

UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

“SISTEMA INTEGRAL PARA EL CONTROL Y ADMINISTRACIÓN DE INSUMOS MÉDICOS Y PERSONAL ADMINISTRATIVO”

CASO: CLINICA DE ESPECIALIDADES ADOLFO KOLPING.

Para optar al título de Licenciatura en Ingeniería de Sistemas

MENCIÓN: INFORMÁTICA Y COMUNICACIONES

Postulante: Univ. Swania Betty Guarachi Velasco

Tutor Metodológico: Lic. Ing. Dionicio Henry Pacheco Ríos

**Tutor Revisor: M.Sc. Lic. Ing. Juan Fernando Chambi
Guachalla**

Tutor Especialista: Ing. Leandro Poma Alanoca

EL ALTO – BOLIVIA

2023

**DECLARACIÓN JURADA DE
AUTENTICIDAD Y RESPONSABILIDAD**

Yo, **SWANIA BETTY GUARACHI VELASCO** estudiante con C.I. 9937741LP mediante la presente **declaro** de manera pública que la propuesta del **TRABAJO DE GRADO** titulada “**SISTEMA INTEGRAL PARA EL CONTROL Y ADMINISTRACION DE INSUMOS MEDICOS Y PERSONAL ADMINISTRATIVO**” es original, siendo resultado de mi trabajo personal y no constituye una copia o replica de trabajos similares elaborados,

Autorizo la publicación del resumen de mi propuesta en internet y me comprometo a responder a todos los cuestionamientos que se desprenden de su lectura.

Asimismo, me hago responsable ante la universidad o terceros, de cualquiera irregularidad o daño que pudiera ocasionar, por el incumplimiento de lo declarado.

De identificarse falsificación, plagio, fraude, o que el **TRABAJO DE GRADO** haya sido publicado anteriormente; asumo las consecuencias y sanciones que de mi acción se deriven, responsabilizándome por todas las cargas legales que se deriven de ello someténdome a las normas establecidas y vigentes de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

El Alto, noviembre del 2023.

Swania Betty Guarachi Velasco

C.I. 9937741 LP

e-mail:vani.bet@hotmail.com

DEDICATORIA

El presente proyecto va dedicado con mucho cariño a mis a mis queridos padres German Guarachi García, Graciela Velasco Rengel y todas las personas que creyeron en mí y que siempre me dieron todo su apoyo y colaboración incondicional.

AGRADECIMIENTOS

Agradezco a Dios A Dios, por darme la inspiración, la vida y el conocimiento necesario en mi diario vivir, agradezco a toda mi familia, por el apoyo que me brindo a lo largo de toda mi vida, e impulsarme a seguir y no rendirme.

A mi tutor Metodológico Ingeniero Dionicio Henry Pacheco Ríos por ser una persona extraordinaria me brindo su conocimiento, enseñanza, tiempo, paciencia, comentarios y observaciones que me permiten culminar el presente proyecto.

A mi tutor Especialista Ingeniero Leandro Poma Alanoca por compartir sus conocimientos, brindarme sus orientaciones, sugerencias con paciencia motivación durante el desarrollo del presente proyecto.

A mi tutor Revisor Ingeniero Juan Fernando Chambi Guachalla por la colaboración y la orientación en el desarrollo de la documentación.

Al director Administrativo de la Clinica de especialidades Adolfo Kolping” por la colaboración, hoy en día el sistema implementado con éxito.

A la Universidad Pública de El Alto la carrera Ingeniería de Sistemas por acogerme en sus ambientes todos estos años de estudio.

INDICE DE CONTENIDO

	Página
1. MARCO PRELIMINAR	1
1.1. Introducción	1
1.2. Antecedentes	2
1.2.1. Institucionales	2
1.2.2. Académicos.....	2
1.3. Planteamiento del Problema.....	3
1.3.1. Principal	4
1.3.2. Secundarios	4
1.4. Objetivos.....	5
1.4.1. General	5
1.4.2. Específicos.....	6
1.5. Justificación	7
1.5.1. Técnica.....	7
1.5.2. Económica	7
1.5.3. Social	7
1.6. Metodología	8
1.6.1. Metodología de Ingeniería.....	8
1.6.2. Métrica de calidad al software	9
1.6.3. Medición y Estimación.....	9
1.6.4. Seguridad.....	9
1.6.5. Pruebas al software.....	10
1.7. Herramientas	11

1.7.1. Servidor Web Apache	11
1.7.2. Gestor de Base de Datos PostgreSQL.....	11
1.7.3. Herramientas para el desarrollo	11
1.8. Límites y Alcances.....	14
1.8.1. Límites.....	14
1.8.2. Alcances.....	14
1.9. Aportes	15
2. MARCO TEORICO	17
2.1. Introducción	17
2.2. Sistema.....	17
2.3. Sistema Integral.....	18
2.4. Administración	18
2.5. Control	18
2.6. Control Administrativo.....	19
2.7. Inventario.....	19
2.8. Stock.....	19
2.9. Stock mínimo	20
2.10. Stock disponible.....	20
2.11. Sistema de Inventarios/Insumos.....	20
2.12. Sistema de Información	21
2.13. Almacén.....	21
2.14. Modelo de ciclo de vida	22
2.14.1. Ciclo de vida iterativo	22

2.15. Metodología de desarrollo de software	23
2.15.1. UWE - Ingeniería web basada en UML	23
2.16. Seguridad de aplicaciones web	31
2.17. Modelo vista controlador (MVC)	32
2.17.1 Modelo.....	33
2.17.2 Vista	33
2.17.3 Controlador.....	33
2.17.4 Características del Modelo Vista Controlador.....	33
2.18. Calidad del software	34
2.18.1. Factores de calidad ISO/IEC 9126	34
2.19. Seguridad de la Información	44
2.19.1. ISO/IEC 27002	44
2.20. Pruebas de software	45
2.20.1. Pruebas de Caja Blanca.....	45
2.20.2. Pruebas de Caja Negra.....	46
2.20.3. Pruebas de estrés	47
2.21. Medición y Estimación de Software	48
2.21.1. Estimación de software con COSMIC	48
2.22. Herramientas	55
2.22.1. Servidor Web Apache	55
2.22.2. Gestor de Base de Datos PostgreSQL.....	56
2.22.3. PHP.....	58
2.22.4. Laravel	59

2.22.5. Livewire	60
2.22.6. Tailwind Css	61
2.22.7. Alpine Js.....	62
2.22.8. MagicDraw	62
3. MARCO APLICATIVO	65
3.1. Introducción	65
3.2. Obtención de requisitos	65
3.2.1. Descripción de los actores	66
3.3. Análisis de Requerimientos	67
3.3.1. Requisitos funcionales	67
3.3.2. Requisitos No Funcionales.....	71
3.3.3. Modelo de Requisitos.....	71
3.4. Diseño del Sistema	76
3.4.1. Modelo de Contenido	76
3.4.2. Modelo de Navegación.....	78
3.5. Modelo de Presentación.....	81
3.5.1. Modelo de Presentación: Login	81
3.5.2. Modelo de Presentación: Página de inicio	82
3.5.3. Modelo de Presentación: Administrador	83
3.5.4. Modelo de Presentación: rrhh.....	84
3.5.5. Modelo de Presentación: Encargado de insumos	85
3.5.6. Modelo de Presentación: Solicitante	86
3.6. Codificación del Sistema.....	87

3.7. Implementación	89
3.8. Interfaz Inicio de Sesión.....	89
3.8.1. Interfaz de inicio	90
3.8.2. Módulos que integran el sistema	91
4. CALIDAD DEL SOFTWARE, SEGURIDAD Y ESTIMACIÓN DE COSTOS	101
4.1. Métricas de Calidad del Software Mediante la Norma ISO/IEC 9126	101
4.1.1. Funcionalidad	101
4.1.2. Confiabilidad	105
4.1.3. Usabilidad	106
4.1.4. Mantenibilidad	108
4.1.5. Portabilidad	109
4.2. Estimación de Software con COSMIC	110
4.3. Sistema de Gestión de Seguridad de la Información ISO-27002.....	111
4.3.1. Seguridad Lógica	112
4.3.2. Seguridad Física	113
4.3.3. Seguridad Organizativa	113
4.4. Pruebas de Software	113
4.4.1. Pruebas de Caja Blanca.....	113
4.4.2. Pruebas de Caja Negra.....	116
4.4.3. Pruebas de estrés	120
5. Conclusiones y Recomendaciones	122
5.1. Conclusiones	122
Bibliografía	124

INDICE DE TABLAS

Tabla 1 Estereotipo de modelo de requisitos	26
Tabla 2 Estereotipo de modelo de navegación	29
Tabla 3 Estereotipo de modelo de presentación.....	30
Tabla 4 Técnica de obtención de requisitos.....	65
Tabla 5 Requisitos Funcionales	67
Tabla 6 Requisitos No Funcionales	71
Tabla 7 Caso de Uso Perfil Administrador	74
Tabla 8 Caso de Uso Perfil RRHH.....	75
Tabla 9 Caso de Uso Perfil Encargado de Insumos	76
Tabla 10 Características de Funcionalidad.....	101
Tabla 11 Parámetros para la medición	102
Tabla 12 Cálculo de puntos de función sin ajustar.....	102
Tabla 13 Parámetros de medición (Factores de Ponderación).....	103
Tabla 14 Ajuste de preguntas para determinar la usabilidad	107
Tabla 15 Copias de seguridad	112
Tabla 16 Identificación de las clases de equivalencia de registro de ingreso de artículos	117
Tabla 17 Caso de prueba de equivalencia valida de registro de ingreso de artículos	117
Tabla 18 Caso de prueba de equivalencia invalida de registro de ingreso de articulo	117
Tabla 19 Identificación de las clases de equivalencia registro de artículos	119

INDICE DE FIGURAS

Figura 1 Grafica general del sistema.....	17
Figura 2 Modelo de Ciclo de Vida Iterativo.....	22
Figura 3 Lenguaje de Modelado Unificado.....	24
Figura 4 Diagrama de casos de uso.....	27
Figura 5 Modelo de contenido.....	27
Figura 6 Modelo de Navegación.....	29
Figura 7 Modelo de Presentación.....	30
Figura 8 Características de la iso 9126.....	35
Figura 9 Características de la funcionabilidad.....	36
Figura 10 Características de la Confiabilidad.....	38
Figura 11 Características de la Usabilidad.....	39
Figura 12 Característica de Eficiencia.....	40
Figura 13 Característica de Mantenibilidad.....	41
Figura 14 Característica de Portabilidad.....	42
Figura 15 Características de la Calidad en Uso.....	44
Figura 16 Estructura de dominios de ISO/IEC 27002.....	45
Figura 17 Representación de pruebas de Caja Blanca y Caja Negra.....	47
Figura 18 Visión general del COSMIC.....	50
Figura 19 El proceso de medición COSMIC.....	51
Figura 20 Movimientos de datos.....	53
Figura 21 Tendencia PHP (Escala logarítmica).....	59

Figura 22	Diagrama de caso de uso general del sistema.....	72
Figura 23	Diagrama de Caso de Uso Perfil Administrador	73
Figura 24	Diagrama Caso de Uso Perfil RRHH.....	74
Figura 25	Diagrama Caso de Uso Perfil Encargado de Insumos	75
Figura 26	Diagrama de Contenido.....	77
Figura 27	Diagrama de Navegación Administrador	78
Figura 28	Diagrama de Navegación Recursos Humanos.....	79
Figura 29	Diagrama de Navegación Encargado de Insumos	80
Figura 30	Diagrama de Navegación inicio de sesión.....	81
Figura 31	Diagrama de Navegación Menú principal.....	82
Figura 32	Diagrama de presentación de administrador	83
Figura 33	Diagrama de presentación de recursos humanos	84
Figura 34	Diagrama de presentación de encargado de insumos	85
Figura 35	Diagrama de presentación del solicitante.....	86
Figura 36	Codificación de controlador	87
Figura 37	Codificación de controlador-Item	87
Figura 38	Ccodificación de vista.....	88
Figura 39	Codificación de vista-item.....	88
Figura 40	Diagrama de Navegación inicio de sesión.....	89
Figura 41	Autenticación del sistema.....	90
Figura 42	Pantalla de Inicio	90
Figura 43	Registro de usuarios.....	91
Figura 44	Listado de usuarios	91

Figura 45 Perfil de usuario.....	92
Figura 46 Listado de cargos	92
Figura 47 Listado de artículos	93
Figura 48 Alerta para eliminar registro de articulo/insumo	93
Figura 49 Reporte movimiento articulo.....	94
Figura 50 Salidas de artículos	95
Figura 51 Registro de ingreso	95
Figura 52 Listado del personal	96
Figura 53 Registro del personal.....	96
Figura 54 Registro de ingreso del personal.....	97
Figura 55 Registro de solicitud	98
Figura 56 Listado de solicitudes	98
Figura 57 Reporte Kardex	99
Figura 58 Grafo de flujo creacion de registro de ingreso.....	114
Figura 59 Prueba de caja negra de registro de ingreso.....	116
Figura 60 Prueba de caja negra de registro de articulo.....	118

RESUMEN

En la actualidad, la información se reconoce como un recurso invaluable para las organizaciones, y el sistema de información desempeña un papel fundamental en su gestión, asegurando su manejo de manera efectiva y eficiente.

El presente proyecto de grado, se presenta como alternativa de solución a estos problemas de crecimiento a través del desarrollo de un “Sistema integral para el control y administración de insumos médicos y personal administrativo” la cual pretende automatizar los procesos que en la clínica se realiza, para poder acceder de manera inmediata a la información de los artículos/insumos, ya que estos procesos son de vital importancia para la clínica y es necesario controlar la información que se genera día a día.

Para lograr el desarrollo del proyecto, se ha hecho uso de la metodología UWE - UML Based Web Engineering haciendo uso de algunas de las herramientas principales como el servidor web Apache, el gestor de base de datos PostgreSQL, PHP, Laravel, Livewire, Tailwind CSS y Alpine JS.

Para evaluar la calidad del software se utilizó ISO 9126 y finalmente para la estimación del costo de producto se usó COSMIC basado en la medición del tamaño de los requerimientos funcionales.

Observando el caso de la clínica de especialidades Adolfo Kolping y su exitosa implementación de un Sistema Integral para el control y administración de insumos médicos y personal administrativo, podemos deducir varias conclusiones.

LISTADO DE ABREVIATURAS

API: Interfaz de programación de aplicaciones.

ASF: Apache Software Foundation.

COSMIC: Consorcio Internacional de Medición Común del Software (Common Software Measurement International Consortium)

CSS: Hojas de Estilo en Cascada (Cascading Style Sheets)

FUR: Requisitos Funcionales de los Usuarios.

FTP: Protocolo de Transferencia de Archivos(File Transfer Protocol).

HTML: Lenguaje de Marcas de Hipertexto.

HTTP: Protocolo de transferencia de hipertextos (Hypertext Transfer Protocol)

ISO: Organización Internacional de Normalización (International Organization for Standardization).

MVC: Modelo Vista Controlador (Multi-View Controller).

MVCC: Control de concurrencia de versiones múltiples.

PHP: Pre - procesador de hipertexto. (Hypertext Pre-Processor.).

PGSQL: PostgreSQL

PITR: Point-in-time-recuperación.

SGSI: sistema de seguridad de la información (Information Security Management System).

SQL: Lenguaje de Consulta Estructurada (Structured Query Language).

UML: lenguaje unificado de modelado (Unified Modeling Language).

UWE: Lenguaje Unificado de Modelado (UML-based web engineering).

TIC: Tecnologías de la Información y la Comunicación.

URL: Localizador Uniforme de Recursos (Uniform Resource Locator).

CAPITULO I

1. MARCO PRELIMINAR

1.1. Introducción

En las últimas décadas, el área de la salud ha experimentado avances tecnológicos significativos que han transformado la forma en que se brinda atención médica. Estos avances han sido posibles gracias al rápido desarrollo de la tecnología y su aplicación en el campo médico. La introducción de sistemas y dispositivos tecnológicos ha permitido mejorar el diagnóstico, el tratamiento y la gestión de los recursos en el sector de la salud. El desarrollo de sistemas web específicos para el control y gestión de insumos médicos es un ejemplo de cómo la tecnología ha impactado positivamente en este ámbito.

El enfoque principal del presente proyecto de grado es la automatización de los procesos de la unidad de Almacén de Insumos Médicos al mismo tiempo realizar el control de personal médico/administrativo. El propósito es implementar un sistema que controle de manera eficiente los ingresos y egresos de los insumos médicos, mantenga un adecuado control del stock, emita inventarios y ofrezca diversas opciones que agilicen el manejo de la información. Para lograr el objetivo propuesto, se hará uso de nuevas tecnologías que permitan obtener información confiable, facilitando así el seguimiento y control de medicamentos e insumos, así como del personal administrativo. La implementación de este sistema no solo reducirá significativamente el tiempo invertido en el manejo de la información de los artículos, sino que también brindará soluciones a los problemas que se presentan, cumpliendo así con los requerimientos de la institución de manera efectiva.

Dicho proyecto servirá para proporcionar información precisa y oportuna sobre los artículos, y permitirá un control efectivo del personal administrativo a través de Internet. Esto permitirá que todo el personal realice consultas y reservas de insumos de acuerdo a sus necesidades de manera más eficiente y accesible.

1.2. Antecedentes

1.2.1. Institucionales

La Fundación Padre Adolfo Kolping (Inicialmente Obra Kolping Bolivia) es una entidad que se instituyó en Bolivia desde el año 1983, la misma a lo largo del tiempo vino realizando múltiples proyectos en el territorio boliviano. Su carácter y fundamentos la hacen una Institución peculiar, que busca la auto-superación individual y fortalecimiento de la sociedad civil. Este proyecto productivo de la Fundación Padre Adolfo Kolping, nace a la vida institucional un 16 de mayo de 1994, aunque se tiene conocimiento de que estuviese funcionando unos meses antes en los terrenos de lo que actualmente es el Colegio Kolping de Villa Dolores con un consultorio de Medicina General y de Odontología. Hoy en día la clínica queda ubicada en la calle Constantino Medina entre calles 7 y 8 zona villa dolores.

1.2.2. Académicos

- (Arias, 2019) **“Implementación de un sistema integrado de suministro de medicamentos e insumos médicos para mejorar el sistema de gestión de almacén en una cadena farmacéutica”**. Universidad Tecnológica del Perú. Plantea como objetivo implementar un Sistema integrado de suministro de medicamentos e insumos médicos para una farmacia, mediante herramientas de programación en software como ser: C# para el lenguaje de programación, SQL Server 2014 como gestor de base de datos, IIS como servidor web y Microsoft Visual Studio 2013.
- (Carrasco, 2015) **“Desarrollo de sistema para la gestión y control de inventario de productos ofrecidos por la Farmacia Veterinaria BioBío”**. Universidad del Bío Bío, Chile. El objetivo que plantea es Diseñar e implementar un sistema software para la gestión y control de inventario y registro de compra/venta de productos ofrecidos por la farmacia veterinaria Bío-Bío, la Metodología y herramienta: metodología iterativa incremental., herramientas Yii

1.1.15: PHP Framework, Sybase PowerDesigner 16, Apache versión 2.4.7, MySQL versión 5.6.16, Sublime text 2, Php 5.3, PhpMyAdmin 4.1.6, Yii-bootstrap-2.1.0.

- (Salamanca, 2017) “**SISTEMA WEB PARA EL CONTROL DE: INSUMOS Y PRODUCTOS DEL ÁREA DE PRODUCCION Y ALMACENES**”. Universidad Mayor de San Andrés (UMSA), plantea como objetivo desarrollar un Sistema Web que proporcione información oportuna y confiable del área de producción y almacenes de manera que optimice la administración de la información. Utilizando así la metodología ágil AUP (Agile Unified Process), para el desarrollo de la aplicación web utiliza el lenguaje PHP, y servidor de base de datos MySQL.
- (Calle, 2020) “**Portal web para el Control de Almacenes y Activos Fijos**”. Universidad Pública de El Alto. plantea el siguiente objetivo, el de dar un nuevo servicio a la administración en el sector de almacenes controlando las entradas de (reactivos, embalajes, material de vidrio, plástico, cartón, etiquetas, para producto terminado. se apoya en las metodologías, SCRUM, UWE para iniciar la primera fase de desarrollo, y para luego hacer el uso de las herramientas PHP, MYSQL, BOOSTSTRAP, JAVA SCRIPT, HTML, AJAX. También hace uso de la arquitectura de software modelo vista controlador (MVC), la cual es muy utilizada en las aplicaciones web.

1.3. Planteamiento del Problema

En la clínica de especialidades Adolfo Kolping, se ha identificado un problema en los procesos administrativos como el control y administración de insumos y personal administrativo, el registro, la recepción, revisión y entrega de diferentes tipos de insumos. Actualmente, se utiliza un sistema manual de registros, lo que resulta en un trabajo lento y complicado.

El proceso de recepción de insumos implica controlar y registrar información como el origen (proveedores) y procedencia de los insumos. Además, se requiere mantener

un conjunto de documentos de constancia. Sin embargo, la falta de una información precisa y actualizada dificulta la tarea de proporcionar la información necesaria a los empleados(médicos/personal). Esto ha generado molestias y confusiones, afectando la eficiencia en el desempeño de sus funciones.

El problema radica en la falta de un sistema adecuado para el registro y control del personal administrativo y para los insumos médicos, así como la ausencia de una gestión eficiente de la documentación. Esto ha generado retrasos, dificultades en la obtención de información precisa y actualizada, y una disminución en la eficiencia general de los procesos administrativos.

1.3.1. Principal

La clínica de especialidades Adolfo Kolping, realiza el proceso de registro, control y manipulación de datos de manera manual y con algunas tablas Excel poco confiables e ineficientes, que al registrar o actualizar los datos genera mucha inconsistencia, los reportes de información son morosos e incluso existe pérdida de información y descontento de los usuarios.

1.3.2. Secundarios

- No contar con reportes precisos ni historial de insumos/artículos médicos, genera dificultades para toma de decisiones basadas en datos concretos y actualizados.
- Ineficiencia en el registro, búsqueda y control del movimiento de entradas y salidas de artículos en los almacenes debido a procesos manuales, lo cual genera demoras en la entrega de información y en el trabajo.
- Control inadecuado de la existencia de artículos en los almacenes, lo que provoca pedidos innecesarios debido a un seguimiento inoportuno del stock, y además genera retrasos en las actividades del personal.

- Realizar solicitudes de insumos/artículos de manera manual, genera pérdida de tiempo, solicitudes innecesarias, posibles errores humanos y demoras en las aprobaciones.
- La ausencia de alertas y envíos de correos electrónicos representa un obstáculo para la comunicación efectiva acerca del estado de las solicitudes, dando lugar a desconocimiento y retrasos en la toma de decisiones.
- La gestión ineficiente de los datos del personal administrativo impacta negativamente en la generación de informes o reportes, dificultando la obtención de datos precisos y oportunos sobre el personal.
- La falta de una gestión adecuada de permisos, vacaciones y correspondencias en el personal de la clínica genera desorganización, conflictos en la programación y afectando la continuidad del servicio.
- La ausencia de un proceso sistematizado para la asignación de turnos de fin de semana del personal administrativo genera ineficiencias en la planificación, falta de visibilidad y comunicación transparente entre los involucrados.

Después de hacer este análisis se plantea la siguiente interrogante ¿De qué manera se puede realizar el control y gestión de insumos/artículos médicos y personal administrativo, manteniendo toda la información precisa y confiable en los procesos de recepción, registro y distribución?

1.4. Objetivos

1.4.1. General

Desarrollar un sistema integral para el control y administración de insumos/artículos médicos y personal administrativo, con la finalidad de automatizar registros y simplificar procesos, garantizando disponibilidad de información precisa en la Clínica de especialidades Adolfo Kolping.

1.4.2. Específicos

- Desarrollar opciones de visualización, generación de reportes y seguimiento de insumos/artículos, brindando información detallada y confiable.
- Automatizar de manera precisa y eficiente los movimientos de entrada y salida de artículos, eliminando la necesidad de procesos manuales y reduciendo las demoras en la entrega de información.
- Automatizar el control del stock mínimo y disponible, evitando pedidos innecesarios.
- Automatizar la gestión de solicitudes de insumos/artículos con el propósito de optimizar el proceso, reducir la pérdida de tiempo, minimizar solicitudes innecesarias y prevenir errores humanos.
- Implementar opciones de envío de notificaciones y correos electrónicos automatizados en el proceso de gestión de solicitudes de insumos/artículos, con el fin de mejorar la comunicación interna y proporcionar información oportuna sobre el estado de las solicitudes.
- Implementar un módulo de gestión integral del personal administrativo que optimice la recopilación, organización y acceso a los datos, garantizando la generación eficiente de informes y reportes precisos para facilitar la toma de decisiones.
- Desarrollar opciones de gestión de permisos, vacaciones y correspondencias para el personal de la clínica, con el fin de mejorar la organización, minimizar conflictos en la programación y garantizar la continuidad eficiente de los servicios médicos.
- Implementar un proceso sistematizado para la asignación de turnos de fin de semana del personal administrativo, con el propósito de mejorar la eficiencia en la planificación.
-

1.5. Justificación

1.5.1. Técnica

El sistema integral a implementar brindara ventajas técnicas al controlar insumos y personal médico, a través de una plataforma centralizada que administrara inventarios en tiempo real, evitando pérdidas o escasez. Además, automatizara tareas como la gestión de horarios, reduciendo errores y carga manual. La clínica cuenta con equipos computacionales para optimizar procesos y generar informes actualizados.

1.5.2. Económica

Desde una perspectiva económica, la implementación de un sistema integral para el control y administración de insumos médicos y personal administrativo puede generar importantes ahorros de costos. Mediante una gestión más eficiente de los inventarios, se evitan compras innecesarias y se reducen las pérdidas por obsolescencia o vencimiento de los insumos. Asimismo, la automatización de tareas administrativas permite una asignación más efectiva de los recursos humanos, optimizando la productividad y evitando la necesidad de contratar personal adicional. En conjunto, estas medidas contribuyen a mejorar la eficiencia operativa y a reducir los gastos en la clínica.

1.5.3. Social

La implementación de un sistema integral para el control y administración de insumos médicos y personal administrativo también tiene un impacto social significativo. En primer lugar, mejora la calidad de la atención médica al garantizar la disponibilidad oportuna de los insumos necesarios para los procedimientos y tratamientos. Además, la automatización de tareas administrativas permite una atención más personalizada y enfocada en el paciente, ya que el personal puede dedicar más tiempo a brindar cuidado directo en lugar de realizar tareas administrativas. Este enfoque centrado en la paciente mejora la experiencia general de los pacientes y fortalece la relación

médico-paciente. En última instancia, el sistema contribuye a una gestión más efectiva de los recursos de la clínica, promoviendo una utilización óptima de los mismos y garantizando la sostenibilidad de los servicios de salud a largo plazo.

1.6. Metodología

La metodología es aquella guía que nos va revelando que hacer y cómo proceder cuando se quiere obtener una investigación. Es posible decir que la metodología es aquel enfoque que permite observar un problema de una forma total, sistemática, disciplinada y con cierta disciplina.

La metodología que se utilizara UWE – UML Based Web Engineering es un enfoque de ingeniería de software para el dominio web cuyo objetivo es cubrir todo el ciclo de vida del desarrollo de aplicaciones web. El aspecto clave que distingue a UWE es la dependencia de los estándares.

El enfoque principal del enfoque UWE es proporcionar un Lenguaje de modelado específico de dominio basado en UML que también incluye características de seguridad, metodología basada en modelos, soporte de herramientas para el diseño sistemático, y soporte de herramientas para la generación (semi) automática de aplicaciones web. (Orozco, 2012)

1.6.1. Metodología de Ingeniería

Las metodologías ágiles son una serie de técnicas para la gestión de proyectos que han surgido como contraposición a los métodos clásicos de gestión como CMMI (Modelo de Capacidad y Madurez). Aunque surgieron en el ámbito del desarrollo de software, también han sido exportadas a otro tipo de proyectos. Todas las metodologías que se consideran ágiles cumplen con el manifiesto ágil que no es más que una serie de principios que se agrupan en 4 valores:

- Los individuos y su interacción, por encima de los procesos y las herramientas.
- El software que funciona, frente a la documentación exhaustiva.

- La colaboración con el cliente, por encima de la negociación contractual.
- La respuesta al cambio, por encima del seguimiento de un plan.

1.6.2. Métrica de calidad al software

ISO/IEC 9126: El estándar ISO/IEC 9126: 2001 presenta un marco conceptual para el modelo de calidad y define un conjunto de características y sub características, las que debe cumplir todo producto software. (Verity, 2021)

El estándar ISO-9126 establece que cualquier componente de la calidad del software puede ser descrito en términos de una o más de las siete características básicas: funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad, portabilidad y satisfacción; cada una de ellas se detalla a través de un conjunto de sub características que permiten profundizar en la evaluación de la calidad de productos de software. (Verity, 2021)

1.6.3. Medición y Estimación

COSMIC es un método de segunda generación que determina el tamaño del software a partir del número de interacciones entre los componentes de los requerimientos funcionales, el método COSMIC puede aplicarse a diversos tipos de software, incluyendo aplicaciones de negocios, sistemas de información gerencial, software en tiempo real, infraestructura, e inclusive software científico y de ingeniería. (Vasquez, 2015)

1.6.4. Seguridad

La norma ISO 27002 (anteriormente denominada ISO 17799) es un estándar para la seguridad de la información que ha publicado la organización internacional de normalización y la comisión electrotécnica internacional. La versión más reciente de la norma ISO 27002:2013.

La norma ISO 27002 proporciona diferentes recomendaciones de las mejores prácticas en la gestión de la seguridad de la información a todos los interesados y responsables para iniciar, implementar o mantener sistemas de gestión de la seguridad de la información. La seguridad de la información se define en el estándar como “la preservación de la confidencialidad, integridad y disponibilidad. (ISO, 2017)

1.6.5. Pruebas al software

- **Caja negra:** Las Pruebas de Caja Negra, es una técnica de pruebas de software en la cual la funcionalidad se verifica sin tomar en cuenta la estructura interna de código, detalles de implementación o escenarios de ejecución internos en el software. (Terrera, 2017)
- En las pruebas de caja negra el Tester solo se centra en las entradas y salidas de la aplicación, sin preocuparse por el contenido interno. Lo que pase por dentro es indiferente, solo importa que, si se realiza cierta acción, la salida sea la indicada según los requerimientos. (Paulenko, 2019)
- **Caja blanca:** Las pruebas de caja blanca (también conocidas como pruebas de caja de cristal o pruebas estructurales) se centran en los detalles procedimentales del software, por lo que su diseño está fuertemente ligado al código fuente, lo cual, esto significa que tenemos que realizar un “estudio”, o más bien, un análisis de nuestro código. (Kimbo Cesar, 2021)
- **Prueba de estrés (stress):** Consiste en probar los límites que un sistema puede soportar. En este tipo de pruebas se suele enviar más peticiones de las que el software podría atender normalmente para saber el comportamiento de la aplicación. (Daniel, 2018).

1.7. Herramientas

1.7.1. Servidor Web Apache

El servidor HTTP Apache es un servidor web de software libre desarrollado por la Apache Software Foundation (ASF). El producto obtenido de este proyecto es un servidor de código fuente completo, descargable y gratuito.

Apache es robusto y con un ciclo de desarrollo muy rápido gracias a la gran cantidad de colaboradores voluntarios de que dispone. Es también un servidor estable, eficiente, extensible y multiplataforma. Desde el año 1996 es el servidor más utilizado en Internet y es el utilizado en los sistemas GNU/Linux. (Mifsuf Talón, 2012)

1.7.2. Gestor de Base de Datos PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del Mercado.

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. (Zea, et al., 2017)

1.7.3. Herramientas para el desarrollo

En el presente se describen las principales herramientas que ayudara al desarrollo de Sistemas de gestión de la inmobiliaria.

1.7.3.1. PHP

PHP es el lenguaje de lado servidor más extendido en la web. Nacido en 1994, se trata de un lenguaje de creación relativamente reciente, aunque con la rapidez con la que evoluciona Internet parezca que ha existido toda la vida. Es un lenguaje que ha tenido una gran aceptación en la comunidad de desarrolladores, debido a la potencia

y simplicidad que lo caracterizan, así como al soporte generalizado en la mayoría de los servidores de hosting, hasta los más simples y económicos.

La facilidad de PHP se basa en que permite embeber pequeños fragmentos de código dentro de lo que sería una página común creada con HTML. Esos scripts PHP nos permiten realizar determinadas acciones de una forma fácil y eficaz, pudiendo realizar todo tipo de tareas, de las más simples a las más complejas. Esta combinación de PHP dentro del marco de un documento HTML es lo que permite a desarrolladores sin prácticamente nada de experiencia crear comportamientos atractivos de una manera sencilla, una de las claves del éxito del lenguaje.

PHP es lo que se denomina una tecnología del lado del servidor, que ahora se suele englobar dentro del término Backend. (Alvarez, et al., 2018)

1.7.3.2. Laravel

Laravel es un framework de PHP para ayudarnos en un tipo de desarrollo sobre aplicaciones escritas en este lenguaje de programación. Este framework o más bien podría llamarlo compañero de ahora en adelante, nos ayuda en muchas cosas al desarrollar una aplicación, por medio de sus sistema de paquetes y de ser un framework del tipo MVC (Modelo-Vista-Controlador) da como resultado que podamos “despreocuparnos” (por así decirlo) en ciertos aspectos del desarrollo, cómo instanciar clases y métodos para usarlos en muchas partes de nuestra aplicación sin la necesidad de escribirlo y repetirlos muchas veces con lo que eso conlleva a la hora de modificar algo en el código. (Altube, 2021)

1.7.3.3. Livewire

Livewire es un framework de código abierto utilizado para el desarrollo de software full-stack. Fue desarrollado en Laravel, de Taylor Otwell, en conjunto con TailwindCSS. Una de las novedades en Laravel 8 fue la incorporación de Livewire a todos los

proyectos nuevos. En esta guía, te vamos a enseñar a instalar laravel livewire en versiones anteriores a la 8, junto con todas las novedades.

Gracias a Livewire, el desarrollo de aplicaciones web full stack mejoró notablemente. Ahora es posible utilizar blade en conjunto con Vue, todo en un mismo archivo. Olvídate de escribir código separado en el front y en el back. Vas a poder renderizar el javascript desde blade con componentes reactivos. (Perez, 2020)

1.7.3.4. Tailwind CSS

Tailwind CSS es un framework de CSS que ha ganado mucha popularidad en los últimos años, y por una buena razón. Este framework proporciona clases predefinidas para construir interfaces de usuario con HTML. A diferencia de otros frameworks de CSS, que suelen tener estilos personalizados para cada elemento, Tailwind CSS proporciona clases que pueden ser combinadas para definir el aspecto y la funcionalidad de los elementos.

Una de las principales ventajas de Tailwind CSS es que proporciona clases que son altamente reutilizables y personalizables. En lugar de escribir CSS personalizado para cada elemento de una página web, Tailwind CSS proporciona clases para las propiedades más comunes, como colores, tipografía, márgenes, rellenos, tamaños y posiciones. Al utilizar estas clases, los desarrolladores pueden construir interfaces de usuario de manera más rápida y eficiente, sin tener que escribir CSS personalizado. (Morales, 2023)

1.7.3.5. Alpine JS

Alpine.js es un framework minimalista y robusto que resulta ideal para proyectos que no requieren muchas características o complejidad, como landing pages o standalone components. Su tamaño es mínimo, ocupando solo 6.4kb en su versión minificada y comprimida, y su complejidad es baja, lo que lo hace fácil de usar. Está basado en Vue.js y React.js y cuenta con un kit de animaciones. (Martinez, 2020)

1.8. Límites y Alcances

1.8.1. Límites

Dentro de los límites del presente proyecto se toman los siguientes puntos los cuales deben ser tomados en cuenta al momento de la aplicabilidad del sistema web de control y gestión de insumos médicos.

- El proyecto no está orientado en un sistema de contabilidad.
- El sistema no contempla compatibilidad con los sistemas o software existentes en la institución.
- El proyecto no podrá realizar transacciones monetarias.
- Limitaciones de conectividad pueden afectar la operatividad del sistema en entornos remotos sin acceso a internet constante, comprometiendo su funcionamiento en tiempo real.

1.8.2. Alcances

Los alcances del sistema incluirán la implementación de los procesos de registro, control y generación de informes de artículos, y será capaz de realizar las siguientes tareas:

Módulo Principal o Inicio:

- Interfaz principal del sistema.
- Acceso a los demás módulos.
- Resumen de información clave o estadísticas relevantes.

Módulo de Solicitudes de Artículos/Insumos:

- Creación y gestión de solicitudes de insumos por parte del personal.
- Seguimiento del estado de las solicitudes (pendientes, aprobadas, rechazadas).
- Comunicación automática de la aprobación o rechazo de solicitudes.

Módulo de Administración:

- Gestión centralizada de usuarios y roles.
- Configuración y personalización del sistema.
- Acceso a registros para el seguimiento de cambios importantes.

Módulo de Registro de Artículos/Insumos:

- Registro y actualización de la entrada y salida de artículos.
- Seguimiento de inventarios y niveles de existencias.
- Información detallada de cada artículo, incluyendo fechas de caducidad.

Módulo de Recursos Humanos:

- Administración de información del personal (contacto, experiencia, especialidades, certificaciones).
- Asignación y seguimiento de tareas y responsabilidades.

1.9. Aportes

El sistema integral para el control y gestión de insumos médicos y del personal médico/administrativo aportará eficiencia en la asignación de recursos, mejora en la gestión de inventarios, comunicación y colaboración mejoradas, información precisa para la toma de decisiones, ahorro de costos y optimización de procesos, lo que contribuirá a una gestión más eficaz y eficiente en la clínica.

CAPITULO II

2. MARCO TEORICO

2.1. Introducción

En el presente capítulo se darán definiciones sobre metodologías y herramientas en el cual se sustenta el siguiente proyecto de grado.

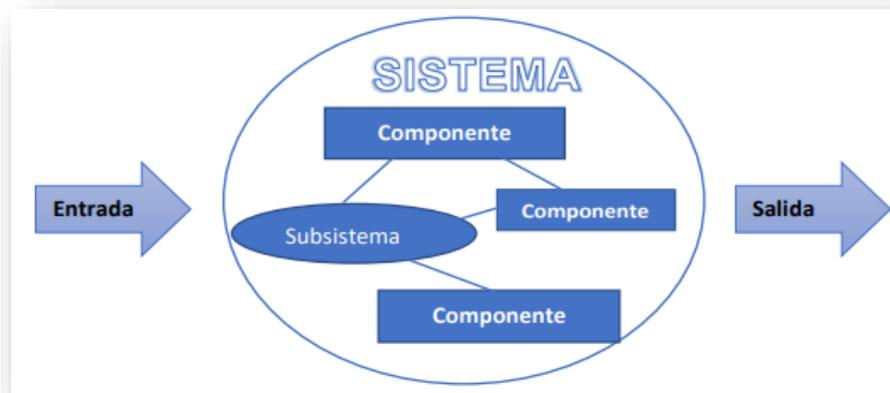
2.2. Sistema

Se entiende por un sistema a un conjunto ordenado de componentes relacionados entre sí, ya se trate de elementos materiales o conceptuales, dotado de una estructura, una composición y un entorno particulares. Se trata de un término que aplica a diversas áreas del saber, como la física, la biología y la informática o computación. (Raffino, 2019)

Un sistema es un conjunto de partes o elementos organizados y relacionados que interactúan entre sí para lograr un objetivo. Los sistemas reciben datos, energía o materia del ambiente (entrada) y proveen información, energía o materia (salida). (Peralta, 2008)

Figura 1

Grafica general del sistema



Nota: Información obtenida de (Alegsa, 2018)

2.3. Sistema Integral

Es un sistema de información que hace uso intensivo y extensivo de las Tecnologías de la Información y la Comunicación (TIC) para integrar o centralizar la gestión de la información dentro de una organización. (Sistema Integral de Información de extensión, 2015)

2.4. Administración

La administración como el proceso de plantear, organizar, liderar y controlar el trabajo de los miembros de la organización y de utilizar todos los recursos disponibles de la empresa para alcanzar los objetivos organizacionales establecidos. (Stoner, 1995)

La administración es el conjunto de las funciones o proceso de planificar, organizar, liderar, dirigir y controlar el trabajo de los miembros de la organización de manera eficaz y utilizar todos los recursos para lograr los objetivos organizacionales.

2.5. Control

Control es la función gerencial que implica monitorear actividades para garantizar que se estén realizando según lo planeado y corregir las desviaciones importantes. Los gerentes no pueden saber si sus unidades están funcionando realmente como es debido sino hasta que evalúan qué actividades se han realizado y comparan el desempeño real con el estándar deseado. Un sistema de control eficaz asegura que las actividades se realicen de tal forma que se logren las metas de la organización. La eficacia de un sistema de control depende de qué tanto éste facilite el logro de las metas. Cuanto más ayude el sistema de control a los gerentes a lograr sus metas organizacionales, mejor será ese sistema (Robbins, Decenzo, et al., 2013)

Es una función de administración que se elabora para asegurar que los hechos concuerden con los planes. Para que sea eficaz, se debe enfocar al presente, se debe centrar en la corrección y no en el error, debe, asimismo, ser específico, de tal forma

que se concentre en los factores claves que afecten los resultados.es universal y abarca todas las fases de la empresa. (Scalan, 1987)

2.6. Control Administrativo

El control administrativo se considera como un sistema de realimentación similaral que opera en el termostato de los hogares.

Este sistema coloca la función de control en una perspectiva más compleja y realista que si se la considera sólo como cuestión de establecer estándares, medir el desempeño y corregir las desviaciones.

Los gerentes miden el desempeño real, comparan esta medición contra los estándares e identifican y analizan las desviaciones, y entonces, para hacer las correcciones necesarias, deben desarrollar un programa de acción correctiva e instrumentarlo para llegar al desempeño deseado. (Harold Koontz, 2012)

2.7. Inventario

El inventario es el conjunto de artículos o mercancías que se acumulan en el almacén pendientes de ser utilizados en el proceso productivo o comercializados. Otra definición de inventario vinculada al ámbito económico es la relación ordenada de bienes de una organización.

El concepto inventario o stock resulta muy importante en las empresas con el propósito de que las demandas de los consumidores sean atendidas para que no se vea interrumpido el proceso productivo ante la falta de materias primas. Pueden considerarse como una herramienta reguladora que mantiene el equilibrio entre los flujos reales de entrada y los de salida. (Garcia, 2017).

2.8. Stock

Se considera stock aquella cantidad de un producto que se encuentra acumulada en un lugar determinado, fija o bien en movimiento hacia sus centros de distribución. Su función es la de servir de instrumento de regulación de toda la cadena logística, con el fin de conseguir un flujo de materiales continuo. (Yubero, 2001)

Los stocks también permiten:

- Absorber las diferencias entre las previsiones de demanda hechas y los movimientos reales que se producen.
- Evitar rupturas del flujo de materiales por circunstancias diversas, como por ejemplo los desajustes en los sistemas de transporte de reposición, de transportes a clientes, demandas imprevistas, incumplimiento por parte de proveedores.
- Especialización en la producción.
- Utilizar economías de escala.

2.9. Stock mínimo

Este stock mínimo será el que permita que la tienda siga proveyendo de servicio a los consumidores, sin que estos noten carencias de servicio o sin que se rompa la cadena del mismo. En el cálculo del stock mínimo se deben tener en cuenta factores tales como el tiempo de entrega de nuevos pedidos, de forma que el volumen de unidades se mantenga siempre dentro de unos límites, por lo recomendable es asegurarse de hacer los pedidos antes de que el alcance el stock mínimo, así, incluso ante un imprevisto la empresa puede seguir manteniendo la calidad de sus servicios. (Rivero, 2019)

2.10. Stock disponible

Este stock es la suma del stock neto y los pedidos enviados de tus proveedores que no has recibido todavía.

Es el stock físico más los pedidos en curso del artículo a los proveedores, menos la demanda insatisfecha. (Sandoval Caicedo, 2012)

2.11. Sistema de Inventarios/Insumos

Un sistema de inventarios es un conjunto de normas, métodos y procedimientos aplicados de manera sistemática para planificar y controlar los materiales y productos

que se emplean en una organización. Este sistema puede ser manual o automatizado. (Mena, 2011)

Sistema de inventario una de ellas es la capacidad de predicción; con el registro de un inventario se puede establecer una cuenta ordenada de todos los productos esto permite tener un equilibrio en entradas y salidas y sobre todo permite predecir lo que se necesita en el momento exacto. (Muller, 2004)

Sistema de inventario conjunto de normas, métodos y procedimientos de registro con la cual se puede establecer una cuenta ordenada de todos los productos, la cual permite tener equilibrio y control en las entradas y salidas de los productos en el momento exacto.

2.12. Sistema de Información

Se puede plantear la definición técnica de un sistema de información como un conjunto de componentes interrelacionados que recolectan (o recuperan), procesan, almacenan y distribuyen información para apoyar los procesos de toma de decisiones y de control en una organización. Además de apoyar la toma de decisiones, la coordinación y el control, los sistemas de información también pueden ayudar a los gerentes y trabajadores del conocimiento a analizar problemas, visualizar temas complejos y crear nuevos productos (Laudon, et al., 2016)

2.13. Almacén

“Un almacén se puede considerar como un centro de producción en el que se efectúa una serie de procesos relacionados con:

Recepción, control, adecuación y colocación de artículos recibidos (procesos de entrada).

Almacenamiento de artículos en condiciones eficaces para su conservación, identificación, selección y control (proceso de almacenaje).

Recogida de artículos y preparación de la expedición de acuerdo con los requerimientos de los clientes (procesos de salida)". (Tejero, 2006)

2.14. Modelo de ciclo de vida

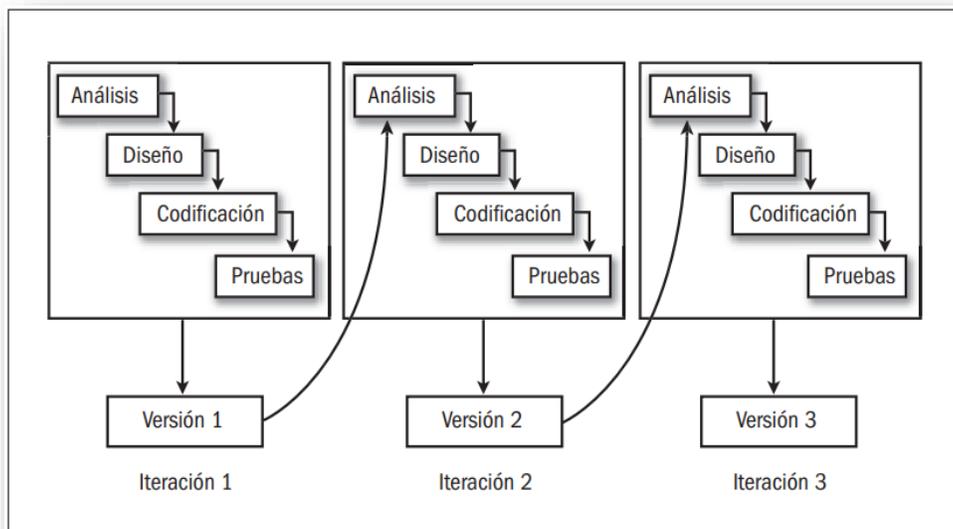
2.14.1. Ciclo de vida iterativo

También derivado del ciclo de vida de cascada, este modelo busca reducir el riesgo que surge entre las necesidades del usuario y el producto final por malos entendidos durante la etapa de solicitud de requerimientos.

Es la iteración de varios ciclos de vida en cascada. Al final de cada iteración se le entrega al cliente una versión mejorada o con mayores funcionalidades del producto. El cliente es quién luego de cada iteración, evalúa el producto y lo corrige o propone mejoras. Estas iteraciones se repetirán hasta obtener un producto que satisfaga al cliente.

Figura 2

Modelo de Ciclo de Vida Iterativo



Nota: Información obtenida de (Cantone, 2006, p. 27)

Se suele utilizar en proyectos en los que los requerimientos no están claros de parte del usuario, por lo que se hace necesaria la creación de distintos prototipos para presentarlos y conseguir la conformidad del cliente.

Podemos adoptar el modelo mencionado en aplicaciones medianas a grandes, en las que el usuario o cliente final no necesita todas las funcionalidades desde el principio del proyecto. es un candidato ideal para este tipo de modelo de ciclo de vida. (Cantone, 2006)

2.15. Metodología de desarrollo de software

2.15.1. UWE - Ingeniería web basada en UML

UWE es un enfoque de ingeniería de software para el dominio web cuyo objetivo es cubrir todo el ciclo de vida del desarrollo de aplicaciones web. El aspecto clave que distingue a UWE es la dependencia de los estándares.

El enfoque principal del enfoque UWE es proporcionar un Lenguaje de modelado específico de dominio basado en UML que también incluye características de seguridad, metodología basada en modelos, soporte de herramientas para el diseño sistemático, y soporte de herramientas para la generación (semi) automática de aplicaciones web.

UWE utiliza notación UML pura y tipos de diagramas UML siempre que sea posible para el análisis y diseño de aplicaciones web, es decir, sin extensiones de ningún tipo. Para las características específicas de la Web, como los nodos y enlaces de la estructura de hipertexto, el perfil UWE incluye estereotipos, valores etiquetados y restricciones definidas para los elementos de modelado. La extensión UWE cubre navegación, presentación, procesos de negocio y aspectos de adaptación. La notación UWE se define como una extensión ligera del UML.

Además, UWE proporciona soporte de herramientas para el diseño de modelos, comprobaciones de coherencia de modelos y generación semiautomática de sistemas

web. ArgoUWE y MagicUWE son complementos que admiten la notación del perfil UWE y las transformaciones para ayudar al trabajo del diseñador. Para la generación semiautomática de aplicaciones web se implementaron diferentes enfoques. (UWE – UML-based Web Engineering, 2016)

Figura 3

Lenguaje de Modelado Unificado



Nota: Información obtenida de (LMU, 2016)

2.15.1.1. Características de UWE

La característica de UWE es el hecho de ser un enfoque basado en estándares que no se limita al uso del UML, sino que también usa XMI como formato de intercambio de modelos, MOF para metamodelado, los principios basados en modelos del enfoque MDA, el lenguaje de transformación del modelo QVT y XML.

2.15.1.2. Fases de la uwe

a) Captura, análisis y especificación de requisitos

En simple palabras y básicamente, durante esta fase, se adquieren, reúnen y especifican las características funcionales y no funcionales que deberá cumplir la aplicación web.

Trata de diferente forma las necesidades de información, las necesidades de navegación, las necesidades de adaptación y las de interfaz de usuario, así como algunos requisitos adicionales. Centra el trabajo en el estudio de los casos de uso.

- **Requerimientos**

Requerimientos Funcionales

Los requerimientos funcionales para un sistema refieren lo que el sistema debe hacer.

Tales requerimientos dependen del tipo de software que se esté desarrollando, de los usuarios esperados del software y del enfoque general que adopta la organización cuando se escriben los requerimientos. Al expresarse como requerimientos del usuario, los requerimientos funcionales se describen por lo general de forma abstracta que entiendan los usuarios del sistema. Sin embargo, requerimientos funcionales más específicos del sistema detallan las funciones del sistema, sus entradas y salidas, sus excepciones, etcétera.

Los requerimientos funcionales del sistema varían desde requerimientos generales que cubren lo que tiene que hacer el sistema, hasta requerimientos muy específicos que reflejan maneras locales de trabajar o los sistemas existentes de una organización.

Requerimientos no Funcionales

Los requerimientos no funcionales, como indica su nombre, son requerimientos que no se relacionan directamente con los servicios específicos que el sistema entrega a sus usuarios. Pueden relacionarse con propiedades emergentes del sistema, como fiabilidad, tiempo de respuesta y uso de almacenamiento. De forma alternativa, pueden definir restricciones sobre la implementación del sistema, como las capacidades de los dispositivos I/O las representaciones de datos usados en las interfaces con otros sistemas. (Sommerville, 2011)

- **Modelo de requisitos**

El modelo de Requisitos (a veces también llamado modelo de caso de uso) define la funcionalidad para una aplicación y actores que la usan en un alto nivel de abstracción. Esto se puede hacer utilizando diagramas de casos de uso UML simples.

Además, UWE enriquece el diagrama de casos de uso de UML con estereotipos, por ejemplo: el estereotipo «webUseCase» que se puede aplicar en paquetes UML y casos de uso. Su etiqueta permite definir expresiones de alto nivel que deben ser verdaderas para poder realizar acciones contenidas. El estereotipo «browsing» se puede aplicar en casos de uso de UML para expresar que un caso de uso solo contiene actividades de exploración, es decir, se ejecutan consultas que no cambian los datos de la aplicación. El estereotipo « processing » expresa que, la mayoría de las veces, los datos de la aplicación se crean o cambian mientras se procesa el caso de uso.

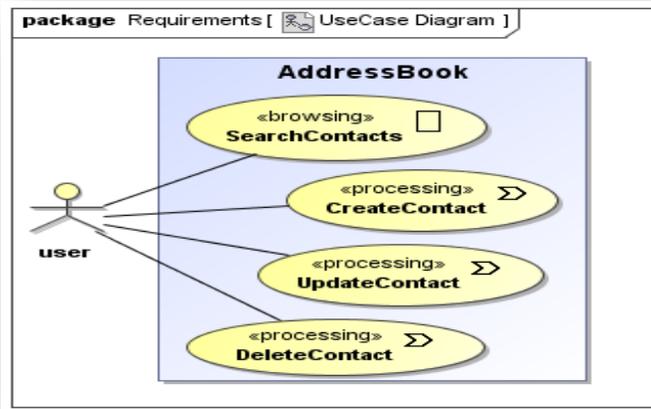
Tabla 1

Estereotipo de modelo de requisitos

Nombres de estereotipos y sus íconos	
 browsing	 processing
 webUseCase	

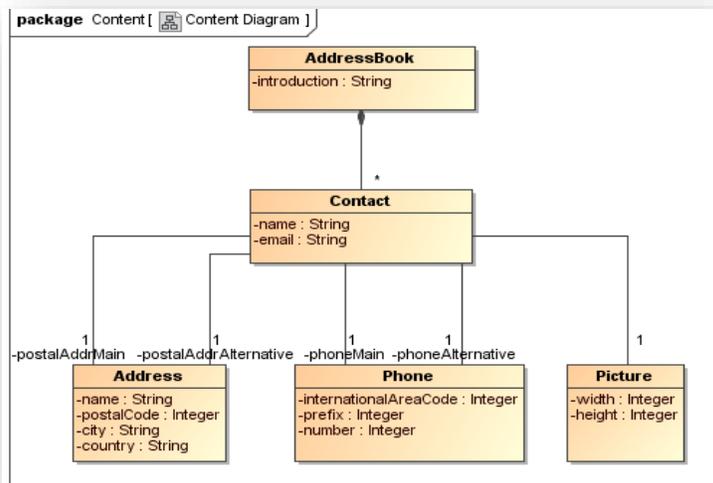
Nota: Información obtenida de (LMU – Ludwig-Maximilians-Universität München, 2016)

Figura 4
Diagrama de casos de uso



Nota: Información obtenida de (LMU – Ludwig-Maximilians-Universität München, 2016)

Figura 5
Modelo de contenido



Nota: Información obtenida de (LMU – Ludwig-Maximilians-Universität München, 2016)

b) Diseño del sistema

Se basa en la especificación de requisitos producido por el análisis de los requerimientos (fase de análisis), el diseño define cómo estos requisitos se cumplirán, la estructura que debe darse a la aplicación web.

- **Modelo de Contenido (Conceptual)**

El modelo de contenido contiene la estructura de datos utilizada por la aplicación. La “estructura de datos” generalmente se refiere a la estructura de clases importantes en una aplicación orientada a objetos o a la estructura lógica de datos en una base de datos y, en muchas ocasiones, ambas están estrechamente relacionadas. Para lenguajes no orientados a objetos, las clases de contenido pueden representar módulos principales o archivos de código y sus relaciones entre sí, en caso de que esto resulte útil para planificar la estructura de la aplicación o para fines de documentación.

- **Modelo de navegación**

En un sistema para la web es útil saber cómo están enlazadas las páginas. Ello significa que necesitamos un diagrama conteniendo nodos y enlaces (links).

¿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.

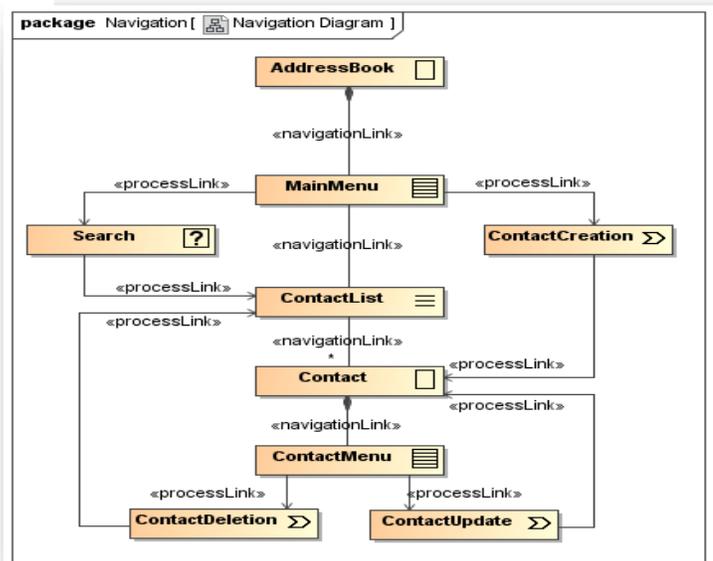
UWE provee diferentes estereotipos, los que presentaremos mediante nuestro ejemplo. La forma más simple de obtener un Diagrama de Navegación básico es utilizando la Transformación de contenido a navegación.

Tabla 2
 Estereotipo de modelo de navegación

Nombres de estereotipos y sus iconos	
 clase de navegación	 menú
 índice	 pregunta
 visita guiada	 clase de proceso
 nodo externo	

Nota: Información obtenida de (LMU – Ludwig-Maximilians-Universität München, 2016)

Figura 6
 Modelo de Navegación



Nota: Información obtenida de (LMU – Ludwig-Maximilians-Universität München, 2016)

- **Modelo de presentación**

El modelo de presentación de UWE esboza bloques de construcción lógicos de la interfaz de usuario de una aplicación web. Los diagramas de estructura compuesta se pueden usar para expresar la composición como clases y propiedades UML anidadas.

Tabla 3

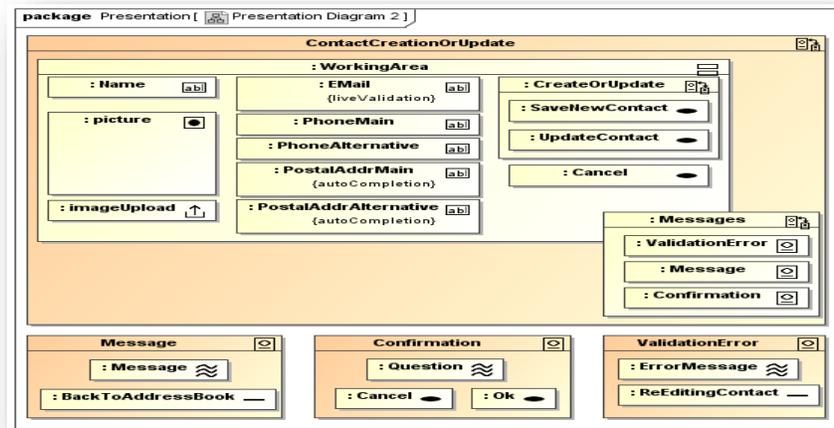
Estereotipo de modelo de presentación

Nombres de estereotipos y sus íconos	
 grupo de presentación	 página de presentación
 texto	 entrada de texto
 ancla	 Subir archivo
 botón	 imagen
 formulario	 componente del cliente
 alternativas de presentación	 selección

Nota: Información obtenida de (LMU – Ludwig-Maximilians-Universität München, 2016)

Figura 7

Modelo de Presentación



Nota: Información obtenida de (LMU – Ludwig-Maximilians-Universität München, 2016)

c) Codificación del software

Durante esta etapa se realizan las tareas que comúnmente se conocen como programación; que consiste, esencialmente, en llevar a código fuente, en el lenguaje de programación elegido, todo lo diseñado en la fase anterior.

d) Pruebas

Las pruebas se utilizan para asegurar el correcto funcionamiento de secciones de código.

e) La Instalación o Fase de Implementación:

Proceso por el cual los programas desarrollados son transferidos apropiadamente al computador destino, inicializados, y, eventualmente, configurados; todo ello con el propósito de ser ya utilizados por el usuario final.

Esto incluye la implementación de la arquitectura, de la estructura del hiperespacio, del modelo de usuario, de la interfaz de usuario, de los mecanismos adaptativos y las tareas referentes a la integración de todas estas implementaciones.

f) El Mantenimiento

es el proceso de control, mejora y optimización del software ya desarrollado e instalado, que también incluye depuración de errores y defectos que puedan haberse filtrado de la fase de pruebas de control. (Medina, 2016)

2.16. Seguridad de aplicaciones web

La seguridad de las aplicaciones Web es un tema crítico y complejo para los desarrolladores Web. Ya que requiere realizar estudios y tener entendimiento sobre los puntos de vulnerabilidad en cada aplicación Web.

Se puede clasificar los puntos más importantes:

Seguridad en el servidor

- Servidor Web
- Servidor de Base de Datos
- Lenguaje de programación

Seguridad en la Aplicación

- Control de Acceso
- Validación de datos de entrada
- Pruebas de código

Seguridad en la Comunicación

- Protocolos de seguridad
- Certificados de Comunicación

Copias de Seguridad

- Backup de la Base de Datos
- Backup del control de Acceso

Las pruebas de seguridad están diseñadas para probar la vulnerabilidad en el ambiente del lado del cliente, las comunicaciones de red que ocurren mientras los datos pasan del cliente al servidor y de vuelta y el ambiente del lado del servidor. Cada uno de estos dominios puede recibir ataques y es labor de quien prueba la seguridad descubrir las debilidades que pueden explotar quienes tengan la intención de hacerlo (Pressman R. , 2005)

2.17. Modelo vista controlador (MVC)

El modelo MVC es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el Sistema

de Gestión de Base de Datos y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista.

2.17.1 Modelo

Esta es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos; por ejemplo, no permitiendo comparar un número de unidades negativo, calculando si hoy es el cumpleaños del usuario o los totales, impuestos o importes en un carrito de la compra.

2.17.2 Vista

Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.

2.17.3 Controlador

Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

2.17.4 Características del Modelo Vista Controlador

Aunque se pueden encontrar diferentes implementaciones de MVC, la característica del flujo que sigue el control generalmente es el siguiente:

- El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace, etc.)
- El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.
- El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el. Los controladores complejos

están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.

- El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo. El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, el patrón de observador puede ser utilizado para proveer cierta dirección entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio.
- La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

2.18. Calidad del software

- **Calidad**

Grado en el que un conjunto de características inherentes de un objeto cumple con los requisitos. El término calidad puede utilizarse acompañado de adjetivos tales como pobre, buena o excelente. “Inherente”, en contraposición a “asignado”, significa que existe en el objeto. (iso, 2016)

- **Calidad de Software**

capacidad del producto de software para satisfacer necesidades establecidas e implícitas cuando se usa bajo condiciones específicas. Esta definición difiere de la definición de calidad ISO 9000: 2005 principalmente porque la definición de calidad del software se refiere a la satisfacción de las necesidades declaradas e implícitas, mientras que la definición de calidad ISO 9000 se refiere a la satisfacción de los requisitos. (iso, 2014)

2.18.1. Factores de calidad ISO/IEC 9126

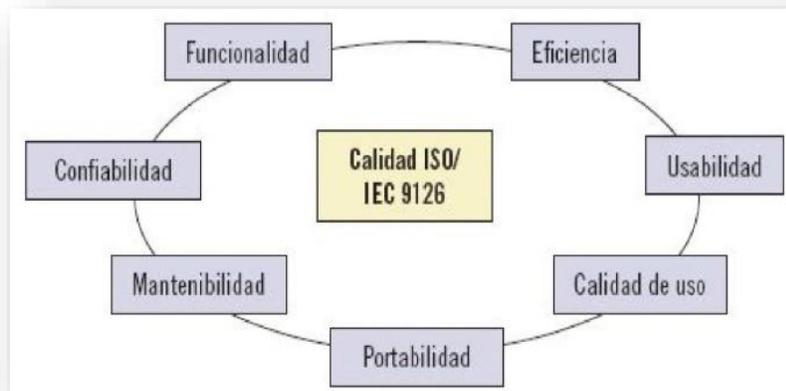
La ISO 9126 / IEC es una norma creada por la Organización Internacional de Normalización en 1992 utilizada para valorar la calidad de un software, Evaluación de

productos de software de tecnología de la información Características de calidad y pautas para su uso, ofrece la posibilidad de concretar y examinar la calidad del software desde diversas características relacionadas con operaciones, requisitos, progreso, utilización, valoración, apoyo técnico, conservación y auditoría de software.

Esta normativa se define por medio de seis principios: funcionalidad, mantenibilidad, eficiencia, confiabilidad, usabilidad, portabilidad y una séptima que no es principal, pero si valorable, que es la calidad de uso.

Figura 8

Características de la iso 9126



Nota: Información obtenida de (Durán Portillo, 2015)

2.18.1.1. Funcionalidad

Se entiende por funcionalidad de software cuando cumple eficazmente con todas las tareas para las que fue ideado o programado, ya que este a su vez fue creado para satisfacer las necesidades de un cliente concreto.

Figura 9
Características de la funcionabilidad



Nota: Información obtenida de (Durán Portillo, 2015)

Dentro de este apartado se distinguen cinco apartados para evaluar que tal producto cumpla la ISO 9126:

- **Seguridad:** Uno de los criterios más importantes en cualquier software, consiste en que el programa es capaz de proteger tanto la información del programa como la que los usuarios depositan o comparten en el programa para su mejor funcionamiento o para identificarse en unos perfiles determinados del mismo.
La seguridad es esencial, ya que protege la información de los usuarios de otros usuarios o software no autorizados que quieren acceder a ella.
- **Interoperabilidad:** para es una de las características fundamentales para que sea atractivo al usuario, ya que la interoperabilidad es la capacidad de un producto para que el usuario pueda interactuar con él. Así como también de interactuar entre varios softwares para que tenga como resultado el buen funcionamiento del producto.

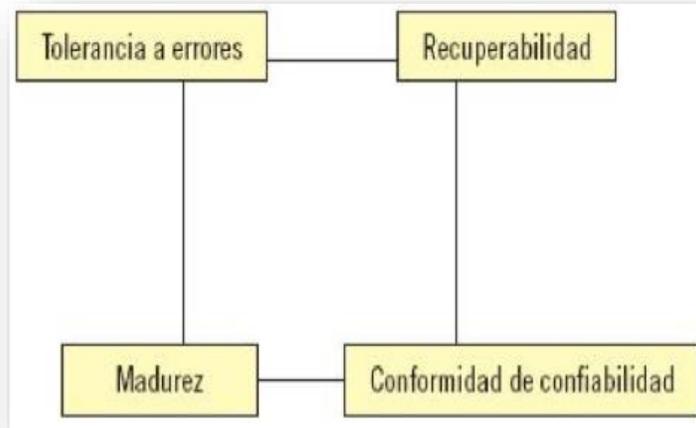
- **Conformidad de la funcionalidad:** esta característica hace mención a la buena competencia de cumplir los estándares necesarios para cumplir la funcionalidad del software.
- **Exactitud:** se da en un producto cuando este funciona perfectamente, realizando las funciones para las que fue diseñado en un tiempo concreto.
- **Adecuación:** una de las características fundamentales de cualquier software o producto es que este cumple las funciones para las que fue creado y los usuarios se vean satisfechos de las necesidades por las cuales adquirieron o usan dicho producto.

2.18.1.2. Confiabilidad

La confiabilidad de un producto es la idoneidad del funcionamiento correcto del mismo cuando el usuario está haciendo un correcto uso del software. Dicha confiabilidad se especifica en cuatro criterios a seguir:

- **Madurez:** cuando el usuario hace un uso incorrecto del software el programa intenta evitar los fallos y los comunica al usuario que está realizando operaciones incorrectas y que de ahí deriva el mal funcionamiento del software. También el mal funcionamiento puede deberse a un problema de hardware.
- **Tolerancia a errores:** esta característica versa sobre la suficiencia del producto a seguir, funcionando, aunque ocurran errores durante su utilización.
- **Recuperabilidad:** Teniendo en cuenta que ningún producto o software está a salvo de los errores que puedan ocurrir durante el uso del mismo, este criterio analiza la capacidad que tiene el producto de recuperar la información ante un fallo.
- **Conformidad de la fiabilidad:** es el criterio que recoge la fiabilidad del funcionamiento del producto según los estándares o normativas recogidas en la ISO para su buen funcionamiento.

Figura 10
Características de la Confiabilidad

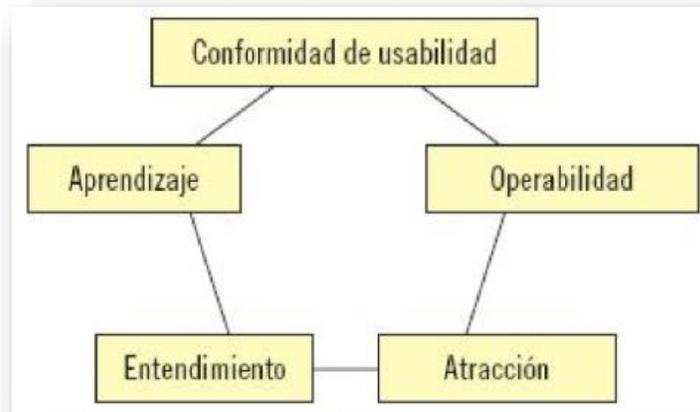


Nota: Información obtenida de (Durán Portillo, 2015)

2.18.1.3. Usabilidad

Se denomina usabilidad a la capacidad que tiene un producto de ser comprendido, estudiado y utilizado de una manera sencilla y sugestiva. Quienes mejor pueden valorar la usabilidad de un producto son sus usuarios finales, ya que nadie como ellos, pueden valorar lo que reciben del programa y si satisface sus expectativas.

Figura 11
Características de la Usabilidad



Nota: Información obtenida de (Durán Portillo, 2015)

Para valorar el resultado obtenido tras meses de trabajo el producto es examinado bajo cinco criterios fundamentales:

- **Entendimiento:** cuando se habla de entendimiento se hace referencia a que el programa debe ser fácil de utilizar para que el usuario le sea menos tediosa su utilización. Para ello, no solo tiene que tener una buena interface, sino también disponer de todos los archivos y documentación necesaria para saber cómo utilizarlo correctamente.
- **Aprendizaje:** es la capacidad que tiene el software para enseñar al usuario cuál es la manera correcta de utilizarlo, para lo cual se puede tratar de realizar un software intuitivo, dejar una ayuda virtual o manuales impresos para su uso.
- **Operabilidad:** se la conoce así a la capacidad que da el producto al usuario para que este pueda utilizarlo.
- **Atracción:** se podría calificar como la chispa del producto, es la primera característica que el usuario inconscientemente percibirá de él, ya que esta habla del interface y representación del producto, su tipografía, imágenes, etc.

- **Conformidad de su uso:** es la que resume las otras cuatro y los estándares y normativas de la ISO. (Durán, 2015)

2.18.1.4. Eficiencia

La capacidad del producto software para proveer un desempeño apropiado, de acuerdo a la cantidad de recursos utilizados y bajo las condiciones planteadas.

- **Comportamiento en el tiempo:** la capacidad del producto software para proveer tiempos apropiados de respuesta y procesamiento, y ratios de rendimiento cuando realiza su función bajo las condiciones establecidas.
- **Utilización de recursos:** la capacidad del producto software para utilizar apropiadas cantidades y tipos de recursos cuando éste funciona bajo las condiciones establecidas.
- **Conformidad de eficiencia:** la capacidad del producto software para adherirse a normas o convenciones relacionadas a la eficiencia.

Figura 12

Característica de Eficiencia



Nota: Información obtenida de (Franco Díaz, Anaya Castro, y Valenzuela Venicio, 2019)

2.18.1.5. *Mantenibilidad*

Capacidad del producto software para ser modificado. Las modificaciones pueden incluir correcciones, mejoras o adaptación del software a cambios en el entorno, y en requerimientos y especificaciones funcionales.

- **Analizabilidad:** la capacidad del producto software para ser diagnosticado por deficiencias o causas de fallas en el software o la identificación de las partes a ser modificadas.
- **Cambiabilidad:** la capacidad del software para permitir que una determinada modificación sea implementada.
- **Estabilidad:** la capacidad del producto software para evitar efectos inesperados debido a modificaciones del software.
- **Testeabilidad:** la capacidad del software para permitir que las modificaciones puedan ser validadas.
- **Conformidad de facilidad de mantenimiento:** la capacidad del software para adherirse a estándares o convenciones relativas a la facilidad de mantenimiento.

Figura 13
Característica de Mantenibilidad



Nota: Información obtenida de (Franco Díaz, Anaya Castro, y Valenzuela Venicio, 2019)

2.18.1.6. Portabilidad

La capacidad del software para ser trasladado de un entorno a otro.

- **Adaptabilidad:** la capacidad del producto software para ser adaptado a diferentes entornos sin aplicar acciones o medios diferentes de los previstos para el propósito del software considerado.
- **Instalabilidad:** la capacidad del producto software para ser instalado en un entorno definido.
- **Co existencia:** la capacidad del producto software para co-existir con otro producto software independiente dentro de un mismo entorno compartiendo recursos comunes.
- **Reemplazabilidad:** la capacidad del producto software para ser utilizado en lugar de otro producto software, para el mismo propósito y en el mismo entorno.
- **Conformidad de portabilidad:** la capacidad del software para adherirse a estándares o convenciones relacionados a la portabilidad.

Figura 14
Característica de Portabilidad



Nota: Información obtenida de (Franco Díaz, Anaya Castro, y Valenzuela Venicio, 2019)

2.18.1.7. Calidad de Uso

La norma ISO/IEC 9126-1 define la calidad en uso como: la perspectiva del usuario de la calidad del producto software cuando éste es usado en un ambiente específico y un contexto de uso específico. Ésta mide la extensión para la cual los usuarios pueden conseguir sus metas en un ambiente particular, en vez de medir las propiedades del software en sí mismo

Se hace la aclaración que un usuario es cualquier tipo de posible usuario y cuyos requerimientos pueden ser diferentes; por ejemplo, un operador del software tiene un requerimiento diferente que un responsable del mantenimiento del software.

- **Efectividad:** la capacidad del producto software para permitir a los usuarios lograr las metas especificadas con precisión y completitud en un contexto de uso específico.
- **Productividad:** la capacidad del producto software para permitir a los usuarios emplear cantidades apropiadas de recursos en relación a la efectividad lograda en un contexto de uso específico.
- **Integridad:** la capacidad del producto software para lograr niveles aceptables de riesgo de daño a las personas, negocio, software, propiedad o entorno en un contexto de uso específico.
- **Satisfacción:** la capacidad del producto software para satisfacer a los usuarios en un contexto de uso específico. (Díaz, et al., 2019)

Figura 15

Características de la Calidad en Uso



Nota: Información obtenida de (Durán Portillo, 2015)

2.19. Seguridad de la Información

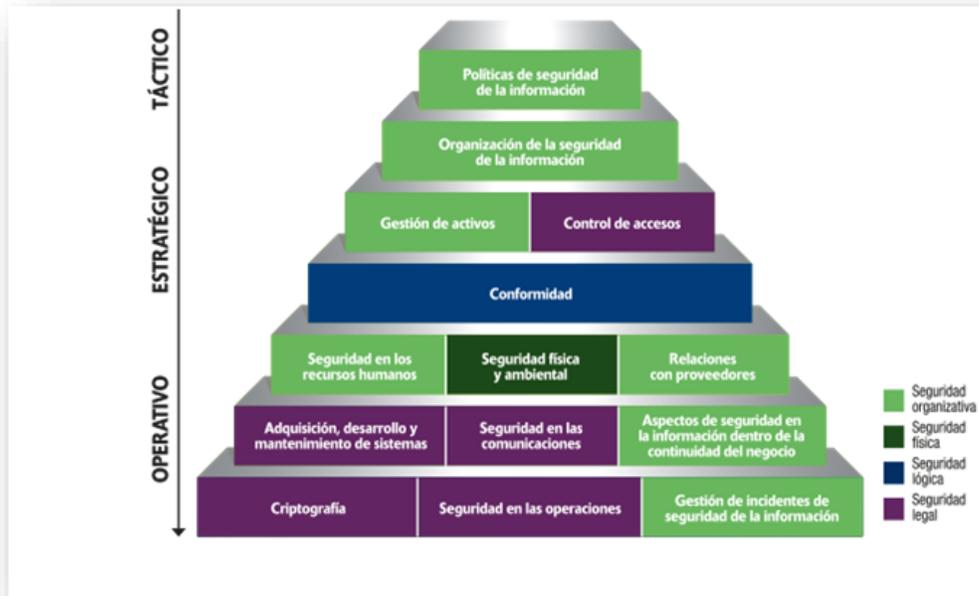
2.19.1. ISO/IEC 27002

La Norma UNE-ISO/IEC 27002 establece las directrices y principios generales para el comienzo, la implementación, el mantenimiento y la mejora de la gestión de la seguridad de la información en una organización. Es un catálogo de buenas prácticas, obtenido a partir de la experiencia y colaboración de numerosos participantes, los cuales han alcanzado un consenso acerca de los objetivos comúnmente aceptados para la gestión de la seguridad de la información.

Los objetivos de control y los controles de esta norma internacional tienen como fin servir de guía para el desarrollo de pautas de seguridad internas y prácticas efectivas de gestión de la seguridad. Por ello, la elección de los controles permanece sujeta a lo detectado en un análisis de riesgos previo, y el grado de implementación de cada uno de los controles se llevará a cabo de acuerdo a los requisitos de seguridad identificados

y a los recursos disponibles de la organización para alcanzar así un balance razonable entre seguridad y coste. (Gómez, et al.,2012)

Figura 16
Estructura de dominios de ISO/IEC 27002



Nota: Información obtenida de (AENOR INTERNACIONAL, S.A.U., 2016)

2.20. Pruebas de software

El objetivo de las pruebas no es asegurar la ausencia de defectos en un software, únicamente puede demostrar que existen defectos en el software.

2.20.1. Pruebas de Caja Blanca

A este tipo de técnicas se le conoce también como Técnicas de Caja Transparente o de Cristal. Este método se centra en cómo diseñar los casos de prueba atendiendo al comportamiento interno y la estructura del programa. Se examina así la lógica interna del programa sin considerar los aspectos de rendimiento.

El objetivo de la técnica es diseñar casos de prueba para que se ejecuten, al menos una vez, todas las sentencias del programa, y todas las condiciones tanto en su vertiente verdadera como falsa.

Como se ha indicado ya, puede ser impracticable realizar una prueba exhaustiva de todos los caminos de un programa. Por ello se han definido distintos criterios de cobertura lógica, que permiten decidir qué sentencias o caminos se deben examinar con los casos de prueba. Estos criterios son:

- Cobertura de Sentencias
- Cobertura de Decisión
- Cobertura de Condiciones
- Cobertura Decisión/Condición
- Cobertura de Condición Múltiple
- Cobertura de Caminos

2.20.2. Pruebas de Caja Negra

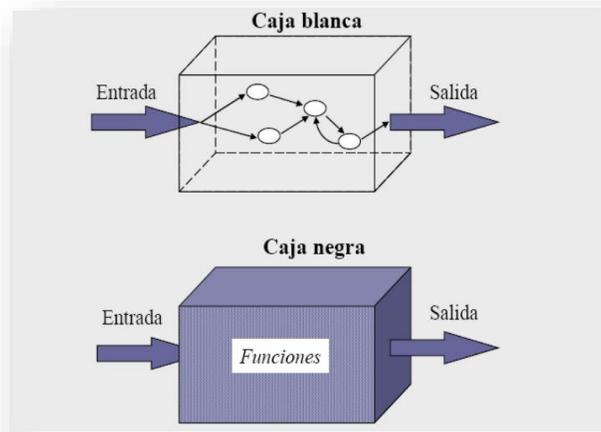
También conocidas como Pruebas de Comportamiento, estas pruebas se basan en la especificación del programa o componente a ser probado para elaborar los casos de prueba. El componente se ve como una “Caja Negra” cuyo comportamiento sólo puede ser determinado estudiando sus entradas y las salidas obtenidas a partir de ellas. No obstante, como el estudio de todas las posibles entradas y salidas de un programa sería impracticable se selecciona un conjunto de ellas sobre las que se realizan las pruebas. Para seleccionar el conjunto de entradas y salidas sobre las que trabajar, hay que tener en cuenta que en todo programa existe un conjunto de entradas que causan un comportamiento erróneo en nuestro sistema, y como consecuencia producen una serie de salidas que revelan la presencia de defectos. Entonces, dado que la prueba exhaustiva es imposible, el objetivo final es pues, encontrar una serie de datos de entrada cuya probabilidad de pertenecer al conjunto de entradas que causan dicho comportamiento erróneo sea lo más alto posible.

Al igual que ocurría con las técnicas de Caja Blanca, para confeccionar los casos de prueba de Caja Negra existen distintos criterios. Algunos de ellos son:

- Particiones de Equivalencia.
- Análisis de Valores Límite.
- Métodos Basados en Grafos.
- Pruebas de Comparación.
- Análisis Causa-Efecto. (Departamento LSI, 2004).

Figura 17

Representación de pruebas de Caja Blanca y Caja Negra



Nota: Información obtenida de (Departamento LSI, 2004)

2.20.3. Pruebas de estrés

Una Prueba de Estrés es el proceso en el cual se eligen las actividades a probar en un sitio para codificarlas y ejecutarlas en un tiempo determinado desde una ubicación remota. Nos permiten identificar y planear ante la posibilidad de fallas en el funcionamiento de la plataforma, preferiblemente de manera preventiva. (Cinquegrani, 2020)

Lo que se hace es medir la capacidad de la infraestructura, los tiempos de respuesta a actividades específicas correspondientes a un número incremental de usuarios generados de manera remota y buscar el número límite de usuarios antes de la negación de servicio de la plataforma. En otras palabras, cuantos usuarios interactuando de manera simultánea con la plataforma puede tolerar la infraestructura de tu sitio. (Cinquegrani, 2020)

2.21. Medición y Estimación de Software

En la mayoría de las empresas donde se produce software para apoyar el negocio, las prácticas de estimación y planificación son débiles. En general, los administradores estiman el costo y la duración del proyecto a desarrollar utilizando solamente el juicio de un experto, lo que produce cronogramas y presupuestos poco acertados.

Es importante medir el proceso de ingeniería de software y el producto que se elabora porque es la forma más objetiva de comprender y mejorar el proceso de desarrollo y el producto que se elabora. Si no se realizan mediciones, las decisiones se basan solo en evaluaciones subjetivas, lo que puede llevar a malas estimaciones o interpretaciones erróneas del proceso. Para establecer objetivos de mejora es necesario conocer el estado actual de desarrollo del software. (Salazar, 2009)

La estimación se utiliza para definir el presupuesto del proyecto, y el producto se ajusta para que se cumpla la cifra del presupuesto. Un proyecto que está dentro de presupuesto puede lograr esto a expensas de las características en el software a desarrollar. Las organizaciones necesitan hacer evaluaciones de esfuerzo y costo del software. (Sommerville, 2011)

2.21.1. Estimación de software con COSMIC

El método de Medición de COSMIC es la segunda generación de métodos de medición de tamaño funcional. Este ofrece un nivel de confiabilidad compatible con todos los tipos de software. Es de dominio público y el acceso a su documentación no tiene

costo. El método tiene reconocimiento total de la ISO/IEC. Posee una base conceptual compatible con la ingeniería de software moderna.

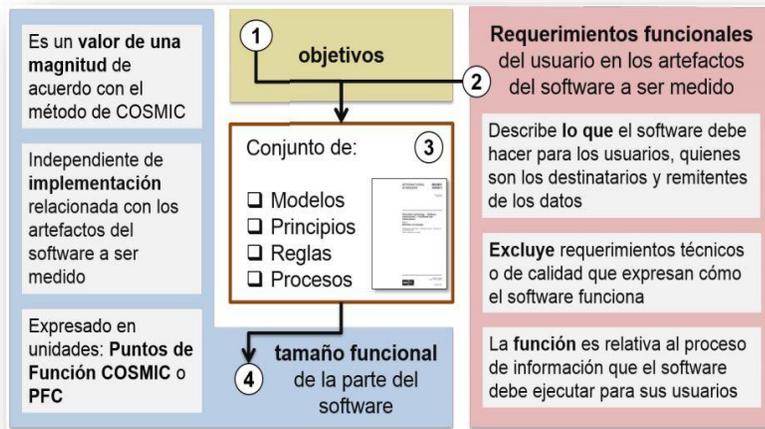
Los métodos anteriores no siempre tienen una aplicación amplia o suficiente para atender las necesidades del mercado ni funcionan apenas con acceso restringido. La planeación y medición del desempeño tiene mayor exactitud y además tiene la habilidad de capturar el tamaño a partir de múltiples perspectivas. (Vazquez, 2015)

El método COSMIC define los principios, reglas y un proceso para medir un tamaño funcional estándar de una pieza de software. El tamaño funcional es una medida de la cantidad de funcionalidad proporcionada por el software, completamente independiente de cualquier consideración técnica o de calidad.

El método COSMIC es un método estandarizado internacionalmente (ISO/IEC 19761) para medir un tamaño de los requisitos funcionales de la mayoría de los dominios de software, incluido el software de aplicación comercial (o sistema de información de gestión), software en tiempo real, software de infraestructura y algunos tipos de software científico/de ingeniería.

COSMIC significa Consorcio Internacional de Medición de Software Común. Fue formado en 1998 por un grupo de expertos en medición de software de Australia, Europa y América del Norte, con el objetivo de desarrollar un nuevo método para medir el tamaño del software basado en principios de ingeniería de software y criterios de metrología bien establecidos. Sus publicaciones son completamente abiertas y están disponibles para descarga gratuita. (COSMIC, 2019)

Figura 18
Visión general del COSMIC



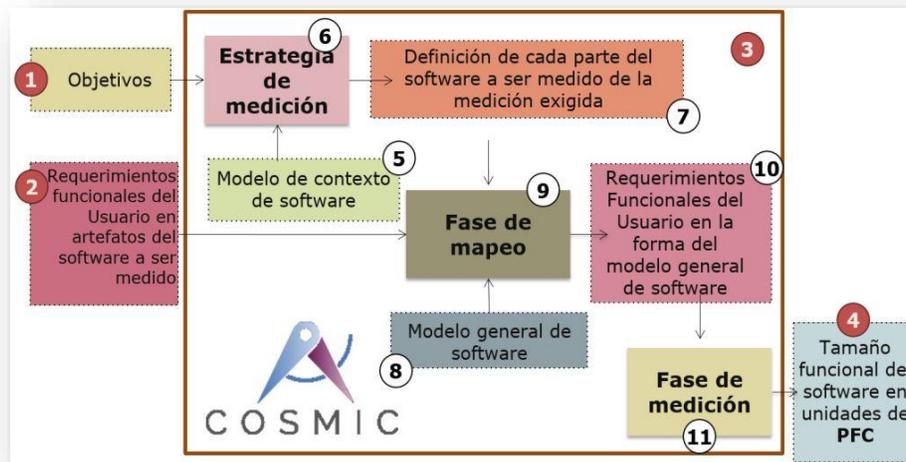
Nota: Información obtenida de (Vazquez, 2015)

2.21.1.1. Medición de Requerimientos Funcionales de Usuario

COSMIC fue diseñado para trabajar con requisitos funcionales en cualquier capa de la arquitectura de software y en cualquier grado de desglose de componentes. El método de medición COSMIC consiste en la aplicación de un conjunto de modelos, principios, reglas y procesos a los Requisitos Funcionales de los Usuarios (FUR) de una determinada aplicación software. El resultado es un valor de una cantidad numérico, que representa el tamaño funcional de la aplicación software de acuerdo con el método COSMIC.

El proceso de medición COSMIC se compone de tres fases:

Figura 19
El proceso de medición COSMIC



Nota: Información obtenida de (Vazquez, 2015)

2.21.1.2. Fase de estrategia de medición

En esta fase se describen los parámetros clave que deben ser considerados en la primera fase del proceso de medición, antes de realmente comenzar a medir. Estos son:

- **El propósito de la medición**, es decir, para lo que será utilizado el resultado. El propósito determina los otros parámetros de una medición.
- **El alcance global del software a ser medido** y, si el software se compone de más de una parte que deba ser medida por separado, los alcances de las mediciones de las partes individuales. También tenemos que determinar la capa en la que cada pieza de software se encuentra y tal vez el nivel de descomposición de las piezas de software que se desean medir.
- **Los usuarios funcionales** de cada pieza de software a medir. Estos son los emisores y destinatarios de los datos a/desde el software que se desea medir; pueden ser seres humanos, dispositivos de hardware u otras piezas de

software. Como los diferentes usuarios funcionales pueden tener requisitos para los diferentes subgrupos de la misma funcionalidad global, el tamaño funcional variará con la elección de los usuarios funcionales.

- **El nivel de granularidad** de los artefactos disponibles del software que se va a medir. Por ejemplo, tal vez la única especificación de los requisitos no se define en todo el detalle necesario para una medición precisa con el método COSMIC. Por tanto, debemos decidir cómo derivar los FUR a medir y/o si una variante de aproximación de tamaño se debería utilizar.

La determinación de estos parámetros ayuda a responder a las preguntas de qué tamaño debe medirse, qué tan preciso queremos la medición, etc. El registro de los parámetros habilita a los futuros usuarios de una medición para decidir la forma de interpretarla.

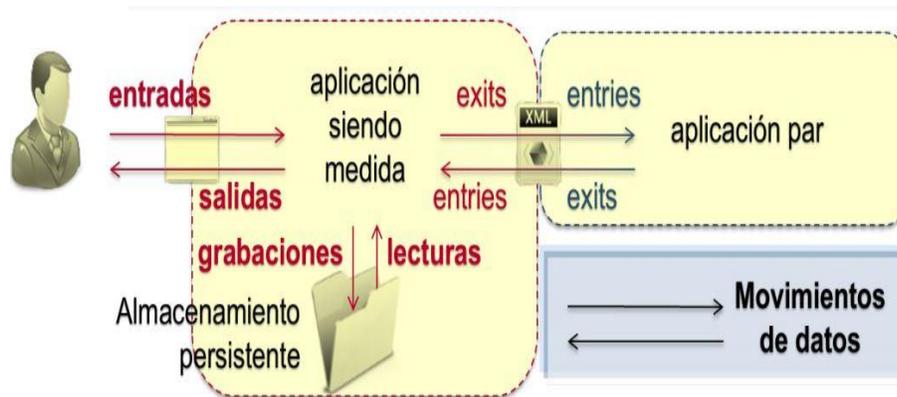
2.21.1.3. Fase de representación(Mapeo)

Segunda fase del proceso, de medición mediante la definición de los conceptos clave del Modelo Genérico de Software y el proceso a seguir en la representación de los FUR del software en el modelo, con la finalidad de que los FUR se puedan medir. Estos conceptos clave del Modelo Genérico de Software son:

- **Representación de los FUR al Modelo Genérico de Software:** Un evento hace que un usuario funcional solicite un servicio de la pieza de software que se está midiendo. Tal evento se llama un 'evento desencadenante' y el servicio solicitado un 'proceso funcional'.
- **Identificación de los procesos funcionales:** Los procesos funcionales se componen de dos tipos de subprocesos que, o bien mueven datos (movimientos de datos) o manipulan datos (manipulación de datos). Los subprocesos de manipulación de datos no se reconocen por separado, pero se considera que se explica por los movimientos de datos con los que están asociados.

- **Identificación objetos de interés y grupos de datos:** Un movimiento de datos mueve un grupo de datos. Un grupo de datos se compone de atributos de datos que todos describen un objeto de interés, es decir, un objeto que es de interés en el mundo del usuario funcional relacionado.
- **Identificando los movimientos de datos:** Hay cuatro tipos de movimientos de datos. Entradas y Salidas cada uno mueve un grupo de datos dentro y fuera de un proceso funcional a través de una frontera de/hacia un usuario funcional respectivamente. Lectura y Escritura cada un mueve un grupo de datos entre el proceso funcional y el almacenamiento persistente.

Figura 20
Movimientos de datos



Nota: Información obtenida de (Vazquez, 2015)

2.21.1.4. Fase de medición

Esta fase trata de la última etapa del proceso de medición. En primer lugar, se define la unidad de medida COSMIC. A continuación, se dan las reglas para asignar un tamaño a los FUR del software que se está midiendo. Las reglas se definen para saber cómo agregar tamaños de diferentes piezas de software. Además, las normas se definen para saber cómo medir cambios al software que son necesarios.

- **Aplicando la unidad de medida COSMIC:** cada movimiento de datos (Entrada, Salida, Lectura o Escritura) que se requiera ser añadido, modificado o eliminado por el software que se está midiendo también se mide como 1 CFP (Punto de Función COSMIC).
- **Agregando los resultados de medición:** Este paso consiste en sumar los resultados de todos los movimientos de datos identificados, en un único valor de tamaño funcional. Este paso se realiza según las siguientes reglas.
 - a) Para cualquier proceso funcional, el tamaño funcional de cada movimiento de datos individual debe ser agregado en un único valor de tamaño funcional en unidades de CFP para luego sumar todos juntos.

$$\begin{aligned} \text{Tamaño (proceso funcional}_i) &= \sum \text{tamaño(Entradas}_i) + \sum \text{tamaño(Salidas}_i) \\ &+ \sum \text{tamaño(Lecturas}_i) + \sum \text{tamaño(Escrituras}_i) \end{aligned}$$

- b) Para cualquier proceso funcional, el tamaño funcional de los cambios en sus Requisitos Funcionales de Usuario se sumará al tamaño de movimientos de datos que han sido añadidos, modificados o eliminados en el proceso funcional para dar un tamaño del cambio en unidades de CFP, según la siguiente fórmula.

$$\begin{aligned} \text{Tamaño (Cambio (proceso funcional}_i)) &= \sum \text{tamaño (movimientos datos añadidos}_i) \\ &+ \sum \text{tamaño (movimientos datos modificados}_i) \\ &+ \sum \text{tamaño (movimientos datos eliminados}_i) \end{aligned}$$

- c) El tamaño de una pieza de software dentro de alcance definido se obtendrá sumando los tamaños de sus procesos funcionales, sujeto a las normas e) y f) a continuación.
- d) El tamaño de cualquier cambio en una pieza de software dentro de un alcance definido se obtendrá sumando los tamaños de todos los cambios de todos los procesos funcionales, sujeto a las normas e) y f) a continuación.

- e) Los Tamaños de piezas de software o de cambios de piezas de software podrán sumarse sólo si se mide al mismo nivel de granularidad de los FUR el proceso funcional.
- f) Los Tamaños de piezas de software y/o cambios en los tamaños de piezas de software dentro de una capa o de diferentes capas serán sumados sólo si tiene sentido hacerlo, a efectos de la medición.
- g) El tamaño de una pieza de software se obtiene sumando los tamaños de sus componentes (independientemente de cómo se descompone) proporcionados
 - el tamaño de las contribuciones de los movimientos de datos inter-componentes son eliminadas y
 - sólo una Salida se identifica para todos los mensajes de error/confirmación emitidos por un proceso funcional a un usuario funcional humano.
- h) Si el método COSMIC se extiende localmente, entonces el tamaño medido por la extensión local debe ser informado por separado y no puede ser añadido al tamaño obtenido por el método estándar, medido en CFP.

2.22. Herramientas

2.22.1. Servidor Web Apache

El servidor Apache es considerado un servidor web de código abierto y de libre distribución; que puede ser usado en sistemas como Windows, Linux, Macintosh y otros.

Apache es una aplicación que permite implementar un servidor web en su computadora personal, asignándole un nivel de servidor local no importando el sistema operativo donde se encuentra, ya que tiene compatibilidad abierta. Su más cercana competencia es el Internet Information Server, más conocido como IIS, que pertenece a Microsoft y tiene las mismas funcionalidades de apache, pero es considerado un software propietario; por lo tanto, solo funciona para sistemas Microsoft.

Apache tiene una fuerte afinidad con el lenguaje de programación PHP por eso tiene librerías que soportan al PHP. (Torres, 2014)

2.22.1.1. Características de Apache

una de las principales características que presenta Apache es que funciona en plataformas virtuales muy utilizadas. Al principio, Apache se utilizaba para ser el primer Servidor Web basado en Unix, pero ya no es verdad. Apache no solo funciona en la mayoría de las versiones de Unix, sino que además funciona en Windows y en muchos otros sistemas operativos de escritorio y de tipo servidor.

Apache presenta muchas otras características, entre ellas un elaborado índice de directorios; un directorio de alias; negociación de contenidos; informe de errores HTTP configurable; ejecución SetUID de programas CGI; gestión de recursos para procesos hijos; integración de imágenes del lado del servidor; reescritura de las URL; comprobación de la ortografía de las URL; y manuales online. (Kabir, 2003).

2.22.2. Gestor de Base de Datos PostgreSQL

PostgreSQL es un sistema de gestión de base de datos relacional de objetos basado en POSTGRES, Versión 4.2, desarrollado en el Departamento de Informática de la Universidad de California en Berkeley. POSTGRES fue pionero en muchos conceptos que solo estuvieron disponibles en algunos sistemas de bases de datos comerciales mucho más tarde.

PostgreSQL es un descendiente de código abierto de este código original de Berkeley. Admite una gran parte del estándar SQL y ofrece muchas características modernas (consultas complejas, llaves extranjeras, disparadores, vistas actualizables, integridad transaccional, control de concurrencia multivisión). Además, PostgreSQL puede ser extendido por el usuario de muchas maneras, por ejemplo, agregando nuevos (tipos de datos, funciones, operadores, Funciones agregadas, métodos de índice, lenguajes de procedimiento)

Y debido a la licencia liberal, PostgreSQL puede ser utilizado, modificado y distribuido por cualquier persona de forma gratuita para cualquier propósito, ya sea privado, comercial o académico. (The PostgreSQL Global Development Group, 2019, p. 30)

2.22.2.1. Características

- **Tipos de datos:** Primitivas, Estructurado, Documento, Geometría, Personalizaciones.
- **Integridad de los datos:** ÚNICO, NO NULO, Claves primarias, Llaves extranjeras, Restricciones de exclusión, Cerraduras explícitas, cerraduras de aviso.
- **Concurrencia, rendimiento:** Indexación, Indexación avanzada, optimizador de consultas, escaneos de solo índice, estadísticas de varias columnas, Transacciones, transacciones anidadas, Control de concurrencia de versiones múltiples (MVCC), Paralelización de consultas de lectura y creación de índices de árbol B, Particionamiento de tabla, Todos los niveles de aislamiento de transacciones definidos en el estándar SQL, Just-in-time (JIT) compilación de expresiones.
- **Fiabilidad, recuperación ante desastres:** Registro de escritura anticipada (WAL), Replicación asíncrona, síncrona, lógica, Point-in-time-recuperación (PITR), recursos seguros activos, Tablespace.
- **Seguridad:** Autenticación mediante GSSAPI, SSPI, LDAP, SCRAM-SHA-256, y más, Robusto sistema de control de acceso, Seguridad de columna y nivel de fila, Autenticación multifactor con certificados y un método adicional.
- **Extensibilidad:** Funciones y procedimientos almacenados, Lenguajes de procedimiento (PL/PGSQL, Perl, Python y muchos más), Expresiones de ruta SQL/JSON, Conectar con otras bases de datos o arroyos con una interfaz estándar SQL, Interfaz de almacenamiento personalizable para tablas, Muchas extensiones que proporcionan funcionalidad adicional, incluido PostGIS.

- **Internacionalización:** Búsqueda de texto, Soporte para juegos de caracteres internacionales, por ejemplo, a través de intercalaciones de UCI, Colaciones insensibles a mayúsculas y minúsculas

2.22.3. PHP

PHP es el lenguaje interpretado de lado del servidor que se caracteriza por su potencia, versatilidad, robustez y modularidad. Los programas escritos en PHP son embebidos directamente en el código HTML y ejecutado por el servidor web a través de un intérprete antes de transferir al cliente que lo ha solicitado un resultado en forma de código HTML puro. Al ser un lenguaje que sigue la corriente open source, tanto el intérprete como su código fuente son totalmente accesibles de forma gratuita en la red.

Por su flexibilidad, PHP resulta un lenguaje muy sencillo de aprender; especialmente para programadores familiarizados con el lenguaje como C, Perl o Java, debido a las similitudes de sintaxis entre ellos.

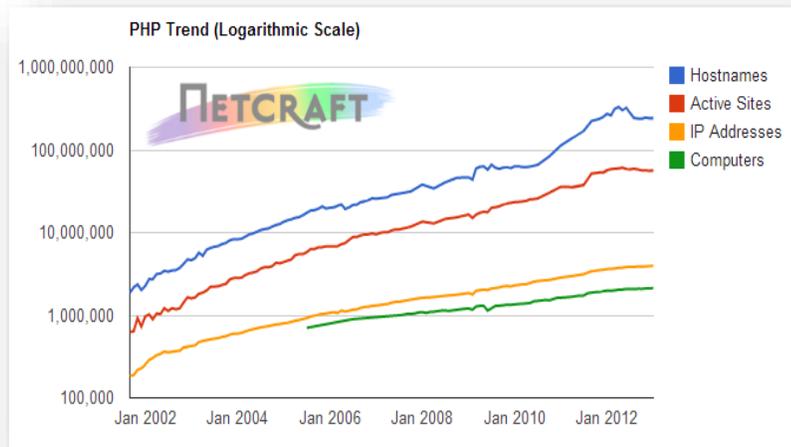
Por supuesto, es un lenguaje multiplataforma; los programas funcionan igual sobre diferentes plataformas, trabajando sobre la mayoría de servidores web y estando preparado para interactuar con más de 20 tipos de bases de datos. No obstante, al ser un lenguaje inicialmente concebido para entornos Unix, es sobre este sistema operativo sobre el que se pueden aprovechar mejor sus prestaciones.

Inicialmente diseñado para realizar poco más que contadores y libros de visita de páginas, en la actualidad PHP permite realizar una multitud de tareas útiles para el desarrollo web. Por ejemplo, dispone entre otras, de:

- Funciones de correo electrónico que pueden ser utilizados para programar completos sistemas de correo electrónico vía web.
- Funciones de administración y gestión de bases de datos específicas para la mayoría de gestores comerciales y funciones para conexiones ODBC con bases de datos en sistemas Microsoft.

- Funciones de gestión de directorios y ficheros, incluso para la transferencia mediante FTP.
- Funciones de tratamiento de imágenes y librerías de funciones graficas
- Funciones de generación y lectura de cookies.
- Funciones para la generación de documentos PDF. (Cobo, et al., 2005)

Figura 21
Tendencia PHP (Escala logarítmica)



Nota: Información obtenida de (Netcraft, 2013)

2.22.4. Laravel

Laravel es un framework de código abierto para el desarrollo de aplicaciones web en PHP 5 que posee una sintaxis simple, expresiva y elegante. Fue creado en 2011 por Taylor Orwell, inspirándose en Ruby on Rails y Symfony, de los cuales ha adoptado sus principales ventajas.

Laravel facilita el desarrollo simplificando el trabajo con tareas comunes como la autenticación, el enrutamiento, gestión sesiones, el almacenamiento en caché, etc. Algunas de las principales características y ventajas de Laravel son:

- Está diseñado para desarrollar bajo el patrón MVC (modelo - vista - controlador), centrándose en la correcta separación y modularización del código. Lo que facilita el trabajo en equipo, así como la claridad, el mantenimiento y la reutilización del código.
- Integra un sistema ORM de mapeado de datos relacional llamado Eloquent, aunque también permite la construcción de consultas directas a base de datos mediante su Query Builder.
- Permite la gestión de bases de datos y la manipulación de tablas desde código, manteniendo un control de versiones de las mismas mediante su sistema de Migraciones.
- Utiliza un sistema de plantillas para las vistas llamado Blade, el cual hace uso de la cache para darle mayor velocidad. Blade facilita la creación de vistas mediante el uso de layouts, herencia y secciones.
- Facilita la extensión de funcionalidad mediante paquetes o librerías externas. De esta forma es muy sencillo añadir paquetes que nos faciliten el desarrollo de una aplicación y nos ahorren mucho tiempo de programación.

Incorpora un intérprete de línea de comandos llamado Artisan que nos ayudará con un montón de tareas rutinarias como la creación de distintos componentes de código, trabajo con la base de datos y migraciones, gestión de rutas, cachés, colas, tareas programadas, etc. (Gallego, 2018)

2.22.5. Livewire

Livewire es un framework full-stack escrito en PHP para Laravel, que permite a los desarrolladores crear aplicaciones web dinámicas e interactivas sin necesidad de escribir código JavaScript. Con Livewire, puedes crear componentes de la interfaz de usuario que se actualizan de forma asíncrona sin necesidad de recargar la página, lo que hace que sea más fácil crear aplicaciones web enriquecedoras y fluidas.

Se basa en el concepto de componentes, que son fragmentos reutilizables de código que pueden ser utilizados para crear una interfaz de usuario. Cada componente puede tener su propia lógica de negocio y su propia vista, lo que hace que sea fácil crear interfaces de usuario complejas y reutilizables.

También incluye soporte para validación de formularios y manejo de errores, lo que lo hace ideal para aplicaciones que necesitan procesar datos del usuario y validar la entrada. En resumen, Livewire es una herramienta muy útil para crear aplicaciones web dinámicas y fluidas sin tener que escribir mucho código JavaScript.

Pero no sólo eso, además nos ofrece la posibilidad de hacer todo lo que nos ofrece Vuejs, emisión de eventos, propiedades calculadas, ciclo de vida de un componente, validaciones en tiempo real y cualquier cosa que puedas imaginar, y lo mejor es que todo nos permite hacerlo escribiendo código PHP con Laravel. (Israel, 2022)

2.22.6. Tailwind Css

Tailwind CSS es un framework CSS que permite un desarrollo ágil, basado en clases de utilidad que se pueden aplicar con facilidad en el código HTML y unos flujos de desarrollo que permiten optimizar mucho el peso del código CSS.

Tailwind CSS es una potente herramienta para el desarrollo frontend. Está dentro de la clasificación de los frameworks CSS o también llamados frameworks de diseño. Permite a los desarrolladores y diseñadores aplicar estilos a los sitios web de una manera ágil y optimizada.

Permite escribir los estilos por medio de clases que se incluyen dentro del código HTML y que afectan a un aspecto muy concreto y específico de las CSS, por ejemplo, el fondo de un elemento, el color del texto o simplemente el margen por la parte de arriba. Este enfoque se conoce como "Atomic CSS", por aplicarse mediante estilos muy determinados y simples. En Tailwind CSS a estas clases se les llama "utility classes" o clases de utilidad en español. (Miguel, 2020)

2.22.7. Alpine Js

Es una librería javascript creada por Caleb Porzio, también creador del componente Livewire para Laravel. Está inspirada en otros frameworks como AngularJS, VueJS o TailwindCSS que nos permite enriquecer nuestro lenguaje HTML con propiedades declarativas y reactivas de una manera fácil, rápida y ligera, como alternativa a frameworks como React.js y Vue.js que con su crecimiento empiezan a requerir de gestores de tareas para facilitarnos su manejo.

Alpine nos ofrece 14 directivas y 6 propiedades mágicas. Por directivas vamos a entender que son una serie de funciones que podemos agregar a nuestro código HTML para hacerlo "más inteligente" y nos permita emplear esta propiedad reactiva en nuestra aplicación. Nosotros únicamente vamos a utilizar las siguientes:

- x-data: Nos permite declarar un nuevo scope (espacio de trabajo) del componente.
- x-show: Muestra, o no, un elemento dependiendo de un resultado booleano.
- x-bind: Asigna el valor de un atributo a partir del resultado de una expresión javascript.
- x-model: Mantiene la entrada del elemento sincronizado con los datos del componente.
- x-on: Adjunta un evento a un elemento y ejecuta una expresión javascript cuando se emite el evento.
- x-text: Actualiza el texto que contiene un elemento.
- x-for: Permite crear tantos nodos en el DOM como elementos contenga un array dado. (Aguilar, 2020)

2.22.8. MagicDraw

MagicDraw es una herramienta visual de modelado UML y CASE con soporte de trabajo en equipo. Diseñada para analistas de negocios, analistas de software,

programadores, ingenieros de control de calidad y redactores de documentación, esta herramienta de desarrollo dinámica y versátil facilita el análisis y el diseño de sistemas y bases de datos orientados a objetos (OO). Proporciona el mejor mecanismo de ingeniería de código de la industria (con soporte completo de ida y vuelta para Java, C #, C ++, WSDL, Esquema XML y lenguajes de programación CORBA IDL), así como modelado de esquemas de bases de datos, generación de DDL e instalaciones de ingeniería inversa. (No Magic, Inc., 2015)

CAPITULO III

3. MARCO APLICATIVO

3.1. Introducción

La aplicación del proyecto se realizará siguiendo la metodología UWE, cuyo objetivo es determinar el análisis, diseño y desarrollo de Sistema Integral Para el Control y Administración de Insumos Médicos y Personal Administrativo para mostrar las ventajas que otorga el sistema.

3.2. Obtención de requisitos

En este apartado se presentan los requisitos que son satisfechos por el sistema. Todos los requisitos aquí expuestos son esenciales. Los requisitos se especifican de manera que sea fácil comprobar si el sistema los ofrece o no y si los ofrece de manera adecuada.

Tabla 4

Técnica de obtención de requisitos

TAREA	CARACTERISTICAS
Entrevista	Se realizó las entrevistas en la clínica de especialidades Adolfo Kolping con el siguiente personal: Director de la clínica de especialidades Adolfo Kolping que a la vez es médico especialista en la clínica. Administrador de Sistemas. Encargado de insumos.
Documentación	Se tuvo acceso a algunos documentos físicos y digitales con los que cuenta la organización (Anexo).
Observaciones	En la clínica de especialidades Adolfo Kolping se presentan problemas al momento de administrar los insumos médicos y personal administrativo ya que los procesos se realizan de forma manual.

3.2.1. Descripción de los actores

La identificación de Actores nos permitió conocer a las personas involucradas en el proceso de administración de la clínica de especialidades ADOLFO KOLPING, a objeto de formar los casos de uso. A continuación, se muestra la lista de actores, junto con una descripción de sus actividades relacionadas con el sistema.

- **Director:** Máximo responsable y líder del establecimiento médico. Además de su rol de director, es un médico cirujano con experiencia en el campo de la medicina.
Supervisar y tomar decisiones estratégicas para la clínica.
Participar en la toma de decisiones médicas cruciales y administrativas.
- **Administrador/ Sistemas:** Personal encargado del control y administración del sistema informático de la clínica y de la base de datos de insumos médicos y personal administrativo.
Realizar funciones de adición, edición y eliminación de registros en el sistema.
Asegurar la disponibilidad y seguridad de la información.
- **Encargado de insumos:** Encargado de gestionar los recursos médicos y asegurar su disponibilidad para el personal médico.
Realiza funciones de registrar, editar y eliminar insumos médicos en el sistema.
Coordinar la distribución de insumos a las áreas correspondientes.
Mantener un inventario actualizado de los insumos médicos.
- **RRHH:** Responsable de gestionar el personal de la clínica, desde la contratación hasta la administración de recursos humanos.
Realiza funciones de registrar y mantener la información del personal de la clínica.
Coordinar procesos de contratación y desvinculación de empleados.
Gestionar aspectos relacionados con beneficios y capacitación del personal.
- **Solicitante:** Los Solicitantes son el conjunto de todo el personal de la clínica que utiliza el sistema para llevar a cabo sus tareas diarias.

Acceder al sistema para registrar información relevante y realiza distintos tipos de solicitudes según sus necesidades.

3.3. Análisis de Requerimientos

En la especificación de requerimientos hay que identificar cada función y listarlas en grupos lógicos, Estas funciones se pueden clasificar en 3 categorías:

- **Evidentes:** Las evidentes deben realizarse, y el usuario debe saber que se han realizado.
- **Ocultas:** Las ocultas también deben realizarse, y puede que no sean visibles para el usuario. Muchas de estas funciones se omiten (erróneamente) durante el proceso de obtención de requerimientos.
- **Superfluas:** Las superfluas son opcionales, y su inclusión no repercute significativamente en el costo ni en otras funciones.

3.3.1. Requisitos funcionales

Los requisitos funcionales se detallan a continuación en la siguiente tabla, se muestran las características que necesita el sistema a partir de la información obtenida como parte de las tareas de obtención de requisitos.

Tabla 5
Requisitos Funcionales

Nro.	Descripción/función	Categoría
RF01	Consultar reportes e indicadores del sistema	Evidente
RF02	Aprobar solicitudes importantes	Evidente
RF03	Gestionar usuarios y permisos.	Evidente
RF04	Configurar parámetros del sistema	Evidente

RF05	Registro de información de personal administrativo	Evidente
RF06	Realizar copias de seguridad de la BD	Evidente
RF07	Registro de insumos médicos	Evidente
RF08	Registro y actualización de stock de insumos	Evidente
RF09	Registro de proveedores	Evidente
RF10	Generación de solicitudes de materiales	Evidente
RF11	Registro de entradas de insumos	Evidente
RF12	Registro de salidas de insumos	Evidente
RF13	Distribución y control de insumos médicos	Evidente
RF14	Registro de roles de usuarios	Evidente
RF15	Permisos de usuarios	Evidente
RF16	Registro y aprobación de licencias y permisos	Evidente
RF17	Gestión del personal	Evidente
RF18	Generar reporte del personal	Evidente
RF19	Registro de ausencias del personal y solicitudes de permisos	Oculto
RF20	Gestión usuarios no activos	Oculto
RF21	Envío de notificaciones y alertas	Evidente
RF22	Visualización de información detallada de artículos	Evidente

RF23	Autenticación y control de acceso	Evidente
RF24	Exportación de reportes a Excel	Evidente
RF25	Exportación de reportes a Pdf	Evidente
RF26	Registro de usuarios	Evidente
RF27	Registro de cargos o puestos de trabajo	Evidente
RF28	Gestión usuarios	Evidente
RF29	Gestión de roles	Evidente
RF30	Generar reportes de ingreso y salida de artículos	Evidente
RF31	Realizar búsqueda avanzada de artículos	Evidente
RF32	Monitoreo de performance del sistema	Oculto
RF33	Copias de seguridad automáticas	Oculto
RF34	Cálculo automático de indicadores	Oculto
RF35	Gestión de proveedores	Evidente
RF36	Verificación de consistencia de datos ingresados	Oculto
RF37	Generar reporte de artículos	Evidente
RF38	Generar reporte de movimiento artículos	Evidente
RF39	Validaciones de rangos y formatos en formularios	Oculto
RF40	Encriptación de datos sensibles en la base de datos	Oculto

RF41	Monitorización de accesos fallidos e intentos de intrusión	Oculto
RF42	Generar reporte de solicitudes	Evidente
RF43	Configuración del perfil de usuario	Oculto
RF44	Rastreo de caducidades de insumos	Oculto
RF45	Búsqueda de usuarios mediante fechas	Evidente
RF46	Búsquedas mediante paginas	Evidente
RF47	Búsqueda del listado del personal mediante fechas	Evidente
RF48	Búsqueda de personal por diferentes criterios (nombre, cargo, fecha ingreso, etc)	Evidente
RF49	Búsqueda de insumos médicos por diferentes criterios (nombre, código, proveedor, etc)	Evidente
RF50	Gestión de Vacaciones y Ausencias	Evidente
RF51	Gestión de solicitudes de cambio en el sistema	Oculto
RF52	Escalabilidad del sistema	Oculto
RF53	Registro de solicitudes de compra de insumos.	Oculto
RF54	Resguardar las instrucciones SQL realizadas en las tablas o entidades más importantes del sistema	Oculto

3.3.2. Requisitos No Funcionales

Tabla 6

Requisitos No Funcionales

Nro.	Descripción
RNF1	El sistema debe poseer interfaces amigables al usuario
RNF2	El sistema debe poseer un diseño Responsive a fin de garantizar la adecuada visualización en múltiples dispositivos
RNF3	El sistema debe proporcionar mensajes de error que sean informativos y orientados a usuario final
RNF4	El sistema debe permitir al usuario ingresar desde cualquier dispositivo con acceso a internet
RNF5	El sistema debe asegurar que los datos estén protegidos del acceso no autorizado
RNF6	El sistema debe contar con manuales de usuario estructurados adecuadamente

3.3.3. Modelo de Requisitos

En esta sección se plasma el análisis de requerimientos del sistema mediante el diseño del modelo de requisitos (Diagramas de casos de uso) que describe el comportamiento del sistema frente a las acciones de los actores o usuarios del mismo, así como las funcionalidades del sistema.

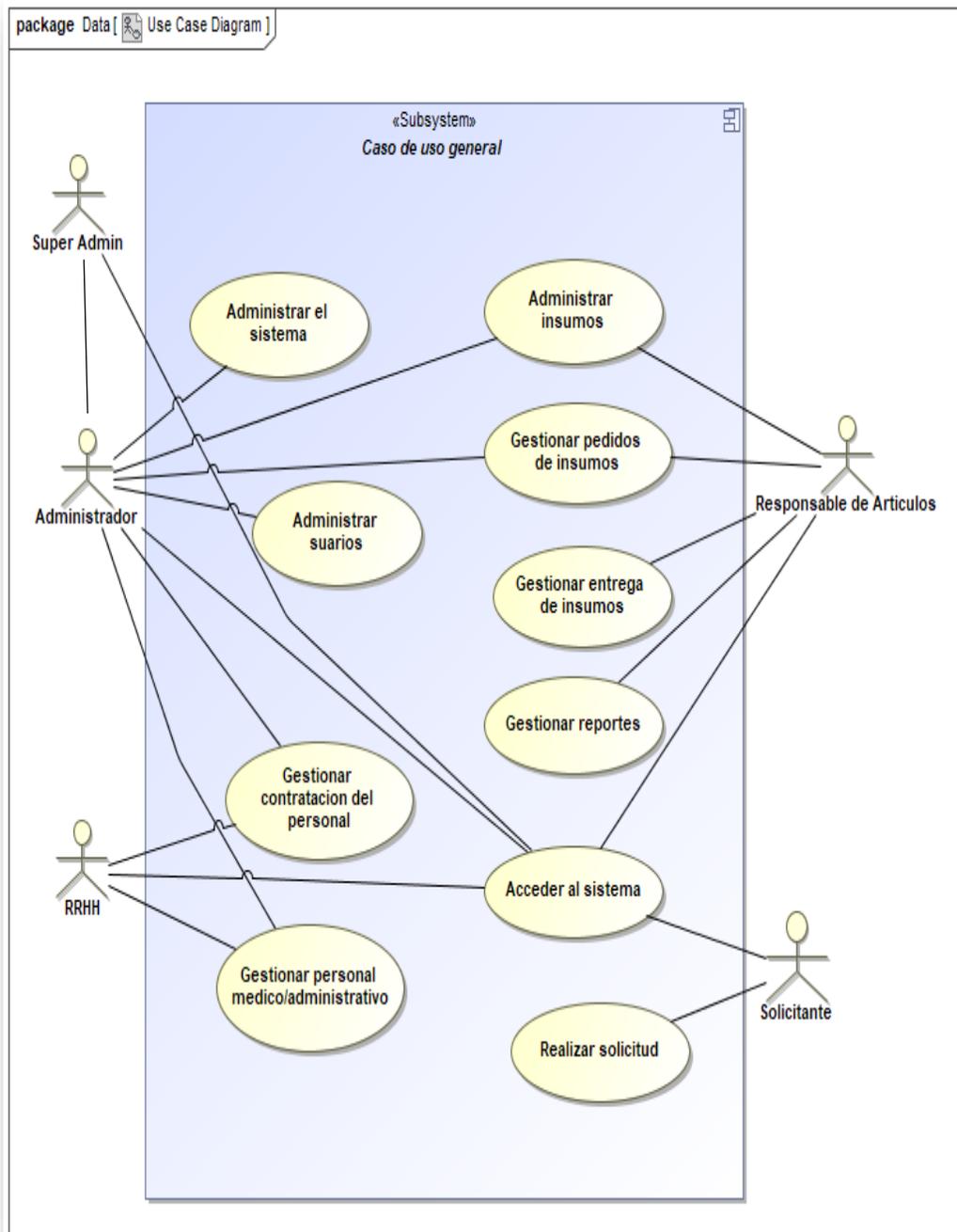
- **Diagrama de Caso de Uso general**

A continuación, se realiza el modelado donde se aprecia la interacción de los actores

sobre los casos de uso del sistema.

Figura 22

Diagrama de caso de uso general del sistema



- Diagrama de Caso de Uso Perfil Administrador

Figura 23

Diagrama de Caso de Uso Perfil Administrador

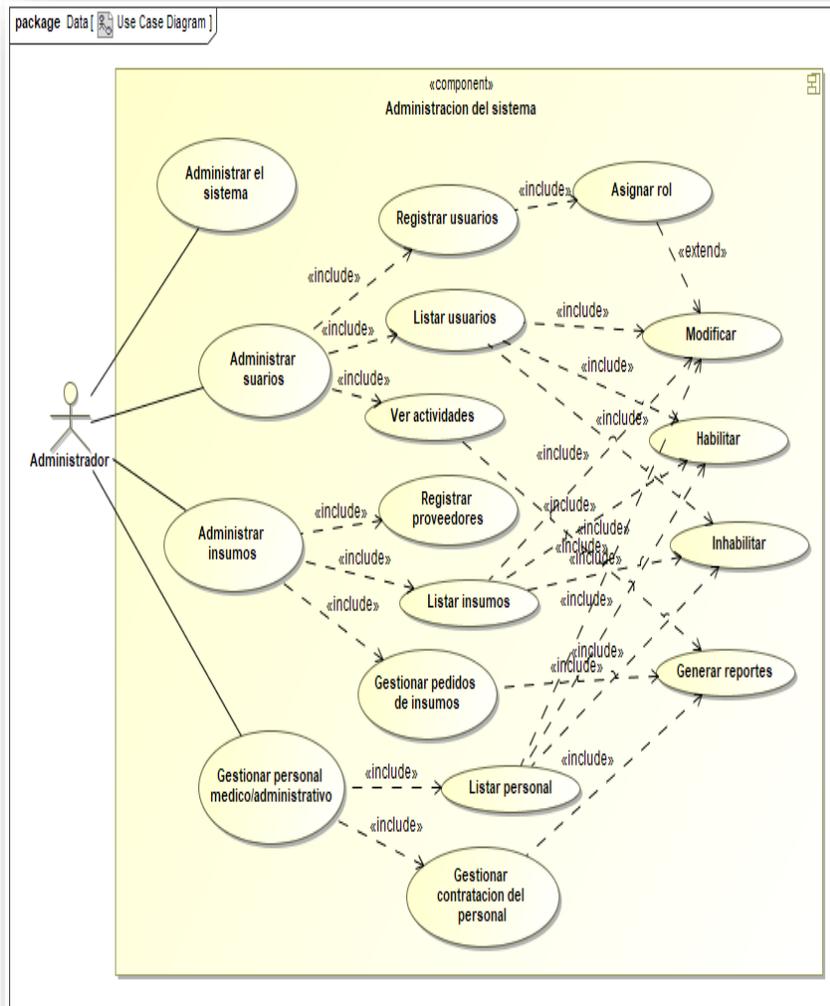


Tabla 7
Caso de Uso Perfil Administrador

Caso de Uso: Administración del sistema	
Actor	Administrador
Tipo	Primario
Descripción	El administrador registra, modifica y designa el rol de cada usuario en base a los requerimientos. Restringe el acceso al sistema habilitando o deshabilitando a un usuario.

- **Diagrama Caso de Uso Perfil Gerente General**

Figura 24
Diagrama Caso de Uso Perfil RRHH

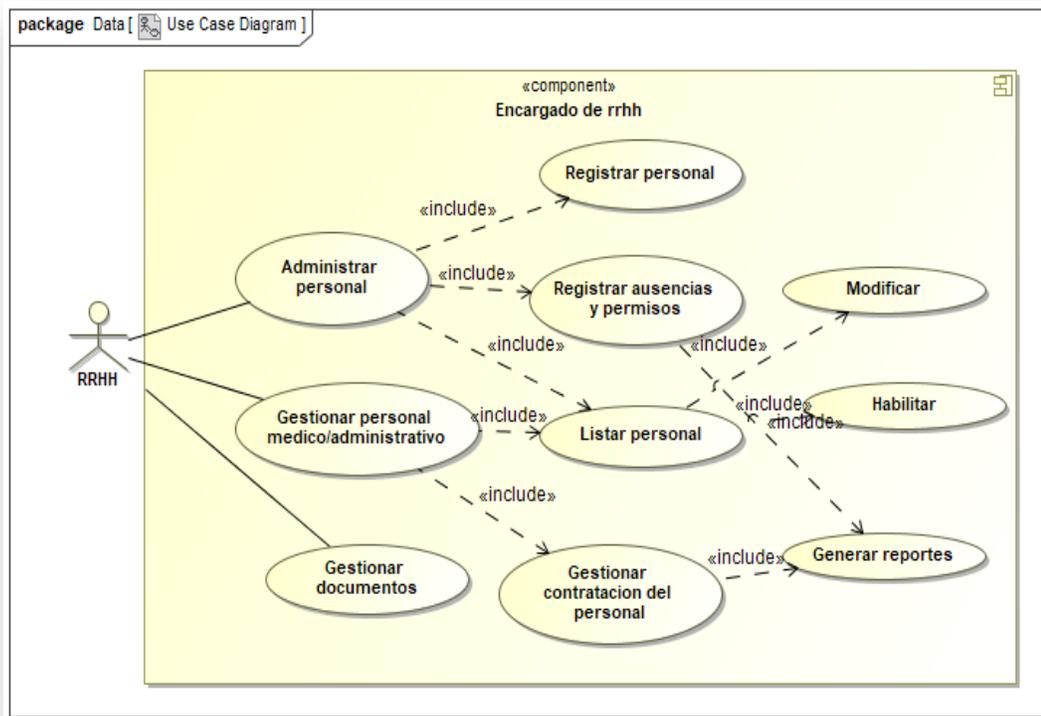


Tabla 9
Caso de Uso Perfil Encargado de Insumos

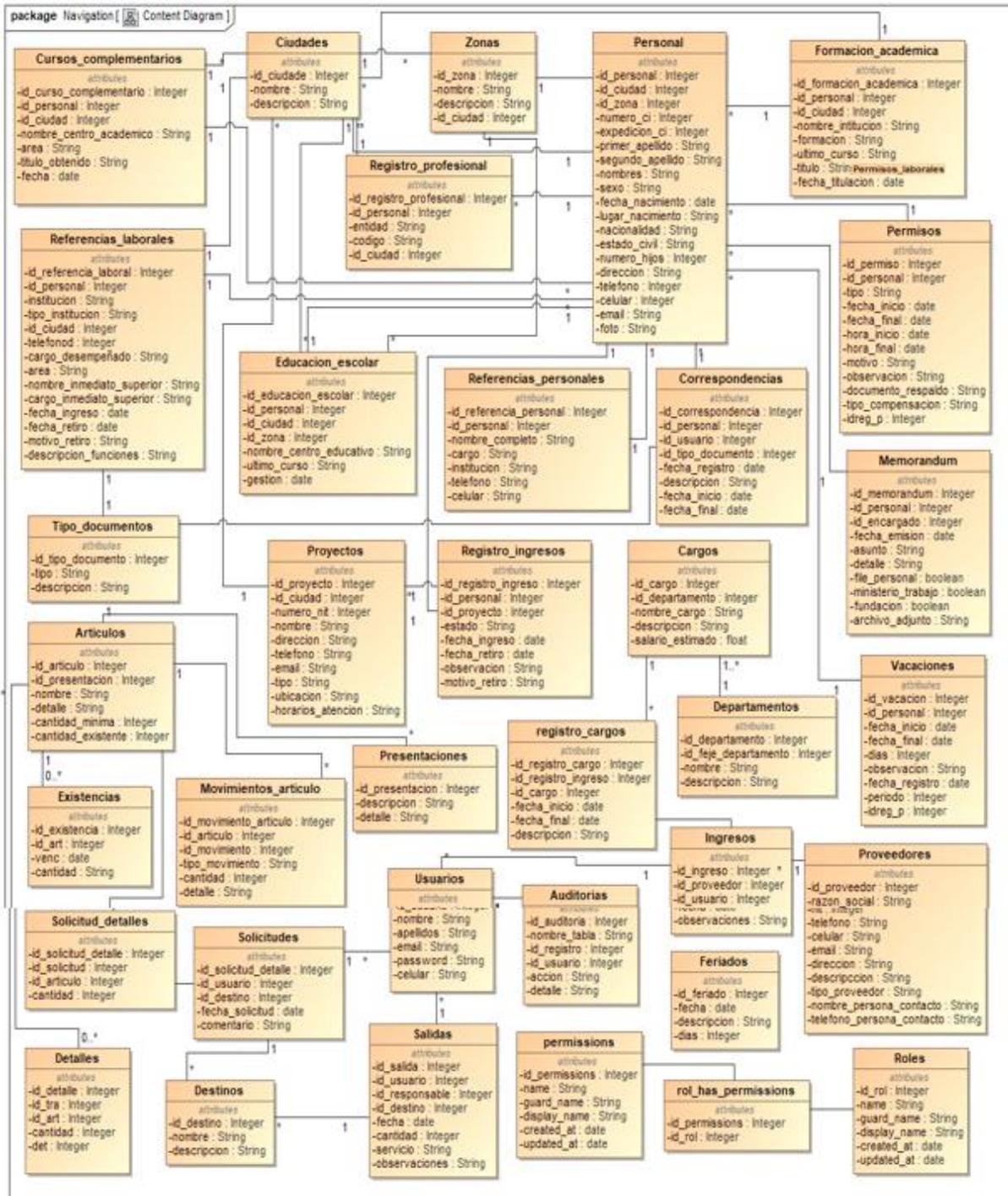
Caso de Uso: Encargado de Insumos	
Actor	Encargado de Insumos y Solicitante
Tipo	Secundario
Descripción	El encargado de Insumos tiene acceso a las listas de todos los insumos médicos. Su función principal es garantizar la disponibilidad y el adecuado abastecimiento de insumos médicos esenciales para el funcionamiento de la clínica. se encarga de coordinar la distribución de insumos a los solicitantes de las distintas áreas. Además, tiene permitido la administración de los insumos (registrar productos, editar, elimina, etc.) y las opciones de generar reportes.

3.4. Diseño del Sistema

3.4.1. Modelo de Contenido

A continuación, se elabora el modelo de contenido donde se puede apreciar la información relevante del dominio para el sistema web.

Figura 26
Diagrama de Contenido



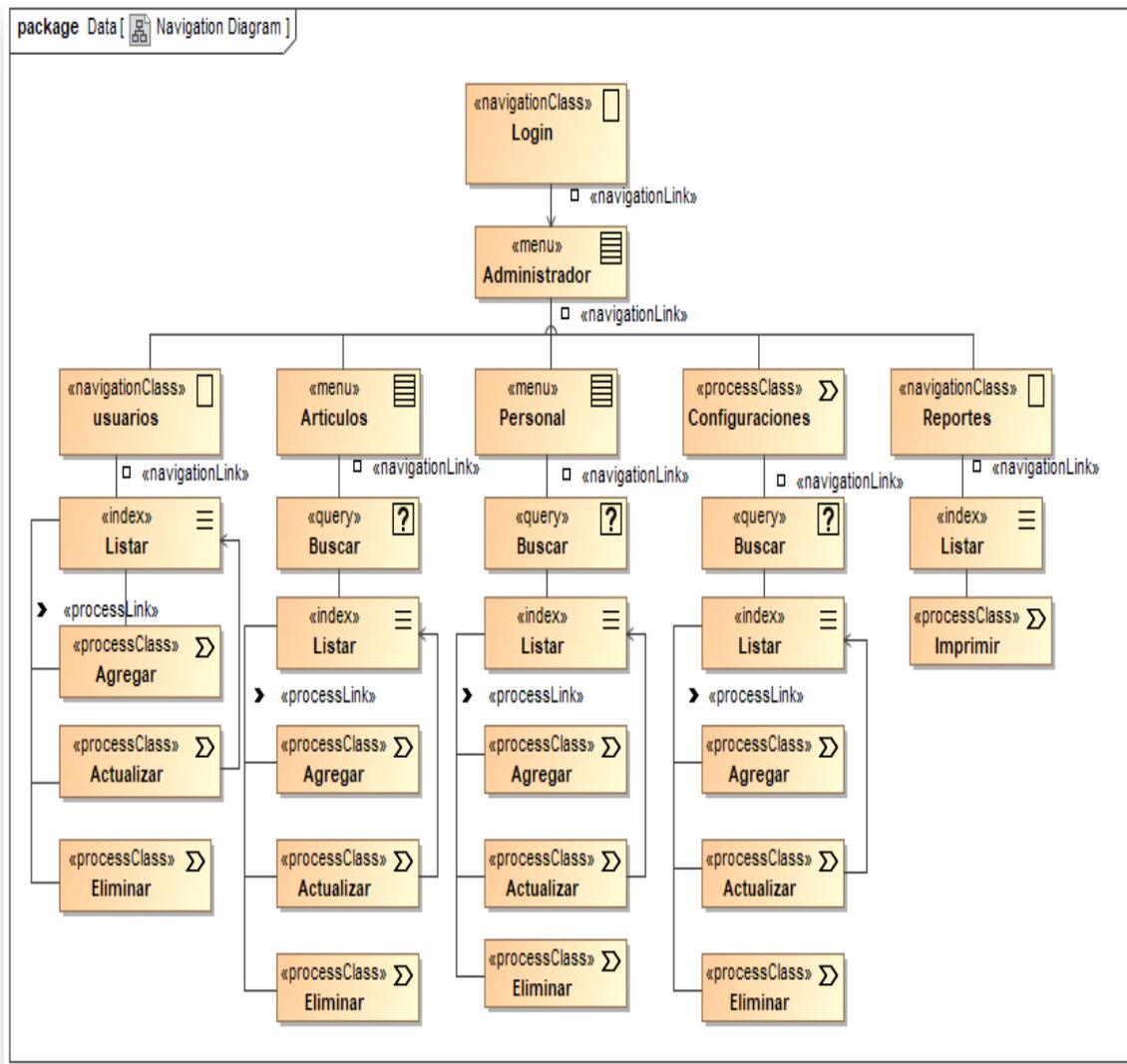
3.4.2. Modelo de Navegación

A continuación, se hace el modelo donde se aprecia la interacción de los usuarios en la navegación del sistema:

3.4.2.1. Modelo de navegación: Administrador

Figura 27

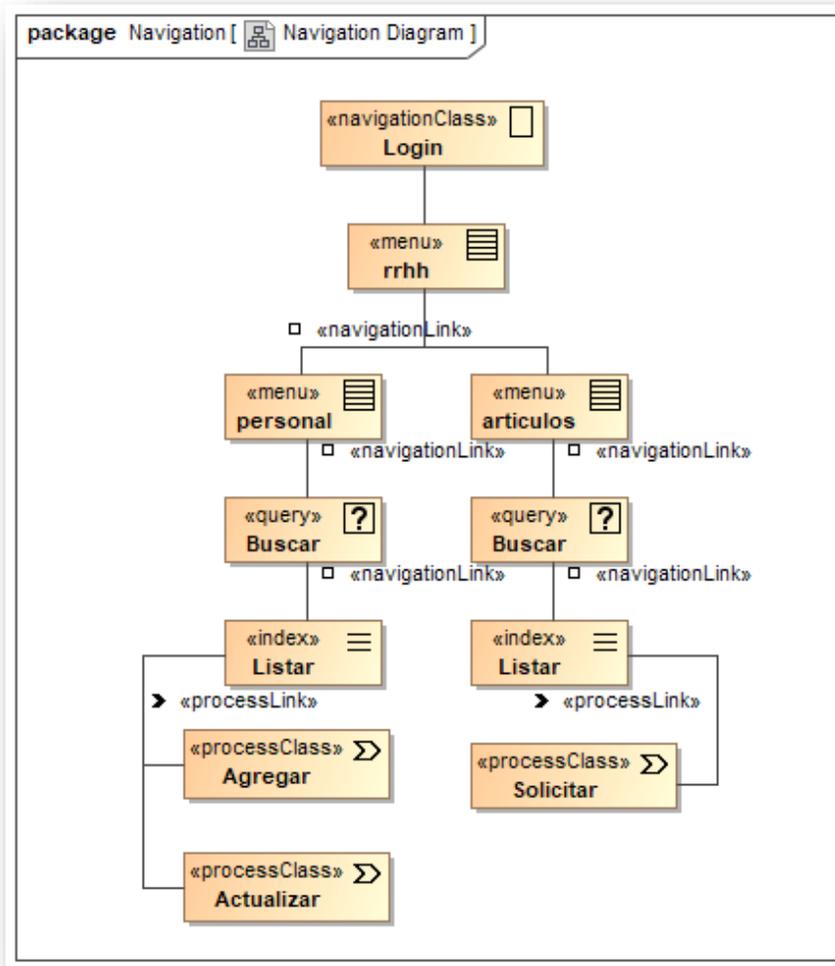
Diagrama de Navegación Administrador



3.4.2.2. Modelo de navegación: RRHH

Figura 28

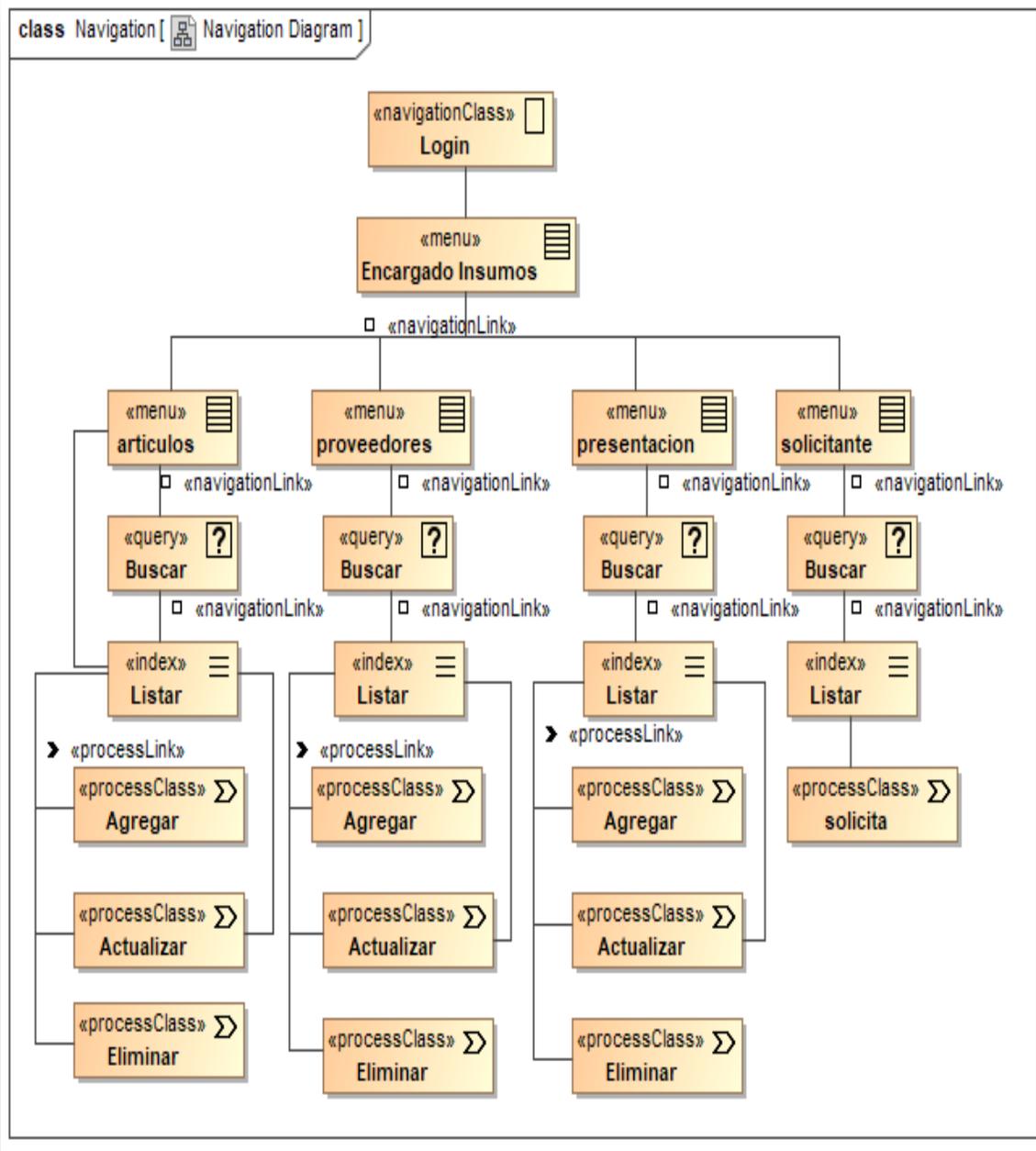
Diagrama de Navegación Recursos Humanos



3.4.2.3. Modelo de Navegación: Encargado de insumos

Figura 29

Diagrama de Navegación Encargado de Insumos

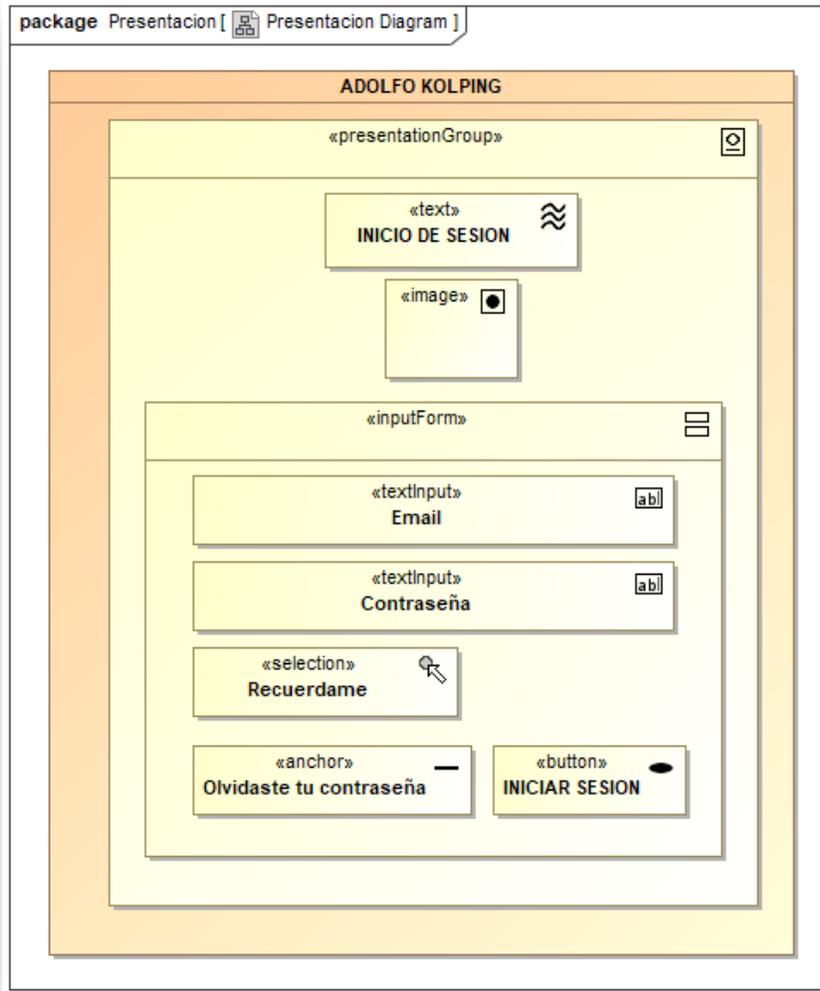


3.5. Modelo de Presentación

3.5.1. Modelo de Presentación: Login

Figura 30

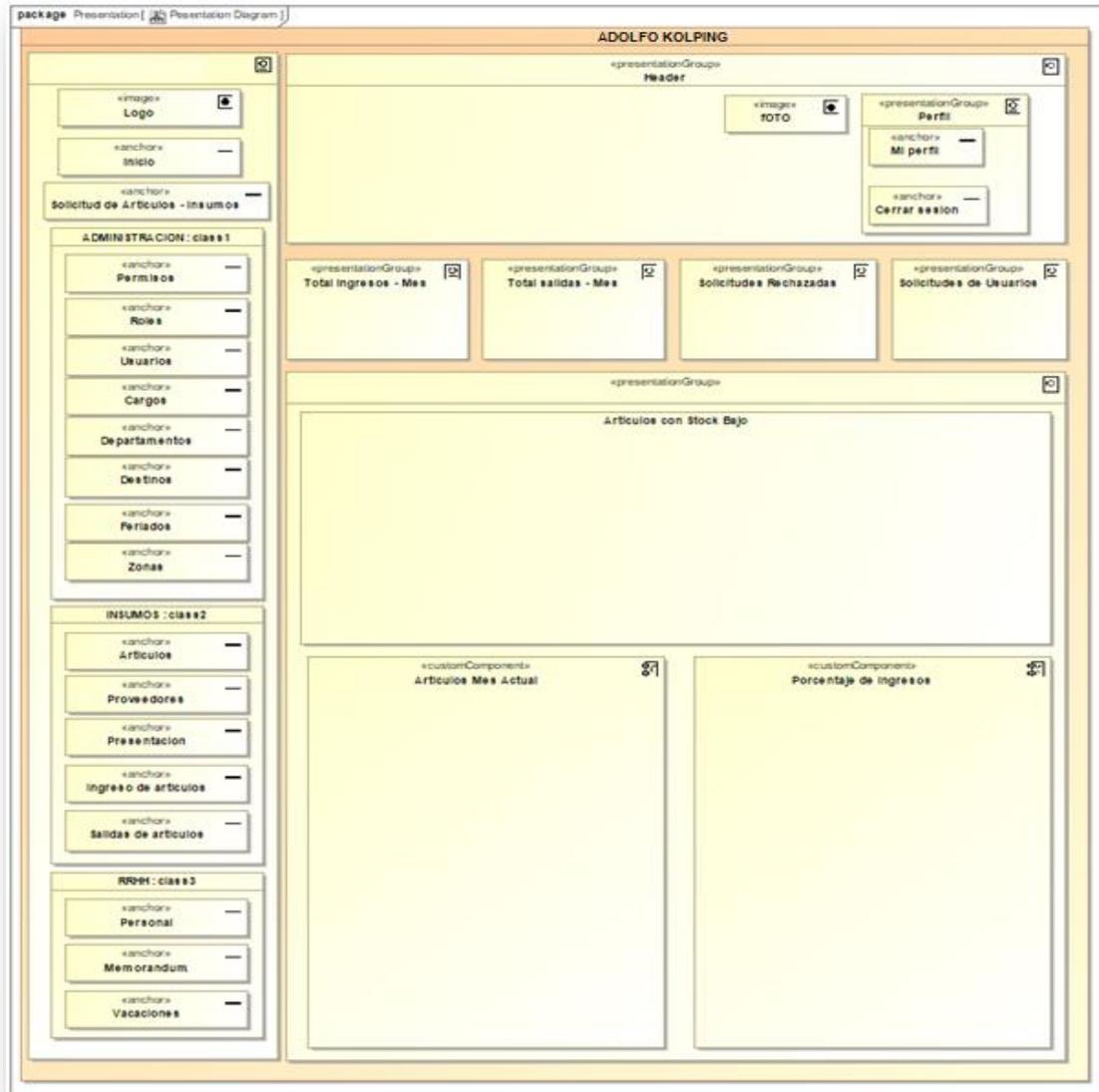
Diagrama de Navegación inicio de sesión



3.5.2. Modelo de Presentación: Página de inicio

Figura 31

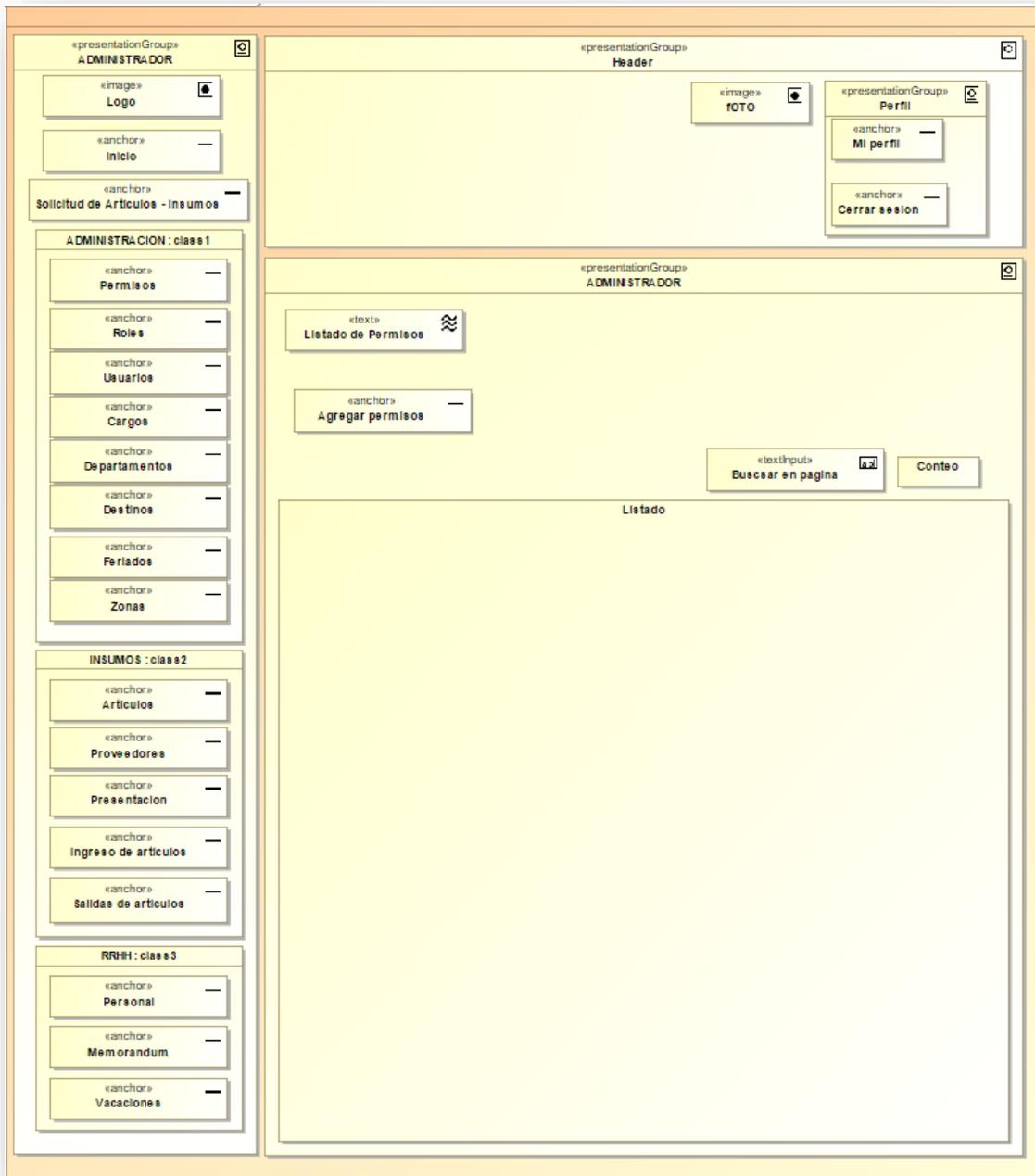
Diagrama de Navegación Menú principal



3.5.3. Modelo de Presentación: Administrador

Figura 32

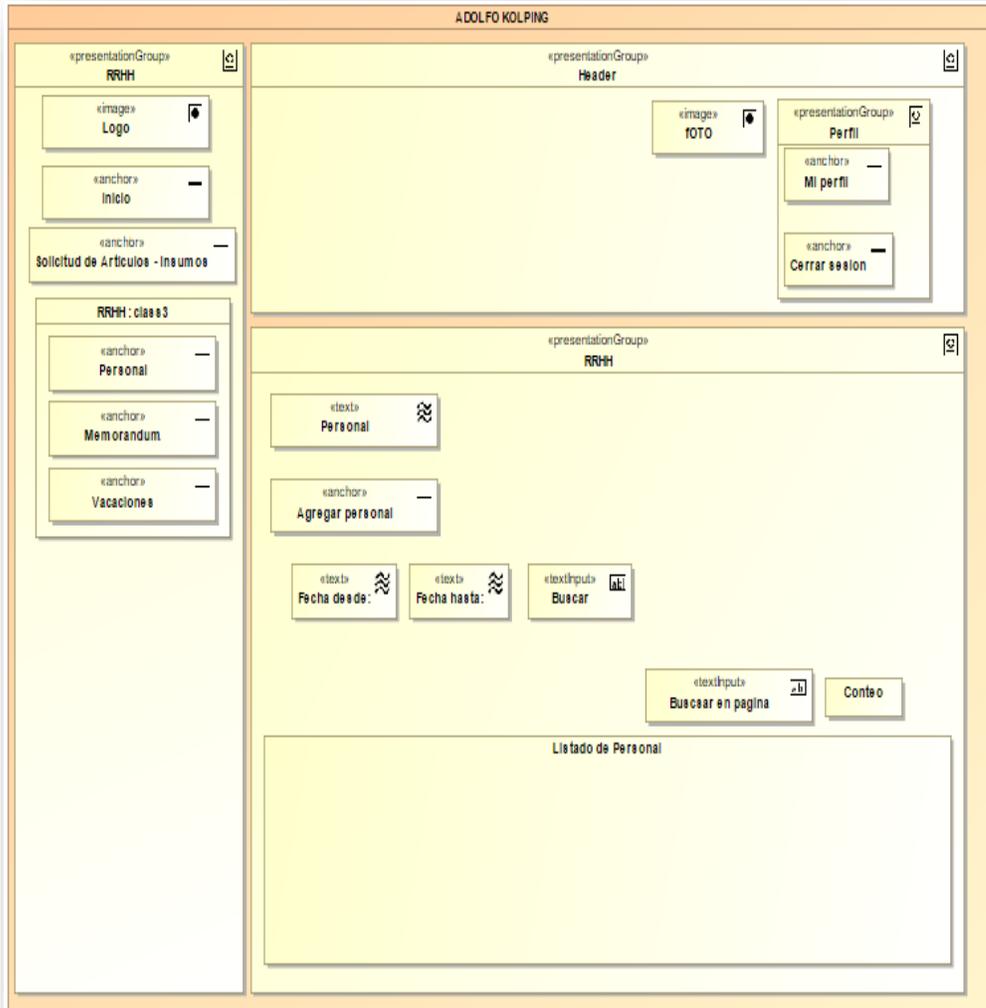
Diagrama de presentación de administrador



3.5.4. Modelo de Presentación: rrhh

Figura 33

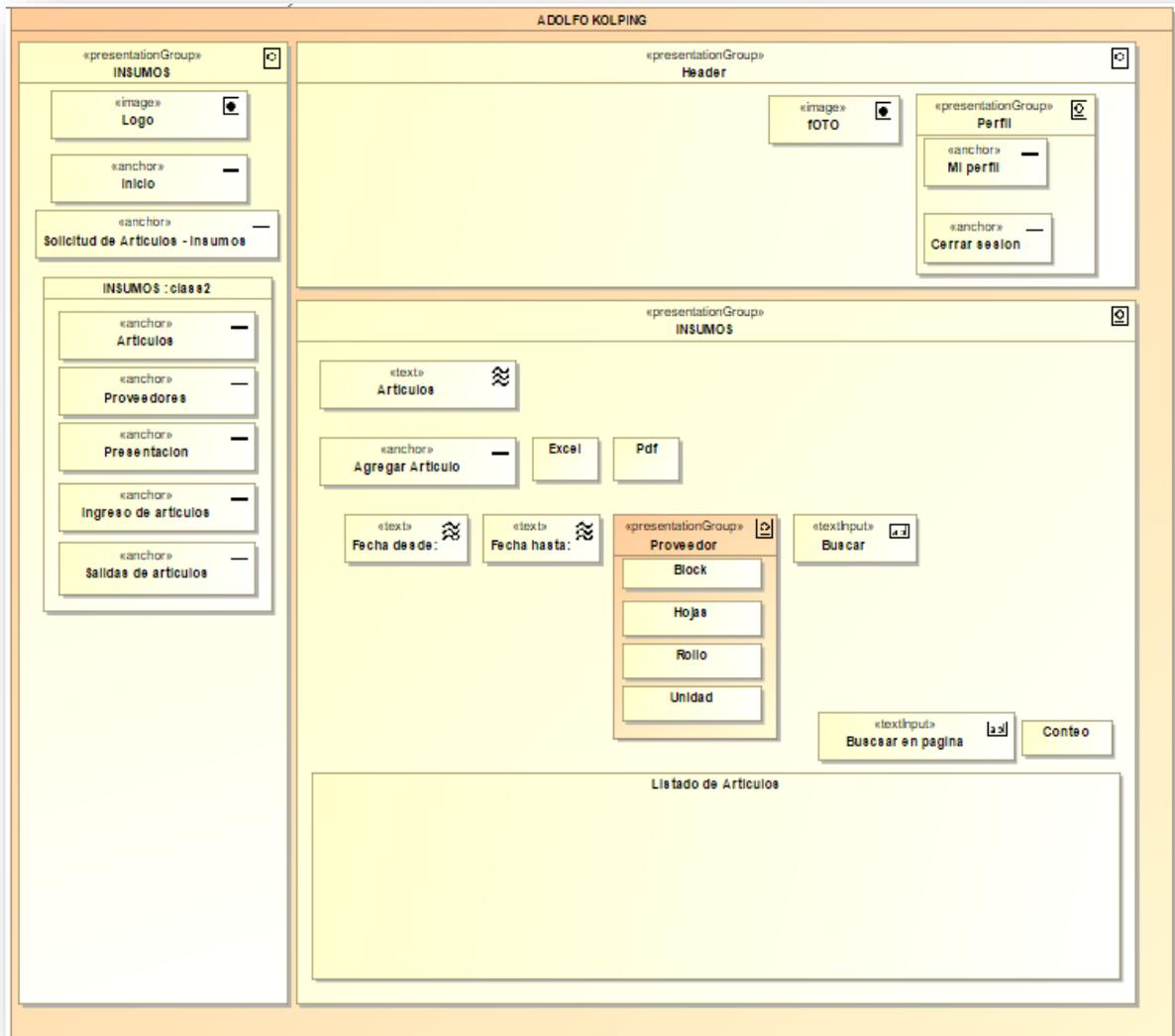
Diagrama de presentación de recursos humanos



3.5.5. Modelo de Presentación: Encargado de insumos

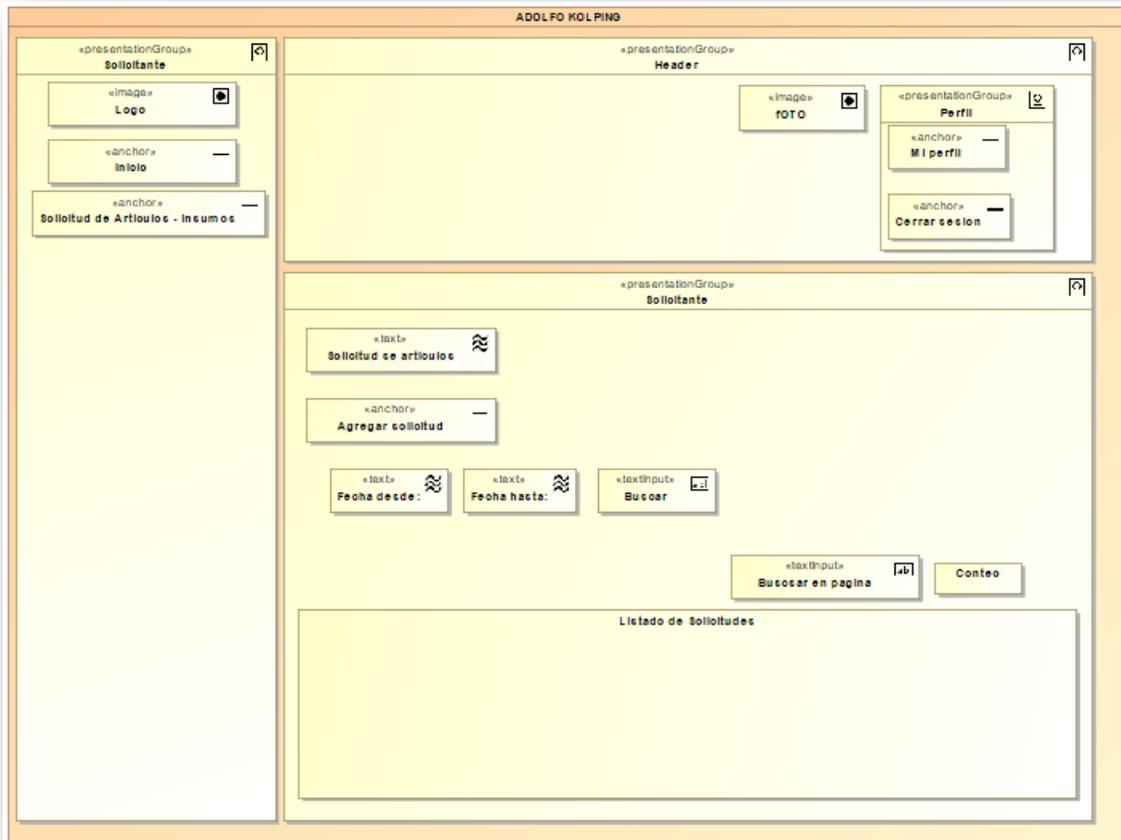
Figura 34

Diagrama de presentación de encargado de insumos



3.5.6. Modelo de Presentación: Solicitante

Figura 35
Diagrama de presentación del solicitante



3.7. Implementación

Sistema Integral Para el Control y Administración de Insumos Médicos y Personal Administrativo

Clínica de especialidades “ADOLFO KOLPING”

Figura 40

Diagrama de Navegación inicio de sesión



3.8. Interfaz Inicio de Sesión

Realizar la Autenticación de cada usuario para poder ingresar al módulo correspondiente que se les fue asignado.

Se deberá ingresar con una cuenta que fue asignada por el administrador indicando su Usuario y Contraseña.

3.8.2. Módulos que integran el sistema

3.8.2.1. módulo de administración

Figura 43
Registro de usuarios

Registro de Usuario | [Volver](#)

Nombre (*)
Escriba aquí

Apellidos (*)
Escriba aquí

Email (*)
Escriba aquí

Password (*)
Escriba aquí

Celular
Escriba aquí

Rol de usuario (*)

Super Administrador Administrador Recursos Humanos Responsable de Insumos/Artículos

Solicitante de Insumos/Artículos

Registrar

Figura 44
Listado de usuarios

Listado de Usuarios (9)

+ Agregar Usuario

Fecha desde: 01/01/2023 Fecha hasta: 05/11/2023 **Buscar**

Buscar en página: Buscar en página 10

#	NOMBRE	APELLIDOS	EMAIL	CELULAR	ESTADO	ÚLTIMA SESIÓN	EDITAR	ELIMINAR
1	Juan José	Perez	jperez@kolping.com.bo	70000000	Normal		Editar	Eliminar
2	Lourdes Florencia	Flores Achu	lflores@kolping.com.bo		Normal		Editar	Eliminar
3	Giovana Lucy	Poma Espejo	g.poma@kolping.com.bo		Normal		Editar	Eliminar
4	Patricia	Salas	p.salas@kolping.com.bo		Normal		Editar	Eliminar
5	Lourdes	Cabrera Colque	lcabrera@kolping.com.bo	76282881	Normal		Editar	Eliminar
6	Edwin Cristian	Rivas	e.rivas@kolping.com.bo	70119908	Normal		Editar	Eliminar
7	Ronald	Rios Mamani	r.rios@kolping.com.bo	72344517	Normal		Editar	Eliminar
8	Swanía	Guarachi Velasco	vani.bet@hotmail.com	79551771	Normal		Editar	Eliminar
9	Admin	Lebet	info@lebet.com		Normal		Editar	Eliminar

Figura 45
Perfil de usuario

Figura 46
Listado de cargos

#	NOMBRE CARGO	DEPARTAMENTO	DESCRIPCIÓN	SALARIO ESTIMADO	EDITAR	ELIMINAR
1	ANESTESIOLOGIA	MEDICO			Editar	Eliminar
2	ASISTENTE ADMINISTRATIVO(A)	ADMINISTRATIVO			Editar	Eliminar
3	CAJERO(A)	ADMINISTRATIVO			Editar	Eliminar
4	CARDIOLOGIA	MEDICO			Editar	Eliminar
5	CIRUGIA GENERAL	MEDICO			Editar	Eliminar
6	CIRUGIA PEDIATRICA	MEDICO			Editar	Eliminar
7	CIRUGIA PEDIATRICA	MEDICO			Editar	Eliminar
8	CIRUGIA PLÁSTICA	MEDICO			Editar	Eliminar
9	COLOPROCTOLOGIA	MEDICO	PROCTOLOGO		Editar	Eliminar
10	DIRECTOR GENERAL	DIRECCION	DR.		Editar	Eliminar

3.8.2.2. módulo insumos

Figura 47
Listado de artículos

CLÍNICA KOLPING

Inicio
Solicitudes de Artículos - Insumos

INSUMOS

- Artículos
- Proveedores
- Presentación
- Ingreso de artículos
- Salida de artículos

Patricia Salas
Responsable de Insumos/Artículos

Listado de Artículos (198)

+ Agregar Artículo | Excel | PDF

Fecha desde: 01/01/2023 | Fecha hasta: 05/12/2023 | Presentación: --Todos-- | Buscar

Buscar en página: Buscar en página | 10

#	ARTÍCULO	PRESENTACIÓN	DETALLE	CANTIDAD MÍNIMA	CANTIDAD EXISTENTE	MOVIMIENTO ARTÍCULO	EDITAR	ELIMINAR
1	ADMINISTRACION DE MEDICAMENTOS	BLOCK	FORMULARIO	2	0	Artículo	Editar	Eliminar
2	AIRLIFT MISTIC	LITROS	LIQUIDO PARA LIMPIEZA	5	22	Artículo	Editar	Eliminar
3	ALCOHOL ANTIBACTERIAL SCOTT	UNIDAD	DE USO PARA DISPENSADORES	3	83	Artículo	Editar	Eliminar
4	AMBIENTADOR POETT 291 GR	UNIDAD	INDUSTRIA ARGENTINA	5	90	Artículo	Editar	Eliminar
5	AQUANET PARA CABELLO 397 GR	UNIDAD	DE USO PARA ENFERMERIA	2	2	Artículo	Editar	Eliminar
6	ARCHIVADOR DE PALANCA GRANDE	UNIDAD		1	2	Artículo	Editar	Eliminar
7	ARCHIVADOR DE PALANCA PEQUEÑO	UNIDAD		1	10	Artículo	Editar	Eliminar
8	ASAP 20	LITROS	DE USO PARA LIMPIEZA	2	40	Artículo	Editar	Eliminar
9	BARBIJO KN95	UNIDAD	DESECHABLE	15	36	Artículo	Editar	Eliminar
10	BARBIJOS DESECHABLES	UNIDAD		5	8	Artículo	Editar	Eliminar

Mostrando 1 a 10 de 198 Resultados

< 1 2 3 4 5 6 7 8 9 10 ... 19 20 >

Figura 48
Alerta para eliminar registro de artículo/insumo

CLÍNICA KOLPING

Inicio
Solicitudes de Artículos - Insumos

INSUMOS

- Artículos
- Proveedores
- Presentación
- Ingreso de artículos
- Salida de artículos

Patricia Salas
Responsable de Insumos/Artículos

Listado de Artículos (24)

+ Agregar Artículo | Excel | PDF

Fecha desde: 01/01/2023 | Fecha hasta: 06/11/2023 | Presentación: --Todos-- | Buscar

Buscar en página: Buscar en página | 10

#	ARTÍCULO	PRESENTACIÓN	DETALLE	CANTIDAD MÍNIMA	CANTIDAD EXISTENTE	EDITAR	ELIMINAR
1	AMBIENTADOR POETT 291 GR			5	91	Editar	Eliminar
2	BARBIJO KN95			15	31	Editar	Eliminar
3	BIRUTA TRAMONTINA PEQUEÑA			1	149	Editar	Eliminar
4	CEPILLOS DE BAÑO			1	8	Editar	Eliminar
5	CERA TIGRE SACHET PEQUEÑO			1	93	Editar	Eliminar
6	ESPONJAS PEGUERAS	Unidad	DE USO PARA LIMPIEZA	5	179	Editar	Eliminar
7	GOMA DE PISO ARGENTINO	Unidad	DE USO PARA LIMPIEZA	2	8	Editar	Eliminar
8	GOMA DE PISO NACIONAL	Unidad	DE USO PARA LIMPIEZA	2	12	Editar	Eliminar
9	GUANTES DE GOMA TALLA 7	Par	DE USO PARA LIMPIEZA	3	0	Editar	Eliminar
10	GUANTES DE GOMA TALLA 7 1/2	Par	DE USO PARA LIMPIEZA	3	13	Editar	Eliminar

Mostrando 1 a 10 de 24 Resultados

< 1 2 3 >

¿Está seguro de que desea eliminar este registro?

Esta operación es irreversible

Cancelar Confirmar

Figura 49
Reporte movimiento articulo

		
MOVIMIENTO DEL ARTÍCULO		
ARTÍCULO: ADMINISTRACION DE MEDICAMENTOS		ID: 38
SALDO ANTERIOR: 0.00		
ENTRADAS:		
FECHA:	DETALLE:	CANTIDAD:
01/10/2020		8
29/12/2020		10
25/08/2022		4
INGRESOS: 22		
SALIDAS:		
ADMINISTRACION:		
FECHA:	RESPONSABLE:	CANTIDAD:
24/04/2023	Juan José Perez	1
SUBTOTAL: 1		
ENFERMERIA:		
FECHA:	RESPONSABLE:	CANTIDAD:
14/12/2020	Dasia Emser Beatty	1
23/11/2020	Dasia Emser Beatty	1
06/06/2022	Juan José Perez	1
27/06/2022	Juan José Perez	1
11/07/2022	Juan José Perez	1
25/07/2022	Juan José Perez	1
01/08/2022	Juan José Perez	1
15/08/2022	Juan José Perez	1
19/10/2022	Juan José Perez	1
24/10/2022	Juan José Perez	1
14/11/2022	Juan José Perez	1
19/12/2022	Juan José Perez	1
03/01/2023	Juan José Perez	1
12/12/2022	Juan José Perez	1
16/01/2023	Juan José Perez	1
30/01/2023	Juan José Perez	1
13/02/2023	Juan José Perez	1
06/03/2023	Juan José Perez	1
10/04/2023	Juan José Perez	1
08/05/2023	Juan José Perez	1
SUBTOTAL: 20		

Figura 51
Registro de ingreso

Registro de ingreso | 

Usuario (*) **Fecha (*)**

Proveedor (*) **Observaciones**

Detalle de ingreso de Artículos

#	ARTÍCULO	CANTIDAD	PRECIO UNITARIO	OBSERVACIÓN	X
1	<input type="text" value="--Todos--"/>	<input type="text" value="Escriba aquí"/>	<input type="text" value="Escriba aquí"/>	<input type="text" value="Escriba aquí"/>	<input type="text" value="X"/>

[+ Agregar Artículo](#)

Figura 50
Salidas de artículos

CLÍNICA KOLPING

Inicio | Solicitudes de Artículos - Insumos

INSUMOS

- Artículos
- Proveedores
- Presentación
- Ingreso de artículos
- Salida de artículos**

Patricia Salas
Responsable de Insumos y Medicamentos

Listado de Salidas (290)

Fecha desde: Fecha hasta: Destino:

Buscar en página: 10

#	DESTINO	USUARIO	FECHA	CANTIDAD ARTÍCULOS	OBSERVACION	VER
1	ENFERMERIA	Juan José Perez	15/05/2023	1		<input type="button" value="Ver"/>
2	ENFERMERIA	Juan José Perez	10/04/2023	1		<input type="button" value="Ver"/>
3	ENFERMERIA	Juan José Perez	13/03/2023	1		<input type="button" value="Ver"/>
4	ENFERMERIA	Juan José Perez	13/02/2023	1		<input type="button" value="Ver"/>
5	QUIROFANO	Juan José Perez	15/05/2023	1		<input type="button" value="Ver"/>
6	QUIROFANO	Juan José Perez	13/02/2023	1		<input type="button" value="Ver"/>
7	QUIROFANO	Juan José Perez	09/01/2023	1		<input type="button" value="Ver"/>
8	ENFERMERIA	Juan José Perez	15/05/2023	1		<input type="button" value="Ver"/>
9	ENFERMERIA	Juan José Perez	06/02/2023	1		<input type="button" value="Ver"/>
10	ENFERMERIA	Juan José Perez	22/05/2023	1		<input type="button" value="Ver"/>

Mostrando 1 a 10 de 290 Resultados

3.8.2.3. módulo recursos humanos

Figura 52
Listado del personal

Listado de Personal (166)

Fecha desde: dd/mm/aaaa Fecha hasta: dd/mm/aaaa **Buscar**

Buscar en página: 10

#	NÚMERO C.I.	NOMBRES	PRIMER APELLIDO	SEGUNDO APELLIDO	EMAIL	CELULAR	CARGO ACTUAL	ESTADO CIVIL	CIUDAD	ZONA	NÚMERO DE HIJOS	KARDEX PERSONAL	INGRESO	ELIMINAR
1	9937714	Serania	Guarachi	Velasco	swaniaguarachi@gmail.com	79551771		soltero-a	El Alto	Ciudad Satélite	0	Kardex	Ingreso	Eliminar
2	3780659	Evans	Zboncak	Graham	bcris@esample.com	77290109	ENDOCRINOLOGIA	casado-a	El Alto	Urbanización Libertal	0	Kardex	Ingreso	Eliminar
3	73934169	Sylvia	Batz	Gerhold	tnikolaus@esample.net	63185686	HEMATOLOGIA	soltero-a	El Alto	Nuevos Horizontes	4	Kardex	Ingreso	Eliminar
4	83448471	Henriette	Zemlak	Crona	tatyana39@esample.net	7497651	ODONTOLOGIA	otro	El Alto	Santiago I	1	Kardex	Ingreso	Eliminar
5	61542375	Dante	Crist	Pacocha	rosalyn44@esample.net	75704262	NEFROLOGIA	union libre	El Alto	Complejo	3	Kardex	Ingreso	Eliminar
6	38214211	Rickie	Rippin	Wyman	koopp.felix@esample.net	63417108	LIMPIEZA/LAVANDERÍA/COCINA	otro	El Alto	Caluyo	3	Kardex	Ingreso	Eliminar
7	41538093	Elaina	Berge	Murazik	edaniel@esample.com	74526561	CARDIOLOGIA	union libre	El Alto	Nuevos Horizontes	5	Kardex	Ingreso	Eliminar
8	78155302	Stella	Lemke	Runofsson	lemuel85@esample.com	62082134	PSIQUIATRIA	viudo-a	El Alto	Río Seco	3	Kardex	Ingreso	Eliminar
9	95343061	Dino	Schuppe	Runofsdottir	llewellyn.cruickshank@esample.net	72224339	MEDICO GENERAL	divorciado-a	El Alto	German Busch	1	Kardex	Ingreso	Eliminar
10	73975487	Kaitlin	Beier	Bahringer	kerluke.hilton@esample.com	68446101	NEUROCIURGÍA	separado-a	El Alto	Villa Alemania	3	Kardex	Ingreso	Eliminar

Mostrando 1 a 10 de 166 Resultados

Figura 53
Registro del personal

Registro del personal | [Volver](#)

- Personal
Datos de la persona
- Pre Grado
Datos educación escolar
- Post Grado
Datos educación superior
- Información Laboral
Referencias laborales
- Referencias Personales

Información Personal

Numero CI (*)
Escriba aquí Exp

Primer Apellido (*)
Escriba aquí

Fecha de Nacimiento (*)
dd/mm/aaaa

Numero de hijos (*) **Sexo (*)**

Nacionalidad (*)
Escriba aquí

Direccion (*)
Escriba aquí

Celular (*)
Escriba aquí

Foto
La imagen debe ser JPG, PNG, GIF o SVG.

Siguiente

Nombres (*)
Escriba aquí

Segundo Apellido
Escriba aquí

Estado Civil (*)
Seleccione

Lugar de Nacimiento (*)
Escriba aquí

Ciudad (*) **Zona (*)**

Telefono
Escriba aquí

Email (*)
Escriba aquí

Figura 3. 1

Figura 54
Registro de ingreso del personal

Registro de Ingreso | [Volver](#)

Personal
Elaina Berge Murazik

Proyecto (*)
CLINICA DE ESPECIALIDADES ADOLFO KOLPING

Fecha Ingreso(*)
dd/mm/aaaa

Estado (*)
Seleccione

Cargo (*)
Seleccione

Observacion (*)
Escriba aquí

Registrar

Listado de Registro de Ingresos (1) | [Volver](#)

+ Agregar Ingreso

Buscar en página: 10

#	PERSONAL	PROYECTO	CARGO	FECHA INGRESO	ESTADO	FECHA RETIRO
1	Elaina Berge Murazik	CLINICA DE ESPECIALIDADES ADOLFO KOLPING	CARDIOLOGIA	02/12/2022	Activo	

3.8.2.4. módulo solicitante

Figura 56
Listado de solicitudes

Registro de solicitud | Volver

Usuario (*) Juan José Pérez Fecha Solicitud(*) 14/11/2023

Destino (*) --Todos-- Detalle Escribe aquí

Detalle de Solicitud de Artículos

#	ARTÍCULO	CANTIDAD	X
1	--Todos--	Escribe aquí	X

+ Agregar Artículo

Registrar

Figura 55
Registro de solicitud

Registro de solicitud | Volver

Usuario (*) Lourdes Cabrera Colque Fecha Solicitud(*) 05/12/2023

Destino (*) --Todos-- Detalle Escribe aquí

Detalle de Solicitud de Artículos

#	ARTÍCULO	CANTIDAD	X
1	--Todos--	Escribe aquí	X

+ Agregar Artículo

Registrar

Figura 57
Reporte Kardex



FUNDACIÓN PADRE ADOLFO KOLPING

No.- 159

KARDEX DEL PERSONAL

PROYECTO: CLINICA DE ESPECIALIDADES ADOLFO KOLPING
PUESTO: PSIQUIATRIA
FECHA DE INGRESO: 21/11/2022

1. INFORMACIÓN PERSONAL

APELLIDO PATERNO: Lemke	SEXO: No Especifica
APELLIDO MATERNO:	CI: 78155302
NOMBRES: Stella	NACIONALIDAD: Armenia
LUGAR DE NACIMIENTO: West Norbert	FECHA DE NACIMIENTO: 17/02/2007
ESTADO CIVIL: viudo-a	No. DE HIJOS: 3
DIRECCIÓN: 28070 Dariana Lakes Suite 189 Kozeyburgh, RI 77200	CIUDAD: El Alto
ZONA: Río Seco	CELULAR: 62082134
EMAIL: lemuel85@example.com	TELÉFONO DOMICILIO: 2247415

2. INFORMACIÓN ACADÉMICA

Educación Escolar

CENTRO ACADÉMICO: MARIA AUXILIADORA	GESTIÓN: 2007
CIUDAD: La Paz	ÚLTIMO CURSO: BACHILLERATO

Educación Universitaria Técnica

CENTRO ACADÉMICO: NUESTRA SEÑORA DE LA PAZ	FECHA TÍTULO: 27/06/2015
CARRERA: MEDICO CIRUJANO	
TÍTULO OBTENIDO: egresado	
CENTRO ACADÉMICO: INSTITUTO NACIONAL DE PSIQUIATRIA GREGORIO PACHECO SUCRE	FECHA TÍTULO: 01/01/2020
CARRERA: MEDICO PSIQUIATRA	
TÍTULO OBTENIDO: egresado	

Postgrados o Cursos Complementarios

CENTRO ACADÉMICO: UNIVERSIDAD MAYOR REAL Y PONTIDICIA SAN FRANCISCO X	FECHA TÍTULO: 01/01/2021
AREA DE FORMACION: ESPECIALIDAD EN PSIQUIATRIA FORENCE	
TÍTULO OBTENIDO: ESPECIALIDAD	
LUGAR: Sucre	

Registro Profesional

CAPITULO IV

4. CALIDAD DEL SOFTWARE, SEGURIDAD Y ESTIMACIÓN DE COSTOS

4.1. Métricas de Calidad del Software Mediante la Norma ISO/IEC 9126

4.1.1. Funcionalidad

La funcionalidad se representa mediante la medida indirecta del software, se toma la métrica de punto función, el cual se usa como medio para medir la funcionabilidad. Para la funcionabilidad o medición del sistema, Los valores de los dominios de información se definen de la forma siguiente:

Tabla 10
Características de Funcionalidad

Características	Descripción
Número de entradas de usuario	Corresponde a cada interacción en la que el usuario proporciona datos específicos al software.
Número de salida de usuario	Se refiere a cada resultado que el software presenta al usuario, incluyendo informes, pantallas, mensajes de error, entre otros.
Número de peticiones de usuario	En este caso, se consideran las interacciones interactivas en las que el software responde inmediatamente con una salida interactiva.
Número de archivos	Se refiere a un conjunto lógico de datos, que puede formar parte de una base de datos más amplia o existir como un archivo independiente.

Número de interfaces externas

Engloba todas las interfaces que el software utiliza para interactuar con dispositivos legibles por computadora y transmitir información.

En base a lo mencionado se tiene los siguientes datos.

Tabla 11

Parámetros para la medición

Parámetro de medición	Cuenta
Número de entradas de usuario	113
Número de salida de usuario	55
Número de peticiones de usuario	38
Número de archivos	15
Número de interfaces externas	1

Nota: *Elaboración Propia*

Para realizar el cálculo de la cuenta total con los factores de ponderación se considera que todas las funciones identificadas serán de complejidad media.

Tabla 12

Cálculo de puntos de función sin ajustar

Parámetro de medición	Conteo	Baja	Media	Alta	Total
Número de entradas de usuario	113	3	4	6	452
Número de salida de usuario	55	4	5	7	275
Número de peticiones de usuario	38	3	4	6	152

Número de archivos	15	7	10	15	150
Número de interfaces externas	1	5	7	10	7
Cuenta Total					1036

Para determinar los valores de factor de ajuste se indica según corresponda a las siguientes preguntas:

Tabla 13
Parámetros de medición (Factores de Ponderación)

Escala	Sin importancia	Incidental	Moderado	Medio	Significativo	Esencial	Fi
	0	1	2	3	4	5	
Factor							
1. ¿El sistema requiere respaldo y recuperación confiables?					X		4
2. ¿Se requieren comunicaciones de datos?				X			4
3. ¿Existen funciones de procesamiento distribuidas?				X			3
4. ¿El desempeño es crucial?		X					2
5. ¿Se ejecuta el sistema en un entorno operativo existente enormemente utilizado?					X		4

6. ¿El sistema requiere entrada de datos en línea?	X	5
7. ¿La entrada de datos en línea requiere que la transacción de entrada se construya sobre múltiples pantallas u operaciones?	X	3
8. ¿Se actualizan los archivos maestros en línea?	X	3
9. ¿Las entradas, salidas, archivos o consultas son complejos?	X	3
10. ¿El procesamiento interno es complejo?	X	4
11. ¿El código se diseña para ser reutilizable?	X	3
12. ¿La conversión y la instalación se incluyen en el diseño?	X	2
13. ¿Se ha diseñado el sistema para Soportar múltiples instalaciones en diferentes organizaciones?	X	3
14. ¿La aplicación se diseña para facilitar el cambio y su uso por parte del usuario?	X	5
TOTAL Fi		48

A continuación, calculamos los puntos de función mediante la siguiente ecuación.

$$PF = \text{cuenta total} * (0.65 + 0.01 * \Sigma Fi)$$

Reemplazando en la ecuación se tiene el siguiente resultado.

$$PF = 1036 * (0.65 + 0.01 * 48)$$

$$PF = 1170.68$$

Considerando el factor de ajuste el PF máximo que se puede alcanzar, calculamos al 100% el nivel de confianza.

$$PF_{max} = \text{cuenta total} * (0.65 + 0.01 * \Sigma Fi)$$

$$PF_{max} = 1036 * (0.65 + 0.01 * 70)$$

$$PF_{max} = 1398.6$$

Entonces la funcionalidad del software está dada de la siguiente manera:

$$\text{Funcionalidad} = PF/PF_{max}$$

$$\text{Funcionalidad} = 1170.68/1398.6$$

$$\text{Funcionalidad} = 0.84$$

$$\text{Funcionalidad} = 0.84 * 100 = 84\%$$

Por lo tanto, se concluye que la funcionalidad del sistema es del 84%, esto significa que el sistema tiene un 84% de funcionar sin fallar en la operatividad y un 16% de colapsar el sistema.

4.1.2. Confiabilidad

Para calcular la confiabilidad del sistema se toma en cuenta el periodo de tiempo en el cual se ejecuta y obtiene las muestras respectivas.

$$F(t) = f * e(-\mu * t)$$

La probabilidad en el que el sistema posiblemente tenga fallas está dada por:

$$P(T \leq t) = F(t)$$

La probabilidad en el que no falla es sistema está dado por:

$$P(T \leq t) = 1 - F(t)$$

Para esto consideramos un periodo de 20 días como tiempo de prueba donde se aprecia que cada 10 ejecuciones se presentan alguna falla

Conociendo la funcionalidad de $f = 84\%$ del sistema calculamos para el periodo establecido.

$$F(t) = f * e(-\mu * t)$$

$$F(t) = 0.84 * e(-1/10 * 20)$$

$$F(t) = 0.113 * 100 = 11.3\%$$

Por lo tanto, tenemos las siguientes probabilidades:

$$P(T \leq t) = F(t) = 0.113 * 100 = 11.3\%$$

$$P(T \leq t) = 1 - F(t) = 1 - 0.113 = 0.887 = 88.7\%$$

Por lo tanto, el sistema tiene un grado de confiabilidad del 88.6% en un periodo de prueba de 20 días.

4.1.3. Usabilidad

Para determinar la usabilidad del sistema se utiliza la siguiente ecuación:

$$FU = \Sigma(x_i/n) * 100$$

Dónde: x_i es la sumatoria de valores y n es el número de preguntas, la escala de valoración de las preguntas se da entre 1 y 5.

Tabla 14
Ajuste de preguntas para determinar la usabilidad

n	Preguntas	Respuestas		Valores (x_i)
		Si	No	
1	¿Es suficiente la información que se ofrece en pantalla para saber a qué institución corresponde el sitio?	5	0	1
2	¿Le parece adecuada la selección de contenidos presentes en el menú principal?	5	0	1
3	Al hacer click en un contenido del menú ¿halló en la información ofrecida lo que esperaba encontrar?	4	1	0.8
4	¿los nombres de los enlaces, son suficientemente descriptivos, hacia las cuales llevan?	5	0	1
5	¿cree que la navegación interna del sitio le permite explorarlo adecuadamente?	4	0	0.8
6	¿Se logran distinguir las zonas clickeables de las no clickeables?	5	0	1
7	Los íconos presentes en el sitio, ¿le parecen lo suficientemente legibles, reconocibles, y explicativos?	5	0	1

8	¿le parece optimo el tiempo de espera al momento de generar algún reporte?	4	1	0.8
9	¿puede utilizar el sistema con facilidad?	5	0	1
10	¿le parece que el software proporciona la seguridad e integridad adecuada?	5	0	1
11	¿durante el uso del sistema no se produjeron errores?	4	1	0.8
Total				10.2

Procedemos a calcular la usabilidad con la ecuación planteada.

$$FU = \Sigma(x_i/n) * 100$$

$$FU = 10.2 / 11 * 100$$

$$FU = 92.73 * 100$$

$$FU = 92.73\%$$

Por lo tanto, se deduce que el sistema tiene un grado de usabilidad del 92.73%

4.1.4. Mantenibilidad

Para el cálculo de mantenibilidad se utiliza el IMS índice de madurez del software. Por lo que el IMS se determina con la siguiente ecuación.

$$IMS = \frac{[Mt - (fa + fc + fd)]}{Mt}$$

Donde:

Mt = Numero de módulos de la versión actual = 5

Fa = Numero de módulos en la versión actual que se han añadido = 1

Fc = Numero de módulos en la versión actual que se han modificado = 0

Fd = Número de módulos de la anterior versión que se han eliminado en la versión actual. = 0

Calculando:

$$IMS = \frac{[5 - (0 + 1 + 0)]}{5}$$

$$IMS = \frac{4}{5}$$

$$IMS = 0.8 * 100 = 80\%$$

Por lo tanto, se puede decir que el sistema tiene un grado de madurez del 80% que es la facilidad de mantenimiento.

4.1.5. Portabilidad

La portabilidad se puede expresar como:

Grado de portabilidad GP= 1 - (ET/ER)

donde:

ET: medida de los recursos necesarios para mover el sistema a otro entorno (Target Environment).

ER: medida de los recursos necesarios para crear el sistema en el entorno residente (Resident Environment)

Donde:

Sí GP > 0 se concluye que la portabilidad es más aceptable que el redesarrollo

Sí GP = 1 se concluye que la portabilidad es perfecta

Sí GP < 0 se concluye que el desarrollo es más aceptable que la portabilidad

GP= 1 - (50/500)

GP= 0.9

Por lo tanto, el sistema puede transportarse de un entorno a otro en un 90%.

4.2. Estimación de Software con COSMIC

- **Estrategia de medición**

Lo determinamos identificando cuales son los requerimientos funcionales a medir, de acuerdo a lo planteado en el apartado 3.2.1 del capítulo 3 contamos con 54 requerimientos funcionales

- **Mapeo y medición**

En el método COSMIC, se utiliza la ingeniería de software de nuestro proyecto para determinar cuáles son los procesos funcionales y movimientos de datos que lo componen.

De esta forma, hemos determinado que nuestro proyecto tiene una medición de:

60 puntos de función COSMIC (54 CFP).

- **Costos del equipo de trabajo de desarrollo de software**

para determinar el desarrollo de una unidad de medida del tamaño del software, necesitamos valernos de la información de proyectos pasados que tengan la organización. También podemos usar información de otras fuentes, otras organizaciones.

En el caso de este sistema es una propuesta que aportara de mucho a la clínica, supongamos que tenemos un equipo de desarrollo de software y sabemos que su costo mensual es de 4000bs.

- **Determinar el costo por unidad de medida**

Para determinar cuánto cuesta desarrollar cada punto de función se utiliza la siguiente formula:

$$\text{Costo por punto de funcion} = \frac{\text{Costo mes del equipo de trabajo}}{\text{Puntos de funcion del mes}}$$

Puntos de función estimados por mes en medio año = $54/6 = 9$

$$\text{Costo por punto de función} = \frac{4000}{9}$$

$$\text{Costo por punto de función} = 444.44 \text{ Bs}$$

- **Estimación de costos de un proyecto de software**

podemos determinar el costo del proyecto de software usando la siguiente formula:

$$\text{Costo del proyecto de software} = \text{Tamaño del software} * \text{Costo por punto de funcion}$$

$$\text{Costo del proyecto de software} = 54 * 444.44 \text{ Bs}$$

$$\text{Costo del proyecto de software} = 23999.76 \text{ Bs}$$

- **Tiempo que dura el proyecto de desarrollo de software**

$$\text{Duración del proyecto} = \frac{\text{puntos de función COSMIC}}{\text{puntos de función COSMIC mes}}$$

$$\text{Duración del proyecto} = \frac{54}{9}$$

$$\text{Duración del proyecto} = 6 \text{ Meses}$$

Por lo tanto, se concluye que el proyecto de software, en este caso el sistema durará 6 meses (medio año) en desarrollarse y costará 23999.76 Veinte y tres mil novecientos noventa y nueve bolivianos y 76 centavos.

4.3. Sistema de Gestión de Seguridad de la Información ISO-27002

La ISO-27002 evalúa y rectifica la implementación mediante el cumplimiento de las normas, así como la mejora continua de un conjunto de controles que permiten reducir el riesgo de sufrir incidentes de seguridad en el funcionamiento de la institución en

cuanto a la seguridad de la información, para lo cual se tomó los siguientes tipos de seguridad:

4.3.1. Seguridad Lógica

Los respaldos o (back-up) de la base de datos del sistema se deberá realizar de acuerdo a la siguiente tabla:

Tabla 15
Copias de seguridad

DESCRIPCION	DURACION
En periodo de registro de insumos	1 vez por semana
En periodo de registro de solicitud de insumos - artículos	1 vez por semana
En periodo de registro de ingresos de artículos	1 vez por día
En periodo de control de salidas de artículos	1 vez por día
En periodo de registro de personal	1 vez por semana
En periodo de registro de usuarios	1 vez por semana
En periodo de registro historial de personal	1 vez por día

El Personal que interviene y los usuarios deberán cambiar la contraseña del sistema periódicamente 1 vez cada 20 días o 1 vez al mes.

En caso de ser el administrador del sistema se recomienda cambiar su contraseña periódicamente.

- **Identificación y autenticación:** Evita la entrada de personas no autorizadas mediante un estricto control de acceso, que requiere un usuario y una contraseña rigurosamente gestionados.

- **Encriptación:** Se utiliza una encriptación de alta seguridad (algoritmo SHA1) para proteger de manera efectiva las contraseñas, un dato crítico para acceder al sistema.

4.3.2. Seguridad Física

Se recomienda los back-up o las copias que sean almacenadas en distintos lugares. Los back-up de la base de datos deberán ser protegidas en áreas seguras, que solo permita el acceso a personal autorizado.

- **Equipamiento:** Se asegura la protección y el mantenimiento continuo de los equipos e instalaciones que componen los activos de la clínica.
- **Control de acceso físico al área de Sistemas:** Se limita la entrada a áreas críticas a personal autorizado, reduciendo así el riesgo de accesos no autorizados y posibles incidentes fraudulentos.

4.3.3. Seguridad Organizativa

La información referente al sistema debe recibir un nivel de protección apropiada como ser:

- **Gestión de archivos:** Etiquetar y manejar el back-up de acuerdo a la fecha en que se realizaron los mismos.

4.4. Pruebas de Software

Para asegurar el correcto funcionamiento del sistema, se utiliza las Técnicas de caja blanca o estructurales y Técnicas de caja negra o funcionales (técnicas de evaluación dinámica o prueba).

4.4.1. Pruebas de Caja Blanca

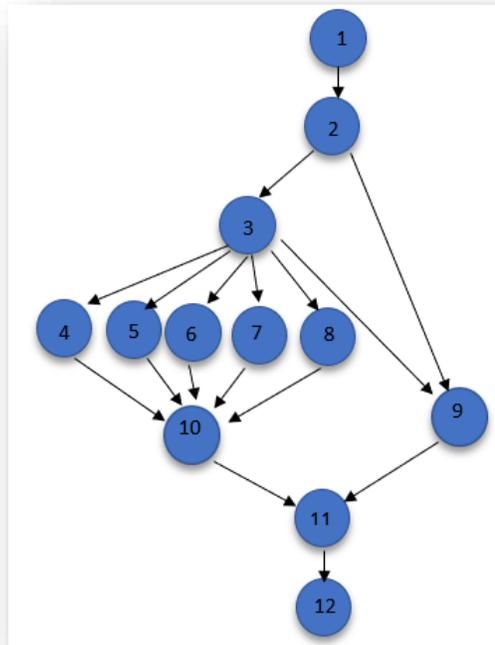
De acuerdo a lo planteado en el capítulo anterior respecto a las pruebas de caja blanca el criterio que se va a aplicar es la de cobertura de caminos.

Para aplicar este criterio de evaluación se aplica la Prueba del Camino Básico, que consiste en obtener la complejidad ciclomática $V(G)$.

- **Paso1:** Representar el programa en un grafo de flujo

Figura 58

Grafo de flujo creacion de registro



Donde:

- Inicio (1)
- Menú principal o inicio (2)
- Selecciona proveedor (3)
- Ingresa fecha (4)
- Ingresa detalles (5)
- Valida datos (6)
- Muestra error (7)
- Registra ingreso (8)
- Actualiza stock (9)
- Muestra confirmación (10)
- Fin ciclo Sistema (3)
- Fin del Sistema (12)

Analizando el grafo generado a partir de las características del sistema ahora se procede a determinar la complejidad ciclomática del grafo.

$$V(G) = A - N + 2$$

Donde se hallaron;

A = 17 (Artista)

N = 12 (Nodos)

Por lo tanto: $V(G) = 15 - 11 + 2 = 7$

Determinar el conjunto básico de caminos linealmente independientes, caminos que deben ser aprobados dadas ciertas variables son 7.

Estos caminos son los siguientes:

Camino 1: 1-2-3-5-6-10-11-12

Camino 2: 1-2-3-7-10-11-12

Camino 3: 1-2-3-8-10-11-12

Camino 4: 1-2-3-9-10-11-12

Camino 5: 1-2-4-6-10-11-12

Camino 6: 1-2-4-8-10-11-12

Camino 7: 1-2-4-9-10-11-12

Preparar los casos de prueba para forzar la ejecución de cada camino.

Camino 1: Ingresar datos válidos en el ingreso de detalles

Camino 2: Ingresar datos inválidos en el ingreso de detalles que muestren un error

Camino 3: Ingresar y registrar datos que permitan registrar el ingreso.

Camino 4: Registrar ingreso y verificar que se actualiza el stock

Camino 5: Ingresar fecha válida y pasar por validación de datos

Camino 6: Registrar ingreso con fecha válida sin pasar por validación de detalles

Camino 7: Registrar ingreso con fecha válida y verificar actualización de stock sin validación de detalles y concluye.

4.4.2. Pruebas de Caja Negra

El criterio que se va a aplicar para la evaluación de esta prueba es la de Particiones de Equivalencia y Análisis de Valores Límite.

4.4.2.1. Particiones de Equivalencia

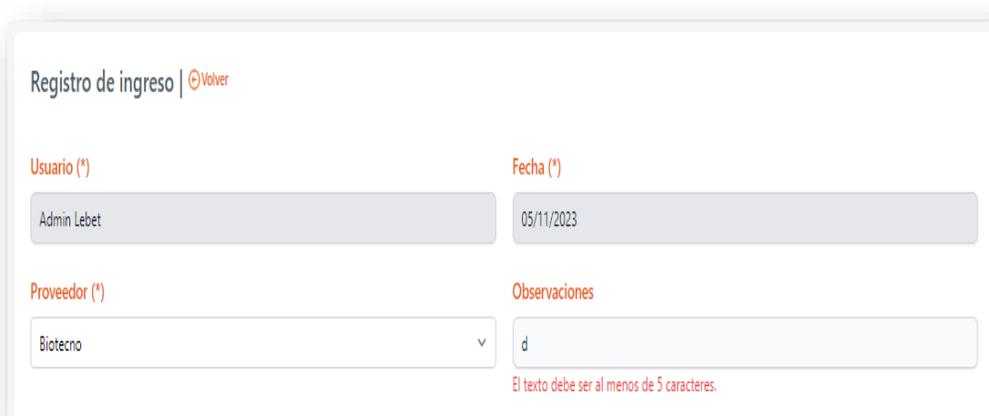
A continuación, se procede a evaluar la creación o registro de ingreso de artículos

El diseño de casos de prueba según esta técnica consta de dos pasos:

- **Paso 1:** Identificar las clases de equivalencia.

Figura 59

Prueba de caja negra de registro de ingreso



Registro de ingreso | [Volver](#)

Usuario (*)	Fecha (*)
<input type="text" value="Admin Lebet"/>	<input type="text" value="05/11/2023"/>
Proveedor (*)	Observaciones
<input type="text" value="Biotecno"/>	<input type="text" value="d"/>

El texto debe ser al menos de 5 caracteres.

Tabla 16*Identificación de las clases de equivalencia de registro de ingreso de artículos*

Entrada	Regla a aplicar	Clases validas	Clases invalidas
Proveedor	Condición booleana (¿se seleccionó?)	1. Se selecciona un proveedor	2. No se selecciona un proveedor
Observación	Rango de valores de caracteres (5-400)	3. $5 < \text{Caracteres} \leq 400$	4. ingresa menos de 5 caracteres 5. ingresa más de 400 caracteres

- **Paso 2:** Identificar los casos de prueba.

A continuación, se hace el diseño de casos de prueba para cubrir las clases de equivalencia validas (1 y 3)

Tabla 17*Caso de prueba de equivalencia valida de registro de ingreso de artículos*

Proveedor	Observación	Clases validas cubiertas
Biotechno	Equipos e insumos médicos	1, 3

Tabla 18*Caso de prueba de equivalencia invalida de registro de ingreso de articulo*

Proveedor	Observación	Clases invalidas cubiertas
-----------	-------------	----------------------------

	2, 3
Gedesa Ltda	1, 4
Trabeik	1, 5

Para verificar más datos de prueba se toman los casos de creación de Plan de Pagos las cuales se muestran y describen a continuación.

Figura 60

Prueba de caja negra de registro de artículo

Registro de artículos | [Volver](#)

Nombre (*)

 nombre debe contener al menos 2 caracteres.

Presentación (*)

Detalle (*)

 detalle debe contener al menos 3 caracteres.

Cantidad Mínima (*)

 El campo cantidad mínima es obligatorio.

Cantidad Existente (*)

 El campo cantidad existente es obligatorio.

Durante el proceso de registro de un artículo, es necesario completar todos los campos obligatorios. Si por alguna razón omitimos llenar un campo necesario, el sistema nos notificará de inmediato con un mensaje claro y preciso, indicando que el campo en cuestión es obligatorio. Esta característica garantiza que no se pasen por alto detalles importantes en el registro de artículos, lo que contribuye a mantener la integridad y la calidad de la información en nuestro sistema.

Tabla 19*Identificación de las clases de equivalencia registro de artículos*

Entrada	Regla a aplicar	Clases validas	Clases invalidas
Presentación	Conjunto de valores admitidos	1. Unidad 2. Par 3. Block 4. Hojas 5. Rollo 6. Litros 7. Kilogramos 8. Cajita 9. Paquete 10. Sobres	11. no es un valor permitido
Nombre	Rango de valores de caracteres (2-100)	12. 2<caracteres<=100	13. ingresa menos de 2 14. ingresa mas de 100
Detalle	Rango de valores de caracteres (3-255)	15. 2<caracteres<=255	16. ingresa menos de 2 17. ingresa mas de 255
Cantidad mínima	Condición booleana (¿es número?) + valor mayor a 0	18. 0<numero<=2147483647	19 no es numero
Cantidad existente	Condición booleana (¿es número?) valor mayor a 0	20. 0<numero<=2147483647	21. no es numero

4.4.3. Pruebas de estrés

Estas pruebas buscan evaluar cómo se comporta un sistema cuando se enfrenta a condiciones que superan su capacidad normal, con el fin de descubrir posibles debilidades, errores o fallos. El objetivo es determinar si la plataforma puede manejar volúmenes de usuarios concurrentes, cargas de trabajo intensas o picos repentinos de tráfico. Estas pruebas son esenciales para garantizar la robustez y el rendimiento adecuado del sistema bajo diversas circunstancias.

Para esta prueba se utilizó un servidor de los de la Universidad con algunas variaciones en el Sistema Operativo donde se obtuvieron resultados que van con el rendimiento óptimo que exigen a la plataforma

- Usuarios establecidos: 80
- Tiempo de prueba: 10 min
- Solicitudes realizadas: 998
- Fallas HTTP: 0
- Pico de solicitud: 9 (Solicitudes por segundo)

CAPITULO V

5. CONCLUSIONES Y RECOMENDACIONES

En la culminación del presente proyecto de grado, se llegaron a las siguientes conclusiones y recomendaciones que se citan a continuación:

5.1. Conclusiones

- El sistema desarrollado ha logrado con éxito integrar y coordinar la gestión de artículos médicos, mejorando significativamente la eficiencia operativa y reduciendo los tiempos de respuesta en la clínica.
- Este sistema representa una contribución valiosa al campo de la salud al abordar de manera efectiva los desafíos en la gestión de insumos médicos, garantizando un flujo más eficiente y una mejor atención a los pacientes.
- La implementación del sistema ha llevado a una disminución del 30% en el tiempo dedicado a la gestión de inventarios, permitiendo que el personal se enfoque más en la atención al paciente y en actividades clínicas.
- Se observado una mejora significativa en la calidad de atención al paciente, ya que el sistema ha reducido los errores en la administración de insumos médicos y ha facilitado el acceso rápido a información relevante del paciente."
- La seguridad y privacidad de la información médica han sido prioritarias en el diseño del sistema, con medidas de seguridad robustas y protocolos éticos para garantizar la confidencialidad de los datos del paciente.
- Para el futuro, se podrían explorar integraciones adicionales con tecnologías emergentes, como inteligencia artificial, para mejorar aún más la precisión en la gestión de inventarios y optimizar la asignación de personal.

5.2. Recomendaciones

En base a las normas de seguridad propuestas y las conclusiones realizadas durante las pruebas se elaboran las siguientes recomendaciones:

- Capacitar adecuadamente al personal que utilizará el sistema para aprovechar todas sus funcionalidades.
- Se recomienda mucha discreción en el manejo de sus usuarios y contraseñas ya que el sistema contiene información de mucha importancia.
- Realizar pruebas exhaustivas antes del despliegue en producción para detectar posibles errores o mejoras.
- Elaborar manuales de usuario detallados para cada perfil de la aplicación.
- Establecer políticas de respaldo y restauración de la base de datos para proteger la información.
- Monitorear el rendimiento y estabilidad del sistema una vez implementado para realizar mejoras oportunas.
- Ampliar la funcionalidad del sistema en el futuro desarrollando módulos adicionales según nuevos requerimientos.
- Evaluar la integración con otros sistemas existentes en la clínica para seguir optimizando procesos.

Al alcanzar todos los objetivos específicos, se logró desarrollar un Sistema Integral para el Control y Administración de Insumos Médicos y Personal Administrativo. Este sistema representa una contribución tecnológica significativa al reducir el tiempo necesario para registrar, consultar y buscar información relacionada con la clínica. Es importante destacar que la gestión de esta información se lleva a cabo de manera segura y confiable.

BIBLIOGRAFÍA

- AENOR INTERNACIONAL, S.A.U. (2016). *AENOR*. Obtenido de <https://portal.aenormas.aenor.com/revista/309/ciberseguridad2-2-309.html>
- Aguilar, M. (12 de 11 de 2020). *Solucionex*. Obtenido de <https://www.solucionex.com/blog/descubriendo-alpinejs-el-framework-liviano-alternativa-jquery>
- Alegsa, L. (2018).
- Altube, R. (31 de Marzo de 2021). *openwebinars.net*. Obtenido de Qué es Laravel: Características y ventajas: <https://openwebinars.net/blog/que-es-laravel-caracteristicas-y-ventajas/>
- Alvarez, M., Alvarez, R., Hernández, B., & Lopez, D. (30 de mayo de 2018). *Manual de PHP*. Carrizo, León, España.
- Aranda Usón, A., & Zabalza Bribian, I. (2010). *Ecodiseño y análisis de ciclo de vida*. Zaragoza. España: Prensas Universitarias de Zaragoza.
- Arias. (2019). *Implementación de un sistema integrado de suministro de medicamentos e insumos medicos para mejorar el sistema de gestion de almacen en una cadena farmaceutica*.
- Calle, G. (2020). *Portal web para el control de almacenes y activos fijos*.
- Cantone, D. (2006). *Implementacion Y Debugging: La Biblia De La Programacion*. Richmond: Creative Andina Corp.
- Carrasco, S. (2015). *Desarrollo de sistema para la gestión y control de inventario de productos ofrecidos por la Farmacia Veterinaria BioBío*. .
- Cobo, Á., Gómez, P., Pérez, D., & Rocha, R. (2005). *PHP y MySQL Tecnologías para el desarrollo de aplicaciones web*. Madrid: Ediciones Díaz de Santos.

- COSMIC. (2019). *COSMIC*. Obtenido de Método COSMIC La segunda generación en medición de tamaño funcional: <https://cosmic-sizing.org/cosmic-fsm/>
- Departamento LSI. (03 de Noviembre de 2004). *Lenguajes y Sistemas informáticos*. Obtenido de <http://www.lsi.us.es/docencia/docencia.php>
- Durán, D. (2015). *Gestion de la calidad de productos editoriales multimedia ARGN0110*. Antequera (Málaga): IC Editorial.
- Franco Díaz, J. A., Anaya Castro, L., & Valenzuela Venicio, J. (2019). *Estándares y Normas para el desarrollo de Software*. Obtenido de <https://sites.google.com/site/sistemasdecalidadenti/iso-iec-9126>
- Gallego, A. (2018). *academia.edu*. Obtenido de https://www.academia.edu/32488225/Laravel_
- Garcia, I. (2017). *Inventario*.
- Gómez Fernández, L., & Andrés Álvarez, A. (2012). *Guía de aplicación de la Norma UNE-ISO/IEC 27001 sobre seguridad en sistemas de información para pymes*. Madrid: AENOR.
- Harold Koontz, H. W. (2012). *Administración una Perspectiva Global y Empresarial*. Mexico: Karen Estrada.
- International Organization for Standardization. (Julio de 2011). *ISO 14006:2011*. Obtenido de Sistemas de gestión ambiental - Directrices para la incorporación del ecodiseño: <https://www.iso.org/obp/ui#iso:std:iso:14006:ed-1:v1:es:sec:4.2>
- Isla Visual. (19 de Mayo de 2016). *Curso de Maquetación CSS Y BOOTSTRAP 3*. Madrid, Madrid, España.
- iso. (2014). *iso*. Obtenido de Ingeniería de sistemas y software: <https://www.iso.org/obp/ui/#iso:std:iso-iec:25000:ed-2:v1:en>
- ISO, N. (2017). Obtenido de <https://www.pmg-ssi.com/2017/08/norma-iso-27002-politica-seguridad/>

Israel. (30 de 12 de 2022). *cursosdesarrolloweb*. Obtenido de <https://www.cursosdesarrolloweb.es/blog/laravel-livewire-una-nueva-forma-de-replantear-tus-desarrollos-con-laravel>

Kabir, M. (2003). *La biblia de Servidor Apache 2*. Madrid: anaya multimedia.

Laudon, K., & Laudon, J. (2016). *Sistemas de información gerencial*. México,: Pearson educación.

LMU – Ludwig-Maximilians-Universität München. (10 de 8 de 2016). <http://uwe.pst.ifi.lmu.de/index.html>. Obtenido de <http://uwe.pst.ifi.lmu.de/index.html>

LMU – Ludwig-Maximilians-Universität München. (10 de 08 de 2016). *UWE – UML-based Web Engineering*. Obtenido de <http://uwe.pst.ifi.lmu.de/exampleMusicPortal.html>

LMU – Ludwig-Maximilians-Universität München. (10 de 08 de 2016). *UWE – UML-based Web Engineering*. Obtenido de <http://uwe.pst.ifi.lmu.de/exampleHospInfo.html#modelTop>

LMU – Ludwig-Maximilians-Universität München. (10 de 08 de 2016). *UWE – UML-based Web Engineering*. Obtenido de <http://uwe.pst.ifi.lmu.de/teachingTutorial.html>

Ludwig-Maximilians-Universität München. (10 de 10 de 2016). *UWE – UML-based Web Engineering*. Obtenido de <http://uwe.pst.ifi.lmu.de/index.html>

Marks, S. (01 de febrero de 2013). Google Maps V3 API CodeIgniter Library. Birmingham , Inglaterra, Reino Unido.

Martinez, J. (08 de Marzo de 2020). *jhonny9550.medium.com*. Obtenido de Alpine.js será tu mejor aliado.: <https://jhonny9550.medium.com/alpine-js-ser%C3%A1-tu-mejor-aliado-d5d9a8756edb>

Medina, F. (04 de Febrero de 2016). *SCRIBD*. Obtenido de <https://es.scribd.com/document/298033442/Metodologia-UWE-UML#>

Mena. (2011). *Sistema de Inventario*.

Mifsuf Talón, E. (2012). *Apache*. Madrid: MINISTERIO DE EDUCACIÓN, CULTURA Y DEPORTE - ÁREA DE EDUCACIÓN.

Miguel. (2020). *DESROLLWEB.COM*. Obtenido de <https://desarrolloweb.com/home/tailwind-css>

Morales, J. (24 de Febrero de 2023). *wwwwhatsnew.com*. Obtenido de ¿QUÉ ES TAILWIND CSS Y CUÁLES SON SUS BENEFICIOS?: <https://wwwwhatsnew.com/2023/02/24/que-es-tailwind-css-y-cuales-son-sus-beneficios/>

Muller. (2004). *Fundamentos de Administración de Inventarios*. Grupo Editorial Norma.

MX, E. D. (03 de 10 de 2014). *Definición MX*. Obtenido de <https://definicion.mx/inmobiliaria/>

Netcraft. (31 de Enero de 2013). *PHP solo crece y crece*. Obtenido de <https://news.netcraft.com/archives/2013/01/31/>

Orozco, J. (Abril de 2012). *Diseñar sistemas de informacion*. Obtenido de modelos del ciclo de vida: <https://sites.google.com/site/sistemasbasicos4g/modelos-de-ciclo-de-vida>

Patton, J. (21 de Enero de 2008). *No sé lo que quiero, pero sé cómo conseguirlo*. Obtenido de https://www.jpattonassociates.com/dont_know_what_i_want/

Peralta. (2008). *Sistema de informacion*. Obtenido de colombia:Ecolink.

Perez, F. (2020). *Laritus*. Obtenido de https://www.latirus.com/blog/2020/09/27/todo-lo-que-tenes-que-saber-sobre-livewire-en-laravel/#google_vignette

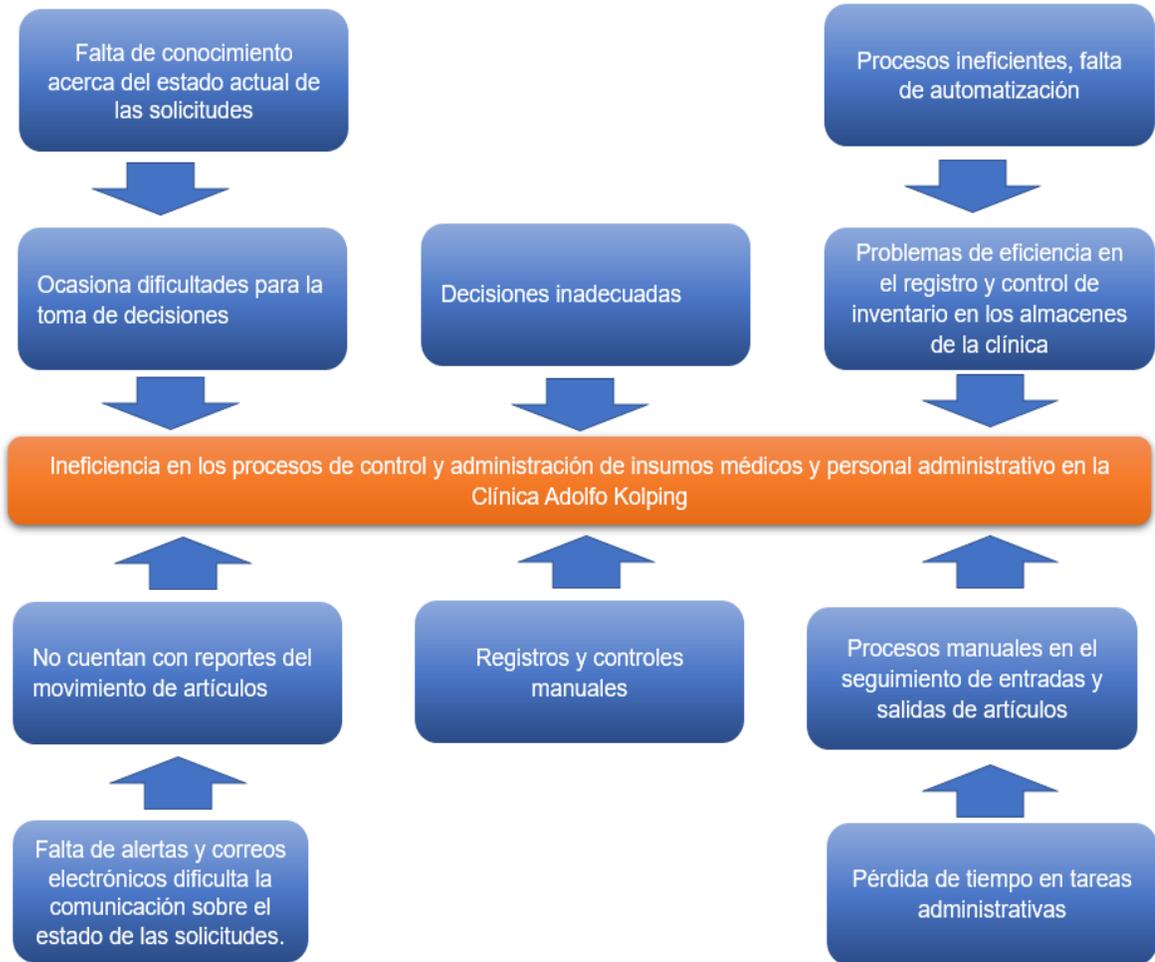
- Periañez, J. y. (2004). *Introducción a la Teoría General de la Administración*. septima edicion de chiavenato idalverto.
- Pressman. (2010).
- Q-Success. (Septiembre de 2019). *W3Techs - Encuestas de tecnología de la World Wide Web*. Obtenido de <https://w3techs.com/>
- Raffino, M. E. (29 de Agosto de 2019). *Concepto.de*. Obtenido de <https://concepto.de/sistema/>
- Rivero, M. (08 de junio de 2019). *Web y Empresas*. Obtenido de <https://www.webyempresas.com/stock-minimo-y-maximo/>
- Robbins, S., Decenzo, D., & Coulter, M. (2013). *Fundamentos de administracion*. México: pearson educación.
- Salamanca, T. (2017). *Sistema Web para el control de: INSUMOS Y PRODUCTOS DEL ÁREA DE PRODUCCION Y ALMACENES*.
- Salazar, G. (Junio de 2009). *Estimación de proyectos de software: un caso práctico*. Obtenido de <https://www.redalyc.org/articulo.oa?id=83512213006>
- Sandoval Caicedo, A. (20 de septiembre de 2012). *Tipos de Stocks*. Obtenido de https://es.slideshare.net/dial_sandoval/tipos-de-stock-inventarios
- Scalan, B. (1987). *El sistema de control de gestion*.
- Sistema Integral de Informacion de extension*. (2015). Obtenido de <http://www2.mdp.edu.ar/index.php/extension/sistema-integral-de-informacion-de-extension>
- Sommerville, I. (2011). *Ingeniería de Software* (Novena edición ed.). México,: CÁMARA NACIONAL DE LA INDUSTRIA EDITORIAL MEXICANA.
- Spurlock, J. (2013). *Bootstrap*. Sebastopol: O'Reilly.
- Stoner, J. y. (1995). *Administracion*. Mexico: Sexta Edicion.

- Tejero, J. J. (2006). *La Gestión Operativa de la Empresa*. Madrid-España: Esic.
- The PostgreSQL Global Development Group. (2019). *PostgreSQL 11.5 Documentation*. California.
- Torres, M. (2014). *Desarrollo de aplicaciones web con PHP*. Lima: Macro EIRL.
- UWE – UML-based Web Engineering. (2016). Obtenido de <https://uwe.pst.ifi.lmu.de/index.html>
- Valencia Ruiz, C. (16 de Junio de 2018). CodeIgniter Rocks. Barcelona, Cataluña, España.
- Van Lancker, L. (2014). *JQuery: el framework JavaScript de la Web 2.0*. Barcelona: Ediciones Eni.
- Vasquez, C. (2015). *Medición y estimación: Método COSMIC*. Obtenido de <http://www.pmoinformatica.com/2018/02/medicion-estimacion-metodo-cosmic.html>
- Vazquez, C. E. (25 de Septiembre de 2015). *Fatto*. Obtenido de Medición, Estimación y Requisitos de Software: <http://www.fattocs.com/es/recursos/articulos.html>
- www003. (s.f.). *Sistemas de gestión de la calidad — Fundamentos y vocabulario*. Obtenido de <https://www.iso.org/obp/ui/es/#iso:std:iso:9000:ed-4:v1:es>
- Yubero, M. (2001). *Manual de Logística Integral*. Obtenido de https://books.google.com.bo/books?id=dxTImJ4ipCMC&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false
- Zapata, J. (2014). *Fundamentos de la gestión de inventarios*. Medellín: Centro Editorial Esumer.
- Zea Ordóñez, M. P., Molina Ríos, J. R., & Redrován Castillo, F. F. (2017). *Administración de Bases de Datos con PostgreSQL*. Alcoy (Alicante): ÁREA DE INNOVACIÓN Y DESARROLLO, S.L.

ANEXOS

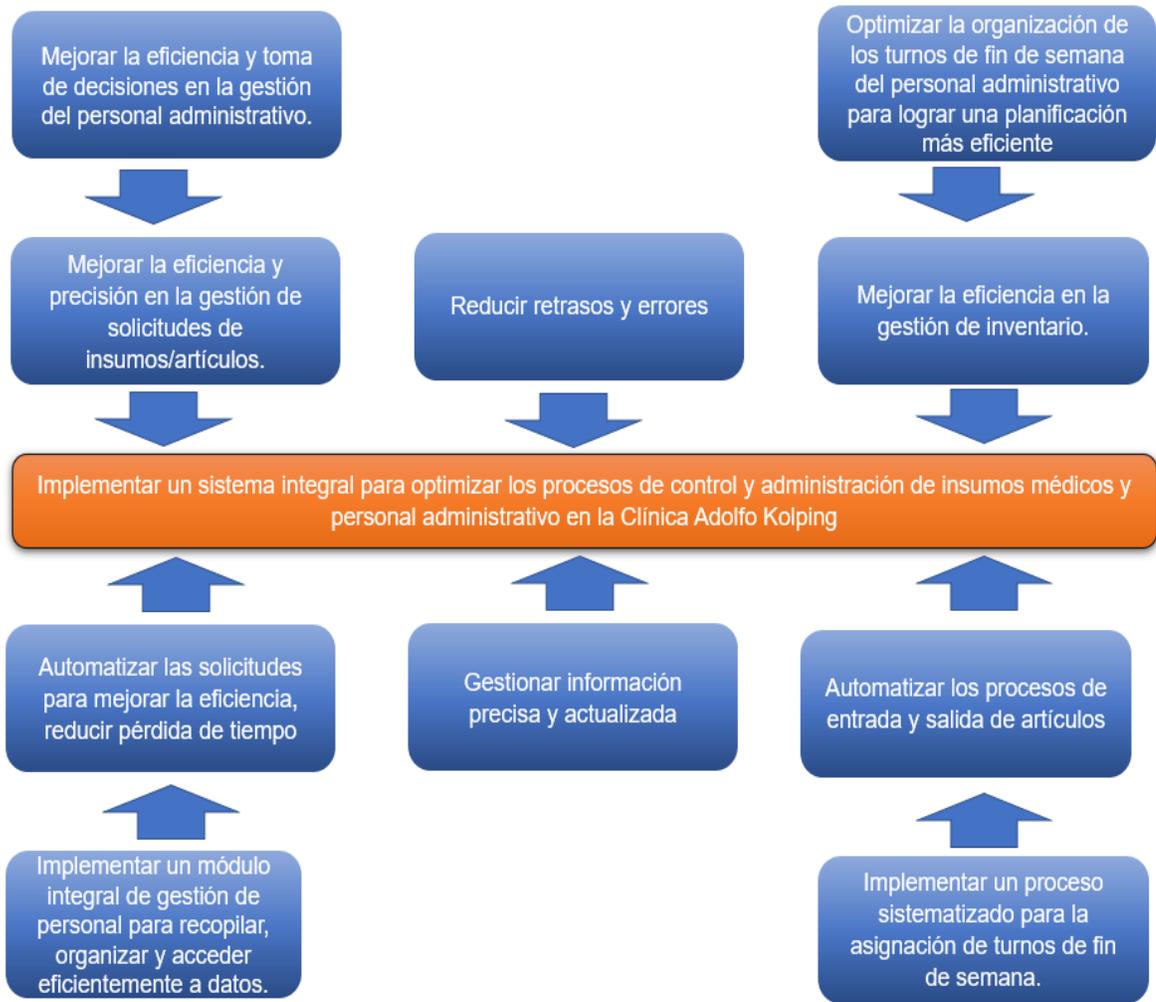
ANEXO A

ÁRBOL DE PROBLEMAS



ANEXO B

ÁRBOL DE OBJETIVOS



ANEXO C

DOCUMENTACIÓN OTORGADA POR LA CLINICA DE ESPECIALIDADES ADOLFO KOLPING

REPORTE DE ARTICULOS

Fecha: 06/07/2023

ID	ARTICULO	CANT_MIN	EXISTENCIA
38	ADMINISTRACION DE MEDICAMENTOS	2	8
69	AIRLIFT MISTIC	5	23
47	ALCOHOL ANTIBACTERIAL SCOTT	3	84
23	AMBIENTADOR POETT 291 GR	5	91
29	AQUANET PARA CABELLO 397 GR	2	3
117	ARCHIVADOR DE PALANCA GRANDE	1	2
118	ARCHIVADOR DE PALANCA PEQUEÑO	1	7
164	ASAP 20	2	20
1	BARBIJO KN95	15	15
55	BARBIJOS DESECHABLES	5	9
52	BASUREROS DE PLASTICO CON TAPA	3	0
65	BH 38	5	4
9	BIRUTA TRAMONTINA PEQUEÑA	5	149
122	BLOCK MENSAJERO SIN ADHESIVO	2	19
79	BOLIGRAFO AZUL	5	56
80	BOLIGRAFO NEGRO	5	99
77	BOLIGRAFO ROJO	5	9
78	BOLIGRAFO VERDE	5	24
56	BOLSAS NEGRAS GRANDES	100	1536
57	BOLSAS NEGRAS PEQUEÑAS	200	1390
58	BOLSAS ROJAS GRANDES	100	1981
59	BOLSAS ROJAS PEQUEÑAS	200	1333
198	CALCULADORA	1	3
81	CARPICOLA MONOPOL GRANDE	2	3
82	CARPICOLA MONOPOL MEDIANO	3	25
194	CDS	1	9
12	CEPILLOS DE BAÑO	2	6
32	CEPILLOS DE ROPA	2	32
174	CEPILLOS GRANDES DE PISO	1	2
30	CEPILLOS PEQUEÑOS DE MANO	2	25
34	CERA OLA DE PISO FLOTANTE 1 LITRO	2	5
20	CERA TIGRE SACHET PEQUEÑO	5	93
76	CLIPS NUMERO 4	5	15
75	CLIPS NUMERO 8	5	10
66	CLOROSPAR	2	0
45	CONCENTIMIENTO INFORMADO ANESTESIOLOGIA	2	8
186	CONSENTIMIENTO DE OFTALMOLOGIA	1	8
43	CONSENTIMIENTO INFORMADO	2	12
147	CONSENTIMIENTO INFORMADO PARA CIRUGIA	1	11
145	CONSENTIMIENTO INFORMADO PARA CIRUGIA	2	0
148	CONSENTIMIENTO INFORMADO PARA LAPARATOMIA	1	14
187	CONSENTIMIENTO INFORMADO UTI	1	8
138	CONTROL CISTOCLISIS	3	8
140	CONTROL DE INGRESOS Y EGRESOS DE LIQUIDOS	1	11
37	CONTROL DE SIGNOS VITALES	2	15



Clínica de Especialidades Adolfo

KOLPING

La Paz, 23 de noviembre 2023

Señor.
ING. DIONICIO HENRY PACHECO RÍOS
TUTOR METODOLOGICO TALLER II
Presente. -

REF. AVAL DE CONFORMIDAD

Distinguido tutor metodológico:

Mediante la presente tengo a bien comunicarle mi conformidad del Trabajo de Grado denominado **“SISTEMA INTEGRAL PARA EL CONTROL Y ADMINISTRACION DE INSUMOS MEDICOS Y PERSONAL ADMINISTRATIVO”** CASO: **CLINICA DE ESPECIALIDADES ADOLFO KOLPING** de modalidad **PROYECTO DE GRADO**, de la Univ. **SWANIA BETTY GUARACHI VELASCO**, con **CI. 9937741 LP**, de tal forma cabe recalcar que el sistema satisface los requerimientos de la institución, de esta forma se dio cumplimiento de los objetivos del presente.

El presente SISTEMA fue IMPLEMENTADO satisfactoriamente en la institución. Es cuando certifico, en honor a la verdad, para fines consiguientes del interesado para su defensa publica y evaluación correspondiente a la materia de Taller de Grado II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública De El Alto.

Atentamente.


S.I.B. Ronald Osmar Ríos Mamani
INGENIERO DE SISTEMAS
R.N.I. 51973

Ing. Ronald Osmar Ríos Mamani
Sistemas
Clínica de Especialidades Adolfo Kolping



El Alto, noviembre de 2023

Señor
Ing. William Roque Roque
DIRECTOR DE CARRERA INGENIERIA DE SISTEMAS
HONORABLE CONSEJO DE CARRERA
Presente. -

Ref. Aval de conformidad

Distinguido Ingeniero:

Mediante la presente tengo a bien comunicarle mi conformidad del proyecto de grado **"Sistema Integral Para el Control y Administración de Insumos Médicos y Personal Administrativo"** Caso: **Clínica de Especialidades Adolfo Kolping**, que propone la postulante Swania Betty Guarachi Velasco, con cédula de identidad 9937741 LP, para su defensa pública evaluación correspondiente a la materia de Taller de Licenciatura II, de acuerdo a reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente,



TUTOR METODOLÓGICO
Lic. Ing. Dionicio Henry Pacheco Ríos

El Alto, noviembre de 2023

Señor
Ing. Dionicio Henry Pacheco Ríos
TUTOR METODOLOGICO TALLER II
Presente. -

Ref. Aval de conformidad

Distinguido Ingeniero:

Mediante la presente tengo a bien comunicarle mi conformidad del proyecto de grado "**Sistema Integral Para el Control y Administración de Insumos Médicos y Personal Administrativo**" Caso: **Clínica de Especialidades Adolfo Kolping**, que propone la postulante Swania Betty Guarachi Velasco, con cédula de identidad 9937741 LP, para su defensa pública evaluación correspondiente a la materia de Taller de Licenciatura II, de acuerdo a reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente,



TUTOR/REVISOR
M.Sc. Ing. Juan Fernando Chambi Guachalla

El Alto, noviembre de 2023

Señor
Lic. Ing. Dionicio Henry Pacheco Rios
TUTOR METODOLOGICO TALLER II
Presente -

Ref. Aval de conformidad

Distinguido Ingeniero.

Mediante la presente tengo a bien comunicarle mi conformidad del proyecto de grado "**Sistema Integral Para el Control y Administración de Insumos Médicos y Personal Administrativo**" Caso: **Clinica de Especialidades Adolfo Kolping**, que propone la postulante Swania Betty Guarachi Velasco, con cédula de identidad 9937741 LP, para su defensa pública evaluación correspondiente a la materia de Taller de Licenciatura II, de acuerdo a reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente,


TUTOR ESPECIALISTA
Ing. Leandro Poma Alanoca