

**SISTEMA DE GESTIÓN ORIENTADO A LA WEB PARA EL CONTROL DE  
VACUNAS EN UNA ENTIDAD UNIVERSITARIA.**

Luis Carlos Torres Trujillo  
Septiembre 2018

Universidad Nacional Abierta y a Distancia – UNAD  
Escuela de Ciencias Básicas Tecnología e Ingeniería  
Ingeniería de Sistemas

Nota de aceptación

---

---

---

---

Firma del presidente del jurado

---

Firma del jurado

---

Firma del jurado

---

Firma del jurado

Sistema de Gestión orientado a la web para el control de vacunas en una Entidad Universitaria.

por Luis Carlos Torres Trujillo.

Asesora de Proyecto:

Ing. Mireya Gómez Argote.

Escuela de Ciencias Básicas Tecnología e Ingeniería.

El presente proyecto busca optimizar y agilizar los procesos relacionados con la prestación de los servicios de enfermería en el Bienestar Universitario y la gestión de la información del control de vacunas en los estudiantes de la facultad de Ciencias de la Salud de la fundación Universitaria Juan Corpas.

Management System Web Oriented for Vaccines Control in a University Entity.

by Luis Carlos Torres Trujillo.

Project Advisor:

Ing. Mireya Gómez Argote.

Escuela de Ciencias Básicas Tecnología e Ingeniería.

The present Project looks forward to optimize and speed up the processes related to the provision of nursing services in the University Welfare and the management of vaccine control information in the students of the Faculty of Health Sciences of the Juan University Foundation.

## **Agradecimientos**

v

El autor desea expresar su más sincero agradecimiento a la Ingeniera Mireya Gómez Argote por su colaboración en el desarrollo del proyecto. Además, manifiesta su agradecimiento especial a Ronald Steven López, cuya familiaridad con las necesidades e ideas de la entidad Universitaria resultó de enorme utilidad durante la fase inicial de programación de ese proyecto.

A la empresa Optime Consulting Andina S.A.S, empresa donde laboro, por su apoyo y facilitar mis labores para poder desplazarme y atender los requerimientos que demando el presente proyecto. Gracias también a todos los integrantes del Jurado por sus valiosos comentarios.

## **Dedicatoria**

vi

Primero que todo agradecer a Dios por sus bendiciones e iluminarme el camino, con su protección y bendición me ayudo a serle frente a todos los problemas durante mis estudios.

A mis padres y hermanos por estar presente en todos los momentos de mi vida, por sus consejos y por motivarme a culminar la carrera, por infundir en mí, la lucha y el deseo de superación, resaltando el apoyo en los momentos de duda, desesperación y felicidad. Por ultimo a los amigos que nos regala a vida, porque se convirtieron en una familia más.

**Modelo Vista Controlador (MVC):** es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

**Vista:** Uno de los tres componentes del patrón modelo-vista-controlador. Se refieren a los archivos que contienen el código HTML para la visualización de las páginas que se muestran al usuario.

**Controlador:** Uno de los tres componentes del patrón modelo-vista-controlador. Representa el componente el cual define la lógica de la aplicación y cómo responde a las acciones del usuario. Actúa como intermediario entre la vista y el modelo.

**Modelo:** Uno de los tres componentes del patrón modelo-vista-controlador. Es el encargado de trabajar con los datos almacenados en la base de datos. Se encarga de realizar operaciones sobre la base de datos, tales como hacer consultas, insertar, modificar, entre otras, para si es necesario pasar los datos a los controladores.

**Historia Clínica Ocupacional:** Documento privado, obligatorio y sometido a reserva, en el cual se registran cronológicamente las condiciones de salud del paciente, los actos médicos y los demás procedimientos ejecutados por el equipo de salud que interviene en su atención. Dicho documento únicamente puede ser conocido por terceros previa autorización del paciente o en los casos previstos por la ley.

**Ajax:** Siglas de Asynchronous Javascript y XML.

viii

**CSS:** Siglas de cascading style sheets. Es el lenguaje utilizado para definir la presentación de documentos HTML.

**Javascript:** Es un lenguaje de programación para la elaboración de funciones sofisticadas en páginas web.

**Laravel:** Framework de código abierto y libre, creado en PHP por Taylor Otwell para el desarrollo de aplicaciones web usando el patrón de arquitectura modelo-vista-controlador.

**PHP:** Lenguaje de programación de propósito general, usado principalmente para la creación de aplicaciones web del lado del servidor.

**Bootstrap:** Es un framework del lado del cliente de código abierto que permite por medio de plantillas crear páginas HTML junto con CSS y Javascript con estilos y además con la capacidad de ajustarse dinámicamente dependiendo del dispositivo desde el cual se visualizan las páginas.

**Atención en enfermería:** Se refiere a todos los procedimientos de atención al paciente que se realizan en las oficinas de enfermería del departamento de bienestar universitario.

**Atención en salud:** Se refiere a uno de los procedimientos que se realizan en las oficinas de enfermería del departamento de bienestar universitario, y que consiste en un procedimiento



donde se hace uso de insumos médicos, un equipo médico y donde se especifica un motivo de consulta.

**Control de vacunas:** Se refiere a los registros de vacunas que se llevan en las oficinas de la facultad de ciencias de la salud de los estudiantes que están realizando sus prácticas.

**Requerimiento:** Es una especificación documentada de una funcionalidad que debe ser capaz de realizar el sistema.

En el presente documento se pretende explicar los motivos que llevaron al desarrollo de la aplicación web que busca la optimización y agilización de los procesos relacionados con la prestación de los servicios de enfermería y los objetivos que se lograron con el desarrollo de la misma.

Se presenta la definición del problema y su contexto, la justificación del rol de las TIC's en el apoyo de los procesos para la toma de decisiones dentro de una organización, lo cual lleva a agilizar y automatizar ciertas tareas, tales como el control de la información y la generación de informes de forma automatizada y así lograr implementar el plan "Cero Papel", estrategia del Plan Nacional de Desarrollo que busca racionalizar el uso de recursos de papel en las entidades del estado promoviendo la eficiencia y productividad, reduciendo costos, tiempo y espacios de almacenamiento.<sup>1</sup>

Todo esto como producto de las necesidades que se encontraron en las oficinas del área de salud del bienestar universitario, y en la facultad de Ciencias de la salud de la Fundación Universitaria Juan N. Corpas. Por todo lo anterior, se pone a consideración la aplicación web y las diferentes actividades que se llevaron a cabo para su desarrollo, tales como la metodología de desarrollo usada, es decir ICONIX, y las diferentes etapas que conforman la metodología, es decir, el análisis de requerimientos, análisis y diseño preliminar, el diseño detallado y la implementación.

---

<sup>1</sup> Ministerio de Tecnologías de la Información y las Comunicaciones. (s.f.). Cero Papel en la administración pública. Buenas prácticas para reducir el consumo en la oficina. Recuperado el 14 de agosto de 2018, de [http://estrategia.gobiernoenlinea.gov.co/623/articles-8257\\_papel\\_buenaspracticas.pdf](http://estrategia.gobiernoenlinea.gov.co/623/articles-8257_papel_buenaspracticas.pdf).

## Tabla de Contenidos

xi

Capítulo 1. Aspectos Generales.....	1
1.1. Título.....	1
1.2. El Problema.....	1
1.2.1. Descripción del Problema.....	1
1.2.2. Formulación del Problema.....	3
1.2.3. Justificación.....	3
1.3. Objetivos.....	5
1.3.1. Objetivo General.....	5
1.3.2. Objetivos Específicos.....	5
1.4. Tipo de Investigación.....	6
Cualitativa, Descriptiva.....	6
1.5. Alcance.....	7
Capítulo 2. Marcos de Referencia.....	8
2.1. Marco Referencial.....	8
2.1.1. Marco Teórico.....	8
2.1.2. Sistema de Información de salud y registros de salud electrónicos.....	8
2.1.3. Metodología de desarrollo de software.....	9
2.1.4. Metodología de desarrollo de ICONIX.....	10
2.1.5. Etapas de la metodología de desarrollo de ICONIX.....	10
2.1.5.1. Análisis de requerimientos.....	11
2.1.5.2. Análisis y Diseño Preliminar.....	12
2.1.5.3. Diseño Detallado.....	13
2.1.6. Implementación.....	14
2.1.7. UML.....	14
2.1.7.1. Diagrama de Clases.....	17
2.1.7.2. Diagrama de Caso de Uso.....	20
2.1.7.2.1. Caso de Uso en formato expandido.....	22
2.1.7.3. Diagrama de Secuencia.....	23
2.1.7.4. Diagrama de Robustez.....	26
2.1.8. Aplicación WEB.....	28

2.1.9. PHP .....	28
2.1.10. Laravel .....	28
2.1.11. Servidor WEB.....	29
2.1.12. Bases de Datos .....	30
2.2. Estado del Arte.....	31
2.3. Marco Contextual.....	38
2.3.1. Fundación Universitaria Juan N. Corpas. ....	38
2.3.2. Bienestar Universitario. ....	39
2.3.3. Visión.....	39
2.3.4. Misión. ....	40
2.3.5. Propósitos.....	41
2.3.6. Marco Legal. ....	44
Capítulo 3. Ingeniería del Proyecto. ....	45
3.1. Metodología de Desarrollo. ....	45
3.2. Análisis de Requerimientos. ....	45
3.2.1. Captura de Requerimientos.....	45
3.2.2. Modelo de Dominio. ....	47
3.2.3. Diagramas de Casos de Uso.....	48
3.2.4. Casos de Uso en Formato Expandido. ....	50
3.3. Análisis y Diseño Preliminar. ....	52
3.4. Diseño Detallado.....	53
3.5. Implementación.....	54
3.5.1. Diagrama de Clases.....	55
3.5.2. Diagrama Entidad – Relación. ....	55
3.5.3. Arquitectura de Software. ....	55
3.5.4. Código Fuente.....	59
Capítulo 4. Conclusiones y Recomendaciones. ....	69
4.1. Conclusiones.....	69
4.2. Recomendaciones. ....	71
Lista de referencias .....	74

Figure 1. Esquema de los diagramas UML y su clasificación. Fuente: El Autor. .... 16

Figure 2. Formato de un diagrama de clase. Fuente: El autor. Software VioletUMLEditor. .... 17

Figure 3. Representación gráfica de las relaciones entre los diagramas de clases. Fuente: El autor.  
Software Corel Draw X7. .... 19

Figure 4. Formato de un Caso de Uso. Fuente: El autor. .... 20

Figure 5. Representación gráfica de un actor en un caso de uso. Fuente: El autor. Software Corel  
Draw X7. .... 21

Figure 6. Representación gráfica de las relaciones usadas en un caso de uso. Fuente: El autor.  
Software Corel Draw X7. .... 21

Figure 7. Representación gráfica de un objeto y su línea de vida en un diagrama de secuencia.  
Fuente: El autor. Software Corel Draw X7. .... 24

Figure 8. Mensaje síncrono en un diagrama de secuencia. Fuente: El autor. Software Corel Draw  
X7. .... 25

Figure 9. Mensaje de respuesta en un diagrama de secuencia. Fuente: El autor. Software Corel  
Draw X7. .... 25

Figure 10. Mensaje asíncrono en un diagrama de secuencia. Fuente: El autor. Software Corel  
Draw X7. .... 26

Figure 11. Símbolos de un diagrama de robustez. Fuente: El autor. Software Corel Draw X7. .. 27

Figure 12. Modelo de dominio del sistema. Fuente: El autor. Software Violet UML Editor. .... 48

Figure 13. Diagrama de caso de uso para el módulo de control de inventario. Fuente: El autor.  
Software Visual Paradigm. .... 49

Figure 14. Detalle caso de uso gestionar medicamento. Fuente: El autor. Software Visual  
Paradigm. .... 50

Figure 15. Diagrama de robustez para el caso de uso registrar medicamento. Fuente: El autor.  
Software Visual Paradigm Community Edition. .... 52

Figure 16. Diagrama de secuencia para el caso de uso - Registrar Medicamento. Fuente: El autor.  
Software Visual Paradigm Community Edition. .... 54

Figure 17. Diagrama de la arquitectura cliente - servidor. Fuente: El autor. Software Visio  
Profesional. .... 56

Figure 18. Diagrama de componentes del modelo - vista - controlador de Laravel en la aplicación. Fuente: El autor. Software Visual Paradigm. ....	xiv 57
Figure 19. Diagrama de despliegue de la aplicación. Fuente: El autor. Software Visual Paradigm. ....	58
Figure 20. Vista que muestra el listado de medicamentos registrados. Fuente: El autor. ....	60
Figure 21. vista del formulario de registro de medicamento. Fuente: El autor. ....	61
Figure 22. Mensaje de éxito al crear un nuevo medicamento. Fuente: El autor. ....	61
Figure 23. Formulario de registro de medicamentos con mensaje de error. Fuente: El autor. ....	62
Figure 24. Código fuente vista de registro de medicamento. Parte 1. Fuente: El autor. ....	63
Figure 25. Código fuente vista de registro de medicamento. Parte 2. Fuente: El autor. ....	64
Figure 26. Código fuente vista de registro de medicamento. Parte 3. Fuente: El autor. ....	65
Figure 27. Código fuente controlador de registro de medicamento. Parte 1. Fuente: El autor. ....	66
Figure 28. Código fuente controlador de registro de medicamento. Parte 2. Fuente: El autor. ....	67

## Lista de Tablas

xv

Table 1. Cuadro de resumen del Estado del Arte. Fuente: El autor.....	31
Table 2. Requerimiento N° 1 del sistema. Fuente: El autor.....	46
Table 3. Casos de uso registrar medicamento en formato expandido. Fuente: El autor.....	51

## **Capítulo 1.**

### **Aspectos Generales.**

#### **1.1. Título.**

Desarrollo de un sistema de información web para el servicio de enfermería en la facultad de Ciencias de la salud de la Fundación Universitaria Juan N. Corpas.

#### **1.2. El Problema.**

##### 1.2.1. Descripción del Problema.

En las oficinas de la facultad de Ciencias de la Salud, pertenecientes a la dependencia de Bienestar Universitario, se ofrecen tres tipos de servicios a los estudiantes, docentes u otra persona que esté vinculada a la institución.

Estos tres servicios son consulta médica, enfermería y salud ocupacional. Durante el proceso de atención a los usuarios se deben llenar una serie de formatos físicos, donde se registra la información acerca de los mismos. De acuerdo a los datos consultados de los archivos de los informes mensuales, se encontró que el número de registros que se generan durante un periodo académico de seis meses, es alrededor de 2100.<sup>2</sup>

Adicionalmente, se lleva a cabo la programación de citas médicas y registro de antecedentes de historiales clínicos para cada estudiante o funcionario de la institución, los cuales también se escriben en formatos físicos. Por otro lado, una vez que la persona

---

<sup>2</sup> Información validada en las oficinas de enfermería del Bienestar Universitario de la Fundación Universitaria Juan N. Corpas.



asiste a una cita médica se lleva a cabo todo el registro requerido dentro de su historia clínica con el cual, si es necesario, se le realiza y programa un seguimiento para evaluar su estado de salud. Estos registros son documentados en formatos impresos.

Igualmente sucede con los procedimientos llevados a cabo por la jefe de enfermería como registro de suministro de medicamentos de planificación, medicina general y primeros auxilios. Estos llevan a un proceso de inventariar los fármacos existentes, el cual es engorroso por el hecho de utilizar medios físicos y manuales para contabilizar la existencia de todo tipo de medicamentos y su control sobre las características que poseen como las fechas de vencimiento.

Luego de haber realizado todos los procesos de registro en un determinado periodo, es necesario generar informes estadísticos que muestren el estado y el uso de todos los insumos y servicios prestados en las oficinas de Enfermería. Estos informes se hacen a partir de los registros físicos que se han almacenado durante todo un periodo donde se han brindado servicios a la comunidad universitaria. Dado que se manejan muchos registros físicos se hace dificultoso y demorado el manejo de toda la información para así generar los informes estadísticos.

Existe además otra situación dentro de la facultad de Ciencias de la Salud, relacionada con las hojas de vida que contiene la información de las vacunas aplicadas en los estudiantes de medicina que inician o renuevan sus actividades prácticas dentro de los hospitales que están vinculados con la institución. Los estudiantes deben llenar una serie de formatos físicos donde la información requerida es sobre el control de las dosis

suministradas que tienen aplicadas para que los hospitales puedan establecer si el estudiante puede o no iniciar sus actividades de práctica.

Este proceso al igual que ocurre en las oficinas de enfermería, genera mucho gasto de papel y hace que el manejo de toda la información recolectada sea demorada y dificultosa.

### 1.2.2. Formulación del Problema

¿Cómo podría un sistema de información web agilizar los procesos relacionados con la prestación de los servicios de enfermería en el Bienestar Universitario y la gestión de la información del control de vacunas en los estudiantes de la facultad de Ciencias de la salud de la Fundación Universitaria Juan N. Corpas?

### 1.2.3. Justificación.

De acuerdo al Plan Nacional de Desarrollo 2010 – 2014<sup>3</sup>, el Gobierno Nacional busca mejorar su gestión pública siendo una de sus estrategias la denominada “Cero Papel”<sup>4</sup>, que consiste en la sustitución de los flujos documentales en papel por soportes y medios electrónicos, sustentados en las Tecnologías de la Información y las Telecomunicaciones.

---

<sup>3</sup> Gobierno de Colombia. (s.f.). Departamento Nacional de Planeación. Recuperado el 14 de agosto de 2018, de Plan Nacional de Desarrollo: <https://www.dnp.gov.co/Plan-Nacional-de-Desarrollo/PND-2010-2014/Paginas/Plan-Nacional-De-2010-2014.aspx>

<sup>4</sup> Consejo Profesional Nacional de Arquitectura y sus Profesiones Auxiliares. (s.f.). Política cero papel del consejo profesional nacional de arquitectura y sus profesiones auxiliares - CPNAA. Recuperado el 14 de agosto de 2018, de <https://cpnaa.gov.co/sites/default/files/docs/poli%CC%81tica%20cero%20papel.pdf>

Esta estrategia busca ir en favor del ambiente e incrementar la eficiencia administrativa, identificando y aplicando buenas prácticas para reducir el consumo de papel, de acuerdo con la guía que ya está disponible en el Programa Gobierno en Línea.

Mediante la implementación de la estrategia gubernamental “Cero Papel”, existe la oportunidad de generar buenos hábitos relacionados con el uso del papel y de aprovechamiento de las tecnologías de la información para mejorar la productividad, reduciendo costos, tiempo y espacios donde se almacenan los archivos físicos.

Después de que se realizan numerosos registros y consulta en las oficinas de enfermería y en la facultad de Ciencias de la Salud, se obtiene un número alto de informes y registros hechos a mano que describen las mismas. A partir de estos, se elabora un historial y estadísticas sobre el control de las vacunas, el uso de los insumos, medicamentos y recepción de citas médicas, además de un control de inventario de los insumos usados para dichas actividades. La generación de estos registros e informes hace que se impriman documentos de borrador dos o tres veces para ser revisados. Esto nos indica que no se aprovechan las herramientas tecnológicas que son alternativas al papel.

Agilizar este proceso se sustenta en lo establecido por la estrategia “Cero Papel”, en cuanto a que se debe identificar, racionalizar, simplificar y automatizar los trámites y los procesos, procedimientos y servicios internos.

El gobierno busca que en la administración pública se establezca una cultura de eficiencia de gestión documental a través de la tecnología y de uso racional de los recursos disponibles. También se debe cumplir con toda la normatividad que exige

SNIES<sup>5</sup> para poner a disposición la información de la institución ante agencias del sector, entes del gobierno y la comunidad en general, para facilitar la orientación de las IES en los procesos de mejoramiento a partir de la identificación de mejores prácticas.

Esto permite a los funcionarios de las oficinas de Enfermería y la facultad de Ciencias de la Salud tomar decisiones de una manera más organizada ya que con un sistema de información se podrá tener estadísticas a la mano sobre el control de vacunas, el uso de insumos y medicamento usados, junto con sus historiales de uso.

### **1.3. Objetivos.**

#### **1.3.1. Objetivo General.**

Desarrollar un sistema de información web que permita gestionar los datos generados por la prestación de servicios de salud en las oficinas de enfermería del Bienestar Universitario y el control de vacunas de los estudiantes de la facultad de facultad de Ciencias de la salud de la Fundación Universitaria Juan Corpas.

#### **1.3.2. Objetivos Específicos.**

- Analizar los procesos de manejo actual de la información en las oficinas de enfermería del Bienestar Universitario y en la facultad de Ciencias de la Salud con el fin de realizar el correspondiente levantamiento de requisitos para el diseño y desarrollo del sistema de información.

---

<sup>5</sup> Gobierno de Colombia. (s.f.). Sistema Nacional de Información de la Educación Superior - SNIES. Recuperado el 14 de agosto de 2018, de <https://www.mineduacion.gov.co/sistemasdeinformacion/1735/w3-article-211884.html>

- Diseñar un sistema de información que permita gestionar los datos producidos por la unidad de información de una forma más adecuada y agilizar los procesos de citas, historias clínicas, registro de vacunación, consulta y generación de inventarios.
- Desarrollar un módulo que permita apoyar las actividades relacionadas con la elaboración de informes estadísticos del gasto de los medicamentos y el mantenimiento de los implementos que se usan dentro de las oficinas de Enfermería. Al igual que un módulo donde se gestione la información requerida de vacunas de los estudiantes que se encuentran en su proceso de realización de prácticas.

#### **1.4. Tipo de Investigación.**

Cualitativa, Descriptiva

En las oficinas de Enfermería se llevan a cabo procedimientos de registro y consulta, con los cuales se les brinda un servicio de atención a estudiantes, docentes y funcionarios. De esta forma se generan documentos con información que llevan un registro de todos los servicios de salud, medicamentos e insumos prestados. Así mismo, en la facultad de Ciencias de la Salud los estudiantes deben diligenciar una serie de formatos físicos donde reposa la información requerida por los hospitales para establecer si el estudiante puede o no iniciar sus prácticas dentro de un hospital.

Se busca especificar las propiedades de todos aquellos procedimientos que son realizados en las oficinas de Enfermería y en la facultad de Ciencias de la Salud mediante

un análisis de los mismos. Para ello se selecciona una serie de actividades y se mide cada una de ellas independientemente, para así describir lo que se investiga. Se miden conceptos o variables a los que se refieren.

### **1.5. Alcance.**

Para el desarrollo del proyecto se implementará un conjunto adecuado de controles, que abarcan políticas, prácticas, procedimientos, estructuras organizacionales y funciones de software como lo son:

- Módulo de control de Inventario.
  - Medicamentos
  - Insumos
- Módulo de gestión de citas.
- Módulo de gestión de consulta médica.
  - Administración de medicamentos
  - Servicio de inyectología
  - Toma de tensión arterial
  - Toma de glucometría.
- Módulo de gestión de historias clínicas.
- Módulo de gestión de los medicamentos de planificación familiar.
- Módulo de control de vacunas a estudiantes practicantes.

## Capítulo 2.

### Marcos de Referencia.

#### 2.1. Marco Referencial.

##### 2.1.1. Marco Teórico.

Se presenta a continuación el marco teórico relacionado con los sistemas de información de salud y la metodología de desarrollo de software ICONIX.

##### 2.1.2. Sistema de Información de salud y registros de salud electrónicos.

En un sentido general, un sistema es un conjunto de componentes que interactúan entre sí con el objetivo de alcanzar una meta. Un sistema de información es un conjunto de personas, datos, procesos y tecnología de la información que interactúan para recopilar, procesar, guardar y proporcionar como salida la información necesaria para brindar un soporte a una organización<sup>6</sup>.

En el campo de la salud, un sistema de información es aquel que integra la recolección, procesamiento y reporte de información necesaria para el mejoramiento de los servicios de salud, a través de una mejor gestión en todos los niveles del sistema<sup>7</sup>.

Los sistemas de información en salud tienen como componente principal los registros de salud electrónicos, los cuales son registros almacenados en un sistema de

---

<sup>6</sup> WHITTEN, J. L. (2008). ANÁLISIS DE SISTEMAS. DISEÑO Y MÉTODOS. México: Whitten Bentley. Recuperado agosto 14, 2018

<sup>7</sup> BERNAL ACEVEDO, o. &. (2011). Sistemas de información en el sector salud en Colombia. Revista Gerencia y políticas de Salud. Recuperado 19 agosto, 2018, de [http://www.scielo.org.co/scielo.php?script=sci\\_arttext&pid=S1657-70272011000200006&lang=pt](http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S1657-70272011000200006&lang=pt)

información electrónico que se apoya de una base de datos para cumplir su tarea. Un registro de salud electrónico es una versión digital de los registros de salud de un paciente. Los registros de salud electrónicos son registros en tiempo real, que se centran en el paciente y que hacen que la información esté disponible de manera casi instantánea y segura a sus usuarios autorizados.

Un sistema de registro de salud electrónico, está hecho para ir más allá de los datos registrados de manera estándar, ya que pueden dar una observación más amplia del cuidado del paciente. Estos registros pueden contener historias clínicas, diagnósticos, planes de tratamiento, fechas de inmunización, alergias, entre otras cosas. Los registros de salud electrónicos ofrecen varias ventajas, como lo son la reducción de consumo de papel, aumentar la productividad, apoyar la toma de decisiones, permitir el acceso inmediato a la información del paciente entre otras<sup>8</sup>.

### 2.1.3. Metodología de desarrollo de software.

Una metodología de desarrollo de software, también llamada proceso de software, es una serie de pasos que se realizan de forma consecutiva y secuencial que permite llevar a cabo de forma organizada un proyecto de desarrollo de software.

Define un conjunto de actividades, acciones y tareas que se ejecutan cuando va a crearse algún producto del trabajo. Permite que las personas que hacen el trabajo busquen y elijan el conjunto apropiado de acciones y tareas para el trabajo. Se busca siempre

---

<sup>8</sup> OFFICE OF THE NATIONAL COORDINATOR FOR HEALTH INFORMATION TECHNOLOGY. (21 de marzo de 2018). What is an electronic health record? Recuperado el 25 de agosto de 2018, de <https://www.healthit.gov/faq/what-electronic-health-record-ehr>



entregar el software en forma oportuna y con calidad suficiente para satisfacer a quienes patrocinaron su creación y a aquellos que lo usaran.

La estructura del proceso de la metodología del desarrollo de software establece el fundamento para el proceso completo de la ingeniería de software por medio de la identificación de un número de actividades que sean aplicables a todos los proyectos de software, sin importar su tamaño o complejidad<sup>9</sup>.

#### 2.1.4. Metodología de desarrollo de ICONIX.

ICONIX es una metodología de desarrollo de software minimalista, orientada por casos de uso como proceso de modelamiento el cual se ajusta al desarrollo ágil. Usa un conjunto de diagramas de UML, y provee de un método confiable para ir desde los casos de uso hasta el código fuente en el menor número de pasos posibles<sup>10</sup>.

#### 2.1.5. Etapas de la metodología de desarrollo de ICONIX.

ICONIX consta de cuatro etapas, las cuales son. Análisis de requisitos, análisis y diseño preliminar, diseño detallado e implementación<sup>11</sup>. A continuación, se describe de manera más detallada en que consiste cada una de estas etapas.

---

<sup>9</sup> PRESSMAN, R. S. (2010). Ingeniería de software. Un enfoque práctico. México: McGraw-Hill.

<sup>10</sup> ASSOCIATES, M. &. (s.f.). Agile Development with ICONIX process. Recuperado el 2 de agosto de 2018, de <http://www.methodsandtools.com/archive/archive.php?id=22>

<sup>11</sup> ROSENBERG, D. y. (2007). Use Case Driven Object Modeling with UML. Theory and Practice. EE.UU.: Apress.

#### 2.1.5.1. Análisis de requerimientos.

La primera etapa de la metodología está relacionada con la captura de requerimientos. Como con todos los proyectos de desarrollo de software, se debe definir cuáles son las funcionalidades que el sistema debe ser capaz de realizar<sup>12</sup>.

La etapa de análisis de requerimientos se describe a continuación en orden ascendente:

- **Requerimientos funcionales.** Se define cuáles deben ser las funcionalidades que el Sistema debe realizar. Dependiendo de cómo el proyecto se encuentre organizado, los requerimientos pueden ser hechos por un equipo de analistas.
- **Modelado de dominio.** Entender el espacio del problema en términos no ambiguos.
- **Requerimientos de comportamiento.** Consiste en definir, por medio de casos de uso, como el usuario y el sistema van a interactuar. Además, desde este momento se recomienda comenzar a elaborar prototipos para la interfaz gráfica, identificando todos los casos de uso que se van a implementar, o al menos elaborar una lista de los casos de uso iniciales, los cuales se esperan que tengan modificaciones conforme se exploran los requerimientos.
- **Revisión de los requerimientos.** Se hace una revisión de los requerimientos para verificar que los mismos coincidan con las expectativas de los usuarios que harán uso del sistema.

---

<sup>12</sup> Ibid., p. 7

### 2.1.5.2. Análisis y Diseño Preliminar.

En esta etapa, el diseño preliminar actúa como un paso intermedio entre el análisis y el diseño. La etapa de diseño preliminar, también conocida como análisis de robustez, involucra la realización del diseño exploratorio que se necesita para entender mejor los requerimientos, refinando y quitando las ambigüedades de aquellos requerimientos como resultado del diseño exploratorio, y enlazando los requerimientos de comportamiento a los objetos del modelo de dominio<sup>13</sup>.

La etapa de análisis y diseño preliminar consiste de los siguientes pasos:

- **Análisis de robustez.** Se dibuja un diagrama de robustez para cada caso de uso, el cual es una especie de imagen de los pasos de un caso de uso en términos de objetos. Mientras se hace esto, se vuelve a escribir el caso de uso cuando sea necesario.
- Se actualiza el modelo de dominio mientras se escribe el caso de uso y mientras se elabora el diagrama de robustez. Aquí se podrán descubrir clases aun no descubiertas, corregir ambigüedades y añadir atributos a los objetos, como, por ejemplo, identificar que un objeto de la clase “libro” posee un atributo llamado “título”.
- Se nombra todas las funciones de la lógica del software, también llamados controladores, que se necesitan para hacer que el caso de uso funcione. Luego se vuelven a escribir el primer borrador de casos de uso elaborado.

---

<sup>13</sup> Ibid., p. 9

- **Revisión del diseño preliminar.** La revisión del diseño preliminar ayuda a verificar que los diagramas de robustez, el modelo de dominio y los casos de uso concuerden los unos con los otros. Esta revisión es una entrada entre la etapa de análisis y diseño preliminar y la etapa del diseño detallado.

#### 2.1.5.3. Diseño Detallado.

El diseño detallado es la etapa que consiste en la construcción correcta del sistema. Se espera que, para esta etapa, ya se tenga una buena comprensión de lo que el sistema y su conceptualización significa, ya que se ha hecho bastante trabajo para desarrollar ese entendimiento<sup>14</sup>.

La etapa del diseño detallado consiste de los siguientes pasos:

- **Elaboración de diagramas de secuencia.** Se elabora un diagrama de secuencia por cada caso de uso, para mostrar en detalle cómo se va a implementar el caso de uso. La principal función de los diagramas de secuencia es asignar comportamientos a las clases.
- Se actualiza el modelo de dominio mientras se elabora el diagrama de secuencia, y se añaden operaciones o funciones a los objetos del dominio. Para este momento, los objetos se están invirtiendo en un modelo estático, o diagrama de clases, la cual es una parte crucial del diseño detallado.
- **Revisión del modelo detallado.** Consiste en asegurarse que el diseño detallado coincida con lo que se ha especificado en los requerimientos. Así mismo, se

---

<sup>14</sup> Ibid., p. 1

revisa la calidad del diseño y se verifica la continuidad de los mensajes en los diagramas de secuencia.

#### 2.1.6. Implementación.

La implementación es el proceso donde se inicia la codificación y además se elaboran las pruebas. Para este punto el diseño detallado es crucial ya que los modelos elaborados en el diseño deben ser la base para saber exactamente que funcionalidades del software van a someterse a pruebas unitarias, de tal manera que se puedan conducir las pruebas unitarias desde el modelo, de una manera similar a la generación de código a partir de los diagramas de clases de tallados<sup>15</sup>.

La implementación consiste de los siguientes pasos:

- **Codificación y pruebas unitarias.** Se describe el código y las pruebas unitarias.
- **Pruebas de integración y de escenario.** Se elaboran las pruebas de escenarios basadas en los casos de uso, de tal manera que se están probando tanto el curso básico como los cursos alternativos.
- Se hace una revisión del código y una actualización del modelo para prepararse para la siguiente ronde de trabajo de desarrollo.

#### 2.1.7. UML.

En ingeniería de software, UML son las siglas de Unified Modeling Language, que traduce lenguaje de modelado unificado. UML es un lenguaje de modelado

---

<sup>15</sup> Ibid., p. 15

estandarizado que consiste de un conjunto de diagramas desarrollados para ayudar a los desarrolladores de aplicaciones y sistemas, poder especificar, visualizar, construir y documentar los elementos de un sistema de software.

El UML representa una colección de las mejores prácticas de ingeniería que se han probado ser exitosas en el modelado de grandes procesos de desarrollo<sup>16</sup>.

Antes de la aparición de UML, el desarrollo de sistemas era con frecuencia una propuesta al azar, ya que los analistas de sistemas intentaban evaluar los requerimientos de sus clientes, generar un análisis de requerimientos en algún tipo de notación que ellos mismos comprendieran incluso aunque el cliente no lo comprendiera, dar este análisis a uno o varios programadores y esperar a que el producto final cumpliera con lo que el cliente deseaba.

En los principios de la computación, los programadores no realizaban análisis muy profundos sobre el problema a resolver. Con frecuencia comenzaba a escribir el programa desde el principio, y el código necesario se escribían conforme se requería. En la actualidad esta manera de realizar un proyecto de desarrollo de software es inapropiado. Hoy en día es necesario contar con un plan bien analizado<sup>17</sup>.

UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas<sup>18</sup>. Debido a que UML es un lenguaje, cuenta con reglas para combinar tales elementos. La finalidad de los diagramas UML, es representar una especie

---

<sup>16</sup> VISUAL PARADIGM. What is Unified Modeling Language (UML)? [En línea]. [Citado agosto 2 de 2018]. Disponible en internet: <<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/>>

<sup>17</sup> SCHMULLER, Joseph. Aprendiendo UML en 24 horas. México: Prentice Hall, s.f. 7 p.

<sup>18</sup> Ibid., p. 8

de plano del sistema que se desarrolla o se está desarrollando, de tal manera que los analistas y desarrolladores puedan hacerse una imagen de cómo está compuesto el sistema en su totalidad.

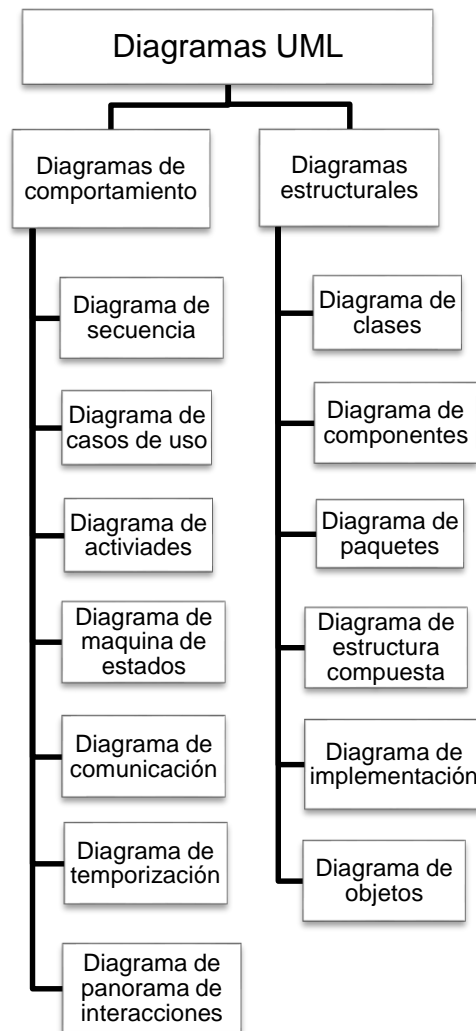


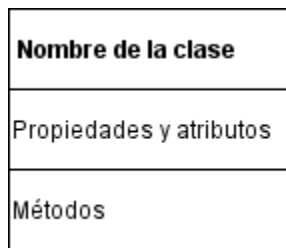
Figure 1. Esquema de los diagramas UML y su clasificación. Fuente: El Autor.

A continuación, se explican los diagramas UML que son usados en la metodología ICONIX, los cuales son: diagrama de clase, diagrama de secuencia y

diagrama de casos de uso. El diagrama de análisis de robustez también es un diagrama usado en la metodología ICONIX, pero no hace parte del estándar UML.

#### 2.1.7.1. Diagrama de Clases.

En UML el diagrama de clases es el diagrama más comúnmente usado, y la base principal de toda solución orientada a objetos. El diagrama describe las clases dentro de un sistema, sus atributos y operaciones, y la relación entre cada clase. Las clases se agrupan para crear diagramas, al crear diagramas de sistemas grandes<sup>19</sup>. A continuación, se presenta un ejemplo de un diagrama de clase:



*Figure 2. Formato de un diagrama de clase. Fuente: El autor. Software VioletUMLEditor.*

La ilustración describe a un diagrama de clase en general. De acuerdo a Schumuller<sup>20</sup>, un diagrama de clase está compuesto de un nombre, el cual describe el propósito de la clase y su categoría, por unas propiedades y atributos los cuales son

---

<sup>19</sup> LUCID SOFTWARE INC. ¿Qué es el lenguaje unificado de modelado (UML)? [En línea]. [Citado agosto 02 de 2018] Disponible en internet: < <https://www.lucidchart.com/pages/es/qu%C3%A9-es-el-lenguaje-unificado-de-modelado-uml> >

<sup>20</sup> SCHMULLER, Joseph. Aprendiendo UML en 24 horas. México: Prentice Hall, s.f. 27 p



propios de la clase y sirven para describirla, y unos métodos los cuales describen las funciones y acciones que están relacionadas con la clase.

Las clases en un diagrama de clases están relacionadas entre sí. En UML se usan varias formas gráficas para denotar tales asociaciones. A continuación, se describen estas asociaciones<sup>21</sup>:

- **Herencia.** Es una asociación que describe que cierta clase es una generalización o especialización de cierta categoría de clases. Por ejemplo, la clase hija en este caso hereda los atributos y la funcionalidad de la clase padre.
- **Asociación.** Es una asociación que define que las dos clases están relacionadas de alguna manera.
- **Agregación.** Es una asociación que describe que una de las clases pertenece a otra, es decir, que es una de las partes que conforman la clase. Esto significa que la asociación representa a una clase que está compuesta de una colección de clases. Se denota por una línea con un rombo en uno de sus extremos el cual se encuentra adjunto a una de las clases. La otra clase en este caso representa una de las partes de la primera clase mencionada.
- **Composición.** Es una asociación que define una variación de la asociación de agregación. La composición establece que una clase es un compuesto de otra clase, y debido a esto, solo existe como parte de la clase a la cual está asociada como compuesto.

---

<sup>21</sup> ATHURALIYA, Amanda. A Guide to UML Class Diagram Relationships [En línea]. [Citado 02 de agosto de 2018]. LINKEDIN, 24 de mayo de 2016. Disponible en internet: <  
<https://www.linkedin.com/pulse/guide-uml-class-diagram-relationships-amanda-athuraliya> >

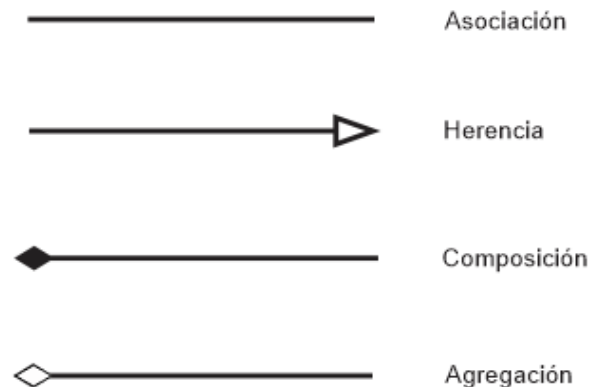


Figure 3. Representación gráfica de las relaciones entre los diagramas de clases.

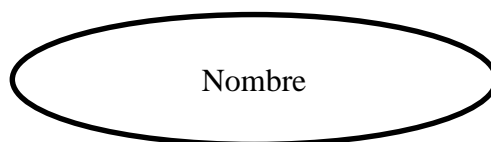
Fuente: El autor. Software Corel Draw X7.

El anterior gráfico denota como se representan gráficamente las relaciones de asociación entre las clases de un diagrama de clases. Las relaciones entre las clases también incluyen lo que se conoce como multiplicidad lo cual indica el número de instancias de una clase que están enlazadas con una o más instancias de la otra clase. Teniendo como ejemplo dos clases A y B, las relaciones de multiplicidad son las siguientes:

- **Uno a Uno.** Define una relación en la que una instancia de la clase A, esta solo asociada con una instancia de la clase B.
- **Uno a Muchos.** Define una relación en la que una instancia de la clase A, está asociada a dos o más instancias de la clase B.
- **Muchos a Muchos.** Define una relación en la que dos o más instancias de la clase A, están asociadas con dos o más instancias de la clase B.

### 2.1.7.2. Diagrama de Caso de Uso.

Un diagrama de caso de uso, es la representación gráfica de los requerimientos funcionales del sistema, es el modelo de la funcionalidad que se espera del sistema y su entorno. Los casos de uso permiten que se relacione lo que se necesita del sistema y como el sistema responde a tales necesidades<sup>22</sup>.



*Figure 4. Formato de un Caso de Uso. Fuente: El autor.*

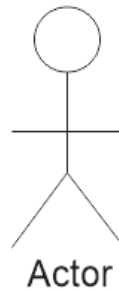
La ilustración anterior muestra que un caso de uso se denota por una elipse, y dentro de ella se escribe el nombre del caso de uso. Los casos de uso llevan por nombre una acción o verbo que debe realizar el sistema, como, por ejemplo, registrar paciente, o consultar factura.

Un caso de uso también menciona a un actor, el cual representa a un rol que interactúa con el sistema. Este actor puede ser una persona o también otro sistema que puede asumir un rol cuando se está realizando alguna acción. Un actor se representa por medio de un dibujo de un “hombre palo” o “stick man”. En caso de ser un sistema normalmente se representa con el dibujo de una computadora.

---

<sup>22</sup> VISUAL PARADIGM. What is use case diagram? [En línea]. [Citado 02 de agosto de 2018]. Disponible en internet: < <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/> >

Ya que para cada caso de uso puede existir uno o más actores, a cada uno de los actores se les debe asignar un nombre el cual describe el rol que cumplen dentro del sistema. La siguiente ilustración muestra la representación gráfica de un actor:



*Figure 5. Representación gráfica de un actor en un caso de uso. Fuente: El autor.*

*Software Corel Draw X7.*

Dentro de los diagramas de caso de uso también se manejan relaciones, la siguiente ilustración muestra una representación gráfica de las relaciones usadas para un caso de uso:



*Figure 6. Representación gráfica de las relaciones usadas en un caso de uso. Fuente: El autor. Software Corel Draw X7.*

Las relaciones se describen de la siguiente manera<sup>23</sup>:

- **Relación de Asociación.** Es la asociación más básica y se usa para asociar a los actores con los casos de uso.
- **Relación Include.** La asociación include, es usada para denotar que el comportamiento de un caso de uso incluye el comportamiento de otro caso de uso. Es decir, una relación include desde un caso de uso A, hacia un caso de uso B, indica que una instancia del caso de uso A debe también implicar que exista también una instancia del caso de uso B.
- **Relación Extends.** La asociación extends, desde una instancia de un caso de uso A hacia un caso de uso B, indica que las relaciones que realiza una instancia de B pueden ser extendida por el comportamiento de las acciones de una instancia de A, pero solamente cuando la instancia del caso de uso B, cumpla ciertas condiciones. En otras palabras, significa que el caso de uso A se instanciara opcionalmente ya que solo se instancia cuando se cumplen ciertas condiciones.

#### 2.1.7.2.1. Caso de Uso en formato expandido.

Un caso de uso en formato expandido es una descripción más detallada de los diferentes procesos que ocurren para llevar a cabo una tarea que hace parte de la funcionalidad de un sistema. Los casos de uso se construyen a partir de un caso de uso de más alto nivel.

---

<sup>23</sup> UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA. Diagramas de casos de uso. [En línea]. Colombia. [Citado 15 de agosto de 2018]. Disponible en internet: < [http://stadium.unad.edu.co/ovas/10596\\_9839/diagramas\\_de\\_casos\\_de\\_uso.html](http://stadium.unad.edu.co/ovas/10596_9839/diagramas_de_casos_de_uso.html) >

Un caso de uso en formato expandido está compuesto de una sección de encabezado, el cual contiene el nombre del caso de uso, los actores, y una breve descripción. La otra sección es el cuerpo del caso de uso y describe los eventos típicos y los eventos alternos<sup>24</sup>.

#### 2.1.7.3. Diagrama de Secuencia.

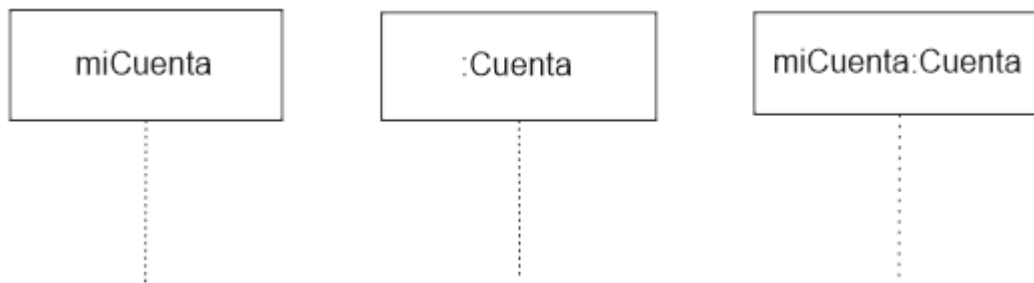
Un diagrama de secuencias<sup>25</sup>, es un diagrama de interacción el cual describe a los objetos y clases que están involucrados en un determinado escenario, especificado por un caso de uso y los mensajes que se intercambian entre los objetos de tal forma que la funcionalidad de aquel escenario se pueda llevar a cabo. Los diagramas de secuencia se elaboran para describir de esta manera a un caso de uso y su flujo de eventos tanto principales como alternativos. En los diagramas de secuencia se usan una serie de gráficos para dibujarlos.

Un objeto en un diagrama de secuencia de UML, se representa dibujado como un rectángulo el cual contiene el nombre del objeto. Los objetos se pueden nombrar de tres formas: el nombre del objeto, el nombre del objeto y su clase, o simplemente el nombre de la clase, esto es, un objeto anónimo. Las líneas de tiempo o de vida, se denotan por una línea vertical intercalada la cual se dibuja debajo de cada uno de los objetos, y se usan para denotar la existencia del objeto durante cierto periodo de tiempo.

---

<sup>24</sup> THE COMPUTER TECHNOLOGY DOCUMENTATION PROJECT. UML expanded use case. [En línea]. [Citado 15 de agosto de 2018]. Disponible en internet: < <http://www.comtechdoc.org/independent/uml/begin/umlexusecase.html> >

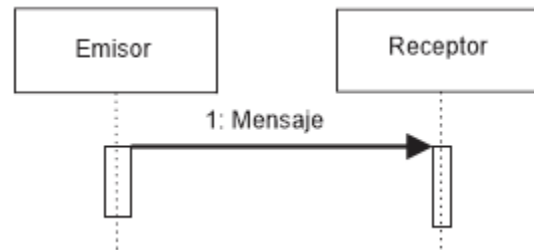
<sup>25</sup> VISUAL PARADIGM. Sequence diagram [En línea]. [Citado 15 de agosto de 2018]. Disponible en internet: < <https://www.visual-paradigm.com/learning/handbooks/software-design-handbook/sequence-diagram.jsp> >



*Figure 7. Representación gráfica de un objeto y su línea de vida en un diagrama de secuencia. Fuente: El autor. Software Corel Draw X7.*

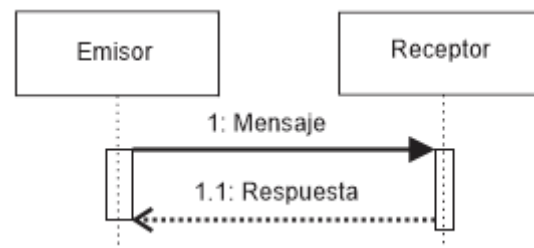
En la ilustración anterior, se muestra la representación gráfica de un objeto dentro de un diagrama de secuencia. El primer rectángulo es nombrado solo como el objeto, el segundo representa a un objeto anónimo y el tercero es un objeto con su nombre y la clase a partir del cual se instancia. Las entidades que participan en un escenario definido por un caso de uso, se organizan de manera horizontal como se puede ver en la ilustración.

En los diagramas de secuencia, también se usan mensajes, los cuales denotan la invocación de operaciones y son mostradas de manera horizontal. Se dibujan desde el emisor hasta el receptor. El orden se indica por la posición en la que se encuentra de manera vertical el mensaje, siendo el primero el que se encuentra en la parte superior del diagrama. A continuación, se muestra los tipos de mensajes en la ilustración:



*Figure 8. Mensaje síncrono en un diagrama de secuencia. Fuente: El autor. Software Corel Draw X7.*

La anterior ilustración muestra un mensaje de tipo síncrono. Se muestra como una línea sólida con una punta de flecha llena. Es un mensaje usado para comunicaciones normales entre el emisor y el receptor. Un mensaje síncrono es una operación que espera una respuesta para poder seguir ejecutándose.

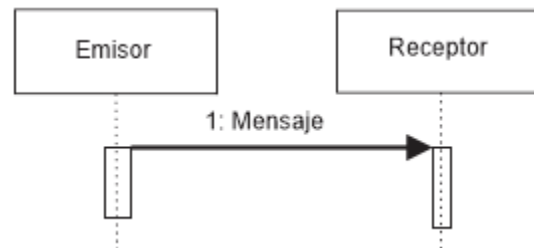


*Figure 9. Mensaje de respuesta en un diagrama de secuencia. Fuente: El autor. Software Corel Draw X7.*

La anterior ilustración muestra un mensaje de retorno o respuesta, la cual se muestra como una línea intercalada con una flecha abierta. Es un mensaje usado para comunicaciones normales entre el emisor y el receptor. Un mensaje de retorno es una



operación que tiene un mensaje de ida, y a cambio obtiene su respuesta para poder seguir ejecutándose.



*Figure 10. Mensaje asíncrono en un diagrama de secuencia. Fuente: El autor. Software Corel Draw X7.*

La anterior ilustración muestra un mensaje de tipo asíncrono. Se muestra con una línea sólida con una punta de flecha abierta. Un mensaje asíncrono es una operación que se ejecuta, pero sin esperar respuesta para seguir su ejecución.

#### 2.1.7.4. Diagrama de Robustez.

Un diagrama de robustez es un diagrama que consiste en una especie de combinación entre un diagrama de clases y un diagrama de actividades. Es una representación del comportamiento descrito por un caso de uso, ya que muestra las clases que participan y su comportamiento. Un diagrama de robustez muestra el flujo de acciones que se llevan a cabo para un caso de uso en términos de objetos<sup>26</sup>. El flujo de acciones se indica por una línea entre los objetos la cual posee una punta en flecha que indica el sentido.

---

<sup>26</sup> ROSENBERG, Doug y STEPHENS, Matt. Use Case Driven Object Modeling with UML. Theory and Practice. 1 ed. EE. UU: Apress, 2007, 103 p.

En la siguiente ilustración se muestran los diferentes símbolos usados para dibujar un diagrama de robustez:

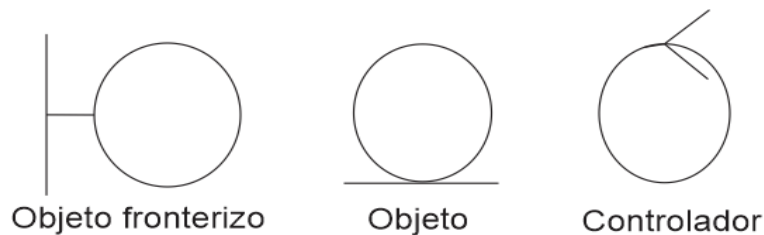


Figure 11. Símbolos de un diagrama de robustez. Fuente: El autor. Software Corel Draw X7.

Los diagramas de robustez usan tres tipos de estereotipos para las clases, los cuales son:

- **Objetos fronterizos.** Representan la interfaz entre el sistema y el mundo. Los objetos fronterizos son típicamente páginas web, ventanas graficas o pantallas.
- **Objetos entidad.** Representan las clases que se modelaron en el modelo de dominio.
- **Controladores.** Son las operaciones entre los objetos entidad y los objetos fronterizos que describen las acciones que se llevan a cabo durante el flujo de eventos.

Resulta útil pensar los objetos fronterizos y los objetos entidades como sustantivos, y de los controladores como verbos. Para dibujar el diagrama de robustez, es bueno seguir las siguientes recomendaciones: los sustantivos pueden hablar a los verbos (y viceversa), los sustantivos no pueden hablar con otros sustantivos y los verbos pueden hablar con otros verbos.

### 2.1.8. Aplicación WEB

Una aplicación web, es una aplicación informática que está alojada en un servidor remoto que permite a los usuarios hacer uso de la misma por medio de un navegador web a través de internet o de una intranet<sup>27</sup>.

### 2.1.9. PHP

PHP, por sus siglas en inglés Hypertext Pre-processor, es un lenguaje de programación de propósito general, de código abierto ampliamente usado que es especialmente adecuado para el desarrollo web y puede ser incrustado en HTML. PHP es normalmente usado como lenguaje de programación del lado del servidor, esto es, el código se ejecuta sobre el servidor, generando HTML el cual es enviado al cliente. El cliente recibe los resultados de ejecutar el script (archivo de ordines), pero no se dan cuenta cual es el código subyacente<sup>28</sup>.

### 2.1.10. Laravel

Laravel es un framework para el desarrollo de aplicaciones web del lado del servidor. Se describe en su página web oficial como “elegante y expresivo”. Sigue el patrón de arquitectura Modelo – Vista - Controlador, el cual permite a los desarrolladores realizar de manera fácil y ágil las tareas comunes usadas en la mayoría de los proyectos

---

<sup>27</sup> ECURED. Aplicación web [En línea]. Cuba. Oficina nacional para la informatización. s.f. Rev. 13 de septiembre de 2013. [citado agosto 02 de 2018] Disponible en internet: <[https://www.ecured.cu/Aplicaci%C3%B3n\\_web](https://www.ecured.cu/Aplicaci%C3%B3n_web) >

<sup>28</sup> THE PHP GROUP. What is PHP? [En línea]. [Citado 2 de agosto de 2018]. Disponible en internet: <<http://php.net/manual/en/intro-what-is.php>>

web, es de código abierto, esta licenciado bajo los términos de la licencia MIT, y es uno de los frameworks de PHP más populares.

Laravel es un producto de inteligencia colectiva, compuesta de muchas librerías robustas y herramientas hechas por otros desarrolladores, lo cual ayuda a reducir muchas de las tareas repetitivas y complejas a las que a veces se enfrentan los desarrolladores, lo que quiere decir que los desarrolladores pueden con menos código hacer un buen trabajo, y hacer más con buena calidad en cierto tiempo dado.

Laravel también es un framework que viene equipado con herramientas de seguridad, para de esta manera asegurar la aplicación de posibles amenazas. No requiere que los desarrolladores posean un profundo y vasto conocimiento de seguridad. De esta manera, Laravel tiene soporte de funcionalidades básicas para prevenir vulnerabilidades de seguridad, como la inyección SQL.

Laravel también viene equipado con el patrón arquitectónico Modelo – Vista – Controlador, para de esta manera permitir organizar mejor el código<sup>29</sup>.

#### 2.1.11. Servidor WEB

Un servidor web es un sistema que entrega contenido o servicios a usuarios finales a través de internet. Un servidor consiste de un servidor físico, un sistema operativo servidor y un software usado para facilitar la comunicación por medio del protocolo HTTP.

---

<sup>29</sup> TAYLOR OTWELL. Introduction [En línea]. [Citado 2 de agosto de 2018]. Disponible en internet: <<https://laravel.com/docs/4.2/introduction>>

El trabajo principal de un servidor web, es mostrar el contenido del sitio web. Si un servidor no está disponible de forma pública y es usado internamente, entonces se le llama un servidor e intranet.

Cuando alguien solicita un sitio web añadiendo el URL o dirección web, como Google Chrome o Firefox, el navegador envía una solicitud a través de la red para poder verla página web correspondiente para esa dirección IP. El servidor web es solicitado para que presente el contenido del sitio web al navegador del usuario<sup>30</sup>.

#### 2.1.12. Bases de Datos

Una Base de Datos, es una colección organizada de datos. Muchas bases de datos contienen múltiples tablas, las cuales incluyen varios campos distintos. Por ejemplo, una base de datos de una compañía puede incluir tablas para sus registros sobre productos, empleados y estados financieros. Cada una de estas tablas tendría campos diferentes que son relevantes para la información almacenada de la tabla.

Los datos son organizados en filas, columnas y tablas, y están indexados para hacer más fácil encontrar información relevante. Los datos son actualizados, expandidos y eliminados conforme información nueva es añadida. Las bases de datos procesan funciones para actualizarse a sí mismas, realizando búsquedas en los datos que contiene y ejecutando aplicaciones sobre los mismos.

---

<sup>30</sup> MOZILLA DEVELOPER NETWORK. ¿Qué es un servidor WEB? [En línea]. [Citado 2 de agosto de 2018]. Disponible en internet: <[https://developer.mozilla.org/es/docs/Learn/Common\\_questions/Que\\_es\\_un\\_servidor\\_WEB](https://developer.mozilla.org/es/docs/Learn/Common_questions/Que_es_un_servidor_WEB)>

Para administrar la base de datos, se usa un DBMS (Sistema de gestión de base de datos), que es un tipo de software que permite definir, manipular, obtener y manejar datos almacenados dentro de una base de datos. Un RDBMS (Sistema de gestión de base de datos) es un tipo de software de gestión de base de datos que fue desarrollado en los años 1970, basado en el modelo relacional, y es todavía la forma más popular de gestionar una base de datos<sup>31</sup>.

## 2.2. Estado del Arte.

A continuación, se hace un resumen de la literatura revisada en lo referente a proyectos de desarrollo de software cuyo objetivo ha sido el desarrollo de un sistema de información web para apoyar al bienestar institucional, al bienestar universitario o a la facultad de Ciencias de la Salud de cierta institución. También se revisaron proyectos donde se haya hecho uso de la metodología de desarrollo ICONIX de la tecnología Laravel.

*Table 1. Cuadro de resumen del Estado del Arte. Fuente: El autor.*

<i>Título</i>	<b>Metodología de Desarrollo</b>	<b>Framework</b>	<b>Tecnología</b>
Implementación del sistema de área de Salud Sidasa para el	No especifica	Ninguno	PHP, JQuery, Pchart

<sup>31</sup> TECHTARGET. Database (DB) [En línea]. [Citado 3 de agosto de 2018]. Disponible en internet: <<https://searchsqlserver.techtarget.com/definition/database> >

departamento de bienestar estudiantil de la Upse: módulos odontología y psicología.			
Sistema integrado de gestión de servicios para las áreas de salud y psicología de la división de bienestar estudiantil de la institución educativa Magdalena Ortega de la Unión V.	Modelo “Lineal secuencial”	Codeigniter	PHP
Diseño e implementación de un sistema de automatización de procesos para la Dirección de Bienestar Universitario (DBU) de la Universidad Central del Ecuador.	XP	.NET	C#
Desarrollo e implementación de un sistema de gestión de talento humano y bienestar estudiantil para el instituto superior cordillera.	No especifica	Ninguno	PHP
Implementación de una solución web y móvil para la gestión vehicular basada en Arquitectura de Aspectos y metodologías ágiles: Un enfoque educativo de la teoría a la práctica.	Iconix	Ruby on rails	Ruby
Desarrollo de un sistema web para	No especificado	Laravel	PHP,

la gestión de imágenes y álbumes			Javascript, JQuery, Bootstrap.
Plataforma web para colecciones botánicas.	ICONIX	Laravel	PHP, JQuery, Bootstrap
Desarrollo de una aplicación Web Móvil para optimizar las consultas de infracciones de tránsito en la gerencia de transporte y tránsito de la Municipalidad Provincial del Santa	ICONIX	Laravel	AngularJS, PHP
Money transfer management system	No especificado	Laravel	PHP, Bootstrap, JQuery
Implementación del servicio web 'calificación de jugadores en línea para la Asociación de Fútbol Amateur de Pichincha (A.F.A.P.) utilizando la metodología Iconix.	ICONIX	Ninguno	PHP
Desarrollo de un Módulo para la Gestión Bibliográfica en Moodle	ICONIX	Ninguno	PHP
Establishment of web interface for tensiomyography measurements display.	No especificado	Laravel	PHP



Event Management System	No especificado	Laravel	PHP, Bootstrap, JQuery
The Development of Mobile Client Application in Yogyakarta Tourism and Culinary Information System Based on Social Media Integration	ICONIX	.NET	Visual Basic .NET
Diseño de una aplicación windows para la gestión de corretaje inmobiliario de la empresa G & S inmobiliaria utilizando la metodología ICONIX y tecnología windows presentation foundation	ICONIX	Windows Presentation Foundation	C++, C#

Como se puede ver en el recuadro anterior, existen proyectos previamente elaborados donde se han desarrollado una aplicación web para apoyar las labores de una oficina de bienestar estudiantil, y proyectos donde se ha usado la metodología de desarrollo ICONIX o donde se ha usado la tecnología Laravel.

Los cuatro proyectos desarrollados para apoyar las labores de una oficina de bienestar estudiantil<sup>32</sup>, representan el 26% de los proyectos revisados. De estos podemos evidenciar el apoyo que pueden brindar las TIC's en la gestión de pacientes, ya que lo que se busca es facilitar a los usuarios encargados el hacer un seguimiento de los

---

<sup>32</sup> AVELLA MARTINEZ, Laura Yaneth y PARRA RUIZ, Paola Patricia. Tecnologías de la información y la comunicación (TICS) en el sector salud. [En línea]. 2013. Disponible en internet: < <http://bdigital.unal.edu.co/11172/1/laurayanethavellamartinez.2013.pdf> >

pacientes, y también de los diferentes medicamentos, equipos e insumos que hacen parte del inventario.

De esta manera, los usuarios podrán obtener datos para la toma de decisiones relacionadas tanto con los pacientes como con el inventario de medicamentos, de insumos médicos y de equipos médicos.

Por otro lado, vemos que la metodología de desarrollo ICONIX es usada en un 46.6% de los proyectos revisados, siendo la metodología más usada dentro de esto. ICONIX es una metodología ágil<sup>33</sup> la cual se adapta a muchos tipos de proyectos. Hace uso de solamente 4 diagramas UML buscando de esta manera hacer que el desarrollador no se pierda en detalles innecesarios.

ICONIX hace uso del análisis de robustez para reducir la ambigüedad en las descripciones de los casos de uso y para proporcionar una manera de pasar de la etapa de análisis a la etapa de diseño, ya que se analizan los casos de uso y se identifican los objetos para cada caso de uso. Además, ICONIX funciona bien en tornos ágiles y busca siempre que cada componente que se ha obtenido a partir del análisis concuerde con los componentes del diseño, para de esta manera evitar confusiones.

Por último, se evidencia que el 40% de los proyectos hacen uso de la tecnología Laravel, siendo el framework más usado dentro de los proyectos revisados. Laravel una tecnología que se ha vuelto muy popular en poco tiempo gracias a sus diferentes funcionalidades, haciendo que la mayoría de las tareas comunes y repetitivas puedan realizarse más fácilmente, tareas que con recurrentes en muchos proyectos. Es de código

---

<sup>33</sup> MOSTAFA, Ashour A. N. A review of different agile methods. [En línea]. 2016. Disponible en internet: < [https://www.researchgate.net/publication/301346660\\_Five\\_different\\_agile\\_methods](https://www.researchgate.net/publication/301346660_Five_different_agile_methods) >

abierto y totalmente libre para ser usado. Además, hace uso del patrón de diseño MVC (Modelo – Vista – Controlador) para de esta manera permitir trabajar de forma individual en cada uno de los componentes de la aplicación. Incorpora además elementos caracterizados por brindar un conjunto de herramientas para la seguridad de la aplicación, para realizar pruebas unitarias, para generar datos de prueba, para gestionar la autenticación de usuarios, entre otras cosas<sup>34</sup>.

Comparando los proyectos, podemos ver que el 66% hacen uso del lenguaje de programación PHP, el 13.3% de C# y el 21% usan otros lenguajes de programación. En cuanto a herramientas para el diseño de interfaz gráfica del lado del cliente, el 33% hacen uso de la librería JQuery, y el 26% hacen uso de la librería Bootstrap, siendo ambas herramientas más usadas dentro de los proyectos.

En comparación con el proyecto de desarrollo del presente trabajo, podemos decir que el 13% hacen uso tanto de la metodología ICONIX como del framework Laravel. La diferencia radica en que uno de estos proyectos hace uso de AngularJS como framework del lado del cliente, mientras que el proyecto del presente trabajo no usa algún framework para tal objetivo, sino que usa la librería JQuery.

Comparando el proyecto del presente trabajo con los cuatro proyectos revisados cuyo objetivo es apoyar las labores de oficinas de bienestar estudiantil, podemos mencionar que el primero de estos proyectos está enfocado al área de psicología y odontología, buscando crear un sistema de gestión de fichas de pacientes. El segundo es

---

<sup>34</sup> DOVZHENKO, Oleksii. Why Laravel will become most popular PHP MVC framework in 2018 [En línea]. LINKEDIN, 2018. [citado 25 de agosto de 2018] Disponible en internet: <  
<https://www.linkedin.com/pulse/why-laravel-become-most-popular-php-mvc-framework-2018-dovzhenko>  
>

un proyecto que optimiza la prestación de servicios en el área de salud y psicología, al desarrollar un sistema para administrar los datos de los pacientes. El tercero es un proyecto donde se automatizan los procesos relacionados con becas estudiantiles, trabajo social, seguros, psicología y orientación profesional. Finalmente, el cuarto proyecto es un desarrollo de una aplicación para el seguimiento de todos los alumnos en cuanto a asesorías y orientación estudiantil, como también para apoyar el proceso de evaluación a docentes nuevos.

Así, se puede ver que el proyecto del presente trabajo se enfoca en apoyar la labor de las oficinas de enfermería del bienestar universitario de la facultad de Ciencias de la Salud de la fundación Universitaria Juan Corpas, buscando llevar un control de inventario de los diferentes medicamentos, insumos y equipo médicos, algo que no es mencionado en los otros proyectos. Además, busca llevar un control de los pacientes que acuden a los servicios ofrecidos por las oficinas de enfermería, llevando un registro de los datos en formato electrónico de los diferentes procesos realizados tales como registro de inyectología, entrega de medicamentos, entre otros, algo que si está presente en los otros proyectos siendo bastante similar.

Por otro lado, también busca apoyar el control de vacunas de estudiantes practicantes de la facultad de Ciencias de la Salud, algo en lo que difiere con los otros proyectos. Así mismo, busca apoyar las oficinas de enfermería en lo relacionado con la gestión de historias clínicas ocupacionales, para de esta manera permitir llevar un registro de las mismas en formato electrónico. Por último, se puede señalar que uno de los propósitos del presente proyecto se fundamenta en el plan “Cero Papel”, estipulado por el

Gobierno Nacional en su plan de desarrollo, que busca reducir el consumo de papel sustituyendo el papel por soportes electrónicos para de esta manera favorecer el medio ambiente y aumentar la eficiencia administrativa.

### **2.3. Marco Contextual.**

#### 2.3.1. Fundación Universitaria Juan N. Corpas.

La Fundación Universitaria Juan N. Corpas, es una institución pública de la educación superior que sus actividades el 3 de febrero de 1971, obra concebida por su Fundador el DOCTOR JORGE PIÑEROS CORPAS, en diciembre de 1970 como un proyecto de beneficio social donde el bienestar común, la igualdad de derechos sin ningún tipo de discriminación y el respeto por la vida son considerados pilares fundamentales<sup>35</sup>.

La Fundación Universitaria Juan N. Corpas implementa el concepto de Calidad de Vida como filosofía institucional, que abarca a toda la comunidad universitaria procurando su bienestar integral y promoviendo su desarrollo.

---

<sup>35</sup> Fundación Universitaria JUAN N. CORPAS [Citado 9 agosto de 2018]. Disponible en internet < <http://www.juanncorpas.edu.co/la-institucion/historia/>>

### 2.3.2. Bienestar Universitario.

Bienestar es el conjunto de políticas y programas que buscan el desarrollo integral de cada uno de los miembros de la comunidad Corpista, el mejoramiento de su calidad de vida y en consecuencia el mejoramiento de la Fundación Universitaria Juan N. Corpas<sup>36</sup>.

Basado en la Ley 30/92: " Bienestar universitario orienta el desarrollo físico, psicoactivo, espiritual y social de los miembros de la comunidad académica en las áreas de salud, recreación y deportes, desarrollo humano, promoción socio económica, cultural, ética y ecológica, a través del conjunto de políticas y programas encaminados a alcanzar la formación integral y el mejoramiento de la calidad de vida de la comunidad de la Escuela mediante acciones intencionalmente formativas "

Bienestar aporta al proceso educativo mediante acciones intencionalmente formativas que permitan el desarrollo del ser humano en todas sus dimensiones a partir del trabajo y de la reflexión sobre las áreas de salud, recreación y deportes, desarrollo humano y la promoción socio-económica, cultural y ecológica.

### 2.3.3. Visión.

La Escuela de Medicina de la Fundación Universitaria Juan N. Corpas, es una entidad académica de Educación Superior en los niveles de Pregrado y Postgrado, que inspirada en el espíritu de nuestro Fundador, el Dr. Jorge Piñeros Corpas y en su visión social del ejercicio de la profesión, integra el profundo estudio de la ciencia médica con

---

<sup>36</sup> Fundación Universitaria JUAN N. CORPAS [Citado 9 agosto de 2018]. Disponible en internet < <http://www.juanncorpas.edu.co/bienestar/>

la práctica dirigida y la investigación científica, y en su proyecto educativo correlaciona las diferentes vertientes de aproximación a la problemática de la salud del país.

Nuestra Misión es Formar Médicos - Cirujanos Generales de excelencia profesional y humana, con énfasis en Medicina Familiar, visión sistémica y bio-psico-social, dentro de un marco de uso pertinente de la investigación, la ciencia y la tecnología con profundo enfoque social.

Para el cumplimiento de nuestra Misión, además del recurso humano, la Facultad de Medicina cuenta con recursos científicos y físicos que integran en el alumno los conocimientos teóricos y prácticos, y lo preparan activamente para el desarrollo de su profesión, y que incluyen:

- Un campus universitario
- Una institución hospitalaria con servicios y equipos médicos especializados
- Campos de práctica anatomoclínica y quirúrgica
- Convenios interinstitucionales a nivel nacional e internacional
- Un Laboratorio-Departamento de Ciencias Básicas, Laboratorios de investigación en Farmacología Vegetal, Departamento de Informática y otros.<sup>37</sup>

#### 2.3.4. Misión.

Ser reconocida tanto en el ámbito nacional como internacional por:

- La excelencia en la formación de sus egresados

---

<sup>37</sup> Fundación Universitaria JUAN N. CORPAS [Citado 9 agosto de 2018]. Disponible en internet <<http://www.juanncorpas.edu.co/oferta-academica/pregrado/medicina/mision-y-vision/>>

- El liderazgo en la investigación e implementación en soluciones alternativas e innovadoras frente a los problemas de salud que afectan a la población en general.
- La implementación de la Atención Primaria y de la Medicina Familiar como modelos de gestión en salud que contribuyan efectivamente al mejoramiento de la calidad de vida dentro del criterio de desarrollo social integral.<sup>38</sup>.

### 2.3.5. Propósitos.

Con el fin de cumplir con los objetivos plasmados en su misión y visión, la Fundación Universitaria pretende<sup>39</sup>:

- Generar un ambiente académico que gire en torno a la búsqueda y apropiación del conocimiento en todas sus manifestaciones culturales, históricas, científicas y sociales, y que se compromete a proteger, cuidar y utilizar apropiadamente los recursos naturales del país, fomentando el amor a la ecología.
- Velar por el diseño y actualización de Planes Curriculares que respondan a las necesidades sociales, científicas, culturales y económicas del país, buscando un desarrollo profesional específico, idóneo y aplicable a las necesidades permanentes de su entorno.
- Revisar y optimizar permanentemente los procesos de evaluación, con el fin de generar criterios claros y definidos, con transparencia, coherencia y pertinencia en torno al logro de los objetivos y competencias esperados en los educandos.

---

<sup>38</sup> Fundación Universitaria JUAN N. CORPAS [Citado 9 agosto de 2018]. Disponible en internet <<http://www.juanncorpas.edu.co/oferta-academica/pregrado/medicina/mision-y-vision/>>

<sup>39</sup> Fundación Universitaria JUAN N. CORPAS [Citado 9 agosto de 2018]. Disponible en internet <<http://www.juanncorpas.edu.co/la-institucion/propositos/>>



- Generar mecanismos que faciliten la continuidad en los procesos de capacitación docente dentro de las nuevas concepciones pedagógicas y las exigencias de la globalización.
- Definir los perfiles, condiciones y procesos que permitan una adecuada selección estudiantil, siendo coherentes con los postulados de la misión y visión institucional.
- Generar procesos que permitan la revisión permanente y la actualización de las políticas para la educación continuada y el apoyo y seguimiento a los egresados, como una medida de mejoramiento de la calidad y de fortalecimiento de las relaciones de la institución con la sociedad.
- Optimizar los mecanismos que garanticen el continuo mejoramiento y adecuación de la infraestructura y dotación de biblioteca, laboratorios, medios audiovisuales, sistemas de informática y procesos de producción, edición y publicación de materiales educativos de todas las áreas académicas con el fin de estimular y facilitar la búsqueda permanente del conocimiento.
- Fortalecer el Bienestar Universitario de la Fundación, para que, a través de sus diferentes actividades, se favorezca el desarrollo personal de los miembros de la comunidad universitaria, complementando su proyecto de vida.
- Hacer de la gestión académico-administrativa una actividad dinámica que facilite el desarrollo de los procesos institucionales de manera eficiente y efectiva, velando por la construcción de mecanismos ágiles, con el fin de propiciar una

óptima calidad de servicio y un eficiente apoyo a la docencia, bienestar y proyección social.

- Promover en los diversos programas académicos la capacitación de los estudiantes en investigación formativa, como instrumento fundamental para la adquisición del juicio crítico y de las capacidades de análisis, interpretación, síntesis, argumentación y proposición, necesarias para el desarrollo del pensamiento científico.
- Generar estrategias académicas que busquen la excelencia en todas sus disciplinas, incluyendo los conocimientos administrativos y de gestión propios a cada área del saber.
- Fortalecer los medios de comunicación al interior de la Comunidad Académica y al exterior de la misma a nivel local, nacional e internacional, de tal manera que le permita a cada uno de los miembros participar activamente del quehacer institucional e interinstitucional y proyectarse académicamente dentro de las instituciones que homologan su actividad educativa.
- Complementar el desarrollo personal propiciando una formación integral, entendiendo el ser humano como una persona en la que se conjugan los aspectos biológicos, psicológicos, sociales y espirituales, considerándolo dentro de un contexto en el que se relacionan los saberes con la práctica.
- Apoyar la interacción de las artes con la ciencia y con todas las manifestaciones culturales, que conlleve al desarrollo individual y colectivo, y que genere una repercusión social consecuente con la participación de la Fundación en el proceso

continuo de transformación que requiere el país en la búsqueda de la integración con las culturas del mundo.

- Desarrollar cada vez más un criterio de interdisciplinariedad, flexibilidad, autonomía y apertura a todos los saberes universales.

#### 2.3.6. Marco Legal.

De acuerdo a las actividades realizadas por el programa de salud del Bienestar Universitario, se debe cumplir con la siguiente normatividad:

- **Resolución 1995 de 1999.** Por el cual se establecen normas para el manejo de la Historia Clínica.
- **Resolución 2346 de 2007.** Por la cual se regula la práctica de evaluaciones médicas ocupacionales y el manejo y contenido de las historias clínicas ocupacionales.
- **Ley estatutaria 1581 de 2012.** Por la cual se dictan disposiciones generales para la protección de datos personales.

## **Capítulo 3.**

### **Ingeniería del Proyecto.**

#### **3.1. Metodología de Desarrollo.**

Para el desarrollo de la aplicación web, se estableció usar la metodología de desarrollo de software llamada ICONIX. A continuación, se describe el desarrollo de cada uno de los procesos de ICONIX durante el desarrollo de la aplicación.

#### **3.2. Análisis de Requerimientos.**

A continuación, se describe los pasos que se llevaron a cabo durante el análisis de requerimientos.

##### **3.2.1. Captura de Requerimientos.**

La captura de requerimientos consiste en definir lo que el sistema debería ser capaz de realizar. Para la captura de requerimientos se realizaron entrevistas a las enfermeras y los médicos de las oficinas del servicio de enfermería, y así mismo se realizaron entrevistas a los funcionarios de la facultad de Ciencias de la Salud encargados de gestionar el control de vacunas de estudiantes practicantes de los programas de enfermería y medicina. En la siguiente ilustración se muestra un ejemplo de un requerimiento funcional que se definió para el desarrollo de la aplicación:

Table 2. Requerimiento N° 1 del sistema. Fuente: El autor.

<b>Requerimiento No.</b>	<b>1</b>
<b>Módulo</b>	Control del inventario de medicamentos e insumos
<b>Usuario</b>	Enfermero
<b>Descripción del requerimiento</b>	<p>El sistema debe permitir hacer un registro de entrega de medicamento a un estudiante, docente, funcionario egresados u otros (personas que se encuentran dentro de la institución pero que no pertenecen a ella) en un formulario llamado Registro de administración o entrega de medicamentos, especificando los siguientes datos:</p> <ul style="list-style-type: none"> <li>• Fecha</li> <li>• Nombre</li> <li>• Apellido</li> <li>• Teléfono</li> <li>• Usuario (E: estudiante, F: Funcionario, Eg: Egresado, O: Otros, D: Docente)</li> <li>• Código</li> <li>• CC – DI</li> <li>• Programa</li> <li>• Semestre</li> <li>• Medicamento administrado</li> <li>• Lote</li> <li>• Cantidad</li> <li>• Firma del usuario</li> </ul>

	Durante la elección del medicamento, el sistema mostrará las características del mismo.
--	---

### 3.2.2. Modelo de Dominio.

Después de hacer una lista de los requerimientos, se procede a elaborar el modelo de dominio inicial, el cual es modificado conforme se avanza en el proyecto. La elaboración de un modelo de dominio inicial antes de la elaboración de los casos de uso ayuda a reducir la ambigüedad, ya que cuando se escriben los casos de uso, tienden a ser más abstractos, algo vagos y ambiguos.

Par evitar esto, la metodología ICONIX sugiere que los casos de uso tienen que ser ajustados a la realidad, es decir, los casos de uso deben referenciar los objetos del dominio por su nombre. De esta manera se pretende unir las partes estáticas y dinámicas del modelo.

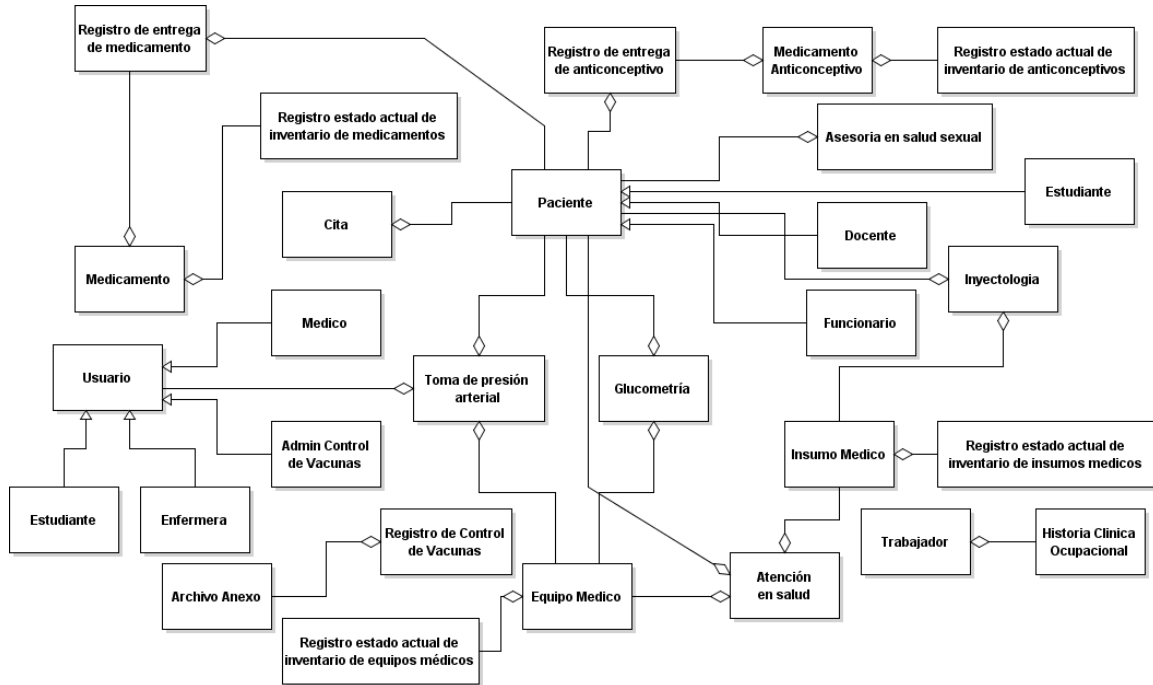


Figure 12. Modelo de dominio del sistema. Fuente: El autor. Software Violet UML Editor.

En la ilustración anterior el modelo de dominio se elabora de acuerdo a lo especificado en los requerimientos. En este caso se modela como un diagrama de clases, pero sin sus atributos ni métodos.

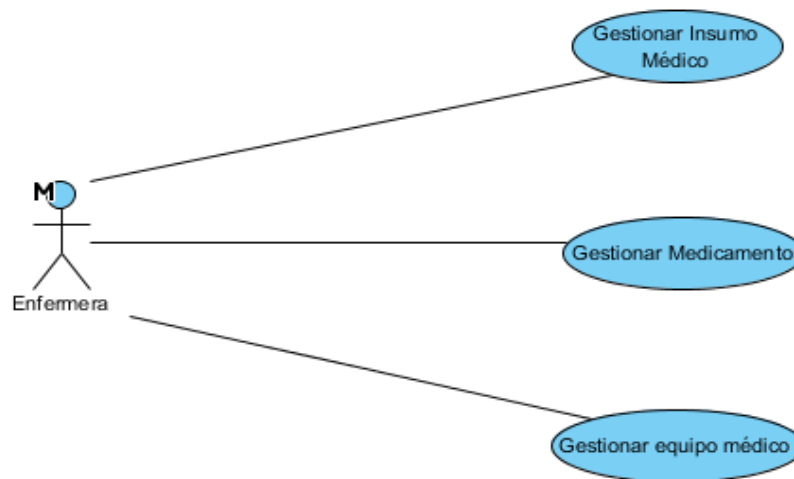
### 3.2.3. Diagramas de Casos de Uso.

El siguiente paso es la realización de los casos de uso, los cuales describen en termino resumidos la funcionalidad que debe cumplir el sistema. En los diagramas de casos de uso los usuarios que se definieron para el sistema fueron:

- Enfermera.
- Medico.

- Estudiante.
- Usuario control de vacunas.

La siguiente ilustración muestra el diagrama de caso de uso para el módulo de control de inventario de medicamentos.



*Figure 13. Diagrama de caso de uso para el módulo de control de inventario. Fuente: El autor. Software Visual Paradigm.*



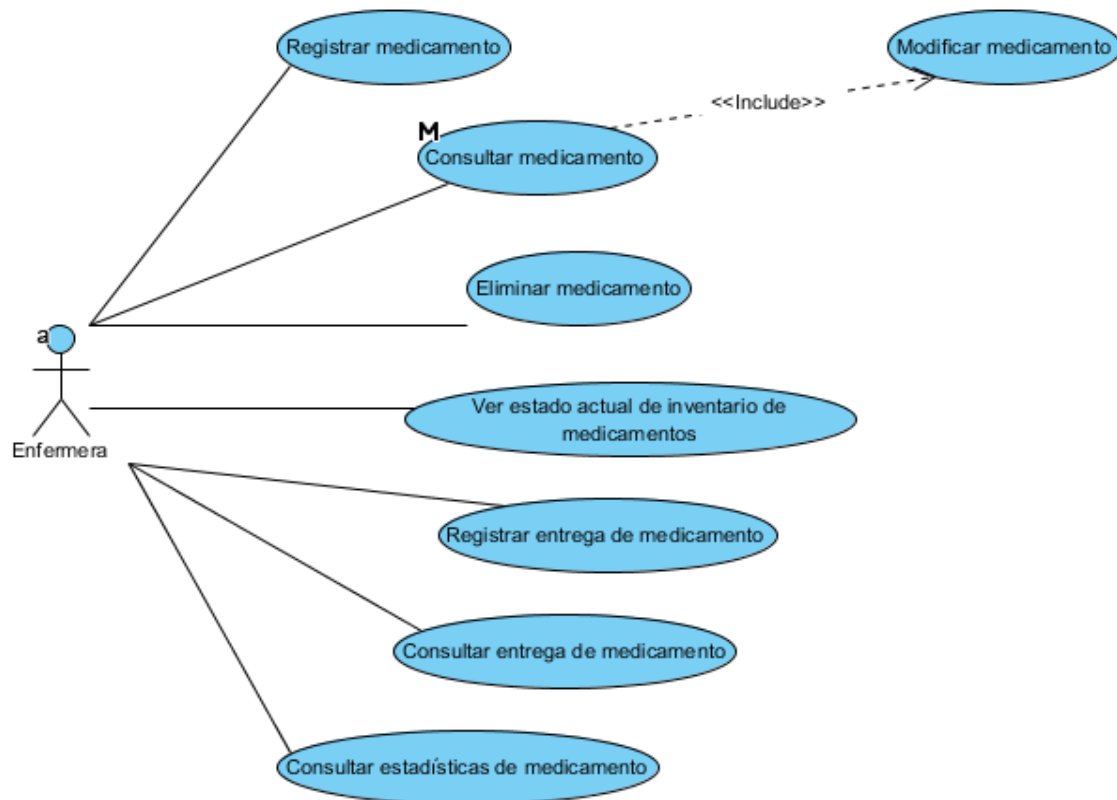


Figure 14. Detalle caso de uso gestionar medicamento. Fuente: El autor. Software Visual Paradigm.

La primera de las dos ilustraciones anteriores, muestra los casos de uso de manera general del módulo de control de inventarios. El segundo diagrama muestra en detalle en que consiste el caso de uso llamado gestionar medicamento, el cual está compuesto por varias funcionalidades, como el registrar un medicamento o consultar este último.

#### 3.2.4. Casos de Uso en Formato Expandido.

En esta etapa se elaboran los casos de uso en formato expandido, los cuales describen de manera más detallada las actividades que conforman un caso de uso en

particular, es decir, paso a paso, junto con sus vías alternativas. A continuación, se muestra el caso de uso en formato expandido para el registro de medicamento:

*Table 3. Casos de uso registrar medicamento en formato expandido. Fuente: El autor.*

<b>Caso de Uso</b>	Registrar Medicamento
<b>Actores</b>	Enfermera
<b>Propósito</b>	Registrar en la base de datos un nuevo medicamento y sus datos
<b>Tipo</b>	Principal
<b>Descripción</b>	Una enfermera realiza el registro de un nuevo medicamento en la base de datos del sistema.
<b>Referencias cruzadas</b>	
<b>Curso normal de los eventos</b>	
<b>Acción de los actores</b>	<b>Respuesta del sistema</b>
1. La enfermera selecciona la opción para registrar un medicamento del menú.	El sistema despliega un formulario en blanco para el registro de un medicamento.
2. La enfermera llena el formulario y lo envía al sistema.	El sistema registra el nuevo medicamento y envía un mensaje de confirmación de registro exitoso.
<b>Cursos alternativos</b>	
Acción 2: Si la enfermera no llena los campos obligatorios, el sistema valida y notifica al usuario de los campos que debe llenar en el formulario y no realiza el registro de medicamento.	

En la ilustración anterior, se puede ver en formato expandido para el caso de uso registrar medicamento, es descrito más detalladamente, mostrando cada paso de manera secuencial y también las vías alternativas.

### 3.3. Análisis y Diseño Preliminar.

En esta etapa del proceso, se realiza un diagrama de robustez para cada uno de los casos de uso definidos previamente. El diagrama de robustez es una visualización de los pasos que ocurren para cada caso de uso. Mientras se realiza un diagrama de robustez, se vuelve a escribir el caso de uso correspondiente.

Así mismo, se actualiza el modelo de dominio mientras se vuelve a escribir el caso de uso y se dibuja el diagrama de robustez. Haciendo esto se pueden descubrir clases nuevas, corregir ambigüedades y se añaden atributos a los objetos de dominio.

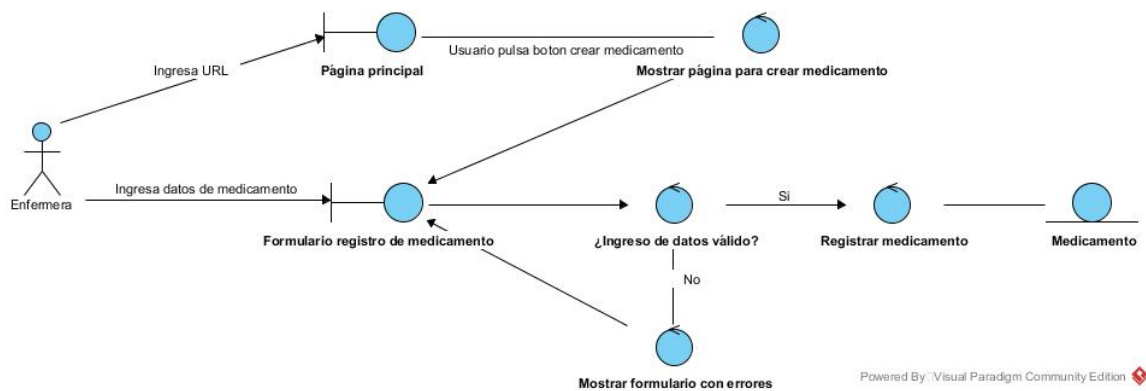


Figure 15. Diagrama de robustez para el caso de uso registrar medicamento. Fuente: El autor. Software Visual Paradigm Community Edition.

En este caso, el diagrama de robustez muestra los diferentes elementos involucrados en el caso de uso. La secuencia muestra que primero se ingresa una URL que lleva a la página principal de la aplicación.

Al hacer clic en el botón para crear un nuevo medicamento, un controlador se encarga de mostrar al usuario el formulario de registro de medicamento. El usuario luego procede a llenar el formulario y enviarlo. Un controlador se encarga de validar los datos, y si se ha llenado correctamente el formulario, se registra el medicamento, sino, se muestra un mensaje de error al usuario.

### **3.4. Diseño Detallado.**

En esta etapa del proceso, se procede a realizar un diagrama de secuencias para cada caso de uso. Como se mencionó antes, los diagramas de secuencia sirven para describir a los objetos y clases que están involucrados en un determinado escenario especificado por un caso de uso.

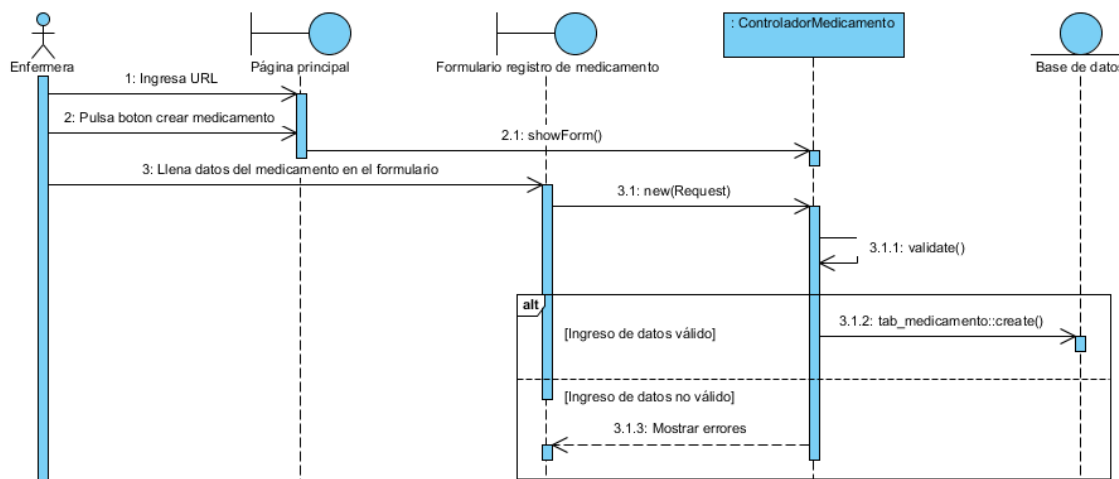


Figure 16. Diagrama de secuencia para el caso de uso - Registrar Medicamento. Fuente: El autor. Software Visual Paradigm Community Edition.

El diagrama de secuencia de la ilustración anterior muestra cómo se asigna los nombres a los métodos ejecutados por el controlador llamado ControladorMedicamento. Al hacer click en crear medicamento, se ejecuta un método llamado showForm() del objeto ControladorMedicamento, el cual muestra al usuario el formulario de registro.

Luego el usuario procede a llenar el formulario y enviarlo. Una vez se envía, se ejecuta el método new, el cual recibe unos parámetros. El controlador valida los datos enviados por medio del método validate() y luego decide si guarda el medicamento por medio del método tab\_medimento::create() o si decide mostrar los errores al usuario.

### 3.5. Implementación.

A continuación, se describe la implementación de la aplicación

### 3.5.1. Diagrama de Clases.

El diagrama de clases se obtiene a partir del modelo de dominio, durante la fase de diseño detallado. Al haber llegado a la etapa de implementación, el modelo de dominio se ha convertido en un diagrama de clases, ya que, en la etapa del diseño detallado, como se mencionó anteriormente, al elaborar los diagramas de secuencia se añaden operaciones a los objetos del modelo de dominio.

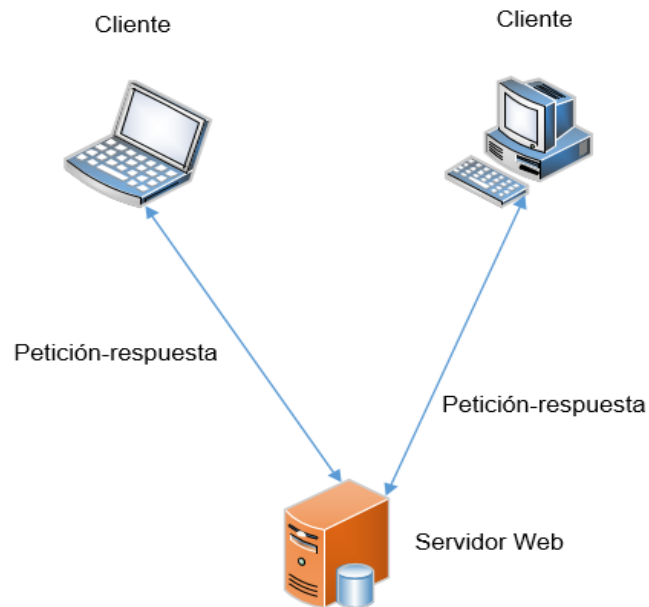
### 3.5.2. Diagrama Entidad – Relación.

El diagrama entidad – relación se obtiene a partir del diagrama de clases obtenido durante las agregaciones y modificaciones que se hacen al modelo de dominio el cual se convierte en un diagrama de clases conforme se avanza en el proyecto.

Los diagramas de clases contienen los atributos necesarios para generar el modelo de la base de datos. El diagrama entidad – relación, se elaboró en el software MySQL Workbench.

### 3.5.3. Arquitectura de Software.

Con los previos pasos anteriores realizados, el siguiente paso es la identificación de la arquitectura del sistema web que se implementara. En este caso, la arquitectura cliente servidor es usada para la implementación de la aplicación web en la red y se ilustra en el siguiente diagrama:



*Figure 17. Diagrama de la arquitectura cliente - servidor. Fuente: El autor. Software Visio Profesional.*

En este escenario, los clientes realizan peticiones al servidor web, el cual es el encargado de entregar y administrar los recursos y servicios que son usados por el cliente. Los servicios en este caso serían los ofrecido por el servidor web el cual aloja la aplicación y se encargara de recibir las peticiones de los clientes y procesar las acciones que se realicen al hacer uso de la aplicación retornando una respuesta a la petición.

Este servidor maneja Apache como servidor web para alojar la aplicación y hace uso del servidor de base de datos MySQL. Los clientes acceden a los recursos por medio de un navegador web, realizan peticiones al servidor y este procesa la petición y devuelve una respuesta, como por ejemplo una página HTML. El siguiente diagrama de componentes muestra el patrón de arquitectura y diseño modelo – vista – controlador de la aplicación:

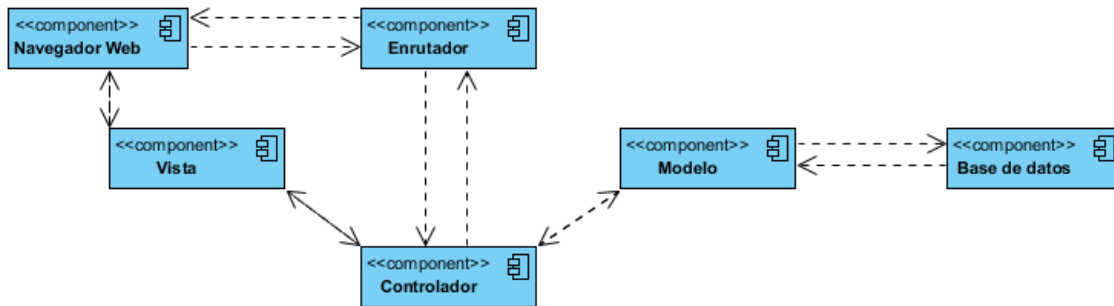


Figure 18. Diagrama de componentes del modelo - vista - controlador de Laravel en la aplicación. Fuente: El autor. Software Visual Paradigm.

El diagrama de componentes anterior muestra el patrón de arquitectura y diseño modelo – vista – controlador. En este caso, el cliente desde su navegador web realiza una petición HTTP a la aplicación. Laravel usa lo que llama enrutadores para definir las reglas que rigen como se accede a los servicios ofrecidos por la aplicación. Los enrutadores notifican a los controladores como debe responder la aplicación de acuerdo al tipo de petición que el cliente hace, y luego el controlador se encarga de enviar la respuesta adecuada al cliente.

El controlador si es necesario, notificara a uno o varios modelos de que necesita los datos correspondientes alojados en la base de datos y luego envía estos, a la vista para ser mostrada en el navegador del cliente. La vista es la página HTML que se muestra al cliente en el navegador web.

El siguiente diagrama de despliegue muestra la estructura arquitectónica de la aplicación y sus respectivos componentes, donde también se puede apreciar el patrón de diseño modelo – vista – controlador de Laravel.



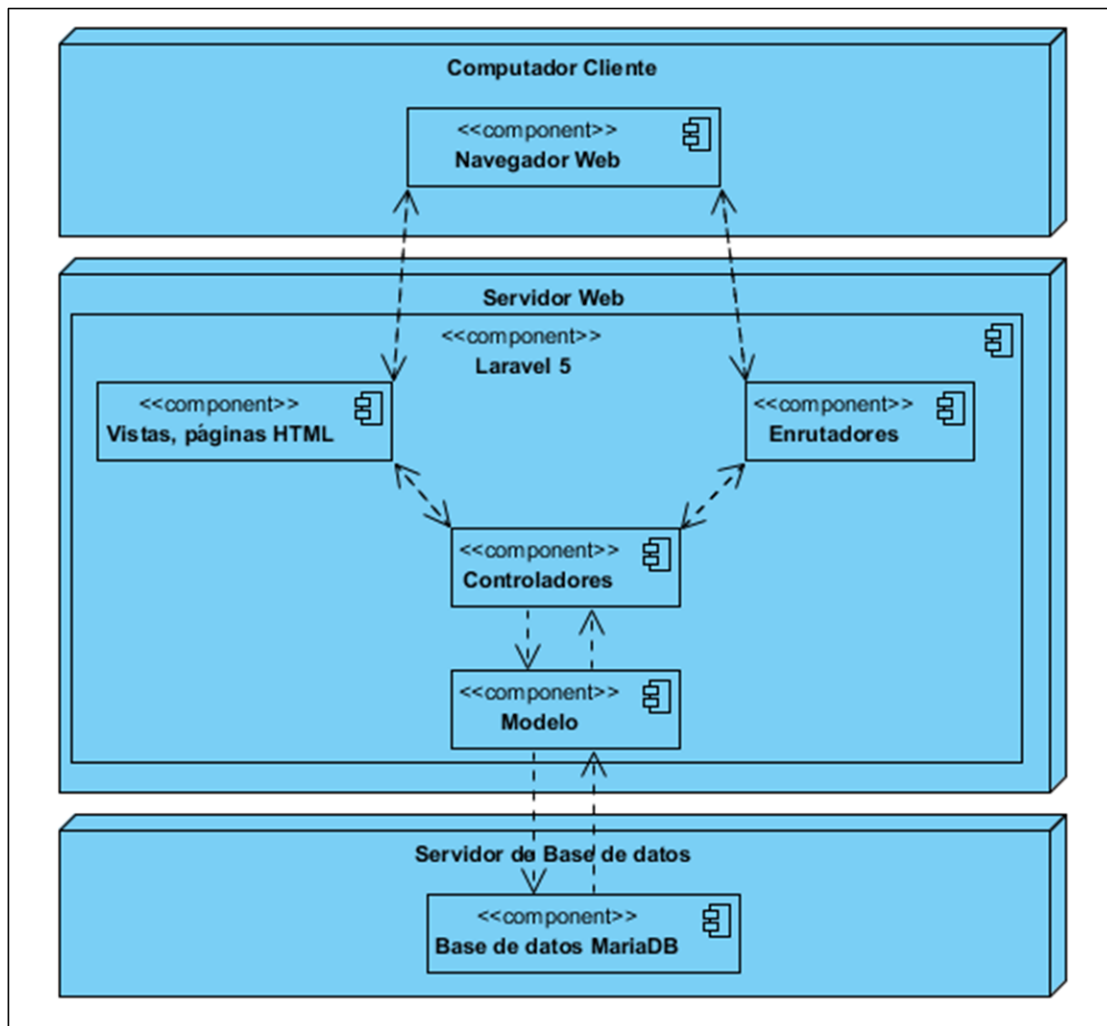


Figure 19. Diagrama de despliegue de la aplicación. Fuente: El autor. Software Visual Paradigm.

Como se puede ver, el diagrama muestra que la arquitectura de la aplicación está compuesta por los nodos clientes, el nodo donde se aloja el servidor web y el servidor de base de datos.

Dentro de estos nodos tenemos a los diferentes componentes que conforman la estructura de la arquitectura de la aplicación. El servidor web está compuesto por el servidor Apache, y la aplicación alojada con sus componentes principales, es decir, los enrutadores, las vistas, los controladores y los modelos implementados usando Laravel. El cliente es el que hace uso del navegador web para realizar peticiones al servidor, y el servidor de base de datos está representado por el servidor MySQL.

#### 3.5.4. Código Fuente.

A continuación, se representa una vista de la aplicación y se mostrara en detalle como el código fuente de la aplicación opera para lograr la funcionalidad para la que fue elaborada.

En la siguiente vista se puede observar la página de administración de medicamentos, desde aquí se puede visualizar una tabla con la lista de los medicamentos registrados y su respectiva información. Aparecen además otras opciones. Para crear un medicamento, se debe seleccionar la opción llamada “Crear medicamento”.

**Medicamentos**

Home > Medicamentos

Lista de Medicamentos [+ Crear Medicamento](#)

Mostrar 10 registros Buscar:

Id	Nombre	Laboratorio	Fecha de Vencimiento	Existencia	Presentacion	Lote	INVIMA	Acciones
1	Acetaminofen	MK	2018-08-22	50	Tabletas	621D-21	OI-4FGN	<a href="#">✎</a> <a href="#">✉</a>
2	Ibuprofeno	MK	2018-06-03	80	Pastas	AN31D-21	DKW-A3231	<a href="#">✎</a> <a href="#">✉</a>
3	Buscapina	Boehringer Ingelheim	2018-06-27	93	Tabletas	JFHR-27	JDELMAX2	<a href="#">✎</a> <a href="#">✉</a>
4	Naproxeno	Bayer	2018-06-27	100	Tabletas	KZXWQ-82	JDELMAX2	<a href="#">✎</a> <a href="#">✉</a>
5	Nytax	Genfar	2018-06-27	100	Tabletas	OPTR-53	JDELMAX2	<a href="#">✎</a> <a href="#">✉</a>
6	Cetirizina	Boehringer Ingelheim	2019-03-20	80	Pastillas	ERIQP-45	NU5NAS	<a href="#">✎</a> <a href="#">✉</a>

Mostrando registros del 1 al 6 de un total de 6 registros Anterior **1** Siguiente

Consultar estadísticas para un medicamento

Medicamento:  Desde:  Hasta:

Figure 20. Vista que muestra el listado de medicamentos registrados. Fuente: El autor.

A continuación, se muestra el formulario de registro de medicamentos, el cual debe llenarse con los datos correspondientes, como lo son el nombre, laboratorio, fecha de vencimiento, existencia en el inventario, presentación, entre otros datos:

**Crear Medicamento**

**Nombre**  
Diclofenaco 50 mg

**Laboratorio**  
American generics

**Fecha de Vencimiento:**  
05/14/2019

**Existencia**  
90

**Presentacion**  
Tabletas

**Lote**  
UIWADL-WHQ

**Registro INVIMA**  
UR-OAP1236

**Observaciones**  
Medicamento antiinflamatorio no esteroideo. Se utiliza para reducir la inflamación y para tratar el dolor.

Figure 21. vista del formulario de registro de medicamento. Fuente: El autor.

Después de llenar los datos, se presiona en la opción “Crear” para almacenar el medicamento y su información en la base de datos.

**Medicamentos**

Lista de Medicamentos

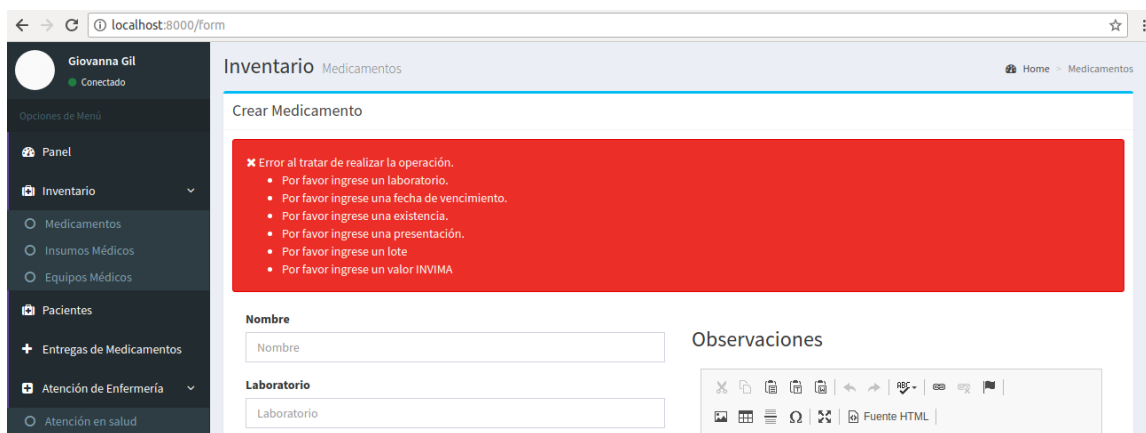
✓ El medicamento ha sido agregado exitosamente

Mostrar: 10 registros

Id	Nombre	Laboratorio	Fecha de Vencimiento	Existencia	Presentacion	Lote	INVIMA	Acciones
1	Acetaminofen	MK	2018-08-22	50	Tabletas	621D-21	OI-4FGN	
2	Ibuprofeno	MK	2018-06-03	80	Pastas	AN31D-21	DKW-A3231	
3	Buscapina	Boehringer Ingelheim	2018-06-27	93	Tabletas	JFHR-27	JDELMAX2	

Figure 22. Mensaje de éxito al crear un nuevo medicamento. Fuente: El autor.

Como se puede observar, en la lista de medicamentos ahora aparece creado el medicamento en cuestión, el Diclofenaco, junto con su información. En caso de que después de intentar almacenar el medicamento de la base de datos, haya algún error, el sistema arroja un error en pantalla, notificando el error.



*Figure 23. Formulario de registro de medicamentos con mensaje de error. Fuente: El autor.*

En la parte superior del formulario, aparecen los errores en cuestión resaltados en rojo.

El código fuente detrás del funcionamiento de los procesos que se acaba de mostrar se explicara a continuación. Primero, el formulario de registro de medicamento este hecho usando las plantillas de blade del framework Laravel, las cuales usan los lenguajes HTML y PHP para la generación de las páginas. También se hizo uso de la librería Bootstrap para dar estilo a los distintos elementos que se presentan en las páginas

de la aplicación, y la librería de Javascript JQuery para añadirle funcionalidades a las páginas.

```

1  @extends('base.app')
2
3  @section('migas')
4  <h1>
5  Inventario
6  <small>Medicamentos</small>
7  </h1>
8  <ol class="breadcrumb">
9  <li><a href="{{ route('Home') }}"><i class="fa fa-dashboard"></i> Home</a></li>
10 <li class="active">Medicamentos</li>
11 </ol>
12 @endsection
13
14 @section('contenido')
15
16 <div class="col-md-12">
17   <div class="box box-info">
18     <div class="box-header with-border">
19       <h3 class="box-title">Crear Medicamento</h3>
20     </div>
21     <!-- /.box-header -->
22     <!-- form start -->
23
24     <div class="box-body">
25       <form role="form" action="{{ route('create_medicamento') }}" method="POST">
26
27         <input type="hidden" name="_token" value="{{ csrf_token() }}">
28
29         @if ($errors->any())
30
31         <div class="alert alert-danger" role="alert">
32           <i class="fa fa-close" aria-hidden="true"></i>
33           Error al tratar de realizar la operación.
34           <ul>
35             @foreach ($errors->all() as $error)
36               <li>{{ $error }}</li>
37             @endforeach
38           </ul>
39         </div>
40         @endif
41
42         <div class="col-md-6">
43           <div class="form-group">
44             <label>Nombre</label>
45             <input type="text" class="form-control" id="nombre" name="nombre" placeholder="Nombre">
46           </div>
47           <div class="form-group">
48             <label>Laboratorio</label>
49             <input type="text" class="form-control" id="laboratorio" name="laboratorio" placeholder="Laboratorio">
50           </div>

```

Figure 24. Código fuente vista de registro de medicamento. Parte 1. Fuente: El autor.

En la ilustración anterior, podemos ver código HTML con PHP incrustado. Dado a que se está usando el motor Blade de Laravel, se hace uso de unas expresiones para realizar ciertas funcionalidades, como por ejemplo `@extends('base.app')` que lo que hace es especificar que una vista hereda de otra vista padre. En este caso, la vista actual hereda de una vista llamada `app.blade.php` ubicada en la carpeta `base`.

La expresión `@section('migas')` y `@section('contenido')` definen contenido el cual es puesto en la sección de la vista donde especifica la expresión. Las secciones terminan con la expresión `@endsection`. Otro punto a tener en cuenta es que en Blade, las variables que se pasan a la vista se muestran entre llaves. Blade también permite escribir expresiones de estructuras de control como lo son `if`, `for`, `foreach`, los cuales en el código se ven anteceditas por un `@`.

La función `route` es una función de Laravel la cual obtiene la ruta del controlador que procesara los datos enviados en el formulario. En este caso la ruta se llama `'create_medicamento'` y es una ruta definida previamente usando Laravel.

```

51 <!-- Date -->
52 <div class="form-group">
53 <label>Fecha de Vencimiento:</label>
54 <div class="input-group date">
55 <div class="input-group-addon">
56 <i class="fa fa-calendar"></i>
57 </div>
58 <input type="text" class="form-control" id="fecha_vencimiento" name="fecha_vencimiento" placeholder="
01/01/2000">
59 </div>
60 <!-- /input group -->
61 </div>
62 <!-- /form_group -->
63 <div class="form-group">
64 <label>Existencia</label>
65 <input type="number" class="form-control" id="existencia" name="existencia" placeholder="Cantidad" min="1">
66 </div>
67 <div class="form-group">
68 <label>Presentacion</label>
69 <input type="text" class="form-control" id="presentacion" name="presentacion" placeholder="Jarabe, Pastas">
70 </div>
71 <div class="form-group">
72 <label>Lote</label>
73 <input type="text" class="form-control" id="lote" name="lote" placeholder="">
74 </div>
75 <div class="form-group">
76 <label>Registro INVIMA</label>
77 <input type="text" class="form-control" id="invima" name="invima" placeholder="">
78 </div>
79
80 </div>
81 <div class="col-md-6">
82 <h3 class="box-title">Observaciones
83 </h3>
84 <div class="box-body pad">
85 <textarea id="observaciones" name="observaciones" rows="10" cols="80">
86 Descripción
87 </textarea>
88 </div>
89 </div>
90 <!-- /.box -->
91 <!-- /.box-body -->
92 <div class="col-md-12 box-footer">
93 <a class="btn btn-warning" href="{{ route('view_medicamentos') }}">Cancelar</a>
94 <button type="submit" class="btn btn-info pull-right">Crear</button>
95 </div>
96 <!-- /.box-footer -->
97 </form>
98 </div>
99

```

Figure 25. Código fuente vista de registro de medicamento. Parte 2. Fuente: El autor.

En la ilustración vemos código HTML, elementos junto con sus atributos respectivos, algunos de los cuales nos permiten dar estilo y organización al contenido usando Bootstrap.

```
98         </div>
99     </div>
100 </div>
101 </div>
102
103 @endsection
104
105 @section('js')
106
107 <script type="text/javascript">
108
109     $(function () {
110         CKEDITOR.replace('observaciones')
111         //bootstrap WYSIHTML5 - text editor
112         $('#textarea').wysihtml5()
113     })
114     //Date picker
115     $('#fecha_vencimiento').datepicker({
116         autoclose: true
117     });
118 </script>
119
120 @endsection
```

Figure 26. Código fuente vista de registro de medicamento. Parte 3. Fuente: El autor.

En la ilustración anterior, vemos una sección definida como ‘js’, la cual define la sección donde se escribe código JavaScript para darle nuevas funcionalidades a la vista. En este caso, hacemos uso de librerías JavaScript para poder añadir un calendario y un editor de textos pequeño.

Los archivos que definen las vistas en la aplicación que serán visualizadas por el usuario, se definen como archivos con formato .php. Laravel viene con un motor de



plantillas llamado Blade, el cual añade algunas funcionalidades para facilitar la creación y manipulación de las vistas.

En la ilustración anterior, se puede ver que hay segmentos de código anteceditos con un símbolo de arroba (@), lo cual se usa para que el motor de plantillas de Blade sepa que se trata de sentencias de código php. Este código hace parte del motor de Blade. Después de esto se usa lenguaje HTML para definir los elementos necesarios para la creación del formulario, y así mismo se aplican los estilos necesarios a los elementos. Una vez un usuario llena el formulario de la vista y lo envía, el controlador encargado de procesar la acción se activa. A continuación, se muestra una ilustración de un fragmento de código del controlador:

```
public function new(Request $request){  
    $request->validate([  
        'nombre' => 'required',  
        'laboratorio' => 'required',  
        'fecha_vencimiento' => 'required',  
        'existencia' => 'required|numeric',  
        'presentacion' => 'required',  
        'lote' => 'required',  
        'invima' => 'required',  
    ],  
    [  
        'nombre.required' => 'Por favor ingrese un nombre.',  
        'laboratorio.required' => 'Por favor ingrese un laboratorio.',  
        'fecha_vencimiento.required' => 'Por favor ingrese una fecha de vencimiento.',  
        'existencia.required' => 'Por favor ingrese una existencia.',  
        'presentacion.required' => 'Por favor ingrese una presentación.',  
        'lote.required' => 'Por favor ingrese un lote',  
        'invima.required' => 'Por favor ingrese un valor INVIMA'  
    ]  
);  
}
```

Figure 27. Código fuente controlador de registro de medicamento. Parte 1. Fuente: El autor.

En la ilustración anterior, tenemos una función dentro del controlador definida como `new`, la cual recibe unos parámetros los cuales son los parámetros enviados al momento de enviar al formulario. Estos parámetros se pasan como parámetros de objeto tipo `Request`.

El controlador se encarga de primero realizar una validación de los parámetros recibidos usando la función `validate` del objeto, y si la validación encuentra errores notifica al usuario de estos. En la función `validate` se pasa un arreglo en formato llave – valor. Por ejemplo, “nombre” => “required” significa que el elemento llamado “nombre” debe estar definido, esto es, que el usuario haya llenado el campo nombre en el formulario. El segundo arreglo funciona de manera similar, con la diferencia de que aquí se definen los mensajes que se deben mostrar da validación al usuario. La siguiente ilustración muestra la sección de código faltante de la función `new` del controlador:

```

$date = date_create($request->fecha_vencimiento);
$date = date_format($date, 'Y-m-d');

|

    tab_medicamento::create([
        'nombre'           => $request->nombre,
        'laboratorio'      => $request->laboratorio,
        'fecha_vencimiento' => $date,
        'existencia'       => $request->existencia,
        'presentacion'     => $request->presentacion,
        'lote'              => $request->lote,
        'invima'           => $request->invima,
        'observaciones'    => $request->observaciones,
    ]);

    return redirect()->route('view_medicamentos')->with('exito', 'El medicamento ha sido
    agregado exitosamente');
    //return Redirect::to(route('view_medicamentos'));

}

```

Figure 28. Código fuente controlador de registro de medicamento. Parte 2. Fuente: El autor.

La ilustración muestra a una variable llamada \$date, la cual está recibiendo el valor del atributo fecha\_vencimiento de la petición después de que su valor se usa para crear un objeto de tipo date. Después, la misma variable es convertida al formato de fechas que se indica en el código.

Luego, el controlador hace uso de la clase modelo de medicamento, en este caso llamada tab\_medicamento, para realizar la creación del nuevo registro en la base de datos, de esta manera siguiendo el patrón de diseño modelo – vista – controlador, como se indicó anteriormente. La clase modelo hace uso de su método create, pasando un arreglo de parámetros en formato llave – valor, en este caso son los atributos que se obtuvieron del objeto Request enviados desde el formulario, para almacenar el medicamento en la base de datos.

Finalmente, se retorna al usuario a la vista del panel de administración de medicamentos, usando el método ‘to’ de la clase Redirect que usa para redireccionar a cierta ruta definida en el archivo web.php que define las rutas de la aplicación.

## Capítulo 4.

### Conclusiones y Recomendaciones.

#### 4.1. Conclusiones.

- Fue importante y dispendioso aplicar todos los conocimientos adquiridos en el estudio de la ingeniería de sistemas, donde se pudo analizar las actividades que se realizan en el programa de salud de las oficinas del bienestar universitario y las actividades que se realizan para llevar a cabo el control de vacunas de los estudiantes de la facultad de Ciencias de la Salud, para lograr así identificar el problema.

En este proceso se identificó los siguientes problemas:

- Demora al averiguar el número de veces que ha sido entregado cierto medicamento en un rango de fechas.
- Demora al averiguar el número de veces que un paciente ha recibido cierto tipo de atención en enfermería.
- Demora al averiguar el número de veces que se ha realizado cierto tipo de atención en enfermería por programa en un rango de fechas.
- Demora al revisar el inventario de medicamentos e insumos médicos y sus respectivas existencias.
- Necesidad de formular un plan para promover el uso preferente de herramientas electrónicas para evitar en lo posible el uso y consumo de

papel para atender los lineamientos de la estrategia gubernamental “Cero Papel”.

- Reprocesamiento de datos al tener que ser revisados varias veces, demora al procesar los datos de control de vacunas de los estudiantes y organizar los mismos.
  - Demora al registrar una atención de enfermería
- El desarrollo de esta aplicación implicó investigar sobre la tecnología de desarrollo web Laravel. De esta manera se adquirieron nuevos conocimientos que se convierten en valor agregado a lo ya aprendido durante el transcurso de los estudios del programa de ingeniería de sistemas.

Laravel se usó como framework del lado del servidor, aprovechando su implementación del patrón de diseño – vista – controlador, logrando esta manera organiza mejor los componentes de la aplicación y tener un mejor control sobre estos a la hora de trabajar con ellos.

Dado el conjunto de funcionalidades que posee la tecnología Laravel, se agilizaron tareas que son comunes en el escenario de desarrollo de la aplicación web, tales como el procesamiento de datos, el control de rutas, y elementos básicos de seguridad, ahorrando y acortando tiempo de codificación.

Por otro lado, Laravel permitió trabajar fácilmente las operaciones CRUD en la base de datos, por medio de su componente Eloquent, y el motor de

plantillas llamado Blade, que facilito la generación de las vistas a mostrar en el lado del cliente, es decir, las páginas que componen la aplicación.

- Se desarrolló una aplicación web siguiendo la metodología de desarrollo ICONIX, por medio de la cual se definieron los requerimientos funcionales de la aplicación, se analizaron los requerimientos, se realizó un diseño de la aplicación y posteriormente se realizó la implementación de la misma.

ICONIX permitió desarrollar la aplicación usando solo cuatro tipos de diagramas, siendo tres del estándar UML y uno llamado diagrama de robustez.

#### **4.2. Recomendaciones.**

- Se recomienda que se continúe con el trabajo realizado hasta el momento en el desarrollo de la aplicación, donde se involucren activamente la facultad de Ciencias de la Salud, y así mismo las oficinas del programa del bienestar universitario. De esta manera, el proyecto podría crecer y optimizarse en cuanto a funcionalidades conforme sea necesario realizar correcciones y nuevas necesidades, ideas o sugerencias surjan a futuro.

Algunas de las funcionalidades que se recomiendan sean implementadas en el futuro son:

- Implementación del módulo de consultas externas.
- Implementación del módulo de historias clínicas de control de maternidad.

- Implementación de un lector biométrico o un dispositivo de firmas electrónicas como medio de constancias de atención en enfermería, atención médica y en la elaboración de historias clínicas ocupacionales y su respectivo concepto.
  - Explorar la implementación de mejoras en cuanto al módulo de estadísticas para generar reportes que aumenten el apoyo a las actividades de los usuarios.
- 
- Se recomienda que la aplicación sea alojada en un servidor que cuente con al menos 8 GB de RAM, un almacenamiento de 1 TB y con un poder de procesamiento de cuatro núcleos, para tratar de minimizar problemas de acceso por parte de usuarios, y problemas de rendimiento y almacenamiento de la aplicación web y el servidor. Es necesario y recomendable que el servidor tenga instalado una versión de PHP 7.0 ya que en esta versión se desarrolló la aplicación.
  - Se recomienda que se investigue otras tecnologías de desarrollo web que puedan ser implementadas conjuntamente con las ya implementadas en la aplicación desarrollada, para mejorar y optimizar las funcionalidades. Algunas de estas tecnologías con potencial de implementación son:
    - **AngularJS.** Framework desarrollado por Google para la creación de aplicaciones web del lado del cliente.

- **Vue.js.** Framework del lado del cliente para la creación de aplicaciones web.
- **ReactJS.** Framework desarrollado por Facebook.



### Lista de referencias

ASSOCIATES, M. &. (n.d.). *Agile Development with ICONIX process*. Retrieved agosto 2, 2018, from <http://www.methodsandtools.com/archive/archive.php?id=22>

BERNAL ACEVEDO, o. &. (2011). Sistemas de información en el sector salud en Colombia. *Revista Gerencia y políticas de Salud*. Retrieved 19 agosto, 2018, from [http://www.scielo.org.co/scielo.php?script=sci\\_arttext&pid=S1657-70272011000200006&lang=pt](http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S1657-70272011000200006&lang=pt)

Consejo Profesional Nacional de Arquitectura y sus Profesiones Auxiliares. (n.d.). *Politica cero papel del consejo profesional nacional de arquitectura y sus profesiones auxiliares - CPNAA*. Retrieved agosto 14, 2018, from <https://cpnaa.gov.co/sites/default/files/docs/poli%CC%81tica%20cero%20papel.pdf>

Gobierno de Colombia. (n.d.). *Departamento Nacional de Planeacion*. Retrieved agosto 14, 2018, from Plan Nacional de Desarrollo: <https://www.dnp.gov.co/Plan-Nacional-de-Desarrollo/PND-2010-2014/Paginas/Plan-Nacional-De-2010-2014.aspx>

Gobierno de Colombia. (n.d.). *Sistema Nacional de Informacion de la Educacion Superior - SNIES*. Retrieved agosto 14, 2018, from

<https://www.mineducacion.gov.co/sistemasdeinformacion/1735/w3-article-211884.html>

Ministerio de Tecnologías de la Información y las Comunicaciones. (s.f.). *Cero Papel en la administracion publica. Buenas practicas para reducir el consumo en la oficina.*

Recuperado el 14 de agosto de 2018, de [http://estrategia.gobiernoenlinea.gov.co/623/articles-8257\\_papel\\_buenaspracticass.pdf](http://estrategia.gobiernoenlinea.gov.co/623/articles-8257_papel_buenaspracticass.pdf)

OFFICE OF THE NATIONAL COORDINATOR FOR HEALTH INFORMATION TECHNOLOGY. (2018, marzo 21). *What is an electronic health record?* Retrieved agosto 25, 2018, from <https://www.healthit.gov/faq/what-electronic-health-record-ehr>

PRESSMAN, R. S. (2010). *Ingeniería de software. Un enfoque práctico.* Mexico: McGraw-Hill.

ROSENBERG, D. y. (2007). *Use Case Driven Object Modeling with UML. Theory and Practice.* EE.UU.: Apress.

WHITTEN, J. L. (n.d.). *ANÁLISIS DE SISTEMAS. DISEÑO Y MÉTODOS.* Mexico: Whitten Bentley. Retrieved agosto 14, 2018