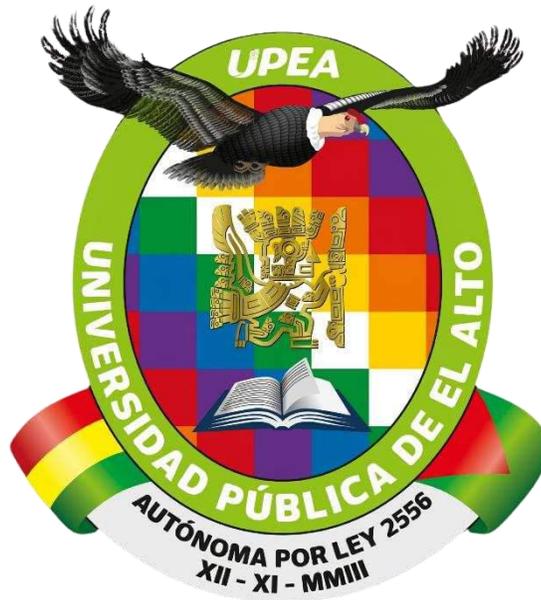


UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

**“SISTEMA WEB DE CONTROL DE ENTRADA Y SALIDA DE
CILINDROS DE OXÍGENO MEDIANTE CÓDIGO QR”
CASO: LABORATORIO INDUSTRIAL DE OXÍGENO MEDICINAL - “G.A.M.E.A.”**

**Para Optar al Título de Licenciatura en Ingeniería de Sistemas
MENCIÓN: INFORMÁTICA Y COMUNICACIONES**

Postulante: Aldair Franklin Calle Quispe
Tutor Metodológico: Lic. Ing. Helen Fanny Suntura Escobar
Tutor Especialista: Ing. Francisco Luis Paez Rocha
Tutor Revisor: M. Sc. Lic. Ing. David Carlos Mamani Quispe

EL ALTO – BOLIVIA

2024

DECLARACIÓN JURADA DE AUTENTICIDAD Y RESPONSABILIDAD

Yo, **ALDAIR FRANKLIN CALLE QUISPE** estudiante con C.I. **9889103 LP** mediante la presente **declaro** de manera pública que la propuesta del **TRABAJO DE GRADO** titulada “**SISTEMA WEB DE CONTROL DE ENTRADA Y SALIDA DE CILINDROS DE OXÍGENO MEDIANTE CÓDIGO QR**”. **CASO: LABORATORIO INDUSTRIAL DE OXÍGENO MEDICINAL - “G.A.M.E.A.”** 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 sometiéndome a las normas establecidas y vigentes de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

El Alto, junio del 2024.

Aldair Franklin Calle Quispe
9889103 L.P.
aldair.calle10@gmail.com

DEDICATORIA

Este proyecto ha requerido de esfuerzo y mucha dedicación, por lo que va dedicado a mi persona, que a pesar de las adversidades que se presentaron a lo largo del camino, en ningún momento se dejó caer, siempre creyendo y confiando en su capacidad y conocimiento para salir adelante, teniendo la convicción necesaria para poder alcanzar y lograr este paso más en la vida.

Ante todo, se la dedico a mi padre, quien me enseñó que incluso la tarea más grande se puede lograr, si se hace un paso a la vez; se la dedico a mi madre, quien me enseñó que el mejor conocimiento que se puede tener, es el que se aprende por sí mismo. A mis padres, por ser los pilares más importantes de mi vida, por demostrarme siempre su cariño y apoyo incondicional, que sin importar nuestras diferencias, hicieron lo posible mi función como persona.

Va dedicado también a mis hermanos y sobrinos que de alguna manera supieron alentarme y apoyarme en todo momento para que culmine mi carrera; a toda mi familia, que me acompañó en todo momento, aportando a mi formación tanto profesional y como ser humano.

AGRADECIMIENTO

Agradecer principalmente a Dios, por brindarme salud, fortaleza y capacidad; por guiarme a lo largo de mi existencia y dejarme vivir este momento tan importante de mi formación profesional.

Un especial agradecimiento a mis padres, por su amor, trabajo y sacrificio en todos estos años, gracias a ellos he logrado llegar hasta aquí y convertirme en lo que soy, es un orgullo y privilegio ser su hijo.

Mi gratitud a todos mis docentes de la carrera “Ingeniería de Sistemas” de la Universidad Pública de El Alto, que con su amplia experiencia y conocimientos, me orientaron para la culminación de mi carrera.

Y de manera particular reconocer a mis Tutores del proyecto por su ayuda, paciencia y dedicación, guiándome en la elaboración del presente trabajo de grado; darles las gracias por haberme brindado su amistad y apoyo, para desarrollarme como persona y profesional.

Por último, agradecer al Gobierno Autónomo Municipal de El Alto, por abrirme las puertas de su institución para poder demostrar mi capacidad profesional y al Laboratorio Industrial de Oxígeno Medicinal, por brindarme la oportunidad y la confianza para el desarrollo de un sistema para mi Proyecto de Grado.

RESUMEN

Los sistemas de información coadyuvan a cumplir los objetivos trazados en una empresa, siendo este una herramienta indispensable y vital para estas entidades. Por tanto, la mayor parte de las empresas se inclinan por hacer uso de un sistema que logre la organización de su información, de forma óptima y segura.

El Laboratorio Industrial de Oxígeno Medicinal, tiene como función producir y distribuir oxígeno medicinal seguro y de calidad a todos los hospitales, centros de salud y población en general de la Ciudad de El Alto. El problema principal radicaba en el control de entrada y salida de cilindros de oxígeno al inventario del almacén, generando retrasos al realizar un pedido o una distribución de estos a los distintos centros de salud.

Para resolver los problemas existentes en el laboratorio, se desarrolla un “Sistema web de control de entrada y salida de cilindros de oxígeno mediante código QR”, con el objetivo de automatizar y mejorar los procesos de registro de la distribución de los cilindros de acuerdo a los pedidos existentes. Aplicando la metodología de desarrollo de software AUP (Proceso Unificado Ágil) y UML (Lenguaje de Modelado Unificado) para el diseño del sistema, MySQL como gestor de base de datos, PHP (Hypertext Preprocessor) como lenguaje de programación y el framework “CodeIgniter” en su versión 4.0.

En conclusión, con la implementación del Sistema Web para el seguimiento y administración de los cilindros de oxígeno de manera eficiente y segura, se logró cumplir con todos los objetivos planteados, cubriendo satisfactoriamente las necesidades y requerimientos del Laboratorio Industrial de Oxígeno Medicinal – “G.A.M.E.A”.

Palabras Clave: Control, Cilindros, Pedidos, Distribución, Código QR.

ABSTRACT

The information systems help to fulfill the objectives outlined in a company, being this an indispensable and vital tool for these entities. Therefore, most of the companies are inclined to make use of a system that achieves the organization of their information, in an optimal and safe way.

The Industrial Laboratory of Medicinal Oxygen has the function of producing and distributing safe and quality medicinal oxygen to all hospitals, health centers and the general population of the city of El Alto. The main problem was the control of the entry and exit of oxygen cylinders to the warehouse inventory, generating delays when placing an order or distributing them to the different health centers.

To solve the existing problems in the laboratory, a "Web system to control the entry and exit of oxygen cylinders through QR code" is developed, with the aim of automating and improving the processes of registration of the distribution of cylinders according to existing orders. Applying the software development methodology AUP (Agile Unified Process) and UML (Unified Modeling Language) for the system design, MySQL as database manager, PHP (Hypertext Preprocessor) as programming language and the framework "CodeIgniter" in its version 4.0.

In conclusion, with the implementation of the Web System for the tracking and administration of oxygen cylinders in an efficient and safe way, it was possible to fulfill all the objectives set, covering satisfactorily the needs and requirements of the Industrial Laboratory of Medicinal Oxygen - "G.A.M.E.A.".

Keywords: Control, Cylinders, Ordering, Distribution, QR Code.

ÍNDICE GENERAL

CAPÍTULO I	1
1. MARCO PRELIMINAR.....	1
1.1. INTRODUCCIÓN	1
1.2. ANTECEDENTES	2
1.2.1. Antecedentes Institucionales.....	2
1.2.1.1. Misión	3
1.2.1.2. Visión.....	3
1.2.1.3. Objetivos	3
1.2.1.4. Organigrama	4
1.2.2. Antecedentes Afines al Proyecto.....	5
1.2.2.1. Antecedentes Internacionales	5
1.2.2.2. Antecedentes Nacionales.....	6
1.3. PLANTEAMIENTO DEL PROBLEMA	6
1.3.1. Problema Principal.....	7
1.3.2. Problemas Secundarios	7
1.3.3. Formulación del Problema	8
1.4. OBJETIVOS.....	8
1.4.1. Objetivo General	8
1.4.2. Objetivos Específicos.....	9
1.5. JUSTIFICACIÓN.....	9
1.5.1. Justificación Técnica	9
1.5.2. Justificación Económica.....	10
1.5.3. Justificación Social.....	10
1.6. METODOLOGÍA	11
1.6.1. Proceso Unificado Ágil (AUP)	11
1.6.2. Lenguaje de Modelado Unificado (UML)	11
1.7. MÉTRICAS DE CALIDAD	12
1.7.1. Estándar ISO/IEC 9126.....	12
1.8. SEGURIDAD DEL SISTEMA	12
1.8.1. Norma ISO 27001	12
1.9. ESTIMACIÓN DE COSTOS.....	12
1.9.1. Modelo COCOMO II.....	12

1.10. HERRAMIENTAS.....	13
1.10.1. Lenguaje PHP.....	13
1.10.2. Gestor MySQL.....	13
1.10.3. CodeIgniter.....	13
1.10.4. Código QR.....	13
1.11. LÍMITES Y ALCANCES.....	14
1.11.1. Límites.....	14
1.11.2. Alcances.....	14
1.12. APORTES.....	15
CAPÍTULO II	17
2. MARCO TEÓRICO.....	17
2.1. INTRODUCCIÓN.....	17
2.2. GESTIÓN DE PRODUCCIÓN Y OPERACIONES.....	17
2.2.1. Pedidos y Distribución.....	18
2.2.2. Administración de Almacenes.....	19
2.3. ENTRADAS Y SALIDAS DE ALMACÉN.....	20
2.3.1. Entradas en el Inventario.....	20
2.3.2. Salidas en el Inventario.....	21
2.4. CONTROL DE INVENTARIO.....	22
2.4.1. Sistema de Inventario.....	22
2.4.1.1. Tipos de Sistemas.....	23
2.4.2. Método de Control.....	23
2.5. CILINDROS DE OXÍGENO.....	24
2.5.1. Oxígeno.....	25
2.5.1.1. Oxígeno Medicinal.....	25
2.5.2. Cilindros.....	26
2.6. CÓDIGO QR.....	27
2.6.1. Ventajas y Características.....	27
2.6.2. Estructura.....	28
2.6.3. Funciones.....	30
2.7. INGENIERÍA DE SOFTWARE.....	30
2.7.1. Sistema Web.....	31
2.8. METODOLOGÍA DE DESARROLLO DE SOFTWARE.....	32
2.8.1. Metodologías Ágiles.....	32

2.9. PROCESO UNIFICADO ÁGIL	34
2.9.1. Fases o Ciclo de Vida	35
2.9.1.1. Iniciación.....	36
2.9.1.2. Elaboración.....	37
2.9.1.3. Construcción.....	38
2.9.1.4. Transición	39
2.9.2. Disciplinas.....	40
2.9.3. Principios o Filosofías	41
2.10. LENGUAJE UML	42
2.10.1. Tipos de Diagrama.....	43
2.10.1.1. Diagramas de Comportamiento.....	43
2.10.1.2. Diagramas Estructurales.....	44
2.11. HERRAMIENTAS DE DESARROLLO	46
2.11.1. Base de Datos.....	46
2.11.1.1. Sistema Gestor de Base de Datos.....	46
2.11.2. Gestor MySQL	47
2.11.2.1. Características	48
2.11.3. Lenguaje de Programación	49
2.11.3.1. Características	50
2.11.4. Lenguaje PHP.....	50
2.11.5. CodeIgniter.....	51
2.11.5.1. Características	52
2.11.5.2. Estructura.....	53
2.12. MÉTRICAS DE CALIDAD	55
2.12.1. Estándar ISO/IEC 9126.....	55
2.12.1.1. Características	56
2.13. ESTIMACIÓN DE COSTOS.....	63
2.13.1. Modelo COCOMO II.....	63
2.13.1.1. Modelos de Desarrollo	64
2.13.1.2. Métodos	64
2.14. SEGURIDAD	68
2.14.1. Triada CID.....	69
2.14.1.1. Confidencialidad	69
2.14.1.2. Integridad.....	70

2.14.1.3. Disponibilidad.....	70
2.14.2. Norma ISO 27001	71
2.14.2.1. Estructura	72
2.15. PRUEBAS DE SOFTWARE.....	73
2.15.1. Caja Blanca	74
2.15.1.1. Prueba de Camino Básico.....	75
2.15.1.2. Tipos de Pruebas	76
2.15.2. Caja Negra.....	76
2.15.2.1. Tipos de Pruebas	77
2.15.3. Pruebas de Estrés	78
2.15.3.1. Formas de Prueba	78
2.15.3.2. Características Principales.....	79
CAPÍTULO III	80
3. MARCO APLICATIVO	80
3.1. INTRODUCCIÓN.....	80
3.2. METODOLOGÍA AUP	81
3.2.1. Fase de Inicio	81
3.2.1.1. Situación Actual	82
3.2.1.2. Descripción de Actores	83
3.2.1.3. Determinación de Requerimientos	84
3.2.2. Fase de Elaboración	87
3.2.2.1. Diagramas de Casos de Uso	87
3.2.3. Fase de Construcción	92
3.2.3.1. Diagrama de Clases.....	92
3.2.3.2. Diagrama de Actividades.....	93
3.2.3.3. Diagrama de Secuencia.....	95
3.2.3.4. Diagrama de Colaboración.....	96
3.2.3.5. Diagrama de Estados.....	96
3.2.3.6. Diagrama de Distribución	97
3.2.4. Fase de Transición.....	98
3.2.4.1. Diagrama de Componentes	98
3.2.4.2. Diseño de Interfaz	99
3.3. PRUEBAS DE SOFTWARE.....	109
3.3.1. Prueba de Caja Blanca	110

3.3.2. Prueba de Caja Negra	112
3.3.2.1. Prueba Registro de Pedidos	114
3.3.3. Prueba de Estrés	115
3.3.3.1. Resultados de Rendimiento	117
3.3.4. Prueba de Funcionalidad	117
3.3.4.1. Resultado Total	122
CAPÍTULO IV	123
4. ANÁLISIS DE CALIDAD, SEGURIDAD Y COSTOS	123
4.1. MÉTRICAS CALIDAD	123
4.1.1. Funcionalidad	123
4.1.1.1. Técnica Punto de Función.....	123
4.1.2. Confiabilidad	128
4.1.3. Usabilidad	129
4.1.4. Mantenibilidad	131
4.1.5. Portabilidad	132
4.1.6. Resultados de Calidad Global	133
4.2. SEGURIDAD DEL SISTEMA	134
4.2.1. Autenticación	135
4.2.1.1. Encriptación	135
4.2.2. Autorización	136
4.2.3. Registro de Auditoria.....	137
4.2.4. Base de Datos	137
4.3. ESTIMACIÓN DE COSTO	138
4.3.1. Modelo COCOMO II.....	138
4.3.1.1. Modelos de Desarrollo	139
4.3.1.2. Costo del Desarrollo del Software	139
CAPÍTULO V	144
5. CONCLUSIONES Y RECOMENDACIONES	144
5.1. CONCLUSIONES	144
5.2. RECOMENDACIONES	145
BIBLIOGRAFÍA.....	146
ANEXOS.....	155

ÍNDICE DE FIGURAS

Figura 1.1. Organigrama del Gobierno Autónomo Municipal de El Alto	4
Figura 2.1. Gestión de Producción y Operaciones	18
Figura 2.2. Características Principales de un Cilindro	26
Figura 2.3. Estructura de un Código QR	29
Figura 2.4. Ciclo de Vida del Proceso Unificado Ágil.....	35
Figura 2.5. Hitos del Proceso Unificado Ágil	40
Figura 2.6. Ejemplo de Diagramas de Comportamiento	44
Figura 2.7. Ejemplo de Diagramas Estructurales	45
Figura 2.8. Sistema Gestor de Base de Datos	47
Figura 2.9. Patrones de Arquitectura MVC	53
Figura 2.10. Técnicas de Prueba	74
Figura 3.1. Planificación del Desarrollo de Software	80
Figura 3.2. Diagrama de Caso de Uso del Sistema Web	87
Figura 3.3. Diagrama de Caso de Uso: Registro de Usuarios	88
Figura 3.4. Diagrama de Caso de Uso: Registro de Centros.....	88
Figura 3.5. Diagrama de Caso de Uso: Registro de Cilindros	89
Figura 3.6. Diagrama de Caso de Uso: Registro de Pedidos	90
Figura 3.7. Diagrama de Caso de Uso: Registro de Distribuciones.....	90
Figura 3.8. Diagrama de Caso de Uso: Registro de Entrada y Salida	91
Figura 3.9. Diagrama de Clases del Sistema	92
Figura 3.10. Modelo Relacional (Base de Datos)	93
Figura 3.11. Diagrama de Actividades del Sistema Web	94
Figura 3.12. Diagrama de Actividades: Ingreso al Sistema	94
Figura 3.13. Diagrama de Secuencia: Registro de Pedido	95
Figura 3.14. Diagrama de Secuencia: Registro de Distribución	95
Figura 3.15. Diagrama de Colaboración: Registro de Módulos	96
Figura 3.16. Diagrama de Estados del Sistema Web	97
Figura 3.17. Diagrama de Distribución del Sistema Web	98
Figura 3.18. Diagrama de Componentes del Sistema Web.....	99
Figura 3.19. Inicio de Sesión del Sistema Web	100
Figura 3.20. Panel Principal del Administrador y Técnico.....	100
Figura 3.21. Panel Principal del Responsable.....	101

Figura 3.22. Interfaz Registro de Usuarios.....	101
Figura 3.23. Interfaz Registro de Centros	102
Figura 3.24. Interfaz Registro de Cilindros	103
Figura 3.25. Interfaz de Pedidos Realizados.....	103
Figura 3.26. Detalle de Pedidos Pendientes	104
Figura 3.27. Interfaz Registro de Pedidos	104
Figura 3.28. Distribuciones del Administrador	105
Figura 3.29. Distribuciones del Responsable	105
Figura 3.30. Interfaz de Inventario del Almacén	106
Figura 3.31. Interfaz Disponibilidad de Cilindros	106
Figura 3.32. Interfaz Móvil de Salidas	107
Figura 3.33. Interfaz Móvil de Entradas	107
Figura 3.34. Interfaz de Externos.....	108
Figura 3.35. Reportes Mensuales y Diarios	108
Figura 3.36. Generación de Reportes	109
Figura 3.37. Reportes Anuales de Centros	109
Figura 3.38. Grafo de Pruebas Actividades del Usuario	110
Figura 3.39. Ventana de Acceso al Sistema.....	113
Figura 3.40. Ventana de Registrar Pedido	114
Figura 3.41. Vinculación al Servidor.....	115
Figura 3.42. Asignación de Usuarios Virtuales	116
Figura 3.43. Rendimiento del Sistema	116
Figura 3.44. Resultado de la Prueba.....	116
Figura 3.45. Resultado de la Prueba: Acceso al Sistema Web.....	118
Figura 3.46. Resultado de la Prueba: Administración de Módulos.....	119
Figura 3.47. Resultado de la Prueba: Registro de Pedidos	120
Figura 3.48. Resultado de la Prueba: Registro de Entrada y Salida.....	121
Figura 3.49. Verificación y Autenticación.....	135
Figura 3.50. Encriptando Contraseña	136
Figura 3.51. Desencriptando Contraseña.....	136
Figura 3.52. Auditoria en la Base de Datos	137

ÍNDICE DE TABLAS

Tabla 2.1. Metodologías Tradicionales y Ágiles.....	33
Tabla 2.2. Factores de Ponderación.....	57
Tabla 2.3. Coeficientes del Modelo COCOMO II	65
Tabla 2.4. Factor de Conversión LDC/PF	66
Tabla 2.5. Valores de Atributo de Costes FAE.....	67
Tabla 3.1. Obtención de Requisitos.....	82
Tabla 3.2. Funciones Básicas del Sistema	84
Tabla 3.3. Descripción de Caso de Uso: Registro de Usuarios	88
Tabla 3.4. Descripción de Caso de Uso: Registro de Centros	89
Tabla 3.5. Descripción de Caso de Uso: Registro de Cilindros.....	89
Tabla 3.6. Descripción de Caso de Uso: Registro de Pedidos.....	90
Tabla 3.7. Descripción de Caso de Uso: Registro de Distribuciones	91
Tabla 3.8. Descripción de Caso de Uso: Registro de Entrada y Salida	91
Tabla 3.9. Valores Límites de Inicio de Sesión	113
Tabla 3.10. Prueba de Caja Negra: Inicio de Sesión	113
Tabla 3.11. Valores Límites de Registro de un Pedido	114
Tabla 3.12. Prueba de Caja Negra: Registro Pedido.....	115
Tabla 3.13. Prueba de Funcionalidad: Acceso al Sistema Web.....	118
Tabla 3.14. Prueba de Funcionalidad: Administración de Módulos.....	119
Tabla 3.15. Prueba de Funcionalidad: Registro de Pedidos	120
Tabla 3.16. Prueba de Funcionalidad: Registro de Entrada y Salida	121
Tabla 4.1. Factores de Ponderación de Punto de Función	125
Tabla 4.2. Ajuste del Factor de Complejidad	126
Tabla 4.3. Encuesta de Usabilidad del Sistema.....	130
Tabla 4.4. Valores Determinantes en la Mantenibilidad	131
Tabla 4.5. Resultados de Métricas de Calidad	133
Tabla 4.6. Coeficientes del Método COCOMO II	139
Tabla 4.7. Factor de Conversión LDC/PF	140
Tabla 4.8. Valores de Atributo de Costes FAE.....	141

ÍNDICE DE ECUACIONES

<i>Ecuación (1)</i> Punto de Función	58
<i>Ecuación (2)</i> Funcionalidad.....	59
<i>Ecuación (3)</i> Confiabilidad	59
<i>Ecuación (4)</i> Probabilidad de Falla	60
<i>Ecuación (5)</i> Probabilidad de No Falla	60
<i>Ecuación (6)</i> Usabilidad	60
<i>Ecuación (7)</i> Índice de Madurez	62
<i>Ecuación (8)</i> Portabilidad	62
<i>Ecuación (9)</i> Esfuerzo Requerido	66
<i>Ecuación (10)</i> Tiempo Requerido.....	66
<i>Ecuación (11)</i> Número de Personas	66
<i>Ecuación (12)</i> Complejidad Ciclomática.....	75
<i>Ecuación (13)</i> Punto de Función Máximo.....	127
<i>Ecuación (14)</i> Líneas de Código	140
<i>Ecuación (15)</i> Miles de Líneas de Código.....	140
<i>Ecuación (17)</i> Costo Total	143

LISTA DE ABREVIATURAS

1. **GAMEA:** Gobierno Autónomo Municipal de El Alto
2. **LIOM:** Laboratorio Industrial de Oxígeno Medicinal
3. **TIC:** Tecnologías de Información y Comunicaciones
4. **AUP:** Proceso Unificado Ágil
5. **RUP:** Proceso Racional Unificado
6. **UML:** Lenguaje Unificado de Modelado
7. **QR:** Código de Respuesta Rápida
8. **SDLC:** Ciclo de Vida del Desarrollo de Software
9. **PHP:** Pre Procesador de Hipertexto
10. **HTML:** Lenguaje de Etiquetas de Hipertexto
11. **MVC:** Modelo Vista Controlador
12. **SQL:** Lenguaje de Consulta Estructurada
13. **SGBD:** Sistema Gestor de Base de Datos
14. **DBMS:** Sistema de Administración de Bases de Datos
15. **SGSI:** Sistema de Gestión de Seguridad de la Información
16. **ISO:** Organización Internacional de Normalización
17. **TDD:** Desarrollo de Software Basado en Pruebas
18. **HTTP:** Protocolo de Transferencia de Hipertexto
19. **URL:** Localizador Uniforme de Recursos
20. **FTP:** Protocolo de Transferencia de Ficheros
21. **FAE:** Factor de Ajuste de Experiencia
22. **LDC:** Líneas De Código
23. **KLDC:** Kilo Líneas De Código



CAPÍTULO I

MARCO PRELIMINAR

1. MARCO PRELIMINAR

1.1. INTRODUCCIÓN

En la actualidad el uso acelerado y el constante crecimiento de las Tecnologías de Información y Comunicaciones (TIC), se han introducido en todas las actividades económicas, comerciales y empresariales. Por tal motivo, se puede evidenciar que las prestaciones de estas tecnologías dan origen a los sistemas informáticos, los cuales tienen la característica del tratamiento automático de la información mediante dispositivos electrónicos y sistemas computacionales, brindando de esta manera procesos de control, supervisión, automatización y optimización de recursos, coadyuvando en la toma de decisiones oportunas, confiables y efectivas, limitando los riesgos y mejorando las ganancias de una empresa o institución.

El Gobierno Autónomo Municipal de El Alto, conjuntamente con la Secretaría Municipal de Salud, frente a la situación de emergencia sanitaria que vivió el país a causa de la pandemia del Coronavirus (COVID-19), pusieron en marcha una planta generadora de oxígeno ubicado en la Zona Mercedario del Distrito 4 de la ciudad de El Alto, que tiene como función principal producir y distribuir oxígeno medicinal seguro y de calidad, para abastecer a todos los hospitales, centros de salud y población en general, con la finalidad de evitar la escasez de este gas vital en todo el Municipio Alteño.

El Laboratorio Industrial de Oxígeno Medicinal, realiza el control de la entrada y salida de cilindros de oxígeno a través de formularios y planillas de trabajo, que posteriormente son registrados en hojas de Microsoft Excel, provocando demora al momento de realizar las distribuciones de este gas a los centros de salud, presentando problemas con la identificación de cilindros e inconvenientes con la información que se maneja en el inventario del almacén; de esta manera, el método de control actual no cubre todas las necesidades del laboratorio, lo que provoca que sea deficiente para los requerimientos del usuario.

Por lo tanto, se desarrollará un Sistema Web de control de entrada y salida de cilindros de oxígeno mediante código QR, que cumpla con todos los requerimientos del personal en cuanto al registro de entradas y salidas, integrando los módulos de pedidos y distribución de cilindros de oxígeno; un sistema que procesará la información y los datos en forma integrada y segura, con el fin de satisfacer las necesidades de los usuarios y dar soluciones a los problemas detectados en el método de control actual. Utilizando las herramientas de MySQL como gestor de base de datos y PHP como lenguaje de programación, aplicando la metodología de desarrollo AUP (Proceso Unificado Ágil) y UML (Lenguaje de Modelado Unificado) para el diseño del sistema.

1.2. ANTECEDENTES

1.2.1. Antecedentes Institucionales

El Gobierno Autónomo Municipal de El Alto, también conocido por sus siglas “G.A.M.E.A.”, es el nombre oficial de la instancia de gobierno del Municipio de El Alto o Alcaldía de El Alto, que administra el territorio comprendido por el municipio homónimo. La entidad inició su funcionamiento bajo el nombre de Alcaldía tras la creación de la entidad territorial en 1985, previamente este territorio era administrado como parte del municipio vecino de La Paz.

El Laboratorio Industrial de Oxígeno Medicinal, asimismo nombrado gracias a sus siglas como “LIOM”, fue creado con la finalidad de evitar el desabastecimiento de este gas vital en el Municipio Alteño, tomando en cuenta la población en general y los establecimientos de salud de tercer, segundo y primer nivel de atención. Por tanto, al realizar las gestiones correspondientes para obtener todos los permisos correspondientes, le otorgo la viabilidad a obtención de la documentación que le permite producir, distribuir y comercializar el Oxígeno Medicinal en la ciudad de El Alto.

1.2.1.1. Misión

"Nuestra misión es proporcionar un suministro confiable y de alta calidad de oxígeno medicinal e industrial a nuestros clientes, ayudando a mejorar la calidad de vida de las personas y a impulsar el desarrollo económico de la región, todo ello a través de un compromiso inquebrantable con la seguridad, la eficiencia y la sostenibilidad en todas nuestras operaciones."

1.2.1.2. Visión

"Nuestra visión es ser líderes en la generación y distribución de oxígeno medicinal e industrial, reconocidos por la calidad de nuestros productos y servicios, nuestra innovación constante, y nuestro compromiso con el desarrollo sostenible. Queremos ser la opción preferida de nuestros clientes, colaboradores y comunidades, y contribuir al mejoramiento de la calidad de vida de las personas y al desarrollo económico de la región, siempre con un enfoque en la seguridad, la eficiencia y la excelencia en todas nuestras operaciones."

1.2.1.3. Objetivos

Los cinco objetivos esenciales del Laboratorio Industrial de Oxígeno Medicinal son:

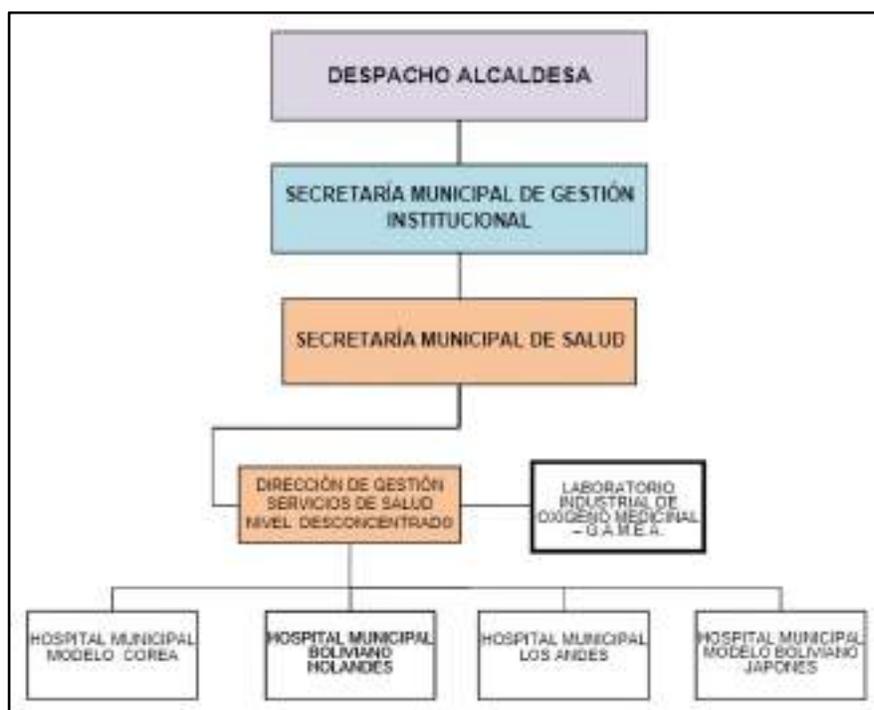
- Proporcionar un suministro constante de oxígeno a la comunidad: El objetivo principal del Laboratorio Industrial de Oxígeno Medicinal es garantizar un suministro constante y confiable de oxígeno a la comunidad, especialmente durante situaciones de emergencia como la pandemia del COVID-19.
- Mantener altos estándares de calidad: Es importante que el Laboratorio Industrial de Oxígeno Medicinal mantenga altos estándares de calidad en el proceso de producción y distribución del oxígeno para garantizar que el producto final cumpla con los requisitos de calidad y seguridad.

- Optimizar los costos de producción: El Laboratorio Industrial de Oxígeno Medicinal debería tener como objetivo optimizar los costos de producción para asegurarse de que el precio de venta sea competitivo en el mercado y al mismo tiempo maximizar la rentabilidad.
- Expandir la capacidad de producción: Si la demanda de oxígeno aumenta, la planta debería estar en capacidad de ampliar su capacidad de producción para satisfacer la demanda de la comunidad.
- Establecer una relación sólida con la comunidad y los proveedores: Es importante que el Laboratorio Industrial de Oxígeno Medicinal desarrolle relaciones sólidas con la comunidad y los proveedores para mantener una comunicación efectiva y garantizar una cooperación efectiva en momentos de emergencia.

1.2.1.4. Organigrama

Figura 1.1.

Organigrama del Gobierno Autónomo Municipal de El Alto



Nota. Estructura Organizativa, Gobierno Autónomo Municipal, 2023, www.elalto.gob.bo

1.2.2. Antecedentes Afines al Proyecto

1.2.2.1. Antecedentes Internacionales

- Vargas Guzmán, K. A. y León Castañeda, D. M. (2017). *“Implementación de código QR Como Método de codificación, para sistemas de inventario a través de un aplicativo móvil y servicios web”*. [Repositorio Universidad Distrital]. Aplicación móvil diseñada e implementada exclusivamente para un cliente en específico, básicamente es un sistema de inventario que permite mantener un control sobre las compras y ventas realizadas por la tienda, adicionalmente cuenta con otras opciones como contactos, administración de cuentas y consultas de productos por medio de códigos QR.
- Monteza Torres, C. (2019). *“Diseño e implementación de un sistema web para la mejora de procesos en la gestión de almacén de la empresa Carrocería Lima Traylers S.A.C.”*. [Universidad Peruana De Las Américas]. El proyecto consiste en diseñar e implementación de un sistema web para mejorar el proceso de la gestión de almacén, para su mayor productividad a base de resultados que serán desarrollados a corto plazo de la empresa Carrocería Lima Traylers.
- Santamaría Vílchez, C. A. (2018). *“Implementación de un sistema web adaptativo para apoyar en la gestión comercial utilizando el Método de costo promedio en la empresa Ferrotumi S.A.C.”*. [Universidad Católica Santo Toribio de Mogrovejo]. El trabajo tiene como objetivo mejorar la gestión comercial de la empresa Ferrotumi S.A.C. Donde su mayor inconveniente es tiempo de demora en la atención al cliente, debido a que no se conoce el stock disponible de cada producto, por lo cual se toma tiempo en verificar; pretendiendo agilizar el proceso de las ventas, las solicitudes de pedidos por parte de los clientes hacia la empresa, así como también optimizando el tiempo de despacho de la mercadería a los clientes.

1.2.2.2. Antecedentes Nacionales

- Mamani Anave, N. I. (2016). *“Sistema web de gestión de inventarios basado en código QR. Caso: empresa ITSEVEN”*. [Universidad Mayor de San Andrés]. Sistema web de gestión de inventarios para la empresa ITSEVEN basado en códigos QR. El cual permite realizar los registros y control de los productos y/o insumos para la verificación y control de los servicios que ofrece la empresa enfocándose en un servicio posventa, realizando la verificación de la información con un dispositivo móvil que tenga una aplicación capaz de leer los códigos QR.
- Condori Maldonado, J. L. (2014). *“Sistema de Pedidos Vía Web y Control de Inventarios CASO: Plásticos V.J.F. Ltda”*. [Universidad Mayor de San Andrés]. Sistema de pedidos vía web y control de inventarios que optimiza los procesos de pedidos y brinda información rápida y fidedigna sobre el control de inventarios de la empresa Plásticos V.J.F. Ltda.
- Quispe Suri, R. (2020). *“Sistema Web para la gestión de producción, pedidos y entregas a través de código QR”*. [Universidad Pública de El Alto]. Sistema web para la gestión de producción, pedidos y entrega a través de código QR en la empresa de fábrica de calaminas aceros Isacar S.R.L. y tener una buena planificación en la producción de calaminas.

1.3. PLANTEAMIENTO DEL PROBLEMA

En Laboratorio Industrial de Oxígeno Medicinal, no cuenta con un sistema centralizado y automatizado que garantice un control preciso de las entradas y salidas de sus cilindros al almacén; por lo que la falta de un mecanismo eficiente y seguro, para el seguimiento y administración de los cilindros de oxígeno, da lugar a diferentes problemas en el laboratorio, como la demora en los pedidos, dificultades en la distribución y errores en el inventario.

Así que el problema radica principalmente en la gran cantidad de información que se genera en el registro de entrada y salida de los cilindros al inventario. Dicha información se registra de forma manual, lo cual implica registros erróneos, extravío y mala manipulación de información, que tiene como consecuencia pérdidas económicas.

La identificación de cilindros de oxígeno es prácticamente demoroso y complicado para el personal, desconociendo su disponibilidad, su localización y su estado. Los pedidos de los centros de salud, son realizados en persona donde se emite una orden de trabajo detallando los cilindros requeridos, tomando mucho tiempo para este proceso simple. Las distribuciones realizadas son descritas y elaboradas a través de hojas Excel, sin ser una forma segura de resguardar ni de respaldar esa información tan esencial.

De esta manera, la gestión manual de los cilindros de oxígeno, sin un sistema adecuado para los respectivos procesos, puede llegar a generar riesgos en la salud de los habitantes de la ciudad de El Alto, que requieran oxígeno de manera inmediata.

1.3.1. Problema Principal

El Laboratorio Industrial de Oxígeno Medicinal, actualmente no cuenta con un sistema que pueda controlar la entrada y salida de cilindros de oxígeno de forma segura y eficiente, ya que el registro de dichos procesos se realiza de forma manual, lo cual involucra demora en realizar los pedidos, retrasos en la distribución de cilindros, inexactitud en el inventario del almacén y pérdida de información.

1.3.2. Problemas Secundarios

- Dificultad en la identificación de cilindros: En situaciones de emergencia, es vital contar con un método eficiente para identificar y localizar rápidamente los cilindros de oxígeno disponibles, lo cual puede ser un desafío sin un sistema de registro adecuado.

- Inseguridad en el inventario: Sin un registro adecuado del inventario, los cilindros de oxígeno pueden perderse o ser objeto de robos, lo que implica una pérdida financiera para la institución y una disminución en la disponibilidad de cilindros de oxígeno.
- Retrasos en los pedidos: En momentos de urgencia con respecto a la salud, es necesario y preciso contar con un sistema que realice la solicitud de pedidos en línea basado en la web, reduciendo el tiempo necesario para la adquisición de cilindros de oxígeno por parte de los centros de salud que los requieran o necesiten con urgencia.
- Falta de control en la distribución: La ausencia de un registro detallado de los cilindros de oxígeno enviados a los centros de salud, dificulta el seguimiento y monitoreo de los mismos, desconociendo sus fechas de trazabilidad, recarga y mantenimiento, lo que podría llevar a su mal funcionamiento afectando la atención médica oportuna a los pacientes.

1.3.3. Formulación del Problema

¿De qué manera se puede tener un control de la entrada y salida de cilindros de oxígeno en el Laboratorio Industrial de Oxígeno Medicinal, donde se registre la distribución de los cilindros de acuerdo a los pedidos de los distintos centros de salud?

1.4. OBJETIVOS

1.4.1. Objetivo General

Desarrollar un Sistema Web de control de entrada y salida de cilindros de oxígeno mediante código QR, para el Laboratorio Industrial de Oxígeno Medicinal - "G.A.M.E.A.", que permita tener un registro adecuado de la distribución de acuerdo a los pedidos de los centros de salud, donde se proporcione información precisa del inventario de cilindros, garantizando una gestión eficiente a fin de optimizar la toma de decisiones.

1.4.2. Objetivos Específicos

- Recolectar información del proceso de entrada y salida de los cilindros, para determinar los requerimientos del método de control actual a partir de un análisis previo.
- Diseñar una interfaz gráfica sencilla e intuitiva, para facilitar la interacción entre el sistema y el usuario, agilizando los tiempos de acceso rápido a la información.
- Implementar el registro de los cilindros de oxígeno generando su respectivo código QR de manera única y fiable, para su identificación y localización de forma rápida y sencilla.
- Automatizar el registro del inventario, para tener una información adecuada y precisa de todos los cilindros de oxígeno, conociendo en todo momento su disponibilidad.
- Desarrollar un módulo para el pedido de cilindros de oxígeno, para acelerar y optimizar este proceso de solicitud, facilitando el trabajo a los respectivos centros de salud.
- Sistematizar el proceso de lectura y almacenamiento de cilindros mediante QR, para el registro de las operaciones de recarga, entrada y salida.
- Generar reportes en base a las distribuciones realizadas de los cilindros de oxígeno, a los respectivos centros de salud, para la toma de decisiones acertadas y oportunas.

1.5. JUSTIFICACIÓN

1.5.1. Justificación Técnica

Para el desarrollo del Sistema Web, es necesario el uso de un equipo de cómputo y herramientas de desarrollo de software que la mayoría son de uso libre, de esta manera no se generará un costo por su utilización. En cuanto al hardware el Laboratorio Industrial de Oxígeno Medicinal, cuenta con todas las instalaciones para que el sistema opere de manera óptima, disponiendo de computadoras portátiles (laptops) y dispositivos electrónico portátiles (celulares) para el facilitar el escaneo de cilindros, contando a la vez con acceso a internet por medio de red inalámbrica (Wi-Fi).

Por lo tanto, al cumplir con las herramientas necesarias para el desarrollo, hacen que el sistema sea factible técnicamente, porque el laboratorio cuenta con los equipos y recursos necesarios para el desarrollo, implementación y el normal funcionamiento del sistema.

1.5.2. Justificación Económica

La implementación del sistema web permitirá tener un control adecuado de la información, logrando con esto reducir la intervención de recurso humano para realizar la elaboración de planillas y formularios en el laboratorio, reduciendo de esta manera gastos en material de escritorio. El software a utilizar en el desarrollo del proyecto no necesita licencia, por lo que es más económico al momento de la implementación, será desarrollado en un lenguaje de programación basados en Web, con un gestor de base de datos fiable, los mismos que tienen la característica de ser software libre.

Estará alojado sobre un servidor proporcionado por el Gobierno Autónomo Municipal de El Alto, ya que al formar parte de su dependencia no generará un costo para el laboratorio. En este entendido el proyecto se justifica económicamente, analizando los ahorros que se generan a través de la implementación del sistema web, contando con una información útil, organizada y confiable.

1.5.3. Justificación Social

El Sistema Web beneficiará directamente a los usuarios Administradores y Técnicos del laboratorio, ya que le permitirá tener el control adecuado de los registros de entrada y salida de cilindros, información con respecto al inventario y la coordinación de la distribución de acuerdo a los pedidos realizados mediante un control sistemático. El Laboratorio Industrial de Oxígeno Medicinal, tendrá la seguridad de contar con sistema eficaz y adecuado para el control de sus cilindros de oxígeno, brindando así un servicio de primer nivel a la comunidad.

Los responsables de cada centro de salud serán los beneficiados indirectos, ya que son los que acceden al sistema para solicitar los cilindros requeridos en su lugar de trabajo. De esta manera, el sistema se justifica socialmente, ya que los usuarios del laboratorio ya no realizarán el trabajo tedioso de registrar los cilindros de forma manual y al reducir el desgaste humano el personal aumenta su nivel tanto competitivo como productivo.

1.6. METODOLOGÍA

1.6.1. Proceso Unificado Ágil (AUP)

El Proceso Unificado Ágil o Agile Unified Process (AUP) en inglés, es una versión simplificada del Proceso Unificado de Rational (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. (Ariagna, 2011)

Las fases del ciclo de desarrollo son:

- INCEPCIÓN
- ELABORACIÓN
- CONSTRUCCIÓN
- TRANSICIÓN

1.6.2. Lenguaje de Modelado Unificado (UML)

El lenguaje de modelado unificado (UML) es un estándar para la representación visual de objetos, estados y procesos dentro de un sistema. Por un lado, el lenguaje de modelado puede servir de modelo para un proyecto y garantizar así una arquitectura de información estructurada; por el otro, ayuda a los desarrolladores a presentar la descripción del sistema de una manera que sea comprensible para quienes están fuera del campo. (Soriano, 2018)

1.7. MÉTRICAS DE CALIDAD

1.7.1. Estándar ISO/IEC 9126

El estándar ISO/IEC 9126, presenta un marco conceptual para el modelo de calidad y define un conjunto de características y subcaracterísticas, las que debe cumplir todo producto software. 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 subcaracterísticas que permiten profundizar en la evaluación de la calidad de productos de software. (Verity, 2022)

1.8. SEGURIDAD DEL SISTEMA

1.8.1. Norma ISO 27001

La ISO 27001, es la norma internacional que proporciona un marco de trabajo para los sistemas de gestión de seguridad de la información (SGSI) con el fin de proporcionar confidencialidad, integridad y disponibilidad continuada de la información, así como cumplimiento legal. (NQA, 2013)

1.9. ESTIMACIÓN DE COSTOS

1.9.1. Modelo COCOMO II

Permite realizar estimaciones en función del tamaño del software, y de un conjunto de factores de coste y de escala. En los factores de coste se incluyen aspectos relacionados con la naturaleza del sistema, equipo, y características propias del proyecto. Los factores de escala incluyen la parte de escala producida a medida que un proyecto de software incrementa su tamaño. (Gracia, 2012)

1.10. HERRAMIENTAS

1.10.1. Lenguaje PHP

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor), es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. (PHP, 2001)

1.10.2. Gestor MySQL

MySQL es un sistema de gestión de bases de datos relacionales (RDBMS) de código abierto respaldado por Oracle y basado en el lenguaje de consulta estructurado (SQL), se asocia más a menudo con las aplicaciones web y la publicación en línea. (TechTarget, 2013)

1.10.3. CodeIgniter

Es una framework PHP de código abierto, desarrollada por primera vez en 2006. Permite la creación de aplicaciones web más rápida en comparación con otras framework. CodeIgniter utiliza el Model View Controller (MVC), el cual es necesario para optimizar el trabajo de programación de las aplicaciones. Consiste en un patrón de arquitectura de software que tiene la finalidad de separar la lógica de control, de la interfaz de usuario y los datos de la plataforma durante el proyecto de desarrollo de app. (APPYWEB, 2013)

1.10.4. Código QR

En inglés QR Code, son un tipo de códigos de barras bidimensionales; a diferencia de un código de barras convencional, la información está codificada dentro de un cuadrado, permitiendo almacenar gran cantidad de información alfanumérica. Los códigos QR son fácilmente identificables por su forma cuadrada y por los tres cuadros ubicados en las esquinas superiores e inferior izquierda. (Códigos QR, 2009)

1.11. LÍMITES Y ALCANCES

1.11.1. Límites

El Sistema Web de control de entrada y salida de cilindros de oxígeno mediante código QR, se limita a las siguientes condiciones:

- Al sistema no podrán acceder usuarios externos, ya sea para para solicitar cilindros u otro proceso, ya que está programado para dar acceso solo al personal asociado al laboratorio.
- El sistema no contará con una extensión de aplicación móvil, ya que este se considera más accesible para el registro mediante un lector QR.
- El sistema no realizará la facturación electrónica de pedidos y/o venta de cilindros a usuarios externos, debido a que no se incluirá un módulo para este proceso.

1.11.2. Alcances

El Sistema a desarrollar almacenará la información en una base de datos gestionada a través de un administrador, por lo que brindará una información rápida y confiable. También será flexible en cuanto a su manejo, ya que se adaptará con facilidad a los requerimientos del personal del laboratorio.

El sistema estará conformado por la implementación del proceso de registro, control y reportes de los cilindros de oxígeno, así que el sistema contará con los siguientes módulos:

- Módulo de Usuarios: Realizará el registro y control de los usuarios que tendrán acceso al sistema, de acuerdo al rol y/o tipo de privilegio con el que cuente.
- Módulo de Centros: Encargado de realizar el registro de los centros de salud asociados al laboratorio, los cuales efectuaran pedidos de oxígeno y de esta manera a los que se procederá con la distribución.

- Módulo de Cilindros: Encargado de realizar el registro de los cilindros de oxígeno con los que se cuenta en el laboratorio, asignándoles un código QR único para facilitar el movimiento en las entradas y salidas que se ejecuten.
- Módulo de Pedido: Realizará el registro de los pedidos de cilindros de oxígeno de los respectivos centros de salud, a través de sus usuarios responsables.
- Módulo de Distribución: Llevará a cabo el registro de la información de la salida de cilindros de oxígeno en base a los pedidos.
- Módulo de Inventario: Encargado de almacenar la información de los cilindros de oxígeno, así como su estado y ubicación de los mismos, para su determinado control.
- Módulo de Externos: Realizará el registro de pedidos y distribución de cilindros de oxígeno, de los usuarios externos que requieran en algún caso particular de este gas vital.
- Módulo de Reportes: Se generará reportes del movimiento de los cilindros de oxígeno a cada centro asociado mediante fechas diarias, mensuales y anuales, con fines informativos para la toma de decisiones.

1.12. APORTES

La implementación de un Sistema Web de control de entrada y salida de cilindros de oxígeno mediante código QR, proporcionará numerosos aportes al Laboratorio Industrial de Oxígeno Medicinal - "G.A.M.E.A.".

Algunos de estos aportes incluyen:

- Mejora en la gestión del inventario: El sistema web permitirá un control más preciso y automatizado del inventario de cilindros de oxígeno. Esto facilitará el seguimiento de la cantidad de cilindros disponibles, su ubicación y su estado, lo que a su vez optimizará la planificación de la distribución y asegurará que haya suficientes cilindros disponibles cuando sean necesarios.

- Mayor eficiencia en la distribución: Con el sistema de código QR, la entrada y salida de los cilindros de oxígeno se registrarán de manera rápida y precisa. Esto agilizará el proceso de distribución, permitiendo una asignación más eficiente de cilindros a los lugares donde se necesiten, incluyendo hospitales, centros de salud y otros establecimientos médicos.
- Información de actividades: Al ser un sistema web, toda la información adecuada y precisa con respecto a las actividades del laboratorio, será accesible desde cualquier lugar las 24 horas del día y con una seguridad total, facilitando de esta manera la adquisición de los cilindros de oxígeno a los centros de salud y población que lo requiera.
- Reducción de errores y pérdidas: El uso de códigos QR en el sistema minimizará los errores humanos y la pérdida de cilindros. Cada cilindro estará debidamente registrado y su ubicación podrá ser rastreada en todo momento, lo que evitará situaciones de pérdida o robo, permitiendo una mayor responsabilidad y control sobre los recursos.
- Seguimiento del mantenimiento: El sistema facilitará el seguimiento de las fechas de mantenimiento, inspección y recarga de los cilindros de oxígeno. Esto ayudará a garantizar que los cilindros estén en buen estado y cumplan con los requisitos de seguridad, evitando el uso de cilindros que estén fuera de fecha o en mal estado.
- Generación de reportes y análisis: El sistema permitirá generar reportes detallados y análisis sobre la salida de los cilindros de oxígeno. Estos reportes proporcionarán datos valiosos sobre la demanda, el consumo y otros indicadores relevantes; estos datos pueden ayudar en la toma de decisiones, la planificación de recursos y la optimización de los procesos relacionados con el suministro de oxígeno medicinal.



CAPÍTULO II

MARCO TEÓRICO

2. MARCO TEÓRICO

2.1. INTRODUCCIÓN

En el desarrollo de este capítulo, se tendrá y describirá el soporte teórico, contextual o legal de los conceptos que se utilizarán para la implementación del proyecto, así como de las tecnologías de información que se emplearan en su progreso, ya que las bases teóricas y los conceptos son claves para el desarrollo del Sistema Web.

Esta etapa es el fundamento de toda la investigación, integrado por un conjunto de conocimientos con el fin de apoyar el estudio del proyecto.

2.2. GESTIÓN DE PRODUCCIÓN Y OPERACIONES

La gestión de producción corresponde a la utilización de métodos y técnicas, con el fin de llevar las materias a convertirse en productos acabados. Este proceso radica en una cadena de acciones en las que se relacionan los elementos indicados, desde la participación del recurso humano, manipulando las materias primas por medio de las máquinas necesarias, con el fin de lograr una distribución por producto con el nivel de calidad y cantidad esperados.

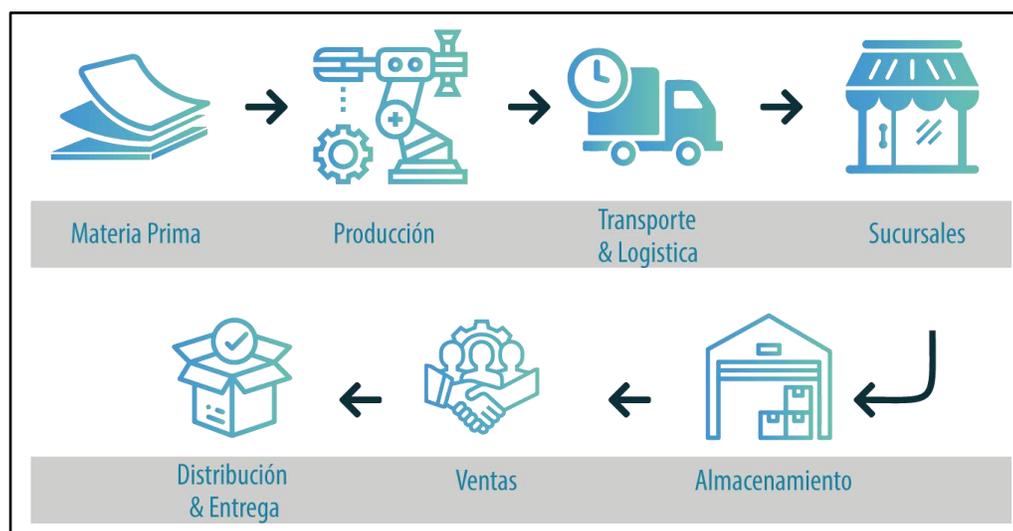
La gestión de producción y operaciones permite una planificación ideal, una organización adecuada y una supervisión final de los pasos que hacen parte de la línea de elaboración y entrega final del producto y/o servicio. De esta forma, se garantiza que la productividad empresarial se vea reflejada de forma efectiva y eficaz en los objetivos de la gestión de producción y operaciones, para que los insumos disponibles se conviertan en bienes. Hay dos procesos clave que intervienen en la gestión de producción:

- La Gestión de la Cadena de Suministros.
- La Gestión Logística.

En estos dos puntos se invierte la mayor cantidad de costos, gastos e inversiones; por esta razón se ven reflejados en los resultados que muestran mayor impacto para la organización, para ello se suman otros factores importantes, como el uso controlado de recursos y la rentabilidad. (Beetrack, 2020)

Figura 2.1.

Gestión de Producción y Operaciones



Nota. Cadena de suministro, Equipo Evidence, 2021, evidencetec.com

2.2.1. Pedidos y Distribución

La Gestión de Pedidos y Distribución consiste en actividades que resultan de la cumplimentación de órdenes de pedido del cliente, a la vez que se asegura el máximo valor de la cadena de suministro y servicio al cliente.

Un pedido se puede definir como un compromiso en firme entre dos partes (proveedor y cliente) que reúne todas las condiciones mínimas necesarias para establecer una relación comercial entre ellas de manera que una de las partes (el proveedor) pone a disposición de la otra (cliente) los productos o servicios comprometidos, bajo las condiciones pactadas.

Cabe destacar su distinción respecto a la intención de compra, la cual no implica compromiso firme entre ambas partes y que, por esa cualidad, no forma parte de la Gestión de Pedidos y Distribución, aunque sí para la realización de previsiones; es decir, una intención de compra es un pedido en estado potencial

Así, el proceso comienza con la llegada de un pedido y termina cuando el pedido es enviado, aceptado y finalmente cobrado. Es importante tener en mente que la Gestión de Pedidos y Distribución abarca todos los pedidos que lleguen a la compañía, bien sean externos (lo más habitual) o internos (pedidos entre plantas, almacenes o los que realiza producción). (Pilot, 2015)

2.2.2. Administración de Almacenes

La gestión de almacenes se define como el proceso de la función logística que trata la recepción, almacenamiento y movimiento dentro de un mismo almacén, hasta el punto de consumo de cualquier material (materias primas), semielaborados, terminados, así como el tratamiento e información de los datos generados.

La gestión de almacenes tiene como objetivo optimizar un área logística funcional, que actúa en dos etapas de flujo como lo son el abastecimiento y la distribución física, constituyendo por ende la gestión de una de las actividades más importantes para el funcionamiento de una organización.

El objetivo general de una gestión de almacenes consiste en garantizar el suministro continuo y oportuno de los materiales y medios de producción requeridos para asegurar los servicios de forma ininterrumpida y rítmica. La gestión de almacenes se sitúa en el mapa de procesos logísticos entre la gestión de existencias y el proceso de gestión de pedidos y distribución.

De esta manera, el ámbito de responsabilidad (en cuya ampliación recae la evolución conceptual del almacenamiento) del área de almacenes, nace en la recepción de la unidad física en las propias instalaciones y se extiende hasta el mantenimiento del mismo en las mejores condiciones para su posterior tratamiento. (López, 2019)

2.3. ENTRADAS Y SALIDAS DE ALMACÉN

Los registros de entradas y salidas en un almacén son una parte importante de la gestión de inventario, estos registros ayudan a los administradores o gerentes de almacén a mantener un seguimiento de los productos que entran y salen del almacén, lo que les permite asegurarse de que el inventario está al día y que los productos están disponibles para los respectivos clientes.

Ayudan a identificar los productos que están siendo más y menos vendidos, esto sirve para ajustar el inventario para satisfacer las necesidades de los clientes, también sirve para saber si los productos están siendo mal manejados o mal almacenados, lo que permite tomar medidas para mejorar el manejo de los productos. Los registros de entradas y salidas son muy útiles de igual manera para identificar si los productos están siendo robados o mal utilizados, esto definitivamente ayuda a tomar medidas para prevenir el robo y mejorar la seguridad del almacén.

2.3.1. Entradas en el Inventario

Las entradas en el inventario de un almacén generalmente incluyen información como el nombre del producto o material, la cantidad almacenada, el precio unitario, el costo total, la fecha de entrada y la fecha de vencimiento; esta información se utiliza para llevar un seguimiento de los productos y materiales almacenados, así como para determinar cuándo se necesitan reponer.

Las entradas en el inventario también se utilizan para calcular el costo de los productos y materiales almacenados, esto se hace multiplicando la cantidad almacenada por el precio unitario; esta información se utiliza para determinar el costo total y para calcular el costo promedio de los productos y materiales almacenados. Las entradas en el inventario también se utilizan para llevar un seguimiento de los movimientos de los productos y materiales almacenados.

Esto incluye el seguimiento de los productos y materiales que entran y salen del almacén, así como el seguimiento de los productos y materiales que se transfieren entre almacenes; esta información se utiliza para determinar cuándo se necesitan reponer los productos y materiales almacenados.

2.3.2. Salidas en el Inventario

Las salidas en el almacén son movimientos de mercancías que se realizan para abastecer a los clientes y se pueden clasificar en dos tipos principales: salidas de materiales para la producción y salidas de productos terminados para la venta.

Las salidas en el inventario de un almacén, son un registro de los productos y materiales que se almacenan en el almacén; estas salidas se utilizan para llevar un seguimiento de los productos y materiales en existencia, así como para determinar cuándo se necesitan reponer; las salidas en el inventario también se utilizan para calcular el costo de los productos y materiales almacenados.

Las salidas en un inventario son los productos o materiales que se han vendido, entregado o utilizado durante un período de tiempo determinado, estas salidas se registran en el inventario para mantener un registro de los productos o materiales que se han utilizado y para calcular el costo de los productos o materiales que se han vendido. (Yano, 2023)

2.4. CONTROL DE INVENTARIO

El control de inventarios, es un sistema que permite que una empresa gestione las existencias que almacena; de esta forma, además de saber qué tiene, identifica cuáles productos debe mover más rápido, cuáles son los que escasean, cómo es su rotación y en cuáles invierte más recursos para su correcto almacenaje. El inventario es fundamental para cualquier organización, todas las empresas deben implementar este proceso, aunque al inicio sea una tarea que requiere tiempo.

El control de inventarios, ayuda a mantener un balance en las existencias de un almacén y los productos o artículos que tienen mayor demanda. Puede reducir costos porque refleja oportunamente lo que no tiene una rotación saludable y lo que debe surtirse lo más pronto posible; esto evita retrasos en los pedidos o que se estropeen materias primas por un almacenamiento deficiente o prolongado. (Rodriguez, 2023)

2.4.1. Sistema de Inventario

Un sistema de inventario es el conjunto de normas, métodos y procedimientos que se utiliza para planificar y controlar los productos o materiales que utiliza una empresa, de manera que esta pueda funcionar eficazmente. Este sistema permite conocer la cantidad de artículos, estimar cuándo hay que reabastecerlos y conciliar las existencias físicas con las registradas en la documentación.

Las empresas que almacenan mercancías necesitan llevar un correcto y exhaustivo control de inventarios; un sistema de inventario eficiente permite una gestión de manera adecuada de las mercancías, evita las pérdidas por daños, reduce los costes de almacenamiento, optimiza el ciclo de compra y la planificación del flujo de caja, además de ayudar a identificar nuevas oportunidades de negocios.

2.4.1.1. Tipos de Sistemas

Las empresas pueden utilizar diferentes tipos de sistemas de control de inventarios; según la naturaleza de la mercancía, se puede hacer referencia a inventarios de materia prima, de productos en proceso o productos terminados. Los sistemas de control de inventarios también se clasifican según el proceso logístico; el inventario en existencia se refiere a los productos que se encuentran en almacén, mientras que el inventario en tránsito contabiliza los productos que se están moviendo en la red logística.

Los sistemas de control basados en la funcionalidad son muy útiles, ya que se realiza un inventario normal para asegurar la demanda de los productos y un inventario de seguridad para cubrir las fluctuaciones de la demanda y posibles problemas de suministro; al final se hace el inventario disponible, que incluye todas las existencias en almacén. (Eserp, 2021)

2.4.2. Método de Control

Para controlar el inventario de cilindros de oxígeno en un laboratorio de manera efectiva, es crucial implementar un método de control para garantizar que siempre haya suficiente suministro disponible para las actividades diarias, evitando interrupciones y optimizando el uso de recursos. Por lo tanto, se puede considerar los siguientes pasos:

- 1) **Inventario Inicial:** Comenzar por realizar un inventario inicial detallado de todos los cilindros de oxígeno disponibles en el laboratorio. Registrar el número de identificación único de cada cilindro y su capacidad de almacenamiento.
- 2) **Registro y Etiquetado:** Etiquetar cada cilindro de manera única y llevar un registro actualizado de todos los cilindros en existencia. Utilizar un sistema de código de barras, códigos QR o simplemente un registro manual en una hoja de cálculo, asegurando de incluir la capacidad del cilindro y la fecha de última recarga.

- 3) Monitoreo de Entradas y Salidas: Implementar un sistema para registrar todas las entradas y salidas de cilindros de oxígeno del laboratorio. Cada vez que se distribuya o se devuelva un cilindro, registrar la fecha, el número de cilindro, el propósito y quién autorizó la salida.
- 4) Alertas de Reordenamiento: Establecer niveles de inventario mínimo y máximo. Cuando el inventario de cilindros de oxígeno caiga por debajo del nivel mínimo establecido, generar una alerta para iniciar el proceso de reordenamiento o recarga. Esto puede ser manual o automatizado a través de software de gestión de inventarios.
- 5) Auditorías Regulares: Realizar auditorías periódicas del inventario para verificar la precisión de los registros. Esto incluye comparar los registros físicos con los registros en el sistema de inventario para identificar discrepancias y corregirlas de inmediato.
- 6) Seguridad y Mantenimiento: Implementar medidas de seguridad adecuadas para el manejo de cilindros de oxígeno, esto incluye almacenarlos en áreas bien ventiladas y seguras para prevenir riesgos potenciales. Lo cual garantiza que estén en condiciones óptimas de funcionamiento y prolonga su vida útil.
- 7) Capacitación y Procedimientos: Capacitar al personal sobre los procedimientos adecuados para manejar y utilizar los cilindros de oxígeno de manera segura. Esto incluye cómo leer las etiquetas de los cilindros y cómo registrar adecuadamente las entradas y salidas.
- 8) Revisión y Mejora Continua: Regularmente revisar el método de control de inventario y realizar ajustes según sea necesario para mejorar la eficiencia y la precisión. Incorporar retroalimentación del personal involucrado en el manejo diario de los cilindros.

2.5. CILINDROS DE OXÍGENO

Un cilindro de oxígeno (también llamado tanque o botellón de oxígeno), es un recipiente de almacenaje de oxígeno, tanto bajo presión en cilindros de gas o como oxígeno líquido en tanques de almacenaje criogénicos, que son utilizados para almacenar gas para distintos propósitos. (Diaz, 2023)

2.5.1. Oxígeno

Es el gas más importante para los seres vivos, sin él no sería posible la vida vegetal ni animal, se encuentra en el aire que respiramos en menor proporción que el nitrógeno (21% oxígeno, 78% nitrógeno y 1% argón, incluidos gases raros). El oxígeno es un gas incoloro, inodoro e insípido; no es inflamable, pero favorece la combustión, cualquier material se quema mucho más activamente en una atmósfera rica en oxígeno.

La forma de suministro de oxígeno a un establecimiento de atención médica es definida por el perfil del consumo, diario, semanal y mensual. Estos parámetros son utilizados por los fabricantes y distribuidores de gases medicinales, para definir y proponer al Responsable Farmacéutico del establecimiento, la forma óptima de suministro de gases. (Olmos, 2016)

2.5.1.1. Oxígeno Medicinal

El oxígeno medicinal, es una mezcla de gases que contiene no menos de 90% ni más de 96% de oxígeno, es recetado a aquellos pacientes cuyo padecimiento se refleja en bajas concentraciones de oxígeno en la sangre, por lo que requieren la oxigenación suplementaria. El oxígeno medicinal que se almacena en tanques, no debe ser tratado con ningún compuesto tóxico o irritante para el sistema respiratorio de los pacientes usuarios.

Los tanques son estrictamente especiales para su uso con oxígeno medicinal y son diferentes a los tanques usados para oxígeno industrial y cualquier otro tipo de gas, pues el oxígeno medicinal debe de ser pre filtrado para que esté limpio y se encuentre libre de restos de otros gases, vapores y bacterias; mientras que el oxígeno industrial no cuenta con este proceso dentro de su presurización en los tanques. Evitar ante cualquier circunstancia y por seguridad llenar otro tanque con oxígeno medicinal o algún tanque medicinal con cualquier gas industrial para evitar accidentes o afectaciones. (García J. , 2022)

2.5.2. Cilindros

Este tipo de cilindro está diseñado para contener oxígeno a altas presiones en forma de gas comprimido. Son fabricados en acero al carbón o aluminio de una sola pieza y están diseñados para soportar altas presiones, tienen una válvula específica de acuerdo al gas que contienen, dicha válvula se protege con un capuchón o caperuza protectora. Los cilindros que contienen oxígeno se identifican por el color verde en el hombro (en ocasiones también son de color blanco), así como por las etiquetas con la descripción de su contenido; además, se especifican grabando líneas de golpe en el cuerpo las características propias del cilindro: fecha de la prueba hidráulica, fecha de fabricación y el número de serie.

Algunos cilindros tienen una cruz de color roja que indica que el contenido es de calidad medicinal y no debe utilizarse en ninguna otra aplicación. Por lo general para suministrar oxígeno a los establecimientos médicos se utilizan cilindros de 6 a 8 m³, con presiones que fluctúan entre 150 a 200 kg/cm². (Olmos, 2016)

Figura 2.2.

Características Principales de un Cilindro



Nota. Características del cilindro para oxígeno, Praxair, 2012, www.praxair.co.in

- 1) Caperuza o capuchón protector de la válvula.
- 2) Etiqueta de identificación del producto que especifica las características del gas.
- 3) Color verde de identificación del producto en el hombro del cilindro.
- 4) La cruz en color rojo indica que el contenido es de grado medicinal.
- 5) Válvula tipo volante.

2.6. CÓDIGO QR

Un código QR (Quick Response Code, código de respuesta rápida), es un método de representación y almacenamiento de información en una matriz de puntos bidimensional. Esta simbología 2D tiene su origen en 1994 en Japón, cuando la empresa Denso Wave subsidiaria de Toyota, la desarrolla para mejorar la trazabilidad del proceso de fabricación de vehículos.

Fue diseñada con el objetivo principal de conseguir una decodificación sencilla y rápida de la información contenida. Muy comunes en Japón y cada vez más extendidos a nivel mundial (gracias a su empleo para codificar URL's de internet y a las aplicaciones de decodificación existentes para teléfonos móviles con cámara), se caracterizan por disponer de 3 cuadrados en las esquinas, que facilitan el proceso de lectura.

2.6.1. Ventajas y Características

Un código QR consiste en un conjunto de puntos negros (u oscuros), ubicados según una determinada codificación en un patrón cuadrado sobre fondo blanco (o claro). Sus características y ventajas principales se muestran a continuación:

- Alta capacidad de codificación de datos; hasta 7.089 caracteres numéricos o 2.953 bytes.
- Decodificación sencilla y a alta velocidad; desde lectores hardware o aplicaciones software.
- Mayor densidad de datos y poco espacio necesario para impresión del código; en torno a 1/10 respecto al código de barras tradicional.

- Soporte de múltiples lenguajes y códigos de caracteres; numéricos, alfanuméricos, binarios o cualquier formato de datos mediante la definición de extensiones.
- Adaptabilidad del código a los datos; tamaño en puntos de la matriz según contenido almacenado.
- Permite otras variantes; como Micro QR o hasta 16 estructuras añadidas.
- Capacidad de corrección de errores; restauración de hasta un 30% de los datos.
- Aplicación de máscaras a los datos; mayor diferenciación de niveles claros y oscuros.
- Facilidad de lectura del código; independencia de la orientación (decodificación en 360°), detección de distorsión, inversión de umbrales, estructuras en espejo.
- Confidencialidad; facilidad de cifrado del código QR.
- Popularización de uso gracias a diversos factores; publicación de especificaciones, gratuidad de uso, integración con dispositivos móviles, aplicación fuera del entorno industrial, robustez, etc.

2.6.2. Estructura

La representación bidimensional de un código QR se denomina símbolo, cada símbolo está formado por cuadros negros o blancos llamados módulos, que representan el 0 y el 1 binario respectivamente. Los módulos están ubicados en una estructura cuadrada, que contiene dos grandes bloques de módulos: los patrones de función y la región de codificación.

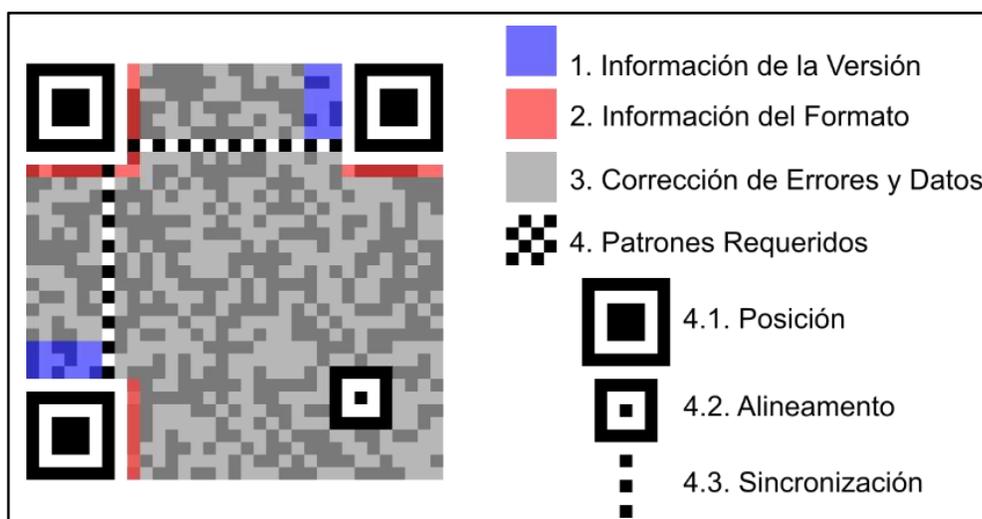
En cada símbolo existen un conjunto de módulos que no contienen datos codificados, sino información necesaria para su decodificación; son los denominados patrones de función y existen de varios tipos:

- Patrón de localización: Patrón de función que existe por triplicado en el símbolo, situado en las esquinas superiores y la inferior izquierda. Sirven para calcular la orientación rotacional del símbolo.

- Patrón de alineamiento: Secuencia alternada de módulos blancos y negros que ayuda a calcular las coordenadas de los módulos del símbolo.
- Patrón temporizador: Patrón de función que permite resincronizar las coordenadas de mapeo del símbolo ante posibles distorsiones moderadas.
- Separador: Patrón de función formado por módulos blancos, cuyo ancho es de un módulo y que separa los patrones localizadores del resto del símbolo.

Figura 2.3.

Estructura de un Código QR



Nota. Funcionamiento de un código QR, Raúl García, 2020, www.tuexperto.com

Los datos codificados, por su parte, se agrupan en conjuntos de 8, denominados codewords, que adoptan diversas formas según su ubicación en la estructura; la región de codificación es la región del símbolo no ocupada por patrones de función y sí por codewords de datos y de corrección de errores, así como por la información de formato y versión.

La información de formato es un patrón codificado, que contiene información sobre el grado de corrección de errores con el que se han codificado los datos de la región de codificación y el tipo de máscara que se les ha aplicado. (Ordóñez, 2012)

2.6.3. Funciones

Como pueden almacenar diferentes tipos de información, los códigos QR se utilizan para muchos propósitos. Las funciones más usadas son:

- URL: abre una página web en un navegador.
- vCard: crea un nuevo contacto.
- Localización geográfica: muestra una localización en Google Maps.
- Texto: muestra un texto estático.
- E-mail: envía un correo electrónico con información.
- SMS: envía un SMS con información.
- WiFi: se conecta a una red wifi.
- Bitcoin: realiza una transferencia de criptomoneda.
- Twitter: redirige a un perfil de Twitter o publica un texto predeterminado.
- Facebook: redirige a una página de Facebook.
- PDF: descarga un archivo en formato PDF.
- MP3: reproduce un archivo en formato MP3.
- App Store: abre la App Store para descargar una aplicación.
- Imagen: muestra una imagen.

2.7. INGENIERÍA DE SOFTWARE

La ingeniería del software según Pressman (1982), es una disciplina o área de la informática o ciencias de la computación, que ofrece técnicas y métodos para desarrollar y mantener software de calidad que resuelvan problemas de todo tipo. La creación del software es un proceso intrínsecamente creativo y la Ingeniería del Software trata de sistematizar este proceso con el fin de acotar el riesgo del fracaso en la consecución del objetivo creativo, por medio de técnicas que se han demostrado adecuadas en base a la experiencia previa.

La informática aporta herramientas y procedimientos que se apoyan en la ingeniería de software con el fin de mejorar la calidad de los productos de software, aumentar la productividad y trabajo de los ingenieros desarrolladores de software, facilitar el control del proceso de desarrollo de software y suministrar a los desarrolladores las bases para construir software de alta calidad en una forma eficiente.

El objetivo principal que busca la ingeniería de software es convertir el desarrollo de software en un proceso formal, con resultados predecibles, que permitan obtener un producto final de alta calidad y satisfaga las necesidades y expectativas del cliente. Según Gacitúa (2003), la Ingeniería de Software es un proceso intensivo de conocimiento, que abarca la captura de requerimientos, diseño, desarrollo, prueba, implantación y mantenimiento.

2.7.1. Sistema Web

Los Sistemas Web o también denominado “Aplicaciones Web”, se definen como aplicaciones de software que se pueden usar en un servicio web por medio de internet o de una intranet (red local) desde un navegador, sin la necesidad de que estén creados e instalados sobre una plataforma o sistemas operativos (Windows, Linux). Por otro lado, las aplicaciones web tiene un aspecto similar al de los sitios web, pero tienen funcionalidades muy potentes, ya que los sistemas web trabajan con bases de datos, lo cual se puede procesar y mostrar información de forma dinámica para los usuarios.

Un sistema web es muy utilizado por ser práctico y rápido, además que se emplea en cualquier navegador web (Google Chrome, Mozilla Firefox, Internet Explorer, etc.) sin importar el sistema operativo; de hecho, se puede acceder a los navegadores mencionados a través de un servidor donde se aloje el sistema sin la necesidad de instalarlas en una computadora, laptop, etc. Lo que significa, a que no será necesario en aprender a manejar nuevos programas que puedan ser costosos y logrando trabajar en cualquier lugar donde se encuentre.

Los sistemas desarrollados sobre plataformas online tienen claras diferencias con otro tipo de sistemas, por lo que es de gran utilidad tanto para las empresas que lo utilizan, como para los usuarios que operan en el sistema; las diferencias de este tipo se reflejan en el costo, la velocidad de adquisición de datos, la optimización de las tareas del usuario y el logro de un control estable. (Admin, 2015)

2.8. METODOLOGÍA DE DESARROLLO DE SOFTWARE

Las metodologías de desarrollo de software son un conjunto integrado de técnicas y métodos organizativos, que se aplican para diseñar y desarrollar software informático, permitiendo abordar de forma homogénea y abierta cada una de las actividades del ciclo de vida de un proyecto de desarrollo. El objetivo de las distintas metodologías, es el de intentar organizar los equipos de trabajo para planificar, diseñar, construir, probar y entregar software de alta calidad de manera eficiente y efectiva.

Estas metodologías, son un proceso de software detallado y completo, que establecen una estructura para el ciclo de vida del software, que incluye la definición de requisitos, el diseño, la codificación, la prueba, la implementación y el mantenimiento. También establecen roles y responsabilidades para los miembros del equipo, procesos para la gestión de proyectos, la comunicación y el seguimiento del progreso.

2.8.1. Metodologías Ágiles

Las metodologías ágiles de desarrollo de software, son ampliamente utilizadas debido a su alta flexibilidad y capacidad para adaptarse rápidamente a los cambios. Permiten a los equipos de trabajo ser más productivos y eficientes al tener claras las tareas que deben realizar en cada momento; además, permite adaptar el software a las necesidades que van surgiendo en el camino, lo que facilita construir aplicaciones más funcionales y satisfactorias.

Estas metodologías ágiles se fundamentan en el enfoque incremental, en el cual se añaden nuevas funcionalidades a la aplicación final en cada ciclo de desarrollo; sin embargo, los ciclos son más cortos y rápidos, esto implica que se realicen cambios y mejoras de forma gradual, incorporando pequeñas funcionalidades en lugar de cambios masivos.

Este tipo de metodologías permiten construir equipos de trabajo, los cuales se reúnen periódicamente para compartir las novedades y los avances realizados; de esta manera, se construye y perfecciona progresivamente el producto final. Además, permite una interacción con el cliente, quien puede aportar nuevos requisitos o correcciones a medida que el proyecto avanza, ya que puede observar en tiempo real el progreso realizado. (Santander, 2020)

Tabla 2.1.

Metodologías Tradicionales y Ágiles

METODOLOGÍAS TRADICIONALES	METODOLOGÍAS ÁGILES
Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.	Basadas en heurísticas provenientes de prácticas de producción de código.
Cierta resistencia a los cambios.	Especialmente preparados para cambios durante el proyecto.
Impuestas externamente.	Impuestas internamente (por el equipo).
Proceso mucho más controlado, con numerosas políticas/normas.	Proceso menos controlado, con pocos principios.
El cliente interactúa con el equipo de desarrollo mediante reuniones.	El cliente es parte del equipo de desarrollo.
Más artefactos.	Pocos artefactos.
Más roles.	Pocos roles.
Grupos grandes y posiblemente distribuidos.	Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio.
La arquitectura del software es esencial y se expresa mediante modelos.	Menos énfasis en la arquitectura del software.
Existe un contrato prefijado.	No existe contrato tradicional o al menos es bastante flexible.

Nota. Diferencia entre las metodologías, ArevaloMaria, 2011, arevalomaria.wordpress.com

2.9. PROCESO UNIFICADO ÁGIL

El Proceso Unificado Ágil o Agile Unified Process (AUP) en inglés, fue desarrollado por Scott Ambler en septiembre del 2005. Es una metodología, que se describe de una manera simple y fácil de entender, para el desarrollo de aplicaciones de software de negocio, utilizando técnicas ágiles y conceptos que aún se mantienen válidos en la metodología RUP, por lo tanto, es una versión simplificada del Proceso Unificado Racional (RUP). El AUP aplica técnicas ágiles, incluyendo desarrollo dirigido por pruebas (TDD), modelado ágil, gestión de cambios ágil, y refactorización de base de datos para mejorar la productividad. (Cordero, 2005)

El proceso unificado (Unified Process o UP), es un marco de desarrollo software iterativo e incremental. A menudo es considerado como un proceso altamente ceremonioso porque especifica muchas actividades y artefactos involucrados en el desarrollo de un proyecto software. Dado que es un marco de procesos, puede ser adaptado y la más conocida es RUP (Rational Unified Process) de IBM.

La metodología de desarrollo de software (AUP), se preocupa especialmente de la gestión de riesgos. Propone que aquellos elementos con alto riesgo obtengan prioridad en el proceso de desarrollo y sean abordados en etapas tempranas del mismo; para ello, se crean y mantienen listas identificando los riesgos desde etapas iniciales del proyecto. Especialmente relevante en este sentido, es el desarrollo de prototipos ejecutables durante la base de elaboración del producto, donde se demuestre la validez de la arquitectura para los requisitos clave del producto y que determinan los riesgos técnicos.

El proceso AUP establece un modelo más simple que el que aparece en RUP por lo que reúne en una única disciplina, las disciplinas de modelado de negocio, requisitos, análisis y diseño; el resto de disciplinas (implementación, pruebas, despliegue, gestión de configuración, gestión y entorno) coinciden con las restantes de RUP.

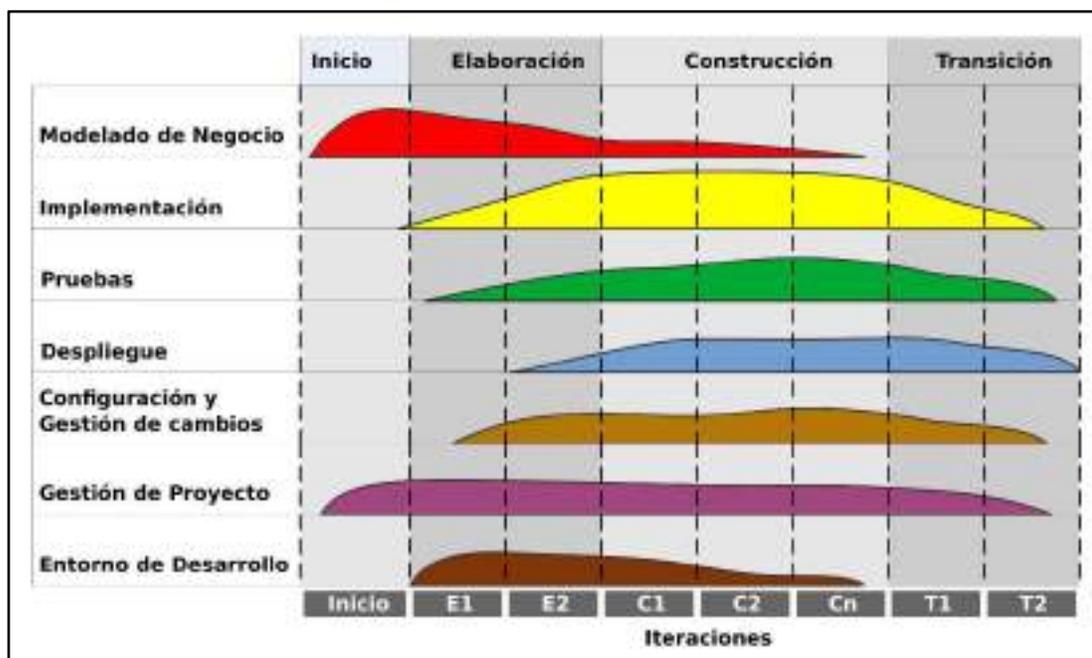
Esta metodología de desarrollo de software plantea un ciclo de vida iterativo, que se basa en la ampliación y refinamiento sucesivo del sistema, mediante múltiples iteraciones con una retroalimentación cíclica y adaptación, como elementos principales que dirigen para converger hacia un sistema adecuado.

2.9.1. Fases o Ciclo de Vida

Al igual que en RUP, en AUP se establecen cuatro fases que transcurren de manera consecutiva y que acaban con hitos claros alcanzados, las fases de desarrollo de software son: *Inicio*, *elaboración*, *construcción* y *transición*. Cada fase se desarrolla mediante iteraciones genéricas; la iteración genérica se refiere a que los cinco flujos de trabajo fundamentales se repiten en cada iteración precedida por la planificación y seguida por la evaluación. (Calle, 2014)

Figura 2.4.

Ciclo de Vida del Proceso Unificado Ágil



Nota. Fases y Disciplinas de la Metodología AUP, Ambler, 2006, www.researchgate.net

2.9.1.1. Iniciación

El principal objetivo es de establecer el modelado del negocio, realizar un estudio de factibilidad del sistema propuesto para después decidir si el sistema es viable o no, identificando todas las entidades externas con las cuales el producto va a interactuar, definiendo si está a un alto nivel, con la descripción de los casos de uso más significativos.

Para la planificación de la fase de inicio se debe empezar por llevar a cabo los siguientes pasos: reunir y organizar la información recogida antes de que el proyecto empiece, descubrir lo que falta para cuando se disponga de suficiente información presentar el plan de desarrollo de la primera iteración, también presentar los criterios de evaluación que indican cuando la iteración a alcanzado los objetivos; estos criterios se basan en decidir el ámbito del sistema, resolver ambigüedades en los requisitos necesarios en esta fase, determinar una arquitectura candidata, mitigar los riesgos críticos.

En esta fase, el desarrollo de una iteración transcurre a través de los flujos de trabajo fundamentales, debiendo avanzar sustancialmente en los componentes de modelado del negocio y captura de requerimientos del sistema. Las características fundamentales que se deben contemplar y ejecutar en esta etapa, serán descritas a continuación.

Objetivo principal:

- Establecer el alcance inicial y la viabilidad del proyecto.

Actividades típicas:

- Identificación del problema a resolver y la visión del proyecto.
- Análisis preliminar de los requisitos.
- Evaluación de la viabilidad técnica, económica y operativa.
- Definición de la arquitectura inicial y la estrategia de desarrollo.

2.9.1.2. Elaboración

Los objetivos principales de esta fase son: conseguir una arquitectura estable para guiar el sistema a lo largo de su vida futura, recopilar la mayor parte de los requisitos que aun queden pendientes, controlar la observación y control de los riesgos críticos y completar los detalles del plan del proyecto en el modelado del negocio si es que faltase completarse.

Esta fase ayuda a planificar con gran precisión la fase de construcción, concentrándose en el análisis y diseño del sistema hasta poder conseguir en las últimas iteraciones algún prototipo que muestre el funcionamiento del sistema, dado que el trabajo de una iteración durante la fase de elaboración a lo largo de los flujos de trabajo es fundamental.

Al final de la fase de elaboración, se empieza a planificar en forma más detallada la primera iteración de la fase de construcción y esbozar en términos más generales las iteraciones restantes. Prácticamente se basa en el establecimiento de un plan de proyecto y una arquitectura correcta del software, profundizando en la comprensión de los requisitos del sistema a través del equipo de desarrollo.

Las características fundamentales que se deben contemplar y ejecutar en esta etapa, serán descritas a continuación.

Objetivo principal:

- Refinar los requisitos y planificar el desarrollo detallado.

Actividades típicas:

- Especificación detallada de requisitos y casos de uso.
- Diseño arquitectónico detallado.
- Prototipado para validar la arquitectura y los requisitos críticos.
- Planificación detallada del proyecto, identificando iteraciones y entregables.

2.9.1.3. Construcción

El objetivo principal de esta fase es la capacidad de operación inicial; es decir, un producto listo para ser distribuido y ser sometido a pruebas. Esta fase requiere un mayor número de iteraciones que las fases anteriores, para cada iteración se selecciona algún caso de uso, se refina su análisis y diseño, se procede a su implementación y pruebas a diario, se empieza con la elaboración de material de apoyo al usuario.

El trabajo de una iteración durante la fase de construcción discurre a lo largo de los flujos de trabajo; aquí se detallan y analizan los requisitos restantes, pero la carga de trabajo en estos dos flujos es relativamente leve o casi nula (análisis y requerimientos), la mayor parte del trabajo está siendo realizado en los dos flujos siguientes (diseño e implementación).

Los productos de esta fase son: el plan de proyecto para la fase de transición, el sistema software ejecutable, todos los artefactos incluyendo el modelo del sistema, la descripción de la arquitectura, el análisis y diseño del negocio que refleja la situación al final de la fase y una versión preliminar del manual del usuario.

Las características fundamentales que se deben contemplar y ejecutar en esta etapa, serán descritas a continuación.

Objetivo principal:

- Desarrollar el software de acuerdo con los requisitos y el diseño establecidos.

Actividades típicas:

- Implementación del código.
- Pruebas unitarias y de integración.
- Revisiones continuas y ajustes en base a retroalimentación.
- Gestión de cambios y versiones.

2.9.1.4. Transición

Esta fase comienza a menudo con la entrega de una versión beta del sistema, mediante la distribución del producto software a una muestra representativa de la comunidad de usuario, la operación en paralelo con el sistema que se va a reemplazar y el entrenamiento de los usuarios y el personal de mantenimiento. El sistema ofrece la confianza suficiente como para operar en el entorno del usuario, el cual debe cumplir los requisitos establecidos en las fases anteriores.

El proyecto recibirá información para determinar si el sistema hace lo que demanda su usuario y el negocio, descubrir riesgos inesperados, anotar problemas no resueltos, encontrar fallos, eliminar ambigüedades, centrarse en áreas en los que los usuarios muestren deficiencias y necesitan información o formación.

El producto de esta fase es el software, documentos legales, módulos del sistema, descripción completa y actualizada de la arquitectura, manuales y materiales de formación. Las características fundamentales que se deben contemplar y ejecutar en esta etapa, serán descritas a continuación.

Objetivo principal:

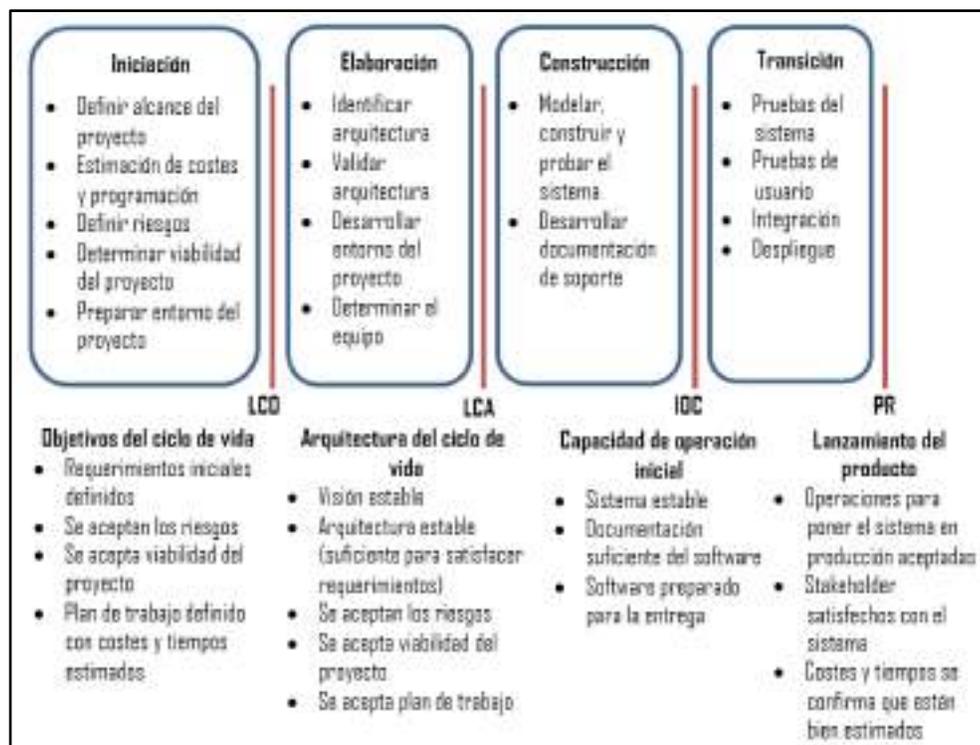
- Preparar el producto para su despliegue y uso por los usuarios finales.

Actividades típicas:

- Pruebas finales (de usuario, de rendimiento, etc.).
- Documentación del usuario y del sistema.
- Entrenamiento del usuario final y soporte inicial.
- Despliegue del software en el entorno de producción.

A continuación, se muestran los hitos por cada fase de la metodología AUP.

Figura 2.5.

Hitos del Proceso Unificado Ágil

Nota. Hitos por cada fase del AUP, Cibertec, 2022, Análisis y Diseño de Sistemas I

2.9.2. Disciplinas

Las disciplinas se realizan de manera iterativa, definiendo las actividades que los miembros del equipo de desarrollo realizan para construir, validar y entregar software de trabajo que satisfaga las necesidades de sus partes interesadas. Las disciplinas o actividades de cada iteración del AUP son las siguientes:

- **Modelo.** Se busca entender el negocio de la organización, el problema de dominio que se abordan en el proyecto y determinar una solución viable para resolver el problema de dominio.
- **Aplicación.** Se transforma su modelo o modelos en código ejecutable y realizar un nivel básico de las pruebas, en particular la unidad de pruebas (Unit testing).

- **Prueba.** Consiste en realizar una evaluación objetiva para garantizar la calidad, esto incluye la búsqueda de defectos, validar que el sistema funciona tal como está establecido y verificando que se cumplan los requisitos.
- **Despliegue.** Es la elaboración de un plan para la entrega del sistema y la ejecución del mismo para que el sistema este a disposición de los usuarios finales.
- **Gestión de configuración.** Procede a gestionar el acceso a los artefactos del proyecto, esto incluye no sólo el seguimiento de las versiones de los artefactos en el tiempo, sino también el control y gestión del cambio para estos.
- **Administración del proyecto.** Dirige las actividades que se llevan a cabo dentro del proyecto, esto incluye la gestión de riesgos, la dirección de personas (la asignación de tareas, el seguimiento de los progresos, etc), coordinación con el personal y los sistemas fuera del alcance del proyecto, para asegurarse de que el software final sea entregado a tiempo y dentro del presupuesto.
- **Entorno.** Es un soporte para el resto de los esfuerzos para garantizar un proceso adecuado, con orientación (normas y directrices) y herramientas (hardware, software, etc), estando disponibles para el equipo según sea necesario.

2.9.3. Principios o Filosofías

AUP es una metodología ágil, porque está basada en los siguientes principios:

- El personal sabe lo que está haciendo. La gente no va a leer de forma detallado la documentación del proceso, pero algunos quieren una orientación de alto nivel y/o formación de vez en cuando. AUP proporciona enlaces a muchos de los detalles, pero no obliga a aquellos que no lo deseen.
- Simplicidad. Todo se describe de forma concisa, utilizando un puñado de páginas y no miles de ellos.

- Agilidad. La metodología se ajuste a los valores y principios de desarrollo de software y la Alianza Ágil.
- Centrarse en actividades de alto valor. La atención se centra en las actividades que se ve que son esenciales para el de desarrollo, no en todas las actividades que suceden o forman parte del proyecto.
- Independencia de herramientas. Se puede usar cualquier conjunto de herramientas que se desee, lo recomendable es utilizar las herramientas que mejor se adapten al trabajo, que a menudo son las herramientas simples o incluso herramientas de código abierto.
- Adaptación. Habrá que adaptar el producto para cumplir con las necesidades propias. La AUP es de fácil acomodo común, a través de cualquier herramienta de edición de HTML; no se necesita comprar una herramienta especial, o tomar un curso para adaptarlo.

2.10. LENGUAJE UML

El Lenguaje Unificado de Modelado (UML), fue creado para forjar un lenguaje de modelado visual común y semántica y sintácticamente rico para la arquitectura, el diseño y la implementación de sistemas de software complejos, tanto en estructura como en comportamiento. En general, los diagramas UML describen los límites, la estructura y el comportamiento del sistema y los objetos que contiene.

UML no es un lenguaje de programación, pero existen herramientas que se pueden usar para generar código en diversos lenguajes usando los diagramas UML, que guarda una relación directa con el análisis y el diseño orientados a objetos. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema; ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados.

Es importante remarcar que UML es un "lenguaje de modelado", para especificar o para describir métodos o procesos; se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir, en otras palabras, es el lenguaje en el que está descrito el modelo. Prácticamente UML es un lenguaje visual en el que se trabaja con cajas, flechas y diagramas. Esto nos permite poder representar las ideas de cómo queremos estructurar nuestros programas de una forma mucho más visual. (Quiróz, 2018)

2.10.1. Tipos de Diagrama

UML usa elementos y los asocia de diferentes formas, para formar diagramas que representan aspectos estáticos o estructurales de un sistema y diagramas de comportamiento, que captan los aspectos dinámicos de un sistema.

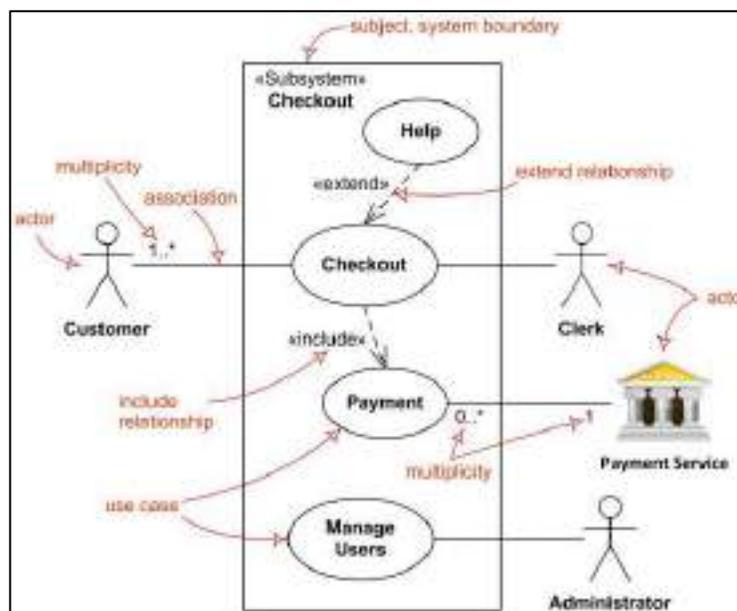
2.10.1.1. Diagramas de Comportamiento

- Diagrama de Caso de Uso: Representa una funcionalidad particular de un sistema, se crea para ilustrar cómo se relacionan las funcionalidades con sus controladores o actores internos/externos.
- Diagramas de Actividades: Flujos de trabajo de negocios u operativos representados gráficamente, para mostrar la actividad de alguna parte o componente del sistema; los diagramas de actividades se usan como una alternativa a los diagramas de estados.
- Diagrama de Estados: Son similar a los diagramas de actividades, que describen el comportamiento de objetos que se comportan de diversas formas en su estado actual.
- Diagrama de Secuencia: Muestra cómo los objetos interactúan entre sí y el orden de la ocurrencia, representan interacciones para un escenario concreto.
- Diagrama de Comunicación: Similar a los diagramas de secuencia, pero el enfoque está en los mensajes que se pasan entre objetos; la misma información se puede representar usando un diagrama de secuencia y objetos diferentes.

- Diagrama de Temporización: Al igual que en los diagramas de secuencia, se representa el comportamiento de los objetos en un período de tiempo dado. Si hay un solo objeto, el diagrama es simple; si hay más de un objeto, las interacciones de los objetos se muestran durante ese período de tiempo particular.

Figura 2.6.

Ejemplo de Diagramas de Comportamiento



Nota. Diagrama de Clases, Antonio Muñoz, 2022, jamj2000.gitbooks.io

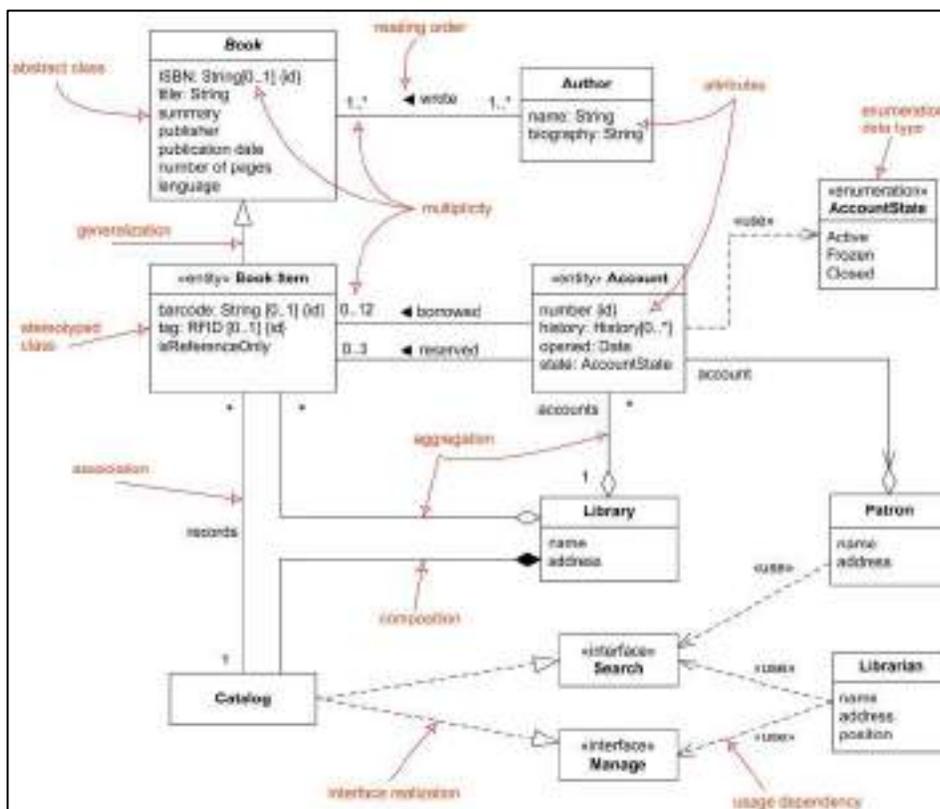
2.10.1.2. Diagramas Estructurales

- Diagrama de Clases: El diagrama UML más comúnmente usado y la base principal de toda solución orientada a objetos. Las clases dentro de un sistema, atributos, operaciones y la relación entre cada clase, las clases se agrupan para crear diagramas de clases al crear diagramas de sistemas grandes.
- Diagrama de Componentes: Muestra la relación estructural de los elementos del sistema de software, muy frecuentemente empleados al trabajar con sistemas complejos con componentes múltiples; los componentes se comunican por medio de interfaces.

- Diagrama de Implementación: Ilustra el hardware y software del sistema; útil cuando se implementa una solución de software, en múltiples máquinas con configuraciones únicas.
- Diagrama de Objetos: Muestra la relación entre objetos con ejemplos del mundo real e ilustra cómo se verá un sistema en un momento dado; como los datos están disponibles dentro de los objetos, estos pueden usarse para clarificar relaciones entre objetos.
- Diagrama de Paquetes: Hay dos tipos especiales de dependencias que se definen entre paquetes: la importación de paquetes y la fusión de paquetes. Los paquetes pueden representar los diferentes niveles de un sistema para revelar la arquitectura, se pueden marcar las dependencias de paquetes para mostrar el mecanismo de comunicación entre niveles. (Muro, 2018)

Figura 2.7.

Ejemplo de Diagramas Estructurales



Nota. Diagrama de Clases, Antonio Muñoz, 2022, jamj2000.gitbooks.io

2.11. HERRAMIENTAS DE DESARROLLO

2.11.1. Base de Datos

Una base de datos o banco de datos, es un conjunto de información perteneciente a un mismo contexto, ordenada de modo sistemático para su posterior recuperación, análisis y/o transmisión. Las bases de datos, son el producto de la necesidad humana de almacenar la información; es decir, de preservarla contra el tiempo y el deterioro, para poder acudir a ella posteriormente. En ese sentido, la aparición de la electrónica y la computación brindó el elemento digital indispensable para almacenar enormes cantidades de datos en espacios físicos limitados, gracias a su conversión en señales eléctricas o magnéticas.

En la conformación de una base de datos se pueden seguir diferentes modelos y paradigmas, cada uno dotado de características, ventajas y dificultades, haciendo énfasis en su estructura organizacional, su jerarquía, su capacidad de transmisión o de interrelación, etc. Esto se conoce como modelos de base de datos y permite el diseño, la implementación de algoritmos y otros mecanismos lógicos de gestión. (Etecé, 2021)

2.11.1.1. Sistema Gestor de Base de Datos

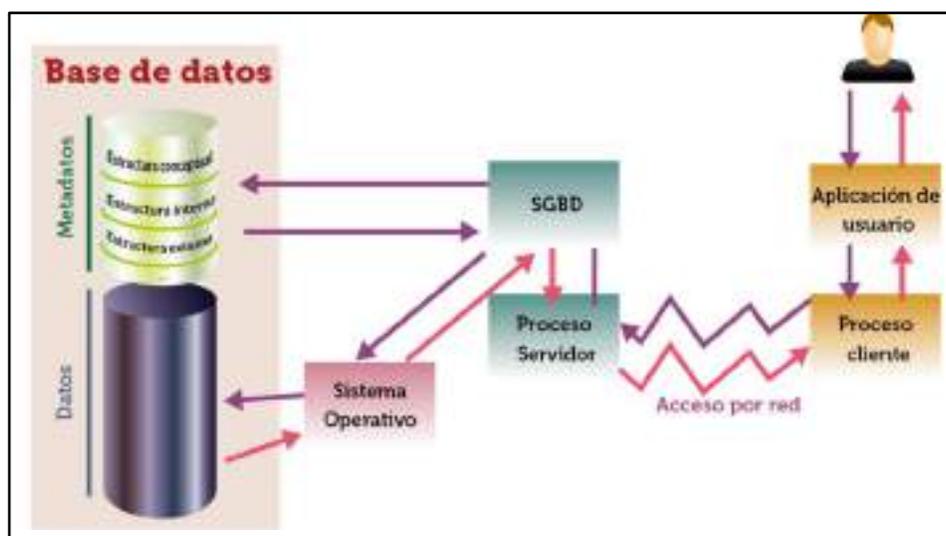
Un sistema gestor de base de datos (SGBD), es un conjunto de programas invisibles para el usuario final con el que se administra y gestiona la información que incluye una base de datos, permitiendo el almacenamiento, modificación y extracción de la información.

Los gestores de datos o gestores de base de datos, son esenciales para cualquier área de negocio, y deben ser gestionados con esmero, ya que permiten administrar todo acceso a la base de datos, pues tienen el objetivo de servir de interfaz entre esta, el usuario y las aplicaciones. Entre sus funciones se encuentran la de permitir a los usuarios almacenar la información, modificar datos y acceder a los activos de conocimiento de la organización.

Así mismo, el gestor de base de datos también se ocupa de realizar consultas y hacer análisis para generar informes; además, los sistemas de gestión de base de datos pueden entenderse como una colección de datos interrelacionados, estructurados y organizados en el ecosistema formado por dicho conjunto de programas que acceden a ellos y facilitan su gestión. (Pérez, 2021)

Figura 2.8.

Sistema Gestor de Base de Datos



Nota. Funcionamiento del SGBD, Jorge Sánchez, 2016, jorgesanchez.net

2.11.2. Gestor MySQL

MySQL, es el sistema de gestión de bases de datos relacional más extendido en la actualidad al estar basada en código abierto, desarrollado originalmente por MySQL AB lanzado en 1995, que fue adquirida por Sun Microsystems en 2008 y está su vez comprada por Oracle Corporation en 2010, la cual ya era dueña de un motor propio InnoDB para MySQL. Es un sistema de base de datos disponible gratuitamente, que cuenta con una doble licencia, es fácil de usar en comparación con otros programas de bases de datos como Microsoft SQL Server y Oracle Database, etc.

Puede ser usado con cualquier lenguaje de programación, pero es ampliamente usado con PHP; además, puede ejecutarse en múltiples plataformas con un esquema de información para definir y administrar sus metadatos. A su vez, puede instalarse en su sistema local o incluso en el servidor, Siendo una solución realmente flexible, escalable, rápida y fiable. (Robledano, 2019)

2.11.2.1. Características

MySQL presenta algunas ventajas, que lo hacen muy interesante y práctico para los programadores y desarrolladores, la más evidente es que trabaja con bases de datos relacionales; es decir, utiliza tablas múltiples que se interconectan entre sí para almacenar la información y organizarla correctamente. Al ser basada en código abierto es fácilmente accesible y la inmensa mayoría de programadores que trabajan en desarrollo web han pasado usar MySQL en alguno de sus proyectos, porque al estar ampliamente extendido cuenta además con una ingente comunidad que ofrece soporte a otros usuarios.

- **Arquitectura Cliente y Servidor:** MySQL basa su funcionamiento en un modelo cliente y servidor; es decir, clientes y servidores se comunican entre sí de manera diferenciada para un mejor rendimiento. Cada cliente puede hacer consultas a través del sistema de registro para obtener datos, modificarlos, guardar estos cambios o establecer nuevas tablas de registros, por ejemplo.
- **Compatibilidad con SQL:** SQL es un lenguaje generalizado dentro de la industria. Al ser un estándar MySQL ofrece plena compatibilidad por lo que si has trabajado en otro motor de bases de datos no tendrás problemas en migrar a MySQL.
- **Vistas:** Desde la versión 5.0 de MySQL, se ofrece compatibilidad para poder configurar vistas personalizadas del mismo modo que podemos hacerlo en otras bases de datos SQL. En bases de datos de gran tamaño las vistas se hacen un recurso imprescindible.

- Procedimientos almacenados: MySQL posee la característica de no procesar las tablas directamente, sino que a través de procedimientos almacenados es posible incrementar la eficacia de nuestra implementación.
- Desencadenantes: MySQL permite además poder automatizar ciertas tareas dentro de nuestra base de datos. En el momento que se produce un evento otro es lanzado para actualizar registros u optimizar su funcionalidad.
- Transacciones: Una transacción representa la actuación de diversas operaciones en la base de datos como un dispositivo, el sistema de base de registros avala que todos los procedimientos se establezcan correctamente o ninguna de ellas; es decir, el sistema opta por preservar la integridad de la base de datos resguardando la información.

2.11.3. Lenguaje de Programación

Un lenguaje de programación es un lenguaje formal (es decir, con reglas gramaticales bien definidas) que le proporciona a una persona la capacidad de escribir o programar una serie de secuencias de órdenes en forma de algoritmos o instrucciones, escritas en una sintaxis que la computadora entiende e interpreta en lenguaje de máquina, con el fin de controlar el comportamiento físico y/o lógico de un ordenador. A todo este conjunto de órdenes escritas mediante un lenguaje de programación se le denomina programa.

Por tanto, programar viene a ser el proceso de crear un software fiable mediante la escritura, prueba, depuración, compilación o interpretación y mantenimiento del código fuente de dicho programa informático.

En palabras simples, es el conjunto de instrucciones a través del cual los humanos interactúan con las computadoras, que permiten procesar de forma rápida y eficientemente grandes y complejas cantidades de información, de manera que se puedan obtener diversas clases de datos o ejecutar determinadas tareas.

2.11.3.1. Características

Para utilizar un lenguaje de programación de manera efectiva, se le debe estudiar y comprender desde tres perspectivas:

- *Sintaxis*: El conjunto de símbolos y reglas para formar sentencias.
- *Semántica*: Las reglas para transformar sentencias en instrucciones lógicas.
- *Pragmática*: Utilizando las construcciones particulares del lenguaje.

En palabras sencillas, las letras forman palabras que forman oraciones; en los lenguajes de programación, los caracteres forman sentencias, que en conjunto forman instrucciones. (Mendoza, 2020)

2.11.4. Lenguaje PHP

PHP (Hypertext Pre-Processor), es un lenguaje de programación de uso general que se adapta especialmente al desarrollo web, favoreciendo la conexión entre los servidores y la interfaz de usuario. Fue creado inicialmente por el programador danés-canadiense Rasmus Lerdorf en 1994, donde originalmente significaba Personal Home Page (Página personal).

El código PHP suele ser procesado en un servidor web por un intérprete PHP, implementado como un módulo o como un ejecutable de interfaz de entrada común; donde el resultado del código PHP interpretado y ejecutado puede ser cualquier tipo de dato, como el HTML generado que formaría la totalidad o parte de una respuesta HTTP.

PHP es un lenguaje de programación destinado a desarrollar aplicaciones para la web y crear páginas web, favoreciendo la conexión entre los servidores y la interfaz de usuario. Existen diversos sistemas de plantillas, sistemas de gestión de contenidos y frameworks que pueden emplearse para organizar o facilitar la generación de esa respuesta.

Entre los factores que hicieron que PHP se volviera tan popular, destaca el hecho de que es de código abierto, no hay restricciones de uso vinculadas a los derechos, el usuario puede usarlo para programar en cualquier proyecto y comercializarlo sin problemas.

PHP generalmente es definido como un lenguaje del lado del servidor, esto significa que se aplica en la programación que tiene lugar en el servidor web, responsable de ejecutar la aplicación o más a menudo en un sitio web. Este trabajo previo permite cargar los elementos de una página antes de mostrarlos al usuario que accede a un sitio web, el código PHP se ejecuta en el servidor que, al leer los comandos, puede activar todos los elementos funcionales y la interfaz visual del sitio web. (Souza, 2020)

2.11.5. CodeIgniter

CodeIgniter es un framework de desarrollo de aplicaciones web de código abierto y basado en PHP, fue creado por la compañía de software norteamericana EllisLab en febrero de 2006, y ahora es mantenido por la comunidad de desarrolladores. CodeIgniter proporciona una estructura ligera y sencilla para desarrollar aplicaciones web rápidas y eficientes, siguiendo el patrón de diseño MVC. Su objetivo es permitirle desarrollar proyectos mucho más rápido de lo que podría hacerlo si estuviera escribiendo código desde cero, proporcionando un amplio conjunto de bibliotecas para tareas comúnmente necesarias, así como una interfaz simple y una estructura lógica para acceder a estas bibliotecas.

CodeIgniter le permite concentrarse creativamente en su proyecto minimizando la cantidad de código necesario para una tarea determinada. Siempre que es posible, se ha mantenido lo más flexible posible, permitiéndole trabajar de la manera que desee, sin verse obligado a trabajar de una manera determinada. La estructura puede tener partes centrales fácilmente ampliadas o reemplazadas por completo para que el sistema funcione como se lo necesite. (CodeIgniter, 2019)

2.11.5.1. Características

Algunos de los puntos más interesantes sobre este framework, en comparación con otros productos similares, son los siguientes:

- **Versatilidad:** En comparación con otros frameworks PHP, es capaz de trabajar la mayoría de los entornos o servidores, incluso en sistemas de alojamiento compartido, donde sólo tenemos un acceso por FTP para enviar los archivos al servidor y donde no tenemos acceso a su configuración.
- **Compatibilidad:** CodeIgniter al menos hasta el momento, es compatible con la versión PHP 4, lo que hace que se pueda utilizar en cualquier servidor, incluso en algunos antiguos; por supuesto, funciona correctamente también en PHP 5.
- **Facilidad de instalación:** No es necesario más que una cuenta de FTP para subir CodeIgniter al servidor y su configuración se realiza con apenas la edición de un archivo, donde debemos escribir cosas como el acceso a la base de datos. Durante la configuración no necesitaremos acceso a herramientas como la línea de comandos, que no suelen estar disponibles en todos los alojamientos.
- **Flexibilidad:** CodeIgniter es bastante menos rígido que otros frameworks; define una manera de trabajar específica, pero en muchos de los casos podemos seguirla o no y sus reglas de codificación muchas veces nos las podemos saltar para trabajar como más a gusto encontremos. Algunos módulos como el uso de plantillas son totalmente opcionales, esto ayuda muchas veces también a que la curva de aprendizaje sea más sencilla al principio.
- **Ligereza:** El núcleo de CodeIgniter es bastante ligero, lo que permite que el servidor no se sobrecargue interpretando o ejecutando grandes porciones de código. La mayoría de los módulos o clases que ofrece se pueden cargar de manera opcional, sólo cuando se van a utilizar realmente.

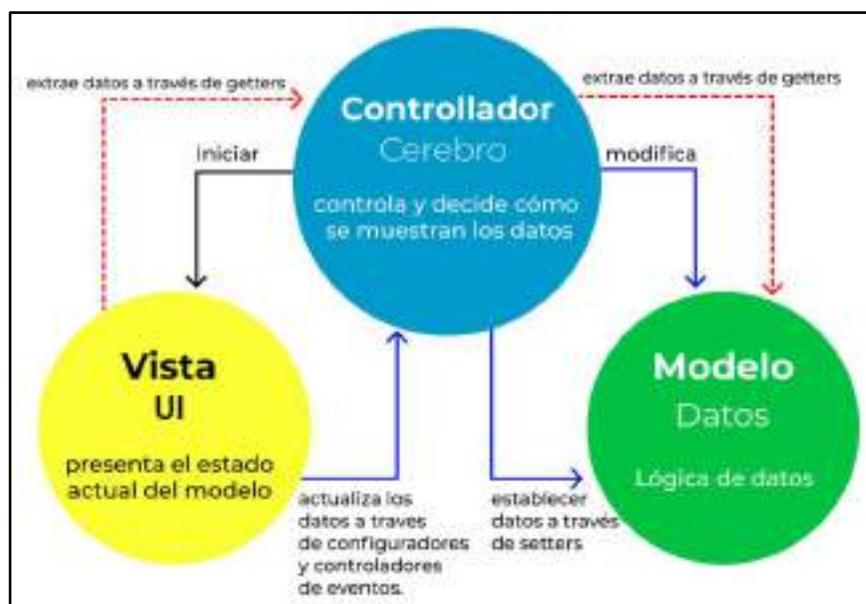
- Extensibilidad: El sistema puede ser fácilmente extendido a través del uso de plugins y librerías asistentes, o a través de extensión de clases o ganchos del sistema.
- Documentación tutorializada: La documentación de CodeIgniter es fácil de seguir y de asimilar, porque está escrita en modo de tutorial, esto nos facilita mucho la referencia rápida para consultar sobre una función o un método en concreto. (Alvarez, 2009)

2.11.5.2. Estructura

El diseño orientado al rendimiento de este framework de desarrollo web se revela en su parca arquitectura, pues se basa en el patrón Modelo – Vista – Controlador. El principio fundamental que sustenta a la arquitectura de desarrollo MVC es la estricta separación entre el código y la presentación, gracias a una estructura modular de software y a la externalización del código PHP. Esta separación se realiza en estos tres grupos: el modelo (model), la vista (view) y el controlador (controller).

Figura 2.9.

Patrones de Arquitectura MVC



Nota. Gráfica de patrones MVC, Rafael Hernandez, 2021, www.freecodecamp.org

- **Modelo:** Representa la estructura de datos de una aplicación web desarrollada con CodeIgniter; para ello, en el código fuente se definen las denominadas clases (“model classes”), que contienen funciones especiales con las cuales se puede recibir, insertar o actualizar la información de la base de datos.
- **Vista:** Es aquello que se le presenta al usuario final; por lo general, se trata de un documento HTML en el cual se ha insertado contenido de forma dinámica con PHP, convirtiéndose en una especie de plantilla. CodeIgniter permite definir fragmentos de una página web o páginas RSS como vista. Normalmente las aplicaciones web utilizan varias vistas, que toman su contenido desde el mismo modelo, de tal forma que es posible presentar diversas características del programa en vistas diferentes.
- **Controlador:** Es el medio entre el modelo, la vista y cualquier otro recurso necesario para procesar una petición HTTP o generar una página web de forma dinámica. Este componente recibe las peticiones entrantes, valida la entrada, selecciona la vista deseada y le entrega el contenido que el modelo ha cargado desde una base de datos.

La estructura MVC permite diseñar software de forma flexible, ya que se pueden substituir, editar y reutilizar los módulos individuales de programación muy fácilmente. Los cambios que se realizan en un componente no suelen tener ningún efecto en el código fuente de otros componentes, siempre y cuando estos cambios no tengan lugar en los puntos de contacto entre unos y otros. La mencionada división permite desarrollar la lógica y el diseño de forma independiente, si los desarrolladores del backend y del frontend trabajan en paralelo, las aplicaciones se pueden terminar mucho más rápidamente.

Aun cuando CodeIgniter utiliza el patrón MVC, los usuarios no están obligados a hacerlo por completo, si el controlador y la vista son componentes obligatorios, no sucede lo mismo con el modelo, cuyo uso como vínculo con la base de datos es meramente opcional. (DesarrolloWeb, 2020)

2.12. MÉTRICAS DE CALIDAD

Las métricas de software se utilizan para estimar el proceso y el esfuerzo de costos o la gestión de la calidad. Independientemente de los beneficios, las métricas de software no se utilizan de forma coherente en la práctica. También se utilizan para definir parámetros de software y procesos de desarrollo; en pocas palabras, son una medida de calidad, porque crean modelos formales de comparación y evaluación en forma de medidas unidimensionales.

Las métricas especiales permiten la evaluación de estructuras y productos de software complejos: idealmente, el proceso de evaluación comienza en la fase de desarrollo, a esto le siguen los estudios métricos de las fases de desarrollo logradas. Si las métricas de software se utilizan generalmente durante la vida útil y el desarrollo posterior de un producto de software, las desviaciones cualitativas de los objetivos se pueden identificar y corregir en una etapa temprana. Los estudios de métricas que se utilizan de forma constante y coherente, ayudan a que los desarrollos de software sean más predecibles. (Garcia J. , 2021)

2.12.1. Estándar ISO/IEC 9126

La ISO, bajo la norma ISO-9126, ha establecido un estándar internacional para la evaluación de la calidad de productos de software el cual fue publicado en 1992 con el nombre de “Information technology – Software product evaluation: Quality characteristics and guidelines for their use”, en el cual se establecen las características de calidad para productos de software. El estándar ISO-9126 establece que cualquier componente de la calidad del software puede ser descrito en términos de una o más de seis características básicas, las cuales son: funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad. Cada una de las cuales se detalla a través de un conjunto de características y subcaracterísticas que debe cumplir todo sistema, ya que permite profundizar en la evaluación de la calidad de productos de software. (Figuroa, 2012)

Previo a la creación de esta norma existían diferentes estándares ISO relacionados con la calidad de software, incluyendo la ISO/IEC 9001 y la ISO/IEC 12207, las cuales buscan la adopción de un enfoque basado en procesos para mejorar la eficacia de un sistema de gestión de calidad y definir los procesos del ciclo de vida del software respectivamente. Sin embargo, estas normas no permitían resolver la dificultad que hay para cuantificar y calificar la mayoría de características que definen un software.

Es allí donde recae la importancia de la norma ISO/IEC 9126, debido a que al basarse en el modelo de McCall, establece una guía acerca de los elementos que deben considerarse al evaluar un software para así generar métricas propias que guíen tanto el desarrollo como su valoración.

2.12.1.1. Características

A continuación, se detalla cada una de las características de calidad que establece el estándar ISO-9126.

a) Funcionalidad

En este grupo se conjunta una serie de atributos que permiten calificar si un producto de software maneja en forma adecuada el conjunto de funciones que satisfagan las necesidades para las cuales fue diseñado. Para este propósito se establecen los siguientes atributos:

- **Adecuación.** Se enfoca a evaluar si el software cuenta con un conjunto de funciones apropiadas para efectuar las tareas que fueron especificadas en su definición.
- **Exactitud.** Este atributo permite evaluar si el software presenta resultados o efectos acordes a las necesidades para las cuales fue creado.

- Interoperabilidad. Permite evaluar la habilidad del software de interactuar con otros sistemas previamente especificados.
- Conformidad. Evalúa si el software se adhiere a estándares, convenciones o regulaciones en leyes y prescripciones similares.
- Seguridad. Se refiere a la habilidad de prevenir el acceso no autorizado, ya sea accidental o premeditado, a los programas y datos.

La métrica punto función (PF) se usa de manera efectiva como medio para medir la funcionalidad que entrega un sistema. PF se deriva empleando una relación empírica basada en medidas contables del dominio de la información del software y la complejidad de este.

Tabla 2.2.

Factores de Ponderación

PARÁMETROS DE MEDICIÓN	CUENTA	FACTORES DE PONDERACIÓN		
		Simple	Medio	Complejo
Entradas de Usuario	X	3	4	6
Salidas de Usuario	X	4	5	7
Peticiones de Usuario	X	3	4	6
Archivos	X	7	10	15
Interfaces Externas	X	5	7	10
<i>Cuenta Total</i>				

Nota. Valores de ajuste de complejidad, Pressman, 2002, Ingeniería de Software

- 1) Número de entradas de usuario. Se cuenta cada entrada de usuario que proporciona diferentes datos orientados a la aplicación. Las entradas se deberían diferenciar de las peticiones, las cuales se cuentan de forma separada.

- 2) Número de salidas de usuario. Se cuenta cada salida que proporciona al usuario información orientada a la aplicación. En este contexto la salida se refiere a informes, pantallas, mensajes de error, etc.
- 3) Número de peticiones de usuario. Una petición se define como una entrada interactiva que produce la generación de alguna respuesta del software inmediata en forma de salida interactiva. Se cuenta cada petición por separado.
- 4) Número de archivos. Se cuenta cada archivo maestro lógico (se refiere a un grupo lógico de datos que puede ser una parte de una gran base de datos o un archivo independiente).
- 5) Número de interfaces externas. Se cuentan todas las interfaces legibles por la máquina (por ejemplo: archivos de datos de cinta o disco) que se utilizan para transmitir información a otro sistema.

Para calcular el Punto Función se utilizará la siguiente fórmula:

$$PF = T * [X + (Min_{(y)} * \sum F_i)] \quad (1)$$

Donde:

- PF : Medida de la funcionalidad.
- T : Nivel de complejidad del sistema con respecto al usuario, es la suma de los valores de entradas, salidas, peticiones, archivos e interfaces externas.
- X : Ajuste de complejidad según el dominio de la información, el grado de confiabilidad estimada varía entre 1 a 100%.
- $Min_{(y)}$: Error mínimo aceptable al de la complejidad, probabilidad subjetiva estimada del dominio de la información, esta tasa de error estimado es del 1%.
- F_i : Son los valores de ajuste de complejidad, donde $(1 \leq i \leq 14)$.

Obteniendo los datos anteriores y considerando un grado de confiabilidad mínimo del 65% es que se calcula el valor de PF .

Así también, considerando el máximo ajuste de la complejidad $\sum F_i = 70$ se calcula al 100% el nivel de confianza como PF_{max}

Con los máximos valores de ajuste de complejidad obtenidos, se puede ya calcular la funcionalidad real del sistema como se ve a continuación.

$$Funcionalidad = \frac{PF}{PF_{max}} \quad (2)$$

b) Confiabilidad

Aquí se agrupan un conjunto de atributos que se refieren a la capacidad del software de mantener su nivel de ejecución bajo condiciones normales en un periodo de tiempo establecido. Las sub características que el estándar sugiere son:

- Nivel de Madurez. Permite medir la frecuencia de falla por errores en el software.
- Tolerancia a fallas. Se refiere a la habilidad de mantener un nivel específico de funcionamiento en caso de fallas del software o de cometer infracciones de su interfaz específica.
- Recuperación. Se refiere a la capacidad de restablecer el nivel de operación y recobrar los datos que hayan sido afectados directamente por una falla, así como al tiempo y el esfuerzo necesarios para lograrlo.

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

$$F_{(t)} = f * e^{(-\lambda*t)} \quad (3)$$

Donde:

- f : Es la funcionalidad del sistema.
- λ : Es la probabilidad de error que puede tener el sistema.
- t : Tiempo que dura una gestión en el sistema.

Para determinar la confiabilidad se toma en cuenta las fallas que puedan ocurrir en un tiempo determinado después su implantación, para medir el tiempo promedio de fallos se calcula bajo la siguiente manera:

$$P(T \leq t) = F(t) \quad (4)$$

$$P(T > t) = 1 - F(t) \quad (5)$$

c) Usabilidad

Consiste de un conjunto de atributos que permiten evaluar el esfuerzo necesario que deberá invertir el usuario para utilizar el sistema.

- **Comprensibilidad.** Se refiere al esfuerzo requerido por los usuarios para reconocer la estructura lógica del sistema y los conceptos relativos a la aplicación del software.
- **Facilidad de Aprender.** Establece atributos del software relativos al esfuerzo que los usuarios deben hacer para aprender a usar la aplicación.
- **Operabilidad.** Agrupa los conceptos que evalúan la operación y el control del sistema.

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

$$FU = [(\sum X_i/n) * 100] \quad (6)$$

Donde:

- X_i : Es la sumatoria de valores.
- n : Es el número de preguntas.

Para ello, se realiza una serie de preguntas que permiten ver cuán sencillo, fácil de aprender y manejar es para los usuarios. Esta comprensión por parte de los usuarios con relación al sistema evalúa los siguientes casos: Entendimiento, aprendizaje, operabilidad, atracción y conformidad de uso.

d) Eficiencia

Esta característica permite evaluar la relación entre el nivel de funcionamiento del software y la cantidad de recursos usados. Los aspectos a evaluar son:

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

Se considera una tabla con una serie de preguntas y sus respectivas evaluaciones, para calcular la eficiencia de un producto de software.

e) Mantenibilidad

La mantenibilidad, se refiere a los atributos que permiten medir el esfuerzo necesario para realizar modificaciones al software, ya sea por la corrección de errores o por el incremento de funcionalidad.

En este caso, se tienen los siguientes factores:

- Capacidad de análisis. Relativo al esfuerzo necesario para diagnosticar las deficiencias o causas de fallas, o para identificar las partes que deberán ser modificadas.
- Capacidad de modificación. Mide el esfuerzo necesario para modificar aspectos del software, remover fallas o adaptar el software para que funcione en un ambiente diferente.
- Estabilidad. Permite evaluar los riesgos de efectos inesperados debidos a las modificaciones realizadas al software.
- Facilidad de Prueba. Se refiere al esfuerzo necesario para validar el software una vez que fue modificado.

Para hallar mantenibilidad del sistema se utiliza el índice de madurez de software (IMS), que proporciona una indicación de la estabilidad de un producto de software.

$$IMS = \frac{Mt - (Fm + Fa + Fe)}{Mt} \quad (7)$$

Donde:

- *Mt*: Número de módulos total de la versión actual.
- *Fm*: Número de módulos de la versión actual que se modificaron.
- *Fa*: Número de módulos de la versión actual que se agregaron.
- *Fe*: Número de módulos en la versión anterior que se eliminaron de la versión actual.

f) Portabilidad

En este caso, se refiere a la habilidad del software de ser transferido de un ambiente a otro, y considera los siguientes aspectos:

- Adaptabilidad. Evalúa la oportunidad para adaptar el software a diferentes ambientes sin necesidad de aplicarle modificaciones.
- Facilidad de Instalación. Es el esfuerzo necesario para instalar el software en un ambiente determinado.
- Conformidad. Permite evaluar si el software se adhiere a estándares o convenciones relativas a portabilidad.
- Capacidad de reemplazo. Se refiere a la oportunidad y el esfuerzo usado en sustituir el software por otro producto con funciones similares.

Para poder medir la portabilidad del sistema se tiene la siguiente fórmula, que indica el grado de portabilidad que tiene un software.

$$GP = 1 - (ET/ER) \quad (8)$$

Donde:

- *ET*: Medida de los recursos necesarios para llevar el sistema a otro entorno.
- *ER*: Medida de los recursos necesarios para crear el sistema en el entorno residente.

2.13. ESTIMACIÓN DE COSTOS

La estimación de los costos de desarrollo de software es un factor muy importante en el análisis de los proyectos informáticos, constituye un tema estratégico contar con indicadores para medir el costo de los mismos, garantizando la eficiencia, excelencia, calidad y la competitividad. El análisis de costo es el proceso de identificación de los recursos necesarios para llevar a cabo el trabajo o proyecto eficientemente.

La evaluación del costo determina la calidad y cantidad de los recursos necesarios en términos de dinero, esfuerzo, capacidad, conocimientos y tiempo incidiendo en la gestión empresarial; existen un conjunto de métricas que pueden ser aplicables a cualquier tipo de proyecto de software para calcular el costo de los mismos. (Roque, 2014)

2.13.1. Modelo COCOMO II

Es un modelo de estimación que se encuentra en la jerarquía de modelos de estimación de software con el nombre de COCOMO por Constructive Cost Model (Modelo Constructivo de Coste), donde el modelo original se ha convertido en uno de los modelos de estimación de coste del software más utilizado y estudiado en la industria.

Es una herramienta basada en las líneas de código, la cual la hace muy poderosa para la estimación de costos y no como otros que solamente miden el esfuerzo en base al tamaño, donde representa el más extenso modelo empírico para la estimación de software. COCOMO II permite realizar estimaciones en función del tamaño del software, y de un conjunto de factores de coste y de escala.

En los factores de coste se incluyen aspectos relacionados con la naturaleza del sistema, equipo, y características propias del proyecto; en los factores de escala incluye la parte de escala producida a medida que un proyecto de software incrementa su tamaño. (Gómez y Migani, 2017)

2.13.1.1. Modelos de Desarrollo

Los modelos de COCOMO II, se adaptan tanto a las necesidades de los diferentes sectores, como al tipo y cantidad de información disponible en cada etapa del ciclo de vida de desarrollo, lo que se conoce por granularidad de la información. Estos tres modelos son:

- 1) Composición de Aplicación. Se emplea en desarrollos de software durante la etapa de prototipado. En este caso, se utiliza puntos objeto para estimar el tamaño del software, lo cual está acorde al nivel de información que se tiene en la etapa de planificación y el nivel de precisión requerido en la estimación de proyectos de esta naturaleza.
- 2) Diseño Temprano. Se utiliza en las primeras etapas del desarrollo, en las cuales se evalúan las alternativas de hardware y software de un proyecto. En estas etapas se tiene poca información, lo que concuerda con el uso de puntos función, para estimar tamaño y el uso de un número reducido de factores de costo.
- 3) Post Arquitectura. Se aplica en la etapa de desarrollo propiamente dicho, cuando la arquitectura del proyecto está completamente definida. Este modelo de estimación es el más detallado y se aplica durante el desarrollo y mantenimiento de productos de software.

2.13.1.2. Métodos

El modelo COCOMO desarrollado por Barry M. Boehm, se engloba en el grupo de los modelos algorítmicos que tratan de establecer una relación matemática la cual permite estimar el esfuerzo y tiempo requerido para desarrollar un producto.

Por un lado, COCOMO II define tres modos de desarrollo o tipos de proyectos:

- 1) Orgánico: Proyectos relativamente sencillos menores de 50 KDLC, en los cuales se tiene experiencia de proyectos similares y se encuentran en entornos estables.
- 2) Semi-Acoplado: Proyectos intermedios en complejidad y tamaño, menores de 300 KDLC, donde la experiencia en este tipo de proyectos es variable, y las restricciones intermedias.
- 3) Empotrado: Proyectos bastante complejos, en los que apenas se tiene experiencia y se engloban en un entorno de gran innovación técnica. Además, se trabaja con unos requisitos muy restrictivos y de gran volatilidad.

Tabla 2.3.

Coefficientes del Modelo COCOMO II

PROYECTO DE SOFTWARE	a	b	c	d
Orgánico	2,4	1,05	2,5	0,38
Semi-Acoplado	3,0	1,12	2,5	0,35
Empotrado	3,6	1,20	2,5	0,32

Nota: Constantes de Complejidad, Pressman, 2005, Ingeniería de Software

Y por otro lado existen diferentes modelos que define COCOMO:

- 1) Modelo Básico: Se basa exclusivamente en el tamaño expresado en LDC.
- 2) Modelo Intermedio: Además del tamaño del programa incluye un conjunto de medidas subjetivas llamadas conductores de costes.
- 3) Modelo Avanzado: Incluye todo lo del modelo intermedio además del impacto de cada conductor de coste en las distintas fases de desarrollo.

De esta manera, es conveniente emplear el modelo intermedio, dado que realiza las estimaciones con bastante precisión.

Las fórmulas de este modelo de COCOMO II son las siguientes:

$$E = a * (KLDC)^b * FAE \quad (9)$$

$$T = c * (E)^d \quad (10)$$

$$NP = E/T \quad (11)$$

Dónde:

- *E*: Esfuerzo requerido por el proyecto expresado en persona/mes.
- *T*: Tiempo requerido por el proyecto expresado en meses.
- *NP*: Número de personas requeridas para el proyecto.
- *a, b, c, d*: Constantes con valores definidos según cada sub-modelo.
- *KLDC*: Cantidad de líneas de código distribuidas en miles.
- *FAE*: Ajuste del factor de complejidad.

Para calcular el Esfuerzo, se necesita hallar la variable KLDC (Kilos de Líneas de Código), para ello se utiliza la siguiente la tabla.

Tabla 2.4.

Factor de Conversión LDC/PF

LENGUAJE	NIVEL	FACTOR LDC/PF
C	2.5	128
Java	6	53
Ansi Cobol	3	107
Visual Basic	7	46
ASP	9	36
PHP	11	29
Visual C++	9.5	34
SQL	12	12

Nota: Factor LCD/PF de lenguajes de programación, QSM, 2017, McLean

De igual manera se debe de hallar la variable FAE, la cual se obtiene mediante la multiplicación de los valores evaluados en los diferentes 15 conductores de coste.

Tabla 2.5.

Valores de Atributo de Costes FAE.

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

Nota: Ajuste del factor de complejidad, Barry Boehm, 1981, COCOMO 81

2.14. SEGURIDAD

La seguridad de la información, es el conjunto de medidas y técnicas utilizadas para controlar y salvaguardar todos los datos que se manejan dentro de una empresa o institución y asegurar que los datos no salgan del sistema que ha establecido la organización. Es una pieza fundamental para que las compañías puedan llevar a cabo sus operaciones sin asumir riesgos, puesto que los datos que se manejan son esenciales para el devenir del negocio; teniendo en cuenta que la seguridad de la información debe hacer frente a estos riesgos, analizarlos, prevenirlos y encontrar soluciones rápidas para eliminarlos si se diera el caso.

De forma mayoritaria, los sistemas de las organizaciones se basan en las nuevas tecnologías, no se deben confundir seguridad de la información y seguridad informática que, si bien están íntimamente relacionadas, no siendo el mismo concepto. Es importante comprender que cualquier organización, independientemente de su tamaño, cuenta con datos confidenciales, bien de sus clientes, bien de sus trabajadores o bien de ambos, y que por ello tiene que establecer las medidas de seguridad en protección de datos necesarios para garantizar el correcto tratamiento de estos. (Ayudaley, 2020)

Partiendo del hecho de que la seguridad de la información puede variar en función de las características de cada organización y del sector al que dedique su actividad económica, se puede hablar de una serie de objetivos comunes que comparten todas las organizaciones del ámbito de la seguridad de la información y la protección de datos. Encontramos estos objetivos de la seguridad de la información en la norma ISO 27001, que proporciona un modelo para la implantación de sistemas de gestión de seguridad de la información (SGSI), cuyo fin principal es la protección de los activos de información, es decir, equipos, usuarios e información. Para establecer este sistema ISO de seguridad de la información hay que tener en cuenta tres aspectos fundamentales: integridad, confidencialidad y disponibilidad.

2.14.1. Triada CID

La triada de seguridad de la información, es un concepto fundamental en el campo de la ciberseguridad y forma la base de cualquier estrategia de seguridad efectiva; compuesta por confidencialidad, integridad y disponibilidad, proporciona un marco sólido para proteger la información en entornos digitales

Estos tres aspectos deben abordarse de manera equilibrada para garantizar una adecuada gestión de la seguridad de la información.

2.14.1.1. Confidencialidad

La confidencialidad se refiere a la protección de la información contra el acceso no autorizado, el objetivo es garantizar que solo las personas autorizadas tengan acceso a la información confidencial. Algunos ejemplos de medidas que garantizan la confidencialidad incluyen:

- **Cifrado de datos:** Utilizar algoritmos de cifrado para proteger los datos sensibles, de modo que solo puedan ser leídos por las personas que tienen la clave de descifrado adecuada.
- **Control de acceso:** Implementar mecanismos como contraseñas, autenticación de dos factores y listas de control de acceso para limitar quién puede acceder a la información confidencial.
- **Políticas de seguridad:** Establecer políticas y procedimientos claros sobre cómo se debe manejar y compartir la información confidencial dentro de una organización.
- **Enmascaramiento de datos:** Es un método que permite presentar información sensible de manera que se mantiene su estructura, pero sin revelar la información original.
- **Clasificación de datos:** Clasificar la información según su nivel de sensibilidad ayuda a determinar qué medidas de seguridad son las más apropiadas para cada tipo de dato.

2.14.1.2. Integridad

La integridad se refiere a la protección de la precisión y la integridad de la información. Implica asegurarse de que los datos no sean alterados de manera no autorizada, garantizando su exactitud y confiabilidad. Algunos ejemplos de medidas que garantizan la integridad incluyen:

- Firmas digitales: Utilizar firmas digitales para verificar la integridad de los datos y asegurar que no hayan sido modificados durante el tránsito o el almacenamiento.
- Registros de auditoría: Mantener registros de auditoría detallados que rastreen los cambios y las actividades relacionadas con la información, lo que facilita la detección de alteraciones no autorizadas.
- Copias de seguridad: Realizar copias de seguridad regulares de la información para garantizar que, en caso de daño o alteración, se pueda restaurar una versión válida y confiable.
- Sistemas de detección de intrusos: Los sistemas de detección de intrusos (IDS) pueden ayudar a identificar cualquier actividad sospechosa que pueda indicar una amenaza a la integridad de los datos.

2.14.1.3. Disponibilidad

La disponibilidad se refiere a asegurar que la información esté disponible y accesible cuando sea necesario, esto implica garantizar que los sistemas y los datos estén disponibles y funcionando correctamente para los usuarios autorizados. Algunos ejemplos de medidas que garantizan la disponibilidad incluyen:

- Redundancia y respaldo: Establecer sistemas redundantes y realizar copias de seguridad para garantizar la disponibilidad de los datos en caso de fallas o pérdidas.

- **Mantenimiento y actualización:** Realizar un mantenimiento regular de los sistemas y asegurarse de que los parches y actualizaciones de seguridad estén al día para evitar interrupciones debido a vulnerabilidades conocidas.
- **Monitoreo de la red:** El monitoreo regular de la red puede ayudar a identificar y solucionar los problemas antes de que afecten a la disponibilidad de los datos.
- **Planes de continuidad del negocio:** Desarrollar planes y procedimientos para garantizar la disponibilidad de los sistemas y la información en caso de interrupciones, como desastres naturales o fallas del sistema. (Escalante, 2023)

2.14.2. Norma ISO 27001

ISO 27001 es una norma internacional emitida por la Organización Internacional de Normalización (ISO), esta se basa en la teoría de gestión de calidad o por sus siglas en inglés PDCA (Plan, Do, Check, Act) “Planificar, Hacer, Verificar, Actuar” y describe cómo gestionar la seguridad de la información en una empresa para la mejora continua de los sistemas de información y garantizar la ciberseguridad de los activos de información. La filosofía que rige a esta norma es la investigación de los riesgos para la futura creación de un plan de tratamiento adecuado.

La norma ISO 27001 es un estándar internacional que establece los requisitos para la implementación, mantenimiento y mejora continua de un Sistema de Gestión de la Seguridad de la Información (SGSI). Este sistema se utiliza para proteger la confidencialidad, integridad y disponibilidad de la información; la norma proporciona un marco para la seguridad de la información que ayuda a las organizaciones a identificar y gestionar sus riesgos de seguridad de la información de manera efectiva. Los requisitos de la norma son de carácter genérico y están destinados a ser aplicables a todas las organizaciones, independientemente de su tipo, tamaño o naturaleza.

Al igual que otros estándares de sistemas de gestión ISO, la certificación ISO/IEC 27001 es posible pero no obligatoria y puede ser implementada por cualquier tipo de organización, con o sin fines de lucro. Algunas organizaciones eligen implementar el estándar para beneficiarse de las mejores prácticas que contiene, mientras que otras deciden que también desean obtener la certificación para asegurar a los clientes que se han seguido sus recomendaciones, pues esto conlleva ciertos beneficios.

2.14.2.1. Estructura

- Objeto y campo de aplicación. La norma comienza aportando orientaciones sobre el uso, finalidad y modo de aplicación de este estándar.
- Referencias normativas. Son los documentos indispensables para la aplicación de ISO 27001, entre ellas se incluye la declaración de aplicabilidad.
- Términos y definiciones. Es un glosario que describe la terminología aplicable a este estándar, lo que facilita la comprensión de la norma.
- Contexto de la organización. Recoge indicaciones sobre el conocimiento de la organización y su contexto, luego plantea la comprensión de las necesidades y expectativas de las partes interesadas, así como también la determinación del alcance del SGSI; este es uno de los requisitos más importantes de la norma.
- Planificación. Se deben establecer objetivos de seguridad de la información y el modo de lograrlos; es decir, un plan de tratamiento de riesgos.
- Soporte. En esta cláusula la norma señala que para el buen funcionamiento del SGSI la organización debe contar con los recursos, competencias, conciencia, comunicación e información documentada pertinente en cada caso.
- Operación. Esta parte de la norma indica que se debe planificar, implementar y controlar los procesos de la organización. También se debe hacer una valoración de los riesgos de la seguridad de la información y un tratamiento de ellos.

- Liderazgo. Aquí se destaca la necesidad de que los empleados de la organización han de contribuir al establecimiento de la norma. Para ello, la alta dirección ha de demostrar su liderazgo y compromiso con la elaboración de una política de seguridad que conozca la organización y deberá asignar roles, responsabilidades y autoridades dentro de la misma.
- Evaluación del desempeño. En este punto se establece la necesidad y forma de llevar a cabo el seguimiento, la medición, el análisis, la evaluación, la auditoría interna y la revisión por la dirección del Sistema de Gestión de Seguridad de la Información; esto para asegurar que funciona según lo planificado.
- Mejora. Por último, encontramos las obligaciones que tendrá una organización cuando encuentre una inconformidad y la importancia de mejorar continuamente la conveniencia, adecuación y eficacia del SGSI. (Martinez, 2022)

2.15. PRUEBAS DE SOFTWARE

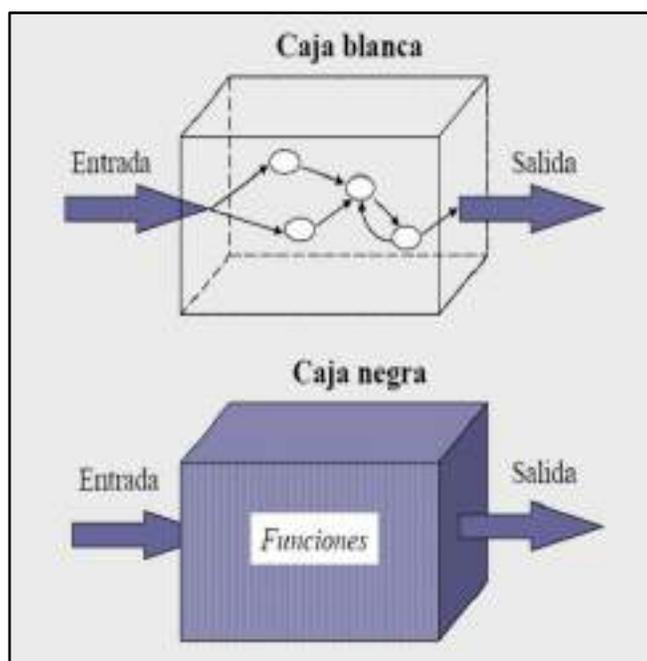
Las pruebas de software son las investigaciones empíricas y técnicas, cuyo objetivo es proporcionar información objetiva e independiente sobre la calidad del producto a la parte interesada o stakeholder. Es una actividad más en el proceso de control de calidad, para evaluar y verificar que un producto o aplicación de software haga lo que se supone que debe hacer; los beneficios de las pruebas incluyen la prevención de errores, la reducción de los costos de desarrollo y la mejora del rendimiento.

Las pruebas de software son una parte integral del ciclo de vida del desarrollo de software (SDLC), son la forma en que se puede estar seguro acerca de la funcionalidad, el rendimiento y la experiencia de usuario. Ya sea que realice sus pruebas manualmente o mediante automatización, cuanto antes y con mayor frecuencia se realicen, más probable es que se identifique errores, asegurando que la aplicación de software haya sido revisada y auditada exhaustivamente antes de que esté frente a los usuarios.

Las pruebas de software se pueden dividir en dos tipos diferentes: pruebas funcionales y no funcionales. Diferentes aspectos de una aplicación de software requieren diferentes tipos de pruebas, cada uno de estos ofrece una excelente visibilidad de la aplicación, desde el código hasta la experiencia del usuario. (Técnicos, 2020)

Figura 2.10.

Técnicas de Prueba



Nota. Pruebas de Caja Blanca y Negra, Antonio Muñoz, 2023, www.pmoinformatica.com

2.15.1. Caja Blanca

La prueba de caja blanca, es una técnica de prueba de software que consiste en probar el código y el diseño interno del software para verificar el flujo de entrada - salida y comprobar el diseño, la usabilidad y la seguridad del software; permiten inspeccionar el funcionamiento interno del sistema, al mismo tiempo que verifican que las entradas dan lugar a salidas específicas y esperadas. La caja blanca es un paso esencial en las pruebas de software, porque es la única que tiene en cuenta cómo funciona el propio código. (Chernyak, 2023)

Las pruebas de caja blanca, aprovechan el amplio conocimiento de los aspectos internos de una aplicación para desarrollar casos de prueba altamente específicos. Los ejemplos de pruebas que se pueden realizar durante las pruebas de caja blanca incluyen:

- Comprobación de ruta
- Validación de salida
- Pruebas de seguridad
- Prueba de bucle
- Prueba de flujo de datos

2.15.1.1. Prueba de Camino Básico.

Esta técnica fue propuesta por Thomas McCabe en el año 1976 y consiste en definir un conjunto básico de caminos usando la medida de complejidad llamada complejidad ciclomática (VG). La complejidad ciclomática determina el número de caminos a probar, mediante la siguiente fórmula:

$$V(G) = \text{Aristas} - \text{Nodos} + 2 \quad (12)$$

Un grafo G se define como un par (V, E), donde V es un conjunto cuyos elementos son denominados vértices o nodos y E es un subconjunto de pares no ordenados de vértices y que reciben el nombre de aristas o arcos.

Los pasos en las pruebas de caminos básicos son:

- 1) Dibujar grafo de flujo.
- 2) Determinar la complejidad ciclomática del grafo.
- 3) Determinar los caminos linealmente independientes.
- 4) Diseñar los casos de prueba.

2.15.1.2. Tipos de Pruebas

La prueba de caja blanca se puede realizar para varios propósitos diferentes, los tres tipos de pruebas de caja blanca son:

- **Pruebas unitarias:** Las pruebas unitarias están diseñadas para garantizar que cada componente o función de una aplicación funcione correctamente, esto ayuda a garantizar que la aplicación cumpla con los requisitos de diseño durante todo el proceso de desarrollo.
- **Pruebas de integración:** Las pruebas de integración se centran en las interfaces entre los distintos componentes dentro de una aplicación. Realizado después de la prueba unitaria, garantiza que no solo cada componente funcione bien de forma aislada, sino también que puedan trabajar juntos de manera efectiva.
- **Pruebas de regresión:** Los cambios pueden dañar cosas dentro de una aplicación. Garantizan que el código siga superando los casos de prueba existentes después de realizar actualizaciones de funcionalidad o seguridad en una aplicación. (CRT, 2011)

2.15.2. Caja Negra

Las pruebas de Caja Negra, también conocidas como pruebas de comportamiento, 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. Se enfoca solamente en las entradas y salidas del sistema, sin preocuparse de la estructura interna del programa; para obtener el detalle de cuáles deben ser esas entradas y salidas, se basa en los requerimientos de software y especificaciones funcionales.

En otras palabras, de una caja negra nos interesará su forma de interactuar con el medio que le rodea entendiendo qué es lo que hace, pero sin dar importancia a cómo lo hace; por tanto, de una caja negra deben estar muy bien definidas sus entradas y salidas (interfaz).

La prueba de Caja Negra no es una alternativa a las técnicas de prueba de la Caja Blanca, sino un enfoque complementario que intenta descubrir diferentes tipos de errores a los encontrados en los métodos de la Caja Blanca. Muchos autores consideran que estas pruebas permiten encontrar: funciones incorrectas o ausentes, errores de interfaz, errores en estructuras de datos o en accesos a las bases de datos externas, errores de rendimiento, errores de inicialización y terminación.

Para preparar los casos de pruebas hacen falta un número de datos que ayuden a la ejecución de los estos casos y que permitan que el sistema se ejecute en todas sus variantes, pueden ser datos válidos o inválidos para el programa según si lo que se desea es hallar un error o probar una funcionalidad. Los datos se escogen atendiendo a las especificaciones del problema, sin importar los detalles internos del programa, a fin de verificar que el programa corra bien. (Terrera, 2017)

2.15.2.1. Tipos de Pruebas

La prueba de caja negra es una metodología para realizar pruebas, estas pueden diseñarse para lograr algunos objetivos diferentes, incluyendo:

- Pruebas funcionales: Las pruebas funcionales tienen como objetivo validar que una aplicación hace lo que se supone que debe hacer. Por ejemplo, las pruebas funcionales pueden probar el mecanismo de autenticación de una aplicación, para verificar que los usuarios legítimos puedan autenticarse exitosamente mientras se rechazan los intentos de inicio de sesión no válidos. Los tipos comunes de pruebas funcionales incluyen comprobaciones de cordura, pruebas de integración y pruebas de sistema.
- Pruebas no funcionales: Las pruebas no funcionales evalúan qué tan bien una aplicación realiza sus funciones principales. Ejemplos de pruebas incluyen pruebas de rendimiento, usabilidad, escalabilidad y seguridad.

- Pruebas de regresión: Las pruebas de regresión están diseñadas para garantizar que un cambio en una aplicación no afecte la funcionalidad. Por ejemplo, las pruebas de regresión deben realizarse después de parchear una vulnerabilidad, para garantizar que el parche no haya provocado que la aplicación falle las pruebas funcionales o no funcionales.

2.15.3. Pruebas de Estrés

Las pruebas de estrés de software, también conocidas como pruebas de carga o pruebas de rendimiento, son un tipo de evaluación que se realiza en aplicaciones y sistemas de software para determinar su capacidad de funcionamiento, bajo condiciones extremas o cargas de trabajo intensas. Su propósito principal es identificar debilidades en el software que podrían llevar a problemas de rendimiento o fallos en situaciones de uso real.

Las pruebas de estrés de software implican someter la aplicación a condiciones que superen su capacidad normal de funcionamiento, con el fin de evaluar su comportamiento en situaciones límite.

2.15.3.1. Formas de Prueba

El objetivo de estas pruebas es identificar problemas de rendimiento, tales como cuellos de botella, fugas de memoria, problemas en el código, y otros problemas que podrían surgir en situaciones de uso real.

- Cargas de usuarios: Incrementar el número de usuarios que acceden a la aplicación para verificar cómo responde a una gran demanda.
- Cargas de datos: Añadir grandes cantidades de datos o transacciones para evaluar cómo maneja la aplicación grandes volúmenes de información.
- Escalabilidad: Probar la capacidad del sistema para crecer o disminuir en respuesta a cambios en la carga.

- Saturación: Llevar el sistema al límite de su capacidad para ver cómo responde y si se produce un fallo.
- Tiempo de respuesta: Medir el tiempo que toma para que la aplicación responda a las solicitudes de los usuarios bajo carga pesada.

2.15.3.2. Características Principales

Dentro de las características principales de las pruebas de estrés de software se encuentra la evaluación de la estabilidad de la aplicación. Al someter la aplicación a cargas elevadas, se busca determinar el momento en el que el programa podría experimentar un fallo y al mismo tiempo entender su capacidad de recuperación ante errores.

El proceso de pruebas de estrés de software implica la entrega de los componentes de hardware, como la CPU, la memoria y el espacio en disco, entre otros; para evaluar la resistencia del software en relación con el uso de estos recursos.

- Carga intensiva.
- Identificación de fallos.
- Monitorización y métrica.
- Simulación de situaciones extremas.
- Escalabilidad.
- Evaluación de rendimiento.
- Corrección de problemas.

En conclusión, estas pruebas se realizan para evaluar la capacidad de las aplicaciones y sistemas en condiciones extremas, son imprescindibles para garantizar que el software funcione sin problemas en entornos de producción. (Sentrio, 2023)



CAPÍTULO III

MARCO APLICATIVO

3. MARCO APLICATIVO

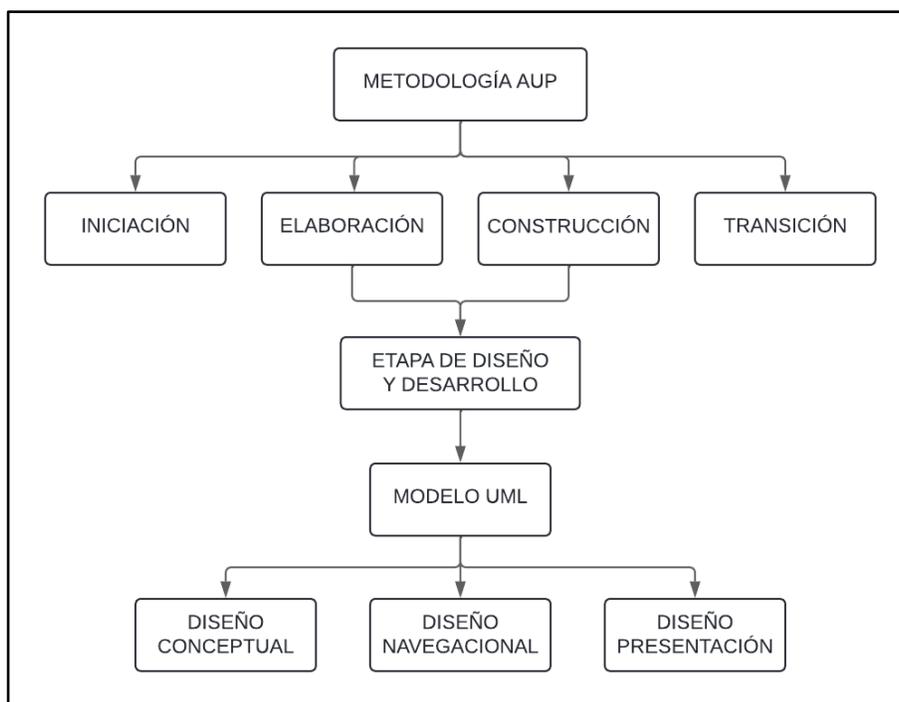
3.1. INTRODUCCIÓN

El objetivo de este capítulo, es realizar el análisis y diseño del Sistema Web para el Laboratorio Industrial de Oxígeno Medicinal – “G.A.M.E.A.”, con el uso de los instrumentos, métodos, técnicas y herramientas mencionadas en el capítulo anterior.

De esta manera, se aplicará la metodología AUP (Proceso Unificado Ágil) en sus respectivas fases, para organizar el tiempo y las actividades a realizar. Y el modelo UML (Lenguaje Unificado de Modelado) con sus diferentes herramientas, que ayudara a modelar el uso del sistema orientado a la web; esto con el objetivo de producir software de alta calidad que cumpla con los requerimientos de los actores, dentro de una planificación que cubra el ciclo de vida del desarrollo de software.

Figura 3.1.

Planificación del Desarrollo de Software



3.2. METODOLOGÍA AUP

El Proceso Unificado Ágil, es la metodología ágil que se aplicará en el desarrollo del proyecto, ya que es una disciplina que nos permite mantener un orden debidamente estricto el cual asigna responsabilidades en una empresa, brindando la facilidad de utilizar Lenguaje Unificado de Modelado de forma práctica, además es un apoyo para realizar muchos procesos que existen para modelar o documentar el sistema de una empresa.

La metodología AUP al ser un proceso iterativo, permite una comprensión creciente de los requerimientos a la vez que se va haciendo crecer el sistema; con este modelo iterativo aborda las tareas más riesgosas primero, logrando así reducir los riesgos del proyecto y tener un subsistema ejecutable tempranamente. Las herramientas de esta metodología proveen un enfoque estructurado para realizar las tareas en el desarrollo del proyecto, asegurando la producción.

AUP (Proceso Racional Ágil), es una herramienta determinada por ciclos y fases para el proceso del modelado, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades a realizar.

Está conformado por cuatro fases que se asignan a una toma de decisiones, las cuales serán descritas y desarrolladas a continuación.

3.2.1. Fase de Inicio

En esta fase de inicio, se analiza todos los problemas que se presentan en el Laboratorio Industrial de Oxígeno Medicinal y los requerimientos que se necesiten, para resolver dichos problemas. Así mismo, se comenzarán a identificar los actores que intervienen en los procesos de entrada y salida de cilindros de oxígeno.

3.2.1.1. Situación Actual

El Laboratorio Industrial de Oxígeno Medicinal, tiene como función principal producir y distribuir oxígeno medicinal a todos los hospitales, centros de salud y población en general, con la finalidad de evitar la escasez de este gas vital en la ciudad de El Alto.

Tabla 3.1.

Obtención de Requisitos

ENTREVISTA	Se realizaron varias entrevistas directas con el administrador general del laboratorio.
OBSERVACIÓN	Se observaron los procesos que se realizan desde el pedido hasta la distribución de cilindros de oxígeno.
DOCUMENTACIÓN	Se consiguieron copias de ordenes de trabajo y hojas de Excel del manejo de inventarios.

El registro de la entrada y salida de cilindros de oxígeno, se realiza de forma manual a través de planillas y formularios, lo que implica pérdida de información, registros erróneos y considerables atrasos en la entrega de cilindros, lo cual tiene como consecuencia pérdidas económicas y pérdida de potenciales asociados.

La identificación de cilindros de oxígeno en el laboratorio, es prácticamente complicado y demoroso para el personal, desconociendo la disponibilidad de los cilindros en almacén, el estado en que se encuentran, si cuentan o no con oxígeno recargado y sin saber la localización de estos de manera confiable e inmediata.

Los pedidos de cilindros de oxígeno por parte de los centros de salud o en algún caso particular de una persona externa, son realizados generalmente de forma personal, donde se emite una orden de trabajo detallando la capacidad y cantidad de los cilindros requeridos, tomando mucho tiempo para este simple proceso.

Las distribuciones realizadas de cilindros de oxígeno, son descritas y detalladas a través de reportes que están realizados en hojas de Excel, pero en un solo equipo de cómputo por lo que el acceso a esa información no es inmediato, además de no ser una forma segura de resguardar ni de respaldar esa información esencial.

3.2.1.2. Descripción de Actores

La descripción de los actores o la identificación de los mismos, en términos generales se refiere a los usuarios del sistema los cuales interactúan, aportan y reciben información del sistema, para coadyuvar a sus tareas cotidianas o necesidades demandadas.

Por lo tanto, se dará a conocer una lista de los actores o usuarios identificados.

Administrador: Es el encargado de coordinar y supervisar los aspectos relacionados con los pedidos y distribuciones de los cilindros de oxígeno.

- Revisa los pedidos de cilindros de los centros de salud.
- Autoriza la distribución de los cilindros de oxígeno.
- Solicita reportes de logística para la toma de decisiones.

Técnico: Es el encargado de controlar y registrar todas las entradas y salidas de cilindros de oxígeno del inventario de almacén.

- Recarga los cilindros de oxígeno para su pronta distribución.
- Registra los cilindros que entran y salen del almacén.
- Verifica el estado de los cilindros existentes en almacén.

Responsable: Es el encargado de su centro de salud, quien realiza la tarea de solicitar los cilindros de oxígeno que se requieran en su área de trabajo.

- Realiza el pedido de cilindros al laboratorio.
- Verifica los pedidos que se le son distribuidos.
- Supervisa los cilindros de su centro a cargo.

3.2.1.3. Determinación de Requerimientos

3.2.1.3.1. Funcionales

Los requerimientos funcionales del sistema, se enfocan principalmente en lo que debe realizar el sistema de forma interna, sin incluir aspectos de interfaz o interacción; se pueden clasificar por medio de tres categorías:

- Evidente: Función que se realizará y el usuario tendrá conocimiento del mismo.
- Oculto: Función que debe realizarse y puede que no sean visibles para el usuario.
- Superfluo: Función que es opcional y su inclusión no repercute en el costo ni en otras funciones.

Tabla 3.2.

Funciones Básicas del Sistema

FUNCIÓN	CATEGORÍA	DESCRIPCIÓN
Registrar Usuario	<i>Evidente</i>	Registra a los usuarios que tendrán acceso al sistema web.
Registrar Cilindro	<i>Evidente</i>	Registra los cilindros de oxígeno con los que cuenta en el laboratorio.
Registrar Centro	<i>Evidente</i>	Registra los centros de salud asociados al laboratorio.
Registrar Pedido	<i>Evidente</i>	Registra los pedidos realizados por los encargados de los centros.
Registrar Distribución	<i>Evidente</i>	Registra las distribuciones de cilindros en base a los pedidos.
Actualizar Información	<i>Oculto</i>	Actualiza el almacén con relación a la entrada y salida de cilindros.
Generar Reporte	<i>Superfluo</i>	Genera reportes de egresos de cilindros del laboratorio.

En la tabla anterior se reflejan las funciones del sistema web, donde la primera columna hace referencia a la cantidad de funciones de una tarea o módulo en específico, la segunda columna muestra las clasificaciones que pueden tener cada función, la tercera columna describe las funciones que en sí engloba un módulo.

Las características que necesita el sistema a partir de la información obtenida, será descrita a continuación en base a su funcionalidad, detallando cada aspecto que realizará un determinado módulo.

a) Administración del sistema:

Base de Datos: Diseñar la base de datos según los requerimientos planteados.

Inicio de Sesión: Definirá los permisos de acceso que tiene el usuario a diferentes módulos del sistema, con respecto al privilegio con el que cuenten.

b) Módulo de Usuarios:

Usuario: Permitirá crear y modificar a usuarios del sistema, se les asignará un nombre de usuario y contraseña para poder ingresar al sistema, con un privilegio de tipo de usuario determinado por el administrador.

c) Módulo de Centros:

Centro: Se podrá insertar y editar centros de salud asociados al laboratorio, asignándoles datos específicos como nombre, ubicación de referencia y un encargado responsable que se encuentra ligado al módulo de usuarios.

d) Módulo de Cilindros:

Cilindro: Se insertarán, editaran y eliminaran lógicamente cilindros de oxígeno, a estos se les asignarán la serie del cilindro (generando su código QR) y una detallada descripción, estableciendo tanto su capacidad, como el dominio al que pertenece.

e) Gestión de Almacén:

Pedido: Se permitirá crear, eliminar y modificar el detalle de los pedidos que realicen los usuarios responsables de los centros de salud, de acuerdo a los cilindros de oxígeno que requieran en base a la información de la disponibilidad en almacén.

Distribución: Las distribuciones de cilindros tienen una relación directa con la salida de cilindros del almacén conforme a los pedidos existente, previamente autorizada por el usuario administrador del sistema.

Inventario: El inventario del almacén se actualizará de forma automática, al momento de registrar las entradas y salidas de cilindros a través de un escáner QR, en función a los pedidos y distribuciones registrados en el sistema.

3.2.1.3.2. No Funcionales

- La aplicación web debe visualizarse y funcionar correctamente en cualquier navegador, como ser: Google Chrome, Opera, Microsoft Edge, Mozilla Firefox, etc.
- La aplicación web debe estar implementada en un servidor web y de base de datos, con el propósito de estar disponible las 24 horas del día.
- Soporte y mantenimiento periódico de la aplicación web, para resguardar el buen funcionamiento del mismo.
- Implementar la aplicación en un servidor donde se pueda almacenar copias de seguridad de manera sencilla y segura.
- La aplicación deberá tener una interfaz amigable para el usuario, donde las consultas deben realizarse en menos de 5 segundos de manera eficaz.
- El sistema deberá permitir realizar actualizaciones inmediatas sin recargar la página, por lo cual no deberá existir pérdida de información.
- El sistema debe indicar la obligatoriedad de llenar los campos en cada uno de los formularios, así como los caracteres permitidos en cada uno de ellos.

3.2.2. Fase de Elaboración

En la fase de elaboración, ya se tienen determinados los puntos importantes para comenzar a elaborar el proyecto, en especial teniendo claramente la visión de lo que se va a realizar; es por eso que los propósitos de esta fase serán el de analizar el modelo completo de los diagramas de casos de uso, el dominio del problema, establecer los cimientos de la arquitectura, desarrollar el plan del proyecto y definir los riesgos que puedan ocasionar algunos elementos en el desarrollo del proyecto, dándose las observaciones respectivas por el cliente o especialista.

3.2.2.1. Diagramas de Casos de Uso

Son diagramas que presentan el comportamiento e interacción entre los actores del sistema, que sin duda alguna es una técnica para comprender los procesos de la organización o en este caso del Laboratorio. En este punto se presenta y describe los diagramas de caso de uso del sistema propuesto, dando una explicación de los personajes o entidades que participarán y las actividades que deben realizarse para llevar a cabo el desarrollo del sistema.

Figura 3.2.

Diagrama de Caso de Uso del Sistema Web

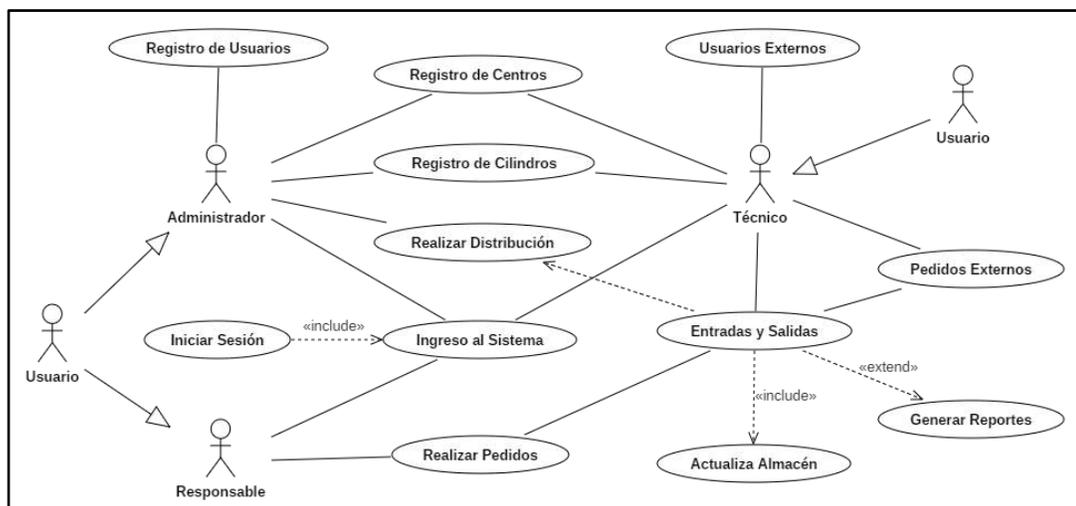


Figura 3.3.

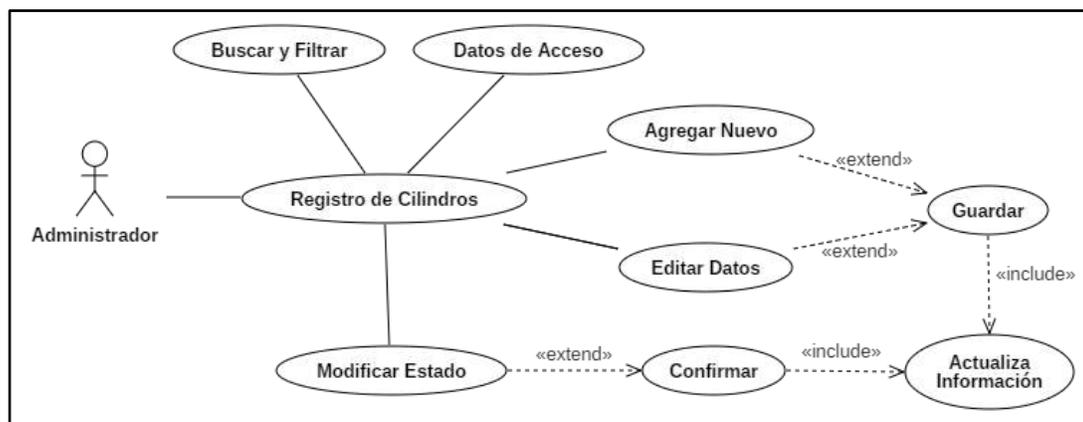
Diagrama de Caso de Uso: Registro de Usuarios

Tabla 3.3.

Descripción de Caso de Uso: Registro de Usuarios

CASO DE USO	Registro de Usuarios
ACTORES	Administrador
PROPÓSITO	Agregar, modificar, activar y buscar usuarios.
RESUMEN	El Administrador ingresa al módulo de usuarios del sistema, donde podrá realizar el registro de un nuevo usuario, la modificación de sus datos personales, cambiar el estado del mismo en alguna excepción y las búsquedas que se requiera según los campos con que cuenta el usuario.

Figura 3.4.

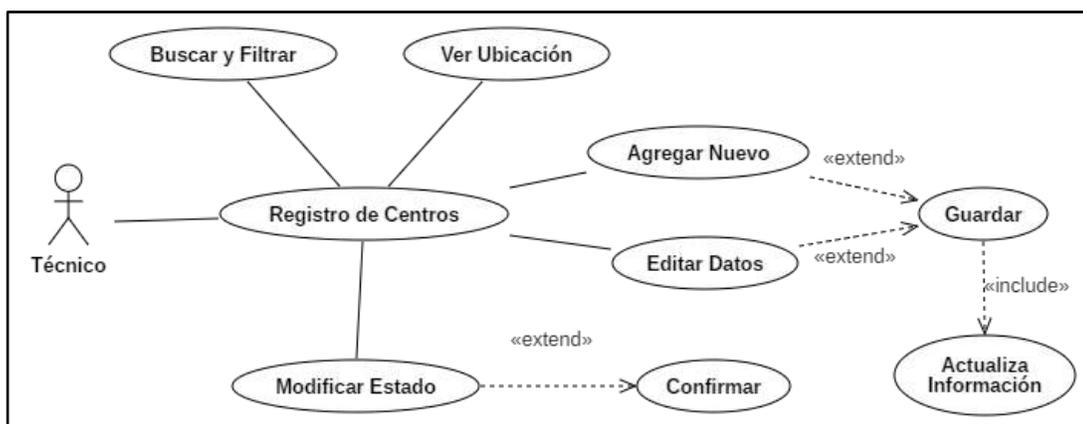
Diagrama de Caso de Uso: Registro de Centros

Tabla 3.4.

Descripción de Caso de Uso: Registro de Centros

CASO DE USO	Registro de Centros
ACTORES	Administrador - Técnico
PROPÓSITO	Agregar, modificar, activar y buscar centros.
RESUMEN	El Usuario ingresa al módulo de centros del sistema, donde podrá realizar el registro de un nuevo centro, la modificación de sus datos, cambiar su estado y usuario a cargo del centro en alguna excepción y las búsquedas que se requiera según los campos que contenga el centro.

Figura 3.5.

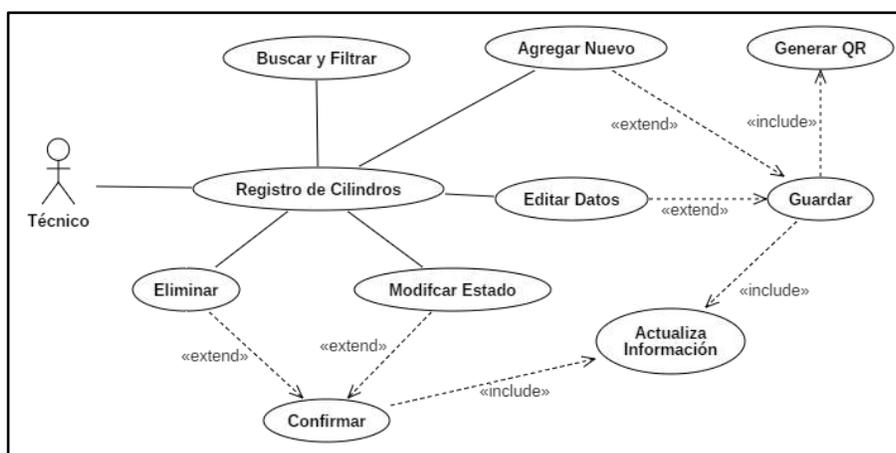
Diagrama de Caso de Uso: Registro de Cilindros

Tabla 3.5.

Descripción de Caso de Uso: Registrar de Cilindros

CASO DE USO	Registro de Cilindros
ACTORES	Administrador - Técnico
PROPÓSITO	Agregar, modificar, eliminar, activar y buscar cilindros.
RESUMEN	El Usuario ingresa al módulo de cilindros del sistema, donde podrá realizar el registro de un nuevo cilindro, la modificación de sus datos, cambio de estado o la eliminación de un registro en caso de alguna excepción y las búsquedas que se requiera según los campos con que cuenta el cilindro.

Figura 3.6.

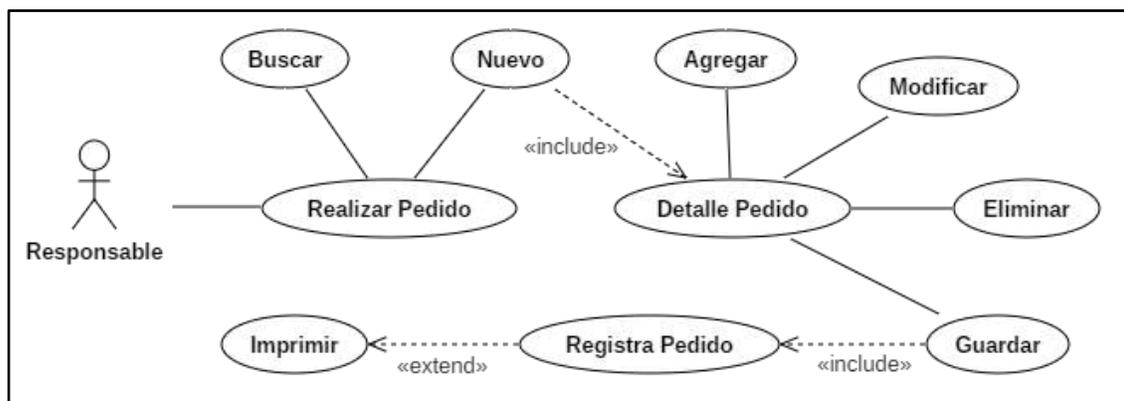
Diagrama de Caso de Uso: Registro de Pedidos

Tabla 3.6.

Descripción de Caso de Uso: Registro de Pedidos

CASO DE USO	Registro de Pedidos
ACTORES	Responsable
PROPÓSITO	Realizar detalle del pedido, capacidad y cantidad.
RESUMEN	El usuario ingresa al módulo de pedidos del sistema, para realizar un nuevo pedido de cilindros que se requieran en su centro, en el detalle podrá seleccionar una capacidad con su respectiva cantidad, donde contará con la edición y eliminación de estos en alguna excepción que se presente.

Figura 3.7.

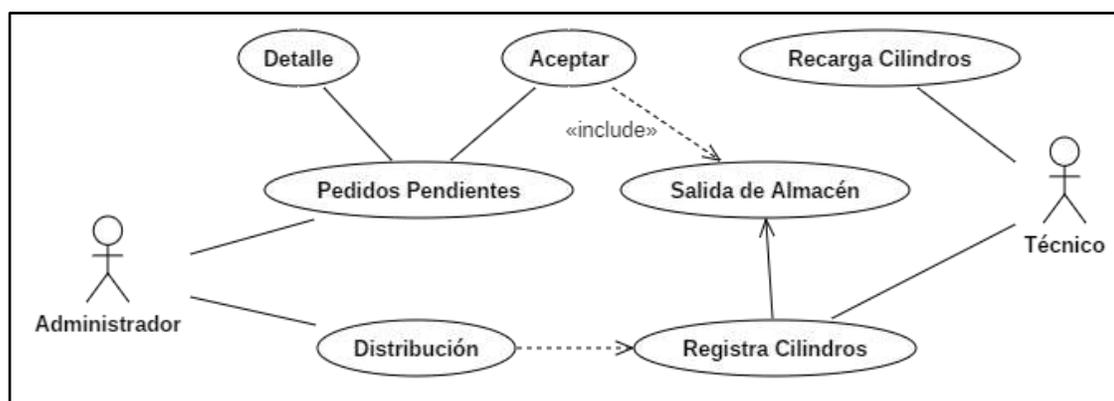
Diagrama de Caso de Uso: Registro de Distribuciones

Tabla 3.7.

Descripción de Caso de Uso: Registro de Distribuciones

CASO DE USO	Registro de Distribuciones
ACTORES	Administrador (Técnico)
PROPÓSITO	Cambiar estado del pedido y realizar su distribución.
RESUMEN	El Administrador ingresa al módulo de pedidos del sistema, donde observará el detalle de todos los pedidos pendientes, para cambiar su estado y así aprobar su salida del almacén; una vez registrados los cilindros de un pedido en específico a través de un escaneo QR, se realizará la distribución.

Figura 3.8.

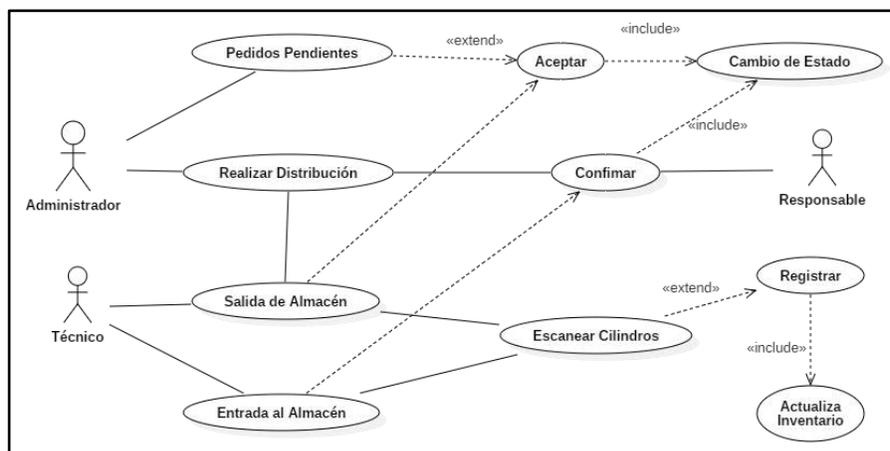
Diagrama de Caso de Uso: Registro de Entrada y Salida

Tabla 3.8.

Descripción de Caso de Uso: Registro de Entrada y Salida

CASO DE USO	Registro de Entrada y Salida
ACTORES	Técnico (Administrador)
PROPÓSITO	Escanear los cilindros que egresen e ingresen del almacén.
RESUMEN	El Usuario ingresa al módulo de almacén del sistema, para ejecutar un escaneo QR de los cilindros que entren o salgan del laboratorio, los cuales estarán ligados estrictamente y dependerán de los pedidos y distribuciones que se realizaron y aprobaron anteriormente por los otros usuarios del sistema.

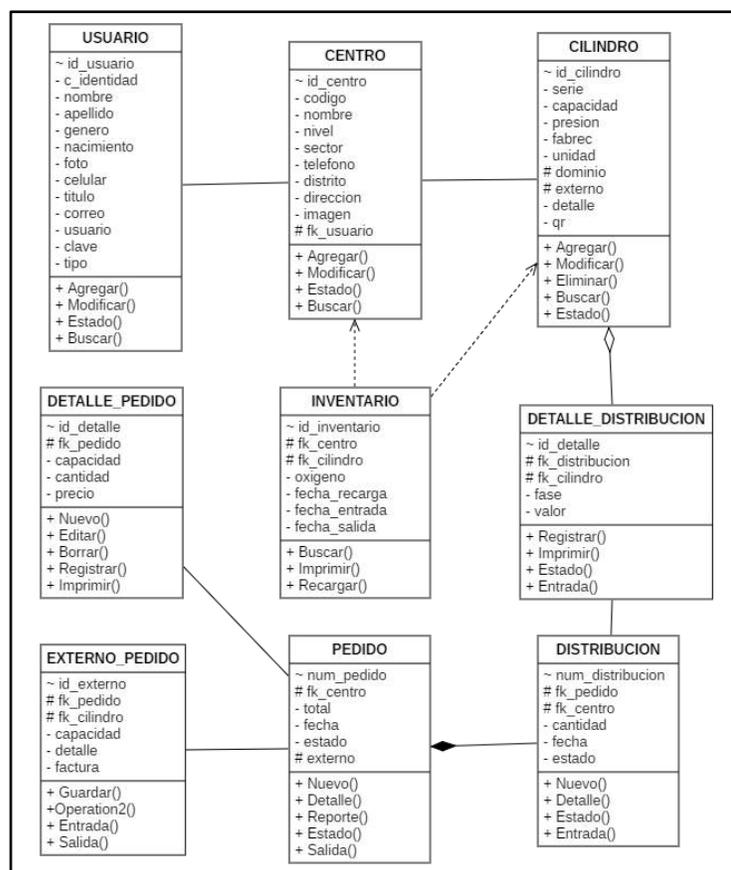
3.2.3. Fase de Construcción

Es la fase en la que la mayoría de decisiones son aceptadas, donde el proyecto está en ejecución en una plataforma definida y todos los componentes restantes se desarrollan e incorporan al producto. Es por eso que el propósito de esta fase es completar la funcionalidad del sistema, para ello se deben clarificar los requisitos pendientes, administrar los cambios de acuerdo a las evaluaciones realizados por los usuarios y ejecutar las mejoras para el proyecto, aplicando los diagramas de clases, diagramas de secuencias, diagramas de estado y diagramas de colaboración.

3.2.3.1. Diagrama de Clases

Figura 3.9.

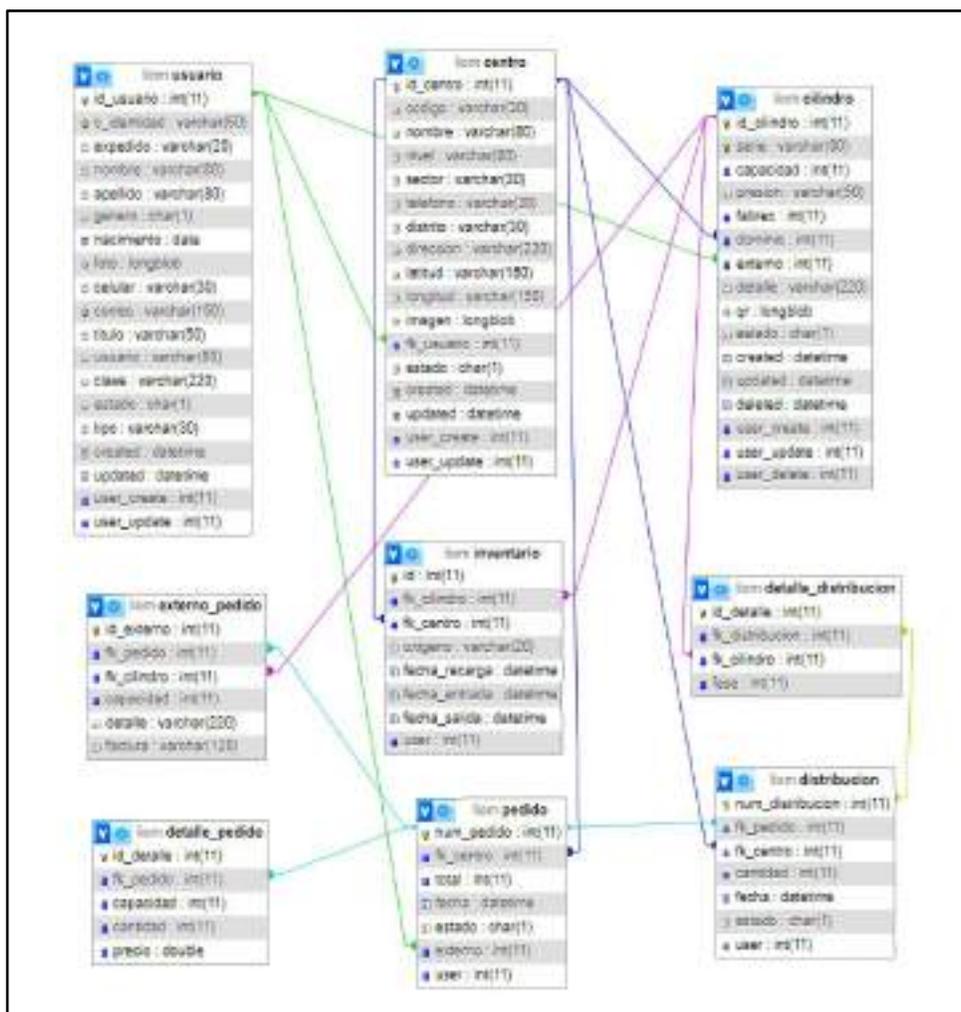
Diagrama de Clases del Sistema



El diagrama de clases es de una estructura estática, que describe gráficamente la estructura de un sistema mostrando sus clases, atributos, operaciones (métodos) y las relaciones entre ellos. Se define a una clase como categoría o grupo de cosas que tiene atributos o acciones similares.

Figura 3.10.

Modelo Relacional (Base de Datos)



3.2.3.2. Diagrama de Actividades

Los diagramas de actividades muestran de forma gráfica los procesos actuales que se realizan con el usuario y el sistema, describiendo cómo interactúan ambos.

Muestra un proceso de negocio o un proceso de software, como un flujo de trabajo a través de una serie de acciones, tanto las personas, los componentes de software o los equipos pueden realizar estas acciones.

Figura 3.11.

Diagrama de Actividades del Sistema Web

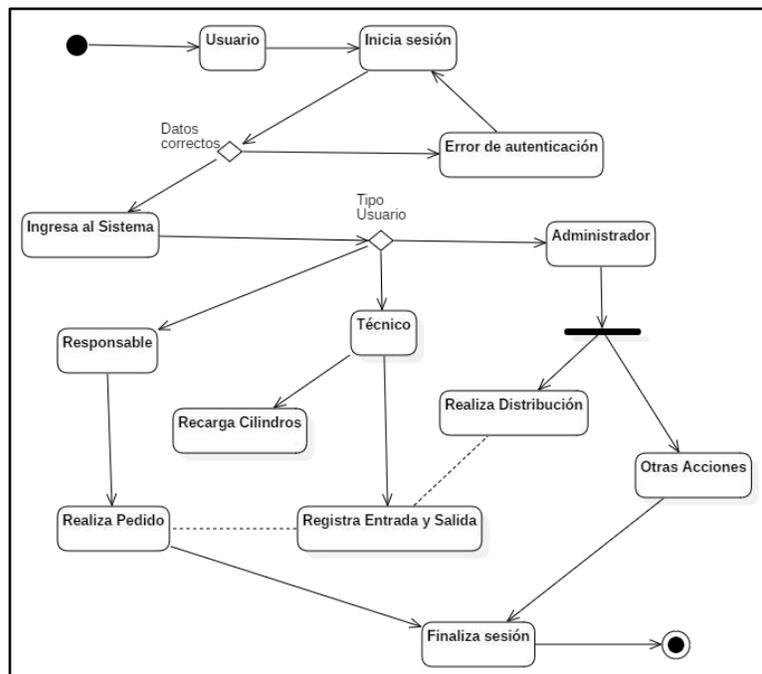
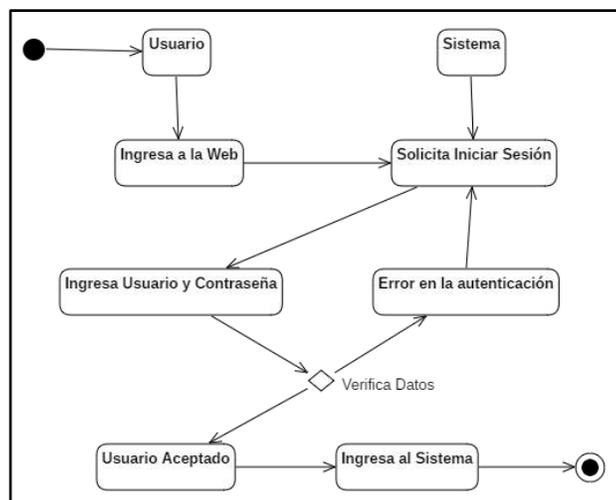


Figura 3.12.

Diagrama de Actividades: Ingreso al Sistema



3.2.3.3. Diagrama de Secuencia

El diagrama de secuencias se realiza a partir de la descripción de un caso de uso, mostrando la interacción de un conjunto de objetos de una aplicación a través del tiempo, en el cual se indicarán los módulos o clases que formaran parte del programa y las llamadas que se hacen cada uno de ellos para realizar una tarea determinada, por esta razón permite observar la perspectiva cronológica de las interacciones.

Figura 3.13.

Diagrama de Secuencia: Registro de Pedido

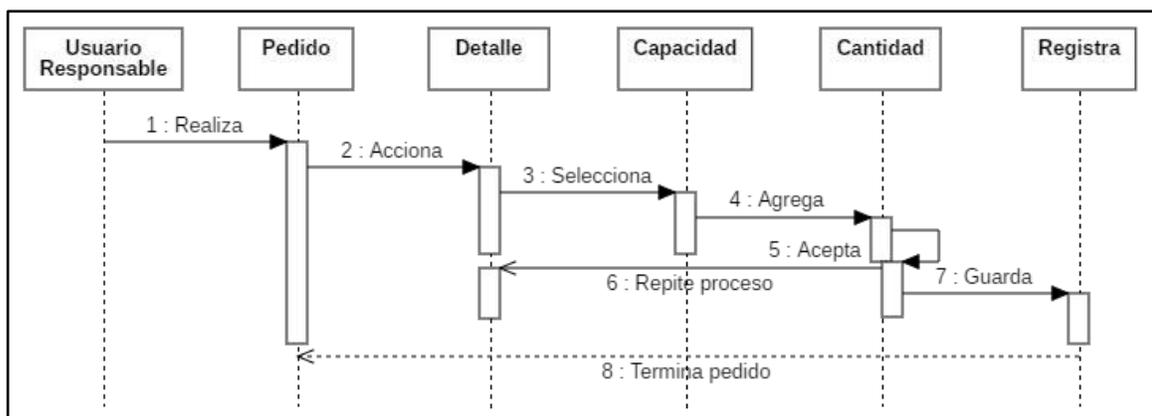
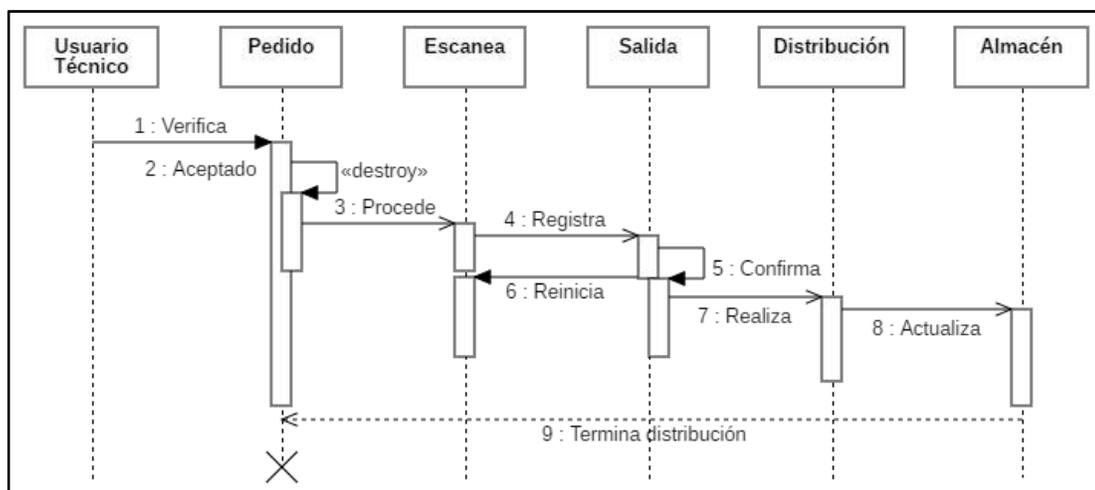


Figura 3.14.

Diagrama de Secuencia: Registro de Distribución

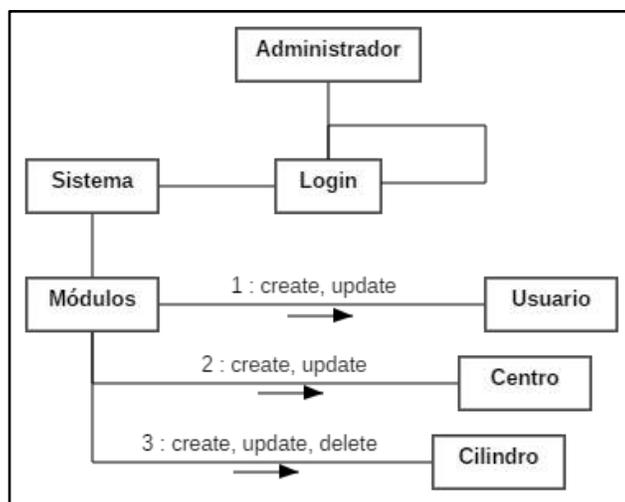


3.2.3.4. Diagrama de Colaboración

Un diagrama de colaboración, describe en forma gráfica el comportamiento de sistemas, subsistemas y operaciones, representando los objetos que intervienen. Son también llamados diagramas de comunicación, por lo que esencialmente muestran las interacciones entre objetos organizados y enlazados entre ellos, alrededor de los roles; por otra parte, no muestra el tiempo como una dimensión aparte, por lo que resulta necesario etiquetar con números de secuencia, tanto la secuencia de mensajes como los hilos concurrentes.

Figura 3.15.

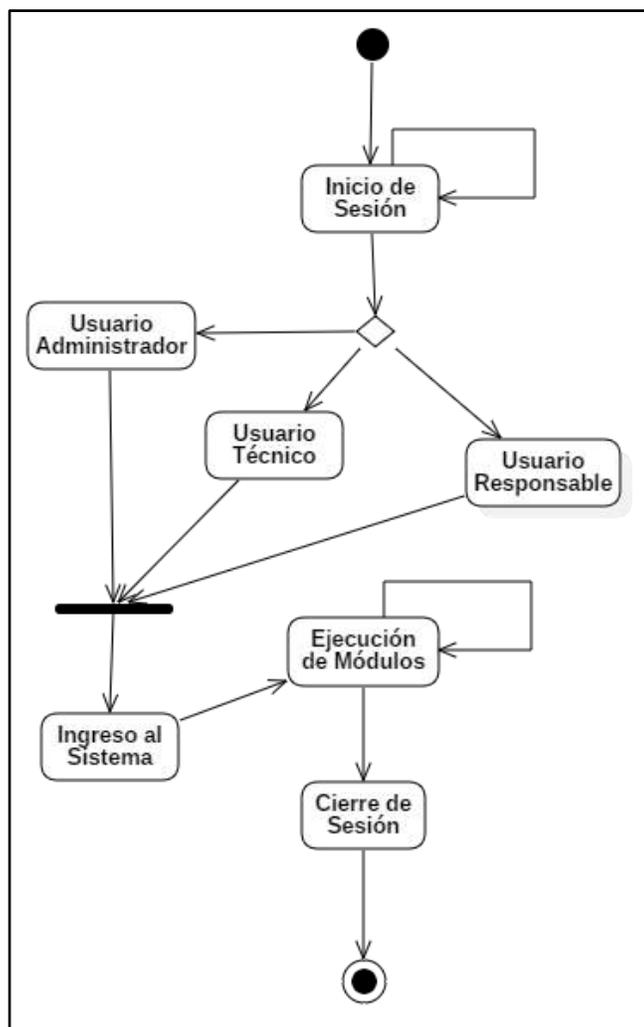
Diagrama de Colaboración: Registro de Módulos



3.2.3.5. Diagrama de Estados

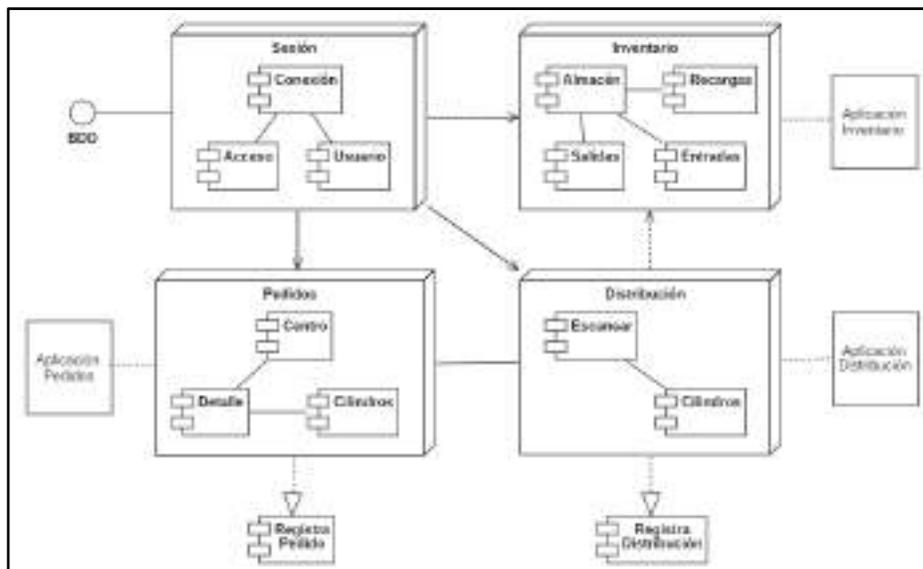
Los diagramas de estado son un método conocido para explicar el comportamiento de un sistema, donde se explican todos los estados posibles en los que puede ingresar un objeto particular y la manera en que modifica el estado del objeto, como resultado de los eventos que llegan a él. Se utilizan para dar una descripción abstracta del comportamiento de un sistema; este comportamiento es analizado y representado por una serie de eventos que pueden ocurrir en uno o más estados posibles.

Figura 3.16.

Diagrama de Estados del Sistema Web**3.2.3.6. Diagrama de Distribución**

Un diagrama de distribución muestra la disposición física de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos; es donde se representa la estructura de hardware, en donde estará el sistema o software, para ello cada componente se representa como nodos, que son cualquier elemento que sea un recurso de hardware, es decir, es la denominación genérica para los equipos.

Figura 3.17.

Diagrama de Distribución del Sistema Web**3.2.4. Fase de Transición**

En esta fase se tendrá el producto para ser implantado en la empresa, en todo caso ser distribuido en donde se requieran el producto, sugiriendo así las aprobaciones si es que el usuario se adecua al sistema. El propósito de esta fase es asegurar que el software esté disponible para los usuarios finales, ajustar los errores y defectos encontrados en las pruebas de aceptación, capacitar a los usuarios y proveer el soporte técnico necesario; esta fase contará con el diagrama de componentes, más los detalles de la implementación del sistema y su funcionamiento.

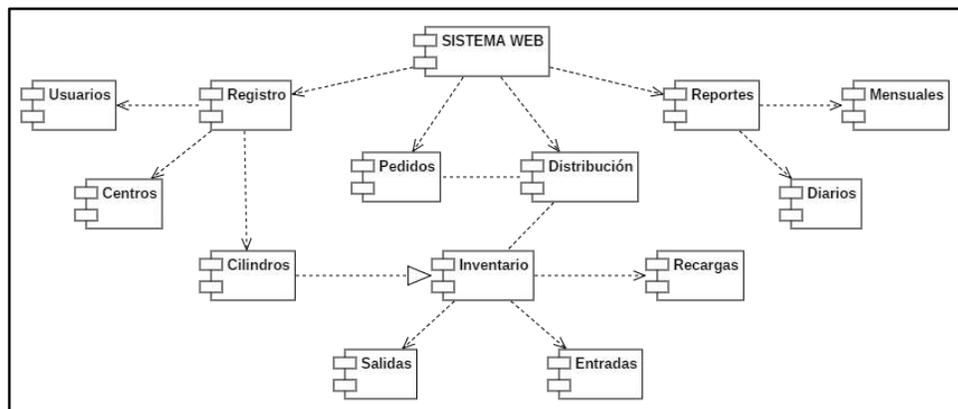
3.2.4.1. Diagrama de Componentes

El diagrama de componentes, permite mostrar todos los elementos del diseño del sistema, proporcionando una visión física de la construcción del sistema de información que muestra la organización de los componentes software, además de sus interfaces y las dependencias entre ellos.

Un diagrama de componentes permite visualizar con más facilidad la estructura general del sistema, el comportamiento del servicio que estos componentes proporcionan y utilizan a través de las interfaces.

Figura 3.18.

Diagrama de Componentes del Sistema Web



3.2.4.2. Diseño de Interfaz

En el diseño de interfaz se debe incluir el dialogo entre el usuario y el sistema, donde el usuario solicita servicios al sistema y le indica cuando realizar cierta acción. La disposición de encabezados, títulos de pantallas y ubicación de datos, hacen que la interfaz con el usuario sea más amigable y no confundan a este con respecto a que acciones debe realizar.

Con el diseño de las interfaces del sistema se presenta como se desenvuelve el sistema para que sea manipulable, obteniendo también la comunicación con la base de datos, donde la información sea manejable por los usuarios permitidos.

- Login del Sistema: En la siguiente imagen se observa el Login del Sistema Web, en donde el usuario debe identificarse para poder iniciar sesión a través de un usuario y contraseña, De esta manera, validando los datos de información se podrá ingresar al sistema y tener acceso a los distintos módulos.

Figura 3.19.

Inicio de Sesión del Sistema Web



- **Página Principal:** En la siguiente imagen se observa la página principal del Sistema Web, dependiendo del tipo de usuario asignado con sus respectivos privilegios; el Administrador es el único que dispondrá con el acceso completo a todos los módulos del sistema, ya que la mayoría de los módulos no serán visibles para los otros usuarios.

Figura 3.20.

Panel Principal del Administrador y Técnico

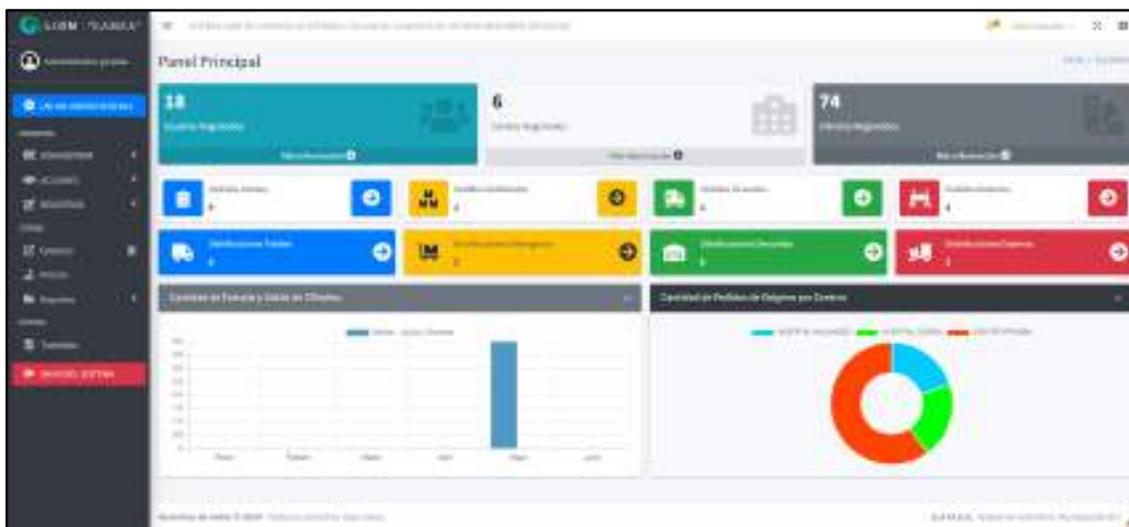


Figura 3.21.

Panel Principal del Responsable

- **Módulo de Usuarios:** En la siguiente figura se observa la página del módulo de usuarios, donde el Usuario Administrador podrá realizar el registro de un nuevo usuario que acceda al Sistema; así mismo podrá modificar sus datos personales, actualizar su estado, realizar búsquedas, otros. Cabe destacar la asignación automática de un usuario y contraseña con los datos del usuario, esto para su autenticación al momento de iniciar sesión.

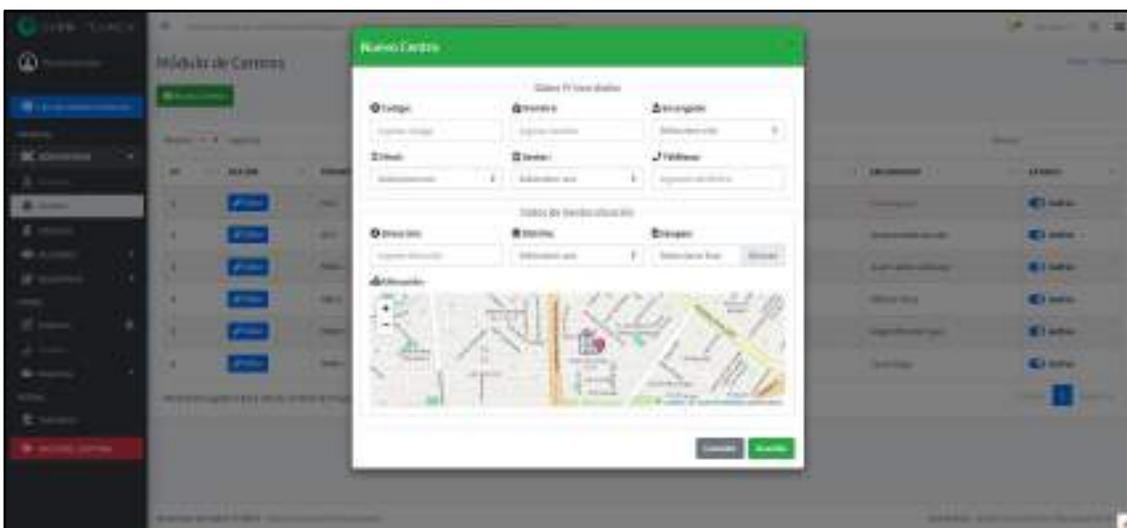
Figura 3.22.

Interfaz Registro de Usuarios

- **Módulo de Centros:** En la siguiente figura se observa la página del módulo de centros, donde el Usuario Administrador o Técnico podrá realizar el registro de un nuevo centro asociado al Laboratorio; así mismo, modificar sus datos correspondientes, actualizar su estado (activo / inactivo), realizar búsquedas según sus campos y demás. Cabe destacar la asignación de un encargado del centro, esto con relación a los usuarios que se tengan registrados en el sistema con el rol Responsable.

Figura 3.23.

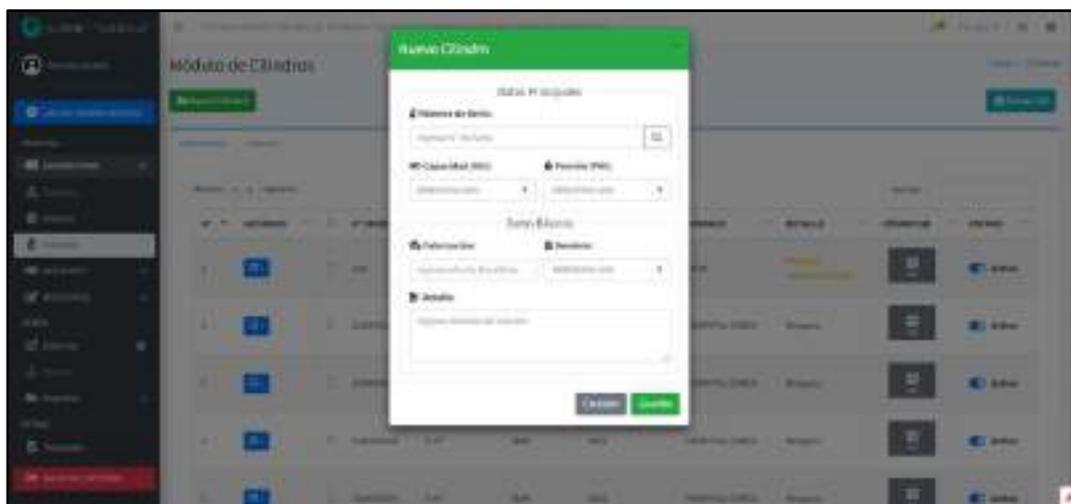
Interfaz Registro de Centros



- **Módulo de Cilindros:** En la siguiente figura se observa la página del módulo de cilindros, donde el Usuario Administrador o Técnico, podrá realizar el registro de un nuevo cilindro, modificar características del mismo, eliminar un registro en caso de alguna excepción, actualizar su estado (activo / inactivo), realizar búsquedas según sus campos y entre otros.

Es importante resaltar, la generación automática de su código QR en base a su número de serie, al momento de insertar o editar un cilindro, ya que permite una identificación única y eficiente, facilitando su lectura en el sistema, el seguimiento de su ubicación, la verificación de su estado y la generación de informes detallados.

Figura 3.24.

Interfaz Registro de Cilindros

- **Módulo de Pedidos:** Este módulo se podrá observar de maneras distintas de acuerdo al Usuario que lo acceda. Las siguientes imágenes muestran como el Usuario Administrador, visualizará todos los pedidos realizados, donde logrará verificar el detalle de cada uno; así mismo, podrá aceptar o rechazar los que estén pendientes, autorizando de esta manera la salida de cilindros del almacén a un respectivo centro.

Figura 3.25.

Interfaz de Pedidos Realizados

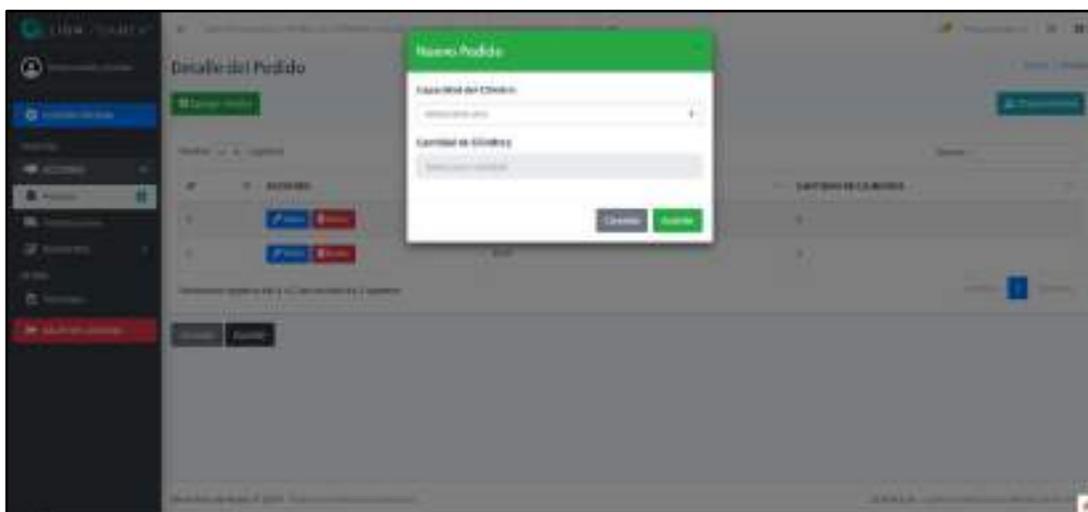
ID	DETALLE PEDIDO	CENTRO SOLICITANTE	TOTAL CILINDROS	FECHA PEDIDO	ESTADO PEDIDO	ACCIONES
1	Verificar	CENTRO MEDICA	4 cilindros	16/06/2024 12:11:07	Completado	[Yellow Button]
2	Verificar	CENTRO MEDICA	1 cilindro	16/06/2024 12:11:05	Completado	[Yellow Button]
3	Verificar	CENTRO MEDICA	1 cilindro	16/06/2024 12:11:05	Completado	[Yellow Button]

Figura 3.26.

Detalle de Pedidos Pendientes

En esta imagen se observa como el Usuario Responsable, podrá realizar el registro de pedidos dando lugar a su detalle, teniendo en cuenta que solo en este último se podrá insertar, editar y eliminar un registro por si existe alguna equivocación. Cabe destacar la validación de campos, incluyendo la disponibilidad de cilindros antes de guardar el pedido por completo.

Figura 3.27.

Interfaz Registro de Pedidos

- Módulo de Distribuciones: Este módulo se podrá observar de maneras distintas de acuerdo al Usuario, teniendo en cuenta primero al Administrador y posteriormente al Responsable.

En esta primera imagen se visualiza el registro de todas las distribuciones realizadas a los centros de acuerdo a los pedidos recibidos, en donde se logrará verificar el detalle de cada uno; el Usuario Técnico, solo podrá ver las distribuciones registradas por su persona.

Figura 3.28.

Distribuciones del Administrador



The screenshot shows the 'Módulo de Distribución' interface. It features a sidebar menu on the left with options like 'Inicio', 'Acciones', 'Distribuciones', 'Cilindros', 'Reservaciones', 'Reportes', and 'Configuración'. The main content area displays a table of distribution records with the following columns: 'N°', 'DETALLE DISTRIBUCIÓN', 'PERIODO MUESTREADO', 'CENTRO DESTINATARIO', 'CANTIDAD CILINDROS', 'FECHA DISTRIBUCIÓN', 'ESTADO LÍNEA', and 'SERVIDOR EXPRESO'. The table contains three rows of data, each with a 'Detalle' button and a 'Cilindros' button.

N°	DETALLE DISTRIBUCIÓN	PERIODO MUESTREADO	CENTRO DESTINATARIO	CANTIDAD CILINDROS	FECHA DISTRIBUCIÓN	ESTADO LÍNEA	SERVIDOR EXPRESO
1	[Detalle]	PERIODO 1	CENTRO PRIMA	1	20-01-2024 10:43:14	[Estado]	[Botón]
2	[Detalle]	PERIODO 1	CENTRO PRIMA	1	20-01-2024 10:43:14	[Estado]	[Botón]
3	[Detalle]	PERIODO 2	CENTRO PRIMA	1	20-01-2024 10:43:14	[Estado]	[Botón]

En esta segunda imagen se visualiza solo las distribuciones hechas al centro del encargado, donde podrá verificar el detalle de estos, para así confirmar la llegada cilindros a su área de trabajo; permitiendo de esta manera, su futura entrada al almacén del Laboratorio.

Figura 3.29.

Distribuciones del Responsable



The screenshot shows the 'Módulo de Distribución' interface with a modal window titled 'Detalle Distribución' open. The modal displays the title 'Distribución de Cilindros N°2' and a table with the following columns: 'N°', 'CILINDRO (SERIE)', and 'CAPACIDAD DEL'. The table contains three rows of data.

N°	CILINDRO (SERIE)	CAPACIDAD DEL
1	20541234	0 m ³
2	20541234	0 m ³
3	01212121	0 m ³

- **Módulo de Inventarios:** En las siguientes imágenes se observan las páginas de inventario del almacén, donde se podrá visualizar en qué centro se localiza cada cilindro, el estado en que se encuentre, sus fechas de entrada y salida, su disponibilidad y demás. De igual manera, en este se registra la recarga de los cilindros a través de un escaneo y también se realizan búsquedas según en campo de información que se requiera.

Figura 3.30.

Interfaz de Inventario del Almacén



Figura 3.31.

Interfaz Disponibilidad de Cilindros

The screenshot displays the 'Inventario del Almacén' (Warehouse Inventory) interface, specifically the 'Disponibilidad de Cilindros' (Cylinder Availability) section. The table shows the following data:

ID	Disponibilidad de Cilindros	Cantidad	Porcentaje
1	INVENTARIO DE...	---	100%
2	INVENTARIO DE...	---	100%
3	INVENTARIO DE...	---	100%
4	INVENTARIO DE...	---	100%
5	INVENTARIO DE...	---	100%
6	INVENTARIO DE...	---	100%
7	INVENTARIO DE...	---	100%
8	INVENTARIO DE...	---	100%

- Entradas y Salidas: En las siguientes imágenes se observan las páginas del registro de entradas y salidas, donde el Usuario Técnico podrá escanear los QR de los cilindros, estos ligados estrictamente a los módulos de pedidos y distribuciones, ya que depende del estado de los mismos para su respectivo ingreso o egreso del almacén.

Figura 3.32.

Interfaz Móvil de Salidas

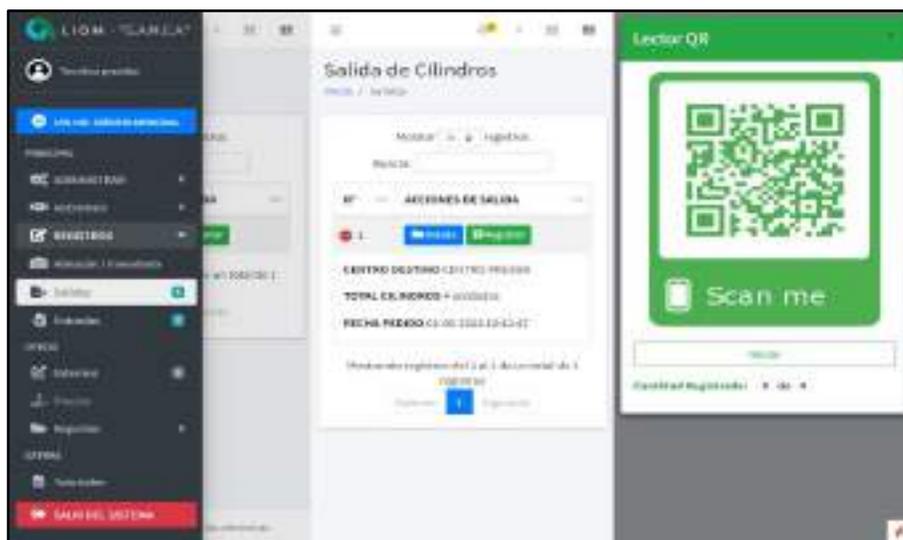
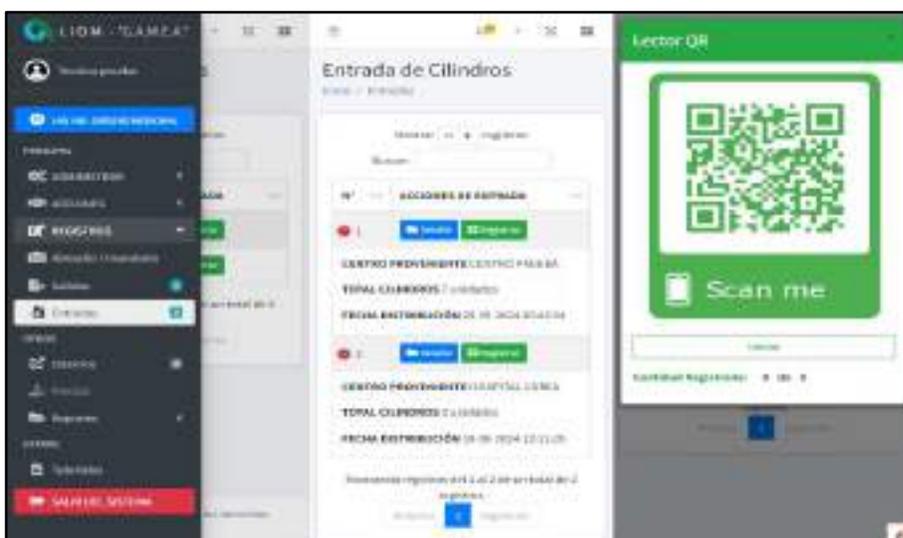


Figura 3.33.

Interfaz Móvil de Entradas



- **Módulo de Externos:** En la siguiente figura se observa la página del módulo de externos, donde el Usuario Administrador o Técnico podrá realizar el registro de un nuevo pedido externo, con relación a un usuario externo y sus correspondientes cilindros; así mismo, se podrá realizar su respectiva distribución en base al proceso de registrar su salida.

Figura 3.34.

Interfaz de Externos



- **Generador de Reportes:** En las siguientes imágenes se observan las páginas de reportes, donde los usuarios Administrador y Técnico podrán generar y filtrar reportes (anuales, mensuales y diarios) de forma automática, a partir de todas las distribuciones realizadas a los distritos centros.

Figura 3.35.

Reportes Mensuales y Diarios



Figura 3.36.

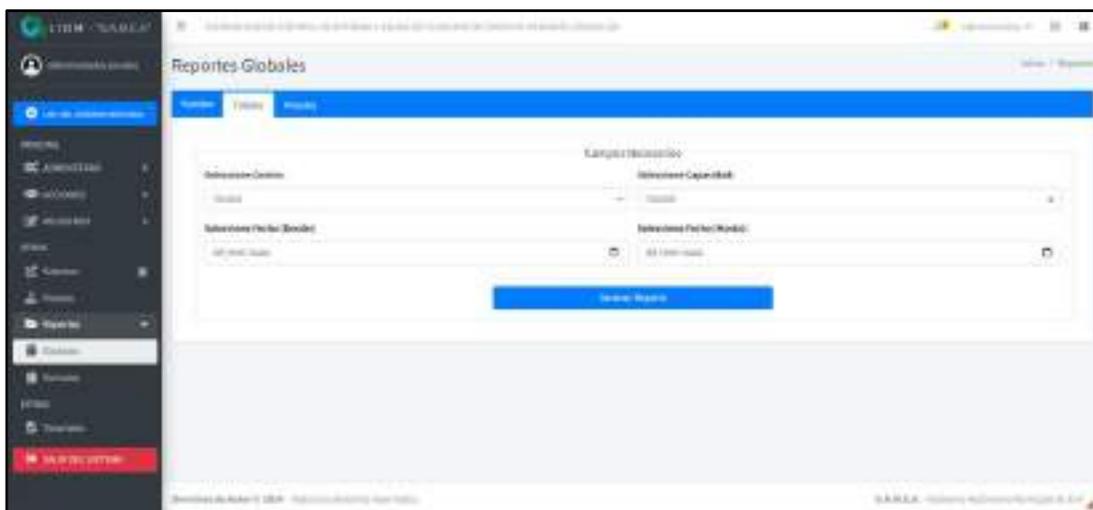
Generación de Reportes

Figura 3.37.

Reportes Anuales de Centros

CENTRO	SECTOR	REQUERIDO EN 2018
CENTRO DE INVESTIGACIONES	SECTOR AGRICOLA	24
CENTRO DE LOGISTICA	SECTOR INDUSTRIAL	22
CENTRO DE MANTENIMIENTO	SECTOR INDUSTRIAL	21
CENTRO DE OPERACIONES	SECTOR INDUSTRIAL	21
CENTRO DE SERVICIOS	SECTOR INDUSTRIAL	21
REQUERIDO EN 2018		108
CENTRO DE SERVICIOS	SECTOR INDUSTRIAL	11
TOTAL DE SERVICIOS		108

3.3. PRUEBAS DE SOFTWARE

Una vez finalizado el desarrollo de las cuatro etapas de la metodología, se realizará las pruebas necesarias a los módulos desarrollados, para garantizar el funcionamiento del sistema antes de subirlo a producción.

Para las pruebas de software se utiliza el método de pruebas de caja negra y caja blanca, donde se evalúan las entradas introducidas por los usuarios y se analizan el resultado devuelto por el sistema; además de la prueba de funcionalidad.

3.3.1. Prueba de Caja Blanca

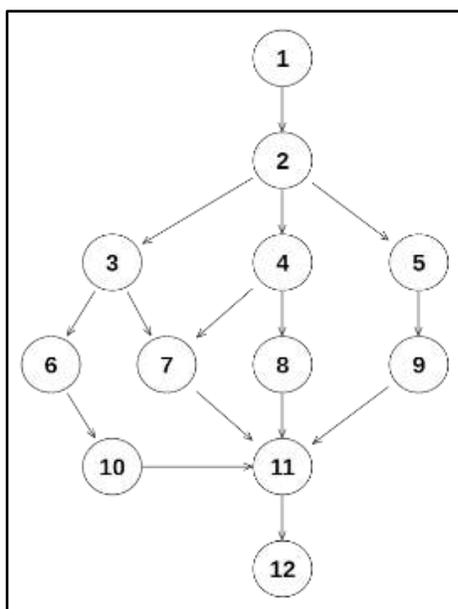
Este método de prueba, se orienta al cálculo de las regiones que deben ser consideradas como partes independientes del sistema, estableciendo cuales son las entradas que se ejecutan, asegurando así que cada región se ejecute al menos una vez.

Esta prueba se aplica en los procesos más relevantes del modelo de pronóstico de demanda y el sistema en cual se realiza su aplicación siendo técnica de prueba del camino básico con grafos, complejidad ciclo matica y derivación de casos de prueba.

De forma general, se empleará el diseño de las actividades que realiza el usuario (responsable) dentro del sistema, elaborando el grafo del programa de la siguiente manera:

Figura 3.38.

Grafo de Pruebas Actividades del Usuario



Se encontraron: $A = 15$ (Aristas) y $N = 12$ (Nodos)

Donde:

- 1) Inicio del Sistema
- 2) Menú Principal
- 3) Módulo Pedidos
- 4) Módulo Distribuciones
- 5) Módulo Inventario
- 6) Nuevo Pedido
- 7) Detalles y Reportes
- 8) Confirmar Distribución
- 9) Disponibilidad Cilindros
- 10) Registrar Pedido
- 11) Culmina Ciclo Sistema
- 12) Fin del Sistema

Por tanto, analizado el grafo generado a partir de las características del sistema, ahora se procede a determinar la complejidad ciclomática del grafo mediante la ecuación (12) :

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

$$V(G) = 15 - 12 + 2$$

$$V(G) = 5$$

Los caminos que deben ser probados dadas ciertas variables son 5, para determinar el conjunto básico de caminos linealmente independientes. Estos caminos son los siguientes:

- **Camino 1: 1 - 2 - 3 - 6 - 10 - 11 - 12**
- **Camino 2: 1 - 2 - 3 - 7 - 11 - 12**

- *Camino 3: 1 - 2 - 4 - 7 - 11 - 12*
- *Camino 4: 1 - 2 - 4 - 8 - 11 - 12*
- *Camino 5: 1 - 2 - 5 - 9 - 11 - 12*

Ahora hay que preparar los casos de prueba para forzar la ejecución de cada camino; esta última condición establece que, para la ejecución de ciertos caminos, se deben establecer las condiciones en las que al menos se ejecuta los nodos establecidos en el camino.

- *Camino 1:* Se registra un nuevo pedido de cilindros al laboratorio, especificando su detalle por medio de su respectiva capacidad y cantidad.
- *Camino 2:* Se revisa la información de los pedidos, como el estado y su detalle; así también se imprime el reporte de un pedido en específico.
- *Camino 3:* Se revisa la información de las distribuciones, como el estado y su detalle; así también se imprime el reporte de una distribución en específico.
- *Camino 4:* Se confirma la distribución de un pedido en particular, validando de esta manera el almacenamiento de cilindros al centro a cargo.
- *Camino 5:* Se revisa la información de cilindros en el centro; así también ve la cantidad de cilindros disponibles en el laboratorio para futuras solicitudes.

3.3.2. Prueba de Caja Negra

Las pruebas de caja negra o también conocidas como pruebas de comportamiento, se centran en los requisitos funcionales del software. Este método de prueba, nos ayuda a evaluar su forma de interactuar del sistema, las entradas que recibe y las salidas que produce, esto acorde a las expectativas del usuario final.

Para realizar la prueba de caja negra se realiza las pruebas correspondientes a la interfaz de Inicio de Sesión, mostrada a continuación:

Figura 3.39.

Ventana de Acceso al Sistema

Tabla 3.9.

Valores Límites de Inicio de Sesión

CAMPOS	ENTRADA VALIDA	ENTRADA INVALIDA
Usuario	<i>Cadena de Texto</i>	Caracteres especiales, espacios en blanco.
Contraseña	<i>Cadena de Texto</i>	Caracteres especiales, espacios en blanco.

Tabla 3.10.

Prueba de Caja Negra: Inicio de Sesión

VALORES	CAMPOS	DESCRIPCIÓN
ENTRADAS	Usuario Contraseña	Administrador *****
SALIDAS	Mensaje de campos requeridos. Ventana de alerta por datos de acceso equivocados.	El sistema valida que no se ingresen datos en blanco. El sistema informa si el usuario y/o contraseña no son correctos.
RESULTADO	Acceso a la página principal del sistema.	El sistema realiza la autenticación de usuario, si los datos proporcionados son válidos concede acceso al sistema.

3.3.2.1. Prueba Registro de Pedidos

En el proceso de registrar pedidos cumple con la función de ingresar los datos de capacidad y cantidad al sistema, de esta forma podrá ser utilizado para la distribución de cilindros, posterior a una autorización para las salidas y entradas al inventario del almacén.

Una vez realizada la prueba de caja negra a la interfaz de registro de un pedido, se evidenciará que la misma cumple con la función de registro del pedido, obligando al usuario a registrar los campos obligatorios.

Figura 3.40.

Ventana de Registrar Pedido

Tabla 3.11.

Valores Límites de Registro de un Pedido

CAMPOS	ENTRADA VALIDA	ENTRADA INVALIDA
Capacidad	<i>Selección múltiple</i>	Caracteres especiales, espacio de selección.
Cantidad	<i>Cadena numérica</i>	Caracteres especiales, iniciando en uno.

Tabla 3.12.

Prueba de Caja Negra: Registro Pedido

VALORES	CAMPOS	DESCRIPCIÓN
ENTRADAS	Capacidad Cantidad	1 al 8 (m ³) 1 al N (Stock)
SALIDAS	Mensaje de campos requeridos. Mensaje de alerta por cilindros limitados.	El sistema valida que no se ingresen datos en blanco. El sistema informa si la cantidad de cilindros solicitados, supera los disponibles en el laboratorio.
RESULTADO	Se registra el pedido junto a su detalle, con un mensaje de confirmación.	El sistema valida si los datos ingresados son correctos y registra la información del pedido en la BDD.

3.3.3. Prueba de Estrés

El objetivo de las pruebas de estrés es saturar la aplicación web hasta un punto de quiebre donde aparezcan defectos potencialmente peligrosos, no para decir que el sistema no funciona, sino lo que se intenta es mejorar la aplicación web, reduciendo riesgos que puedan dar origen a una caída del sistema.

Por tal motivo, el proyecto fue sometido a pruebas de estrés con uso del servicio de Apache JMeter, que sirve para probar el rendimiento de aplicaciones web.

Figura 3.41.

Vinculación al Servidor

Figura 3.42.

Asignación de Usuarios Virtuales

Figura 3.43.

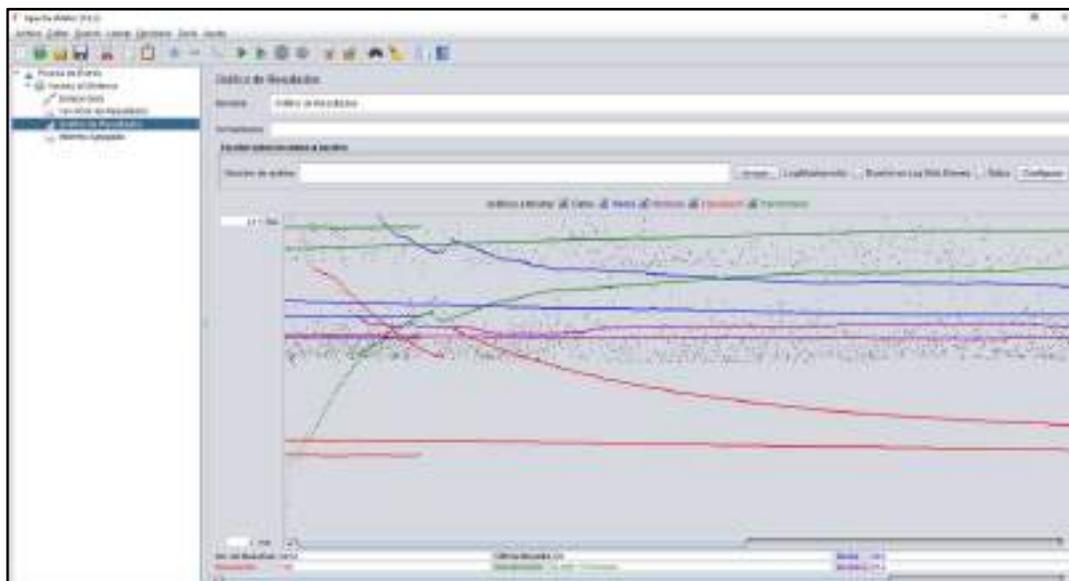
Rendimiento del Sistema

Figura 3.44.

Resultado de la Prueba

Muestra	P. Usuarios	Ejecución	Inicio	Fin	95% (ms)	90% (ms)	99% (ms)	Max	Min	Error	Requisitos	Ejecución	Código de salida
Test	ACM	381	254	275	487	680	10	1001	5.00%	34.00s	12/23/21	44	

3.3.3.1. Resultados de Rendimiento

Los resultados encontrados después de las pruebas realizadas a través del software de Apache JMeter,, son descritos a continuación:

- Usuarios virtuales: 10
- Tiempo: 2 min.
- Solicitudes realizadas: 4204
- Rendimiento: 34,9/sec.
- Fallas HTTP: 5,0%
- Pico de solicitud: 10 req./sec.

3.3.4. Prueba de Funcionalidad

Las pruebas de funcionalidad, son necesarias para garantizar el funcionamiento óptimo del sistema, tomando en cuenta los casos de uso representativos del mismo. El uso de las pruebas funcionales es para asegurar el correcto trabajo de la entrada de datos, la navegación en el sistema, los procedimientos y la obtención de resultados.

La prueba se realizó en base a formularios, facilitando al Usuario con distintas opciones de respuesta, según la satisfacción de su persona con el sistema, siendo estas valoraciones:

Acceptable = 55; Regular = 70; Bueno = 85; Óptimo: 100

La valoración obtenida, será la sumatoria de valores de los campos de respuesta, dividida sobre el valor máximo que se puede obtener de las preguntas elaboradas, para al final multiplicarlas por el porcentaje total; esta calificación será el resultado final de la prueba a los formularios del Sistema Web.

Se evaluó la satisfacción del Usuario, en base a preguntas de las siguientes tablas:

Tabla 3.13.

Prueba de Funcionalidad: Acceso al Sistema Web

N°	Descripción	Aceptable	Regular	Bueno	Óptimo
1	¿Valida el usuario y contraseña, para el ingreso del sistema?				✓
2	¿El tiempo de ingreso al sistema es menor a los cinco segundos?			✓	
3	¿Existe el nivel de seguridad adecuado en el sistema web?				✓
4	¿Ingresa al sistema con el privilegio de usuario correspondiente?				✓
5	¿No se presenta alguna dificultad al momento de iniciar sesión?			✓	

$$\text{Resultado Obtenido} = 100 + 85 + 100 + 100 + 85 = 470$$

$$\text{Resultado } R_1 = (470/500) * 100\% = 94\%$$

Figura 3.45.

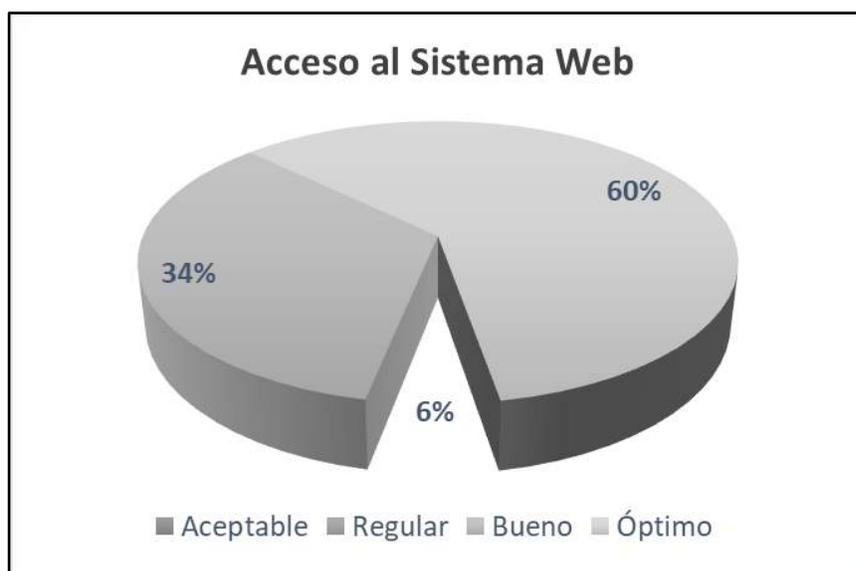
Resultado de la Prueba: Acceso al Sistema Web

Tabla 3.14.

Prueba de Funcionalidad: Administración de Módulos

N°	Descripción	Aceptable	Regular	Bueno	Óptimo
1	¿El ingreso a los distintos módulos del sistema es en poco tiempo?			✓	
2	¿Las búsquedas se realizan de una manera rápida y eficaz?				✓
3	¿Los campos están validados al momento de agregar y/o editar?			✓	
4	¿La información de tablas se actualizan de forma automática?				✓
5	¿El sistema tiene una interfaz accesible y fácil de utilizar?			✓	

$$\text{Resultado Obtenido} = 85 + 100 + 85 + 100 + 85 = 455$$

$$\text{Resultado } R_2 = (455/500) * 100\% = 91\%$$

Figura 3.46.

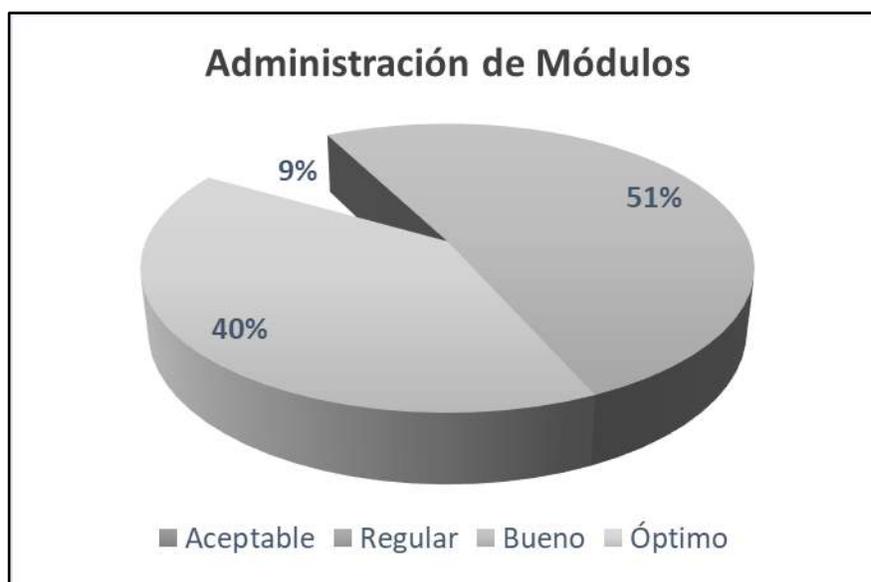
Resultado de la Prueba: Administración de Módulos

Tabla 3.15.

Prueba de Funcionalidad: Registro de Pedidos

N°	Descripción	Aceptable	Regular	Bueno	Óptimo
1	¿Ingresa a la página de registro de pedidos en poco tiempo?			✓	
2	¿El estado de los pedidos son entendibles y están ordenados bien?			✓	
3	¿Es correcta y se verifica la información de detalle del pedido?				✓
4	¿La información de la disponibilidad de cilindros es visible?			✓	
5	¿Los campos están validados al momento de agregar y/o editar?				✓
6	¿Se guarda el detalle del pedido y se redirecciona la página web?			✓	

$$\text{Resultado Obtenido} = 85 + 85 + 100 + 85 + 100 + 85 = 540$$

$$\text{Resultado } R_3 = (540/600) * 100\% = 90\%$$

Figura 3.47.

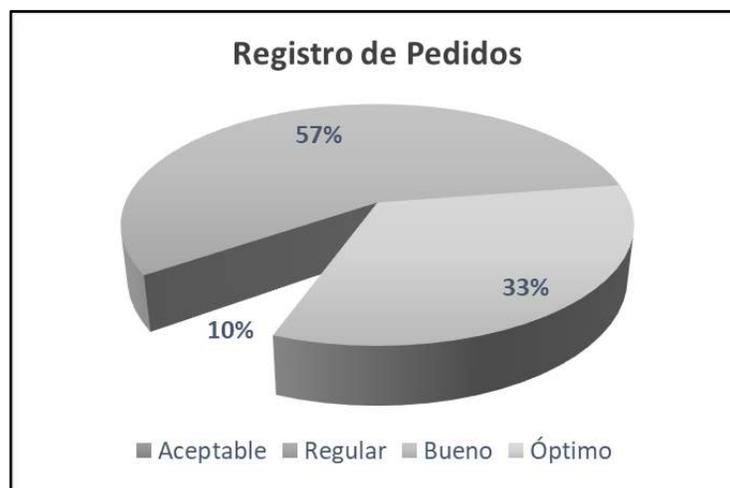
Resultado de la Prueba: Registro de Pedidos

Tabla 3.16.

Prueba de Funcionalidad: Registro de Entrada y Salida

N°	Descripción	Aceptable	Regular	Bueno	Óptimo
1	¿Ingresa a la página de entradas y salidas de cilindros en poco tiempo?			✓	
2	¿Se muestra solo los registros pendientes para su entrada y/o salida?				✓
3	¿Se puede verificar la información de detalle antes del registro?			✓	
4	¿El registro por medio de código QR es de manera rápida y eficaz?				✓
5	¿Están validados los QR escaneados a través del sistema web?				✓
6	¿Se actualiza el inventario del almacén automáticamente?			✓	

$$\text{Resultado Obtenido} = 85 + 100 + 85 + 100 + 100 + 85 = 555$$

$$\text{Resultado } R_4 = (555/600) * 100\% = 92.5\% \approx 93\%$$

Figura 3.48.

Resultado de la Prueba: Registro de Entrada y Salida

3.3.4.1. Resultado Total

$$\text{Resultado Total} = \frac{(R_1 + R_2 + R_3 + R_4)}{R_T}$$

$$\text{Resultado } R_F = \frac{(94 + 91 + 90 + 93)}{4}$$

$$R_F = 368/4 = 92\%$$

Una vez realizado las pruebas de usuario, a través de las preguntas del formulario para cada caso de estudio, se determina que el sistema satisface las necesidades del usuario en un 92% con la prueba de funcionalidad, que se encuentra en una escala de calificación de bueno a óptimo.



CAPÍTULO IV

ANÁLISIS DE CALIDAD, SEGURIDAD Y COSTOS

4. ANÁLISIS DE CALIDAD, SEGURIDAD Y COSTOS

4.1. MÉTRICAS CALIDAD

Las Métricas de Calidad proporcionan una indicación de cómo se ajusta el software a los requerimientos implícitos y explícitos del cliente. De esta manera, se dará uso de la norma ISO/IEC 9126, mismo que propone una jerarquía de atributos de calidad como la de un producto o servicio para satisfacer las necesidades del usuario, que siempre buscará esta cualidad en todos los productos, no solamente de equipos sino también de programas.

El estándar ISO-9126 establece que cualquier componente de la calidad del software puede ser descrito en un conjunto estructurado de características y sub características, que serán detallados a continuación.

4.1.1. Funcionalidad

La funcionalidad de un sistema permite calificar si un producto de software, maneja en forma adecuada el conjunto de funciones que satisfagan las necesidades para las cuales fue diseñado. El objetivo es revelar problemas y errores en lo que concierne a la funcionalidad del sistema y su conformidad al comportamiento, expresado o deseado por el usuario.

La funcionalidad de un software, se puede medir de acuerdo a la complejidad del sistema, para realizar la medida indirecta del software se toma la métrica de punto de función, el cual se usa como medio para medir la funcionabilidad de entrega del sistema.

4.1.1.1. Técnica Punto de Función

El punto función es una métrica orientada a la función del software y del proceso por el cual se desarrolla, se centra en la funcionalidad o utilidad del programa.

Esta técnica permite cuantificar el tamaño de la Aplicación Web en unidades independientes del lenguaje de programación o la metodología utilizada.

Para el cálculo de punto función se toma en cuenta cinco características, el dominio de información, como son números de entrada, salida, condiciones, archivos e interfaz externa. Luego se realiza el cálculo de punto de función hallando la suma de estas características, parámetros de medición y el factor de ponderación también llamado punto medio de ponderación.

- Número de entradas de usuario. Se cuenta cada entrada de usuario que proporciona diferentes datos orientados a la aplicación. Las entradas se deberían diferenciar de las peticiones, las cuales se cuentan de forma separada.
- Número de salidas de usuario. Se cuenta cada salida que proporciona al usuario información orientada a la aplicación. En este contexto la salida se refiere a informes, pantallas, mensajes de error, etc.
- Número de peticiones de usuario. Una petición se define como una entrada interactiva que produce la generación de alguna respuesta del software inmediata en forma de salida interactiva. Se cuenta cada petición por separado.
- Número de archivos. Se cuenta cada archivo maestro lógico (se refiere a un grupo lógico de datos que puede ser una parte de una gran base de datos o un archivo independiente).
- Número de interfaces externas. Se cuentan todas las interfaces legibles por la máquina (por ejemplo: archivos de datos de cinta o disco) que se utilizan para transmitir información a otro sistema.

Para calcular el Punto Función se utilizará la fórmula de la ecuación (1) :

$$PF = Total * [X + (Min_{(y)} * \sum F_i)]$$

Donde:

- PF : Medida de la funcionalidad.
- Total: Nivel de complejidad del sistema con respecto al usuario, es la suma de los valores de entradas, salidas, peticiones, archivos e interfaces externas.
- X : Ajuste de complejidad según el dominio de la información, el grado de confiabilidad estimada varía entre 1 a 100%.
- $Min_{(y)}$: Error mínimo aceptable al de la complejidad, probabilidad subjetiva estimada del dominio de la información, esta tasa de error estimado es del 1%.
- F_i : Son los valores de ajuste de complejidad, donde $(1 \leq i \leq 14)$.

Para calcular el punto de función se tiene que realizar el cálculo de la cuenta total con los factores de ponderación según Pressman, especificados en la siguiente tabla:

Tabla 4.1.

Factores de Ponderación de Punto de Función

PARÁMETROS DE MEDICIÓN	CUENTA	FACTORES DE PONDERACIÓN			TOTAL
		Simple	Medio	Complejo	
Entradas de Usuario	6	3	4	6	24
Salidas de Usuario	9	4	5	7	45
Peticiones de Usuario	12	3	4	6	48
Archivos	9	7	10	15	90
Interfaces Externas	0	5	7	10	0
Cuenta Total					207

Para determinar los valores de ajuste de complejidad se indica según corresponda a las preguntas de la siguiente tabla:

Tabla 4.2.

Ajuste del Factor de Complejidad

Importancia		0	1	2	3	4	5	x
Nº	FACTORES DE COMPLEJIDAD	Sin Influencia	Incidental	Moderado	Medio	Significativo	Esencial	F_i
1.	¿Requiere el sistema copias de seguridad y de recuperación fiables?					x		4
2.	¿Se requiere comunicación de datos?						x	5
3.	¿Existen funciones de procesamiento distribuido?					x		4
4.	¿Es crítico el rendimiento?				x			3
5.	¿Se ejecuta el sistema en un entorno operativo existente y fuertemente utilizado?					x		4
6.	¿Requiere el sistema entrada de datos interactiva?					x		4
7.	¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas o variadas opciones?				x			3
8.	¿Se actualizan los archivos maestros de forma interactiva?					x		4
9.	¿Son complejos las entradas, las salidas, los archivos o peticiones?					x		4
10.	¿Es complejo el procesamiento interno del sistema?						x	5
11.	¿Se ha diseñado el código para ser reutilizable?					x		4
12.	¿Están incluidas en el diseño la conversión y la instalación?				x			3
13.	¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?					x		4
14.	¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?						x	5
Factores de complejidad		Total (ΣF_i)						56

Con la obtención de los anteriores datos y considerando un grado de confiabilidad mínimo del 65%, se calculará el valor del punto de función reemplazando en la ecuación general, los valores obtenidos en las tablas anteriores.

$$PF = \text{Total} * [0,65 + (Min_{(y)} * \sum F_i)]$$

$$PF = 207 * [0,65 + (0,01 * 56)]$$

$$PF = 207 * [0,65 + (0,56)]$$

$$PF = 207 * [1,21]$$

$$PF = 250,47$$

Según Pressman (2002), determina que la escala de punto de función se clasifica como se detalla a continuación:

- $PF > 300 =$ Optimo
- $200 < PF < 300 =$ Bueno
- $100 < PF < 200 =$ Suficiente
- $PF < 100 =$ Deficiente

Después de reemplazar en la formula la obtención de resultados y basándose en la escala de punto de función, el sistema tiene una funcionalidad buena ya que los puntos de función encontrados son de 250,47.

Considerando el máximo ajuste de la complejidad $\sum F_i = 70$ calculamos al 100% el nivel de confianza de siguiente manera:

$$PF_{max} = 207 * [0,65 + (0,01 * 70)] \quad (13)$$

$$PF_{max} = 207 * [0,65 + (0,70)]$$

$$PF_{max} = 207 * [1,35]$$

$$PF_{max} = 279,45$$

Con los máximos valores de ajuste de complejidad obtenidos, se puede ya calcular la funcionalidad real del sistema con la ecuación (2) :

$$Funcionalidad = \frac{PF}{PF_{max}}$$

$$Funcionalidad = \frac{250,47}{279,45}$$

$$Funcionalidad = 0.896 * 100$$

$$Funcionalidad = 89.6\%$$

Por lo tanto, con el resultado obtenido se concluye que la Funcionalidad del sistema es de aproximadamente un 90%, esto determina que el sistema responde de manera óptima y de forma satisfactoria a las funcionalidades requeridas por el usuario.

4.1.2. Confiabilidad

La confiabilidad permite evaluar la relación entre el nivel de funcionalidad y la cantidad de recursos usados, es decir, representa el tiempo que el software está disponible para su uso, la misma se calcula utilizando la privacidad de que un sistema presente fallas. En otras palabras, es la capacidad del software de mantener su nivel de ejecución bajo condiciones normales en un periodo de tiempo establecido.

A continuación, se muestra el nivel de confiabilidad del sistema con la ecuación (3) :

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

Donde:

- f : Es la funcionalidad del sistema.
- λ : Es la probabilidad de error que puede tener el sistema.
- t : Tiempo que dura una gestión en el sistema.

Considerando un periodo de 20 días como tiempo de prueba, donde se define que cada 10 ejecuciones se presente 1 falla. Y tomando en cuenta la funcionalidad del sistema del 89% obtenido en el punto anterior, se reemplaza estos datos en la ecuación general de confiabilidad:

$$F_{(t)} = 0,896 * 2,718^{(-0,1*20)}$$

$$F_{(t)} = 0,896 * 0,135$$

$$F_{(t)} = 0,12096 * (100)$$

$$F_{(t)} = 12,096 \%$$

Reemplazando en la fórmula de probabilidades de las ecuaciones (4) y (5) tenemos los siguiente:

$$P(T \leq t) = F(t) \rightarrow P(T \leq t) = 0,12096 = 12,096\%$$

$$P(T > t) = 1 - F(t) \rightarrow P(T > t) = 1 - 0,12096 = 0,879 \approx 88\%$$

Por lo tanto, interpretando los datos obtenidos podemos decir que la probabilidad de hallar una falla es de un 12% y que la confiabilidad del sistema es de aproximadamente un 88%, en un periodo de 20 días como tiempo de prueba.

4.1.3. Usabilidad

Usabilidad es la facilidad de uso, un conjunto de atributos relacionados con el esfuerzo necesario para su uso y en la valoración individual de tal uso, por un establecido o implicado conjunto de usuarios. Para determinar la usabilidad del sistema se utiliza la ecuación (6) :

$$FU = [(\sum X_i/n) * 100]$$

Donde:

- X_i : Es la sumatoria de valores
- n : Es el número de preguntas

Para determinar el porcentaje de la usabilidad del sistema se optó por realizar una encuesta a los usuarios del sistema sobre el manejo, la comprensión, y la facilidad de aprender el sistema. Para responder a las preguntas se debe considerar una escala de valores: 0 - 5

La siguiente tabla nos muestra los resultados de la encuesta que se realizó:

Tabla 4.3.

Encuesta de Usabilidad del Sistema

N°	PREGUNTAS	SI	NO	EVALUACIÓN
1	¿El acceso al sistema es complicado?	0	5	1
2	¿Puede utilizar el sistema con facilidad?	4	1	0,8
3	¿Son comprensibles las respuestas del sistema?	5	0	1
4	¿El sistema permitió la retroalimentación de información?	3	2	0,6
5	¿El sistema cuenta con interfaz gráfica agradable a la vista?	5	0	1
6	¿La respuesta del sistema es satisfactoria?	5	0	1
7	¿Le parecen complicadas las funciones del sistema?	2	3	0,6
8	¿Los resultados que proporciona el sistema facilitan el trabajo?	5	0	1
9	¿El sistema responde rápido a sus solicitudes?	4	1	0,8
10	¿Durante el uso del sistema se produjo errores?	1	4	0,8
Promedio Total			8,6	

Reemplazando en la fórmula de usabilidad tenemos lo siguiente:

$$FU = [(\sum X_i/n) * 100]$$

$$FU = [(8,6/10) * 100]$$

$$FU = [0,86 * 100]$$

$$FU = 86 \%$$

Por lo tanto, de acuerdo a los datos obtenidos en la tabla de usabilidad, se concluye que el sistema tiene una usabilidad del 86%, esto indica que el usuario de la aplicación tiene una conformidad hacia el sistema, que refleja la comprensión o entendimiento del mismo con respecto a la capacidad del sistema.

4.1.4. Mantenibilidad

La Mantenibilidad se refiere a los atributos que permiten medir el esfuerzo necesario para realizar modificaciones al software, ya sea por la corrección de errores o por el incremento de funcionalidad.

Para hallar mantenibilidad del sistema se utiliza el índice de madurez de software (IMS), que proporciona una indicación de la estabilidad de un producto de software. El índice de madurez del software se calcula con la ecuación (7) :

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

Recopilando la información requerida por la fórmula, se obtuvo la información que se muestra en la siguiente tabla:

Tabla 4.4.

Valores Determinantes en la Mantenibilidad

SÍMBOLO	DESCRIPCIÓN	VALOR
Mt	Número de módulos total de la versión actual.	10
Fm	Número de módulos de la versión actual que se modificaron.	0
Fa	Número de módulos de la versión actual que se agregaron.	1
Fe	Número de módulos en la versión anterior que se eliminaron de la versión actual.	0

Ahora se calcula el índice de madurez de software, usando los valores obtenidos:

$$IMS = \frac{10 - (0 + 1 + 0)}{10}$$

$$IMS = \frac{10 - (1)}{10}$$

$$IMS = \frac{9}{10}$$

$$IMS = 0,9 * 100 = 90\%$$

Por lo tanto, el sistema tiene un índice de estabilidad de un 90% que es la facilidad de mantenimiento, el 10% restante es el margen de error correspondiente a los cambios y modificaciones.

4.1.5. Portabilidad

Se evalúa la oportunidad para adaptar el software a diferentes ambientes sin necesidad de aplicarle modificaciones, es decir ser transferido de forma efectiva y eficiente de un entorno hardware, software, operacional o de utilización a otro.

Para poder medir la portabilidad del sistema se usará la siguiente formula, que indica el grado de portabilidad que tiene un software mediante la ecuación (8) :

$$GP = 1 - (ET/ER)$$

Donde:

- *ET*: Medida de los recursos necesarios para llevar el sistema a otro entorno.
- *ER*: Medida de los recursos necesarios para crear el sistema en el entorno residente.
- Si $GP > 0$, la portabilidad es más rentable que el redesarrollo.
- Si $GP = 1$, la portabilidad es perfecta.
- Si $GP < 0$, el redesarrollo es más rentable que la portabilidad.

Entonces si la factibilidad estimada para trasportar el sistema es de 1 día a otro entorno y la implementación del mismo en otro entorno es de 1 mes; con esta información requerida por la fórmula, se procede a calcular el grado de portabilidad.

$$GP = 1 - \frac{1}{31}$$

$$GP = 1 - 0,0322$$

$$GP = 0,96 * 100 = 96\%$$

Por lo tanto, se concluye que el sistema tiene un grado de portabilidad del 96%, estando apto para funcionar bajo distintas plataformas.

4.1.6. Resultados de Calidad Global

La calidad global del sistema está directamente relacionada con el grado de satisfacción con el usuario que accede al sistema, siendo el total de todos los cálculos realizados según lo propuesto por el estándar de calidad ISO 9126; de acuerdo a estos resultados obtenidos, se puede establecer la calidad del sistema en base a todos los parámetros medidos anteriormente, que se detallan a continuación.

Tabla 4.5.

Resultados de Métricas de Calidad

CARACTERÍSTICAS	RESULTADOS
Funcionalidad	90%
Confiabilidad	88%
Usabilidad	86%
Mantenibilidad	90%
Portabilidad	96%
Calidad Global	90%

Finalmente, con el resultado obtenido se llega a la conclusión de que la calidad del sistema es de un 90%, lo que se interpreta como la satisfacción que tiene un usuario al interactuar con el sistema.

4.2. SEGURIDAD DEL SISTEMA

La seguridad en el área de la informática como en las otras áreas, se basa en la protección de los activos, estos activos pueden ser elementos tan tangibles como un servidor o una base de datos, o pueden ser la reputación de una empresa. Generalmente se puede evaluar la seguridad de un activo en base a tres aspectos principales: Integridad, Disponibilidad y Confidencialidad.

Estos tres aspectos a su vez dependen de otros tres elementos principales que engloban prácticamente todos los distintos controles que pueden establecer en un sistema informático.

- **Autenticación:** Los usuarios de aplicaciones o servicios deben ser identificadas de forma única, sean usuarios finales, otros servicios o computadoras externas.
- **Autorización:** No solo es necesario saber quiénes acceden a los activos de un sistema, también es necesario establecer que es o que lo pueden hacer ellos; un nivel de autorización dado, determina qué tipo de operaciones o transacciones pueden afectar un usuario sobre un recurso dado.
- **Registro de la auditoria:** Luego de efectuar una operación, es importante que esta sea registrada adecuadamente, es esencial si se quiere evitar el repudio de transacciones efectuadas por un usuario.
- **Base de datos:** Se toma en cuenta la vulnerabilidad de estos datos, por lo cual se contempla tomar las medidas de seguridad correspondientes.

4.2.1. Autenticación

Para la identificación se realiza el registro de un usuario con todos sus datos personales, que son guardados en la base de datos del servidor. De esta manera, para la autenticación se realiza la verificación de estos datos que fueron registrados con un usuario y contraseña, realizando su respectiva validación.

Figura 3.49.

Verificación y Autenticación



4.2.1.1. Encriptación

Se aplica la encriptación de seguridad para la contraseña del usuario, un dato de suma importancia para el ingreso al sistema. De este modo, el framework CodeIgniter que se utilizó ofrece el servicio de cifrado, que proporciona cifrado de datos simétrico (clave secreta) bidireccional; donde el servicio creará una instancia y/o inicializará un controlador de cifrado para adaptarlo a sus parámetros.

Figura 3.50.

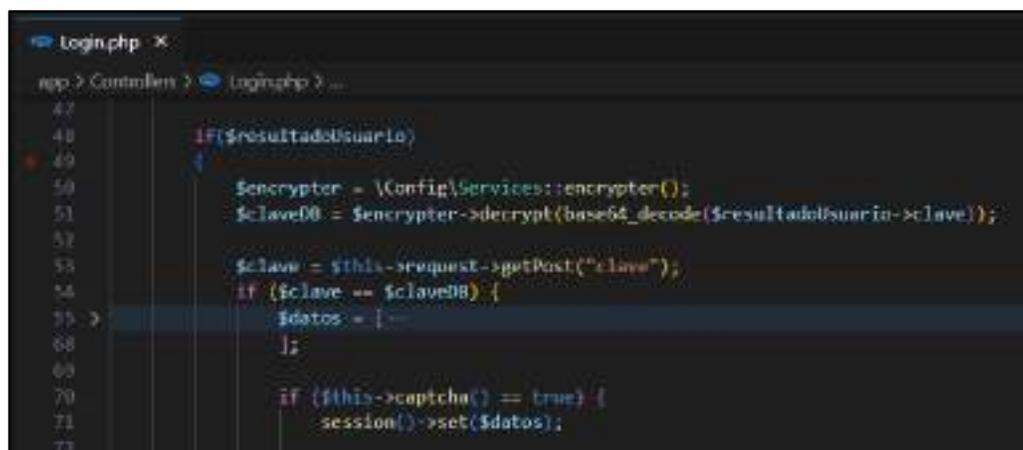
Encriptando Contraseña


```

Usuarios.php X
app > Controllers > Usuarios.php > ...
143
144     $usuario = new Usuario();
145
146     $clave = $this->request->getVar('txtClave');
147     $encrypter = \Config\Services::encrypter();
148     $password = base64_encode($encrypter->encrypt($clave));
149

```

Figura 3.51.

Desencriptando Contraseña


```

Login.php X
app > Controllers > Login.php > ...
47
48     IF($resultadoUsuario)
49     {
50         $encrypter = \Config\Services::encrypter();
51         $claveDB = $encrypter->decrypt(base64_decode($resultadoUsuario->clave));
52
53         $clave = $this->request->getPost("clave");
54         if ($clave == $claveDB) {
55             $datos = [
56                 ];
57
58             if ($this->captcha() == true) {
59                 session()->set($datos);
60             }
61         }
62     }
63

```

4.2.2. Autorización

Se realizó un control a través de transacciones autorizadas (privilegios) los cuales se definieron para cada tipo de usuario. En el sistema se tiene tres tipos de usuarios, a cada tipo de usuario se le identifico las acciones que puede realizar en su respectiva área.

- Administrador: Es el encargado de coordinar y supervisar los aspectos relacionados con los pedidos y distribuciones de los cilindros de oxígeno.
- Técnico: Es el encargado de controlar y registrar todas las entradas y salidas de cilindros de oxígeno del inventario de almacén.

4.3. ESTIMACIÓN DE COSTO

Después de la elaboración e implementación del sistema, el siguiente paso y uno de los más importantes es el análisis de costo y beneficio, que consiste en determinar con cierto grado de certeza los recursos de hardware y software, costo, tiempo y esfuerzo necesarios para el desarrollo de los mismos. Para la estimación se dará uso del modelo constructivo de costes COCOMO II (Constructive Cost Model).

4.3.1. Modelo COCOMO II

Es un modelo que permite realizar estimaciones en función al tamaño del software y de un conjunto de factores de costo y beneficio; los factores de costos, describen aspectos relacionados con la naturaleza del producto, hardware utilizado, personal involucrado y las características del proyecto.

COCOMO II posee tres modelos de estimación: básico, intermedio y detallado; los mismos se representan con las ecuaciones (9), (10) y (11) :

$$E = a * (KLDC)^b * FAE$$

$$T = c * (E)^d$$

$$NP = E/T$$

Dónde:

- *E*: Esfuerzo requerido por el proyecto expresado en persona/mes.
- *T*: Tiempo requerido por el proyecto expresado en meses.
- *NP*: Número de personas requeridas para el proyecto.
- *a, b, c, d*: Constantes con valores definidos según cada sub-modelo.
- *KLDC*: Cantidad de líneas de código distribuidas en miles.
- *FAE*: Ajuste del factor de complejidad.

4.3.1.1. Modelos de Desarrollo

- Modo Orgánico: Es un pequeño grupo de programadores experimentados, desarrollando proyectos de software en un entorno familiar.
- Modo Semi-Acoplado: Corresponde a un esquema intermedio entre el modo orgánico y el rígido, el grupo de desarrollo puede incluir una mezcla de personas experimentadas y no experimentadas.
- Modo Empotrado: El proyecto tiene fuertes restricciones, que pueden estar relacionadas con la funcionalidad y/o pueden ser técnicas.

A continuación, se describe las constantes de acuerdo con los modos mencionados anteriormente.

Tabla 4.6.

Coeficientes del Método COCOMO II

PROYECTO DE SOFTWARE	a	b	c	d
Orgánico	2,4	1,05	2,5	0,38
Semi - acoplado	3,0	1,12	2,5	0,35
Empotrado	3,6	1,20	2,5	0,32

Nota: Constantes de Complejidad, Pressman, 2005, Ingeniería de Software

Por lo tanto, tomando como base esta información, se optará por el coeficiente Orgánico, utilizando sus valores correspondientes en los cálculos próximos.

4.3.1.2. Costo del Desarrollo del Software

Para el cálculo del desarrollo del software, se tiene como partida el punto de función obtenido anteriormente en la funcionalidad del sistema, cuyo valor se expresa a través de la ecuación (1) : $PF = 250,47$

Este resultado se debe convertir a KLDC (Kilos de Líneas de Código), para ello se utiliza la siguiente la tabla.

Tabla 4.7.

Factor de Conversión LDC/PF

LENGUAJE	NIVEL	FACTOR LDC/PF
C	2.5	128
Java	6	53
Ansi Cobol	3	107
Visual Basic	7	46
ASP	9	36
PHP	11	29
Visual C++	9.5	34
SQL	12	12

Nota: Factor LCD/PF de lenguajes de programación, QSM, 2017, McLean

Entonces, realizando los cálculos necesarios y escogiendo el valor del lenguaje de programación PHP de la tabla se tendrá:

$$LDC = PF * Factor LCD/PF \quad (14)$$

$$LDC = 250,47 * 29$$

$$LDC = 7263,63$$

Calculando el número de líneas distribuidas en el sistema KLDC se tiene:

$$KLDC = LDC / 1000 \quad (15)$$

$$KLDC = 7263,63 / 1000$$

$$KLDC = 7,263$$

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

Tabla 4.8.

Valores de Atributo de Costes FAE.

CONDUCTORES DE COSTE	VALORACIÓN					
	Muy bajo	Bajo	Normal	Alto	Muy alto	Extra alto
Atributos del Software						
Fiabilidad requerida del software	0,75	0,88	1,00	1,15	1,40	-
Tamaño de la base de datos	-	0,94	1,00	1,08	1,16	-
Complejidad del producto	0,70	0,85	1,00	1,15	1,30	1,65
Atributos del Hardware						
Restricciones del tiempo de ejecución	-	-	1,00	1,11	1,30	1,66
Restricciones del almacenamiento principal	-	-	1,00	1,06	1,21	1,56
Volatilidad de la máquina virtual	-	0,87	1,00	1,15	1,30	-
Tiempo de respuesta del ordenador	-	0,87	1,00	1,07	1,15	-
Atributos del Personal						
Capacidad de análisis	1,46	1,19	1,00	0,86	0,71	-
Experiencia en la aplicación	1,29	1,13	1,00	0,91	0,82	-
Capacidad de los programadores	1,42	1,17	1,00	0,86	0,70	-
Experiencia en la máquina virtual	1,21	1,10	1,00	0,90	-	-
Experiencia en el lenguaje de programación	1,14	1,07	1,00	0,95	-	-

Atributos del Proyecto						
Prácticas de programación modernas	1,24	1,10	1,00	0,91	0,82	-
Utilización de herramientas software	1,24	1,10	1,00	0,91	0,83	-
Restricciones de tiempo de desarrollo	1,23	1,08	1,00	1,04	1,10	-
TOTAL	FAE = 0,486					

Nota: Ajuste del factor de complejidad, Barry Boehm, 1981, COCOMO 81

Por lo cual, el resultado de la multiplicación de los valores evaluados anteriormente en la tabla, se considera el factor de ajuste:

$$FAE = 0,486$$

A continuación, se realizarán los cálculos de costo del sistema a partir de todos los resultados obtenidos anteriormente, reemplazando estos valores en las ecuaciones nombradas en un principio.

- Calculando el Esfuerzo requerido:

$$E = 2,4 * (7,263)^{1,05} * 0,486$$

$$E = 2,4 * 8,019 * 0,486$$

$$E = 9,353$$

$$E \approx 9 \text{ Persona/Mes}$$

- Calculando el Tiempo de desarrollo:

$$T = 2,5 * (9,353)^{0,38}$$

$$T = 2,5 * 2,338$$

$$T = 5,84$$

$$T \approx 6 \text{ Meses}$$

- Calculando el Personal promedio:

$$NP = 9,353/5,84$$

$$NP = 1,6$$

$$NP \approx 2 \text{ Personas}$$

De esta manera, con los cálculos obtenidos se proseguirá a estimar el Costo Total del sistema, basándose en el salario mínimo actual del país que es de Bs. 2.500 mensuales, donde esta cifra será tomada en cuenta para la siguiente estimación:

$$CT = NP * T * Sueldo Mes \quad (16)$$

$$CT = 2 * 6 * 2.500$$

$$CT = 30.000 \text{ Bolivianos}$$

Por lo tanto, el costo estimado de desarrollo del software es de unos Bs. 30.000, equivalente en dólares americanos a \$us 4.342 aproximadamente.



CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5. CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

Se desarrollo un Sistema Web de control de entrada y salida de cilindros de oxígeno mediante código QR, para el Laboratorio Industrial de Oxígeno Medicinal - "G.A.M.E.A.", que permite tener un registro adecuado de la distribución de acuerdo a los pedidos de los distintos centros de salud, el cual proporciona una información precisa del inventario de cilindros garantizando así una gestión eficiente.

Tomando en cuenta los objetivos planteados en el presente proyecto, se tiene las siguientes conclusiones:

- Se recolectó información del proceso de entrada y salida de los cilindros, que determinó los requerimientos del método de control a partir de un análisis previo.
- Se diseñó una interfaz gráfica sencilla e intuitiva, que facilitó la interacción entre el sistema y el usuario, agilizando los tiempos de acceso rápido a la información.
- Se implementó el registro de los cilindros de oxígeno generando su respectivo código QR, que facilitó su identificación y localización de forma rápida y sencilla.
- Se automatizó el registro del inventario, logrando tener una información adecuada y precisa de todos los cilindros de oxígeno, conociendo en todo momento su disponibilidad.
- Se desarrolló un módulo para el pedido de cilindros de oxígeno, que aceleró y mejoró el proceso de solicitud, facilitando el trabajo a los respectivos centros de salud.
- Se sistematizó el proceso de lectura y almacenamiento de cilindros mediante QR, optimizando el registro de las operaciones de recarga, entrada y salida.
- Se consiguió generar reportes en base a las distribuciones realizadas de los cilindros de oxígeno, que ayuda en la toma de decisiones acertadas y oportunas.

En conclusión, el sistema desarrollado cumple con todos los objetivos planteados, logrando cubrir satisfactoriamente todas las necesidades y requerimientos de los usuarios en el Laboratorio Industrial de Oxígeno Medicinal – “G.A.M.E.A”.

5.2. RECOMENDACIONES

Finalmente, tras la conclusión del presente proyecto, se sugiere tomar en cuenta las siguientes recomendaciones, con el fin de buscar el mejoramiento del sistema:

- Capacitar a los usuarios para poder operar el sistema de forma correcta y adecuada, basándose en el manual de usuario proporcionado.
- Modificar las contraseñas de acceso al sistema de forma periódica, para proteger las cuentas de usuario dando mayor seguridad al sistema.
- Resguardar toda la información del sistema, realizando copias de seguridad de la base de datos a través del administrador general.
- Implementar en el sistema un módulo de venta de cilindros de oxígeno a la población que lo requiera, contando con una facturación electrónica, para tener un control total de las operaciones que realiza el laboratorio.
- Desarrollar de aplicación móvil vinculada al sistema web, para mejorar y agilizar el proceso de escaneo de los cilindros de oxígeno mediante el lector QR, además de ser más práctico y fácil de manipular para el personal.
- Utilizar las mismas herramientas de desarrollo web que se utilizaron, en caso de alguna modificación o actualización en el sistema, según las nuevas necesidades que puedan llegar a surgir en un futuro.
- Realizar el mantenimiento correspondiente a los equipos de cómputo con los que cuenta el laboratorio, para un buen funcionamiento del Sistema Web.



BIBLIOGRAFÍA Y WEBGRAFÍA

BIBLIOGRAFÍA

- Craig Larman, (2002). *UML y Patrones: Una Introducción al Análisis y Diseño Orientado a Objetos y al Proceso Unificado*. Segunda Edición: Madrid 2003, Pearson
- Fowler, M. & Kendall, S., (1999). *UML Gota a Gota*. Primera Edición: México 2000, Pearson Adisson Wesley
- Grady Booch, P.H., (1996). *Análisis y Diseño Orientado a Objetos*. Segunda Edición: Madrid 2001, Adisson Wesley
- Grady Booch, P.H., (1999). *El Proceso Unificado de Desarrollo de Software*. Primera Edición: Madrid 2000, Adisson Wesley
- Johansen, O., (1982). *Introducción a la teoría General de Sistemas*. Primera Edición: México 2015, Limusa
- Kendall, K. & Kendall, J., (1988). *Análisis y Diseño de Sistemas*. Octava Edición: México 2011, Pearson
- Pressman, R.G., (1982). *Ingeniería del Software: Un enfoque Practico*. Séptima Edición: México 2010, McGraw-Hill
- Sampieri, R.H., (1991). *Metodología de la Investigación*. Primera Edición: México 1997, McGraw-Hill
- Sommerville, I. (2011). *Ingeniería de Software*: Novena Edición: México 2011, Pearson
- Admin. (21 de Agosto de 2015). *AddApp*. Recuperado el Septiembre de 2023, de <https://www.addappto.com/que-es-un-sistema-web/>
- Alvarez, M. A. (23 de Noviembre de 2009). *DesarrolloWeb*. Recuperado el Agosto de 2023, de <https://desarrolloweb.com/articulos/codeigniter.html>
- APPYWEB. (2013). Recuperado el Mayo de 2023, de <https://www.appyweb.es/diccionario/codeigniter/#:~:text=Es%20una%20framework%20PHP%20de,de%20programaci%C3%B3n%20de%20las%20aplicaciones>
- Ariagna, R. D. (2011). *EcuRed*. Recuperado el Mayo de 2023, de https://www.ecured.cu/Agile_Unified_Process
- Ayudaley. (14 de Julio de 2020). Recuperado el Septiembre de 2023, de <https://ayudaleyprotecciondatos.es/2020/07/14/seguridad-de-la-informacion/>
- Beetrack. (03 de Febrero de 2020). *DispatchTrack*. Recuperado el Septiembre de 2023, de <https://www.beetrack.com/es/blog/gestion-de-produccion-y-operaciones>

- Calle, D. M. (2014). *Library*. Recuperado el Agosto de 2023, de <https://1library.co/article/agile-unified-process-aup-metodolog%C3%ADas-desarrollo-software.zllnke2z>
- Chernyak, A. Z. (09 de Mayo de 2023). *ZapTest*. Recuperado el Septiembre de 2023, de Unlited Software Automation: <https://www.zaptest.com/es/pruebas-de-caja-blanca-que-es-como-funciona-retos-metricas-herramientas-y-mas>
- Codelgniter, F. (2019). *Codeigniter*. Recuperado el Agosto de 2023, de https://codeigniter.com/user_guide/intro/index.html
- Códigos QR. (2009). Recuperado el Mayo de 2023, de <https://www.codigos-qr.com/>
- Cordero, J. L. (2005). *Ingenieria de Software*. Recuperado el Agosto de 2023, de https://ingenieriadesoftware.mex.tl/63758_aup.html
- CRT, C. P. (08 de Mayo de 2011). *Check Point*. Recuperado el Septiembre de 2023, de Software Technologies: <https://www.checkpoint.com/es/cyber-hub/cyber-security/what-is-white-box-testing/>
- DesarrolloWeb. (16 de Marzo de 2020). *Ionos*. Obtenido de DigitalGuide: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/codeigniter-framework-php-rapido-y-versatil/>
- Diaz, R. R. (06 de Enero de 2023). *EMS*. Recuperado el Octubre de 2023, de EMS Solutions International: <https://emssolutionsint.blogspot.com/2011/01/cuanto-durara-el-o2-dependiendo-del.html>
- Escalante, M. (01 de Junio de 2023). *abc xperts*. Recuperado el Octubre de 2023, de <https://abcxperts.com/que-es-la-triada-en-seguridad-de-la-informacion/>
- Eserp. (17 de Septiembre de 2021). *Eserp*. Recuperado el Septiembre de 2023, de <https://es.eserp.com/articulos/metodo-control-inventarios/>
- Etecé, E. (5 de Agosto de 2021). *Concepto*. Recuperado el Agosto de 2023, de <https://concepto.de/base-de-datos/>
- Figuroa, M. A. (30 de Enero de 2012). *Unam*. Recuperado el Septiembre de 2023, de NacionMulticultural: <https://www.nacionmulticultural.unam.mx/empresasindigenas/docs/2094.pdf>
- Garcia, J. (15 de Julio de 2021). *TecnoSimple*. Recuperado el Septiembre de 2023, de <https://tecno-simple.com/que-son-las-metricas-de-software/>
- Garcia, J. (01 de Febrero de 2022). *OASA*. Recuperado el Octubre de 2023, de <https://www.oasanorte.com/blogs/blog/oxigeno-medicinal-que-es-diferencias-y-especificaciones>

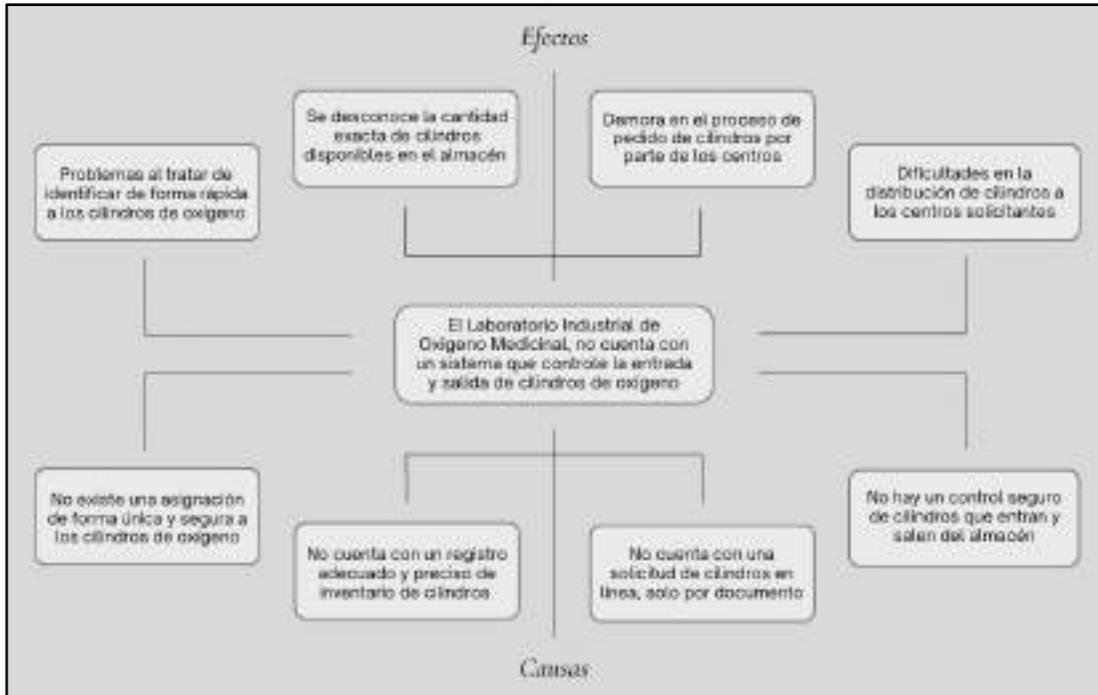
- Gómez y Migani, A. y. (29 de Mayo de 2017). *EcuRed*. Recuperado el Septiembre de 2023, de https://www.ecured.cu/COCOMO_II
- Gracia, L. (07 de Febrero de 2012). *Un Poco de Java*. Obtenido de <https://unpocodejava.com/2012/02/07/modelos-de-estimacion-un-poco-sobre-cocomo-ii/>
- López, B. S. (24 de Julio de 2019). *IngenieriaIndustrial*. Recuperado el Septiembre de 2023, de <https://www.ingenieriaindustrialonline.com/gestion-de-almacenes/que-es-la-gestion-de-almacenes/>
- Martinez, J. G. (28 de Julio de 2022). *DeltaProtect*. Recuperado el Septiembre de 2023, de <https://www.deltaprotect.com/blog/que-es-iso-27001>
- Mendoza, M. L. (16 de Julio de 2020). *OpenWebinars*. Recuperado el Agosto de 2023, de <https://openwebinars.net/blog/que-es-un-lenguaje-de-programacion/>
- Muro, A. (5 de Julio de 2018). *Procomsys*. Recuperado el Agosto de 2023, de <https://procomsys.wordpress.com/2018/07/05/diagramas-uml-lenguaje-unificado-de-modelado/>
- NQA. (2013). Recuperado el Mayo de 2023, de <https://www.nqa.com/es-mx/certification/standards/iso-27001#:~:text=La%20ISO%2027001%3A2013%20es%20la%20norma%20internacional%20que%20proporciona,informaci%C3%B3n%20as%C3%AD%20como%20cumplimiento%20legal>
- Olmos, S. O. (Enero de 2016). *Cufcd*. Recuperado el Octubre de 2023, de CEMA: <http://www.cufcd.edu.mx/calidad/v20/documentacion/CM/CEMA-MN-OP-3.pdf>
- Ordóñez, J. L. (05 de Septiembre de 2012). *Acta*. Recuperado el Septiembre de 2023, de https://www.acta.es/medios/articulos/comunicacion_e_informacion/063009.pdf
- Pérez, S. D. (09 de Septiembre de 2021). *Intelequia*. Recuperado el Agosto de 2023, de <https://intelequia.com/blog/post/gestor-de-base-de-datos-qu%C3%A9-es-funcionalidades-y-ejemplos>
- PHP. (2001). *php.net*. Recuperado el Mayo de 2023, de <https://www.php.net/manual/es/intro-what-is.php>
- Pilot. (Noviembre de 2015). *LogisPyme*. Recuperado el Septiembre de 2023, de <https://logispyme.files.wordpress.com/2015/11/pedidos1y2.pdf>
- Quiróz, D. S. (Septiembre de 2018). *UML*. Recuperado el Agosto de 2023, de openwebinars.net: <https://openwebinars.net/blog/que-es-uml-unified-modeling-language>

- Robledano, Á. (24 de Septiembre de 2019). *OpenWebinars*. Recuperado el Agosto de 2023, de <https://openwebinars.net/blog/que-es-mysql/>
- Rodriguez, J. (18 de Julio de 2023). *HubSpot*. Recuperado el Septiembre de 2023, de <https://blog.hubspot.es/sales/que-es-control-de-inventarios>
- Roque, D. D. (25 de Noviembre de 2014). *Gestiopolis*. Recuperado el Septiembre de 2023, de <https://www.gestiopolis.com/estimacion-de-costos-de-desarrollo-de-software/>
- Santander, U. (21 de Diciembre de 2020). *BecasSantander*. Recuperado el Septiembre de 2023, de <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html>
- Sentrio. (10 de Enero de 2023). *SENTRIO*. Recuperado el Octubre de 2023, de <https://sentrio.io/blog/pruebas-estres-y-escalabilidad-devops/>
- Soriano, F. (26 de Noviembre de 2018). *IONOS*. Recuperado el Septiembre de 2023, de Digital Guide: <https://www.ionos.com/es-us/digitalguide/paginas-web/desarrollo-web/uml-lenguaje-unificado-de-modelado-orientado-a-objetos/>
- Souza, I. d. (9 de Marzo de 2020). *RockContent*. Recuperado el Agosto de 2023, de <https://rockcontent.com/es/blog/php/>
- TechTarget. (2013). *ComputerWeekly*. Recuperado el Mayo de 2023, de <https://www.computerweekly.com/es/definicion/MySQL>
- Técnicos, C. (16 de Octubre de 2020). *LoadView*. Recuperado el Septiembre de 2023, de <https://www.loadview-testing.com/es/blog/tipos-de-pruebas-de-software-diferencias-y-ejemplos/>
- Terrera, G. (26 de Febrero de 2017). *TestingBaires*. Recuperado el Septiembre de 2023, de <https://testingbaires.com/pruebas-caja-negra-enfoque-practico/>
- Verity. (28 de Julio de 2022). Obtenido de <https://www.verity.cl/que-es-norma-iso-iec-9126-2001/>
- Yano, N. (02 de Marzo de 2023). *Bsale*. Recuperado el Septiembre de 2023, de <https://www.bsale.com.mx/article/almacen-registros-de-entradas-y-salidas>

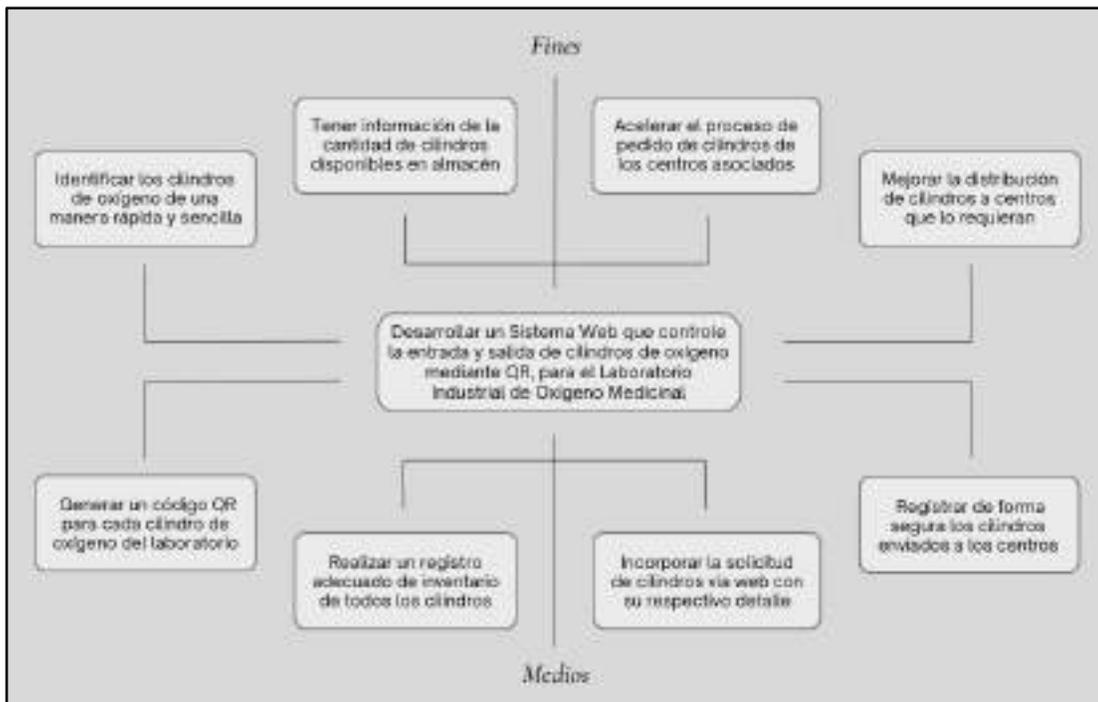


ANEXOS

Anexo 1. Árbol de Problemas



Anexo 2. Árbol de Objetivos



Anexo 5. Planillas en Excel

	A	B	C	D	E	F	G	H	I	J
1										
2	REPORTE DE ENTREGAS REALIZADAS EN LA GESTION 2023 EXPRESADO EN m3									
3	GESTION 2023	HOSPITAL HOLANDES	HOSPITAL LOS ANDES	HOSPITAL COREA	HOSPITAL JAPONES	CENTROS DE SALUD	SUMA 161		CANTIDAD EN M3 MENSUAL	TOTAL MENSUAL EN BOLIVIA
4	ENERO	2505	346	268	30	0	0		3149	Bs. 39.511,00
5	FEBRERO	2534	230	163	65	0	39		3031	Bs. 38.303,50
6	MARZO	3677	331	95	102	69	38		4312	Bs. 54.464,00
7	ABRIL	3822	636	243	138	37	50		4926	Bs. 62.026,00
8	MAYO	3813	1005	452	166	107	93		5636	Bs. 71.355,50
9	JUNIO	651	126	48	34	18	0		877	Bs. 10.995,00
10	JULIO	1361	483	102	116	77	84		2223	Bs. 28.510,50
11	AGOSTO	3025	433	227	146	44	50		3925	Bs. 49.625,00
12	SEPTIEME	3277	434	1406	306	66	113		5602	Bs. 70.961,00
13	OCTUBRE	3033	268	1268	269	83	84		5005	Bs. 63.458,50
14	NOVIEMB	0	0	0	0	0	0		0	Bs. 0,00
15	DICIEMBR	0	0	0	0	0	0		0	Bs. 0,00
16										
17	CANT. DE CILINDR TOTALE	4422	766	664	261	100	286	CANTIDAD DE CILINDRO	6499	
18	m3	27698	4292	4272	1372	501	551	CANTIDAD EN m3	38686	
19	BS.-	*****	*****	*****	*****	*****	*****	TOTAL EN	Bs. 492.318,30	
20	Reporte actualizado al 11 DE AGOSTO de 2023									
21										

MANUAL DE USUARIO

SISTEMA WEB DE CONTROL DE
ENTRADA Y SALIDA DE CILINDROS DE
OXÍGENO MEDIANTE CÓDIGO QR
“LIOM” - G.A.M.E.A

Versión v1

UNIDAD DE SISTEMAS

ÍNDICE

1. INTRODUCCIÓN	1
2. OBJETIVO	1
3. REQUERIMIENTOS	1
3.1. CONOCIMIENTOS BÁSICOS	1
3.2. REQUERIMIENTOS DE HARDWARE	1
3.3. REQUERIMIENTOS DE SOFTWARE	1
4. GENERALIDADES	1
4.1. INGRESO AL SISTEMA	1
4.2. INICIO DE SESIÓN	2
4.3. CERRAR SESIÓN	3
4.4. CAMBIAR CONTRASEÑA	3
5. ROL ADMINISTRADOR Y TÉCNICO	4
5.1. MENÚ LATERAL IZQUIERDO	4
5.2. PÁGINA PRINCIPAL	4
5.3. PANEL ADMINISTRAR	5
5.3.1. MÓDULO USUARIOS	5
5.3.1.1. CREAR NUEVO USUARIO	5
5.3.1.2. ACTUALIZAR DATOS DE UN USUARIO	6
5.3.1.3. VER DATOS DE ACCESO	6
5.3.1.4. HABILITAR Y DESHABILITAR UN USUARIO	6
5.3.2. MÓDULO CENTROS	7
5.3.2.1. CREAR NUEVO CENTRO	7
5.3.2.2. ACTUALIZAR DATOS DE UN CENTRO	8
5.3.2.3. HABILITAR Y DESHABILITAR UN CENTRO	8
5.3.3. MÓDULO CILINDROS	9
5.3.3.1. CREAR NUEVO CILINDRO	9
5.3.3.2. ACTUALIZAR DATOS DE UN CILINDRO	10
5.3.3.3. ELIMINAR UN CILINDRO	10
5.3.3.4. OBTENER QR DE CILINDROS	10
5.3.3.5. HABILITAR Y DESHABILITAR UN CILINDRO	11
5.4. PANEL ACCIONES	11
5.4.1. MÓDULO PEDIDOS	11
5.4.1.1. INFORMACIÓN DE UN PEDIDO	12
5.4.1.2. IMPRIMIR INFORME DE UN PEDIDO	12
5.4.2. MÓDULO DISTRIBUCIONES	12
5.4.2.1. INFORMACIÓN DE UNA DISTRIBUCIÓN	13
5.4.2.2. IMPRIMIR INFORME DE UN PEDIDO	13
5.5. PANEL REGISTROS	14
5.5.1. MÓDULO ALMACÉN	14
5.5.1.1. RECARGAR CILINDROS MEDIANTE QR	14
5.5.2. SALIDAS	15
5.5.2.1. REGISTRAR SALIDA MEDIANTE QR	15

5.5.3. ENTRADAS	15
5.5.3.1. REGISTRAR ENTRADA MEDIANTE QR	15
5.6. PANEL OTROS	16
5.6.1. EXTERNOS	16
5.6.1.1. AGREGAR PEDIDO EXTERNOS	16
5.6.1.2. CREAR NUEVO USUARIO EXTERNO	17
5.6.1.3. REGISTRAR SALIDA EXTERNA	17
5.6.2. PRECIOS	18
5.6.2.1. ACTUALIZAR PRECIOS DE CILINDROS	18
5.6.3. REPORTES	18
5.6.3.1. GLOBALES	18
5.6.3.2. PARCIALES	19
6. ROL RESPONSABLE	20
6.1. MENÚ LATERAL IZQUIERDO	20
6.2. PÁGINA PRINCIPAL	20
6.3. PANEL ACCIONES	21
6.3.1. MÓDULO PEDIDOS	21
6.3.1.1. INFORMACIÓN DE UN PEDIDO	21
6.3.1.2. IMPRIMIR INFORME DE UN PEDIDO	21
6.3.1.3. AGREGAR NUEVO PEDIDO	22
6.3.1.4. EDITAR REGISTRO DE UN PEDIDO	22
6.3.1.5. BORRAR REGISTRO DE UN PEDIDO	23
6.3.2. MODULO DISTRIBUCIONES	23
6.3.2.1. INFORMACIÓN DE UNA DISTRIBUCIÓN	24
6.3.2.2. IMPRIMIR INFORME DE UN PEDIDO	24
6.4. PANEL REGISTROS	24
6.4.1. MÓDULO ALMACÉN	24
7. PANEL EXTRAS	25
7.1. TUTORIALES	25

LABORATORIO INDUSTRIAL DE OXÍGENO MEDICINAL

1. INTRODUCCIÓN

El Sistema Web “LIOM” de control de entrada y salida de cilindros de oxígeno mediante código QR, es una herramienta informática fundamental para la administración eficiente de los cilindros en el almacén del laboratorio.

Este manual ha sido elaborado con la intención de facilitar al usuario la operación de los diferentes procesos que se administra en el sistema de información, para el uso correcto y apropiado del sistema.

2. OBJETIVO

Instruir a los respectivos usuarios con el uso correcto del sistema y la solución de los problemas que puedan suceder en cada operación. El manual facilita el proceso de creación, actualización en los respectivos módulos, describiendo las actividades de pedidos y distribuciones de cilindros de oxígeno, a fin de optimizar estos movimientos fundamentales que realiza el laboratorio.

3. REQUERIMIENTOS

3.1. CONOCIMIENTOS BÁSICOS

- Manejo de navegadores de Internet.
- Conocimiento básico en lector QR.

3.2. REQUERIMIENTOS DE HARDWARE

Los requisitos mínimos de hardware para el correcto funcionamiento del sistema son:

- Computadora de escritorio o personal (laptop).
- Conexión a internet.

3.3. REQUERIMIENTOS DE SOFTWARE

Sistema Operativo:

- Windows (Versión 7 o superior)

Navegadores:

- Google Chrome (Versión 3.5 o superior)
- Opera (Versión 3.5 o superior)
- Microsoft Edge (Versión 3.5 o superior)
- Mozilla Firefox (Versión 3.5 o superior)

4. GENERALIDADES

4.1. INGRESO AL SISTEMA

En la URL de su navegador, ingrese el siguiente enlace:

<https://gamealiom.000webhostapp.com/>

A continuación, le parecerá una ventana como la mostrada en (Fig.1).



Figura 1. Dirección URL del Sistema

4.2. INICIO DE SESIÓN

Seguidamente aparecerá la ventana de ingreso, mostrada en (Fig.2), donde cada usuario deberá ingresar sus datos correctamente.



Figura 2. Login del Sistema

IMPORTANTE:

Introducir el nombre de Usuario y Contraseña, presionar el botón “Iniciar sesión”. El sistema validará los datos introducidos, si sus datos son correctos ingresará al panel principal del sistema, caso contrario el sistema lo hará saber con un mensaje, como se muestra en la (Fig.3) y deberá intentar nuevamente. En caso de Ingresar los datos correctos inicia el sistema y vera el panel principal. (Fig.7)



Figura 3. Error de Autenticación

4.3. CERRAR SESIÓN

Para cerrar la sesión nos dirigimos a la parte superior derecha de la ventana para desplegar las opciones de usuario y elegimos Cerrar Sesión (Fig.4).

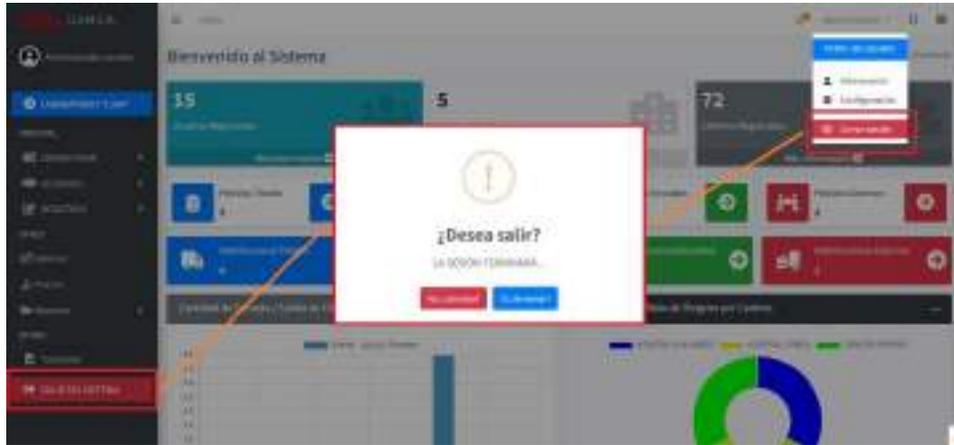


Figura 4. Salir de Sistema

IMPORTANTE:

Una vez finalizada la sesión nos reenviará a la ventana del “Login”, y nos volverá a pedir nuestro nombre de usuario y contraseña. (Cierra la sesión cada vez que dejara de manipular el sistema)

4.4. CAMBIAR CONTRASEÑA

Para cambiar el password o contraseña esta opción se encontrará a lado superior del botón de Cerrar Sesión (Perfil de Usuario). Esta opción se encontrará habilitada para todos los roles (Administrador, Técnico y Responsable).

Aparecerá una ventana nueva, la cual nos permitirá ingresar la nueva Contraseña y repetir la misma en cajas de texto. (Fig.5)



Figura 5. Opciones de Usuario

5. ROL ADMINISTRADOR Y TÉCNICO

5.1. MENÚ LATERAL IZQUIERDO

En este apartado muestra el nombre del Sistema, el nombre del usuario que inicio sesión, y el menú de opciones que dependerá del rol de usuario (Fig.6).

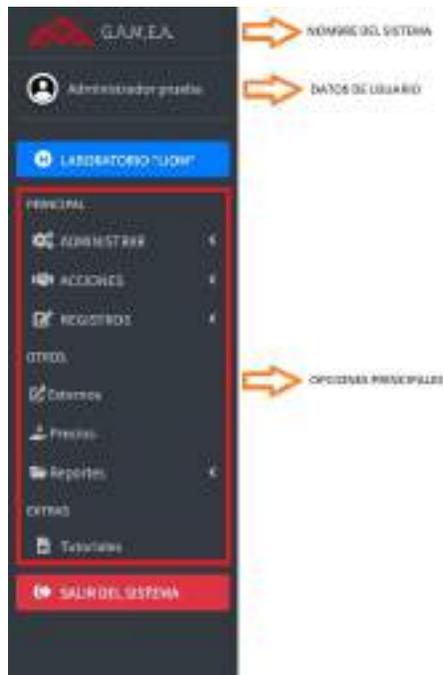


Figura 6. Menú del Sistema

5.2. PÁGINA PRINCIPAL

En la parte central de la página se muestra los accesos directos a cada opción del sistema y datos estadísticos de los registros realizados hasta el momento. (Fig.7).



Figura 7. Panel Principal

5.3.1.2. ACTUALIZAR DATOS DE UN USUARIO

Si desea corregir algún dato o modificarlo, debe presionar el botón despegable “Editar Información” en el campo de ACCIONES y le abrirá una ventana emergente, debe sobrescribir los datos ya registrados anteriormente y presionar el botón “Actualizar”, caso contrario presione “Cancelar”. (Fig.10)



Figura 10. Ventana de Modificar Usuario

5.3.1.3. VER DATOS DE ACCESO

Si por algún motivo, un usuario olvida y/o desconoce sus datos de acceso al sistema, debe presionar el botón despegable “Datos de Acceso” en el campo de ACCIONES y le abrirá una ventana emergente, ahí podrá ver la información correspondiente de cada usuario, para dejar de ver “Cerrar”. (Fig.11)

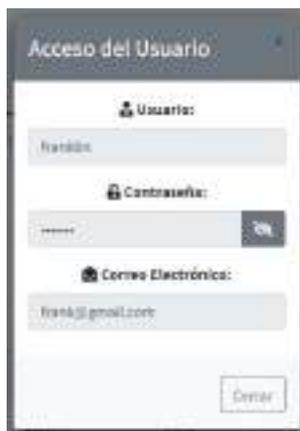


Figura 11. Ventana de Datos de Acceso

5.3.1.4. HABILITAR Y DESHABILITAR UN USUARIO

Si por motivos de contrato o cambio de personal, se debe deshabilitar a un usuario para que ya no tenga acceso al sistema, presionando el botón switch (interruptor) en el campo de ESTADO, y le abrirá una ventana emergente, presione “Si” para confirmar la acción, caso contrario presione “No”. (Fig.12)



Figura 12. Ventana de confirmación de Estado

5.3.2. MÓDULO CENTROS

En este módulo se muestra todos los centros registrados en el sistema con su respectiva información, los cuales estén asociados al laboratorio para los respectivos pedidos y distribuciones de cilindros de oxígeno. (Fig.13)



Figura 13. Página de Centros

5.3.2.1. CREAR NUEVO CENTRO

Para crear un nuevo usuario deberá hacer click en el botón “Nuevo Centro”, le abrirá una ventana emergente donde deberá llenar los campos y si desea registrarlo presione el botón “Guardar”, caso contrario presione “Cancelar”. (Fig.14)



Figura 14. Ventana de agregar Centro

IMPORTANTE:

Para registrar la ubicación del centro presione “Ver Mapa” donde se desplegará un plano de geolocalización, darle click en el lugar específico donde será representado por un marcador.

5.3.2.2. ACTUALIZAR DATOS DE UN CENTRO

Si desea corregir algún dato o modificarlo, debe presionar el botón despegable “Editar” en el campo de ACCIONES y le abrirá una ventana emergente, debe sobrescribir los datos ya registrados anteriormente y presionar el botón “Actualizar”, caso contrario presione “Cancelar”. (Fig.15)

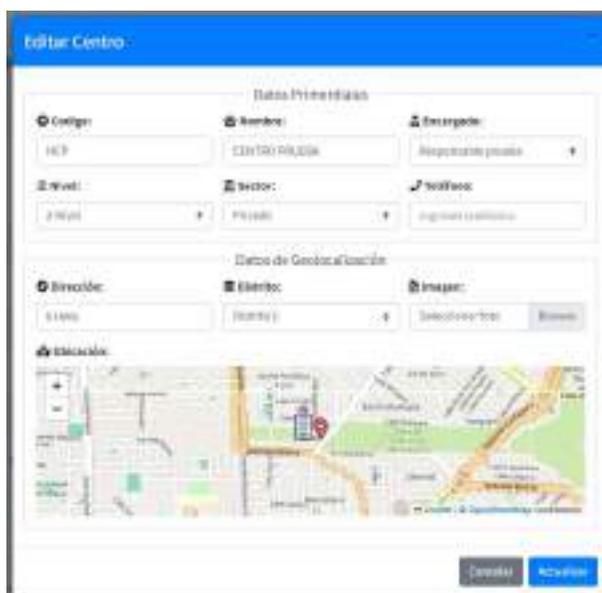


Figura 15. Ventana de modificar Centro

IMPORTANTE:

Para modificar o editar la ubicación del centro presione “Ver Mapa” donde se desplegará un plano de geolocalización, arrastre y suelte el marcador en específico al nuevo lugar de registro.

5.3.2.3. HABILITAR Y DESHABILITAR UN CENTRO

Si por motivos de acuerdos, terminar la asociación u otros, se debe deshabilitar un centro del sistema para que ya no sea tomado en cuenta, presionando el botón switch (interruptor) en el campo de ESTADO, y le abrirá una ventana emergente, presione “Si” para confirmar la acción, caso contrario presione “No”. (Fig.16)



Figura 16. Ventana de confirmación de Estado

5.3.3. MÓDULO CILINDROS

En este módulo se muestra todos los cilindros registrados en el sistema con su respectiva información, ya sean del dominio del laboratorio o de alguna persona externa. (Fig.17)



Figura 17. Página de Cilindros

5.3.3.1. CREAR NUEVO CILINDRO

Para crear un nuevo cilindro deberá hacer click en el botón “Nuevo Cilindro”, le abrirá una ventana emergente donde deberá llenar los campos y si desea registrarlo presione el botón “Guardar”, caso contrario presione “Cancelar”. (Fig.18)



Figura 18. Ventana de agregar Cilindro

IMPORTANTE:

- Una vez llenados los campos requeridos para un nuevo cilindro, el sistema generará de forma automática un código QR único, a partir del número de serie correspondiente al cilindro.
- Cada cilindro registrado entra de manera automática al inventario del almacén perteneciente al laboratorio, para tener información acerca de su ubicación y estado.

5.3.3.2. ACTUALIZAR DATOS DE UN CILINDRO

Si desea corregir algún dato o modificarlo, debe presionar el botón despegable “Editar” en el campo de ACCIONES y le abrirá una ventana emergente, debe sobrescribir los datos ya registrados anteriormente y presionar el botón “Actualizar”, caso contrario presione “Cancelar”. (Fig.19)



Figura 19. Ventana de modificar Cilindro

5.3.3.3. ELIMINAR UN CILINDRO

Si por motivos de daño, pérdida o durabilidad de un cilindro, es necesario darle de baja del sistema, para eso debe presionar el botón despegable “Eliminar” en el campo de ACCIONES y le abrirá una ventana emergente, presione “Si” para confirmar la acción, caso contrario presione “No”. (Fig.20)



Figura 20. Ventana de eliminar un cilindro

5.3.3.4. OBTENER QR DE CILINDROS

Si por algún motivo, el QR adjunto a un cilindro se desgasta o despegas, es necesario imprimir nuevos modelos con esta información, para eso debe seleccionar de la primera columna de la tabla los cilindros que requiera dándole click en su respectiva casilla, luego presionar el botón “Extraer QR”, se abrirá un pdf con la información esencial de cada cilindro, para poder descargarlo e imprimirlo posteriormente. (Fig.21)



Figura 21. Extraer QR de Cilindros

5.3.3.5. HABILITAR Y DESHABILITAR UN CILINDRO

Si por motivos de mantenimiento o recalibración, se debe deshabilitar un cilindro del sistema para que no sea utilizado hasta volverlo a habilitar, esto presionando el botón switch (interruptor) en el campo de ESTADO, donde le abrirá una ventana emergente, presione “Si” para confirmar la acción, caso contrario presione “No”. (Fig.22)



Figura 22. Ventana de confirmación de Estado

5.4. PANEL ACCIONES

5.4.1. MÓDULO PEDIDOS

En este módulo se muestra la información de todos los pedidos que realizan los centros a través de sus encargados asignados, así también los pedidos que se realizan personas externas. (Fig.23)



Figura 23. Página de Pedidos

IMPORTANTE:

Para autorizar la distribución de un pedido ir a la pestaña de PENDIENTES, presionar el botón “Aceptar” en el campo de ESTADO / ETAPA de la tabla para permitir la salida de cilindros, caso contrario presionar “Rechazar” para cancelar el pedido por algún motivo en particular; como se muestra en la siguiente imagen. (Fig.24)

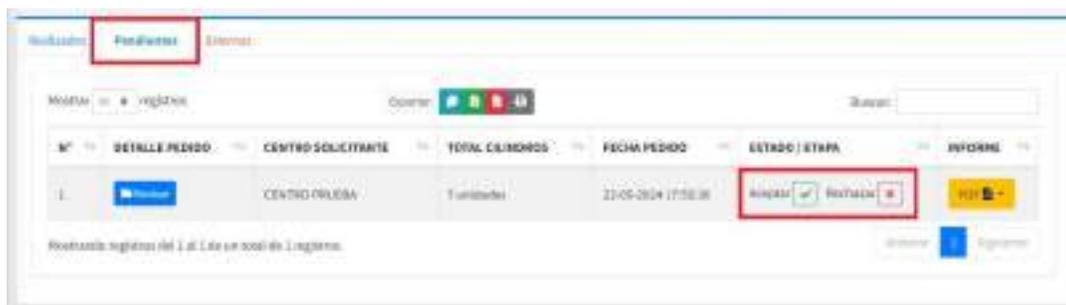


Figura 24. Aceptar o rechazar Pedido

5.4.1.1. INFORMACIÓN DE UN PEDIDO

Si desea ver el detalle en específico de un pedido, debe presionar el botón “Revisar” en el campo de DETALLE PEDIDO y le abrirá una ventana emergente, donde muestra la capacidad y cantidad de cilindros requeridos por parte del centro que realizó el pedido. (Fig. 15)



Figura 25. Detalle de un pedido

5.4.1.2. IMPRIMIR INFORME DE UN PEDIDO

Si desea imprimir el informe físico de un pedido como comprobante, debe presionar el botón despegable “PDF” en el campo INFORME de la tabla, donde debe elegir si desea membretado o no, una vez elegido se abrirá el documento en una página aparte para poder descargarlo e imprimirlo posteriormente. (Fig.23)

5.4.2. MÓDULO DISTRIBUCIONES

En este módulo se muestra la información de todas las distribuciones que se realizan a los centros, así también las distribuciones que se realizan a personas externas. (Fig.26)



Figura 26. Página de Distribuciones

5.4.2.1. INFORMACIÓN DE UNA DISTRIBUCIÓN

Si desea ver el detalle en específico de una distribución, debe presionar el botón “Revisar” en el campo de DETALLE PEDIDO y le abrirá una ventana emergente, donde muestra la serie de los cilindros de oxígeno que se enviaron a un centro en específico de acuerdo al pedido que realizó. (Fig.27)

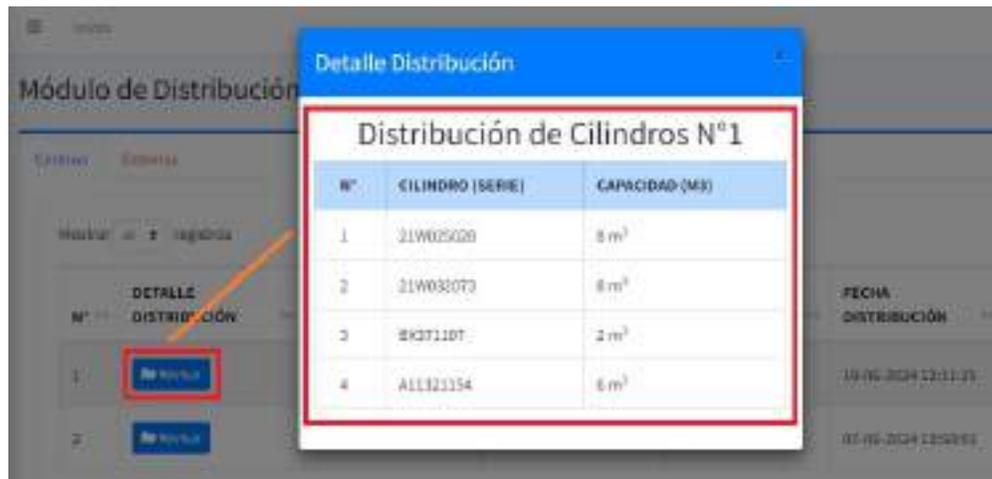


Figura 27. Detalle de una distribución

5.4.2.2. IMPRIMIR INFORME DE UN PEDIDO

Si desea imprimir el informe físico de una distribución como comprobante, debe presionar el botón despegable “PDF” en el campo IMPRIMIR INFORME de la tabla, donde debe elegir si desea membretado o no, una vez elegido se abrirá el documento para poder descargarlo e imprimirlo posteriormente. (Fig.26)

5.5. PANEL REGISTROS

5.5.1. MÓDULO ALMACÉN

En este módulo se muestra el inventario del almacén con la información acerca de todos los cilindros de oxígeno del laboratorio (Fig.28). Así también, en la segunda pestaña se muestra el porcentaje disponible de cilindros dentro del laboratorio, con respecto a su capacidad a través de una tabla. (Fig.29)



N°	CILINDRO (SERIE)	LOCALIZACIÓN	ESTADO DEL CILINDRO	FECHA RECARGA	ÚLTIMA SALIDA	ÚLTIMA ENTRADA
1	22000001	LABO	NO	10-01-2020 09:12	10-01-2020 09:12	10-01-2020 12:10
2	22000002	LABO	NO			10-01-2020 12:10
3	22000003	LABO				
4	22000004	LABO				
5	22000005	HOSPITAL MUNICIPAL				
6	22000006	LABO				
7	22000007	LABO				

Figura 28. Página de Almacén



N°	TRANSFORMACIONES	CANTIDAD	PORCENTAJE
1	NO RECARGADO	0	0.00%
2	CON RETORNO CARGA	1	100.00%
3	CON RETORNO CARGA	0	0.00%
4	CARGA RECARGADO	0	0.00%
5	CARGA RECARGA CARGA	0	0.00%
6	CON RETORNO CARGA	46	100.00%
7	CON RETORNO CARGA	0	0.00%
8	CON RETORNO CARGA	88	100.00%

Figura 29. Disponibilidad de Cilindros

5.5.1.1. RECARGAR CILINDROS MEDIANTE QR

Para registrar la recarga de oxígeno de un cilindro, presionar el botón “Recargar Cilindros” y le abrirá una ventana emergente, presione “Iniciar” para empezar a escanear el código QR de los cilindros ya recargados, presionar “Terminar” para concluir el registro y cerrar la ventana emergente. (Fig.28)

IMPORTANTE:

Realizar el escaneo a través de un dispositivo móvil, ya que el sistema está adaptado para funcionar correctamente por este medio, siendo más accesible y cómodo para el usuario.

5.5.2. SALIDAS

En este apartado se muestra todos los pedidos aceptados para su distribución, donde se podrá registrar la salida de los cilindros requeridos a través de un escáner QR.

5.5.2.1. REGISTRAR SALIDA MEDIANTE QR

Para registrar la salida de cilindros, presionar el botón “Registrar” y le abrirá una ventana emergente, presione “Iniciar” para empezar a escanear el código QR de los cilindros que serán distribuidos a partir de lo requerido en el pedido, una vez concluya con el total de los cilindros requeridos se mostrara el botón “Terminar”, presiónalo para terminar el escaneo y registrar la distribución. (Fig.30)



Figura 30. Salida de Cilindros

5.5.3. ENTRADAS

En este apartado se muestra todas las distribuciones devueltas y/o confirmadas por parte de los encargados de cada centro, donde se podrá registrar la entrada de los cilindros al almacén del laboratorio a través de un escáner QR.

5.5.3.1. REGISTRAR ENTRADA MEDIANTE QR

Para registrar la entrada de cilindros, presionar el botón “Registrar” y le abrirá una ventana emergente, presione “Iniciar” para empezar a escanear el código QR de los cilindros que pertenecen a cada distribución, una vez escanee los cilindros que retornaron se mostrara el botón “Terminar”, presiónalo para terminar el escaneo y registrar la entrada. (Fig.31)

IMPORTANTE:

No será necesario escanear todos los cilindros pertenecientes a cada distribución, esto debido a que no siempre retornan todos los enviados, sino solo los que ya fueron utilizados y vaciados de oxígeno en su totalidad.



Figura 31. Entrada de Cilindros

5.6. PANEL OTROS

5.6.1. EXTERNOS

5.6.1.1. AGREGAR PEDIDO EXTERNOS

Para crear un nuevo pedido deberá hacer click en el botón “Agregar Pedido”, le abrirá una ventana emergente donde deberá llenar los campos y si desea registrarlo presione el botón “Guardar”, caso contrario presione “Cancelar”. (Fig.32)

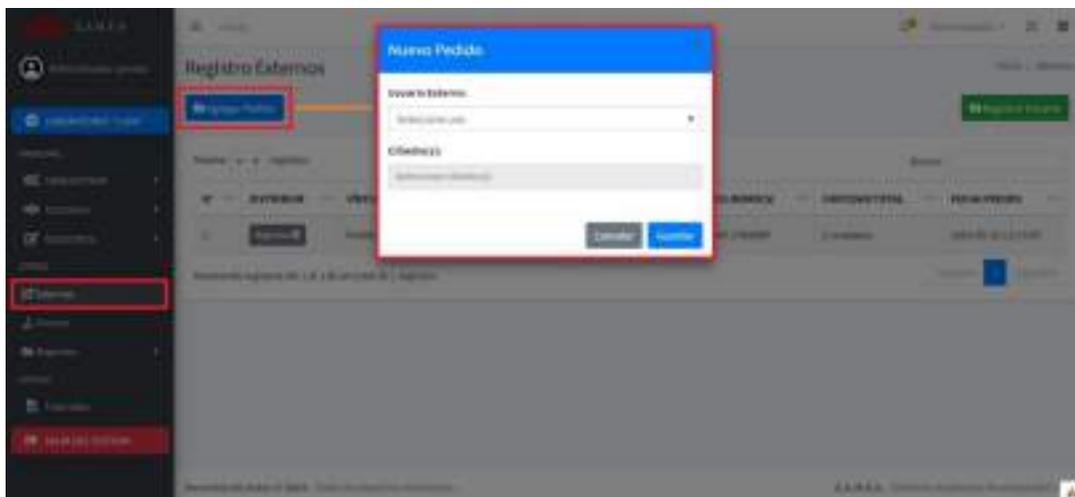


Figura 32. Registro Pedido Externo

IMPORTANTE:

Solo se podrá registrar un pedido externo en base a los usuarios externos que estén registrados. Al seleccionar al usuario, se desplegarán los cilindros registrados anteriormente y pertenecientes al mismo.

5.6.1.2. CREAR NUEVO USUARIO EXTERNO

Para crear un nuevo usuario deberá hacer click en el botón “Registrar Usuario”, le abrirá una ventana emergente donde deberá llenar los campos y si desea registrarlo presione el botón “Guardar”, caso contrario presione “Cancelar”. (Fig.33)



Figura 33. Registro Usuario Externo

5.6.1.3. REGISTRAR SALIDA EXTERNA

Para registrar la salida (venta) de cilindros, presionar el botón “Registrar” y le abrirá una ventana emergente, presione “Iniciar” para empezar a escanear el código QR de los cilindros que serán distribuidos a partir de lo requerido en el pedido, una vez concluya con el total de los cilindros requeridos se mostrara el botón “Terminar”, presiónalo para terminar el escaneo y registrar la distribución. (Fig.34)



Figura 34. Registro Distribución Externa

5.6.2. PRECIOS

En este apartado se muestra los precios por metro cúbico de los cilindros de oxígeno de acuerdo a las capacidades con las que se cuenta, donde están separados por los que están vigentes y los que ya han caducado. (Fig.35)



#	CAPACIDAD EQUIPO	VALOR POR METRO CUBICO	CODIGO POR EQUIPO	FECHA DEL CARGO	ESTADO DEL PRECIO
1	100L	10.00	10.00	2024-03-20 11:00:07	Actualizar
2	100L	10.00	10.00	2024-03-20 11:00:08	Actualizar
3	100L	10.00	10.00	2024-03-20 11:00:10	Actualizar
4	100L	10.00	10.00	2024-03-20 11:00:11	Actualizar
5	100L	10.00	10.00	2024-03-20 11:00:12	Actualizar

Figura 35. Página de Precios de Cilindros

5.6.2.1. ACTUALIZAR PRECIOS DE CILINDROS

Si desea actualizar o modificar el valor del precio, debe presionar el botón “Actualizar” y le abrirá una ventana emergente, donde debe seleccionar la capacidad y editar el precio del mismo, una vez realizado presionar el botón “Actualizar”, caso contrario presione “Cancelar”. (Fig.36)

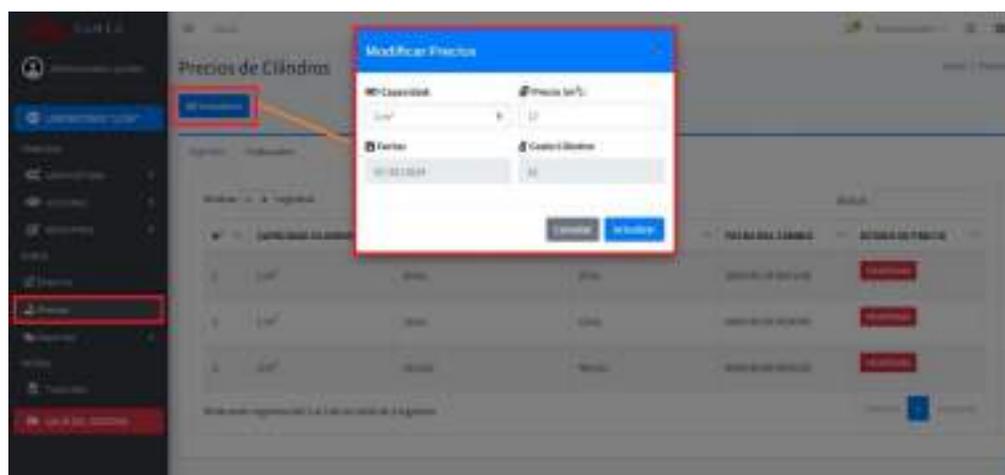


Figura 36. Actualiza precio de capacidades

5.6.3. REPORTE

5.6.3.1. GLOBALES

En este apartado se generan reportes de todos los cilindros que fueron recargados y distribuidos, dando la opción de filtrar por centro y por capacidad del cilindro, en un intervalo de fechas requeridas. (Fig.37).

6. ROL RESPONSABLE

6.1. MENÚ LATERAL IZQUIERDO

En este apartado muestra el nombre del Sistema, el nombre del usuario que inicio sesión y el nombre del centro a cargo de cada responsable (Fig.40).



Figura 40. Menú del Encargado

6.2. PÁGINA PRINCIPAL

En la parte central de la página se muestra los accesos directos a cada acción de pedidos y distribuciones, junto a datos estadísticos de los registros realizados del centro. (Fig.41).



Figura 41. Panel Principal del Encargado

6.3. PANEL ACCIONES

6.3.1. MÓDULO PEDIDOS

En este módulo se muestra la información de todos los pedidos que realiza el centro a través del encargado asignado, resaltando el estado en que se encuentran los mismos. (Fig.42)



Figura 42. Página de Pedidos

6.3.1.1. INFORMACIÓN DE UN PEDIDO

Si desea ver el detalle en específico de un pedido, debe presionar el botón “Revisar” en el campo de DETALLE PEDIDO y le abrirá una ventana emergente, donde muestra la capacidad y cantidad de cilindros requeridos por parte del centro. (Fig.43)

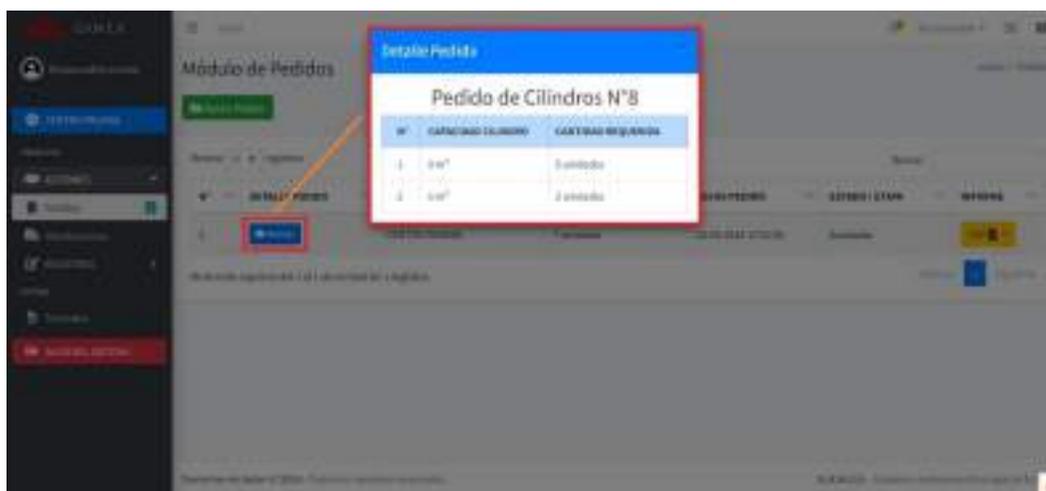


Figura 43. Detalle de un Pedido

6.3.1.2. IMPRIMIR INFORME DE UN PEDIDO

Si desea imprimir el informe físico de un pedido como comprobante, debe presionar el botón despegable “PDF” en el campo INFORME de la tabla, donde debe elegir si desea membretado o no, una vez elegido se abrirá el documento para poder descargarlo e imprimirlo posteriormente. (Fig.42)

6.3.1.3. AGREGAR NUEVO PEDIDO

Para crear un nuevo pedido deberá hacer click en el botón “Nuevo Pedido”, le enviará a una página para describir el detalle, presionando el botón “Agregar” abrirá una ventana emergente donde deberá seleccionar la capacidad de cilindro y la cantidad que requiera, si desea registrarlo presione el botón “Guardar”, caso contrario presione “Cancelar”. (Fig.44)

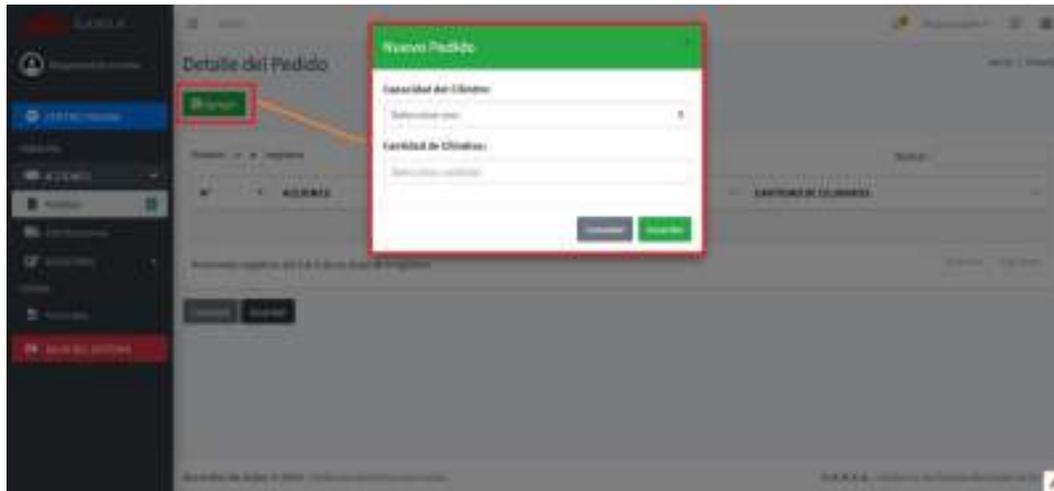


Figura 44. Ventana de registro del Pedido

IMPORTANTE:

Para registrar el pedido presione el botón “Guardar” ubicado debajo de la tabla que contiene sus pedidos específicos, caso contrario presione “Cancelar”, como se muestra en la siguiente imagen. (Fig.45)



Figura 45. Página detalle de un Pedido

6.3.1.4. EDITAR REGISTRO DE UN PEDIDO

Si desea corregir algún dato o modificarlo, debe presionar el botón despegable “Editar” en el campo de ACCIONES y le abrirá una ventana emergente, debe sobrescribir los datos ya registrados anteriormente y presionar el botón “Actualizar”, caso contrario presione “Cancelar”. (Fig.45)

6.3.1.5. BORRAR REGISTRO DE UN PEDIDO

En existe alguna equivocación al momento de solicitar y necesita eliminar un pedido específico, debe presionar el botón despegable “Borrar” en el campo de ACCIONES y le abrirá una ventana emergente, presione “Si” para confirmar la acción, caso contrario presione “No”. (Fig.45)

6.3.2. MODULO DISTRIBUCIONES

En este módulo se muestra la información de todas las distribuciones que se realizan al centro a cargo, así también las distribuciones que requieren ser confirmadas. (Fig.46)



Figura 46. Página de Distribuciones

IMPORTANTE:

Para confirmar la distribución de un pedido ir a la pestaña de PENDIENTES, marcar el botón “Confirmar E/S” en el campo de ESTADO / ETAPA de la tabla, una vez allá recibido o devuelto los cilindros de acuerdo a cada distribución; esto para tener la certeza de que no se halla presentado algún problema en el proceso de logística. (Fig.47)



Figura 47. Confirmar una Distribución

6.3.2.1. INFORMACIÓN DE UNA DISTRIBUCIÓN

Si desea ver el detalle en específico de una distribución, debe presionar el botón “Revisar” en el campo de DETALLE DISTRIBUCIÓN y le abrirá una ventana emergente, donde muestra la serie de los cilindros de oxígeno que se enviaron al centro. (Fig.48)



Figura 48. Detalle de una Distribución

6.3.2.2. IMPRIMIR INFORME DE UN PEDIDO

Si desea imprimir el informe físico de una distribución como comprobante, debe presionar el botón despegable “PDF” en el campo IMPRIMIR INFORME de la tabla, donde debe elegir si desea membretado o no, una vez elegido se abrirá el documento para poder descargarlo e imprimirlo posteriormente. (Fig.46)

6.4. PANEL REGISTROS

6.4.1. MÓDULO ALMACÉN

En este módulo se muestra el inventario del almacén con la información acerca de los cilindros de oxígeno que se encuentran el centro (Fig.49).

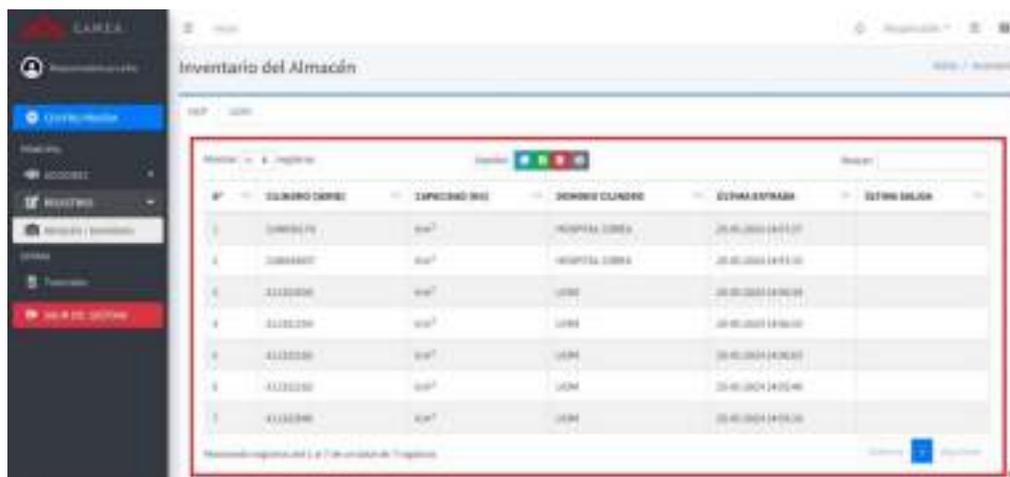


Figura 49. Almacén de un Centro

Así también, en la segunda pestaña se muestra información de los cilindros disponibles en el laboratorio, con respecto a su capacidad a través de una tabla, esto para futuros pedidos tomando en cuenta su disponibilidad. (Fig.29)

7. PANEL EXTRAS

7.1. TUTORIALES

Para una mejor comprensión del Sistema se desarrolló Tutoriales, para que todo aquel usuario nuevo o antiguo pueda entender y aprender con mayor facilidad el uso sistema. (Estos videos se encuentran en todos los roles y estos pueden variar dependiendo del rol). (Fig.50)



Figura 50. Tutoriales y Manual del Sistema

IMPORTANTE:

Para poder visualizar el presente Manual de Usuario, solo debe hacer click en “Descargar Manual”, el archivo se comenzará a descargar automáticamente en formato PDF.