

UNIVERSIDAD CENTRAL DE VENEZUELA FACULTAD DE CIENCIAS

ESCUELA DE COMPUTACIÓN

APLICACIONES CON TECNOLOGÍA A INTERNET

**Tópicos a considerar para el desarrollo de una aplicación Web orientada al área de Recursos humanos de cualquier organización**

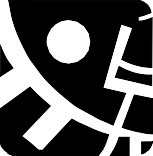
Trabajo Especial de Grado presentado ante la ilustre Universidad Central de Venezuela por:

Br. Kevin Miranda Br. Luis Campos

Para optar por el título de Licenciado en Computación

Tutor:

Prof. José Sánchez UCV Caracas, Febrero 2018

**Universidad Central de Venezuela**

Facultad de Ciencias Escuela de Computación

Aplicaciones con Tecnología a Internet

**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 **Kevin Miranda** C.I.: **26.150.260** y **Luis Campos** C.I.: **18.033.084** con el título “Tópicos a considerar para el desarrollo de una aplicación Web orientada al área de Recursos humanos de cualquier organización”, 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 de del a las , para que sus autores lo defendieron en forma pública, en

, 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 de del , dejándose también constancia de que actuó como coordinadora del Jurado la Profesora Concettina di Vasta

Prof. José Sánchez (Tutor)

Profa. Brenda López Profa. Mercy Ospina (Jurado) (Jurado)

**Agradecimientos**

UNIVERSIDAD CENTRAL DE VENEZUELA FACULTAD DE CIENCIAS

ESCUELA DE COMPUTACIÓN

APLICACIONES CON TECNOLOGÍA A INTERNET

**Tópicos a considerar para el desarrollo de una aplicación Web orientada al área de Recursos humanos de cualquier organización**

**Autores**: Kevin Miranda

Luis Campos

**Tutor**: Prof. José Sánchez

**Fecha**: XX/02/2018

**RESUMEN**

El presente Trabajo Especial de Grado describe la problemática que presentan las empresas para gestionar un proceso de selección de manera eficiente debido a la gran carga de currículums que deben filtrarse y cuestionarios que deben aplicarse de forma manual abarcando grandes pérdidas económicas en traslados, refrigerios, documentos impresos, entre otros; y a los candidatos de requerir realizar múltiples visitas a la sucursal de la empresa para realizar cada una de las etapas del proceso de selección y la incertidumbre que genera la falta de respuesta oportuna de algunas empresas. Conociendo las necesidades e inconvenientes existentes, se plantea el desarrollo de una aplicación web que permita a las empresas publicar y configurar las vacantes disponibles de forma que se procesen ciertas etapas del proceso de forma automatizada, dando respuesta a los candidatos que se postulen de forma automatizada y permita a los candidatos obtener información completa y detallada de las vacantes al cual le interesen postularse y pueda realizar las etapas del proceso asociado a la vacante a cualquier día y hora de la semana llamada INSERTE EL NOMBRE DE LA APLICACION, una aplicación web creada con tecnologías de software libre (HTML5, CSS3, JavaScript, jQuery, Bootstrap, PHP, Laravel y PostgreSQL) utilizando la metodología XP para el desarrollo ágil de la aplicación.

**Palabras clave:** candidato, configurable, proceso de selección, etapas de selección, aplicación web.

**TABLA DE CONTENIDO**

[CAPÍTULO 1 - PROBLEMA DE INVESTIGACIÓN 11](#_gjdgxs)

* 1. [Planteamiento del Problema 11](#_30j0zll)
  2. [Objetivos 13](#_1fob9te)
  3. [Arquitectura de la solución 13](#_3znysh7)
  4. [Justificación de la solución 15](#_2et92p0)

[CAPÍTULO 2 - MARCO CONCEPTUAL 16](#_tyjcwt)

* 1. [Antecedentes 17](#_3dy6vkm)
     1. [Alimentación Requerimiento Nutricional y Adecuación CANIA (ARNAC) 17](#_1t3h5sf)
     2. [NUTRIMIND 18](#_4d34og8)
     3. [WHO Anthro 20](#_2s8eyo1)
     4. [WHO AnthroPlus 23](#_17dp8vu)
     5. [Método manual 23](#_3rdcrjn)
  2. [Conceptos Básicos relacionados con el área de Nutrición 27](#_26in1rg)
  3. [Arquitectura cliente-servidor 28](#_lnxbz9)
  4. [Aplicación web 29](#_35nkun2)
     1. [Funcionamiento de una aplicación web 29](#_1ksv4uv)
     2. [Ventajas 31](#_44sinio)
     3. [Desventajas 31](#_2jxsxqh)
     4. [Patrón de Modelo Vista Controlador (MVC) 32](#_z337ya)
  5. [Tecnologías del lado del cliente 34](#_3j2qqm3)
     1. [Lenguaje de marcas de hipertexto 34](#_1y810tw)
     2. [Hojas de estilo en cascada 34](#_4i7ojhp)
     3. [Javascript 34](#_2xcytpi)
     4. [JQuery 35](#_1ci93xb)
     5. [Materialize 35](#_3whwml4)
  6. [Tecnologías del lado del servidor 35](#_2bn6wsx)
     1. [PHP 36](#_qsh70q)
     2. [Sistemas Manejadores de Bases de Datos 38](#_3as4poj)

[CAPÍTULO 3 - MARCO METODOLÓGICO 39](#_1pxezwc)

[3.1 Metodologías de Desarrollo 39](#_49x2ik5)

[3.1.1 Metodologías Ágiles 40](#_2p2csry)

* 1. [Análisis Preliminar 48](#_147n2zr)
  2. [Diagrama de Casos de Uso 49](#_3o7alnk)
  3. [Diseño de la Solución 52](#_23ckvvd)
  4. [Desarrollo de la solución 52](#_ihv636)
     1. [Base de Datos 52](#_32hioqz)
     2. [Tecnologías involucradas en el desarrollo de la Aplicación 55](#_1hmsyys)
  5. [Flujo de la aplicación web e Interfaces de Usuario 59](#_41mghml)
     1. [Pantalla de acceso 59](#_2grqrue)
     2. [Interfaz “Dashboard” 59](#_vx1227)
     3. [Interfaz “Mis pacientes” 60](#_3fwokq0)
     4. [Interfaz “Registro del Paciente” 61](#_1v1yuxt)
     5. [Interfaz “Paciente” 62](#_4f1mdlm)
     6. [Interfaz “Historia Dietética” 63](#_2u6wntf)
     7. [Interfaz “Antropometría” 64](#_19c6y18)
     8. [Interfaz “Nuevo Laboratorio” 66](#_3tbugp1)
     9. [Interfaz “Dietas” 68](#_28h4qwu)
     10. [Interfaz “Menú” 70](#_nmf14n)
     11. [Interfaz “Mis Alimentos” 71](#_37m2jsg)
     12. [Interfaz “Registro de Alimentos” 72](#_1mrcu09)
     13. [Interfaz “Alimento” 72](#_46r0co2)
  6. [Las Iteraciones 74](#_2lwamvv)
     1. [Primera Iteración: configuraciones básicas de la aplicación 74](#_111kx3o)
     2. [Segunda Iteración: navegación, pacientes y laboratorios 74](#_3l18frh)
     3. [Tercera Iteración: antropometrías y fórmulas: 75](#_206ipza)
     4. [Cuarta Iteración: alimentos y dietas 76](#_4k668n3)
     5. [Quinta Iteración: creación y visualización de menús 76](#_2zbgiuw)
     6. [Sexta iteración: curvas de crecimiento 77](#_1egqt2p)
  7. [Las pruebas 78](#_3ygebqi)

[CONCLUSIONES 84](#_2dlolyb)

* 1. [Limitaciones 86](#_sqyw64)
  2. [Recomendaciones y Trabajos Futuro 86](#_3cqmetx)

[REFERENCIAS BIBLIOGRÁFICAS Y DIGITALES 87](#_1rvwp1q)

**INTRODUCCIÓN**

La necesidad de contratar personal en las empresas viene dado por diferentes causas como expansión de la empresa, generación de nuevos puestos de trabajo, rotación de personal, un requerimiento o simple necesidad de fuerza laboral, y lo cual depende de las capacidades y aptitudes del talento humano, por lo cual es de vital importancia realizar una selección adecuada de candidatos que contribuyan al éxito de las organizaciones.

En la actualidad, la mayoría de los usuarios revisan portales de empleo a través de internet, y pueden visualizar información en relación a las vacantes de su interés. El usuario decide postularse, y de ser seleccionado comienza un proceso de evaluación por parte de la empresa. Dicho proceso involucra la interacción de la empresa y del usuario, y en el cual se origina una serie de inconvenientes para ambas partes, originando pérdidas de tiempo, dinero, de recursos humanos aptos para dichas vacantes, frustración, problemas de competitividad en las empresas, entre otras.

Entre las causas se puede determinar la creación de procesos ineficiente en la selección, evaluación y contratación de talento humano que se encuentra en la búsqueda de oportunidades laborales; y en donde algunas empresas deciden crear sus propios procesos de selección, evaluación y contratación de personal, y en otros casos, contratan a otras empresas para realizar dichas labores, pero incluso, las empresas especializadas en la búsqueda y selección de talento, han perdido a excelentes profesionales, por utilizar procesos convencionales que no les permite manejar un volumen considerable de candidatos, de forma eficaz.

Por lo antes expuesto, se propone el desarrollo de una aplicación web, que sea configurable, que permita generar y ofrecer información acerca de las diferentes vacantes ofrecidas por una empresa, que permita definir y personalizar las diferentes etapas de selección y evaluación de potenciales candidatos, que permita atender a una gran cantidad de candidatos de manera simultánea, y proporcionales una respuesta rápida; en cada una de las etapas del proceso de selección; permitiendo que las empresas puedan descartar a diversos candidatos, de forma rápida y eficiente, sin originar los inconvenientes descritos; y que administre la información obtenida de los candidatos de manera que permita generar métricas que la empresa pueda utilizar para tomar mejores decisiones acerca de sus procesos de selección.

Por dicha razón se ha optado por la propuesta de utilizar tecnologías del lado del cliente como Bootstrap, JavaScript, y jQuery; así como tecnologías del lado del servidor como el lenguaje de programación PHP, el Framework Laravel, utilizando el patrón MCV, y patrones de diseño para su construcción, utilizando la metodología de desarrollo XP, ya que permiten un desarrollo rápido robusto y con mucho soporte documental de como aplicarlo, además de proveer una abstracciones útiles entre las fases del proyecto, aplicables al desarrollo y progreso de la aplicación, y que puede escalar sin importar la cantidad de personas e independientemente del propósito u de negocio de la aplicación.

Para lograr esto, 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

## Planteamiento del Problema

Hoy en día, el éxito de las organizaciones y empresas depende en gran medida de una buena selección de candidatos que desean aportar sus conocimientos, experiencias y aptitudes, para mejorar el funcionamiento y desarrollo de las tareas de la empresa. La necesidad de reclutar nuevos candidatos en las empresas, viene dado por diferentes causas como requerir personal que contribuya a realizar, de forma eficaz y eficiente, una serie de actividades que contribuyan al logro de las diversas áreas de una organización y así alcanzar la misión y visión de dicha organización, o debido a factores externos tales como expansión de la empresa, lo que genera nuevos puestos de trabajos, pérdida de persona o despido, entre otros.

Para cualquier empresa es sobreentendida la necesidad de contratar nuevo personal constantemente, sobre todo si se tiene un crecimiento de cualquier magnitud, o cualquier tarea sin atender que podría ser crucial para el desarrollo y progreso de la empresa, es por eso que se tiene la necesidad no solo de contratar, sino de captar candidatos que llenen dichos roles de la manera más eficiente posible, es simple estadística, mientras más candidatos se puedan procesar, más probabilidades se tienen de contratar a un candidato que satisfaga las expectativas de la organización. Sin embargo, muchas organizaciones no poseen una forma de trabajo eficiente que permita evaluar a una gran cantidad de candidatos en tiempo eficiente, debido a que, generalmente dichas evaluaciones son realizadas por una sola persona que suele dedicar parte de su tiempo para aplicar estos procesos de selección, evaluación y contratación, generando tiempos de respuestas dilatados para todos los participantes.

El proceso de reclutamiento de personal comienza a raíz de un requerimiento o solicitud que posee la empresa en desarrollar una serie de actividades, en un área determinada, donde la empresa carece de personal calificado para ejecutar dicha labor, donde posteriormente se procede a determinar el nombre de la vacante, responsabilidades y requerimientos mínimos, además de las propuestas salariales. Una vez que la empresa determina dichos datos sobre la vacante, se procede a realizar la publicación de la vacante, ya sea, en el sitio web de la empresa o en sitios web donde potenciales candidatos pueden encontrar información de la vacante, donde posteriormente, se inicia un proceso de solicitud de una serie de candidatos para ocupar una vacante. Dicho proceso trae una serie de inconvenientes tanto para el candidato como para la Empresa.

Para el candidato, es necesario que el candidato proporcione una documentación a la empresa, entre ellas se incluye el Currículum vitae, impreso, principalmente y en ciertos casos, copia de diplomas o títulos obtenidos, referencias laborales, certificaciones, y otros.

Al momento de realizar la primera entrevista, el postulante puede recibir información de temas asociados al horario laboral, como: flexibilidad de horario laboral y de estudio (si el candidato lo requiere), mientras cumple sus funciones en la organización, considerándose como un proceso largo y tedioso, sin embargo, el caso mencionado ocurre particularmente cuando la organización no especifica los beneficios y expectativas económicas que ofrecen; acarreando pérdidas de recursos a los involucrados, por conceptos de transportes, impresión y copias de documentos requeridos, y de tiempos en tránsito. Posterior a la primera entrevista, se suelen realizar una serie de evaluaciones técnicas, psicológicas, verbales, numéricas, y de comprensión espacial, y demás aspectos que son de importancia para la organización con el propósito de evaluar al candidato antes de decidir, originando que dicho candidato, deba realizar una inversiones adicionales de recursos sean por transporte, recursos o de tiempo, normalmente en horarios de oficina, que implica realizar solicitudes para asistir a dichos procesos para posteriormente recuperar el trabajo del día.

Luego de realizar los procesos realizados por la organización hacia cada uno de los postulados, la decisión final toma una cantidad de tiempo considerable en proporcionar una respuesta que le de información al postulado, sobre si fue seleccionado o no, generando incertidumbre la cual causa que los postulados abandonen el seguimiento y decidan iniciar procesos de selección con otras organizaciones.

Una vez finalizado todo el proceso de selección, y notificar a los postulados, se procede a entregar a cada uno de los postulados seleccionados una propuesta salarial, incluyendo los beneficios adicionales que adquieren al ingresar a la empresa; a partir de entonces, el postulado realiza el análisis de la propuesta y procede a tomar una decisión sobre la misma donde, en el caso de no ser de su agrado, es rechazada causando que todo el tiempo invertido en el proceso de selección se pierda. Los postulados que rechazan la propuesta salarial se retiran de la sucursal generando opiniones negativas hacia la empresa debido a que el tiempo y dinero invertido en el proceso se pierden ya que los beneficios y salarios son, por lo general, mencionados al final de todo el proceso de reclutamiento.

En cambio para la Empresa, por lo general, la información de la vacante es detallada al candidato de forma oral, por lo que la organización debe proporcionar información de la vacante al potencial candidato, de forma oral, o por medio de publicaciones en periódicos, revistas y sitios web, por lo que deben invertir un tiempo considerable de tiempo y dinero para atraer candidatos.

Existe la posibilidad, que el candidato decida retirarse del proceso debido a que no se cumple con las expectativas salariales, flexibilidad en los horarios laborales o fijación de horario que beneficie tanto al candidato como a la organización; como también la organización puede determinar que el candidato no cumple con los requisitos para la vacante. Todo el proceso requiere una inversión de tiempo mínimo de una hora por cada candidato, por lo que sí existe una cantidad elevada de candidatos, el costo en tiempo llega a ser muy ineficiente para las organizaciones.

Siempre que un candidato supera la primera etapa del proceso y decide continuar con el proceso de selección, las organizaciones proceden a realizar evaluaciones para conocer las destrezas profesionales de los candidatos, su nivel capacidad lógica y resolución de problemas, trabajo en equipo y bajo presión, entre otras; o entrevistas realizadas para conocer los valores, personalidad y aspectos de interés para el reclutador. Dicho proceso puede tomar hasta meses en algunas organizaciones, por lo que muchos candidatos no deciden esperar ese tiempo para tener una respuesta, y se postulan a otras vacantes laborales en otras organizaciones. Existen organizaciones que no le preguntan a los candidatos, cuáles son sus expectativas salariales al principio del proceso, a lo cual, luego de pasar el largo proceso de evaluación y selección, la empresa contacta al candidato seleccionado, y realizan una propuesta económica, vía email, o de forma presencial, la cual es posible que candidato no acepte, por no reflejar sus expectativas, o que el candidato ya consiguió empleo; originando que, la organización pierda dinero, por haber utilizado recursos que generan costos referentes a luz, telefono, impresión de material las pruebas a los candidatos, entre otras. Imagine, estimado lector, las pérdidas que puede tener una empresa, si aplicase dicha dinámica de selección, a 2 o 3 vacantes, y que por cada vacante, se disponga de al menos 10 candidatos. Por último, en la mayoría de los casos, las organizaciones fallan en notificar a los potenciales candidato si lograron superar una etapa o no de forma rápida y práctica, por lo que los candidatos asumen que no superaron el proceso, y deciden postularse a otras vacantes o continuar procesos de selección en otras organizaciones.

De lo antes expuesto, surge una propuesta que pretende mitigar el desperdicio de recursos para ambas partes, y a su vez ofrecer una aplicación web configurable, que permita agilizar los tiempos de respuesta para los mencionados, a través de una aplicación web configurable que les permita a las organizaciones definir la vacantes que requiere la organización, las responsabilidades asociadas a dicha vacante, determinar a qué persona reportaría el candidato seleccionado, los requisitos y cualidades que sean indispensables, y los que sean opcionales, información sobre flexibilidad y horarios de trabajo, propuesta salarial de dicha vacante; entre otros; que permita que las organizaciones puedan definir, de manera automatizada, una metodología de selección, evaluación y contratación de personal en la aplicación, según sus preferencias, que contribuya a que dicho proceso, sea lo más eficaz y eficiente, para ambas partes, en tiempo y en costo, que permita que los candidatos y los reclutadores, puedan obtener feedback oportuno, en cada una de las etapas del proceso de selección, mediante el uso de pantallas configurables y parametrizables; y, en caso de requerirse, complementarlo por medio de entrevistas en línea utilizando vías de comunicación a distancia conocidas como Skype.

## Objetivos

* + 1. **Objetivo General**
       - Desarrollar una aplicación Web configurable que agilice la selección del proceso de reclutamiento y selección definido por cada organización.
    2. **Objetivos Específicos**
       - Analizar los requerimientos funcionales y no funcionales de la aplicación a desarrollar.
       - Elaborar una dinámica de búsqueda, evaluación y selección del personal que sea configurable, y que contribuya a automatizar el proceso de búsqueda, selección y contratación de personal, para eliminar o mitigar los inconvenientes descritos en el punto anterior.
       - Elaborar procedimientos flexibles y configurables, que puedan adaptarse a cualquier empresa que requiera contratar de personal, sin importar su escala y complejidad.
       - Desarrollar los requerimientos de la aplicación Web, empleando las mejores prácticas de programación, tales como: la documentación del código fuente, el uso de patrones de diseño, y de técnicas de Interacción Humano-Computador para agilizar el proceso de reclutamiento y selección de personal en las organizaciones.
       - Realizar pruebas de validación y aceptación de la aplicación web.

## Arquitectura de la solución

La arquitectura seleccionada para el desarrollo de la aplicación web configurable es la de cliente servidor, donde el usuario accede por medio del cliente, es decir, el navegador web e inicia la comunicación con un servidor web por medio de la red, realizando una comunicación de tipo petición respuesta donde el cliente envía solicitudes al servidor, y el servidor se encargará de emitir una respuesta adecuada solicitado por dicho cliente.

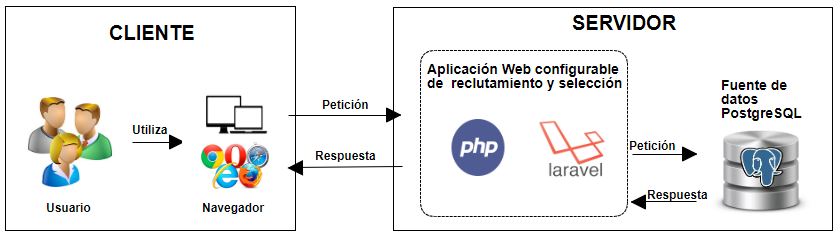
Las herramientas seleccionadas para el desarrollo de la aplicación web son el lenguaje PHP y el marco de trabajo Laravel. Estas tecnologías se seleccionaron por las siguientes razones:

* Experiencia previa en el manejo de PHP y el framework Laravel
* PHP es uno de los lenguajes más utilizados en el desarrollo web.
* PHP y Laravel son fáciles de utilizar y configurar según las necesidades específicas de la aplicación.
* PHP al ser un lenguaje que tiene muchos años, existe una gran cantidad de librerías o módulos que facilitan el desarrollo de la aplicación
* PHP es multiplataforma y compatible con diversidad bases de datos con su ORM desarrollado.
* 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 almacena la información de la aplicación web configurable se diseñó con el Sistema Manejador de Base de Datos PostgreSQL, debido a que, a diferencia de MySQL, Postgre tiene mayor capacidad de procesar un alto flujo de datos en aplicaciones grandes; al igual que Oracle, sin embargo, la licencia de Oracle tiene un costo adicional, por lo que se opta por la opción gratuita además de contar con un ORM que facilita la integración y el uso con el marco de trabajo Laravel.

En la base de datos se cargan los datos de los candidatos que van postularse en las distintas vacantes; los datos de las empresas y empleadores, donde cada cuenta empleador debe estar asociado a una empresa; a su vez se cargan datos de las vacantes, pool de cuestionarios a utilizar en las distintas vacantes de la empresa, y configuraciones y preferencias asociadas a cada empleador y candidato.

Finalmente la elaboración de las vistas de la aplicación web se realiza utilizando HTML5, CSS3, Jquery y Javascript, por lo que la arquitectura de la aplicación Web configurable quedaría reflejado de la siguiente manera:

**Figura 1.1 – Arquitectura de la Solución**

## Justificación de la solución

El objetivo prioritario de la presente investigación, no es documentar ni informar cuáles son los gastos que afronta una empresa durante los procesos de reclutamiento, sino proponer una solución eficiente y de mayor flexibilidad que las herramientas convencionales que se encuentran en la actualidad, y demostrar que dichos procesos pueden ser optimizados de manera que, las empresas puedan agilizar sus procedimientos de contratación, y que a su vez les permite incrementar las posibilidades de seleccionar el candidato idóneo para la empresa contratante, a modo de proporcionar un feedback eficiente a los candidatos que participan en el proceso de selección, de una forma cómoda y flexible, y que además, contribuya a mitigar inconvenientes. Una contratación errónea acarrea gastos significativos, pérdidas y consecuencias que no son sencillas de manejar, como emocionales, legales, morales, monetarias, entre otras; es por ello que cualquier aporte que contribuya a tomar mejores decisiones durante el proceso de reclutamiento, debe ser aprovechado.

Cabe resaltar que la contratación de personal es un proceso complejo donde existen una serie de prácticas que son de uso común, y que probablemente no sean las más eficientes, y de serlo, dichas prácticas pueden ser obsoletas y no adaptarse al dinamismo y a las necesidades del mercado actual.

El realizar el presente estudio y recopilar datos relacionados al reclutamiento, va a permitir analizar patrones o prácticas comunes que serán de utilidad para el desarrollar un sistema eficiente para el beneficio de las personas que sean involucradas, con el fin de dar flexibilidad a los candidatos en realizar el proceso de selección y, a su vez, el seguimiento de su estado durante todo el proceso desde la comodidad de su casa, sin restricciones de tiempo, lo que evita que el candidato deba solicitar permisos para retirarse de la oficina para realizar entrevistas con la incertidumbre de saber si dicha entrevista es beneficiosa o no.

# CAPÍTULO 2 - MARCO CONCEPTUAL

En este capítulo se trata en detalle el estado actual de las aplicaciones relacionadas con el proceso de selección de las distintas empresas, así como se abordan los conceptos relacionados con el desarrollo de aplicaciones web como su arquitectura, funcionamiento, tecnologías utilizadas y el sistema manejador de base de datos.

## Antecedentes

Es importante conocer el proceso de selección así como también los procedimientos que siguen cada una de las empresas al momento de realizar una búsqueda de personal.

El proceso de reclutamiento y selección de empresas consta de una serie de etapas que se aplica a cada candidato que desee aplicar para el cargo ofrecido. Inicialmente, la empresa debe agendar y manejar el tiempo de tal forma que puedan atender a la mayor cantidad de candidatos en un día determinado, sin embargo, muchas empresas no poseen una forma de trabajo eficiente que permita evaluar a una gran cantidad de candidatos en el menor tiempo posible, debido a que, generalmente, dichas evaluaciones son realizadas por una sola persona que dedica parte de su tiempo a aplicar el proceso de selección, evaluación y contratación de a un candidato a la vez, lo que origina que el proceso sea lento, lo que puede ocasionar que los candidatos decidan retirarse del proceso afirma (Ernesto et al., 2012).

Por otro lado, las empresas realizan impresiones de cuestionarios y encuestas que son aplicados a cada candidato, y lo cual, les genera pérdidas económicas que acorde a la infografía de (RecruiterBox, 2015), los costos son relacionados a los impuestos por la publicación, beneficios y salarios para el reclutador, costos relacionados al transporte del postulado, costos de impresión (que incluye los cartuchos de tinta tanto negro como a color), gastos de energía eléctrica, en algunos casos los procesos de reclutamiento toman mucho tiempo durante el día generando gastos en refrigerios para el/los candidato(s), si el candidato decide abandonar el proceso de selección, o si no es seleccionado para continuar; y si el candidato es seleccionado, se pueden encontrar con el escenario de que dicho candidato ya aceptó una propuesta laboral de otra empresa, debido a la lentitud del proceso y a la falta de feedback a lo largo del mencionado. Además, la lentitud del proceso y falta de feedback oportuno, deja al candidato en incertidumbre, ya que desconoce, cuál es la posibilidad que tiene, de ser seleccionado, y poder incursionar en el vacante lo que causa que la empresa pierda candidatos potenciales durante el proceso.

Para el candidato, un proceso de selección mal estructurada conlleva a múltiples pérdidas económicas relacionado al costo del transporte hacia las instalaciones físicas de la empresa u organización, impresión de múltiples documentos solicitados de forma física como lo son los currículums vitae, certificado de cursos, títulos escolares, donde, al momento de que el candidato obtenga información del salario y beneficios ofrecidos por la organización, puede arrepentirse de participar en el proceso, información que podían reflejarse en la publicación de la vacante por lo que los costos mencionados se pierden y la empresa no tiene que responsabilizarse por ello, ocasionando que el candidato se retire de la instalación insatisfecho de los resultados disminuyendo la probabilidad de que aplique nuevamente o recomiende a otros posibles candidatos para la empresa. Adicional a lo mencionado se suma el costo en tiempo que requieren los candidatos para poder desplazarse a las instalaciones de la organización y poder postularse en ella, tiempo que se pueden asumir en horas extras en oficina, oportunidades perdidas, entre otras.

Por otro lado, muchos candidatos se sienten incómodos con los diversos filtros que aplica una empresa para seleccionar, a lo que consideran como un candidato ideal, debido a la necesidad de tener que solicitar permisos laborales para asistir presencialmente a cada una de las etapas del proceso de reclutamiento y selección. Dichas causas originan que un candidato no desee continuar con un proceso de solicitud de empleo en una empresa, por considerarlos burocrático, lento, costoso, molesto, entre otras razones.

En cualquiera de los casos mencionados, la empresa reclutadora y el potencial candidato han perdido tiempo o dinero en dicho proceso, originando que además se generen experiencias desagradables y que alguna de las 2 partes no tenga ganas de volver a participar en algún otro proceso de selección con dichas empresa o candidato, y en el caso de las empresas, que generen referencias negativas hacia otros potenciales candidatos, y que ante el llamado de una empresa determinada los candidatos opten por no responderle.

En Venezuela y en otras partes del mundo se han realizado diversas aplicaciones web relacionadas con la gestión de contrataciones y filtrado de personas por capacidades profesionales. Por ejemplo, en Venezuela se cuenta con portales web como Bumeran, compuTrabajo, Beebee, etc; que son plataformas que ayudan a los usuarios a encontrar empleo, y tienen bastante características en común, pero en general se destacan por tener procedimientos cerrados e invariantes, que no ofrecen flexibilidad, ya que no permiten configurar y/o generar cuestionarios o evaluaciones a realizar durante el procedimiento en cuanto a las empresas, y para el postulado existen pocos estados durante todo el proceso de reclutamiento generando incertidumbre de su posición durante el proceso y el hecho de realizar una postulación a una vacante pueden resultar tediosos según (Ernesto, Gabriela, Leonor & Ursula, 2012) por la cantidad requerida de información como lo son la subida de curriculums digitales, certificaciones, títulos educativos, salario esperado (en el caso de que la empresa no lo refleje en la publicación), preferencias, entre otros, el cual podrían ser configurados por el postulante para autocompletar al momento de postularse a una vacante. A continuación se analizan las aplicaciones Web existentes junto a sus limitaciones:

**2.1.1 Bumeran**

Constituye una opción para la búsqueda de empleos. Tiene una gran cantidad de currículums vitae en línea, y ofertas de trabajo por cargo o región (entre otros filtros).

Los postulantes cargan sus currículums vitae, y las empresas utilizan ese repositorio para la búsqueda de candidatos potenciales. Entre las funcionalidades del portal, destacan la clasificación de candidatos, pre-filtrado de currículums, notificaciones instantáneas, según (Bumeran, 2018).

Adicionalmente, cuando una empresa se registra, debe pagar una suscripción para poder utilizar los servicios del portal, y cuyo repositorio de trabajo genérico, se basa en el siguiente principio: “El portal de empleo hace la primera parte del proceso de selección, que termina con varias entrevistas personales, donde el reclutador elige a una persona entre varias. Si la persona no resultó la indicada, no significa que el portal haya sido poco eficiente”, plantea (Navarro, 2014), y es por ello que no ofrecen mecanismos adicionales para la preselección de candidatos, tales como pruebas, o preguntas claras acerca de los requisitos de la vacante; y actúan principalmente como un portal de anuncios de empleo, y no como una herramienta para seleccionar candidatos.

De lo antes expuesto, se plantea que, parte de la propuesta del presente trabajo de investigación, se basa en que las empresas puedan crear y configurar preguntas de su interés, que ayuden a definir si el postulante es adecuado para iniciar el proceso de selección, en el cargo al cual intenta postularse. Dichos pasos, adicionales también proporcionan datos estadísticos para saber si las pruebas son adecuadas, o si los requisitos del puesto son excesivos, otorgándole a la empresa un feedback adicional de su proceso de selección. A continuación, en las Figuras X,X, se muestran algunas de las principales interfaces de Bumeran:



Figura X – Interfaz de Búsqueda de empleo (Bumeran, 2018)

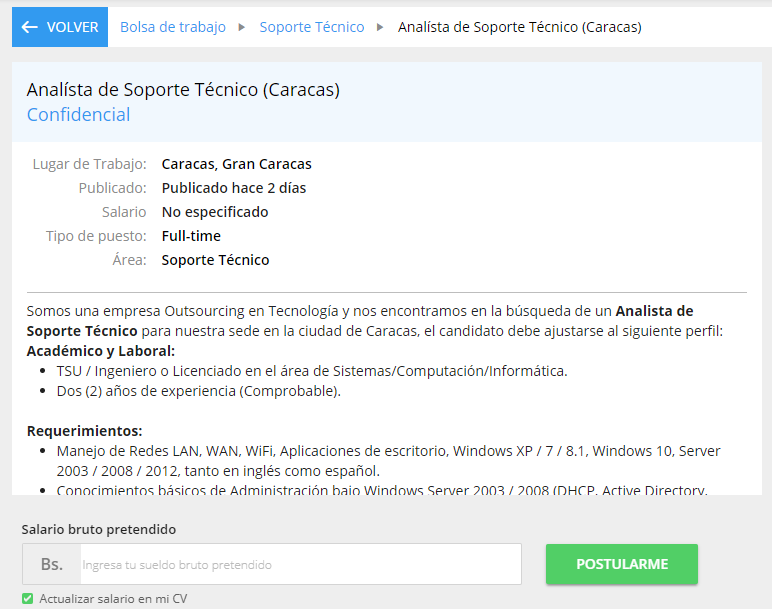


Figura X – Datos ingresados para publicación de una vacante (Bumeran, 2018).

**2.1.2 Computrabajo**

CompuTrabajo se denomina en su portal como “la mayor red de trabajo de habla hispana”, ya que cuenta con presencia y oficinas, en varios países de América Latina, según (DGNET, 2018).

En dicho portal, los postulantes pueden registrarse y buscar empleo, por localización, cargos profesionales y categorías; y se encarga de enviar correos con ofertas a las personas registradas, según (Maura, 2014). Además, los postulantes registran un currículum que luego pueden reutilizar permitiendo descargarlo en formato pdf, facilitando la importación a otra aplicación que requiera el currículum, y utilizando formularios incrementales, permiten añadir o remover segmentos de dicho currículum, de forma sencilla.

Posee en su base de datos, más de 12.000 ofertas de trabajo, solo en Venezuela, y el registro en la Web, es completamente gratuito, plantea (Maura, 2014). Dicha fuente, describe ciertas desventajas, que fueron mencionadas a continuación:

• Al momento de consultar a los candidatos que se postulan para una vacante, dicha información, únicamente se puede visualizar a través de correo electrónico, y no provee un mecanismo en donde se visualice el listado de los postulantes, junto a sus hojas de vida o currículums.

• Cada vez que un candidato se postula, a un nuevo puesto o vacante, el postulante debe reenviar el currículum manualmente.

• El portal Web no tienen una organización eficiente, para realizar el envío de la hoja de vida para una vacante en particular.

Adicionalmente, (DGNET, 2018, Prestación Servicios) describe que el portal se compone de 2 secciones: Una sección que ofrece servicios para los candidatos, y una sección para las organizaciones. En particular, los servicios más destacados, y que los diferencian de Bumeran, se describen a continuación:

Registro de hoja de vida o currículum, el cual se realiza a través de un formulario que se encuentra definido en el portal, y que fue dividido, en 4 pasos. Entre los datos que deben ser registrados se encuentran: Datos de identificación y contacto, experiencia laboral y estudios realizados, plantea (DGNET, 2018, Prestación Servicios. Marco Legal).

Además, los postulantes pueden anunciarse a las organizaciones, y los candidatos se pueden postular a diferentes vacantes mediante un botón, en cuyo caso, el candidato permite que la empresa pueda visualizar los datos registrados de la hoja de vida que ha ingresado en el portal.

En cuanto a los empleadores, proporcionan la opción de poder remitir el perfil de los candidatos, a otras empresas o vacantes creadas por la misma empresa, si los candidatos configuran dicha característica, en su perfil profesional, según (GCNET, 2018, Prestación de servicios).

A continuación, en las Figuras X,X, se muestran algunas de las principales interfaces de CompuTrabajo:

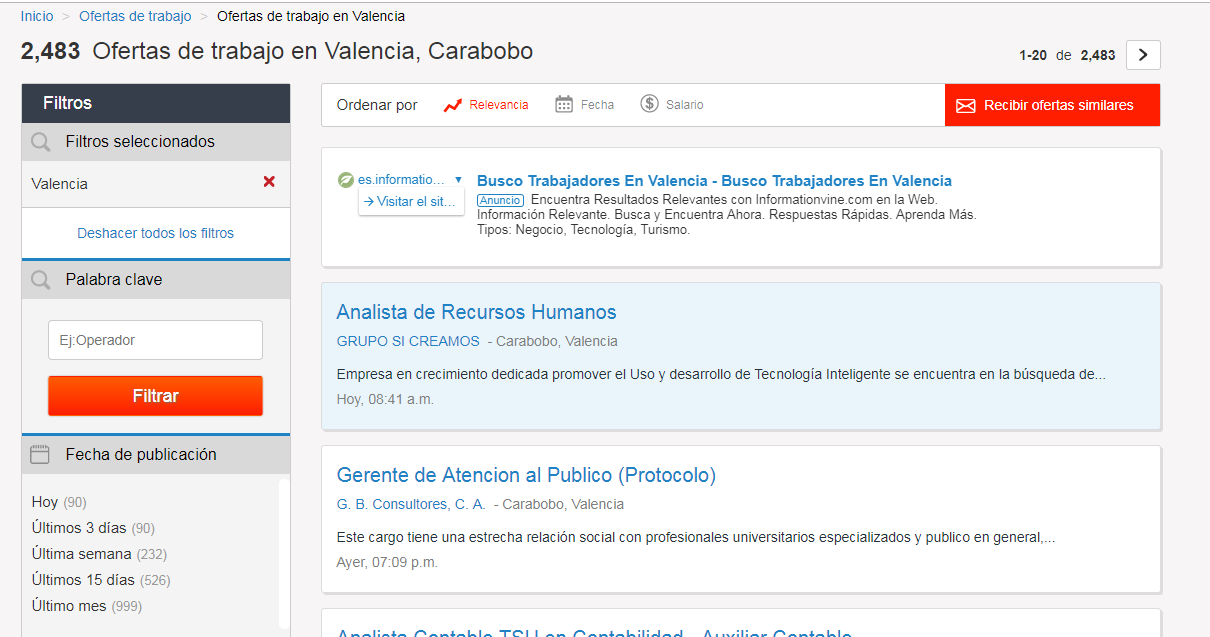


Figura X – Interfaz de Búsqueda de empleo **(REF)**

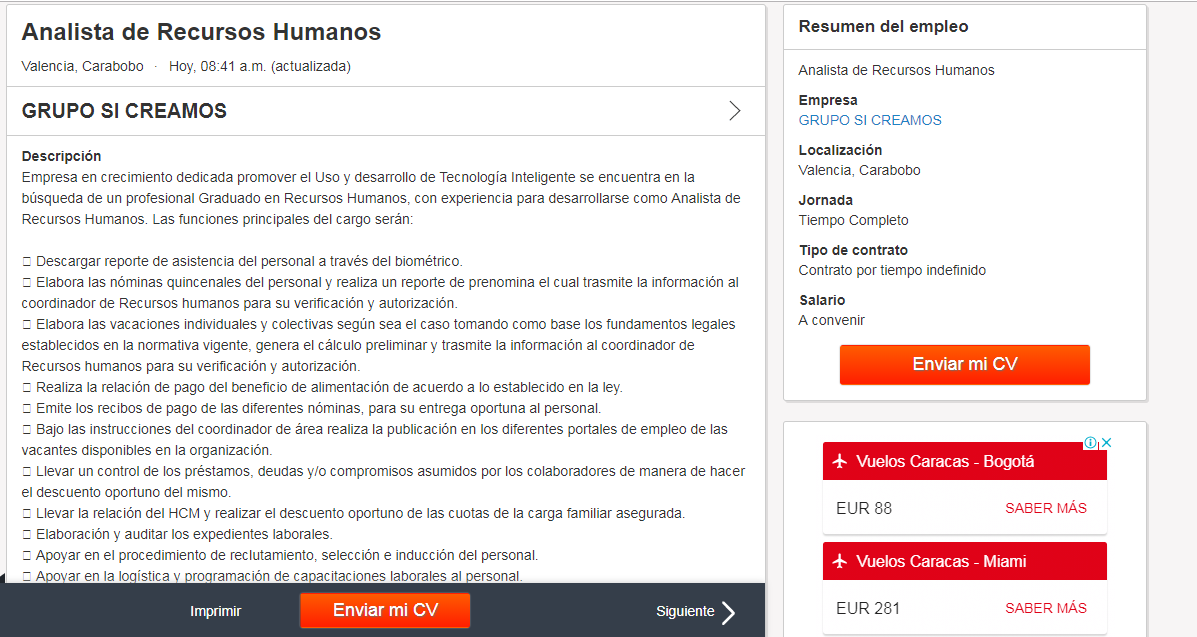


Figura X – Datos ingresados para publicación de una vacante **(REF).**

**2.1.3 Empléate**

Es un portal que funciona como un repositorio de 2.000.000 de perfiles de usuarios, y 80.000 empresas suscritas, contando con miles de ofertas de empleo que se actualizan durante todo el día. Al registrarse en la plataforma, los candidatos pueden realizar búsquedas de ofertas de empleo, por palabras clave, ciudad, cargo, estado, e incluso categorías. Además de ser un repositorio de ofertas laborales, el portal cuenta con otros recursos, como las capacitaciones a través de cursos en línea (Empleate, 2018).

A pesar de que es un servicio pago para las empresas, sus servicios fundamentales son variados, comenta (Empleate, 2018), y tienen varios servicios como asesorías, sistema de controles adicionales, el poder controlar el tiempo de publicación de las ofertas laborales, ofrecen asesoría en la construcción de los perfiles necesarios para las vacantes requeridas, enseñan diferentes medios de reclutamiento, tercerizan entrevistas presenciales con candidatos, revisan referencias personales, elaboran informes de evaluación, y preseleccionan perfiles.

A continuación, en las Figuras X,X, se muestran algunas de las principales interfaces de Empléate:



Figura X – Interfaz de Búsqueda de empleo (Empléate, 2018).

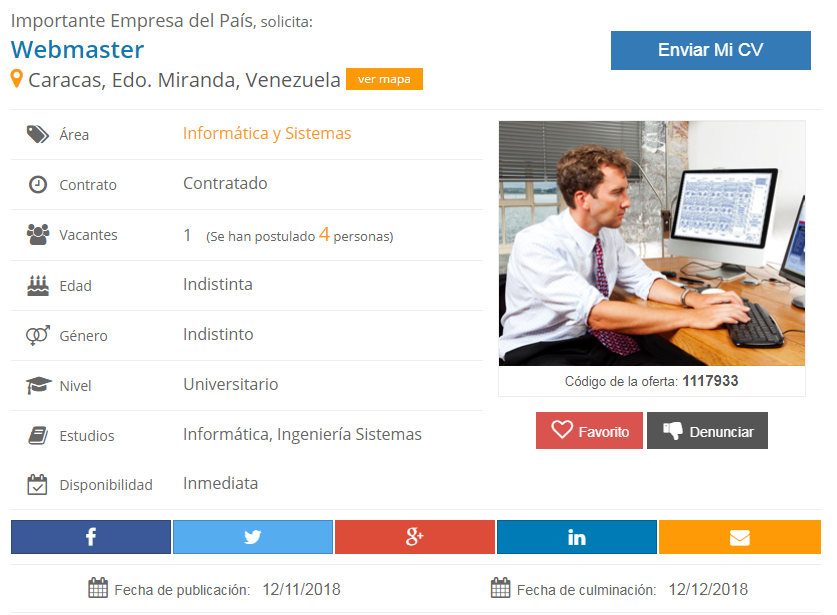


Figura X – Datos ingresados para publicación de una vacante (Empléate, 2018).

## Inferencias sobre los antecedentes

Los portales descritos, tienen diferentes maneras de solicitar datos acerca de los usuarios que se registran en el portal, y que posteriormente puedan ser utilizados como un currículum vitae dentro de los mencionados, y lo cual, es de vital importancia, ya que proporciona datos que permiten evaluar los diferentes perfiles profesionales de las personas que utilizan la plataforma, y que además, es una característica esencial para el desarrollo de los sistemas de búsqueda de personal, ya que sus motores de búsqueda y filtrado, se basan en dicha información principalmente.

Adicionalmente, tienen la particularidad de ser heterogéneos en su propósito, y su uso; se suelen dividir o clasificar, según el rol con que el que se desee utilizar la plataforma, y en donde las empresas tienen diferentes mecanismos para describir sus necesidades, incluyendo las necesidades de personal que requieren para ocupar una o más vacantes en particular.

Dichos mecanismos, son de vital importancia en diversos portales, y que inclusive, para la elaboración de la propuesta del presente trabajo de investigación, dichas características serían consideradas, para hacer un uso más eficiente de la información; y adicionalmente, tomar en consideración, la usabilidad que proporcionan a los usuarios, para hacer que las tareas que ofrecería la aplicación, sean lo más cómodas e intuitivas posibles, y así estimular el crecimiento de los datos; y en donde se pudiera agilizar o filtrar la cantidad de aspirantes que aplican a una vacante determinada, a través de ciertas preguntas que se le vayan realizando al candidato, acerca de la vacante a la que se ha postulado; y que dichas preguntas, puedan ser configurables por la empresa que publicó la oferta de trabajo.

En relación a algunos mecanismos como las búsquedas, filtrado de información, y publicación de propuestas laborales; dichos mecanismos, suelen ser comunes en los portales de empleo, y de vital importancia para su funcionamiento; y que es difícil describirlos como puntos fuertes. Sin embargo, lo realmente notable, es el uso y el análisis que hacen de los datos proporcionados por los usuarios, para obtener información de los mencionados.

Por otro lado, es un problema común para las empresas, que no se pueda disponer de una manera eficaz de determinar, si los candidatos que se han postulado a una vacante en particular, o si los candidatos que sean de interés para la empresa, han cambiado su estado laboral, o de estudios.

Según (Maura, 2014), el problema descrito es difícil automatizar por cualquier portal, ya que dicha información, requiere de la intervención del usuario que se registra en el portal, para poder crear su perfil profesional. Sin embargo, mecanismos como la caducidad, o el registro de los últimos cambios a los perfiles profesionales de los candidatos, podrían ayudar con el criterio de selección de los mencionados; ya que deja vislumbrar lo reciente, de la información otorgada por el perfil profesional del candidato.

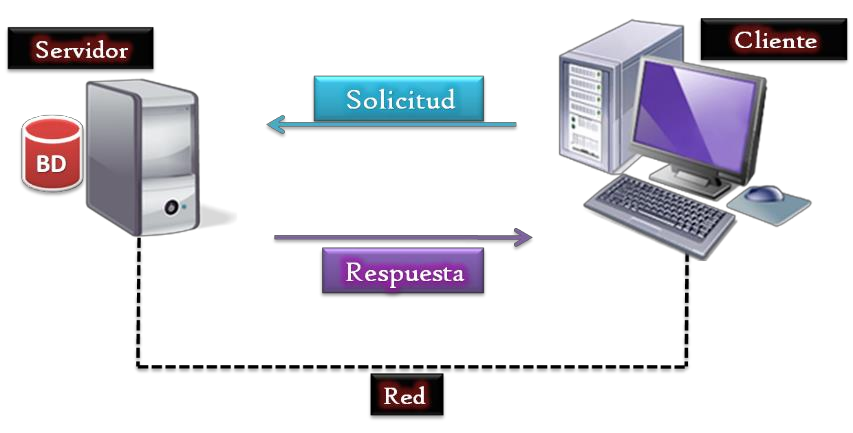
## Arquitectura cliente-servidor

La arquitectura cliente-servidor es una arquitectura de red que consiste básicamente, en un cliente en el cual realiza una petición al servidor (programa) toma esa petición y la procesa con el fin de dar una respuesta. Para ello, el servidor tiene asignado un puerto conocido por el Sistema Operativo por el cual va a recibir las peticiones del cliente, donde espera de manera pasiva las peticiones, en cambio, el cliente utiliza cualquier puerto no utilizado por el Sistema Operativo para realizar la petición **(REF).**

El cliente es un programa ejecutable el cual realiza las conexiones con el servidor, enviando a través de ella las peticiones y pasa a un estado de espera hasta recibir una respuesta, sin embargo, el tiempo de conexión es finita **(REF)**, por lo que una vez que recibe una respuesta o sobrepasa el tiempo de espera de la conexión, la misma se cierra y finaliza el trabajo.

El servidor es un programa que ofrece un servicio respondiendo peticiones del cliente y generando una respuesta al solicitante. Los servidores pueden ejecutar distintos tipos de tareas según la solicitud, pudiendo ser sencillas como recibir la hora del servidor o complejas como es el procesamiento de datos enviados junto a la solicitud del cliente (como registro en el servidor, generar datos, entre otros) **(REF** [**https://www.ecured.cu/Arquitectura\_Cliente\_Servidor**](https://www.ecured.cu/Arquitectura_Cliente_Servidor)**).**

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



**Figura X - Arquitectura cliente-servidor.**

### 2.3.1 Ventajas

Las principales ventajas que ofrece la arquitectura cliente servidor se detallan a continuación (REF <http://somebooks.es/ventajas-e-inconvenientes-de-la-arquitectura-clienteservidor/>):

* Debido al poco valor que genera el cliente en el esquema del servidor, las necesidades de configuración y administración por parte del cliente son menores.
* Es una arquitectura escalable, por lo que añadir recursos tanto a un cliente como a un servidor como memoria física, procesador o nodos de red, no genera problemas al resto de la arquitectura.
* Al mantener un servicio de autentificación centralizado, la posibilidad de un acceso no permitido aplicando las políticas de seguridad recomendadas son reducidas considerablemente.
* Al estar distribuidas ciertas funcionalidades, el mantenimiento de un servidor no ocasiona (o de forma mínima) afectaciones al usuario al momento realizar la comunicación con el servidor.

### 2.3.1 Desventajas

Las principales desventajas que se encuentra al momento de utilizar la arquitectura cliente servidor se detallan a continuación (REF <http://somebooks.es/ventajas-e-inconvenientes-de-la-arquitectura-clienteservidor/>):

* Los costos de mantenimiento suelen ser elevados por el alto nivel de conocimiento técnico requerido para el manejo del mismo de la forma más óptima.
* Una gran cantidad de clientes puede generar congestión en el servidor al momento de realizar las peticiones ocasionando retrasos en las respuestas del servidor, timeout, error de transmisión, entre otros.
* En referencia a la desventaja anterior, se tiene la alta dependencia del servidor para que el sistema funciona, es decir, cuando ocurra una falla crítica en el servidor en el cual no pueda realizar el procesamiento de información, no se podrá continuar con el uso de la aplicación debido a que no existe el servidor que responda a las peticiones de los clientes (la opción más óptima es contar con servidores backup en caso de fallas).
* Debido al esfuerzo que debe realizar el servidor para responder una inmensa cantidad de peticiones al día, es imprescindible contar con el hardware y software correcto para el funcionamiento óptimo del servidor.

## Aplicación web

Una aplicación web es un tipo de aplicación cliente/servidor donde el cliente, que vendría siendo el navegador o browser web, y el servidor web utilizado para realizar tareas a través de internet utilizando los protocolos de comunicación estandarizados (HTTP/HTTPS). (REF: <https://www.maxcdn.com/one/visual-glossary/web-application/>)

### 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.

Las aplicaciones Web requieren un servidor Web quien se encarga de manejar y procesar las solicitudes del cliente, un servidor de aplicación donde se ejecutan las tareas y, generalmente, se guarda la información por medio de una Base de Datos. Adicionalmente se encuentra codificadas en lenguajes como JavaScript para las interacciones del usuario y HTML para la capa de presentación entendibles por el navegador ya que el mismo es quien tiene la capacidad de ejecutar las aplicaciones Web. El flujo se puede representar de la siguiente manera (REF: <https://www.maxcdn.com/one/visual-glossary/web-application/>):

1.-El usuario desencadena una petición al servidor Web por medio de internet a través del navegador Web o la aplicación de usuario.

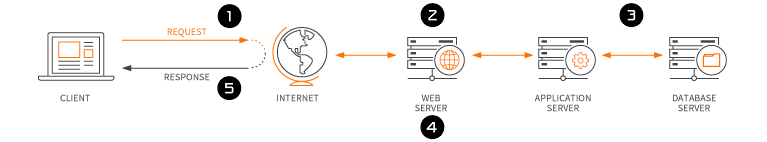
2.-El servidor Web escucha la petición del cliente y lo transfiere al servidor de la aplicación Web capaz de procesar la solicitud.

3.-El servidor de la aplicación Web procesa la tarea asociada a la petición, ya sea generando consultas en base de datos, procesar información, creación de algún artefacto, y finalmente genera una respuesta o resultado de la petición.

4.-El servidor de la aplicación Web envía los resultados al servidor Web con la información o datos procesados según la petición.

5.-El servidor Web envía la respuesta al cliente quien genero la petición y ésta aparece en la vista del cliente a través del navegador o aplicación de usuario.

A continuación, en la Figura 2.15 se muestra de manera gráfica el proceso de petición de una servidor web.



**Figura X – Flujo de una aplicación Web.** (REF: <https://www.maxcdn.com/one/visual-glossary/web-application/>):

### Ventajas

Las aplicaciones Web cuentan con una serie de ventajas detalladas a continuación (REF: https://www.yeeply.com/blog/ventajas-y-desventajas-de-una-web-app/):

* Se ejecutan dentro del propio navegador Web del dispositivo por medio de una dirección URL.
* Las aplicaciones Web, al ser ejecutadas por medio del navegador, están disponibles para cualquier dispositivo bajo cualquier sistema operativo que se encuentre conectado a la red, lo cual otorga una alta compatibilidad.
* Para el usuario, no es necesario realizar instalaciones adicionales ni realizar actualizaciones de la aplicación debido a que son aplicadas del lado del servidor, lo que implica que el usuario final siempre tendrá la última versión liberada de la aplicación.
* Utilizar una aplicación Web reduce en gran medida el uso de recursos locales del cliente ya que todas las tareas y procedimientos son realizados en el servidor Web.

### Desventajas

Las aplicaciones Web cuentan con una serie de desventajas detalladas a continuación (REF: https://www.yeeply.com/blog/ventajas-y-desventajas-de-una-web-app/):

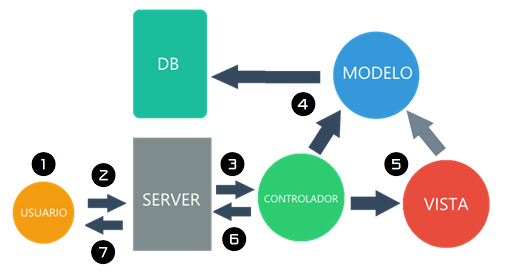
* El navegador toma un rol importante en el funcionamiento de la aplicación Web, por lo que la misma debe estar correctamente configurada para su funcionamiento, ya sea la versión más actualizada de flash, cookies, entre otros.
* En cuanto a seguridad, debido a que las aplicaciones Web se encuentran en un dominio público, pueden ser objetivo de un ataque cibernético con el fin de obtener información confidencial de forma no autorizada o saturar el servidor Web con el fin de que su funcionamiento sea detenido, por lo que es imprescindible contar con protocolos de seguridad para el resguardo de la información.
* Es absolutamente necesario la conexión a Internet para poder utilizar aplicaciones Web.
* Una gran flujo de cliente enviando peticiones de forma concurrente puede causar una sobrecarga en el servidor Web lo cual disminuye de manera considerable los tiempos de respuesta o incluso detiene su funcionamiento en su totalidad.

### Patrón de Modelo Vista Controlador (MVC)

El patrón Modelo Vista Controlador (MVC) es un patrón de diseño en el que especifica que la aplicación consta de un modelo de datos, presentación de los datos en una vista e información de control, el cual es frecuentemente utilizado para el desarrollo de aplicaciones Web ya que detallan de forma concreta los componentes de la aplicación en tres capas bien definidas y que se explican a continuación (REF:http://www.wcreator.com.ve/noticias/item/32-que-es-mvc-y-por-que-es-tan-usado-en-el-desarrollo-web):

* **Modelo:** En esta capa que se encarga del manejo de los datos, es decir, es la capa que contiene los mecanismos para acceder a la información y actualizar el estado del mismo así como la administración para la recuperación de los datos y el mantenimiento de la consistencia e integridad. En dichas capas se suelen implementar los ORM (Object Relational Mapping) el cual permite por medio de un framework y con un cierto nivel de abstracción comunicarse con la base de datos SQL.
* **Vista:** En esta capa se encuentra el código el cual presenta los datos proporcionados por el modelo al usuario a través de una interfaz por el cual el usuario puede interactuar.
* **Controlador:** Esta capa enlaza el modelo con la vista, donde envía al modelo un conjunto de comandos por el cual actualiza el estado de la información para posteriormente enviar otro conjunto de comandos a la vista para cambiar su presentación con la nueva información. Es importante destacar que el controlador no es quien se encarga de manipular la data ni generar las salidas
  + - 1. **Flujo de Control del MVC**

El patrón MVC sigue un flujo específico para el procesamiento de las interacciones del usuario con el servidor, en la siguiente imagen se muestra de forma gráfica el flujo el cual será detallado a continuación: (REF: <http://www.wcreator.com.ve/noticias/item/32-que-es-mvc-y-por-que-es-tan-usado-en-el-desarrollo-web>)



**Figura X – Flujo del modelo MVC.** (REF: <http://www.wcreator.com.ve/noticias/item/32-que-es-mvc-y-por-que-es-tan-usado-en-el-desarrollo-web>).

Donde:

1. El usuario ingresa a la página por medio de un URL.
2. El usuario interactúa con la aplicación y envía una petición al servidor.
3. El servidor hace un llamado al controlador.
4. El controlador hace un llamado al modelo necesario el cual atiende la operación y realiza las operaciones de datos correspondientes, donde finalmente retorna el resultado.
5. El controlador hace un llamado a la vista donde envía a su vez el resultado de la petición devuelto por el modelo y la vista pasa a presentar los datos.
6. El controlador retorna la vista al servidor.
7. Finalmente el servidor muestra el resultado al cliente.
   * + 1. **Ventajas del MVC**

El patrón MVC cuenta con las siguiente ventajas (REF☺):

* + Al estar separado cada componente se evita combinar código de servidor con el código de presentación.
  + Permite la reutilización de código ya que los modelos pueden ser llamado por uno o más controladores.
  + Se automatiza la actualización de las vistas, es decir, no tienen que ser cambiadas por el programador para que su contenido esté actualizado.
  + El patrón está enfocado en separar responsabilidades.

## Sistema Manejador de Contenidos

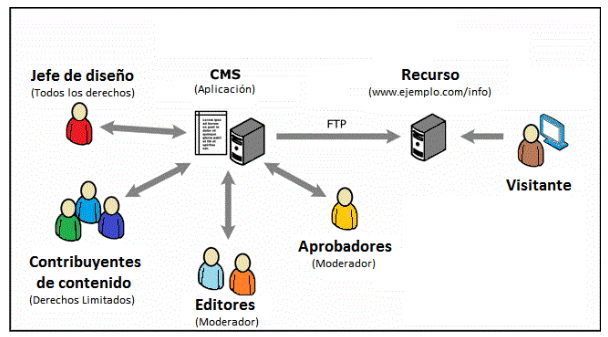
Un sistema manejador de contenido o CMS (Content Management System) es un programa o aplicación alojado en línea que permite administrar, editar, crear y eliminar contenidos en un sitio web, permitiendo al usuario controlar el contenido que se muestra. Los datos o contenidos mostrados en el sitio web se alojan en una o varias bases de datos. Los sistemas manejadores de contenido permiten proveer a usuarios con distintos roles y permisos de cualquier nivel para realizar modificaciones de contenido, datos o información que se muestra dando la ventaja de adaptarse a los nuevos formatos o tecnología de forma eficiente (Alonso, 2018).

### Organización y Estructura de un CMS

La información se encuentra organizadas en documentos y archivos, y a su vez, cada documento y archivo se encuentra dividido en uno o varios esquemas, lo que permite clasificar la información o contenido que puede ser recuperado con facilidad para aplicarse en una plantilla o vista. La estructura de los datos permite que se puedan desarrollar procesos automatizados de mantenimiento de forma periódica o ante eventos que desencadenan el proceso (Alonso 2018). Todo el contenido se almacena en un formato nativo que permite a los desarrolladores utilizar herramientas de software para para la creación del contenido y mantenimiento del mismo. Cuando se recupera la información, el proceso de ensamblaje agrupa el contenido del documento o archivo de acuerdo al esquema asociado y lo formatea de acuerdo a la plantilla o vista en la que se muestre.

### 2.5.2 Flujo y funcionamiento de un CMS

Los CMS se basan en el uso de flujos de trabajos para la administración de la información. En general todos los CMS incluyen flujos de trabajo por defecto, sin embargo, es posible realizar los flujos de forma personalizada de acuerdo a las necesidades de la organización. En todo CMS existe un Flujo de trabajo fundamental para la administración del contenido, como se muestra en la figura X:



**Figura X –** Flujo de trabajo, funcionamiento de un CMS. (EcuRed, s.f)

La forma en que trabajo un sistema manejador de contenido puede variar según el número de usuarios que participen en la administración del mismo, en general, el flujo de trabajo de un CMS se describen de la siguiente manera según (EcuRed, s.f):

El jefe del diseño es quien se encarga de realizar el diseño, estructura, funcionalidades y accesibilidad creando un conjunto de plantillas donde se incluyen campos y regiones editables, características personalizables de navegación y de diseño dentro del sistema manejador de contenido. Debido a lo mencionado, el jefe de diseño tiene todos los derechos para la creación, edición y eliminación de los contenidos.

Los contribuyentes de contenido son quienes se encargan de tomar las plantillas generadas por el jefe de diseño e incluyen el contenido o información en ellas para ser mostrado hacia el usuario. Debido a lo mencionado, los contribuyentes tienen derechos limitados en los que se incluye la edición y eliminación de elementos contenidos en las plantillas. Los editores son quienes se encargan de hacer revisiones a al conjunto de contenidos generados por contribuyentes y realizan ediciones tanto estructural como de diseño en caso de ser necesario. Debido a lo mencionado, los editores tienen únicamente derechos de moderación para realizar la edición de los elementos contenidos en las plantillas generadas por los contribuyentes.

Los aprobadores son quienes se encargan de realizar la última revisión del contenido y, por medio de los lineamientos dictados por la empresa, se evalúa si dicho contenido es aceptable o no para mostrarse en el sitio web. Debido a lo mencionado, los aprobadores tienen derechos de moderación para aceptar o rechazar el contenido generado en las fases anteriores. Una vez que el contenido es diseñado, editado y revisado, cualquier usuario podrá acceder al recurso por medio del URL.

## Framework de Desarrollo Web

Un framework es una estructura de software que se componen de librerías y componentes personalizables para facilitar el desarrollo de una aplicación o componente de software (Gutiérrez, 2008). Análogamente, un framework de desarrollo Web es un tipo de framework cuyos componentes apoyan al desarrollo de sitios, aplicaciones y servicios webs dinámicos por medio de librerías que ofrecen funcionalidades estándares como la conexión con base de datos y mecanismos para facilitar la integración con otras herramientas para implementar las capas de negocio y presentación (Gutiérrez, 2008). Todo framework incluye un conjunto de componentes que facilitan el desarrollo de características y funcionalidades a la aplicación web, detalladas a continuación.

### 2.6.1 Librerías

Las librerías son una colección de recursos utilizados para el desarrollo de aplicaciones donde se incluye configuración de datos, documentación, código prescrito y rutinas, clases o tipos de datos (Woodhouse, 2015).

### 2.6.2 APIs (Application Program Interface)

Un API o interfaz de programación de aplicaciones es un conjunto de rutinas, funciones y procedimientos útiles para conectar dos software entre sí para realizar la comunicación con la fuente de datos (back-end) con los datos a mostrar (front-end) ya sea en formato XML o JSON (Woodhouse, 2015).

### 2.6.3 Scaffolding

Scaffolding es una técnica de programación soportado por los frameworks MVC (Modelo, Vista, Controlador) donde se especifica cómo y cuándo se tiene acceso a la fuente de datos (Woodhouse, 2015).

### 2.6.4 AJAX

La mayoría de los frameworks de desarrollo web incorporan la tecnología Asynchronous JavaScript And XML, mejor conocido como AJAX, el cual permite a los sitios web actualizarse de forma asíncrona realizando un intercambio de datos con el servidor sin la necesidad de recargar la página (Woodhouse, 2015). Un ejemplo de una llamada AJAX se puede observar en la siguiente figura:

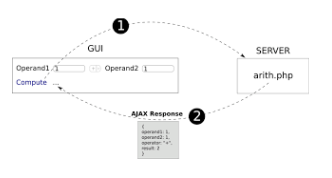


Figura 4 - Ejemplo de una llamada AJAX (Woodhouse, 2015).

### 2.6.5 Caching

El proceso de caching consiste en el almacenamiento de datos en la memoria caché, la memoria cache es un área de almacenamiento temporal que guarda archivos cargados de sitios webs de forma que al momento de abrir el mismo sitio nuevamente, carguen de forma más rápida incrementando el rendimiento (Woodhouse, 2015).

### 2.6.6 Seguridad

Todo framework de desarrollo web provee mecanismos de seguridad, estos mecanismos buscan resguardar la información de personajes que no tengan acceso a ella, como información privada de los usuarios registrados, de la aplicación u organización en general (Woodhouse, 2015).

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:

## Tecnologías del lado del cliente

Las tecnologías del lado del cliente son ejecutadas e interpretadas del lado del cliente en una aplicación web, es decir, en el navegador web. En muchos casos el buen funcionamiento de las tecnologías va a depender del tipo de navegador web y de la versión de cada uno de ellos. Por lo general las tecnologías mencionadas a continuación son utilizadas para mostrar información, darle formato a dicha información, solicitar datos, etc. Las tecnologías a utilizadas para el desarrollo del trabajo son HTML5, CSS3, Bootstrap, JavaScript y jQuery.

A continuación se procede a explicar brevemente cada una de ellas:



### HTML5 Lenguaje de Marcas de Hipertexto

El Lenguaje de Marcas de Hipertexto, mejor conocido como HTML (HiperText Markup Language), es un lenguaje de programación utilizado para el desarrollos de sitios web diseñado por el físico Tim Berners Lee en el año 1989, esta se publicó en un documento formal en el año 1991, convirtiéndose en un estándar para el desarrollo de páginas web en el año 1995 ya que ofrecía una estructura básica para la definición de como se muestra en contenido de una página web, como lo son las imágenes, formularios, textos generales, botones, entre otros.

HTML 5 establece una serie de nuevos elementos y atributos que reflejan el uso típico de los sitios web modernos. Incorpora nuevas etiquetas con mayor significado semántico, mejora la integración con las tecnologías de desarrollo web actual, permite desplegar objetos y animaciones en formato 3D (tridimensional), tiene un mejor soporte multimedia y ahorra la necesidad de tener que procesar medios de audio o video a través de complementos instalados en el navegador web.

### Hojas de estilo en cascada

HTML 5 establece una serie de nuevos elementos y atributos que reflejan el uso típico de los sitios web modernos. Incorpora nuevas etiquetas con mayor significado semántico, mejora la integración con las tecnologías de desarrollo web actual, permite desplegar objetos y animaciones en formato 3D (tridimensional), tiene un mejor soporte multimedia y ahorra la necesidad de tener que procesar medios de audio o video a través de complementos instalados en el navegador web.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.)

### 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.

### 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.).

### 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.

## 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

### 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 *scripting*, 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** | * 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 *frameworks*. * Compatible con diversas bases de datos. * Multiplataforma. * Código abierto. |
| **Desventajas** | * 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 *framework* 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. |

### 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** | * 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** | * Los vendedores de *software* 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.

## 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 | |

### 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.

1. Aceptar que los requerimientos cambien, incluso en etapas tardías del desarrollo. Los procesos ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
2. Entregar *software* funcional frecuentemente, entre dos (2) semanas y dos (2) meses, con preferencia al periodo de tiempo más corto posible.
3. Los responsables de negocio y los desarrolladores trabajan juntos de forma cotidiana durante todo el proyecto.
4. 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.
5. El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
6. El *software* funcionando es la medida principal de progreso.
7. Los procesos ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.
8. La atención continua a la excelencia técnica y al buen diseño mejora la agilidad.
9. La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
10. Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
11. 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).

* + - 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 3.1podemos 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.

* + - 1. **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.
        + E**quipo 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.

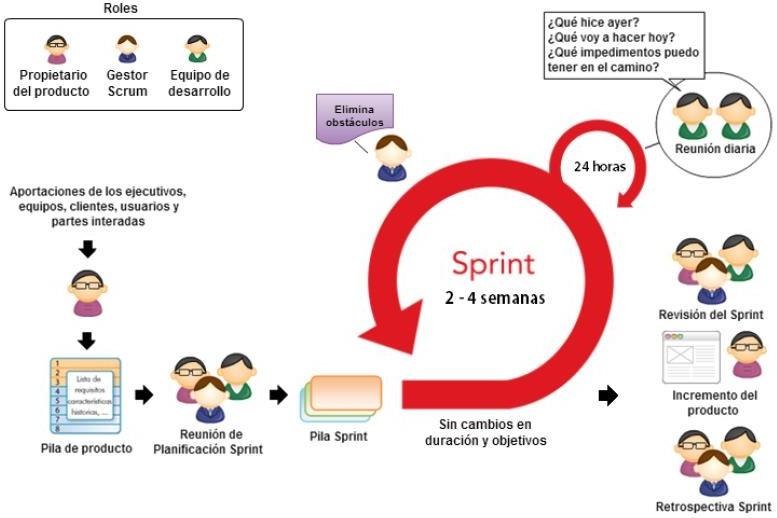
**Acciones y Artefactos**

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:
   * ¿Qué hice ayer?
   * ¿Qué voy a hacer hoy?
   * ¿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.

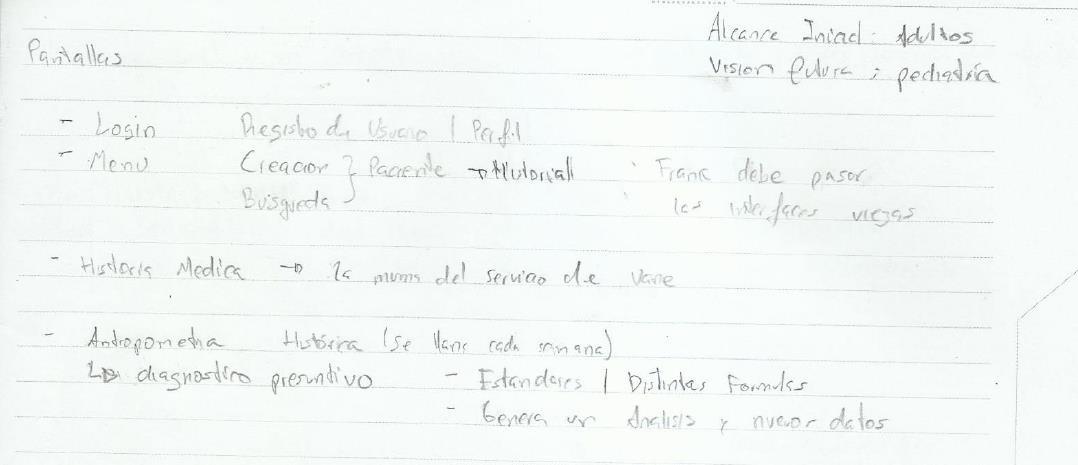
Ya que el objetivo era desarrollar una aplicación usable en un corto periodo de tiempo, el equipo decide realizar el desarrollo final de la aplicación utilizando la metodología Scrum, la cual permite que el equipo se adapte a los plazos de tiempo dados, adicionalmente fueron empleados algunos de los 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**

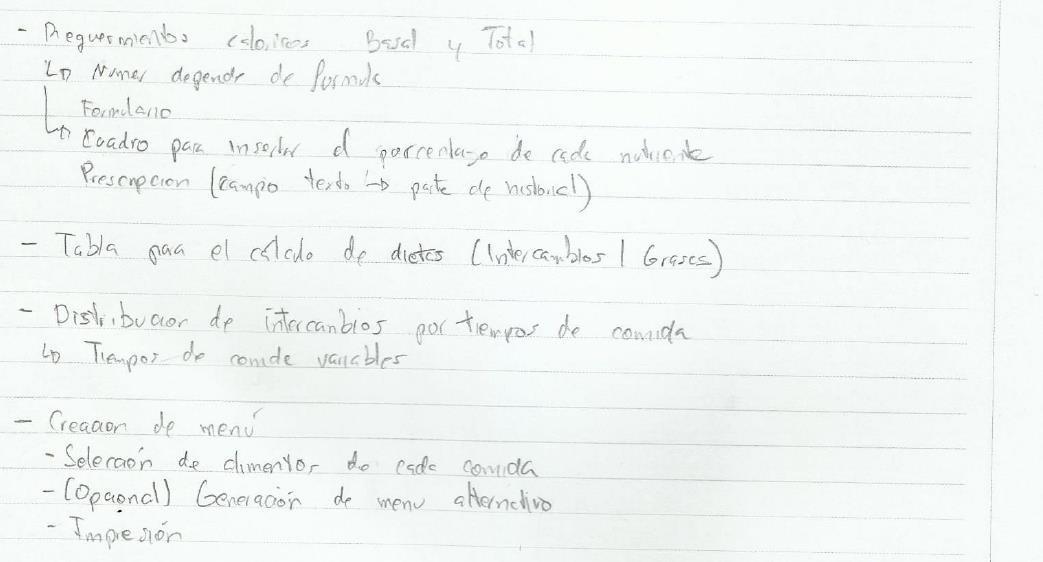
En este capítulo se describen aspectos y características relevantes de la aplicación y se detallan los pasos que se realizaron para el desarrollo de la aplicación web, siguiendo los lineamentos de la metodología scrum con apoyo de algunos de los artefactos utilizados en la metodología AgilUs.

* 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)**

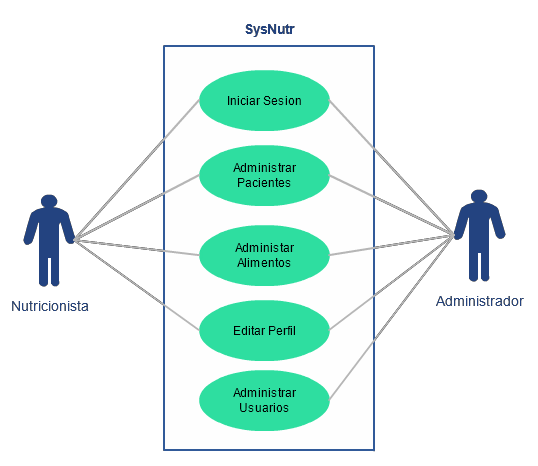
* 1. **Diagrama de Casos de Uso**

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.

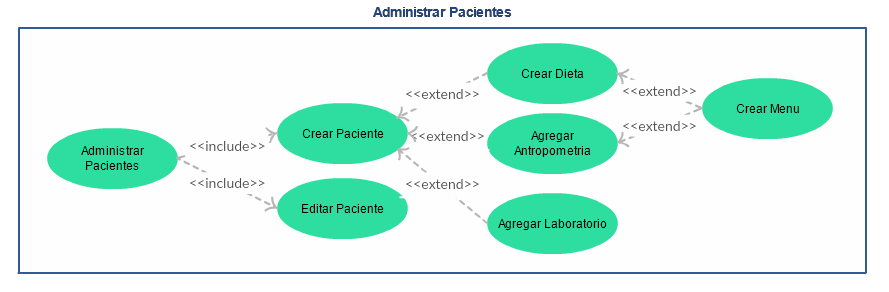
Para el esquema de la aplicación en general se tienen dos usuarios:

* + - **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.

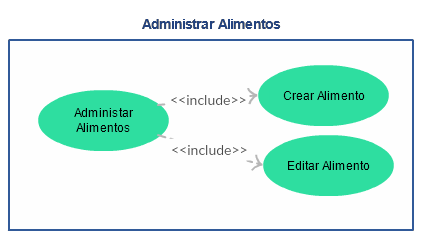
En las Figura 4.3, 4.4, 4.5 y 4.6 se pueden observar los diagramas de casos de uso Nivel 1 y Nivel 2 basado en los requerimientos del sistema desarrollado.



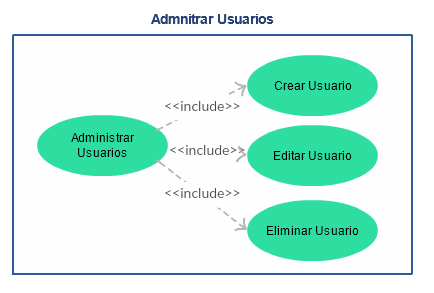
**Figura 4.3 – Modelo de Casos de Uso – Nivel 1**



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



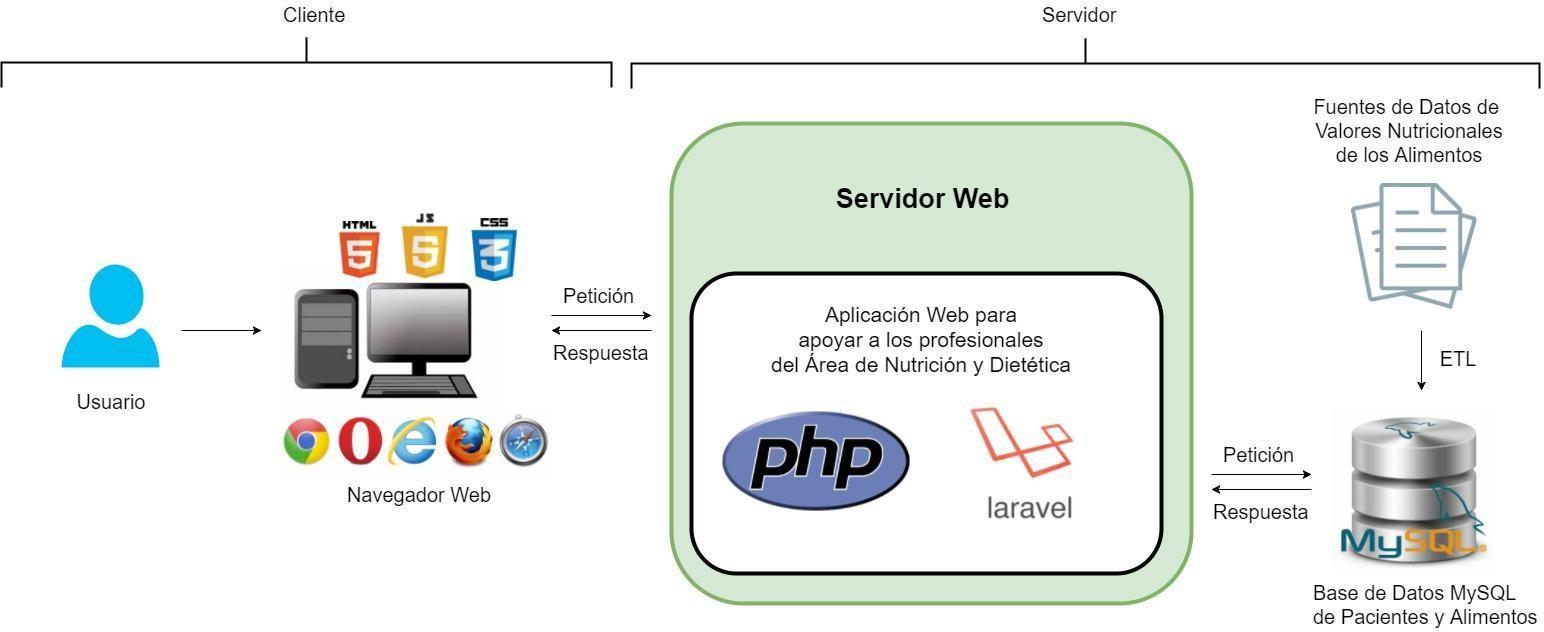
**Figura 4.5– Modelos de Casos de Uso – Nivel 2 – Administrar Alimento**



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

* 1. **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.



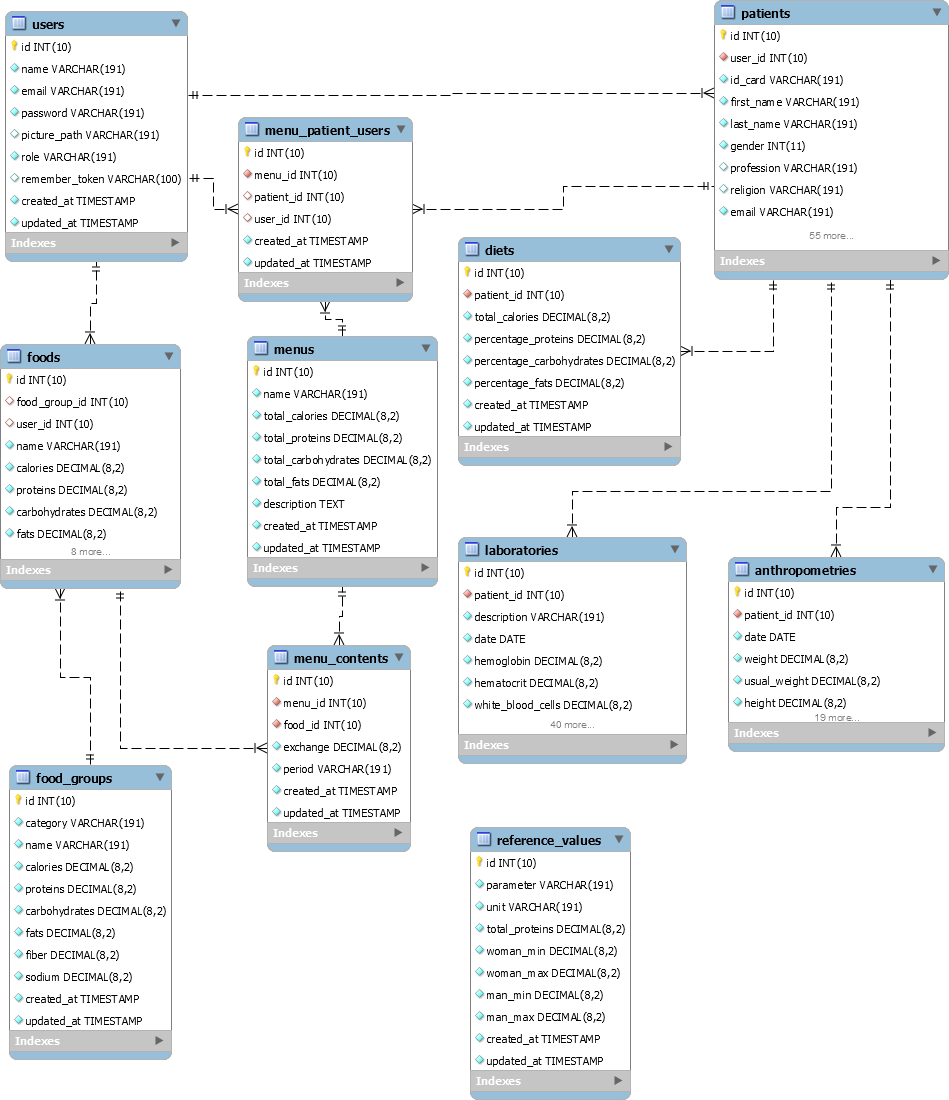
**Figura 4.7 – Arquitectura de la Solución Propuesta**

* 1. **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.

* + 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 él 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 antropometrías 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 haya posea, un paciente puede tener varias antropometrías.
      * **Foods:** contiene tantos los alimentos más utilizados en la sociedad venezolana como cualquier alimento que el nutricionista desee agregar a su repertorio. S 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 formula dietética y numero 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
    1. **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.

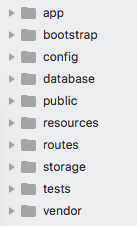
El entorno de trabajo que se decide utilizar para desarrollar la aplicación es Laravel en su versión 5.4. Para hacer uso de este entorno de trabajo es necesario instalar el paquete correspondiente a través del gestor de paquetes Composer y luego crear un proyecto de Laravel dentro del servidor HTTP. 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.

* + - 1. **Estructura de los Directorios**

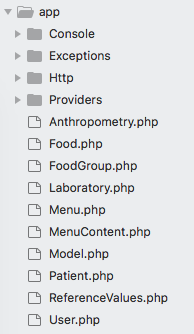
Al crear un proyecto de Laravel se crean los directorios que podemos ver en la siguiente figura 4.9:



**Figura 4.9 - Directorios**

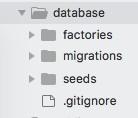
Para efectos del desarrollo de la aplicación se debe prestar especial atención a los directorios “app”, “database”, “public”, “resources” y “routes”. El contenido de estos directorios lo explicamos a continuación:

* + - * + Directorio “app”: En este directorio se encuentran todos los archivos correspondientes al modelo de datos en los cuales de 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).



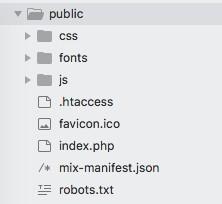
**Figura 4.10 – App folder**

* + - * + 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).



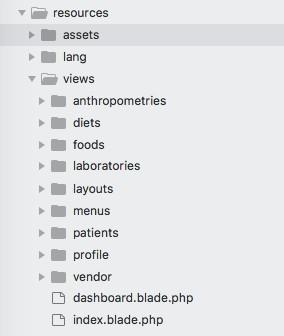
**Figura 4.11 – Database folder**

* + - * + 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
  1. **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.

* + 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.

* + 1. **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 realizados 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.

* + 1. **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).

* + 1. **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.



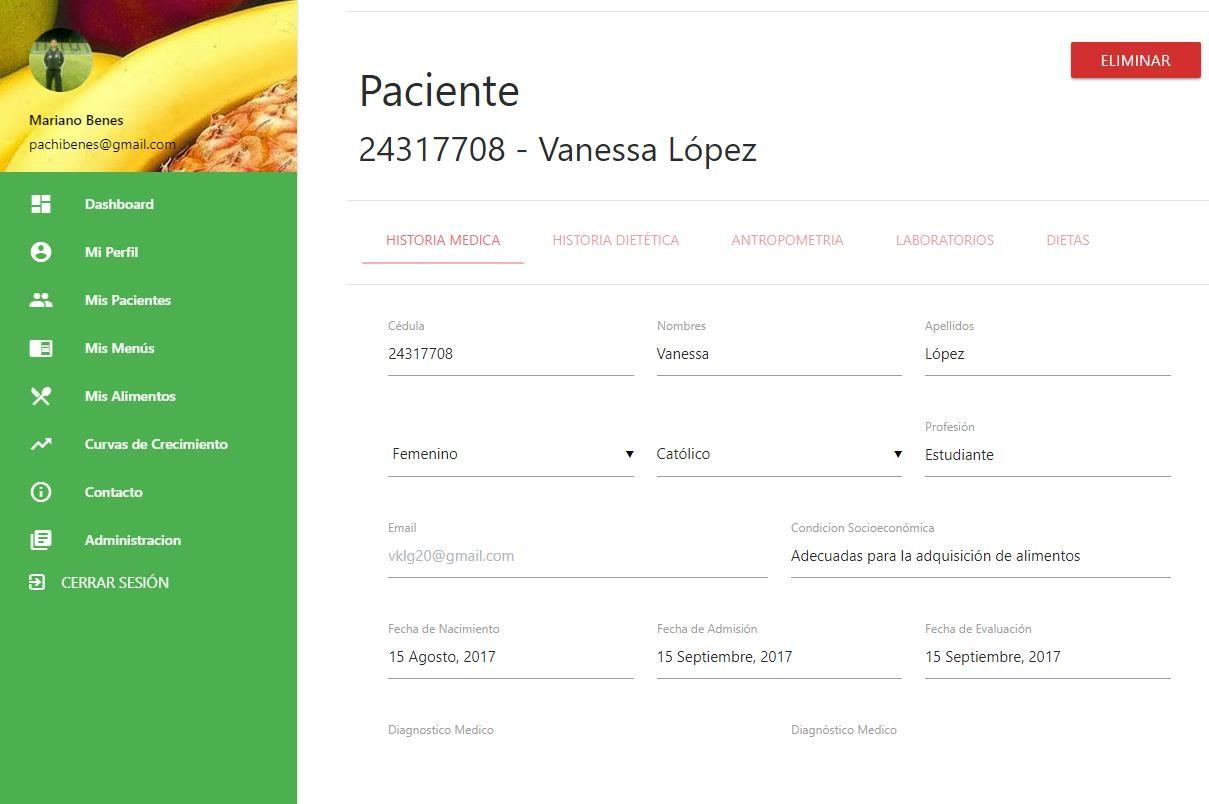
**Figura 4.16: Registro de Pacientes.**

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.

* + 1. **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.

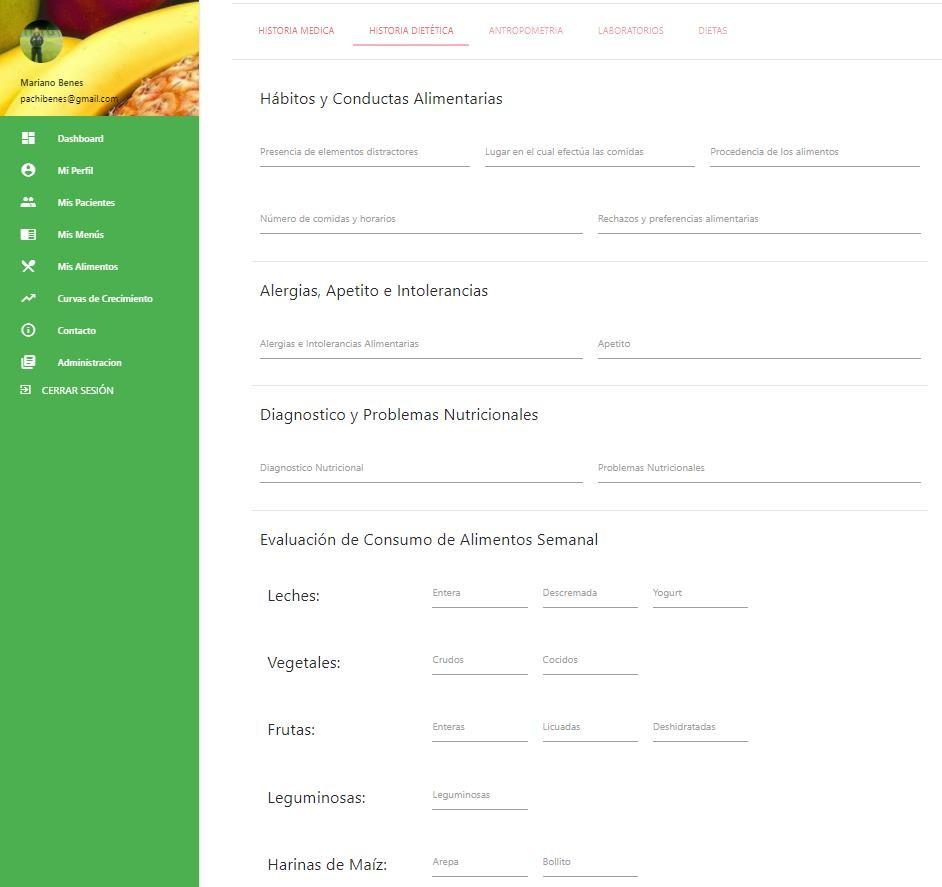


**Figura 4.18: Pacientes**

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 s 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.

* + 1. **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).

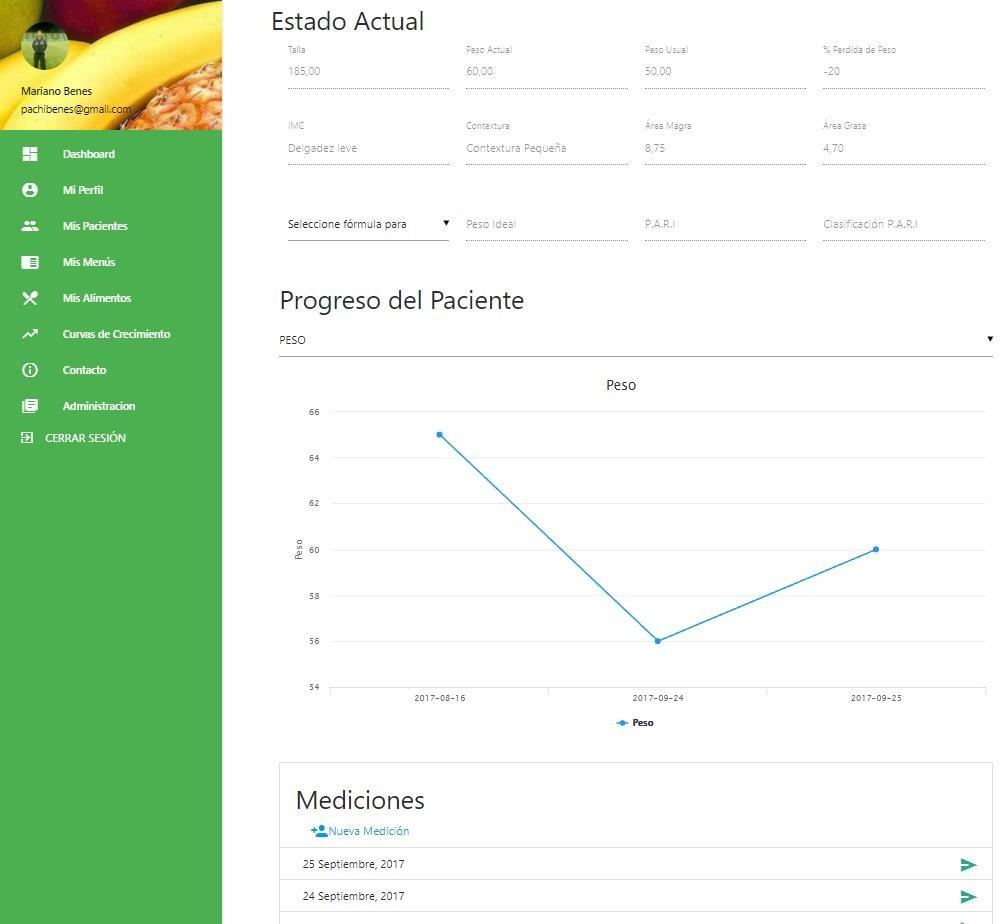


**Figura 4.19: Historia Dietética**

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 hace 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.

* + 1. **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.

* + 1. **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

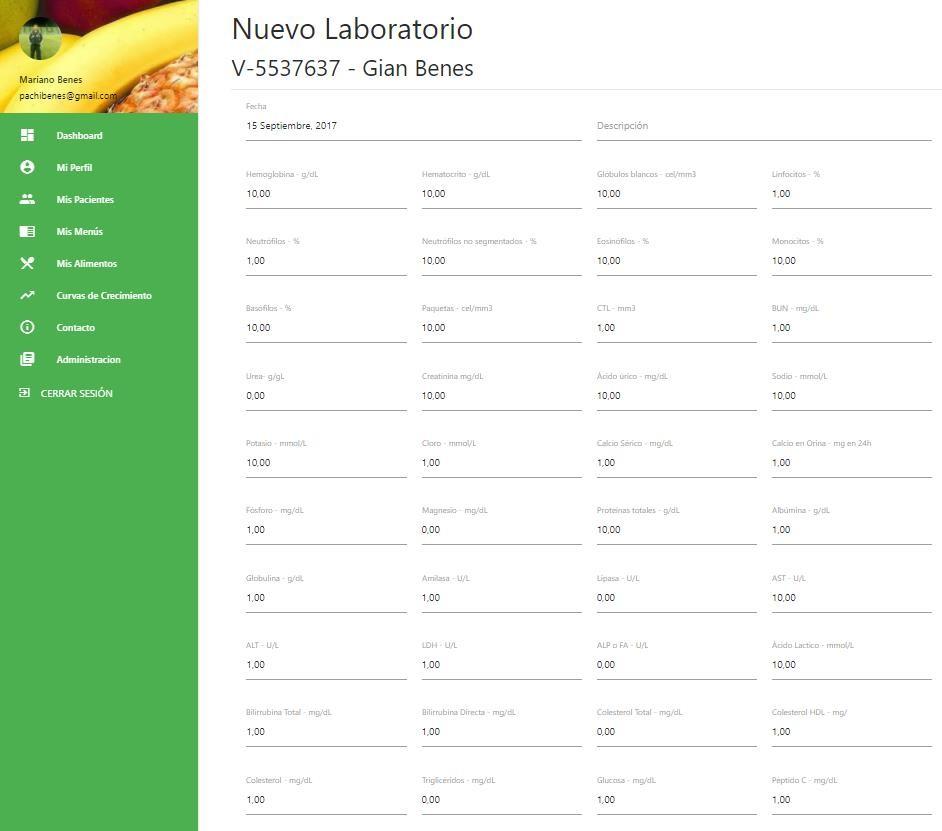


**Figura 4.21: Laboratorios**

Como cada cierto tiempo los laboratorios se deben repetir, es necesario poder guardan 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.



**Figura 4.22: Nuevo Laboratorio**

* + 1. **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.

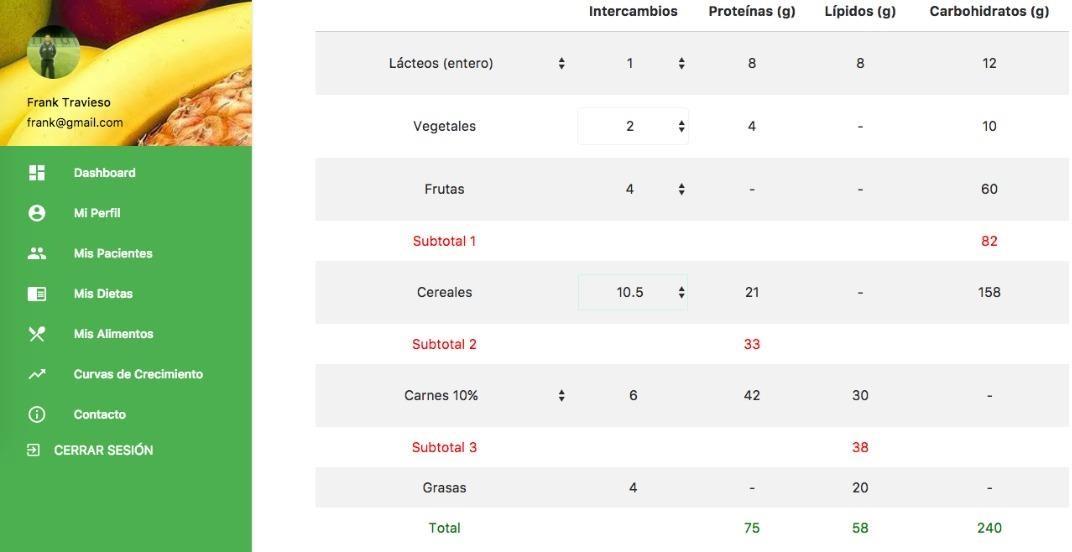
Como se muestra en la figura 4.23, en la ventana de dietas se muestra en la parte superior el requerimiento calórico total del paciente (que no es más que, lo que el paciente debe consumir durante todo el día). En esta interfaz, el usuario debe colocar el peso con el cual desea trabajar y las kcal por 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.



**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.



**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.



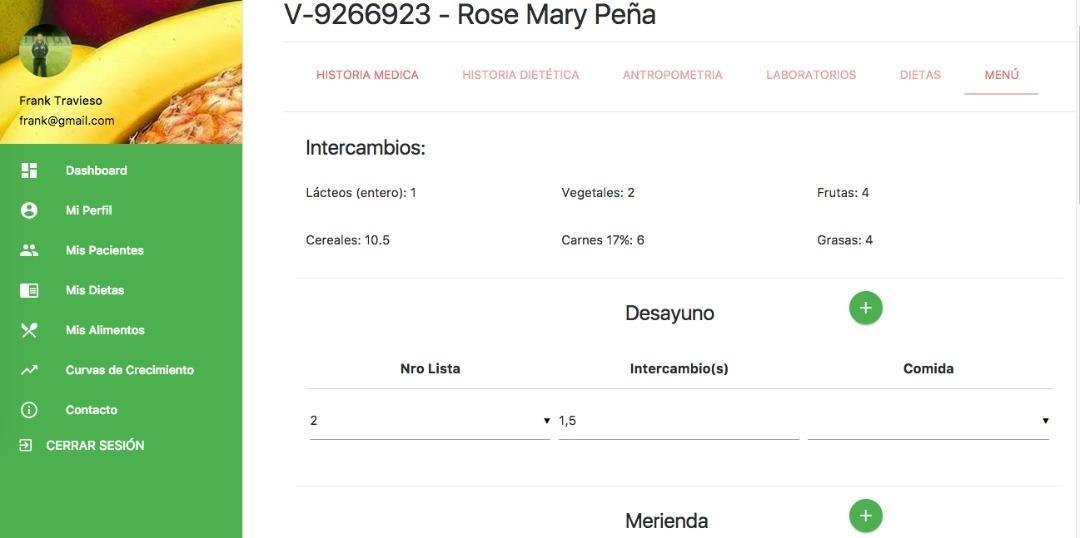
**Figura 4.25: Dietas (Leches)**



**Figura 4.26: Dietas (Carnes)**

* + 1. **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



**Figura 4.27: Menú**

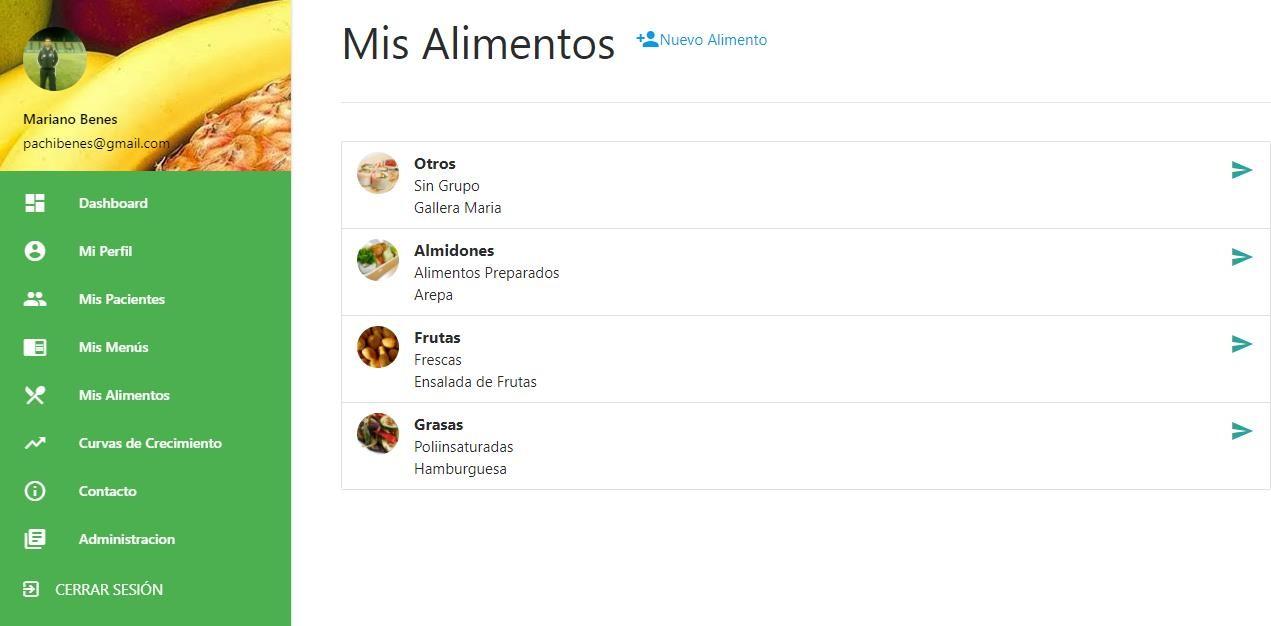
Por tiempo de comida se determina primero el grupo de alimentos y el número de intercambios de este alimento para esta comida en específico, cómo se muestra en la imagen 4.28



**Figura 4.28: Menú**

* + 1. **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.

* + 1. **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



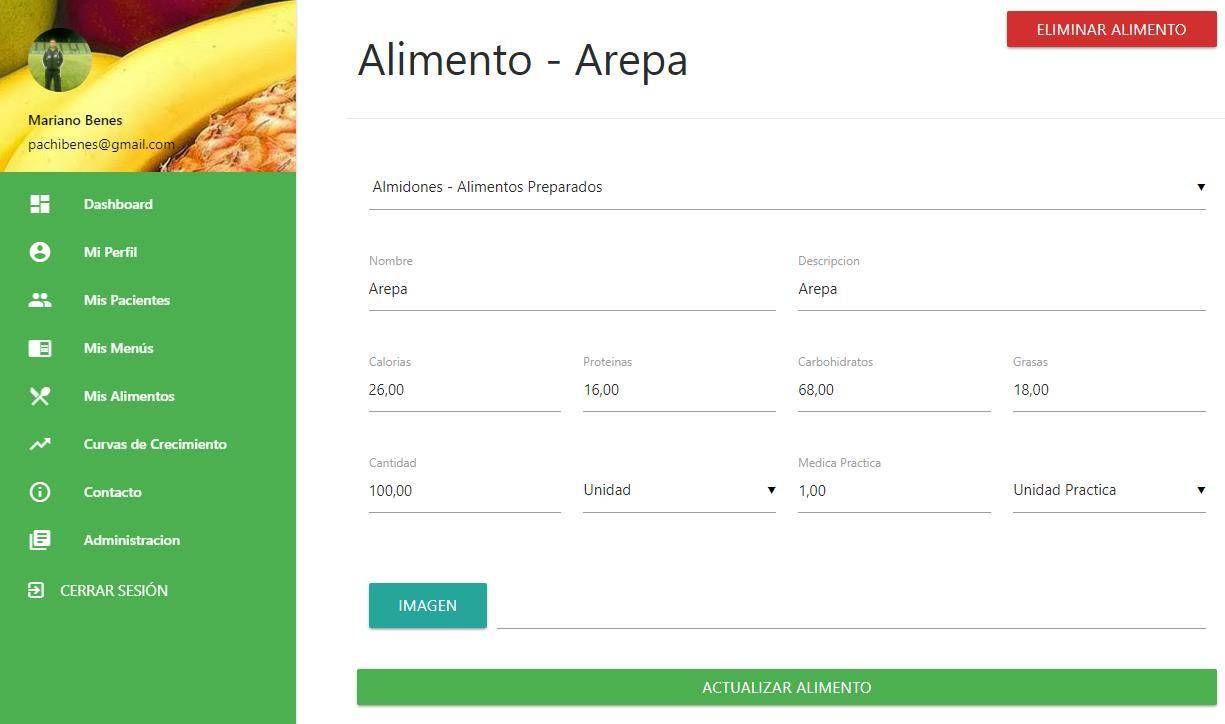
**Figura 4.30: Registro de Alimento**

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.

* + 1. **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



**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.

* 1. **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.

* + 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.
    1. **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.
    1. **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 al 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.

.

* + 1. **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.
    1. **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.
    1. **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.
      * Graficas 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.
  1. **Las pruebas**

A lo largo del desarrollo de la aplicación se realizaron múltiples pruebas tanto 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 hasta que finalmente se cumplieran las funcionalidades requeridas para el mismo.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre del Modulo** | **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 Modulo** | **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** |
|  |  | Creación de Usuario | Aprobado |
|  |  | Visualización de Usuario | Aprobado |
| Menus | 1 | 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 seria 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 brindo 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 cuento 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.
  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.
  1. **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 sofware que incorpora la usabilidad.* Caracas, Venezuela: Universidad Central de Venezuela.

Chávez, M. A. (Marzo de 2015). *Los Frameworks de Ruby*. Obtenido de Medium: https://medium.com/codeando-con-ruby/los-frameworks-de-ruby-1b340f311885#.n0spfvm48

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 ABCDiestisttas: <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>

*Mozilla Developer Network*. (Octubre de 2016). Obtenido de CSS: https://developer.mozilla.org/es/docs/Web/CSS

*Nutrición*. (2017). Obtenido de Organizacion Mundial de la Salud: <http://www.who.int/topics/nutrition/es/> P., C. (2015). *Procesos Ágiles para el desarrollo de apliaciones web.* Madrid, España: Universidad Rey

Juan Carlos. Obtenido de [http://www.dlsi.ua.es/~jaime/webe/articulos/s112.pdf](http://www.dlsi.ua.es/%7Ejaime/webe/articulos/s112.pdf)

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.

S, L. M. (2002). *Programacion de Aplicaciones Web: Historia, Principios Basicos y Clientes Web.* Cuba: Editorial Club Universitario.

*Universidad Nutricion*. (s.f.). Obtenido de Universidad Nutricion:<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&Ite](http://www2.paho.org/hq/index.php?option=com_docman&amp;task=doc_download&amp;gid=17835&amp;Ite) mid=270

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