

UNIVERSIDAD AMAZÓNICA DE PANDO
ÁREA DE CIENCIAS Y TECNOLOGÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

**“SISTEMA INFORMÁTICO DE DIAGNÓSTICO DE INSUFICIENCIA
RENAL EN LA UNIDAD DE ENFERMEDADES NO TRANSMISIBLES DEL
SEDES PANDO”**

**PROYECTO DE GRADO PRESENTADO PARA OBTENER EL TÍTULO
ACADÉMICO DE LICENCIADO EN INGENIERÍA DE SISTEMAS**

Postulante : Mary Lenny Sarzuri Nina
Tutor : MSc. Fredy Morales Blanco
Asesor : Ing. Juan Carlos Gallardo Jiménez

Cobija - Pando – Bolivia

2016

AGRADECIMIENTOS

En primer lugar, agradezco a Dios por haberme guiado, y cuidado en todo momento.

Agradecer a mis padres, de todo corazón por la comprensión y confianza a lo largo de mis estudios, a pesar de la distancia, me brindaron su apoyo.

A mis hermanos que siempre me brindaron su apoyo a seguir adelante.

Agradecer a mi tutor Ing. Msc Fredy Morales Blanco por el tiempo que ha dedicado en este trabajo y la paciencia que ha tenido.

A mi asesor Ing. Juan Carlos Gallardo Jiménez por brindarme su apoyo en el proceso de desarrollo desde el inicio del proyecto, al Lic. Javier Patty Magne y al Lic. Samuel Fuentes por haberme guiado en el proceso de desarrollo del presente proyecto.

Al Dr. Gamarra responsable del programa de enfermedades no transmisibles por brindarme el apoyo en el desarrollo del proyecto.

Por otra parte, dar las gracias a todos los docentes que han formado parte de la enseñanza durante todos estos largos años de la carrera; así como a todos los compañeros con los que he compartido horas de clase y de prácticas. También quiero mencionar al resto de compañeros, que, de forma desinteresada, han estado ahí cuando ha hecho falta.

Se cierra una etapa en mi vida, pero se abre otra, llena de nuevas ilusiones y retos profesionales.

DEDICATORIA :

El presente proyecto se lo dedico a mi padre Luis Gualberto Sarzuri Matias por el apoyo que me dio hasta los últimos días de su vida y que, a pesar de todo, ha sido la persona que me lleno de motivación para seguir adelante con el estudio. A mi madre Rosalía Nina de Sarzuri que me ha acompañado en todo momento y ha sido un pilar fundamental en lograr mis objetivos, por su gran esfuerzo de apoyarnos a mí a mis hermanos. A mis hermanos Lizzett, Jhenny, Johana, Diego, Berty, y Dayana por estar apoyándome en todo momento, y estar siempre unidos.

Mary Lenny Sarzuri Nina

RESUMEN

A la presente síntesis se proporciona un enfoque general sobre la elaboración del presente proyecto de grado, desarrollo de sistema informático de diagnóstico de insuficiencia renal.

En la actualidad, las enfermedades no transmisibles se han incrementado por el ritmo acelerado de vida, el sedentarismo, la mala nutrición y otros factores individuales, físicos, económicos y sociales; ante este hecho los gobiernos, con las instituciones de salud en particular, se preocupan por ofrecer programas que colaboren, con la intención de prevenir y controlar los factores de riesgo aplicando la detección precoz de estas enfermedades, debido a que sale más costoso el tratamiento de su recuperación. Tomando en cuenta este problema se desarrolló un sistema informático de diagnóstico de insuficiencia renal, para ayudar a optimizar el control de enfermedades del programa de prevención en la unidad de enfermedades no transmisibles del SEDES Pando, donde fundamentalmente se hace el control del diagnóstico de insuficiencia renal en pacientes de los diferentes centros de salud de la ciudad de Cobija.

Para el desarrollo del sistema se usó la metodología de Proceso Unificado con la herramienta UML, lenguaje de programación PHP, base de datos Mysql.

Una vez concluido el desarrollo del sistema informático en base a la metodología UP, se cuenta con un sistema amigable que permite dar a conocer el diagnóstico de Insuficiencia Renal, además genera los diferentes tipos de búsqueda, listados y reportes, con características que presenta la información de una manera gráfica almacenando la información general y el diagnóstico de los pacientes en la unidad de enfermedades no trasmisibles del SEDES Pando.

Palabras Claves: Enfermedades no Transmisibles, Prevención, Sistema Informático, UML.

ÍNDICE

CAPITULO I MARCO INTRODUCTORIO

1. MARCO INTRODUCTORIO.....	1
1.1. ANTECEDENTES	2
1.2. PLANTEAMIENTO DEL PROBLEMA.....	3
1.2.1. Problema central.....	3
1.3. OBJETIVOS.....	4
1.3.1. Objetivo general	4
1.3.2. Objetivos específicos	4
1.4. METODOLOGÍA	4
1.5. ALCANCES	6
1.6. ORGANIZACIÓN DEL DOCUMENTO	7

CAPITULO II MARCO TEÓRICO

2. MARCO TEÓRICO	8
2.1. CONTEXTO TEÓRICO	9
2.1.1. Sistemas de información	9
2.1.2. Clasificación de los sistemas.....	9
2.1.3. Modelo informático de salud.	11
2.1.4. Sistemas informáticos en la salud	12
2.1.5. Lenguajes de programación	12
2.1.6. Base de datos.....	14
2.2. ASPECTOS TEÓRICOS ACERCA DE LA SALUD	14
2.2.1. Enfermedades no Transmisibles	14
2.2.2. Clasificación de las Enfermedades no Transmisibles	14
2.2.3. Prevención de Enfermedades no Transmisibles.....	16
2.2.4. Factores de influencia	19
2.2.5. Falta de monitorización de Enfermedades no Transmisibles.....	20

2.3.	MARCO LEGAL DE L PROGRAMA DE ENFERMEDADES NO TRANSMISIBLES DEL SEDES.....	21
2.3.1.	Programa de Enfermedades no Transmisibles	22
2.3.2.	Contexto institucional del Programa de Enfermedades no Transmisibles	22
2.3.3.	El programa de prevención y control de enfermedades crónicas.....	23
2.4.	MARCO METODOLÓGICO	25
2.4.1.	Proceso Unificado	25
2.4.2.	Lenguaje UML	29
2.4.3.	Herramientas de programación	36
2.4.4.	Factores de la calidad ISO 9126.....	41
CAPITULO III		
<u>MARCO APLICATIVO</u>		
3.	MARCO APLICATIVO.....	42
3.1.	FASES DE LA METODOLOGÍA DEL PROCESO UNIFICADO	43
3.1.1.	Fase de inicio.....	43
3.1.2.	Fase de elaboración	50
3.1.3.	Fase de construcción	85
3.1.4.	Fase de transición	92
3.2.	VALORACIÓN DEL SISTEMA.....	100
3.2.1.	Funcionalidad.....	101
3.2.2.	Fiabilidad	101
3.2.3.	Mantenimiento	103
3.2.4.	Portabilidad	103
3.2.5.	Conclusión	104
CAPITULO IV		
<u>CONCLUSIONES Y RECOMENDACIONES</u>		
4.	CONCLUSIONES Y RECOMENDACIONES	105
4.1.	CONCLUSIÓN	106
4.2.	RECOMENDACIONES	107
	BIBLIOGRAFÍA	108

ÍNDICE DE FIGURAS

<i>FIGURA 2.1: Clasificación de sistema según el nivel organizacional.....</i>	<i>10</i>
<i>FIGURA 2.2: Partes del Modelo informático en la salud.....</i>	<i>11</i>
<i>FIGURA 2.3: Organigrama del área de Enfermedades no Transmisibles.....</i>	<i>23</i>
<i>FIGURA 2.4: Proceso de desarrollo de software.....</i>	<i>25</i>
<i>FIGURA 2.5: Casos de uso que enlaza los flujos de trabajo fundamental.....</i>	<i>26</i>
<i>FIGURA 2.6 : Iteraciones del proceso de análisis de software.....</i>	<i>28</i>
<i>FIGURA 2.7: Clase.....</i>	<i>30</i>
<i>FIGURA 2.8: Interfaz.....</i>	<i>30</i>
<i>FIGURA 2.9: Colaboración.....</i>	<i>31</i>
<i>FIGURA 2.10: Caso de uso.....</i>	<i>31</i>
<i>FIGURA 2.11: Componente.....</i>	<i>31</i>
<i>FIGURA: 2.12 Nodo.....</i>	<i>32</i>
<i>FIGURA: 2.13 Relaciones de clases UML.....</i>	<i>34</i>
<i>FIGURA 3.14: Modelo de Proceso de Negocio de la unidad de Enfermedades no Transmisibles.....</i>	<i>44</i>
<i>FIGURA 3.15: Diagrama de caso de uso inicial control paciente.....</i>	<i>48</i>
<i>FIGURA 3.16: Diagrama de actividades registrar paciente.....</i>	<i>49</i>
<i>FIGURA 3.17: Casos de uso autenticar usuario.....</i>	<i>51</i>
<i>FIGURA 3.18 : Caso de uso administrar usuario.....</i>	<i>52</i>
<i>FIGURA 3.19: Caso de uso controlar paciente.....</i>	<i>54</i>
<i>FIGURA 3.20: Caso de uso controlar enfermedades.....</i>	<i>55</i>
<i>FIGURA 3.21: Caso de uso controlar centro de Salud.....</i>	<i>59</i>
<i>FIGURA 3.22: Diagrama de clases autenticación.....</i>	<i>60</i>
<i>FIGURA 3.23: Diagrama de clases usuario.....</i>	<i>61</i>
<i>FIGURA 3.24: Diagrama de clases registrar paciente.....</i>	<i>62</i>
<i>FIGURA 3.25: Diagrama de clases centro de salud.....</i>	<i>63</i>
<i>FIGURA 3.26: Diagrama de clases glucosa.....</i>	<i>64</i>
<i>FIGURA 3.27: Diagrama de clases diástole.....</i>	<i>65</i>
<i>FIGURA 3.28 : Diagrama de clases sístole.....</i>	<i>66</i>
<i>FIGURA 3.29: Diagrama de clase diabetes.....</i>	<i>67</i>
<i>FIGURA 3.30: Diagrama de clase presión arterial.....</i>	<i>68</i>
<i>FIGURA 3.31: Diagrama de secuencia autenticar.....</i>	<i>69</i>
<i>FIGURA 3.32: Diagrama de secuencia registrar usuario.....</i>	<i>70</i>
<i>FIGURA 3.33: Diagrama de secuencia registro de paciente.....</i>	<i>71</i>
<i>FIGURA 3.34: Diagrama de secuencia modificar paciente.....</i>	<i>72</i>
<i>FIGURA 3.35: Diagrama de secuencia control de centro de salud.....</i>	<i>73</i>
<i>FIGURA 3.36: Diagrama de secuencia registrar glucosa.....</i>	<i>74</i>
<i>FIGURA 3.37: Diagrama de Secuencia de registrar presión arterial.....</i>	<i>75</i>
<i>FIGURA 3.38: Diagrama de secuencia registrar diabetes.....</i>	<i>76</i>

<i>FIGURA 3.39: Diseño de la arquitectura.....</i>	<i>77</i>
<i>FIGURA 3.40: Modelo lógico de la base de datos</i>	<i>79</i>
<i>FIGURA 3.41: Modelo físico de la base de datos</i>	<i>80</i>
<i>FIGURA 3.42: Interfaz autenticación usuario.....</i>	<i>86</i>
<i>FIGURA 3.43: Interfaz de usuario.....</i>	<i>86</i>
<i>FIGURA 3.44: Registro paciente.....</i>	<i>87</i>
<i>FIGURA 3.45: Formulario de registro centro de salud</i>	<i>87</i>
<i>FIGURA 3.46: Formulario de registro detalle paciente</i>	<i>88</i>
<i>FIGURA 3.47: Lista del estado de la presión arterial.....</i>	<i>89</i>
<i>FIGURA 3.48: Lista del estado de la diabetes</i>	<i>90</i>
<i>FIGURA 3.49: Resultado del diagnóstico de los pacientes.....</i>	<i>91</i>
<i>FIGURA 3.50: Diagrama de despliegue.....</i>	<i>92</i>
<i>FIGURA 3.51: Diagrama de componentes.....</i>	<i>93</i>

ÍNDICE DE TABLAS

TABLA 1.1: Fases de la metodología de Proceso Unificado.....	6
TABLA 2.2: Cuadro del estado de Insuficiencia Renal Crónica	18
TABLA 3.3: Modelo inicial de requerimientos	45
TABLA 3.4: Requisitos no funcionales	46
TABLA 3.5 Actores del sistema.....	47
TABLA 3.6: Requerimientos	51
TABLA 3.7: Caso de uso administrar usuario.....	53
TABLA 3.8: Descripción del caso de uso controlar paciente.....	55
TABLA 3.9: Descripción del caso de uso controlar enfermedades	58
TABLA 3.10: Descripción caso de uso administrar centro de salud.....	60
TABLA 3.11 : Tabla centro salud	81
TABLA 3.12 : Tabla lugar.....	81
TABLA 3.13 : Tabla paciente.....	82
TABLA 3.14 Tabla presión arterial	82
TABLA 3.15 : Tabla diabetes.....	82
TABLA 3.16 : Tabla diástole.....	83
TABLA 3.17 : Tabla sístole	83
TABLA 3.18 : Tabla glicemia.....	83
TABLA 3.19 : Tabla glucosa.....	84
TABLA 3.20 : Tabla detalle paciente.....	84
TABLA 3.21 : Tabla irc.....	85
TABLA 3.22: Prueba de integración autenticar usuario.....	94
TABLA 3.23: Prueba de integración registrar usuario	95
TABLA 3.24: Prueba de integración registrar usuario	95
TABLA 3.25: Prueba de integración registrar paciente.....	96
TABLA 3.26: Prueba de integración registrar detalle paciente	98
TABLA 3.27: Prueba unitaria autenticar	98
TABLA 3.28: Prueba unitaria registrar paciente	99
TABLA 3.29: Prueba unitaria registro detalle paciente.....	99
TABLA 3.30: Escala de calificación para el Sistema informático de diagnóstico de IR ..	100
TABLA 3.31. Resultado de la medida funcionalidad	101
TABLA 3.32. Resultado de la medida fiabilidad.....	101
TABLA 3.33. Resultado de la medida usabilidad	102
TABLA 3.34. Resultado de la medida eficiencia.....	102
TABLA 3.35. Resultado de la medida mantenimiento	103
TABLA 3.36. Resultado de la medida portabilidad	103
TABLA 3.37. Resultado generales	104

CAPITULO I

1. MARCO INTRODUCTORIO

1.1. ANTECEDENTES

En la actualidad, el avance tecnológico en el área de la medicina, ha incrementado las expectativas del ser humano; pero las diferentes patologías, de las enfermedades no transmisibles aún continúan amenazando la salud de la población.

La insuficiencia renal crónica es una de las principales causas de morbilidad y muerte en Bolivia, constituyéndose en un desafío para el sistema de salud. La primera causa de enfermedad renal crónica en Bolivia, es la diabetes, siendo la segunda causa la hipertensión arterial, ambas con un adecuado control pueden evitar o retrasar su aparición. Otras causas control médico, pueden evitar que se dañen los riñones. Lo que pocos saben, es que esta enfermedad se puede prevenir, porque casi el 85% de las causas de la enfermedad se pueden detectar y controlar precozmente, y así evitar el daño a los riñones.

Tres de cada diez pacientes que tienen enfermedad renal crónica fallecen cada año en Bolivia, según los datos del Programa de Salud Renal dependiente del Ministerio de Salud.

La responsable del Programa de Salud Renal, Silvia Paz, explicó a La Razón que los adultos mayores tienen un alto riesgo de padecer una enfermedad renal crónica, lo que deriva en la complicación de otros órganos y el aumento de riesgo de muerte. Este mal puede requerir tratamiento de hemodiálisis o trasplante de riñón, si el paciente no se trata a tiempo. (Pérez, 2015)

El departamento que tiene más enfermos renales es La Paz, con 45%, le siguen Cochabamba con el 23,6%, Santa Cruz con 17,7% y Chuquisaca 6,3%, Potosí y Tarija, el restante porcentaje; Beni y Pando no se conoce la incidencia pero existen muchos enfermos renales. La insuficiencia renal está catalogada dentro de las enfermedades crónicas y su incidencia está en crecimiento permanente al igual que las enfermedades que llevan a esta situación como la hipertensión arterial y la diabetes. (Busch, 2014)

Es por esta razón viendo la necesidad de monitorear las causas y evolución de esta enfermedad, así como la de prevenir con una mayor antelación situaciones de emergencia ante las mismas, es que se quiere desarrollar un sistema para el diagnóstico de insuficiencia renal crónica.

A continuación, se mencionarán ejemplos de algunos sistemas informáticos que se realizaron, que son referidos a la salud.

“Diseño y elaboración de un sistema de información para el manejo de historias clínicas de pacientes con enfermedades pediátricas.”

“Sistema informático para el almacenamiento y manejo de datos clínicos y microbiológicos del paciente con fibrosis quística.”

1.2. PLANTEAMIENTO DEL PROBLEMA

La incidencia de Enfermedades no Transmisibles en la población pandina, se ha incrementado en los últimos tiempos, pese a que no hay datos reales pues no existe información estadística que lo confirme; pero según el Dr. Gamarra, responsable de la unidad de enfermedades no transmisibles del SEDES Pando, se han incrementado pacientes con dichas enfermedades, el cual se debe atender rápidamente, y realizar la prevención necesaria, se ha visto que la unidad carece de mecanismos necesarios para la administración de información de los pacientes y conocer un diagnóstico oportuno, pues la unidad solo cuenta con una base de datos de Excel donde solo guarda registros de información de los pacientes.

1.2.1. Problema central

Con todo lo anterior mencionado se plantea el siguiente problema principal:

“Deficiencia en el control de información del diagnóstico de insuficiencia renal en la unidad de enfermedades no transmisibles de SEDES Pando”.

1.3. OBJETIVOS

1.3.1. Objetivo general

Desarrollar un sistema informático para el control eficiente de información del diagnóstico de insuficiencia renal, en la unidad de enfermedades no transmisibles del SEDES Pando, mediante la aplicación de la metodología Proceso Unificado (UP).

1.3.2. Objetivos específicos

- Recabar la información dentro de la unidad de enfermedades no transmisibles, para la comprensión del proceso de actividades y requerimientos, mediante el diagrama modelado de negocios UML.
- Analizar los requerimientos para determinar los alcances y limitaciones del sistema, mediante el modelo de casos de uso.
- Diseñar la arquitectura del sistema informático para construir las funcionalidades y componentes de los módulos del sistema, mediante diagramas UML.
- Implementar el diseño del sistema informático, para demostrar su funcionalidad utilizando un lenguaje de programación orientado a objetos.
- Realizar las pruebas del sistema informático, para verificar las funcionalidades requeridas aplicando pruebas unitarias y pruebas de integración.

1.4. METODOLOGÍA

La metodología adoptada para el presente proyecto de grado es el Proceso Unificado y la utilización del lenguaje de modelado (UML).

El Proceso Unificado es un proceso de desarrollo de software. Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario a un sistema software. Por otro lado, cabe señalar que el verdadero aspecto definitorio del

Proceso Unificado se resume en tres frases clave – dirigido por casos de uso, centrado en la arquitectura, e iterativo e incremental. (Ivar Jacobson, 2000).

Sin embargo, el Proceso Unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto. Está basado en componentes, lo cual quiere decir que el sistema software en construcción está formado por componentes software interconectados a través de interfaces bien definidas.

FASES	OBJETIVO	TÉCNICAS	HERRAMIENTAS	PRODUCTOS
INICIO	Analizar el negocio para obtener los requerimientos y de delimitar el sistema analizado	Cuestionario Revisión de documentos, entrevistas,	Diagramas BPMN Diagramas de casos de uso UML	Modelo de negocio con la descripción de la estructura orgánica, procesos del negocio Modelo inicial de requerimientos Modelo inicial de casos de uso
ELABORACIÓN	Describir los modelos casos de uso analizar y diseñar la arquitectura del software	Análisis de requerimientos Descripción de casos de uso Análisis y diseño de diagrama de clases Diseño de la base de datos	Diagramas de casos de uso UML Diagrama de clases UML Diseño de la base de datos en Power Designer	Modelo de requerimientos terminado Modelo de casos de uso Base de datos Artefactos del análisis terminado

CONSTRUCCIÓN	Producir la primera versión operativa del sistema informático	Los módulos en el diseño detallado se codifican y se les aplica las pruebas de unidad	Lenguaje de programación php orientada a objetos y método de pruebas unitarias	La arquitectura terminada
TRANSICIÓN	Implantar el producto en su entorno de operación	Instalación del sistema informático		Implantación del software en la entidad

TABLA 1.1: *Fases de la metodología de Proceso Unificado*
FUENTE: *Elaboración propia*

1.5. ALCANCES

El sistema de diagnóstico de insuficiencia renal, procesa información en forma automatizada, e informa a los usuarios y al profesional médico sobre el estado del paciente.

El sistema recolecta información del paciente, sus datos completos, datos clínicos referentes a los análisis que se haya realizado en el centro médico.

El sistema realiza reportes de los resultados del diagnóstico, que pueda padecer con relación a las enfermedades no transmisibles de obesidad, diabetes, hipertensión arterial tales que estas inciden a que pueda padecer insuficiencia renal.

El sistema brinda información oportuna sobre el estado de diagnóstico del paciente, pero esta no dará el tratamiento.

Cabe mencionar que no se realizara la parte final de la metodología que contempla la implantación.

1.6. ORGANIZACIÓN DEL DOCUMENTO

Capítulo 1: Este capítulo comprende la parte introductoria del documento, el análisis principal del tema considerando en primer lugar el problema central los objetivos, del desarrollo del sistema que ira en beneficio de la unidad del SEDES PANDO; se describe la metodología utilizada para el desarrollo del proyecto y al finalizar se mencionan los alcances obtenidos.

Capítulo 2: En este capítulo se encuentra el marco teórico, donde forman parte el contexto teórico de los temas necesarios que se plantean en capítulo 1, la información del tema central del objetivo, como también el marco metodológico, y las herramientas que se utilizaron para desarrollar el sistema.

Capítulo 3: En este capítulo se muestran los procesos realizados que van de acuerdo a la metodología UP, así mismo se muestran los resultados obtenidos y pruebas de desarrollo.

Capítulo 4: En este capítulo concluimos el desarrollo del proyecto mencionando resultados planteados en el objetivo, siempre con recomendaciones, a poder evolucionar mejorar el sistema, también se muestran la bibliografía y anexos.

CAPITULO II

2. MARCO TEÓRICO

2.1. CONTEXTO TEÓRICO

2.1.1. Sistemas de información

El uso de los ordenadores en todos los ámbitos de la vida laboral, hoy es una realidad, que si bien ha aportado muchos beneficios no ha estado exenta de problemas, si estas dificultades no han sido tenidas en cuenta en un principio, han llevado al fracaso de muchos proyectos.

La complejidad del manejo de la realidad sanitaria, que genera una gran cantidad de datos, obliga a contar con herramientas que permitan seleccionar y manejar información, en vez de datos, de una forma ágil y segura (Información = Datos x Proceso). Para la obtención y manejo de información es muy útil la informática y, esta utilidad, es la que justificaría el cambio del "bolígrafo por el ordenador". Parece obvio, por tanto, que no utilizar ordenadores en la atención sanitaria es cerrarse a potenciales mejoras en el manejo de la información generada de los pacientes.

En este sentido un Sistema de Información, sanitario o de cualquier otro tipo, permite conocer la distancia, y las alternativas con que contamos para conseguir una meta, la cual debe previamente ser definida (la información es para la acción), (Salamea, 2003)

2.1.2. Clasificación de los sistemas

Los Sistemas son un conjunto integrado de personas, equipos (Hardware), programas (Software), métodos y modelos con el objetivo de entregar a la organización, la información requerida para apoyar la toma de decisiones.

Según el nivel organizacional que apoyan se clasifican en:



FIGURA 2.1: *Clasificación de sistema según el nivel organizacional*

FUENTE: (Salamea, 2003)

Al plantearse un sistema de información hay que considerar tres escenarios diferentes:

- **Un sistema de no integración:** en el que se desarrollan aplicaciones que responden a las necesidades de un departamento o servicio concreto, sin prever las relaciones con los sistemas existentes y mucho menos con los venideros, éste, aunque aporta beneficios a corto plazo implica una serie de problemas, como la producción de datos irreconciliables, que no pueden ser explotados por el resto del sistema, generando datos inconsistentes y duplicidad de tareas. Por lo que terminan creando un sistema redundante con una baja fiabilidad de los datos obtenidos.
- **Un sistema totalmente integrado:** aquel que cubre todas las actividades desarrolladas y todos los datos que éstas precisan. Los problemas que este modelo presenta son, la dificultad para la gestión, tanto en el diseño de una base de datos que reúna todas las necesidades, como también los recursos informáticos para su implementación; además requiere un conocimiento profundo del sistema para poder operarlo y se corre el riesgo de convertir a una herramienta en el fin u objetivo último.
- **Un sistema parcialmente integrado:** en el que por un lado se divide el área de trabajo en elementos manejables y por otro, define, planifica y controla los mismos. El objetivo es lograr un grado realista de integración, la ventaja de este planteamiento es que es más “ecológico”, consume menos recursos, por lo que la posibilidad de

implantarlo y que una vez implantado prospere, es mayor. Además, plantea un aspecto interesante, como es considerar que un fallo o carencia de una parte del sistema de información, no tiene por qué influir de manera tan importante en el resto del sistema. (Salamea, 2003)

2.1.3. Modelo informático de salud.

El desarrollo de la informática en salud actual, tiene su origen en el crecimiento de las estadísticas en el siglo XVIII y XIX . Como una nueva y creciente disciplina, la estadística tomó el reto de medir y monitorear a sociedades en permanente expansión. La informática en salud tiene como meta la expansión y comunicación de la información, pero ésta es sólo una parte y no el equivalente del complejo proceso de la generación de conocimiento.

El modelo informático en Salud consta de tres partes:



FIGURA 2.2: *Partes del Modelo informático en la salud*

FUENTE: (Salamea, 2003)

El modelo informático de salud consta de tres partes esenciales: datos, información y conocimiento. Estos elementos ordenados en una jerarquía, en la que los datos se encuentran en la base del modelo; una vez ordenados, producen información pudiendo llegar a la potencial generación de conocimiento. El modelo informático se acerca mucho a los objetivos de la medicina basada en la evidencia (MBE), particularmente en lo relativo a la búsqueda, evaluación, revisión y utilización de la información.

2.1.4. Sistemas informáticos en la salud

La creación de sistemas informáticos para su uso en los distintos servicios de salud, no es una actividad nueva, como un ejemplo de esto, tenemos a los sistemas implantados por el CLAP (Centro Latinoamericano de Perinatología), que son el SIP (Sistema Informático Perinatal), el SIA (Sistema Informático del Adolescente), y el SIN (Sistema Informático del Niño); los que a lo largo de su evolución, desde 1995, han sido revisados y mejorados, constituyendo un referente para la recolección de datos y su procesamiento; facilitando la consecución de información uniforme, y así facilitar la realización de investigaciones metodológicamente similares, replicables y la obtención de datos comparables, fin último en lo concerniente a la producción de conocimiento. Según (Salamea, 2003).

2.1.5. Lenguajes de programación

Según (Morales, 2014). Un lenguaje de programación es básicamente un sistema estructurado de comunicación, similar al humano, el cual nos permite comunicarnos por medio de signos, ya sean palabras, sonidos o gestos. Refiriéndonos a los aparatos, este sistema está organizado para que se entiendan entre sí y a su vez interprete las instrucciones que debe ejecutar.

El término programación se define como un conjunto de instrucciones consecutivas y ordenadas que llevan a ejecutar una tarea específica. Dichas instrucciones se denominan “código fuente”, el cual es único para cada lenguaje y está diseñado para cumplir una función o propósito específico. Usan diferentes normas o bases para controlar el comportamiento de un dispositivo y también pueden ser usados para crear programas informáticos. En la actualidad, hay más de cien lenguajes de programación diferentes, estos se clasifican en dos tipos principales:

- **Lenguajes de programación de bajo nivel.** Son aquellos utilizados para controlar el hardware (partes tangibles) del aparato y dependen directamente de la máquina, es decir, que no pueden ser usados en aparatos diferentes para los que fueron creados. Estos lenguajes son los que ordenan las operaciones esenciales para el funcionamiento del dispositivo. También es conocido como código máquina. Su funcionamiento es complejo, por lo que lo utilizan principalmente los fabricantes de

hardware. Con ellos se pueden programar tareas como reproducción de audio o video, mostrar imágenes, realizar operaciones matemáticas, movimiento del puntero, asignación, liberación de memoria, entre otras.

- **Lenguajes de programación de alto nivel.** Estos lenguajes son más parecidos al humano. No dependen de la máquina y sirven principalmente para crear programas informáticos que puedan solucionar distintos tipos de necesidades.

Ahora bien, algunos de los lenguajes de programación más utilizados son SQL, PHP, C y Java.

- **SQL:** son las siglas de Structured Query Language (Lenguaje Estructurado de Consulta). Su función principal es actuar sobre una base de datos y extraer su contenido para almacenar, introducir, actualizar, eliminar y consultar información. Suele ser usado en la construcción de páginas web y aplicaciones de escritorio.
- **PHP:** es un acrónimo recursivo que significa PHP Hypertext Pre-processor. Es utilizado para el desarrollo de sitios online, ya que es posible combinarlo con HTML. Es un lenguaje de lado del servidor, es decir, que primero se ejecuta en éste y posteriormente regresa al navegador del usuario como resultado una página web.
- **C:** lenguaje orientado principalmente a los sistemas operativos. Se puede crear un software para sistemas operativos como Windows o Linux, también es muy usado en robótica para crear simuladores o para experimentos informáticos, físicos, matemáticos, etc.
- **Java:** es uno de los lenguajes favoritos ya que permite construir programas que gestionan la memoria del dispositivo donde reside la aplicación, como por ejemplo herramientas, juegos y apps. Es utilizado en dispositivos móviles, aparatos de televisión y computadores personales. Suele ser asociado a Java Script, que es otro tipo de lenguaje para crear pequeños programas encargados de realizar acciones dentro de páginas webs, como por ejemplo crear efectos especiales o para definir la interactividad con los usuarios.

2.1.6. Base de datos

La base de datos puede definirse como una colección de datos interrelacionados almacenados en conjunto sin redundancias perjudiciales o innecesarias; su finalidad es la de servir a una aplicación o más, de la mejor manera posible; los datos se almacenan de modo que resulten independientes de los programas que lo usan; se emplean métodos bien determinados para incluir datos nuevos y para modificar o extraer los datos almacenados. Dícese que un sistema comprende una colección de base de datos cuando estas son totalmente independientes desde el punto de vista estructural. (Martin, 1977)

2.2. ASPECTOS TEÓRICOS ACERCA DE LA SALUD

2.2.1. Enfermedades no Transmisibles

Según. (Busch, 2014) La vida moderna con sus ritmos acelerados y elevado sentido de la competencia, hace que las personas descuiden paulatinamente algunas áreas fundamentales. Por ignorancia, falta de tiempo o estrés acumulado, mucha gente ya no se cuida en sus hábitos alimenticios y consumen comidas rápidas, congeladas y pastelería industrial, costumbre que con el tiempo menoscaba al ser humano, sobreviniendo la obesidad y una serie de complicaciones de salud.

Sin embargo, existe la posibilidad de atenuar los efectos de la enfermedad o prevenirla con la simple inclusión de hábitos saludables a la rutina diaria. Teniendo en cuenta lo citado, se puede decir que las ECNT son enfermedades que se pueden prevenir. Es por eso que los gobiernos en general y las instituciones de salud en particular se preocupan por ofrecer programas que colaboren con esta intención debido a que sale más costoso el tratamiento de estas enfermedades que su prevención.

2.2.2. Clasificación de las Enfermedades no Transmisibles

Según (Orlando, 2012) La OMS, el Plan Nacional de Salud Pública (2007-2010) y algunas instituciones médicas presentan como ENT a las siguientes enfermedades:

- **Obesidad:** El sobrepeso y la obesidad se definen como una acumulación anormal o excesiva de grasa que puede ser perjudicial para la salud. El índice de masa corporal (IMC) es un indicador simple de la relación entre el peso y la talla que se utiliza frecuentemente para identificar el sobrepeso y la obesidad en los adultos. Se calcula dividiendo el peso de una persona en kilos por el cuadrado de su talla en metros (kg/m²). Según la OMS, un IMC igual o superior a 25 determina sobrepeso. Un IMC igual o superior a 30 determina obesidad. El IMC proporciona la medida más útil del sobrepeso y la obesidad en la población, puesto que es la misma para ambos sexos y para los adultos de todas las edades. Sin embargo, hay que considerarla a título indicativo porque es posible que no se corresponda con el mismo nivel de grosor en diferentes personas.
- **La hipertensión arterial (HTA):** es una enfermedad caracterizada por un incremento continuo de las cifras de presión sanguínea en las arterias. Aunque no hay un umbral estricto que permita definir el límite entre el riesgo y la seguridad, de acuerdo con consensos internacionales, una presión sistólica sostenida por encima de 139 mmHg o una presión diastólica sostenida mayor de 89 mmHg, están asociadas con un aumento medible del riesgo de aterosclerosis y por lo tanto, se considera como una hipertensión clínicamente significativa.
- **Diabetes:** Es una enfermedad en la que los niveles de glucosa (azúcar) de la sangre están muy altos. La glucosa proviene de los alimentos que se consumen. La insulina es una hormona que ayuda a que la glucosa entre a las células para suministrarles energía. En la diabetes tipo uno, el cuerpo no produce insulina. En la diabetes tipo dos, el tipo más común, el cuerpo no produce o no usa la insulina adecuadamente. Sin suficiente insulina, la glucosa permanece en la sangre.
Con el tiempo, el exceso de glucosa en la sangre puede causar problemas serios. Puede provocar lesiones en los ojos, los riñones y los nervios. La diabetes también puede causar enfermedades cardíacas, derrames cerebrales e incluso la necesidad de amputar un miembro.
El ejercicio, el control de peso y respetar el plan de comidas puede ayudar a controlar la diabetes. También debe controlar el nivel de glucosa y, si tiene receta médica, tomar medicamentos.

- **Insuficiencia Renal Crónica:** (Dra. Aida Venado Estrada, 2010) Los riñones realizan varias funciones en el organismo: 1) filtran la sangre y eliminan productos de desecho del metabolismo, así como sustancias endógenas y exógenas, 2) mantienen el balance hidroelectrolítico, 3) regulan el equilibrio ácido – base, 4) secretan hormonas como la eritropoyetina y la renina y 5) modifican sustancias como la vitamina D, para la regulación del fósforo y el calcio.

Los riñones están constituidos por unidades funcionales llamadas nefronas las cuales están formadas por un glomérulo y un túbulo. El glomérulo es un conjunto de vasos sanguíneos a través del cual se filtran más de 150 litros de sangre al día. Este ultra filtrado del plasma que contiene moléculas pequeñas como urea, creatinina, glucosa y iones pasa al espacio capsular y posteriormente a los túbulos. En los túbulos se reabsorbe agua y sustancias químicas útiles como aminoácidos y iones, concentrándose las sustancias de desecho y el exceso de agua que terminan excretándose en 1 o 2 litros de orina al día.

Las causas de insuficiencia renal crónica (IRC) se pueden agrupar en enfermedades vasculares, enfermedades glomerulares, túbulo intersticiales y uropatías obstructivas.

2.2.3. Prevención de Enfermedades no Transmisibles

Según. (Dr Santos Depine, 2007) Identificar precozmente pacientes con riesgo de/o enfermedad renal de acuerdo a Estrategias de Salud Poblacional, de Grupos de Riesgo y/o Mixtas.

Factores de riesgo y grupos vulnerables para desarrollar enfermedad renal crónica.

- Edad > 60 años
- Mujeres en estado de gravidez
- Hipertensión arterial
- Diabetes mellitus
- Historia familiar de enfermedad renal crónica
- Enfermedad sistémica
- Anormalidades del tracto urinario

- Infección del tracto urinario
- Obesidad
- Tabaquismo

La Clasificación por Estadios, ha sido uno de los adelantos más importantes en el campo de la nefrología. Ha permitido revalorizar la determinación de la Creatinina Sérica (Crs) como marcador de la función renal. Es sabido que, en la mayoría de los casos, debe existir una pérdida cercana al 50 % de la función renal, para que se manifieste un incremento de la creatinina sérica e incluso en estos casos tiene variaciones en relación a la masa muscular del paciente. Recientemente, la utilización de ecuaciones de predicción de la Velocidad de Filtrado Glomerular (VFG), ha permitido “corregir” el dato aislado de la Crs, ajustándolo en función de Estadios, según su VFG. Existen numerosas ecuaciones de predicción, siendo las más habituales la de Cockcroft-Gault y la de MDRD:

- **Ecuación de Cockcroft-Gault** (estima clearance)

Cl Cr (hombre) en ml/min = $(140 - \text{edad}) \times \text{Peso} / (72 \times \text{Crs})$

Cl Cr (mujer) en ml/min = $(140 - \text{edad}) \times \text{Peso} \times 0,85 / (72 \times \text{Crs})$

➤ **Márgenes del estado de Insuficiencia Renal**

Clasificación de los pacientes por Estadios, según FG o VFG.

FG = Filtrado Glomerular.

Estadio ERC	Descripción	FG Ml/min/1,73 m	Sinónimos	Clasificación según tratamiento	Plan de Actuación
1	Daño renal con FG normal	Mayor a 90	Normal		<ul style="list-style-type: none"> - Diagnóstico-tratamiento - Frenar progresión - Prevención cardiovascular
2	Daño renal con FG levemente	60-89	Daño Renal Leve	T si receptor de trasplante	<ul style="list-style-type: none"> - Frenar progresión - Prevención cardiovascular
3	FG Moderado	30-59	Insuficiencia Renal Crónica moderada		<ul style="list-style-type: none"> - Igual que el anterior - Tratar complicaciones ERC
4	FG Severa	15-29	Insuficiencia Renal Crónica avanzada, pre terminal		<ul style="list-style-type: none"> - Igual que el anterior - Preparar tratamiento sustitutivo
5	Fallo renal	Menor a 15	Fallo Renal, uremia, Insuficiencia Renal terminal	D si diálisis (hemodiálisis, diálisis peritoneal)	<ul style="list-style-type: none"> - Tratamiento sustitutivo - Prevención cardiovascular

TABLA 2.2: *Cuadro del estado de Insuficiencia Renal Crónica*
FUENTE: (Dr Santos Depine, 2007)

2.2.4. Factores de influencia

Según (Chan, 2010) Un alto porcentaje de ENT puede prevenirse mediante la reducción de sus cuatro factores de riesgo comportamentales más importantes: el tabaquismo, el sedentarismo, el uso nocivo del alcohol y las dietas malsanas. La influencia de estas conductas de riesgo y de otras causas metabólicas y fisiológicas de la epidemia mundial de ENT abarca lo siguiente:

- **El tabaquismo:** alrededor de 6 millones de personas mueren a causa del tabaco cada año, tanto por el consumo directo como por el pasivo. Hacia 2030 esa cifra aumentará hasta los 7,5 millones, lo que representará el 10% del total de muertes. Se estima que el tabaquismo causa aproximadamente el 71% de los casos de cáncer de pulmón, el 42% de las enfermedades respiratorias crónicas y alrededor del 10% de las enfermedades cardiovasculares. La mayor incidencia de tabaquismo entre los hombres se da en los países de ingresos medios bajos; para el total de la población, la prevalencia de tabaquismo es más elevada en los países de ingresos medios altos.
- **El sedentarismo:** aproximadamente 3,2 millones de personas mueren a causa del sedentarismo cada año. Las personas con poca actividad física corren un riesgo entre un 20% y un 30% mayor que las otras de morir por cualquier causa. La actividad física regular reduce el riesgo de padecer depresión, enfermedades cardiovasculares, hipertensión, diabetes y cáncer de mama o colon. El sedentarismo es más frecuente en los países de ingresos altos, pero actualmente también se detectan niveles muy altos del problema en algunos países de ingresos medios, sobre todo entre las mujeres.
- **El uso nocivo del alcohol:** aproximadamente 2,3 millones de personas mueren a causa del uso nocivo del alcohol cada año, lo que representa alrededor del 3,8% de todas las muertes que tienen lugar en el mundo. Más de la mitad de dichas muertes son provocadas por ENT como cáncer, enfermedades cardiovasculares y cirrosis hepática. Si bien el consumo per cápita entre la población adulta es mayor en los países de ingresos altos, alcanza un nivel similar en los países de ingresos medios altos muy poblados.
- **La dieta no saludable:** el consumo de fruta y verdura en cantidades suficientes reduce el riesgo de padecer enfermedades cardiovasculares, cáncer de estómago y

cáncer colorrectal. La mayoría de las poblaciones consumen niveles de sal mucho más elevados que los recomendados por la OMS para prevenir enfermedades; un consumo elevado de sal es un factor determinante que aumenta el riesgo de padecer hipertensión y enfermedades cardiovasculares. Las cardiopatías están relacionadas con un consumo elevado de grasas saturadas y ácidos grasos trans.

2.2.5. Falta de monitorización de Enfermedades no Transmisibles

El suministro de datos precisos por los países es vital para lograr disminuir el número de muertes y discapacidades que provocan las ENT en todo el mundo. No obstante, una parte significativa de países apenas disponen de datos utilizables sobre la mortalidad y tienen unos sistemas de vigilancia precarios. Además, a menudo los datos sobre las ENT no están integrados en los sistemas nacionales de información sanitaria. La mejora de la vigilancia y la monitorización a nivel nacional debe ser prioritaria a la hora de luchar contra las ENT. En entornos con recursos escasos y capacidad limitada, los sistemas viables y sostenibles pueden ser simples y aun así producir datos valiosos.

Hay tres componentes esenciales de la vigilancia de las ENT constituyen un marco que todos los países deberían establecer y consolidar:

- ✓ La monitorización de las exposiciones (factores de riesgo);
- ✓ La monitorización de los resultados (morbilidad y mortalidad específica de enfermedades)
- ✓ Las respuestas del sistema sanitario, que incluyen asimismo la capacidad del país para prevenir las ENT en términos de políticas y planes, infraestructura, recursos humanos y acceso a la atención sanitaria esencial, medicinas incluidas.

A fin de remediar las serias deficiencias de la vigilancia y la monitorización de las ENT, deben tomarse medidas efectivas:

- ✓ Los sistemas de vigilancia de las ENT deberían reforzarse e integrarse en los sistemas nacionales de información sanitaria existentes.
- ✓ La monitorización y la vigilancia de los factores de riesgo de comportamientos metabólicos en entornos con escasos recursos, deberían tener la máxima prioridad.

- ✓ El registro de los pacientes con riesgos de padecer estas enfermedades a fin de prevenir y hacer tomar una concienciación
- ✓ Un registro fiable de la mortalidad adulta constituye un requisito primordial para monitorizar las ENT en todos los países.

Reforzar la vigilancia constituye una prioridad a nivel nacional y mundial. Existe una necesidad de hacer esfuerzos a fin de mejorar la cobertura y la calidad de los datos sobre la mortalidad, llevar a cabo encuestas regulares sobre los factores de riesgo a escala nacional con métodos normalizados y evaluar periódicamente la capacidad de los países para prevenir y controlar las ENT.

2.3. MARCO LEGAL DEL PROGRAMA DE ENFERMEDADES NO TRANSMISIBLES DEL SEDES

Considerando que, el Plan Estratégico de Salud (PES) elaborado en el marco del Plan General de Desarrollo Económico y Social (PGDES), constituye un instrumento fundamental en la lucha contra la pobreza; Que, uno de los fundamentos de dicho Plan se orienta a profundizar la descentralización de la gestión de salud, promoviendo la eficiencia y eficacia, en las instituciones públicas a nivel departamental y una sólida articulación con los organismos sectoriales a nivel nacional; Que, es necesario impulsar la especialización, coherencia tecnológica, unidad sistémica y capacidad resolutiva en la gestión de la salud, mediante redes de servicios organizadas territorialmente, articulándolas con la gestión municipal y la participación de las organizaciones de la sociedad civil, acercando los servicios al usuario;

Que en cumplimiento de lo dispuesto por el Art. 30 del Decreto Supremo N° 25060 de 2 de junio de 1998, debe reglamentarse la organización del Servicio Departamental de Salud, en las Prefecturas de Departamento. Según. (Suarez, 1998)

2.3.1. Programa de Enfermedades no Transmisibles

Según (Miranda, 2012) Dependencia: de la Unidad de Epidemiología.

Resolución Ministerial N° 1126 de 22 de septiembre del 2010 de aprobación del Plan Nacional 2010 –2015 de Enfermedades No Transmisibles.

Resolución Ministerial N° 0384 de 19 de abril de 2010 de aprobación de aplicación de la Norma y Procedimientos para la Prevención y Control de Enfermedades No Transmisibles y sus factores de riesgo.

2.3.2. Contexto institucional del Programa de Enfermedades no Transmisibles

Nombre de la unidad: Área de Enfermedades no transmisibles

✓ Naturaleza

Ejecutar políticas nacionales e instructivas de prevención, control y supervisión de vigilancia y control de enfermedades no transmisibles

Coordinar acciones técnicas con la Unidad de Epidemiología para el desarrollo de vigilancia y control de enfermedades no transmisibles, por niveles de atención.

Cumplir con los planes, programas diseñados y aprobados por el ministerio de salud y deportes.

Ejecutar actividades, tareas técnicas y operativas en coordinación con la Unidad de Epidemiología, cumpliendo con las metas establecidas.

- ✓ Prestar servicios de acuerdo a normativa vigente.
- ✓ Coordinar la recopilación de información epidemiológica de los SEDES, en función a la competencia del área, en coordinación con el SNIS.
- ✓ Vigilancia epidemiológica activa en coordinación con los SEDES.
- ✓ Elaborar estudios de investigación epidemiológica en el área de enfermedades crónicas no transmisibles y factores de riesgo asociados a nivel nacional.

- ✓ Proporcionar capacitación técnica a personal de primer y segundo nivel de atención de los SEDES en la prevención, manejo y control de enfermedades no transmisibles y factores de riesgo asociado.
- ✓ Gestionar el fortalecimiento de infraestructura, recurso humano e insumos médicos de centros de primer, segundo y tercer nivel de los SEDES para la resolución de enfermedades no transmisibles priorizadas de acuerdo a perfil epidemiológico.

2.3.2.1. Organigrama de la unidad del programa Enfermedades no Transmisibles

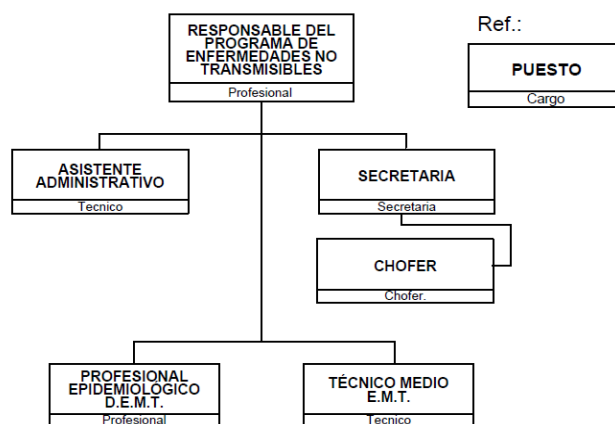


FIGURA 2.3: Organigrama del área de Enfermedades no Transmisibles

FUENTE: (Miranda, 2012)

2.3.3. El programa de prevención y control de enfermedades crónicas

Este Programa fue producto de un trabajo y actividades sistematizadas, con plena participación de las áreas específicas del Ministerio de Salud y Deportes y una Comisión de Representantes Nacionales de las Organizaciones vinculadas a la problemática de las enfermedades renales, convocadas mediante la Resolución Ministerial No. 0314 del 10 de mayo de 2007.

El producto final incluye la planificación y programación del Programa con las herramientas del Marco Lógico. El primer paso fue tomar conocimiento de la problemática de la Enfermedad Renal Permanente y su impacto social y económico en Bolivia y lograr las

adhesiones y el consenso para la implementación de un Programa Nacional de Prevención y Control de Enfermedades Renales. Para ello se realizó un “Taller de Prevención de Enfermedades Renales”, que mostró que Bolivia al igual que todos los países del globo, presenta una creciente prevalencia de pacientes en diálisis, sin el necesario crecimiento proporcional de los recursos humanos, técnicos y económicos para darle una adecuada respuesta. Además, la ausencia de estrategias de promoción y prevención sistematizadas, no han permitido hasta ahora mejorar los indicadores de enfermedad, modificar la inequidad observada en algunas regiones del país por la dificultosa accesibilidad de los pacientes a la consulta médica en general y a la nefrológica en particular en forma precoz.

Por otra parte, la creciente carga de enfermedad renal en su fase final y con requerimiento de tratamiento de diálisis y / o trasplante, ha motivado que los tratamientos no puedan ser efectuados de acuerdo a las normas internacionales en relación a la frecuencia y calidad de práctica.

Finalmente se han establecido las políticas de intervención del Programa bajo el concepto de “Control” de la Enfermedad Renal y teniendo un enfoque particularizado en la búsqueda precoz de la lesión del endotelio vascular, común a todas aquellas enfermedades que cursan de una manera simultánea a la enfermedad renal y a la enfermedad cardiovascular, cuyo principal exponente está dado por la Diabetes (principal causa de enfermedad renal en la mayoría de los países del mundo, incluyendo a Bolivia), la Obesidad, la Hipertensión Arterial y las Dislipidemias y por ello el Programa se ha planificado bajo estrategias de Control Cardiovascular, Cerebral, Renal y Endocrino metabólica (CaCeREM).

Por ello se han establecido las bases de articulación con el Programa Nacional de Enfermedades No Transmisibles (Pacheco, 2007).

2.4. MARCO METODOLÓGICO

2.4.1. Proceso Unificado

(Ivar Jacobson, 2000) En primer lugar, el Proceso Unificado es un proceso de desarrollo de software. Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema Software. Sin embargo, el Proceso Unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto.

El Proceso Unificado utiliza el lenguaje Unificado de Modelado (Unified Modeling Language, UML) para preparar todos los esquemas de un sistema software. De hecho, UML es una parte esencial del Proceso Unificado, sus desarrollos fueron paralelos.

No obstante, los verdaderos aspectos definatorios del Proceso Unificado se resumen en tres frases clave, dirigido por casos de uso, centrado en la arquitectura, e iterativo e incremental. Un proceso define quien está haciendo que, y cuando, además dice como alcanzar un determinado objetivo. En la ingeniería de software el objetivo es construir un producto de software, vale decir, que todos los proyectos necesitan de un proceso que guie sus actividades.

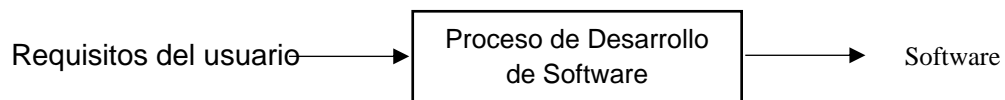


FIGURA 2.4: *Proceso de desarrollo de software*

FUENTE: (Ivar Jacobson, 2000)

RUP y UML están estrechamente relacionados entre sí pues mientras el primero establece las actividades y los criterios para conducir un sistema desde su máximo nivel de abstracción, el segundo ofrece notación grafica necesaria para representar los sucesos, modelos, que se obtienen de procesos de refinamiento.

RUP se define como procesos dirigido por:

- Casos de Uso.
- Centrado a la Arquitectura.
- Iterativo e incremental.

2.4.1.1. Dirigido por casos de uso

Procesos de desarrollo de software utiliza los casos de uso como una herramienta para la obtención de requisitos de usuario. Donde los casos de uso son para definir la funcionalidad del sistema, y guían al desarrollador en la construcción de la arquitectura del sistema.

La descripción obtenida de los requerimientos debe ser comprendida por casos de uso que nos ayudan a recopilar la información acerca de la interacción que tiene los usuarios en este caso actores con el sistema. Un caso de uso es una secuencia, reacciones que el sistema lleva a cabo para ofrecer un resultado de valor a algún actor, que sirven para realizar pruebas sobre los componentes desarrollados. Los casos de uso enlazan los flujos de trabajo fundamentales. El proyecto progresa a través de estos flujos de trabajo, que inician en los casos de uso.



FIGURA 2.5: Casos de uso que enlaza los flujos de trabajo fundamental

FUENTE: (Larman C. , 1999)

2.4.1.2. Centrado en la arquitectura

En el caso de software la arquitectura se refiere a un conjunto de decisiones significativas acerca de la organización de un sistema, la elección de los elementos acerca de la organización de un sistema software, la selección de los elementos estructurales a partir de las cuales se componen el sistema con su respectivo comportamiento y las interacciones entre esos elementos y la composición de esos elementos estructurales.

La necesidad de una arquitectura radica en poder comprender el sistema, es decir que todos los que están involucrados con su desarrollo deben entender el problema al cual va enfocado el sistema de software para satisfacer las demandas individuales y de la organización mediante la utilización de los diagramas definidos por UML.

2.4.1.3. Iterativo e incremental

Jacobson en su libro “El Proceso Unificado de Desarrollo de Software”, explica que en esta fase proporciona la estrategia para desarrollar un producto de software en pasos pequeños manejables:

- ✓ Planificar un poco
- ✓ Especificar, diseñar e implementar un poco
- ✓ Integrar, probar y ejecutar un poco cada iteración.

“Un ciclo de vida iterativo se basa en el agrandamiento y perfeccionamiento secuencial de un sistema a través de múltiples ciclos de desarrollo de análisis, diseño implementación y pruebas”.

Las ventajas de un desarrollo de software con un ciclo de vida iterativo se dan gracias a la retroalimentación en cada ciclo por lo cual se crea un sistema más robusto. En cada incremento que y tiene el sistema se va perfeccionando, lo cual permite al usuario realizar las modificaciones requeridas en el transcurso del tiempo.

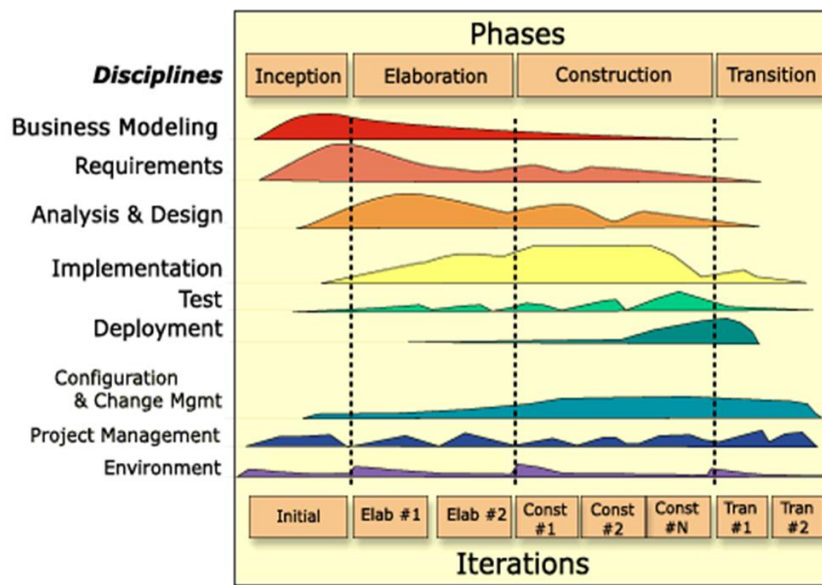


FIGURA 2.6 : Iteraciones del proceso de análisis de software
FUENTE: (Ivar Jacoboson, 2000)

2.4.1.4. Ciclo de vida de la metodología UP

De acuerdo a lo anteriormente mencionado las fases pertenecen a la parte dinámica de la metodología, el ciclo de vida del software del UP se descompone en cuatro fases secuenciales.

✓ Fase de inicio

La fase de inicio trata de responder a estas preguntas más comunes que se debe realizar antes de iniciar un proyecto, sin embargo, no se pretende una estimación precisa o la captura de todos los requisitos más bien se trata de explorar el problema lo justo para decidir si va a continuar el proyecto, uno de los objetivos principales de esta fase es establecer el ámbito del proyecto y encontrar los escenarios básicos que definen la funcionalidad mediante el modelado de caso de uso.

✓ Fase de elaboración

El propósito de la fase de colaboración es analizar el dominio del problema, establecer los cimientos de la arquitectura, desarrollar el plan de proyecto y eliminar los mayores riesgos, en esta fase se construye un prototipo de la arquitectura, que debe evolucionar en iteraciones

sucesivas hasta convertirse en el sistema final. Cuando termina esta fase se llega al punto de no retorno del proyecto es a partir de ese momento pasamos de las relativamente ligeras y de poco riesgo dos primeras fases, a afrontar la fase de construcción.

✓ **Fase de construcción**

La finalidad principal de esta fase es alcanzar la capacidad operacional del producto de forma incremental a través de las sucesivas iteraciones, durante esta fase, todos los componentes, características y requisitos que hayan sido elaborados hasta ahora, serán implementados, integrados, obteniéndose una versión del producto que se pueda poner en manos de los usuarios.

✓ **Fase de transición**

El objetivo de esta fase es poner el producto en manos de los usuarios finales, para lo que típicamente se requerirá desarrollar nuevas versiones actualizadas del producto, completar la documentación, entrenar al usuario en el manejo del producto y en general, tareas relacionadas con el ajuste, configuración, instalación y usabilidad del producto.

2.4.2. Lenguaje UML

El lenguaje Unificado de Modelado (UML) es un lenguaje para especificar, visualizar, construir y documentar los artefactos de los sistemas software (OMG01). Esta se ha convertido en la notación visual estándar de facto para el modelado orientado a objetos. Comenzó como una iniciativa de Grady Booch y Jim Rumbaugh en 1994 para combinar las notaciones visuales de sus dos populares métodos, los métodos de Booch y OMT (Objet Modeling Technique). Más tarde se les unió Ivar Jacobson, el creador del método Objeto, y el grupo comenzó a ser conocido como los tres amigos. Muchos otros contribuyeron a UML, quizás más notablemente Cris Kobryn, que lidera el proceso de refinamiento que todavía continúa. (Larman C. , 2003)

2.4.2.1. Elementos estructurales

Según (Aларcon, 2000) A continuación, se van a describir todos los elementos que componen los bloques estructurales de UML, así como su notación, para que nos sirva de introducción y se vaya generando un esquema conceptual sobre UML.

- **Clases:** Una clase es una descripción de un conjunto de objetos que comparten los mismos atributos, operaciones, relaciones y semántica. Una clase implementa una o más interfaces. Gráficamente se representa como un rectángulo que incluye su nombre, sus atributos y sus operaciones.

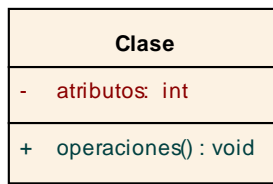


FIGURA 2.7: Clase
FUENTE: *Elaboración propia*

- **Interfaz:** Una interfaz es una colección de operaciones que especifican un servicio de una determinada clase o componente. Una interfaz describe el comportamiento visible externamente de ese elemento, puede mostrar el comportamiento completo o sólo una parte del mismo.



FIGURA 2.8: Interfaz
FUENTE: *Elaboración propia*

- **Colaboración:** Define una interacción y es una sociedad de roles y otros elementos que colaboran para proporcionar un comportamiento cooperativo mayor que la suma de los comportamientos de sus elementos. Las colaboraciones tienen una dimensión tanto estructural como de comportamiento. Una misma clase puede participar en diferentes colaboraciones. Las colaboraciones representan la implementación de

patrones que forman un sistema. Se representa mediante una elipse con borde discontinuo.

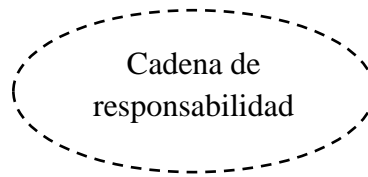


FIGURA 2.9: *Colaboración*
FUENTE: *Elaboración propia*

- **Casos de uso:** Un caso de uso es la descripción de un conjunto de acciones que un sistema ejecuta y que produce un determinado resultado que es de interés para un actor particular. Un caso de uso se utiliza para organizar los aspectos del comportamiento en un modelo. Un caso de uso es realizado por una colaboración. Se representa como en la figura 6, una elipse con borde continuo.



FIGURA 2.10: *Caso de uso*
FUENTE: *Elaboración propia*

- **Componentes:** Un componente es una parte física y reemplazable de un sistema que conforma con un conjunto de interfaces y proporciona la implementación de dicho conjunto. Un componente representa típicamente el empaquetamiento físico de diferentes elementos lógicos, como clases, interfaces y colaboraciones

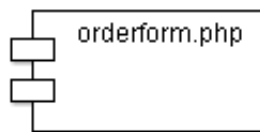


FIGURA 2.11: *Componente*
FUENTE: *Elaboración propia*

- **Nodo:** Un nodo es un elemento físico que existe en tiempo de ejecución y representa un recurso computacional que, por lo general, dispone de algo de memoria y, con frecuencia, de capacidad de procesamiento. Un conjunto de componentes puede residir en un nodo.

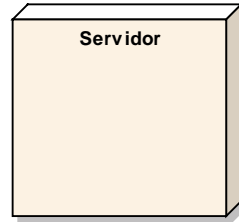


FIGURA: 2.12 *Nodo*
FUENTE: *Elaboración propia*

2.4.2.2. Elementos de comportamiento

Los elementos de comportamiento son las partes dinámicas de un modelo. Se podría decir que son los verbos de un modelo y representan el comportamiento en el tiempo y en el espacio. Los principales elementos son los dos que siguen.

- **Interacción:** Es un comportamiento que comprende un conjunto de mensajes intercambiados entre un conjunto de objetos, dentro de un contexto particular para conseguir un propósito específico. Una interacción involucra otros muchos elementos, incluyendo mensajes, secuencias de acción (comportamiento invocado por un objeto) y enlaces (conexiones entre objetos).
- **Máquinas de estados:** Es un comportamiento que especifica las secuencias de estados por las que van pasando los objetos o las interacciones durante su vida en respuesta a eventos, junto con las respuestas a esos eventos. Una máquina de estados involucra otros elementos como son estados, transiciones (flujo de un estado a otro), eventos (que disparan una transición) y actividades (respuesta de una transición)

2.4.2.3. Elementos de agrupación

Forman la parte organizativa de los modelos UML. El principal elemento de agrupación es el paquete, que es un mecanismo de propósito general para organizar elementos en grupos.

- **Paquete:** un paquete es puramente conceptual (sólo existe en tiempo de desarrollo). Gráficamente se representa como una carpeta conteniendo normalmente su nombre y, a veces, su contenido.

2.4.2.4. Elementos de anotación

Los elementos de anotación son las partes explicativas de los modelos UML. Son comentarios que se pueden aplicar para describir, clasificar y hacer observaciones sobre cualquier elemento de un modelo.

- **Nota:** es el principal tipo de anotación que simplemente es un símbolo para mostrar restricciones y comentarios junto a un elemento o un conjunto de elementos.

2.4.2.5. Relaciones

Existen cuatro tipos de relaciones entre los elementos de un modelo UML. Dependencia, asociación, generalización y realización, estas se describen a continuación:

- **Dependencia:** Es una relación semántica entre dos elementos en la cual un cambio a un elemento (el elemento independiente) puede afectar a la semántica del otro elemento (elemento dependiente). Se representa como una línea discontinua, posiblemente dirigida, que a veces incluye una etiqueta.
- **Asociación:** Es una relación estructural que describe un conjunto de enlaces, los cuales son conexiones entre objetos.
- **Composición:** Composición es un tipo especial de agregación que denota una fuerte posesión de la Clase “Todo”, a la Clase “Parte”. Se grafica con un rombo diamante relleno contra la clase que representa el todo.
- **Agregación:** La agregación es una relación en la que la clase “Todo” juega un rol más importante que la Clase "Parte", pero las dos clases no son dependientes una de otra. Se grafica con un rombo diamante vacío contra la Clase “Todo”. Multiplicidad y roles de los objetos involucrados.
- **Generalización:** Es una relación de especialización / generalización en la cual los objetos del elemento especializado (el hijo) pueden sustituir a los objetos del

elemento general (el padre). De esta forma, el hijo comparte la estructura y el comportamiento del padre.

- **Realización:** Es una relación semántica entre clasificadores, donde un clasificador especifica un contrato que otro clasificador garantiza que cumplirá. Se pueden encontrar relaciones de realización en dos sitios: entre interfaces y las clases y componentes que las realizan, y entre los casos de uso y las colaboraciones que los realizan.

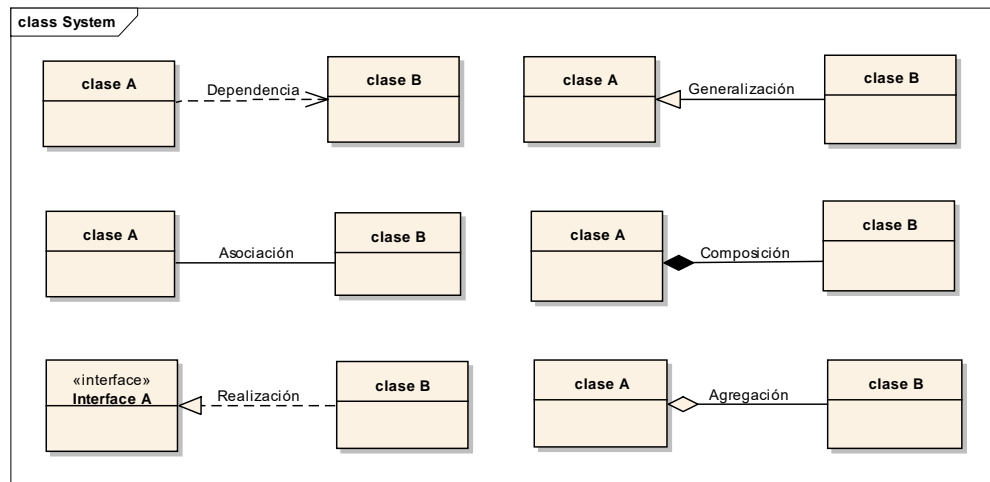


FIGURA: 2.13 Relaciones de clases UML
FUENTE: Elaboración propia

2.4.2.6. Diagramas de UML

(kenneth kendall, 2011). Los diagramas representan gráficamente un conjunto de elementos con sus relaciones, el lenguaje cuenta con nueve diagramas para visualizar el sistema desde varias perspectivas las cuales son las siguientes:

- **Diagramas de casos de uso:** Un caso de uso es una descripción de la secuencia de interacciones que se producen entre un actor y el sistema, cuando el actor usa el sistema para llevar a cabo una tarea específica. Expresa una unidad coherente de funcionalidad, y se representa en el Diagrama de Casos de Uso mediante una elipse con el nombre del caso de uso en su interior. El nombre del caso de uso debe reflejar la tarea específica que el actor desea llevar a cabo usando el sistema.

- **Diagrama de clases:** Las metodologías orientadas a objetos trabajan para descubrir las clases, atributos, métodos y relaciones entre las clases. Como la programación ocurre a nivel de clase, definir clases es una de las tareas más importantes del análisis orientado a objetos. Los diagramas de clases muestran las características estáticas del sistema y no representan ningún procesamiento en especial. Un diagrama de clases también muestra la naturaleza de las relaciones entre las clases.
- **Diagramas de Interacción:** Los diagramas de interacción muestran la manera en que interactúan entre si los objetos del sistema de información, existen dos tipos de diagramas de interacción, diagrama de secuencia y diagrama de colaboración.
 - **Diagrama de secuencia:** Los diagramas de secuencia pueden ilustrar una sucesión de interacciones entre clases o instancias de objetos a través del tiempo. A menudo, los diagramas de secuencia se derivan del análisis de casos de uso y se usan en diseño de sistemas para derivar las iteraciones, las relaciones los métodos de los objetos en el sistema. Los diagramas de secuencia se utilizan para mostrar el patrón general de las actividades o iteraciones en un caso de uso.
 - **Diagrama de colaboración:** Un diagrama de colaboración muestra una interacción organizada basándose en los objetos que toman parte en la interacción y los enlaces entre los mismos (en cuanto a la interacción se refiere). A diferencia de los diagramas de secuencia, los diagramas de colaboración muestran las relaciones entre los roles de los objetos.
- **Diagrama de estado:** El diagrama de estados o de transición de estados, es otra herramienta para determinar los métodos de las clases. Se utiliza para examinarlos distintos estados que puede tener un objeto.

Los objetos existen en estos diversos estados, que son las condiciones de un objeto definen el estado en que se encuentra el objeto y algunas veces hay un atributo tal como Estado del pedido; pendiente, en recolección, empaquetado, enviado, recibido, etcétera; que indica el estado.
- **Diagrama de actividad:** Los diagramas de actividad muestran la secuencia de actividades en un proceso, incluyendo las actividades secuenciales y paralelas,

además de las decisiones que se toman. Por lo general se crea un diagrama de actividad para un caso de uso y puede mostrar los distintos escenarios posibles.

- **Diagrama de componentes:** Un diagrama de componentes muestra las dependencias entre los componentes de software, incluye el código fuente, el código compilado y las imágenes de carga ejecutables.
- **Diagrama de despliegue:** Un diagrama de despliegue muestra sobre cual componente de hardware se instala (o despliega) cada componente de software. También muestra los enlaces de comunicación entre los componentes de hardware.

Para la elaboración del modelado del sistema se utiliza los diagramas de casos de uso, diagrama de clases, diagrama de secuencia, diagrama de componentes y diagrama de despliegue.

2.4.3. Herramientas de programación

2.4.3.1. Hypertext pre-processor

(PHP.NET, 2012). PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas, es decir, olvidemos de las eventuales páginas o sitios estáticos que realizábamos anteriormente en HTML ahora con PHP nos permite una conexión con bases de datos y volver dinámico nuestro sitio.

Los investigadores razonan que PHP es un lenguaje de programación dinámica para la realización de páginas web, como también es más seguro su codificación de esta manera no permitiendo la visualización de su código de programación.

✓ Características

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.

- Tiene manejo de excepciones desde PHP5.

2.4.3.2. Hyper text markup language

Según (Gaucht, 2012) HTML es un lenguaje de etiquetas, un listado de elementos que usualmente se utilizan en pares y que pueden ser anidados (totalmente contenidos uno dentro del otro). HTML5 provee básicamente tres características: estructura, estilo y funcionalidad. Nunca fue declarado oficialmente, pero, incluso cuando algunas APIs (Interface de Programación de Aplicaciones) y la especificación de CSS3 por completo no son parte del mismo, HTML5 es considerado el producto de la combinación de HTML, CSS y Javascript. Estas tecnologías son altamente dependientes y actúan como una sola unidad organizada bajo la especificación de HTML5. HTML está a cargo de la estructura, CSS presenta esa estructura y su contenido en la pantalla y Javascript hace el resto es extremadamente significativo. (Luca, 2011)

2.4.3.3. Hojas de estilo (CSS)

Oficialmente CSS nada tiene que ver con HTML5. CSS no es parte de la especificación y nunca lo fue. Este lenguaje es un complemento desarrollado para superar las limitaciones y reducir la complejidad de HTML. Al comienzo, atributos dentro de las etiquetas HTML proveían estilos esenciales para cada elemento, pero a medida que el lenguaje evolucionó, la escritura de códigos se volvió más compleja y HTML por sí mismo no pudo más satisfacer las demandas de diseñadores. En consecuencia, CSS pronto fue adoptado como la forma de separar la estructura de la presentación. Desde entonces, CSS ha crecido y ganado importancia, pero siempre desarrollado en paralelo, enfocado en las necesidades de los diseñadores y apartado del proceso de evolución de HTML. (Gaucht, 2012)

2.4.3.4. Base de datos MySQL

(Gilfillan, 2010). Es la base de datos de código abierto más popular del mundo. Código abierto significa que todo el mundo puede acceder al código fuente, es decir, al código de programación de MySQL. Todo el mundo puede contribuir para incluir elementos, arreglar problemas, realizar mejoras o sugerir optimizaciones. Y así ocurre. MySQL ha pasado de ser una "pequeña" base de datos a una completa herramienta y ha conseguido superar a una gran

cantidad de bases de datos comerciales (lo que ha asustado a la mayor parte de los proveedores comerciales de bases de datos). Por lo tanto, su rápido desarrollo se debe a la contribución de mucha gente al proyecto, así como a la dedicación del equipo de MySQL.

Son muchas las razones para escoger MySQL como solución de misión crítica para la administración de datos:

- Coste: El coste es gratuito para la mayor parte de los usos y su servicio de asistencia resulta económico.
- Asistencia: Ofrece contratos de asistencia a precios razonables y existe una nutrida y activa comunidad MySQL.
- Velocidad: Es mucho más rápido que la mayor parte de sus rivales.
- Portabilidad: Se ejecuta en la inmensa mayoría de sistemas operativos y, la mayor parte de los casos, los datos se pueden transferir de un sistema a otro sin dificultad.
- Facilidad de uso: Resulta fácil de utilizar y de administrar. Gran parte de las viejas bases de datos presentan problemas por utilizar sistemas obsoletos, lo que complica innecesariamente las tareas de administración.
- Las herramientas de MySQL son potentes y flexibles, sin sacrificar su capacidad de uso.

2.4.3.5. XAMPP

Es un paquete de instalación que integra en una sola aplicación, un servidor web Apache, intérpretes de lenguaje de scripts PHP, un servidor de base de datos MySQL, el popular administrador de base de datos escrito en PHP, MySQL, entre otros módulos. Permite instalar de forma sencilla Apache en tu propio ordenador, sin importar tu sistema operativo (Linux, Windows, MAC o Solaris). Y lo mejor de todo es que su uso es gratuito. Es una herramienta de desarrollo que permite probar el trabajo (páginas web o programación, por ejemplo) en el ordenador sin necesidad de tener que acceder a internet. (Marco, 2014)

2.4.3.6. Framework Laravel

Laravel es un framework de PHP brillante que está equipado con un montón de características interesantes, incluyendo enrutamiento RESTful, PHP nativo motor ligero y muchos más. Construido con varios componentes de Symfony, laravel ofrece a las aplicaciones web una increíble base de código confiable y bien probado. Algunas otras características interesantes de laravel incluyen: un poderoso conjunto de librerías, un ORM increíble, fácil enrutamiento y una autenticación simple. (Ramos, 2015)

Los componentes de laravel son los siguientes (Miguel Angel Alvarez, 2010):

- ✓ **Las vistas** no son más que los archivos PHP desde donde tenemos que realizar la salida de la aplicación. Es un concepto que esperamos que ya se tenga en la cabeza cuando estamos introduciéndonos en Laravel, puesto que no pertenece en sí al framework PHP sino al MVC en general. Son una de las capas que tiene el sistema MVC, que trata de la separación del código según sus responsabilidades. En este caso, las vistas mantienen el código de lo que sería la capa de presentación.
- ✓ **Controladores.** Su función es la de definir el código a ejecutar como comportamiento frente a una acción solicitada dentro de la aplicación. Generalmente para poder desempeñar su labor se apoyan en los modelos y las vistas. El controlador sabe qué métodos del modelo debe invocar, ya sea para actualizar cierta información o para obtener ciertos datos, así como las vistas que deben presentar la información como respuesta al usuario, después de la realización de las acciones necesarias.
- ✓ **Los Modelos** son uno de los componentes principales de las aplicaciones desarrolladas bajo el patrón MVC, que tienen la responsabilidad de acceder a los datos, modificarlos, etc. En el patrón además los modelos mantienen lo que se llama la lógica de negocio, que son las reglas que deben cumplirse para trabajar con los datos. En Laravel los modelos se controlan por un ORM llamado Eloquent, al menos los modelos que están implementados como datos en una base de datos, pero no es un requisito, de modo que podríamos trabajar con otros ORM o incluso bajar a un nivel más bajo y trabajar con PDO directamente, o con las extensiones de nuestra base de datos en particular.

2.4.3.7. Seguridad del sistema

Protección de rutas y autenticación de usuarios. - El objetivo de la seguridad de Laravel es impedir que los usuarios que no tengan suficientes permisos no accedan a ciertas áreas de la aplicación. Pueden utilizarse filtros en las rutas de la aplicación para permitir que sólo los usuarios autenticados o con suficientes permisos puedan acceder a la ruta indicada. Laravel proporciona el filtro auth por defecto, pero pueden crearse tantos filtros como se desee.

2.4.3.8. Seguridad de la aplicación

Por otra parte, Laravel implementa internamente defensas ante ataques externos en la aplicación. Al utilizar Eloquent y Fluent Query Builder en los modelos de datos, protege la base de datos de ataques SQL injection escapeando cualquier tipo de input que pueda entrar por los formularios intentando insertar código SQL malicioso.

Laravel utiliza una llave de autenticación, única por cada aplicación, para encriptar las cookies asegurándose de que no se puedan modificar externamente.

También proporciona protección ante falsificaciones de peticiones (CSRF), asignaciones masivas y ataques por secuencia de comandos (XSS).

2.4.3.9. Métricas de calidad

(Pressman R. S., 1998). Los desarrolladores de software más hastiados estarán de acuerdo que el software de alta calidad es una de las metas más importantes. La calidad del software es una compleja mezcla de factores que varían a través de diferentes aplicaciones y según el cliente que las pida.

La calidad del software se define también, en términos de ausencia de errores en el funcionamiento del sistema. El ajuste a las necesidades del usuario, el sistema debe ser flexible u susceptible a modificaciones que se puedan realizar de manera rápida y oportuna. El sistema debe alcanzar un desempeño apropiado en términos de tiempo, volumen y espacio.

2.4.4. Factores de la calidad ISO 9126

El estándar ISO 9126 se desarrolló con la intención de identificar los atributos clave del software de cómputo. Este sistema identifica seis atributos clave de la calidad:

- **Funcionalidad.** Grado en el que el software satisface las necesidades planteadas según las establecen los atributos siguientes: adaptabilidad, exactitud, interoperabilidad, cumplimiento y seguridad.
- **Confiabilidad.** Cantidad de tiempo que el software se encuentra disponible para su uso, según lo indican los siguientes atributos: madurez, tolerancia a fallas y recuperación.
- **Usabilidad.** Grado en el que el software es fácil de usar, según lo indican los siguientes sub atributos: entendible, aprendible y operable.
- **Eficiencia.** Grado en el que el software emplea óptimamente los recursos del sistema, según lo indican los sub atributos siguientes: comportamiento del tiempo y de los recursos.
- **Facilidad de recibir mantenimiento.** Facilidad con la que pueden efectuarse reparaciones al software, según lo indican los atributos que siguen: analizable, cambiable, estable, susceptible de someterse a pruebas.
- **Portabilidad.** Facilidad con la que el software puede llevarse de un ambiente a otro según lo indican los siguientes atributos: adaptable, instalable, conformidad y sustituible.

Igual que otros factores de la calidad del software estudiados en las subsecciones anteriores, los factores ISO 9126 no necesariamente conducen a una medición directa. Sin embargo, proporcionan una base útil para hacer mediciones indirectas y una lista de comprobación excelente para evaluar la calidad del sistema. (Pressman R. , 2002)

CAPITULO III

3. MARCO APLICATIVO

3.1. FASES DE LA METODOLOGÍA DEL PROCESO UNIFICADO

En este capítulo se muestra la descripción del ciclo del desarrollo del " Sistema Informático para la Administración del Diagnóstico de Insuficiencia Renal ", que hace uso de instrumentos métodos y técnicas descritas en el capítulo anterior "Marco teórico".

Se realizó el lineamiento del Proceso Unificado de Desarrollo de Software, utilizando el Lenguaje de Modelado Unificado (UML) representándose en los esquemas de un sistema de software de acuerdo los modelos descritos por RUP.

3.1.1. Fase de inicio

En esta fase de acuerdo a la metodología y siguiendo los pasos de las disciplinas, se realizaron entrevistas y observaciones a la unidad de enfermedades no transmisibles del SEDES Pando con la finalidad de obtener el flujo de información para el control de pacientes, para adquirir los requerimientos del sistema.

3.1.1.1. Modelo de análisis

El modelo de análisis es el estudio del funcionamiento de las acciones que ejerce la unidad de enfermedades no transmisibles.

En el diagrama de la Figura 3.14, se representa el análisis de flujo de actividades en la entidad determinando el modelo de proceso de negocio, donde comienza con el ingreso de expedientes de pacientes de los diferentes centros de salud del municipio de Cobija, luego son registrados todos los datos del paciente, tomando en cuenta principalmente, la información de los datos de análisis, que muestran en los expedientes de cada paciente.

Se hace un análisis de estos datos y se da como resultado el diagnóstico del paciente.

✓ Modelo de proceso de negocio de la entidad de ENT

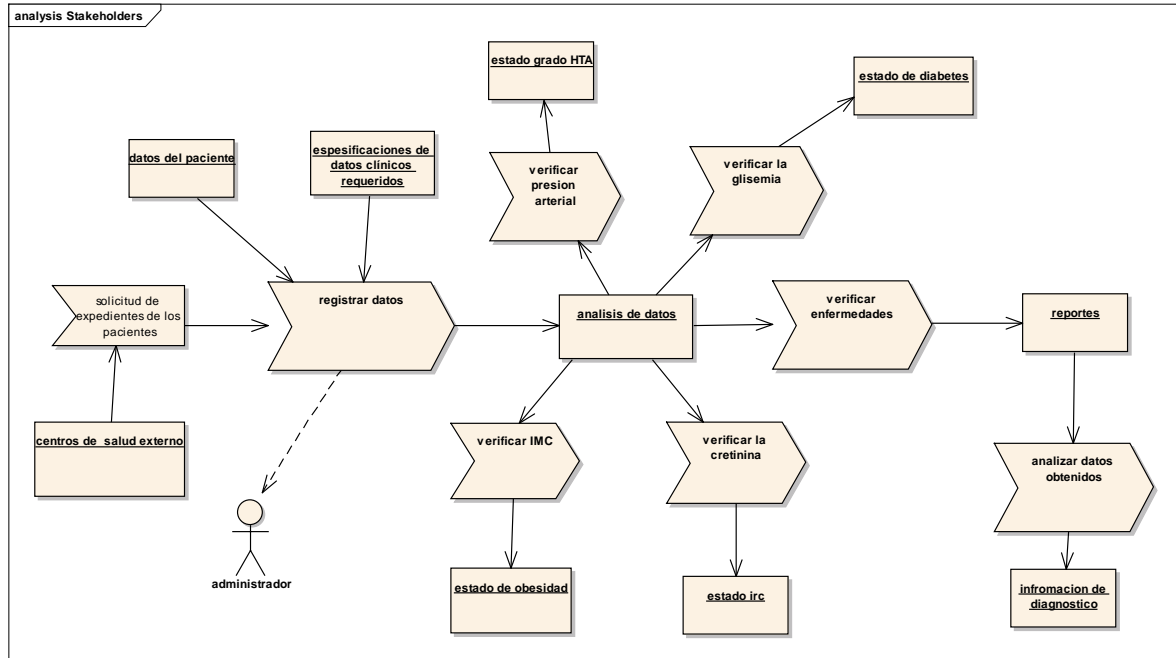


FIGURA 3.14: Modelo de Proceso de Negocio de la unidad de Enfermedades no Transmisibles
FUENTE: Elaboración propia

✓ Análisis de la unidad

La unidad de Enfermedades no Transmisibles del SEDES Pando se encarga de registrar todos los datos de pacientes de los diferentes centros de salud del departamento de Pando, para poder mostrar resultados de los de sus análisis de pacientes siendo como sede central municipio de Cobija. Estos procesos se realizan manualmente por el encargado, guardando los registros en hojas de cálculo de Excel lo que representa dificultad en el momento de dar el diagnóstico correspondiente del paciente.

✓ Involucrados

Administrador Es el director responsable de la unidad de enfermedades no transmisibles, es la persona que necesita estar informado teniendo un control y seguimiento de los datos de los pacientes.

Secretaria Es la que recibe los expedientes que llegan de los pacientes, posteriormente realiza el registro.

- ✓ **Unidades proveedoras de datos** son todos los centros de salud de Cobija que hacen llegar los datos de los pacientes.

3.1.1.2. Modelo inicial de los requerimientos

Los requerimientos son declaraciones que identifican atributos, capacidades, características o cualidades que necesita cumplir un sistema de software para que tenga valor y utilidad para el usuario. En otras palabras, los requerimientos muestran qué elementos y funciones son necesarias para el proyecto.

Nº	REQUERIMIENTOS
R1	Autenticar Usuario
R3	Control de Pacientes
R4	Control de Enfermedades no Transmisibles

*TABLA 3.3: Modelo inicial de requerimientos
FUENTE: Elaboración propia*

3.1.1.3. Requerimientos funcionales

El sistema verifica la autenticación de ingreso a este por parte de los usuarios autorizados y que el acceso se ha realizado con éxito y que la conexión a la base de datos está disponible. El sistema recibe los datos que los pacientes almacenándolos en la base de datos.

Para el administrador, es preciso que el login de usuario se realice en la página de inicio, desde un formulario que solicite el un nombre de usuario y contraseña para ingresar, si es nuevo se debe registrar desde un formulario que solicite sus datos personales.

✓ **Requerimiento de usuario**

El sistema verifica que el acceso del mismo ha sido correcto y que la conexión de la base de datos se encuentra disponible.

- **Login del sistema**

Se debe registrar todos los sucesos que ocurre en el sistema a fin de reconstruir todas las actividades realizadas en el sistema.

3.1.1.4. Requerimientos no funcionales.

Los requisitos no funcionales describen los atributos que tiene el software (portabilidad, eficiencia, confiabilidad, robustez, rendimiento, etc.)

	Requisito	Descripción
1	Interfaz para el usuario	Diseño sencillo, una interfaz simple de usar e interactiva para que al usuario le sea fácil el trabajo con el sistema.
2	Software	<ul style="list-style-type: none">• La aplicación se hospeda en un servidor web Apache.• El servidor de base de datos debe ser Mysql.• La máquina debe tener un navegador instalado, preferentemente Mozilla Firefox. (Versiones recientes).
3	Manejo	La aplicación se podrá ejecutar en la mayoría de los sistemas operativos tales como Microsoft Windows XP/7 y Linux

TABLA 3. 4: *Requisitos no funcionales*
FUENTE: *Elaboración propia*

3.1.1.5. Identificación de los actores

Siguiendo metodología y en la captura de requisitos se identificaron a los siguientes actores.

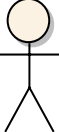
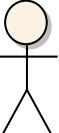
Actores	Descripción
 Administrador	Administrador: Es responsable de la unidad de enfermedades no transmisibles.
 Secretaria	Secretaria: Realiza la tarea de recepcionar los expedientes de pacientes de los distintos centros de salud.

TABLA 3.5 *Actores del sistema*
FUENTE: *Elaboración propia*

- ✓ **Administrador:** Es el usuario con más privilegios sobre el sistema. Permite utilizar en su totalidad todas las opciones disponibles del sistema.
- ✓ **Secretaria:** Es la persona encargada de registrar eliminar, modificar los pacientes, también genera los reportes del diagnóstico del paciente.

3.1.1.6. Modelo de casos de uso inicial control paciente

En este diagrama de caso de uso se representa el registro del paciente al Sistema, modificación y eliminación, a criterio del administrador.

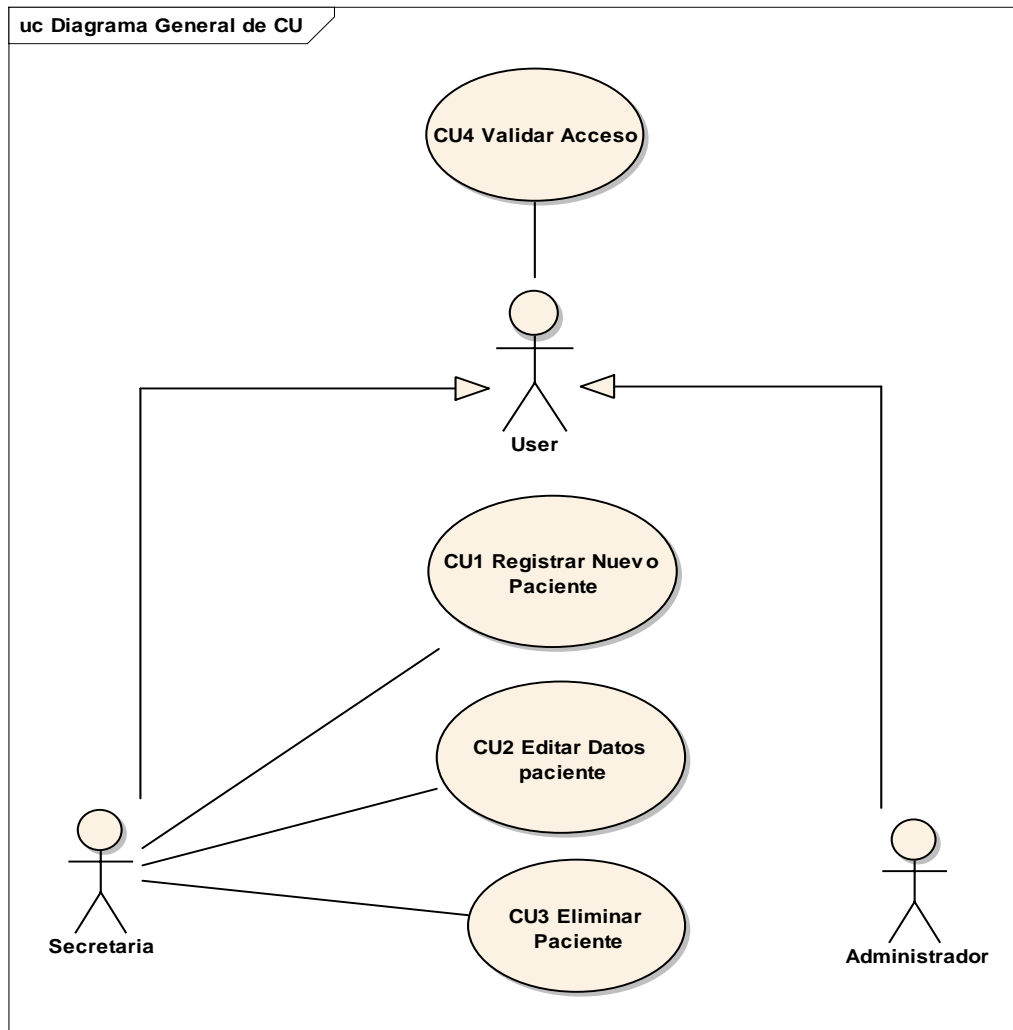


FIGURA 3.15: Diagrama de caso de uso inicial control paciente
FUENTE: Elaboración propia

3.1.1.7. Diagrama de actividades registrar paciente

Diagrama de actividades de actividades que muestra como principal flujo el registro del paciente.

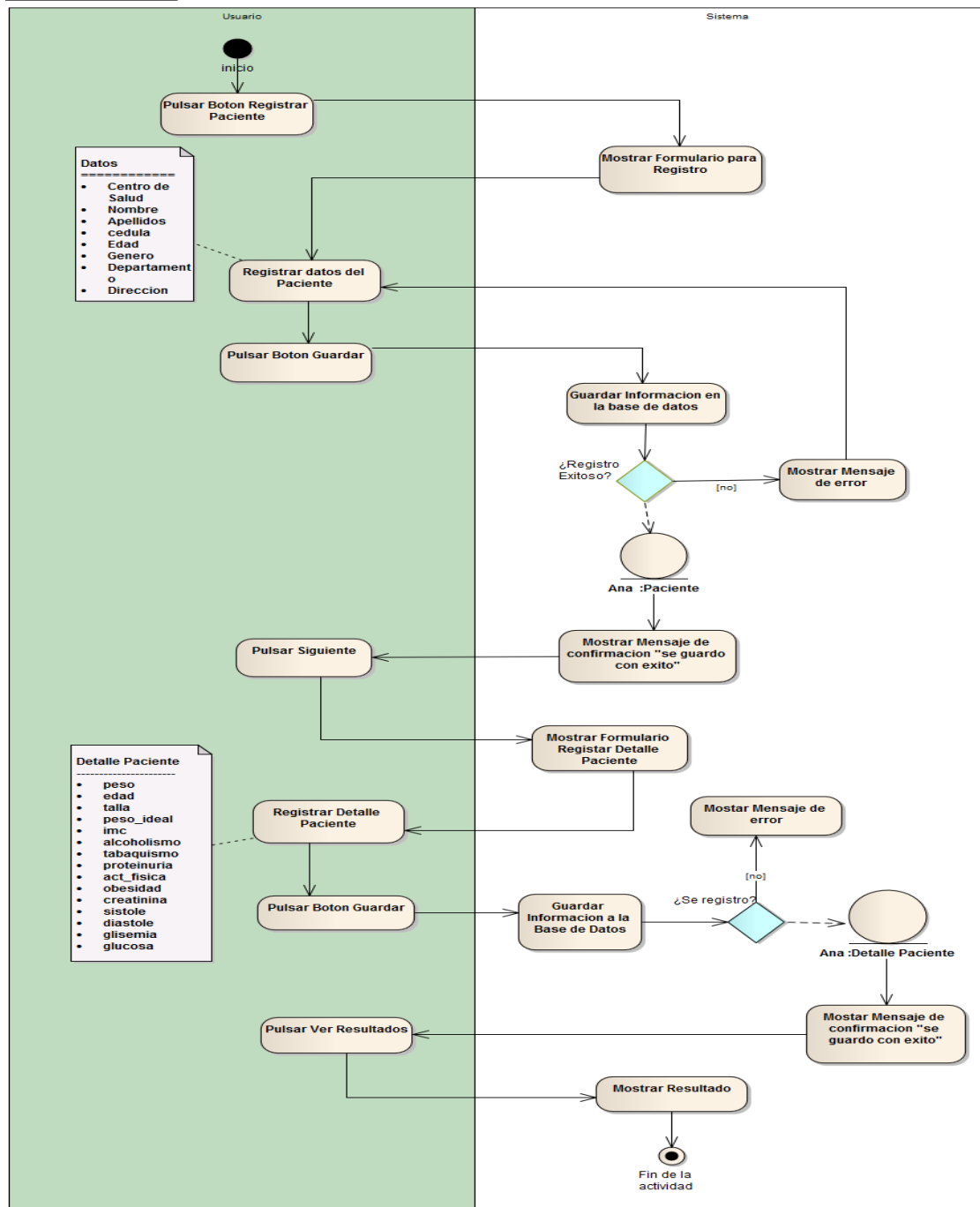


FIGURA 3.16: Diagrama de actividades registrar paciente
FUENTE: Elaboración propia

3.1.2. Fase de elaboración

El objetivo de esta fase es formular la línea base de la arquitectura. Se toma en cuenta el modelo de requerimientos terminado, también implica desarrollar alrededor del 80 por ciento de los casos de uso, abordar los riesgos que interfieran en la consecución de este objetivo. En esta fase acrecentaremos el entorno de desarrollo para la fase de construcción.

3.1.2.1. Modelo de requerimientos terminado

Un proyecto no puede ser exitoso sin una especificación correcta de los requerimientos, donde describe las necesidades o deseos de un producto. En la Tabla 2 vista anteriormente se muestran los requerimientos aplicados para el sistema.

N °	REQUERIMIENTOS	DESCRIPCIÓN
R1	Autenticar Usuario	
R2	Administrar Usuario	Registrar usuario Modificar usuario Eliminar usuario
R3	Control de Pacientes	Agregar paciente Modificar paciente Listar paciente Realizar reportes
R4	Control de Enfermedades no Transmisibles	Registrar presión arterial Registrar diabetes Registrar glucosa Registrar diástole Registrar sístole

R5	Control de centros de salud	Registrar centro de salud Modificar centro de salud Listar centro de salud
-----------	------------------------------------	--

TABLA 3.6: Requerimientos
FUENTE: Elaboración propia

3.1.2.2. Diagramas de casos de uso

El sistema informático de diagnóstico está constituido por los siguientes casos de uso.

➤ Diagrama de caso de uso autenticar CU1

En este Diagrama de casos de uso hace referencia a la descripción de las acciones del sistema desde el punto de vista del usuario, se describen las funcionalidades del sistema mediante el caso de uso general del sistema, resultante al proceso estudiado anteriormente del diagrama de modelo del negocio.

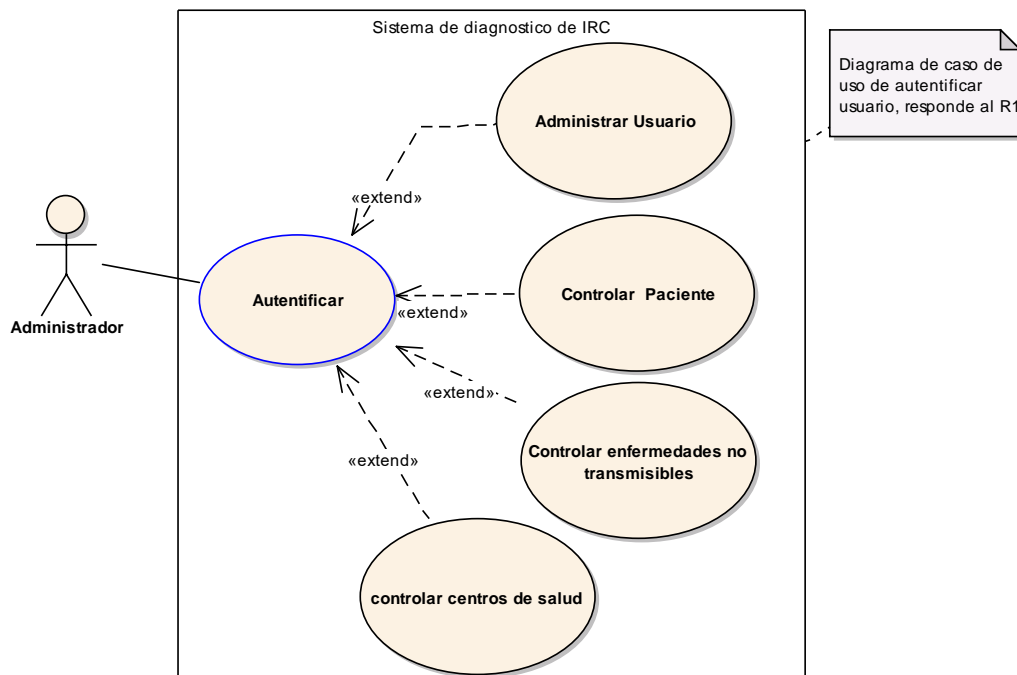


FIGURA 3.17: Casos de uso autenticar usuario
FUENTE: Elaboración propia

➤ **Diagrama de caso de uso control usuario CU2**

En este diagrama se representa el registro de datos del usuario del Sistema, al igual que se adicionan nuevos usuarios, estos pueden ser modificados y eliminados a criterio del administrador quien es el único que tiene el acceso a trabajar en dicho modulo.

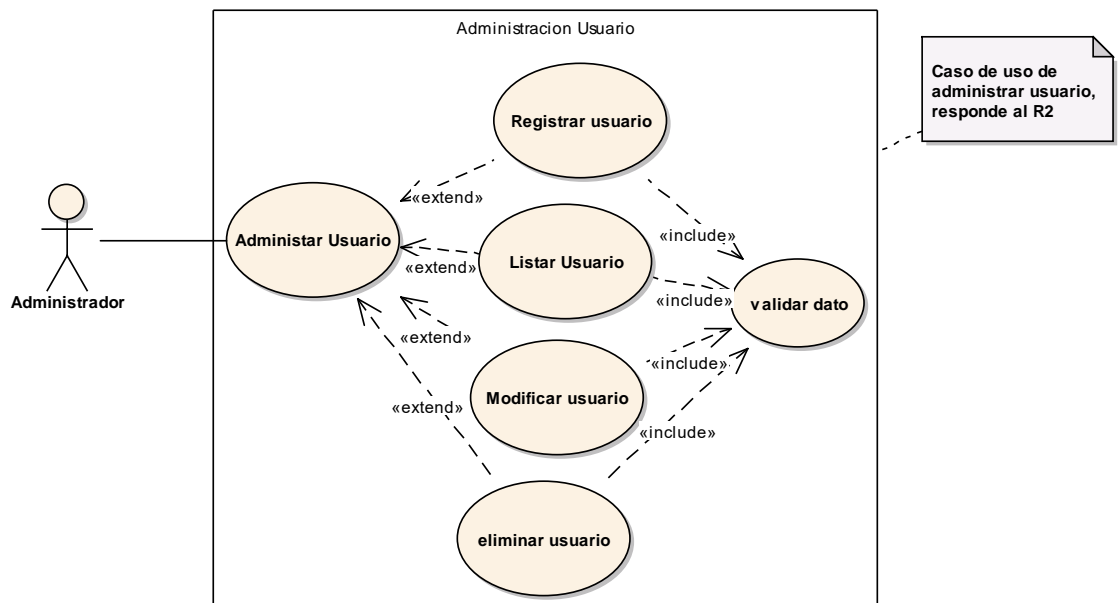


FIGURA 3.18 : *Caso de uso administrar usuario*
Fuente: *Elaboración propia*

➤ **Descripción del caso de uso CU2**

Caso de Uso:	Administrar de Usuario
Actores:	Administrador
Propósito:	Administrar los datos de los usuarios, permitiendo la creación, actualización y/o eliminación del administrador y de los demás usuarios de la unidad de enfermedades no transmisibles
Flujo de Evento:	<p>Registrar usuario</p> <ul style="list-style-type: none"> • Introducir los datos del nuevo usuario. • Validar datos. • Guardar registro del usuario. <p>Modificar usuario</p> <ul style="list-style-type: none"> • Introducir código de usuario. • Actualizar cambios. • Guardar el registro. <p>Reporte de usuario</p> <ul style="list-style-type: none"> • Mostrar datos del Empleado.

TABLA 3.7: *Caso de uso administrar usuario*
FUENTE: *Elaboración propia*

➤ **Diagrama de caso de uso controlar paciente CU3**

En este diagrama se muestra las altas, bajas y modificación del paciente.

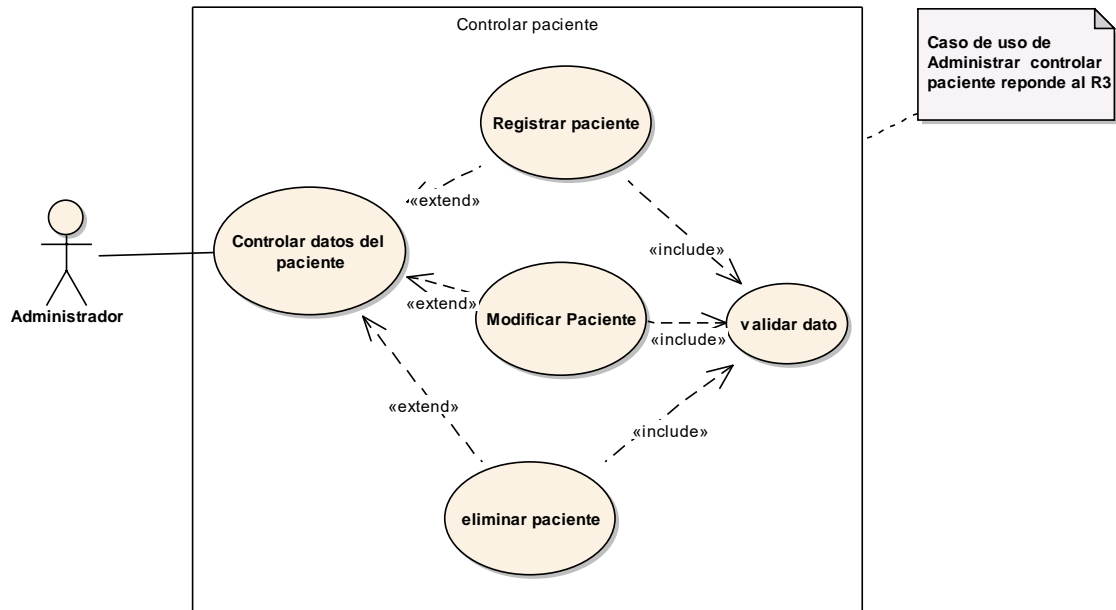


FIGURA 3.19: *Caso de uso controlar paciente*
Fuente: *Elaboración propia.*

➤ **Descripción de caso de uso CU3**

Caso de Uso:	Controlar paciente
Actores:	Administrador, Secretaria
Propósito:	Administrar los datos de los pacientes, permitiendo la creación, actualización y/o eliminación de los pacientes de los Centros de Salud.
Flujo de Evento:	<p>Registrar paciente</p> <ul style="list-style-type: none"> • Introducir los datos del Paciente. • Validar datos. • Guardar registro del Paciente. <p>Modificar paciente</p> <ul style="list-style-type: none"> • Introducir datos del paciente.

	<ul style="list-style-type: none"> • Actualizar cambios. • Guardar el registro. <p>Reporte de paciente</p> <ul style="list-style-type: none"> • Mostrar datos del paciente.
--	---

TABLA 3.8: Descripción del caso de uso controlar paciente
FUENTE: Elaboración propia

➤ **Diagrama de caso de uso controlar enfermedades CU4**

En este diagrama se muestra el registro para control de las enfermedades.

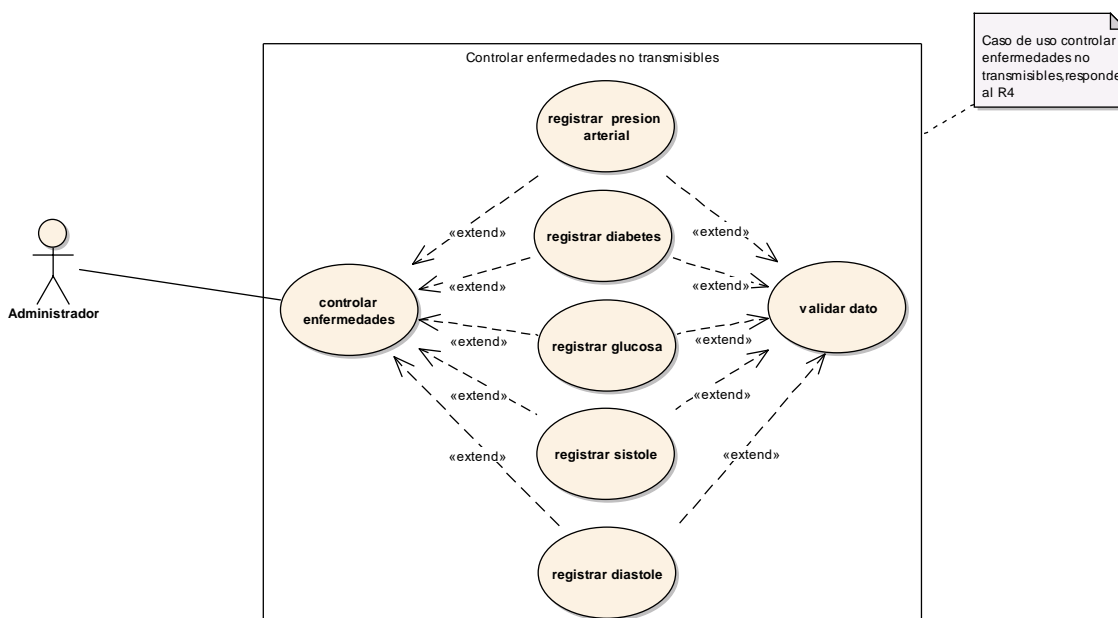


FIGURA 3.20: Caso de uso controlar enfermedades
FUENTE: Elaboración propia.

➤ Descripción de caso de uso controlar enfermedades CU4

Caso de Uso:	Registrar presión arterial
Actores:	Administrador
Propósito:	Administrar los datos de la enfermedad de presión arterial, permitiendo la creación, actualización de los datos.
Flujo de Evento:	<p>Nuevo dato presión arterial</p> <ul style="list-style-type: none"> • Introducir los datos de la presión arterial. • Validar datos. • Guardar registro de la presión arterial <p>Modificar presión arterial</p> <ul style="list-style-type: none"> • Introducir datos. • Actualizar cambios. • Guardar el registro. <p>Reporte de la presión arterial</p> <p>Mostrar datos de la presión arterial.</p>
Caso de Uso:	Registrar diabetes
Actores:	Administrador
Propósito:	Administrar los datos de la enfermedad de diabetes, permitiendo la creación, actualización de los datos.
Flujo de Evento:	<p>Nuevo dato diabetes</p> <ul style="list-style-type: none"> • Introducir los datos del estado diabetes. • Validar datos. • Guardar registro del estado diabetes. <p>Modificar diabetes</p> <ul style="list-style-type: none"> • Introducir datos. • Actualizar cambios.

	<ul style="list-style-type: none"> • Guardar el registro. <p>Reporte de diabetes</p> <p>Mostrar datos del estado diabetes.</p>
Caso de Uso:	Registrar glucosa
Actores:	Administrador
Propósito:	Administrar los datos de la glucosa, permitiendo la creación, actualización y/o eliminación de los datos de las enfermedades no transmisibles
Flujo de Evento:	<p>Registrar glucosa</p> <ul style="list-style-type: none"> • Introducir los datos de la glucosa. • Validar datos. • Guardar registro de la glucosa. <p>Modificar glucosa</p> <ul style="list-style-type: none"> • Introducir datos. • Actualizar cambios. • Guardar el registro. <p>Reporte de la glucosa</p> <p>Mostrar datos de la glucosa.</p>
Caso de Uso:	Registrar diástole
Actores:	Administrador
Propósito:	Controlar los datos del valor diastólico, permitiendo la creación, actualización y/o eliminación de los datos.
Flujo de Evento:	<p>Nuevo dato diástole</p> <ul style="list-style-type: none"> • Introducir los datos de la diástole. • Validar datos. • Guardar registro de la diástole. <p>Modificar diástole</p>

	<ul style="list-style-type: none"> • Introducir datos. • Actualizar cambios. • Guardar el registro. <p>Reporte de datos</p> <p>Mostrar datos de la diástole.</p>
Caso de Uso:	Registrar sístole
Actores:	Administrador
Propósito:	Controlar los datos de la sístole, permitiendo la creación, actualización y/o eliminación de los datos.
Flujo de Evento:	<p>Nuevo dato siástole</p> <ul style="list-style-type: none"> • Introducir los datos de la sístole. • Validar datos. • Guardar registro de la sístole. <p>Modificar sístole</p> <ul style="list-style-type: none"> • Introducir datos. • Actualizar cambios. • Guardar el registro. <p>Reporte de datos</p> <p>Mostrar datos de la sístole.</p>

TABLA 3.9: Descripción del caso de uso controlar enfermedades
FUENTE: Elaboración propia

➤ **Diagrama de caso de uso controlar centros de salud CU5**

En este diagrama se muestra el registro modificación y eliminación de los centros de salud,

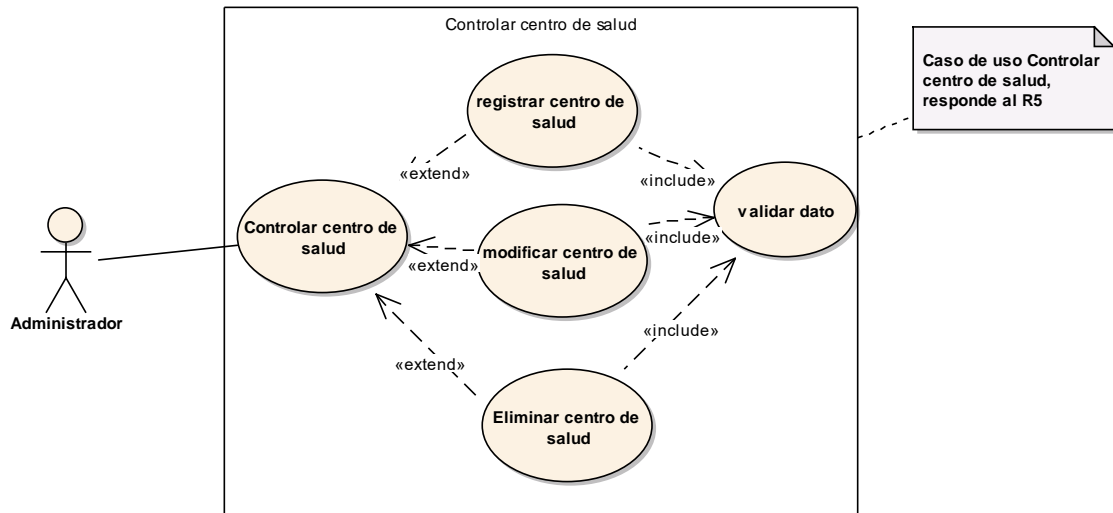


FIGURA 3.21: Caso de uso controlar centro de Salud
FUENTE: Elaboración propia

➤ **Descripción del caso de uso CU5**

Caso de Uso:	Controlar Centros de Salud
Actores:	Administrador, Secretaria
Propósito:	Controlar los datos de los centros de salud, permitiendo la creación, actualización y/o eliminación de los centros de salud de la ciudad de Cobija que forman parte de la intervención de la unidad de enfermedades no transmisibles
Flujo de Evento:	<p>Nuevo centro de salud</p> <ul style="list-style-type: none"> • Introducir los datos del Centro de Salud. • Validar datos. • Guardar registro del Centro de salud. <p>Modificar centro de Salud</p> <ul style="list-style-type: none"> • Introducir datos del centro de salud. • Actualizar cambios.

	<ul style="list-style-type: none"> • Guardar el registro. <p>Reporte de datos</p> <ul style="list-style-type: none"> • Mostrar datos del Centro de Salud.
--	--

TABLA 3.10: Descripción caso de uso administrar centro de salud
FUENTE: Elaboración propia

3.1.2.3. Diagrama de clases

➤ Diagrama de clases autenticar usuario

En la figura 3.22, diagrama de clases usuario donde comienza en la página principal de autenticación de usuario capturando los datos en el formulario, realiza el control de autenticación de usuario llamando a los métodos de la clase Usuario manteniendo la continua conexión a la Base de Datos.

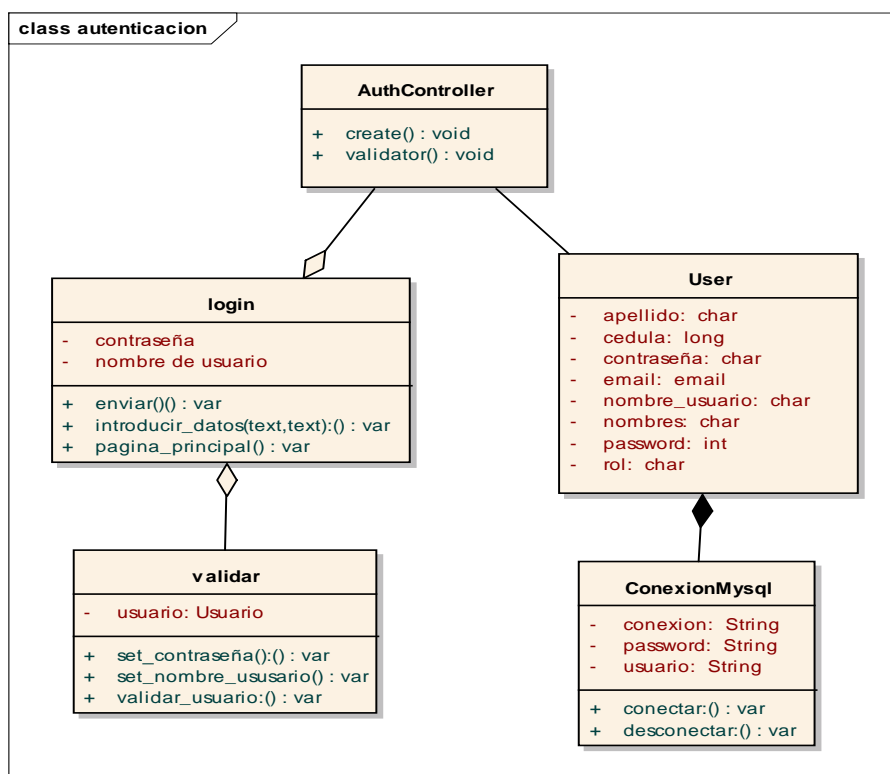


FIGURA 3.22: Diagrama de clases autenticación
FUENTE: Elaboración propia

➤ Diagrama de clases de control de usuario

En la figura 3.23, diagrama de clases usuario, se muestra las tres funcionalidades que se realiza dentro del módulo control usuario que son registrar, modificar y listar, cada una llama a la clase UserController realizando internamente la parte lógica del sistema pasa a llamar los métodos de la clase Usuario manteniendo la continua conexión a la base de datos.

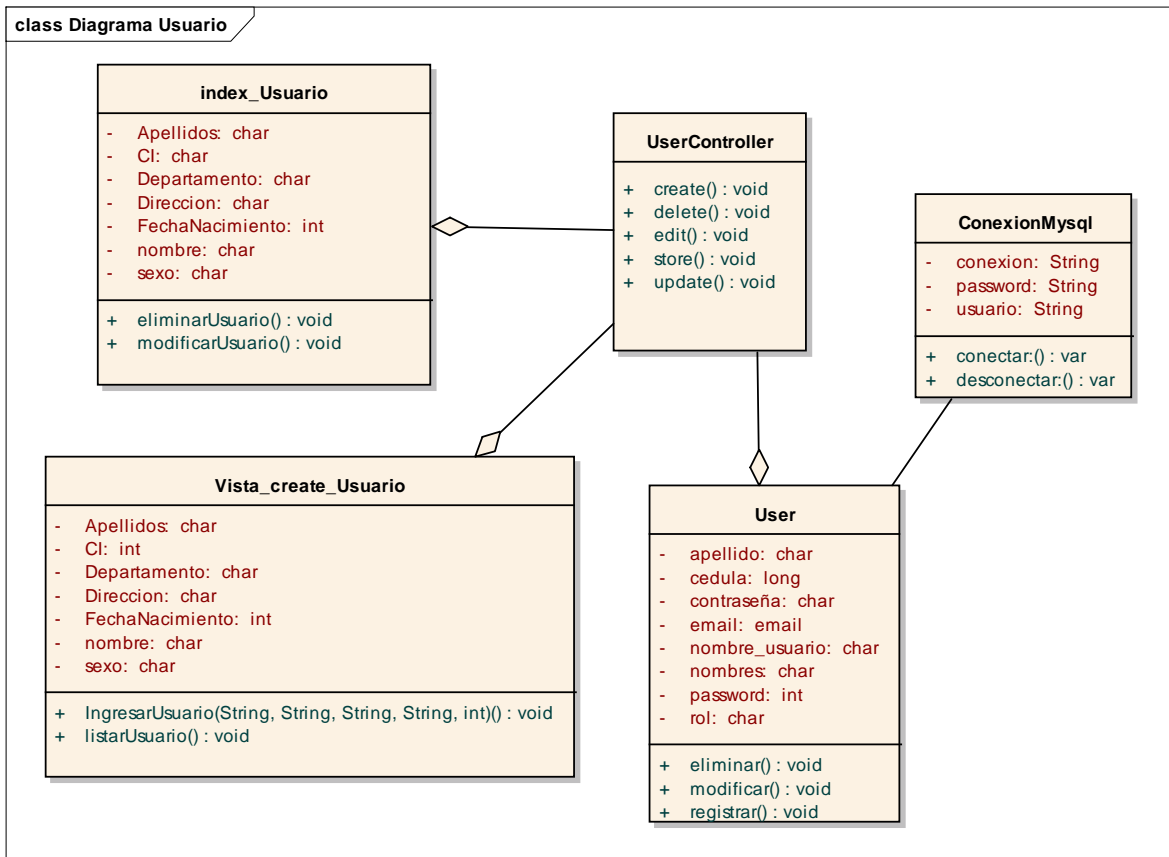


FIGURA 3.23: Diagrama de clases usuario
FUENTE: Elaboración propia

➤ Diagrama de clases controlar paciente

En la figura 3.24, diagrama de clases paciente, se muestra las tres funcionalidades que se realiza dentro del módulo control paciente que son registrar, modificar y listar, cada una llama a la clase PacienteController realizando internamente la parte lógica del sistema pasa a llamar los métodos de la clase paciente y detalle paciente manteniendo la continua conexión a la base de datos.

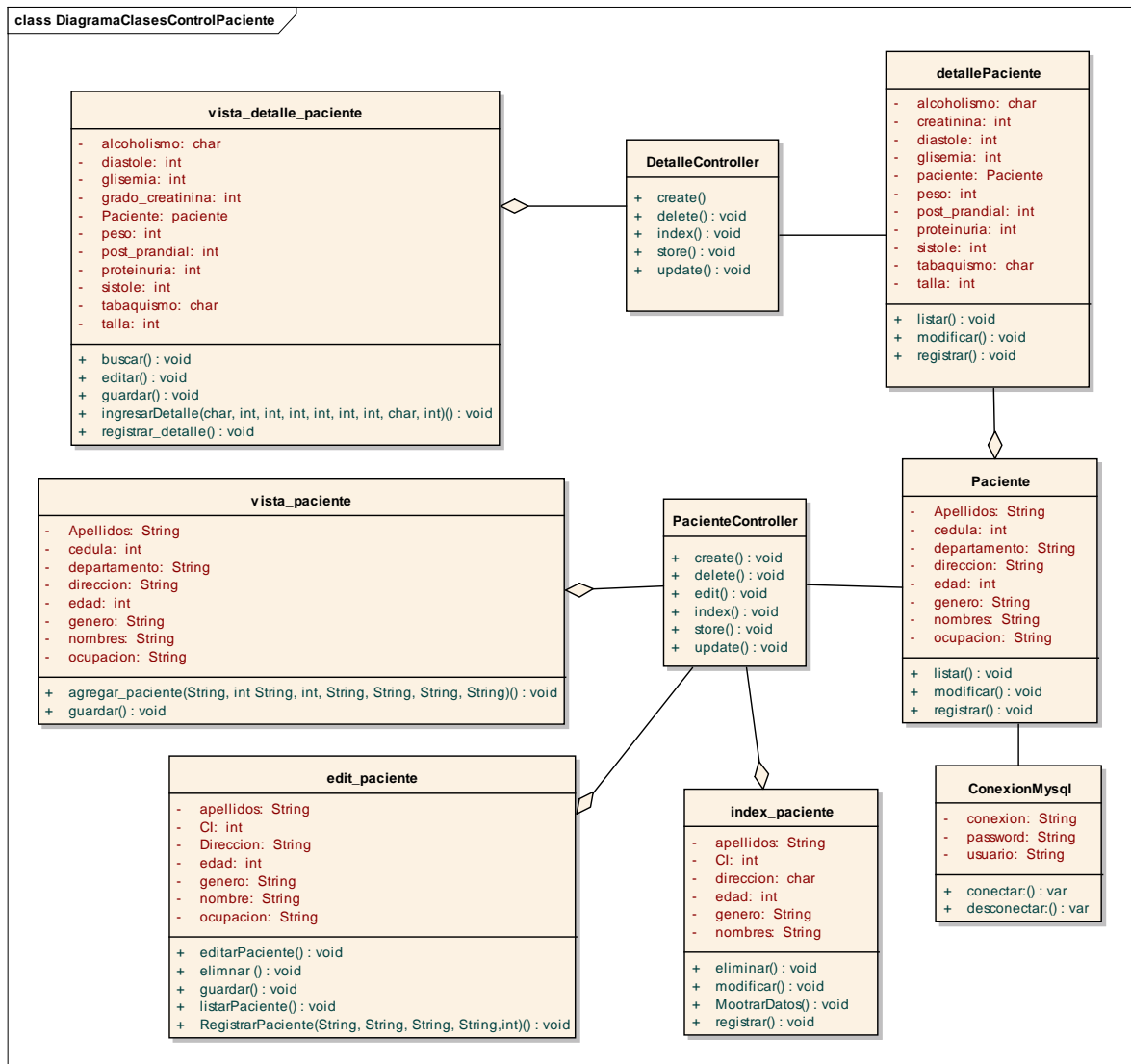


FIGURA 3.24: Diagrama de clases registrar paciente

FUENTE: Elaboración propia.

➤ **Diagrama de clases controlar centro de salud**

En la figura 3.25 se muestra el diagrama de clases centros de salud, donde el modelo centro de salud hace conexión con una clase controlador CentroController donde se hallan las funciones para modificar, eliminar, listar.

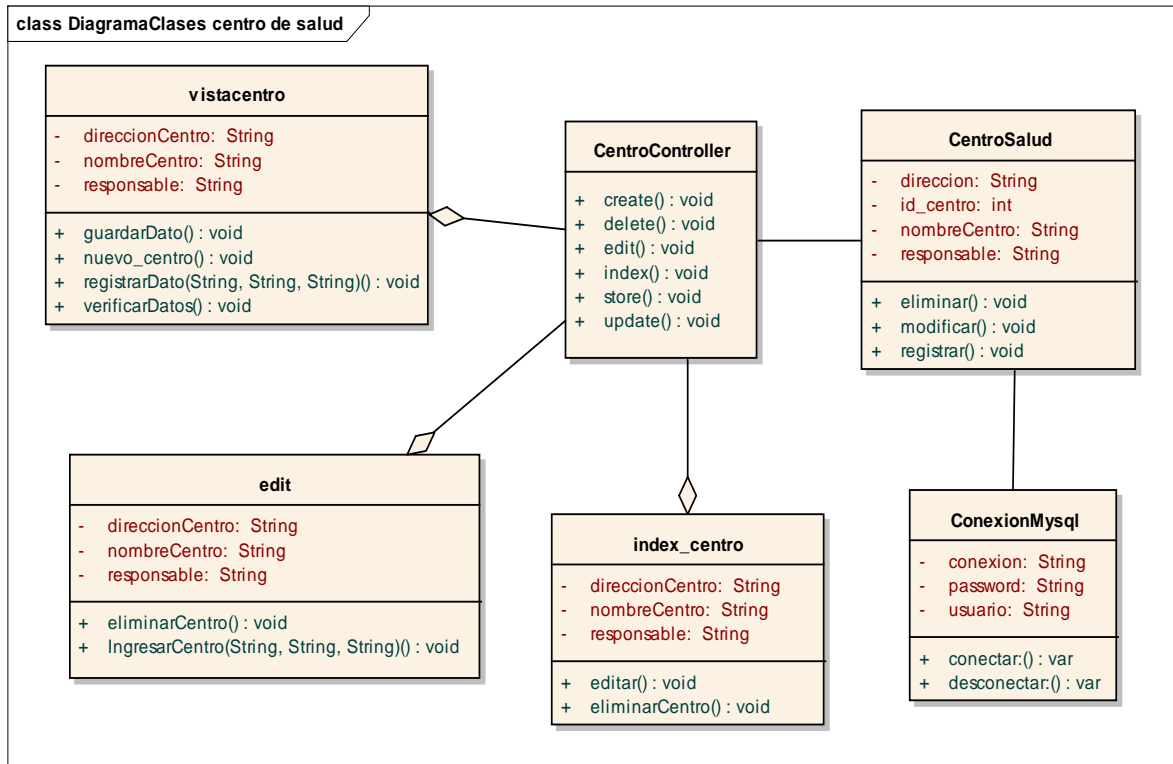


FIGURA 3.25: *Diagrama de clases centro de salud*
FUENTE: *Elaboración propia*

➤ Diagrama de clase registro glucosa

En la figura 3.26, diagrama de la clase glucosa se muestra las tres funcionalidades que se realiza dentro del módulo control enfermedades que son registrar, modificar y listar, cada una llama a la clase Controller realizando internamente la parte lógica del sistema pasa a llamar los métodos de la clase Glucosa, manteniendo la continua conexión a la base de datos.

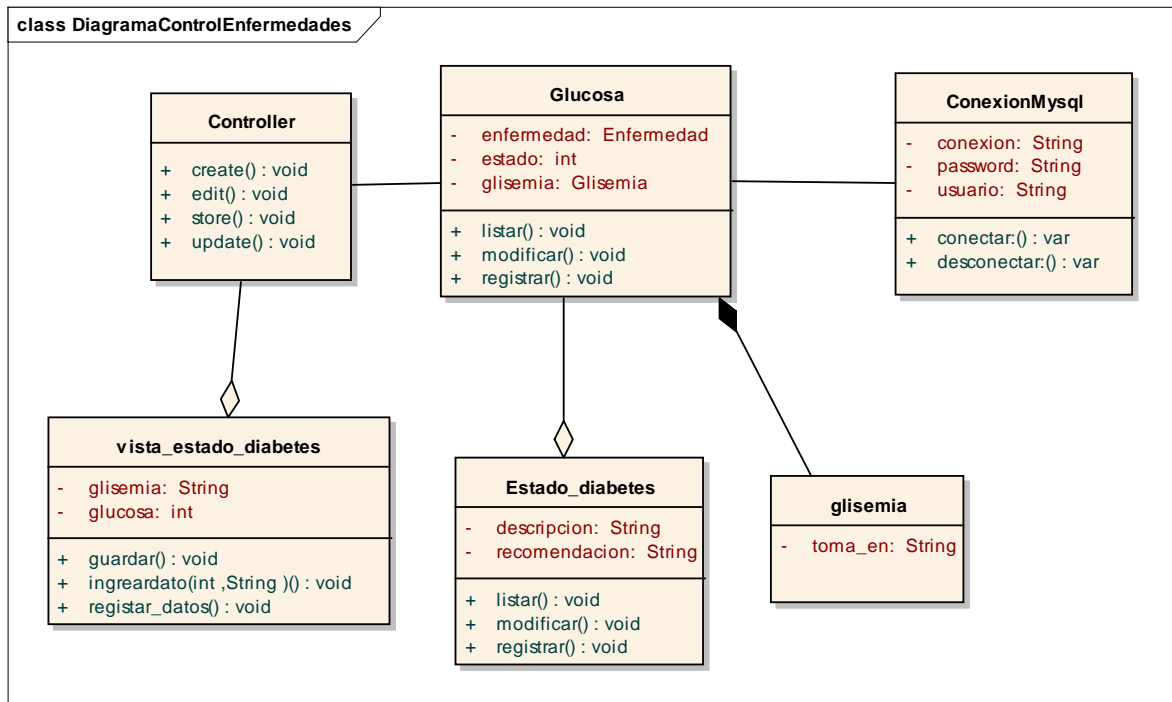


FIGURA 3.26: Diagrama de clases glucosa
FUENTE: Elaboración propia

➤ Diagrama de clase registro diástole

En la figura 3.27, diagrama de clases diastole se muestra las tres funcionalidades que se realiza dentro del módulo Control Enfermedad que son registrar, modificar y listar, cada una llama a la clase DiastoleController realizando internamente la parte lógica del sistema pasa a llamar los métodos de la clase diastole manteniendo la continua conexión a la Base de Datos.

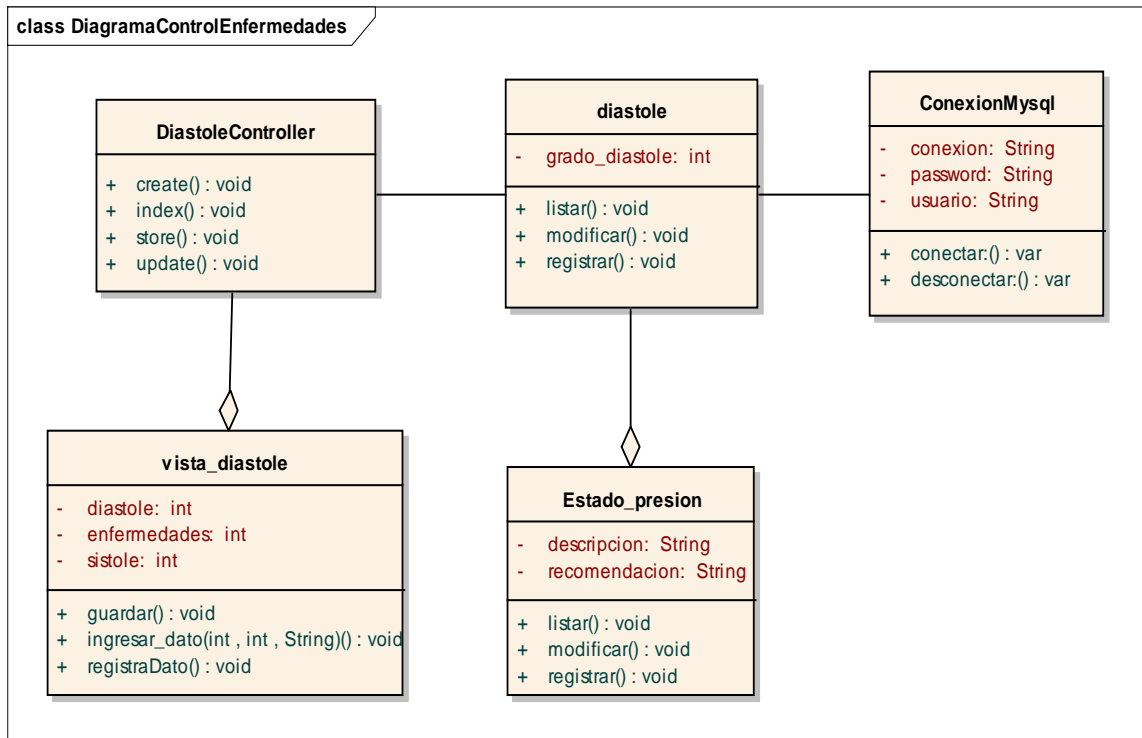


FIGURA 3.27: Diagrama de clases diástole
FUENTE: Elaboración propia

➤ **Diagrama de clases registro sístole**

En la figura 3.28, diagrama de clases sístole se muestra las tres funcionalidades que se realiza dentro del módulo Control Enfermedad que son registrar, modificar y listar, cada una llama a la clase SístoleController realizando internamente la parte lógica del sistema pasa a llamar los métodos de la clase Presión Arterial manteniendo la continua conexión a la base de datos.

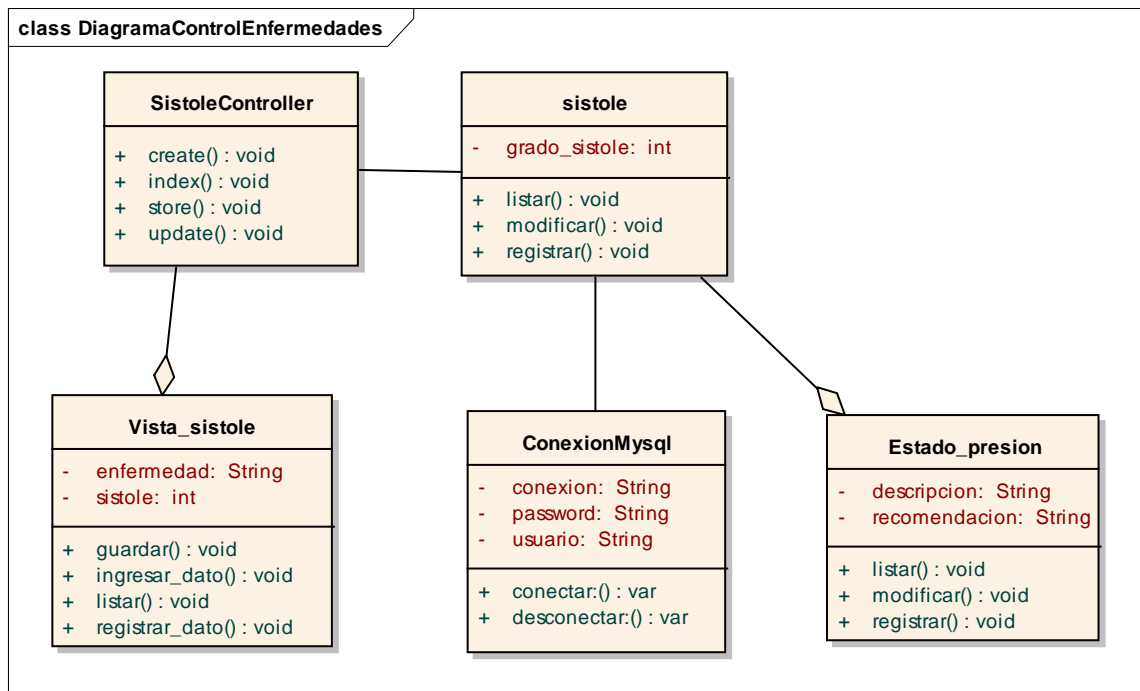


FIGURA 3.28 : *Diagrama de clases sístole*
FUENTE: *Elaboración propia*

➤ **Diagrama de clase registro diabetes**

En la figura 3.39, diagrama de clases diabetes se muestra las tres funcionalidades que se realiza dentro del módulo Control Enfermedad que son registrar, modificar y listar, cada una llama a la clase DiabetesController realizando internamente la parte lógica del sistema pasa a llamar los métodos de la clase Diabetes manteniendo la continua conexión a la base de datos.

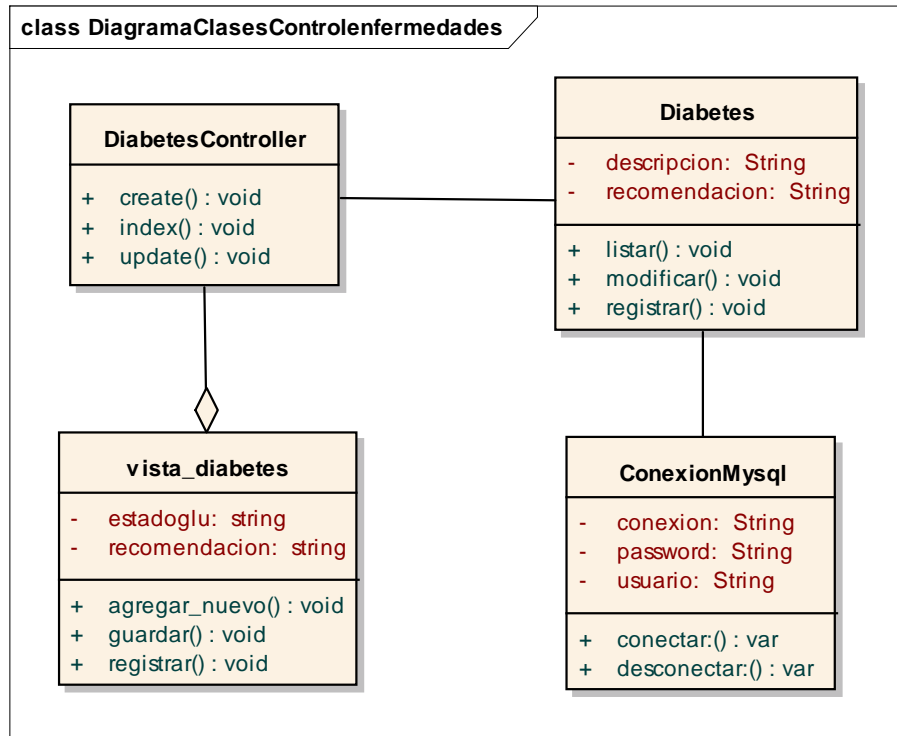


FIGURA 3.29: *Diagrama de clase diabetes*
FUENTE: *Elaboración propia*

➤ **Diagrama de clases registro presión arterial**

En la figura 3.30, diagrama de clases Presión Arterial se muestra las tres funcionalidades que se realiza dentro del módulo Control Enfermedad que son registrar, modificar y listar, cada una llama a la clase PresionController realizando internamente la parte lógica del sistema pasa a llamar los métodos de la clase Presión Arterial manteniendo la continua conexión a la Base de Datos.

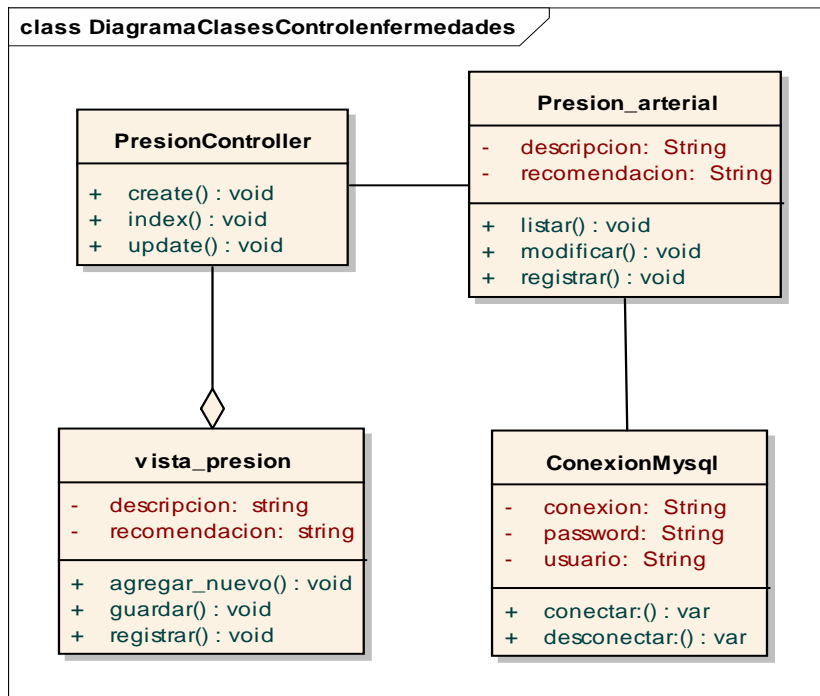


FIGURA 3.30: *Diagrama de clase presión arterial*
FUENTE: *Elaboración propia*

3.1.2.4. Diagrama de secuencia

➤ Diagrama de secuencia autenticar usuario

El usuario ingresa su nombre de usuario y su contraseña

El usuario presiona el botón “Ingresar”

El sistema valida usuario y clave

El sistema busca nivel de acceso del usuario.

El sistema muestra pantalla principal o de bienvenida y el menú en base al perfil.

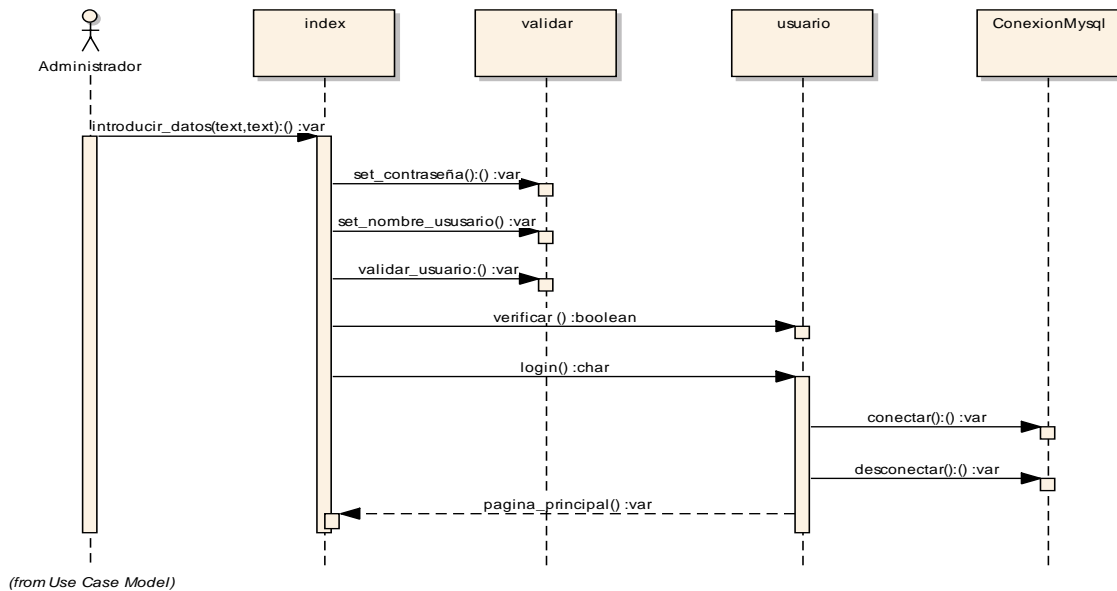


FIGURA 3.31: Diagrama de secuencia autenticar
FUENTE: Elaboración propia

➤ Diagrama de secuencia registrar usuario

El usuario presiona el botón registrar entonces el sistema muestra un formulario de registro.

El usuario registra los datos correspondientes, y luego presiona la opción guardar, entonces el sistema muestra en pantalla que se registró correctamente.

El usuario presiona el botón listar usuarios y el sistema muestra todos los datos correspondientes. El usuario presiona la opción “editar”, luego el sistema muestra el formulario del usuario que quiere modificar, luego el usuario presiona la opción “guardar”, y el sistema guarda y se muestra los datos corregidos.

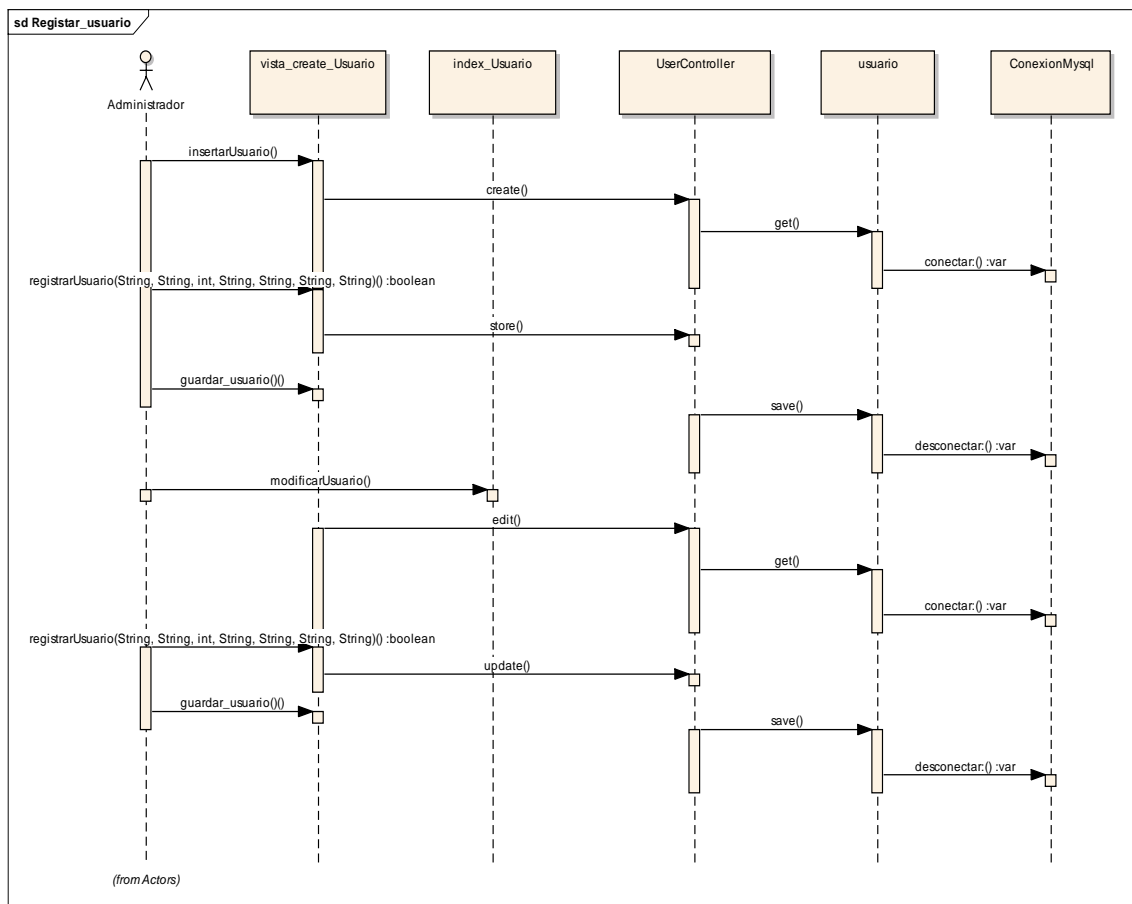


FIGURA 3.32: Diagrama de secuencia registrar usuario
FUENTE: Elaboración propia

➤ Diagrama de secuencia registrar paciente

El usuario presiona el botón registrar entonces el sistema muestra un formulario de registro.

El usuario registra los datos personales del paciente, y luego presiona la opción guardar, entonces el sistema muestra en pantalla que se registró correctamente.

El usuario presiona el botón listar pacientes y el sistema muestra todos los datos correspondientes.

El usuario presiona la opción “registrar detalle”, luego el sistema muestra el formulario de registro de detalle paciente, el usuario ingresa los datos al terminar presiona la opción “guardar”, el sistema guarda y muestra un mensaje que se guardó correctamente.

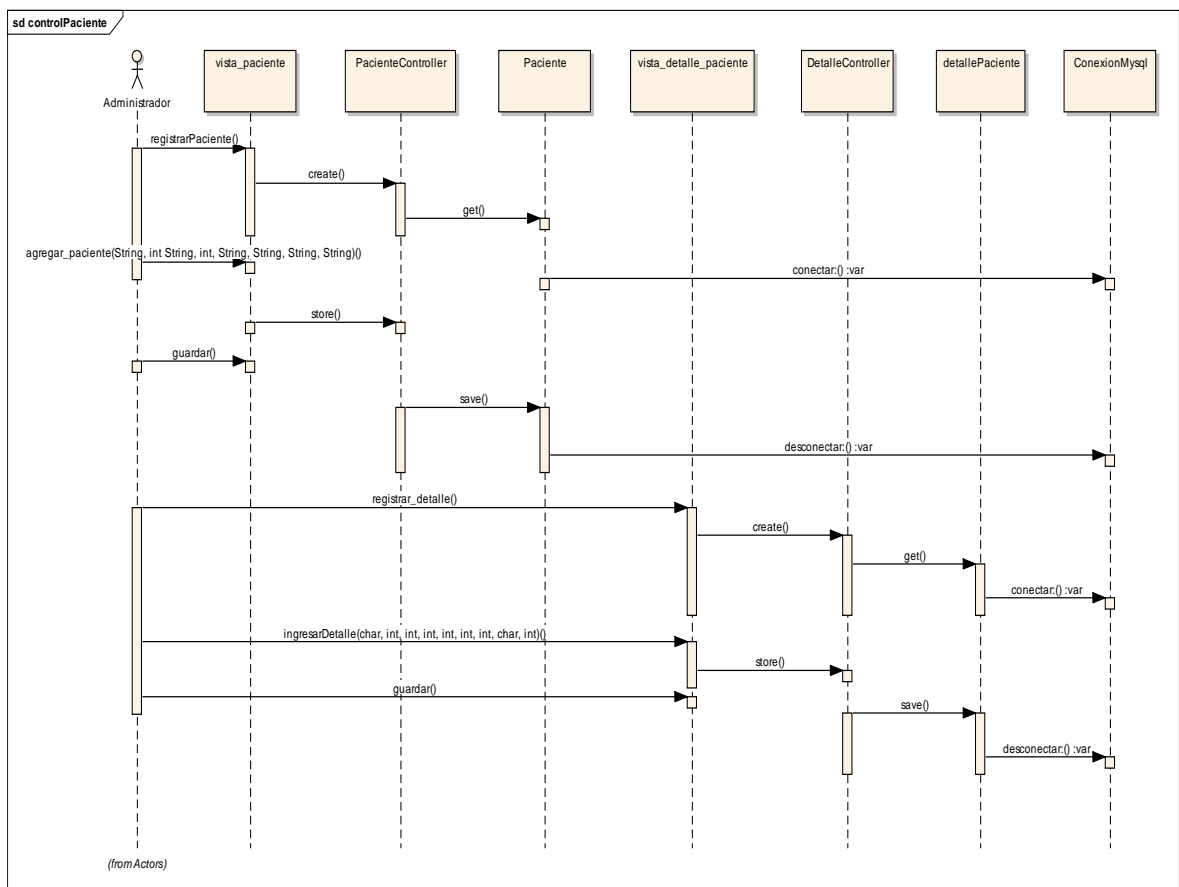


FIGURA 3.33: Diagrama de secuencia registro de paciente
FUENTE: Elaboración propia

➤ Diagrama de secuencia modificar paciente

El usuario presiona la opción “editar” entonces el sistema muestra un formulario para registro.

El usuario modifica los datos del paciente, y luego presiona la opción guardar, entonces el sistema muestra los registros con los datos modificados.

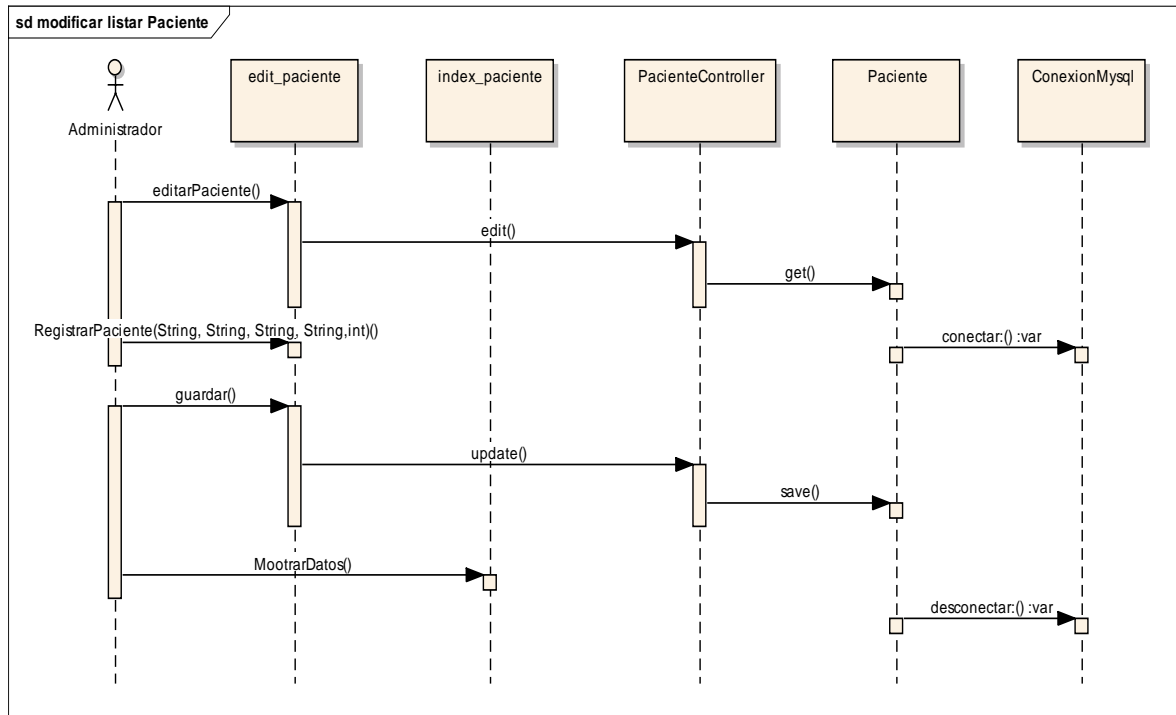


FIGURA 3.34: Diagrama de secuencia modificar paciente
FUENTE: Elaboración propia

➤ **Diagrama de secuencia registrar centro de salud**

El usuario presiona la opción “registrar “entonces el sistema muestra un formulario para registrar centro de salud. El usuario registra los datos de centro de salud, y luego presiona la opción guardar, y el sistema muestra en pantalla que se registró correctamente.

El usuario presiona la opción “editar centro”, entonces el sistema muestra el formulario para registro de centros de salud, el usuario ingresa los datos y al terminar presiona la opción “guardar”, el sistema guarda y muestra los registros modificados.

El usuario presiona el botón listar centros de salud y el sistema muestra todos los datos correspondientes a centros de salud.

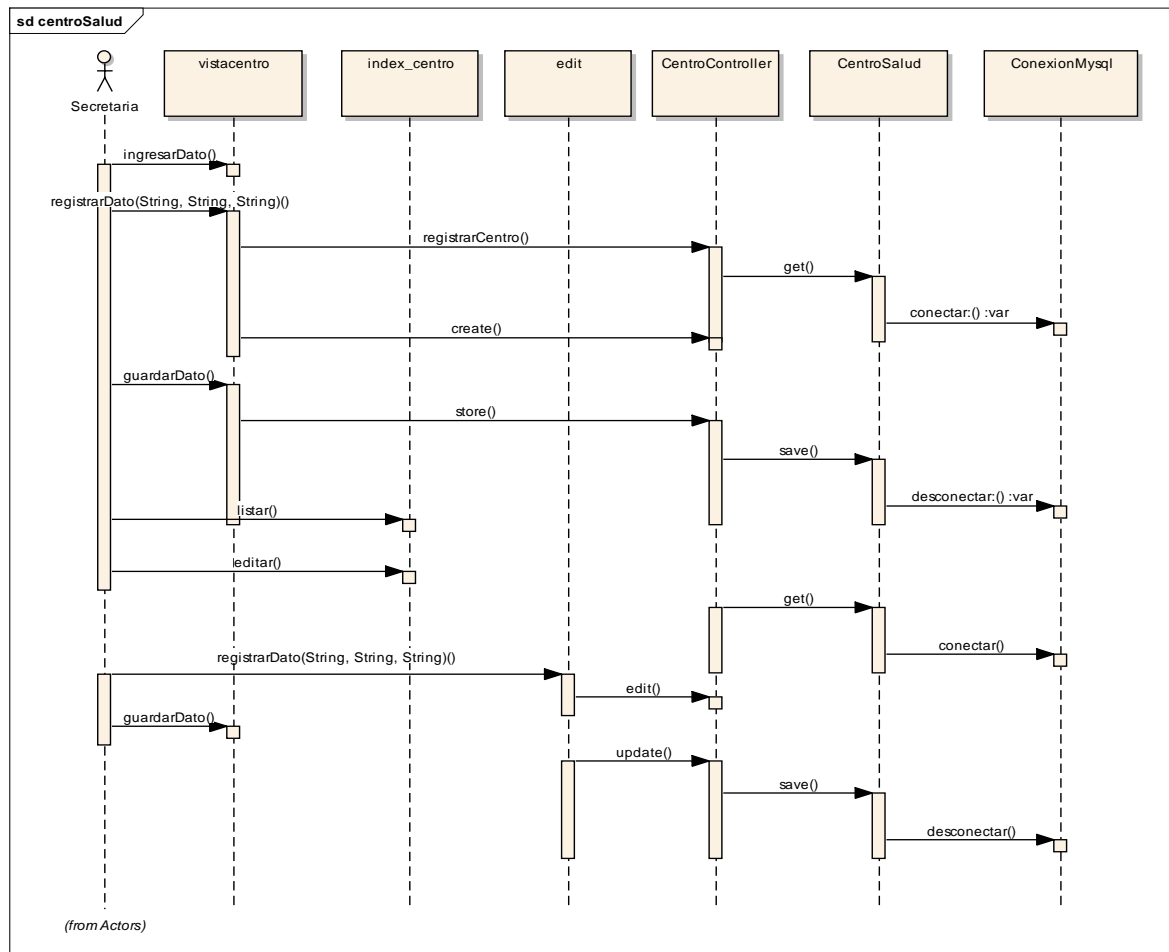


FIGURA 3.35: Diagrama de secuencia control de centro de salud
FUENTE: Elaboración propia

➤ Diagrama de secuencia registrar glucosa

El usuario presiona la opción “registrar” entonces el sistema muestra un formulario para registrar la glucosa. El usuario registra los datos correspondientes de glucosa, y luego presiona la opción guardar, y el sistema muestra en pantalla que se registró correctamente.

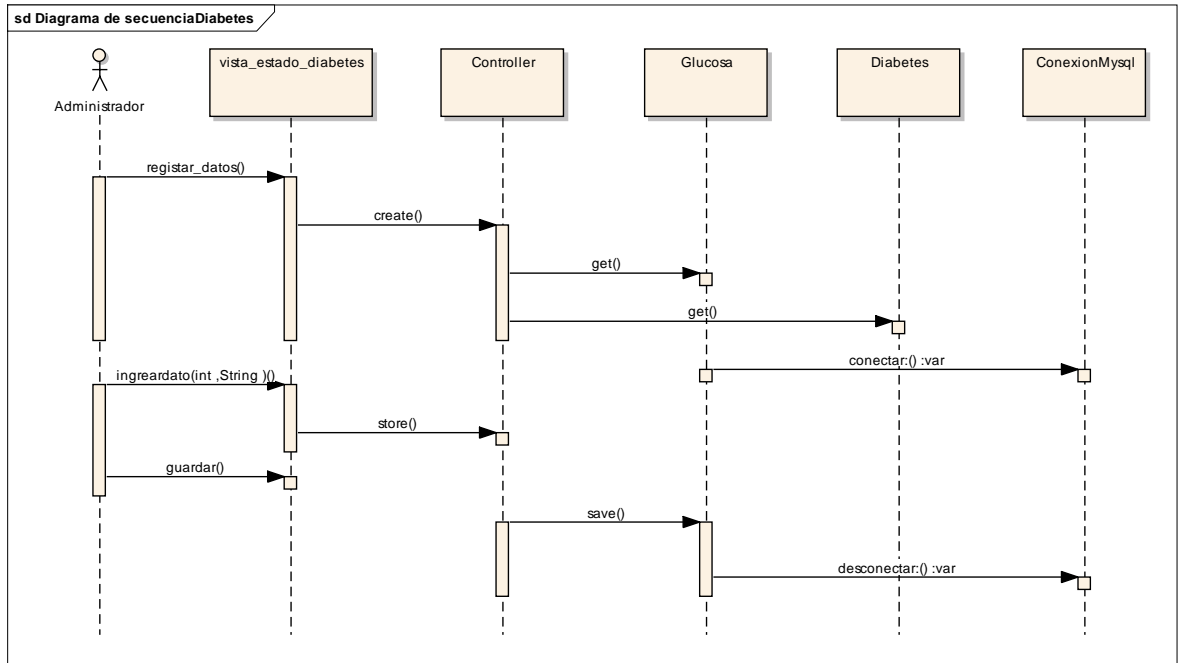


FIGURA 3.36: Diagrama de secuencia registrar glucosa
FUENTE: Elaboración propia

➤ **Diagrama de secuencia registrar presión arterial**

El usuario presiona la opción “registrar” entonces el sistema muestra un formulario para registrar la presión arterial. El usuario registra los datos de presión arterial, y luego presiona la opción guardar, y el sistema muestra en pantalla que se registró correctamente.

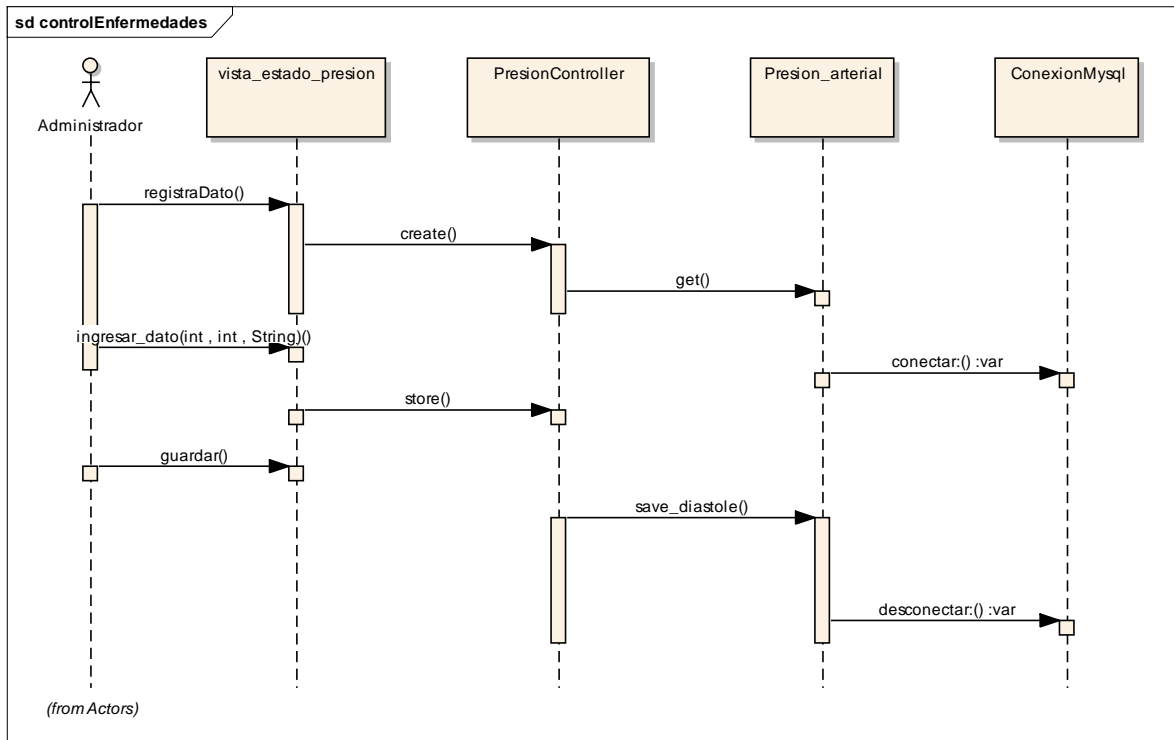


FIGURA 3.37: Diagrama de Secuencia de registrar presión arterial
FUENTE: Elaboración propia

➤ **Diagrama de secuencia registrar diabetes**

El usuario presiona la opción “registrar “entonces el sistema muestra un formulario para registrar la diabetes. El usuario registra los datos de diabetes, y luego presiona la opción guardar, y el sistema muestra en pantalla que se registró correctamente.

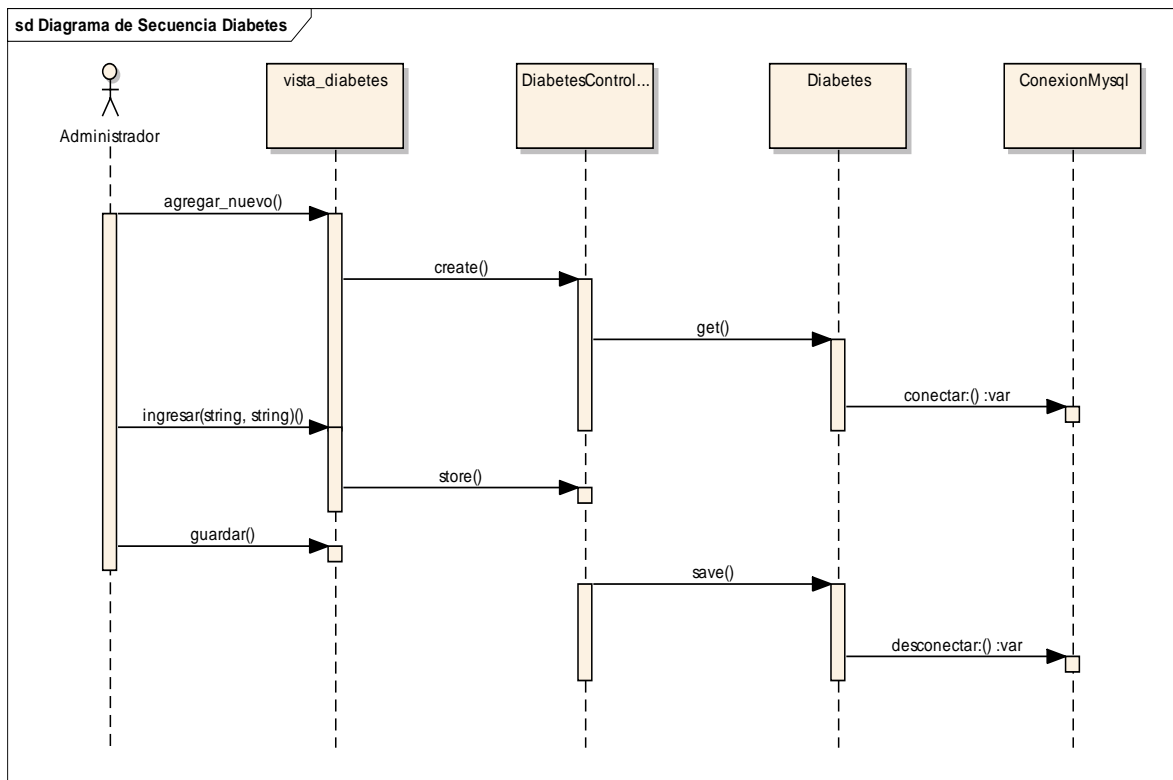


FIGURA 3.38: *Diagrama de secuencia registrar diabetes*

FUENTE: *Elaboración propia*

3.1.2.5. Diseño de la arquitectura

Se muestra la arquitectura del patrón MVC que utiliza para el desarrollo de proyecto.

Cuando se ingresa a una url directamente desde el navegador se hace mediante una petición http de tipo GET, esta solicitud se envía al archivo routes.php ubicado dentro de app/Http/routes.php, en caso de no existir esto dará un error, si la ruta existe, se llevará a un controlador en el cuál se encuentra la lógica, el controlador interactuará con un modelo opcionalmente para recuperar información de una base de datos. Esta información llega al controlador y desde el controlador se invoca una vista, las vistas se encuentran en el directorio resources/views, finalmente la vista se carga y se muestra en el navegador.

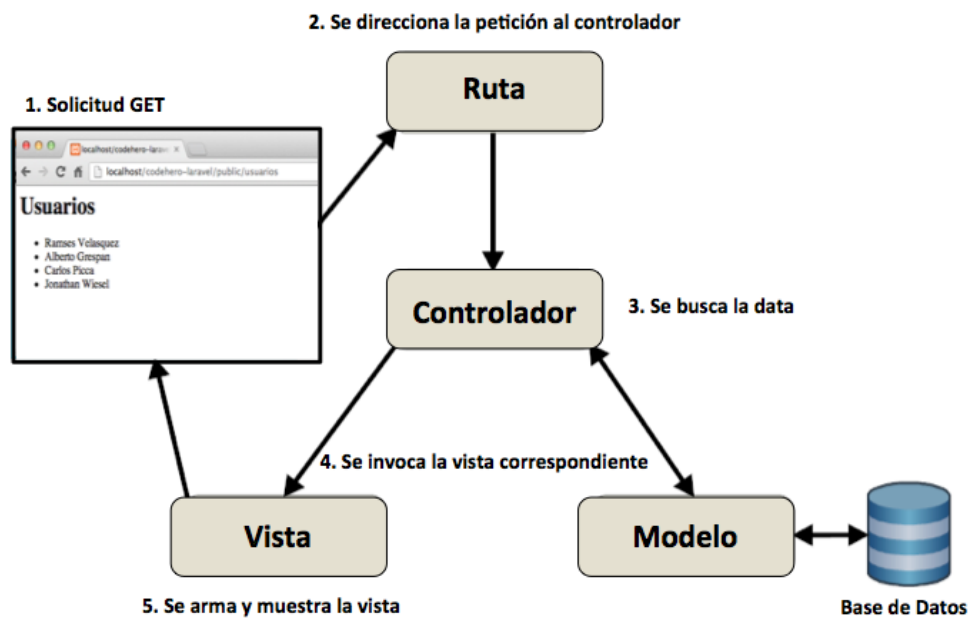


FIGURA 3.39: *Diseño de la arquitectura*
FUENTE: (Werner, 2015)

3.1.2.6. Elaboración del Modelo Lógico de la Base de Datos

El diseño lógico en el proceso de construir un esquema de la información que utiliza el sistema, basándose en un modelo de base de datos específico, independiente del SGBD que se vaya a utilizar.

En esta etapa se transforma el esquema conceptual en un esquema lógico que utilizara las estructuras de datos del modelo de base de datos en el que se basa el SGBD que se vaya a utilizar.

El diseño lógico es una fuente de información para el diseño físico. Además, desempeña un papel importante durante la etapa de mantenimiento del sistema, ya que permite que los futuros cambios que se realicen sobre los programas de aplicación o sobre los datos, se representen correctamente en la base de datos.

➤ **Modelo Lógico de la Base de Datos**

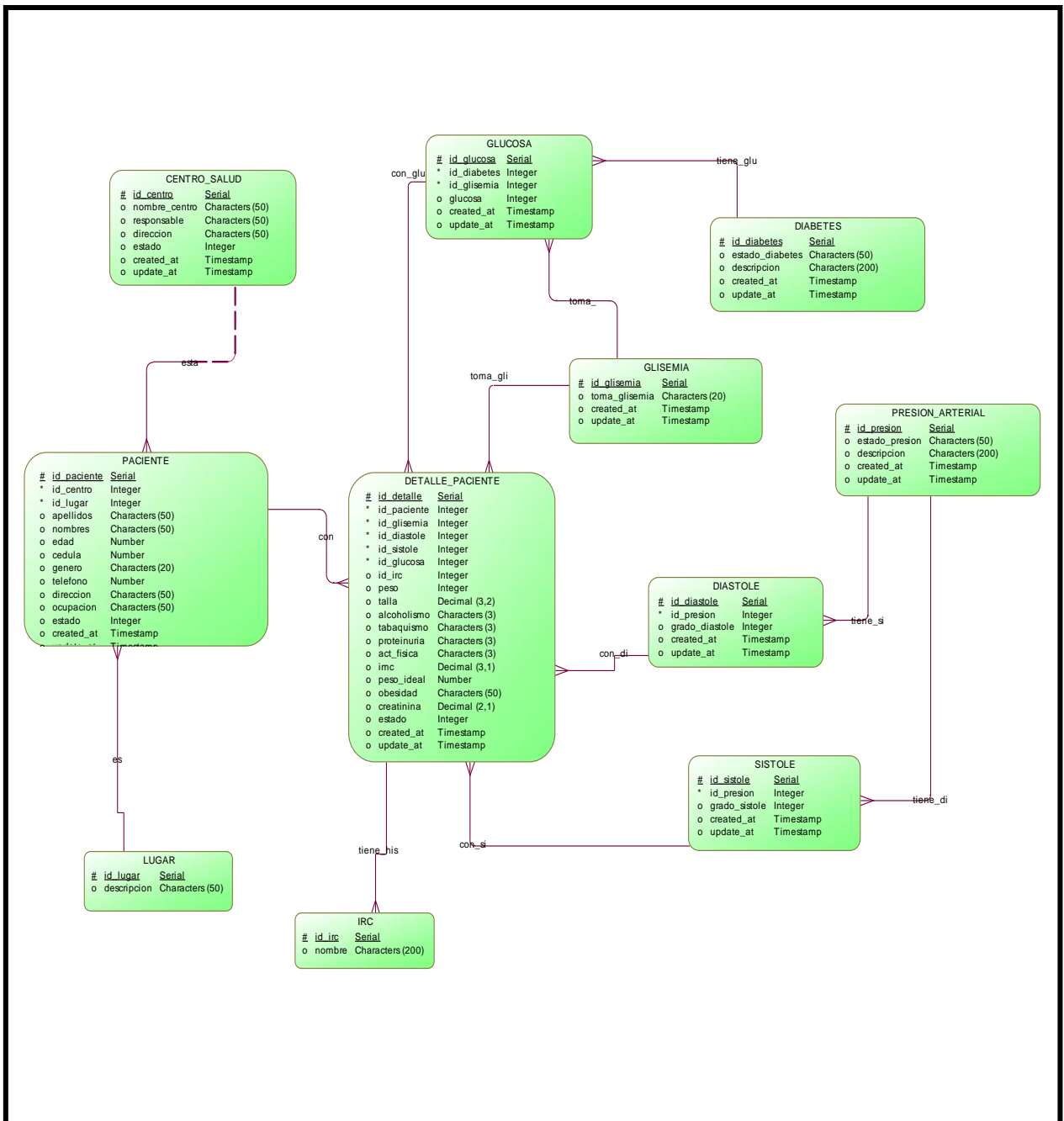


FIGURA 3.40: Modelo lógico de la base de datos

FUENTE: *Elaboración propia*

3.1.2.7. Diccionario de datos

La base de datos diseñada está compuesta de 11 tablas acorde a las necesidades en el sistema para la unidad de enfermedades no transmisibles del SEDES Pando.

A continuación, se describirá los atributos de cada tabla de la base de datos y su correspondiente observación de ser necesaria:

Tabla centro_salud

Centro_salud					
Atributo	Tipo	Key	Null	FK	Observación
Id_centro	increments	Pk	No		Identificador de la tabla
nombre_centro	string(50)		No		
Dirección	string(50)		No		
responsable	string(50)		No		
Estado	Integer		No		
created_at	Timestamp		No		Fecha de creación
update_at	timestamp		No		Fecha de última modificación

TABLA 3.11 : Tabla centro salud
FUENTE: Elaboración propia

Tabla lugar

Lugar					
Atributo	Tipo	Key	Null	FK	Observación
Id_lugar	increments	Pk	No		Identificador de la tabla
descripcion	string(50)		No		

TABLA 3.12 : Tabla lugar
FUENTE: Elaboración propia

Tabla paciente

Paciente					
Atributo	Tipo	Key	Null	FK	Observación
Id_paciente	Increments	Pk	No		Identificador de la tabla
id_centro	Integer			Fk	
id_lugar	Integer			Fk	
Apellidos	string(50)		No		
Nombres	string (50)		No		
edad	Number		No		
Cedula	Number		No		

Genero	string(20)		No		
Dirección	string(50)		No		
Teléfono	Number		No		
ocupacion	string (50)		No		
Estado	Integer		No		
created_at	timestamp		No		Fecha de creación
update_at	timestamp		No		Fecha de última modificación

TABLA 3.13 : *Tabla paciente*
FUENTE: *Elaboración propia*

Tabla presión_arterial

Presion_arterial					
Atributo	Tipo	Key	Null	FK	Observación
Id_presion	Increments	Pk	No		Identificador de la tabla
estado_presion	string(50)		No		
descripción	String(200)		No		
created_at	Timestamp		No		Fecha de creación
update_at	Timestamp		No		Fecha de última modificación

TABLA 3.14 *Tabla presión arterial*
FUENTE: *Elaboración propia*

Tabla diabetes

Diabetes					
Atributo	Tipo	Key	Null	FK	Observación
Id_diabetes	Increments	Pk	No		Identificador de la tabla
estado_diabetes	String(50)		No		
descripción	String(200)		No		
created_at	timestamp		No		Fecha de creación
update_at	timestamp		No		Fecha de última modificación

TABLA 3.15 : *Tabla diabetes*
FUENTE: *Elaboración propia*

Tabla diastole

Diástole					
Atributo	Tipo	Key	Null	FK	Observación
Id_diastole	Increments	Pk	No		Identificador de la tabla
Id_presion	integer		No	Fk	
grado_diastole	integer		No		
created_at	timestamp		No		Fecha de creación
update_at	timestamp		No		Fecha de última modificación

TABLA 3.16 : Tabla diástole
FUENTE: Elaboración propia

Tabla sistole

Sístole					
Atributo	Tipo	Key	Null	FK	Observación
Id_sistole	increments	Pk	No		Identificador de la tabla
Id_presion	integer		No	Fk	
grado_sistole	integer		No		
created_at	timestamp		No		Fecha de creación
update_at	timestamp		No		Fecha de última modificación

TABLA 3.17 : Tabla sistole
FUENTE: Elaboración propia

Tabla glisemia

Glisemia					
Atributo	Tipo	Key	Null	FK	Observación
Id_glisemia	increments	Pk	No		Identificador de la tabla
Toma_glisemia	String(20)		No		
created_at	timestamp		No		Fecha de creación
update_at	timestamp		No		Fecha de última modificación

TABLA 3.18 : Tabla glicemia
FUENTE: Elaboración propia

Tabla glucosa

Glucosa					
Atributo	Tipo	Key	Null	FK	Observación
Id_glucosa	increments	Pk	No		Identificador de la tabla
Id_diabetes	Integer		No	Fk	
Id_glisemia	Integer		No	Fk	
Glucosa	Integer		No		
created_at	timestamp		No		Fecha de creación
update_at	timestamp		No		Fecha de última modificación

TABLA 3.19 : Tabla glucosa
FUENTE: Elaboración propia

Tabla detalle_paciente

Detalle_paciente					
Atributo	Tipo	Key	Null	FK	Observación
Id_detalle	increments	Pk	No		Identificador de la tabla
Id_Sístole	Integer		No	Fk	
Id_Diástole	Integer		No	Fk	
Id_Glucosa	Integer		No	Fk	
Id_Paciente	Integer		No	Fk	
Id_Glisemia	Integer		No	Fk	
Id_irc	Integer		No		
Peso	Integer		No		
talla	Decimal(3 ,2)		No		
peso_ideal	Number		No		
imc	Decimal(3, 1)		No		
obesidad	String(50)		No		
creatinina	Decimal(2, 1)		No		
alcoholismo	String(3)		No		
tabaquismo	String(3)		No		
proteinuria	String(3)		No		
act_fisica	String(4)		No		
estado_detalle	Integer		No		
created_at	timestamp		No		Fecha de creación
update_at	timestamp		No		Fecha de última modificación

TABLA 3.20 : Tabla detalle paciente
FUENTE: Elaboración propia

Tabla irc

Irc					
Atributo	Tipo	Key	Null	FK	Observación
Id_irc	increments	Pk	No		Identificador de la tabla
nombre	String(200)		No		

TABLA 3.21 : Tabla irc
FUENTE: Elaboración propia

Una vez terminado con los requerimientos, el análisis, diseño del sistema y concluido el diseño de la base de datos, queda realizar la construcción de los módulos utilizando las herramientas mencionadas en el anterior capítulo. A continuación, se presentan las interfaces del sistema ya desarrollado según a los requerimientos del usuario

3.1.3. Fase de construcción

En esta fase es donde pasamos a moldear el producto que vamos a mostrar al usuario con sus requerimientos, además de que en esta etapa se realiza la implantación y las pruebas al mismo.

3.1.3.1. Interfaz del sistema

La interfaz de usuario es el mecanismo a través del cual se establece un dialogo entre el sistema y el usuario, la comunicación puede ser físico mediante el teclado o símbolo por medio de iconos.

La siguiente imagen muestra la interfaz del sistema informático para el control de acceso al sistema por autenticación de usuarios donde cada usuario puede acceder al sistema, debe ingresar su nombre de usuario y su contraseña para acceder al sistema.

➤ **Control Acceso al Sistema**



*FIGURA 3.42: Interfaz autenticación usuario
FUENTE: Elaboración propia*

➤ **Interfaz de Usuario**

En la figura 3.43 se presenta la pantalla principal del Sistema, en la cual se podrá acceder a los diferentes módulos para realizar las funciones asignadas en el menú horizontal



*FIGURA 3.43: Interfaz de usuario
FUENTE: Elaboración propia*

➤ Registro de paciente

En la figura 3.44 se presenta la pantalla para el formulario en la cual se podrá registrar los datos personales de los pacientes.



*FIGURA 3.44: Registro paciente
FUENTE: Elaboración propia*

➤ Registro de centro de salud

En la figura 3.45 se presenta la pantalla para el formulario en la cual se podrá registrar los Centros de Salud.



*FIGURA 3.45: Formulario de registro centro de salud
FUENTE: Elaboración propia*

➤ **Registro de detalle del paciente**

En la figura 3.46 se presenta la pantalla para el formulario en la cual se podrá registrar los datos del detalle paciente, pacientes que previamente han sido registrados con sus datos personales.

Registro Detalle Paciente

Buscar Paciente :

Tabaquismo: Alcoholicismo: Actividad Física:

Estado del peso y la Estatura

Peso en kg: Estatura:

Obesidad:

Grado IMC:

Peso Ideal:

Ingresar la Proteinuria

Proteinuria:

Ingresando los Margenes de la glucosa

Toma Glicemia: Glucosa:

Estado de la Presion Arterial

Sistole: Diastole:

Estado de la Creatinina

Creatinina:

*FIGURA 3.46: Formulario de registro detalle paciente
FUENTE: Elaboración propia.*

✓ Presión arterial

En la figura 3.47 muestra el listado del estado de la presión arterial donde se establece los tipos de enfermedades que pueda padecer el paciente, con la opción de poder editar a preferencia del usuario, en esta interfaz se tiene también la opción para volver al inicio de la aplicación, y el panel vertical para más configuraciones.

The screenshot shows a web application interface. At the top, there is a blue header with a home icon, a menu icon, the text 'Inicio', and the user name 'usuario: mary'. Below the header is a dark vertical sidebar with icons for home, user, heart, and list. The main content area is a window titled 'Configuración' with a sub-header 'Presion Arterial'. Inside the window, there is a blue button labeled 'Nuevo Registro'. Below this is a table with the following data:

#	TIPO DE ENFERMEDAD	RECOMENDACION	
1	PRESION ARTERIAL NORMAL	EN PRESION ARTERIAL APARENTEMENTE SANO	Editar
2	PRESION BAJA	Lleva una dieta alta en proteínas, la cuál debe incluir riñones, hígado, carnes de órganos, patatas horneadas con cáscara, verduras de hojas verdes, semillas de cereales	Editar
3	PREHIPERTENCION	alerta temprana para prevenir la hipertensión	Editar
4	PRESION ALTA	Reduzca el consumo de sal, evite la ingestión de bebidas alcohólicas, no fume, trate de reducir el peso y mantenga su peso normal.	Editar

FIGURA 3.47: Lista del estado de la presión arterial
FUENTE: Elaboración propia

✓ Estado de la diabetes

En la figura 3.48 muestra el listado del estado de diabetes donde se establece los tipos de enfermedades que pueda padecer el paciente, con la opción de poder editar a preferencia del usuario, en esta interfaz se tiene también la opción para volver al inicio de la aplicación, y el panel vertical para más configuraciones.



FIGURA 3.48: *Lista del estado de la diabetes*
FUENTE: *Elaboración propia*

✓ **Resultado del diagnóstico de los pacientes**

En la figura 3.49 muestra el listado del diagnóstico general de los pacientes con la opción de editar lo datos requeridos.

#	PACIENTE	EDAD	GENERO	TALLA	IMC	PESO	PESO IDEAL	ESTADO PESO	ESTADO DIABETES	HIPERTENCION	R IRC	OPERACIONES
1	SOFIA GUZMAN	89	F	1.67	16.1	45	66.00	Falta de peso.	DIABETES	PRESION ALTA	INSUFICIENCIA RENAL AVANZADA PRETERMINAL	
2	RUDY PAREDES	25	M	1.67	24.7	69	66.00	Sin Obesidad.	NO PADECE DE DIABETES	PRESION ARTERIAL NORMAL	DAÑO RENAL LEVE	
3	FREDDY MOEALES	35	M	1.65	25.7	70	64.00	Sobrepeso. Debe adelgazar	PREDIABETES	PRESION ALTA	DAÑO RENAL LEVE	
4	RUBEN LOZA	23	M	1.56	26.7	65	55.00	Sobrepeso. Debe adelgazar	NO PADECE DE DIABETES	PRESION ALTA	DAÑO RENAL LEVE	
5	MARIANA MORALES	45	F	1.67	20.1	56	66.00	Sin Obesidad.	NO PADECE DE DIABETES	PRESION BAJA	INSUFICIENCIA RENAL MODERADA	

FIGURA 3.49: *Resultado del diagnóstico de los pacientes*
FUENTE: *Elaboración propia*

3.1.4. Fase de transición

En esta fase inicia la etapa de implementación del sistema dentro de la unidad de Enfermedades no Transmisibles del SEDES PANDO. Para esta fase se realiza la utilización de los diagramas de despliegue y el diagrama de componentes.

3.1.4.1. Diagrama de despliegue

Este es un diagrama estructurado que muestra la arquitectura del sistema desde el punto de vista de la distribución de los artefactos del software en los destinos de despliegue.

El diagrama muestra la configuración de nodos de procesamiento en tiempo de ejecución y las instancias de componentes y objetos que se encuentran dentro de esos nodos. Los componentes representan manifestaciones de ejecución de unidades de código.

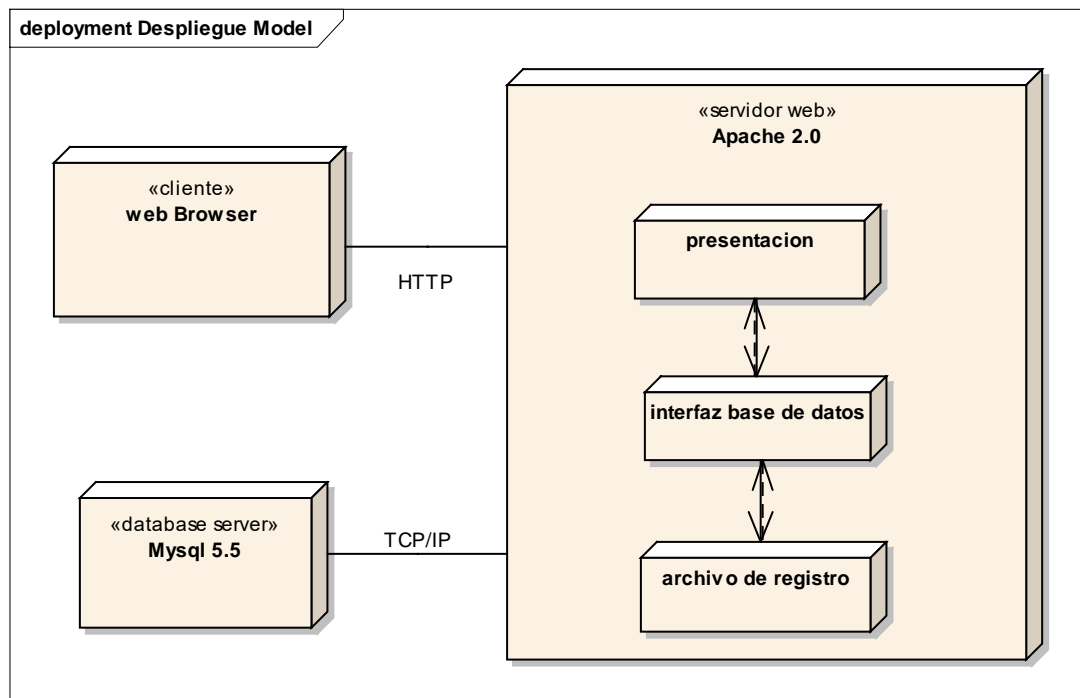


FIGURA 3.50: Diagrama de despliegue
FUENTE: Elaboración propia

3.1.4.2. Diagrama de componentes

Con este tipo de diagramas se representan entidades reales, siendo los, componentes de software. Un componente de software puede ser una tabla, un archivo de datos, un ejecutable.

En la figura 3.51 se muestra el diagrama de componentes donde se puede identificar las interfaces generales, en este diagrama el dominio del diagnóstico depende de los datos de los pacientes para verificar su estado, el componente enfermedad representa a los tipos de enfermedades del sistema.

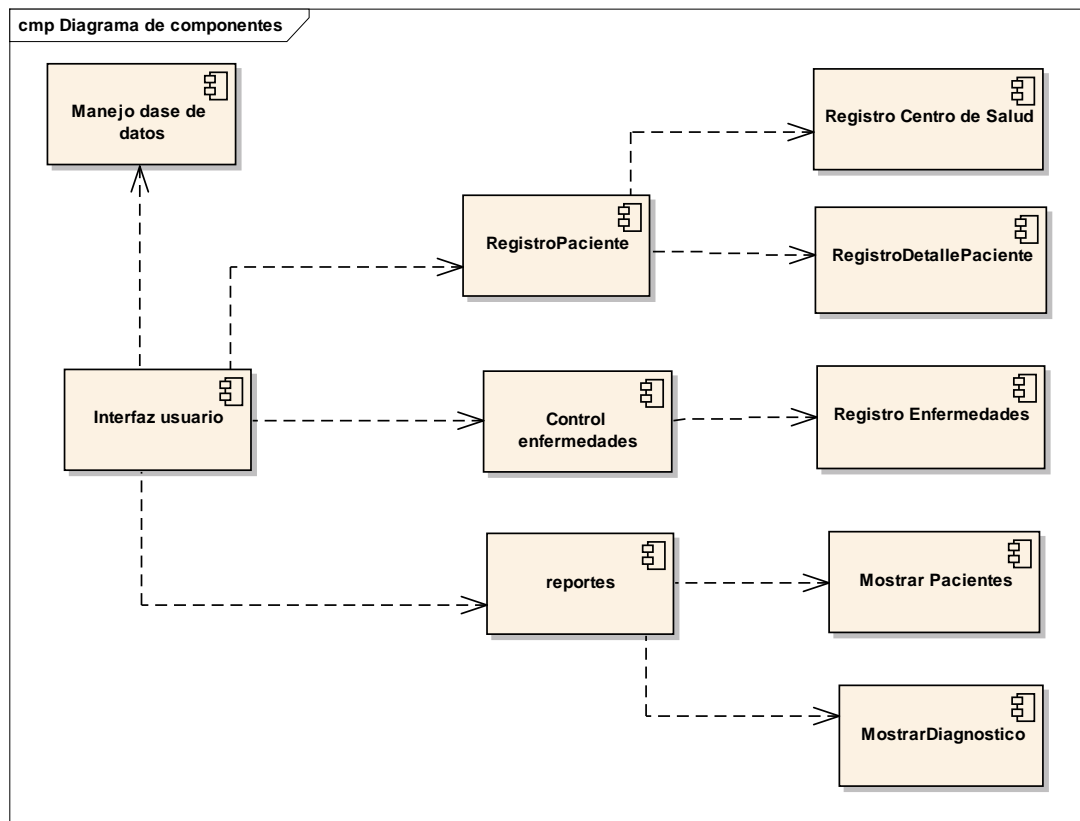


FIGURA 3.51: Diagrama de componentes
FUENTE: Elaboración propia

3.1.4.3. Pruebas de Integración

Prueba de integración Autenticar			
Datos de Entrada: nombre de usuario, contraseña			
Entrada	Condición de entrada	Salida esperada	Condiciones de salida
root , 123456	Nombre de usuario password	Mensaje de bienvenida	satisfactorio
---, ---	Nombre de usuario password	El campo de usuario es requerido El campo de password es requerido	
root , 1234578	Nombre de usuario password	Mensaje de error Estas credenciales no coinciden con nuestros registros	

TABLA 3.22: *Prueba de integración autenticar usuario*
FUENTE: Elaboración Propia

Prueba de integración Registrar usuario			
Datos de Entrada: nombre, apellido, cedula, telefono, email, password, usuario			
Entrada	Condición de entrada	Salida esperada	Condición de salida
Mary Sarhuri 6785954 root sarhuri51@gmail.com 1234567	nombre:si apellido:si cedula:si usuario:si email:si password:si	Mensaje de datos se registró correctamente	satisfactorio
---	nombre:no apellido:no cedula:no usuario:no email:no password:no	El campo nombre es requerido El campo apellido es requerido El campo cedula es requerido El campo usuario es requerido El campo email es requerido El campo password es requerido	
Mary Sarhuri 67859 root sarhuri51 1234567	nombre:si apellido:si cedula:si usuario:si email:si password:si	El campo cedula debe tener mínimo 7 caracteres Introduzca un correo electrónico	

<p>9-’/* 9-’/* 6785954 root sarzuri51@gmail.com 1234567</p>	<p>nombre:no apellido:no cedula:si usuario:si email:si password:no</p>	<p>El campo nombre debe ser de tipo alfabético El campo apellido debe ser de tipo alfabético El campo cedula ya fue registrado</p>	
--	---	--	--

TABLA 3.23: Prueba de integración registrar usuario
FUENTE: Elaboración propia

Prueba de integración modificar usuario		
Datos de Entrada: nombre, apellido, cedula, usuario, email, password,		
Entrada	Condición de entrada	Salida esperada
<p>Ana Lopez 6785954 root sarzuri51@gmail.com 1234567</p>	<p>nombre:si apellido:si cedula:si usuario:si email:si password:si</p>	<p>Se registró correctamente</p>
<p>--- --- --- --- --- ---</p>	<p>nombre:no apellido:no cedula:no usuario:no email:no password:no</p>	<p>El campo nombre es requerido El campo apellido es requerido El campo cedula es requerido El campo usuario es requerido El campo email es requerido El campo password es requerido</p>
<p>Mary Sarzuri 67859 root sarzuri51 1234567</p>	<p>nombre:si apellido:si cedula:si usuario:si email:si password:si</p>	<p>El campo cedula debe tener mínimo 7 caracteres En el campo email debe ingresar un correo electrónico</p>
<p>9-’/* 9-’/* 6785fff root sarzuri51@gmail.com 1234567</p>	<p>nombre:no apellido:no cedula:si usuario:si email:si password:no</p>	<p>El campo nombre debe ser de tipo alfabético El campo apellido debe ser de tipo alfabético El campo cedula debe ser de tipo numérico</p>

TABLA 3.24: Prueba de integración registrar usuario
FUENTE: Elaboración propia

Prueba de integración Registrar Paciente			
Datos de Entrada: apellidos, genero, edad, nombres cedula dirección teléfono, ocupación, lugar_id			
Entrada	Condición de entrada	Salida esperada	Condiciones de salida
Gutiérrez Mabel Femenino 34 7845136 Barrio Mapajo 78451236 Enfermera Pando Santa Clara	apellidos : si nombres : si genero : si edad : si cedula : si dirección : si teléfono : si ocupación : si lugar_id : si centro_id : si	Mensaje “Se ha registrado correctamente”	satisfactorio
--- --- Femenino --- --- --- --- 78451236 --- --- ---	apellidos: no nombres : no genero : si edad : no cedula : no dirección: no teléfono : si ocupación : no lugar_id : no centro_id : no	Mensaje de error El campo apellido es requerido El campo nombre es requerido El campo edad es requerido El campo cedula es requerido El campo dirección es requerido El campo ocupación es requerido El campo departamento es requerido El campo centro de salud es requerido	
-9/* ,09/* Femenino fr 784 Barrio Mapajo 78451230 Enfermera Pando Santa Clara	apellidos : si nombres : si genero : si edad : si cedula : si dirección : si teléfono : si ocupación : si lugar_id : si centro_id : si	El campo apellido debe ser de tipo alfabético El campo nombre debe ser de tipo alfabético El campo edad debe ser de tipo numérico El campo cedula debe contener mínimo siete caracteres	

TABLA 3.25: Prueba de integración registrar paciente

FUENTE: *Elaboración propia*

Prueba de integración Registrar Detalle Paciente			
Datos de Entrada: peso, talla, peso_ideal, imc, alcoholismo, tabaquismo, proteinuria, act_fisica, obesidad, creatinina, sistole_id, diastole_id, glucosa_id, pacientes_id, glicemia_id			
Entrada	Condición de entrada	Salida esperada	Condiciones de salida
Mariana morales 65 1.67 66 23.31 Sin Obesidad No No Positivo No 1.7 80 90 75 Ayunas	pacientes_id:si peso:si talla:si peso_ideal:si imc:si obesidad:si alcoholismo:si tabaquismo:si proteinuria:si act_fisica:si creatinina:si sistole_id:si diastole_id:si glucosa_id:si glisemia_id:si	Mensaje “Se ha registrado correctamente”	satisfactorio
Mariana morales --- --- --- --- --- --- --- --- --- --- --- --- --- --- ---	pacientes_id:si peso:no talla:no peso_ideal:no imc:no obesidad:no alcoholismo:no tabaquismo:no proteinuria:no act_fisica:no creatinina:no sistole_id:no diastole_id:no glucosa_id:no glisemia_id:no	El campo peso es requerido El campo talla es requerido El campo peso ideal es requerido El campo imc es requerido El campo obesidad es requerido El campo alcoholismo es requerido El campo tabaquismo es requerido El campo proteinuria es requerido El campo actividad física es requerido El campo creatinina es requerido El campo sístole es requerido El campo diástole es requerido El campo glucosa es requerido El campo glicemia es requerido	

Mariana morales nn fr 23.31 Sin Obesidad No No Positivo No 1.7 80 90 75 Ayunas	pacientes_id:si peso:no talla:no peso_ideal:si imc:si obesidad:si alcoholismo:si tabaquismo:si proteinuria:si act_fisica:si creatinina:si sistole_id:si diastole_id:si glucosa_id:si glisemia_id:si	El campo peso debe ser de tipo numérico El campo talla debe ser de tipo decimal	
---	---	--	--

TABLA 3.26: Prueba de integración registrar detalle paciente
FUENTE: Elaboración propia

3.1.4.4. Pruebas Unitarias

Prueba Unitaria de Autenticar	
Entrada	Salida
Nom_usuario , cont_usuario	1
Nom_usuario , null	0

TABLA 3.27: Prueba unitaria autenticar
FUENTE: Elaboración propia

Prueba unitaria de registro paciente	
Entrada	Salida
apellidos, nombres, genero, edad, cedula, dirección, teléfono, ocupación, lugar_id, centro_id	1
Null, nombres, genero, edad, cedula, dirección, teléfono, ocupación, lugar_id, centro_id	0
apellidos, null, genero, edad, cedula, dirección, teléfono, ocupación, lugar_id, centro_id	0
apellidos, nombres, null, cedula, edad, dirección, teléfono, ocupación, lugar_id, centro_id	0
apellidos, nombres, genero, edad, null dirección, teléfono, ocupación, lugar_id, centro_id	0

apellidos, nombres, genero, cedula, edad, null, teléfono, ocupación, lugar_id, centro_id	0
apellidos, nombres, genero, edad, null dirección, null, ocupación, lugar_id, centro_id	0
apellidos, nombres, genero, edad, cedula, null, teléfono, null, lugar_id, centro_id	0

TABLA 3.28: Prueba unitaria registrar paciente
FUENTE: Elaboración propia

Prueba unitaria de registro detalle paciente	
Entrada	Salida
pacientes_id, peso, talla, peso_ideal, imc,,obesidad, alcoholismo, tabaquismo, proteinuria, act_fisica, creatinina, sistole_id, diastole_id, glucosa_id, glisemia_id	1
pacientes_id, null, talla, peso_ideal, imc,,obesidad, alcoholismo, tabaquismo, proteinuria, act_fisica, creatinina, sistole_id, diastole_id, glucosa_id, glisemia_id	0
pacientes_id, peso, null, talla, peso_ideal, imc,,obesidad, alcoholismo, tabaquismo, proteinuria, act_fisica, creatinina, sistole_id, diastole_id, glucosa_id, glisemia_id	0
pacientes_id, peso, null, peso_ideal, imc,,obesidad, alcoholismo, tabaquismo, proteinuria, act_fisica, creatinina, sistole_id, diastole_id, glucosa_id, glisemia_id	0
pacientes_id, peso, talla, peso_ideal, imc,,obesidad, alcoholismo, tabaquismo, proteinuria, act_fisica, null, sistole_id, diastole_id, glucosa_id, glisemia_id	0

TABLA 3.29: Prueba unitaria registro detalle paciente
FUENTE: Elaboración propia

3.2. VALORACIÓN DEL SISTEMA

Basado en la norma ISO/IEC 9126 se ha elaborado un cuestionario de evaluación (Ver Anexo A), permite evaluar la calidad del Sistema, este resultado del software fue valorado a criterio de: un usuario identificado como administrador del sistema.

Se ha asignado un peso (valor numérico) a cada atributo del Sistema Informático de Diagnóstico de Insuficiencia Renal, de acuerdo al grado de importancia o nivel de necesidad exigida por el usuario del sistema, va desde 1 que significa muy poco importante, hasta el 10 que significa extremadamente importante, se ajustó cada pregunta de la evaluación a un atributo exigido, con estos resultados se aplicó una forma para obtener las métricas de cada atributo.

El usuario califica un puntaje desde 1 al 5, de acuerdo a la siguiente escala:

Parámetro	Puntaje
Muy Mala	1
Mala	2
Media	3
Buena	4
Muy Buena	5

TABLA 3.30: *Escala de calificación para el Sistema informático de diagnóstico de IR*
FUENTE: *Elaboración propia.*

De acuerdo a la norma ISO/IEC 9126, esta se categoriza por seis elementos para realizar la evaluación del sistema. A continuación, se detalla estos elementos:

3.2.1. Funcionalidad

REF.	ATRIBUTO	PESO	RESULTADO
1	Adecuación	10	5
2	Exactitud	10	4
3	Seguridad	10	4

TABLA 3.31. Resultado de la medida funcionalidad

FUENTE: *Elaboración propia.*

$$\text{Medida de funcionalidad} = \frac{10 * 5 + 10 * 4 + 10 * 4}{(10 + 10 + 10)} = 4,3$$

De acuerdo al resultado obtenido, se califica como valor aceptable, por lo que el Sistema Informático de Diagnóstico de Insuficiencia Renal cumple con la métrica de funcionalidad de acuerdo a los requerimientos exigidos.

3.2.2. Fiabilidad

REF.	ATRIBUTO	PESO	RESULTADO
4	Madurez	10	5
5	Recuperabilidad	10	4

TABLA 3.32. Resultado de la medida fiabilidad

FUENTE: *Elaboración propia.*

$$\text{Medida de fiabilidad} = \frac{10 * 5 + 10 * 4}{(10 + 10)} = 4.5$$

De acuerdo al resultado obtenido, se califica como valor aceptable de la evaluación del Sistema Informático de Diagnostico de Insuficiencia Renal (tiene las entradas validadas, se recupera la información en caso de un incidente) lo cual significa que el sistema es seguro y recupera los datos almacenados fácilmente.

3.3.1. Usabilidad

Ref.	Atributo	Peso	Resultado
6	Entendimiento	10	4
7	Aprendizaje	10	3
8	Operabilidad	10	4
9	Atracción	10	4

TABLA 3.33. Resultado de la medida usabilidad
FUENTE: Elaboración propia.

$$\text{Medida de usabilidad} = \frac{10 * 4 + 10 * 3 + 10 * 3 + 10 * 4}{(10 + 10 + 10 + 10)} = 3,75$$

De acuerdo al resultado obtenido en su métrica de Usabilidad, se establece que el Sistema Informático de Diagnóstico de Insuficiencia Renal tiene una aceptación normal por parte del usuario, en el manejo de las interfaces y la facilidad de aprendizaje. Por el cual se otorga casi una media puntuación, en esta fase es importante estar entre la puntuación de cuatro a cinco.

3.3.2. Eficiencia

Ref.	Atributo	Peso	Resultado
10	Comportamiento de Tiempos	10	5
11	Utilización de los recursos	10	4

TABLA 3.34. Resultado de la medida eficiencia
FUENTE: Elaboración propia.

$$\text{Medida de eficiencia} = \frac{10 * 5 + 10 * 4}{(10 + 10)} = 4,5$$

De acuerdo al resultado obtenido, se establece que el Sistema Informático de Diagnóstico de Insuficiencia Renal, cumple con los requerimientos exigidos en un determinado tiempo (hace

lo que tiene que hacer, no demora mucho tiempo su ejecución), por lo que el software obtiene la puntuación aceptable de acuerdo a la métrica de eficiencia.

3.2.3. Mantenimiento

Ref.	Atributo	Peso	Resultado
12	Capacidad de ser analizado	10	3
13	Confiabilidad	8	4
14	Estabilidad	5	3
15	Facilidad de Prueba	5	4

TABLA 3.35. Resultado de la medida mantenimiento
FUENTE: Elaboración propia.

$$\text{Medida de mantenimiento} = \frac{10 * 3 + 8 * 5 + 5 * 3 + 5 * 4}{(10 + 8 + 5 + 5)} = 3,7$$

De acuerdo al resultado obtenido, se establece que el Sistema Informático de Diagnóstico de Insuficiencia Renal cumple con los requerimientos mínimos exigidos para su mantenimiento y aceptabilidad. Por lo que se obtuvo una puntuación de **3.75**, lo cual significa que es aceptable.

3.2.4. Portabilidad

Ref.	Atributo	Peso	Resultado
16	Adaptabilidad	5	5
17	Coexistencia	5	5
18	Facilidad de Instalación	10	4

TABLA 3.36. Resultado de la medida portabilidad
FUENTE: Elaboración propia.

$$\text{Medida de portabilidad} = \frac{5 * 5 + 5 * 5 + 10 * 4}{(5 + 5 + 10)} = 4,5$$

De acuerdo al resultado obtenido, se establece que el Sistema Informático de Diagnóstico de Insuficiencia Renal cumple con los requerimientos exigidos para su portabilidad.

3.2.5. Conclusión

Atributos	Puntaje
Funcionalidad	4,3
Fiabilidad	4,5
Usabilidad	3,75
Eficiencia	4,5
Mantenimiento	3,75
Portabilidad	4,5
Promedio	4,21

TABLA 3.37. *Resultado generales*
FUENTE: *Elaboración propia.*

Como se observa en la (TABLA 3.37), indica que el Sistema Informático de Diagnóstico de Insuficiencia Renal cumple con las Métricas de Calidad de la Norma **ISO/IEC 9126**, por lo que **SÍ** es pertinente utilizarlo en la unidad de enfermedades no transmisibles.

CAPITULO IV

4. CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIÓN

- ✓ Se logró diseñar el sistema informático usando la metodología de proceso unificado por ser esta metodología una de las más usadas y reconocidas.
- ✓ Se logró alcanzar con el objetivo planteado del desarrollo del sistema informático de diagnóstico de insuficiencia renal en la unidad de enfermedades no transmisibles del SEDES PANDO.
- ✓ Se logró obtener resultados positivos, con lo cual se cumple con el objetivo general que ha contribuido de manera eficiente en minimizar el tiempo en la búsqueda, y la actualización de información del diagnóstico de pacientes.
- ✓ Permite realizar un mejor control de pacientes para que la unidad pueda tomar medidas de prevención, se muestra el resultado del estado de diagnóstico de los pacientes con información precisa y confiable, la interfaz es amigable para el usuario, con pantallas comprensibles y de fácil manejo
- ✓ El conocimiento adquirido durante la carrera fue útil, dado que el concepto aprendido ha contribuido a la creación del sistema.

4.2. RECOMENDACIONES

Con la finalización del presente proyecto se pueden dar las siguientes recomendaciones:

- ✓ Se debe tener cuidado respecto a las claves de acceso que son amigables a los usuarios por única vez.
- ✓ Utilizar las herramientas similares al desarrollo del sistema para futuras construcciones de software.
- ✓ Se recomienda realizar un análisis respectivo de los requisitos que manifieste el usuario para poder satisfacer con las necesidades del usuario.
- ✓ Para la generación de los diagramas UML es necesario saber, que herramientas nos ayudan para su creación, como también saber los conceptos y funciones de cada uno de los diagramas al momento de la implementación representativa de cada uno de los objetos involucrados en el desarrollo del proyecto.

BIBLIOGRAFÍA

- Alarcon, R. (2000). *Diseño Orientado a Objetos con UML*. Madrid - España: Eidos.
- Busch, V. D. (10 de Agosto de 2014). *Hoy Bolivia.com*. Obtenido de La donación de órganos cadavéricos permite salvar vidas humanas:
http://hoybolivia.com/Noticia.php?IdNoticia=116202&tit=la_donacion_de_organos_cadavericos_permite_salvar_vidas_humanas
- Chan, D. M. (2010). *Informe sobre la situación Mundial de las Enfermedades no Transmisibles*. suiza: OMS.
- Dr Santos Depine, D. A. (mayo de 2007). *Programa Nacional de Prevención y Control de Enfermedades Renales*. Obtenido de MINISTERIO DE SALUD Y DEPORTES:
https://www.google.com.bo/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0ahUKEwjo0-6Z5eDMAhUJJh4KHQJ6Bt0QFggdMAA&url=http%3A%2F%2Fwww.fundacionvidasaludable.org%2FDocs%2FRenalHealthProgram%2FProg_Prevenci%25C3%25B3n_%2520Control_Enf_Renales_
- Dra. Aida Venado Estrada, D. J. (2010). *INSUFICIENCIA RENAL CRÓNICA*. Mexico: LAMAC.
- Gaucht, J. D. (2012). En J. D. Gaucht, *El Gran Libro de HTML5. CSS3 y Javascript* (pág. 42). Barcelona- España: Marcombo S.A.
- Gilfillan, I. (2010). *Mysql*. Anaya.
- Ivar Jacobson, G. B. (2000). *El Proceso Unificado de desarrollo de Software*. Madrid - España: Pearson Educacion S.A.
- kenneth kendall, J. k. (2011). *Análisis y Diseño de Sistemas*. Mexico: Pearson Educacion Octava edicion.
- Larman, C. (1999). *UML y patrones*. Mexico: 1ra edicion.
- Larman, C. (2003). *UML y Patrones 2da edicion*. En C. Larman. Madrid España: Pearson Educacion S.A.
- Luca, J. I. (2011). *HTML5*. Madrid España: ANAYA MULTIMEDIA.
- Marco, B. S. (21 de septiembre de 2014). *Páginas web con PHP*. Obtenido de Que es Xampp: http://www.mclibre.org/consultar/php/otros/in_php_xampp.html

- Martin, J. (1977). Organizacion de las Bases de Datos. En *organizacion logica* (pág. 19). Mexico: Prentice/Hall.
- Miguel Angel Alvarez, C. R. (10 de abril de 2010). *desarrolloweb.com*. Recuperado el 20 de octubre de 2015, de Manual de Laravel 5:
<http://desarrolloweb.com/manuales/manual-laravel-5.html>
- Miranda, N. H. (2012). *Manual de Organizacion y Funciones* . La Paz Bolivia.
- Morales, R. (1 de Septiembre de 2014). *Lenguajes de programación: ¿qué son y para qué sirven?* . Obtenido de Colombia Digital:
<https://colombiadigital.net/actualidad/articulos-informativos/item/7669-lenguajes-de-programacion-que-son-y-para-que-sirven.html>
- Orlando, J. (29 de abril de 2012). *Salud Pública*. Obtenido de LAS ENFERMEDADES CRÓNICAS NO TRANSMISIBLES (ECNT):
<http://saludpublicajaviuis.blogspot.com/2012/04/las-enfermedades-cronicas-no.html>
- Pacheco, D. A. (2007). *Programa Nacional de Prevencion de Enfermedades Renales*. La Paz.
- Pérez, W. (22 de marzo de 2015). *La Razon*. Obtenido de Enfermos renales en Bolivia:
http://www.la-razon.com/sociedad/enfermos-renales-Bolivia-tercera-edad_0_2238376248.html
- PHP.NET*. (17 de enero de 2012). Obtenido de <http://php.net/manual/es/book.mysql.php>
- Pressman, R. (2002). Factores de la calidad. En R. Pressman, *INGENIERÍA DEL SOFTWARE. UN ENFOQUE PRÁCTICO septima edicion* (pág. 343). Mexico: ISBN.
- Pressman, R. S. (1998). *Ingenieria de Software un Enfoque Practico*. 5ta Edicion.
- Ramos, M. R. (16 de junio de 2015). *Los 10 mejores Frameworks de PHP para el 2015*. Obtenido de <http://www.elwebmaster.com/general/los-10-mejores-frameworks-de-php-para-el-2015>
- Salamea, C. G. (2003). *La Informática Médica y los Sistemas de Información*. Santiago.
- Suarez, H. B. (27 de noviembre de 1998). *Ministerio de Salud*. Obtenido de Normativas leyes y decretos: Url: <http://www.minsalud.gob.bo>