



**UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO**  
**FACULTAD DE CIENCIAS DE LA INGENIERÍA**  
**CARRERA DE INGENIERÍA EN SISTEMAS**

**Tema de la Tesis**

**“SISTEMA DE ADMINISTRACIÓN, COORDINACIÓN Y GESTIÓN DE  
PROYECTOS DE DESARROLLO DE SOFTWARE BAJO MODALIDAD DE  
ÓRDENES DE TRABAJO (ODT) APLICANDO ESTADÍSTICA”**

**Previo a la obtención del Título de:**  
**Ingeniero en Sistemas**

**Autor**

**ORLANDO TOMAS BRIONES INTRIAGO**

**Director**

**Ing. Carlos Caizaguano Chimbo, Msc.**

**Quevedo – Ecuador**  
**2013**

## DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS

Yo, **Orlando Tomas Briones Intrigo**, declaro que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Universidad Técnica Estatal de Quevedo, puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

f. \_\_\_\_\_

**Orlando Tomas Briones Intrigo**

## **CERTIFICACIÓN DEL DIRECTOR DE TESIS**

El suscrito, Ing. Carlos Oswaldo Caizaguano Chimbo, Mcs, Docente de la Universidad Técnica Estatal de Quevedo, certifica que el egresado Orlando Tomas Briones Intriago, realizó la tesis de grado previo a la obtención del título de Ingeniero en Sistemas titulada **“SISTEMA DE ADMINISTRACIÓN, COORDINACIÓN Y GESTIÓN DE PROYECTOS DE DESARROLLO DE SOFTWARE BAJO MODALIDAD DE ÓRDENES DE TRABAJO (ODT) APLICANDO ESTADÍSTICA”**, bajo mi dirección, habiendo cumplido con las disposiciones reglamentarias establecidas para el efecto.

---

Ing. Carlos Oswaldo Caizaguano Chimbo, Mcs  
**DIRECTOR DE TESIS**

**UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO**  
**FACULTAD DE CIENCIAS DE LA INGENIERÍA**  
**CARRERA DE INGENIERÍA EN SISTEMAS**

SISTEMA DE ADMINISTRACIÓN, COORDINACIÓN Y GESTIÓN DE PROYECTOS DE DESARROLLO DE SOFTWARE BAJO MODALIDAD DE ÓRDENES DE TRABAJO (ODT) APLICANDO ESTADÍSTICA.

Presentado al Consejo Directivo como requisito previo a la obtención del título de INGENIERO EN SISTEMAS.

Aprobado:

---

PRESIDENTE DEL TRIBUNAL DE TESIS

---

MIEMBRO DEL TRIBUNAL DE TESIS

---

MIEMBRO DEL TRIBUNAL DE TESIS

QUEVEDO – LOS RÍOS – ECUADOR  
AÑO 2013

## **DEDICATORIA**

El presente trabajo lo dedico mis tres hijos Nasly, María José y Orlando; por ser la razón de mi esfuerzo, dedicación e impulso para seguir adelante en busca de la excelencia y la felicidad.

A mi familia, por haberme apoyado en momentos difíciles y de flaqueza, por sus consejos y ayuda incondicional, especialmente a mi hermana Ángela.

## **AGRADECIMIENTO**

A Dios, por haberme brindado fortaleza y sabiduría, por haberme encaminado en el sendero del bien.

A docentes y amigos, por haber compartido enseñanzas y consejos.

## RESUMEN

El Área de Mantenimiento de Aplicaciones de la Empresa TCS Ecuador presentaba inconvenientes en la gestión y coordinación de atención de ODT's (Orden de Trabajo), al no contar con registros de actividades consistentes e informes estadísticos, que sustenten la toma de decisiones para una correcta dirección del Área por parte de las líneas de supervisión.

El presente trabajo de investigación presenta una solución cuyo objetivo es proveer una herramienta a todos los niveles de la organización del Área de Mantenimiento de Aplicaciones de la empresa TCS Ecuador, capaz de proveer un medio de administración, gestión y coordinación de actividades de desarrollo de proyectos de software bajo la modalidad de órdenes de trabajo (ODT).

A través del desarrollo e implementación de un sistema informático (**SAGMA**) que integre las necesidades del área, con una interfaz amigable y de fácil manejo, se mejora la gestión permitiendo una mejor toma de decisiones.

Además, **SAGMA** apoya la gestión diaria de los desarrolladores ofreciendo una base de conocimientos de los elementos de software (Servidores, Bases de Datos Aplicaciones) y ODT's históricas para identificar soluciones más efectivas en el menor tiempo posible.

Una vez implementado el sistema informático **SAGMA** y con soporte de las pruebas correspondientes, se demostró al aumento de la eficiencia en la gestión técnica y administrativa del Área de Mantenimiento de Aplicaciones.

## **SUMMARY**

The Maintenance Area Business Applications TCS Ecuador had disadvantages in the management and care coordination ODT's (Work Order), by not having records of consistent activities and statistical reports to support decision making for Area right direction by lines of supervision.

This research paper presents a solution aimed at providing a tool at all levels of the organization Area Application Maintenance Company TCS Ecuador, capable of providing a means of administration, management and coordination of project development activities software in the form of work orders (ODT).

Through the development and implementation of a computer system (SAGMA) that integrates the needs of the area, with a friendly interface and easy to use, management is improved allowing for better decision making.

Furthermore, SAGMA supports the daily management of developers offering a knowledge base of software elements (Servers, Databases Applications) and ODT's to identify historical most effective solutions in the shortest possible time.

Once implemented SAGMA computer system and supported by evidence, it was shown to increase the efficiency of the technical and administrative management of Application Maintenance Area.



# ÍNDICE DE CONTENIDO

<b>CAPÍTULO I</b>	16
<b>MARCO CONTEXTUAL DE LA INVESTIGACIÓN</b>	16
1.1. INTRODUCCIÓN	17
1.2. SITUACIÓN ACTUAL DE LA PROBLEMÁTICA	18
1.2.1. Análisis del Problema	18
1.2.2. Formulación	19
1.2.3. Sistematización	19
1.3. JUSTIFICACIÓN	19
1.4. OBJETIVOS	20
1.4.1. Objetivo General	20
1.4.2. Objetivos Específicos	20
1.5. RESULTADOS ESPERADOS	21
1.6. HIPÓTESIS	21
1.6.1. Planteamiento	21
1.6.2. Matriz de Operacionalización	22
<b>CAPÍTULO II</b>	23
<b>MARCO TEÓRICO DE LA INVESTIGACIÓN</b>	23
2.1. FUNDAMENTACIÓN CONCEPTUAL	24
2.1.1. Tata Consultancy Services (TCS, por sus siglas en inglés)	24
2.1.2. Orden de Trabajo (ODT)	24
2.1.3. Acuerdos de Nivel de Servicio (Service Level Agreement – SLA)	25
2.1.4. Mantenimiento de Aplicaciones	26
2.1.5. Áreas de Apoyo	26
2.1.6. Ambiente de Desarrollo	27
2.1.7. Ambiente de Test	27
2.1.8. Ambiente de Producción	28
2.2. FUNDAMENTACIÓN TEÓRICA	28
2.2.1. Método Cuasi-Experimental	28
2.2.1.1. Generalidades	28
2.2.1.2. Conceptos	28
2.2.2. Técnicas de Investigación	30
2.2.2.1. Encuesta	30
2.2.2.1.1. Etapas en una Investigación con Encuestas	32
2.2.2.2. Entrevista	39
2.2.3. POBLACIÓN Y MUESTRA	40
2.2.3.1. Conceptos Preliminares	40
2.2.3.2. Individuo	40

2.2.3.3.	Universo.....	40
2.2.3.4.	Población.....	41
2.2.3.5.	Muestra.....	41
2.2.4.	Software.....	43
2.2.4.1.	Tipos de software.....	44
2.2.4.2.	Aspectos legales sobre el software.....	45
2.2.5.	Sistema Informático.....	46
2.2.6.	Sistema Operativo.....	46
2.2.7.	Base de Conocimientos.....	46
2.2.8.	Base de Datos.....	47
2.2.9.	Sistema de Administración de Bases de Datos (DBMS, por sus siglas en inglés)	47
2.2.9.1.	Función y Ventajas del DBMS.....	48
2.2.9.2.	Tipos de Bases de Datos.....	51
2.2.10.	Lenguaje de Consulta Estructurada (SQL, por sus siglas en inglés).....	51
2.2.10.1.	Partes de SQL.....	52
2.2.11.	Microsoft SQL Server.....	53
2.2.12.	Entorno de Desarrollo Integrado (IDE, por sus siglas en inglés).....	53
2.2.13.	Lenguaje de Programación.....	54
2.2.14.	Framework.....	54
2.2.15.	Microsoft Visual Studio .NET.....	54
2.2.16.	Programación Orientada a Objetos (OOP, por sus siglas en inglés).....	55
2.2.16.1.	Principios de la POO.....	56
2.2.16.2.	Ventajas / Desventajas POO.....	57
2.2.16.3.	Paso de Mensajes.....	59
2.2.16.4.	Clases.....	60
2.2.16.5.	Herencia.....	61
2.2.16.6.	Polimorfismo.....	62
2.2.16.7.	Sobrecarga.....	63
2.2.16.8.	Objeto.....	63
2.2.17.	Metodología de Desarrollo del Software.....	65
2.2.17.1.	Metodologías Ágiles - RAD.....	65
2.2.17.2.	Etapas de la metodología RAD.....	65
2.2.18.	Lenguaje Unificado de Modelamiento (UML, por sus siglas en inglés).....	66
2.2.18.1.	Conceptos de Modelado de Software.....	67
2.3.	MARCO REFERENCIAL.....	78
CAPÍTULO III.....		80
METODOLOGÍA DE LA INVESTIGACIÓN.....		80
3.1.	MATERIALES Y MÉTODOS.....	81
3.1.1.	Equipos y Materiales.....	81

3.2.	<b>MÉTODOS</b>	83
3.2.1.	Método Inductivo	83
3.2.2.	Método Deductivo	84
3.3.	<b>TIPOS DE INVESTIGACIÓN</b>	84
3.3.1.	Metodología Cuasi-Experimental	84
3.4.	<b>TÉCNICAS DE INVESTIGACIÓN</b>	85
3.4.1.	Población	85
3.4.2.	Muestra	86
3.4.3.	Encuestas	87
3.4.3.1.	Resultados de Encuestas aplicadas al personal con cargo de Analista / Desarrollador	87
3.4.3.3.	Resultados de Encuestas aplicadas al personal con cargo de Facilitador / Gerente de Área	97
3.4.4.	Entrevista	102
3.4.4.1.	Entrevista realizada al Ing. Pablo Córdova (Desarrollador Sénior, 6 años en el Área de Mantenimiento de Aplicaciones) y al Ing. Danilo Oña (Desarrollador Junior, 45 días en el Área de Mantenimiento de Aplicaciones)	103
3.4.4.2.	Entrevista realizada a Coordinadores del Área de Mantenimiento de Aplicaciones con 3 años de experiencia en el cargo	104
3.4.4.3.	Entrevista realizada a Facilitadores del Área de Mantenimiento de Aplicaciones con 4 años de experiencia en el cargo	104
	<b>CAPÍTULO IV</b>	105
	<b>DESARROLLO DEL SISTEMA</b>	105
4.1.	<b>FACTIBILIDAD</b>	106
4.1.1.	Factibilidad Técnica	106
4.1.2.	Factibilidad Operacional	107
4.1.3.	Factibilidad Económica	107
4.2.	<b>METODOLOGÍA DE DESARROLLO DE SOFTWARE</b>	107
4.2.1.	Metodologías Ágiles – RAD	107
4.2.2.	Análisis	108
4.2.2.1.	Especificación de requerimientos	108
4.2.2.1.1.	Propósito	108
4.2.2.1.2.	Alcance	108
4.2.2.1.3.	Referencias	110
4.2.2.1.4.	Descripción General del Sistema Informático SAGMA	110
4.2.2.1.5.	Requerimientos específicos funcionales	111
4.2.2.1.6.	Casos de Uso	114
4.2.2.1.7.	Requerimientos no Funcionales	126
4.2.2.1.8.	Diagramas de Casos de Uso	127
4.2.2.1.9.	Diagramas de Secuencia	130
4.2.2.1.10.	Diagrama de Clases	134
4.2.2.1.11.	Diagrama de Capas	136

4.2.3.	Diseño.....	137
4.2.3.1.	Diseño de Interfaz de Usuario (GUI) .....	137
4.2.3.2.	Diseño de Base de Datos .....	154
4.2.3.2.1.	Modelo Entidad Relación.....	154
4.2.3.2.2.	Diseño Global .....	155
4.2.3.2.3.	Diccionario de Datos .....	159
4.2.4.	Desarrollo .....	171
4.2.4.1.	Base de Datos .....	172
4.2.4.2.	Clases .....	177
4.2.4.3.	Formulario .....	181
4.2.5.	Implantación .....	186
CAPÍTULO V .....		187
RESULTADOS Y DISCUSIÓN .....		187
5.1.	PRUEBAS Y RESULTADOS .....	188
5.1.1.	Pruebas de Ejecución.....	188
5.1.2.	Hardware Utilizado para las Pruebas .....	195
5.2.	RESULTADO DE LA COMPROBACIÓN DE HIPÓTESIS.....	196
5.2.1.	Variable Independiente .....	196
5.2.1.1.	Software .....	196
5.2.2.	Variable Dependiente .....	196
5.2.2.1.	Eficiencia .....	196
5.2.2.2.	Eficacia.....	196
5.2.2.3.	Económico .....	197
5.3.	ANÁLISIS DE RESULTADOS .....	197
5.4.	CONCLUSIONES.....	200
5.5.	RECOMENDACIONES.....	201
CAPÍTULO VI .....		203
LITERATURA CITADA .....		203
6.1.	BIBLIOGRAFÍA .....	204
6.2.	ENLACES WEB .....	205

## ÍNDICE DE TABLAS

<b>TABLA 1:</b> Equipos, Materiales y Recursos Humanos	75
<b>TABLA 2:</b> Personal que labora en el Área de Mantenimiento de Aplicaciones	85
<b>TABLA 3:</b> ¿Lleva un registro de las actividades diarias?	87
<b>TABLA 4:</b> ¿En que lleva los registros de actividades diarias?	88
<b>TABLA 5:</b> ¿Considera acertada las asignaciones de ODT's por parte de los Coordinadores?	89
<b>TABLA 6:</b> ¿Conoce los sistemas, bases de datos, servidores a los cuales el Área de Mantenimiento de Aplicaciones da soporte?	90
<b>TABLA 7:</b> ¿Cree que es necesario contar con un registro histórico de soluciones?	91
<b>TABLA 8:</b> ¿Cree usted que la implementación de un sistema informático especializado sería útil para mejorar los procesos del Área de Mantenimiento de Aplicaciones?	92
<b>TABLA 9:</b> ¿Conoce la capacidad de atención y carga de trabajo de sus subordinados?	94
<b>TABLA 10:</b> ¿Lleva un registro sobre el incumplimiento en el tiempo de entrega de las soluciones por parte de sus subordinados?	95
<b>TABLA 11:</b> ¿Conoce las actividades en las que sus subordinados distribuyen el tiempo?	96
<b>TABLA 12:</b> ¿Cree usted que la implementación de un sistema informático especializado sería útil para mejorar los procesos del Área de Mantenimiento de Aplicaciones?	97
<b>TABLA 13:</b> ¿Cree usted que el Área de Mantenimiento de Aplicaciones tiene la capacidad suficiente de atención?	98
<b>TABLA 14:</b> ¿Conoce el tiempo extra de trabajo del área en general?	99
<b>TABLA 15:</b> ¿Considera necesario que el Área de Mantenimiento de Aplicaciones cuente con un registro de usuarios por cliente, en el que conste información relevante, en caso de requerir comunicación directa?	100
<b>TABLA 16:</b> ¿Cree necesario que el área de Mantenimiento de Aplicaciones cuente con información estadística para determinar el rendimiento general del área?	101
<b>TABLA 17:</b> ¿Cree usted que la implementación de un sistema informático especializado sería útil para mejorar los procesos del Área de Mantenimiento de Aplicaciones?	102
<b>Tabla 18:</b> Cuadro comparativo de uso de <b>SAGMA</b>	198
<b>Tabla 19:</b> Tiempos de respuesta en diferentes procesos	200

## ÍNDICE DE FIGURAS

Figura 1. Interacción entre el usuario final y la Base de Datos.....	48
Figura 2. Ilustración de Encapsulamiento.....	56
Figura 3. Ejemplo de Herencia.....	61
Figura 4. Ilustración de Polimorfismo.....	63
Figura 5. Ilustración de Objeto.....	63
Figura 6. Ilustración de un proceso de análisis basado en UML.....	67
Figura 7. Ilustración de Diagrama de Clases.....	70
Figura 8. Ilustración de Diagrama de Objetos.....	70
Figura 9. Ilustración de Diagrama de Componentes.....	71
Figura 10. Ilustración de Diagrama de Despliegue.....	71
Figura 11. Ilustración de Diagrama de Paquetes.....	72
Figura 12. Ilustración de Diagrama de Estructura Compuesta.....	72
Figura 13. Ilustración de Diagrama de Casos de Uso.....	73
Figura 14. Ilustración de Diagrama de estados.....	73
Figura 15. Ilustración de Diagrama de Actividades.....	74
Figura 16. Ilustración de Diagrama de Secuencia.....	75
Figura 17. Ilustración de Diagrama de Comunicación (Colaboración).....	76
Figura 18. Ilustración de Diagrama de Tiempo.....	77
Figura 19. Ilustración de Diagrama de Revisión de Interacciones.....	78

## ÍNDICE DE ANEXOS

Diagnóstico	208
Árbol de Problemas	209
Formulación del Problema	210
Matriz de Relación	211
Sistematización	212
Encuestas	213
Tabla T-Student	219
Certificación de revisión Técnica	221

## **CAPÍTULO I**

### **MARCO CONTEXTUAL DE LA INVESTIGACIÓN**



## **1.1. INTRODUCCIÓN**

Las empresas de desarrollo de software están en constante aplicación de metodologías y herramientas, con el fin de optimizar la gestión del tiempo empleado por los desarrolladores, en todo el ciclo de vida de sus proyectos.

Mediante la aplicación de metodologías se busca aumentar la eficiencia y optimización en todos los niveles del proceso de desarrollo de software para el cumplimiento de metas y objetivos previamente definidos; el uso de herramientas de gestión de proyectos se emplea para la coordinación y distribución de las actividades de los desarrolladores.

El desafío de las líneas de supervisión, está en la correcta asignación de tareas, para encaminar al desarrollador al cumplimiento de los objetivos en el menor tiempo posible, y la entrega de un producto de calidad, para lo cual se deben basar en el conocimiento de la situación actual y futura de sus subordinados.

El presente proyecto de software tiene por objetivo proveer una herramienta a todos los niveles de la organización del Área de Mantenimiento de Aplicaciones de la empresa TCS Ecuador, capaz de proveer un medio de administración, gestión y coordinación de actividades de desarrollo de proyectos de software bajo la modalidad de órdenes de trabajo (ODT).

El proyecto contribuirá a mejorar la eficiencia del Área de Mantenimiento de Aplicaciones, fortaleciendo las métricas de medición de cumplimiento de la empresa, aumentando la capacidad de atención y solución de requerimientos, a su vez, eliminar la asignación incoherente de órdenes de trabajo por parte de los coordinadores, y el incumplimiento de entrega de soluciones de los desarrolladores en el tiempo previamente estimado.

## **1.2. SITUACIÓN ACTUAL DE LA PROBLEMÁTICA**

### **1.2.1. Análisis del Problema**

El Área de Mantenimiento de Aplicaciones de la empresa TCS Ecuador, presenta una serie de problemas relacionados a la gestión de ODT (Orden de Trabajo), los problemas derivados de la incorrecta asignación a desarrolladores por parte de los coordinadores, ocasionan incumplimiento en los tiempos de entrega estimados a los usuarios beneficiarios de la ODT.

La incorrecta y deficiente administración del tiempo de los desarrolladores ocasiona acumulación de ODT sin resolución oportuna, malestar en los usuarios beneficiarios, y posteriores reclamos formales e incumplimiento del SLA (Acuerdo de Nivel de Servicio). El desconocimiento del tiempo promedio de atención a una ODT por parte de otras áreas impide la planificación de atención de otras ODT.

La falta de una base de conocimientos especializada da origen al desconocimiento del Ecosistema Informático (Aplicaciones, Componentes, Bases de Datos, Servidores) por parte de los desarrolladores, aumenta el tiempo de resolución y en ocasiones genera versionamientos con errores en producción.

Al no contar con una fuente de datos almacenados y relacionados coherentemente, no se puede obtener estadística referente al rendimiento de los desarrolladores, capacidad real de atención del Área de Mantenimiento de Aplicaciones, identificación de complejidad de la ODT, interacción y tiempos de respuesta con las áreas de apoyo (Librerías, Control de Cambios, Centro de Cómputo, QA, Base de Datos, Plataforma Centralizada, Plataforma Distribuida, Delivery).

El incumplimiento del SLA genera sanciones económicas y efectos negativos en la encuesta de satisfacción anual realizada a los usuarios.

### **1.2.2. Formulación**

¿Cómo mejorar la asignación de ODT a los desarrolladores de software del Área de Mantenimiento de Aplicaciones de la empresa TCS Ecuador?

### **1.2.3. Sistematización**

- ¿Cuáles son los elementos de la Base de Conocimientos para la correcta asignación de la ODT?
- ¿Cómo lograr obtener un mayor rendimiento de los desarrolladores en las actividades asignadas?
- ¿Cómo obtener una estimación de tiempo coherente para el cumplimiento y gestión eficiente de la ODT?

## **1.3. JUSTIFICACIÓN**

El Área de Mantenimiento de Aplicaciones de la empresa TCS Ecuador, dispone del sistema informático para la asignación de ODT llamado UNICENTER. El sistema informático tiene por objetivo apoyar la gestión diaria de desarrolladores y líneas de supervisión; lamentablemente no cuenta con los elementos necesarios para el soporte, carece de reportes que apoyen la gestión y control administrativa; estos problemas inciden directamente en el cumplimiento efectivo de las actividades diarias de todo el personal del área.

Ante esta situación adversa es necesario el desarrollo de un sistema informático que minimice significativamente los efectos derivados de los problemas, evitando el incumplimiento de compromisos con los usuarios.

El presente proyecto tiene como objetivo el desarrollo de un sistema informático denominado “**SAGMA** (Sistema de Administración y Gestión de Mantenimiento de Aplicaciones)” para optimizar la gestión de coordinadores en la asignación de ODT a los desarrolladores, apoyo a las líneas de supervisión como facilitadores y gerentes en el seguimiento de las soluciones de los requerimientos, mediante informes a nivel de detalle e informes estadísticos. El proyecto contará con una Base de Conocimientos Especializada con los elementos necesarios para dar respaldo y soporte a la gestión del desarrollador y las líneas de supervisión.

La importancia de la implementación del proyecto radica en la necesidad de aumentar la productividad y eficacia del Área de Mantenimiento de Aplicaciones, y así, lograr el cumplimiento de las metas y objetivos planteados en el plan estratégico.

## **1.4. OBJETIVOS**

### **1.4.1. Objetivo General**

Desarrollar un sistema de asignación de ODT en el Área de Mantenimiento de Aplicaciones de la empresa TCS Ecuador, que permita administrar de forma eficiente y segura los procesos de desarrollo y mantenimiento de software.

### **1.4.2. Objetivos Específicos**

- Determinar los elementos de la Base de Conocimientos necesarios para proveer información relacionada a la ODT.
- Proveer de reportes estadísticos a las líneas de supervisión mediante el registro de actividades de los desarrolladores, para ejercer un control y seguimiento efectivo.

- Proporcionar datos estadísticos relacionados a la complejidad de la ODT y rendimiento de los desarrolladores para una correcta estimación de tiempos.

## **1.5. RESULTADOS ESPERADOS**

- Las ODT se asignan de acuerdo a la capacidad del desarrollador.
- Se utiliza una Base de Conocimientos Especializada con los elementos necesarios para el desarrollo de una solución.
- El registro categorizado de actividades de los desarrolladores genera reportes para control y seguimiento de líneas de supervisión sobre su rendimiento.
- Se cuenta con estadística para toma de decisiones de asignación de ODT basado en la complejidad y el rendimiento de los desarrolladores.

## **1.6. HIPÓTESIS**

### **1.6.1. Planteamiento**

Un Sistema Informático basado en Base de Conocimientos Especializada que cuente con los componentes del Ecosistema Informático permitirá la correcta asignación de recursos para resolver la ODT.

### 1.6.2. Matriz de Operacionalización

VARIABLE	CONCEPTO	DIMENSIÓN	INDICADOR
<b><u>INDEPENDIENTE</u></b> Sistema Informático basado en Bases de Conocimientos.	Alberga las descripciones de los conceptos de conocimiento y sus relaciones, los criterios de búsqueda y las reglas de deducción que dependerán del tipo de problemas a resolver y el área de conocimiento concreta: medicina, prospección minera, planificación financiera, programación de la producción, entre las más comunes.	<b>SOFTWARE</b>	<ul style="list-style-type: none"> <li>• Reducción de tiempo de resolución de ODT.</li> <li>• Reducción de errores en versionamientos.</li> <li>• Aumento de productividad de desarrolladores.</li> </ul>
<b><u>DEPENDIENTE</u></b> Asignación eficaz de ODT.	Una ODT (Orden de Trabajo) es un requerimiento descriptivo de una petición para llevar a cabo un trabajo determinado y ejecutado por una persona capaz.	<b>EFICIENCIA</b>	<ul style="list-style-type: none"> <li>• Aumento de número de ODT resueltas por desarrollador.</li> </ul>
		<b>EFICACIA</b>	<ul style="list-style-type: none"> <li>• Porcentaje de cumplimiento de entrega de soluciones.</li> </ul>
		<b>ECONÓMICO</b>	<ul style="list-style-type: none"> <li>• Reducción de multas por incumplimiento de SLA.</li> <li>• Reducción de costos por concepto de horas extras.</li> </ul>

## **CAPÍTULO II**

### **MARCO TEÓRICO DE LA INVESTIGACIÓN**

## **2.1. FUNDAMENTACIÓN CONCEPTUAL**

### **2.1.1. Tata Consultancy Services (TCS, por sus siglas en inglés)**

Tata Consultancy Services es una empresa de tecnología parte del conglomerado multinacional TATA con sede en la India, brinda servicios BPO (Subcontratación de Procesos de Negocios, del inglés: Business Process Outsourcing) en Ecuador desde el año 2007, la empresa está ubicada en la ciudad de Quito perteneciente a la provincia de Pichincha (Ecuador), su principal actividad es el desarrollo y mantenimiento de aplicaciones para el sector financiero y gubernamental.

El área de tecnología de software está encargada del desarrollo, implementación y mantenimiento de todo el Ecosistema Informático.

### **2.1.2. Orden de Trabajo (ODT)<sup>1</sup>**

Representa un ticket generado para atención de un requerimiento de soporte tecnológico solicitado por un usuario beneficiario y asignado a un desarrollador. En una ODT el usuario especifica el tipo de requerimiento, la prioridad y la descripción detallada.

Una ODT de tipo Incidente representa un requerimiento de revisión inmediato de un error en una determinada aplicación; la ODT de tipo Solicitud representa un requerimiento de cambio de funcionalidad o adición de un nuevo proceso en una aplicación.

---

<sup>1</sup>Facilitadores (2013). Área de Mantenimiento de Aplicaciones. ODT.



### **2.1.3. Acuerdos de Nivel de Servicio (Service Level Agreement – SLA)<sup>2</sup>**

Los Acuerdos de Servicio son parte fundamental de la Gestión de Servicios de TI, permiten determinar con claridad las características del servicio prestado, es importante el conocimiento de los acuerdos de servicio para una buena relación entre el cliente y el prestador de servicios. Los acuerdos de servicio son parte de la contratación de prestación de servicios de TI y deben ser documentados a conformidad de las partes.

Los SLA tienen como objetivos:

- Identificar y definir las necesidades del cliente.
- Simplificar temas que requieran tratamiento complejo.
- Reducir las zonas de conflicto.
- Eliminar expectativas poco realistas.

Un documento de Acuerdo de Servicios, puede abarcar una amplia gama de tópicos, entre los principales están los siguientes:

- Definición de los Servicios a ser entregados.
- Esquema de cumplimiento basado en parámetros y tiempos de atención.
- Seguimiento y presentación de informes.
- Deberes y responsabilidades del cliente.

---

<sup>2</sup> Garzón Gavilanes, Pablo Oswaldo (2009). Tesis de Grado “Datawarehouse de Mesa de Servicios para el Área de Mantenimiento de Aplicaciones”. Definición de SLA. Pág. 11.

- Control de casos excepcionales.

#### **2.1.4. Mantenimiento de Aplicaciones**

Se encarga del mantenimiento y cambios al ecosistema de software cuyo tiempo de desarrollo no tome más de 40 horas. Los requerimientos son atendidos solo con el respaldo de una ODT.

#### **2.1.5. Áreas de Apoyo**

- **Delivery**

Se encargada del análisis, documentación, desarrollo de proyectos nuevos o implementación de cambios considerables de una aplicación en particular.

- **Librarian**

Se encarga de la custodia, entrega de fuentes, control de versiones, implementación de fuentes nuevas y modificadas de todo el ecosistema de software en los ambientes de desarrollo, test y producción.

- **Base de Datos**

Se encarga de la implementación de afectación a objetos de base de datos nuevos o modificados, monitoreo de las bases de datos de producción.

- **Control de Cambios**

Se encarga de la calendarización para la implementación de fuentes nuevos o modificados por parte de Librarian y/o Base de Datos y la ejecución de procesos batch por parte de Centro de Cómputo.

- **Plataforma Centralizada**

Se encarga del monitoreo de la granja de servidores Unix y AS400 y sus configuraciones.

- **Plataforma Distribuida**

Se encarga del monitoreo de la granja de servidores Windows y sus configuraciones.

- **QA (Control de Calidad)**

Área encargada de asegurar la calidad de los versionamientos de aplicaciones nuevas y cambios de funcionalidades a fuentes existentes.

- **Centro de Cómputo**

Área encargada de la ejecución de procesos batch y monitoreo del ecosistema de software en ambiente de producción.

#### **2.1.6. Ambiente de Desarrollo**

Área donde se desarrolla software, conformado por los equipos de cómputo de los desarrolladores, servidores de desarrollo como por ejemplo los de base de datos, o cualquier otro servicio.

#### **2.1.7. Ambiente de Test**

Es el área donde se prueban las aplicaciones por parte del área de QA, en este ambiente se debe detectar los errores o incidencias funcionales y no funcionales, de manera que el software pueda llegar a producción cumpliendo con todos los requerimientos funcionales.

### **2.1.8. Ambiente de Producción**

Es el ambiente final donde las aplicaciones son publicadas y monitoreadas para un correcto funcionamiento y uso de usuarios.

## **2.2. FUNDAMENTACIÓN TEÓRICA**

### **2.2.1. Método Cuasi-Experimental<sup>3</sup>**

#### **2.2.1.1. Generalidades**

El diseño más habitual es el estudio antes-después o pre-post. En la investigación Cuasi-Experimental se identifica la variable independiente mediante la letra X. La variable dependiente es designada con la letra O. Su notación es: O1 X O2.

Dónde:

X = Sistema Informático basado en Bases de Conocimientos.

O1 = Medición antes del experimento (Con los sistemas actuales).

O2 = Medición después del experimento (Con el uso del sistema desarrollado).

En este diseño se verifica una observación (O1) antes de implantar la variable independiente y otra después de aplicar la aplicación (O2). Las observaciones se consiguen a través de la aplicación de una prueba u observación.

#### **2.2.1.2. Conceptos**

En ocasiones, a pesar de que la VI (variable independiente) sea una condición manipulada, el investigador no puede establecer los mínimos controles en lo

---

<sup>3</sup> Diego Redolar Ripoll. (2010). Fundamentos de psicobiología. Editorial UOC. Metodología Cuasi-Experimental. Pág. 57.

que se refiere a la utilización de un procedimiento probadamente aleatorio: asignación aleatoria sin restricciones, bloqueo o contrabalanceo. En estos casos se incumple una de las características de la metodología experimental: el uso de la aleatorización.

A pesar de que, en la metodología cuasi-experimental, existe una imposibilidad de poder asignar aleatoriamente los sujetos a las diferentes condiciones experimentales, su objetivo es estudiar relaciones causales. No obstante, en la metodología cuasi-experimental, se estudian relaciones causales con técnicas que no dan un 100% de seguridad sobre la atribución causal.

Normalmente, la metodología cuasi-experimental se suele usar en el ámbito más aplicado, cuando interesa evaluar el efecto de una intervención determinada.

Si entre un tratamiento determinado y una respuesta estudiada existe una relación funcional (validez de conclusión estadística), cabe preguntarse si el tratamiento ha sido la causa de las variaciones encontradas en la respuesta, o bien si estas variaciones se hubieran podido obtener en ausencia del tratamiento.

Existe una serie de amenazas en el diseño cuasi-experimental que nos dificultan saber si el tratamiento ha sido la causa de los cambios encontrados en una respuesta (VD, variable dependiente), o bien si estos se hubieran podido obtener en ausencia de dicho tratamiento. Se trata de amenazas contra la validez interna:

**1) Maduración.** Se trata de cambios producidos en el propio sujeto entre la media pre y la medida post que producen modificaciones de los valores de la VD.

**2) Historia.** Se trata de cambios producidos en el contexto o en el entorno del sujeto entre la media pre y la medida post que producen modificaciones de los valores de la VD.

**3) Selección.** Se trata de diferencias iniciales entre los grupos debido a características de los sujetos no controlados por el investigador. De esta forma, los diferentes grupos de tratamiento (niveles de la VI) se encuentran desequilibrados. Puede ser la amenaza más grave contra la validez interna.

**4) Administración de Pruebas.** Se trata de cambio producidos por la administración repetida de una prueba.

**5) Instrumentación.** Se trata de cambios producidos en la calibración del instrumento de medida a los largo del proceso de la investigación.

**6) Regresión Estadística a la Media.** Se trata de una tendencia de los datos extremos a acercarse a su medida cuando se repiten las medidas. Esta amenaza disminuye si tenemos diversas observaciones pre y post.

**7) Pérdida no Aleatoria de Sujetos.** Se trata de una mortalidad selectiva. Se produce por el abandonamiento no aleatorio de los sujetos de la investigación. Dicho abandonamiento se asocia a uno de los niveles de la VI.

## **2.2.2. Técnicas de Investigación**

### **2.2.2.1. Encuesta<sup>4</sup>**

Se puede definir la encuesta como una técnica de obtención de datos de interés sociológico mediante la interrogación a los miembros de la sociedad. Es, al mismo tiempo, un tipo de investigación y una técnica de recogida de

---

<sup>4</sup> Manuel Palencia-Lefler. (2008). 90 Técnicas de Relaciones Públicas: Manual de Comunicación Corporativa. Profit Editorial. Encuesta. Pág. 109.

datos, denominada también «sondeo de opinión», o «pollster» (en países anglosajones), o investigación demoscópica (en Alemania).

Sus características fundamentales son:

- a) Que se trata de un procedimiento de interrogación;
- b) Tiene por objeto un gran colectivo;
- c) Generalmente usa una muestra representativa;
- d) Su finalidad es obtener información cuantitativa; y
- e) Puede abarcar aspectos objetivos o subjetivos de la colectividad.

El instrumento básico de observación es el cuestionario.

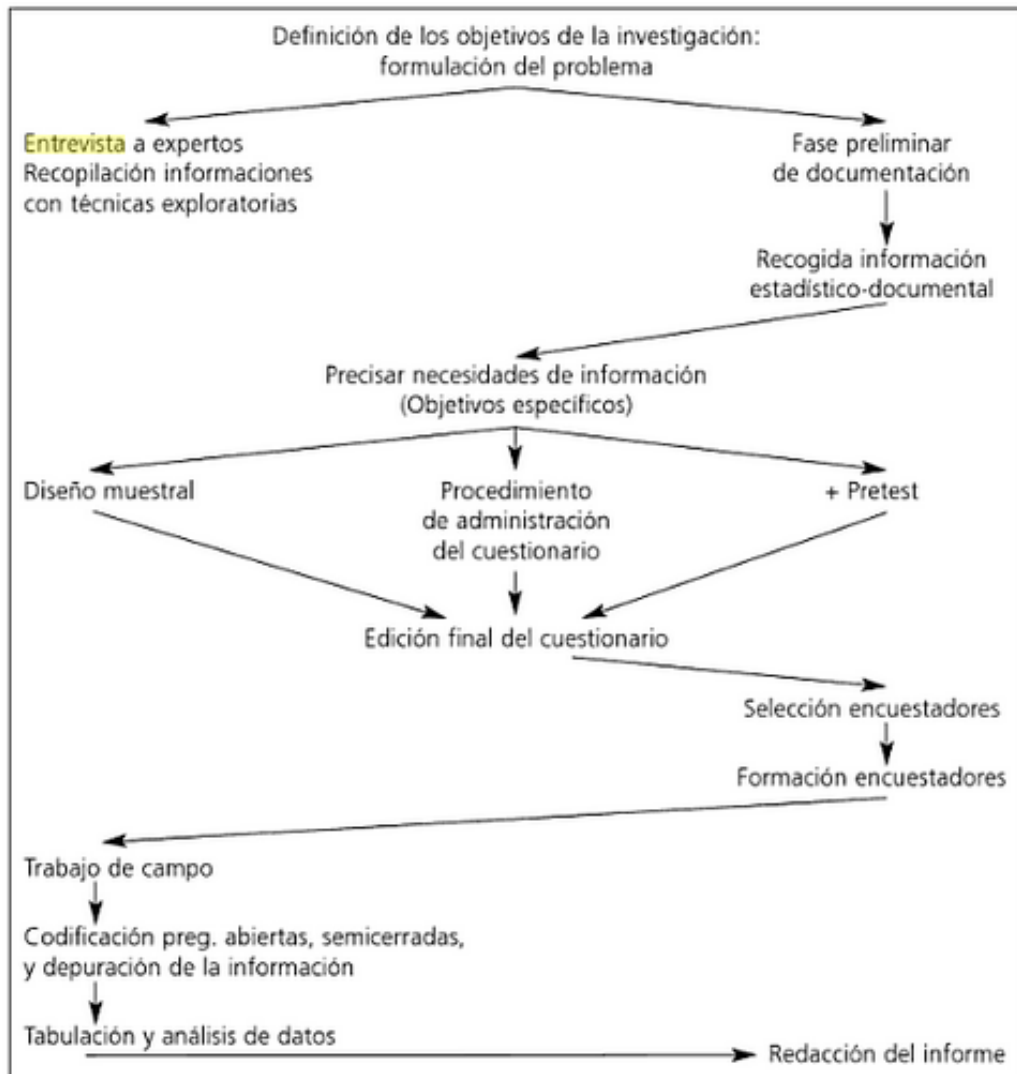
La encuesta es la Técnica de Investigación Social más utilizada y popular a lo largo de los últimos 50 años. Aunque, debe reconocerse que su uso ha sido abusivo. Por otra parte, ha demostrado grandes limitaciones.

La primera: los temas a preguntar tienen que ser de actualidad y afectar a muchas personas para conseguir un elevado número de respuestas. La segunda y definitiva: es una técnica cara.

Gran parte del éxito de la técnica está en el cuestionario, cuya finalidad es obtener información sincera.

### 2.2.2.1.1. Etapas en una Investigación con Encuestas<sup>5</sup>

**Figura 20. Etapas en una Investigación con Encuestas**



#### 1) Delimitación de objetivos y formulación del problema de investigación

La primera fase en el diseño de una investigación mediante encuesta comienza con una delimitación clara de las cuestiones o materias a investigar, elaborando un listado de los temas sobre los que se quiere obtener información. El planteamiento del problema debe ser claro y unívoco, exponiendo claramente la naturaleza del problema.

<sup>5</sup> Vidal Díaz de Rada. (2009). Análisis de datos de encuestas: Desarrollo de una investigación completa utilizando SPSS. Editorial UOC. Etapas en una Investigación con Encuestas. Pág. 19.



Una de las mejores formas de llevar a cabo una delimitación es la revisión y análisis de las diversas aportaciones ya realizadas a fin de descubrir cuanto se conoce sobre ese tema. Grande y Abascal (1999: 34-35) señalan varias estrategias para realizar la delimitación de objetivos y formulación del problema, estrategias que pueden utilizarse de forma aislada o conjunta:

- a. Consulta a expertos. Conversaciones y discusiones con las personas que toman decisiones o puedan aportar ideas, con el fin de realizar una puesta en común de conocimientos, valoraciones e inquietudes sobre el problema.
- b. Búsqueda y análisis de datos disponibles.
- c. Análisis de casos de situaciones similares para conocer cómo se actuó.

## **2) Elaboración de los objetivos específicos**

Esta determinación del tema elegido se concretará en la formulación de los objetivos de la investigación, aunque el objetivo no se define desde la propia investigación sino a partir de lo que se pretende con su realización, es decir, considerando el destino o utilidad de la información recogida. La formulación de un objetivo general y una serie de objetivos específicos donde concreta el objetivo general a nivel de los diversos aspectos, dimensiones y perspectivas que se desean analizar. Centrados en estos objetivos, es preciso distinguir claramente entre el objetivo general y los específicos:

- El objetivo general propone lo que se desea obtener con la investigación planteada. Presenta el enunciado claro y preciso de las metas que se persiguen con esta investigación, delimitando el ámbito temático concreto y la especificación de la población -diana del estudio. Para consecución del objetivo general será necesario apoyarse en los objetivos específicos.

- Los objetivos específicos indican lo que se pretende lograr en cada una de las etapas de la investigación, implicando así un mayor nivel de concreción temporal, temática y estratégica. Es conveniente evaluar estos objetivos en cada paso a fin de conocer los distintos niveles de resultados.

### **3) Elaboración del cuestionario y procedimiento de administración**

Seguidamente se procede a la elaboración del cuestionario mediante la Operacionalización de las variables formuladas en la etapa anterior. La elaboración de un cuestionario responde generalmente a tres objetivos: estimar magnitudes, describir una población y verificar hipótesis. En este momento se decide también el procedimiento de administración de la encuesta: personal, telefónica, postal, etc.

### **4) Prueba del cuestionario**

Definido el cuestionario, es necesario una prueba de éste con el objetivo de conocer su adecuación a los objetivos de la investigación; proceso conocido con el nombre “pretest” o “prueba piloto”. Para ello el investigador debe realizar varias pruebas del cuestionario no entre sus colegas o familiares, sino con extraños en sus hogares (o en el lugar donde se administre el cuestionario). Una prueba de este tipo suele comprender:

- Averiguar la adecuación de las preguntas realizadas (se entienden, no comenten sesgos, etc.), si el orden del cuestionario es pertinente, y el lenguaje es apropiado para la población objeto de estudio.
- Utilizar diferentes versiones del cuestionario en el que se presentan diversos tipos de preguntas para ver cuál funciona mejor.
- Aplicación, mediante entrevista personal, de lo que finalmente será un cuestionario rellenado por el propio entrevistado (autoadministrado), con el fin de detectar problemas de comunicación.

- Formulación de preguntas abiertas que serán utilizadas en la elaboración de categorías.
- Diferentes procedimientos de administración del cuestionario, para determinar la adecuación de éstos a los objetivos de la encuesta.

En esta fase, además de realizar una prueba del cuestionario, se “aprovecha la situación” para considerar otros elementos del proceso de investigación. Así Alvira (2004: 19) y Cea D’Ancona (2004: 300-3001) señala que –además de probar el cuestionario- el pretest suele utilizarse para:

- Analizar la organización del estudio y verificar hasta qué punto es adecuada.
- Comprobar la idoneidad del marco muestral empleado, verificar la adecuación de la selección muestral efectuada, y conocer el tipo de muestreo seleccionado para, en caso que sea preciso, estimar los parámetros que permitan determinar el tamaño muestral necesario.
- Estimar el porcentaje aproximado de *no respuesta* que se obtendrá en la encuesta, así como la planificación de estrategias para solucionar la no respuesta (visitas, incentivos, etc.).
- Valorar la adecuación de cómo se realizará la administración del cuestionario (entrevista personal, telefónica, por correo, internet, etc.). en ocasiones se prueban varios para ver cuál es el que mejor funciona.
- Planificar los contenidos a transmitir en la preparación de los entrevistadores, insistiendo en los elementos más complicados (detectados por el pretest).

- Estimar el tiempo necesario para llevar a cabo el trabajo del campo, y así poder realizar una estimación de tiempos y costes con más precisión.

## **5) Construcción de la muestra**

Una vez que el instrumento de medida está terminado y han tenido lugar todos los procesos verifcatorios de fiabilidad y validez, llega el momento de la localización de la población de interés que fue definida en la etapa de los objetivos, proceso que comienza con la elaboración de un marco de muestreo donde aparecen recogidos todos los elementos de la población. Una vez listada la población objeto de estudio se procede a la selección de una serie de “informantes privilegiados” empleando para ello los procedimientos desarrollados por la teoría muestral.

## **6) Selección de los entrevistadores**

En las encuestas telefónicas y personales el entrevistador es un componente esencial de la recogida de información en la medida que puede influir en la cooperación de los entrevistadores y en calidad de la información recogida. El entrevistador influye en el entrevistado con sus rasgos sociodemográficos, mediante la experiencia obtenida y por las expectativas originadas en la selección de cada entrevistado.

## **7) Formación y entrenamiento del personal seleccionado**

Una vez seleccionados los encuestadores llega el momento de proceder con su formación. Aunque en la selección de entrevistadores se ha podido escoger a las personas más capacitadas, es necesario realizar un periodo de formación en el que deben tratarse todos los aspectos implicados en la actuación del entrevistador. Bajo la premisa que el encuestador no nace, niño que se hace, es preciso diferenciar dos tipos de formación: una formación general donde transmiten los conocimientos y estrategias básicas para realizar cualquier tipo

de encuesta; y una formación específica referida a la investigación que se está desarrollando en ese momento.

Dentro del primer tipo de formación destinada al entrenamiento general de los entrevistadores se tratan aspectos como el proceso de localización del encuestado, la realización de la entrevista, estrategias a utilizar para el manejo de las situaciones problemáticas, y cómo realizar una primera revisión de la entrevista. Cuando el trabajo de campo se realiza por entrevistadores experimentados, o se confía a una empresa especializada en este tipo de tareas, se puede obviar esta formación específica referida a la investigación que se está desarrollando.

#### **8) Realización del trabajo de campo y supervisión de las entrevistas**

Terminado el periodo de formación es el momento de comenzar con la recogida de datos, planificando detalladamente las fechas en las se realizarán las entrevistas, la labor de los coordinadores de campo, la localización y el horario de las persona a la que acudir cuando aparezcan problemas, etc.

La última tarea de la recogida de información consiste en la supervisión y control de las entrevistas. En esta tarea no debe considerarse únicamente como una labor de “control” de los entrevistadores, aunque en las personas que realizan entrevistas por primera vez es conveniente comprobar su grado de honestidad y buena fe, al tipo que se evalúa su grado de pericia en el desarrollo de este trabajo. De este modo el proceso de supervisión y control ayuda a los entrevistadores a que determinados errores puntuales no se conviertan en hábitos.

## **9) Codificación de preguntas y depuración de la información**

Una vez finalizado el proceso de recogida de la información tiene lugar una tarea de revisión de los cuestionarios, donde se realiza una inspección y corrección de las respuestas. La primera revisión se realiza inmediatamente después de realizar la entrevista, durante la realización del trabajo de campo, y en ella se repasan todas las preguntas y comprueba que no se olvidó ninguna, se buscan contradicciones lógicas en las respuestas, etc. Tratando también de facilitar y asegurar la comprensión de las preguntas para los codificadores.

Una segunda revisión se realiza cuando los cuestionarios han llegado al instituto de investigación, y consiste en la precodificación de la información difícil (ocupaciones, preguntas semi-abiertas, etc.), así como la preparación de resúmenes para simplificar el proceso de codificación de las preguntas abiertas.

## **10) Tabulación y análisis de datos**

Finalizada la fase de preparación de la información comienza la etapa de análisis de datos. En un primer momento el objetivo se centra en obtener un conocimiento detallado de cada una de las variables utilizadas en la investigación, empleando para ello distribuciones de frecuencias, estadísticos univariantes y representaciones gráficas.

Una distribución de frecuencias es una tabla donde se muestran las elecciones de las distintas categorías que componen la variable. Cuando la variable se ha medido a nivel de intervalo es aconsejable la utilización de estadísticos que además permiten presentar la información en un formato más reducido. El análisis y la presentación de la información mejoran notablemente cuando se utilizan gráficos para la presentación de los resultados.

El análisis de una variable permite un primer conocimiento de la realidad objeto de estudio, además de preparar los datos para que puedan ser utilizados en las relaciones bivariantes. Aunque el primer conocimiento de la realidad obtenido mediante el análisis univariante es un paso previo e imprescindible antes de proceder con las relaciones entre variables, una mayor riqueza de análisis presenta la utilización de diversas técnicas de análisis bivalente o multivariante. La elección entre cada una de estas técnicas requiere, en primer lugar, considerar la escala en la que se ha medido cada variable (nominal, ordinal intervalo o razón), y en segundo lugar delimitar claramente la relación que se establece entre las variables: diferencia de porcentajes, diferencia de medias, relación significativa entre variables, etc.

## **11) Redacción del informe**

La labor de tratamiento y análisis de la información precede a la elaboración de un informe donde se presentan los hallazgos de la investigación. Este proceso de presentación y difusión de la información es uno de los aspectos más importante de la investigación, en la medida que es la única forma de dar a conocer los resultados al demandante de la misma, a los medios de comunicación y a la comunidad científica.

El informe de la investigación es, en definitiva, el trabajo de investigación en sí mismo puesto que dentro del informe se explican los objetivos de la investigación, los métodos y técnicas utilizadas para lograr los objetivos, así como las conclusiones del trabajo.

### **2.2.2.2. Entrevista<sup>6</sup>**

Las entrevistas permiten corregir fácilmente los errores producidos en el cumplimiento del cuestionario, bien preguntando al entrevistador unas horas

---

<sup>6</sup> Vidal Díaz de Rada. (2009). Análisis de datos de encuestas: Desarrollo de una investigación completa utilizando SPSS. Editorial UOC. Entrevista. Pág. 26

después de la realización de la entrevista, o incluso volviendo a solicitar la cooperación del entrevistado para algunos aspectos puntuales de la entrevista.

Esta corrección de errores es muy efectiva unas horas después de la entrevista, mientras que no serviría para mucho si se hiciera unos días después.

### **2.2.3. POBLACIÓN Y MUESTRA<sup>7</sup>**

#### **2.2.3.1. Conceptos Preliminares**

La estadística obtiene y estudia datos sobre diferentes individuos, que no tienen que ser necesariamente personas, hombres o mujeres. El conjunto de todos los individuos posibles constituye el universo.

En general, no interesan los datos de todos los individuos, de todo un universo, sino que se estudian poblaciones. Aún, así, obtener y analizar los datos de toda una población suele ser imposible, por lo que, en la práctica, suele seleccionarse una *muestra* de individuos de la población; únicamente en estos individuos estudiaremos los datos que nos interesan.

#### **2.2.3.2. Individuo**

Es cada elemento que lleva asociada una medida, un número de orden o una característica predeterminada.

#### **2.2.3.3. Universo**

Es el conjunto, finito o infinito, de todos los posibles individuos que cumplen ciertas propiedades.

---

<sup>7</sup> Joaquín Tomás-Sábado. (2010). Fundamentos de bioestadística y análisis de datos para enfermería. Univ. Autónoma de Barcelona. Población y Muestra. Pág. 21.



#### **2.2.3.4. Población**

Es el conjunto de todos los individuos que cumplen ciertas propiedades y de quienes deseamos estudiar ciertos datos. Podemos entender que una población abarca todo el conjunto de elementos de los cuales podemos obtener información, entendiendo que todos ellos han de poder ser identificados. La población deberá ser definida sobre la base de las características que la delimitan, que la identifican y que permiten la posterior selección de los elementos que se pueden entender como representativos (muestra).

Hay que distinguir entre población diana o población objetivo (aquella población a la que se desea extrapolar los resultados del estudio) y población accesible (aquella población cuyos individuos son directamente accesibles al investigador para seleccionar la muestra).

Así mismo, una población puede ser finita (por ejemplo, todos los enfermos de los hospitales de Cataluña o los estudiantes de enfermería de España) o infinita (los posibles resultados de sucesivas tiradas de una moneda o el conjunto de los números pares).

La población de la presente investigación será el total de personal que labora en el Área de Mantenimiento de Aplicaciones de la empresa TCS Ecuador.

#### **2.2.3.5. Muestra**

Una muestra es una porción de algo. Si deseamos preguntar a un conjunto de cinco mil personas su opinión sobre un determinado fenómeno, tenemos dos opciones: efectuar las preguntas persona por persona o efectuar las preguntas solamente a una muestra de estas personas, es decir, a un grupo de elementos representativos de ese conjunto.

La muestra es una parte o un subconjunto de la población en el que se observa el fenómeno a estudiar y de donde sacaremos unas conclusiones

generalizables a toda la población. En general, se considera que una muestra es grande cuando el número de individuos es igual o superior a 30, y una muestra es pequeña cuando los individuos son menores a 30.

Para que la muestra sea representativa de la población, deberá cumplir unas condiciones básicas:

- Han de delimitarse y definirse claramente las características que conforman la totalidad de la población.
- Han de haber garantías de que cada elemento de la población tiene las mismas posibilidades de figurar en la muestra. En consecuencia, deberá utilizarse el procedimiento de muestreo adecuado.
- La muestra deberá tener el tamaño adecuado para poder extrapolar los resultados obtenidos al conjunto de la población con garantías de fiabilidad.

La muestra de la presente investigación, se considerará como resultado de la siguiente fórmula:

$$\frac{k^2 N p q}{e^2 (N - 1) + k^2 p q}$$

- ✓ **N:** es el tamaño de la población o universo (número total de posibles encuestados).
- ✓ **k:** es una constante que depende del nivel de confianza que asignemos. El nivel de confianza indica la probabilidad de que los resultados de nuestra investigación sean ciertos: un 95,5 % de confianza es lo mismo que decir que nos podemos equivocar con una probabilidad del 4,5%. Los valores de k se obtienen de la tabla de la distribución normal estándar N (0,1).

Los valores de k más utilizados y sus niveles de confianza son:

K	1,15	1,28	1,44	1,65	1,96	2	2,58
Nivel de Confianza	75%	80%	85%	90%	95%	95,5%	99%

(Por tanto si pretendemos obtener un nivel de confianza del 95% necesitamos poner en la fórmula  $k=1,96$ )

✓ **e:** es el error muestral deseado, en tanto por uno. El error muestral es la diferencia que puede haber entre el resultado que obtenemos preguntando a una muestra de la población y el que obtendríamos si preguntáramos al total de ella.

✓ **p:** proporción de individuos que poseen en la población la característica de estudio. Este dato es generalmente desconocido y se suele suponer que  $p=q=0.5$  que es la opción más segura.

✓ **q:** proporción de individuos que no poseen esa característica, es decir, es  $1-p$ . **n:** tamaño de la muestra (número de encuestas que vamos a hacer).

#### 2.2.4. Software<sup>8</sup>

Se da el nombre de software a los programas que flexibilizan y hacen accesible el uso de las computadoras. De manera general, es posible afirmar que un programa es un conjunto de instrucciones que causa que una computadora ejecute una función particular. En los inicios de la computación, el software era el elemento menos importante y menos costoso de un sistema basado en computadora; sin embargo, actualmente el software generalmente es el elemento más caro de los sistemas de información.

---

<sup>8</sup> Andrés Gómez de Silva Garza, Ignacio de Jesús Ania Briseño. (2008). Introducción a la Computación. Cengage Learning Editores. Software. Pág. 23.

#### **2.2.4.1. Tipos de software**

Existen dos categorías principales en las que podemos dividir al software; de sistema y de aplicación.

- **Software de Sistema**

El software de sistema está constituido por los programas que se encargan del control y administración de los recursos de cómputo, y por los que permiten la interacción entre los usuarios y los sistemas de aplicación o el hardware. Ejemplos de este tipo de software son los traductores, (compiladores, intérpretes y ensambladores) y los sistemas operativos como Windows, XP o Vista, y varias versiones de Unix, por ejemplo AIX de IBM o Solaris de Sun.

- **Software de Aplicación**

El software de aplicación está compuesto por los programas que le permiten a una computadora realizar actividades específicas de procesamiento de información y ofrecer una funcionalidad a los usuarios finales.

Este tipo de software puede ser clasificado en las siguientes categorías: software de productividad, software para negocios, software educativo y software de entretenimiento. Algunos ejemplos de software de productividad son los procesadores de textos, las hojas de cálculo, los administradores de bases de datos, los editores de páginas web, los programas que soportan el desarrollo de presentaciones multimedia, los programas de estadística y los de ingeniería asistida por computadora.

Dentro del software para negocios se habla de dos subcategorías: el software estándar y el software específico. El estándar satisface de manera genérica una función específica de las empresas, como sistemas de nómina o de finanzas, y se puede personalizar para cada empresa. La integración de diversos módulos funcionales estándar constituye un software empresarial que,

generalizando, administra las operaciones vitales internas de la empresa (ERP), la cadena de suministro (SCM) y las relaciones con los clientes (CRM). El software específico, por otro lado, es desarrollado a la medida de una empresa para satisfacer necesidades particulares de ésta.

En el software educativo se incluyen los libros electrónicos, los programas tutoriales que capacitan al usuario sobre temas específicos, y los programas de referencia como las enciclopedias electrónicas. Por último, los juegos electrónicos se agrupan dentro del software de entretenimiento.

#### **2.2.4.2. Aspectos legales sobre el software**

Los vendedores de software normalmente tienen derechos de autor sobre los productos que desarrollan. El usuario compra el derecho a utilizar el software y las licencias impiden que lo compartan o modifiquen. Como alternativas al uso de software comercial se tienen el shareware y el freeware.

El shareware es software que se distribuye gratuitamente para propósitos de prueba bajo la hipótesis de que el usuario querrá pagar por él posteriormente. La mayoría de los vendedores de software ofrecen una versión shareware de sus programas con una fecha de expiración de operación. Otro tipo de shareware (como como liteware) ofrece sólo algunas de las capacidades del producto comercial. El freeware es software que se ofrece, normalmente, sin costo. Sin embargo, cuando se habla de freeware se habla de software libre, no necesariamente de software gratuito. Existen múltiples licencias de software libre, pero la más conocida es la Licencia Pública General (GPL) del proyecto GNU. Esta licencia está diseñada para que el usuario tenga la libertad de distribuir copiar del software libre (y cobrar por él si lo desea), y pueda estudiarlos, adaptarlo, mejorarlo, o utilizar partes de él en nuevos programas libres.

### **2.2.5. Sistema Informático<sup>9</sup>**

Un sistema informático es un conjunto de elementos físicos y lógicos (Computadores, personas, etc.) que, mediante la explotación de aplicaciones informáticas, y de forma coordinada y cooperativa, permiten resolver un problema.

### **2.2.6. Sistema Operativo<sup>10</sup>**

El sistema operativo, como un componente de una computadora, es un ejemplo de interacción entre software y hardware. Con dicha interacción se pueden realizar dos de las funciones principales de toda computadora: ejecutar programas y guardar información en archivos. El sistema operativo es un sistema (programa, paquete de software) que es desarrollado con las técnicas de programación, algoritmos y estructuras de datos.

### **2.2.7. Base de Conocimientos<sup>11</sup>**

Alberga las descripciones de los conceptos de conocimiento y sus relaciones, los criterios de búsqueda y las reglas de deducción que dependerán del tipo de problemas a resolver y el área de conocimiento concreta: medicina, prospección minera, planificación financiera, programación de la producción, etc.

---

<sup>9</sup> Modesto Castrillón, Antonio C. Domínguez, Santiago Candela, Luis Doreste, David Freire, Agustín Salgado, Sunil L. Khemchandani, Daniel Hernández. (2011). Fundamentos de Informática y Programación para Ingeniería. Definición de Sistema Informático. Cap. 1. Pág. 4.

<sup>10</sup> Andrés Gómez de Silva Garza, Ignacio de Jesús Ania Briseño. (2008). Introducción a la Computación. Cengage Learning Editores. Sistemas Operativos. Pág. 175.

<sup>11</sup> Carmen de Pablos Heredero, José Joaquín López Hermoso Agius Santiago Martín – Romo Romero, Sonia Medina Salgado. (2011). Organización y transformación de los sistemas de información en la empresa. 1era Edición. ESIC EDITORIAL. Base de Conocimiento. Pág. 165.

### **2.2.8. Base de Datos<sup>12</sup>**

Por lo general, una eficiente administración de datos requiere el uso de una base de datos de computadora. Una base de datos es una estructura computarizada compartida e integrada que guarda un conjunto de:

- Datos del usuario final, es decir, datos sin elaborar que son de interés para el usuario final.
- Metadatos, o datos acerca de los datos, por medio de los cuales los datos del usuario final son integrados y manejados.

Los metadatos dan una descripción de las características de los datos y del conjunto de relaciones que enlaza los datos encontrados dentro de la base de datos. Por ejemplo, el componente de metadatos guarda información como el nombre de cada elemento de datos, el tipo de valores (numéricos, fechas o texto) guardados en cada elemento de datos, si el elemento de datos puede dejarse vacío y así sucesivamente. Los metadatos dan información que complementa y expande el valor y uso de los datos. En pocas palabras, los metadatos presentan una imagen más completa de los datos en la base de datos.

### **2.2.9. Sistema de Administración de Bases de Datos (DBMS, por sus siglas en inglés)**

Es un conjunto de programas que maneja la estructura de la base de datos y controla el acceso a los datos guardados en ella. En cierto sentido, una base de datos se asemeja a un muy bien organizado gabinete electrónico de archivos en el que un poderoso software, conocido como sistema de administración de base de datos, ayuda a manejar el contenido del gabinete.

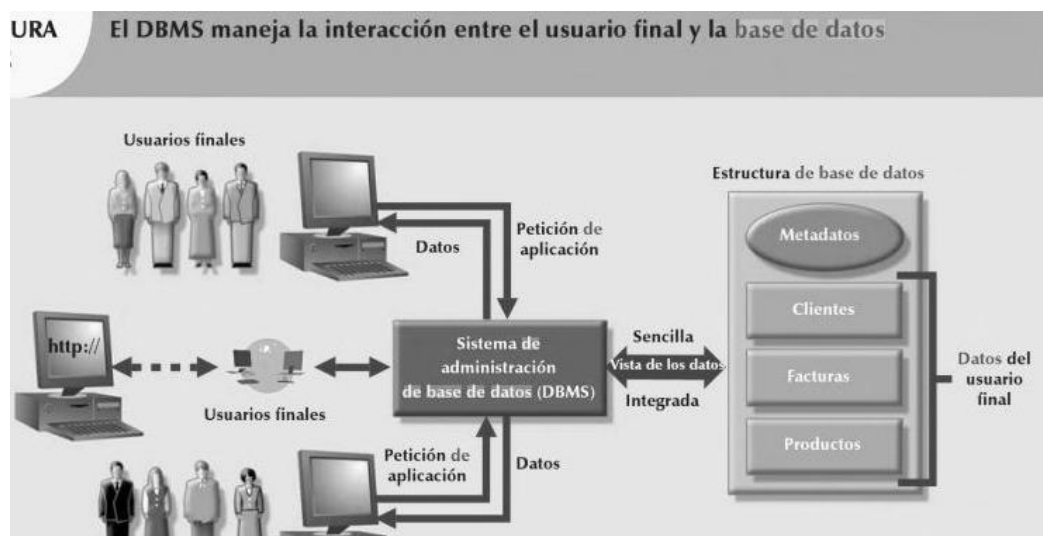
---

<sup>12</sup>Carlos Coronel, Steven Morris, Peter Rob. (2011). Bases de Datos, Diseño, Implementación y Administración. 9na Edición. Cengage Learning Editores S.A. Base de Datos. Pág. 7.

### 2.2.9.1. Función y Ventajas del DBMS

El DBMS sirve como intermediario entre el usuario y la base de datos. La estructura misma de la base de datos está guardada como un conjunto de archivos y la única forma de tener acceso a los datos de estos archivos es a través del DBMS. La figura destaca el punto de que el DBMS presenta al usuario final una sola vista integrada de los datos de la base de datos (o programa de aplicación). El DBMS recibe todas las peticiones de aplicación y las traduce en las complejas operaciones requeridas para cumplirlas. El DBMS oculta gran parte de la complejidad interna de la base de datos de los programas de aplicación a los usuarios. El programa de aplicación puede ser escrito por un programados usando un lenguaje de programación como Visual Basic.NET, Java o C, o podría ser creado por medio de un programa de utilería del DBMS.

**Figura 1. Interacción entre el usuario final y la Base de Datos**



Tener un DBMS entre las aplicaciones del usuario final y la base de datos ofrece algunas ventajas importantes. Primero, el DBMS hace posible que los datos de la base sean compartidos entre múltiples aplicaciones o usuarios. En segundo término, el DBMS integra las numerosas y diferentes vistas de usuarios de los datos en un solo depósito de datos que lo abarca todo.



Debido a que los datos son el material sin elaborar esencial del cual se deriva información, se debe tener un buen método para manejarlos. El DBMS ayuda a hacer que la administración de datos sea más eficiente y efectiva. En particular, el DBMS proporciona ventajas como:

- **Los datos se comparten mejor.** El DBMS ayuda a crear un ambiente en el que los usuarios finales tienen mejor acceso a más datos y mejor manejados.
- **Mejor seguridad de datos.** Mientras más usuarios tengan acceso a los datos, mayores son los riesgos para la seguridad de los datos. Las compañías invierten mucho tiempo, trabajo y dinero en asegurar que los datos comparativos se usen en forma apropiada. Un DBMS ofrece un marco para la aplicación eficaz de privacidad de datos y políticas de seguridad.
- **Mejor integración de datos.** Un acceso más amplio a datos bien manejados promueve una vista integrada de las operaciones de la organización y una vista más clara de todo lo que sucede en ella. Se vuelve más fácil ver cómo las actividades en un segmento de la compañía afectan a otros segmentos.
- **Mínima inconsistencia de datos.** Hay inconsistencia de datos cuando diferentes versiones de éstos aparecen en lugares diferentes. Por ejemplo, existe inconsistencia de datos cuando el departamento de ventanas de una compañía guarda el nombre de un representante como "William G. Brown", el departamento de personal como "Bill Brown" o cuando la oficina regional de ventas de una compañía muestra el precio de un producto como \$45.95 y su oficina nacional de ventas muestra el precio del mismo producto como \$43.95. la probabilidad de inconsistencias de datos se reduce en gran medida en una base de datos diseñada en forma correcta.
- **Mejor acceso a datos.** El DBMS hace posible producir respuestas rápidas a consultas ad hoc. Desde la perspectiva de una base de datos, una consulta

es una petición específica hecha al DBMS para manipulación de datos, por ejemplo leer o actualiza los datos. Dicho en pocas palabras, una consulta es una pregunta y una consulta ad hoc es una pregunta hecha a propósito para la ocasión. El DBMS envía una respuesta (llamada conjunto de resultados sobre la consulta) a la aplicación. Por ejemplo, usuarios finales, al trabajar con grande cantidades de datos de ventas, podría desear respuestas rápidas a preguntas (consultas ad hoc) como:

- ✓ ¿Cuál fue el volumen de ventas en dólares por producto durante los últimos seis meses?
- ✓ ¿Cuál es la cifra de bono de ventas para cada uno de los vendedores durante los últimos tres meses?
- ✓ ¿Cuántos de nuestros clientes tienen saldos de crédito de \$3.000 o más?

- **Mejor toma de decisiones.** Los datos mejor manejados y un destacado acceso a ellos hacen posible generar información de mayor calidad, en la que se basan las mejores decisiones. La calidad de la información generada depende de la calidad de los datos fundamentales. La calidad de los datos en un método completo para promover la precisión, validez y puntualidad de los datos. Si bien el DBMS no garantiza calidad de datos, da una estructura para facilitar iniciativas de calidad da datos.

- **Productividad aumentada del usuario final.** La disponibilidad de datos, combinada con las herramientas que transforman datos en información útil, hacen posible que los usuarios finales tomen decisiones rápidas e informadas que pueden hacer la diferencia entre éxito y fracaso en la economía global.

### **2.2.9.2. Tipos de Bases de Datos**

Un DBMS puede soportar números tipos de bases de datos. Las bases de datos se pueden clasificar según el número de usuarios, ubicación(es) de la base de datos y el tipo y grado de uso esperado.

El número de usuarios determina si la base de daros se clasifica como de un solo usuario o de usuarios múltiples. Una base de datos de un solo usuario soporta a un solo usuario a la vez. En otra palabras, si el usuario A usa la base de datos, los usuario B y C deben esperar hasta que el usuario A termine.

Una base de datos de usuario múltiples soporta a múltiples usuarios al mismo tiempo. Cuando una base de datos multiusuarios soporta a un número relativamente pequeño de usuarios (por lo general menos de 50) o un departamento específico dentro de una organización, se le da el nombre de base de datos de grupo de trabajo. Cuando la base de es utilizada por toda la organización y soporta a numerosos usuario (más de 50, por lo general cientos) de muchos departamentos, la base de datos se conoce como base de datos empresarial.

La ubicación también podría usarse para clasificar la base de datos. Por ejemplo, una base de datos que soporte datos ubicados en un solo sitio se denomina base de datos centralizada. Una base de datos que soporte datos distribuidos en vario lugares diferentes recibe el nombre de base de datos distribuida.

### **2.2.10. Lenguaje de Consulta Estructurada (SQL, por sus siglas en inglés)<sup>13</sup>**

SQL (*Structured Query Language*) es un lenguaje de programación diseñado específicamente para el acceso a Sistemas de Gestión de Bases de Datos

---

<sup>13</sup>G. Quintana, M. Marqués, J. L. Aliaga, M. J. Aramburu. (2008). Aprende SQL. Publicaciones de la Universidad Jaume. SQL. Pág. 8.

Relacionales (SGBDR). Como la mayor parte de los sistemas actuales son de este tipo y como el lenguaje SQL es el más ampliamente usado en éstos, se puede decir sin ningún género de dudas que este lenguaje es empleado mayoritariamente en los sistemas existentes hoy en día e indiscutiblemente no tiene rival alguno.

Este lenguaje es empleado en sistemas informáticos que van desde ordenadores personales muy básicos con apenas 64 MB de espacio en memoria central hasta los más potentes multiprocesadores y multicomputadores con decenas de procesadores superescalares de 64 bits.

Las principales ventajas que aporta SQL son dos:

- Su enorme difusión pues es empleado en la gran mayoría de los sistemas actuales.
- Su elevada expresividad. Por ejemplo, operaciones que costaría semanas de duro esfuerzo en ser desarrolladas en un lenguaje de programación tradicional pueden ser realizadas con SQL en tan sólo unos minutos.

El lenguaje SQL es un lenguaje de cuarta generación. Es decir, en este lenguaje se indica qué información se desea obtener o procesar, pero no cómo se debe hacer. Es labor interna del sistema elegir la forma más eficiente de llevar a cabo la operación ordenada por el usuario.

#### **2.2.10.1. Partes de SQL**

El lenguaje SQL consta de dos partes claramente diferenciadas:

- Lenguaje de Definición de Datos (en inglés *Data Definition Language* o *DDL*): Incluye aquellas sentencias que sirven para definir los datos o para modificar su definición, como por ejemplo la creación de tablas, índices, etc.

- Lenguaje de Manipulación de Datos (en inglés *Data Manipulation Language* o DML): Incluye aquellas sentencias que sirven para manipular o procesar los datos, como por ejemplo la inserción, borrado, modificación o actualización de datos en las tablas.

#### **2.2.11. Microsoft SQL Server<sup>14</sup>**

SQL Server es un sistema de Gestión de Bases de Datos Relacionales (SGBDR), lo que le confiere una gran capacidad de gestionar datos, conservando su integridad y su coherencia.

SQL Server se encarga de:

- Almacenar datos.
- Verificar las restricciones de datos definidas.
- Garantizar la coherencia de los datos que almacena, incluso en caso de error (parada repentina) del sistema.
- Asegurar las relaciones entre los datos definidos por los usuarios.

#### **2.2.12. Entorno de Desarrollo Integrado (IDE, por sus siglas en inglés)<sup>15</sup>**

Un entorno de desarrollo integrado o en inglés *Integrated Development Environment* (IDE) es un programa compuesto por un conjunto de herramientas para un programador. Puede dedicarse exclusivamente a un solo lenguaje de programación o bien, poder utilizarse para varios.

---

<sup>14</sup>Jérôme Gabillaud. (2010). Recursos Informáticos SQL Server 2008 – Administración. Ediciones ENI. Sistema de Gestión de Bases de Datos Relacionales. Pág. 17.

<sup>15</sup>Vicente Javier Eslava Muñoz. (2012). Aprendiendo a programar paso a paso con C. Bubok Publishing S.L. Entorno de –Desarrollo Integrado (IDE). Pág. 10.

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI.

#### **2.2.13. Lenguaje de Programación<sup>16</sup>**

Un lenguaje de programación es un lenguaje artificial que define, de forma secuencial y sistemática, las instrucciones y códigos que debe procesar un ordenador, mediante instrucciones traducidas por la máquina y que son comprensibles para ella.

#### **2.2.14. Framework<sup>17</sup>**

Un framework es un conjunto de bibliotecas, herramientas y normas a seguir que ayudan a desarrollar aplicaciones. Los frameworks los desarrollan los programadores de sistemas. Un framework está compuesto por varios segmentos / componentes que interactúan los unos con los otros.

#### **2.2.15. Microsoft Visual Studio .NET<sup>18</sup>**

Microsoft Visual Studio .NET es un entorno integrado de desarrollo (IDE) compartido y único para todos los lenguajes .NET. El entorno proporciona acceso a todas las funcionalidades del .NET Framework, así como a muchas otras funcionalidades que hacen que el desarrollo de aplicaciones se más ágil.

Visual Studio .NET soporta los siguientes lenguajes de programación:

- Visual Basic .NET.

---

<sup>16</sup>Andrea Pastor. (2010). Cultura General. Ediciones Paraninfo S.A. Lenguaje de Programación. Pág. 412.

<sup>17</sup>Jérôme Lafosse. (2010). Expert IT Struts 2 - El framework de desarrollo de aplicaciones Java EE. Ediciones ENI. Framework. Pág. 11.

<sup>18</sup>Jordi Conesa Caralt, Jordi Ceballos Villach, David Gañán Jiménez, Àngels Rius Gavidia. (2010). Introducción a .NET. Editorial UOC. Microsoft Visual Studio.NET. Pág. 20.

- Visual C#.
- Visual C++.
- Visual F#.
- JavaScript.
- ASP.NET.

#### **2.2.16. Programación Orientada a Objetos (OOP, por sus siglas en inglés)<sup>19</sup>**

La programación orientada a objetos (POO) no sustituye a ninguna metodología ni lenguaje de programación anterior, es un enfoque distinto. Todos los programas que se realizan con una orientación a objetos se pueden realizar igualmente mediante programación estructurada.

La programación orientada a objetos es una nueva filosofía frente a los programas tradicionales o estructurados. En la programación tradicional, las estructuras de datos y los procedimientos que las manipulan se encuentran separados. En la programación orientada a objetos, se incluyen en el mismo módulo al que se denomina objeto. Este giro, supone una nueva forma de pensar en el proceso de diseño y desarrollo de programas.

Se puede definir la programación orientada a objetos (POO) como un paradigma de programación que considera un programa como colección de objetos, cada uno de ellos como una instancia de una clase, que interactúan enviándose mensajes.

---

<sup>19</sup> Ángel Cobo Yera. (2009). Programar desde un punto de vista científico. Editorial Visión Libros. Programación Orientada a Objetos. Pág. 129.

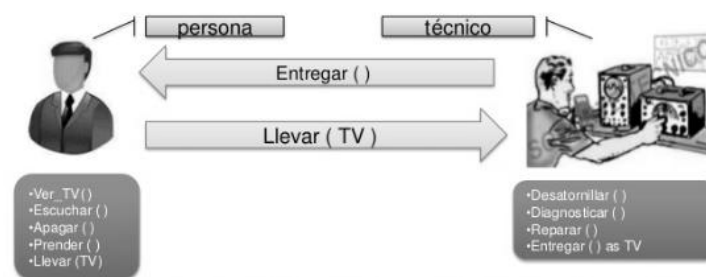
### 2.2.16.1. Principios de la POO

La POO se basa en los siguientes principios:

- **Encapsulamiento**

Significa ocultación de la información. Cada objeto tiene un conjunto de propiedades que no pueden ser accedidos directamente desde el exterior. Para conseguir la información de un objeto, debe realizarse mediante mensajes a él, siempre que el solicitante tenga los permisos de acceso a ella.

**Figura 2. Ilustración de Encapsulamiento**



- **Abstracción**

Es la capacidad de aislar la información del diseño de los detalles de la implementación. En el proceso de abstracción no debemos preocuparnos por la implementación de cada método, solamente debemos definirlo de forma general. Por medio de la abstracción definimos las características esenciales de un objeto en el mundo real.

- **Modularidad**

Consiste en descomponer una aplicación en partes más pequeñas (Módulos). Con la programación orientada a objetos, podemos dividir nuestra aplicación en módulos separados, normalmente clases, y estos módulos pueden compilarse y modificarse sin afectar a los demás.



- **Jerarquía**

Las clases se organizan en una estructura jerárquica por la relación de herencia. Donde se definen tres niveles de clases.

- ✓ **Raíz o Superclase.** Es una clase especial situada en el punto más alto de la organización jerárquica.
- ✓ **Clases Intermedias o Subclases.** Descienden de otra clase y heredando sus propiedades de forma transitiva. Si C hereda de B y B de A entonces C hereda de A.
- ✓ **Clases Terminales.** Desciende de una subclase y no tienen descendientes, los objetos normalmente son instancias de estas clases. Al igual que las anteriores heredarán las propiedades de las clases de las que descenden.

#### **2.2.16.2. Ventajas / Desventajas POO**

La OOP proporciona las siguientes ventajas sobre otros lenguajes de programación:

- **Protección de la información**

El encapsulamiento de datos, junto con la definición de niveles de protección proporciona un alto nivel de seguridad de la información.

- **Permite el trabajo en equipo**

Cada programador puede desarrollar clases diferentes, y posteriormente enlazarlas.

- **Aumento de claridad**

Tanto los datos que componen los objetos, como los procedimientos que los manipulan, están agrupados en clases, que se corresponden con las estructuras de información que el programa trata.

- **Facilitar la extensibilidad**

La ampliación de un programa orientado a objetos es más simple ya que se limita a ampliar clases existentes o añadir nuevas clases.

- **Facilita las modificaciones y correcciones**

Al tener relacionados los procedimientos que manipulan los datos con los datos a tratar, cualquier cambio que se realice sobre ellos quedará reflejado automáticamente en cualquier lugar donde estos datos aparezcan.

- **Obtener componentes reutilizables**

La noción de objeto permite que programas que traten las mismas estructuras de datos reutilicen clases empleadas en otros programas e incluso los procedimientos que los manipulan. De esta forma, el desarrollo de un programa puede llegar a ser una simple combinación de objetos ya definidos donde éstos están relacionados de una manera particular.

Como desventajas, se citan:

- **Velocidad de ejecución**

Del mecanismo de ligadura dinámica, se necesita un 75% más de tiempo extra.

- **Curva de aprendizaje**

El uso de librerías de clases, en las herramientas orientadas a objetos viene en una jerarquía compleja de biblioteca usando herencia, polimorfismo, etc.

- **Interfaz**

Son las propiedades y métodos de la clase que son accesibles desde el exterior, pero sin incluir la implementación de los métodos. Con él los objetos se comunican con el exterior. La implementación de los métodos contiene el código que sustenta la interfaz y es de carácter privado.

### **2.2.16.3. Paso de Mensajes**

Los mensajes son el mecanismo que permite la comunicación entre objetos. El paso de mensaje es una solicitud a un objeto para realizar una acción específica. En la práctica, esto supone la ejecución de un método por parte de un objeto determinado.

En el paso de mensajes se debe especificar:

- El nombre del objeto que recibe el mensaje.
- Selector del mensaje (nombre del método).
- Argumentos en caso necesario.

Ejemplo:

Objeto.Selector (parámetros)

Como se puede apreciar, en el envío de mensajes no se hace referencia a la implementación del método. Esto permite modificar la implementación de los

objetos sin cambiar los programas que lo utilizan. Un objeto se comporta como una caja negra, el usuario necesita conocer su interfaz pero no los detalles de implementación.

Existen dos métodos especiales en un objeto:

- **Métodos constructores.** Permiten la creación de objetos, con la inicialización de éstos.
- **Métodos destructores.** Liberan el objeto de la memoria. Los métodos pueden heredarse de unos objetos a otros, por lo tanto un objeto puede disponer de un método de dos maneras diferentes:
- **Métodos propios.** Están incluidos dentro de la clase que instancia al objeto.
- **Métodos heredados.** Están definidos en un objeto antepasado que lo heredó.

#### 2.2.16.4. Clases

- **Concepto**

Las clases son plantillas, que permiten agrupar a objetos que comparten las mismas propiedades y comportamiento. Una clase es una implementación, de un tipo abstracto de datos (TAD).

Una clase es una estructura estática, mientras que un objeto es una estructura dinámica que sólo existe durante la ejecución del programa. Todo objeto es una instancia de una determinada clase. Dos instancias diferentes de una misma clase comparten el mismo conjunto de procedimientos y propiedades.

Por ejemplo, existen rectángulo con diferentes tamaños pertenecientes que comparte los atributos ancho y alto, y el método dibujar, pero pertenecen a la clase rectángulo.

- **Componentes**

Atributos o propiedades que determinan las estructuras de datos de cada objeto. Los atributos en tiempo de ejecución contienen los valores que caracterizan al objeto.

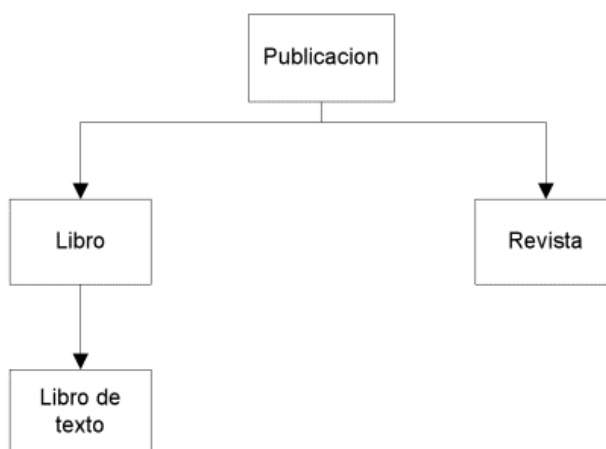
#### 2.2.16.5. Herencia

- **Concepto**

La herencia es la propiedad que permite definir una clase nueva a partir de una clase ya existente. Así podemos ver objetos como subclase de otra, que será su superclase, de la que heredan sus atributos y métodos.

Las instancias de una clase hija pueden tener acceso a los atributos y métodos de las clases de la jerarquía de la que heredan. Un ejemplo podría ser:

**Figura 3. Ejemplo de Herencia**



Al ser la herencia transitiva, la clase libro de texto hereda de publicación a través de la clase libro, sus métodos y atributos.

- **Uso de la herencia**

Cuando una subclase hereda de una superclase, la subclase puede por lo general:

- ✓ Añadir nuevos atributos y métodos.
- ✓ Redefinir las operaciones heredadas, proporcionando una nueva implementación de las mismas.

Cuando una clase descendiente define un método con el mismo nombre que el usado en la clase paterna, el método de la clase antecesora. Así, cuando se le pasa un mensaje a un objeto, se activará el método de la clase hija y no el original de la clase padre. Es lo que se llama reemplazo o refinamiento.

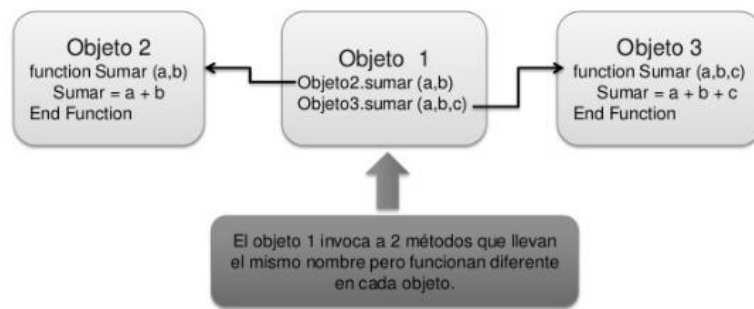
Una regla básica de la herencia es la especialización, consiste en que una clase B herede de una clase A, bajo la relación “es un”.

#### **2.2.16.6. Polimorfismo**

El polimorfismo es la propiedad por la cual una entidad puede tomar diferentes formas. En la POO el polimorfismo hace referencia a que un identificador de un objeto pueda referenciar en tiempo de ejecución a instancias de diferentes clases.

El polimorfismo también aparece en un paso de parámetros cuando en la definición del método se ha utilizado un parámetro y se pasa en tiempo de ejecución otro parámetro.

**Figura 4. Ilustración de Polimorfismo**



#### 2.2.16.7. Sobrecarga

La sobrecarga consiste en definir dos o más métodos con el mismo nombre pero con argumentos de distinto tipo o en distinto número. Es un mecanismo estático y el compilador se encargará de utilizar el método adecuado dependiendo de los argumentos que se le pasen.

#### 2.2.16.8. Objeto

- **Concepto**

Es el concepto principal sobre el cual se fundamenta la POO. Un objeto puede definirse como una encapsulación de datos, que definen el estado del objeto y métodos que representan el comportamiento del objeto.

**Figura 5. Ilustración de Objeto**



Los objetos son casos particulares de entidades llamadas clases donde se definen las características comunes de los objetos.

Los objetos son estructuras dinámicas, no existen hasta la ejecución del programa. Durante su ejecución se crearán un conjunto de objetos, que ocupan un espacio de memoria, y serán instancias de un conjunto de clases.

Por ejemplo, si se tiene una clase de tipo Pila, instancias de esta clase (objetos) puedan ser una pila de números reales, una pila de números enteros, etc.

- **Propiedades**

Un objeto presenta tres aspectos:

✓ **Estado.** Son los valores actuales de cada uno de los atributos del objeto. Estos valores sólo pueden ser modificados y consultados por la implementación de los métodos asociados a cada objeto.

✓ **Identidad.** Todo objeto tiene una identidad única. La identidad expresa que aunque dos objetos sean exactamente iguales en los valores de sus atributos, son distintos entre sí. Además, un objeto durante la ejecución de un programa puede variar los valores de sus atributos sin que varíe su identidad.

✓ **Interfaz.** Son las propiedades y métodos de la clase que son accesibles desde el exterior, pero sin incluir la implementación de los métodos. Con él los objetos se comunican con el exterior. La implementación de los métodos contiene el código que sustenta la interfaz y es de carácter privado.



## **2.2.17. Metodología de Desarrollo del Software**

### **2.2.17.1. Metodologías Ágiles - RAD<sup>20</sup>**

A principios de la década del '90, surgió un enfoque que fue bastante revolucionario para su momento ya que iba en contra de toda creencia de que mediante procesos altamente definidos se iba a lograr obtener software en tiempo, costo y con la requerida calidad. El enfoque fue planteado por primera vez por Martin y se dio a conocer en la comunidad de Ingeniería de Software con el nombre de RAD o Rapid Application Development. RAD consistía en un entorno de desarrollo altamente productivo, en el que participaban grupos pequeños de programadores utilizando herramientas que generaban código en forma automática tomando como entradas sintaxis de alto nivel. En general, se considera que este fue uno de los primeros hitos en pos de la agilidad en los procesos de desarrollo.

La historia de las Metodologías Ágiles y su apreciación como tales en la comunidad de la Ingeniería de Software tiene sus inicios en la creación de una de las metodologías utilizada como arquetipo: XP - eXtreme Programming, que nace de la mente de Kent Beck, tomando ideas recopiladas junto a Ward Cunningham.

### **2.2.17.2. Etapas de la metodología RAD<sup>21</sup>**

- La etapa de Definición Conceptual que define las funciones del negocio y las áreas sujeto de datos que el sistema apoyará y determina el alcance del sistema.

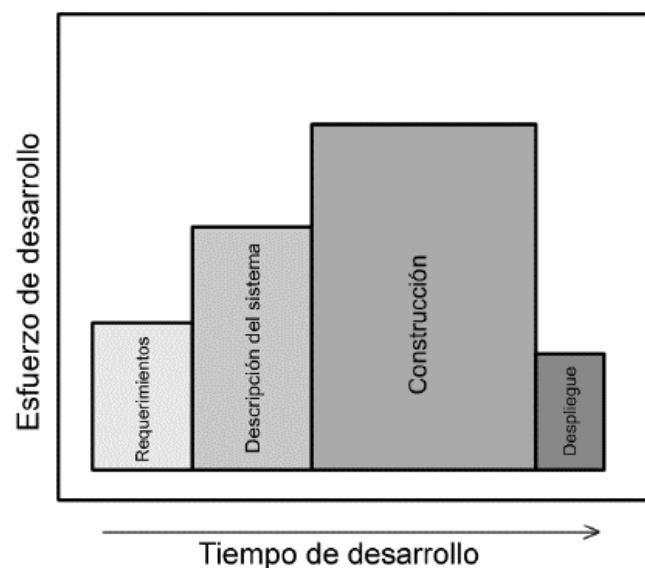
---

<sup>20</sup> Amaro Calderón Sarah Damaris, Valverde Rebaza Jorge Carlos. (2007). Metodologías Ágiles. Universidad Nacional de Trujillo. Metodologías Ágiles.

<sup>21</sup> Israel García Portuondo, Lissette Fonseca Suárez. (2010). Desarrollo Rápido de Aplicaciones Apoyado en MDA. Facultad Regional "Mártires de Artemisa" de la Universidad de las Ciencias Informáticas (UCI). Etapas de la Metodología RAD. Pág. 2.

- La etapa de Diseño Funcional que usa los talleres para modelar los datos y los procesos del sistema y para construir un prototipo de trabajo de los componentes críticos del sistema.
- La etapa de Desarrollo que completa la construcción física de la base de datos y del sistema de aplicación, construye el sistema de conversión y elabora ayudas de usuarios y planes de trabajo a desarrollar o de despliegue.
- La etapa de Despliegue que incluye la puesta a prueba y la capacitación del usuario final, la conversión de datos y la implementación del sistema de aplicación.

**Ilustración del Modelo RAD**



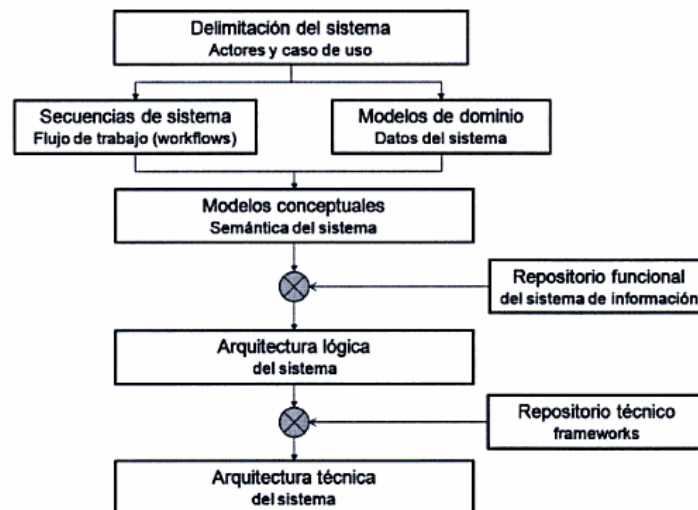
#### 2.2.18. Lenguaje Unificado de Modelamiento (UML, por sus siglas en inglés)<sup>22</sup>

Es una familia de gráficas con sus correspondientes símbolos diseñados para describir software, programas y sistemas de información. Esta nomenclatura permite describir procesos con la finalidad de desarrollar los sistemas de

<sup>22</sup> David Muñoz Negrón. (2009). Administración de operaciones. Enfoque de administración de procesos de negocios. Cengage Learning Editores. Lenguaje Unificado de Modelamiento UML. Pág. 49.

información que apoyarán sus correspondientes procesos, de esta manera se facilita la reingeniería de procesos y el desarrollo de sistemas de información. Existen herramientas CASE (*Computer Aided Software Engineering*) que traducen gráficas UML a código de software y viceversa.

**Figura 6. Ilustración de un proceso de análisis basado en UML**



#### 2.2.18.1. Conceptos de Modelado de Software<sup>23</sup>

Un modelo es una abstracción de un sistema o entidad del mundo real. Una abstracción es una simplificación, que incluye sólo aquellos detalles relevantes para algún determinado propósito. El modelado permite abordar la complejidad de los sistemas. Los conceptos de modelado son los siguientes:

- **Sistema.** Colección de elementos, posiblemente divididos en subsistemas, organizados para lograr un propósito. Esta descrito por un conjunto de modelos.
- **Modelo.** Simplificación completa y autoconsistente de la realidad, creado para comprender mejor un sistema.

<sup>23</sup> Patricia López, Francisco Ruíz. (2011). Ingeniería del Software I. Universidad Cantabria. Lenguaje Unificado de Modelado – UML. Tomado de <http://ocw.unican.es/enseanzas-tecnicas/ingenieria-del-software-i/materiales-de-clase-1/is1-t02-trans.pdf>.

Los modelos a utilizar los define la metodología que se aplique en el proceso de desarrollo. Los principales modelos son los siguientes:

- ✓ Modelo de casos de uso.
- ✓ Modelo de análisis.
- ✓ Modelo de diseño.
- ✓ Modelo de implementación.
- ✓ Modelo de despliegue.
- ✓ Modelo de pruebas.
- **Vista.** Proyección de la organización y estructura de un modelo de un sistema, centrada en un aspecto. Incluye un subconjunto de los elementos incluidos en el modelo.

Existen 5 tipos de vistas arquitecturales:

- ✓ **Vista de Casos de Uso.** Captura la funcionalidad del sistema tal y como es percibido por los usuario finales, analistas y encargados de pruebas.
- ✓ **Vista de Diseño.** Captura las clases, interfaces y colaboraciones que describen el sistema en el dominio del problema y el dominio de la solución.
- ✓ **Vista de Interacción.** Captura el flujo de control entre las diversas partes del sistema, incluyendo los posibles mecanismos de concurrencia y sincronización. Abarca en especial requisitos no funcionales como el rendimiento, escalabilidad y capacidad de procesamiento.

✓ **Vista de Implementación.** Captura los artefactos que se utilizan para ensamblar y poner en producción el sistema software real. Define la arquitectura física del sistema.

✓ **Vista de Despliegue.** Captura las características de instalación y ejecución del sistema. Contiene los nodos y enlaces que forman la topología hardware sobre la que se ejecuta el sistema software.

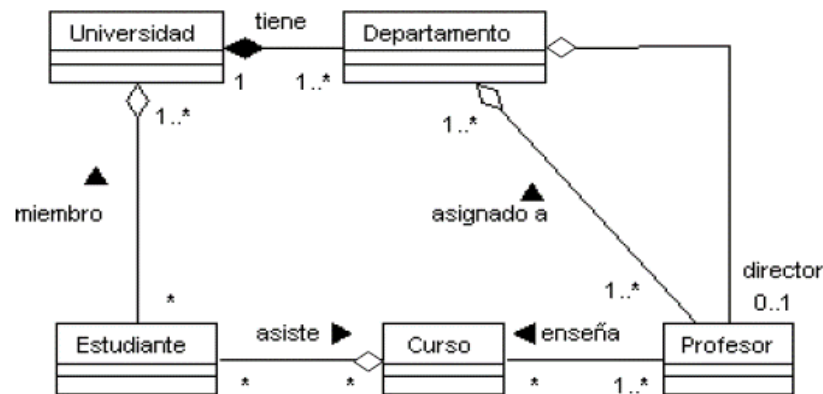
- **Diagrama.** Representación gráfica de un conjunto de elementos del modelo y sus relaciones. En UML generalmente corresponde a un grafo conexo de nodos (elementos) y arcos (relaciones).

El UML 2 incluye diagramas divididos en estructurales y diagramas de comportamiento:

Diagramas estructurales, sirven para visualizar, especificar, construir y documentar los aspectos estáticos de un sistema:

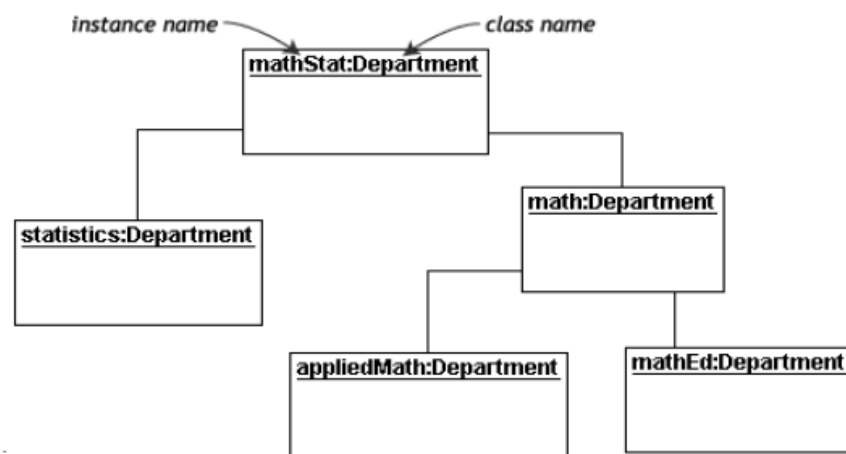
✓ **Diagrama de Clases.** Muestran un conjunto de clases, interfaces y colaboraciones, así como las relaciones entre ellos.

**Figura 7. Ilustración de Diagrama de Clases**



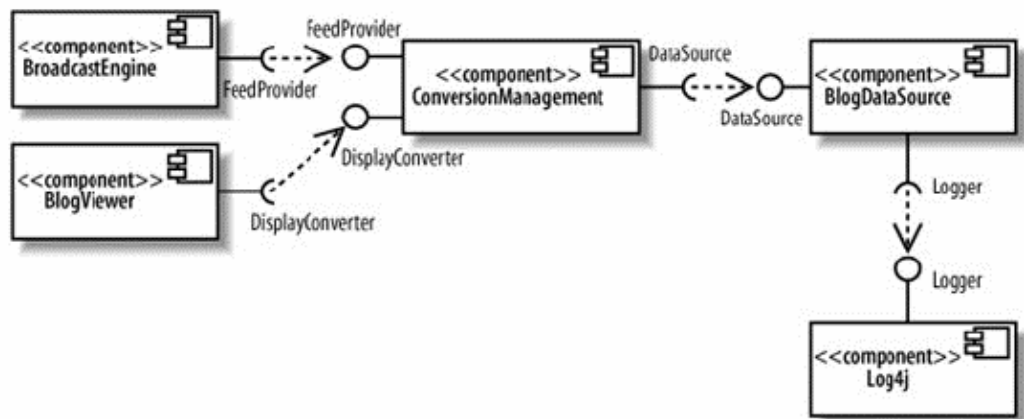
✓ **Diagrama de Objetos.** Muestra un conjunto de objetos y sus relaciones. Representan instantáneas estáticas de instancias de los elementos existentes en los diagramas de clases. Describen la vista de diseño estática pero desde el punto de vista de casos reales.

**Figura 8. Ilustración de Diagrama de Objetos**



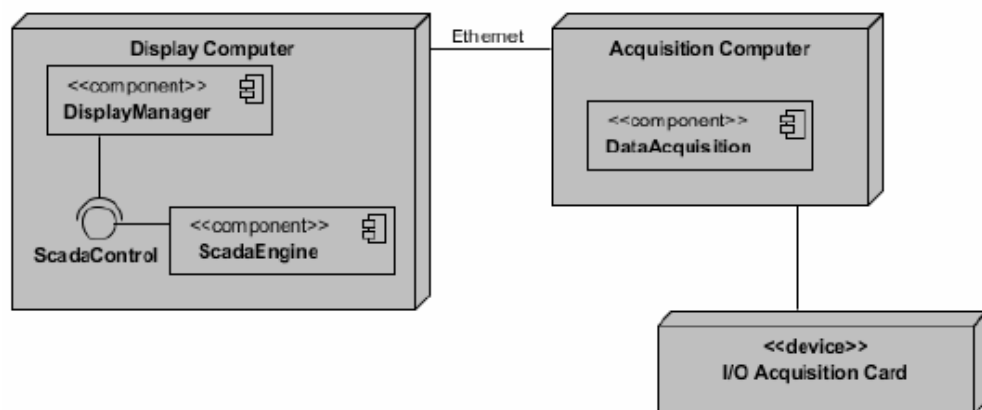
✓ **Diagrama de Componentes.** Describen la estructura del software mostrando la organización y las dependencias entre un conjunto de componentes. Pueden representar la encapsulación de un componente con sus interfaces, puertos y estructura interna (posiblemente formada por otros componentes anidados y conectores).

**Figura 9. Ilustración de Diagrama de Componentes**



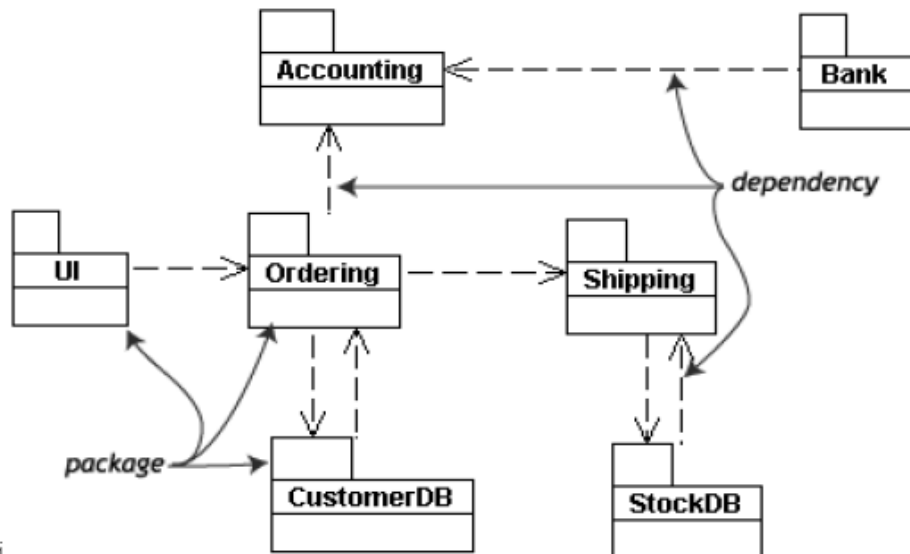
✓ **Diagrama de Despliegue.** Muestran un conjunto de nodos y sus relaciones. Describen la vista de despliegue estática de una arquitectura. Cada nodo (hardware) suele albergar uno o más componentes. Muestran el hardware, el software y el middleware usado para conectar las máquinas.

**Figura 10. Ilustración de Diagrama de Despliegue**



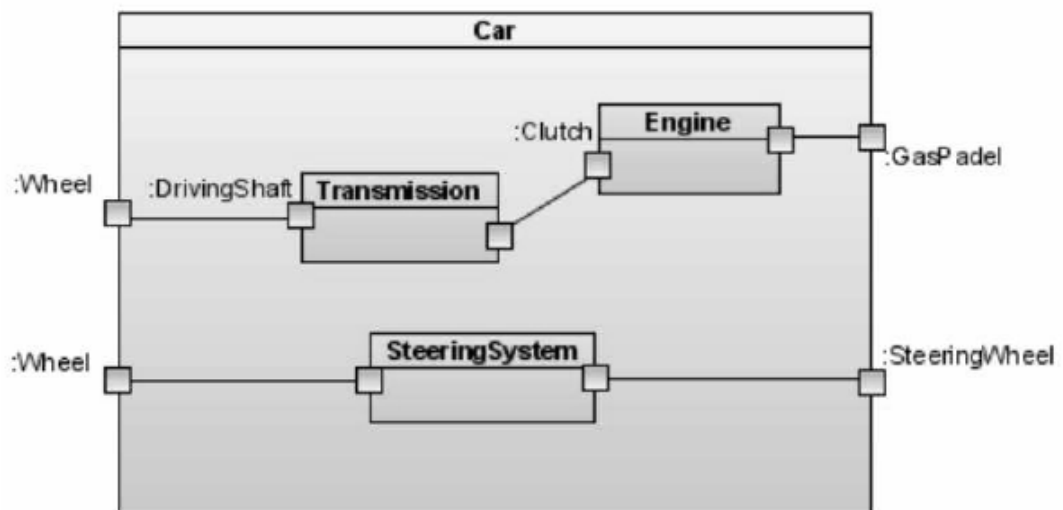
✓ **Diagrama de Paquetes.** Muestran la descomposición del propio modelo en unidades organizativas (paquetes) y sus dependencias. Sirven para simplificar los diagramas de clases complejos, permitiendo el agrupamiento de los clasificadores en paquetes.

**Figura 11. Ilustración de Diagrama de Paquetes**



✓ **Diagrama de Estructura Compuesta.** Muestran la estructura interna (incluyendo partes y conectores) de un clasificador estructurado o una colaboración. Muy parecidos a los diagramas de componentes.

**Figura 12. Ilustración de Diagrama de Estructura Compuesta**

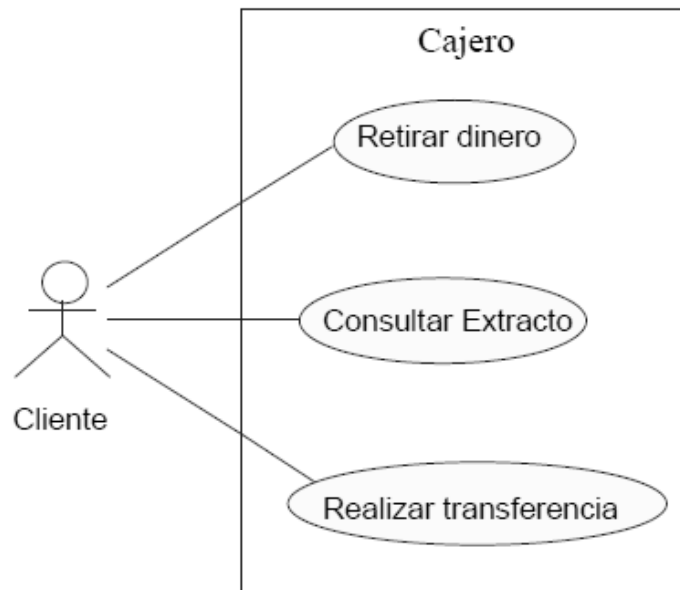


Diagramas de comportamiento, sirven para visualizar, especificar, construir y documentar los aspectos dinámicos de un sistema:

✓ **Diagrama de Casos de Uso.** Muestran un conjunto de casos de uso y actores (tipo especial de clase) y sus relaciones.

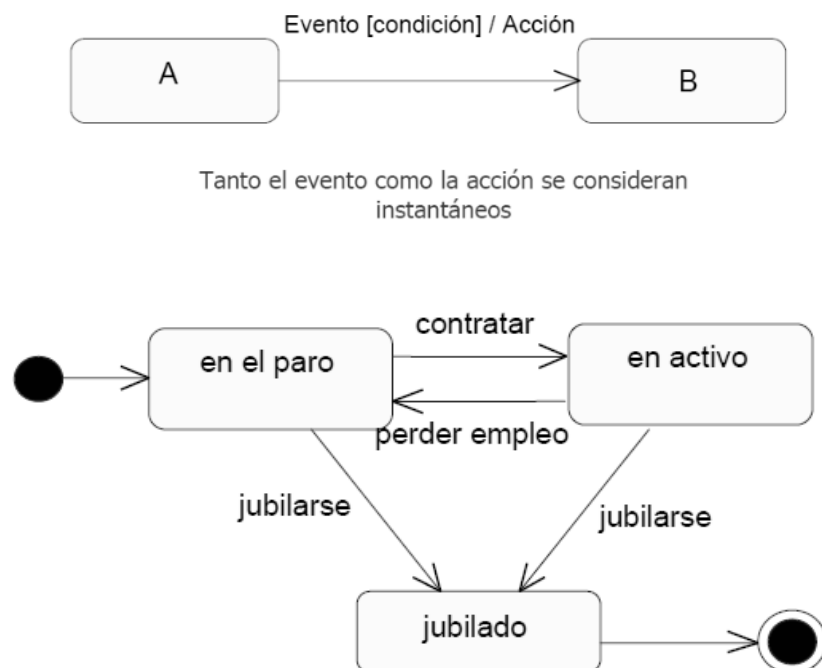


**Figura 13. Ilustración de Diagrama de Casos de Uso**



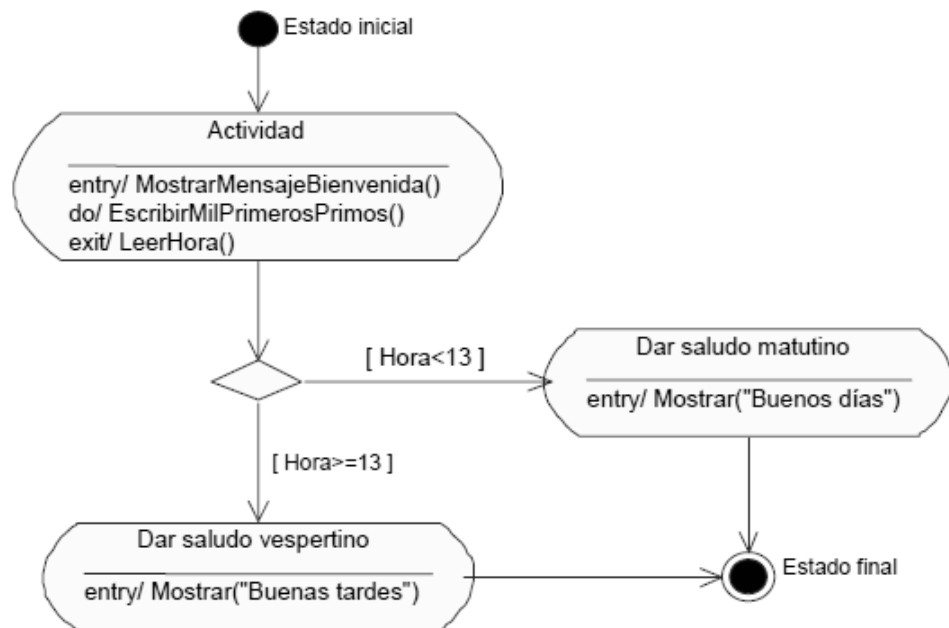
✓ **Diagrama de Estados.** Muestran máquinas de estados, que constan de estados, transiciones, eventos y acciones.

**Figura 14. Ilustración de Diagrama de estados**



✓ **Diagrama de Actividades.** Muestran el flujo paso a paso de una computación (proceso, flujo de control o flujo de datos). Una actividad muestra un conjunto de acciones, el flujo entre ellas y los valores producidos o consumidos.

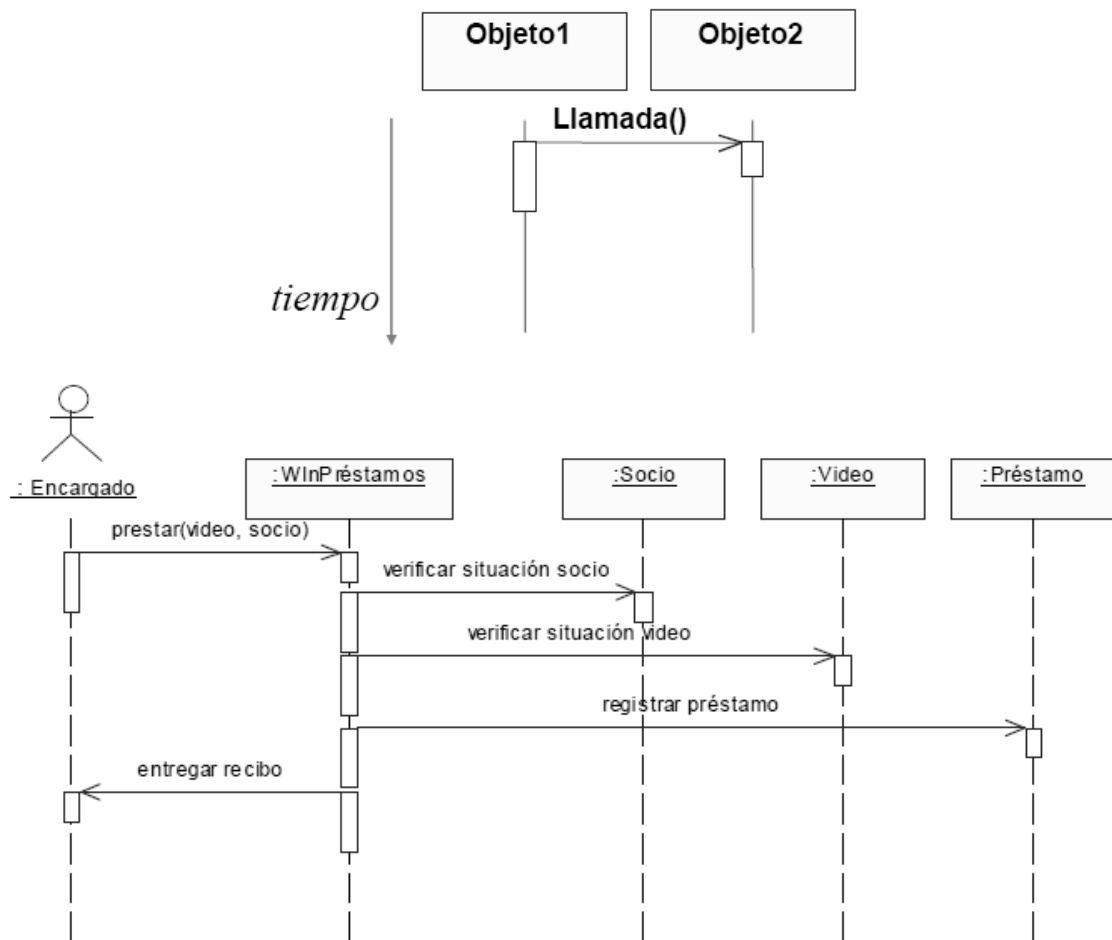
**Figura 15. Ilustración de Diagrama de Actividades**



✓ **Diagrama de Interacción (secuencia, comunicación, tiempos, revisión de interacciones).** Son un grupo especial de diagramas de comportamiento que muestran una interacción.

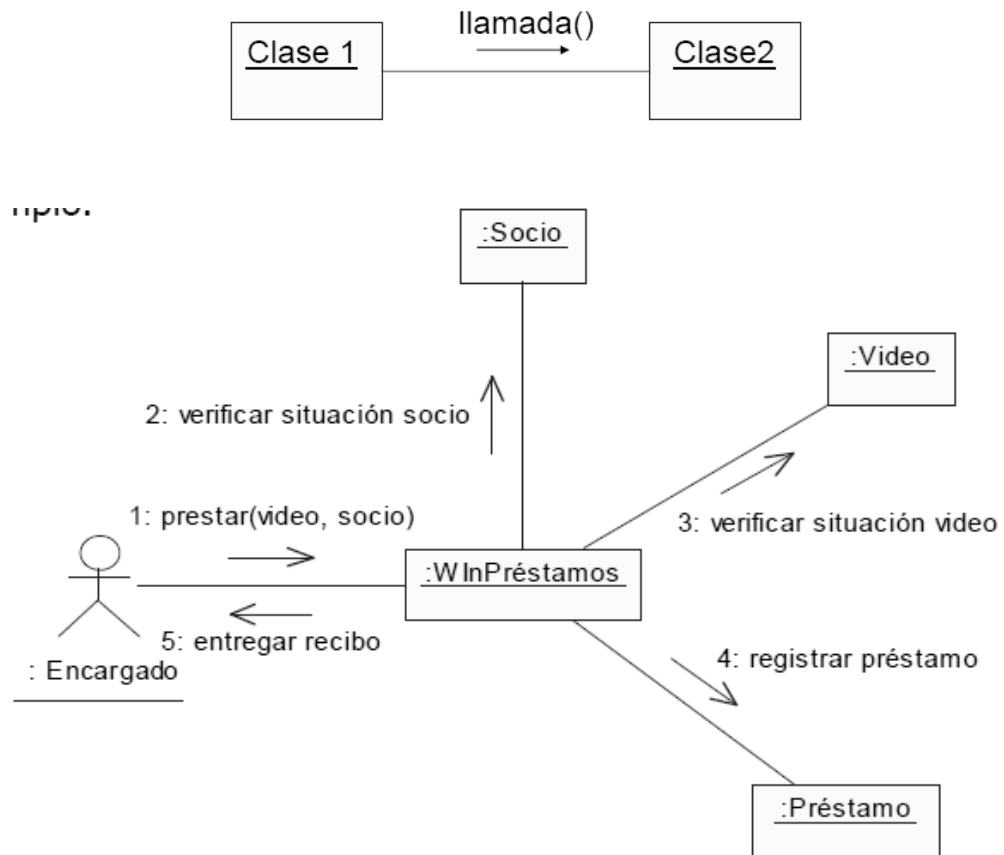
▪ **Diagramas de Secuencia.** Es un diagrama de interacción que resalta la ordenación temporal de los mensajes intercambiados durante la interacción. Presentan un conjunto de roles y los mensajes enviados y recibidos por las instancias que interpretan dichos roles. Habitualmente, sirven para mostrar como interaccionan unos objetos con otros en un caso de uso o un escenario de ejecución.

**Figura 16. Ilustración de Diagrama de Secuencia**



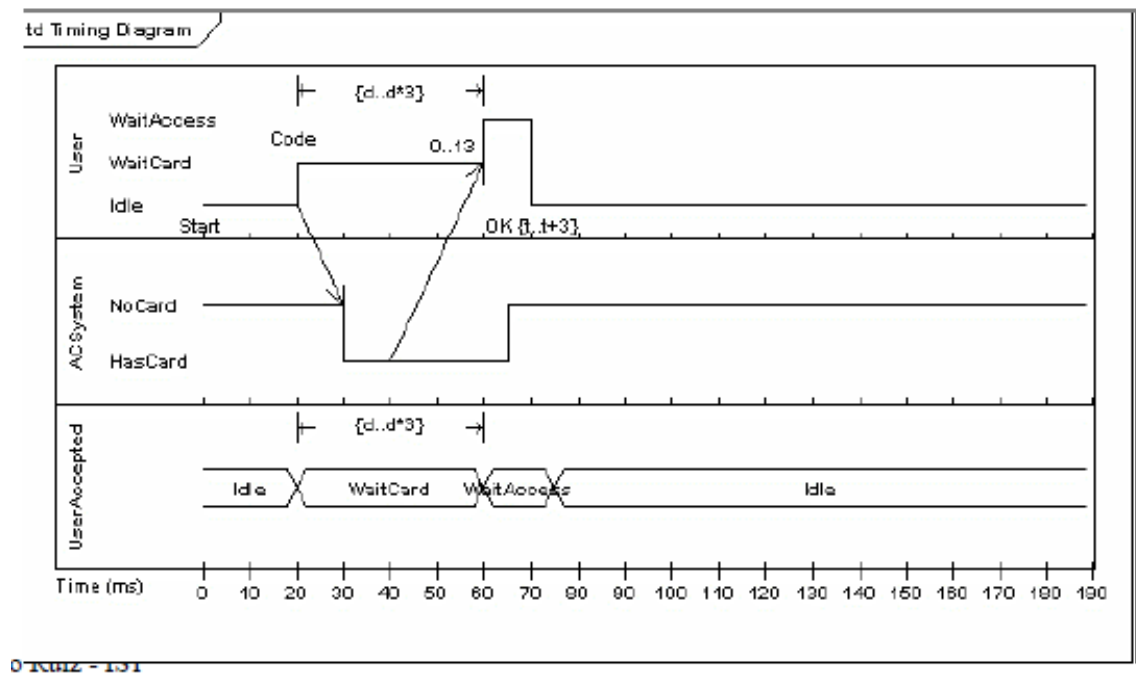
- **Diagramas de Comunicación (Colaboración).** Es un diagrama de interacción que resalta la organización estructural de los objetos o roles que envían y reciben mensajes en la interacción. Muestran un conjunto de objetos, enlaces entre ellos y los mensajes enviados y recibidos. Se usan para modelar el comportamiento dinámico de un caso de uso.

**Figura 17. Ilustración de Diagrama de Comunicación (Colaboración)**



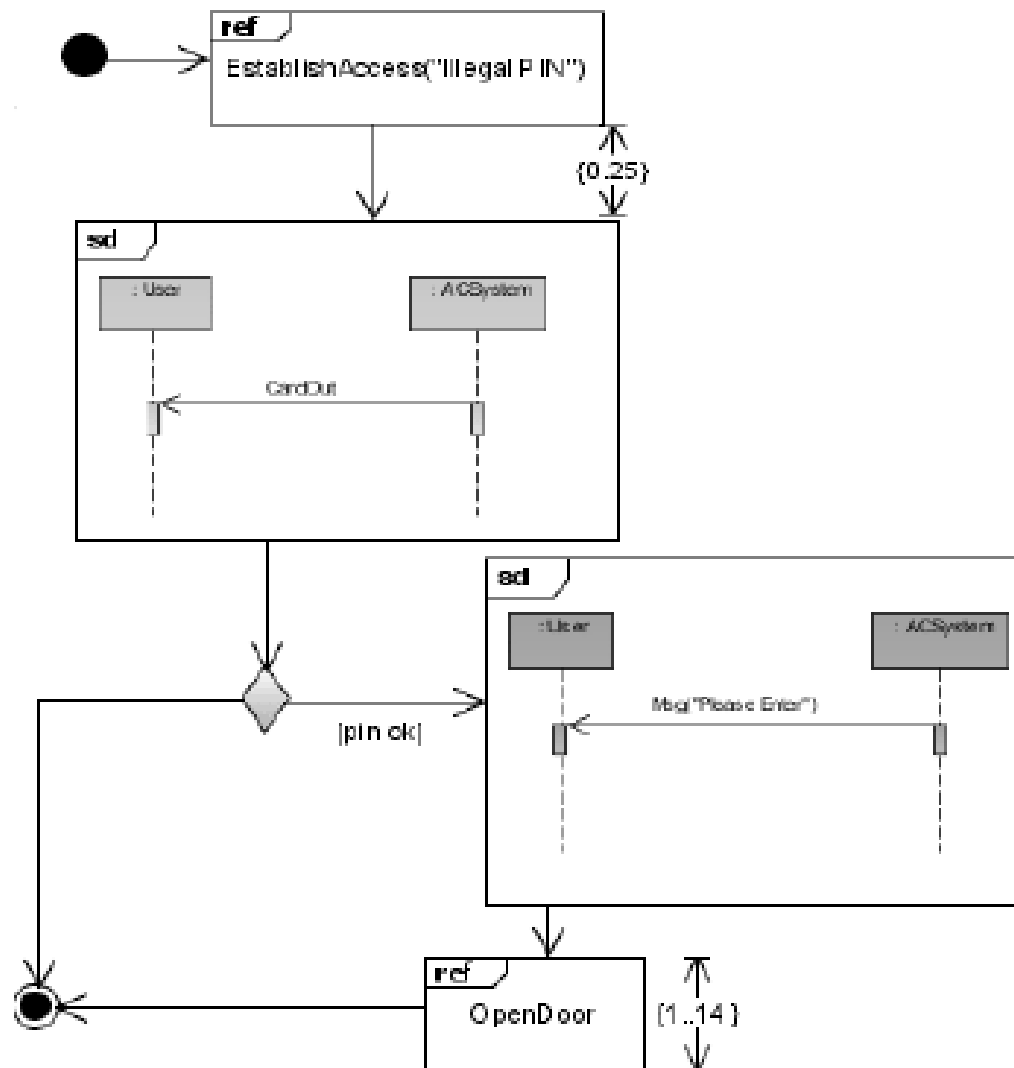
- **Diagramas de Tiempo.** Muestran los tiempos reales en la interacción entre diferentes objetos o roles. Comportamiento de los objetos en un periodo determinado de tiempo. Son una forma especial de diagramas de secuencia.

**Figura 18. Ilustración de Diagrama de Tiempo**



- **Diagramas de Revisión de Interacciones.** Aportan una visión general del flujo de control de las interacciones. Híbrido entre diagrama de actividad y diagrama de secuencia.

### Ilustración de Diagrama de Revisión de Interacciones



### 2.3. MARCO REFERENCIAL

El Área de Mantenimiento de Aplicaciones de la empresa TCS Ecuador, cuenta con el sistema informático UNICENTER, el cual es un sistema de proveedor externo. Tiene como objetivo principal dar seguimiento a toda la gestión de las ODT, desde la apertura o creación por solicitud de un usuario beneficiario, asignación a un desarrollador, transferencia a una de las áreas de soporte, registro de actividades, hasta la solución y cierre de la ODT.

UNICENTER es un sistema referencial para el desarrollo del presente proyecto unificando los conceptos de gestión de ODT y gestión de actividades de los desarrolladores.

## **CAPÍTULO III**

### **METODOLOGÍA DE LA INVESTIGACIÓN**



### 3.1. MATERIALES Y MÉTODOS

#### 3.1.1. Equipos y Materiales

Los materiales y recursos necesarios para el desarrollo del presente proyecto son los siguientes:

**Tabla 1. Equipos, Materiales y Recursos Humanos**

HARDWARE				
CANT.	DESCRIPCIÓN	CARACTERÍSTICAS	COSTO	TOTAL
1	<b>HP ProLiant ML110 G7/649255-205</b> – Servidor de Base de Datos y Publicación de Aplicación.	<ul style="list-style-type: none"> <li>• Procesador Intel® Quad Core Xeon® E3-1220 3.10GHz.</li> <li>• Tarjeta Madre Intel® C200 chipset.</li> <li>• Tarjeta de Red NC107i PCIe Gigabit NIC 10/100/1000.</li> <li>• Memoria 2 GB (1x2GB) PC3-10600E DDR3.</li> <li>• Disco Duro 500GB SATA.</li> <li>• Unidad Óptica HP.</li> <li>* El área de mantenimiento de aplicaciones cuenta con el equipo.</li> </ul>	0.00	0.00
1	<b>Toshiba Satellite A505-S6005</b> – Equipo para desarrollo del Sistema Informático.	<ul style="list-style-type: none"> <li>• Procesador Intel® Core I3 M 330 2.13GHz.</li> <li>• Tarjeta Madre TOSHIBA Satellite A505.</li> <li>• Memoria RAM 8GB DDR3 1333.</li> <li>• Disco Duro 500GB SATA.</li> <li>• Unidad Óptica.</li> </ul>	820.00	820.00
1	<b>Impresora Samsung ML-2165 Laser</b> – Impresora para impresión de documentación referente al proyecto y pruebas de impresión de reportes.	<ul style="list-style-type: none"> <li>• Hasta 20 ppm en A4.</li> <li>• Procesador de 300 MHz.</li> <li>• Pantalla de 2 LED (tóner, estado).</li> <li>• Interfaz USB 2.0 de alta velocidad.</li> <li>• Memoria de 8 MB.</li> <li>• Ciclo de carga de trabajo de hasta 10.000 páginas.</li> </ul>	91.84	91.84
<b>TOTAL</b>				<b>911.84</b>

SOFTWARE				
CANT.	DESCRIPCIÓN	DETALLE	COSTO	TOTAL
1	Microsoft Windows Server 2012	<ul style="list-style-type: none"> <li>• Sistema Operativo para servidores.</li> <li>* La empresa cuenta con licencias disponibles.</li> </ul>	0.00	0.00
4	Microsoft Windows 8 x64	<ul style="list-style-type: none"> <li>• Sistema Operativo para equipo de desarrollo.</li> <li>* La empresa cuenta con licencias disponibles.</li> </ul>	0.00	0.00
1	Microsoft SQL Server 2012 Enterprise	<ul style="list-style-type: none"> <li>• DBMS para el desarrollo del Back End del Sistema Informático.</li> <li>* La empresa cuenta con licencias disponibles.</li> </ul>	0.00	0.00
1	Microsoft Visual Studio 2012Ultimate	<ul style="list-style-type: none"> <li>• IDE para el desarrollo del Front End del Sistema Informático.</li> <li>* La empresa cuenta con licencias disponibles.</li> </ul>	0.00	0.00
1	Crystal Reports 2012	<ul style="list-style-type: none"> <li>• Software para el diseño de los reportes.</li> </ul>	0.00	0.00
1	DotNetBar for WinForms Single Developer v11.4	<ul style="list-style-type: none"> <li>• Componentes para desarrollo de interfaces del Front End del Sistema Informático.</li> </ul>	299.00	299.00
1	SysBrisa.Data.dll	<ul style="list-style-type: none"> <li>• DLL para acceso a datos.</li> </ul>	650.00	650.00
1	SysBrisa.ConfiguracionPC.dll	<ul style="list-style-type: none"> <li>• DLL para configuración de estaciones de trabajo.</li> </ul>	180.00	180.00
1	SysBrisa.Errores.dll	<ul style="list-style-type: none"> <li>• DLL para tratamiento de errores.</li> </ul>	180.00	180.00
1	SysBrisa.General.dll	<ul style="list-style-type: none"> <li>• DLL con clases generales para manejo de consultas y controles visuales.</li> </ul>	180.00	180.00
1	Spread for WinForms v5.	<ul style="list-style-type: none"> <li>• Componentes para desarrollo de interfaces del Front End del Sistema Informático.</li> </ul>	895.00	895.00
1	MZ-Tools for Visual Studio.NET v7.0	<ul style="list-style-type: none"> <li>• MZ-Tools es un Add-in para Visual Studio.NET. Incluye opciones que aceleran el proceso de desarrollo y gestión de proyectos.</li> </ul>	79.95	79.95
1	Microsoft Office 2013.	<ul style="list-style-type: none"> <li>• Paquete ofimático para redacción de documentación.</li> <li>* La empresa cuenta con licencias disponibles.</li> </ul>	0.00	0.00
TOTAL				2,463.95

<b>SUMINISTROS</b>				
<b>CANT.</b>	<b>DESCRIPCIÓN</b>	<b>DETALLE</b>	<b>COSTO</b>	<b>TOTAL</b>
5	Resmas de hojas A4.	• Impresión de documentación del proyecto y reportes de pruebas.	3.75	18.75
2	Lápices.	• Redacción.	0.75	1.50
2	Lapiceros.	• Redacción.	1.25	2.50
1	Cuaderno académico de 100 hojas.	• Redacción y notas varias.	3.80	3.80
2	Borradores.	• Redacción.	0.45	0.90
1	Pendrive 32Gb.	• Almacenamiento temporal de archivos.	31.36	31.36
<b>TOTAL</b>				<b>58.81</b>
<b>RECURSOS HUMANOS</b>				
<b>CANT.</b>	<b>DESCRIPCIÓN</b>	<b>DETALLE</b>	<b>COSTO</b>	<b>TOTAL</b>
4	Desarrollador (Orlando Briones Intriago).	• 4 meses de redacción, análisis, desarrollo y documentación del proyecto.	1,000.00	4,000.00
<b>TOTAL</b>				<b>4,000.00</b>
<b>PRESUPUESTO TOTAL</b>				<b>7,434.60</b>

Valores expresados en dólares americanos.

El Área de Mantenimiento de Aplicaciones provee las licencias correspondientes para la implementación del sistema informático.

Al tratarse de un trabajo de graduación, el desarrollador no recibe remuneración extra.

Los gastos de suministros de oficina corren por parte del investigador.

## 3.2. MÉTODOS

### 3.2.1. Método Inductivo

Este método científico a través de sus pasos esenciales: la observación de los hechos para su registro (procesos de administración del Área de Mantenimiento de Aplicaciones); la experimentación (Implementación del

sistema informático **SAGMA** para probar hipótesis), la comparación (sistema actual con metodología usada antes del sistema), la abstracción (estudiar aspectos comunes a varios procesos de la institución, excluyendo los demás) que permitirá llegar a una generalización que constituye una ley, norma o principio universalmente aceptado.

El uso de este método ayudó verificar que la investigación y desarrollo del sistema informático **SAGMA** es útil para el Área de Mantenimiento de Aplicaciones de la empresa TCS Ecuador.

### **3.2.2. Método Deductivo**

La sistematización de los procesos en la administración del Área de Mantenimiento de Aplicaciones de la empresa TCS Ecuador, tiende a mejorar la productividad de las empresas, permitiendo tener procesos eficientes, más confiables que mejore la calidad del servicio, de las que se extraen conclusiones en las que se aplican a casos particulares.

La presente investigación, mediante este método comprobó un efecto positivo en el Área de Mantenimiento de Aplicaciones de la Empresa TCS Ecuador tras la implantación del sistema informático **SAGMA**. La sistematización de procesos en la toma de decisiones para la asignación de ODT a los desarrolladores mejoró notablemente.

## **3.3. TIPOS DE INVESTIGACIÓN**

### **3.3.1. Metodología Cuasi-Experimental**

El diseño más habitual es el estudio antes-después o pre-post. En la investigación Cuasi-Experimental se identifica la variable independiente mediante la letra X. La variable dependiente es designada con la letra O. Su notación es: O1 X O2.

Para la presente investigación se consideró apropiado el uso del método Cuasi-Experimental, de un sólo grupo con medidas pretest y posttest, es decir, con una medición antes y después de implementar el sistema informático **SAGMA**.

Se identifica la variable independiente mediante la letra X, la variable dependiente con la letra O. Si existe una medida antes del tratamiento la denominaremos pretest, cuando la medida se realiza tras el tratamiento la denominamos posttest. Su notación es: **O1 X O2**.

Dónde:

**X = SAGMA.**

**O1** = Medición antes del experimento (Con los sistemas actuales).

**O2** = Medición después del experimento (Con el uso del sistema **SAGMA**).

En este diseño se verifica una observación (**O1**) antes de implantar el sistema informático **SAGMA** y otra después de aplicar la aplicación (**O2**). Las observaciones se consiguen a través de la aplicación de una prueba u observación descrita en el capítulo V.

### 3.4. TÉCNICAS DE INVESTIGACIÓN

#### 3.4.1. Población

La población objeto de la investigación está integrada por el personal que labora dentro del Área de Mantenimiento de Aplicaciones (126 personas, ver tabla de distribución por cargo).

**Tabla 2.** Personal que labora en el Área de Mantenimiento de Aplicaciones

<b>POBLACIÓN</b>	<b>N° PERSONAS.</b>
Analista / Desarrollador	115
Coordinador	7

Facilitador	3
Gerente de Área	1
<b>Total</b>	<b>126</b>

**Fuente:** Área de Mantenimiento de Aplicaciones

### 3.4.2. Muestra

Las encuestas se realizaron al 100% de la población, excepto a la población Analista / Desarrollador, para la cual se obtuvo el resultado de la muestra con la siguiente formula:

$$\frac{k^2 N p q}{e^2 (N - 1) + k^2 p q}$$

**n:** (?) Tamaño de la muestra para encuesta.

**N:** (126) tamaño de la población o universo.

**k:** (1.96) nivel de confianza.

**e:** (0.05) error muestral deseado.

**p:** (0.5) proporción de individuos que poseen en la población la característica de estudio.

**q:** (0.5) proporción de individuos que no poseen esa característica.

$$n = \frac{k^2 N \cdot p \cdot q}{e^2 (N - 1) + k^2 p \cdot q}$$

$$n = \frac{1.96^2 (115) (0.5) (0.5)}{(0.05^2) (115 - 1) + (1.96^2) (0.05) (0.05)}$$

$$n = \frac{110.446}{1.2454} = 88.68315401$$

### 3.4.3. Encuestas

Esta técnica permitió obtener información del personal del Área de Mantenimiento de Aplicaciones, mediante un cuestionario de preguntas, cuyas respuestas permitieron conocer la realidad de los diversos factores que inciden significativamente en rendimiento del Área y su gestión administrativa.

#### 3.4.3.1. Resultados de Encuestas aplicadas al personal con cargo de Analista / Desarrollador

La encuesta se aplicó a un total de 88 personas, escogidos al azar. El objetivo fue conocer su opinión y percepción en lo referente al manejo técnico y administrativo del Área de Mantenimiento de Aplicaciones.

Las preguntas y respuestas resultado de la encuesta son las siguientes:

- **Registro de actividades diarias**

**Objetivo:** Conocer la cultura de registro de actividades diarias del personal técnico.

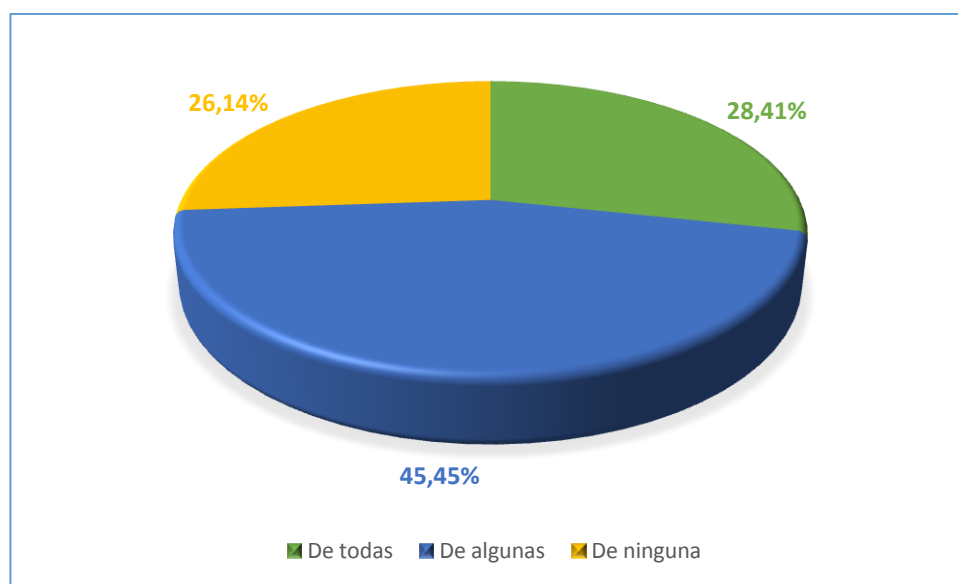
**Interpretación:** De la muestra observada, el 25% del personal técnico (Analistas / Desarrolladores) llevan registros de actividades diarias, el 40% registra eventualmente sus actividades y un 23% no realiza ningún registro. Esta situación impide conocer a las líneas de supervisión las actividades de sus subordinados.

**TABLA 3:** ¿Lleva un registro de las actividades diarias?

DETALLE	RESULTADOS	PORCENTAJE (%)
De todas	25	28.41
De algunas	40	45.45
De ninguna	23	26.14
<b>TOTAL</b>	<b>88</b>	<b>100</b>

**Fuente:** Encuestas

**Gráfico:** ¿Lleva un registro de las actividades diarias?



- **Medio en el que se llevan los registros**

**Objetivo:** Conocer los tipos de archivos en los que el personal técnico registra sus actividades.

**Interpretación:** De la muestra observada el 12.5% usa las herramientas existentes, el 68.18% ha diseñado su propio archivo digital (Microsoft Word, Microsoft Excel) y el 19.32% lleva registros manuales en cuadernos / libretas.

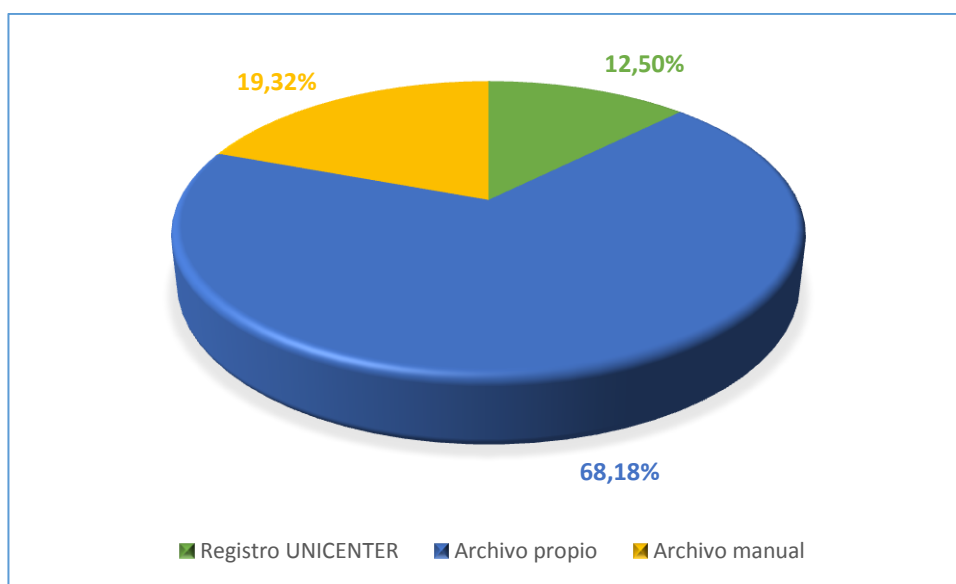
**TABLA 4:** ¿En que lleva los registros de actividades diarias?

DETALLE	RESULTADOS	PORCENTAJE (%)
Registro UNICENTER	11	12.50
Archivo propio	60	68.18
Archivo manual	17	19.32
<b>TOTAL</b>	<b>88</b>	<b>100</b>

**Fuente:** Encuestas



**Gráfico:** ¿En que lleva los registros de actividades diarias?



- **Asignación de ODT's**

**Objetivo:** Conocer la percepción referente a la asignación de ODT's al personal técnico por parte de los coordinadores.

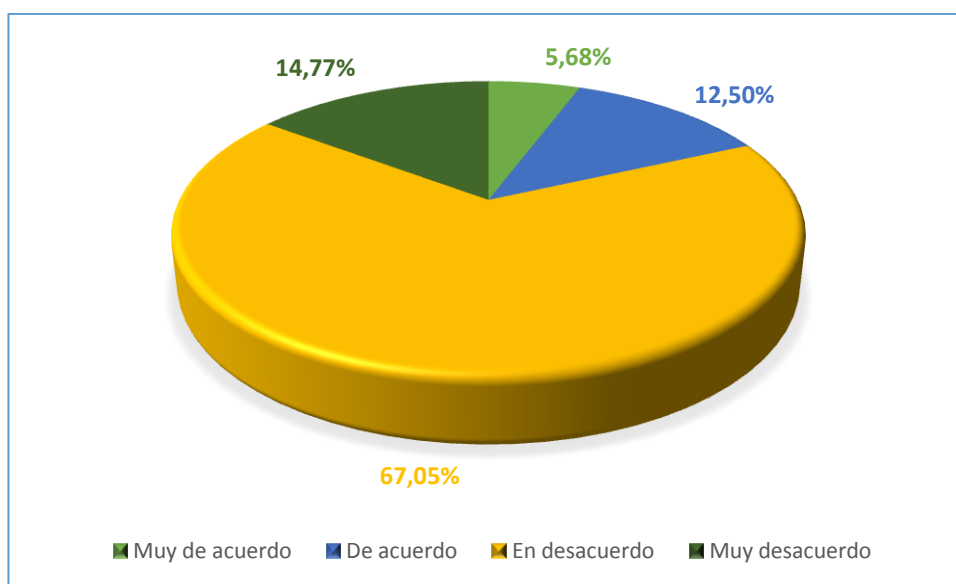
**Interpretación:** Más del 80% del personal técnico está en desacuerdo en lo referente a la asignación de ODT's, esta situación incide de forma negativa en el rendimiento general del área.

**TABLA 5:** ¿Considera acertada las asignaciones de ODT's por parte de los Coordinadores?

DETALLE	RESULTADOS	PORCENTAJE (%)
Muy de acuerdo	5	5.68
De acuerdo	11	12.50
En desacuerdo	59	67.05
Muy desacuerdo	13	14.77
<b>TOTAL</b>	<b>88</b>	<b>100</b>

**Fuente:** Encuestas

**Gráfico:** ¿Considera acertada las asignaciones de ODT's por parte de los Coordinadores?



- **Base de Conocimientos del Ecosistema Informático**

**Objetivo:** Obtener el grado de conocimiento del personal técnico acerca el ecosistema informático.

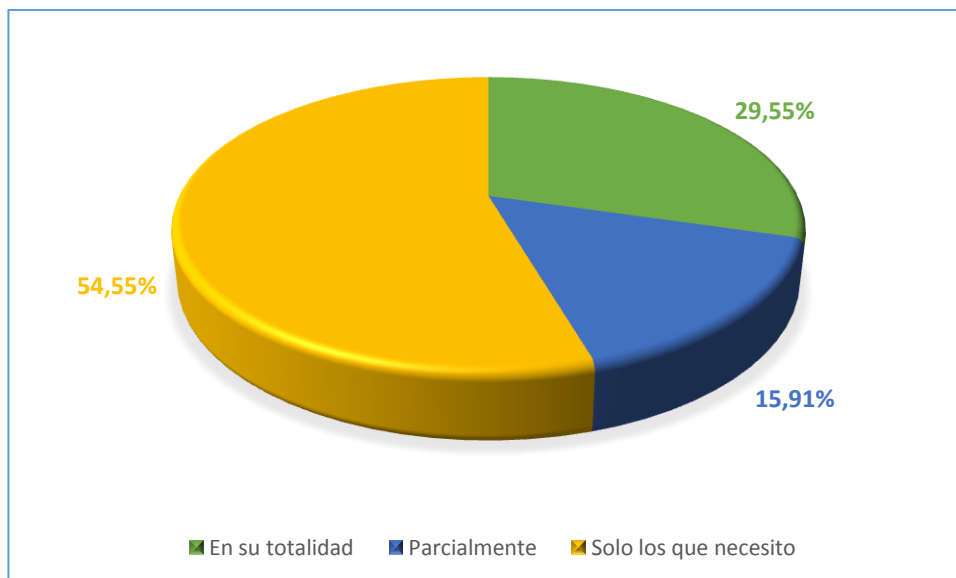
**Interpretación:** De la muestra observada, más del 50% del personal técnico solo conoce el ecosistema que necesita. Esta situación representa una desventaja y un riesgo en la atención de ODT's altamente prioritarias.

**TABLA 6:** ¿Conoce los sistemas, bases de datos, servidores a los cuales el Área de Mantenimiento de Aplicaciones da soporte?

DETALLE	RESULTADOS	PORCENTAJE (%)
En su totalidad	26	29.55
Parcialmente	14	15.91
Solo los que necesito	48	54.55
<b>TOTAL</b>	<b>88</b>	<b>100</b>

**Fuente:** Encuestas

**Gráfico:** ¿Conoce los sistemas, bases de datos, servidores a los cuales el Área de Mantenimiento de Aplicaciones da soporte?



- **Base de Conocimientos de Soluciones**

**Objetivo:** Conocer la necesidad de contar con una base de conocimientos para sustento y apoyo de soluciones en curso.

**Interpretación:** Más del 70% del personal técnico del Área de Mantenimiento de Aplicaciones cree que es altamente necesario contar con una base de conocimientos

**TABLA 7:** ¿Cree que es necesario contar con un registro histórico de soluciones?

DETALLE	RESULTADOS	PORCENTAJE (%)
Altamente necesario	63	71.59
Necesario	17	19.32
No es necesario	8	9.09
<b>TOTAL</b>	<b>88</b>	<b>100</b>

**Fuente:** Encuestas

**Gráfico:** ¿Cree que es necesario contar con un registro histórico de soluciones?



- **Implementación de un Sistema Informático en el Área**

**Objetivo:** Conocer la necesidad de automatizar la gestión, coordinación y administración del Área de Mantenimiento de Aplicaciones mediante la implantación de un Sistema Informático.

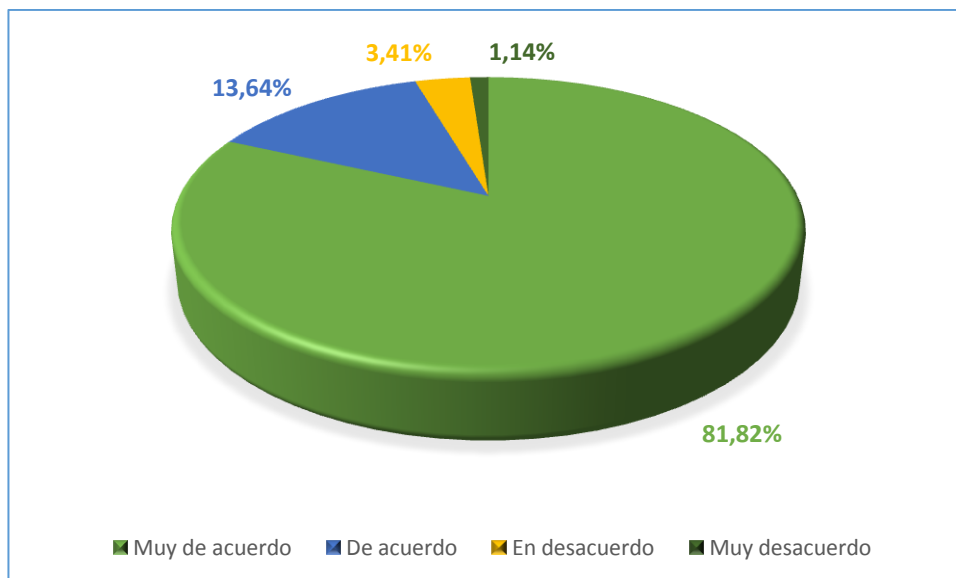
**Interpretación:** El 80% del personal técnico encuestado manifestó la necesidad y utilidad de un sistema informático para el soporte de la gestión en el Área de Mantenimiento de Aplicaciones.

**TABLA 8:** ¿Cree usted que la implementación de un sistema informático especializado sería útil para mejorar los procesos del Área de Mantenimiento de Aplicaciones?

DETALLE	RESULTADOS	PORCENTAJE (%)
Muy de acuerdo	72	81.82
De acuerdo	12	13.64
En desacuerdo	3	3.41
Muy desacuerdo	1	1.14
<b>TOTAL</b>	<b>88</b>	<b>100</b>

**Fuente:** Encuestas

**Gráfico:** ¿Cree usted que la implementación de un sistema informático especializado sería útil para mejorar los procesos del Área de Mantenimiento de Aplicaciones?



#### 3.4.3.2. Resultados de Encuestas aplicadas al personal con cargo de Coordinador

La encuesta se aplicó a un total de 7 personas, que representan el 100% de los coordinadores. El objetivo fue conocer su opinión y percepción en lo referente a la coordinación del Área de Mantenimiento de Aplicaciones.

Las preguntas y respuestas resultado de la encuesta son las siguientes:

- **Carga de trabajo**

**Objetivo:** Conocer la gestión de los coordinadores en referencia a la administración de carga de trabajo de sus subordinados.

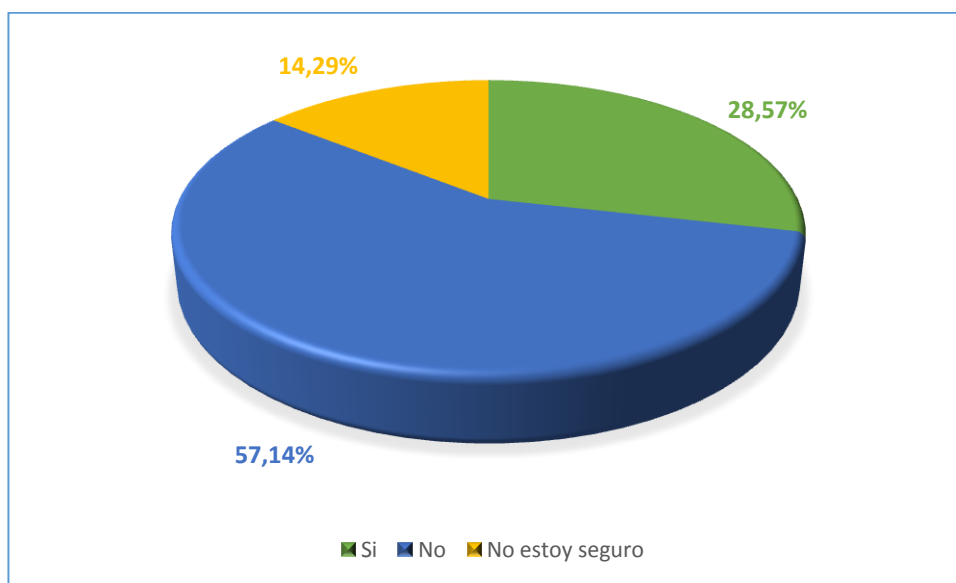
**Interpretación:** Más del 50% de los coordinadores desconoce la carga de trabajo real de sus subordinados (personal técnico), lo cual, es altamente preocupante. Esta situación provoca la asignación no equilibrada de trabajo.

**TABLA 9:** ¿Conoce la capacidad de atención y carga de trabajo de sus subordinados?

DETALLE	RESULTADOS	PORCENTAJE (%)
Si	2	28.57
No	4	57.14
No estoy seguro	1	14.29
<b>TOTAL</b>	<b>7</b>	<b>100</b>

**Fuente:** Encuestas

**Gráfico:** ¿Conoce la capacidad de atención y carga de trabajo de sus subordinados?



- **Incumplimiento de trabajo**

**Objetivo:** Conocer el control y prevención de incumplimiento en los tiempos de entrega de las ODT's.

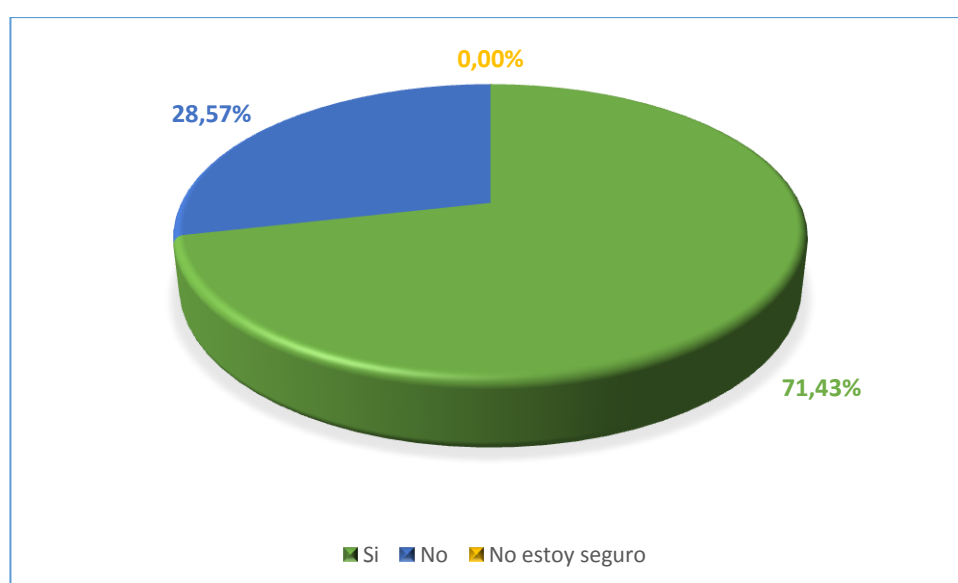
**Interpretación:** Más de la mitad de los coordinadores lleva algún tipo de registro y control sobre los tiempos de entrega, lo cual es una fortaleza en el área. Sin embargo el 28.57% no tiene un control sobre esta actividad, generando incumplimientos en las metas y objetivos del área.

**TABLA 10:** ¿Lleva un registro sobre el incumplimiento en el tiempo de entrega de las soluciones por parte de sus subordinados?

DETALLE	RESULTADOS	PORCENTAJE (%)
Si	5	71.43
No	2	28.57
No estoy seguro	0	0.00
<b>TOTAL</b>	<b>7</b>	<b>100</b>

**Fuente:** Encuestas

**Gráfico:** ¿Lleva un registro sobre el incumplimiento en el tiempo de entrega de las soluciones por parte de sus subordinados?



- **Administración de trabajo**

**Objetivo:** Conocer la administración por parte de los coordinadores sobre el trabajo que desempeña el personal técnico.

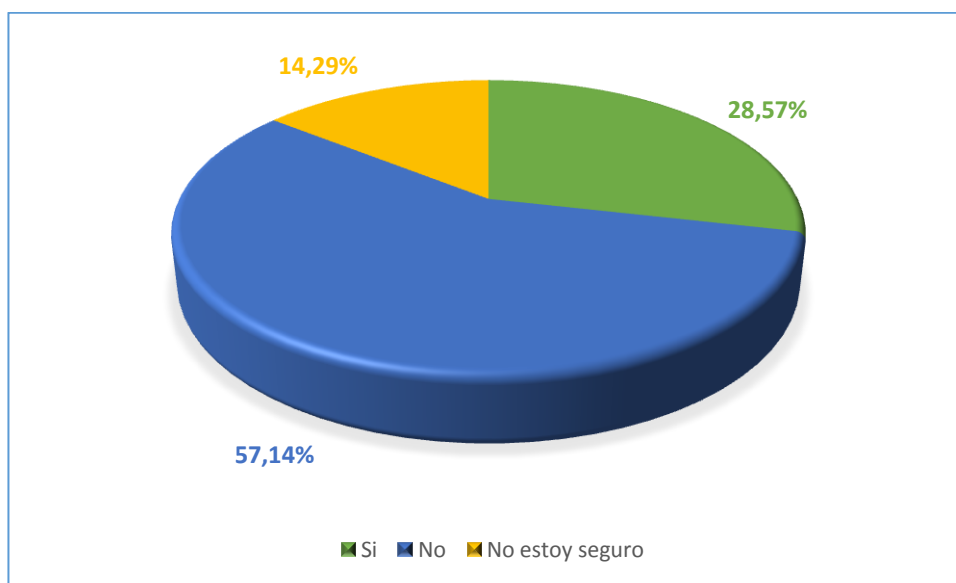
**Interpretación:** El 57.14% de los coordinadores no conoce la distribución de las actividades de su personal técnico, esta situación impide generar estadísticas de atención y control oportuno de las ODT's.

**TABLA 11:** ¿Conoce las actividades en las que sus subordinados distribuyen el tiempo?

DETALLE	RESULTADOS	PORCENTAJE (%)
Si	2	28.57
No	4	57.14
No estoy seguro	1	14.29
<b>TOTAL</b>	<b>7</b>	<b>100</b>

**Fuente:** Encuestas

**Gráfico:** ¿Conoce las actividades en las que sus subordinados distribuyen el tiempo?



- **Implementación de un Sistema Informático en el Área**

**Objetivo:** Conocer la necesidad de automatizar la gestión, coordinación y administración del Área de Mantenimiento de Aplicaciones mediante la implantación de un Sistema Informático.

**Interpretación:** El 85.71% del personal encuestado manifestó la necesidad y utilidad de un sistema informático para el soporte de la gestión en el Área de Mantenimiento de Aplicaciones.

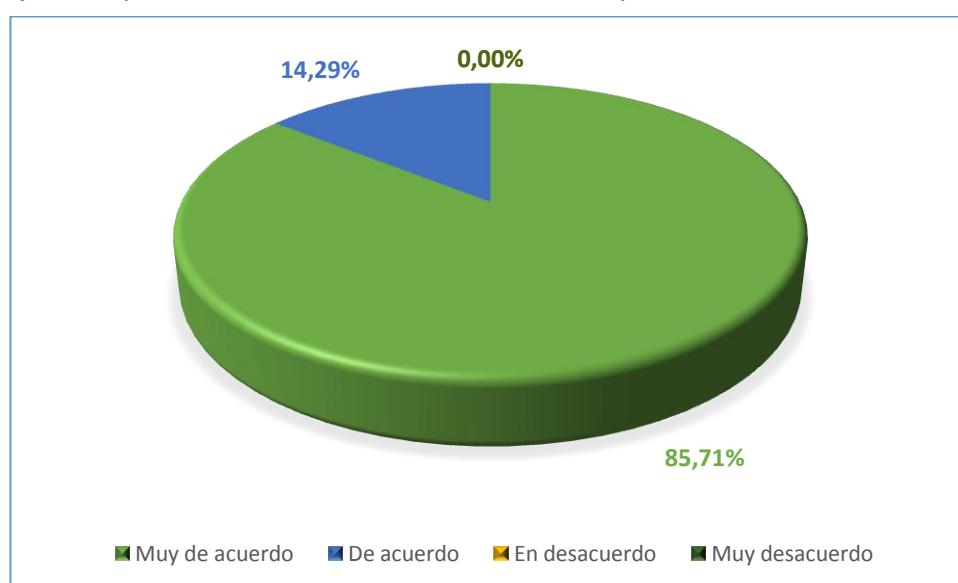


**TABLA 12:** ¿Cree usted que la implementación de un sistema informático especializado sería útil para mejorar los procesos del Área de Mantenimiento de Aplicaciones?

DETALLE	RESULTADOS	PORCENTAJE (%)
Muy de acuerdo	6	85.71
De acuerdo	1	14.29
En desacuerdo	0	0.00
Muy desacuerdo	0	0.00
<b>TOTAL</b>	<b>7</b>	<b>100</b>

**Fuente:** Encuestas

**Gráfico:** ¿Cree usted que la implementación de un sistema informático especializado sería útil para mejorar los procesos del Área de Mantenimiento de Aplicaciones?



### 3.4.3.3. Resultados de Encuestas aplicadas al personal con cargo de Facilitador / Gerente de Área

La encuesta se aplicó a un total de 4 personas, que representan el 100% del personal administrativo. El objetivo fue conocer su opinión y percepción en lo referente a la administración del Área de Mantenimiento de Aplicaciones.

Las preguntas y respuestas resultado de la encuesta son las siguientes:

- **Capacidad de atención del Área de Mantenimiento de Aplicaciones**

**Objetivo:** Conocer la percepción del área administrativa referente a la capacidad de atención de ODT's.

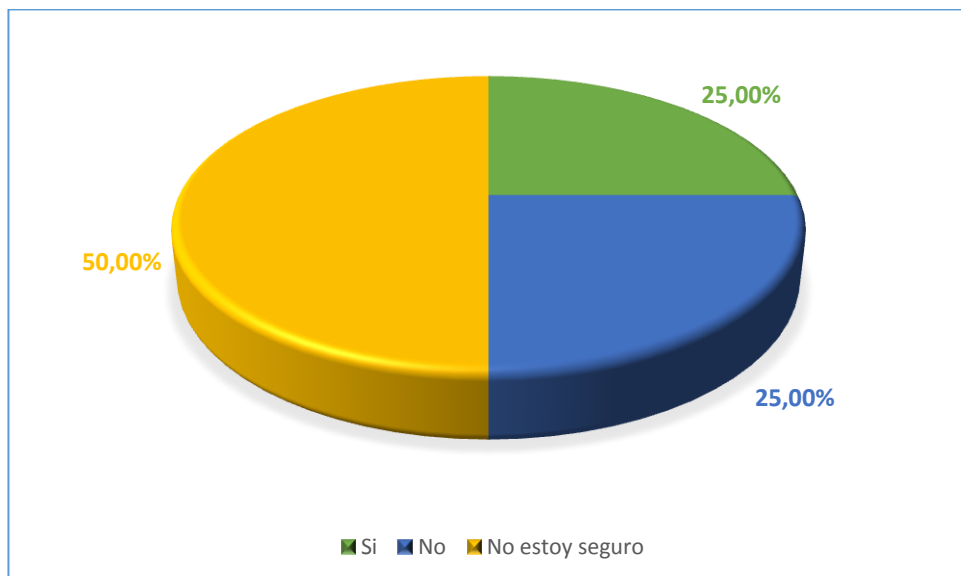
**Interpretación:** El 50% del personal encuestado indicó que no está seguro de la capacidad de resolución de ODT's del Área de Mantenimiento de Aplicaciones. Esta situación es una debilidad del área y una desventaja en el cumplimiento de los objetivos.

**TABLA 13:** ¿Cree usted que el Área de Mantenimiento de Aplicaciones tiene la capacidad suficiente de atención?

DETALLE	RESULTADOS	PORCENTAJE (%)
Si	1	25.00
No	1	25.00
No estoy seguro	2	50.00
<b>TOTAL</b>	<b>4</b>	<b>100</b>

**Fuente:** Encuestas

**Gráfico:** ¿Cree usted que el Área de Mantenimiento de Aplicaciones tiene la capacidad suficiente de atención?



- **Carga horaria excedente del personal del Área de Mantenimiento de Aplicaciones**

**Objetivo:** Conocer la gestión y administración de la carga horaria excedente del área.

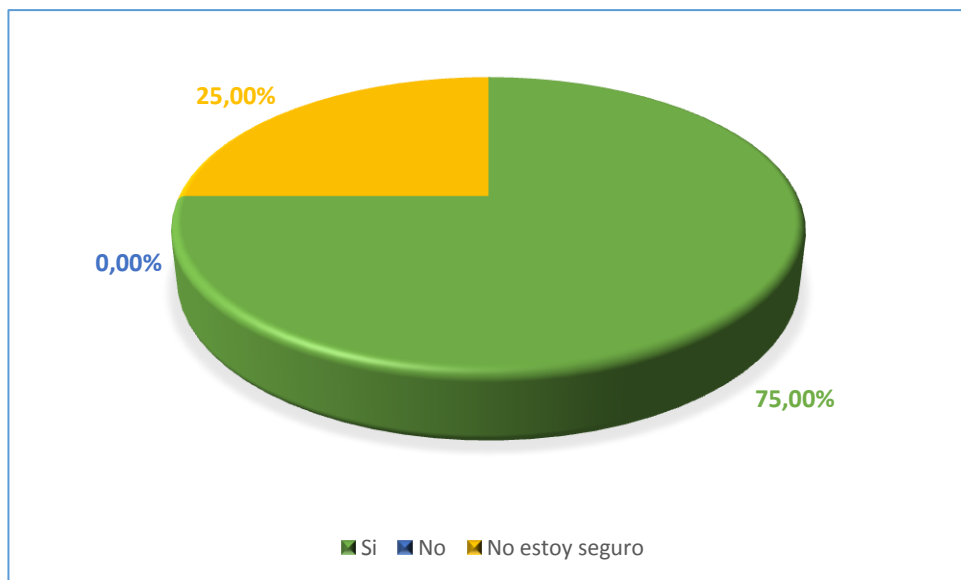
**Interpretación:** El 25% del personal encuestado no está seguro de conocer la carga horaria excedente en el personal técnico, lo cual dificulta la compensación a los desarrolladores, generando descontento y malestar.

**TABLA 14:** ¿Conoce el tiempo extra de trabajo del área en general?

DETALLE	RESULTADOS	PORCENTAJE (%)
Si	3	75.00
No	0	0.00
No estoy seguro	1	25.00
<b>TOTAL</b>	<b>4</b>	<b>100</b>

**Fuente:** Encuestas

**Gráfico:** ¿Conoce el tiempo extra de trabajo del área en general?



- **Comunicación con el cliente**

**Objetivo:** Conocer la disponibilidad de recursos que faciliten la comunicación directa entre el resolutor (personal técnico) y el beneficiario de la ODT.

**Interpretación:** el 75% del área de administración indica la necesidad de contar con un registro de usuarios para facilitar la comunicación directa entre el personal técnico y los beneficiarios de la ODT.

**TABLA 15:** ¿Considera necesario que el Área de Mantenimiento de Aplicaciones cuente con un registro de usuarios por cliente, en el que conste información relevante, en caso de requerir comunicación directa?

DETALLE	RESULTADOS	PORCENTAJE (%)
Altamente necesario	3	75.00
Necesario	0	0.00
No es necesario	1	25.00
<b>TOTAL</b>	<b>4</b>	<b>100</b>

**Fuente:** Encuestas

**Gráfico:** ¿Considera necesario que el Área de Mantenimiento de Aplicaciones cuente con un registro de usuarios por cliente, en el que conste información relevante, en caso de requerir comunicación directa?



- **Análisis estadístico del Área de Mantenimiento de Aplicaciones**

**Objetivo:** Conocer la necesidad de análisis de las diferentes situaciones del Área de Mantenimiento de Aplicaciones.

**Interpretación:** El 100% del personal administrativo considera altamente necesario la medición del rendimiento del área en general. Esta situación

genera oportunidades de mejora en la resolución de ODT's y efectividad en la toma de decisiones.

**TABLA 16:** ¿Cree necesario que el área de Mantenimiento de Aplicaciones cuente con información estadística para determinar el rendimiento general del área?

DETALLE	RESULTADOS	PORCENTAJE (%)
Altamente necesario	4	100.00
Necesario	0	0.00
No es necesario	0	0.00
<b>TOTAL</b>	<b>4</b>	<b>100</b>

**Fuente:** Encuestas

**Gráfico:** ¿Cree necesario que el área de Mantenimiento de Aplicaciones cuente con información estadística para determinar el rendimiento general del área?



- **Implementación de un Sistema Informático en el Área**

**Objetivo:** Conocer la necesidad de automatizar la gestión, coordinación y administración del Área de Mantenimiento de Aplicaciones mediante la implantación de un Sistema Informático.

**Interpretación:** El 100% del personal encuestado manifestó la necesidad y utilidad de un sistema informático para el soporte de la gestión en el Área de Mantenimiento de Aplicaciones.

**TABLA 17:** ¿Cree usted que la implementación de un sistema informático especializado sería útil para mejorar los procesos del Área de Mantenimiento de Aplicaciones?

DETALLE	RESULTADOS	PORCENTAJE (%)
Altamente necesario	3	75.00
Necesario	1	25.00
No es necesario	0	0.00
<b>TOTAL</b>	<b>4</b>	<b>100</b>

**Fuente:** Encuestas

**Gráfico:** ¿Cree usted que la implementación de un sistema informático especializado sería útil para mejorar los procesos del Área de Mantenimiento de Aplicaciones?



#### 3.4.4. Entrevista

Mediante entrevistas realizadas al personal técnico y administrativo del Área de Mantenimiento de Aplicaciones, se obtuvo información relevante sobre la dirección del área. Esta técnica tiene la finalidad de identificar los requerimientos necesarios para el desarrollo del sistema informático.

**3.4.4.1. Entrevista realizada al Ing. Pablo Córdova (Desarrollador Sénior, 6 años en el Área de Mantenimiento de Aplicaciones) y al Ing. Danilo Oña (Desarrollador Junior, 45 días en el Área de Mantenimiento de Aplicaciones)**

- **¿Cuál es el proceso que realiza cuando su coordinador le asigna una ODT?**

**Ing. Pablo Córdova:** En base a mi experiencia, determino si el requerimiento del usuario, beneficiario de la ODT requiere atención tecnológica o de gestión. En caso de requerir atención tecnológica, se valida si los datos de la ODT (Aplicación y Descripción) son los correctos, caso contrario me comunicó telefónicamente o vía email con el beneficiario.

Una vez que se obtiene todos los datos relacionados al error o solicitud, se procede con la estimación de tiempos de resolución y probables actividades a realizar.

Eventualmente reviso documentación de ODT's atendidas y relacionadas.

Una vez atendido el requerimiento, procedo a registrar los tiempos por cada actividad realizada.

**Ing. Danilo Oña:** Actualmente no cuento con la experiencia suficiente, por lo que suelo hacer consultas a los desarrolladores con más experiencia.

En ocasiones me asignan ODT's y tardo mucho tiempo en resolverlas, esto se debe a que no conozco las aplicaciones y los servidores en los cuales están publicados los fuentes.

#### **3.4.4.2. Entrevista realizada a Coordinadores del Área de Mantenimiento de Aplicaciones con 3 años de experiencia en el cargo**

- **¿Cuál es el proceso que realiza para asignar una ODT a un desarrollador, y el seguimiento o avance de la resolución?**

Generalmente las ODT's se asignan previa validación de la carga de trabajo del personal disponible. En ocasiones los desarrolladores a mi cargo no cuentan con un registro de ODT's asignadas y sus respectivos avances, esta situación dificulta la asignación correcta y balanceada de los errores o soluciones.

En cuanto al seguimiento se solicita un informe periódico a cada desarrollador en un archivo de Excel, en este archivo se solicita que se categorice las actividades realizadas y el tiempo que le tomó realizar dicha actividad.

#### **3.4.4.3. Entrevista realizada a Facilitadores del Área de Mantenimiento de Aplicaciones con 4 años de experiencia en el cargo**

- **¿Cuánta con algún tipo de registro de rendimiento de sus subordinados?**

El rendimiento de los grupos resolutores bajo mi cargo es medido en base a las actividades realizadas en el día a día, mis líneas de supervisión requieren ciertos tipos de reportes como horas extras, actividades realizadas, requerimientos generados y atendidos por grupos o áreas, entre otros.

La efectividad de cada desarrollador se mide en base a las ODT's asignadas y las ODT's resueltas en el mes en curso.



## **CAPÍTULO IV**

### **DESARROLLO DEL SISTEMA**

## 4.1. FACTIBILIDAD

Una vez definida la problemática y las causas que influyen en el desarrollo de un sistema informático, previo análisis mediante técnicas de investigación, es pertinente realizar un estudio de factibilidad que determine la infraestructura tecnológica, factores operacionales y económicos que implica la implantación del sistema en el Área de Mantenimiento de Aplicaciones.

### 4.1.1. Factibilidad Técnica

Debido a los recursos tanto de hardware como de software con que cuenta la empresa TCS Ecuador, es posible desarrollar el sistema sin mayores dificultades, sin la adquisición de equipos o aplicaciones adicionales.

A continuación se indican los equipos y aplicaciones existentes para poner en marcha el sistema informático.

- **Hardware y Software**

DESCRIPCIÓN	CARACTERÍSTICAS
<b>HP ProLiant ML110 G7/649255-205</b> – Servidor de Base de Datos y Publicación de Aplicación.	<ul style="list-style-type: none"><li>• Procesador Intel® Quad Core Xeon® E3-1220 3.10GHz.</li><li>• Tarjeta Madre Intel® C200 chipset.</li><li>• Tarjeta de Red NC107i PCIe Gigabit NIC 10/100/1000.</li><li>• Memoria 2 GB (1x2GB) PC3-10600E DDR3.</li><li>• Disco Duro 500GB SATA.</li><li>• Unidad Óptica HP.</li><li>• Microsoft Windows Server 2012.</li><li>• Framework v4.5.</li><li>• Microsoft SQL Server 2012 Enterprise.</li></ul>

#### **4.1.2. Factibilidad Operacional**

La empresa TCS Ecuador acuerda colaborar con el desarrollo del sistema informático en el Área de Mantenimiento de Aplicaciones debido al gran aporte que el presente proyecto brindaría a la gestión técnica y administrativa, dicho acuerdo consiste en permitir realizar investigaciones, encuestas y documentación de procesos.

#### **4.1.3. Factibilidad Económica**

El costo de licencias de software, elementos hardware y personal técnico necesarios para el desarrollo del proyecto son sustentables para la empresa TCS Ecuador desde el punto de vista costo-beneficio.

El financiamiento de los costos correspondientes a suministros utilizados para el desarrollo del proyecto será asumido por su autor.

La implementación del sistema informático ofrece beneficios económicos a la empresa TCS Ecuador mediante el control del incumplimiento del SLA firmados con los clientes.

### **4.2. METODOLOGÍA DE DESARROLLO DE SOFTWARE**

#### **4.2.1. Metodologías Ágiles – RAD**

La metodología empleada para el desarrollo del sistema informático es RAD (Rapid Application Development RAD), para un desarrollo rápido utilizando el enfoque de construcción basado en componentes, reutilizando código ya existente.

Para el desarrollo del sistema informático **SAGMA** se utilizará las siguientes herramientas software:

- Gestión de Base de Datos Relacional: Microsoft SQL Server 2012.
- IDE para el desarrollo de Aplicación de Escritorio: Microsoft Visual Studio 2012 Ultimate.
- Lenguaje de Programación: Visual Basic.NET.
- Suite de Controles y Componentes: Spread for Net, DotNetBar, Paquete de DLL SysBRISA.

#### **4.2.2. Análisis**

##### **4.2.2.1. Especificación de requerimientos**

Para la especificación de requerimientos se empleará la Norma IEEE-830, la cual permitirá establecer las funciones del sistema informático a implementar basándose en encuestas y entrevistas a usuarios claves del Área de Mantenimiento de Aplicaciones.

###### **4.2.2.1.1. Propósito**

El presente documento se toma como referencia para la especificación de requerimientos y forma parte de la documentación oficial del sistema informático **SAGMA**, para la gestión, coordinación y administración de proyectos de desarrollo y mantenimiento de software de la empresa TCS Ecuador. Además de servir como una guía para los desarrolladores, administradores y usuarios en general.

###### **4.2.2.1.2. Alcance**

**SAGMA** será el sistema informático que se desarrollará para el Área de Mantenimiento de Aplicaciones de la empresa TCS Ecuador, con la finalidad de

mejorar y optimizar la atención y resolución de errores y solicitudes de los clientes.

**SAGMA** dará soporte al personal técnico del área, en lo referente a la resolución de ODT's, mediante el uso de información histórica y gestión de actividades categorizadas, optimizando el tiempo de desarrollo e investigación mediante el uso de una base de conocimientos.

**SAGMA** permitirá optimizar las actividades de coordinación del área, en lo referente a la creación y asignación de ODT's, seguimientos a las actividades del personal técnico, seguimiento a incumplimientos de SLA.

**SAGMA** dará soporte analítico y estadístico al personal administrativo del área para mejorar la toma de decisiones.

Principales funcionalidades del sistema informático **SAGMA**:

- Gestión de Recursos de la Empresa: Departamentos, Áreas, Grupos, Cargos, Talento Humano.
- Gestión de Clientes: Mantenimiento, Áreas de Soporte, Usuarios (Beneficiarios y Autorizadores).
- Gestión del Ecosistema Informático: Servidores, Base de Datos, Aplicaciones y Procesos.
- Gestión de ODT's (Errores y Solicitudes): Categorización, Creación, Seguimientos.
- Gestión de Actividades: Registro de Actividades del personal del área.
- Reportes Analíticos y Estadísticos.

#### **4.2.2.1.3. Referencias**

- ANSI/IEEE 830: Guía de IEEE para la especificación de requerimientos software.
- Encuestas y entrevistas al personal técnico y administrativo del Área de Mantenimiento de Aplicaciones.

#### **4.2.2.1.4. Descripción General del Sistema Informático SAGMA**

- **Perspectiva del Software**

**SAGMA**, como un sistema informático, incrementará la eficiencia y eficacia del Área de Mantenimiento de Aplicaciones de la empresa TCS Ecuador, mediante el soporte a las actividades de todo el personal técnico y administrativo en lo referente a solución de ODT's.

- **Funciones del Software**

✓ **Gestión de Recursos de la Empresa: Departamentos, Áreas, Grupos, Cargos, Talento Humano.**

Módulo del sistema diseñado para la administración de los recursos de la empresa. Diseñado para la creación y mantenimiento de áreas, roles y talento humano. El software deberá administrar datos personales y laborales de los empleados.

✓ **Gestión de Clientes: Mantenimiento, Áreas de Soporte, Usuarios (Beneficiarios y Autorizadores).**

Módulo del sistema diseñado para la administración de clientes. Diseñado para la creación y mantenimiento de clientes, áreas de soporte y usuarios.

✓ **Gestión del Ecosistema Informático: Servidores, Base de Datos, Aplicaciones y Procesos.**

Módulo del sistema diseñado para la administración del ecosistema informático de clientes. Diseñado para la creación y mantenimiento de servidores, bases de datos, aplicaciones y procesos.

✓ **Gestión de ODT's (Errores y Solicitudes): Categorización, Creación, Seguimientos.**

Módulo del sistema diseñado para la administración y gestión de requerimientos (ODT's) de tipo errores y solicitudes. El módulo permitirá el análisis previa asignación de ODT a personal técnico.

✓ **Gestión de Actividades: Registro de Actividades del personal del área.**

Módulo del sistema diseñado para el registro y gestión de actividades del personal del área. Permitirá la categorización de las actividades.

✓ **Reportes Analíticos y Estadísticos.**

Módulo del sistema diseñado para la toma de decisiones y generación de informes analíticos y estadísticos.

#### **4.2.2.1.5. Requerimientos específicos funcionales**

- **Gestión de Recursos de la Empresa: Departamentos, Áreas, Grupos, Cargos, Talento Humano.**

**Introducción:** **SAGMA** permite el ingreso de datos necesarios de los elementos de la Empresa.

**Entrada:** Se ingresará los datos necesarios de cada recurso de la Empresa.

**Proceso:** Al crear un nuevo recurso de la Empresa, el sistema deberá validar que se especifiquen los campos necesarios y que no existan registros duplicados.

**Salida:** Datos actualizados en la Base de Datos.

- **Gestión de Clientes: Mantenimiento, Áreas de Soporte, Usuarios (Beneficiarios y Autorizadores).**

**Introducción:** **SAGMA** permite el ingreso de datos necesarios de los elementos de los Clientes.

**Entrada:** Se ingresará los datos necesarios de cada recurso de los Clientes.

**Proceso:** Al crear un nuevo recurso de los Clientes, el sistema deberá validar que se especifiquen los campos necesarios y que no existan registros duplicados.

**Salida:** Datos actualizados en la Base de Datos.

- **Gestión del Ecosistema Informático: Servidores, Base de Datos, Aplicaciones y Procesos.**

**Introducción:** **SAGMA** permite el ingreso de datos necesarios de los elementos del Ecosistema Informático.

**Entrada:** Se ingresará los datos necesarios de cada uno de los elementos del Ecosistema Informático.

**Proceso:** Al crear un nuevo elemento del Ecosistema Informático, el sistema deberá validar que se especifiquen los campos necesarios y que no existan registros duplicados.



**Salida:** Datos actualizados en la Base de Datos.

- **Gestión de ODT's (Errores y Solicitudes): Categorización, Creación, Seguimientos.**

**Introducción:** **SAGMA** permite el ingreso de datos necesarios referente a las Órdenes de Trabajo (ODT).

**Entrada:** Se ingresará los datos necesarios de cada uno de los elementos de las Órdenes de Trabajo (ODT).

**Proceso:** Al crear un nuevo elemento de la Orden de Trabajo (ODT), el sistema deberá validar que se especifiquen los campos necesarios y que no existan registros duplicados.

**Salida:** Datos actualizados en la Base de Datos.

- **Gestión de Actividades: Registro de Actividades del personal del área.**

**Introducción:** **SAGMA** permite el ingreso de datos necesarios referente a las Actividades de los recursos del Área de Mantenimiento de Aplicaciones.

**Entrada:** Se ingresará los datos necesarios de cada actividad de los recursos del Área de Mantenimiento de Aplicaciones.

**Proceso:** Al crear una nueva actividad, el sistema deberá validar que se especifiquen los campos necesarios y que no existan registros duplicados.

**Salida:** Datos actualizados en la Base de Datos.

- **Reportes Analíticos y Estadísticos.**

**Introducción:** **SAGMA** permite el ingreso de parámetros necesarios para generar reportes e informes Analíticos y Estadísticos del Área de Mantenimiento de Aplicaciones.

**Entrada:** Se ingresará los parámetros necesarios de cada Reporte e Informe.

**Proceso:** Al ingresar los parámetros, el sistema deberá validar que se especifiquen de manera correcta.

**Salida:** Presentación de Reportes e Informes en pantalla de usuario.

#### **4.2.2.1.6. Casos de Uso**

Dada la misma estructura y funcionamiento de los siguientes formularios (ventanas de trabajo de usuario) en las acciones de Registrar, Buscar, Actualizar, Eliminar; se definen los siguientes casos de uso modelo.

- **Gestión de Recursos de la Empresa:** Departamento, Área, Grupo, Cargo, Recursos (Talento Humano).

- ✓ **Departamento:** Se carga el cuadro de lista Estado.
- ✓ **Área:** Se cargan los cuadros de lista Departamento y Estado.
- ✓ **Grupo:** Se cargan los cuadros de lista Departamento, Área y Estado.
- ✓ **Cargo:** Se carga el cuadro de lista Estado.
- ✓ **Recursos:** Se cargan los cuadros de lista Estado, Departamento, Área, Grupo y Cargo.

- **Gestión de Recursos de Clientes:** Cliente, Área (Áreas de Soporte), Usuario (Beneficiarios y Autorizadores).

- ✓ **Cliente:** Se carga el cuadro de lista Estado.
- ✓ **Área:** Se cargan los cuadros de lista Cliente y Estado.
- ✓ **Usuario:** Se cargan los cuadros de lista Cliente, Área y Estado.

- **Gestión del Ecosistema Informático:** Servidores, Base de Datos, Aplicaciones y Procesos.

- ✓ **Servidores:** Se cargan los cuadros de lista Sistema Operativo, Arquitectura, Tipo y Estado.
- ✓ **Base de Datos:** Se cargan los cuadros de lista Cliente, Tipo, Estado, Servidor de Desarrollo, Servidor de Test, Servidor de Producción.
- ✓ **Aplicaciones:** Se cargan los cuadros de lista Tipo, Estado, Cliente, Base de Datos, Servidor de Desarrollo, Servidor de Test, Servidor de Producción.
- ✓ **Procesos:** Se cargan los cuadros de lista Responsable, Tipo, Calendarización y Estado.

- **Gestión de ODT's (Errores y Solicitudes):** Categoría de ODT, Efectos, Tipo de Error, Tipo de Solución.

- ✓ **Categoría de ODT:** Se carga el cuadro de lista Estado.
- ✓ **Efectos de ODT:** Se carga el cuadro de lista Estado.

- ✓ **Tipo de Error:** Se carga el cuadro de lista Estado.
- ✓ **Tipo de Solución:** Se carga el cuadro de lista Estado.
- **Gestión de Actividades:** Tipo de Actividad.
  - ✓ **Tipo de Actividad:** Se carga el cuadro de lista Estado.

REF. #	FUNCIONES DE FORMULARIO: DEPARTAMENTO	CATEGORÍA
R1.1	Guardar datos ingresados en el formulario Departamento.	Oculto
R1.2	Buscar Departamentos registrados.	Evidente
R1.3	Verificar si Departamento existe.	Evidente
R1.4	Cargar Estados del catálogo del sistema en el cuadro de lista Estado.	Oculto
R1.5	Eliminar Departamento.	Oculto

<b>CASO DE USO</b>	Registrar Departamento.
<b>CU. ID.</b>	SAGMA_CU_01.
<b>ACTORES</b>	Usuario.
<b>PROPÓSITO</b>	Agregar Departamento en el Sistema.
<b>RESUMEN</b>	El Usuario ingresa los datos del Departamento y los registra en el Sistema.
<b>TIPO</b>	Esencial, Primario.
<b>REFERENCIAS CRUZADAS</b>	R1.1, R1.3, R1.4.
<b>PRECONDICIONES</b>	1. Usuario registrado en el Sistema. 2. Usuario debidamente autenticado en el Sistema.

CURSO NORMAL DE EVENTOS	
ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1. Este Caso de Uso inicia cuando el Usuario va a registrar los datos de un nuevo Departamento en el Sistema.	
2. El Usuario da Click en el botón Departamento del módulo Empresa.	
	3. Muestra el formulario Departamento.
	4. Carga los Estados en el cuadro de lista Estado.
5. El Usuario ingresa los campos necesarios.	
6. Presiona la tecla F5 o da Click sobre el botón Guardar para registrar los datos.	
	7. Realiza validación de campos ingresados y seleccionados, los

8. El Caso de Uso termina cuando el usuario da Click sobre el botón OK del mensaje emergente.	guarda en el Sistema y emite el mensaje "Registro realizado correctamente con código #".
<b>CURSOS ALTERNOS</b>	
<b>LÍNEA 7</b>	Si existe algún campo vacío, se emite la respectiva notificación indicando y seleccionado el campo que deber ser ingresado y pasa el control del sistema al Usuario.
<b>LÍNEA 7</b>	Si el Departamento se encuentra registrado en la base de datos, se emite la notificación "El registro ya existe. No se realizó la transacción." Y se pasa el control del sistema al Usuario.
<b>LÍNEA 7</b>	Si ocurre algún error al momento de guardar los datos, se notifica dicho error y se pasa el control del sistema al Usuario.
<b>POST CONDICIONES</b>	
El Departamento ha sido registrado en el Sistema.	

<b>CASO DE USO</b>	Actualizar Departamento.
<b>CU. ID.</b>	SAGMA_CU_02.
<b>ACTORES</b>	Usuario.
<b>PROPÓSITO</b>	Actualizar Departamento en el Sistema.
<b>RESUMEN</b>	El Usuario busca el registro que desea actualizar, modifica los datos del Departamento y registra los cambios.
<b>TIPO</b>	Esencial, Primario.
<b>REFERENCIAS CRUZADAS</b>	R1.1, R1.2, R1.3, R1.4.
<b>PRECONDICIONES</b>	1. Usuario registrado en el Sistema. 2. Usuario debidamente autenticado en el Sistema.
<b>CURSO NORMAL DE EVENTOS</b>	
<b>ACCIÓN DEL ACTOR</b>	<b>RESPUESTA DEL SISTEMA</b>
1. Este Caso de Uso inicia cuando el Usuario va a modificar los datos de Departamento existente en el Sistema.	
2. El Usuario da Click en el botón Departamento del módulo Empresa.	
	3. Muestra el formulario Departamento.
	4. Carga los Estados en el cuadro de lista Estado.
5. El presiona la tecla F3 o da Click en el botón Buscar.	
	6. Muestra el listado de Departamentos registrados.
7. El Usuario elige el registro que desea actualizar y modifica los datos.	
8. Presiona la tecla F5 o da Click sobre el botón Guardar para registrar los datos.	
	9. Realiza validación de campos

10. El Caso de Uso termina cuando el usuario da Click sobre el botón OK del mensaje emergente.	ingresados y seleccionados, los guarda en el Sistema y emite el mensaje “# registro(s) actualizado(s).”.
<b>CURSOS ALTERNOS</b>	
<b>LÍNEA 9</b>	Si existe algún campo vacío, se emite la respectiva notificación indicando y seleccionado el campo que deber ser ingresado y pasa el control del sistema al Usuario.
<b>LÍNEA 9</b>	Si el Departamento se encuentra registrado en la base de datos, se emite la notificación “El registro ya existe. No se realizó la transacción.” Y se pasa el control del sistema al Usuario.
<b>LÍNEA 9</b>	Si ocurre algún error al momento de guardar los datos, se notifica dicho error y se pasa el control del sistema al Usuario.
<b>POST CONDICIONES</b>	
El Departamento ha sido actualizado y registrado en el Sistema.	

<b>CASO DE USO</b>	Eliminar Departamento.
<b>CU. ID.</b>	SAGMA_CU_03.
<b>ACTORES</b>	Usuario.
<b>PROPÓSITO</b>	Eliminar Departamento en el Sistema.
<b>RESUMEN</b>	El Usuario busca el registro que desea eliminar del Sistema.
<b>TIPO</b>	Esencial, Primario.
<b>REFERENCIAS CRUZADAS</b>	R1.1, R1.2, R1.3, R1.4, R1.5.
<b>PRECONDICIONES</b>	1. Usuario registrado en el Sistema. 2. Usuario debidamente autenticado en el Sistema.
<b>CURSO NORMAL DE EVENTOS</b>	
<b>ACCIÓN DEL ACTOR</b>	<b>RESPUESTA DEL SISTEMA</b>
1. Este Caso de Uso inicia cuando el Usuario va a eliminar un Departamento existente en el Sistema.	
2. El Usuario da Click en el botón Departamento del módulo Empresa.	3. Muestra el formulario Departamento.
	4. Carga los Estados en el cuadro de lista Estado.
5. El presiona la tecla F3 o da Click en el botón Buscar.	6. Muestra el listado de Departamentos registrados.
7. El Usuario elige el registro que desea eliminar.	
8. Presiona la tecla F6 o da Click sobre el botón Eliminar.	9. Presenta el mensaje emergente

10. El Usuario da Click sobre el botón SI del mensaje emergente.	“Desea eliminar permanentemente el registro de la base de datos...?”.
12. El Caso de Uso termina cuando el usuario da Click sobre el botón OK del mensaje emergente.	11. Eliminar el registro y emite el mensaje “# registro(s) eliminado(s).”.
<b>CURSOS ALTERNOS</b>	
<b>LÍNEA 10</b>	Si no desea eliminar el registro, termina el caso de uso.
<b>LÍNEA 11</b>	Si ocurre algún error al momento de guardar los datos, se notifica dicho error y se pasa el control del sistema al Usuario.
<b>POST CONDICIONES</b>	
El Departamento ha sido eliminado en el Sistema.	

REF. #	FUNCIONES DE FORMULARIO: ODT	CATEGORÍA
R2.1	Guardar datos ingresados en el formulario ODT.	Oculto
R2.2	Buscar ODT's registradas.	Evidente
R2.3	Cargar Desarrolladores en el cuadro de lista Asignatario.	Oculto
R2.4	Cargar Clientes en el cuadro de lista Cliente.	Oculto
R2.5	Cargar Áreas en el cuadro de lista Área Solicitante.	Oculto
R2.6	Cargar Usuarios Beneficiarios en el cuadro de lista Beneficiario.	Oculto
R2.7	Cargar Usuarios Autorizadores en el cuadro de lista Autorizador.	Oculto
R2.8	Cargar Aplicaciones en el cuadro de lista Aplicación.	Oculto
R2.9	Cargar Categorías de ODT en el cuadro de lista Categoría.	Oculto
R2.10	Cargar Tipos de Errores en el cuadro de lista Tipo de Error.	Oculto
R2.11	Cargar Efectos en el cuadro de lista Efecto.	Oculto
R2.12	Cargar Tipos de Soluciones en el cuadro de lista Tipo de Solución.	Oculto
R2.13	Cargar Desarrolladores en el cuadro de lista Resolutor.	Oculto
R2.14	Cargar Estados del catálogo del sistema en el cuadro de lista Estado.	Oculto
R2.15	Eliminar ODT.	Oculto

<b>CASO DE USO</b>	Registrar ODT.
<b>CU. ID.</b>	SAGMA_CU_04.
<b>ACTORES</b>	Usuario.
<b>PROPÓSITO</b>	Agregar ODT en el Sistema.
<b>RESUMEN</b>	El Usuario ingresa los datos de la Orden de Trabajo y los registra en el Sistema.
<b>TIPO</b>	Esencial, Primario.
<b>REFERENCIAS CRUZADAS</b>	R2.1, R2.3, R2.4, R2.5, R2.6, R2.7, R2.8, R2.9, R2.10, R2.11, R2.12, R2.13, R2.14.
<b>PRECONDICIONES</b>	1. Usuario registrado en el Sistema. 2. Usuario debidamente autenticado en el Sistema.

CURSO NORMAL DE EVENTOS	
ACCIÓN DEL ACTOR	RESPUESTA DEL SISTEMA
1. Este Caso de Uso inicia cuando el Usuario va a registrar los datos de una nueva ODT en el Sistema.  2. El Usuario da Click en el botón ODT del módulo Gestión.       5. El Usuario ingresa los campos necesarios.  6. El Usuario selecciona el Cliente del cuadro de lista Cliente.   8. Presiona la tecla F5 o da Click sobre el botón Guardar para registrar los datos.    10. El Caso de Uso termina cuando el usuario da Click sobre el botón OK del mensaje emergente.	3. Muestra el formulario Orden de Trabajo.  4. Carga los cuadros de lista Prioridad, Tipo, Asignar a, Estado, Cliente, Categoría, Tipo de Error, Efecto, Tipo de Solución, Resolutor.      7. El sistema Carga los cuadros de lista Área Solicitante, Autorizador, Beneficiario, Aplicación.   9. Realiza validación de campos ingresados y seleccionados, los guarda en el Sistema y emite el mensaje "Registro realizado correctamente con código #".
CURSOS ALTERNOS	
<b>LÍNEA 9</b>	Si existe algún campo vacío, se emite la respectiva notificación indicando y seleccionado el campo que deber ser ingresado y pasa el control del sistema al Usuario.
<b>LÍNEA 9</b>	Si ocurre algún error al momento de guardar los datos, se notifica dicho error y se pasa el control del sistema al Usuario.
POST CONDICIONES	
La ODT ha sido registrada en el Sistema.	

<b>CASO DE USO</b>	Actualizar ODT.
<b>CU. ID.</b>	SAGMA_CU_05.
<b>ACTORES</b>	Usuario.
<b>PROPÓSITO</b>	Actualizar ODT en el Sistema.
<b>RESUMEN</b>	El Usuario busca el registro que desea actualizar, modifica los datos de la ODT y registra los cambios.
<b>TIPO</b>	Esencial, Primario.
<b>REFERENCIAS CRUZADAS</b>	R2.1, R2.2, R2.3, R2.4, R2.5, R2.6, R2.7, R2.8, R2.9, R2.10, R2.11, R2.12, R2.13, R2.14.



<b>PRECONDICIONES</b>		1. Usuario registrado en el Sistema. 2. Usuario debidamente autenticado en el Sistema.
<b>CURSO NORMAL DE EVENTOS</b>		
<b>ACCIÓN DEL ACTOR</b>		<b>RESPUESTA DEL SISTEMA</b>
1. Este Caso de Uso inicia cuando el Usuario va a modificar los datos de una ODT existente en el Sistema.  2. El Usuario da Click en el botón ODT del módulo Gestión.          5. El presiona la tecla F3 o da Click en el botón Buscar.  7. El Usuario elige el registro que desea actualizar y modifica los datos.  8. Presiona la tecla F5 o da Click sobre el botón Guardar para registrar los datos.       10. El Caso de Uso termina cuando el usuario da Click sobre el botón OK del mensaje emergente.		3. Muestra el formulario Orden de Trabajo.  4. Carga los cuadros de lista Prioridad, Tipo, Asignar a, Estado, Cliente, Categoría, Tipo de Error, Efecto, Tipo de Solución, Resolutor.  6. Muestra el listado de ODT's registradas.       9. Realiza validación de campos ingresados y seleccionados, los guarda en el Sistema y emite el mensaje "# registro(s) actualizado(s)".
<b>CURSOS ALTERNOS</b>		
<b>LÍNEA 9</b>	Si existe algún campo vacío, se emite la respectiva notificación indicando y seleccionado el campo que deber ser ingresado y pasa el control del sistema al Usuario.	
<b>LÍNEA 9</b>	Si ocurre algún error al momento de guardar los datos, se notifica dicho error y se pasa el control del sistema al Usuario.	
<b>POST CONDICIONES</b>		
La ODT ha sido actualizada y registrada en el Sistema.		

<b>CASO DE USO</b>	Eliminar ODT.
<b>CU. ID.</b>	SAGMA_CU_06.
<b>ACTORES</b>	Usuario.
<b>PROPÓSITO</b>	Eliminar ODT en el Sistema.
<b>RESUMEN</b>	El Usuario busca el registro que desea eliminar del Sistema.
<b>TIPO</b>	Esencial, Primario.
<b>REFERENCIAS CRUZADAS</b>	R2.1, R2.2, R2.3, R2.4, R2.5, R2.6, R2.7, R2.8, R2.9, R2.10, R2.11, R2.12, R2.13, R2.14, R2.15.

<b>PRECONDICIONES</b>		1. Usuario registrado en el Sistema. 2. Usuario debidamente autenticado en el Sistema.	
<b>CURSO NORMAL DE EVENTOS</b>			
<b>ACCIÓN DEL ACTOR</b>		<b>RESPUESTA DEL SISTEMA</b>	
1. Este Caso de Uso inicia cuando el Usuario va a eliminar una ODT existente en el Sistema.		3. Muestra el formulario Orden de Trabajo.  4. Carga los cuadros de lista Prioridad, Tipo, Asignar a, Estado, Cliente, Categoría, Tipo de Error, Efecto, Tipo de Solución, Resolutor.  6. Muestra el listado de ODT's registradas.    9. Presenta el mensaje emergente "Desea eliminar permanentemente el registro de la base de datos...?".  11. Eliminar el registro y emite el mensaje "# registro(s) eliminado(s)".	
2. El Usuario da Click en el botón ODT del módulo Gestión.			
5. El presiona la tecla F3 o da Click en el botón Buscar.			
7. El Usuario elige el registro que desea eliminar.			
8. Presiona la tecla F6 o da Click sobre el botón Eliminar.			
10. El Usuario da Click sobre el botón SI del mensaje emergente.			
12. El Caso de Uso termina cuando el usuario da Click sobre el botón OK del mensaje emergente.			
<b>CURSOS ALTERNOS</b>			
<b>LÍNEA 10</b>	Si no desea eliminar el registro, termina el caso de uso.		
<b>LÍNEA 11</b>	Si ocurre algún error al momento de guardar los datos, se notifica dicho error y se pasa el control del sistema al Usuario.		
<b>POST CONDICIONES</b>			
La ODT ha sido eliminada en el Sistema.			

REF. #	FUNCIONES DE FORMULARIO: ACTIVIDADES	CATEGORÍA
R3.1	Guardar datos ingresados en el formulario Actividades.	Oculto
R3.2	Buscar Actividades registradas.	Evidente
R3.3	Cargar Tipos de Actividades en el cuadro de lista Tipo.	Oculto
R3.4	Cargar Estados del catálogo del sistema en el cuadro de lista Estado.	Oculto
R3.5	Eliminar Actividad.	Oculto

CASO DE USO	Registrar Actividad.		
CU. ID.	SAGMA_CU_07.		
ACTORES	Usuario.		
PROPÓSITO	Agregar Actividades en el Sistema.		
RESUMEN	El Usuario ingresa los datos de la Actividad y los registra en el Sistema.		
TIPO	Esencial, Primario.		
REFERENCIAS CRUZADAS	R3.1, R3.3, R3.4.		
PRECONDICIONES	1. Usuario registrado en el Sistema. 2. Usuario debidamente autenticado en el Sistema.		
CURSO NORMAL DE EVENTOS			
ACCIÓN DEL ACTOR		RESPUESTA DEL SISTEMA	
1. Este Caso de Uso inicia cuando el Usuario va a registrar los datos de una nueva Actividad en el Sistema.			
2. El Usuario da Click en el botón Actividades del módulo Gestión.		3. Muestra el formulario Registro de Actividades.	
		4. Carga los Tipos de Actividades en el cuadro de lista Tipo y los Estados en el cuadro de lista Estado.	
5. El Usuario ingresa los campos necesarios.			
6. El Usuario digita el número de la ODT.		7. El sistema Carga los datos de la ODT.	
8. Presiona la tecla F5 o da Click sobre el botón Guardar para registrar los datos.		9. Realiza validación de campos ingresados y seleccionados, los guarda en el Sistema y emite el mensaje “Registro realizado correctamente con código #”.	
10. El Caso de Uso termina cuando el usuario da Click sobre el botón OK del mensaje emergente.			
CURSOS ALTERNOS			
LÍNEA 7	Si no existe el número de ODT especificado por el Usuario, se notifica y se pasa el control al Usuario.		
LÍNEA 9	Si existe algún campo vacío, se emite la respectiva notificación indicando y seleccionado el campo que deber ser ingresado y pasa el control del sistema al Usuario.		
LÍNEA 9	Si ocurre algún error al momento de guardar los datos, se notifica dicho error y se pasa el control del sistema al Usuario.		
POST CONDICIONES			

La Actividad ha sido registrada en el Sistema.

CASO DE USO	Actualizar Actividad.		
CU. ID.	SAGMA_CU_08.		
ACTORES	Usuario.		
PROPÓSITO	Actualizar Actividad en el Sistema.		
RESUMEN	El Usuario busca el registro que desea actualizar, modifica los datos de la Actividad y registra los cambios.		
TIPO	Esencial, Primario.		
REFERENCIAS CRUZADAS	R3.1, R3.2, R3.3, R3.4.		
PRECONDICIONES	1. Usuario registrado en el Sistema. 2. Usuario debidamente autenticado en el Sistema.		
CURSO NORMAL DE EVENTOS			
ACCIÓN DEL ACTOR		RESPUESTA DEL SISTEMA	
1. Este Caso de Uso inicia cuando el Usuario va a modificar los datos de una Actividad existente en el Sistema.		3. Muestra el formulario Registro de Actividades.  4. Carga los Tipos de Actividades en el cuadro de lista Tipo y los Estados en el cuadro de lista Estado.  6. Muestra el listado de Actividades registradas por el Usuario.  9. Realiza validación de campos ingresados y seleccionados, los guarda en el Sistema y emite el mensaje “# registro(s) actualizado(s).”.	
2. El Usuario da Click en el botón Actividades del módulo Gestión.			
5. El presiona la tecla F3 o da Click en el botón Buscar.			
7. El Usuario elige el registro que desea actualizar y modifica los datos.			
8. Presiona la tecla F5 o da Click sobre el botón Guardar para registrar los datos.			
10. El Caso de Uso termina cuando el usuario da Click sobre el botón OK del mensaje emergente.			
CURSOS ALTERNOS			
LÍNEA 9	Si existe algún campo vacío, se emite la respectiva notificación indicando y seleccionado el campo que deber ser ingresado y pasa el control del sistema al Usuario.		
LÍNEA 9	Si ocurre algún error al momento de guardar los datos, se notifica dicho error y se pasa el control del sistema al Usuario.		
POST CONDICIONES			

La Actividad ha sido actualizada y registrada en el Sistema.

CASO DE USO	Eliminar Actividad.		
CU. ID.	SAGMA_CU_09.		
ACTORES	Usuario.		
PROPÓSITO	Eliminar Actividad en el Sistema.		
RESUMEN	El Usuario busca el registro que desea eliminar del Sistema.		
TIPO	Esencial, Primario.		
REFERENCIAS CRUZADAS	R3.1, R3.2, R3.3, R3.4, R3.5.		
PRECONDICIONES	1. Usuario registrado en el Sistema. 2. Usuario debidamente autenticado en el Sistema.		
CURSO NORMAL DE EVENTOS			
ACCIÓN DEL ACTOR		RESPUESTA DEL SISTEMA	
1.	Este Caso de Uso inicia cuando el Usuario va a eliminar una Actividad existente en el Sistema.	3. Muestra el formulario Orden de Trabajo.  4. Carga los Tipos de Actividades en el cuadro de lista Tipo y los Estados en el cuadro de lista Estado.  6. Muestra el listado de Actividades registradas por el usuario.  9. Presenta el mensaje emergente “Desea eliminar permanentemente el registro de la base de datos...?”.  11. Eliminar el registro y emite el mensaje “# registro(s) eliminado(s).”.	
2.	El Usuario da Click en el botón Actividades del módulo Gestión.		
5.	El presiona la tecla F3 o da Click en el botón Buscar.		
7.	El Usuario elige el registro que desea eliminar.		
8.	Presiona la tecla F6 o da Click sobre el botón Eliminar.		
10.	El Usuario da Click sobre el botón SI del mensaje emergente.		
12.	El Caso de Uso termina cuando el usuario da Click sobre el botón OK del mensaje emergente.		
CURSOS ALTERNOS			
LÍNEA 10	Si no desea eliminar el registro, termina el caso de uso.		
LÍNEA 11	Si ocurre algún error al momento de guardar los datos, se notifica dicho error y se pasa el control del sistema al Usuario.		
POST CONDICIONES			

La Actividad ha sido eliminada en el Sistema.
---

#### **4.2.2.1.7. Requerimientos no Funcionales**

- **Requerimientos de Rendimiento**

**SAGMA**, optimizará el uso de recursos hardware y las capacidades del entorno de software en el cual estará instalado, ofreciendo al usuario tiempos de respuesta aceptables.

- **Requerimientos de Interfaz de Usuario**

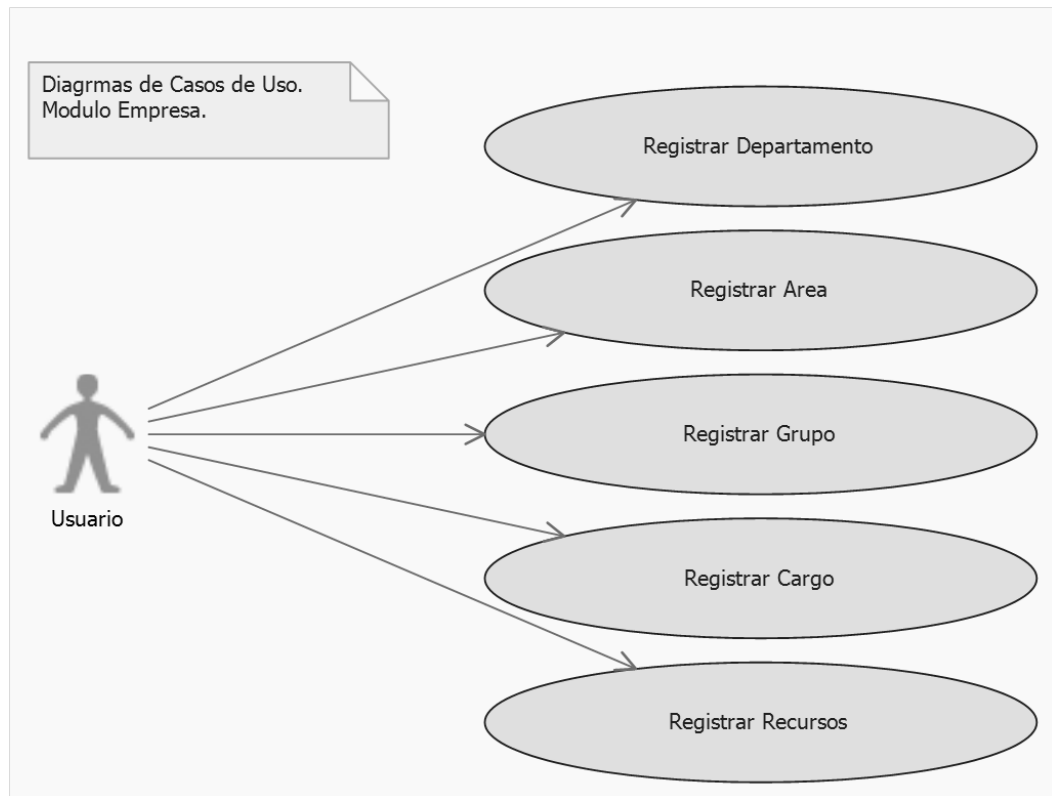
**SAGMA**, tendrá una interfaz amigable y elegante para la comodidad de los usuarios.

- **Requerimientos de Seguridad**

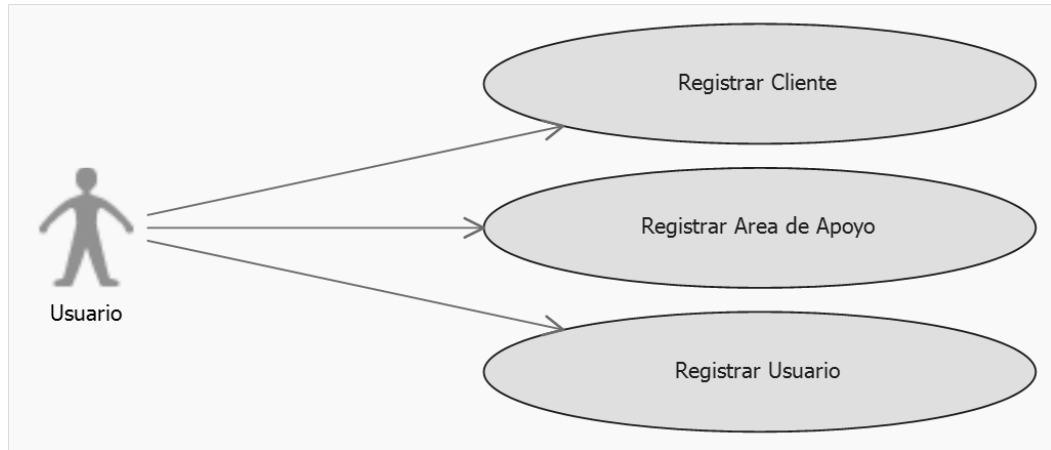
**SAGMA**, aplicará autenticación de usuario y conexiones a la base de datos bajo estricto requerimiento, ofreciendo registros de auditoría sobre toda acción realizada en el sistema.

#### 4.2.2.1.8. Diagramas de Casos de Uso

- **Módulo Empresa**

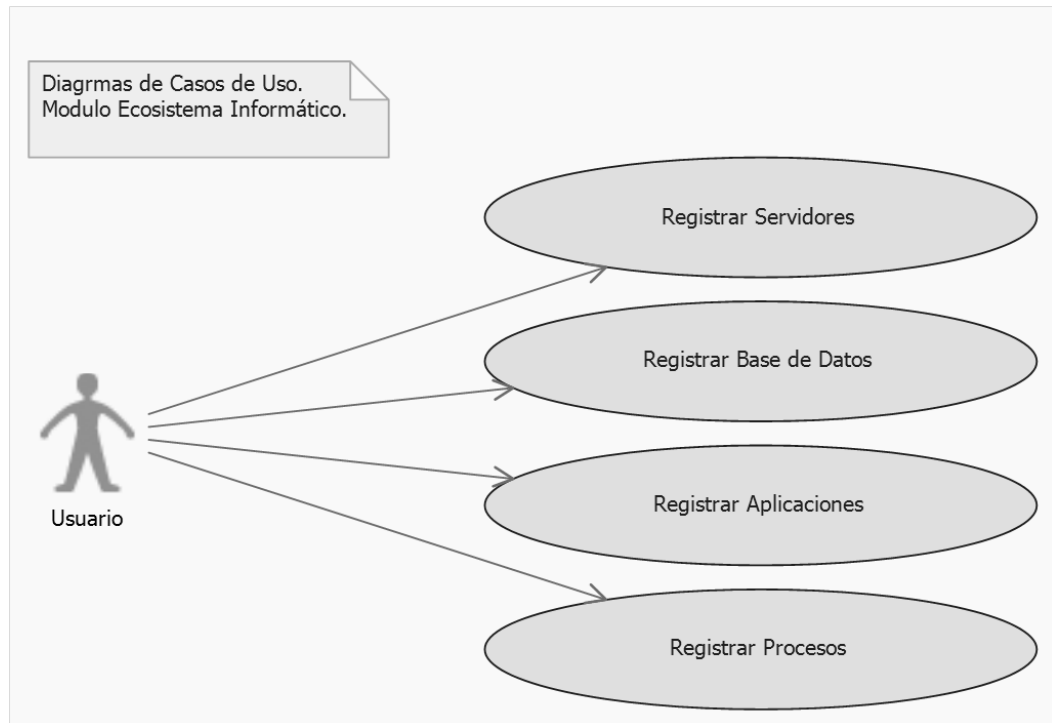


- **Módulo Cliente**

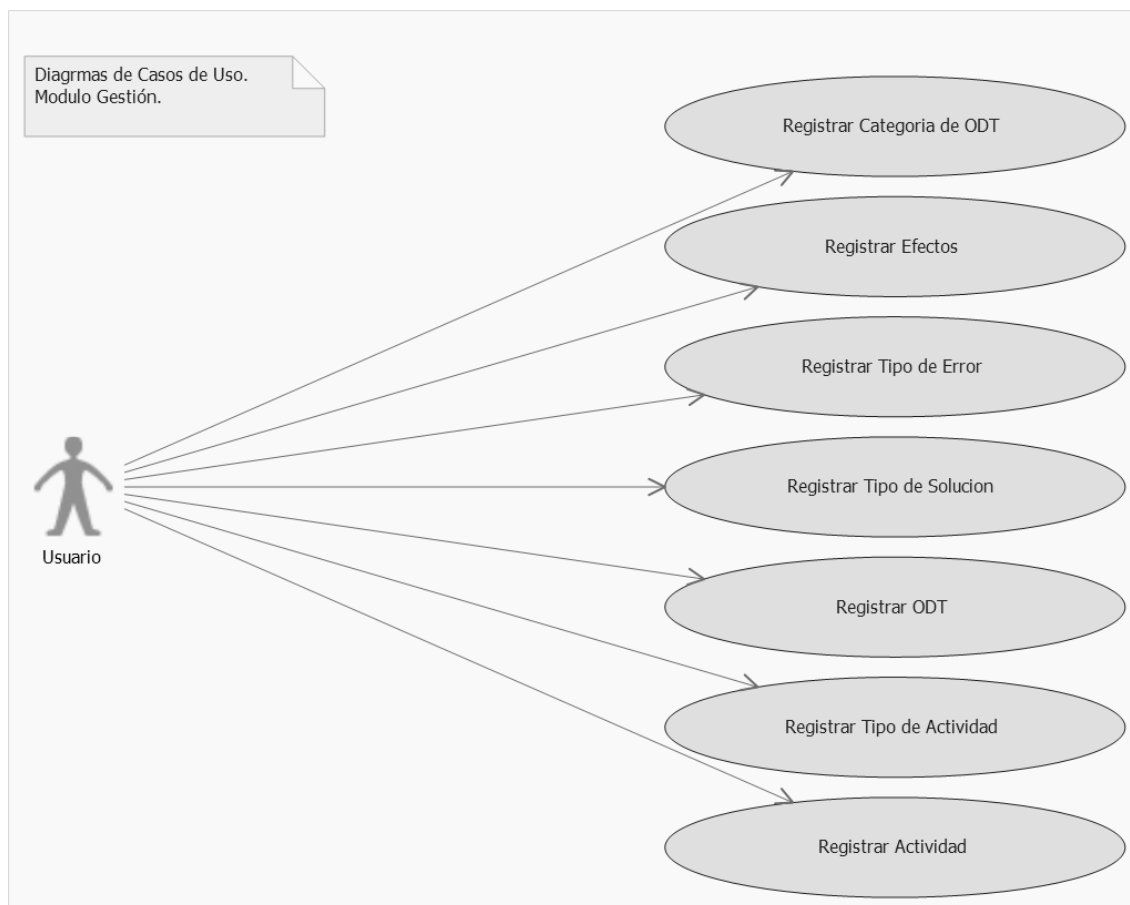




- **Módulo Ecosistema Informático**



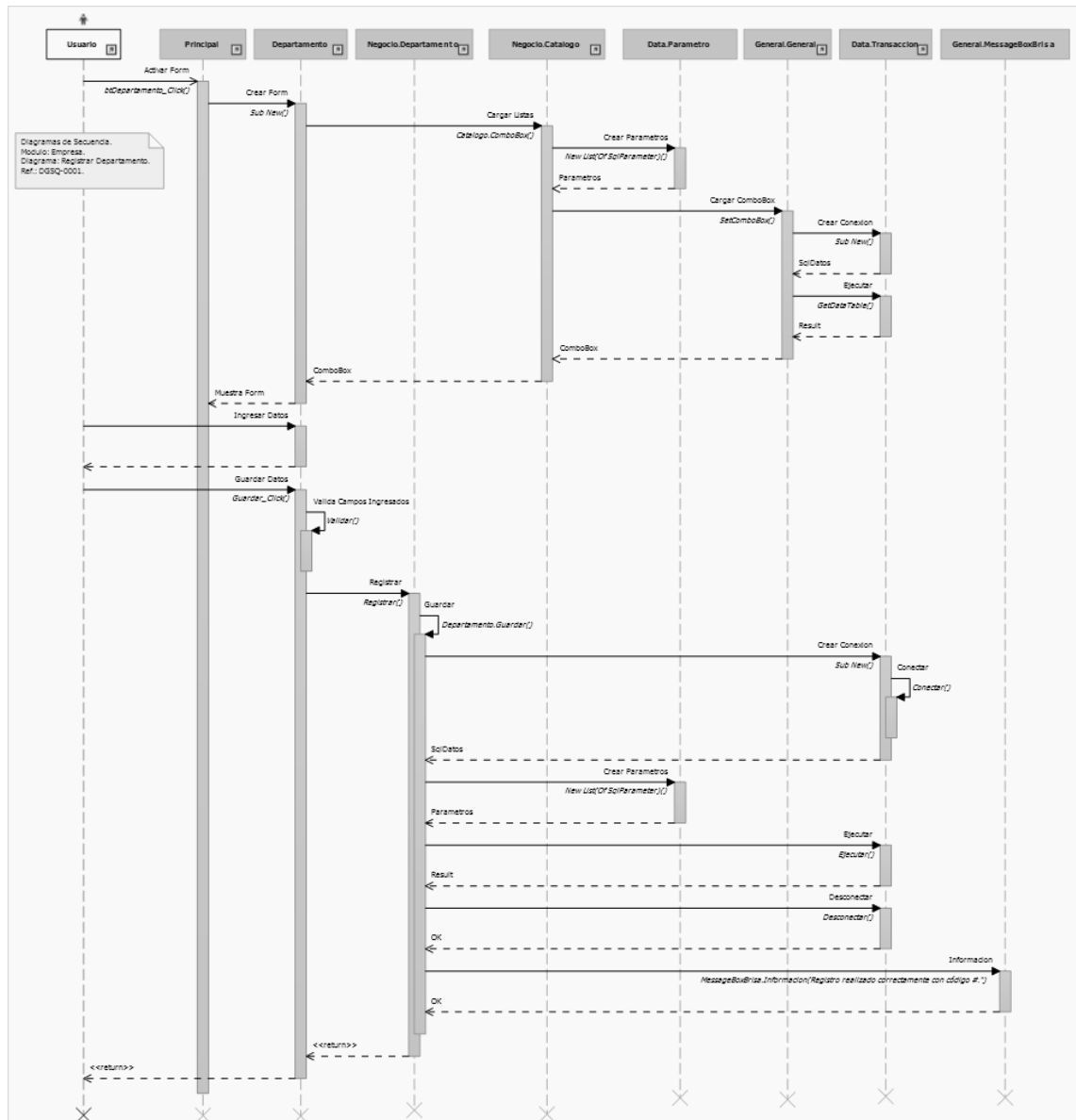
- **Módulo Gestión**



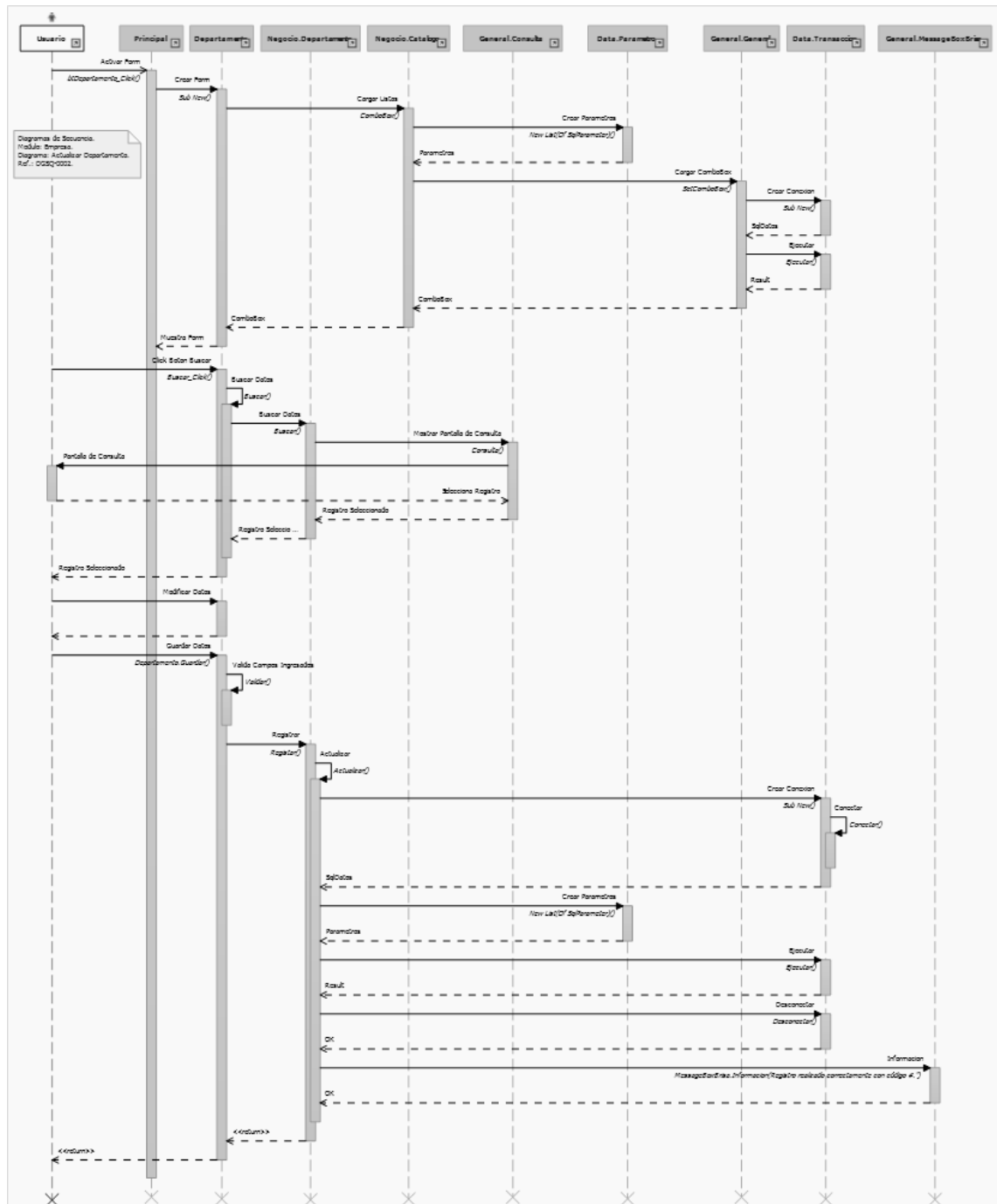
#### 4.2.2.1.9. Diagramas de Secuencia

Teniendo en cuenta que la secuencia de pasos para guardar, actualizar y eliminar es repetitiva para todos los diagramas, se ha considerado el diseño de los más importantes. Se toma como modelo la entidad Departamento del módulo Empresa.

- Registrar Departamento



- Actualizar Departamento



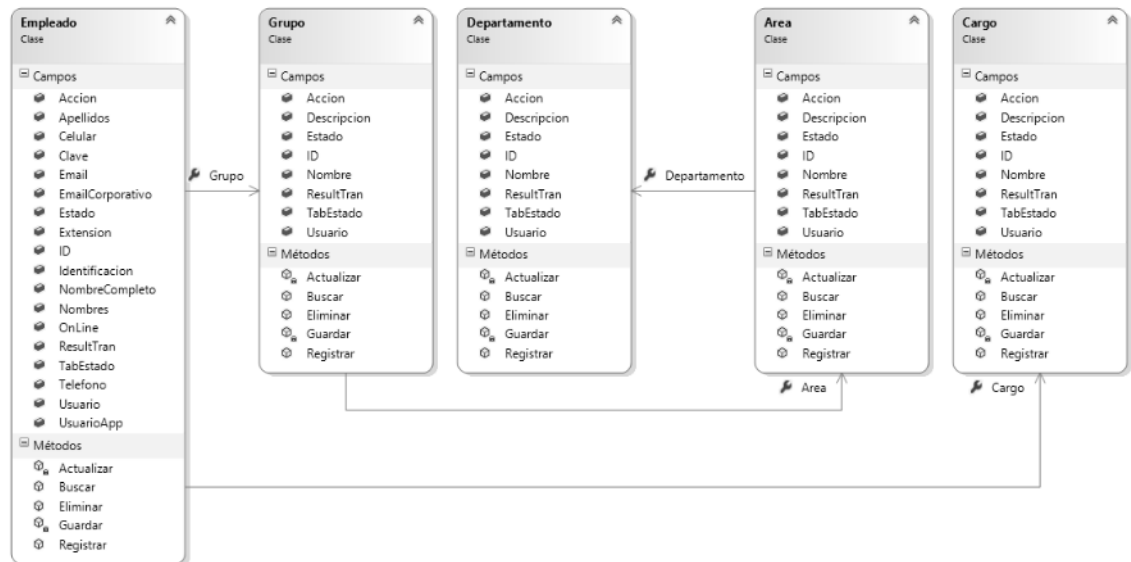
```

sequenceDiagram
    participant Usuario
    participant Principal
    participant Departamento
    participant NegocioDepartamento as Negocio Departamento
    participant NegocioCatalogo as Negocio Catalogo
    participant GeneralConsulta as General Consulta
    participant DataParametro as Data Parametro
    participant GeneralGeneral as General General
    participant DataTransaccion as Data Transaccion
    participant GeneralMensajeBorrado as General MensajeBorrado

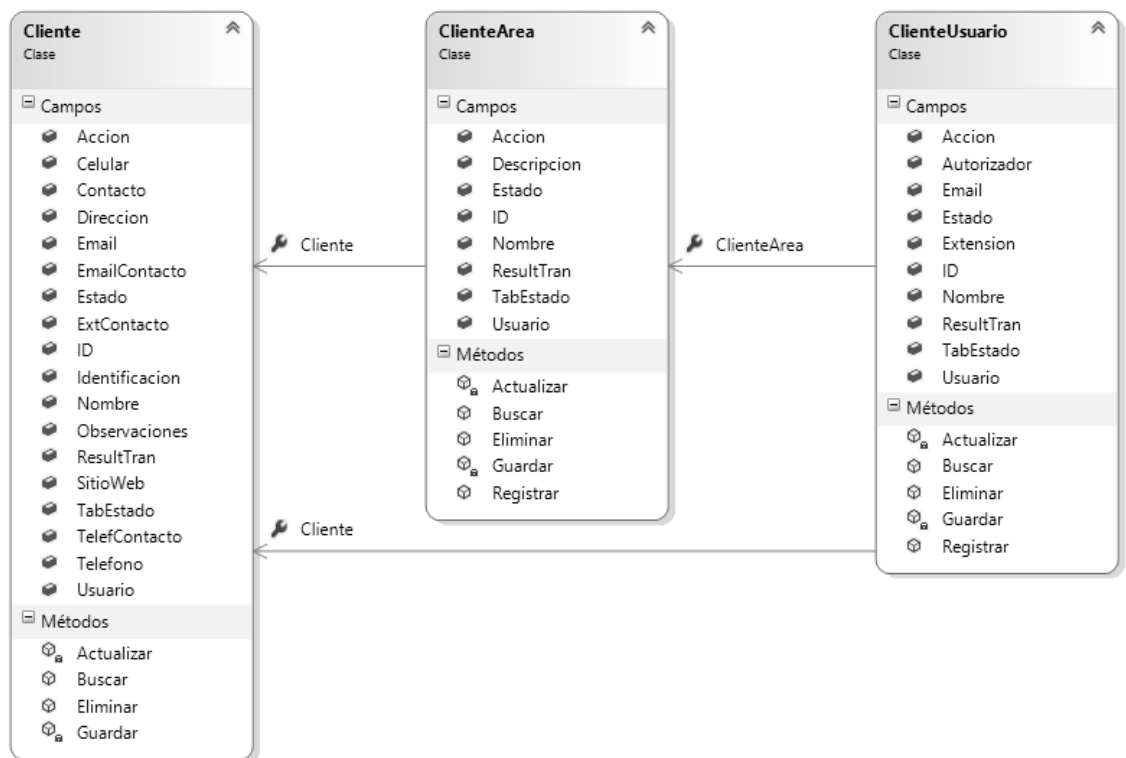
    Usuario->>Principal: Aceptar Param
    Principal->>Departamento: Eliminar_Click()
    Departamento->>NegocioDepartamento: Eliminar()
    NegocioDepartamento->>NegocioCatalogo: Eliminar()
    NegocioCatalogo->>GeneralConsulta: Eliminar()
    GeneralConsulta->>DataParametro: Eliminar()
    DataParametro->>GeneralGeneral: Eliminar()
    GeneralGeneral->>DataTransaccion: Eliminar()
    DataTransaccion->>GeneralMensajeBorrado: Eliminar()
    GeneralMensajeBorrado-->>DataTransaccion: MensajeBorrado()
    DataTransaccion-->>GeneralGeneral: MensajeBorrado()
    GeneralGeneral-->>DataParametro: MensajeBorrado()
    DataParametro-->>GeneralConsulta: MensajeBorrado()
    GeneralConsulta-->>NegocioCatalogo: MensajeBorrado()
    NegocioCatalogo-->>NegocioDepartamento: MensajeBorrado()
    NegocioDepartamento-->>Departamento: MensajeBorrado()
    Departamento-->>Principal: MensajeBorrado()
    Principal-->>Usuario: MensajeBorrado()
    Usuario->>Principal: Aceptar
  
```

#### 4.2.2.1.10. Diagrama de Clases

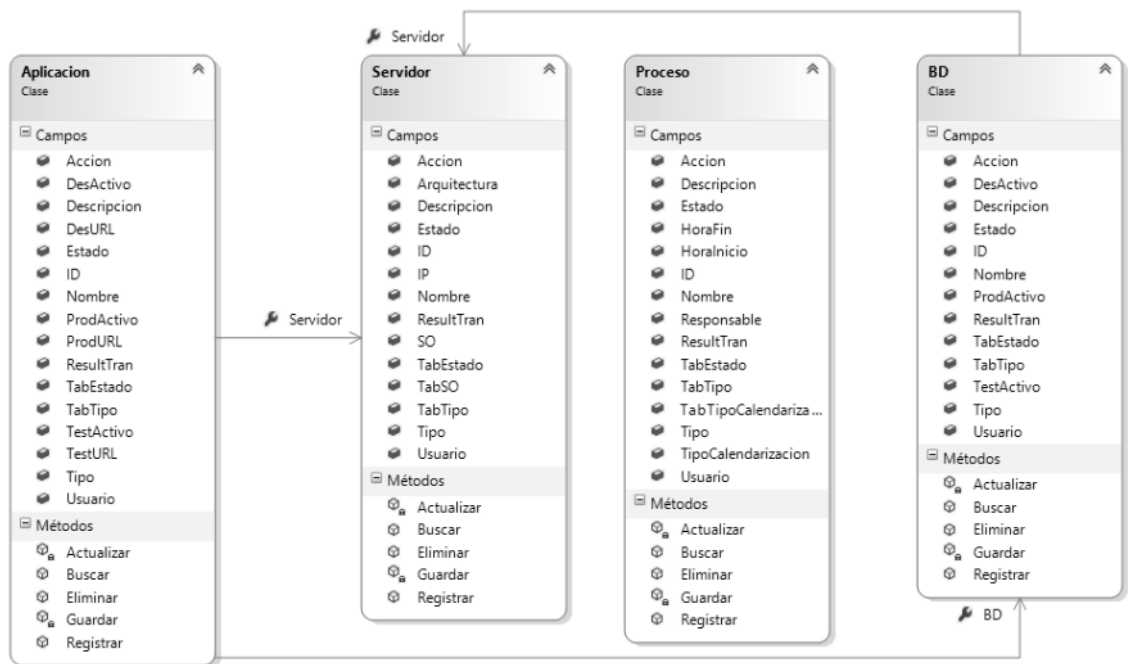
- **Módulo Empresa**



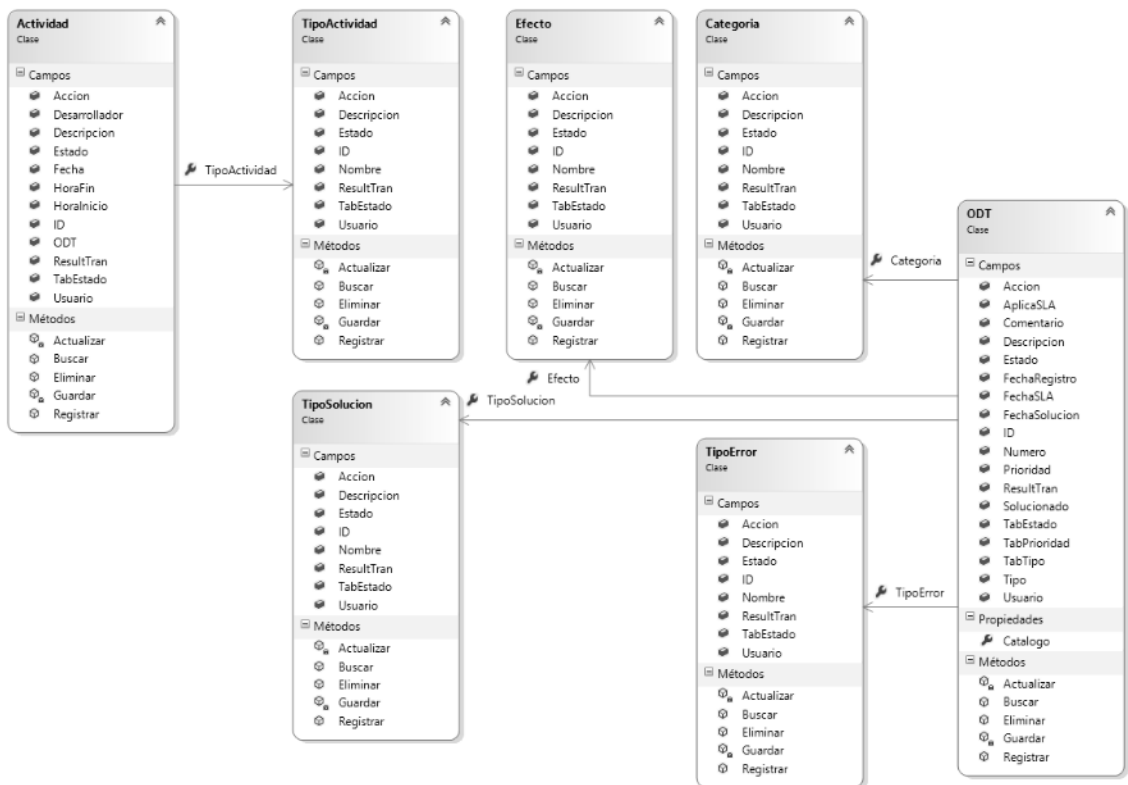
- **Módulo Cliente**



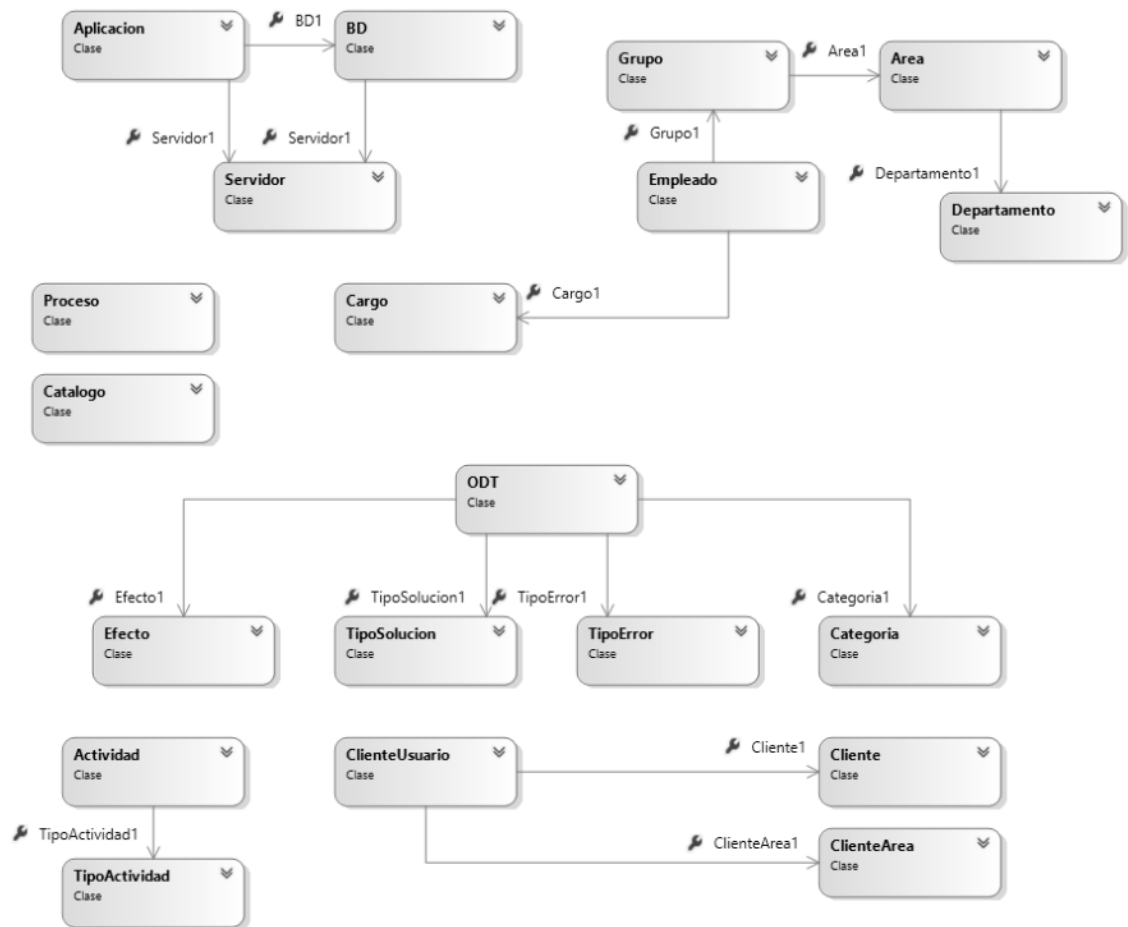
- **Módulo Ecosistema Informático**



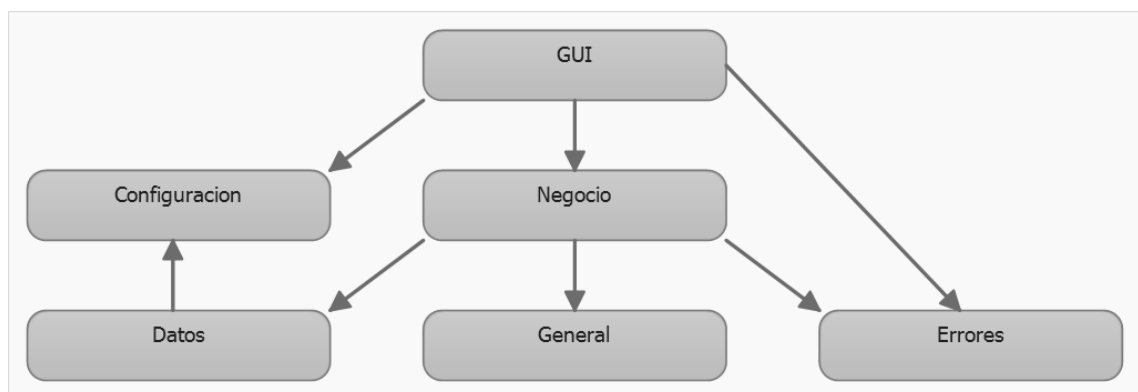
- **Módulo Gestión**



- **Diagrama Global**



#### 4.2.2.1.11. Diagrama de Capas





### 4.2.3. Diseño

En esta fase se realiza el diseño de la interfaz gráfica de usuario (GUI) y el diseño de la Base de Datos.

#### 4.2.3.1. Diseño de Interfaz de Usuario (GUI)

- **Módulo General**

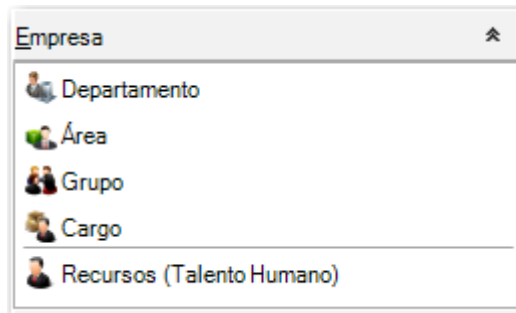
#### Pantalla Principal



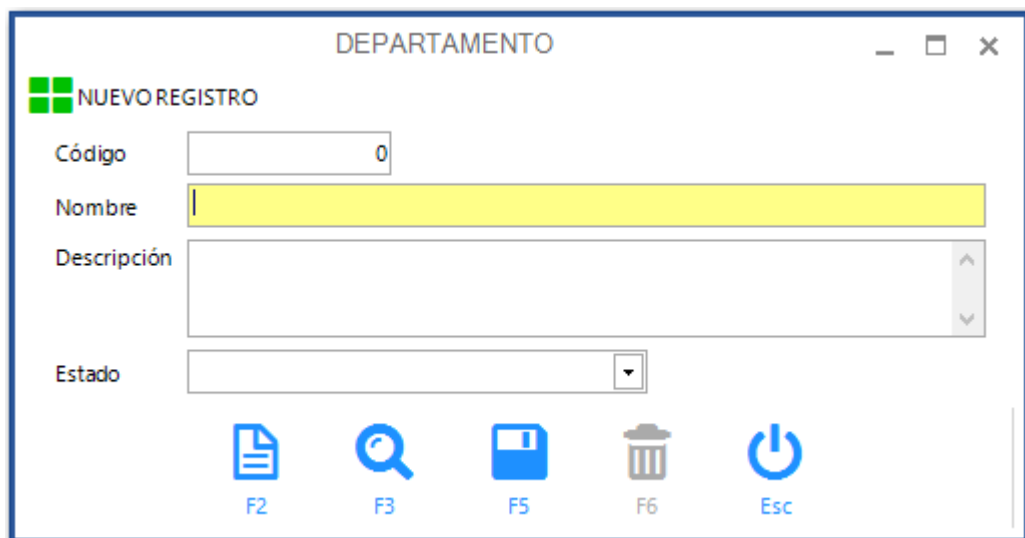
#### Inicio de Sesión

- **Módulo Empresa**

### Menú de Módulo




### Formulario Departamento

A screenshot of a software window titled 'DEPARTAMENTO'. The window has a title bar with standard Windows controls (minimize, maximize, close). Below the title bar, there is a green icon and the text 'NUEVO REGISTRO'. The main area contains four input fields: 'Código' (with the value '0'), 'Nombre' (highlighted in yellow), 'Descripción' (a text area with a scrollbar), and 'Estado' (a dropdown menu). At the bottom of the window, there are five function keys represented by icons and text: 'F2' (document icon), 'F3' (magnifying glass icon), 'F5' (floppy disk icon), 'F6' (trash can icon), and 'Esc' (power button icon).

## Formulario Área




AREA


 NUEVO REGISTRO






Código

Nombre

Descripción


Departamento    

Estado  

 F2  F3  F5  F6  Esc

## Formulario Grupo




GRUPO




 NUEVO REGISTRO


Código






Nombre

Descripción

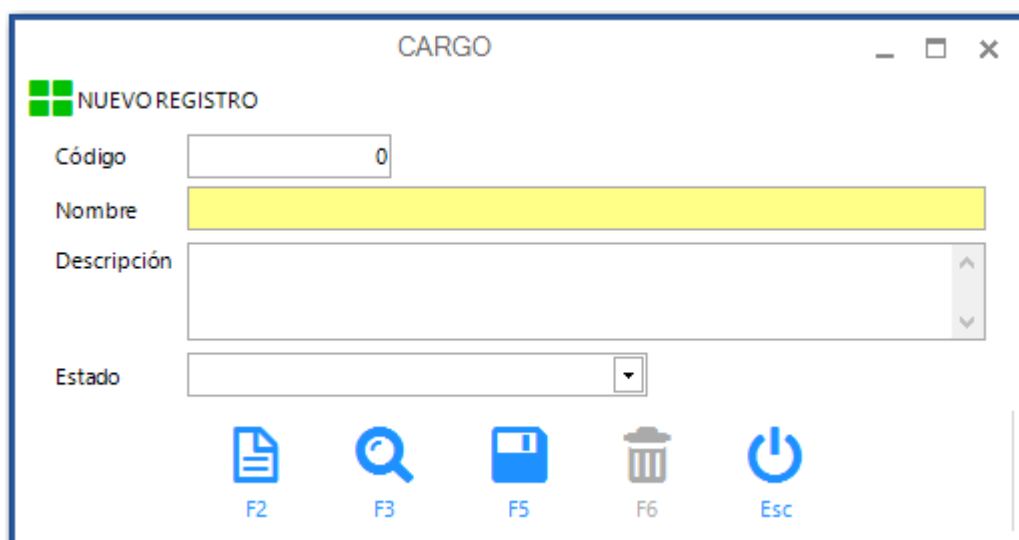
Departamento    

Área    

Estado  

 F2  F3  F5  F6  Esc

## Formulario Cargo



**CARGO**

**NUEVO REGISTRO**

Código

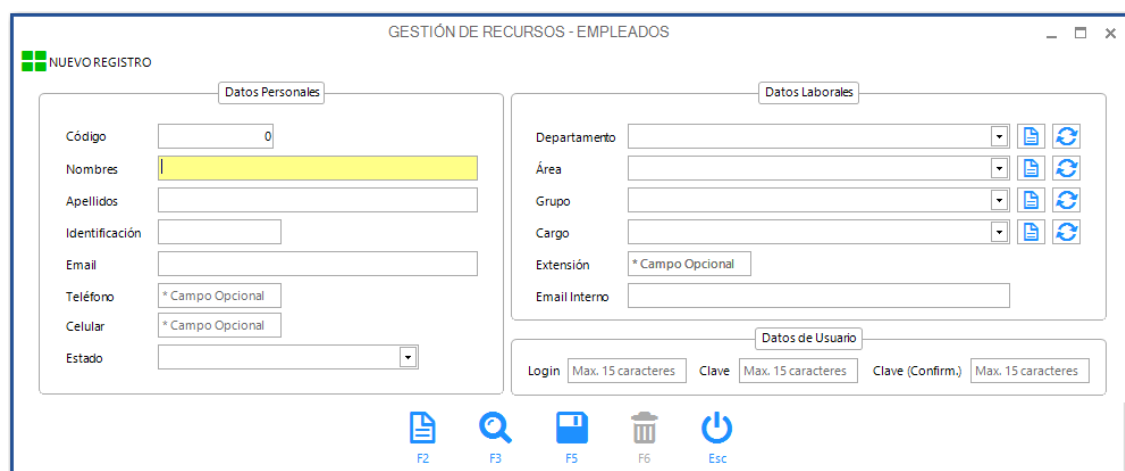
Nombre

Descripción

Estado

F2 F3 F5 F6 Esc

## Formulario Recursos (Talento Humano)



**GESTIÓN DE RECURSOS - EMPLEADOS**

**NUEVO REGISTRO**

**Datos Personales**

Código

Nombres

Apellidos

Identificación

Email

Teléfono  \* Campo Opcional

Celular  \* Campo Opcional

Estado

**Datos Laborales**

Departamento

Área

Grupo

Cargo

Extensión  \* Campo Opcional

Email Interno

**Datos de Usuario**

Login  Max. 15 caracteres

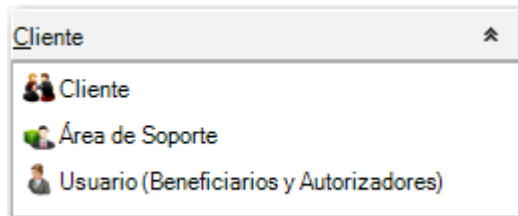
Clave  Max. 15 caracteres

Clave (Confirm.)  Max. 15 caracteres

F2 F3 F5 F6 Esc

- **Módulo Cliente**

### Menú de Módulo




### Formulario Cliente

A screenshot of a web form titled 'Formulario Cliente'. The form is divided into two main sections: 'Datos Generales' and 'Datos de Contacto'. The 'Datos Generales' section includes fields for 'Código' (with a value of 0), 'Identificación' (highlighted in yellow), 'Nombre', 'Dirección', 'Teléfono(s)', 'Celular(es)', 'Email', 'Sitio Web', and 'Estado'. The 'Datos de Contacto' section includes fields for 'Nombre', 'Teléfono(s)', 'Extensión', and 'Email'. There is also a large text area for 'Observaciones / Notas de Cliente'. The form has a blue border and a blue header bar. At the bottom, there are five icons representing different actions: F2 (New), F3 (Search), F5 (Save), F6 (Delete), and Esc (Cancel).

## Formulario Área de Soporte




AREA

 NUEVO REGISTRO






Código

Nombre

Descripción

Cliente    


Estado

F2 F3 F5 F6 Esc

## Formulario Usuario (Beneficiario y Autorizadores)

GESTIÓN DE CLIENTES - USUARIOS

 NUEVO REGISTRO




Código




Nombre

Extensión






Email

☐ Autorizador

Cliente    

Área    

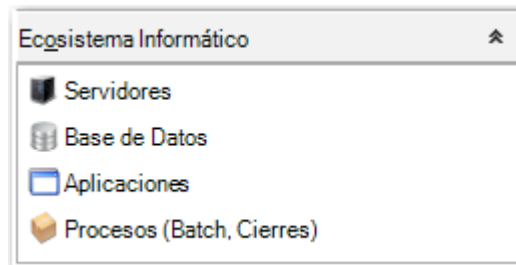
Estado

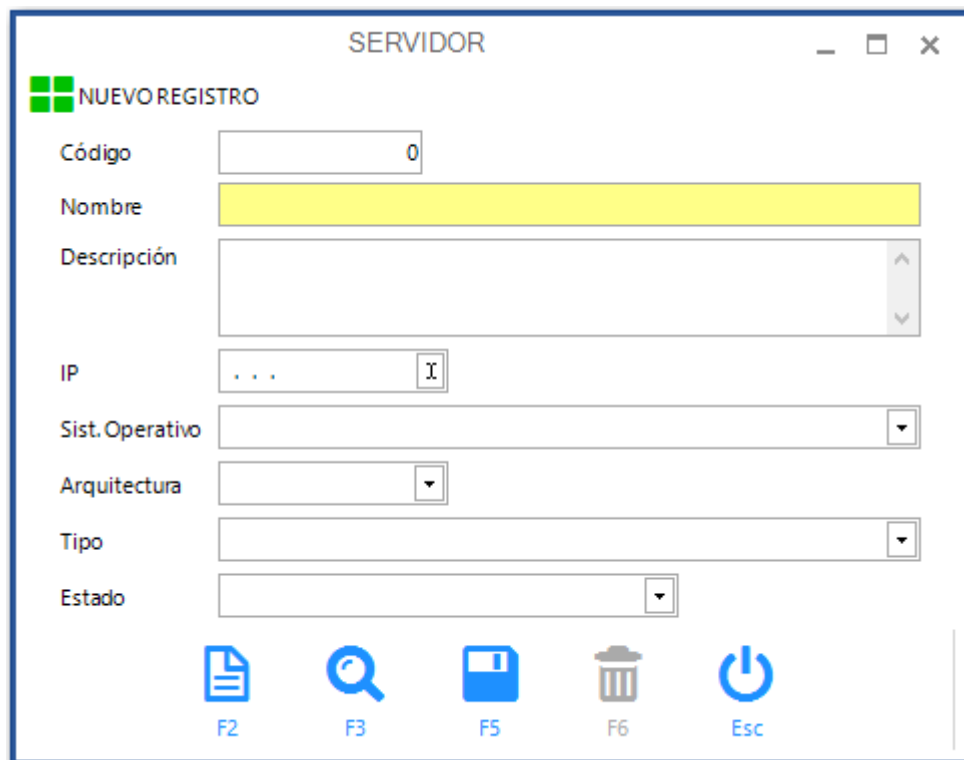
F2 F3 F5 F6 Esc

- **Módulo Ecosistema Informático**

### Menú de Módulo



### Formulario Servidores

A screenshot of a web form titled "SERVIDOR" with a "NUEVO REGISTRO" button. The form contains several input fields: "Código" (text box with "0"), "Nombre" (text box with a yellow highlight), "Descripción" (text area), "IP" (text box with "..."), "Sist. Operativo" (dropdown menu), "Arquitectura" (dropdown menu), "Tipo" (dropdown menu), and "Estado" (dropdown menu). At the bottom, there are five icons with labels: a document icon labeled "F2", a magnifying glass labeled "F3", a floppy disk labeled "F5", a trash can labeled "F6", and a power button labeled "Esc".

## Formulario Base de Datos

NUEVO REGISTRO BASE DE DATOS

**Datos Generales**

Código

Nombre

Descripción

Cliente

Tipo

Estado

**Servidores**

Desarrollo

☐ Activo

Test

☐ Activo

Producción

☐ Activo

F2 F3 F5 F6 Esc

## Formulario Aplicaciones

NUEVO REGISTRO APLICACION

**Datos Generales**

Código

Nombre

Descripción

Tipo

Estado

**Cliente / Base de Datos**

Cliente

BD

**Servidores**

**Desarrollo** ☐ Activo

URL / PATH

**Test** ☐ Activo

URL / PATH

**Producción** ☐ Activo


URL / PATH

F2 F3 F5 F6 Esc



### Formulario Procesos (Batch, Cierres)



PROCESO

 NUEVO REGISTRO

Código

Nombre

Descripción

Responsable   






Hora Inicio

Hora Fin

Tipo


Calendarización

Estado





 F2  F3  F5  F6  Esc

- **Módulo Gestión**



### Menú de Módulo

Gestión (ODT, Actividades) 



**Clasificación**

-  Categoría de ODT
-  Efectos
-  Tipo de Error
-  Tipo de Solución

**ODT**


-  ODT (Orden de Trabajo)
-  Tras la Pista

**Registro de Actividades**

-  Tipo de Actividad
-  Actividades

### Formulario Categoría de ODT

ODT - CATEGORIA






 NUEVO REGISTRO

Código

Nombre


Descripción

Estado

 F2  F3  F5  F6  Esc

### Formulario Efectos

ODT - EFECTO






 NUEVO REGISTRO

Código

Nombre


Descripción

Estado

 F2  F3  F5  F6  Esc

### Formulario Tipo de Error

ODT - TIPO DE ERROR






 NUEVO REGISTRO

Código

Nombre


Descripción

Estado

 F2    F3    F5    F6    Esc

### Formulario Tipo de Solución

ODT - TIPO DE SOLUCION






 NUEVO REGISTRO

Código

Nombre

Descripción

Estado

 F2    F3    F5    F6    Esc

## Formulario ODT (Orden de Trabajo)

ORDEN DE TRABAJO

NUEVO REGISTRO

Registro **Análisis para Asignación**

**Datos Generales**

Código:

Prioridad:

Tipo:

☒ SLA Fecha SLA:

Descripción: max. 2500 caracteres

[Análisis para Asignación](#)

Asignar a:

Estado:

**Clasificación**

Categoría:

Tipo de Error:

Efecto:

**Solución**

☐ Solucionado

Tipo de Solución:

Fecha Solución:

Resolutor:

Comentario:

**Cliente / Usuarios / Aplicación**

Cliente:

Área Solic.:

Autorizador:

Beneficiario:

Aplicación:

OSHI BD: SAGMA PC: TOMYTOSHI Usuario: OBRIONES Versión: 1.0.0.0 Alt + F4: Salir

ORDEN DE TRABAJO

NUEVO REGISTRO

Registro **Análisis para Asignación**

**Cliente / Aplicación**

Cliente:

Aplicación:

Drag a column header here to group by that column

OSHI BD: SAGMA PC: TOMYTOSHI Usuario: OBRIONES Versión: 1.0.0.0 Alt + F4: Salir

## Formulario Tras la Pista

TRAS LA PISTA... ODT's RELACIONADAS

Search bar with magnifying glass icon and power icon.

Status bar: BD: SAGMA PC: TOMYTOSHI Usuario: OBRIONES Versión: 1.0.0.0 Alt + F4: Salir

## Formulario Tipo de Actividad

TIPO DE ACTIVIDAD

NUEVO REGISTRO

Código: 0

Nombre: [highlighted]

Descripción: [text area]

Estado: [dropdown menu]

Toolbar: F2 (document icon), F3 (search icon), F5 (save icon), F6 (trash icon), Esc (power icon)

Formulario Actividades

NUEVO REGISTRO

REGISTRO DE ACTIVIDADES

RegistroActividades del Día

Datos Generales

Código0

Descripción

Fecha2014-01-05Hora Inicio19:35Hora Fin19:36

Tipo

EstadoACTIVO

ODT0

Estado

Tipo

Cliente

Area Solicitante

Autorizador

Beneficiario

Aplicación

Categoría

Tipo de Error

Efecto

Asignatario

SLA

Fecha SLA

Descripción

F2

F3

F5

F6

Esc

NUEVO REGISTRO

REGISTRO DE ACTIVIDADES

RegistroActividades del Día

ID	DESDE	HASTA	ACTIVIDAD	ODT	TIPO	ESTADO REG.

F2

F3

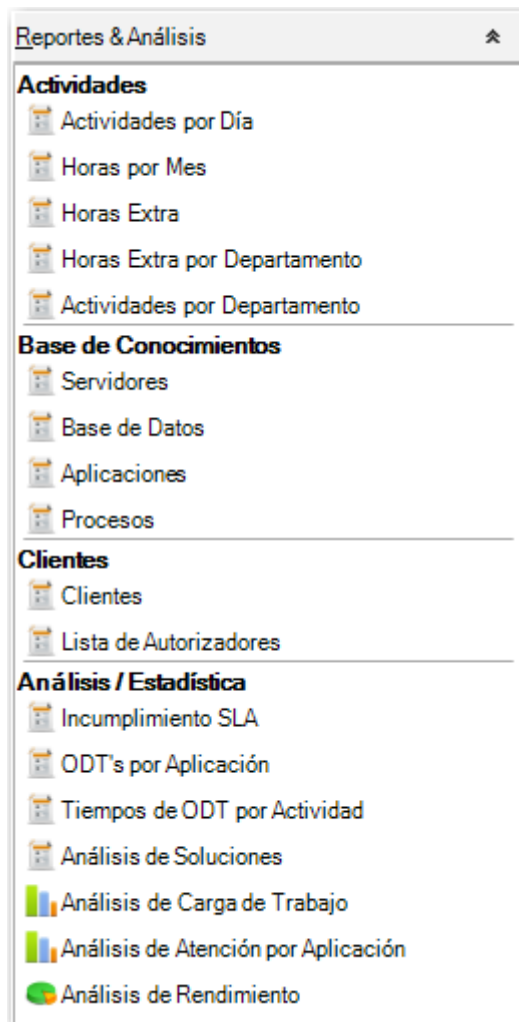
F5

F6

Esc



- **Módulo Reportes y Análisis**






### Menú de Módulo



- **Descripción de Opciones Generales del Sistema**



### Barra de Botones

BOTÓN	DESCRIPCIÓN
 F2	Restablece los controles del formulario.
 F3	Inicia búsqueda de un registro.

 F5	Guarda / Actualiza un registro en la base de datos.
 F6	Eliminar un registro en la base de datos.
 Esc	Cierra ventana actual.
	Sincroniza una lista con registros de la base de datos.
 F5	Presenta en pantalla un reporte.



## Pantalla de Consultas Generales

CONSULTA GENERAL: DEPARTAMENTO

Texto a buscar   

6 Registros encontrados - Código de registro: 2 - Fila actual: 1

NOMBRE	ESTADO
ADMINISTRATIVO	ACTIVO
AUDITORIA INTERNA	ACTIVO
FABRICA DE SOFTWARE	ACTIVO
PRODUCCION	ACTIVO
SEGURIDAD INFORMATICA	ACTIVO
SERVICIOS TECNOLOGICOS	ACTIVO

 Enter: Selección de registro  Esc: Salir




## Pantalla de Presentación de Reportes

REPORTES.

SAP CRYSTAL REPORTS®

2013-11-04

Informe principal



**REGISTRO DE ACTIVIDADES**  
Desde 2013-11-03 Hasta 2013-11-04  
BRIONES INTRIAGO ORLANDO TOMAS(SIGLO21 - DESARROLLADOR)

ODT	TIPO	ACTIVIDAD	INICIO	FIN	ESFUERZO (MINUT)	REGISTRO
<b>2013-11-04</b>						
000000001	GENERICA	0- REGISTRO DE HORAS DE INGRESO	08:00	08:02	2	2013-11-04 00:00
000000976	SOLICITUD	TAREAS DE IMPLEMENTACION IQMS	08:59	09:30	31	2013-11-04 00:00
000016460	SOLICITUD	AUDITORIA DE DOCUMENTACION	09:30	11:20	110	2013-11-04 00:00
000019634	SOLICITUD	TAREAS ADMINISTRATIVAS	11:20	12:50	90	2013-11-04 00:00
000018476	SOLICITUD	CAPACITACION	13:50	16:30	160	2013-11-04 00:00
000009958	ERROR	AUDITORIA DE DOCUMENTACION	16:30	17:15	45	2013-11-04 00:00
000025635	ERROR	PRUEBAS UNITARIAS	17:15	18:25	70	2013-11-04 00:00
000000001	GENERICA	1- REGISTRO DE HORAS DE SALIDA	20:00	20:02	2	2013-11-04 00:00

(8) Actividades, (8.50) Horas de esfuerzo, (510) Minutos de esfuerzo.

Nº de página actual: 1      Nº total de páginas: 1      Factor de zoom: 100%

Esc: Salir

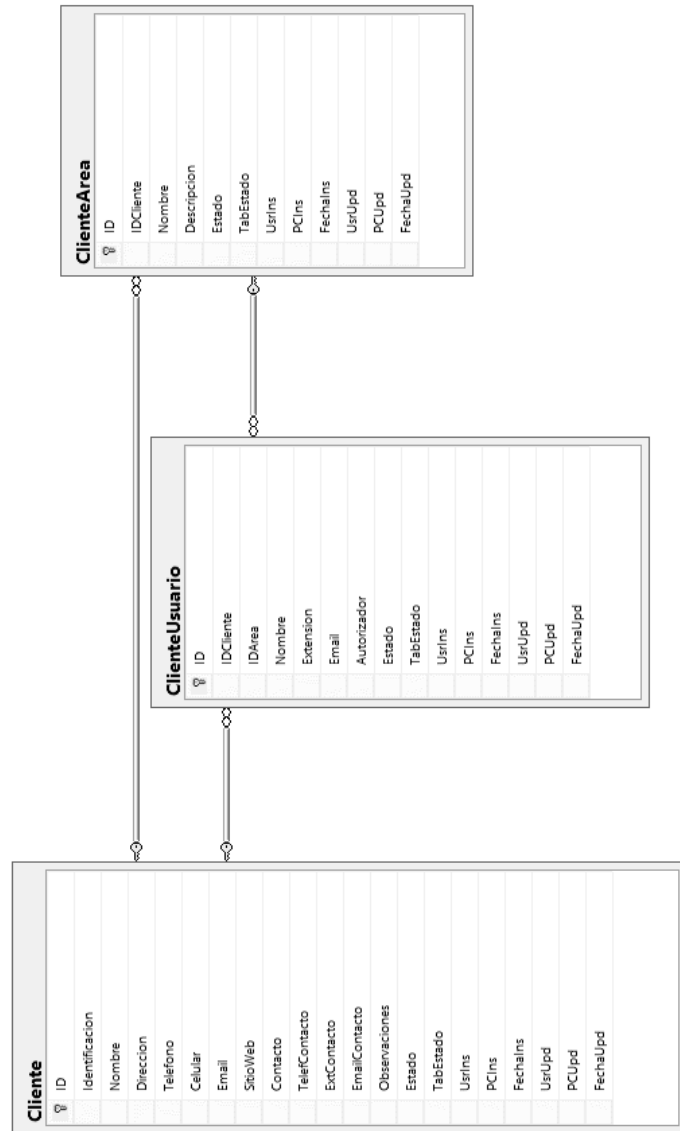


#### 4.2.3.2.2. Diseño Global

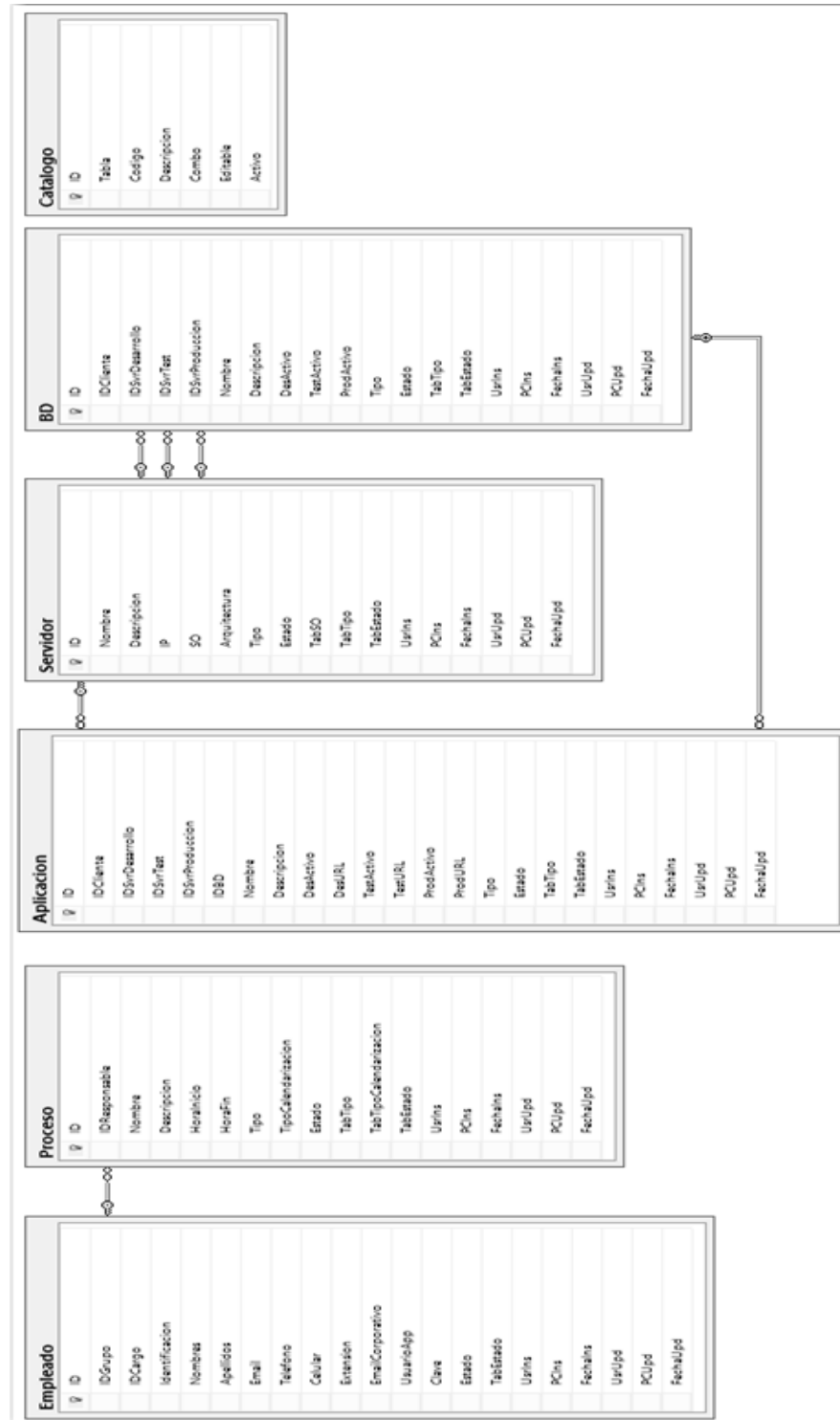
- Módulo Empresa



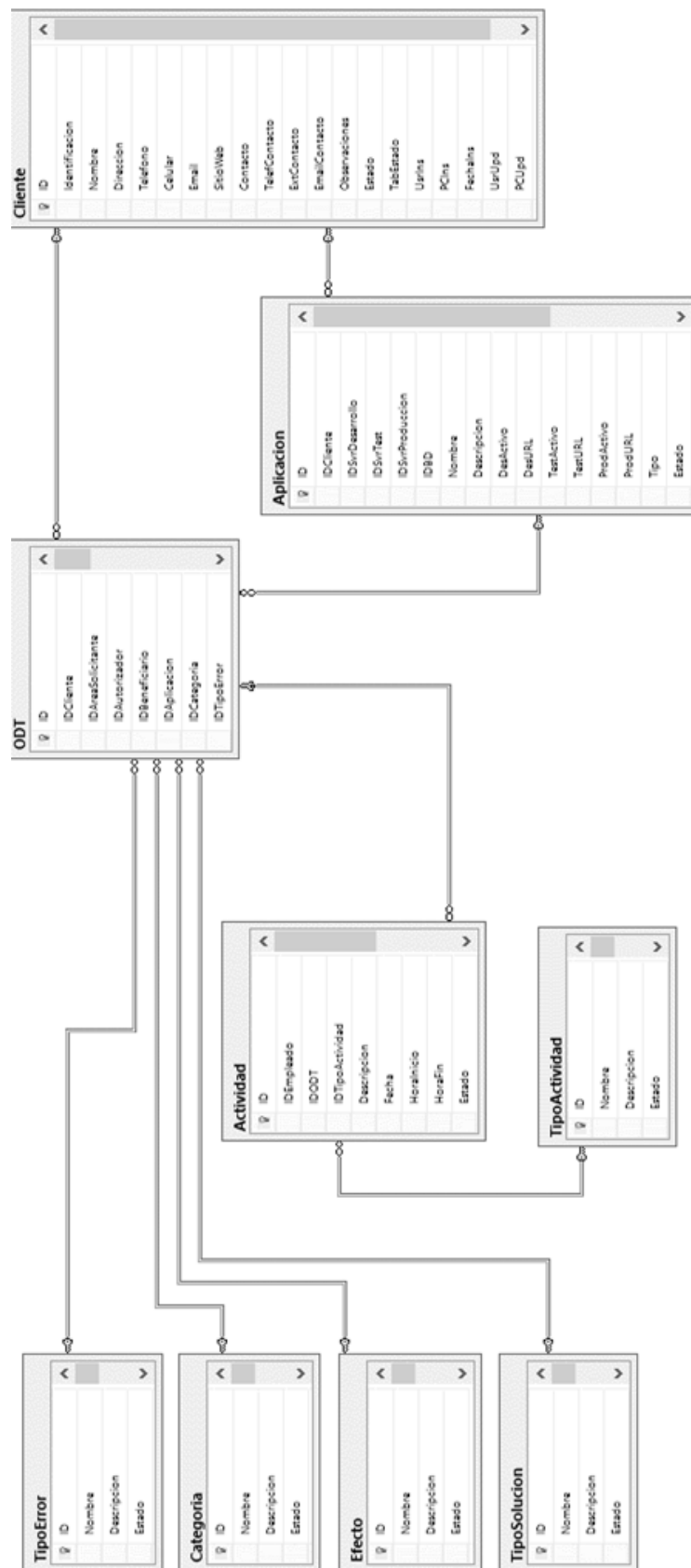
- **Módulo Cliente**



- **Módulo Ecosistema Informático**



- Módulo Gestión



#### 4.2.3.2.3. Diccionario de Datos

**Tabla Actividad**

**Columns**

Key	Column name	Domain	Data type	Not null	Unique	Check	Default
<b>PK</b>	ID		Integer	YES	NO		
<b>FK</b>	IDEmpleado		Integer	YES	NO		
<b>FK</b>	IDODT		Integer	YES	NO		
<b>FK</b>	IDTipoActividad		Integer	YES	NO		
	Descripcion		Varchar (1500)	YES	NO		
	Fecha		Date	YES	NO		
	HoraInicio		Time(7)	YES	NO		
	HoraFin		Time(7)	YES	NO		
	Estado		Varchar (50)	YES	NO		
	TabEstado		Varchar (50)	YES	NO		
	UsrIns		Varchar (15)	YES	NO		
	PCIns		Varchar (25)	YES	NO		
	Fechalns		Smalldatetime	YES	NO		
	UsrUpd		Varchar (15)	NO	NO		
	PCUpd		Varchar (25)	NO	NO		
	FechaUpd		Smalldatetime	NO	NO		

**Relationships**

Constraint (relationship) name	Relationship type	Parent table	Child table	Card.
<b>FK_Actividad_Empleado (FK_Actividad_Empleado)</b>	Non-identifying	Empleado	Actividad	1:N
<b>FK_Actividad_ODT (FK_Actividad_ODT)</b>	Non-identifying	ODT	Actividad	1:N
<b>FK_Actividad_TipoActividad (FK_Actividad_TipoActividad)</b>	Non-identifying	TipoActividad	Actividad	1:N

**Comments**

Registro de actividades referentes a la ODT.

**User-defined variables**

Name	Value
<b>Owner</b>	dbo
<b>Non clustered primary key</b>	No
<b>Primary key fill factor</b>	
<b>Data space for Table</b>	PRIMARY
<b>Data space for Primary key</b>	PRIMARY
<b>Text Filegroup for Table (TEXTIMAGE_ON)</b>	

**Tabla Aplicacion**

**Columns**

Key	Column name	Domain	Data type	Not null	Unique	Check	Default
<b>PK</b>	ID		Integer	YES	NO		
<b>FK</b>	IDCliente		Integer	NO	NO		
<b>FK</b>	IDSvrDesarrollo		Integer	NO	NO		
<b>FK</b>	IDSvrTest		Integer	NO	NO		
<b>FK</b>	IDSvrProduccion		Integer	NO	NO		
<b>FK</b>	IDBD		Integer	YES	NO		
	Nombre		Varchar (50)	YES	NO		
	Descripcion		Varchar (500)	YES	NO		
	DesActivo		Bit	YES	NO		
	DesURL		Varchar (80)	YES	NO		
	TestActivo		Bit	YES	NO		
	TestURL		Varchar (80)	YES	NO		
	ProdActivo		Bit	YES	NO		
	ProdURL		Varchar (80)	YES	NO		
	Tipo		Varchar (50)	YES	NO		
	Estado		Varchar (50)	YES	NO		

TabTipo	Varchar (50)	YES	NO
TabEstado	Varchar (50)	YES	NO
UsrIns	Varchar (15)	YES	NO
PCIns	Varchar (25)	YES	NO
Fechalns	Smalldatetime	YES	NO
UsrUpd	Varchar (15)	NO	NO
PCUpd	Varchar (25)	NO	NO
FechaUpd	Smalldatetime	NO	NO

#### Relationships

Constraint (relationship) name	Relationship type	Parent table	Child table	Card.
<b>FK_ODT_Aplicacion (FK_ODT_Aplicacion)</b>	Non-identifying	Aplicacion	ODT	1:N
<b>FK_Aplicacion_BD (FK_Aplicacion_BD)</b>	Non-identifying	BD	Aplicacion	1:N
<b>FK_Aplicacion_Cliente (FK_Aplicacion_Cliente)</b>	Non-identifying	Cliente	Aplicacion	1:N
<b>FK_Aplicacion_Servidor (FK_Aplicacion_Servidor)</b>	Non-identifying	Servidor	Aplicacion	1:N
<b>FK_Aplicacion_Servidor1 (FK_Aplicacion_Servidor1)</b>	Non-identifying	Servidor	Aplicacion	1:N
<b>FK_Aplicacion_Servidor2 (FK_Aplicacion_Servidor2)</b>	Non-identifying	Servidor	Aplicacion	1:N

#### Comments

Catálogo de aplicaciones.

#### User-defined variables

Name	Value
<b>Owner</b>	dbo
<b>Non clustered primary key</b>	No
<b>Primary key fill factor</b>	
<b>Data space for Table</b>	PRIMARY
<b>Data space for Primary key</b>	PRIMARY
<b>Text Filegroup for Table (TEXTIMAGE_ON)</b>	

### Tabla Area

#### Columns

Key	Column name	Domain	Data type	Not null	Unique	Check	Default	Comments
<b>PK</b>	ID		Integer	YES	NO			ID del registro.
<b>FK</b>	IDDepartamento		Integer	YES	NO			Referencia externa.
	Nombre		Varchar (50)	YES	NO			Descripción del registro.
	Descripcion		Varchar (500)	YES	NO			
	Estado		Varchar (50)	YES	NO			Estado del registro.
	TabEstado		Varchar (50)	YES	NO			Tabla dentro del catálogo.
	UsrIns		Varchar (15)	YES	NO			
	PCIns		Varchar (25)	YES	NO			
	Fechalns		Smalldatetime	YES	NO			
	UsrUpd		Varchar (15)	NO	NO			
	PCUpd		Varchar (25)	NO	NO			
	FechaUpd		Smalldatetime	NO	NO			

#### Relationships

Constraint (relationship) name	Relationship type	Parent table	Child table	Card.
<b>FK_Grupo_Area (FK_Grupo_Area)</b>	Non-identifying	Area	Grupo	1:N
<b>FK_Area_Departamento (FK_Area_Departamento)</b>	Non-identifying	Departamento	Area	1:N

#### Comments

Áreas dentro de los departamentos de la empresa.



**User-defined variables**

Name	Value
<b>Owner</b>	dbo
<b>Non clustered primary key</b>	No
<b>Primary key fill factor</b>	
<b>Data space for Table</b>	PRIMARY
<b>Data space for Primary key</b>	PRIMARY
<b>Text Filegroup for Table (TEXTIMAGE_ON)</b>	

**Tabla Auditoria****Columns**

Key	Column name	Domain	Data type	Not null	Unique	Check	Default
	Fecha		Date	YES	NO		
	Hora		Time(7)	YES	NO		
	UsuarioSql		Varchar (50)	YES	NO		
	UsuarioApp		Varchar (15)	YES	NO		
	PC		Varchar (50)	YES	NO		
	Aplicacion		Varchar (200)	YES	NO		
	Accion		Varchar (50)	YES	NO		
	Descripcion		Varchar (8000)	YES	NO		

**User-defined variables**

Name	Value
<b>Owner</b>	dbo
<b>Non clustered primary key</b>	No
<b>Primary key fill factor</b>	
<b>Data space for Table</b>	PRIMARY
<b>Data space for Primary key</b>	
<b>Text Filegroup for Table (TEXTIMAGE_ON)</b>	PRIMARY

**Tabla BD****Columns**

Key	Column name	Domain	Data type	Not null	Unique	Check	Default
<b>PK</b>	ID		Integer	YES	NO		
<b>FK</b>	IDCliente		Integer	YES	NO		
<b>FK</b>	IDSvrDesarrollo		Integer	NO	NO		
<b>FK</b>	IDSvrTest		Integer	NO	NO		
<b>FK</b>	IDSvrProduccion		Integer	NO	NO		
	Nombre		Varchar (50)	YES	NO		
	Descripcion		Varchar (500)	YES	NO		
	DesActivo		Bit	YES	NO		
	TestActivo		Bit	YES	NO		
	ProdActivo		Bit	YES	NO		
	Tipo		Varchar (50)	YES	NO		
	Estado		Varchar (50)	YES	NO		
	TabTipo		Varchar (50)	YES	NO		
	TabEstado		Varchar (50)	YES	NO		
	UsrIns		Varchar (15)	YES	NO		
	PCIns		Varchar (25)	YES	NO		
	FechaIns		Smalldatetime	YES	NO		
	UsrUpd		Varchar (15)	NO	NO		
	PCUpd		Varchar (25)	NO	NO		
	FechaUpd		Smalldatetime	NO	NO		

**Relationships**

Constraint (relationship) name	Relationship type	Parent table	Child table	Card.
<b>FK_Aplicacion_BD (FK_Aplicacion_BD)</b>	Non-identifying	BD	Aplicacion	1:N
<b>FK_BD_Cliente (FK_BD_Cliente)</b>	Non-identifying	Cliente	BD	1:N
<b>FK_BD_Servidor (FK_BD_Servidor)</b>	Non-identifying	Servidor	BD	1:N
<b>FK_BD_Servidor1 (FK_BD_Servidor1)</b>	Non-identifying	Servidor	BD	1:N
<b>FK_BD_Servidor2 (FK_BD_Servidor2)</b>	Non-identifying	Servidor	BD	1:N

**Comments**

Catálogo de bases de datos.

#### User-defined variables

Name	Value
<b>Owner</b>	dbo
<b>Non clustered primary key</b>	No
<b>Primary key fill factor</b>	
<b>Data space for Table</b>	PRIMARY
<b>Data space for Primary key</b>	PRIMARY
<b>Text Filegroup for Table (TEXTIMAGE_ON)</b>	

### Tabla Cargo

#### Columns

Key	Column name	Domain	Data type	Not null	Unique	Check	Default
<b>PK</b>	ID		Integer	YES	NO		
	Nombre		Varchar (50)	YES	NO		
	Descripcion		Varchar (500)	YES	NO		
	Estado		Varchar (50)	YES	NO		
	TabEstado		Varchar (50)	YES	NO		
	UsrIns		Varchar (15)	YES	NO		
	PCIns		Varchar (25)	YES	NO		
	Fechalns		Smalldatetime	YES	NO		
	UsrUpd		Varchar (15)	NO	NO		
	PCUpd		Varchar (25)	NO	NO		
	FechaUpd		Smalldatetime	NO	NO		

#### Relationships

Constraint (relationship) name	Relationship type	Parent table	Child table	Card.
<b>FK_Empleado_Cargo (FK_Empleado_Cargo)</b>	Non-identifying	Cargo	Empleado	1:N

#### User-defined variables

Name	Value
<b>Owner</b>	dbo
<b>Non clustered primary key</b>	No
<b>Primary key fill factor</b>	
<b>Data space for Table</b>	PRIMARY
<b>Data space for Primary key</b>	PRIMARY
<b>Text Filegroup for Table (TEXTIMAGE_ON)</b>	

### Tabla Catalogo

#### Columns

Key	Column name	Domain	Data type	Not null	Unique	Check	Default	Comments
<b>PK</b>	ID		Integer	YES	NO			
	Tabla		Varchar (50)	YES	NO			Tabla propietaria del catálogo.
	Codigo		Varchar (50)	YES	NO			Codigo del catálogo.
	Descripcion		Varchar (50)	YES	NO			Descripción del catálogo.
	Combo		Bit	YES	NO			Indica si el catálogo se muestra en controles basados en listas.
	Editable		Bit	YES	NO			Indica si el registro es editable.
	Activo		Bit	YES	NO			Estado del registro. True: Activo, False: Inactivo.

#### Comments

Tabla de catálogos de la BD.

#### User-defined variables

Name	Value
<b>Owner</b>	dbo

<b>Non clustered primary key</b>	No
<b>Primary key fill factor</b>	
<b>Data space for Table</b>	PRIMARY
<b>Data space for Primary key</b>	PRIMARY
<b>Text Filegroup for Table (TEXTIMAGE_ON)</b>	

## Tabla Categoria

### Columns

Key	Column name	Domain	Data type	Not null	Unique	Check	Default
<b>PK</b>	ID		Integer	YES	NO		
	Nombre		Varchar (50)	YES	NO		
	Descripcion		Varchar (500)	YES	NO		
	Estado		Varchar (50)	YES	NO		
	TabEstado		Varchar (50)	YES	NO		
	UsrIns		Varchar (15)	YES	NO		
	PCIns		Varchar (25)	YES	NO		
	Fechalns		Smalldatetime	YES	NO		
	UsrUpd		Varchar (15)	NO	NO		
	PCUpd		Varchar (25)	NO	NO		
	FechaUpd		Smalldatetime	NO	NO		

### Relationships

Constraint (relationship) name	Relationship type	Parent table	Child table	Card.
<b>FK_ODT_Categoria (FK_ODT_Categoria)</b>	Non-identifying	Categoria	ODT	1:N

### User-defined variables

Name	Value
<b>Owner</b>	dbo
<b>Non clustered primary key</b>	No
<b>Primary key fill factor</b>	
<b>Data space for Table</b>	PRIMARY
<b>Data space for Primary key</b>	PRIMARY
<b>Text Filegroup for Table (TEXTIMAGE_ON)</b>	

## Tabla Cliente

### Columns

Key	Column name	Domain	Data type	Not null	Unique	Check	Default	Comments
<b>PK</b>	ID		Integer	YES	NO			ID del registro.
	Identificacion		Varchar (13)	YES	NO			Identificación única del cliente.
	Nombre		Varchar (50)	YES	NO			Descripción del registro.
	Direccion		Varchar (80)	YES	NO			Dirección de cliente.
	Telefono		Varchar (25)	YES	NO			
	Celular		Varchar (25)	YES	NO			
	Email		Varchar (35)	YES	NO			Email del cliente.
	SitioWeb		Varchar (50)	YES	NO			Sitio web del cliente.
	Contacto		Varchar (50)	YES	NO			Contacto directo del cliente.
	TelefContacto		Varchar (25)	YES	NO			Teléfono de contacto.
	ExtContacto		Varchar (10)	YES	NO			Extensión telefónica de contacto.
	EmailContacto		Varchar (35)	YES	NO			
	Observaciones		Varchar (250)	YES	NO			
	Estado		Varchar (50)	YES	NO			Estado del registro.
	TabEstado		Varchar (50)	YES	NO			Tabla dentro del catálogo.

UsrIns	Varchar (15)	YES	NO
PCIns	Varchar (25)	YES	NO
Fechalns	Smalldatetime	YES	NO
UsrUpd	Varchar (15)	NO	NO
PCUpd	Varchar (25)	NO	NO
FechaUpd	Smalldatetime	NO	NO

#### Relationships

Constraint (relationship) name	Relationship type	Parent table	Child table	Card.
<b>FK_Aplicacion_Cliente</b> (FK_Aplicacion_Cliente)	Non-identifying	Cliente	Aplicacion	1:N
<b>FK_BD_Cliente</b> (FK_BD_Cliente)	Non-identifying	Cliente	BD	1:N
<b>FK_ClienteArea_Cliente</b> (FK_ClienteArea_Cliente)	Non-identifying	Cliente	ClienteArea	1:N
<b>FK_Usuario_Cliente</b> (FK_Usuario_Cliente)	Non-identifying	Cliente	ClienteUsuario	1:N
<b>FK_ODT_Cliente</b> (FK_ODT_Cliente)	Non-identifying	Cliente	ODT	1:N

#### Comments

Tabla de clientes.

#### User-defined variables

Name	Value
<b>Owner</b>	dbo
<b>Non clustered primary key</b>	No
<b>Primary key fill factor</b>	
<b>Data space for Table</b>	PRIMARY
<b>Data space for Primary key</b>	PRIMARY
<b>Text Filegroup for Table (TEXTIMAGE_ON)</b>	

## Tabla ClienteArea

#### Columns

Key	Column name	Domain	Data type	Not null	Unique	Check	Default
<b>PK</b>	ID		Integer	YES	NO		
<b>FK</b>	IDCliente		Integer	YES	NO		
	Nombre		Varchar (50)	YES	NO		
	Descripcion		Varchar (500)	YES	NO		
	Estado		Varchar (50)	YES	NO		
	TabEstado		Varchar (50)	YES	NO		
	UsrIns		Varchar (15)	YES	NO		
	PCIns		Varchar (25)	YES	NO		
	Fechalns		Smalldatetime	YES	NO		
	UsrUpd		Varchar (15)	NO	NO		
	PCUpd		Varchar (25)	NO	NO		
	FechaUpd		Smalldatetime	NO	NO		

#### Relationships

Constraint (relationship) name	Relationship type	Parent table	Child table	Card.
<b>FK_ClienteArea_Cliente</b> (FK_ClienteArea_Cliente)	Non-identifying	Cliente	ClienteArea	1:N
<b>FK_ClienteUsuario_ClienteArea</b> (FK_ClienteUsuario_ClienteArea)	Non-identifying	ClienteArea	ClienteUsuario	1:N
<b>FK_ODT_ClienteArea</b> (FK_ODT_ClienteArea)	Non-identifying	ClienteArea	ODT	1:N

#### User-defined variables

Name	Value
<b>Owner</b>	dbo
<b>Non clustered primary key</b>	No
<b>Primary key fill factor</b>	
<b>Data space for Table</b>	PRIMARY
<b>Data space for Primary key</b>	PRIMARY
<b>Text Filegroup for Table (TEXTIMAGE_ON)</b>	

## Tabla ClienteUsuario

### Columns

Key	Column name	Domain	Data type	Not null	Unique	Check	Default
<b>PK</b>	ID		Integer	YES	NO		
<b>FK</b>	IDCliente		Integer	YES	NO		
<b>FK</b>	IDArea		Integer	YES	NO		
	Nombre		Varchar (50)	YES	NO		
	Extension		Varchar (10)	YES	NO		
	Email		Varchar (35)	YES	NO		
	Autorizador		Bit	YES	NO		
	Estado		Varchar (50)	YES	NO		
	TabEstado		Varchar (50)	YES	NO		
	UsrIns		Varchar (15)	YES	NO		
	PCIns		Varchar (25)	YES	NO		
	Fechalns		Smalldatetime	YES	NO		
	UsrUpd		Varchar (15)	NO	NO		
	PCUpd		Varchar (25)	NO	NO		
	FechaUpd		Smalldatetime	NO	NO		

### Relationships

Constraint (relationship) name	Relationship type	Parent table	Child table	Card.
<b>FK_Usuario_Cliente (FK_Usuario_Cliente)</b>	Non-identifying	Cliente	ClienteUsuario	1:N
<b>FK_ClienteUsuario_ClienteArea (FK_ClienteUsuario_ClienteArea)</b>	Non-identifying	ClienteArea	ClienteUsuario	1:N
<b>FK_ODT_ClienteUsuario (FK_ODT_ClienteUsuario)</b>	Non-identifying	ClienteUsuario	ODT	1:N
<b>FK_ODT_ClienteUsuario1 (FK_ODT_ClienteUsuario1)</b>	Non-identifying	ClienteUsuario	ODT	1:N

### User-defined variables

Name	Value
<b>Owner</b>	dbo
<b>Non clustered primary key</b>	No
<b>Primary key fill factor</b>	
<b>Data space for Table</b>	PRIMARY
<b>Data space for Primary key</b>	PRIMARY
<b>Text Filegroup for Table (TEXTIMAGE_ON)</b>	

## Tabla Departamento

### Columns

Key	Column name	Domain	Data type	Not null	Unique	Check	Default	Comments
<b>PK</b>	ID		Integer	YES	NO			ID del registro.
	Nombre		Varchar (50)	YES	NO			Descripción del registro.
	Descripcion		Varchar (500)	YES	NO			
	Estado		Varchar (50)	YES	NO			Estado del registro.
	TabEstado		Varchar (50)	YES	NO			Tabla dentro del catálogo.
	UsrIns		Varchar (15)	YES	NO			
	PCIns		Varchar (25)	YES	NO			
	Fechalns		Smalldatetime	YES	NO			
	UsrUpd		Varchar (15)	NO	NO			
	PCUpd		Varchar (25)	NO	NO			
	FechaUpd		Smalldatetime	NO	NO			

### Relationships

Constraint (relationship) name	Relationship type	Parent table	Child table	Card.
<b>FK_Area_Departamento (FK_Area_Departamento)</b>	Non-identifying	Departamento	Area	1:N

### Comments

Departamento en la Empresa.

#### User-defined variables

Name	Value
<b>Owner</b>	dbo
<b>Non clustered primary key</b>	No
<b>Primary key fill factor</b>	
<b>Data space for Table</b>	PRIMARY
<b>Data space for Primary key</b>	PRIMARY
<b>Text Filegroup for Table (TEXTIMAGE_ON)</b>	

### Tabla Efecto

#### Columns

Key	Column name	Domain	Data type	Not null	Unique	Check	Default
<b>PK</b>	ID		Integer	YES	NO		
	Nombre		Varchar (50)	YES	NO		
	Descripcion		Varchar (500)	YES	NO		
	Estado		Varchar (50)	YES	NO		
	TabEstado		Varchar (50)	YES	NO		
	UsrIns		Varchar (15)	YES	NO		
	PCIns		Varchar (25)	YES	NO		
	Fechalns		Smalldatetime	YES	NO		
	UsrUpd		Varchar (15)	NO	NO		
	PCUpd		Varchar (25)	NO	NO		
	FechaUpd		Smalldatetime	NO	NO		

#### Relationships

Constraint (relationship) name	Relationship type	Parent table	Child table	Card.
<b>FK_ODT_Efecto (FK_ODT_Efecto)</b>	Non-identifying	Efecto	ODT	1:N

#### User-defined variables

Name	Value
<b>Owner</b>	dbo
<b>Non clustered primary key</b>	No
<b>Primary key fill factor</b>	
<b>Data space for Table</b>	PRIMARY
<b>Data space for Primary key</b>	PRIMARY
<b>Text Filegroup for Table (TEXTIMAGE_ON)</b>	

### Tabla Empleado

#### Columns

Key	Column name	Domain	Data type	Not null	Unique	Check	Default
<b>PK</b>	ID		Integer	YES	NO		
<b>FK</b>	IDGrupo		Integer	YES	NO		
<b>FK</b>	IDCargo		Integer	YES	NO		
	Identificacion		Varchar (10)	YES	NO		
	Nombres		Varchar (30)	YES	NO		
	Apellidos		Varchar (30)	YES	NO		
	Email		Varchar (35)	YES	NO		
	Telefono		Varchar (25)	YES	NO		
	Celular		Varchar (25)	YES	NO		
	Extension		Varchar (10)	YES	NO		
	EmailCorporativo		Varchar (35)	YES	NO		
	UsuarioApp		Varchar (15)	YES	NO		
	Clave		Varbinary (max)	YES	NO		
	Estado		Varchar (50)	YES	NO		
	TabEstado		Varchar (50)	YES	NO		
	UsrIns		Varchar (15)	YES	NO		
	PCIns		Varchar (25)	YES	NO		
	Fechalns		Smalldatetime	YES	NO		
	UsrUpd		Varchar (15)	NO	NO		
	PCUpd		Varchar (25)	NO	NO		

FechaUpd	Smalldatetime	NO	NO	
Relationships				
Constraint (relationship) name	Relationship type	Parent table	Child table	Card.
FK_Empleado_Cargo (FK_Empleado_Cargo)	Non-identifying	Cargo	Empleado	1:N
FK_Actividad_Empleado (FK_Actividad_Empleado)	Non-identifying	Empleado	Actividad	1:N
FK_ODT_Empleado (FK_ODT_Empleado)	Non-identifying	Empleado	ODT	1:N
FK_ODT_Empleado1 (FK_ODT_Empleado1)	Non-identifying	Empleado	ODT	1:N
FK_Proceso_Empleado (FK_Proceso_Empleado)	Non-identifying	Empleado	Proceso	1:N
FK_Empleado_Grupo (FK_Empleado_Grupo)	Non-identifying	Grupo	Empleado	1:N
User-defined variables				
Name	Value			
Owner	dbo			
Non clustered primary key	No			
Primary key fill factor				
Data space for Table	PRIMARY			
Data space for Primary key	PRIMARY			
Text Filegroup for Table (TEXTIMAGE ON)	PRIMARY			

## Tabla Grupo

Columns								
Key	Column name	Domain	Data type	Not null	Unique	Check	Default	Comments
PK	ID		Integer	YES	NO			ID del registro.
FK	IDArea		Integer	YES	NO			Referencia externa.
	Nombre		Varchar (50)	YES	NO			Descripción del registro.
	Descripcion		Varchar (500)	YES	NO			
	Estado		Varchar (50)	YES	NO			Estado del registro.
	TabEstado		Varchar (50)	YES	NO			Tabla dentro del catálogo.
	UsrIns		Varchar (15)	YES	NO			
	PCIns		Varchar (25)	YES	NO			
	Fechalns		Smalldatetime	YES	NO			
	UsrUpd		Varchar (15)	NO	NO			
	PCUpd		Varchar (25)	NO	NO			
	FechaUpd		Smalldatetime	NO	NO			
Relationships								
Constraint (relationship) name				Relationship type		Parent table	Child table	Card.
FK_Grupo_Area (FK_Grupo_Area)				Non-identifying		Area	Grupo	1:N
FK_Empleado_Grupo (FK_Empleado_Grupo)				Non-identifying		Grupo	Empleado	1:N
Comments								
Grupo dentro del área.								
User-defined variables								
Name				Value				
Owner				dbo				
Non clustered primary key				No				
Primary key fill factor								
Data space for Table				PRIMARY				
Data space for Primary key				PRIMARY				
Text Filegroup for Table (TEXTIMAGE ON)								

## Tabla ODT

<b>Columns</b>							
Key	Column name	Domain	Data type	Not null	Unique	Check	Default
<b>PK</b>	ID		Integer	YES	NO		

<b>FK</b>	IDCliente	Integer	YES	NO
<b>FK</b>	IDAreaSolicitante	Integer	YES	NO
<b>FK</b>	IDAutorizador	Integer	YES	NO
<b>FK</b>	IDBeneficiario	Integer	YES	NO
<b>FK</b>	IDAplicacion	Integer	YES	NO
<b>FK</b>	IDCategoria	Integer	YES	NO
<b>FK</b>	IDTipoError	Integer	YES	NO
<b>FK</b>	IDEfecto	Integer	YES	NO
<b>FK</b>	IDEmpleado	Integer	YES	NO
<b>FK</b>	IDResolutor	Integer	NO	NO
<b>FK</b>	IDTipoSolucion	Integer	NO	NO
	Prioridad	Varchar (50)	YES	NO
	Tipo	Varchar (50)	YES	NO
	AplicaSLA	Bit	YES	NO
	FechaSLA	Smalldatetime	NO	NO
	Descripcion	Varchar (2500)	YES	NO
	Solucionado	Bit	YES	NO
	FechaSolucion	Smalldatetime	NO	NO
	Comentario	Varchar (500)	YES	NO
	Estado	Varchar (50)	YES	NO
	TabPrioridad	Varchar (50)	YES	NO
	TabTipo	Varchar (50)	YES	NO
	TabEstado	Varchar (50)	YES	NO
	UsrIns	Varchar (15)	YES	NO
	PCIns	Varchar (25)	YES	NO
	FechaIns	Smalldatetime	YES	NO
	UsrUpd	Varchar (15)	NO	NO
	PCUpd	Varchar (25)	NO	NO
	FechaUpd	Smalldatetime	NO	NO

#### Relationships

Constraint (relationship) name	Relationship type	Parent table	Child table	Card.
<b>FK_ODT_Aplicacion (FK_ODT_Aplicacion)</b>	Non-identifying	Aplicacion	ODT	1:N
<b>FK_ODT_Categoria (FK_ODT_Categoria)</b>	Non-identifying	Categoria	ODT	1:N
<b>FK_ODT_Cliente (FK_ODT_Cliente)</b>	Non-identifying	Cliente	ODT	1:N
<b>FK_ODT_ClienteArea (FK_ODT_ClienteArea)</b>	Non-identifying	ClienteArea	ODT	1:N
<b>FK_ODT_ClienteUsuario (FK_ODT_ClienteUsuario)</b>	Non-identifying	ClienteUsuario	ODT	1:N
<b>FK_ODT_ClienteUsuario1 (FK_ODT_ClienteUsuario1)</b>	Non-identifying	ClienteUsuario	ODT	1:N
<b>FK_ODT_Efecto (FK_ODT_Efecto)</b>	Non-identifying	Efecto	ODT	1:N
<b>FK_ODT_Empleado (FK_ODT_Empleado)</b>	Non-identifying	Empleado	ODT	1:N
<b>FK_ODT_Empleado1 (FK_ODT_Empleado1)</b>	Non-identifying	Empleado	ODT	1:N
<b>FK_Actividad_ODT (FK_Actividad_ODT)</b>	Non-identifying	ODT	Actividad	1:N
<b>FK_ODT_TipoError (FK_ODT_TipoError)</b>	Non-identifying	TipoError	ODT	1:N
<b>FK_ODT_TipoSolucion (FK_ODT_TipoSolucion)</b>	Non-identifying	TipoSolucion	ODT	1:N

#### User-defined variables

Name	Value
<b>Owner</b>	dbo
<b>Non clustered primary key</b>	No
<b>Primary key fill factor</b>	
<b>Data space for Table</b>	PRIMARY
<b>Data space for Primary key</b>	PRIMARY
<b>Text Filegroup for Table (TEXTIMAGE_ON)</b>	

### Tabla Proceso

#### Columns

Key	Column name	Domain	Data type	Not null	Unique	Check	Default
<b>PK</b>	ID		Integer	YES	NO		
<b>FK</b>	IDResponsable		Integer	YES	NO		
	Nombre		Varchar (50)	YES	NO		



Descripcion	Varchar (500)	YES	NO
HoraInicio	Time(7)	YES	NO
HoraFin	Time(7)	YES	NO
Tipo	Varchar (50)	YES	NO
TipoCalendarizacion	Varchar (50)	YES	NO
Estado	Varchar (50)	YES	NO
TabTipo	Varchar (50)	YES	NO
TabTipoCalendarizacion	Varchar (50)	YES	NO
TabEstado	Varchar (50)	YES	NO
UsrIns	Varchar (15)	YES	NO
PCIns	Varchar (25)	YES	NO
Fechalns	Smalldatetime	YES	NO
UsrUpd	Varchar (15)	NO	NO
PCUpd	Varchar (25)	NO	NO
FechaUpd	Smalldatetime	NO	NO

#### Relationships

Constraint (relationship) name	Relationship type	Parent table	Child table	Card.
<b>FK_Proceso_Empleado (FK_Proceso_Empleado)</b>	Non-identifying	Empleado	Proceso	1:N

#### Comments

Catálogos de Procesos. Ejem.: Procesos Batch, Procesos de Cierre.

#### User-defined variables

Name	Value
<b>Owner</b>	dbo
<b>Non clustered primary key</b>	No
<b>Primary key fill factor</b>	
<b>Data space for Table</b>	PRIMARY
<b>Data space for Primary key</b>	PRIMARY
<b>Text Filegroup for Table (TEXTIMAGE_ON)</b>	

### Tabla Servidor

#### Columns

Key	Column name	Domain	Data type	Not null	Unique	Check	Default
<b>PK</b>	ID		Integer	YES	NO		
	Nombre		Varchar (50)	YES	NO		
	Descripcion		Varchar (500)	YES	NO		
	IP		Varchar (15)	YES	NO		
	SO		Varchar (50)	YES	NO		
	Arquitectura		Tinyint	YES	NO		
	Tipo		Varchar (50)	YES	NO		
	Estado		Varchar (50)	YES	NO		
	TabSO		Varchar (50)	YES	NO		
	TabTipo		Varchar (50)	YES	NO		
	TabEstado		Varchar (50)	YES	NO		
	UsrIns		Varchar (15)	YES	NO		
	PCIns		Varchar (25)	YES	NO		
	Fechalns		Smalldatetime	YES	NO		
	UsrUpd		Varchar (15)	NO	NO		
	PCUpd		Varchar (25)	NO	NO		
	FechaUpd		Smalldatetime	NO	NO		

#### Relationships

Constraint (relationship) name	Relationship type	Parent table	Child table	Card.
<b>FK_Aplicacion_Servidor (FK_Aplicacion_Servidor)</b>	Non-identifying	Servidor	Aplicacion	1:N
<b>FK_Aplicacion_Servidor1 (FK_Aplicacion_Servidor1)</b>	Non-identifying	Servidor	Aplicacion	1:N
<b>FK_Aplicacion_Servidor2 (FK_Aplicacion_Servidor2)</b>	Non-identifying	Servidor	Aplicacion	1:N
<b>FK_BD_Servidor (FK_BD_Servidor)</b>	Non-identifying	Servidor	BD	1:N

<b>FK_BD_Servidor1 (FK_BD_Servidor1)</b>	Non-identifying	Servidor	BD	1:N
<b>FK_BD_Servidor2 (FK_BD_Servidor2)</b>	Non-identifying	Servidor	BD	1:N

#### Comments

Catálogo de servidores.

#### User-defined variables

Name	Value
<b>Owner</b>	dbo
<b>Non clustered primary key</b>	No
<b>Primary key fill factor</b>	
<b>Data space for Table</b>	PRIMARY
<b>Data space for Primary key</b>	PRIMARY
<b>Text Filegroup for Table (TEXTIMAGE_ON)</b>	

## Tabla TipoActividad

#### Columns

Key	Column name	Domain	Data type	Not null	Unique	Check	Default
<b>PK</b>	ID		Integer	YES	NO		
	Nombre		Varchar (50)	YES	NO		
	Descripcion		Varchar (500)	YES	NO		
	Estado		Varchar (50)	YES	NO		
	TabEstado		Varchar (50)	YES	NO		
	UsrIns		Varchar (15)	YES	NO		
	PCIns		Varchar (25)	YES	NO		
	Fechalns		Smalldatetime	YES	NO		
	UsrUpd		Varchar (15)	NO	NO		
	PCUpd		Varchar (25)	NO	NO		
	FechaUpd		Smalldatetime	NO	NO		

#### Relationships

Constraint (relationship) name	Relationship type	Parent table	Child table	Card.
<b>FK_Actividad_TipoActividad (FK_Actividad_TipoActividad)</b>	Non-identifying	TipoActividad	Actividad	1:N

#### Comments

Catálogo de Tipos de Actividades.

#### User-defined variables

Name	Value
<b>Owner</b>	dbo
<b>Non clustered primary key</b>	No
<b>Primary key fill factor</b>	
<b>Data space for Table</b>	PRIMARY
<b>Data space for Primary key</b>	PRIMARY
<b>Text Filegroup for Table (TEXTIMAGE_ON)</b>	

## Tabla TipoError

#### Columns

Key	Column name	Domain	Data type	Not null	Unique	Check	Default
<b>PK</b>	ID		Integer	YES	NO		
	Nombre		Varchar (50)	YES	NO		
	Descripcion		Varchar (500)	YES	NO		
	Estado		Varchar (50)	YES	NO		
	TabEstado		Varchar (50)	YES	NO		
	UsrIns		Varchar (15)	YES	NO		
	PCIns		Varchar (25)	YES	NO		
	Fechalns		Smalldatetime	YES	NO		
	UsrUpd		Varchar (15)	NO	NO		
	PCUpd		Varchar (25)	NO	NO		
	FechaUpd		Smalldatetime	NO	NO		

#### Relationships

Constraint (relationship) name	Relationship type	Parent table	Child table	Card.
--------------------------------	-------------------	--------------	-------------	-------

<b>FK_ODT_TipoError (FK_ODT_TipoError)</b>	Non-identifying	TipoError	ODT	1:N
<b>Comments</b>				
Catálogo de Tipos de Error.				
<b>User-defined variables</b>				
Name		Value		
Owner		dbo		
Non clustered primary key		No		
Primary key fill factor				
Data space for Table		PRIMARY		
Data space for Primary key		PRIMARY		
Text Filegroup for Table (TEXTIMAGE_ON)				

## Tabla TipoSolucion

Columns							
Key	Column name	Domain	Data type	Not null	Unique	Check	Default
PK	ID		Integer	YES	NO		
	Nombre		Varchar (50)	YES	NO		
	Descripcion		Varchar (500)	YES	NO		
	Estado		Varchar (50)	YES	NO		
	TabEstado		Varchar (50)	YES	NO		
	UsrIns		Varchar (15)	YES	NO		
	PCIns		Varchar (25)	YES	NO		
	Fechalns		Smalldatetime	YES	NO		
	UsrUpd		Varchar (15)	NO	NO		
	PCUpd		Varchar (25)	NO	NO		
	FechaUpd		Smalldatetime	NO	NO		
Relationships							
Constraint (relationship) name			Relationship type	Parent table	Child table	Card.	
FK_ODT_TipoSolucion (FK_ODT_TipoSolucion)			Non-identifying	TipoSolucion	ODT	1:N	
Comments							
Catálogo de Tipos de Soluciones.							
User-defined variables							
Name			Value				
Owner			dbo				
Non clustered primary key			No				
Primary key fill factor							
Data space for Table			PRIMARY				
Data space for Primary key			PRIMARY				
Text Filegroup for Table (TEXTIMAGE_ON)							

### 4.2.4. Desarrollo

En esta fase se realiza la codificación del sistema y la base de datos. Se aplican el estándar de programación de código fuente CamelCase (Tipo UpperCamelCase). Debido a la estructura similar de los procedimientos almacenados, funciones, clases, formulario; se describe el código fuente correspondiente al objeto Departamento.

#### 4.2.4.1. Base de Datos

- **Tablas**

```
USE [SAGMA]
GO
```

```
/***** Object: Table [dbo].[Departamento]    Script Date: 2014-01-05 22:57:14 *****/
SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON
GO
```

```
SET ANSI_PADDING ON
GO
```

```
CREATE TABLE [dbo].[Departamento](
    [ID] [int] IDENTITY(1,1) NOT NULL,
    [Nombre] [varchar](50) NOT NULL,
    [Descripcion] [varchar](500) NOT NULL,
    [Estado] [varchar](50) NOT NULL,
    [TabEstado] [varchar](50) NOT NULL,
    [UsrIns] [varchar](15) NOT NULL,
    [PCIns] [varchar](25) NOT NULL,
    [FechaIns] [smalldatetime] NOT NULL,
    [UsrUpd] [varchar](15) NULL,
    [PCUpd] [varchar](25) NULL,
    [FechaUpd] [smalldatetime] NULL,
    CONSTRAINT [PK_Departamento] PRIMARY KEY CLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS =
ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

```
GO
```

```
SET ANSI_PADDING OFF
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'ID del registro.' ,
@level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'Departamento',
@level2type=N'COLUMN',@level2name=N'ID'
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Descripción del registro.' ,
@level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'Departamento',
@level2type=N'COLUMN',@level2name=N'Nombre'
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Estado del registro.' ,
@level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'Departamento',
@level2type=N'COLUMN',@level2name=N'Estado'
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Tabla dentro del catálogo.' ,
@level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'Departamento',
@level2type=N'COLUMN',@level2name=N'TabEstado'
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Departamento en la Empresa.' ,
@level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'Departamento'
GO
```

- **Procedimientos Almacenados**

- ✓ **Insertar**

```

USE [SAGMA]
GO
/***** Object:  StoredProcedure [dbo].[Ins_Departamento]    Script Date: 2014-01-05 22:43:33
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
--Autor          : Orlando Briones IntriaGo
--Tipo            : SP - INS
--Propósito      : Insertar un nuevo registro
--Fecha de Creaión : 2013-11-20
--Fecha de Modificación :
ALTER PROCEDURE [dbo].[Ins_Departamento]
    @Nombre AS VARCHAR(50),
    @Descripcion AS VARCHAR(500),
    @Estado AS VARCHAR(50),
    @Usuario AS VARCHAR(15)
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @Result INT

    BEGIN TRY
        BEGIN TRANSACTION Trans
        IF EXISTS(SELECT ID FROM dbo.Departamento WITH (NOLOCK) WHERE Nombre =
@Nombre AND Estado != 'ELI') BEGIN
            SET @Result = -2
        END ELSE BEGIN
            INSERT INTO dbo.Departamento
            (
                Nombre, Descripcion, Estado,
                TabEstado,
                UsrIns, PCIns, FechaIns
            )
            VALUES
            (
                @Nombre, @Descripcion, @Estado,
                'ESTGENERAL',
                @Usuario, HOST_NAME(), GETDATE()
            )

            SET @Result = SCOPE_IDENTITY()
        END

        SELECT ISNULL(@Result, -1)
        COMMIT TRANSACTION Trans
    END TRY
    BEGIN CATCH
        DECLARE @TextError VARCHAR(800)
        ROLLBACK TRANSACTION Trans

        SET @TextError = 'Se ha producido el siguiente error al tratar de realizar la
transacción:'
        SET @TextError = @TextError + CHAR(13) + 'Error Nº: ' +
LTRIM(RTRIM(ERROR_NUMBER()))
        SET @TextError = @TextError + CHAR(13) + 'Línea Nº: ' +
LTRIM(RTRIM(ERROR_LINE()))
        SET @TextError = @TextError + CHAR(13) + 'Procedimiento: ' + ERROR_PROCEDURE()
        SET @TextError = @TextError + CHAR(13) + 'Texto del Error: ' + ERROR_MESSAGE()
    
```

```

        RAISERROR(@TextError, 16, 1)
        SELECT (-1)
    END CATCH
END

```

## ✓ Actualizar

```

USE [SAGMA]
GO
/***** Object:  StoredProcedure [dbo].[Upd_Departamento]    Script Date: 2014-01-05 22:47:00
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
--Autor          : Orlando Briones Intriaño
--Tipo           : SP - UPD
--Propósito      : Actualizar un registro existente
--Fecha de Creación : 2013-11-20
--Fecha de Modificación :
ALTER PROCEDURE [dbo].[Upd_Departamento]
    @Accion AS VARCHAR(50),
    @ID AS INT,
    @Nombre AS VARCHAR(50),
    @Descripcion AS VARCHAR(500),
    @Estado AS VARCHAR(5),
    @Usuario AS VARCHAR(15)
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @Result INT

    BEGIN TRY
        BEGIN TRANSACTION Trans
        IF @Accion = 'TODO' BEGIN
            IF EXISTS(SELECT ID FROM dbo.Departamento WITH (NOLOCK) WHERE ID
!= @ID AND Nombre = @Nombre AND Estado != 'ELI') BEGIN
                SET @Result = -2
            END ELSE BEGIN
                UPDATE dbo.Departamento
                SET
                    Nombre = @Nombre,
                    Descripcion = @Descripcion,
                    Estado = @Estado,
                    UsrUpd = @Usuario,
                    PCUpd = HOST_NAME(),
                    FechaUpd = GETDATE()

                WHERE
                    ID = @ID

                SET @Result = @@ROWCOUNT
            END
        END

        SELECT ISNULL(@Result, -1)
        COMMIT TRANSACTION Trans
    END TRY
    BEGIN CATCH
        DECLARE @TextError VARCHAR(800)
        ROLLBACK TRANSACTION Trans

        SET @TextError = 'Se ha producido el siguiente error al tratar de realizar la
transacción:'
        SET @TextError = @TextError + CHAR(13) + 'Error Nº: ' +
LTRIM(RTRIM(ERROR_NUMBER()))
        SET @TextError = @TextError + CHAR(13) + 'Línea Nº: ' +
LTRIM(RTRIM(ERROR_LINE()))
        SET @TextError = @TextError + CHAR(13) + 'Procedimiento: ' + ERROR_PROCEDURE()
    
```

```

        SET @TextError = @TextError + CHAR(13) + 'Texto del Error: ' + ERROR_MESSAGE()

        RAISERROR(@TextError, 16, 1)
        SELECT (-1)
    END CATCH
END

```

## ✓ Eliminar

```

USE [SAGMA]
GO
/***** Object:  StoredProcedure [dbo].[Del_Departamento]    Script Date: 2014-01-05 22:49:28
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
--Autor          : Orlando Briones Intriago
--Tipo            : SP - DEL
--Propósito       : Eliminar un registro existente
--Fecha de Creación : 2013-11-20
--Fecha de Modificación :
ALTER PROCEDURE [dbo].[Del_Departamento]
    @ID AS INT = NULL,
    @Usuario AS VARCHAR(15)
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @Result INT

    BEGIN TRY
        BEGIN TRANSACTION Trans
            UPDATE dbo.Departamento
                SET
                    Estado = 'ELI',
                    UsrUpd = @Usuario,
                    PCUpd = HOST_NAME(),
                    FechaUpd = GETDATE()
                WHERE
                    ID = @ID

            SET @Result = @@ROWCOUNT

            SELECT ISNULL(@Result, -1)
        COMMIT TRANSACTION Trans
    END TRY
    BEGIN CATCH
        DECLARE @TextError VARCHAR(800)
        ROLLBACK TRANSACTION Trans

        SET @TextError = 'Se ha producido el siguiente error al tratar de realizar la
transacción:'
        SET @TextError = @TextError + CHAR(13) + 'Error Nº: ' +
LTRIM(RTRIM(ERROR_NUMBER()))
        SET @TextError = @TextError + CHAR(13) + 'Línea Nº: ' +
LTRIM(RTRIM(ERROR_LINE()))
        SET @TextError = @TextError + CHAR(13) + 'Procedimiento: ' + ERROR_PROCEDURE()
        SET @TextError = @TextError + CHAR(13) + 'Texto del Error: ' + ERROR_MESSAGE()

        RAISERROR(@TextError, 16, 1)
        SELECT (-1)
    END CATCH
END

```

## ✓ Seleccionar

```
USE [SAGMA]
GO
/***** Object: StoredProcedure [dbo].[Sel_Departamento]    Script Date: 2014-01-05 22:50:11
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
--Autor          : Orlando Briones Intriaño
--Tipo           : SP - SEL
--Propósito      : Seleccionar registros de la BD
--Fecha de Creación : 2013-11-20
--Fecha de Modificación :
ALTER PROCEDURE [dbo].[Sel_Departamento]
    @Accion AS VARCHAR(50),
    @ID AS INT = NULL,
    @Texto AS VARCHAR(50) = NULL,
    @Estado AS VARCHAR(25) = NULL
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @Result INT

    BEGIN TRY
        IF @Accion = 'PORID' BEGIN
            SELECT
                D.ID, D.Nombre, D.Descripcion, D.Estado
            FROM
                dbo.Departamento AS D WITH (NOLOCK)
            WHERE
                D.ID = @ID
                AND D.Estado != 'ELI'
        END ELSE IF @Accion = 'BUSQGENERAL' BEGIN
            SELECT
                D.ID AS 'ID', D.Nombre AS 'NOMBRE',
                dbo.Fun_GetCatalogoDescpcion(D.TabEstado, D.Estado) AS 'ESTADO'
            FROM
                dbo.Departamento AS D WITH (NOLOCK)
            WHERE
                @Estado LIKE '%' + D.Estado + '%'
                AND (D.Nombre LIKE '%' + @Texto + '%' OR D.Descripcion LIKE '%'
+ @Texto + '%')
            ORDER BY
                D.Nombre
        END ELSE IF @Accion = 'COMBO' BEGIN
            SELECT
                D.ID, D.Nombre
            FROM
                dbo.Departamento AS D WITH (NOLOCK)
            WHERE
                D.Estado != 'ELI'
            ORDER BY
                D.Nombre
        END
    END TRY
    BEGIN CATCH
        DECLARE @TextError VARCHAR(800)

        SET @TextError = 'Se ha producido el siguiente error al tratar de realizar la
transacción:'
        SET @TextError = @TextError + CHAR(13) + 'Error Nº: ' +
LTRIM(RTRIM(ERROR_NUMBER()))
        SET @TextError = @TextError + CHAR(13) + 'Línea Nº: ' +
LTRIM(RTRIM(ERROR_LINE()))
        SET @TextError = @TextError + CHAR(13) + 'Procedimiento: ' + ERROR_PROCEDURE()
```



```

        SET @TextError = @TextError + CHAR(13) + 'Texto del Error: ' + ERROR_MESSAGE()

        RAISERROR(@TextError, 16, 1)
        SELECT (-1)
    END CATCH
END

```

## • Funciones

```

USE [SAGMA]
GO
/***** Object:  UserDefinedFunction [dbo].[Fun_GetArquitectura]    Script Date: 2014-01-05
22:55:34 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

ALTER FUNCTION [dbo].[Fun_GetArquitectura]
(
    @Value AS TINYINT
)
RETURNS VARCHAR(50)
AS
BEGIN
    DECLARE @Result AS VARCHAR(50)

    SET @Result =
        CASE @Value
            WHEN 32 THEN '32 BITS (x32)'
            WHEN 64 THEN '64 BITS (x64)'
            ELSE ''
        END

    RETURN ISNULL(@Result, '')
END

```

### 4.2.4.2. Clases

```

Imports System.Windows.Forms
Imports System.Data.SqlClient
Imports SysBrisa.General
Imports SysBrisa

Public Class Departamento
    Public Accion As String
    Public ID As Integer
    Public Nombre As String
    Public Descripcion As String
    Public Estado As String
    Public TabEstado As String
    Public Usuario As String
    Public ResultTran As Integer

    Public Sub New()
        Try
            With Me
                .Accion = TipoTransaccion.Accion.Guardar
                .ID = 0
                .Nombre = ""
                .Descripcion = ""
                .Estado = ""
                .TabEstado = "ESTGENERAL"
                .Usuario = ""
            End With
        End Try
    End Sub
End Class

```

```

        .ResultTran = 0
    End With
    Catch exApp As Exception
        Dim Bitacora As New Errores.Bitacora(exApp,
"Negocio\Empresa\Departamento", "New", True)
        Bitacora.RegistrarEvento()
    Finally
        'Codigo final
    End Try
End Sub

Public Overrides Function ToString() As String
    Return Me.Nombre
End Function

Public Function Buscar(ByVal iID As Integer) As Boolean
    Dim Result As Boolean = False
    Dim SqlDatos As New Data.Transaccion()
    Dim Tabla As New DataTable()
    Dim Parametros As New List(Of SqlParameter)()

    Try
        Parametros.Add(Data.Parameter.NewParametro("Accion", "PORID",
SqlDbType.VarChar))
        Parametros.Add(Data.Parameter.NewParametro("ID", iID, SqlDbType.Int))
        Tabla = SqlDatos.GetDataTable("dbo.Sel_Departamento", Parametros)

        If Tabla.Rows.Count > 0 AndAlso Tabla.Columns.Count > 0 Then
            Me.ID = IIf(Tabla.Rows(0).IsNull(0), 0, Tabla.Rows(0).Item(0))
            Me.Nombre = IIf(Tabla.Rows(0).IsNull(1), "",
Tabla.Rows(0).Item(1))
            Me.Descripcion = IIf(Tabla.Rows(0).IsNull(2), "",
Tabla.Rows(0).Item(2))
            Me.Estado = IIf(Tabla.Rows(0).IsNull(3), "",
Tabla.Rows(0).Item(3))
            Result = True
        End If
    Catch exApp As Exception
        Dim Bitacora As New Errores.Bitacora(exApp,
"Negocio\Empresa\Departamento", "Buscar", True)
        Bitacora.RegistrarEvento()
    Finally
        SqlDatos.Desconectar()
    End Try

    Return Result
End Function

Public Function Buscar(ByVal iID As Integer, SqlDatos As Data.Transaccion) As Boolean
    Dim Result As Boolean = False
    Dim Tabla As New DataTable()
    Dim Parametros As New List(Of SqlParameter)()

    Try
        Parametros.Add(Data.Parameter.NewParametro("Accion", "PORID",
SqlDbType.VarChar))
        Parametros.Add(Data.Parameter.NewParametro("ID", iID, SqlDbType.Int))
        Tabla = SqlDatos.GetDataTable("dbo.Sel_Departamento", Parametros)

        If Tabla.Rows.Count > 0 AndAlso Tabla.Columns.Count > 0 Then
            Me.ID = IIf(Tabla.Rows(0).IsNull(0), 0, Tabla.Rows(0).Item(0))
            Me.Nombre = IIf(Tabla.Rows(0).IsNull(1), "",
Tabla.Rows(0).Item(1))
            Me.Descripcion = IIf(Tabla.Rows(0).IsNull(2), "",
Tabla.Rows(0).Item(2))
            Me.Estado = IIf(Tabla.Rows(0).IsNull(3), "",
Tabla.Rows(0).Item(3))
            Result = True
        End If
    Catch exApp As Exception
        Dim Bitacora As New Errores.Bitacora(exApp,
"Negocio\Empresa\Departamento", "Buscar", True)
        Bitacora.RegistrarEvento()
    End Try
End Function

```

```

        Finally
    End Try

    Return Result
End Function

Public Function Buscar(ByVal Texto As String, Estado As String, Optional ByRef iID As
Object = Nothing) As Boolean
    Dim Result As Boolean = False
    Dim Parametros As New List(Of SqlParameter)()

    Try
        Parametros.Add(Data.Parameter.NewParameter("Accion", "BUSQGENERAL",
SqlDbType.VarChar))
        Parametros.Add(Data.Parameter.NewParameter("Texto", Texto,
SqlDbType.VarChar))
        Parametros.Add(Data.Parameter.NewParameter("Estado", Estado,
SqlDbType.VarChar))
        Dim Consulta As New SysBrisa.General.Consulta("Departamento",
"dbo.Sel_Departamento", Parametros)
        Consulta.ShowDialog()

        If Consulta.DatoRetorno.Trim.Length > 0 Then
            If Not Object.Equals(iID, Nothing) Then iID =
Consulta.DatoRetorno
                Result = Me.Buscar(Convert.ToInt32(Consulta.DatoRetorno))
            End If
        Catch exApp As Exception
            Dim Bitacora As New Errores.Bitacora(exApp,
"Negocio\Empresa\Departamento", "Buscar", True)
            Bitacora.RegistrarEvento()
        Finally
            'Codigo final
        End Try

        Return Result
    End Function

Public Shared Function Registrar(ByVal Entidad As Departamento, ByVal Accion As
TipoTransaccion.Accion) As Boolean
    Dim Result As Boolean = False

    Try
        If Accion = TipoTransaccion.Accion.Guardar Then
            Result = Entidad.Guardar()
        ElseIf Accion = TipoTransaccion.Accion.Actualizar Then
            Result = Entidad.Actualizar()
        End If
        Catch exApp As Exception
            Dim Bitacora As New Errores.Bitacora(exApp,
"Negocio\Empresa\Departamento", "Registrar", True)
            Bitacora.RegistrarEvento()
        Finally
            'Codigo final
        End Try

        Return Result
    End Function

Private Function Guardar() As Boolean
    Dim Result As Boolean = False
    Dim SqlDatos As New Data.Transaccion()
    Dim Parametros As New List(Of SqlParameter)()

    Try
        Parametros.Add(Data.Parameter.NewParameter("Nombre", Me.Nombre,
SqlDbType.VarChar))
        Parametros.Add(Data.Parameter.NewParameter("Descripcion",
Me.Descripcion, SqlDbType.VarChar))
        Parametros.Add(Data.Parameter.NewParameter("Estado", Me.Estado,
SqlDbType.VarChar))

```

```

        Parametros.Add(Data.Parametro.NewParametro("Usuario", Me.Usuario,
SqlDbType.VarChar))
        Result = SqlDatos.Ejecutar("dbo.Ins_Departamento", Parametros,
Me.ResultTran)

        If Not Object.Equals(Me.ResultTran, Nothing) OrElse Me.ResultTran <> 0
Then
            Select Case Convert.ToInt32(Me.ResultTran)
                Case Is > 0 : MessageBoxBrisa.Informacion("Registro
realizado correctamente con código " & Me.ResultTran.ToString() & ".")
                Case -1 : MessageBoxBrisa.Exclamacion("La transacción no
se realizó porque ocurrió un error en el Servidor de Datos.")
                Case -2 : MessageBoxBrisa.Exclamacion("El registro ya
existe. No se realizó la transacción.")
            End Select
        End If
        Catch exApp As Exception
            Dim Bitacora As New Errores.Bitacora(exApp,
"Negocio\Empresa\Departamento", "Guardar", True)
            Bitacora.RegistrarEvento()
        Finally
            'Codigo final
            SqlDatos.Desconectar()
        End Try

        Return Result
    End Function

    Private Function Actualizar() As Boolean
        Dim Result As Boolean = False
        Dim SqlDatos As New Data.Transaccion()
        Dim Parametros As New List(Of SqlParameter)()

        Try
            Parametros.Add(Data.Parametro.NewParametro("Accion", "TODO",
SqlDbType.VarChar))
            Parametros.Add(Data.Parametro.NewParametro("ID", Me.ID, SqlDbType.Int))
            Parametros.Add(Data.Parametro.NewParametro("Nombre", Me.Nombre,
SqlDbType.VarChar))
            Parametros.Add(Data.Parametro.NewParametro("Descripcion",
Me.Descripcion, SqlDbType.VarChar))
            Parametros.Add(Data.Parametro.NewParametro("Estado", Me.Estado,
SqlDbType.VarChar))
            Parametros.Add(Data.Parametro.NewParametro("Usuario", Me.Usuario,
SqlDbType.VarChar))
            Result = SqlDatos.Ejecutar("dbo.Upd_Departamento", Parametros,
Me.ResultTran)

            If Not Object.Equals(Me.ResultTran, Nothing) Then
                Select Case Convert.ToInt32(Me.ResultTran)
                    Case Is > 0 And Result = True :
                        MessageBoxBrisa.Informacion(Me.ResultTran.ToString() & " registro(s) actualizado(s).")
                    Case -1 : MessageBoxBrisa.Exclamacion("La transacción no
se realizó porque ocurrió un error en el Servidor de Datos.")
                    Case -2 : MessageBoxBrisa.Exclamacion("El registro ya
existe. No se realizó la transacción.")
                End Select
            End If
            Catch exApp As Exception
                Dim Bitacora As New Errores.Bitacora(exApp,
"Negocio\Empresa\Departamento", "Actualizar", True)
                Bitacora.RegistrarEvento()
            Finally
                'Codigo final
                SqlDatos.Desconectar()
            End Try

            Return Result
        End Function

    Public Shared Function Eliminar(iID As Integer, Usuario As String) As Boolean
        Dim Result As Boolean = False
        Dim Entidad As New Departamento()

```

```

        Dim SqlDatos As New Data.Transaccion()
        Dim Parametros As New List(Of SqlParameter)()

        Try
            Parametros.Add(Data.Parameter.NewParametro("ID", iID, SqlDbType.Int))
            Parametros.Add(Data.Parameter.NewParametro("Usuario", Usuario,
SqlDbType.VarChar))
            Result = SqlDatos.Ejecutar("dbo.Del_Departamento", Parametros,
Entidad.ResultTran)

            If Not Object.Equals(Entidad.ResultTran, Nothing) Then
                Select Case Convert.ToInt32(Entidad.ResultTran)
                    Case Is >= 0 And Result = True :
                        MessageBoxBrisa.Informacion(Entidad.ResultTran.ToString() & " registro(s) eliminado(s).")
                    Case -1, -2 : MessageBoxBrisa.Exclamacion("El registro
posee enlaces externos, no se puede eliminar. No se realizó la transacción.")
                End Select
            End If
        Catch exApp As Exception
            Dim Bitacora As New Errores.Bitacora(exApp,
"Negocio\Empresa\Departamento", "Eliminar", False)
            Bitacora.RegistrarEvento()
        Finally
            'Codigo final
            SqlDatos.Desconectar()
        End Try

        Return Result
    End Function

    Public Shared Sub ComboBox(ByRef Combo As ComboBox)
        Dim Parametros As New List(Of SqlClient.SqlParameter)()

        Try
            Parametros.Add(Data.Parameter.NewParametro("Accion", "COMBO",
SqlDbType.VarChar))
            SysBrisa.General.General.SetComboBox(Combo, "dbo.Sel_Departamento",
Parametros)
        Catch exApp As Exception
            Dim Bitacora As New Errores.Bitacora(exApp,
"Negocio\Empresa\Departamento", "ComboBox", True)
            Bitacora.RegistrarEvento()
        Finally
            'Codigo final
        End Try
    End Sub
End Class

```

#### 4.2.4.3. Formulario

```

Imports SysBrisa.General
Imports DevComponents
Imports DevComponents.DotNetBar
Imports DevComponents.DotNetBar.Controls

Public Class Departamento
    Private Accion As TipoTransaccion.Accion

    Public Sub New()
        ' Llamada necesaria para el diseñador.
        InitializeComponent()
        ' Agregue cualquier inicialización después de la llamada a
        InitializeComponent().
        Me.Nuevo_Click(Me.btNuevo, Nothing)
    End Sub

    Public Sub New(ByVal iID As Integer)
        InitializeComponent()
        Me.iiID.Value = iID
    End Sub

```

```

End Sub

Private Function Buscar(ByVal iID As Integer, Optional ByVal Texto As String = "") As
Boolean
    Dim Result As Boolean = False
    Dim Entidad As New SAGMA.Negocio.Departamento()

    Try
        With Me
            .Nuevo_Click(.btNuevo, Nothing)

            If iID > 0 Then Result = Entidad.Buscar(iID) Else Result =
Entidad.Buscar(Texto, Globales.EstBusqGeneral, Nothing)

            If Result Then
                .Accion = TipoTransaccion.Accion.Actualizar
                .lbAccion.Text = "ACTUALIZAR REGISTRO"

                .iiID.Tag = Entidad.ID
                .iiID.Value = Entidad.ID
                .txNombre.Text = Entidad.Nombre
                .txDescripcion.Text = Entidad.Descripcion
                .cbEstado.SelectedItem = Entidad.Estado
                .txNombre.SelectionStart = .txNombre.Text.Length
                .btEliminar.Enabled = True
                .txNombre.Focus()
            Else
                If iID > 0 Then MessageBoxBrisa.Exclamacion("No se
encontro el registro con codigo " & iID & ".")
                .txNombre.Focus()
            End If
        End With
        Catch exApp As Exception
            Dim Bitacora As New SysBrisa.Errores.Bitacora(exApp,
"SAGMA\Empresa\Departamento", "Buscar", True)
            Bitacora.RegistrarEvento()
        Finally
            'Codigo final
        End Try

        Return Result
    End Function

Private Function Validar() As Boolean
    Dim Result As Boolean = True

    Try
        With Me
            If .txNombre.Text.Trim.Length = 0 Then
                MessageBoxBrisa.Validacion("El campo Nombre no es
opcional.")
                .txNombre.Focus()
                Return False
            End If

            If .cbEstado.GetItemText(.cbEstado.SelectedItem).Length = 0 Then
                MessageBoxBrisa.Validacion("El campo Estado no es
opcional.")
                .cbEstado.DroppedDown = True
                .cbEstado.Focus()
                Return False
            End If
        End With
        Catch exApp As Exception
            Dim Bitacora As New SysBrisa.Errores.Bitacora(exApp,
"SAGMA\Empresa\Departamento", "Validar", True)
            Bitacora.RegistrarEvento()
        Finally
            'Codigo final
        End Try

        Return Result
    End Function

```

```

Private Sub InicializarControles(Item As Object)
    Try
        If TypeOf Item Is TextBox Then
            DirectCast(Item, TextBox).Clear()
        ElseIf TypeOf Item Is Editors.IntegerInput Then
            DirectCast(Item, Editors.IntegerInput).Value = 0
            DirectCast(Item, Editors.IntegerInput).Tag = Nothing
        ElseIf TypeOf Item Is Editors.DoubleInput Then
            DirectCast(Item, Editors.DoubleInput).Value = 0
        ElseIf TypeOf Item Is Editors.DateTimeAdv.DateTimeInput Then
            DirectCast(Item, Editors.DateTimeAdv.DateTimeInput).Value = New
Date(Date.Now.Year, Date.Now.Month, Date.Now.Day, 0, 0, 0)
        ElseIf TypeOf Item Is ComboBox Then
            DirectCast(Item, ComboBox).SelectedItem = Nothing
        ElseIf TypeOf Item Is CheckBox Then
            DirectCast(Item, CheckBox).Checked = False
        ElseIf TypeOf Item Is RadioButton Then
            DirectCast(Item, RadioButton).Checked = False
        ElseIf TypeOf Item Is CheckBoxX Then
            DirectCast(Item, CheckBoxX).Checked = False
        ElseIf TypeOf Item Is DataGridView Then
            DirectCast(Item, DataGridView).Rows.Clear()
        ElseIf TypeOf Item Is ItemPanel Then
            DirectCast(Item, ItemPanel).Items.Clear()
            DirectCast(Item, ItemPanel).Refresh()
        End If

        If DirectCast(Item, Control).Controls.Count = 0 Then Return

        For Each SubItem As Control In DirectCast(Item, Control).Controls
            Me.InicializarControles(SubItem)
        Next
    Catch exApp As Exception
        Dim Bitacora As New SysBrisa.Errores.Bitacora(exApp,
"SAGMA\Empresa\Departamento", "InicializarControles", True)
        Bitacora.RegistrarEvento()
    Finally
        'Codigo final
    End Try
End Sub

Private Sub Form_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
    Try
        With Me
            SAGMA.Negocio.Catalogo.ComboBox(.cbEstado, "ESTGENERAL")
        End With
    Catch exApp As Exception
        Dim Bitacora As New SysBrisa.Errores.Bitacora(exApp,
"SAGMA\Empresa\Departamento", "Form_Load", True)
        Bitacora.RegistrarEvento()
    Finally
        'Codigo final
    End Try
End Sub

Private Sub Form_Shown(ByVal sender As Object, ByVal e As System.EventArgs) Handles
Me.Shown
    Me.txNombre.Focus()
End Sub

Private Sub Form_KeyDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles Me.KeyDown
    Try
        Select Case e.KeyCode
            Case Keys.F2 : Me.Nuevo_Click(Me.btNuevo, Nothing)
            Case Keys.F3 : Me.Buscar(0, Me.txNombre.Text)
            Case Keys.F5 : Me.Guardar_Click(Me.btGuardar, Nothing)
            Case Keys.F6 : Me.Eliminar_Click(Me.btEliminar, Nothing)
            Case Keys.Escape : Me.Salir_Click(Me.btSalir, Nothing)
        End Select
    Catch exApp As Exception

```

```

        Dim Bitacora As New SysBrisa.Errores.Bitacora(exApp,
"SAGMA\Empresa\Departamento", "Form_KeyDown", True)
        Bitacora.RegistrarEvento()
    Finally
        'Codigo final
    End Try
End Sub

Private Sub Form_FormClosed(ByVal sender As Object, ByVal e As
System.Windows.Forms.FormClosedEventArgs) Handles Me.FormClosed
    Me.Dispose()
End Sub

Private Sub TextBox_KeyPress(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles txDescripcion.KeyPress
    Try
        With Me
            If e.KeyChar = Strings.Chr(13) Then e.Handled = True
        End With
    Catch exApp As Exception
        Dim Bitacora As New SysBrisa.Errores.Bitacora(exApp,
"SAGMA\Empresa\Departamento", "TextBox_KeyPress", True)
        Bitacora.RegistrarEvento()
    Finally
        'Codigo final
    End Try
End Sub

Private Sub TabFocus_KeyDown(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles _
    iiID.KeyDown, txNombre.KeyDown, txDescripcion.KeyDown, cbEstado.KeyDown

    Try
        If e.KeyCode = Keys.Enter Then Me.SelectNextControl(DirectCast(sender,
Control), True, True, True)
    Catch exApp As Exception
        Dim Bitacora As New SysBrisa.Errores.Bitacora(exApp,
"SAGMA\Empresa\Departamento", "TabFocus_KeyDown", True)
        Bitacora.RegistrarEvento()
    Finally
        'Codigo final
    End Try
End Sub

Private Sub TextBox_SelectTexto(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles _
    txNombre.Enter, txDescripcion.Enter

    Try
        DirectCast(sender, TextBox).SelectAll()
    Catch exApp As Exception
        Dim Bitacora As New SysBrisa.Errores.Bitacora(exApp,
"SAGMA\Empresa\Departamento", "TextBox_SelectTexto", True)
        Bitacora.RegistrarEvento()
    Finally
        'Codigo final
    End Try
End Sub

Private Sub Buscar_KeyPress(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles iiID.KeyPress
    Try
        With Me
            If e.KeyChar = Strings.Chr(13) Then
                If .iiID.Value > 0 Then .Buscar(.iiID.Value)
            End If
        End With
    Catch exApp As Exception
        Dim Bitacora As New SysBrisa.Errores.Bitacora(exApp,
"SAGMA\Empresa\Departamento", "Buscar_KeyPress", True)
        Bitacora.RegistrarEvento()
    Finally
        'Codigo final

```



```

        End Try
    End Sub

    Private Sub Nuevo_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btNuevo.Click
        Try
            With Me
                .Accion = TipoTransaccion.Accion.Guardar
                .btEliminar.Enabled = False
                .lbAccion.Text = "NUEVO REGISTRO"
                .IniciarControles(Me)
                .cbEstado.SelectedValue = "ACT"
                .txNombre.Focus()
            End With
            Catch exApp As Exception
                Dim Bitacora As New SysBrisa.Errores.Bitacora(exApp,
"SAGMA\Empresa\Departamento", "Nuevo_Click", True)
                Bitacora.RegistrarEvento()
            Finally
                'Codigo final
            End Try
        End Sub

    Private Sub Buscar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btBuscar.Click
        Try
            With Me
                .Buscar(0, .txNombre.Text)
            End With
            Catch exApp As Exception
                Dim Bitacora As New SysBrisa.Errores.Bitacora(exApp,
"SAGMA\Empresa\Departamento", "Buscar_Click", True)
                Bitacora.RegistrarEvento()
            Finally
                'Codigo final
            End Try
        End Sub

    Private Sub Guardar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btGuardar.Click
        Dim Entidad As New SAGMA.Negocio.Departamento()

        Try
            With Me
                If Not .Validar() Then Return

                Entidad.ID = Convert.ToInt32(.iiID.Tag)
                Entidad.Nombre = .txNombre.Text.Trim()
                Entidad.Descripcion = .txDescripcion.Text.Trim()
                Entidad.Estado = .cbEstado.SelectedValue
                Entidad.Usuario = Globales.Usuario.UsuarioApp

                If SAGMA.Negocio.Departamento.Registrar(Entidad, Accion) Then
.Nuevo_Click(.btNuevo, Nothing) Else .txNombre.Focus()
                End With
            Catch exApp As Exception
                Dim Bitacora As New SysBrisa.Errores.Bitacora(exApp,
"SAGMA\Empresa\Departamento", "Guardar_Click", True)
                Bitacora.RegistrarEvento()
            Finally
                'Codigo final
            End Try
        End Sub

    Private Sub Eliminar_Click(sender As System.Object, e As System.EventArgs) Handles
btEliminar.Click
        Dim DialogResult As DevComponents.DotNetBar.eTaskDialogResult

        Try
            With Me
                DialogResult = MessageBoxBrisa.Pregunta("Desea eliminar
permanentemente el registro de la base de datos...?")
            End With
        End Try
    End Sub

```

```

        If Object.Equals(DialogResult,
DevComponents.DotNetBar.eTaskDialogResult.No) Then Return

        If Object.Equals(.iiID.Tag, Nothing) Then
            MessageBoxBrisa.Validacion("Especifique un Código o ID
válido para ejecutar la transacción.")
            .iiID.Focus()
            Return
        Else
            If
SAGMA.Negocio.Departamento.Eliminar(DirectCast(.iiID.Tag, Integer), Globales.Usuario.UsuarioApp)
Then .Nuevo_Click(.btNuevo, Nothing) Else .txNombre.Focus()
            End If
        End With
        Catch exApp As Exception
            Dim Bitacora As New SysBrisa.Errores.Bitacora(exApp,
"SAGMA\Empresa\Departamento", "Eliminar_Click", True)
            Bitacora.RegistrarEvento()
        Finally
            'Codigo final
        End Try
    End Sub

    Private Sub Salir_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btSalir.Click
        Me.Close()
    End Sub
End Class

```

#### 4.2.5. Implantación

Se realizó la implantación del sistema en el servidor dedicado SVRMNTBDAPP01 para la publicación de la Base de Datos y la publicación de la Aplicación.

Una vez publicado, se realizan pruebas de usuarios durante 5 días, el resultado de dichas pruebas se describe en el siguiente capítulo.

## **CAPÍTULO V**

### **RESULTADOS Y DISCUSIÓN**

## 5.1. PRUEBAS Y RESULTADOS

### 5.1.1. Pruebas de Ejecución

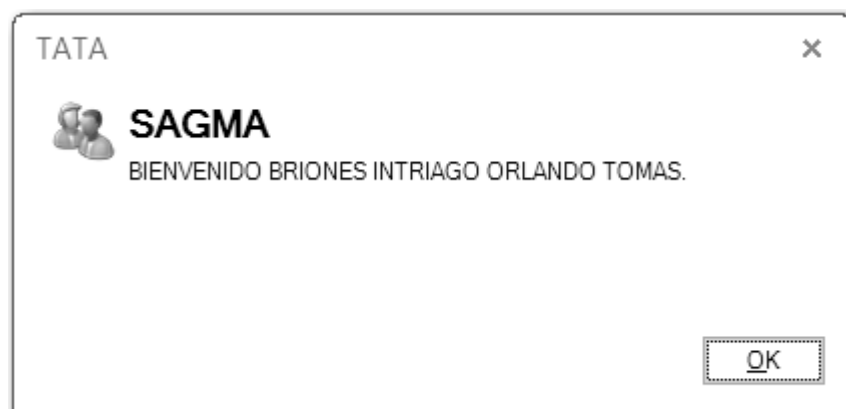
Para garantizar el funcionamiento esperado del sistema **SAGMA**, se migró la información existente en el sistema UNICENTER correspondiente a los meses de Agosto, Septiembre y Octubre del año 2013. Se trabaja con 30254 registros de ODT.

Las pruebas de calidad y rendimiento del sistema se realizaron en el siguiente orden en el módulo de gestión y reportes:

- **Acceso al Sistema**

Se realizan pruebas utilizando el usuario OBRIONES. Se verifica el acceso rápido al sistema (1 segundo aproximadamente desde el inicio de sesión hasta la habilitación de opciones generales). Una vez que el usuario se inicia sesión, se valida el registro de auditoria en la base de datos.





	Fecha	Hora	UsuarioSql	UsuarioApp	PC	Aplicacion	Accion	Descripcion
2	2013-12-05	23:41:45.7600000	sa	OBRIONES	TOMYTOSHI	.Net SqlClient Data Provider	INICIO DE SESION	INICIO DE SESION
3	2013-12-05	23:16:33.3370000	sa	OBRIONES	TOMYTOSHI	.Net SqlClient Data Provider	CIERRE DE SESION	CIERRE DE SESION
4	2013-12-05	23:14:46.5570000	sa	OBRIONES	TOMYTOSHI	.Net SqlClient Data Provider	INICIO DE SESION	INICIO DE SESION
5	2013-12-05	23:14:01.1700000	sa	OBRIONES	TOMYTOSHI	.Net SqlClient Data Provider	CIERRE DE SESION	CIERRE DE SESION
6	2013-12-05	23:12:09.7170000	sa	OBRIONES	TOMYTOSHI	.Net SqlClient Data Provider	INICIO DE SESION	INICIO DE SESION
7	2013-12-05	21:22:15.7330000	sa	OBRIONES	TOMYTOSHI	.Net SqlClient Data Provider	CIERRE DE SESION	CIERRE DE SESION
8	2013-12-05	21:16:47.5570000	sa	OBRIONES	TOMYTOSHI	.Net SqlClient Data Provider	INICIO DE SESION	INICIO DE SESION
9	2013-12-05	19:50:51.5900000	sa	OBRIONES	TOMYTOSHI	.Net SqlClient Data Provider	CIERRE DE SESION	CIERRE DE SESION
10	2013-12-05	19:49:46.2230000	sa	OBRIONES	TOMYTOSHI	.Net SqlClient Data Provider	INICIO DE SESION	INICIO DE SESION

- **Registro de ODT**

Se realiza el registro de la ODT N° 30255, dicho registro le toma al usuario 1 minutos y 30 segundos. Se valida registro de auditoria en la base de datos.

Sistema de Administración y Gestión de Mantenimiento de Aplicaciones (SAGMA)

**Menú Principal**

- Empresa
- General
- Cliente
- Ecosistema Informático
- Gestión (ODT, Actividades)
- Clasificación**
  - Categoría de ODT
  - Efectos
  - Tipo de Error
  - Tipo de Solución
- ODT** (Orden de Trabajo)
  - Tras la Pista
- Registro de Actividades**
  - Tipo de Actividad
  - Actividades
- Reportes & Análisis

**ACTUALIZAR REGISTRO**

**Registro** **Análisis para Asignación**

**Datos Generales**

Código: 30255

Prioridad: P1 - PRIORIDAD ALTA

Tipo: SOLICITUD

SLA: 2014-01-06

Descripción: cliente: rivera pichocagon mso: 201307150722130000 0859293131960001 CECASMMXXXX 130716 SALOME RIVERA PICHUGAGON PROCESO OK-Cash:PENDIENTE USD 50.00 PROCESADO VENTANILLA

**Clasificación**

Categoría: NO DEFINIDO

Tipo de Error: ERROR DE APLICACION

Efecto: SUSPENSION DEFINITIVA

**Solución**

Solucionado: ☐

Tipo de Solución: Fecha Solución: 2014-01-06

Resolutor:

Comentario:

**Cliente / Usuarios / Aplicación**

Cliente: BANCO GENERAL RUMINAHUI (222222222222)

Área Solic.: BANCA PREFERENCIAL

Autorizador: ANTAMBA CAJAMARCA LUIS ANTONIO

Beneficiario: FERNANDEZ ESCOBAR MONICA JACQUELINE

Aplicación: SIG (APLICACION WEB)

F2 F3 F5 F6 Esc

SELECT \* FROM dbo.ODT O WHERE O.ID = 30255

Solucionado	FechaSolucion	Comentario	Estado	TabPrioridad	TabTipo	TabEstado	UsrIns	PCIns	FechaIns	UsrUpd	PCUpd	FechaUpd
0	NULL		ABI	ODTPRIORIDAD	TIPOODT	ESTODT	OBRIONES	TOMYTOSHI	2013-07-18 00:00:00	NULL	NULL	NULL

Idados Mensajes

Se realiza una actualización al registro N° 30255, se obtiene un tiempo de respuesta de actualización inferior a un segundo. Se valida registro de auditoria en la base de datos.

Sistema de Administración y Gestión de Mantenimiento de Aplicaciones (SAGMA)

**Menú Principal**

- Empresa
- General
- Cliente
- Ecosistema Informático
- Gestión (ODT, Actividades)
- Clasificación**
  - Categoría de ODT
  - Efectos
  - Tipo de Error
  - Tipo de Solución
- ODT** (Orden de Trabajo)
  - Tras la Pista
- Registro de Actividades**
  - Tipo de Actividad
  - Actividades
- Reportes & Análisis

**ACTUALIZAR REGISTRO**

**Registro** **Análisis para Asignación**

**Datos Generales**

Código: 30255

Prioridad: P1 - PRIORIDAD ALTA

Tipo: SOLICITUD

SLA: 2014-01-06

Descripción: cliente: rivera pichocagon mso: 201307150722130000 0859293131960001 CECASMMXXXX 130716 SALOME RIVERA PICHUGAGON PROCESO OK-Cash:PENDIENTE USD 50.00 PROCESADO VENTANILLA

**Clasificación**

Categoría: NO DEFINIDO

Tipo de Error: ERROR DE APLICACION

Efecto: SUSPENSION DEFINITIVA

**Solución**

Solucionado: ☐

Tipo de Solución: Fecha Solución: 2014-01-06

Resolutor:

Comentario:

**Cliente / Usuarios / Aplicación**

Cliente: BANCO GENERAL RUMINAHUI (222222222222)

Área Solic.: BANCA PREFERENCIAL

Autorizador: ANTAMBA CAJAMARCA LUIS ANTONIO

Beneficiario: FERNANDEZ ESCOBAR MONICA JACQUELINE

Aplicación: SIG (APLICACION WEB)

**Información General**

1 registro(s) actualizado(s).

OK

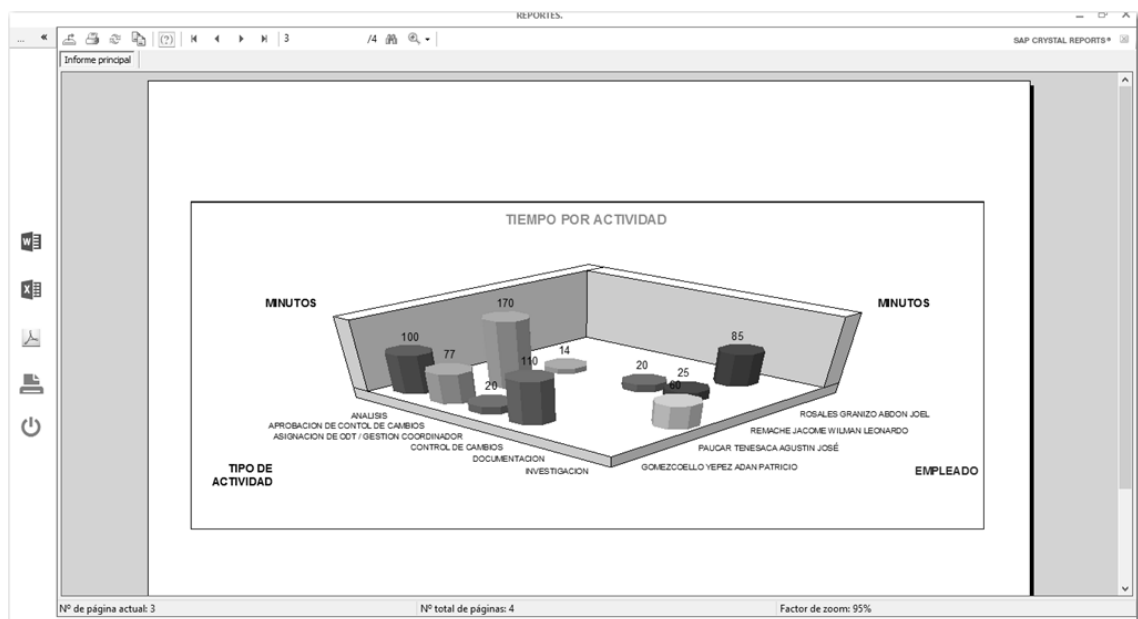
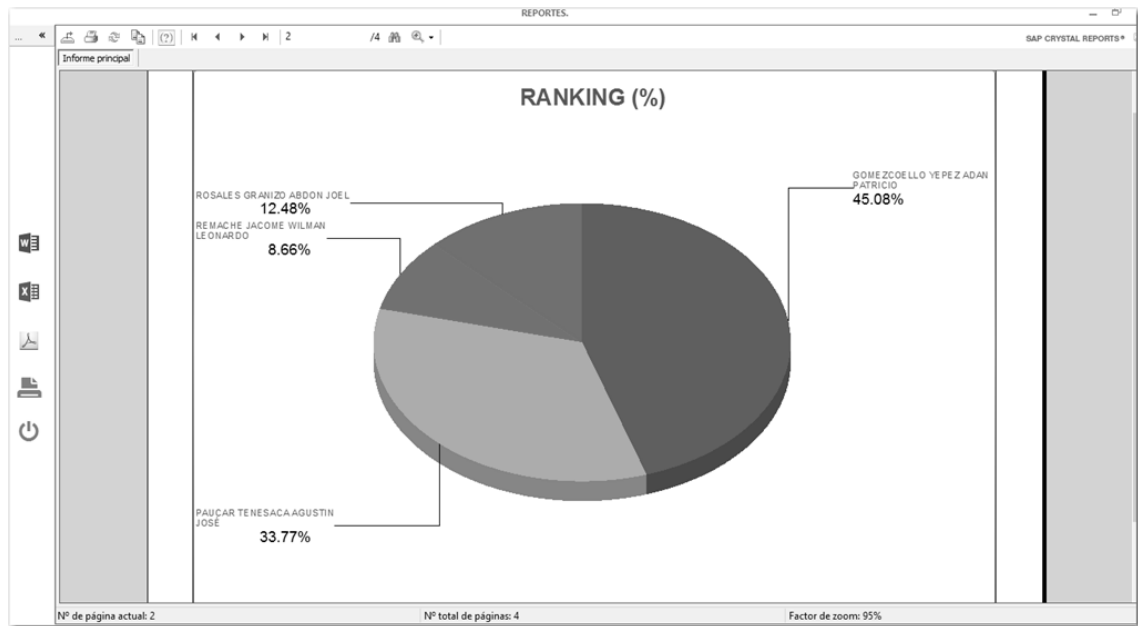
F2 F3 F5 F6 Esc

SELECT \* FROM dbo.ODT O WHERE O.ID = 30255

ados

Mensajes

shaSolucion	Comentario	Estado	TabPrioridad	TabTipo	TabEstado	UsrIn	PCIn	Fechaln	UsrUpd	PCUpd	FechaUpd
JLL		ABI	ODTPRIORIDAD	TIPOODT	ESTODT	OBRIONES	TOMYTOSHI	2013-07-18 00:00:00	OBRIONES	TOMYTOSHI	2013-07-18 00:00:00



Utilización de la herramienta “Tras la Pista”, la herramienta presenta un historial de ODT’s atendidas, permitiendo un análisis sustentado en la eficiencia de trabajos realizados.





REPORTES.

SAP CRYSTAL REPORTS

Informe principal

2013-12-04

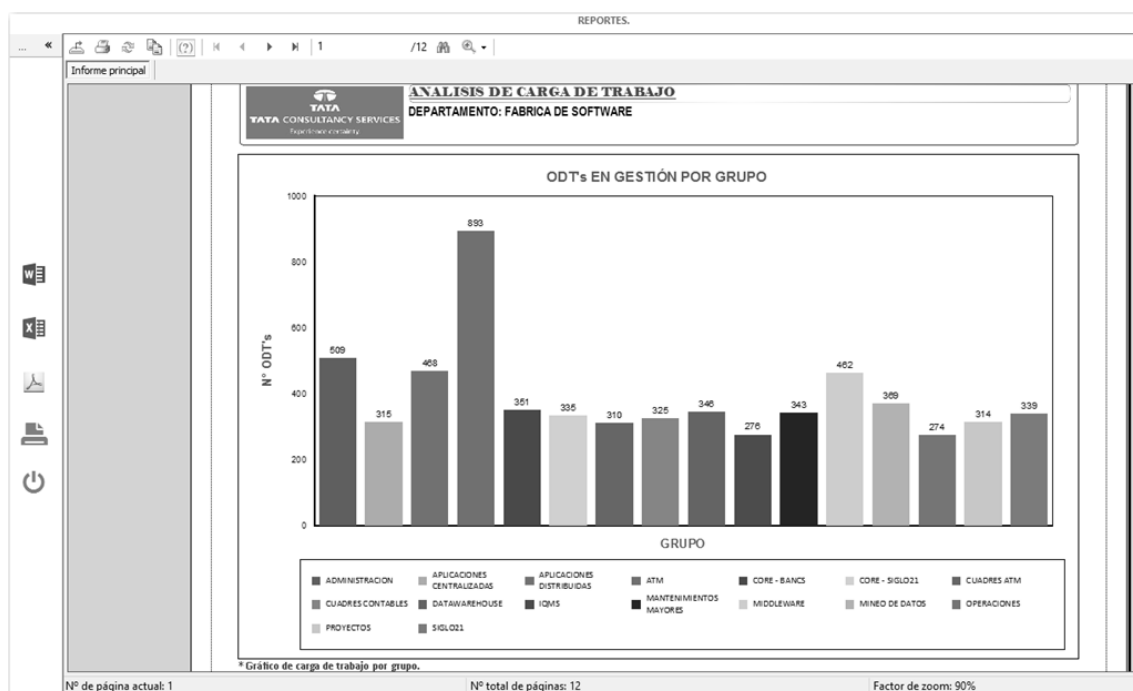
**REGISTRO DE ACTIVIDADES**  
Desde 2013-12-04 Hasta 2013-12-04  
BRIONES INTRIAGO ORLANDO TOMAS(SIGLO21 - DESARROLLADOR)

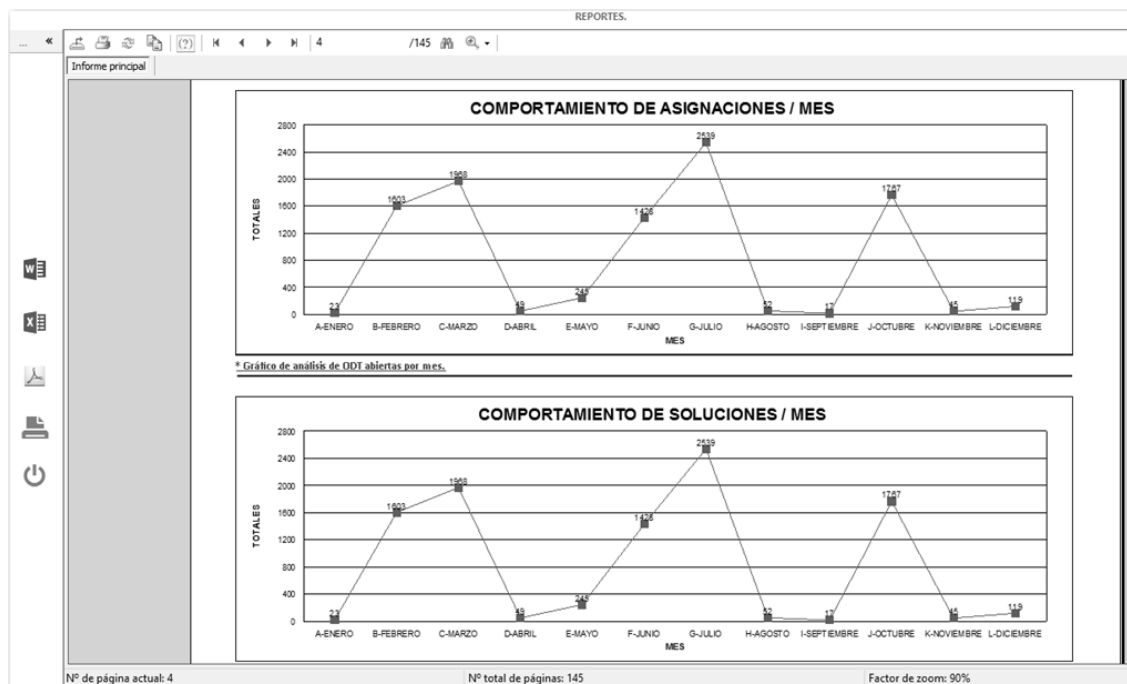
ODT	TIPO	ACTIVIDAD	INICIO	FIN	ESFUERZO (MINUT)	REGISTRO
2013-12-04						
000000001	GENERICA	0- REGISTRO DE HORAS DE INGRESO	08:00	08:02	2	2013-11-29 00:00
000007559	ERROR	CONTROL DE CAMBIOS	09:00	13:00	240	2013-11-29 00:00
000000053	ERROR	CONTROL DE CAMBIOS	14:00	14:30	30	2013-11-29 00:00
000003138	SOLICITUD	SOPORTE A LIBERARIAN	14:30	15:10	40	2013-11-29 00:00
000011139	SOLICITUD	ANALISIS	15:10	17:20	130	2013-11-29 00:00
000026788	ERROR	PRUEBAS UNITARIAS	17:20	18:05	45	2013-11-29 00:00
000000001	GENERICA	1- REGISTRO DE HORAS DE SALIDA	20:00	20:02	2	2013-11-29 00:00

(7) Actividades, (8.15) Horas de esfuerzo, (489) Minutos de esfuerzo.

Nº de página actual: 1      Nº total de páginas: 1      Factor de zoom: 100%

Ejecución de reporte (Reporte Estadístico) de análisis de carga de trabajo.  
Tiempo de ejecución inferior a 1 segundo.





### 5.1.2. Hardware Utilizado para las Pruebas

Para la ejecución de las pruebas se utilizó una estación de trabajo con las siguientes características:

- CPU: Intel Core 2 Duo 2.1 GHz.
- RAM: 3 Gb.
- HDD: 250 Gb.

El servidor de datos para las pruebas posee las siguientes características:

- CPU: Intel® Quad Core Xeon® E3-1220 3.10GHz.
- RAM: 2 Gb.
- HDD: 500 Gb.

## 5.2. RESULTADO DE LA COMPROBACIÓN DE HIPÓTESIS

### 5.2.1. Variable Independiente

#### 5.2.1.1. Software

El sistema **SAGMA**, mediante su interfaz amigable y sencilla otorga confiabilidad y usabilidad ágil al usuario, permitiéndole familiarizarse rápidamente con su uso.

Las operaciones de ingreso, actualización y eliminación de registros se realizan en una media de 1 segundo, reduciendo el tiempo invertido en cada actividad al personal del Área de Mantenimiento de Aplicaciones.

El sistema **SAGMA**, mediante el análisis automático para la asignación de ODT al personal técnico, optimiza el trabajo de los coordinadores, aumenta el rendimiento general del área, reduce el tiempo de resolución de ODT's. Todos estos los factores mencionados influyen en la productividad general.

### 5.2.2. Variable Dependiente

#### 5.2.2.1. Eficiencia

El sistema **SAGMA**, con pruebas intermitentes, durante un mes en el Área de Mantenimiento de Aplicaciones, demostró ser una herramienta eficiente, el personal técnico y administrativo aumentó su productividad. Para el mes de Noviembre (Mes de Pruebas), se contabilizó un mayor número de ODT's resueltas.

#### 5.2.2.2. Eficacia

El uso del sistema **SAGMA**, disminuyó los tiempos de entrega de soluciones de ODT's en el Área de Mantenimiento de Aplicaciones, demostrándose así su

eficacia. Los resultados en tiempos de respuesta y número de ODT's atendidas son altamente aceptables.

#### **5.2.2.3. Económico**

La aplicación del sistema **SAGMA**, como herramienta de apoyo en las actividades cotidianas, de todo el personal del Área de Mantenimiento de Aplicaciones, influyó de forma positiva bajos los requerimientos de calidad esperados.

Se disminuyó el incumplimiento de entrega de soluciones en un 40%, uno de los factores de mayor impacto en el incumplimiento era el desconocimiento de las aplicaciones, mediante el uso de la base de conocimientos del sistema SAGMA se logró reducir el impacto.

Con la reducción de tiempos de atención de ODT's con aplicación de SLA, se disminuyó el número de horas extras de toda el Área en general.

### **5.3. ANÁLISIS DE RESULTADOS**

Mediante la aplicación del sistema **SAGMA**, el personal técnico cuenta con una herramienta efectiva para el desarrollo de sus actividades diarias. La toma de decisiones por parte del personal administrativo denotó una mejora considerable en las actividades como la asignación de ODT's, seguimientos de actividades, medición del rendimiento del área, control sobre horas trabajadas y horas excedentes.

**Tabla 18:** Cuadro comparativo de uso de **SAGMA**

	DETALLE	Tiempo de Solución de ODT utilizando <b>SAGMA</b>	Tiempo de Solución de ODT sin el uso de <b>SAGMA</b>
		Tiempo en Minutos	Tiempo en Minutos
1	ODT Prioridad P0	45	180
2	ODT Prioridad P1	65	500
3	ODT Prioridad P2	375	900
4	ODT Prioridad P3	400	1100
5	ODT Prioridad P4	480	1750
6	ODT Prioridad P5	520	1800
<b>TOTAL</b>		<b>1885</b>	<b>6230</b>
<b>MEDIA</b>		<b>314.17</b>	<b>1,038.33</b>
<b>DESVIACIÓN ESTÁNDAR</b>		<b>189.50</b>	<b>596.67</b>

La tabla anterior muestra los tiempos de solución medido en minutos de las ODT's por prioridad. Como se puede observar la media del grupo observado que se obtuvo usando **SAGMA** es 314.17 y la media del grupo que no usó **SAGMA** es 1,038.33; la diferencia entre los dos grupos es evidente, se busca demostrar si la diferencia es significativa o si ésta ocurre por mera casualidad.

**H0** = Un Sistema Informático basado en Base de Conocimientos Especializada que cuente con los componentes del Ecosistema Informático no permitirá la correcta asignación de recursos para resolver la ODT.

**De igual forma se plantea una hipótesis alterna:**

**Ha** = Un Sistema Informático basado en Base de Conocimientos Especializada que cuente con los componentes del Ecosistema Informático permitirá la correcta asignación de recursos para resolver la ODT.

Se calcula el error estándar de la diferencia entre dos medias, lo cual también se lo conoce como margen de error de la prueba t.

$$S_{x_1-x_2} = \sqrt{\frac{\sum x_1^2 + \sum x_2^2}{n_1 + n_2 - 2} \left( \frac{1}{n_1} + \frac{1}{n_2} \right)}$$

$$S_{x1-x2} = \sqrt{\frac{189.50 + 596.67}{6 + 6 - 2} \left( \frac{1}{6} + \frac{1}{6} \right)}$$

$$S_{x1-x2} = 5.12$$

El valor de **5.12** es la diferencia esperada entre ambos grupos, se verificar la siguiente pregunta:

¿Esta diferencia es lo suficientemente mayor que la diferencia prevista como para poder rechazar la hipótesis de nulidad?

Para esto, se obtiene la razón de ambos números, lo que se conoce como razón t.

$$T = \frac{\bar{X}_1 - \bar{X}_2}{S_{x1-x2}}$$

$$T = \frac{314.17 - 1,038.33}{5.12}$$

$$T = -141.44$$

La diferencia observada es de **141.44** veces mayor que la esperada, según una verdadera hipótesis de nulidad. **¿Es lo suficientemente grande para rechazar la hipótesis de nulidad a nivel 0.05?**; para despejar esta duda se calcula los **grados de libertad (gl)**.

$$gl = n1 + n2 - 2$$

$$gl = 6 + 6 - 2$$

$$gl = 10$$

Con 10 grados de libertad, a un nivel de 0.05 y basándose en la tabla T-STUDENT se encuentra una razón de t de **1.8125**.

La razón calculada de **141.44** es mayor que **1.8125**, lo que demuestra que la diferencia antes de usar **SAGMA** y después de usar **SAGMA**, es mayor que el valor que se necesitaba para rechazar la hipótesis de nulidad al nivel de significación de 0.05; por lo tanto, los datos son suficientemente significativos para llegar a la conclusión que la diferencia no depende de la casualidad y se acepta **H<sub>a</sub>**.

$141.44 > 1.8125$ ; se acepta la hipótesis alterna y se puede asegurar que:

***Un Sistema Informático basado en Base de Conocimientos Especializada que cuente con los componentes del Ecosistema Informático permitirá la correcta asignación de recursos para resolver la ODT.***

**Tiempos de respuesta en otros procesos con el uso del sistema SAGMA.**

**Tabla 19:** Tiempos de respuesta en diferentes procesos

	DETALLE	Tiempo de Solución de ODT utilizando SAGMA	Tiempo de Solución de ODT sin el uso de SAGMA
		Tiempo en Minutos	Tiempo en Minutos
1	Asignación de ODT	3	35
2	Generación de Reporte de Rendimiento	1	9
3	Generación de Reporte de Atención de ODT	1	9
<b>TOTAL</b>		5	53
<b>MEDIA</b>		1.44	14.15
<b>DESVIACIÓN ESTÁNDAR</b>		0.94	12.26

El sistema informático **SAGMA**, es una herramienta de trabajo capaz de elevar el rendimiento del Área de Mantenimiento de Aplicaciones de la Empresa TCS Ecuador.

## **5.4. CONCLUSIONES**

- Se mejora la asignación de Órdenes de Trabajo (ODT) en el Área de Mantenimiento de Aplicaciones de la Empresa TCS Ecuador, mediante la implementación del Sistema Informático **SAGMA**, tomando en consideración la



capacidad y experiencia de los desarrolladores. Se optimiza los procesos y actividades en general del personal técnico y administrativo.

- Se diseña una Base de Conocimientos Especializada que cuenta con información de Servidores, Base de Datos, Aplicaciones y soluciones históricos de ODT's, mediante la cual los desarrolladores optimizan sus actividades de desarrollo y gestión de ODT's basándose en el conocimiento recopilado en el sistema **SAGMA**.
- El desarrollo del sistema informático **SAGMA**, influyó positivamente en la administración eficiente y segura, de los procesos de desarrollo y mantenimiento de software en el Área de Mantenimiento de Aplicaciones de la Empresa TCS Ecuador. La toma de decisiones es apoyada por los reportes e informes analíticos y estadísticos, los mismos que se generan a partir del registro categorizado de actividades por parte de los desarrolladores.
- El Área de Mantenimiento de Aplicaciones de la Empresa TCS Ecuador, cuenta con reportes estadísticos ágiles y confiables para la toma de decisiones de asignación de ODT, basándose en el rendimiento de los desarrolladores.

## 5.5. RECOMENDACIONES

Una vez finalizada la presente investigación, se ponen en consideración las siguientes recomendaciones.

- ✓ Implementación de seguridad al sistema **SAGMA**, mediante la aplicación de perfiles de usuario, con la finalidad de establecer roles y responsabilidades en los procesos.
- ✓ Implementar funcionalidad de envío de notificaciones automáticas vía email a usuarios beneficiarios de ODT, con el objetivo de informar el avance de la solución.

- ✓ Establecer una cultura organizacional en el registro de actividades diaria del personal del Área de Mantenimiento de Aplicaciones, con el fin de contar con una base de conocimientos sólida y confiable para el desarrollo de soluciones futuras.
  
- ✓ Implementar funcionalidades adicionales al Sistema Informático **SAGMA** para extender su usabilidad a las áreas de apoyo dentro de la empresa.

## **CAPÍTULO VI**

### **LITERATURA CITADA**

## 6.1. BIBLIOGRAFÍA

- Facilitadores, (2013). Área de Mantenimiento de Aplicaciones. ODT.
- Garzón Gavilanes, Pablo Oswaldo (2009). Tesis de Grado “Datawarehouse de Mesa de Servicios para el Área de Mantenimiento de Aplicaciones”. Definición de SLA. Pág. 11.
- Andrés Gómez de Silva Garza, Ignacio de Jesús Ania Briseño. (2008). Introducción a la Computación. Cengage Learning Editores. Software. Pág. 23.
- Modesto Castrillón, Antonio C. Domínguez, Santiago Candela, Luis Doreste, David Freire, Agustín Salgado, Sunil L. Khemchandani, Daniel Hernández. (2011). Fundamentos de Informática y Programación para Ingeniería. Definición de Sistema Informático. Cap. 1. Pág. 4.
- Andrés Gómez de Silva Garza, Ignacio de Jesús Ania Briseño. (2008). Introducción a la Computación. Cengage Learning Editores. Sistemas Operativos. Pág. 175.
- Carmen de Pablos Heredero, José Joaquín López Hermoso Agius Santiago Martín – Romo Romero, Sonia Medina Salgado. (2011). Organización y transformación de los sistemas de información en la empresa. 1era Edición. ESIC EDITORIAL. Base de Conocimiento. Pág. 165.
- Carlos Coronel, Steven Morris, Peter Rob. (2011). Bases de Datos, Diseño, Implementación y Administración. 9na Edición. Cengage Learning Editores S.A. Base de Datos. Pág. 7.
- G. Quintana, M. Marqués, J. L. Aliaga, M. J. Aramburu. (2008). Aprende SQL. Publicaciones de la Universidad Jaume. SQL. Pág. 8.
- Jérôme Gabillaud. (2010). Recursos Informáticos SQL Server 2008 – Administración. Ediciones ENI. Sistema de Gestión de Bases de Datos Relacionales. Pág. 17.
- Vicente Javier Eslava Muñoz. (2012). Aprendiendo a programar paso a paso con C. Bubok Publishing S.L. Entorno de –Desarrollo Integrado (IDE). Pág. 10.
- Andrea Pastor. (2010). Cultura General. Ediciones Paraninfo S.A. Lenguaje de Programación. Pág. 412.

- Jérôme Lafosse. (2010). Expert IT Struts 2 - El framework de desarrollo de aplicaciones Java EE. Ediciones ENI. Framework. Pág. 11.
- Jordi Conesa Caralt, Jordi Ceballos Villach, David Gañán Jiménez, Àngels Rius Gavidia. (2010). Introducción a .NET. Editorial UOC. Microsoft Visual Studio.NET. Pág. 20.
- Ángel Cobo Yera. (2009). Programar desde un punto de vista científico. Editorial Visión Libros. Programación Orientada a Objetos. Pág. 129.
- David Muñoz Negrón. (2009). Administración de operaciones. Enfoque de administración de procesos de negocios. Cengage Learning Editores. Lenguaje Unificado de Modelamiento UML. Pág. 49.
- Diego Redolar Ripoll. (2010). Fundamentos de psicobiología. Editorial UOC. Metodología Cuasi-Experimental. Pág. 57.
- Manuel Palencia-Lefler. (2008). 90 Técnicas de Relaciones Públicas: Manual de Comunicación Corporativa. Profit Editorial. Encuesta. Pág. 109.
- Vidal Díaz de Rada. (2009). Análisis de datos de encuestas: Desarrollo de una investigación completa utilizando SPSS. Editorial UOC. Etapas en una Investigación con Encuestas. Pág. 19.
- Vidal Díaz de Rada. (2009). Análisis de datos de encuestas: Desarrollo de una investigación completa utilizando SPSS. Editorial UOC. Entrevista. Pág. 26
- Joaquín Tomás-Sábado. (2010). Fundamentos de bioestadística y análisis de datos para enfermería. Univ. Autónoma de Barcelona. Población y Muestra. Pág. 21.
- Amaro Calderón Sarah Damaris, Valverde Rebaza Jorge Carlos. (2007). Metodologías Ágiles. Universidad Nacional de Trujillo. Metodologías Ágiles.
- Israel García Portuondo, Lissette Fonseca Suárez. (2010). Desarrollo Rápido de Aplicaciones Apoyado en MDA. Facultad Regional "Mártires de Artemisa" de la Universidad de las Ciencias Informáticas (UCI). Etapas de la Metodología RAD. Pág. 2.

## **6.2. ENLACES WEB**

Patricia López, Francisco Ruíz. (2011). Ingeniería del Software I. Universidad Cantabria. Lenguaje Unificado de Modelado – UML. Tomado de <http://ocw.unican.es/enseanzas-tecnicas/ingenieria-del-software-i/materiales-de-clase-1/is1-t02-trans.pdf>.

## **ANEXOS**

# PROBLEMATIZACIÓN

## DIAGNÓSTICO

- ✓ Carece de estadística de rendimiento de los desarrolladores.
- ✓ Carece de control de cumplimiento de tiempo asignado.
- ✓ No se cuenta con una Base de Conocimientos Especializada.
- ✓ No existe control de tiempo out de desarrolladores.
- ✓ No existe asignación de complejidad de las ODT.
- ✓ Asignación de ODT a personal no especializado.
- ✓ Consumo de tiempo innecesario de desarrollo.
- ✓ Incorrecta estimación de tiempos de entrega.
- ✓ Desconocimiento de aplicaciones, bases de datos y servidores por parte de los desarrolladores.
- ✓ Desconocimiento de tiempo de disponibilidad de los desarrolladores.
- ✓ Incumplimiento de tiempos de entrega.
- ✓ Versionamientos con errores.
- ✓ Incumplimientos del SLA (Service Level Agreement o Acuerdo de Nivel de Servicio).
- ✓ Pérdidas Económicas.



## ÁRBOL DE PROBLEMAS



## FORMULACIÓN DEL PROBLEMA

PROBLEMA	OBJETIVOS	RESULTADOS ESPERADOS
¿Cómo mejorar la asignación de ODT a los desarrolladores de software del Área de Mantenimiento de Aplicaciones de la empresa TCS Ecuador?	Mejorar la asignación de ODT en el Área de Mantenimiento de Aplicaciones de la empresa TCS Ecuador, mediante la implementación de un Sistema Informático basado en Bases de Conocimientos.	Las ODT se asignan de acuerdo a la capacidad del desarrollador.
	ESPECÍFICOS	
¿Cuáles son los elementos de la Base de Conocimientos para la correcta asignación de ODT?	Determinar los elementos de la Base de Conocimientos necesarios para proveer información relacionada a la ODT.	Se utiliza una Base de Conocimientos Especializada con los elementos necesarios para el desarrollo de una solución.
¿Cómo lograr obtener un mayor rendimiento de los desarrolladores en las actividades asignadas?	Proveer de reportes estadísticos a las líneas de supervisión mediante el registro de actividades de los desarrolladores, para ejercer un control y seguimiento efectivo.	El registro categorizado de actividades de los desarrolladores genera reportes para control y seguimiento de líneas de supervisión sobre su rendimiento.
¿Cómo obtener una estimación de tiempo coherente para el cumplimiento y gestión eficiente de ODT?	Proporcionar datos estadísticos relacionados a la complejidad de la ODT y rendimiento de los desarrolladores para una correcta estimación de tiempos.	Se cuenta con estadística para toma de decisiones de asignación de ODT basado en la complejidad y el rendimiento de los desarrolladores.

## MATRIZ DE RELACIÓN

PROBLEMA	OBJETIVOS	HIPÓTESIS
¿Cómo mejorar la asignación de ODT a los desarrolladores de software del Área de Mantenimiento de Aplicaciones de la empresa TCS Ecuador?	Mejorar la asignación de ODT en el Área de Mantenimiento de Aplicaciones de la empresa TCS Ecuador, mediante la implementación de un Sistema Informático basado en Bases de Conocimientos.	VI: Sistema Informático basado en Bases de Conocimientos. VD: Asignación eficaz de ODT. Un Sistema Informático basado en Base de Conocimientos Especializada que cuente con los componentes del Ecosistema Informático permitirá la correcta asignación de recursos para resolver la ODT.
	<b>ESPECÍFICOS</b>	
¿Cuáles son los elementos de la Base de Conocimientos para la correcta asignación de ODT?	Determinar los elementos de la Base de Conocimientos necesarios para proveer información relacionada a la ODT.	VI: Elementos de Base de Conocimientos. VD: Gestión de ODT. Una Base de Conocimientos Especializada que cuente con los componentes del Ecosistema Informático permitirá la correcta asignación de recursos para resolver la ODT.
¿Cómo lograr obtener un mayor rendimiento de los desarrolladores en las actividades asignadas?	Proveer de reportes estadísticos a las líneas de supervisión mediante el registro de actividades de los desarrolladores, para ejercer un control y seguimiento efectivo.	VI: Registro de Actividades. VD: Reportes estadísticos a líneas de supervisión. Mediante el registro de actividades diarias se podrá generar estadística a nivel gerencial para un control y seguimiento efectivo.
¿Cómo obtener una estimación de tiempo coherente para el cumplimiento y gestión eficiente de ODT?	Proporcionar datos estadísticos relacionados a la complejidad de la ODT y rendimiento de los desarrolladores para una correcta estimación de tiempos.	VI: Datos estadísticos relacionados. VD: Estimación de tiempos. Mediante la aplicación de estadística se podrá mejorar la estimación de tiempos de resolución de las ODT y el rendimiento de los desarrolladores.

## SISTEMATIZACIÓN

CAUSA	EFEECTO	PRONÓSTICO	CONTROL DEL PRONÓSTICO
Falencias en el control de cumplimiento de tiempo asignado.	Descontento y reclamos de usuarios beneficiarios de la ODT.	Asignaciones de ODT no organizadas y uso de tiempo no administrado.	Establecer reglas de notificaciones de caducidad a los coordinadores y desarrolladores.
Desconocimiento de tiempo de disponibilidad de los desarrolladores.	Administración deficiente del tiempo de los desarrolladores.	Pérdida de tiempo y acumulación de ODT en cola de asignación y resolución.	Establecer asignaciones automáticas de ODT.
Escasa estadística de rendimiento de los desarrolladores.	Asignación de ODT a personal no especializado.	Versionamientos de soluciones a producción con errores.	Asignar ODT a desarrolladores con capacidad de resolución.
	Incorrecta estimación de tiempos de entrega.	Incumplimiento de los tiempos de entrega y SLA (Acuerdo de nivel de Servicio) ocasionando sanciones económicas a la empresa.	Contar con una base de conocimientos que brinde información de soluciones relacionadas.
	Consumo de tiempo innecesario de desarrollo.	Poca disponibilidad de desarrolladores para atención de ODT prioritarias.	Reorganizar ODT por prioridad y complejidad.
Base de Conocimientos no especializada.	Desconocimiento de aplicaciones, bases de datos y servidores por parte de los desarrolladores.	Aumenta el tiempo de desarrollo y retrasos en los tiempos de entrega.	Diseño de una Base de Conocimientos Especializada con los elementos necesarios para brindar toda la información relacionada a la ODT y rendimiento de los desarrolladores.



**UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO**  
**FACULTAD DE CIENCIAS DE LA INGENIERÍA**  
**INGENIERÍA EN SISTEMAS**



**ENCUESTAS**

**Lugar / Empresa:** TCS Ecuador.

**Área:** Mantenimiento de Aplicaciones.

**Dirigida a:** Analistas - Desarrolladores.

**Objetivo:** Conocer la opinión del personal del Área en lo referente al manejo técnico y administrativo del Área de Mantenimiento de Aplicaciones.

**Encuestador:** Orlando Briones Intriago.

**1. ¿Lleva un registro de las actividades diarias?**

*\* Marque solo una respuesta.*

De todas ( )

De algunas ( )

De ninguna ( )

**2. ¿En que lleva los registros de actividades diarias?**

*\* Marque una o más respuestas.*

Registro UNICENTER ( )

Archivo propio ( )

Archivo manual ( )

**3. ¿Considera acertada las asignaciones de ODT's por parte de los Coordinadores?**

*\* Marque solo una respuesta.*

Muy de acuerdo ( )

De acuerdo ( )

En desacuerdo ( )

Muy desacuerdo ( )

**4. ¿Conoce los sistemas, bases de datos, servidores a los cuales el Área de Mantenimiento de Aplicaciones da soporte?**

*\* Marque solo una respuesta.*

En su totalidad ( )

Parcialmente ( )

Solo los que necesito ( )

**5. ¿Cree que es necesario contar con un registro histórico de soluciones?**

*\* Marque solo una respuesta.*

Altamente necesario ( )

Necesario ( )

No es necesario ( )

**6. ¿Cree usted que la implementación de un sistema informático especializado sería útil para mejorar los procesos del Área de Mantenimiento de Aplicaciones?**

*\* Marque solo una respuesta.*

Muy de acuerdo ( )

De acuerdo ( )

En desacuerdo ( )

Muy desacuerdo ( )



**UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO**  
**FACULTAD DE CIENCIAS DE LA INGENIERÍA**  
**INGENIERÍA EN SISTEMAS**



**ENCUESTAS**

**Lugar / Empresa:** TCS Ecuador.

**Área:** Mantenimiento de Aplicaciones.

**Dirigida a:** Coordinadores.

**Objetivo:** Conocer la opinión del personal del Área en lo referente al manejo técnico y administrativo del Área de Mantenimiento de Aplicaciones.

**Encuestador:** Orlando Briones Intriago.

**1. ¿Conoce la capacidad de atención y carga de trabajo de sus subordinados?**

*\* Marque solo una respuesta.*

Si ( )

No ( )

No estoy seguro ( )

**2. ¿Lleva un registro sobre el incumplimiento en el tiempo de entrega de las soluciones por parte de sus subordinados?**

*\* Marque solo una respuesta.*

Si ( )

No ( )

No estoy seguro ( )

**3. ¿Conoce las actividades en las que sus subordinados distribuyen el tiempo?**

*\* Marque solo una respuesta.*

Si ( )

No ( )

No estoy seguro ( )

**4. ¿Cree usted que la implementación de un sistema informático especializado sería útil para mejorar los procesos del Área de Mantenimiento de Aplicaciones?**

***\* Marque solo una respuesta.***

Muy de acuerdo ( )

De acuerdo ( )

En desacuerdo ( )

Muy desacuerdo ( )





**UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO**  
**FACULTAD DE CIENCIAS DE LA INGENIERÍA**  
**INGENIERÍA EN SISTEMAS**



**ENCUESTAS**

**Lugar / Empresa:** TCS Ecuador.

**Área:** Mantenimiento de Aplicaciones.

**Dirigida a:** Facilitadores / Gerente de Área.

**Objetivo:** Conocer la opinión del personal del Área en lo referente al manejo técnico y administrativo del Área de Mantenimiento de Aplicaciones.

**Encuestador:** Orlando Briones Intriago.

**1. ¿Cree usted que el Área de Mantenimiento de Aplicaciones tiene la capacidad suficiente de atención?**

*\* Marque solo una respuesta.*

Si ( )

No ( )

No estoy seguro ( )

**2. ¿Conoce el tiempo extra de trabajo del área en general?**

*\* Marque solo una respuesta.*

Si ( )

No ( )

No estoy seguro ( )

**3. ¿Considera necesario que el Área de Mantenimiento de Aplicaciones cuente con un registro de usuarios por cliente, en el que conste información relevante, en caso de requerir comunicación directa?**

*\* Marque solo una respuesta.*

Altamente necesario ( )

Necesario ( )

No es necesario ( )

**4. ¿Cree necesario que el área de Mantenimiento de Aplicaciones cuente con información estadística para determinar el rendimiento general del área?**

***\* Marque solo una respuesta.***

Altamente necesario ( )

Necesario ( )

No es necesario ( )

**5. ¿Cree usted que la implementación de un sistema informático especializado sería útil para mejorar los procesos del Área de Mantenimiento de Aplicaciones?**

***\* Marque solo una respuesta.***

Muy de acuerdo ( )

De acuerdo ( )

En desacuerdo ( )

Muy desacuerdo ( )

# **TABLA T-STUDENT**

Tabla t-Student



Grados de libertad	0.25	0.1	0.05	0.025	0.01	0.005
1	1.0000	3.0777	6.3137	12.7062	31.8210	63.6559
2	0.8165	1.8856	2.9200	4.3027	6.9645	9.9250
3	0.7649	1.6377	2.3534	3.1824	4.5407	5.8408
4	0.7407	1.5332	2.1318	2.7765	3.7469	4.6041
5	0.7267	1.4759	2.0150	2.5706	3.3649	4.0321
6	0.7176	1.4398	1.9432	2.4469	3.1427	3.7074
7	0.7111	1.4149	1.8946	2.3646	2.9979	3.4995
8	0.7064	1.3968	1.8595	2.3060	2.8965	3.3554
9	0.7027	1.3830	1.8331	2.2622	2.8214	3.2498
10	0.6998	1.3722	1.8125	2.2281	2.7638	3.1693
11	0.6974	1.3634	1.7959	2.2010	2.7181	3.1058
12	0.6955	1.3562	1.7823	2.1788	2.6810	3.0545
13	0.6938	1.3502	1.7709	2.1604	2.6503	3.0123
14	0.6924	1.3450	1.7613	2.1448	2.6245	2.9768
15	0.6912	1.3406	1.7531	2.1315	2.6025	2.9467
16	0.6901	1.3368	1.7459	2.1199	2.5835	2.9208
17	0.6892	1.3334	1.7396	2.1098	2.5669	2.8982
18	0.6884	1.3304	1.7341	2.1009	2.5524	2.8784
19	0.6876	1.3277	1.7291	2.0930	2.5395	2.8609
20	0.6870	1.3253	1.7247	2.0860	2.5280	2.8453
21	0.6864	1.3232	1.7207	2.0796	2.5176	2.8314
22	0.6858	1.3212	1.7171	2.0739	2.5083	2.8188
23	0.6853	1.3195	1.7139	2.0687	2.4999	2.8073
24	0.6848	1.3178	1.7109	2.0639	2.4922	2.7970
25	0.6844	1.3163	1.7081	2.0595	2.4851	2.7874
26	0.6840	1.3150	1.7056	2.0555	2.4786	2.7787
27	0.6837	1.3137	1.7033	2.0518	2.4727	2.7707
28	0.6834	1.3125	1.7011	2.0484	2.4671	2.7633
29	0.6830	1.3114	1.6991	2.0452	2.4620	2.7564
30	0.6828	1.3104	1.6973	2.0423	2.4573	2.7500
31	0.6825	1.3095	1.6955	2.0395	2.4528	2.7440
32	0.6822	1.3086	1.6939	2.0369	2.4487	2.7385
33	0.6820	1.3077	1.6924	2.0345	2.4448	2.7333
34	0.6818	1.3070	1.6909	2.0322	2.4411	2.7284
35	0.6816	1.3062	1.6896	2.0301	2.4377	2.7238
36	0.6814	1.3055	1.6883	2.0281	2.4345	2.7195
37	0.6812	1.3049	1.6871	2.0262	2.4314	2.7154
38	0.6810	1.3042	1.6860	2.0244	2.4286	2.7116
39	0.6808	1.3036	1.6849	2.0227	2.4258	2.7079
40	0.6807	1.3031	1.6839	2.0211	2.4233	2.7045
41	0.6805	1.3025	1.6829	2.0195	2.4208	2.7012
42	0.6804	1.3020	1.6820	2.0181	2.4185	2.6981
43	0.6802	1.3016	1.6811	2.0167	2.4163	2.6951
44	0.6801	1.3011	1.6802	2.0154	2.4141	2.6923
45	0.6800	1.3007	1.6794	2.0141	2.4121	2.6896
46	0.6799	1.3002	1.6787	2.0129	2.4102	2.6870
47	0.6797	1.2998	1.6779	2.0117	2.4083	2.6846
48	0.6796	1.2994	1.6772	2.0106	2.4066	2.6822
49	0.6795	1.2991	1.6766	2.0096	2.4049	2.6800

50	0.6794	1.2987	1.6759	2.0086	2.4033	2.6778
51	0.6793	1.2984	1.6753	2.0076	2.4017	2.6757
52	0.6792	1.2980	1.6747	2.0066	2.4002	2.6737
53	0.6791	1.2977	1.6741	2.0057	2.3988	2.6718
54	0.6791	1.2974	1.6736	2.0049	2.3974	2.6700
55	0.6790	1.2971	1.6730	2.0040	2.3961	2.6682
56	0.6789	1.2969	1.6725	2.0032	2.3948	2.6665
57	0.6788	1.2966	1.6720	2.0025	2.3936	2.6649
58	0.6787	1.2963	1.6716	2.0017	2.3924	2.6633
59	0.6787	1.2961	1.6711	2.0010	2.3912	2.6618
60	0.6786	1.2958	1.6706	2.0003	2.3901	2.6603
61	0.6785	1.2956	1.6702	1.9996	2.3890	2.6589
62	0.6785	1.2954	1.6698	1.9990	2.3880	2.6575
63	0.6784	1.2951	1.6694	1.9983	2.3870	2.6561
64	0.6783	1.2949	1.6690	1.9977	2.3860	2.6549
65	0.6783	1.2947	1.6686	1.9971	2.3851	2.6536
66	0.6782	1.2945	1.6683	1.9966	2.3842	2.6524
67	0.6782	1.2943	1.6679	1.9960	2.3833	2.6512
68	0.6781	1.2941	1.6676	1.9955	2.3824	2.6501
69	0.6781	1.2939	1.6672	1.9949	2.3816	2.6490
70	0.6780	1.2938	1.6669	1.9944	2.3808	2.6479
71	0.6780	1.2936	1.6666	1.9939	2.3800	2.6469
72	0.6779	1.2934	1.6663	1.9935	2.3793	2.6458
73	0.6779	1.2933	1.6660	1.9930	2.3785	2.6449
74	0.6778	1.2931	1.6657	1.9925	2.3778	2.6439
75	0.6778	1.2929	1.6654	1.9921	2.3771	2.6430
76	0.6777	1.2928	1.6652	1.9917	2.3764	2.6421
77	0.6777	1.2926	1.6649	1.9913	2.3758	2.6412
78	0.6776	1.2925	1.6646	1.9908	2.3751	2.6403
79	0.6776	1.2924	1.6644	1.9905	2.3745	2.6395
80	0.6776	1.2922	1.6641	1.9901	2.3739	2.6387
81	0.6775	1.2921	1.6639	1.9897	2.3733	2.6379
82	0.6775	1.2920	1.6636	1.9893	2.3727	2.6371
83	0.6775	1.2918	1.6634	1.9890	2.3721	2.6364
84	0.6774	1.2917	1.6632	1.9886	2.3716	2.6356
85	0.6774	1.2916	1.6630	1.9883	2.3710	2.6349
86	0.6774	1.2915	1.6628	1.9879	2.3705	2.6342
87	0.6773	1.2914	1.6626	1.9876	2.3700	2.6335
88	0.6773	1.2912	1.6624	1.9873	2.3695	2.6329
89	0.6773	1.2911	1.6622	1.9870	2.3690	2.6322
90	0.6772	1.2910	1.6620	1.9867	2.3685	2.6316
91	0.6772	1.2909	1.6618	1.9864	2.3680	2.6309
92	0.6772	1.2908	1.6616	1.9861	2.3676	2.6303
93	0.6771	1.2907	1.6614	1.9858	2.3671	2.6297
94	0.6771	1.2906	1.6612	1.9855	2.3667	2.6291
95	0.6771	1.2905	1.6611	1.9852	2.3662	2.6286
96	0.6771	1.2904	1.6609	1.9850	2.3658	2.6280
97	0.6770	1.2903	1.6607	1.9847	2.3654	2.6275
98	0.6770	1.2903	1.6606	1.9845	2.3650	2.6269
99	0.6770	1.2902	1.6604	1.9842	2.3646	2.6264
100	0.6770	1.2901	1.6602	1.9840	2.3642	2.6259
$\infty$	0.6745	1.2816	1.6449	1.9600	2.3263	2.5758