



UNIVERSIDAD CENTRAL DE VENEZUELA

FACULTAD DE CIENCIAS

ESCUELA DE COMPUTACIÓN

CENTRO DE INVESTIGACIÓN EN SISTEMAS DE INFORMACIÓN

Desarrollo de una aplicación web para la gestión de pacientes y apoyo a los profesionales del área de nutrición y dietética

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

Br. Mariano Benes

Br. Frank Travieso

Para optar por el título de Licenciado en Computación

Tutora:

Profa. Concettina Di Vasta

UCV Caracas, Septiembre 2017



ACTA DEL VEREDICTO

Quienes suscriben, Miembros del Jurado designado por el Consejo de la Escuela de Computación para examinar el Trabajo Especial de Grado, presentado por los Bachilleres **Mariano Benes C.I.: 19.658.781** y **Frank Travieso C.I.: 24.224.073** con el título "Desarrollo de una aplicación web para la gestión de pacientes y apoyo a los profesionales del área de nutrición y dietética", a los fines de cumplir con el requisito legal para optar al título de Licenciado en Computación, dejan constancia de lo siguiente:

Leído el trabajo por cada uno de los Miembros del Jurado, se fijó el día 06 de Octubre del 2017 a las 08:00 a.m., para que sus autores lo defendieran en forma pública, en la Sala I de la Escuela de Computación, mediante una exposición oral de su contenido, y luego de la cual respondieron satisfactoriamente a las preguntas que les fueron formuladas por el Jurado, todo ello conforme a lo dispuesto en la Ley de Universidades y demás normativas vigentes de la Universidad Central de Venezuela. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobarlo.

En fe de lo cual se levanta la presente acta, en Caracas el 06 de Octubre del 2017, dejándose también constancia de que actuó como coordinadora del Jurado la Profesora Concettina di Vasta

Prof. Concettina Di Vasta

(Tutora)

Prof. Yusneyi Carballo

(Jurado)

Prof. Mercy Ospina

(Jurado)

Agradecimientos

A mis padres Rose Mary Peña y Frank Fernando Travieso por toda la educación que me han dado a lo largo de mi vida, gracias a eso nunca he perdido el norte y he cumplido cada una de las metas que me he propuesto y sobre todo por todo el apoyo que me han dado en cada momento para que pueda seguir adelante con cada uno de los retos que se me han planteado en la vida, por hacer que nunca me haya faltado nada.

A mi tía Durbis Peña por siempre estar brindándome apoyo en todo momento, por siempre ser comprensiva y cariñosa conmigo, por ser mi amiga, escucharme y guiarme cada vez que lo necesito.

A mis amigos Joaquín Molina y Carlos Riera. Al primero por ser un amigo incondicional durante los últimos 6 años en lo académico, en lo laboral y en lo personal por siempre brindarme ayuda en todo momento. Al segundo por ser un gran compañero de trabajo y un buen amigo, por brindarme ayuda siempre para el crecimiento profesional.

A mis amigos Carlos Mora, Daniela De Vivo y Karina De Sousa, por ser excelentes compañeros de carrera tanto en lo académico como en lo personal, por siempre brindarme su apoyo en ambos ámbitos.

Frank L. Travieso P.

A dios por darnos la oportunidad de vivir esta experiencia, tanto como personas y futuros profesionales en Computación, así como por colocar a cada persona en mi vidas, durante todo el camino recorrido en esta carrera, ya que todas han dejado un aprendizaje importante en mi

A todos aquellos profesores, amigos y conocidos que en algún momento de la carrera me brindaron el apoyo o la ayuda que estuviera a su alcance y que han hecho esto posible.

A mis padres y familiares por el amor, educación, los principios y el apoyo incondicional que nos han brindado, ya que, sin eso, cada uno de los logros que hasta hoy se han materializado, no fuesen sido posible de la misma manera que hoy lo son.

A la Profa. Concettina Di Vasta, por su labor excepcional como tutor académico, a lo largo de la realización y ejecución de este proyecto especial de grado.

A mi novia Vanessa López por la paciencia, el tiempo, esfuerzo, dedicación y disposición que nos brindaron durante la realización de esta aplicación, ya que, sin los conocimientos ofrecidos por ellas la realización de este trabajo no fuese sido posible.

Mariano C. Benes S.

UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN
CENTRO DE INVESTIGACIÓN EN SISTEMAS DE INFORMACIÓN

Desarrollo de una aplicación web para la gestión de pacientes y apoyo a los profesionales del área de nutrición y dietética

Autores: Mariano Benes
Frank Travieso

Tutora: Profa. Concettina Di Vasta

Fecha: 02/10/2017

RESUMEN

El presente Trabajo Especial de Grado describe la problemática que presentan los profesionales del área de nutrición en Venezuela para llevar control del estado nutricional de todos sus pacientes, así como la labor manual necesaria para calcular todos los valores requeridos para poder realizar planes dieto-terapéuticos acordes a los requerimientos de los mismos. Conociendo las necesidades e inconvenientes existentes, se plantea el desarrollo de una aplicación web para la gestión de pacientes y apoyo a los profesionales del área de nutrición y dietética llamada SysNutr, una aplicación web creada con tecnologías de software libre (HTML5, CSS3, JavaScript, jQuery, Bootstrap, PHP, Laravel y MySQL) bajo una combinación de las metodologías ágiles AgilUs y Scrum, la cual permite registrar y gestionar los datos del paciente, realizar cálculos automatizados de requerimientos y crear, gestionar y modificar menús personalizados que cumplan con las necesidades de cada paciente, de esta manera simplificando el trabajo requerido por parte de los nutricionistas para realizar sus labores.

Palabras clave: nutrición, nutricionista, gestión de pacientes, menús, plan dieto terapéutico, aplicación web.

TABLA DE CONTENIDO

CAPÍTULO 1 - PROBLEMA DE INVESTIGACIÓN	9
1.1 Planteamiento del Problema	9
1.2 Objetivos	11
1.3 Arquitectura de la solución	11
1.4 Justificación de la solución	13
CAPÍTULO 2 - MARCO CONCEPTUAL	14
2.1 Antecedentes	15
2.1.1 Alimentación Requerimiento Nutricional y Adecuación CANIA (ARNAC)	15
2.1.2 NUTRIMIND	16
2.1.3 WHO Anthro	18
2.1.4 WHO AnthroPlus	21
2.1.5 Método manual	21
2.2 Conceptos Básicos relacionados con el área de Nutrición	25
2.3 Arquitectura cliente-servidor	26
2.4 Aplicación web	27
2.4.1 Funcionamiento de una aplicación web	27
2.4.2 Ventajas	29
2.4.3 Desventajas	29
2.4.4 Patrón de Modelo Vista Controlador (MVC)	30
2.5 Tecnologías del lado del cliente	32
2.5.1 Lenguaje de marcas de hipertexto	32
2.5.2 Hojas de estilo en cascada	32
2.5.3 Javascript	32
2.5.4 JQuery	33
2.5.5 Materialize	33
2.6 Tecnologías del lado del servidor	33
2.6.1 PHP	34
2.6.2 Sistemas Manejadores de Bases de Datos	36
CAPÍTULO 3 - MARCO METODOLÓGICO	37
3.1 Metodologías de Desarrollo	37
3.1.1 Metodologías Ágiles	38

4.1	Análisis Preliminar	46
4.2	Diagrama de Casos de Uso	47
4.3	Diseño de la Solución	50
4.4	Desarrollo de la solución	50
4.4.1	Base de Datos	50
4.4.2	Tecnologías involucradas en el desarrollo de la Aplicación	53
4.5	Flujo de la aplicación web e Interfaces de Usuario	57
4.5.1	Pantalla de acceso	57
4.5.2	Interfaz “Dashboard”	57
4.5.3	Interfaz “Mis pacientes”	58
4.5.4	Interfaz “Registro del Paciente”	59
4.5.5	Interfaz “Paciente”	60
4.5.6	Interfaz “Historia Dietética”	61
4.5.7	Interfaz “Antropometría”	62
4.5.8	Interfaz “Nuevo Laboratorio”	64
4.5.9	Interfaz “Dietas”	66
4.5.10	Interfaz “Menú”	68
4.5.11	Interfaz “Mis Alimentos”	69
4.5.12	Interfaz “Registro de Alimentos”	70
4.5.13	Interfaz “Alimento”	70
4.6	Las Iteraciones	72
4.6.1	Primera Iteración: configuraciones básicas de la aplicación	72
4.6.2	Segunda Iteración: navegación, pacientes y laboratorios	72
4.6.3	Tercera Iteración: antropometrías y fórmulas:	73
4.6.4	Cuarta Iteración: alimentos y dietas	74
4.6.5	Quinta Iteración: creación y visualización de menús	74
4.6.6	Sexta iteración: curvas de crecimiento	75
4.7	Las pruebas	76
CONCLUSIONES		82
4.1	Limitaciones	84
4.2	Recomendaciones y Trabajos Futuro	84
REFERENCIAS BIBLIOGRÁFICAS Y DIGITALES		85

INTRODUCCIÓN

Desde que el hombre habita en el planeta tierra ha buscado y encontrado diferentes maneras de alimentarse de manera adecuada, un ejemplo de esto es el uso del fuego para la cocción de los alimentos o el uso de sal para su conservación, todo esto con el fin de alimentarse, obtener todos los nutrientes provenientes de estos y convertirlos posteriormente en energía, para realizar las actividades diarias.

El ser humano, desde la antigüedad, acostumbra a consumir alimentos en cantidades tales que satisfagan su apetito, sin preocuparse en la calidad de los mismos, lo cual ocasiona un desequilibrio tanto energético como nutritivo, expresándose este desequilibrio como enfermedades relacionadas a la alimentación (hipertensión arterial, diabetes, obesidad, desnutrición, entre otras).

La nutrición y dietética puede definirse como una ciencia o como un proceso vital; es una ciencia debido a que esta se encarga de estudiar la relación que existe entre la alimentación y diferentes patologías; y es un proceso vital, debido a que gracias a esta, el organismo es capaz de convertir los alimentos en elementos nutritivos, cuya función es mantener, reparar y aportar energía a cada una de las células especializadas que conforman el cuerpo humano, por este motivo la nutrición y dietética se considera una de las áreas más importantes y necesarias de la medicina preventiva y curativa, ya que ayuda a mantener el buen estado de salud del paciente a través de la implementación de planes dieto terapéuticos que se adapten a las características biológicas, antropométricas y fisiológicas de cada individuo con el fin de garantizar el funcionamiento óptimo del organismo.

Existen un conjunto de datos esenciales que se necesitan para poder realizar un plan dieto terapéutico óptimo para un paciente, entre ellos tenemos su contextura, género, edad, índice de masa corporal (IMC), requerimientos calóricos, estado nutricional, frecuencia e intensidad de la actividad física, entre otros. Algunos de estos se obtienen a través de una encuesta que se le aplica al paciente, pero otros involucran la medición de las proporciones del mismo y en base a estas medidas realizar una serie de procedimientos y cálculos.

Actualmente los profesionales de nutrición realizan todos estos procedimientos a papel y lápiz; y en ocasiones utilizando hojas de Excel, el problema de estos métodos de procesamiento y análisis es que pueden llegar a ser tediosos y repetitivos, lo que causa que se desperdicie tiempo importante realizando cálculos en vez de aprovechar este tiempo tratando a los pacientes en sí, y/o analizando los cálculos obtenidos.

Para solventar esta situación surge este trabajo de investigación, cuyo objetivo es desarrollar una aplicación web para la gestión de pacientes y apoyo a los profesionales del área de nutrición y dietética

Para lograr este desarrollo, se ha estructurado este documento en cuatro (4) capítulos. El capítulo 1, “Problema de Investigación”, en el cual se plantea el problema al que se le busca solución, y los objetivos de este trabajo, así como la solución al problema, la arquitectura de la solución y la justificación de la solución. El capítulo 2, “Marco Conceptual”, que consta de los antecedentes a este trabajo y de todos los conceptos relevantes y necesarios para el desarrollo de este Trabajo Especial de Grado (TEG). El capítulo 3, “Marco Metodológico”, describe las metodologías que fueron seleccionadas para ser usadas en el desarrollo de este trabajo y, por último, el capítulo 4, “Marco Aplicativo”, donde se describe las fases de la metodología seleccionada para el desarrollo de la aplicación y se explican características tales como: arquitectura utilizada, base de datos, funcionalidades, interfaces y pruebas realizadas para determinar el correcto funcionamiento de la aplicación.

Al finalizar se presentan las conclusiones, limitaciones, recomendaciones y trabajos futuros.

CAPÍTULO 1 - PROBLEMA DE INVESTIGACIÓN

1.1 Planteamiento del Problema

El objetivo principal del nutricionista es garantizar el óptimo funcionamiento del organismo del paciente, para esto es necesario determinar el requerimiento nutricional del mismo, con la finalidad de poder cubrirlo, mediante la prescripción de un plan dietético que permita garantizar un estado físico y nutricional adecuado de acuerdo con su grupo etario, estado fisiológico y patológico.

Para lograr lo anteriormente mencionado, es necesario realizar tanto un examen físico y antropométrico donde se determinará: la talla, peso, circunferencias y pliegues; exámenes bioquímicos que determinan la cantidad de nutrientes y componentes sanguíneos; y por último una prueba socioeconómica la cual permite identificar la existencia de riesgos no orgánicos de desnutrición al detectar aquellos factores socioeconómicos y psicológicos que impiden la cobertura de los requerimientos nutricionales del paciente, debido a que limitan el acceso a los alimentos induciendo alteraciones en la ingesta o aprovechamiento de los mismos. Estos resultados determinan el estado nutricional actual del paciente (lo cual se realiza mediante la aplicación de fórmulas y parámetros nacionales e internacionales) y dictan las pautas a seguir para realizar un plan dieto terapéutico personalizado, que contribuya a mejorar el estado nutricional y de salud del mismo, sin empeorar la condición patológica asociada.

Estos planes consisten en una serie de comidas o preparaciones distribuidas a lo largo del día diseñadas para cumplir con las necesidades del paciente, así como, corregir y evitar la alteración de los elementos que componen la bioquímica sanguínea del mismo.

Para realizar los procesos mencionados anteriormente los nutricionistas se guían por el programa internacional de biología y en lo que se refiere al campo del deporte, el ISAK (The International Society for the Advancement of Kinanthropometry) creó una metodología que facilita y estandariza las mediciones que deben ser realizadas al paciente.

En la actualidad existen pocas aplicaciones diseñadas para apoyar al nutricionista durante sus tareas cotidianas, a continuación, hablaremos un poco sobre ellas y sobre sus ventajas y desventajas, posteriormente se explican con más nivel de detalle.

En primer lugar, hablaremos un poco de ARNAC (Alimentación Requerimiento Nutricional Adecuación CANIA), esta aplicación desarrollada por el Centro de Atención Nutricional Infantil Antímamo (CANIA), fue diseñada para calcular requerimientos nutricionales y para realizar estimaciones de consumo de nutrientes (cálculo de calorías según lo que el paciente informa).

Luego, se tiene Nutrimind, esta aplicación, tanto web como de escritorio, ofrece distintas herramientas para analizar la composición corporal del paciente basado en las mediciones antropométricas del mismo, además ayuda a calcular valores como el IMC (Índice de Masa Corporal), el índice cintura-cadera, pesos teóricos, entre otros; mostrando todos estos resultados de forma gráfica.

Además de las anteriormente mencionadas existen otras dos aplicaciones creadas por la Organización Mundial de Salud, llamadas WHO Anthro y WHO Anthroplus, las cuales determinan el estado nutricional de pacientes pediátricos con edades comprendidas entre 0 a 19 años de edad, mediante el monitoreo del crecimiento del mismo, debido a que este crecimiento está condicionado principalmente por la alimentación; sin embargo aunque estas herramientas son útiles para determinar el estado nutricional del paciente pediátrico de acuerdo a su antropometría, estas no permiten determinar los requerimientos del paciente ni realizar un plan dieto terapéutico para el mismo.

En este sentido, si bien existen diferentes herramientas para realizar la evaluación nutricional o los planes terapéuticos de los pacientes, ninguna de ellas integra todos los procedimientos y funcionalidades necesarias para realizar en una sola aplicación todo lo concerniente a la evaluación nutricional y posterior tratamiento del paciente, por ende existe la necesidad de crear una aplicación de fácil uso y acceso que además sea adaptable a las necesidades de los profesionales del área de nutrición, principalmente en Venezuela.

Por lo antes dicho, se busca crear una aplicación en la que los mismos profesionales puedan almacenar, consultar y reemplazar la información de todos sus pacientes, realizar los análisis pertinentes, calcular de manera automática valores que de manera manual tomen mucho tiempo realizar, capaz de presentar los resultados de forma concisa y fácil de entender, y que les facilite luego a los nutricionistas la elaboración de dietas y menús que satisfagan las necesidades de sus pacientes, ya que en la actualidad todos estos procedimientos se realizan de manera manual, lo que conlleva a invertir en grandes cantidades de material de papelería y archivos, lo que con el tiempo puede volverse un proceso desorganizado y tedioso de manejar.

1.2 Objetivos

1.2.1 Objetivo General

- Desarrollar una aplicación web para apoyar a los profesionales del área de nutrición en la creación y desarrollo de planes dieto terapéuticos personalizados y en el seguimiento de los pacientes.

1.2.2 Objetivos Específicos

- Analizar los requerimientos funcionales y no funcionales de la aplicación a desarrollar.
- Conocer los conceptos, metodologías y procesos utilizados en el área de nutrición y dietética.
- Diseñar e implementar la base de datos donde se almacenan los datos de los pacientes, usuarios y las propiedades de los alimentos.
- Diseñar las interfaces de la aplicación web que permiten almacenar y gestionar, de manera fácil e intuitiva, tanto la información de los pacientes como sus planes dieto terapéuticos.
- Implementar las funcionalidades necesarias de la aplicación web.
- Realizar pruebas funcionales y aceptación de la aplicación web.

1.3 Arquitectura de la solución

La arquitectura seleccionada es cliente-servidor. El cliente, que en este caso es cualquier navegador web, inicia la comunicación con un servidor web, a través de la red, realizando una petición al servidor de la página web y este servidor web se encargará de emitir la respuesta devolviendo la página web solicitada por el cliente, esta arquitectura será explicada con más profundidad en el marco conceptual de este trabajo.

Las herramientas seleccionadas para el desarrollo de la aplicación web son el lenguaje PHP y el marco de trabajo Laravel. Estas tecnologías se seleccionaron por las siguientes razones:

-
- PHP y Laravel son fáciles de aprender y usar.
 - PHP es uno de los lenguajes más utilizados en el desarrollo web.
 - PHP es multiplataforma y compatible con diversidad bases de datos.
 - Laravel es uno de los marcos de trabajo de PHP más usados por lo que cuenta con una muy buena documentación y una amplia comunidad de desarrolladores colaborativos.
 - Laravel brinda una serie de herramientas que facilitan el desarrollo de aplicaciones web y provee una estructura bastante cómoda para la organización del proyecto.

La base de datos que contiene toda la información de la aplicación web se diseña con el Sistema Manejador de Bases de Datos MySQL, debido a que éste es fácil de configurar e instalar, se puede ejecutar en una máquina con escasos recursos, no es muy costoso y principalmente porque soporta gran variedad de sistemas operativos realizando a gran velocidad las operaciones necesarias. Además, por su conectividad, velocidad y seguridad MySQL Server suele ser apropiado para bases de datos que trabajen con aplicaciones web. Otra de las razones por las que se selección MySQL como Sistema Manejador de Bases de Datos para la aplicación es su fácil integración y uso con el marco de trabajo Laravel.

En la base de datos se carga los datos sobre los alimentos que pueden usar los nutricionistas para la realización de los planes dieto-terapéuticos. Esta información es proporcionada por profesionales del área y pasa por un proceso de extracción y transformación antes de ser cargada a la base de datos. La elaboración de las vistas de la aplicación web se realiza utilizando HTML, CSS3 y JavaScript, que son lenguajes que se ejecutan e interpretan del lado del cliente sin realizar peticiones al servidor web. En la figura 1.1 se muestra la arquitectura de la aplicación web.

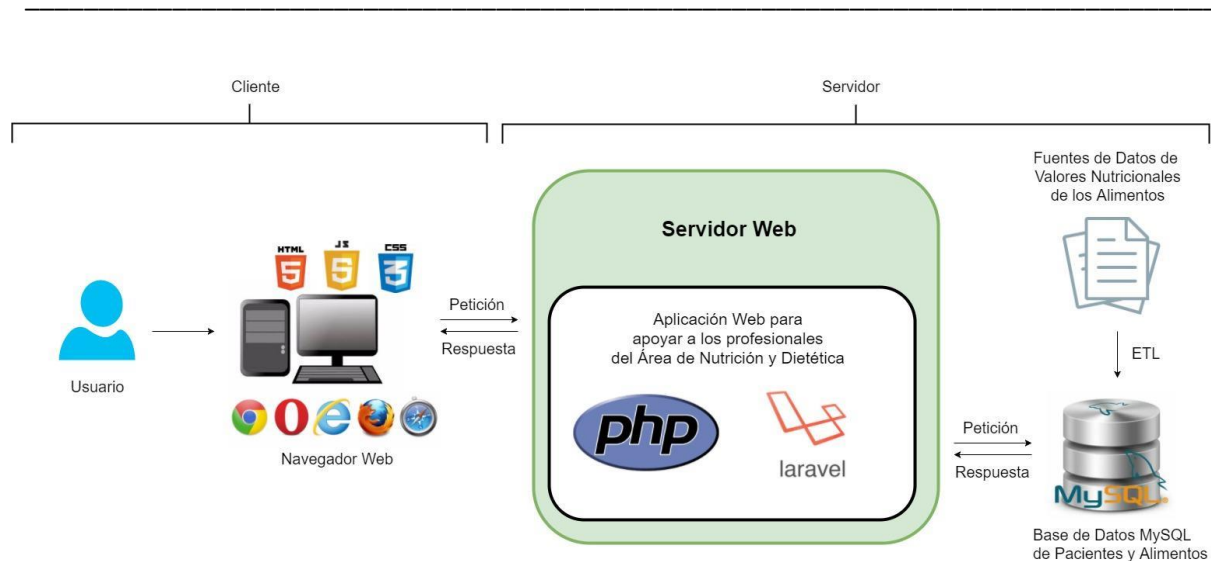


Figura 1.1 – Arquitectura de la Solución

1.4 Justificación de la solución

Esta aplicación permite a los profesionales del área de nutrición realizar actividades como: calcular requerimientos nutricionales de cada paciente 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); hacer seguimiento del estado nutricional de los mismos, todo esto de una manera rápida y eficiente; haciendo un seguimiento oportuno y veraz de cada uno de los pacientes sin importar la patología o condición de base que este posea.

Las aplicaciones web tienen múltiples ventajas a la hora de permitir acceso a distintas herramientas de trabajo desde un solo lugar de manera inmediata y desde cualquier lugar, gracias a los dispositivos inteligentes, evitando el traslado y uso innecesario de otras herramientas como: hojas de papel, calculadoras y lápices. Tener una aplicación web que permita a los nutricionistas crear y gestionar las dietas de sus pacientes sería de gran ayuda, ya que facilita de gran manera su trabajo y les ahorra tiempo que podría ser empleado de mejor manera, interactuando con sus pacientes para conocer mejor sus necesidades; además puede utilizarse para optimizar tiempo a la hora de realizar diagnósticos y entregas de dieta a cada uno de los pacientes atendidos; también puede utilizarse fuera del consultorio sin necesidad de trasladar carpetas y archivos con esta información (debido a que toda esta se encuentra en el dispositivo inteligente) y por último, esta puede ser utilizada para realizar evaluación nutricional en grandes comunidades donde se atienden más de 50 pacientes por día, haciendo más rápido y efectivo el trabajo del nutricionista.

CAPÍTULO 2 - MARCO CONCEPTUAL

En este capítulo se trata en detalle el estado actual de aplicaciones relacionadas con la gestión de pacientes en el área de la nutrición y dietética, así como se abordan conceptos relacionados con el desarrollo de aplicaciones web tales como arquitectura, funcionamiento, tecnologías utilizadas y sistemas manejadores de base de datos.

Antes de conocer los antecedentes de esta investigación, es necesario conocer la metodología y procedimientos que deben seguir los nutricionistas para realizar un plan dieto terapéutico personalizado. Lo primero que el nutricionista debe hacer es la evaluación nutricional del paciente mediante un interrogatorio que le permite registrar todos sus datos personales, así como antecedentes familiares y personales, también registra la antropometría del mismo (peso, talla, circunferencias, diámetros y pliegues) y por último registra los valores bioquímicos previamente realizados por un laboratorio; todo esto con el fin de arrojar un diagnóstico nutricional integral.

Una vez obtenido el diagnóstico del paciente se procede a determinar el requerimiento calórico nutricional del mismo, el cual indica la cantidad de energía en forma de kilocalorías que será cubierto por los alimentos consumidos durante todas las comidas del día (generalmente son: 3 comidas principales y dos meriendas). Posteriormente se procede a elaborar la fórmula dietética nutricional, la cual consiste en dividir el 100% del requerimiento en los macronutrientes con lo cual se debe garantizar un aporte de: proteínas (15%), grasas (30%) y carbohidratos (55%). Posteriormente este debe convertirse en kilocalorías y en gramos de alimento que representa cada uno.

Una vez obtenido esto, se debe calcular la cantidad de cada tipo de alimento (leche y derivados, vegetales, frutas, cereales, carnes y grasas), que se deben proporcionar para cubrir las kilocalorías determinadas con anterioridad; finalmente se procede a realizar de acuerdo a la cantidad de cada alimento a proporcionar el plan dieto-terapéutico o dieta, que no es más que la combinación de todos estos alimentos en preparaciones atractivas y distribuidas de manera adecuada a lo largo del día, según la preferencia de los pacientes y sus alergias o intolerancias alimentarias, con el fin de evitar ayunos y permitir un consumo constante de alimentos garantizando el aporte energético adecuado para el paciente.

Los procedimientos anteriormente descritos para la realización del plan dieto-terapéutico del paciente, hoy en día se siguen realizando de manera manual, mediante la aplicación y uso de distintas fórmulas para determinar el requerimiento calórico, la distribución de nutrientes y las cantidades de alimentos a proporcionarle a cada paciente. Este proceso puede volverse tedioso además de generar gran cantidad de carpetas con archivos de cada uno de los pacientes que son atendidos.

2.1 Antecedentes

A continuación, se explican con detalles las aplicaciones existentes mencionadas anteriormente que se analizan durante este trabajo de investigación.

2.1.1 Alimentación Requerimiento Nutricional y Adecuación CANIA (ARNAC)

ARNAC es una aplicación web desarrollada por la empresa Manapro para el Centro de Atención Nutricional Infantil Antímano (CANIA) perteneciente a Empresas Polar. ARNAC es una herramienta para la evaluación de consumo y prescripción dietética orientada exclusivamente a la población infantil, es decir, niños con edades comprendidas entre los 0 y los 18 años. Esta aplicación puede ser accedida desde el año 2016 sólo por aquellos nutricionistas y médicos nutriólogos a los que CANIA les otorgue credenciales de acceso, lamentablemente la misma brinda funcionalidades muy limitadas ya que esta diseñada para tratar exclusivamente a pacientes pediátricos, dejando así por fuera muchas herramientas y funcionalidades que el profesional en nutrición podría utilizar para simplificar su trabajo. En la Figura 2.1 se presenta la pantalla de inicio de la aplicación.



Figura 2.1 – Pantalla de inicio de sesión.

2.1.2 NUTRIMIND

Nutrimind es una aplicación tanto web como de escritorio creada en el año 2007 por especialistas del área, originaria de México, esta aplicación está diseñada para apoyar al nutricionista durante la creación de dietas y control de pacientes. Se comercializa con gran éxito en toda América latina y España.

Para complementar la creación y planificación de dietas, la aplicación ofrece todos los cálculos necesarios para realizar el menú del paciente, así como también ofrece preparaciones alternativas que cumplan con los requerimientos que establece la dieta creada.

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 provee funcionalidades sencillas como la inclusión de fotografías de cada paciente para fácil reconocimiento, instrucciones que muestran donde se realiza cada medición según los estándares internacionales, generación de gráficos de evolución de todas las mediciones antropométricas, signos vitales, calorías en la dieta, entre otras.

Al ver todo esto podemos decir que efectivamente es una herramienta bastante completa con muchas funcionalidades que permiten a los profesionales del área realizar su trabajo de una manera más cómoda y rápida, lamentablemente para su uso se debe adquirir una licencia, además que en la versión de escritorio solo se permite su instalación en dos computadores, por lo que instalarlo en un centro médico, donde no se disponga de una buena conexión a internet, el costo de la licencia puede generar una inversión bastante grande dependiendo de la cantidad de computadoras en las que desee utilizarse. El pago debe hacerse en dólares.

Otro problema 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 podrá hacerlo lo cual limita su uso a las funcionalidades provistas con su descarga.

A continuación, en las Figuras 2.2, 2.3, 2.4 y 2.5 se muestran algunas de las principales interfaces de Nutrimind:

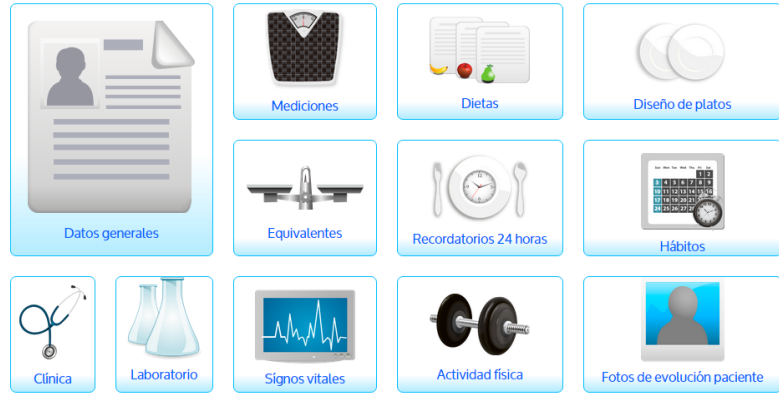


Figura 2.2 - Menú Principal de Nutrimind



Figura 2.3 - Requerimiento Nutricional del Paciente



Figura 2.4 - Ventana de Creación de menú

Nombre: Grupo alimenticio:

Nutrientes por cada 100g de alimento					
Carbohidratos	<input type="text" value="64.3"/> g	Proteínas	<input type="text" value="6.8"/> g	Lípidos	<input type="text" value="0"/> g
Agua	<input type="text" value="0"/> g	Fibra	<input type="text" value="2.1"/> g	Calcio	<input type="text" value="0"/> mg
Hierro	<input type="text" value="0"/> mg	Magnesio	<input type="text" value="0"/> mg	Fósforo	<input type="text" value="0"/> mg
Potasio	<input type="text" value="0"/> mg	Sodio	<input type="text" value="1107"/> mg	Zinc	<input type="text" value="0"/> mg
Manganeso	<input type="text" value="0"/> mg	Flúor	<input type="text" value="0"/> mg	Selenio	<input type="text" value="0"/> mg
Vitamina A	<input type="text" value="0"/> µg	Vitamina B1	<input type="text" value="0"/> µg	Vitamina B2	<input type="text" value="0"/> µg
Vitamina B3	<input type="text" value="0"/> µg	Vitamina B5	<input type="text" value="0"/> µg	Vitamina B6	<input type="text" value="0"/> µg
Vitamina B9	<input type="text" value="0"/> µg	Vitamina B12	<input type="text" value="0"/> µg	Vitamina C	<input type="text" value="0"/> µg

Figura 2.5 - Pantalla de creación de alimentos

2.1.3 WHO Anthro

Es un software para el uso en computadoras personales de escritorio o laptops que utilicen Windows MS. Se ha desarrollado para facilitar la aplicación de los estándares de crecimiento de la OMS (Organización Mundial de la Salud o por sus siglas en inglés WHO) para el monitoreo del crecimiento y desarrollo motor en niños individualmente y poblaciones de niños hasta los 5 años de edad. Se publicó una primera versión de este software en el 2006 en conjunto con el primer lanzamiento de los estándares de crecimiento de la OMS (peso para la edad, talla para la edad, peso para la edad, IMC para la edad y logro de seis hitos motores gruesos). En el 2007 la OMS publicó el segundo conjunto de estándares para los indicadores de perímetro cefálico para la edad, perímetro medio del brazo para la edad, pliegues tricótipal y subescapular para la edad; desencadenando la necesidad de actualizar el software. Esto dio la oportunidad de mejorar más el software, incluyendo las versiones en los idiomas francés y español, así como una opción de ayuda en línea. WHO Anthro consiste de tres módulos:

- Calculador antropométrico
- Examen Individual
- Encuesta Nutricional

En la Figura 2.6 se presenta el menú principal de la aplicación.

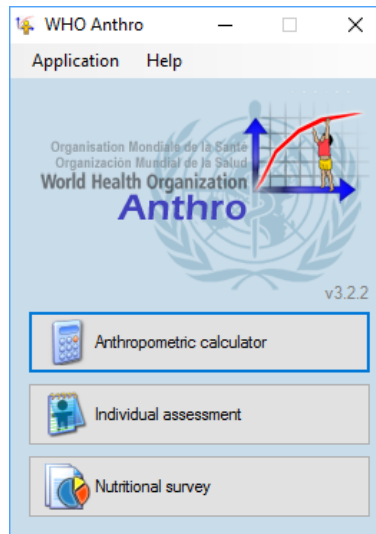


Figura 2.6 - Pantalla Principal

En la figura 2.7 podemos ver la pantalla para capturar los datos antropométricos del paciente.

Anthropometric calculator

Help

Date of visit: 21/ 4/2017

Sex: Female Male

Date of birth: 21/ 4/2016

Age: 11mo

Weight (kg): 9.00

Length/height (cm): 73.00

BMI: 16.9

Measured: Recumbent Standing

Oedema: No Yes

Head circumference (cm): 45.00

MUAC (cm): 15.00

Triceps skinfold (mm): 8.00

Subscapular skinfold (mm): 7.00

Results

Measurement	Percentile	z-score
Weight-for-length	61.4	0.29
Weight-for-age	51.9	0.05
Length-for-age	34.8	-0.39
BMI-for-age	64.1	0.36
HC-for-age	53.1	0.08
MUAC-for-age	74.3	0.65
TSF-for-age	49.9	0.00
SSF-for-age	65.0	0.38

Figura 2.7 - Pantalla para introducir los datos antropométricos del paciente

Cada módulo proporciona funciones específicas (Figura 2.8 y 2.9), es decir, permite evaluar el estado nutricional de un niño, dar seguimiento del desarrollo y crecimiento de un niño a través del tiempo o evaluar encuestas nutricionales, respectivamente.

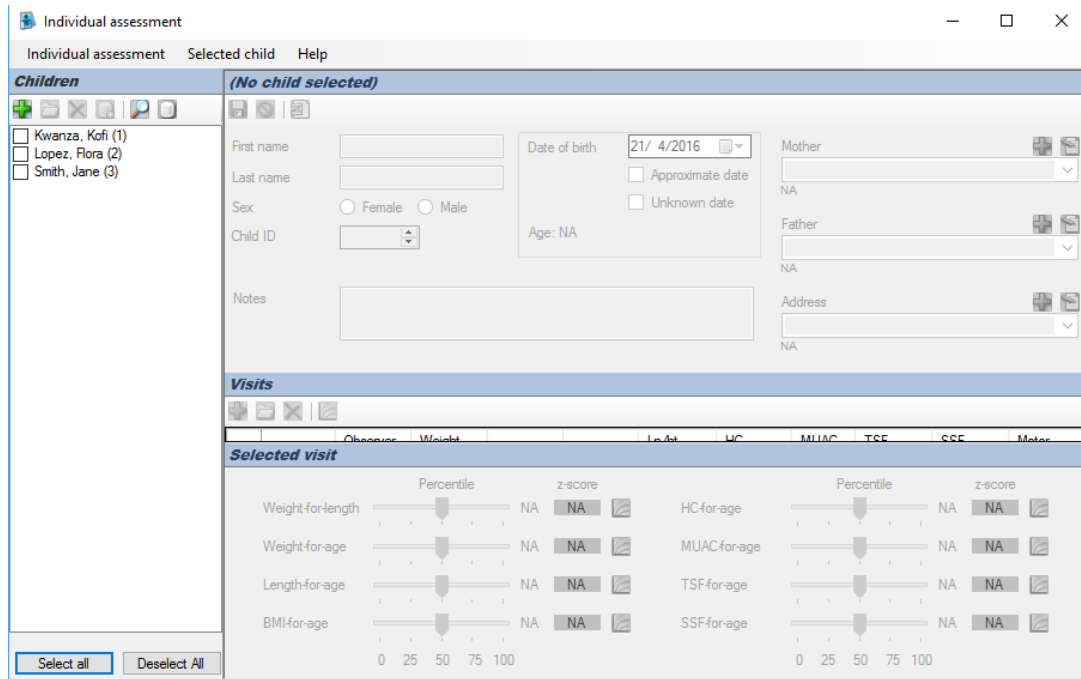


Figura 2.8 - Seguimiento de Evolución de Paciente

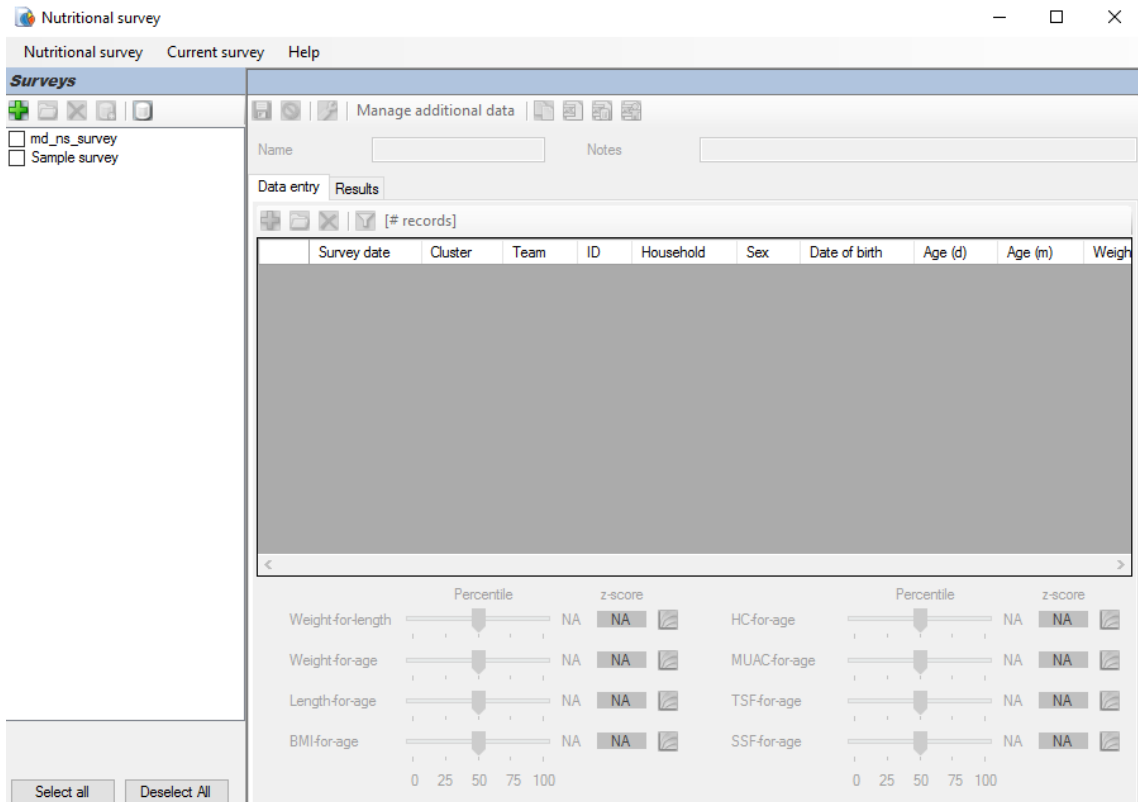


Figura 2.9 - Encuesta Nutricional

2.1.4 WHO AnthroPlus

WHO AnthroPlus es un software para usar en ordenadores personales de escritorio o portátiles que utilizan MS Windows. Fue desarrollado para facilitar y ampliar la aplicación de la OMS 2007 (Anthro, anteriormente explicada) ésta abarca la población entre 5-19 años (población que no estaba incluida en la aplicación Anthro), esta aplicación es usada para evaluar el crecimiento de los niños y adolescentes en edad escolar y adolescentes. Para mostrar la continuidad con los estándares del programa de crecimiento infantil de la OMS 0-5años (Anthro) que se incluyen en AnthroPlus para los tres indicadores que se aplican, es decir, peso para la edad, estatura para la edad e IMC para la edad. Este software permite el monitoreo del crecimiento en individuos y poblaciones de niños desde el nacimiento hasta los 19 años de edad. WHO Anthro, a su vez, es un software que se publicó en 2006 junto con el primer conjunto de estándares de crecimiento infantil de la OMS (es decir, peso por edad, altura por edad, peso por estatura, IMC por edad). En 2008, la WHO Anthro se actualizó para incluir el segundo conjunto de indicadores de crecimiento alcanzado: circunferencia de la cabeza para la edad, circunferencia del brazo para la edad, tríceps y pliegue de la piel subescapular para la edad, y permitir a los usuarios elegir un francés o español Idioma. Esta aplicación permite a los usuarios importar datos previamente recogidos en la OMS Anthro y seguir monitoreando el peso de los niños, el índice de masa (Módulo de Evaluación Individual) y analizar datos de encuestas incluyendo Preescolar, niños en edad escolar y adolescentes. Así, AnthroPlus Facilita la detección de delgadez, bajo peso, sobrepeso y obesidad en individuos y poblaciones De 0-19 años. Además, incluye la opción de cambiar del idioma inglés predeterminado a francés, español y ruso.

Al ser ambas aplicaciones (WHO Anthro y WHO AnthroPlus) diseñadas por la misma institución (la OMS), las interfaces son idénticas en ambos casos, la única diferencia es el rango de edades para los cuales se utiliza la aplicación.

2.1.5 Método manual

Si bien, anteriormente fueron explicados diferentes aplicaciones para determinar el estado nutricional o realizar planes terapéuticos para los pacientes; en Venezuela aún los nutricionistas siguen utilizando el método manual para realizar los cálculos para determinar el estado nutricional del paciente y posteriormente realizar la prescripción dietética y plan nutricional, debiendo utilizar así; hojas blancas, calculadoras y bolígrafos. Sin embargo, cuentan con modelos estandarizados que permiten realizar esta tarea de manera más eficiente.

Algunas fórmulas (Figura 2.10, Figura 2.11) para determinar el estado nutricional son:

FORMULAS		
PESO IDEAL O TEÓRICO		
1. Hamwi:		
$\text{♂ Hombres: } \{ \text{Talla (cm)} - 152 \} \times 1,08 + 48,5$ $\text{♀ Mujeres: } \{ \text{Talla (cm)} - 152 \} \times 0,88 + 45,5$		
Contextura (Grant) = Talla (cm) / C. Muñeca		
Contextura	Hombres	Mujeres
Pequeña	>10,4	>11
Mediana	9,6 – 10,4	10,1 - 11
Grande	< 9,6	< 10,1
2. Brocca: Talla <152		
P.I = Talla -100		
Se debe corregir solo para Contextura Pequeña donde se resta de un 5-10% (5% preferiblemente)		
3. Clínica Mayo; West:		
$\text{♂ Hombres: } 22,1 \times (\text{Talla})^2 \text{ (mts)}$ $\text{♀ Mujeres: } 20,6 \times (\text{Talla})^2 \text{ (mts)}$		
4. Por IMC:		
Para Pacientes Bajo Peso		
P.I= 18,5 x (Talla)² (mts)		
Para Pacientes Sobrepeso u Obesidad		
P.I= 24,9 x (Talla)² (mts)		
El promedio es:		
P.I= 21,7 x (Talla)² (mts)		
5. Peso Ajustado (Obesos)		
PAjustado = $\{ (\text{Pactual} - \text{Pideal}) \times 0,25 \} + \text{Pideal}$		

Figura 2.10 - Fórmulas Peso Ideal/Teórico. Fuente: Escuela de Nutrición y Dietética UCV

2. Ecuación Harris-Benedict:	
HOMBRES: $MB = 66,47 + (13,75 \times \text{Peso Actual}) + (5 \times \text{Talla (cm)}) - 6,76 \times \text{Edad}$ MUJERES: $MB = 655,1 + (9,56 \times \text{Peso Actual}) + (1,85 \times \text{Talla (cm)}) - (4,8 \times \text{Edad})$	
RCT= MB x FA x FS	
Factor de Actividad:	
Pacientes Encamados: 1,2	
Pacientes Deambulando: 1,3	
Factor de Estrés:	
Cirugía Electiva: 1-1,2	
Peritonitis: 1,2-1,5	
Trauma de tejidos Blandos: 1,1-1,3	
Fracturas múltiples: 1,2-1,3	
Cáncer: 1,2	
Neumonía: 1,2	
Nota:	
Peso Actual si el IMC < 40 kg/m ²	
Peso Ideal si el IMC > 40 kg/m ²	
Embarazo: 1,1	
Crecimiento y Lactancia: 1,4	
Sepsis: 1,4-1,8	
Quemadura 0-20%: 1-1,5	
Quemadura 20-40%: 1,5-1,8	
Quemadura 40-100%: 1,8-2	

Figura 2.11 - Ecuación Harris-Benedict. Fuente: Escuela de Nutrición y Dietética UCV

Todas estas fórmulas son calculadas de manera manual, ya que no existe una aplicación que las determine de manera automática, posteriormente todos los resultados son pasados a una hoja en blanco donde se realiza la historia del paciente, y donde se procede a realizar la fórmula dietética que consiste en la distribución porcentual de las kilocalorías al día por macronutriente (proteínas, grasas y carbohidratos).

Posteriormente se procede a distribuir el requerimiento en los alimentos que conforman la dieta del paciente, esto se realiza con el siguiente formato impreso (Figura 2.12):

TABLA PARA EL CÁLCULO DE DIETAS POR EL SISTEMA DE INTERCAMBIOS													
Pacite. _____ años		RCT: _____ kcal	Prot. _____ g		GT. _____ g	CHO _____ g		Fibra _____ g		Na _____ mEq			
Alimento según lista de	n° Inte	Proteínas (gramos)	Grasas (gramos)	CHO (gramos)	Sodio (mEq)				Fibra (gramos)				
Leche 1%	8	2,5	8	12	Cfeadiv	5				X	X	X	X
Leche entera					Cfeadiv	6							
Vegetales / Hortalizas	2	X	X	5	Sfeadiv	0,5				Licuada o puré	0,5		
					Cfeadiv	2							Entera
Frutas	X	X	X	15	X	X	X	X		Licuada o puré	0,5		
					Cfeadiv	10							Entera
SUBTOTAL													
Panes	2	0,8	X	15	Sfeadiv	0,5				Refinado	0,5		
					Cfeadiv	5							Integral
Cereales y harinas	2	X	X	15	Sfeadiv	0,5				Refinado	0,5		
					Cfeadiv	1,5							Integral
Alimentos preparados					Cfeadiv	15				X	X	X	X
Tubérculos o Plátano	6	X	X	15	X	X	X	X		Licuada o puré	1,3		
					X	X	X	X					Entera
Leguminosas					X	X	X	X		Licuada o puré	1,3		
SUBTOTAL													
Carnes magra (hasta 10% de grasa)	7	3	5	X	Sfeadiv	0,5							
					Cfeadiv	5							
Carnes semigrasa (menos de 17% de grasa)	7	5	X	X	Sfeadiv	0,5				X	X	X	X
					Cfeadiv	5							
Carnes gordas (más de 17% de grasa)	7	8	X	X	Sfeadiv	0,5							
					Cfeadiv	5							
					Embalsamadas	18							
SUBTOTAL													
Grasas: aceites, margarina, etc	X	X	5	X	X	X	X	X		X	X	X	X
					Cfeadiv	2							
TOTAL													

Figura 2.12 - Tabla para el cálculo de dietas por sistema de intercambios. Fuente: Escuela de Nutrición y Dietética UCV

Finalmente, la distribución de alimentos determinada con anterioridad se plasma en otro formato impreso (Figura 2.13) donde se plasma el plan nutricional del paciente:

Nombre: _____ Historia: _____

PLAN DE ALIMENTACION

Prescripción:

Calorías: _____ Proteínas: _____% _____ g. Carbohidratos: _____% _____ g. Grasas: _____% _____ g.

DESAYUNO:

- _____ Vaso de leche descremada o yogurt descremado (_____ cc)
- _____ Rebanada de pan o _____ canilla o _____ arepa pequeña o _____ paq. de galletas de soda
- _____ Rebanada de queso suave, jamón de pavo o pollo o _____ huevo (no frito)
- _____ Cucharadita de margarina suave o líquida
- _____ Ración de Frutas

MERIENDA: _____

ALMUERZO:

- _____ Taza de arroz, pasta o granos o _____ plátano (no frito) o _____ trozos de ocumo, papa, yuca, batata o apio
- _____ Gr. o _____ cdas. de pollo o pavo sin piel o pechado o carne de res (magra)
- _____ Ración de Vegetales
- _____ Cucharadita de aceite de oliva o canola o _____ rebanada de aguacate
- _____ Ración de Frutas

MERIENDA: _____

CENA:

- _____ Taza de arroz, pasta o granos o _____ plátano (no frito) o _____ paq. de galletas de soda o _____ rebanada de pan o canilla o _____ arepa
- _____ Ración de Vegetales
- _____ Cucharadita de aceite de oliva o canola o _____ rebanada de aguacate
- _____ Ración de Frutas

MERIENDA: _____

Raciones:

- **Vegetales:** 1 ración
1/4 taza: auyama, zanahoria, guisante, cebolla, vainitas, hinojo, remolacha, nabo
1 taza: acelga, ajo porro, alcachofa, alfalfa, berenjena, berro, brócoli, champiñones, cebollín, celery, chayota, coliflor, espárragos, espinaca, escarola, lechuga, pepino, perejil, pimentón, rábano, repollo o tomate
- **Frutas:** 1 ración
1/4 unidad: cambur, toronja; 1 unid. Mediana: naranja, mandarina, níspero, mango, manzana, guayaba, melocotón, pera, ciruela americana, granada; 2 unid: durazno, higos, parchita; 3 unid: ciruela pasa; 5 unid: mamón, ciruela de huesito; 15 unid: uvas; 1 taza: patilla, lechosa, melón, fresas, mora; 1/2 taza de guanábana. Jugo de naranja, jugo de toronja; 1 rueda delgada de piña;

Figura 2.13 - Menú Ejemplo. Fuente: Escuela de Nutrición y Dietética UCV

Todos estos datos son llenados de manera manual a cada uno de los pacientes que se atienden, en estos formatos ya preestablecidos.

2.2 Conceptos Básicos relacionados con el área de Nutrición

Nutrición: La nutrición es la ingesta de alimentos en relación con las necesidades dietéticas del organismo. Una buena nutrición (una dieta suficiente y equilibrada combinada con el ejercicio físico regular) es un elemento fundamental de la buena salud.

Debido a esto nace la nutrición como una posible solución, la cual mediante la evaluación nutricional, planes terapéuticos y posterior desarrollo del menú o plan dieto-terapéutico, permiten encontrar un equilibrio en la ingesta de alimentos tanto en calidad y cantidad apropiada para asegurar el correcto desarrollo y funcionamiento del organismo a la vez que controlan y reducen el riesgo a padecer enfermedades no transmisibles del adulto también llamadas enfermedades relacionadas con la nutrición.

Nutricionista: es un profesional de la salud, con titulación universitaria, reconocido como un experto en alimentación, nutrición y dietética, con capacidad para intervenir en la alimentación de una persona o grupo, desde los siguientes ámbitos de actuación: la nutrición en la salud y en la enfermedad, el consejo dietético, la investigación y la docencia, la salud pública desde los organismos gubernamentales, las empresas del sector de la alimentación, la restauración colectiva y social.

El objetivo principal del nutricionista como profesional es garantizar una adecuada alimentación y nutrición a cada paciente según la patología asociada al mismo y según sus requerimientos.

Requerimiento nutricional: Los requerimientos nutricionales son las necesidades que los organismos vivos tienen de los diferentes nutrientes para su óptimo crecimiento, mantenimiento y funcionamiento en general.

Plan dieto-terapéutico: Tienen como objetivo reestablecer los posibles desequilibrios bioquímicos y metabólicos a los que se enfrenta el organismo y que en ocasiones pueden alterar el bienestar físico, esto es puesto en práctica mediante la prescripción de preparaciones distribuidas a lo largo día.

En nuestra investigación no hemos encontrado en Venezuela desarrollo de aplicaciones web que ayuden a los nutricionistas a crear y desarrollar planes dieto terapéuticos personalizados y que les permitan hacer seguimiento a la evolución del paciente. En vistas generales solo se ha desarrollado una aplicación web que quiera cubrir esa necesidad llamada Nutrimind de la cual no se puede obtener mucha información debido a que la herramienta es paga y totalmente privativa. Por ello a continuación haremos un análisis conceptual de lo requerido para desarrollar una aplicación como esta.

2.3 Arquitectura cliente-servidor

La arquitectura cliente-servidor es una arquitectura de red en la que cada ordenador o proceso en la red es cliente o servidor (Luján, Sergio, 2002), por lo que ambos, cliente y servidor, son los componentes principales de la misma. En esta arquitectura, el cliente inicia la comunicación con el servidor a través de la red solicitando recursos o haciendo peticiones de servicios, como puede ser solicitar un archivo, mientras que el servidor se encarga de atender dichas solicitudes, respondiendo con el recurso o la información solicitada (Pérez, 2016).

Los servidores suelen ser ordenadores potentes, es decir, con una gran capacidad de procesamiento y almacenamiento, dedicados a gestionar unidades de disco (servidor de ficheros), impresoras (servidor de impresoras), tráfico de red (servidor de red), datos (servidor de bases de datos) o aplicaciones (servidor de aplicaciones), mientras que los clientes suelen ser máquinas menos potentes (Luján, Sergio, 2002) aunque con la evolución de la tecnología en los últimos años un cliente podría llegar a ser igual o más potente que un servidor.

Dentro de los clientes se suelen distinguir dos clases: los clientes inteligentes, conocidos como “rich client”, y los clientes tontos conocidos como “thin client”. Los clientes inteligentes son ordenadores completos, con todo el hardware y software necesarios para poder funcionar de forma independiente. En cambio, los clientes tontos son terminales que no pueden funcionar de forma independiente, ya que necesitan de un servidor para ser operativos.

En la figura 2.14 se puede observar la representación de un cliente-servidor:

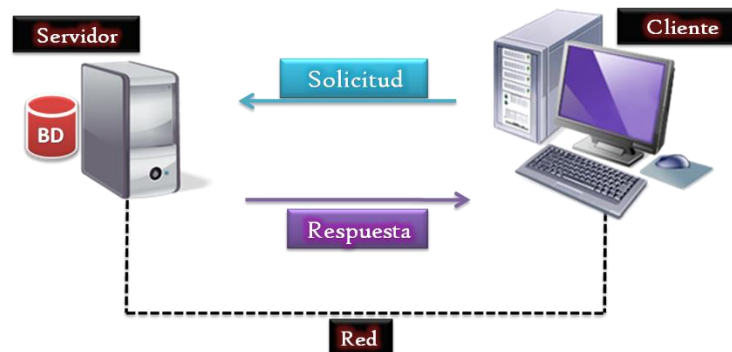


Figura 2.14 - Arquitectura cliente-servidor.

2.4 Aplicación web

Una aplicación web (web-based application) es un tipo especial de aplicación cliente/servidor, donde tanto el cliente (el navegador o explorador) como el servidor (el servidor web) y el protocolo mediante el que se comunican (HTTP) están estandarizados y no han de ser creados por el programador de aplicaciones (Luján, Sergio, 2002).

2.4.1 Funcionamiento de una aplicación web

Las aplicaciones web se basan en la previamente explicada arquitectura cliente-servidor, donde en este caso, el cliente es un navegador web como lo son Google Chrome y Mozilla Firefox, y el servidor es un servidor web que procesa toda la lógica de la aplicación y almacena las páginas HTML de la misma.

Una aplicación Web es un conjunto de páginas Web estáticas y dinámicas. Una página Web estática es aquella que no cambia cuando un usuario la solicita: el servidor Web envía la página al navegador Web solicitante sin modificarla. Por el contrario, el servidor modifica las páginas Web dinámicas antes de enviarlas al navegador solicitante. La naturaleza cambiante de este tipo de página es la que le da el nombre de dinámica (Adobe Dreamweaver, 2016).

A continuación, en la Figura 2.15 se muestra el proceso de petición de una página estática a un servidor web.

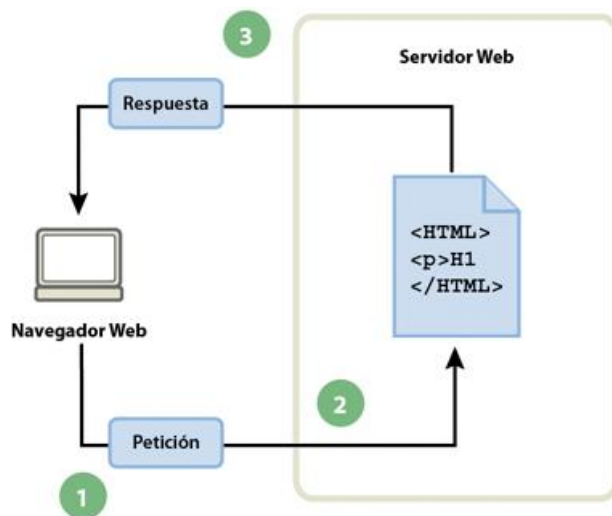


Figura 2.15 - Petición de una página web estática.

1. El navegador Web solicita la página estática.
2. El servidor localiza la página.
3. El servidor Web envía la página al navegador solicitante.

A diferencia de las páginas estáticas, cuando el servidor recibe una petición de una página dinámica, el mismo reacciona de manera distinta: se procederá a transferir la página a un software especial que se encargará de finalizar la página, este software se denomina servidor de aplicaciones. El servidor de aplicaciones en función de las instrucciones del código que posee la página completará la misma y el resultado será una página estática en formato HTML que el servidor de aplicaciones devolverá al servidor web, la cual será finalmente enviada al navegador solicitante

El servidor de aplicaciones también permite trabajar con recursos del lado del servidor, como las bases de datos. Sin embargo, el servidor de aplicaciones no puede comunicarse con la base de datos directamente, necesita de un controlador que actúe como intermediario para establecer la comunicación con ésta.

A continuación, en la Figura 2.16 se muestra el proceso completo de petición de páginas dinámicas de una aplicación web; a diferencia de las páginas estáticas con como mencionamos antes ya poseen todo el contenido dentro del HTML solicitado, los datos de las páginas web dinámicas usualmente son extraídos de una base de datos.

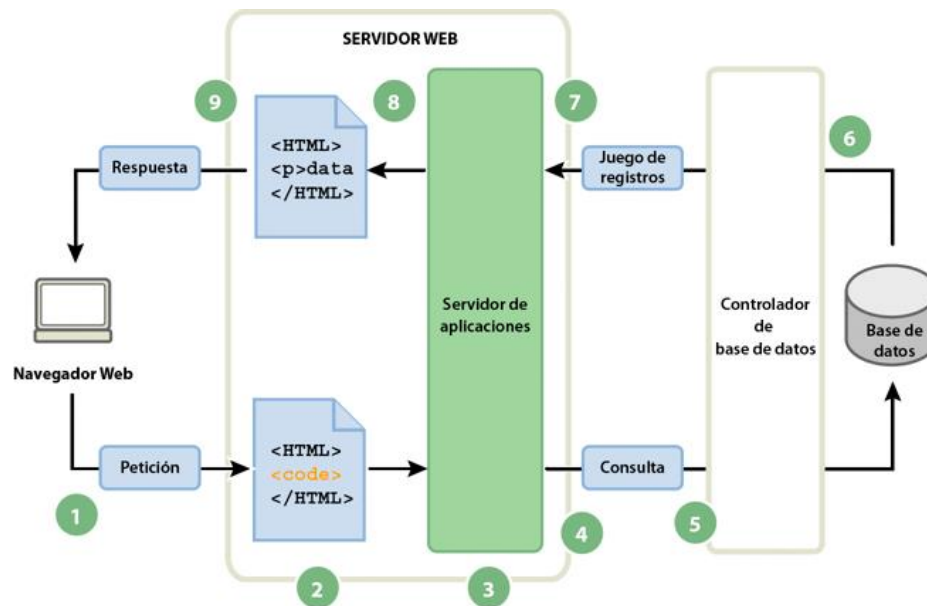


Figura 2.16 - Petición de una página web dinámica con datos extraídos de una base de datos.

-
1. El navegador Web solicita la página dinámica.
 2. El servidor Web localiza la página y la envía al servidor de aplicaciones.
 3. El servidor de aplicaciones busca instrucciones en la página.
 4. El servidor de aplicaciones envía la consulta al controlador de la base de datos.
 5. El controlador ejecuta la consulta en la base de datos.
 6. El juego de registros se devuelve al controlador.
 7. El controlador pasa el juego de registros al servidor de aplicaciones.
 8. El servidor de aplicaciones inserta los datos en una página y luego pasa la página al servidor Web.
 9. El servidor Web envía la página finalizada al navegador solicitante.

2.4.2 Ventajas

Algunas ventajas de estas aplicaciones son:

- Desde el punto de vista del usuario, no es necesario manejar actualizaciones, ya que éstas son aplicadas del lado servidor, esto implica que el usuario final siempre tendrá la última versión de la aplicación disponible.
- Al ser ejecutadas a través de un navegador, están disponibles en todo momento, ya que se puede acceder a ellas a través de cualquier dispositivo que se encuentre conectado a la red.
- Posee una alta compatibilidad entre plataformas, dado que se ejecutan a través de un navegador.
- Poseen un bajo consumo de recursos locales ya que todas las tareas y procedimientos se realizan en el servidor.

2.4.3 Desventajas

Algunas de las desventajas de estas aplicaciones son:

- El navegador sobre el cual se ejecuta la aplicación debe estar correctamente configurado para trabajar de manera correcta, debemos considerar cosas como la versión del mismo, las Cookies, configuración flash, entre otras.
- Como este tipo de aplicaciones están alojadas en un dominio público, son vulnerables a ataques maliciosos, por este motivo se deben tomar medidas de seguridad para resguardar los datos de los usuarios.
- Una gran afluencia de usuarios puede afectar negativamente el desempeño de la aplicación, debido a la sobrecarga de procesamiento que recae en el servidor.

2.4.4 Patrón de Modelo Vista Controlador (MVC)

El patrón de arquitectura Modelo Vista Controlador (MVC) es un patrón de arquitectura frecuentemente utilizado para el desarrollo de aplicaciones web ya que permite definir o separar los componentes de la aplicación en tres capas bien definidas y que se explican a continuación:

- **Modelo:** Esta capa se ocupa del manejo de los datos de su almacenamiento, recuperación y de mantener la consistencia e integridad de los mismos, también se deben proveer maneras de acceder a los datos en esta capa, en esta se define el modelo del negocio y es independiente de la vista y el controlador. También es importante mencionar que en esta capa suele utilizarse un ORM (Object Relational Mapping, Mapeo Objeto Relacional), el cual permite tener cierto nivel de abstracción a la hora de trabajar la base de datos, en vez de usar directamente sentencias SQL (Structured Query Language, Lenguaje de Sentencias Estructuradas), que suele depender del motor de base de datos con el que se esté trabajando (Pérez,2016).
- **Vista:** Es la presentación del modelo al usuario y permite la interacción del usuario con la aplicación, es decir, por medio de esta se presentan los datos al usuario y se obtienen los datos de entrada que ingresa el mismo para ser procesados.
- **Controlador:** Se encarga de actuar como intermediario entre la vista y el modelo, procesando los datos ingresados o las peticiones realizadas por el usuario, modificando el modelo y emitiendo los resultados generados del procesamiento de los datos. También se puede decir que interactúa con la lógica de datos para ejecutar las reglas de negocio que tiene que cumplir la aplicación (Luján, Sergio, 2002).

2.4.4.1 Flujo de Control del MVC

Para entender cómo funciona el patrón MVC en conjunto se pueden mencionar muchas variantes de flujo, sin embargo, el flujo que describe Pérez (2016) es bastante bueno para describir el funcionamiento de las tres capas en conjunto, por ello nos guiamos del mismo para explicar el siguiente flujo:

1. El usuario interactúa con la aplicación a través de la vista realizando alguna petición o acción, por ejemplo, ingresa datos a un formulario o pulsa un botón.
2. El controlador es notificado sobre la acción del usuario y gestiona el evento, usualmente mediante el uso de un gestor de eventos.

-
3. El controlador se comunica con el modelo solicitando consultar o actualizar los datos.
 4. El modelo interactúa con la base de datos de forma directa o mediante el uso de una capa de abstracción, que puede ser un ORM como se mencionó anteriormente, y realiza la actualización o consulta de los datos.
 5. Ya terminada la consulta o actualización de los datos, el modelo se comunica con el controlador enviando los datos obtenidos o notificando los cambios realizados.
 6. Por último, el controlador le envía la respuesta a la vista y esta actúa en función de la respuesta recibida (actualizar la interfaz, por ejemplo).

En la figura 2.17 se puede observar el proceso descrito anteriormente:

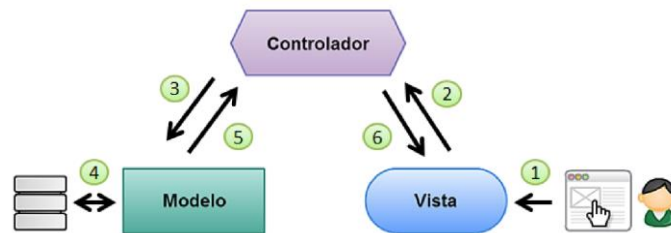


Figura 2.17 – Flujo de Control del Patrón MVC

2.4.4.2 Ventajas del MVC

El patrón MVC es utilizado regularmente para desarrollar aplicaciones web ya que muestra las siguientes ventajas principales:

- Permite que hacerle mantenimiento a la aplicación sea más fácil.
- Facilita el desarrollo, ya que la implementación se hace de forma modular.
- Se automatiza la actualización de las vistas, es decir, no tienen que ser cambiadas por el programador para que su contenido esté actualizado.
- Aumenta la escalabilidad de la aplicación.
- Las modificaciones en las vistas no tienen repercusiones en el modelo.

2.5 Tecnologías del lado del cliente

En una aplicación web hay tecnologías que son ejecutadas o interpretadas por un navegador web y de estas depende la aplicación para que posea el funcionamiento deseado y trabaje de manera eficiente, ya que esto puede variar dependiendo de la versión y el tipo de navegador que esté utilizando el usuario. Todas estas tecnologías son llamadas tecnologías del lado del cliente y son utilizadas para que mostrar la información de la aplicación web y para que el usuario pueda interactuar con la misma realizando peticiones al servidor web, también se utilizan para intentar que el formato con el que se muestra la información al usuario sea agradable a la vista y esté organizado.

Por todo lo antes mencionado a continuación se explican algunas de ellas:

2.5.1 Lenguaje de marcas de hipertexto

El Lenguaje de Marcas de Hipertexto conocido como HTML (HyperText Markup Language) “es el lenguaje con el que se escriben las páginas web, haciendo uso de etiquetas para describir la forma en la que deberían aparecer los textos y los gráficos en un navegador web” (EcuRed, 2012).

2.5.2 Hojas de estilo en cascada

Hojas de Estilo en Cascada CSS (Cascading Style Sheets) es el lenguaje utilizado para describir la presentación de documentos HTML. CSS describe como debe ser renderizado el elemento estructurado en pantalla, en papel, hablado o en otros medios. (Mozilla Developer Network, s.f.)

2.5.3 Javascript

JavaScript (JS) es un lenguaje ligero e interpretado, orientado a objetos con funciones de primera clase, más conocido como el lenguaje de script para páginas web, pero también usado en muchos entornos sin navegador. Es un lenguaje script multi-paradigma, basado en prototipos, dinámico, soporta estilos de programación funcional, orientada a objetos e imperativa. (Mozilla Developer Network, s.f.)

Según España (2003) JavaScript permite crear pequeños programas llamados *Script*, que pueden acompañar un documento HTML o estar contenidos en su interior. Dicho programa se ejecuta en el navegador web cuando se carga el documento, o cuando se produce algún evento, como puede ser hacer clic en algún elemento por parte del usuario.

Estos *Scripts* permiten realizar ciertas acciones como lo son desplegar mensajes de alerta, realizar ciertos cálculos, agregar efectos a la hora de desplegar ciertos elementos en la pantalla y ocultarlos también y controlar otra serie de eventos. Realizar estas acciones en el lado de navegador es ventajoso, ya

que el usuario no obtiene tiempo de respuesta prolongados a la hora de validar datos introducidos en un formulario, por ejemplo, ya que al ser ejecutado en el navegador no se deben hacer peticiones al servidor y esperar su respuesta, sino que la validación se realiza de manera casi inmediata.

Hoy en día existe un gran número de herramientas, como bibliotecas, basadas en este lenguaje muy útil para el desarrollo de aplicaciones web, algunas de las cuales se mencionan a continuación.

2.5.4 JQuery

jQuery es una librería de JavaScript rápida, pequeña y con una gran cantidad de funcionalidades. Permite simplificar la manipulación y navegación de documentos HTML, manejo de eventos, y agregar interacción con la técnica AJAX¹ a través del uso de una API² que es compatible con los navegadores más utilizados. (jQuery, s.f.)

2.5.5 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 de CSS predefinidas y componentes que agilizan el desarrollo de interfaces de usuario amigables y agradables a la vista.

2.6 Tecnologías del lado del servidor

Las tecnologías del lado del servidor permiten construir la lógica de negocio de la aplicación web, con estas se pueden procesar las solicitudes que emite el usuario desde el lado del cliente a través del navegador web, para así emitir las respuestas necesarias para el funcionamiento completo de la aplicación, ellas también permiten comunicarse con el sistema encargado de hacer el manejo de los datos de la aplicación.

A continuación, se explican una serie de tecnologías que son de uso frecuente en el lado del servidor

¹ **JavaScript Asíncrono y XML (AJAX)** no es una tecnología por sí misma, es un término que describe un nuevo modo de utilizar conjuntamente varias tecnologías existentes. Permite que las aplicaciones web capaces de actualizarse continuamente sin tener que volver a cargar la página completa.

² La **interfaz de programación de aplicaciones**, abreviada como **API** (del inglés: *Application Programming Interface*), es el conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro *software* como una capa de abstracción.

para desarrollar aplicaciones web.

2.6.1 PHP

PHP (acrónimo de "*PHP: Hipertexto Preprocessor*") es un lenguaje de *scripting* de propósito general de código abierto, interpretado y multiplataforma que está pensado principalmente para el desarrollo de aplicaciones web de contenido dinámico y que está enfocado principalmente en ser un lenguaje del lado del servidor, aunque también puede ser embebido en páginas HTML.

Su sintaxis recurre a C, Java y Perl, siendo así sencillo de aprender. El objetivo principal de este lenguaje es permitir a los desarrolladores web escribir dinámica y rápidamente páginas web generadas (PHP Group, s.f.). Por estar desarrollado bajo políticas de código abierto ha tenido muchas contribuciones por parte de muchos desarrolladores a lo largo de su historia, pero fue creado originalmente por Rasmus Lerdorf en 1994.

Otra de las características principales de PHP es que cuenta con una amplia cantidad de extensiones que permiten trabajar con distintas bases de datos, enlaces a librerías, protocolos, etc.

2.6.1.1 Marcos de trabajo de PHP

PHP cuenta con un gran número de *frameworks* (marcos de trabajo), que facilitan el desarrollo de las aplicaciones web. Elegir el *framework* adecuado para un proyecto no es tarea fácil, pero para hacer una buena elección es indispensable conocer distintas opciones.

Analizando los frameworks existentes para PHP se decidió utilizar Laravel debido a sus funcionalidades y fácil integración con MySQL

- **Laravel:** Laravel es un *framework* para el desarrollo de aplicaciones web desarrollado en PHP 5. Posee una sintaxis expresiva, esto puede ser debido a que su filosofía es desarrollar código simple y elegante. Laravel trata de facilitar el proceso de desarrollo facilitando las tareas comunes usadas en la mayoría de los proyectos web, como autenticación, enrutamiento, sesiones y almacenamiento en caché. Incorpora también el patrón MVC, permite gestionar de manera sencilla casi cualquier base de datos y su documentación es bastante completa.

PHP	
Definición	Lenguaje de <i>scripting</i> , interpretado, multiplataforma y de propósito general, aunque está enfocado principalmente para generar aplicaciones web.
Objetivo	Permitir a los desarrolladores web escribir páginas generadas dinámicamente con rapidez.
Facilidad de Aprendizaje	La sintaxis de PHP es más parecida a JavaScript, Java o C, por lo que resulta más sencillo de aprender para las personas que conozcan alguno de esos lenguajes.
Marcos de Trabajo	Yii, CodeIgniter, Symfony, Laravel, entre otros.
Grandes proyectos en los que ha sido utilizado	Wikipedia, WordPress, Joomla!, Drupal, Magento, Prestashop, Symfony, Laravel.
Ventajas	<ul style="list-style-type: none"> - Es uno de los lenguajes más utilizados en el desarrollo web. - La curva de aprendizaje es muy corta. - Instalación y configuración sencilla. - Comunidad enorme y muchísima documentación. - Diversos <i>frameworks</i>. - Compatible con diversas bases de datos. - Multiplataforma. - Código abierto.
Desventajas	<ul style="list-style-type: none"> - La orientación a objetos de PHP está incluida a la fuerza, ya que en un principio no se concibió para esto. - Si no se utiliza un <i>framework</i> donde se sigan un conjunto conceptos y de buenas prácticas, el código puede quedar muy desordenado y sin separación de responsabilidades por encontrarse incrustado dentro del HTML.

2.6.2 Sistemas Manejadores de Bases de Datos

Según Silberschatz A. et al. (2002) un Sistema Manejador de Base de Datos (SMBD) “consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos” y su objetivo principal “es proporcionar una forma de almacenar y recuperar la información de una base de datos de manera que sea tanto práctica como eficiente”.

2.5.4.2 MySQL

MySQL es un Sistema Manejador de Base de Datos Relacional, de código abierto, multihilo y multiusuario. Fue creado por la compañía sueca MySQL AB en 1995, por los desarrolladores Michael Widenius, David Axmark y Allan Larsson, con el objetivo principal de ofrecer opciones eficientes y fiables de gestión de datos para usuarios domésticos y profesionales. (Chacaltana Gonzalo, 2014)

MySQL	
Definición	MySQL es un sistema manejador de base de datos relacional, de código abierto, multihilo y multiusuario que ofrece opciones eficientes y fiables de gestión de datos para usuarios domésticos y profesionales.
Límite	MySQL Server maneja bases de datos de terabytes, pero el código puede compilarse en una versión reducida adecuada para dispositivos embebidos y de mano.
Ventajas	<ul style="list-style-type: none">- Es código abierto.- Facilidad de configuración e instalación.- Puede ser ejecutado en una máquina con escasos recursos.- Bajo costo en requerimientos para la elaboración de bases de datos.- Soporta gran variedad de sistemas operativos.- Gran velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.- Su conectividad, velocidad y seguridad hacen de MySQL Server altamente apropiado para bases de datos asociadas a sitios webs.
Desventajas	<ul style="list-style-type: none">- Los vendedores de <i>software</i> propietario que utilizan MySQL deben liberar su código o adquirir una licencia propietaria de Oracle.- La velocidad de respuesta se puede ver un poco degradada si aumenta el tamaño de la base de datos.

CAPÍTULO 3 - MARCO METODOLÓGICO

Para asegurar el éxito del desarrollo de software y evitar que se convierta en una actividad caótica es necesario hacer uso de una metodología de desarrollo, éstas nos proveen de una dirección a seguir para la producción de un software completo y de calidad. Por esta razón, es de vital importancia seleccionar la metodología adecuada, la cual debe satisfacer todas las necesidades de desarrollo y su selección depende del tipo de software a implementar y de sus requerimientos.

En este capítulo se describen las metodologías de desarrollo dirigiendo el enfoque hacia las metodologías ágiles, ya que son las más adecuadas para proyectos donde es necesaria la flexibilidad y la adaptación a los cambios, como es el caso de las aplicaciones web, seleccionando la que mejor se adapte al caso particular de estudio.

3.1 Metodologías de Desarrollo

La metodología para el desarrollo de software es un modo sistemático de realizar, gestionar y administrar un proyecto, de tal manera que pueda llevarse a cabo con altas probabilidades de éxito, esta comprende los procesos a seguir para idear, implementar y mantener un producto de software, desde que surge la necesidad hasta que se cumple el objetivo.

En la actualidad se pueden clasificar estas metodologías en 2 grandes grupos las metodologías tradicionales y las metodologías ágiles:

Las metodologías tradicionales, o también llamadas pesadas, están orientadas al control de los procesos de desarrollo del software, con el objetivo de conseguir un *software* más eficiente y predecible. Se establecen rigurosamente las actividades a desarrollar, herramientas a utilizar y notaciones que se usarán; una vez que esta todo detallado comienza el ciclo de desarrollo.

Las metodologías pesadas han sido las más prevalentes por muchos años, sin embargo, siempre han sido criticadas por su reducida practicidad, su carácter excesivamente burocrático y su poca adaptación a los cambios. En consecuencia, han comenzado a ser sustituidas por el uso de metodologías ágiles o ligeras, que se caracterizan por ser iterativas e incrementales, menos rígidas y fácilmente adaptables a los cambios, involucrando al usuario en el desarrollo.

En la Tabla 3.1 podemos observar una comparación entre las metodologías tradicionales y las ágiles

Tabla 3.1 – Características de las Metodologías Ágiles

Metodologías Ágiles
Basadas en heurísticas provenientes de prácticas de producción de código
Especialmente preparados para cambios durante el proyecto
Impuestas internamente (por el equipo)
Proceso menos controlado, con pocos principios
El cliente es parte del equipo de desarrollo
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio
Pocos artefactos y roles
Menos énfasis en la arquitectura del software

3.1.1 Metodologías Ágiles

Las metodologías ágiles son un conjunto de métodos de Ingeniería del Software, que se basan en el desarrollo iterativo e incremental, teniendo presente cambios y respondiendo a los mismos, al final de cada iteración se evalúan las prioridades del proyecto con la finalidad de cumplir los objetivos, durante todo el proceso de desarrollo se cuenta con un grupo de desarrolladores auto-organizados y multidisciplinarios.

En la creación del Manifiesto de Desarrollo Ágil de *Software (2001)* se enumeraron los principales valores que siguen las metodologías ágiles:

- Individuos y su interacción, por encima de los procesos y las herramientas.
- *Software* funcional, por encima de la documentación exhaustiva.
- Colaboración con el cliente, por encima de la negociación contractual.
- Respuesta ante el cambio, por encima del seguimiento de un plan.

Estos valores inspiraron la creación de 12 principios que conforman las principales características de las metodologías ágiles:

1. La mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de *software* con valor.
2. Aceptar que los requerimientos cambien, incluso en etapas tardías del desarrollo. Los procesos ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
3. Entregar *software* funcional frecuentemente, entre dos (2) semanas y dos (2) meses, con preferencia al periodo de tiempo más corto posible.
4. Los responsables de negocio y los desarrolladores trabajan juntos de forma cotidiana durante todo el proyecto.
5. Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
6. El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
7. El *software* funcionando es la medida principal de progreso.
8. Los procesos ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.
9. La atención continua a la excelencia técnica y al buen diseño mejora la agilidad.
10. La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
11. Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
12. A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

En la actualidad, las metodologías ágiles tienen gran aceptación a nivel mundial, ya que son una buena alternativa al proceso de desarrollo de software tradicional, por su simplicidad y velocidad para crear sistema y su excepcional interacción con el cliente.

A continuación, en base al estudio realizado en el seminario, se presentan las metodologías ágiles que se consideraron más adecuadas para el desarrollo del Trabajo Especial de Grado (TEG).

3.1.1.1 Metodología AgilUS

La metodología AgilUs nace como línea de investigación del Centro Ingeniería de *Software* y Sistemas (ISYS) de la Escuela de Computación de la Universidad Central de Venezuela. Se basa en el concepto de usabilidad, en la necesidad de desarrollar *software* usable. Se fundamenta en el análisis centrado en el usuario y en la participación de especialistas, con el objetivo de evolucionar el *software*, a fin de que éste alcance el mayor grado de usabilidad una vez culminado su desarrollo (Acosta Alecia, 2011).

AgilUs es un método de desarrollo iterativo e incremental que pone el mayor peso del desarrollo en la obtención de la usabilidad. Se centra en que la construcción de las interfaces de usuario se realice desde el comienzo del desarrollo del sistema y no al final. Esta metodología reduce la cantidad de actividades y artefactos que se generan, propicia la participación del usuario y se adapta a los cambios que puedan surgir en el desarrollo. Está orientada al desarrollo de sistemas con alto grado de interactividad con los usuarios (*front-end*) y menos complejidad en la lógica de la aplicación (*back-end*).

Principios

Acosta (2011) explica que AgilUs centra el desarrollo de *software* en los siguientes principios:

- **Integra la Interacción Humano Computador (IHC) y la Ingeniería de *Software* (IS):** IS e IHC son complementarias, no son disciplinas excluyentes. Un diseño centrado en el usuario impacta positivamente en la calidad del *software* (ISO 9126-1).
- **La usabilidad debe considerarse desde el principio del desarrollo:** la usabilidad aumenta la calidad del *software*, entonces es conveniente incluirla desde el principio en el desarrollo como uno de los requerimientos fundamentales para impactar positivamente en la calidad del producto final.
- **La usabilidad determina la utilidad:** un *software* se considera útil en la medida que pueda ser usado a fin de producir resultados, en forma eficiente, intuitiva y satisfactoria para los usuarios.
- **El usuario determina la usabilidad:** la usabilidad no es una propiedad abstracta. Un *software* sólo será considerado usable en un contexto específico y por un tipo de usuario específico. El objetivo es lograr que todos los usuarios del *software* encuentren usables las tareas que pueden realizar.

Ciclo de Vida

En cada etapa del desarrollo se incluyen actividades para la construcción de la usabilidad, El Ciclo de Vida engloba la definición de requisitos, análisis, prototipo y entrega. En la figura 3.1 podemos observar el ciclo de vida de la metodología AgilUS.



Figura 3.1 – Etapas del Ciclo de Vida AgilUS

A continuación, se describe las cuatro (4) etapas de AgilUs (Acosta Alecia, 2011):

1. **Requisitos:** se realiza el análisis global del problema a solucionar, se estudian productos similares existentes, se genera un perfil de usuario, y se define la lista de requerimientos a desarrollar. Esta etapa es importante en el desarrollo del *software*, ya que un mal análisis de requerimientos traería como consecuencia un *software* que no cumple con las necesidades del usuario.
2. **Análisis:** se lleva a cabo el análisis de la solución a desarrollar, se emplean diagramas de casos de uso y modelo de objetos del dominio, siguiendo la notación UML, para definir las funcionalidades que tendrá el producto a desarrollar.

-
3. **Prototipo:** se implementa un prototipo rápido de la interfaz de usuario a partir de los patrones de interacción, el cual va evolucionando hasta convertirse en el producto final y se realizan evaluaciones de usabilidad apropiadas a esta etapa: evaluaciones heurísticas y listas de comprobación.
 4. **Entrega:** se aplican las pruebas al sistema para certificar que la aplicación desarrollada sea un *software* usable y sin errores, finalmente se pone en producción la aplicación.

3.1.1.2 Metodología Scrum

Es una metodología ágil y flexible de gestión y está especialmente indicada para proyectos donde se necesita obtener resultados pronto, los requerimientos son cambiantes o poco definidos, y la innovación, la competitividad, la flexibilidad y la productividad son fundamentales (Proyectos Ágiles, 2015).

Scrum tiene dos (2) características resaltantes: la primera es que el desarrollo de *software* se realiza mediante iteraciones, denominadas *Sprints*, cuya duración es de dos (2) o cuatro (4) semanas. El resultado de cada *Sprint* es un incremento ejecutable que se muestra al Propietario del producto. La segunda característica importante son las reuniones a lo largo del proyecto, entre ellas destaca la reunión diaria de quince (15) minutos del Equipo de desarrollo para la coordinación e integración.

Roles

- **Propietario del producto:** representa a todos los interesados en el producto final. Se focaliza en la parte de negocio y es responsable de la inversión del proyecto. Traslada la visión del proyecto al equipo, formaliza las prestaciones en historias a incorporar en la pila de producto y las prioriza de forma regular.
- **Gestor de Scrum (*Scrum Manager*):** responsable del proceso Scrum que certifica que todos los miembros del equipo hagan su trabajo de manera adecuada, guiándolos para que cumplan las reglas y los procesos de la metodología. Trabaja con el Propietario del producto para maximizar la satisfacción del usuario.
- **Equipo de desarrollo (*Team*):** grupo de profesionales que desarrollan el proyecto de manera conjunta llevando a cabo las tareas a las que se comprometen al inicio de cada *Sprint*.

Existen otros actores secundarios que pueden aportar algo en determinado momento del flujo de trabajo, como lo son los usuarios y otras partes interesadas.

Acciones y Artefactos

Al ser Scrum una metodología iterativa, las acciones se repiten periódicamente:

- 1. Pila de producto (*Product backlog*):** corresponde a una lista de requerimientos ordenados por prioridad, llamadas historias, realizada generalmente por el Propietario del producto. Esta lista nunca llega a ser una lista completa y definitiva, es un documento dinámico que incorpora constantemente las nuevas necesidades del sistema.
- 2. Sprint Planning Meeting (Reunión de planificación *Sprint*):** reunión cuyo objetivo es planificar la Pila de *Sprint* a partir de la Pila de producto, suelen participar los tres (3) actores principales de la metodología. El Equipo de desarrollo se compromete a realizar las tareas del *Sprint*.
- 3. Pila de *Sprint* (*Sprint backlog*):** lista de tareas necesarias para cumplir con las historias de la Pila de producto y son determinadas por el Equipo de desarrollo. Estas tareas se deben durante la duración del *Sprint*. Mientras una Pila de *Sprint* se inicia no debe ser alterada o modificada, hay que esperar a que concluya para hacerlo.
- 4. Daily Scrum Meeting (Reunión diaria de Scrum):** es una reunión que realiza el Equipo de desarrollo todos los días durante el *Sprint*, en la cual se busca identificar obstáculos, verificar el avance de las tareas y planificar las tareas del día.
- 5. Revisión de *Sprint*:** una vez finalizada una Pila de *Sprint*, se revisa rápidamente si se ha obtenido un producto que el Propietario del producto pueda ver y tocar, siendo el Gestor de Scrum quien muestra los avances.
- 6. Retrospectiva de *Sprint*:** el Propietario del producto revisa con el Equipo de desarrollo los objetivos marcados inicialmente en la Pila de *Sprint* concluido, se aplican los cambios y ajustes que sean necesarios, y se marcan los aspectos positivos (para repetirlos) y los aspectos negativos (para evitar que se repitan) de la Pila de *Sprint*.

Ciclo de Vida

El Ciclo de vida de Scrum es iterativo e incremental y se puede detallar en la siguiente figura 3.2:

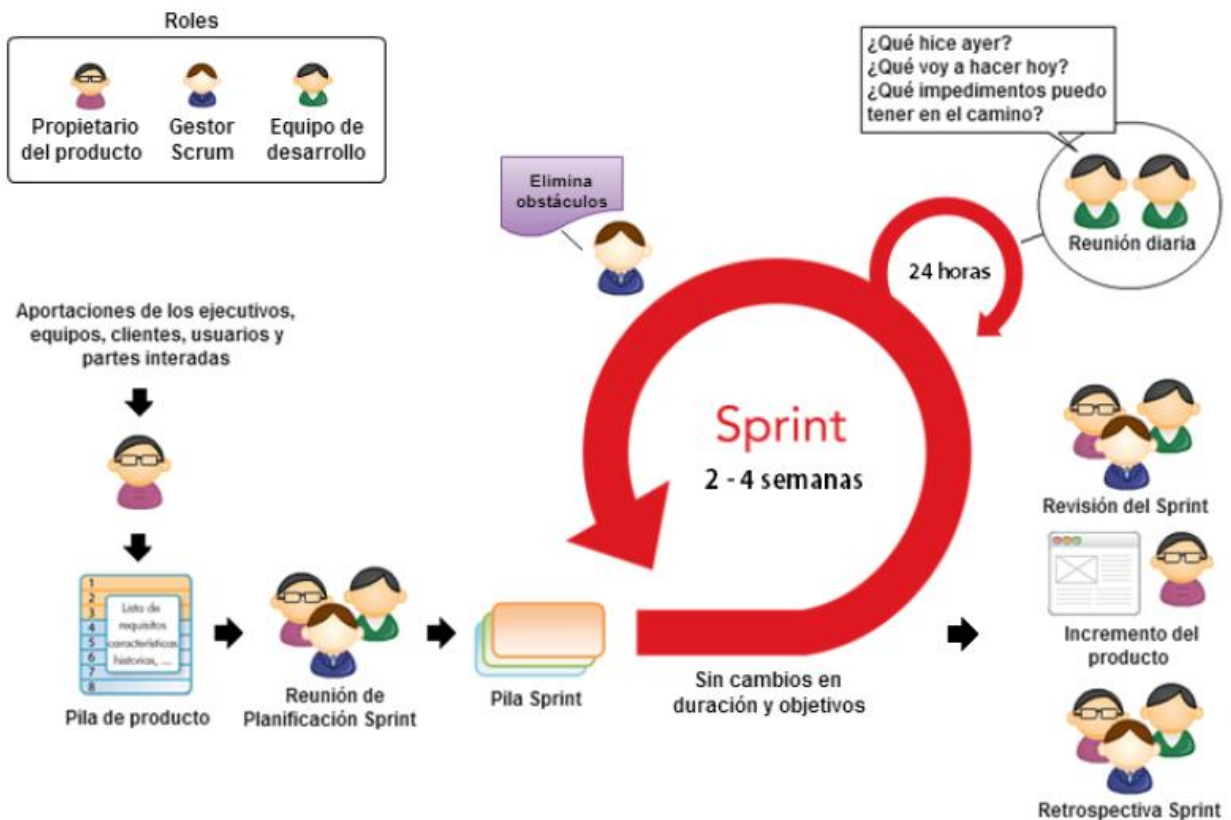


Figura 3.2 – Ciclo de Vida Scrum

Proceso de Desarrollo

El proceso de desarrollo de un proyecto Scrum se resume en los siguientes puntos:

1. El Propietario del producto crea una visión de trabajo del producto final, definiendo un conjunto de requerimientos ordenados por prioridad en la Pila de producto.
2. Se realiza la reunión de Planificación del *Sprint*, donde el Equipo de desarrollo selecciona los requerimientos prioritarios de la Pila de producto y los coloca en la Pila de *Sprint*. Seguidamente, elaboran una lista de las tareas necesarias para desarrollar los requerimientos seleccionados y se da inicio al *Sprint*.

-
3. Cada día el Equipo de desarrollo tiene una reunión de 15 minutos, donde cada integrante responde las siguientes preguntas:
 - o ¿Qué hice ayer?
 - o ¿Qué voy a hacer hoy?
 - o ¿Qué impedimentos puedo tener en el camino?
 4. Durante el *Sprint*, el Gestor Scrum se encarga de eliminar cualquier obstáculo que pueda surgir, logrando que el Equipo de desarrollo pueda cumplir con su compromiso.
 5. El último día del *Sprint* se realiza la reunión de Revisión del *Sprint*, donde se verifica si se ha obtenido un producto que pueda ver y tocar el Propietario del producto, y si es así el Gestor Scrum será el encargado de mostrar los avances.
 6. También se realiza la reunión de Retrospectiva del *Sprint*, donde se revisan los objetivos marcados inicialmente en la Pila de *Sprint* que acaba de concluir y se aplican los cambios y ajustes necesarios. También se analiza el proceso de trabajo en el *Sprint*, los aspectos positivos, para repetirlos y los aspectos negativos, para evitar que se repitan, siendo el Gestor Scrum el encargado de eliminar los obstáculos identificados.
 7. El Equipo de desarrollo repite el ciclo hasta que el Propietario del producto decida liberar el producto.

Ya que el objetivo era desarrollar una aplicación usable en un corto periodo de tiempo, el equipo decide realizar el desarrollo final de la aplicación utilizando la metodología Scrum, la cual permite que el equipo se adapte a los plazos de tiempo dados, adicionalmente fueron empleados algunos de los artefactos que ofrece la Metodología AgilUS tales como: encuestas los usuarios y prototipos no funcionales que en conjunto con las reuniones planificadas al final de cada sprint con los propietarios de producto Vanessa López y Durbis Peña permitieron desarrollar una aplicación que puede ser fácilmente utilizada por cualquier profesional del área.

CAPITULO 4 - MARCO APLICATIVO

En este capítulo se describen aspectos y características relevantes de la aplicación y se detallan los pasos que se realizaron para el desarrollo de la aplicación web, siguiendo los lineamientos de la metodología scrum con apoyo de algunos de los artefactos utilizados en la metodología AgilUs.

4.1 Análisis Preliminar

Para realizar el desarrollo de la aplicación web se tomaron en cuenta varios aspectos importantes. Primero, como se mencionó anteriormente, se necesita que esta aplicación sea accedida desde cualquier dispositivo. Segundo, las fórmulas utilizadas deben seguir los estándares del Instituto Nacional de Nutrición y la OMS. Tercero, utilizar tecnologías software libre. Y, por último, ser una aplicación sencilla de utilizar para un nutricionista. En las figuras 4.1 y 4.2 podemos observar algunos de los requerimientos definidos durante el análisis.

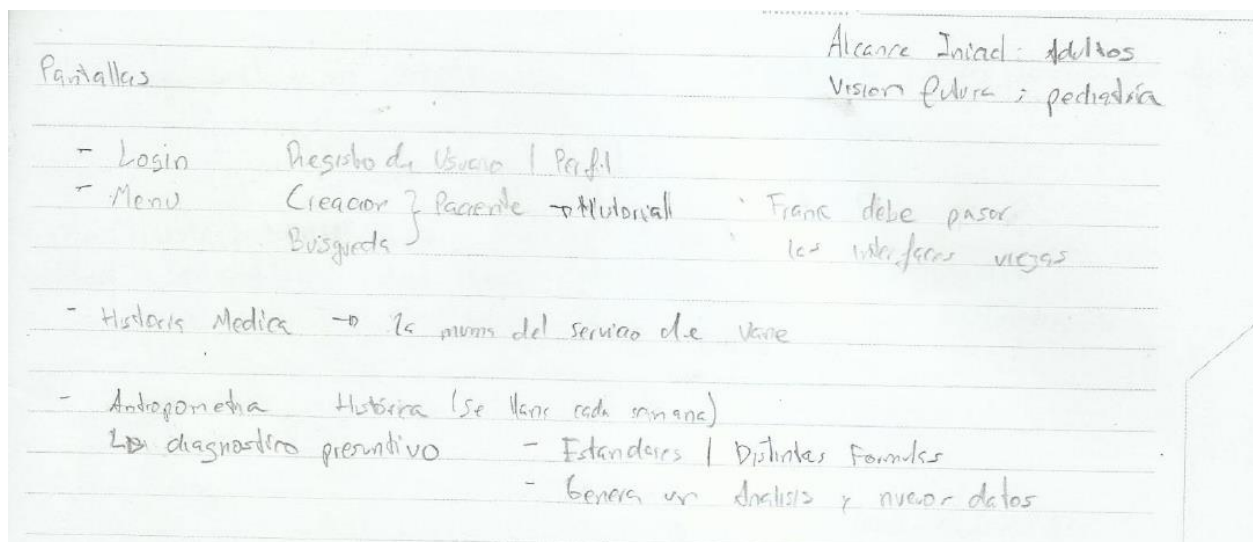


Figura 4.1– Requerimientos Iniciales (Login, Historias, Antropometría)

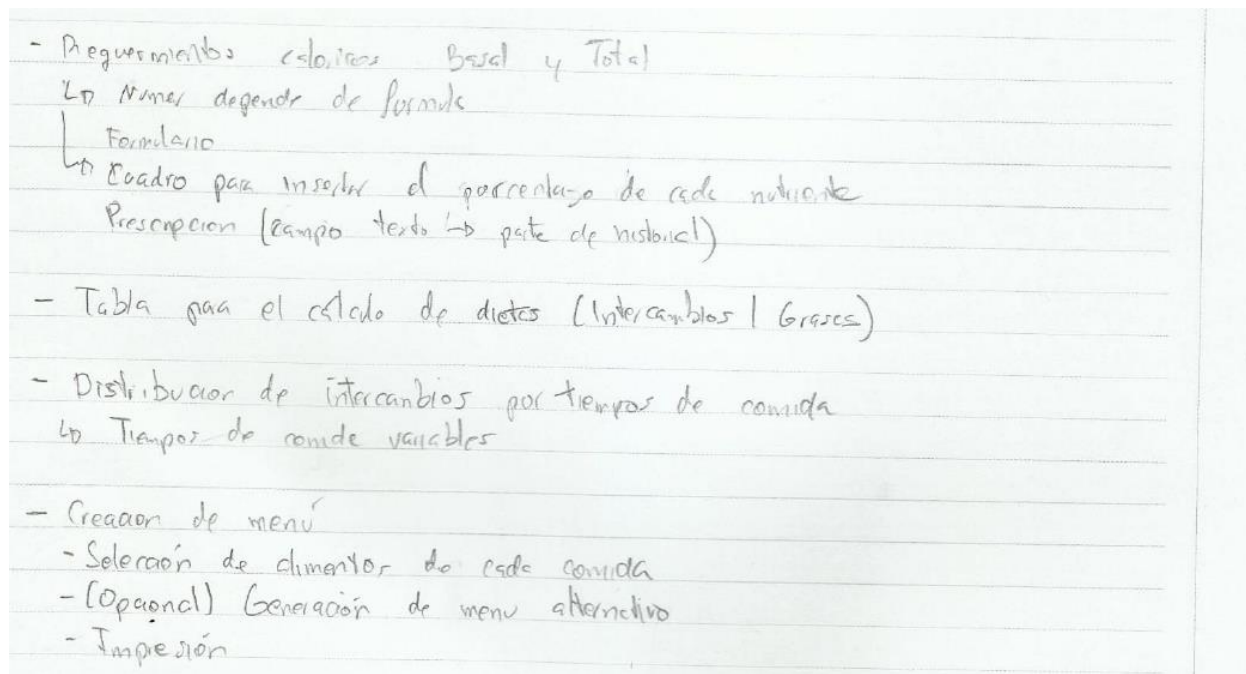


Figura 4.2– Requerimientos Iniciales (Calóricos, Distribución de Intercambios y Menús)

4.2 Diagrama de Casos de Uso

El modelo de casos de uso permite especificar las funcionalidades y el comportamiento del sistema mediante su interacción con usuarios u otros sistemas. Además, con este modelo se llega a un lenguaje estándar que es entendido tanto por los expertos como por los usuarios y que permite el análisis del dominio de la aplicación.

Para el esquema de la aplicación en general se tienen dos usuarios:

- **Nutricionista:** aquellos profesionales del área de nutrición que realizan sus estudios a los distintos pacientes. Puede registrar, modificar, eliminar y consultar la información que le pertenece, tal como: su perfil de usuario, menús plantilla, pacientes y cualquier información asociada los mismos.
- **Administrador:** puede realizar cualquier acción dentro de la aplicación que realice un nutricionista, adicionalmente puede registrar, modificar y eliminar usuarios (nutricionistas) y por último puede reasignar pacientes a otros nutricionistas en caso de ser necesario.

En las Figura 4.3, 4.4, 4.5 y 4.6 se pueden observar los diagramas de casos de uso Nivel 1 y Nivel 2 basado en los requerimientos del sistema desarrollado.

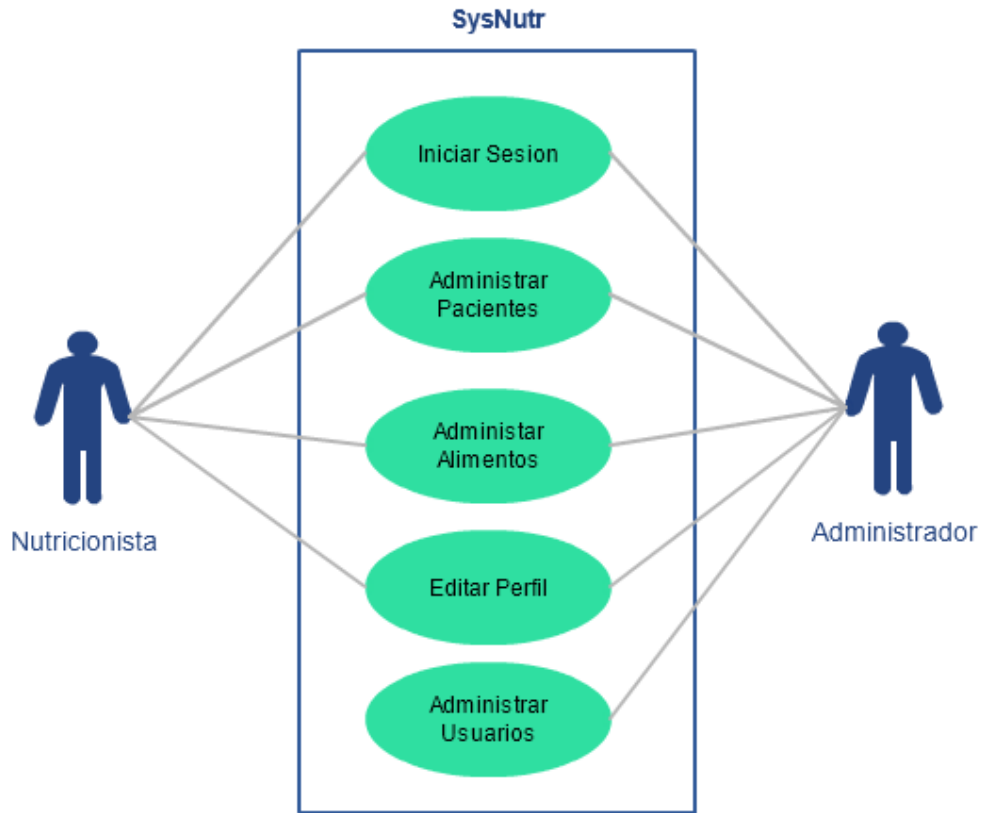


Figura 4.3 – Modelo de Casos de Uso – Nivel 1

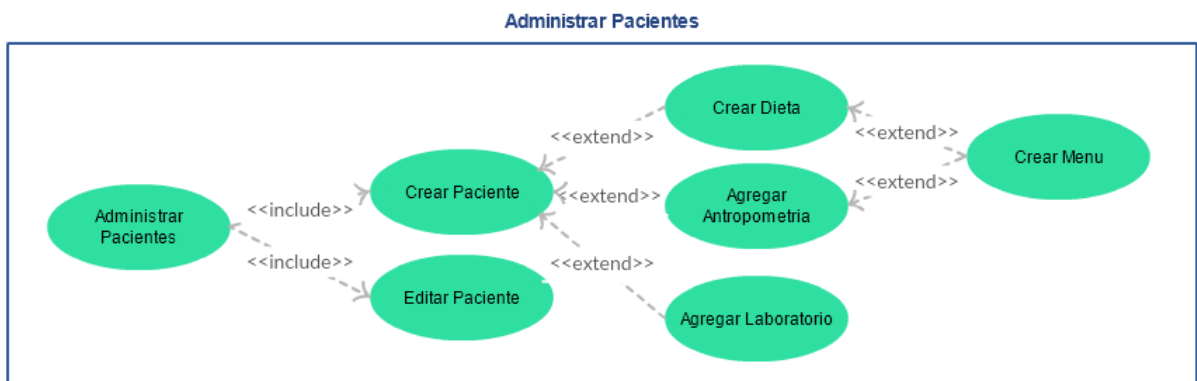


Figura 4.4– Modelos de Casos de Uso – Nivel 2 – Administrar Paciente

Administrar Alimentos

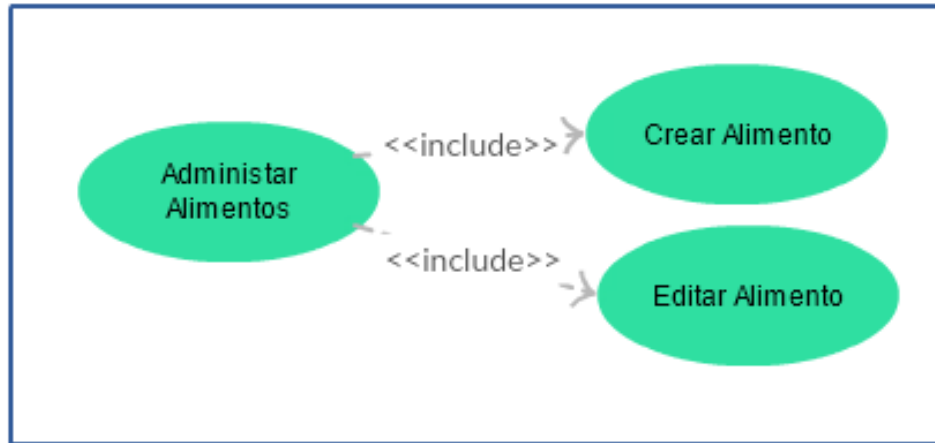


Figura 4.5– Modelos de Casos de Uso – Nivel 2 – Administrar Alimento

Administrar Usuarios

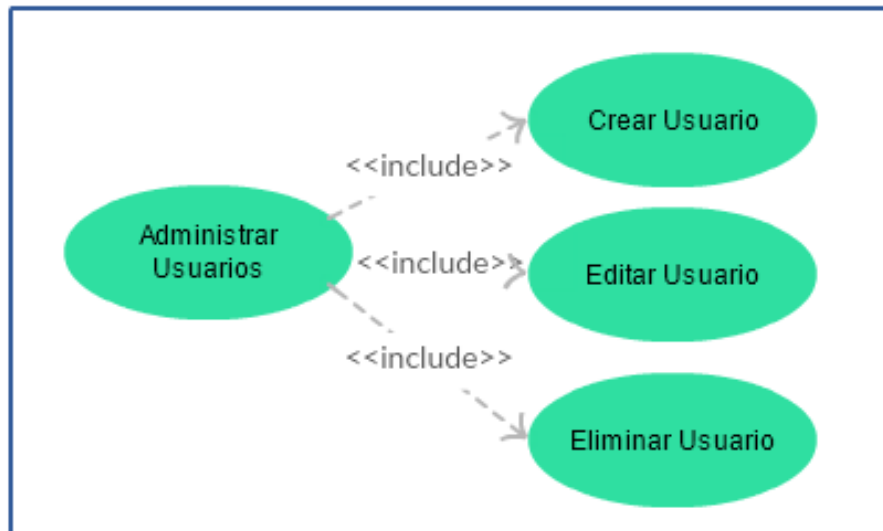


Figura 4.6– Modelos de Casos de Uso – Nivel 2 – Administrar Usuarios

4.3 Diseño de la Solución

La arquitectura que se propone es cliente-servidor. El cliente, que en este caso es cualquier navegador web, inicia la comunicación con un servidor web, a través de la red, realizando una petición al servidor de la página web y este servidor web se encarga de emitir la respuesta devolviendo la página web solicitada por el cliente, lo cual se puede visualizar en la Figura 4.7.

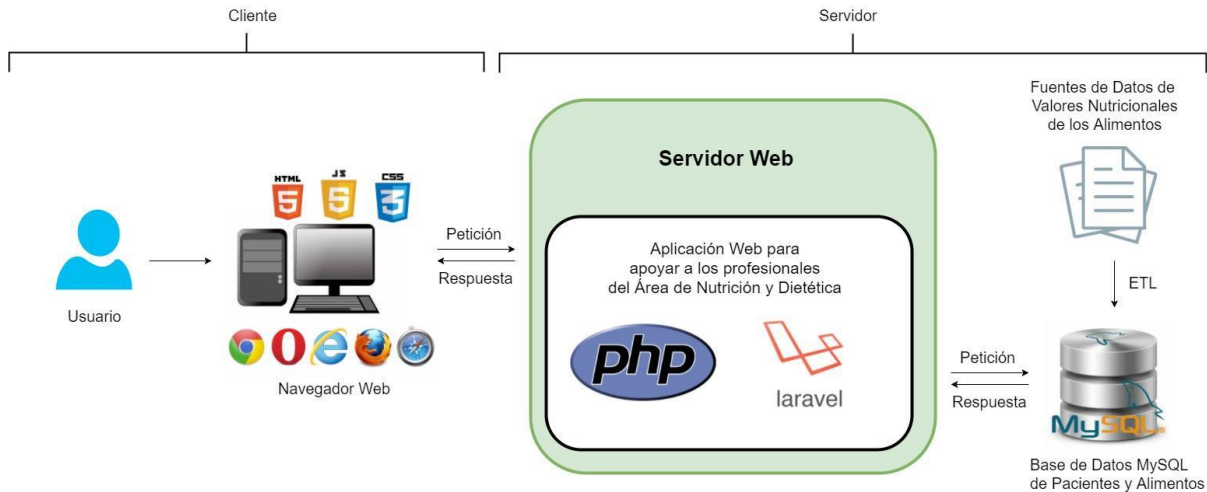


Figura 4.7 – Arquitectura de la Solución Propuesta

4.4 Desarrollo de la solución

Una vez definido el flujo y el diseño de la solución se comienza con el desarrollo de la aplicación, el cual se explica de forma detallada a continuación.

4.4.1 Base de Datos

La base de datos donde se almacena toda la información para el funcionamiento de la aplicación web se diseña usando el Sistema Manejador de Bases de Datos MySQL, en la figura 4.8 se puede observar el modelo relacional de la misma.

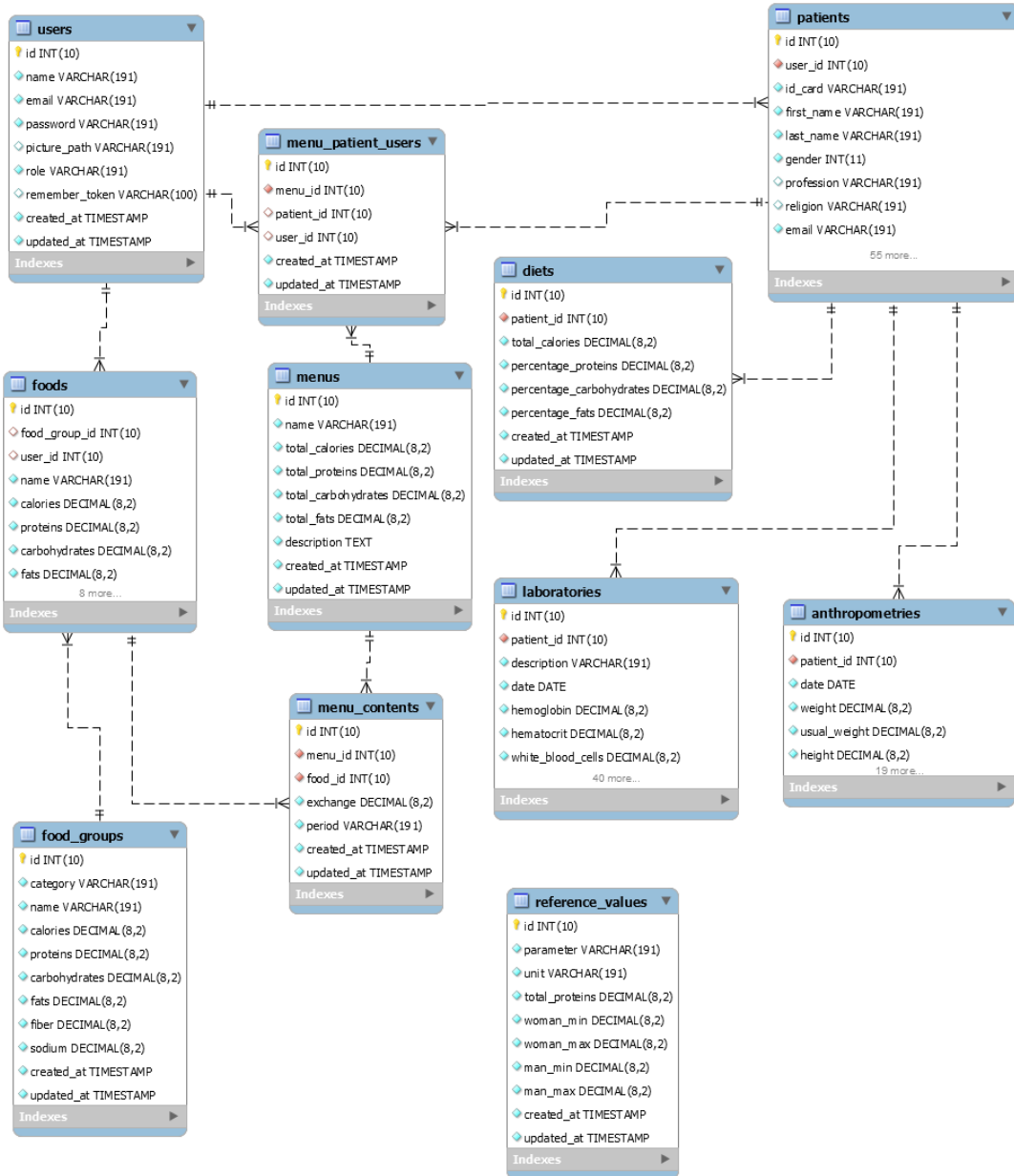


Figura 4.8 – Modelo relacional de la Base de Datos

En la base de datos se carga los datos sobre los alimentos que pueden usar los nutricionistas para la realización de los planes dieto-terapéuticos. Esta información es proporcionada por profesionales del área y pasa por un proceso de extracción y transformación antes de ser cargada a la base de datos. Esta información se obtuvo de tablas contenidas dentro de un documento Word, las cuales indican por grupo los valores nutricionales para cada uno de los alimentos consumidos por la población venezolana.

Esta información se extrajo de estas tablas, se agrupo y normalizo por grupo de alimento y por último se colocó en un formato especial para que pudiera ser insertado directamente a la base de datos.

El modelo de base de datos consiste de 11 tablas, explicadas a continuación:

- **Users:** contiene a todos los usuarios existentes en el sistema, tanto administradores como nutricionistas registrados y sus datos personales.
- **Patients:** contiene todos los datos de un paciente, entre ellos, datos personales, histórica médica, historia dietética, tolerancias de alimentos, hábitos y evaluación nutricional. Un paciente puede estar relacionado con un único nutricionista.
- **Anthropometries:** contiene las medidas antropométricas de un paciente por cada consulta, estas 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 tantos los alimentos más utilizados en la sociedad venezolana como cualquier alimento que el nutricionista desee agregar a su repertorio. Se almacenan 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:** contiene los valores de la fórmula dietética y número de intercambios que se deben ingerir
- **Menus:** contiene la distribución de alimentos a lo largo del día
- **Menu_Patient_Users:** esta tabla indica que menús han sido asignados a los pacientes y los nutricionistas dueños de estos menús
- **Menu_Contents:** contiene que alimentos 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:** contiene los rangos de valores aceptables por sexo para cada uno de los valores de un examen de laboratorio

4.4.2 Tecnologías involucradas en el desarrollo de la Aplicación

Toda aplicación web necesita contar con un servidor HTTP, un servidor de aplicaciones y un servidor de base de datos. Debido a que este trabajo consta del desarrollo de una aplicación web se tuvieron que elegir estos componentes. En el caso del servidor HTTP se decide utilizar Apache Web Server, para el servidor de aplicaciones, en cambio, se decide utilizar PHP Hipertexto Processor y por último se selecciona MySQL Database Server como servidor de base de datos. Para lograr instalar de manera sencilla estos componentes en el servidor de desarrollo se utiliza un paquete llamado MAMP.

El entorno de trabajo que se decide utilizar para desarrollar la aplicación es Laravel en su versión 5.4. Para hacer uso de este entorno 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 HTTP. Por otro lado, para la realización de las interfaces de usuario se utilizan los lenguajes HTML y CSS juntos con la biblioteca JQuery, la cual es incluida automáticamente por Laravel al crear el proyecto, y el framework Materialize que permite desarrollar de manera rápida las interfaces y que éstas sean agradables a la vista del usuario y fáciles de usar.

Para llevar control de la evolución y los cambios del código y facilitar el desarrollo en equipo se escoge el software de control de versiones Git, mediante el cual se puede realizar el desarrollo en paralelo de las vistas y controladores de la aplicación en el desarrollo de cada módulo y así avanzar de manera más rápida llevando control de los posibles errores que se pudieran generar al fusionar el código y a su vez guardar el historial de cambios en las distintas etapas del desarrollo.

Estas tecnologías fueron seleccionadas para el desarrollo de la aplicación ya que se rigen por el principio de código abierto lo que permite adquirirlas de manera gratuita, así como también instalarlas y configurarlas de manera fácil y al ser mantenidas constantemente por la comunidad de desarrolladores suelen añadirse nuevas características y mejoras de manera con esfuerzos mínimos. Por otra parte, además de adaptarse bastante bien a los requerimientos de la aplicación permiten ahorrar código a la hora del desarrollo y al ser utilizados por una gran cantidad de desarrolladores en el mundo se puede encontrar bastante documentación y ayuda en los diferentes foros de desarrolladores lo que permite tanto tener un código de calidad como ahorrar tiempo de desarrollo.

4.4.2.1 Estructura de los Directorios

Al crear un proyecto de Laravel se crean los directorios que podemos ver en la siguiente figura 4.9:

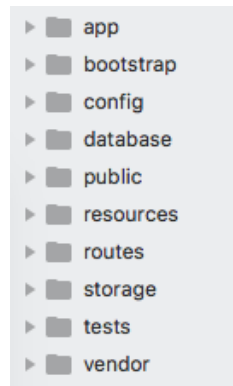


Figura 4.9 - Directorios

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

- Directorio “app”: En este directorio se encuentran 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. En este directorio también se encuentra el subdirectorio “Http/Controllers” en donde se definen y desarrollan los controladores que se encargarán de manejar todas las peticiones que se realicen al servidor. En este directorio también se encuentran otros archivos y subdirectorios creados por Laravel para realizar el proceso de autenticación (Figura 4.10).

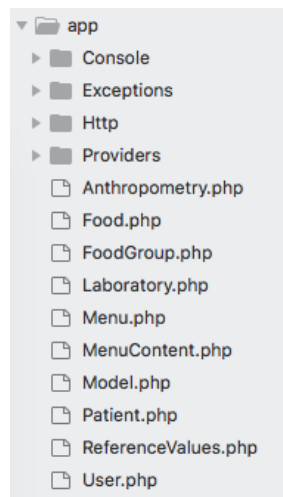


Figura 4.10 – App folder

-
- Directorio “database”: Dentro de este directorio se encuentra 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 de estos y sus restricciones. Durante este desarrollo también se decidió utilizar el subdirectorio “seeds”, para la realización de procesos de extracción, transformación y carga de datos ya que en este se colocan archivos para realizar inserciones automáticas en la base de datos (Ver Figura 4.11).

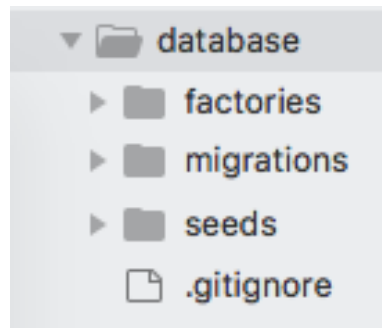


Figura 4.11 – Database folder

- Directorio “public”: En este directorio podemos encontrar subdirectorios llamados “css”, “fonts” y “js” donde se colocan los archivos que contienen las clases css, las fuentes y las funcionalidades programadas en lenguaje Javascript respectivamente las cuales se utilizan en las interfaces de usuario. También podemos encontrar en este directorio otros archivos que definen permisos de acceso a la aplicación, la página cargada en principio, el icono que se muestra en la pestaña del navegador al abrir la aplicación, entre otras cosas (Figura 4.12).

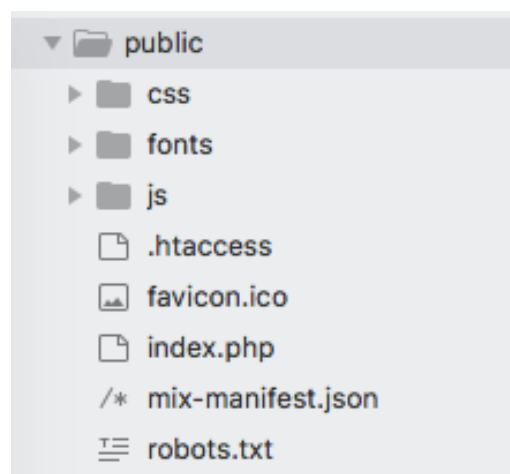


Figura 4.12 – Public folder

-
- Directorio “resources”: En este directorio se organizan 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 (Figura 4.13).

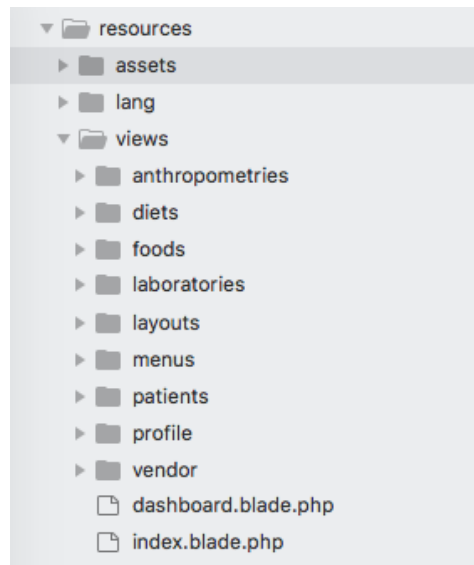


Figura 4.13 - Resources folder

- Directorio “routes”: Por último, tenemos el directorio “routes” en este solo se hace uso de un archivo llamado “web.php” en el que se define que acción ejecutar en respuesta de cada llamada

4.5 Flujo de la aplicación web e Interfaces de Usuario

El principal interés al realizar esta aplicación fue cumplir con cada uno de los requisitos descritos por el cliente; uno de los cuales consiste en realizar una aplicación de fácil acceso y cuyo flujo de trabajo similar al que generalmente estos desarrollan en la cotidianidad de la jornada laboral, en el consultorio o en la clínica donde estos laboran.

A continuación, se describe el flujo de la aplicación y las interfaces asociadas a casa paso.

4.5.1 Pantalla de acceso



Figura 4.14: Pantalla de acceso

En la figura 4.14 se muestra la pantalla diseñada para tener acceso a la aplicación, donde mediante la introducción del correo electrónico y la contraseña asignada por el administrador, el profesional en nutrición puede acceder a la aplicación y a toda la funcionalidad que ofrece de la misma.

4.5.2 Interfaz “Dashboard”

La figura 4.15 muestra la pantalla diseñada para que el profesional de nutrición pueda elegir la acción que desea realizar y pueda visualizar además los últimos movimientos realizados con anterioridad, referidos a las últimas actividades realizadas por el mismo, en relación a sus pacientes, menús y alimentos.



Figura 4.15: Dashboard

Donde en la parte superior izquierda, se encuentra la identificación del usuario reflejando así: el nombre, correo e imagen del mismo. En la parte inferior izquierda se encuentra el menú de navegación, donde el profesional en nutrición puede elegir la operación que desea desempeñar en un momento dado. Finalmente, en la parte central se encuentran los últimos movimientos o actividades realizadas por el profesional en relación a: sus pacientes (tanto nuevos como consultados), menús y alimentos agregados.

4.5.3 Interfaz “Mis pacientes”

Para continuar con el flujo de trabajo de los nutricionistas una vez iniciada la sesión, la primera actividad que realiza el profesional es la de atender a cada uno de los pacientes citados en la consulta. En la imagen 4.16 muestra la ventana titulada “Mis Pacientes”, esta ventana aparece posterior a la selección de la opción “mis pacientes” en el menú de navegación (ubicado a la Izquierda).

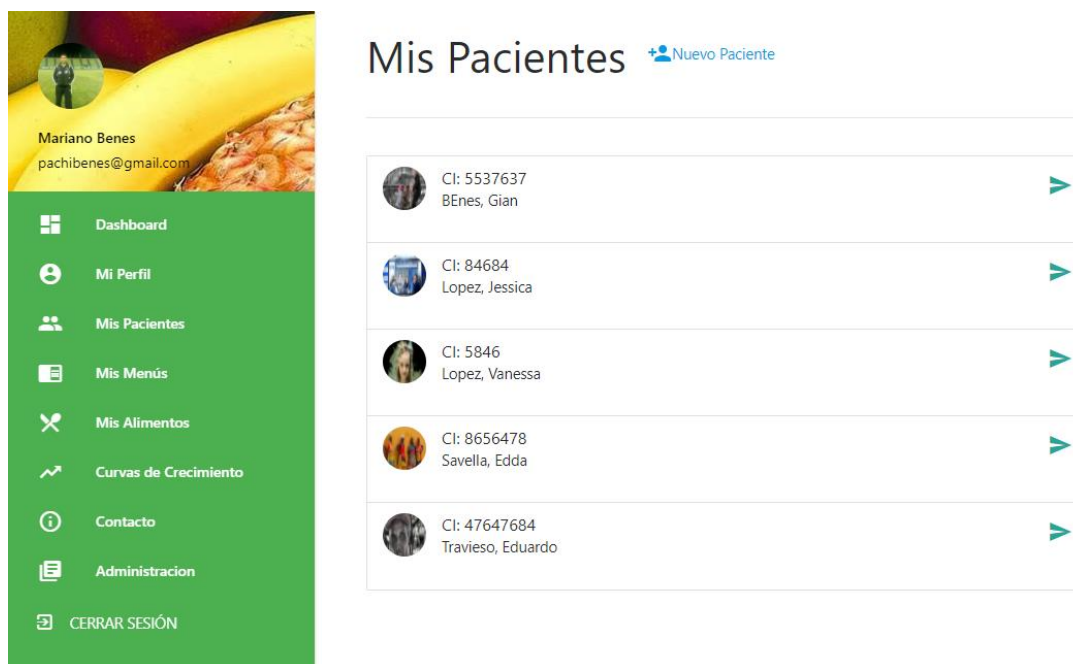


Figura 4.16: Mis Pacientes

En la parte central superior al lado de “Mis pacientes”, se encuentra la opción de “Nuevo paciente” donde el nutricionista puede agregar a cada uno de los pacientes que asisten por primera vez a la consulta.

La parte superior de esta ventana muestra una sección titulada “Últimos” donde el profesional en nutrición puede visualizar los últimos cinco (5) pacientes atendidos, independientemente del día, mes o año de evaluación. Estos se muestran en esta sección para que el nutricionista pueda tener un acceso más rápido a la historia del paciente que esté siendo atendido en ese momento, si es un paciente frecuente.

La parte inferior de la ventana muestra una sección titulada “Todos”, la cual fue diseñada para buscar y localizar al resto de pacientes previamente registrados (que no se encuentren entre los últimos 5 atendidos).

4.5.4 Interfaz “Registro del Paciente”

Al hacer clic en opción “Nuevo paciente” en la sección “Mis pacientes” expuesto en la imagen 4.16, lo redirecciona a la siguiente a la ventana de “registro de paciente”, mostrada en la figura 4.17.

Registro de Paciente

Mariano Benes
pachibenes@gmail.com

- Dashboard
- Mi Perfil
- Mis Pacientes
- Mis Menús
- Mis Alimentos
- Curvas de Crecimiento
- Contacto
- Administración
- CERRAR SESIÓN

Cédula: V-12345689

Nombres: _____ Apellidos: _____

Género: _____ Religión: _____ Profesión: _____

Email: _____ Condicion Socioeconómica: _____

Fecha de Nacimiento: _____ Fecha de Admisión: _____ Fecha de Evaluación: _____

Resumen del Caso: _____ Diagnóstico Medico: _____

Antecedentes Personales: _____ Antecedentes Familiares: _____

Tratamiento Nutricional y Médico: _____ Evaluacion Funcional: _____

Figura 4.16: Registro de Pacientes.

En la imagen 4.16, se muestra el formulario general donde se registra la historia médico nutricional el paciente a evaluar, considerando como datos principales: datos personales, hábitos psicobiológicos, antecedentes personales y familiares y los tratamientos nutricionales y médicos que esté cumpliendo el paciente al momento de ser evaluado. Al completar un ítem e ir al siguiente, se puede usar el mouse o la tecla “tabular” del teclado.

Una vez completado todo el formulario del paciente, se debe hacer clic en la parte inferior “Agregar paciente”, para finalizar el proceso de registrar al paciente a la base de datos de la aplicación. La información contenida en esta ventana puede ser editada si el profesional así lo decide.

4.5.5 Interfaz “Paciente”

Después de registrar al paciente se procede a realizar todo lo correspondiente a la evaluación nutricional, todo lo conveniente a esta evaluación se realiza desde la ventana que se muestra en la figura 4.18.

Figura 4.18: Pacientes

Donde la opción “Historia médica”, se refiere a los datos personales e historia registrada al agregar al paciente a la base de datos, por ende, la evaluación nutricional del paciente continuará a partir de las pestañas “Historia Dietética”, “Antropometría”, “Laboratorio” y “Dietas”, siguiendo ese orden. La opción “eliminar” que se encuentra en la parte superior derecha en esta ventana, se crea con la finalidad de eliminar todos los datos y registro de un paciente. En la parte inferior se encuentra la opción “Actualizar datos”, la cual está diseñada para guardar cambios que se deseen realizar en los datos del paciente.

4.5.6 Interfaz “Historia Dietética”

Una vez culminada la historia médica, y siguiendo la secuencia de trabajo del nutricionista se hará clic en la pestaña historia dietética, reflejada en la figura 4.18 (se encuentra en letras rojas).

Figura 4.19: Historia Dietética

Donde toda la información se ingresa en un formulario y contempla cada de uno de los ítems relacionados a la historia dietética del paciente. Al finalizar esta acción se debe hacer clic en parte inferior del formulario en la opción “Actualizar datos”, para que estos se guarden en la base de datos de la aplicación. La información contenida en esta ventana puede ser editada si el profesional así lo decide.

4.5.7 Interfaz “Antropometría”

Después de culminado el interrogatorio y registro de la información del paciente en la “Historia dietética”, se procede a tomar y registrar las medidas antropométricas del mismo.

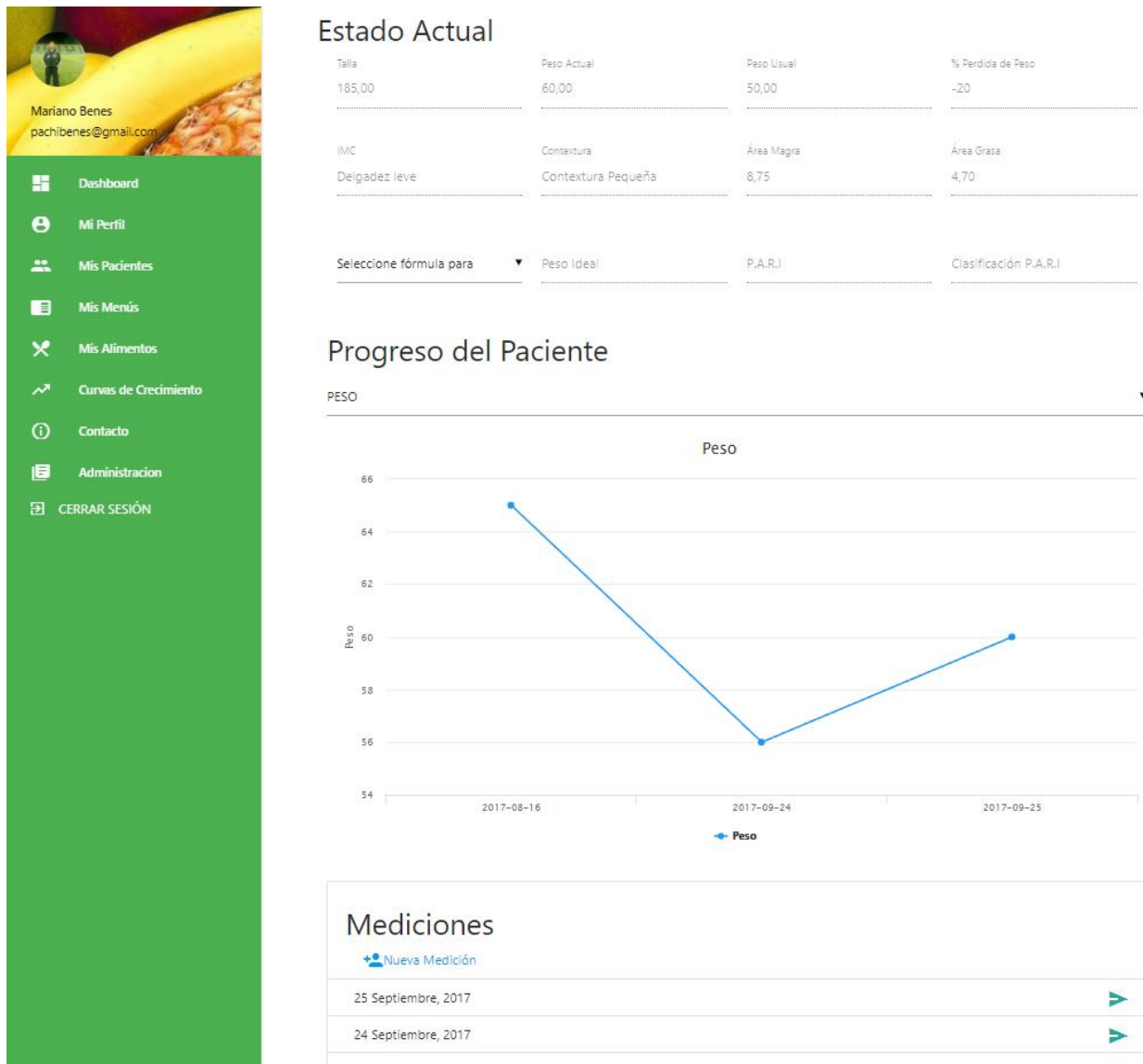


Figura 4.20: Antropometría

En la figura 4.20 se muestra la ventana correspondiente a la antropometría, la cual, en manera de formulario, muestra aquellos datos antropométricos de interés para la evaluación del paciente; para agregar estos se debe hacer clic en la opción “nueva medición” y al hacer clic, esto redirige al formulario anteriormente señalado en la figura 4.17, después de agregar todos los datos es necesario hacer clic en la opción “Agregar antropometría” para que esto quede registrado en la base de datos.

Algunos valores antropométricos, como el peso o los pliegues pueden variar de una consulta a otra; por esto, en cada consulta puede agregarse una nueva medición y estas se guardan según la fecha.

Esta ventana muestra la sección “Estado Actual” donde se presentan los últimos datos antropométricos registrados del paciente, además se encuentra la sección “Progreso del paciente”, donde se generan gráficos que comparan la evolución del paciente en cada consulta, en relación a su peso, su Índice de Masa Corporal y sus Áreas.

En la sección “Mediciones”, que se encuentra en la parte inferior de esta ventana se muestran cada una de la medición antropométrica realizada al paciente, en cada una de las citas a las que ha asistido.

4.5.8 Interfaz “Nuevo Laboratorio”

Posterior a la introducción de la antropometría se procede a registrar los exámenes bioquímicos o laboratorios del paciente, de acuerdo con la modalidad de trabajo del nutricionista.

Donde una vez completado y guardado el cuestionario correspondiente a la antropometría se hace clic en la pestaña “laboratorio” y esto redirige a la ventana mostrada en la figura 4.21



Paciente ELIMINAR

V-5537637 - Gian Benes

HISTORIA MÉDICA HISTORIA DIETÉTICA ANTROPOMETRÍA LABORATORIOS DIETAS

+ Nuevo Laboratorio

20 Septiembre, 2017 - Finalmente normal	➤
18 Septiembre, 2017 - Otros	➤
17 Septiembre, 2017 - Valores normalizados	➤
16 Septiembre, 2017 - Sangre - Aumento de colesterol	➤
15 Septiembre, 2017 - Orina	➤

Figura 4.21: Laboratorios

Como cada cierto tiempo los laboratorios se deben repetir, es necesario poder guardar diferentes resultados de laboratorios realizados en distintas fechas, para esto se crea una opción llamada “Nuevo Laboratorio”, el cual se utiliza para registrar cada uno de los laboratorios que posea el paciente.

Al presionar esta opción se muestran la ventana de la figura 4.22 donde se pueden el formulario con todos los valores de cada uno de los parámetros bioquímicos que se pueden registrar, dependiendo de la especificidad de los exámenes de laboratorio que haga entrega (en físico) el paciente durante la consulta nutricional.

Una vez agregados todos los datos, se debe hacer clic en la opción “Agregar laboratorio” y estos se guardan en la base de datos.

Nuevo Laboratorio
V-5537637 - Gian Benes

Fecha: 15 Septiembre, 2017

Descripción			
Hemoglobina - g/dL	Hematocrito - g/dL	Glóbulos blancos - cel/mm ³	Linfocitos - %
10,00	10,00	10,00	1,00
Neutrófilos - %	Neutrófilos no segmentados - %	Eosinófilos - %	Monocitos - %
1,00	10,00	10,00	10,00
Basófilos - %	Paquetas - cel/mm ³	CTL - mm ³	BUN - mg/dL
10,00	10,00	1,00	1,00
Urea - g/gL	Creatinina mg/dL	Ácido úrico - mg/dL	Sodio - mmol/L
0,00	10,00	10,00	10,00
Potasio - mmol/L	Cloro - mmol/L	Calcio Sérico - mg/dL	Calcio en Orina - mg en 24h
10,00	1,00	1,00	1,00
Fósforo - mg/dL	Magnesio - mg/dL	Proteínas totales - g/dL	Albumina - g/dL
1,00	0,00	10,00	1,00
Globulina - g/dL	Amilasa - U/L	Lipasa - U/L	AST - U/L
1,00	1,00	0,00	10,00
ALT - U/L	LDH - U/L	ALP o FA - U/L	Ácido Láctico - mmol/L
1,00	1,00	0,00	10,00
Bilirubina Total - mg/dL	Bilirubina Directa - mg/dL	Colesterol Total - mg/dL	Colesterol HDL - mg/
1,00	1,00	0,00	1,00
Colesterol - mg/dL	Triglicéridos - mg/dL	Glucosa - mg/dL	Péptido C - mg/dL
1,00	0,00	1,00	1,00

Figura 4.22: Nuevo Laboratorio

4.5.9 Interfaz “Dietas”

Posterior a la evaluación y agregado de los datos de laboratorio, ya se puede empezar a realizar la dieta correspondiente a las necesidades del paciente.

Como se muestra en la figura 4.23, en la ventana de dietas se muestra en la parte superior el requerimiento calórico total del paciente (que no es más que, lo que el paciente debe consumir durante todo el día). En esta interfaz, el usuario debe colocar el peso con el cual desea trabajar y las kcal por kg de peso que se adapten al mismo, después de colocar estos datos el resultado se arroja de manera automática. Bajo este valor se encuentran los datos del paciente necesarios para determinar el requerimiento calórico total.



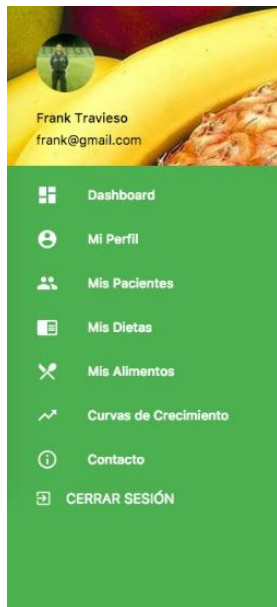
The screenshot shows a web application interface for diet management. On the left is a green sidebar menu with the following items: Dashboard, Mi Perfil, Mis Pacientes, Mis Menús, Mis Alimentos, Curvas de Crecimiento, Contacto, and Administración. The main content area has a top navigation bar with tabs: HISTORIA MEDICA, HISTORIA DIETÉTICA, ANTROPOMETRIA, LABORATORIOS, DIETAS (selected), and MENÚS. Below the tabs, the 'Requerimiento Calórico' is calculated as 75.64 kg (West) multiplied by 25 kcal, resulting in 1891 kcal. Patient information includes: Género: Masculino, Edad: 57, Estatura: 185.00 cms, and Peso Actual: 60.00 kg. A table below shows the distribution of macronutrients:

FD	%	kcal	g
Proteínas	8.03	152	0,5 = 38
Lípidos	0	0	0
Carbohidratos	91.97	1739	434.75

Figura 4.23: Dietas 1

Seguido a los datos del paciente se encuentra la fórmula dietética, la cual está compuesta por los macronutrientes (proteínas, lípidos y carbohidratos), el porcentaje de estos macronutrientes que se aportaran, las kilocalorías y los gramos que representan. En esta sección el usuario soló debe colocar los valores correspondientes a los gramos de proteínas y el porcentaje de grasas que desea aportar con la dieta, el resto del cuadro se determina de manera automática.

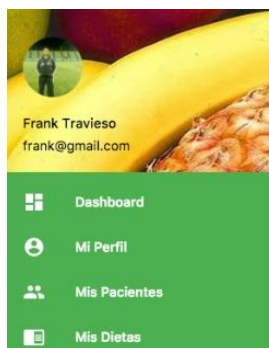
Posteriormente, después de distribuir porcentualmente los macronutrientes, se deben distribuir mediante el método por intercambios, el cual se muestra en la figura 4.24, en esta sección de la opción “Dietas”, el nutricionista debe colocar los intercambios correspondientes a: lácteos, vegetales y frutas y automáticamente el resto de los intercambios con su respectivas proteínas, lípidos y carbohidratos son determinados automáticamente por la aplicación.



	Intercambios	Proteínas (g)	Lípidos (g)	Carbohidratos (g)
Lácteos (entero)	1	8	8	12
Vegetales	2	4	-	10
Frutas	4	-	-	60
Subtotal 1				82
Cereales	10.5	21	-	158
Subtotal 2		33		
Carnes 10%	6	42	30	-
Subtotal 3			38	
Grasas	4	-	20	-
Total		75	58	240

Figura 4.24: Dietas 2

La sección de lácteos y carnes, se subdividen en grupos (los cuales dependen de la cantidad de grasa que contengan), y según los requerimientos y la patología asociada del paciente, el nutricionista deberá elegir uno de estos subgrupos antes de colocar los intercambios; así como se muestra en la figura 4.25 y 4.26.



	Intercambios	Proteínas (g)	Lípidos (g)	Carbohidratos (g)
<ul style="list-style-type: none"> ✓ Lácteos (entero) Lácteos (bajo en grasa) Lácteos (muy bajo en grasa) 	0	8	8	12
Vegetales	0	4	-	10
Frutas	0	-	-	60
Subtotal 1				82
Cereales	10	21	-	158

Figura 4.25: Dietas (Leches)

Vegetales	0	4	-	10
Frutas	0	-	-	60
Subtotal 1				82
Cereales	10	21	-	158
Subtotal 2		33		
<input checked="" type="checkbox"/> Carnes 10% <input checked="" type="checkbox"/> Carnes 17% <input type="checkbox"/> Carnes +17%	6	42	30	-
Subtotal 3				38

Figura 4.26: Dietas (Carnes)

4.5.10 Interfaz “Menú”

Para finalizar el flujo de trabajo del nutricionista, este debe crear un menú personalizado y adaptado a las necesidades del paciente, el cual se realiza en la interfaz que se muestra en la figura 4.26, en donde se realiza la distribución de los intercambios por tiempo de comida

V-9266923 - Rose Mary Peña

HISTORIA MEDICA HISTORIA DIETÉTICA ANTROPOMETRIA LABORATORIOS DIETAS MENÚ

Intercambios:

Lácteos (entero): 1	Vegetales: 2	Frutas: 4
Cereales: 10.5	Carnes 17%: 6	Grasas: 4

Desayuno +

Nro Lista	Intercambio(s)	Comida
2	▼ 1,5	▼

Merienda +

Figura 4.27: Menú

Por tiempo de comida se determina primero el grupo de alimentos y el número de intercambios de este alimento para esta comida en específico, cómo se muestra en la imagen 4.28



Figura 4.28: Menú

4.5.11 Interfaz “Mis Alimentos”

A parte del flujo usual, los nutricionistas pueden crear y administrar sus propios alimentos. En la imagen 4.29 se muestra la ventana titulada “Mis Alimentos”, esta ventana aparece posterior a la selección de la opción “Mis Alimentos” en el menú de navegación (ubicado a la Izquierda).

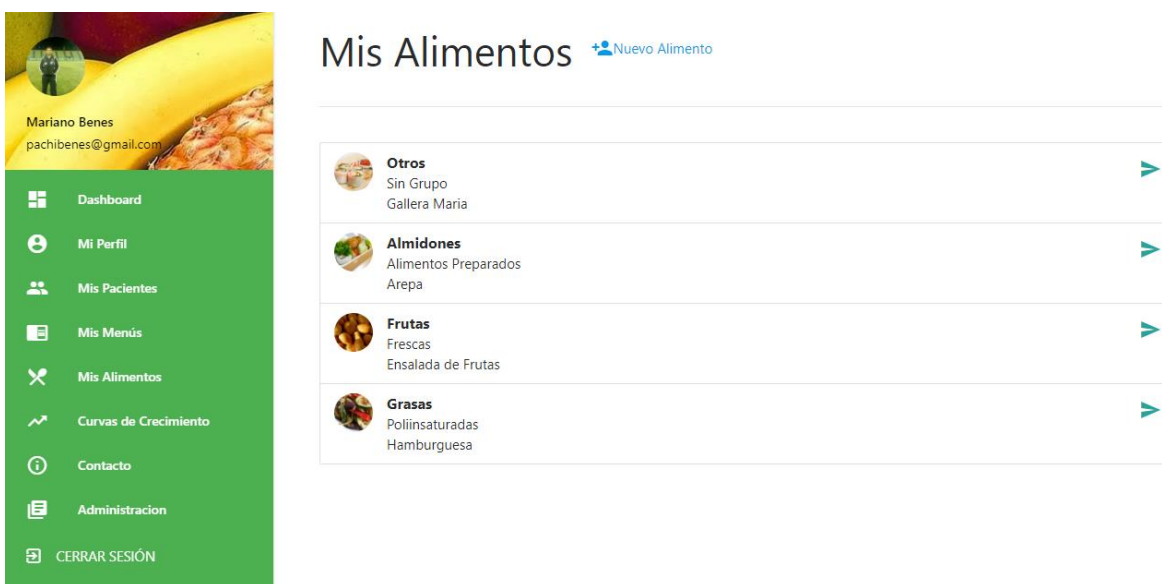


Figura 4.29: Mis Alimentos

En la parte central superior al lado de “Mis alimentos”, se encuentra la opción de “Nuevo alimento” donde el nutricionista puede agregar alimentos personalizados a su repertorio, incluyendo los macronutrientes que el alimento contenga, unidades y medidas. Adicionalmente puede incluir una foto de mismo.

La parte central de la ventana se muestran todos los alimentos que han sido creados por un usuario, donde puede hacer clic en cualquiera de ellos para editar su información.

4.5.12 Interfaz “Registro de Alimentos”

Al hacer clic en opción “Nuevo alimento” en la sección “Mis alimentos” expuesto en la imagen 4.28, lo redirecciona a la siguiente a la ventana de “registro de paciente”, mostrada en la imagen 4.30

Registro de Alimentos - 1 Intercambio

Grupo de Alimento

Nombre Descripción

Calorías Proteínas Carbohidratos Grasas

0 0 0 0

Cantidad Unidad Medica Practica Unid. Practica

0 0

IMAGEN

AGREGAR ALIMENTO

Figura 4.30: Registro de Alimento

En la imagen 4.30, se muestra el formulario general donde se registran los datos del alimento a agregar, considerando como datos principales: datos personales, hábitos psicobiológicos, antecedentes personales y familiares y los tratamientos nutricionales y médicos que esté cumpliendo el paciente al momento de ser evaluado. Al completar un ítem e ir al siguiente, se puede usar el mouse o la tecla “tabular” del teclado.

Una vez completado todo el formulario del alimento, se debe hacer clic en la parte inferior “Agregar alimento”, para finalizar el proceso de registrar el alimento en la base de datos de la aplicación. La información contenida en esta ventana puede ser editada si el profesional así lo decide.

4.5.13 Interfaz “Alimento”

Al hacer clic cualquier de los alimentos registrados en la sección “Mis alimentos” expuesto en la imagen 4.28, lo redirecciona a la siguiente a la ventana de “Alimento”, mostrada en la imagen 4.31

Mariano Benes
pachibenes@gmail.com

- Dashboard
- Mi Perfil
- Mis Pacientes
- Mis Menús
- Mis Alimentos
- Curvas de Crecimiento
- Contacto
- Administración
- CERRAR SESIÓN

Alimento - Arepa

ELIMINAR ALIMENTO

Almidones - Alimentos Preparados

Nombre	Descripcion		
Arepa	Arepa		
Calorias	Proteinas	Carbohidratos	Grasas
26,00	16,00	68,00	18,00
Cantidad	Unidad	Medica Practica	Unidad Practica
100,00	Unidad	1,00	Unidad Practica

IMAGEN

ACTUALIZAR ALIMENTO

Figura 4.31: Alimento

La opción “eliminar” que se encuentra en la parte superior derecha en esta ventana, se crea con la finalidad de eliminar un alimento creado por el usuario. En la parte inferior se encuentra la opción “Actualizar alimento”, el cual está diseñado para guardar cambios que se deseen realizar en las propiedades del alimento.

4.6 Las Iteraciones

Después de un análisis del Inventario de Funcionalidades se determina que se podía realizar la elaboración de la aplicación web en cinco iteraciones. Cada una de ellas con un tiempo aproximado de 1 semana en la que se realizan diversas actividades. A continuación, se describen las iteraciones realizadas, las actividades, los entregables y las reuniones realizadas durante el proceso.

4.6.1 Primera Iteración: configuraciones básicas de la aplicación

En esta primera iteración se realizan todas las funcionalidades básicas de la aplicación en la que se crea la estructura de directorios, la base de datos y el sistema principal de manejo de autenticación y autorización de usuarios. Las principales actividades realizadas fueron:

- Instalación del ambiente que aloja el servidor de desarrollo.
- Creación de la tabla de usuarios del sistema.
- Configuración de la librería de desarrollo y de la estructura de directorios.
- Conexión entre el servidor de desarrollo y la Base de Datos.
- Diseño del tema principal de la aplicación.
- Implementación de la pantalla de inicio.
- Proceso de verificación de credenciales.
- Creación de la pantalla de inicio del administrador con el acceso al módulo de creación de usuarios.

Al finalizar esta iteración el producto entregado cumple con las siguientes funcionalidades:

- Creación, modificación y eliminación de cuentas de usuario del sistema
- Ingreso de un usuario autenticado en el sistema mediante la verificación de un nombre de usuario y su contraseña asociada.
- Dependiendo del perfil de usuario que ingresa al sistema, se muestra la pantalla de inicio de cada uno de ellos.

4.6.2 Segunda Iteración: navegación, pacientes y laboratorios

La segunda iteración se enfoca en el desarrollo de todo lo relacionado con la pantalla principal tras iniciar sesión incluida la barra de navegación del sistema, adicionalmente se diseña el módulo de creación de paciente, historia médica y el módulo de exámenes de laboratorio. En tal sentido, se incluyeron en esta segunda iteración las siguientes actividades:

-
- Creación de la pantalla principal al iniciar sesión (dashboard).
 - Creación de las tablas de pacientes y laboratorios.
 - Implementación de los formularios de creación, edición y eliminación de pacientes.
 - Proceso de registro y asignación de exámenes de laboratorio.

Los entregables de la segunda iteración comprenden las siguientes funcionalidades:

- Pantalla principal del sistema.
- Barra de Navegación.
- Creación, modificación y eliminación de exámenes de laboratorio.
- Creación, modificación y eliminación de datos pacientes.

4.6.3 Tercera Iteración: antropometrías y fórmulas:

En la reunión de planificación correspondiente a la tercera iteración, se discuten diferentes temas acerca de los entregables de la segunda iteración y se detectan problemas con los datos registrados sobre el paciente, ya que no se habían tomado en cuenta información importante sobre la historia médica, patologías y tratamiento del mismo. Por otro lado, el enfoque principal de esta iteración fue el módulo de antropometría y todos los cálculos asociados a las mediciones del paciente. En consecuencia, se planificaron para esta iteración las siguientes actividades:

- Inclusión de datos faltantes a los formularios del paciente.
- Creación de la tabla de antropometrías.
- Proceso de registro y asignación de medidas antropométricas.
- Ajuste de peso según obesidad, edema o amputación.
- Calculo de fórmulas antropométricas.
- Graficas del progreso del paciente.
- Estado actual.

Los entregables para esta iteración cubren las siguientes funcionalidades:

- Creación, modificación y eliminación de datos antropométricos.
- Creación, modificación y eliminación de laboratorios.
- Visualización del progreso del paciente durante el tratamiento a través de graficas.
- Visualización del estado actual del paciente.

Después de la entrega y las pruebas realizadas a las nuevas funcionalidades del sistema, se determina en la reunión de revisión que faltan varios cálculos antropométricos relacionados al requerimiento calórico del paciente, pero como estos cálculos solo son utilizados a la hora de realizar una dieta se decide desarrollar estos requerimientos en la siguiente iteración junto con la visualización y creación de las mismas.

4.6.4 Cuarta Iteración: alimentos y dietas

En esta iteración se procede a desarrollar los procesos de creación, modificación y eliminación de los alimentos que son utilizados en el proceso de creación de menús, así como también se desarrollan los cálculos pendientes relacionados con el requerimiento calórico del paciente y a su vez el módulo de creación de dietas en el cual se utilizan dichos cálculos para culminar la realización de una de las funcionalidades principales de la aplicación.

Se determinan entonces las siguientes tareas de desarrollo en esta iteración:

- Implementación del formulario para la creación, modificación y eliminación de alimentos.
- Implementación del formulario para la creación, modificación y eliminación de una dieta asignada a un paciente.

Los entregables para esta iteración cubren las siguientes funcionalidades:

- Creación, modificación y eliminación de alimentos.
- Creación, modificación y eliminación de dietas.
- Visualización de los alimentos creados por el nutricionista.
- Visualización de las dietas asignadas a cada paciente.

4.6.5 Quinta Iteración: creación y visualización de menús

Las funcionalidades desarrolladas en esa iteración corresponden a la creación, modificación y eliminación de los menús que los nutricionistas hacen como ejemplo a sus pacientes para que estos hagan sus dietas. También se desarrolla el módulo en que los nutricionistas pueden crear, modificar y eliminar menús de ejemplo que, en principio, no están asociados específicamente a un paciente pero que pueden ser asignados a un paciente.

Se determinan entonces las siguientes tareas de desarrollo en esta iteración:

- Implementación del formulario para la creación, modificación y eliminación de un menú asignado a un paciente.
- Implementación del formulario para la creación, modificación y eliminación de una dieta estándar por parte del nutricionista.

Los entregables para esta iteración cubren las siguientes funcionalidades:

- Creación, modificación y eliminación de los menús asignados a cada paciente.
- Creación, modificación y eliminación de los menús estándar creados por el nutricionista.
- Visualización de los menús asignados a cada paciente.
- Visualización de los menús estándar creados por el nutricionista.
- Asignación de un menú estándar a un paciente.

4.6.6 Sexta iteración: curvas de crecimiento

En esta última iteración se desarrolla el módulo de curvas de crecimiento, en el cual se automatizan los cálculos correspondientes al índice de masa corporal y el despliegue de las gráficas donde se puede observar la evolución de los pacientes menores de 19 años en cuanto a su peso, talla e índice de masa corporal, así como también permitirle al nutricionista observar en el gráfico en que percentil se ubica cada una de las mediciones. Se realizan entonces las siguientes actividades durante esta iteración:

- Implementación del formulario para recolección de datos del paciente que permita añadir varias consultas.
- Cálculo de índice de masa corporal con los datos recolectados en el formulario.
- Gráficas de ubicación del paciente con respecto a los percentiles estándar.

Los entregables para esta iteración cubren las siguientes funcionalidades:

- Recolección de datos del paciente.
- Añadir consultas.
- Visualización de evolución del paciente y ubicación con respecto a los percentiles estándar.

4.7 Las pruebas

A lo largo del desarrollo de la aplicación se realizaron múltiples pruebas para verificar el cumplimiento de los requerimientos funcionales. Para ello se utilizó un instrumento de evaluación, presentado en la figura 4.32, que consiste en una tabla donde se listan las funcionalidades requeridas o que se quieren evaluar de cada módulo junto con el estatus de aprobación (para cada funcionalidad) por parte del usuario final del sistema. Las pruebas fueron realizadas de forma cíclica por cada módulo con el soporte de profesionales del área de nutrición hasta que finalmente los mismos aprobaran o no aprobaran las funcionalidades necesarias para el módulo.

Nombre del Módulo	Sprint	Funcionalidad	Estatus
Modulo X	N	Funcionalidad 1	Estatus 1
		Funcionalidad 2	Estatus 2
		Funcionalidad N	Estatus N

Figura 4.32 - Instrumento de evaluación para las pruebas

En la Figura 4.33 se presenta el primer ciclo de pruebas llevado a cabo para el módulo de Pacientes

Nombre del Módulo	Ciclo	Funcionalidad	Estatus
Pacientes	1	Creación de Pacientes	No Aprobado
		Visualización de Paciente	Aprobado
		Edición de Paciente	Aprobado
		Listado de Pacientes	Aprobado
		Historia Médica	Aprobado
		Hábitos Psicobiológicos	No Aprobado
		Historia Dietética	No Aprobado
		Evaluación de Consumo de Alimentos	No Aprobado

Figura 4.33 – Primer ciclo de pruebas funcionales para el módulo de Pacientes

En el primer ciclo de pruebas para el módulo de Pacientes, los usuarios aprobaron la mayoría de las funcionalidades correspondientes al módulo. Sin embargo, cuatro funcionalidades que presentaban errores en los formularios, en los cuales faltaban datos importantes. Se integraron cambios y correcciones sobre las funcionalidades no aprobadas y se procedió a realizar un segundo ciclo de pruebas, el cual se presenta en la tabla 4.34:

Nombre del Modulo	Ciclo	Funcionalidad	Estatus
Pacientes	2	Creación de Pacientes	Aprobado
		Hábitos Psicobiológicos	Aprobado
		Historia Dietética	Aprobado
		Evaluación de Consumo de Alimentos	Aprobado

Figura 4.34 – **Segundo ciclo de pruebas funcionales para el módulo de Pacientes**

En el segundo ciclo de pruebas para el módulo de Pacientes, los usuarios aprobaron las funcionalidades que faltaban por aprobación, quedando así culminado el desarrollo de este módulo.

En la Figura 4.35 se presenta el ciclo de pruebas llevado a cabo para el módulo de Laboratorios

Nombre del Modulo	Ciclo	Funcionalidad	Estatus
Laboratorios	1	Creación de Laboratorios	Aprobado
		Asignación de Laboratorios	Aprobado
		Visualización de Valores	Aprobado
		Listado de Laboratorios	Aprobado
		Edición de Laboratorio	Aprobado

Figura 4.35 – **Primer ciclo de pruebas funcionales para el módulo de Laboratorios**

El total de las funcionalidades comprendidas por este módulo fue aprobado en el primer ciclo de pruebas con los usuarios finales, por lo que no hubo necesidad de hacer otro ciclo de pruebas con este módulo y queda así culminado el desarrollo del mismo.

En la Figura 4.36 se presenta el primer ciclo de pruebas llevado a cabo para el módulo de Antropometrías.

Nombre del Modulo	Ciclo	Funcionalidad	Estatus
Antropometrías	1	Creación de Antropometrías	No Aprobado
		Asignación de Mediciones	Aprobado
		Visualización de Mediciones	Aprobado
		Listado de Mediciones	Aprobado
		Edición de Mediciones	Aprobado
		Cálculos de Formulas	No Aprobado
		Graficas	No Aprobado
		Ultimas Antropometrías del Paciente	No Aprobado

Figura 4.36 – Primer ciclo de pruebas funcionales para el módulo de Antropometrías

En el primer ciclo de pruebas para el módulo de Antropometrías, los usuarios aprobaron la mayoría de las funcionalidades correspondientes al módulo. Sin embargo, faltaron por aprobar cuatro funcionalidades. Todas estaban relacionadas con datos faltantes que se debían ser ingresados o calculados. Se integraron cambios y correcciones sobre las funcionalidades no aprobadas y se procedió a realizar un segundo ciclo de pruebas, el cual se presenta en la tabla 4.37:

Nombre del Modulo	Ciclo	Funcionalidad	Estatus
Antropometrías	2	Creación de Antropometrías	Aprobado
		Cálculos de Formulas	Aprobado
		Graficas	Aprobado
		Ultimas Antropometrías del Paciente	Aprobado

Figura 4.37 – Segundo ciclo de pruebas funcionales para el módulo de Antropometrías

En el segundo ciclo de pruebas para el módulo de Antropometrías, los usuarios aprobaron las funcionalidades que faltaban por aprobación, quedando así culminado el desarrollo de este módulo.

En la Figura 4.38 se presenta el ciclo de pruebas llevado a cabo para el módulo de Alimentos

Nombre del Modulo	Ciclo	Funcionalidad	Estatus
Alimentos	1	Creación de Alimentos	Aprobado
		Visualización de Alimentos	Aprobado
		Listado de Alimentos	Aprobado
		Edición de Alimentos	Aprobado

Figura 4.39 – Primer ciclo de pruebas funcionales para el módulo de Alimentos

El total de las funcionalidades comprendidas por este módulo fue aprobado en el primer ciclo de pruebas con los usuarios finales, por lo que no hubo necesidad de hacer otro ciclo de pruebas con este módulo y queda así culminado el desarrollo del mismo.

En la Figura 4.40 se presenta el ciclo de pruebas llevado a cabo para el módulo de Dietas

Nombre del Modulo	Ciclo	Funcionalidad	Estatus
Dietas	1	Creación de Dietas	No Aprobado
		Asignación de Dietas	Aprobado
		Visualización de Dieta	No Aprobado
		Listado de Dietas	No Aprobado
		Edición de Dieta	Aprobado
		Cálculos de Formula Dietética	No Aprobado
		Calculo de Requerimiento Calórico	No Aprobado
		Asignación de Intercambios	No Aprobado

Figura 4.40 – Primer ciclo de pruebas funcionales para el módulo de Dietas

En el primer ciclo de pruebas para el módulo de Dietas, fue el menos aprobado de todos ya que los usuarios aprobaron solamente 2 de las funcionalidades correspondientes. Al ser modulo que más tuvo problemas se debió hacer gran énfasis en la segunda fase de desarrollo del mismo. Se integraron cambios y correcciones sobre las funcionalidades no aprobadas y se procedió a realizar un segundo ciclo de pruebas, el cual se presenta en la tabla 4.41:

Nombre del Modulo	Ciclo	Funcionalidad	Estatus
Dietas	2	Creación de Dietas	Aprobado
		Visualización de Dieta	Aprobado
		Listado de Dietas	Aprobado
		Cálculos de Formula Dietética	Aprobado
		Calculo de Requerimiento Calórico	Aprobado
		Asignación de Intercambios	Aprobado

Figura 4.41 – Segundo ciclo de pruebas funcionales para el módulo de Dietas

En el segundo ciclo de pruebas para el módulo de Dietas, los usuarios aprobaron las funcionalidades que faltaban por aprobación, quedando así culminado el desarrollo de este módulo.

En la Figura 4.42 se presenta el ciclo de pruebas llevado a cabo para el módulo de Menús

Nombre del Modulo	Ciclo	Funcionalidad	Estatus
Menus	1	Creación de Menus	Aprobado
		Asignación de Menus	Aprobado
		Visualización de Menú	Aprobado
		Edición de Menus	Aprobado

Figura 4.42 – Primer ciclo de pruebas funcionales para el módulo de Menús

El total de las funcionalidades comprendidas por este módulo fue aprobado en el primer ciclo de pruebas con los usuarios finales, por lo que no hubo necesidad de hacer otro ciclo de pruebas con este módulo y queda así culminado el desarrollo del mismo.

En la Figura 4.43 se presenta el ciclo de pruebas llevado a cabo para el módulo de Administrador

Nombre del Modulo	Ciclo	Funcionalidad	Estatus
Menus	1	Creación de Usuario	Aprobado
		Visualización de Usuario	Aprobado
		Listado de Usuarios	Aprobado
		Edición de Usuario	Aprobado
		Asignación de Pacientes a Usuarios	Aprobado

Figura 4.43 – **Primer ciclo de pruebas funcionales para el módulo de Administrador**

El total de las funcionalidades comprendidas por este módulo fue aprobado en el primer ciclo de pruebas con los usuarios finales, por lo que no hubo necesidad de hacer otro ciclo de pruebas con este módulo y queda así culminado el desarrollo del mismo.

En la Figura 4.44 se presenta el ciclo de pruebas llevado a cabo para el módulo de Curvas de Crecimiento

Nombre del Modulo	Ciclo	Funcionalidad	Estatus
Curvas de Crecimiento	1	Cálculos Antropométricos	Aprobado
		Graficación del Paciente	Aprobado
		Graficación de Percentiles	Aprobado

Figura 4.44 – **Primer ciclo de pruebas funcionales para el módulo de Curvas de Crecimiento**

El total de las funcionalidades comprendidas por este módulo fue aprobado en el primer ciclo de pruebas con los usuarios finales, por lo que no hubo necesidad de hacer otro ciclo de pruebas con este módulo y queda así culminado el desarrollo del mismo.

CONCLUSIONES

Los objetivos de este Trabajo Especial de Grado se cumplieron de manera satisfactoria al desarrollar una aplicación web que les permite a los profesionales del área de nutrición llevar a cabo procesos que le permiten realizar diagnóstico y tratamiento de sus pacientes de manera más sencilla y automatizar muchos procesos que realizaban manualmente lo que les hacía desperdiciar mucho tiempo y recursos materiales, así como también pueden con esta aplicación hacer seguimiento a la evolución de sus pacientes.

A pesar de poder hacer una evaluación para determinar si la aplicación sería aceptada en su mayoría por los profesionales del área o si su impacto en el área sería significativo se logró cumplir con las necesidades principales expresadas por los nutricionistas involucrados en este proceso de desarrollo, a los cuales se pudo brindar satisfacción y comodidad con el desarrollo de esta aplicación ya que la misma más allá de cubrir sus necesidades principales con respecto a los procesos de diagnóstico y tratamiento también les ofrece otras funcionalidades como por ejemplo la creación de plantillas de menús que agilizan el proceso de tratamiento o el ingreso de nuevos alimentos los cuales pueden utilizar en el proceso de creación de menús.

Por otra parte, el uso de la metodología Scrum permitió agilizar el proceso de desarrollo cumpliendo con los tiempos estimados cumpliendo con todos los requerimientos y garantizando que la aplicación desarrollada tuviera una excelente calidad desde sus partes más pequeñas hasta su parte global.

Es importante resaltar que las tecnologías escogidas permitieron que el desarrollo fuese rápido y organizado ya que la estructura que brinda el marco de trabajo Laravel hace posible el desarrollo eficaz de los módulos, en cada uno de los cuales se podían desarrollar tanto la lógica de la aplicación como las interfaces de usuario en paralelo. Esta estructura también brindó comodidad a la hora de organizar el código y utilizar el patrón MVC, así como también hacer este mucho más legible y entendible lo que será beneficioso para desarrolladores que quieran mejorar la aplicación en un futuro o extraer algo de ella. También fue notorio que el uso de Laravel junto con HTML, CSS, JQuery y Materialize hizo posible que se pudieran presentar resultados rápidos y satisfactorios a los problemas planteados. Es de mucha importancia mencionar que estas tecnologías hacen que la aplicación pueda tener un alcance importante a futuro, sobre todo en Venezuela ya que son soportadas por los navegadores más populares, en el caso de las utilizadas del lado del cliente, y por una gran cantidad de servidores, en el caso de las utilizadas del lado del servidor.

En cuanto a las pruebas podemos concluir que se cumplieron todas las expectativas iniciales en cuanto a las funcionalidades de la aplicación, en cada fase debido a la falta de conocimiento de la metodología de trabajo de los nutricionistas existieron problemas en el diseño de ciertos módulos, afortunadamente después cada reunión con los nutricionistas se pudieron aclarar todas las dudas existentes por parte del equipo hasta ese punto y de esta manera solventar todos los problemas encontrados

En general se puede concluir que con este desarrollo se proporciona a los nutricionistas una serie importantes de mejoras y beneficios para su desenvolvimiento profesional, entre los cuales podemos mencionar:

- Facilidad a la hora de almacenar, acceder y controlar los datos de los pacientes que acuden a sus consultas.
- Disminución de los tiempos de duración por consulta, permitiéndoles atender a mayor cantidad de pacientes en menor cantidad de tiempo.
- Reducción de costos en cuanto a materiales se refiere (papel, archivadores, etc.).
- Generación automatizada y personalizada de las dietas y menús relacionados con cada paciente.
- Acceso a la información desde cualquier lugar donde se cuente con internet y un navegador web, antes el acceso a esta información se veía limitado al consultorio o el sitio donde estuvieran los archivos físicos.
- Facilidad de realizar trabajos de campo al poder acceder desde cualquier dispositivo móvil que posea internet y un navegador web.

4.1 Limitaciones

Durante el desarrollo del proyecto se presentaron las siguientes limitaciones:

- La falta de un servidor en el cual se pudiera instalar la aplicación durante el desarrollo para permitir la realización de pruebas por parte de los profesionales en el área de nutrición de manera remota, por lo cual para que los mismos probaran e hicieran las respectivas observaciones se tuvieron que realizar reuniones presenciales.
- No se contaba con la ayuda de un diseñador especializado en el área de diseño web por lo que se invirtió mucho tiempo en idear y crear las interfaces de usuario y se hizo difícil evaluar si las mismas eran agradables a la vista, estaban bien distribuidos los elementos dentro de ellas, entre otras características.

4.2 Recomendaciones y Trabajos Futuro

Para un futuro se proponen los siguientes trabajos relacionados a este tema:

- Agregar un módulo basado solo en pacientes hospitalizados ya sea en habitaciones o en las salas de cuidados críticos, donde se incluyan las fórmulas especiales para determinar su requerimiento y el factor de estrés que estos ameritan según la patología presentada.
- Crear un módulo específico para la población deportiva, que permita calcular sus requerimientos según la demanda del mismo 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 encuentre 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.
- Crear un módulo que le permita a los pacientes acceder al sistema y visualizar toda su información.
- 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 de que pueda ser personalizado para cada empresa o particular.

REFERENCIAS BIBLIOGRÁFICAS Y DIGITALES

- Adobe Dreamweaver*. (4 de Diciembre de 2016). Obtenido de Aspectos básicos de las aplicaciones Web:
<https://helpx.adobe.com/es/dreamweaver/using/web-applications.html>
- Alecia, A. (2011). *AgilUs: un metodo ágil de desarrollo de software que incorpora la usabilidad*. Caracas, Venezuela: Universidad Central de Venezuela.
- Cáceres, P. (2015). *Procesos Ágiles para el desarrollo de aplicaciones web*. Madrid, España: Universidad Rey Juan Carlos. Obtenido de <http://www.dlsi.ua.es/~jaime/webe/articulos/s112.pdf>
- Chávez, M. A. (Marzo de 2015). *Los Frameworks de Ruby*. Obtenido de Medium:
<https://medium.com/codeando-con-ruby/los-frameworks-de-ruby-1b340f311885#.n0spfv48>
- EcuRed. (Julio de 2012). *Lenguaje de MARcado de Hipertexto*. Obtenido de EcuRed:
http://www.ecured.cu/index.php/Lenguaje_de_Marcado_de_Hipertexto
- Educación, C. d. (Marzo de 2015). *Objetivo de la Educación Nutricional*. Obtenido de CEIP:
<http://ceiplapatacona.edu.gva.es/servicios-y-recursos/plan-de-comedor-escolar/objetivos-de-la-educacion-alimentaria/>
- Ferrer, A. G.-R. (28 de 08 de 2010). *¿Qué es un dietista-nutricionista?* Obtenido de ABCDietistas:
http://www.abcdietas.com/articulos/dietetica/dietista_nutricionista.html
- Kinanthropometry, I. S. (s.f.). *International Standards for Anthropometric Assessment*. Obtenido de
<http://www.ceap.br/material/MAT17032011184632.pdf>
- Luján, S. (2002). *Programación de aplicaciones web: historia, principios básicos y clientes web*. Alicante: Editorial Club Universitario. Obtenido de <http://gplsi.dlsi.ua.es/~slujan/libros>.
- Mozilla Developer Network*. (Octubre de 2016). Obtenido de CSS:
<https://developer.mozilla.org/es/docs/Web/CSS>
- Nutrición*. (2017). Obtenido de Organización Mundial de la Salud:
<http://www.who.int/topics/nutrition/es/>
- Perez, V. (Mayo de 2016). *Sistema de Gestión de Información Científica*. Caracas, Venezuela.
- Queralt, D. M. (s.f.). *Requerimientos nutricionales*. Obtenido de Salud Canales Mapfre.
- Universidad Nutrición*. (s.f.). Obtenido de Universidad Nutrición:
<http://www.universidadnutricion.org/software-de-nutricion.php>
- Violeta, A. (02 de Febrero de 2002). *Educación nutricional: para una alimentación saludable*. Obtenido de En plenitud: <http://www.enplenitud.com/educacion-nutricional-para-una-alimentacion-saludable.html>
- World Health Organization. (2007). *WHO Anthro*. Obtenido de WHO:
http://www2.paho.org/hq/index.php?option=com_docman&task=doc_download&gid=17835&Itemid=270

World Health Organization. (2009). *Who AnthroPlus*. Obtenido de WHO:
http://www.who.int/growthref/tools/who_anthroplus_manual.pdf