuenta la calidad y cantidad de los alimentos que ingieren, y lo cual debe ser de suma importancia, ya que, ingerir alimentos en mal estado, por cualidades como la textura, apariencia o la higiene; y consumir porciones de alimentos que solo satisfagan su apetito; puede traer como consecuencia, una mala nutrición y conllevar a enfermedades como hipertensión arterial, diabetes, obesidad, desnutrición, entre otras.

La Nutrición y Dietética, se ocupa de estudiar el proceso biológico a través del cual el organismo metaboliza los nutrientes que ingiere a través de los alimentos, y los transforma en la energía necesaria para mantenerse en funcionamiento. Asimismo, se dedica a diagnosticar y tratar aquellas enfermedades ocasionadas por una nutrición deficiente, como las mencionadas anteriormente, a través de la implementación de planes dieto-terapéuticos que se adapten a las características biológicas, antropométricas y fisiológicas de cada paciente, con el fin de garantizar el funcionamiento óptimo del organismo.

Para poder realizar un óptimo plan dieto-terapéutico para un paciente, se necesita tomar en cuenta un conjunto de datos como: la edad, género, la complexión física, el tipo de vida que desarrolle (sedentaria, activa), índice de masa corporal (IMC), requerimientos calóricos, estado nutricional, entre otros. Dichos datos son esenciales, ya que la alimentación no será igual para todos los pacientes.

Actualmente existen algunas aplicaciones que apoyan a los profesionales del área de nutrición, haciendo su trabajo mucho más fácil, como realizar la gestión de la información de sus pacientes, evitando así la necesidad de utilizar papel y lápiz, o en ocasiones, hojas de Excel. Sin embargo, se considera que dichas aplicaciones no apoyan de igual forma a los pacientes del área, ya que no cuentan con herramientas gratuitas que posean toda la información acerca de su tratamiento, y se dificulta la comunicación entre el paciente y el nutricionista, causando, mucha deserción o abandono de los planes nutricionales planteados por los nutricionistas.

Dada la situación planteada, el objetivo del siguiente trabajo de investigación es desarrollar una aplicación Web/móvil que apoye a los profesionales y pacientes en el área de Nutrición y Dietética; y para ello se debe implementar el módulo de soporte

para los pacientes en la aplicación Web, ya existente llamada SysNutr. A través de dicho módulo, los pacientes pueden observar, sus avances logrados a lo largo de su tratamiento, y los estímulos recibidos de acuerdo a su evolución. Además, se generan indicadores de forma gráfica, que son de interés para los médicos nutriólogos.

El presente documento, se estructura en cuatro (4) capítulos, y los cuales, se describen brevemente a continuación.

El capítulo 1: Problema de investigación, en el cual se describen los problemas presentes, además el objetivo general y específicos del presente trabajo de investigación, así como también, la solución al problema, la arquitectura y la justificación de la solución.

El capítulo 2: Marco Conceptual, formado por los antecedentes del trabajo y los conceptos principales, necesarios para el desarrollo del Trabajo Especial de Grado (TEG).

El capítulo 3: Marco Metodológico, en donde se describe la metodología que es seleccionada para ser utilizada en el desarrollo del TEG.

El capítulo 4: Marco Aplicativo; en dicho capítulo se describen las fases de la metodología seleccionada para el desarrollo de la aplicación Web/móvil, y se explican sus características, tales como: arquitectura utilizada, base de datos, funcionalidades, e interfaces y pruebas realizadas para determinar el correcto funcionamiento de la aplicación.

Por último, se reportan las conclusiones obtenidas tras la elaboración del presente TEG. También se proporcionan las recomendaciones, trabajos futuros y referencias bibliográficas.

CAPÍTULO I: PROBLEMA DE INVESTIGACIÓN

En el presente capítulo se proporciona información acerca del planteamiento y delimitación de la problemática, el objetivo general y los objetivos específicos, y finalmente, la arquitectura y la justificación del TEG, descritos a continuación:

1.1. Planteamiento del problema

Los seres humanos tienen necesidades básicas como lo son respirar, ingerir líquidos y alimentarse, sin embargo, los mismos tienden a desordenar su alimentación, ya que actualmente existe una falta de conocimiento acerca de la nutrición, que existe en diversas sociedades, independientemente de si se encuentran en países desarrollados, o no; por otro lado, existe una cultura obsesionada con la comida rápida, la dieta y la imagen corporal, y por lo cual, adquieren problemas de nutrición que pueden causar trastornos nutricionales como: bulimia, anorexia y obesidad. De acuerdo a sus edades, se puede conocer que las necesidades nutricionales de cada ser humano, pueden cambiar.

La alimentación saludable a lo largo de toda la vida, debe ser natural, nutritiva y equilibrada, no obstante, para llevar una alimentación adecuada se deben tomar en cuenta ciertos aspectos, como por ejemplo, comer al menos 4 o 5 comidas al día, ingerir suficiente cantidad de agua, reducir o evitar el consumo de bebidas alcohólicas o azucaradas y dulces, crear buenos hábitos alimenticios, entre otros. Es importante destacar, que la alimentación no será igual en todos los adultos, ya que dependerá de la edad, la complexión física, el tipo de vida que desarrolle (sedentaria, activa), y en ciertos casos, si padecen de problemas en su estado de salud (diabetes, obesidad, hipertensión, entre otros), plantea (Elisa Blazquez Blanco, 2017), y para los cuales necesitan llevar una dieta específica que permita mejorar los valores bioquímicos corporales, y por último; los problemas socioeconómicos de cada individuo; en donde una persona de un sustrato medio alto, no dispone del mismo poder adquisitivo que el de una persona de sustrato social medio bajo. Por lo antes mencionado, los seres humanos buscan apoyo en profesionales de la medicina, específicamente en el área de la nutrición, los cuales, hoy en día cuentan con aplicaciones existentes para llevar el control nutricional de las diferentes personas, de acuerdo a sus diferentes necesidades y requerimientos, y tomando en cuenta, adicionalmente, su poder

adquisitivo. Dichas aplicaciones, ayudan al médico nutriólogo a llevar información personalizada de los diferentes pacientes, según sus requisitos, sin la necesidad de invertir tanto en gestionar dicha información en archivos físicos, sino en presentar el seguimiento del desarrollo del paciente, desde el inicio del mismo hasta su culminación, mediante gráficos de fácil entendimiento, y con datos precisos a nivel de medidas y pesos de los mismos, lo cual es algo de gran utilidad para los profesionales del área de nutrición.

En la actualidad existen aplicaciones diseñadas para apoyar al médico nutriólogo, y al paciente, durante el proceso de evaluación nutricional. En primer lugar, se hace mención a Nutrium, la cual es una aplicación que fue diseñada para simplificar las tareas de los profesionales de nutrición, y ayuda a los pacientes a alcanzar sus objetivos. Sin embargo, a pesar de ser una aplicación bastante completa y estar conformada por una aplicación Web, y por una aplicación móvil, tiene ciertas desventajas; la principal, es que es un software privado que solo proporciona 14 días de prueba de manera gratuita, y si bien cuenta, tanto con una Web, como con una móvil, no posee ambos módulos integrados, es decir, la aplicación Web solo apoya a los médicos nutriólogos, y la móvil a los pacientes, según (Nutrium, s.f.).

Seguidamente, se habla de Nutrimind, la cual es una aplicación Web, móvil y de escritorio, que ofrece distintas herramientas para analizar la composición corporal del paciente basado en las mediciones físicas del mismo. Además, ayuda a calcular valores como el IMC (Índice de Masa Corporal), el índice cintura-cadera, pesos teóricos (manejados en kilogramos), entre otros; mostrando todos los resultados, a través de gráficos de torta y de barra, según (Nutrimind, s.f.).

Y por último, existe una aplicación desarrollada en Venezuela, llamada SysNutr, que a diferencia de las ya mencionadas anteriormente, sólo apoya a los médicos nutriólogos, la cual les permite simplificar su trabajo llevando a cabo el control del estado nutricional de todos sus pacientes, y facilitar la labor manual necesaria para calcular todos los valores requeridos, para poder realizar planes dieto-terapéuticos acordes a los requerimientos de los mismos, plantea (Travieso & Benes, 2017).

De acuerdo a lo antes mencionado sobre dichas aplicaciones, si bien es verdad que son de mucha utilidad para los médicos nutriólogos, haciendo su trabajo mucho más fácil, se considera que para los pacientes, las aplicaciones carecen de formas de estimulación y apoyo, para que los mismos logren mantener los regímenes que a

veces son muy estrictos, lo que causa a menudo, mucha deserción o abandono de los planes nutricionales planteados por los nutricionistas, ya que el paciente no cuenta con herramientas gratuitas que posean toda la información acerca de su tratamiento, y se dificulta la comunicación entre paciente y el nutricionista.

Por lo tanto, dadas las limitaciones de dichas aplicaciones, surge la necesidad de incorporar a la aplicación Web ya desarrollada (SysNutr) que apoya a los médicos nutriólogos, un módulo de soporte para los pacientes, en el cual, los mismos tienen acceso a los avances logrados a lo largo de su tratamiento, indicando para ello, su peso y algunas medidas de su cuerpo, incluyendo también consultas sobre modificaciones de los planes dietéticos por diferentes razones, y toda la información que concierne a su tratamiento. Así mismo, además de realizar una ampliación a la aplicación Web, también se desarrolla una aplicación móvil que apoya a los nutricionistas, permitiéndoles visualizar información correspondiente a los pacientes, citas pendientes, menús, entre otros; de igual modo, proporciona apoyo a los pacientes en el seguimiento de su plan dieto-terapéutico. Y a su vez, en la aplicación Web/móvil se muestran indicadores, de forma gráfica, que son de interés para los médicos nutriólogos.

Las aplicaciones tienen ciertos aspectos técnicos a nivel de la plataforma en la que se encuentran todos los datos almacenados de la aplicación, y que se deben optimizar, cómo: mejorar la escalabilidad y capacidad, dado que, en un futuro la cantidad de datos puede crecer de manera abrupta, y ocasionar problemas de velocidad o acceso a los mismos; por lo que surge la necesidad de fortalecer la plataforma realizando la migración de un SMBD (Sistema Manejador de Base de datos), MySQL a PostgreSQL.

A continuación, dada la problemática planteada se plantea el objetivo general del TEG, y los objetivos específicos, de la siguiente manera:

1.2. Objetivos

1.2.1. General

Desarrollar una aplicación Web/móvil que apoye a los profesionales y pacientes en el área de Nutrición y Dietética, basada en la aplicación Web SysNutr.

1.2.2. Específicos

- Analizar los requerimientos funcionales y no funcionales de la aplicación Web/móvil a desarrollar.
- Diseñar las interfaces de la aplicación Web/móvil que permiten observar, de manera fácil e intuitiva, la información de los pacientes, y de sus planes dieto-terapéuticos.
- Definir y diseñar la arquitectura Web/móvil para la aplicación propuesta.
- Migrar la base de datos del SMBD MySQL, al SMBD PostgreSQL.
- Desarrollar e implementar las funcionalidades necesarias de la aplicación Web/móvil para el apoyo de los pacientes del área de Nutrición y Dietética.
- Definir, diseñar e implementar indicadores que permitan evaluar la gestión del médico nutriólogo.
- Realizar las validaciones y pruebas necesarias, para asegurar el correcto funcionamiento de la aplicación Web/móvil.

En el siguiente punto se describe la arquitectura utilizada para la solución del TEG.

1.3. Arquitectura de la solución

La arquitectura seleccionada es de tipo Cliente/Servidor, ya que dicha arquitectura es la utilizada para el desarrollo de la aplicación Web SysNutr. Por lo tanto, un usuario (el cliente) a través de un navegador Web o dispositivo móvil, es el responsable de iniciar la comunicación con un servidor de aplicaciones, a través de internet, posiblemente para realizar alguna petición de una aplicación Web/móvil alojada en el mencionado.

Luego, el servidor responde la petición, las envía al navegador o dispositivo móvil, y es mostrada al usuario.

En la Figura 1.1, se refleja de manera visual, la arquitectura de la solución propuesta.



Figura 1.1. Arquitectura de la solución.

Para el desarrollo del módulo de pacientes del área de Nutrición y Dietética, que es incorporado en la aplicación Web ya desarrollada (SysNutr), y que apoya a los profesionales en dicha área, se continúa con el uso de las mismas tecnologías con las que fue implementada dicha aplicación, para agilizar el tiempo de desarrollo, ya que de ser con otras tecnologías, se tendría que invertir tiempo en la integración entre ambas. Dichas tecnologías son: el lenguaje de programación PHP, junto con el marco de trabajo Laravel, dado que son fáciles de utilizar, y tienen una curva de aprendizaje muy corta; PHP es uno de los lenguajes más utilizados en el desarrollo Web, y Laravel es uno de los marcos de trabajo de PHP más utilizados, por lo que cuenta con una amplia comunidad de desarrolladores colaborativos, y una muy buena documentación; además PHP es multiplataforma y compatible con diversas bases de datos; y por último, Laravel provee una estructura bastante cómoda para la organización del proyecto, y ofrece una serie de herramientas que facilitan el desarrollo de aplicaciones Web, como *Eloquent ORM (Object Relationship Management)*, para el manejo de bases de datos, y para el manejo de migraciones, se utilizaría *Composer* (manejador de dependencias), entre otras.

En cuanto al desarrollo de la aplicación móvil, se utiliza un lenguaje de programación que es un superconjunto de JavaScript, llamado TypeScript junto con el marco de trabajo IONIC, ya que dicho *framework*, por ser compatible con Angular, es de las mejores elecciones, debido a la gran cantidad de componentes que tiene; además en IONIC se puede utilizar el lenguaje HTML (*HyperText Markup Language*) y hojas de estilo CSS (*Cascading Style Sheets*), lo que facilita el aprendizaje y el desarrollo con dichas tecnologías, y posee mucha más documentación que cualquier otro *framework* de desarrollo para aplicaciones híbridas.

Se utiliza una base de datos, para almacenar toda la información de las aplicaciones, y cuyo diseño, se realiza utilizando el Sistema Manejador de Base de Datos (SMBD) PostgreSQL, dicho SMBD, soporta un gran número de peticiones simultáneas, tiene una alta capacidad de almacenamiento, mantiene su velocidad de respuesta al aumentar el tamaño de la base de datos, y ofrece estabilidad y confiabilidad. La aplicación en su totalidad, se aloja en el Servidor Heroku, el cual es fácil de configurar.

El diseño de las vistas de las aplicaciones, son estructuradas con HTML; para mejorar la presentación o Look and Feel de la aplicación, se utiliza CSS, y para definir el comportamiento de la página, se utiliza JavaScript, tecnologías que se ejecutan e interpretan en el lado del cliente, sin realizar peticiones al servidor.

Posteriormente, se describe a continuación, la justificación de la solución lograda para el desarrollo del TEG.

1.4. Justificación de la solución

Las aplicaciones Web/móvil poseen muchas ventajas, entre ellas, se destaca permitir el acceso inmediato, desde cualquier lugar, y que múltiples usuarios puedan acceder a las funcionalidades de la aplicación, y a la información almacenada en las bases de datos, evitando el traslado y el uso innecesario de herramientas como: hojas de papel, calculadoras y lápices. El poder contar con aplicaciones que permitan a los nutricionistas, gestionar procesos, como la creación de dietas a sus pacientes y a su vez, permitirles llevar un seguimiento continuo de su tratamiento, ya que dichas aplicaciones no solo facilitan el trabajo de los nutricionistas, sino también las actividades realizadas por sus pacientes, al permitir un contacto más fluido y constante, entre ambas partes; además de ahorrar tiempo a los nutricionistas, al realizar diagnósticos y entregas de dietas a cada uno de los pacientes de acuerdo a sus necesidades; además dicha aplicación, puede utilizarse fuera del consultorio, y evitar la necesidad de tener que trasladar carpetas y archivos con dicha información; asimismo, puede ser utilizada para realizar una evaluación nutricional en grandes comunidades donde se atiendan más de 50 pacientes por día, haciendo más rápido y efectivo el trabajo del nutricionista. Por último, dichas aplicaciones poseen una base de datos, que permite una mayor velocidad de procesamiento y acceso a los datos.

CAPÍTULO II: MARCO CONCEPTUAL

Conocer los antecedentes y las bases conceptuales de un trabajo de investigación, es de vital importancia, ya que permite tener claro cómo abordar la problemática planteada. En el siguiente capítulo, se trata con detalle, el estado actual de diversas aplicaciones que son destinadas a apoyar a los médicos y pacientes del área de nutrición, y conceptos relacionados con la misma, asimismo se abordan conceptos que corresponden con herramientas y tecnologías de desarrollo de aplicaciones Web/móvil, tales como: Arquitectura, funcionamiento, tecnologías utilizadas y sistemas manejadores de base de datos.

Luego de conocer los antecedentes de dicha investigación, es necesario aclarar que los seres humanos, para llevar un control de su estado nutricional, deben acudir a un médico nutriólogo, el cual les realiza una serie de estudios para determinar su estado físico y nutricional, y a partir de los resultados obtenidos, fijar un plan dieto-terapéutico personalizado que debe llevar a cabo, para lograr tener una vida más saludable.

A continuación, se detallan los antecedentes que se utilizan como referencia para el desarrollo de la aplicación Web/móvil.

2.1. Antecedentes

Hoy en día, ciertos procesos tales como: la realización de un plan dieto-terapéutico para el paciente, y lograr un seguimiento del mismo y mantener el contacto con el médico nutriólogo, se siguen realizando de manera manual. Dichos procesos pueden volverse tediosos, además de que según la cantidad de pacientes que son atendidos, se puede generar una gran cantidad de carpetas con archivos por cada uno de ellos, aumentando así la probabilidad de pérdida y difícil acceso de los mencionados. Por otro lado, pueden ocasionar deserción de los pacientes, al no tener un seguimiento constante del tratamiento y contacto continuo con el nutricionista.

A continuación, se describe detalladamente, en qué consisten las aplicaciones utilizadas actualmente para apoyar al médico nutriólogo y al paciente, durante el proceso de evaluación nutricional que son mencionadas anteriormente y que son analizadas durante el trabajo de investigación.

2.1.1. NUTRIUM

Nutrium es un software de nutrición que consta de una aplicación Web y móvil, originaria de Braga - Portugal, que es utilizado para apoyar a los profesionales en el área de nutrición, simplificando sus tareas y ayudando a los pacientes a lograr sus objetivos. Dicho software, ofrece una serie de funcionalidades para facilitar el acompañamiento del plan de alimentación del paciente, como, por ejemplo, permite la consulta online del progreso y del cumplimiento de los objetivos, como mejorar sus hábitos alimenticios, subir o bajar de peso, entre otros; logrando dichos objetivos en tiempo real, además simplifica la creación de planes de alimentación realizando los cálculos nutricionales de forma automática, y los cuales pueden ser modificados y enviados por el nutricionista en formato PDF al paciente. La aplicación provee funciones importantes, como la programación y gestión de consultas, en la cual, el paciente y el nutricionista, pueden establecer citas; por otro lado, incluye notificaciones durante las comidas, el registro de datos personales, clínicos y alimentarios, así como también registra y analiza los datos antropométricos del paciente, entre otras funcionalidades, según (Nutrium, s.f.).

Si bien todos los aspectos que abarca el software mencionado, de lo expuesto anteriormente, se puede observar que es una plataforma suficientemente extensa que cuenta con una gama bastante amplia de funcionalidades, las cuales permiten simplificar el trabajo, tanto del nutricionista como para el paciente. Lamentablemente tiene ciertos aspectos negativos, uno de los principales, es que es un software privado, es decir, se necesita pagar para poder disfrutar de él, solo provee un periodo 14 días de prueba, y a su vez se necesita, de conexión a internet para el uso de la aplicación.

A continuación, en las figuras 2.1, 2.2, 2.3, 2.4, 2.5, 2.6 y 2.7; se muestran algunas de las principales interfaces, tanto de la aplicación Web, como la móvil del software:

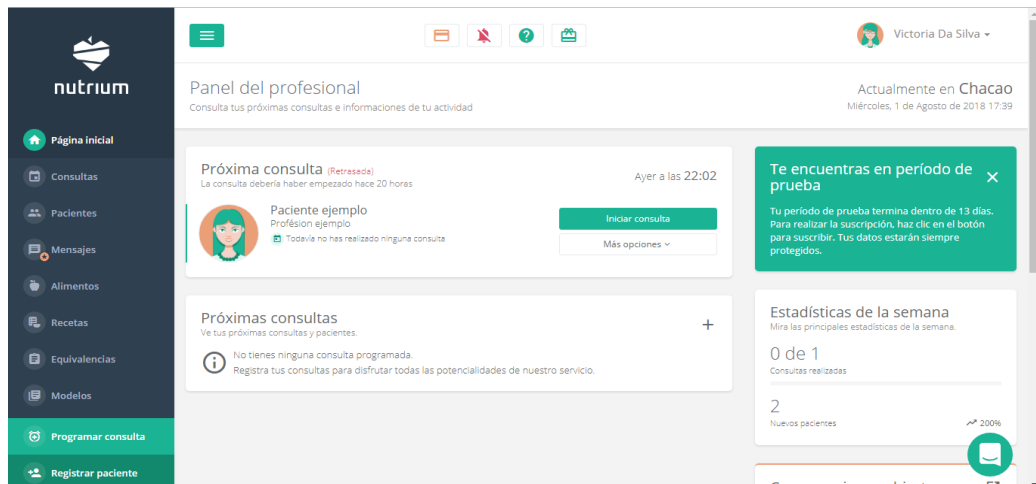


Figura 2.1 - Página principal de Nutrium. Fuente (Nutrium, s.f.).

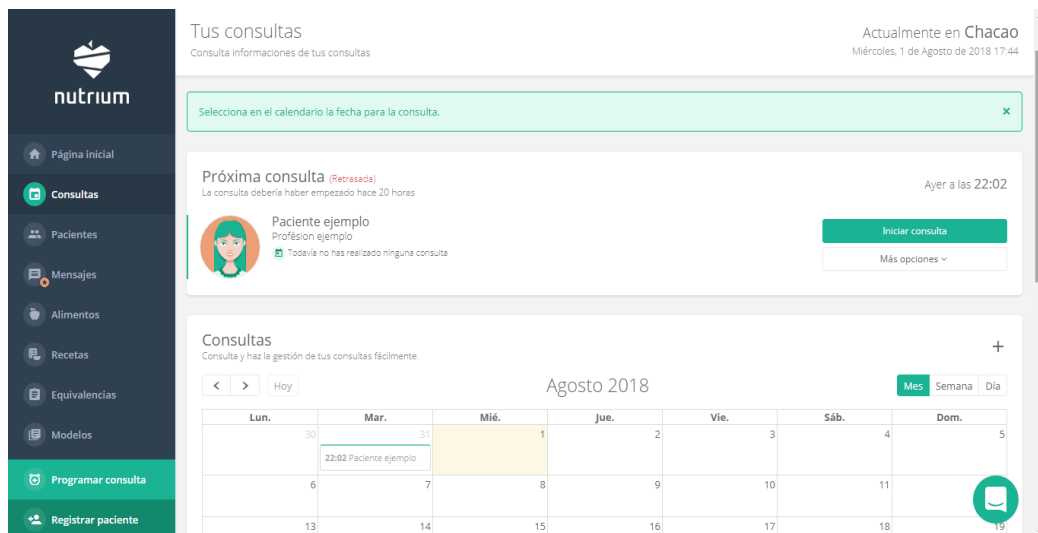


Figura 2.2 - Gestión y programación de citas. Fuente (Nutrium, s.f.).

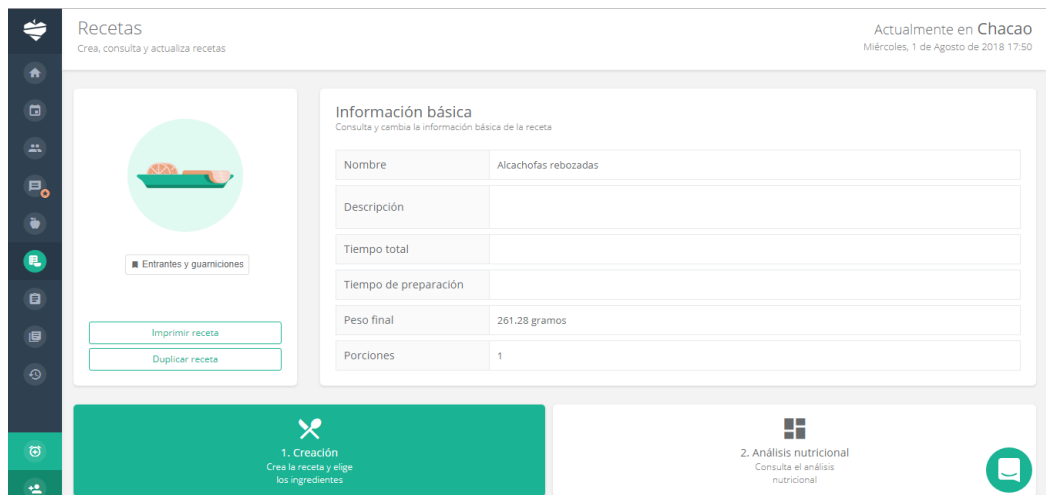


Figura 2.3 - Creación y modificación de receta. Fuente (Nutrium, s.f.).

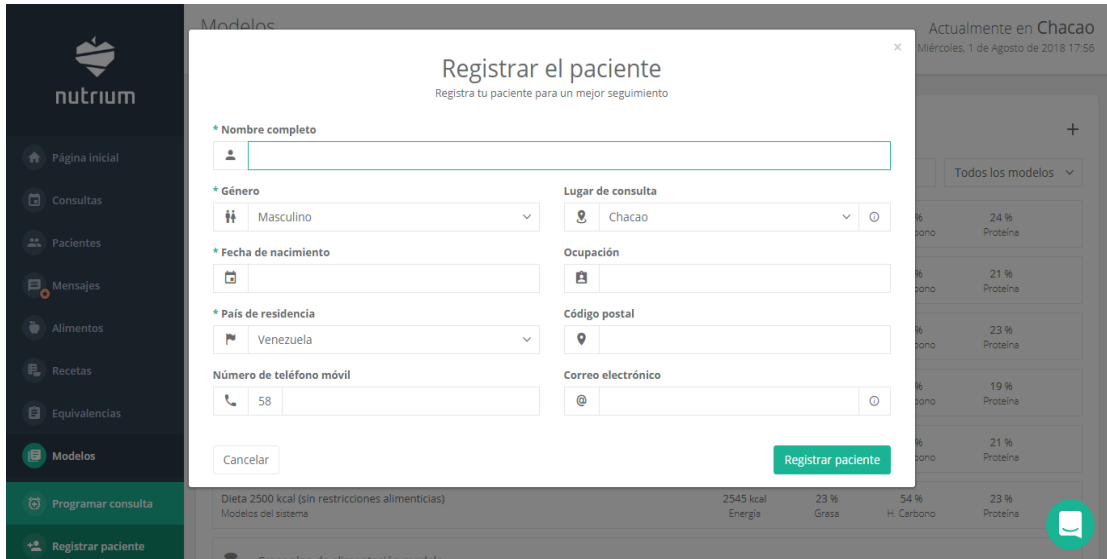


Figura 2.4 - Registro de pacientes. Fuente (Nutrium, s.f.).

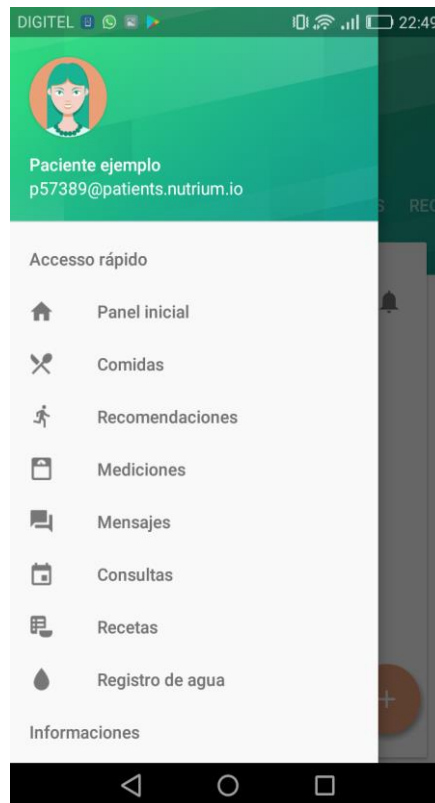


Figura 2.5 - Menú principal. Fuente (Nutrium, s.f.).

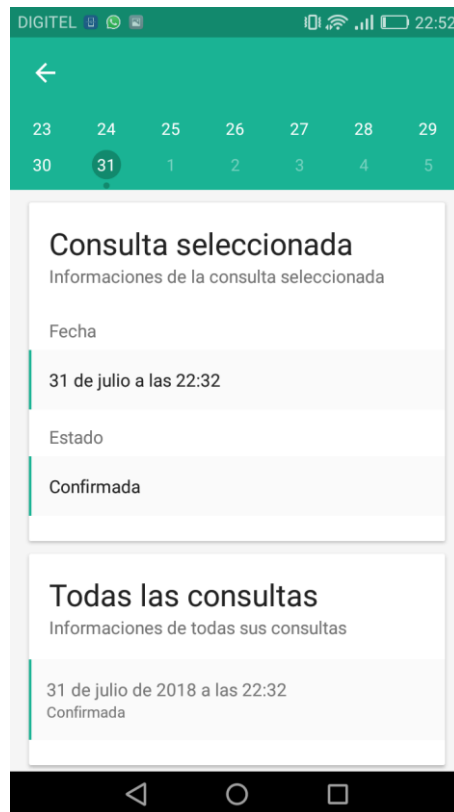


Figura 2.6 - Gestión y Programación de consultas. Fuente (Nutrium, s.f.).

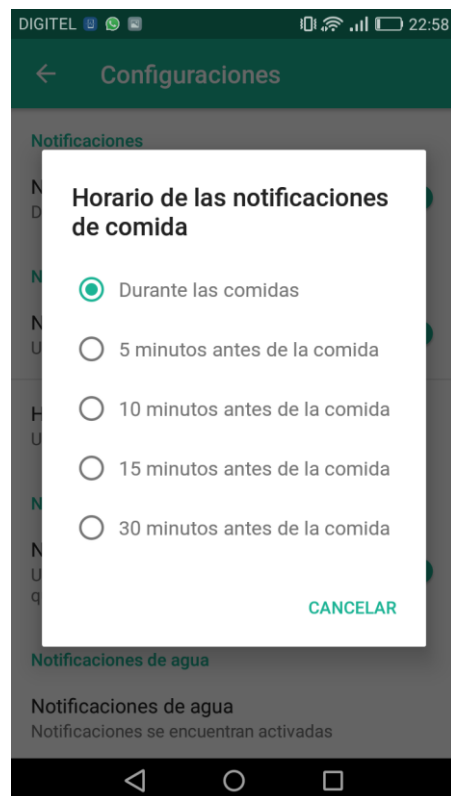


Figura 2.7 - Gestión de notificaciones durante las comidas. Fuente (Nutrium, s.f.).

2.1.2. NUTRIMIND

Nutrimind es una aplicación tanto Web como de escritorio, creada en el año 2007, y recientemente se incluye una aplicación móvil en el año 2017. Nutrimind es originaria de México, se comercializa con gran éxito en toda América Latina y España, y la cual, está diseñada para apoyar al nutricionista durante la creación de dietas y el control de pacientes, y al paciente, en el seguimiento de su tratamiento, según (Nutrimind, s.f.).

La herramienta permite tratar a todo tipo de pacientes, pero está enfocada en los pacientes de cultura occidental (debido a su tipo de alimentación), y a su vez, para complementar la creación y planificación de dietas, la aplicación Web realiza todos los cálculos necesarios para elaborar el menú del paciente; además ofrece preparaciones alternativas que cumplan con los requerimientos que establece la dieta creada. Así mismo, proporciona funcionalidades sencillas, como la inclusión de una foto de perfil de cada paciente para su reconocimiento, instrucciones que indican en donde se realiza cada medición según los estándares internacionales, la generación de gráficos de evolución de todas las mediciones exactas del cuerpo, signos vitales, calorías en la dieta, generación de cuadros de equivalencia, entre otras. Y por otro lado, la aplicación móvil permite realizar funcionalidades, como el ofrecer a los pacientes, el llevar un calendario de sus citas, visualizar su evolución y los menús de acuerdo a su dieta establecida, además permite registrar diariamente la actividad física del paciente y su estado de ánimo, con la finalidad, de lograr un aproximado más cercano de su evolución en el tratamiento, entre otras funcionalidades, plantea (Nutrimind, s.f.).

De lo expuesto, se puede decir que efectivamente es una herramienta con muchas funcionalidades, y que les permite a los profesionales del área de nutrición, y a los pacientes, el realizar sus actividades de una manera más cómoda y rápida. Sin embargo, para su uso se debe adquirir una licencia, y en la versión de escritorio solo se permite su instalación en dos computadores, por lo que instalarlo en un centro médico, que disponga de una buena conexión a internet, el costo de la licencia (en dólares) puede constituir una inversión bastante considerable, dependiendo de la cantidad de computadoras en las que se desee utilizar dicha aplicación. Así mismo, otra desventaja que posee, es que el software no es de código abierto (OpenSource), por lo cual, si se desea adaptar a una necesidad específica, no se puede hacerlo, lo

cual limita su uso, a las funcionalidades mencionadas inicialmente. Adicionalmente, si bien es cierto que dicho software posee muchas funcionalidades; la aplicación Web no cuenta con herramientas que proporcionen ayuda para el paciente, y por el contrario, la aplicación móvil no posee herramientas que apoyen al médico nutriólogo, sino, únicamente al paciente.

En la Figura 2.8 se muestra el menú principal de la aplicación Web.

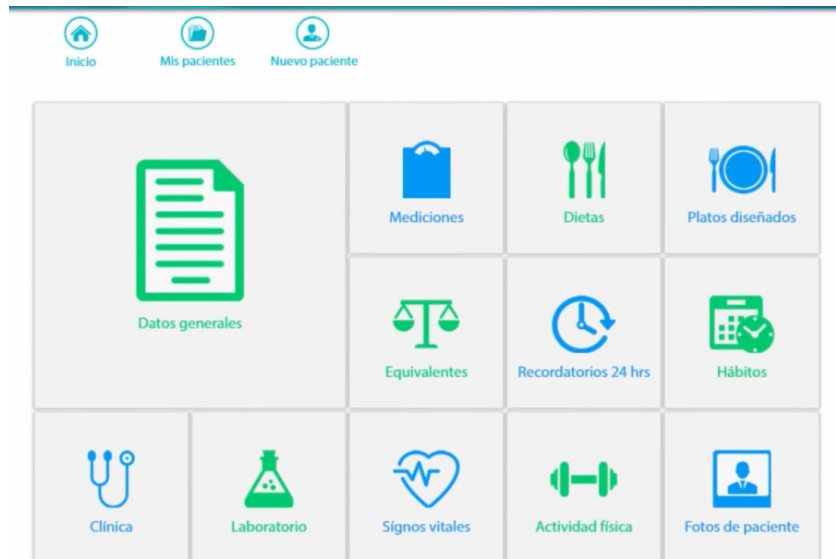


Figura 2.8 Página Principal. Fuente (Nutrimind, s.f.).

En la Figura 2.9 se puede observar cómo crear un menú de alimentos.

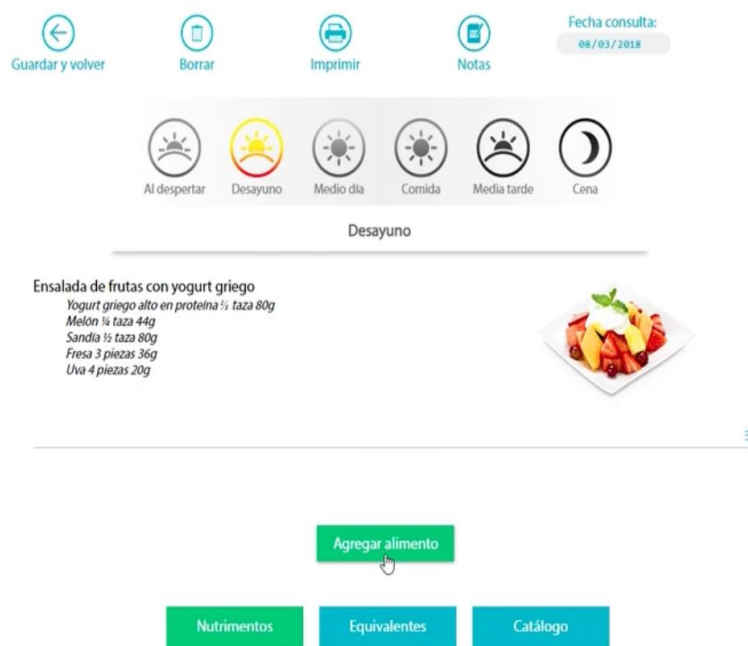


Figura 2.9 - Creación de menú. Fuente (Nutrimind, s.f.).

En la Figura 2.10 se percibe la generación de un cuadro de equivalencia de un paciente.



Figura 2.10 - Generación de cuadro de equivalencia. Fuente (Nutrimind, s.f.).

En la Figura 2.11 se presenta el cálculo calórico por actividad.

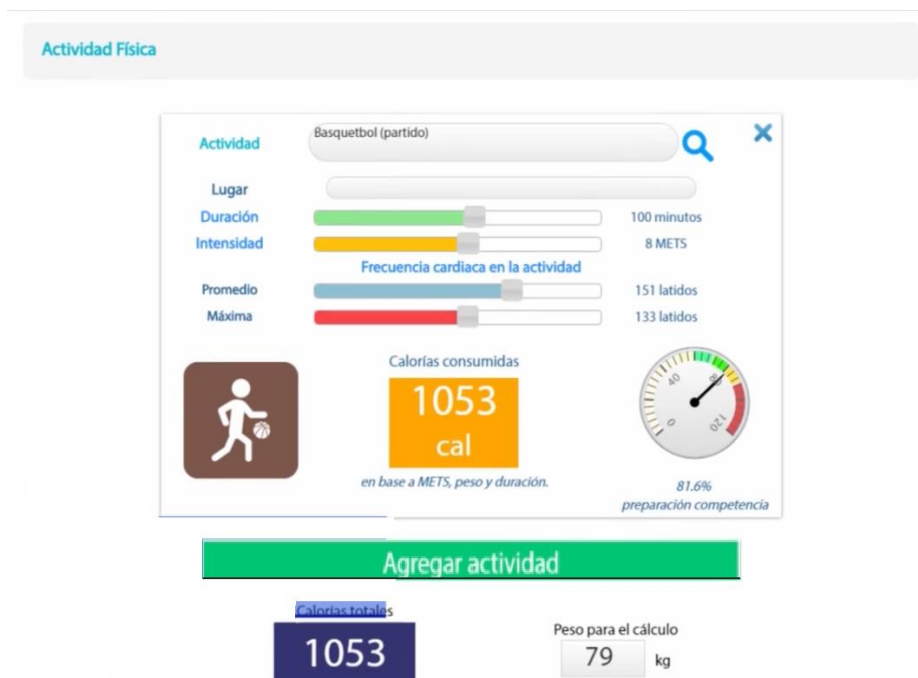


Figura 2.11 - Cálculo Calórico. Fuente (Nutrimind, s.f.).

En la Figura 2.12 se observa la página principal de la aplicación móvil.

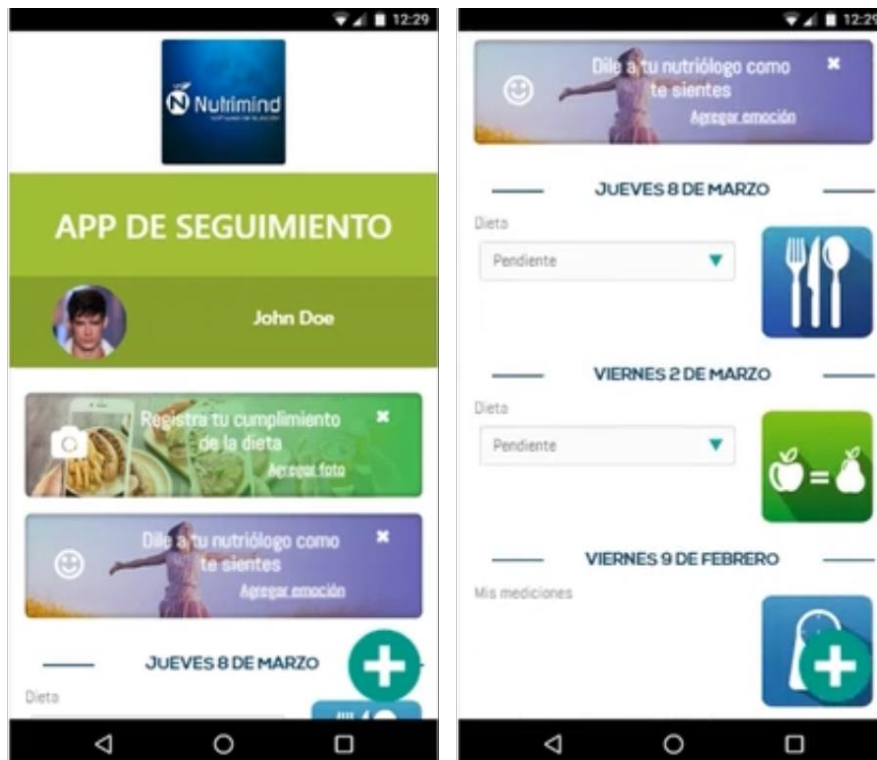


Figura 2.12 - Pantalla principal. Fuente (Nutrimind, s.f.).

En la Figura 2.13 se puede apreciar la lista de menú del día del paciente.

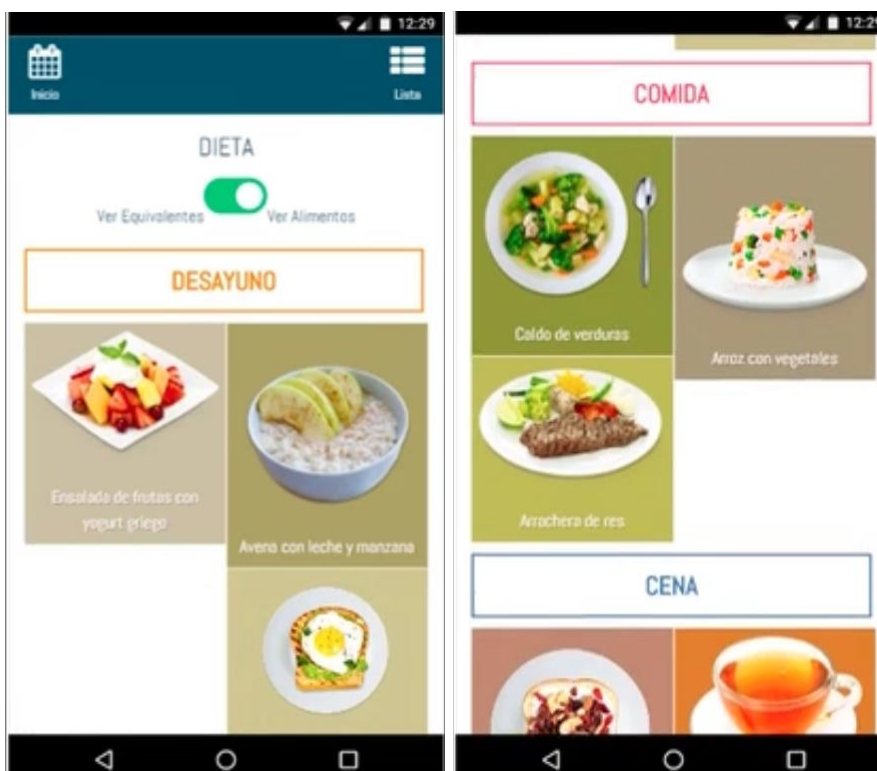


Figura 2.13 - Menú del día. Fuente (Nutrimind, s.f.).

2.1.3. SYSNUTR

SysNutr es una aplicación Web desarrollada en Venezuela, por los licenciados Frank Travieso y Mariano Benes como TEG en la Facultad de Ciencias, de la Universidad Central de Venezuela, la cual está diseñada para apoyar a los profesionales en el área de nutrición, permitiendo registrar y gestionar los datos del paciente, el realizar cálculos automatizados de requerimientos nutricionales de cada uno de ellos, de acuerdo a su sexo, edad, contextura y condición patológica; kilocalorías que debe ingerir durante el día, distribuida en cada tiempo de comida (desayuno, almuerzo, cena y meriendas); crear, gestionar y modificar menús personalizados que cumplan con las necesidades de cada paciente, y de esa manera se simplifica el trabajo requerido por parte de los nutricionistas para realizar sus labores, según (Travieso & Benes, 2017).

Dadas las funcionalidades ya mencionadas de dicha aplicación, se puede observar que, si bien proporciona gran apoyo para el médico nutriólogo, no ofrece funcionalidades que apoyen a los pacientes del mismo.

A continuación, se presentan algunas de las principales interfaces de la aplicación Web SysNutr (Consultar figuras 2.14, 2.15, 2.16, 2.17, y 2.18).

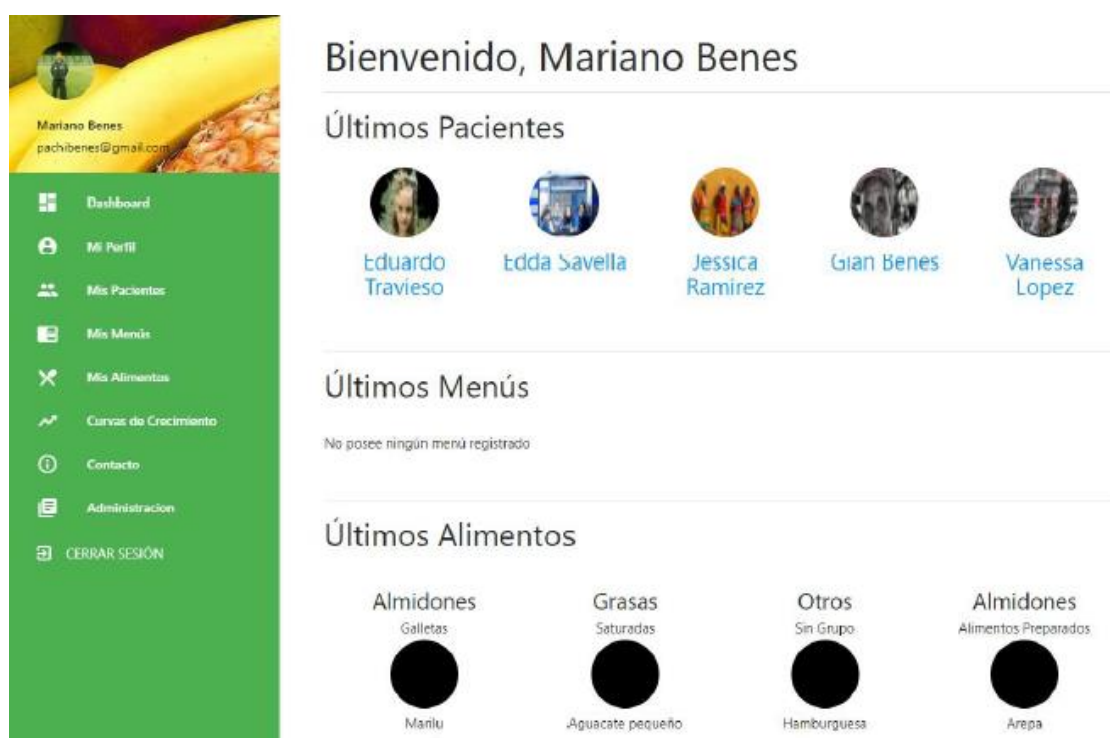


Figura 2.14 - Dashboard. Fuente (Travieso & Benes, 2017).

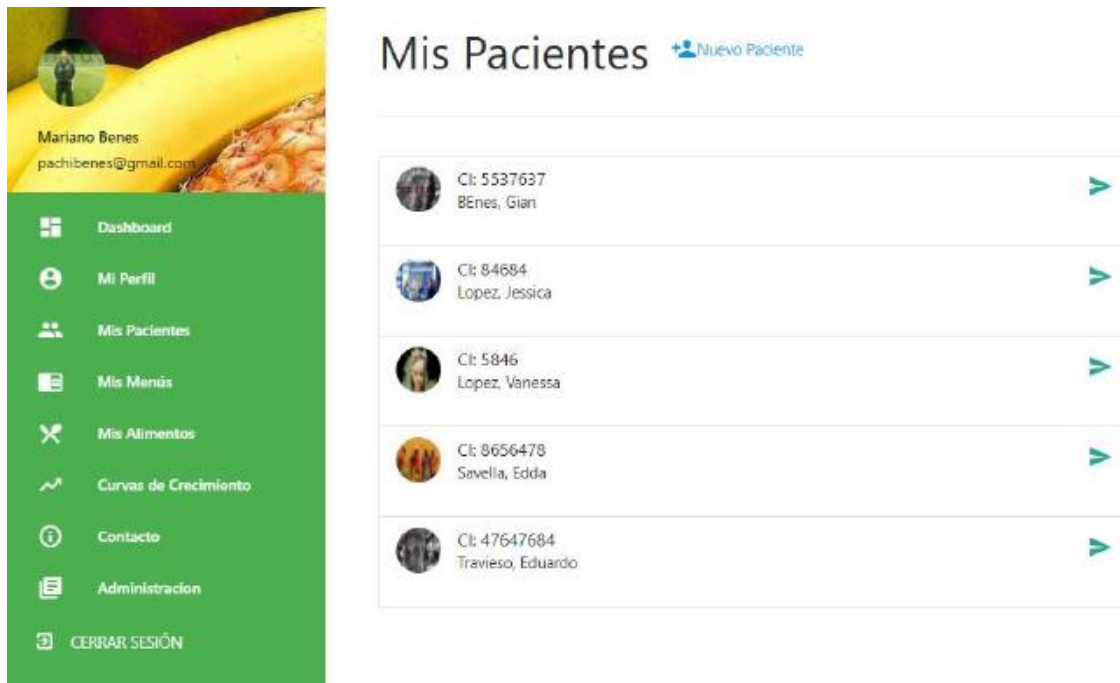


Figura 2.15 - Mis Pacientes. Fuente (Travieso & Benes, 2017).

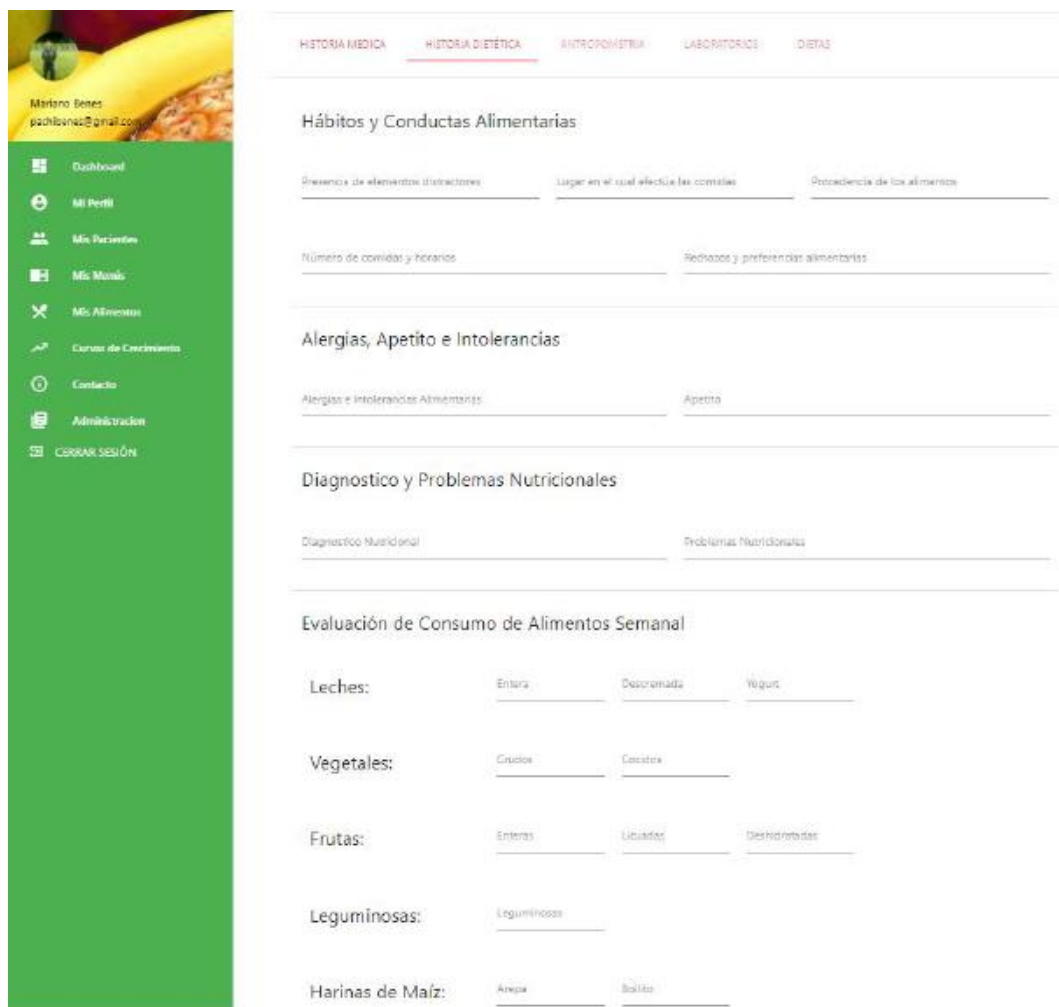


Figura 2.16 - Historia Dietética. Fuente (Travieso & Benes, 2017).



Figura 2.17 - Dietas. Fuente (Travieso & Benes, 2017).

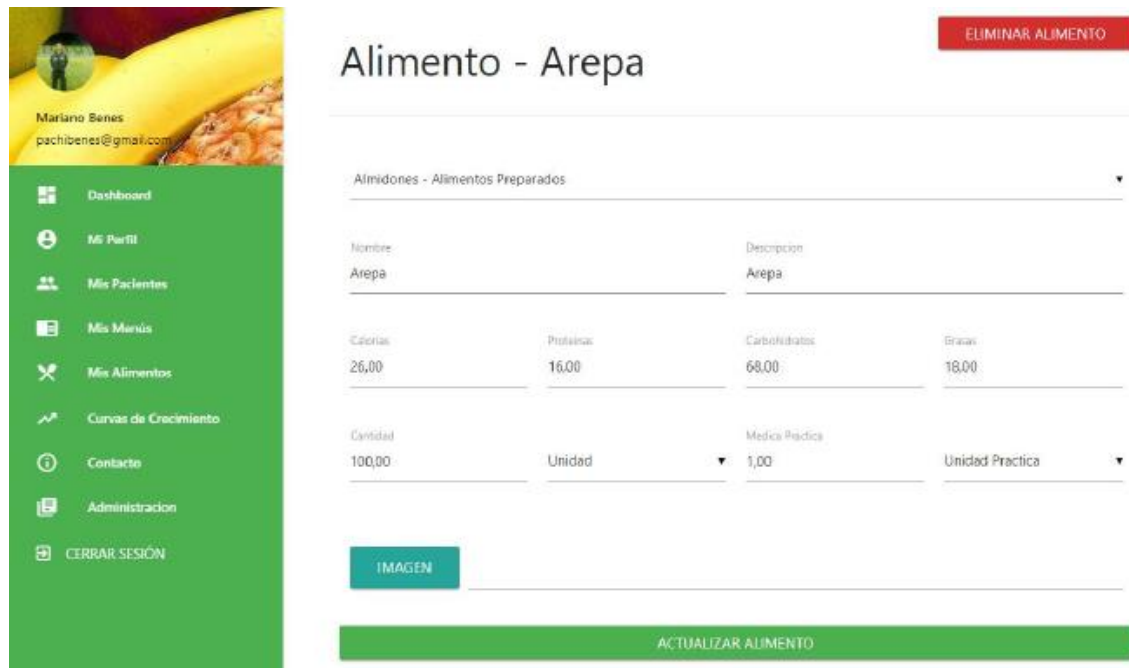


Figura 2.18 - Registro de Alimentos. Fuente (Travieso & Benes, 2017).

En el siguiente punto se describen los conceptos más relevantes para comprender un poco más sobre el área de Nutrición.

2.2. Conceptos básicos relacionados al área de nutrición

Nutrición

Para lograr una nutrición adecuada, existen ciertos procesos que suceden en el cuerpo después de comer, como la obtención, asimilación y digestión de los

nutrientes por el organismo, y que deben cumplir algunos objetivos importantes, como lo son: el proveer energía para el mantenimiento de sus funciones y actividades, suministrar materiales para la reproducción, la formación, el crecimiento y reparación de las estructuras corporales, disminuir el riesgo de algunas enfermedades, y proporcionar las sustancias necesarias para regular los procesos metabólicos, entre otros, plantea (Carbajal, 2013).

Por lo tanto, se puede definir la nutrición, como el conjunto de procesos por los cuales el hombre absorbe, ingiere, transforma y utiliza las sustancias que se encuentran en los alimentos, y que deben cumplir los objetivos anteriormente descritos, según (Carbajal, 2013). Se debe acotar, que el proceso nutritivo es inconsciente e involuntario, y el que resulte satisfactorio o no, solo depende de una elección acertada de los alimentos que se deben ingerir, según (Rodríguez, 2013).

Alimentación

Es un proceso a través del cual, se toman una serie de productos o sustancias del mundo exterior, que se encuentran contenidas en los alimentos que forman parte de nuestra dieta, y que son necesarias para la nutrición, según (Carbajal, 2013).

Alimento

Es toda aquella sustancia o producto, que proporciona materiales asimilables que llevan a cabo una función nutritiva para el organismo, luego de haberse consumido, plantea (Carbajal, 2013).

Plan dieto-terapéutico o Dieta

Es un régimen, que en circunstancias específicas realizan personas sanas, enfermas o convalecientes en el comer y beber. Además, el plan dieto-terapéutico o dieta, estudia una forma de proporcionar a cada persona o grupo de personas, según sus circunstancias y estado fisiológico, los alimentos necesarios para su adecuado desarrollo. Es decir, aplica e interpreta los principios y conocimientos científicos de la Nutrición, elaborando una dieta adecuada para el hombre, ajustada a las recomendaciones y objetivos nutricionales más actuales, y que además, tomando en cuenta también las posibilidades, gustos, costumbres y el contexto socio-cultural, para que sea lo más personalizada posible y cumpla su objetivo, el cual es, que sea aceptada y consumida, según (Carbajal, 2013).

Nutricionista

Es una persona capacitada y legalmente reconocida para poder proporcionar una buena alimentación, tanto para grupos de personas de cualquier edad o de individuos en buen estado de salud, como de los que necesitan un régimen terapéutico. Por tanto, un nutricionista conocido también como dietista o nutriólogo, es un profesional preparado para aconsejar a sus pacientes, acerca de nutrición. Dicho nutricionista puede ejercer sus funciones en el sector público y privado, o de manera autónoma. Algunos de sus ámbitos de especialización son: industria alimentaria, nutrición clínica, salud pública, restauración colectiva, investigación y docencia, plantea (GES Formación, 2016).

En vistas generales, de la investigación realizada, se puede observar que se han desarrollado dos aplicaciones que tienen el propósito de cubrir las necesidades, tanto del nutricionista como del paciente, llamadas Nutrium y Nutrimind, y de las cuales no se puede obtener mucha información, debido a que ambas herramientas solicitan una remuneración, y son totalmente privativas. Por otro lado, como se mencionó anteriormente, en Venezuela existe el desarrollo de una aplicación Web llamada SysNutr, que si bien apoya a los nutricionistas en el seguimiento de la evolución de sus pacientes; al crear y desarrollar planes dieto-terapéuticos personalizados, no ayuda al paciente en el seguimiento continuo de su tratamiento. En cuanto a aplicaciones móviles, en el país no se ha desarrollado ninguna aplicación que apoye, ni a médicos, ni a los pacientes del área de nutrición.

A continuación, se elabora un análisis que emplea una variedad de conceptos relacionados al desarrollo de una aplicación Web/móvil.

2.3. Arquitectura cliente-servidor

El esquema Cliente/Servidor, se representa como una arquitectura distribuida que permite a los usuarios finales, obtener acceso a la información en forma transparente, aún en entornos multiplataforma. Usualmente la mayoría del trabajo pesado, se hace en el proceso llamado servidor, y el proceso cliente sólo se ocupa de la interacción con el usuario, según (Reenskaug & Coplien, 2009).

El esquema Cliente/Servidor, como su nombre lo sugiere, está representado por dos elementos:

Las Aplicaciones Cliente, quiénes normalmente, manejan todas las funciones relacionadas con la manipulación y el despliegue de datos, por lo que se encuentran desarrolladas sobre plataformas que permiten construir interfaces gráficas de usuario (GUI), además de acceder a los servicios de cómputo distribuidos, en cualquier parte de una red. Por otro lado, las *Aplicaciones de los Servidores*, que proporcionan uno o más servicios al cliente y devuelven los resultados, según (Reenskaug & Coplien, 2009).

El funcionamiento de dicha arquitectura se enfoca en que un cliente realiza una petición al servidor, a través de una red de comunicación común. Una vez que el servidor recibe la petición, procede a su procesamiento y devuelve una respuesta a través de la red, para que posteriormente sea visualizada por el usuario final, plantea (Hernández & Martínez, 1997).

En algunos casos, existen procesos auxiliares que se encargan de recibir las solicitudes del cliente, procesar requerimientos de base de datos, procesar la lógica de la aplicación, y realizar validaciones a nivel de las bases de datos, recibir su respuesta, y enviarla al cliente.

Uno de los aspectos que más ha promovido el uso de sistemas Cliente/Servidor, es la existencia de plataformas de hardware cada vez más económicas, lo que constituye a su vez, una de las más palpables ventajas del esquema, es decir, la posibilidad de utilizar máquinas considerablemente más económicas, que las requeridas por una solución centralizada basada en sistemas grandes.

Además, se pueden utilizar componentes, tanto de hardware como de software, y de varios fabricantes, lo cual contribuye considerablemente a la reducción de costos, y favorece la flexibilidad en la implantación y actualización de soluciones, según (Reenskaug & Coplien, 2009).

A continuación, se proporciona un cuadro (Ver Tabla 1) con algunas de las ventajas y desventajas de la Arquitectura Cliente/Servidor, según (Ruiz, 2013):

Tabla 1: Ventajas e inconvenientes de la Arquitectura cliente/servidor

Ventajas	Inconvenientes
<p>Centralización de los recursos. Los recursos se administran en el servidor, y de esa manera, se evitan conflictos por inconsistencia o redundancia de la información en la base de datos.</p>	<p>Coste elevado. El mantenimiento y la instalación, proporcionan una elevada carga en cuanto al costo, debido a la complejidad técnica y funcional del servidor.</p>
<p>Mejora de la seguridad. Al establecer un mecanismo central de autenticación, se reduce la probabilidad de acceso indebido.</p>	<p>Dependencia del servidor. La red es completamente construida alrededor del servidor, por consecuencia si la misma deja de funcionar, o lo hace de manera incorrecta, afectará a toda la infraestructura.</p>
<p>Administración centrada en el servidor. La necesidad de administración por parte de los clientes es mínima.</p>	<p>Seguridad. Si el servidor donde está almacenada la información y la aplicación, no tienen los niveles de seguridad necesarios para resguardar los datos, pueden ser vulnerables a los ataques de agentes maliciosos</p>
<p>Escalabilidad de la instalación. Se puede agregar o eliminar clientes, sin que se afecte el funcionamiento de la red.</p>	<p>Cuello de botella. Se requieren de mecanismos adicionales, para prevenir cuellos de botella o congestión por parte del servidor.</p>

En la Figura 2.19, se puede observar la representación gráfica de la arquitectura cliente/servidor:

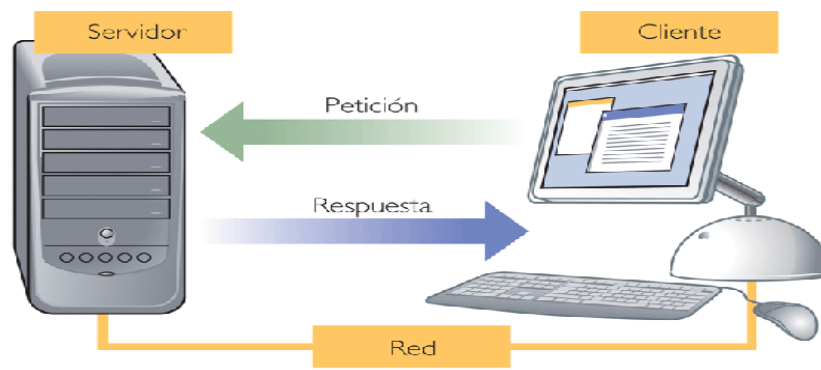


Figura 2.19 - Representación visual de la arquitectura cliente/servidor. Fuente (Ingsoftwareisc, 2016).

2.4. Aplicaciones Web

En la presente sección, se hace referencia a la definición, características, ventajas, desventajas, y a la arquitectura MVC (Modelo-Vista-Controlador), de la siguiente manera:

2.4.1. Definición

En la Ingeniería de Software, se llama aplicación Web, a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor Web, a través de internet o de una intranet, mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores Web, y en la que se confía la ejecución de la aplicación al navegador, plantea (Peñafiel, s.f.).

2.4.2. Características generales

Las aplicaciones Web constan de diferentes características, las cuales se describen a continuación, según (Peñafiel, s.f.):

- Actualmente las aplicaciones Web son populares, debido a que permiten una comunicación más fluida y dinámica en la computación Cliente-Servidor.
- El alto grado de desarrollo actual, permite la actualización y el mantenimiento (vía internet) de dichas aplicaciones, sin que se tenga que distribuir e instalar software específico o versiones individuales, a un usuario cada vez.
- Una página Web, puede contener elementos que permiten una comunicación activa, entre el usuario y la información (vía Servidor); logrando que el usuario tenga acceso a los datos de forma interactiva, gracias a que la página responderá a cada una de sus acciones.
- Las aplicaciones Web dinámicas, generan una serie de páginas en un formato estándar, como HTML (*HyperText Markup Language*) o XHTML (*eXtensible HyperText Markup Language*), soportada por los navegadores Web comunes.
- Se utilizan lenguajes interpretados en el lado del cliente, tales como JavaScript, para añadir elementos dinámicos a la interfaz de usuario.

- Generalmente cada página Web, se envía al cliente como un documento estático, pero la secuencia de páginas ofrece al usuario, una experiencia interactiva.
- Las aplicaciones Web para internet e intranet, presentan una serie de ventajas y beneficios con respecto al software de escritorio, para aprovechar y acoplar los recursos de una organización, de una forma mucho más práctica que el software tradicional.
- Por medio de las tecnologías actuales en el lado del cliente, se pueden crear aplicaciones Web responsive, es decir, de aplicaciones que también son compatibles con diversos dispositivos móviles.

En una aplicación Web se suelen distinguir tres niveles, como se puede observar en la Figura 2.20:

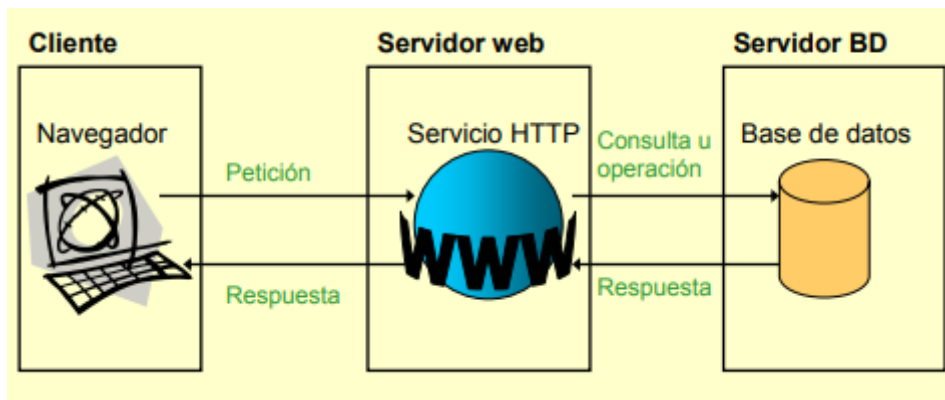


Figura 2.20 - Arquitectura Web de tres niveles. Fuente (Quispe, 2016).

- **El nivel de presentación** compuesto por las páginas HTML que el usuario solicita a un servidor Web, y que se visualizan en un cliente Web (habitualmente un navegador de internet).
- **El nivel de aplicación**, que contiene los módulos que implementan la lógica de la aplicación, y que se ejecutan en un servidor de aplicaciones que procesa las solicitudes realizadas por el cliente, en un servidor Web o de aplicaciones, y utiliza una tecnología Web dinámica, como: Java Servlets, ASP.NET, CGI, PHP, Ruby on Rails, entre otras, para procesar y enviar una respuesta, a dichas solicitudes.
- Y **el nivel de datos**, en donde se almacenan los datos de la aplicación, y que normalmente son gestionados por un sistema de gestión de bases de datos.

Se pueden observar a continuación algunas de las ventajas y desventajas al utilizar aplicaciones Web.

2.4.3. Ventajas de las aplicaciones Web

(Peñañiel Mayra, s.f.), plantea, que entre las ventajas más resaltantes de las aplicaciones Web, destacan las siguientes:

- **Compatibilidad multiplataforma:** Para las aplicaciones basadas en Web, es mucho más sencillo la compatibilidad multiplataforma, que las aplicaciones de software descargables, ya que no se necesita de un sistema operativo específico, debido a que las aplicaciones son utilizadas a través de un navegador Web.
- **Inmediatez de acceso:** Las aplicaciones basadas en Web, no necesitan ser descargadas, instaladas y configuradas, debido a que se encuentran alojadas en servidores remotos. Se pueden acceder vía online, y además se encuentran preparadas para trabajar, sin importar cuál es su configuración o su hardware.
- **Menos requerimientos de memoria:** Las aplicaciones tienen menos demandas de la memoria RAM por parte del usuario final, que de los programas instalados localmente, ya que todas las tareas y procedimientos se realizan en el servidor.
- **Menos Bugs:** Debido a que son aplicaciones basadas en Web, los bugs pueden ser corregidos al instante, de forma veraz y sencilla, siendo así, menos propensas a colgarse y a crear problemas técnicos, debido a problemas de software, o a conflictos de hardware con otras aplicaciones existentes, o con protocolos o software personal interno.
- **Múltiples usuarios concurrentes:** Es decir, son aplicaciones que pueden ser utilizadas por múltiples usuarios al mismo tiempo.

2.4.4. Desventajas de las aplicaciones Web

(Rivas, 2010) plantea, que a pesar de que las aplicaciones Web poseen una variedad de ventajas, como las mencionadas anteriormente, también existen algunas desventajas, como las descritas a continuación:

- **Dependencia a *plugins*:** Existen algunas aplicaciones Web que requieren de *plugins* específicos, para funcionar de manera óptima.

Para comprender mejor dicha desventaja, es necesario conocer el concepto de plugin, el cual, se define como un fragmento o componente de código, hecho para ampliar las funciones de un programa o de una herramienta, según (Tagle, 2014).

Ahora bien, el inconveniente es que constantemente dichos *plugins* deben ser actualizados, debido a que la aplicación se fía de eso, y de lo contrario es posible que la misma no funcione de forma adecuada.

- **Necesidad de Internet:** Para el uso de las aplicaciones Web, es imprescindible contar con conexión a Internet, lo cual puede ser considerado un problema, ya que en ocasiones no se cuenta con dicho recurso.
- **Los datos no están a la completa disposición:** Otro inconveniente es, que dado a que los datos se encuentran alojados en servidores remotos, es que si la aplicación Web no fue desarrollada, utilizando prácticas para el desarrollo de aplicaciones Web seguras, o se cae el servicio, los datos podrían quedar expuestos fácilmente.
- **Rendimiento:** Los múltiples accesos a la aplicación, que son realizados de manera simultánea, pueden ocasionar una sobrecarga en el servidor, y afectar el rendimiento de la aplicación.

A continuación, se describe la arquitectura MVC (Modelo Vista Controlador) utilizada para el desarrollo de aplicaciones Web.

2.4.5. Arquitectura MVC (Modelo Vista Controlador)

La arquitectura MVC, como un patrón de arquitectura de software, se basa en la separación de conceptos, es decir, separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos; y así mismo, se basa en la reutilización de código para facilitar el proceso de desarrollo de software. Dicha arquitectura, fue introducida como parte de la versión *Smalltalk-80* (*Smalltalk* es un lenguaje de programación). Su objetivo es reducir el esfuerzo de programación, en la implementación de sistemas múltiples. Tiene como principal característica, que el modelo, las vistas y los controladores, se tratan como entidades separadas; de

manera que, cualquier cambio que se produzca en el modelo, se refleje automáticamente en cada una de las vistas, plantea (Murazzo, 2010).

A continuación, se proporciona una descripción de los tres niveles en los que se basa dicha arquitectura, según (Cake Software Foundation, 2016):

- **Modelo:** El modelo representa la parte de la aplicación que implementa la lógica del negocio, es decir, es donde se trabaja con los datos, convirtiéndolos en conceptos significativos para la aplicación, con ayuda de mecanismos para almacenar y obtener información.
- **Vista:** La vista es responsable de realizar una interpretación de los datos provenientes del modelo, para producir una interfaz con la que el usuario puede interactuar.
- **Controlador:** Es el encargado de gestionar las peticiones de los usuarios. Le compete responder a las acciones solicitadas, para generar cambios en el modelo, o generar la vista apropiada; según la petición del usuario. Normalmente es identificado como el enlace entre la vista y el modelo.

2.4.5.1. Funcionamiento del patrón MVC

Si bien, pueden existir diversas implementaciones del patrón MVC, el flujo que generalmente se sigue, es el siguiente:

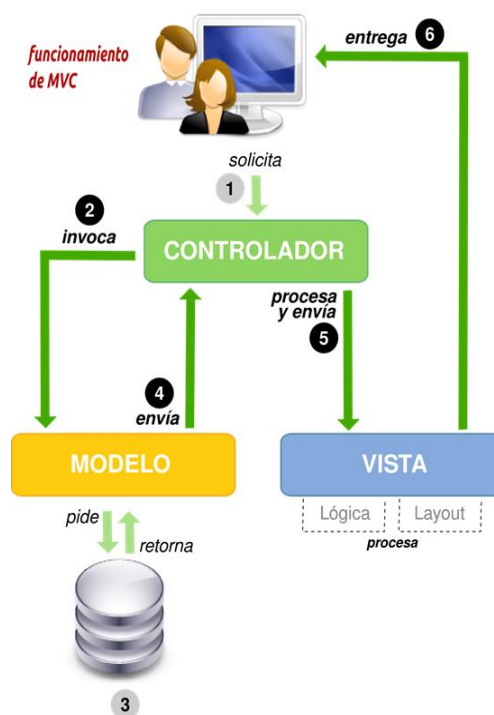


Figura 2.21 - Flujo del patrón MVC. Fuente (Undercode, 2013).

1. El usuario interactúa con la interfaz de alguna manera (por ejemplo, pulsa un botón, enlace, etc.). Dicha acción, genera una petición que es enviada al controlador.
2. El controlador recibe la petición. Si el controlador, ha verificado que la petición ha superado los filtros previos, gestiona el evento de entrada, e invoca al modelo para que sea atendida; de lo contrario, si la petición es rechazada, el controlador envía una respuesta de error a la vista.
3. El modelo resuelve la solicitud, que se puede tratar desde inserciones en bases de datos, modificaciones, consultas o algún requerimiento relacionado al proceso de negocio; y que no necesariamente está relacionado a un mecanismo de persistencia de datos.
4. El modelo, que es el encargado del almacenamiento de los datos, envía la respuesta al controlador, con la información solicitada por el usuario.
5. El controlador al recibir la respuesta del modelo, delega a los objetos de la vista, la tarea de desplegar la interfaz de usuario, y mostrar la respuesta obtenida, ante la petición que ha realizado el usuario.
6. La vista obtiene los datos del controlador, generando la interfaz apropiada para el usuario, aplicando estilos, organizando información, y construyendo la página que el usuario visualiza a través del navegador. Finalmente, la interfaz espera por nuevas interacciones de usuario para iniciar nuevamente el proceso mencionado.

Así como se explicó anteriormente, de que se tratan las aplicaciones Web, a su vez, se describe a continuación las aplicaciones móviles.

2.5. Aplicaciones móviles

En la presente sección, se hace referencia a la definición, y tipos de aplicaciones móviles, junto con sus ventajas y desventajas respectivamente, de la siguiente manera:

2.5.1. Definición

La aparición de las aplicaciones móviles no conforma un hecho reciente en la sociedad actual, de hecho, mucho antes de que aparecieran los teléfonos inteligentes,

ya se encontraban disponibles. El origen de las primeras aplicaciones móviles, fue a finales de los años 90. Dichas aplicaciones se dieron a conocer como aplicaciones de agenda, videojuegos como el snake, tonos de llamada, calendario, entre otros; las mismas, fueron implementadas mediante teléfonos celulares o móviles de segunda generación, cumpliendo funciones muy elementales, y con un diseño bastante simple. Desde entonces han ido evolucionando con los avances tecnológicos. Las aplicaciones móviles, o “apps”, en esencia no dejan de ser un software, incluso se puede decir que las aplicaciones son para los teléfonos, lo que los programas son para los ordenadores de escritorio, al menos así lo describe Javier Cuello, Especialista en Interacción Visual y Diseñador de productos digitales, según (Lutty, 2014).

Para David Gloag (Jefe de departamento de desarrollo de software del CV Diagnostix Inc, Ontario Canadá), una aplicación móvil consiste en un programa de software diseñado para dispositivos particulares, y que surgen como una respuesta a una necesidad del hombre de mantenerse conectado con los demás, de estar siempre informados, y de aumentar la productividad en las distintas áreas de trabajo.

De acuerdo a la naturaleza de las aplicaciones, se pueden clasificar en dos grupos: Híbridas y Nativas.

2.5.2. Aplicaciones móviles nativas

Las aplicaciones nativas, son aquellas desarrolladas bajo un lenguaje y entorno de desarrollo específico, de acuerdo al dispositivo móvil. Dependiendo de la plataforma para la cual se desea construir una aplicación, las herramientas de desarrollo pueden variar, tal es el caso de Java para Android, y Swift para sistemas operativos de iOS. Dicha característica, permite que una aplicación nativa tenga un funcionamiento más fluido y estable, para el sistema operativo que fue creada, plantea (Yañez, 2017), y adicionalmente, comenta, que desarrollar aplicaciones móviles nativas, tiene ciertas ventajas que el usuario puede notar en el uso de la misma, tales como las mencionadas a continuación:

2.5.2.1. Ventajas

- La velocidad de ejecución de dichas aplicaciones permite obtener el mejor rendimiento posible, comparado con cualquier otro tipo de app (aplicación móvil), debido a que las mismas se desarrollan en un lenguaje específico para cada plataforma, permitiendo a los desarrolladores acceder a las API's (interfaz de programa de aplicaciones); además de que el consumo de memoria y recursos, es menor.
- Como se mencionó anteriormente, dichas aplicaciones se desarrollan en un lenguaje específico para cada plataforma, lo cual ofrece el acceso a todas las funciones del terminal, permitiendo una mayor funcionalidad a la aplicación, logrando mejorar la experiencia de usuario, y obteniendo el máximo provecho de la terminal.
- Mayor escalabilidad de la aplicación para los programadores, en términos de tamaño, ya que no afecta el funcionamiento de la misma.
- Mayor visibilidad en las tiendas de Android y Apple, por lo que su posicionamiento para descargas es más elevado.
- Se actualizan de forma automática, cada vez que se publica una nueva versión de la misma.
- Son aplicaciones responsive, es decir que pueden adaptarse al tamaño de pantalla del dispositivo.

Pese a tener muchas ventajas, también existen varios inconvenientes a tener en cuenta, descritos a continuación.

2.5.2.2. Desventajas

- El tiempo y costo de desarrollo, es más elevado, ya que para publicar la app en la tienda de Android o de iOS, se debe desarrollar dos veces la misma app, debido a que cada desarrollo es diferente, y se necesitan desarrolladores que posean conocimientos y experiencia, en una o ambas plataformas.
- No se pueden tener dos aplicaciones idénticas para las dos plataformas (Android e iOS), pues cada una de ellas tiene sus funciones, y por lo tanto los desarrollos no pueden ser idénticos.

- Necesitan la aprobación de cada una de las tiendas, para ser publicadas.

2.5.3. Aplicaciones móviles híbridas

Previo a la definición de aplicaciones híbridas, es importante resaltar el concepto de framework, el cual, según (Gutiérrez, 2008), es una estructura de software, que se compone de librerías y componentes personalizables, para facilitar el desarrollo de una aplicación o componente de software.

Tomando en cuenta dicho concepto, las aplicaciones híbridas, son aquellas que combinan, aspectos de las aplicaciones nativas y de las aplicaciones Web, según más convenga. Las aplicaciones híbridas, se crean utilizando lenguajes de desarrollo Web, y un framework dedicado a la creación de aplicaciones híbridas, plantea (Yañez, 2017). Entre las ventajas que ofrece el desarrollo de dichas aplicaciones, se mencionan las siguientes:

2.5.3.1. Ventajas

- Permiten que, con un solo y único desarrollo, puedan ser utilizadas en distintas plataformas, por lo que, para diferentes sistemas como Android o iOS, no se deben realizar desarrollos separados.
- Se reduce el costo de desarrollo, cuando se realiza el lanzamiento de la misma aplicación para diferentes plataformas.
- Su mantenimiento es también, más sencillo, y permite la reutilización de código.
- Mayor rendimiento dentro del sistema operativo.
- El desarrollo se realiza en lenguajes mucho más sencillos, disminuyendo así, el nivel de dificultad en la construcción de la misma.

2.5.3.2. Desventajas

Pese a las ventajas que se han mencionado, también existen desventajas importantes, a considerar.

- El rendimiento de dichas aplicaciones es menor, ya que, en medidas generales, consumen más memoria RAM que una aplicación nativa.

- Debido a que dichas aplicaciones, no están desarrolladas en el lenguaje específico de cada plataforma, el acceso a las funcionalidades del dispositivo móvil, es limitado, y no permite sacarle el máximo rendimiento al mismo.
- Se requiere de una conexión a internet para poder ejecutarse, ya que se necesita de un navegador Web para poder ejecutarla.

Por otro lado, en el siguiente punto se describirán algunas de las herramientas tecnológicas utilizadas para el desarrollo de aplicaciones Web.

2.6. Herramientas tecnológicas para el desarrollo de aplicaciones Web

En dicho ítem, se hace referencia a las definiciones de las tecnologías del lado del cliente, tecnologías del lado del servidor, SMD, y las tecnologías o lenguajes de programación, que están relacionados a dichos tópicos, de la siguiente manera:

2.6.1. Tecnologías del lado del cliente

Las aplicaciones Web generan dinámicamente, una serie de páginas en un formato estándar, que es soportado por navegadores Web. Dichas páginas, son creadas utilizando lenguajes interpretados del lado del cliente, y que pueden ser directamente manejados por el navegador, y no necesitan un pretratamiento, tales como: HTML, CSS y JavaScript. Dichos lenguajes, o también conocidos como tecnologías, son de suma importancia para el buen funcionamiento de dichas aplicaciones.

A continuación, se mencionan algunas de las tecnologías que se utilizan del lado del cliente, para el desarrollo de una aplicación Web.

2.6.1.1. *HyperText Markup Language (HTML)*

HTML, traducido al español como “Lenguaje de Marcas de Hipertexto”, es un lenguaje de marcado, que permite desarrollar aplicaciones Web, estableciendo una estructura básica, y un código para la definición de contenido de una aplicación Web, como textos, imágenes, vídeos, entre otros; mediante el uso de etiquetas. Dicho lenguaje, posee muchas etiquetas con las cuales se pueden definir: Hipervínculos,

saltos de línea, estilos, scripts, tablas, imágenes, entre otros, plantean (Asensio & Menéndez, s.f.).

HTML es un estándar creado por la W3C (*World Wide Web Consortium*), la cual es una comunidad internacional que se dedica al desarrollo de estándares que aseguran el crecimiento de la Web a largo plazo, debido a que se relacionan con casi todas las tecnologías ligadas a la Web, sobre todo en lo referente a su escritura e interpretación. La W3C, desde el año 1996, se ha encargado de mejorar y publicar constantes revisiones de dicho lenguaje (HTML 3, HTML 3.0, HTML 4), ampliándolo y depurándolo con cada versión, plantean (Asensio & Menéndez, s.f.).

Actualmente se utiliza la versión cinco de HTML (HTML 5), la cual fue la última revisión aprobada en el año 2014. Dicha versión, proporciona características como: nuevos atributos para los formularios, la inclusión más sencilla de videos en las páginas Web, mejoras de contenido, etiquetado semántico, una forma diferente de crear la estructura básica de un sitio Web, entre otras.

La estructura básica de un sitio Web, en HTML, se describe en la siguiente imagen:



Figura 2.22 - Estructura básica de un documento HTML. Fuente (Delgado, 2018).

2.6.1.2. *Cascading Style Sheets (CSS)*

CSS, traducido en español como “Hojas de Estilo en Cascada”, es una tecnología utilizada para definir la presentación (*aspecto visual*) de un documento HTML o página Web. La idea es separar el contenido o información, de su presentación visual. Se debe tener claro, que una página Web, es realmente un documento de texto, en el cual, se escribe código HTML, con el que se crea el

contenido de una Web. En dicho documento se genera un árbol, que puede ser modificado mediante el uso del lenguaje de programación CSS, y que, unido al código HTML, permite darle forma, color, posición (y otras características visuales) a una página, plantea (W3C, 2018).

CSS fue desarrollada por la organización W3C, la cual, lo considera un lenguaje esencial para el desarrollo de aplicaciones Web, ya que proporciona un mayor control del lenguaje HTML, ya que CSS utiliza el concepto de separación de presentación y contenido, lo que simplifica el código HTML, incluyendo sólo información y datos, relativos al significado de la información a transmitir (el contenido), y todos los aspectos relacionados con el estilo (diseño, colores, formas, entre otros), se encuentren en un documento CSS independiente (la presentación). Dicha tecnología, permite crear hojas de estilo que definen cada elemento a mostrarse en el sitio Web, como, por ejemplo, los encabezados, los enlaces, imágenes, entre otros. El término “en cascada”, indica que diferentes hojas de estilo se pueden aplicar sobre la misma página, plantea (Luján, 2002).

Al igual que con HTML, la W3C se ha encargado de publicar constantes revisiones del lenguaje, como el caso de CCS1 y CSS2, añadiendo así numerosas opciones en cuanto a la personalización, en términos del manejo de colores, márgenes, fuentes, técnicas de posicionamiento más precisas que permiten visualizar los elementos, en el lugar deseado en la página, entre otras opciones, plantean (Asensio & Menéndez, s.f).

La última versión que se utiliza es la tres (CSS3), la cual ofrece una gran variedad de alternativas muy importantes para las necesidades del diseño Web actual, tales como: funciones avanzadas de movimiento, bordes redondeados, degradados, sombras, entre otras.

Por otra parte, según (Eguiluz, 2009), entre una de las características principales de CSS, se encuentra: su flexibilidad, y las diferentes opciones que ofrece para realizar una misma tarea. De hecho, las formas más conocidas para aplicar CSS en un documento HTML son: Definir un archivo externo de tipo CSS, el cual contiene los estilos, y que posteriormente se enlaza con el documento HTML a través de la etiqueta <link>. La segunda forma, es incluir el CSS en el mismo documento HTML, utilizando la etiqueta <style>, y la última forma, es aplicar el estilo directamente en la etiqueta, incluyendo en la etiqueta del documento HTML, el atributo style. En la figura 2.23, se pueden observar las 3 opciones mencionadas.

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Esto es un ejemplo de CSS</title>
6   <!--Hoja de estilo vinculada-->
7   <link rel="stylesheet" href="estilo.css" type="text/css">
8   <!--Hoja de estilo incluida en el mismo HTML-->
9   <style type="text/css">
10    h1{
11      background: violet;
12      border-radius: 10%;
13    }
14  </style>
15 </head>
16 <body>
17   <h1>Hola Mundo!</h1>
18   <!--Hoja de estilo directamente en la etiqueta-->
19   <p style="font-size: 20px; color: blue; font-family: serif">Esto es un parrafo</p>
20 </body>
21 </html>

```

Figura 2.23 - Ejemplo de CSS

2.6.1.3. JavaScript

JavaScript es un lenguaje de programación ligero, e interpretado por la mayoría de los navegadores, y que les proporciona a las páginas Web, efectos y funciones complementarias, a las consideradas como estándar en HTML, lo que permite mejorar la interacción con los usuarios, y fue desarrollado por la compañía Netscape en el año 1995, plantea (Luján, 2002).

En esencia, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas desarrollados en JavaScript, se pueden probar directamente en cualquier navegador, sin la necesidad de utilizar procesos intermedios.

La integración de código JavaScript en un sitio Web es realmente sencilla, ya que existen al menos 2 formas distintas de realizarlo, las cuales son: Definir en un archivo externo de tipo JavaScript, las instrucciones de dicho lenguaje, y utilizar la etiqueta <script></script> dentro del HTML, para enlazar el código JavaScript, con el código HTML. Y la segunda forma, es incluir el código JavaScript en el mismo documento HTML, colocando las etiquetas <script> </script> en cualquier parte del documento, y dentro de las mismas, las instrucciones inherentes al lenguaje JavaScript, comenta (Equiluz, 2009).

En la Figura 2.24, se presentan ambas formas de integración de código JavaScript.

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <meta charset="UTF-8">
5  |   <title>Esto es un ejemplo de JavaScript</title>
6  |   <!--Archivo JavaScript enlazado-->
7  |   <script type="text/javascript" src="prueba.js"></script>
8  </head>
9  <body>
10 |   <h2>Esto es una prueba numero</h2>
11 |   <p id="demo"></p>
12 |   <!--Codigo JavaScript en el mismo HTML-->
13 |   <script type="text/javascript">
14 |       function myFunction(p1, p2){
15 |           return p1 * p2;
16 |       }
17 |       document.getElementById("demo").innerHTML = myFunction(1, 2);
18 |   </script>
19 </body>
20 </html>

```

Figura 2.24 - Ejemplo de JavaScript

2.6.1.4. JQuery

JQuery es una librería de JavaScript, pequeño y con muchas funciones. Hace que cosas, como la manipulación de documentos HTML, el manejo de eventos, la animación, y que las operaciones que utilicen AJAX (acrónimo de Asynchronous JavaScript And XML, es decir, JavaScript y XML asíncrono), sean mucho más sencillos, con una API que es fácil de utilizar, y que funciona en una multitud de navegadores. Con una combinación de versatilidad y extensibilidad, jQuery ha cambiado la forma en que millones de personas escriben JavaScript, facilitando la programación con dicho lenguaje, según (jQuery, 2018).

Así pues, dicho *framework* JavaScript, ofrece una infraestructura con la que se genera una mayor facilidad, para la creación de aplicaciones complejas del lado del cliente, ya que provee funciones a los desarrolladores que permiten tener un mayor control del DOM (Modelo de Objetos del Documento), que es un API (del inglés: *Application Programming Interface*), para documentos HTML. Por ejemplo, jQuery proporciona ayudas para la creación de interfaces de usuario, para generar efectos dinámicos con mayor facilidad, que en JavaScript, y ayuda a generar aplicaciones que hacen un uso intensivo de AJAX, según (jQuery, 2018).

2.6.1.5. Bootstrap

Bootstrap ha constituido un conjunto de herramientas de código abierto para desarrollar aplicaciones con HTML, CSS y JS. Se trata de un framework que ofrece la posibilidad de crear rápidamente prototipos, o sitios Web completos, de manera más sencilla, gracias al uso de sus clases de CSS predefinidas, y potentes complementos creados en jQuery. Bootstrap se ha caracterizado por tratarse de una excelente herramienta para crear interfaces de usuarios limpias, y totalmente responsive, es decir, adaptables a cualquier tipo de dispositivo y pantalla, independientemente de su tamaño, plantea (Getbootstrap, 2018).

2.6.1.6. AngularJS

AngularJS es un framework de JavaScript que brinda la posibilidad de escribir aplicaciones Web del lado del cliente. Permite utilizar HTML como lenguaje de plantilla, y ofrece extender su sintaxis para expresar los componentes de la aplicación de manera clara. A su vez, permite sincronizar automáticamente, los datos de las vistas con sus objetos JavaScript, a través de dos vías de enlace de datos, para ayudar a estructurar la aplicación, plantea (AngularJS, 2018).

2.6.1.7. Materialize

Materialize es un *framework* de HTML, CSS, and JavaScript para el desarrollo de sitios Web, siguiendo la normativa de diseño llamada *Material Design*, y permite que las interfaces de usuario sean adaptables a distintos dispositivos. Provee clases CSS predefinidas, y componentes que agilizan el desarrollo de interfaces de usuario, amigables y agradables a la vista, según (Materialize, s.f.).

2.6.2. Tecnologías del lado del servidor

Las tecnologías del lado del servidor, son aquellas que son reconocidas, ejecutadas e interpretadas por el propio servidor, y que se envían a través de la red al cliente, en un formato comprensible para él. Dichas tecnologías, son utilizadas para coordinar funciones en conjunto con las tecnologías del lado del cliente, como permitir el acceso a bases de datos u operaciones que no pueden ser realizadas en el código

del cliente propiamente. A continuación, se describen algunas de las tecnologías, y algunos lenguajes de programación, del lado del servidor: El servidor de páginas Web Apache, Heroku y PHP.

2.6.2.1. Servidor Apache

Un servidor Web o servidor HTTP (Protocolo de transferencia de archivos de hipertexto), es un programa informático que procesa una aplicación del lado del servidor, y que, realizando conexiones bidireccionales y/o unidireccionales, síncronas o asíncronas con el cliente, genera una respuesta a la aplicación que se encuentra en el lado del cliente. El código recibido por el cliente, suele ser compilado y ejecutado por un navegador Web, y para la transmisión de todos los datos, suele utilizarse algún protocolo. Generalmente se usa el protocolo HTTP para las comunicaciones, asevera (Apache Foundation, 2008).

El servidor HTTP Apache, es un servidor Web de código abierto, multiplataforma, que implementa el protocolo HTTP, y la noción de sitio virtual o sitio Web, el cual, es un conjunto de páginas Web que son accesibles desde un mismo dominio o subdominio de la World Wide Web (WWW). El servidor Apache, es desarrollado y mantenido por una comunidad de usuarios, bajo la supervisión de la Apache Software Foundation, dentro del proyecto HTTP Server (httpd). Apache es utilizado principalmente, para enviar páginas Web estáticas y dinámicas en la WWW, plantea (Apache Foundation, 2008).

Dicha herramienta, tiene una amplia aceptación en la red desde 1996, puesto que ha sido el servidor HTTP más utilizado. Así mismo, jugó un papel fundamental en el desarrollo de la WorldWideWeb, y alcanzó su máxima cuota de mercado en 2005, siendo el servidor empleado en el 70% de los sitios Web en el mundo. En 2009, se convirtió en el primer servidor Web que alojó más de 100 millones de sitios Web. La ventaja del uso de Apache, es que proporciona un servidor seguro, eficiente y extensible, que proporciona servicios de HTTP que se encuentran sincronizados con los estándares HTTP actuales, asevera (Apache Foundation, 2008).

2.6.2.2. Heroku

Es una plataforma en la Nube que soporta distintos lenguajes de programación, y funciona como hosting de aplicación. Los desarrolladores utilizan Heroku para implementar, administrar y escalar aplicaciones modernas, y es una plataforma elegante, flexible, fácil de usar, y ofrece a los desarrolladores un camino más sencillo para comercializar sus aplicaciones, asevera (Heroku, s.f.).

2.6.2.3. PHP

PHP es el acrónimo de Hipertext Preprocessor. Es un lenguaje de programación del lado del servidor, es de código abierto, rápido, con una gran librería de funciones y mucha documentación, lo que facilita el desarrollo de aplicaciones. Es un lenguaje utilizado para programar scripts del lado del servidor multiplataforma, y que se incrustan dentro del código HTML, plantea (Alvarez, 2001).

Para detallar un poco en qué consiste, que sea un lenguaje del lado del servidor, es aquel que se ejecuta en un servidor Web, justo antes de que se envíe la página, a través de internet al cliente. Las páginas que se ejecutan en el servidor, pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución del código PHP, lo que genera una gran ventaja al ser es compatible con todos los navegadores, ya que la página resultante, contiene únicamente código HTML. En la siguiente Figura 2.25, se puede observar el proceso anteriormente descrito por (Alvarez, 2001).



Figura 2.25 - Flujo de una página en php. Fuente (Alvarez, 2001).

Por otra parte, Rasmus Lerdorf fue originalmente el creador de dicho lenguaje, en el año 1994, pero como PHP ha sido desarrollado en política de software libre, a lo largo de su historia ha tenido muchas contribuciones de otros desarrolladores. Actualmente, la última versión de PHP se encuentra en la 7.0.

Marcos de trabajo de PHP

Existen una gran variedad de *frameworks* que facilitan el uso y desarrollo de sitios Web desarrollados con PHP. Sin embargo, para el presente desarrollo, se emplea el marco de trabajo Laravel, ya que la aplicación Web (SysNutr) que se utiliza como base para el presente desarrollo, es realizada con dicho framework; seguidamente, se presenta una breve definición del mismo:

Laravel: Es un *framework* creado en el año 2011, y actualmente se encuentra en continuo desarrollo. Dicho *framework* ha sido de código abierto, y es utilizado para el desarrollo de aplicaciones Web y servicios Web con PHP 5, permitiendo el uso del paradigma de programación Orientada a Objetos, y del patrón MVC. Su filosofía es desarrollar código PHP, de forma simple y elegante, evitando el "código espagueti" (es un término peyorativo utilizado para nombrar a los programas de computación que tienen una estructura de control de flujo compleja e incomprensible).

Laravel facilita tareas simples en el desarrollo Web como lo son el enrutamiento, la autenticación, sesiones y almacenamiento en caché, ya que consta de una sintaxis expresiva, la cual, facilita la realización de las actividades mencionadas, y además que posee una amplia documentación, según (Sierra, 2013).

A continuación, se proporciona un cuadro con las características más destacables de PHP, a modo de resumen, en base a lo definido anteriormente.

Tabla 2: Resumen de PHP

	PHP
Definición	Es un lenguaje de programación del lado del servidor, de código abierto, rápido, con una gran librería de funciones y mucha documentación, lo que ha facilitado el desarrollo de aplicaciones. Es un lenguaje utilizado para programar scripts del lado del servidor multiplataforma, que se incrustan dentro del código HTML.
Objetivo	Conceder facilidades a los desarrolladores de aplicaciones Web, para crear contenido y aplicaciones dinámicas.
Curva de aprendizaje	PHP resulta sencillo de aprender, para personas con conocimientos en lenguajes como JavaScript, Java o C, ya que su sintaxis es parecida a la de dichos lenguajes.
Marcos de trabajo/ Frameworks	Laravel, CodeIgniter, Yii, CakePHP, Symfony, entre otros
Grandes proyectos en los que ha sido utilizado	<i>Wikipedia, WordPress, Joomla!, Drupal, Magento, Prestashop, Symfony, Laravel.</i>
Ventajas	<ul style="list-style-type: none">• Es uno de los lenguajes más utilizados en el desarrollo Web. Cuenta con una comunidad de usuarios grande, lo cual es uno de los factores que influye a que posea muchísima documentación.• Instalación y configuración sencilla.• La curva de aprendizaje es muy corta.• Compatible con un gran número de bases de datos.• Diversos marcos de trabajo.• Lenguaje multiplataforma.• Es de código abierto.
Desventajas	<ul style="list-style-type: none">• PHP no cuenta con estándares de codificación, ni guías de buenas prácticas para sus desarrolladores.• Si no se utiliza un marco de trabajo donde se sigan un conjunto de buenas prácticas, el código puede quedar desordenado, difícil de leer por quedar incrustado dentro de HTML.

2.6.2.4. Sistemas Manejadores de Base de Datos (SMBD)

Los sistemas manejadores de bases de datos, surgen como una necesidad de las empresas, de poder manejar grandes y complejos volúmenes de datos, al tiempo que requieren compartir la información con un conjunto de clientes (que pueden ser

tanto aplicaciones como usuarios), de una manera segura. Ante dicho enfoque, un sistema manejador de bases de datos deberá ofrecer soluciones de forma fiable, rentable y de alto rendimiento, plantea (Menéndez & Barzanalla, 2000).

Según (Silberschatz, Korth, & Sundarshan, 2002), un SMDB consiste en una colección de datos interrelacionados, y un grupo de programas para acceder a los mismos. El objetivo principal de un SMDB, es proveer un entorno que sea conveniente y eficiente para las personas que lo utilizan, para la recuperación y almacenamiento de la información. Dichos sistemas se diseñan para almacenar grandes cantidades de información. La gestión de los datos involucra la definición de estructuras para el almacenamiento de la información, y la provisión de mecanismos para la gestión de la información. Además, los SMDB deben proporcionar la seguridad de la información almacenada, en caso de caídas del sistema o intentos de accesos sin autorización.

Actualmente, existe una gran variedad de Sistemas Manejadores de Bases de Datos. Muchos de ellos son software propietario, tales como: Oracle, Microsoft SQL Server, IBM Informix, Sybase, entre otros. Y por otra parte, se encuentran los SMDB de código abierto como PostgreSQL y MySQL. A continuación, se explica el SMDB PostgreSQL, ya que es el utilizado para el desarrollo de la aplicación Web/móvil.

PostgreSQL

PostgreSQL ha sido un avanzado sistema gestor de bases de datos relacionales, orientado a objetos y de código abierto. PostgreSQL utiliza un modelo Cliente/Servidor, y emplea multiprocesos en vez de multihilos, para garantizar la estabilidad del sistema, ofreciendo así, control de concurrencia multiversión, la cual, es una técnica avanzada para mejorar el rendimiento de la base de datos en un entorno multiusuario, y así mismo, soporta casi toda la sintaxis SQL (incluyendo subconsultas, transacciones, tipos y funciones definidas por el usuario). Además, cuenta también cuenta con un conjunto amplio de APIs, para la mayoría de lenguajes de programación (algunos de ellos: C, C++, Java, PHP, Perl, TCL y Python), asevera (PostgreSQL, 2010).

❖ Características

A continuación, se presentan algunas de las características más destacables de PostgreSQL, según (PostgreSQL, 2010):

- Es una base de datos totalmente ACID, siglas del inglés “*Atomicity, Consistency, Isolation and Durability*”; traducido como Atomicidad, Consistencia, Aislamiento y Durabilidad.
- Cuenta con integridad referencial.
- Posee unidad lógica de almacenamiento (tablespace).
- Contempla la posibilidad de realizar copias de seguridad en caliente (*Online/hot backups*).
- Posee múltiples métodos de autenticación, tales como: autenticaciones basadas en contraseñas, autenticación de identidad, autenticación de certificado, entre otros.
- Acceso encriptado vía SSL (Secure Socket Layer).
- Se encuentra disponible para Linux y UNIX en todas sus variantes (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), y Windows para versiones de 32 y 64bits.
- Provee una alta concurrencia, mediante un sistema denominado MVCC (Acceso concurrente multiversión, por sus siglas en inglés).
- Posee API's para programar en C/C++, Java, .Net, Perl, Python, Ruby, ODBC, PHP, Lisp, Scheme, Qt y muchos otros lenguajes de programación.

En el presente cuadro, se pueden englobar, las características más destacables de PostgreSQL, a modo de resumen, según todo lo descrito anteriormente del SMD.

Tabla 3: Resumen de PostgreSQL

	PostgreSQL
Definición	PostgreSQL ha sido un sistema gestor de bases de datos relacionales orientado a objetos y de código abierto, que utiliza un modelo Cliente/Servidor, y utiliza multiprocesos en vez de multihilos para garantizar la estabilidad del sistema.
Licencia	Se distribuye bajo la licencia PostgreSQL, una licencia (de código abierto), la cual concede la permisología para realizar cualquier modificación del código fuente, o distribuir el software para cualquier propósito, siempre que el aviso de copyright aparezca en todas las copias.
Límite	Con PostgreSQL se han podido manejar bases de datos de gran tamaño, en gigabytes y terabytes. La escalabilidad de los datos ha sido uno de sus objetivos en los últimos años.

Ventajas	<ul style="list-style-type: none"> • Es de código abierto. • Posee un buen sistema de seguridad mediante la gestión de usuarios, grupos de usuarios y contraseñas. • Gran velocidad de respuesta, incluso en situaciones donde incrementa el tamaño de la base de datos. • Brinda estabilidad y confiabilidad. • Posee gran escalabilidad. Ajustable al número de CPU's y a la cantidad de memoria que posee el sistema de forma óptima. • Puede operar sobre diferentes plataformas basadas en Unix y Windows. • Gran capacidad de almacenamiento. • Admite un gran número de peticiones simultáneas a la base de datos de forma correcta.
Desventajas	<ul style="list-style-type: none"> • Consume más recursos que MySQL. • Cuenta con cabeceras de intersección, lo cual degrada el rendimiento de la base de datos al momento de insertar o actualizar.

Así como se mencionan las herramientas tecnológicas para el desarrollo de aplicaciones Web. También, en el siguiente punto se describirán algunas de las herramientas tecnológicas utilizadas para el desarrollo de aplicaciones móviles.

2.7. Herramientas tecnológicas para el desarrollo de aplicaciones móviles

Con la creciente popularidad de las plataformas móviles, se han introducido nuevas tecnologías que facilitan el desarrollo de las aplicaciones para los distintos dispositivos. De acuerdo a la naturaleza de las aplicaciones que se deseen construir, la selección de una o varias herramientas puede cambiar dependiendo de las necesidades.

Si se trata de un desarrollo nativo, es natural utilizar herramientas o lenguajes propios de cada plataforma, como es el caso de Android Studio y Java, para el caso de Android, y Xcode para el desarrollo de iOS. Ahora bien, en el caso del desarrollo de aplicaciones híbridas existen varios *frameworks* que permiten llevar a cabo el desarrollo de aplicaciones móviles, sin embargo, Ionic y React Native se han convertido en los más utilizados en el área. Por lo tanto, se considera conveniente, abordar con detalle, cada uno de ellos.

2.7.1. IONIC

Ionic es un *framework* para el desarrollo de aplicaciones móviles creado por Drifty Co, en el año 2013. Está basado en AngularJS, y utiliza tecnologías Web para desarrollar y renderizar las aplicaciones en distintas plataformas. Adicionalmente utiliza PhoneGap y Cordova, que son otros *frameworks* de desarrollo de aplicaciones de código abierto, que utilizan una gran cantidad de librerías para acceder a las funciones nativas de los dispositivos, según (De la Serna, 2017).

El *framework* de Ionic sigue el lema “*Write once, run everywhere*”, lo que significa que el mismo código se podrá ejecutar en todas las plataformas que se requieran. Así mismo, ofrece una cantidad de componentes previamente desarrollados y con estilos predefinidos, que permite adaptar sus elementos a cada una de las diferentes interfaces de usuario (UI) para dar una impresión “nativa”, y así facilitar el desarrollo de las mismas. Grandes marcas, como Baskin Robbins, Diesel y McDonald’s, han optado por Ionic para desarrollar sus aplicaciones, según (Ashwini, 2017).

A continuación, se mencionan algunas de las ventajas y desventajas de dicho *framework*, según (Ashwini, 2017).

Ventajas

- Al compilar a través de Apache Cordova, se generan códigos para distintas plataformas como iOS y Android.
- Facilidad de implementación para el desarrollador, ya que es sencillo de aprender y trabajar.
- Si se conoce AngularJS, se puede escribir código en TypeScript, lo que permite desarrollar aplicaciones multiplataformas mediante el uso de TypeScript.
- Provee al desarrollador, acceso a las funcionalidades nativas de los dispositivos, mediante el sistema de *plugins*, debido a la integración que posee con Cordova.
- En caso de tener una aplicación Web, es sencillo convertirla en una aplicación móvil, ya que el uso de JavaScript permite que el código sea reutilizable.
- Facilita el desarrollo de proyectos de forma más efectiva, ya que los principales lenguajes a utilizar son: HTML con CSS y JavaScript, los cuales son muy conocidos por la comunidad de desarrolladores.

Desventajas

- Podrían presentarse problemas de rendimiento, si se desea utilizar el código nativo en repetidas ocasiones.
- El desarrollo de gráficos avanzados o transiciones interactivas resulta difícil.

CAPÍTULO III: MARCO METODOLÓGICO

Blanco (2008), señala que una metodología de desarrollo, es una colección de documentación formal referente a los procesos, las políticas y los procedimientos que intervienen en el desarrollo del software. La finalidad de una metodología de desarrollo, es garantizar la eficacia y la eficiencia en el proceso de generación de software, y que para lograr el desarrollo de una solución tecnológica de la manera más estructurada, planificada y controlada posible, es necesario tener y seguir una metodología de desarrollo de software.

En el presente capítulo, se mencionan las fases, roles y artefactos de la metodología ágil, eXtreme Programming (XP), también conocido como Programación Extrema.

Metodología para el desarrollo

Las metodologías de desarrollo, consisten en un conjunto de etapas, procedimientos, técnicas, herramientas y documentación, que ayuda a los desarrolladores a crear un nuevo software, de manera sistemática. Dichas metodologías, han buscado, el cumplir con tres necesidades de relevancia a saber: el generar un proceso de desarrollo controlado, el tener un proceso normalizado en una organización (que cumpla con los estándares de desarrollo), y finalmente, el crear mejores aplicaciones que conllevan a una mejor calidad, asevera (Ecured, s.f.).

3.1. Programación Extrema (XP)

La programación extrema o eXtreme Programming (XP), es una metodología de desarrollo formulada por Kent Beck, y que se centra en aumentar las relaciones interpersonales, como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y facilitando un buen ambiente de trabajo. La programación extrema se diferencia de las metodologías tradicionales principalmente, en que pone más énfasis en la adaptabilidad, que en la previsibilidad, asegura (Wells, 1999).

Se describen a continuación los artefactos utilizados en la metodología XP.

3.1.1. Artefactos XP

Entre los documentos que se suelen generar, durante la aplicación de la metodología XP, se mencionan los siguientes, según (Wells, 1999):

- **Historias de usuario**

Las Historias de usuarios sustituyen a los documentos de especificación funcional, y a los casos de uso. Dichas historias, representan una breve descripción del comportamiento del sistema, y las cuales, son escritas por el cliente en su propio lenguaje, es decir, sin lenguaje técnico; y se emplean para realizar estimaciones de tiempo; y para el plan de lanzamientos, sustituyen un gran documento de requisitos, y presiden la creación de las pruebas de aceptación. La diferencia más importante entre dichas historias, y los tradicionales documentos de especificación funcional, se encuentran en el nivel de detalle requerido.

- **Tareas de ingeniería**

Son tarjetas que se elaboran para ayudar y simplificar la programación de una historia de usuario, de modo que se vinculan más al desarrollador, ya que permite tener un acercamiento con el código.

- **Pruebas de aceptación**

Son tarjetas que se utilizan para realizar las pruebas de cada historia de usuario, con la finalidad de confirmar que han sido implementadas correctamente.

- **Tarjetas CRC (Clase - Responsabilidad - Colaborador)**

Las Tarjetas CRC, describen las clases utilizadas en la programación de una historia, permitiendo conocer, cuáles de ellas componen al sistema, y cuales interactúan entre sí.

A continuación, se definen los roles utilizados en la metodología XP.

3.1.2. Roles

En dicha metodología, se pueden determinar los siguientes roles que se describen a continuación, según (Extreme Programming, s.f.):

- **Cliente:** Es el encargado de escribir las historias de usuario, asignarle prioridad a las mismas, y decidir cuáles deben implementarse en cada iteración, aportando así, mayor valor al negocio. De igual manera, es quien debe realizar las pruebas funcionales para validar la implementación de cada historia.
- **Programador:** Es quien convierte las historias de usuario, en código. Una vez concebidas, el programador es el encargado de determinar el tiempo de desarrollo en el que se debe completar cada una de ellas, y cuáles son las tareas que les compete.
- **Encargado de pruebas:** Ayuda al cliente en la elaboración de pruebas funcionales, para validar las historias de usuario. También es responsable de realizar *tests*, e informar los resultados al resto del equipo. Debe, además, llevar un control de las herramientas utilizadas para el soporte de las pruebas.
- **Encargado de seguimiento:** Realiza el seguimiento que se ha trazado en la planificación. Durante cada iteración, evalúa el progreso de la misma, de acuerdo al cumplimiento de las tareas asignadas. Mantiene un contacto continuo con el equipo de desarrollo, y realiza los cambios necesarios para cumplir los objetivos.
- **Entrenador:** Su papel fundamental, es de guiar y orientar al equipo. También se encarga de la elección de metodologías, y herramientas para el desarrollo.
- **Gestor:** Es el dueño del equipo y sus problemas. Es quien debe conseguir los recursos necesarios para manejar los problemas, y administrar las reuniones. Debe estar familiarizado con la idea general del proyecto, aunque en ocasiones sucede, que el gestor es ejercido por el mismo cliente.

- **Consultor:** Definido como un rol opcional, el consultor es un miembro externo del equipo, con conocimientos particulares, al cual se puede acudir para resolver un problema específico.

Seguidamente se mencionan las fases o etapas utilizadas en dicha metodología.

3.1.3. Fases o Etapas

En la metodología XP, se determinan 4 fases para lograr un correcto cumplimiento de los objetivos planteados, descritas a continuación, según (Extreme Programming, s.f.):

Planeación

La fase de planeación, comienza recopilando Historias de usuarios que describen las características y funcionalidad del software que se va a elaborar. Cada historia (similar a los casos de usos), es escrita por el cliente quien asigna un valor o prioridad a la misma. Seguidamente, los programadores evalúan cada historia y le asignan un costo o tiempo de desarrollo. Si se estima que la historia requiere más de 3 semanas de desarrollo, se pide al cliente que la descomponga en historias más cortas. Así mismo, si existe posibilidad de una mayor complejidad de desarrollo que no se haya contemplado, para ambos casos, se debe asignar de nuevo, un valor y costo. Es importante observar, que en cualquier momento es posible escribir nuevas historias.

Una vez realizadas las estimaciones, se establece una reunión de planificación, con los distintos actores del proyecto (cliente, desarrolladores, gerentes), con la finalidad de definir un plan o cronograma de entregas, en los que todos se encuentren de acuerdo. Ya acordado el cronograma, comienza una fase de iteraciones, en donde las historias de usuarios seleccionadas para cada entrega, son desarrolladas y probadas, de acuerdo al orden preestablecido.

La fase de planeación, se lleva a cabo mediante reuniones diarias de seguimiento, con el fin de llevar a cabo todo lo antes mencionado, y compartir problemas y soluciones que se presenten durante el desarrollo.

Diseño

El diseño XP sigue rigurosamente el principio MS (Mantenlo Sencillo), es decir, se enfoca en diseños simples y claros, ya que los mismos siempre se prefieren, sobre una representación más compleja.

XP ha estimulado el uso de tarjetas CRC, como un mecanismo eficaz para visualizar un software, en un entorno orientado a objetos. Las tarjetas CRC son el único producto de trabajo de diseño que se generan como parte del proceso XP.

Dicha metodología promueve el rediseño, técnica que se utiliza como método para la optimización del diseño, que consiste en escribir nuevamente, parte del código de un programa, sin cambiar su funcionalidad. Es una manera disciplinada de limpiar el código, disminuyendo la probabilidad de obtener errores. En otras palabras, cuando se rediseña, se mejora el diseño del código después de haber sido escrito.

Por otra parte, para que el proceso de diseño XP sea óptimo, se utilizan las metáforas, que se utilizan como una manera sencilla de explicar el propósito del proyecto, y guiar la estructura y arquitectura del mismo. Así mismo, se utilizan las soluciones “*spike*” (programas de prueba), para estimar el tiempo de desarrollo, en caso de haber algún riesgo que implique mayor complejidad en el mismo.

Codificación

Una vez desarrolladas las historias de usuario, y realizado el trabajo de diseño preliminar, el equipo no inicia la codificación, sino que desarrolla una serie de pruebas unitarias a cada una de las historias que se van a incluir, al momento de la entrega que se va ejecutando. Dado que fue creada la prueba unitaria, el desarrollador se encuentra capacitado para centrarse en lo que debe implementarse para pasar la prueba.

XP promueve la programación en parejas, es decir, que todo el código sea escrito en pareja, y así incrementar la calidad del código sin impactar en la fecha de entrega.

Pruebas

Como se mencionó anteriormente, las pruebas deben ser definidas antes de realizar la codificación. Se debe asegurar que todo código implementado, cumpla correctamente con las pruebas unitarias, y así, verificar que funciona. Por lo tanto, el sistema y el conjunto de pruebas, debe ser guardado junto con el código, para que

pueda ser utilizado por otros desarrolladores, en caso de tener que corregir, cambiar o recodificar parte del mismo.

En la metodología XP, además de las pruebas unitarias, también se tienen las pruebas de aceptación, que son definidas por el cliente, y se centran en las características y funcionalidad generales del sistema, que son visibles por parte del mismo.

En la Figura 3.1 se ilustran las fases de la metodología XP.

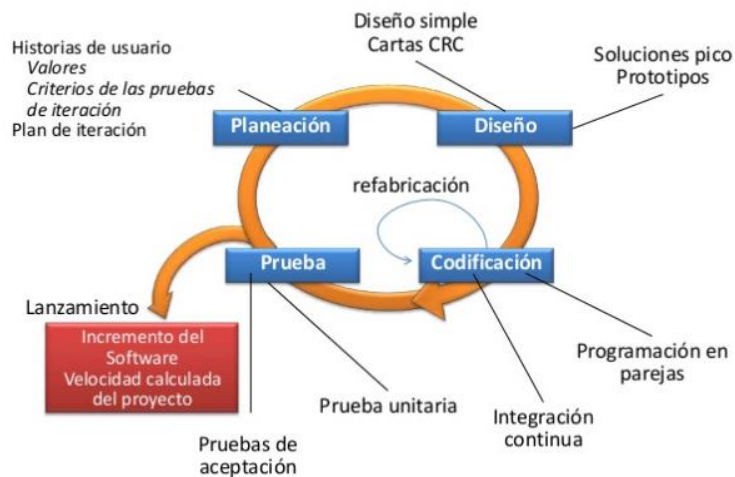


Figura 3.1 - Fases de la metodología XP. Fuente (Extreme Programming, s.f.).

En la Tabla 4, se proporciona un breve resumen de las características más destacadas, de la metodología ágil mencionada.

Tabla 4: Resumen de XP

	XP
Definición	Es una metodología de desarrollo formulada por Kent Beck, centrada en aumentar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y facilitando un buen ambiente de trabajo
Roles	<ul style="list-style-type: none"> • Cliente • Programador • Encargado de prueba • Encargado de Seguimiento • Entrenador • Gestor • Consultor

Fases	<ul style="list-style-type: none"> • Planeación • Diseño • Codificación • Pruebas
Tiempo de entrega	Las iteraciones de entrega son de 1 a 3 semanas
Artefactos	<ul style="list-style-type: none"> • Historias de usuario • Tareas de ingeniería • Pruebas de aceptación • Tarjetas CRC
Ventajas	<ul style="list-style-type: none"> • Menor tasa de errores, al utilizar la programación en pareja • Se tiene comunicación continua con el cliente • Da lugar a una programación sumamente organizada • Las empresas y equipos pequeños, pueden adaptarse mejor a esta metodología • Se hacen pruebas continuas durante el proyecto • Implementa una forma de trabajo que se adapta fácilmente a las circunstancias
Desventajas	<ul style="list-style-type: none"> • Dependencia hacia el cliente • Altas comisiones en caso de fallar • El cliente puede cambiar la fecha de entrega en cualquier momento, y por ende, la duración puede ser muy pequeña • Es recomendable utilizarlo solo en proyectos a corto plazo

Para el desarrollo de la Aplicación Web/móvil, se decide usar una adaptación de la metodología XP, por su facilidad para abordar proyectos pequeños en un periodo corto de tiempo, debido a que se quería contar con una versión del producto, lo más pronto posible, para realizar una prueba piloto con usuarios reales, y así detectar errores y valorar la experiencia de los actores involucrados durante su uso. Dicha adaptación de la metodología está centrada en las 4 fases del ciclo de desarrollo de XP, con determinadas actividades definidas de acuerdo con las necesidades del proyecto.

CAPÍTULO IV: MARCO APLICATIVO

En el presente capítulo se realiza una descripción del proceso de creación de la aplicación Web/móvil, que es desarrollada bajo una adaptación de la metodología de desarrollo ágil XP. Adicionalmente, se indican qué elementos se toman en consideración en dicha adaptación, y las actividades realizadas en cada una de las fases del ciclo de vida de XP.

A continuación, se explican de forma detallada cada una de las fases implementadas para el desarrollo de la aplicación Web/móvil.

4.1. Fases del Proyecto

La construcción de la aplicación Web/móvil, se realiza en cuatro fases: Planificación, Diseño, Codificación y Pruebas. Se implementa una única iteración, y en la cual se muestran avances a los docentes de la Universidad Central de Venezuela, Concettina Di Vasta y José Sánchez, con el fin de acelerar los tiempos de desarrollo, para obtener una versión del aplicativo lo más pronto posible, que permitiera realizar una prueba, en la cual se simula el proceso de evaluación nutricional y el seguimiento del plan dieto-terapéutico, con data aleatoria y usuarios reales, donde se valora el comportamiento de la misma y la experiencia de los usuarios durante su uso, y así; determinar si cumple con las necesidades de los profesionales y pacientes del área de Nutrición y Dietética. En la Figura 4.1 se pueden observar los elementos tomados en cuenta para cada una de las fases del ciclo de desarrollo de la adaptación de XP

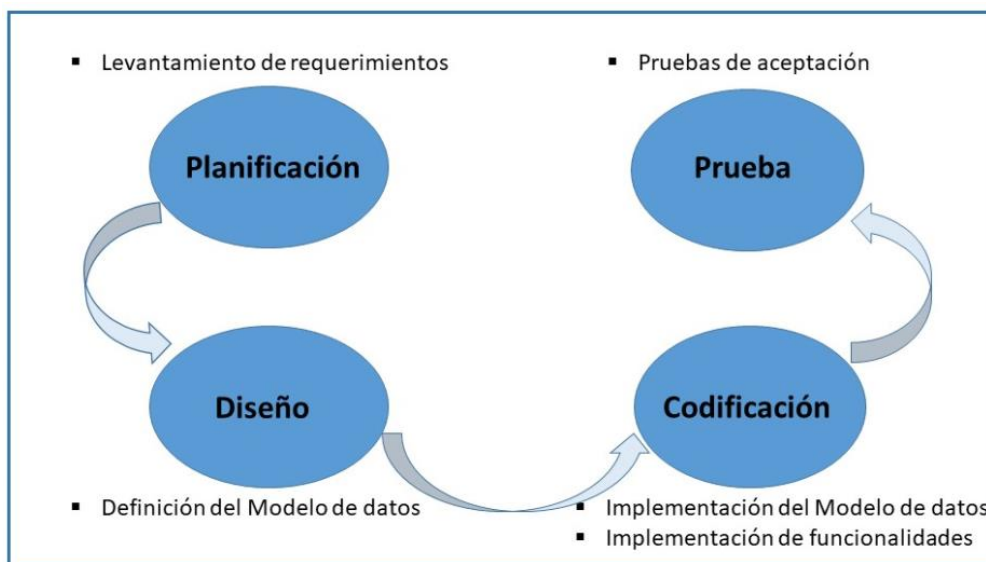


Figura 4.1 - Fases del ciclo de vida de la adaptación de XP.

Posteriormente se procede a describir, con detalle, cada una de las fases del ciclo de vida de la adaptación de la metodología XP, y las actividades realizadas en ellas durante el desarrollo de la aplicación Web/móvil.

4.1.1. Fase de planificación

La fase de planificación consta de reuniones presenciales e interacción electrónica con Jacqueline Torcat y Durbis Peña (médicos nutriólogos), y Concettina Di Vasta y José Sánchez (tutores y docentes de la Escuela de Computación, UCV), donde se levantan los requerimientos funcionales y no funcionales, necesarios para completar la automatización del proceso de evaluación nutricional, y llevar el seguimiento del plan dieto-terapéutico. Así se tuvo una visión general del producto final al que se quería llegar.

Seguidamente, se mencionan los requerimientos iniciales, definidos de acuerdo a las necesidades de los usuarios (profesionales y pacientes en el área de nutrición).

4.1.1.1. Requerimientos iniciales

En dicho ítem, se realiza una descripción de los requerimientos funcionales y no funcionales definidos durante la fase de planificación, de la siguiente manera:

Requerimientos funcionales

Como usuario nutricionista:

- Iniciar/Cerrar sesión en la aplicación móvil.
- Consultar la lista de pacientes, con su respectivo historial médico y menús, a través de la aplicación móvil.
- Consultar citas y reportes de pacientes, dentro de la aplicación Web/móvil.
- Crear citas para pacientes en la aplicación Web.
- Reprogramación de citas.
- Asignar menús a pacientes.
- Mostrar y generar indicadores como: cuántos pacientes tienen antecedentes familiares (diabetes, obesidad, entre otras), la cantidad de pacientes con edades entre 18 a 50 años, que presentan desnutrición, entre otros, dichos indicadores serán de interés para los médicos nutriólogos, y desplegados de forma gráfica.
- Crear, asignar y eliminar recomendaciones para los pacientes en la aplicación Web.
- Enviar mensajes entre el paciente y el nutricionista.

Como usuario paciente:

- Iniciar/Cerrar sesión en la aplicación Web/móvil.
- Reprogramación de citas.
- Crear reportes sobre su avance diario.
- Enviar mensajes entre el paciente y el nutricionista.
- Consultar perfil, plan dieto-terapéutico, menús, citas, recomendaciones y progreso.
- Realizar el cálculo de calorías por plato, dentro de la aplicación móvil.

Adicionalmente, se determinan los requerimientos no funcionales, descritos a continuación:

Requerimientos no funcionales

- La aplicación Web/móvil desarrollada, es usable y posee una interfaz intuitiva.
- Es portable, de modo que puede ser accedida desde cualquier dispositivo.
- La aplicación Web/móvil es capaz de prever errores, mediante la definición de validaciones.
- Es mantenible, por tanto, su actualización y corrección de fallas, se realiza con un esfuerzo mínimo.

Para comprender mejor las funcionalidades y el comportamiento de la aplicación Web/móvil, se realizan las historias de usuario para describir con más detalle cada una de ellas.

4.1.1.2. Historias de usuario

En la Tabla 5 se proporcionan las historias de usuario, a través de las cuales se pueden observar, los requerimientos funcionales de la aplicación Web/móvil que surgen, luego de varias reuniones con los clientes (profesionales y pacientes en el área de nutrición), definiendo así, las funcionalidades deseadas para el producto final. Adicionalmente se lista y explica, cada una de dichas funcionalidades, siguiendo el modelo de historias de usuarios de (Cohn, 2011), en el cual se establece lo siguiente: El número de historia, nombre, prioridad, riesgo de desarrollo, descripción y observaciones.

Tabla 5: Historias de usuario.

Número	Nombre
1	Autenticación de usuarios en la aplicación móvil.
2	Visualización de consultas.
3	Gestión y reprogramación de citas.
4	Asignación de menús.
5	Generación de indicadores.
6	Gestión y asignación de recomendaciones.
7	Envío de mensajes entre usuarios.
8	Calculadora de calorías.
9	Generación de reportes de pacientes.

A continuación, se describen en detalle cada una de las historias de usuario ya mencionadas.

Tabla 6: Historia de usuario – Autenticación de usuarios.

Historia de usuario	
Número: 1	Usuario: Nutricionista y Paciente.
Nombre: Autenticación de usuarios en la aplicación móvil.	
Prioridad: 1	Riesgo de desarrollo: Bajo
Descripción: Permite a los usuarios iniciar y cerrar sesión en la aplicación móvil, mediante su email y contraseña.	
Observaciones:	

Tabla 7: Historia de usuario – Visualización de consultas.

Historia de usuario	
Número: 2	Usuario: Nutricionista y Paciente.
Nombre: Visualización de consultas.	
Prioridad: 1	Riesgo de desarrollo: Bajo
Descripción: Permite que los nutricionistas realicen consultas de información que le pertenece, tales como: su perfil de usuario, sus citas, menús plantilla, lista de pacientes y cualquier información asociada los mismos. Además, a dichos pacientes, se les permite realizar consultas de toda la información referente a su plan dieto-terapéutico, tales como: menús, recomendaciones, su progreso y cita.	
Observaciones:	

Tabla 8: Historia de usuario – Gestión y reprogramación de citas.

Historia de usuario	
Número: 3	Usuario: Nutricionista y Paciente.
Nombre: Gestión y reprogramación de citas.	
Prioridad: 1	Riesgo de desarrollo: Bajo
Descripción: Permite a los nutricionistas establecer y reprogramar citas a pacientes, y como usuario paciente, permite solicitar un cambio de cita.	
Observaciones:	

Tabla 9: Historia de usuario – Asignación de menús.

Historia de usuario	
Número: 4	Usuario: Nutricionista.
Nombre: Asignación de menús.	
Prioridad: 1	Riesgo de desarrollo: Bajo
Descripción: Permite asignar un menú previamente creado, a un paciente en específico.	
Observaciones:	

Tabla 10: Historia de usuario – Generación de indicadores.

Historia de usuario	
Número: 5	Usuario: Nutricionista.
Nombre: Visualizar indicadores.	
Prioridad: 1	Riesgo de desarrollo: Bajo
Descripción: Permite al nutricionista realizar consultas, y visualizarlos de forma gráfica.	
Observaciones:	

Tabla 11: Historia de usuario - Gestión y asignación de recomendaciones.

Historia de usuario	
Número: 6	Usuario: Nutricionista.
Nombre: Gestión y asignación de recomendaciones.	
Prioridad: 1	Riesgo de desarrollo: Bajo
Descripción: Permite al nutricionista crear, eliminar y asignar recomendaciones a un paciente en específico.	
Observaciones:	

Tabla 12: Historia de usuario - Envío de mensajes entre usuarios.

Historia de usuario	
Número: 7	Usuario: Nutricionista y paciente.
Nombre: Envío de mensajes entre usuarios.	
Prioridad: 1	Riesgo de desarrollo: Bajo
Descripción: Permite al paciente enviar mensajes a su nutricionista, y el mismo puede responder a su mensaje.	
Observaciones:	

Tabla 13: Historia de usuario - Calculadora de calorías.

Historia de usuario	
Número: 8	Usuario: Paciente.
Nombre: Calculadora de calorías.	
Prioridad: 1	Riesgo de desarrollo: Bajo
Descripción: Permite al paciente calcular las calorías de un determinado plato.	
Observaciones:	

Tabla 14: Historia de usuario - Generación de reportes de pacientes.

Historia de usuario	
Número: 9	Usuario: Paciente.
Nombre: Generación de reportes de pacientes.	
Prioridad: 1	Riesgo de desarrollo: Bajo
Descripción: Permite al paciente realizar un reporte de su avance diario, el cual, al ser completado, será enviado a su nutricionista.	
Observaciones:	

4.1.2. Fase de diseño

Una vez culminada la fase de planificación, con los requerimientos e historias de usuario ya definidos, se inicia la fase de diseño. Se establece el modelo de datos y la arquitectura a utilizar, descritos a continuación:

4.1.2.1. Modelo de datos

La base de datos donde se almacena todos los datos necesarios para el funcionamiento de la aplicación Web SysNutr, fue diseñada inicialmente con el SMDB MySQL; para el desarrollo de la aplicación Web/móvil se continúa con el mismo modelo de base de datos, sin embargo, se realiza una migración al SMDB PostgreSQL.

El modelo de datos relacional de la aplicación Web/móvil, está compuesto por 20 tablas que se describen a continuación:

- **Usuario:** Contiene la información general de los usuarios que pueden ingresar a la aplicación Web/móvil, como nombre, email y password.
- **Patients:** Almacena todos los datos de un paciente, entre ellos, datos personales, historia médica, historia dietética, tolerancias de alimentos, hábitos y evaluación nutricional. Un paciente puede estar relacionado con un único nutricionista.
- **Antecedentes:** Comprende los tipos de antecedentes personales y familiares.
- **Antecedentes_patient:** Indica los antecedentes personales y familiares que posea cada paciente.
- **Anthropometries:** Almacena las medidas antropométricas de un paciente por cada consulta. Dichas medidas, son necesarias para realizar todos los cálculos internos de la aplicación. Un paciente puede tener varias antropometrías.
- **Laboratories:** Contiene todos los exámenes paraclínicos que un paciente posea. Un paciente puede tener varios exámenes registrados.
- **Foods:** Contiene los alimentos más utilizados en la sociedad venezolana, y cualquier alimento que el nutricionista desee agregar a su repertorio. De dichos alimentos, se almacena su contenido de nutrientes, y cantidades por intercambio.
- **Foods_Groups:** Contiene los grupos mediante los cuales se agrupan los elementos de la tabla alimentos. Alimentos del mismo grupo poseen nutrientes similares.
- **Diets:** Almacena los valores de la fórmula dietética, y el número de intercambios que se deben ingerir.
- **Menus:** Contiene la distribución de alimentos a lo largo del día.

- **Menu_Patient_Users:** Indica que menús han sido asignados a los pacientes y a los nutricionistas, dueños de dichos menús.
- **Menu_Contents:** Contiene los alimentos que conforman un menú creado, así como, las cantidades asignadas de los mismos, y la hora del día en la cual deben ser ingeridos.
- **Reference_Values:** Comprende los rangos de valores aceptables por sexo, para cada uno de los valores de un examen de laboratorio.
- **Citas:** Almacena todas las citas establecidas por el nutricionista, a sus pacientes.
- **Notifications:** Contiene todas las notificaciones de los nutricionistas y pacientes, ya sean por solicitud de cambio de cita, o mensajes entre ambos usuarios.
- **Reportes:** Almacena todos los reportes de avance diario de cada paciente, indicando a qué paciente y nutricionista pertenece, y un estatus que denote si es completado o no el reporte.
- **Preguntas:** Comprende todas las preguntas que debe responder el paciente para completar su reporte de avance diario.
- **Reporte_preguntas:** Indica todas las respuestas a cada una de las preguntas asociadas a un reporte.
- **Recomendaciones:** Contiene las recomendaciones generales y específicas para un paciente, creadas y asignadas por un nutricionista.
- **Recomendación_patient:** Indica que recomendaciones específicas han sido asignados a los pacientes y los nutricionistas dueños de dichas recomendaciones.

En la Figura 4.4 se expone la representación del modelo de datos relacional en el cual se observa las tablas mencionadas.

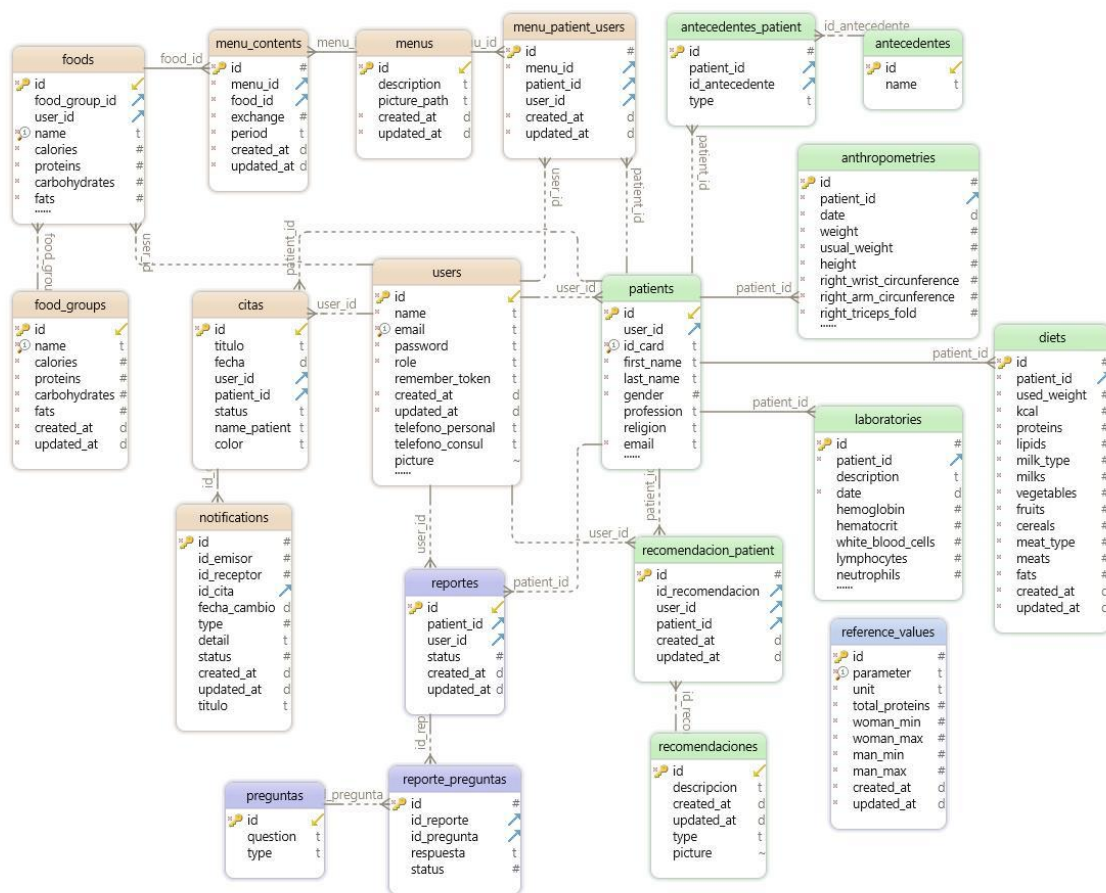


Figura 4.4 - Modelo relacional de la Base de Datos.

En el siguiente punto se indica el diseño de arquitectura seleccionada para el desarrollo de la aplicación Web/móvil.

4.1.2.2. Diseño de la Arquitectura Tecnológica de la Solución

La aplicación Web/móvil está desarrollada, utilizando la arquitectura Cliente/Servidor, en donde el cliente inicia la comunicación, realizando peticiones al servidor, y dicho servidor se encarga de procesar la petición, y generar una respuesta al mismo. Utiliza un modelo de tres capas, integrado por las capas de presentación, lógica de negocio y datos.

En la capa de presentación se crean interfaces dinámicas, estructurales y agradables para los usuarios, utilizando tecnologías como HTML, CSS y JavaScript para la aplicación Web; y IONIC y TypeScript para la aplicación móvil.

En segundo lugar, se encuentra la capa de lógica de negocio, la cual responde a los eventos realizados por el usuario, mediante las interfaces y se encarga de realizar la lógica apropiada para dar respuesta a dichos eventos. En dicha capa se emplea el framework Laravel, el cual brinda un conjunto de funcionalidades ya establecidas que se adaptan a los requerimientos del desarrollo.

Y por último, la capa de datos, que se encarga de almacenar, recuperar y asegurar la integridad de los mismos. Para su correcta manipulación, en el caso de dicha capa, se utiliza el SMDB PostgreSQL. En la Figura 4.5 se proporciona el esquema de la arquitectura utilizada.



Figura 4.5 – Arquitectura de la aplicación Web/móvil.

Una vez definida la arquitectura de la solución, se utilizan tecnologías para realizar el desarrollo de la aplicación Web/móvil.

4.1.2.3. Tecnologías utilizadas

Para el desarrollo de la aplicación Web/móvil se seleccionan tecnologías del lado del cliente y el servidor, y que son clasificadas en: Framework, Lenguajes de Programación y Bases de Datos. Para el desarrollo del módulo de pacientes del área de Nutrición y Dietética, que es incorporado en la aplicación Web ya desarrollada (SysNutr), se continúa con el uso de las mismas tecnologías con las que es implementada dicha aplicación, para agilizar el tiempo de desarrollo. Dichas tecnologías, son el *framework* basado en PHP, Laravel, en su versión 5.4. Para hacer uso de dicho marco de trabajo, es necesario instalar el paquete correspondiente a

través del gestor de paquetes Composer, y luego crear un proyecto de Laravel dentro del servidor. En cuanto al desarrollo de la aplicación móvil, se utiliza el *framework* IONIC, basado en Angular, junto con el lenguaje de programación TypeScript, el cual es un superconjunto de JavaScript.

Por otro lado, para la parte visual y algunas funcionalidades del lado del cliente, se utilizan los lenguajes HTML, CSS y el *framework* Materialize que permite desarrollar de manera rápida, las interfaces y que dichas interfaces sean agradables a la vista del usuario y fáciles de utilizar; además, se utilizan las librerías de JQuery, la cual es incluida automáticamente por Laravel al crear el proyecto, y Chart.js, que ofrece funcionalidades para la presentación de consultas mediante gráficas. Por último, en torno a la base de datos, se selecciona el SMD PostgreSQL, por su facilidad de uso, versatilidad y buena integración con Laravel.

Dichas tecnologías son seleccionadas para el desarrollo de la aplicación Web/móvil, ya que se rigen por el principio de código abierto, lo que permite adquirirlas de manera gratuita, y el poder instalarlas y configurarlas de manera fácil; y al ser mantenidas constantemente por la comunidad de desarrolladores, suelen añadirse nuevas características y mejoras con un mínimo esfuerzo. Por otra parte, además de adaptarse fácilmente a los requerimientos de la aplicación, permiten obtener un código de gran calidad y ahorrar tiempo de desarrollo, ya que al ser utilizados por una gran cantidad de desarrolladores en el mundo, se puede encontrar bastante documentación y ayuda en los diferentes foros.

4.1.3. Fase de codificación

En dicha fase, se implementan las funcionalidades que son definidas en las historias de usuario, en base a los requerimientos funcionales de la aplicación Web/móvil. Al crear un proyecto con el *framework* Laravel se crea la estructura de directorios y archivos que se aprecia en la Figura 4.6.

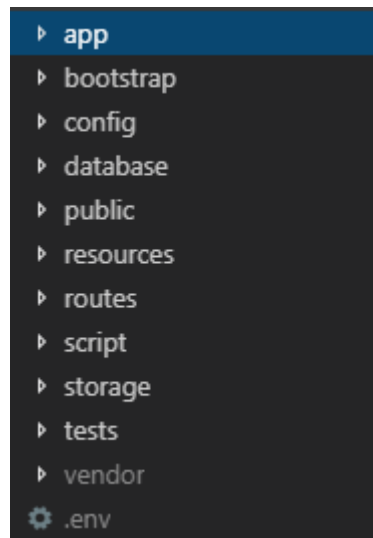


Figura 4.6 - Estructura de directorios del proyecto en Laravel

Para efectos del desarrollo de la aplicación se debe prestar especial atención a los directorios “app”, “database”, “public”, “resources” y “routes”. El contenido de dichos directorios se explican a continuación:

- **Directorio “app”:** Contiene todos los archivos correspondientes al modelo de datos en los cuales se definen las relaciones existentes entre las distintas tablas de la base de datos. Además, contiene el subdirectorio “Http/Controllers”, en el cual se definen y desarrollan los controladores que se encargan de manejar todas las peticiones que se realicen al servidor. En dicho directorio, también se encuentran otros archivos y subdirectorios creados por Laravel para realizar el proceso de autenticación (Ver Figura 4.7).

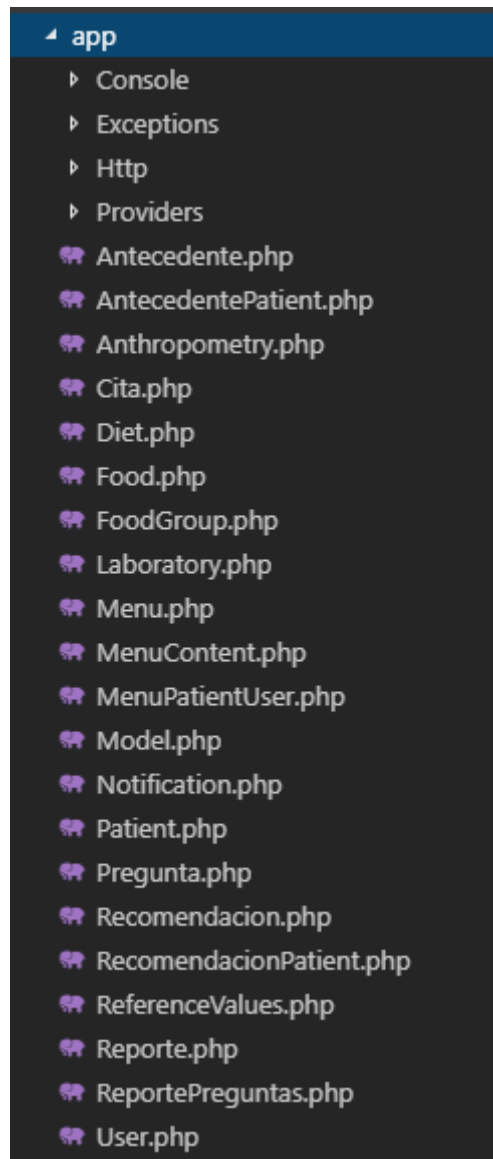


Figura 4.7 - App folder.

- **Directorio “database”:** Comprende el subdirectorio “migrations”, en el cual se colocan todos los archivos que definen cada una de las tablas de la base de datos con sus respectivos atributos, el tipo de dato y sus restricciones. Durante el desarrollo también se decidió utilizar el subdirectorio “seeds”, para la realización de procesos de extracción, transformación y carga de datos, ya que en dicho directorio se colocan archivos para realizar inserciones automáticas en la base de datos (Ver Figura 4.8).

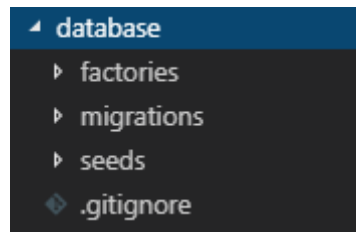


Figura 4.8 - Database folder.

- **Directorio “public”:** Contiene los subdirectorios llamados “css”, “fonts” y “js”, en los cuales se encuentran los archivos que contienen las clases css, las fuentes y las funcionalidades programadas en lenguaje Javascript que respectivamente son utilizadas en las interfaces de usuario. Y a su vez, podemos encontrar en dicho directorio, otros archivos que definen permisos de acceso a la aplicación, como la página cargada en principio, el icono que se muestra en la pestaña del navegador al abrir la aplicación, entre otras cosas (Ver Figura 4.9).

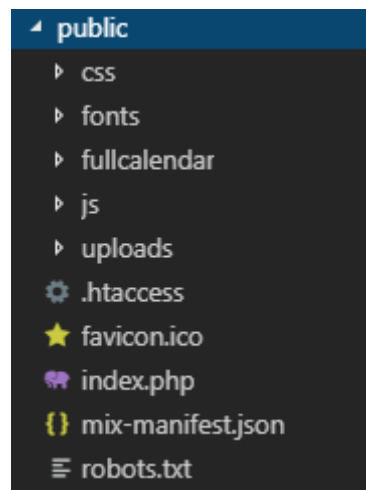


Figura 4.9 - Public folder.

- **Directorio “resources”:** Incluye todos los archivos en los cuales se desarrollan las interfaces de usuario de la aplicación, es decir, los códigos HTML. Se suele crear un subdirectorio para almacenar las vistas de cada módulo de la aplicación, y mantener el código más organizado (Ver Figura 4.10).

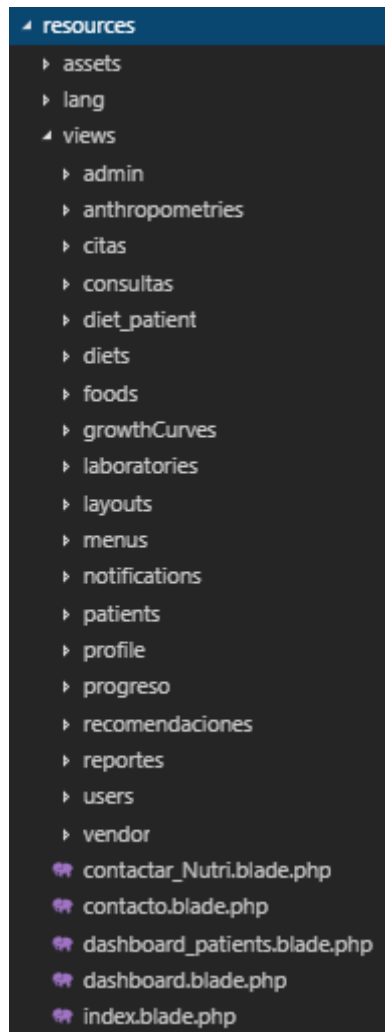


Figura 4.10 - Resources folder.

- **Directorio “routes”:** es un directorio más sencillo, en el que solo se utilizan dos archivos: “web.php” en el que se define qué acción se ejecuta en respuesta de cada llamada a través de la Web, y “api.php”, en el que de igual manera, se define qué acción se ejecuta en respuesta de cada llamada, pero a través de la aplicación móvil.

Si bien se explica la estructura de directorios de un proyecto en Laravel, en cuanto a la creación de un proyecto con el *framework* IONIC, la estructura es la siguiente:

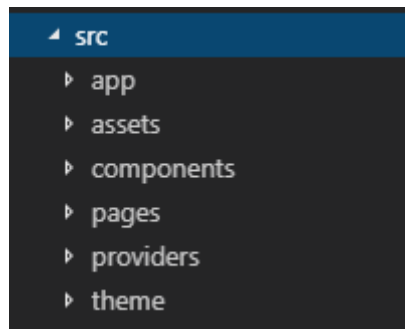


Figura 4.11 - Estructura de directorios del proyecto en IONIC.

En el caso de la estructura de directorios de IONIC, “src” es la carpeta más importante, y en la que se encuentra la mayor parte del trabajo, ya que contiene todos los archivos en donde se definen las pantallas, el estilo y el comportamiento que tendrá la aplicación. Por lo tanto, los directorios de mayor importancia que se deben mencionar son: “app”, “pages” y “providers”. El contenido de dichos, directorios, se explican a continuación:

- **Directorio “app”:** Contiene todos los archivos necesarios para iniciar la aplicación. Entre los más importantes están: “app.component.ts” también llamado componente raíz, el cual es el primer componente que se carga en la aplicación; y “app.module.ts”, que contiene las declaraciones de todos los componentes que se utilizan en la aplicación; cualquier componente o provider que se crea, debe estar declarado en dicho archivo para que se pueda utilizar (Ver Figura 4.12).

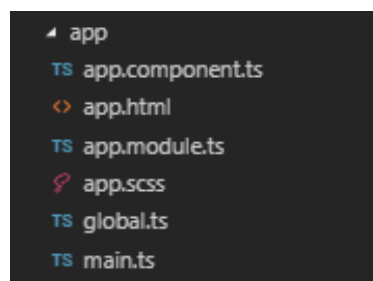


Figura 4.12 – App folder.

- **Directorio “pages”:** En el que se alojan todas las páginas que contenga la aplicación, es decir, una vista o una pantalla. Dentro de cada carpeta de una página, se encuentran tres archivos: un archivo .html, el cual contiene la

plantilla html de la página, un archivo .scss, donde se modifica el estilo de los componentes de la página, y un archivo .ts que es un archivo typescript, el cual contiene el controlador de la página, que es en donde se define el comportamiento de la misma (Ver Figura 4.13).

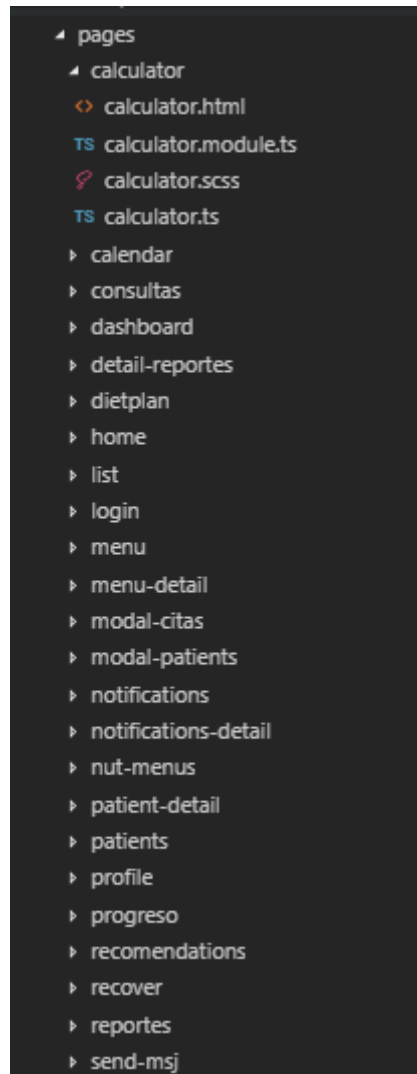


Figura 4.13 – Pages folder.

- **Directorio “providers”:** Comprende los archivos que se utilizan para el manejo de datos, como su inserción o extracción de la base de datos.

Además, puede ser utilizado para encapsular código común a muchos lugares de la aplicación, en lugar de repetirlo, y se puede aislar en un mismo lugar e inyectarlo en donde se necesite (Ver Figura 4.14).

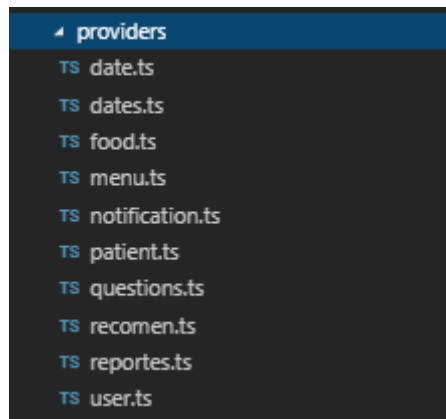


Figura 4.14 – Providers folder

4.1.4. Fase de pruebas

Mediante la fase de prueba, se concluye con el ciclo de desarrollo de la aplicación Web/móvil, y en donde se busca evaluar el comportamiento de la misma, para determinar el cumplimiento de todos los requerimientos establecidos en las historias de usuario, y para lo cual, se utilizan las pruebas de aceptación.

El detalle de la aplicación de dichas pruebas, se exhibe a continuación.

Las pruebas de aceptación son realizadas junto con los clientes (médicos nutricionistas y pacientes).

La idea de dichas pruebas, es comprobar el cumplimiento de los requerimientos obtenidos en las historias de usuario. Las pruebas se realizan por cada sección de cada módulo (los cuales son: el módulo de nutricionistas, y el módulo de pacientes), hasta que finalmente los clientes aprueban las funcionalidades necesarias para el módulo.

A continuación, se describen en detalle cada una de las pruebas de aceptación.

Tabla 15: Prueba de aceptación – Módulo nutricionista: Mis menús.

N.º Prueba	1
Módulo	Nutricionistas
Sección	Mis menús
Funcionalidades	Consulta de menús en la aplicación móvil
	Asignación de Menú
Problemas u Observaciones	En dicha prueba se aprueban las dos funcionalidades. Sin embargo, en la asignación del menú, hubo inconvenientes, ya que no se contemplaban todos los casos, como por ejemplo, que un mismo menú podría estar asignado a varios pacientes. Dicho inconveniente es solucionado satisfactoriamente.

Tabla 16: Prueba de aceptación – Módulo nutricionista: Mis citas.

N.º Prueba	2
Módulo	Nutricionistas
Sección	Mis citas
Funcionalidades	Consulta de calendario de citas
	Crear Cita
	Reprogramación de citas
Problemas u Observaciones	Se aprueban todas las funcionalidades en dicha prueba. Sin embargo, se realizan pequeños ajustes solicitados por el nutricionista, en cuanto a la información enviada como notificación al paciente al reprogramar la cita.

Tabla 17: Prueba de aceptación – Módulo nutricionista: Reportes.

N.º Prueba	3
Módulo	Nutricionistas
Sección	Reportes
Funcionalidades	Consulta de reportes
	Eliminación de reporte
Problemas u Observaciones	Se aprueban todas las funcionalidades de dicha prueba. No obstante, el nutricionista sugiere algunas preguntas que podrían ser añadidas al reporte diario del paciente, y las cuales fueron agregadas.

Tabla 18: Prueba de aceptación – Módulo nutricionista: Recomendaciones.

N.º Prueba	4
Módulo	Nutricionistas
Sección	Recomendaciones
Funcionalidades	Consulta de recomendaciones
	Asignación de recomendaciones
	Eliminación de recomendaciones
Problemas u Observaciones	En dicha prueba, el nutricionista sugiere establecer dos tipos de recomendaciones, las cuales serían generales y específicas para cada paciente, proporcionando las recomendaciones generales. Dicha sugerencia es tomada en cuenta, aprobando así todas las funcionalidades.

Tabla 19: Prueba de aceptación – Módulo nutricionista: Notificaciones.

N.º Prueba	5
Módulo	Nutricionistas
Sección	Notificaciones
Funcionalidades	Consulta de notificaciones
	Envío de mensajes a pacientes
	Eliminación de notificaciones
Problemas u Observaciones	Todas las funcionalidades son aprobadas sin inconvenientes

Tabla 20: Prueba de aceptación – Módulo nutricionista: Consultas.

N.º Prueba	6
Módulo	Nutricionistas
Sección	Consultas
Funcionalidades	Visualización de indicadores
Problemas u Observaciones	Algunos de los indicadores, son reflejados en gráficos que el nutricionista considera que no son los adecuados, y por lo tanto, se modifican para su mejor visualización, y su posterior aprobación.

Tabla 21: Prueba de aceptación – Módulo de Pacientes: Mi perfil.

N.º Prueba	7
Módulo	Pacientes
Sección	Mi perfil
Funcionalidades	Consulta de información de perfil
Problemas u Observaciones	En dicha prueba, la funcionalidad es aprobada. Sin embargo, algunos pacientes solicitan ver un poco de su información médica, la cual es añadida a su perfil.

Tabla 22: Prueba de aceptación – Módulo de Pacientes: Mi plan.

N.º Prueba	8
Módulo	Pacientes
Sección	Mi plan
Funcionalidades	Consulta de información de plan dieto-terapéutico
Problemas u Observaciones	La funcionalidad es aprobada sin inconvenientes

Tabla 23: Prueba de aceptación – Módulo de Pacientes: Mis menús.

N.º Prueba	9
Módulo	Pacientes
Sección	Mis menús
Funcionalidades	Consulta de información de menús
Problemas u Observaciones	La funcionalidad es aprobada sin inconvenientes

Tabla 24: Prueba de aceptación – Módulo de Pacientes: Mis citas.

N.º Prueba	10
Módulo	Pacientes
Sección	Mis citas
Funcionalidades	Consulta de calendario de cita
	Solicitud de cambio de cita
Problemas u Observaciones	Todas las funcionalidades son aprobadas sin inconvenientes

Tabla 25: Prueba de aceptación – Módulo de Pacientes: Mis recomendaciones.

N.º Prueba	11
Módulo	Pacientes
Sección	Mis recomendaciones
Funcionalidades	Consulta de recomendaciones
Problemas u Observaciones	La funcionalidad es aprobada sin inconvenientes

Tabla 26: Prueba de aceptación – Módulo de Pacientes: Mi progreso.

N.º Prueba	12
Módulo	Pacientes
Sección	Mi progreso
Funcionalidades	Consulta de progreso
Problemas u Observaciones	La funcionalidad es aprobada sin inconvenientes

Tabla 27: Prueba de aceptación – Módulo de Pacientes: Contactar a Nutricionista.

N.º Prueba	13
Módulo	Pacientes
Sección	Contactar a Nutricionista
Funcionalidades	Consulta de datos de contacto del nutricionista
	Envío de mensajes a nutricionista
Problemas u Observaciones	Todas las funcionalidades son aprobadas sin inconvenientes

Tabla 28: Prueba de aceptación – Módulo de Pacientes: Notificaciones.

N.º Prueba	14
Módulo	Pacientes
Sección	Notificaciones
Funcionalidades	Consulta de notificaciones
	Enviar mensaje a nutricionista
Problemas u Observaciones	Todas las funcionalidades son aprobadas sin inconvenientes

Tabla 29: Prueba de aceptación – Módulo de Pacientes: *Dashboard*.

N.º Prueba	15
Módulo	Pacientes
Sección	<i>Dashboard</i>
Funcionalidades	Realización de reporte de avance diario
Problemas u Observaciones	La funcionalidad es aprobada sin inconveniente

Tabla 30: Prueba de aceptación – Módulo de Pacientes: Calculadora de Calorías.

N.º Prueba	16
Módulo	Pacientes
Sección	Calculadora de Calorías
Funcionalidades	Cálculo de calorías
Problemas u Observaciones	La funcionalidad es aprobada, sin embargo, los pacientes consideran que la lista de alimentos que se proporciona es escasa. Por lo tanto, se busca una amplia lista de alimentos con su respectiva cantidad de calorías, y la cual es añadida posteriormente.

El principal interés al realizar dicha aplicación, es cumplir con cada uno de los requisitos descritos por los clientes; uno de los cuales, consiste en realizar una aplicación de fácil acceso, y cuyo flujo de trabajo sea simple e intuitivo.

Seguidamente, se presenta el flujo del funcionamiento de la aplicación Web/móvil, para los profesionales y pacientes en el área de nutrición.

4.2. Flujo de la aplicación Web/móvil como usuario paciente

A continuación, se describe el flujo de la aplicación Web/móvil, en el módulo de pacientes, y sus respectivas interfaces asociadas a cada paso.

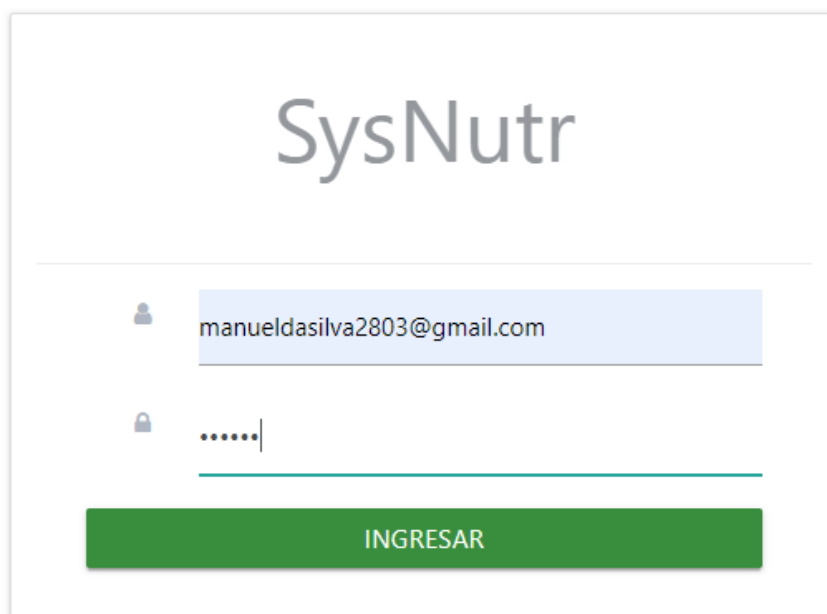


Figura 4.15 - Pantalla de acceso, aplicación Web.

En la Figura 4.15, se muestra la pantalla diseñada para tener acceso a la aplicación Web. Dicha interfaz, fue diseñada para la aplicación Web SysNutr, en donde se solicita, colocar el correo electrónico y la contraseña, para que el profesional y los pacientes en el área de nutrición, pueden acceder a la aplicación.

Adicionalmente, se proporciona la pantalla de acceso que fue diseñada para la aplicación móvil, y que se puede observar en la siguiente figura:



Figura 4.16 – Pantalla de acceso, aplicación móvil.

Al ingresar en la aplicación, se puede observar la interfaz descrita a continuación:

Interfaz “Dashboard” del módulo de pacientes

En la Figura 4.17 se observa la pantalla de la aplicación Web, para que el paciente del área de nutrición pueda elegir la acción que desea realizar, además de visualizar sus recomendaciones generales, calendario de cita, y una sección en la que introduce los datos de su avance diario.

Figura 4.17 – Dashboard del módulo de pacientes, aplicación Web.

En la parte superior izquierda, se encuentra la identificación del usuario reflejando así: el nombre, correo e imagen del mencionado. Y en la parte inferior izquierda, se encuentra el menú de navegación, en donde el paciente puede seleccionar la operación que desea realizar en un momento dado.

Al igual que en la aplicación Web, se diseña una interfaz “Dashboard” para aplicación móvil, y la cual puede observarse en la Figura 4.18.

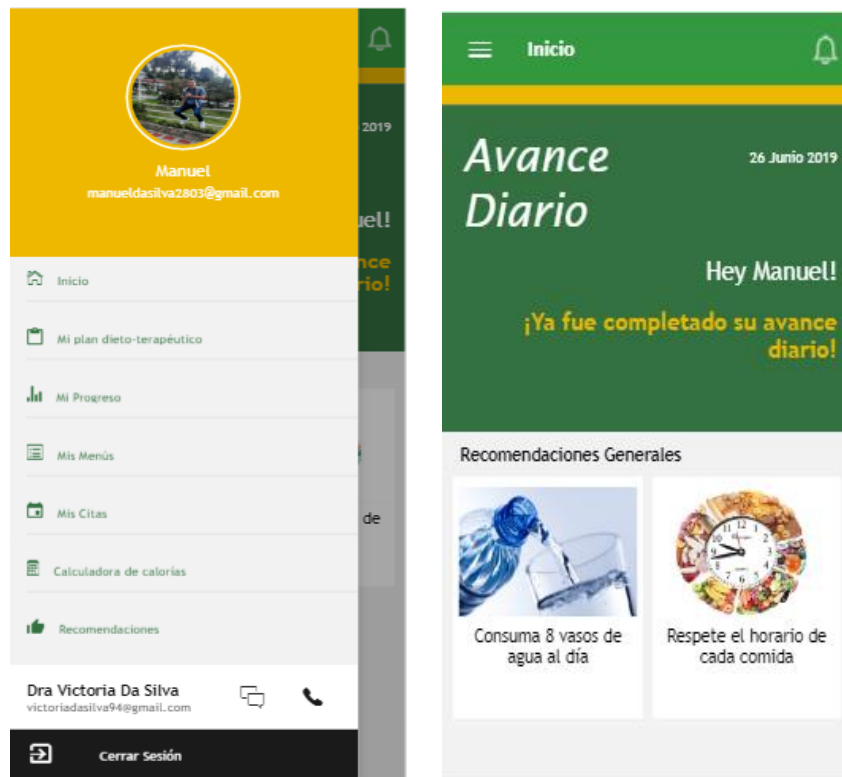


Figura 4.18 – Dashboard del módulo de paciente, aplicación móvil.

Para continuar con el flujo de las actividades realizadas por el paciente, una vez iniciada la sesión, el mismo puede consultar información de su perfil, su plan dieto-terapéutico, sus menús, su calendario de cita, sus recomendaciones, su progreso, y sus mensajes o notificaciones. A su vez, puede contactar a su nutricionista enviando un mensaje.

Posteriormente, se observan y explican cada una de las interfaces de las actividades mencionadas.

Interfaz “Mi Perfil”

En las Figuras 4.19 y 4.20, se presenta la interfaz denominada “Mi Perfil”, en la cual se observa la información básica del paciente, la información de contacto, y en la aplicación Web, la información médica más resaltante.



The screenshot shows a web interface for a patient's profile. On the left is a green sidebar menu with options: Notificaciones, Dashboard, Mi Perfil, Mi Plan, Mis Menús, Mis Citas, Mis Recomendaciones, Mi Progreso, Contactar a Nutricionista, and CERRAR SESIÓN. The main content area is divided into three sections: Información de básica, Información de contacto, and Información médica.

Información de básica		
Ci: V-27045457	Nombres: Manuel	Apellidos: Da Silva De Amoreira
Género: Masculino	Religión: Católico	Profesión: Estudiante
Condición Socioeconómica: Media	Fecha de Nacimiento: 28 Marzo, 2000	

Información de contacto		
Email: manuelsilva2803@gmail.com	Teléfono: 04125580343	Dirección: Calle Colombia cruce con quinta Av. Edificio Amoreira, Cata.

Información médica		
Edad: 19 años	Antecedentes Personales: Ninguna	Antecedentes Familiares: Cancer, Afecciones cardiacas, Diabetes
Peso: 65.00 kg	Diagnóstico Nutricional: Se observa conciente, tranquilo, orientado, mucosas orales y tegumentos con adecuada hidratación. Se dedica a estudiar. Refiere ser persona de recursos económicos medios. Antecedentes heredo-familiares: DMII (abuela paterno) y de HTA (abuelo materno).	Diagnóstico Médico: Paciente masculino de 19 años de edad a interconsulta de nutrición referida de consulta externa a control de peso. Se observa conciente, tranquilo, orientado, mucosas orales y tegumentos con adecuada hidratación. Se dedica a estudiar. Refiere ser persona de recursos económicos medios. Antecedentes heredo-familiares: DMII (abuela paterno) y de HTA (abuelo materno).
Estatura: 170.00 cms	Problemas Nutricionales: No tiene problemas nutricionales.	Fecha de Admisión: 15 Marzo, 2019

Figura 4.19 - Perfil del paciente, aplicación Web.



The screenshot shows the mobile application interface for the patient's profile. It features a green header with a back arrow and the word 'Perfil'. Below the header is a yellow bar with the patient's name 'Manuel', age '19 años', and weight '65.00 Kg'. The main content is divided into two sections: Información Básica and Información de Contacto.

Información Básica	
Nombre(s)	Manuel
Apellido(s)	Da Silva De Amoreira
Edad	19 años
Estatura	0.00 mts

Información de Contacto	
Correo	manuelsilva2803@gmail.com
Teléfono	04125580343

Figura 4.20 - Perfil del paciente, aplicación móvil.

Interfaz “Mi Plan”

Al hacer clic en la opción del menú “Mi Plan”, el paciente puede observar la información sobre su plan dieto-terapéutico, tales como: su requerimiento calórico total (es decir, lo que debe consumir durante todo el día), además se encuentra la fórmula dietética, la cual está compuesta por los macronutrientes (proteínas, lípidos y carbohidratos), el porcentaje de los macronutrientes que se aportan, las kilocalorías y los gramos que representan.

Posteriormente, se muestran los intercambios correspondientes a: lácteos, vegetales, frutas, cereales, carnes y grasas, con sus respectivas proteínas, lípidos y carbohidratos. Para comprender mejor, es necesario conocer el concepto de intercambio, el cual es conjunto de alimentos unificados a un mismo valor de energía y macronutrientes, de manera que, aquellos alimentos del mismo grupo, pueden ser intercambiables entre sí, sin suponer una variación significativa en dichos valores, según (González, 2018).

En las Figuras 4.21 y 4.22, se observan las interfaces de la información del plan dieto-terapéutico.

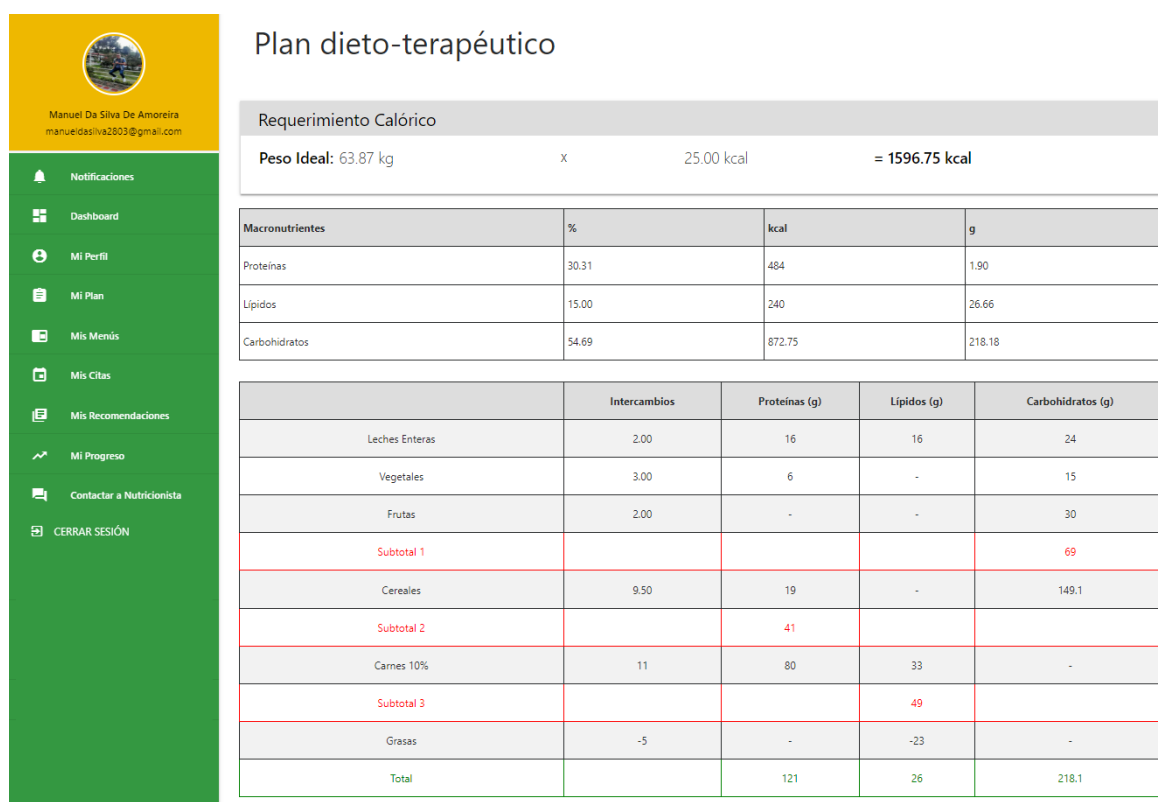


Figura 4.21 - Plan dieto-terapéutico, aplicación Web.

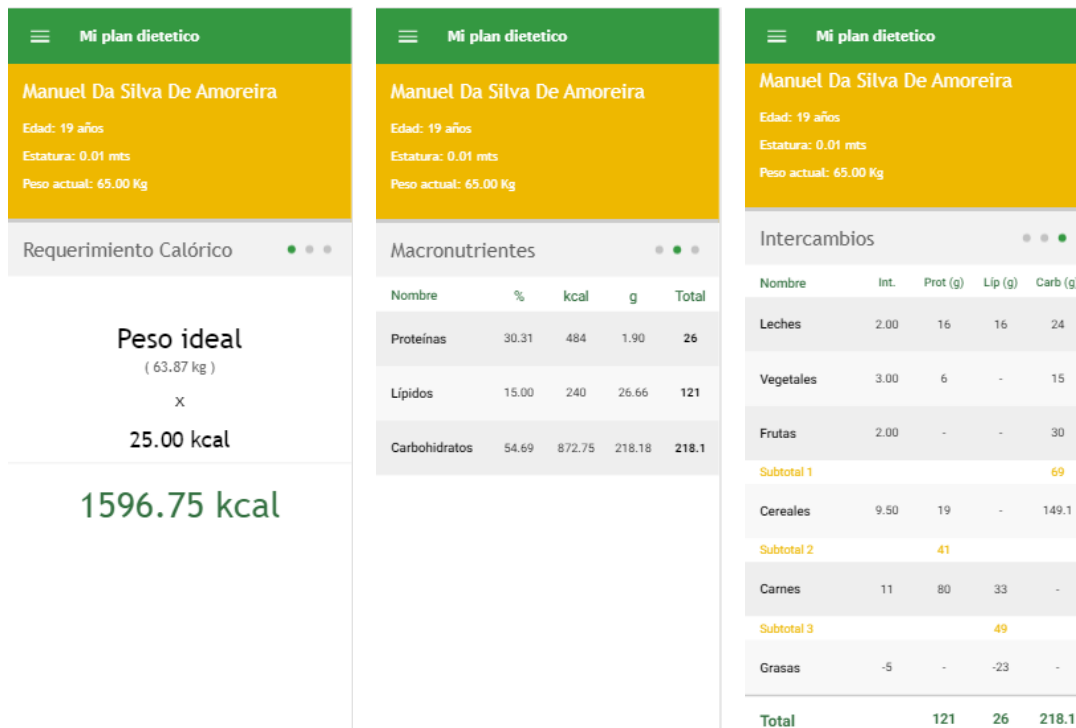


Figura 4.22 - Plan dieto-terapéutico, aplicación móvil.

Interfaz “Mis menús” del módulo de pacientes

El nutricionista al crear el plan dieto-terapéutico, le asigna uno o varios menús al paciente, según sus requerimientos nutricionales, y los cuales, se listan en la interfaz de “Mis menús”, a través de las Figuras 4.23 y 4.24, descritas a continuación:

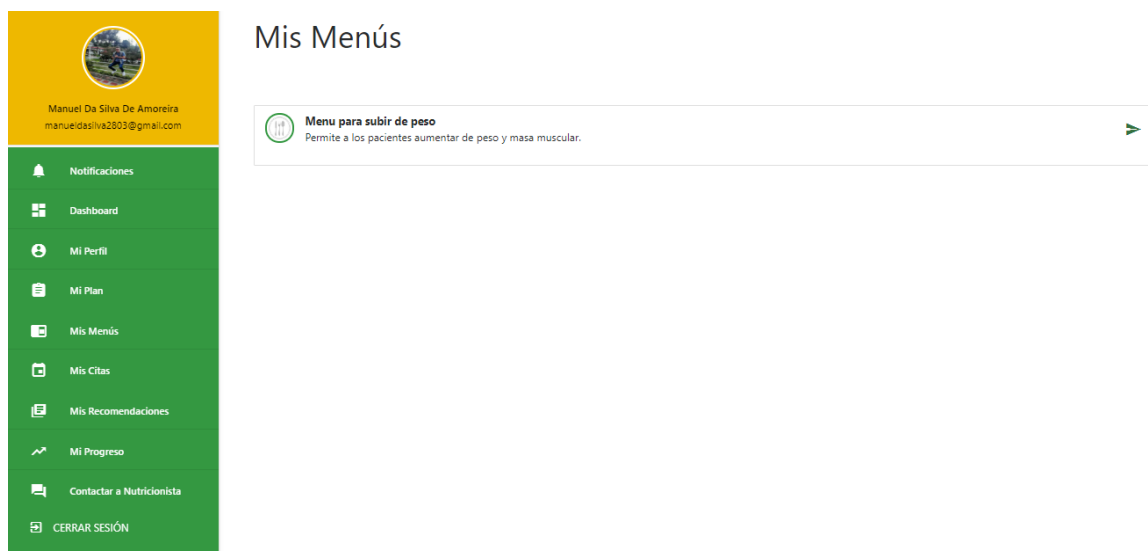


Figura 4.23 - Lista de menús del módulo de pacientes, aplicación Web.

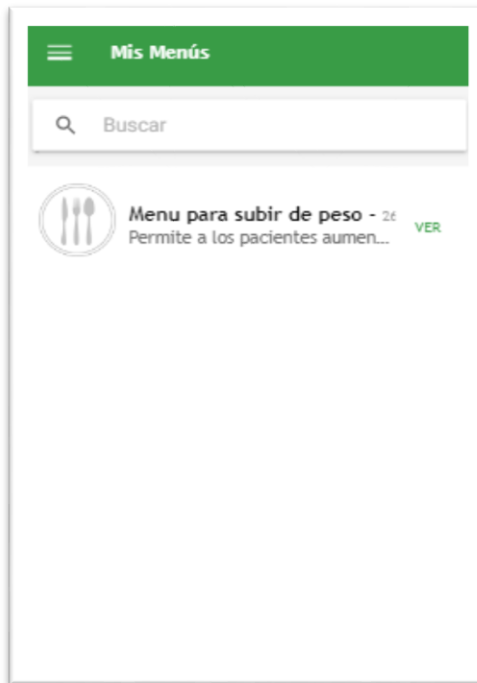



Figura 4.23 - Lista de menús del módulo de pacientes, aplicación móvil.

Cada uno de los menús tiene su respectivo detalle en el que se observa el nombre, la descripción y los intercambios repartidos entre cada comida tales como desayuno, almuerzo, cena y meriendas. En las Figuras 4.24 y 4.25, se puede observar las interfaces del detalle de un menú.



Manuel Da Silva De Amoreira
manuedasilva2803@gmail.com

- 🔔 Notificaciones
- 🏠 Dashboard
- 👤 Mi Perfil
- 📅 Mi Plan
- 📖 Mis Menús
- 📅 Mis Citas
- 📄 Mis Recomendaciones
- 📈 Mi Progreso
- 📧 Contactar a Nutricionista
- 🚪 CERRAR SESIÓN

Menú

26 Junio, 2019

Nombre del Menu: Menu para subir de peso **Descripción:** Permite a los pacientes aumentar de peso y masa muscular.

DESAYUNO		
Intercambios	Grupo	Comida
1.00	Leches Enteras	Leche entera
2.00	Almidones y Cereales	Muesli
1.00	Almidones y Cereales	Pan Sándwich blanco o integral
1.00	Carnes Gordas	Queso Amarillo tipo Gouda
1.00	Grasas	Aguacate
MERIENDA		
Intercambios	Grupo	Comida
2.00	Frutas	Mandarina
1.00	Almidones y Cereales	Platano
2.00	Grasas	Nueces

Figura 4.24 - Detalle de menú del módulo de pacientes, aplicación Web.

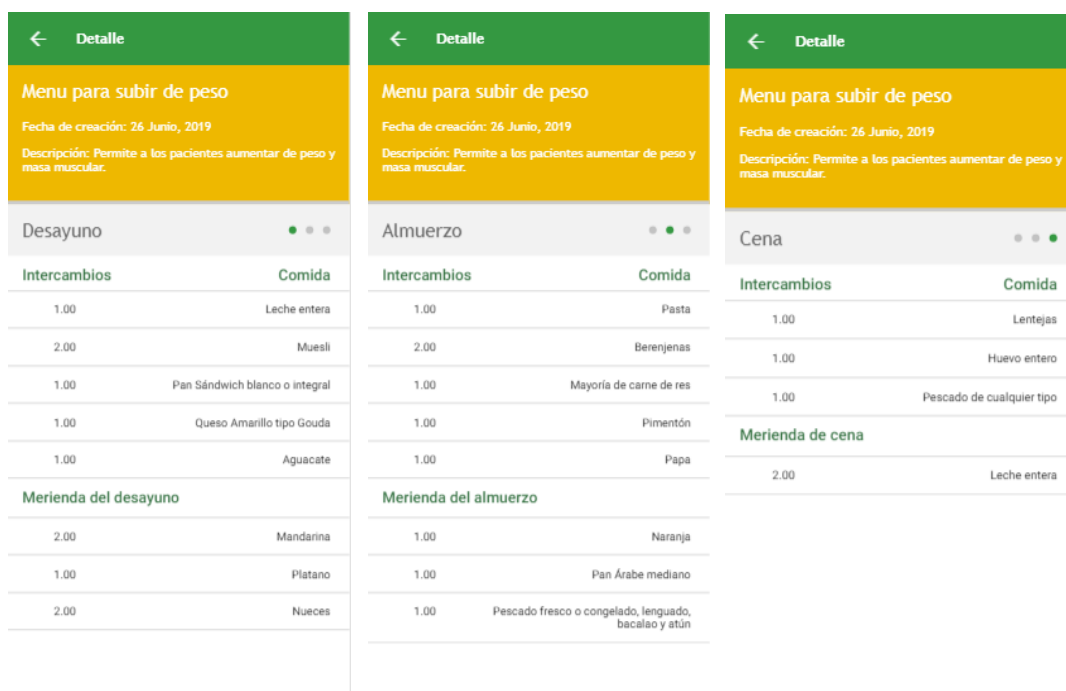


Figura 4.25 - Detalle de menú del módulo de pacientes, aplicación móvil.

Interfaz “Mis Citas” del módulo de pacientes

El médico nutricionista le establece una cita al paciente, a través de la aplicación Web, la cual automáticamente el paciente puede observar en su calendario de cita como se muestra en la Figura 4.26.

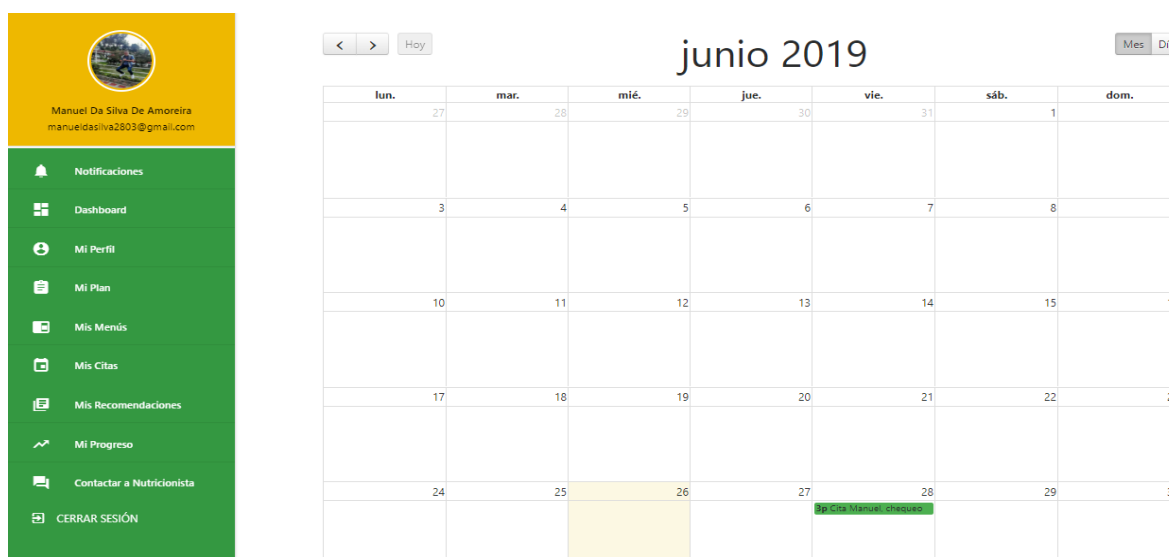


Figura 4.26 - Calendario de cita del paciente, aplicación Web.

En dicha interfaz, el paciente puede solicitar un cambio de cita a su nutricionista, realizando un clic en la cita que se muestra en el calendario, y en donde se despliega un formulario, en donde el paciente establece la fecha y hora que desea modificar la cita, como se observa en la Figura 4.27.

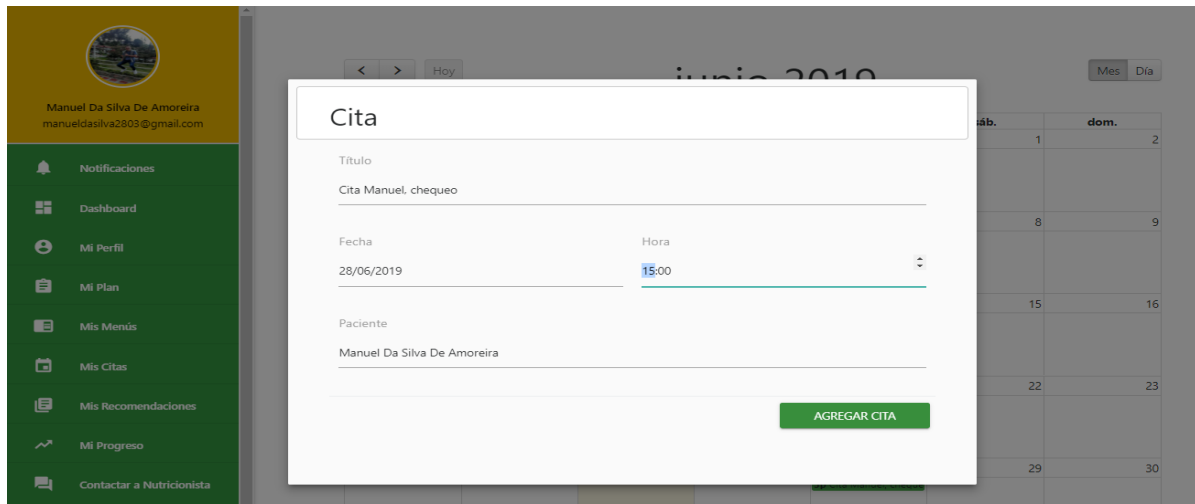


Figura 4.27 - Solicitud de cambio de cita, aplicación Web.

Al hacer clic en “Agregar cita”, se envía una notificación al nutricionista, de su solicitud de cambio de cita, y la cual, puede aceptar o rechazar.

En cuanto a la aplicación móvil, de igual modo se tiene el calendario con su cita, y dicho proceso de solicitud de cambio de cita, se realiza de la misma manera que la aplicación Web, a diferencia de que se presiona un botón denominado “Solicitar cambio”, y seguidamente se realiza el proceso descrito anteriormente. En las Figuras 4.28 y 4.29, se percibe lo descrito.

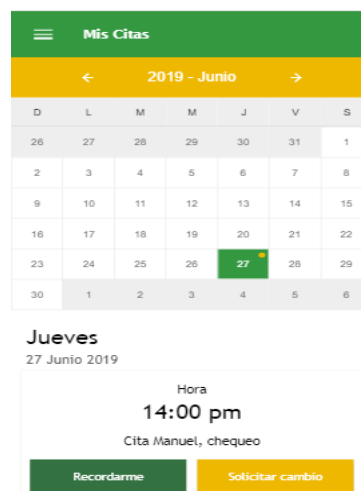


Figura 4.28 - Calendario de cita del paciente, aplicación móvil.

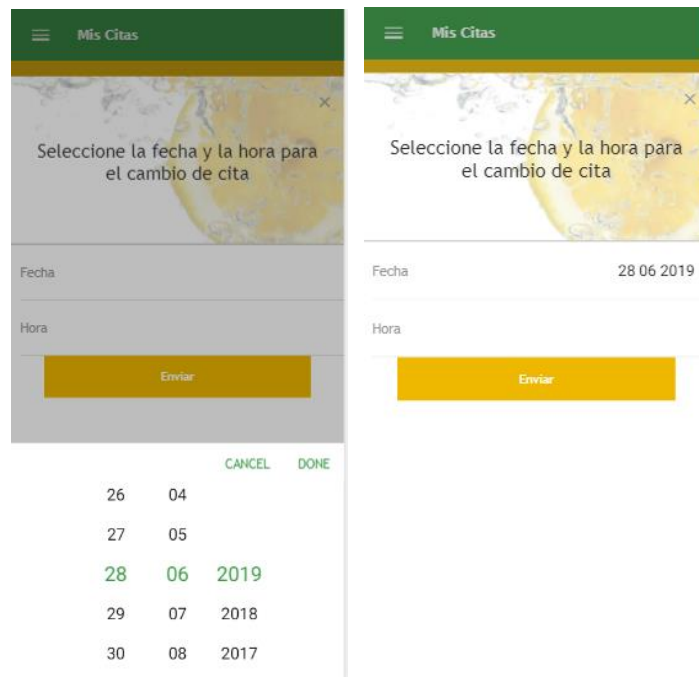


Figura 4.29 - Solicitud de cambio de cita, aplicación móvil.

Interfaz “Mis Recomendaciones” del módulo de pacientes

En la interfaz denominada “Mis Recomendaciones”, el paciente puede observar la lista de recomendaciones generales, y las específicas para él. En las Figuras 4.30 y 4.31, se muestran las interfaces de dicha lista de recomendaciones.



Figura 4.30 - Lista de recomendaciones del paciente, aplicación Web.

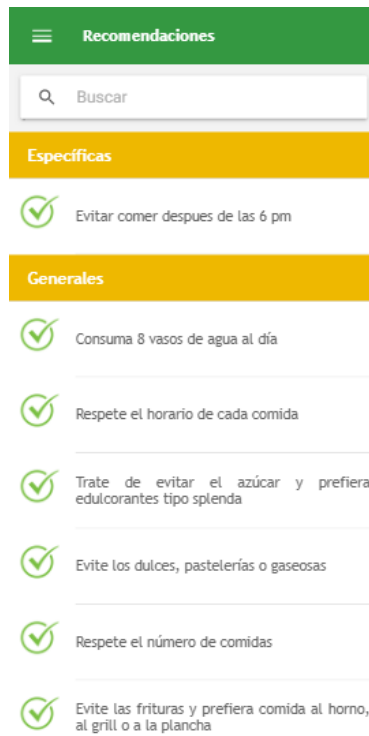


Figura 4.31 - Lista de recomendaciones del paciente, aplicación móvil.

Interfaz “Mi Progreso”

En la Figuras 4.32 y 4.33, se presenta lo que se observa en la interfaz denominada, “Mi progreso”, en la que se muestra el progreso del paciente, desde el comienzo de su tratamiento, dado su peso, su IMC, su área magra y área grasa.

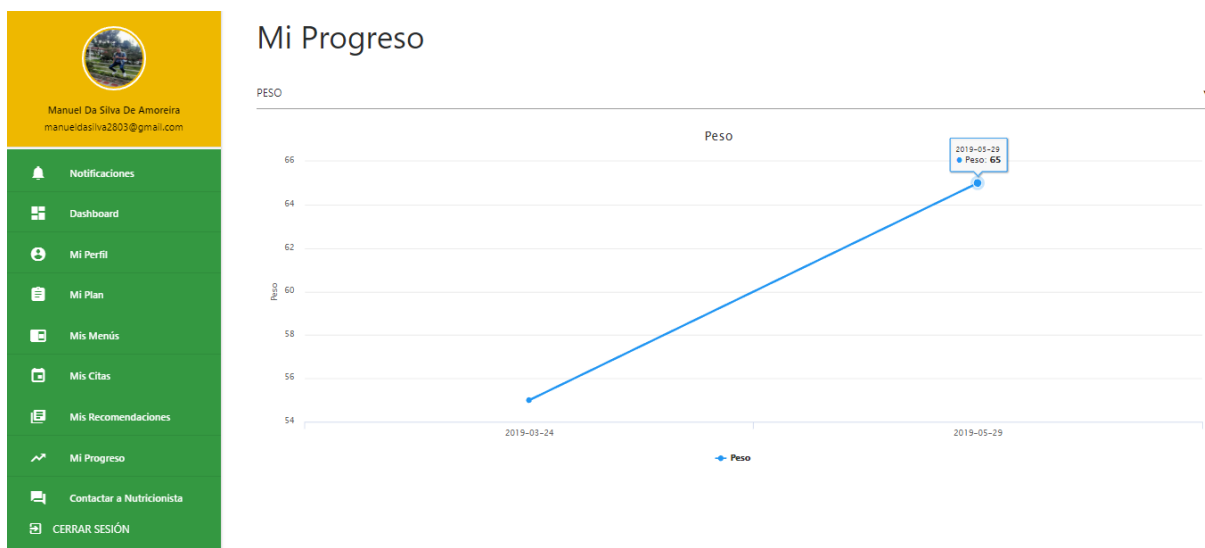


Figura 4.32 - Progreso del paciente según su peso, aplicación Web.



Figura 4.33 - Progreso del paciente según su peso, aplicación móvil.

Interfaz “Contactar a Nutricionista”

En dicha interfaz, el usuario paciente tiene la opción de contactar a su nutricionista, a través del envío de mensajes, al hacer clic en la opción del menú “Contactar a Nutricionista”, en la que se despliegan los datos de contacto del nutricionista, y un formulario en el que debe colocar el asunto y el mensaje que desea enviar. En la Figura 4.34, se percibe la interfaz de dicho proceso en la aplicación Web.

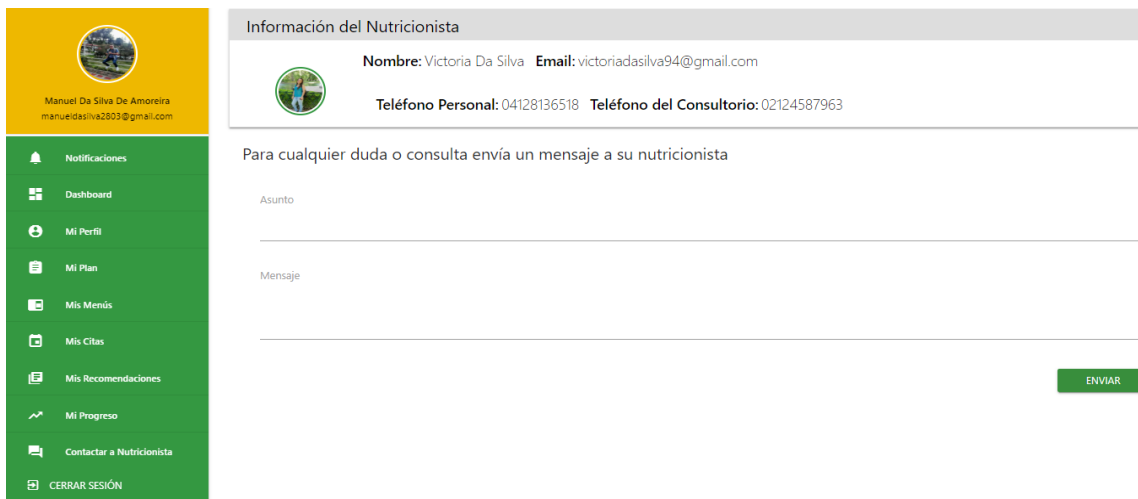



Figura 4.34 - Contactar a su nutricionista, aplicación Web.

Dicho proceso, se realiza de igual modo en la aplicación móvil, al hacer clic en el icono  , originando que se despliegue la interfaz con el formulario para el envío del mensaje, descrita en la Figura 4.35.

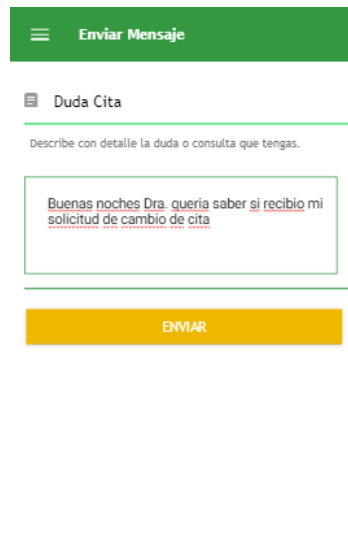



Figura 4.35 - Contactar a su nutricionista, aplicación móvil.

Por otro lado, el paciente a través de la aplicación móvil, también puede contactar a su nutricionista, llamando a sus números telefónicos, al hacer clic en el icono  .

Interfaz “Notificaciones” del módulo de pacientes

En las Figuras 4.36 y 4.37, se presentan las interfaces de la opción denominada “Notificaciones”, en la cual el paciente puede observar los mensajes enviados y recibidos, ya sean de solicitud de cambio de cita, o los mensajes de respuesta del nutricionista a cualquier solicitud, duda o consulta que haya sido enviada.

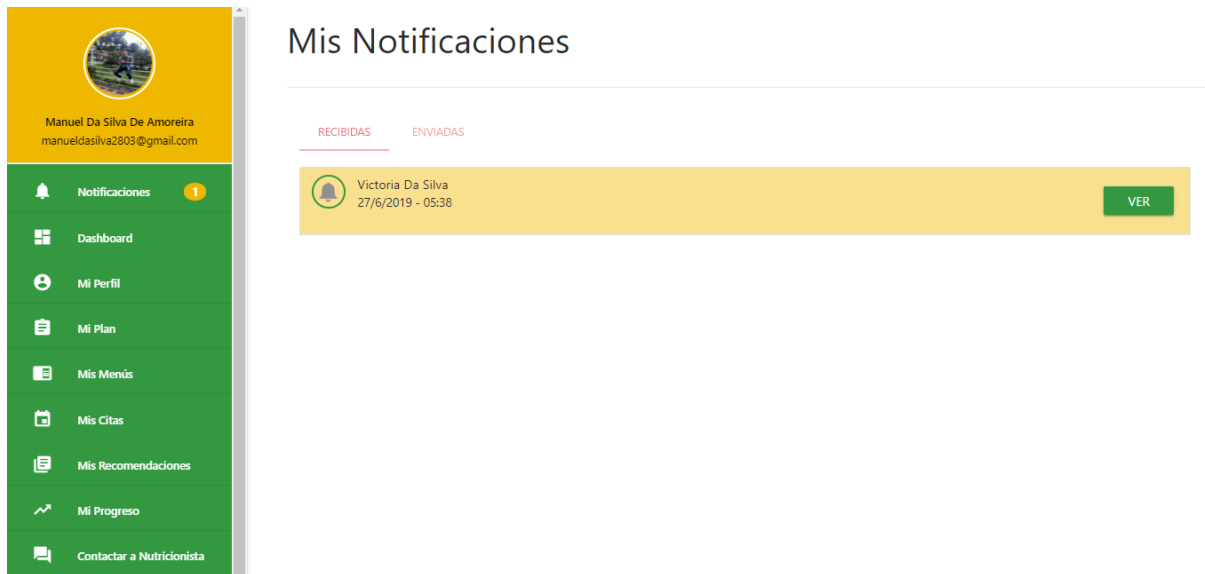


Figura 4.36 - Bandeja de entrada y salida del paciente, aplicación Web.

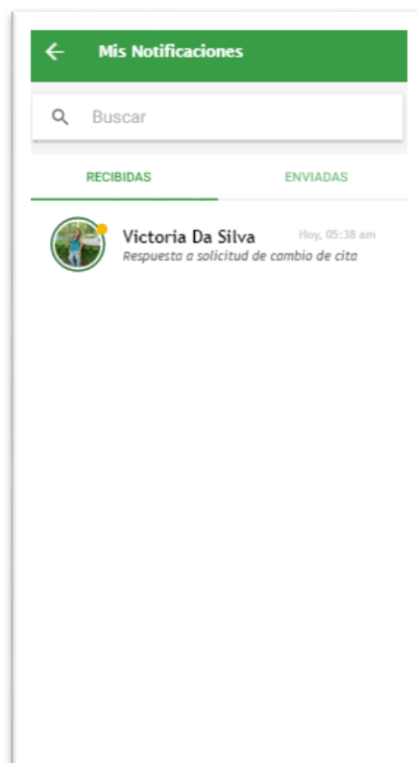


Figura 4.37 - Bandeja de entrada y salida del paciente, aplicación móvil.

Asimismo, cada notificación tiene su respectivo detalle, en la cual se muestran datos como: el nombre de usuario que envía y recibe el mensaje, la fecha y hora de envío, y el contenido del mensaje; y en caso de que la notificación sea por solicitud

de cambio de cita, adicionalmente, se muestra la fecha y hora del cambio de cita solicitado. En la figura 4.38, puede percibirse, detalles inherentes a las notificaciones.



Figura 4.38 - Detalle de notificación de un paciente, aplicación Web.

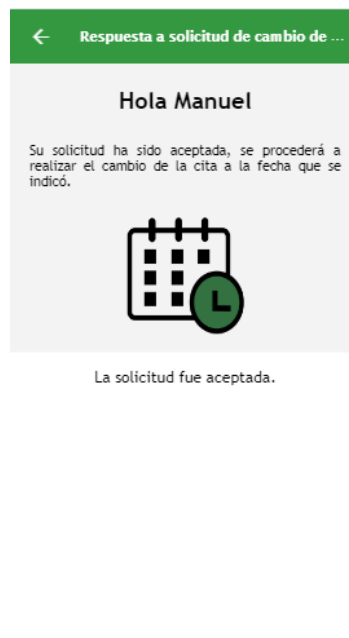


Figura 4.39 - Detalle de notificación de un paciente, aplicación móvil.

Interfaz “Calculadora”

Al hacer clic en la opción del menú de navegación “Calculadora de calorías”, se observa una interfaz, en la cual, el paciente puede calcular las kilocalorías que va ingerir en un determinado plato. El proceso para realizar dicho cálculo, es el siguiente:

El paciente selecciona la opción denominada “Por plato”; posteriormente en la siguiente interfaz, el mismo debe seleccionar la opción “Agregar alimento”, desplegándose seguidamente, la interfaz para agregar dicho alimento; y en la cual, se deben completar diversos datos, tales como: la categoría y subcategoría en la que se encuentra el alimento, e indicar la cantidad en porciones, o medida de gramos o mililitros. En la Figura 4.40, se pueden observar las interfaces asociadas a dicho proceso.

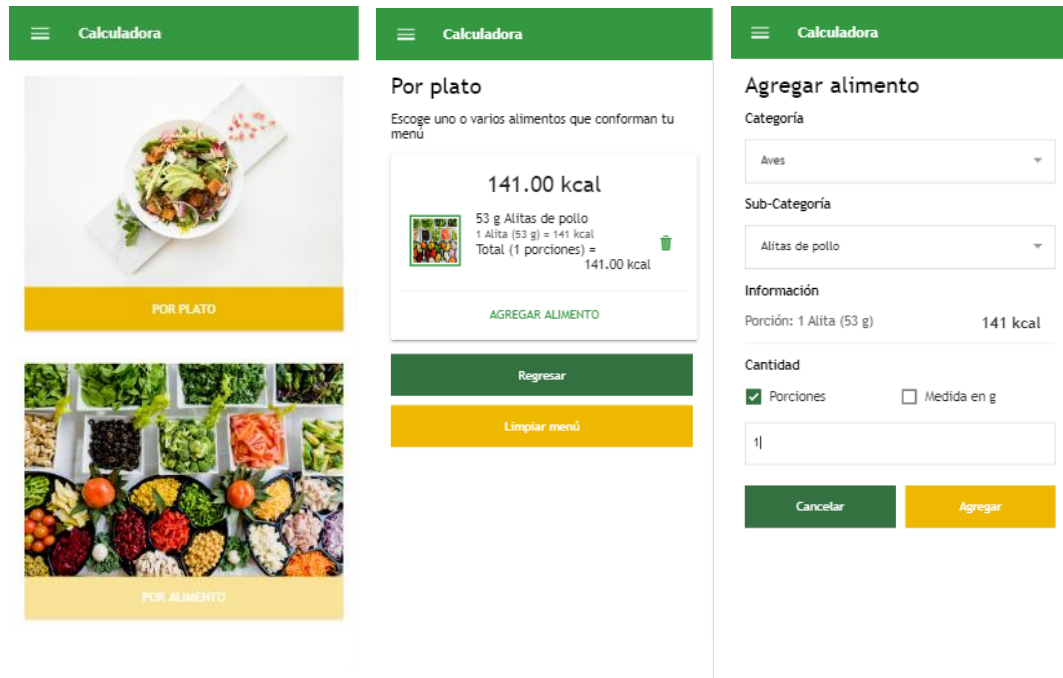


Figura 4.40 – Calculadora de kilocalorías.

Si bien se explica en detalle, el flujo de la aplicación Web/móvil como usuario paciente, también es necesario describir el detalle del flujo de la aplicación Web/móvil como usuario nutricionista, y en el cual, para el caso del flujo de la aplicación Web, solo se detallarán las funciones e interfaces desarrolladas en el presente proyecto.

4.3. Flujo de la aplicación Web/móvil como usuario nutricionista

A continuación, se describe el flujo de la aplicación Web/móvil en el módulo de nutricionistas, y sus respectivas interfaces asociadas a cada paso.

Interfaz “Dashboard” del módulo de nutricionista

Una vez iniciada la sesión en la aplicación Web/móvil, como usuario nutricionista a través de la misma pantalla de acceso del módulo de pacientes, se observa la interfaz “Dashboard”, la cual en el caso de la aplicación Web, la misma ya se encontraba diseñada, y solo se realizaron cambios a nivel de estilo y en el menú de navegación (Ver Figura 4.41).



Figura 4.41 - Dashboard del módulo de nutricionista, aplicación Web.

En la parte superior izquierda de la presente figura, se muestra la identificación del usuario (nombre, correo e imagen del mismo). Y en la parte inferior izquierda, se el menú de navegación, donde el nutricionista selecciona la operación que desea realizar. Todo lo dicho se observa igualmente, en el módulo de pacientes, sin embargo, en dicha interfaz se muestran, además, los últimos movimientos realizados con anterioridad, referidos a las últimas actividades realizadas por el nutricionista, en relación a sus pacientes, menús y alimentos.

En el caso de la aplicación móvil, la interfaz “Dashboard” comprende un grupo de opciones que el nutricionista puede seleccionar, de acuerdo a la actividad que desea realizar en un momento dado. Dicha interfaz, se puede observar en la siguiente figura:

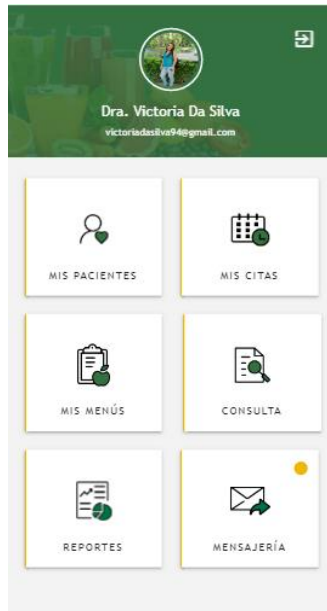


Figura 4.42 - Dashboard del módulo de nutricionista, aplicación móvil.

Continuando con el flujo de las actividades realizadas por el nutricionista, una vez iniciada la sesión, el mismo puede consultar información, tales como: su lista de pacientes, sus menús plantilla, su calendario de citas, los reportes de avance diario de cada paciente, y sus mensajes o notificaciones. A su vez, puede responder los mensajes enviados por sus pacientes, visualizar consultas de forma gráfica, asignar menús a pacientes y crear recomendaciones.

Posteriormente, se observan y explican cada una de las interfaces de las actividades mencionadas.

Interfaz “Consultas”

El usuario nutricionista, al hacer clic en la opción “Consultas”, puede visualizar indicadores de forma gráfica, tales como: Cantidad de pacientes hombres y mujeres, cantidad de pacientes con antecedentes familiares (diabetes, obesidad, afecciones cardíacas, entre otras), cantidad de pacientes con obesidad, entre 18 y 50 años, y la cantidad de pacientes con desnutrición entre 18 y 50 años. En las Figuras 4.43 y 4.44, se observa, para las interfaces de la aplicación Web/móvil, uno de los indicadores mencionados.

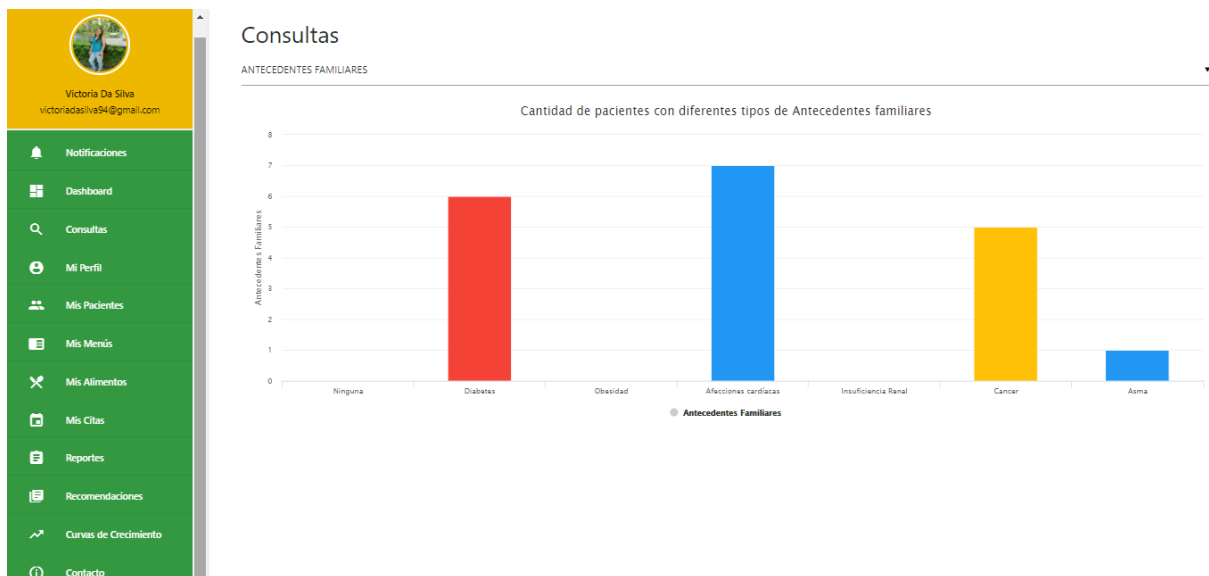


Figura 4.43 - Indicador de antecedentes familiares, aplicación Web.

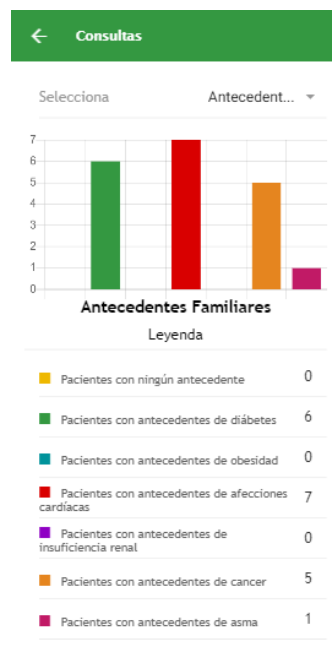


Figura 4.44 - Indicador de antecedentes familiares, aplicación móvil.

Interfaz “Mis menús” del módulo de nutricionista

El nutricionista puede crear menús y plantillas, para luego asignarlos a pacientes en específico, de acuerdo a sus requerimientos nutricionales. De igual modo que en el módulo de pacientes, la lista de dichos menús se puede observar en las interfaces de “Mis menús” de la aplicación web/móvil, como se muestra en las Figuras 4.45 y 4.46.

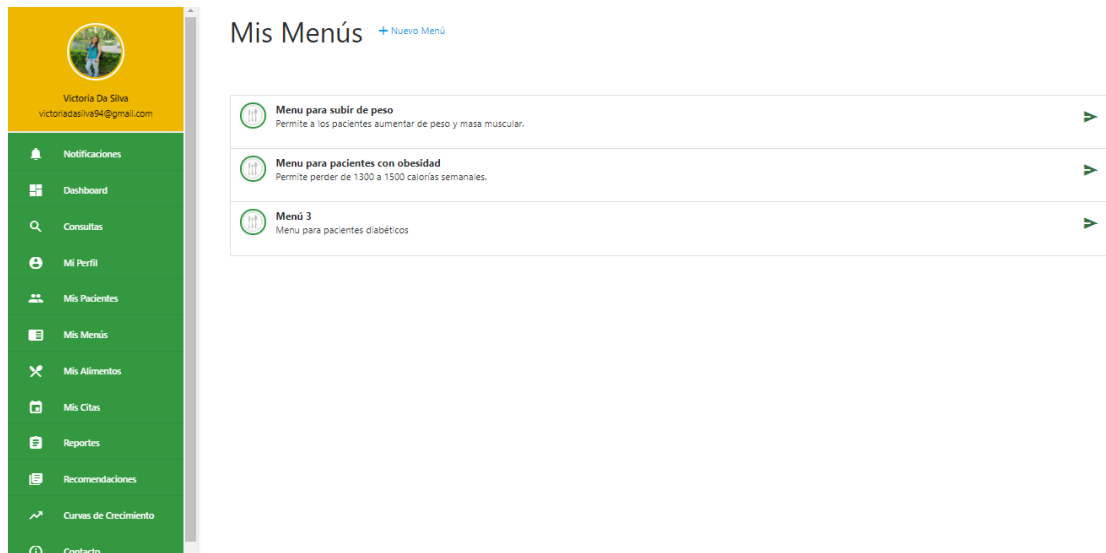


Figura 4.45 - Lista de menús del módulo del nutricionista, aplicación Web.

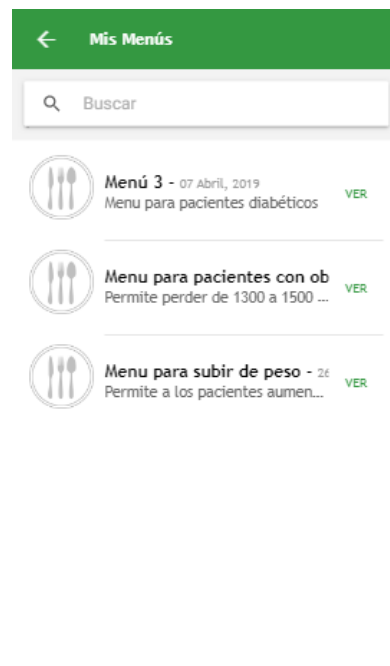
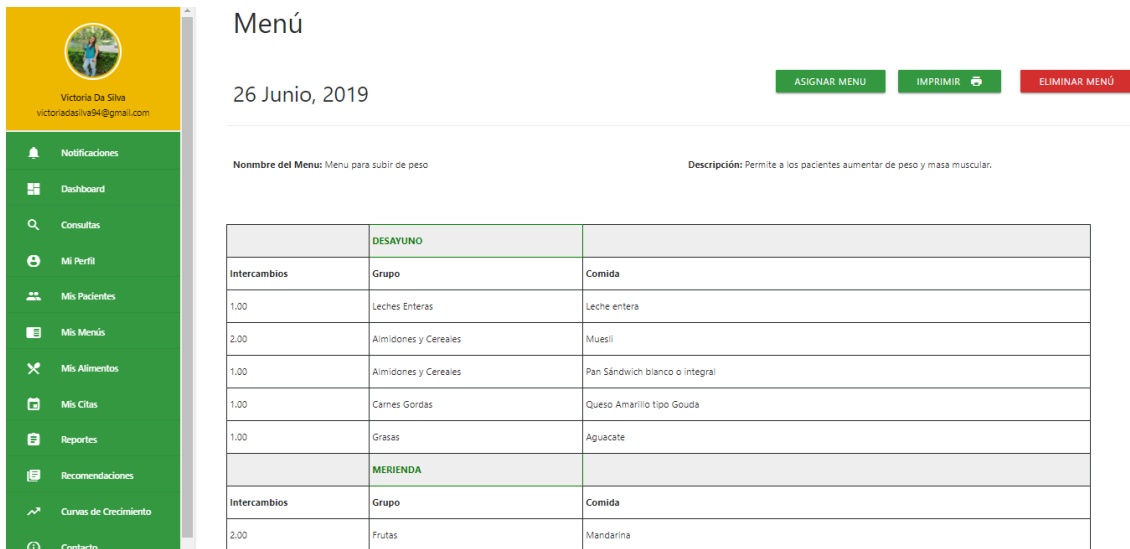


Figura 4.46 Lista de menús del módulo del nutricionista, aplicación móvil.

Cada uno de dichos menús posee su respectivo detalle, en el que se observa el nombre, la descripción y los intercambios repartidos entre cada comida, tales como: desayuno, almuerzo, cena y meriendas. Ver Figuras 4.47 y 4.48 en las que se percibe la interfaz de la aplicación Web/móvil del detalle de un menú.



Menú

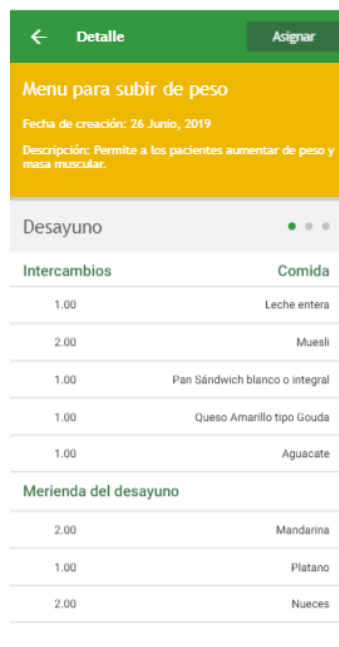
26 Junio, 2019

ASIGNAR MENU IMPRIMIR ELIMINAR MENU

Nombre del Menú: Menu para subir de peso **Descripción:** Permite a los pacientes aumentar de peso y masa muscular.

DESAYUNO		
Intercambios	Grupo	Comida
1.00	Leches Enteras	Leche entera
2.00	Almídones y Cereales	Muesli
1.00	Almídones y Cereales	Pan Sándwich blanco o integral
1.00	Carnes Gordas	Queso Amarillo tipo Gouda
1.00	Grasas	Aguacate
MERIENDA		
Intercambios	Grupo	Comida
2.00	Frutas	Mandarina

Figura 4.47 - Detalle de un menú del módulo del nutricionista, aplicación Web.



Detalle Asignar

Menu para subir de peso

Fecha de creación: 26 Junio, 2019

Descripción: Permite a los pacientes aumentar de peso y masa muscular.

Desayuno

Intercambios	Comida
1.00	Leche entera
2.00	Muesli
1.00	Pan Sándwich blanco o integral
1.00	Queso Amarillo tipo Gouda
1.00	Aguacate
Merienda del desayuno	
2.00	Mandarina
1.00	Platano
2.00	Nueces

Figura 4.48 - Detalle de un menú del módulo del nutricionista, aplicación móvil.

En dichas interfaces de la aplicación Web/móvil del detalle de un menú, se tiene la opción de asignar el menú, y en donde el nutricionista, al seleccionar dicha opción, se le despliega la lista de sus pacientes, y de los cuales debe seleccionar el paciente al que desea asignar el menú. En las Figuras 4.49 y 4.50 se observan las interfaces del flujo ya mencionado.

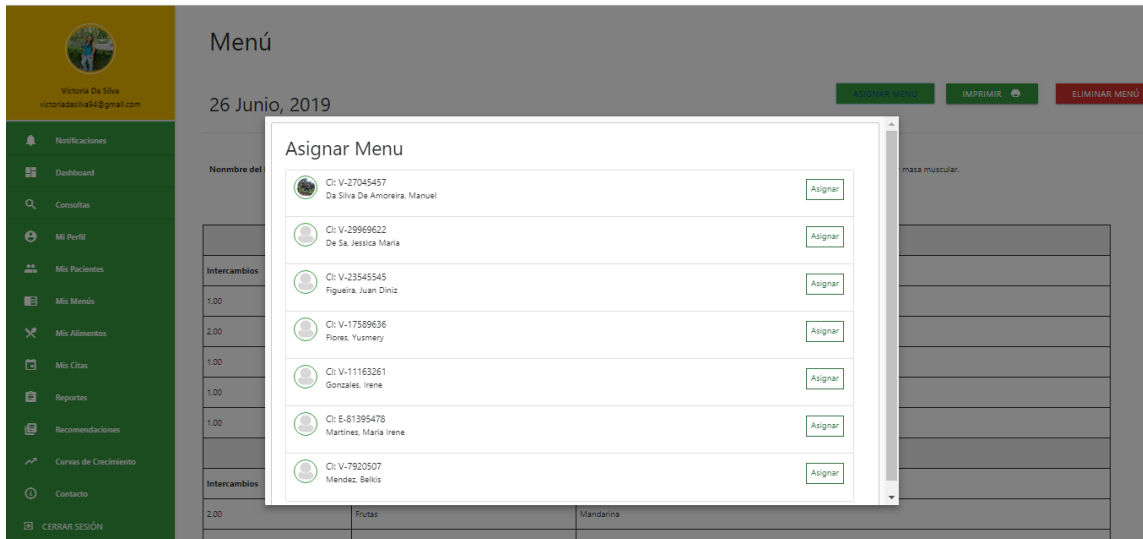


Figura 4.49 - Asignación de menú, aplicación Web.

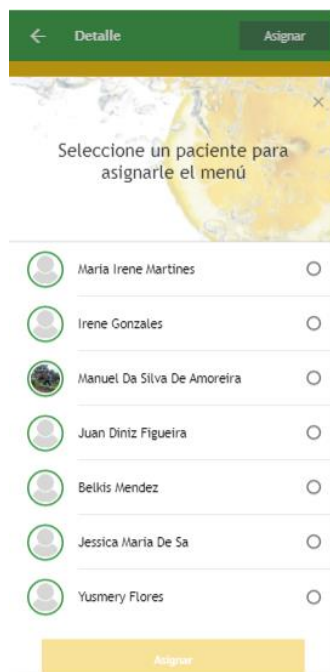


Figura 4.50 - Asignación de menú, aplicación móvil.

Interfaz “Mis Citas” del módulo de nutricionista

En dicha interfaz, el médico nutricionista puede crear una cita para un paciente, a través de la aplicación Web. Al hacer clic en el día en que se desea programar la cita, se despliega un formulario en el que se debe indicar un título, hora y el paciente al que se desea establecer la cita. En la Figura 4.51, se observa lo descrito.

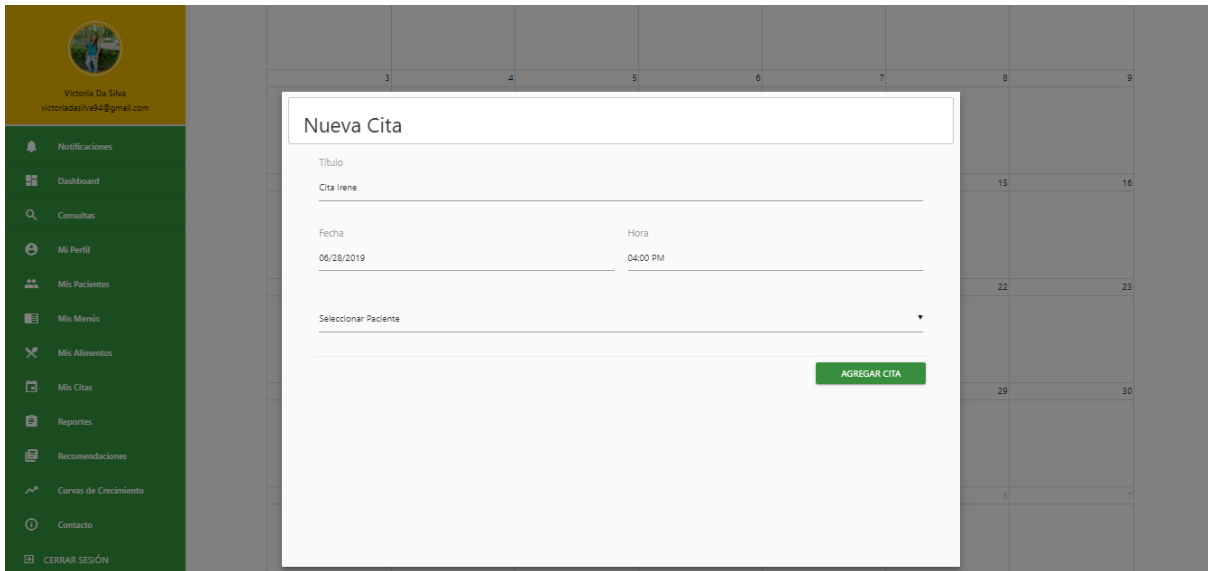


Figura 4.51 - Crear cita.

Además, todas las citas establecidas por el nutricionista, se visualizan en su calendario a través de la aplicación Web/móvil, como se muestra en las Figuras 4.52 y 4.53, descritas a continuación:

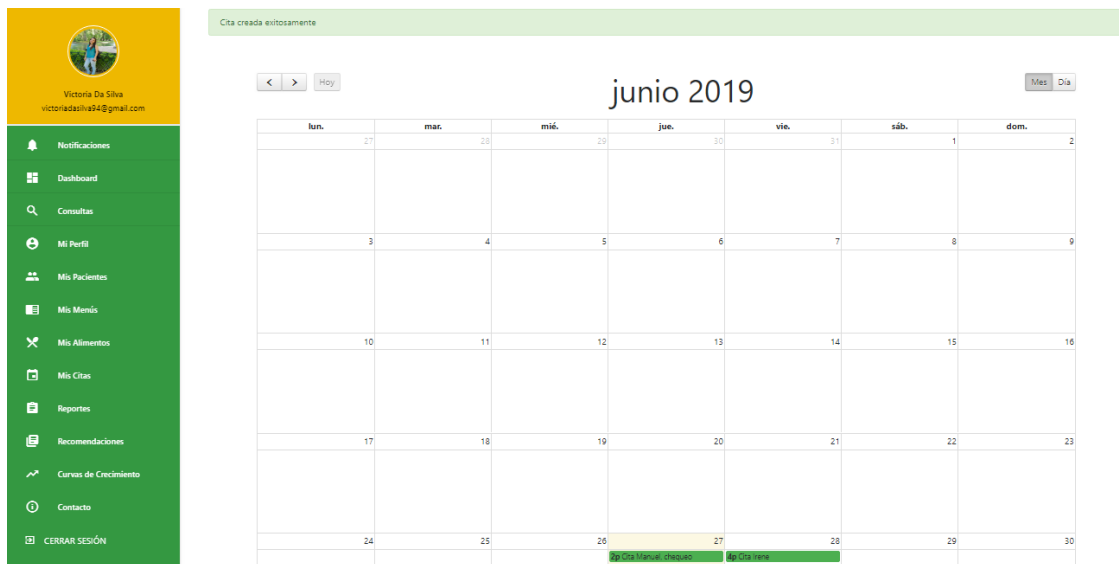


Figura 4.52 - Calendario de citas del nutricionista, aplicación Web.

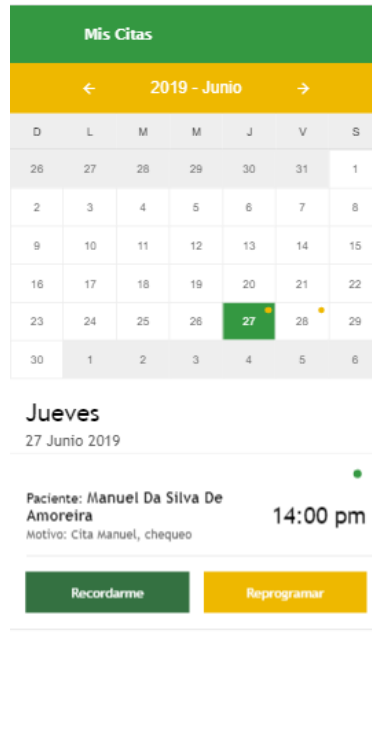


Figura 4.53 - Calendario de citas del nutricionista, aplicación móvil.

Por otro lado, el nutricionista puede reprogramar una cita ya creada. A través de la aplicación Web, se selecciona la cita que se desea reprogramar, y posteriormente se despliega un formulario en donde el nutricionista establece la fecha y hora que desea reprogramar la cita. Previo a completar dichos campos, se debe hacer clic en el botón “Cambio de cita” para habilitar la edición de los mismos. Lo descrito, puede apreciarse en la Figura 4.54.

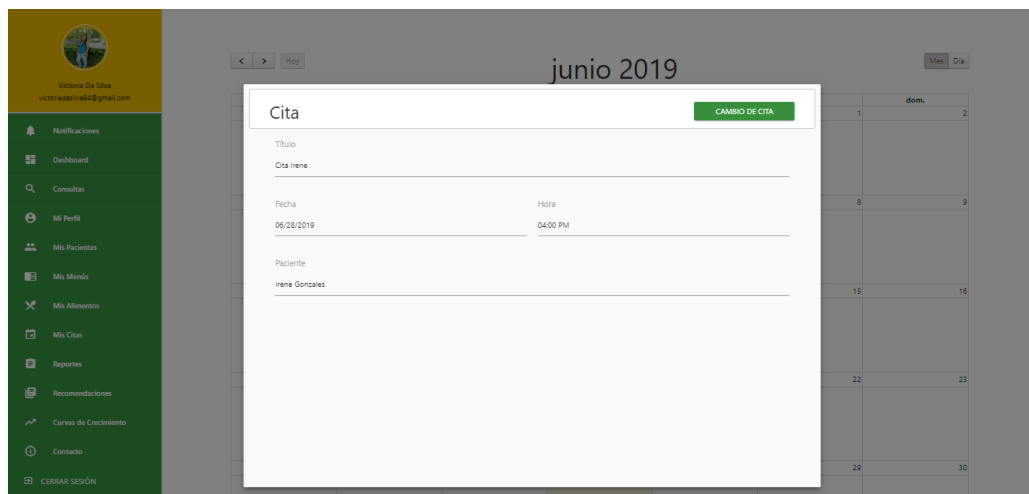


Figura 4.54 - Reprogramación de cita, aplicación Web.

En el caso de la aplicación móvil, dicho proceso de reprogramación de cita se realiza de igual modo que en la aplicación Web, a diferencia que se presiona el botón

de “Reprogramar”, y seguidamente se muestra el mismo formulario descrito anteriormente. Dicho proceso, puede ser descrito con las interfaces de la Figura 4.55.

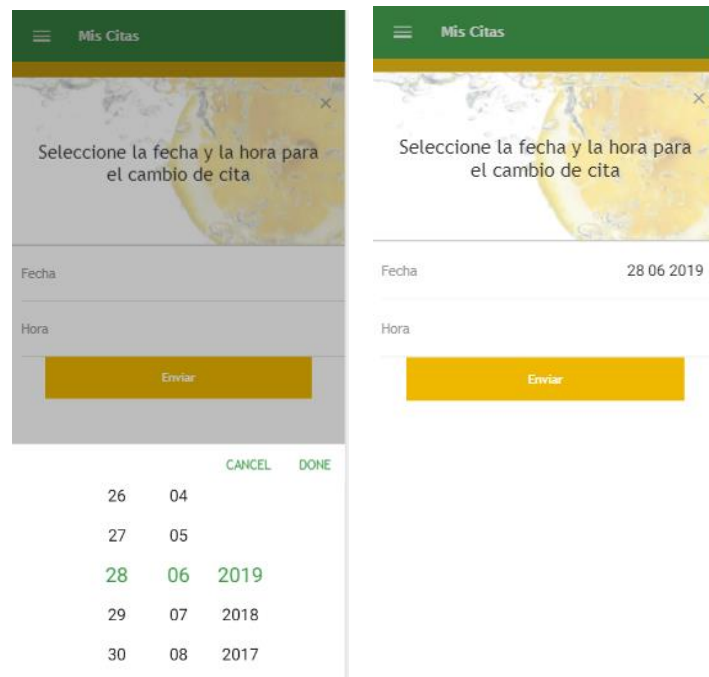


Figura 4.55 - Reprogramación de cita, aplicación móvil.

Interfaz “Reportes”

En dicha interfaz, los usuarios pacientes, cada día realizan un reporte de su avance diario, y el cual al ser completado, el mismo puede ser visualizado por su respectivo nutricionista, puesto que el nutricionista tendrá una lista de todos los reportes de sus pacientes, como se muestra en las Figuras 4.56 y 4.57.



Figura 4.56 - Lista de reportes de cada paciente, aplicación Web.

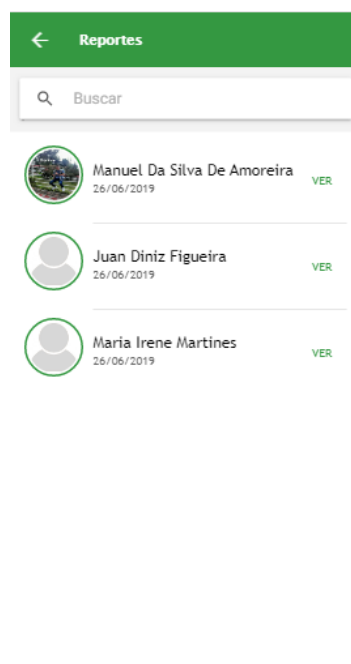


Figura 4.57 - Lista de reportes de cada paciente, aplicación móvil.

Dichos reportes tienen su respectivo detalle, el cual consta de un conjunto de preguntas y respuestas, que pueden ser visualizada en las Figuras 4.58 y 4.59.

Reporte del paciente: Manuel Da Silva De Amoreira	
Pregunta	Respuesta
¿Ya te pesaste el día de hoy?	Si
¿Cuánto fue el peso registrado?	64
¿Lograste hacer algún tipo de ejercicio físico?	Si
¿Durante cuanto tiempo?	2 horas
¿Recuerdas la cantidad de Kcal quemadas mientras te ejercitaste?	No
¿Cuántas horas lograste dormir?	7 horas
¿Desayunaste?	Si
¿Qué desayunaste?	Pan con queso amarillo
¿Comiste merienda del desayuno?	Si
¿Qué merendaste?	Un cambur
¿Almorzaste?	Si
¿Qué comiste en el almuerzo?	Arroz con espinaca, pollo y ensalada mixta
¿Comiste una merienda después del almuerzo?	No
¿Cenaste?	Si
¿Qué comiste en la cena?	3 empanadas de carne pchada
¿Cuántos vasos de agua tomaste hoy?	5 vasos

Figura 4.58 – Detalle de reporte, aplicación Web.


Reporte - 26/06/2019	
 <p>Manuel Da Silva De Amoreira</p>	
¿Ya te pesaste el día de hoy?	Si
¿Cuánto fue el peso registrado?	64
¿Lograste hacer algún tipo de ejercicio físico?	Si
¿Durante cuanto tiempo?	2 horas
¿Recuerdas la cantidad de Kcal quemadas mientras te ejercitaste?	No
¿Cuántas horas lograste dormir?	7 horas
¿Desayunaste?	

Figura 4.59 – Detalle de reporte, aplicación móvil.

Interfaz “Recomendaciones”

En la interfaz mencionada, el usuario nutricionista puede, a través de la aplicación Web, crear recomendaciones para luego asignarlas a sus pacientes, y las cuales, luego de ser creadas, se listan en la interfaz “Recomendaciones”, como se puede observar en la Figura 4.60.

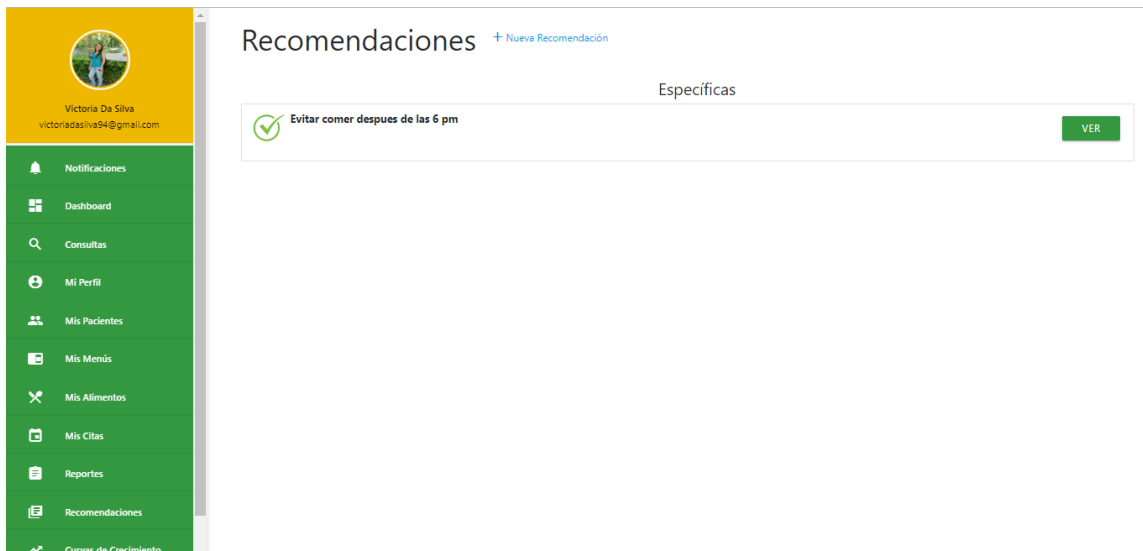


Figura 4.60 – Lista de recomendaciones.

El proceso de creación de dichas recomendaciones, se realiza comenzado con un clic en la opción “Nueva Recomendación”, originándose que se despliegue posteriormente, una interfaz en la que se muestra un formulario, en donde el nutricionista especifica la descripción de la recomendación, y de forma opcional, una imagen. En la Figura 4.61 se percibe la interfaz con lo anteriormente descrito.

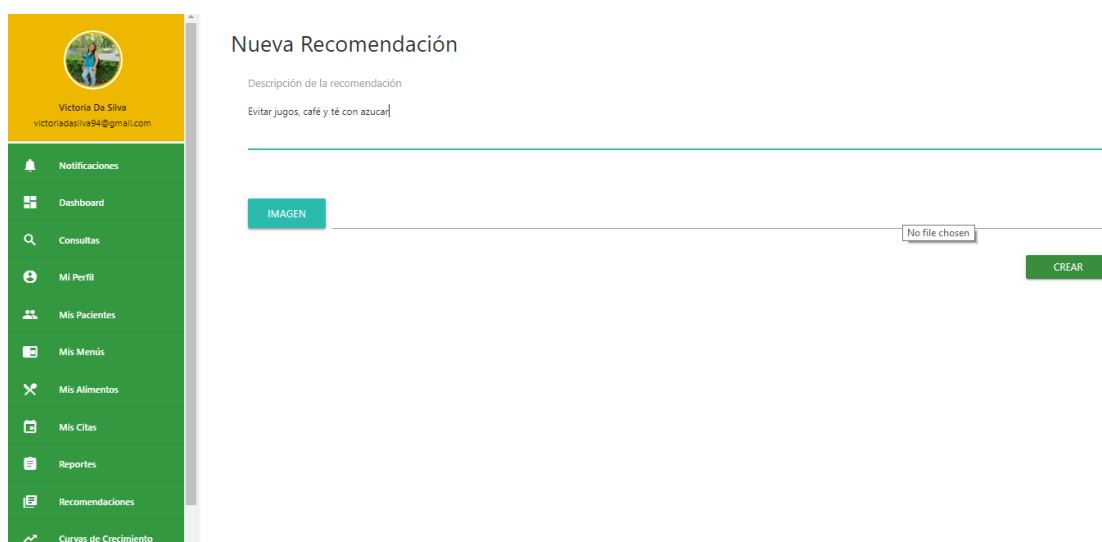


Figura 4.61 – Crear Recomendaciones.

Además, cada recomendación tiene su respectiva interfaz de detalle, en la que se encuentra la opción de asignar, la cual permite al nutricionista, asignar una recomendación a un paciente en específico. Ver Figura 4.62 para observar la asignación respectiva.

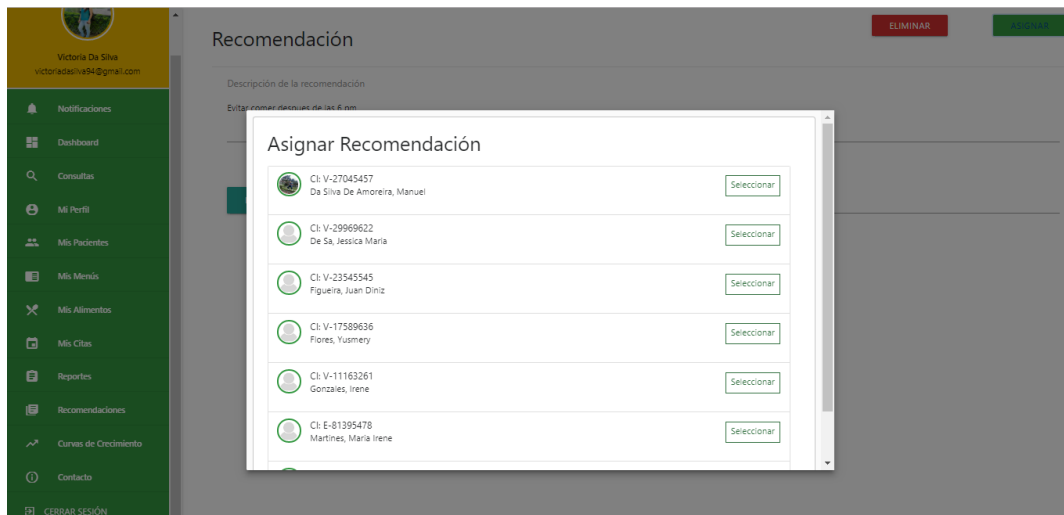


Figura 4.62 – Asignación de recomendación.

Interfaz “Mis Pacientes”

En la Figura 4.63, se muestra la interfaz denominada “Mis pacientes”, en la cual, el nutricionista, a través de la aplicación móvil, puede observar su lista de pacientes.

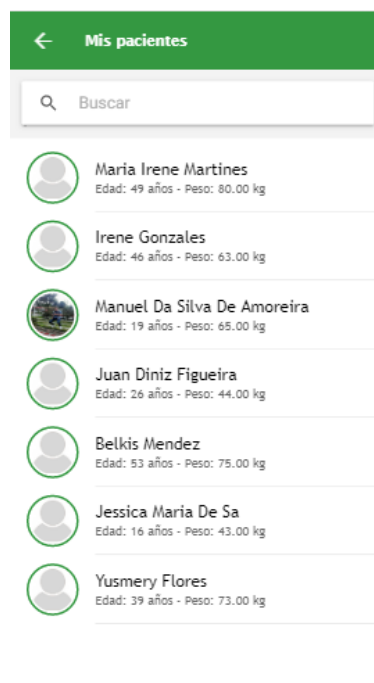


Figura 4.63 – Lista de pacientes.

Adicionalmente, por cada paciente, el nutricionista puede observar su detalle, el cual comprende datos tales como: la información básica, sus respectivas medidas,

la información médica, y sus hábitos. En la Figura 4.64, se muestran las interfaces del detalle de la información de un paciente, en la aplicación móvil.

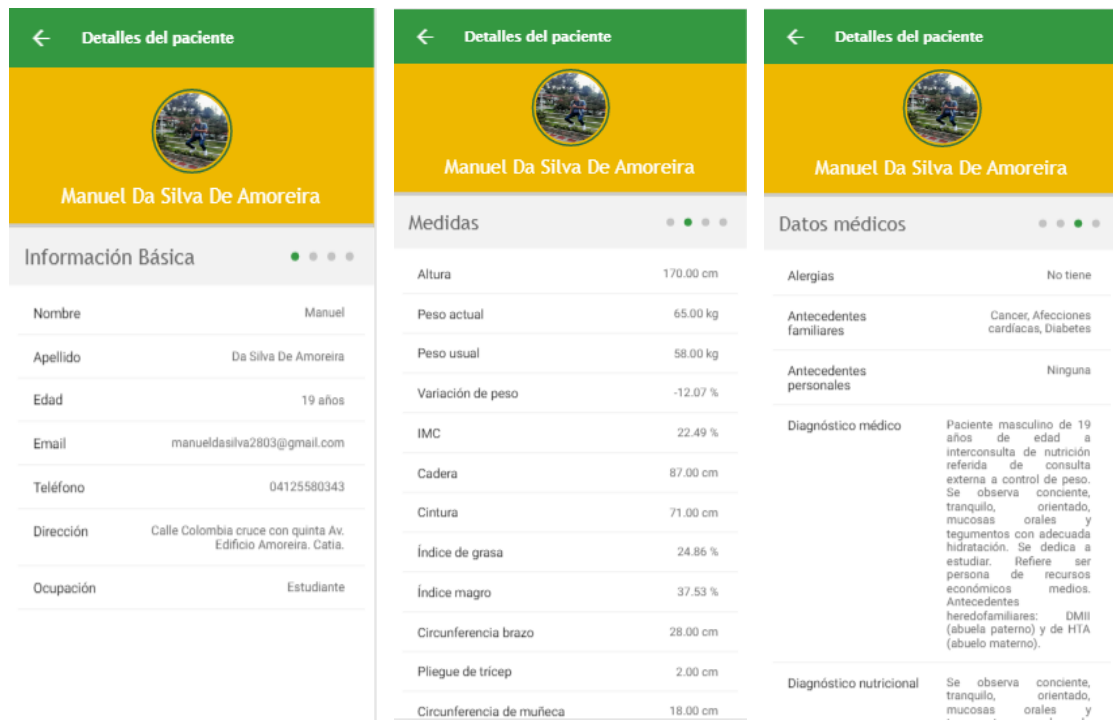


Figura 4.64 - Detalle del paciente.

CONCLUSIONES

Se puede concluir, que se cumplió satisfactoriamente con el objetivo general del presente TEG, el cual consistió en el desarrollo de una aplicación Web/móvil que apoye a los profesionales y pacientes en el área de Nutrición y Dietética, basada en la aplicación Web ya desarrollada, llamada SysNutr.

Gracias al uso de una adaptación de la metodología XP, dada las cuatro fases de su ciclo de desarrollo, se diseñó una solución de manera estructurada, siguiendo un proceso esquematizado, en el cual, la constante comunicación entre el equipo de desarrollo y los clientes (profesionales y pacientes en el área de nutrición), fue de gran ayuda para completar la construcción del aplicativo, despejando cualquier duda acerca del funcionamiento del proceso de evaluación nutricional, y el seguimiento de un plan dieto-terapéutico.

Es importante destacar, que la decisión de continuar con la misma arquitectura, y con las mismas tecnologías del desarrollo de la aplicación Web SysNutr, permitió agilizar el tiempo de desarrollo, y de una manera organizada, ya que la estructura que brinda el marco de trabajo Laravel, ofrece un desarrollo eficaz de las distintas funcionalidades, además de poder implementar, tanto la lógica de la aplicación, como las interfaces de usuario en paralelo. A su vez, dicha estructura, al utilizar el patrón MVC, brindó comodidad a la hora de organizar el código, haciéndolo mucho más legible y entendible para realizar modificaciones o extraer código que resultara útil para nuevas vistas o funcionalidades. Además, en definitiva, el uso del *framework* Laravel junto con HTML, CSS, JQuery y Materialize, permitió que se pudieran presentar resultados rápidos y satisfactorios a los problemas planteados; y en cuanto a la migración del SMD MySQL, al SMD PostgreSQL, ofreció la manipulación de datos de una manera óptima.

Al contar con una versión preliminar del producto, se aplicaron pruebas de aceptación para detectar y corregir errores. Gracias a dichas pruebas, se pudieron hacer ajustes en torno a los problemas detectados. Los resultados obtenidos, determinaron que se desarrolló un producto que cumple con las funcionalidades esperadas, y que es intuitivo y fácil de utilizar por parte de los usuarios involucrados.

La aplicación Web/móvil fue llevada a cabo, siguiendo buenas prácticas de desarrollo, y cumpliendo con propiedades fundamentales como la usabilidad,

escalabilidad, portabilidad y seguridad, lo cual facilitará su posible crecimiento, a través del desarrollo de nuevos módulos por parte de otros desarrolladores.

En general, se puede concluir que a través del desarrollo de la aplicación Web/móvil, se proporciona a los profesionales y pacientes en el área de nutrición, una serie importante de mejoras y beneficios para el desenvolvimiento de sus actividades, entre las cuales se destacan las siguientes:

- Facilidad para los nutricionistas en el seguimiento del tratamiento de sus pacientes.
- Facilidad para los pacientes al poder llevar un seguimiento continuo de su plan dieto-terapéutico, observando información como: su progreso, sus menús, sus recomendaciones, entre otros.
- Y acceso a los datos desde cualquier lugar donde se cuente con internet, ya que se puede acceder a la aplicación desde un dispositivo móvil o un navegador web.

RECOMENDACIONES Y TRABAJOS FUTUROS

A continuación, se listan una serie de recomendaciones y/o trabajos futuros que podrían ser de interés para próximos desarrollos.

- Crear un módulo específico para la población deportiva, que permita calcular sus requerimientos según la demanda, aunado a una sección que contenga solo suplementos deportivos, para incluir a las dietas de los mismos.
- Crear un módulo capaz de generar dietas y esquemas de alimentación personalizados, para pacientes que se encuentren recibiendo nutrición enteral (vía sondas nasogástricas, nasoyeyunal o gastrostómo); que reciban nutrición parenteral periférica o central (vía intravenosa), aunado a una sección que contenga solo suplementos y nutrimentos especializados, para cada patología y requerimiento de los pacientes.
- Agregar aspectos de accesibilidad a la aplicación, como, por ejemplo: cambio de colores, cambio de idioma y cambio del tamaño de la fuente, con la finalidad que pueda ser personalizado para cada empresa o particular.
- Desarrollar en la aplicación Web/móvil, el envío de notificaciones de recordatorio, ya sea por hora de comida, cita pendiente, entre otras.

REFERENCIAS BIBLIOGRÁFICAS Y DIGITALES

AngularJS. (2018). Sobre AngularJS. Recuperado de: <https://angularjs.org/>

Alvarez, M. (2001). DesarrolloWeb.com. Qué es PHP. Recuperado de: <https://www.desarrolloWeb.com/articulos/392.php>

Apache. (2008). *The Apache Software Foundation - Apache httpd server - Web Server*. Recuperado de: <http://www.apache2.com>

Asensio, B., & Menéndez, R. (s.f). Desarrollo Aplicaciones Web. Recuperado de: <http://www.um.es/docencia/barzana/DAWEB/Lenguaje-de-programacion-HTML-1.pdf>

Ashwini, A. (2017). *React Native Vs Ionic 2: Comparison*. Recuperado de: <https://medium.com/swlh/react-native-vs-ionic-2-comparison-50aba900be6c>

Basrai, M. (2017). *Top Ruby Frameworks to Speed Up Web Development*. Recuperado de: <https://www.icletech.com/blog/top-ruby-frameworks-to-speed-up-Web-development>

Blanco, S. (2008). Metodologías de desarrollo. Recuperado de: <http://www.marblestation.com/?p=644>

Blazquez, E. (2017). Fundación Vivo Sano. NUTRICIÓN POR EDADES. Recuperado de: <http://www.vivosano.org/nutricion-por-edades/>

Carbajal, A. (s.f). Departamento de Nutrición. Facultad de Farmacia. Universidad Complutense de Madrid. Manual de Nutrición y Dietética. Recuperado de: <https://www.ucm.es/data/cont/docs/458-2013-07-24-cap-1-conceptos-basicos.pdf>

Cake Software Foundation, Inc. (2016). Entendiendo el Modelo - Vista - Controlador. Recuperado de: <https://book.cakephp.org/2.0/es/cakephp-overview/understanding-model-view-controller.html>

Cohn, M. (2011). Introduction to User Stories. Broomfield, EU. Recuperado de: <https://www.mountaingoatsoftware.com/presentations/introduction-to-user-stories>.

Cubillos, C. (s.f.). Escuela de Ingeniería Informática Pontificia Universidad Católica de Valparaíso. Arquitectura Cliente/Servidor. Recuperado de: <http://ocw.pucv.cl/cursos-1/arquitectura-de-sistemas-de-software/materiales-de-clases/Web-cliente-servidor>

De la Serna Martínez, A. (2017). Escogiendo un *framework* para desarrollo híbrido: *Ionic VS React Native*. Recuperado de: <https://labs.beeva.com/escogiendo-un-framework-para-desarrollo-h%C3%ADbrido-ionic-vs-react-native-e69406283549>

Delgado, H. (2018). Estructura básica de una página Web - html, head y body. Recuperado de: <https://disenoWebakus.net/domine-html-y-dhtml-primeros-pasos.php>

EcuRed. (s.f.). Metodologías de desarrollo de Software. Recuperado de: https://www.ecured.cu/Metodologias_de_desarrollo_de_Software

Eguiluz, J. (2009). Introducción a XHTML. Recuperado de librosWeb: <http://librosWeb.es/libro/xhtml/>

Extreme Programing. (s.f.). Sobre *eXtreme Programing*. Recuperado de: <https://iswugextremeprog.wordpress.com/>

GES Formación. (2016). Nutricionista: funciones y salidas profesionales. Recuperado de: <https://www.gesformacion.edu.es/blog/nutricionista-funciones-salidas-profesionales/>

Getbootstrap. (2018). Sobre Bootstrap. Recuperado de: <https://getbootstrap.com/>

González, A. (2018). Dieta por intercambios. Recuperado de: <https://eresdeportista.com/dieta-nutricion/dieta-por-intercambios/>

Heroku. (s.f.). Sobre Heroku. Recuperado de: <https://www.heroku.com/about>

Hernández, E., & Martínez, L. (1997). *Cliente/servidor*. Obtenido de <http://clubdeinvestigacion.com/docs/informe23.pdf>

Ingsoftwareisc. (2016). Diseño de Software de arquitectura Cliente-Servidor. Recuperado de: <https://ingsoftwareisc.wordpress.com/2016/05/27/1-5-diseno-de-software-de-arquitectura-cliente-servidor/>

jQuery. (2018). Sobre jQuery. Recuperado de: <https://jquery.com/>

Luján, S. (2002). Programación de aplicaciones Web: historia, principios básicos y clientes Web. Recuperado de: <https://gplsi.dlsi.ua.es/~slujan/materiales/pi-cliente2-muestra.pdf>

Lutty, R. (2014). Desarrollo de aplicaciones móviles. Recuperado de: http://repositorio.unapiquitos.edu.pe/bitstream/handle/UNAP/4515/Robertho_Tesis_Titulo_2_014.pdf?sequence=1&isAllowed=y

Materialize. (s.f.). Sobre Materialize. Recuperado de: <https://materializecss.com/about.html>

Menéndez, R., & Barzanalla, A. (2000). Informática Bases de Datos. Recuperado de um.es: <http://www.um.es/docencia/barzana/DIVULGACION/INFORMATICA/sgbd.html>

Murazzo, M., Millán, F., Rodríguez, N., Segura, D., & Villafañe, D. (2010). Desarrollo de aplicaciones para Cloud Computing. Universidad Nacional de San Juan, Facultad de Ciencias Exactas, Físicas y Naturales. CACIC 2010 - XVI Congreso Argentino de Ciencias de la Computación. Trabajos presentados (p. 941-949). ISBN: 978-950-9474-49-9. Recuperado de: http://sedici.unlp.edu.ar/bitstream/handle/10915/19374/Documento_completo.pdf?sequence=1

Nutrimind. (s.f.). Sobre Nutrimind. Recuperado de: <https://www.nutrimind.net/>

Nutrium. (s.f.). Software de Nutrición para Dietistas-Nutricionistas y Pacientes. Recuperado de: <https://nutrium.io/es>

Peñafiel, M. (s.f.). SCRIBD. Aplicaciones Web. Recuperado de: <https://es.scribd.com/doc/136052164/APLICACIONES-WEB-pdf>

Pérez, A. (2007). Desarrollo de herramientas Web de gestión docente. Recuperado de: upct.es: <http://repositorio.upct.es/bitstream/handle/10317/179/pfc2475.pdf?sequence=1>

PostgreSQL. (2010). Sobre PostgreSQL. Recuperado de: PostgreSQL Global Development Group Sitio Web: http://www.postgresql.org/es/sobre_postgresql

Quispe, W. (2016). Arquitectura de Cliente Servidor de tres capas. Recuperado de: <https://es.slideshare.net/anibalsmit/arquitectura-de-clienteservidor-de-tres-capas>

Reenskaug, T. & Coplien, J. (2009). The DCI Architecture: A New Vision of Object-Oriented Programming. Recuperado de: http://www.artima.com/articles/dci_vision.html.

Rivas, M. (2010). Ventajas y desventajas de las aplicaciones Web. Recuperado de: Vida MMR: <http://www.vidamrr.com/2010/03/ventajas-y-desventajas-de-las.html>

Rodríguez, A. (2013). Teoría y práctica de la intervención socio familiar con familias multiproblemáticas. Recuperado de: <https://eprints.ucm.es/17792/1/T34107.pdf>

Ruiz, P. (2013). Ventajas e inconvenientes de la arquitectura cliente/servidor. Recuperado de SomeBooks.es: <http://somebooks.es/ventajas-e-inconvenientes-de-la-arquitectura-clienteservidor/>

Schwaber y Sutherland (2014). La Guía de Scrum: Las Reglas del juego. Recuperado de: <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-ES.pdf>

Serracanta, M. (2018). EAE. Usos y limitaciones de la metodología Scrum. Recuperado de: <https://retos-directivos.eae.es/usos-y-limitaciones-de-la-metodologia-scrum/>

Sierra, F., Acosta, J., Ariza, J., & Salas, M. (2013). Estudio y análisis de los framework en php basados en el modelo vista controlador para el desarrollo de software orientado a la Web. Recuperado de: <http://publicaciones.unisimonbolivar.edu.co/rdigital/innovacion/index.php/identific/article/viewFile/73/91>

Silberschatz, A., Korth, H., & Sudarshan, S. (2002). Fundamentos de Base de Datos. Recuperado de: <https://unefazuliasistemas.files.wordpress.com/2011/04/fundamentos-de-bases-de-datos-silberschatz-korth-sudarshan.pdf>

Tagle, J. (2014). ¿Qué es un plugin y para que funciona?. Recuperado de: <https://wpavanzado.com/que-es-un-plugin/>

Travieso, F & Benes, M. (2017). Desarrollo de una aplicación Web para la gestión de pacientes y apoyo a los profesionales del área de nutrición y dietética. Caracas, Venezuela.

Undercode. (2013). Modelo Vista Controlador. Recuperado de: <https://underc0de.org/foro/java/modelo-vista-controlador/>

Villa, L. (2005). La filosofía Basecamp y Ruby on Rails. Recuperado de:
<https://desarrolloWeb.com/articulos/1797.php>

Wells, D. (1999). *The Rules of Extreme Programming*. Recuperado de
<http://www.extremeprogramming.org/rules.html>

W3C. (2018). Sobre *Cascading Style Sheets*. Recuperado de:
<https://www.w3.org/Style/CSS/Overview.en.html>

Yañez, C. (2017). CEAC. Diferencias entre apps Nativas o Híbridas. Recuperado de:
<https://www.ceac.es/blog/diferencias-entre-apps-nativas-o-hibridas>