

UNIVERSIDAD  
CATÓLICA  
DE CUENCA

**UNIVERSIDAD CATÓLICA DE CUENCA**

*Comunidad Educativa al Servicio del Pueblo*

**UNIDAD ACADÉMICA DE INFORMÁTICA,  
CIENCIAS DE LA COMPUTACIÓN E INNOVACIÓN  
TECNOLOGICA.**

**CARRERA DE INGENIERÍA DE SISTEMAS**

**SISTEMA DE GESTIÓN DE INVENTARIOS, MOVIMIENTOS DE  
MATERIAL BÉLICO Y DE INTENDENCIA A NIVEL NACIONAL  
PARA LA DIRMOV.**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL  
TÍTULO DE INGENIERO EN SISTEMAS.**

**AUTOR: CRISTIAN ISRAEL SIGUENCIA SALAZAR.**

**DIRECTOR: ING. JOSÉ CARRILLO ZENTENO.**

**CAÑAR- ECUADOR**

**2022**

**DIOS, PATRIA, CULTURA Y DESARROLLO**



# **UNIVERSIDAD CATÓLICA DE CUENCA**

*Comunidad Educativa al Servicio del Pueblo*

## **UNIDAD ACADÉMICA DE INFORMÁTICA, CIENCIAS DE LA COMPUTACIÓN E INNOVACIÓN TECNOLOGICA.**

### **CARRERA DE INGENIERÍA DE SISTEMAS**

#### **SISTEMA DE GESTIÓN DE INVENTARIOS, MOVIMIENTOS DE MATERIAL BÉLICO Y DE INTENDENCIA A NIVEL NACIONAL PARA LA DIRMOV.**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL  
TÍTULO DE INGENIERO EN SISTEMAS.**

**AUTOR:** CRISTIAN ISRAEL SIGUENCIA SALAZAR.

**DIRECTOR:** ING. JOSÉ CARRILLO ZENTENO.

**CAÑAR- ECUADOR**

**2022**

**DIOS, PATRIA, CULTURA Y DESARROLLO**



## **DEDICATORIA**

Dedico con todo mi corazón mi trabajo de titulación a mi madre quien siempre me apoyo para poder llegar a esta instancia de mis estudios, siendo así una parte fundamental sin la cual no lo habría logrado. También la dedico a mi tía quien ha sido mi mayor motivación para nunca rendirme ante cualquier adversidad y poder llegar a ser una persona de bien.

## **AGRADECIMIENTO:**

Primeramente, agradezco a Dios por permitirme tener tan buena experiencia dentro de la Universidad, a mi familia por apoyarme en cada decisión y proyecto.

A esta prestigiosa casa de estudios por haberme aceptado ser parte de ella abriéndome las puertas de su seno científico para poder estudiar mi carrera.

A todos los docentes que me brindaron sus conocimientos y su apoyo para seguir adelante día a día.

Y para finalizar, también agradezco a mi tutor Ing. José Carrillo por haberme brindado la oportunidad de recurrir a su capacidad y conocimiento científico, así como también haberme tenido toda la paciencia del mundo para guiarme durante todo el desarrollo del proyecto tecnológico.



## CERTIFICACIÓN

Certificó que el presente trabajo fue desarrollado por el Est. Cristian Israel Siguenca Salazar,  
bajo mi supervisión.

Ing. José Antonio Carrillo Zenteno

DIRECTOR DEL TRABAJO INVESTIGATIVO

UNIVERSIDAD CATÓLICA DE CUENCA.



## DECLARACION

Yo, Cristian Israel Siguencia Salazar, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y que he consultado las referencias bibliográficas que se incluyen en este documento.

La Universidad Católica de Cuenca extensión Cañar puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y la Normativa actual de la institución.

Sr. Cristian Israel Siguencia Salazar

Alumno

UNIVERSIDAD CATÓLICA DE CUENCA.



## **APROBACIÓN DE TRIBUNAL DE GRADO**

El tribunal designado por el honorable consejo directivo de la Universidad Católica de Cuenca Extensión Cañar, Facultad de Ingeniería de Sistemas instalado para receptor la sustentación del trabajo final de investigación con el tema: “PROPUESTA DE DISEÑO DE UN SISTEMA DE GESTIÓN DE INVENTARIOS, MOVIMIENTOS DE MATERIAL BÉLICO Y DE INTENDENCIA A NIVEL NACIONAL PARA LA DIRMOV”, transcurrido el tiempo reglamentario procede a consignar la calificación de (\_\_\_\_\_/100).

Cañar, \_\_\_\_\_ de \_\_\_\_\_ del 2022

---

**PRESIDENTE**

---

**DIRECTOR**

---

**DELEGADO**

---

**SECRETARIO**

## Resumen

El presente proyecto tecnológico tiene como finalidad brindar una solución integral a la falta de sistematización en los procesos de captura de información de ingreso, inventarios y movimientos de material bélico y de intendencia dentro de las bodegas de la DIRMOV a nivel nacional, se detalla el estudio sobre operaciones de inventario y análisis de requerimientos realizado a la organización, correspondiente a como se llevan los procedimientos de bodegaje, movimientos y manejo de materiales, generando como resultado final un sistema de gestión de inventarios, basado en una aplicación de escritorio Java generada en Netbeans, la disposición de consumo de datos en la web mediante un API RESTful creado mediante el API Java JAX-RS, el almacenamiento de datos en SQL Server 2019, y la metodología de desarrollo de software en espiral. Implementando características funcionales específicas permitiendo la entrada de datos de forma ágil y segura, sistematizando los procesos de las bodegas al momento de generar movimientos conjuntamente con el registro de la documentación generada para cada proceso.

**Palabras Clave:** sistema, inventario, sql server, fuerzas armadas, java, restful.

### **Abstract**

This technological research aims to provide an integral solution to the lack of systematization in data capture processes for the input, stock and positioning of warfare material and supplies inside the DIRMOV warehouses at national level; the study on stock operations and analysis of requirements made to the organization is detailed. It shows how warehousing, movements and handling of materials are carried out. As a result, an inventory management system is created which is based on a Java desktop application using Netbeans, data usage availability on the web using a RESTful API created through the Java JAX-RS API, data storage in SQL Server 2019; the spiral software development methodology is also carried out, implementing specific functional features that allow data entry in an agile and safe way; systematizing warehouses processes when moving things from one place to another, together with the registration of the documentation made for each process.

**Keywords:** system, inventory, sql server, armed forces, java, restful.



## Contenido

DEDICATORIA.....	I
AGRADECIMIENTO:.....	II
APROBACIÓN DE TRIBUNAL DE GRADO .....	V
Resumen .....	VI
Abstract .....	VII
Tabla de Ilustraciones.....	9
CAPITULO I.....	11
Introducción.....	11
Marco conceptual .....	13
Marco teórico.....	17
Metodología.....	20
Metodología de revisión teórica .....	20
Metodología de desarrollo de software .....	27
Resultados.....	34
Conclusiones.....	36
CAPITULO II.....	38
Manual de usuario .....	38
Aplicación de escritorio.....	38
Ingreso al Sistema.....	39
Pantalla inicio.....	40
Pantalla material bélico administración de privilegios .....	42
Pantalla administración de material bélico .....	43
Pantalla solicitud de actualización material bélico .....	45
Pantalla historial solicitud actualización material bélico.....	45
Pantalla actualizar registro de material bélico.....	47
Pantalla mantenimiento material bélico.....	48
Pantalla administración intendencia.....	49
Pantalla administración usuarios.....	51
Pantalla consulta arma.....	52
Pantalla consulta intendencia.....	53
Pantalla monitoreo movimientos arma e intendencia .....	54
Pantalla movimiento egreso material bélico e intendencia.....	56
Pantalla movimiento ingreso material bélico e intendencia. ....	59
Pantalla aprobación egreso material bélico e intendencia. ....	60



<i>Pantalla aprobación préstamo devolución egreso material bélico e intendencia. ....</i>	<i>61</i>
<i>Pantallas tablas auxiliares. ....</i>	<i>63</i>
<i>Capitulo III.....</i>	<i>67</i>
<i>Manual del programador.....</i>	<i>67</i>
<i>Programación.....</i>	<i>67</i>
<i>Análisis de requerimientos.....</i>	<i>68</i>
<i>Base de datos .....</i>	<i>69</i>
<i>Tablas.....</i>	<i>72</i>
<i>Desarrollo API_RESTfull .....</i>	<i>81</i>
<i>Aplicación de escritorio.....</i>	<i>85</i>
<i>Librerías y superclases capa de presentación.....</i>	<i>88</i>
<i>Código capa de presentación.....</i>	<i>95</i>
<i>Bibliografía .....</i>	<i>108</i>
<i>ANEXOS.....</i>	<i>110</i>
<i>Anexo 1: Protocolo de Tesis .....</i>	<i>110</i>
<i>Anexo 2: Certificado para subir a repositorio institucional .....</i>	<i>123</i>



## Tabla de Ilustraciones

<i>Ilustración 1. Resultados formula búsqueda sin filtro de fecha SciELO. Fuente: Autor</i>	22
<i>Ilustración 2. Resultados formula búsqueda con filtro de fecha SciELO. Fuente: Autor</i>	23
<i>Ilustración 3. Resultados formula búsqueda sin filtro de fecha Google Académico. Fuente: Autor</i>	23
<i>Ilustración 4. Resultados formula búsqueda con filtro de fecha Google Académico. Fuente: Autor</i>	24
<i>Ilustración 5. Resultados formula búsqueda sin filtro de fecha Biblioteca UCACUE. Fuente: Autor</i>	24
<i>Ilustración 6. Resultados formula búsqueda con filtro de fecha Biblioteca UCACUE. Fuente: Autor</i>	25
<i>Ilustración 7. Tabla cronograma de actividades metodología. Fuente: Autor</i>	33
<i>Ilustración 8. Ventana emergente comprobación sistema. Fuente: Autor</i>	39
<i>Ilustración 9. Ventana inicio sesión sistema. Fuente: Autor</i>	39
<i>Ilustración 10. Ventana inicio sistema. Fuente: Autor</i>	40
<i>Ilustración 11. Ventana normativa legal sistema. Fuente: Autor</i>	41
<i>Ilustración 12. Ventana contactos sistema. Fuente: Autor</i>	41
<i>Ilustración 13. Ventana centros sistema. Fuente: Autor</i>	42
<i>Ilustración 14. Menú administración arma sistema. Fuente: Autor</i>	42
<i>Ilustración 15. Ventana base centro sistema. Fuente: Autor</i>	43
<i>Ilustración 16. Ventana administración material bélico sistema. Fuente: Autor</i>	44
<i>Ilustración 17. Ventana solicitud actualización sistema. Fuente: Autor</i>	45
<i>Ilustración 18. Menú historial solicitud sistema. Fuente: Autor</i>	46
<i>Ilustración 19. Ventana historial solicitud actualización sistema. Fuente: Autor</i>	46
<i>Ilustración 20. Ventana actualizar sistema. Fuente: Autor</i>	47
<i>Ilustración 21. Menú mantenimiento material bélico sistema. Fuente: Autor</i>	48
<i>Ilustración 22. Ventana mantenimiento arma sistema. Fuente: Autor</i>	49
<i>Ilustración 23. Ventana mantenimiento sistema. Fuente: Autor</i>	49
<i>Ilustración 24. Menú intendencia sistema. Fuente: Autor</i>	50
<i>Ilustración 25. Ventana mantenimiento intendencia sistema. Fuente: Autor</i>	50
<i>Ilustración 26. Menú administración usuario sistema. Fuente: Autor</i>	51
<i>Ilustración 27. Ventana mantenimiento usuario sistema. Fuente: Autor</i>	51
<i>Ilustración 28. Menú consultas material bélico sistema. Fuente: Autor</i>	52
<i>Ilustración 29. Ventana consultas material bélico sistema. Fuente: Autor</i>	53
<i>Ilustración 30. Menú consulta intendencia sistema. Fuente: Autor</i>	53
<i>Ilustración 31. Ventana consulta intendencia sistema. Fuente: Autor</i>	54
<i>Ilustración 32. Menú monitoreo material bélico sistema. Fuente: Autor</i>	55
<i>Ilustración 33. Ventana monitoreo material bélico sistema. Fuente: Autor</i>	56
<i>Ilustración 34. Menú movimiento egreso material bélico sistema. Fuente: Autor</i>	56
<i>Ilustración 35. Ventana movimiento egreso material bélico sistema. Fuente: Autor</i>	58
<i>Ilustración 36. Ventana acta material bélico sistema. Fuente: Autor</i>	59
<i>Ilustración 37. Menú movimiento ingreso material bélico sistema. Fuente: Autor</i>	59
<i>Ilustración 38. Ventana movimiento ingreso material bélico sistema. Fuente: Autor</i>	60
<i>Ilustración 39. Menú aprobación egreso definitivo material bélico sistema. Fuente: Autor</i>	60
<i>Ilustración 40. Ventana aprobación egreso definitivo material bélico sistema. Fuente: Autor</i>	61
<i>Ilustración 41. Menú aprobación egreso definitivo material bélico sistema. Fuente: Autor</i>	61
<i>Ilustración 42. Ventana aprobación acta egreso definitivo material bélico sistema. Fuente: Autor</i>	62
<i>Ilustración 43. Ventana aprobación acta egreso movimiento material bélico sistema. Fuente: Autor</i>	62
<i>Ilustración 44. Ventana aprobación firma egreso material bélico sistema. Fuente: Autor</i>	63
<i>Ilustración 45. Ventana aprobar firma movimiento ingreso material bélico sistema. Fuente: Autor</i>	63
<i>Ilustración 46. Menú campos material bélico sistema. Fuente: Autor</i>	64
<i>Ilustración 47. Menú campos material intendencia sistema. Fuente: Autor</i>	64
<i>Ilustración 48. Menú campos material usuario sistema. Fuente: Autor</i>	65
<i>Ilustración 49. Ventana tabla auxiliar genérica sistema. Fuente: Autor</i>	66
<i>Ilustración 50. Modelo físico entidad relación base de datos. Fuente: Autor.</i>	71
<i>Ilustración 51. Tabla detalle mantenimiento base de datos. Fuente: Autor.</i>	72
<i>Ilustración 52. Tabla mantenimiento base de datos. Fuente: Autor.</i>	72
<i>Ilustración 53. Tabla detalle mantenimiento usuario base de datos. Fuente: Autor.</i>	73
<i>Ilustración 54. Tabla acta intendencia base de datos. Fuente: Autor.</i>	73



<i>Ilustración 55. Tabla arma base de datos. Fuente: Autor.</i>	74
<i>Ilustración 56. Tabla intendencia base de datos. Fuente: Autor.</i>	75
<i>Ilustración 57. Tabla detalle movimiento arma base de datos. Fuente: Autor.</i>	75
<i>Ilustración 58. Tabla movimiento intendencia base de datos. Fuente: Autor.</i>	76
<i>Ilustración 59. Tabla detalle historial intendencia base de datos. Fuente: Autor.</i>	76
<i>Ilustración 60. Tabla historial intendencia base de datos. Fuente: Autor.</i>	77
<i>Ilustración 61. Tabla detalle movimiento intendencia base de datos. Fuente: Autor.</i>	77
<i>Ilustración 62. Tabla acta base de datos. Fuente: Autor.</i>	77
<i>Ilustración 63. Tabla usuario base de datos. Fuente: Autor.</i>	78
<i>Ilustración 64. Tabla detalle movimiento usuario base de datos. Fuente: Autor.</i>	78
<i>Ilustración 65. Tabla movimiento material bélico base de datos. Fuente: Autor.</i>	79
<i>Ilustración 66. Tabla historial base de datos. Fuente: Autor.</i>	79
<i>Ilustración 67. Tabla detalle historial material bélico base de datos. Fuente: Autor.</i>	80
<i>Ilustración 68. Tabla tipo usuario base de datos. Fuente: Autor.</i>	80
<i>Ilustración 69. Conexión base de datos Netbeans. Fuente: Autor.</i>	81
<i>Ilustración 70. Conector jdbc sql server Netbeans. Fuente: Autor.</i>	81
<i>Ilustración 71. Datos de conexión sql server Netbeans. Fuente: Autor.</i>	82
<i>Ilustración 72. Aplicación web Netbeans. Fuente: Autor.</i>	82
<i>Ilustración 73. RESTful Web Services Netbeans. Fuente: Autor.</i>	83
<i>Ilustración 74. Tablas sql Client Netbeans. Fuente: Autor.</i>	83
<i>Ilustración 75. Archivo .war RESTful web service. Fuente: Autor.</i>	84
<i>Ilustración 76. Configuración Servidor Glassfish. Fuente: Autor.</i>	84
<i>Ilustración 77. Server Admin Glassfish. Fuente: Autor.</i>	85
<i>Ilustración 78. Aplicación JAVA Netbeans. Fuente: Autor.</i>	85
<i>Ilustración 79. Capa de datos Netbeans. Fuente: Autor.</i>	86
<i>Ilustración 80. RESTfull JAVA Client Netbeans. Fuente: Autor.</i>	86
<i>Ilustración 81. Capa de negocios Netbeans. Fuente: Autor.</i>	87
<i>Ilustración 82. Capa de presentación Netbeans. Fuente: Autor.</i>	88
<i>Ilustración 83. Librerías pantalla administración material bélico sistema. Fuente: Autor.</i>	93
<i>Ilustración 84. Librerías mantenimiento intendencia sistema. Fuente: Autor.</i>	94
<i>Ilustración 85. Librerías consulta material bélico sistema. Fuente: Autor.</i>	94
<i>Ilustración 86. Código pantalla inicio sistema. Fuente: Autor.</i>	95
<i>Ilustración 87. Código pantalla administración material bélico sistema. Fuente: Autor.</i>	96
<i>Ilustración 88. Código mantenimiento material bélico sistema. Fuente: Autor.</i>	97
<i>Ilustración 89. Código mantenimiento material bélico parte dos sistemas. Fuente: Autor.</i>	98
<i>Ilustración 90. Variables solicitud actualización material bélico sistema. Fuente: Autor.</i>	98
<i>Ilustración 91. Código mantenimiento intendencia sistema. Fuente: Autor.</i>	99
<i>Ilustración 92. Código mantenimiento usuario sistema. Fuente: Autor.</i>	100
<i>Ilustración 93. Código consulta material bélico sistema. Fuente: Autor.</i>	101
<i>Ilustración 94. Código consulta material bélico sistema. Fuente: Autor.</i>	102
<i>Ilustración 95. Código monitoreo material bélico sistema. Fuente: Autor.</i>	103
<i>Ilustración 96. Código monitoreo material intendencia sistema. Fuente: Autor.</i>	104
<i>Ilustración 97. Código movimiento material bélico sistema. Fuente: Autor.</i>	104
<i>Ilustración 98. Código movimiento material bélico parte dos sistemas. Fuente: Autor.</i>	105
<i>Ilustración 99. Código pantalla prestamos activos material bélico sistema. Fuente: Autor.</i>	106
<i>Ilustración 100. Librerías y código pantallas tablas auxiliares genéricas sistema. Fuente: Autor.</i>	107
<i>1 Cuadro comparativo Modelos de desarrollo de software (José Alejandro Ferto Martínez)</i>	116
<i>2 Metodología de desarrollo en espiral -Autoría Propia</i>	118

## CAPITULO I

### Introducción

Son incontables los aportes que ofrecen los avances tecnológicos en todo tipo de contexto laboral y dentro de procesos de una organización. En Ecuador incontables instituciones han adicionado el proceso de automatización, obteniendo mayor eficiencia en sus procesos debido al rápido avance en los campos de la informática, exigiendo así a la DIRMOV<sup>1</sup> implementar métodos más sofisticados que faciliten la ejecución del trabajo con mayor eficiencia y calidad.

Las fuerzas armadas, siendo un pilar fundamental de protección y organización nacional, no estando exento del avance tecnológico al que se someten todas las organizaciones del país, por tales motivos se han planteado el perfeccionamiento tecnológico como uno de sus objetivos vitales.

La DIRMOV controla y registra material bélico y de intendencia, así como genera movimientos de dicho material a las distintas bases-centros del Ecuador. Actualmente estos procesos no se ejecutan de forma eficiente, por tal motivo es indispensable un sistema de gestión de inventario de material bélico y de intendencia que guarde un enfoque general, donde cada módulo que conforma un proceso tenga una función específica bien delimitada, brindando una mejor asistencia a los encargados, mejorando la organización, disposición, trayectoria, control y valoración a las actividades de trabajo que se desarrollan en la organización.

Con el desarrollo de este sistema se da solución a los inconvenientes con respecto a los procesos de registro de inventarios de material bélico y de intendencia, el exceso de órdenes y solicitudes en proceso, las entregas tardías de material, acumulación de trabajo, que genera pérdida de tiempo y recursos a la organización, mediante la implementación de funcionalidades específicas

---

<sup>1</sup> Dirección de Movilización del Comando Conjunto de las Fuerzas Armadas

para el registro de usuarios con su correspondiente localización, acceso únicamente a los registros y funcionalidades disponibles para una base-centro determinada, registro del material bélico y de intendencia, registro de movimientos entre los diferentes centros del país, filtros de búsqueda exportación de reportes, aprobación de solicitudes y mantenimiento.

Para el desarrollo del sistema se realizaron los siguientes pasos: obtención de requerimientos, desarrollo de una base de datos, creación de un API\_RESTfull<sup>2</sup>, desarrollo de una aplicación de escritorio, y ejecución de pruebas.

---

<sup>2</sup> Representational State Transfer- Transferencia de Estado Representacional

## Marco conceptual

**Java.** Es un lenguaje de programación orientado a objetos y multiplataforma, los programas se desarrollan una sola vez y se ejecutaron en diversos sistemas operativos, para lograr esto Java se maneja en un ambiente de cumplimiento para los programas agrupados (JRE<sup>3</sup>). (Guevara, 2016, pág. 8).

**Inventario.** Es un registro de los bienes, activos u objetos tangibles de una organización, que se incluyen dentro de los procesos, y sirven de soporte a las actividades diarias. (Durán, 2012, pág. 55).

**Material bélico.** Es todo elemento que será registrado en el sistema de inventario que sea perteneciente a la categoría de material de guerra (Real Academia Española, 2021).

**Material de intendencia.** Son todos aquellos elementos destinados al suministro de las fuerzas armadas y que sean distribuidos a los campamentos, edificios o bases militares. (Real Academia Española).

**Automatizar.** Consiste en utilizar la tecnología para realizar partes de un proceso en los que exista una mínima o nula intervención humana, se aplica a procesos repetitivos mejorando la rapidez con la que se desarrollan y el control de calidad frente a trabajos muy rápidos y complejos . (UA, 2017, pág. 5)

**REST.** Según la tesis doctoral de Roy Fielding, publicada en el año 2000, el término REST (Representational State Transfer) se traduce como transferencia de estado representacional, es una definición de arquitectura de software en la cual se indica cómo efectuar el intercambio y administración de datos a través de servicios web, contiene un conjunto de normas de diseño que debe de cumplir un sistema.

---

<sup>3</sup> Java Runtime Environment, también conocido como, conjunto de utilidades de empaquetado

El presente diseño arquitectónico posee las siguientes características:

- La arquitectura es cliente servidor.
- Todas las peticiones al servicio no requieren inicio de sesión por parte de los usuarios, y son independientes de las demás.
- Los requerimientos de infraestructura de la red necesariamente deben de implementar un sistema de cachés.
- El sistema es completamente accesible
- Implementa una interfaz uniforme, los recursos están alineados con la dirección única y un punto valido para el acceso.
- La identificación de recursos se realiza mediante URIs, proporcionando un mejor espacio para el direccionamiento general.
- El sistema esta generado mediante capas con la finalidad de implementar escalabilidad.
- Los mensajes son auto descriptivos, para que los recursos se desprendan de su representación y se puedan obtener en una diversidad de formatos, como por ejemplo JSON, XML, texto plano, etc. (Computación, Dept. Ciencia de la, 2013, pág. 2).

**API RESTful.** Es una interfaz de programación de aplicaciones que se desarrolló en base a la arquitectura REST. (Computación, Dept. Ciencia de la, 2013)

**JSON.** Es un formato de texto plano utilizado en el intercambio de datos, para su manipulación se utiliza normalmente el lenguaje JavaScript. La estructura gramatical es bastante simple necesitando únicamente agrupar datos y valores. En primer lugar, los elementos están encerrados entre de llaves { datos }, en cuanto a los diferentes elementos sus valores se asignan en pares, con la estructura: nombre : valor, y se separan

mediante comas, por último, una secuencia de elementos están contenidos dentro de corchetes [ y ]. (Computación, Dept. Ciencia de la, 2013).

**JAX-RS.** Es un API JAVA para servicios RESTful. Dicho API es utilizado por cualquiera que desee implementar un servicio web de acuerdo con el estilo arquitectónico REST. Proporcionando anotaciones correspondientes a cada petición de recursos necesario como: Path, Get, Put, Delete, Head.

**Jersey.** Brinda los conectores para los servicios web mediante anotaciones Java. estas anotaciones contienen el código requerido por las diferentes clases para conectarse sin dificultades con determinados Frameworks o librerías determinadas. (Dept. Ciencia de la Computación, 2013, pág. 5).

**Base de datos.** Es un conjunto bien estructurado de datos pertenecientes a entidades relacionales dentro de un mismo contexto, que pueden ser administrados y analizados proporcionando información sobre los procesos o transacciones de una organización. (Paré, 2015, pág. 18).

**URI.** (Uniform Resource Identifier) traducido como identificador uniforme de recursos, corresponde a un hiper-enlace a un recurso, permitiendo de forma única el intercambio de representaciones entre cliente servidor. Para la identificación de determinados objetivos de una interacción con clientes el servicio web RESTful proporciona un conjunto de recursos propios automáticamente generados. REST indica los hiper-enlaces que serán utilizados en la web para establecer los servicios. (Dept. Ciencia de la Computación, 2013, pág. 4).

**Peticiones HTTP.** Son un conjunto de métodos para indicar la acción que se desea ejecutar sobre un determinado recurso. Cada uno de estas peticiones implementa una semántica diferente, pero comparten algunas características similares. Dentro de este



contexto, las peticiones HTTP permiten la administración de los recursos como crear, recuperar, modificar y borrar. (Dept. Ciencia de la Computación, 2013, pág. 7).

## Marco teórico

En el trabajo “sistematización del inventario para el almacén de pinturas y ferretería Ferrecolor” se expone la utilización de herramientas informáticas con el beneficio de ejercer mayor control de los productos y servicios para la comercialización. La persona encargada realizaba su inventario por medio de hojas de cálculo lo que no podía tener la información clara, estable y precisa acerca de los productos para ofrecer a los clientes. Por medio del análisis de requerimientos se gestionó el inventario a través de la aplicación de buenas prácticas con la aplicación estándar de JAVA EE, el framework JSF<sup>4</sup> y Mysql como el principal motor de la base de datos. Mediante esta solución se almacena para poder usar el administrador, para realizar sus actividades tales como datos de los inventarios, así como también se enfoca en los recursos de manera óptima para el crecimiento empresarial. (Romero, 2019)

El trabajo analizado guarda relación con el proyecto aquí desarrollado y es de suma importancia para entender los procesos y métodos de recolección de requerimientos, para su posterior análisis y división en: requerimientos funcionales y no funcionales, donde se describe punto por punto cada nivel de gestión y acceso que proveerá el software y su interacción con el usuario final mediante la representación en casos de uso, diagramas de secuencia y de procesos.

Según el trabajo de desarrollo de software “boceto y desarrollo de un sistema de inventario e ingreso de información de computadoras y material tecnológico basado en java” cumple con el propósito de mantener la información sobre inventarios de computadores de una forma rápida, pertinente y sistemática. La metodología utilizada se basa en espiral, desarrollada con una arquitectura cliente-servidor, el lenguaje de programación Java, dentro de la creación de la parte gráfica de la interfaz se utilizaron

---

<sup>4</sup> framework para aplicaciones Java basadas en web

JFrames, menús y verificación de acceso, permitiendo la comunicación de forma intuitiva entre el usuario y el sistema al momento de crear registros, al finalizar el desarrollo del sistema se pudo cubrir varias funcionalidades planteadas desde el inicio como generar reportes, registro de datos, estado y mantenimiento, seguridad en cuanto a saber quién fue el encargado que realizó el registro o modificación, fechas de cada mantenimiento, encargado del computador. (Morocho, 2018)

Se realizó la implementación del sistema con 14 módulos todos estos congruentes y en base a los requerimientos del usuario para la creación de un inventario, en este trabajo se presentan aspectos necesarios para entender los procesos de registro considerando importante la sección de programación orientada a objetos lo que conforma el sistema para el inventario en tiempo real, el trabajo se desarrolla en NetBeans detallando el proceso de creación por medio de aspectos importantes de software conocidos como módulos.

Para el proyecto “Sistema de administración de control de inventario en almacén” se expone el proceso de desarrollo e implementación del sistema en una organización que realiza trabajos de facturación de materiales de acero, la característica esencial de este sistema se fundamenta en mantener el control de productos en stock.

En cuanto a la ejecución de este proyecto se utilizan varios fundamentos de aplicación y desarrollo tecnológico. Programación Java, es la rama primaria para este proyecto, estructurando la mayor cantidad de información para el manejo de base de datos. La metodología aplicada se basa en el modelo en espiral, ya que, basándose en las particularidades del proyecto se ejecuta un análisis de requerimientos en base a las necesidades del cliente, continuando posteriormente así con la realización de las demás etapas de este modelo. En cuanto a herramientas empleadas en la creación del sistema incluyen Netbeans, para la parte de programación, Sql Server 2016 para la creación de la base de datos, JasperReport para la generación de reportes. (JUÁREZ, 2019)

Este trabajo engloba la mayoría de herramientas utilizadas para el desarrollo del software, se detallan los procesos de tratamiento de la información de ingreso, entrega un formato de interfaz de administración de registros, creación de roles y privilegios con la finalidad de limitar determinadas acciones a los usuarios, se detalla la creación de una ventana de inicio de sesión, CRUD para cada entidad del sistema y aplicación de consultas.

En el trabajo “Generación automática de API REST a partir de Java, basada en transformación de Modelos (MDD<sup>5</sup>)” Se presenta la línea de ejecución en la que se plantea un conjunto de procesos para la creación de un API REST mediante la utilización de la herramienta que implementa Java “RESTful web services from database”, en el marco de desarrollo aplicado a modelos (Model-Driven Development, MDD), empleando este proceso metodológico como alternativa de desarrollo de modelos, diferenciándose de otras formas convencionales, las cuales se fundamentan en generar un AST (Abstract Syntax Tree) por medio de algún convertidor de JAVA. Además, este planteamiento proporciona código hacia distintas implementaciones de un API\_RESTful a partir de un modelo JAVA. (Ariel Arsaute, 2018)

Es de suma importancia conocer el proceso de implementación de un interfaz de comunicación entre los clientes y el servidor que manejara la aplicación de escritorio basado en Java, profundizando así en el alcance de esta tecnología utilizada actualmente por las principales redes digitales del mercado por su gran accesibilidad rapidez y eficiencia en la carga y subida de datos mediante una conexión en la web.

---

<sup>5</sup> Desarrollo Dirigido por Modelos

## Metodología

### Metodología de revisión teórica

Para la realización de esta parte del documento se llevó a cabo una investigación teórica sistemática de la literatura científica publicada en relación con el desarrollo de software.

Para la ejecución, se siguieron los lineamientos establecidos en la declaración PRISMA<sup>6</sup> generando una adecuada revisión sistemática. A continuación, se detalla cada punto del proceso de creación.

### Búsqueda inicial

En de la primera parte de búsqueda realizada se combinaron los términos ‘Sistema’ e ‘Inventario’ en las bases de datos SciELO, biblioteca virtual UCACUE y Google Académico. Consecutivamente, se extendió la búsqueda con una combinación, utilizando operadores AND y OR, de los términos: ‘API\_RESTful’, ‘SQL Server’, ‘Fuerzas armadas’, ‘Java’ y que se encuentren publicados desde el año 2018 en adelante.

Esta búsqueda genero una enorme cantidad de resultados, en su mayoría repetidos o nada importantes para la revisión, pero por otro lado se evidencio de forma global el alcance del tema.

---

<sup>6</sup> Preferred Reporting Items for Systematic reviews and Meta-Analyses, es un método de revisiones sistemáticas sobre elementos de informe preferidos para revisiones sistemáticas y metanálisis.

## Diagrama final de revisión sistemática

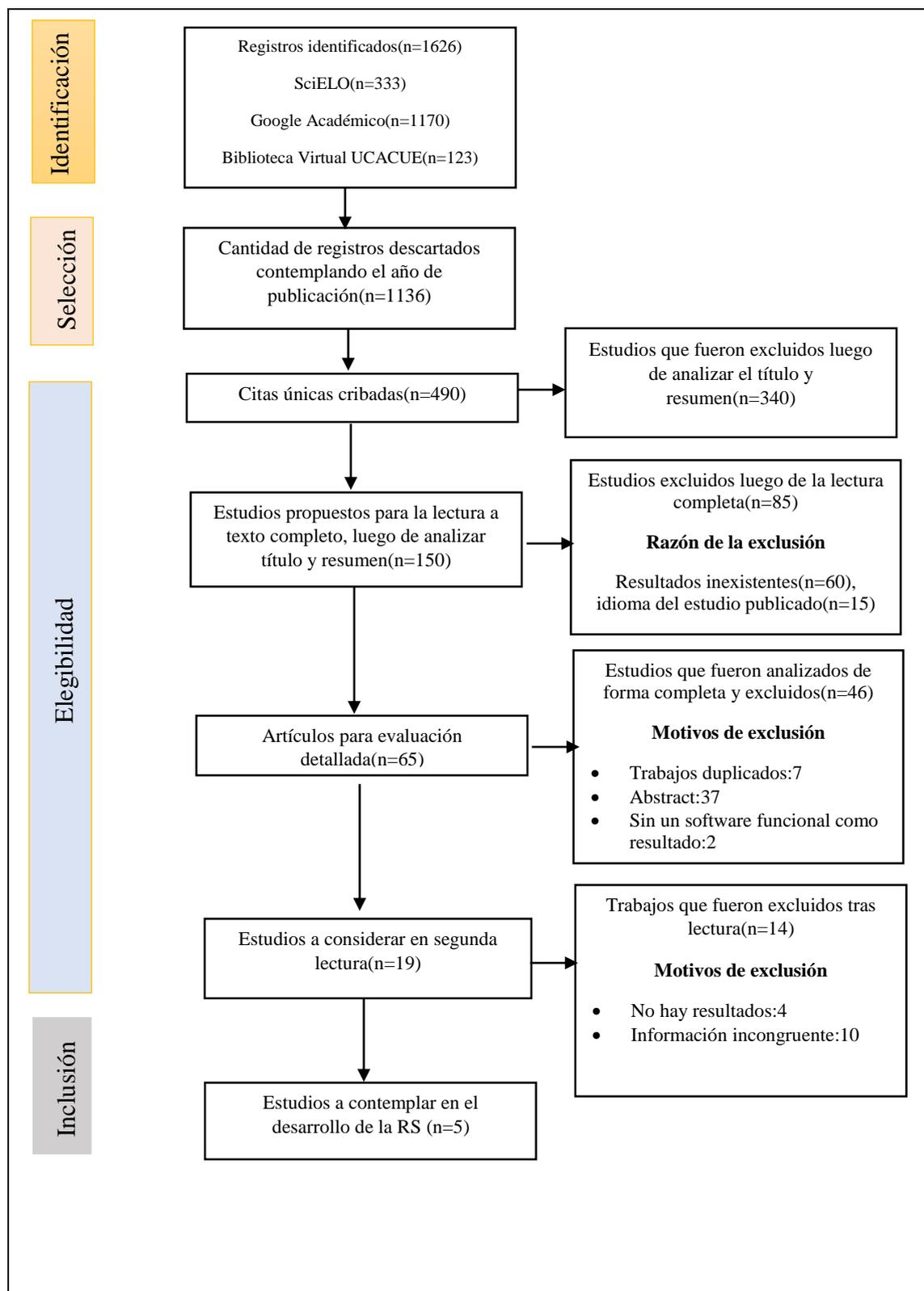
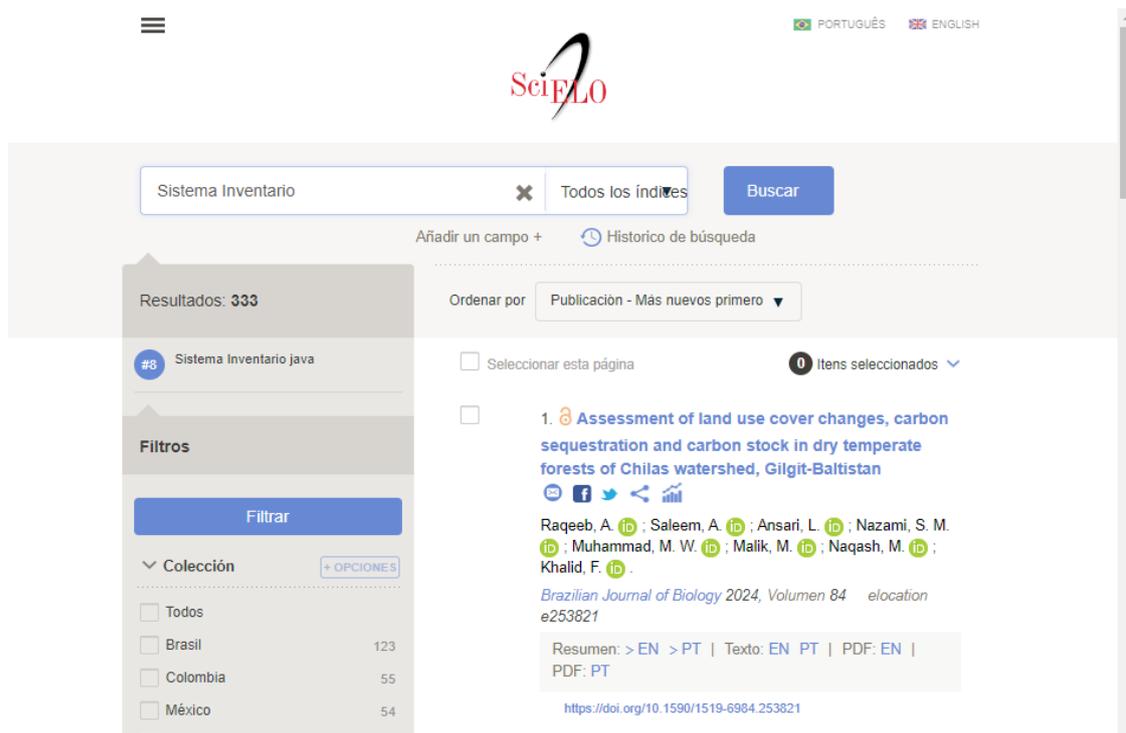


Diagrama- 1- resultados modelo PRISMA Fuente:Autor

Específicamente, se consiguieron 125 resultados en SciELO, 323 en Google Académico(n=323) y 42 en la Biblioteca Virtual UCACUE en primera instancia sin contar el año de publicación y segunda instancia con el filtro de fecha de 2018 en adelante.

Resultados de la búsqueda en la biblioteca electrónica SciELO con los términos “Sistema” e “Inventario”.



The screenshot shows the SciELO search interface. At the top, there is a search bar with the query 'Sistema Inventario' and a 'Buscar' button. Below the search bar, the results are displayed. On the left side, there is a sidebar with filters, including 'Colección' with options for 'Todos', 'Brasil' (123), 'Colombia' (55), and 'México' (54). The main content area shows a list of results, with the first result being 'Assessment of land use cover changes, carbon sequestration and carbon stock in dry temperate forests of Chilas watershed, Gilgit-Baltistan'. The authors listed are Raqeeb, A.; Saleem, A.; Ansari, L.; Nazami, S. M.; Muhammad, M. W.; Malik, M.; Naqash, M.; and Khalid, F. The journal is 'Brazilian Journal of Biology 2024, Volumen 84' with the eLocation 'e253821'. There are also links for 'Resumen' and 'PDF' in both English and Portuguese, and a DOI link: <https://doi.org/10.1590/1519-6984.253821>.

Ilustración 1. Resultados formula búsqueda sin filtro de fecha SciELO. Fuente: Autor

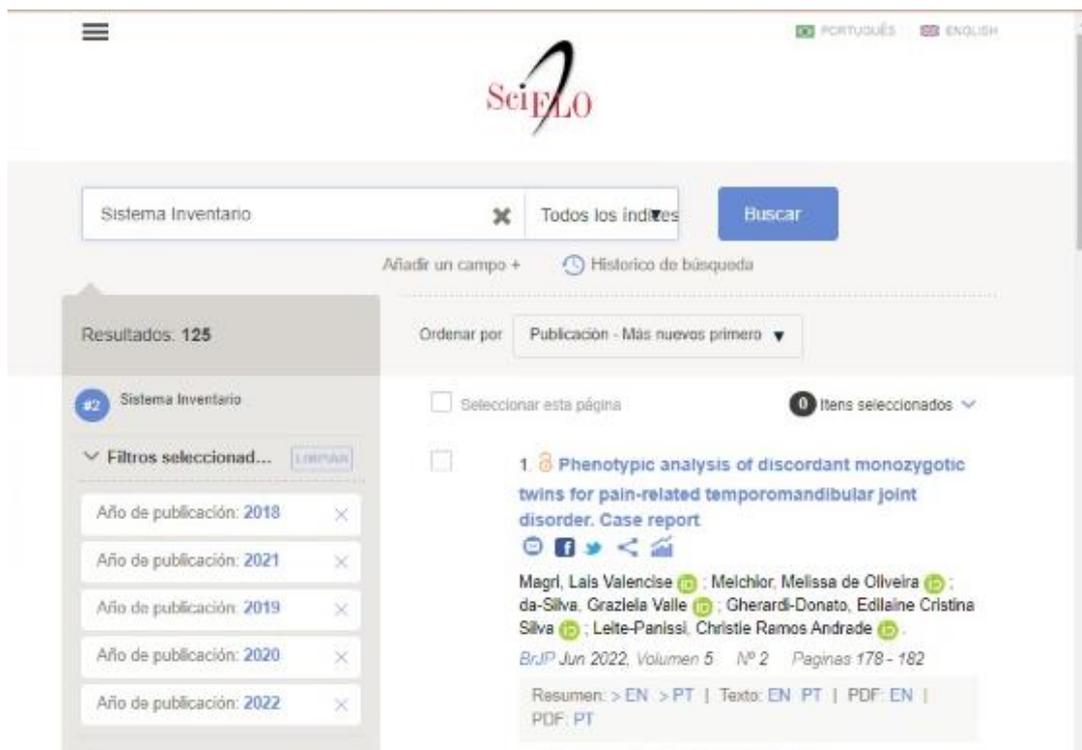


Ilustración 2. Resultados formula búsqueda con filtro de fecha SciELO. Fuente: Autor

Resultados de la búsqueda en la biblioteca electrónica Google Académico con los términos “Sistema”, “Inventario”, “fuerzas armadas” y “Java”.



Ilustración 3. Resultados formula búsqueda sin filtro de fecha Google Académico. Fuente: Autor

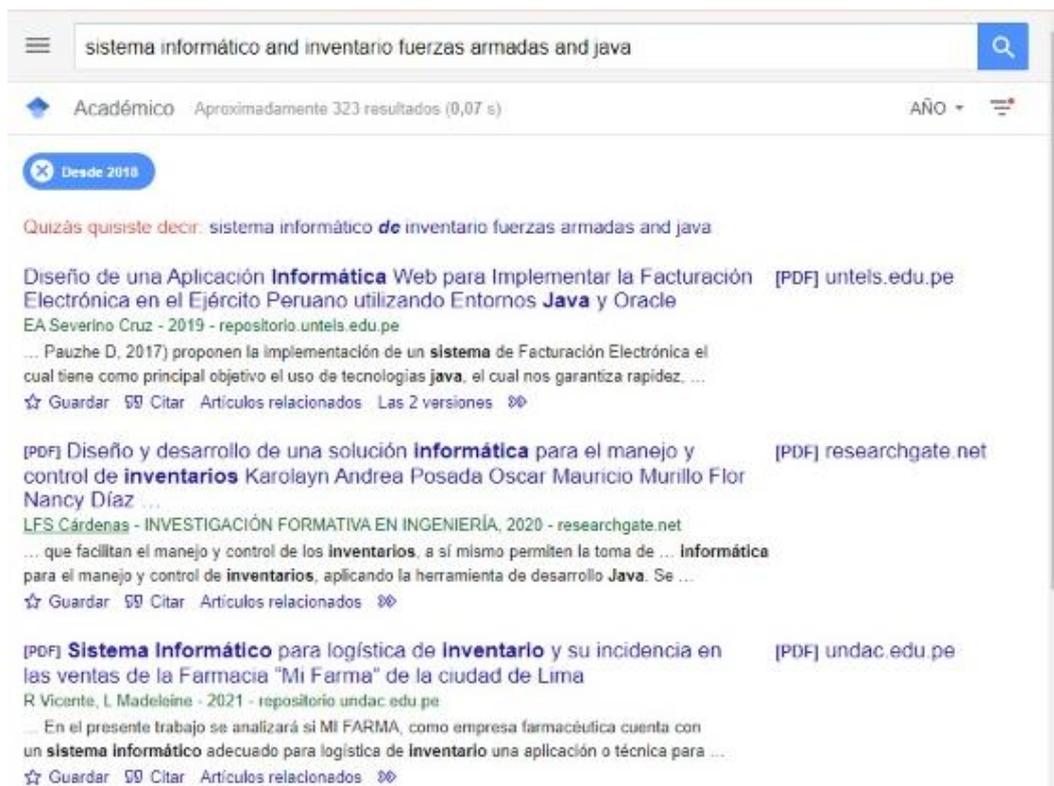


Ilustración 4. Resultados formula búsqueda con filtro de fecha Google Académico. Fuente: Autor

Por último, los resultados de la búsqueda en la biblioteca electrónica de la UCACUE con los términos “Sistema”, “Inventario” y “Java”.



Ilustración 5. Resultados formula búsqueda sin filtro de fecha Biblioteca UCACUE. Fuente: Autor



*Ilustración 6. Resultados formula búsqueda con filtro de fecha Biblioteca UCACUE. Fuente: Autor*

Antes de continuar con la elección de artículos, se delimitaron los criterios de inclusión y exclusión.

### **Criterios de inclusión**

- Presentar una exploración práctica y no de revisiones, análisis de caso único, libros y manuales.
- Que utilicen técnicas de programación orientada a objetos basadas en JAVA Netbeans para generar código basado en 3 capas y se documente su desarrollo.
- Que incluyan el proceso de creación de software de inventario, entendido dentro del marco de las fuerzas armadas.
- Que se documente la creación de una base de datos en un contexto relacional con la base de datos SQL Server.
- Que se utilicen criterios de evaluación de los procesos que cumple el software complementado con la síntesis e interpretación de los resultados.
- Que presente el desarrollo de un API\_RESTfull como solución para cubrir la administración de datos en la web.

### **Criterios de exclusión**

- Se excluyen los estudios que se refieran al proceso de desarrollo de software para dispositivos con sistema operativo Android.
- El software generado no fue realizado con ninguna herramienta de interés.
- El sistema se basa en un aplicativo web y no proporciona seguridad en los procesos.
- El estudio cubre solo la parte de minería de datos.
- El estudio no aporta un software terminado y funcional.

En base a los criterios, y únicamente con la revisión del título y resumen, se consideraron apropiados 150 artículos (tras eliminar 340 duplicados, sin resultados o sin objetivos entre las tres bases de datos) se decidió ejecutar el análisis de manera completa y, partiendo de la lectura, se eliminaron 85 artículos, esencialmente por basarse en aplicativos móviles o web (n = 46), por no tratarse de estudios empíricos o no presentar un software completamente funcional como resultado (n = 14), por utilizar diferentes bases de datos no relacionales o utilizar una metodología muy diferente de la metodología en espiral, lo que ocasionaría una mala interpretación y síntesis del proceso de desarrollo del software.

Por ultimo, 5 artículos cumplieron con los criterios de inclusión y se consideraron para realizar el estudio sistemático. En los que se expone el proceso de creación de una base de datos relacional en SQL Server, Estructuración de interfaces gráficas y modelado en Netbeans, fases de diseño que incluyen los pasos para generar un API\_RESTfull dentro de un contexto basado en un servicio web distribuido, documentación de buenas prácticas para la obtención de requerimientos, y seguridad de la información.

### **Búsqueda manual**

Luego de haber elegido los 5 estudios, y su correspondiente lectura en profundidad, tomando como principio fundamental sus referencias, se incluyó un nuevo análisis de

búsqueda dentro del rango establecido (2018 hasta el presente), considerando Scopus, pero en esta búsqueda no se evidenciaron nuevos estudios que aportaran información valiosa o cumplieran con los parámetros establecidos.

### **Metodología de desarrollo de software**

Los motivos que llevaron a la utilización de la metodología en espiral se fundamentan principalmente, en que combina el modelo en cascada con la metodología por iteraciones, evitando sus inconvenientes y centrándose en la resolución de problemas que puedan generar incertidumbre para el proyecto, mediante varias espirales que se repiten, este modelo garantiza el desarrollo eficiente al basarse en patrones de riegos y multiprocesos, teniendo en cuenta que cada etapa debe iniciar con un objetivo y finaliza con la revisión del avance.

Es de suma importancia al aplicarse en del desarrollo de un proyecto de mediano a alto riesgo, y los requisitos no sean completamente claros o complejos y de ser necesarias varias modificaciones en todo momento, la revisión permanente aporta a la gestión de riesgos y continuamente existe intervención del cliente.

Se complementa este análisis con el estudio realizado sobre otras metodologías de desarrollo de software en el siguiente cuadro comparativo, describiendo sus características principales, procesos, ventajas y desventajas.

## Cuadro comparativo

Tabla 1 Cuadro comparativo de modelos de desarrollo de software Fuente: Autor

Modelos	Espiral	Construcción de prototipos	Incremental	Evolutivo	Cascada
<b>Descripción</b>	Es una metodología de desarrollo de software evolutivo que relaciona sus propiedades iterativas de construcción de modelos con las características controladas y sistemáticas propias de un modelo lineal secuencial.	Brinda una perspectiva excelente, con un diseño rápido, definiendo los requisitos y objetivos.	Administra sucesiones lineales de manera escalonada durante el transcurso del tiempo y el avance en el calendario. Cada repetición lineal genera un «aumento» del software.	Se basa en el proceso de una versión inicial que luego de presentarse se va depurando acorde a las opiniones, recomendaciones o nuevos requerimientos proporcionados por el cliente y el usuario final. Los procesos de descripción, desarrollo y aprobación se entretajan.	Propone una dirección metodológica, secuencial, para el avance del desarrollo de software.
<b>Proceso</b>	Consta de seis pasos a cumplir: 1. Comunicación con el cliente. 2. Planificación 3. Análisis de riesgos 4. Ingeniería 5. Construcción y adaptación 6. Evaluación del cliente	1. Escuchar al cliente 2. Construir/Revisar la maqueta 3. El cliente prueba la maqueta.	1. Análisis 2. Diseño 3. Código 4. Pruebas.	1. Descripción del sistema 2. Especificación 3. Desarrollo 4. Validación 5. Re-especificación	1. Análisis 2. Diseño 3. Código 4. Pruebas
<b>Ventajas</b>	<ul style="list-style-type: none"> <li>• Los costos y la organización se adaptan a la retroalimentación frente a la valoración del cliente.</li> <li>• El administrador del proyecto acuerda el número</li> </ul>	<ul style="list-style-type: none"> <li>• Esta metodología es útil siempre y cuando se hayan obtenido y comunicado los requerimientos del</li> </ul>	<ul style="list-style-type: none"> <li>• Minimiza los tiempos de desarrollo iniciales, a través del desarrollo parcial del software.</li> </ul>	<ul style="list-style-type: none"> <li>• El sistema va concordando a los requerimientos del cliente.</li> <li>• El cliente comprende mejor sus propios requisitos.</li> </ul>	<ul style="list-style-type: none"> <li>• Es una metodología simple y lineal</li> <li>• Consiste en una secuencia de pasos de</li> </ul>



	<p>proyectado de iteraciones necesarias para terminar el software.</p> <ul style="list-style-type: none"> <li>• Se acomoda y emplea a lo largo de la vida de la aplicación de escritorio.</li> </ul>	<p>sistema, sin identificar los requerimientos específicos de entrada, procesos y salida.</p>	<ul style="list-style-type: none"> <li>• Genera un impacto favorable ante el cliente, que se fundamenta en la entrega oportuna y rápida en etapas tempranas de segmentos operativos del software.</li> </ul>	<p>fácil ejecución, es muy utilizado por todo lo que engloba su funcionamiento.</p>	
<p><b>Desventajas</b></p>	<ul style="list-style-type: none"> <li>• En algunos casos llega a ser complicado persuadir a grandes clientes de que un enfoque evolutivo emplea procesos repetitivos.</li> <li>• Necesita una enorme destreza para realizar análisis de riesgos.</li> <li>• Si un riesgo significativo no es detectado y administrado, por consiguiente se generarían inconvenientes.</li> </ul>	<ul style="list-style-type: none"> <li>• La generación de un prototipo en la mayoría de casos no cumple con la mayor parte de requerimientos específicos que se localizaran posteriormente.</li> <li>• Debido a la finalidad de realizar un prototipo de manera rápida e imprecisa, comúnmente se pasan por alto aspectos importantes, como la eficiencia y la estabilidad del software a largo plazo.</li> </ul>	<ul style="list-style-type: none"> <li>• La mayor parte de funcionalidades complementarias, algunas distinguidas, se encuentran sin ejecutar.</li> <li>• El método ejecuta la transformación del software principal con la finalidad de cubrir de mejor forma los requerimientos del cliente</li> </ul>	<ul style="list-style-type: none"> <li>• El proceso no se visibiliza de forma optimizada.</li> <li>• Las modificaciones repetidas provocan elementos erróneos en la estructura del software. Agrega modificaciones en donde se transforma en una actividad difícil y costosa.</li> </ul>	<ul style="list-style-type: none"> <li>• Su aplicación a proyectos de la vida real en muy pocas ocasiones cumple con los parámetros y procesos que ofrece el modelo.</li> <li>• Es difícil que el cliente explique claramente todos los requerimientos.</li> </ul>

**Información primaria:** Se consiguió por medio de los encargados de las diversas dependencias de la organización, especialmente los que se relacionan con la descripción de los procesos de bodegaje; los detalles de características de procesos de gestión de materia, se consiguieron también aplicando un enfoque basado en la observación directa sobre fuentes de información primaria fundamentada en indicaciones proporcionadas por los encargados de las áreas de bodega y departamento de informática. La demás información secundaria es información agregada que engloba la exploración y estudio de documentos e informes del sistema de información de la empresa en relación a su contexto actual, complementado con el análisis de diferentes fuentes de datos referentes a investigaciones sobre procesos de gestión de inventarios.

**Proceso metodológico:** El modelo en espiral permite que el progreso de creación de un programa se fundamente en el período de vida del software en forma iterativa, en esta sección se pone en marcha la ejecución del proceso de desarrollo basado en la guía evolutiva por medio del ciclo de la vida para la realización de prototipos, es por ello que se encarga de realizar avances acerca de las tareas.

Se consideraron las siguientes actividades propias del método en espiral aplicadas al presente trabajo que se ejecutaron de la siguiente forma:

1. Dar aviso al cliente: Se determina un horario para realizar reuniones vía zoom y presenciales obteniendo como resultado una mejor comunicación entre el desarrollador y el cliente sobre el avance del sistema, mejorando también la obtención de detalles sobre aspectos funcionales importantes.
2. Planificación: Se definen los recursos necesarios relacionadas con el desarrollo de cada avance del proyecto, así como los requerimientos.

3. Análisis de riesgos: Se aplica una metodología de análisis de riesgos “What if” esta herramienta sencilla, eficiente y fácil de aplicar para todo tipo de administrador, que consiste en una técnica de análisis inductivo que parte de la verificación de información determinada sobre un proceso, generando preguntas útiles para la verificación del desarrollo e instalación de un sistema. Se utiliza al culminar el segundo paso del ciclo de gestión cuando recién se considera poner en marcha la ejecución de la planificación.
4. Ingeniería: Se construyen representaciones sobre una parte del avance del sistema.
5. Construcción y adaptación: En este punto del modelo espiral requiere cubrir las necesidades de construcción, aprobación, instalación del sistema, con objetivos de dar soporte al usuario.
6. Evaluación del cliente: Se obtienen aspectos positivos de los clientes mediante representaciones realizadas del software en la etapa de implementación e instalación generando retroalimentación.

### **Aplicación metodología espiral**

**Dar aviso al cliente:** Se crearon reuniones virtuales y presenciales donde se definieron las bases fundamentales para la construcción de un bosquejo en el cual se especifiquen las limitaciones, el alcance, funciones principales y disposición de recursos de la organización para el desarrollo del sistema.

**Planificación:** La creación de una planificación de actividades estaba programada para el primer mes de desarrollo se obtuvieron algunos de los requerimientos del sistema de forma general, una descripción de los procesos y se realizó una valoración del estado actual de las bodegas de material bélico y de intendencia de la DIRMOV, en esta fase inicial, se tuvieron en cuenta datos y privilegios de usuarios según su base-centro, y se realizaron valoraciones de tiempo estimado, y planificación del calendario y los recursos disponibles para las fases posteriores.

**Análisis de riesgos:** Con la aplicación de la metodología “What if” se validó la planificación de actividades generada realizando cambios conforme surgían inconvenientes con cada causa y efecto de realizar un cambio o aplicar una forma específica de desarrollo para cada requerimiento, también se tuvieron en cuenta los plazos de tiempo y documentación entregada por el cliente.

**Ingeniería:** Se cumplió con el desarrollo de las siguientes actividades:

- Obtener los requerimientos del sistema.
- Desarrollar una base de datos mediante SQL Server 2019.
- Crear un API Rest por medio del API JAX-RS.
- Implementar los requerimientos del sistema mediante la creación de un software en el lenguaje Java y el IDE Netbeans.
- Realizar pruebas de ejecución y funcionamiento del software.

**Construcción y adaptación:** En esta etapa se terminó el desarrollo del software y se validó como prototipo, según el alcance acordado.

**Evaluación del cliente:** El prototipo se instaló y probó en varios ordenadores, que se manejan en condiciones controladas, para posteriormente evaluar en base a los resultados de seguimiento y ejecución de funciones específicas.

## Cronograma de actividades

ACTIVIDADES	TIEMPO															
	Sem 1	Sem 2	Sem 3	Sem 4	Sem 5	Sem 6	Sem 7	Sem 8	Sem 9	Sem 10	Sem 11	Sem 12	Sem 13	Sem 14	Sem 15	Sem 16
1 Obtener los requerimientos del sistema. Crear reuniones virtuales donde se definan las bases fundamentales para la construcción de un bosquejo donde se definan las limitaciones, el alcance y funciones principales del sistema. Documentar las entidades con sus correspondientes atributos y la distribución de accesos a cada menú que implementara la interfaz gráfica del sistema.																
2 Desarrollar una base de datos. Creación de un Modelo de datos conceptual Creación de un Modelo de datos lógico Creación de un Modelo de datos Físico en la base de datos SQL SERVER.																
3 Crear un API Rest. Crear una conexión mediante del IDE de desarrollo Neatbeans con la base de datos SQL SERVER. Aplicar el proceso de RESTFULL que implementa Neatbeans para la generación del API REST correspondiente. Modificar el código generado en el proceso REST-FULL para que cumpla con las funciones CRUD específicas de cada una de las entidades.																
4 Implementar los requerimientos del sistema mediante la creación de un software. Crear la capa de presentación con el menú principal conjuntamente con las interfaces gráficas para las 48 entidades. Crear la capa de datos con las 48 entidades y sus correspondientes atributos. Crear la capa de negocios con el proceso CRUD para las 48 entidades correspondientes. Creación de la capa de seguridad para el inicio de sesión de acuerdo a los privilegios de cada usuario.																
5 Realizar pruebas de funcionamiento y ejecución del software. Comprobar la precisión al momento de realizar los procesos de carga de datos, siendo este un factor crucial para el éxito del sistema. Comprobar el proceso de actualización de registros tanto de material bélico como material de intendencia y movimientos de material. Revisar los campos del listados de registros al momento de generar un FindAll(Recuperar todos los registros). Revisar el proceso de eliminación de un registro y el proceso de aprobación por parte de cada uno de los responsables. Poner en marcha el sistema con la correcta instalación de la base de datos, iniciar el servidor y descargar el aplicativo ejecutable en las maquinas correspondientes.																

Ilustración 7. Tabla cronograma de actividades metodología. Fuente: Autor

## Resultados

Los resultados se dividen en dos partes la primera involucra los resultados que se obtuvieron con la aplicación del análisis a la organización, y el segundo con el desarrollo del sistema.

**Resultados del análisis a la organización:** estos resultados se fundamentan alrededor de las respuestas a las preguntas y objetivos de investigación establecidos. A continuación, se presenta la información derivada del desarrollo del sistema de gestión de inventarios y movimientos, de esta manera se brindó a la organización un instrumento valioso para el registro, control y organización.

Mediante la ejecución de una sucesión de actividades de levantamiento y análisis de requerimientos, se logró obtener un conjunto importante de información de los diversos sistemas que brindan control de inventarios.

Con el análisis y diagnóstico se comprobó, que, a pesar de existir algunos controles elementales simples, los resultados del análisis aplicado fueron los siguientes puntos:

- La DIRMOV no posee un sistema que se encargue exclusivamente del control de inventario.
- Inconsistencias de inventarios (sistema vs físico).
- Demoras, alteraciones, redundancia y falta de controles en el proceso de registro de material, esto se exterioriza por la mala asignación en las bases-centros de la DIRMOV.
- Demora de digitación e impresión de reportes, preparación, notificaciones de entrega y consultas.
- Carencia de clasificación por tipo de material que permita identificar con mayor rapidez y por características específicas, todo esto impide identificar que materiales son los requeridos, dicho escenario, coloca a la organización en un escenario de peligros, inminentes.

- Falta de restricciones al momento de realizar las diferentes actividades de acuerdo con sus definiciones o conveniencias individuales, por lo que puede generar conflictos en la ejecución de operaciones sin autorización.
- Existencia de errores o irregularidades con el manejo del inventario, tales como: registros incorrectos o incongruentes con un campo de datos.

**Resultados obtenidos con el desarrollo del sistema:** Para determinar este punto se tomaron en cuenta los lineamientos iniciales y objetivos a cumplir.

- El sistema cubre todos y cada uno de los requerimientos programados al inicio.
- El sistema ofrece seguridad de los datos, mediante la utilización de privilegios de numerosos tipos de usuarios, con diferentes permisos cada uno, como son: de lectura, eliminación, actualización y escritura sobre los registros y acceso a ventanas.
- El sistema permite integrar a los procesos de registro y movimientos en un solo lugar, reduciendo así la pérdida de tiempo; agilizando la recolección de datos y optimando la visualización de los resultados obtenidos.
- Los datos que se registrar el sistema son confiables, de forma que ayudan en la toma de decisiones dentro de la organización permitiendo validar si se cumple con la planificación del plan operativo de la DIRMOV.

## Conclusiones

- Se concluye con el análisis y desarrollo realizado que la existencia de un sistema de inventarios mejora en gran medida la gestión de material evitando la pérdida de tiempo y recursos.
- Con la implementación del sistema se logró el centralizado de toda la información en una sola base de datos.
- La implementación del sistema de inventario garantizo a la organización una disminución de las fallas, que se presentan dentro de las diferentes Bases-Centros y así llevar una eficiente y exitosa administración de los recursos existentes.
- Las herramientas tecnológicas utilizadas favorecieron el desarrollo e implementación del sistema de gestión de inventario.
- Con el motor de base de datos SQL Server 2019 se diseñó la estructura y organización de los datos de forma óptima, teniendo en cuenta la integridad de los mismos, dando como resultado una base de datos sólida en cuanto a integridad referencial.
- Por otro lado, el desarrollo del aplicativo en lenguaje de programación Java permitió ejecutarlo en diferentes sistemas operativos como Windows y Linux.
- Con el despliegue del API RESTful se obtuvo una excelente interfaz de comunicación entre el sistema y la base de datos, generando así una correcta administración de los recursos.
- El modelo de desarrollo en espiral brindo el soporte necesario y correspondiente al trabajo de un solo desarrollador para la creación del sistema, generando una aplicación de gestión de inventarios funcional que permitirá a la organización tener mayor claridad sobre el estado de su inventario.



- La implementación de diversos filtros de búsqueda, mejoró a la consulta de elementos disponibles y la verificación del estado en el que se encuentran dentro de un determinado proceso de préstamo.
- Mediante la combinación del trabajo de diversas áreas de conocimiento como son, el diseño de interfaces gráficas, desarrollo de API\_RESTfull, codificación, diseño de base de datos, obtención de requerimientos y la ingeniería en general, se logró la creación de un sistema funcional acorde a los requerimientos definidos.

## CAPITULO II

### Manual de usuario

El sistema es una aplicación Java de escritorio que facilita el registro, actualización y administración de la información sobre los procesos que involucren elementos tanto de material bélico como material de intendencia para la DIRMOV.

El sistema está desarrollado en lenguaje Java a través del IDE NetBeans, el API RESTful fue generado mediante JAX-RS y se manejan los datos de las peticiones en formato JSON, la base de datos se creó mediante SQL server 2019, el modelado lógico, conceptual y físico en PowerDesigner.

#### Aplicación de escritorio

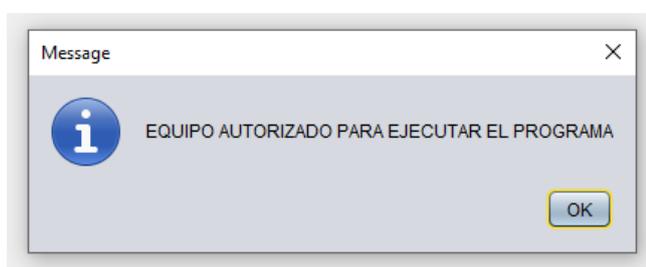
El sistema es un aplicativo diseñado teniendo en cuenta los siguientes parámetros:

1. Facilidad de manejo, comprensión y aprendizaje que involucren los procesos del sistema. Es decir, el sistema debe permitir el manejo por parte de usuarios de todo tipo sin la necesidad de que sean expertos
2. Presentación estática de un determinado contexto de operación (fondo).
3. El elemento o proceso de interés será de fácil identificación
4. Diseño simple e intuitivo, adaptable al usuario mediante el establecimiento de elementos como menús, tablas, cajas de texto, barras de acciones e iconos de acceso práctico.
5. La interacción del usuario con el sistema se basará en acciones directas sobre elementos visuales (iconos, tablas, imágenes, opciones de selección, botones, Cajas de texto, barras de desplazamiento).
6. Las acciones serán rápidas, y con la posibilidad de ser reversibles.

7. Presencia de apartados de Ayuda y Consulta
8. Procedimientos bien definidos ante errores, restricciones ante la ausencia de privilegios de usuario, limitando operaciones no autorizadas.

### **Ingreso al Sistema**

- Al momento de ejecutar el sistema se verifica la dirección MAC del ordenador, siendo este un identificador único que pertenece a un dispositivo de red de un computador y que deberá constar en un registro dentro de la base de datos.



*Ilustración 8. Ventana emergente comprobación sistema. Fuente: Autor*

- Para iniciar sesión en el sistema se presenta una pantalla donde se deben ingresar las credenciales correspondientes y dar clic en el botón ingresar para validar los datos, de ser correctos se desplegará el menú principal con un mensaje de bienvenido, caso contrario se mostrará un mensaje de datos incorrectos.



*Ilustración 9. Ventana inicio sesión sistema. Fuente: Autor*

- Una vez que se realiza el ingreso exitoso al sistema, se tiene acceso a las diferentes funcionalidades que este implementa de acuerdo a los privilegios de usuario y el acceso a la información correspondiente a su Base-Centro.

## Pantalla inicio

Los primeros cuatro menús principales son de acceso público para todos los usuarios, son pantallas de carácter informativo, donde se ofrece un informe detallado sobre los servicios que brinda la institución, normativa legal, datos de contacto, y centros que disponen en todo el Ecuador.

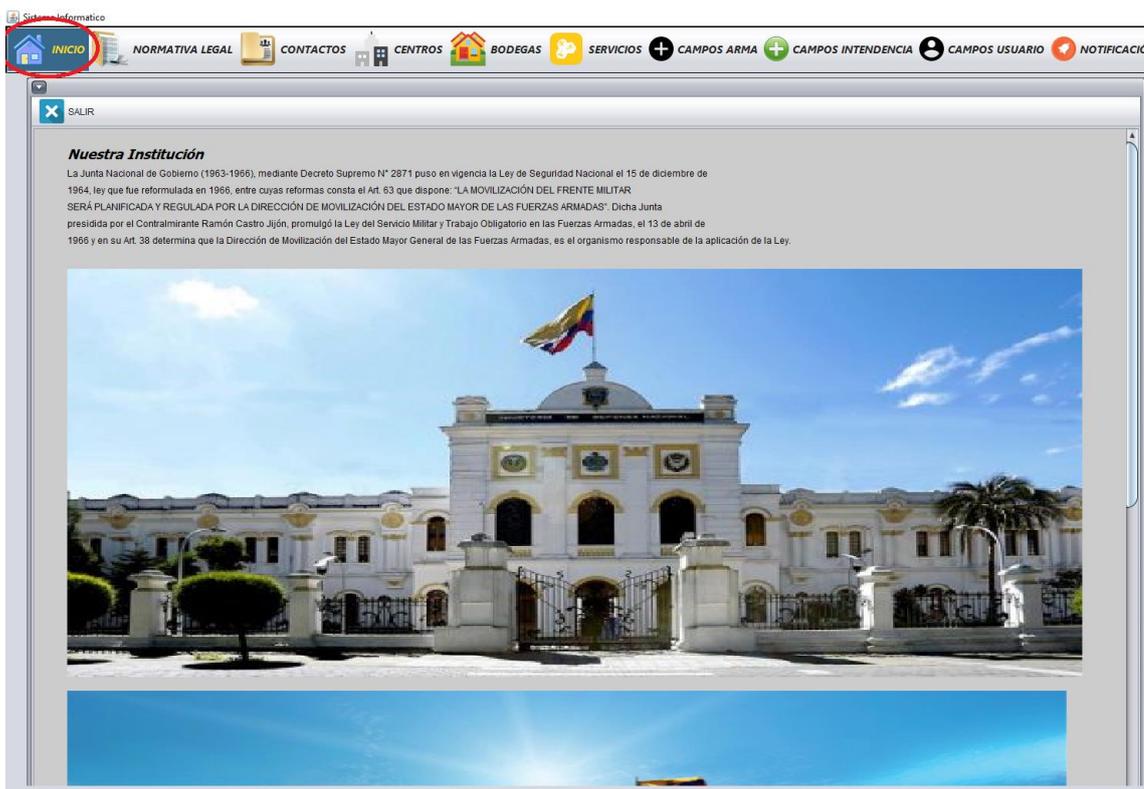


Ilustración 10. Ventana inicio sistema. Fuente: Autor



Ilustración 11. Ventana normativa legal sistema. Fuente: Autor

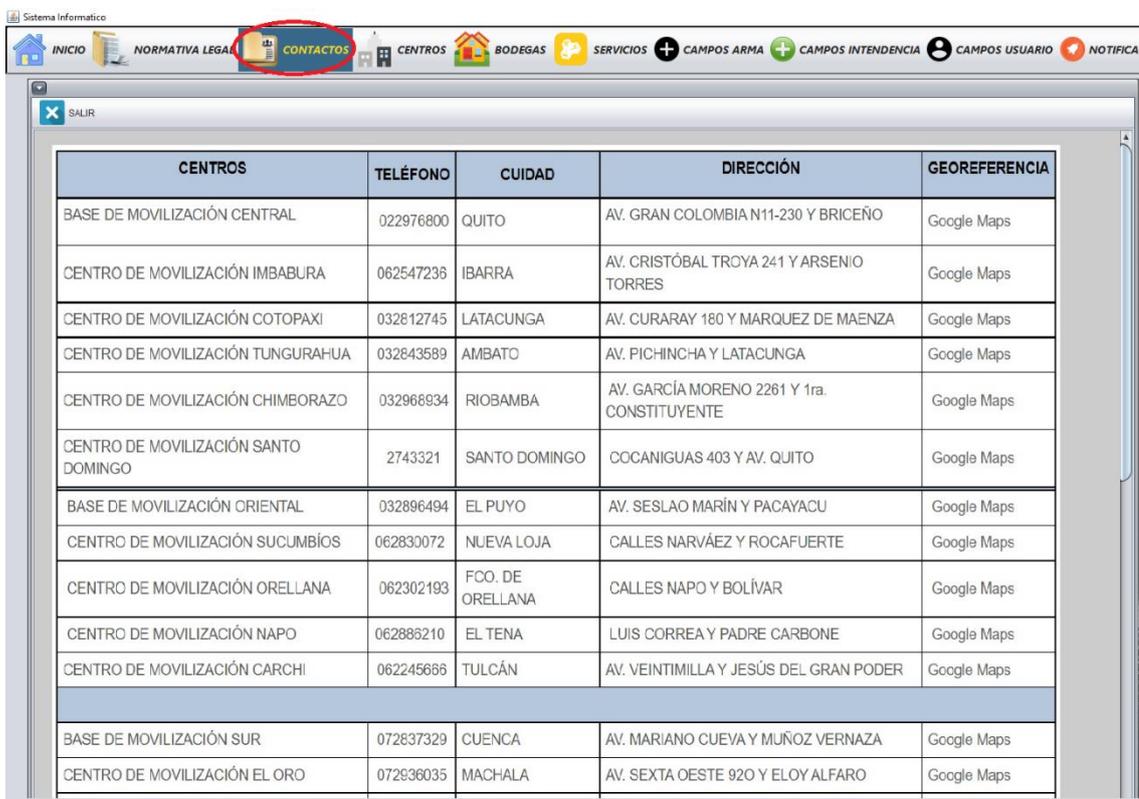


Ilustración 12. Ventana contactos sistema. Fuente: Autor

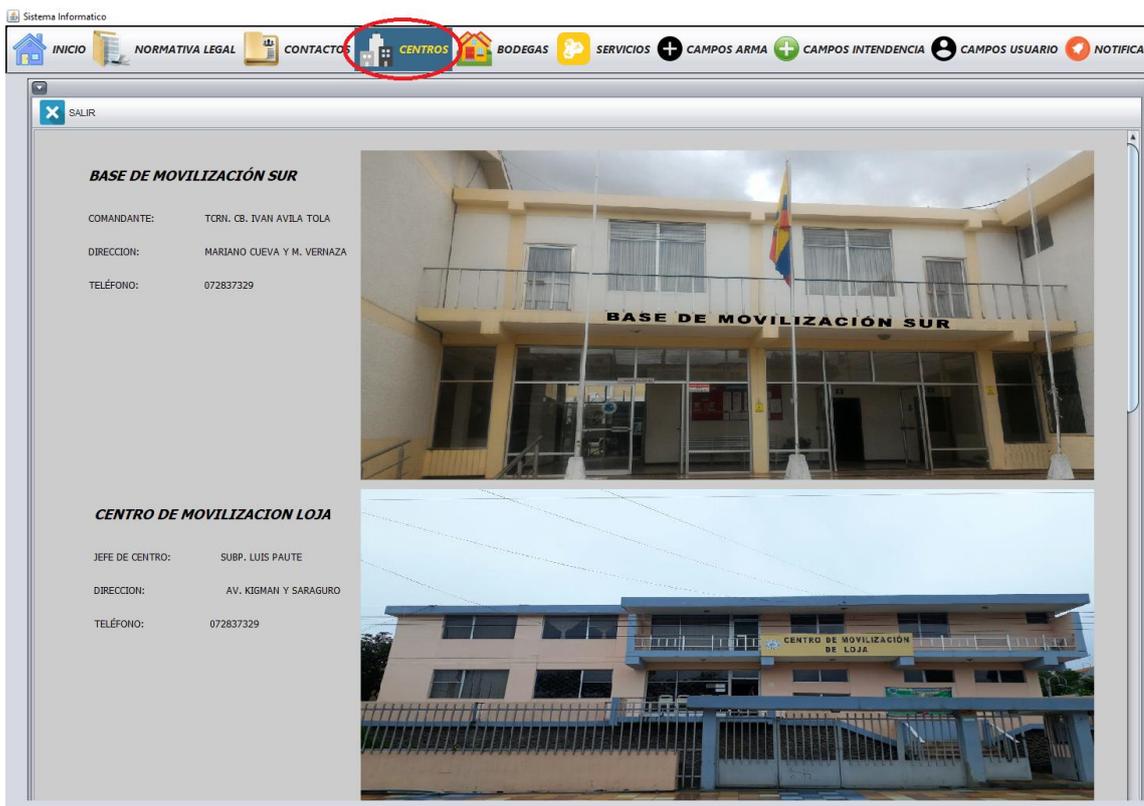


Ilustración 13. Ventana centros sistema. Fuente: Autor

### Pantalla material bélico administración de privilegios

Para acceder a esta pantalla se debe seleccionar el menú principal “Bodegas” luego “Arma” y por último el submenú “Administración arma” al iniciar se ejecuta el proceso de verificación de privilegios conjuntamente con la aplicación del filtro de registros y opciones propias de la Base-Centro al que corresponda el usuario correspondiente.



Ilustración 14. Menú administración arma sistema. Fuente: Autor

1. En el apartado de la izquierda se encuentran los diferentes atributos de creación para un registro, donde depende de la Base-Centro del usuario que posea, para bloquear

el combo “Base-Centro” restringiendo la posibilidad de que un usuario pueda crear un registro que no corresponda a su ubicación.

2. También se desplegará un combo con la lista de Base-Centro disponibles si el usuario que ingreso posee privilegios totales.
3. Se observan los registros luego de la aplicación de un filtro de Base-Centro logrando así obtener una mejor visión de la información de forma precisa.



Ilustración 15. Ventana base centro sistema. Fuente: Autor

### Pantalla administración de material bélico

En esta pantalla se tiene a disposición las funciones necesarias para administrar los procesos que correspondan al manejo de material bélico.

1. Como primera función se presenta “Ocultar campos” esta opción ofrece la posibilidad de listar los registros sin campos innecesarios reduciendo así la información de cada elemento y facilitando la visualización de los campos principales.
2. Esta opción permite añadir campos más detallados a la vista principal de los registros.
3. Si se necesita acceder a la información de un elemento en concreto se debe de ingresar el número de serie del registro en el campo correspondiente y se lista el registro en la tabla.

4. Esta es la tabla que contiene los registros de los elementos de material bélico ingresados y disponibles para su administración dentro de la base de datos.
5. Al dar clic en un registro se cargan los datos en los campos de la parte izquierda, y paralelamente se realiza la verificación si el elemento está en proceso de actualización o si se puede generar una solicitud ahora, también si el estado de la solicitud es aprobado y puede actualizar.
6. Estos combos corresponden a las tablas auxiliares donde brindan ayuda al seleccionar una opción en vez de tener que escribir la información mediante la carga de los registros creados con anterioridad.
7. Estos campos permiten ingresar información irrepitible en algunos casos donde es propia de cada elemento de material bélico, como su número de serie y ubicación.
8. Por último, los menús superiores permiten, limpiar los campos, guardar los registros, desplegar el panel de búsqueda, generar una solicitud de actualización de campos de un elemento, y actualizar.

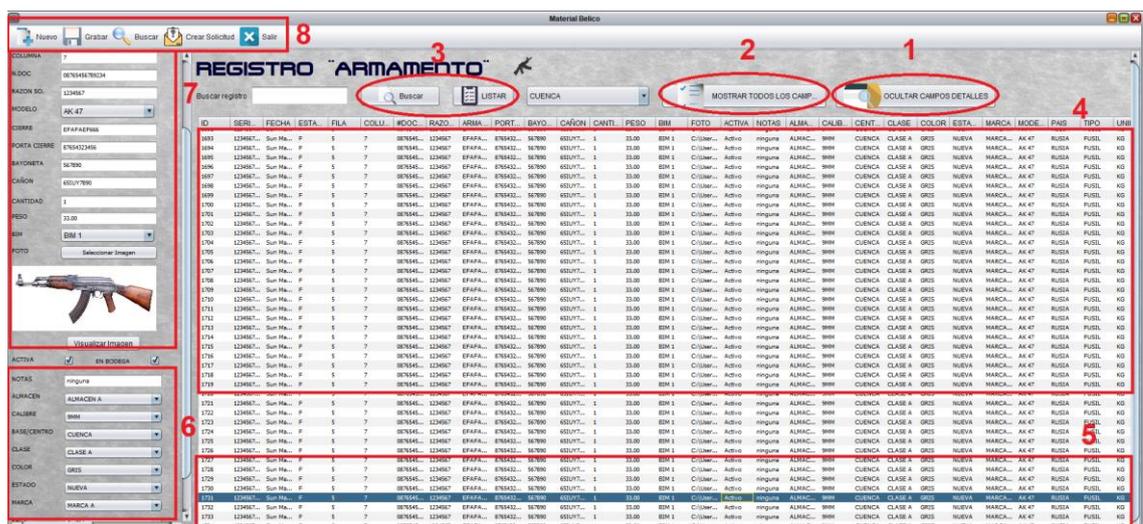


Ilustración 16. Ventana administración material bélico sistema. Fuente: Autor

## Pantalla solicitud de actualización material bélico

Se presentan los campos correspondientes para registrar la información necesaria de una solicitud de actualización.

1. Se graba la solicitud una vez los campos estén completos y se verifique que sean correctos.
2. Subir un archivo “PDF” donde consta la información impuesta por el personal administrativo interno de la institución.
3. Visualizar el documento para verificar si es el correcto antes de generar el registro.
4. Selección de los campos a ser modificados del registro correspondiente



Ilustración 17. Ventana solicitud actualización sistema. Fuente: Autor

## Pantalla historial solicitud actualización material bélico

Se accede a esta opción a través del menú “Bodegas”, luego “Arma” y por último el sub menú “Historial solicitud actualización” donde se visualizan los campos necesarios para administrar las solicitudes de modificación de campos de un registro.



Ilustración 18. Menú historial solicitud sistema. Fuente: Autor

1. Al momento de seleccionar un registro se procede a verificar los privilegios correspondientes del usuario, y cargar la información en los campos de la izquierda.
2. Se listan las solicitudes que estén pendientes de aprobación.
3. Se listan todos los registros de solicitudes donde conste el usuario sin importar si están aprobadas o no.
4. Se visualizan los usuarios que han solicitado esa modificación de registro, detallando al solicitante y los usuarios que hayan aprobado la solicitud.
5. Se visualiza el documento con información pertinente al proceso de actualización.
6. Se visualizan los campos que solicitan ser modificados.
7. El usuario con privilegios puede aprobar o rechazar esta solicitud.

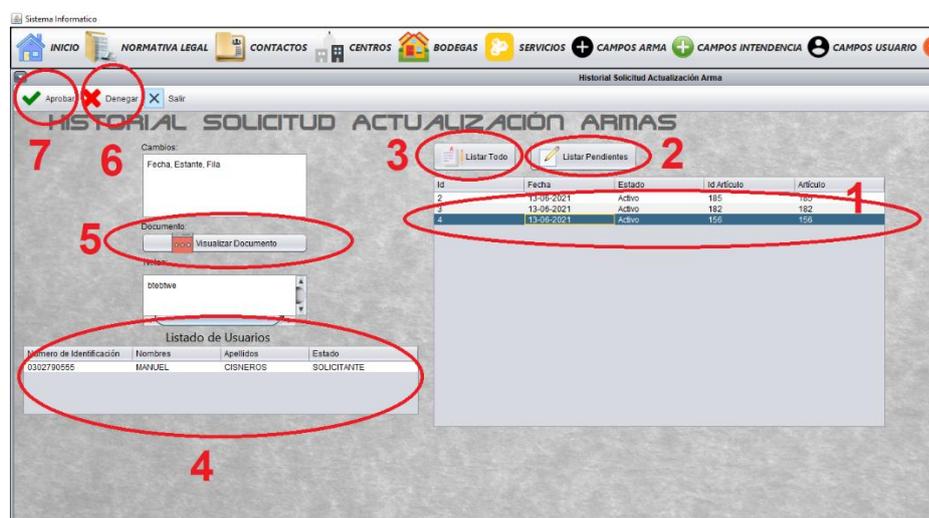
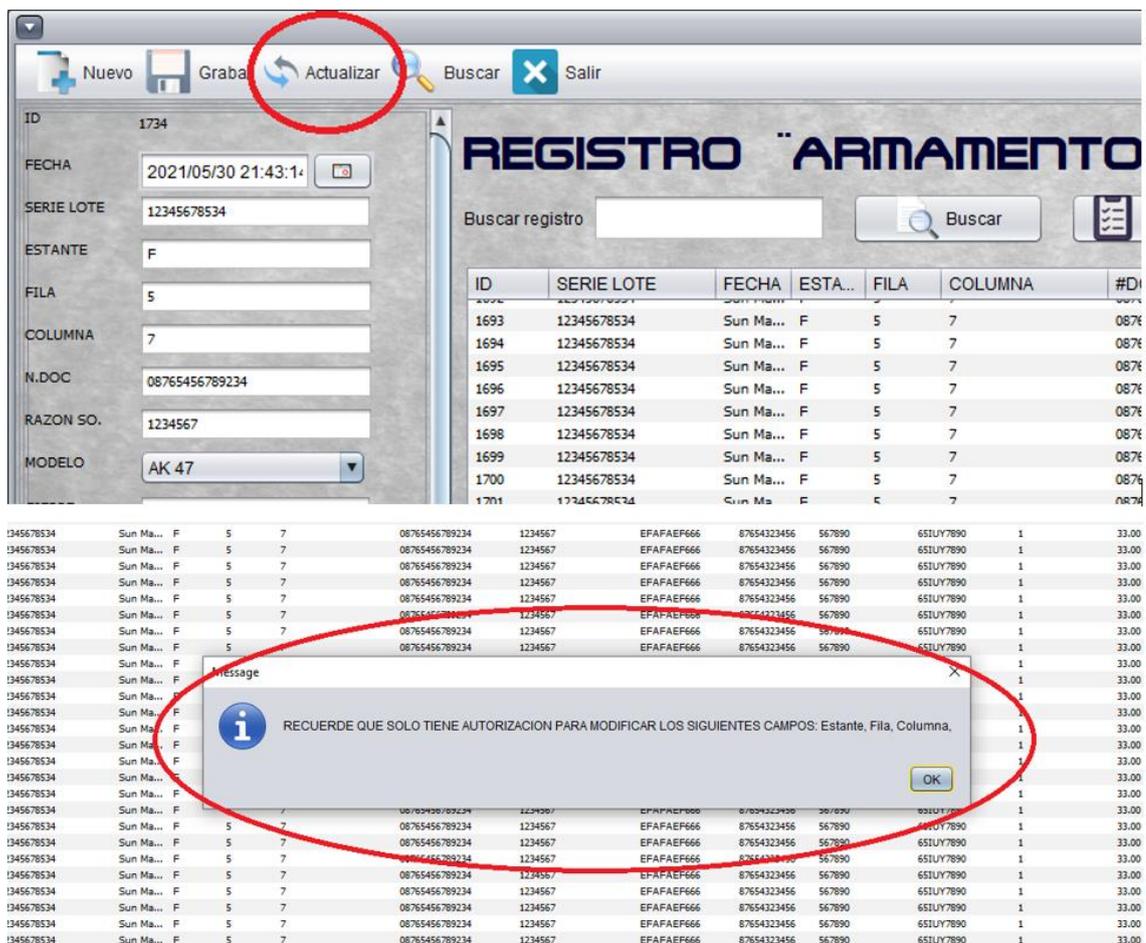


Ilustración 19. Ventana historial solicitud actualización sistema. Fuente: Autor

## Pantalla actualizar registro de material bélico

Para cumplir con este proceso se debe de seleccionar el registro y se verifica internamente si la solicitud ya está aprobada y se activa la opción en el menú principal “Actualizar” caso contrario se visualiza un mensaje de “Solicitud no aprobada” o “El usuario 123... rechazo la solicitud” y se procede a eliminar la solicitud activa, con la posibilidad de crear otra.

Tomando como verdad que ya se aprobó la solicitud, se puede modificar los campos, pero solo los que se incluyeron en la solicitud, caso contrario se actualizara y pedirá que verifique los campos.



The screenshot displays the 'REGISTRO ARMAMENTO' application interface. At the top, a menu bar includes 'Nuevo', 'Graba', 'Actualizar', 'Buscar', and 'Salir'. The 'Actualizar' button is highlighted with a red circle. Below the menu, there are input fields for 'ID' (1734), 'FECHA' (2021/05/30 21:43:14), 'SERIE LOTE' (12345678534), 'ESTANTE' (F), 'FILA' (5), 'COLUMNA' (7), 'N.DOC' (08765456789234), 'RAZON SO.' (1234567), and 'MODELO' (AK 47). A search bar labeled 'Buscar registro' is also present. The main area contains a table with the following data:

ID	SERIE LOTE	FECHA	ESTA...	FILA	COLUMNA	#DI
1693	12345678534	Sun Ma...	F	5	7	0876
1694	12345678534	Sun Ma...	F	5	7	0876
1695	12345678534	Sun Ma...	F	5	7	0876
1696	12345678534	Sun Ma...	F	5	7	0876
1697	12345678534	Sun Ma...	F	5	7	0876
1698	12345678534	Sun Ma...	F	5	7	0876
1699	12345678534	Sun Ma...	F	5	7	0876
1700	12345678534	Sun Ma...	F	5	7	0876
1701	12345678534	Sun Ma...	F	5	7	0876

In the foreground, a message dialog box is displayed with the following text: 'RECUERDE QUE SOLO TIENE AUTORIZACION PARA MODIFICAR LOS SIGUIENTES CAMPOS: Estante, Fila, Columna.' The dialog box has an 'OK' button. The background table continues with many more rows of data, including columns for 'EFAPAEF666', '87654323456', '567890', and '651UY7890'.

Ilustración 20. Ventana actualizar sistema. Fuente: Autor

## Pantalla mantenimiento material bélico

Para acceder a esta pantalla se selecciona el menú principal “Bodegas” luego “Arma” y finalmente el sub menú “Administración mantenimiento arma”.

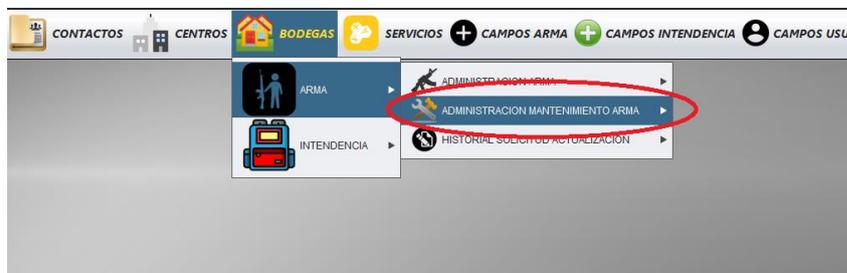


Ilustración 21. Menú mantenimiento material bélico sistema. Fuente: Autor

1. Listar los registros de mantenimientos anteriormente realizados por el usuario correspondiente.
2. Se busca un registro que cumpla con los parámetros y privilegios.
3. Se visualizan en la tabla los elementos detallando su, ID, tipo de mantenimiento, fecha, estado y notas.
4. Limpia los campos para facilitar el proceso de registro.
5. Guarda un registro de mantenimiento siempre y cuando todos los campos cumplan con los requerimientos.
6. Muestra el panel de búsqueda.
7. Permite seleccionar los campos de encabezado de registro.
8. Busca un elemento de material bélico, agrega detalles del proceso de mantenimiento y los añade a la lista.
9. Permite activar la selección de un elemento de material bélico que se desee eliminar.
10. Elimina el registro seleccionado.



Ilustración 22. Ventana mantenimiento arma sistema. Fuente: Autor

1. Selecciona un registro de acuerdo a la Base-Centro del usuario.
2. Verifica los privilegios y muestra un mensaje para realizar la modificación o no.

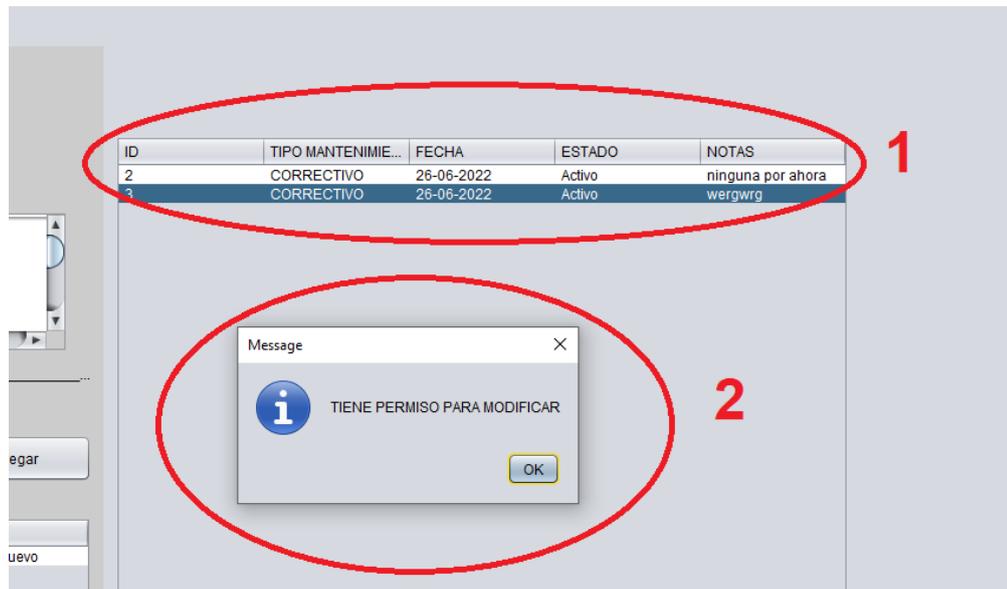


Ilustración 23. Ventana mantenimiento sistema. Fuente: Autor

### Pantalla administración intendencia

Para acceder a esta pantalla se selecciona el menú principal “Bodegas”, luego “Intendencia” y por último el sub menú “Administración intendencia”.

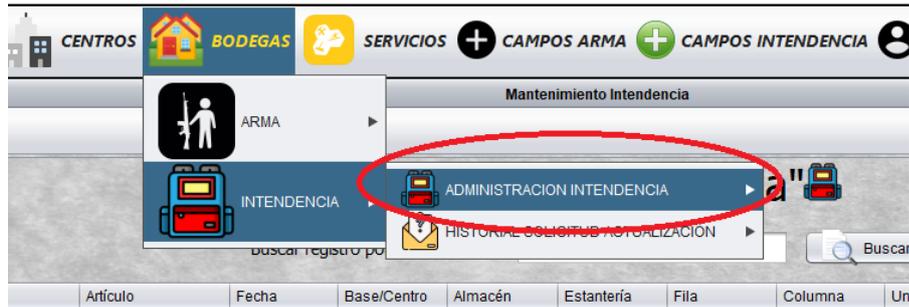


Ilustración 24. Menú intendencia sistema. Fuente: Autor

1. Busca un registro que cumpla con el parámetro de búsqueda “nombre del registro”.
2. Lista los registros que concuerden con los datos de Base-Centro del usuario.
3. Se visualiza la información de registros con la posibilidad de generar una solicitud de actualización de un campo al dar clic.
4. Campos para ingresar o seleccionar los datos de un registro.
5. Limpia los campos para facilitar el proceso.
6. Comprueba los datos ingresados y guarda el registro.
7. Activa el panel de búsqueda.
8. Los campos de los combos permiten seleccionar un dato para no escribirlo evitando errores de ingreso.

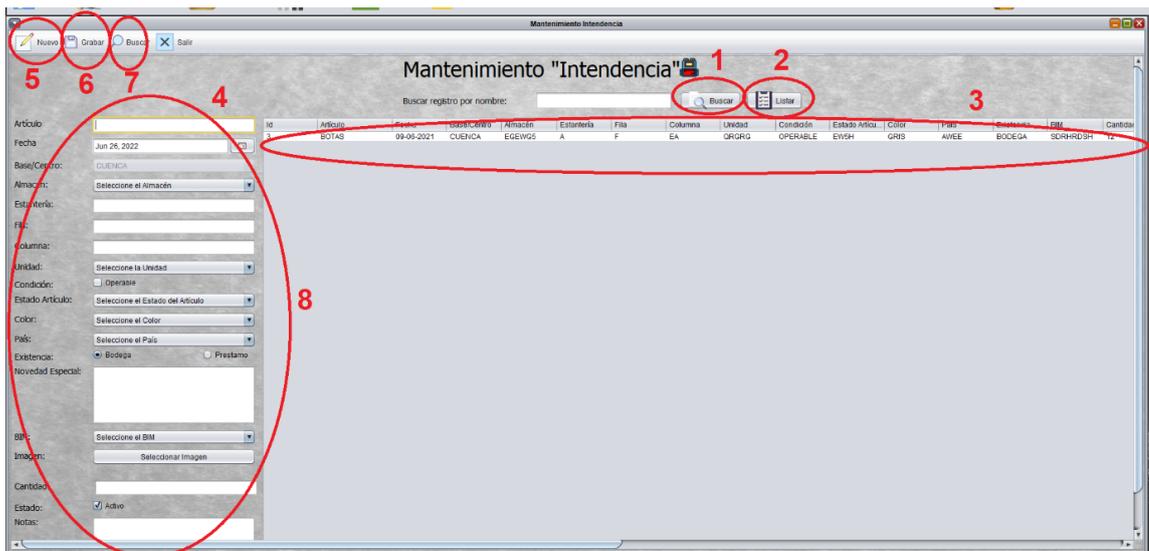


Ilustración 25. Ventana mantenimiento intendencia sistema. Fuente: Autor

## Pantalla administración usuarios

Para acceder a esta pantalla se selecciona el menú principal “Servicios” y luego el sub menú “administración usuarios”.

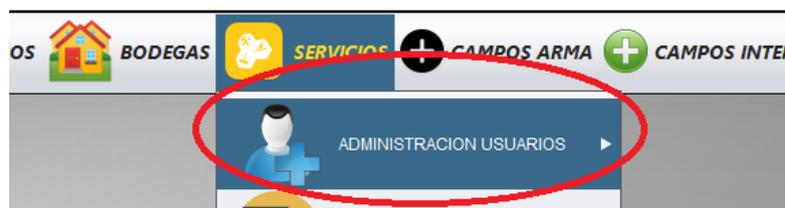


Ilustración 26. Menú administración usuario sistema. Fuente: Autor

1. Busca un registro de acuerdo al parámetro de búsqueda “número de identificación”.
2. Tabla donde se listan los registros de usuarios, al momento de dar clic se verificarán y cargarán los datos en los campos de la izquierda.
3. Campos de ingreso de datos para la generación de un registro.
4. Verifica los datos ingresados en los campos y generar el registro y comprueba que un elemento esta seleccionado en la tabla tendrá como función actualizar.
5. Activa el panel de búsqueda.
6. Elimina un registro.

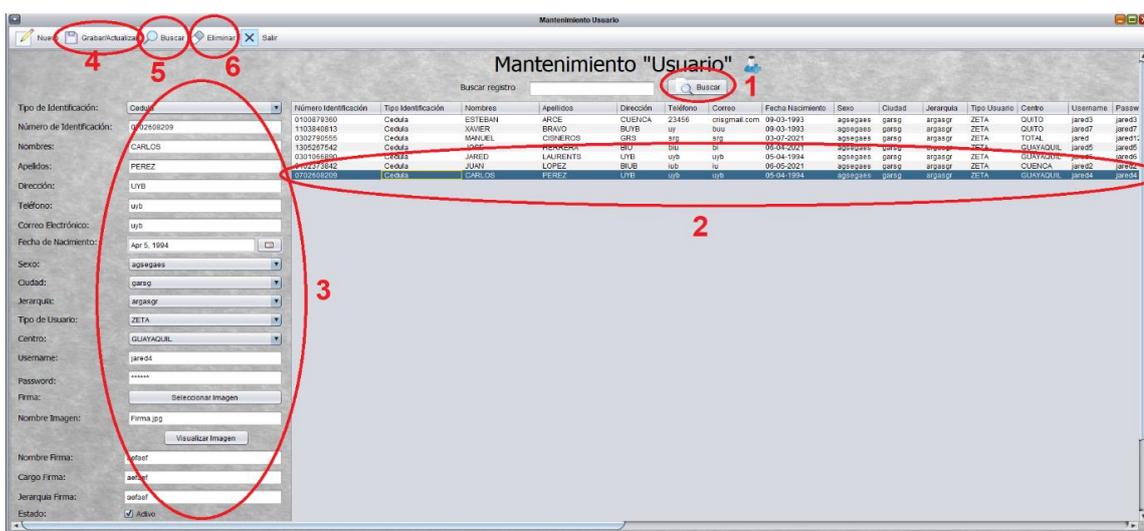


Ilustración 27. Ventana mantenimiento usuario sistema. Fuente: Autor

## Pantalla consulta arma

Para acceder a esta pantalla se selecciona el menú principal “Servicio” y el sub menú “Consultas” y luego en el submenú “Arma”.



Ilustración 28. Menú consultas material bélico sistema. Fuente: Autor

1. Aquí se visualizan los registros correspondientes.
2. El primer combo posee los encabezados de la tabla que son las columnas específicas de la tabla y se listarán de acuerdo a la opción seleccionada.
3. En el segundo combo filtra los registros por el tipo de material bélico.
4. En el tercer combo se filtran los registros de acuerdo al estado del elemento de material bélico.
5. En el quinto combo se puede elegir la base centro al que correspondan los registros.
6. El sexto combo filtra los registros de acuerdo al BIM al que correspondan.
7. El séptimo combo filtra los registros por modelo.
8. En este apartado se debe ingresar el número de serie o el número de documento y se filtrará un solo registro.
9. Por se puede agregar usuarios al documento generado, con la posibilidad de exportar los registros a Excel si se necesita realizar modificaciones extras.

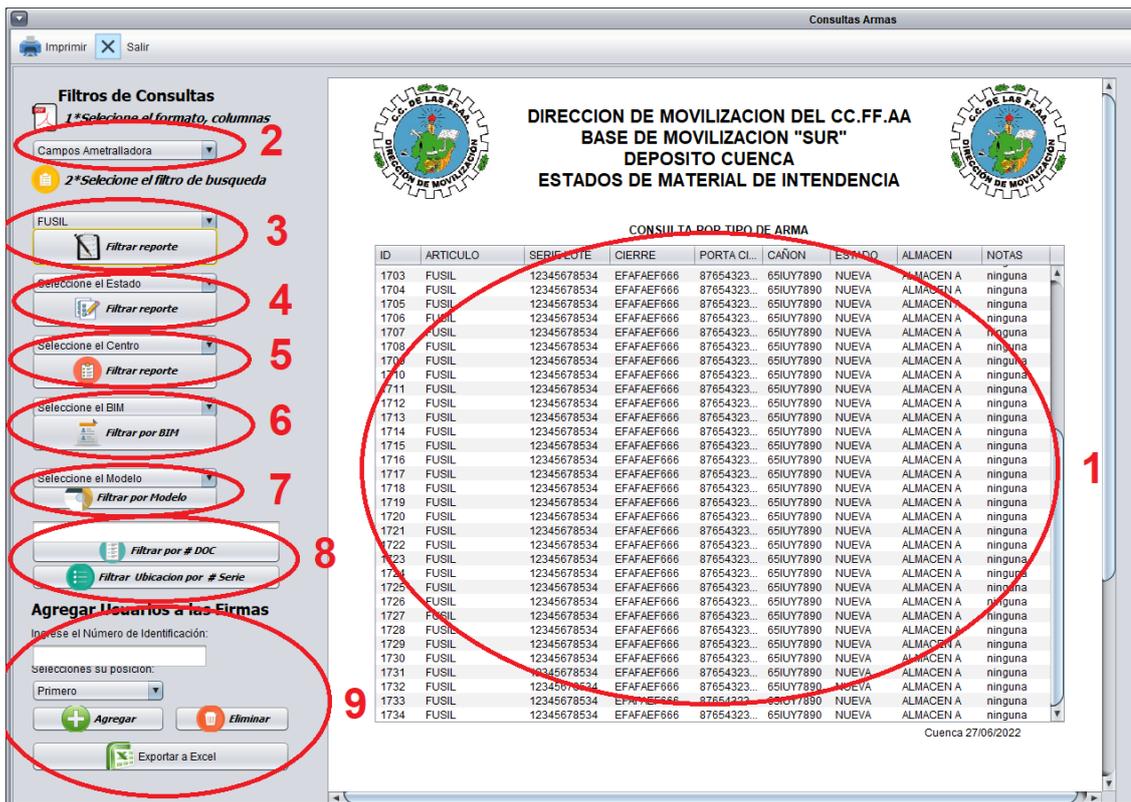


Ilustración 29. Ventana consultas material bélico sistema. Fuente: Autor

## Pantalla consulta intendencia

Para acceder a esta pantalla seleccionamos el menú principal “Servicios” y el sub menú “Consultas” y luego en el submenú “Intendencia”.



Ilustración 30. Menú consulta intendencia sistema. Fuente: Autor

1. Se visualizan los registros de material de intendencia.
2. El primer combo permite seleccionar los filtros de búsqueda de acuerdo al estado de los elementos.
3. El segundo combo permite seleccionar la Base-Centro como filtro de búsqueda.

4. El cuarto combo permite filtrar los registros por Bim.
5. También se implementa un campo para ingresar el nombre del material para la búsqueda.
6. Se filtra por el nombre de la ubicación.
7. Se exportan los registros a Excel para su posterior análisis y manejo.
8. Se ingresan usuarios al pie de página del informe.
9. Por último, se puede imprimir el reporte.

Ilustración 31. Ventana consulta intendencia sistema. Fuente: Autor

### Pantalla monitoreo movimientos arma e intendencia

Para acceder a esta pantalla se selecciona el menú principal “Servicios” y el sub menú “Monitoreo movimientos” y luego “Arma” o “Intendencia”, las funcionalidades detalladas a continuación son las mismas en todos los aspectos tanto para el monitoreo de material bélico como intendencia.



Ilustración 32. Menú monitoreo material bélico sistema. Fuente: Autor

1. Lista de registros de movimientos de material.
2. Se puede imprimir el contenido de la tabla principal.
3. Se filtran los registros de acuerdo al parámetro tipo de movimiento.
4. Se filtran los registros de acuerdo al parámetro centro.
5. También implementa una funcionalidad para filtrar registros de acuerdo a un intervalo de tiempo.
6. Se filtran los registros de acuerdo a su fecha de devolución.
7. Se puede visualizar el documento de oficio adjunto al movimiento de material.
8. Se puede visualizar el documento de memorándum adjunto al movimiento de material.
9. Para cada movimiento se asigna una lista de arma que por armonizar la visualización se podrán listar con esta opción en una nueva ventana flotante y no en la tabla principal.
10. Se puede exportar el contenido de la tabla en formato Excel para su posterior manejo.



Ilustración 33. Ventana monitoreo material bélico sistema. Fuente: Autor

### Pantalla movimiento egreso material bélico e intendencia

Para acceder a esta pantalla se selecciona el menú principal “Servicios” y el sub menú “Movimientos egreso” y luego “Movimientos egreso armas” o “Movimiento egreso intendencia”, las funcionalidades detalladas a continuación son las mismas en todos los aspectos tanto para el movimiento de material bélico como intendencia buscando simplificar la explicación y facilitar su manejo.



Ilustración 34. Menú movimiento egreso material bélico sistema. Fuente: Autor

1. Como primer aspecto importante se comprueba el privilegio del usuario y su Base-Centro para otorgar o no la opción generar un movimiento, una vez realizada esta



- comprobación, se debe seleccionar el tipo de movimiento, en este apartado existen dos opciones egreso definitivo y préstamo.
2. Seguidamente se debe seleccionar el centro al cual se realizará el movimiento en caso de ser préstamo.
  3. Se selecciona una fecha por defecto en la cual se registra el movimiento.
  4. Y si se realizara un movimiento de egreso definitivo no es necesario seleccionar una fecha de devolución, ya que esta estará solo disponible para movimientos de préstamo.
  5. Se debe subir el documento de memorándum correspondiente.
  6. Se debe subir el documento de Oficio de acuerdo al proceso interno de la institución.
  7. El estado activo por defecto al momento de la creación y unas notas por si se necesita adjuntar más información.
  8. En esta tabla se reflejarán los elementos seleccionados y que constarán en el registro de movimiento.
  9. En esta tabla se puede seleccionar los elementos a ser agregados al movimiento
  10. En este campo se debe ingresar la cantidad de cada elemento a ser agregado al movimiento, cumpliendo con los parámetros respectivos de no sobrepasar la cantidad existente.
  11. Si por error humano se agregaron elementos equivocados existe la opción eliminar que quita el elemento de la tabla y por ende del movimiento.
  12. Para la búsqueda de un registro en específico se deberá ingresar el id o número de serie del elemento tomando en cuenta varios parámetros se listará o no el registro, ya que dependerá de la Base-Centro al que pertenezca y de su estado.
  13. Por último, se realiza la verificación de datos y se procede a guardar el movimiento no sin antes solicitar obligatoriamente generar un acta de movimiento.

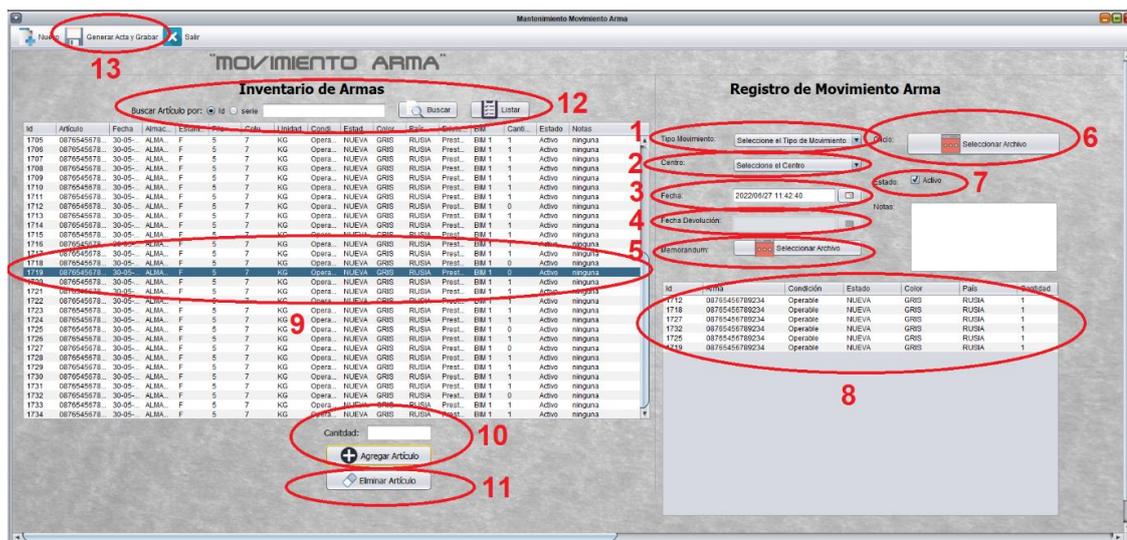


Ilustración 35. Ventana movimiento egreso material bélico sistema. Fuente: Autor

1. En el primer apartado se procederá a ingresar el número de cédula del primer involucrado en el proceso de movimiento de material conjuntamente con la posición dentro del documento y clic en agregar, automáticamente se asignan sus datos de nombres, cargo, título de firma a los campos correspondientes o si se desea eliminar se sigue el mismo proceso esta vez dando clic en eliminar.
2. Estos son los campos que se imprimirán y guardarán en el acta de movimiento con los usuarios y datos correspondientes agregados anteriormente.
3. Se debe llenar los campos generales como título, cuerpo del documento, y subtítulo para constancia del proceso.
4. Para comprobación se visualizará una tabla con todos los elementos incluidos en el movimiento correspondiente.
5. Por último, se da clic en guardar, se verificarán los datos ingresados y se generara el registro del acta.

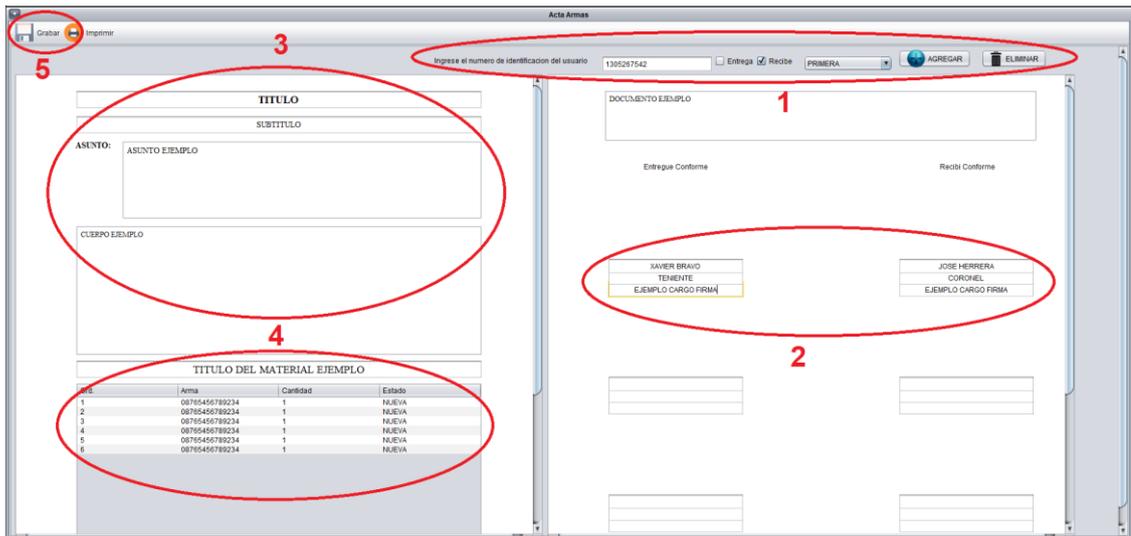


Ilustración 36. Ventana acta material bélico sistema. Fuente: Autor

### **Pantalla movimiento ingreso material bélico e intendencia.**

Para acceder a esta pantalla se selecciona el menú principal “Servicios” y el sub menú “Movimientos ingreso” y luego “Creación acta de ingreso armas” o “Creación acta de ingreso intendencia”, las funcionalidades detalladas a continuación son las mismas en todos los aspectos tanto para el movimiento de material bélico como intendencia buscando simplificar la explicación y facilitar su manejo.

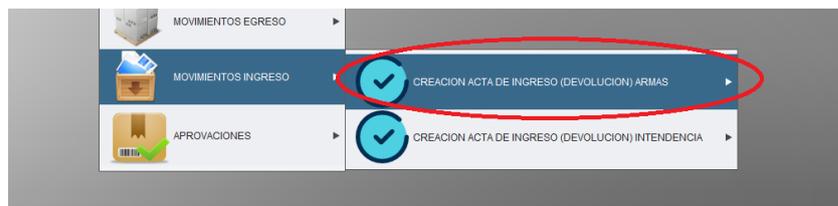


Ilustración 37. Menú movimiento ingreso material bélico sistema. Fuente: Autor

1. En este apartado se lista la información de registro del acta correspondiente, conjuntamente con los elementos del movimiento.
2. Permite seleccionar un movimiento y generar un acta de retorno de los elementos correspondientes solo de los registros que estén pendientes.
3. Para una mejor comprobación se deberá visualizar los documentos pdf verificando así si toda la información esta correcta.

4. El siguiente paso de cumplir con los parámetros necesarios se realizará el registro de la información.
5. De ser necesario se puede imprimir el Acta correspondiente.

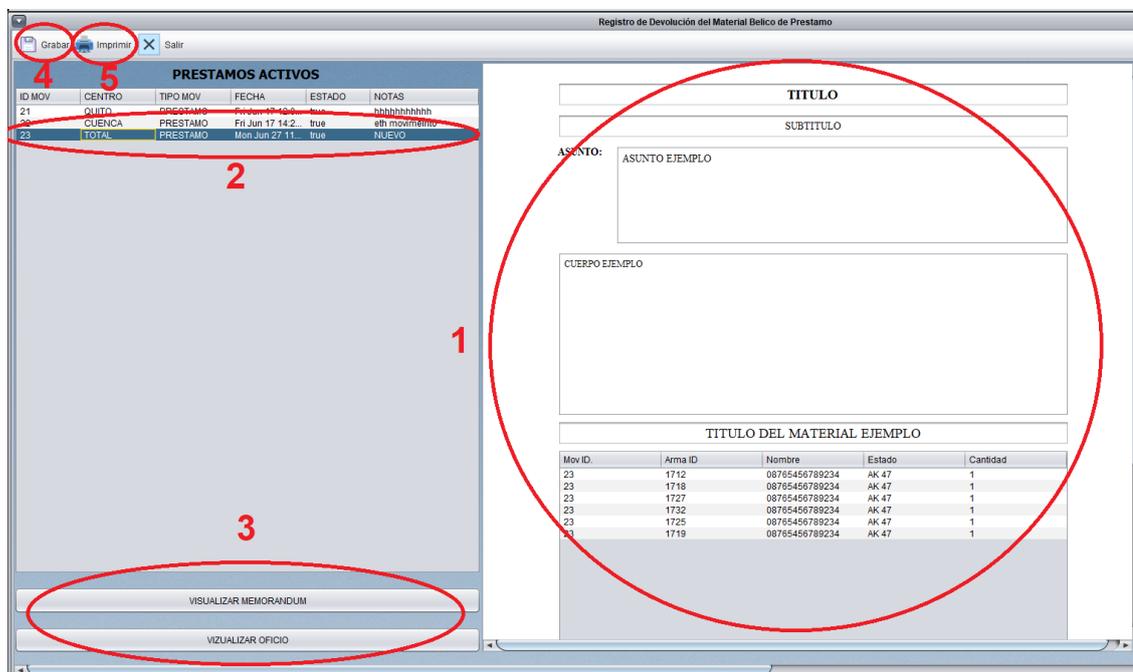


Ilustración 38. Ventana movimiento ingreso material bélico sistema. Fuente: Autor

### Pantalla aprobación egreso material bélico e intendencia.

Para acceder a esta pantalla se selecciona el menú principal “Aprobaciones” y el sub menú “Aprobación armas” y luego “Aprobaciones egreso definitivo”, las funcionalidades detalladas a continuación son las mismas en todos los aspectos tanto para aprobaciones de movimientos de material bélico como intendencia buscando simplificar la explicación y facilitar su manejo.



Ilustración 39. Menú aprobación egreso definitivo material bélico sistema. Fuente: Autor

En esta ventana constan las firmas de los usuarios que hayan aprobado el acta de egreso definitivo, conjuntamente con los elementos del movimiento y por último la opción de aprobar.

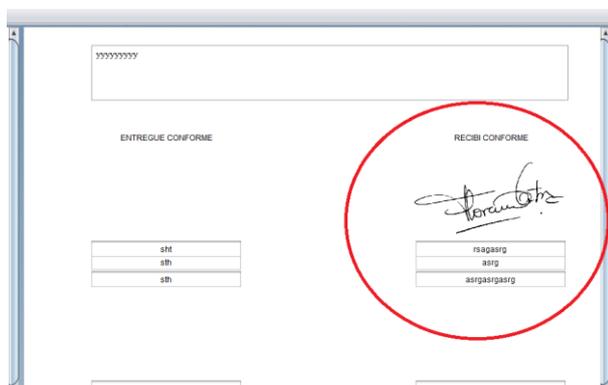


Ilustración 40. Ventana aprobación egreso definitivo material bélico sistema. Fuente: Autor

### **Pantalla aprobación préstamo devolución egreso material bélico e intendencia.**

Para acceder a esta pantalla se selecciona el menú principal “Aprobaciones” y el sub menú “Aprobación armas” y luego “Aprobaciones prestamos devolución”, las funcionalidades detalladas a continuación son las mismas en todos los aspectos tanto para aprobaciones de movimientos de material bélico como intendencia buscando simplificar la explicación y facilitar su manejo.



Ilustración 41. Menú aprobación egreso definitivo material bélico sistema. Fuente: Autor

1. Esta ventana emergente se presentan tres opciones la primera permite acceder al registro de movimiento de egreso par posteriormente aprobar de igual forma la segunda opción permite acceder al registro de movimiento de retorno o ingreso en caso de no existir aún no se realizará ninguna acción.

2. Aquí esta todos los registros de movimientos que incluyen al usuario y los filtros correspondientes de Base-Centro
3. Una vez seleccionado el movimiento correspondiente se activará la opción aprobar y se verificará todo esto en el apartado de acta donde ya constará la firma en el lado derecho, en caso de ser un acta de ingreso solo cambiará la posición de las firmas.

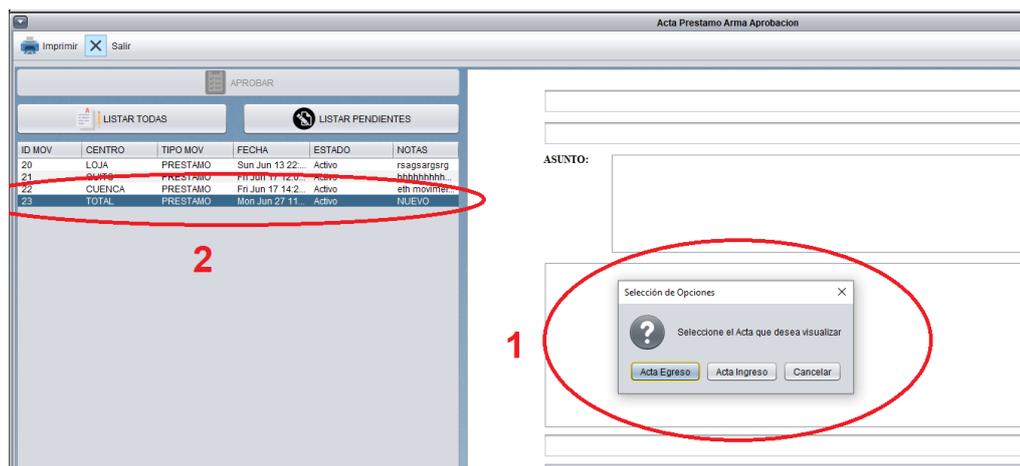


Ilustración 42. Ventana aprobación acta egreso definitivo material bélico sistema. Fuente: Autor

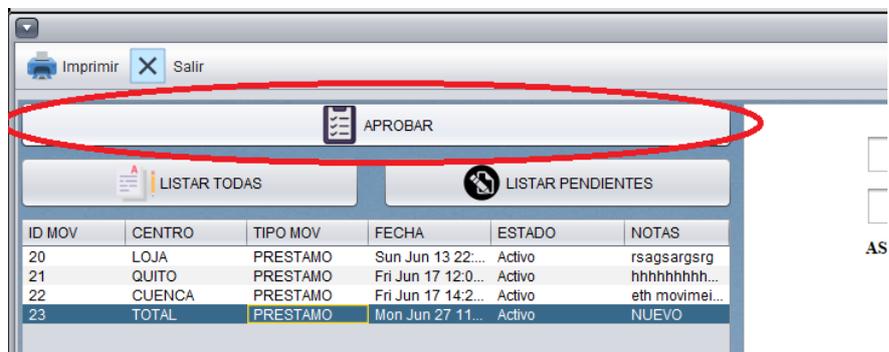


Ilustración 43. Ventana aprobación acta egreso movimiento material bélico sistema. Fuente: Autor



Ilustración 44. Ventana aprobación firma egreso material bélico sistema. Fuente: Autor

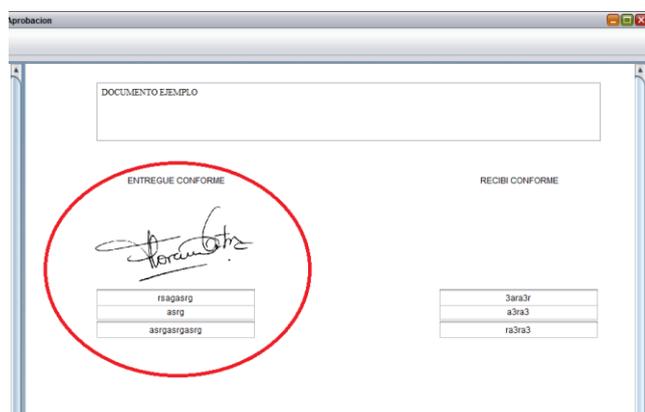


Ilustración 45. Ventana aprobar firma movimiento ingreso material bélico sistema. Fuente: Autor

### **Pantallas tablas auxiliares.**

Para acceder a estas pantallas se selecciona el menú principal “Campos arma” o “Campos intendencia” o “Campos usuario” dentro de las cuales se listarán varios submenús que poseen la misma estructura para el manejo y almacenamiento de información. Las funcionalidades detalladas a continuación son las mismas en todos los aspectos buscando simplificar la explicación y facilitar su manejo se detallan las opciones principales a continuación.

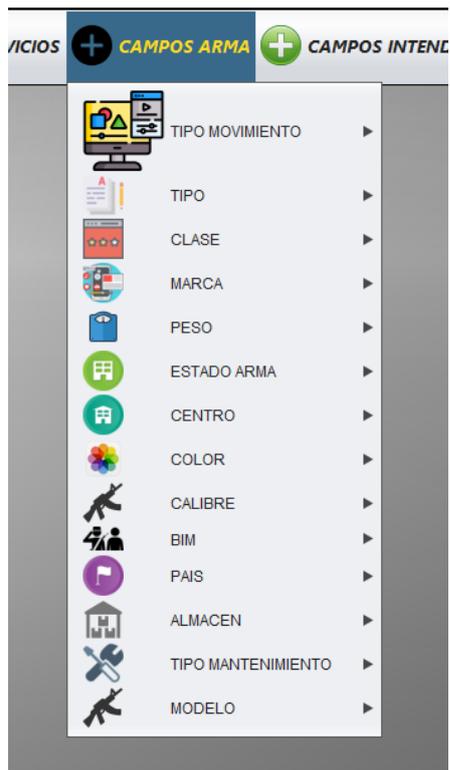


Ilustración 46. Menú campos material bélico sistema. Fuente: Autor



Ilustración 47. Menú campos material intendencia sistema. Fuente: Autor



*Ilustración 48. Menú campos material usuario sistema. Fuente: Autor*

1. Tabla donde se visualizan los registros correspondientes, al momento de seleccionar uno de los elementos se activa el modo “actualizar” y se colocan los datos en los campos de la izquierda.
2. Campo para almacenar notas de información extra sobre el registro.
3. Esta opción de estado permite añadir o excluir un registro del combo correspondiente.
4. Este campo almacena el nombre que aparecerá como opción en el combo al que se asignará.
5. Se puede buscar un registro ingresando su id.
6. Este botón permite aplicar la búsqueda de registros en base a al filtro de id ingresado.
7. Limpia los campos y activa el modo guardar.
8. Graba o actualiza un registro con la información de los campos, validando la información ingresada.
9. Activa el panel de búsqueda.
10. Elimina un registro en base al id ingresado.

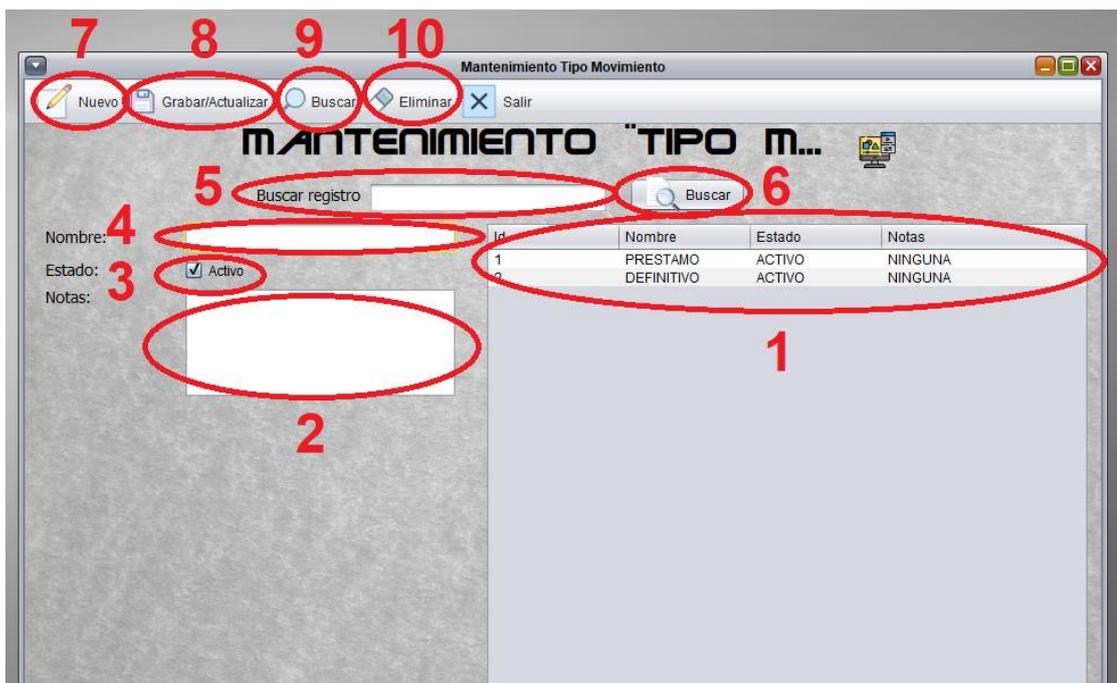


Ilustración 49. Ventana tabla auxiliar genérica sistema. Fuente: Autor

## Capítulo III

### Manual del programador

#### Programación

Para el desarrollo del presente proyecto se dividió el trabajo en 3 partes para mejorar el marco metodológico y obtener un mejor producto final.

Creación de la base de datos, programación del API\_RESTfull, desarrollo de la aplicación de escritorio, y en cuanto a factores de funcionalidad, incluye aspectos como:

- **Corrección:** El sistema desarrollado satisface todos los requerimientos determinados por el cliente.
- **Usabilidad:** En cuestión de manejo es sencillo de aprender.
- **Integridad:** El sistema cumple con la calidad al no poseer ningún tipo de efectos secundarios.
- **Fiabilidad:** El sistema no tiene ningún defecto, y tampoco falla mientras está en ejecución.
- **Eficiencia:** Forma en que el sistema maneja los recursos disponibles. El software hace uso eficiente del espacio de almacenamiento y sus funciones se ejecutan según dentro de tiempo deseados.
- **Seguridad:** Medidas adecuadas para preservar los datos ante amenazas externas.

## **Análisis de requerimientos**

En este apartado se detallan todos los requisitos que cubre el sistema, se buscó un diseño independiente y escalable, especialmente que soporte la totalidad de funcionalidades requeridas.

### **Requisitos funcionales**

- R1. Gestión de usuarios (guardar, eliminar, buscar, actualizar)
- R2. Gestión de Bases- Centros (guardar, eliminar, buscar, actualizar).
- R3. Gestión de material bélico y de intendencia y sus movimientos (guardar, eliminar, buscar, actualizar) gestión de material asignado, y los movimientos que van surgiendo conjuntamente con la verificación de datos ingresados, listar campos detallados, generar solicitudes de actualización.
- R4. El modelo ha de permitir gestionar solicitudes de actualización de registros (solicitud, aprobación, rechazo, y notificación) listar solicitudes pendientes y visualizar usuarios involucrados.
- R5. Control de inventario.
- R6. Procesos que permitan extraer reportes de monitoreo y consultas.
- R7. Filtrar reportes que incluyan el ingreso de rangos de fechas, lista total de elementos prestados, dados de baja, según la base-centro.
- R8. Reporte de elementos en base a su estado, base-centro, bim, modelo, número de documento, número de serie, tipo de material.
- R9. Registro de mantenimientos sobre material bélico, modificar, registrar quien realizo el mantenimiento y quien realiza actualizaciones.
- R10. Administrar documentación externa en formato pdf que se adjuntara con cada registro de movimientos, conjuntamente con la generación de actas donde consten los usuarios involucrados, cuerpo del documento y materiales incluidos.

- R11. El sistema debe permitir generar un movimiento de ingreso con la generación del acta de devolución al seleccionar un registro de salida.
- R12. Validación y asignación de privilegios a los usuarios.
- R13. Por último, existirán notificaciones de procesos pendientes o en etapa de retraso de devolución.

### **Requisitos no funcionales**

- R14. Disposición de mantenimiento. Se realizará el desarrollo en base estándares de nomenclatura, generando así un esquema comprensible.
- R15. La base de datos deberá ser escalable para que en el transcurso del tiempo se pueda ir cubriendo progresivamente todas las necesidades que se generen durante su vigencia.
- R16. Pruebas de funcionamiento para garantizar la correcta ejecución de las funcionalidades implementadas, conjuntamente con el control de errores y anomalías informáticas.

### **Base de datos**

Para el óptimo desempeño de la base de datos, así como prevenir redundancia en la información, se generó el modelo en base a las condiciones de diseño adecuadas, acorde a las formas de normalización.

El desarrollo de la base de datos fue realizado en SQL Server 2019, para la cual se implementa un guía de base de datos relacional en un modelo copo de nieve, consta de 40 tablas tomando en consideración los siguientes parámetros de creación:

- Mantener la precisión e integridad de todos los datos de la organización trabajando así coordinadamente para cumplir con sus metas.
- Rapidez, al manejar estructuras sistemáticas y correctamente diseñadas.



- Menor redundancia
- Seguridad y comprobación de errores
- Exclusión de inconsistencias
- Integridad: validación y la persistencia de los registros almacenados.
- Seguridad: resguardo de la base de datos ante usuarios sin autorización.
- Concurrencia: Ante la existencia de varios usuarios que pueden acceder simultáneamente a un mismo registro.
- Recuperación: Este sistema establece controles para preservar los datos frente a la presencia de fallos en el sistema.



## Tablas

**DETALLEMANTENIMIENTO:** Esta tabla intermedia almacena los datos de un arma y un mantenimiento, debido a que un mantenimiento puede tener varias armas y un arma puede pertenecer a varios mantenimientos.

DETALLEMANTENIMIENTO			
	Column Name	Data Type	Allow Nulls
🔑	ARMAID	CANTIDAD:int	<input type="checkbox"/>
🔑	MANTENIMIENTOID	CANTIDAD:int	<input type="checkbox"/>
	NOTAS	TEXTOCORTO:varch...	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

*Ilustración 51. Tabla detalle mantenimiento base de datos. Fuente: Autor.*

**MANTENIMIENTO:** Aquí se almacenan los registros de mantenimientos, que vienen a ser datos sobre la evaluación del estado del material correspondiente y de alguna modificación nueva.

MANTENIMIENTO			
	Column Name	Data Type	Allow Nulls
🔑	MANTENIMIENTOID	CANTIDAD:int	<input type="checkbox"/>
	TIPOMANTENIMIENTOID	CANTIDAD:int	<input checked="" type="checkbox"/>
	MANTENIMIENTOFECHA	FECHA:datetime	<input checked="" type="checkbox"/>
	MANTENIMIENTOESTADO	BOOLEAN:bit	<input checked="" type="checkbox"/>
	MANTENIMIENTONOTAS	varchar(8000)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

*Ilustración 52. Tabla mantenimiento base de datos. Fuente: Autor.*

**DETALLE\_MI\_USUARIO:** En esta tabla se almacenan los registros de movimientos de material de inventario y los usuarios que intervienen en este proceso, ya que en un movimiento intervienen varios encargados y un encargado puede pertenecer a varios movimientos de material de intendencia.

DETALLE MI USUARIO			
	Column Name	Data Type	Allow Nulls
🔑	MOVIMIENTOIID	CANTIDAD:int	<input type="checkbox"/>
🔑	USUARIONUMIDENTIFIC...	ID:varchar(30)	<input type="checkbox"/>
	USUARIOACCION	TEXTOCORTO:varch...	<input checked="" type="checkbox"/>
	NOTAS	TEXTOCORTO:varch...	<input checked="" type="checkbox"/>
	USUARIOACCIONDOS	TEXTOCORTO:varch...	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Ilustración 53. Tabla detalle mantenimiento usuario base de datos. Fuente: Autor.

**ACTAINTENDENCIA:** Contiene la información sobre un acta que se genera al momento de realizar un movimiento de material.

ACTAINTENDENCIA			
	Column Name	Data Type	Allow Nulls
🔑	ACTAID	CANTIDAD:int	<input type="checkbox"/>
	MOVIMIENTOIID	CANTIDAD:int	<input checked="" type="checkbox"/>
	ACTATITULO	TEXTOCORTO:varch...	<input checked="" type="checkbox"/>
	ACTALUGAR	TEXTOLARGO:varch...	<input checked="" type="checkbox"/>
	ACTAASUNTO	TEXTOSUPERLARGO...	<input checked="" type="checkbox"/>
	ACTACUERPO	TEXTOSUPERLARGO...	<input checked="" type="checkbox"/>
	ACTATITULODOS	TEXTOCORTO:varch...	<input checked="" type="checkbox"/>
	ACTANOTAS	TEXTOSUPERLARGO...	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Ilustración 54. Tabla acta intendencia base de datos. Fuente: Autor.

**ARMA:** Contiene la información sobre el material bélico de todo tipo teniendo en consideración para el inventario, y englobando elementos dados de baja pero que se mantienen en el sistema.

Column Name	Data Type	Allow Nulls
ARMAID	CANTIDAD:int	<input type="checkbox"/>
ARMAFECHA	FECHA:datetime	<input checked="" type="checkbox"/>
ALMACENID	CANTIDAD:int	<input checked="" type="checkbox"/>
ARMAESTANTERIA	ID:varchar(30)	<input checked="" type="checkbox"/>
ARMAFILA	ID:varchar(30)	<input checked="" type="checkbox"/>
ARMACOLUMNA	ID:varchar(30)	<input checked="" type="checkbox"/>
ARMANRODOCUMENTO	ID:varchar(30)	<input checked="" type="checkbox"/>
ARMARAZONSOCIAL	TEXTOCORTO:varcha...	<input checked="" type="checkbox"/>
CLASEID	CANTIDAD:int	<input checked="" type="checkbox"/>
TIPOID	CANTIDAD:int	<input checked="" type="checkbox"/>
MARCAID	CANTIDAD:int	<input checked="" type="checkbox"/>
ARMAMODELO	CANTIDAD:int	<input checked="" type="checkbox"/>
CALIBREID	CANTIDAD:int	<input checked="" type="checkbox"/>
ARMASERIELOTE	ID:varchar(30)	<input checked="" type="checkbox"/>
ARMACIERRE	ID:varchar(30)	<input checked="" type="checkbox"/>
ARMAPORTACIERRE	ID:varchar(30)	<input checked="" type="checkbox"/>
ARMABAYONETA	ID:varchar(30)	<input checked="" type="checkbox"/>
ARMACANON	ID:varchar(30)	<input checked="" type="checkbox"/>
ARMACANTIDAD	CANTIDAD:int	<input checked="" type="checkbox"/>
ARMAPESO	CANTIDADDECIMAL...	<input checked="" type="checkbox"/>
UNIDADMEDIDAID	CANTIDAD:int	<input checked="" type="checkbox"/>
BIMID	CANTIDAD:int	<input checked="" type="checkbox"/>
ESTADOID	CANTIDAD:int	<input checked="" type="checkbox"/>
COLORID	CANTIDAD:int	<input checked="" type="checkbox"/>
PAISID	CANTIDAD:int	<input checked="" type="checkbox"/>
CENTROID	CANTIDAD:int	<input checked="" type="checkbox"/>
ARMAFOTO	varchar(MAX)	<input type="checkbox"/>
ARMAESTADO	BOOLEAN:bit	<input checked="" type="checkbox"/>

Ilustración 55. Tabla arma base de datos. Fuente: Autor.

**INTENDENCIA:** De la misma forma en esta tabla se almacena información sobre el material de intendencia de todo tipo teniendo en consideración para el inventario, y englobando elementos dados de baja pero que se mantienen en el sistema.

INTENDENCIA			
	Column Name	Data Type	Allow Nulls
	INTENDENCIAID	CANTIDAD:int	<input type="checkbox"/>
	INTENDENCIAFECHA	FECHA:datetime	<input checked="" type="checkbox"/>
	CENTROID	CANTIDAD:int	<input checked="" type="checkbox"/>
	ALMACENID	CANTIDAD:int	<input checked="" type="checkbox"/>
	INTENDENCIAESTANTERIA	ID:varchar(30)	<input checked="" type="checkbox"/>
	INTENDENCIAFILO	ID:varchar(30)	<input checked="" type="checkbox"/>
	INTENDENCIACOLUMNNA	ID:varchar(30)	<input checked="" type="checkbox"/>
	UNIDADID	CANTIDAD:int	<input checked="" type="checkbox"/>
	INTENDENCIAARTICULO	TEXTOCORTO:varch...	<input checked="" type="checkbox"/>
	INTENDENCIACONDICION	BOOLEAN:bit	<input checked="" type="checkbox"/>
	ESTADOID	CANTIDAD:int	<input checked="" type="checkbox"/>
	COLORID	CANTIDAD:int	<input checked="" type="checkbox"/>
	PAISID	CANTIDAD:int	<input checked="" type="checkbox"/>
	INTENDENCIAEXISTENCIA	BOOLEAN:bit	<input checked="" type="checkbox"/>
	INTENDENCIANOVEDAD	TEXTOLARGO:varch...	<input checked="" type="checkbox"/>
	INTENDENCIAFOTO	varchar(MAX)	<input type="checkbox"/>
	BIMID	CANTIDAD:int	<input checked="" type="checkbox"/>
	INTENDENCIACANTIDAD	CANTIDAD:int	<input checked="" type="checkbox"/>
	INTENDENCIAESTADO	BOOLEAN:bit	<input checked="" type="checkbox"/>
	INTENDENCIANOTAS	TEXTOLARGO:varch...	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Ilustración 56. Tabla intendencia base de datos. Fuente: Autor.

DETALLEMOVIMIENTOARMA: En esta tabla se guardan los datos del material bélico, y del movimiento, esta es una tabla intermedia debido a que un movimiento tiene varias armas y un arma puede pertenecer a varios movimientos.

DETALLEMOVIMIENTOARMA			
	Column Name	Data Type	Allow Nulls
	ARMAID	CANTIDAD:int	<input type="checkbox"/>
	MOVIMIENTOID	CANTIDAD:int	<input type="checkbox"/>
	CANTIDADARMA	CANTIDAD:int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Ilustración 57. Tabla detalle movimiento arma base de datos. Fuente: Autor.

MOVIMIENTOINTENDENCIA: En esta tabla se guardan los datos del movimiento de material de intendencia.

MOVIMIENTOINTENDENCIA			
	Column Name	Data Type	Allow Nulls
🔑	MOVIMIENTOID	CANTIDAD:int	<input type="checkbox"/>
	CENTROID	CANTIDAD:int	<input checked="" type="checkbox"/>
	TIPOMOVIMIENTOID	CANTIDAD:int	<input checked="" type="checkbox"/>
	MOVIMIENTOSALIDA	CANTIDAD:int	<input checked="" type="checkbox"/>
	MOVIMIENTOIFECHA	FECHA:datetime	<input checked="" type="checkbox"/>
	MOVIMIENTOIMEMORA...	varchar(MAX)	<input checked="" type="checkbox"/>
	MOVIMIENTOOFICIO	varchar(MAX)	<input checked="" type="checkbox"/>
	MOVIMIENTOTIEMPO	FECHA:datetime	<input checked="" type="checkbox"/>
	MOVIMIENTOESTADO	BOOLEAN:bit	<input checked="" type="checkbox"/>
	MOVIMIENTONOTAS	TEXTOSUPERLARGO...	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Ilustración 58. Tabla movimiento intendencia base de datos. Fuente: Autor.

**DETALLEHISTORIALINTENDENCIA:** Aquí se guardan los datos de un usuario y un historial de intendencia que consta dentro de una solicitud de actualización de campos de un registro, es una tabla intermedia debido a que puede contener varios usuarios y un usuario estar contenido en varios historiales.

DETALLEHISTORIALINTENDENCIA			
	Column Name	Data Type	Allow Nulls
🔑	USUARIONUMIDENTIFICA...	ID:varchar(30)	<input type="checkbox"/>
🔑	HISTORIALID	CANTIDAD:int	<input type="checkbox"/>
	USERERROR	BOOLEAN:bit	<input checked="" type="checkbox"/>
	NOTAS	TEXTOCORTO:varchar...	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Ilustración 59. Tabla detalle historial intendencia base de datos. Fuente: Autor.

**HISTORIALINTENDENCIA:** En esta tabla se guardan los datos principales sobre las solicitudes de actualización de campos de material de intendencia.

HISTORIALINTENDENCIA			
	Column Name	Data Type	Allow Nulls
	HISTORIALID	CANTIDAD:int	<input type="checkbox"/>
	INTENDENCIAID	CANTIDAD:int	<input checked="" type="checkbox"/>
	HISTORIALFECHA	FECHA:datetime	<input checked="" type="checkbox"/>
	HISTORIALCAMBIOS	TEXTOLARGO:varcha...	<input checked="" type="checkbox"/>
	HISTORIALDOCUMENTO	varchar(MAX)	<input checked="" type="checkbox"/>
	HISTORIALESTADO	BOOLEAN:bit	<input checked="" type="checkbox"/>
	HISTORIALNOTAS	TEXTOCORTO:varchar...	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Ilustración 60. Tabla historial intendencia base de datos. Fuente: Autor.

DETALLEMOVIMIENTOINTENDENCIA: En esta tabla se guardan los datos del material de intendencia, y del movimiento, esta es una tabla intermedia debido a que un movimiento tiene varias armas y un arma puede pertenecer a varios movimientos.

DETALLEMOVIMIENTOINTENDENCIA			
	Column Name	Data Type	Allow Nulls
	MOVIMIENTOIID	CANTIDAD:int	<input type="checkbox"/>
	INTENDENCIAID	CANTIDAD:int	<input type="checkbox"/>
	CANTIDADINTENDENCIA	CANTIDAD:int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Ilustración 61. Tabla detalle movimiento intendencia base de datos. Fuente: Autor.

ACTA: Contiene los datos del acta que surge debido al movimiento de material es el encabezado puede contener mucha información sobre detalles generados por el personal interno.

ACTA			
	Column Name	Data Type	Allow Nulls
	ACTAID	CANTIDAD:int	<input type="checkbox"/>
	MOVIMIENTOID	CANTIDAD:int	<input checked="" type="checkbox"/>
	ACTATITULO	TEXTOCORTO:varchar...	<input checked="" type="checkbox"/>
	ACTALUGAR	TEXTOLARGO:varcha...	<input checked="" type="checkbox"/>
	ACTAASUNTO	TEXTOSUPERLARGO:...	<input checked="" type="checkbox"/>
	ACTACUERPO	TEXTOSUPERLARGO:...	<input checked="" type="checkbox"/>
	ACTATITULODOS	TEXTOCORTO:varchar...	<input checked="" type="checkbox"/>
	ACTANOTAS	TEXTOSUPERLARGO:...	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Ilustración 62. Tabla acta base de datos. Fuente: Autor.

USUARIO: Contiene los datos de un usuario y englobando elementos dados de baja pero que se mantienen en el sistema.

USUARIO			
	Column Name	Data Type	Allow Nulls
🔑	USUARIONUMIDENTIFICA...	ID:varchar(30)	<input type="checkbox"/>
	IDENTIFICACIONID	CANTIDAD:int	<input checked="" type="checkbox"/>
	SEXOID	CANTIDAD:int	<input checked="" type="checkbox"/>
	CIUDADID	CANTIDAD:int	<input checked="" type="checkbox"/>
	CENTROID	CANTIDAD:int	<input checked="" type="checkbox"/>
	TIPOUSUARIOID	CANTIDAD:int	<input checked="" type="checkbox"/>
	JERARQUIAID	CANTIDAD:int	<input checked="" type="checkbox"/>
	USUARIO NOMBRES	TEXTOCORTO:varchar...	<input checked="" type="checkbox"/>
	USUARIO APELLIDOS	TEXTOCORTO:varchar...	<input checked="" type="checkbox"/>
	USUARIO DIRECCION	TEXTOCORTO:varchar...	<input checked="" type="checkbox"/>
	USUARIO TELEFONO	TEXTOCORTO:varchar...	<input checked="" type="checkbox"/>
	USUARIO CORREO	TEXTOCORTO:varchar...	<input checked="" type="checkbox"/>
	USUARIO FECHANAC	FECHA:datetime	<input checked="" type="checkbox"/>
	USUARIO USERNAME	TEXTOCORTO:varchar...	<input checked="" type="checkbox"/>
	USUARIO PASSWORD	ID:varchar(30)	<input checked="" type="checkbox"/>
	USUARIO FIRMA	varchar(MAX)	<input checked="" type="checkbox"/>
	USUARIO FIRMA NOMBRE	TEXTOCORTO:varchar...	<input checked="" type="checkbox"/>
	USUARIO FIRMA CARGO	TEXTOCORTO:varchar...	<input checked="" type="checkbox"/>
	USUARIO FIRMA JERARQUIA	TEXTOCORTO:varchar...	<input checked="" type="checkbox"/>
	USUARIO ESTADO	BOOLEAN:bit	<input checked="" type="checkbox"/>
	USUARIO NOTAS	TEXTOCORTO:varchar...	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Ilustración 63. Tabla usuario base de datos. Fuente: Autor.

DETALLE\_MA\_USUARIO: Esta tabla almacena a un usuario y el id del movimiento de material bélico, es una tabla intermedia debido a que un usuario pertenece a varios movimientos y un movimiento tiene varios usuarios.

DETALLE_MA_USUARIO			
	Column Name	Data Type	Allow Nulls
🔑	USUARIONUMIDENTIFICA...	ID:varchar(30)	<input type="checkbox"/>
🔑	MOVIMIENTOID	CANTIDAD:int	<input type="checkbox"/>
	USUARIO ACCION	TEXTOCORTO:varchar...	<input checked="" type="checkbox"/>
	NOTAS	TEXTOCORTO:varchar...	<input checked="" type="checkbox"/>
	USUARIO ACCIONDOS	TEXTOCORTO:varchar...	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Ilustración 64. Tabla detalle movimiento usuario base de datos. Fuente: Autor.

MOVIMIENTOARMAS: Aquí se guardan los datos sobre el movimiento de material bélico es la tabla principal, y los datos de registros inactivos permanecen guardados.

MOVIMIENTOARMAS			
	Column Name	Data Type	Allow Nulls
🔑	MOVIMIENTOID	CANTIDAD:int	<input type="checkbox"/>
	CENTROID	CANTIDAD:int	<input checked="" type="checkbox"/>
	TIPOMOVIMIENTOID	CANTIDAD:int	<input checked="" type="checkbox"/>
	MOVIENTOSALIDA	CANTIDAD:int	<input checked="" type="checkbox"/>
	MOVIMIENTOFECHA	FECHA:datetime	<input checked="" type="checkbox"/>
	MOVIMIENTOEMORAND...	varchar(MAX)	<input checked="" type="checkbox"/>
	MOVIMIENTOOFICIO	varchar(MAX)	<input checked="" type="checkbox"/>
	MOVIMIENTOTIEMPO	FECHA:datetime	<input checked="" type="checkbox"/>
	MOVIMIENTOESTADO	BOOLEAN:bit	<input checked="" type="checkbox"/>
	MOVIMIENTONOTAS	TEXTOSUPERLARGO:...	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Ilustración 65. Tabla movimiento material bélico base de datos. Fuente: Autor.

**HISTORIAL:** Esta tabla contiene los datos sobre una solicitud de actualización de material bélico, subyacente a un registro generado de forma errónea.

HISTORIAL			
	Column Name	Data Type	Allow Nulls
🔑	HISTORIALID	CANTIDAD:int	<input type="checkbox"/>
	ARMAID	CANTIDAD:int	<input checked="" type="checkbox"/>
	HISTORIALFECHA	FECHA:datetime	<input checked="" type="checkbox"/>
	HISTORIALCAMBIOS	TEXTOLARGO:varcha...	<input checked="" type="checkbox"/>
	HISTORIALDOCUMENTO	varchar(MAX)	<input checked="" type="checkbox"/>
	HISTORIALESTADO	BOOLEAN:bit	<input checked="" type="checkbox"/>
	HISTORIALNOTAS	TEXTOCORTO:varchar...	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Ilustración 66. Tabla historial base de datos. Fuente: Autor.

**DETALLEHISTORIALARMA:** Esta tabla intermedia contiene un usuario y un historial para el registro de usuarios que solicitan o aprueban una solicitud de actualización de campos de material bélico.

DETALLE HISTORIA ALARMA			
	Column Name	Data Type	Allow Nulls
🔑	USUARIONUMIDENTIFICA...	ID:varchar(30)	<input type="checkbox"/>
🔑	HISTORIALID	CANTIDAD:int	<input type="checkbox"/>
	USERERROR	BOOLEAN:bit	<input checked="" type="checkbox"/>
	NOTAS	TEXTOCORTO:varchar...	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Ilustración 67. Tabla detalle historial material bélico base de datos. Fuente: Autor.

**TIPOUSUARIO:** Contiene los datos para el acceso a cada pantalla del sistema de esta dependerán los privilegios de cada usuario, teniendo la posibilidad de generar varios tipos combinando el acceso a cada función.

TIPOUSUARIO			
	Column Name	Data Type	Allow Nulls
	TIPOUSUARIONOMBRE	TEXTOCORTO:varc...	<input checked="" type="checkbox"/>
	TIPOUSUARIOESTADO	BOOLEAN:bit	<input checked="" type="checkbox"/>
	TIPOUSUARIONOTAS	TEXTOCORTO:varc...	<input checked="" type="checkbox"/>
	ESTADOINTENDENCIA	TEXTOCORTO:varc...	<input checked="" type="checkbox"/>
	UNIDADINTENDENCIA	TEXTOCORTO:varc...	<input checked="" type="checkbox"/>
	ALMACENINTENDENCIA	TEXTOCORTO:varc...	<input checked="" type="checkbox"/>
	CIUDADUSUARIO	TEXTOCORTO:varc...	<input checked="" type="checkbox"/>
	TIPOUSUARIO	TEXTOCORTO:varc...	<input checked="" type="checkbox"/>
	SEXOUSUARIO	TEXTOCORTO:varc...	<input checked="" type="checkbox"/>
	IDENTIFICACIONUSUARIO	TEXTOCORTO:varc...	<input checked="" type="checkbox"/>
	JERARQUIAUSUARIO	TEXTOCORTO:varc...	<input checked="" type="checkbox"/>
	TIPOARMA	TEXTOCORTO:varc...	<input checked="" type="checkbox"/>
	CLASEARMA	TEXTOCORTO:varc...	<input checked="" type="checkbox"/>
	MARCAARMA	TEXTOCORTO:varc...	<input checked="" type="checkbox"/>
	PESOARMA	TEXTOCORTO:varc...	<input checked="" type="checkbox"/>
	ESTADOARMA	TEXTOCORTO:varc...	<input checked="" type="checkbox"/>
	CENTROARMA	TEXTOCORTO:varc...	<input checked="" type="checkbox"/>
	COLORARMA	TEXTOCORTO:varc...	<input checked="" type="checkbox"/>
	CALIBREARMA	TEXTOCORTO:varc...	<input checked="" type="checkbox"/>
	PAISARMA	TEXTOCORTO:varc...	<input checked="" type="checkbox"/>
	ALMACENARMA	TEXTOCORTO:varc...	<input checked="" type="checkbox"/>
	TIPOMANTENIMIENTO	TEXTOCORTO:varc...	<input checked="" type="checkbox"/>
	BIM	TEXTOCORTO:varc...	<input checked="" type="checkbox"/>
	MODELO	TEXTOCORTO:varc...	<input checked="" type="checkbox"/>
	REGISTROUSUARIO	TEXTOCORTO:varc...	<input checked="" type="checkbox"/>
	CONSULTAS	TEXTOCORTO:varc...	<input checked="" type="checkbox"/>
	MONITOREO	TEXTOCORTO:varc...	<input checked="" type="checkbox"/>
	MOVIMIENTOARMA	TEXTOCORTO:varc...	<input checked="" type="checkbox"/>
	MOVIMIENTOINTENDE...	TEXTOCORTO:varc...	<input checked="" type="checkbox"/>
	TIPOMOVIMIENTO	TEXTOCORTO:varc...	<input checked="" type="checkbox"/>
	REGISTROARMA	TEXTOCORTO:varc...	<input checked="" type="checkbox"/>
	REGISTROINTENDENCIA	TEXTOCORTO:varc...	<input checked="" type="checkbox"/>
	MANTENIMIENTOARMA	TEXTOCORTO:varc...	<input checked="" type="checkbox"/>
	APROBARSAARMA	BOOLEAN:bit	<input checked="" type="checkbox"/>
	APROBARSAINTENDEN...	BOOLEAN:bit	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Ilustración 68. Tabla tipo usuario base de datos. Fuente: Autor.

## Desarrollo API\_RESTfull

Para cubrir el proceso de desarrollo del proyecto se utilizó una conexión a base de datos SQL Server desde Netbeans, la cual permite obtener el acceso a las tablas para poder realizar un API\_RESTfull con el API Java JAX-RS.

A continuación, se detallarán los diferentes pasos para el proceso de creación.

1. Como primer paso se crea una nueva conexión.

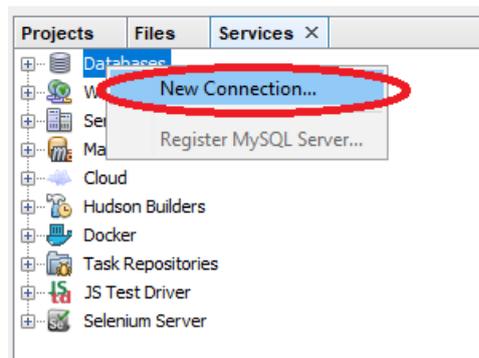


Ilustración 69. Conexión base de datos Netbeans. Fuente: Autor.

2. El segundo paso es seleccionar el conector jdbc correspondiente a la versión de SQL Server.

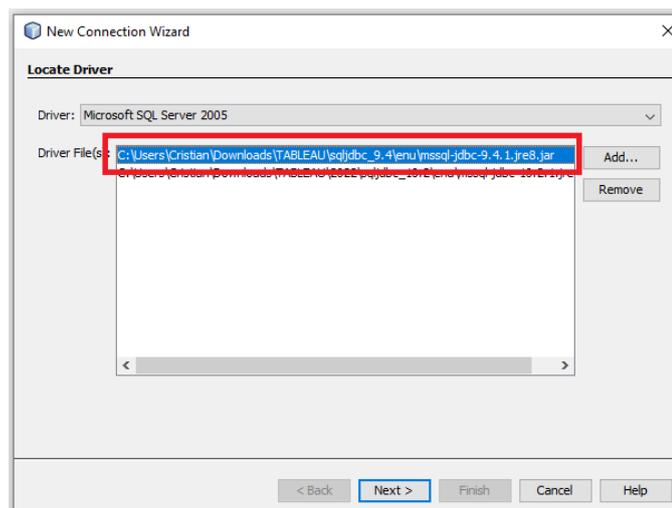


Ilustración 70. Conector jdbc sql server Netbeans. Fuente: Autor.

3. En el tercer paso se ingresan los datos necesarios para crear la URL de conexión con las credenciales de ingreso generadas al instalar la base de datos.

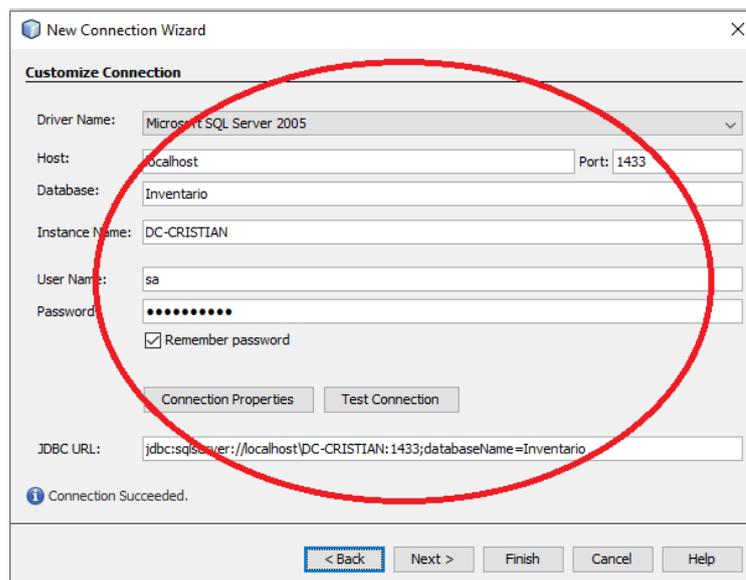


Ilustración 71. Datos de conexión sql server Netbeans. Fuente: Autor.

4. El siguiente paso es generar un aplicativo web.

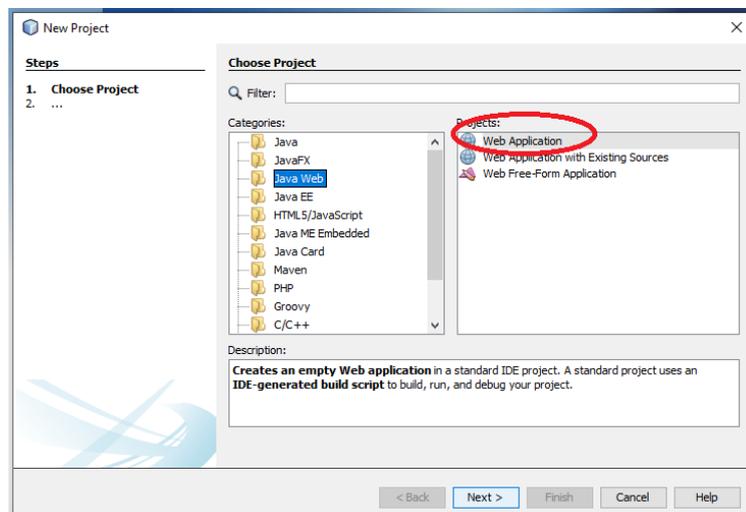


Ilustración 72. Aplicación web Netbeans. Fuente: Autor.

5. En el nuevo proyecto generado se selecciona “RESTfull Web service from database” que permitirá elegir la conexión anteriormente creada para acceder a la base de datos.

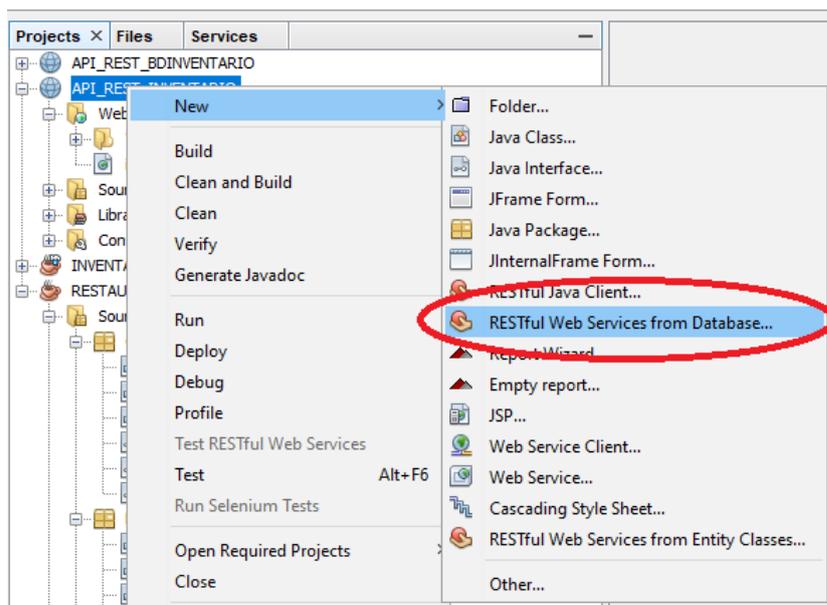


Ilustración 73. RESTful Web Services Netbeans. Fuente: Autor.

6. En este apartado permite seleccionar las tablas de las cuales se desea generar el RESTfull.

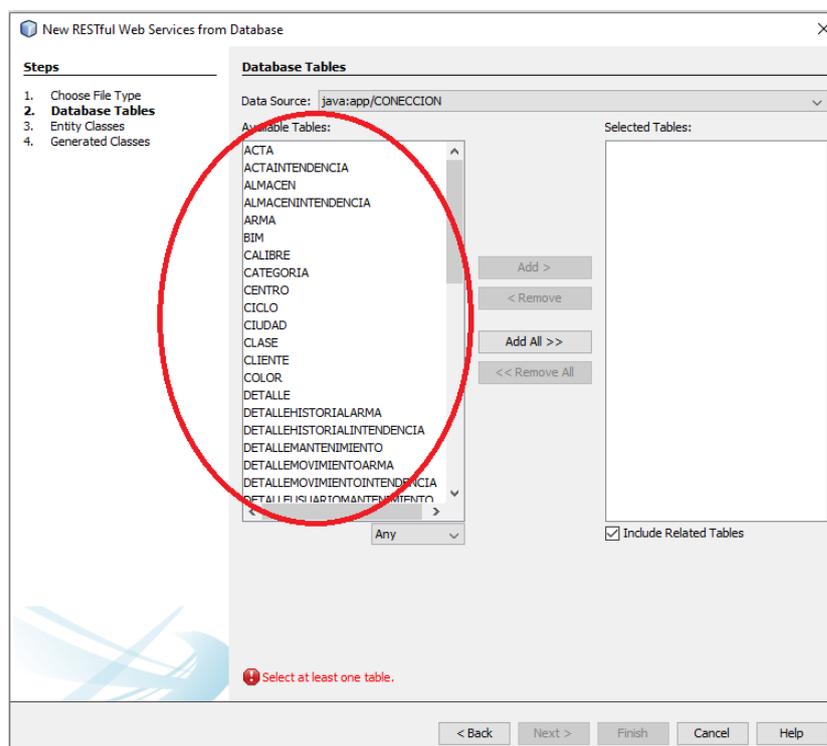


Ilustración 74. Tablas sql Client Netbeans. Fuente: Autor.

- Una vez se generen los archivos correspondientes en sus paquetes y dentro del proyecto se limpia y construye el proyecto generando un archivo war.

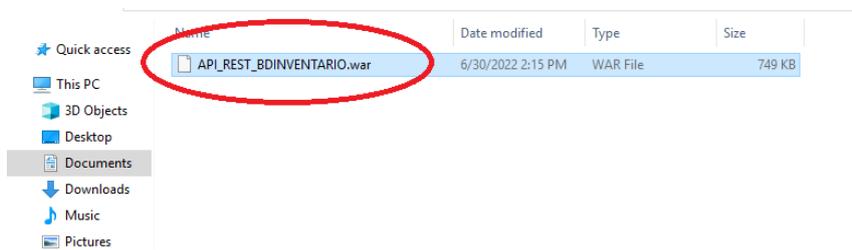


Ilustración 75. Archivo .war RESTful web service. Fuente: Autor.

- Se ejecuta por consola los comando para iniciar el servidor, ejecutar el dominio establecido, y desplegar el API\_RESTfull, comandos que intervienen “asadmin start-domain domain3” y “asadmin deploy API\_REST\_BDINVENTARIO.war”.

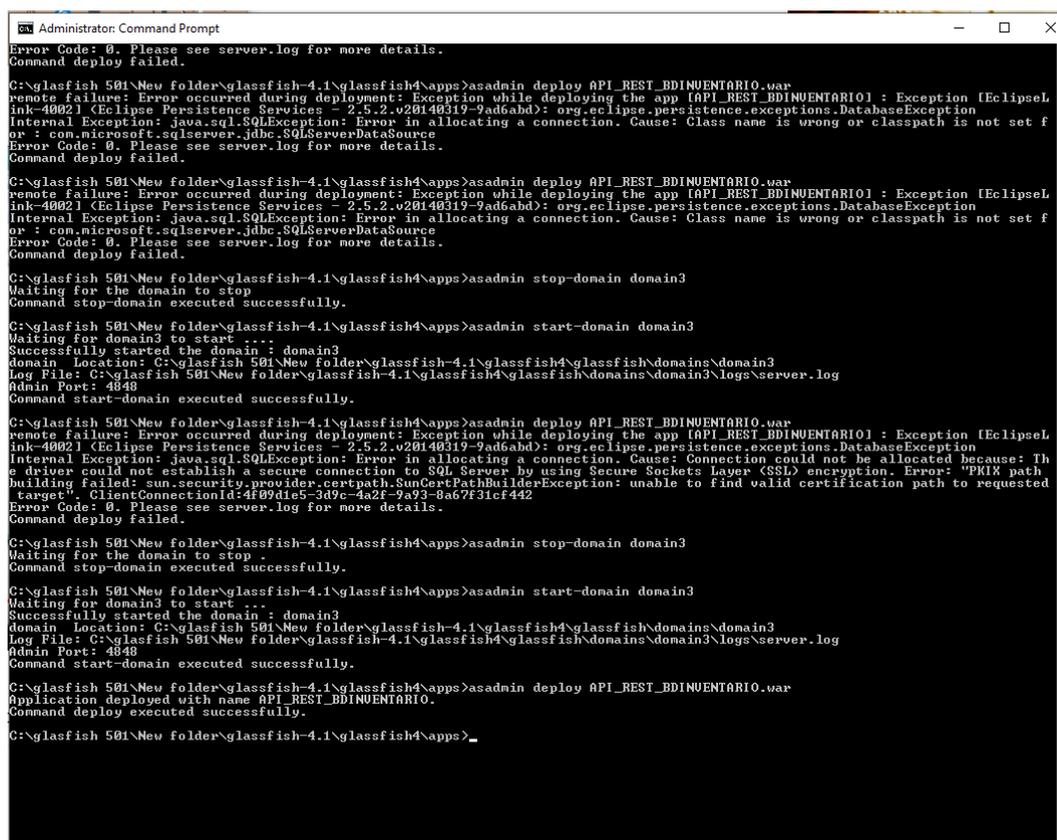


Ilustración 76. Configuración Servidor Glassfish. Fuente: Autor.

- Una vez se sube al servidor en este caso Glassfish se realiza la comprobación, abriendo la consola y verificando que se visualice en aplicaciones

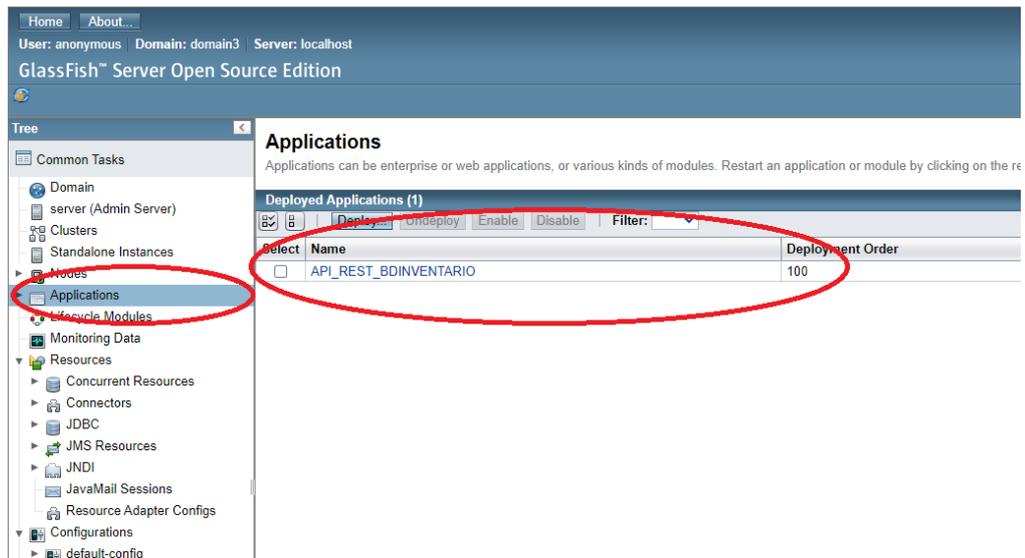


Ilustración 77. Server Admin Glassfish. Fuente: Autor.

## Aplicación de escritorio

1. Se genera una aplicación Java

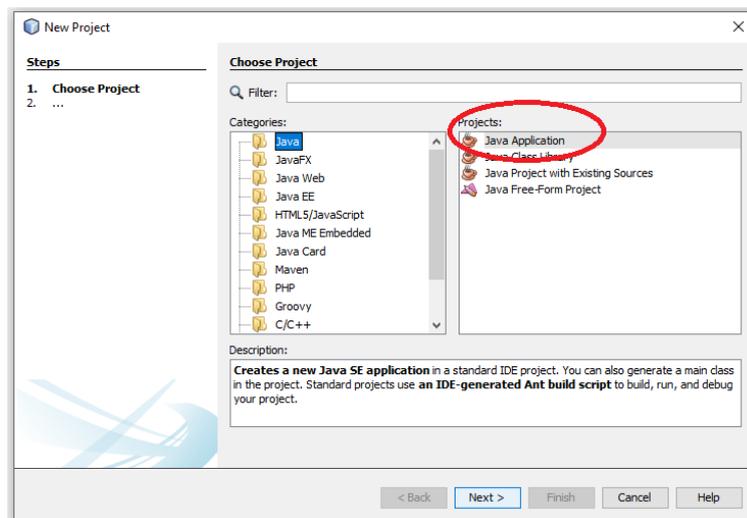


Ilustración 78. Aplicación JAVA Netbeans. Fuente: Autor.

2. Una vez se genere el proyecto se agrega el paquete “Entity” al proyecto nuevo que contiene la capa de modelo con las entidades correspondientes a cada tabla de la base de datos.

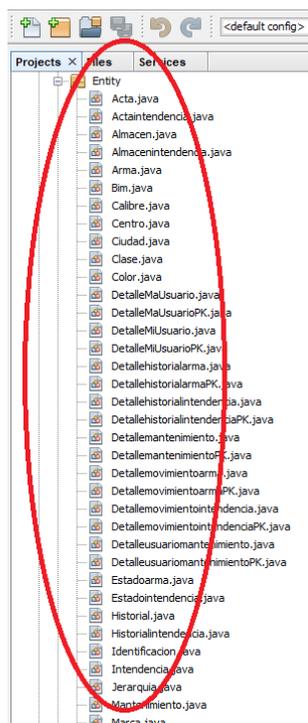


Ilustración 79. Capa de datos Netbeans. Fuente: Autor.

3. La capa de controlador se representa con “CRUD” y se genera cada RESTfull Client individualmente para cada entidad, los mismos que contienen los procesos créate, read, update y delete, todos estos procesos corresponden al manejo de datos JSON que son recuperados desde el API\_RESTfull y los presenta como datos manejables en lenguaje Java.

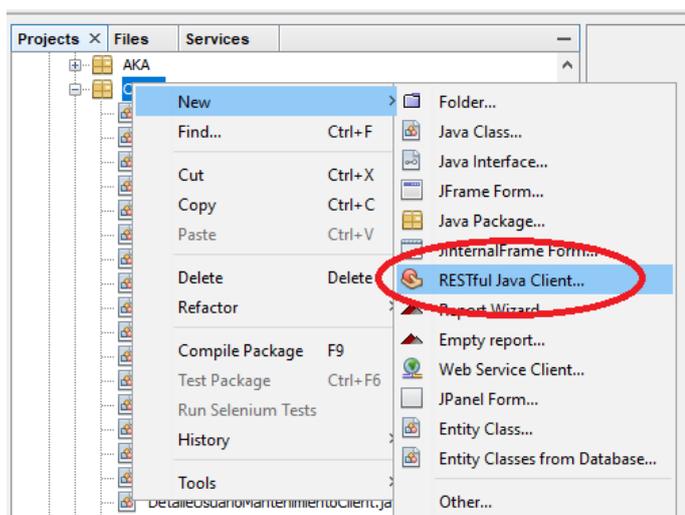
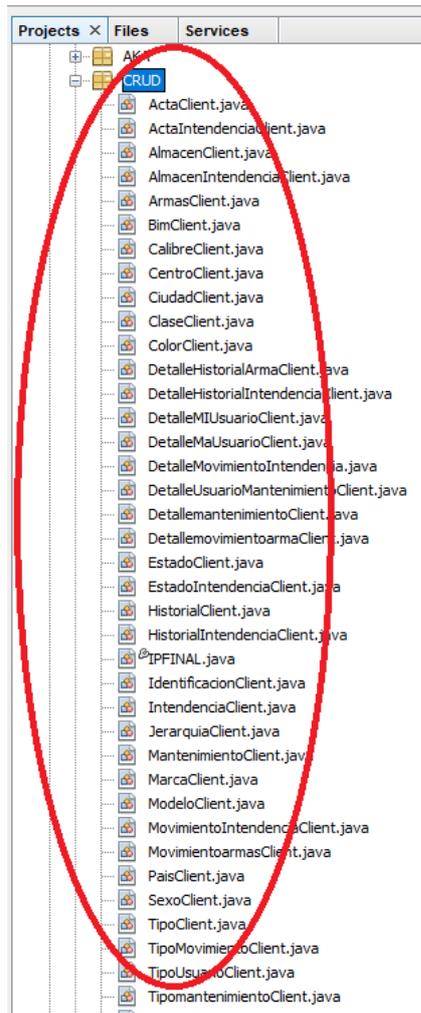


Ilustración 80. RESTfull JAVA Client Netbeans. Fuente: Autor.



*Ilustración 81. Capa de negocios Netbeans. Fuente: Autor.*

4. Por último, la capa de presentación incluye todas las interfaces graficas que permiten interactuar al usuario con el sistema.

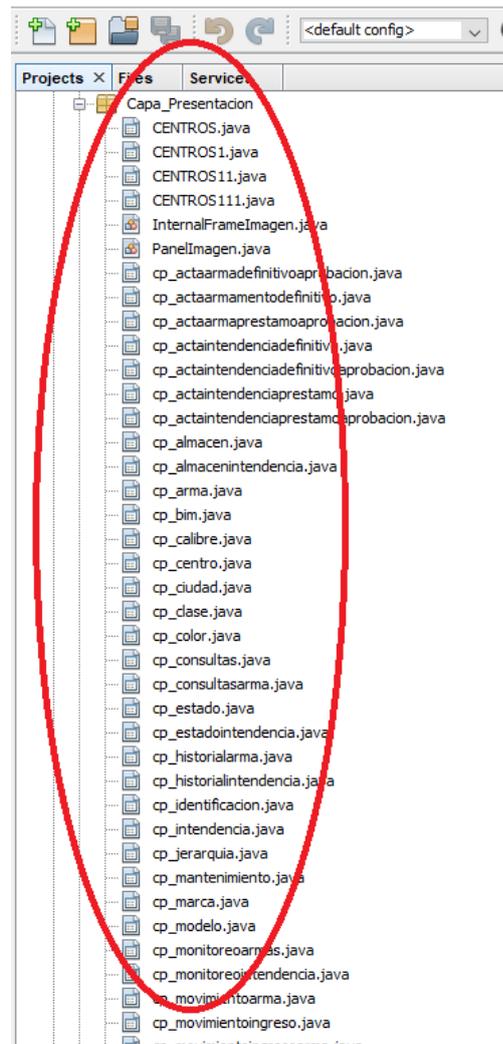


Ilustración 82. Capa de presentación Netbeans. Fuente: Autor.

### Librerías y superclases capa de presentación

En el proceso de desarrollo del sistema para implementar funcionalidades acordes a los objetivos planteados se utilizaron diferentes librerías que se listan a continuación:

- **BorderLayout:** Esta superclase permite crear un esquema completo para integrar los componentes de una interfaz de acuerdo a los componentes de paneles.
- **Color:** Se basa en generar un modelo de color RGB, conformado principalmente por la mezcla de los tres colores principales, su creación es mediante números comprendidos entre 0 y 255.



- **Graphics:** Esta superclase permite la creación de varias figuras geométricas, con un entorno de trabajo basado en AWT, incluyendo mostrar texto y muchas opciones de diseño más.
- **Image:** Esta superclase abstracta implementa imágenes gráficas, que se obtiene de manera específica de la plataforma.
- **Toolkit:** Es una superclase de tipo abstracto, todas las herramientas o implementaciones del Abstract Window Toolkit. Como las subclasses de la clase Toolkit se utilizan para generar los diversos componentes e implementaciones específicas de ferrents internos, muchos eventos de GUI se pueden integrar en el formulario de usuario, incluso si se indica claramente lo contrario. Además, muchas operaciones de GUI se pueden ejecutar en forma formateada. Este hecho significa que se define el estado de un componente e inmediatamente considera el estado, el valor del reclamo puede no reflejar una alternativa solicitada.
- **BufferedImage:** Se describe un Image con un búfer accesible de datos de imagen. A BufferedImage se compone de un ColorModel y un Raster de datos de imagen. El número y tipos de bandas en SampleModel del Raster debe coincidir con el número y tipos requeridos por ColorModel para representar su color y componentes alfa. Todos los BufferedImage objetos tienen una coordenada en la esquina superior izquierda de (0, 0). Por lo tanto, cualquiera Raster que se use para construir a BufferedImage debe tener  $\text{minX}=0$  y  $\text{minY}=0$ . Esta clase se basa en los métodos de obtención y configuración de datos de Raster, y en los métodos de caracterización del color de ColorModel.
- **FileWriter:** Esta clase se utiliza principalmente para escribir archivos de caracteres, utilizando constructores predefinidos y definiendo un tamaño de búfer determinado en bytes. Para determinar y definir estos valores se creará un OutputStreamWriter



en `FileOutputStream`. La disponibilidad de archivos o la capacidad de crear archivos depende de la plataforma subyacente. En particular, algunas plataformas le permiten abrir un archivo para escribir con solo un `FileWriter` (u otro objeto que escribe archivos) a la vez. En tales casos, los diseñadores de esta clase tendrán error si el archivo en se encuentra abierto. `FileWrite` está creado para generar una sucesión de caracteres.

- **InputStream:** Esta es una superclase de tipo abstracto de la mayoría de clases que representan un flujo de entrada de bytes. El software que necesita determinar una subclase de `InputStream` persistentemente deben suministrar un método que devuelva el siguiente byte de entrada.
- **InetAddress:** Esta clase hace referencia a una dirección de Protocolo de Internet (IP). Una dirección IP es un número sin firmar de 32 o 128 bits utilizado por IP, un protocolo de nivel inferior sobre el que se construyen protocolos como UDP y TCP. La arquitectura de direcciones IP está definida por RFC 790: Números asignados , RFC 1918: Asignación de direcciones para Internet privadas , RFC 2365: Multidifusión de IP con alcance administrativo y RFC 2373: Arquitectura de direccionamiento IP versión 6 . Una instancia de `InetAddress` consiste en una dirección IP y posiblemente su nombre de host correspondiente (dependiendo de si está construido con un detalle de host o si se realiza una determinación de nombre de host inversa).
- **NetworkInterface:** Esta clase hace referencia a una interfaz de red conformada por un nombre y varias direcciones IP establecidas para esta interfaz. Se utiliza para identificar la interfaz local en la que se une a un grupo de multidifusión. Las interfaces normalmente se conocen con nombres como "le0".



- **ImageIO:** Una clase que contiene métodos de conveniencia estáticos para ubicar directorios con archivos ImageReadery ImageWritercorreos, y realizar una codificación y decodificación simple.
- **JDesktopPane:** Un contenedor utilizado para crear una interfaz de varios documentos o un escritorio virtual. Usted crea JInternalFrameobjetos y los agrega al archivo JDesktopPane. JDesktopPanese extiende JLayeredPanepara administrar los marcos internos potencialmente superpuestos. También mantiene una referencia a una instancia DesktopManagerestablecida por la clase de interfaz de usuario para el aspecto actual (L&F). Tenga en cuenta que JDesktopPane no admite bordes.
- **JDialog:** La clase principal para crear una ventana de diálogo. Puede usar esta clase para crear un cuadro de diálogo personalizado o invocar los muchos métodos de clase JOptionPanepara crear una variedad de cuadros de diálogo estándar. El JDialogcomponente contiene a JRootPane como su único hijo. El contentPanedebe ser el padre de cualquier hijo del JDialog. Como conveniencia addy sus variantes, removey setLayoutse han anulado para reenviar a los contentPaneque sean necesarios.
- **JFrame:** Como todos los demás contenedores de nivel superior de JFC/Swing, a JFramecontiene a JRootPanecomo único elemento secundario. El panel de contenido proporcionado por el panel raíz debe, como regla, contener todos los componentes que no son de menú que muestra el JFrame. Esto es diferente del Framecaso AWT.  
Como conveniencia addy sus variantes, removey setLayoutse han anulado para reenviar a la contentPanese según sea necesario.
- **JOptionPane:** facilita la aparición de un cuadro de diálogo genérico donde se pide al usuario datos de entrada o simplemente les muestra un mensaje.



- **Printable:** La Printable interfaz se implementa mediante los print métodos del pintor de páginas actual, al que llama el sistema de impresión para representar una página. Al crear un Pageable, se utilizan pares de PageFormat instancias e instancias que implementan esta interfaz para describir cada página. Se llama a la instancia que implementa Printable para imprimir los gráficos de la página.

- **ByteArrayInputStream:** Contiene un búfer interno que contiene bytes que se pueden leer de la secuencia. Un contador interno realiza un seguimiento del siguiente byte que suministrará el read método.

Cerrar un ByteArrayInputStream no tiene ningún efecto. Los métodos de esta clase se pueden llamar después de que se haya cerrado la secuencia sin generar una IOException.

- **File:** Una representación abstracta de nombres de rutas de archivos y directorios. Las interfaces de usuario y los sistemas operativos utilizan *cadena*s de nombres de rutas dependientes del sistema para nombrar archivos y directorios. Esta clase presenta una vista abstracta e independiente del sistema de los nombres de ruta jerárquicos.

- **Base64:** Esta clase se fundamenta únicamente en métodos estáticos para conseguir codificadores y decodificadores con la finalidad de generar una codificación Base64. La implementación de esta clase admite los siguientes tipos de Base64 como se detalla en RFC 4648 y RFC 2045 .

- **Poi:** Es una librería en el paquete HMEF de poi-scratchpad (Apache POI) permite el manejo de archivos creación, lectura, escritura, modificación. Este paquete se utiliza para leer archivos TNEF (Microsoft Outlook y Microsoft Office). En nuestro proyecto se utiliza para generar un archivo XML con datos obtenidos de una tabla de reporte, si una aplicación usa poi-scratchpad para analizar archivos TNEF y la

aplicación permite que los usuarios que no son de confianza los proporcionen, entonces un archivo cuidadosamente diseñado puede causar una excepción de falta de memoria.

```
import java.awt.BorderLayout;
import java.awt.Component;
import java.awt.Desktop;
import java.awt.Dialog;
import java.awt.Dimension;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.Image;
import java.awt.Toolkit;
import java.awt.Window;
import java.awt.image.BufferedImage;
import java.awt.print.PageFormat;
import java.awt.print.Printable;
import static java.awt.print.Printable.NO_SUCH_PAGE;
import static java.awt.print.Printable.PAGE_EXISTS;
import java.awt.print.PrinterException;
import java.awt.print.PrinterJob;
import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.math.BigDecimal;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.sql.Timestamp;
import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Base64;
import java.util.Collection;
import java.util.Date;
import java.util.LinkedList;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.imageio.ImageIO;
import javax.swing.AbstractButton;
import javax.swing.DefaultComboBoxModel;
import javax.swing.Icon;
```

*Ilustración 83. Librerías pantalla administración material bélico sistema. Fuente: Autor.*

```
import com.toedter.calendar.JDateChooser;
import java.awt.BorderLayout;
import java.awt.Desktop;
import java.awt.Dialog;
import java.awt.Graphics;
import java.awt.Image;
import java.awt.Toolkit;
import java.awt.Window;
import java.io.ByteArrayInputStream;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import java.util.Date;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Base64;
import java.util.List;
import javax.swing.AbstractButton;
import javax.swing.ImageIcon;
import javax.swing.JDialog;
import javax.swing.JFileChooser;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;
import javax.swing.SwingUtilities;
import javax.swing.SwingWorker;
import javax.swing.filechooser.FileNameExtensionFilter;
```

Ilustración 84. Librerías mantenimiento intendencia sistema. Fuente: Autor.

```
import CRUD.ArmasClient;
import CRUD.BimClient;
import CRUD.CentroClient;
import CRUD.EstadoClient;
import CRUD.EstadoIntendenciaClient;
import CRUD.IntendenciaClient;
import CRUD.ModeloClient;
import CRUD.TipoClient;
import CRUD.UsuarioClient;
import Entity.Usuario;
import java.awt.Color;
import java.awt.Desktop;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.Image;
import java.awt.Toolkit;
import java.awt.print.PageFormat;
import java.awt.print.Printable;
import static java.awt.print.Printable.NO_SUCH_PAGE;
import static java.awt.print.Printable.PAGE_EXISTS;
import java.awt.print.PrinterException;
import java.awt.print.PrinterJob;
import java.io.BufferedOutputStream;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.BorderFactory;
import javax.swing.Icon;
import javax.swing.ImageIcon;
import javax.swing.JFileChooser;
import javax.swing.JOptionPane;
import javax.swing.JTable;
import javax.swing.JTextField;
import javax.swing.filechooser.FileNameExtensionFilter;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableColumnModel;
import org.apache.poi.hssf.usermodel.HSSFWorkbook;
import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.ss.usermodel.Row;
import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.ss.usermodel.Workbook;
import org.apache.poi.xssf.usermodel.XSSFWorkbook;
```

Ilustración 85. Librerías consulta material bélico sistema. Fuente: Autor.

## Código capa de presentación

Para la primera pantalla del sistema se utilizaron varias instancias de todos los JFrame correspondientes a cada pantalla que al momento de dar clic en una opción del menú superior principal creara una instancia y la mostrara.

1. Al iniciar el programa esta pantalla se ajustará a la resolución del monitor del equipo que la esté ejecutando y posicionará los componentes en el centro.
2. En la segunda parte verificara si el computador que está corriendo el programa está autorizado verificando si su dirección MAC consta en el sistema.

```

public cp_principal() {
    initComponents();
    ocultar();
    creditos1.setHorizontalAlignment(JLabel.CENTER);
    creditos2.setHorizontalAlignment(JLabel.CENTER);

    int ancho = java.awt.Toolkit.getDefaultToolkit().getScreenSize().width;
    int alto = java.awt.Toolkit.getDefaultToolkit().getScreenSize().height;
    this.setBounds((ancho / 2) - (this.getWidth() / 2), (alto / 2) - (this.getHeight() / 2), 500, 500);

    txtcontraseña.setText("jared123");
    txtusername.setText("jared");
    StringBuilder sb = new StringBuilder();
    NetworkInterface a;
    String linea;
    try {
        a = NetworkInterface.getByInetAddress(InetAddress.getLocalHost());
        byte[] mac = a.getHardwareAddress();

        for (int i = 0; i < mac.length; i++) {
            sb.append(String.format("%02X%s", mac[i], (i < mac.length - 1) ? "-" : ""));
        }
        FileWriter fwriter = new FileWriter("mac.dat");
        fwriter.write("MAC: " + sb.toString());
        fwriter.close();

        // lmac.setText("SE ha registrado la MAC exitosamente.");
    } catch (Exception e) {
        e.printStackTrace();
    }

    String prue = sb.toString();
    if (MAC(prue)) {
        JOptionPane.showMessageDialog(null, "EQUIPO AUTORIZADO PARA EJECUTAR EL PROGRAMA");
    } else {
        JOptionPane.showMessageDialog(null, "EQUIPO NO AUTORIZADO PARA EJECUTAR EL PROGRAMA");
        this.dispose();
        System.exit(0);
    }
}

```

Ilustración 86. Código pantalla inicio sistema. Fuente: Autor.

1. Verifica los privilegios del usuario y activa las opciones del menú según corresponda.
2. Define una imagen de fondo.

3. Se cargan los datos en la tabla principal de acuerdo a la Base-Centro.
4. Verifica si el usuario tiene privilegios totales para activar un combo de filtro de registros.
5. Se cargan los registros en los combos referentes a los elementos de creados anteriormente en las tablas auxiliares.

```

public cp_arma() {
    initComponents();
    lbl_ruta.setVisible(false);
    float escalar = 0.84F;
    float escalar2 = 0.99F;
    int ancho = (int) (Toolkit.getDefaultToolkit().getScreenSize().width * escalar2);
    int alto = (int) (Toolkit.getDefaultToolkit().getScreenSize().height * escalar);
    this.setSize(ancho, alto);
    String tipo = cp_principal.userlogin.getTipousuarioid().getRegistroarma();
    if (tipo.equals("Editor")) {
    } else if (tipo.equals("Creador")) {
    } else if (tipo.equals("Lector")) {
        Grabar.setVisible(false);
    } else if (tipo.equals("Total")) {
    }
}

scroll.setOpaque(false);
jScrollPane.getViewport().setOpaque(false);
setImage("/Iconos/BackgroundIntendencia.jpg");

obj2.llenarte(tabla, baseo);
obj2.tabla(tabla);
obj3.combo(cmb_almacen);
obj4.combo(cmb_calibre);

if (baseo.equalsIgnoreCase("TOTAL")) {
    obj5.combo(cmb_centro);
} else {
    obj5.combo2(cmb_centro, baseo);
    cmb_centro.setEnabled(false);
}

obj5.combo(cmb_centro2);
obj6.combo(cmb_clase);
obj7.combo(cmb_color);
obj8.combo(cmb_estado);
obj9.combo(cmb_marca);
obj10.combo(cmb_pais);
obj11.combo(cmb_tipo);
obj12.combo(cmb_unidad);
obj14.combo(cmbmodel);
obj13.Combo(cmbbin);
  
```

Ilustración 87. Código pantalla administración material bélico sistema. Fuente: Autor.

1. Verifica los privilegios del usuario, en el caso de tener privilegios totales se activa un combo para filtrar los registros en base a la base-centro.

2. Verifica los privilegios del usuario y activa las opciones del menú según corresponda, conjuntamente con esto se realiza una carga de datos a los combos correspondientes, se crea un objeto de tipo ArmaClient para acceder al método buscar que retorna un registro en base al número de serie ingresado para adjuntar al registro del mantenimiento.

```

public cp mantenimiento() {
    initComponents();
    if (userlog.getCentroid().getCentronombre().equalsIgnoreCase("TOTAL")) {
        obj5.combo(cmb_centro2);
    } else {
        cmb_centro2.setVisible(false);
    }

    float escalar = 0.84F;
    float escalar2 = 0.99F;
    int ancho = (int) (Toolkit.getDefaultToolkit().getScreenSize().width * escalar2);
    int alto = (int) (Toolkit.getDefaultToolkit().getScreenSize().height * escalar);
    this.setSize(ancho, alto);
    String tipo = cp_principal.userlogin.getTipousuarioid().getMantenimientoarma();
    if (tipo.equals("Editor")) {
        jMenu1.setVisible(false);
        Bandera2 = true;
    } else if (tipo.equals("Creador")) {
        jMenu1.setVisible(false);
    } else if (tipo.equals("Lector")) {
        jMenu1.setVisible(false);
        Grabar.setVisible(false);
    } else if (tipo.equals("Total")) {
        Bandera2 = true;
    }
}
  
```

Ilustración 88. Código mantenimiento material bélico sistema. Fuente: Autor.

1. Se definieron los datos que se presentaran en los capos al momento de abrir la pantalla, se cargaron los registros en el combo, predefinimos la fecha actual en el JDateChooser, se definió un nuevo modelo para rellenar las columnas de la tabla principal.
2. Se realiza la comprobación de los privilegios del usuario, se rellenan las filas con todos los registros si tiene privilegios totales caso contrario se rellenan con registros según la Base-Centro.

```
1
usuario.setText(userlog.getUsuarionumidentificacion() + ", " + userlog.getUsuarioapellidos());
objtipo.Combo(jComboBox1);
jDateChooser1.setDate(new Date());
// setImage("/imagenes/8.jpg");
//objnew.llenarte(tabla);
jPanell.setVisible(true);
mdl3.addColumn("Id");
mdl3.addColumn("N.DOCUMENTO");
mdl3.addColumn("Modelo");
mdl3.addColumn("Notas");
tabla_libro.setModel(mdl3);

2
if (userlog.getCentroid().getCentronombre().equalsIgnoreCase("TOTAL")) {
    objManteni.llenarte(tabla);
    //objnew.llenarte(tabla);
    jPanell.setVisible(false);
} else {
    objManteni.llenarte(tabla, userlog.getCentroid().getCentronombre());
    //objnew.llenarte(tabla);
    jPanell.setVisible(false);
}

if (userlog.getCentroid().getCentronombre().equalsIgnoreCase("TOTAL")) {
    Grabar.setVisible(false);
    JOptionPane.showMessageDialog(null, "RECUERDE QUE NO PUEDE CREAR REGISTROS CON EL USUARIO BASE/ CENTRO : TOTAL , ");
}
}
```

Ilustración 89. Código mantenimiento material bélico parte dos sistemas. Fuente: Autor.

1. Es esta pantalla se necesitan instancias de objetos como son los de actas de cada solicitud generada anteriormente, que se obtienen a través del método findall que se accede a través de historial client, conjuntamente con esto se define e instancia un objeto de tipo Detalle historial arma porque la solicitud debe de contener varios usuarios involucrados en el cambio, que para mejorar la accesibilidad de obtienen todos sus datos a través del método find de usuario client que tiene como parámetro la cedula y devuelve un usuario y el archivo array de tipo byte que almacena el documento pdf adjunto a la solicitud.

```
public class cp_solicitudesarma extends InternalFrameImagen {

    HistorialClient hclient = new HistorialClient();
    String anime = "rrr";
    DetalleHistorialArmaClient dhclient = new DetalleHistorialArmaClient();
    UsuarioClient uclient = new UsuarioClient();
    byte[] ArchivoBytesVisualizar;
```

Ilustración 90. Variables solicitud actualización material bélico sistema. Fuente: Autor.

1. Para implementar las funcionalidades de esta pantalla se definió en primer lugar un objeto de tipo Intendencia cliente para acceder a sus métodos de guardar, buscar, actualizar, y eliminar, los objetos en base a las clases de las tablas auxiliares para acceder al método findall que retorna una lista de registros que serán cargados en los combos correspondientes, un array de tipo byte que almacenara el documento pdf adjunto, un detalle historial para guardar los elementos que consten en el registro, y por último los datos del usuario de la pantalla de inicio de sesión.

```
public class cp_intendencia extends InternalFrameImagen {  
  
    IntendenciaClient client = new IntendenciaClient();  
    AlmacenIntendenciaClient aclient = new AlmacenIntendenciaClient();  
    CentroClient centroclient = new CentroClient();  
    UnidadClient uclient = new UnidadClient();  
    BimClient bclient = new BimClient();  
    EstadoIntendenciaClient eclient = new EstadoIntendenciaClient();  
    ColorClient cclient = new ColorClient();  
    PaisClient pclient = new PaisClient();  
    Almacenintendencia almacen;  
    Centro centro;  
    Bim bim;  
    Unidad unidad;  
    Estadointendencia estadointendencia;  
    Color color;  
    Pais pais;  
    String ruta_archivo = "";  
    byte[] ArchivoBytesVisualizar;  
    int VerificadorClickTable = 0;  
    public static Intendencia intenciaselected;  
    HistorialIntendenciaClient hiclient = new HistorialIntendenciaClient();  
    DetalleHistorialIntendenciaClient dhiclient = new DetalleHistorialIntendenciaClient();  
    String VerificacionCambios = "";  
    Historialintendencia historialactualizacion;  
    String baseo = cp_principal.userlogin.getCentroid().getCentronombre();  
  
    /**  
     * Creates new form cp_intendencia  
     */  
}
```

*Ilustración 91. Código mantenimiento intendencia sistema. Fuente: Autor.*

1. En esta implementación se tuvo en cuenta una instancia del objeto usuario client para acceder a los métodos find, select, findall, save, para administrar las funciones principales de guardar, encontrar, listar, y eliminar, como segunda instancia se tiene el tipo de identificación para elegir un tipo de identificación ya sea pasaporte o cedula entre otras opciones predefinidas, para cargar los datos en los combos se tienen

instancias de las clases de las tablas auxiliares como sexo, tipo, jerarquía, ciudad, base-centro, y un array de tipo byte para guardar el documento adjunto de ser necesario.

```
public class cp_usuario extends InternalFrameImagen {  
  
    boolean bandera = false;  
    boolean Bandera2 = false;  
    UsuarioClient client = new UsuarioClient();  
    IdentificacionClient iclient = new IdentificacionClient();  
    SexoClient sclient = new SexoClient();  
    TipoUsuarioClient tclient = new TipoUsuarioClient();  
    JerarquiaClient jclient = new JerarquiaClient();  
    CiudadClient cclient = new CiudadClient();  
    CentroClient centroclient = new CentroClient();  
    Identificacion identificacion;  
    Sexo sexo;  
    Jerarquia jerarquia;  
    Tipousuario tipousuario;  
    Centro centro;  
    Ciudad ciudad;  
    String ruta_archivo = "";  
    byte[] ArchivoBytesVisualizar;  
    int VerificadorClickTable = 0;  
  
    /**  
     * Creates new form cp_usuario  
     */  
}
```

*Ilustración 92. Código mantenimiento usuario sistema. Fuente: Autor.*

1. Para esta pantalla primero se definieron los objetos con los cuales se trabajará para acceder a sus métodos y atributos, empezando con los datos de los centros que se interesa obtener los registros para cárgalos al combo, también con estado, tipo de arma, modelo y el BIM, todos estos objetos se crearon a partir de las clases auxiliares para los campos de material bélico.
2. En esta sección se cargan los registros de cada clase en los combos correspondientes para poder seleccionar una opción como filtro de una consulta.

```
public class cp_consultasarma extends javax.swing.JInternalFrame implements Printable {  
  
    CentroClient obj5 = new CentroClient();  
    EstadoClient obj8 = new EstadoClient();  
    TipoClient obj11 = new TipoClient();  
    ModeloClient obj14 = new ModeloClient();  
    int veri = 0;  
    int cont = 0;  
    BimClient obj13 = new BimClient();  
    int x = 0;  
    int ww = 0;  
    EstadoIntendenciaClient eclient = new EstadoIntendenciaClient();  
  
    /**  
     * Creates new form cp_consultas  
     */  
    public cp_consultasarma() {  
        initComponents();  
        float escalar = 0.84F;  
        float escalar2 = 0.99F;  
        int ancho = (int) (Toolkit.getDefaultToolkit().getScreenSize().width * escalar2);  
        int alto = (int) (Toolkit.getDefaultToolkit().getScreenSize().height * escalar);  
        this.setSize(ancho, alto);  
  
        obj8.combo(cmb_estado);  
        obj11.combo(cmb_tipo);  
        obj5.combo(cmb_Centro);  
        obj14.combo(cmbmodel);  
    }  
}
```

Ilustración 93. Código consulta material bélico sistema. Fuente: Autor.

1. En este caso los objetos con los cuales se trabajaron para acceder a sus métodos y atributos, son los datos de BIM, estado y centro para cárgalos a los combos, también todos estos objetos se crearon a partir de las clases auxiliares para los campos de material intendencia.
2. En esta sección se cargan los registros de cada clase en los combos correspondientes para poder seleccionar una opción como filtro de una consulta de material de intendencia.

```
public class cp_consultas extends javax.swing.JInternalFrame implements Printable {  
  
    BimClient obj13 = new BimClient();  
    int veri = 0;  
    int cont = 0;  
    int x = 0;  
    int ww = 0;  
    EstadoIntendenciaClient eclient = new EstadoIntendenciaClient();  
    CentroClient cclient = new CentroClient();  
  
    /**  
     * Creates new form cp_consultas  
     */  
    public cp_consultas() {  
        initComponents();  
        float escalar = 0.84F;  
        float escalar2 = 0.99F;  
        int ancho = (int) (Toolkit.getDefaultToolkit().getScreenSize().width * escalar2);  
        int alto = (int) (Toolkit.getDefaultToolkit().getScreenSize().height * escalar);  
        this.setSize(ancho, alto);  
        eclient.Combo(cmb_estado);  
        cclient.combo(cmb_centro);  
    }  
}
```

Ilustración 94. Código consulta material bélico sistema. Fuente: Autor.

1. En esta pantalla se filtran los movimientos de material bélico en base al centro, tipo y otros filtros, se necesitaron también una instancia de movimiento, de detalle del movimiento y de arma para poder mostrar todo el contenido del registro conjuntamente con las propiedades para imprimir por pantalla el material correspondiente incluido, se agregaron unas instancias de arreglos de tipo byte para almacenar los documentos adjuntos al registro como son; oficios y memorándums.
2. En esta sección se cargan los registros de cada clase en los combos correspondientes para poder seleccionar una opción como filtro de una consulta de movimientos material bélico.

```
public class cp_monitoreoarmas extends javax.swing.JInternalFrame implements Printable {  
  
    CentroClient cclient = new CentroClient();  
  
    TipoMovimientoClient tmclient = new TipoMovimientoClient();  
    int veri = 0;  
    int cont = 0;  
    int x = 0;  
    ArrayList<String> nnnn = new ArrayList();  
    byte[] ArchivoBytesVisualizarmemo;  
    String ArchivoString = "";  
    String ArchivoString2 = "";  
  
    byte[] ArchivoBytesVisualizar;  
    byte[] ArchivoBytesVisualizar2;  
    int ww = 0;  
    MovimientoarmasClient mi = new MovimientoarmasClient();  
    DetallemovimientoarmaClient mpp = new DetallemovimientoarmaClient();  
    ArmasClient maa = new ArmasClient();  
  
    /**  
     * Creates new form cp_consultas  
     */  
    public cp_monitoreoarmas() {  
        initComponents();  
        float escalar = 0.84F;  
        float escalar2 = 0.99F;  
        int ancho = (int) (Toolkit.getDefaultToolkit().getScreenSize().width * escalar2);  
        int alto = (int) (Toolkit.getDefaultToolkit().getScreenSize().height * escalar);  
        this.setSize(ancho, alto);  
        cclient.combo(cmbcentro);  
        tmclient.Combo(cmbtipomovimiento);  
    }  
}
```

Ilustración 95. Código monitoreo material bélico sistema. Fuente: Autor.

1. Se aplica el mismo proceso con diferentes campos de filtro los movimientos de material bélico en base al centro, tipo, etc. Se necesitó también una instancia de movimiento, de detalle del movimiento y de intendencia para poder mostrar todo el contenido del registro conjuntamente con las propiedades para imprimir por pantalla el material correspondiente incluido, se agregaron unas instancias de arreglos de tipo byte para almacenar los documentos adjuntos al registro como son; oficios y memorándums.
2. En esta sección se cargan los registros de cada clase en los combos correspondientes para poder seleccionar una opción como filtro de una consulta de movimientos material de intendencia.

```
public class cp_monitoreointendencia extends javax.swing.JInternalFrame implements Printable {  
  
    CentroClient cclient = new CentroClient();  
    ArrayList<String> nnnn = new ArrayList();  
    TipoMovimientoClient tmclient = new TipoMovimientoClient();  
    int veri = 0;  
    int cont = 0;  
    int x = 0;  
    byte[] ArchivoBytesVisualizar;  
    byte[] ArchivoBytesVisualizar2;  
    int ww = 0;  
    IntendenciaClient maa = new IntendenciaClient();  
    DetalleMovimientoIntendencia mpp = new DetalleMovimientoIntendencia();  
    MovimientoIntendenciaClient mi = new MovimientoIntendenciaClient();  
  
    /**  
     * Creates new form cp_consultas  
     */  
    public cp_monitoreointendencia() {  
        initComponents();  
        float escalar = 0.84F;  
        float escalar2 = 0.99F;  
        int ancho = (int) (Toolkit.getDefaultToolkit().getScreenSize().width * escalar2);  
        int alto = (int) (Toolkit.getDefaultToolkit().getScreenSize().height * escalar);  
        this.setSize(ancho, alto);  
        cclient.combo(cmbcentro);  
        tmclient.Combo(cmbtipomovimiento);  
    }  
}
```

Ilustración 96. Código monitoreo material intendencia sistema. Fuente: Autor.

En esta pantalla se definieron instancias de las clases auxiliares para centro y tipos de movimiento, principales de la capa de controlador para que se listen las armas disponibles para movimiento, registros del movimiento detalle validación del usuario sus privilegios con archivos byte para almacenar documentas en formato pdf .

```
public class cp_movimientoarma extends javax.swing.JInternalFrame {  
  
    boolean Bandera2 = false;  
  
    public static Date DROPFECHA;  
  
    CentroClient obj5 = new CentroClient();  
    DetallemovimientoarmaClient dmclient = new DetallemovimientoarmaClient();  
    ArmasClient iclient = new ArmasClient();  
    MovimientoarmasClient mclient = new MovimientoarmasClient();  
    TipoMovimientoClient tmclient = new TipoMovimientoClient();  
  
    CentroClient cclient = new CentroClient();  
    String ruta_archivomemo = "";  
    byte[] ArchivoBytesVisualizarmemo;  
    String ruta_archivoofi = "";  
    byte[] ArchivoBytesVisualizarofi;  
    Tipomovimiento tmovimiento;  
    Centro centro;  
  
    UsuarioClient uclient = new UsuarioClient();  
    Usuario usuario = uclient.find(Usuario.class, cp_principal.cedulita);  
    public static DefaultTableModel mdl2 = new DefaultTableModel();  
    public static Movimientoarmas m;  
    DefaultTableModel mdl = new DefaultTableModel();  
    Usuario userlog = uclient.find(Usuario.class, cp_principal.cedulita);  
}
```

Ilustración 97. Código movimiento material bélico sistema. Fuente: Autor.

1. Se definen los tamaños del JinternalFrame y se asigna el tamaño correcto a definir para esta pantalla de acuerdo al monitor que lo esté ejecutando, conjuntamente con esto se verifica que tenga los permisos correspondientes el usuario.
2. Se asignan los registros a los campos del combo, se genera un combo de acuerdo el centro y el usuario para poder listar los elementos de acuerdo al filtro, se activan los botones para visualizar archivos pdf, se genera una nueva fecha actual.
3. Se crea un modelo de tabla con las columnas correspondientes al detalle del movimiento y se asigna un tamaño adecuado a la pantalla.
4. Por último, se verifica los privilegios del usuario informando de que los usuarios con privilegios totales solo tendrán acceso de lector y no podrán realizar funciones de creación.

```
public cp movimientoarma() {  
    initComponents();  
    float escalar = 0.84F;  
    float escalar2 = 0.99F;  
    int ancho = (int) (Toolkit.getDefaultToolkit().getScreenSize().width * escalar2);  
    int alto = (int) (Toolkit.getDefaultToolkit().getScreenSize().height * escalar);  
    this.setSize(ancho, alto);  
    String tipo = cp_principal.userlogin.getTipousuarioid().getMovimientoarma();  
    if (tipo.equals("lector")) {  
        grabar.setVisible(false);  
    }  
    cmblient.combo(cmbtipomovimiento);  
    cmblient.comboMovimientoFiltro(cmbcentro, usuario.getCentroid());  
    btnvisualizarmemorandum.setVisible(false);  
    btnvisualizaroficio.setVisible(false);  
    chkestado.setSelected(true);  
    fecha.setDate(new Date());  
    ((JTextField) this.fecha.getDateEditor()).setEditable(false);  
    this.fecha.setSelectableDateRange(this.fecha.getDate(), this.fecha.getDate());  
    fechadevolucion.setEnabled(false);  
    ((JTextField) this.fechadevolucion.getDateEditor()).setEditable(false);  
    this.fechadevolucion.setMinSelectableDate(new Date());  
    mdl.addColumn("Id");  
    mdl.addColumn("Arma");  
    mdl.addColumn("Condición");  
    mdl.addColumn("Estado");  
    mdl.addColumn("Color");  
    mdl.addColumn("País");  
    mdl.addColumn("Cantidad");  
    tblmovimientodetalle.setModel(mdl);  
    TableColumnModel columnModel = tblmovimientodetalle.getColumnModel();  
    columnModel.getColumn(0).setPreferredWidth(20);  
    columnModel.getColumn(1).setPreferredWidth(90);  
    columnModel.getColumn(2).setPreferredWidth(50);  
    columnModel.getColumn(3).setPreferredWidth(50);  
    columnModel.getColumn(4).setPreferredWidth(50);  
    columnModel.getColumn(5).setPreferredWidth(50);  
    columnModel.getColumn(6).setPreferredWidth(20);  
    cmbid.setSelected(true);  
    if (userlog.getCentroid().getCentronombre().equalsIgnoreCase("TOTAL")) {  
        grabar.setVisible(false);  
        JOptionPane.showMessageDialog(null, "Recuerde que para realizar movimientos debe ingresar con un usuario con BASE/CENTRO diferente a total");  
    }  
}
```

Ilustración 98. Código movimiento material bélico parte dos sistemas. Fuente: Autor.

Se incluyen las variables globales para acceder a los métodos necesarios que permitan listar los registros de las tablas auxiliares, archivos pdf, e información de privilegios del usuario.

```
public class cp_movimientoingresoarma extends javax.swing.JInternalFrame implements Printable {  
  
    UsuarioClient uclient = new UsuarioClient();  
    ArmasClient iclient = new ArmasClient();  
    DetallemovimientoarmaClient dmiclient = new DetallemovimientoarmaClient();  
    JFileChooser seleccionar_archivo = new JFileChooser();  
    MovimientoarmasClient mclient = new MovimientoarmasClient();  
    ActaClient aclient = new ActaClient();  
    byte[] ArchivoBytesVisualizar;  
    byte[] ArchivoBytesVisualizar2;  
    int x = 0;  
  
    byte[] ArchivoBytesVisualizarmemo;  
    String ArchivoString = "";  
    String ArchivoString2 = "";  
  
    DetalleMaUsuario detalleactu = new DetalleMaUsuario();  
    int cont = 0;  
    int veri = 1;  
    DetalleMaUsuarioClient ddt = new DetalleMaUsuarioClient();  
    List<DetalleMaUsuario> Listadetalle = new ArrayList<>();  
    List<Usuario> usuarios = new ArrayList();  
    List<DetalleMaUsuario> DetalleMaUsuariolist = new ArrayList();  
    Usuario usuario = uclient.find(Usuario.class, cp_principal.cedulita);  
    int verificar = 0;  
}
```

*Ilustración 99. Código pantalla prestamos activos material bélico sistema. Fuente: Autor.*

1. Se importan las librerías y paquetes necesarios para acceder a sus métodos CRUD.
2. Se crean dos variables booleanas para almacenar acciones del sistema al momento de generar un nuevo registro o actualizar, variables de tipo y detalle donde se accede a las funciones para listar, buscar, agregar, eliminar, actualizar los registros.
3. Un requerimiento fundamental es de poder contar con niveles de privilegios, en esta parte se verifica los diferentes tipos de usuario para conceder o no el permiso de realizar determinadas acciones.

```
import CRUD.TipoMovimientoClient;
import CRUD.MovimientoIntendenciaClient;
import Entity.MovimientoIntendencia;
import Entity.TipoMovimiento;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
```

1

```
/**
 *
 * @author Dremko
 */
public class cp_tipomovimiento extends InternalFrameImagen {

    boolean bandera = false;
    boolean Bandera2 = false;
    TipoMovimientoClient client = new TipoMovimientoClient();
    MovimientoIntendenciaClient miclient = new MovimientoIntendenciaClient();

    /**
     * Creates new form cp_tipomovimiento
     */
    public cp_tipomovimiento() {
        initComponents();
        String tipo = cp_principal.userlogin.getTipousuario().getTipomovimiento();
        if (tipo.equals("Editor")) {
            Eliminar.setVisible(false);
            Bandera2 = true;
        } else if (tipo.equals("Creador")) {
            Eliminar.setVisible(false);
        } else if (tipo.equals("Lector")) {
            Eliminar.setVisible(false);
            grabar.setVisible(false);
        } else if (tipo.equals("Total")) {
            Bandera2 = true;
        }

        setImage("/Iconos/BackgroundIntendencia.jpg");
    }
}
```

2

3

Ilustración 100. Librerías y código pantallas tablas auxiliares genéricas sistema. Fuente: Autor.

## Bibliografía

- Ariel Arsaute, F. Z. (2018). *http://sedici.unlp.edu.ar*. Obtenido de sedici:  
[http://sedici.unlp.edu.ar/bitstream/handle/10915/67777/Documento\\_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y](http://sedici.unlp.edu.ar/bitstream/handle/10915/67777/Documento_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y)
- Arsaute, A. (2018). *http://sedici.unlp.edu.ar*. Obtenido de sedici:  
[http://sedici.unlp.edu.ar/bitstream/handle/10915/67777/Documento\\_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y](http://sedici.unlp.edu.ar/bitstream/handle/10915/67777/Documento_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y)
- Computación, Dept. Ciencia de la. (2013). *Dept. Ciencia de la Computación*. Obtenido de Dept. Ciencia de la Computación:  
<http://www.jtech.ua.es/j2ee/restringido/cw/sesion11-apuntes.pdf>
- Durán, Y. (2012). *redalyc*. Obtenido de <https://www.redalyc.org/pdf/4655/465545892008.pdf>
- Guevara, J. M. (2016). *esuva*. Obtenido de <https://www.tesuva.edu.co/phocadownloadpap/Fundamentos%20de%20programcion%20en%20Java.pdf>
- Jean Layonel Quiñónez Medina. (2016). *pucese*. Obtenido de [repositorio.pucese.edu.ec/bitstream/123456789/815/1/QUINONEZ%20MEDINA%20JEAN%20LAYONEL.pdf](https://repositorio.pucese.edu.ec/bitstream/123456789/815/1/QUINONEZ%20MEDINA%20JEAN%20LAYONEL.pdf)
- JOSÉ CAMILO ARIAS ARIAS. (2016). *opac*. Obtenido de [opac.pucv.cl](http://opac.pucv.cl/pucv_txt/txt-6500/UCD6681_01.pdf):  
[http://opac.pucv.cl/pucv\\_txt/txt-6500/UCD6681\\_01.pdf](http://opac.pucv.cl/pucv_txt/txt-6500/UCD6681_01.pdf)
- JUÁREZ, I. C. (2019). *https://koha.uptlaxponente.edu.mx*. Obtenido de [uptlaxponente](https://koha.uptlaxponente.edu.mx/memorias/SIS.dir/2019.dir/SISTEMA%20DE%20ADMINISTRACI%C3%93N%20PARA%20EL%20CONTROL%20DE%20INVENTARIO%20EN%20ALMAC%C3%89N.pdf):  
<https://koha.uptlaxponente.edu.mx/memorias/SIS.dir/2019.dir/SISTEMA%20DE%20ADMINISTRACI%C3%93N%20PARA%20EL%20CONTROL%20DE%20INVENTARIO%20EN%20ALMAC%C3%89N.pdf>
- Morocho, S. T. (2018). *sangabrielriobamba*. Obtenido de [www.sangabrielriobamba.edu.ec](http://www.sangabrielriobamba.edu.ec):  
<https://www.sangabrielriobamba.edu.ec/tesis/sistemas/tesis011.pdf>
- NESTOR JAVIER MUNCA CADENA. (2010). *repository*. Obtenido de [repository.uniminuto.edu](https://repository.uniminuto.edu):  
[https://repository.uniminuto.edu/bitstream/10656/2084/1/TTI\\_MuncaCadenaNestorJavier\\_2010.pdf](https://repository.uniminuto.edu/bitstream/10656/2084/1/TTI_MuncaCadenaNestorJavier_2010.pdf)
- Paré, R. C. (2015). *uoc*. Obtenido de [uoc](http://www.uoc.edu):  
<https://www.uoc.edu/pdf/masters/oficiales/img/913.pdf>
- Real Academia Española. (2021). *bélico*. En *Diccionario de la Lengua Española (edición de tricentenario)*. Obtenido de <https://dle.rae.es/b%C3%A9lico>
- Real Academia Española. (s.f.). *Intendencia*. En *diccionario de la Lengua Española (edición de tricentenario)*. Recuperado el 16 de Diciembre de 2021, de <https://dle.rae.es/intendencia?m=form>



Romero, E. M. (2019). *repository*. Obtenido de repository.ucc.edu.co:  
[https://repository.ucc.edu.co/bitstream/20.500.12494/8557/3/2019\\_Dise%C3%B1o\\_implementaci%C3%B3n\\_sistema.pdf](https://repository.ucc.edu.co/bitstream/20.500.12494/8557/3/2019_Dise%C3%B1o_implementaci%C3%B3n_sistema.pdf)

UA. (2017). *Automatizar*. Obtenido de  
[https://rua.ua.es/dspace/bitstream/10045/18432/1/Tema%201\\_Introduccion.pdf](https://rua.ua.es/dspace/bitstream/10045/18432/1/Tema%201_Introduccion.pdf)

## ANEXOS

### Anexo 1: Protocolo de Tesis

A. TÍTULO
Sistema de gestión de inventarios, movimientos de material bélico y de intendencia a nivel nacional para la DIRMOV.

B. DOMINIO, LÍNEA Y ÁMBITOS DE INVESTIGACIÓN			
<b>Tecnologías de Información y Comunicación</b>	<b>Ciencias exactas, naturales y tecnológicas</b>	Analítica de datos	
		Ingeniería de software	X
		Algoritmos de computadora	
		Inteligencia de negocios	
		Gobierno de TI	
		Auditoría y seguridad informática	
		Simulación	

C. PLANTEAMIENTO DEL PROBLEMA
<p>El principal inconveniente que se presenta en la DIRMOV, es el no poder gestionar los registros de movimientos y creación de inventarios de material bélico y de intendencia, conjuntamente con solicitudes de aprobación y seguridad en el manejo de los mismos de forma eficiente, existe un déficit de control de la información debido al uso precario de hojas de cálculo en Excel limitando la funcionalidad a un entorno local sin ninguna base de datos compartida por lo que la interacción y comprobación en tiempo real de las transacciones y registros que realicen los demás usuarios son nulas, provocando una evidente acumulación de solicitudes de aprobaciones, y dificultad en el registro y verificación de los mismos.</p> <p>Esta falta de eficiencia en la creación de registros provoca complicaciones a los encargados y un retraso considerable en la verificación de información, ya que todas estas actividades ordinarias que deben realizar controles y documentación de las mismas, que se pueden optimizar con la incorporación de un sistema de inspección enfatizado para encargados y los</p>

jefes departamentales representantes de la DIRMOV, siendo los encargados de administrar y controlar ese tipo de recursos que le son de suma importancia para el pueblo ecuatoriano.

La disciplina de ingeniería de software perteneciente al campo de ciencias de la computación pretende dar solución a este escenario de crisis, incrementando así los niveles de desarrollo tecnológico mediante el uso de medios con tecnología de punta.

#### **D. OBJETIVO GENERAL**

Desarrollar un sistema de gestión de inventarios, movimientos de material bélico y de intendencia a nivel nacional para la DIRMOV.

#### **E. OBJETIVOS ESPECÍFICOS**

1. Obtener los requerimientos del sistema.
2. Desarrollar una base de datos.
3. Crear un API Rest.
4. Implementar los requerimientos del sistema mediante la creación de un software.
5. Realizar pruebas e implementar el software.

#### **F. JUSTIFICACIÓN**

Los motivos que llevaron a la realización de este sistema surgen principalmente debido al notable desarrollo tecnológico que ha avanzado mucho en estos últimos años, por tal motivo la DIRMOV también depende en gran medida de la tecnología para cumplir con sus actividades de manera ágil y correcta, por esto la necesidad de poder contar con un sistema informático que cubra sus requerimientos, es por esto que puesto a brindar la ayuda correspondiente con el fin de cubrir estos problemas generando una solución a la falta de control del manejo del informático.

#### **G. ALCANCE**

El sistema implementará funcionalidades específicas para el registro de usuarios con su correspondiente localización y acceso solamente a los registros y funcionalidades disponibles para esa base/centro, para el registro del material bélico y de intendencia conjuntamente con funcionalidades para poder registrar los movimientos entre los diferentes centros del país, también incluirá filtros de búsqueda exportación de reportes, aprobación de solicitudes y mantenimiento.

## H. CONCEPTOS RELACIONADOS

**Java.** Es un lenguaje de programación muy valorado debido al software desarrollado de este lenguaje se puede elaborar en varias plataformas con métodos operativos como Windows, Mac, OS y Linux. (Guevara, 2016, pág. 8).

**Inventario.** Son todos aquellos apartados o stocks deslucidos en los procesos de la organización, personifica una de las alteraciones más significativas de las organizaciones con los activos. (Durán, 2012, pág. 55).

**Material bélico.** Dentro del marco de trabajo para el desarrollo del sistema hace referencia a todo material que será registrado que sea perteneciente a la guerra (Real Academia Española, 2021).

**Material de intendencia.** Este concepto hace referencia a todo tipo de material que será inscrito en el método referente a un organismo y multitud destinado al suministro de las fuerzas militares y a la distribución de los campamentos o edificios en que se alojan (Real Academia Española).

**Automatizar.** En nuestro entorno de desarrollo es importante entender este concepto que hace referencia a todo método competente de transformar trabajos previamente y en terminantes de espacio y tiempo sin escasez de mediación condesciende. (UA, 2017, pág. 5)

**El estilo REST (Representational State Transfer).** Consiste en establecer servicios Web de manera óptima. Mediante el mecanismo esencial está dirigido a los valores como son las URLs por medio de la utilización en AJAX (Dept. Ciencia de la Computación, 2013, pág. 2).

**Servicio web RESTful.** El hiperenlace es la magnífica manera de mercantilizar caracteres entre consumidoras y servidores. (Dept. Ciencia de la Computación, 2013).

**JSON.** Los mecanismos están comprendidos de llaves {y}; los valores de las disímiles síntesis que se agrupan en pares, con la estructura: "nombre": "valor", y están separados por comas, las sucesiones de síntesis están por medio de broches [ y] (Dept. Ciencia de la Computación, 2013).

**JAX-RS y Jersey.** La herramienta de java forma parte de los medios para realizar servicios web. REST, en el proyecto JAX-RS dio inicio por Java Community Process (JCP) con el propósito de establecer una API para servicios RESTful. El API conocido como JAX-311 o JAX-RS. Es empleado por diferentes ámbitos para su implementación de un framework. (jtech, 2013, pág. 22).

**Base de Datos.** Hace referencia a la representación compuestas y vinculados de formas necesarias para todo tipo de SI. Esta debe poseer autoridad para ser empleada de manera compartida por varios usuarios de diferente índole. Es por ello que una base de datos está vinculado estructurado de información y entidades y sus interrelaciones, representada de manera única e integrada. (Paré, 2015, pág. 18).

**Integridad de los datos.** Permiten por el medio del sustento de la aptitud de las reseñas en diferentes indoles. La abundancia puede provocar pérdida de honradez de los datos, para perder la ciencia y estabilidad de errores que puedan surgir en la información. (pág. 28).

**Seguridad.** Los SGBD se emplea para hacer un informe en beneficio de la empresa de manera óptima y segur. (pág. 31).

## I. TRABAJOS RELACIONADOS

(Esneider Martin Romero, 2019) Por medio del análisis de requerimientos se gestionó el inventario mediante la aplicación de buenas prácticas por medio de la aplicación estándar de JAVA EE, el framework JSF y como el principal motor de la base de datos. Mediante esta solución se almacena para poder usar el administrador, para realizar sus actividades tales como datos de los inventarios así como también se enfoca en los recursos de manera óptima para el crecimiento empresarial.

Este trabajo guarda relación con nuestro proyecto y es de suma importancia para entender los procesos y métodos de recolección de requerimientos, para su posterior análisis y división en: requerimientos funcionales y no funcionales.

Donde se describe punto por punto cada nivel de gestión y acceso que proveerá el software y su interacción con el usuario final mediante la representación en casos de uso, diagramas de secuencia y de procesos.

En el documento realizado por (JOSÉ CAMILO ARIAS ARIAS, 2016) se presenta el desarrollo acerca de una herramienta, con el objetivo de dar a conocer el sistema de automatización de un bar. Se identificaron las necesidades de acuerdo al estado en el que se encontraba el lugar. Fue por ello que se desarrolló una herramienta por medio de un prototipo para realizar interfaces de fácil y óptima seguridad. Para la autenticación de las personas que laboran dentro del lugar utilizaban las tarjetas de RFID, por lo que la herramienta es de gran utilidad para identificar los productos de consumo y administración del local.

La parte del trabajo que es de suma importancia se centra principalmente en la utilización de la técnica de desarrollo en espiral considerando como un modelo para la realización del prototipo y que se utiliza al momento de ser desarrollado el software por una sola persona, teniendo ventajas significativas como brindar al beneficiario final un panorama de antecedentes de una parte de las instrucciones en cada ciclo de desarrollo.

Esta metodología es aplicable a nuestro proyecto al momento de identificar requerimientos generales del sistema llevando a cabo un modelado de diseño rápido, generando una parte del software que será visible para el usuario final, como son el caso de las interfaces, que de alguna manera son entendibles y se puede recibir una retroalimentación para afinar más detalles.

Según (NESTOR JAVIER MUNCA CADENA, 2010) Expone en su trabajo proceso de diseño de un software para cubrir las necesidades acerca de una organización que labora con productos y servicios con el objetivo de controlar el ingreso y la salida del producto del local, por medio de este software se a creado una plataforma en java con relación a los objetos con propósitos de procesar los datos y los objetos para brindar mayor seguridad en la información así como también en la optimización de trabajo por medio de herramientas como es la plataforma de Excel.

Este proyecto sirve como guía para la construcción de una propuesta de recurso de software en una expresión de programación encaminado a objetos (java) para el proceso de datos, utilizando la metodología del CRUD. Mediante una Interfaz gráfica de formularios de captura de datos y presentación del proyecto, procesamiento e interfaz gráfica de NETBEANS,

JAVA, afirmación de clases y subclases con sus respectivos atributos para la incorporación de la base de datos.

Según (TRUJILLO MOROCHO SAMANTHA ELIZABETH, 2018) Con el propósito de mantener la información de inventarios de los equipos de cómputo de una forma ágil, oportuna e informatizada. Se utilizó una metodología en cascada basada en una arquitectura Cliente Servidor, utilizando el lenguaje de programación JAVA en su versión 8.0 que permitió crear formularios, contenidos, mediante el uso de IDEs y herramientas que permitieron la creación de las pantallas que los usuarios ejecutan al momento de subir información en el sistema

Se realizó la implementación del sistema con 2 módulos los cuales están relacionados, de acuerdo a los requerimientos del usuario para la creación de un inventario.

En este trabajo se prestan aspectos necesarios para entender los procesos de registro considerando un aspecto importante la sección de la programación orientada a objetos lo que conforma el sistema para el inventario en tiempo real.

Otro aspecto relevante el trabajo es que se desarrolla en NetBeans y se detalla el proceso de creación por medio de aspectos importantes de software conocidos como módulos.

Según (Jean Layonel Quiñónez Medina, 2016) Expone en su proyecto de tesis, la creación de una aplicación contable que permita la administración de los activos fijos de la PUCESE. Al tener la PUCESE la necesidad de llevar la administración contable de los activos fijos, se presentó la proposición de frena para realizar un sistema informático que permita acceder a administrar dichos activos fijos que posee la institución.

Uno de los principales requerimientos que se necesitaba tener en la aplicación, es que cuente con el historial de la ubicación de los activos fijos, además el tema de la depreciación de los mismos. La aplicación contable cumplió con los objetivos establecidos que eran crear la aplicación contable y mejorar la administración contable de los activos de una institución pública.

Se conoce la importancia en este trabajo que radica por la aplicación de las fases del diseño de software empezando con el Diagrama de flujos de datos, Esquema de Contexto, esquema de casos de Uso, Descripción por las Tablas con Datos, Diseño de Datos, y Diseño de la interfaz y creación de un modelo de 3 capas MVC.

## J. METODOLOGÍA

Los motivos que llevaron a la selección del modelo de metodología en espiral se basaron principalmente en el análisis comparativo expuesto en la siguiente tabla (José Alejandro Ferto Martínez):

Modelos	Descripción	Pasos	Ventajas	Desventajas
Secuencial o Cascada	Sugiere un enfoque sistemático, secuencial, para el desarrollo del software.	<ol style="list-style-type: none"> <li>1. Análisis</li> <li>2. Diseño</li> <li>3. Código</li> <li>4. Pruebas</li> </ol>	<ul style="list-style-type: none"> <li>• Es un modelo sencillo y disciplinado</li> <li>• Es fácil aprender a utilizarlo y comprender su funcionamiento</li> <li>• Ha sido muy usado</li> </ul>	<ul style="list-style-type: none"> <li>• Los proyectos reales raras veces siguen el modelo secuencial que propone el modelo</li> <li>• Es difícil que el cliente exponga explícitamente todos los requisitos</li> </ul>
Evolutivo	Consta del desarrollo de una versión inicial que luego de exponerse se va refinando de acuerdo de los comentarios o nuevos requerimientos por parte del cliente o del usuario final. Las fases de especificación, desarrollo y validación se entrelazan en vez de separarse.	<ul style="list-style-type: none"> <li>• Descripción del sistema</li> <li>• Especificación</li> <li>• Desarrollo</li> <li>• Validación</li> <li>• Re-especificación</li> </ul>	<ul style="list-style-type: none"> <li>• El sistema se va ajustando a las necesidades del cliente.</li> <li>• El cliente entiende mejor sus propios requerimientos</li> </ul>	<ul style="list-style-type: none"> <li>• El proceso no es visible.</li> <li>• Los cambios continuos tienden a corromper la estructura del software. Incorporar cambios en él se convierte cada vez más en una tarea difícil y costosa.</li> </ul>
Incremental	Aplica secuencias lineales de forma escalonada mientras progresa el tiempo en el calendario. Cada secuencia lineal produce un «incremento» del software.	<ol style="list-style-type: none"> <li>1. Análisis</li> <li>2. Diseño</li> <li>3. Código</li> <li>4. Pruebas (se vuelven a repetir las veces necesarias)</li> </ol>	<ul style="list-style-type: none"> <li>• Reduce el tiempo de desarrollo inicial, ya que se implementa la funcionalidad parcial.</li> <li>• Provee un impacto ventajoso frente al cliente, que es la entrega temprana de partes operativas del</li> </ul>	<ul style="list-style-type: none"> <li>• Muchas funciones suplementarias (algunas conocidas, otras no) quedan sin extraer.</li> <li>• El plan afronta la modificación del producto central a fin de cumplir mejor las necesidades del cliente</li> </ul>
Construcción de prototipos	Ofrece un enfoque mejor, con un diseño rápido, definiendo los requisitos y objetivos.	<ol style="list-style-type: none"> <li>1. Escuchar al cliente</li> <li>2. Construir/Revisar la maqueta</li> <li>3. El cliente prueba la maqueta (Se vuelve a comenzar)</li> </ol>	<ul style="list-style-type: none"> <li>• Este modelo es útil cuando el cliente conoce los objetivos generales para el software, pero no identifica los requisitos detallados de entrada, procesamiento o salida.</li> </ul>	<ul style="list-style-type: none"> <li>• El usuario tiende a crearse unas expectativas cuando ve el prototipo de cara al sistema final. A causa de la intención de crear un prototipo de forma rápida, se suelen desatender aspectos importantes, tales como la calidad y el mantenimiento a largo plazo, etc.</li> <li>• Se puede utilizar un sistema operativo o lenguaje de programación inadecuado simplemente porque está disponible y porque es conocido</li> </ul>
Espiral	Es un modelo de proceso de software evolutivo que conjuga la naturaleza iterativa de construcción de prototipos con los aspectos controlados y sistemáticos del modelo lineal secuencial	<p>Contiene seis regiones de tareas:</p> <ol style="list-style-type: none"> <li>1. Comunicación con el cliente</li> <li>2. Planificación</li> <li>3. Análisis de riesgos</li> <li>4. Ingeniería</li> <li>5. Construcción y adaptación</li> <li>6. Evaluación del cliente</li> </ol>	<ul style="list-style-type: none"> <li>• El coste y la planificación se ajustan con la realimentación ante la evaluación del cliente.</li> <li>• El gestor del proyecto ajusta el número planificado de iteraciones requeridas para completar el software.</li> <li>• Puede adaptarse y aplicarse a lo largo de la vida del Software de computadora.</li> </ul>	<ul style="list-style-type: none"> <li>• Puede resultar difícil convencer a grandes clientes (particularmente en situaciones bajo contrato) de que el enfoque evolutivo es controlable.</li> <li>• Requiere una considerable habilidad para la evaluación del riesgo.</li> <li>• Si un riesgo importante no es descubierto y gestionado, indudablemente surgirán problemas.</li> </ul>
Desarrollo basado en componentes	Incorpora muchas de las características del modelo en espiral. Es evolutivo por naturaleza, y exige un enfoque iterativo para la creación del software. Sin embargo, el modelo de desarrollo basado en componentes configura aplicaciones desde componentes preparados de software (llamados «clases»)	<ol style="list-style-type: none"> <li>1. Comunicación con el cliente</li> <li>2. Planificación</li> <li>3. Análisis de riesgos.</li> <li>4. Construcción y evaluación de la ingeniería.</li> <li>5. Evaluación del cliente.</li> </ol>	<ul style="list-style-type: none"> <li>• Reutilización del software.</li> <li>• Simplifica las pruebas.</li> <li>• Ciclos de desarrollo más cortos</li> </ul>	<ul style="list-style-type: none"> <li>• Los "compromisos" en los requisitos son inevitables, por lo cual puede que el software no cumpla las expectativas del cliente.</li> <li>• Las actualizaciones de los componentes adquiridos no están en manos de los desarrolladores del sistema.</li> </ul>

101 Cuadro comparativo Modelos de desarrollo de software (José Alejandro Ferto Martínez)

Para (UNEMI, 2011), la definición acerca de un modelo importante para la creación de un software es de acuerdo a las actividades que se desarrollen mediante el diseño, así como también se puede determinar por medio de las etapas del sistema como se consideran las siguientes:

1. Un modelo de ciclo de vida del software:
2. Relata los períodos importantes del desarrollo de software.
3. Precisa los periodos principales deseadas para ser aprovechadas durante las etapas.
4. Ayuda a administrar el adelanto del progreso, y Provee una plaza de adeudo para la enunciación de un minucioso sumario de progreso de software.
5. Las Guías son esenciales para establecer en varias técnicas de desarrollo de los proyectos es por ello la importancia de suministrar el mantenimiento con el objetivo de permitir la reutilización de recursos, así como la guía correspondiente para el avance y control.

**MODELO ESPIRAL** El modelo permite el progreso del software basado en el período de vida del software es por ello que es iterativo, mayoritariamente está en procesos de desarrollo para ello se debe seguir los siguientes pasos:

1. Establecer o determinar los imparciales. Se establece los objetivos específicos para asemejar las restricciones del sumario y del sistema de software, y se procede a diseñar una organización minuciosa de encargo mediante la identificación de los riesgos.
2. Análisis del riesgo. Se desarrolla un estudio informativo de cada riesgo identificado en el proyecto estableciendo pasos para mitigar el riesgo de maneras estratégicas.
3. Desarrollar, comprobar y validar.

Se realiza el examen de riesgo, se eligen un ejemplar para el adelanto del sistema de software y se ejecuta.

4. Planificar. El proyecto se somete a revisión para poder tomar decisiones con un tipo de ciclo de acuerdo a los planes de la fase del proyecto.

Se puede mencionar que por medio de las iteraciones espirales con el objetivo de que el sistema funcione de manera correcta, es por ello que se utiliza ese modelo para evaluar el riesgo.

Es necesario identificar el riesgo en el proyecto para evitar posibles errores como puede ser al utilizar un lenguaje de programación, así como también la identificación necesaria de compiladores que estén aptos para el código eficiente.

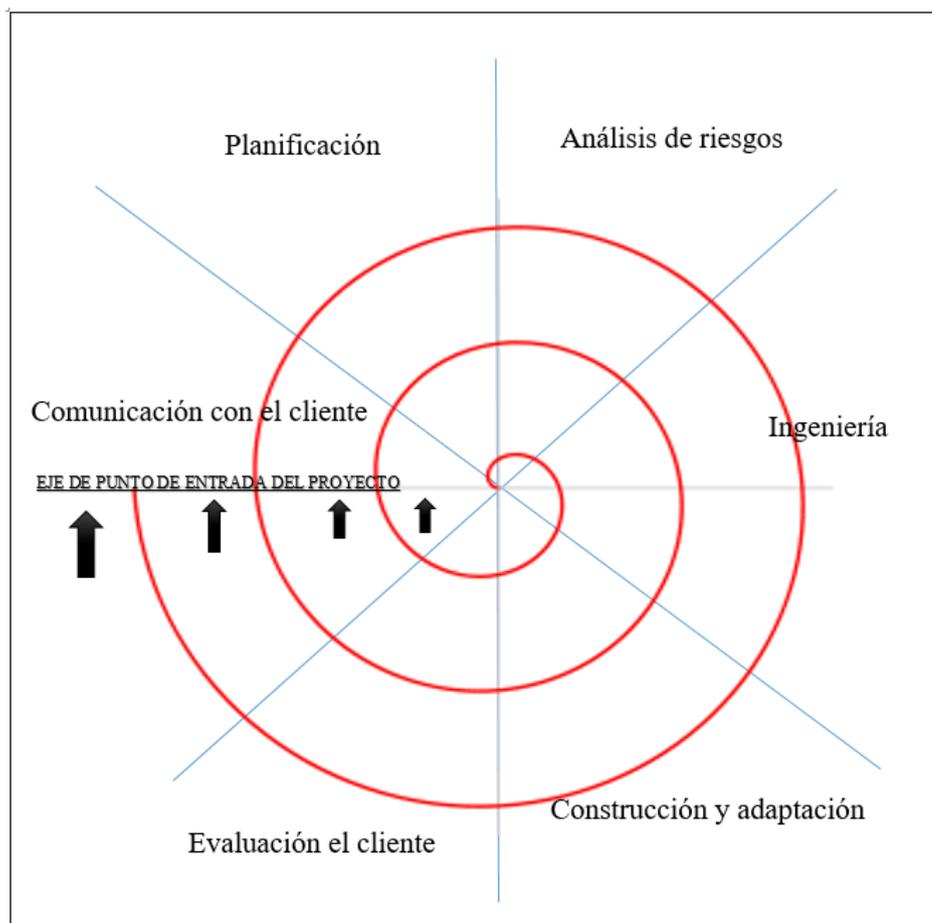
El modelo espiral da indicios por los objetivos planteados como el principal es acerca del rendimiento, es por ello que se logra cumplir con los objetivos acerca de los riesgos identificados existentes, tomando en consideración la resolución de estos riesgos para poder cumplir con las actividades propuestas y dar cumplimiento del ciclo de la espiral.

#### CARACTERÍSTICAS DEL MODELO EN ESPIRAL PARA EL DESARROLLO DE SOFTWARE.

Hace referencia a una guía evolutiva por medio del ciclo de la vida para la realización de prototipos, es por ello que se encarga de realizar regiones acerca de las tareas.

Se pueden considerar las siguientes regiones de tareas tales como:

1. Dar Aviso al cliente: Se encarga de determinar la comunicación entre el desarrollador y el cliente.
1. Planificación: Se define los recursos necesarios relacionadas con el proyecto como son los requerimientos.
2. Análisis de riesgos: se debe aplicar el modelo en espiral, necesario para la evaluación de riesgos técnicos con relación al proyecto.
3. Ingeniería: de debe construir más de una representación para la aplicación de la misma.
4. Construcción y adaptación: Es requerida en el modelo espiral por las necesidades de construcción, aprobación, instalación con objetivos de dar soporte al usuario.
5. Evaluación el cliente: Es necesaria para obtener aspectos positivos de los clientes mediante representaciones realizadas del software en la etapa de la implementación e instalación.



102 Metodología de desarrollo en espiral -Autoría Propia



K. CRONOGRAMA DE ACTIVIDADES								
N°	ACTIVIDAD	MES						MEDIOS DE VERIFICACIÓN
		I	II	III	IV	V	VI	
<b>1</b>	<b>Obtener los requerimientos del sistema.</b>							
2	Crear reuniones virtuales donde se definan las bases fundamentales para la construcción de un bosquejo donde se definan las limitaciones, el alcance y funciones principales del sistema.	x						Revisión del material recibido donde se detalla la información correspondiente a cada proceso dentro de la DIRMOV para la comprobación.
3	Documentar las entidades con sus correspondientes atributos y la distribución de accesos a cada menú que implementara la interfaz gráfica del sistema.	x						Comprobación de los campos de cada entidad documentados en los archivos recibidos de parte de la DIRMOV.
<b>4</b>	<b>Desarrollar una base de datos.</b>							
5	Creación de un Modelo de datos conceptual		x					Comprobación de relaciones y dependencias funcionales de cada tabla en base a la lista de requerimientos .
6	Creación de un Modelo de datos lógico		x					Comprobación de errores mediante el compilador de power designe en cuanto a la generación del modelo lógico.
7	Creación de un Modelo de datos Físico en la base de datos SQL SERVER.		x					Comprobación de errores mediante el compilador de power designe en cuanto a la generación del modelo físico y revisión dentro de la base de datos SQL SERVER.



<b>8</b>	<b>Crear un API Rest.</b>							
<b>9</b>	Crear una conexión mediante el IDE de desarrollo Neatbeans con la base de datos SQL SERVER.			x				Utilización de una instancia de Conexión para obtener una verificación de acceso correcto a la base de datos SQL SERVER.
<b>10</b>	Aplicar el proceso de RESTFULL que implementa Neatbeans para la generación del API REST correspondiente.			x				Verificación mediante POSTMAN de envío y recepción de paquetes de tipo JASON.
<b>11</b>	Modificar el código generado en el proceso REST-FULL para que cumpla con las funciones CRUD específicas de cada una de las entidades.			x				Verificación mediante POSTMAN de envío y recepción de paquetes específicos para cada entidad de tipo JASON.
<b>12</b>	<b>Implementar los requerimientos del sistema mediante la creación de un software.</b>							
<b>13</b>	Crear la capa de presentación con el menú principal conjuntamente con las interfaces gráficas para las 48 entidades.				x			Comprobación mediante la comparación con los campos correspondientes a cada entidad en la base de datos.
<b>14</b>	Crear la capa de datos con las 48 entidades y sus correspondientes atributos.				x			Verificación de la asignación a cada campo implementado en la interfaz con su correspondiente atributo dentro de la capa de datos.
<b>15</b>	Crear la capa de negocios con el proceso CRUD para las 48 entidades correspondientes.					x		Comprobación mediante la ejecución del proceso CRUD y obtención de respuestas correctas.
<b>16</b>	Creación de la capa de seguridad para el inicio de sesión de acuerdo a los					x		Prueba de acceso mediante el intento de ingreso por parte de



	privilegios de cada usuario.							usuarios no autorizados. O que no poseen los privilegios adecuados.
<b>17</b>	<b>Realizar pruebas e implementar el software.</b>							
<b>18</b>	Comprobar la precisión al momento de realizar los procesos de carga de datos, siendo este un factor crucial para el éxito del sistema.						x	Aplicación de las comprobaciones correspondientes mediante un método directo de ingreso de datos.
<b>19</b>	Comprobar el proceso de actualización de registros tanto de material bélico como material de intendencia y movimientos de material.						x	Comprobación de errores mediante la utilización del compilador de Neatbeans y Postman y revision en la base de datos.
<b>20</b>	Revisar los campos del listados de registros al momento de generar un FindAll(Recuperar todos los registros). Revisar el proceso de eliminación de un registro y el proceso de aprobación por parte de cada uno de los responsables.						x	Comprobación de errores mediante la utilización del compilador de Neatbeans y Postman y revision en la base de datos.
<b>21</b>	Poner en marcha el sistema con la correcta instalación de la base de datos, iniciar el servidor y descargar el aplicativo ejecutable en las maquinas correspondientes.						x	Comprobación por parte de los usuarios finales.



### L. DECLARACIÓN FINAL

Los abajo firmantes declaramos bajo juramento que el proyecto descrito en este documento no ha sido presentado a otra institución nacional o internacional para su financiamiento, no causa perjuicio al ambiente, es de nuestra autoría y no transgrede norma ética alguna.

### M. PARTICIPANTES

DIRECTOR:	José Antonio Carrillo Zenteno
ESTUDIANTE 1	Cristian Israel <u>Siguencia</u> Salazar

### N. FIRMAS DE RESPONSABILIDAD

Lugar:	Cañar
Fecha:	28 de enero de 2022

Firmas:

Nombre: José Antonio Carrillo Zenteno  
CC:0103304531  
**Director del Proyecto**

Nombre: Cristian Siguencia  
C.C.:0302790555  
**Estudiante / Egresado**

### O. APROBACIÓN

Firmas:

Nombre:  
CC:  
**Primer Par Revisor**

Nombre:  
C.C.:  
**Segundo Par Revisor**



## Anexo 2: Certificado para subir a repositorio institucional

### AUTORIZACIÓN DE PUBLICACIÓN EN EL REPOSITORIO INSTITUCIONAL

Yo, **Cristian Israel Siguenca Salazar** portador de la cédula de ciudadanía N° **0302790555**. En calidad de autor y titular de los derechos patrimoniales del trabajo de titulación “**SISTEMA DE GESTIÓN DE INVENTARIOS, MOVIMIENTOS DE MATERIAL BÉLICO Y DE INTENDENCIA A NIVEL NACIONAL PARA LA DIRMOV**” de conformidad a lo establecido en el artículo 114 Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación, reconozco a favor de la Universidad Católica de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos y no comerciales. Autorizo además a la Universidad Católica de Cuenca, para que realice la publicación de este trabajo de titulación en el Repositorio Institucional de conformidad a lo dispuesto en el artículo 114 de la Ley Orgánica de Educación Superior.

Cañar, **07 de octubre de 2022**

  
F.....

**Cristian Israel Siguenca Salazar**

**C.I. 0302790555**