

UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

“SISTEMA WEB PARA LA GESTIÓN DE HISTORIAS CLÍNICAS
ELECTRÓNICAS PERSONALIZABLES POR ESPECIALIDAD”
CASO: COLEGIO MÉDICO DEPARTAMENTAL DE LA PAZ

Para optar al título de Licenciatura en Ingeniería de Sistemas
Mención: Mención Informática y Comunicaciones

Postulante: Jhenny Blanca Sinka Colmena
Tutor Metodológico: Ing. Marisol Arguedas Balladares
Tutor Especialista: Dr. Gimmy Nardó Sanjines Tudela, Ph. D. - P.Ph.D.
Tutor Revisor: Lic. Maria Magdalena Aguilar Guanto

EL ALTO – BOLIVIA

2019

Dedicatoria

Dedico este trabajo a mi madre Victoria Colmena por hacer de mi todo lo que hoy soy.

A mi querida familia por estar siempre a mi lado brindándome apoyo incondicional en todo momento.

Agradecimientos

Primero agradecer a Dios por haberme dado una familia, amigos y una vida maravillosa.

Al Dr. Ricardo Landívar Córdova, que con su gran calidad como persona y profesionalismo me brindo la oportunidad de trabajar en su entorno con el Colegio Medico Departamental de La Paz y motivar el desarrollo del presente trabajo y al Colegio Medico Departamental de La Paz por incentivar y apoyar el desarrollo del presente proyecto.

A mi tutor Dr. Gimmy Nardo Sanjinez Tudela, Ph. D.-P.Ph.D. que, con su gran calidad como persona, profesionalismo y experiencia, acompañó el desarrollo de este trabajo desde los inicios apoyando y aportando siempre con sus conocimientos y consejos oportunos.

A mi tutora Ing. Marisol Arguedas Balladares, quien con gran calidad como profesional realizo el seguimiento de este trabajo. Los consejos, observaciones, correcciones, acompañados de su larga experiencia fueron aporte invaluable para la elaboración de este proyecto.

A la Lic. María Magdalena Aguilar Guanto por haberme brindado la oportunidad de adquirir nuevos conocimientos y también a supervisar el presente proyecto.

Y por último a mi familia y las personas más allegadas a mi por apoyarme en este proceso y hacer que se haga una realidad, dándome siempre su apoyo incondicional.

RESUMEN

El presente proyecto de grado titulado “Sistema web para la gestión de Historias Clínicas Electrónicas personalizables por especialidad Caso: Colegio Médico Departamental de La Paz” se clasifica dentro de los sistemas de información, los mismos que son considerados como aplicaciones que ayudan al trabajo diario de personas dependientes de una empresa u organización.

En la actualidad los sistemas de información en el área de la salud se convirtieron en una pieza fundamental y precisa para el buen desempeño en la atención prestada por los profesionales de salud, esto para que el desarrollo de sus procesos sustantivos y de apoyo sean más óptimos y orientados a brindar una mejor atención con calidad y calidez.

Por la necesidad de informatizar y automatizar las tareas del registro de historias clínicas de los profesionales en salud es que surge este proyecto, colaborando así con los médicos en tener a la mano los antecedentes de sus pacientes, obteniendo a la vez información actualizada y disponible para el personal de administración.

El sistema web para la gestión de historias clínicas electrónicas, posee los siguientes módulos: Administración de usuarios, que realiza el registro y modificación de los usuarios que acceden al sistema; Asignación de ficha para la atención médica, la cual registra a un nuevo paciente si se dio el caso o lo busca para asignarle una ficha de atención; Registro de triage, realiza el registro de los signos vitales antes de que el paciente ingrese a la consulta con el médico especialista; Gestión de historia clínica, se podría decir que este es el corazón del sistema ya que en este módulo se hace el registro de todo lo concerniente con la salud del paciente, su estado y su evolución; y por último el módulo de reportes, el cual genera la información requerida por parte del usuario para la elaboración de informes médicos o certificados médicos.

El sistema fue desarrollado bajo la metodología ágil Scrum el cual permite organizar y priorizar aspectos esenciales del desarrollo del sistema, con un gestor de base de datos PostgreSQL y lenguaje de programación PHP versión 7.

La calidad del producto esta basada en la norma de calidad ISO 9126, mediante la cual se probaron las funcionalidad y usabilidad del sistema. La seguridad esta basada a través de autenticación de usuario y roles de función para el sistema.

Tabla de Contenidos

1. CAPTULO I: MARCO PRELIMINAR	1
1.1. Introducción	2
1.2. Antecedentes	3
1.2.1. Antecedentes institucionales.....	3
1.2.2. Antecedentes académicos.....	4
1.3. Planteamiento del problema.....	6
1.3.1. Formulación del problema.....	7
1.4. Objetivos.....	8
1.4.1. Objetivo general.....	8
1.4.2. Objetivos específicos.....	8
1.5. Justificación	8
1.5.1. Justificación técnica.....	8
1.5.2. Justificación económica.....	9
1.5.3. Justificación social.....	9
1.6. Metodología.....	9
1.6.1. Método de ingeniería.....	10
1.6.2. Métricas y Pruebas de calidad	11
1.6.2.1. Métricas de calidad.....	11
1.6.2.2. Pruebas de calidad.....	12
1.7. Herramientas.....	12
1.8. Limites y alcances.....	12
1.8.1. Límites.....	12
1.8.2. Alcances.....	13
1.9. Aportes.....	14
2. CAPITULO II: MARCO TEORICO.....	15
2.1. Definición de Colegio Médico.....	16
2.2. Definición de establecimiento de salud	16
2.2.1. Categorización en establecimientos de salud.....	17
2.3. Expediente Clínico.....	17
2.3.1. Norma técnica del expediente clínico.....	17
2.3.2. ¿Qué es un expediente clínico?.....	18
2.3.3. Contenido del expediente clínico.....	18
2.3.3.1. Contenido general.....	18
2.3.3.2. Contenido específico.....	19
2.3.3.3. Contenido adicional.....	21
2.4. Ingeniería de requerimientos	22

2.4.1.	¿Para qué un proceso de ingeniería de requerimientos?	24
2.4.2.	Definición de ingeniería de requerimientos.....	24
2.4.3.	Actividades de la ingeniería de requerimientos.....	25
2.4.4.	Recolección de datos.	27
2.4.4.1.	Entrevistas.	27
2.4.4.2.	Cuestionarios.....	28
2.4.4.3.	Encuesta.....	28
2.4.4.4.	La observación.....	29
2.4.4.5.	Diagramas de Flujo.....	29
2.4.5.	Características de los requerimientos.	29
2.4.6.	Importancia de la ingeniería de requerimientos.....	30
2.4.7.	Requerimientos funcionales y no funcionales.	31
2.4.7.1.	Requerimientos funcionales.	32
2.4.7.2.	Requerimientos no funcionales.	33
2.5.	Ingeniería de software.....	35
2.5.1.	Metodología de desarrollo de software.....	36
2.5.1.1.	Metodologías tradicionales.....	36
2.5.1.2.	Metodologías ágiles.....	37
2.5.1.3.	Diferencia entre metodologías tradicionales y ágiles.	38
2.6.	Metodología SCRUM	39
2.6.1.	¿Qué es Scrum?	39
2.6.2.	Teoría scrum.	40
2.6.3.	Ciclo de vida de scrum.....	41
2.6.3.1.	Antes del desarrollo (pregame).	42
2.6.3.2.	Fase del desarrollo (development phase).	44
2.6.3.3.	Después del desarrollo (postgame).....	45
2.6.4.	Roles y responsabilidades de scrum.	46
2.6.4.1.	Los cerdos.....	46
2.6.4.1.	Las gallinas.....	50
2.6.5.	Eventos de scrum (reuniones).....	50
2.6.5.1.	¿Qué es un sprint?	51
2.6.5.2.	Planificación del Sprint.	52
2.6.5.3.	Objetivo del sprint.	53
2.6.5.4.	Scrum diario.	53
2.6.5.5.	Revisión del sprint.....	54
2.6.5.6.	Retrospectiva del sprint.	55
2.6.6.	Elementos de Scrum.	56
2.6.6.1.	Product Backlog (backlog de producto).	56

2.6.6.2.	Sprint Backlog (backlog del sprint).....	57
2.6.6.3.	Incremento.....	58
2.6.6.4.	Transparencia en los elementos.....	58
2.7.	Ingeniería web.....	59
2.7.1.	Definición.....	60
2.8.	Metodología UWE.....	60
2.8.1.	Lenguaje unificado de modelado (UML).....	60
2.8.1.	UML – based web engineering (UML).....	61
2.8.2.	Modelo de requisitos.....	64
2.8.2.1.	Casos de uso.....	64
2.8.2.2.	Actividades.....	65
2.8.3.	Modelo de contenido.....	67
2.8.4.	Modelo de navegación.....	68
2.8.5.	Modelo de presentación.....	69
2.8.6.	Modelo de proceso.....	70
2.8.6.1.	Modelo de Estructura del Proceso.....	70
2.8.6.1.	Modelo de flujo del Proceso.....	71
2.9.	Calidad de software.....	72
2.10.	Modelo de calidad establecido por el estándar ISO 9126.....	74
2.10.1.	¿Qué es la norma ISO 9126?.....	74
2.10.2.	Estándar.....	74
2.10.1.	Características.....	75
2.10.1.1.	Funcionalidad.....	75
2.10.1.2.	Confiabilidad.....	76
2.10.1.3.	Usabilidad.....	77
2.10.1.4.	Eficiencia.....	77
2.10.1.5.	Mantenibilidad.....	78
Capacidad de análisis.....	78	
2.10.1.6.	Portabilidad.....	79
2.11.	Estimación de Costos.....	79
2.12.	Modelo COCOMO.....	81
3.	CAPITULO III: MARCO APLICATIVO.....	86
3.1.	Introducción.....	87
3.2.	Diagnóstico del sistema actual.....	87
3.2.1.	Descripción de la institución.....	87
3.2.1.1.	Especialidades.....	88
3.2.1.2.	Organización de la institución.....	89
3.2.2.	Actual forma de manejo de la información.....	90

3.2.2.1.	Lista de actores del negocio.	92
3.2.2.2.	Diagramas de casos de uso del negocio.	92
3.2.2.3.	Diagramas de secuencias del negocio.	96
3.2.3.	Identificación de Roles Scrum.	98
3.3.	Pre – Game (Antes del desarrollo).....	98
3.3.1.	Creación del product backlog (Pila de producto).	98
3.3.1.1.	Requerimientos funcionales.	98
3.3.1.2.	Requerimientos no funcionales.	100
3.3.2.	Modelos de casos de uso general del sistema.	100
3.3.2.1.	Identificación de actores del sistema.	101
3.3.2.2.	Diagrama general de casos de uso del sistema.	101
3.3.3.	Diseño de la base de datos.	102
3.3.3.1.	Diagrama entidad-relación general.	103
3.3.3.2.	Estructura de la base de datos.	103
3.4.	Game (durante el desarrollo).	104
3.4.1.	SPRINT 1: Módulo de inicio de sesión.	105
3.4.1.1.	Planificación del Sprint 1.	105
3.4.1.2.	Diagrama de caso de uso del Sprint 1.	105
3.4.1.3.	Diseño navegacional del Sprint 1.	106
3.4.1.4.	Diseño Presentación del Sprint 1.	107
3.4.1.5.	Pantallas del Sprint 1.	107
3.4.1.6.	Prueba unitaria del Sprint 1.	108
3.4.2.	SPRINT 2: Módulo de administración de usuario.	108
3.4.2.1.	Planificación del Sprint 2.	108
3.4.2.2.	Diagrama de caso de uso del Sprint 2.	109
3.4.2.3.	Diseño navegacional del Sprint 2.	112
3.4.2.4.	Diseño de presentación del Sprint 2.	112
3.4.2.5.	Pantallas del Sprint 2.	113
3.4.2.6.	Prueba unitaria del Sprint 2.	114
3.4.3.	SPRINT 3: Módulo de Asignación de ficha.	114
3.4.3.1.	Planificación del Sprint 3.	114
3.4.3.2.	Diagrama de caso de uso del Sprint 3.	115
3.4.3.3.	Diseño navegacional del Sprint 3.	118
3.4.3.4.	Diseño de presentación del Sprint 3.	118
3.4.3.5.	Pantallas del Sprint 3.	120
3.4.3.6.	Prueba unitaria del Sprint 3.	121
3.4.4.	SPRINT 4: Módulo de Registro de Triage.	121
3.4.4.1.	Planificación del Sprint 4.	121

3.4.4.2.	Diagrama de caso de uso del Sprint 4.	122
3.4.4.3.	Diseño navegacional del Sprint 4.	124
3.4.4.4.	Diseño de presentación del Sprint 4.	125
3.4.4.5.	Pantallas del Sprint 4.	128
3.4.4.6.	Prueba unitaria del Sprint 4.	128
3.4.5.	SPRINT 5: Módulo de gestión de historia clínica.	129
3.4.5.1.	Planificación del Sprint 5.	129
3.4.5.2.	Diagrama de caso de uso del Sprint 5.	130
3.4.5.3.	Diseño navegacional del Sprint 5.	133
3.4.5.4.	Diseño de presentación del Sprint 5.	134
3.4.5.5.	Pantallas del Sprint 5.	140
3.4.5.6.	Prueba unitaria del Sprint 5.	141
3.4.6.	SPRINT 6: Módulo de generación de reportes.	141
3.4.6.1.	Planificación del Sprint 6.	141
3.4.6.2.	Diagrama de caso de uso del Sprint 6.	142
3.4.6.3.	Diseño navegacional del Sprint 6.	142
3.4.6.4.	Diseño de presentación del Sprint 6.	143
3.4.6.5.	Pantallas del Sprint 6.	144
3.4.6.6.	Prueba unitaria del Sprint 6.	144
3.5.	Post – Game (Después del desarrollo).	144
3.5.1.	Pruebas de caja negra.	144
3.6.	Calidad de software - ISO 9126.	148
3.6.1.	Usabilidad.	148
3.6.2.	Funcionalidad.	150
3.6.3.	Confiabilidad.	153
3.6.4.	Mantenibilidad.	154
3.6.5.	Portabilidad.	155
3.7.	Estimación de Costos - COCOMO.	157
3.7.1.	Costo del Software desarrollado.	157
3.7.2.	Costo de implementación de la aplicación.	161
3.7.3.	Costo de elaboración del proyecto.	161
3.7.4.	Costo total del proyecto.	162
4.	CAPÍTULO IV: CONCLUSIONES Y RECOMENDACIONES.	163
4.1.	Conclusiones.	164
4.2.	Recomendaciones.	165

Lista de tablas

Tabla 1. Tipos de categorización de establecimientos de salud	17
Tabla 2. Ejemplo tabla de requerimientos funcionales.....	33
Tabla 3. Ejemplo de tabla de requerimientos no funcionales.....	35
Tabla 4. Comparativa metodología Tradicional y ágil	38
Tabla 5. Tareas de la fase Preamble.....	43
Tabla 6. Tareas de la fase de desarrollo.....	45
Tabla 7. Esquema de modos de desarrollo de software.....	82
Tabla 8. Valores constantes por modo de desarrollo.....	83
Tabla 9. Ecuaciones por tipo de modelo COCOMO: Básico e intermedio.....	83
Tabla 10. Ecuaciones por tipo de modelo COCOMO: Básico e intermedio.....	84
Tabla 11. Especialidades dependientes al Colegio Médico Departamental de La Paz	88
Tabla 12. Identificación de actores en el negocio.	92
Tabla 13. Descripción del CUN para solicitar atención médica.....	93
Tabla 14. Descripción del CUN para registrar triaje	94
Tabla 15. Descripción del CUN para consulta médica.....	96
Tabla 16. Identificación de roles de Scrum	98
Tabla 17. Requerimientos funcionales del sistema	99
Tabla 18. Requerimientos no funcionales del sistema.	100
Tabla 19. Listado del Product Backlog	100
Tabla 20. Identificación de actores en el sistema.	101
Tabla 21. Sprint Backlog: Inicio de Sesión	105
Tabla 22. Descripción del Caso de Uso: Registro de Usuario.....	106
Tabla 23. Prueba unitaria: Inicio de Sesión.....	108
Tabla 24. Backlog: Administración de usuarios.....	109
Tabla 25. Descripción del Caso de Uso: Registro de Usuario.....	110
Tabla 26. Descripción de Caso de Uso: Asignar Rol	111
Tabla 27. Descripción de Caso de Uso: Modificar e inhabilitar usuario.....	111
Tabla 28. Prueba unitaria: Administración de Usuarios	114
Tabla 29. Sprint Backlog: Asignar ficha de atención.....	115
Tabla 30. Descripción del Caso de Uso: Registro de Paciente.....	116
Tabla 31. Descripción de Caso de Uso: Listar Pacientes	117
Tabla 32. Descripción de Caso de Uso: Asignar ficha	117
Tabla 33. Descripción de Caso de Uso: Listar Médicos.....	117
Tabla 35. Prueba unitaria: Asignar ficha de atención.....	121
Tabla 36. Sprint Backlog: Registro de Triaje	122
Tabla 37. Descripción del Caso de Uso: Listar pacientes en espera.....	123

<i>Tabla 38.</i> Descripción del Caso de Uso: Administrar medicamentos	124
<i>Tabla 39.</i> Descripción del Caso de Uso: Registrar resultados de laboratorio.....	124
<i>Tabla 40.</i> Prueba unitaria: Inicio de Sesión	129
<i>Tabla 41.</i> Sprint Backlog: Gestión de Historias Clínicas	129
<i>Tabla 42.</i> Descripción del Caso de Uso: Registro de anamnesis.....	131
<i>Tabla 43.</i> Descripción del Caso de Uso: Registro de examen clínico	131
<i>Tabla 44.</i> Descripción del Caso de Uso: Emisión de diagnóstico	131
<i>Tabla 45.</i> Descripción del Caso de Uso: Registro de plan de trabajo.....	132
<i>Tabla 46.</i> Descripción del Caso de Uso: Registro del tratamiento	132
<i>Tabla 47.</i> Descripción del Caso de Uso: Registro de evolución	133
<i>Tabla 48.</i> Prueba unitaria: Gestión de historia clínica.....	141
<i>Tabla 49.</i> Sprint Backlog: Generación de reportes	141
<i>Tabla 50.</i> Descripción del Caso de Uso: Generar reportes	142
<i>Tabla 51.</i> Prueba unitaria: Generación de Reportes	144
<i>Tabla 52.</i> Caso de Prueba: Buscar Historia Clínica.....	145
<i>Tabla 53.</i> Caso de Prueba: Registrar Atención Médica.....	146
<i>Tabla 54.</i> Caso de Prueba: Registrar nuevo paciente	146
<i>Tabla 55.</i> Caso de Prueba: Gestionar Historia Clínica	147
<i>Tabla 56.</i> Encuesta de usabilidad del sistema.....	149
<i>Tabla 57.</i> Conteo de parámetros punto de función con sus respectivas características y pesos	151
<i>Tabla 58.</i> Ajuste de complejidad con el cuestionario correspondiente para su evaluación	152
<i>Tabla 59.</i> Cálculo de Portabilidad.....	156
<i>Tabla 60.</i> Resultados Finales	156
<i>Tabla 61.</i> Conversión de puntos de función	157
<i>Tabla 62.</i> Valores de las constantes de acuerdo al modelo COCOMO	158
<i>Tabla 63.</i> Conductores de costos	158
<i>Tabla 64.</i> Resultados Finales	161
<i>Tabla 65.</i> Resultados Finales	162

Lista de figuras

<i>Figura 1:</i> Organigrama Colegio Médico Departamental de La Paz.....	4
<i>Figura 2:</i> Características del modelo de calidad establecido por el estándar ISO 9126.	11
<i>Figura 3:</i> Ciclo de vida de Scrum.	42
<i>Figura 4:</i> Ejemplo de distribución de roles en scrum.....	46
<i>Figura 5:</i> Eventos de la Metodología Scrum.	51
<i>Figura 6:</i> Modelos de UWE.	63
<i>Figura 7:</i> Nombres de estereotipos y sus íconos casos de uso.	64
<i>Figura 8:</i> Casos de uso de un ejemplo de agenda de direcciones.....	65
<i>Figura 9:</i> Nombres de estereotipos y sus íconos actividades.	65
<i>Figura 10:</i> Actividad para el caso de uso "CreateContact" del ejemplo agenda de direcciones.	66
<i>Figura 11:</i> Modelo de contenido para el ejemplo agenda de direcciones.	67
<i>Figura 12:</i> Nombres de estereotipos y sus íconos del modelo de navegación.	68
<i>Figura 13:</i> Modelo de navegación para el ejemplo agenda de direcciones.	68
<i>Figura 14:</i> Nombres de estereotipos y sus íconos del modelo de presentación.	69
<i>Figura 15:</i> Modelo de presentación para el ejemplo agenda de direcciones.	70
<i>Figura 16:</i> Modelo de estructura del proceso para el ejemplo agenda de direcciones.	71
<i>Figura 17:</i> Nombres de estereotipos y sus íconos del modelo de proceso.	71
<i>Figura 18:</i> Modelo de flujo del proceso para el ejemplo agenda de direcciones	72
<i>Figura 19:</i> Atributos del modelo de calidad ISO 9126.....	75
<i>Figura 20:</i> Conceptualización básica de COCOMO.	82
<i>Figura 21:</i> Organigrama del Colegio Médico Departamental de La Paz	90
<i>Figura 22:</i> Flujograma del proceso de atención al paciente.	91
<i>Figura 23:</i> Caso de uso para solicitar atención médica.	93
<i>Figura 24:</i> Caso de uso para registrar triaje.....	94
<i>Figura 25:</i> Caso de uso para consulta médica.	95
<i>Figura 26:</i> Diagrama de secuencia para la solicitar de atención médica.....	96
<i>Figura 27:</i> Diagrama de secuencia para registrar triaje.....	97
<i>Figura 28:</i> Diagrama de secuencia para consulta médica	97
<i>Figura 29:</i> Diagrama de Casos de Uso general del sistema.	102
<i>Figura 30:</i> Diagrama Entidad – Relación.	103
<i>Figura 31:</i> Estructura general de la Base de Datos.....	104
<i>Figura 32:</i> Diagrama de Caso de Uso: Inicio de Sesión.....	106
<i>Figura 33:</i> Diagrama de Navegación: Inico de Sesión.	107
<i>Figura 34:</i> Diagrama de Navegación: Inico de Sesión.	107
<i>Figura 35:</i> Diagrama de Navegación: Inico de Sesión.....	108

<i>Figura 36.</i> Diagrama de Caso de Uso: Administración de Usuarios.....	110
<i>Figura 37.</i> Diagrama de Navegación: Administración de Usuario.	112
<i>Figura 38.</i> Diagrama de Presentación: Registro nuevo usuario.	112
<i>Figura 39.</i> Diagrama de Presentación: Administración de Usuario.	113
<i>Figura 40.</i> Captura de pantalla de registro de nuevo usuario.....	113
<i>Figura 41.</i> Captura de pantalla de detalle de usuarios.....	114
<i>Figura 42.</i> Diagrama de Caso de Uso: Asignación de Turno.....	116
<i>Figura 43.</i> Diagrama de Navegación: Asignar Ficha de Atención.....	118
<i>Figura 44.</i> Diagrama de Presentación: Mostrar agenda.	118
<i>Figura 45.</i> Diagrama de Presentación: Asignar ficha	119
<i>Figura 46.</i> Diagrama de Presentación: Registrar paciente.	119
<i>Figura 47.</i> Diagrama de Presentación: Admisión de Paciente	120
<i>Figura 48.</i> Diagrama de Presentación: Admisión de Paciente.....	120
<i>Figura 49.</i> Diagrama de Presentación: Admisión de Paciente	121
<i>Figura 50.</i> Diagrama de Caso de Uso: Registro de Triage.....	123
<i>Figura 51.</i> Diagrama de Navegación: Registro de triaje.	125
<i>Figura 52.</i> Diagrama de Presentación: Registro de triaje	125
<i>Figura 53.</i> Diagrama de Presentación: Registro de signos vitales.	126
<i>Figura 54.</i> Diagrama de Presentación: Registro de medicamentos.	127
<i>Figura 55.</i> Diagrama de Presentación: Registro de resultados de laboratorio.	127
<i>Figura 56.</i> Captura de Pantalla: Registro de Triage.....	128
<i>Figura 57.</i> Captura de Pantalla: Registro de Triage.....	128
<i>Figura 58.</i> Diagrama de Caso de Uso: Gestión de Historia Clínica	130
<i>Figura 59.</i> Diagrama de Navegación: Gestión de Historia Clínica.....	133
<i>Figura 60.</i> Diagrama de Presentación: Gestión de Historia Clínica.....	134
<i>Figura 61.</i> Diagrama de Presentación: Historia Clínica - Antecedentes.....	134
<i>Figura 62.</i> Diagrama de Presentación: Historia Clínica – Examen Clínico.	135
<i>Figura 63.</i> Diagrama de Presentación: Historia Clínica – Exámenes Complementarios ...	135
<i>Figura 64.</i> Diagrama de Presentación: Historia Clínica – EC Generar Orden.....	136
<i>Figura 65.</i> Diagrama de Presentación: Historia Clínica – EC Registrar Resultados.	136
<i>Figura 66.</i> Diagrama de Presentación: Historia Clínica – EC Examen de Laboratorio	137
<i>Figura 67.</i> Diagrama de Presentación: Historia Clínica – EC Imogeneología.....	137
<i>Figura 68.</i> Diagrama de Presentación: Historia Clínica – Diagnóstico.....	138
<i>Figura 69.</i> Diagrama de Presentación: Historia Clínica – Seguimiento.....	138
<i>Figura 70.</i> Diagrama de Presentación: Historia Clínica – Registrar Seguimiento.	139
<i>Figura 71.</i> Diagrama de Presentación: Historia Clínica – Evolución del Paciente.	139
<i>Figura 72.</i> Captura de pantalla: Gestión de Historias Clínicas.	140
<i>Figura 73.</i> Captura de pantalla: Gestión de Historias Clínicas.	140

<i>Figura 74.</i> Diagrama de Caso de Uso: Generar Reportes.	142
<i>Figura 75.</i> Diagrama de Navegación: Generar Reportes.....	143
<i>Figura 76.</i> Diagrama de Presentación: Generar Reportes	143
<i>Figura 77.</i> Captura de pantalla: Generar Reportes.	144

CAPÍTULO I

MARCO PRELIMINAR

RESUMEN

En este capítulo se introduce a la temática de investigación propuesta, se describe el problema, los objetivos que conducen a la solución del problema, herramientas de software empleadas, los alcances y las limitaciones para el desarrollo de software.

1.1. Introducción

Las herramientas en tecnologías de información se han incorporado a todas las actividades del que hacer del hombre, conforme se difunde con gran rapidez el uso de las computadoras dentro de las organizaciones, surgen muchas inquietudes acerca de su uso para mejorar la productividad y mejorar los objetivos de la organización. Con la evolución de las tecnologías informáticas, nuestra sociedad, tiende cada día a automatizar labores cotidianas que anteriormente lo realizaban de forma manual. Esto ha llevado a varias instituciones u organizaciones a automatizar y aprovechar de este modo los recursos tecnológicos para mejorar el desempeño de sus actividades.

En los últimos años se han visto cambios muy significativos en cuanto al uso de las tecnologías de la información y la comunicación (TIC), se va difundiendo con gran rapidez el uso de las computadoras dentro de las organizaciones, las cuales desean mejorar la productividad y optimizar los objetivos de la organización. Con la evolución de las tecnologías informáticas, nuestra sociedad, tiende cada día a automatizar labores cotidianas que anteriormente lo realizaban de forma manual. Esto ha llevado a varias instituciones u organizaciones a automatizar y aprovechar de este modo los recursos tecnológicos para mejorar el desempeño de sus actividades. Como es lógico la medicina no ha escapado a esta revolución tecnológica. Se está dando un cambio trascendental en la forma de generar, consultar y comunicar la información clínica. Ya es posible pensar que han desaparecido muchas de las barreras que impedían una comunicación a distancia, simultánea y en cualquier momento con otros profesionales asistenciales.

En nuestro país distintos establecimientos de salud públicos y privados no cuentan con un elemento integrador de datos ya que estos se encuentran dispersos en diferentes archivos lo cual dificulta la atención de los pacientes,

provocando que los niveles de insatisfacción con los servicios de salud y las demandas de mejora en la calidad de la atención sean constantes. Por otro lado, la gran cantidad de pacientes que acuden a establecimientos de salud generan cantidades de información que son registrados de manera física, generando así demora al momento de realizar la atención, por ello surge la necesidad de sistematizar estas actividades cotidianas por medio de las Tecnologías de la Información y Comunicación.

Ante esta situación el Colegio Médico Departamental de La Paz busca el automatizar e integrar los procesos, en cuanto al manejo de información asociada a los pacientes por lo cual plantea el desarrollo e implementación de una plataforma web de fácil, ágil y portátil administración de Historiales Clínicos y que al mismo tiempo presente reportes al médico. En ese sentido se desarrollará un sistema web para la gestión de historias clínicas electrónicas personalizables por especialidad la cual podrá almacenar toda la información de los distintos pacientes, sin el temor que la información brindada por el paciente se pierda, más al contrario se guarde de manera segura. El desarrollo de la misma se realizó bajo la metodología de desarrollo ágil SCRUM, con la herramienta de UWE lenguaje que permite modelar, construir y documentar los elementos que forman un sistema, se utilizó como gestor de base de datos PostgreSQL y PHP como lenguaje de programación.

1.2. Antecedentes

1.2.1. Antecedentes institucionales. El Colegio Médico Departamental de La Paz, ubicado calle Ballivian Nro.1266 a media cuadra de la plaza Murillo, es una institución académica, científica y gremial, con reconocimiento nacional e internacional. Fue creada por D.S. 9944 del 1 de octubre de 1971, a través de la fundación del Colegio Médico de Bolivia, es la institución de los médicos del

departamento de La Paz, con actividades académica, científica, gremial y ético social, constituyendo un instrumento orgánico institucionalizado.

El colegio médico departamental de La Paz se encuentra organizado de la siguiente manera

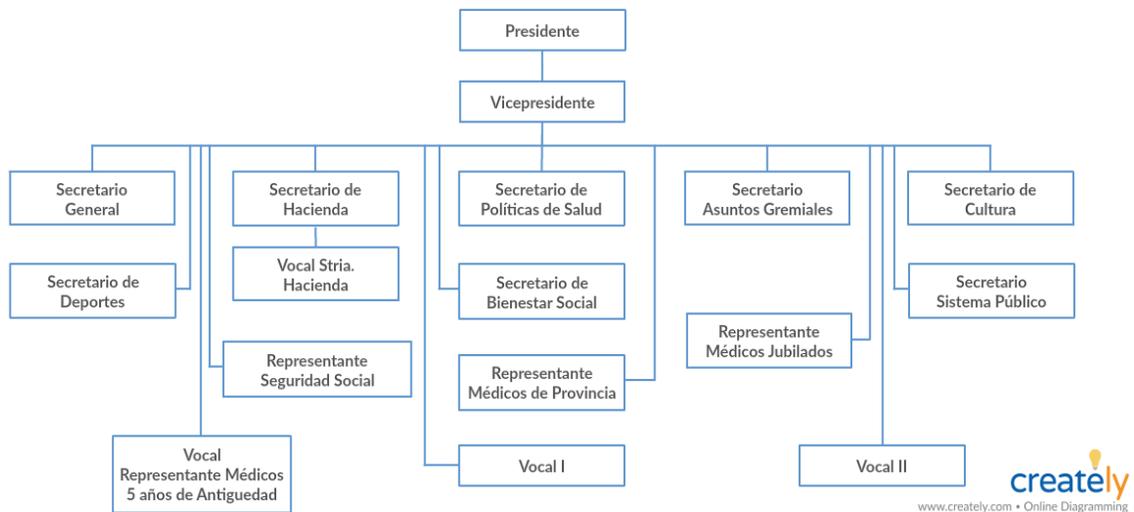


Figura 1: Organigrama Colegio Médico Departamental de La Paz.

Fuente: Colegio Médico Departamental de La Paz

Elaboración: Propia

Se pretende beneficiar a parte de los médicos que se encuentran dentro de la mesa directiva a los médicos que se encuentran asociados a la misma.

1.2.2. Antecedentes académicos. En el presente apartado se muestran los trabajos relacionados al tema objeto de estudio.

El trabajo realizado por Gutiérrez, (2017) muestra la problemática que tenía el “Centro médico quirúrgico Erzengel” la cual no contaba con un sistema de gestión hospitalaria de consulta externa, el manejo y registro de la documentación de los pacientes se realizaba manualmente en diferentes carpetas y hojas Excel. Para dar solución a la problemática se desarrollo el Sistema web de administración de historias clínicas el sistema se realizó bajo la metodología de

desarrollo ágil SCRUM, con una base de datos relacional, el diseño visual mediante un framework y la funcionalidad por medio del Modelo Vista Controlador. (Gutierrez Flores, 2017)

Por otro lado Centellas, (2015) muestra el proceso que realizo para el desarrollo un Sistema de Control y Gestión de Historiales Clínicos apoyado en dispositivos móviles para el Centro Medico La Paz, indica que el registro de historiales clínicos se lo realizaba de manera manual, razón por la cual se necesita una herramienta que automatice estas tareas debido al crecimiento de pacientes que consultan al centro médico, la misma que realizo bajo la metodología Programación Extrema (XP) y Java como lenguaje de programación. (Centellas Coarite, 2015)

El Sistema web de registro y seguimiento de pacientes, elaboración y emisión de análisis efectuados, registro y guía medica, farmacéutica y hospitalaria para el Laboratorio Clínico – Adolfo Kolping realizado por Mamani, (2014) permite centralizar toda la información de los pacientes a través de una base de datos, evitando así demoras en cuanto a tiempo de registro y actualización de información, crear informes de laboratorio y realizar inscripción de médicos y centros médicos. Para la cual utilizo la metodología de desarrollo OpenUp – UWE, y para la implementación de la aplicación web tecnologías libres como ser RUBY apoyados en el Framework TUBY ON RAILS, como gestor de base de datos PostgreSQL. (Mamani Llipe, 2014)

El mismo año Lozano, (2014) implementó un Sistema de administración y control de Historiales Clínicos para que de este modo se pueda almacenar los datos relevantes del paciente (universitario/a), de los consultorios dependientes del Departamento de Bienestar Social de la Universidad Mayor de San Andrés cuya actividad principal es brindar atención medica eficaz y eficiente. La misma fue desarrollada bajo la metodología de Modelo de Diseño Hipermedia Orientado a

Objetos OOADM, con la herramienta de UML (Unified Modeling Language), Postgresql como gestor de base de datos y Java con tecnologías NetBeans como lenguaje de programación. (Lozano Flores, 2014)

En Santo Domingo - Ecuador Gonzalez & Crespo, (2018) desarrollaron e implementaron un sistema informático para la gestión de las historias clínicas en los estudios de imagen medicas del laboratorio clínico Cedylabe en la provincia de santo domingo. Dicho sistema fue elaborado con la finalidad de reducir los tiempos empleados en el manejo de registros clínicos para los nuevos clientes y para la actualización de los datos de aquellos que ya consten en los registros, el cual es de suma importancia para prevenir la perdida de información y facilitar el proceso de las comparaciones entre los registros equivalentes. Se utilizo la metodología de desarrollo de software XP, utiliza una arquitectura Modelo Vista Controlador, Django como framework web para el desarrollo ágil, Bootstrap y JavaScript como herramienta de programación frontend, PostgreSQL para la gestión de la base de datos, Nginx y Unicorn para el soporte del servicio, además de otras librerías específicas para detalles de la aplicación. (González Alcolado & Crespo Obaco, 2018)

1.3. Planteamiento del problema

El ministro de Salud en conferencia de prensa informo que el Gobierno proyectaba digitalizar las historias clínicas de pacientes para garantizar su tratamiento, además de implementar un plan de control médico anual, en el marco de las recomendaciones que dejó el Encuentro Nacional por la Salud y la Vida, Rodolfo Rocabado recalco que "Es importante tener un mejor sistema informático que nos permita acceder a la información en tiempo real a nivel nacional (...), avanzar a lo que viene siendo la historia clínica digital, que también es un pedido de la gente. Otra cosa que plantea es que debería existir un control médico anual

de todos los ciudadanos", dijo en una entrevista con medios estatales. (La Razon, 2018)

Bajo esta iniciativa el Colegio Médico Departamental de La Paz, con alrededor de 52 especialidades y 50 médicos aproximadamente asociados por cada especialidad decide impulsar la creación de un Sistema de Historiales Clínicos Electrónicos para la gestión de los pacientes, ellos aseguran que la historia clínica en papel debería pasar por una transformación a historia clínica electrónica o historia clínica digital cuyo uso debería ser fomentado en la practica asistencial de los establecimientos de salud, no sólo permitiría mejorar la calidad de atención del paciente, sino que además lograría unificar en un único historial clínico todo el conjunto de documentación clínica que se genera a lo largo de la vida del paciente, la cual puede encontrarse recogida en varios historiales clínicos, de modo que al estar digitalizada toda esta documentación el personal que lo asiste podrá acceder de manera integra a la misma, y no sólo a la parte que se encuentra almacenada en el propio centro o que puede ser aportada por el paciente. De este modo el ciudadano podrá recibir una atención sanitaria de calidad en condición de igualdad efectiva, para lo cual es necesario que el profesional que asiste al paciente cuente con el máximo nivel de información del paciente.

1.3.1. Formulación del problema. Las historias clínicas necesitan de actualizaciones constantemente, sin importar donde se atiende al paciente (clínica, hospital, domicilio, entre otros) lo que genera, el uso excesivo de tiempo en la búsqueda de Historias clínicas, recetas, entre otros causando una información a destiempo, otro factor importante es la pérdida de historiales clínicos lo que causa que los médicos no puedan brindar una atención de calidad al paciente.

1.4. Objetivos

1.4.1. Objetivo general. Desarrollar un Sistema web para la gestión de historias clínicas electrónicas personalizables por especialidad que permita mejorar las tareas de registro, modificación, búsqueda de pacientes y elaboración de reportes para que de este modo se pueda acceder de manera rápida, eficiente, confiable y segura a una historia clínica completa, facilitando la entrega oportuna y precisa del registro de datos y manejo de la información en los consultorios al médico especialista.

1.4.2. Objetivos específicos. Los objetivos específicos planteados para el proyecto son:

- Automatizar el registro, modificación y búsqueda de pacientes, la emisión de reportes, informes e historial clínico del mismo.
- Realizar el Análisis y Diseño del Sistema web para la gestión de historias clínicas.
- Diseñar e Implementar una base de datos para el manejo de la información referente a las historias clínicas.
- Ejecutar pruebas de calidad.

1.5. Justificación

1.5.1. Justificación técnica. El Sistema web para la gestión de historias clínicas personalizables por especialidad se desarrollará en base a software libre bajo la plataforma Linux, pero será adaptable a cualquier plataforma, empleando PHP y PostgreSQL como herramientas principales para el desarrollo, tendrá un diseño responsivo el cual permitirá que se pueda acceder desde cualquier dispositivo (teléfono móvil, tablet o computadora de escritorio), además que será compatible con cualquier navegador.

1.5.2. Justificación económica. El empleo de tecnología en cualquier área es planteado con el fin de mejorar los procesos con los cuales se trabaja en cualquier contexto. Dados los procesos actuales por los cuales se manejan las historias clínicas, el uso de la tecnología significa una inversión que en general mejoraría dichos procesos. Con el empleo del sistema web se reducirían costos en tiempo y materiales de escritorio, además será desarrollado bajo plataformas libres que no incluyen costos para la compra de licencias externas lo cual hace que el presente trabajo sea sustentable económicamente.

1.5.3. Justificación social. La salud es un tema muy delicado que debe ser manejado con la mayor responsabilidad posible, el buen manejo de la documentación de los pacientes es vital para asegurar su buena atención. En Bolivia existe una gran cantidad de casos de negligencia médica denunciados y aunque existen leyes que sancionan a los profesionales médicos que son encontrados culpables no se está trabajando en prevenir la negligencia y una manera de prevención es mejorar el manejo de Historias Clínicas también es un respaldo para los médicos que son acusados injustamente por el mismo motivo.

1.6. Metodología

Como metodología se utilizará SCRUM que se basa en un proceso iterativo e incremental de diseño y desarrollo además que cuenta con un modelo de desarrollo ágil y flexible, el cual está formado por un conjunto de buenas prácticas que permiten mejorar el trabajo en equipo, y obtener mejores resultados de un proyecto. Sus principales características son:

- Define una serie de roles, de actividades y de documentos de soporte bastante estructurado.
- Considera que un proyecto puede sufrir modificaciones en su objetivo a lo largo de su realización.

- Comunicación verbal entre los miembros del equipo.
- Predisposición y respuesta al cambio.
- Se tiene distintas iteraciones, con el objetivo de conseguir pequeños productos de valor en corto tiempo

1.6.1. Método de ingeniería. El presente proyecto usara como método de ingeniería UWE (UML-Based Web Engineering) que es una propuesta para el desarrollo de aplicaciones web enfocado sobre el diseño sistemático, la personalización y la generación semiautomática de escenarios que guíen el proceso de desarrollo de una aplicación Web, esta cubre todo el ciclo de vida tipo de aplicaciones centrando además su atención en aplicaciones personalizadas o adaptativas.

UWE es una propuesta basada en el proceso unificado y UML pero adaptados a la web. En requisitos separa las fases de captura, definición y validación. Hace además una clasificación y un tratamiento especial dependiendo del carácter de cada requisito.

El método contempla los siguientes modelos:

- Análisis de requisitos
- Modelo de contenido
- Modelo de navegación
- Modelo de presentación
- Modelo de proceso

1.6.2. Métricas y Pruebas de calidad

1.6.2.1. Métricas de calidad. La métrica que se usara es ISO 9126 que es un estándar internacional para la evaluación de la calidad del software, se basa en que el objetivo no es necesariamente alcanzar una calidad perfecta, sino la necesaria y suficiente para cada contexto de uso a la hora de la entrega y del uso del software por parte de los usuarios y es necesario comprender las necesidades reales de los usuarios con tanto detalle como sea posible (requisitos).

Razones por la cual se definen los siguientes aspectos de calidad: interna (medible a partir de las características intrínsecas, como el código fuente), externa (medible en el comportamiento del producto, como en una prueba) y en uso (durante la utilización efectiva por parte del usuario).

La ISO 9126 nos permite definir un modelo de calidad, para nuestra organización, en base a las 6 características que se indican en la calidad en el ciclo de vida del software. El modelo de calidad que definamos nos dará como resultado el grado de calidad de cada uno de nuestros productos software.



Figura 2: Características del modelo de calidad establecido por el estándar ISO 9126.
Elaboración: Propia

1.6.2.2. Pruebas de calidad. Se plantea realizar pruebas de caja negra para comprobar el funcionamiento de los códigos que se vayan a implementar en el sistema.

Las pruebas de caja negra (Black-Box Testing) son pruebas funcionales. Se parte de los requisitos funcionales, a muy alto nivel, para diseñar pruebas que se aplican sobre el sistema sin necesidad de conocer como está construido por dentro (Caja negra). Las pruebas se aplican sobre el sistema empleando un determinado conjunto de datos de entrada y observando las salidas que se producen para determinar si la función se está desempeñando correctamente por el sistema bajo prueba.

1.7. Herramientas

Se utilizó herramientas CASE de apoyo para el diseño del sistema, como el Creately, que permite realizar el modelado del sistema. Para elaborar el código del programa, se utilizará en el lenguaje PHP ya que ofrece diversas características de desarrollo y uso, pero lo más importante es que es de uso libre y por tanto no requiere licencias de uso. Esta plataforma permite utilizar otros lenguajes de programación y manejo de hipertexto como HTML5, HTML y archivos JSON.

Para el almacenamiento, diseño y construcción de la base de datos, se hará uso del gestor de base de datos PostgreSQL en su versión 11 y gracias a su robustez y madurez, ofrece escudos de seguridad a distintos niveles.

1.8. Límites y alcances

1.8.1. Límites. Los límites que presenta el proyecto son:

- Función solo con conexión a internet ya que se centralizara los datos de diferentes médicos en una misma base de datos en un servidor externo.

- Las interfaces del sistema estarán restringidas a los tipos de usuarios autorizados del sistema
- Actividades económicas y la administración de un consultorio o un hospital no serán tomadas en cuenta.
- Solo podrán acceder los médicos y otros actores identificados al software y no así los pacientes.

1.8.2. Alcances. El presente proyecto realiza una investigación aplicada con las siguientes características:

- Controlar el acceso al sistema mediante la validación de la identidad de un usuario, otro sistema o dispositivo antes de autorizar su acceso.
- Las interfaces del sistema estarán restringidas a los tipos de usuarios autorizados del sistema.
- Preservación de la confidencialidad, integridad y disponibilidad de la información.
- Registro del médico, enfermera o secretaria.
- Recepción de pacientes.
- Registro de historias clínicas personalizables por especialidad.
- Registro de notas de evolución.
- Impresión de recetas médicas, informes médicos y certificados médicos.
- Generación de reportes.
- Cargado de Vademécum en la base de datos y accesible desde la aplicación.
- Generar orden para exámenes de laboratorio.
- Adaptabilidad a diferentes dispositivos.

1.9. Aportes

El Sistema web para la gestión de historias clínicas personalizables por especialidad será una herramienta que brinde información oportuna y confiable de manera móvil como de escritorio, que sirva de apoyo a la toma de decisiones sin importar donde se encuentre un paciente.

Se utilizará una arquitectura multinivel o programación por capas. En dichas arquitecturas a cada nivel se le confía una misión simple, lo que permite el diseño de arquitecturas escalables (que pueden ampliarse con facilidad en caso de que las necesidades aumenten).

Para el modelo del Software, se utilizarán elementos y diagramas de desarrollo UWE, basado en el modelo UML. Las tecnologías a usar en el presente trabajo serán PHP en su versión 7, con el framework de desarrollo CodeIgniter 3, base de datos PostgreSQL.

Se pretende también dar conocimiento y un empujón para la investigación y desarrollo de aplicaciones dinámicas dibujando los formularios desde la base de datos.

CAPÍTULO II

MARCO TEÓRICO

RESUMEN

En este capítulo, describiremos e introduciremos los principios y conceptos básicos para la realización del proyecto, sin embargo no se puede dar una teoría completa acerca de las metodologías, técnicas y herramientas que se utilizara, por el contrario se trata de presentar una base para la fácil comprensión de la misma.

2.1. Definición de Colegio Médico

Un colegio médico es una asociación profesional que reúne a los médicos de una jurisdicción administrativa concreta (un país, una región, una provincia). En los países anglosajones es frecuente que se agrupen por especialidades médicas (cardiólogos, médicos de cabecera, etc). (Wikipedia, 2019)

Los colegios de médicos actúan como salvaguarda de los valores fundamentales de la profesión médica: la deontología ¹y el código ético. Además de llevar la representación en exclusiva a nivel nacional e internacional de los médicos colegiados, tiene como función la ordenación y la defensa de la profesión médica. En la mayoría de los países la colegiación suele ser obligatoria. Un colegio profesional o colegio oficial es una corporación de derecho público de carácter gremial integrada por quienes ejercen las llamadas profesiones liberales y suelen estar amparados por el Estado. Sus miembros asociados son conocidos como colegiados. (Wikipedia, 2019)

2.2. Definición de establecimiento de salud

Los establecimientos de salud son instalaciones esenciales destinadas a proporcionar atención de salud con garantía de eficacia, eficiencia, oportunidad y calidez. La obligatoriedad de cumplir adecuadamente con la atención de los enfermos tiene connotaciones técnicas, administrativas, éticas y penales; exigencias que se mantienen en todos los momentos y circunstancias. (cidbimena, 2010)

Entiéndase por Establecimientos de Salud a los hospitales cualesquiera sea su tamaño o especialidad públicos o privados, policlínicos, Centros de Salud

¹ La deontología es una disciplina que se enfoca en estudiar el “deber ser” del hombre, es decir, su orientación moral y su desarrollo en este sentido.

Hospital, Puestos Médicos, Puestos Sanitarios, clínicas y consultorios privados y todo aquel en el que se realizan actividades concernientes a la salud de las personas. (Infoleyes, 2016)

La finalidad de los establecimientos de salud, es la de proveer a las personas servicios integrales que tiendan a prevenir las enfermedades, promover, de salud, reparar las enfermedades y rehabilitar a los impedidos, así como servir de centros de investigación y de enseñanza para la formación de los recursos humanos que requiere el país. (Infoleyes, 2016)

2.2.1. Categorización en establecimientos de salud Cabe resaltar que la vigencia de la categorización es por un plazo de tres años. Luego de este lapso, el establecimiento debe solicitar una recategorización. (conexionesan, 2018)

Tabla 1. Tipos de categorización de establecimientos de salud

Niveles de Atención	Perú	Panamá	Bolivia	México	Chile
I	P. S.	Puesto de Salud	Puestos de Salud	Promoción y atención ambulatoria	Postas Rurales (enfermeras)
	C. S.	Centros de Salud	Centros de Salud		Consultorios Generales urbanos y rurales
	C. S. Con Internam.	Centro Materno infantiles			
II	Hospital I	Hospital del área de salud	Hospitales Distritales	Hospitales Generales con especialidades básicas	Hospitales Básicos
	Hospital II				
III	Hospital III	Hospitales Regionales	Hospitales Regionales ó Departamentales	Hospitales de Alta Complejidad	Hospitales Generales
	Institutos Especializados	Hospitales Nacionales			
		Hospitales Especializados	Institutos		

Fuente: (SlideShare, 2012)

2.3. Expediente Clínico

2.3.1. Norma técnica del expediente clínico Según (Ministerio de Salud y Deportes, 2008) la Norma Técnica del Expediente Clínico (abreviada de ahora

en adelante como NTEC) es una norma elaborada el año 2008 por el Ministerio de Salud y Deportes, con el propósito de describir un lineamiento para el buen manejo de toda la documentación que constituye en un nexo entre un centro de salud y sus pacientes. Las normas presentadas son de cumplimiento obligatorio en todo el Sistema Nacional de Salud mediante sus actores que incluyen a los ejecutivos de los Entes Gestores como ser hospitales, policlínicos, centros de salud, postas, clínicas, etc.; y su personal como ser enfermería, trabajadores sociales, médicos, estadistas, archivistas, etc.

2.3.2. ¿Qué es un expediente clínico? La NTEC define al Expediente Clínico como el conjunto de documentos escritos e iconográficos evaluables que constituyen el historial clínico de una persona que ha recibido o recibe atención en un establecimiento de salud. Su manejo debe ser escrupuloso porque en él se encuentran todos los datos que nos permiten encarar de la mejor manera el estado de salud y enfermedad del paciente y su respectivo tratamiento. (Ministerio de Salud y Deportes, 2008)

2.3.3. Contenido del expediente clínico

2.3.3.1. Contenido general. El expediente clínico integra dos partes: Asistencial y Administrativa.

a) Asistencial. El contenido asistencial incluye todos los documentos referidos al proceso salud enfermedad de la persona, durante la consulta, hospitalización y seguimiento ambulatorio, avalados por quienes participan en su atención. (Ministerio de Salud y Deportes, 2008).

b) Administrativa. El contenido administrativo proporciona datos generales que permiten identificar en forma sencilla a cada paciente. Estos datos son: número del expediente clínico, fecha de ingreso, hora, nombre, ocupación, edad, fecha y

lugar de nacimiento, sexo, raza, lugar de procedencia, domicilio, ocupación, teléfono (fijo o móvil), datos de los padres, familiar o persona responsable, seguro medico (si lo tuviese), servicio o unidad de hospitalización y N° de cama ocupada. Además, incorpora documentos (formularios) administrativos. (Ministerio de Salud y Deportes, 2008).

2.3.3.2. Contenido específico. El contenido específico del expediente clínico se desagrega en documentos individuales, lo que otorga a cada uno de ellos, importancia y significación propia. Estos documentos son:

a) Historia clínica. Es el documento central del expediente clínico que además de señalar los datos generales del paciente y sus antecedentes personales, familiares, no patológicos, patológicos y gineco-obstétricos en el caso de la mujer; describe las condiciones actuales de su estado de salud-enfermedad, investigadas y recogidas a través de la anamnesis o interrogatorio y el examen físico general y especial. Concluye estableciendo el diagnóstico presuntivo, diagnósticos diferenciales y una propuesta básica de la conducta y tratamiento. El responsable para su elaboración, dentro de las 8 horas de transcurrida la hospitalización, es el médico tratante. En los hospitales de enseñanza, la Historia Clínica puede ser elaborada por delegación del médico tratante al médico residente de la unidad o servicio donde se hospitalizo al paciente, o por el estudiante de último año que se encuentra cumpliendo su Internado Rotatorio, bajo la supervisión y revisión del médico de planta, quien necesariamente dará su conformidad, estampando su nombre, sello y firma al pie del documento. (Ministerio de Salud y Deportes, 2008).

b) Informes de exámenes complementarios. Son todos aquellos formularios o documentos del expediente clínico que contienen la información de los exámenes o procedimientos complementarios de diagnóstico y tratamiento que hubiesen sido realizados en el paciente. Necesariamente deben consignar el nombre, firma y sello de los responsables de su ejecución y estar plasmados en formularios

expresamente diseñados para responder cabalmente a los requerimientos e investigación de quienes los solicitan. (Ministerio de Salud y Deportes, 2008).

c) Notas de evolución. Son aquellas notas escritas por diversos métodos, que describen cronológicamente el curso de la evolución del estado de salud-enfermedad del paciente durante su hospitalización o durante sus controles por consulta externa. Deben registrar los estados estacionarios o cambios que se presentan en el paciente, ya sea por la evolución clínica misma, por los resultados de la aplicación del tratamiento y procedimientos empleados, o por cualquier otra causa. Son responsables de su elaboración u omisión, el médico tratante y quienes coadyuvan con él en el tratamiento de su paciente. (Ministerio de Salud y Deportes, 2008).

d) Interconsultas. Son las valoraciones clínicas o procedimientos que hacen otros profesionales autorizados a solicitud del médico tratante, o de quienes coadyuvan con él en el tratamiento o atención de su paciente. Se registran como notas de evolución o en formularios expresamente diseñados para el efecto, especialmente para interconsultas interinstitucionales, siendo responsables de su ejecución y registro oportuno, él o los profesionales que atendieron la respectiva interconsulta, consignando su nombre, firma y sello. (Ministerio de Salud y Deportes, 2008).

e) Notas de informe profesional. Son los informes clínicos o de procedimientos diagnósticos o terapéuticos, escritos por los profesionales que estuvieron a cargo del paciente, antes, durante o después de su internación, en relación a su estado de salud-enfermedad, actual o pasado. Pueden emitirse de manera individual o como efecto de una junta médica y necesariamente deben consignar nombre, sello y firma de los profesionales que lo realicen. La petición de informe médico puede estar vinculada a motivos de interés particular como de origen legal o público. Por tanto, el valor de prueba y de garantía que el ordenamiento jurídico y

la sociedad confieren a los informes médicos, obliga a extremar el rigor de su contenido, evitando incluir en ellos términos ambiguos o informaciones insuficientes o excesivas que pueden confundir al destinatario. Entre los informes médicos se encuentra la epicrisis que es el documento emitido por el médico responsable al finalizar cada proceso asistencial de un paciente en un centro sanitario, y en el que se incluye, además de un breve resumen de la historia clínica, los datos más relevantes de la actividad asistencial prestada, y las correspondientes recomendaciones terapéuticas. (Ministerio de Salud y Deportes, 2008).

f) Notas de indicaciones médicas. Son aquellas notas agrupadas en una misma sección de expediente clínico, que contienen las indicaciones o prescripciones médicas, con descripción del medicamento en nombre genérico, dosis, concentración, frecuencia, vía de administración y cuidados de la misma. Son responsables de su elaboración u omisión, el médico tratante, o por delegación los profesionales médicos de planta o médicos residentes que lo colaboran para el manejo del paciente. (Ministerio de Salud y Deportes, 2008).

Existen alrededor de quince documentos individuales dentro de la clasificación del contenido específico que comprende un expediente clínico, pero para el desarrollo del sistema solo utilizaremos seis de los documentos individuales.

2.3.3.3. Contenido adicional. El expediente clínico también incorpora otros documentos adicionales que pueden o no existir, de acuerdo a circunstancias propias de cada caso. Dichos documentos son:

a) Certificado médico. Es el informe médico emitido en un formulario expreso del Colegio Médico de Bolivia, solicitado por el paciente o por su representante legal, a cualquiera de los profesionales médicos que lo hubiese tratado, como

efecto de su estado de salud-enfermedad actual o pasado. (Ministerio de Salud y Deportes, 2008).

b) Recetas. La receta o prescripción médica constituye el documento legal que avala la dispensación bajo prescripción facultativa, su fin es terapéutico y la emite el médico tratante o interconsultado, en un formulario institucional expreso, que debe ser llenado en letra claramente legible, consignando los siguientes datos: (Ministerio de Salud y Deportes, 2008).

- Nombre completo del paciente
- Nombre genérico del medicamento prescrito
- Vía y cuidados especiales para la administración
- Concentración y dosificación
- Frecuencia de administración en 24 horas
- Duración del tratamiento
- Precauciones especiales para la administración del producto o productos prescritos
- Indicaciones especiales para el paciente
- Datos de identificación del prescriptor (nombre completo, firma y matrícula profesional).

En la clasificación del contenido adicional existen 10 documentos individuales de los cuales solo veremos 2 para el desarrollo del sistema.

2.4. Ingeniería de requerimientos

El diseño y construcción de software de computadora es difícil, creativo y sencillamente divertido. En realidad, elaborar software es tan atractivo que muchos desarrolladores de software quieren ir directo a él antes de haber tenido el

entendimiento claro de lo que se necesita. Argumentan que las cosas se aclararán a medida que lo elaboren, que los participantes en el proyecto podrán comprender sus necesidades sólo después de estudiar las primeras iteraciones del software, que las cosas cambian tan rápido que cualquier intento de entender los requerimientos en detalle es una pérdida de tiempo, que las utilidades salen de la producción de un programa que funcione y que todo lo demás es secundario. Lo que hace que estos argumentos sean tan seductores es que tienen algunos elementos de verdad. Pero todos son erróneos y pueden llevar un proyecto de software al fracaso. (Pressman, 2010)

El espectro amplio de tareas y técnicas que llevan a entender los requerimientos se denomina ingeniería de requerimientos. Desde la perspectiva del proceso del software, la ingeniería de requerimientos es una de las acciones importantes de la ingeniería de software que comienza durante la actividad de comunicación y continúa en la de modelado. Debe adaptarse a las necesidades del proceso, del proyecto, del producto y de las personas que hacen el trabajo. (Pressman, 2010)

La ingeniería de requerimientos tiende un puente para el diseño y la construcción. Pero, ¿dónde se origina el puente? Podría argumentarse que principia en los pies de los participantes en el proyecto (por ejemplo, gerentes, clientes y usuarios), donde se definen las necesidades del negocio, se describen los escenarios de uso, se delinear las funciones y características y se identifican las restricciones del proyecto. Otros tal vez sugieran que empieza con una definición más amplia del sistema, donde el software no es más que un componente del dominio del sistema mayor. Pero sin importar el punto de arranque, el recorrido por el puente lo lleva a uno muy alto sobre el proyecto, lo que le permite examinar el contexto del trabajo de software que debe realizarse; las necesidades específicas que deben abordar el diseño y la construcción; las prioridades que guían el orden en el que se efectúa el trabajo, y la información, las

funciones y los comportamientos que tendrán un profundo efecto en el diseño resultante. (Pressman, 2010)

2.4.1. ¿Para qué un proceso de ingeniería de requerimientos? El proceso de ingeniería de requerimiento es un conjunto estructurado de actividades, mediante las cuales obtenemos, validamos y mantenemos el documento de especificación de requerimientos. (Pérez Huebe, 2005)

Las actividades del proceso incluyen la extracción de requerimientos, el análisis, la negociación y la validación. (Pérez Huebe, 2005)

No existe un proceso único que sea válido de aplicar en todas las organizaciones. Cada organización debe desarrollar su propio proceso de acuerdo al tipo de producto que se esté desarrollando, a la cultura organizacional, y al nivel de experiencia y habilidad de las personas involucradas en la ingeniería de requerimientos. (Pérez Huebe, 2005)

El proceso de recopilar, analizar y verificar las necesidades del cliente para un sistema llamado Ingeniería de Requerimientos. “La meta de la Ingeniería de Requerimientos es entregar una especificación de requisitos de software correcta y completa”. (Pérez Huebe, 2005)

2.4.2. Definición de ingeniería de requerimientos. Algunas definiciones para la Ingeniería de requerimientos según distintos autores.

- Según Boehm “La ingeniería de requerimientos es la disciplina para desarrollar una especificación completa, consistente y no ambigua, la cual servirá como base para acuerdos comunes entre todas las partes involucradas y en donde se describen las funciones que realizara el sistema” (Pérez Huebe, 2005)

- Starts argumenta que “La ingeniería de requerimientos es el proceso por el cual se transforman los requerimientos declarados por los clientes, ya sean hablados o escritos, a especificaciones precisas, no ambiguas, consistentes y completas del comportamiento del sistema, incluyendo funciones, interfaces, rendimiento y limitaciones” (Pérez Huebe, 2005)
- En un reporte tecnológico de ingeniería de software define a la ingeniería de requerimientos como “El proceso mediante el cual se intercambian deferentes puntos de vista para recopilar y modelar lo que el sistema va a realizar. Este proceso utiliza una combinación de métodos, herramientas y actores, cuyo producto es un modelo del cual se genera un documento de requerimientos” (Pérez Huebe, 2005)
- Una definición más general es la de Brain “Ingeniería de requerimientos es un enfoque sistémico para recolectar, organizar y documentar los requerimientos del sistema; es también el proceso que establece y mantiene acuerdos sobre los cambios de requerimientos, entre los clientes y el equipo del proyecto” (Pérez Huebe, 2005)

Dadas las definiciones anteriores, se aporta que la Ingeniería de Requerimientos es una sucesión de pasos que ayudan a obtener la definición clara, consistente y compacta de las especificaciones correctas que precisan el comportamiento de la información que requiere el usuario para su sistema. (Pérez Huebe, 2005)

2.4.3. Actividades de la ingeniería de requerimientos. La ingeniería de requerimientos proporciona el mecanismo apropiado para entender lo que desea el cliente, analizar las necesidades, evaluar la factibilidad, negociar una solución razonable, especificar la solución sin ambigüedades, validar la especificación y administrar los requerimientos a medida que se transforman en un sistema funcional. Incluye siete tareas diferentes: concepción, indagación, elaboración, negociación, especificación, validación y administración. Es importante notar que

algunas de estas tareas ocurren en paralelo y que todas se adaptan a las necesidades del proyecto. (Pressman, 2010)

1. **Concepción.** ¿Cómo inicia un proyecto de software? ¿Existe un solo evento que se convierte en el catalizador de un nuevo sistema o producto basado en computadora o la necesidad evoluciona en el tiempo? No hay respuestas definitivas a estas preguntas. (Pressman, 2010)
2. **Indagación.** En verdad que parece muy simple: preguntar al cliente, a los usuarios y a otras personas cuáles son los objetivos para el sistema o producto, qué es lo que va a lograrse, cómo se ajusta el sistema o producto a las necesidades del negocio y, finalmente, cómo va a usarse el sistema o producto en las operaciones cotidianas. Pero no es simple: es muy difícil. (Pressman, 2010)
3. **Elaboración.** La información obtenida del cliente durante la concepción e indagación se expande y refina durante la elaboración. Esta tarea se centra en desarrollar un modelo refinado de los requerimientos que identifique distintos aspectos de la función del software, su comportamiento e información. (Pressman, 2010)
4. **Negociación.** No es raro que los clientes y usuarios pidan más de lo que puede lograrse dado lo limitado de los recursos del negocio. También es relativamente común que distintos clientes o usuarios propongan requerimientos conflictivos con el argumento de que su versión es “esencial para nuestras necesidades especiales”. (Pressman, 2010)
5. **Especificación.** En el contexto de los sistemas basados en computadora (y software), el término especificación tiene diferentes significados para distintas personas. Una especificación puede ser un documento escrito, un conjunto de modelos gráficos, un modelo matemático formal, un conjunto de escenarios de uso, un prototipo o cualquier combinación de éstos. Algunos sugieren que para una especificación debe desarrollarse y utilizarse una “plantilla estándar”, con el argumento de que esto conduce a

requerimientos presentados en forma consistente y por ello más comprensible. (Pressman, 2010)

- 6. Validación.** La calidad de los productos del trabajo que se generan como consecuencia de la ingeniería de los requerimientos se evalúa durante el paso de validación. La validación de los requerimientos analiza la especificación a fin de garantizar que todos ellos han sido enunciados sin ambigüedades; que se detectaron y corrigieron las inconsistencias, las omisiones y los errores, y que los productos del trabajo se presentan conforme a los estándares establecidos para el proceso, el proyecto y el producto. (Pressman, 2010)
- 7. Administración de los requerimientos.** Los requerimientos para sistemas basados en computadora cambian, y el deseo de modificarlos persiste durante toda la vida del sistema. La administración de los requerimientos es el conjunto de actividades que ayudan al equipo del proyecto a identificar, controlar y dar seguimiento a los requerimientos y a sus cambios en cualquier momento del desarrollo del proyecto. (Pressman, 2010)

2.4.4. Recolección de datos. La recolección de datos se refiere al uso de una gran diversidad de técnicas y herramientas que pueden ser utilizadas por el analista para desarrollar los sistemas de información, los cuales pueden ser la entrevistas, la encuesta, el cuestionario, la observación, el diagrama de flujo y el diccionario de datos. (Shirley, 2012)

Todos estos instrumentos se aplicarán en un momento en particular, con la finalidad de buscar información que será útil a una investigación en común. (Shirley, 2012)

2.4.4.1. Entrevistas. Las entrevistas se utilizan para recabar información en forma verbal, a través de preguntas que propone el analista. Quienes responden pueden ser gerentes o empleados, los cuales son usuarios actuales del

sistema existente, usuarios potenciales del sistema propuesto o aquellos que proporcionarán datos o serán afectados por la aplicación propuesta. El analista puede entrevistar al personal en forma individual o en grupos algunos analistas prefieren este método a las otras técnicas que se estudiarán más adelante. Sin embargo, las entrevistas no siempre son la mejor fuente de datos de aplicación. (Shirley, 2012)

Dentro de una organización, la entrevista es la técnica más significativa y productiva de que dispone el analista para recabar datos. En otras palabras, la entrevista es un intercambio de información que se efectúa cara a cara. Es un canal de comunicación entre el analista y la organización; sirve para obtener información acerca de las necesidades y la manera de satisfacerlas, así como concejo y comprensión por parte del usuario para toda idea o método nuevos. Por otra parte, la entrevista ofrece al analista una excelente oportunidad para establecer una corriente de simpatía con el personal usuario, lo cual es fundamental en transcurso del estudio. (Shirley, 2012)

2.4.4.2. Cuestionarios. Los cuestionarios proporcionan una alternativa muy útil para la entrevista; sin embargo, existen ciertas características que pueden ser apropiada en algunas situaciones e inapropiadas en otra. Al igual que la entrevistas, deben diseñarse cuidadosamente para una máxima efectividad. (Shirley, 2012)

2.4.4.3. Encuesta. La encuesta es un instrumento de investigación para obtener información representativa de un grupo de personas. Se trata de aplicar un cuestionario a determinado número de individuos, con el objeto de obtener un resultado. El requisito es que debe aplicarse a un número representativo. (Shirley, 2012)

2.4.4.4. La observación. Otra técnica útil para el analista en su progreso de investigación, consiste en observar a las personas cuando efectúan su trabajo. Como técnica de investigación, la observación tiene amplia aceptación científica. Los sociólogos, psicólogos e ingenieros industriales utilizan extensamente esta técnica con el fin de estudiar a las personas en sus actividades de grupo y como miembros de la organización. (Shirley, 2012)

El propósito de la organización es múltiple: permite al analista determinar que se está haciendo, como se está haciendo, quien lo hace, cuando se lleva a cabo, cuánto tiempo toma, dónde se hace y por qué se hace. (Shirley, 2012)

2.4.4.5. Diagramas de Flujo. Es una representación pictórica de los pasos en proceso. Útil para determinar cómo funciona realmente el proceso para producir un resultado. El resultado puede ser un producto, un servicio, información o una combinación de los tres. Al examinar cómo los diferentes pasos en un proceso se relacionan entre sí, se puede descubrir con frecuencia las fuentes de problemas potenciales. (Shirley, 2012)

Los diagramas de flujo se pueden aplicar a cualquier aspecto del proceso desde el flujo de materiales hasta los pasos para hacer la venta u ofrecer un producto. Con frecuencia este nivel de detalle no es necesario, pero cuando se necesita, el equipo completo de trabajos más pequeños puede agregar niveles según sea necesario durante el proyecto. (Shirley, 2012)

2.4.5. Características de los requerimientos. Las características de los requerimientos son sus propiedades principales. Un conjunto de requerimientos en estado de madurez, deben presentar una serie de atributos tanto individualmente como en grupo. Las más importantes son las siguientes: (Pérez Huebe, 2005)

- **Necesario:** Si su omisión provoca una deficiencia en el sistema a construir, y además su capacidad, características físicas o factor de calidad no pueden ser reemplazados por otras capacidades del producto o del proceso.
- **Conciso:** si es fácil de leer y entender. Su redacción debe ser simple y clara para aquellos que vayan a consultarlos en un futuro.
- **Completo:** si no necesita ampliar detalles en su redacción, es decir, si se proporciona la información suficiente para su comprensión.
- **Consistente:** si no es contradictorio con otro requerimiento.
- **No ambiguo:** cuando tiene una sola interpretación. El lenguaje usado en su definición no debe causar confusiones al lector.

2.4.6. Importancia de la ingeniería de requerimientos. Los principales beneficios que se obtienen de la Ingeniería de Requerimientos son:

- Permite gestionar las necesidades del proyecto en forma estructurada: Cada actividad de la IR² consiste de una serie de pasos organizados y bien definidos. (Pérez Huebe, 2005)
- Mejora la capacidad de predecir cronogramas de proyectos, así como sus resultados: La IR proporciona un punto de partida para controles subsecuentes y actividades de mantenimiento, tales como estimación de costos, tiempo y recursos necesarios. (Pérez Huebe, 2005)
- Disminuye los costos y retrasos del proyecto: es sabido que reparar errores por un mal desarrollo no descubierto a tiempo, es sumamente caro; especialmente aquellas decisiones tomadas durante la IR, ya que es una de las etapas de mayor importancia en el ciclo de desarrollo de software y de las primeras en llevarse a cabo. (Pérez Huebe, 2005)

² Ingeniería de Requerimientos

- Mejora la calidad del software: La calidad en el software tiene que ver con cumplir un conjunto de requerimientos (funcionalidad, facilidad de uso, confiabilidad, desempeño, etc.). (Pérez Huebe, 2005)
- Mejora la comunicación entre equipos: La especificación de requerimientos representa una forma de consenso entre clientes y desarrolladores. Si este consenso no ocurre, el proyecto no será exitoso. (Pérez Huebe, 2005)
- Evita rechazos de usuarios finales: La ingeniería de requerimientos obliga al cliente a considerar sus requerimientos cuidadosamente y revisarlos dentro del marco del problema, por lo que se le involucra durante todo el desarrollo del proyecto. (Pérez Huebe, 2005)

2.4.7. Requerimientos funcionales y no funcionales. A menudo, los requerimientos de sistemas de software se clasifican en funcionales y no funcionales, o como requerimientos de dominio: (Sommerville, 2005)

1. *Requerimientos funcionales.* Son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. En algunos casos, los requerimientos funcionales de los sistemas también pueden declarar explícitamente lo que el sistema no debe hacer. (Sommerville, 2005)
2. *Requerimientos no funcionales.* Son restricción de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. Los requerimientos no funcionales a menudo se aplican al sistema en su totalidad. Normalmente apenas se aplican a características o servicios individuales del sistema. (Sommerville, 2005)
3. *Requerimientos del dominio.* Son requerimientos que provienen del dominio de aplicación del sistema y que refleja las características y restricciones de ese dominio. Pueden ser funcionales o no funcionales. (Sommerville, 2005)

En realidad, la distinción entre diferentes tipos de requerimientos no es tan clara como sugieren estas definiciones. Por ejemplo, un requerimiento del usuario sobre seguridad podría parecer un requerimiento no funcional. Sin embargo, cuando se desarrolla en detalle, puede generar otros requerimientos que son claramente funcionales, como la necesidad de incluir en el sistema recursos para la autenticación del usuario. (Sommerville, 2005)

2.4.7.1. Requerimientos funcionales. Los requerimientos funcionales de un sistema describen lo que el sistema debe hacer. Estos requerimientos dependen del tipo del software que se desarrolle, de los posibles usuarios del software y del enfoque general tomado por la organización al redactar requerimientos. Cuando se expresan como requerimientos del usuario, habitualmente se describen de una forma bastante abstracta. Sin embargo, los requerimientos funcionales del sistema describen con detalle la función de este, sus entradas y salidas, excepciones, etcétera. Los requerimientos funcionales para un sistema de software se pueden expresar de diferentes formas. (Sommerville, 2005)

La impresión en la especificación de requerimientos es la causa de muchos de los problemas de la ingeniería de software. Para un desarrollador de sistemas es natural dar interpretaciones de un requerimiento ambiguo con el fin de simplificar su implementación. Sin embargo, a menudo no es lo que el cliente desea. Se deben establecer nuevos requerimientos y hacer cambios en el sistema. Por su puesto, esto retrasa la entrega de este e incrementa los costes. (Sommerville, 2005)

En principio, la especificación de requerimientos funcionales de un sistema debe estar completa y ser consistente. La completitud significa que todos los servicios solicitados por el usuario deben estar definidos. La consistencia significa que los requerimientos no deben tener definiciones contradictorias. En la práctica,

para sistemas grandes y complejos, es prácticamente imposible alcanzar los requerimientos de consistencia y completitud. (Sommerville, 2005)

Una razón de esto es que es fácil cometer errores y omisiones cuando se redactan especificaciones para sistemas grandes y complejos. Otra razón es que los stakeholders³ del sistema tienen necesidades diferentes, y a menudo contradictorias. Estas contradicciones pueden no ser obvias cuando los requerimientos se especifican por primera vez, por lo que se incluyen requerimientos contradictorios en la especificación. Es posible que los problemas surjan solamente después de un análisis más profundo o, a veces, después de que se termine el desarrollo y el sistema se entregue al cliente. (Sommerville, 2005)

Tabla 2. Ejemplo tabla de requerimientos funcionales

<i>Id</i>	<i>Requerimiento</i>	<i>Descripción</i>	<i>Prioridad</i>
RF1	Requerimiento RF1	Descripción RF1	Prioridad RF1
RF2	Requerimiento RF2	Descripción RF2	Prioridad RF2
RF3	Requerimiento RF3	Descripción RF3	Prioridad RF2
.	.	.	.
.	.	.	.
.	.	.	.
RFN	Requerimiento RFN	Descripción RFN	Prioridad RFN

Elaboración: Propia

2.4.7.2. Requerimientos no funcionales. Los requerimientos no funcionales, como su nombre sugiere, son aquellos requerimientos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de este como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento. De forma alternativa, definen las restricciones del sistema como la capacidad de los dispositivos de entrada/salida y las

³ Stakeholder es una palabra del inglés que, en el ámbito empresarial, significa ‘interesado’ o ‘parte interesada’, y que se refiere a todas aquellas personas u organizaciones afectadas por las actividades y las decisiones de una empresa.

representaciones de datos que se utilizan en las interfaces del sistema. (Sommerville, 2005)

Los requerimientos no funcionales rara vez se asocian con características particulares del sistema. Más bien, estos requerimientos especifican o restringen las propiedades emergentes del sistema. Por lo tanto, pueden especificar el rendimiento del sistema, la protección, la disponibilidad, y otras propiedades emergentes. Esto significa que a menudo son más críticos que los requerimientos funcionales particulares. Los usuarios del sistema normalmente pueden concentrar formas de trabajar alrededor de una función del sistema que realmente no cumple sus necesidades. Sin embargo, el incumplimiento de un requerimiento no funcional puede significar que el sistema entero sea inutilizable. Por ejemplo, si un sistema de vuelo no cumple sus requerimientos de fiabilidad, no se certificará como seguro para el funcionamiento; si un sistema de control de tiempo real no cumple sus requerimientos de rendimiento, las funciones de control no funcionarían correctamente. (Sommerville, 2005)

Los requerimientos no funcionales no solo se refieren al sistema software a desarrollar. Algunos de estos requerimientos pueden restringir el proceso que se debe utilizar para desarrollar el sistema. Ejemplo de requerimientos de procesos son la especificación de los estándares de calidad que se deben utilizar en el proceso, una especificación que el diseño debe producir con una herramienta CASE particular y una descripción del proceso a seguir. (Sommerville, 2005)

Los requerimientos no funcionales surgen de las necesidades del usuario, debido a las restricciones en el presupuesto, a las políticas de la organización, a la necesidad de interoperabilidad con otros sistemas software o hardware, o a factores externos como regulaciones de seguridad o legislaciones sobre privacidad. (Sommerville, 2005)

Tabla 3. Ejemplo de tabla de requerimientos no funcionales

<i>Id</i>	<i>Propiedad</i>	<i>Medida</i>
RNF1	Propiedad RNF1	Medida RNF1
RNF2	Propiedad RNF2	Medida RNF2
RNF3	Propiedad RNF3	Medida RNF3
.	.	.
.	.	.
.	.	.
RNFN	Propiedad RNFN	Medida RNFN

Elaboración: Propia

2.5. Ingeniería de software

La ingeniería de software es una disciplina de la ingeniería que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de éste después de que se utiliza. En esta definición, existen dos frases claves: (Sommerville, 2005)

1. *Disciplina de la ingeniería.* Los ingenieros hacen que las cosas funcionen. Aplican teorías, métodos y herramientas donde sean convenientes, pero las utilizan de forma selectiva y siempre tratando de descubrir soluciones a los problemas, aun cuando no existan teorías y métodos aplicables para resolverlos. Los ingenieros también saben que deben trabajar con restricciones financieras y organizacionales, por lo que buscan soluciones tomando en cuenta estas restricciones. (Sommerville, 2005)
2. *Todos los aspectos de producción de software.* La ingeniería del software no sólo comprende los procesos técnicos del desarrollo de software, sino también con actividades tales como la gestión de proyectos de software y desarrollo de herramientas, métodos y teorías de apoyo a la producción de software. (Sommerville, 2005)

En general, los ingenieros de software adoptan un enfoque sistemático y organizado en su trabajo, ya que es la forma más efectiva de producir software de

alta calidad. Sin embargo, aunque la ingeniería consiste en seleccionar el método más apropiado para un conjunto de circunstancias, un enfoque más informal y creativo de desarrollo podría ser efectivo en algunas circunstancias. El desarrollo informal es apropiado para el desarrollo de sistemas basados en Web, los cuales requieren una mezcla de técnicas de software y diseño gráfico. (Sommerville, 2005)

2.5.1. Metodología de desarrollo de software. Lograr la construcción de un sistema informático eficiente, que cumpla con los requerimientos planteados, es una tarea realmente intensa y sobre todo difícil de cumplir. Las metodologías para el desarrollo del software imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Una metodología de desarrollo de software tiene como principal objetivo aumentar la calidad del software que se produce en todas y cada una de sus fases de desarrollo. No existe una metodología de software universal, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigiéndose así que el proceso sea configurable. Las metodologías de desarrollo se pueden dividir en dos grupos de acuerdo con sus características y los objetivos que persiguen: ágiles y robustas o tradicionales. (EcuRed, 2011)

2.5.1.1. Metodologías tradicionales. Están formadas por un conjunto de fases o actividades en las que no tienen en cuenta la naturaleza evolutiva del software, las cuales son:

- **Ciclo de vida.** También se le conoce como modelo lineal secuencia o modelo en cascada. Plantea un enfoque sistemático, secuencial para el desarrollo de software, que comienza en un nivel de sistemas y continúa con el análisis, diseño, codificación, pruebas y mantenimiento. Este modelo comprende una primera actividad como lo es La Ingeniería y modelado de sistemas de información, en el cual se establece el sistema de nivel

superior y se deben establecer los requisitos de la empresa en la que se encuentra. Los requisitos se recogen del sistema con una pequeña parte de análisis y diseño. (Huanca Cantuta, 2015)

- **Basado en Prototipos.** Este paradigma se inicia con la recolección de requerimientos. El desarrollador y el cliente encuentran y definen los objetivos globales para el software, identifican los requisitos conocidos y las áreas del esquema en donde es obligatoria más definición. Luego aparece un “diseño rápido”. El diseño rápido está centrado en una representación de los aspectos del software que serán visibles para el usuario-cliente. El diseño rápido lleva a la construcción de un prototipo. El prototipo lo evalúa el cliente-usuario y lo utiliza para refinar los requisitos del software a desarrollar. La interacción ocurre cuando el prototipo satisface las necesidades del cliente, a la vez que permite que el desarrollador comprenda mejor lo que se necesita hacer. (Huanca Cantuta, 2015)
- **Modelo DRA.** El Desarrollo Rápido de Aplicaciones (DRA) es un modelo de proceso del desarrollo del software lineal secuencial que enfatiza un ciclo de desarrollo extremadamente corto. El modelo DRA es una adaptación a “alta velocidad” del modelo lineal secuencial en el que se logra el desarrollo rápido utilizando un enfoque de construcción basado en componentes. (Huanca Cantuta, 2015)

Están guiadas por una fuerte planificación. Centran su atención en llevar una documentación exhaustiva de todo el proceso de desarrollo y en cumplir con un plan de proyecto, definido en la fase inicial del mismo. Entre las metodologías robustas se encuentran: MSF (por sus siglas en inglés Microsoft Solution Framework), MÉTRICA 3 y RUP (siglas de Rational Unified Process). (EcuRed, 2011)

2.5.1.2. Metodologías ágiles. Un proceso es ágil cuando el desarrollo de software es incremental (entregas pequeñas de software, con ciclos rápidos),

cooperativo (cliente y desarrolladores trabajan juntos constantemente con una cercana comunicación), sencillo (el método en sí mismo es fácil de aprender y modificar, bien documentado), y adaptable (permite realizar cambios de último momento). (Huanca Cantuta, 2015)

Se caracterizan por hacer énfasis en la comunicación cara a cara, es decir, se basan en una fuerte y constante interacción, donde clientes desarrolladores y desarrolladores trabajan constantemente juntos, estableciéndose así una estrecha comunicación. Estas metodologías están orientadas al resultado del producto y no a la documentación; exige que el proceso sea adaptable, permitiendo realizar cambios de último momento. Se puede hacer mención dentro de las metodologías ágiles a: XP (por sus siglas en inglés Extreme Programming), Scrum y Crystal Methodologies. (EcuRed, 2011)

2.5.1.3. Diferencia entre metodologías tradicionales y ágiles. La siguiente tabla muestra aspectos relevantes de las metodologías de desarrollo tradicional contrastándolas con los aspectos relevantes de las metodologías de desarrollo ágil.

Tabla 4. Comparativa metodología Tradicional y ágil

<i>Metodologías Tradicionales</i>	<i>Metodologías Ágiles</i>
Desarrollo lento debido al estricto cumplimiento de los procesos.	Desarrollo rápido gracias al cumplimiento de las características del manifiesto ágil.
El equipo de trabajo sigue una estricta serie de procesos sin posibilidad de saltarse hacia otros roles y responsabilidades.	El equipo de trabajo puede intercambiar roles en cualquier momento y transferir conocimiento de manera colectiva.
Grupos pequeños y trabajando en el mismo sitio.	Grupos grandes y distribuidos.
El cliente interactúa con el equipo de desarrollo pocas veces	El cliente es parte del equipo de desarrollo constantemente.
El contrato es totalmente fijo e inmodificable	El contrato puede estar sujeto a cambios a lo largo del desarrollo.

Fuente: (González Alcolado & Crespo Obaco, 2018)

2.6. Metodología SCRUM

En el año 1986 Takeuchi y Nonaka publicaron el artículo “The New Product Development Game” el cual daría a conocer una nueva forma de gestionar proyectos en la que la agilidad, flexibilidad, y la incertidumbre son los elementos principales. (Trigas Gallego, 2012)

Nonaka y Takeuchi se fijaron en empresas tecnológicas que, estando en el mismo entorno en el que se encontraban otras empresas, realizaban productos en menos tiempo, de buena calidad y menos costes. (Trigas Gallego, 2012)

Observando a empresas como Honda, HP, Canon... etc., se dieron cuenta de que el producto no seguía unas fases en las que había un equipo especializado en cada una de ellas, sino que se partía de unos requisitos muy generales y el producto lo realizaba un equipo multidisciplinario que trabajaba desde el comienzo del proyecto hasta el final.

Se comparó esta forma de trabajo en equipo, con la colaboración que hacen los jugadores de Rugby y la utilización de una formación denominada **SCRUM**. (Trigas Gallego, 2012)

En 1996, Jeff Sutherland y Ken Schwaber presentaron las prácticas que se usaban como proceso formal para el desarrollo de software y que pasarían a incluirse en la lista Agile Alliance. (Trigas Gallego, 2012)

2.6.1. ¿Qué es Scrum? Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos. (ProyectosAgiles.org)

En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales. (ProyectosAgiles.org)

Scrum también se utiliza para resolver situaciones en que no se está entregando al cliente lo que necesita, cuando las entregas se alargan demasiado, los costes se disparan o la calidad no es aceptable, cuando se necesita capacidad de reacción ante la competencia, cuando la moral de los equipos es baja y la rotación alta, cuando es necesario identificar y solucionar ineficiencias sistemáticamente o cuando se quiere trabajar utilizando un proceso especializado en el desarrollo de producto. (ProyectosAgiles.org)

La estructura está compuesta de Equipos Scrum que llevan a cabo una serie de funciones, tienen diferentes utensilios y eventos y siguen una serie de reglas. Cada parte de la estructura tiene un propósito, que trabaja hacia el éxito de Scrum. (Goncalves, 2019)

2.6.2. Teoría scrum. Para entender Scrum, es importante echar un vistazo a sus fundamentos básicos. Scrum se ha fundado sobre una teoría empírica de control de procesos, que también se conoce como empirismo. Lo que afirma esta teoría es que el conocimiento se basa en la toma de decisiones y en la experiencia de los factores conocidos. (Goncalves, 2019)

Por tanto, Scrum busca cómo optimizar la predictibilidad y controlar el riesgo utilizando un método Iterativo e Incremental. Para que esto suceda, hay tres pilares que se deben implementar. Estos son la Transparencia, la Inspección y la Adaptación. (Goncalves, 2019)

1. *Transparencia.* Hay partes de este proceso que necesitan ser visibles para aquellos responsables de los resultados. Esto requiere definiciones estándar de todos los aspectos para asegurarse de que hay un entendimiento común de todas las observaciones. Considera estos ejemplos: (Goncalves, 2019)
 - Todos los participantes en el proceso deben usar un lenguaje común.
 - Tanto aquellos que llevan a cabo el trabajo, como los que aceptan el resultado, deben tener una definición estándar de “Finalizado”.
2. *Inspección.* Aquellos que usan Scrum necesitan inspeccionar periódicamente los instrumentos Scrum y el progreso del Objetivo en el Sprint. Esto asegura que puedan detectar variaciones no deseadas a tiempo. Las inspecciones deben llevarse a cabo de vez en cuando, de manera que no se interrumpa el trabajo en progreso. Los inspectores cualificados se aseguran de que todas las inspecciones se hacen de manera correcta y son beneficiosas. (Goncalves, 2019)
3. *Adaptación.* Después de una inspección, puede revelarse que uno o más aspectos del proceso se ha desviado de los límites aceptables. Esto significa que el producto final podría no ser aceptable y que es necesario hacer algún ajuste en el proceso o el material que se está usando. Cuanto antes ocurra, mejor, de manera que se eviten más desviaciones. (Goncalves, 2019)

2.6.3. Ciclo de vida de scrum. A Scrum se incorporó a las siguientes actividades: Requisitos, análisis, diseño, evolución y entrega.

Cuando se comienza cualquier proyecto nuevo, se comienza la discusión sobre los requisitos, luego se analiza, después se diseña y finalmente se entrega el producto. Pero todas estas reuniones son útiles para la administración de nivel superior, ya que se necesita trabajar con el equipo central. Scrum consta de tres

fases: Pre-juego (Planificación + Arquitectura), en algunos otros textos como Planificación, Juego (reunión Sprint + Scrum) “Development” o Desarrollo y Post-juego (Demo + cierre) o Finalización. (Chimarro Chipantiza, Mazon Olivo, & Cartuche Calva, 2015)

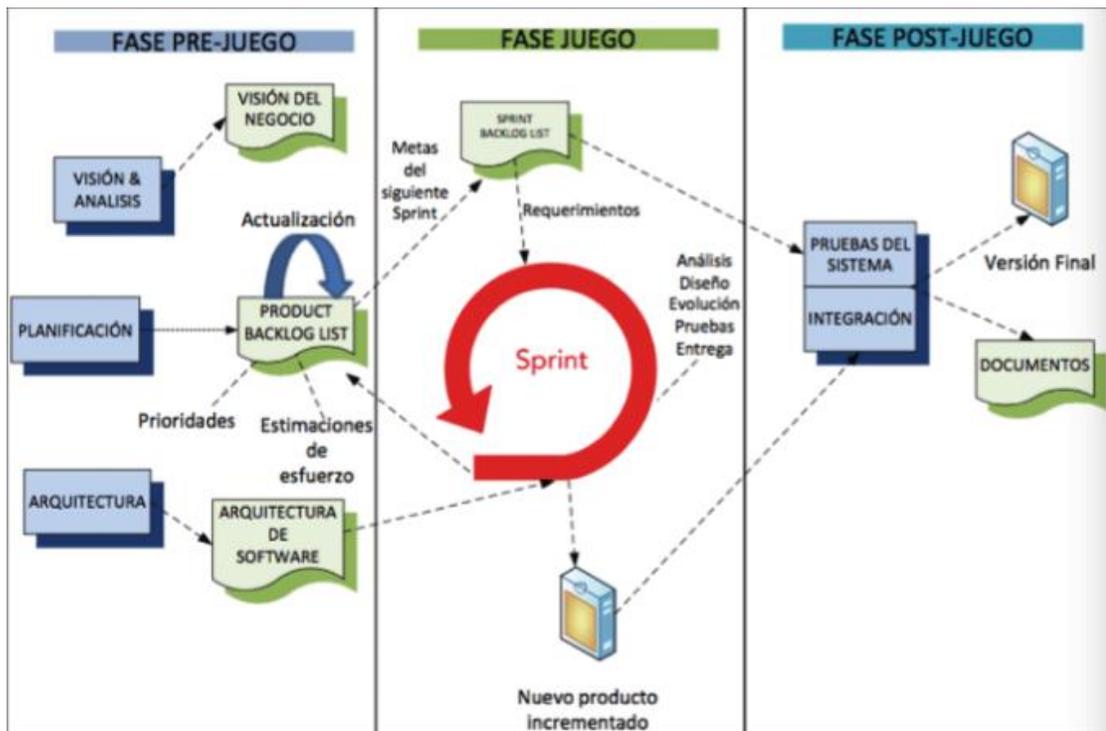


Figura 3. Ciclo de vida de Scrum.

Fuente: (Chimarro Chipantiza, Mazon Olivo, & Cartuche Calva, 2015)

2.6.3.1. Antes del desarrollo (pregame). La fase de Pregame incluye dos subfases los cuales son: Planning y Arquitectura.

1. *Planning.* Consiste en la definición del sistema que será construido. Para esto se crea la lista Product Backlog a partir del conocimiento que actualmente se tiene del sistema. En ella se expresan los requerimientos priorizados y a partir de ella se estima el esfuerzo requerido. La Product Backlog List es actualizada constantemente con ítems nuevos y más detallados, con estimaciones más precisas y cambios en la prioridad de los ítems. (Peralta, 2003)

2. *Architecture/High level design*. El diseño de alto nivel del sistema se planifica a partir de los elementos existentes en la Product Backlog List. En caso de que el producto a construir sea una mejora a un sistema ya existente, se identifican los cambios necesarios para implementar los elementos que aparecen en la lista Product Backlog y el impacto que pueden tener estos cambios. Se sostiene una Design Review Meeting para examinar los objetivos de la implementación y tomar decisiones a partir de la revisión. Se preparan planes preliminares sobre el contenido de cada release. (Peralta, 2003)

Entrada: La concepción inicial del producto que tienen los accionistas o interesados. **Tareas:** Las tareas y asignación responsabilidades de la fase de pregame son descritas en la siguiente tabla:

Tabla 5. Tareas de la fase Pregame

Nombre de la tarea	Descripción	Responsables
Crear la Product Backlog List y controlar su consistencia	Posibles elementos de esta lista son requerimientos técnicos y del negocio, funciones, errores a reparar, defectos, mejoras y actualizaciones tecnológicas requeridas. Es importante controlar la consistencia de la lista. Para esto se agregan, modifican, eliminan, especifican y priorizan sus elementos	Product Owner
Priorizar la Product Backlog List	Esta actividad se basa en considerar que elementos tienen más o menos influencia en el éxito del proyecto en un momento dado; considerando que los elementos con mayor prioridad se realizan primero.	Product Owner
Effort Estimation	Es un proceso iterativo que reúne toda la información que haya acerca un elemento para tener un mayor nivel de precisión en la estimación. Siempre se mide el esfuerzo que falta para cumplir con el / los objetivos tanto a nivel de la lista Product Backlog como para el Sprint Backlog (lo que resta).	Product Owner Scrum Team
Design Review Meeting	En esta instancia se comunica el diseño a los interesados para revisar el cumplimiento de los ítems especificados en el Product Backlog	

Fuente: (Peralta, 2003)

Verificación: Deben estar realizadas todas las tareas requeridas.

Salida: Product Backlog List, Arquitectura.

2.6.3.2. Fase del desarrollo (development phase). La fase de Development también llamada Game Phase es la parte ágil del Scrum. En esta fase se espera que ocurran cosas impredecibles. Para evitar el caos Scrum define prácticas para observar y controlar las variables técnicas y del entorno, así también como la metodología de desarrollo que hayan sido identificadas y puedan cambiar. Este control se realiza durante los Sprints. Dentro de variables de entorno encontramos: tiempo, calidad, requerimientos, recursos, tecnologías y herramientas de implementación. (Peralta, 2003)

Una vez realizada la especificación correspondiente, se lleva a cabo la elaboración del proyecto con un continuo seguimiento a cargo del mismo grupo de desarrollo. En cada iteración del Game se realizarán las siguientes tareas:

- 1. Planeación del Sprint:** Antes de comenzar cada Sprint o Iteración, se lleva a cabo dos reuniones consecutivas, en la primera se mejora y se prioriza nuevamente el backlog del producto, además de elegir las metas de la iteración. En la segunda reunión se deben considerar el cómo alcanzar los requerimientos y crear el Backlog del Sprint. (Yujra Choquetopa, 2018)
- 2. Desarrollo del Sprint.** El trabajo generalmente se organiza en iteraciones de 30 días (Sprints). El Sprint es el desarrollo de la nueva funcionalidad para el producto. Esta fase provee la siguiente documentación. Backlog del Sprint con las actividades desarrolladas, los responsables y la duración de cada actividad. (Yujra Choquetopa, 2018)
- 3. Revisión del Sprint.** Al final de cada iteración se lleva a cabo una reunión de revisión, donde se presenta la nueva funcionalidad del software, las metas e inclusive la información de las funciones, diseño, ventajas, inconvenientes y esfuerzos del equipo. (Yujra Choquetopa, 2018)

Entrada: Product Backlog List

Tareas:

Tabla 6. Tareas de la fase de desarrollo

Nombre de la tarea	Descripción	Responsables
Sprint Planning Meeting	Es una reunión organizada por el Scrum Master, que se realiza en dos fases. La primera fase tiene como objetivo establecer que ítems de la Product Backlog List van a ser realizados durante el Sprint.	Scrum Master Customer, User Management Product Owner Scrum Team
	En la segunda fase se decide como se van a alcanzar los objetivos del Sprint. En esta fase se crea la Sprint Backlog, indicando qué tareas debe desempeñar el equipo para cumplir con dichos objetivos.	Scrum Team Scrum Master Product Owner
Daily Scrum Meeting	Las reuniones se realizan en el mismo lugar y a la misma hora cada día. Idealmente en la mañana para definir el trabajo para el día. Tienen una duración de 15 minutos y los participantes se quedan parados. Estas reuniones no se utilizan para resolver problemas. En ellas se realizan tres preguntas: ¿Qué hiciste ayer?, ¿Qué harás hoy? y ¿Qué obstáculos ves en tu camino? Los participantes son clasificados según el compromiso que tengan con las actividades del proyecto en dos categorías: gallinas y chanchos Los chanchos son los que están más comprometidos y por lo tanto son los que pueden hablar y brindar opiniones. Esto ayuda a evitar reuniones innecesarias.	Scrum Team
Sprint Review Meeting	Es una reunión informal que tiene como regla que su preparación no puede tomar más de 2 horas. En ella el equipo presenta lo que ha logrado durante el Sprint. Generalmente toma la forma de una demo de las nuevas características o la arquitectura.	Customers Management Product Owner Otros interesados

Fuente: (Peralta, 2003)

Verificación: Durante un sprint se puede acortar funcionalidad, pero la fecha de entrega debe ser respetada.

Salida: Incremento del producto

2.6.3.3. Después del desarrollo (postgame). La fase PostGame contiene el cierre del release. Para ingresar a esta fase se debe llegar a un acuerdo respecto a las variables del entorno por ejemplo que los requerimientos fueron

completados. El sistema está listo para ser liberado y es en esta etapa en la que se realiza integración, pruebas del sistema y documentación. (Peralta, 2003)

Demostración. Se debe entregar el incremento de software al cliente para que el propietario del producto y la parte interesada puedan demostrar y evaluar la funcionalidad que se ha implementado.

Cierre. Se realiza la preparación para el lanzamiento, incluida la documentación final, las pruebas preparatorias previas al lanzamiento y el lanzamiento.

2.6.4. Roles y responsabilidades de scrum. Los roles se dividen en 2 grupos: cerdos y gallinas, esto surge en el chiste sobre un cerdo y una gallina y su intención de poner un restaurante. (Trigas Gallego, 2012)



Figura 4. Ejemplo de distribución de roles en scrum.

Fuente: (Trigas Gallego, 2012)

2.6.4.1. Los cerdos. Son personas que están comprometidas con el proyecto y el proceso de Scrum. (Trigas Gallego, 2012)

Tres miembros componen el Equipo Scrum básico, y son el Product Owner, el Equipo de Desarrollo y el Scrum Master. Se espera que estos equipos se auto-organicen y sean multifuncionales. Cuando se auto-organizan, pueden elegir la mejor manera de finalizar el trabajo y no tener en cuenta la orientación que puedan dar personas que no sean del equipo. (Goncalves, 2019)

Como equipos multifuncionales, tienen todas las competencias para realizar el trabajo, sin depender de otras personas fuera del equipo. El objetivo del equipo es optimizar la flexibilidad, la creatividad y la productividad. (Goncalves, 2019)

Los Equipos Scrum entregan productos de manera iterativa e incremental, aprovechando el feedback que les llega. Las entregas incrementales de productos “Finalizados” aseguran que siempre está disponible una versión funcional del producto. Si quieres conseguir un equipo óptimo, echa un vistazo a estos dos artículos: “Cómo Conseguir un Gran Equipo Ágil” y “Cómo Llevar a Cabo una Reunión Inicial de Equipo”. (Goncalves, 2019)

a) Product Owner. Cuando se trata de maximizar el valor del producto y el trabajo del Equipo de Desarrollo, es el Product Owner el responsable de ello. Esto varía según los Equipos Scrum y las personas del equipo. El Product Owner tiene la responsabilidad de gestionar el Backlog del Producto. Esta gestión incluye: (Goncalves, 2019)

- Expresar claramente los elementos del backlog de Producto
- Ordenar los elementos del Backlog de Producto para alcanzar las misiones y los objetivos
- Optimizar el valor del trabajo que realiza el Equipo de Desarrollo
- Asegurar que el Backlog de Producto es visible, transparente y claro.

Esto revela en qué debería trabajar el Equipo Scrum y asegura que el Equipo de Desarrollo comprende perfectamente los elementos del Backlog de Producto al nivel requerido. Aunque el Product Owner podría delegar este trabajo en el Equipo de Desarrollo, son responsables del resultado. (Goncalves, 2019)

El Product Owner es una persona y no un comité. Si existe un comité en funcionamiento, pueden presentar sus deseos en el backlog de Producto. Aquellos

que quieran hacer cualquier ajuste, necesitan consultar al Product Owner. (Goncalves, 2019)

Para garantizar que el Product Owner tiene éxito, todas las personas de la organización deben respetar sus decisiones, que son visibles en el contenido y orden del Backlog de Producto. No hay nadie capaz de ordenar al Equipo de Desarrollo trabajar con requisitos distintos, y el Equipo de Desarrollo también tiene sus acciones limitadas bajo las instrucciones de otra persona. (Goncalves, 2019)

b) Scrum Master. El Scrum Master tiene la responsabilidad de asegurar que se ha entendido y aprobado el método Scrum. Trabajan con el Equipo Scrum, por lo que pueden adherirse a la teoría, prácticas y reglas de Scrum. El Scrum Master es esencialmente el líder-ayudante del Equipo Scrum. Si te encuentras en la situación de que eres el Scrum Master de dos equipos, echa un vistazo a: “Las Dos Preguntas más Importantes sobre Liderar 2 Equipos como Scrum Master”. (Goncalves, 2019)

El Scrum Master ayuda a las personas que no están en el Equipo Scrum a entender cuáles de sus interacciones con el Equipo Scrum son útiles y cuáles no. (Goncalves, 2019)

c) Equipo de Desarrollo. Es un equipo formado por profesionales que trabajan para entregar un Incremento de producto “Finalizado” que se pueda lanzar al final de cada Sprint. Solo los miembros de este equipo pueden crear el Incremento. (Goncalves, 2019)

La organización asegura que el Equipo de Desarrollo está empoderado para organizar y gestionar su trabajo. La sinergia que ocurre, como resultado, optimizará la eficiencia y efectividad del Equipo de Desarrollo. Estos equipos tienen las siguientes características: (Goncalves, 2019)

- Se auto-organizan. No reciben instrucciones ni consejo de nadie sobre cómo convertir el Backlog de Producto en Incrementos de funcionalidades potencialmente liberables.
- Son multifuncionales y tienen todas las habilidades necesarias para crear un Incremento de producto.
- Scrum no otorga ningún título al Equipo de Desarrollo. Todo el mundo es desarrollador, sin importar el trabajo que desempeñen. Esto se aplica sin excepciones.
- Dentro del Equipo de Desarrollo, el Scrum reconoce que no hay sub-equipos. Esto ocurre incluso cuando hay muchos dominios que tienen que ser tratados, como el análisis de negocios o el testeo. Esto se aplica sin excepciones.
- Cada miembro del Equipo de Desarrollo podría tener una habilidad o zona de confort especial, pero si hablamos de responsabilidades, se considera al equipo como un todo.

El Equipo de Desarrollo se compone de muchas personas, siendo tres el número óptimo. Esto asegura que se mantienen ágiles y son un lo suficientemente grandes para finalizar todo el trabajo que se debe hacer en un Sprint. (Goncalves, 2019)

Si el equipo tiene menos de tres miembros, la interacción puede verse reducida, lo que significa que se tendrá menor productividad. Además, los Equipos de Desarrollo pequeños pueden ver restringidas sus capacidades durante el Sprint, significando que podrían no ser capaces de entregar un Incremento potencialmente liberable. (Goncalves, 2019)

Los equipos grandes, como aquellos con más de nueve miembros, necesitan mucha más coordinación. Acaban generando demasiada complejidad como para gestionar un proceso empírico. Al contar al Equipo de Desarrollo, no se

cuenta al Product Owner ni al Scrum Master a no ser que también estén realizando el trabajo del Backlog del Sprint. (Goncalves, 2019)

2.6.4.1. Las gallinas. Aunque no son parte del proceso de Scrum, es necesario que parte de la retroalimentación de la salida del proceso y así poder revisar y plantear cada sprint. (Trigas Gallego, 2012)

a) Usuarios. Es el destinatario final del producto. (Goncalves, 2019)

b) Stakeholders. Son las personas a las que el proyecto les producirá un beneficio. Participan durante las revisiones del Sprint. (Goncalves, 2019)

c) Managers. Toma las decisiones finales participando en la selección de los objetivos y requisitos. (Goncalves, 2019)

2.6.5. Eventos de scrum (reuniones). Son normalmente eventos programados que se usan en Scrum. Normalmente regulan y minimiza la necesidad de llevar a cabo reuniones no planeadas. Todos estos eventos son de tiempo limitado, lo que significa que tienen una duración máxima. (Goncalves, 2019)

En el momento que un evento, como un Sprint, comienza, tiene una duración fija que no puede cambiarse. Una vez que se cumple el objetivo del evento, todos los eventos restantes pueden finalizarse. Esto asegura que se gasta el mínimo tiempo posible en el proceso. (Goncalves, 2019)

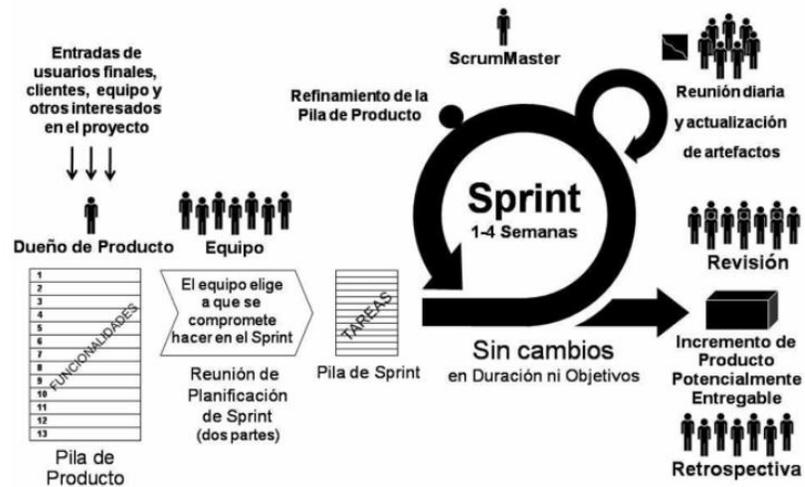


Figura 5. Eventos de la Metodología Scrum.
Fuente: (Araque, 2017)

Cada evento da la oportunidad de inspeccionar o adaptar algo y ha sido diseñado para permitir una transparencia radical. Hay cuatro eventos formales programados en el método Scrum, que son: (Goncalves, 2019)

- Planificación del Sprint
- Scrum Diario
- Revisión del Sprint
- Retrospectiva del Sprint

2.6.5.1. ¿Qué es un sprint? El corazón del método Scrum es el Sprint. Se puede definir como un periodo de tiempo de un mes o menos en el que se crea un producto liberable, utilizable y “Finalizado”. Normalmente tienen una duración consistente durante un periodo de desarrollo. Los sprints deberían tener duraciones constantes durante todo el desarrollo. Un nuevo Sprint comienza solo cuando el anterior ha finalizado. (Goncalves, 2019)

Los Sprints contienen y consisten de la Planificación del Sprint, los Scrums Diarios, la Revisión del Sprint, la Retrospectiva del Sprint y el Trabajo de

Desarrollo. Cuando el Sprint está en curso, no debería haber ninguna alteración que pudiera afectar al objetivo del mismo, los objetivos cualitativos no descienden, y el enfoque podría ser aclarado y renegociado entre el Product Owner y el Equipo de Desarrollo. (Goncalves, 2019)

Es seguro considerar cada Sprint como un proyecto de un mes en el que se ha de conseguir algo. Por esta razón, en el Sprint se define claramente qué se va a construir, diseñar, un plan para la producción, el trabajo y el producto final. Si el Sprint durara más de un mes, la definición de lo que se va a crear cambiaría, y el riesgo podría subir junto a la complejidad general. Los Sprints son importantes ya que aseguran que el trabajo es predecible y puede ser inspeccionado y adaptado mientras se trabaja por el objetivo del Sprint. Limitan el riesgo, particularmente si nos fijamos en el coste. (Goncalves, 2019)

2.6.5.2. Planificación del Sprint. Cualquier trabajo que va a hacerse en el Sprint se planea en la Planificación del Sprint. La planificación une al Equipo Scrum y se crea mediante un trabajo colaborativo. La planificación del Sprint tiene una duración concreta de ocho horas como máximo para un Sprint de un mes; aquí puedes encontrar sugerencias sobre cómo llevar a cabo una efectiva planificación del sprint. (Goncalves, 2019)

Cuando el Sprint dura menos, también el evento tiene a ser más corto. Depende del Scrum Master asegurar que el evento tiene lugar y que los asistentes entienden la razón del mismo. El Scrum Master enseñará al equipo a mantenerse en la duración establecida. (Goncalves, 2019)

En la Planificación se responden las preguntas sobre qué se puede liberar en el Incremento del siguiente Sprint, y cómo se conseguirá realizar el trabajo que se va a liberar. (Goncalves, 2019)

2.6.5.3. Objetivo del sprint. Es un objetivo que se pone en el Sprint, que se consigue con facilidad una vez que se ha implementado el Backlog de Producto. Actúa, de alguna manera, como una guía al Equipo de Desarrollo sobre la razón por la que se crea el Incremento. El Objetivo del Sprint se establece en la reunión de Planificación del Sprint. (Goncalves, 2019)

Se asegura que el Equipo de Desarrollo tiene un nivel de flexibilidad, en relación a la funcionalidad implementada en el Sprint. El Objetivo del Sprint se entrega a través de los Elementos del Backlog del Producto y asegura que el Equipo de Desarrollo puede trabajar conjuntamente, en lugar de dividirse en diferentes iniciativas. (Goncalves, 2019)

El Equipo de Desarrollo trabaja con el Objetivo del Sprint siempre en la mente, y esto afecta su funcionalidad y uso de la tecnología. Cuando haya un cambio en las expectativas, el Equipo de Desarrollo y el Product Owner facilitarán la colaboración para negociar el enfoque que se da al Backlog de Producto en el Sprint. (Goncalves, 2019)

2.6.5.4. Scrum diario. Cada día, el Equipo de Desarrollo deja 15 minutos para sincronizar sus actividades y desarrollar un plan para las siguientes 24 horas. Esto se conoce como el Scrum Diario. Incluye una inspección del trabajo que se ha hecho desde el anterior Scrum Diario y posteriormente se enfatiza sobre el trabajo que ha de realizarse antes del siguiente. (Goncalves, 2019)

Es esencial que la hora y el lugar del Scrum Diario sean siempre iguales, ya que esto evitará cualquier complejidad. En la reunión, el Equipo de Desarrollo se centrará en: (Goncalves, 2019)

- Qué se hizo el día anterior que ayudó a alcanzar el Objetivo del Sprint
- Qué se ha hecho hoy para ayudar a alcanzar el Objetivo del Sprint

- Qué impedimentos pueden poner trabas a la consecución del Objetivo del Sprint

Esto significa que el Scrum Diario es como una manera de chequear el grado de consecución del Objetivo del Sprint. Asegura que el trabajo se basa en el Backlog del Sprint y facilita la tarea de alcanzar el objetivo. Se desafía al Equipo de Desarrollo durante el Scrum Diario a trabajar conjuntamente como un equipo auto-organizado a través de discusiones detalladas, en las cuales puede que tengan que adaptar o repetir el trabajo del resto del Sprint. (Goncalves, 2019)

Es el Scrum Master quien asegura que esta reunión se realiza a diario, aunque está dirigida por el Equipo de Desarrollo. Los Scrum Masters aseguran que el equipo no se sobrepasa de los 15 minutos, y hacen cumplir las reglas de participación. (Goncalves, 2019)

Los beneficios del Scrum Diario son numerosos, incluyendo la mejora en la comunicación, la no necesidad de otras reuniones, la identificación de trabas al desarrollo, la rápida toma de decisiones y el aumento del conocimiento general del Equipo de Desarrollo. Hace un tiempo escribí un artículo sobre “Cómo llevar a cabo un Scrum Diario efectivo”, al que puedes echar un vistazo. (Goncalves, 2019)

2.6.5.5. Revisión del sprint. La Revisión del Sprint se lleva a cabo una vez que el Sprint ha finalizado. En ella, se inspecciona el Incremento y se adapta el Backlog de Producto si fuera necesario. Cuando la Revisión del Sprint tiene lugar, el Equipo Scrum y las partes interesadas evalúan qué se ha hecho durante el Sprint. (Goncalves, 2019)

El resultado de la discusión podría llevar a adoptar cambios en el backlog de Producto, así como a la revisión de lo que podría ser optimizado con el objetivo de añadir valor. La revisión sirve para compartir información y no es una reunión

sobre el estado del proyecto. Solo se mantiene para conseguir algo de feedback y para asegurar que existe colaboración. Duraría unas cuatro horas si el Sprint fuera de un mes. Si el Sprint fuera más corto, también lo sería la reunión. (Goncalves, 2019)

Después de la Revisión del Sprint, el Backlog de Producto se suele revisar, y se definen los elementos del Backlog de Producto que se utilizarán probablemente en el siguiente Sprint. Puede que se hagan ajustes generales para encontrar nuevas oportunidades. (Goncalves, 2019)

2.6.5.6. Retrospectiva del sprint. Es una oportunidad para el Equipo Scrum de realizar una inspección sobre lo que se ha realizado y desarrollar un plan para conseguir mejoras en el siguiente Sprint. Ocurre una vez que la Revisión del Sprint se ha finalizado, antes de que la Planificación del Sprint vuelva a comenzar. (Goncalves, 2019)

Es una reunión con una duración concreta de unas tres horas para Sprints de un mes. Sprints más cortos darán lugar a reuniones más cortas. El Scrum Master se asegurará de que la reunión tiene lugar y los asistentes entienden claramente su propósito. (Goncalves, 2019)

El Scrum Master usará esta reunión para animar al Equipo Scrum a mejorar en las prácticas y procesos del desarrollo. Esto asegurará que el siguiente Sprint sea más efectivo y disfrutable. Se implementan los planes para elevar la calidad del producto, estableciendo una definición apropiada de producto "Finalizado". (Goncalves, 2019)

Una vez que se ha completado la Retrospectiva del Sprint, el Equipo Scrum habrá identificado qué se puede mejorar en el siguiente Sprint. Se adoptarán estas mejoras y serán inspeccionadas por el propio Equipo Scrum. La Retrospectiva del

Sprint es una oportunidad formal de centrarse en la inspección y la adaptación. (Goncalves, 2019)

2.6.6. Elementos de Scrum. Representan el trabajo o valor para proporcionar transparencia, así como oportunidades para la inspección y la adaptación. Se definen como específicamente diseñadas para maximizar la transparencia sobre la información clave, de manera que todos tienen el mismo conocimiento del elemento en cuestión.

2.6.6.1. Product Backlog (backlog de producto). Esto se refiere a una lista ordenada de todas las cosas que se requieren para desarrollar un producto. Contiene todos los requisitos para cualquier corrección que se tenga que hacer a un producto; el responsable de ello es el Product Owner. (Goncalves, 2019)

Es un trabajo en progreso (WIP), y no llega a tener nunca una versión final. Establece los requisitos y evoluciona a la vez que el producto. Sufre cambios basándose en qué es más conveniente, útil y competitivo para el producto y existirá mientras que exista el producto. (Goncalves, 2019)

Contiene una lista de todas las funciones, requisitos, características, mejoras y arreglos, que constituyen los cambios que tienen que realizarse al producto en la siguiente release. Los elementos del Backlog del Producto tienen una descripción, orden, estimación y valor. (Goncalves, 2019)

Cuanto más se use el producto, y obtenga valor, más feedback puede esperarse del mercado. Esto resultará en el crecimiento del Backlog del Producto, al mismo tiempo que los requisitos evolucionan. También puede sufrir cambios basados en los requisitos de negocio, condiciones del mercado y la tecnología. (Goncalves, 2019)

Donde hay muchos Equipos Scrum, puede que necesiten trabajar juntos en un producto. En Este caso, solo un Backlog de Producto será utilizado para describir el producto. Para refinar el Backlog del producto, los detalles, las estimaciones y el orden de los elementos podría ser modificado. (Goncalves, 2019)

Esto lo realizan mediante la colaboración el Product Owner y el Equipo de Desarrollo. Mientras se refina, también se revisan los elementos que se quedan dentro. Es el Equipo Scrum quien decide cómo se realizará el perfeccionamiento, de manera que no se coja más del 10% de la capacidad del Equipo de Desarrollo. El Product Owner tiene la capacidad de actualizar los elementos del Backlog del Producto en cualquier momento. (Goncalves, 2019)

Hay elementos del Backlog del Producto ordenados más arriba y otros más abajo, siendo los que están más arriba los más aclarados y detallados. Son los que están arriba los elementos que pueden ser “Finalizados” por el Equipo de Desarrollo, y pueden ser considerados como “Listos” para la selección en la Planificación del Sprint. El Equipo de Desarrollo es responsable de todas las estimaciones, ya que son los que van a realizar el trabajo. (Goncalves, 2019)

2.6.6.2. *Sprint Backlog (backlog del sprint)*. Se conoce a los elementos del Backlog de Producto que se han seleccionado para el Sprint como el Backlog del Sprint. Incluyen también el plan para entregar el Incremento del Producto y alcanzar el Objetivo del Sprint. Fundamentalmente, el Backlog del Sprint es un pronóstico del Equipo de Desarrollo que se centra en saber cuán funcional será cada Incremento, así como el trabajo que es necesario para convertir la funcionalidad en un Incremento “Finalizado”. (Goncalves, 2019)

El Backlog del Sprint asegura que todo el trabajo realizado por el Equipo de Desarrollo es visible y que se puede alcanzar el Objetivo del Sprint. El Backlog del

Sprint es esencialmente un plan que contiene el detalle necesario para que en el Scrum Diario se comprendan los cambios realizados. (Goncalves, 2019)

El Equipo de Desarrollo puede modificar el Backlog del Sprint durante todo el Sprint y después surgir durante el Sprint, mientras que el Equipo de Desarrollo trabaja en el plan. Esto se debe a que se aprende más sobre el trabajo necesario para alcanzar el Objetivo del Sprint. (Goncalves, 2019)

Si existe la necesidad de finalizar el trabajo, el Equipo de Desarrollo lo añade al Backlog del Sprint. Mientras se completa, el trabajo restante se actualiza. Por el camino, se elimina cualquier elemento del plan que no sea necesario. (Goncalves, 2019)

Solo el Equipo de Desarrollo puede cambiar el Backlog del Producto en el curso de un Sprint. El Backlog del Sprint es como una huella visible para el Equipo de Desarrollo y pertenece solamente a este equipo. (Goncalves, 2019)

2.6.6.3. Incremento. Se puede resumir todo el trabajo que resta del Backlog del Sprint en cualquier momento. El Incremento es la suma de esto y del valor de cualquier otro Incremento de Sprints previos. El incremento final al terminar el Sprint debería estar “Finalizando”, que significa que está en una condición utilizable y cumple con la definición establecida por el Equipo Scrum. La condición utilizable es indispensable, sea cual sea la decisión del Product Owner de lanzarlo o no. (Goncalves, 2019)

2.6.6.4. Transparencia en los elementos. Para que el método Scrum sea como es, la transparencia es esencial. Es la base de las decisiones que se toman para optimizar el valor y el control del riesgo. El grado de cumplimiento de la transparencia determina si las decisiones son correctas o no. Si los elementos no

son completamente transparentes, entonces las decisiones tienen defectos y su valor disminuye. Además, el riesgo aumentará. (Goncalves, 2019)

El Scrum Master debe trabajar con todas las partes participantes, incluyendo el Product Owner y el Equipo de Desarrollo, para asegurar que los elementos son totalmente transparentes. En el caso de que no exista una fuerte transparencia, hay prácticas para salir adelante. (Goncalves, 2019)

El Scrum Master debe ayudar a todos a aplicar las prácticas más apropiadas para conseguir una buena transparencia. Un Scrum Master puede detectar una transparencia incompleta mediante la inspección de los elementos, el razonamiento sobre los patrones y la observación cuidadosa de toda la información disponible. (Goncalves, 2019)

Desde aquí, se pueden detectar las diferencias entre los resultados reales y los esperados. El Scrum Master ha de trabajar con el Equipo Scrum y la organización para aumentar la transparencia de los elementos. Esto se consigue mediante el aprendizaje, el convencimiento y los cambios, y requiere un largo camino. (Goncalves, 2019)

2.7. Ingeniería web

Con la difusión y avance que obtuvo la World Wide Web e Internet, se ha logrado que numerosas actividades giren entorno y dependan de los servicios que ofrecen estas como ser variedad de contenido y funcionalidades, que responden a las necesidades de los usuarios. (Mamani Llipe, 2014)

Con este cambio, dentro de la administración de empresas se vio un giro trascendental, como ser, la interacción con los clientes sin la necesidad de atenderlos personalmente y a la vez reduciendo costos en diversos aspectos

dentro de la empresa o ente. De esta manera las empresas han optado por la implementación de sistemas de información en la web. (Mamani Llipo, 2014)

2.7.1. Definición. La ingeniería web es la aplicación de metodologías sistemáticas, disciplinadas y cuantificables al desarrollo eficiente, operación y evolución de aplicaciones de alta calidad en la World Wide Web. (Mamani Llipo, 2014)

La ingeniería web se debe al crecimiento desenfrenado que está teniendo la Web está ocasionando un impacto en la sociedad y el nuevo manejo que se le está dando a la información en las diferentes áreas en que se presenta ha hecho que las personas tiendan a realizar todas sus actividades por esta vía. (Mamani Llipo, 2014)

Si bien la ingeniería del software, brinda fases y pasos a seguir para el desarrollo de un sistema de información o software; un sistema web requiere un trato diferente, por las especificaciones y características que conlleva Powell (1998 citado en Pressman, 2002) resume que los asistentes Web “Implican una mezcla de publicación impresa o ente y desarrollo de software, de marketing e informática, de comunicaciones internas y relaciones externas. Y de arte y tecnología.” (Mamani Llipo, 2014)

2.8. Metodología UWE

2.8.1. Lenguaje unificado de modelado (UML). El Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). (Aspiazu Gutierrez, 2016)

Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados. (Aspiazu Gutierrez, 2016)

Es importante remarcar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo. (Aspiazu Gutierrez, 2016)

Se puede aplicar en el desarrollo de software gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado Racional o RUP), pero no especifica en sí mismo qué metodología o proceso usar. (Aspiazu Gutierrez, 2016)

UML no puede compararse con la programación estructurada, pues UML significa Lenguaje Unificado de Modelado, no es programación, solo se diagrama la realidad de una utilización en un requerimiento. Mientras que, programación estructurada, es una forma de programar como lo es la orientación a objetos, la programación orientada a objetos viene siendo un complemento perfecto de UML, pero no por eso se toma UML sólo para lenguajes orientados a objetos. (Aspiazu Gutierrez, 2016)

2.8.1. UML – based web engineering (UML). Es una herramienta que nos permitirá modelar aplicaciones web, utilizada en la ingeniería web, prestando especial atención en sistematización y personalización (sistemas adaptativos).

UWE es una propuesta basada en el proceso unificado y UML, pero adaptados a la web. (Aspiazu Gutierrez, 2016)

En el marco de UWE es necesario la definición de un perfil UML (extensión) basado en estereotipos con este perfil se logra la asociación de una semántica distinta a los diagramas del UML puro, con el propósito de acoplar el UML a un dominio específico, en este caso, las aplicaciones Web. (Aspiazu Gutierrez, 2016)

UWE permite especificar de mejor manera una aplicación Web en su proceso de creación mantiene una notación estándar basada en el uso de UML (Unified Modeling Language) para sus modelos y sus métodos, lo que facilita la transición. La metodología define claramente la construcción de cada uno de los elementos del modelo. (Nieves G., Ucan P., & Menendez D., 2014)

En su implementación se deben contemplar las siguientes etapas y modelos:

- **Análisis de requisitos.** Plasma los requisitos funcionales de la aplicación Web mediante un modelo de casos de uso.
- **Modelo de contenido.** Define, mediante un diagrama de clases, los conceptos a detalle involucrados en la aplicación.
- **Modelo de navegación.** Representa la navegación de los objetos dentro de la aplicación y un conjunto de estructuras como son índices, menús y consultas.
- **Modelo de presentación.** Representa las interfaces de usuario por medio de vistas abstractas.
- **Modelo de proceso.** Representa el aspecto que tienen las actividades que se conectan con cada clase de proceso.

Como se hace notar, UWE provee diferentes modelos que permite describir una aplicación Web desde varios puntos de vista abstractos, dichos modelos están relacionados tal como se muestra en la siguiente figura. (Nieves G., Ucan P., & Menendez D., 2014)

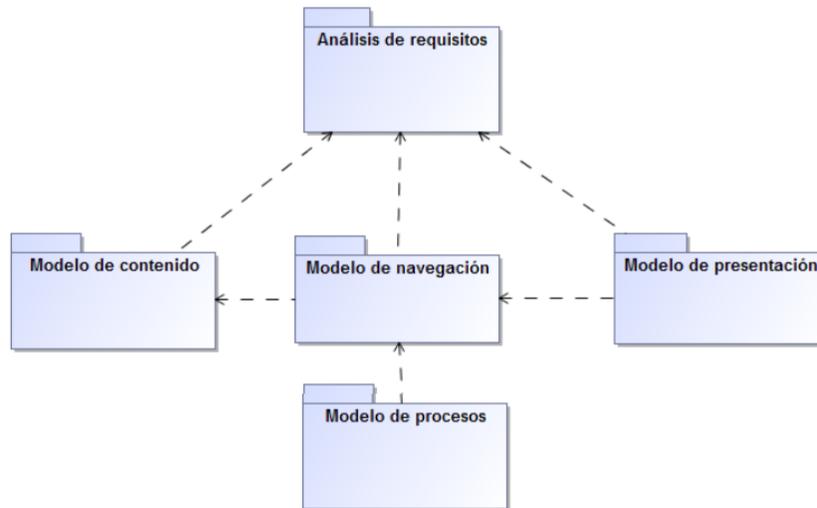


Figura 6. Modelos de UWE.

Fuente: (Nieves G., Ucan P., & Menendez D., 2014)

Cada uno de estos modelos se representa como paquetes UML, dichos paquetes son procesos relacionados que pueden ser refinados en iteraciones sucesivas durante el desarrollo del UWE. (Nieves G., Ucan P., & Menendez D., 2014)

El análisis de requisitos en UWE se modela con casos de uso. Está conformado por los elementos actor y caso de uso. En este sentido, los actores se utilizan para modelar los usuarios de la aplicación Web. (Nieves G., Ucan P., & Menendez D., 2014)

El modelo de contenido es el modelo conceptual del dominio de aplicación tomando en cuenta los requerimientos especificados en los casos de uso y se representa con un diagrama de clases. Basado en el análisis de requisitos y el modelo de contenido se obtiene el modelo de navegación. Éste se representa con

clases de navegación que serán explicados en el caso de estudio de este artículo. Basado en el modelo de navegación y en los aspectos de la interfaz usuario (requisitos), se obtiene el modelo de presentación. Dicho modelo describe la estructura de la interacción del usuario con la aplicación Web. El modelo de navegación puede ser extendido mediante clases de procesos. El modelo del proceso representa el aspecto que tienen las acciones de las clases de proceso. (Nieves G., Ucan P., & Menendez D., 2014)

2.8.2. Modelo de requisitos. Una de las primeras actividades en la construcción de aplicaciones Web es la identificación de los requisitos, en UWE el modelado de requisitos consiste de dos partes: Casos de uso de la aplicación y sus relaciones y actividades describiendo los casos de uso en detalle.

2.8.2.1. Casos de uso. El diagrama de casos de uso está conformado por los elementos actor y caso de uso. Los actores se utilizan para modelar los usuarios de la aplicación Web. Los casos de uso se utilizan para visualizar las diferentes funcionalidades que la aplicación tiene que proporcionar. (Nieves G., Ucan P., & Menendez D., 2014)

En UWE se distinguen casos de uso estereotipados con «browsing» y con «processing» para ilustrar si los datos persistentes de la aplicación son modificados o no. "SearchContact" por ejemplo, modela la búsqueda de contactos y por ello lleva el estereotipo «browsing» pues los datos son solamente leídos y presentados al usuario. Los otros casos de uso por el contrario modelan cambios, lo que se especifica con el estereotipo «processing». (Web Engineering Group, 2016)

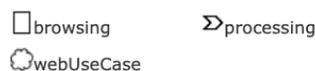


Figura 7. Nombres de estereotipos y sus íconos casos de uso.
Fuente: (Web Engineering Group, 2016)

En la siguiente figura se muestra un ejemplo de Casos de uso de una agenda de direcciones.

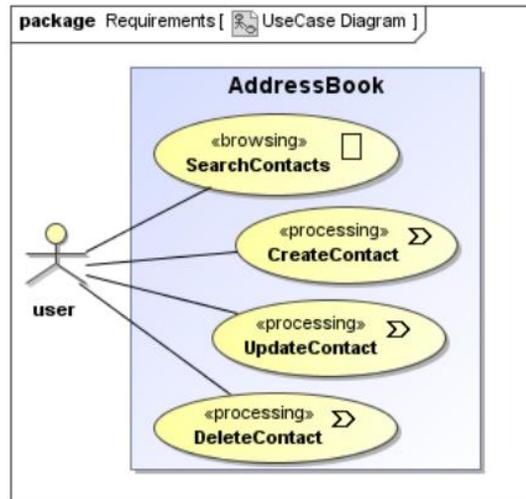


Figura 8. Casos de uso de un ejemplo de agenda de direcciones.
Fuente: (Web Engineering Group, 2016)

2.8.2.2. Actividades. Como con los casos de uso solamente es posible capturar poca información, cada caso de uso puede ser descrito más detalladamente mediante un proceso. Es decir, las acciones que son parte de un caso de uso así como los datos presentados al usuario y aquellos requeridos como entrada de datos pueden ser modelados con precisión como actividades. (Web Engineering Group, 2016)

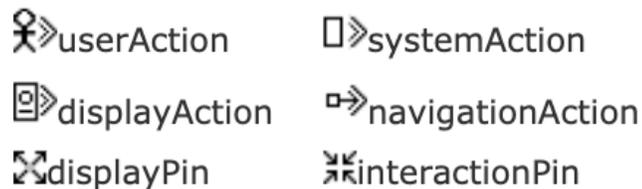


Figura 9. Nombres de estereotipos y sus íconos actividades.
Fuente: (Web Engineering Group, 2016)

Los dos estereotipos «user Action» y «system Action» pueden ser usados análogamente al flujo de procesos. El estereotipo «user Action» es usado para indicar interacciones de usuario en la página web iniciando un proceso o respondiendo a un explícito requisito de información. Por lo contrario, «system

Action» describe acciones que son ejecutados por el sistema. Ambos tipos de acciones pueden ser insertados usando la toolbar. (Web Engineering Group, 2016)

Detalles de las estructuras de datos usadas pueden ser representados por objetos de nodos y pins de acciones. El objeto de nodo es usado para modelar clases de contenido y los pines sus atributos. (Web Engineering Group, 2016)

Durante ingeniería de requisitos es usual determinar qué datos son representados donde y cuando. Para modelar grupos de presentación en UWE son usados el estereotipo «display Action», mientras que los dos pines de acción estereotipados «interaction Pin» y «display Pin» son usados para modelar la entrada y la salida de datos. (Web Engineering Group, 2016)

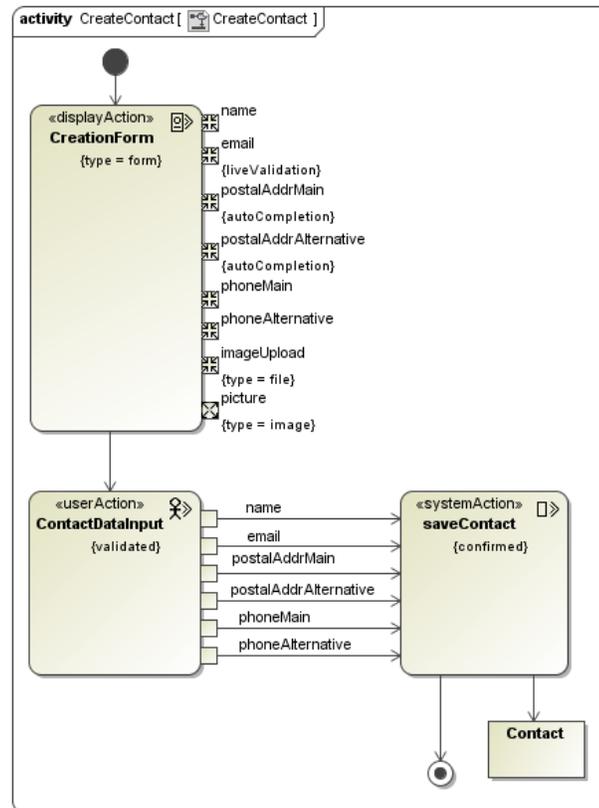


Figura 10. Actividad para el caso de uso "CreateContact" del ejemplo agenda de direcciones.

Fuente: (Web Engineering Group, 2016)

Finalmente, el estereotipo «navigationAction», puede ser usado para modelar opciones de navegación y los elementos asociados de presentación. (Web Engineering Group, 2016)

Como estos estereotipos se utilizan para indicar elementos de presentación durante la etapa de ingeniería de requisitos, aspectos que caracterizan a RIAs pueden ser especificadas mediante valores etiquetados para estos mismos elementos. (Web Engineering Group, 2016)

2.8.3. Modelo de contenido. El diseño conceptual se basa en el análisis de requisitos del paso anterior. Esto incluye los objetos involucrados entre los usuarios y la aplicación. Este modelo propone construir un modelo de clases con estos objetos, ignorando los aspectos de navegación: presentación e interacción, que serán tratados posteriormente. Los principales elementos de modelado son; las clases, asociaciones y paquetes. (Huanca Cantuta, 2015)

En sí es un diagrama UML normal de clases, por ello se debe pensar en las clases que son necesarias para el ejemplo. (Web Engineering Group, 2016)

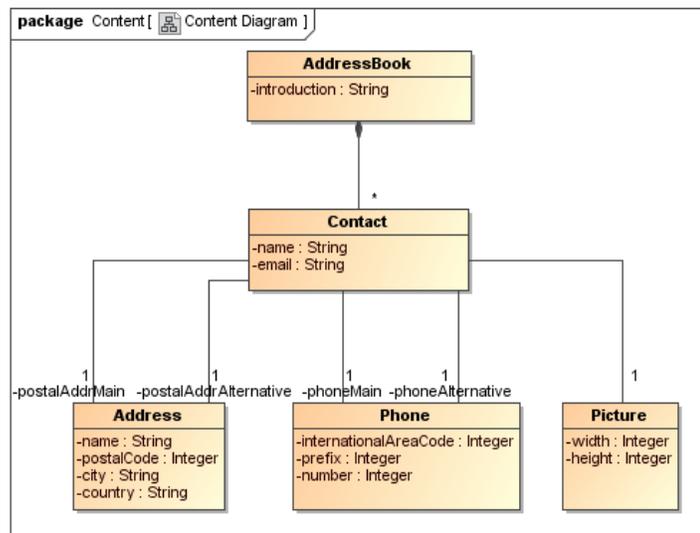


Figura 11. Modelo de contenido para el ejemplo agenda de direcciones.

Fuente: (Web Engineering Group, 2016)

2.8.4. Modelo de navegación. En un sistema para la web es útil saber como están enlazadas las páginas. Ello significa que necesitamos un diagrama conteniendo nodos (nodes) y enlaces (links). (Web Engineering Group, 2016)

¿Pero que es un nodo? Nodos son unidades de navegación y están conectados por medio de enlaces. Nodos pueden ser presentados en diferentes páginas o en una misma página. (Web Engineering Group, 2016)

UWE provee diferentes estereotipos, los que presentaremos mediante el ejemplo.

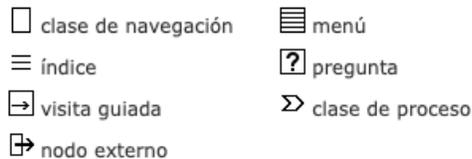


Figura 12. Nombres de estereotipos y sus íconos del modelo de navegación.

Fuente: (Web Engineering Group, 2016)

El diseño navegacional no es solo útil para la generación de la documentación de la estructura de la aplicación, sino que también permite mejorar la estructura de navegabilidad. (Huanca Cantuta, 2015)

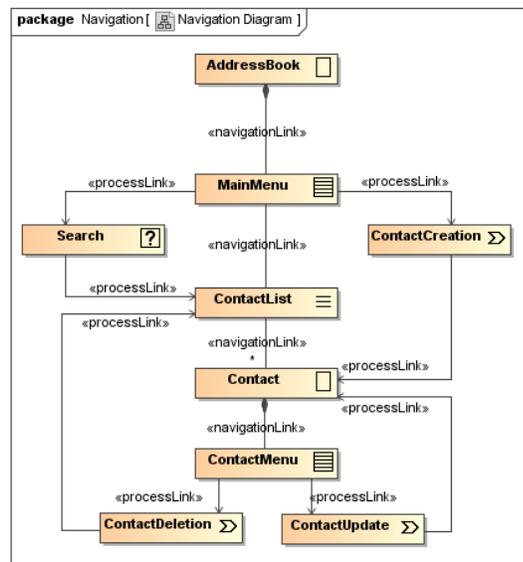


Figura 13. Modelo de navegación para el ejemplo agenda de direcciones.

Fuente: (Web Engineering Group, 2016)

Cuando hablamos de un sistema web, es necesario conocer la relación y los enlaces entre las páginas web, es por eso que en la fase de diseño se describen a través de diagramas la navegación del sistema cumpliendo con lo que se diseño en los casos de uso. (Huanca Cantuta, 2015)

En la figura 12 se muestra el diagrama de navegación de una agenda de direcciones citado en el tutorial de la página oficial de UWE.

2.8.5. Modelo de presentación. Este modelo permite una visión amplia de los procesos de la página web que se representan en los diagramas de navegación; pueden interpretarse también con las interfaces del sistema web, para el caso se tiene estereotipos o iconos que ayudan al diseño de los diagramas de presentación. (Huanca Cantuta, 2015)

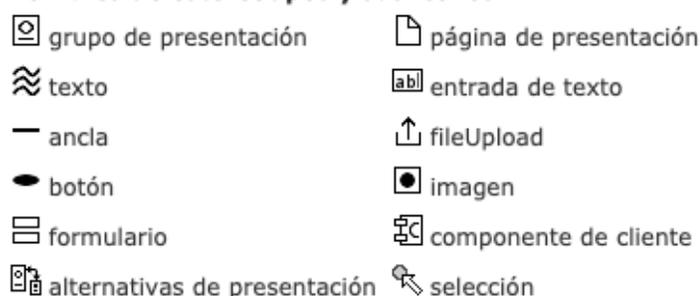


Figura 14. Nombres de estereotipos y sus íconos del modelo de presentación.

Fuente: (Web Engineering Group, 2016)

El diagrama de presentación de la metodología UWE, permite al usuario comprender y analizar, sobre el área de trabajo al que se someterá con la implantación del sistema. (Huanca Cantuta, 2015)

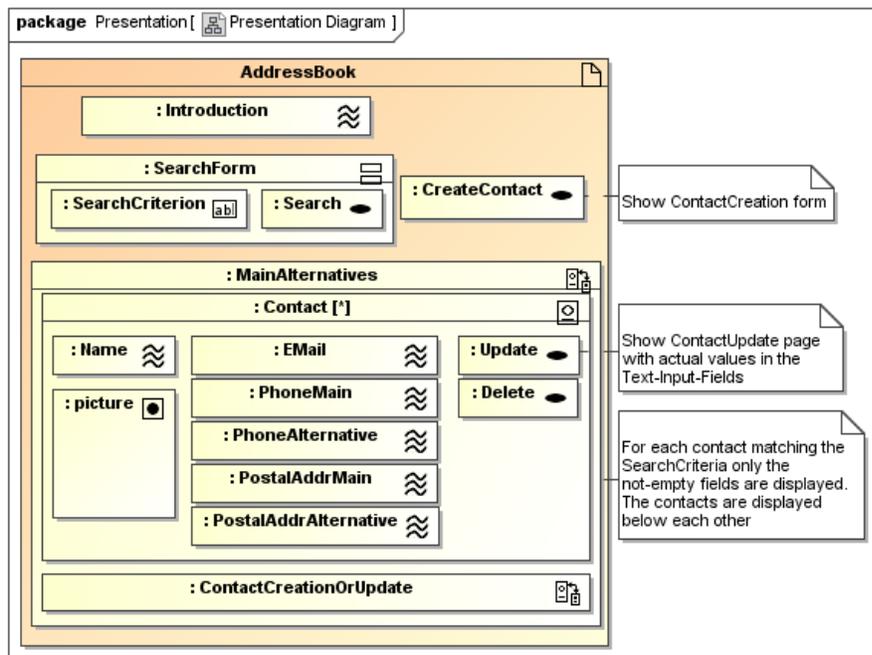


Figura 15. Modelo de presentación para el ejemplo agenda de direcciones.
Fuente: (Web Engineering Group, 2016)

2.8.6. Modelo de proceso. El Modelo de Proceso comprende los siguientes modelos.

- El Modelo de Estructura del Proceso que describe las relaciones entre las diferentes clases de proceso.
- El Modelo de Flujo del Proceso que especifica las actividades conectadas con cada «processClass».

2.8.6.1. Modelo de Estructura del Proceso. Con el fin de describir las relaciones entre las diferentes clases de proceso, creamos un diagrama de clases. Después de haber realizado el diseño tenemos un diagrama de clases con tres clases enmarcadas con un borde rojo: (Web Engineering Group, 2016)

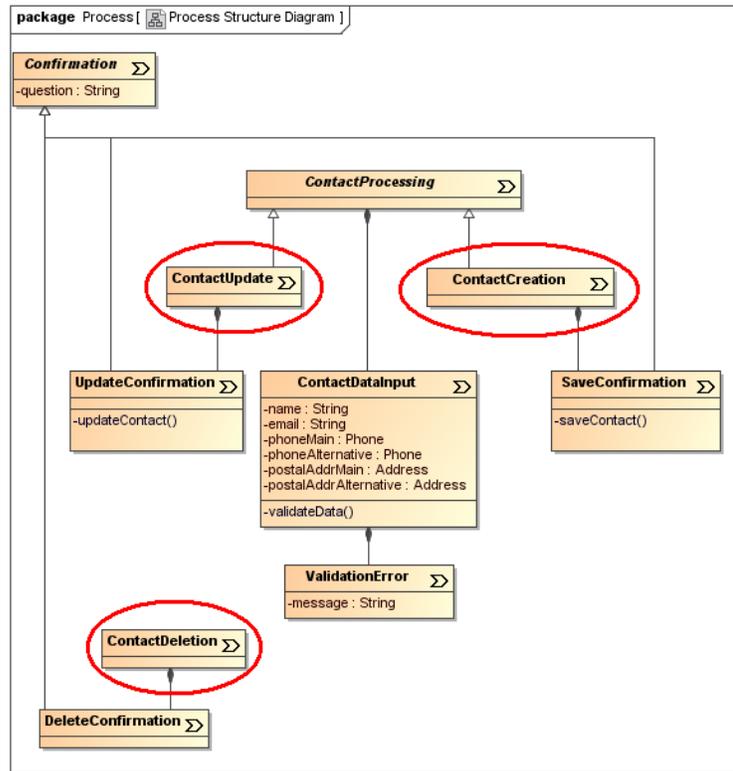


Figura 16. Modelo de estructura del proceso para el ejemplo agenda de direcciones.

Fuente: (Web Engineering Group, 2016)

2.8.6.1. Modelo de flujo del Proceso. Un flujo del proceso (flujo de trabajo) es representado como un diagrama de actividades, describiendo el comportamiento de una clase de proceso por ejemplo, que sucede en detalle, cuando el usuario navega a una clase de proceso . (Web Engineering Group, 2016)



Figura 17. Nombres de estereotipos y sus íconos del modelo de proceso.

Fuente: (Web Engineering Group, 2016)

Los diagramas de actividades incluyen actividades, actores responsables de estas actividades (opcional) y elementos de flujo de control. Estos diagramas representan el flujo del proceso, describiendo el comportamiento de una clase de proceso. (Huanca Cantuta, 2015)

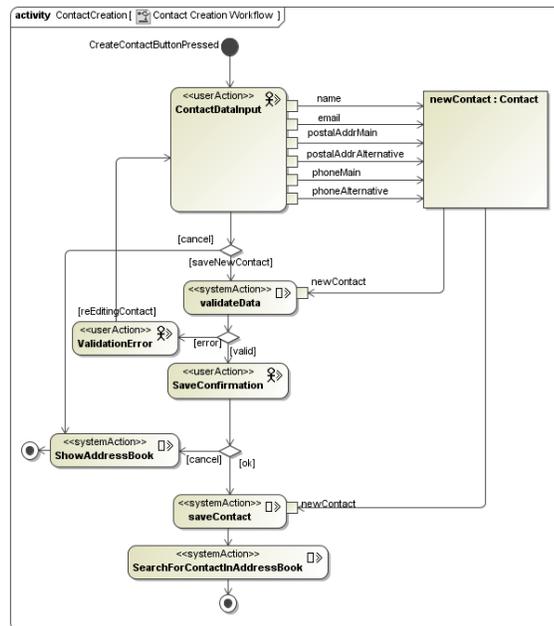


Figura 18. Modelo de flujo del proceso para el ejemplo agenda de direcciones
Fuente: (Web Engineering Group, 2016)

2.9. Calidad de software

Incluso los desarrolladores de software más experimentados estarán de acuerdo en que obtener software de alta calidad es una meta importante. Pero, ¿cómo se define la calidad del software? En el sentido más general se define como: Proceso eficaz de software que se aplica de manera que crea un producto útil que proporciona valor medible a quienes lo producen y a quienes lo utilizan. (Pressman, 2010)

Hay pocas dudas acerca de que la definición anterior podría modificarse o ampliarse en un debate sin fin. Para propósitos de este libro, la misma sirve a fin de enfatizar tres puntos importantes: (Pressman, 2010)

1. Un *proceso eficaz* de software establece la infraestructura que da apoyo a cualquier esfuerzo de elaboración de un producto de software de alta calidad. Los aspectos de administración del proceso generan las

verificaciones y equilibrios que ayudan a evitar que el proyecto caiga en el caos, contribuyente clave de la mala calidad. Las practicas de ingeniería de software permiten al desarrollador analizar el problema y diseñar una solución solida, ambas actividades críticas de la construcción de software de alta calidad. Por ultimo, las actividades sombrilla, tales como administración del cambio y revisiones técnicas, tienen tanto que ver con la calidad como cualquier otra parte de la práctica de la ingeniería de software. (Pressman, 2010)

2. Un *producto útil* entrega contenido, funciones y características que el usuario final desea; sin embargo, de igual importancia es que entrega estos activos en forma confiable y libre de errores. Un producto útil siempre satisface los requerimientos establecidos en forma explícita por los participantes. Además, satisface el conjunto de requerimientos (por ejemplo, la facilidad de uso) con los que se espera que cuente el software de alta calidad. (Pressman, 2010)
3. Al agregar valor para el productor y para el usuario de un producto, el software de alta calidad proporciona beneficios a la organización que lo produce y a la comunidad de usuarios finales. La organización que elabora el software obtiene valor agregado porque el software de alta calidad requiere un menor esfuerzo de mantenimiento, menos errores que corregir y poca asistencia al cliente. Esto permite que los ingenieros de software dediquen más tiempo a crear nuevas aplicaciones y menos a repetir trabajos mal hechos. La comunidad de usuarios obtiene valor agregado porque la aplicación provee una capacidad útil en forma tal que agiliza algún proceso de negocios. El resultado final es 1) mayores utilidades por el producto de software, 2) más rentabilidad cuando una aplicación apoya un proceso de negocios y 3) mejor disponibilidad de información, que es crucial para el negocio. (Pressman, 2010)

2.10. Modelo de calidad establecido por el estándar ISO 9126

Antes de explicar que es el modelo de calidad establecido por el estándar ISO 9126 tenemos que saber algunos conceptos previos para un mejor entendimiento.

2.10.1. ¿Qué es la norma ISO 9126? La norma ISO/IEC 9126 es un estándar internacional para la evaluación del software que surge debido a la necesidad de un modelo unido para expresar la calidad de un software. Fue publicado en 1992 con el nombre de “Information technology - Software product evaluation: Quality characteristics and guidelines for their use”, en el cual se establecen las características de calidad para productos de software. (De los Santos, 2019)

2.10.2. Estándar. El estándar se divide en cuatro partes que trata los siguientes temas: modelo de la calidad, métricas externas, métricas internas y métricas de calidad de uso. A su vez el modelo de calidad que propone este estándar se divide en dos: calidad externa y calidad interna, así como la calidad durante el uso. La primera parte del modelo especifica seis características para la calidad interna y externa, que son además divididas en subcaracterísticas y son el resultado de los atributos o cualidades internas del software. (De los Santos, 2019)

Las métricas internas que se pueden aplicar a un producto de software, al ser métricas internas, se aplican a productos de software no ejecutables. (De los Santos, 2019)

La normativa define seis características de la aplicación, estas seis características son divididas en un número de subcaracterísticas, las cuales representan un modelo detallado para la evolución de cualquier sistema informático. (De los Santos, 2019)

El estándar ISO-9126 (o ISO/IEC 9126) establece que cualquier componente de la calidad del software puede ser descrito en términos de una o más de seis características básicas, las cuales son: funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad; cada una de las cuales se detalla a través de un conjunto de subcaracterísticas que permiten profundizar en la evaluación de la calidad de productos de software. (De los Santos, 2019)

2.10.1. Características. El estándar ISO 9126 se desarrolló con la intención de identificar los atributos clave del software de cómputo.



Figura 19. Atributos del modelo de calidad ISO 9126.

Fuente: (Borbon Ardilla, 2013)

Este sistema identifica seis atributos clave de la calidad los cuales se describen a continuación:

2.10.1.1. Funcionalidad. Es la capacidad del software de cumplir y proveer las funciones para satisfacer las necesidades explícitas e implícitas cuando es utilizado en condiciones específicas. La funcionalidad se divide en 5 criterios: (Borbon Ardilla, 2013)

a) Adecuación. La capacidad del software para proveer un adecuado conjunto de funciones que cumplan las tareas y objetivos especificados por el usuario. (Borbon Ardilla, 2013)

b) Exactitud. La capacidad del software para hacer procesos y entregar los resultados solicitados con precisión o de forma esperada. (Borbon Ardilla, 2013)

c) Interoperabilidad. La capacidad del software de interactuar con uno o más sistemas específicos. (Borbon Ardilla, 2013)

d) Conformidad. La capacidad del software de cumplir los estándares referentes a la funcionalidad. (Borbon Ardilla, 2013)

e) Seguridad. La capacidad del software para proteger la información y los datos de manera que los usuarios o los sistemas no autorizados no puedan acceder a ellos para realizar operaciones, y la capacidad de aceptar el acceso a los datos de los usuarios o sistemas autorizados. (Borbon Ardilla, 2013)

2.10.1.2. Confiabilidad. Es la capacidad del software para asegurar un nivel de funcionamiento adecuado cuando es utilizando en condiciones específicas. En este caso la confiabilidad se amplía al sostener un nivel especificado de funcionamiento y no una función requerida. La confiabilidad se divide en 3 criterios: (Borbon Ardilla, 2013)

a) Nivel de Madurez. La capacidad que tiene el software para evitar fallas cuando encuentra errores. Ejemplo, la forma como el software advierte al usuario cuando realiza operaciones en la unidad de diskett vacía, o cuando no encuentra espacio suficiente el disco duro donde esta almacenando los datos. (Borbon Ardilla, 2013)

b) Tolerancia a fallas. La capacidad que tiene el software para mantener un nivel de funcionamiento en caso de errores. (Borbon Ardilla, 2013)

c) Recuperación. La capacidad que tiene el software para restablecer su funcionamiento adecuado y recuperar los datos afectados en el caso de una falla. (Borbon Ardilla, 2013)

2.10.1.3. Usabilidad. La usabilidad es la capacidad del software de ser entendido, aprendido, y usado en forma fácil y atractiva. Algunos criterios de funcionalidad, fiabilidad y eficiencia afectan la usabilidad, pero para los propósitos de la ISO/IEC 9126 ellos no clasifican como usabilidad. La usabilidad está determinada por los usuarios finales y los usuarios indirectos del software, dirigidos a todos los ambientes, a la preparación del uso y el resultado obtenido. La usabilidad se divide en 3 criterios: (Borbon Ardilla, 2013)

a) Comprensibilidad. La capacidad que tiene el software para permitir al usuario entender si es adecuado, y de una manera fácil como ser utilizado para las tareas y las condiciones particulares de la aplicación. En este criterio se debe tener en cuenta la documentación y de las ayudas que el software entrega. (Borbon Ardilla, 2013)

b) Facilidad de Aprender. La forma como el software permite al usuario aprender su uso. También es importante considerar la documentación. (Borbon Ardilla, 2013)

c) Operabilidad. La manera como el software permite al usuario operarlo y controlarlo. (Borbon Ardilla, 2013)

2.10.1.4. Eficiencia. La eficiencia del software es la forma del desempeño adecuado, de acuerdo a al número recursos utilizados según las condiciones

planteadas. Se debe tener en cuenta otros aspectos como la configuración de hardware, el sistema operativo, entre otros. La eficiencia se divide en 2 criterios: (Borbon Ardilla, 2013)

a) Comportamiento con respecto al tiempo. Los tiempos adecuados de respuesta y procesamiento, el rendimiento cuando realiza su función en condiciones específicas. Ejemplo, ejecutar el procedimiento más complejo del software y esperar su tiempo de respuesta realizar la misma función, pero con más cantidad de registros. (Borbon Ardilla, 2013)

b) Comportamiento con respecto a recursos. La capacidad del software para utilizar cantidades y tipos adecuados de recursos cuando este funciona bajo requerimientos o condiciones establecidas. Ejemplo, los recursos humanos, el hardware, dispositivos externos. (Borbon Ardilla, 2013)

2.10.1.5. *Mantenibilidad.* La capacidad de mantenimiento es la cualidad que tiene el software para ser modificado. Incluyendo correcciones o mejoras del software, a cambios en el entorno, y especificaciones de requerimientos funcionales. El mantenimiento se divide en 4 criterios: (Borbon Ardilla, 2013)

Capacidad de análisis. La forma como el software permite diagnósticos de deficiencias o causas de fallas, o la identificación de partes modificadas. (Borbon Ardilla, 2013)

a) Capacidad de modificación. La capacidad del software para que la implementación de una modificación se pueda realizar, incluye también codificación, diseño y documentación de cambios. (Borbon Ardilla, 2013)

b) Estabilidad. La forma como el software evita efectos inesperados para modificaciones del mismo. (Borbon Ardilla, 2013)

d) Facilidad de Pruebas. La forma como el software permite realizar pruebas a las modificaciones sin poner el riesgo los datos. (Borbon Ardilla, 2013)

2.10.1.6. Portabilidad. La capacidad que tiene el software para ser trasladado de un entorno a otro. La portabilidad se divide en 4 criterios: (Borbon Ardilla, 2013)

a) Adaptabilidad. Es como el software se adapta a diferentes entornos especificados (hardware o sistemas operativos) sin que implique reacciones negativas ante el cambio. Incluye la escalabilidad de capacidad interna (Ejemplo: Campos en pantalla, tablas, volúmenes de transacciones, formatos de reporte, etc.). (Borbon Ardilla, 2013)

b) Facilidad de instalación. La facilidad del software para ser instalado en un entorno específico o por el usuario final. (Borbon Ardilla, 2013)

c) Conformidad. La capacidad que tiene el software para cumplir con los estándares relacionados a la portabilidad. (Borbon Ardilla, 2013)

d) Capacidad de reemplazo. La capacidad que tiene el software para ser reemplazado por otro software del mismo tipo, y para el mismo objetivo. Ejemplo, la reemplazabilidad de una nueva versión es importante para el usuario, la propiedad de poder migrar los datos a otro software de diferente proveedor. (Borbon Ardilla, 2013)

2.11. Estimación de Costos

La estimación de costo y esfuerzo del software nunca será una ciencia exacta. Demasiadas variables (humanas, técnicas, ambientales, políticas) pueden afectar el costo final del software y el esfuerzo aplicado para su desarrollo. Sin

embargo, la estimación del proyecto de software puede transformarse de un arte oscuro a una serie de pasos sistemáticos que proporcionen estimaciones con riesgo aceptable. Para lograr estimaciones confiables de costo y esfuerzo, surgen algunas opciones: (Pressman, 2010)

- Retrase la estimación hasta ir avanzado el proyecto (obviamente se puede lograr estimaciones 100 por ciento precisas después de que el proyecto esté completo).
- Base las estimaciones en proyectos similares que ya estén completos.
- Use técnicas de descomposición relativamente simples para generar estimaciones de costo y esfuerzo de proyecto.
- Use uno o más modelos empíricos para estimación de costo y esfuerzo de software.

Desafortunadamente, la primera opción, aunque atractiva, no es práctica. Las estimaciones de costo deben proporcionarse por anticipado. No obstante, debe reconocer que mientras más espere, más conocerá, y mientras más conozca, menos probabilidades tendrá de cometer errores serios en sus estimaciones. (Pressman, 2010)

La segunda opción puede funcionar razonablemente bien si el proyecto actual es muy similar a esfuerzos anteriores y otros factores que influyen en el proyecto (por ejemplo, el cliente, condiciones del negocio, entorno de ingeniería de software, fechas límite) son aproximadamente equivalentes. Desafortunadamente, la experiencia pasada no siempre es buen indicador de resultados futuros. (Pressman, 2010)

Las opciones restantes son enfoques viables para la estimación del proyecto de software. De manera ideal, las técnicas anotadas para cada opción deben aplicarse en cascada y cada una es una comprobación cruzada para las

demás. Las técnicas de descomposición tienen un enfoque de “divide y vencerás” para la estimación del proyecto. Al descomponer un proyecto en funciones principales y actividades de ingeniería de software relacionadas, la estimación de costo y esfuerzo puede realizarse en forma escalonada. Los modelos de estimación empírica pueden usarse para complementar las técnicas de descomposición y ofrecer un enfoque de estimación potencialmente valioso por derecho propio. (Pressman, 2010)

Las herramientas de estimación automatizadas implementan una o más técnicas de descomposición o modelos empíricos y proporcionan una atractiva opción para estimar. (Pressman, 2010)

2.12. Modelo COCOMO

A continuación, se muestran los principales conceptos concernientes al modelo constructivo de costos (COCOMO, por su acrónimo del idioma inglés CONstructive COst MOdel). COCOMO es un modelo de formulación matemática con un fuerte componente de base empírica, principalmente utilizado para estimación de costos en los proyectos de software. En la siguiente figura se muestran los principales conceptos relacionados con el modelo COCOMO. Este modelo, propuesto por Barry W. Boehm, fue introducido a finales de los años 70 y comienzos de los 80 del siglo pasado en su trabajo, *Software Engineering Economics*. Entre otras características, el modelo COCOMO está orientado a la magnitud del producto final, está basado en estimaciones matemáticas, mide el “tamaño” del proyecto y utiliza las líneas de código como unidad de medida. Dos de los aspectos fundamentales del modelo COCOMO son los submodelos y los modos de desarrollo. (Garita Gonzalez & Lizano Madriz, 2018)

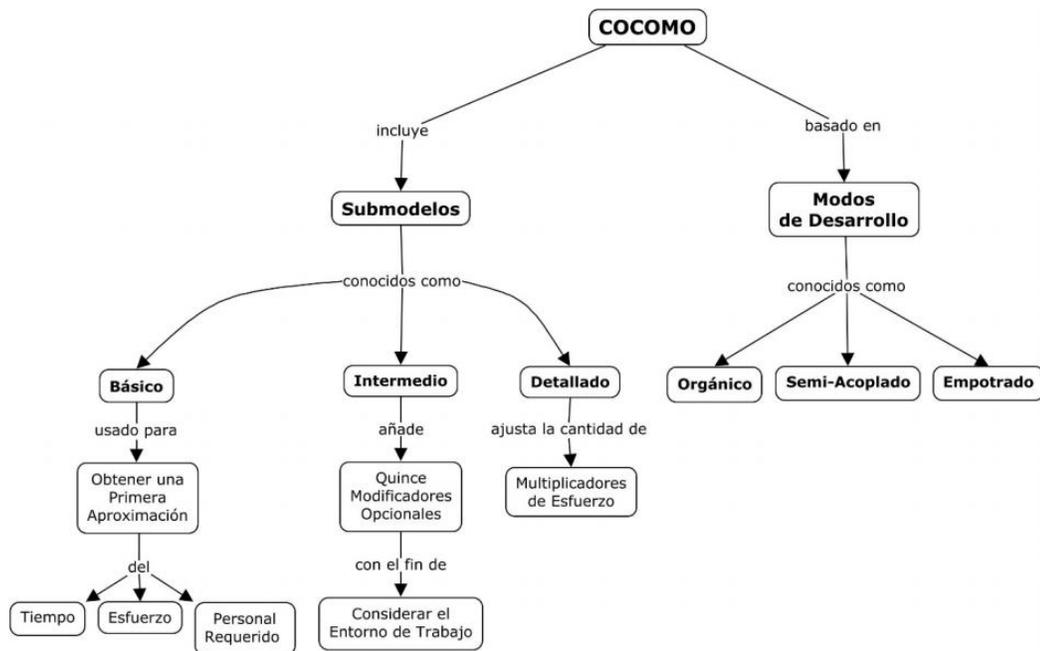


Figura 20. Conceptualización básica de COCOMO.
Fuente: (Garita Gonzalez & Lizano Madriz, 2018)

Los submodelos son tres: básico, intermedio y detallado. Por su parte, los modos de desarrollo son también tres: orgánico, semi-acoplado y empotrado.

Tabla 7. Esquema de modos de desarrollo de software

Modo de desarrollo	Requisitos	Tamaño	Complejidad	Personas	Experiencia
Orgánico	Poco rígidos	Pequeño (<50KLDC)	Pequeña	Pocas	Mucha
Semiacoplado	Poco/medio	Medio (50 a 300KLDC)	Medio	Medio	Medio
Empotrado	Alto	Grande (>300KLDC)	Alta	Alta	Poca

Fuente: (Garita Gonzalez & Lizano Madriz, 2018)

En la tabla 7 se muestra el esquema de modos de desarrollo de software con sus principales características que ayudan a elegir el tipo de modo de desarrollo para un proyecto en particular. En esta tabla, en el caso del tamaño, se consideran las líneas de código fuente del software en unidades de miles de líneas 3 de código (KLDC, por sus siglas en inglés). (Garita Gonzalez & Lizano Madriz, 2018)

Estos modos de desarrollo permiten utilizar cuatro valores constantes. En la siguiente tabla se muestran los modos de desarrollo y los valores constantes respectivos. Estos valores constantes, codificados aquí como “a”, “b”, “c” y “d”, son propuestos por el modelo COCOMO para complementar las ecuaciones de cálculo usadas en el modelo. (Garita Gonzalez & Lizano Madriz, 2018)

Tabla 8. Valores constantes por modo de desarrollo

<i>Modo de desarrollo</i>	COCOMO Básico <i>a</i>	COCOMO Intermedio <i>A</i>	<i>b</i>	<i>c</i>	<i>d</i>
Orgánico	2.4	3.2	1.05	2.50	0.38
Semiacoplado	3.0		1.12		0.35
Empotrado	3.6	2.8	1.20		0.32

Fuente: (Garita Gonzalez & Lizano Madriz, 2018)

Las ecuaciones incluidas en este artículo, son las utilizadas para los submodelos básico e intermedio. Estas ecuaciones se utilizan para calcular el esfuerzo nominal en personas/mes (E), tiempo estimado en meses (T) y personal requerido (P). No se incluyen las ecuaciones para el submodelo detallado, por razones de espacio dentro del desarrollo de la propuesta del enfoque pedagógico descrita en este trabajo. (Garita Gonzalez & Lizano Madriz, 2018)

Tabla 9. Ecuaciones por tipo de modelo COCOMO: Básico e intermedio

Ecuación	Submodelo básico	Submodelo intermedio
Esfuerzo (E)	$(E) = a * (KLDC)b$	$(E) = a * (KLDC)b * ME$
Tiempo (T)	$(T) = c * (E)d$	$(T) = c * (E)d$
Personal (P)	$(P) = E/T$	$(P) = E/T$

Fuente: (Garita Gonzalez & Lizano Madriz, 2018)

En la Tabla 9, se muestran las ecuaciones para esfuerzo nominal en personas/mes (E), tiempo estimado en meses (T) y personal requerido (P) así como los multiplicadores de esfuerzo (ME), utilizados solo en la ecuación de esfuerzo del submodelo intermedio. (Garita Gonzalez & Lizano Madriz, 2018)

Los multiplicadores de esfuerzo, utilizados en la ecuación de esfuerzo del submodelo intermedio, son quince agrupados en cuatro grandes categorías: atributos de producto, atributos de computador, atributos personales y atributos del proyecto. (Garita Gonzalez & Lizano Madriz, 2018)

Tabla 10. Ecuaciones por tipo de modelo COCOMO: Básico e intermedio

Multiplicadores de esfuerzo (ME)			Valoración					
			Muy bajo	Bajo	Nominal	Alto	Muy alto	Extr. alto
Atributos del producto								
1.	RELY	Fiabilidad requerida del software	0,75	0,88	1.00	1,15	1,40	
2.	DATA	Tamaño de la base de datos		0,94	1.00	1,08	1,16	
3.	CPLX	Complejidad del producto	0,70	0,85	1.00	1,15	1,30	
Atributos de la computadora								
4.	TIME	Restricciones del tiempo de ejecución			1.00	1,11	1,30	
5.	STOR	Restricciones del almacenamiento princ.			1.00	1,06	1,21	
6.	VIRT	Inestabilidad de la máquina virtual		0,87	1.00	1,15	1,30	
7.	TURN	Tiempo de respuesta del computador		0,87	1.00	1,07	1,15	
Atributos del personal								
8.	ACAP	Capacidad del analista	1,46	1,19	1.00	0,86	0,71	
9.	AEXP	Experiencia en la aplicación	1,29	1,13	1.00	0,91	0,82	
10.	PCAP	Capacidad de los programadores	1,42	1,17	1.00	0,86	0,70	
11.	VEXP	Experiencia en S.O. utilizado	1,21	1,10	1.00	0,90		
12.	LEXP	Experiencia en el lenguaje de progr.	1,14	1,07	1.00	0,95		
Atributos del proyecto								
13.	MODP	Uso de prácticas de programación modernas	1,24	1,10	1.00	0,91	0,82	
14.	TOOL	Uso de herramientas software	1,24	1,10	1.00	0,91	0,83	
15.	SCED	Restricciones en la duración del proy.	1,23	1,08	1.00	1,04	1,10	

Fuente: (Garita Gonzalez & Lizano Madriz, 2018)

En la tabla 10 se muestran los multiplicadores de esfuerzo. Cada uno de estos multiplicadores de esfuerzo, tiene una valoración que se clasifica en una escala de 6 valores desde “muy bajo”, “bajo”, “nominal”, “alto”, “muy alto” y “extraordinariamente alto”. Estos multiplicadores de esfuerzo ajustan el valor real del esfuerzo. Dos ejemplos nos ayudan a entender el objeto de los multiplicadores

de esfuerzo. Primero, en caso de que no se desee ajustar el esfuerzo nominal, se debe utilizar el valor nominal (es decir, el valor "1") para cualquier multiplicador de esfuerzo. Segundo; si, por el contrario, se desea hacer algún ajuste en particular al esfuerzo nominal por razones de, en este caso, poca experiencia en la aplicación, se podría utilizar el multiplicador AEXP en valor nominal "muy bajo" (1,29). (Garita Gonzalez & Lizano Madriz, 2018)

CAPÍTULO III

MARCO APLICATIVO

RESUMEN

En este capítulo, se realizara el análisis, diseño e implementación del Sistema Web de Historiales Clínicos Electrónicos personalizables por especialidad para el Colegio Médico de La Paz utilizando la metodología ágil Scrum y el método UWE para el modelado del Sistema Web en cada iteración.

3.1. Introducción

El presente capítulo tiene como finalidad describir el desarrollo del sistema web de gestión de historias clínicas, propuesto para el Colegio médico departamental de La Paz. Se hizo uso de la metodología SCRUM para la secuencia de procesos a seguir en el proyecto y la metodología UWE para el modelado de cada Sprint definido en el Product Backlog del sistema.

3.2. Diagnóstico del sistema actual

3.2.1. Descripción de la institución. El Colegio Médico Departamental de La Paz, creado por D.S. 9944 de 1 de octubre de 1971, a través de la fundación del Colegio Médico de Bolivia, es la institución de los médicos del departamento de La Paz, que realiza actividades académicas, científicas, gremiales y ético sociales, constituyéndose en un instrumento orgánico institucionalizado, sus oficinas quedan ubicadas en la calle Ballivian Nro.1266 a media cuadra de la plaza Murillo.

Como misión tienen el velar por el cumplimiento de los Estatutos y Reglamentos del Colegio Médico de Bolivia entre sus colegiados, con la finalidad de promover la excelencia en el ejercicio médico, promoviendo especialmente su desarrollo profesional, laboral, familiar, ético y social.

Su visión es ser considerada como una entidad o institución académica, científica y gremial, con reconocimiento nacional e internacional, por su liderazgo y promover la excelencia en el ejercicio de la atención médica en un entorno adecuado de trabajo, basado en valores de justicia equidad e igualdad.

3.2.1.1. Especialidades.El Colegio Médico Departamental de La Paz cuenta con alrededor de 50 médicos afiliados por especialidad y con 52 especialidades dependientes de su directiva.

Tabla 11. Especialidades dependientes al Colegio Médico Departamental de La Paz

Especialidad	Nro. de Afiliados
Anatomía patológica	35 médicos
Anestesiología	279 médicos
Angiología	3 médicos
Auditoria medica	21 médicos
Cancerología	6 médicos
Cardiología	89 médicos
Cirugía cardiovascular	9 médicos
Cirugía gastroenterológica	20 médicos
Cirugía general	305 médicos
Cirugía oncológica	9 médicos
Cirugía pediátrica	24 médicos
Cirugía plástica	42 médicos
Cirugía torácica y cardiovascular	22 médicos
Coloproctología	15 médicos
Deportología	7 médicos
Dermatología	65 médicos
Emergenciología	2 médicos
Endocrinología	26 médicos
Farmacología	2 médicos
Gastroenterología	72 médicos
Genética	4 médicos
Geriatría	10 médicos
Ginecología y obstetricia	489 médicos
Hematología y hemoterapia	35 médicos
Imagenología	1 médicos
Infectología	4 médicos
Inmuno-alergologia	3 médicos
Medicina aeronáutica y espacial	4 médicos
Medicina crítica y terapia intensiva	60 médicos

Medicina del trabajo	49 médicos
Medicina familiar	279 médicos
Medicina física y rehabilitación	53 médicos
Medicina general	8423 médicos
Medicina interna	234 médicos
Medicina legal	15 médicos
Medicina nuclear	9 médicos
Nefrología	33 médicos
Neumología	83 médicos
Neurocirugía	57 médicos
Neurología	45 médicos
Oftalmología	138 médicos
Oncología	14 médicos
Oncología clínica	6 médicos
Otorrinolaringología	66 médicos
Patología clínica	24 médicos
Pediatría	414 médicos
Psiquiatría	74 médicos
Radiología	137 médicos
Reumatología	24 médicos
Salud pública	207 médicos
Traumatología y ortopedia	189 médicos
Urología	52 médicos

Fuente: Colegio Médico Departamental de La Paz

Elaboración: Propia

3.2.1.2. Organización de la institución. El Colegio médico de La Paz se encuentra organizado como se describe en la siguiente figura.

Todos los miembros de la directiva son médicos especialistas que realizan sus funciones en diferentes establecimientos de salud públicos o privados, siendo ellos los principales interesados en que se pueda generar las historias clínicas de manera informática ya, que de esta manera podrán ayudar a promover la excelencia en el ejercicio de la atención médica siendo está más eficaz y segura.

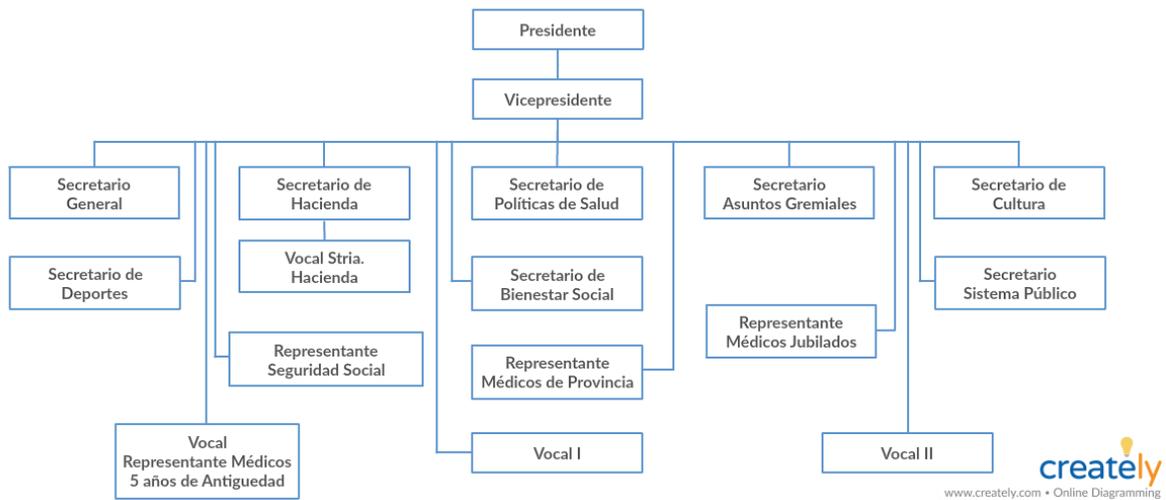


Figura 21. Organigrama del Colegio Médico Departamental de La Paz

Fuente: Colegio Médico Departamental de La Paz

Elaboración: Propia

3.2.2. Actual forma de manejo de la información. Se ha tomado como punto principal de estudio la especialidad de Medicina Interna ya que el Dr. Ricardo Landívar Vicepresidente del Colegio Médico Departamental de La Paz es la persona que brinda toda la información necesaria para la recolección de datos en los diferentes procesos en los que se maneja actualmente las historias clínicas.

El Dr. Ricardo Landívar comenta que la información generada de las Historias Clínicas actualmente es almacenada en los mismos predios de cada establecimiento de salud públicos o privados, cada establecimiento de salud cuenta con una oficina de Archivo Central donde se encuentran ordenados y ubicados con un orden correlativo; pero el manejo de una historia clínica no solo comprende el archivo como tal si no también los procesos desde la atención del paciente en el área de admisión, hasta la atención en el consultorio externo y la entrega de la receta médica.

El proceso de atención al paciente inicia cuando el paciente solicita atención médica, el encargado de admisión busca su historia clínica del paciente,

si encuentra la historia clínica procede a registrar la atención solicitada en un cuaderno (cuaderno de registro de atenciones diarias), agrega una hoja a la historia del paciente para que el médico especialista registre el anamnesis, luego el paciente se dirige a triaje en el cual se registra los signos vitales del paciente; posteriormente, se dirige al consultorio en el cual le brindara la atención requerida, que el médico especialista registrara en la historia clínica, dando validez de la atención colocando su sello y firma en la hoja contenida en la historia clínica.

Con la información recabada se procederá al modelado de negocios, en el cual se comprenderá mucho mejor los procesos con los que trabajan los médicos hacia la atención con los pacientes, mostrando así narrativamente los eventos que lo describiremos gráficamente y nos servirán como punto de partida para derivar el modelo de casos de uso. En la siguiente figura se muestra el flujograma del proceso de atención a un paciente.

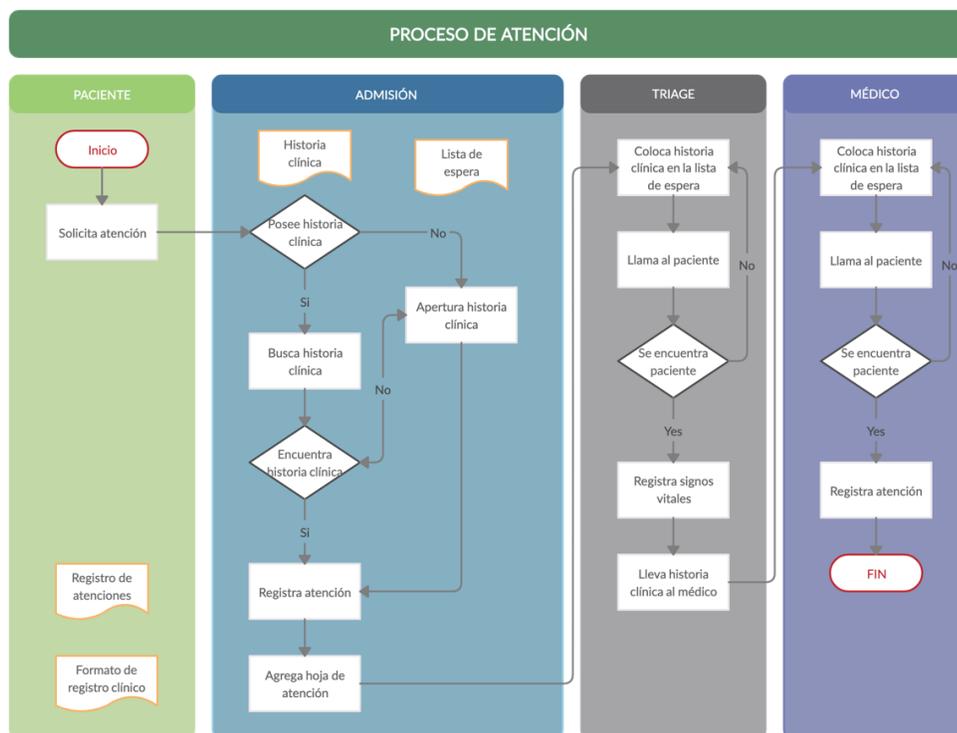


Figura 22. Flujograma del proceso de atención al paciente.

Elaboración: Propia

3.2.2.1. Lista de actores del negocio. Los actores identificados que interactuaran en los siguientes casos de uso del negocio se muestran en la tabla 12.

Tabla 12. Identificación de actores en el negocio.

<i>Actor</i>	<i>Descripción</i>
Encargado de Admisión	Persona encargada de entregar fichas de atención a los pacientes
Encargado de Triage	Persona que toma los signos vitales de los pacientes para después derivarlos al médico especialista
Médico especialista	Profesional en salud que brinda asistencia médica a pacientes
Paciente	Persona que recibe atención médica en el establecimiento de salud

Elaboración: Propia

3.2.2.2. Diagramas de casos de uso del negocio. En los siguientes diagramas de casos de uso del negocio se observan las principales operaciones que realizan en los distintos establecimientos de salud ya sean públicos como privados, comenzando con el ingreso de datos de los pacientes dándole así al paciente un número de registro del historial clínico, seguidamente el diagnóstico general realizado por el Médico especialista, este para el llenado del historial clínico.

Diagramas de casos de uso solicitar atención médica. Al requerir la atención medica en cualquier establecimiento de salud una persona en este caso identificado como paciente, debe apersonarse al establecimiento de salud y solicitar atención médica, la recepcionista verificara si hay la especialidad y si el médico especialista cuenta con disponibilidad para atenderla, si el médico esta libre se le entregara una ficha de atención al paciente.

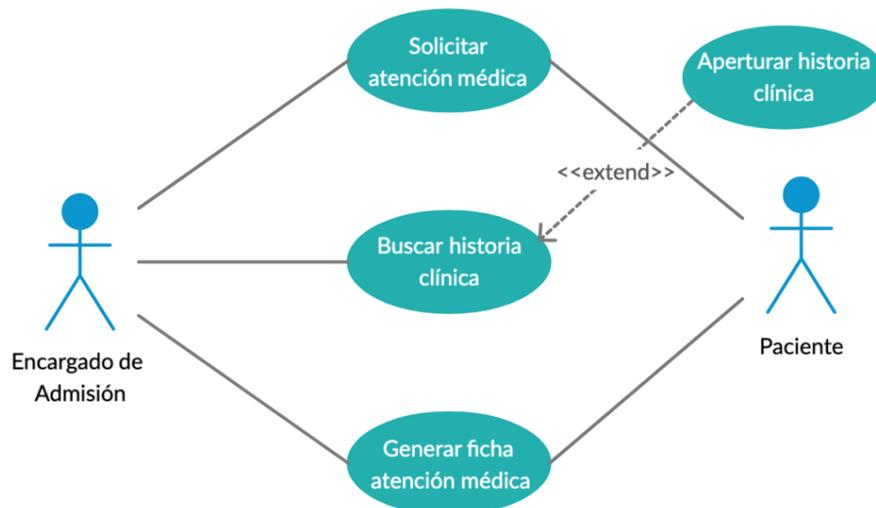


Figura 23. Caso de uso para solicitar atención médica.

Elaboración: Propia

Tabla 13. Descripción del CUN para solicitar atención médica

<i>Descripción de Caso de Uso</i>	
Caso de uso	Solicitar atención médica
Actores:	Recepcionista y Paciente
Descripción	El paciente llega a un establecimiento de salud para la atención médica para alguna especialidad con las que cuenta dicho establecimiento.
Flujo Básico	<ol style="list-style-type: none"> 1. Se acerca a recepción para solicitar ficha de atención para la especialidad que requiera. 2. La recepcionista busca si el paciente cuenta con historia clínica dentro del centro de salud 3. Se hace la entrega de la ficha para atención al paciente
Flujo alterno	<ol style="list-style-type: none"> 1. Se debe hacer la apertura de la historia clínica si el paciente visita por primera vez el establecimiento de salud

Elaboración: Propia

Diagramas de casos de uso registrar triaje. Antes de pasar a consulta con el médico especialista, el paciente pasará a la sala de triaje para que se tomen sus signos vitales (temperatura, frecuencia cardiaca, frecuencia respiratoria, presión arterial, peso, talla), la o él encargado de triaje que por lo común suele ser una enfermera, registrará todos los datos tomados del paciente y los llevará al consultorio del médico especialista.

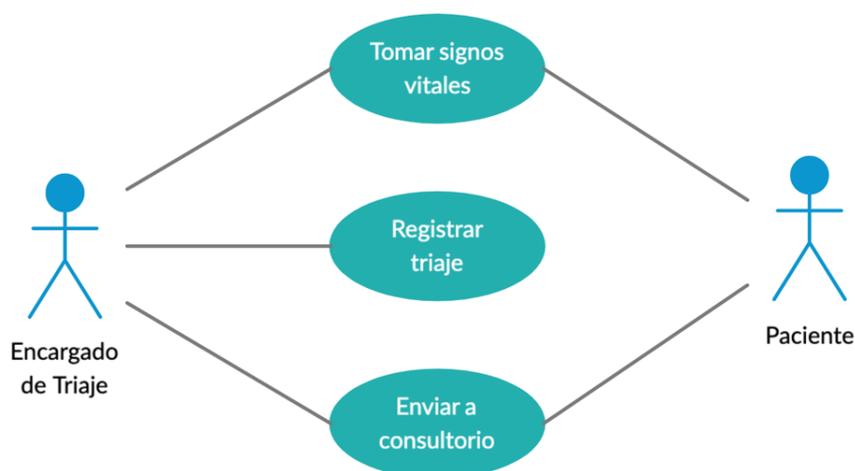


Figura 24. Caso de uso para registrar triaje.

Elaboración: Propia

Tabla 14. Descripción del CUN para registrar triaje

<i>Descripción de Caso de Uso</i>	
Caso de uso	Registrar triaje
Actores:	Encargado de triaje (enfermera) y Paciente
Descripción	Después de haber recibido su ficha a para la atención médica se envía al paciente al área de triaje, donde una enfermera le tomara los signos vitales para después enviarlo al consultorio médico del especialista.
Flujo Básico	<ol style="list-style-type: none"> 1. La enfermera llama al paciente para que pase a la sala de triaje 2. Se registran los signos vitales del paciente antes de enviarlo a consulta 3. La enfermera lleva el registro recabado del paciente al consultorio del médico especialista
Flujo alterno	<ol style="list-style-type: none"> 1. Si el paciente no se encuentra en el momento que la enfermera lo llama pasa a una lista de espera para que se lo vulva a llamar

Elaboración: Propia

Diagramas de casos de uso consulta médica.Una vez registrados los signos vitales del paciente, el mismo espera su turno para ser atendido por el médico especialista.

El proceso de consulta médica comienza con la realización de la anamnesis donde el médico pregunta al paciente el motivo de su consulta, revisa sus

antecedentes en la historia clínica si el paciente tuviera, le realiza la revisión médica para poder determinar un diagnóstico en cuanto a los síntomas. Si se tuviera que reforzar el diagnóstico el médico pide la solicitud de un procedimiento, esto quiere decir que solicita algún examen de laboratorio o estudios de imageneología. Una vez entregados los resultados por el paciente o incluso antes, el médico le explica al paciente el tratamiento a seguir para la mejora de su salud, para lo cual se le da la receta médica al paciente, para finalizar el médico reúne toda la información recabada y lo guarda en la historia clínica del paciente.

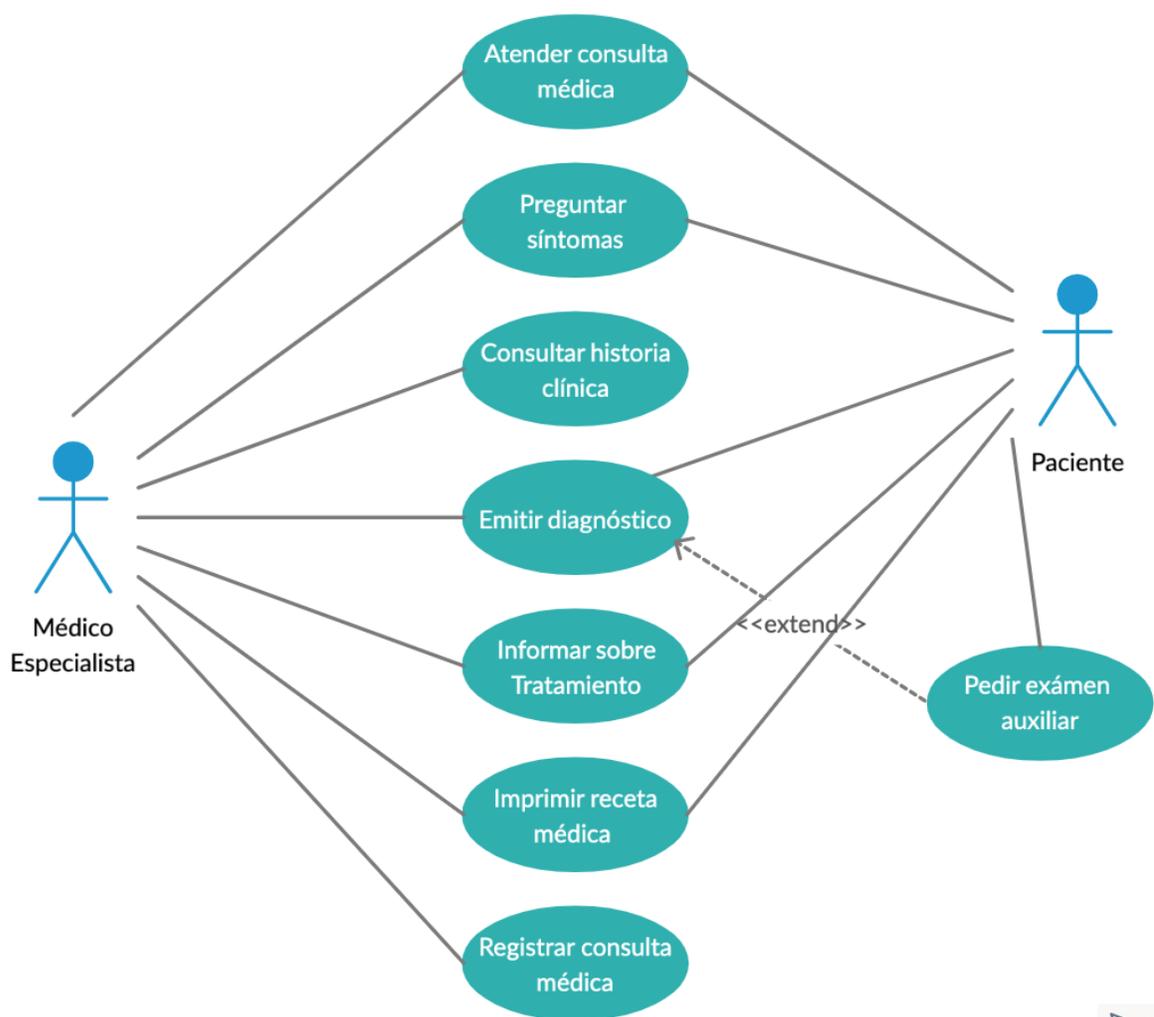


Figura 25. Caso de uso para consulta médica.
Elaboración: Propia

Tabla 15. Descripción del CUN para consulta médica

<i>Descripción de Caso de Uso</i>	
Caso de uso	Consulta médica
Actores:	Médico especialista y Paciente
Descripción	El paciente espera su turno para ser atendido por el médico especialista para que pueda darle alguna solución a los malestares que presenta.
Flujo Básico	<ol style="list-style-type: none"> 1. El médico especialista llama al paciente para que pase al consultorio médico para su atención 2. El médico pregunta malestares y realiza revisión médica 3. El médico emite diagnóstico, explica tratamiento y brinda receta médica al paciente 4. El médico registra y guarda toda la información en la historia clínica
Flujo alterno	<ol style="list-style-type: none"> 1. Si el médico necesitara sustentar su diagnóstico pedirá exámenes de laboratorio o estudios de imagenología

Elaboración: Propia

3.2.2.3. Diagramas de secuencias del negocio. Los diagramas de secuencia describirán detalladamente el proceso que sigue cada caso de uso.

Diagrama de secuencia para solicitar atención médica.

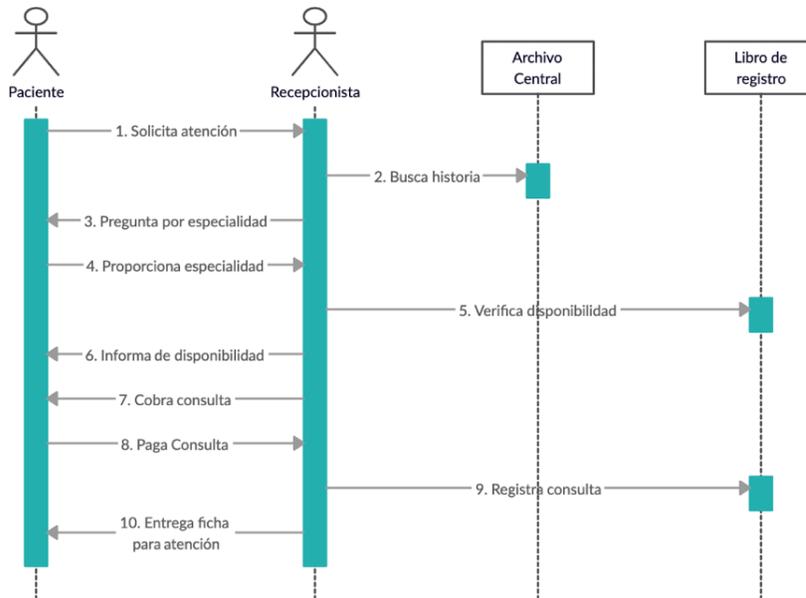


Figura 26. Diagrama de secuencia para la solicitar de atención médica.

Elaboración: Propia

Diagrama de secuencia para registrar triaje.

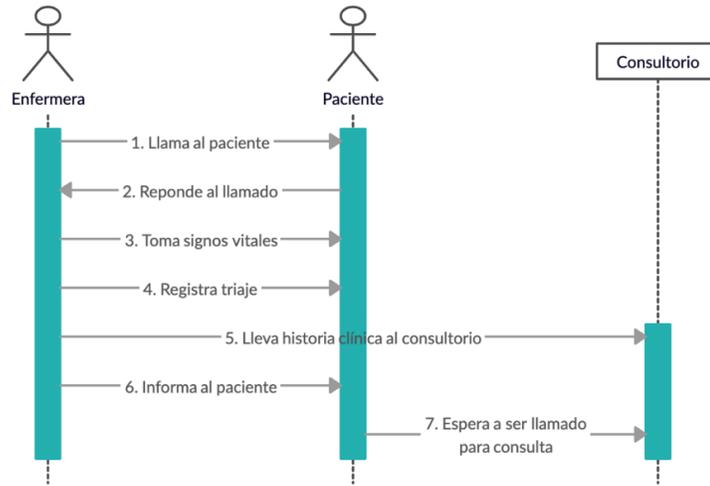


Figura 27. Diagrama de secuencia para registrar triaje.
Elaboración: Propia

Diagrama de secuencia para consulta médica.

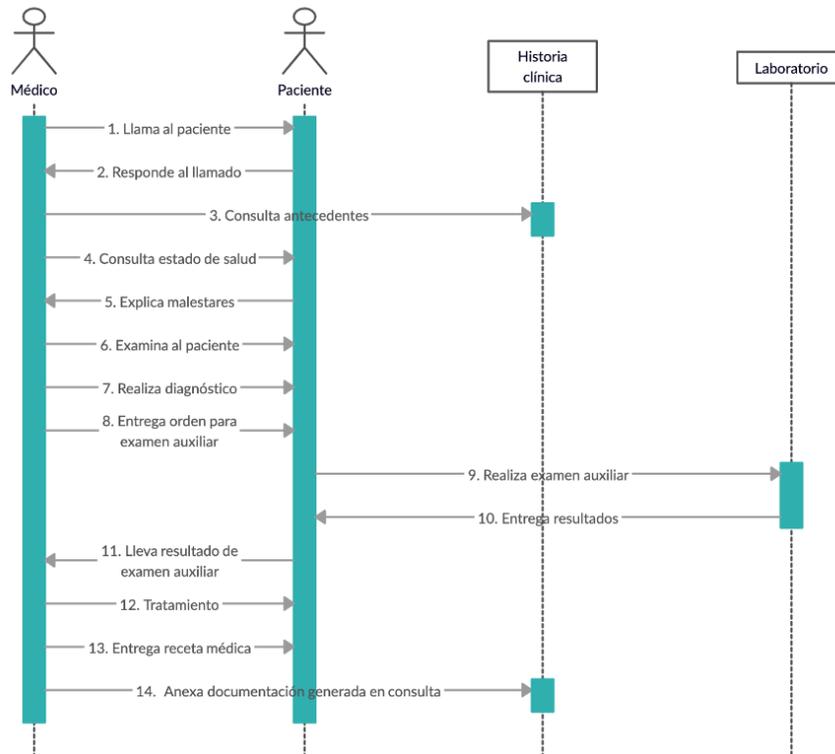


Figura 28. Diagrama de secuencia para consulta médica
Elaboración: Propia

3.2.3. Identificación de Roles Scrum.El grupo de trabajo para el presente proyecto estará conformado de la siguiente manera.

Tabla 16. Identificación de roles de Scrum

<i>Rol</i>		<i>Nombre</i>
Product Owner		Dr. Ricardo Landívar Córdova
Scrum Master		Jhenny Blanca Sinka Colmena
Scrum Team	Analista	Jhenny Blanca Sinka Colmena
	Diseñador	
	Desarrollador	
	Testeador	
Stakeholders		Colegio Médico Departamental de La Paz

Elaboración: Propia

3.3. Pre – Game (Antes del desarrollo)

Después de haber hecho el análisis de la situación actual del sistema y habiendo modelado la situación actual del negocio para un mejor entendimiento del mismo pasaremos a definir la primera fase de Scrum. En esta fase se realiza el Product Backlog con el propósito de descubrir las necesidades que competen al sistema que será desarrollado. En esta etapa utilizando la ayuda de toda la información detallada en el diagnóstico del sistema actual, se realizó una primera definición sencilla y clara de que debía cumplir el sistema.

3.3.1. Creación del product backlog (Pila de producto).Antes de crear el Product Backlog se hará la identificación de los requerimientos funcionales y no funcionales del sistema mismos que serán de ayudar para poder definir de mejor manera el Product Backlog del sistema.

3.3.1.1. Requerimientos funcionales.Para hallar los requerimientos funcionales del proyecto se optó por utilizar como herramienta la matriz de

requerimiento, la cual es efectiva para hallar los casos de uso del sistema que mejorara el proceso de atención y registro de historias clínicas.

Tabla 17. Requerimientos funcionales del sistema

<i>Id</i>	<i>Requerimiento</i>	<i>Descripción</i>	<i>Prioridad</i>
RF1	Restringir ingreso de personal	Solo personal registrado en el sistema podrá acceder al mismo	Alta
RF2	Registrar a personal del establecimiento de salud	Registrar a médicos, enfermeras y asistentes que harán uso del sistema	Alta
RF3	Inhabilitar usuarios	Quitar el acceso al sistema al personal que ya no trabaje en el establecimiento de salud	
RF4	Control por roles	Habilitación y control de usuarios para acceso al sistema	Alta
RF5	Verificación de paciente	Verificar si esta registrado o no un paciente en el establecimiento de salud	Alta
RF6	Registrar historia clínica	Si no se cuneta con una historia clínica en el establecimiento de salud se procede a apertura de la misma	Alta
RF7	Buscar historia clínica	Realizar la verificación de la historia clínica del paciente	Alta
RF8	Registra diagnóstico	Realizar el registro del diagnóstico tentativo del paciente	Alta
RF9	Registrar receta médica	Es de mucha ayuda entregar al paciente la receta médica	Alta
RF10	Registrar procedimientos	Si se requiere exámenes de laboratorio o estudios de imagenología	Alta
RF11	Registrar tratamiento	Realizar el registro del tratamiento asignado al paciente en la consulta	Alta
RF12	Registro de personal	Realizar el registro de personal que administra el consultorio	Alta
RF13	Registrar triaje	Registrar los signos vitales del paciente antes de pasar a la consulta médica	Media
RF14	Registrar número de atención médica	Realizar el registro de las personas que han sido atendidas por el establecimiento de salud	Media
RF15	Informe médico	Generar reporte de servicios brindados en el consultorio	Baja
RF16	Certificado médico	Si el paciente lo solicitara	Baja
RF17	Registrar cita médica	El médico registra una reconsulta con el paciente	Media

Elaboración: Propia

3.3.1.2. Requerimientos no funcionales. Los requerimientos no funcionales deben establecer restricciones en el producto que está siendo desarrollado, en el proceso de desarrollo y en las restricciones específicas que el producto pueda tener.

Tabla 18. Requerimientos no funcionales del sistema.

<i>Id</i>	<i>Propiedad</i>	<i>Medida</i>
RNF1	Interfaz grafica	El sistema debe funcionar con navegador web
RNF2	Repuesta de registro	El tiempo de respuesta del sistema debe ser inmediato
RNF3	Acceso	El acceso es restringido por usuario y contraseña designada por el supervisor
RNF4	Red	Debe contar con acceso a internet para su funcionamiento
RNF5	Fiabilidad	Los datos deben guardarse de manera restringida sin que otro usuario puede acceder a dicha información
RNF6	Disponibilidad	En todo momento siempre y cuando se cuente con conexión a internet

Elaboración: Propia

Habiendo realizado la identificación de los requerimientos funcionales a continuación, se muestra una lista de los módulos determinados para el Product Backlog para ser desarrollados.

Tabla 19. Listado del Product Backlog

<i>Sprint</i>	<i>Módulos</i>	<i>Estimación</i>	<i>Prioridad</i>
1	Módulo de inicio de sesión	horas	1
2	Módulo de administración de usuarios	horas	2
3	Módulo de asignación de ficha de atención	horas	3
4	Módulo de registro de triaje	horas	4
5	Módulo de gestión de historia clínica	horas	5
6	Módulo de reportes médicos	horas	6

Elaboración: Propia

3.3.2. Modelos de casos de uso general del sistema. Los casos de uso nos ayudaran a tener el punto de partida los diferentes diagramas en cada sprint.

3.3.2.1. Identificación de actores del sistema. Debemos recordar que un actor representa un conjunto de roles ya sea humano, dispositivo o cualquier sistema experto puede desempeñar cuando interactúa con un sistema.

Los actores identificados en este caso de uso son los que participaran en la mejora del proceso de registro de historias clínicas y la consulta médica.

Tabla 20. Identificación de actores en el sistema.

<i>Actor</i>	<i>Descripción</i>
 ADMINISTRADOR	Encargado de registrar nuevos usuarios en el sistema y asignarles roles de acuerdo a su función en el establecimiento de salud
 TÉCNICO ADMISIÓN	Encargado de registrar los datos de las atenciones médicas.
 TÉCNICO TRIAJE	Encargado de registrar los signos vitales del paciente.
 MÉDICO	Encargado de registrar la molestia y tratamiento en la historia clínica.
 PACIENTE	Persona que recibe la atención médica en el establecimiento de salud.

Elaboración: Propia

3.3.2.2. Diagrama general de casos de uso del sistema. En la siguiente figura se detalla el diagrama de caso de uso que se utilizará por parte de los actores identificados que interactuaran con el sistema.

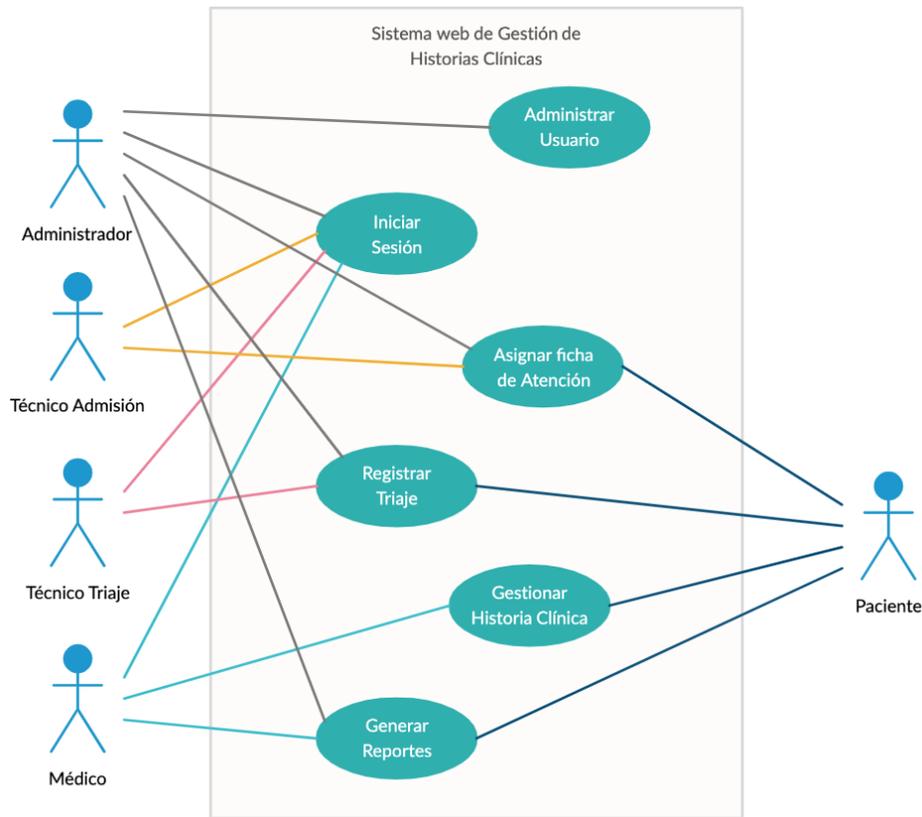


Figura 29. Diagrama de Casos de Uso general del sistema.
Elaboración: Propia

Como se muestra en el caso de uso general del sistema, son cinco los módulos propuestos para el desarrollo del sistema.

Cada una de ellas se identificó después de haber realizado el análisis situacional de los procesos que se llevan a cabo para que un paciente sea atendido un establecimiento de salud.

Cada caso de uso representa a un Sprint el cual será explicado en la fase del Game.

3.3.3. Diseño de la base de datos. Al adquirir toda la información necesaria sobre las actividades realizadas dentro de un establecimiento de salud, se realizó el diseño de la base de datos integrando todos los datos necesarios.

3.3.3.1. Diagrama entidad-relación general. Para el diseño de la base de datos se utilizó el modelo entidad-relación, en el cual se seleccionaron las entidades con sus respectivos atributos y luego se relacionaron dependiendo de cada caso.

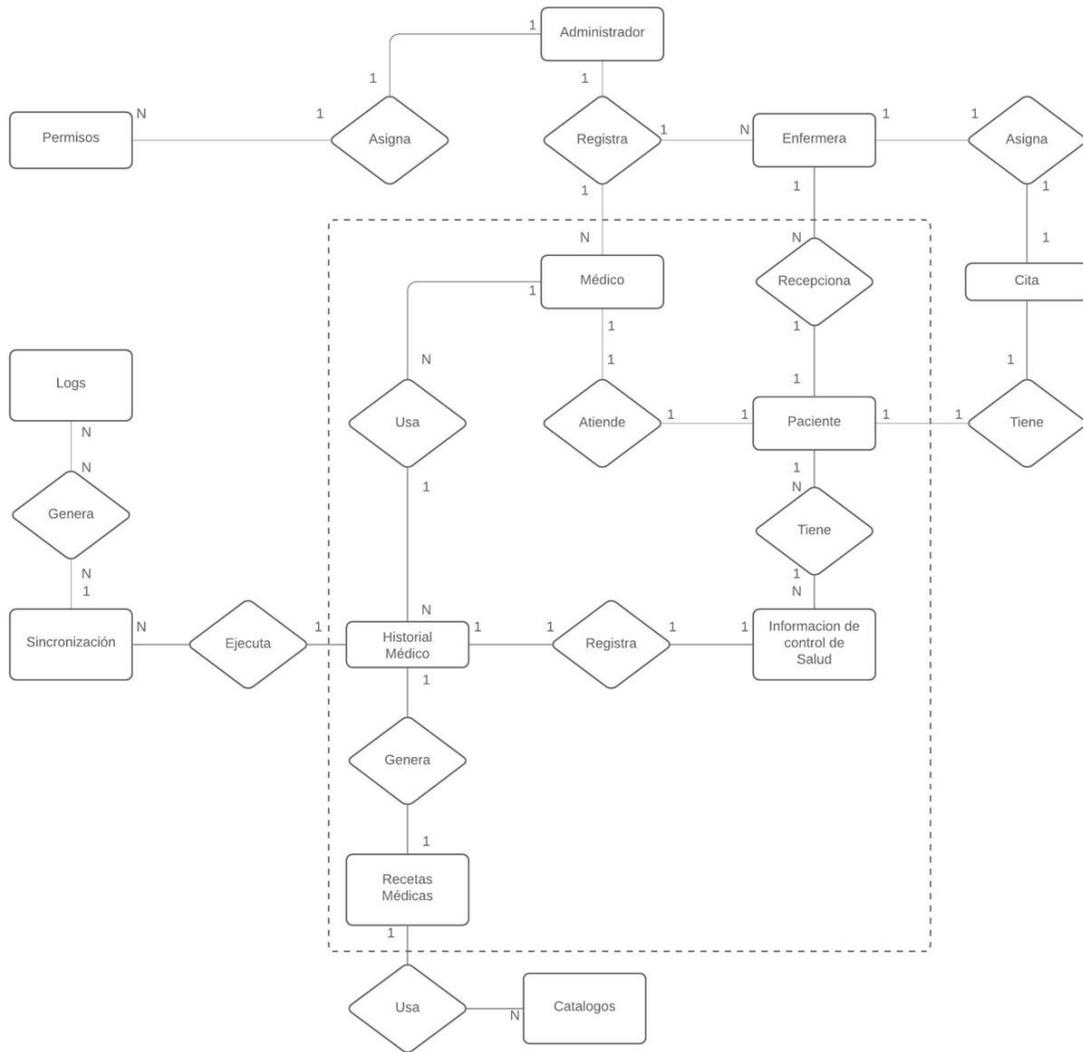


Figura 30. Diagrama Entidad – Relación.

Elaboración: Propia

3.3.3.2. Estructura de la base de datos. Se hizo la base de datos utilizando PostgreSQL y pgAdmin 4 para su entorno gráfico y se procedió a hacer cada tabla con sus respectivos atributos, claves y relaciones. Para ello se realizó la transformación de las entidades y relaciones obtenidas del modelo anterior en

tablas. En la siguiente figura se muestra la base de datos de forma general para el sistema.

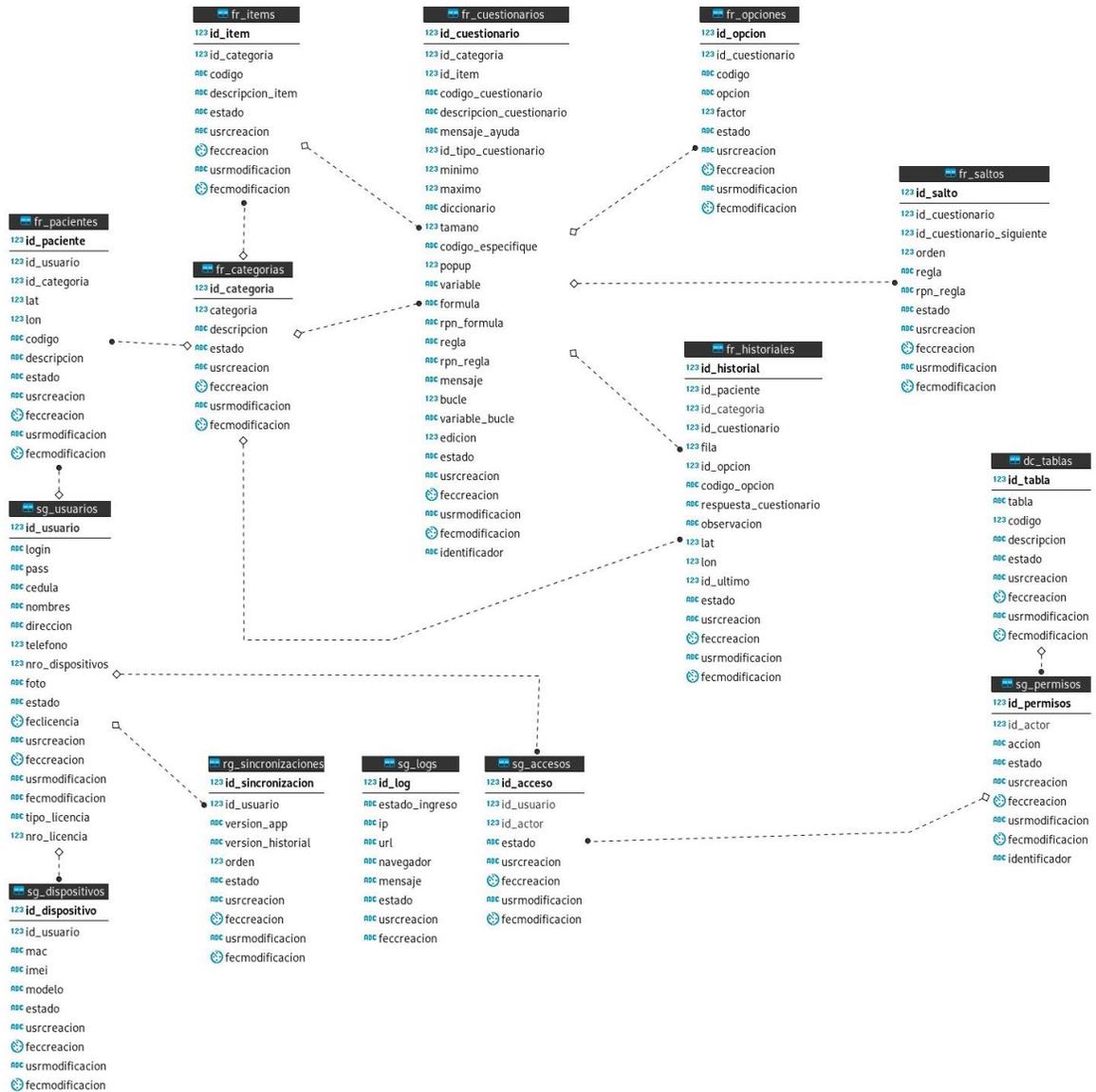


Figura 31. Estructura general de la Base de Datos.
Elaboración: Propia

3.4. Game (durante el desarrollo).

A partir de la obtención de información y el análisis de requerimientos en la Fase del PreGame, es que se continua y elabora de manera más detallada los

procesos por los cuales atravesara el desarrollo de cada uno de los módulos propuestos en el Product Backlog. En esta fase se desarrolló los “Sprints” con la ayuda de la metodología UWE para el modelado del análisis de requisitos (casos de uso), diseño navegacional y diseño de presentación. Se planearon en total seis Sprints con duración de entre 3 días y 4 semanas de acuerdo su complejidad, cada uno de los sprints tiene tareas específicas a ejecutarse con el objetivo de entregar una versión funcional del prototipo al final de cada sprint.

3.4.1. SPRINT 1: Módulo de inicio de sesión.

3.4.1.1. Planificación del Sprint 1. El objetivo del primer Sprint es desarrollar el inicio de sesión para los usuarios que tendrán acceso al sistema, este deberá presentar las funcionalidades descritas en el Sprint Backlog en esta primera iteración.

Tabla 21. Sprint Backlog: Inicio de Sesión

SPRINT 1: Módulo de inicio de sesión			
Número de Sprint:		1	
Inicio del Sprint:		24 de Julio de 2019	
Duración del Sprint:		3 días	
Id	Tareas	Tipo	Estado
1.1	Planificación y análisis de los requerimientos	Planificación	Terminado
1.2	Diseño navegacional	Diseño	Terminado
1.3	Diseño de Presentación	Diseño	Terminado
1.4	Diseño de la interfaz gráfica principal del sistema	Desarrollo	Terminado
1.5	Diseño de la interfaz gráfica para inicio de sesión	Desarrollo	Terminado
1.6	Validación de usuario y password	Desarrollo	Terminado

Elaboración: Propia

3.4.1.2. Diagrama de caso de uso del Sprint 1. En la figura 16 se puede observar el inicio de sesión que tienen los usuarios y su posterior validación por motivos de seguridad para ingresar al sistema.

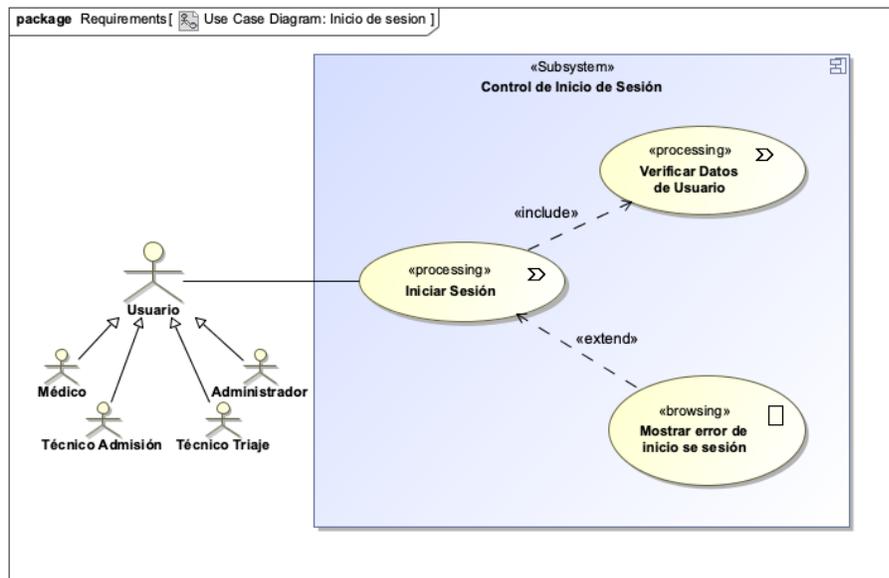


Figura 32. Diagrama de Caso de Uso: Inicio de Sesión.
Elaboración: Propia

En la tabla siguiente tabla observamos con más detalle las funciones que cumple el usuario para ingresar al sistema.

Tabla 22. Descripción del Caso de Uso: Registro de Usuario

Nro. 1	Nombre del Caso de Uso: Iniciar sesión
Actores:	Administrador, Técnico Admisión, Técnico Triaje, Administrador
Descripción:	Permite a los usuarios el ingreso a sus perfiles personales
Precondición:	Los usuarios deben estar previamente registrados en el sistema y tener asignado un rol de funciones
Flujo Básico:	<ol style="list-style-type: none"> 1. Ingresar a la dirección url del sistemas 2. El usuario debe Ingresar usuario y contraseña 3. El sistema verificara si es usuario cuenta con credenciales validas para ingresar al sistema 4. Ingresar al perfil asignado
Flujo alterno:	<ol style="list-style-type: none"> 1. Si el usuario o contraseña no son correctos se visualizara un mensaje error
Postcondición:	Se presenta el menú personalizado

Elaboración: Propia

3.4.1.3. Diseño navegacional del Sprint 1. El diseño de navegación muestra como el usuario antes de acceder a su perfil debe interactuar con el sistema.

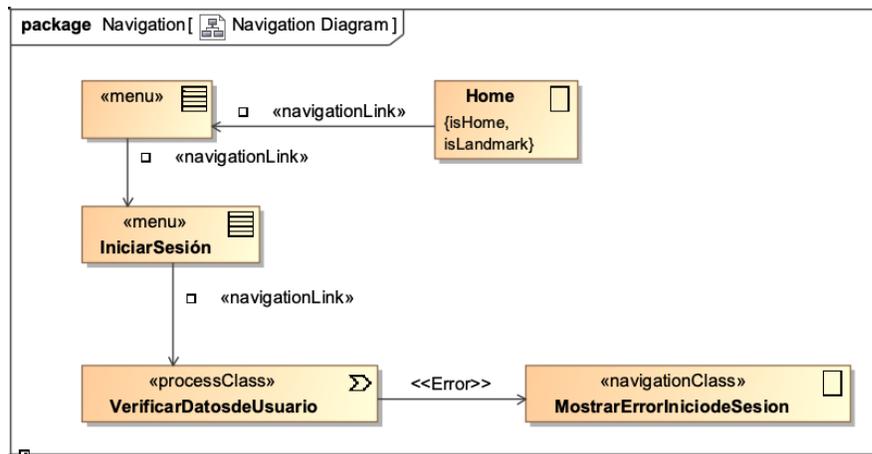


Figura 33. Diagrama de Navegación: Inicio de Sesión.
Elaboración: Propia

3.4.1.4. Diseño Presentación del Sprint 1. El diseño de presentación para el inicio de sesión solo tendrá una pantalla, en la misma aparecerá el mensaje de error si los datos que se ingresen son incorrectos y direccionara al perfil del usuario si los datos ingresados son correctos.

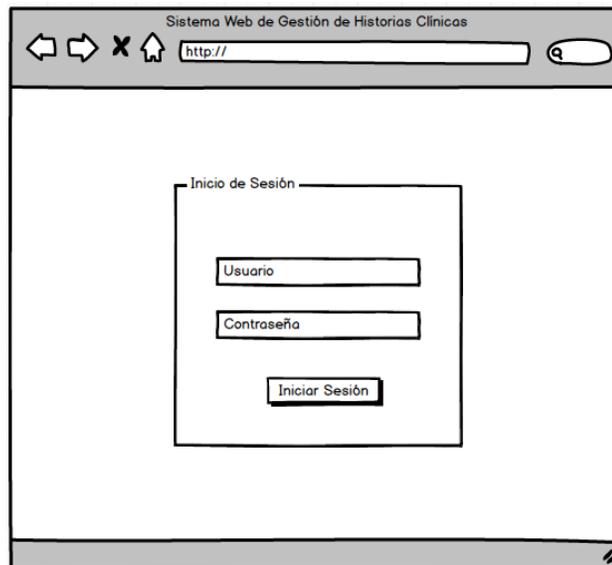


Figura 34. Diagrama de Navegación: Inicio de Sesión.
Elaboración: Propia

3.4.1.5. Pantallas del Sprint 1. A continuación se muestra la captura de pantalla para el inicio de sesión en el sistema.



Figura 35. Diagrama de Navegación: Inicio de Sesión.
Elaboración: Propia

3.4.1.6. Prueba unitaria del Sprint 1. La prueba unitaria a continuación es para comprobar que el modulo funcione correctamente.

Tabla 23. Prueba unitaria: Inicio de Sesión

Id	Descripción de caso de Prueba	Resultado
1	El sistema valida correctamente el usuario y contraseña del usuario.	Cumple
2	El sistema muestra un mensaje de error cuando algunos de los campos ingresados en el formulario de validación son incorrectos.	Cumple

Elaboración: Propia

3.4.2. SPRINT 2: Módulo de administración de usuario.

3.4.2.1. Planificación del Sprint 2. En el segundo Sprint se desarrolló la administración de los usuarios, donde el usuario con rol de administrador puede crear, modificar e inhabilitar a los usuarios que interactuaran con el sistema, además el usuario administrador tendrá a su cargo más funciones mismas que se describirán en los casos de uso de los demás sprint, vale recalcar que la única funcionalidad a la que no tendrá acceso el usuario administrador es a la gestión de historia clínica ya es esta solo será administrada por el médico que tenga a cargo a los pacientes.

Tabla 24. Backlog: Administración de usuarios

SPRINT 2: Módulo de administración de usuarios			
Número de Sprint:		2	
Inicio del Sprint:		29 de Julio de 2019	
Duración del Sprint:		5 días	
Id	Tareas	Tipo	Estado
2.1	Planificación y análisis de los requerimientos	Planificación	Terminado
2.2	Diseño navegacional	Diseño	Terminado
2.3	Diseño de Presentación	Diseño	Terminado
2.4	Diseño de la interfaz gráfica para la administración	Desarrollo	Terminado
2.5	Registrar nuevo usuario	Desarrollo	Terminado
2.6	Asignar rol al usuario registrado	Desarrollo	Terminado
2.7	Modificar usuario	Desarrollo	Terminado
2.8	Inhabilitar usuario registrado	Desarrollo	Terminado

Elaboración: Propia

3.4.2.2. Diagrama de caso de uso del Sprint 2. Para la gestión de usuarios primeramente el administrador debe ingresar al sistema web, dentro del perfil de administración se encontrarán cuatro opciones mismas que se muestran en el diagrama de casos de uso, seguidamente se llenará los datos del usuario se ese fuera el caso o modificar por su defecto.

A continuación, mostramos el caso de uso que refleja lo que el usuario podrá hacer en el caso de la administración de usuarios:

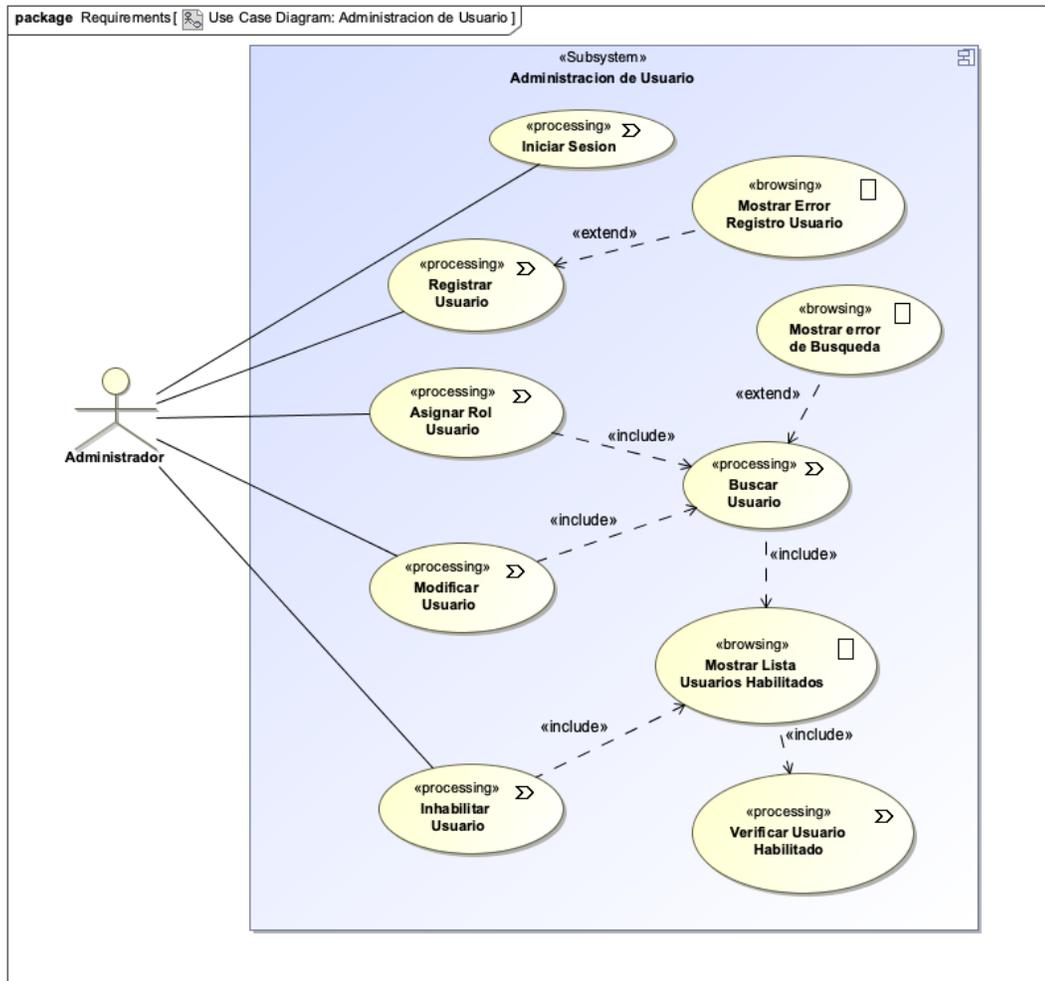


Figura 36. Diagrama de Caso de Uso: Administración de Usuarios.
Elaboración: Propia

Los casos de uso descritos en el diagrama se explicarán con más detalles en las siguientes tablas.

Tabla 25. Descripción del Caso de Uso: Registro de Usuario.

Nro. 2	Nombre del Caso de Uso: Registrar usuario
Actores:	Administrador
Descripción:	Permite al usuario administrador registrar a una nueva persona en el sistema, al cual el sistema automáticamente creara las credenciales de acceso para el nuevo usuario
Precondición:	El usuario no debe estar registrado en el sistema

Flujo Básico:	<ol style="list-style-type: none"> 1. Ingresar al sistema 2. Abrir la pestaña de nuevo usuario 3. Verificar los datos ingresados antes de guardar 4. Guardar datos del nuevo usuario 5. Entregar credenciales al nuevo usuario (usuario y contraseña)
Flujo alterno:	<ol style="list-style-type: none"> 1. Si los datos no son registrados correctamente o el usuario ya está registrado en el sistema el mismo mostrara un mensaje de error en el registro.
Postcondición:	Se le asigna un perfil diferenciado por rol de usuario
Elaboración: Propia	

Tabla 26. Descripción de Caso de Uso: Asignar Rol

Nro. 2	Nombre del Caso de Uso: Registrar usuario
Actores:	Administrador
Descripción:	Permite asignar un rol al usuario, de esta manera se le asigna tareas especializadas a actor del sistema
Precondición:	El usuario debe estar registrado en el sistema
Flujo Básico:	<ol style="list-style-type: none"> 1. Seleccionar usuario a asignar rol 2. Asignar rol 3. Guardar cambios del nuevo usuario
Postcondición:	El usuario tiene asignadas tareas de acuerdo a su rol
Elaboración: Propia	

Tabla 27. Descripción de Caso de Uso: Modificar e inhabilitar usuario

Nro. 2	Nombre del Caso de Uso: Registrar usuario
Actores:	Administrador
Descripción:	Permite modificar o inhabilitar a usuarios
Precondición:	Estar registrado en el sistema y contar con un rol asignado
Flujo Básico:	<ol style="list-style-type: none"> 1. Se muestra un listado de todos los usuarios registrados y habilitados en sus funciones 2. Seleccionar usuario a modificar o inhabilitar 3. Verificar antes de guardar datos de modificación 4. Guardar cambios
Flujo alterno:	<ol style="list-style-type: none"> 1. Para ambos casos se puede realizar la búsqueda del usuario por los diferentes campos que parecen en la tabla del listado de usuarios del sistema
Postcondición:	El usuario tiene asignadas tareas de acuerdo a su rol
Elaboración: Propia	

3.4.2.3. Diseño navegacional del Sprint 2. El diagrama de navegación muestra el flujo de navegación de los diferentes casos de uso para la administración de usuarios descritos anteriormente.

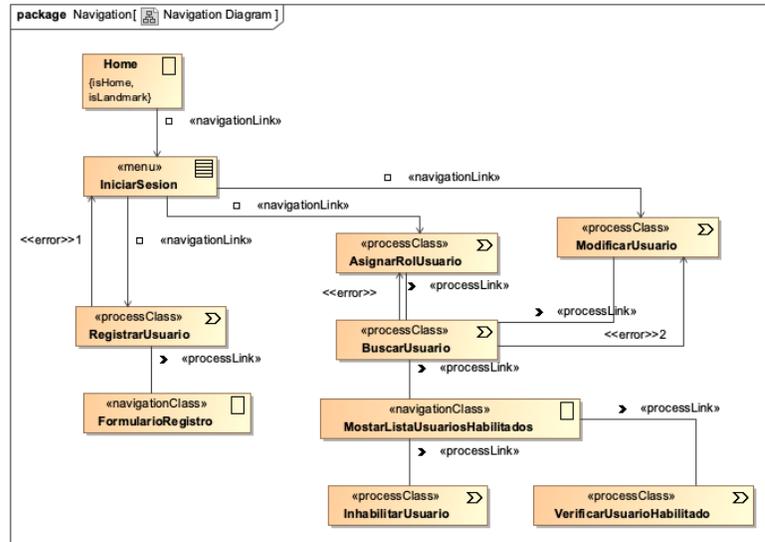


Figura 37. Diagrama de Navegación: Administración de Usuario.
Elaboración: Propia

3.4.2.4. Diseño de presentación del Sprint 2.

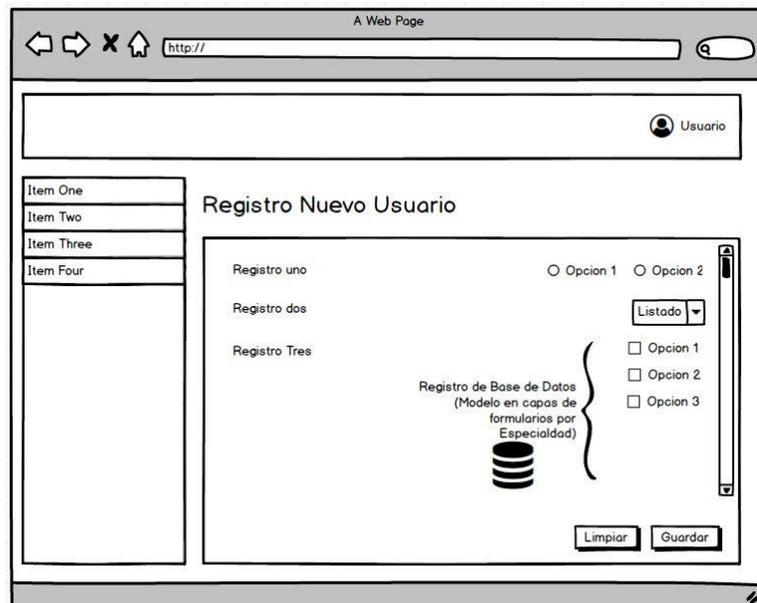


Figura 38. Diagrama de Presentación: Registro nuevo usuario.
Elaboración: Propia

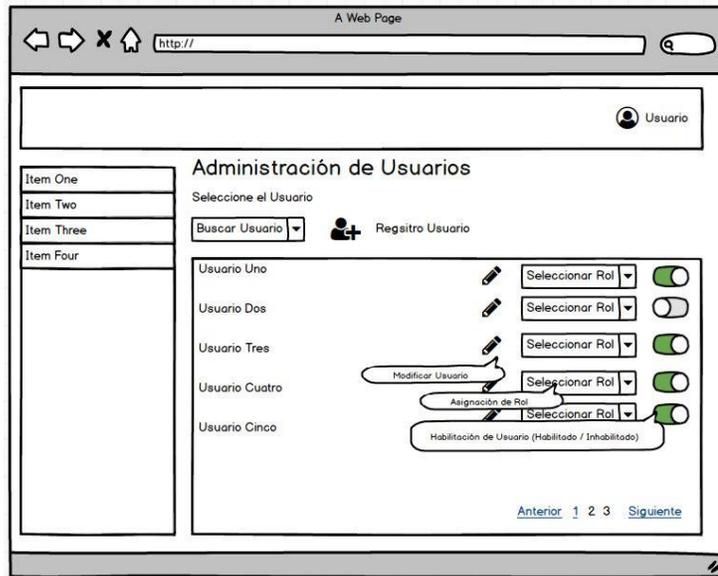


Figura 39. Diagrama de Presentación: Administración de Usuario.
Elaboración: Propia

3.4.2.5. Pantallas del Sprint 2.A continuación se muestra las captura de pantalla para el administrador.

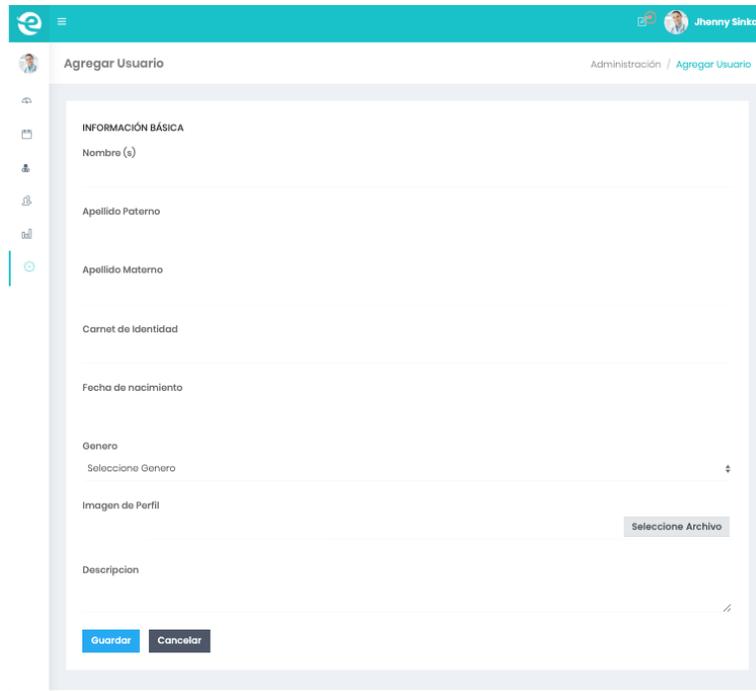


Figura 40. Captura de pantalla de registro de nuevo usuario
Elaboración: Propia

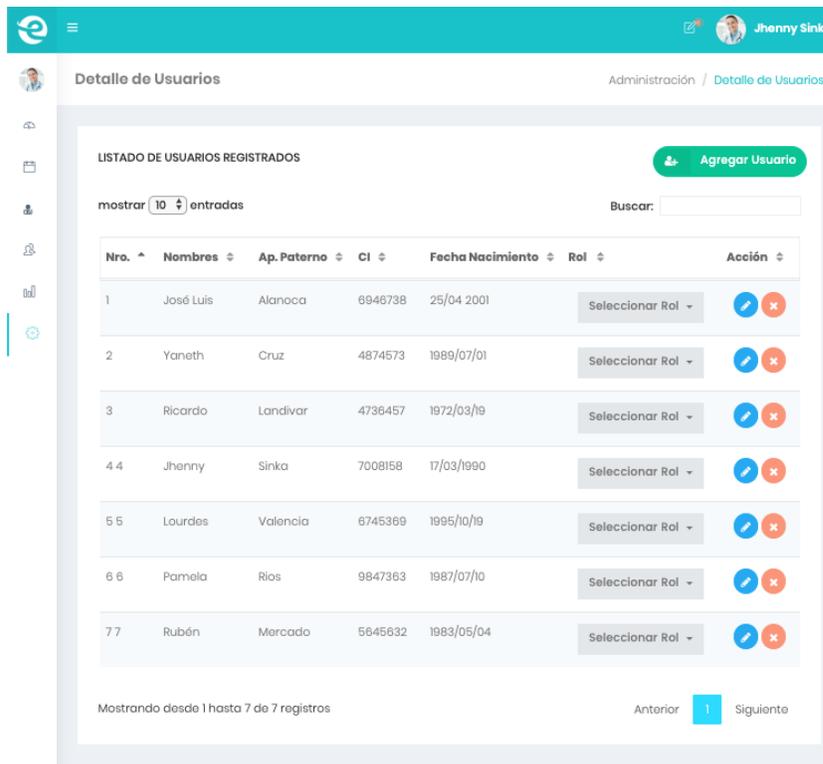


Figura 41. Captura de pantalla de detalle de usuarios.

Elaboración: Propia

3.4.2.6. Prueba unitaria del Sprint 2. La prueba unitaria a continuación es para comprobar que el modulo funcione correctamente.

Tabla 28. Prueba unitaria: Administración de Usuarios

Id	Descripción de caso de Prueba	Resultado
1	El sistema guarda los datos del usuario registrado	Cumple
2	Se puede realizar la asignación al usuario ya registrado en el sistema	Cumple
	El sistema permite la modificación de usuarios registrados	Cumple

Elaboración: Propia

3.4.3. SPRINT 3: Módulo de Asignación de ficha.

3.4.3.1. Planificación del Sprint 3. En el tercer Sprint se realizará el registro del paciente en el caso de que sea nuevo, o la búsqueda del mismo en el

caso de que sea consecutivo y se le asignará un turno para ser atendido por el médico especialista, pero antes de ir a consulta debe pasar al área de triaje.

Tabla 29. Sprint Backlog: Asignar ficha de atención

SPRINT 3: Módulo asignación de ficha			
Número de Sprint:	3		
Inicio del Sprint:	5 de agosto de 2019		
Duración del Sprint:	10 días		
Id	Tareas	Tipo	Estado
3.1	Planificación y análisis de los requerimientos	Planificación	Terminado
3.2	Diseño navegacional	Diseño	Terminado
3.3	Diseño de Presentación	Diseño	Terminado
3.4	Ver disponibilidad del médico	Desarrollo	Terminado
3.5	Buscar registro del paciente	Desarrollo	Terminado
3.6	Registrar paciente	Desarrollo	Terminado
3.7	Asignar a especialidad	Desarrollo	Terminado
3.8	Entregar número de atención	Desarrollo	Terminado

Elaboración: Propia

3.4.3.2. Diagrama de caso de uso del Sprint 3. Para la asignación de ficha de atención el paciente debe estar registrado en el sistema y tener aperturada su historia clínica en el establecimiento de salud, en el caso de que no tenga este registrado en el sistema el personal de Admisión pedirá el documento de identificación para poder realizar el correspondiente registro y aperturar su historia clínica para que el médico especialista pueda atenderla y se le entregara la ficha correspondiente para su atención.

A continuación, se muestra el diagrama de casos de uso lo descrito en el párrafo anterior:

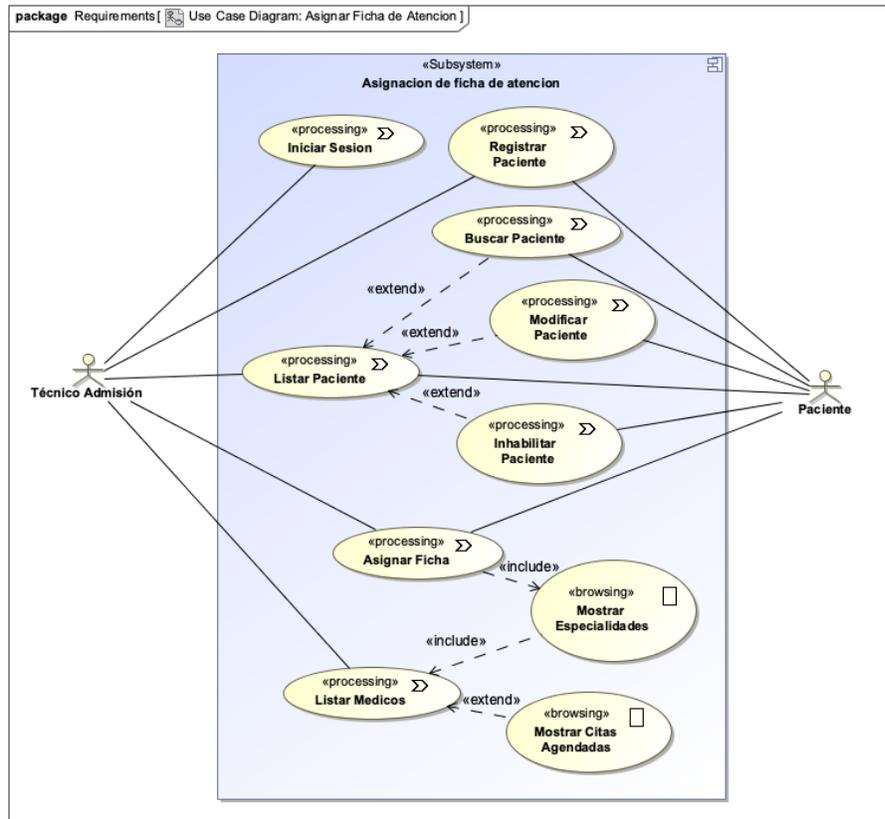


Figura 42. Diagrama de Caso de Uso: Asignación de Turno.

Elaboración: Propia

En la siguiente tabla se muestra de descripción de los casos de uso de la asignación de ficha de atención.

Tabla 30. Descripción del Caso de Uso: Registro de Paciente

Nro. 1	Nombre del Caso de Uso: Registro de Paciente
Actores:	Técnico Admisión
Descripción:	Permite al usuario con rol de técnico de admisión poder hacer el registro de un nuevo paciente
Precondición:	El paciente no debe estar registrado en el sistema y debe portar su documento de identidad
Flujo Básico:	<ol style="list-style-type: none"> 1. Seleccionar la pestaña de registro de paciente 2. Llenar formulario con los datos del paciente 3. Verificar los datos y guardar registro
Postcondición:	El paciente ya tiene una historia clínica aperturada en el establecimiento de salud

Elaboración: Propia

Tabla 31. Descripción de Caso de Uso: Listar Pacientes

Nro. 2	<i>Nombre del Caso de Uso: Listar Pacientes</i>
Actores: Técnico Admisión	
Descripción: Permite listar a todos los pacientes habilitados en el sistema	
Precondición: Estar registrado en el sistema	
Flujo Básico: 1. Seleccionar la opción de listado de pacientes	
Flujo alternativo: 1. En la misma tabla del listado de pacientes se puede realizar la búsqueda por cualquiera de los campos mostrados en la tabla 2. Se puede realizar la modificación de los datos del paciente, pero solo los datos personales, no se podrán modificar datos ingresados por el médico especialista.	
Postcondición: Paciente es encontrado en el sistema	
Elaboración: Propia	

Tabla 32. Descripción de Caso de Uso: Asignar ficha

Nro. 3	<i>Nombre del Caso de Uso: Asignar</i>
Actores: Técnico Admisión	
Descripción: Permite asignar un turno al paciente que solicita atención médica	
Precondición: El paciente debe estar registrado en el sistema	
Flujo Básico: 1. Seleccionar la opción de asignar ficha 2. Seleccionar especialidad médica y al médico 3. Asignar al paciente el turno correspondiente	
Flujo alternativo: 1. Antes de hacer la asignación el técnico de admisión debe verificar si el médico esta de turno y si cuenta con espacio para poder atender al paciente	
Postcondición: Paciente encontrado en el sistema	
Elaboración: Propia	

Tabla 33. Descripción de Caso de Uso: Listar Médicos

Nro. 4	<i>Nombre del Caso de Uso: Listar Médicos</i>
Actores: Técnico Admisión	
Descripción: Permite visualizar el listado de médicos en turno y a cuantos pacientes esta atendiendo	
Flujo Básico: 1. Seleccionar la opción listar médicos	
Postcondición: Listado de médicos actualizado	
Elaboración: Propia	

3.4.3.3. Diseño navegacional del Sprint 3. El diagrama de navegación muestra las opciones dentro del sistema a las cuales puede acceder el técnico de admisión

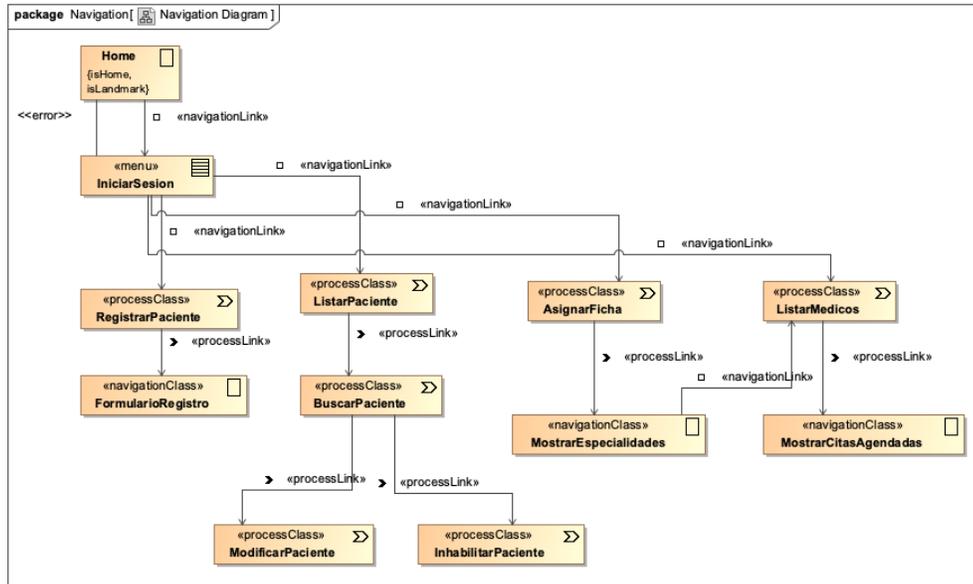


Figura 43. Diagrama de Navegación: Asignar Ficha de Atención.
Elaboración: Propia

3.4.3.4. Diseño de presentación del Sprint 3.

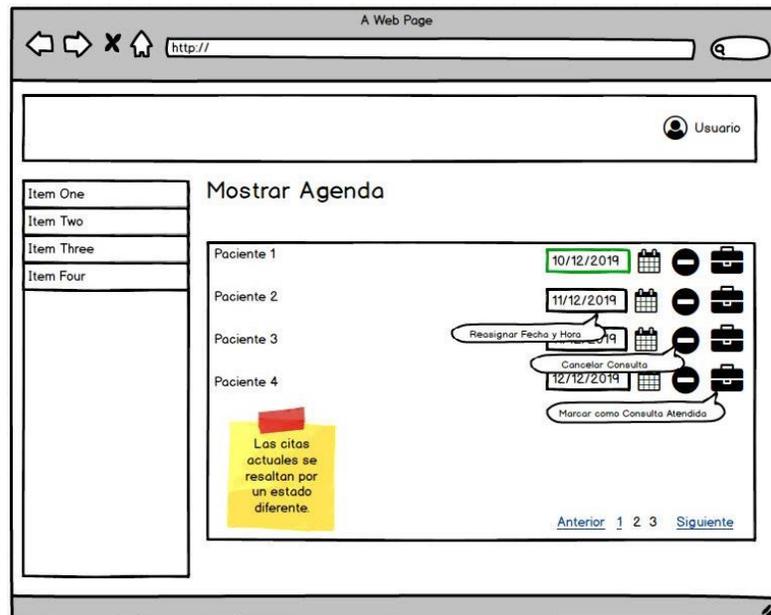


Figura 44. Diagrama de Presentación: Mostrar agenda.
Elaboración: Propia

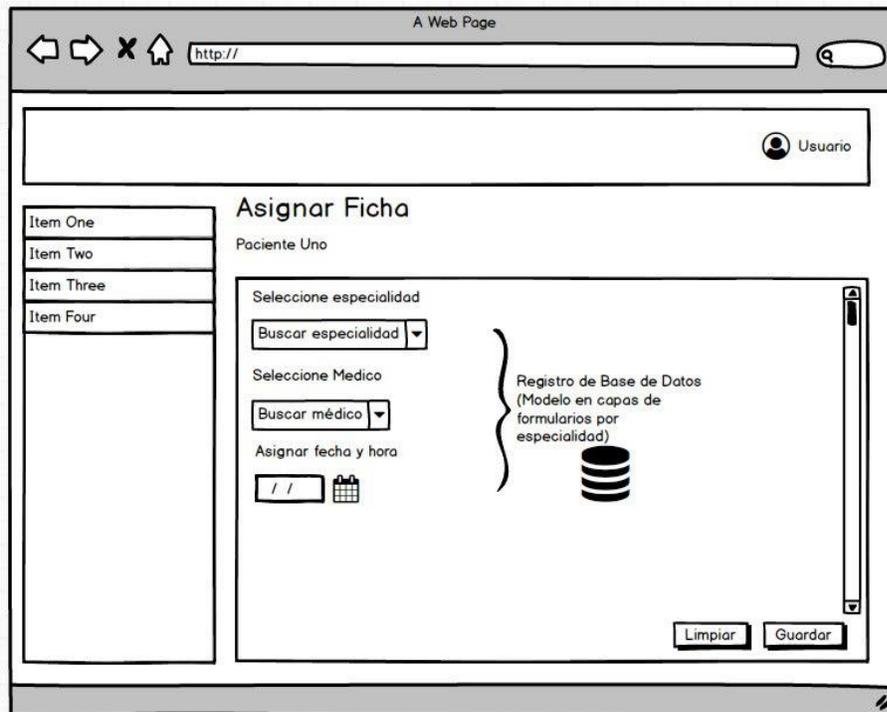


Figura 45. Diagrama de Presentación: Asignar ficha
Elaboración: Propia

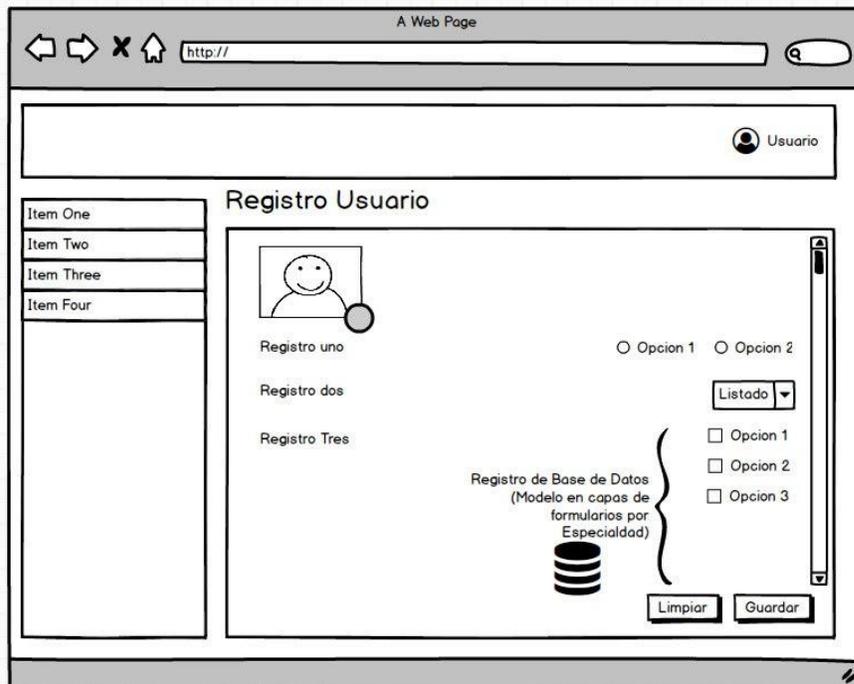


Figura 46. Diagrama de Presentación: Registrar paciente.
Elaboración: Propia

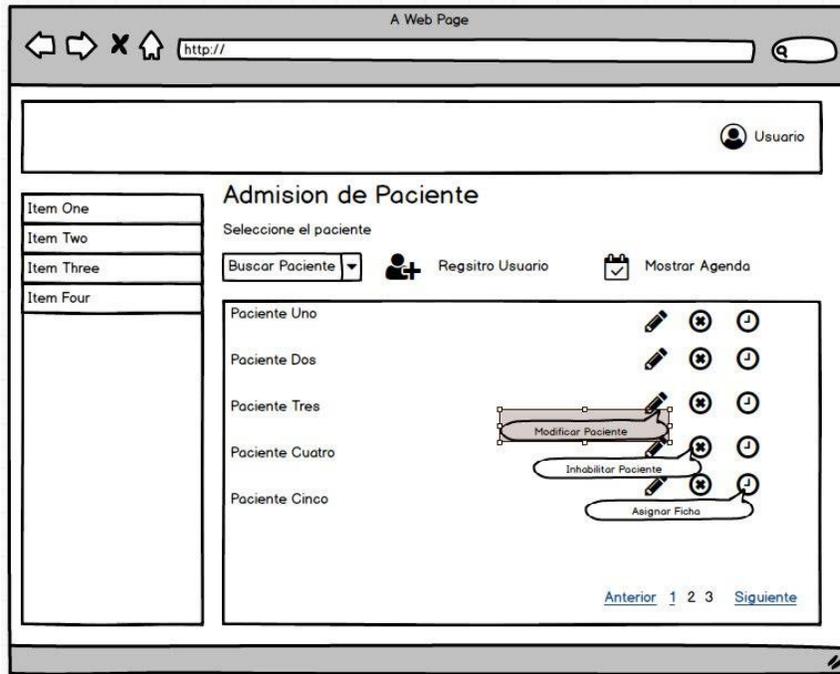


Figura 47. Diagrama de Presentación: Admisión de Paciente
Elaboración: Propia

3.4.3.5. Pantallas del Sprint 3. A continuación se muestra las captura de pantalla para el administrador.

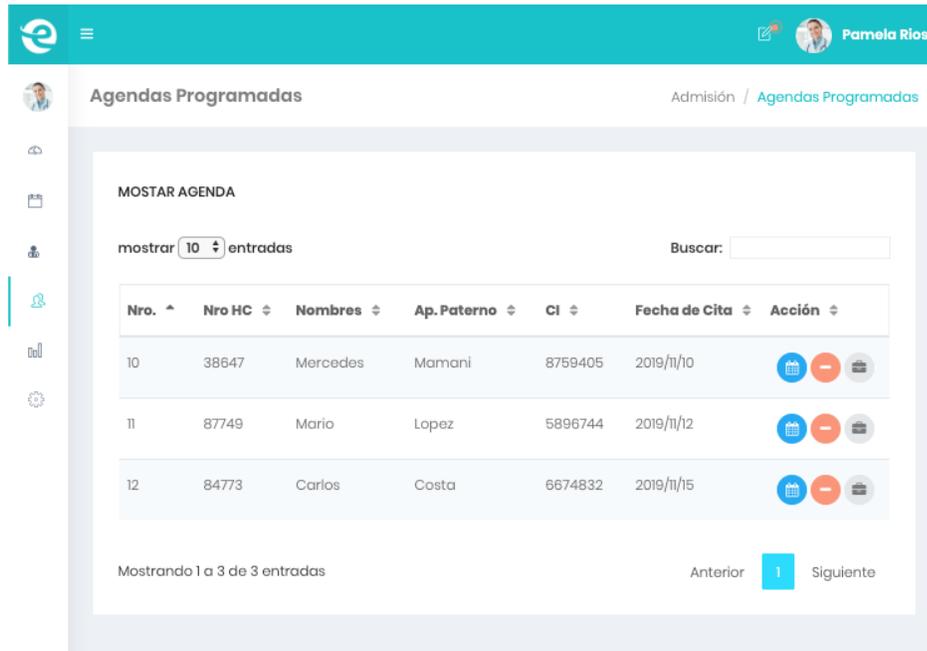


Figura 48. Diagrama de Presentación: Admisión de Paciente
Elaboración: Propia

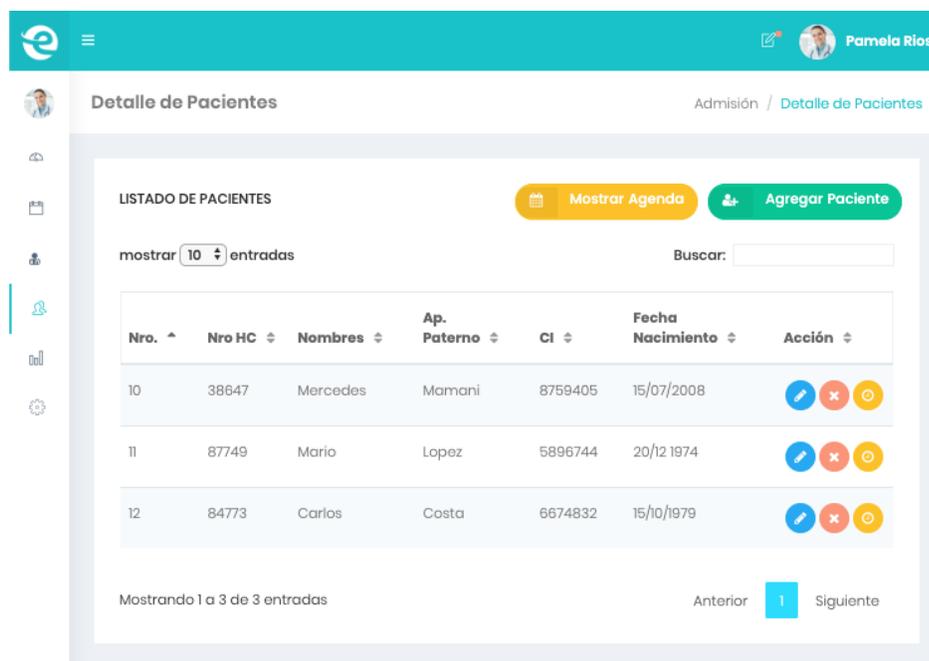


Figura 49. Diagrama de Presentación: Admisión de Paciente
Elaboración: Propia

3.4.3.6. Prueba unitaria del Sprint 3. La prueba unitaria a continuación es para comprobar que el modulo funcione correctamente

Tabla 34. Prueba unitaria: Asignar ficha de atención

Id	Descripción de caso de Prueba	Resultado
1	El sistema guarda correctamente a los pacientes registrados.	Cumple
2	Se puede realizar una búsqueda rápida y sencilla de los pacientes.	Cumple
3	Se modifica correctamente a los pacientes.	Cumple
4	Se muestra una lista actualizada de los turnos y agendas de los médicos	Cumple

Elaboración: Propia

3.4.4. SPRINT 4: Módulo de Registro de Triage.

3.4.4.1. Planificación del Sprint 4. En el cuarto Sprint se realizara el registro del triaje es decir el registro de los signos vitales de paciente como ser: peso, talla, temperatura, frecuencia cardiaca, frecuencia respiratoria y presión

arterial, el registro estar a cargo del Técnico del área de triaje, quien llamara al paciente por orden consecutivo de la ficha de atención de que se le entrego.

En la siguiente tabla se muestran las tareas planificadas para este sprint.

Tabla 35. Sprint Backlog: Registro de Triage

SPRINT 4: Módulo de registro de triaje			
Número de Sprint:		4	
Inicio del Sprint:		19 de agosto de 2019	
Duración del Sprint:		15 días	
Id	Tareas	Tipo	Estado
4.1	Planificación y análisis de los requerimientos	Planificación	Terminado
4.2	Diseño navegacional	Diseño	Terminado
4.3	Diseño de Presentación	Diseño	Terminado
4.4	Buscar paciente	Desarrollo	Terminado
4.5	Registrar signos vitales	Desarrollo	Terminado
4.6	Modificar signos vitales	Desarrollo	Terminado
4.7	Listar pacientes en espera	Desarrollo	Terminado
4.8	Administrar medicamentos	Desarrollo	Terminado
4.9	Registrar resultados de laboratorio	Desarrollo	Terminado

Elaboración: Propia

3.4.4.2. Diagrama de caso de uso del Sprint 4. Para el registro de triaje, el paciente debe estar registrado en el sistema y contar con la ficha de atención asignada por el técnico del área de admisión, una vez pasada la ficha al área de triaje el técnico de esta área llamara a los pacientes según el número de su ficha. Si no se encuentra el paciente a la hora que se la llama, la ficha pasara a una lista de espera.

A continuación, se muestra el diagrama de casos de uso planteado para este módulo del sistema.

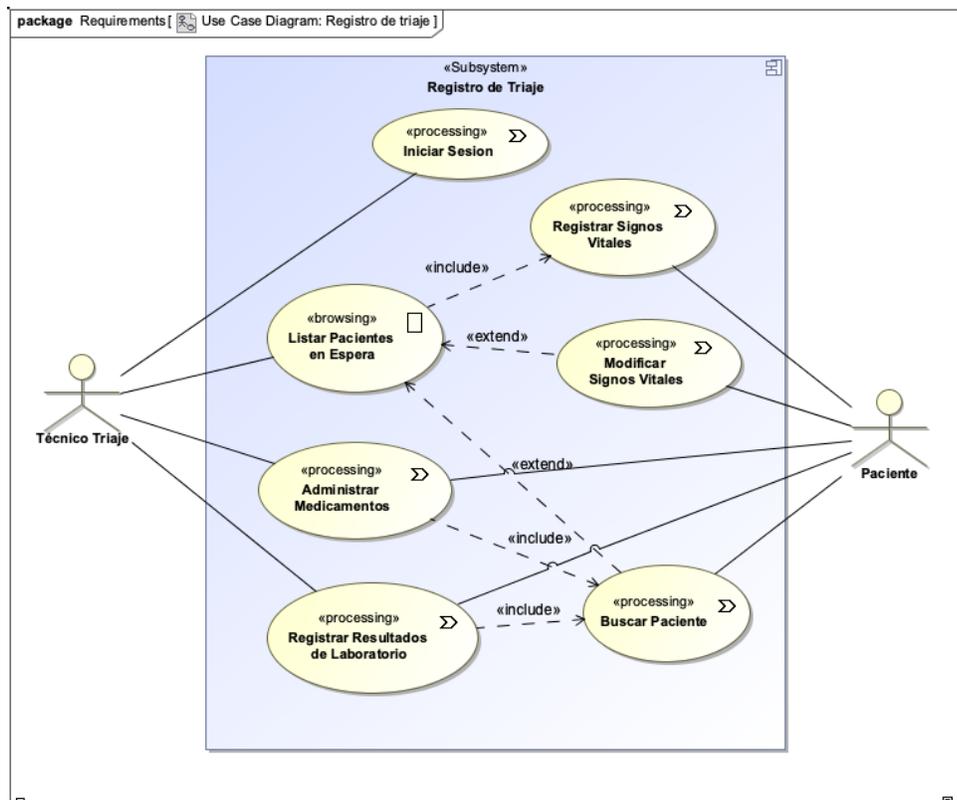


Figura 50. Diagrama de Caso de Uso: Registro de Triage
Elaboración: Propia

En las siguientes tablas se describirán los casos de uso planteados para este Sprint.

Tabla 36. Descripción del Caso de Uso: Listar pacientes en espera

Nro. 1	<i>Nombre del Caso de Uso: Listar pacientes en espera</i>
Actores:	Técnico Triage
Descripción:	Permite al usuario listar todos los pacientes en espera con fichas de atención
Precondición:	Los pacientes deben haber pasado previamente por el área de admisión para que se les otorgue una ficha de atención
Flujo Básico:	<ol style="list-style-type: none"> 1. Ingresar a la opción de listar pacientes 2. Buscar paciente a atender 3. Registrar Triage del paciente 4. Guardar datos tomados
Flujo alterno:	<ol style="list-style-type: none"> 1. Si se tuvo algún error en el registro se puede modificar los datos siempre y cuando el médico no atienda al paciente
Postcondición:	Se tienen los datos del paciente registrados

Elaboración: Propia

Tabla 37. Descripción del Caso de Uso: Administrar medicamentos

Nro. 2	Nombre del Caso de Uso: Listar pacientes en espera
Actores:	Técnico Triage
Descripción:	Permite al usuario registrar la administración de medicamentos a un paciente
Precondición:	El paciente debe estar con ficha de atención habilitada
Flujo Básico:	<ol style="list-style-type: none"> 1. Buscar paciente a administrar medicamento 2. Ingresar a la opción de listar pacientes 3. Verificar nota para la administración del medicamento en el historial clínico del paciente 4. Registrar datos 5. Guardar datos tomados
Postcondición:	Se tienen los datos del paciente registrados
Elaboración: Propia	

Tabla 38. Descripción del Caso de Uso: Registrar resultados de laboratorio

Nro. 2	Nombre del Caso de Uso: Registrar resultados de laboratorio
Actores:	Técnico Triage
Descripción:	Permite registrar los resultados de exámenes complementarios, si así lo dispone el médico
Precondición:	Los pacientes deben llevar el examen complementario
Flujo Básico:	<ol style="list-style-type: none"> 1. Buscar al paciente 2. Ingresar a la opción de registro de exámenes complementarios 3. Registrar datos del examen 4. Guardar registro
Postcondición:	Se tienen los datos del paciente registrados
Elaboración: Propia	

3.4.4.3. Diseño navegacional del Sprint 4. El diseño de navegación muestra el flujo de navegación del técnico de triaje.

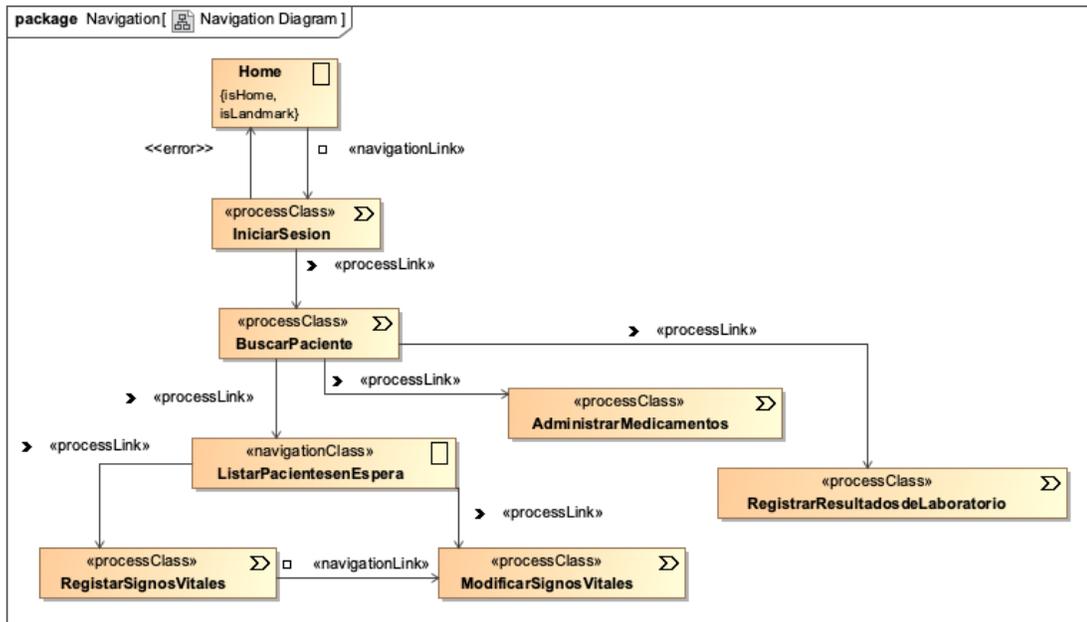


Figura 51. Diagrama de Navegación: Registro de triaje.
Elaboración: Propia

3.4.4.4. Diseño de presentación del Sprint 4.

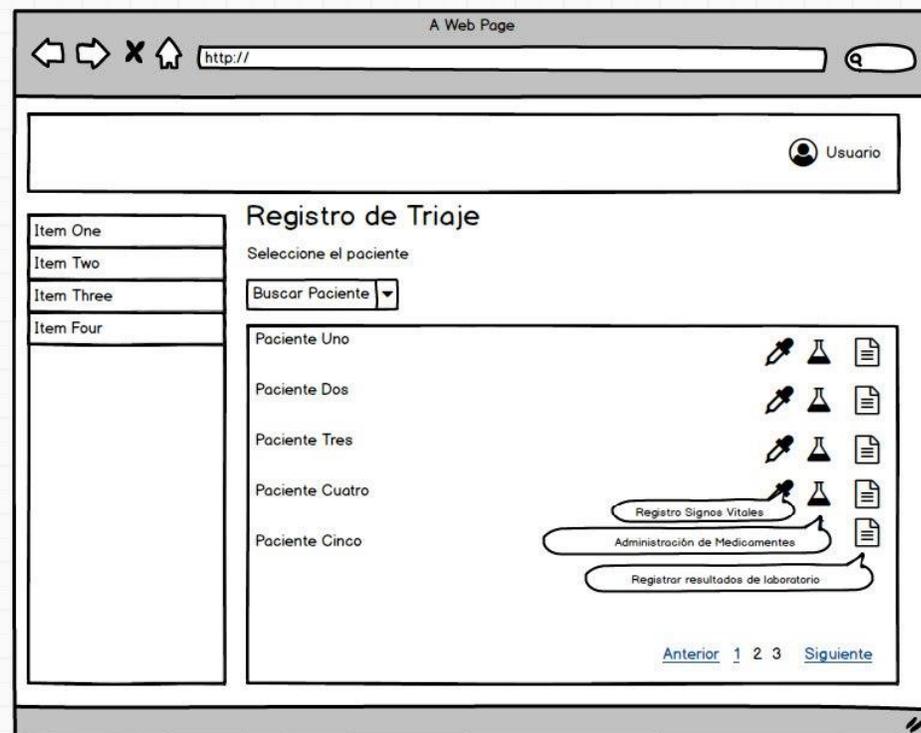


Figura 52. Diagrama de Presentación: Registro de triaje
Elaboración: Propia

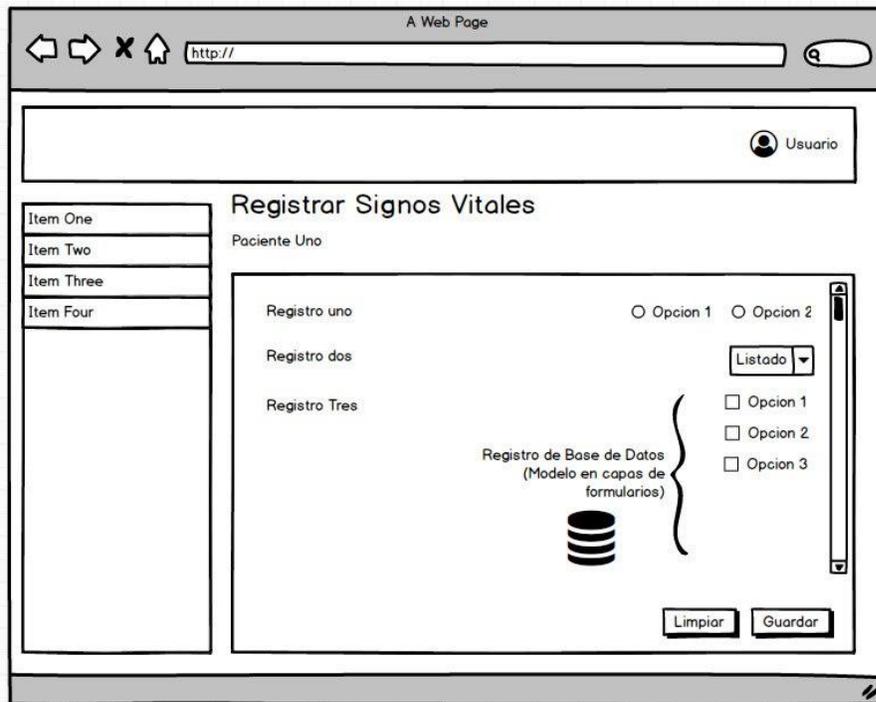


Figura 53. Diagrama de Presentación: Registro de signos vitales.
Elaboración: Propia

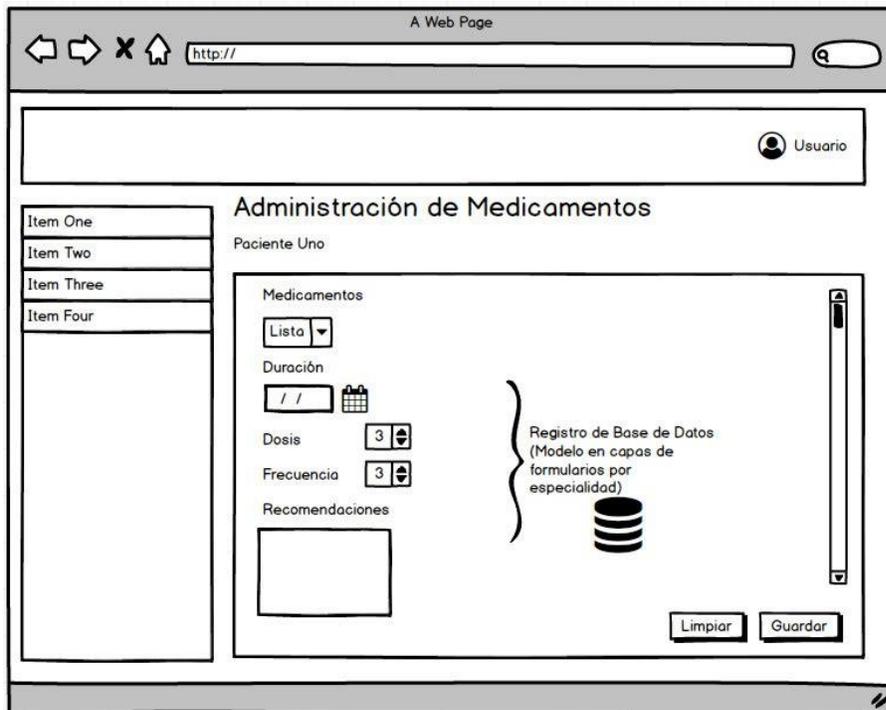


Figura 54. Diagrama de Presentación: Registro de medicamentos.
Elaboración: Propia

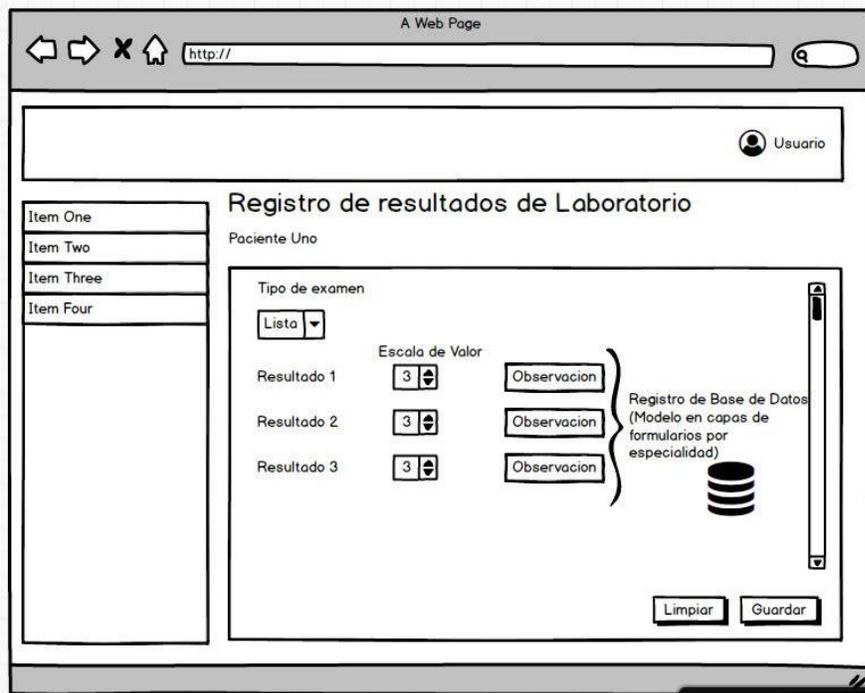


Figura 55. Diagrama de Presentación: Registro de resultados de laboratorio.
Elaboración: Propia

3.4.4.5. Pantallas del Sprint 4. A continuación, se muestra las capturas de pantalla para el administrador.

The screenshot shows the 'Detalle de Paciente a Atender' interface. At the top, there is a teal header with a logo and the user's name 'Lourdes Valencia'. Below the header, the page title is 'Detalle de Paciente a Atender' and the breadcrumb is 'Triage / Detalle de Paciente a Atender'. The main content area is titled 'PERSONAS EN ESPERA' and includes a 'mostrar 10 entradas' dropdown and a search box labeled 'Buscar:'. A table lists two patients with the following data:

Nro.	Nto. Ficha	Nro. HC	Nombres	Ap. Paterno	CI	Fecha de Cita	Acción
10	8	38647	Mercedes	Mamani	8759405	2019/11/10	[Icons: Heart, Camera, Person]
11	9	87749	Mario	Lopez	5896744	2019/11/12	[Icons: Heart, Camera, Person]

At the bottom of the table, it says 'Mostrando 1 a 2 de 2 entradas' and navigation buttons for 'Anterior' and 'Siguiente'.

Figura 56. Captura de Pantalla: Registro de Triage

Elaboración: Propia

The screenshot shows the 'Registro de Triage' interface. At the top, there is a teal header with a logo and the user's name 'Lourdes Valencia'. Below the header, the page title is 'Registro de Triage' and the breadcrumb is 'Triage / Registro de Triage'. The main content area is titled 'SIGNOS VITALES' and contains several input fields for vital signs:

- Temperatura (C)
- Frecuencia Cardiaca (Latidos x minutos)
- Temperatura (mmHg)
- Frecuencia Respiratoria (El pecho x minuto)
- Peso (kg)
- Talla (cm)

At the bottom, there are 'Guardar' and 'Cancelar' buttons.

Figura 57. Captura de Pantalla: Registro de Triage

Elaboración: Propia

3.4.4.6. Prueba unitaria del Sprint 4. La prueba unitaria a continuación es para comprobar que el modulo funcione correctamente

Tabla 39. Prueba unitaria: Inicio de Sesión

Id	Descripción de caso de Prueba	Resultado
1	El sistema busca de manera eficiente al paciente	Cumple
2	El sistema registra correctamente los signos vitales	Cumple
4	El sistema registra correctamente la administración de medicamentos al paciente	Cumple
5	El sistema registra correctamente los exámenes complementarios	Cumple

Elaboración: Propia

3.4.5. SPRINT 5: Módulo de gestión de historia clínica.

3.4.5.1. Planificación del Sprint 5. En el quinto Sprint se realizará todo el registro de lo que comprende una historia clínica, desde la anamnesis, el examen clínico, la emisión del diagnóstico, el plan de trabajo a seguir, el diagnóstico y tratamiento hasta las notas de evolución del paciente. Para el mismo se listan las tareas a seguir para este módulo.

Tabla 40. Sprint Backlog: Gestión de Historias Clínicas

SPRINT 5: Módulo de Gestión de Historias Clínicas			
Número de Sprint:		5	
Inicio del Sprint:		9 de septiembre de 2019	
Duración del Sprint:		25 días	
Id	Tareas	Tipo	Estado
5.1	Planificación y análisis de los requerimientos	Planificación	Terminado
5.2	Diseño navegacional	Diseño	Terminado
5.3	Diseño de Presentación	Diseño	Terminado
5.4	Registrar anamnesis	Desarrollo	Terminado
5.5	Registrar examen clínico	Desarrollo	Terminado
5.6	Emitir diagnóstico	Desarrollo	Terminado
5.7	Registrar plan de trabajo	Desarrollo	Terminado
5.8	Registrar tratamiento	Desarrollo	Terminado
5.9	Registrar evolución	Desarrollo	Terminado
5.10	Registrar antecedentes	Desarrollo	Terminado

En las siguientes tablas se hace la descripción de los casos de uso diagramados en la figura anterior.

Tabla 41. Descripción del Caso de Uso: Registro de anamnesis

Nro. 1	Nombre del Caso de Uso: Registrar anamnesis
Actores:	Médico
Descripción:	Permite al médico registrar los antecedentes y síntomas del paciente que está siendo atendido
Precondición:	El paciente debe contar previamente con ficha de atención
Flujo Básico:	<ol style="list-style-type: none"> 1. El médico busca al paciente a atender 2. Revisa antecedentes anteriores registrados en el sistema 3. Registra el motivo de la consulta 4. Registra síntomas o algún nuevo antecedente que no esté registrado 5. Guarda datos de registrados para pasar a la siguiente fase
Postcondición:	Se tiene registrado la primera fase de la consulta médica
Elaboración: Propia	

Tabla 42. Descripción del Caso de Uso: Registro de examen clínico

Nro. 2	Nombre del Caso de Uso: Registrar examen clínico
Actores:	Médico
Descripción:	Permite al médico registrar los antecedentes y síntomas del paciente que está siendo atendido
Precondición:	Debe haberse registrado previamente el registro de la anamnesis
Flujo Básico:	<ol style="list-style-type: none"> 1. Registra el examen general 2. Registra examen regional 3. Consulta los datos recabados en el triaje 4. Guarda los datos registrados para pasar a la siguiente fase
Postcondición:	Se tiene registrada la segunda fase de la consulta médica
Elaboración: Propia	

Tabla 43. Descripción del Caso de Uso: Emisión de diagnóstico

Nro. 3	Nombre del Caso de Uso: Emitir diagnóstico
Actores:	Médico
Descripción:	Permite al médico registra el diagnóstico emitido en dicha consulta y tener un respaldo de el mismo
Precondición:	Debe haberse registrado previamente el examen clínico

Flujo Básico:	1. Se debe registrar el diagnóstico tomando en cuenta todos los datos ya registrados y los antecedentes existentes en la historia clínica 2. Guarda los datos registrados para pasar a la siguiente fase
Flujo alterno:	1. Si con los datos registrados se puede tomara un diagnóstico el médico puede imprimir la receta la receta médica para el paciente
Postcondición:	Se tiene registrado la tercera fase de la consulta médica
Elaboración: Propia	

Tabla 44. Descripción del Caso de Uso: Registro de plan de trabajo

Nro. 4	Nombre del Caso de Uso: Registrar plan de trabajo
Actores:	Médico y Paciente
Descripción:	Permite al médico registrar el plan de trabajo es decir el procedimiento a seguir, el cual puede ser por ejemplo solicitar un examen complementario
Precondición:	Debe haberse registrado previamente el diagnóstico
Flujo Básico:	1. Se debe registrar si se solicito algún examen complementario de acuerdo al diagnostico emitido 2. Guarda los datos registrados para pasar a la siguiente fase
Flujo alterno:	1. Se imprime orden de laboratorio para que el paciente se realice los exámenes complementarios
Postcondición:	Se tiene registrado la cuarta fase de la consulta médica
Elaboración: Propia	

Tabla 45. Descripción del Caso de Uso: Registro del tratamiento

Nro. 4	Nombre del Caso de Uso: Registrar tratamiento
Actores:	Médico
Descripción:	Permite al médico registrar el tratamiento que debe seguir el paciente para tratar el mal que lo aqueja
Precondición:	Debe haberse registrado previamente el diagnóstico o plan de trabajo
Flujo Básico:	1. Se debe registrar el tratamiento a seguir por el paciente 2. Debe poner observaciones complementarias si así lo decide 3. Guarda los datos registrados para terminar la consulta médica
Flujo alterno:	2. El médico puede pedir que el paciente vuelva a consulta para ver la evolución del tratamiento.
Postcondición:	Se tiene registrada la consulta médica en el historial clínico para su posterior archivo

Elaboración: Propia

Tabla 46. Descripción del Caso de Uso: Registro de evolución

Nro. 4	Nombre del Caso de Uso: Registrar evolución
Actores:	Médico
Descripción:	Permite al médico registrar la evolución del paciente con respecto al tratamiento brindado
Precondición:	El paciente debe solicitar otra atención médica
Flujo Básico:	<ol style="list-style-type: none"> 1. Se debe registrar la evolución del paciente 2. Guarda los datos registrados para terminar la consulta médica
Flujo alterno:	<ol style="list-style-type: none"> 1. El médico puede pedir que el paciente vuelva otra consulta médica.
Postcondición:	Se tiene registrada el seguimiento del paciente en el historial clínico

3.4.5.3. Diseño navegacional del Sprint 5. En el siguiente diagrama de

navegación se puede ver el flujo de una consulta médica.

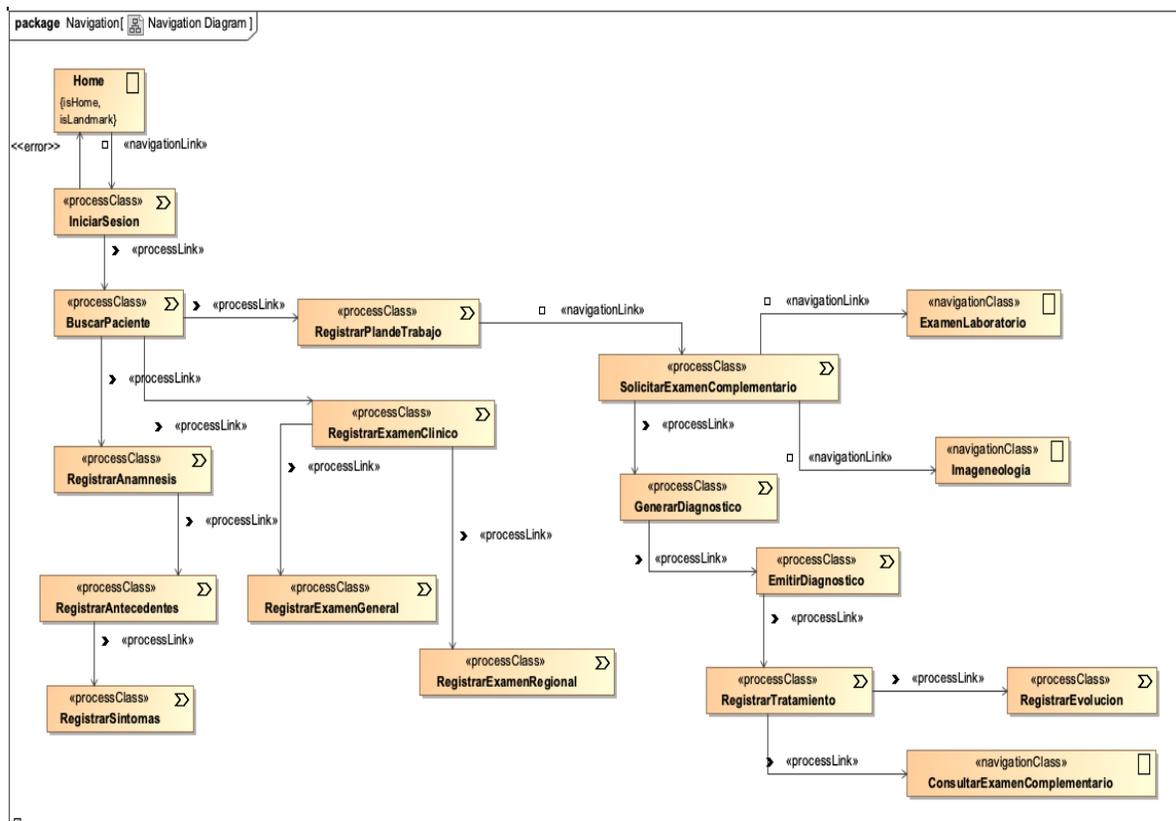


Figura 59. Diagrama de Navegación: Gestión de Historia Clínica

Elaboración: Propia

3.4.5.4. Diseño de presentación del Sprint 5.

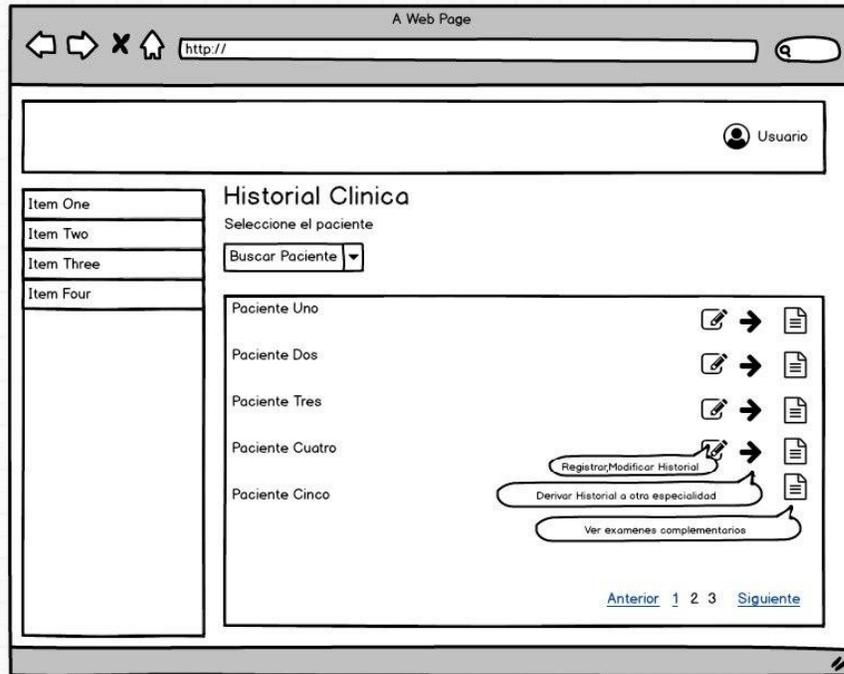


Figura 60. Diagrama de Presentación: Gestión de Historia Clínica.
Elaboración: Propia



Figura 61. Diagrama de Presentación: Historia Clínica - Antecedentes
Elaboración: Propia

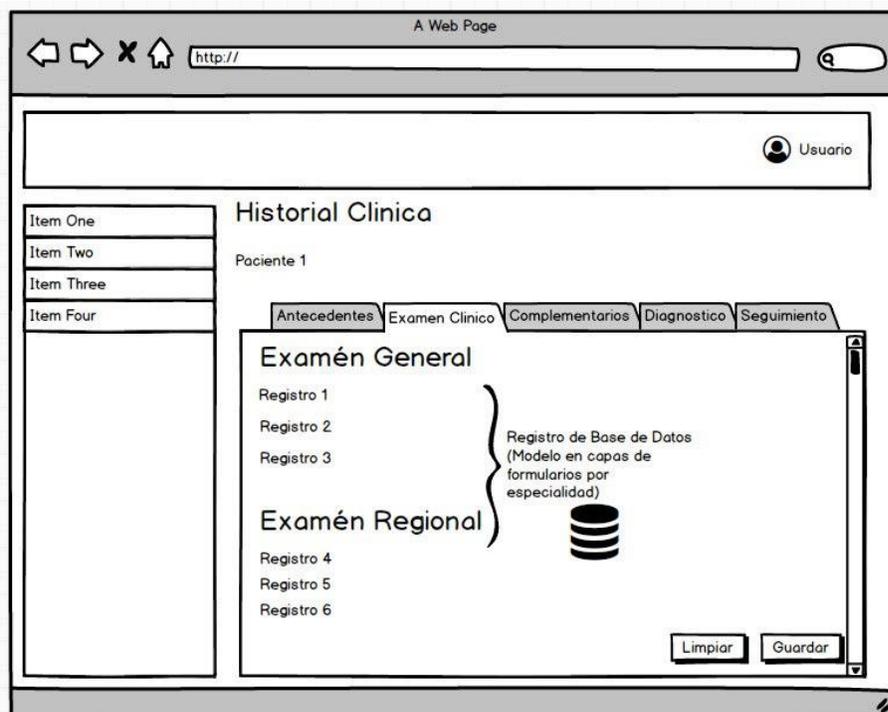


Figura 62. Diagrama de Presentación: Historia Clínica – Examen Clínico.
Elaboración: Propia

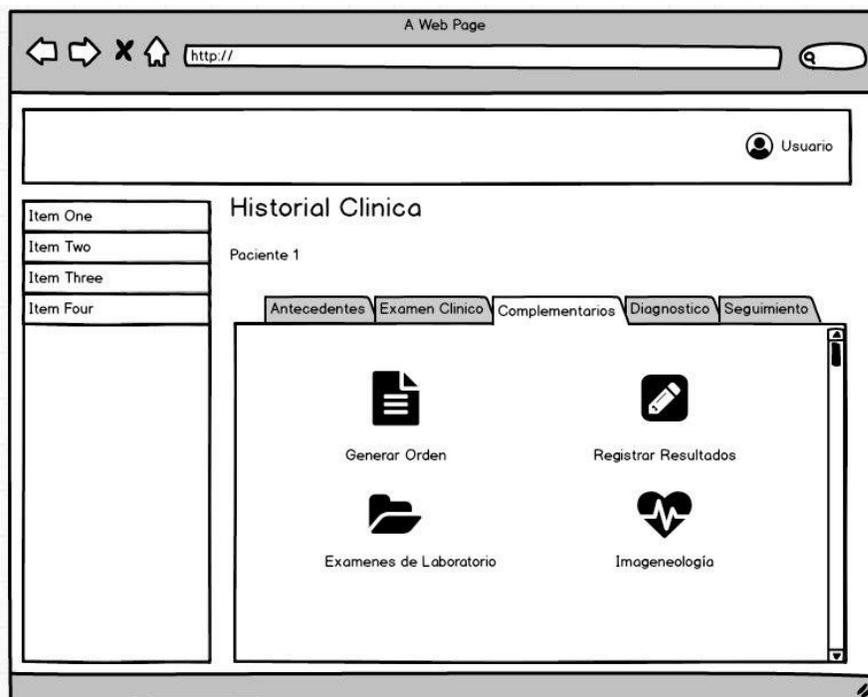


Figura 63. Diagrama de Presentación: Historia Clínica – Exámenes Complementarios
Elaboración: Propia

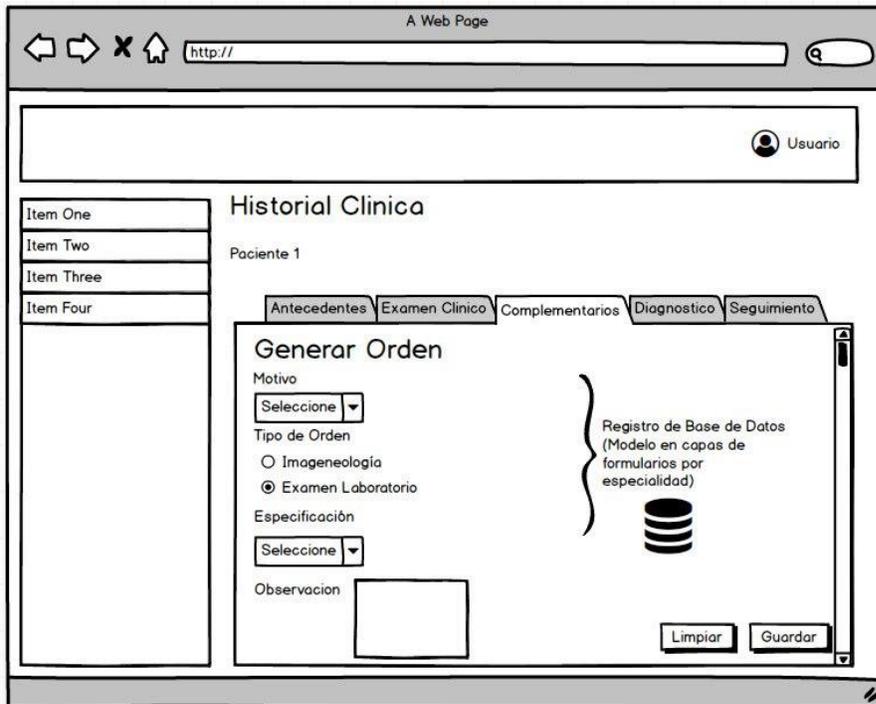


Figura 64. Diagrama de Presentación: Historia Clínica – EC Generar Orden.
Elaboración: Propia

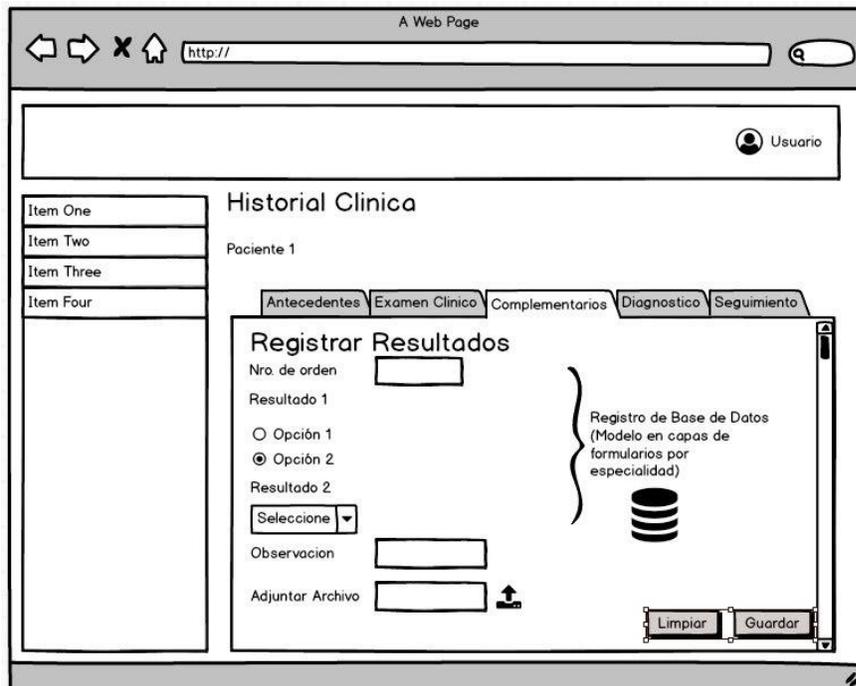


Figura 65. Diagrama de Presentación: Historia Clínica – EC Registrar Resultados.
Elaboración: Propia

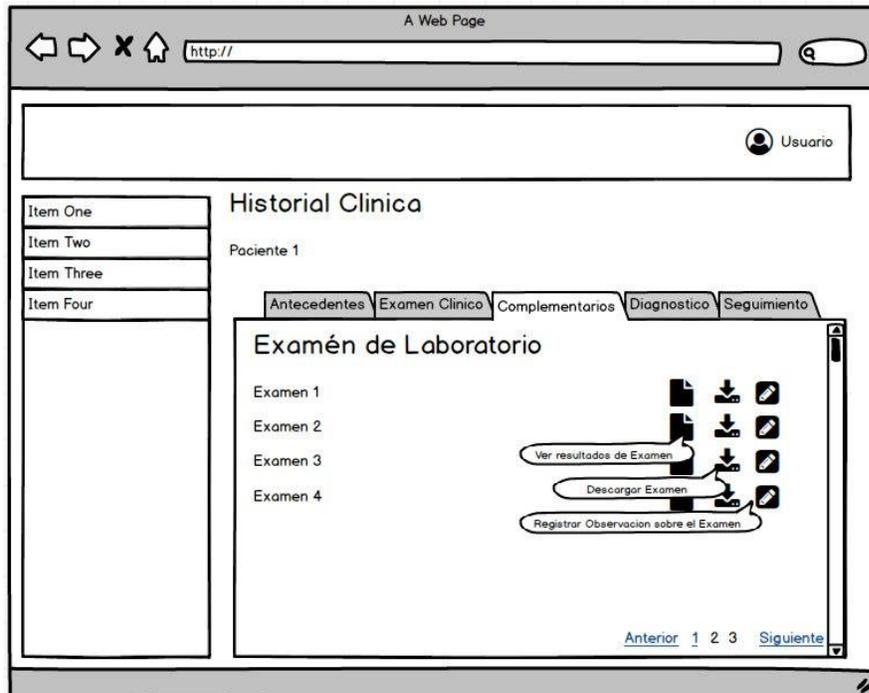


Figura 66. Diagrama de Presentación: Historia Clínica – EC Examen de Laboratorio
Elaboración: Propia

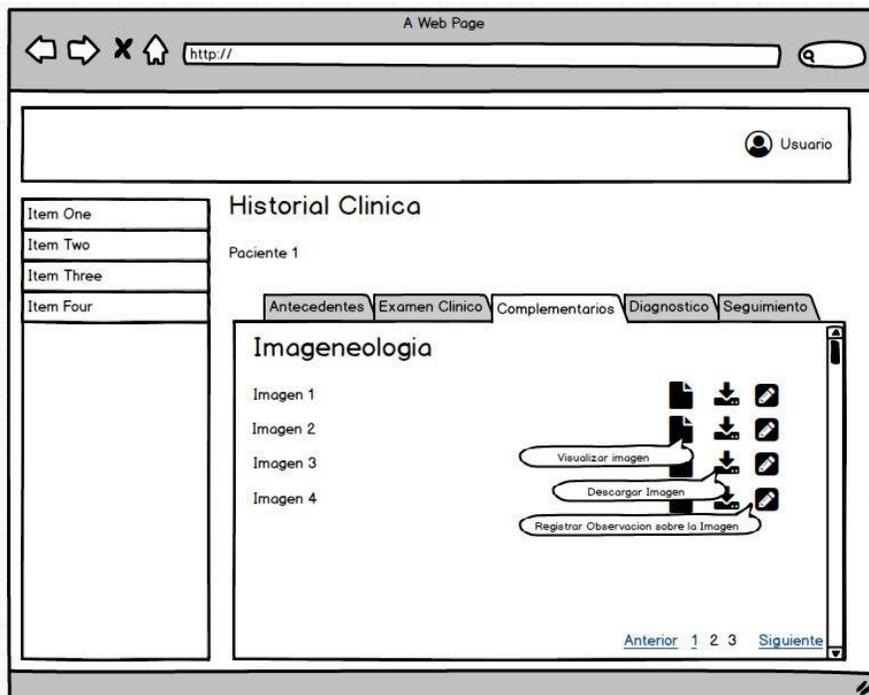


Figura 67. Diagrama de Presentación: Historia Clínica – EC Imageneología
Elaboración: Propia

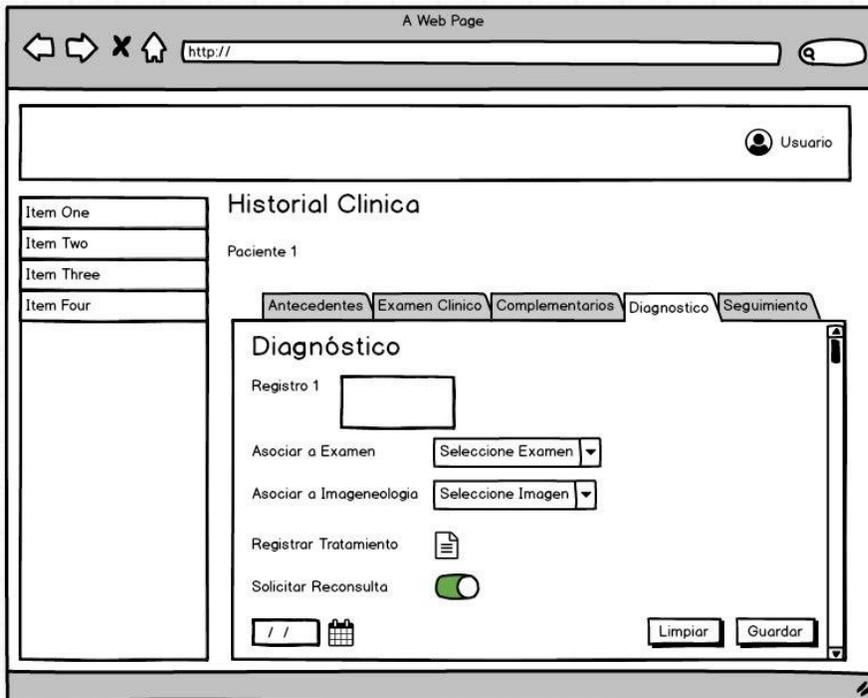


Figura 68. Diagrama de Presentación: Historia Clínica – Diagnóstico
Elaboración: Propia

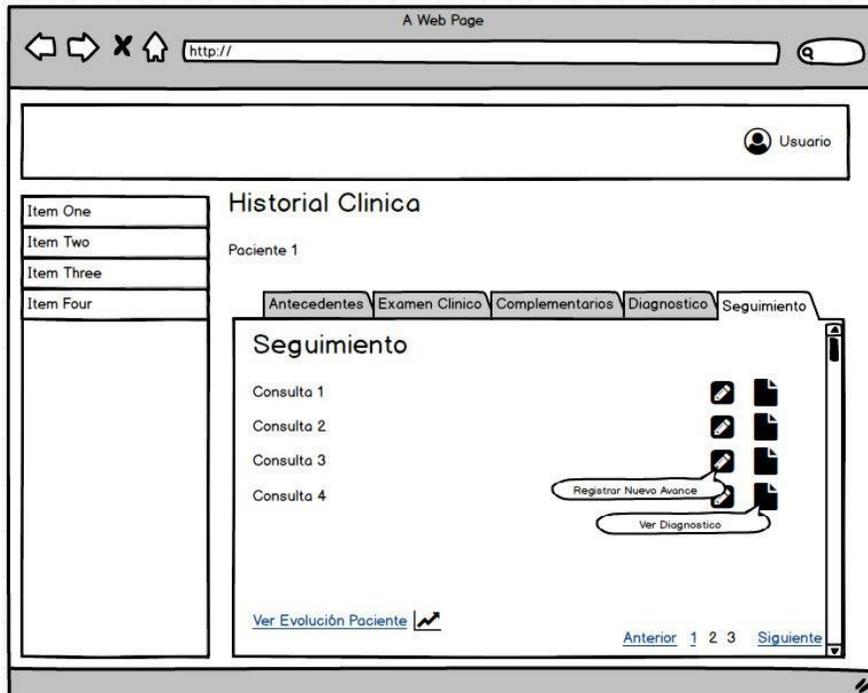


Figura 69. Diagrama de Presentación: Historia Clínica – Seguimiento
Elaboración: Propia

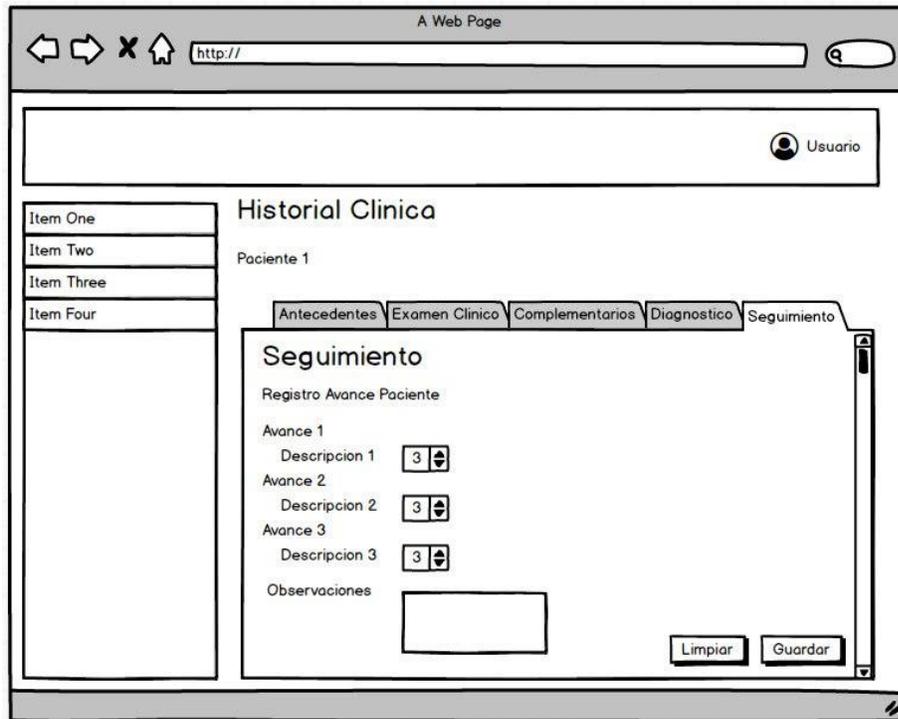


Figura 70. Diagrama de Presentación: Historia Clínica – Registrar Seguimiento.
Elaboración: Propia

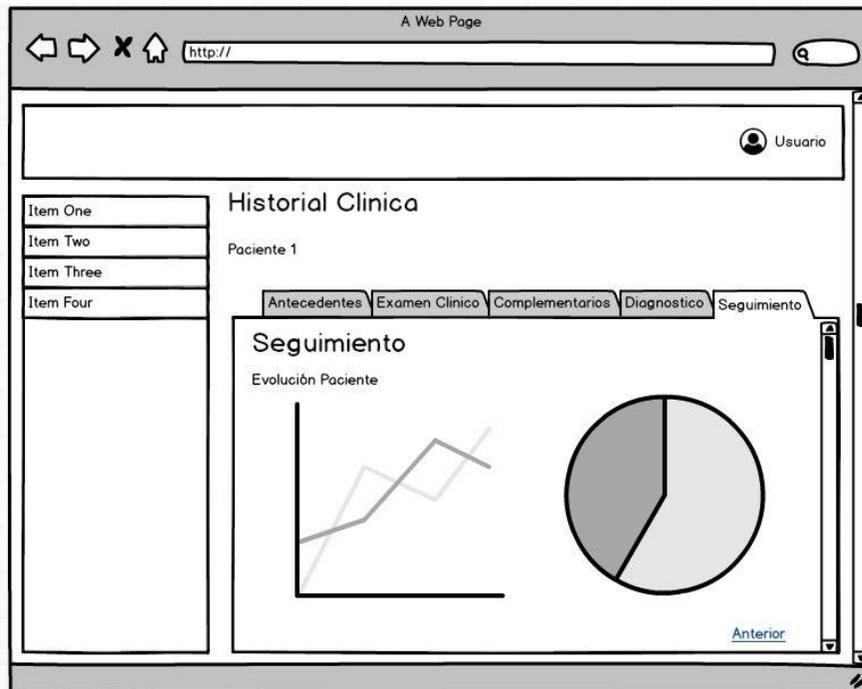


Figura 71. Diagrama de Presentación: Historia Clínica – Evolución del Paciente.
Elaboración: Propia

3.4.5.5. Pantallas del Sprint 5.A continuación se muestra las captura de pantalla para el administrador.

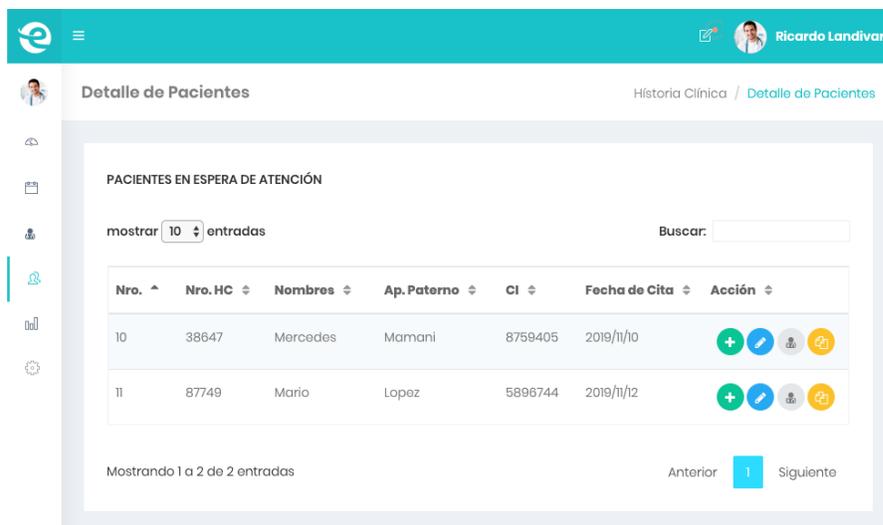


Figura 72. Captura de pantalla: Gestión de Historias Clínicas.
Elaboración: Propia

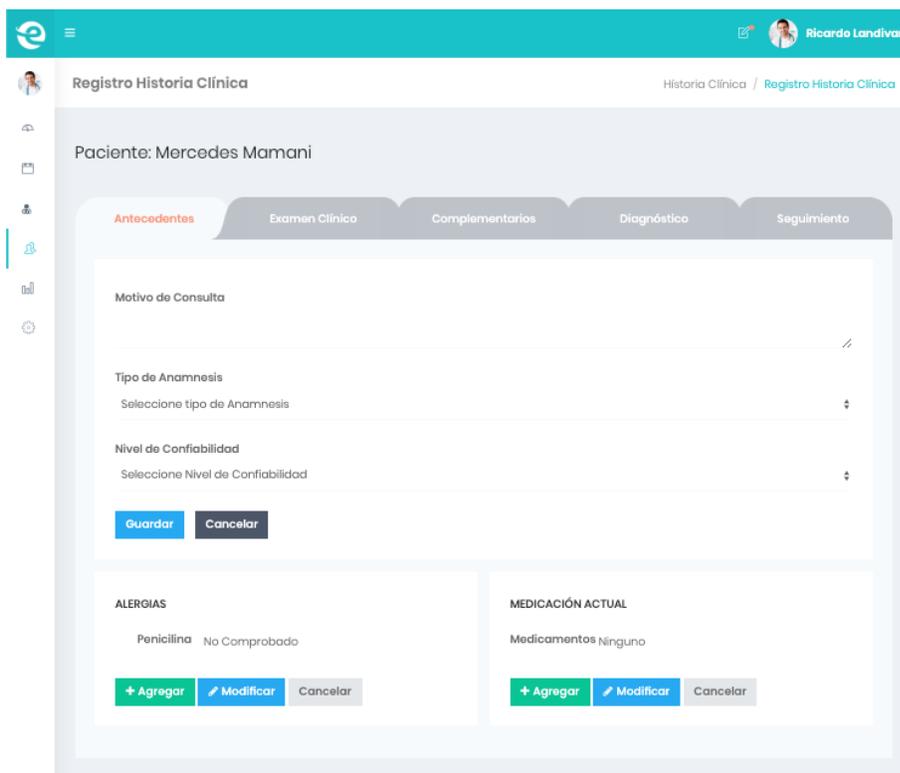


Figura 73. Captura de pantalla: Gestión de Historias Clínicas.
Elaboración: Propia

3.4.5.6. Prueba unitaria del Sprint 5. La prueba unitaria a continuación es para comprobar que el modulo funcione correctamente.

Tabla 47. Prueba unitaria: Gestión de historia clínica

Id	Descripción de caso de Prueba	Resultado
1	El sistema busca de manera eficiente al paciente	Cumple
2	El sistema registra de manera correctamente los datos registrados en las 5 fases	Cumple
4	El sistema no tiene dificultad al enlazar las fases del registro.	Cumple
5	El sistema es amigable y entendible par el registro de toda la consulta médica	Cumple

Elaboración: Propia

3.4.6. SPRINT 6: Módulo de generación de reportes.

3.4.6.1. Planificación del Sprint 6. En el sexto Sprint y último se realizará los reportes solicitados por el Colegio Médico, por el momento solo son dos, el informe médico y el certificado médico. Los mismos deben ser solicitados con anticipación por los pacientes. Para el mismo se listan las tareas a seguir para este módulo.

Tabla 48. Sprint Backlog: Generación de reportes

SPRINT 6: Módulo de Generación de reportes			
Número de Sprint:	6		
Inicio del Sprint:	14 de octubre de 2019		
Duración del Sprint:	3 días		
Id	Tareas	Tipo	Estado
6.1	Planificación y análisis de los requerimientos	Planificación	Terminado
6.2	Diseño navegacional	Diseño	Terminado
6.3	Diseño de Presentación	Diseño	Terminado
6.4	Buscar paciente	Desarrollo	Terminado
6.5	Generar informe médico	Desarrollo	Terminado
6.6	Generar certificado médico	Desarrollo	Terminado
6.7	Imprimir receta	Desarrollo	Terminado

Elaboración: Propia

3.4.6.2. Diagrama de caso de uso del Sprint 6. El diagrama de casos de uso de la generación de reportes muestra la interacción del médico con el sistema para generar el reporte solicitado por el paciente. A continuación, mostramos lo mencionado.

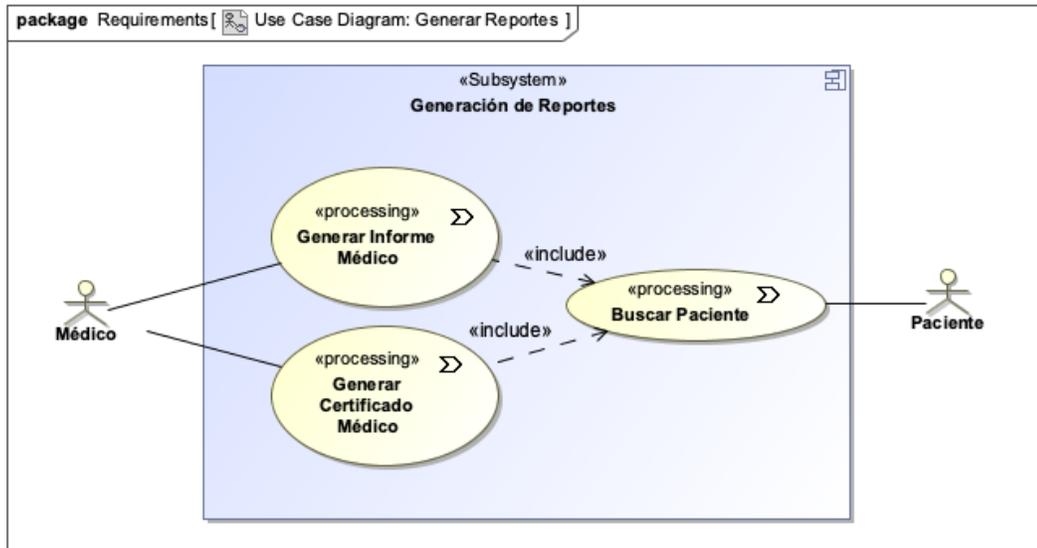


Figura 74. Diagrama de Caso de Uso: Generar Reportes.
Elaboración: Propia

En las siguientes tablas se hace la descripción de los casos de uso diagramados en la figura anterior.

Tabla 49. Descripción del Caso de Uso: Generar reportes

Nro. 1	Nombre del Caso de Uso: Generar reportes
Actores:	Médico y paciente
Descripción:	Permite al médico generar el reporte solicitado
Precondición:	El paciente debe solicitar el documento
Flujo Básico:	<ol style="list-style-type: none"> 1. El médico busca al paciente a atender 2. Modifica y redacta lo que falte 3. Guarda el reporte generado 4. Imprime el reporte
Postcondición:	El paciente tiene el reporte solicitado

Elaboración: Propia

3.4.6.3. Diseño navegacional del Sprint 6. En el siguiente diagrama de navegación se puede ver el flujo de una de la generación de reportes.

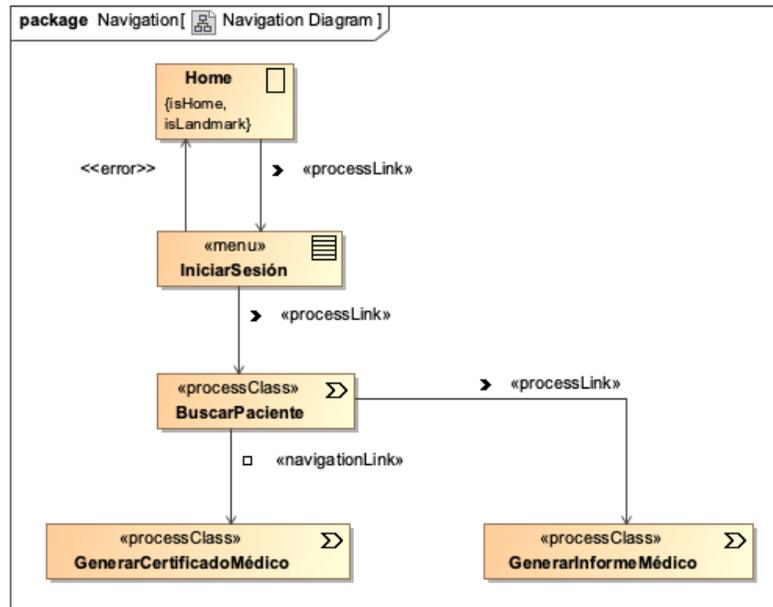


Figura 75. Diagrama de Navegación: Generar Reportes
Elaboración: Propia

3.4.6.4. Diseño de presentación del Sprint 6.

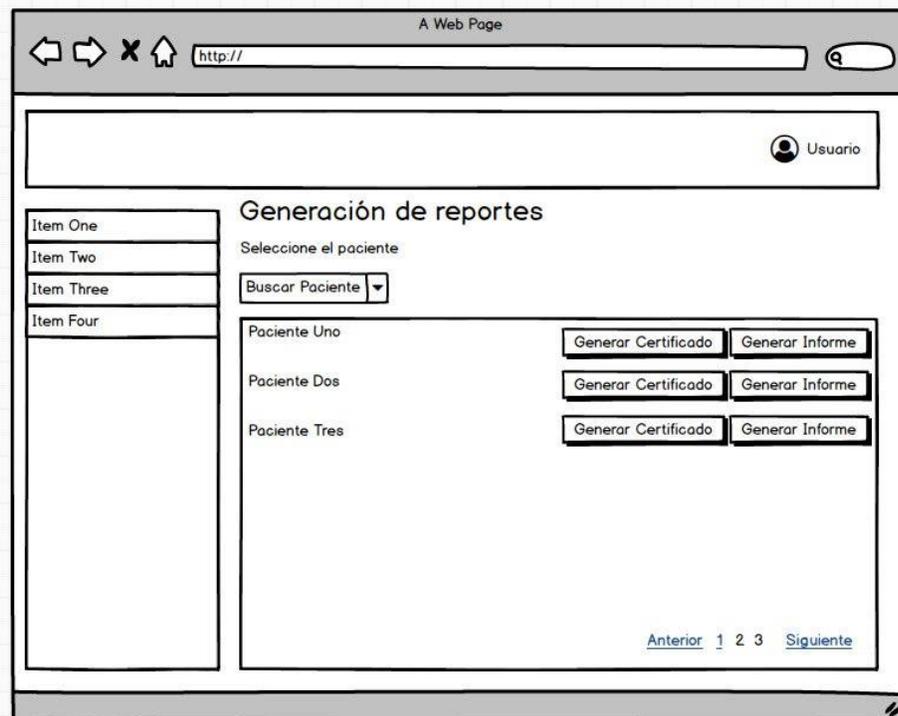


Figura 76. Diagrama de Presentación: Generar Reportes
Elaboración: Propia

3.4.6.5. Pantallas del Sprint 6.A continuación se muestra la captura de pantalla para el administrador.

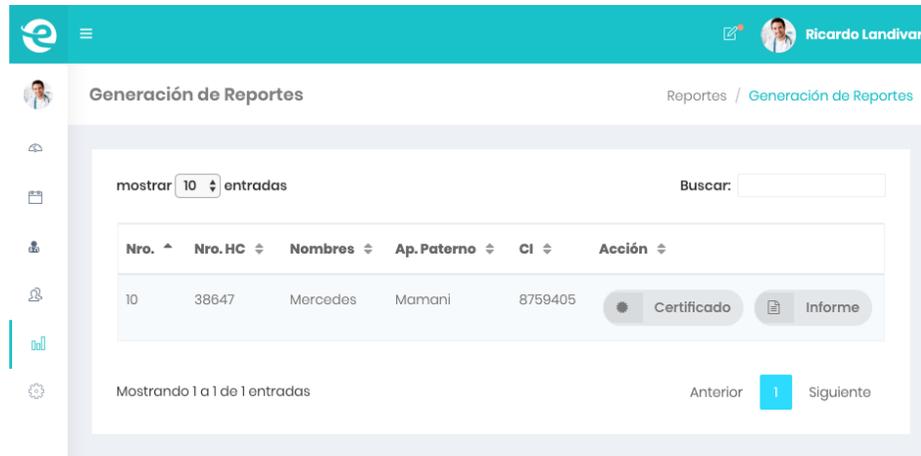


Figura 77. Captura de pantalla: Generar Reportes.
Elaboración: Propia

3.4.6.6. Prueba unitaria del Sprint 6.La prueba unitaria a continuación es para comprobar que el modulo funcione correctamente.

Tabla 50. Prueba unitaria: Generación de Reportes

Id	Descripción de caso de Prueba	Resultado
1	El sistema busca de manera eficiente al paciente	Cumple
2	El genera de manera correcta los reportes	Cumple

Elaboración: Propia

3.5. Post – Game (Después del desarrollo).

Luego de haber concluido con la fase del Game se procederá a realizar las pruebas de desarrollo que se exponen a continuación a través de las pruebas de caja negra y caja blanca, para verificar si existen posibles errores o bugs en el sistema.

3.5.1. Pruebas de caja negra.El objetivo principal de estas pruebas de funcionalidad es verificar que se esté ingresando, procesando y mostrando los datos de acuerdo a lo que se necesita. Por lo cual se utilizará la técnica de prueba

de caja negra, la cual analiza las entrada que recibe y las salidas o respuestas que produce, por lo que deben estar bien definidas (entradas y salidas). Con esta técnica de prueba a aplicar se sabrá claramente que es lo que hace el software, y no como lo hace.

Tabla 51. Caso de Prueba: Buscar Historia Clínica

Caso de Prueba de Respuesta	
Código Caso de Prueba: 01	
Caso de Prueba: <i>Buscar Historia Clínica</i>	
Descripción de Caso de Prueba	
Propósito:	Comprobar que se realice correctamente la búsqueda de la historia clínica de acuerdo a los filtros que utilice el usuario.
Prerrequisitos:	<p>Usuario se encuentre habilitado.</p> <p>El usuario cuente con permisos para acceder a las HC de los pacientes.</p>
Datos de Entrada:	<ul style="list-style-type: none"> • Nro. de documento • Nro. de Historia clínica • Apellido Paterno • Apellido materno • Nombre • Fecha de nacimiento
Pasos:	<p>El usuario deberá seleccionar la opción Buscar desde la parte superior de la tabla de pacientes.</p> <p>El usuario deberá ingresar, según criterio de búsqueda los valores.</p> <p>El sistema mostrara en la primera línea de la tabla el resultado.</p>
Resultado esperado:	<p>Funcionalidad de búsqueda debe ser accedida desde menú principal.</p> <p>Se muestra listado de pacientes según filtros de búsqueda del usuario.</p>
Resultado obtenido:	El sistema retorna el listado de los pacientes coincidentes según criterio de búsqueda.
Observaciones:	Si los datos ingresados en los filtros no coinciden con la información almacenada en la base de datos, el sistema no mostrara ningún resultado.

Elaboración: Propia

Tabla 52. Caso de Prueba: Registrar Atención Médica

Caso de Prueba de Respuesta	
Código Caso de Prueba: 02	
Caso de Prueba: Registrar Atención Médica	
Descripción de Caso de Prueba	
Propósito:	Comprobar el registro del paciente para su consulta y atención médica.
Prerrequisitos:	Usuario se encuentre habilitado. El usuario cuente con permisos para registrar atenciones.
Datos de Entrada:	<ul style="list-style-type: none"> • Tipo de servicio • Especialidad
Pasos:	<p>El usuario deberá seleccionar la opción Generar atención desde el menú del sistema</p> <p>El usuario deberá buscar al paciente para verificar si se encuentra registrado</p> <p>El usuario seleccionara el servicio de salud que el paciente requiere para generar su atención.</p>
Resultado esperado:	<p>Funcionalidad de Generar atención médica.</p> <p>Registro sin errores de la atención a realizarse en la Base de Datos.</p>
Resultado obtenido:	<p>Se pudo realizar la búsqueda de la historia clínica desde el módulo de registro de atención.</p> <p>Se registra la atención del paciente a consulta tomando en cuenta todos los campos del formulario.</p>
Observaciones:	
<i>Elaboración:</i> Propia	

Tabla 53. Caso de Prueba: Registrar nuevo paciente

Caso de Prueba de Respuesta	
Código Caso de Prueba: 03	
Caso de Prueba: Registrar Paciente	
Descripción de Caso de Prueba	
Propósito:	Verificar que los campos del formulario que se muestra, estén definidos de acuerdo al modelo de datos y que la información ingresada se registre

Prerequisitos:	<p>Usuario se encuentre habilitado.</p> <p>El usuario cuente con permisos para poder registrar nuevos pacientes.</p>
Datos de Entrada:	<ul style="list-style-type: none"> • Nro. Documento de identidad • Nombres • Apellido Paterno • Apellido Materno • Fecha de Nacimiento • Dirección
Pasos:	<p>El usuario deberá seleccionar la opción registrar paciente nuevo desde el menú principal o listado de pacientes.</p> <p>El sistema mostrara el formulario para registro de nuevo paciente.</p> <p>El usuario deberá ingresar todos los datos restantes en el formulario.</p>
Resultado esperado:	<p>Funcionalidad de registro de nuevos pacientes.</p> <p>Campos del formulario de acuerdo a datos brindados.</p> <p>Registro de información del paciente en base a los datos.</p>
Resultado obtenido:	<p>Los datos ingresados en los campos del formulario fueron registrados sin ningún tipo de inconveniente.</p>
Observaciones:	<p>Si existiese algún error en el registro de datos, ya sea por registros existentes u otros, el sistema mostrara el mensaje de error.</p>

Elaboración: Propia

Tabla 54. Caso de Prueba: Gestionar Historia Clínica

Caso de Prueba de Respuesta	
Código Caso de Prueba: 04	
Caso de Prueba: Registrar Paciente	
Descripción de Caso de Prueba	
Propósito:	<p>Constatar que los datos ingresados por el usuario se registren y que los campos del formulario estén definidos de acuerdo al modelo.</p>
Prerequisitos:	<p>Usuario se encuentre habilitado.</p> <p>El usuario cuente con permisos para poder registrar nuevos pacientes.</p>
Datos de Entrada:	<ul style="list-style-type: none"> • Motivo de consulta • Síntomas

	<ul style="list-style-type: none"> • Grado de enfermedad • Diagnóstico • Frecuencia y cantidad del medicamento indicado • Orden clínica (si lo requiere)
Pasos:	<p>El usuario deberá seleccionar la opción de registrar atención medica desde el menú del sistema.</p> <p>El sistema mostrara un listado de atención de pacientes en espera.</p> <p>El usuario deberá seleccionar el paciente para su atención.</p> <p>El sistema mostrara los últimos datos de la atención del paciente.</p> <p>El usuario elaborara su diagnostico y lo ingresara al sistema.</p>
Resultado esperado:	Campos del formulario de acuerdo al modelo de datos de atención al paciente.
Resultado obtenido:	Detalle de la atención, diagnostico, indicaciones y medicamentos fueron registrados.
Observaciones:	Si existiese algún error en el registro de datos, el sistema mostrara el mensaje de error correspondiente.

Elaboración: Propia

3.6. Calidad de software - ISO 9126

El objetivo principal es validar el proyecto de manera que este pueda de poder alcanzar la calidad necesaria para satisfacer las necesidades del cliente. La calidad según esta norma ISO 9126 puede ser medida de acuerdo a los factores de: usabilidad, funcionalidad, confiabilidad, mantenibilidad y portabilidad.

3.6.1. Usabilidad. La usabilidad consiste de un conjunto de atributos que permite evaluar el esfuerzo necesario que deberá invertir el usuario para utilizar el sistema, es decir realizar una serie de preguntas que permiten ver cuán sencillo, fácil de aprender y manejar es para los usuarios. Esta comprensión por parte de los usuarios con relación al sistema evalúa los siguientes pasos: entendimiento, aprendizaje, operabilidad, atracción y conformidad de uso.

En la siguiente tabla se observa estos criterios en niveles de porcentajes a los que llegó el sistema en cuanto a su comprensibilidad, para el usuario, y posteriormente se da el porcentaje final de usabilidad del sistema. Se realizó una encuesta al usuario final sobre el manejo, la comprensión y la facilidad de aprender el sistema para medir la usabilidad según la siguiente tabla.

Tabla 55. Encuesta de usabilidad del sistema

<i>Preguntas</i>	<i>Respuestas</i>		<i>Porcentaje de Validación</i>
	<i>Si (1-10)</i>	<i>No (1-10)</i>	
¿El acceso al sistema es complicado?	0	10	100%
¿Las respuesta del sistema son de su comprensibilidad?	8	2	80%
¿Tiene alguna dificultad en realizar los procesos del sistema?	1	9	90%
¿La interfaz de la aplicación es amigable y entendible a su parecer?	10	0	100%
¿Son comprensibles y satisfactorios los datos que muestran en el sistema?	9	1	90%
¿El sistema coadyuva de manera eficiente en los procesos que se realizaban anteriormente?	9	1	90%
¿El sistema satisface los requerimiento del Colegio Médico?	9	1	90%
¿El sistema es de fácil uso bajo su criterio?	8	2	80%
¿Es aplicable el sistema en el control de historiales médicos?	9	1	90%
<i>Promedio Total</i>			90%

Usabilidad = 90%

De acuerdo a los datos obtenidos en la tabla de usabilidad, se concluye que el sistema tiene una usabilidad del 90%. Esto indica que el usuario de la aplicación tiene una conformidad hacia el sistema y que este podrá coadyuvar de alguna manera a solucionar los problemas en cuanto al manejo de historias médicas que se presentaban en la institución.

3.6.2. Funcionalidad. La funcionalidad examina si el sistema satisface los requisitos funcionales esperados. El objetivo es revelar problemas y errores en lo que concierne a la funcionalidad del sistema y su conformidad al comportamiento, expresado o deseado por el usuario. Se cuantifica el tamaño y la complejidad del sistema en términos de las funciones de usuario, puede ser valorado mediante el Punto Función. Se basa en la contabilización de cinco parámetros los cuales se desarrollan a continuación:

- **Número de entradas de usuario:** Se refiere a cada entrada de control del usuario que proporciona diferentes datos al sistema.
- **Número de salidas de usuario:** Se refiere a cada salida de información que proporciona el sistema al usuario, entre estos están: informes, pantallas, mensajes de errores, entre otros.
- **Número de peticiones de usuario:** Una petición se define como una entrada interactiva que produce la generación de alguna respuesta de software en forma de salidas interactivas.
- **Número de Archivos:** Se cuenta archivos maestro lógico, estos pueden ser: grupo lógico de datos que sean parte de una base de datos, o un archivo independiente.
- **Número de interfaces externas:** Se cuenta las interfaces legibles por la máquina que se utilizan para transmitir información a otro sistema.

Para calcular el Punto Función se utilizará la siguiente fórmula:

$$\text{Punto Función (PF)} = \text{Total} * (X + \text{Min} * \text{Su}(Fi))$$

Donde:

- PF.** Medida de la funcionalidad.
- X:** Confiabilidad del proyecto, varía entre el 1% a 100%
- Min (y):** Error mínimo aceptable al de la complejidad, el margen de

error es igual a 0.01.

F_i Son los valores de ajuste de la complejidad, donde $i=1$ a $i=14$.

En la siguiente tabla se calcula el punto función, los cuales miden el software desde una perspectiva del usuario, dejando de lado los detalles de codificación.

Tabla 56. Conteo de parámetros punto de función con sus respectivas características y pesos

<i>Parámetros</i>	<i>Cantidad</i>	<i>Peso Simple</i>	<i>Peso Medio</i>	<i>Peso Completo</i>	<i>Factor Peso</i>
N.E.U.	26	3	4	6	104
N.S.U.	26	4	5	7	130
N.P.U.	6	3	5	6	30
N.F.	6	7	10	25	42
N.I.E.	0	5	7	20	0
Total					306

Fuente: (Taina, 2005)

Donde:

N.E.U.: Es el número de entradas de usuario para proporcionar datos al software.

N.S.U.: Es el número de salidas que proporcionan al usuario información.

N.P.U.: Es el número de peticiones que se realizan al usuario como respuesta a una salida interactiva.

N.F.: Es el número de ficheros maestros que se realizan en la aplicación.

N.I.E.: Es el número de interfaces externas que se utilizan para la transmisión de la información.

Cada organización que utiliza métodos de puntos desarrolla criterios para determinar si una entrada es denominada simple o compleja. Los valores de F_i , se obtiene de los resultados de la siguiente tabla, bajo las ponderaciones descritos en la escala.

Tabla 57. Ajuste de complejidad con el cuestionario correspondiente para su evaluación

<i>Importancia</i>	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>
<i>Escalas</i>	<i>Sin importancia</i>	<i>Incremental</i>	<i>Moderado</i>	<i>Medio</i>	<i>Significativo</i>	<i>Esencial</i>
1. ¿Requiere el sistemas copias de seguridad y de recuperación fiable?						X
2. ¿Se requiere comunicación de datos especializadas para transferir información a la aplicación u obtenerlas de ellas?						X
3. ¿Existe funciones de procesos distribuidos?			X			
4. ¿Es critico el rendimiento?			X			
5. ¿Sera ejecutado el sistema en un entorno existente y fuertemente utilizado?				X		
6. ¿Requiere el sistema entrada de datos interactiva?						X
7. ¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas o variadas operaciones?				X		
8. ¿Se actualiza los archivos de forma interactiva?					X	
9. ¿Son complejas las entradas, salidas, los archivos o las peticiones?				X		
10. ¿Es complejo el procesamiento interno del sistema?						X
11. ¿Se ha diseñado el código para ser reutilizado?					X	
12. ¿Están incluidas en el diseño la conversión y la instalación?						X
13. ¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?						X
14. ¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizados por el usuario?						X
Cuenta Total	$\Sigma fi=56$					

Elaboración: Propia

Para el ajuste se utiliza la ecuación:

$$PF = \text{Cuenta Total} * (\text{Grado de confiabilidad} + \text{Tasas de Error} * \Sigma(fi))$$

$$\text{Punto Función (PF)} = 306 + (0,65 + 0,01 * 56)$$

$$\text{Punto Función Resultante (PF)} = 307,2$$

Para hallar el punto función ideal al 100% de los factores seria 70:

$$PF = 324 + (0,65 + 0,01 * 70)$$

$$PF \text{ IDEAL} = 325,35$$

Calculando del % de funcionalidad real:

$$PF \text{ Real} = (307,2 / 325,35) * 100 = 94$$

Por tanto:

$$\boxed{\text{Funcionalidad} = 94\%}$$

Interpretando, la aplicación tiene una funcionalidad o utilidad del 94% para la empresa, lo que indica que el sistema cumple con los requisitos funcionales de forma satisfactoria.

3.6.3. Confiabilidad. La confiabilidad permite evaluar la relación entre el nivel de funcionalidad y la cantidad de recursos usados, es decir, representa el tiempo que el software está disponible para su uso, la misma se calcula utilizando la privacidad de que un sistema presente fallas:

- Comportamiento con respecto al tiempo: Atributos de software relativos a los tiempos de respuesta y de procesamiento de los datos.
- Comportamiento con respecto a Recursos: Atributos software relativo a la cantidad de recursos usados y la duración de su uso en la realización de funciones.

La función a continuación muestra el nivel de confiabilidad del sistema:

$$f(t) = (\text{Funcionalidad}) * e^{-\lambda t}$$

Se ve el trabajo hasta que se observa un fallo en un instante t , la función es la siguiente:

$$\text{Probabilidad de hallar una falla: } (T \leq t) = F(t)$$

Probabilidad de no hallar una falla: $(T > t) = 1 - F(t)$

Donde:

Funcionalidad = 0,94

$\lambda = 0.14$ (1 error cada 7 ejecuciones)

Tomemos un tiempo t de 12 meses

Ahora hallando la confiabilidad del sistema:

$$F(12) = (0,94) * e(-0,14*12)$$
$$F(12) = 0,175$$

Remplazando en la fórmula de no hallar una falla se tiene:

$$(T > t) = 1 - F(t)$$

$$(T > t) = 1 - 0,175$$

$$(T > t) = 1 - 0,175$$

$$(T > t) = 0,82$$

Con este resultado podemos decir que la probabilidad que el sistema no presente fallas es de 0,82 y que presente fallas es del 0,18.

$$\boxed{\text{Confiabilidad} = 82\%}$$

Con este resultado se concluye que la aplicación tiene un grado de confiabilidad del 82% durante los próximos 12 meses.

3.6.4. Mantenibilidad.La Mantenibilidad se refiere a los atributos que permiten medir el esfuerzo necesario para realizar modificaciones al software, ya sea por la corrección de errores o por el incremento de funcionalidad. Para hallar mantenibilidad del sistema se utiliza el índice de madurez de software (IMS), que proporciona una indicación de la estabilidad de un producto de software.

Para calcular el índice hacen falta una serie de medidas anteriores:

- Mt** = Número de módulos en la versión actual.
- Fm** = Número de módulos en la versión actual que han sido modificados.
- Fa** = Número de módulos en la versión actual que han sido añadidos.
- Fe** = Número de módulos de la versión anterior que se han eliminado en la versión actual.

A partir de estas, el IMS se calcula de la siguiente forma:

$$IMS = \frac{[Mt - (Fa + Fm + Fe)]}{Mt}$$

$$IMS = \frac{[6 - (0 + 1 + 0)]}{6}$$

$$IMS = 0.83$$

La interpretación a este resultado establece un 83%, lo que indica que no requiere de mantenimiento inmediatamente.

$$\boxed{\text{Mantenibilidad} = 83\%}$$

3.6.5. Portabilidad. Capacidad del producto o componente de ser transferido de forma efectiva y eficiente de un entorno hardware, software, operacional o de utilización a otro. Esta característica se subdivide a su vez en las siguientes sub características:

- **Adaptabilidad.** Capacidad del producto que le permite ser adaptado de forma efectiva y eficiente a diferentes entornos determinados de hardware, software, operacionales o de uso.
- **Capacidad para ser instalado.** Facilidad con la que el producto se puede instalar y/o desinstalar de forma exitosa en un determinado entorno.
- **Capacidad para ser reemplazado.** Capacidad del producto para ser utilizado en lugar de otro producto software determinado con el mismo propósito y en el mismo entorno.

Para el cálculo de la portabilidad se tomó en cuenta la siguiente tabla que contiene las características que se mencionaron anteriormente.

Tabla 58. Cálculo de Portabilidad

<i>Factor de Portabilidad</i>	<i>Valor en %</i>
Puede ser transferido de un entorno a otro	100%
Se puede adaptar a otros ambientes con facilidad(Instituciones Similares)	100%
Es fácil de Instalar	100%
Es capaz de reemplazar a una aplicación similar	90%
Total	97%

Elaboración: Propia

La interpretación a este resultado significa que nuestra aplicación web tiene la capacidad de ejecutarse en diferentes sistemas operativos y así mismo fácilmente de integrarse a diferentes instituciones de Salud.

Portabilidad = 97%

Resultados

De acuerdo a los resultados obtenidos se puede establecer la calidad total del sistema en base a los parámetros medidos anteriormente. La calidad está directamente relacionada con el grado de satisfacción con el usuario que ingresa al sistema.

Tabla 59. Resultados Finales

<i>Características</i>	<i>Resultados</i>
Usabilidad	90%
Funcionalidad	94%
Confiabilidad	82%
Mantenibilidad	83%
Portabilidad	97%
Evaluación Total	89%

Elaboración: Propia

Evaluación de calidad total = 89%

3.7. Estimación de Costos - COCOMO

Para determinar el Costo Total del proyecto se tomará en cuenta el costo de software, costo de implementación de la aplicación y elaboración del proyecto.

3.7.1. Costo del Software desarrollado. Para determinar el costo de la aplicación se usa el modelo COCOMO II orientado en los puntos de función, para calcular el esfuerzo, necesitaremos hallar la variable KDLC (Kilo líneas de código), donde los PF son 307,2 y las líneas por cada PF equivalen a 67 según vemos en la tabla que se ilustra a continuación:

Tabla 60. Conversión de puntos de función

Programming language	LOC/FP (average estimate)
assembly language	320
C	128
Cobol	105
Fortran	105
Pascal	90
Ada	70
php	67
Java	31
object-oriented languages	30
fourth generation languages (4GLs)	20
code generators	15
spreadsheets (excel programming)	6
graphical languages (icons) (draw-a-program languages)	4

Fuente: (Taina, 2005)

De esta manera, tras saber que son 67 LDC por cada PF, el resultado de los KDLC será el siguiente:

$$KLDC = \frac{(PF * \text{Lineas de codigo por cada PF})}{1000}$$

$$KLDC = \frac{(307,2 * 67)}{1000}$$

$$KLDC = 20,582$$

En nuestro caso el tipo semi - acoplado será el más apropiado ya que el número de líneas de código es menor a los 5000, y además el proyecto tiene una complejidad media, por consiguiente, los coeficientes que usaremos serán las siguientes:

Tabla 61. Valores de las constantes de acuerdo al modelo COCOMO

Modo de desarrollo	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	Mes-Hombre (nominal)	Tiempo de desarrollo (nominal)
Orgánico	3.2	1.05	2.5	0.38	$E_i = 3.2 * KLOCS^{1.05}$	$T_d = 2.5 * E_i^{0.38}$
Semi-acoplado	3.0	1.12	2.5	0.35	$E_i = 3.0 * KLOCS^{1.12}$	$T_d = 2.5 * E_i^{0.35}$
Acoplado	2.8	1.20	2.5	0.32	$E_i = 3.2 * KLOCS^{1.05}$	$T_d = 2.5 * E_i^{0.32}$

Fuente: (Pressman, 2010)

Tabla 62. Conductores de costos

CONDUCTORES DE COSTE	VALORACIÓN					
	<i>Muy bajo</i>	<i>Bajo</i>	<i>Nominal</i>	<i>Alto</i>	<i>Muy alto</i>	<i>Extr. alto</i>
Fiabilidad requerida del software	0,75	0,88	1.00	1,15	1,40	-
Tamaño de la base de datos	-	0,94	1,00	1,08	1,16	-
Complejidad del producto	0,70	0,85	1.00	1,15	1,30	1,65
Restricciones del tiempo de ejecución	-	-	1.00	1,11	1,30	1,66
Restricciones del almacenamiento principal	-	-	1,00	1,06	1,21	1,56
Volatilidad de la máquina virtual	-	0,87	1,00	1,15	1,30	-
Tiempo de respuesta del ordenador	-	0,87	1.00	1,07	1,15	-
Capacidad del analista	1,46	1,19	1.00	0,86	0,71	-
Experiencia en la aplicación	1,29	1,13	1.00	0,91	0,82	-
Capacidad de los programadores	1,42	1,17	1.00	0,86	0,70	-
Experiencia en S.O. utilizado	1,21	1,10	1,00	0,90	-	-
Experiencia en el lenguaje de programación	1,14	1,07	1.00	0,95	-	-
Prácticas de programación modernas	1,24	1,10	1,00	0,91	0,82	-
Utilización de herramientas software	1,24	1,10	1.00	0,91	0,83	-
Limitaciones de planificación del proyecto	1,23	1,08	1.00	1,04	1,10	-

Fuente: (Castillo, 2017)

De igual manera se debe de hallar la variable FAE, la cual se obtiene mediante la multiplicación de los valores evaluados en los diferentes 15 conductores de coste que se observan en la tabla 63:

$$\text{FAE} = 1,15 * 1 * 0,85 * 1,11 * 1 * 1 * 1,07 * 0,86 * 0,82 * 0,70 * 1 * 0,95 * 1 * 0,91 * 1,08$$

$$\text{FAE} = 0,53508480$$

Justificación:

Atributos de software

- **Fiabilidad requerida del software:** Si se produce un fallo por la venta de un producto, o fallo en algún registro del historial médico, puede ocasionar grandes dificultades en el seguimiento de los pacientes (Valoración Alta).
- **Tamaño de la base de datos:** La base de datos de nuestra aplicación móvil será estándar (Valoración Nominal).
- **Complejidad del producto:** La aplicación no va a realizar cálculos complejos (Valoración Baja). **Atributos de hardware**
- **Restricciones del tiempo de ejecución:** En los requerimientos se exige alto rendimiento (Valoración Alta).
- **Restricciones del almacenamiento principal:** No existen restricciones al respecto (Valoración Nominal).
- **Volatilidad de la máquina virtual:** Se usarán cualquier plataforma indiferente del sistema de operativo (Valoración Nominal).
- **Tiempo de respuesta del ordenador:** Deberá ser amigable con el usuario (Valoración Alta).

Atributos del personal

- **Capacidad del analista:** Capacidad alta relativamente, debido a la experiencia en análisis en proyecto similar (Valoración Alta)
- **Experiencia en la aplicación:** Se tiene cierta experiencia en aplicaciones de esta envergadura (Valoración muy alta).

- **Capacidad de los programadores:** Teóricamente deberá tenerse una capacidad muy alta por la experiencia en anteriores proyectos similares (Valoración muy alta).
- **Experiencia en S.O. utilizado:** Con diseño en plataforma web la experiencia es a nivel usuario (Valoración Nominal).
- **Experiencia en el lenguaje de programación:** Es relativamente alta, dado que se controlan las nociones básicas y las propias del proyecto (Valoración Alta).

Atributos del proyecto

- **Prácticas de programación modernas:** Se usarán prácticas de programación mayormente convencional (Valoración Nominal).
- **Utilización de herramientas software:** Se usarán herramientas estándar que no exigirán apenas formación, de las cuales se tiene cierta experiencia (Valoración Alta).
- **Limitaciones de planificación del proyecto:** Existen pocos límites de planificación. (Valoración Baja).

Cálculo del esfuerzo del desarrollo:

$$E = a \text{ KLDC } e * \text{FAE} = 2,4 * (16,089)^{1,05} * 0,53508480 = 36,35 \text{ [pers/mes]}$$

Cálculo tiempo de desarrollo:

$$T = c \text{ Esfuerzo } d = 2,5 * (36,35)^{0,35} = 8.77 \text{ [meses]}$$

Personal promedio:

$$P = E/T \quad P=36,35/8,77 = 4.01 \text{ personas}$$

Según este resultado será necesario un equipo de 4 personas trabajando alrededor de 9 meses.

Ahora pues sabemos que el salario promedio de un programador oscila entre 4000 Bs y 8200 Bs, estimando con el costo mínimo tomaremos 5000 Bs para calcular el costo de la aplicación.

Costo de la aplicación = Numero de programadores*Salario de un programador

Costo de la aplicación = 4*5000 Costo de la aplicación = 20000 Bs

3.7.2. Costo de implementación de la aplicación.Debido a que la aplicación está desarrollada con software libre y que la misma es utilizada por diferentes instituciones será de libre distribución solo para los encargados de administración de los centros de salud.

3.7.3. Costo de elaboración del proyecto.Por otra parte, el desarrollo como tal del proyecto comprende la siguiente serie de gastos que se detallan a continuación: Costo elaboración del proyecto con los requisitos para su elaboración

Tabla 63. Resultados Finales

<i>Detalle</i>	<i>Importe en bolivianos</i>
Análisis y diseño de la aplicación	7000
Material de escritorio	700
Investigación en internet	800
Otro	2100
Total	10600

Elaboración: Propia

Costo de elaboración del proyecto = 10600 bs.

3.7.4. Costo total del proyecto.El costo total del proyecto es la sumatoria del costo de la aplicación desarrollada, costo de implementación de la aplicación y costo de elaboración del proyecto.

Tabla 64. Resultados Finales

<i>Descripción</i>	<i>Costo total en bolivianos</i>
Costo de la aplicación	20000
Costo de implementación de la aplicación	0
Costo de Elaboración del Proyecto	10600
Total	36000

Elaboración: Propia

Costo total del proyecto = 36000 bs.

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

RESUMEN

En este capítulo se introduce a la temática de investigación propuesta, se describe el problema, los objetivos que conducen a la solución del problema, herramientas de software empleadas, los alcances, las limitaciones para el desarrollo de software.

En este capítulo se introduce a la temática de investigación propuesta. se describe el problema.

4.1. Conclusiones

Se han cumplido con los objetivos planteados realizando los siguientes módulos: módulo de inicio de sesión, módulo de administración de usuarios, módulo de asignación de ficha de atención, módulo de registro de triaje, módulo de gestión de historia clínica y módulo de generación de reportes.

Se logro implementar el diseño de formularios de forma dinámica, permitiendo que se pueda distinguir un diferente historial clínico de acuerdos a las necesidades de cada especialidad y sus características. Siendo estos posibles de modificación y actualización por parte del usuario.

El sistema permite mejorar las tareas de gestión de historias clínicas de manera rápida y confiable, reduciendo de esta manera el tiempo en el registro de datos y manejo de esta información.

Se logro estandarizar y almacenar la información de las historias clínicas de forma estructurada en un repositorio central evitando la duplicidad de la información y asegurando la disponibilidad de la misma por parte de los usuarios (profesionales de salud y personal administrativo), mejorando la gestión de las historias clínicas.

Se obtuvo un software multiplataforma fácil de utilizar, el cual puede ser accedido desde dispositivos tales como computadoras de escritorio, laptop, tablets y teléfonos celulares. Este acceso instantáneo desde cualquier punto, presenta mayor comodidad a la hora de consultar y editar historias clínicas, como la facilidad de atención a domiciliaria.

El sistema posee una interfaz gráfica amigable, que permite un manejo fácil e intuitivo para todo el personal médico de los consultorios.

Se redujo el tiempo en la búsqueda de pacientes y su historial clínico, así mismo también se redujo el tiempo en la generación de reportes.

4.2. Recomendaciones

Al haber concluido el presente proyecto de grado, titulado “Sistema web para la gestión de historias clínicas electrónicas personalizables por especialidad para el colegio médico departamental de La Paz”, se tiene las siguientes recomendaciones para un buen funcionamiento:

Para garantizar la seguridad de la información, se recomienda la generación de backups, diarios, semanales o mensuales.

Para asegurar la integridad de los datos e información contenida en el sistema, se recomienda a los usuarios del sistema, políticas de seguridad, como ser el cambio de contraseña asignado por defecto en el sistema, la longitud de las contraseñas, y el cuidado de las cuentas que hayan iniciado sesión, para evitar el acceso de personas ajenas y maliciosas al sistema.

Una vez que el personal concluya sus labores en el consultorio se recomienda deshabilitar al usuario para que ya no pueda realizar modificación alguna en la información registrada.

Se recomienda realizar el mantenimiento al sistema web, para mantener el buen desempeño del sistema y prevenir posibles fallas.

Se recomienda tomar la posibilidad de crear más módulos para de este modo escalar el sistema y que se convierta en un sistema de gestión hospitalaria completa.

BIBLIOGRAFÍA

- Araque, M. (8 de Febrero de 2017). *Metodología Scrum: qué es y cómo funciona*. Obtenido de we are marketing: <https://www.wearemarketing.com/es/blog/metodologia-scrum-que-es-y-como-funciona.html>
- Aspiazu Gutierrez, A. A. (2016). *Sistema de gestión de exámenes clínicos y enfermedades de la altura*. Proyecto de Grado, Universidad Mayor de San Andres "UMSA".
- Borbon Ardilla, N. I. (12 de Marzo de 2013). *Norma de Evaluación ISO/IEC 9126*. Obtenido de Evaluación de Software: <http://actividadreconocimiento-301569-8.blogspot.com/2013/03/norma-de-evaluacion-isoiec-9126.html>
- Centellas Coarite, M. (2015). *"Sistema de Control y Gestión de Historiales Clínicos apoyado en dispositivos móviles" Caso: Centro Médico La Paz*. Proyecto de Grado, Universidad Mayor de San Andres "UMSA", La Paz - Bolivia.
- Chimarro Chipantiza, V. L., Mazon Olivo, B. E., & Cartuche Calva, J. J. (2015). *La usabilidad en el desarrollo de software*. Obtenido de ResearchGate: https://www.researchgate.net/publication/318280962_La_usabilidad_en_el_desarrollo_de_software
- De los Santos, C. (2 de Julio de 2019). *¿Qué es la norma ISO 9126?* Obtenido de lecasabe Capacitación Práctica: <http://lecasabe.com/que-es-la-norma-iso-9126/>
- EcuRed. (2011). *Metodologías de desarrollo de Software*. Recuperado el 2019, de https://www.ecured.cu/Metodologias_de_desarrollo_de_Software
- Goncalves, L. (25 de Enero de 2019). *Qué es la metodología SCRUM, todo lo que necesitas saber*. Recuperado el 2019, de <https://luis-goncalves.com/es/que-es-la-metodologia-scrum/>
- González Alcolado, A. J., & Crespo Obaco, J. P. (2018). *Sistema informático para la gestión de las historias clínicas en los estudios de imagen médicas*. Trabajo de Titulación, Pontificia Universidad Católica del Ecuador, Santo Domingo - Ecuador.
- Gutierrez Flores, P. (2017). *"Sistema web de Administración de Historias Clínicas" Caso: Centro Médico Quirúrgico Erzengel*. Proyecto de Grado, Universidad Mayor de San Andres "UMSA", La Paz - Bolivia.
- Huanca Cantuta, S. P. (2015). *Sistema web de control de pagos, citas e historiales clínicos "Caso: Clínica Dental Lavadent"*. Proyecto de Grado, Universidad Mayor de San Andres "UMSA", La Paz.
- La Razon. (28 de Octubre de 2018). *Gobierno proyecta digitalizar las historias clínicas de pacientes y plan de control médico anual*. Recuperado el Octubre de 2019, de http://www.la-razon.com/sociedad/Gobierno-digitalizar-historias-clinicas-pacientes_0_3028497139.html
- Lozano Flores, R. (2014). *Sistema de administración y control de Historiales Clínicos para los consultorios de la UMSA*. Proyecto de Grado, Universidad Mayor de San Andres "UMSA", La Paz - Bolivia.
- Mamani Llipes, D. A. (2014). *Sistema web de registro y seguimiento de pacientes, elaboración y emisión de análisis efectuados, registro y guía médica, farmacéutica y hospitalaria. Caso: Laboratorio Clínico - Adolfo Kolping*. Proyecto de Grado, Universidad Mayor de San Andres "UMSA", La Paz - Bolivia.

- Infoleyes. (2016). *Reglamento de Establecimientos de Salud Públicos y Privados*. Obtenido de <https://bolivia.infoleyes.com/norma/3433/reglamento-de-establecimientos-de-salud-p%C3%BAblicos-y-privados-respp>
- cidbimena. (2010). *Aspectos funcionales de establecimientos de salud*. Obtenido de <http://cidbimena.desastres.hn/docum/crid/PPSED/PDF/doc18025/doc18025-f.pdf>
- conexionesan. (20 de Marzo de 2018). *¿Cómo funciona la categorización en establecimientos de salud?* Obtenido de <https://www.esan.edu.pe/apuntes-empresariales/2018/03/como-funciona-la-categorizacion-en-establecimientos-de-salud/>
- SlideShare. (14 de Septiembre de 2012). *Categorización de los establecimientos de salud*. Obtenido de <https://es.slideshare.net/juantolenb/categorizacion-de-los-establecimientos-de-salud>

ANEXOS