

UNIVERSIDAD NACIONAL DEL ALTIPLANO
FACULTAD DE INGENIERÍA MECÁNICA ELÉCTRICA, ELECTRÓNICA Y SISTEMAS
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



“MODELO DE SISTEMA DE GESTIÓN DE VENTAS DESARROLLADO CON ARQUITECTURA ORIENTADA A SERVICIOS (SOA) PARA LA EMPRESA IMPORTADORA BICI&MOTOS EIRL, JULIACA 2012.”

TESIS

PRESENTADO POR:

MILTON LUIS FLORES CONDORI

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO DE SISTEMAS

PUNO – PERÚ

2014

UNIVERSIDAD NACIONAL DEL ALTIPLANO
FACULTAD DE INGENIERÍA MECÁNICA ELÉCTRICA ELÉCTRONICA Y SISTEMAS
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

**“MODELO DE SISTEMA DE GESTIÓN DE VENTAS DESARROLLADO CON
ARQUITECTURA ORIENTADA A SERVICIOS (SOA) PARA LA EMPRESA
IMPORTADORA BICI&MOTOS EIRL, JULIACA 2012.”**

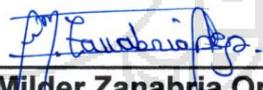
TESIS PRESENTADA POR:
MILTON LUIS FLORES CONDORI

PARA OPTAR EL TÍTULO DE:
INGENIERO DE SISTEMAS

APROBADA POR EL JURADO REVISOR CONFORMADO POR:

PRESIDENTE : 
Mg. Carlos Boris Sosa Maydana

PRIMER MIEMBRO : 
Mg. Oliver Amadeo Vilca Huayta

SEGUNDO MIEMBRO : 
Ing. Milder Zanabria Ortega

DIRECTOR DE TESIS : 
M.Sc. Edelfrè Flores Velásquez

PUNO – PERÚ

ÁREA: Administración de sistemas
TEMA: Sistemas empresariales

2014

DEDICATORIA

Quiero dedicar este trabajo

A Dios

Por haberme permitido llegar hasta este punto y haberme dado salud para lograr mis objetivos, además de su infinita bondad y amor.

A mi madre Julia.

Por haberme apoyado siempre, por sus consejos, sus valores, por la motivación que me ha permitido ser una persona de bien, pero más que nada, por su paciencia y amor.

A mi padre Gregorio.

Por los ejemplos de perseverancia y constancia que lo caracterizan, por el valor mostrado para salir adelante y por su amor.

A mis amigos.

Porque siempre estuvieron pendientes de mi progreso universitario día con día y me dieron su apoyo cuando lo necesité sin pedirme nada a cambio. Ustedes me acompañaron a lo largo de este proceso estuvieron pendientes de que todas las cosas me salieran bien y de que no tomara una mala decisión, les agradezco la confianza que depositaron en mí para poder estar siempre en las buenas y las malas juntos. Gracias por confiar y creer en mí y haber hecho de mi etapa universitaria un trayecto de vivencia que nunca voy a olvidar.

Milton.

AGRADECIMIENTO

Quiero expresar mi más sincero agradecimiento a mis padres, por todo el apoyo que me brindaron durante el desarrollo de mi carrera profesional y por enseñarme que la perseverancia y el esfuerzo son el camino para lograr objetivos.

A los Docentes de la Escuela Profesional de ingeniería de Sistemas quienes con sus enseñanzas enriquecieron mi formación como profesional.

Y de manera especial al Presidente, Miembros del Jurado, Director y Asesor de tesis. Mg. Carlos B. Sosa Maydana, Mg. Oliver A. Vilca Huayta, Ing. Milder Zanabria Ortega, M.Sc. Edelfré Flores Velásquez y al Ing. Willy S. Onque Llanque, agradezco la paciencia y capacidad de guiarme durante todo el proceso de control del trabajo de investigación.

A todos mi mayor reconocimiento y gratitud.

ÍNDICE

RESUMEN.....	15
ABSTRACT.....	16
INTRODUCCIÓN	17
CAPÍTULO I.....	20
PLANTEAMIENTO Y FORMULACIÓN DEL PROBLEMA.....	20
1.1. El Problema.....	20
1.1.1. Planteamiento del Problema.....	20
1.1.2. Definición del Problema.....	22
1.2. Justificación de la Investigación.....	22
1.3. Objetivos de la Investigación.....	24
1.3.1. Objetivo General.....	24
1.3.2. Objetivos Específicos.....	24
1.4. Hipótesis.....	25
1.5. Limitación de la Investigación.....	25
1.6. Operacionalización de Variables.....	25
CAPÍTULO II.....	27
MARCO TEORICO.....	27
2.1. Antecedentes de la Investigación.....	27
2.2. Base Teórica.....	37
2.2.1. Sistema de Gestión.....	37
2.2.2. Sistemas de Información.....	43
2.2.3. Administración de información.....	44
2.2.4. Arquitectura Orientada a Servicios (SOA).....	45
2.2.5. Consumo de Servicios.....	48
2.2.6. Servicios Web.....	49
2.2.7. XML.....	50
2.2.8. SOAP.....	51
2.2.9. WSDL.....	52
2.2.10. Modelado de servicios con SOMF.....	52
2.2.11. Rational Unified Process (RUP).....	74
2.2.12. Modelado de Datos con UML.....	85
2.2.13. C#.....	88
2.2.14. Microsoft SQL Server.....	89

2.2.15. Metodología Documentación de Requerimientos Centrada en el Usuario DoRCU.....	93
2.3. Marco Conceptual.....	95
2.3.1. Assets.....	95
2.3.2. Base de Datos.....	95
2.3.3. Diagrama de Flujo.....	95
2.3.4. Gestión.....	96
2.3.5. Multiusuario.....	96
2.3.6. Modelo.....	96
2.3.7. Proceso de negocio.....	97
2.3.8. Recursos de Información.....	98
2.3.9. Software.....	98
2.3.10. Tecnología de Información (TI).....	98
2.3.11. Usabilidad.....	99
2.3.12. Funcionalidad.....	99
2.3.13. Eficiencia.....	99
2.3.14. Satisfacción del Cliente.....	99
CAPÍTULO III.....	101
MATERIALES Y MÉTODOS.....	101
3.1. Metodología de Investigación.....	101
3.2. Diseño de la Investigación.....	101
3.3. Población y Muestra.....	102
3.3.1. Población.....	102
3.3.2. Muestra.....	102
3.4. Métodos de Recopilación de Datos.....	103
3.5. Métodos de Tratamiento de Datos.....	104
3.6. Material Experimental.....	104
3.6.1. Metodología de Desarrollo del Sistema.....	104
3.6.2. Hardware y Software de Desarrollo.....	107
CAPÍTULO IV.....	108
EXPOSICIÓN Y ANÁLISIS DE LOS RESULTADOS.....	108
4.1. Aplicación de la Metodología.....	108
4.1.1. Recopilación de Requerimientos.....	108
4.1.2. Modelo Funcional y Diseño del Sistema.....	113
4.1.3. Diseño de la Arquitectura.....	159
4.1.4. Implementación del Sistema.....	164
4.1.5. Pruebas.....	177
4.2. Prueba de Hipótesis.....	184
4.2.1. Evaluación del sistema por la empresa.....	184
4.2.2. Planteamiento de la Hipótesis.....	216

4.2.3. Nivel de Significancia.....	216
4.2.4. Zona de Rechazo.....	217
4.2.5. Estadístico de Prueba.....	217
4.2.6. Decisión.....	219
4.2.7. Resultado de la Prueba de la Hipótesis.....	220
CAPÍTULO V.....	222
CONCLUSIONES.....	222
CAPÍTULO VI.....	226
RECOMENDACIÓN Y SUGERENCIAS.....	226
BIBLIOGRAFÍA.....	229
ANEXOS.....	233



ÍNDICE DE FIGURAS

Figura 1. Sistema de Gestión.	37
Figura 2. Esquema de Gestión.	40
Figura 3. Componentes de la arquitectura SOA.....	48
Figura 4. Análisis SOMF	54
Figura 5. Símbolos del análisis de operaciones SOMF.....	56
Figura 6. Análisis contextual de las operaciones SOMF.....	57
Figura 7. Elementos de la integración de negocios orientado a servicios.....	57
Figura 8. Integración de operaciones.	58
Figura 9. Relación lógica del servicio.	60
Figura 10. Asociación de la composición orientada a servicios.....	67
Figura 11. Conectores de la actividad de transacción.....	73
Figura 12. Arquitectura general RUP.....	77
Figura 13. Fases del RUP y sus hitos.....	79
Figura 14. Elementos del diagrama para el diseño de base de datos.....	88
Figura 15. Diagrama de Caso de Uso General.....	114
Figura 16. Diagrama de Caso de Uso Gestionar Usuario.....	115
Figura 17. Diagrama de Caso de Uso Gestionar Proveedor.....	116
Figura 18. Diagrama de Caso de Uso Gestionar Sucursal.....	118
Figura 19. Diagrama de Caso de Uso Gestionar Producto.....	119
Figura 20. Diagrama de Caso de Uso Gestionar Cliente.....	121
Figura 21. Diagrama de Caso de Uso Gestionar Distribuir de Producto.....	122
Figura 22. Diagrama de Caso de Uso Gestionar Compra.....	124
Figura 23. Diagrama de Caso de Uso Gestionar Venta.....	126
Figura 24. Diagrama de Caso de Uso Gestionar Pedido.....	128

Figura 25. Diagrama de Caso de Uso Gestionar Almacén.....	131
Figura 26. Diagrama de Caso de Uso Gestionar Movimiento Comercial.....	134
Figura 27. Diagrama de Secuencia Gestionar Usuario.....	138
Figura 28. Diagrama de Secuencia Gestionar Proveedor.....	139
Figura 29. Diagrama de Secuencia Gestionar Sucursal.....	140
Figura 30. Diagrama de Secuencia Gestionar Producto.....	141
Figura 31. Diagrama de Secuencia Gestionar Cliente.....	142
Figura 32. Diagrama de Secuencia Gestionar Distribución.....	143
Figura 33. Diagrama de Secuencia Gestionar Compras.....	144
Figura 34. Diagrama de Secuencia Gestionar Venta.....	145
Figura 35. Diagrama de Secuencia Gestionar Almacén.....	146
Figura 36. Diagrama de Secuencia Gestionar Movimiento Comercial.....	147
Figura 37. Diagrama de actividades con entidades gestión de ventas.....	148
Figura 38. Diagrama de actividades con entidades gestión de compras.....	148
Figura 39. Diagrama de actividades con entidades gestión de almacén.....	149
Figura 40. Diagrama de actividades con entidades movimiento comercial.....	149
Figura 41. Modelado de objetos del negocio.....	150
Figura 42. Reglas de Negocio.....	151
Figura 43. Modelo de Base de Datos del Sistema.....	152
Figura 44. Modelo de Análisis.....	155
Figura 45. Modelo de Integración.....	156
Figura 46. Modelo de Diseño.....	157
Figura 47. Modelo de diseño lógico de la composición.....	158
Figura 48. Vista física en escenario reducido.....	160
Figura 49. Arquitectura lógica.....	161
Figura 50. Arquitectura lógica detallada.....	163

Figura 51. Ventana de Autenticación.	165
Figura 52. Ventana principal del sistema.	165
Figura 53. Ventana de productos.....	165
Figura 54. Ventana de categoría de productos.	166
Figura 55. Ventana de clientes.	166
Figura 56. Ventana de sucursales.....	167
Figura 57. Ventana de usuarios y empleados.....	167
Figura 58. Ventana de listado de ubicación.	168
Figura 59. Ventana de registro de nueva venta.	168
Figura 60. Mensaje de confirmación de registro de venta.	169
Figura 61. Ventana de listado de venta.	169
Figura 62. Conexión a la base de datos.	170
Figura 63. Prueba de conexión a la base de datos.	170
Figura 64. Adición de las tablas de la base de datos al sistema.	171
Figura 65. Mapeado de tablas requeridas.....	171
Figura 66. Inserción de la clase WCF.	172
Figura 67. Creación de los servicios mediante WCF.	172
Figura 68. Clase de la interfaz del servicio listar producto.	173
Figura 69. Clase del servicio listar producto.....	173
Figura 70. Creación del cliente del servicio.....	174
Figura 71. Referencia al servicio creado.	174
Figura 72. Selección de los servicios detectados.	175
Figura 73. Prueba de invocación del servicio.....	175
Figura 74. Contenido del servicio invocado - contrato del servicio.....	176
Figura 75. Gráfico de barras de la pregunta N° 1 - Pre test.	185
Figura 76. Gráfico de barras de la pregunta N° 2 - Pre test.	186

Figura 77. Gráfico de barras de la pregunta N° 3 - Pre test.	187
Figura 78. Gráfico de barras de la pregunta N° 4 - Pre test.	188
Figura 79. Gráfico de barras de la pregunta N° 5 - Pre test.	189
Figura 80. Gráfico de barras de la pregunta N° 6 - Pre test.	190
Figura 81. Gráfico de barras de la pregunta N° 7 - Pre test.	191
Figura 82. Gráfico de barras de la pregunta N° 8 - Pre test.	192
Figura 83. Gráfico de barras de la pregunta N° 9 - Pre test.	193
Figura 84. Gráfico de barras de la pregunta N° 10 - Pre test.	194
Figura 85. Gráfico de barras de la pregunta N° 1 – Post test.	195
Figura 86. Gráfico de barras de la pregunta N° 2 – Post test.	196
Figura 87. Gráfico de barras de la pregunta N° 3 – Post test.	197
Figura 88. Gráfico de barras de la pregunta N° 4 – Post test.	198
Figura 89. Gráfico de barras de la pregunta N° 5 – Post test.	199
Figura 90. Gráfico de barras de la pregunta N° 6 – Post test.	200
Figura 91. Gráfico de barras de la pregunta N° 7 – Post test.	201
Figura 92. Gráfico de barras de la pregunta N° 8 – Post test.	202
Figura 93. Gráfico de barras de la pregunta N° 9 – Post test.	203
Figura 94. Gráfico de barras de la pregunta N° 10 – Post test.	204
Figura 95. Gráfico de barras de comparación de la pregunta N° 1.	206
Figura 96. Gráfico de barras de comparación de la Pregunta N° 2.	207
Figura 97. Gráfico de barras de comparación de la Pregunta N° 3.	208
Figura 98. Gráfico de barras de comparación de la Pregunta N° 4.	209
Figura 99. Gráfico de barras de comparación de la Pregunta N° 5.	210
Figura 100. Gráfico de barras de comparación de la Pregunta N° 6.	211
Figura 101. Gráfico de barras de comparación de la Pregunta N° 7.	212
Figura 102. Gráfico de barras de comparación de la Pregunta N° 8.	213

Figura 103. Gráfico de barras de comparación de la Pregunta N° 9. 214

Figura 104. Gráfico de barras de comparación de la Pregunta N° 10. 215

Figura 105. Gráfico de T Student –zona de rechazo. 218



ÍNDICE DE TABLAS

Tabla 1. Operacionalización de variables.	25
Tabla 2. Versiones Microsoft SQL Server	92
Tabla 3. Etapas de desarrollo del sistema según la disciplina RUP.	106
Tabla 4. Hardware y Software.....	107
Tabla 5. Recopilación de Requerimiento	110
Tabla 6. Agrupación de requerimientos	112
Tabla 7. Especificación del Caso de Uso Gestionar Usuario.	115
Tabla 8. Especificación del Caso de Uso Gestionar Proveedor.	117
Tabla 9. Especificación del Caso de Uso Gestionar Sucursal.....	118
Tabla 10. Especificación del Caso de Uso Gestionar Producto.	120
Tabla 11. Especificación del Caso de Uso Gestionar Cliente.	121
Tabla 12. Especificación del Caso de Uso Gestionar Distribuir de Producto.	123
Tabla 13. Especificación del Caso de Uso Gestionar Compra.....	124
Tabla 14. Especificación del Caso de Uso Gestionar Venta.	126
Tabla 15. Especificación del Caso de Uso Gestionar Pedido.....	128
Tabla 16. Especificación del Caso de Uso Gestionar Almacén.	131
Tabla 17. Especificación del Caso de Uso Gestionar el Movimiento Comercial.	134
Tabla 18. Especificación de entradas y Salidas.....	179
Tabla 19. Cuadro estadístico de la pregunta N° 1 – Pre test.	185
Tabla 20. Cuadro estadístico de la pregunta N° 2 - Pre test.....	186
Tabla 21. Cuadro estadístico de la pregunta N° 3 - Pre test.....	187
Tabla 22. Cuadro estadístico de la pregunta N° 4 - Pre test.....	188
Tabla 23. Cuadro estadístico de la pregunta N° 5 - Pre test.....	189
Tabla 24. Cuadro estadístico de la pregunta N° 6 - Pre test.....	190

Tabla 25. Cuadro estadístico de la pregunta N° 7 - Pre test.....	191
Tabla 26. Cuadro estadístico de la pregunta N° 8 - Pre test.....	192
Tabla 27. Cuadro estadístico de la pregunta N° 9 - Pre test.....	193
Tabla 28. Cuadro estadístico de la pregunta N° 10 - Pre test.....	194
Tabla 29. Cuadro estadístico de la pregunta N° 1 – Post test.....	195
Tabla 30. Cuadro estadístico de la pregunta N° 2 – Post test.....	196
Tabla 31. Cuadro estadístico de la pregunta N° 3 – Post test.....	197
Tabla 32. Cuadro estadístico de la pregunta N° 4 – Post test.....	198
Tabla 33. Cuadro estadístico de la pregunta N° 5 – Post test.....	199
Tabla 34. Cuadro estadístico de la pregunta N° 6 – Post test.....	200
Tabla 35. Cuadro estadístico de la pregunta N° 7 – Post test.....	201
Tabla 36. Cuadro estadístico de la pregunta N° 8 – Post test.....	202
Tabla 37. Cuadro estadístico de la pregunta N° 9 – Post test.....	203
Tabla 38. Cuadro estadístico de la pregunta N° 10 – Post test.....	204
Tabla 39. Cuadro estadístico resumen de la encuesta Pre test.	218
Tabla 40. Cuadro estadístico resumen de la encuesta Post test.....	218
Tabla 41. Resultados de las encuesta pre test y post test	219
Tabla 42. Promedio y Desviación estándar del pre test y post test	219
Tabla 43. Nivel de aceptación en rango vigesimal.....	219
Tabla 44. Ubicación de los test de acuerdo a promedio.....	220

RESUMEN

El desarrollo de la investigación se basa en la metodología de investigación aplicada con enfoque correlacional, metodología que busca confrontar la teoría con la realidad y analizar la relación entre las variables, para tal efecto se presentó el marco teórico correspondiente al desarrollo de la investigación, acentuándose en lo referente a la arquitectura orientada a servicios (SOA), y gestión de ventas. Para el desarrollo del sistema, como punto de inicio se tuvo que obtener los requisitos de la empresa mediante la metodología de Documentación de Requerimientos Centrada en el Usuario (DORCU), para el diseño de base de datos y diseño del sistema; se utilizó Lenguaje Unificado de Modelado (LUM o UML, por sus siglas en inglés), y para el desarrollo del sistema se utilizó C# y SQL aplicando el enfoque de la arquitectura orientada a servicios (SOA). El ámbito de aplicación y experimentación de la investigación fue la empresa BICI&MOTO. La población de la investigación estuvo conformada por los trabajadores de la empresa, población a quien según el diseño de investigación se aplicó un muestreo determinístico no probabilístico para seleccionar la muestra poblacional, empleando la estadística inferencial para realizar el análisis e interpretación de resultados obtenidos. La principal conclusión a la que se llega en esta investigación es que el modelo del sistema de gestión de ventas desarrollado con arquitectura orientada a servicios contribuye a una adecuada administración de la información en la empresa importadora BICI&MOTOS EIRL.

Palabras clave: Servicios web, sistema de información, gestión de ventas, administración de información, modelado de negocio, escalabilidad, usabilidad.

ABSTRACT

The development of the research is based on the research methodology applied correlational approach, methodology aimed at confronting theory with reality and analyze the relationship between variables, for this purpose the relevant theoretical framework for the development of the research was presented, accentuating regarding service-oriented architecture (SOA), and sales management. To develop the system, as the starting point had to get the business requirements using the methodology Documentation User Centered Requirements (DORCU), for designing database and system design; used Unified Modeling Language (UML or LUM, for its acronym in Spanish), and the development system was used C # and SQL by applying the approach to service-oriented architecture (SOA). The scope of the research and experimentation was the company BICI&MOTO. The research population consisted of employees of the company, people who, according to the research design a non-probabilistic deterministic sampling was applied to select the sample population using inferential statistics for the analysis and interpretation of results. The main conclusion reached in this research is that the system model developed sales management with service oriented architecture contributes to proper management of information in the importer BICI&MOTO EIRL.

Keywords: web services, information systems, sales management, information management, business modeling, scalability, usability.

INTRODUCCIÓN

Dentro del mundo de los negocios en general, llevar el control de información de las diferentes áreas de la empresa ha ido evolucionando junto con la tecnología, debido a que la adecuada administración de la información es un factor sustancial para el éxito empresarial.

La organización de hoy busca principalmente llevar sus actividades de manera eficaz y con el menor uso de recursos económicos y humanos, lo cual es logrado, debido a los grandes avances de la tecnología que ha desarrollado importantes herramientas que permiten realizar de manera más rápida actividades y operaciones específicas o rutinarias. Entre las ventajas que ha proporcionado la tecnología en el ámbito organizacional, se encuentran principalmente la automatización de procesos, administración organizada de la información de un área o departamento, incremento en cuanto a la producción, entre otras.

El uso de las tecnologías de información y comunicación, sirve al propósito de afrontar retos y dar paso a las transformaciones que demandan nuevos cambios que involucren a equipos de trabajos bien integrados y adiestrados para dar respuesta oportuna y actualizada a las demandas de la sociedad. Además de la necesidad de resolver las necesidades del mercado, con respecto al tiempo y a la calidad de software.

Existen muchas empresas que fundamentan sus actividades de gestión comercial mediante la utilización de tecnologías de información y comunicación, la cual a su vez hace uso de diversos sistemas de información con el uso de metodologías compatibles con la aplicación del modelado de servicios que agregan valor al objetivo principal del negocio, por lo que para el buen funcionamiento del negocio se requiere de una aplicación informática que aplique una metodología compatible con el modelado de servicios.

Por otra parte, desde el punto de vista arquitectural, el paradigma de arquitectura orientada a servicios (SOA) conceptualiza el software estableciendo una separación entre las aplicaciones y los componentes de lógica empresarial, proponiendo que éstos sean diseñados como servicios independientes de la tecnología, alto nivel de abstracción, gran independencia, muy reutilizables, y un débil o nulo acoplamiento con las aplicaciones que los consumen o utilizan.

En la presente investigación se une ambos conceptos mediante el análisis, diseño y desarrollo de un sistema de ventas con arquitectura orientada a servicios, utilizando el lenguaje C# y SQL.

El presente trabajo de investigación se presenta en seis capítulos, todos ellos relacionados de manera que haya una coherencia entre las distintas partes y sea de fácil comprensión para los lectores de este trabajo de investigación.

CAPÍTULO I: Capítulo en el que se presenta el planteamiento del problema, formulación del problema, justificación de la investigación, objetivos de la investigación, estos puntos sitúan al lector dentro de la problemática de la investigación; así como de la hipótesis que será demostrada; y de la operacionalización de variables y las limitaciones de la investigación.

CAPÍTULO II: Presenta el marco teórico compuesto de tres partes antecedentes de la investigación, base teórica donde reside toda la teoría necesaria para el mejor entendimiento

del presente trabajo de investigación, y finalmente el marco conceptual necesario para comprender los términos básicos más utilizados a lo largo de la investigación.

CAPÍTULO III: Capitulo en el que se expone el método de investigación constituido por la metodología de la investigación, diseño de la investigación para explicar los métodos que constituyen el presente trabajo de investigación; así como también de la población y muestra, método de recopilación de datos, método de tratamiento de datos; también se encuentra el ítem de material aplicativo en esta sección se hace referencia a la metodología de desarrollo del sistema, y el hardware y software de desarrollo.

CAPÍTULO IV: Capitulo que presenta la exposición y análisis de resultados, aplicación de la metodología en donde se muestra el desarrollo del sistema propuesto aplicando la arquitectura orientada a servicios, puesta a prueba del mismo, esto con respecto a la situación tradicional o anterior a la implementación del sistema; también se presenta los resultados de las encuestas (pre test y post test) realizadas, las cuales ayudan a la demostración de la hipótesis planteada en esta investigación.

Conclusiones a las que se ha llegado como resultado de la implantación y puesta en marcha del sistema.

Recomendaciones y sugerencias para seguir investigando y desarrollando sistemas y trabajos de investigación a futuro.

Finalmente se presenta la bibliografía y los anexos de la investigación.

CAPÍTULO I

PLANTEAMIENTO Y FORMULACIÓN DEL PROBLEMA.

1.1. El Problema.

1.1.1. Planteamiento del Problema.

Toda organización de hoy busca principalmente llevar sus actividades de manera eficaz y con el menor uso de recursos económicos y humanos, lo que es logrado gracias a los grandes avances de la tecnología que ha desarrollado importantes herramientas que permiten realizar de manera rápida actividades y operaciones específicas o rutinarias. (Gomez Vieites & Suares Rey, 2006). Entre las ventajas que ha proporcionado la tecnología en el ámbito organizacional, se encuentran principalmente la automatización de procesos, administración organizada de la información de un área o departamento, incremento en cuanto a la producción, entre otras.

Actualmente, la mayoría de las empresas están realizando cambios es sus sistemas operacionales, a fin de mejorar y actualizar sus procesos operativos. Debido a esto, entra en juego el uso de las tecnologías de información y

comunicación TICs, con el propósito de tener una gerencia capaz de afrontar retos y dar paso a las transformaciones que demandan nuevos cambios y que involucren a equipos de trabajos bien integrados y adiestrados en el área, para dar respuesta oportuna y actualizada a las demandas de la sociedad. Además de la necesidad de resolver las complejidades del mercado, con respecto al tiempo y la calidad de software.

Existen muchas empresas que fundamentan sus actividades de gestión comercial mediante la utilización de tecnologías de información y comunicación. (Rodríguez de Soto & Cuervo Fernández, 2006). La cual a su vez hace uso de diversos sistemas de información con el uso de metodologías compatibles con el modelado de servicios que agregan valor al objetivo principal del negocio. Sin embargo existen muchas otras organizaciones que hacen contraste a esta realidad, las cuales no cuentan con una forma organizada de gestionar la información generada por sus actividades.

BICI&MOTOS EIRL - Juliaca es una empresa cuyo giro de negocio es la importación, compra, venta y distribución motos, repuestos de motos y bicicletas. Actualmente en la mencionada empresa la información se encuentra archivada en diferentes formatos de Word, Excel y cuadernos de registro, lo que dificulta la recolección y sistematización de datos que sirvan para la obtención de información que aporte al conocimiento preciso sobre la cantidad de productos y el movimiento de compra venta de los mismos.

Este tipo de almacenamiento de información genera inconvenientes como la deficiencia en el control de almacén sobre los productos, pérdida de datos por encontrarse en archivos corruptos, retraso en la distribución de productos, dificultad de no tener muchas veces la información en el lugar y tiempo adecuado, la no garantía sobre la unicidad de datos, generación de reportes e informes de gestión con información inconsistente entre otros. Por lo que para el buen funcionamiento del negocio se requiere de una aplicación informática que aplique una metodología compatible con el modelado de servicios.

1.1.2. Definición del Problema.

¿El desarrollo y la implementación del modelo de sistema de gestión de ventas desarrollado con arquitectura orientada a servicios (SOA) contribuirá a una adecuada administración de la información en la empresa BICI&MOTOS EIRL., Juliaca 2012?

1.2. Justificación de la Investigación.

A medida que pasa el tiempo se utiliza en mayor grado las Tecnologías de la Información (TI) para automatizar y dar apoyo a las actividades de una empresa. Actualmente, toda organización exitosa conoce la importancia de la implantación de los sistemas de información para el manejo y control de los datos, esto con la finalidad de dar respuesta a las necesidades de los usuarios en cuanto a servicios y veracidad de la información.

La presente investigación se dirigirá al desarrollo de un sistema orientado a la gestión de ventas en la empresa importadora BICI&MOTOS EIRL, la cual generara beneficios en cuanto a la eficiencia en los procesos y un mejor desempeño de las personas en la empresa.

En el desarrollo de este sistema se empleará la Arquitectura Orientada a Servicios (SOA), ya que es un concepto que define la utilización de servicios para dar soporte a los requisitos del negocio, cabe señalar que se utilizará esta arquitectura para lograr comprender el funcionamiento de cada servicio y su interacción, identificar los problemas de coordinación para su corrección e identificar los servicios que puedan ser reutilizados optimizando los tiempos de modelado.

Este sistema es una solución factible para la empresa, ya que al tener los datos de las operaciones comerciales almacenados de forma adecuada y en un único contenedor; la información que se obtendrá será actualizada y permitirá un control real de las existencias de la empresa, permitiendo a los trabajadores acceso rápido a la información para dar respuestas eficientes y funcionales al momento de realizar las actividades de cada proceso.

Por lo antes mencionado, la elaboración de este sistema de información permitirá a la empresa contar con un software que avalará un mejor acceso a la información, que conlleve a una buena administración de la información en la empresa. Asimismo, contribuirá al adiestramiento de sus empleados en cuanto al manejo de sistemas, lo que permitirá la obtención de resultados que contribuyan al desarrollo de la empresa.

1.3. Objetivos de la Investigación.

1.3.1. Objetivo General.

Desarrollar e implementar un modelo de sistema de gestión de ventas, empleando la arquitectura orientada a servicios (SOA) que permita una adecuada administración de la información en la empresa importadora BICI&MOTOS EIRL.

1.3.2. Objetivos Específicos.

- Determinar los requerimientos del sistema basado en los procesos que la empresa desea automatizar bajo la metodología DoRCU (Documentación de Requerimientos Centrada en el Usuario) para la ingeniería de requerimientos.
- Diseñar la arquitectura del sistema de acuerdo a los requisitos que presente la empresa.
- Desarrollar el sistema con arquitectura orientada a servicios (SOA).
- Implementar el sistema con arquitectura orientada a servicios en la empresa BICI&MOTOS EIRL.
- Realizar la validación del sistema.

1.4. Hipótesis.

El modelo del sistema de gestión de ventas desarrollado con arquitectura orientada a servicios (SOA) contribuye a una adecuada administración de la información en la empresa importadora BICI&MOTOS EIRL - Juliaca.

1.5. Limitación de la Investigación.

La presente investigación se encontró delimitada a la implementación de un sistema de ventas, desarrollo con arquitectura orientada a servicios (SOA) para administrar la información generada por los procesos de importación, ventas, compras de la empresa BICI&MOTOS EIRL, Juliaca, el sistema abarca los módulos de compra, venta, almacén, proveedores, clientes.

1.6. Operacionalización de Variables.

Tabla 1. Operacionalización de variables.

VARIABLES	DIMENSIONES	INDICADORES	ESCALA
INDEPENDIENTE			Ordinal (Likert)
Sistema Desarrollado con SOA.	Usabilidad.	- Usabilidad de Interfaz. - Nivel de complejidad de uso.	- Muy Bueno. - Bueno. - Regular.
	Funcionalidad.	- Cumplimiento de los requerimientos. - Exactitud.	- Deficiente. - Muy Deficiente.
	Eficiencia.	- Registro de datos. - Selección de datos. - Ordenación de datos. - Eliminación de datos.	De intervalo - Menos de 15 minutos. - 15 minutos. - 30 minutos. - 45 minutos. - Más de 45 minutos.
DEPENDIENTE			De intervalo
Gestión de Ventas	Consultas de usuarios.	- Consultas del movimiento de ventas. - Elaboración de Informe. - Impresión de reportes.	- Menos de 15 minutos. - 15 minutos. - 30 minutos.
	Tiempo de Espera del cliente.	- Impresión de boletas. - Consulta de productos. - Compra de producto.	- 45 minutos. - Más de 45 minutos.
	Administración de información.	- Almacenamiento de información.	Ordinal (Likert) - Muy Bueno. - Bueno. - Regular.
		- Información disponible.	- Deficiente. - Muy Deficiente.
	- Recursos para generar informes.	- Mucho. - Suficiente. - Regular. - Poco. - Muy poco.	

Fuente: Ejecutor.

CAPÍTULO II

MARCO TEORICO.

2.1. Antecedentes de la Investigación.

(Brito Urbina, 2011). **Desarrollo de un sistema, basado en Arquitectura Orientada a Servicios (SOA)**, para la automatización de los procesos de entrada y salida de equipos eléctricos, en la Gerencia de Servicios Eléctricos, PDVSA. Informe de pasantías de grado, presentado en la Universidad de Oriente Núcleo de Monagas, Venezuela, para optar al título de Ingeniero en Sistemas.

La investigación se centró en la empresa Petróleos de Venezuela Sociedad Anónima (PDVSA), dedica a la explotación, producción, refinación, petroquímica, mercadeo y transporte del petróleo venezolano.

Esta investigación se centró en la mencionada empresa con el objetivo de desarrollar un sistema de información para mejorar el proceso de Gestión de equipos y materiales, llevar el control de inventario y ser capaz de manejar de manera eficiente

las actividades que este proceso involucra, a fin de que se optimicen y se actualicen los procesos llevados a cabo por la Gerencia.

Luego de revisar la información de la investigación se desprende las siguientes conclusiones importantes como son que: El sistema realiza un gran número de funciones, entre los cuales está la generación de reportes de equipos, proveedores, almacenes y usuarios, además, se podrá verificar de forma precisa la existencia de los equipos en almacén. Y otra que el sistema genera información confiable y oportuna solicitada por la alta gerencia, lo cual facilitará el análisis y las evaluaciones de manera más eficiente al momento de realizar auditorías.

(Medina Bonilla & Pinedo Marín, 2010). **Implementación de un Sistema de Integración para las Bibliotecas Municipales de Lima y Callao utilizando SOA y J2ME.** Tesina presentada en la Universidad Nacional Mayor de San Marcos - Perú, para optar el título profesional de Ingeniero de sistemas.

La investigación aborda el tema de integración de aplicaciones entre los diferentes sistemas de bibliotecas municipales de Lima y Callao, basándose para ello de una arquitectura orientada a servicios (SOA) y como middleware de comunicación un ESB (Enterprise Service Bus), desde el cual también podrá ser accedido mediante dispositivos móviles, utilizando la plataforma J2ME (Java 2 Micro Edition). El objetivo principal de esta investigación fue el de implementar un sistema de integración para las bibliotecas municipales de Lima y Callao utilizando una arquitectura orientada a servicios donde cada biblioteca haga disponible su

información a través de un servicio web y donde se alcance la integración mediante una plataforma ESB.

De esta investigación se desprende que mediante la elaboración de un planteamiento que propone una Arquitectura Orientada a Servicios, la utilización de Web Services para definir cada servicio, la implantación de un Bus de Servicios Empresarial para llevar a cabo la integración y la posibilidad de acceder a esta solución mediante un dispositivo móvil, se puede dar solución de la problemática actual en las Bibliotecas Municipales referente a la integración. Y para las Bibliotecas municipales que no cuentan aún con un sistema de Bibliotecas, se recomienda implantar una base de datos WinISIS que es gratuito y después integrarlo al WWWISIS para que puedan hacerse búsquedas desde la WEB. De esta manera se estaría dando un previo paso para su publicación de sus servicios (WebServices), preparándose para la integración con el Sistema Integrador de Bibliotecas Municipales.

(Fernández Lino, 2009). **Desarrollo de Aplicaciones Basadas en Servicios Aplicado a Incorporar Nuevas Tecnologías.** Tesis de Grado presentado en la Universidad Mayor de San Andrés de Bolivia.

El estudio se centró en realizar un estudio y aplicación de los conceptos relacionados a las Arquitectura Orientada a servicio aplicado a incorporar nuevas tecnologías, a fin de proporcionar lineamientos para el desarrollo de aplicaciones basadas en la SOA, que incorporen este tipo de servicios. Además se presenta los detalles conceptuales referentes a la SOA, se realiza un análisis de los tipos de servicios mediante los que se puede implementar especificando el contexto de uso. Asimismo

se muestran las diferentes plataformas, herramientas, y metodología concernientes al desarrollo basado en servicios.

De esta investigación se puede desprender que a partir del trabajo realizado, quedan abiertas distintas líneas de investigación y aplicación, relacionadas con la SOA como realizar un estudio comparativo sobre las plataformas y herramientas existentes para el desarrollo de aplicaciones basadas en servicios a bien de especificar contexto de uso. Un factor muy importante para el desarrollo orientado a servicios es el diseño, haciéndose necesario realizar un estudio sobre el proceso de diseño a seguir para implementar una SOA, ver opciones de metodologías, herramientas de modelado y medición de calidad. La arquitectura permite recoger sistemas preexistentes (legados), se debe realizar un análisis dirigido a identificar las formas de implementar la incorporación de sistemas legados dentro de una infraestructura basada en servicios. Y las líneas de investigación de diversificar ampliamente, pero es importante empezar la aplicación de la arquitectura orientada a servicios en el medio, empezar a proponer proyectos que usen esta arquitectura, para dar respuesta a requerimientos potenciales demandados por el medio relacionados con integración de aplicaciones, incorporación de tecnologías, procesos y personas de forma inmediata y rápida.

(Espinosa Díaz, 2009). **Propuesta de Implantación de la Arquitectura BPM/SOA para agilizar la Gestión Comercial en la CNT.** Tesis de Grado, presentado en la Universidad San Francisco De Quito, Ecuador. Para la obtención del título de Master En Administración Estratégica De Telecomunicaciones.

La investigación se centró en la empresa CNT que es una Operadora de Telecomunicaciones ubicada en Quito.

El objetivo puntual de la tesis fue demostrar que con el uso de la arquitectura BPM/SOA, la gestión de los procesos del negocio puede ser llevada a cabo de manera más dinámica, es decir menos rígida y por lo tanto más alineada con la agilidad que “el negocio requiere”. Además se analizaron las cuatro generaciones de integración de plataformas de software, y las tecnologías de integración que existen en el mercado, que se utilizan para intercambiar información. Se analizaron sus fortalezas y debilidades, deduciendo que BPM/SOA contribuiría a fortalecer los procesos administrativos de la CNT.

De esta investigación se puede desprender las siguientes conclusiones: BPM/SOA ofrece un orden de complejidad menor, al desarrollar procesos de negocios utilizando un modelador, frente a la implantación y/o modificación de un proceso existente de un aplicativo que debe ser desarrollado cumpliendo todos los pasos y formalidades de la ingeniería de software. Y otra que BPM tiene la capacidad de cambiar los procesos del negocio de manera dinámica. SOA pese a los órdenes de complejidad de integración basados en las variables de ontología y transporte, habilita al BPM un acceso “fácil a las funcionalidades y datos de las aplicaciones. Además se plantea las siguientes recomendaciones La CNT debe implantar la arquitectura BPM/SOA para mejorar de manera muy importante su gestión comercial, debido a que la arquitectura BPM/SOA se centra más en conceptos de negocios que en tecnológicos, pues la actual no soporta de manera dinámica, ágil y flexible la entrega de productos y servicios de telecomunicaciones.

(Sagal Pazmiño, 2008). **Diseño de la SOA para Andes Petroleum LTD.** Tesis de grado presentado en la Escuela Politécnica Nacional de Ecuador. Para la obtención del título de Ingeniero en Sistemas Informáticos.

La investigación se centró en la empresa Andes Petroleum, empresa dedicada a la extracción y producción de petróleo.

El estudio abarcó el tema de diseño para el departamento de IT de esta empresa, realizando un análisis previo sobre la madurez de la SOA en la empresa para posteriormente basarse únicamente en el diseño de SOA.

De esta investigación se puede desprender las siguientes conclusiones importantes como son que: Las arquitecturas de software están orientadas a satisfacer las necesidades de operación de los negocios, considerar su crecimiento y cubrir la totalidad de los procesos de negocio, de apoyo y administrativos de una institución. La implementación de una arquitectura en cualquier organización depende de cómo la arquitectura mejore las necesidades del negocio. Y otra que una arquitectura SOA es muy buena opción para la organización debido a que el medio donde se desenvuelve siempre esta propenso a cambios bruscos que obliga a la empresa en especial al departamento de IT a resolver requerimientos de una manera ágil y con el menor costo.

(Aguilar Esteban, 2007). **Sistema Multicliente con Arquitectura SOA Basado en Tecnología .NET.** Tesis de Grado, presentado en la Universidad de Ingeniería Técnica en Informática de Sistemas de España.

La investigación se centró en la empresa PCEXPRESS, S.L. que es una compañía de España dedicada a la comercialización de productos informáticos al por mayor, que edita catálogos comerciales en papel de forma trimestral.

La investigación tiene por objeto realizar el análisis, diseño e implementación de un sistema software empresarial, concretamente un Catálogo de Productos, utilizando el paradigma arquitectural denominado SOA y la tecnología .NET de Microsoft. En esta investigación se ha desarrollado un sistema central que implementa todos los procesos de negocio relacionados con el catálogo de productos y el almacenamiento de datos en forma de servicios, cumpliendo las premisas básicas dictadas por este paradigma arquitectural.

De esta investigación se desprende las siguientes conclusiones importantes que son que: El modelo arquitectural elegido (SOA) ha resultado enormemente válido y flexible para la implementación de servicios muy reutilizables, e invocables desde aplicaciones cliente de cualquier tipo. Y otra que la metodología y planificación también han jugado un papel importante en el desarrollo del proyecto, permitiendo distribuir temporalmente distintas tareas a realizar desde el comienzo, y guiando en el cumplimiento de los hitos previstos.

(Guana Garcés, 2009). **SOA Una Aproximación a la Arquitectura de Software para Modelar Procesos de Negocio**, artículo presentado en la Universidad de los Andes de Colombia.

En este trabajo se muestra cómo una arquitectura de software SOA permite desarrollar soluciones informáticas de una manera mucho más cercana con las arquitecturas empresariales y con los modelos operativos de las organizaciones actuales, apoyando y representando directamente los procesos de negocio mediante soluciones informáticas que generen plataformas para la ejecución de los negocios de hoy en día.

De este trabajo se puede desglosar las siguientes conclusiones: La arquitectura de software orientada a servicios nos permite diseñar soluciones informáticas que representan fielmente la arquitectura empresarial y el modelo operativo de una organización determinada, desde el comienzo del ciclo de vida del diseño y desarrollo de los servicios que componen estas soluciones están inmersas decisiones de modelaje que involucran en todo sentido los elementos mencionados anteriormente (arquitectura empresarial y el modelo operativo), lo que consecuentemente permite un mejor entendimiento de la aplicación y el desarrollo de un sistema de soporte TI que posee una construcción más lógica y entendible en términos de un alto grado de abstracción organizacional, esto último hace referencia a como el desarrollo a través de SOA cierra la brecha entre el diseño y construcción del mundo virtual de las soluciones informáticas y el de las organizaciones reales sobre las que se implantan.

(Caicedo, Bustos R., & Rojas Diaz, 2008). **Integración de Procesos Utilizando la Arquitectura Orientada a Servicios SOA**, artículo presentado en la Universidad Tecnológica de Colombia.

De esta investigación se desprende las siguientes observaciones importantes con son que: “La integración y coordinación de procesos puede verse simplificada utilizando Arquitecturas de integración como SOA y Web Services.” (Krafzig, Banke, & Slama, 2005). De forma general se puede considerar que un servicio, ya sea del negocio, software o Web, está basado en una estructura sencilla donde existen dos partes. Una realiza solicitudes de servicio y otra representa el proveedor del servicio, el cual recibe las peticiones, las procesa y envía las respuestas adecuadas. Esto se lleva a cabo de forma ordenada mediante el paso de mensajes entre el solicitante y el proveedor. El concepto de servicio del negocio puede entenderse como un mecanismo básico que facilita la organización y la ejecución de los intercambios habituales del negocio. La arquitectura orientada a servicios o SOA permite interconectar y acceder a los servicios a través de una interfaz común independientemente de la tecnología con la que se ha desarrollado el servicio y de su localización.

De esta investigación se puede llegar a las siguientes conclusiones: SOA es una evolución, que permite integrar diferentes tecnologías, buscando interoperabilidad, calidad, control, eficiencia y seguridad; esta arquitectura propone soluciones para construir sistemas capaces de independizar la tecnología del servicio. Todo esto se puede lograr desarrollando un plan estratégico bien ejecutado que permita alinear los recursos de IT de forma más directa con los objetivos de negocio, logrando así una mayor integración con los clientes y proveedores. SOA requiere formación para llevarse a cabo, implica cambios y avances, se necesita cultura frente al cambio, metodología e indicadores del mismo para que sea todo un éxito. Y otra que un proyecto SOA se concibe desde le necesidad de mejorar, no es algo puntual, ya que

afecta en sus tiempos y medidas a toda la organización, se aborda desde la visión de servicios.



2.2. Base Teórica.

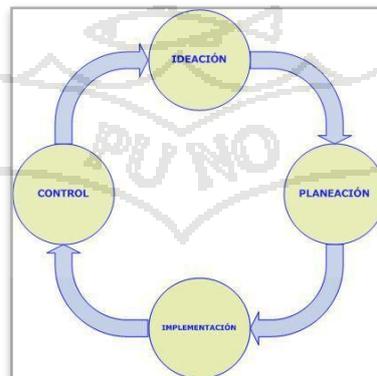
2.2.1. Sistema de Gestión.

Un Sistema de Gestión es un conjunto de etapas unidas en un proceso continuo, que permite trabajar ordenadamente una idea hasta lograr mejoras y su continuidad.

Se establecen cuatro etapas en este proceso, que hacen de este sistema, un proceso circular virtuoso, pues en la medida que el ciclo se repita recurrente y recursivamente, se logrará en cada ciclo, obtener una mejora. Estas cuatro etapas del sistema de gestión son:

- A. Etapa de Ideación.
- B. Etapa de Planeación.
- C. Etapa de Implementación.
- D. Etapa de Control.

Figura 1. Sistema de Gestión.



Fuente: HAX A. y MAJLUF N. Estrategias para el liderazgo competitivo.

A. Etapa de Ideación.

El objetivo de esta etapa es trabajar en la idea que guiará los primeros pasos del proceso de creación que se logra con el sistema de gestión propuesto.

Existen varias metodologías para lograr refinar la idea. Sin embargo, se recomienda una muy práctica.

B. Etapa de Planificación.

Dentro del proceso, la planificación constituye una etapa fundamental y el punto de partida de la acción directiva, ya que supone el establecimiento de sub-objetivos y los cursos de acción para alcanzarlos.

En esta etapa, se definen las estrategias que se utilizarán, la estructura organizacional que se requiere, el personal que se asigna, el tipo de tecnología que se necesita, el tipo de recursos que se utilizan y la clase de controles que se aplican en todo el proceso.

Si bien es cierto que el proceso de planificación depende de las características particulares de cada organización, tal como señalan (Hax & Majluf, 1997, p. 51), dentro de cualquier proceso formal de planificación, existen tres perspectivas básicas comunes: la estrategia corporativa, de negocios y funcional.

El proceso de planificación contiene un número determinado de etapas que hacen de ella una actividad dinámica, flexible y continua. En general, estas etapas consideran, para cada una de las perspectivas mencionadas, el examen del medio externo (identificación de oportunidades y amenazas), la evaluación interna (determinación de fortalezas y debilidades), y concluye con la definición de una postura competitiva sugerida (objetivos y metas).

A nivel corporativo, se obtienen como resultado las directrices estratégicas y los objetivos de desempeño de la organización. Además, se determina la asignación de recursos, la estructura de la organización (que se necesita para poner en práctica exitosamente la estrategia definida), los sistemas administrativos y las directrices para la selección y promoción del personal clave.

A nivel de negocios y funcional, los resultados se enmarcan en propuestas de programas estratégicos de acción y programación de presupuestos. Estas propuestas son, finalmente, evaluadas y consolidadas a nivel corporativo.

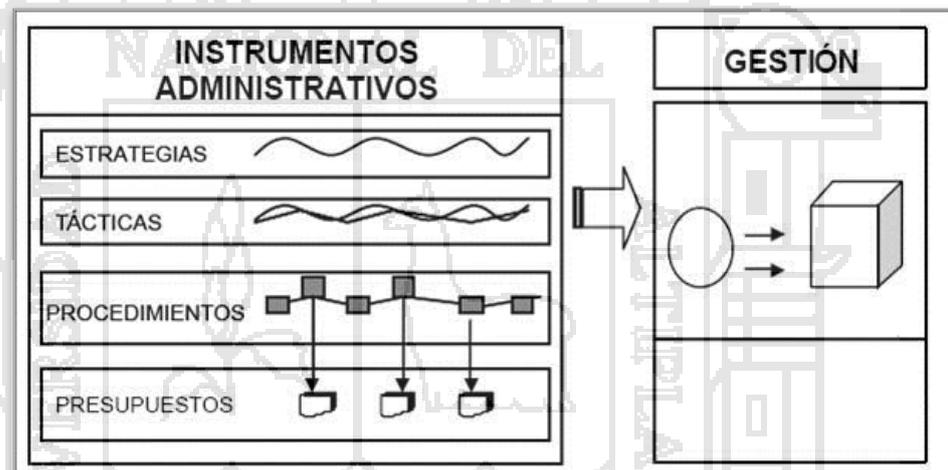
C. Etapa de Implementación o Gestión.

En su significado más general, se entiende por gestión, la acción y efecto de administrar. Pero, en un contexto empresarial, esto se refiere a la

dirección que toman las decisiones y las acciones para alcanzar los objetivos trazados.

Es importante destacar que las decisiones y acciones que se toman para llevar adelante un propósito, se sustentan en los mecanismos o instrumentos administrativos (estrategias, tácticas, procedimientos, presupuestos, etc.), que están sistémicamente relacionados y que se obtienen del proceso de planificación.

Figura 2. Esquema de Gestión.



Fuente: Gonzalo Vergara.

D. Etapa de Control.

Para este concepto se han desarrollado varias definiciones a lo largo de su evolución, sin embargo, todas se centran en la siguiente idea general: “El control es una función administrativa, esencialmente reguladora, que permite verificar (o también constatar, palpar, medir o evaluar), si el elemento seleccionado (es decir, la actividad, proceso, unidad, sistema,

etc.), está cumpliendo sus objetivos o alcanzando los resultados que se esperan”. (Cabrera, 2005).

Es importante destacar que la finalidad del control es la detección de errores, fallas o diferencias, en relación a un planteamiento inicial, para su corrección y/o prevención. Por tanto, el control debe estar relacionado con los objetivos inicialmente definidos, debe permitir la medición y cuantificación de los resultados, la detección de desviaciones y el establecimiento de medidas correctivas y preventivas.

Las etapas básicas del control.

- **Establecimiento de los estándares para la medición.**

Un estándar es una norma o criterio que sirve como base para la evaluación o comparación. Los estándares, deben ser medidas específicas de actuación con base en los objetivos. Son los límites en los cuales se debe encuadrar la organización. Se pueden definir, entre otros, estándares de cantidad, calidad, tiempo y costos.

- **Medición del desempeño.**

Tiene como fin obtener resultados del desempeño para su posterior comparación con los estándares definidos. Luego, es posible detectar si hay desvíos o variaciones en relación a lo esperado.

- **Detección de las desviaciones en relación al estándar establecido.**

Conocer las desviaciones de los resultados es la base para conocer las causas de éstas. Todas las variaciones que se presenten, en relación con los planes, deben ser analizadas detalladamente para conocer las causas que las originaron. Analizar las razones que dieron origen a las variaciones permite eficiencia y efectividad en la búsqueda y aplicación de soluciones.

- **Determinación de acciones correctivas y preventivas.**

Se determinan las acciones correctivas para corregir las causas de las desviaciones y orientar los resultados al estándar definido. Esto puede significar cambios en una o varias actividades, sin embargo, cabe señalar que podría ser necesario que la corrección se realice en los estándares originales, en lugar de las actividades.

En términos preventivos, es importante considerar que lo más significativo es encontrar maneras constructivas que permitan que los

resultados finales cumplan con los parámetros definidos (anticiparse), y no tan sólo en identificar y corregir los errores pasados.

2.2.2. Sistemas de Información.

“Un sistema de información puede definirse técnicamente como un conjunto de componentes interrelacionados que permiten capturar, procesar, almacenar y distribuir la información para apoyar la toma de decisiones y el control de una institución. Además, para apoyar a la toma de decisiones, la coordinación y el control, los sistemas de información pueden también ayudar a los administradores y al personal a analizar problemas, visualizar cuestiones complejas y crear nuevos productos.” (Laudon & Laudon, 2004).

Para estos autores los tipos de sistemas de información se clasifican en:

A. Sistemas de nivel operativo:

Son sistemas de información que hacen el seguimiento de las actividades y las transacciones elementales de la empresa.

B. Sistemas de nivel de conocimientos.

Son sistemas de información en los que se apoyan los trabajadores del conocimiento y de la información de una institución.

C. Sistemas de nivel gerencial.

Son sistemas de información en los que se apoya el seguimiento, control y toma de decisiones y las actividades administrativas de los administradores de nivel medio.

D. Sistemas de nivel estratégico.

Son sistemas de información que apoyan las actividades de planeación a largo de plazos de los niveles de dirección de la institución.

2.2.3. Administración de información.

La administración de Información en esta era Tecnológica está caracterizada por una dualidad. Por un lado, la tecnología puede ser aplicada para automatizar operaciones de acuerdo a una lógica que poco ha cambiado de un sistema del siglo XIX. Suplantar el cuerpo humano con Tecnología que habilita el mismo proceso con mayor continuidad y control. Por otro lado, la misma Tecnología genera simultáneamente Información sobre el proceso productivo y administrativo a través de la cual la organización logra su trabajo. Ofrece un mayor nivel de transparencia y profundidad sobre las actividades que habían sido parcialmente o completamente opacas. De esta forma la administración de Información con Tecnología sobrepasa la lógica tradicional de automatización. (Zuboff).

2.2.4. Arquitectura Orientada a Servicios (SOA).

La Arquitectura Orientada a Servicios (SOA), es un concepto de arquitectura de software que define la utilización de servicios para dar soporte a los requisitos del negocio. Permite además, la creación de sistemas de información altamente escalables que reflejan el negocio de la organización, y que a su vez brindan de una forma bien definida la exposición e invocación de servicios (de forma común, pero no exclusiva de los servicios web), lo cual facilita la interacción entre diferentes sistemas propios o de terceros (Juric, Loganathan, Sarang, & Jennings, 2007).

SOA proporciona una metodología y un marco de trabajo para documentar las capacidades de negocio y puede dar soporte a las actividades de integración y consolidación.

Los conceptos de software probablemente contribuirán a otra capa de los entornos computacionales complejos, un demandante de recursos para el apoyo de presupuestos. La arquitectura orientada a servicios (SOA) ayuda a resolver los problemas de la interoperabilidad, reutilización, y otros problemas asociados con este paradigma. La visión de SOA también se ocupa de los retos de software fuertemente acoplados y clama por una arquitectura que se basa en el acoplamiento de los “assets”.

SOA ayuda a la reducción del tiempo de salida al mercado y la agilidad del negocio. De hecho, la lista de ventajas sigue creciendo. El modelado orientado a

los servicios proporciona mecanismos que nos permitan concebir productos de software que hemos ido construyendo, adquiriendo, y a la integración en las últimas décadas como un servicio orientado a componentes. Es importante que deban de ser tratadas por igual la parte de análisis, diseño, arquitectura y deben ser reconocidos como servicios (Bell, 2008).

El modelado orientado a servicios es una práctica de desarrollo de software que utiliza las disciplinas de modelado y el lenguaje UML para ofrecer soluciones estratégicas y tácticas a los problemas de la empresa. Este paradigma de modelado es una visión integral del análisis, diseño y arquitectura de software de todas las entidades de la organización, concebirlos como un servicio orientado a los “assets”.

SOA ofrece la posibilidad de llamar a un componente de negocios desarrollado en una plataforma X, desde una aplicación corriendo en cualquier plataforma, en cualquier parte del mundo, utilizando para ello protocolos estándar como SOAP, XML y HTTP. (Juric, Loganathan, Sarang, & Jennings, 2007).

La programación orientada a servicios es un complemento de la programación orientada a objetos e implementa mejoras a esta última, producto de la experiencia acumulada en la última década, sobre todo en las áreas de la computación distribuida, instalación de una solución e interoperabilidad entre sistemas.

SOA, a diferencia de las arquitecturas de objetos distribuidos como J2EE, refleja más fielmente los procesos y relaciones del mundo real; es decir, SOA representa una manera más simple y natural de modelar y construir software que soluciona problemáticas de negocios del mundo real (Microsoft Corporation, 2006).

Los sistemas orientados a servicios, son diseñados con un bajo nivel de acoplamiento que facilita la implementación de cambios y estos servicios pueden ser desarrollados en cualquier lenguaje corriendo en diferentes plataformas.

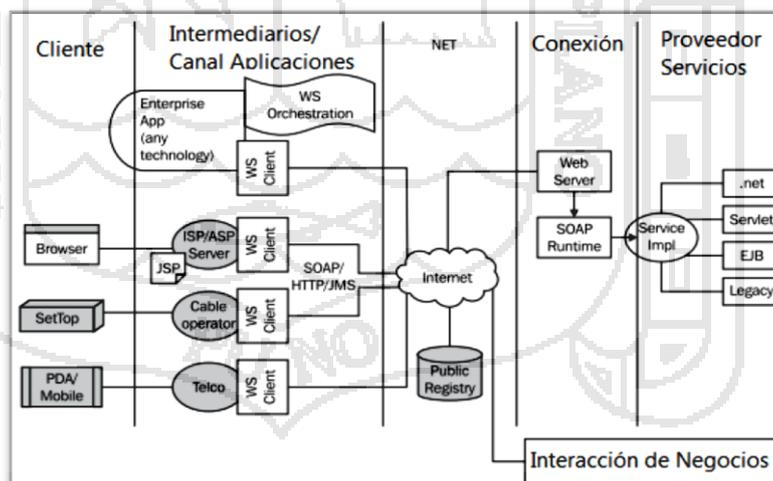
Desde el punto de vista de un usuario, la diferencia entre utilizar servicios y utilizar componentes tradicionales es imperceptible. Desde el punto de vista arquitectónico, en cambio, podemos decir que los servicios a diferencia de los componentes son autónomos, encapsulan sus propios datos de la aplicación.

Estos servicios pueden ser generados por distintos equipos de desarrollo, en diferentes tiempos, plataformas y espacios; es decir, que una aplicación puede ir creciendo y aumentando su funcionalidad a medida que se construyan nuevos servicios que no necesariamente deben estar bajo el control del equipo de desarrollo de la aplicación, por lo que es esencial que entre nuestras aplicaciones y los servicios que ellas consumen, sean débilmente acopladas.

Los servicios son utilizados por una aplicación cliente que les envía mensajes respetando un determinado contrato de servicios, pero internamente esos servicios utilizan los conceptos de la Programación Orientada a Objetos (OOP) (Microsoft Corporation, 2006).

SOA tiene dos partes clave: **servicio** y **arquitectura**. El servicio es esencialmente la vista hacia el exterior de las aplicaciones dentro de la organización TI (tecnologías de la información), donde cada aplicación proporciona los "servicios empresariales" necesarios para el acceso de otras aplicaciones. La "arquitectura" es el enfoque de toda la organización para "utilizar" los servicios. Ya que hay múltiples aplicaciones y soluciones dentro de la organización, tiene que existir una decisión de como proveer un espacio de aplicación y solución que abarca múltiples aplicaciones. Para ello utilizan los servicios expuestos por cada aplicación, y para la estandarización mediante una infraestructura proporcionar y consumir estos servicios, y los servicios que se consumen a través de un proceso basado en frameworks, todo estaría compuesto por SOA "arquitectura" (Juric, Loganathan, Sarang, & Jennings, 2007)

Figura 3. Componentes de la arquitectura SOA



Fuente: (Juric, Loganathan, Sarang, & Jennings, 2007).

2.2.5. Consumo de Servicios.

Después de que los servicios están disponibles en la plataforma SOA existen dos escenarios de uso clave:

A. Consumir los servicios desde una aplicación cliente.

Al consumir, nos referimos a utilizar el servicio, es decir invocar el servicio en cualquier otra aplicación.

B. Composición de los servicios en los procesos de negocio.

El otro método más popular para el uso de los servicios es en los procesos de negocio, por la orquestación de los servicios. Esto consiste esencialmente en la orquestación "encadenar" los servicios disponibles en el dominio para llevar a cabo una función de negocios o procesos. Este es un enfoque de programación relativamente de alto nivel, sus construcciones de flujo de programación pueden definir gráficamente el proceso, tanto como el dibujo de un diagrama de flujo o un diagrama de actividad con UML.

2.2.6. Servicios Web.

Los Servicios web son una innovación en tecnología de la World Wide Web. Un servicio web es un programa de aplicación basado en la web con una interfaz definida que acepta y procesa las solicitudes y devuelve una respuesta al solicitante. Un servicio web no está directamente ligado a una aplicación específica. En algunos aspectos, un servicio web es similar a un modelo cliente-servidor, donde el servicio web es un servidor. Los servicios web se basan en un conjunto de estándares de comunicación, como son XML para la representación

de datos, SOAP (Simple Object Access Protocol) para el intercambio de datos y el lenguaje WSDL (Web Services Description Language) para describir las funcionalidades de un servicio web (Microsoft Corporation, 2006).

2.2.7. XML.

eXtensible Markup Language (XML) recientemente ha ganado mucha notoriedad como una solución tecnológica para el intercambio de mensajes. Se puede utilizar con ventaja para muchos tipos de aplicaciones tales como el movimiento de datos entre los sistemas de negocio o describir una transacción de negocios, una solicitud para persistir o almacenar datos en una base de datos, o una solicitud de servicio.

XML es en realidad una forma de metadatos descriptivos. Por sí solo, XML es una sintaxis para la aplicación de los contenedores de datos descriptivos de un documento, transacción o mensaje. Los esquemas para ampliar las capacidades de los metadatos XML, se efectúa mediante la definición de reglas y restricciones de las características de los datos, tales como la estructura, las relaciones, los valores permitidos, y los tipos de datos. Cuando la información empresarial está contenida en un documento XML o de la transacción y se comparte con otros sistemas empresariales o, posiblemente, con los socios externos de la empresa, podemos ver rápidamente la necesidad de aplicar prácticas eficaces de la arquitectura de datos.

En muchos casos, XML se utiliza para describir el contenido de un documento sencillo. Sin embargo, las transacciones y datos de los mensajes orientados a servicios ayudan a ofrecer oportunidades, aún mayores para las aplicaciones con XML. También es importante XML para describir el contenido de datos y esquemas, que incluye las reglas granulares de metadatos que se aplican a un documento de referencia a XML.

2.2.8. SOAP.

SOAP (Simple Object Access Protocol) proporciona un mecanismo para el intercambio de información estructurada en un entorno descentralizado y distribuido usando XML. SOAP no define en sí mismo ninguna semántica de la aplicación como un modelo de programación, sino que define un mecanismo simple para expresar la semántica de aplicaciones, proporcionando un modelo de paquetes y los mecanismos de codificación dentro de los módulos de datos. Esto permite a SOAP, ser utilizado en una gran variedad de sistemas que van desde sistemas de mensajes a RPC (*Remote Procedure Call*, Llamada a Procedimiento Remoto) (Juric, Loganathan, Sarang, & Jennings, 2007). SOAP consta de tres partes:

- La envoltente SOAP define la construcción de un marco general para expresar lo que está en un mensaje, que debe lidiar con él, y si es opcional u obligatorio.

- Las reglas de codificación SOAP, define un mecanismo de serialización que puede ser utilizado para el intercambio de ejemplos de tipos de datos definidos por la aplicación.
- La representación SOAP RPC, define una convención que se puede utilizar para representar llamadas a procedimientos remotos y respuestas.

2.2.9. WSDL.

WSDL (Web Services Description Language), es el idioma más común de describir los servicios (MacVittie, 2007). WSDL está escrito como un documento XML, WSDL tiene dos partes fundamentales:

- **Interfaz:** la lista de operaciones en el servicio y el contrato de cada operación.
- **El contrato:** incluye la descripción del número y tipo de entrada y salida de la operación.
- **Bindings:** los detalles específicos en donde el servicio se encuentra y el protocolo para acceder al servicio.

2.2.10. Modelado de servicios con SOMF.

Service Oriented Modeling Framework (SOMF), es un modelo impulsado por una metodología moderna de la ingeniería cuya disciplina específica de lenguaje de modelado y las mejores prácticas se centran en el diseño de software en la arquitectura de distintas actividades, empleadas durante diferentes etapas del ciclo de desarrollo de software. Por otra parte, los arquitectos, analistas, modeladores, desarrolladores y administradores de SOMF lo emplean para hacer frente a la arquitectura empresarial, arquitectura de aplicaciones, arquitectura orientada a servicios (SOA), y los desafíos de la computación en la nube para la organización. El marco, escrito por Michael Bell, proporciona una notación independiente de la tecnología que fomenta una visión integral de las entidades de software empresarial (Bell, 2008).

- Modelo de análisis orientado a servicios.
- Modelo de integración de negocios orientada a servicios.
- Modelo de diseño lógico orientada a servicios.

2.2.10.1. Análisis Orientado a Servicios.

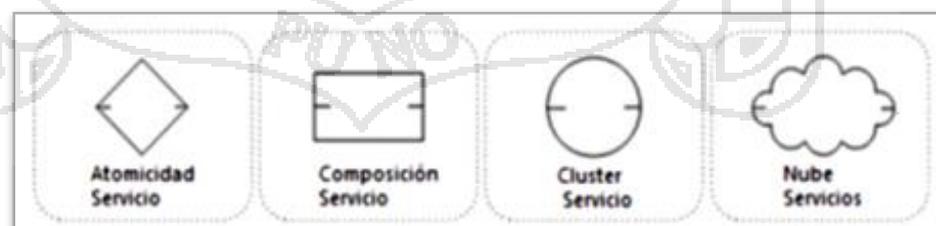
Un servicio se clasifica por sus atributos contextuales y estructurales:

- **Atomicidad del servicio:** Un componente de software indivisible que es muy granular y ejecuta menos funciones técnicas del negocio. Una formación atómica es también una pieza de software que normalmente no está sujeta a las actividades de análisis de descomposición y sus

negocios o funcionalidad tecnológica, no justifica la ruptura de componentes más pequeños.

- **Composición del servicio:** Un servicio compuesto, agrega estructuras más pequeñas y servicios de grano fino. Esta formación se caracteriza por el servicio jerárquico conocido como de grano grueso, entidad que abarca más los procesos de negocio o técnicos. Un servicio compuesto puede agregar servicios atómicos o de otros tipos.
- **Clúster de servicios:** Se trata de un conjunto de servicios distribuidos y relacionados que se han reunido a causa de su relación comercial o tecnológica común. Un grupo de servicios a los afiliados de los servicios y combina su oferta para resolver un problema de negocio. Una estructura de grupos puede agregar formaciones atómicas, así como compuestos.
- **Nube de servicios:** Un conjunto de servicios que se entregan mediante una aplicación computacional en la nube.

Figura 4. Análisis SOMF



Fuentes: Bell, Michael. Service Oriented Modeling.

i. **Análisis de Operaciones (Notaciones de las operaciones).**

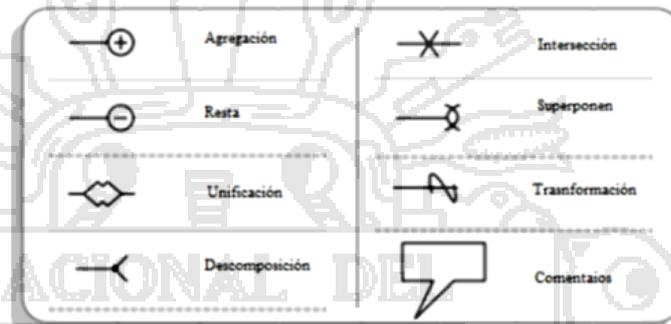
Existen los siguientes símbolos de operaciones que pueden ser utilizados para representar una propuesta de solución en un diagrama de análisis de servicios. Estos iconos muestran las actividades que han ocurrido, describen el proceso por el cual los servicios se transforman para hacer frente al dominio del problema. La notación propuesta profundiza en las actividades de análisis que se llevaron a cabo y describe el estado actual de los servicios y su contribución a la solución global.

Los símbolos se describen a continuación:

- **Agregación:** Muestra de contención de los servicios de la composición de Servicios o Clúster.
- **Resta:** se retira un servicio.
- **Unificación:** Se une a los servicios mediante la creación de un nuevo servicio.
- **Descomposición:** Separa un servicio de la matriz que lo contenía. Creando un servicio más amplio “compuesto”. A diferencia de la resta de los servicios, los activos descompuestos siguen siendo valiosos para la organización y los ambientes en los que se operan.
- **Intersección:** Intersección de dos o más grupos de servicios.
- **Superponen:** Identifica la región de superposición entre dos o más grupos de servicios.

- **Transformación:** Convierte una estructura de servicio a otra formación por ejemplo, desde atomicidad de servicios a composición de servicios.
- **Comentarios:** Un lugar para poner comentarios al lado de cada activo u operación.

Figura 5. Símbolos del análisis de operaciones SOMF.



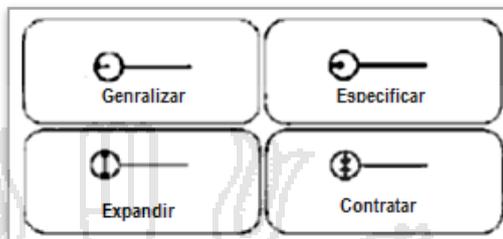
Fuente: Bell, Michael. Service Oriented Modeling.

ii. Análisis contextual de las operaciones SOMF:

- **Generalizar:** Aumenta el nivel de servicio de la abstracción y se amplía la oferta de servicios.
- **Especificar:** Disminuye el nivel de abstracción del servicio y los límites de las ofertas de los servicios.
- **Contratación:** Restringe las operaciones de servicio en el entorno distribuido.
- **Expandir:** Amplía las operaciones de servicio en un entorno distribuido.

A continuación se muestra la notación del análisis contextual de las operaciones SOMF (Bell, 2008).

Figura 6. Análisis contextual de las operaciones SOMF.



Fuente: Bell, Michael. Service Oriented Modeling.

2.2.10.2. Integración de Negocios Orientada a Servicios.

Se debe utilizar una notación de la Integración de negocios orientada a servicios. Es decir un conjunto formal de símbolos que ofrece un lenguaje de integración.

Estos iconos describen los assets del negocio que participan en la integración y representan una serie de operaciones que se pueden utilizar para describir la integración de Negocios.

Figura 7. Elementos de la integración de negocios orientado a servicios.



Fuente: Bell, Michael. Service Oriented Modeling.

i. Notación de la Integración de Operaciones.

Hay seis símbolos de integración que deben ser empleados para describir una iniciativa empresarial orientada a los servicios de integración. Estas actividades deben ser representadas en un esquema de integración. También es imprescindible para identificar los pasos y los diferentes procesos por los cuales se llegó a la propuesta de integración final de los casos. Por lo tanto, este esquema debe registrar el estado actual o futuro de la integración orientada a servicios.

Figura 8. Integración de operaciones.



Fuente: Bell, Michael. Service Oriented Modeling.

Se deben considerar los símbolos de integración después de la operación, que pueden facilitar esta documentación.

- **Integración.** Este símbolo identifica una actividad de integración que ha llevado a cabo entre un servicio y una estructura de dominio o entre un servicio y una perspectiva contextual.

- **Desintegración.** Las actividades de integración también se pueden invertir, es decir, un servicio puede ser retirado de un entorno de dominio por razones estratégicas o tácticas. Por lo tanto, el símbolo identifica como una reversión de las actividades para preservar el proceso por el que la integración se ha producido.

- **Contenido.** La actividad de contención tiene lugar entre los dominios para formar un dominio estructurado jerárquico de capas. Este símbolo también puede ser utilizado para incluir un dominio en una capa del negocio. Esto es similar al símbolo de la agregación de servicios. Sin embargo, aquí los elementos estructurales de negocios están conectados.

- **Separados.** Este símbolo se utiliza para la separación de una capa de dominio de una estructura jerárquica, la formación de capas de dominio, o para quitar un dominio de su negocio que abarca ciertos niveles.

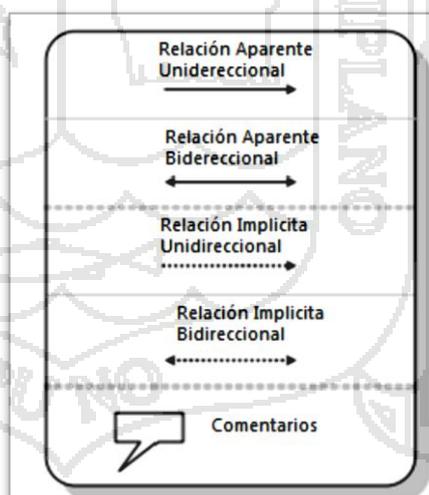
- **Perspectivas.** Este icono identifica una perspectiva de la arquitectura empresarial contextual que se asocia a un dominio de negocio.

- **Comentario.** Se utiliza este icono para describir las acciones de integración o un ámbito empresarial que requieren notas para ahondar en el estado de integración.

ii. Notación Formal de la Relación Lógica del Servicio.

Notación de la relación lógica. Consta de cuatro conectores que hacen posible transmitir una relación aparente o implícita entre los assets de software orientada a servicios, tales como servicios, e incluso los consumidores. Estos símbolos se utilizan para ilustrar las rutas de mensajes de datos intercambiados y la información sobre una red.

Figura 9. Relación lógica del servicio.



Fuente: Bell, Michael. Service Oriented Modeling.

La forma general de un conector de relación de servicio es una línea que indica la dirección e identifica la visibilidad de los servicios. Hay dos categorías de los conectores de relación de servicios.

- **Relación Aparente:** Esto denota una relación visible entre dos servicios o entre un servicio y el consumidor. El término "aparente", describe una ruta del mensaje que vincula directamente las entidades, con la información contenida y que establece la comunicación entre 2 partes, sin hacer uso de intermediarios o servicios de mediación.

- **Relación Implícita:** Este grupo muestra en un icono la relación de servicio invisible. La relación pertenece oculta a las asociaciones indirectas que emplean los agentes para el mensaje de entrega. Por lo tanto, una relación de servicio implícita se forma cuando los intermediarios, los proxys de servicios, o centros de servicios se encuentran entre los consumidores y servicios o entre los servicios.

La dirección de paso de mensajes es también un aspecto importante de esta notación. Hay dos tipos de direcciones de enrutamiento de mensajes a tener en cuenta:

- **Relación Unidireccional.** Una relación de servicio unidireccional identifica un solo sentido de ese mensaje, los mensajes se entregan en una sola dirección. No hay respuesta necesaria. Este tipo de actividad se produce normalmente cuando las partes involucradas en el intercambio de mensajes desea

reducir al mínimo el tráfico de red o simplemente porque un mensaje de respuesta no es necesario.

- **Relación Bidireccional.** Este método de entrega de mensajes denota una conversación típica bidireccional entre un consumidor y un servicio, o entre dos servicios. Esto es similar a los protocolos comunes de comunicación petición/ respuesta. El último emisor siempre espera una respuesta.

Por último, puede utilizarse el icono de comentario, que ayuda a explicar la relación de la naturaleza del servicio.

iii. Roles en el Contexto de Diseño Orientado a Servicios.

El paradigma orientado a servicios presenta tres funciones principales que pueden involucrarse en las decisiones de diseño en términos de enrutamiento de mensajes, la visibilidad, la sincronización de mensajes, y la colaboración de los servicios. Los roles de servicio en una disciplina de diseño orientado al servicio no sólo debe ser coherente, sino que debe estar ligado a un contrato que se establece por encima de cualquier utilización de los servicios.

- **Rol de los Consumidores.** Un cliente es una entidad de software orientada al servicio que está diseñada para adquirir los servicios. Es decir, normalmente las implementaciones de software que no proporcionan servicios en sí mismos. Pero en las soluciones de diseño complejo, el papel de un consumidor que también se puede

aplicar a un servicio. En el contexto de diseño orientado a servicios, un consumidor puede comunicarse con uno o más servicios. También se puede mantener una relación implícita o aparente, unidireccional o bidireccional con sus correspondientes servicios.

El permiso para acceder a un servicio particular debe ser otorgado en un contrato vinculante que puede dar el seguimiento y cumplimiento en tiempo de ejecución. Los consumidores también deben comprometerse al servicio de consumo con ciertos límites y restricciones de disponibilidad del servicio. El consumo es generalmente medido por el volumen de mensajes intercambiados y la frecuencia de la información conforme a lo acordado en el contrato. La disponibilidad, en el servicio, por otra parte, tiene restricciones en el tiempo de acceso impuestas a un consumidor.

- **Rol de Servicio.** Un servicio es una entidad que se compromete a su oferta a través de un contrato vinculante a sus consumidores suscritos. Este acuerdo estipula por lo general lo que está presente en la disponibilidad de un servicio, las tasas de consumo, y el tiempo de respuesta. Además, los servicios pueden mantener una implícita relación ya sea unidireccional y bidireccional con los correspondientes consumidores y servicios.

Como se mencionó anteriormente en la sección de rol del consumidor, un servicio también puede actuar como un consumidor. Imaginemos un servicio que no sólo está obligado a servir a su comunidad de consumo, sino también tiene la obligación de intercambiar mensajes con los servicios.

- **Rol del Intermediario.** Cualquier software orientado a servicios se encuentra entre los consumidores para facilitar la comunicación es considerado como un intermediario.

2.2.10.3. Diseño Orientado a Servicios.

Es más que una formación, una forma visual, un paquete de software que suele ser moldeada por patrones predefinidos, se compone de software orientado a servicios que se utilizan para comunicar una solución del problema que se plantee. Esto no es sólo un método táctico para proporcionar una solución a una preocupación de la organización, sino también un plan estratégico para resolver problemas similares en el futuro. La composición del diseño es el resultado de tres aspectos importantes que contribuyen: asociaciones de servicio, las rutas de intercambio de mensajes, y los patrones generales de los servicios que ofrecen una solución. Ahora cada uno de estos contribuyentes se inspecciona para llevar a una mejor comprensión de la esencia de este esfuerzo de embalaje orientada a servicios.

i. Asociaciones de Servicio.

Durante el ciclo de vida orientado a servicios, se encontraron dos tipos de relaciones de servicio: conceptual y lógico. Estas categorías de asociación de servicios influyen en la construcción de una composición de servicios: En primer lugar, en la fase de la conceptualización de servicios, comúnmente los servicios y las diferencias se identifican en forma de negocio o vínculos tecnológicos entre los servicios que participan en una solución, los cuales son conocidos como afiliaciones conceptuales. En segundo lugar, más adelante en la fase de diseño de servicios, las relaciones lógicas fueron descubiertas entre los servicios que son impulsados por la distribución de mensajes y las necesidades de cambio.

ii. Rutas de Intercambio de Mensajes.

Las relaciones de servicio afectan a las decisiones del diseño de entrega de mensajes. Estas asociaciones de identificar los requisitos permiten establecer las mejores rutas de mensaje para transacciones eficientes. Una ruta de mensajes se refiere a las rutas de red que permiten que la información se intercambie entre los consumidores y servicios.

Por lo tanto, una composición de diseño está influenciada por las relaciones de servicio y formada por las rutas de intercambio de

mensajes y comportamiento en servicio que fueron concebidas antes en la fase de diseño lógico de la relación orientada a servicios.

iii. Patrones de Formación Dirigida por Estilos.

El diseño de la composición orientada a servicios es la construcción de las formaciones por posicionamiento de servicios en ciertas formas, nombradas estilos, para proporcionar soluciones. Estos acuerdos forman patrones visuales que ayudan a permitir un eficiente enrutamiento de mensajes y la ejecución de la transacción. Las formaciones de servicio que se descubrieron también son consideradas como soluciones empaquetadas en las que se comunican las estrategias empleadas para resolver las preocupaciones de la organización.

iv. Diseño de la Composición Orientada a Servicios.

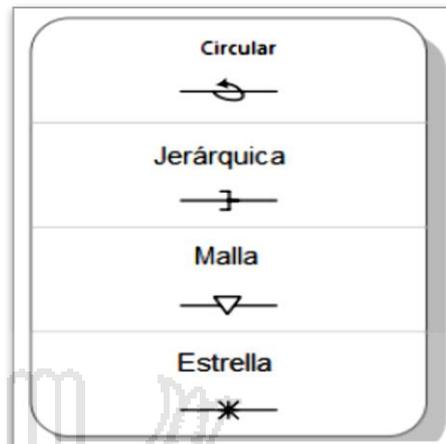
Para ofrecer soluciones efectivas a los planos de diseño orientado a servicios, se proponen una serie de composiciones. Un estilo de composición de diseño es simplemente un patrón.

Es similar a una plantilla que puede servir de guía a la vinculación de las estructuras de la atomicidad, composición y clúster de servicios para comunicar los tipos de problemas que pueden ayudar a resolver y ayudar a forjar las estrategias de diseño, tales como la reutilización de servicios y la interoperabilidad. Sus nombres son los estilos, ya que forman a la composición de la entrega del diseño. En pocas palabras, una composición de diseño orientada a servicios no sólo comunica una solución, sino que también proporciona una estrategia que se puede aprovechar en el futuro.

Las cuatro principales asociaciones conceptuales de los tipos de servicios son: circular, jerárquica, malla y estrella.

El estilo de la asociación como circular se utiliza para describir una composición de diseño circular. De la misma manera, en las asociaciones de tipo jerárquica, malla y estrella de la red se emplean el diseño de la composición para describir una red (Bell, 2008).

Figura 10. Asociación de la composición orientada a servicios.



Fuente: Bell, Michael. Service Oriented Modeling.

- **Estilo de la Composición del Diseño Circular.** El estilo de diseño de la composición circular representa una secuencia de eventos, cada uno de los cuales está representado por un único servicio en una serie. Imaginemos que un número de los servicios están vinculados por algún negocio o asociación tecnológica. El mensaje está en las manos del creador del servicio en el primer mensaje para el próximo servicio que reciben. Posteriormente a la entrega de mensajes siempre participa el próximo servicio. Cuando esta operación se ha completado, el mensaje resultante llega de retorno al punto de partida.

El estilo de composición circular reduce el tráfico de red, evitando intermediarios innecesarios. En cambio, los mensajes se entregan de un servicio a otro hasta que la respuesta se entrega al autor del mensaje. La entrega de mensajes implica una serie de servicios que comparten la carga de procesamiento

de transacciones, en lugar de aplicar la tensión a un único servicio. El estilo de la composición del diseño circular es adecuada para la gestión de negocios y tecnologías, procesos a través de la participación de las partes interesadas directamente sin la intervención de intermediarios.

- **Estilo de la Composición del Diseño Jerárquico.** La asociación jerárquica conceptual del servicio ayuda al proceso a descubrir conceptos e ideas para la asociación de sus atributos comunes. También se muestra la necesidad de identificar la reutilización de clúster de servicios, para poder abordar el aspecto de consumo de los servicios, incluso antes de que sean transportados a su producción. El análisis suele facilitar la estructura de las asociaciones jerárquicas de las capas del servicio.
- **Estilo de la Composición del Diseño Malla.** Con demasiada frecuencia, los profesionales se enfrentan al tiempo de diseño y al tiempo de ejecución de los desafíos ambientales que requieren la colaboración de una planificación del servicio y en la integración del servicio. Estas dificultades surgen debido a la interoperabilidad creciente en el entorno computacional teniendo así complejidades, los sistemas operativos y las plataformas múltiples que emplean las organizaciones.

El estilo de la composición del diseño Malla se puede utilizar para obtener la siguiente lógica de las ventajas del diseño:

- Aliviar el negocio y los desafíos tecnológicos de interoperabilidad.
- Las líneas de negocios y puente de dominio superan las barreras geográficas, ya que permiten la simplificación de las complejas estructuras de distribución de negocios, y facilitan la gestión del control sobre federados y no federados de negocios.
- Aumentar la reutilización de assets.
- **Estilo de la Composición del Diseño Estrella.** El último estilo de diseño es la composición de estrella, que ofrece otro punto de vista de la estrategia de diseño. Los estilos de diseño tratan de resolver los problemas, pero también facilitan un plan de ataque, una estrategia y un mapa de ruta para lograr los objetivos de diseño. Por lo tanto, el estilo de composición estrella añade otro punto de vista de los planos de diseño. Se puede mejorar la creación de un diseño del ambiente débilmente acoplados y ayudar a medir la efectividad al dividir los ambientes en los que se elaboran.

v. Modelado de Transacciones Orientado a Servicios.

En 1983, Theo Haerder y Andreas Reuter presentaron requisitos fundamentales para la ejecución de las transacciones, en el trabajo de investigación “Principios de la transacción orientada a la recuperación de las bases de datos” (MacVittie, 2007).

Su trabajo está centrado en cuatro grandes principios que definen una transacción y se identificaron cuatro atributos más importantes que garantizarán el buen fin de las actividades de operación. Estos grandes principios son por sus siglas en inglés

(ACID), En concreto **ACID** es un acrónimo de **A**tomicity, **C**onsistency, **I**solation and **D**urability: Atomicidad, Consistencia, Aislamiento y Durabilidad en español:

- Atomicidad.
- Coherencia.
- Aislamiento.
- Durabilidad.

Se convirtió en el estándar de la industria para la manipulación de datos fiables y la integridad de las transacciones en un entorno multiusuario. Haerder y Reuter, describen una transacción como: la manera en que se componen varias operaciones de software que interactúan con una base de datos durante un período determinado de tiempos. No hay cambios en los datos, se debe deducir que todas estas actividades han concluido con éxito.

Las propiedades para asegurar la integridad de los datos ACID son las siguientes:

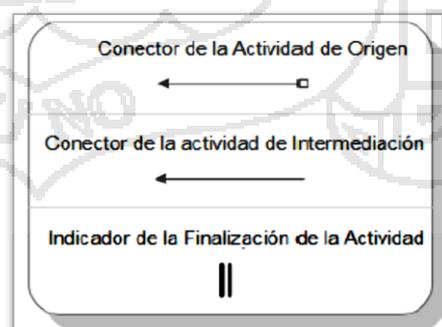
- **Atomicidad:** Una transacción confirma todos los cambios aplicados a una base de datos de su base de actividades, si sus operaciones se ejecutan con éxito. De lo contrario, una cancelación de la operación es responsable de la restauración de todas las modificaciones aplicadas en los datos. Esto se conoce como la condición todo o nada.
- **Coherencia:** Los resultados deben estar comprometidos a ser válidos y no debe perjudicar la integridad de los datos.
- **Aislamiento:** Una transacción debe ser aislada de otras transacciones que se ejecutan simultáneamente. No deben interferir entre sí durante la ejecución.
- **Durabilidad:** Los cambios realizados por las transacciones de éxito deben ser duraderas y persistentes, a pesar de cualquier error que se produzca después.

ACID es un método de procesamiento de transacciones estrechamente unidas. Es decir, este enfoque fue diseñado para hacer frente a los problemas locales de depósito y fiabilidad de las soluciones de cada acción, es decir, los resultados exitosos de transacciones están garantizados sólo por un corto período de tiempo.

En el ambiente de computación orientado a servicios deben ser interoperables requiere un modelo que pueda manejar la interacción y colaboración de servicios en las estructuras complejas y agregadas. Además, se requiere la integridad de las transacciones entre las formaciones de servicios débilmente acoplados dispersados a lo largo de múltiples líneas de negocios y organizaciones. Un esquema de operación orientada a servicio debe ser encargado de resolver desafíos de las transacciones de larga duración (MacVittie, 2007).

- **Conectores de la Actividad.** Existen tres conectores de la actividad de servicios que pueden ayudar en la descripción de la interacción y colaboración de los servicios. Se deben utilizar en cada actividad una sección para describir los pasos de enrutamiento de mensajes entre los servicios participantes y los consumidores (Bell, 2008).

Figura 11. Conectores de la actividad de transacción.



Fuente: Bell, Michael. Service Oriented Modeling.

- **Conector de la Actividad de Origen.** En una sección de actividades, pueden identificarse una serie de asociaciones de los servicios, para realizar una tarea determinada. Por lo tanto, se utiliza este conector para referirse a un inicio de actividad y para ser capaz de identificar un servicio que origina un mensaje (conocido como mensaje origen).
- **Conector de la Actividad de Intermediación.** Una actividad puede implicar múltiples servicios y consumidores. Por lo tanto, para las operaciones de transacción se utiliza el conector de intermediación. Esto también puede ser útil cuando el entorno de producción cuenta con los servicios de proxy o intermediarios orientados a servicios.
- **Indicador de la Finalización de la Actividad.** Un diagrama de las transacciones de servicios, puede contener una gran serie de finalizaciones en las actividades de servicios, se emplea el conector para identificar el estado final de una función determinada.

2.2.11. Rational Unified Process (RUP).

Según IBM Rational Unified Process for System RUP (2002) es la metodología estándar de la industria para la construcción completa del ciclo de ingeniería de software, tanto para sistemas tradicionales como para sistemas web, llamada así por sus siglas en inglés Rational Unified Process. Es un proceso de ingeniería de software, bien definido y estructurado; a la vez que es un producto que provee un marco de proceso adaptable a las necesidades y características de cada proyecto específico.

Esta metodología permite mayor productividad en equipo y la realización de mejores prácticas de software a través de plantillas y herramientas que lo guían en todas las actividades de desarrollo crítico del software.

El proceso de desarrollo de software requiere, por un lado, un conjunto de conceptos, una metodología y un lenguaje propio. A este proceso también se le llama el ciclo de vida del software, su definición viene dada por tres características fundamentales:

- Está dirigido por casos de uso.
- Es un proceso centrado en la arquitectura.
- Es iterativo e incremental.

RUP dirigido por “casos de uso” significa que el proceso de desarrollo sigue una trayectoria que avanza a través de los flujos de trabajo generados por los casos de uso. Los casos de uso se especifican y diseñan en el principio de cada iteración, y son la fuente a partir de la cual los encargados de las pruebas

construyen sus casos de prueba. Los casos de uso describen la funcionalidad total del sistema, pensada en términos de la importancia de la misma para el usuario (no sólo de la funcionalidad en sí). Pero esto no significa que se desarrollen aisladamente respecto de la arquitectura, sino que se desarrollan a la vez, madurando ambos según avanza el ciclo de desarrollo. Los casos de uso guían la arquitectura del sistema (como parte del proceso) y ésta influye en la selección de los casos de uso. La arquitectura involucra los elementos más significativos y está influenciada entre otros por las plataformas de software, los sistemas operativos, los sistemas de gestión de base de datos, además de otros como sistemas heredados y requerimientos no funcionales. Por esta razón eso dice que el RUP está centrado en la arquitectura, lo que invoca más la relación con los principios de la usabilidad. Según lo que establece el RUP (De Carlo et al., 2004) los elementos del RUP son:

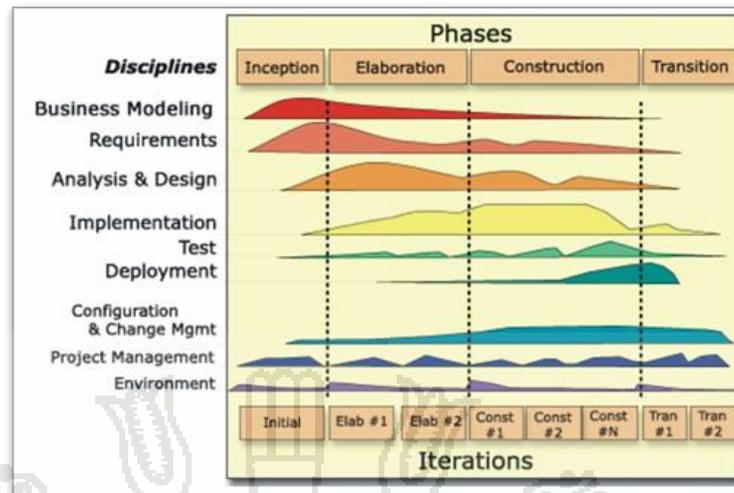
- Actividades, son los procesos que se llegan a determinar en cada iteración. En concreto es una unidad de trabajo que una persona que desempeñe un rol puede ser solicitado a que realice. Las actividades tienen un objetivo concreto, normalmente expresado en términos de crear o actualizar algún producto.
- Roles, definen el comportamiento y responsabilidades de un individuo, o de un grupo de individuos trabajando juntos como un equipo. Una persona puede desempeñar diversos roles, así como un mismo rol puede ser representado por varias personas. Las responsabilidades de un rol

son tanto el llevar a cabo un conjunto de actividades como el ser el dueño de un conjunto de artefactos.

- Artefactos, son un producto, es un trozo de información que es producido, modificado o usado durante el proceso de desarrollo de software. Los productos son los resultados tangibles del proyecto, las cosas que va creando y usando hasta obtener el producto final. Un artefacto puede ser cualquiera de los siguientes (RUP, 2002): un documento, un modelo, y un elemento del modelo.

El RUP (2002) divide los proyectos en pequeños ciclos o iteraciones a través de cada una de las fases por las que pasa el proyecto, las cuales son establecidas claramente, cada una desarrollada en una o más iteraciones que ejecutan actividades definidas para cada flujo de trabajo de los conocidos en cualquier proceso de desarrollo. Concretamente el RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y las cuales se definen de acuerdo al nivel de madurez que alcanzan los productos que se van obteniendo en cada actividad ejecutada. La terminación de cada fase ocurre en el hito correspondiente a cada una, donde se evalúa que se hayan cumplido los objetivos de la fase en cuestión. Y desde la terminación de la fase de inicio se puede ya determinar la factibilidad tanto operativa como económica del proyecto, la cual nos lleva a tomar la decisión de continuarlo o no realizarlo.

Figura 12. Arquitectura general RUP.



Fuente: María de Lourdes Santiago Zaragoza. Desarrollando aplicaciones informáticas con el Proceso de Desarrollo Unificado.

El proceso del desarrollo del software según RUP puede ser descrito en dos dimensiones:

- El eje horizontal representa el tiempo y muestra los aspectos dinámicos del proceso siendo expresados en términos de ciclos, fases, iteraciones e hitos.
- El eje vertical representa los aspectos estáticos del proceso, cómo se describe en términos de actividades, artefactos, empleados y flujos de trabajo.

Es recomendable que a cada una de estas iteraciones se les clasifique y ordene según su prioridad, y que cada una se convierte luego en un entregable al cliente. Esto trae como beneficio la retroalimentación que se tendría en cada entregable o en cada iteración.

RUP tiene dos dimensiones:

- La primera dimensión representa el aspecto dinámico del proceso cuando entra en vigor, y se expresa en términos de fases, interacciones e hitos.
- La segunda dimensión representa el aspecto estático del proceso: la forma en que se describen los términos de los componentes de proceso, disciplinas, actividades, flujos de trabajos, artefactos y roles.

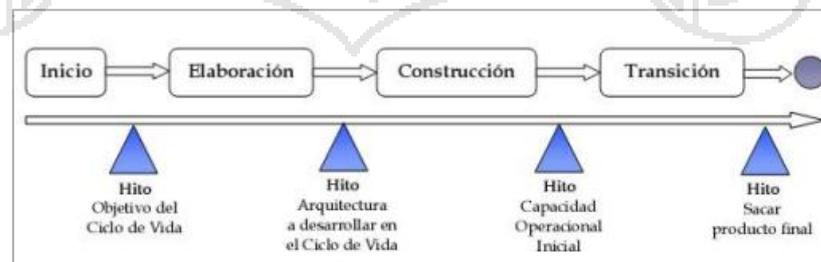
Disciplinas de RUP:

- Modelos de negocio.
- Requisitos.
- Análisis y Diseño.
- Implementación.
- Pruebas.

2.2.11.1. Ciclo de Vida del Software.

El ciclo de vida del software describe el desarrollo de software, desde la fase inicial hasta la fase final.

Figura 13. Fases del RUP y sus hitos.



Fuente: María de Lourdes Santiago Zaragoza. Desarrollando aplicaciones informáticas con el Proceso de Desarrollo Unificado.

El RUP divide en cuatro fases el desarrollo del software:

- Concepción (o inicio), el objetivo es determinar la visión del proyecto.
- Elaboración, el objetivo es determinar la arquitectura óptima.
- Construcción, el objetivo es llevar a obtener la capacidad operacional inicial.
- Transición, el objetivo es llegar a obtener las mejoras del proyecto.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones o flujos de trabajo, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes.

El ciclo de vida que se desarrolla por cada iteración, es llevado bajo dos disciplinas: desarrollo y soporte.

Disciplina de Desarrollo:

- Modelado del Negocio: Entendiendo las necesidades del negocio.
- Análisis de Requerimientos: Traslado de las necesidades del negocio a un sistema automatizado.
- Análisis y Diseño: Traslado de los requerimientos dentro de la arquitectura de software.
- Implementación: Creando software que se ajuste a la arquitectura y que tenga el comportamiento deseado.

- Pruebas: Asegurándose que el comportamiento requerido es el correcto y que todo el solicitado está presente.
- Distribución/Instalación (Despliegue: Hacer todo lo necesario para la salida del proyecto.

2.2.11.2. Disciplinas de Desarrollo.

Las disciplinas de desarrollo RUP representan más de cerca los roles individuales de sus miembros o subgrupos dentro del equipo de desarrollo integral del proyecto.

A. Modelar el Proceso de Negocio.

En el modelado de negocios se documentan los procesos de negocio utilizando los llamados casos de negocio. Esto asegura el entendimiento común entre todos los "involucrados" sobre qué procesos de negocio necesitan ser soportados en la organización.

Se define un modelo de negocio como las soluciones generalizadas que pueden ser implementadas y aplicadas en una situación problemática, y con ello eliminar uno o más de los problemas inherentes. Se compone de los artefactos:

- Casos de uso de negocios.
- Reglas de negocio.
- Diagramas de secuencia.

- Diagramas de actividades.
- Diagramas de objetos de negocio.

- Casos de Uso de Negocios. Es un modelo de las funciones de negocio previsto. Se utiliza como un insumo esencial para identificar los roles y los resultados de la organización.

- Diagramas de Secuencia. El diagrama de secuencia es un tipo de diagrama usado para modelar interacción entre objetos en un sistema, se modela para cada caso de uso. El diagrama de secuencia contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario y mensajes intercambiados entre los objetos.

- Diagramas de actividades. Se utilizan para ilustrar las actividades. En el punto de vista externo, se utilizan diagramas de actividades para la descripción de los procesos de negocio que describen la funcionalidad del sistema empresarial.

Los Diagramas de actividad le permiten pensar funcionalmente. Con el enfoque orientado a objetos. Debido a que es posible describir explícitamente los eventos paralelos, el diagrama de actividades es muy adecuado para la ilustración de los procesos de negocio, ya que los procesos de negocios rara vez se presentan en una manera lineal y con frecuencia presentan paralelismos.

- Diagramas de objeto de negocio. Es un modelo de objeto que describe la realización de casos de uso de negocio. Identificando las entidades de negocios.
- Reglas de Negocio. Son las declaraciones de la política o las condiciones que deben cumplirse.

B. Analizar Requerimientos.

El objetivo de esta actividad es describir qué debería hacer el sistema y permitir que los desarrolladores y el cliente se pongan de acuerdo en esa descripción. Son definidos como una condición o capacidad que un sistema debe cumplir. Los Requisitos se dividen en 2: requerimientos funcionales y requerimientos no funcionales.

- Requerimientos Funcionales: Especifican las acciones que un sistema debe ser capaz de realizar, sin tener en cuenta las limitaciones físicas del sistema. Estos a menudo se describen mejor en la especificación de los casos de uso. Los requerimientos funcionales especifican la entrada y el comportamiento de la producción de un sistema.
- Requerimientos no Funcionales: Normalmente describen el criterio de desempeño, fiabilidad, seguridad y otros parámetros operacionales.

- Prototipo de Interfaz de Usuario. El prototipo puede manifestarse como dibujos en papel o imágenes.

C. Analizar Y Diseñar.

El objetivo del Análisis y Diseño es definir la arquitectura del sistema proveyendo bases sólidas para el proceso de diseño e implementación.

- Análisis. El modelo de análisis contiene las clases de análisis. Las clases de análisis, en conjunto, representan un primer modelo conceptual del sistema. Las clases de análisis rara vez sobreviven sin cambios en el diseño. Muchas de ellas representan colaboraciones del conjunto de objetos, a menudo encapsulados por subsistemas. (Leffingwell, 2010).
- Diseño. Es un modelo de objeto que describe la realización de casos de uso, y sirve como una abstracción del modelo de implementación y su código fuente. El modelo de diseño se utiliza como insumo esencial para las actividades de implementación y pruebas. Se trata de un artefacto completo, compuesto, que abarca todas las clases de diseño, subsistemas, paquetes, las colaboraciones y las relaciones entre ellos. (Leffingwell, 2010).

D. Probar e Implementar.

El propósito de la implementación es el desarrollo del sistema, en el cual se deben obtener finalmente las herramientas necesarias para resolver los requerimientos definidos en las etapas previas.

El sistema se desarrolla a través de la implementación de componentes. RUP describe cómo reutilizar los componentes existentes o implementar nuevos componentes con responsabilidades bien definidas, alcanzando un sistema fácil de mantener e incrementando las posibilidades de reutilización.

E. Soporte y Monitoreo.

Las organizaciones exitosas no sólo automatizan sus procesos de negocio, sino que también controlan su ejecución y la ajustan dinámicamente en respuesta a los resultados en tiempo real.

2.2.12. Modelado de Datos con UML.

Se centra en la creación de tablas de las clases existentes, esto asegura todas las clases que se han creado. Una vez que se ha producido el mapeo de clases a tablas, se puede empezar a buscar formas de optimizar la base de datos, a partir de cómo manejar las tablas que se crearon sobre la base de relaciones de herencia en el modelo de clases y las clases que tomó parte en las relaciones

muchos-a-muchos que se tienen que dividir en las asociaciones de las tablas. A partir de ahí el equipo de diseño de bases de datos comenzará a garantizar la unicidad de las tablas y la aplicación de elementos tales como reglas de uso de restricciones sobre la base de datos.

Hay varias formas de mapear modelos. En este escenario, se valorará la aplicación y los modelos de bases de datos de diseño para el modelo de análisis lógico, y se mapeará el modelo de diseño de la aplicación directamente en el modelo de datos. Esto dará la capacidad de entender la información importante. El asignar el modelo de objetos al modelo de datos ayuda a construir el acceso a datos en iteraciones posteriores. Se mapean las clases a tablas, los atributos a columnas, los tipos a tipos de datos y las asociaciones a relaciones, lo que ayudará a los equipos a entender cómo la aplicación va a interactuar con la base de datos.

No todos los elementos de cada modelo serán asignados. Sólo las clases que son persistentes se asignarán a la base de datos, y no se pueden derivar los atributos dentro de las clases persistentes que no se asignan a las columnas (Naiburg & Maksimchuk, 2001).

A. Mapeo de las Clases a Tablas.

Hay cuatro formas básicas de mapeo de clases a las tablas: uno a uno, uno-a-muchos, muchos-a-uno y muchos a muchos. Es posible que los mapas de manera diferente por varias razones, incluyendo el rendimiento,

seguridad, facilidad de consulta, las preferencias del administrador de la base de datos, estándares corporativos, las necesidades específicas de la base de datos, u otros motivos que pueden haber experimentado. También hay algunas asignaciones que se producen sobre la metodología de base de datos relacional en general: muchos-a-muchos, subtipos, supertipos, y clases de asociación.

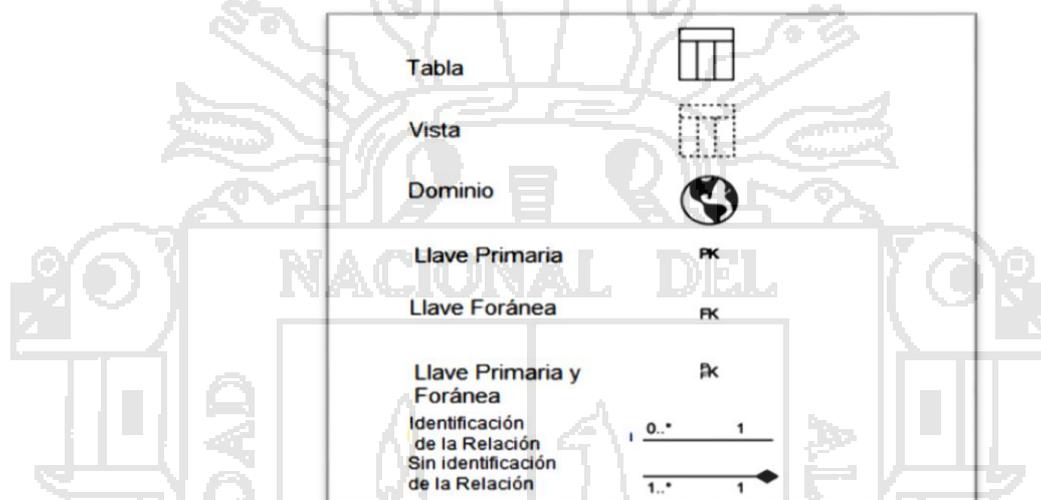
Una ventaja de usar UML sobre las notaciones tradicionales de la entidad-relación (ER) para el modelo lógico es el soporte para una clase de asociación al mismo tiempo que muestra la relación de muchos a muchos (Naiburg & Maksimchuk, 2001).

B. Elementos del Diagrama para Diseño de la Base de Datos.

- Tabla: Agrupación de la información en una base de datos sobre el mismo tema, compuesto por columnas.
- Vista: Un componente de una tabla que tiene un solo atributo de la tabla.
- Dominio: El conjunto válido de valores para un atributo o columna.
- PK: La clave candidata que se elija para identificar las filas de una tabla también conocida como llave primaria.
- FK: una columna o un conjunto de columnas de una tabla que se asignan a la clave primaria de otra tabla también conocida como llave foránea.

- Identificación de la Relación: Una relación entre dos tablas en las que la tabla secundaria debe coexistir con la tabla principal.
- Sin Identificación de la Relación: Una relación entre dos tablas en las que cada tabla puede existir independientemente de otras.

Figura 14. Elementos del diagrama para el diseño de base de datos.



Fuente: Eric J. Naiburg, Robert A. Maksimchuk. UML for Database Design.

2.2.13. C#.

C# es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la ECMA (ECMA-334) e ISO (ISO/IEC 23270). Es uno de los lenguajes de programación diseñados para la infraestructura de lenguaje común.

Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma .NET, similar al de Java, aunque incluye mejoras derivadas de otros lenguajes.

El nombre C Sharp fue inspirado por la notación musical, donde '#' (sostenido, en inglés sharp) indica que la nota (C es la nota do en inglés) es un semitono más alta, sugiriendo que C# es superior a C/C++. Además, el signo '#' se compone de cuatro signos '+' pegados.

Aunque C# forma parte de la plataforma .NET, ésta es una API, mientras que C# es un lenguaje de programación independiente diseñado para generar programas sobre dicha plataforma. Ya existe un compilador implementado que provee el marco Mono - DotGNU, el cual genera programas para distintas plataformas como Windows, Unix, Android, iOS, Windows Phone, Mac OS y GNU/Linux.

2.2.14. Microsoft SQL Server.

Microsoft SQL Server es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional. Sus lenguajes para consultas son T-SQL y ANSI SQL. Microsoft SQL Server constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son Oracle, PostgreSQL o MySQL.

Características de Microsoft SQL Server.

- Soporte de transacciones.

- Soporta procedimientos almacenados.
- Incluye también un entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y los terminales o clientes de la red sólo acceden a la información.
- Además permite administrar información de otros servidores de datos.

Este sistema incluye una versión reducida, llamada MSDE con el mismo motor de base de datos pero orientado a proyectos más pequeños, que en sus versiones 2005 y 2008 pasa a ser el SQL Express Edition, que se distribuye en forma gratuita.

Es común desarrollar completos proyectos complementando Microsoft SQL Server y Microsoft Access a través de los llamados ADP (Access Data Project). De esta forma se completa la base de datos (Microsoft SQL Server), con el entorno de desarrollo (VBA Access), a través de la implementación de aplicaciones de dos capas mediante el uso de formularios Windows. En el manejo de SQL mediante líneas de comando se utiliza el SQLCMD, osql, o PowerShell.

Para el desarrollo de aplicaciones más complejas (tres o más capas), Microsoft SQL Server incluye interfaces de acceso para varias plataformas de

desarrollo, entre ellas .NET, pero el servidor sólo está disponible para Sistemas Operativos.

A. Programación.

- **T-SQL.**

T-SQL (Transact-SQL) es el principal medio de interacción con el Servidor. Permite realizar las operaciones claves en SQL Server, incluyendo la creación y modificación de esquemas de la base de datos, la introducción y edición de los datos en la base de datos, así como la administración del servidor como tal. Esto se realiza mediante el envío de sentencias de T-SQL y declaraciones que son procesadas por el servidor y los resultados (o errores) regresan a la aplicación cliente.

- **Cliente Nativo de SQL.**

Cliente Nativo de SQL es la biblioteca de acceso a datos para los clientes de Microsoft SQL Server versión 2005 en adelante. Implementa nativamente soporte para las características de SQL Server, incluyendo la ejecución de la secuencia de datos tabular, soporte para bases de datos en espejo de SQL Server, soporte completo para todos los tipos de datos compatibles con SQL Server, conjuntos de operaciones asíncronas, las notificaciones de consulta, soporte para cifrado, así como recibir varios conjuntos de resultados en una sola sesión de base de datos.

Cliente Nativo de SQL se utiliza como extensión de SQL Server plugins para otras tecnologías de acceso de datos, incluyendo ADO u OLE DB. Cliente Nativo de SQL puede también usarse directamente, pasando por alto las capas de acceso de datos.

B. Versiones.

Tabla 2. Versiones Microsoft SQL Server.

Historia de versiones			
Versión	Año	Nombre de la versión	Nombre clave
1.0 (OS/2)	1989	SQL Server 1-0	SQL
4.21 (WinNT)	1993	SQL Server 4.21	SEQUEL
6.0	1995	SQL Server 6.0	SQL95
6.5	1996	SQL Server 6.5	Hydra
7.0	1998	SQL Server 7.0	Sphinx
-	1999	SQL Server 7.0 OLAP Tools	Plato
8.0	2000	SQL Server 2000	
8.0	2003	SQL Server 2000 64-bit Edition	Liberty
9.0	2005	SQL Server 2005	Yukon

10.0	2008	SQL Server 2008	Katmai
10.50	2010	SQL Server 2008 R2	Kilimanjaro
11.0	2012	SQL Server 2012	Denali

Fuente: SQL Server.org.

2.2.15. Metodología Documentación de Requerimientos Centrada en el Usuario DoRCU.

La metodología Documentación de Requerimientos Centrada en el Usuario DoRCU, consta de las siguientes etapas:

- **Elicitación de requerimientos.**

Esta es la etapa en donde se adquiere el conocimiento del trabajo del cliente/usuario, se busca comprender sus necesidades y se detallan las restricciones medioambientales.

- **Análisis de Requerimientos.**

En esta etapa se estudian los requerimientos extraídos en la etapa previa a los efectos de poder detectar, entre otros, la presencia de áreas no especificadas, requisitos contradictorios y peticiones que aparecen como vagas e irrelevantes.

- **Especificación de Requerimientos.**

Esta etapa es un proceso de descripción del requerimiento. Si se presentan dificultades para especificar un requerimiento se debe volver a la etapa anterior que se crea conveniente.

- **Validación y Certificación de los Requerimientos.**

Esta etapa final se nutre de las anteriores y realiza la integración y validación final de lo obtenido en cada una de las etapas anteriores dando, como resultado final, el Documento de Requerimientos. Este documento no es uno solo sino que, como mínimo, existen dos que son isomórficos entre sí: uno destinado al cliente/usuario a los efectos de la certificación de los Requisitos y el otro técnico, orientado a nutrir las restantes etapas de la Ingeniería de Software (Báez & Barba Brunner).



2.3.Marco Conceptual.

2.3.1. Assets.

Activo en español, es un Sistema construido con bienes y servicios, con capacidades funcionales y operativas que se mantienen durante el desarrollo de cada actividad socioeconómica específica. (Ortega Paredes, 2012).

2.3.2. Base de Datos.

Las bases de datos son el método preferido para el almacenamiento estructurado de datos. Desde las grandes aplicaciones multiusuario, hasta los teléfonos móviles y las agendas electrónicas utilizan tecnología de bases de datos para asegurar la integridad de los datos y facilitar la labor tanto de usuarios como de los programadores que las desarrollaron. Desde la realización del primer modelo de datos, pasando por la administración del sistema gestor, hasta llegar al desarrollo de la aplicación, los conceptos y la tecnología asociados son muchos y muy heterogéneos. (David Mejías Jiménez et al., 2005).

2.3.3. Diagrama de Flujo.

El Flujograma es un diagrama que expresa gráficamente las distintas operaciones que componen un procedimiento o parte de este, estableciendo su secuencia cronológica. Según su formato o propósito, puede contener información adicional sobre el método de ejecución de las operaciones, el

itinerario de las personas, las formas, la distancia recorrida el tiempo empleado.

(Guillermo Gómez Cejas, 1997).

2.3.4. Gestión.

Conjunto de trámites que se llevan a cabo para resolver un asunto. Dirección, administración de una empresa, negocio, etc. (Rebolledo Saavedra, 2013).

2.3.5. Multiusuario.

Los sistemas operativos multiusuarios son capaces de dar servicio a más de un usuario a la vez, ya sea por medio de varias terminales conectadas a la computadora o por medio de sesiones remotas en una red de comunicaciones, ni importa el número de procesadores en la maquina ni el número de procesos que cada usuario puede ejecutar simultáneamente.

Los recursos que se comparten son por lo regular una combinación de:

- Procesador.
- Memoria.
- Almacenamiento secundario (almacenaje en disco duro).
- Programas.

Periféricos como impresoras, plotters, scanner, etc. (Quintana Méndez, 2011).

2.3.6. Modelo.

Un modelo es la representación real de los datos obtenidos a partir de la información disponible. Ejemplos son el modelo de estructura y el modelo de estilo que representan la estructura analítica y la información de estilo asociada a un documento. El modelo podría ser un árbol, o un grafo orientado, o cualquier otra cosa. (Campos, et al., 2000).

2.3.7. Proceso de negocio.

El modelo del negocio describe el negocio en términos de casos de usos del negocio, que corresponde a lo que generalmente se le llama procesos.

El modelo de casos de uso del negocio es un modelo que describe los procesos de un negocio (casos de uso del negocio) y su interacción con elementos externos (actores), tales como socios y clientes, es decir, describe las funciones que el negocio pretende realizar y su objetivo básico es describir cómo el negocio es utilizado por sus clientes y socios. Implica la determinación de los actores y casos de uso del negocio. Con esta actividad se pretende: Identificar los procesos en el negocio, definir las fronteras del negocio que van a modelarse, Definir quién y qué interactuarán con el negocio y crear diagramas del modelo de casos de uso del negocio.

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados.

Un proceso de negocio es un grupo de tareas relacionadas lógicamente que se llevan a cabo en una determinada secuencia y manera y que emplean los recursos de la organización para dar resultados en apoyo a sus objetivos. (Hernández González, 2004).

2.3.8. Recursos de Información.

Los recursos de información (libros, personas, internet), así como las herramientas tecnológicas (computadoras, cámaras, impresoras), suministran lo necesario para que los alumnos logren desarrollar los productos del proyecto. (Educ.ar S.E., s.f.).

2.3.9. Software.

Software es un término informático que hace referencia a un programa o conjunto de programas de cómputo que incluye datos, procedimientos y pautas y que permite realizar distintas tareas en un sistema informático. Comúnmente se utiliza este término para referirse de una forma muy genérica a los programas de un dispositivo informático.

Es un término procedente del inglés, aceptado por la RAE y formado por oposición al término 'hardware'. (Masadelante.com, 2013).

2.3.10. Tecnología de Información (TI).

Se conoce como tecnología de información (TI) a la utilización de tecnología específicamente computadoras y ordenadores electrónicos para el manejo y procesamiento de información específicamente la captura, transformación, almacenamiento, protección, y recuperación de datos e información. (Meltom Technologies, 2013).

2.3.11. Usabilidad.

En interacción persona-ordenador, la usabilidad se refiere a la claridad y la elegancia con que se diseña la interacción con un programa de ordenador o un sitio web. (Jeffrey Rubín, 1994).

2.3.12. Funcionalidad.

Funcionalidad es lo que un producto puede hacer. Probar la funcionalidad significa asegurar que el producto funciona tal como estaba especificado. (S. Dumas & C. Redish, 1994).

2.3.13. Eficiencia.

La eficiencia se refiere a hacer las cosas bien, es obtener el mejor o máximo rendimiento utilizando un mínimo de recursos. La eficacia, por otra parte, es hacer las cosas de la manera correcta y de esta manera alcanzar el resultado deseado. (AulaFacil S.L., 2009).

2.3.14. Satisfacción del Cliente.

La satisfacción del cliente es un término propio del marketing que hace referencia a la satisfacción que tiene un cliente con respecto a un producto que ha comprado o un servicio que ha recibido, cuándo éste ha cumplido o sobrepasado sus expectativas. (Crece Negocios, 2011).



CAPÍTULO III

MATERIALES Y MÉTODOS.

3.1. Metodología de Investigación.

En la presente investigación se utilizó el tipo de investigación cuasi experimental con enfoque correlacional, con la investigación cuasi experimental es posible inferir relaciones causales entre la variable independiente y la variable dependiente; y el enfoque correlacional pretende analizar la relación entre las variables: Modelo de sistema desarrollado con arquitectura orientada a servicios (SOA) para la empresa BICI&MOTOS EIRL – Juliaca 2012 y la de gestión de ventas.

3.2. Diseño de la Investigación.

El diseño de este trabajo de investigación es cuasi experimental, se precisa que los diseños cuasi experimentales manipulan al menos una variable independiente para ver su efecto y relación con una o más variables dependientes. Por el tipo de investigación

se trabajó con un grupo experimental al cual se le sometió a una prueba de pre test y post test.

Este grupo estuvo conformado por 16 personas pertenecientes a la empresa BICI&MOTOS EIRL. Este diseño con grupo experimental permitió la comparación de resultados del pre test y post test, demostrando que el sistema desarrollado con arquitectura orientada a servicios SOA contribuyó a la adecuada administración de la información en la empresa.

3.3.Población y Muestra.

3.3.1. Población.

La población de la presente investigación estuvo conformada por un total de 20 personas trabajadores de la empresa importadora BICI&MOTOS EIRL - Juliaca.

3.3.2. Muestra.

Para el presente trabajo de investigación se empleó un muestreo no probabilístico de tipo intencional, para hallar el tamaño de la muestra se aplicó la fórmula de muestreo aleatorio simple:

$$n = \frac{n_o}{1 + \frac{n_o}{N}}$$

$$n_o = Z_{\alpha/2}^2 \frac{\sigma^2}{E^2} = Z_{\alpha/2}^2 \frac{(P)(Q)}{E^2}$$

Dónde:

n : Tamaño de muestra.

n_o : Tamaño de muestra aproximado.

N: Tamaño de la población en estudio.

Z_{α} : Valor correspondiente al nivel de significancia.

E : Error de tolerancia de la estimación.

α : Nivel de significancia.

σ^2 : Varianza de la variable.

Para la población de 20 personas se halló la muestra:

Para un nivel de confianza = 95%

$\alpha = 5\%$

E = 10 %

P = Q = $\frac{1}{2}$

$$n_o = Z_{\alpha/2}^2 \frac{(P)(Q)}{E^2} = 1.96^2 \frac{\left(\frac{1}{2}\right)\left(\frac{1}{2}\right)}{0.10^2}$$

$$n_o = 96.04$$

$$n = \frac{96.04}{1 + \frac{96.04}{20}}$$

$$n = 16$$

Por fórmula se obtuvo una muestra de 16 personas, realizando la selección de la siguiente forma:

- El área contable conformada por 2 personas.
- El área de administración conformada por 2 persona.
- El área de almacén conformado por 3 personas.
- En el área de ventas conformada 9 personas.

3.4.Métodos de Recopilación de Datos.

La principal fuente de datos que se utilizó para la demostración de la hipótesis fue la técnica de la “encuesta” pre test y post test, utilizando como instrumento

“cuestionarios”. Encuesta que estuvo dirigida a los trabajadores de la empresa con la finalidad de recoger información referente a la utilización antes y después de la implantación del sistema desarrollado con arquitectura orientada a servicios.

3.5.Métodos de Tratamiento de Datos.

Para el procesamiento de los datos obtenidos mediante los cuestionario se utilizó el software IBM SPSS Statistics 20 y para el tratamiento de los mismos se usó las técnicas de: Matriz de datos, Cálculo de la media o promedio, Varianza y desviación estándar. Para finalmente hacer el respectivo análisis e interpretación.

3.6.Material Experimental.

Modelo de sistemas desarrollado con arquitectura orientada a servicios.

3.6.1. Metodología de Desarrollo del Sistema.

Para el desarrollo del sistema de gestión ventas se utilizó la Metodología Rational Unified Process (RUP) ya que esta proporciona un enfoque disciplinado para la asignación de tareas y responsabilidades, ya que su objetivo es garantizar la producción de software de alta calidad que satisfaga las necesidades de los usuarios finales dentro de un horario predecible y presupuestado. Asimismo, cabe señalar que se tomó en consideración ciertos puntos de la metodología para el desarrollo orientado a servicios.

La metodología para el desarrollo orientado a servicios no es una nueva metodología, sino que se construye sobre el proceso o enfoque de desarrollo que se sigue en la empresa, agregando actividades y componentes específicos para el desarrollo de los servicios, y por lo tanto cualquier proceso podría servir de base. Sin embargo, dada la naturaleza compleja y cambiante de los requerimientos del negocio en el contexto de las organizaciones actuales, se siguió un enfoque iterativo incremental como forma de lidiar con estos cambios, con perspectiva en la generación de artefactos intermedios como forma de ir obteniendo productos claves en el avance del mismo que provean visibilidad sobre el desarrollo.

A continuación se presenta el cuadro de las etapas de desarrollo, productos intermedios obtenidos y la descripción del proceso realizado de forma iterativa hasta el logro del producto siguiente.

Tabla 3. Etapas de desarrollo del sistema según la disciplina RUP.

Disciplina RUP	Etapa	Artefacto	Descripción de la iteración
Modelado del Negocio	Conocimiento de la empresa.	Entrevista con la empresa.	Reuniones coordinadas con la empresa con la finalidad de comprender sus procesos y funcionamiento.
Análisis de Requerimientos	Obtención de Requerimiento.	Documento de requerimientos según la metodología DORCU.	Se realizó en coordinación con la empresa reuniones con procesos de retroalimentación, siguiendo la metodología elegida, para concluir con el documento y obtener los requerimientos de usuario y continuar con la siguiente etapa.
	Análisis del Sistema.	Diagramas de análisis del sistema (Casos de Usos, secuencia, actividades y clases) Diseño de Base de Datos.	Se realizó reuniones con la empresa para conocer el proceso comercial. Se elaboró los diagramas de los procesos comerciales en estudio en coordinación con los involucrados.
Análisis y diseño del Sistema	Diseño del Sistema.	Diseño de interfaces del sistema.	En base a los resultados obtenidos en la etapa de análisis se presentó a la empresa las posibles interfaces a implementar para su aprobación o corrección.
	Desarrollo del Sistema.	Desarrollo del sistema por módulos.	Se desarrolló el sistema por módulos separados, cada módulo desarrollado fue presentado a la empresa para su conformidad.
Pruebas	Prueba del Sistema.	Pruebas de Aceptación y Usabilidad.	Una vez implementado el sistema, y siguiendo la metodología de investigación elegida se desarrolló pruebas (encuesta post test, usabilidad) para su evaluación.

Fuente: Elaboración propia.

3.6.2. Hardware y Software de Desarrollo.

Tabla 4. Hardware y Software.

HARDWARE	CARACTERÍSTICAS
- Notebook	Core I3
- Impresora	Laser
- Computadora Personal	Core I3
SOFTWARE	FUNCIÓN
- Star UML	Modelador UML.
- C#	Herramienta de desarrollo.
- SQL	Gestor de base de datos.
- Erwin	Modelador de bases de datos.
- IBM SPSS statistic 20	Herramienta para las estadísticas.
- Word	Herramienta para la documentación de la investigación.
- PowerPoint	Herramienta para la presentación de la investigación.

Fuente: Elaboración propia.

CAPÍTULO IV

EXPOSICIÓN Y ANÁLISIS DE LOS RESULTADOS.

4.1. Aplicación de la Metodología.

4.1.1. Recopilación de Requerimientos.

4.1.1.1. Requerimientos Funcionales Aplicando DORCU.

La obtención de los requisitos es una de las etapas importantes para desarrollar una aplicación es necesario analizar el problema, para poder ofrecer una solución y así resolver el problema.

Los requerimientos fueron obtenidos mediante la metodología de documentación de requerimientos centrados en el usuario DoRCU la que se basa en la Elitización de requerimientos, Análisis de requerimientos, la Especificación de requerimientos y finalmente la Validación y Certificación de requerimientos. En este proceso, se trabajó en forma

coordinada e iterativa con personal de la empresa, analizando sus procesos y necesidades.

A. Elitización de Requerimientos:

En esta etapa se conformó equipos de trabajo con el personal de la empresa involucrado con el sistema propuesto. El cual estuvo conformado por:

- Dueño de la empresa.
- Administrador de la empresa.
- Encargado de Ventas.
- Encargado de Compras y Almacén.
- Encargado de Desarrollar el Sistema.

Con este grupo se trabajó la recolección de información necesaria para la elaboración del sistema, la cual fue la siguiente:

- Información sobre las ventas realizadas.
- Información sobre las compras realizadas.
- Información sobre las sucursales con las que cuenta la empresa.
- Información sobre los proveedores de la empresa.
- Información sobre los productos con los que cuenta la empresa.
- Información sobre los clientes.
- Información sobre el almacén y los movimientos comerciales.
- Servicio de información a clientes de productos de la empresa.

B. Análisis de Requerimientos.

Se realizó el análisis de los requerimientos planteados en la etapa previa, los que corresponden a los datos tomados en cuenta referentes al proceso general que se lleva a cabo en la empresa desde la compra de productos, el almacenamiento de los mismos, la distribución de los productos a las diferentes sucursales, y la venta al público.

Sobre la base de la información obtenida se procedió a trabajar categorizándola de acuerdo a su nivel de importancia y recurrencia, planteando un marco metodológico y llegando a implementar el modelo estructurado en Casos de Uso.

C. Especificación de Requerimientos.

De acuerdo a lo especificado por el equipo de trabajo de la empresa se especifican los requerimientos en el siguiente cuadro:

Tabla 5. Recopilación de Requerimiento.

REQUERIMIENTO	DESCRIPCIÓN	RESPONSABLE	PRIORIDAD
Gestionar Ventas	Administración, control y almacenamiento de la venta de productos. <ul style="list-style-type: none"> ▪ Ventas al por mayor. ▪ Ventas al por menor. 	<ul style="list-style-type: none"> ▪ Administrador. ▪ Encargado de Ventas. 	Alta
Gestionar Pedido de Ventas	Administración, control y almacenamiento de los pedidos de venta de productos.	<ul style="list-style-type: none"> ▪ Administrador. ▪ Encargado de Ventas. 	Media
Gestionar Compras	Administración, control y almacenamiento de la compra de productos.	<ul style="list-style-type: none"> ▪ Administrador. ▪ Encargado de Compras y Almacén. 	Alta
Gestionar Pedido de Compras	Administración, control y almacenamiento de los pedidos de compra de productos.	<ul style="list-style-type: none"> ▪ Administrador. ▪ Encargado de Compras y Almacén. 	Media
Gestionar Almacén	Control de productos y cantidad existente en almacén y sucursales.	<ul style="list-style-type: none"> ▪ Administrador. ▪ Encargado de Compras y Almacén. 	Alta
Gestionar Distribución de Productos	Administración de la distribución de los productos a las diferentes sucursales y almacenes de la empresa.	<ul style="list-style-type: none"> ▪ Administrador. ▪ Encargado de Compras y Almacén. 	Media
Controlar Movimientos Comerciales	Administración y control de los movimientos comerciales (compras y ventas).	<ul style="list-style-type: none"> ▪ Administrador. 	Media
Gestionar Productos	Registro actualizado de Productos.	<ul style="list-style-type: none"> ▪ Administrador. ▪ Encargado de Ventas. ▪ Encargado de Compras y Almacén. 	Media
Gestionar Clientes	Registro actualizado de Clientes.	<ul style="list-style-type: none"> ▪ Administrador. ▪ Encargado de Ventas. 	Media
Gestionar Proveedores	Registro actualizado de Proveedores.	<ul style="list-style-type: none"> ▪ Administrador. 	Media
Gestionar Sucursales	Registro actualizado de las sucursales y almacenes de la empresa.	<ul style="list-style-type: none"> ▪ Administrador. 	Media

Fuente: Ejecutor.

Fecha: Mayo 2013.

Los requisitos de la tabla anterior se consolidaron en módulos para el sistema agrupándolos de la siguiente forma:

Tabla 6. Agrupación de requerimientos.

MODULO	REALIZA
Módulo de Ventas.	Gestionar Ventas.
Módulo de Compras.	Gestionar Compras.
Módulo de Pedidos.	Gestionar Pedido de Venta.
	Gestionar Pedido de Compra.
Módulo de Almacenamiento.	Gestionar Almacén.
	Gestionar la Distribución de Productos.
Módulo de Administración.	Gestionar Productos.
	Gestionar Proveedores.
	Gestionar Clientes.
	Gestionar Sucursales.
Módulo de Movimientos Comerciales.	Controlar los Movimientos Comerciales.

Fuente: Ejecutor.

Fecha: Mayo 2013.

D. Validación y Certificación de Requerimientos.

Una vez realizadas las entrevistas al personal de la gerencia y encargados de las áreas involucradas de la empresa, se realizó el consolidado de los requerimientos más recurrentes e importantes. Estos requerimientos se procedieron a considerar óptimos y aptos para su desarrollo, siendo documentados, revisados y aceptados.

4.1.1.2. Requerimientos No Funcionales.

- Aplicación de Escritorio para cliente.
- Utilización del lenguaje de programación C#.

- Funcionamiento sobre SQL server.
- Utilización de Web Services.

4.1.2. Modelo Funcional y Diseño del Sistema.

4.1.2.1. Actores del Negocio.

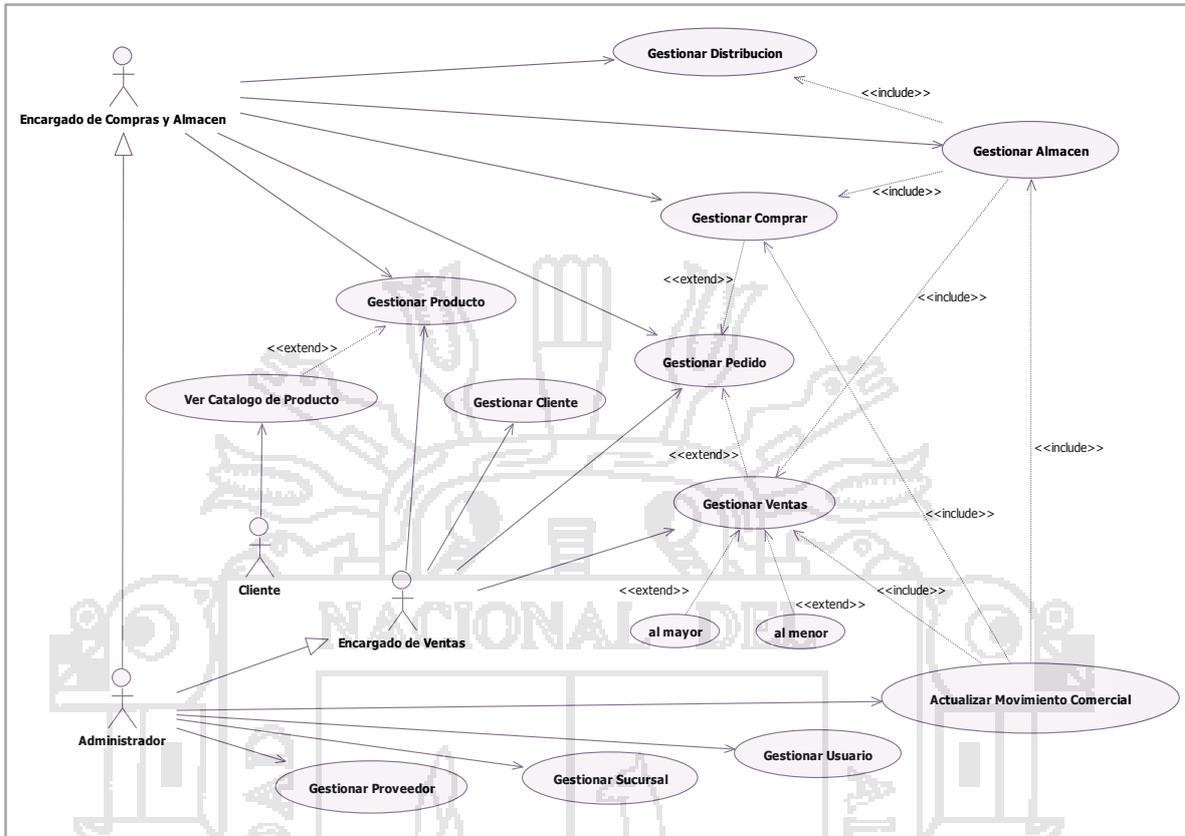
- **Administrador.** Principal actor del sistema, es decir es el encargado directamente del control de todos los módulos del sistema.
- **Encargado de Ventas.** Actor secundario del sistema, con acceso a los módulos: Venta de Productos, Administración de Clientes, y Administración de Productos.
- **Encargado de Compras y Almacén.** Actor secundario del sistema, el cual tiene acceso a los módulos: Compra de Producto, Administración de Productos, Distribución de Productos, y Gestión del Almacén.

4.1.2.2. Diagrama y Especificación de Casos de Uso.

Para determinar las funcionalidades del sistema a ser implementadas acorde con los resultados obtenidos de la etapa anterior, fue necesario hacer un diagrama de casos de uso, el cual progresivamente pasó por etapas de revisión y corrección hasta llegar al diseño de casos de uso que reflejara las funcionalidades principales del sistema en sus módulos.

El siguiente diagrama recoge una vista global de los actores y casos de uso que describen los requisitos funcionales de la aplicación.

Figura 15. Diagrama de Caso de Uso General.

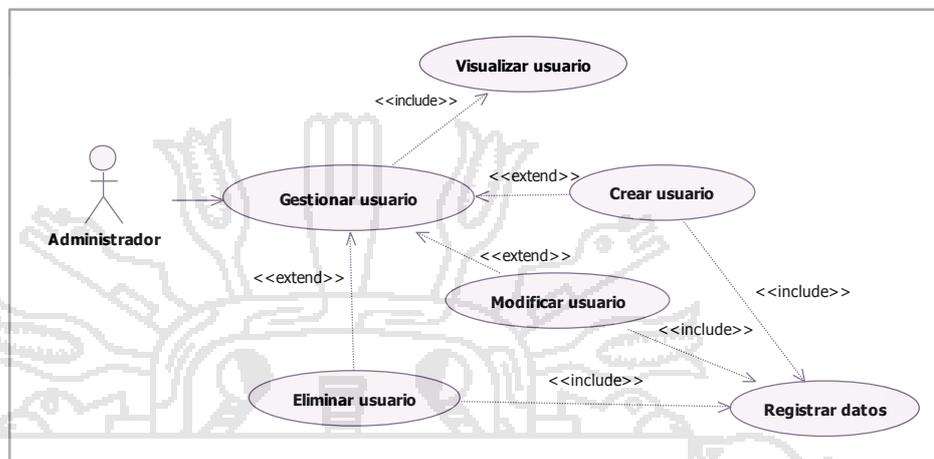


Fuente: Ejecutor.

A. Caso de Uso Gestionar Usuario.

A1. Diagrama de Caso de Uso Gestionar Usuario.

Figura 16. Diagrama de Caso de Uso Gestionar Usuario.



Fuente: Ejecutor.

A2. Especificación del Caso de Uso Gestionar Usuario.

Tabla 7. Especificación del Caso de Uso Gestionar Usuario.

Caso de uso	Gestionar Usuario	
Descripción	Permite definir los criterios para la creación, modificación, y eliminación de los usuarios que hacen uso del sistema.	
Actores	Usuario: Administrador.	
Precondiciones	Estar dentro del sistema autenticado como administrador. Actualizar el registro de usuarios del sistema.	
Flujo de eventos	Administrador	Sistema
	1. Selecciona el menú Gestionar Usuario. 4. Selecciona la opción que quiera realizar.	2. Muestra el listado de los usuarios del sistema. 3. Muestra las opciones de: ver, crear, modificar, y eliminar.

6. Visualiza e ingresa los datos requeridos.	5. Muestra la interfaz de la opción seleccionada.
7. Guarda cambios de la acción realizada.	
9. Finaliza el flujo cuando sale del sistema.	8. Almacena datos.

Flujo alternativo:

En el momento que desee el usuario puede cerrar la vista actual y salir del sistema.

Encadenamiento de errores

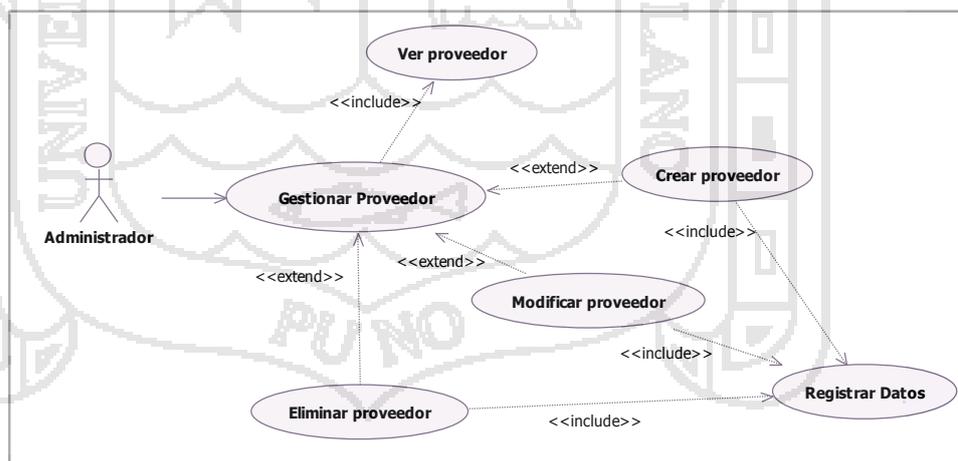
Error al conectarse al servidor, se solicita al usuario volver a intentarlo.
 Error al guardar los datos registrados en la interfaz Gestionar Usuario.

Fuente: Ejecutor.

B. Caso de Uso: Gestionar Proveedor.

B1. Diagrama de Caso de Uso Gestionar Proveedor.

Figura 17. Diagrama de Caso de Uso Gestionar Proveedor.



Fuente: Ejecutor.

B2. Especificación del Caso de Uso Gestionar Proveedor.

Tabla 8. Especificación del Caso de Uso Gestionar Proveedor.

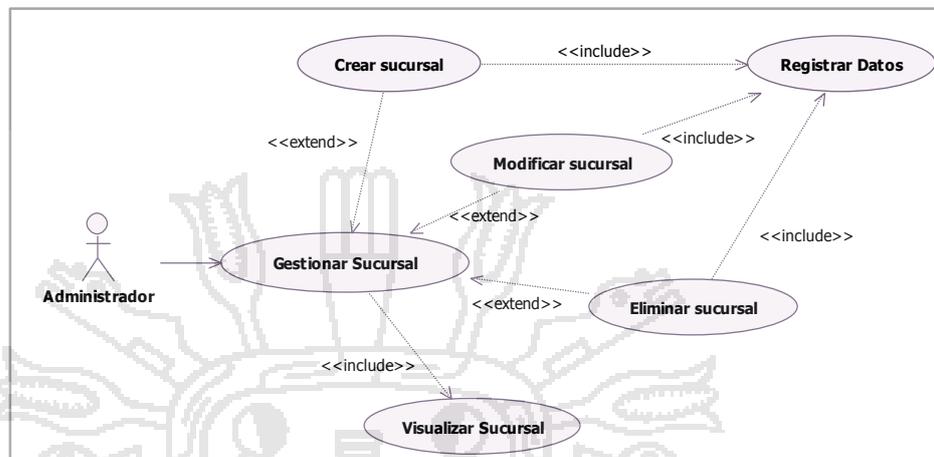
Caso de uso	Gestionar Proveedor	
Descripción	Permite definir los criterios para la creación, modificación, y eliminación de los proveedores de la empresa.	
Actores	Usuario: Administrador.	
Precondiciones	Estar dentro del sistema autenticado como administrador. Actualizar el registro de proveedores en el sistema.	
Flujo de eventos		
	<u>Administrador</u>	<u>Sistema</u>
	<ol style="list-style-type: none"> 1. Selecciona el menú Gestionar Proveedor. 4. Selecciona la opción que quiera realizar. 6. Visualiza e ingresa los datos requeridos. 7. Guarda cambios de la acción realizada. 9. Finaliza el flujo cuando sale del sistema. 	<ol style="list-style-type: none"> 2. Muestra el listado de proveedores de la empresa. 3. Muestra las opciones de: ver, crear, modificar, y eliminar. 5. Muestra la interfaz de la opción seleccionada. 8. Almacena datos.
	Flujo alternativo:	
	En el momento que desee el usuario puede cerrar la vista actual y salir del sistema.	
	Encadenamiento de errores	
	Error al conectarse al servidor, se solicita al usuario volver a intentarlo. Error al guardar los datos registrados en la interfaz Gestionar Proveedor.	

Fuente: Ejecutor.

C. Caso de Uso Gestionar Sucursal.

C1. Diagrama de Caso de Uso Gestionar Sucursal.

Figura 18. Diagrama de Caso de Uso Gestionar Sucursal.



Fuente: Ejecutor.

C1. Especificación del Caso de Uso Gestionar Sucursal.

Tabla 9. Especificación del Caso de Uso Gestionar Sucursal.

Caso de uso	Gestionar Sucursal	
Descripción	Permite definir los criterios para la creación, modificación, y eliminación de las sucursales y almacenes de la empresa.	
Actores	Usuario: Administrador.	
Precondiciones	Estar dentro del sistema autenticado como administrador. Actualizar el registro de sucursales en el sistema.	
Flujo de eventos	Administrador	Sistema
	1. Selecciona el menú Gestionar Sucursal. 4. Selecciona la opción que quiera realizar.	2. Muestra el listado de sucursales y almacenes registrados. 3. Muestra las opciones de: ver crear, modificar, y eliminar. 5. Muestra la interfaz de la opción seleccionada.

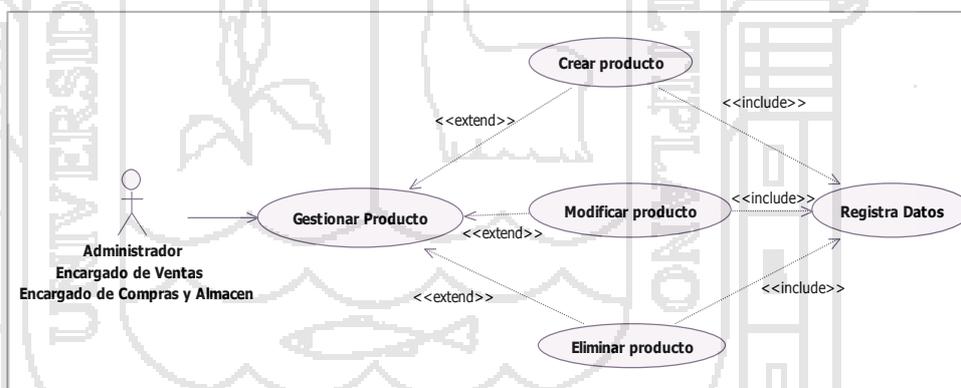
6. Visualiza e ingresa los datos requeridos	8. Almacena datos.
7. Guarda cambios de la acción realizada.	
9. Finaliza el flujo cuando sale del sistema.	
Flujo alterno:	
En el momento que desee el usuario puede cerrar la vista actual y salir del sistema.	
Encadenamiento de errores	
Error al conectarse al servidor, se solicita al usuario volver a intentarlo. Error al guardar los datos registrados en la interfaz Gestionar Sucursal.	

Fuente: Ejecutor.

D. Caso de Uso Gestionar Producto.

D1. Diagrama de Caso de Uso Gestionar Producto.

Figura 19. Diagrama de Caso de Uso Gestionar Producto.



Fuente: Ejecutor.

D2. Especificación del Caso de Uso Gestionar Producto.

Tabla 10. Especificación del Caso de Uso Gestionar Producto.

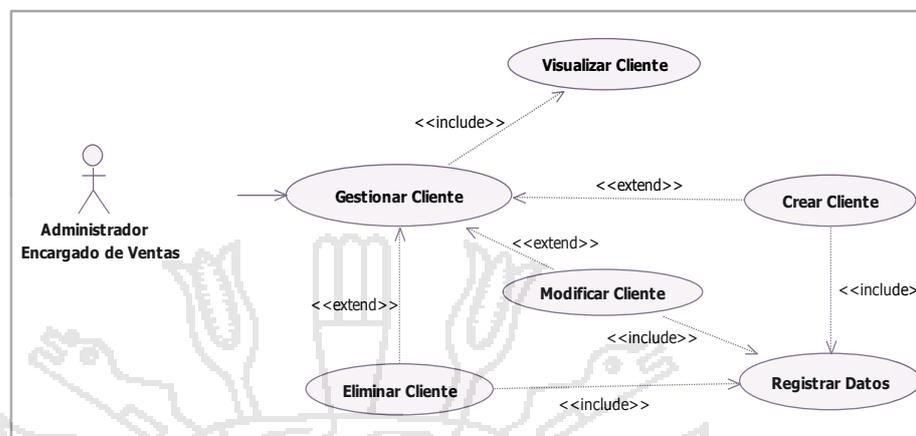
Caso de uso	Gestionar Producto	
Descripción	Permite definir los criterios para la creación, modificación, y eliminación de productos de la empresa.	
Actores	Usuario: Administrador Encargado de Ventas Encargado de Compras y Almacén.	
Precondiciones	Estar dentro del sistema autenticado como Administrador o Encargado de Ventas o Encargado de Compras y Almacén. Actualizar el registro de productos en el sistema.	
Flujo de eventos		
	<u>Administrador o Encargado de Ventas o Encargado de Compras y Almacén</u>	<u>Sistema</u>
	1. Selecciona el menú Gestionar Producto.	2. Muestra el listado de productos registrados.
	4. Selecciona la opción que quiera realizar.	3. Muestra las opciones de: ver, crear, modificar, y eliminar.
	6. Visualiza e ingresa los datos requeridos	5. Muestra la interfaz de la opción seleccionada.
	7. Guarda cambios de la acción realizada.	
	9. Finaliza el flujo cuando sale del sistema.	8. Almacena datos.
Flujo alternativo:		
En el momento que desee el usuario puede cerrar la vista actual y salir del sistema.		
Encadenamiento de errores		
Error al conectarse al servidor, se solicita al usuario volver a intentarlo.		
Error al guardar los datos registrados en la interfaz Gestionar Producto.		

Fuente: Ejecutor.

E. Caso de Uso Gestionar Cliente.

E1. Diagrama de Caso de Uso Gestionar Cliente.

Figura 20. Diagrama de Caso de Uso Gestionar Cliente.



Fuente: Ejecutor.

E2. Especificación del Caso de Uso Gestionar Cliente.

Tabla 11. Especificación del Caso de Uso Gestionar Cliente.

Caso de uso	Gestionar Cliente	
Descripción	Permite definir los criterios para la creación, modificación, y eliminación de clientes de la empresa.	
Actores	Usuario: Administrador. Encargado de Ventas.	
Precondiciones	Estar dentro del sistema autenticado como Administrador o Encargado de Ventas. Actualizar el registro de clientes en el sistema.	
Flujo de eventos		
	<u>Administrador o Encargado de Ventas.</u>	<u>Sistema</u>
	1. Selecciona el menú Gestionar Cliente. 4. Selecciona la opción que quiera realizar.	2. Muestra el listado de clientes registrados. 3. Muestra las opciones de: ver, crear, modificar, y eliminar. 5. Muestra la interfaz de la opción seleccionada.

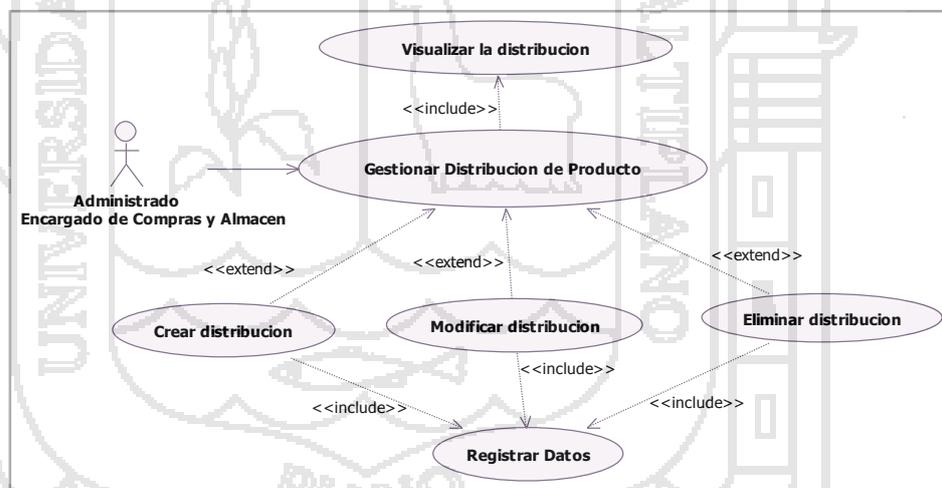
<p>6. Visualiza e ingresa los datos requeridos.</p> <p>7. Guarda cambios de la acción realizada.</p> <p>9. Finaliza el flujo cuando sale del sistema.</p>	<p>8. Almacena datos.</p>
Flujo alterno:	
En el momento que desee el usuario puede cerrar la vista actual y salir del sistema.	
Encadenamiento de errores	
Error al conectarse al servidor, se solicita al usuario volver a intentarlo. Error al guardar los datos registrados en la interfaz Gestionar Cliente.	

Fuente: Ejecutor.

F. Caso de Uso Gestionar Distribuir de Producto.

F1. Diagrama de Caso de Uso Gestionar Distribuir de Producto.

Figura 21. Diagrama de Caso de Uso Gestionar Distribuir de Producto.



Fuente: Ejecutor.

F2. Especificación del Caso de Uso Gestionar Distribuir de Producto.

Tabla 12. Especificación del Caso de Uso Gestionar Distribuir de Producto.

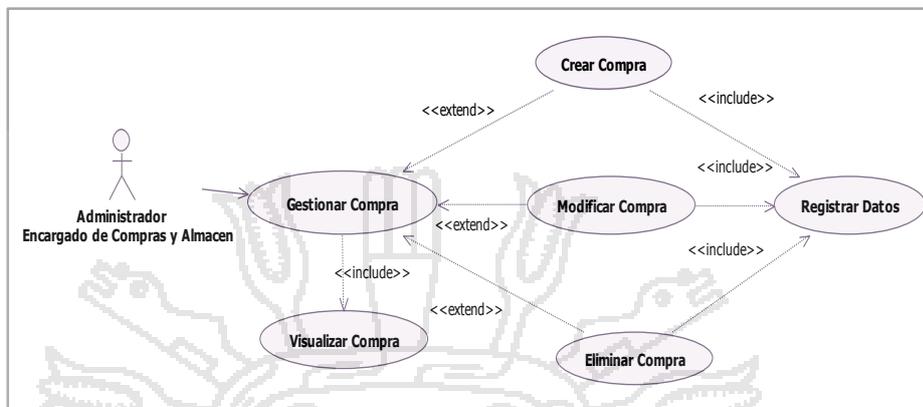
Caso de uso	Gestionar Distribución de Producto	
Descripción	Permite definir los criterios para la distribución de productos a las diferentes sucursales y/o almacenes de la empresa.	
Actores	Usuario: Administrador Encargado de Compras y Almacén.	
Precondiciones	Estar dentro del sistema autenticado como Administrador o Encargado de Compras y Almacén. Existencia de productos a distribuirse o redistribuirse	
Flujo de eventos		
	<u>Administrador o Encargado de Compras y Almacén</u>	<u>Sistema</u>
	<ol style="list-style-type: none"> 1. Selecciona el menú Gestionar la Distribución. 4. Selecciona la opción que quiera realizar. 6. Visualiza e ingresa los datos que requiere la distribución de productos. 7. Guarda cambios de la acción realizada. 9. Finaliza el flujo cuando sale del sistema. 	<ol style="list-style-type: none"> 2. Muestra la distribución de producto por sucursal. 3. Muestra las opciones de: crear, modificar o eliminar. 5. Muestra la interfaz de la opción seleccionada. 8. Almacena datos.
Flujo alterno:		
En el momento que desee el usuario puede cerrar la vista actual y salir del sistema.		
Encadenamiento de errores		
Error al conectarse al servidor, se solicita al usuario volver a intentarlo. Error al guardar los datos registrados en la interfaz de Gestionar la Distribución.		

Fuente: Ejecutor.

G. Caso de Uso Gestionar compra.

G1. Diagrama de Caso de Uso Gestionar compra.

Figura 22. Diagrama de Caso de Uso Gestionar Compra.



Fuente: Ejecutor.

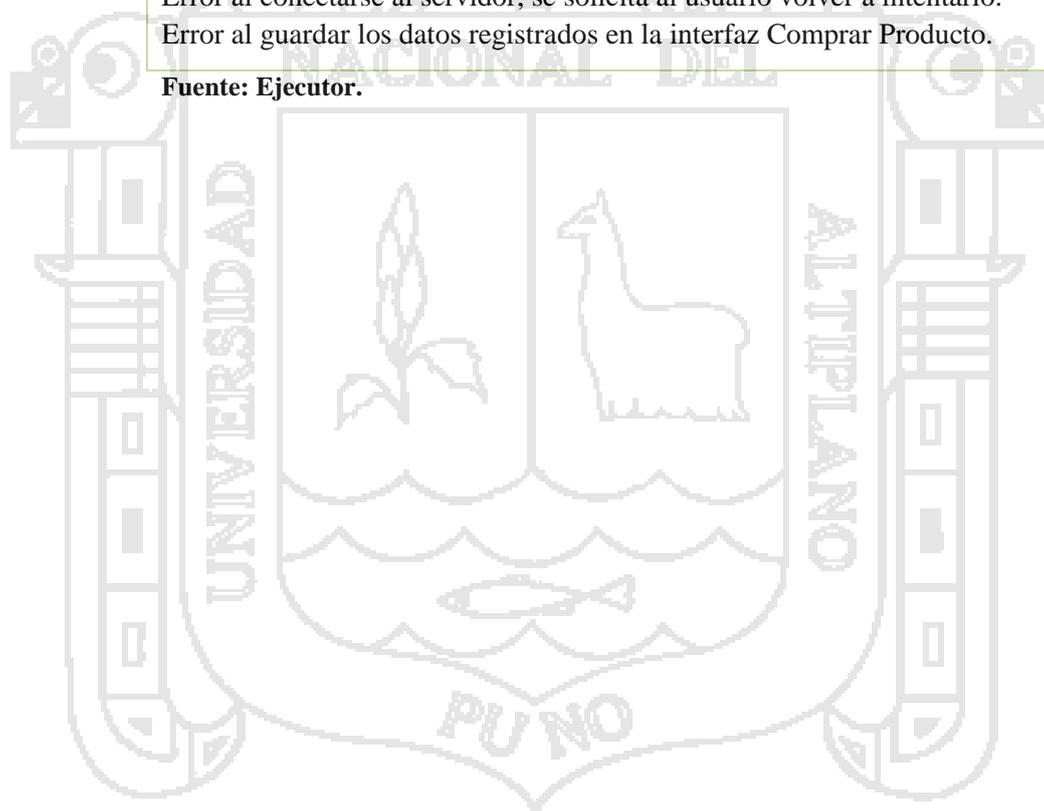
G2. Especificación del Diagrama de Caso de Uso Gestionar Compra.

Tabla 13. Especificación del Caso de Uso Gestionar Compra.

Caso de uso	Gestionar Compra	
Descripción	Permite definir los criterios para realizar las compras de productos al proveedor.	
Actores	Usuario: Administrador Encargado de Compras y Almacén	
Precondiciones	Estar dentro del sistema autenticado como Administrador o Encargado de Compras y Almacén. Verifica la existencia del producto en sucursal/almacén.	
Flujo de eventos		
	<u>Administrador o Encargado de Compras y Almacén</u>	<u>Sistema</u>
	<ol style="list-style-type: none"> 1. Selecciona el menú Comprar Producto. 3. Selecciona la opción que quiera realizar. 	<ol style="list-style-type: none"> 2. Muestra las opciones de: ver, crear modificar, eliminar. 4. Se muestra la interfaz de la opción seleccionada.

<p>5. Visualiza e ingresa los datos requeridos.</p> <p>6. Guarda cambios de la acción realizada</p> <p>8. Finaliza el flujo cuando sale del sistema.</p>	<p>7. Almacena datos.</p>
<p>Flujo alterno:</p>	
<p>En el momento que desee el usuario puede cerrar la vista actual y salir del sistema.</p> <p>Si existen más de 30-unidades/cajas de un producto en el almacén termina el caso de uso.</p> <p>Si no existe productos disponibles por parte del proveedor termina el caso de uso.</p>	
<p>Encadenamiento de errores</p>	
<p>Error al conectarse al servidor, se solicita al usuario volver a intentarlo.</p> <p>Error al guardar los datos registrados en la interfaz Comprar Producto.</p>	

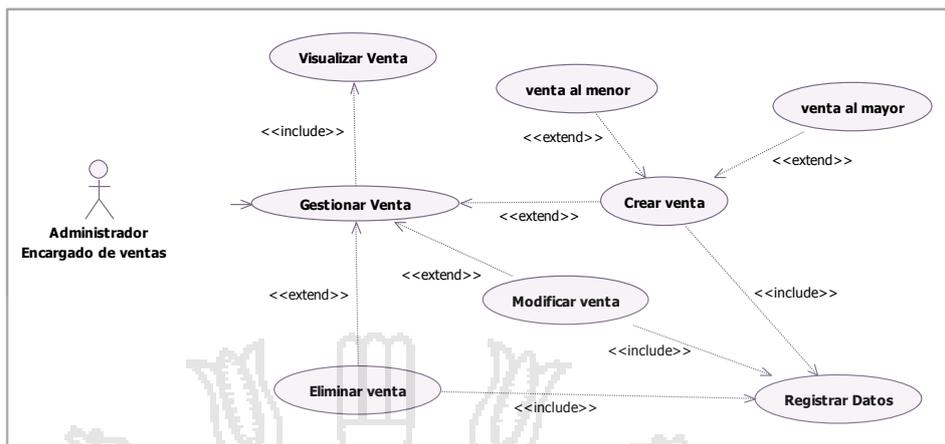
Fuente: Ejecutor.



H. Caso de Uso Gestionar Venta.

H1. Diagrama de Caso de Uso Gestionar Venta.

Figura 23. Diagrama de Caso de Uso Gestionar Venta.



Fuente: Ejecutor.

H2. Especificación del Caso de Uso Gestionar Venta.

Tabla 14. Especificación del Caso de Uso Gestionar Venta.

Caso de uso	Gestionar Venta	
Descripción	Permite definir los criterios para realizar las ventas de productos al cliente. Este caso de uso se extiende por los casos de uso al mayor y al menor.	
Actores	Usuario: Administrador. Encargado de Ventas.	
Precondiciones	Estar dentro del sistema autenticado como Administrador o Encargado de Ventas. Verifica la existencia del producto en sucursal/almacén.	
Flujo de Eventos	<u>Administrador o Encargado de Ventas</u>	<u>Sistema</u>
	1. Selecciona el menú Gestionar Venta.	2. Muestra las opciones de: ver, crear modificar, eliminar.
	3. Selecciona la opción que quiera realizar.	4. Muestra Puntos de Extensión 1 y 2.
	6. Visualiza e ingresa los datos requeridos.	5. Muestra la interfaz de la opción seleccionada.



7. Guarda cambios de la acción realizada.	
9. Finaliza el flujo cuando sale del sistema.	8. Almacena datos.
Punto de Extensión	
<ul style="list-style-type: none"> ❖ PE1. Venta al menor. ❖ PE2. Venta al mayor. 	
Flujo de Eventos de Puntos de Extensión	
PE 1. Venta al menor	
1.1. Selecciona la opción Venta al menor.	
	1.2. Muestra la interfaz de venta al menor.
1.3. Ingresar los datos requeridos para la venta.	
1.4. Verifica el precio del producto unitario.	
1.5. Guarda los detalles de la venta al menor.	1.6. Almacena datos.
PE 2. Venta al mayor	
2.1. Selecciona la opción Venta al mayor.	
	2.2. Muestra la interfaz de venta al mayor.
2.3. Verifica adeudo de cliente.	
2.4. Ingresar los datos requeridos.	
2.5. Verifica el precio del producto al mayor.	
2.6. Guarda los detalles de la venta al mayor.	2.7. Almacena datos.
Flujo alterno:	
En el momento que desee el usuario puede cerrar la vista actual y salir del sistema.	
Encadenamiento de errores	

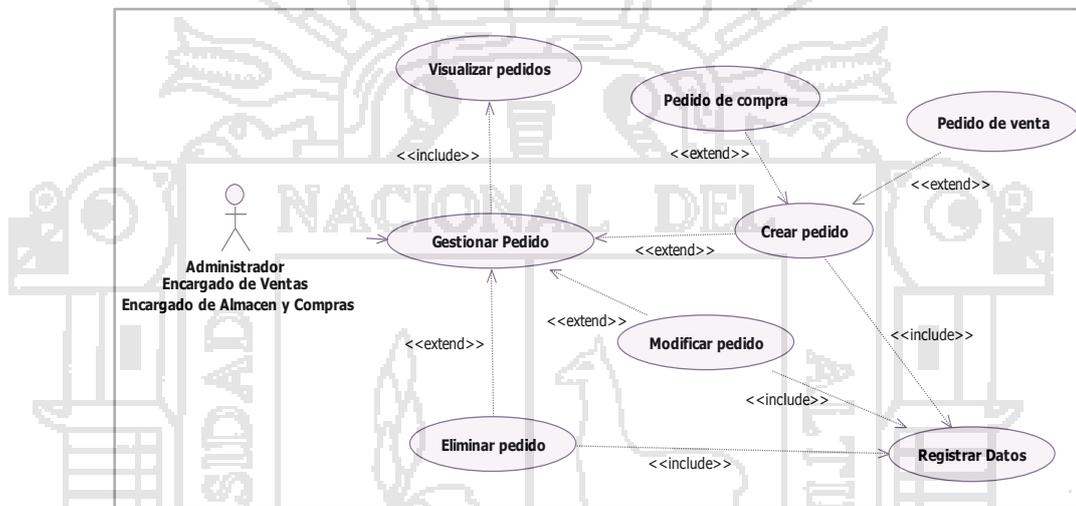
Error al conectarse al servidor, se solicita al usuario volver a intentarlo.
 Error al guardar los datos registrados en la interfaz Gestionar Venta.
 El usuario no seleccione tipo de venta (puntos de extensión 1y 2).
 Si la venta es 350 soles, es considerada venta por mayor.

Fuente: Ejecutor.

I. Caso de Uso Gestionar Pedido.

II. Diagrama de Caso de Uso Gestionar Pedido.

Figura 24. Diagrama de Caso de Uso Gestionar Pedido.



Fuente: Ejecutor.

II. Especificación del Caso de Uso Gestionar Pedido.

Tabla 15. Especificación del Caso de Uso Gestionar Pedido.

Caso de uso	Gestionar Pedido
Descripción	Permite definir los criterios para realizar pedidos de productos al cliente o proveedor. Este caso de uso se extiende por los casos de uso pedido de compra y pedidos de venta.
Actores	Usuario: Administrador. Encargado de Ventas. Encargado de Almacén y Compras.

Precondiciones	Estar dentro del sistema autenticado como Administrador o Encargado de Ventas o Encargado de Almacén y Compras, según sea el caso. Verifica la existencia del producto en sucursal/almacén.	
Flujo de Eventos		
<u>Administrador o Encargado de Ventas o Encargado de Almacén y Compras</u>	<u>Sistema</u>	
<ol style="list-style-type: none"> 1. Selecciona el menú Gestionar Pedido. 3. Selecciona la opción que quiera realizar. 6. Visualiza e ingresa los datos requeridos. 7. Guarda cambios de la acción realizada. 9. Finaliza el flujo cuando sale del sistema. 	<ol style="list-style-type: none"> 2. Muestra las opciones de: ver, crear modificar, eliminar. 4. Muestra Puntos de Extensión 1 y 2. 5. Muestra la interfaz de la opción seleccionada. 8. Almacena datos. 	
Punto de Extensión		
<ul style="list-style-type: none"> ❖ PE1. Pedido de Compra. Usuario: Administrador. Encargado de Almacén y Compras. ❖ PE2. Pedido de Venta. Usuario: Administrador. Encargado de Ventas. 		
Flujo de Eventos de Puntos de Extensión PE 1. Pedido de Compra		
<u>Administrador o Encargado de Almacén y Compras</u>	<u>Sistema</u>	
<ol style="list-style-type: none"> 1.1. Selecciona la opción Pedido de Compra. 	<ol style="list-style-type: none"> 1.2. Muestra la interfaz de pedidos de compra. 	

<p>1.4. Ingresar los datos requeridos para la compra.</p> <p>1.5. Guardar los detalles del pedido de compra.</p>	<p>1.3. Muestra los pedidos pendientes de compra.</p> <p>1.6. Almacena datos.</p>
--	--

PE 2. Pedido de Venta

<u>Administrador o Encargado de Ventas</u>	<u>Sistema</u>
<p>2.1. Selecciona la opción Pedidos de Venta.</p> <p>2.3. Verifica adeudo de cliente</p> <p>2.4. Ingresar los datos requeridos para el pedido de venta.</p> <p>2.5. Guardar los detalles del pedido de venta.</p>	<p>2.2. Muestra la interfaz de pedido de venta.</p> <p>2.6. Almacena datos.</p>

Flujo alterno:

En el momento que desee el usuario puede cerrar la vista actual y salir del sistema.

Encadenamiento de errores

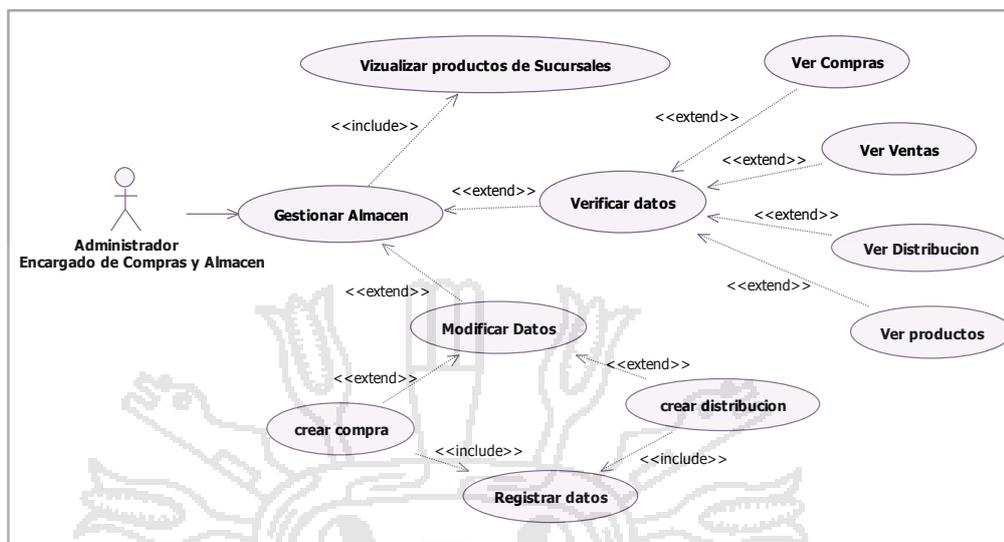
Error al conectarse al servidor, se solicita al usuario volver a intentarlo.
 Error al guardar los datos registrados en la interfaz Gestionar Pedidos.
 El usuario no seleccione tipo de pedido (puntos de extensión 1y 2).

Fuente: Ejecutor.

J. Caso de Uso Gestionar Almacén.

J1. Diagrama de Caso de Uso Gestionar Almacén.

Figura 25. Diagrama de Caso de Uso Gestionar Almacén.



Fuente: Ejecutor.

J2. Especificación del Caso de Uso Gestionar Almacén.

Tabla 16. Especificación del Caso de Uso Gestionar Almacén.

Caso de uso	Gestionar Almacén
Descripción	Permite definir los criterios para llevar el control de entradas y salidas de productos de una sucursal/almacén.
Actores	Usuario: Administrador. Encargado de Compras y Almacén.
Precondiciones	Estar dentro del sistema autenticado como Administrador o Encargado de Compras y Almacén. Verifica la existencia del producto en sucursal/almacén. Verificar la entrada o salida de un producto.
Flujo de eventos	
<u>Administrador o Encargado de Compras y Almacén</u>	<u>Sistema</u>
<ol style="list-style-type: none"> El usuario selecciona el menú Gestionar Almacén. Selecciona la opción sucursal o almacén. 	<ol style="list-style-type: none"> Muestra las sucursales y almacenes de la empresa. Muestra las opciones de ver producto por sucursal o almacén.

<p>5. El usuario selecciona la opción que quiera realizar.</p> <p>7. Verifica e ingresa los datos requeridos según la opción seleccionada.</p> <p>8. Finaliza el flujo cuando sale del sistema.</p>	<p>ver movimientos de: compras, ventas, distribución.</p> <p>6. Muestra la interfaz de la opción seleccionada.</p>
<p>Puntos de Inclusión</p>	
<p>❖ PI 1. Comprar Producto. ❖ PI 2. Vender Producto. PI 3. Distribución de Producto.</p>	
<p>Flujo de Eventos de Puntos de Inclusión</p> <p>PI 1. Comprar Producto</p>	
<p style="text-align: center;"><u>Administrador o Encargado de Compras y Almacén</u></p> <p>1.1. Selecciona Ver Compras.</p> <p>1.3. Verifica los datos de ingresos y stock de productos, si la existencia por almacén o sucursal está por debajo de lo permitido programa compra.</p> <p>1.5. Ingresar los datos requeridos.</p> <p>1.6. Guarda cambios de la acción realizada.</p> <p>1.8. Retorna al menú principal de Gestionar Almacén.</p>	<p style="text-align: center;"><u>Sistema</u></p> <p>1.2. Muestras el listado de compras e ingreso a almacén o sucursal.</p> <p>1.4. Muestra interfaz de crear compra.</p> <p>1.7. Almacena datos.</p>
<p>PI 2.Vender Producto</p>	
<p style="text-align: center;"><u>Administrador o Encargado de Compras y Almacén</u></p> <p>2.1. Selecciona Ver Ventas.</p>	<p style="text-align: center;"><u>Sistema</u></p> <p>2.2. Muestra el listado de ventas y salidas de almacén.</p>



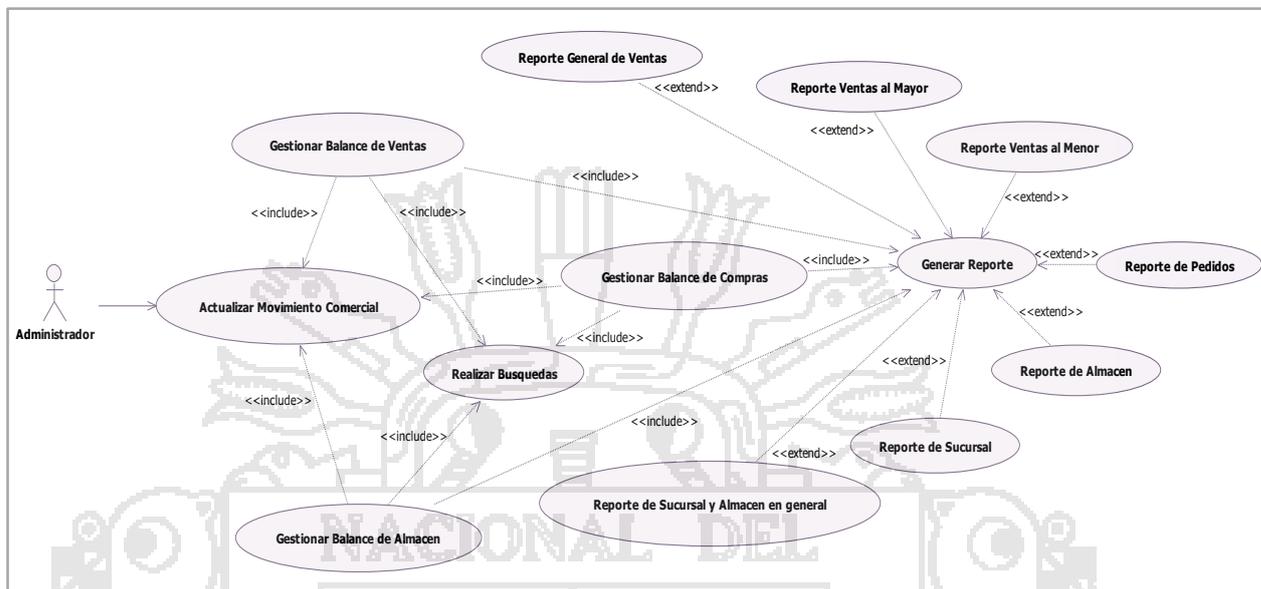
<p>2.3. Verifica los datos de salidas de almacén y sucursales.</p> <p>2.4. Retorna al menú principal de Gestionar Almacén.</p>	
<p>PI 3. Distribuir Producto</p>	
<p style="text-align: center;"><u>Administrador o Encargado de Compras y Almacén</u></p> <p>3.1. Selecciona Ver Distribuciones.</p> <p>3.3. Verifica los datos de distribuciones y stock de productos por sucursales, si la existencia por sucursal está por debajo de lo permitido realiza distribución.</p> <p>3.5. Ingresar los datos requeridos para la distribución.</p> <p>3.6. Guarda cambio de la acción realizada.</p> <p>3.8. Retorna al menú principal de Gestionar Almacén.</p>	<p style="text-align: center;"><u>Sistema</u></p> <p>3.2. Muestra el listado de distribuciones, movimientos de traslados entre sucursales y almacenes.</p> <p>3.4. Muestra interfaz de crear distribución.</p> <p>3.7. Almacena datos.</p>
<p>Flujo alterno:</p>	
<p>En el momento que desee el usuario puede cerrar la vista actual y salir del sistema.</p>	
<p>Encadenamiento de errores</p>	
<p>Error al conectarse al servidor, se solicita al usuario volver a intentarlo. Error al guardar los datos de las interfaces de Gestionar Almacén.</p>	

Fuente: Ejecutor.

K. Caso de Uso Gestionar el Movimiento Comercial.

K1. Diagrama de Caso de Uso Gestionar el Movimiento Comercial.

Figura 26. Diagrama de Caso de Uso Gestionar Movimiento Comercial.



Fuente: Ejecutor.

K2. Especificación del Caso de Uso Gestionar el Movimiento Comercial.

Tabla 17. Especificación del Caso de Uso Gestionar el Movimiento Comercial.

Caso de uso	Gestionar el Movimiento Comercial
Descripción	Permite definir los criterios para llevar el control de compras y ventas de productos de una sucursal/almacén o de la empresa en general de forma diaria, semanal, mensual y anual. Este caso de uso tiene puntos de inclusión de compra, venta y gestión de almacén.
Actores	Usuario: Administrador.
Precondiciones	Estar dentro del sistema autenticado como Administrador. Requiere Visualizar los movimientos comerciales de la empresa.
Flujo de eventos	
<u>Administrador</u>	<u>Sistema</u>
1. Selecciona el menú Movimiento Comercial.	

<p>3. Selecciona la opción que quiera realizar.</p> <p>6. Selecciona la opción que requiera.</p> <p>8. Genera o no los reportes según sus necesidades.</p> <p>9. Finaliza el flujo cuando sale del sistema.</p>	<p>2. Muestra las opciones de Balance de Ventas, Balance de Compras, Balance de Almacén, Balance General.</p> <p>4. Muestra el formulario dependiendo de la opción seleccionada.</p> <p>5. Muestra las opciones de: búsqueda detallada, Generar Reporte, Regresar al Menú Movimiento comercial.</p> <p>7. Muestra la interfaz de la opción seleccionada.</p>
<p>Puntos de Inclusión</p>	
<ul style="list-style-type: none"> ❖ PI 1. Vender Producto ❖ PI 2. Comprar Producto ❖ PI 3. Gestionar Almacén 	
<p>Flujos de Eventos de Puntos de Inclusión</p>	
<p>PI 1. Vender Producto</p>	
<p style="text-align: center;"><u>Administrador</u></p> <p>1. Selecciona Balance de Ventas.</p> <p>3. Selecciona la opción deseada.</p> <p>6. Ingresar los parámetros según la opción requerida.</p> <p>7. Solicita generar reporte.</p> <p>9. Retorna al menú principal de Movimiento Comercial.</p>	<p style="text-align: center;"><u>Sistema</u></p> <p>2. Muestra opciones de Balance del Movimiento Comercial de Ventas: Mayor, Menor, General.</p> <p>4. Muestra la interfaz de la opción solicitada.</p> <p>5. Muestra las opciones para ordenar y buscar las ventas realizadas según detalle requerido.</p> <p>8. Genera Reporte de la información solicitada.</p>
<p>PI2. Comprar Producto.</p>	
<p style="text-align: center;"><u>Administrador</u></p>	<p style="text-align: center;"><u>Sistema</u></p>

<p>1. Selecciona Balance de Compras.</p> <p>3. El usuario selecciona la opción deseada.</p> <p>6. Ingresa los parámetros según la opción requerida.</p> <p>7. Solicita generar reporte.</p> <p>9. Retorna al menú principal de Movimiento Comercial.</p>	<p>2. Muestra la opción de Balance del Movimiento Comercial de Compras.</p> <p>4. Muestra la interfaz de la opción solicitada.</p> <p>5. Muestra las opciones para ordenar y buscar las compras realizadas según detalle requerido.</p> <p>8. Genera Reporte de la información solicitada.</p>
--	--

PI 3. Gestionar Almacén

<u>Administrador</u>	<u>Sistema</u>
<p>1. Selecciona Balance de Almacén.</p> <p>3. Selecciona la opción deseada.</p> <p>6. Ingresa los parámetros según la opción requerida.</p> <p>7. Solicita generar reporte.</p> <p>9. Retorna al menú principal de Movimiento Comercial.</p>	<p>2. Muestra opciones del Balance del Movimiento Comercial de: Sucursal, Almacén, General.</p> <p>4. Muestra la interfaz de la opción solicitada.</p> <p>5. Muestra las opciones para ordenar y buscar la existencia de productos según detalle requerido.</p> <p>8. Genera Reporte final.</p>

Flujo alterno:

En el momento que desee el usuario puede cerrar la vista actual y salir del sistema.
 El usuario selecciona una opción de los puntos de inclusión.
 El usuario desea salir del menú Movimiento Comercial.
 El usuario selecciona la opción regresar al Menú Anterior.
 El parámetro ingresado no está registrado en el sistema.

Encadenamiento de errores

Error al conectarse al servidor, se solicita al usuario volver a intentarlo.
Error al guardar los datos registrados en las interfaces de Movimiento Comercial.
El usuario no seleccione ningún modulo.
El usuario no selecciono ninguna opción del menú.
El usuario no ingreso los datos que se necesiten.
Los datos ingresados son incorrectos. Inténtelo nuevamente.
Ingrese los datos que se requieren por favor.

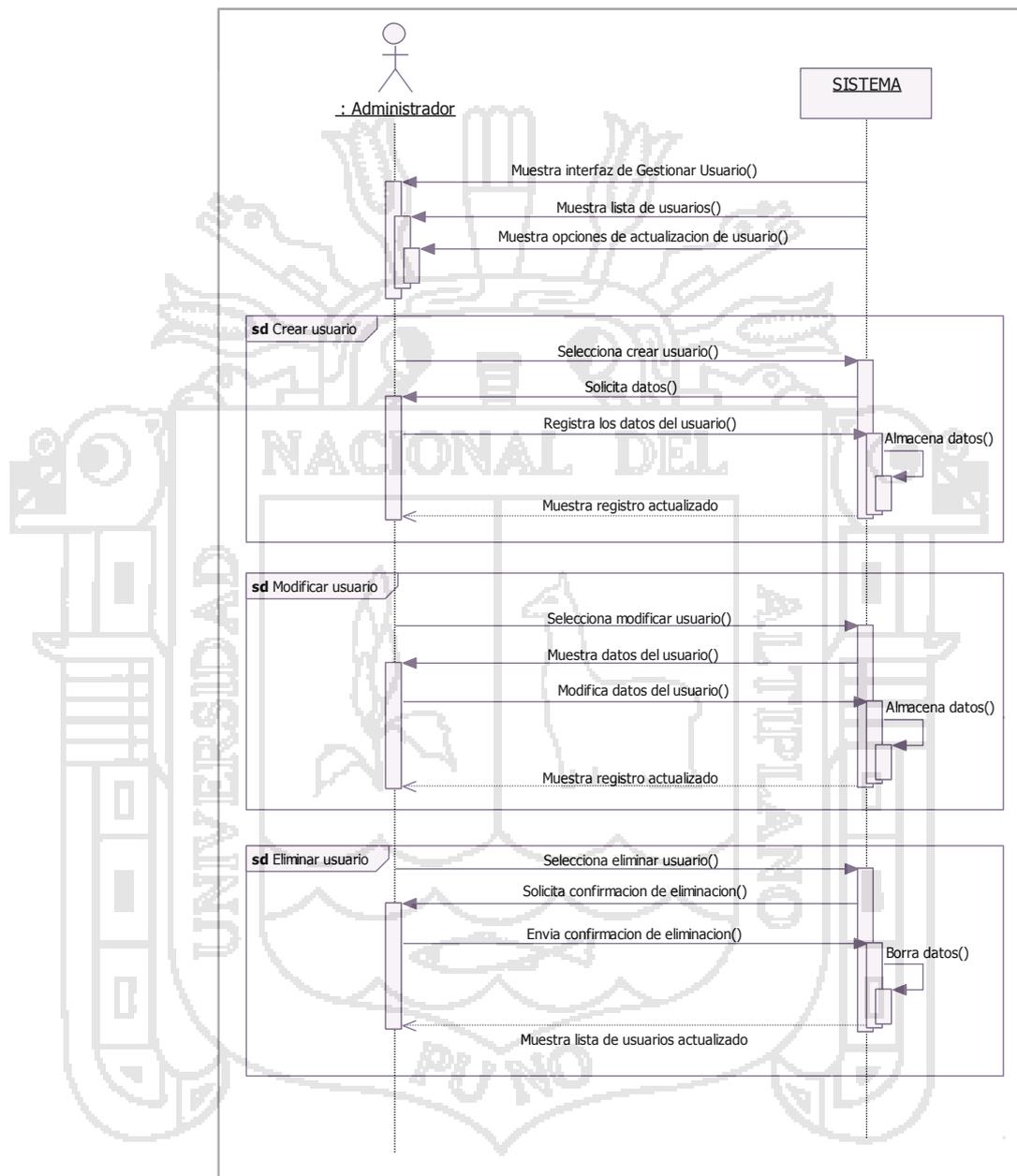
Fuente: Ejecutor.



4.1.2.3. Diagramas de Secuencia.

A. Diagrama de Secuencia Gestionar Usuario.

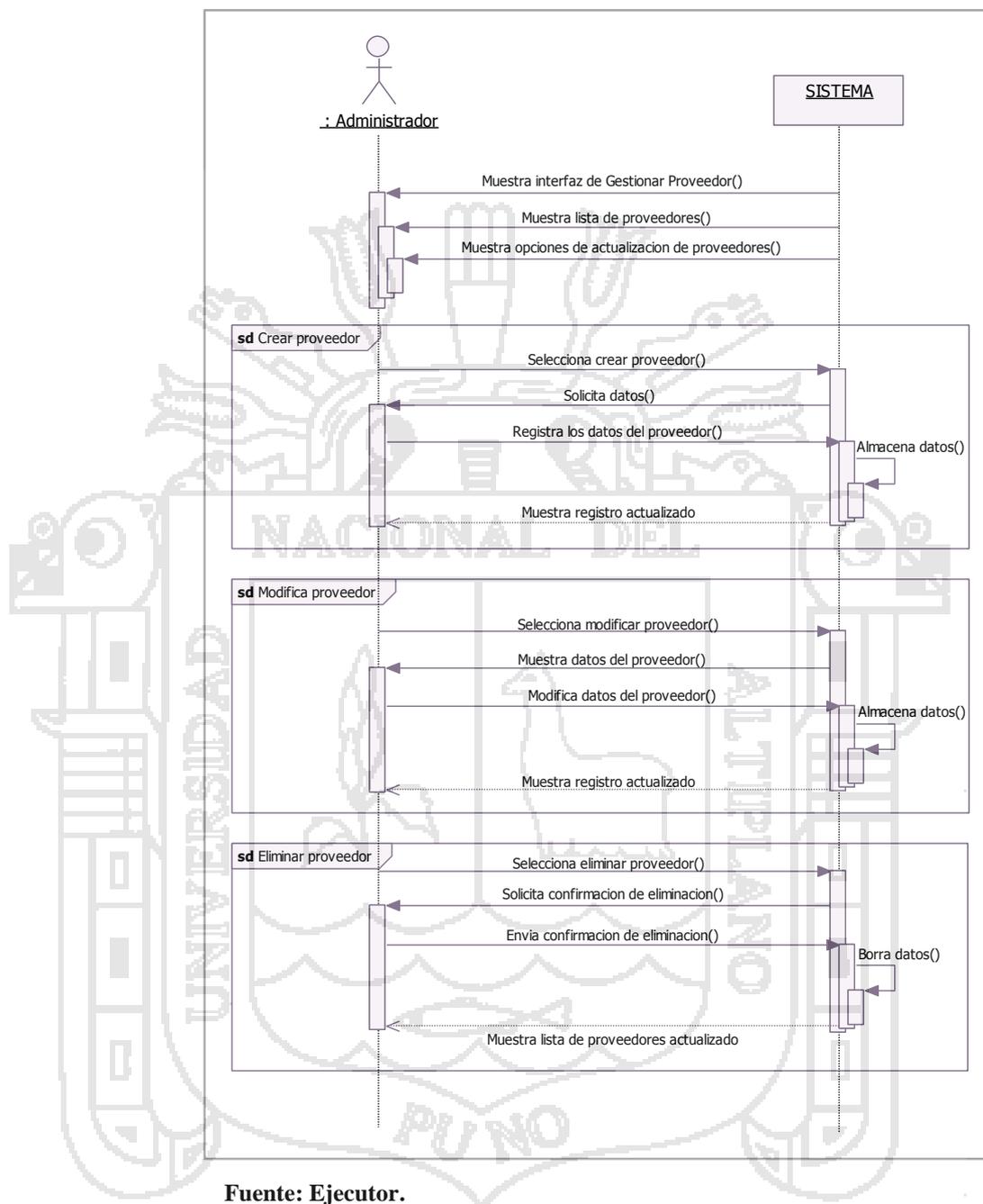
Figura 27. Diagrama de Secuencia Gestionar Usuario.



Fuente: Ejecutor.

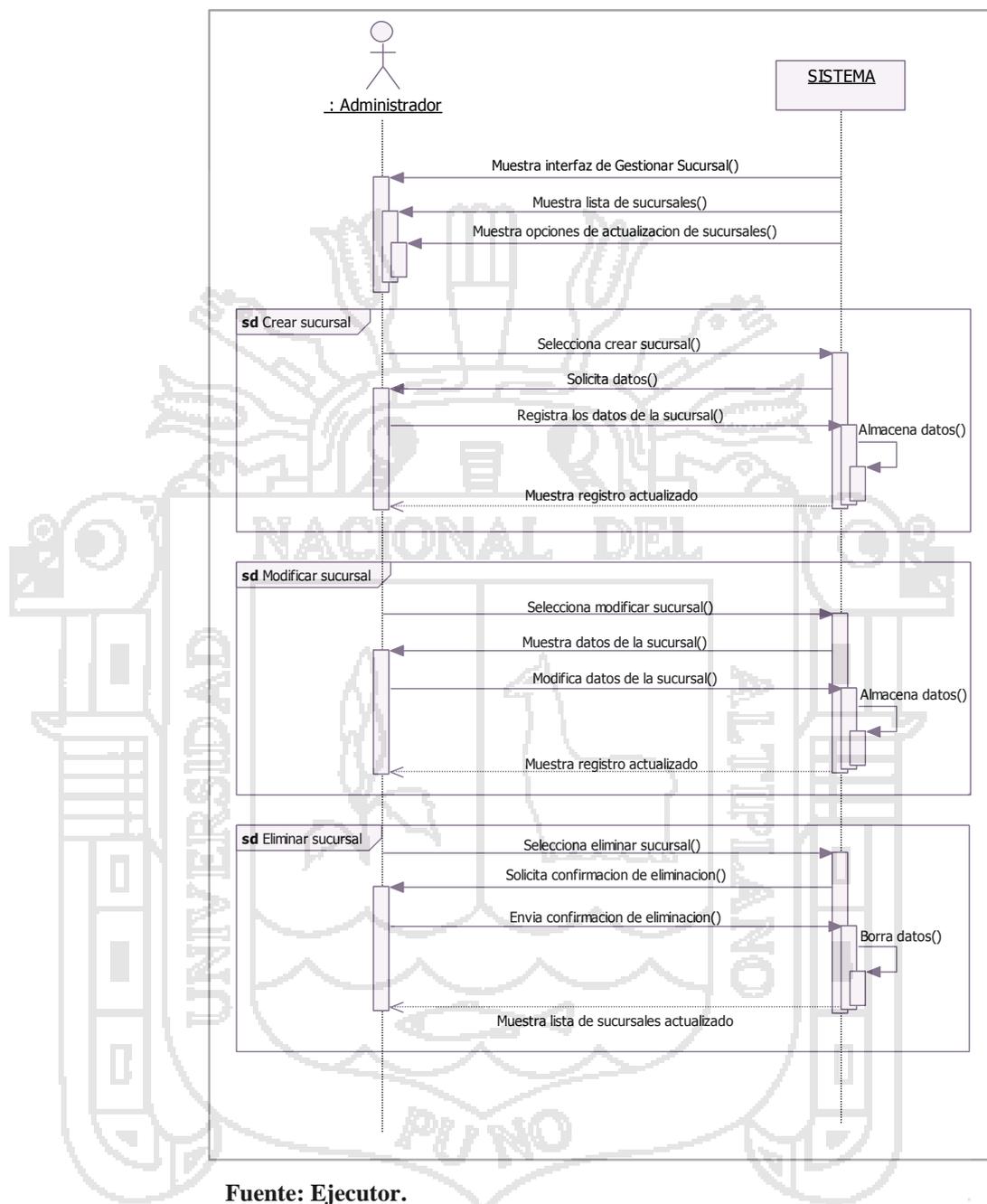
B. Diagrama de Secuencia Gestionar Proveedor.

Figura 28. Diagrama de Secuencia Gestionar Proveedor.



C. Diagrama de Secuencia Gestionar Sucursal.

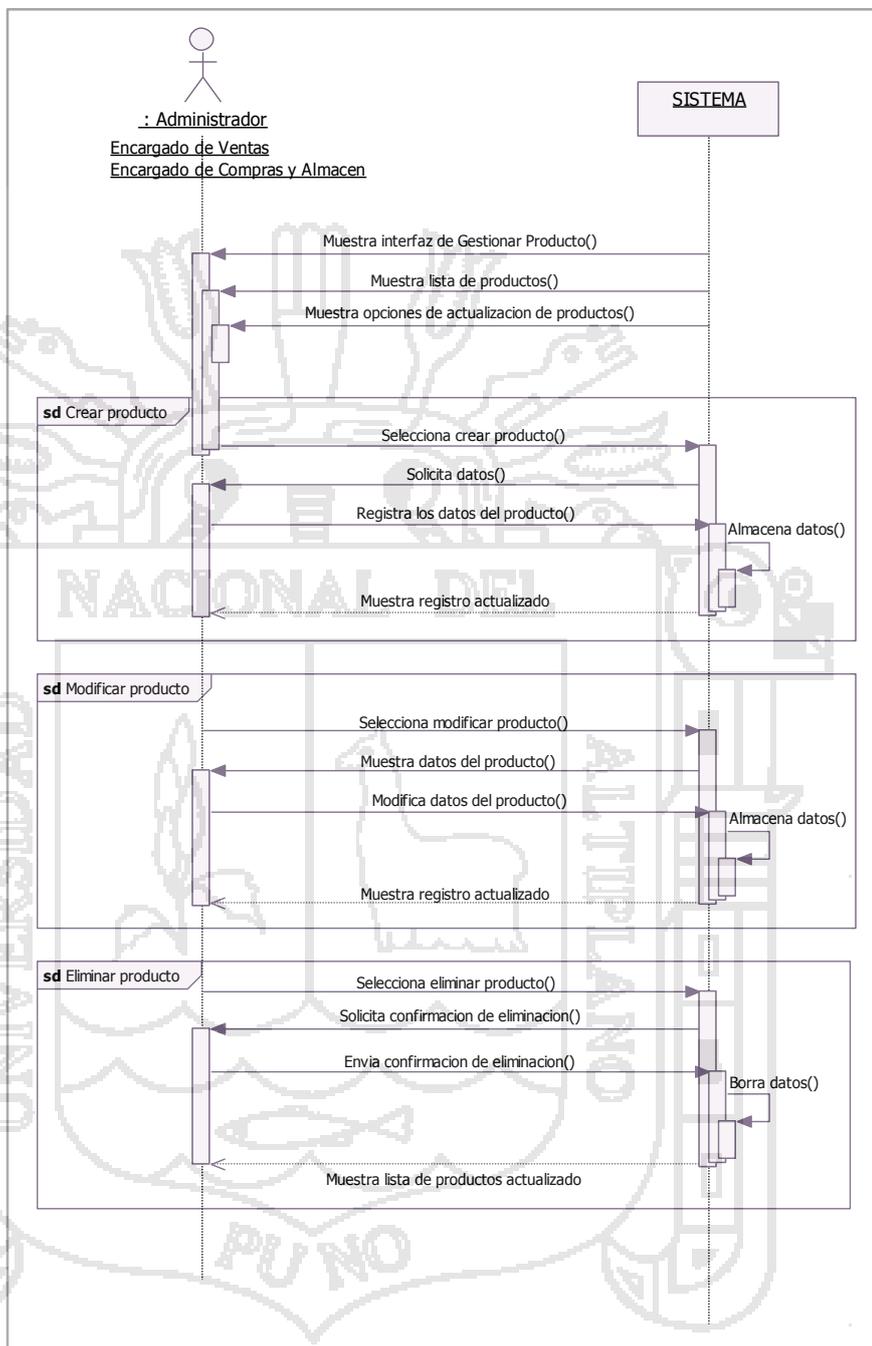
Figura 29. Diagrama de Secuencia Gestionar Sucursal.



Fuente: Ejecutor.

D. Diagrama de Secuencia Gestionar Producto.

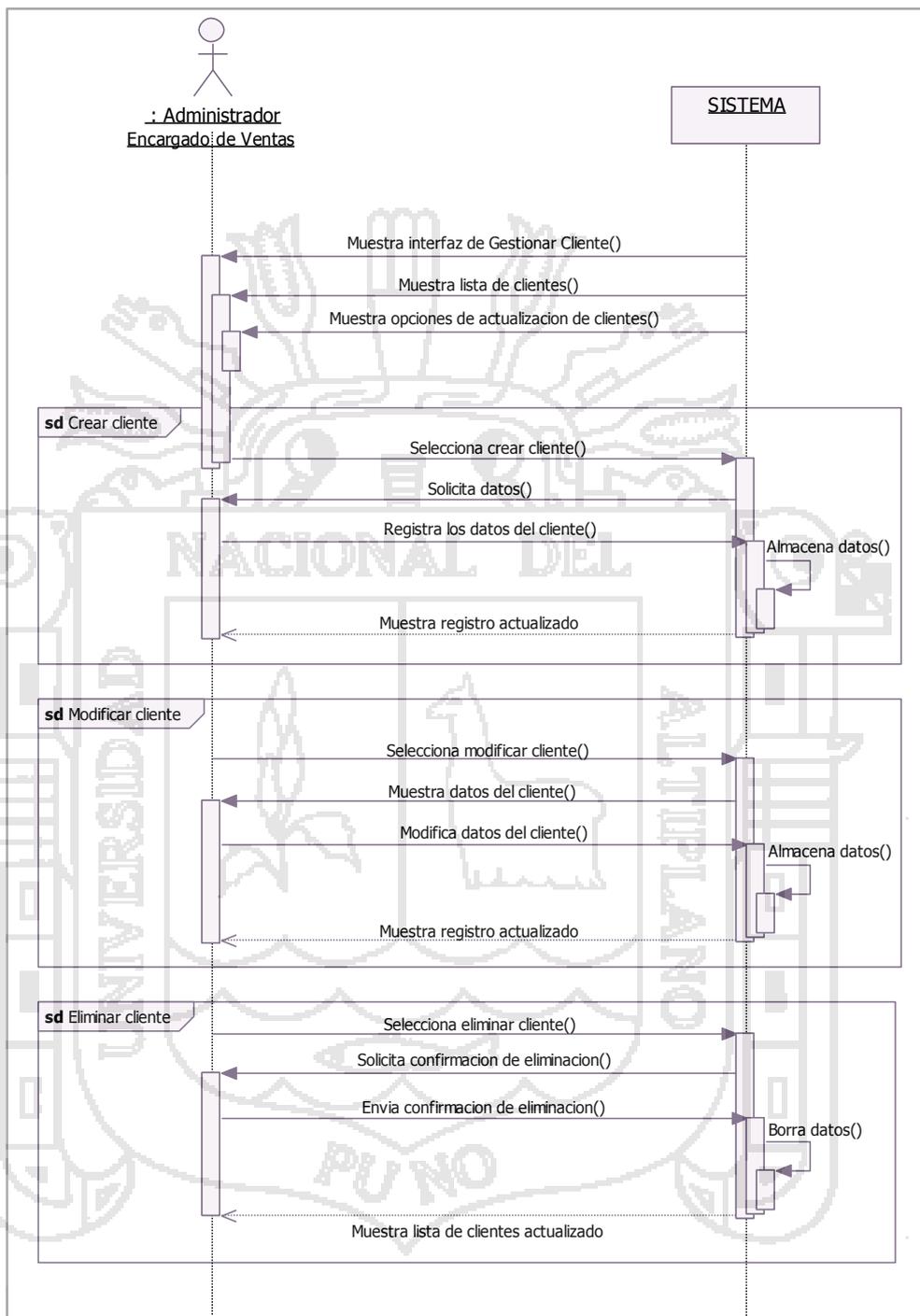
Figura 30. Diagrama de Secuencia Gestionar Producto.



Fuente: Ejecutor.

E. Diagrama de secuencia Gestionar Cliente.

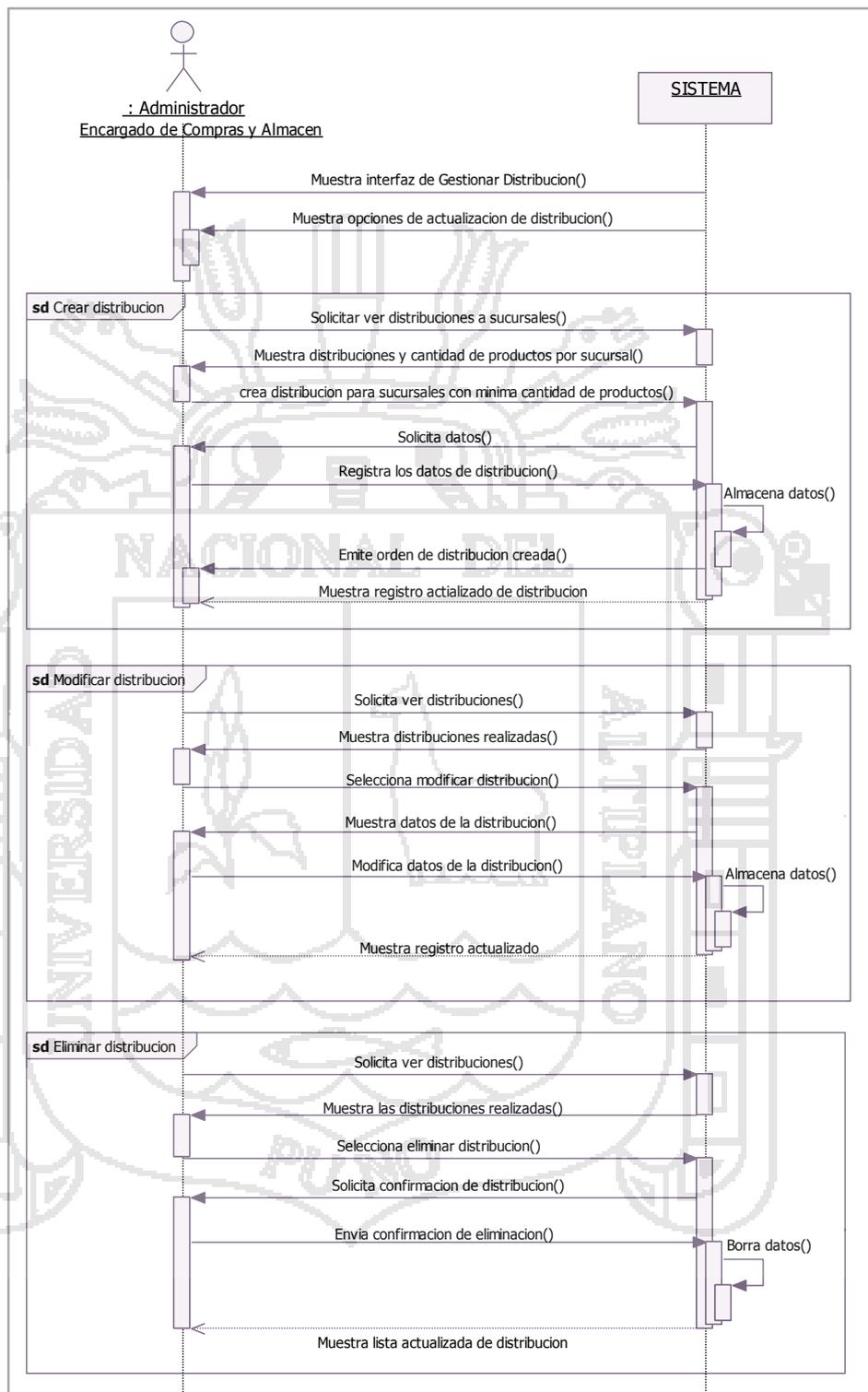
Figura 31. Diagrama de Secuencia Gestionar Cliente.



Fuente: Ejecutor.

F. Diagrama de Secuencia Gestionar Distribución.

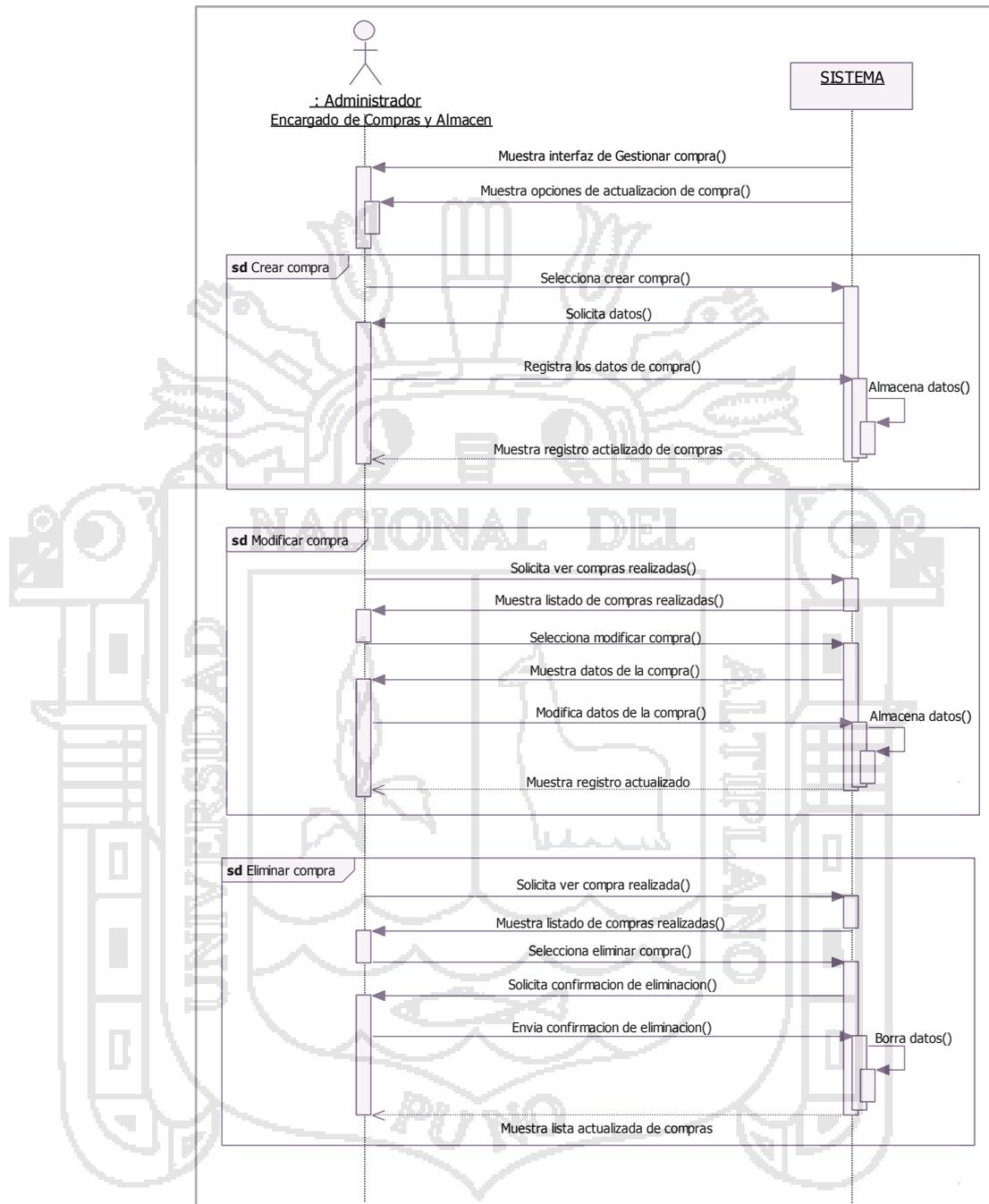
Figura 32. Diagrama de Secuencia Gestionar Distribución.



Fuente: Ejecutor.

G. Diagrama de secuencia Gestionar Compras.

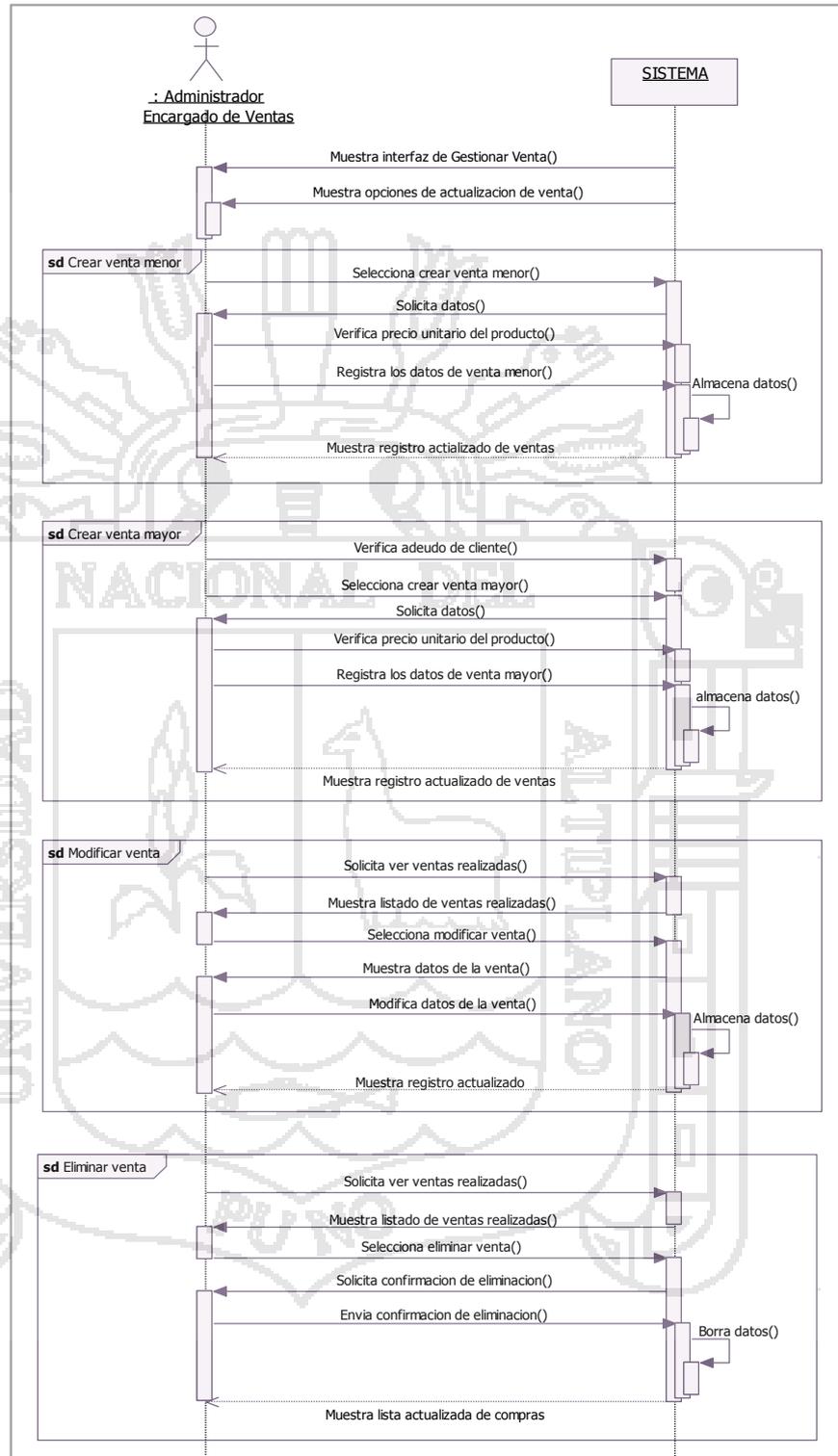
Figura 33. Diagrama de Secuencia Gestionar Compras.



Fuente: Ejecutor.

H. Diagrama de Secuencia Gestionar Venta.

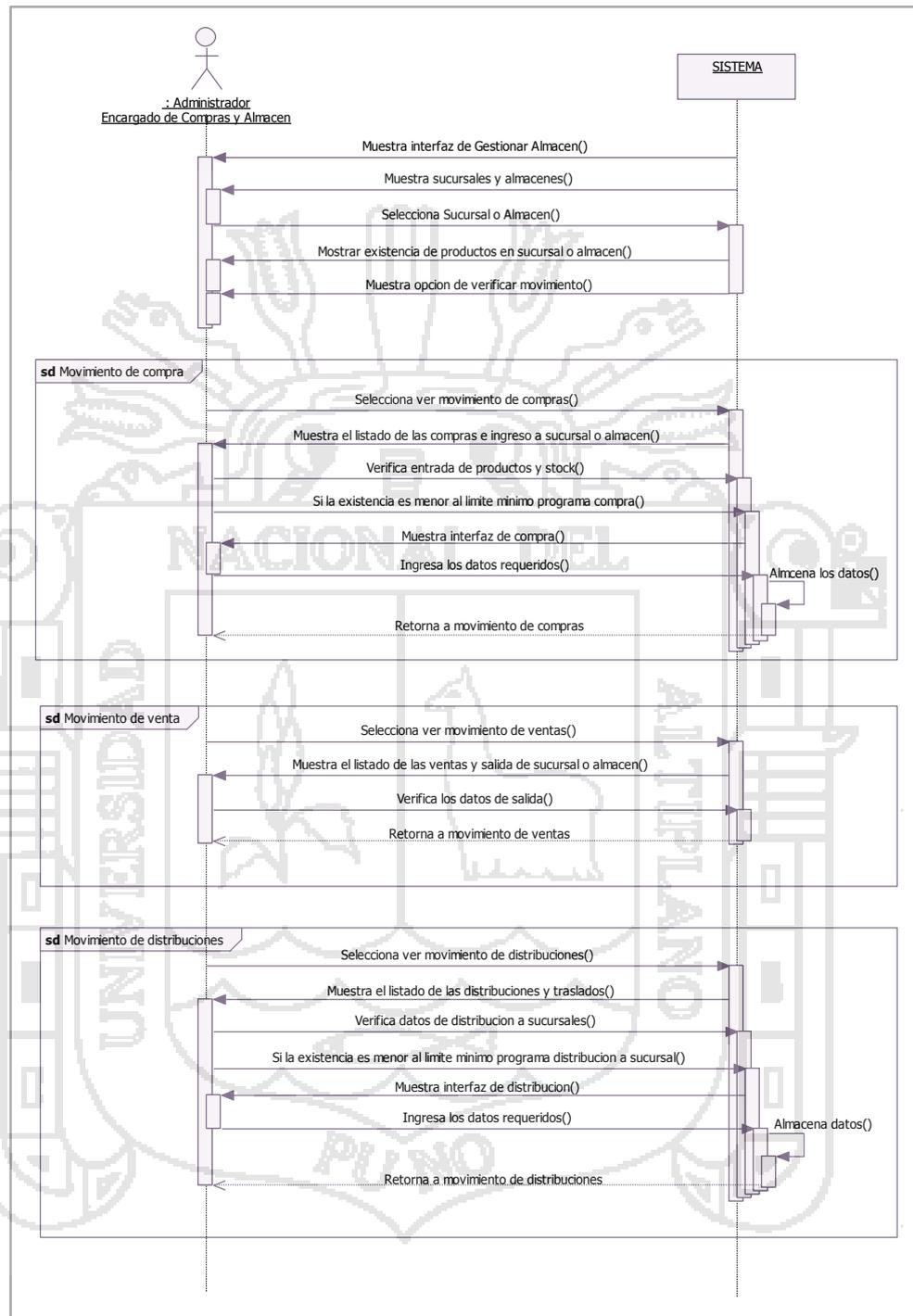
Figura 34. Diagrama de Secuencia Gestionar Venta.



Fuente: Ejecutor.

I. Diagrama de secuencia: Gestionar Almacén.

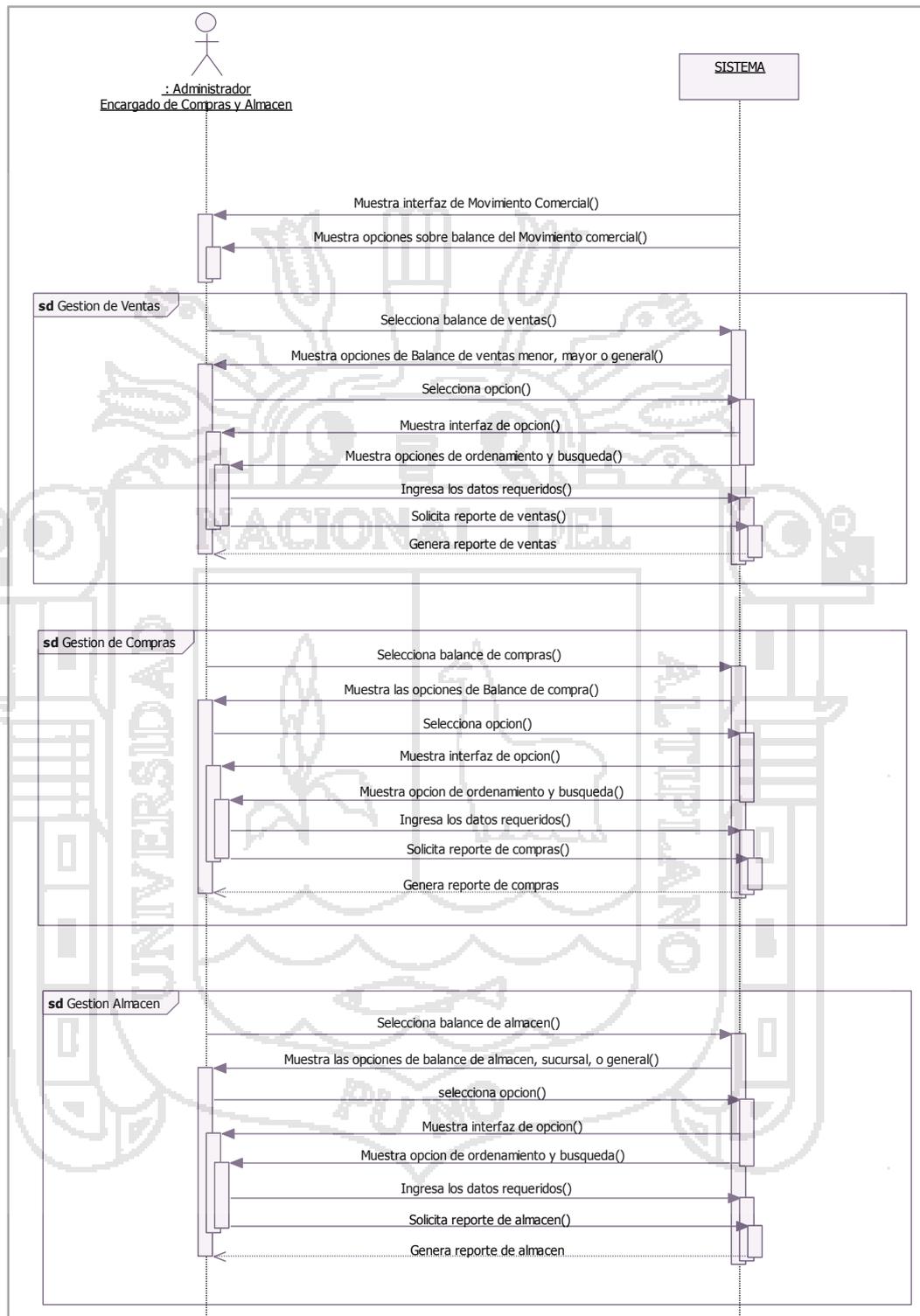
Figura 35. Diagrama de Secuencia Gestionar Almacén.



Fuente: Ejecutor.

J. Diagrama de secuencia: Gestionar Movimiento Comercial.

Figura 36. Diagrama de Secuencia Gestionar Movimiento Comercial.

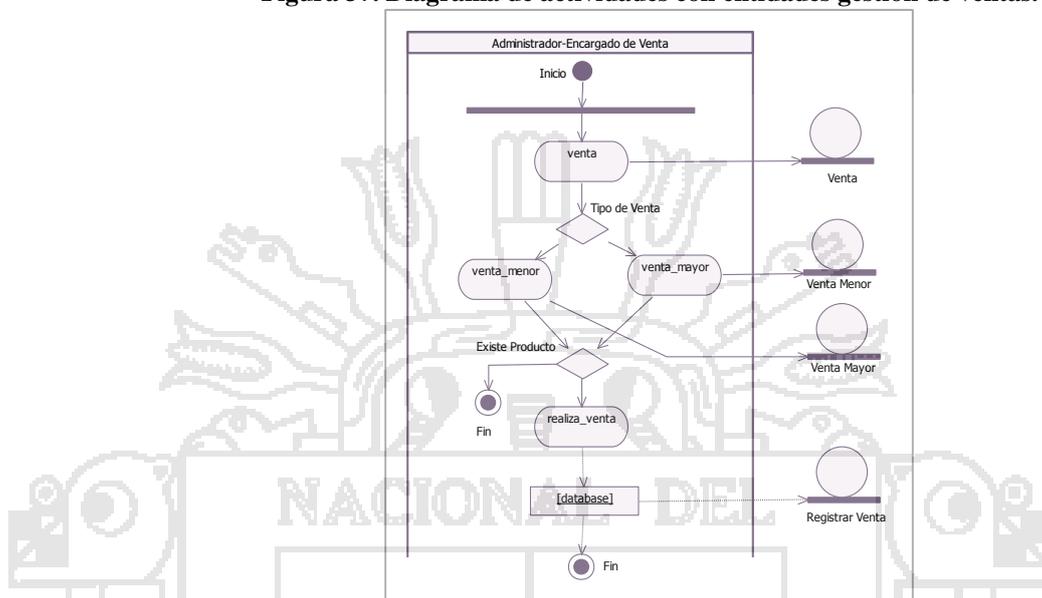


Fuente: Ejecutor.

4.1.2.4. Diagrama de Actividades.

A. Diagrama de Actividades: Gestión de Ventas.

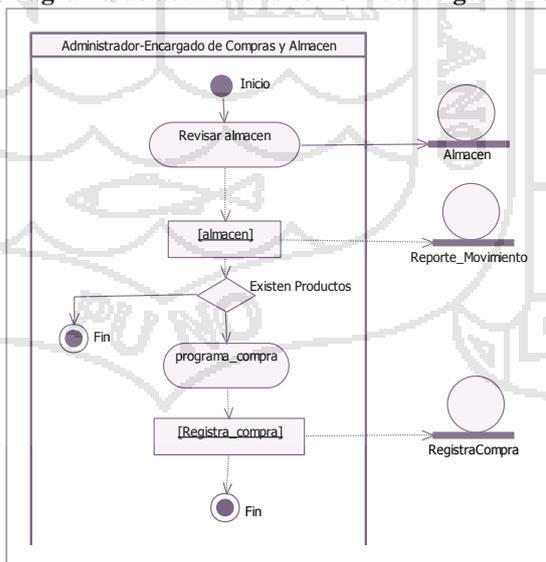
Figura 37. Diagrama de actividades con entidades gestión de ventas.



Fuente: Ejecutor.

B. Diagrama de Actividades: Gestión de Compras.

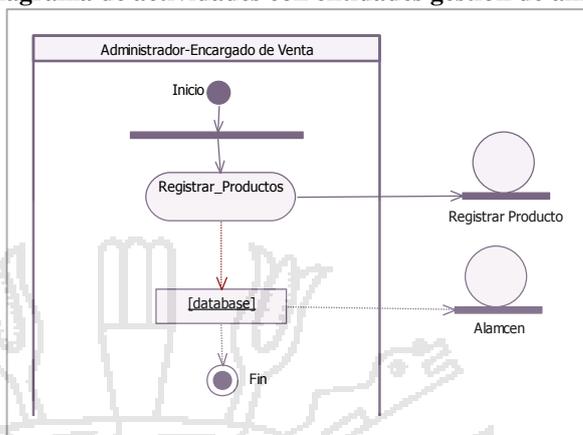
Figura 38. Diagrama de actividades con entidades gestión de compras.



Fuente: Ejecutor.

C. Diagrama de Actividades: Gestión de Almacén.

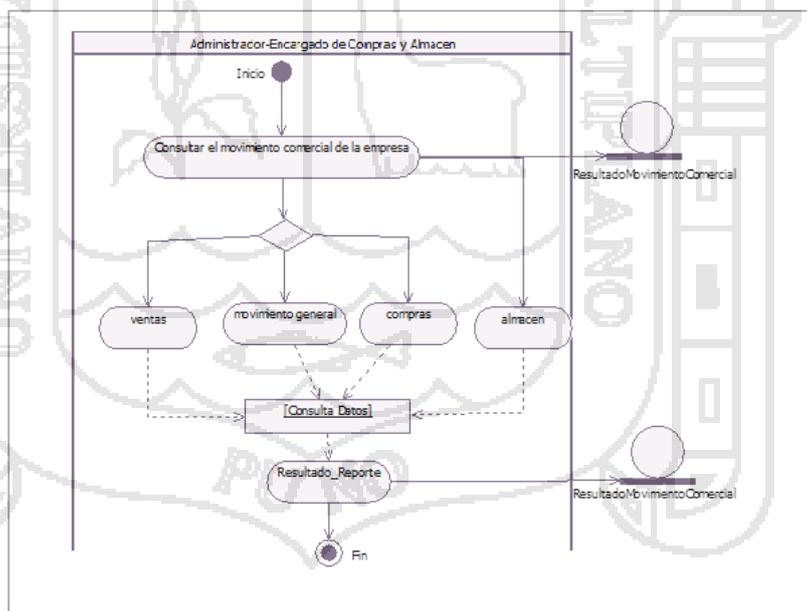
Figura 39. Diagrama de actividades con entidades gestión de almacén.



Fuente: Ejecutor.

D. Diagrama de Actividades: Movimiento Comercial.

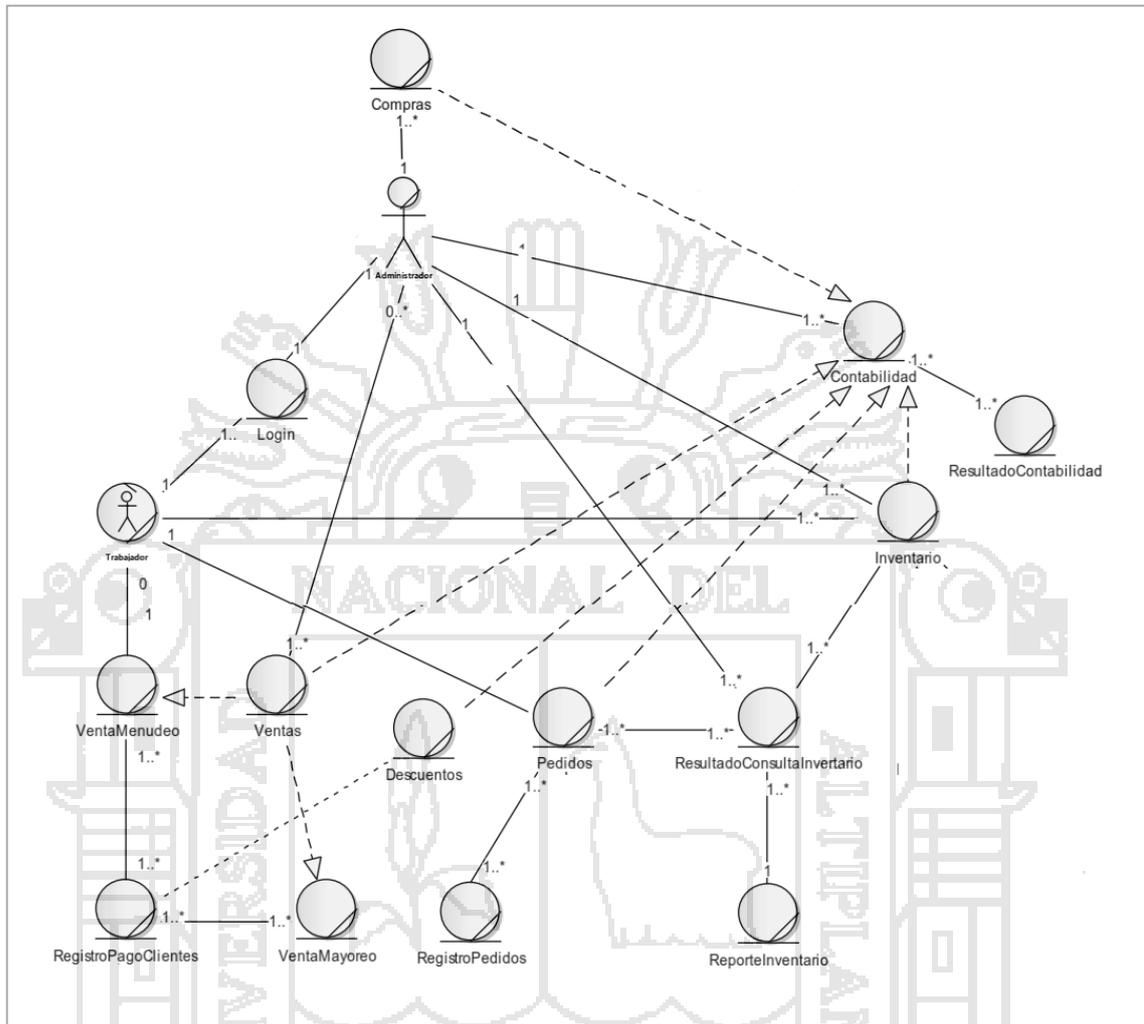
Figura 40. Diagrama de actividades con entidades movimiento comercial.



Fuente: Ejecutor.

4.1.2.5. Diagrama de Colaboración.

Figura 41. Modelado de objetos del negocio.



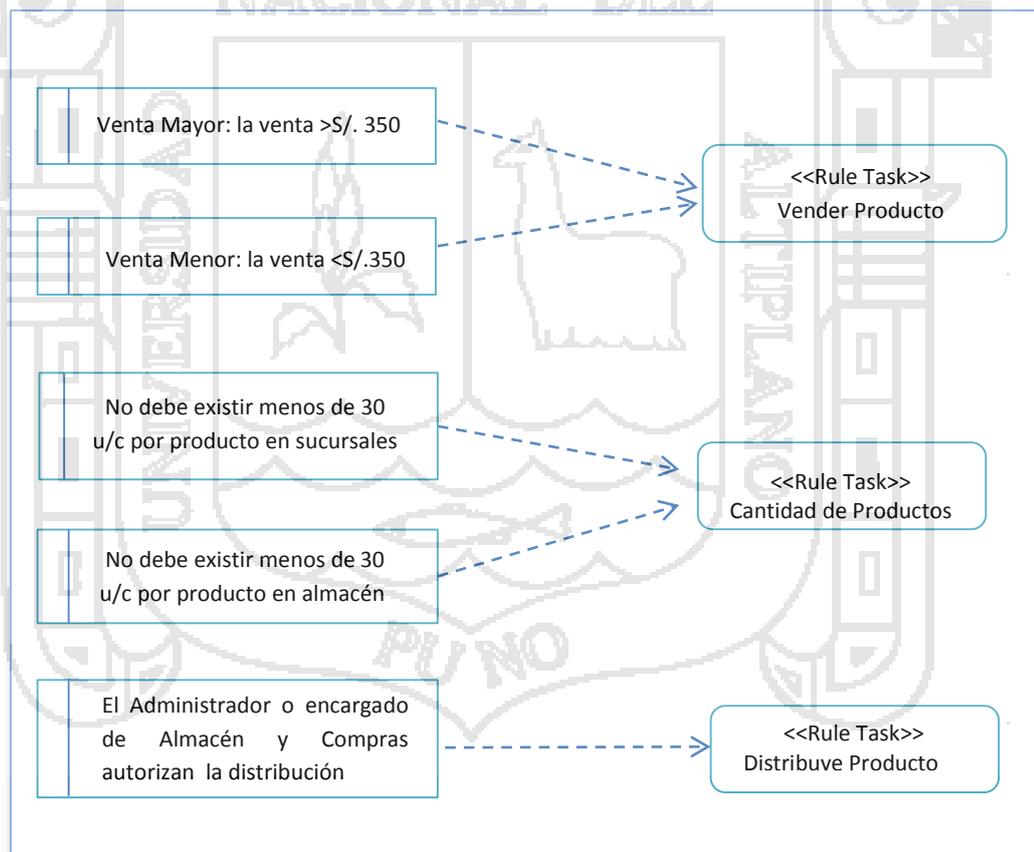
Fuente: Ejecutor.

4.1.2.6. Reglas de Negocio.

Las siguientes figuras muestran las reglas de negocio.

- Venta Mayor: la venta mayor a 350 soles por venta de productos.
- Venta Menor: la venta menor a 350 soles por venta de productos.
- Distribuir producto: El Administrador o Encargado de compras y almacén solo lo pueden efectuar la distribución.
- No deben existir menos de 30 unidades/cajas por producto en almacén.
- No debe existir menos de 30 unidades/cajas por producto en sucursal.

Figura 42. Reglas de Negocio.



Fuente: Ejecutor.

4.1.2.8. Modelado de la Arquitectura Orientada a Servicios SOA.

En esta parte de la investigación se muestran los diagramas del modelado de servicios SOMF que se utilizaron para el sistema.

- Modelo de análisis.
- Modelo integración de negocios.
- Modelo de diseño.

El modelado de servicios necesita el modelado de negocios que se representa en los puntos anteriores, porque los caso de uso de negocio son esenciales para poder identificar la atomicidad, composición y que casos de uso son clúster de servicios.

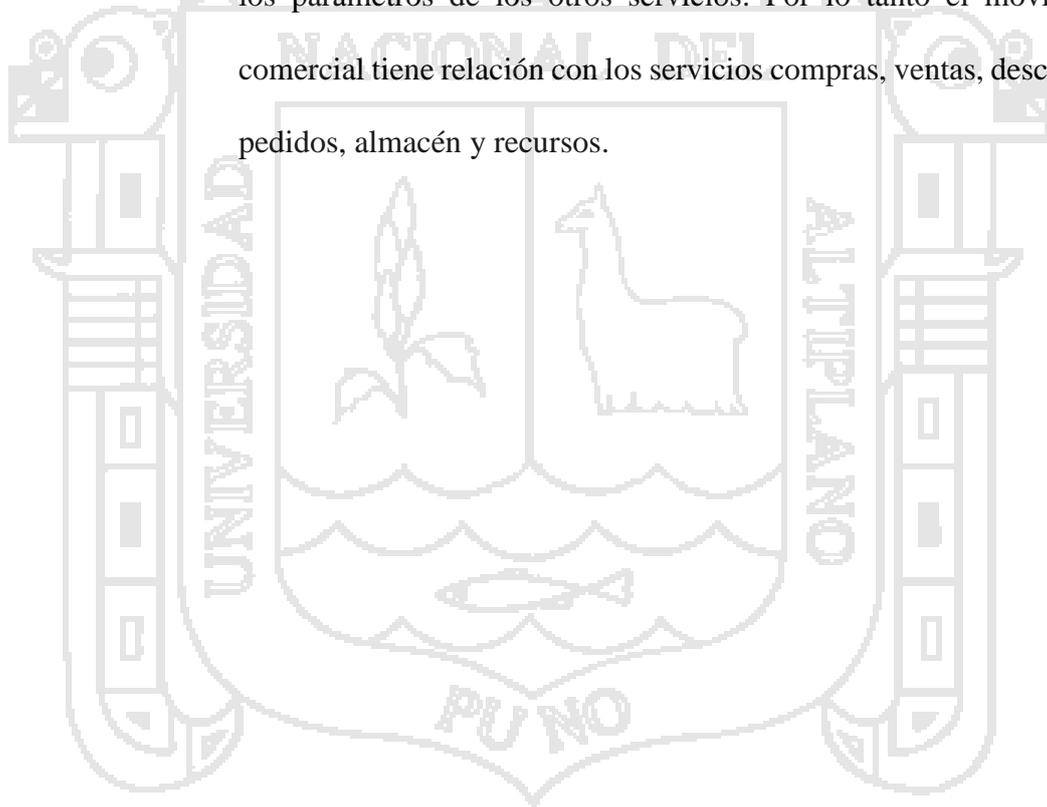
A. Modelo de Análisis.

Se identifican los servicios y el cómo están conformados, por medio de los requerimientos Funcionales. Se identifica el tipo de servicio es decir:

- Servicio atómico.
- Servicio compuesto.
- Clúster de servicios.

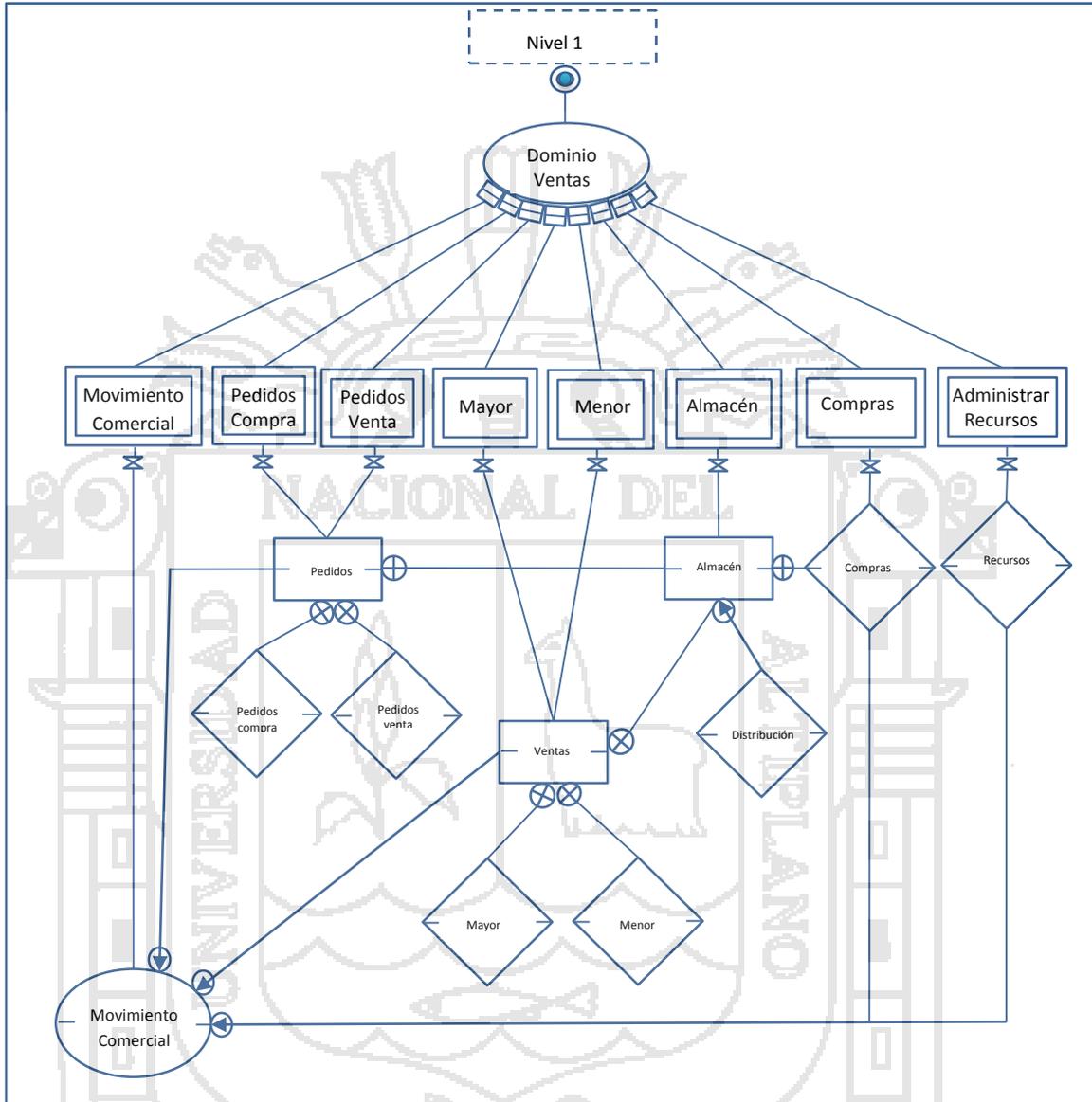
- Los servicios atómicos que se identificaron son: Compras, ventas al Mayor, ventas al Menor, Almacén y Distribución, Pedido de compra, Pedido de Venta.

Los Servicios Compuestos que se identificaron son: ventas, pedidos. Ya que ventas está conformado por ventas al mayor y ventas al menor y pedidos está conformado por pedidos de compra y pedidos de venta. El clúster de servicios es el que es accesible por todos los servicios, es decir, para obtener el movimiento comercial necesitamos los parámetros de los otros servicios. Por lo tanto el movimiento comercial tiene relación con los servicios compras, ventas, descuentos, pedidos, almacén y recursos.



integración. En la figura se observa el modelo SOA ubicada en la parte inferior y los componentes del negocio en la parte superior.

Figura 45. Modelo de Integración.

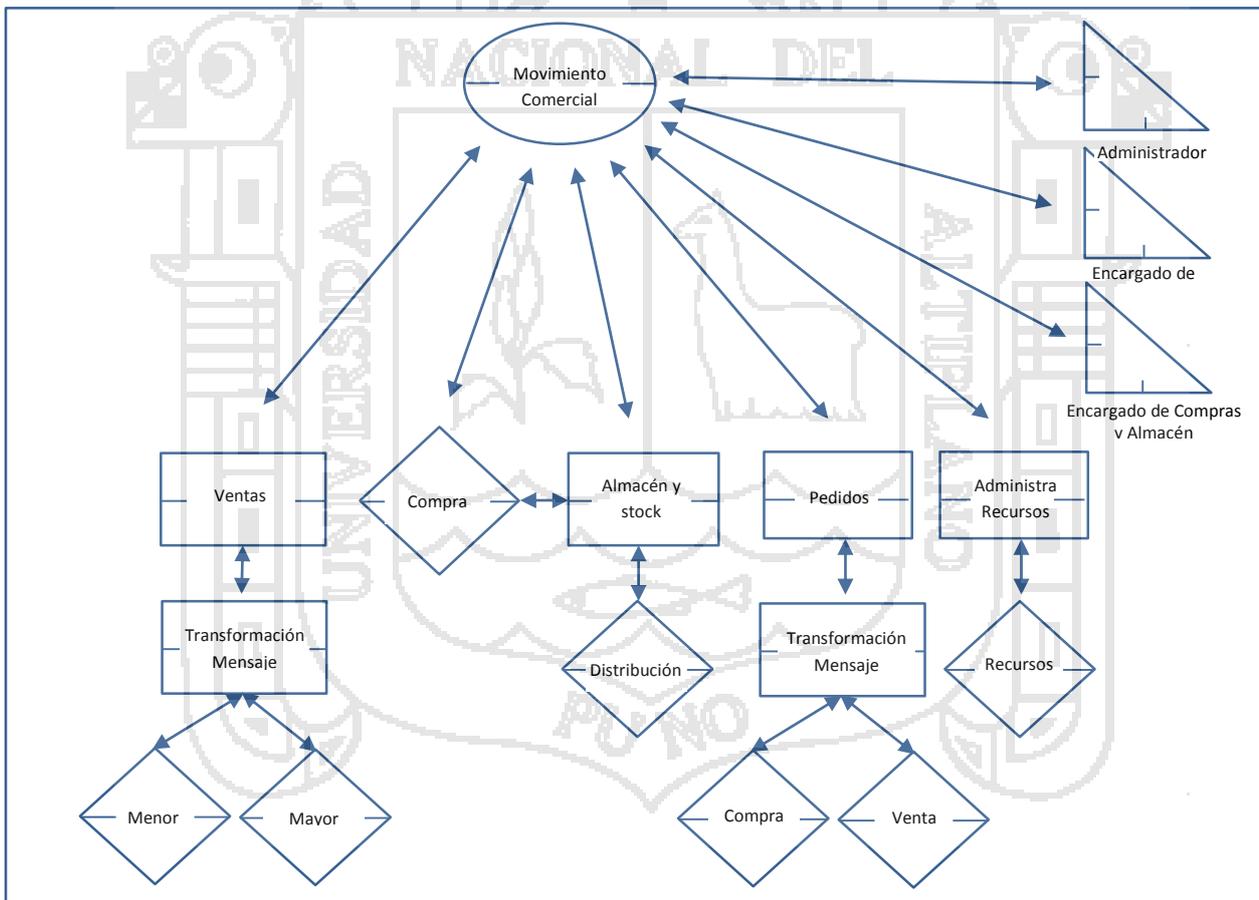


Fuente: Ejecutor.

C. Modelo de Diseño.

El modelo de diseño permite ver el enrutamiento de mensajes la visibilidad, la sincronización de mensajes, y la colaboración de los servicios, es decir la forma de intercambio de la información y transformación de mensaje, en la figura se puede observar que existe la relación bidireccional, método de entrega de mensajes que denota una conversación bidireccional entre un consumidor y un servicio, o entre dos servicios.

Figura 46. Modelo de Diseño.

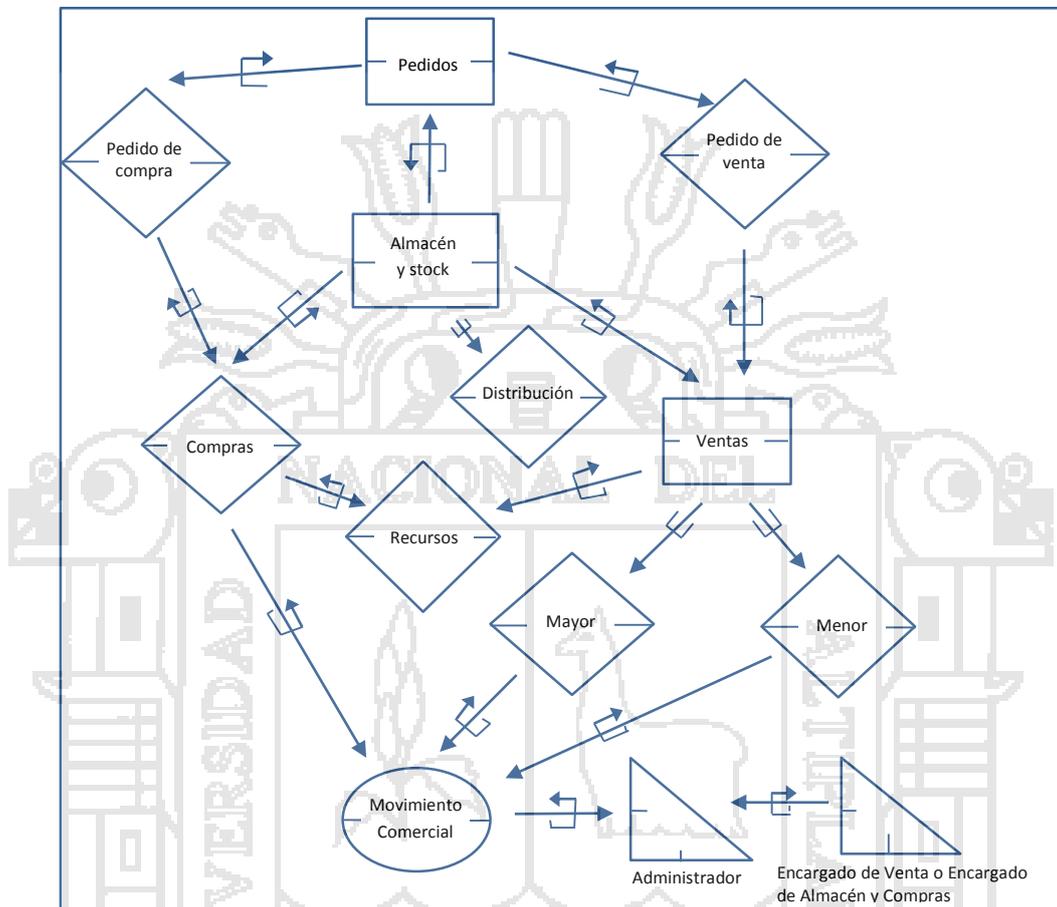


Fuente: Ejecutor.

La composición del diseño es el resultado de la asociación de servicios, las rutas de intercambio de mensajes, y los patrones generales de los servicios

que ofrecen una solución, cada uno de estos participantes se inspecciona para llevar a una mejor comprensión de la esencia del paquete orientado a servicios.

Figura 47. Modelo de diseño lógico de la composición.



Fuente: Ejecutor.

Gracias al modelado de servicios, se comprende el funcionamiento de cada servicio y el cómo están interactuando. Estos resultados facilitan la programación de la aplicación y la forma en que se pueden reutilizar ciertos servicios. Al tener los componentes del negocio representados en servicios, permite utilizar los servicios existentes por medio de una nueva colaboración en el momento que se necesite una nueva interacción.

La capacidad de integrar a clientes y socios es una gran ventaja, ya que SOA está estandarizado. Existen varias especificaciones de SOA, lo cual permite que se pueda publicar parte de los servicios del negocio para que clientes, proveedores y socios puedan acceder a ellos. En particular, Web Services que es adecuado para este propósito.

4.1.3. Diseño de la Arquitectura.

La arquitectura global de la solución viene marcada por el carácter distribuido del sistema, la adopción del modelo SOA y la multitud de tipos de cliente que se van a permitir.

Para lograr una mayor comprensión del modelo pretendido, se van a detallar distintas vistas del sistema:

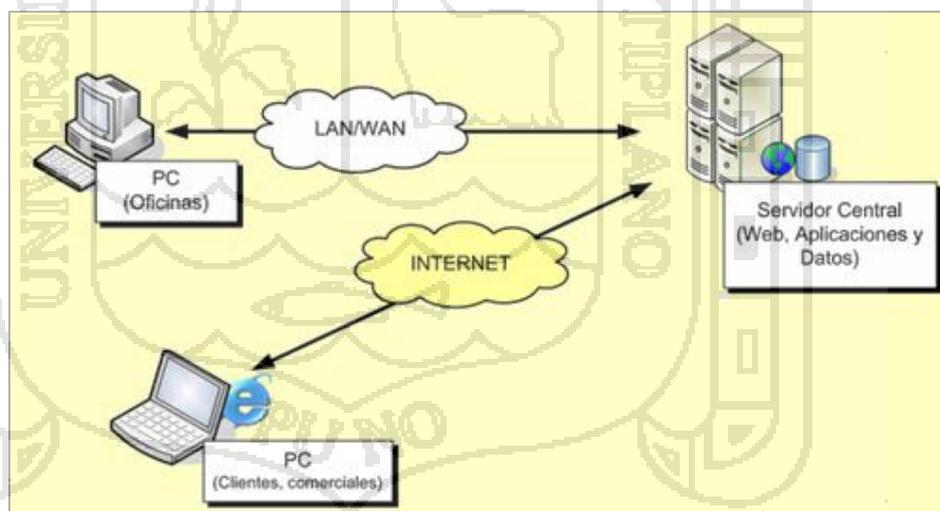
- La vista física, donde se reflejarán cómo están distribuidos físicamente los distintos nodos que intervienen en el sistema.
- La vista lógica, que indica los componentes conceptuales desarrollados para dar cobertura a las necesidades propuestas.

4.1.3.1. Vista Física.

El modelo que va a utilizarse en el diseño del sistema es altamente flexible y escalable. De esta forma, será posible realizar un despliegue físico reducido, óptimo cuando la carga de trabajo (número de usuarios, consultas) no es muy elevado, o bien podrá escalarse para dar servicio en escenarios con requisitos de carga más elevados.

La siguiente figura muestra una posible distribución de nodos en un escenario de uso reducido del sistema. Se puede observar que existe un único servidor central que presta servicios de publicación de Web, actúa como servidor de Aplicaciones, y al mismo tiempo, ejerce de almacén de información.

Figura 48. Vista física en escenario reducido.

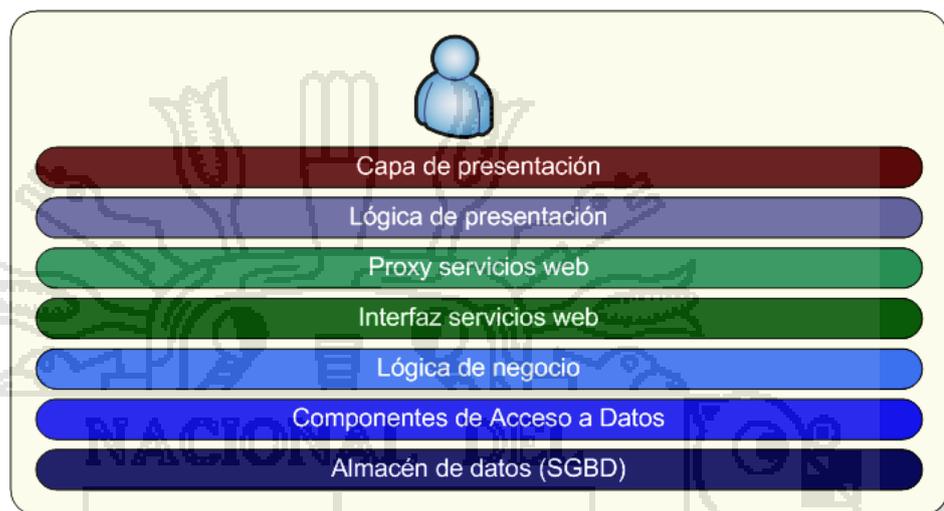


Fuente: Ejecutor.

4.1.3.2. Vista Lógica.

Las capas que estructuran la solución son recogidas en la siguiente figura, y se describen justo a continuación:

Figura 49. Arquitectura lógica.



Fuente: Ejecutor.

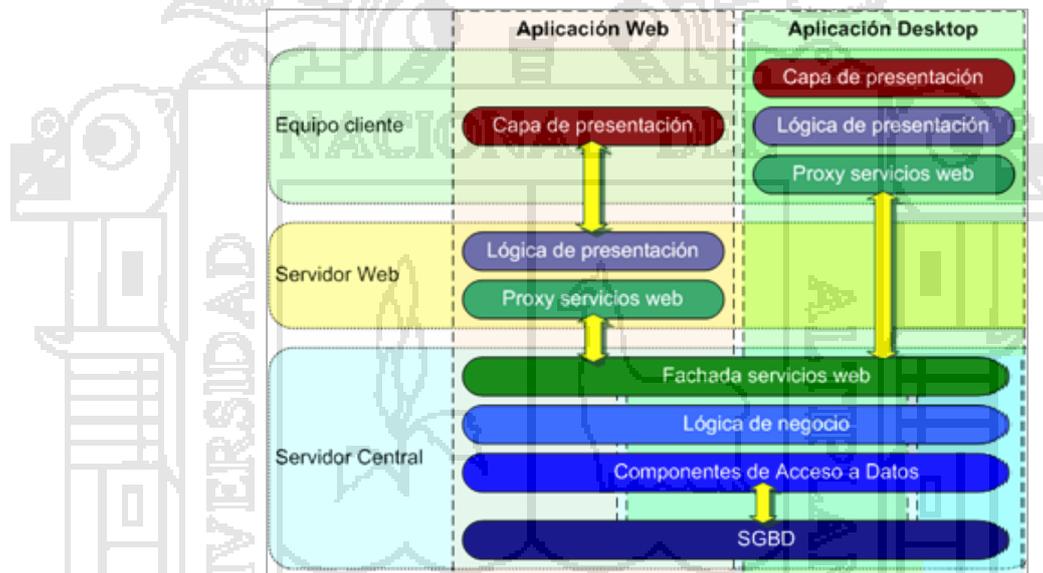
- Almacén de datos, en el que recae la responsabilidad de almacenar y ofrecer los mecanismos de recuperación de información utilizada por el sistema.
- Componentes de Acceso a Datos, que facilitan el acceso al almacén ocultando las particularidades de éste al resto de capas de la solución, lo que permitiría en última instancia la sustitución del SGBD de forma transparente.
- Lógica de Negocio, que utiliza los servicios prestados por los Componentes de Acceso a Datos para implementar las funcionalidades requeridas por la aplicación. Las operaciones incluidas cumplen las características propias a la adopción del paradigma SOA:

- Fachada servicios Web, cuyos componentes actúan de interfaz hacia el exterior de la Lógica de Negocio utilizando Servicios Web XML. En otras palabras, se encargan de recibir los mensajes enviados por los clientes a los servicios, procesarla utilizando los componentes de lógica de negocio, y de devolver la respuesta.
- Proxy servicios web, son los conectores que permiten consumir los servicios prestados por la aplicación desde las aplicaciones cliente de forma transparente. Han sido creados utilizando las herramientas automáticas de Visual Studio y retocados para adaptarlos a las necesidades concretas del sistema.
- Lógica de presentación, cuyos elementos toman las acciones del usuario y las trasladarán al formato específico del dispositivo (y medio) de acceso utilizado.
- Capa de Presentación, la más cercana al usuario, cuya misión será únicamente encargarse de la relación con éste, comunicando sus acciones a los elementos de Lógica de Presentación, y mostrando resultados adaptados al dispositivo de visualización.

La distribución de componentes, sin embargo, no es homogénea, pues según el tipo de cliente podrán ser ubicados en unos nodos u otros. Es importante destacar también las posibilidades de comunicación entre ellos, pues determinan la extensibilidad del modelo que fue descrita desde el punto de vista físico.

El siguiente diagrama recoge la misma información que el anterior, pero detallando la ubicación de los elementos en los distintos nodos según el tipo de aplicación cliente, así como señalando las comunicaciones existentes entre ellos. Las flechas bidireccionales que unen dos capas indican que ambas intercambian mensajes (o pueden hacerlo) utilizando infraestructura de comunicaciones, mientras que las que capas adyacentes se comunican de forma directa, a través de instancias dentro del mismo dominio de aplicación.

Figura 50. Arquitectura lógica detallada.



Fuente: Ejecutor.

4.1.4. Implementación del Sistema.

En base a la arquitectura de software SOA se desarrolló los módulos del sistema comercial utilizando SQL como gestor de base de datos, C# como lenguaje de programación en el nivel de servicios.

Como se ha descrito anteriormente en este documento, este proyecto implementa la lógica de negocio del sistema comercial por lo que es el núcleo de los servicios ofrecidos al resto de aplicaciones. Asimismo, el ensamblado resultante es absolutamente independiente del modelo de comunicaciones elegido para la transferencia de información entre los clientes y los servicios. Esta particularidad hace posible acceder a estas funcionalidades desde un proyecto de servicios web como el que se incluye en la solución.

La aplicación cliente fue desarrollada en C#, WPF (Windows Presentation Foundation). Los servicios web fueron desarrollados en C# Visual Studio 2010, usando WCF (Windows Communication Foundation). El protocolo de comunicación es por medio de SOAP.

A continuación se muestran interfaces de la aplicación cliente de los servicios y el consumo del mismo. En el anexo C se muestra con más detalle el código y servicios del sistema.

Figura 51. Ventana de Autenticación.



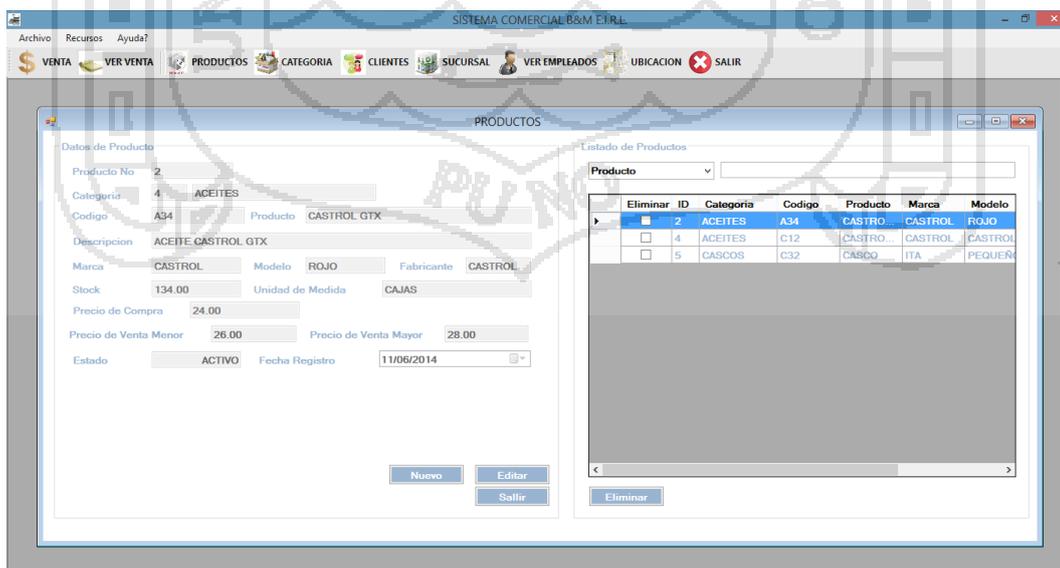
Fuente: Ejecutor.

Figura 52. Ventana principal del sistema.



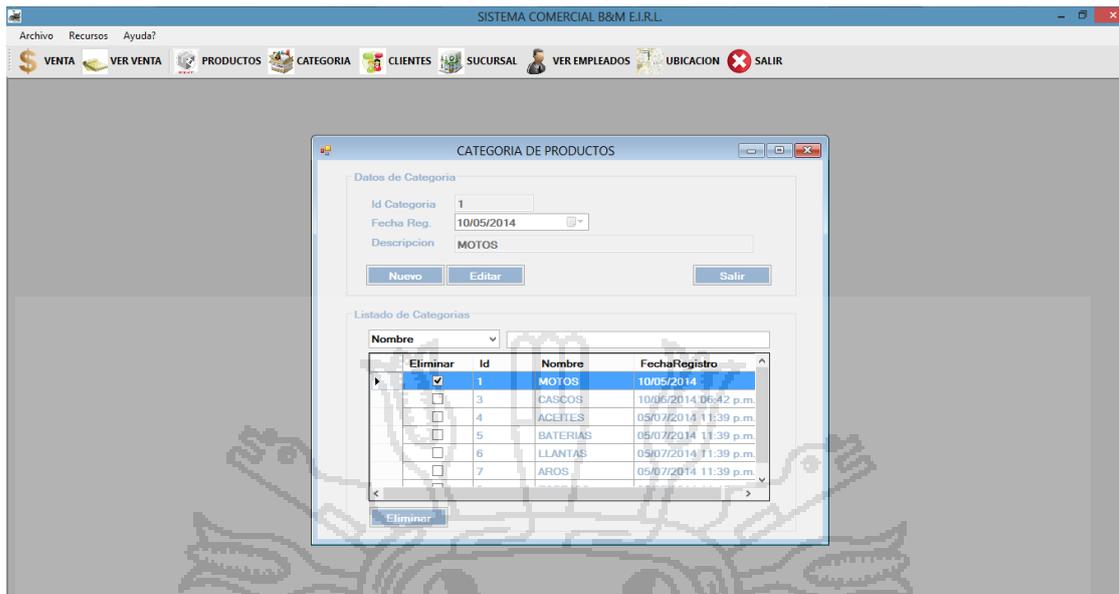
Fuente: Ejecutor.

Figura 53. Ventana de productos.



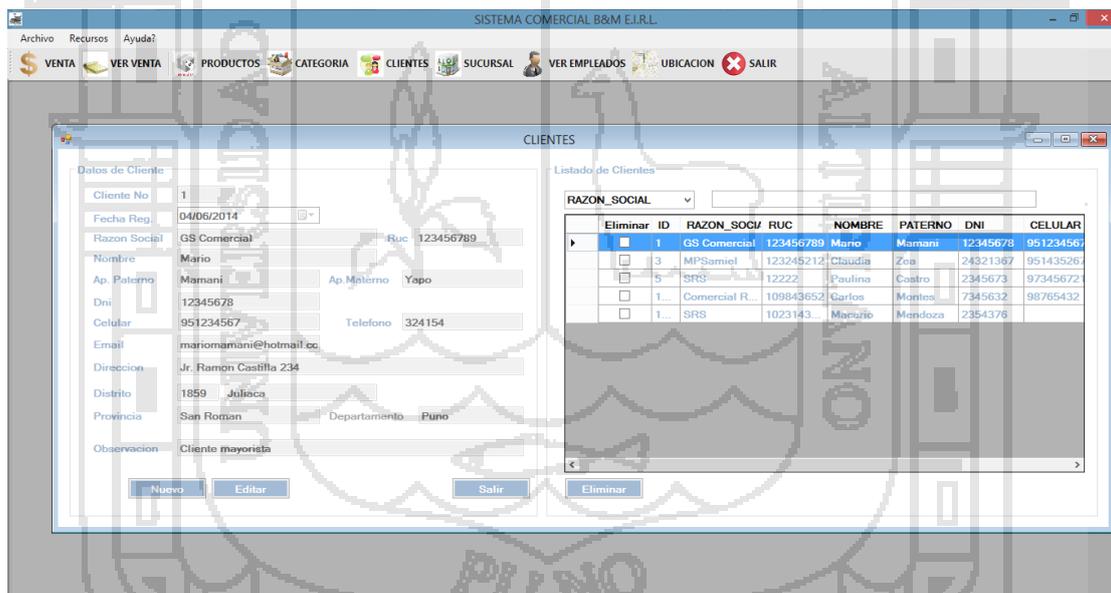
Fuente: Ejecutor.

Figura 54. Ventana de categoría de productos.



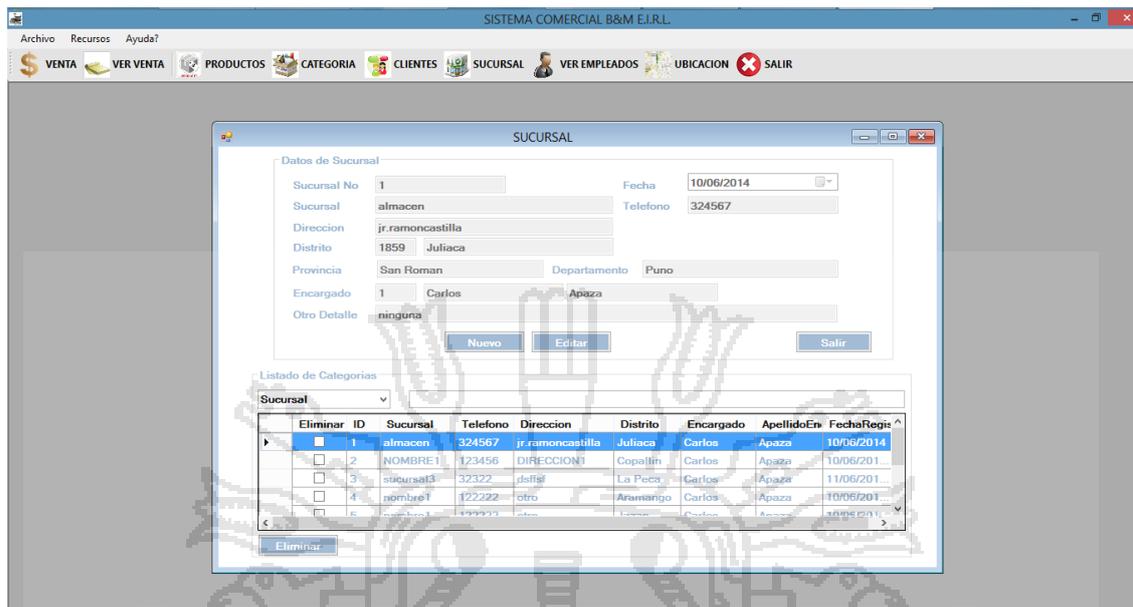
Fuente: Ejecutor.

Figura 55. Ventana de clientes.



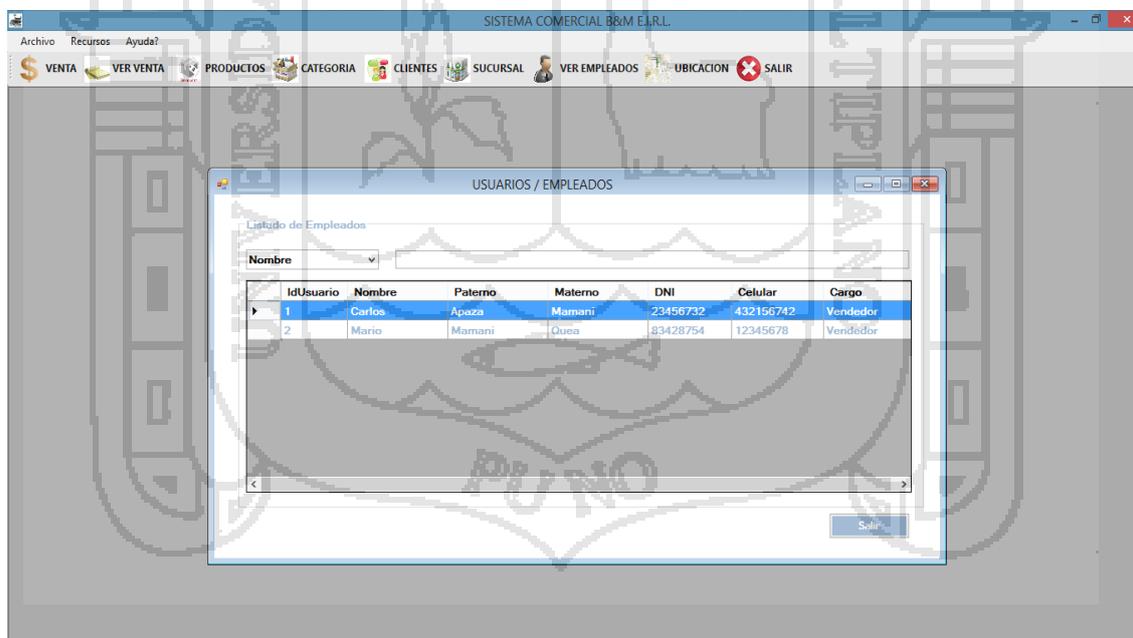
Fuente: Ejecutor.

Figura 56. Ventana de sucursales.



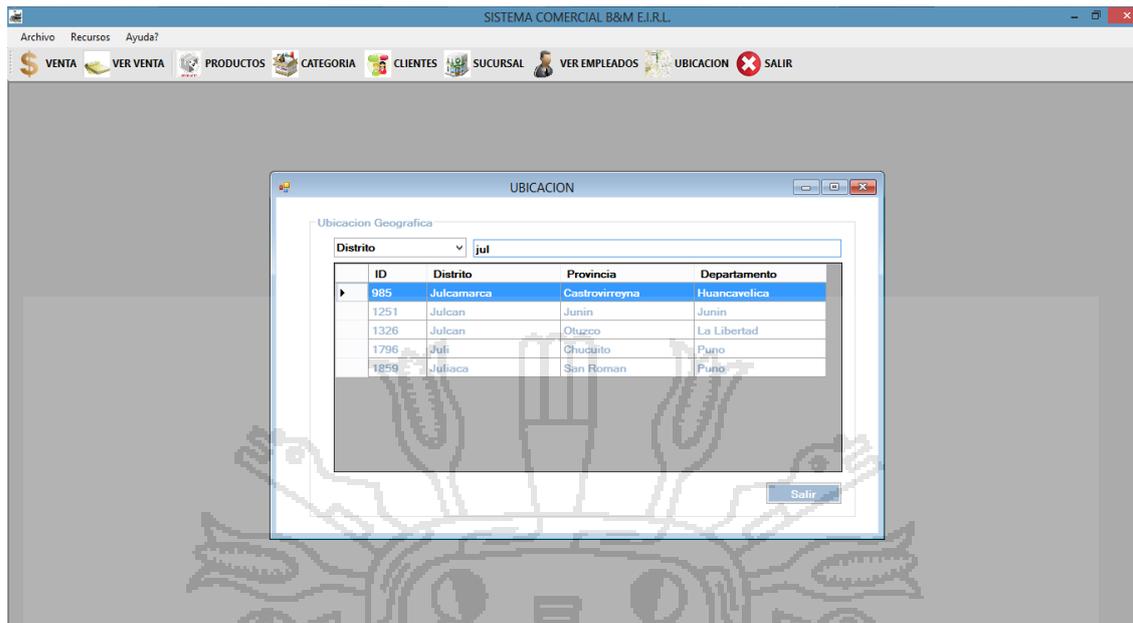
Fuente: Ejecutor.

Figura 57. Ventana de usuarios y empleados.



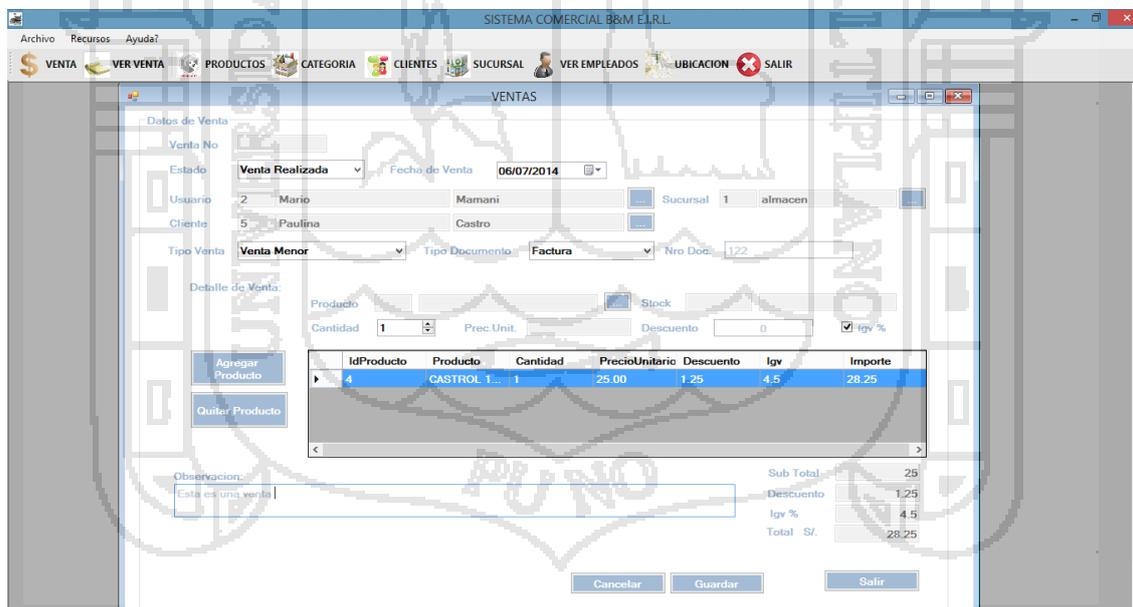
Fuente: Ejecutor.

Figura 58. Ventana de listado de ubicación.



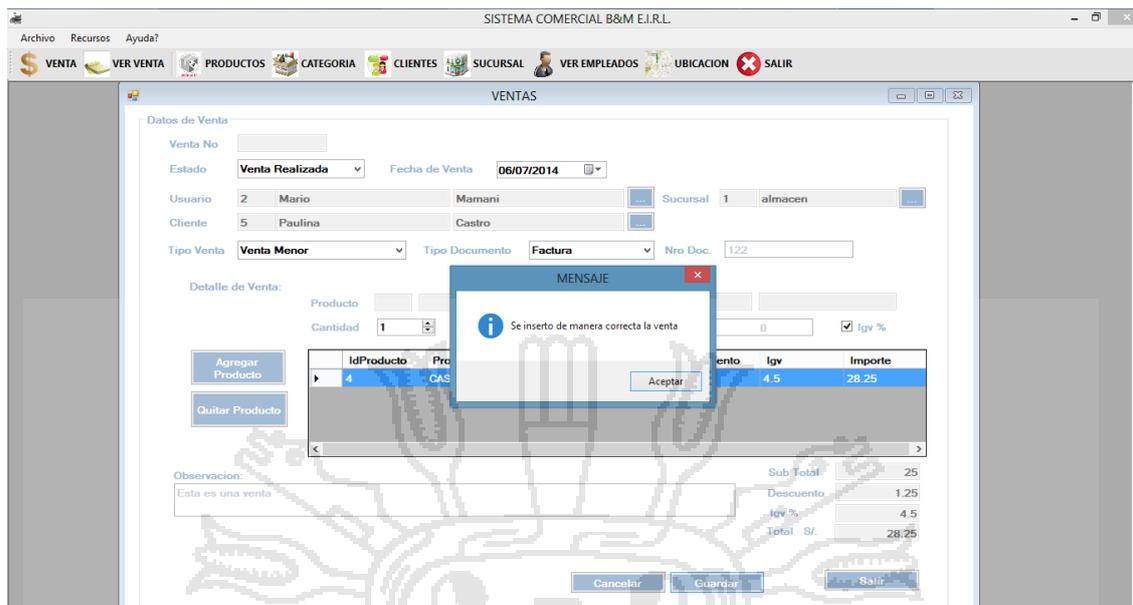
Fuente: Ejecutor.

Figura 59. Ventana de registro de nueva venta.



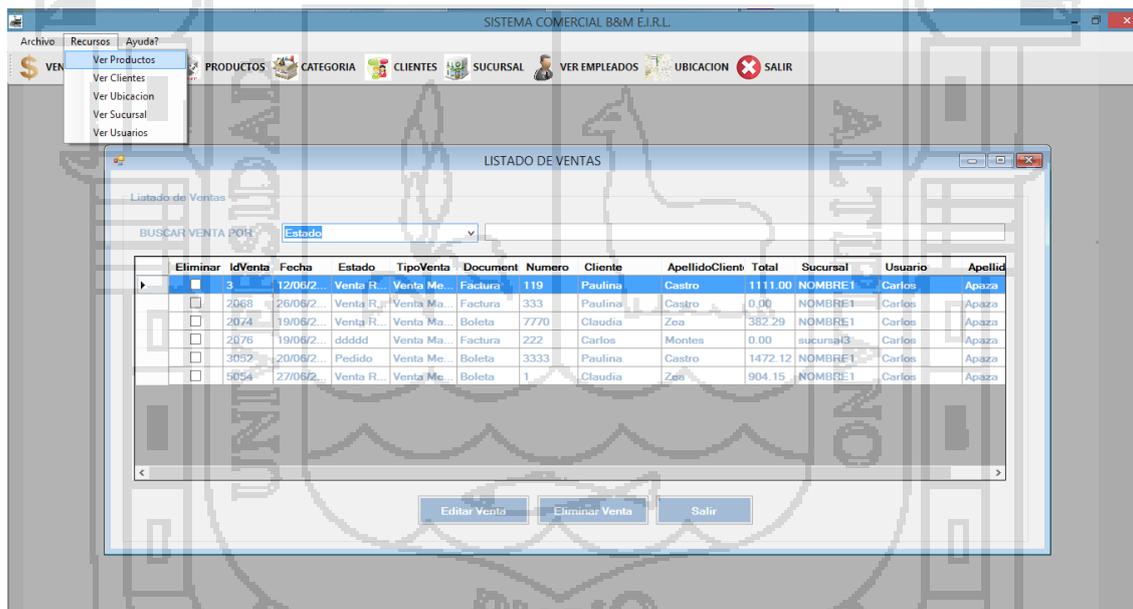
Fuente: Ejecutor.

Figura 60. Mensaje de confirmación de registro de venta.



Fuente: Ejecutor.

Figura 61. Ventana de listado de venta.



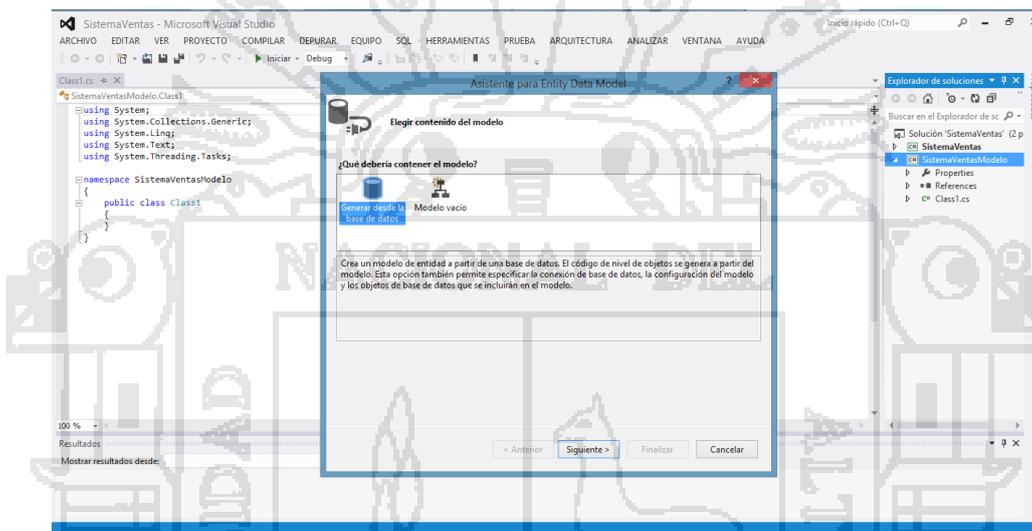
Fuente: Ejecutor.

Para la parte de la implementación se debe de considerar realizar primero la conexión a la base de datos, crear los Servicios y crear un cliente que consuma los servicios en el sistema.

A continuación se describe la secuencia de la implementación del servicio de listar productos.

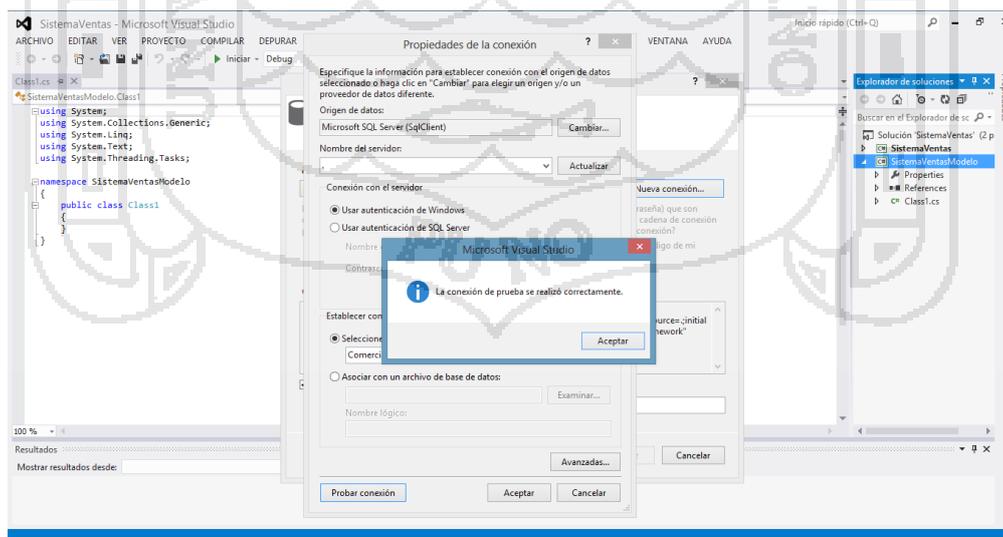
El proceso que se debe seguir es agregar un nuevo proyecto biblioteca de clases “Proyecto Modelo”, en el que se crea un nuevo modelo de la base de datos, en este caso “Comercial” Base de datos del sistema.

Figura 62. Conexión a la base de datos.



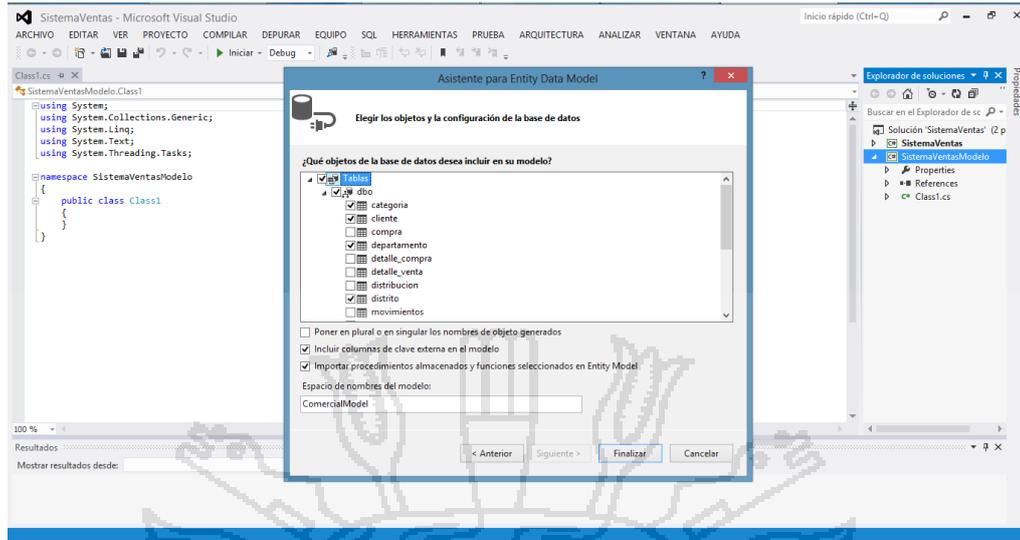
Fuente: Ejecutor.

Figura 63. Prueba de conexión a la base de datos.



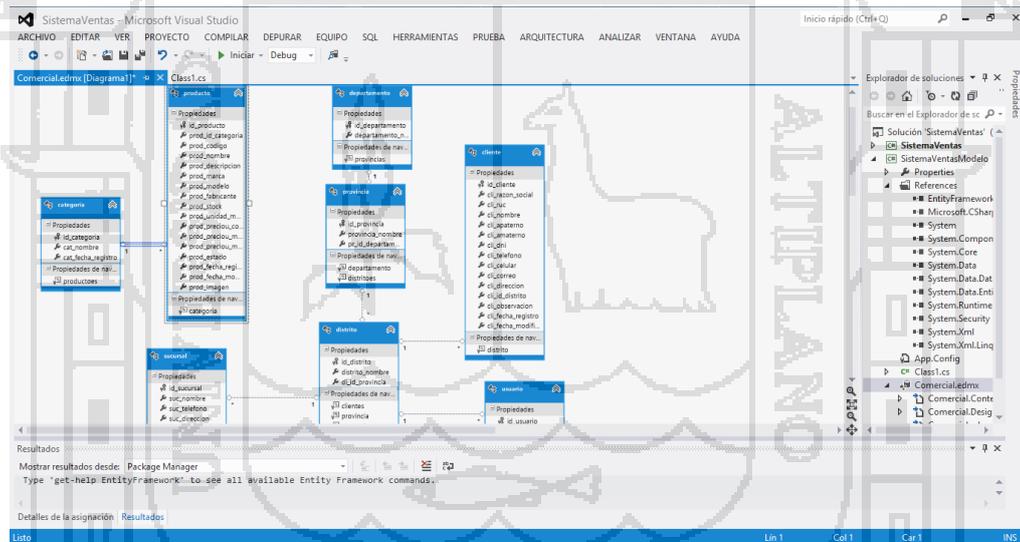
Fuente: Ejecutor.

Figura 64. Adición de las tablas de la base de datos al sistema.



Fuente: Ejecutor.

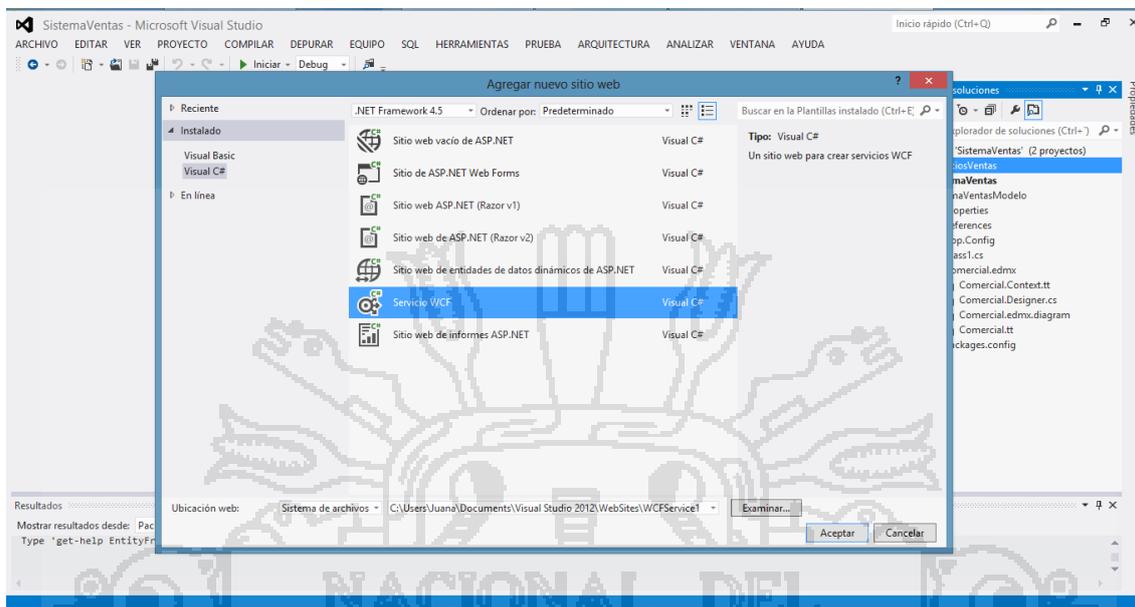
Figura 65. Mapeado de tablas requeridas.



Fuente: Ejecutor.

Para la creación de los servicios se utiliza el WCF de Visual Studio.

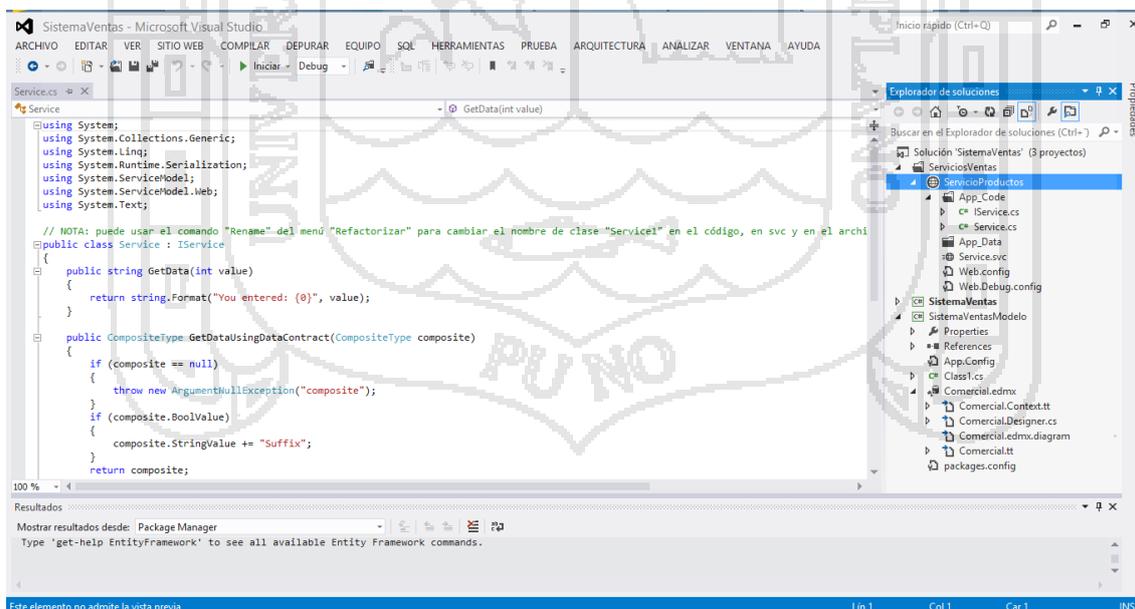
Figura 66. Inserción de la clase WCF.



Fuente: Ejecutor.

Una vez creada la clase se editan de acuerdo a lo requerido por el sistema.

Figura 67. Creación de los servicios mediante WCF.



Fuente: Ejecutor.

Figura 68. Clase de la interfaz del servicio listar producto.

```

IProductoService.cs
using System.Text;

//Este es contrato del servicio.
[ServiceContract]
public interface IProductoService
{
    [OperationContract]
    List<ProductoData> GetAllProducto();
}

//Entiti Fram. no te creas los objetos de la base de datos por loq ue se debe de crear la clase
// que convierte en serializables que puedan ser leidos por la web
[DataContract]
public class ProductoData
{
    [DataMember] public string id;
    [DataMember] public string categoria;
    [DataMember] public string codigo;
    [DataMember] public string nombres;
    [DataMember] public string descripcion;
    [DataMember] public string marca;
    [DataMember] public string modelo;
    [DataMember] public string fabricante;
    [DataMember] public string stock;
    [DataMember] public string unidad_medida;
    [DataMember] public string precio_menor;
    [DataMember] public string precio_mayor;
    [DataMember] public string estado;
}
    
```

Fuente: Ejecutor.

Figura 69. Clase del servicio listar producto.

```

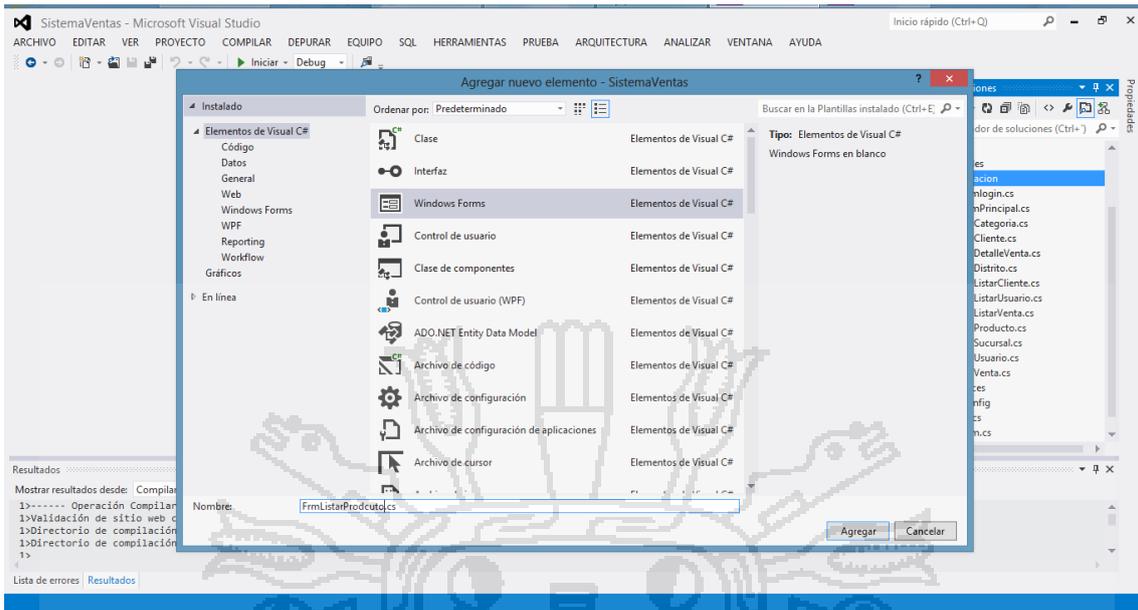
ProductoService.cs
using System.Text;

using SistemaVentasModelo;
//Aqui se encuentra las operaciones del servicio.
public class ProductoService : IProductoService
{
    public List<ProductoData> GetAllProducto()
    {
        ComercialEntities contexto;
        using (contexto = new ComercialEntities())
        {
            List<ProductoData> Misproducto = new List<ProductoData>();
            foreach (var p in contexto.productoes)
            {
                ProductoData pdata = new ProductoData();
                pdata.id = p.id_producto;
                pdata.codigo = p.prod_codigo;
                pdata.nombre = p.prod_nombre;
                pdata.descripcion = p.prod_descripcion;
                pdata.marca = p.prod_marca;
                pdata.modelo = p.prod_modelo;
                pdata.stock = p.prod_stock;
                pdata.unidad_medida = p.prod_unidad_medida;
                pdata.precio_compra = p.prod_precio_compra;
                pdata.precio_menor = p.prod_precio_menor;
                pdata.precio_mayor = p.prod_precio_mayor;
                pdata.estado = p.prod_estado;
                Misproducto.Add(pdata);
            }
            return Misproducto;
        }
    }
}
    
```

Fuente: Ejecutor.

Una vez creado el servicio se crea la aplicación cliente para el consumo del servicio.

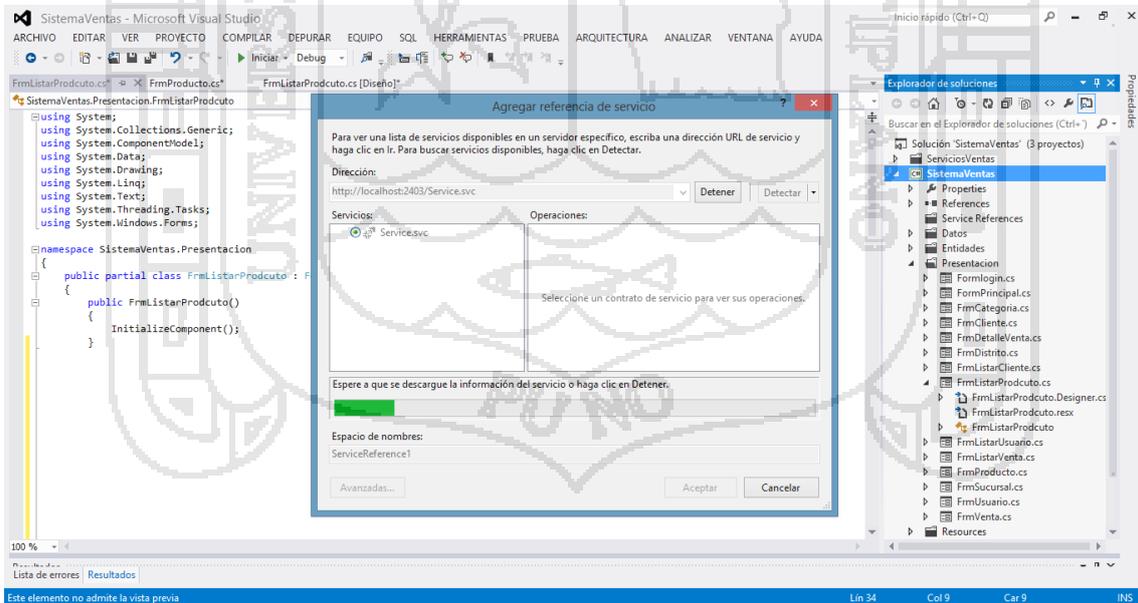
Figura 70. Creación del cliente del servicio.



Fuente: Ejecutor.

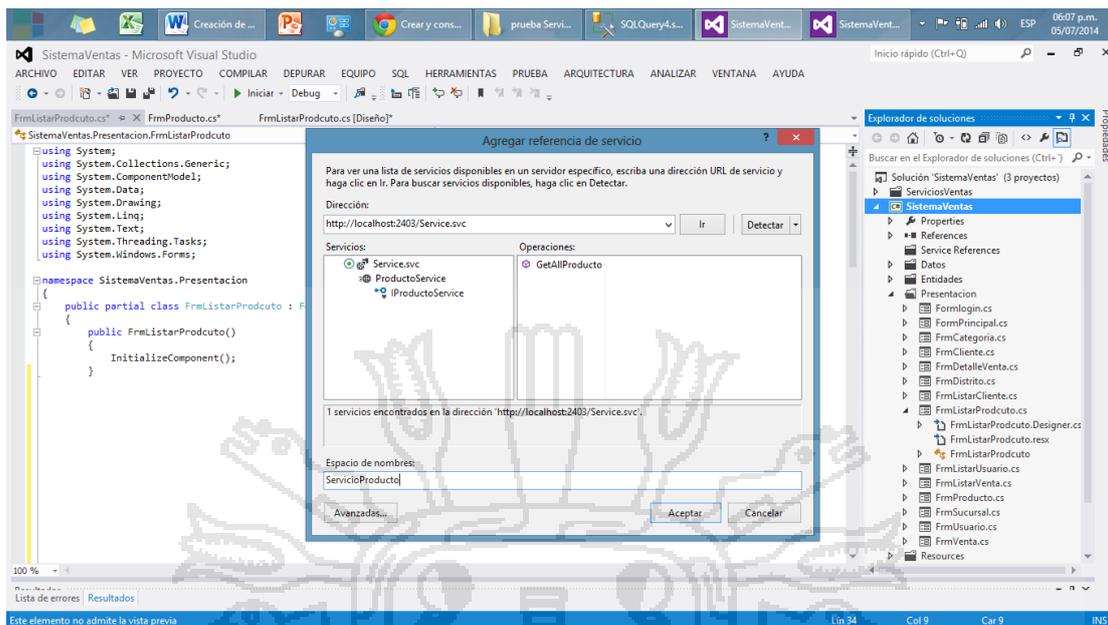
Luego de crear el cliente del servicio, se agrega la referencia al servicio como se muestra en la siguiente figura.

Figura 71. Referencia al servicio creado.



Fuente: Ejecutor.

Figura 72. Selección de los servicios detectados.



Fuente: Ejecutor.

En esta etapa se puede probar si el servicio creado está disponible, para probar la dirección del servicio se coloca la dirección del servicio en un explorador obteniendo el siguiente mensaje.

Figura 73. Prueba de invocación del servicio.



Fuente: Ejecutor.

Una vez invocado el servicio se obtiene el siguiente resultado, este código muestra lo que se debería de mostrar una vez invocado el servicio. Se puede observar los WSDL contrato del servicio y lo que es solicitado. En este caso el listado de los productos.

Figura 74. Contenido del servicio invocado - contrato del servicio

```

localhost2777/Service.svc x
localhost2777/Service.svc?singleWsdL
Investigación Allenc... Crear y consumir un... creacion de tablas u... Survey4Free, ofremo... www.web.onpe.gob... Vacantes CONVOCATORIA C... Convocatorias de pe...
<?xml:stylesheet type="text/xsl" href="http://schemas.xmlsoap.org/wsdl/" xmlns:wsap="http://schemas.xmlsoap.org/ws/2004/08/addressing/policy"
xmlns:wsa10="http://www.w3.org/2005/05/addressing" xmlns:tms="http://tempuri.org/" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing/policy"
xmlns:wsap2="http://schemas.xmlsoap.org/soap/encoding/" xmlns:wsa2="http://schemas.xmlsoap.org/ws/2004/09/mex" xmlns:wsap3="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:wsa3="http://www.w3.org/2007/05/addressing/metadata" xmlns:wsa4="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsap4="http://schemas.xmlsoap.org/ws/2004/09/policy" xmlns:wsa5="http://www.w3.org/2006/05/addressing/wsdl" xmlns:wsap5="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:wsa6="http://docs.oasis-open.org/ws-a/2004/01/oasis-200401-wsa-wssecurity-utility-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
name="IListarProductoService" targetNamespace="http://tempuri.org/" />
<wsdl:types>
<xs:schema xmlns:x="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" targetNamespace="http://tempuri.org/">
<xs:import namespace="http://schemas.datacontract.org/2004/07/" />
<xs:element name="GetAllProducto">
<xs:complexType>
<xs:sequence>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="GetAllProductoResponse">
<xs:complexType>
<xs:sequence>
<xs:element xmlns:q1="http://schemas.datacontract.org/2004/07/" minOccurs="0" name="GetAllProductoResult" nillable="true" type="q1:ArrayOfProductoData"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
<xs:schema xmlns:x="http://www.w3.org/2001/XMLSchema" xmlns:t="http://schemas.xmlsoap.org/2003/10/Serialization/" attributeFormDefault="qualified"
elementFormDefault="qualified" targetNamespace="http://schemas.xmlsoap.org/2003/10/Serialization/">
<xs:element name="anyType" nillable="true" type="xs:anyType"/>
<xs:element name="anyURI" nillable="true" type="xs:anyURI"/>
<xs:element name="base64Binary" nillable="true" type="xs:base64Binary"/>
<xs:element name="boolean" nillable="true" type="xs:boolean"/>
<xs:element name="byte" nillable="true" type="xs:byte"/>
<xs:element name="dateTime" nillable="true" type="xs:dateTime"/>
<xs:element name="decimal" nillable="true" type="xs:decimal"/>
<xs:element name="double" nillable="true" type="xs:double"/>
<xs:element name="float" nillable="true" type="xs:float"/>
<xs:element name="int" nillable="true" type="xs:int"/>
<xs:element name="long" nillable="true" type="xs:long"/>
<xs:element name="QName" nillable="true" type="xs:QName"/>
<xs:element name="short" nillable="true" type="xs:short"/>
<xs:element name="string" nillable="true" type="xs:string"/>
<xs:element name="unsignedByte" nillable="true" type="xs:unsignedByte"/>
<xs:element name="unsignedInt" nillable="true" type="xs:unsignedInt"/>
<xs:element name="unsignedLong" nillable="true" type="xs:unsignedLong"/>
<xs:element name="unsignedShort" nillable="true" type="xs:unsignedShort"/>
<xs:element name="char" nillable="true" type="xs:char"/>
<xs:simpleType name="char">
<xs:restriction base="xs:int"/>
</xs:simpleType>
<xs:element name="duration" nillable="true" type="tns:duration"/>
<xs:simpleType name="duration">
<xs:restriction base="xs:duration">
<xs:pattern value="-?P(\d+D)?(T(\d+H)?(\d+M)?(\d+S)?)?"/>
<xs:minInclusive value="-P10675199DT2H48M5.4775803S"/>
<xs:maxInclusive value="P10675199DT2H48M5.4775803S"/>
</xs:restriction>
</xs:simpleType>
<xs:element name="guid" nillable="true" type="tns:guid"/>
<xs:simpleType name="guid">
<xs:restriction base="xs:string">
<xs:pattern value="(\{da-fa-f\}\{8\}-[\da-fa-f\]{4}-[\da-fa-f\]{4}-[\da-fa-f\]{4}-[\da-fa-f\]{12})"/>
</xs:restriction>
</xs:simpleType>
<xs:attribute name="FactoryType" type="xs:QName"/>
<xs:attribute name="Id" type="xs:ID"/>
<xs:attribute name="Ref" type="xs:IDREF"/>
</xs:schema>
<xs:schema xmlns:x="http://www.w3.org/2001/XMLSchema" xmlns:t="http://schemas.datacontract.org/2004/07/" elementFormDefault="qualified"
targetNamespace="http://schemas.datacontract.org/2004/07/">
<xs:complexType name="ArrayOfProductoData">
<xs:sequence>
<xs:element minOccurs="0" maxOccurs="unbounded" name="ProductoData" nillable="true" type="tns:ProductoData"/>
</xs:sequence>
</xs:complexType>
<xs:element name="ArrayOfProductoData" nillable="true" type="tns:ArrayOfProductoData"/>
<xs:complexType name="ProductoData">
<xs:sequence>
<xs:element minOccurs="0" name="Codigo" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="Descripcion" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="Estado" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="Fabricante" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="Id" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="Marca" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="Modelo" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="Precio Compra" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="Precio Mayor" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="Precio Menor" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="Producto" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="Stock" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="Unidad Medida" nillable="true" type="xs:string"/>
</xs:sequence>
</xs:complexType>
<xs:element name="ProductoData" nillable="true" type="tns:ProductoData"/>
</xs:schema>
</wsdl:types>
<wsdl:message name="IListarProductoService_GetAllProducto_InputMessage">
<wsdl:part name="parameters" element="tns:GetAllProducto"/>
</wsdl:message>
<wsdl:message name="IListarProductoService_GetAllProducto_OutputMessage">
<wsdl:part name="parameters" element="tns:GetAllProductoResponse"/>
</wsdl:message>
<wsdl:portType name="IListarProductoService">
<wsdl:operation name="GetAllProducto">
<wsdl:input wsaw:Action="http://tempuri.org/IListarProductoService/GetAllProducto" message="tns:IListarProductoService_GetAllProducto_InputMessage"/>
<wsdl:output wsaw:Action="http://tempuri.org/IListarProductoService/GetAllProductoResponse" message="tns:IListarProductoService_GetAllProducto_OutputMessage"/>
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="BasicHttpBinding_IListarProductoService" type="tns:IListarProductoService">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
<wsdl:operation name="GetAllProducto">
<soap:operation soapAction="http://tempuri.org/IListarProductoService/GetAllProducto" style="document"/>
</wsdl:operation>
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="IListarProductoService">
<wsdl:port name="BasicHttpBinding_IListarProductoService" binding="tns:BasicHttpBinding_IListarProductoService">
<soap:address location="http://localhost:2777/Service.svc"/>
</wsdl:port>
</wsdl:service>
</wsdl:definitions>
    
```

Fuente: Ejecutor.

4.1.5. Pruebas.

Las pruebas de software permiten verificar y revelar la calidad de un producto software. Son utilizadas para identificar posibles fallos de implementación, calidad, o usabilidad. Las pruebas no tienen el objeto de prevenir errores sino de detectarlos, por lo que se realizaron las pruebas necesarias de los módulos del sistema comercial para comprobar la validez y fiabilidad de la información, y lograr el correcto funcionamiento.

4.1.5.1. Pruebas de Caja Blanca.

Se basa en un examen de los detalles procedimentales. Se comprueban los caminos lógicos del software proponiendo casos de prueba que ejerciten conjuntos específicos de condiciones. Se puede examinar el estado del programa en varios puntos para determinar si el estado real coincide con el estado deseado, con este tipo de prueba se hizo énfasis en la revisión de los puntos que se describen a continuación.

Se configuró los componentes para la validación de los datos utilizando máscaras o las opciones de configuración del componente utilizado.

- Se comprobó que las cantidades máximas de venta de un producto no superen a la cantidad existente en almacén.
- Se revisó que los cálculos de valores totales sean los correctos.

- Se validó la rutina de almacenamiento de datos relacionados, revisando que se almacenen de forma correcta en cada una de sus tablas en la base de datos.
- Se comprobó que los datos que se despliegan en los listados sean exactos.
- Se revisó los mensajes de alerta por posibles errores al ingreso o salida de información.

4.1.5.2. Pruebas de Caja Negra y Usabilidad.

A. Pruebas de Caja Negra.

Las pruebas de caja negra se centran en los requisitos funcionales del software, es decir, la prueba de caja negra permite obtener conjuntos de condiciones de entrada que ejerciten completamente los requisitos funcionales de un programa. Esta prueba sirve para verificar que todas las partes del sistema funcionen correctamente de forma individual como colectiva. A continuación se presenta una tabla donde se especifica las entradas y las salidas deseadas del sistema para el módulo de ventas y los módulos involucrados en el proceso de ventas.

Tabla 18. Especificación de entradas y Salidas.

Módulos	Resultado esperado	Resultado Obtenido
Bienvenida a Sistema de Ventas	Restringir el acceso a personas sin password, registradas en el sistema.	Correcto
Ventana inicial del sistema	Restringir la entrada a cierto tipo de usuarios a áreas determinadas del sistema.	Correcto
Realizar Venta	Desplegar la ventana correspondiente para llenar los campos, insertar venta realizada.	Correcto
Estado de la venta	Presentar una tabla con la información de las ventas, marcar la venta que se desea editar, desplegar la pantalla de editar "venta", actualizar tabla.	Correcto
Editar Estado venta	Editar los campos de la venta, guardar los datos, señalar si se ha cometido un error, señalar que se dio de alta la información correcta.	Correcto
Historial de ventas	Mostrar una lista con todas las ventas, al seleccionar una línea de la lista llenar los campos con la información, poder editar, confirmar que la información se actualizo.	Correcto
Control de ventas	Mostrar la lista con las ventas realizadas, bloquear el botón de agregar detalle, poder actualizar la información general de la venta.	Correcto
Menú de productos	Mostrar el formulario correspondiente, llenar los datos cuando se busca a un producto, abrir la ventana para agregar un producto, poder limpiar campos, editar producto y guardar la información.	Correcto
Menú Categoría	Mostrar el formulario correspondiente, llenar los datos cuando se busca una categoría, abrir la ventana para agregar categoría, poder limpiar los campos, editar y guardar información.	Correcto
Menú Almacén	Poder agregar nuevos productos, mostrar una lista de todos los productos, existencia y ubicación.	Correcto
Menú cliente	Mostrar el formulario correspondiente, llenar los datos cuando se busca un cliente, abrir la ventana para agregar cliente, poder limpiar los campos, editar y guardar información.	Correcto
Menú de administrativos	Encontrar y accesar las ventanas correspondientes.	Correcto
Menú de recursos humanos	Mostrar el formulario correspondiente, llenar los datos cuando se busca a un operador, abrir la ventana para agregar una unidad, poder limpiar campos, editar personal y guardar la información.	Correcto

Módulos	Resultado esperado	Resultado Obtenido
Menú Usuarios	Mostrar el formulario correspondiente, llenar los datos cuando se busca un empleado, abrir la ventana para agregar empleado, poder limpiar los campos, editar y guardar información.	Correcto
Eliminar Registro de Usuario	Realizar la búsqueda previa del empleado, deshabilitar el botón de eliminar hasta que se confirme la búsqueda, abrir una ventana para confirmar la eliminación, eliminar datos.	Correcto
Agregar nuevos usuarios	Mostrar las opciones para poder abrir las ventanas correspondientes.	Correcto

Fuente: Ejecutor.

El objetivo de la prueba es descubrir los puntos débiles y fuertes del sistema así como medir que tan funcional es el mismo, el principal motivo es de tener una clara idea del grado de eficiencia, además de ver las áreas que necesitaban correcciones o ajustes.

Esta prueba considera en su desarrollo entrevistas a la muestra poblacional, en la cual se solicitó a los usuarios del sistema manifestar las observaciones en el momento de uso, la evaluación consideró los aspectos de facilidad de manejo de las ventanas, lo intuitivo que es el uso del sistema, comodidad, y tiempo de uso, aspectos que dan a conocer el grado de satisfacción del usuario. Los resultados obtenidos en esta entrevista se evaluaron y se procedió a realizar los cambios pertinentes.

Cabe resaltar que los usuarios señalaron que; la mayoría de las salidas con el sistema son congruentes con las entradas, sin embargo,

para evitar ambigüedad se realizó la prueba de usabilidad que permite entender de mejor forma las necesidades del usuario.

B. Prueba de Usabilidad.

La finalidad de la prueba de usabilidad es identificar y rectificar las deficiencias y errores existentes. Se intenta asegurar que la aplicación sea fácil de aprender y de usar, su uso sea satisfactorio provea utilidad y funcionalidad, y que cumpla las tareas para las cuales fue desarrollada la aplicación.

Según el autor Jeffrey Rubín existen cuatro tipos diferentes de pruebas, que son prueba de exploración, prueba de evaluación, prueba de validación y prueba de comparación. Para realizar este estudio se implementaron conceptos y metodologías de la llamada prueba de evaluación, cuya finalidad es analizar la facilidad con que un usuario pueda realizar tareas previamente definidas e identificar las deficiencias de usabilidad que se presentan.

Para la realización de la prueba se asignó instrucciones definidas a la muestra poblacional de la investigación para el proceso de venta.

Instrucciones para la prueba:

- Ingresar al sistema mediante su usuario y clave.
- Cargar el menú principal.
- Crear nueva venta.

- Eliminar la venta realizada.
- Buscar los elementos que intervienen en la venta (clientes, producto).
- Búsqueda de las ventas realizadas.
- Actualización los datos de la venta.

Una vez que los usuarios (muestra poblacional) terminaron todas las tareas propuestas se les hizo una serie de preguntas (test) con el propósito de medir el grado de usabilidad y aceptación que ofrece el sistema.

Los aspectos evaluados fueron:

- Interfaz Amigable.
- Grado de complejidad de Uso.
- Cumplimiento de los requerimientos solicitados para ser funcional.
- Tiempo para registrar, seleccionar, ordenar o eliminar datos.
- Tiempo para consultar o imprimir información.
- Tiempo para imprimir boletas o en la consulta de productos.
- Adecuado almacenamiento de información.
- Información disponible al usuario.
- Herramientas para generar reportes.
- Grado de satisfacción de uso.

Los puntos antes señalados se evaluaron a través de una encuesta realizada a los trabajadores de la empresa “Encueta Post Test”, cuyos

resultados se presentan en la parte de Prueba de Hipótesis, asimismo, cabe señalar que la evaluación de estos puntos sirvió para realizar la validación de la hipótesis propuesta en esta investigación.



4.2. Prueba de Hipótesis.

4.2.1. Evaluación del sistema por la empresa.

La evaluación del sistema por parte de la empresa se realizó siguiendo el diseño cuasi experimental el cual propone realizar una encuesta pre test y otra post test a un grupo experimental. En esta investigación se considera grupo experimental a la muestra de la investigación conformada por trabajadores de la empresa BICI&MOTOS.

Diagrama del diseño cuasi experimental.

O₁ X O₂

O₁ Situación Tradicional.

X Contribuye a una adecuada administración de la información en la empresa importadora BICI&MOTOS EIRL.

O₂ Sistema con Enfoque SOA.

4.2.1.1. Situación Tradicional.

El registro, almacenamiento, procesamiento y obtención de datos en archivos Excel por la empresa para fines de la investigación es denominado “Sistema Actual” de la situación tradicional, la cual fue evaluada antes de realizar la implementación del Sistema Propuesto con enfoque SOA, Esta evaluación se realizó mediante el uso de una encuesta (pre test), la cual fue realizada al personal de la empresa BICI&MOTOS EIRL.

Resultados de la Primera Encuesta - Pre test.

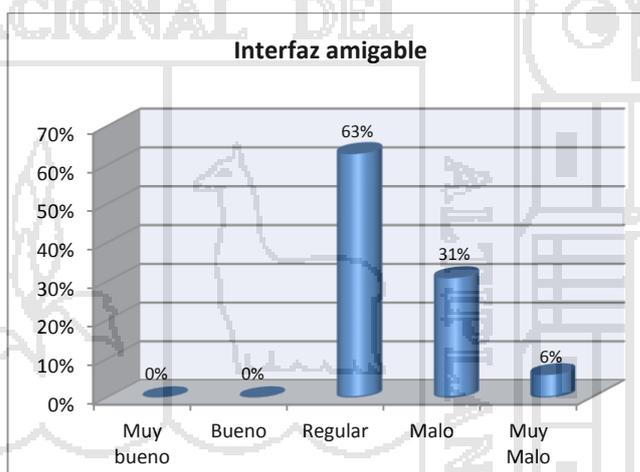
• **Resultados de la Pregunta N° 1.**

Tabla 19. Cuadro estadístico de la pregunta N° 1 – Pre test.

¿Considera usted que la interfaz del sistema es amigable?						
Forma de Medida	Escala					Total
	Muy bueno	Bueno	Regular	Malo	Muy Malo	
Cantidad	0	0	10	5	1	16
Porcentaje	0%	0%	63%	31%	6%	100%

Fuente: Ejecutor.
 Fechas: Setiembre 2012.

Figura 75. Gráfico de barras de la pregunta N° 1 - Pre test.



Fuente: Ejecutor.
 Fechas: Setiembre 2012.

Interpretación: Según los resultados obtenidos de esta pregunta se puede observar que la calificación que logra el sistema tradicional es negativa, ya que los porcentajes obtenidos se distribuyen entre la valoración de regular, malo y muy malo como se puede observar en el grafico anterior.

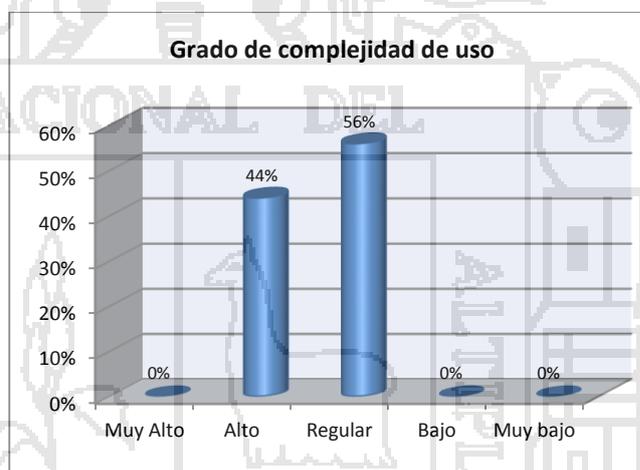
- **Resultado de la Pregunta N° 2.**

Tabla 20. Cuadro estadístico de la pregunta N° 2 - Pre test.

¿Cómo considera que es el grado de complejidad de uso del sistema?						
Forma de Medida	Escala					Total
	Muy Alto	Alto	Regular	Bajo	Muy Bajo	
Cantidad	0	7	9	0	0	16
Porcentaje	0%	44%	56%	0%	0%	100%

Fuente: Ejecutor.
Fechas: Setiembre 2012.

Figura 76. Gráfico de barras de la pregunta N° 2 - Pre test.



Fuente: Ejecutor.
Fechas: Setiembre 2012.

Interpretación: De los resultados obtenidos de esta pregunta la mayoría de los encuestados señala que el grado de complejidad del sistema tradicional es complejo para su uso, como se puede observar en el gráfico anterior el porcentaje total se encuentra distribuido entre la valoración de Alto y Regular.

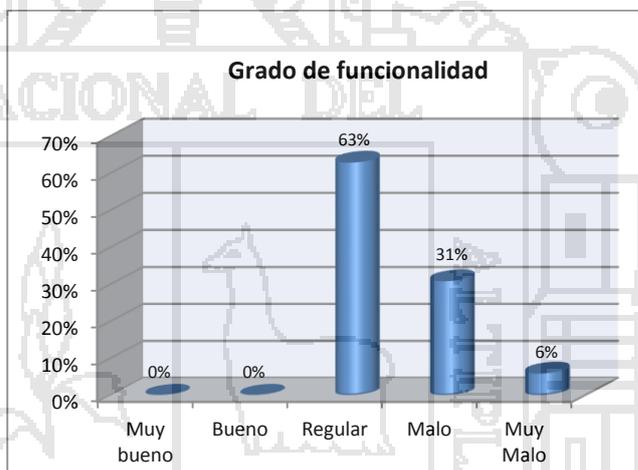
• Resultado de la Pregunta N° 3.

Tabla 21. Cuadro estadístico de la pregunta N° 3 - Pre test.

¿En qué grado considera que el sistema es funcional?						
Forma de Medida	Escala					Total
	Muy bueno	Bueno	Regular	Malo	Muy Malo	
Cantidad	0	0	10	5	1	16
Porcentaje	0%	0%	63%	31%	6%	100%

Fuente: Ejecutor.
 Fechas: Setiembre 2012.

Figura 77. Gráfico de barras de la pregunta N° 3 - Pre test.



Fuente: Ejecutor.
 Fechas: Setiembre 2012.

Interpretación: De los resultados obtenidos se puede apreciar que la valoración de la funcionalidad del sistema tradicional tiende a ser negativa, asimismo del grafico anterior se puede apreciar que los porcentajes obtenidos se encuentran en el rango de Regular, Malo y Muy Malo.

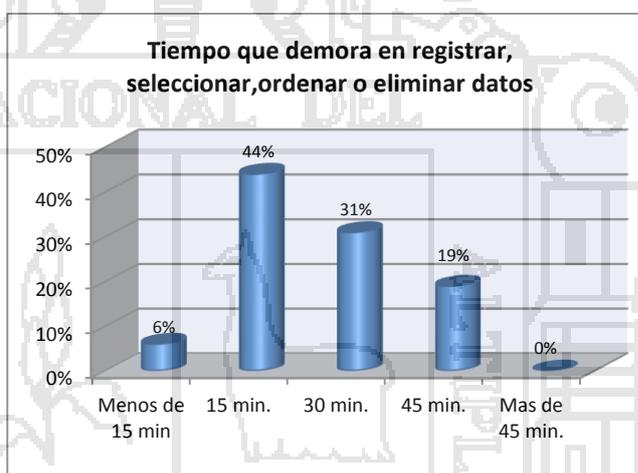
- **Resultado de la Pregunta N° 4.**

Tabla 22. Cuadro estadístico de la pregunta N° 4 - Pre test.

¿Cuánto tiempo se demora en registrar, seleccionar, ordenar o eliminar datos en el sistema?						
Forma de Medida	Escala					Total
	Menos de 15 min	15 min.	30 min.	45 min.	Más de 45 min.	
Cantidad	1	7	5	3	0	16
Porcentaje	6%	44%	31%	19%	0%	100%

Fuente: Ejecutor.
Fechas: Setiembre 2012.

Figura 78. Gráfico de barras de la pregunta N° 4 - Pre test.



Fuente: Ejecutor.
Fechas: Setiembre 2012.

Interpretación: De los resultados obtenidos los cuales se observan en la tabla y el gráfico anterior se puede apreciar la valoración importante en cuanto al tiempo que lleva el registro, selección o eliminación de datos del sistema tradicional bordea el rango de 15 minutos a 45 minutos, pudiéndose apreciar que el promedio de tiempo es los 30 minutos.

- **Resultado de la Pregunta N° 5.**

Tabla 23. Cuadro estadístico de la pregunta N° 5 - Pre test.

¿Cuánto tiempo se demora en consultar o imprimir información del sistema?						
Forma de Medida	Escala					Total
	Menos de 15 min.	15 min.	30 min.	45 min.	Más de 45 min.	
Cantidad	0	2	8	6	0	16
Porcentaje	0%	12%	50%	38%	0%	100%

Fuente: Ejecutor
Fechas: Setiembre 2012

Figura 79. Gráfico de barras de la pregunta N° 5 - Pre test.



Fuente: Ejecutor.
Fechas: Setiembre 2012.

Interpretación: Según el cuadro y el gráfico anterior se puede observar que los resultados obtenidos de esta pregunta el tiempo que lleva el obtener información que resulte ser necesaria y útil del sistema tradicional se encuentra en el rango de tiempo de 30 minutos a 45 minutos.

• Resultado de la Pregunta N° 6.

Tabla 24. Cuadro estadístico de la pregunta N° 6 - Pre test.

¿Cuánto tiempo se demora en imprimir boletas o en la consulta de productos de clientes en el sistema?						
Forma de Medida	Escala					Total
	Menos de 15 min.	15 min.	30 min.	45 min.	Mas de 45 min.	
Cantidad	3	5	8	0	0	16
Porcentaje	19%	31%	50%	0%	0%	100%

Fuente: Ejecutor.
 Fechas: Setiembre 2012.

Figura 80. Gráfico de barras de la pregunta N° 6 - Pre test.



Fuente: Ejecutor.
 Fechas: Setiembre 2012.

Interpretación: El resultado obtenido ante esta pregunta se aprecia en cuadro y grafico anterior, de los mismos se puede observar que el rango de impresión de una boleta o la atención a la consulta de un determinado producto puede tardar hasta los 30 minutos.

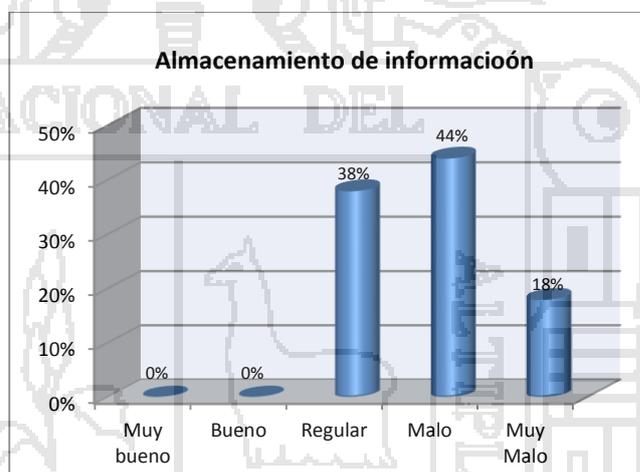
- Resultado de la Pregunta N° 7.

Tabla 25. Cuadro estadístico de la pregunta N° 7 - Pre test.

¿Cómo considera que es el almacenamiento de información en el sistema?						
Forma de Medida	Escala					Total
	Muy bueno	Bueno	Regular	Malo	Muy Malo	
Cantidad	0	0	6	7	3	16
Porcentaje	0%	0%	38%	44%	18%	100%

Fuente: Ejecutor.
Fechas: Setiembre 2012.

Figura 81. Gráfico de barras de la pregunta N° 7 - Pre test.



Fuente: Ejecutor.
Fechas: Setiembre 2012.

Interpretación: Los resultados obtenidos ante esta pregunta se presentan en el cuadro y gráfico anterior, en los mismos en los que se puede apreciar que la valoración en cuanto al almacenamiento de información es negativa, ya que el porcentaje total se encuentra distribuido en los rangos de Regular, Malo y Muy Malo.

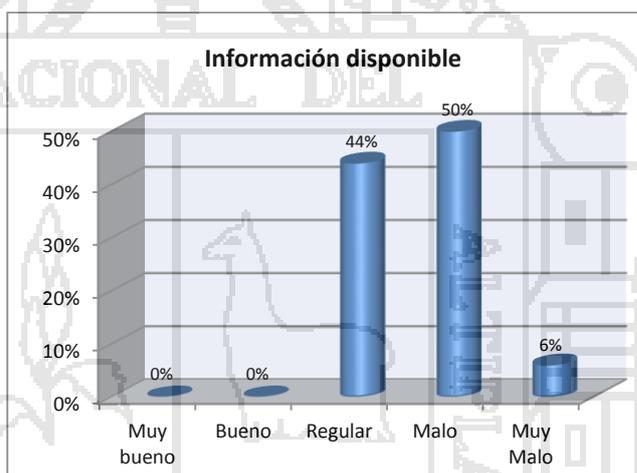
- **Resultado de la Pregunta N° 8.**

Tabla 26. Cuadro estadístico de la pregunta N° 8 - Pre test.

¿Cómo se encuentra la información disponible en el sistema?						
Forma de Medida	Escala					Total
	Muy bueno	Bueno	Regular	Malo	Muy Malo	
Cantidad	0	0	7	8	1	16
Porcentaje	0%	0%	44%	50%	6%	100%

Fuente: Ejecutor.
Fechas: Setiembre 2012.

Figura 82. Gráfico de barras de la pregunta N° 8 - Pre test.



Fuente: Ejecutor.
Fechas: Setiembre 2012.

Interpretación: Del cuadro y gráfico anterior se puede observar que los resultados obtenidos de esta pregunta muestran que la información en el sistema tradicional se encuentra disponible con grado de valoración Regular, Malo y Muy Malo, cuyos porcentajes se pueden apreciar en el cuadro anterior.

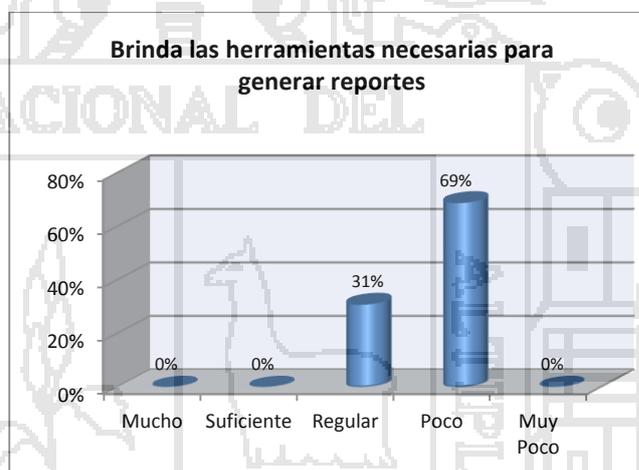
- **Resultado de la Pregunta N° 9.**

Tabla 27. Cuadro estadístico de la pregunta N° 9 - Pre test.

¿El sistema brinda las herramientas necesarias para generar reportes?						
Forma de Medida	Escala					Total
	Muy bueno	Bueno	Regular	Malo	Muy Malo	
Cantidad	0	0	5	11	0	16
Porcentaje	0%	0%	31%	69%	0%	100%

Fuente: Ejecutor
Fechas: Setiembre 2012

Figura 83. Gráfico de barras de la pregunta N° 9 - Pre test.



Fuente: Ejecutor.
Fechas: Setiembre 2012.

Interpretación: Del cuadro y gráfico de esta pregunta se puede observar que los resultados sobre si el sistema tradicional cuenta con herramientas necesarias para generar reportes el porcentaje más alto de la muestra poblacional señalan que cuenta con Pocas herramientas. Se puede observar que el total del porcentaje se distribuye entre la valoración de regular y poco.

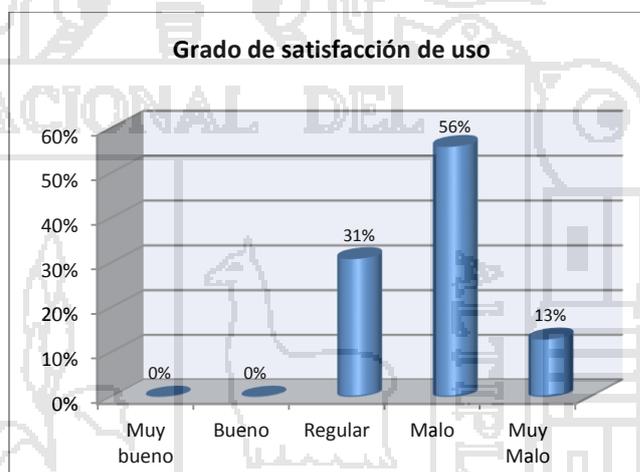
• **Resultado de la Pregunta N° 10.**

Tabla 28. Cuadro estadístico de la pregunta N° 10 - Pre test.

¿Cómo considera que es su grado de satisfacción de uso del sistema?						
Forma de Medida	Escala					Total
	Muy bueno	Bueno	Regular	Malo	Muy Malo	
Cantidad	0	0	5	9	2	16
Porcentaje	0%	0%	31%	56%	13%	100%

Fuente: Ejecutor
 Fechas: setiembre 2012

Figura 84. Gráfico de barras de la pregunta N° 10 - Pre test.



Fuente: Ejecutor.
 Fechas: Setiembre 2012.

Interpretación: El resultado obtenido en esta pregunta representa de forma general el resultado de las preguntas que la anteceden, del cuadro y grafico se puede observar que los encuestados señalan que el grado de satisfacción al usar el sistema tradicional es de Regular, Malo y Muy Malo, esto puede ser debe a diversos factores como los antes señalados en las preguntas antes valoradas.

4.2.1.2. Sistema Propuesto con Enfoque SOA.

Una vez completado las etapas de desarrollo y pruebas del Sistema con Enfoque a SOA se puso a disposición de la Empresa BICI&MOTOS para su valoración y uso, procediéndose con la segunda encuesta (post test).

Resultados de la Segunda Encuesta - Post test.

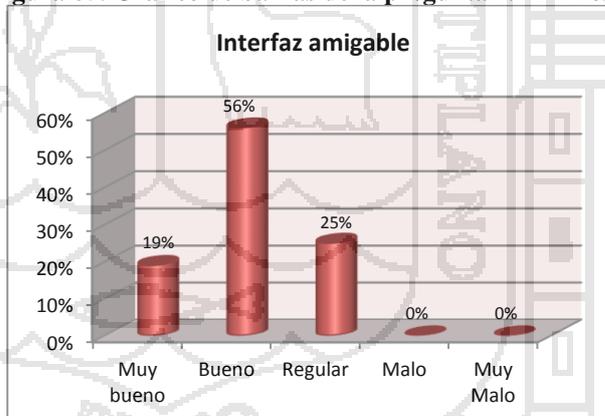
- **Resultado de la Pregunta N° 1.**

Tabla 29. Cuadro estadístico de la pregunta N° 1 – Post test.

¿Considera usted que la interfaz del sistema es amigable?						
Forma de Medida	Escala					Total
	Muy bueno	Bueno	Regular	Malo	Muy Malo	
Cantidad	3	9	4	0	0	16
Porcentaje	19%	56%	25%	0%	0%	100%

Fuente: Ejecutor.
 Fechas: Diciembre 2013.

Figura 85. Gráfico de barras de la pregunta N° 1 – Post test.



Fuente: Ejecutor.
 Fechas: Diciembre 2013.

Interpretación: Según los resultados obtenidos de esta pregunta se puede observar que el porcentaje mayoritario de los encuestados considera que la interfaz es amigable en con una valoración de Bueno.

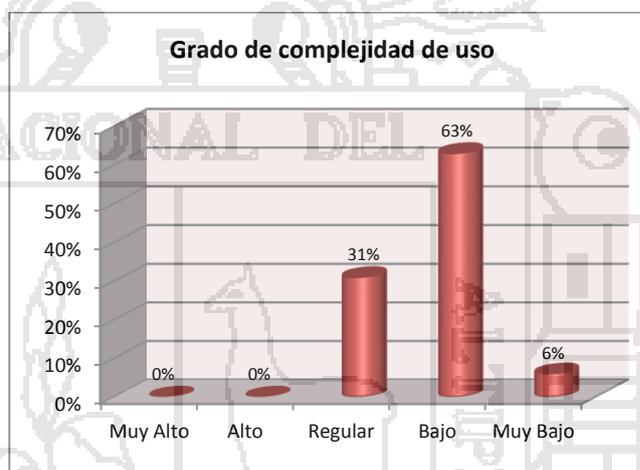
- **Resultado de la Pregunta N° 2.**

Tabla 30. Cuadro estadístico de la pregunta N° 2 – Post test.

¿Cómo considera que es el grado de complejidad de uso del sistema?						
Forma de Medida	Escala					Total
	Muy bueno	Bueno	Regular	Malo	Muy Malo	
Cantidad	0	0	5	10	1	16
Porcentaje	0%	0%	31%	63%	6%	100%

Fuente: Ejecutor.
 Fechas: Diciembre 2013.

Figura 86. Gráfico de barras de la pregunta N° 2 – Post test.



Fuente: Ejecutor.
 Fechas: Diciembre 2013.

Interpretación: Del cuadro y gráfico de esta pregunta se observa que los niveles de complejidad se encuentran en el rango de Regular, Bajo y Muy Bajo siendo el porcentaje más considerable el que tiene valoración de Bajo, ya que esto implica que el sistema no es complejo para cualquier usuario que lo utilice.

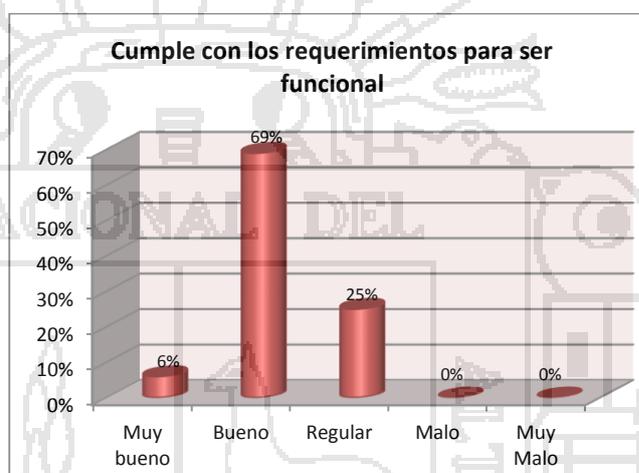
- **Resultado de la Pregunta N° 3.**

Tabla 31. Cuadro estadístico de la pregunta N° 3 – Post test.

¿Considera que el sistema cumple con los requerimientos necesarios para ser funcional?						
Forma de Medida	Escala					Total
	Muy bueno	Bueno	Regular	Malo	Muy Malo	
Cantidad	1	11	4	0	0	16
Porcentaje	6%	69%	25%	0%	0%	100%

Fuente: Ejecutor.
Fechas: Diciembre 2013.

Figura 87. Gráfico de barras de la pregunta N° 3 – Post test.



Fuente: Ejecutor.
Fechas: Diciembre 2013.

Interpretación: Los resultados obtenidos en esta pregunta representan el cumplimiento de los requerimientos que se realizaron para la implementación del sistema propuesto por parte de los trabajadores de la empresa, siendo el porcentaje mayor obtenido de valoración de Bueno, seguido por un 25% con valoración de Regular y un 6% con valoración Muy Bueno.

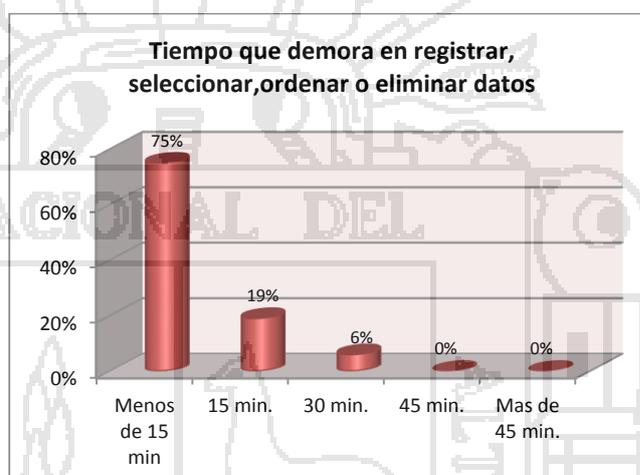
- **Resultado de la Pregunta N° 4.**

Tabla 32. Cuadro estadístico de la pregunta N° 4 – Post test.

¿Cuánto tiempo se demora en registrar, seleccionar, ordenar o eliminar datos en el sistema?						
Forma de Medida	Escala					Total
	Menos de 15 min	15 min.	30 min.	45 min.	Más de 45 min.	
Cantidad	12	3	1	0	0	16
Porcentaje	75%	19%	6%	0%	0%	100%

Fuente: Ejecutor.
 Fechas: Diciembre 2013.

Figura 88. Gráfico de barras de la pregunta N° 4 – Post test.



Fuente: Ejecutor.
 Fechas: Diciembre 2013.

Interpretación: De acuerdo al cuadro y gráfico anterior se observa que el resultado de esta pregunta en cuanto al tiempo que lleva registrar, seleccionar o eliminar datos del sistema tiene como porcentaje mayoritario una valoración de Menos de 15 minutos, seguido por 15 minutos, esto dependiendo de los datos requeridos.

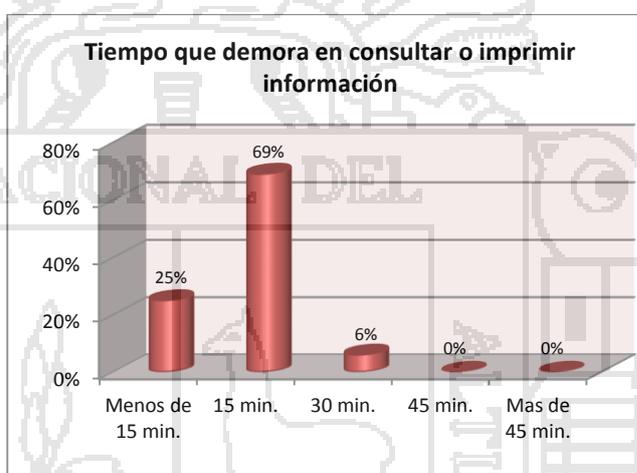
- Encuesta Post Test: Pregunta N° 5.

Tabla 33. Cuadro estadístico de la pregunta N° 5 – Post test.

¿Cuánto tiempo se demora en consultar o imprimir información del sistema?						
Forma de Medida	Escala					Total
	Menos de 15 min	15 min.	30 min.	45 min.	Más de 45 min.	
Cantidad	4	11	1	0	0	16
Porcentaje	25%	69%	6%	0%	0%	100%

Fuente: Ejecutor.
Fechas: Diciembre 2013.

Figura 89. Gráfico de barras de la pregunta N° 5 – Post test.



Fuente: Ejecutor.
Fechas: Diciembre 2013.

Interpretación: De los resultados obtenidos ante esta pregunta se puede observar que el porcentaje más notorio es el que tiene valoración de 15 minutos con un 69%, seguido por Menos de 15 minutos, dependiendo del tipo de operación y del registro de información más elaborada o con más datos.

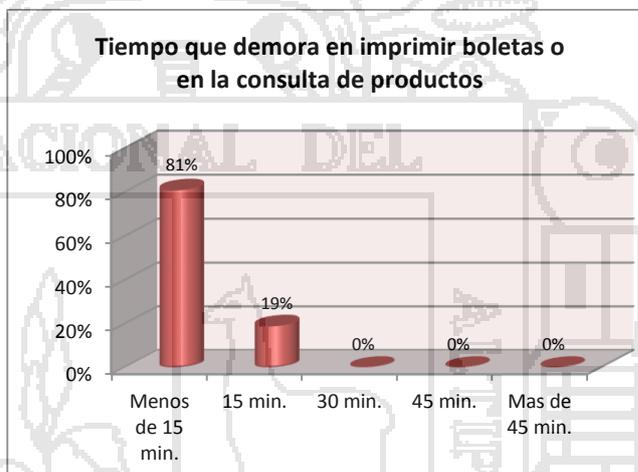
- **Resultado de la Pregunta N° 6.**

Tabla 34. Cuadro estadístico de la pregunta N° 6 – Post test.

¿Cuánto tiempo se demora en imprimir boletas o en la consulta de productos de clientes en el sistema?						
Forma de Medida	Escala					Total
	Menos de 15 min	15 min.	30 min.	45 min.	Más de 45 min.	
Cantidad	13	3	0	0	0	16
Porcentaje	81%	19%	0%	0%	0%	100%

Fuente: Ejecutor.
Fechas: Diciembre 2013.

Figura 90. Gráfico de barras de la pregunta N° 6 – Post test.



Fuente: Ejecutor.
Fechas: Diciembre 2013.

Interpretación: Los resultados obtenidos de esta pregunta se pueden apreciar en el cuadro y gráfico anterior, en los que se aprecia que el tiempo que se demora en imprimir o consultar un producto en el sistema propuesto está dentro del rango de Menos de 15 minutos hasta 15 minutos.

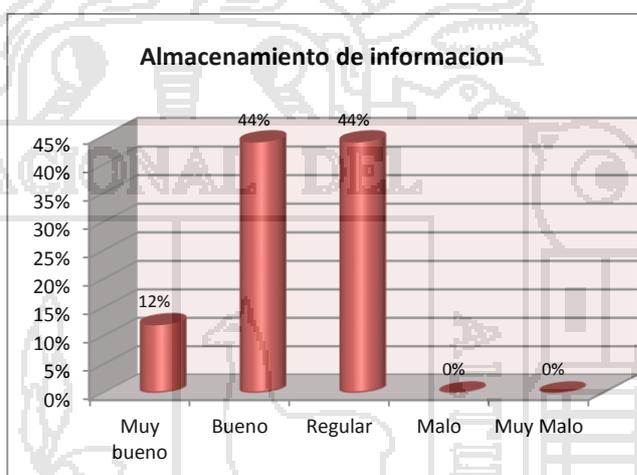
- **Resultado de la Pregunta N° 7.**

Tabla 35. Cuadro estadístico de la pregunta N° 7 – Post test.

¿Cómo considera que es el almacenamiento de información en el sistema?						
Forma de Medida	Escala					Total
	Muy bueno	Bueno	Regular	Malo	Muy Malo	
Cantidad	2	7	7	0	0	16
Porcentaje	12%	44%	44%	0%	0%	100%

Fuente: Ejecutor.
 Fechas: Diciembre 2013.

Figura 91. Gráfico de barras de la pregunta N° 7 – Post test.



Fuente: Ejecutor.
 Fechas: Diciembre 2013.

Interpretación: Los porcentajes obtenidos ante esta pregunta se encuentran en el rango de Regular, Bueno y Muy bueno, estos resultados representan de cierto modo que la información en el sistema propuesto es de acceso limitado por cuentas de usuario, almacenándose los datos solo en el sistema y ya no siendo compartido por los trabajadores o personas externas en algunos casos.

- **Resultado de la Pregunta N° 8.**

Tabla 36. Cuadro estadístico de la pregunta N° 8 – Post test.

¿Cómo se encuentra la información disponible en el sistema?						
Forma de Medida	Escala					Total
	Muy bueno	Bueno	Regular	Malo	Muy Malo	
Cantidad	3	9	4	0	0	16
Porcentaje	19%	56%	25%	0%	0%	100%

Fuente: Ejecutor.
 Fechas: Diciembre 2013.

Figura 92. Gráfico de barras de la pregunta N° 8 – Post test.



Fuente: Ejecutor.
 Fechas: Diciembre 2013.

Interpretación: Los resultados obtenidos muestran que la disponibilidad de la información tiene un alto porcentaje de valoración de Bueno, seguido por un Regular y Muy Bueno, esto representa que el sistema propuesto contiene información consistente y actualizada lo que está disponible de acuerdo al nivel de acceso que se tenga como usuario.

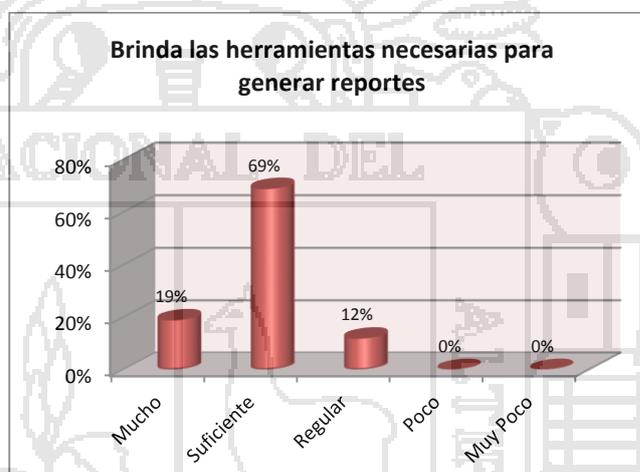
- **Resultados de la Pregunta N° 9.**

Tabla 37. Cuadro estadístico de la pregunta N° 9 – Post test.

¿El sistema brinda las herramientas necesarias para generar reportes?						
Forma de Medida	Escala					Total
	Muy bueno	Bueno	Regular	Malo	Muy Malo	
Cantidad	3	11	2	0	0	16
Porcentaje	19%	69%	12%	0%	0%	100%

Fuente: Ejecutor.
Fechas: Diciembre 2013.

Figura 93. Gráfico de barras de la pregunta N° 9 – Post test.



Fuente: Ejecutor.
Fechas: Diciembre 2013.

Interpretación: Del cuadro y gráfico anterior se puede observar que los resultados obtenidos ante esta pregunta, califican el sistema propuesto con un porcentaje elevado en la valoración de Suficiente con un 69% seguido por Mucho, y Regular.

- Encuesta Post Test: Pregunta N° 10.

Tabla 38. Cuadro estadístico de la pregunta N° 10 – Post test.

¿Cómo considera que es su grado de satisfacción de uso del sistema?						
Forma de Medida	Escala					Total
	Muy bueno	Bueno	Regular	Malo	Muy Malo	
Cantidad	2	14	0	0	0	16
Porcentaje	12%	88%	0%	0%	0%	100%

Fuente: Ejecutor.
 Fechas: Diciembre 2013.

Figura 94. Gráfico de barras de la pregunta N° 10 – Post test.



Fuente: Ejecutor.
 Fechas: Diciembre 2013.

Interpretación: Los resultados obtenidos de esta pregunta se pueden observar en el cuadro y gráfico anterior, en los cuales se puede observar que el nivel de satisfacción de los encuestados ante el uso del sistema propuesto tiene una valoración de Bueno con 89% seguido por un Muy bueno, esto puede representar una recapitulación de las valoraciones anteriores en la preguntas previas a esta y que a la vez la refuerzan.

4.2.1.3. Comparación del Sistema Tradicional con la el Sistema con Enfoque SOA.

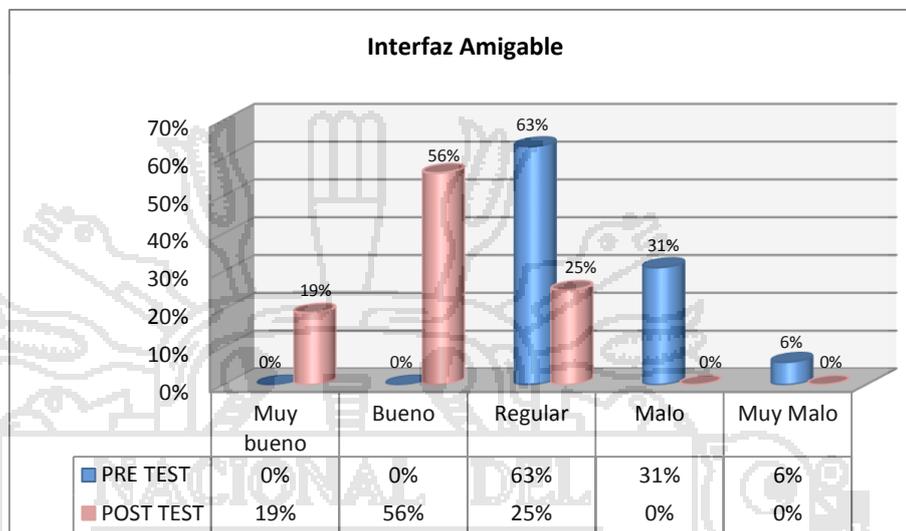
Con los resultados obtenidos en la encuesta de la situación tradicional (pre test) y los resultados obtenidos de la encuesta del uso del Sistema Propuesto con enfoque SOA (post test), se procedió a realizar una comparación de ambas encuestas para poder observar los cambios significativos que tienen ambas, para comprobar si la solución propuesta tuvo o no algún tipo de impacto efecto en la adecuada administración de la información en la Empresa.

- ❖ **PRE TEST:** Sistema de la Situación Tradicional.
- ❖ **POST TEST:** Sistema Propuesto con enfoque SOA.

• Comparación Pre test – Post test de la Pregunta N° 1.

¿Considera usted que es la interfaz del sistema es amigable?

Figura 95. Gráfico de barras de comparación de la pregunta N° 1.



Fuente: Ejecutor.
 Fechas: Febrero 2014.

Interpretación: En el grafico anterior se puede observar la comparación realizada de los porcentajes obtenidos en la encuesta pre test y post test ante la pregunta sobre la interfaz del sistema utilizado, claramente se puede observar que los resultados pre test son un contraste a los resultados post test en cuanto a la valoración que se le asigna. Asimismo, se aprecia que los resultados positivos se obtienen después del uso del Sistema Propuesto. De lo anterior y tomando en cuenta los porcentajes representativos en cada caso se puede señalar que la interfaz del Sistema Propuesto es amigable con una valoración de Bueno, y el Sistema Tradicional con valoración de Regular.

• **Comparación de Encuestas Pre test – Post test: Pregunta N° 2.**

¿Cómo considera que es el grado de complejidad de uso del sistema?

Figura 96. Gráfico de barras de comparación de la Pregunta N° 2.



Fuente: Ejecutor
Fechas: Febrero 2014

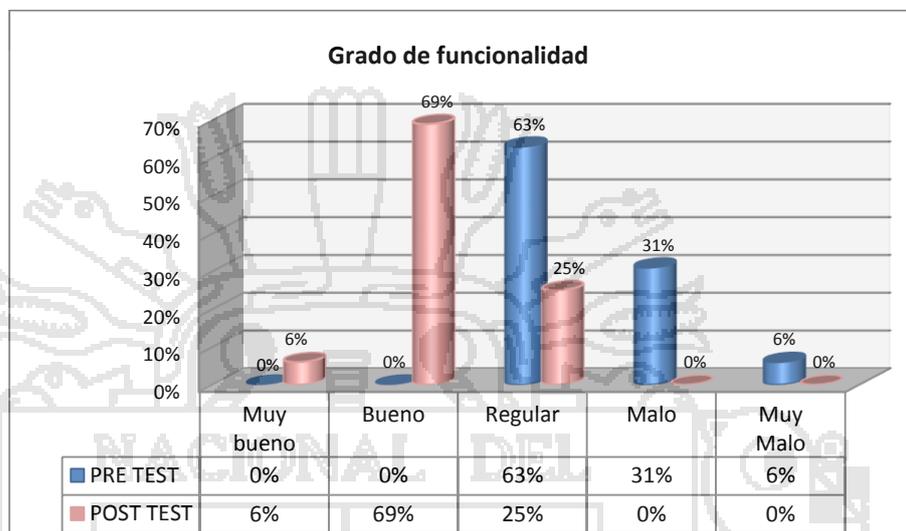
Interpretación: De acuerdo a los resultados obtenidos, los mismos que se presentan en el gráfico anterior, se puede observar el contraste que existe entre los resultados de la encuesta pre test y post test, en el que se observa que el porcentaje total del grado de complejidad al utilizar el sistema tradicional se distribuye entre la valoración de Regular y Alto, a diferencia del porcentaje total del Sistema Propuesto, que se distribuye entre la valoración de Bajo Regular y Muy Bajo.

De lo anterior y tomando en cuenta los porcentajes representativos en cada caso se puede señalar que la complejidad de uso del Sistema Propuesto es Bajo, y del Sistema Tradicional es Regular.

• **Comparación de Encuestas Pre test – Post test: Pregunta N° 3.**

¿En qué grado considera que el sistema es funcional?

Figura 97. Gráfico de barras de comparación de la Pregunta N° 3.



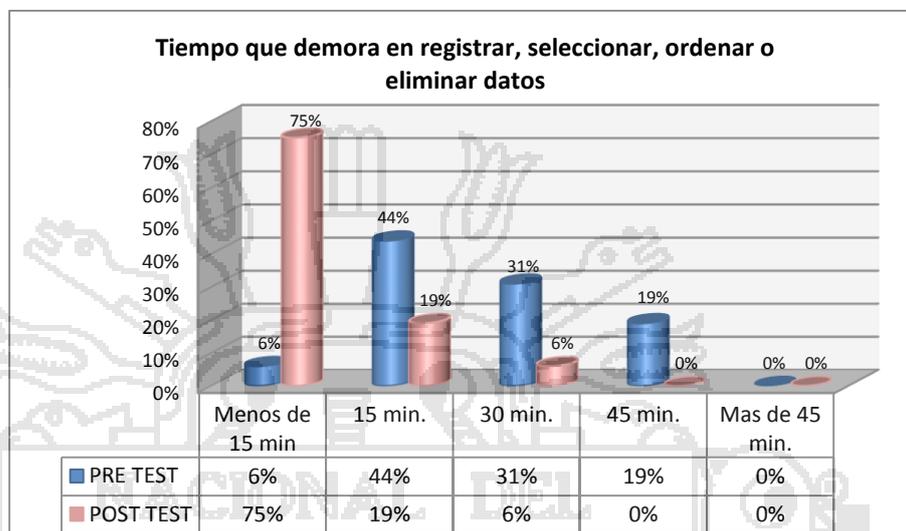
Fuente: Ejecutor.
Fechas: Febrero 2014.

Interpretación: De acuerdo a los resultados obtenidos, los mismos que se presentan en el gráfico anterior, se puede observar el contraste que existe entre los resultados de la encuesta pre test y post test, en el que se observa que el porcentaje total del grado de funcionalidad del Sistema Tradicional se distribuye entre la valoración de Regular, Malo y Muy Malo, a diferencia del porcentaje total del Sistema Propuesto, que se distribuye entre la valoración de Bueno, Regular, y Muy bueno. De lo anterior y tomando en cuenta los porcentajes representativos de cada caso se puede señalar que la funcionalidad del Sistema Propuesto es Bueno, y del Sistema Tradicional que es Regular.

- **Comparación de Encuestas Pre test – Post test: Pregunta N° 4.**

¿Cuánto tiempo se demora en registrar, seleccionar, ordenar o eliminar datos en el sistema?

Figura 98. Gráfico de barras de comparación de la Pregunta N° 4.



Fuente: Ejecutor.

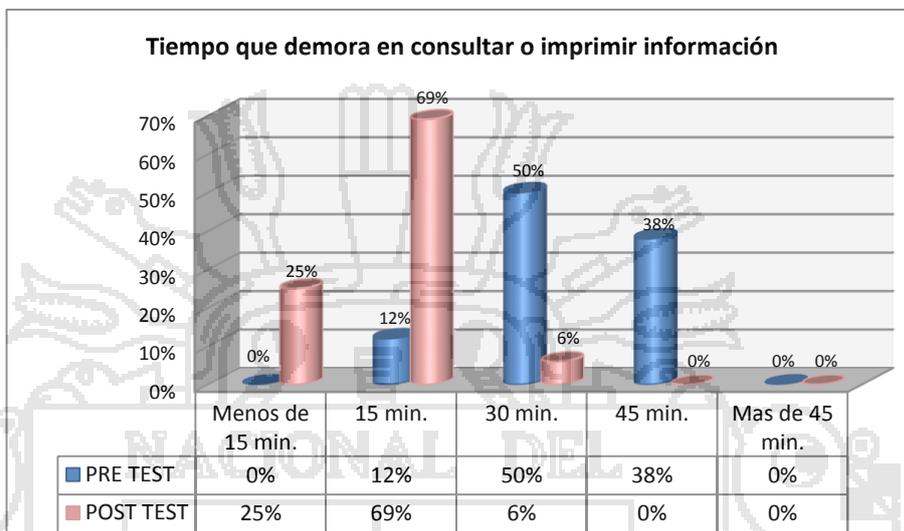
Fechas: Febrero 2014.

Interpretación: De acuerdo a los resultados obtenidos, los mismos que se presentan en el gráfico anterior, se observa que en cuanto al tiempo que se demora en registrar, seleccionar, ordenar o eliminar datos en el Sistema Tradicional el porcentaje total se distribuye entre el rango de Menos de 15 minutos a 45 minutos, y el porcentaje total obtenido del Sistema Propuesto se distribuye en el rango de Menos de 15 minutos a 30 minutos. De lo anterior y tomando en cuenta los porcentajes representativos en cada caso se puede señalar que el tiempo en cuanto al registro, selección, ordenamiento o eliminación de datos en el Sistema Propuesto es de Menos de 15 minutos, y en el Sistema Tradicional es de 15 minutos.

- **Comparación de Encuestas Pre test – Post test: Pregunta N° 5.**

¿Cuánto tiempo se demora en consultar o imprimir información del sistema?

Figura 99. Gráfico de barras de comparación de la Pregunta N° 5.



Fuente: Ejecutor.
 Fechas: Febrero 2014.

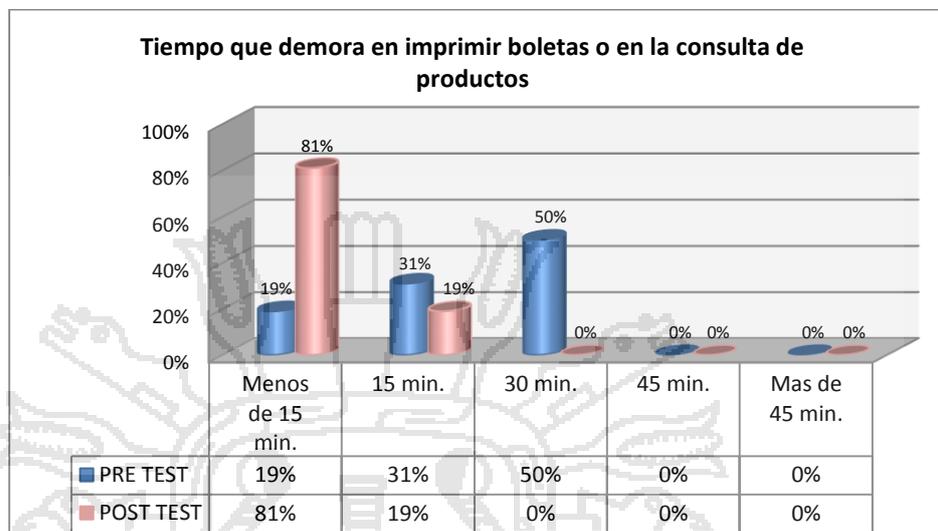
Interpretación: De acuerdo a los resultados obtenidos, los mismos que se presentan en el gráfico anterior, se observa que en cuanto al tiempo que se demora en consultar o imprimir información en el Sistema Tradicional el porcentaje total se distribuye en el rango de 15 minutos a 45 minutos, y el porcentaje total obtenido del Sistema Propuesto se distribuye en el rango de Menos de 15 minutos a 30 minutos.

De lo anterior y tomando en cuenta los porcentajes representativos en cada caso se puede señalar que el tiempo en cuanto al registro, selección, ordenamiento o eliminación de datos en el Sistema Propuesto es de 15 minutos, y en el Sistema Tradicional es de 30 minutos.

- **Comparación de Encuestas Pre test – Post test: Pregunta N° 6.**

¿Cuánto tiempo se demora en imprimir boletas o en la consulta de productos de clientes en el sistema?

Figura 100. Gráfico de barras de comparación de la Pregunta N° 6.



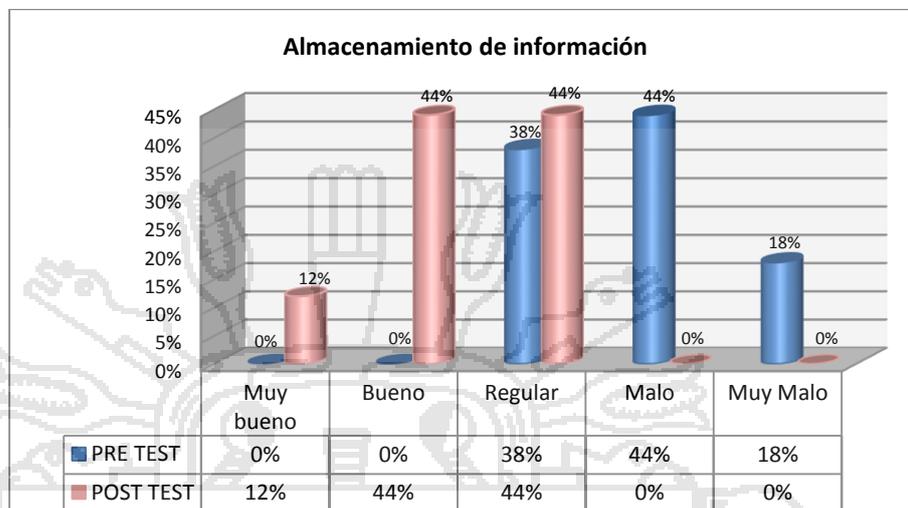
Fuente: Ejecutor.
 Fechas: Febrero 2014.

Interpretación: De acuerdo a los resultados obtenidos, los mismos que se presentan en el gráfico anterior, se observa que en cuanto al tiempo que se demora en imprimir boletas o en la consulta de productos de clientes en el Sistema Tradicional el porcentaje total se distribuye en el rango de 15 minutos a 30 minutos, y el porcentaje total obtenido del Sistema Propuesto se distribuye en el rango de Menos de 15 minutos a 15 minutos. De lo anterior y tomando en cuenta los porcentajes representativos en cada caso se puede señalar que el tiempo que se demora en imprimir boletas o en la consulta de productos de clientes en el Sistema Propuesto es de Menos de 15 minutos y en el Sistema Tradicional es de 30 minutos.

- **Comparación de Encuestas Pre test – Post test: Pregunta N° 7.**

¿Cómo considera que es el almacenamiento de información en el sistema?

Figura 101. Gráfico de barras de comparación de la Pregunta N° 7.



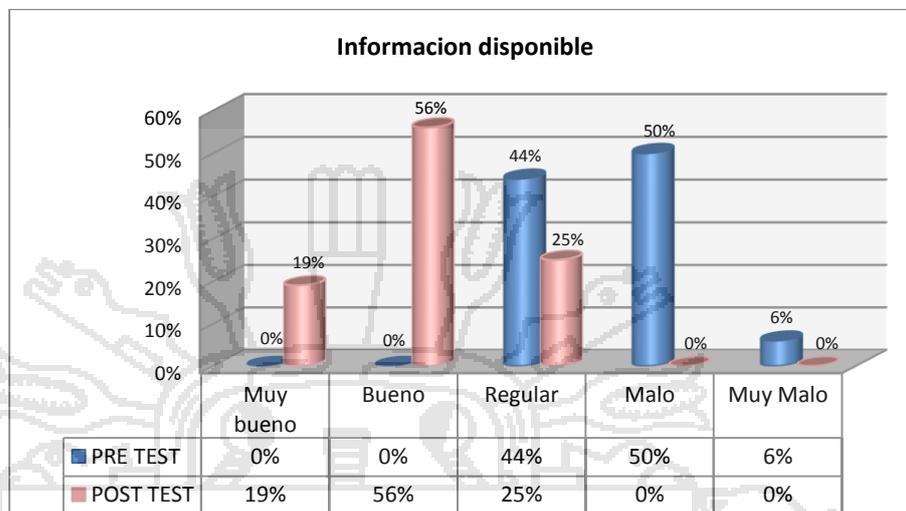
Fuente: Ejecutor.
 Fechas: Febrero 2014.

Interpretación: De acuerdo a los resultados obtenidos, los mismos que se presentan en el gráfico anterior, se puede observar el contraste que existe entre los resultados de la encuesta pre test y post test, en el que se observa que el porcentaje total en cuanto a la valoración del almacenamiento de la información en el Sistema Tradicional se distribuye entre Regular, Malo, y Muy Malo, a diferencia del porcentaje total del Sistema Propuesto, que se distribuye entre Bueno, Regular, y Muy bueno. De lo anterior y tomando en cuenta los porcentajes representativos en cada caso se puede señalar que la el almacenamiento de la información en el Sistema Propuesto es Bueno, y del Sistema Tradicional es Malo.

- **Comparación de Encuestas Pre test – Post test: Pregunta N° 8.**

¿Cómo se encuentra la información disponible en el sistema?

Figura 102. Gráfico de barras de comparación de la Pregunta N° 8.



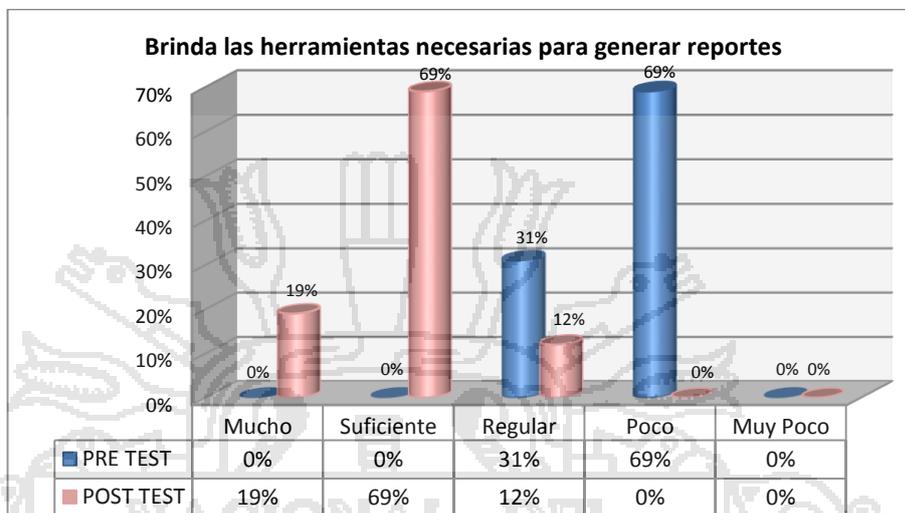
Fuente: Ejecutor.
 Fechas: Febrero 2014.

Interpretación: De acuerdo a los resultados obtenidos, los mismos que se presentan en el gráfico anterior, se puede observar el contraste que existe entre los resultados de la encuesta pre test y post test, en el que se observa que el porcentaje total en cuanto a la valoración de la información disponible en el Sistema Tradicional se distribuye entre Regular, Malo, y Muy Malo, a diferencia del porcentaje total del Sistema Propuesto, que se distribuye entre Bueno, Regular, y Muy bueno. De lo anterior y tomando en cuenta los porcentajes representativos en cada caso se puede señalar que la información disponible en el Sistema Propuesto es Bueno, y en el Sistema Tradicional es Malo.

- .Comparación de Encuestas Pre test – Post test: Pregunta N° 9.

¿El sistema brinda las herramientas necesarias para generar reportes?

Figura 103. Gráfico de barras de comparación de la Pregunta N° 9.



Fuente: Ejecutor.
 Fechas: Mayo 2014.

Interpretación: De acuerdo a los resultados obtenidos, los mismos que se presentan en el gráfico anterior, se puede observar el contraste que existe entre los resultados de la encuesta pre test y post test, en el que se observa que el porcentaje total en cuanto a la valoración de las herramientas necesarias para generar reportes en el Sistema Tradicional se distribuye entre Regular, y Poco, a diferencia del porcentaje total obtenido del Sistema Propuesto, que se distribuye entre Suficiente, Mucho y Regular. De lo anterior y tomando en cuenta los porcentajes representativos en cada caso se puede señalar que el Sistema Propuesto brinda Suficientes herramientas para la generación de reportes a diferencia del Sistema Tradicional que brinda Poco.

- **Comparación de Encuestas Pre test – Post test: Pregunta N° 10.**

¿Cómo considera que es su grado de satisfacción de uso del sistema?

Figura 104. Gráfico de barras de comparación de la Pregunta N° 10.



Fuente: Ejecutor.
 Fechas: Febrero 2014.

Interpretación: De acuerdo a los resultados obtenidos, presentados en el gráfico anterior, se observa el contraste que existe entre los resultados de la encuesta pre test y post test, en el que se aprecia que el porcentaje total en cuanto al grado de satisfacción de uso del Sistema Tradicional se distribuye entre Regular, Malo, y Muy Malo, a diferencia del porcentaje total del Sistema Propuesto, que se distribuye entre Bueno, y Muy bueno, de lo anterior y tomando en cuenta los porcentajes representativos de cada caso, se puede señalar que el grado de satisfacción de uso del Sistema Propuesto es Bueno. Este resultado es importante ya que representa la satisfacción de uso del Sistema propuesto, lo que implica características importantes para el usuario, esto en relación al uso del Sistema Tradicional.

4.2.2. Planteamiento de la Hipótesis.

Para la prueba de la hipótesis se planteó dos, la hipótesis de investigación y una hipótesis nula, De acuerdo a los resultados obtenidos en la prueba T Student se eligió la hipótesis.

4.2.2.1. Hipótesis de Investigación.

H: El modelo del sistema de gestión de ventas desarrollado con arquitectura orientada a servicios (SOA) contribuye a una adecuada administración de la información en la empresa importadora BICI&MOTOS EIRL – Juliaca.

4.2.2.2. Hipótesis Nula.

H₀: El modelo del sistema de gestión de ventas desarrollado con arquitectura orientada a servicios (SOA) no contribuye a una adecuada administración de la información en la empresa importadora BICI&MOTOS EIRL - Juliaca.

4.2.3. Nivel de Significancia.

Para todo valor de probabilidad igual o menor a 0.05 se acepta la Hipótesis de investigación y se rechaza la hipótesis nula.

4.2.4. Zona de Rechazo.

El nivel de probabilidad es de 0.05, para todo valor de probabilidad mayor que 0.05 se acepta la hipótesis nula y se rechaza la hipótesis de investigación.

4.2.5. Estadístico de Prueba.

Para muestras que son menores a 30 es conveniente aplicar la prueba de T – Student para validar la hipótesis, en este caso basándose en los resultados obtenidos de las dos encuestas (pre test y post test). (Ver Tabla 37 y Tabla 38).

Los resultados de la tabla 39 se procesaron devolviéndose como resultado el valor de T – Student.

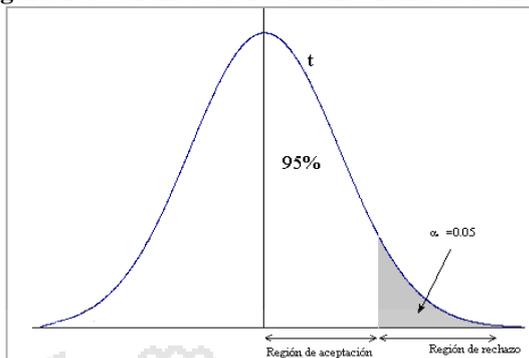
La fórmula aplicada es la siguiente.

$$t = \frac{\bar{d}}{S_d / \sqrt{n}}$$

Dónde:

- t es la prueba de T – Student.
- \bar{d} es el promedio de las diferencias.
- S_d es la suma de los cuadrados de esas diferencias.
- n es el tamaño de la muestra.

Figura 105. Gráfico de T Student –zona de rechazo.



Fuente: Ejecutor.
 Fechas: Febrero 2014.

Tabla 39. Cuadro estadístico resumen de la encuesta Pre test.

ENCUESTA PRE - TEST																
Nivel de Calificación	Muestra															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Muy Bueno	0	0	0	0	0	5	10	5	0	0	0	0	0	0	0	0
Bueno	4	8	8	4	0	8	0	8	12	8	8	4	4	0	4	4
Regular	21	3	15	18	21	3	15	15	12	6	12	15	15	18	12	18
Malo	4	14	4	6	2	12	4	4	4	10	6	8	8	6	10	6
Muy Malo	0	0	1	0	2	0	1	0	1	1	1	0	0	1	0	0
Sub Total	29	25	28	28	25	28	30	32	29	25	27	27	27	25	26	28
Vigesimal	12	10	11	11	10	11	12	13	12	10	11	11	11	10	10	11

Fuente: Ejecutor.
 Fechas: Febrero 2014.

Tabla 40. Cuadro estadístico resumen de la encuesta Post test.

ENCUESTA POST - TEST																
Nivel de Calificación	Muestra															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Muy Bueno	10	5	10	15	15	10	10	20	35	20	10	10	15	20	5	5
Bueno	24	20	20	20	16	24	24	20	8	4	24	16	20	16	20	28
Regular	3	12	9	3	3	3	3	0	3	12	3	12	3	6	9	6
Malo	2	0	0	2	2	2	2	2	0	2	2	0	2	0	2	0
Muy Malo	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Sub Total	39	37	39	40	36	39	39	42	46	38	39	38	40	43	36	39
Vigesimal	16	15	16	16	14	16	16	17	18	15	16	15	16	17	14	16

Fuente: Ejecutor.
 Fechas: Febrero 2014.

En el siguiente cuadro se presenta el resumen de los resultados de la encuesta pre test y post test en sistema vigesimal para hallar el valor de la T Student.

Tabla 41. Resultados de las encuesta pre test y post test.

Encuesta	Muestra															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Pre Test	12	10	11	11	10	11	12	13	12	10	11	11	11	10	10	11
Post Test	16	15	16	16	14	16	16	17	18	15	16	15	16	17	14	16

Fuente: Ejecutor.

Fechas: Febrero 2014.

$$T - \text{Student} = 0.000000000000000405$$

Además se halló los valores del promedio y la desviación estándar.

Tabla 42. Promedio y Desviación estándar del pre test y post test.

Prueba T para $p < 0,05$		
	Pre test	Post test
Promedio	10	16
Desviación Estándar	0.7	1.0

Fuente: Ejecutor.

Fechas: Febrero 2014.

Para Ubicar el rango en el que se encuentran los valores obtenidos tanto en el pre test y el post test se pondera los niveles en base al sistema vigesimal (Ver Tabla 41).

Tabla 43. Nivel de aceptación en rango vigesimal.

Nivel de Aceptación	Rango Vigesimal
Alto	15 – 20
Medio	8 – 14
Bajo	1 – 7

Fuente: Ejecutor.

Fechas: Febrero 2014.

4.2.6. Decisión.

De los valores adquiridos que se observan en la Tabla 39 se obtiene el valor de T – Student que es de 0.00000000000000405, el cual es menor al error permitido (5%) que es de 0.05, por lo tanto, se rechaza la hipótesis nula y se acepta la hipótesis de investigación.

4.2.7. Resultado de la Prueba de la Hipótesis.

Para ver en qué nivel contribuye a una adecuada administración de la información se pondero los niveles Alto, Medio y Bajo en base al sistema vigesimal. De acuerdo a los valores obtenidos se ubican el promedio del pre test (sistema tradicional) y al promedio del post test (implementación del sistema de gestión de ventas desarrollado con arquitectura orientada a servicios (SOA)) en la Tabla 41.

En la antes citada tabla se observa que el nivel de contribución para la adecuada administración de la información que ofrece el Sistema de gestión de ventas desarrollado con arquitectura orientada a servicios (SOA) se ubica en un nivel Alto.

Tabla 44. Ubicación de los test de acuerdo a promedio.

Ubicación en Niveles de los Test

	Pre test	Post test
	Sistema Tradicional	Sistema Propuesto
Promedio	10	16
Nivel	Medio	Alto

Fuente: Ejecutor
Fechas: Febrero 2014

Demostrada la hipótesis se concluye que: **“El modelo del sistema de gestión de ventas desarrollado con arquitectura orientada a servicios (SOA) contribuye a una adecuada administración de la información en la empresa importadora BICI&MOTOS EIRL - Juliaca”**, con un promedio de calificación de 16.



CAPÍTULO V

CONCLUSIONES.

Después del desarrollo de la presente investigación se llegó a las siguientes conclusiones:

Primero.

La especificación concreta de los requerimientos y la utilización de una metodología para ello conduce a un adecuado modelado de negocios uno de los puntos fundamentales para poder desarrollar una aplicación de negocios porque con ello se identifica los procesos de negocio. Utilizar una metodología para documentar los requerimientos del usuario es sumamente importante puesto que esto permite obtener un análisis y diseño adecuado, razón por la cual en esta investigación para la obtención de los requerimientos se utilizó la metodología de documentación de requerimientos centrada en el usuario DOrCU, caracterizada por su orientación al usuario y su flexibilidad, que puede ser aplicada a diferentes paradigmas ya que ofrece la libertad de acción para realizar la

selección e integración de las herramientas empleadas en cada una de las etapas. Este proceso se desarrolló en la etapa de exposición de resultados.

Segundo.

La lógica del negocio de acuerdo a los requerimientos de la empresa obtenidos en la etapa inicial sirve de base para que se pueda realizar la orquestación de servicios logrando el diseño de la arquitectura del sistema. Además la etapa de diseño es importante por que el modelado de servicios ayuda a visualizar como están organizados los mismos y comprobar el comportamiento de cada servicio.

Tercero.

En la etapa de desarrollo se comprende mejor el funcionamiento de la arquitectura orientada a servicios y las grandes ventajas ofrecidas por esta como son: la Interoperabilidad, la Reutilización y la Escalabilidad. La Interoperabilidad se refiere a que se puede consumir el Servicio con cualquier cliente, la Reutilización, se da porque estos servicios en un futuro se pueden ofrecer a otros clientes o se pueden utilizar para otros proyectos y finalmente la Escalabilidad referida al poder mejorar la aplicación cliente. Otro punto importante, es el buen análisis y diseño que facilita la programación y el tiempo de entrega, porque se tiene en concreto lo que se va a realizar, de esta manera es más fácil obtener los servicios.

Cuarto.

Programar en C# ayuda mucho, ya que con ello ahorramos tiempo, por ser intuitivo y accesible de comprender, así como el funcionamiento y estabilidad en la integración de datos. No se debe olvidar que para poder desarrollar algo primero se tiene que entender que hace cada parte. El WSDL es algo muy importante a la hora de ofrecer nuestro servicio o consumirlo, ya que en el especifican las descripciones de cada servicio por ejemplo que parámetros y que tipo de parámetros requiere.

Quinto.

El desarrollo e implementación del sistema de gestión de ventas está orientado principalmente a usuarios habituales de la empresa, por lo que una vez implantado ellos acceden a la información, almacenan datos y generan información, representando esto un aporte significativo a la empresa, para validar que este genera un aporte positivo y a su vez cumple con los requerimientos indicados por la empresa, se accede a las encuestas realizadas en los Test que muestra el grado de relación entre el sistema propuesto y el sistema tradicional, ello una vez que los usuarios han utilizado el sistema propuesto, obteniéndose la calificación del desempeño del sistema propuesto con un promedio de bueno con 16 de preferencia en relación al 10 del sistema tradicional, resultado obtenido de seguir el diseño de investigación planteado.

Sexto.

El adecuado desarrollo secuencial de los objetivos específicos planteados hizo posible el cumplimiento del objetivo principal de la investigación el cual fue: Desarrollar e implementar un modelo de sistema de gestión de ventas, empleando la arquitectura orientada a servicios (SOA) que permita una adecuada administración de la información en la empresa importadora BICI&MOTOS EIRL.

En último lugar se precisa que la presente investigación se realizó con la finalidad de demostrar la hipótesis planteada al inicio de la misma, para lograr este fin se siguió los procedimientos de una metodología y diseño de investigación, proceso por el cual se realizó la comprobación del supuesto llegándose al final de la investigación a su demostración tanto de forma empírica como científica por lo que se afirma que: “El modelo del sistema de gestión de ventas desarrollado con arquitectura orientada a servicios (SOA) contribuye a una adecuada administración de la información en la empresa importadora BICI&MOTOS EIRL - Juliaca”.

CAPÍTULO VI

RECOMENDACIÓN Y SUGERENCIAS

1. Teniendo en consideración que toda aplicación o sistema está centrada en los requerimientos de un determinado usuario o usuarios se recomienda utilizar una metodología adecuada, la metodología utilizada en esta investigación fue Dorcu la misma que se centra en el usuario, es importante que tener claro los requerimientos para lograr un adecuado diseño y posterior desarrollo, asimismo es recomendable contar con el apoyo técnico y de negocio, con el fin de evitar mal interpretaciones en las necesidades, y que las mismas sean reflejadas en el análisis y diseño.
2. La etapa de análisis y diseño es importante porque en esta etapa se conceptúa la lógica del negocio importante para lograr el adecuado diseño de la arquitectura y la orquestación de los servicios, por lo que es recomendable documentar todos los procesos, para su posterior uso en la etapa de desarrollo, y documentación del proyecto, asimismo.

3. La utilización de servicios ofrece ventajas, como la facilidad de programación y tiempo de entrega, por lo que es recomendable como punto importante tener documentación clara de las etapas anteriores a esta etapa, ya que esto facilitara la comprensión de lo que se quiere realizar, asimismo se debe comprender que el proceso de desarrollo de un sistema es dinámico que necesita constantes adaptaciones de acuerdo a las solicitudes del usuario y estas serán mínimas si se trabaja adecuadamente los requerimientos.
4. Se recomienda usar un lenguaje de programación con buen funcionamiento y estabilidad en la integración de datos, y que también puedan ser intuitivos y accesibles a la comprensión ya que son de gran utilidad a la hora de la realizar proyectos de programación. Asimismo se debe tener en cuenta que el modelo arquitectural (SOA) resulta enormemente válido y flexible para la implementación de servicios muy reutilizables, e invocables desde aplicaciones cliente de cualquier tipo.
5. Se recomienda pruebas y verificaciones previas a la implantación del sistema en la empresa, pues esto ayudara a identificar errores lo que permitirá corregir posibles fallas en el sistema, asimismo se recomienda validar el sistema desarrollado luego de terminar la etapa de las capacitaciones correspondientes a los usuarios sobre el uso, para poder documentar el proceso y poder presentar el informe final sobre el proyecto. Se debe incidir que se debe contar con el respaldo de la empresa, ya que la implantación del sistema genera cambios en algunos procesos internos de la empresa.

Los sistemas desarrollados con arquitectura orientada a servicios tienen un gran alcance respecto a las funcionalidades y capacidades que pueden brindar, así como medio para

incrementar la calidad del servicio y los procedimientos internos es por eso que se sugiere hacer futuras investigaciones para evaluar tendencias de uso de esta arquitectura.



BIBLIOGRAFÍA

- Aguilar Esteban, J. M. (2007). *Sistema multicliente con arquitectura SOA basado en tecnología .NET*. Tesis de Grado.
- AulaFacil S.L. (2009). *AulaFacil.com*. Obtenido de <http://www.aulafacil.com/articulos/cms/t5/quienes-somos>
- Báez, M. G., & Barba Brunner, S. I. (s.f.). *Departamento de informatica*. Recuperado el 16 de Mayo de 2012, de <http://www.inf.puc-rio.br/wer01/Mod-Req-1.pdf>
- Baumann, H., Grässle, P., & Baumann, P. (2005). *Uml 2.0 in Action: A Project-based Tutorial*. Packt Publishing.
- Bell, M. (2008). *Service Oriented Modeling*. New Jersey: Jonh Wiley & Sons Inc.
- Bon, J. v. (2008). *Fundamentos de la gestión de servicios de TI: basada en ITIL (3ra ed.)*. Gran Bretaña: Van Haren Publishing, Zaltbommel.
- Brito Urbina, F. J. (2011). *Desarrollo de un sistema, basado en Arquitectura Orientada a Servicios (SOA), para la automatización de los procesos de entrada y salida de equipos eléctricos, en la Gerencia de Servicios Eléctricos, PDVSA*. Informe de pasantías de grado, Núcleo de Monagas.
- Cabrera, E. (20 de marzo de 2005). *Monografias.com S.A.* Recuperado el 12 de abril de 2012, de Monografias.com S.A.: www.monografias.com/trabajos14/control/control.shtml
- Caicedo, S. M., Bustos R., L. S., & Rojas Díaz, J. (Diciembre de 2008). *Scientia et Technica*. Recuperado el 22 de Abril de 2012, de <http://revistas.utp.edu.co/index.php/revistaciencia/article/download/3093/1693>
- Campos, L., Chías, L., Escartín, L., Espinoza, E., Hernández, A., Hinojosa, J., . . . Valdés, J. (2000). *Emprendedores para el desarrollo social*. In *Emprendedores para el desarrollo social* (p. 70). Mexico: Trillas.
- Charaja, F. (2004). *Investigacion Cientifica*. Lima: Nuevo Mundo.
- Crece Negocios. (2011). *Crece Negocios*. Recuperado el 29 de mayo de 2012, de <http://www.crecenegocios.com/informacion/>
- Date, C. J. (2001). *Introducción a los sistemas de Base de Datos (7ma ed.)*. Pearson Educación.
- Educ.ar S.E. . (s.f.). *educ.ar*. Obtenido de www.educ.ar

- Espinosa Díaz, E. d. (2009). *Propuesta De Implantacion De La Arquitectura Bpm/Soa Para Agilizar La Gestion Comercial En La Cnt*. Tesis de Grado, Universidad San Francisco De Quito, Quito.
- Fernández Lino, C. R. (2009). *Desarrollo de Aplicaciones Basadas en Servicios Aplicado a Incorporar Nuevas Tecnologías*. Tesis de Grado, Universidad Mayor de San Andres, La Paz.
- Gomez Vieites, a., & Suares Rey, C. (2006). *Sistemas de Información Herramientas prácticas para la gestión empresarial*. RA-MA EDITORIAL.
- Guana Garcés, V. M. (2009). <https://docs.google.com>. Recuperado el 20 de Abril de 2012, de <http://wwwest.uniandes.edu.co/~vm.guana26/papers/SOA%20para%20Modelar%20Procesos%20de%20Negocio.pdf>
- Hax, A., & Majluf, N. (1997). *Estrategias para el Liderazgo Competitivo. De la visión a los resultados*. Editorial Dolmen.
- Hernández González, A. (diciembre de 2004). *Identificación De Procesos De Negocio*. Obtenido de <http://rii.cujae.edu.cu/index.php/revistaind/article/viewFile/168/152.pdf>
- Hernandez Sampieri, R., Fernández Collado, C., & Baptista Lucio, P. (1997). *Metodología de la Investigación* (cuarta ed.). México: Mc Graw Hill.
- IBM developerWorks. (1999). Recuperado el 1 de mayo de 2012, de <http://www.ibm.com/developerworks/ssa/webservices/tutorials/ws-soa1/index.html>
- Jeffrey Rubín. (14 de mayo de 1994). *Usabilidad*. Recuperado el 30 de mayo de 2012, de <http://es.wikipedia.org/w/index.php?title=Usabilidad&oldid=56159009>
- Juric, M. B., Loganathan, R., Sarang, P., & Jennings, F. (2007). *SOA Approach to Integration XML, Web services in real-world SOA projects*. birmingham: Pakt Publishing Ltd.
- Kenneth E. Kendall, J. E. (2005). *Análisis y diseño de sistemas* (6ta ed.). México: Pearson Education.
- Korth, H., Silberschatz, A., & S., S. (2005). *Fundamentos de Base de Datos*. México: McGraw Hill.
- Krafzig, D., Banke, K., & Slama, D. (2005). *Enterprise SOA, Service Oriented Architecture Best Practices*. Hagerstown, Maryland: Pearson Education Inc.
- Laudon, K. C., & Laudon, J. P. (2004). *Sistemas de Infomación Gerencial*. Mexico: Pearson.
- Leffingwell, D. (2010). *Agile software requirements Lean Requirements Practices for teams, Programs, and the enterprise*. Massachusetts: Addison-Wesley Professional.
- López, G., Echeverría, A., Fierro, P., & Jeder, I. (28 de Diciembre de 2010). *Facultad de Ingeniería U.N.P.S.J.B*. Recuperado el 1 de mayo de 2012, de www.ing.unp.edu.ar/wicc2007/trabajos/ISBD/116.pdf
- MacVittie, L. (2007). SOA challenges and solutions . *F5 Networks* , 3, 4.

- Masadelante.com. (2013). *Significados*. Obtenido de <http://www.masadelante.com/faqs/software-hardware>
- Medina Bonilla, L. E., & Pinedo Marín, L. E. (2010). *Implementación de un sistema de integración para las bibliotecas municipales de Lima y Callao utilizando SOA y J2ME*. Tesis de Grado, Universidad Nacional Mayor De San Marcos, Lima.
- Meltom Technologies. (2013). *degerencia.com*. Obtenido de http://www.degerencia.com/tema/tecnologia_de_informacion
- Microsoft Corporation. (2006). <http://www.microsoft.com/spain/iom/recursos/apo.aspx>. Recuperado el 25 de abril de 2012, de http://download.microsoft.com/download/c/2/c/c2ce8a3a-b4df-4a12-ba18-7e050aef3364/070717-Real_World_SOA.pdf
- Nadler, D. A., & Tushman, M. L. (1997). *Competing by Design - the Power of Organizational Architecture*. New York, NY: Oxford University Press, Inc.
- Naiburg, E. J., & Maksimchuk, R. A. (2001). *UML for Database Design*. Boston Massachusetts: Addison-Wesley.
- Odell, J. J. (1998). *Advanced Object-Oriented Analysis & Design Using UML*. New York, NY: SIGS Books & Multimedia.
- Ortega Paredes, J. G. (2012). Contabilidad Paradigma de reconstrucción a través del giro informático. En J. G. Ortega Paredes, *Contabilidad Paradigma de reconstrucción a través del giro informático* (pág. 33 a 37). Madrid: Editorial EAE.
- OverTI. (2008-2011). *OverTI*. Recuperado el 13 de Agosto de 2011, de Consultoría y Soluciones para la Gestión de Servicios TI: <http://www.overti.es/>
- Perdita, S., & Rob, P. (2002). *Utilización de UML en Ingeniería del Software con Objetos y Componentes*. Madrid: Addison Wesley.
- Peter Rob, C. C. (2006). *Sistemas de bases de datos: diseño, implementación y administración* (5ta ed.). México: International Thomson.
- Pontificia Universidad Católica del Perú. (2011). *Antecedentes de la investigación: "Análisis, diseño e implementación de un Sistema de Administración de incidentes en atención al cliente para una empresa de telecomunicaciones"*. Recuperado el 25 de Marzo de 2012, de <http://tesis.pucp.edu.pe>
- Pressman, R. S. (2002). *Ingeniería de Software un Enfoque Práctico* (6ta ed.). Madrid España: McGraw Hill.
- Pressman, R. S. (2005). *Ingeniería de Software un enfoque práctico* (Quinta ed.). Madrid: Pearson Educación S.A.

- Quintana Méndez, A. (26 de setiembre de 2011). *Software de Sistemas*. Recuperado el 1 de Mayo de 2012, de <http://softwaredesistemaarianaquintana.blogspot.com/2011/09/sistemas-monousuario-y-multiusuario.html>
- Rebolledo Saavedra, G. (2013). *Definicion.de*. Recuperado el 25 de abril de 2012, de <http://b3.bibliotecologia.cl/ar-gestion.htm>
- Rob, P., & Coronel, C. (2006). *Sistemas de bases de datos: diseño, implementación y administración* (Quinta ed.). México: thomson editores S.A.
- Rodríguez de Soto, A., & Cuervo Fernández, E. (2006). Nuevas Tendencias en Sistemas de Información: Procesos y Servicios. *Pecunia*, 129-158.
- Rojas, J. A. (2009). "Propuesta para el uso de técnicas de gestión de los procesos de atención en el área de Soporte Técnico con base en ITIL, para la empresa Corporación Vega S.A.C." Recuperado el 20 de Abril de 2011, de <http://www.buenastareas.com>
- S. Dumas, J., & C. Redish, J. (1994). *A Practical Guide to Usability Testing*. Oregon, USA: Intellect Books.
- Sagal Pazmiño, D. R. (2008). *Diseño de la SOA para Andes Petroleum LTD*. Tesis de Grado, Escuela Politécnica Nacional.
- Sánchez Garreta, J. S. (2003). *Ingeniería de proyectos informáticos*. Universitas.
- Silberschatz, A., Korth, H. F., & Sudarshan, S. (2002). *Fundamentos de Base de Datos* (cuarta ed.). Madrid: McGraw Hill.
- Stevens, P. y. (2002). *Utilización de UML Ingeniería de Software con Objetos y Componentes*. Madrid España: Addison Wesley.
- Tamayo y Tamayo, M. (2001). Metodología de la Investigación Científica. En M. Tamayo y Tamayo, *Metodología formal de la Investigación Científica* (pág. 125). México, DF.: Limusa S.A.
- UCLM, D. d. (1994). *Notas de Visión y apuntes sobre la Ingeniería del Software* (3ra ed.). España: Universidad de Castilla – La Mancha.
- Vergara, G. (31 de Marzo de 2009). *MejoraTuGestion.com*. Recuperado el 17 de Marzo de 2012, de www.mejoratugestion.com/mejora-tu-gestion/que-es-un-sistema-de-gestion
- Zuboff, S. (s.f.). Las Nuevas Reglas Del Juego En La Organizacion A Partir De La Computadora. *Biblioteca-Harvard-De-Administracion-De-Empresas*, 19(370).

ANEXOS

A. ENTREVISTA REALIZADA A LOS TRABAJADORES DE LA EMPRESA BICI&MOTO PARA LA OBTENCIÓN DE LOS REQUERIMIENTOS:

Esta entrevista se realizó con el fin de obtener los requerimientos para realizar el análisis y diseño del sistema, y tener conocimiento de las actividades de la empresa:

Cuestionario:

1. ¿Cuál es el área de la empresa a la cual usted pertenece?
2. ¿Dónde se sitúan en el uso de tecnología de información en comparación con la competencia?
3. ¿Se encuentra en la capacidad de responder rápidamente a las condiciones del mercado y asegurar la productividad de la empresa?
4. ¿Qué información se requiere para consolidar su información y llevar un control adecuado de la empresa?
5. ¿Cómo obtiene esta información en este momento?
6. ¿Cuál es la información que más le ayudan a hacer su trabajo?
7. ¿Qué datos son importantes? ¿Cómo usa esta información?
8. ¿Qué preguntas le han quedado sin responder simplemente porque usted sabe que no hay respuesta disponible?
9. ¿Existe alguna otra información faltante que crea puede tener un impacto significativo en ayudarlo a alcanzar sus metas?
10. ¿Qué tan difícil es en este momento conseguir esa información?
11. ¿Qué oportunidades existen para mejorar dramáticamente el negocio, basado en un mejor acceso a la información?

Información de la Entrevista Realizada para la Obtención de Requerimientos.

1. Descripción de la empresa BICI&MOTO.

La empresa BICI&MOTO, es una empresa privada con fines de lucro, cuya actividad industrial manufacturera está enmarcada en el sector de importaciones de motos y de repuestos de motos.

2. Disponibilidad de información.

- La información no está accesible.
- Existen datos básicos pero deben realizar cálculos sobre ellos si se quiere disponer de agregados, en algunos casos se deben obtener manualmente a partir de varios archivos y sobre otros no existe datos.

3. Información de uso más frecuente.

- Ventas.
- Compras.
- Almacén y Stock.
- Distribución de Productos.
- Otros recursos.

4. Información Necesaria

- Compras.
- Stock.
- Almacén y Ventas.
- Distribución de Productos.
- Sucursal.
- Producto.
- Clientes.
- Proveedores.

5. Frecuencia de uso de información.

- Diaria y en algunos casos semanal.

6. Desempeño del sistema tradicional.

La información disponible se encuentra en archivos Excel y Word, los mismos que son accedidos por los trabajadores de la empresa, pero que no son datos exactos porque en algunas ocasiones las transacciones no son almacenadas. Asimismo no se tiene conocimientos de la cantidad exacta de productos con los que cuenta cada sucursal y almacén de la empresa.

B. MODELO DE ENCUESTA PARA EL PRE-TEST Y POST-TEST.ENTREVISTA EMPRESA BICI&MOTO

Cargo: _____

Fecha: _____

1. ¿Cómo considera usted que es la interfaz del sistema es amigable?
 - a) Muy Bueno
 - b) Bueno
 - c) Regular
 - d) Malo
 - e) Muy Malo
2. ¿Cómo considera que es el grado de complejidad de uso del sistema?
 - a) Muy Bueno
 - b) Bueno
 - c) Regular
 - d) Malo
 - e) Muy Malo
3. ¿Cómo considera que el sistema cumple con los requerimientos necesarios para ser funcional?
 - a) Muy Bueno
 - b) Bueno
 - c) Regular
 - d) Malo
 - e) Muy Malo
4. ¿Cuánto tiempo se demora en registrar, seleccionar, ordenar o eliminar datos en el sistema?
 - a) Menos de 15 minutos
 - b) 15 minutos
 - c) 30 minutos
 - d) 45 minutos
 - e) Más de 45 minutos
5. ¿Cuánto tiempo se demora en consultar o imprimir información del sistema?
 - a) Menos de 15 minutos
 - b) 15 minutos
 - c) 30 minutos
 - d) 45 minutos
 - e) Más de 45 minutos
6. ¿Cuánto tiempo se demora en imprimir boletas o en la consulta de productos de los clientes?
 - a) Menos de 15 minutos
 - b) 15 minutos
 - c) 30 minutos
 - d) 45 minutos
 - e) Más de 45 minutos
7. ¿Cómo considera que es el almacenamiento de información en el sistema?
 - a) Muy Bueno
 - b) Bueno
 - c) Regular
 - d) Malo
 - e) Muy Malo
8. ¿Cómo se encuentra la información disponible en el sistema?
 - a) Muy Bueno
 - b) Bueno
 - c) Regular
 - d) Malo
 - e) Muy Malo
9. ¿El sistema brinda las herramientas necesarias para generar reportes?
 - a) Mucho
 - b) Suficiente
 - c) Regular
 - d) Poco
 - e) Muy poco
10. ¿Cómo considera el grado de satisfacción de uso del sistema?
 - a) Muy Bueno
 - b) Bueno
 - c) Regular
 - d) Malo
 - e) Muy Malo

C. CÓDIGO DEL SERVICIO

Listado de producto.

Interfaz del servicio.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.ServiceModel.Web;
using System.Text;

[ServiceContract]
public interface IListarProductoService
{
    [OperationContract]
    List<ProductoData> GetAllProducto();
}

[DataContract]
public class ProductoData
{
    [DataMember]
    public string Id;
    [DataMember]
    public string Codigo;
    [DataMember]
    public string Producto;
    [DataMember]
    public string Descripcion;
    [DataMember]
    public string Marca;
    [DataMember]
    public string Modelo;
    [DataMember]
    public string Fabricante;
    [DataMember]
    public string Stock;
    [DataMember]
    public string Unidad_Medida;
    [DataMember]
    public string Precio_Compra;
    [DataMember]
    public string Precio_Menor;
    [DataMember]
    public string Precio_Mayor;
    [DataMember]
    public string Estado;
}
```

Código del Servicio.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.ServiceModel.Web;
using System.Text;
using SistemaVentasModelo;

public class ListarProductoService : IListarProductoService
{
    public List<ProductoData> GetAllProducto()
    {
        ComercialEntities contexto;

        using (contexto = new ComercialEntities())
        {
            List<ProductoData> Misproducto = new List<ProductoData>();

            foreach (var p in contexto.productoes)
            {
                ProductoData pdata = new ProductoData();
                pdata.Id = p.id_producto.ToString();
                pdata.Codigo = p.prod_codigo;
                pdata.Producto = p.prod_nombre;
                pdata.Descripcion = p.prod_descripcion;
                pdata.Marca = p.prod_marca;
                pdata.Modelo = p.prod_modelo;
                pdata.Fabricante = p.prod_fabricante;
                pdata.Stock = p.prod_stock.ToString();
                pdata.Unidad_Medida = p.prod_unidad_medida;
                pdata.Precio_Compra = p.prod_preciou_compra.ToString();
                pdata.Precio_Menor = p.prod_preciou_menor.ToString();
                pdata.Precio_Mayor = p.prod_preciou_mayor.ToString();
                pdata.Estado = p.prod_estado;
                Misproducto.Add(pdata);
            }

            return Misproducto;
        }
    }
}

```

Cliente del servicio.

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

```

```

namespace SistemaVentas.Presentacion
{
    public partial class FrmListarProducto : Form
    {
        public FrmListarProducto()
        {
            InitializeComponent();
        }

        private void FrmListarProducto_Load(object sender, EventArgs e)
        {
            ListarProductoServicio.ListarProductoServiceClient
            clienteProducto = new ListarProductoServicio.ListarProductoServiceClient();
            dgvListarProducto.DataSource = clienteProducto.GetAllProducto();
        }

        private void btnSalir_Click(object sender, EventArgs e)
        {
            if (MessageBox.Show("¿Desea Salir?", "Salir",
            MessageBoxButtons.YesNo, MessageBoxIcon.Question,
            MessageBoxDefaultButton.Button1) == System.Windows.Forms.DialogResult.Yes)
            {
                this.Close();
            }
        }

        private void txtBuscar_TextChanged(object sender, EventArgs e)
        {
            try
            {
                DataView dv = new DataView(dt.Copy());
                dv.RowFilter = cmbBuscar.Text + " Like '%" + txtBuscar.Text +
                "%'";

                dgvProducto.DataSource = dv;
                if (dv.Count == 0)
                {
                    lblDatosNoEncontrados.Visible = true;
                }
                else
                {
                    lblDatosNoEncontrados.Visible = false;
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message + ex.StackTrace);
            }
        }
    }
}

```

Ventas.

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using SistemaVentas.Datos;
using SistemaVentas.Entidades;

namespace SistemaVentas.Presentacion
{
    public partial class FrmVenta : Form
    {
        int iVentaId;
        internal int IdProductoSeleccionado = -1;
        double descuento=0;
        double igv = 0;
        double importe = 0;
        double sumasubtotal = 0;
        double sumaigv = 0;
        double sumadescuento = 0;
        private double totalPagar = 0;
        private static FrmVenta _instancia = null;
        private DataTable dtDetalle = new DataTable();

        public FrmVenta()
        {
            InitializeComponent();
        }

        public static FrmVenta GetInstancia()
        {
            if (_instancia == null)
                _instancia = new FrmVenta();
            return _instancia;
        }

        private void FrmVenta_Load(object sender, EventArgs e)
        {
            MostrarGuardarCancelar(true);
            crearDetalle();
        }

        public void crearDetalle()
        {
            if (dtDetalle.Columns.Contains("IdDetalle"))
            {
                dgvDetalleVenta.DataSource = dtDetalle;
                dgvDetalleVenta.Columns["IdDetalle"].Visible = false;
                dgvDetalleVenta.Columns["IdVenta"].Visible = false;
            }
            else
            {

```

```

        dtDetalle = new DataTable();
        dtDetalle.Columns.Add("IdProducto",
System.Type.GetType("System.Int32"));
        dtDetalle.Columns.Add("Producto",
System.Type.GetType("System.String"));
        dtDetalle.Columns.Add("Cantidad",
System.Type.GetType("System.Decimal"));
        dtDetalle.Columns.Add("PrecioUnitario",
System.Type.GetType("System.Decimal"));
        dtDetalle.Columns.Add("Descuento",
System.Type.GetType("System.Decimal"));
        dtDetalle.Columns.Add("Igv", System.Type.GetType("System.Decimal"));
        dtDetalle.Columns.Add("Importe",
System.Type.GetType("System.Decimal"));
        dgvDetalleVenta.DataSource = dtDetalle;
    }
}

private void btnAgregar_Click(object sender, EventArgs e)
{
    bool registrar ;

    if (txtProductoId.Text == "" )
    {
        this.mError("No ha seleccionado aun ningun producto");
    }
    else
    {
        registrar = true;
        foreach (DataRow row in dtDetalle.Rows)
        {
            if (Convert.ToInt32(row["IdProducto"]) ==
Convert.ToInt32(txtProductoId.Text))
            {
                registrar = false;
                this.mError("Ya se encuentra el producto en el detalle");
            }
        }
        if (registrar)
        {
            double subimporte;

            if (txtDescuentoDetalle.Text == "0")
            { descuento =
FDetalleVenta.ObtenerDescuento(Convert.ToDouble(txtCantidad.Value),
Convert.ToDouble(txtPrecioUnitario.Text)); }
            else
            { descuento = Convert.ToDouble(txtDescuentoDetalle.Text);
}

            subimporte = Convert.ToDouble(this.txtPrecioUnitario.Text) *
Convert.ToDouble(txtCantidad.Value);
            if (chkIgv.Checked)
            { igv = FDetalleVenta.ObtenerIgv(Convert.ToDouble(subimporte));
}

            else
            { igv = 0;
}

            importe = (subimporte - descuento + igv);
            sumasubtotal += subimporte;
}
}

```

```

sumadescuento += descuento;
sumaigv += igv;
totalPagar = (sumasubtotal - sumadescuento + sumaigv);

txtDescuento.Text = Convert.ToString(sumadescuento +
Convert.ToDouble(txtDescuento.Text));
txtIgv.Text = Convert.ToString(sumaigv +
Convert.ToDouble(txtIgv.Text));
txtSubtotal.Text = Convert.ToString(sumasubtotal +
Convert.ToDouble(txtSubtotal.Text));
txtTotal.Text = Convert.ToString(totalPagar +
Convert.ToDouble(txtTotal.Text));

if (txtId.Text == "")
{
    DataRow row = dtDetalle.NewRow();
    row["IdProducto"] = txtProductoId.Text;
    row["Producto"] = txtProducto.Text;
    row["Cantidad"] = this.txtCantidad.Value;
    row["PrecioUnitario"] = this.txtPrecioUnitario.Text;
    row["Descuento"] = descuento;
    row["Igv"] = igv;
    row["Importe"] = importe;
    dtDetalle.Rows.Add(row);
}
else // Venta existente
{
    DataRow row = dtDetalle.NewRow();
    row["IdVenta"] = txtId.Text;
    row["IdProducto"] = txtProductoId.Text;
    row["Producto"] = txtProducto.Text;
    row["Cantidad"] = this.txtCantidad.Value;
    row["PrecioUnitario"] = this.txtPrecioUnitario.Text;
    row["Descuento"] = descuento;
    row["Igv"] = igv;
    row["Importe"] = importe;
    dtDetalle.Rows.Add(row);
}

limpiarControlesDetalle();
}
}

private void btnQuitar_Click(object sender, EventArgs e)
{
    try
    {
        if (MessageBox.Show("¿Realmente desea quitar los productos
seleccionados?", "Eliminacion de Productos", MessageBoxButtons.OKCancel,
MessageBoxIcon.Question) == DialogResult.OK)
        {
            foreach (DataRow row in dtDetalle.Rows)
            {
                double QuitarSubtotal = Convert.ToDouble(txtSubtotal.Text);
                double QuitarDescuento =
Convert.ToDouble(txtDescuento.Text);
                double QuitarIgv = Convert.ToDouble(txtIgv.Text);
                double QuitarTotal = Convert.ToDouble(txtTotal.Text);
                QuitarSubtotal = QuitarSubtotal -
(Convert.ToDouble(row["PrecioUnitario"].ToString()) *
(Convert.ToDouble(row["Cantidad"])));
            }
        }
    }
}

```



```

        venta.IdCliente.IdCliente =
Convert.ToInt32(txtIdCliente.Text);
        venta.FechaVenta = txtFecha.Value;
        venta.TipoVenta = cmbTipoVenta.Text;
        venta.TipodocVenta = cmbTipoDocumento.Text;
        venta.NumerodocVenta = txtNumeroDocumento.Text;
        venta.EstadoVenta = cmbEstado.Text;
        venta.SubtotalVenta = Convert.ToDouble(txtSubtotal.Text);
        venta.DescuentoVenta = Convert.ToDouble(txtDescuento.Text);
        venta.IgvVenta = Convert.ToDouble(txtIgv.Text);
        venta.TotalVenta = Convert.ToDouble(txtTotal.Text);
        venta.ObservacionVenta = txtObservacion.Text;

        iVentaId = Convert.ToInt32(FVenta.Insertar(venta));
        foreach (DataRow row in dtDetalle.Rows)
        {
            DetalleVentas detalle = new DetalleVentas();
            detalle.Id_venta.IdVenta = iVentaId;
            detalle.Id_Producto.Id_producto =
Convert.ToInt32(row["IdProducto"].ToString());
            detalle.Cantidad =
Convert.ToDouble(row["Cantidad"].ToString());
            detalle.Precio_unitario =
Convert.ToDouble(row["PrecioUnitario"].ToString());
            detalle.Descuento =
Convert.ToDouble(row["Descuento"].ToString());
            detalle.Igv = Convert.ToDouble(row["Igv"].ToString());
            detalle.Importe =
Convert.ToDouble(row["Importe"].ToString());
            FDetalleVenta.Insertar(detalle);

            FDetalleVenta.DisminuirStock(detalle);
        }
        mOk("Se inserto de manera correcta la venta");
        MostrarGuardarCancelar(false);
        FrmVenta_Load(null, null);
    }
    else
    {
        Venta venta = new Venta();
        venta.IdVenta = Convert.ToInt32(txtId.Text);
        venta.IdSucursal.IdSucursal =
Convert.ToInt32(txtIdSucursal.Text);
        venta.IdCliente.IdCliente =
Convert.ToInt32(txtIdCliente.Text);
        venta.FechaVenta = txtFecha.Value;
        venta.TipoVenta = cmbTipoVenta.Text;
        venta.TipodocVenta = cmbTipoDocumento.Text;
        venta.NumerodocVenta = txtNumeroDocumento.Text;
        venta.EstadoVenta = cmbEstado.Text;
        venta.SubtotalVenta = Convert.ToDouble(txtTotal.Text);
        venta.DescuentoVenta = Convert.ToDouble(txtDescuento.Text);
        venta.IgvVenta = Convert.ToDouble(txtIgv.Text);
        venta.TotalVenta = Convert.ToDouble(txtTotal.Text);
        venta.ObservacionVenta = txtObservacion.Text;
        venta.Idusuario.IdUsuario =
Convert.ToInt32(txtIdUsuario.Text);

        foreach (DataRow row in dtDetalle.Rows)
        {

```

```

        if ((row["IdDetalle"].ToString())=="")
        {
            DetalleVentas detalle = new DetalleVentas();
            detalle.Id_venta.IdVenta =
Convert.ToInt32(row["IdVenta"].ToString());
            detalle.Id_Producto.Id_producto =
Convert.ToInt32(row["IdProducto"].ToString());
            detalle.Cantidad =
Convert.ToDouble(row["Cantidad"].ToString());
            detalle.Precio_unitario =
Convert.ToDouble(row["PrecioUnitario"].ToString());
            detalle.Descuento =
Convert.ToDouble(row["Descuento"].ToString());
            detalle.Igv = Convert.ToDouble(row["Igv"].ToString());
            detalle.Importe =
Convert.ToDouble(row["Importe"].ToString());
            FDetalleVenta.Insertar(detalle);
            FDetalleVenta.DisminuirStock(detalle);
        }
    }
    FVenta.Actualizar(venta);
    mOk("Datos Modificados correctamente");
    this.limpiarControlesDetalle();
    FrmVenta_Load(null, null);
    MostrarGuardarCancelar(false);
    }
}
else
{
    MessageBox.Show("Faltan cargar Datos: \n" + sResultado);
}
}
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message + ex.StackTrace);
}
}

private void btnNuevo_Click(object sender, EventArgs e)
{
    limpiarControlesDetalle();
    limpiarControlesVenta();
    dtDetalle.Reset();
    FrmVenta_Load(null, null);
    MostrarGuardarCancelar(true);
    limpiarControlesDetalle();
    limpiarControlesVenta();
}

private void btnCancelar_Click(object sender, EventArgs e)
{
    dtDetalle.Reset();
    MostrarGuardarCancelar(false);
    DataTable dt = (DataTable)dgvDetalleVenta.DataSource;
    dt.Clear();
    limpiarControlesDetalle();
    limpiarControlesVenta();
}

```

```

    }

    private void btnSalir_Click(object sender, EventArgs e)
    {
        if (MessageBox.Show("¿Desea Salir?", "Salir", MessageBoxButtons.YesNo,
        MessageBoxIcon.Question, MessageBoxDefaultButton.Button1) ==
        System.Windows.Forms.DialogResult.Yes)
        {
            this.Close();
        }
    }

    private void btnEditar_Click(object sender, EventArgs e)
    {
        MostrarGuardarCancelar(true);
    }

    internal void SetModificarVenta(int idventa, string fecha, string estado,
    string tipoventa, string tipodocumento, string numerodocumento,
    string idcliente, string nombrecliente, string paternocliente, string
    subtotal, string descuento, string igv, string total, string observacion,
    string idsucursal, string nombresucursal, string idusuario, string
    nombreusuario, string paternousuario, DataTable DTDetalle)
    {
        txtId.Text = idventa.ToString();
        txtFecha.Text = fecha;
        cmbEstado.Text = estado;
        cmbTipoVenta.Text = tipoventa;
        cmbTipoDocumento.Text = tipodocumento;
        txtNumeroDocumento.Text = numerodocumento;
        txtIdCliente.Text = idcliente;
        txtNombreCliente.Text = nombrecliente;
        txtPaternoCliente.Text = paternocliente;
        txtSubtotal.Text = subtotal;
        txtDescuento.Text = descuento;
        txtIgv.Text = igv;
        txtTotal.Text = total;
        txtObservacion.Text = observacion;
        txtIdSucursal.Text = idsucursal;
        txtNombreSucursal.Text = nombresucursal;
        txtIdUsuario.Text = idusuario;
        txtNombreUsuario.Text = nombreusuario;
        txtPaternoUsuario.Text = paternousuario;
        dgvDetalleVenta.DataSource = DTDetalle;
        dtDetalle = (DataTable)dgvDetalleVenta.DataSource;
    }

    private void mError(string mensaje)
    {
        MessageBox.Show(this, mensaje, "ERROR", MessageBoxButtons.OK,
        MessageBoxIcon.Warning);
    }

    private void mOk(string mensaje)
    {
        MessageBox.Show(this, mensaje, "MENSAJE", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
    }

    private void btnBuscarCliente_Click(object sender, EventArgs e)
    {

```

```

        FrmCliente frmccli = new FrmCliente();
        frmccli.SetFlag("1");
        frmccli.ShowDialog();
    }

    internal void SetCliente(string sIdCliente, string sNombreCliente, string
sPaternoCliente)
    {
        txtIdCliente.Text = sIdCliente;
        txtNombreCliente.Text = sNombreCliente;
        txtPaternoCliente.Text = sPaternoCliente;
    }

    private void btnBuscarSucursal_Click(object sender, EventArgs e)
    {
        FrmSucursal frmsucur = new FrmSucursal();
        frmsucur.SetFlag("1");
        frmsucur.ShowDialog();
    }

    internal void SetSucursal(string IdSucursal, string NombreSucursal)
    {
        txtIdSucursal.Text = IdSucursal;
        txtNombreSucursal.Text = NombreSucursal;
    }

    private void btnBuscarUsuario_Click(object sender, EventArgs e)
    {
        FrmListarUsuario frmusua = new FrmListarUsuario();
        frmusua.SetFlag("1");
        frmusua.ShowDialog();
    }

    internal void SetUsuario(string sIdUsuario, string sNombreUsuario, string
sPaternoUsuario)
    {
        txtIdUsuario.Text = sIdUsuario;
        txtNombreUsuario.Text = sNombreUsuario;
        txtPaternoUsuario.Text = sPaternoUsuario;
    }

    private void btnBuscarProducto_Click(object sender, EventArgs e)
    {
        FrmProducto frmProd = FrmProducto.GetInscance();
        frmProd.SetFlag("1");
        frmProd.ShowDialog();
    }

    internal void SetProducto(string id, string nombre, string stock, string
unidadmedida, string preciounitario)
    {
        txtProductoId.Text = id;
        txtProducto.Text = nombre;
        txtStock.Text = stock;
        txtUnidadMedida.Text = unidadmedida;
        txtPrecioUnitario.Text = preciounitario;
        decimal sub = Convert.ToDecimal(this.txtPrecioUnitario.Text) *
txtCantidad.Value;
    }

    public void MostrarGuardarCancelar(bool b)
    {

```

```

        btnBuscarSucursal.Visible = b;
        btnBuscarUsuario.Visible = b;
        btnBuscarProducto.Visible = b;
        btnBuscarCliente.Visible = b;
        btnNuevo.Visible = !b;
        btnEditar.Visible = !b;
        btnGuardar.Visible = b;
        btnCancelar.Visible = b;
        btnAgregar.Visible = b;
        btnQuitar.Visible = b;
        chkIgv.Enabled = b;

        dgvDetalleVenta.Enabled = b;
        txtFecha.Enabled = b;
        txtObservacion.Enabled = b;

        txtDescuentoDetalle.Enabled = b;
        txtNumeroDocumento.Enabled = b;
        txtCantidad.Enabled = b;
        cmbTipoVenta.Enabled = b;
        cmbEstado.Enabled = b;
        cmbTipoDocumento.Enabled = b;
    }

    public string ValidarDatos()
    {
        string Resultado = "";
        if (txtIdCliente.Text == "") { Resultado = Resultado + " Cliente \n"; }
        if (txtIdUsuario.Text == "") { Resultado = Resultado + " Usuario \n"; }
        if (txtIdSucursal.Text == "") { Resultado = Resultado + " Sucursal \n"; }
    }
    if (cmbTipoVenta.Text == "") { Resultado = Resultado + " Tipo de Venta
\n"; }
    if (cmbEstado.Text == "") { Resultado = Resultado + " Estado de Venta
\n"; }
    if (cmbTipoDocumento.Text == "") { Resultado = Resultado + " Tipo de
Documento \n"; }
    if (txtNumeroDocumento.Text == "") { Resultado = Resultado + " Numero
Documento \n"; }
    return Resultado;
}

private void limpiarControlesDetalle()
{
    this.txtProductoId.Text = string.Empty;
    this.txtProducto.Text = string.Empty;
    this.txtStock.Text = string.Empty;
    this.txtUnidadMedida.Text = string.Empty;
    this.txtPrecioUnitario.Text = string.Empty;
    this.txtDescuentoDetalle.Text = "0";
    this.txtCantidad.Value = 1;
    this.IdProductoSeleccionado = -1;
}

private void limpiarControlesVenta()
{
    txtId.Text = "";
    txtIdSucursal.Text = "";
    txtIdCliente.Text = "";
    txtFecha.Text = "";
    cmbTipoVenta.Text = "";
}

```

```
cmbTipoDocumento.Text = "";  
txtNumeroDocumento.Text = "";  
cmbEstado.Text = "";  
txtSubtotal.Text = "0";  
txtDescuento.Text = "0";  
txtIgv.Text = "0";  
txtTotal.Text = "0";  
txtObservacion.Text = "";  
txtIdUsuario.Text = "";  
txtNombreUsuario.Text = "";  
txtNombreSucursal.Text = "";  
txtNombreCliente.Text = "";  
txtPaternoCliente.Text = "";  
txtPaternoUsuario.Text = "";  
}  
  
private void dgvDetalleVenta_CellClick(object sender,  
DataGridViewCellEventArgs e)  
{  
}  
}  
}
```

