

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

DESARROLLO DE UN SISTEMA DE SOFTWARE PARA LA
GESTIÓN DE COBRANZAS, PARA LA EMPRESA TELCOLSAM
S.A. UTILIZANDO LA TECNOLOGÍA WINDOWS PRESENTATION
FOUNDATION (WPF).

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
ELECTRÓNICA Y REDES DE INFORMACIÓN

ANA MARÍA ZAMBRANO VIZUETE
poliitazam111@hotmail.com

JOSÉ SANTIAGO BONILLA PAZMIÑO
santiagob86@yahoo.com

DIRECTOR: MSc. ING. MIGUEL ÁNGEL HINOJOSA RAZA.
mhinojosa@epn.edu.ec

Quito, Febrero 2009

DECLARACIÓN

Nosotros, Ana María Zambrano Vizuite y José Santiago Bonilla Pazmiño, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Ana María Zambrano Vizuite

José Santiago Bonilla Pazmiño

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Ana María Zambrano Vizueté y José Santiago Bonilla Pazmiño, bajo mi supervisión.

MSC. Ing. Miguel Ángel Hinojosa Raza
DIRECTOR DE PROYECTO

DEDICATORIA

Sin duda alguna, yo dedicaré este proyecto a las personas más importantes en mi vida, mis padres:

A mi papi, porque se que aún soy su pequeña niña. Le dedico este proyecto, la culminación de mi carrera, por ser quien nunca dejo de trabajar para que no nos faltara nada. Además porque sigue siendo mi héroe, porque es una persona admirable y seguir sus pasos, me ha motivado a caminar para adelante.

A mi mejor amiga, mi compañera, mi confidente, mi bendición, mi vida, mi mami. Sólo Dios sabrá porque me eligió para ser la hija de este ángel. Te dedico este trabajo porque gracias a ti sigo y seguiré porque eres mucho más que mi inspiración
¡Por fin lo logré mamita!

Todos los esfuerzos que haga en mi vida, serán para ustedes.
Porque los amo por siempre, mis personas favoritas.

Anita Zambrano Vizueté.

AGRADECIMIENTO

Gracias a mi Dios, mi ser supremo, mi oyente de peticiones. Gracias por acompañarme cada instante de mi vida y ayudarme a culminar mi primer objetivo, ser Profesional.

Gracias a mis hermanos, por aguantar mis estados de estrés y ser un apoyo incondicional en toda mi vida. Nada de lo que soy sería posible sin ustedes.

A mi gran amigo y profesor, Ing. Felipe Navas, que sin su ayuda desde los primeros niveles, no hubiese llegado a mi objetivo. Gracias por sus consejos y paciencia. Muchas Gracias.

Gracias a mi Director de Tesis, Ing. Miguel Hinojosa, que nos ha dedicado todo el tiempo necesario para poder culminar este proyecto.

A Santi, Danny, George, Julián, Verito, mis compañeros de clases desde hace mucho tiempo. Gracias por acogerme en su grupo, sin ustedes no hubiese disfrutado tanto mi época de universidad.

A mi gran compañero y amigo, gracias por compartir este trabajo conmigo, gracias por no dejarme caer en este duro camino. Te admiro mucho mi “Bolito”.

Anita Zambrano Vizueté.

DEDICATORIA

Al Dr. Riber Donoso, profesor del Instituto Tecnológico Superior Bolívar, por haber sido el mentor de haber venido a estudiar a esta institución. Al Club de Periodismo y Cultura (CPC) de la Escuela Politécnica Nacional, por haberme dado las herramientas y una nueva visión para crear mi proyecto de vida. A Carina, por haber colaborado material y espiritualmente con este proyecto de titulación. Pero principalmente a mi familia, especialmente: mi padre, José; mi madre, Irma; y mis hermanos, Tannia (+), Santiago (+), y Andrea; por haber soportado y apoyado mi ausencia durante cinco años para poder culminar exitosamente con este sueño.

José Santiago Bonilla Pazmiño.

AGRADECIMIENTO

A la vida misma, por darme los mejores padres, los mejores hermanos y hermanas, la mejor familia, los mejores compañeros y amigos, y la mejor compañera de Tesis.

José Santiago Bonilla Pazmiño.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS.....	XII
ÍNDICE DE TABLAS.....	XV
RESUMEN.....	1
PRESENTACIÓN.....	3
CAPÍTULO 1	5
1. ANÁLISIS DE REQUERIMIENTOS DE LA EMPRESA.	5
1.1. DETERMINACIÓN DE LA SITUACIÓN ACTUAL.....	5
1.1.1. CARACTERIZACIÓN DE LA EMPRESA TELCOLSAM S.A.	5
1.1.2. DISTRIBUCIÓN ADMINISTRATIVA DE TELCOLSAM S.A.....	6
1.2. ANÁLISIS DE REQUERIMIENTOS.....	7
1.2.1. RECOPIACIÓN DE INFORMACIÓN al personal de telcolsam.....	7
1.2.1.1. Entrevista a la Gerente General.....	7
1.2.1.2. Entrevista al Departamento de Instalaciones.....	8
1.2.1.3. Entrevista al Departamento de Cobranzas.....	9
1.2.2. DETERMINACIÓN DEL FLUJO DE TRABAJO.....	10
1.2.2.1. Flujo normal de trabajo.....	11
1.2.2.2. Flujo de trabajo con ofrecimiento de pago.....	13
1.2.2.3. Flujo de trabajo con gestión de PDVs.....	14
1.2.3. DETERMINACIÓN DE PROBLEMAS.....	16
1.2.3.1. Limitaciones del uso actual de las hojas de cálculo.....	16
1.2.3.1.1. Validaciones.....	16
1.2.3.1.2. Almacenamiento de la información.....	17
1.2.3.2. Distribución de la información.....	17
1.2.3.3. Modificación de la información.....	18
1.2.3.3.1. Modificación de la información de forma no concurrente.....	18
1.2.3.3.2. Actualización de la información en un solo departamento.....	18
1.2.3.4. Toma de decisiones.....	19
1.2.3.4.1. Información redundante en el momento de consultas.....	19
1.2.3.4.2. Obtención de Información útil para la toma de decisiones.....	20
1.2.3.5. Fallas de tipo humano.....	21
1.2.3.5.1. Fallas de apreciación.....	21
1.2.3.5.2. Fallas del canal de distribución de la información.....	21
1.2.3.6. Limitaciones del actual sistema de administración (SAR-II).....	22
1.3. ESPECIFICACIÓN DE REQUERIMIENTOS DEL SOFTWARE.....	23
1.3.1. INTRODUCCIÓN (SECCIÓN 1 DE LA SRS).....	23
1.3.1.1. Propósito.....	23
1.3.1.2. Alcance.....	24
1.3.1.3. Definiciones, siglas, y abreviaciones.....	26

1.3.1.4.	Referencias.....	28
1.3.1.5.	Apreciación global.....	29
1.3.2.	DESCRIPCIÓN GLOBAL (SECCIÓN 2 DE LA SRS).....	29
1.3.2.1.	Perspectiva del producto.....	29
1.3.2.1.1.	Interfaces del sistema.....	31
1.3.2.1.2.	Interfaz con el usuario.....	31
1.3.2.1.3.	Interfaces con el hardware.....	32
1.3.2.1.4.	Interfaces con el software.....	32
1.3.2.1.5.	Interfaces de comunicaciones.....	32
1.3.2.1.6.	Restricciones de memoria.....	33
1.3.2.1.7.	Requisitos de adaptación del Sitio.....	33
1.3.2.2.	Funciones del producto.....	33
1.3.2.3.	Características del usuario.....	38
1.3.2.4.	Restricciones.....	38
1.3.2.5.	Atención y dependencias.....	39
CAPÍTULO 2	41
2.	CARACTERIZACIÓN DE HARDWARE Y SOFTWARE.....	41
2.1.	SITUACIÓN ACTUAL E INSTALACIONES.....	41
2.2.	INFRAESTRUCTURA DE LA RED.....	42
2.2.1.	EQUIPAMIENTO.....	42
2.2.1.1.	Cuarto de Equipos.....	43
2.2.1.1.1.	Servidores y hardware de apoyo.....	43
2.2.1.1.2.	Equipos de interconectividad.....	47
2.2.1.2.	Terminales de usuarios por departamentos.....	49
2.2.1.2.1.	Departamento de Instalaciones.....	49
2.2.1.2.2.	Departamento de Cobranzas.....	50
2.2.1.2.3.	Gerencia General.....	52
2.2.2.	TOPOLOGÍA ACTUAL DE LA RED.....	52
2.2.2.1.	Cableado Estructurado.....	52
2.2.2.2.	Topología Lógica.....	55
2.3.	DESCRIPCIÓN DE LAS APLICACIONES INSTALADAS.....	57
2.3.1.	SERVIDORES.....	57
2.3.2.	TERMINALES.....	59
CAPÍTULO 3	62
3.	ANÁLISIS Y SELECCIÓN DE LAS HERRAMIENTAS DE SOFTWARE.....	62
3.1.	INTRODUCCIÓN TEÓRICA.....	62
3.1.1.	ARQUITECTURA CLIENTE–SERVIDOR.....	62
3.1.1.1.	Definición.....	62
3.1.1.2.	Justificación.....	65
3.1.1.3.	Ventajas y Desventajas.....	66

3.1.2.	<i>ARQUITECTURA EN CAPAS.</i>	68
3.1.2.1.	Definiciones Previas.	69
3.1.2.1.1.	Interfaz de Programación de Aplicaciones (API).	69
3.1.2.1.2.	.NET Framework	69
3.1.2.1.3.	Kit de Desarrollo de Software de Windows (SDK)	72
3.1.2.1.4.	Entorno Integrado de Desarrollo (IDE)	72
3.1.2.2.	Definición de Capas.	74
3.1.2.2.1.	Capa de Presentación, Capa Usuario o Interfaz de Usuario IU:	75
3.1.2.2.2.	Capa de Negocios, Lógica del Negocio o Reglas del Negocio:	76
3.1.2.2.3.	Capa de Datos:	77
3.1.2.3.	Definición de Niveles.	77
3.2.	<i>SELECCIÓN DE LAS HERRAMIENTAS DE SOFTWARE MÁS ADECUADAS.</i>	78
3.2.1.	<i>INTERFAZ GRÁFICA (PRIMERA CAPA).</i>	78
3.2.1.1.	WINDOWS PRESENTATION FOUNDATION (WPF)	79
3.2.1.1.1.	Definición.	79
3.2.1.1.2.	Lenguaje de Programación empleado en WPF.	81
3.2.1.1.3.	Lo nuevo en WPF con .NET Framework 3.5.	83
3.2.2.	<i>REGLAS DEL NEGOCIO (SEGUNDA CAPA).</i>	85
3.2.3.	<i>SISTEMA DE GESTIÓN DE BASE DE DATOS, SGBD O DBMS.</i>	86
3.2.3.1.	Definiciones.	86
3.2.3.2.	Componentes de una Base de Datos	86
3.2.3.3.	Tipos de Bases de Datos	87
3.2.3.3.1.	Base de Datos Orientada a Objetos.	87
3.2.3.3.2.	Base de Datos Relacional	88
3.2.3.3.3.	Base de Datos Objeto- Relacional	89
3.2.3.3.4.	Comparativa de SGBD.	90
3.2.3.4.	Bases de Datos de Código Abierto	92
3.2.3.4.1.	MySQL	93
3.2.3.4.2.	PostgreSQL	94
3.2.3.4.3.	Firebird	94
3.2.3.5.	Bases de Datos de Código Cerrado	95
3.2.3.5.1.	SQL Server	95
3.2.3.5.2.	Oracle	96
3.2.3.6.	Comparativa de Base de Datos	97
3.2.3.6.1.	Principales Características de las diferentes Bases de Datos	98
3.2.3.6.2.	Ventajas y Desventajas.	100
3.2.3.7.	Elección del SGBD para TELCOLSAM S.A.	105
3.2.4.	<i>OTRAS HERRAMIENTAS A UTILIZAR</i>	108
3.2.4.1.	Conector MySQL/NET.	109
3.2.4.2.	Ribbon.	111
3.2.4.3.	WPFToolKit	112
3.2.4.4.	Microsoft.Office.Interop.Excel	113
3.2.5.	<i>RESUMEN DE LAS HERRAMIENTAS SELECCIONADAS.</i>	115

CAPÍTULO 4	118
4. DISEÑO E IMPLANTACIÓN DEL SISTEMA.....	118
4.1. METODOLOGÍA DE DESARROLLO DE SOFTWARE.....	118
4.1.1. UML (Lenguaje de modelado unificado).....	119
4.1.1.1. Definiciones en UML	121
4.1.2. RUP	123
4.1.2.1. Historia de RUP	123
4.1.2.2. Características Esenciales de RUP.....	124
4.2. POINTSOFT.....	125
4.2.1. REQUISITOS.....	125
4.2.1.1. Objetivo del Proyecto POINTSOFT	125
4.2.2. DIAGRAMAS RUP PARA POINTSOFT.....	125
4.2.2.1. Diagrama de Actividades.....	125
4.2.2.2. Diagrama de Casos de Uso.....	126
4.2.2.2.1. Requerimientos e Identificación de Actores.....	127
4.2.2.2.2. Identificación de Casos de Uso.	128
4.2.2.3. Diagrama de Estructura de Estática (de Clases).....	149
4.2.2.3.1. Especificación de Clases de Diseño.....	149
4.2.2.4. Diagrama de Eventos	158
4.2.2.5. Diagrama de Interacción.....	159
4.2.3. DISEÑO DE LA ARQUITECTURA TRES CAPAS.....	161
4.2.3.1. Capa de Base de Datos.....	161
4.2.3.1.1. Definición de Tablas POINTSOFT	162
4.2.3.1.2. Definición de Procedimientos Almacenados de POINTSOFT	168
4.2.3.1.3. Definición de Funciones de POINTSOFT	169
4.2.3.1.4. Migración de datos a MySQL.....	170
4.2.3.2. Capa de Reglas del Negocio.....	177
4.2.3.2.1. Especificación de Librerías.....	177
4.2.3.2.2. Secuencia de Estados de Puntos de Venta.....	184
4.2.3.3. Capa Interfaz Gráfica	187
4.2.3.3.1. Diseño de Interfaces del Sistema.	187
4.2.3.3.2. Navegación.....	187
4.2.3.3.3. Descripción de Botones Principales.	190
4.2.3.3.4. Descripción de Pantallas.	192
4.2.3.3.5. Descripción del Controles ausentes en WPF.....	207
4.2.3.3.6. Validaciones en WPF.....	208
4.2.3.3.7. Respuestas de Éxito ó Error.....	213
4.2.4. IMPLANTACIÓN DE POINTSOFT.....	216
4.2.4.1. Servidor SERVER (Base de Datos).....	216
4.2.4.2. Terminales.....	217
4.2.4.3. Conexión TCP/IP.....	218
4.2.4.4. Guía de Instalación POINTSOFT para el Usuario.....	219

4.2.5. PRUEBAS Y RESULTADOS.....	219
4.2.5.1. Realización de Pruebas diarias.....	219
4.2.5.2. Resultados de las Pruebas.....	220
4.2.5.3. Evaluación Final.....	223
CAPÍTULO 5	226
5. CONCLUSIONES Y RECOMENDACIONES.....	226
5.1. CONCLUSIONES	226
5.2. RECOMENDACIONES.....	229
6. BIBLIOGRAFÍA.....	232
6.1. LIBROS	232
6.2. PÁGINAS WEB	233

ANEXOS

ANEXO 1: XAML.

ANEXO 2: Licencia Ribbon de Software.

ANEXO 3: Control MaskedTextBox.

ANEXO 4: Modelo Vista Controlador y contraposición con 3-Capas.

ANEXO 5: Manual de Usuario POINTOSFT.

ÍNDICE DE FIGURAS

Figura 1.1: Diagrama organizacional de TELCOLSAM S.A.	6
Figura 1.2: Diagrama del flujo normal de trabajo.....	11
Figura 1.3: Diagrama del flujo de trabajo con ofrecimiento de pago.	13
Figura 1.4: Flujo de trabajo con gestión de PDVs.	15
Figura 1.5: Diagrama de Alto Nivel de Casos de Uso de POINTSOFT.....	30
Figura 2.1: Cuarto de equipos en TELCOLSAM S.A.....	43
Figura 2.2: Rack de equipos de interconectividad.....	47
Figura 2.3: Plano de TELCOLSAM S.A. y TELINEMAX S.A.....	53
Figura 2.4: Cableado en TELCOLSAM S.A.....	54
Figura 2.5: Diagrama lógico de la red de TELCOLSAM S.A.	56
Figura 3.1: Arquitectura Cliente – Servidor.....	62
Figura 3.2: Petición – Respuesta en un modelo Cliente - Servidor.	64
Figura 3.3: WPF y .Net Framework 2.0.....	71
Figura 3.4: Arquitectura en tres capas.	74
Figura 3.5: Capa de Presentación.....	75
Figura 3.6: Ejemplo de formulario Microsoft ® Office Word.	76
Figura 3.7: Lógica de Negocios.....	76
Figura 3.8: Capa de Acceso a Datos.....	77
Figura 3.9: Niveles en una Arquitectura en Tres Capas.....	78
Figura 3.10: Ejemplo de lenguaje de Marcado XML y su respectiva representación GUI.....	81
Figura 3.11: Ejemplo de lenguaje Subadyacente, Visual Basic .NET.	82
Figura 3.12: Ejemplo de un diseño ODMG.....	88
Figura 3.13: Ejemplo de un diseño SQL.....	89

Figura 3.14: MySQL Query Browser.	94
Figura 3.15: Componentes Básicos de SQL Server	96
Figura 3.16: Arquitectura de POINTSOFT.	115
Figura 4.1: Aproximación UML.	119
Figura 4.2: Principales aportes a UML.	121
Figura 4.3: Principales diagramas de UML.	122
Figura 4.4: Historia de RUP.....	123
Figura 4.5: Casos de Uso Global en POINTSOFT.....	128
Figura 4.6: Sub-Casos de Uso en el paquete Autenticación.	129
Figura 4.7: Sub-Casos de Uso en el Paquete Localización.	130
Figura 4.8: Sub-Casos de Uso de País, Provincia, Ciudad, Sector, y Barrio.	130
Figura 4.9: Sub-Casos de Uso en el paquete Propietario.	132
Figura 4.10: Sub-Casos de Uso en el paquete Punto de Venta (Datos).	133
Figura 4.11: Sub-Casos de Uso en el paquete Gestión Estados.	134
Figura 4.12: Sub-Casos de Uso en el paquete Instalación.	135
Figura 4.13: Sub-Casos de Uso en Mayorista.....	135
Figura 4.14: Sub-Casos de Uso en Tipo Instalación.	136
Figura 4.15: Sub-Casos de Uso en Instalación PDV.....	137
Figura 4.16: Sub-Casos de Uso en el paquete Personal.	138
Figura 4.17: Sub-Caso de Uso en el Paquete Ruta.	139
Figura 4.18: Sub-Casos de Uso en Periodo.....	140
Figura 4.19: Sub-Casos de Uso en Ruta.....	141
Figura 4.20: Sub-Casos de Uso en el paquete Factor Valor.	142
Figura 4.21: Sub-Casos de Uso en el paquete PDV (Cobranzas).....	143
Figura 4.22: Sub-Casos de Uso en Comisión por PDV.....	143
Figura 4.23: Sub-Casos de Uso en Ruta-Recaudador por PDV.....	144

Figura 4.24: Sub-Casos de Uso en el paquete Reporte Recaudación / SalDOS.	145
Figura 4.25: Sub-Casos de Uso en Gestión de Cobranzas.....	146
Figura 4.26: Sub-Casos de Uso en Reporte de Recaudación.....	147
Figura 4.27: Sub-Casos de Uso en el paquete Reportes.	148
Figura 4.28. Diagrama de Clases POINTSOFT.	150
Figura 4.29: Clases de Localización POINTSOFT.	151
Figura 4.30: Clases Persona POINTSOFT.....	152
Figura 4.31: Clase Perfil POINTSOFT.	152
Figura 4.32: Clase Punto de Venta POINTSOFT.....	153
Figura 4.33: Clases Instalación POINTSOFT.....	153
Figura 4.34: Clases Factor Valor POINTSOFT.	154
Figura 4.35: Clases Ruta-Periodo POINTSOFT.....	155
Figura 4.36: Clases Estado POINTSOFT.....	155
Figura 4.37: Clase Mayorista POINTSOFT.....	156
Figura 4.38: Clase Gestión Cobranzas.	156
Figura 4.39: Clase Reporte de Recaudación POINTSOFT.....	157
Figura 4.40: Diagrama de Interacción País, POINTSOFT.....	160
Figura 4.41. Diagrama de Base de Datos, POINTSOFT.....	161
Figura 4.42: Diagrama de Estados de un Punto de Venta.	186
Figura 4.43: Diagrama de Navegación, POINTSOFT.	189
Figura 4.44: Diagrama de Navegación de Reportes, POINTSOFT.....	190
Figura 4.45: Ventana principal de POINTSOFT.....	192
Figura 4.46: Diagrama de Conexión Actual.....	218

ÍNDICE DE TABLAS

Tabla 1.1: Ejemplo de conversión de texto en número.....	16
Tabla 1.2: Ejemplo de redundancia en las instalaciones.....	19
Tabla 1.3: Módulos para la gestión de localizaciones.	33
Tabla 1.4: Módulos para la gestión de localizaciones.	34
Tabla 1.5: Módulos para la gestión de propietarios.	34
Tabla 1.6: Módulos para la gestión de PDVs.....	34
Tabla 1.7: Módulos para la gestión de PDVs.....	34
Tabla 1.8: Módulos para la gestión de instalaciones.	35
Tabla 1.9: Módulos para la gestión de factores de valor.	35
Tabla 1.10: Módulos para la gestión de rutas.....	35
Tabla 1.11: Módulos para la gestión de cobranzas de PDV.....	35
Tabla 1.12: Módulos para la gestión de reportes de recaudación y saldos.....	36
Tabla 1.13: Módulos para la gestión de personal	36
Tabla 1.14: Módulos para la generación de reportes.	36
Tabla 2.1: Detalle de los servidores y hardware de apoyo de TELCOLSAM S.A.	47
Tabla 2.2: Detalle de equipos de interconectividad.	49
Tabla 2.3: Características Físicas del Terminal de Instalaciones.....	50
Tabla 2.4: Características Físicas de los Terminales de Cobranzas.....	51
Tabla 2.5: Características Físicas del Terminal de Gerencia General.	52
Tabla 2.6: Direccionamiento lógico de TELCOLSAM S.A.	55
Tabla 2.7: Aplicaciones instaladas en los servidores.	58
Tabla 2.8: Aplicaciones instaladas en los terminales.	60
Tabla 3.1: Justificación de la Arquitectura Cliente-Servidor	66
Tabla 3.2: Ventajas Cliente–Servidor.....	67
Tabla 3.3: Inconvenientes de una arquitectura Cliente-Servidor.	68
Tabla 3.4: Comparación de SGBDs	91
Tabla 3.5: Detalle de Características Principales de los SGBD	100

Tabla 3.6: Ventajas de SGBD's.....	103
Tabla 3.7: Desventajas de SGBD's.....	104
Tabla 3.8. MySQL en TELCOLSAM S.A.....	108
Tabla 3.9: Clases del Conector .NET.....	110
Tabla 4.1: Ventajas de una Metodología Orientada a Objetos.	120
Tabla 4.2: Objetivo y Descripción de POINTSOFT.....	125
Tabla 4.3: Actores presentes en POINTSOFT.....	127
Tabla 4.4: Descripción Sub-Casos de Uso en paquete Autenticación.	129
Tabla 4.5: Descripción de Sub-Casos de Uso en paquete Localización.	131
Tabla 4.6: Descripción de Sub-Casos de Uso en paquete Propietario.	132
Tabla 4.7: Descripción Sub-Casos de Uso en paquete Punto de Venta (Datos).	133
Tabla 4.8: Descripción Sub-Casos de Uso en el paquete Gestión Estados.....	134
Tabla 4.9: Descripción Sub-Casos de Uso en Mayorista.	136
Tabla 4.10: Descripción Sub-Caso de Uso Tipo Instalación.....	137
Tabla 4.11: Descripción Sub-Casos de Uso en Instalación de PDV.	138
Tabla 4.12: Descripción Sub-Casos de Uso en Personal.....	139
Tabla 4.13: Descripción Sub-Caso de Uso Periodo.....	140
Tabla 4.14: Descripción Sub-Casos de Uso en Ruta.	141
Tabla 4.15: Descripción Sub-Caso de Uso Factor de Valor.	142
Tabla 4.16: Sub-Casos de Uso en Comisión por PDV.	144
Tabla 4.17: Sub-Casos de Uso en Ruta-Recaudador por PDV.....	145
Tabla 4.18: Descripción Sub-Casos en Gestión de Cobranzas.....	146
Tabla 4.19: Descripción Sub-Casos de Uso en Reporte de Recaudación.	147
Tabla 4.20: Descripción Sub-Casos de Uso en el paquete Reportes.....	148
Tabla 4.21: Diagrama de Eventos, Sub-Caso de Uso Localización.	159
Tabla 4.22: Descripción Tabla Barrio.....	162
Tabla 4.23: Descripción Tabla Ciudad.	162
Tabla 4.24: Descripción Tabla Empleado.	162
Tabla 4.25: Descripción Tabla Estado.	163
Tabla 4.26: Descripción Tabla Factor Valor.....	163

Tabla 4.27: Descripción Tabla Punto de Venta.....	164
Tabla 4.28: Descripción Tabla Mayorista.....	164
Tabla 4.29: Descripción Tabla País.	164
Tabla 4.30: Descripción Tabla Perfil.	164
Tabla 4.31: Descripción Tabla Periodo.	165
Tabla 4.32: Descripción Tabla Persona.	165
Tabla 4.33: Descripción Tabla Provincia.....	165
Tabla 4.34: Descripción Tabla Punto de Venta.....	166
Tabla 4.35: Descripción Tabla Reporte de Recaudación.	167
Tabla 4.36: Descripción Tabla Ruta.....	167
Tabla 4.37: Descripción Tabla Sector.	167
Tabla 4.38: Descripción Tipo Factor Valor.....	167
Tabla 4.39: Descripción Tipo Instalación.	168
Tabla 4.39: Descripción de Procedimientos Almacenados de País	169
Tabla 4.40: Descripción de Funciones en POINTSOFT.	170
Tabla 4.42: Descripción Gráfica Ventana Principal.....	193
Tabla 4.43: Descripción Gráfica Pestaña País	194
Tabla 4.44: Descripción de campos en la Pestaña País	194
Tabla 4.45: Descripción Gráfica Pestaña Provincia.....	194
Tabla 4.46: Descripción de campos en la Pestaña Provincia.....	194
Tabla 4.47: Descripción Gráfica Pestaña Ciudad	195
Tabla 4.48: Descripción de campos en la Pestaña Ciudad	195
Tabla 4.49: Descripción Gráfica Pestaña Sector	195
Tabla 4.50: Descripción de campos en la Pestaña Sector	195
Tabla 4.51: Descripción Gráfica Pestaña Barrio	196
Tabla 4.52: Descripción de campos en la Pestaña Barrio	196
Tabla 4.53: Descripción Gráfica Pestaña Propietario.	196
Tabla 4.54: Descripción de campos en la Pestaña Propietario	197
Tabla 4.55: Descripción Gráfica Pestaña Punto de Venta.....	197
Tabla 4.56: Descripción de campos en la Pestaña Punto de Venta.....	198

Tabla 4.57: Descripción Gráfica Pestaña Mayorista.	198
Tabla 4.58: Descripción de campos en la Pestaña Mayorista.	198
Tabla 4.59: Descripción Gráfica Pestaña Tipo Instalación.	199
Tabla 4.60: Descripción de campos de la Pestaña Tipo Instalación.	199
Tabla 4.61: Descripción Gráfica Pestaña Instalación.	199
Tabla 4.62: Descripción de campos de la Pestaña Instalación.	200
Tabla 4.63: Descripción Gráfica Pestaña Gestión de Cobranzas.....	200
Tabla 4.64: Descripción de campos de la Pestaña Gestión.	200
Tabla 4.65: Descripción Gráfica Pestaña Factor Valor.	201
Tabla 4.66: Descripción de campos de la Pestaña Factor Valor.	201
Tabla 4.67: Descripción Gráfica Pestaña Periodo.	201
Tabla 4.68: Descripción de campos de la Pestaña Periodo.	202
Tabla 4.69: Descripción Gráfica Pestaña Ruta.	202
Tabla 4.70: Descripción de campos de la Pestaña Ruta.	202
Tabla 4.71: Descripción Gráfica Pestaña Comisión por PDV.....	202
Tabla 4.72: Descripción de campos en la Pestaña Comisión por PDV.	203
Tabla 4.73: Descripción Gráfica Pestaña Ruta y Recaudador por PDV.....	203
Tabla 4.74: Descripción de campos de Pestaña Ruta y Recaudador por PDV.....	203
Tabla 4.75: Descripción Gráfica Pestaña Reporte de Recaudación.....	203
Tabla 4.76: Descripción de Campos de la Pestaña Reporte de Recaudación.	204
Tabla 4.77: Descripción Gráfica Pestaña Personal.	205
Tabla 4.78: Descripción de campos de la ventana Personal.	206
Tabla 4.79: Descripción Gráfica Pestaña Personal.	206
Tabla 4.80: Archivos para la Instalación en los Terminales.....	217
Tabla 4.81: Resultado de Pruebas en el Departamento de Instalaciones.....	221
Tabla 4.82: Resultado de Pruebas en el Departamento de Cobranzas	222
Tabla 4.83: Resultado de Pruebas en la Gerencia General.	223
Tabla 4.84: Evaluación por parámetros de POINTSOFT.	224

RESUMEN

El objetivo de este proyecto es la solución a dificultades que se presentan actualmente en la empresa TELCOLSAM S.A. en su departamento de Cobranzas. Se crea un nuevo programa llamado POINTSOFT, que brinda solución a estos problemas; y por tanto, es el encargado de aliviar el trabajo de los empleados, llevar reportes reales y actuales a los principales directivos de la empresa. Además de ofrecer concurrencia de Usuarios y validación de la Información. Este nuevo proyecto, tiene además la incorporación de la herramienta Windows Presentation Foundation en la capa de Usuario, con la finalidad de brindar al usuario una interfaz amigable, intuitiva y de fácil uso.

CAPITULO 1: Análisis de requerimientos de la empresa.

Como primera instancia, se detalla la estructura organizacional que tiene la empresa. Siguiendo se realiza la obtención de Requerimientos de Software siguiendo el estándar *IEEE Std 830-1998*, para lo cual previamente se analiza el flujo de tráfico existente entre los departamentos de Instalaciones y Cobranzas.

Por último para la definición de componentes de Software, se sigue la metodología RUP utilizando UML 2.2.

CAPITULO 2: Caracterización del software y hardware existentes.

Primeramente, se realiza un análisis de la situación actual de TELCOLSAM S.A., definiendo también su empresa aliada TELINEMAX S.A.

Se efectúa un resumen exhaustivo de la infraestructura de la empresa; Comenzando por el detalle de su estructura física contemplando: Espacio físico, distribución de puestos, de puntos de salidas de Telecomunicaciones y salidas telefónicas. Continúa con el análisis de su estructura lógica incluyendo: Equipamiento activo y pasivo actual, diagrama lógico de red, direccionamiento IP y Software instalado.

CAPITULO 3: Análisis y selección de las herramientas de Software.

Este capítulo se centra en conocer las herramientas para la solución de TELCOLSAM S.A.; para lo cual, comenzamos con una introducción teórica definiendo una arquitectura Cliente – Servidor y una arquitectura en tres capas. Para la primera capa (Interfaz Gráfica), se definen los conceptos básicos de Windows Presentation Foundation. En la segunda (Capa de Negocios) y tercera Capa (Base de Datos), se analizan posibles opciones y se deducen las herramientas más adecuadas.

CAPITULO 4: Diseño e implementación del Sistema.

Mediante una metodología de Desarrollo de Software llamada RUP, basada en Objetos y UML, se realiza: El diseño, Ingeniería de Negocios, Implementación y Pruebas.

La primera fase, de diseño, abarca un conjunto de diagramas empleados por RUP, que ayudan al entendimiento y visión del Sistema.

La Ingeniería de Negocios, se refiere a la programación de reglas, clases y clases de conexión llamadas DALs.

La forma de instalación y los archivos necesarios para la misma, son parte de la etapa de Implementación.

Se realizan pruebas del Sistema POINTSOFT, con el número de usuarios esperados, mediante pruebas de Carga. La etapa de Pruebas, se realizan en cada departamento, identificando problemas y errores de programación, dando su solución.

CAPITULO 5: Conclusiones y Recomendaciones.

Se presentarán las conclusiones obtenidas del desarrollo del proyecto y recomendaciones para poder mejorarlo.

PRESENTACIÓN

Gracias a la gran competencia existente actualmente, las empresas se ven obligadas a sistematizar procesos para ahorro de tiempo y dinero. Entre más flujo de información exista, más difícil se tornará el trabajo manual, además de tener un alto grado de posibilidad a errores. Este es el caso de TELCOLSAM S.A., una empresa relativamente nueva que pretende sobresalir del mercado con una nueva Aplicación.

Presentamos una nueva aplicación: **Sistema de Administración Modular de Puntos de Venta POINTSOFT**. Este proyecto, se centra en la automatización de la Gestión de Cobranzas en la empresa; siguiendo una metodología Orientada a Objetos y encontrando las herramientas más adecuadas para la realización del mismo.

POINTSOFT es una aplicación modular en capas, donde su mayor trabajo se ha centrado en la Interfaz Gráfica de Usuario; Al utilizar la nueva tecnología Windows Presentation Foundation, brindamos al usuario un programa amigable, intuitivo, eficiente, con ayudas para ahorro de tiempo de trabajo tal como: accesos rápidos, validación dinámica y ventanas informativas.

Por otro lado, **POINTSOFT**, gestiona la información con la ayuda de un motor de base de datos inteligente llamado MySQL. Proporcionando así a TELCOLSAM S.A. datos reales y actuales.

Esperamos que este trabajo, se convierta en un aporte para la ingeniería, convirtiéndose un pilar para la investigación de la nueva tecnología WPF.

CAPITULO 1



1. ANÁLISIS DE REQUERIMIENTOS DE LA EMPRESA.

1.1. DETERMINACIÓN DE LA SITUACIÓN ACTUAL.

1.1.1. CARACTERIZACIÓN DE LA EMPRESA TELCOLSAM S.A.

TELCOLSAM S.A. fue creado en el año 2005. TELCOLSAM S.A brinda servicio de reventa de telefonía celular, junto con su distribuidor MOVISTAR. Sin embargo, en el año que fue creada la empresa, MOVISTAR aún no llegaba a Ecuador; por tanto, trabaja con dos mayoristas encargados de proveerles los equipos: CABICEL y TECEMOVIL.

TELCOLSAM S.A se caracteriza por realizar instalaciones de puntos de reventa en toda le región a nivel nacional, trabajando directamente con el operador MOVISTAR. Estos puntos de reventa pueden ser monocabinas o bicabinas, siendo más frecuentes las segundas.

La empresa trabaja con un porcentaje de comisión por cada punto de venta, siendo actualmente este porcentaje entre un 10% y un 45%. Este valor es individualmente asignado a cada uno, lo que significa que se paga esta comisión al cliente por la operación del sistema y el préstamo del espacio físico que éste ocupa en el negocio, quedando el resto de ganancias para la empresa.

Para poder realizar el cobro en cada punto, TELCOLSAM hace uso de “autoimpresores” para imprimir los comprobantes de venta. Los autoimpresores son controladores para las cabinas que funcionan desde un computador normal y son autorizadas por el Servicio de Rentas internas (S.R.I).

1.1.2. DISTRIBUCIÓN ADMINISTRATIVA DE TELCOLSAM S.A.

TELCOLSAM S.A. presenta el siguiente organigrama, dividido por departamentos:

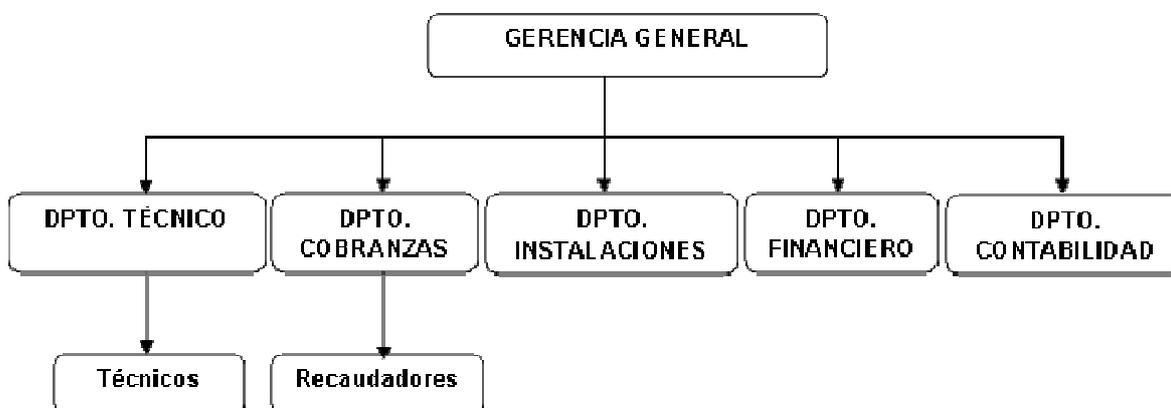


Figura 1.1: Diagrama organizacional de TELCOLSAM S.A.

Actualmente, consta de tres departamentos básicos para su funcionamiento:

- **Departamento de Instalaciones:** Este departamento requiere, por el momento, de tan solo una persona: la Sra. Celia María Rivas. Ella es digitadora de datos en el Departamento de Instalaciones y recepcionista en TELCOLSAM S.A., encargada del ingreso correcto de la información proveniente de la Gerencia General. Por lo cual, el jefe directo de este departamento es la Gerente, la Sra. Cecilia Ponce con C.I. 171056030-9.

Cada dato incorporado hace referencia a cada uno de los diferentes clientes que han hecho contrato con la empresa para la instalación de cabinas. Por tanto, es necesario que este proceso se realice de manera eficiente, sin falla alguna, porque esta será información requerida por otros departamentos.

- **Departamento de Cobranzas:** Por el momento, en este departamento trabajan dos personas: Sr. William Guachamín y Sr. Estuardo Siza, y de igual manera están bajo la supervisión de la Gerente.

Este departamento es el encargado de recibir la información llegada del Departamento de Instalaciones e incorporar otros datos más detallados, referentes a la parte de Cobranzas.

Como su nombre lo muestra, su objetivo es la administración y gestión de la parte financiera de la empresa; o sea, poder tomar decisiones precisas y a tiempo de clientes con deuda, morosos o en regla.

- **Departamento Técnico:** La persona encargada del soporte dentro de la empresa es el MBA. Ing. Marcelo Zambrano, Gerente Técnico de la empresa TELCOLSAM S.A.

Su objetivo es encargarse que el trabajo de la empresa no sea interrumpido por algún fallo en el sistema. Cualquier empleado lleva su queja referente a aspectos técnicos hasta este lugar, para poder seguir su trabajo sin inconveniente alguno.

1.2. ANÁLISIS DE REQUERIMIENTOS.

1.2.1. RECOPIACIÓN DE INFORMACIÓN AL PERSONAL DE TELCOLSAM.

1.2.1.1. Entrevista a la Gerente General.

La entrevista se realizó a la Sra. Cecilia Ponce, Gerente General de TELCOLSAM S.A. Luego de la entrevista se puede resumir las experiencias y actividades de la empresa.

TELCOLSAM S.A. realiza instalación de cabinas en varios puntos a nivel nacional, siendo de mayor acogida la ubicación de bicabinas. El objetivo del Departamento de Cobranzas, en la empresa, se centra en poder tomar decisiones precisas y a tiempo, respecto a gestiones realizadas sobre los puntos de venta. Actualmente, estas decisiones no pueden ser tomadas a tiempo, pudiendo demorarse incluso hasta un mes desde que se detecta el problema hasta que se soluciona, siendo ésta la principal preocupación de la Gerente.

Para que el Departamento de Cobranzas pueda empezar su trabajo es necesario que el Punto de Venta (PDV) se encuentre previamente instalado. Esto significa que, primero, el Departamento de Instalaciones haya incluido ya en sus archivos

este PDV; y segundo, lo haya mandado al Departamento de Cobranzas. Aquí radica otro problema, el paso de información entre estos dos departamentos es de manera manual, haciéndolo a través de una hoja impresa de Microsoft Office Excel. La Gerente expresa su preocupación ya que anteriormente ha ocurrido que esta hoja se ha extraviado por accidente y, por tanto, el proceso no se sigue con normalidad, retrasando todos los procesos dependientes junto con la toma de decisiones.

Por lo citado anteriormente, hemos podido comprobar que los dos departamentos están ampliamente ligados por su información. Para poder tener una administración total del Departamento de Cobranzas, hemos decidido realizar también la administración del Departamento de Instalaciones comenzando por sus necesidades y requerimientos.

TELCOLSAM S.A. tiene un programa que serviría para la administración de los módulos, junto con una base de datos en Firebird, llamado SAR II. Sin embargo, éste no satisface todos los requerimientos para la gestión de cobranzas, ya que no cumple con los requerimientos necesarios básicos.

1.2.1.2. Entrevista al Departamento de Instalaciones.

La entrevista se realizó a la Sra. Celia María Rivas.

El objetivo del Departamento de Instalaciones es presentar información verídica y a tiempo al Departamento de Cobranzas. Para cumplir lo mencionado, debe realizar diferentes operaciones sobre cada PDV. Estas operaciones son:

- Instalación de PDVs.
- Gestiones a PDVs previamente instalados (reubicación de cabinas, bloqueo de los equipos, cambio de equipos, reparaciones, otros dependiendo de la necesidad).
- Cambios de estado a PDVs, según le compete.
- Archivo de información detallada de PDVs.

- Archivo de información detallada de clientes.
- Asignación de los mayoristas que proveen los equipos a los PDVs.
- Asignación de los equipos bases a reutilizar por los Puntos de Venta.
- Creación de la hoja de trabajo, en Microsoft Office Excel, y envío de la misma por correo electrónico a los proveedores.
- Otros, asignados por la Gerencia General.

1.2.1.3. Entrevista al Departamento de Cobranzas.

La entrevista se realizó individualmente a cada uno de los dos empleados del departamento, y también de manera conjunta.

La meta final del trabajo diario de estas personas es proporcionar reportes actualizados a la Gerencia General, todos los meses, para garantizar la continuidad del negocio. Sin embargo, esta tarea no es cumplida por la manera manual de realizar los procesos.

Las tareas que debe cumplir el Departamento de Cobranzas son:

- Cambios de estado de PDVs, según le competa.
- Creación y asignación de rutas a los PDVs.
- Asignación de un recaudador por cada PDV.
- Asignación y actualización de comisiones a cada PDV.
- Archivo de los reportes de recaudación y recibos provenientes de los PDV.
- Gestión de cobranzas para PDVs con cartera vencida.

Los empleados de este departamento aducen que el trabajo realizado es muy largo y tedioso, ya que se debe ingresar una gran cantidad de datos diariamente,

donde cada dígito es importante, por lo que si falla tan solo uno de los números el proceso total falla.

1.2.2. DETERMINACIÓN DEL FLUJO DE TRABAJO.

Para la determinación del flujo de trabajo se seguirá la metodología **DSSD** (Data Structure and Systems Development) o en español **DSED** (Desarrollo de Sistemas basado en Estructura de Datos), también conocida como metodología de Warnier/Orr. DSED consta de un conjunto de normas y reglas para encontrar la solución a un problema, visualizando como se mueven los datos de una entidad a otra. Las definiciones, siglas, y abreviaciones descritas en el **Título “Definiciones, siglas, y abreviaciones.”**, **página 26**, serán las empleadas a continuación.

Siguiendo la metodología se deben contestar 3 preguntas:

- **¿Cuáles son los elementos de información que han de procesarse?**

Principalmente la información que se procesa se centra en los PDVs. Por lo cual, como datos entran: datos de localización, datos del cliente, mayoristas, recaudadores, comisiones, reportes de recaudación, y demás información relevante para la empresa.

- **¿Quiénes o qué son los productores y los consumidores de la información?**

Estas entidades son los dos departamentos detallados anteriormente; comenzando por el Departamento de Instalaciones y, siguiente, el Departamento de Cobranzas, ambos son los encargados de procesar toda la información. Es importante destacar la presencia del recaudador, dentro del Departamento de Cobranzas, quien recopila los reportes de recaudación de los PDVs.

- **¿Cómo ve la información cada productor en el contexto de los demás grupos?**

El Departamento de Cobranzas depende en gran parte del Departamento de Instalaciones, ya que cobranzas no reconoce ningún PDV que instalaciones no haya incorporado a su información. El recaudador en este departamento es de gran importancia, debido a que pueden darse diferentes procesos al momento de hacer la visita al cliente como se detallará posteriormente.

El Departamento de Instalaciones mira a esta información como el cumplimiento de su trabajo y Cobranzas como el requisito para empezar el suyo.

Además, DSED utiliza una notación muy parecida a un diagrama de flujo de datos. Con un círculo se identifica a cada entidad, y con flechas se define el flujo entre éstos.

1.2.2.1. Flujo normal de trabajo.

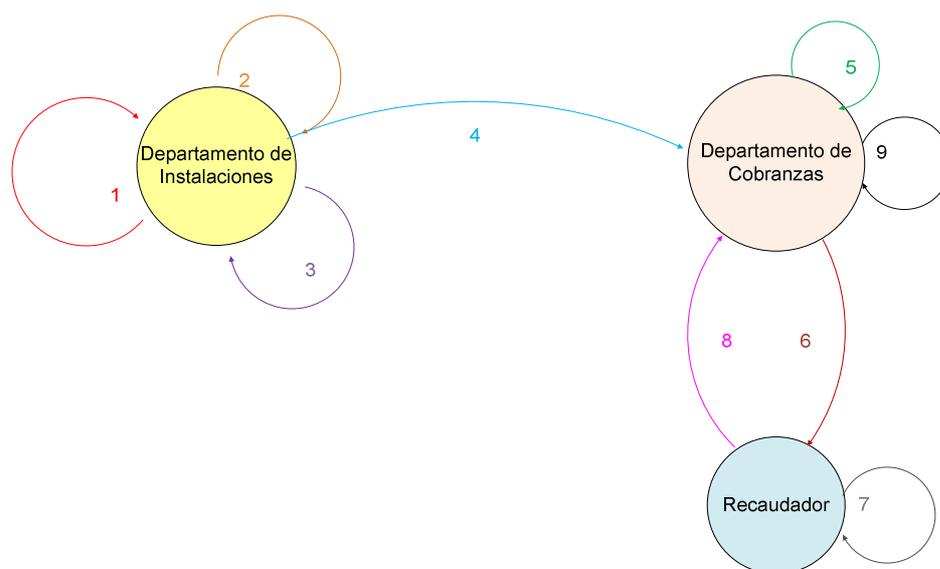


Figura 1.2: Diagrama del flujo normal de trabajo.

- 1) El Departamento de Instalaciones comienza el trabajo. Realiza la creación de nuevos PDVs, lo que significa que un negocio ya establecido ha pedido la instalación de cabinas por medio de la empresa. El PDV se crea con sus datos más importantes como: datos del cliente, localización exacta, código de identificación, teléfonos, entre otros.
- 2) Luego de que se ha creado un PDV, se deben verificar las condiciones de la instalación, esto implica: acordar la fecha y hora de la instalación, y qué y cuántos equipos deben instalarse. Este proceso toma el nombre de “Realizar un pedido de instalación”. En este momento el PDV se encuentra en el estado “PREPARANDO INSTALACIÓN”.
- 3) Siguiendo a este proceso, es necesario identificar de donde se obtendrán los equipos que se van instalar.
- 4) Toda la información previamente ingresada debe ser enviada al otro departamento; actualmente, se lo hace con hojas de cálculo impresas de Microsoft Office Excel.
- 5) Para que este PDV pueda ser cobrado es necesario asignar los siguientes parámetros: un recaudador, una ruta y una comisión. En este paso el PDV pasa al estado “ACTIVO”.
- 6) Los recaudadores son avisados de su nuevo PDV, el cual deben visitarlo en los días y frecuencia de tiempo establecidos en la ruta, para recaudar el dinero.
- 7) Junto al dinero el cliente debe entregar los reportes de recaudación (tickets) impresos por el tarifador. Los tickets son numerados y siempre siguen una secuencia. Estos son el sustento que tiene el cliente para entregar la cantidad correcta de dinero.
- 8) El dinero y todos los reportes de recaudación deben ser entregados al Departamento de Cobranzas. Los encargados del departamento, deben verificar, con ayuda de los tickets, el saldo entregado. Existen situaciones en las que los valores no coinciden por algún motivo y se debe investigar donde se produjo un fallo: en la entrega del dinero por parte del cliente o en la entrega del recaudador.

9) Ahora, en el Departamento de Cobranzas se registra cada ticket. Este proceso es susceptible a fallas debido a la cantidad de números que se deben ingresar manualmente.

Terminado este proceso, ya es posible obtener reportes, pero este proceso también es demoroso porque es realizado con filtros en las hojas de cálculos. Posteriormente, se los imprime para su entrega a la Gerencia General. Las fases 7, 8, 9 se repiten de manera frecuente, si no existe inconveniente alguno.

1.2.2.2. Flujo de trabajo con ofrecimiento de pago.

Adicional al proceso normal, pueden darse otros imprevistos, planteados a continuación:

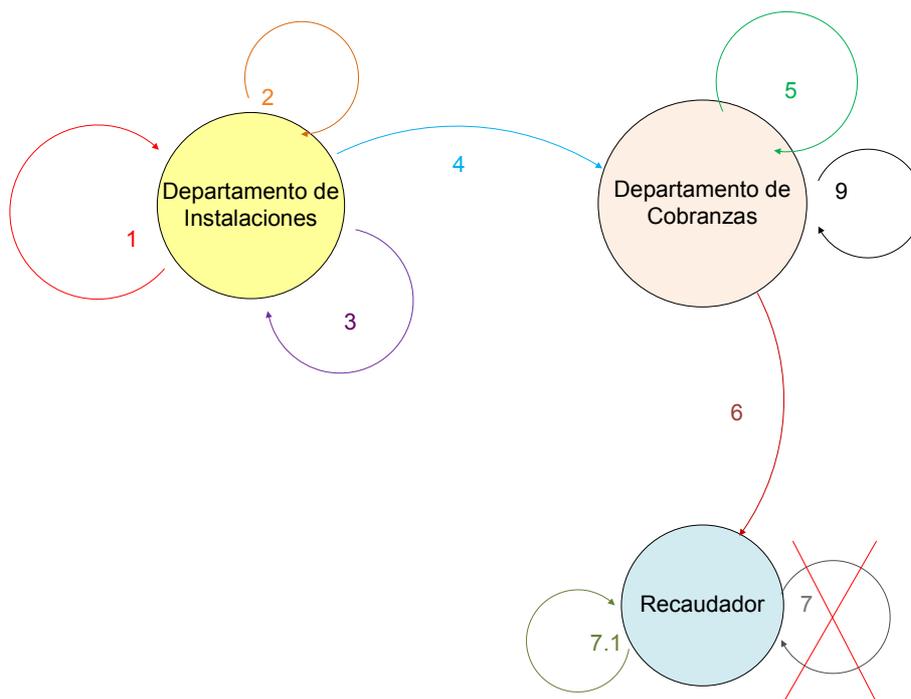


Figura 1.3: Diagrama del flujo de trabajo con ofrecimiento de pago.

7.1) Cuando el recaudador vaya a visitar al cliente, es posible que éste no tenga el dinero total que debe entregar, o tenga sólo un parcial de la cantidad

total. Esto sucede por algún imprevisto que pudo haber tenido el cliente, como:

- Hubo un robo en el local y se llevaron el dinero.
- Tuvo que ocupar el dinero por problemas familiares.
- No encuentra el dinero ese día.
- Otros.

Como este proceso puede hacerse muy frecuente, el Departamento de Cobranzas ingresa un “ofrecimiento de pago” para este PDV, en el cual registra la causa de la falta, cuánto ofrece pagar la próxima vez, y el día de visita.

Como política de la empresa, se podría admitir un máximo de tres “ofrecimientos de pago”. Cuando se cumplan estos tres ofrecimientos se debe suspender el servicio hasta que se iguale en la deuda. Aquí es necesario cambiar el estado del PDV a “INACTIVO”. Sin embargo, los reportes no son generados a tiempo y no se pueden suspender a tiempo los servicios a estos clientes con mora.

Mientras el PDV esté inactivo y el cobro siga pendiente, se añade el estado “COBRANDO” al PDV inactivo.

Si el cliente sigue en mora por más tiempo, se debe tomar la decisión de plantear una demanda legal. Para lo cual es necesario establecer el estado del PDV a “LEGAL”, además de “INACTIVO”, mientras dure la demanda.

1.2.2.3. Flujo de trabajo con gestión de PDVs.

Finalmente, puede realizarse alguna gestión sobre el PDV. Por ejemplo pueden darse reubicaciones o retiros. Todas estas gestiones deben estar detalladas con horas y fechas. En cualquiera de los casos anteriores, es posible que el cliente requiera que se haga una reubicación de los equipos gracias a que el negocio sufra cambios físicos; o también, puede ser que el cliente no requiera más el

servicio y pida el retiro de los equipos. Detallamos el flujo de datos para este proceso:

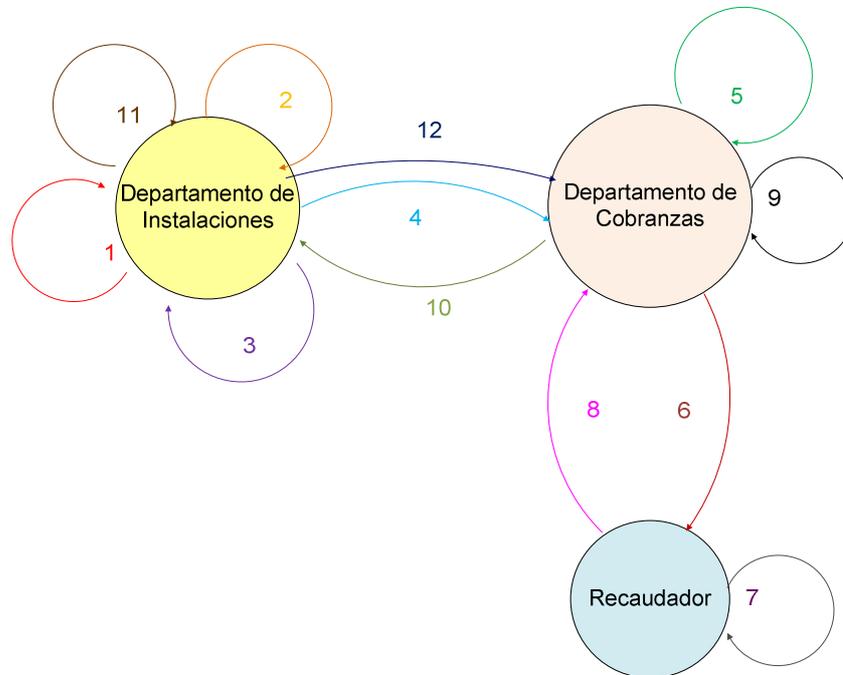


Figura 1.4: Flujo de trabajo con gestión de PDVs.

- 10)** El Departamento de Cobranzas hace llegar el pedido de gestión (reubicación o retiro de equipos, etc.) al Departamento de Instalaciones.
- 11)** El Departamento de Instalaciones se encarga nuevamente de hacer una orden de trabajo.
- 12)** El Departamento de Instalaciones manda la información al Departamento de Cobranzas, para luego seguir el proceso normal. Para gestiones se seguirán las fases 6 ,7 ,8 y 9 sucesivamente.

Por otro lado, si la gestión es un retiro de equipos, la fase 12 será la última que se realice a ese PDV, tomando en cuenta que al PDV ya inactivo se añadirá el nuevo estado “BASES”. Esto significa que los equipos que usaba este PDV, si se encuentran en buen estado, servirán para la instalación de un nuevo PDV.

1.2.3. DETERMINACIÓN DE PROBLEMAS.

1.2.3.1. Limitaciones del uso actual de las hojas de cálculo.

1.2.3.1.1. Validaciones.

Las celdas de la hoja de cálculo que emplean no poseen validaciones complejas para el ingreso de los datos, puesto que este tipo de validaciones requiere conocimientos avanzados. Se determinó que se puede presentar el caso de que datos tipo texto se convierten en datos numéricos, provocando pérdida de información, como en el caso de las cédulas o en el Registro Único de Contribuyentes (RUC), como se ilustra en la **Tabla 1.1**:

Dato ingresado	Dato convertido por la hoja de cálculo
0456321456	456321456
0456321456001	456321456001

Tabla 1.1: Ejemplo de conversión de texto en número.

Este hecho hace que se almacenen cédulas con menor longitud que la original. Por el mismo motivo de las validaciones, se pueden ingresar cédulas inválidas ya que la celda tampoco hace control del dígito verificador, que corresponde al décimo número para Ecuador.

A pesar de las limitantes encontradas, debido a la práctica que han adquirido los usuarios con el sistema basado en hojas de cálculo, la solución de software planteada contendrá reportes en hojas de cálculo para una rápida adaptación con los procesos que ya tienen establecidos.

1.2.3.1.2. Almacenamiento de la información.

Para poder aprovechar las funcionalidades de filtros e hipervínculos, toda la información debe ser almacenada en un solo libro de Microsoft Office Excel. La información que llega al Departamento de Cobranzas es mucho más frecuente que la del Departamento de Instalaciones, por el hecho de almacenar los reportes de recaudación. Esto genera que el archivo final sea de gran tamaño y su manipulación sea demorosa. Por este motivo, el Departamento de Cobranzas mantiene un archivo de reportes por cada mes, lo que limita la aplicación de filtros y retrasa la toma de decisiones.

1.2.3.2. Distribución de la información.

El Departamento de Cobranzas maneja la información en conjunto con la Gerencia General, por lo que el paso de información es imprescindible. Debido a que los datos se actualizan constantemente, en promedio doscientos reportes de recaudación diarios, la información más actualizada debe ser compartida cada vez que se requiera para una toma de decisiones. El intercambio de información se lo realiza a través de una carpeta compartida, copiando el archivo localmente a la computadora de Gerencia.

Este proceso involucra pérdida de tiempo y, además, requiere que ambas partes se hayan puesto de acuerdo previamente. La forma en la que se ponen de acuerdo es mediante una llamada telefónica, utilizando el número de extensión asignado por el Departamento Técnico dentro de la red telefónica interna.

Para obviar este proceso manual, adicionalmente, se instalará el software en Gerencia, brindándole accesos según el perfil requerido.

1.2.3.3. Modificación de la información.

1.2.3.3.1. Modificación de la información de forma no concurrente.

La información almacenada en las hojas de cálculo sólo puede ser accedida por una persona a la vez. Esta limitante se da debido a que el momento en que un usuario ha abierto el archivo, un siguiente usuario que requiera hacer modificaciones al mismo archivo sólo tendrá permisos de lectura sobre éste. El mismo efecto se presenta en la compartición de cualquier archivo dentro de una red Microsoft Windows, y no es exclusivo de las hojas de cálculo.

Las características del negocio, hacen que la información proveniente del Departamento de Instalaciones requiera ser actualizada ante un eventual cambio en algún Punto de Venta (PDV), y que esta información esté disponible en el Departamento de Cobranzas para la toma de decisiones.

Este requerimiento hace que sea necesario el acceso concurrente a la información, proponiendo que éste se realice en una base de datos centralizada.

1.2.3.3.2. Actualización de la información en un solo departamento.

Los cambios que se realiza en un departamento, generan información que debe actualizarse en el otro. Por ejemplo, un cambio de estado de un PDV realizado por el Departamento de Cobranzas, está ligado a una gestión que se realiza en el Departamento de Instalaciones.

La empresa ha optado por establecer personas responsables del manejo de cada hoja de cálculo, según la información que contenga, y esta información únicamente puede ser modificada por el departamento competente. Un cambio que se realice en el Departamento de Instalaciones requiere del paso manual e impreso de los datos actualizados al resto de departamentos. El canal que se

utiliza para pasar la información es susceptible a retardos grandes y extravíos que pueden ocasionar inconsistencias en la información si es que no se llega a actualizar en cada departamento.

Adicionalmente, cada PDV tiene una carpeta con su información actualizada, pero esta información no está en línea por la limitante informática antes mencionada. Incluso el departamento competente no dispone siempre de la información actualizada y se opta por utilizar información que puede ser obsoleta o por recurrir a una consulta manual a las carpetas.

El uso de una base de datos centralizada ya propuesto, adicionado al acceso a la información mediante perfiles de usuario, permitirá la modificación de la información manteniendo las competencias de cada departamento.

1.2.3.4. Toma de decisiones.

1.2.3.4.1. Información redundante en el momento de consultas.

Los datos almacenados conservan similitud en la forma en la que se almacenan. Por ejemplo, en el caso de instalaciones, varios PDVs comparten el mismo mayorista, como se ilustra en la **Tabla 1.2**:

Código del PDV	Nombre del PDV	Mayorista
004	CONFITERIA QUINTA ESENCIA	TECEMOVIL
005	FARMA AHORRO NO 30	CABICEL
006	IVONNE VIVERES II	CABICEL

Tabla 1.2: Ejemplo de redundancia en las instalaciones.

Una sola hoja de cálculo podría almacenar toda la información, pero presenta dos limitantes básicas:

- a. Para filtrar la información, dentro del mismo libro de Microsoft Office Excel, se requiere capacitación en el uso de filtros e hipervínculos.
- b. Información repetida en diferentes celdas genera redundancia excesiva.

El Departamento de Instalaciones utiliza una hoja de cálculo para cada orden de trabajo. Esto permite poder discriminar más rápidamente la información, pero degenera en la creación de excesivos archivos. Esta forma de manejar la información también limita las consultas, puesto que el único criterio de consulta es el nombre del archivo, ya que la información de los PDVs se encuentra dispersa en los diferentes archivos que se hayan creado.

Para superar estas limitantes, se propone la utilización de una base de datos que emplee un lenguaje declarativo, como SQL, el mismo que permitirá realizar consultas complejas que se adapten a los requerimientos del negocio. Además, el uso de procedimientos almacenados permitirá tener estas consultas ya pre-compiladas para un acceso más rápido a la información.

1.2.3.4.2. Obtención de Información útil para la toma de decisiones.

A pesar de poseer las facilidades que brindan los filtros de las hojas de cálculo, la obtención de la información requerida para toma de decisiones puede demorarse hasta veinticuatro horas debido a la gran cantidad de archivos de Microsoft Office Excel que se deben revisar. Aún cuando se pueden programar filtros más complejos, estos terminan siendo insuficientes ya que la información se halla dispersa en varios archivos.

Para solventar esta falencia, se crearán consultas dedicadas a la toma de decisiones, a modo de reportes manuales conservando sus formatos actuales en las hojas de cálculo.

1.2.3.5. Fallas de tipo humano.

Se determinaron dos tipos principales de errores frecuentes que se cometen con el sistema actual:

1.2.3.5.1. Fallas de apreciación.

En el momento de ingresar los datos, es indispensable que la columna y la fila en la que se ingresan correspondan al título de la columna y al PDV al que hace referencia la información. Un error en la ubicación de la celda genera inconsistencias en los datos para un PDV y puede alterar los datos de otro.

En este tipo de fallas también se pueden producir errores de digitación en datos redundantes. Se presentan los casos en que información igual es ingresada en mayúsculas y en otro caso en minúsculas, pareciendo finalmente información diferente. El usuario siempre debe tener presente el formato en el que está trabajando, para impedir conversión o truncamiento de datos.

El software propuesto eliminará los errores de formato, y minimizará los errores de apreciación poniendo información con etiquetas y ayudas visuales de forma que el proceso completo sea intuitivo para el usuario.

1.2.3.5.2. Fallas del canal de distribución de la información.

Por la forma impresa de distribuir la información, se puede producir un retardo excesivo, decayendo en mayor cantidad de tiempo para completar las tareas. Adicionalmente, la información puede extraviarse, requiriendo nuevamente el proceso completo de envío de la información.

1.2.3.6. Limitaciones del actual sistema de administración (SAR-II).

“SAR-II es un Sistema de Administración y Comunicación Remota que permite la gestión y telesupervisión del tarifador CT 8002 y otros equipos de la serie, actualmente en desarrollo.”¹

El SAR-II es un software provisto por DELSAT Group, para adaptarse a las necesidades generales de casi cualquier negocio que opere con sus equipos de cabinas telefónicas, no solamente para Ecuador sino también para otros países de América y Europa, entre los que se encuentran: Argentina, Bolivia, Chile, Colombia, Paraguay, Perú, Venezuela, Estados Unidos, México, Nicaragua, y España. El hecho de que sea tan general, y aún cuando tiene versiones dedicadas por país, ha impedido que requerimientos propios de cada negocio tengan que adaptarse con mecanismos adicionales, como son las hojas de cálculo en el caso de TELCOLSAM S.A.

El SAR-II no permite el ingreso de información detallada de los PDVs. Por ejemplo, sólo posee un campo de dirección para almacenar toda la ubicación exacta de la locación del local donde se van a instalar las cabinas, incluyendo las referencias. Existe la posibilidad de almacenar más información por campo, pero el espacio reservado en la base de datos para éste es muy reducido dada la necesidad real de la empresa. El SAR-II está más orientado a la administración de los equipos antes que al manejo del negocio.

Un mismo propietario puede ser dueño de varios PDVs, como es en el caso de sucursales de un mismo negocio. La información del SAR-II asume que es única la relación entre un propietario y un PDV. Esta limitante también genera redundancia innecesaria de la información.

Adicional a las limitaciones presentadas en el Departamento de Instalaciones, cabe destacar que el SAR-II no posee un módulo para la administración de cobranzas, lo que hace que el proceso se lo haga enteramente en hojas de cálculo, duplicando toda la información existente en la base de datos del SAR-II.

¹ Fuente: <http://www.delsatgroup.com/index.php?option=tarifadores&id=ct8002>

1.3. ESPECIFICACIÓN DE REQUERIMIENTOS DEL SOFTWARE.

Para la determinación de los requerimientos de software, SRS por sus siglas en inglés, se seguirá el estándar *IEEE Std 830-1998: "IEEE Recommended Practice for Software Requirements Specifications"* (Prácticas Recomendadas para las Especificaciones de Requerimientos de Software), publicado el 25 de junio de 1998, que es una revisión del *IEEE Std 830-1993*, actualmente en estado archivado y reemplazado por el más reciente.

Las definiciones que no se establezcan aquí, serán tomadas de las que proporciona el estándar *IEEE Std 610.12-1990: "IEEE standard glossary of software engineering terminology"* (Glosario estándar de terminología de ingeniería de software), publicado el 10 de diciembre de 1990, que se encuentra en estado activo.

1.3.1. INTRODUCCIÓN (SECCIÓN 1 DE LA SRS).

1.3.1.1. Propósito.

El propósito de esta SRS es describir detalladamente los requerimientos funcionales y no funcionales de la primera versión del Sistema de Software para la Gestión de Cobranzas, para la empresa TELCOLSAM S.A.

Esta SRS especifica las características del software, por lo tanto va dirigido para los desarrolladores del software, por un lado, y para validación por la parte interesada², por el otro lado.

² Cualquier persona o entidad que es afectada por las actividades derivadas del uso del software.

1.3.1.2. Alcance.

El Sistema de Software para la Gestión de Cobranzas, denominado "POINTSOF", permitirá la administración de la información proveniente de los Puntos de Venta, orientada a la toma de decisiones en el Departamento de Cobranzas. POINTSOFT permitirá el manejo de la información, por parte del Departamento de Instalaciones y el Departamento de Cobranzas, proveniente de los Puntos de Venta. Para este propósito, se dividirá el sistema de software en módulos que operarán en tres capas. El acceso a los módulos se realizará mediante perfiles de usuario.

POINTSOF contemplará para su desarrollo, principalmente, los siguientes Objetivos del Negocio (ON), Criterios de Éxito (CE) y Riesgos del Negocio (RN):

ON-1: Reducir el tiempo para la toma de decisiones; en los próximos 6 meses desde la implementación en TELCOLSAM S.A. de la primera versión del software.

Métrica: Intervalo de tiempo entre la petición de información por parte de Gerencia al Departamento de Cobranzas, hasta que la información es enviada a Gerencia.

ON-2: Incrementar el grado de satisfacción de los empleados de cada departamento donde se implemente el software; en los próximos 6 meses desde la implementación en TELCOLSAM S.A. de la primera versión del software.

Métrica: Encuesta directa a los empleados de cada área.

CE-1: Tener el 75% de los empleados, que trabajan actualmente en cada departamento donde se implemente el software, utilizando apropiadamente el sistema según la capacitación que se brinde; en los próximos 3 meses desde la implementación en TELCOLSAM S.A. de la primera versión del software.

Métrica: Número de usuarios, por departamento, que tengan instalado el software en sus terminales, o que utilicen el software en terminales compartidas.

RN-1: La implementación del software puede generar nuevos roles que tengan que ser asumidos por el usuario.

Probabilidad = 0.25; Impacto = 2/10.

RN-2: Algunos empleados pueden no adaptarse al nuevo sistema dentro del tiempo especificado en los Objetivos del Negocio.

Probabilidad = 0.25; Impacto = 8/10.

RN-3: Algunos empleados pueden no utilizar el sistema y volver al sistema anterior basado en hojas de cálculo, debido a la familiaridad de 4 años que tienen con ese sistema.

Probabilidad = 0.15; Impacto = 9/10.

NOTA: Los datos de “probabilidad” e “impacto” se obtuvieron directamente TELCOLSAM S.A. en entrevista directa a la Gerencia.

POINTSOFTE incluirá las siguientes Características Principales (CP):

CP-1: Aplicación de escritorio, basado en pestañas.

CP-2: Programación modular, que permita escalabilidad, dividido en tres capas.

CP-3: Acceso a los módulos que se habilitarán mediante perfiles asociados a cuentas de usuario, cada cuenta con un nombre de usuario y contraseña.

CP-4: Crear, ver, modificar, y borrar información en los formularios, según las necesidades obtenidas de las reglas del negocio, a excepción del formulario de “reportes”, donde únicamente se podrá ver.

CP-5: Generar reportes a “petición del usuario”, y permitir exportarlos a hojas de cálculo.

CP-6: Proveer acceso a través de la Intranet corporativa.

1.3.1.3. Definiciones, siglas, y abreviaciones.

- **Siglas y abreviaciones.**

- a) **DAL:** Data Access Layer (Capa de Acceso a Datos)
- b) **Excel:** Microsoft® Office Excel.
- c) **GUI:** Graphic User Interface (Interfaz Gráfica de Usuario).
- d) **IVA:** Impuesto al Valor Agregado.
- e) **MDI:** Multiple Document Interface (Interfaz de Múltiples Documentos).
- f) **OMG:** Object Management Group.
- g) **PDV:** Punto de Venta.
- h) **RUC:** Registro Único de Contribuyentes.
- i) **SAR:** Sistema de Administración Remota.
- j) **SDI:** Single Document Interface (Interfaz de Documento Único).
- k) **SO:** Sistema Operativo.
- l) **SRS:** Software Requirements Specifications (Especificaciones de Requerimientos de Software).
- m) **TBD:** To Be Defined (Ha ser definido).
- n) **UML:** Unified Modeling Language (Lenguaje Unificado de Modelado).
- o) **WAN:** Wide Area Network (Red de área extendida).
- p) **Windows:** Sistema Operativo Microsoft® Windows.
- q) **WPF:** Windows Presentation Foundation.

- **Definiciones.**

- a) **Bases:** Son los equipos que han sido retirados y sirven como equipos en otro nuevo Punto de Venta. Se entiende que un Punto de Venta es bases de otro cuando sus equipos son utilizados para la instalación en otro Punto de Venta.
- b) **Bi-cabinas:** Es la unión de dos cabinas, que se instalan conjuntamente.
- c) **Cabina:** Es una pequeña estructura que en su interior posee un teléfono.
- d) **Cliente:** Es el propietario del Punto de Venta.
- e) **Estado:** Determina la fase en que está en ese momento el Punto de Venta.
- f) **Factor de Valor:** Se refiere a un “impuesto” o una “comisión”. Impuesto, si está anclado a todos los Puntos de Venta y se aplica al valor del subtotal del reporte de recaudación; y, comisión, si está anclado a cada Punto de Venta, y se aplica al valor del total del reporte de recaudación.
- g) **Framework:** Es una capa intermedia, entre el Sistema Operativo y la aplicación de software, cuya finalidad es dar soporte a la aplicación para su desarrollo e implementación. Puede contener bibliotecas y un lenguaje interpretado.
- h) **Frecuencia:** O “Periodo”, es el tiempo entre visitas de un recaudador a un Punto de Venta.
- i) **Gestión:** Son operaciones que se realizan a un Punto de Venta.
- j) **Localización:** Consta de país, provincia, ciudad, barrio, sector, dirección y referencia.
- k) **Mayorista:** Es el proveedor de equipos para TELCOLSAM S.A.
- l) **Ofrecimiento de pago:** Ocurre cuando un cliente no tiene el dinero que ha facturado el tarifador, y se compromete a saldar un valor determinado en un día específico.

- m) Pestaña:** O “Tab”, es un elemento muestra el mismo contenido que se puede alojar en una ventana, pero sin necesidad de cambiar de ventana.
- n) Punto de Venta:** Es un negocio previamente constituido donde se hace la instalación de una o más cabinas. Pueden ser tiendas, centros comerciales, víveres, entre otros.
- o) Tarifador:** Es el equipo que permite controlar el gasto por acceso telefónico usado en Puntos de Venta.
- p) Tipo de Instalación:** Es el proceso que se va a realizar al PDV en la instalación.
- q) Recaudador:** Es la persona, que trabaja en TELCOLSAM S.A., encargada de recaudar el dinero facturado por el tarifador.
- r) Reporte de Recaudación o Ticket:** Son los reportes impresos que el cliente entrega al recaudador, proporcionados por el tarifador, comprobando la cantidad de dinero que el cliente debe entregar. Cada reporte tiene una numeración única.
- s) Retiro:** Significa la clausura total del contrato y devolución de los equipos a TELCOLSAM S.A.
- t) Ruta:** Comprende los días de la semana unido a una frecuencia en que un recaudador debe visitar los Puntos de Venta asignados.

1.3.1.4. Referencias.

Referirse al Título “**BIBLIOGRAFÍA**”, de la página 232.

1.3.1.5. **Apreciación global.**

La siguiente parte de la SRS está organizado de la siguiente manera:

- La Sección 2 ofrece una descripción global del software. Contiene una descripción de los factores que influyen en la utilización y desarrollo del software, y restricciones que el mismo presenta, únicamente mirando su contexto.
- La Sección 3, desarrollada en el **Título “Diagrama de Casos de Uso”, página 126**, detallará los requerimientos específicos, desde el punto de vista del diseño, que el software poseerá. Se empleará la metodología “RUP” organizada por la clase del usuario.

1.3.2. **DESCRIPCIÓN GLOBAL (SECCIÓN 2 DE LA SRS).**

1.3.2.1. **Perspectiva del producto.**

POINTSOF es un sistema de gestión de PDVs que reemplaza el proceso manual de gestión mediante el uso de hojas de cálculo, orientado principalmente a la gestión de cobranzas. Tendrá una interfaz amigable al usuario, de fácil aprendizaje, y que proporcionará mecanismos para la culminación rápida y correcta de los procesos.

POINTSOF será un producto independiente y autónomo que trabajará en plataformas basadas en Windows. No tiene interrelación con otras soluciones de software existentes en TELCOLSAM S.A. Únicamente tendrá relación con Excel para la exportación de reportes. Esto permitirá realizar esta funcionalidad de exportación de reportes sin requerir que Excel esté instalado localmente en la terminal, o remotamente en un servidor.

Para definir los componentes del sistema, se seguirá UML 2.2 sobre los Casos de Uso, como está definida en el Documento OMG Número: **formal/2009-02-02**³. El diagrama de la **Figura 1.5**, muestra la interoperabilidad entre los diferentes actores del sistema, con un diagrama de Casos de Uso de Alto Nivel⁴:

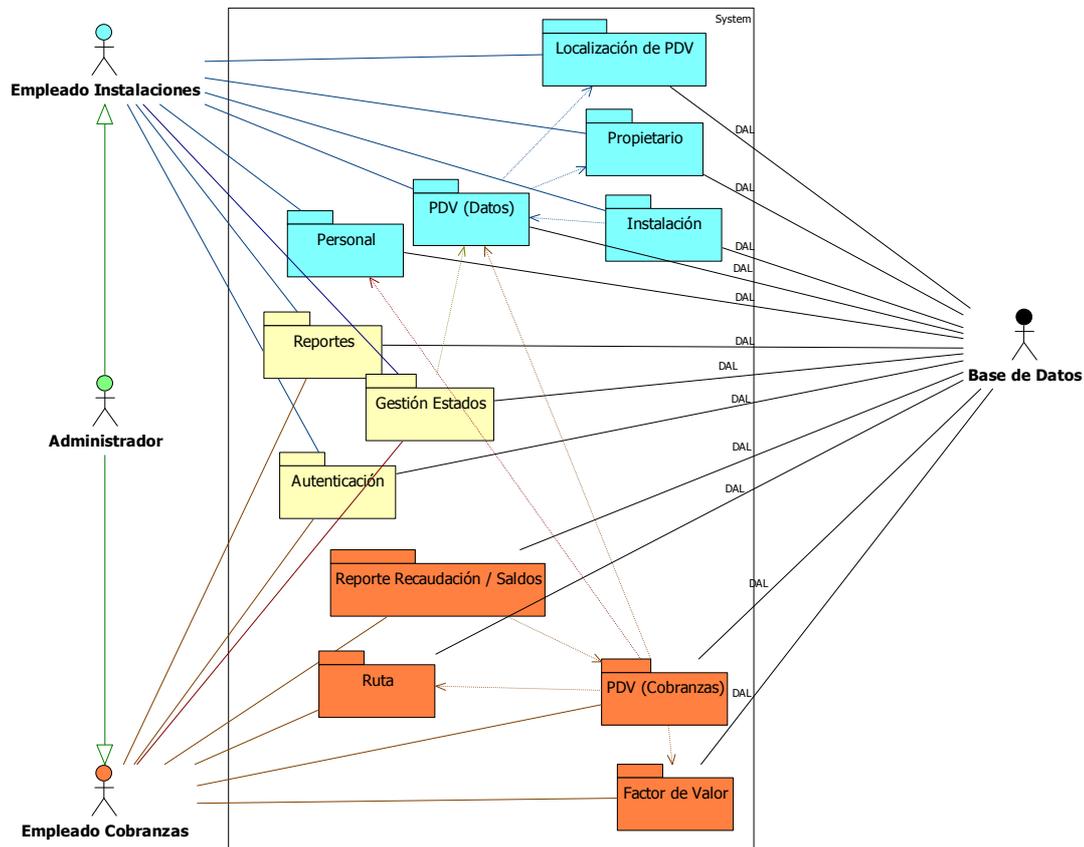


Figura 1.5: Diagrama de Alto Nivel de Casos de Uso de POINTSOFT.

El diagrama está diseñado con paquetes, representados por rectángulos en forma de carpeta. Dentro de cada paquete se encuentran los casos de uso agrupados según su similitud y funcionalidad. Un paquete representa un módulo a implementar, y se separan por colores según los perfiles de usuario o los roles. Existen paquetes que pueden ser accedidos por varios actores, como es el caso de Autenticación y Reporte. Todos los paquetes tienen comunicación con la “Base

³ “OMG Unified Modeling Language™ (OMG UML), Superstructure” *Version 2.2*.

⁴ High-Level Uses-Case.

de Datos” a través de su DAL. La descripción de las funciones se las hará en el **Título “Funciones del producto.”, página 33.**

Cabe destacar que el actor “Administrador” es una generalización de “Empleado Instalaciones” y “Empleado Cobranzas”, representado por las flechas que apuntan hacia esos dos actores. Esto significa que el actor “Administrador” puede acceder a los paquetes de ambos actores.

1.3.2.1.1. Interfaces del sistema.

POINTSOFTE contendrá tres interfaces para la interacción con los diferentes módulos, correspondientes a cada una de las tres capas en las que se programará:

- a) Capa de Presentación: Una GUI para la gestión de la información, basada en pestañas.
- b) Capa de Aplicación: Una librería que contenga las clases e implemente reglas del negocio.
- c) Capa de Datos: Una librería que contenga la DAL para la comunicación entre la capa de aplicación y la base de datos.

POINTSOFTE no contempla la unificación con otros sistemas y, por lo tanto, no provee interfaces para que sistemas de terceros se acoplen de manera alguna.

1.3.2.1.2. Interfaz con el usuario.

El usuario accederá a la aplicación por medio de una GUI, que contendrá una ventana principal y varias pestañas secundarias. Se desarrollará un GUI de escritorio que sea intuitiva, amigable, y con acceso rápido a las funciones más comunes; para obtener el máximo rendimiento y reduciendo el tiempo necesario para la realización de las tareas.

1.3.2.1.3. Interfaces con el hardware.

POINTSOFTE tendrá las siguientes interfaces con el hardware:

- **De entrada:** Ratón y Teclado.
- **De salida:** Monitor.

Todas las interfaces estarán siempre activas para el funcionamiento del software, y requerirá de todas para su funcionamiento.

1.3.2.1.4. Interfaces con el software.

Para realizar la interconexión con el Sistema Operativo, la base de datos, y Excel; se emplearán bibliotecas de terceros. En el **Título “Otras herramientas a utilizar”, página 108**, se determinará el software más adecuado y las librerías necesarias dependiendo del software seleccionado, según los siguientes criterios:

TBD-1: Para el Sistema Operativo.- Un Framework que sirva de soporte a la aplicación a desarrollarse, procurando hacer el software independiente de la versión del SO.

TBD-2: Para la Base de Datos.- Un Conector (de ser necesario), para la comunicación entre la aplicación y la base de datos.

TBD-3: Para los Reportes.- Una Librería, para la exportación de los reportes a Excel.

1.3.2.1.5. Interfaces de comunicaciones.

La red de área local existente en TELCOLSAM S.A. trabaja con el estándar IEEE 802.3u-1995⁵, que permite que POINTSOFT trabaje con la familia de protocolos TCP/IP. La conexión a la base de datos se hará con un criterio orientado a la

⁵ IEEE Standards for Local and Metropolitan Area Networks: Supplement to Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications Media Access Control (MAC) Parameters, Physical Layer, Medium Attachment Units, and Repeater for 100 Mb/s Operation, Type 100BASE-T (Clauses 21-30).

conexión. Para más información sobre la red local, referirse al **Título “INFRAESTRUCTURA DE LA RED.”**, página 42.

1.3.2.1.6. Restricciones de memoria.

Los recursos de memoria serán manejados por el Framework que se escoja, según las especificaciones del mismo.

1.3.2.1.7. Requisitos de adaptación del Sitio.

Todas las terminales en las que se instale el software deberán soportar la instalación del Framework y poseer Excel instalado para permitir la modificación manual de los reportes.

1.3.2.2. Funciones del producto.

A continuación se describen globalmente los funcionamientos de los casos de uso especificados en la **Figura 1.5, página 30**, agrupados en paquetes:

- **Autenticación.**

Paquete	Descripción	Funciones
Autenticación	Verificación de credenciales y permisos por perfiles.	Ingresar credenciales.

Tabla 1.3: Módulos para la gestión de localizaciones.

- **Localización de PDV.**

Paquete	Descripción	Funciones
País	País de localización del PDV.	Ver, agregar, modificar y eliminar países.
Provincia	Provincia de localización del PDV.	Ver, agregar, modificar y eliminar provincias, dentro de un país.
Ciudad	Ciudad de localización del PDV.	Ver, agregar, modificar y eliminar ciudades, dentro de una provincia.
Sector	Sector de localización del PDV.	Ver, agregar, modificar y eliminar sectores.
Barrio	Barrio de localización del PDV.	Ver, agregar, modificar y eliminar, dentro de una ciudad, en un sector.

Tabla 1.4: Módulos para la gestión de localizaciones.

- **Propietarios.**

Paquete	Descripción	Funciones
Propietario	Información del dueño del local comercial donde funciona el PDV.	Ver, agregar, modificar y eliminar propietarios; buscar Propietarios.

Tabla 1.5: Módulos para la gestión de propietarios.

- **PDV (Datos).**

Paquete	Descripción	Funciones
Punto de Venta	Información general acerca del PDV.	Ver, agregar, modificar y eliminar PDVs; buscar PDVs; buscar Propietarios.

Tabla 1.6: Módulos para la gestión de PDVs.

- **Estados.**

Paquete	Descripción	Funciones
Gestión de Estados	Información sobre los Estados de un PDV.	Ver y modificar estados de un PDV.

Tabla 1.7: Módulos para la gestión de PDVs.

- **Instalación.**

Paquete	Descripción	Funciones
Mayorista	Información de los mayoristas.	Ver, agregar, modificar y eliminar mayoristas.
Tipo de Instalación	Información sobre los tipos de instalaciones.	Ver, agregar, modificar y eliminar tipos de instalaciones.
Instalación	Administración de las instalaciones.	Ver, agregar, modificar y eliminar instalaciones a un PDV; buscar PDVs.

Tabla 1.8: Módulos para la gestión de instalaciones.

- **Factores de Valor.**

Paquete	Descripción	Funciones
Factor de valor	Gestión de factores de multiplicación a una cantidad monetaria.	Ver, agregar, modificar y eliminar factores de valor.

Tabla 1.9: Módulos para la gestión de factores de valor.

- **Ruta.**

Paquete	Descripción	Funciones
Período	Gestión de periodos de recaudación.	Ver, agregar, modificar y eliminar periodos.
Ruta	Gestión de rutas de recaudación.	Ver, agregar, modificar y eliminar rutas.

Tabla 1.10: Módulos para la gestión de rutas.

- **PDV (Cobranzas).**

Paquete	Descripción	Funciones
Comisión por PDV	Gestión de comisiones por PDV.	Ver, modificar comisiones a un PDV; buscar PDVs.
Ruta y Recaudador por PDV	Gestión de rutas y recaudadores por PDV.	Ver, modificar la ruta y el recaudador de un PDV; buscar PDVs.

Tabla 1.11: Módulos para la gestión de cobranzas de PDV.

- **Reportes de Recaudación / Saldos.**

Paquete	Descripción	Funciones
Reporte de Recaudación	Gestión de reportes de recaudación.	Ver, agregar, modificar, y eliminar reportes de recaudación a un PDV; buscar PDVs.
Ofrecimiento de pago	Gestión de ofrecimientos de pago para PDV con cartera vencida.	Ver, agregar, modificar, y eliminar ofrecimientos de pago a un PDV; buscar PDVs.

Tabla 1.12: Módulos para la gestión de reportes de recaudación y saldos.

- **Personal.**

Paquete	Descripción	Funciones
Personal	Gestión de empleados de TELCOLSAM S.A. y cuentas de usuario.	Ver, agregar, modificar, y eliminar empleados; buscar empleados; asociar un perfil a un empleado.

Tabla 1.13: Módulos para la gestión de personal

- **Reportes.**

Paquete	Descripción	Funciones
Reportes	Generación de reportes manuales y exportación a Microsoft Office Excel.	Generar reportes según criterios y parámetros.

Tabla 1.14: Módulos para la generación de reportes.

Los reportes cumplen las siguientes especificaciones:

- **Sobre los Puntos de Venta.**

- Detalle de Todos los Puntos de Venta.- Información de todos PDVs incluida la información de Cobranzas.
- Historial de Estados de un Punto de Venta.- Historial de estados por los que ha pasado un PDV.
- Historial de Instalaciones de un Punto de Venta.- Historial de instalaciones terminadas que se le han hecho a un PDV.
- Historial de Cobranzas de un Punto de Venta.- Historial de ofrecimientos de pago y gestiones de cobranzas que se ha hecho a un PDV.

- **Sobre los Estados.**

- e. Instalación Activo sin Recaudador.- PDVs cuya instalación ha sido completada pero no se le ha asignado un recaudador.
- f. Activos con Recaudador.- PDVs en estado activo y que se les haya asignado un recaudador.
- g. Bloqueados.- PDVs en estado bloqueado.
- h. Inactivos.- PDVs en estado inactivo.
- i. Bases.- PDVs en estado bases.

- **Sobre los Reportes de Recaudación.**

- j. Por fechas y Punto de Venta.- Reportes de recaudación dado un rango de fechas y un PDV.
- k. Por fechas y Todos los Puntos de Venta.- Reportes de recaudación dado un rango de fechas para todos los PDVs.

- **Sobre los Recaudadores.**

- l. Puntos de Venta por Recaudador.- Puntos de Venta a los que está asignado un recaudador.

- **Sobre las Instalaciones.**

- m. Instalaciones pendientes.- Instalaciones que se han programado, y aún no se han completado. Equivale a la orden de trabajo.
- n. Historial de Instalaciones por Fechas.- Instalaciones completadas para todos los PDVs, dado un rango de fechas.

1.3.2.3. Características del usuario.

Los usuarios de POINTSOFT deben estar familiarizados con el uso de las interfaces de hardware del sistema especificados en el **Título “Interfaces con el hardware.”, página 32**. Adicionalmente, deben tener experiencia de al menos un mes en el uso de Windows y las interacciones que este SO puede ofrecer por medio de su sistema de ventanas.

No se requerirá conocimiento de bases de datos, ni lenguajes de programación. El conocimiento de Excel básico es opcional.

1.3.2.4. Restricciones.

POINTSOFT presentará las siguientes Limitaciones (LI):

- LI-1:** No incluirá alarmas automáticas, ni reportes automáticos. Todos serán por eventos lanzados por el usuario.
- LI-2:** Las aplicaciones de escritorio únicamente se comunicarán con el servidor centralizado, y a través de la red local. No contempla interfaces para comunicarse con otras aplicaciones existentes.
- LI-3:** No contemplará la interacción con cualquier otro sistema de software presente en TELCOLSAM S.A.
- LI-4:** POINTSOFT se apegará a las licencias pertinentes, en caso de utilizar software de terceros.
- LI-5:** POINTSOFT soportará hasta diez terminales simultáneas, sin degradar el servicio.
- LI-6:** Para auditoría de software solamente se proporcionará los reportes manuales. No se creará un log para la auditoría del software.
- LI-7:** POINTSOFT únicamente presentará una GUI en español.
- LI-8:** POINTSOFT funcionará solo en redes que soporten TCP/IP.

LI-9: No se encriptarán los datos para su transmisión a través de la Intranet.

LI-10: El tiempo de respuesta será de máximo 5 segundos para la carga de 300 registros.

LI-11: El proyecto POINTSOFT culminará después de la implementación en TELCOLSAM S.A. de la primera versión del software y, por lo tanto, no contempla la verificación de los ON y CR.

1.3.2.5. Atención y dependencias.

Se consideran las siguientes Dependencias (DE):

DE-1: Los equipos de red, servidores y terminales deben mantenerse hasta la implementación del sistema.

DE-2: El SO y su versión no puede variar después del análisis de requerimientos.

CAPITULO 2



2. CARACTERIZACIÓN DE HARDWARE Y SOFTWARE.

2.1. SITUACIÓN ACTUAL E INSTALACIONES.



TELCOLSAM S.A. trabaja conjuntamente con otra empresa llamada TELINEMAX S.A., ambas se encuentran registradas en Ecuador, y ubicadas en la misma oficina:

UBICACIÓN: Quito Avenida Colón y Rábida.

EDIFICIO: Conjunto Ave María, 2do piso, Oficina 201.

Estas dos empresas comparten una misma estructura física; la oficina tiene subdivisiones para cada departamento y un sitio de trabajo para cada integrante de los mismos. Además, se encuentran ligados lógicamente, lo que significa que trabajan unidos al mismo cableado estructurado, a la misma red, y a los mismos servicios.

TELCOLSAM S.A. contrata a TELINEMAX S.A. para poder dar soporte a sus clientes. TELINEMAX S.A. es una empresa de prestación de servicios y productos en telecomunicaciones, informática, servicios neumáticos y otros; “se dedica a proporcionar a sus clientes asesoría y soluciones tecnológicas que se adapten a sus necesidades específicas, convirtiéndolos en un Socio Estratégico, al ser parte del desarrollo activo de su empresa.”⁶

⁶ FUENTE: <http://telinemax.com.ec/empresa/empresa.htm>

TELINEMAX S.A. brinda soporte técnico a empresas multinacionales tales como Telefónica Movistar y Alegro PCS, siendo estas su mejor carta de presentación al ofrecerles tarificación de telefonía e Internet respectivamente.

Para la realización de este proyecto fue necesaria la entrevista al Gerente Técnico de la empresa, puesto que él facilitó el ingreso a las instalaciones y brindó la accesibilidad a los dispositivos para poder hacer la toma de datos mediante software.

2.2. INFRAESTRUCTURA DE LA RED.

TELCOLSAM S.A. y TELINEMAX S.A. en conjunto tienen un total de 24 puntos de trabajo. Cada punto de trabajo consta básicamente de:

- Un CPU (Unidad Central de Procesamiento).
- Un monitor.
- Dispositivos de entrada: teclado y ratón.

Algunos puestos de trabajo constan también de dispositivos de salida como parlantes, y comparten impresoras a través de la red local.

2.2.1. EQUIPAMIENTO.

Primeramente, se realizará una lista de equipos en la empresa TELCOLSAM S.A. detallando su hardware en forma tabulada.

2.2.1.1. Cuarto de Equipos.



Figura 2.1: Cuarto de equipos en TELCOLSAM S.A.

2.2.1.1.1. Servidores y hardware de apoyo.

Por la importancia de función de los equipos mencionados, la información fue obtenida de manera manual, o sea, no fue posible instalar ninguna clase de aplicación que ayude con la recopilación de información de hardware.

SERVIDORES Y HARDWARE DE APOYO			
NOMBRE	MODELO Y CARACTERÍSTICAS FÍSICAS	FUNCIONES CUMPLIR	A
 <p>SERVIDOR: Zeus</p>	<p>SISTEMA: Microsoft Windows Server 2003 R2 Enterprise Edition Service Pack 2 Registrado a nombre de TELINEMAX S.A.</p> <p>EQUIPO: Intel (R) Pentium (R) D CPU</p> <p>Procesador: Intel Pentium 3.0 GHz Intel Pentium 3.0 GHz</p> <p>RAM: 2.99 GB, 1,99 GB</p> <p>Unidades: DVD/CD-ROM</p>	<p>Su principal función es la de recolectar información llegada por un puerto. Éstas son llamadas que entran a un Administrador Remoto con el fin de obtener tele-supervisiones de los equipos.</p> <p>Dentro de su sistema se encuentra instalado el SAR II, más su base de datos Firebird 2.0.0.</p> <p>Este servidor es usado</p>	

Continúa en la página 44.

	<p>Puertos: 2 puertos USB</p> <p>Dominio: DATADOSMIL.local Zeus.DATADOSMIL.local</p>	<p>mayoritariamente por TELINEMAX S.A. para supervisar los equipos de Clientes de TELCOLSAM S.A.</p>
 <p>SERVIDOR: Respaldos</p>	<p>SISTEMA: Microsoft Windows 2000 Service Pack 4 Registrado a nombre de Telinemax S.A.</p> <p>EQUIPO: Procesador: Intel (R) Pentium 2.40 GHz RAM: 522032 KB Unidades: DVD/CD-ROM Puertos: 2 puertos USB Dominio: DATADOSMIL.local Respaldos.DATADOSMIL.local</p>	<p>Su función radica en respaldar dinámicamente la información importante de cada usuario ubicada en la red.</p> <p>Todos los usuarios de TELINEMAX S.A. y TELCOLSAM S.A. tienen información respaldada por cada mes.</p>
 <p>SERVIDOR: Fenix</p>	<p>SISTEMA: Microsoft Windows 2000 Service Pack 4 Registrado a nombre de Telcolsam S.A.</p> <p>EQUIPO: Procesador: Intel (R) Pentium 2.26 GHz RAM: 490992 KB Unidades: DVD/CD-ROM Puertos: 2 puertos USB Dominio: Telcolsam.com Fenix.Telcolsam.com</p>	<p>Tiene un sistema contable denominado Latinium, que es usado por TELCOLSAM S.A. para controlar su estado económico.</p> <p>Éste es el único servidor con diferente dominio ya que TELCOLSAM S.A. anteriormente trabajaba separado de TELINEMAX S.A.</p> <p>Es el servidor de dominio para TELCOLSAM S.A.</p>

Continúa en la página 45.

 <p>SERVIDOR: Linux</p>	<p><u>SISTEMA:</u> Centos 4.0 Registrado a nombre de Telcolsam S.A.</p> <p><u>EQUIPO:</u> Procesador: Intel (R) Pentium 2.26 GHz RAM: 512 MB Unidades: DVD/CD-ROM Puertos: 4 puertos USB Dominio: Telcolsam.com Fenix.Telcolsam.com</p>	<p>Es el firewall de la Empresa; por tanto, se tiene un control detallado de navegación, con perfiles y prohibiciones. Además, es el servidor de correo de la Empresa.</p>
 <p>SERVIDOR: Server</p>	<p><u>SISTEMA:</u> Microsoft Windows 2000 Service Pack 4 Registrado a nombre de Datadosmil S.A.</p> <p><u>EQUIPO:</u> Procesador: Intel (R) Xeon (TM) 2.40 GHz RAM: 1048084 KB Unidades: 4, DVD/CD-ROM. Puertos: 4 puertos USB Dominio: DATADOSMIL.local server.DATADOSMIL.local</p>	<p>Encargado de ser el servidor de archivos, tiene un programa contable llamado Spyral, fabricado por Carrasco⁷, usado por TELINEMAX S.A. y DATADOSMIL S.A</p>  <p>Es el servidor DHCP (Protocolo Configuración Dinámica de Anfitrión)</p>

Continúa en la página 46.

⁷ Empresa expertos en comunicación creativa, diseñadores de Software.

 <p style="text-align: center;">MONITOR</p>	<p><u>MONITOR:</u> Adaptador gráfico VGA estándar.</p> <p>Marca: Apex.</p> <p>Tamaño: 19 pulgadas.</p> <p><u>TECLADO:</u> Marca: Genius.</p> <p>Descripción: Teclado en Español.</p> <p><u>RATÓN:</u> Marca: Mega.</p> <p>Descripción: Ratón de Bola.</p>	<p>Todos los servidores son manejados a través de un solo monitor, un teclado y un ratón.</p>
 <p style="text-align: center;">CONMUTADOR</p>	<p><u>DESCRIPCIÓN:</u> D-Link DFM 560EL external.</p> <p>Versión: 9.3.3.</p> <p>Sistemas Operativos soportados: Windows XP, Windows NT SP1, Windows ME, Windows NT SP2, Windows 2000 SP3, Windows 98.</p> <p>NOTA: Con una correcta configuración adicional, el conmutador funciona con Sistemas Operativos Linux.</p>	<p>A través de este conmutador se realiza la selección del servidor con el que se va a trabajar. Conecta cualquiera de los servidores a un único monitor, teclado y ratón.</p>

Continúa en la página 47.

	<p>DESCRIPCIÓN:</p> <p>Fabricante: Tripp.lite (Power Protection) Smart Online.</p> <p>Voltajes: 100[V], 110[V], y 120[V].</p> <p>Forma de Instalación: Tipo torre⁸.</p>	<p>Es una fuente de suministro eléctrico que posee una batería con el fin de seguir dando energía a un dispositivo en el caso de interrupción eléctrica</p>
<p>UPS (Sistema de alimentación ininterrumpida)</p>		

Tabla 2.1: Detalle de los servidores y hardware de apoyo de TELCOLSAM S.A.

2.2.1.1.2. *Equipos de interconectividad.*



Figura 2.2: Rack de equipos de interconectividad.

⁸ FUENTE: <http://www.tripplite.com/es/products/product-series.cfm?txtSeriesID=661&EID=33531>

EQUIPOS DE INTERCONECTIVIDAD	
NOMBRE	MODELO Y CARACTERÍSTICAS FÍSICAS
 <p style="text-align: center;">Router</p>	<p>Fabricante: Punto Net.</p> <p>Modelo: ZYXEL P-334 BROADBAND FIREW ROUTER.</p> <p>Descripción: 10/100BASE-TX/RJ-45. Full/ Half dúplex. 4 Puertos LAN. 1 Puerto WAN ADSL.</p>
 <p style="text-align: center;">Switch</p>	<p>Fabricante: 3COM.</p> <p>Modelo: 3C16792A OfficeConnect DualSpeed Switch16.</p> <p>Descripción: 10/100BASE-TX/RJ-45. Store-and-forward. Autonegociación Full/Half dúplex 16 puertos con auto-detección de velocidad</p>
 <p style="text-align: center;">Switch</p>	<p>Fabricante: 3COM.</p> <p>Modelo: 3C16794 OfficeConnect DualSpeed Switch8.</p> <p>Descripción: 8 autosensing 10/100 Ethernet 10/100BASE-TX, RJ-45. Store-and-forward. Autonegociación Full/Half duplex</p>

Continúa en la página 49.

 <p style="text-align: center;">Switch</p>	<p>Fabricante: Advatek Networks. Modelo: ANS-16P 16-Port 10/100 Ethernet Switch. Descripción: Arquitectura Store-and-Forward IEEE 802.3, 802.3u (Ethernet, FastEthernet). Autonegociación.</p>
 <p style="text-align: center;">Radio LAN / PC</p>	<p>Fabricante: Punto Net Sistemas Operativos: Windows 95 / 98, Windows XP/Vista, Linux Red Hat/SuSe/Ubuntu, Mac OS. Descripción: Interfaz de Radio para interconexión con Antena.</p>

Tabla 2.2: Detalle de equipos de interconectividad.

2.2.1.2. Terminales de usuarios por departamentos.

2.2.1.2.1. Departamento de Instalaciones.

Para la recolección de información en este departamento se usó la versión de prueba **EVEREST Ultimate Edition versión 5.3**, de Lavalys. “EVEREST realiza un extenso y detallado análisis del sistema, mostrando prácticamente todos los aspectos referentes a hardware, software, configuración de red y más”.⁹ El programa fue ejecutado el 2009-10-08 a las 11:00, obteniendo los siguientes resultados:

⁹ FUENTE: <http://www.lavalys.com/products.php?ps=UE&lang=en&page=9>

DEPARTAMENTO DE INSTALACIONES	
NOMBRE	MODELO Y CARACTERÍSTICAS FÍSICAS
 	<p><u>Computador Recepción/Instalaciones:</u></p> <p>Nombre: RECEPCIÓN (Celia María)</p> <p>MAC: 00-04-75-8A-E5-BE.</p> <p>Sistema operativo: Windows XP Professional.</p> <p>CPU: Intel Premium 4 CPU 2.00Ghz.</p> <p>Video: NVIDIA RIVA TNT2 Model 64/Model 64 Pro (Microsoft Corporation) (32 MB).</p> <p>Impresora: hp LaserJet 3380 PCL 6.</p> <p>Nombre DNS: recepcion.DATADOSMIL.local.</p> <p>Procesadores: Intel Corporation Pentium (R) 4 128MB, 400Mhz.</p> <p>Caché: 512 KB SRAM</p> <p>RAM: 256 MB.</p> <p>Puertos de Comunicación: COM1, COM2, LPT1, DOT4, COM5.</p>

Tabla 2.3: Características Físicas del Terminal de Instalaciones.

2.2.1.2.2. Departamento de Cobranzas.

De igual manera, se ejecutó el programa **EVERST** para obtener la información.

Las fechas en las que fueron realizadas fueron:

- Estuardo Siza: 2009-10-08 a las 10:10.
- William Guachamín: 2009-10-08 a las 10:31.

DEPARTAMENTO DE COBRANZAS	
NOMBRE	MODELO Y CARACTERÍSTICAS FÍSICAS
 	<p><u>Computador Estuardo Siza:</u></p> <p>Nombre: ESIZA</p> <p>MAC: 00-16-EC-27-61-2E</p> <p>Sistema operativo: Windows XP Professional.</p> <p>CPU: Intel Pentium 4 531, 3000 MHz (15 x 200)</p> <p>Video: VIA/S3G UniChrome Pro.</p> <p>Impresora: Ccalderon\Xerox WorkCentre PE16</p> <p>Nombre DNS: esiza.DATADOSMIL.local</p> <p>Procesadores: Intel Intel(R) Pentium(R) 4 CPU 3.00GHz.</p> <p>Caché: 1024 KB SRAM</p> <p>RAM: 256 MB.</p> <p>Puertos de Comunicación: COM1, LPT1</p>
 	<p><u>Computador William Guachamín:</u></p> <p>Nombre: WGUACHAMIN</p> <p>MAC: 00-16-EC-57-C9-9A</p> <p>Sistema operativo: Windows XP Professional</p> <p>CPU: Intel Celeron D 331, 2666 MHz (20 x 133)</p> <p>Video: VIA/S3G UniChrome Pro</p> <p>Impresora: Ccalderon\Xerox WorkCentre PE16</p> <p>Nombre DNS: wguachamin.DATADOSMIL.local</p> <p>Procesadores: Intel Intel(F Continúa... ;PU 3.00GHz</p> <p>Caché: 1024 KB SRAM</p> <p>RAM: 256 MB.</p> <p>Puertos de Comunicación: COM1, LPT1</p>

Tabla 2.4: Características Físicas de los Terminales de Cobranzas.

2.2.1.2.3. Gerencia General.

Con la misma metodología, se realizó la prueba al terminal de la Sra. Cecilia Ponce el 2009-10-08 a las 11:30.

GERENCIA GENERAL	
NOMBRE	MODELO Y CARACTERÍSTICAS FÍSICAS
 	<p><u>Computador Cecilia Ponce:</u></p> <p>Nombre: CPONCE</p> <p>MAC: : 00-16-EC-32-8A-22</p> <p>Sistema operativo: Windows XP Professional.</p> <p>CPU: Intel Celeron D 331, 2666 MHz (20 x 133)</p> <p>Monitor: Samsung 19"</p> <p>Impresora: Lexmark.</p> <p>Nombre DNS: omzambrano.DATADOSMIL.local</p> <p>Procesadores: Intel Intel(R) Pentium(R) 4 CPU 2.00GHz.</p> <p>Caché: 512 KB SRAM</p> <p>RAM: 512 MB.</p> <p>Puertos de Comunicación: COM1, LPT1.</p>

Tabla 2.5: Características Físicas del Terminal de Gerencia General.

2.2.2. TOPOLOGÍA ACTUAL DE LA RED.

2.2.2.1. Cableado Estructurado.

El edificio donde se encuentra la empresa también posee departamentos domiciliarios y otras empresas. En el año 2007, TELINEMAX S.A. compró una planta entera en este edificio, la cual llevaba tres departamentos, dos de gran tamaño y un pequeño.

TELINEMAX S.A. tuvo que adecuar sus instalaciones para poder trasladarse, para esto requería la implementación de un cableado estructurado. Ya que la empresa trabaja directamente en el campo de las telecomunicaciones, no hubo problema con que ellos mismos realizaran el cableado estructurado.

TELINEMAX S.A. comenzó su trabajo en dos departamentos, uno grande y otro pequeño; el cableado fue realizado nada más para estos dos departamentos, sin saber que un año más tarde se uniría en la misma estructura TELCOLSAM S.A.

Con la llegada de TELCOLSAM S.A., fue necesario un cambio físico, quedándose en los dos departamentos grandes; esta distribución se mantiene hasta la actualidad. Sin embargo, se radicó un problema, el cableado estructurado no fue implementado en el último departamento; por tanto, TELINEMAX S.A. estuvo obligado a rediseñar un cableado sin perder lo antes ya realizado.

Gracias a este rediseño, el cableado nunca fue certificado, puesto que se realizaron detalles que no entran en las normas, con el fin de solucionar esta carencia de la manera más rápida y económica posible.

A continuación, se presenta un plano con la distribución física en la empresa:

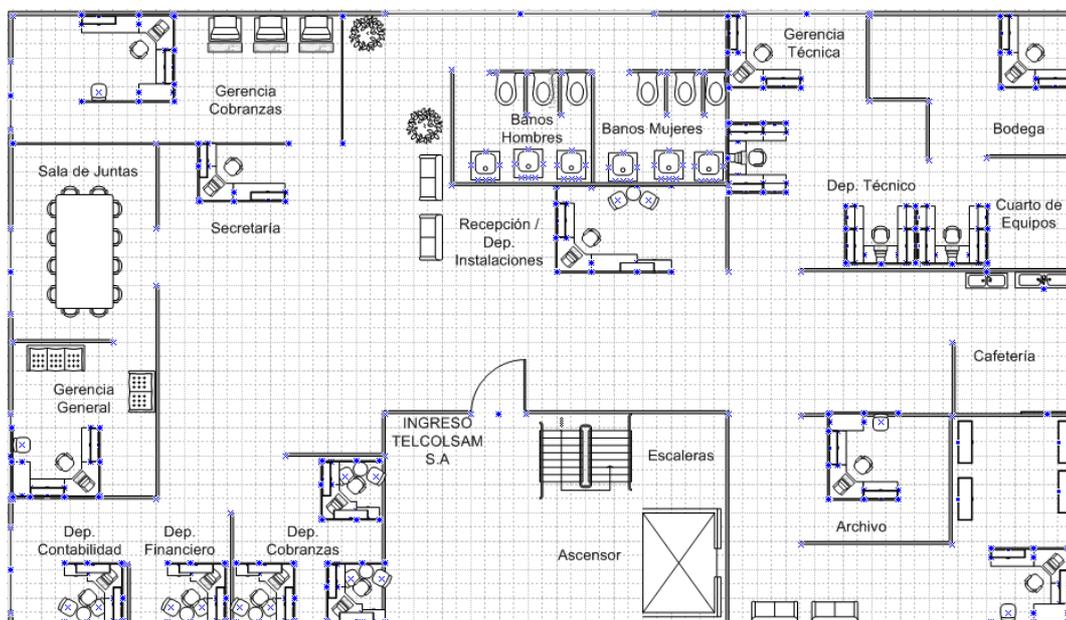


Figura 2.3: Plano de TELCOLSAM S.A. y TELINEMAX S.A.

Hoy por hoy, TELCOLSAM S.A. tiene cubierto aproximadamente el noventa por ciento de su estructura física con su cableado. Este cableado se encuentra realizado con cables CAT 5Ee en su totalidad, y usando canaletas para su distribución.

Mediante un seguimiento de los puntos de datos existentes, se pudo realizar el siguiente plano del cableado:

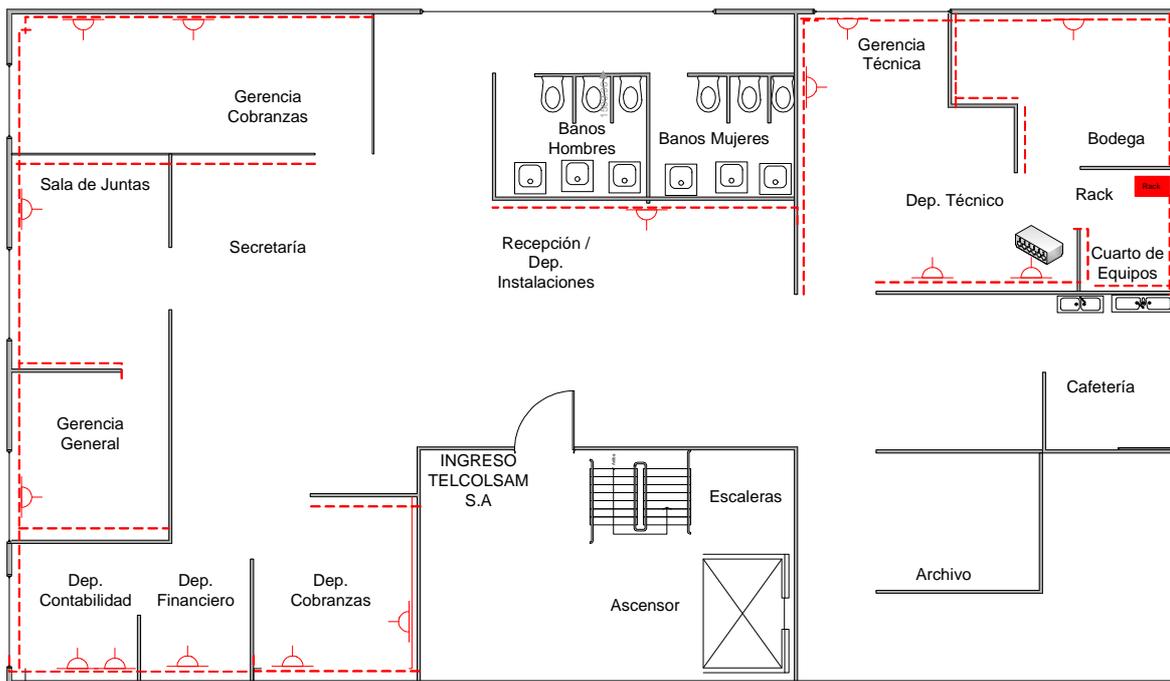
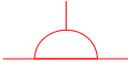


Figura 2.4: Cableado en TELCOLSAM S.A.

Figura	Descripción
	Canaletas
	Salida de Comunicaciones. Consta de 1 punto de voz y otro de datos
	Rack/ Cuarto de Equipos
	Switch/ Conmutador

Cabe destacar que las divisiones entre puestos de trabajo y departamentos son de paneles de madera, por lo cual, no fue necesario destruir paredes para poder pasar los cables.

Gracias al cambio físico que tuvo TELINEMAX S.A. al recibir a TELCOLSAM S.A., existen problemas con el número de puntos necesarios; por lo tanto, como solución, en estos puntos de datos fue incluido un Switch.

Además, es importante resaltar que el espacio donde se encuentra el “Archivo”, no tiene cableado estructurado. No ha sido necesario ya que en este lugar no trabaja persona alguna sino, por el contrario, solamente es usado para depositar papeles de importancia menor de años anteriores.

2.2.2.2. Topología Lógica.

TELCOLSAM S.A. es una empresa no muy extensa; por tanto, trabaja con una dirección privada 192.168.0.0. Consta de 9 puestos de trabajos distribuidos de la siguiente manera:

DIRECCIONAMIENTO LÓGICO TELCOLSAM S.A.		
Departamento	PC	Dirección IP
Gerencia General	C. Ponce	DHCP
Cobranzas	W. Guachamín	192.168.1.19 / 255.255.255.0
Cobranzas	E. Siza	192.168.1.20 / 255.255.255.0
Instalaciones	C. Rivas	192.168.1.8 / 255.255.255.0
Técnico	M. Zambrano	192.168.1.9 / 255.255.255.0
Financiero	V. Ortiz	DHCP
Financiero	M. Chuqui	192.168.1.23 / 255.255.255.0
Contabilidad	D. Moya	192.168.1.22 / 255.255.255.0

Tabla 2.6: Direccionamiento lógico de TELCOLSAM S.A.

TELCOLSAM S.A. es una red híbrida, puesto que se encuentra compuesta de una parte inalámbrica y otra cableada.

En la parte cableada, para poder salir al exterior, PUNTO NET presta sus servicios haciendo una conexión con el Cerro Pichincha para luego inalámbricamente conectarse con TELCOLSAM S.A. Este enlace WAN se da en la red 200.105.227.0. Para recoger dicha señal enviada desde una antena, se dispone de un Punto de Acceso WAN Radio LAN/ PC instalado en la terraza del edificio.

A inicios de este año, TELCOLSAM S.A. integró a su arquitectura una parte inalámbrica, puesto que llegan visitantes con computadores portátiles y en su mayoría de casos requieren salida a Internet, además de obtener movilidad en todo el espacio. Las características de todos los equipos vistos en este apartado se encuentran en el **Título “Equipos de interconectividad.”2.2.1.1.1, página 47.**

A continuación presentamos un diagrama lógico de la Empresa:

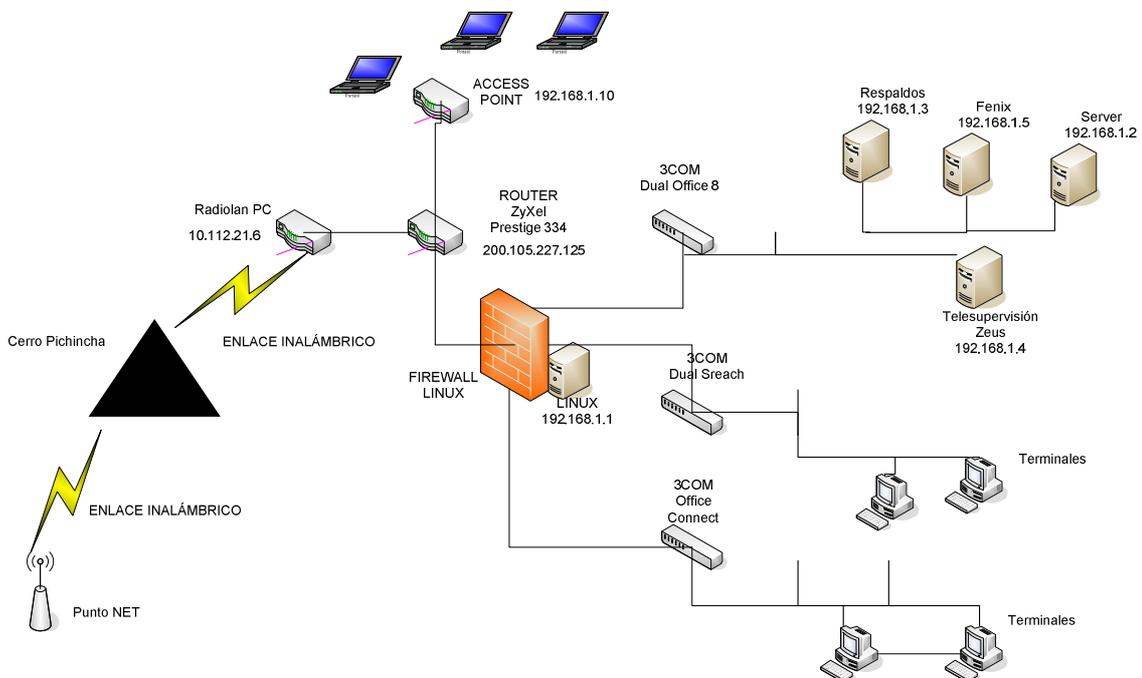


Figura 2.5: Diagrama lógico de la red de TELCOLSAM S.A.

2.3. DESCRIPCIÓN DE LAS APLICACIONES INSTALADAS.

Es importante determinar el software instalado en la empresa para, de esta manera, tener una mejor visión de la situación actual, y cuál es la carga que soporta cada terminal dependiendo del tipo de trabajo que realiza.

A continuación, se realizará una tabulación de las aplicaciones instaladas en los diferentes dispositivos.

2.3.1. SERVIDORES.

El software para los servidores fue obtenido de manera manual, usando el Panel de Control (Agregar o Quitar programas).

SERVIDORES					
APLICACIÓN	ZEUS	RESP	FENIX	LINUX	SERVER
Adobe Reader 7.0 – español			X		
Adobe Flash Player Active X					X
Advanced SystemCare 3	X	X	X		
Windows Media 9			X		
C-Media 3D Audio			X		
Avast! Server Edition	X	X			
Database Comparer v. 3.2		X			
Firebird 2.0.0	X	X	X		
Herramientas de administración remota	X	X			
IB Expert 2007 trial edition	X				
Microsoft .NET Framework 2.0	X		X		
Microsoft Office XP Professional con FrontPage					X

Continúa en la página 58.

Microsoft Office Basic Edition 2003	X	X	X		
Remote Administrator v 2.2	X	X	X		X
SAR II – Sistema de Administración Remota	X		X		
ProSavageDDR and Utilities			X		
WinZip			X		X
Microsoft Internet Security and Acceleration Server			X		
Microsoft Visual FoxPro 9.0 Professional Express			X		
Microsoft SQL Server Express 2005			X		
Microsoft SQL Server 2000					X
Java (TM)6 Update 2/3/5/7			X		
Internet Explorer Q903235					X
WinRAR archiver	X				X
Sacet_Ecuador					X
SaciAnex 5.2					X
Iomega Automatic Backup					X

Tabla 2.7: Aplicaciones instaladas en los servidores.

Podemos observar que el Servidor Linux solo tiene aplicaciones instaladas por defecto. El Gerente Técnico sustenta las siguientes justificaciones:

- a) No existen muchos virus ya que su arquitectura es difícil de vulnerar, incluso al actualizar el kernel (Núcleo) los virus van quedando inutilizados.

- b) Los virus no atacan a Linux gracias a su sistema de archivos. Además, para ejecutar un posible virus sería necesario correrlo como root (Administrador)
- c) Al ser un SO libre, la comunidad ayuda grandemente en la resolución de dificultades que éste presente.

2.3.2. TERMINALES.

Tal como se mencionó anteriormente, para obtener esta información nos valimos de la herramienta EVEREST.

Siglas	Descripción
PC-IN	Computador Celia María Rivas (Instalaciones)
PC-WG	Computador William Guachamín (Cobranzas)
PC-ES	Computador Estuardo Siza (Cobranzas)
PC-CP	Computador Cecilia Ponce (Gerencia General)

TERMINALES				
APLICACIÓN	PC-IN	PC-WG	PC-ES	PC-CP
Adobe Acrobat 5.0	X	X	X	X
Ardilla_cervecera 1.00		X		
Avast! Antivirus	X	X	X	X
Firebird 2.0.0	X	X	X	X
Google Toolbar for Internet Explorer	X	X	X	X
Hp LaserJet-all-in-one	X			
Java(TM) 6 Update 15		X	X	X
Java(TM) 6 Update 3	X	X	X	X
Microsoft .NET Framework 1.1	X	X	X	X
Microsoft .NET Framework 2.0	X	X	X	X
Microsoft .NET Framework 2.0 Service Pack 1	X	X	X	X
Microsoft .NET Framework 3.0 Service Pack 1	X	X	X	X

Continúa en la página 60.

NET Installation Assistance for VB6		X		
Microsoft .NET Framework 3.5	X	X	X	
Microsoft Internationalized Domain Names	X			
Microsoft Silverlight			X	
Microsoft Office Professional Edition 2003	X	X	X	X
Remote Administrator v2.2	X	X	X	X
SARII - Sistema de Administración Remota	X	X	X	X
TuneUp Utilities 2008	X			
Microsoft Internet Explorer 7.0.5730.13	X			
Open XML Editor 1.4	X			
Open XML Editor 1.4	X			
MSXML 6.0 Parser (KB933579)	X			
XML Paper Specification Shared Components Pack 1.0	X	X	X	
Hotfix for Windows XP	X			
MSN Toolbar			X	X

Tabla 2.8: Aplicaciones instaladas en los terminales.

TELCOLSAM S.A. hoy en día no tiene equipos de punta, sin embargo funcionan bien para el trabajo que realizan (hojas de cálculo en Excel, por ejemplo). La visión de TELCOLSAM S.A. es que en un futuro cercano se pueda proveer a cada departamento de máquinas de punta, lo que significaría cambiar a un SO como Windows Vista, o el más reciente Windows 7.

Aquí encontramos una de las justificaciones de este proyecto de titulación, puesto que el programa realizado con la tecnología WPF se caracteriza por su enriquecido manejo de interfaces gráficas en estos SO, pero sin negar la posibilidad de trabajar con el SO actual que tenga cada terminal.

CAPITULO 3



3. ANÁLISIS Y SELECCIÓN DE LAS HERRAMIENTAS DE SOFTWARE.

Dentro de los objetivos de la realización de este proyecto, se encuentra la utilización de una arquitectura Cliente–Servidor junto con una arquitectura en tres capas. Por tanto, como primera instancia, se procederá a detallar estos términos.

3.1. INTRODUCCIÓN TEÓRICA.

3.1.1. ARQUITECTURA CLIENTE–SERVIDOR.

3.1.1.1. Definición.

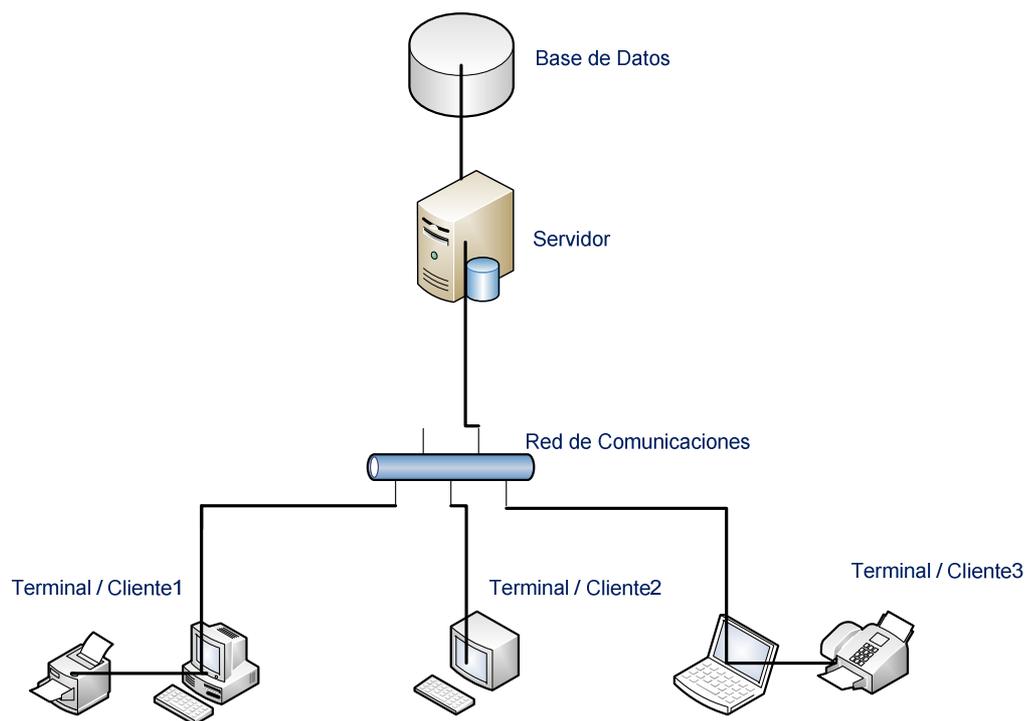


Figura 3.1: Arquitectura Cliente – Servidor

Una arquitectura se define como la esquemmatización de un plan o proyecto general, para asegurar que las necesidades de los usuarios sean resueltas.

La arquitectura Cliente-Servidor es un modelo de computación en el que el procesamiento para ejecutar una aplicación se divide entre dos o más procesos que cooperan entre sí.

Una arquitectura Cliente–Servidor, como su nombre lo indica, tiene dos elementos claramente determinados, según su hardware:

- **CLIENTE o FRONTEND:** Son terminales, estaciones de trabajo, que realizan peticiones o solicitudes a un servidor, de uno o varios servicios.

Estos servicios pueden ser peticiones: de entrega de información a una base de datos, de información contenida en archivos o los archivos en sí mismos, o peticiones de volcado de datos en un dispositivo de salida o almacenamiento.

También se lo conoce como dispositivo maestro, ya que tiene un papel activo en la comunicación al ser quien tome la iniciativa para un trabajo.

Entre las operaciones que generalmente realizan están:

- El manejo de una interfaz de Usuario.
 - Validación de datos de entrada
 - Generación de consultas, informes y peticiones sobre información en una determinada Base de Datos.
 - Abrir y cerrar el canal de comunicaciones al iniciar y terminar la ejecución del proceso.
 - Esperar y recibir las respuestas por parte del Servidor
-
- **SERVIDOR o BACKEND:** Son máquinas bastante potentes (poseen un hardware poderoso), que trabajan como recolectores de información, actuando como SGBD (Sistema Gestor de Base de Datos), o en inglés DBMS (Data Base Management System).

Un solo servidor es capaz de estar conectado a una gran cantidad de Clientes, dependiendo de su capacidad; o sea, podrá servir a todos y cada uno de manera concurrente.

A este elemento también se lo conoce como dispositivo esclavo. Se lo llama así puesto que desempeña un trabajo pasivo en la comunicación; en otras palabras, si el cliente no hace una petición, el servidor no realizará procedimiento alguno.

Entre sus operaciones más relevantes, están:

- Esperar las diferentes solicitudes por parte de los clientes.
- Luego de recibir una petición, su responsabilidad es entender y procesar dicha solicitud para posteriormente mandar su respuesta al Cliente.
- Ya que trabajamos en una arquitectura TCP/IP, el servidor es el encargado de proporcionar un puerto por el cual se deben conectar los clientes para realizar su petición.
- Gestión de elementos compartidos. Podría ser una Base de Datos o también periféricos de hardware conectados.
- Si es el caso, al estar conectado con otras redes de área local o extensa, deberá cumplir con sus responsabilidades pertinentes.

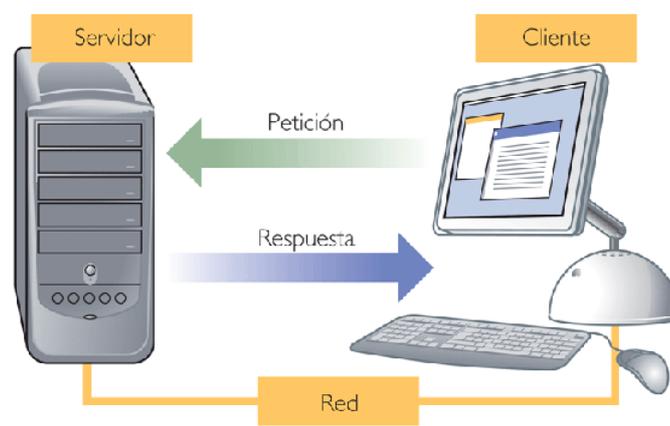


Figura 3.2: Petición – Respuesta en un modelo Cliente - Servidor¹⁰.

¹⁰FUENTE:http://www.kalipedia.com/kalipediamedia/ingenieria/media/200708/21/informatica/20070821kpinginf_30.Ees.SCO.png

3.1.1.2. Justificación.

Hoy por hoy, los sistemas informáticos son de real importancia en la estructura de una empresa, en general, agilizan las operaciones, minimizan el papeleo, todo para un objetivo primordial: aumentar la competitividad de la empresa.

A través de los años la tecnología se ha ido modificando. El modelo Cliente–Servidor entra a cambiar ciertos aspectos con el fin de evolucionar a otra etapa con mayores beneficios. Algunos aspectos que este modelo ha venido a evolucionar, son:

COMPARATIVA ACTUAL CLIENTE–SERVIDOR		
	ANTES	DESPUÉS
AVANCE TECNOLÓGICO	Rigidez: Se limitaba a un solo equipo de procesamiento.	Un servidor puede procesar varias solicitudes de diferentes Clientes. De igual manera, un cliente puede hacer peticiones a diferentes servidores.
	Vinculación al Sistema: Toda la información se encuentra al alcance de cualquier usuario. No existe seguridad del sistema ya que se encuentra en una sola capa.	Existe un repositorio de datos, encontrándose seguros. El usuario no tiene vinculación total con el Sistema, solamente con su propia interfaz.
	Solapamiento, duplicación y redundancia: Con la idea de que todos los usuarios tengan la información necesaria a la mano, se trabajaba con varios archivos con la misma información y diferentes cambios, provocando redundancia y duplicación.	La información se encuentra al alcance de todos los usuarios, totalmente actualizada y sin duplicidad.
	Imposibilidad de migración de la plataforma: Los requerimientos de la aplicación limitarían a la posibilidad de migrar de plataforma.	Ya que el Usuario es independiente del Servidor, las plataformas pueden ser cambiadas sin ningún problema.

Continúa en la página 66.

EXIGENCIAS DEL CANAL	Factor de tiempo crítico: Como en el caso de TELCOLSAM S.A., se trabaja con un único archivo compartido, incapacitando a los demás usuarios el uso de dicha información.	Los usuarios pueden hacer uso de la información de manera simultánea, de esta forma se optimiza el tiempo de trabajo.
	Desmotivación del Usuario u Operador: El usuario tiene un trabajo masivo, repetitivo y desmotivador.	El modelo brinda al usuario mayores beneficios como: validaciones en tiempo de ejecución gracias a una capa de negocios, creando un usuario analista de sus datos entrantes.
ENTORNO GENERAL	Adaptación a la capacidad del ordenador: Todo trabajo es procesado por cada ordenador, por tanto, la capacidad de respuesta se encuentra netamente ligado a sus características	Los procesos más demandantes son ejecutados por el servidor. Ya que éste tiene una gran capacidad hardware no tendrá ningún inconveniente y, por el contrario, el cliente realiza un procesamiento liviano, envía peticiones y recibe resultados.
	Ordenadores costosos: Para tener mayor satisfacción en la respuesta, se requiere subir las características de cada ordenador.	Se puede trabajar con los actuales sin necesidad de adquirir otros de mejor capacidad.
	Limitación de la Aplicación: Con el afán de que los ordenadores puedan rendir, se limitaba los beneficios de la aplicación.	La aplicación puede ser exclusiva para cada empresa, y puede ser realizada a la medida de cada una.

Tabla 3.1: Justificación de la Arquitectura Cliente-Servidor¹¹.

3.1.1.3. Ventajas y Desventajas.

A continuación se detalla las principales ventajas que se obtiene al implementar una arquitectura Cliente–Servidor:

¹¹ FUENTE: www.dlsi.ua.es/asignaturas/sid/sid2001-t4.ps

VENTAJAS DE LA ARQUITECTURA CLIENTE-SERVIDOR	
RECURSOS CENTRALIZADOS	Los clientes son conectados a través de la red a un punto central llamado Servidor. Desde ahí puede administrar cualquier recurso común a los usuarios.
CONSISTENCIA DE LA INFORMACIÓN	Gracias al control centralizado de los recursos compartidos.
SEGURIDAD MEJORADA	Esto radica en que la cantidad de puntos de entrada no le resta seguridad al sistema.
ADMINISTRACIÓN	El punto fuerte de esta arquitectura es el Servidor, por lo cual se le entrega un mantenimiento constante; mientras que los clientes requieren una administración leve.
RED ESCALABLE HORIZONTALMENTE	El número de clientes puede crecer a medida que la empresa lo requiera, sólo limitada si así lo quiere su administrador o limitada por las capacidades del servidor.
RED ESCALABLE VERTICALMENTE	Se refiere a la posibilidad de migración a otras plataformas. Mientras más pasa el tiempo los SO van mejorando y actualizando y, sin duda, el poder actualizar nuestras plataformas es de gran ayuda, ya que no es necesario un cambio en la aplicación.
INTER-OPERATIVIDAD ENTRE DISTINTOS HARDWARE Y SOFTWARE	Ofrece la capacidad de trabajar con el equipamiento actual, y no es necesario hacer un financiamiento extra para adquirir elementos nuevos. Además de no estar atados a que toda la red tenga una misma estructura de hardware ni de software, esta arquitectura permite operar con cualquier variedad de los mismos.
COSTOS	Contribuye considerablemente a la reducción de costos, puesto que, brinda una gran versatilidad en fabricantes y costos.
INTERACCIÓN CON EL USUARIO	Es muy favorable al presentar interfaces más interactivas, amigables, de fácil empleo, e intuitivas para el usuario.

Tabla 3.2: Ventajas Cliente–Servidor.

Por otro lado, esta arquitectura presenta algunos inconvenientes, que se citan a continuación:

INCONVENIENTES DE LA ARQUITECTURA CLIENTE-SERVIDOR	
CONGESTIÓN DE TRÁFICO	El problema se presenta cuando una gran cantidad de peticiones simultáneas llegan al Servidor, haciendo que éste tenga problemas de rendimiento.
SERVIDOR CAÍDO	Cuando este suceso se presenta, el servidor deja sin servicio a todos sus clientes. El servidor es clave, si se pierde el punto central, no existe manera de una conexión.
SOFTWARE Y HARDWARE EN EL SERVIDOR	Entre mayor sea su trabajo, mayor deberían ser sus características; por tanto, aumentará su costo.
RECURSOS COMPARTIDOS	El cliente no dispone de los recursos que existen en el servidor. Por ejemplo, no es posible imprimir directamente, sin acceso al recurso que ofrece el servidor.
MANTENIMIENTO	El mantenimiento se dificulta, pues implica diagnosticar fallas en tantos diferentes sistemas (hardware y software) como se tenga en la red.

Tabla 3.3: Inconvenientes de una arquitectura Cliente-Servidor.

3.1.2. ARQUITECTURA EN CAPAS.

La programación por capas se define como un estilo de programación cuyo objetivo esencial es la separación de la Capa de Presentación, Capa de Negocios y la Capa de Datos (aunque pueden existir diferentes capas, o incluso diferente número de capas). De esta manera, si existe algún cambio, no es necesario interactuar con todas las capas y por tanto con todo el código.

Una de las ventajas que proporciona esta arquitectura es la gran distribución del trabajo; o sea, cada cliente se encuentra realizando su trabajo independientemente del otro. Cada grupo de trabajo está totalmente abstraído del

resto de niveles donde el cliente solamente conocerá su respectiva interfaz de Usuario.

3.1.2.1. Definiciones Previas.

3.1.2.1.1. Interfaz de Programación de Aplicaciones (API).

La API es un biblioteca, generalmente dinámica o DLL (Dynamic Link Library), utilizada para conseguir abstracción en la programación de las capas inferiores, brindando simplicidad en la programación, o incluso independencia del SO o del lenguaje de programación.

Visto desde código en bajo nivel (lenguaje de máquina), tiene como propósito proporcionar un grupo de funciones y procedimientos de uso general. La gran ventaja radica en que el programador evita el trabajo de programar funcionalidades ya existentes. Sin embargo, las APIs de alto nivel generalmente pierden flexibilidad; por tanto, al elegir una API se debe llegar a un equilibrio entre su potencia, simplicidad y pérdida de flexibilidad.

Entre las APIs más destacadas se encuentran: Microsoft WMI, Microsoft Win32 API, Microsoft Framework .NET, OpenGL, Java EE, entre otros.

3.1.2.1.2. .NET Framework.

Un “Framework” es un soporte definido, en el cual se apoyan otras aplicaciones para poder ser desarrolladas. Generalmente, consta de programas y bibliotecas para ayudar a unir los diferentes componentes de una aplicación.

.NET Framework se encuentra incluido en los Sistemas Operativos: Windows Server 2008, Windows Vista y Windows 7; sin embargo, es posible instalar de forma manual en Windows XP y en la familia de Windows Server 2003. Por otra

parte, una versión “reducida” se encuentra en dispositivos inteligentes en el SO Windows Mobile.

Los componentes claves para .NET Framework son:

- **El Entorno Común de Ejecución para Lenguajes o CLR:** Administra el código en tiempo de ejecución, proporcionando los servicios básicos (administración de memoria, control de excepciones, control de hilos de ejecución). Este es el verdadero núcleo de .NET Framework.

La herramienta de desarrollo compila el código fuente de cualquiera de los lenguajes soportados por .NET en un código intermedio, el MSIL (Microsoft Intermediate Lenguaje), similar al BYTECODE de Java. Para generarlo, el compilador se basa en la especificación CLS (Common Language Specification) que determina las reglas necesarias para crear el código MSIL compatible con el CLR. Para ejecutarse se necesita un segundo paso, un compilador JIT (Just-In-Time) es el que genera el código máquina real que se ejecuta en la plataforma del cliente.

De esta forma se consigue con .NET independencia de la plataforma de hardware. La compilación JIT la realiza el CLR a medida que el programa invoca métodos. El código ejecutable obtenido se almacena en la memoria caché del ordenador, siendo recompilado de nuevo sólo en el caso de producirse algún cambio en el código fuente.

- **La Biblioteca de Clases Base o BCL de .NET Framework:** Define un conjunto funcional mínimo para que sea soportado por un Sistema Operativo. Contiene los tipos básicos como clases de entrada/salida de datos (XML, TCP/IP), seguridad, manejo de dispositivos periféricos, etc.

Incluye dos componente ADO.NET y ASP.NET:

- **ADO.NET** se deriva de ActiveX Data Object y es usado para acceso a información y servicios de datos en un SGBD.
- **ASP.NET** no es otra cosa que un Framework para aplicaciones web. Su primera aparición fue junto con la versión de .NET Framework 1.0.

- **El conjunto de lenguajes de programación:** Gracias al CLI (Infraestructura común de Lenguajes)¹², actualmente .NET soporta ya más de cincuenta lenguajes de programación, dando una gran versatilidad en el desarrollo de un programa, sin tener mayores diferencias si se programa con uno u otro lenguaje.

Entre los lenguajes más usados en el desarrollo .NET están: C#, Visual Basic .NET, Delphi (Object Pascal), C++, J#, Perl, Python, Fortran, Cobol y PowerBuilder.

WPF para poder tener todo su potencial en marcha sugiere un Framework versión 3.0 o superior, sin embargo Microsoft ha lanzado el primer Service Pack para su Framework .NET 2.0. Entre las mejoras que proporciona este Service Pack podemos destacar mejoras de seguridad importantes y soporte a los prerequisites de .NET Framework 3.0 Service Pack 1 y .NET Framework 3.5.

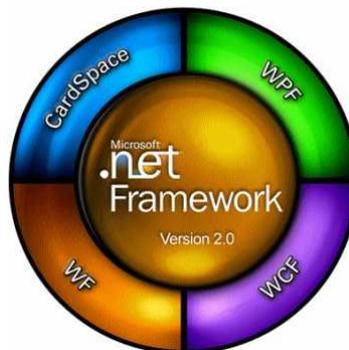


Figura 3.3: WPF y .Net Framework 2.0.¹³

¹² CLI es una especificación estandarizada cuya característica esencial es permitir que diferentes lenguajes de alto nivel puedan ejecutarse en múltiples plataformas sin necesidad de recompilar el código fuente.

¹³ FUENTE: http://www.visualbeta.es/wp-content/uploads/2007/12/Framework-net-2_0.jpg

3.1.2.1.3. *Kit de Desarrollo de Software de Windows (SDK).*

SDK se presenta como un API incluyendo herramientas, documentación y ejemplos que necesitan los desarrolladores para escribir, generar, probar e implementar aplicaciones de .NET Framework. Es compatible con los SO de las familias Windows 2000; Windows Server 2003; Windows Vista, Windows XP, Windows 7.

Los SDKs frecuentemente incluyen, también, códigos de ejemplo y notas técnicas de soporte u otra documentación de soporte para ayudar a clarificar ciertos puntos del material de referencia primario.

3.1.2.1.4. *Entorno Integrado de Desarrollo (IDE).*



Un IDE es un entorno de programación que suele consistir de un editor de código, un compilador y un constructor de interfaz gráfica.

Visual Studio 2008 es un IDE creado por Microsoft®. Éste presenta un conjunto de herramientas destinadas a escribir y modificar código, además a detectar y corregir errores en las diferentes aplicaciones. Soporta varios lenguajes de programación y se han desarrollado las extensiones necesarias para muchos otros no nativos de Microsoft®.

La primera versión fue Microsoft Visual Studio 97. Ésta incluía Visual Basic 5.0 y Visual C++ 5.0 para realizar software para Windows específicamente, mientras que Visual J++ 1.1 era para Java y Windows. Los otros lenguajes eran Visual Fox

Pro 5.0 para la Bases de Datos y Visual InterDev 1.0 para crear sitios dinámicos con ASP.

Microsoft Visual Studio 6.0 fue lanzada al mercado en el siguiente año, 1998 y de igual manera se introdujeron otros lenguajes de programación.

Microsoft Visual Studio .NET 2002 fue un salto drástico, ya que cambia completamente la estructura. Ahora con .NET los programas no se compilan para generar un ejecutable máquina (un archivo .exe conocido comúnmente para el ambiente Microsoft® Windows), sino que se genera un archivo intermedio para poder ser ejecutado en diferentes plataformas de hardware y software, como: Linux, Solaris o Mac. A esta versión se le agregan nuevos lenguajes que son: Visual J# (sucesor del desaparecido Visual J++), Visual C# (no confundir con Visual C++) y Visual Basic .NET.

Microsoft Visual Studio .NET 2003 no tuvo un fuerte cambio, lo tuvo la versión anterior, mientras que la versión del 2002 era la versión 7.

Microsoft Visual Studio 2005 tuvo un cambio muy importante. Para este IDE, es posible trabajar con el Framework .NET 2.0, compilar hacia aplicaciones de 32 bits y 64 bits y trabajar con Microsoft SQL Server 2005. Visual Studio 2008 fue publicado el 17 de Noviembre de 2007 en inglés, mientras que la versión en castellano no fue publicada hasta el 2 de Febrero de 2008.

El nuevo Framework (.Net 3.5) con el que trabaja Visual Studio 2008 está diseñado para aprovechar las ventajas que ofrece Windows Vista y el actual Windows 7 a través de sus subsistemas "Windows Communication Foundation" (WCF) y "Windows Presentation Foundation" (WPF).

Es importante destacar que si se instala Microsoft Visual Studio 2005 o posterior (excepto las ediciones Express), no es necesario instalar el SDK de .NET Framework por separado.

Microsoft, en agosto del 2009 anuncia la nueva generación de Visual Studio (Microsoft Visual Studio 2010 beta 2), junto a un nuevo Framework 4.0.

Los requisitos¹⁴ que debe tener un sistema para un correcto funcionamiento en este IDE son:

- **Sistema Operativo:** Windows Server 2003; Windows Server 2008; Windows Vista; Windows XP, y actualmente Windows 7.
- **Mínimo:** 1.6 GHz CPU, 384 MB RAM, pantalla 1024x768, disco duro 5400 RPM
- **Recomendado:** 2.2 GHz o mayor de CPU, 1024 MB o mayor en RAM, pantalla de 1280x1024, 7200 RPM o mayor de Disco Duro.
- **Sobre Windows Vista o Windows 7:** 2.4 GHz CPU, 768 MB RAM.

3.1.2.2. Definición de Capas.

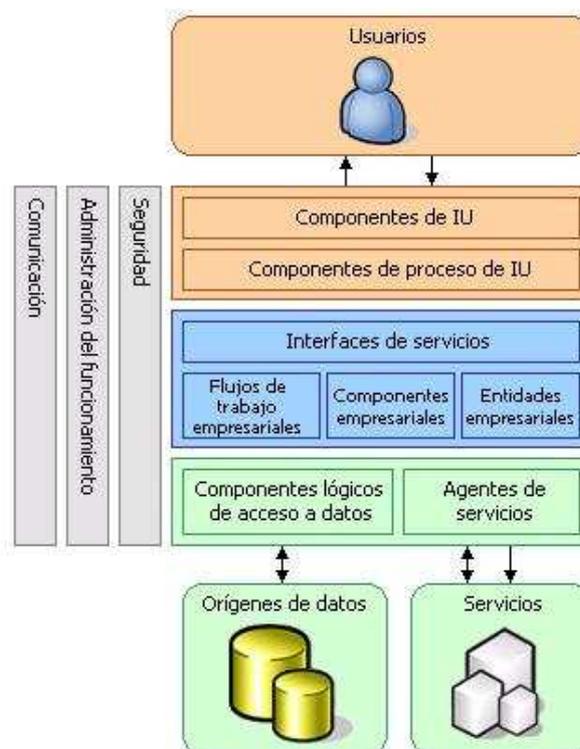


Figura 3.4: Arquitectura en tres capas.

¹⁴**Fuente:** <http://www.microsoft.com/downloads/details.aspx?FamilyID=75cbcbcd-b0e8-40ea-adae-85714e8984e3&displaylang=en#Requirements>

Este tipo de arquitectura es la más utilizada, debido a que ha tenido una significativa evolución en los últimos años.

Para su correcto funcionamiento lo más común es la división en tres capas básicas, citadas a continuación:

3.1.2.2.1. *Capa de Presentación, Capa Usuario o Interfaz de Usuario IU.*

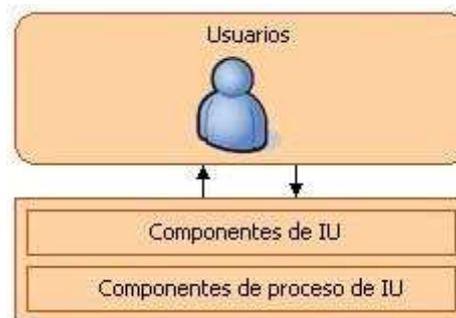


Figura 3.5: Capa de Presentación.

Esta es la capa que interactúa directamente con el Usuario. Típicamente, se encuentra conformada de una interfaz gráfica, que se necesita sea muy intuitiva y amigable. Generalmente, estas interfaces están representadas por formularios, tales como los usualmente empleados en: Word, Excel, navegadores web, entre otros.

Se crea un cliente “tonto” que carece de toda lógica del negocio, y simplemente ofrece alguna otra funcionalidad más que la visión y petición de datos.

Esta capa se encarga de recopilar la información haciendo un filtrado de formatos básicos e intercambiar datos con las capas inferiores.

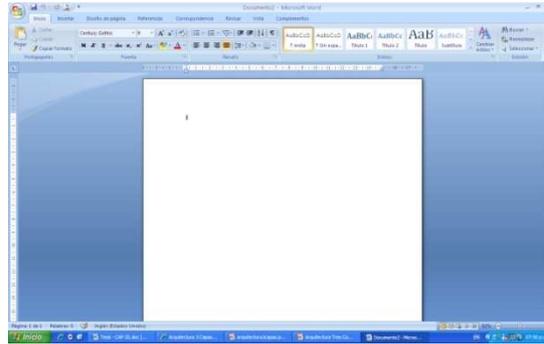


Figura 3.6: Ejemplo de formulario Microsoft ® Office Word.

3.1.2.2.2. *Capa de Negocios, Lógica del Negocio o Reglas del Negocio.*

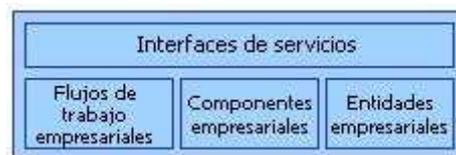


Figura 3.7: Lógica de Negocios.

Como su nombre lo especifica, en esta capa se realizan y se establecen todas las reglas que deben cumplirse según la lógica de la empresa. Es una capa intermedia que se caracteriza por recibir las solicitudes por parte de un cliente; por tanto, se encuentra conectada a una primera capa, la Capa de Presentación. Por otra parte, debe procesar las solicitudes del Cliente; para esto, es necesario obtener la información archivada en otra capa, la Capa de Datos, a la cual también está conectada.

Su objetivo principal es la comunicación entre las capas restantes (Presentación y Datos), pero también es la encargada de ofrecer seguridad, integridad de los datos, perfiles de usuario, y otros definidos por la lógica del negocio.

Toda aplicación mediana o grande que utilice esta capa, deberá ser robusta y realizar validaciones, cálculos en tiempo real y otras interacciones de usuarios.

3.1.2.2.3. Capa de Datos.

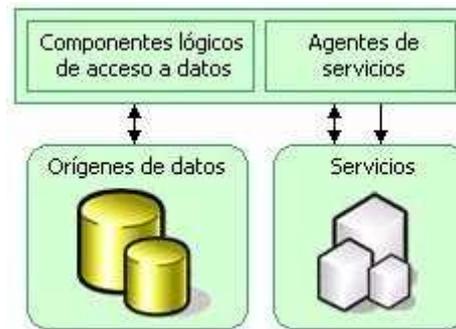


Figura 3.8: Capa de Acceso a Datos.

Es la capa en la cual se encuentran almacenados los datos, y su función específica es el acceso a éstos.

La Capa de Negocio se encuentra directamente conectada a ésta, recibiendo solicitudes de almacenamiento o de recuperación de la información. Puede estar formada por uno más SGBD, para realizar todos estos procesos de manipulación de información.

3.1.2.3. Definición de Niveles.

Comúnmente se suele confundir entre los términos de capas y niveles, siendo totalmente diferentes. La arquitectura tres capas podría estar en un solo terminal, o tal vez, dos capas en un solo terminal y la restante en otro terminal; de la misma manera, se podrían tener tres capas en tres diferentes terminales.

Los niveles se refieren a la distribución física que tienen las capas de la arquitectura.

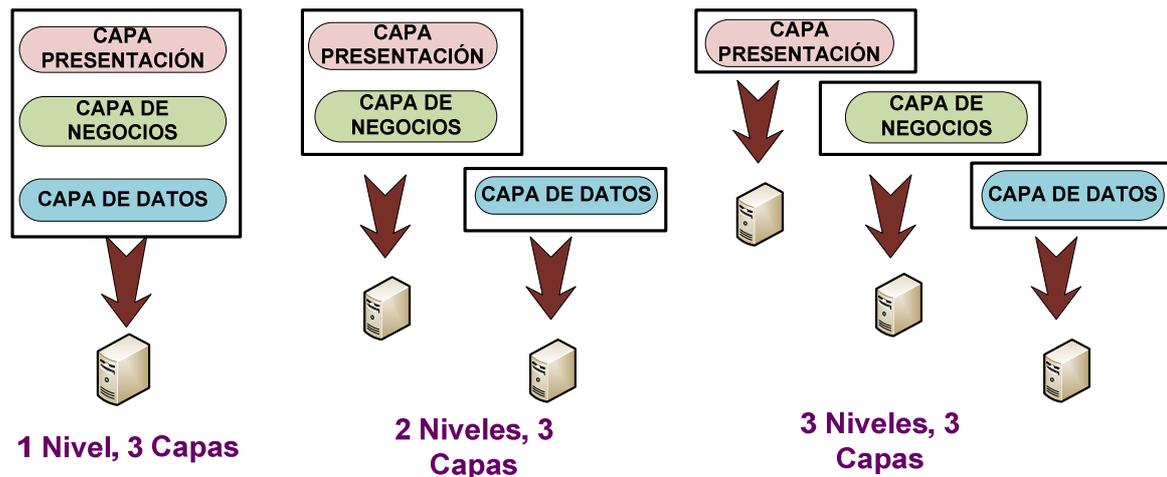


Figura 3.9: Niveles en una Arquitectura en Tres Capas.

Vale la pena recalcar que en la realización del programa se ha decidido usar una arquitectura en tres capas y dos niveles; por tanto, dos de las tres capas (Presentación y Reglas del Negocio) se depositarán en cada uno de los terminales, mientras que la Capa de Datos será ubicada en uno de los Servidores.

NOTA: La elección del Servidor donde ubicará la última capa, será detallada posteriormente al ver las características del SGBD a escoger.

3.2. SELECCIÓN DE LAS HERRAMIENTAS DE SOFTWARE MÁS ADECUADAS.

3.2.1. INTERFAZ GRÁFICA (PRIMERA CAPA).

Una interfaz gráfica consta de un grupo de ventanas en las cuales el usuario interactúa con una aplicación. Este proyecto de titulación se encuentra orientado a emplear una nueva tecnología llamada WPF (Windows Presentation Foundation). Esta herramienta se especializa en brindar al Usuario una interfaz amigable e intuitiva para un mejor acople al programa, utilizando de mejor manera el hardware de video disponible.

3.2.1.1. WINDOWS PRESENTATION FOUNDATION (WPF).

3.2.1.1.1. Definición.

WPF es una nueva innovación y generación de las tecnologías Microsoft®, además de ser uno de los pilares del Sistema Operativo Windows Vista y Windows 7; ofrece ampliar las características visuales en diferentes aplicaciones como Web o de Escritorio.

Su gran beneficio es que crea aplicaciones de Clientes Windows, incluyendo entre sus interfaces: animaciones, videos, audio, navegación, gráficos y animaciones 3D (en 3 dimensiones) y cualquier tipo de documentos soportados por Microsoft®.

Entre los ejemplos más reconocidos sobre la utilización de esta herramienta están:

- **Yahoo! Messenger:** Es un programa de mensajería instantánea, que permite el intercambio de archivos, videos, fotos, etc. Brinda al usuario una gran amplitud para que su interacción con este sea lo más agradable posible.¹⁵
- **New York Times Reader:** Es una nueva revolución en Comercio, aunque su acceso o descarga tiene un costo semanal, evita los molestos papeles y con la ventaja de volver a leer las noticias pasadas en un mismo lugar y a tan solo un clic de distancia. En pocos minutos consigue toda la información, para luego ser impresa desde un ordenador en el hogar.¹⁶

¹⁵ FUENTE: <http://mx.messenger.yahoo.com/>

¹⁶FUENTE:<https://timesreader.nytimes.com/webapp/wcs/stores/servlet/TimesReader?storeId=10001&catalogId=10001>

- **Contoso Healthcare Sample Application (Aplicación de ejemplo para la atención sanitaria):** Es un nuevo sitio para la comunidad, donde se interactúa con personal preparado en Medicina, para resolver problemas de salud.¹⁷

WPF permite cuatro tipos de aplicaciones:

- **Aplicaciones independientes:** Tienen un estilo tradicional de Windows, que se instalan en el equipo cliente y se ejecutan desde él.
- **Aplicaciones del explorador XAML (XBAPs):** Son aplicaciones compuestas de páginas de navegación que se exploran y hospedan por defecto sobre Windows Internet Explorer.
- **Bibliotecas de control personalizadas:** Ensamblados no ejecutables que contienen controles reutilizables.
- **Bibliotecas de clases:** Ensamblados no ejecutables que contienen clases reutilizables.

Para poder aprovechar al máximo el hardware, WPF se basa en la construcción de vectores, incluyendo un gran conjunto de características de programación, como: XAML (Lenguaje de Marcado de Aplicaciones Extensible), controles, enlace a datos, gráficos 2D y 3D, animaciones, estilos, plantillas, documentos, multimedia, texto, y tipografía.

Para comenzar una aplicación WPF existen por lo menos dos requisitos mínimos: el primero es **Microsoft .NET Framework**, para crear aplicaciones que incorporen otros elementos de la biblioteca de clases de .NET Framework; y por otro lado, un **SDK (Kit de Desarrollo de Software de Windows)**. Teniendo estos dos, es posible generar la aplicación desde una ventana de comandos, pero lo más recomendable es la instalación de un **IDE** como **Microsoft Visual Studio**.

¹⁷ FUENTE: <http://windowsclient.net/downloads/folders/wpfsamples/entry3754.aspx>

3.2.1.1.2. Lenguaje de Programación empleado en WPF.

WPF tiene la capacidad para programar mediante código de lenguaje marcado y subadyacente, una herramienta muy usada en Visual Studio 2005 y 2008. En general, se usa XAML para su apariencia y los lenguajes subadyacente para implementar su comportamiento:

- **Código Marcado:** Es más conocido mediante las siglas XAML, y obviamente su estructura es basada en XML. Es utilizado para crear la GUI, con herramientas comunes como ventanas, cuadros de diálogo, controles de Usuario, gráficos, etc.

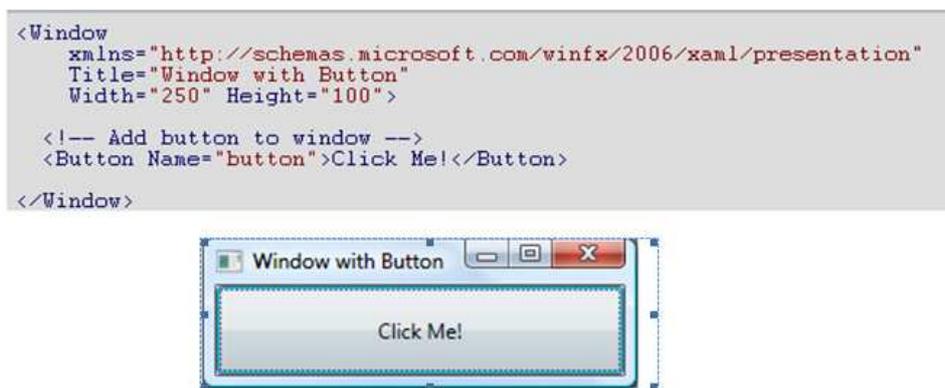


Figura 3.10: Ejemplo de lenguaje de Marcado XML y su respectiva representación GUI.

VER ANEXO 1.

- **Código Subadyacente:** Es un lenguaje de alto nivel, que proporciona la funcionalidad a los elementos realizados con el lenguaje marcado XAML. Por ejemplo, cuando el Usuario haga clic en un botón, éste despliegue el mensaje "Hello, Windows Presentation Foundation".

Los lenguajes más empleados son: Visual Basic y C#.

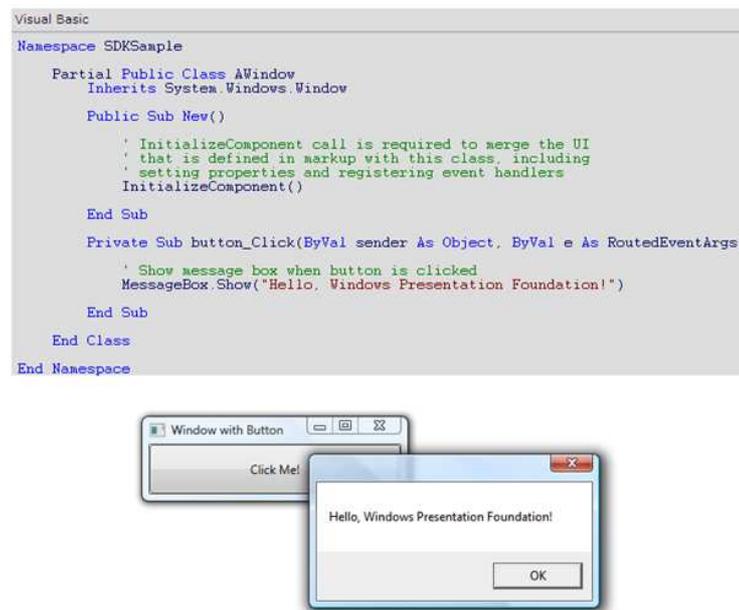


Figura 3.11: Ejemplo de lenguaje Subadyacente, Visual Basic .NET.

En la Arquitectura en tres capas, la Capa de Presentación será programada en XAML, mientras que las Reglas del Negocio serán programadas por uno de los lenguajes Subadyacentes.

Por otra parte, al mantener separados estos dos lenguajes obtendremos las siguientes ventajas:

- No se encuentra ligado el código de la apariencia con el código del comportamiento, ayudando así a un mantenimiento sencillo, puesto que al producirse errores no tendremos diferentes códigos rondando en un mismo sitio.
- Brinda a la aplicación modularidad, al tener la posibilidad de extender el código, siendo los nuevos elementos independientes de otros previamente creados.
- Ofrece una ayuda a los programadores al poder interactuar al mismo tiempo en dos diferentes códigos para luego ser unidos.
- Se pueden utilizar varias herramientas de diseño para implementar y compartir el lenguaje marcado XAML. Microsoft Expression

Blend proporciona una experiencia apropiada para los diseñadores, mientras que Visual Studio 2005/2008 está dirigido a los programadores.

- La globalización y localización de las aplicaciones WPF se ha simplificado en gran medida.

3.2.1.1.3. Lo nuevo en WPF con .NET Framework 3.5.

La versión Framework .NET 3.5 no tiene grandes diferencias con la versión 3.0, solamente fue un agregado de más librerías.

En un principio WPF salió junto con Framework 3.0, Sin embargo Microsoft ha evolucionado un nuevo escalón con el Framework 3.5, dando unas nuevas cualidades al programa.

- Primeramente, compatibilidad entre las diferentes versiones soportadas, WPF 2.0 + Service Pack 2.0, WPF 3.0 y WPF 3.5.

Tomando en cuenta que sobre la plataforma 3.5 corre cualquier aplicación realizada en 2.0 y 3.0; por el contrario, una aplicación 3.5 solamente correrá bajo el motor 3.0 o 2.0 si sus características están presentes en ese motor.

- Ahora, las XBAPs (Aplicaciones de Explorador) pueden ejecutarse sin problemas bajo Firefox¹⁸, desde la versión 2.0. Si este es el explorador por defecto, las XABPs respetarán su configuración sin utilizar Internet Explorer.

Cabe destacar que las características de seguridad presentes en Internet Explorer también se encontrarán disponibles en las ejecutadas en Firefox.

- Las aplicaciones WPF 3.5 tienen la posibilidad de crear, obtener y eliminar cookies.

¹⁸ Mozilla Firefox es un navegador web multiplataforma y varias versiones: Microsoft Windows, Mac OS X, GNU/Linux y algunos sistemas Unix, donde su código es Software Libre

- Es posible agregar un nuevo elemento XAML mediante el editor XAML de Visual Studio, asignarle un nombre (mediante el atributo Name), hacer referencia a él desde código subyacente y ver sus miembros desde el explorador de IntelliSense.
- Mejora la productividad en el lenguaje de marcado XAML.
- Ofrece mayor número de propiedades para diferentes controles, permitiendo alcanzar una interfaz más atractiva.
- Los enlaces de datos se facilitan en depuración gracias a nuevas propiedades integradas, además brindan nuevas alternativas para validación de datos.

CONCLUSIÓN:

Viendo las ventajas que proporciona la utilización de un IDE, y que dentro de los requisitos de WPF está .NET Framework, hemos decidido usar Visual Studio 2008 como entorno de desarrollo en nuestro proyecto ya que tiene todas las características que facilitan el uso de WPF para la realización de una aplicación que cumpla todos los requerimientos pedidos por TELCOLSAM S.A.

Además, como pudimos observar en el capítulo anterior, **Título “Terminales de usuarios por departamentos.”, página 49**, los computadores de TELCOLSAM S.A. cumplen con los requisitos mínimos que se necesita para la ejecución del programa.

La visión de la Gerencia General es cambiar de equipamiento lo más temprano posible y, de esta manera, la aplicación a realizar en este proyecto de titulación podrá ser usada más eficientemente en estos nuevos equipos.

Por tanto, concluimos que la utilización de Microsoft Visual Studio 2008 puede ser empleada sin ningún inconveniente, gracias a los SO actuales en los terminales de TELCOLSAM S.A. que soportan el .NET Framework 3.5 y su Service Pack versión 1.

Por último, indicamos que la Capa de Presentación será programada utilizando dos lenguajes:

- **XAML:** Para el uso de la tecnología WPF, netamente para la Interfaz Visual del Usuario.
- **Visual Basic .NET:** Se ha escogido este lenguaje para la programación de eventos¹⁹ en el programa Visual Studio 2008, ya que su estructura es muy similar al lenguaje humano y tiene mejor soporte de IntelliSense.

3.2.2. REGLAS DEL NEGOCIO (SEGUNDA CAPA).

Las Reglas del Negocio describe las políticas, normas, operaciones, definiciones y restricciones presentes en una organización y que son de vital importancia para alcanzar los objetivos.

Estas reglas deben encontrarse en todo el programa, por tanto debe tener comunicación total, con la Base de Datos y con la Presentación.

Ya que la interfaz gráfica se programará con la ayuda del IDE Visual Studio 2008, en el lenguaje de XAML, lo más sencillo es la utilización del mismo programa para dar una función a eventos posibles. También recordamos que Visual Studio 2008 brinda una gran libertad para interactuar con Microsoft Office al poder instalar e incorporar nuevas librerías facilitando grandemente este trabajo, puesto que uno de los requerimientos importantes para TELCOLSAM S.A. son los reportes exportados a Excel.

Se usará Visual Studio 2008 con C# porque es un lenguaje maduro, moderno y bien conocido; además de que se tiene mucha ayuda en la Comunidad para posibles dificultades en el desarrollo.

¹⁹ Un evento es una llamada al código que se produce cuando el usuario realiza una acción como, por ejemplo, pulsar un botón o seleccionar un elemento de una lista. Los eventos responden a la acción del usuario sobre un control ejecutando una función situada en el código.

3.2.3. SISTEMA DE GESTIÓN DE BASE DE DATOS, SGBD O DBMS.

3.2.3.1. Definiciones.

Una **Base de Datos** es un banco ordenado de datos, el cual permite almacenar información para un uso futuro. Hoy por hoy existe, una gran variedad de Base de Datos de donde escoger, por lo cual se realizará una comparación de las bases de datos más relevantes en el mercado actual.

Un **SGBD** es un sistema para gestionar la información de una base de datos, donde una gestión se describe mediante diferentes acciones como la: inserción, eliminación, actualización, o cualquier otra operación que se pueda realizar a la información.

3.2.3.2. Componentes de una Base de Datos.

- a) **Hardware:** Se refiere a los dispositivos de almacenamiento en donde reside la base de datos, así como a los dispositivos periféricos necesarios para su uso.
- b) **Software:** Es el programa que se conoce como SGBD, el cual maneja todas las solicitudes formuladas por los usuarios a la base de datos.
- c) **Usuarios:** Existen tres clases de usuarios relacionados con una Base de Datos:
 - El Programador de Aplicaciones, quien crea programas de aplicación que utilizan la base de datos.
 - El Usuario de la Base de Datos, quien accede a la Base de Datos por medio de un lenguaje de consulta o de programas de aplicación.
 - El Administrador de la Base de Datos (DBA DataBase Administrator), quien se encarga del control general del Sistema de Base de Datos.

3.2.3.3. Tipos de Bases de Datos.

Existen diferentes tipos de Base de Datos como: jerárquicas, transaccional, de red, entre otros; sin embargo, las bases de datos relacionales y las orientadas a objetos son las más usadas y con mejores características.

3.2.3.3.1. Base de Datos Orientada a Objetos.

Los datos son almacenados en objetos, tal cual la programación orientada a objetos. ODBMS (Sistema Gestor de Base de Datos Orientada a Objetos) es la integración de una base de datos con los diferentes lenguajes de programación orientados a objetos como Java, C#, Visual Basic .NET, etc. Éstos deben ser usados para sistemas que necesiten un buen rendimiento en manipulación de tipos de datos complejos.

Los ODBMS proporcionan los costes de desarrollo más bajos y el mejor rendimiento gracias a que almacenan objetos en disco y tienen una integración transparente con el programa escrito en un lenguaje de programación orientado a objetos. Esto se consigue al almacenar exactamente el modelo de objeto usado a nivel aplicativo, lo que reduce los costes de desarrollo y mantenimiento.

Estas bases de datos, basadas en lenguajes de programación, presentan su fortaleza en: tipos de datos complejos, integración con diferentes lenguajes, un elevado rendimiento, simplifican la realización de los modelos y consultas.

El ODBMS más relevante es ODMG (Object Database Management Group)

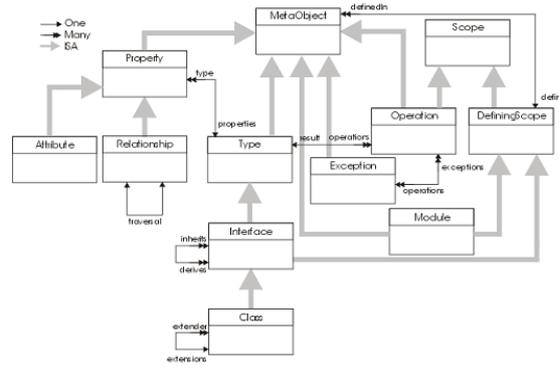


Figura 3.12: Ejemplo de un diseño ODMG.²⁰

3.2.3.3.2. Base de Datos Relacional.

Se construye a través de tablas, registros (líneas), campos (columnas); estas tablas se relacionan por medio de un campo en común en dos o más tablas que tienen las mismas características. A diferencia de una base de datos orientada a objetos, en una relacional existen RDBMS (Sistema de Gestión de Base de Datos Relacional), con el mismo objetivo que la anterior.

Las bases de datos relacionales ya han sido implementadas hace muchos años presentando un buen rendimiento para aplicaciones de negocio o sistemas de información. Pasan por un proceso al que se le conoce como normalización de una base de datos, el cual es entendido como el proceso necesario para que una base de datos sea utilizada de manera óptima.

Las fortalezas de este tipo de base de datos son: tipos de datos sencillos, lenguajes de consulta potentes, protección elevada; su desventaja sale a la luz cuando sus soluciones no tienen nada que ver con datos de negocio, como: imágenes, multimedia o información geográfica, ya que el modelo relacional no es apropiado por no poseer estructuras de datos para soportar este tipo de datos.

²⁰ FUENTE: http://www.jot.fm/issues/issue_2004_11/article5/images/fig2.gif

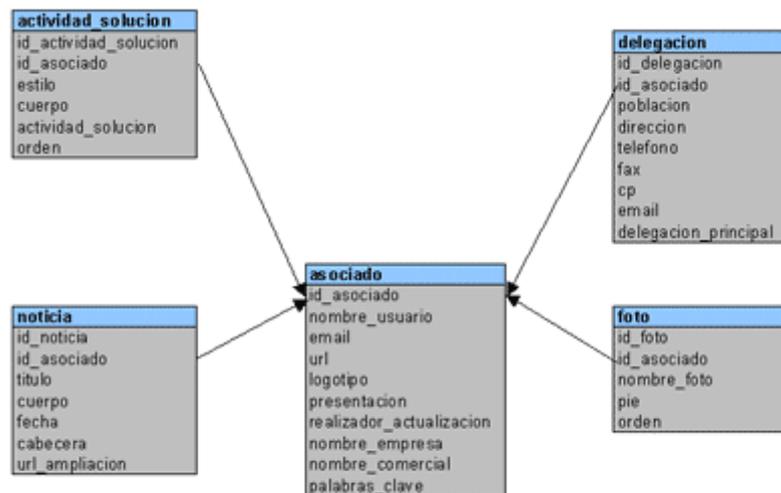


Figura 3.13: Ejemplo de un diseño SQL.²¹

Existe software exclusivamente dedicado a tratar con bases de datos relacionales. Entre los gestores o manejadores más actuales y populares encontramos: MySQL, PostgreSQL, Oracle, DB2 y Microsoft SQL Server.

Estas descripciones son válidas en general, pero hay que tener en cuenta que algunos sistemas de base de datos no respetan estas fronteras. Por ejemplo, algunos sistemas de base de datos orientados a objetos construidos alrededor de lenguajes de programación persistentes se implementan sobre sistemas de base de datos relacionales.

3.2.3.3.3. Base de Datos Objeto-Relacional.

Es una mezcla de las características entre los dos tipos de base de datos vistos anteriormente. ORDBMS (Sistema de Gestión de Base de Datos Objeto Relacional) es su manejador con los mismos objetivos que los anteriores respectivamente. Un ORDBMS soporta una forma de SQL llamada SQL3 y tiene un modelo relacionado porque los datos están guardados en forma de tablas con

²¹ FUENTE: <http://www.desarrolloweb.com/articulos/images/img-presupuesto/asociacion/tablas.gif>

renglones y columnas. Sin embargo, este modelo debe ser modificado para soportar las características de un modelo orientado a objetos. Sus principales características son: extensión de la base de datos, soporte de objetos completos e inherencia. Este modelo permite que los usuarios puedan definir los tipos de datos, funciones y operadores.

Los ORDBMS tienen ventajas como la capacidad de guardar cantidades grandes de información y acceso de alta velocidad. También tiene el apoyo de los vendedores principales.

ORDBMS más relevantes son SQL3 y Oracle8.

3.2.3.3.4. Comparativa de SGBD.

A continuación se presenta un cuadro comparativo, con los lenguajes representativos de cada modelo:

Criterio	RDBMS	ODBMS	ORDBMS
Estándar	SQL 2.	ODMG 2.0.	SQL 3.
Características Orientada a Objetos	No apoya a estas características.	Apoyo extensivo.	Apoyo limitado.
Uso	Es fácil de usar	Bien para programadores; Algún acceso de SQL para usuarios finales	Fácil de usar con excepción de algunas extensiones
Tipos de datos complejos	No apoya tipo de datos abstractos.	Apoya una gran variedad de tipos de datos y datos relacionales complejos.	Apoya tipo de datos abstractos y relaciones complejos.
Desempeño	Tiene un buen desempeño.	Relativamente poco desempeño.	Desempeño muy bueno.
Edad	Relativamente viejo y muy maduro.	Es relativamente maduro.	Todavía se encuentra en proceso de desarrollo.

Continúa en la página 91.

Uso de lenguaje SQL	Apoyo extensivo de SQL.	Su lenguaje es OQL, muy similar a SQL pero con características adicionales con estructuras Orientada a Objetos.	SQL 3 cada vez evoluciona incorporando estructura Orientada a Objetos.
Consultas	Datos simples con consultas.	Complejos datos sin consultas.	Complejos datos con consultas.
Ventajas	Su dependencia de SQL, y su optimización de consultas es relativamente sencilla.	Puede manejar todo tipo de aplicaciones complejas, se puede reusar el código.	Habilidad de consulta, aplicaciones complejas y la habilidad a manejar aplicaciones grandes y complejas.
Desventajas	Se dificulta el mapeo objeto-dato en la base. Inhabilidad de manejar aplicaciones complejas como: multimedia.	Desempeño pobre por la optimización compleja en consultas.	Desempeño pobre en aplicaciones Web.
Mercado	Tiene un mercado extenso y muchos vendedores	Falta el apoyo de los vendedores por el tamaño del mercado de SGBDs.	Tiene un buen futuro. Parece que todos los vendedores de SGBDs quieren este producto.

Tabla 3.4: Comparación de SGBDs²².

Recordamos que este es un gran problema que la aplicación debe abarcar, ya que la empresa no trabaja con una base de datos, solamente maneja la información con hojas de Excel.

TELCOLSAM S.A. a través de los años, ha decidido usar a Excel como su base de datos similar a las funcionalidades de un RDBMS. Se llega a esta conclusión

²² FUENTE: <http://www.acm.org/crossroads/espanol/xrds7-3/ordbms.html>

puesto que los datos que se ha recolectado de la empresa se presentan como tablas (registros y campos).

Por otro lado, el trabajo diario en el Departamento de Cobranzas depende de la información anterior. Nuestro proyecto de titulación abarcará la migración de datos a la nueva base de datos escogida, porque de lo contrario se impediría la continuidad del negocio.

Se ha elegido usar una base de datos relacional para la aplicación, debido a las siguientes razones:

- La información en TELCOLSAM S.A. se encuentra registrada en tablas, tal como una base de datos relacional. El presentar una base de datos relacional simplificaría a gran escala la futura migración que requiere la empresa.
- Una base de datos relacional es muy eficiente al trabajar con datos de negocios, con grandes herramientas que garantizan evitar la duplicidad de registros y garantiza la integridad referencial; así, al eliminar un registro elimina todos los registros relacionados dependientes.
- Por último, el lenguaje para una base de datos relacional ha sido anteriormente estudiado, teniendo sólidas fuentes para realizar un trabajo ágil, flexible y eficiente.

3.2.3.4. Bases de Datos de Código Abierto.

El término **Código abierto** comprende la distribución libre del código fuente, el cual empezó a usarse en 1998 por usuarios de la comunidad de Software Libre, tratando de usarlo como remplazo al original software libre.

En pocas palabras la definición de código abierto representa que “se puede mirar el código fuente”, lo cual es un criterio más débil y flexible que el del software libre; un programa de código abierto puede ser software libre, pero también puede serlo un programa semi-libre o incluso uno completamente propietario.

Los beneficios del código abierto:

- Es factible modificarlo para poder ser adaptado a las necesidades de la empresa.
- Existe software libre sin costo alguno, con características similares al software pagado.
- Ya que toda la información es abierta, de igual manera se encontrarán los estados, errores, etc. Por tanto, estos son rápidos para reaccionar si surgen problemas. La comunidad de usuarios y desarrolladores son una gran ayuda.

Entre las más conocidas y usadas SGBD de código abierto se encuentran MySQL y PostgreSQL; por tanto, las detallaremos a continuación.

3.2.3.4.1. *MySQL.*



MySQL se deriva de dos partes; La primera “My”, referida a la hija del cofundador Monty Widenius, mientras que “SQL” se refiere a un lenguaje de Consulta Estructurado, el cual se encuentra estandarizado en ANSI/ISO SQL.

Su logo es un delfín, y su nombre es “Sakita”, que fue escogido por los fundadores de MySQL AB (David Axmark, Allan Larsson, y Michael Widenius)

Este es un RDBMS en una Arquitectura Cliente–Servidor, detallado en **Título “ARQUITECTURA CLIENTE–SERVIDOR”, página 62.**

Presenta un código escrito en C++ y C, totalmente accesible para usarlo y modificarlo. Existen dos versiones principales: la versión libre, MySQL Community, escrita originalmente por Michael Widenius; y la versión comercial, MySQL Enterprise.

Actualmente ambas distribuciones son desarrolladas por Sun Microsystems.

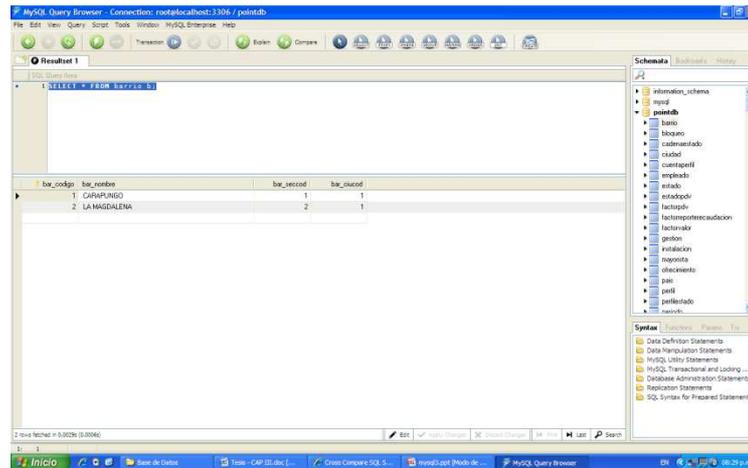


Figura 3.14: MySQL Query Browser.

3.2.3.4.2. PostgreSQL.



PostgreSQL es un sistema de gestión de bases de datos objeto-relacional. Utiliza un modelo Cliente-Servidor y usa multiprocesos, en vez de multihilos, para garantizar la estabilidad del sistema.

PostgreSQL se encuentra a cargo de una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. Esta comunidad toma el nombre de PGDG (Grupo de Desarrollo Global PostgreSQL).

3.2.3.4.3. Firebird.



Firebird es una base de datos relacional que ofrece muchas características de SQL ANSI estándar y que funciona en Linux, Windows, MacOSX y una variedad de plataformas UNIX. Está desarrollado bajo C y C++, basada en código fuente liberado por

Inprise Corp. (ahora conocida como Borland Software Corp.)

Ofrece una concurrencia excelente, alto rendimiento y un poderoso lenguaje de procedimientos almacenados y disparadores.

Es necesario recordar que esta base de datos es usada por TELINEMAX S.A. y en otros casos por TELCOLSAM S.A. por tener vinculación con el programa actual SAR II.

3.2.3.5. Bases de Datos de Código Cerrado.

Por el contrario, el término Código Cerrado, se refiere a que el código fuente es exclusivo de su fabricante, un gran ejemplo es Microsoft®; todos sus códigos fuentes no se encuentran disponibles al público. Por tanto, el software no libre utiliza código cerrado.

La familia de SQL Server, y Oracle son las más importantes SGBD de código cerrado, y serán detalladas a continuación.

3.2.3.5.1. *SQL Server.*



SQL server es un motor de base de datos relacional producido por Microsoft®. El más actual de esta familia es SQL Server 2008 implementa estándares ANSI SQL.

Presenta una gran ventaja al tener una integración directa con Microsoft Visual Studio, el Microsoft Office System y un conjunto de nuevas herramientas de desarrollo, incluido el Business Intelligence Development Studio.



Figura 3.15: Componentes Básicos de SQL Server²³ .

Presenta siete diferentes versiones que son:

- Enterprise Edition (Empresarial).
- Standard Edition (Versión Estándar).
- Workgroup Edition (Para Grupos de Trabajo).
- Express Edition (Edición Personal).
- Compact Edition (Para dispositivos móviles).
- Developer Edition (Para desarrolladores de Software).
- 64-Bit Platform Edition (Para plataformas de 64 Bits).

3.2.3.5.2. Oracle.



Es un sistema de gestión de base de datos relacional desarrollado por Oracle Corporation, considerado como uno de los sistemas más complejos, y además sus últimas versiones han sido certificadas para trabajar con GNU/ Linux.

Oracle es la primera compañía de software que desarrolla e implementa software para empresas cien por ciento activado por Internet a través de toda su línea de

²³ FUENTE: www.uaem.mx/posgrado/mcruz/cursos/miic/sqlserver2.ppt

productos: base de datos, aplicaciones comerciales y herramientas de desarrollo de aplicaciones y soporte de decisiones.

Oracle es el proveedor mundial líder de software para administración de información, y la segunda empresa de software.

Oracle cuenta con 6 ediciones:

- Oracle Database Enterprise Edition (EE).
- Oracle Database Standard Edition (SE).
- Oracle Database Standard Edition One (SE1).
- Oracle Database Express Edition (XE), única versión gratuita.
- Oracle Database Personal Edition (PE).
- Oracle Database Lite Edition (LE).

3.2.3.6. Comparativa de Base de Datos.

Las comparaciones se realizan con las versiones más actuales de cada SGBD, de la siguiente manera:

- **MySQL 5.1.**
- **PostgreSQL/8.1.3.**
- **Microsoft SQL Server 2008.**
- **Oracle 11g** (Sin embargo, la compañía dice que es probable que tarde al menos dos o tres años en comenzar a migrar a la versión 11g desde el programa anterior, 10g. Por tanto, las comparaciones se realizarán en la versión 10g).

3.2.3.6.1. Principales Características de las diferentes Bases de Datos.

SGBD	CARACTERÍSTICAS PRINCIPALES
<p data-bbox="300 913 411 949">MySQL</p> 	<p data-bbox="561 495 804 524">PORTABILIDAD</p> <ul data-bbox="513 533 1410 757" style="list-style-type: none"> • Gran Velocidad y Robustez • Usa tablas en disco B-Tree muy rápidas con compresión de índice. • Multiproceso, es decir puede usar varias CPU si éstas están disponibles. • Puede trabajar en distintas plataformas y S.O. distintos. <p data-bbox="561 792 756 822">SEGURIDAD</p> <ul data-bbox="513 831 1410 902" style="list-style-type: none"> • Tiene un Sistema de Contraseñas y privilegios para conectar con el servidor. <p data-bbox="561 947 823 976">ESCALABILIDAD</p> <ul data-bbox="513 985 1410 1171" style="list-style-type: none"> • Registros de longitud fija y variables. • Diversos tipos de columnas como: enteros de 1, 2, 3, 4, y 8 bytes, coma flotante, doble precisión, carácter, fechas, enumerados, etc. • Todas las columnas pueden tener valores por defecto. <p data-bbox="561 1211 807 1240">CONECTIVIDAD</p> <ul data-bbox="513 1249 1410 1473" style="list-style-type: none"> • Los clientes usan TCP/IP (para cualquier plataforma), en Windows pueden usar “names pipes” y en Unix utilizan “socket unix” para conectarse al servidor. • El servidor soporta mensajes de error en distintos lenguajes (se puede escoger). • Todos los comandos tienen –help para ayudas.
<p data-bbox="261 1809 450 1845">PostgreSQL</p> 	<p data-bbox="561 1559 922 1588">ALTA CONCURRENCIA</p> <ul data-bbox="513 1597 1410 1704" style="list-style-type: none"> • Permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. <p data-bbox="561 1749 823 1778">ESCALABILIDAD</p> <ul data-bbox="513 1787 1410 2011" style="list-style-type: none"> • Los usuarios pueden crear sus propios tipos de datos, los que pueden ser por completo indexables gracias a la infraestructura GiST de PostgreSQL. • Tiene gran variedad de tipos nativos de datos: Precisión binaria, texto largo ilimitado, direcciones IP, dirección MAC, vector, figuras geométricas, etc.

Continúa en la página 99.

	<ul style="list-style-type: none"> • Permite crear una amplia funcionalidad a través de su sistema de activación de disparadores (<i>triggers</i>). <p>LENGUAJE SQL COMPRENSIVO</p> <ul style="list-style-type: none"> • PostgreSQL soporta el estándar SQL99 e incluye características avanzadas tales como las uniones (<i>joins</i>). <p>API FLEXIBLE</p> <ul style="list-style-type: none"> • La flexibilidad del API de PostgreSQL ha permitido a los vendedores proporcionar soporte al desarrollo fácilmente para el RDBMS PostgreSQL.
<p>SQL Server</p> 	<p>FÁCIL DE INSTALAR Y CONFIGURAR</p> <ul style="list-style-type: none"> • Rápida descarga (Express Edition). • Simple interfaz de Configuración. • Instalación sin intervención del usuario. <p>FÁCIL DE UTILIZAR Y ADMINISTRAR</p> <ul style="list-style-type: none"> • Herramienta nueva y ligera para la administración y edición de consultas. • Optimizador avanzado de consultas. • Ajuste automatizado. • Asistentes para las tareas más comunes. • Administrador de equipo para iniciar y detener el servicio. <p>SEGURIDAD</p> <ul style="list-style-type: none"> • Tres niveles de seguridad de acceso al código: seguro, acceso externo, no seguro. • Compatibilidad con Active Directory • Compatibilidad con la autenticación de Windows. <p>COMPATIBILIDAD CON XML</p> <ul style="list-style-type: none"> • Tipo XML nativo con datos estructurados y semiestructurados. • Compatibilidad con XQuery. • Compatibilidad con esquemas XML. <p>INTEGRACIÓN CON VISUAL STUDIO</p> <ul style="list-style-type: none"> • Depuración integrada. • Se instala con Visual Studio. • Explorador de datos de Visual Studio para diseñar esquemas, agregar datos y realizar consultas en bases de datos locales.

Continúa en la página 100.

<p style="text-align: center;">Oracle</p> 	<p style="text-align: center;">FÁCIL DE UTILIZAR Y ADMINISTRAR</p> <ul style="list-style-type: none"> • Posee herramientas de administración gráficas que son mucho más intuitiva y cómoda de utilizar. • Ayuda a analizar datos y efectuar recomendaciones concernientes a mejorar el rendimiento y la eficiencia en el manejo de aquellos datos que se encuentran almacenados. • Apoya en el diseño y optimización de modelos de datos. • Asiste a desarrolladores con sus conocimientos de SQL y de construcción de procedimientos almacenados y triggers, entre otros. • Apoya en la definición de estándares de diseño y nomenclatura de objetos. • Documenta y mantiene un registro periódico de las mantenciones, actualizaciones de hardware y software, cambios en las aplicaciones y, en general, todos aquellos eventos relacionados con cambios en el entorno de utilización de una base de datos. • Es una base de datos muy orientada a Internet.
--	---

Tabla 3.5: Detalle de Características Principales de los SGBD.

3.2.3.6.2. Ventajas y Desventajas.

A continuación se presentan las principales ventajas de cada gestor de base de datos.

SGBD	VENTAJAS
<p style="text-align: center;">MySQL</p> 	<ul style="list-style-type: none"> • MySQL tiene gran penetración en el mercado en los servidores de datos orientados a <i>WEB</i>. • MySQL se distribuye bajo licencia GNU/GPL, lo que lo hace en ciertas circunstancias gratuito. • MySQL fue diseñado para acceso rápido a la información, por lo cual usa un sistema de archivo de almacenamiento plano, que aumenta la eficiencia de la lectura. • Brinda un sistema de estructuración de información (tablas, columnas, índices, etc.) y una interfaz SQL para poder acceder a ello. • Disponible para muchos Sistemas Operativos.

Continúa en la página 101.

	<ul style="list-style-type: none"> • MySQL, ante la ventaja de tener una velocidad de respuesta rápida, incorpora elementos que le faltaban: <ul style="list-style-type: none"> ◦ InnoDB - para añadir integridad referencial y transaccionalidad. ◦ Librerías API - para usarlas como paquetes embebidos en las aplicaciones. • MySQL es recomendable para proyectos no demasiado grandes y las características faltantes al MySQL se pueden programar de manera individual según requerimientos del cliente.
<p style="text-align: center;">PostgreSQL</p> 	<ul style="list-style-type: none"> • Tiene una alta concurrencia, mediante un sistema denominado MVCC (Acceso Concurrente Multi-Versión) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. • Con PostgreSQL, no hay problemas con demandas por software ilícito, puesto que no hay costo asociado a la licencia del software. • Tiene una importante comunidad de profesionales y entusiastas de PostgreSQL de los que su compañía puede obtener beneficios y contribuir. • En contraste a muchos sistemas de bases de datos comerciales, es extremadamente común que compañías reporten que PostgreSQL nunca ha presentado caídas en varios años de operación de alta actividad. • El código fuente está disponible para todos sin costo. • PostgreSQL está disponible en casi cualquier Unix (34 plataformas en la última versión estable), y una versión nativa de Windows está actualmente en estado beta de pruebas. • Existen varias herramientas gráficas de alta calidad para administrar las bases de datos (pgAdmin, pgAccess) y para hacer diseño de bases de datos (Tora, Data Architect). • PostgreSQL implementa un subconjunto extendido de los estándares SQL92 y SQL99. • PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos (Primary Key, Foreign key). • PostgreSQL implementa los tipos de datos definidos para el estándar SQL3 y aumenta algunos otros.

Continúa en la página 102.

<p style="text-align: center;">SQL Server</p> 	<ul style="list-style-type: none"> • Se mejora la protección de la información, con capacidades de auditoría completas, cifrado de datos transparente y configuración para minimizar las amenazas de seguridad al habilitar sólo los servicios que necesita. • SQL Server 2008 ofrece una recopilación más amplia de datos de rendimiento, un repositorio centralizado de datos para almacenar los datos de rendimiento y una compresión de datos mejorada que le permite almacenar más datos de manera eficaz. • SQL Server 2008 ofrece un rendimiento mejorado de las consultas y un almacenamiento de datos eficiente y rentable que permite administrar y redimensionar grandes cantidades de usuarios y de datos. • Con SQL Server 2008 los usuarios pueden crear informes complejos con Microsoft® Office Word y Microsoft® Office Excel y compartirlos de manera interna y externa. • Cuando SQL Server no tiene tareas de usuario para procesar, comienza a escribir automáticamente las memorias intermedias sucias del caché al disco, dando las siguientes ventajas: <ul style="list-style-type: none"> ○ Mejor utilización de la CPU. ○ Menor necesidad de limpieza de las memorias intermedias durante el procesamiento de las transacciones. ○ Puntos de verificación más rápidos. ○ Menor tiempo de recuperación.
<p style="text-align: center;">Oracle</p> 	<ul style="list-style-type: none"> • Su transportabilidad: corre, hoy en día, sobre decenas de plataformas: Linux, Windows, HP UX, Solaris, etc. • La potencia de sus instrumentos de desarrollo de aplicaciones. • La riqueza de su diccionario de datos. • Los mecanismos encargados de la seguridad y la confidencialidad. • Una experiencia probada sobre el terreno y una buena presencia de Oracle a nivel de formación, consejo y soporte técnico. • Oracle Database 11g ofrece nuevas funcionalidades para mejorar los tiempos de respuesta de las aplicaciones. Con Connection Pooling y Query Results Cache los resultados de las consultas (query) más utilizadas son almacenadas y reutilizadas de manera transparente a las aplicaciones, mejorando los tiempos de respuesta y haciendo un uso más efectivo de los recursos hardware disponibles. • Oracle Database 11g incorpora un nuevo compilador Java just-in-time que proporciona un alto rendimiento en la ejecución de procedimientos almacenados en Java sin necesidad de utilizar un compilador de terceros.

Continúa en la página 103.

	<ul style="list-style-type: none"> • También se incorpora una integración nativa con Visual Studio 2005 para facilitar el desarrollo de aplicaciones .NET con Oracle. • Oracle es la base de datos con mas orientación hacia INTERNET
--	---

Tabla 3.6: Ventajas de SGBD's.

A continuación se presentan las desventajas más relevantes de cada gestor de base de datos.

SGBD	DESVENTAJAS
<p data-bbox="300 1070 411 1104">MySQL</p> 	<ul style="list-style-type: none"> • MySQL fue creado con la intención de ser un gestor rápido accediendo a la información, pero debido a su difusión y aceptación, MySQL empezó a ser un producto adoptado por diversos ámbitos para los cuales ni fue diseñado, ni está preparado. • El motor MyISAM, que es el más rápido para consultas, no tiene incorporado en su estructura ningún tipo de "integridad referencial". InnoDB posee esta característica, pero es más lento que MyISAM. • MyISAM no está diseñado para ser un almacén de datos muy cambiantes en el tiempo, por ejemplo: un sitio de publicación de una revista diariamente.
<p data-bbox="261 1626 448 1659">PostgreSQL</p> 	<ul style="list-style-type: none"> • PostgreSQL es muy sólido y soporta transacciones, pero es mucho más lento que otros SGBD. En comparación con su principal competidor es de 2 a 3 veces más lento que MySQL. • Consume gran cantidad de recursos. • Tiene un límite de 8K por fila, aunque se puede aumentar a 32K, con una disminución considerable del rendimiento. • No soporta la construcción de procedimientos basados en web. • La comunidad opina que no funciona muy bien bajo la plataforma Windows. • No cumple la totalidad de los estándares SQL 92, y SQL 99.

Continúa en la página 104.

<p>SQL Server</p> 	<ul style="list-style-type: none"> • La principal desventaja de Microsoft SQL SERVER es la enorme cantidad de memoria RAM que utiliza para la instalación y utilización del software. • Se lo emplea para practicar (edición Express), no es útil, porque tiene demasiadas restricciones. • Comparado con su principal competidor de código cerrado (Oracle), la relación calidad-precio está por debajo. • Presentan deficiencias con datos gráficos, multimedia, CAD y sistemas de información geográfica. • No se manipulan de forma manejable los bloques de texto como tipo de dato.
<p>Oracle</p> 	<ul style="list-style-type: none"> • El mayor inconveniente de Oracle es quizás su precio. Incluso las licencias de Personal Oracle son excesivamente caras. • Otro problema es la necesidad de ajustes. Un error frecuente consiste en pensar que basta instalar el Oracle en un servidor y enchufar directamente las aplicaciones clientes. Un Oracle mal configurado puede ser desesperantemente lento. • También es elevado el coste de la formación, y sólo últimamente han comenzado a aparecer buenos libros sobre asuntos técnicos distintos de la simple instalación y administración.

Tabla 3.7: Desventajas de SGBD's.

Hemos visto las características más relevantes de cada gestor de Base de Datos, sin embargo para las siguientes comparaciones descartaremos Oracle como posible SGBD para nuestra aplicación por las siguientes razones:

Oracle se considera uno de los gestores con más renombre en el mercado por sus grandes aplicaciones a nivel mundial; sin embargo, dadas las características de la empresa, o sea, necesitando una aplicación relativamente pequeña (solamente cinco terminales simultáneos en el peor de los casos), creemos que sería demasiado el empleo de Oracle.

Por otro lado Oracle aunque es muy potente, es una base de datos extremadamente cara y además es más orientada a aplicaciones WEB.

3.2.3.7. Elección del SGBD para TELCOLSAM S.A.

Vistas las principales ventajas y desventajas entre las posibles opciones de un gestor de base de datos a emplear, hemos decidido usar MySQL 5.1.

- Una de las razones más importantes es que mantiene una licencia gratuita.
- Descartamos PostgreSQL por dos razones:

- Velocidad: El gran problema en TELCOLSAM S.A. es el trabajo diario con hojas de cálculo, lo cual consume mucho tiempo. TELCOLSAM S.A. desea agilizar el proceso para un objetivo primordial “Toma de Decisiones”.

Deseamos que los usuarios finales se sientan a gusto con la aplicación a realizar, que sea rápida y eficiente, por tanto nos inclinamos por esta opción.

- CONSUMO DE RECURSOS: Como pudimos ver en el **Título “Terminales de usuarios por departamentos.”, página 49**. Los equipos actuales no tienen características elevadas, por tanto, debemos cuidar los recursos de los terminales para no enfrentarnos a que estos se cuelguen. MySQL intenta cuidar los recursos del sistema de la mejor manera, aún en aplicaciones Clientes que implementen su API.
- Las utilidades de administración de este gestor son envidiables para muchos de los gestores comerciales existentes, debido a su gran facilidad de configuración e instalación.
- Tiene una probabilidad muy reducida de corromper los datos, incluso en los casos en los que los errores no se produzcan en el propio gestor, sino en el sistema en el que está.
- MySQL presenta muchas ventajas, que harán que nuestro trabajo sea rápido y potente: gracias a su agilidad en consultas, transacciones y demás, adicionada a la poca carga de trabajo que soportará la BDD.

- Por otro lado, MySQL a través de un conector no tiene ningún problema para la comunicación TCP/IP con el IDE Visual Studio 2008, el cual ya se definió que será empleado para la construcción de las dos capas (Vista – reglas del negocio).
- Y por último, MySQL tienen un lenguaje sencillo y potente, en el cual ya tenemos experiencia.

En el siguiente cuadro consta que el gestor de bases escogido (MySQL 5.1) cumple a cabalidad con las necesidades de la empresa.

MYSQL 5.1 EN TELCOLSAM S.A.	
Concurrencia	<p>Según los requerimientos de TELCOLSAM S.A. tenemos que la carga total actualmente del sistema es de un máximo de cinco personas. Suponiendo que estas cinco personas harán empleo de la base de datos sea para: búsquedas, ingresos, eliminaciones, actualizaciones, etc., la concurrencia será baja.</p> <p>MySQL presenta buena respuesta a concurrencia puesto que mientras está leyendo los datos de una tabla, éste la bloquea, de tal manera que nadie más puede escribir en ella, aunque si pueden leerla.</p> <p>Cuando MySQL está actualizando o eliminando filas de una tabla, éste la bloquea, y por lo tanto nadie puede al menos leerla, pero la poca concurrencia minimiza este inconveniente.</p>
Respaldos	<p>Actualmente, existen una serie de comandos que permite hacer una copia de seguridad de una o múltiples bases de datos.</p> <p>Hay varias formas de realizar un “backup” en MySQL, desde la línea de comandos (usando el comando mysqldump) o usando una aplicación externa como MySQL Administrator.</p> <p>De esta manera, se tendrá siempre un respaldo de todos los datos que se hayan ingresado, brindando seguridad e integridad.</p>

Continúa en la página 107.

<p>Procedimientos almacenados , disparadores y funciones</p>	<p>Anteriormente MySQL no soportaba el empleo de Procedimientos Almacenados, sin embargo desde la versión 5.0 ya lo hacen. Se empleará procedimientos almacenados y funciones en TELCOLSAM S.A. para las acciones de: ingreso, búsqueda, eliminación, actualización. Los procedimientos almacenados mejoran el rendimiento ya que se pre-compilan y se ejecutan más rápido.</p>
<p>Compatibilidad con .NET</p>	<p>Ya que la Presentación y las Reglas del Negocio se programaran en el IDE Visual Studio 2008, es de real importancia la entera compatibilidad. El Connector/Net de MySQL permite a los desarrolladores crear fácilmente aplicaciones .NET que requieren seguridad, alto rendimiento y conectividad con MySQL. Este conector implementa las interfaces requeridas de ADO.NET y se integra dentro de sus herramientas. Los desarrolladores pueden construir aplicaciones para MySQL usando el Connector/Net en lenguajes .NET, tal como Visual Basic .NET, C++ o C#.</p>
<p>Soporte y Documentación</p>	<p>Al ser MySQL un Gestor de Base de Datos de licenciamiento libre, existe mucha documentación y ayuda de la comunidad para la solución de problemas e inseguridades que se presenten. Si se llegará a tener alguna dificultad, se podrá recurrir a cualquiera de éstos para solucionarlos.</p>
<p>Licenciamiento</p>	<p>Esta es una gran ventaja, MySQL tiene un licenciamiento gratuito. Siempre se tratará que la empresa tenga el menor gasto posible. Además, se toma en cuenta que el pago por una solución pagada a TELCOLSAM S.A. le representaría un gasto que no desea.</p>
<p>Cuentas de Usuario</p>	<p>Es importante tener una buena administración de la base de datos brindando una mayor seguridad a los datos de la base de datos. Las cuentas de usuario están ligadas a los diferentes niveles de permisos.</p>

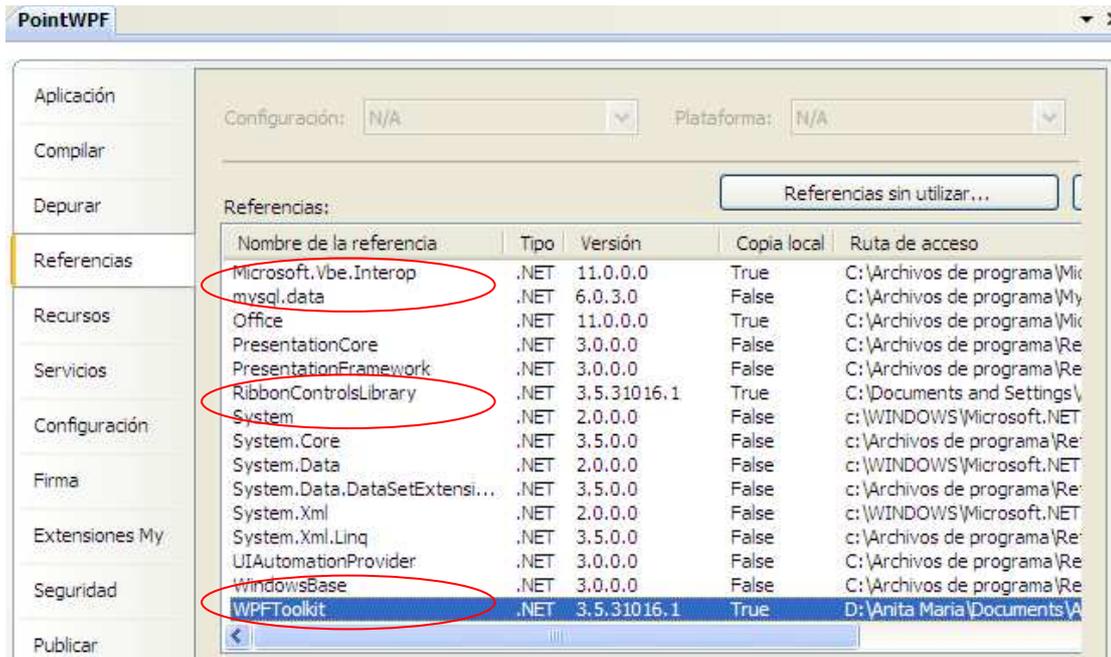
Continúa en la página 108.

<p>Bloqueos en el nivel de Usuario</p>	<p>Los bloqueos por Usuarios pueden ser a distintos niveles: Global, Tabla, Columna y Fila. Sin embargo MySQL permite realizar un bloqueo a nivel de "String", haciendo una comparación con éste y permitiendo o denegando el acceso.</p>
<p>Seguridad</p>	<p>Almacenar contraseñas en MySQL como texto plano (sin encriptar) nunca será una buena idea. Como buenos administradores de bases de datos debemos tener cuidado de proteger la información de nuestros usuarios. Afortunadamente, MySQL provee diversas opciones para proteger contraseñas.</p> <p>Es posible utilizar SHA1 para almacenar únicamente los hashes de las contraseñas, ya que es una función no reversible.</p>
<p>Integridad de Datos</p>	<p>TELCOLSAM S.A. requiere que los reportes generados muestren datos reales y actuales, por tanto, la integridad pasa a ser un requisito para la empresa.</p> <p>Podemos decir de manera simple que integridad referencial significa que cuando un registro en una tabla haga referencia a un registro en otra tabla, el registro correspondiente debe existir</p> <p>MySQL presenta un gran número de herramientas para garantizar la integridad de los datos como: claves primarias y foráneas, claves únicas, etc.</p>

Tabla 3.8: MySQL en TELCOLSAM S.A.

3.2.4. OTRAS HERRAMIENTAS A UTILIZAR.

Para cumplir con los requerimientos de TELCOLSAM S.A. y no incurrir en la programación de herramientas ya existente, es necesaria la inclusión de APIs de terceros para utilizar las funcionalidades que éstas presentan. Igualmente, debido a los requerimientos de WPF, es indispensable el uso del .NET Framework para su desarrollo, y su versión adecuada será determinada en este apartado.



3.2.4.1. Conector MySQL/.NET.

Como hemos podido observar, para la interconexión de la Capa de Datos (MySQL 5.1), con las Reglas del Negocio (Visual Studio 2008) es necesario la utilización de un conector.

El Connector/Net de MySQL incluye soporte completo para:

- Características de MySQL desde la versión 5.0 (procedimientos almacenados, Etc.)
- Características de MySQL 4.1 (Sentencias preparadas del lado del servidor, Unicode y Acceso a memoria compartida, etc.)
- Soporte de paquetes grandes para enviar y recibir filas y datos de más de 2 gigabytes de tamaño.
- Soporte para conectar usando sockets TCP/IP, tuberías, o memoria compartida en Windows.
- Soporte de conectividad usando sockets TCP/IP o sockets Unix.
- Soporte con el Framework .NET de código abierto desarrollado por Novell.
- Completamente manejable, no utiliza las librerías cliente de MySQL.

Funciona en cualquier plataforma que soporte el Framework .NET, principalmente en versiones recientes de Microsoft Windows, y es soportado en Linux.

Puede ser instalado a través de un paquete instalador de Windows (.msi), o bien, a través de un sencillo archivo ZIP, y contiene varias clases que son usadas para conectar a la base de datos, ejecutar consultas y sentencias y manejar resultados de consultas.

Las siguientes son las principales clases del Connector/Net:

Clase	Descripción
MySqlCommand	Representa una sentencia SQL a ejecutar sobre una base de datos MySQL.
MySqlCommandBuilder	Genera automáticamente comandos de tabla sencilla usados para aplicar los cambios hechos a un dataset con la base de datos asociada.
MySqlConnection	Representa una conexión a un servidor de base de datos.
MySqlDataAdapter	Representa un conjunto de comandos y una conexión que son usados para llenar un dataset y actualizar una base de datos.
MySqlDataReader	Provee un medio de lectura de filas de datos únicamente hacia adelante.
MySqlException	La excepción que es lanzada cuando MySQL devuelve un error.
MySqlHelper	Clase de ayuda que hace más fácil trabajar con el proveedor.
MySqlTransaction	Representa una transacción SQL que se hará en una base de datos.

Tabla 3.9: Clases del Conector .NET²⁴.

²⁴ FUENTE: <http://www.mysql-hispano.org/page.php?id=41&pag=3>

La versión más reciente de este conector es Connector/NET 6.2, sin embargo este se encuentra en estado “beta”, lo que significaría que podría existir algún problema con el mismo. Es por esta razón que se ha decidido usar la versión anterior a esta la cual presenta las siguientes características²⁵:

Connector .NET 6.1.3:

- Soporte para MySQL 4.1, MySQL 5.0, MySQL 5.1, MySQL 5.4.
- Nueva arquitectura para cuentas de usuarios.
- Seguridad con Certificados SSL.
- MySQL Website Configuration Tool.

Con esto se determina el **TBD-2 “Para la Base de Datos”** del **Título “Interfaces con el software., página 32.**

3.2.4.2. Ribbon.

Ribbon es una herramienta creada por Microsoft en octubre del 2008, es un elemento de las interfaces gráficas que consta de una franja con múltiples funciones que el programa puede realizar. Por lo general, los ribbons suelen ubicarse en la parte superior de la ventana, aunque hay programas en donde se permite personalizar su ubicación y apariencia.

El objetivo del ribbon es ubicar todas las funcionalidades de una aplicación en un solo lugar para que sean fáciles de alcanzar.

Microsoft implementó ribbons primeramente en Microsoft Office 2007, aunque también es incluida actualmente en Paint, WordPad de Windows 7, Windows Live Movie Maker, y Microsoft Office 2010. Por ejemplo, Microsoft Office 2007 emplea un ribbon llamado "Office Fluent Ribbon", que reemplaza los menús y las barras de herramientas.

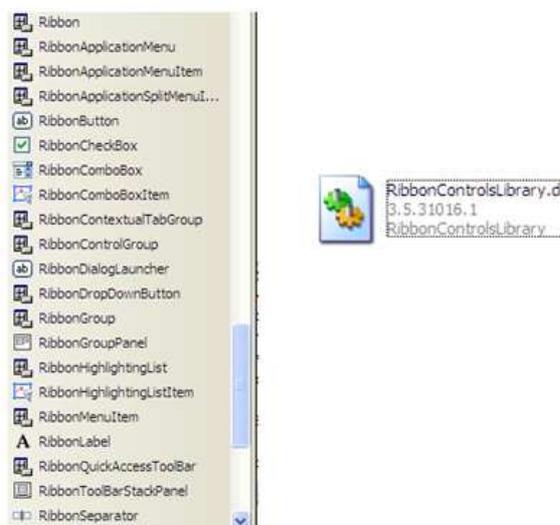
25 FUENTE: <http://dev.mysql.com/doc/refman/5.1/en/connector-net-versions.html>

En POINTSOFT, como menú, se utilizará un control Ribbon, agrupando las funcionalidades en un RibbonTab. De esta manera, cada vez que se agregue más funcionalidad a la aplicación, más módulos, estos irán enriqueciendo las opciones del menú presentadas en el Ribbon.

Por defecto, el IDE utilizado (Visual Studio 2008) no presenta este nuevo control; sin embargo, puede ser descargado bajo una licencia gratuita por Internet. Esta es una librería llamada **RibbonControlsLibrary.dll** la cual es agregada al proyecto mediante referencia.

Esta librería depende del .NET Framework 3.5 Service Pack 1 o posteriores, por lo que la versión 3.5 SP1 será obligatoria en los terminales.

Con esto se determina el **TBD-1 “Para el Sistema Operativo” del Título “Interfaces con el software., página 32.**

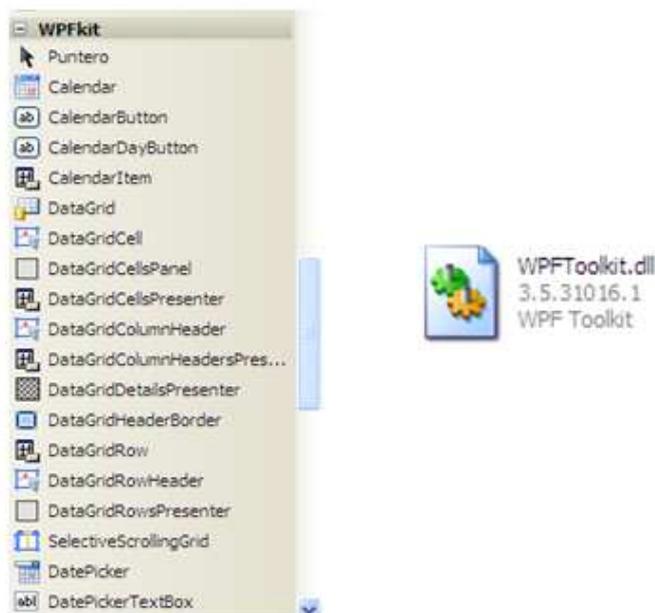


VER ANEXO 2.

3.2.4.3. WPFToolkit.

Esta es una librería que presenta un conjunto de controles que han sido diseñados fuera del normal .NET Framework, de la misma manera que la librería

anterior **RibbonControlsLibrary.dll**. Esta librería, al brindar nuevas funcionalidades con nuevos controles, permite un trabajo más rápido y eficiente.



3.2.4.4. Microsoft.Office.Interop.Excel.

Recalcando las necesidades de TELCOLSAM S.A., los reportes serán migrados a Excel, puesto que los empleados actualmente realizan el trabajo bajo este programa. Por tal motivo, tienen una alta familiaridad con el mismo.

Para poder hacer uso de Excel desde POINTSOFT es necesaria la agregación de una nueva librería llamada: **Microsoft.Office.Interop.Excel**

Como los datos de un documento de Excel están muy estructurados, el modelo de objetos de la librería es jerárquico y sencillo.

Para tener acceso a esta librería, procedimos a descargarnos **Redistributable Primary Interop Assemblies** el cual tiene los argumentos necesarios para trabajar con Office 2003 o a su vez con Office 2007:

- Microsoft Office Access 2007
- Microsoft Office Excel 2007

- Microsoft Office InfoPath 2007
- Microsoft Office Outlook 2007
- Microsoft Office PowerPoint 2007
- Microsoft Office Project 2007
- Microsoft Office Publisher 2007
- Microsoft Office SharePoint Designer 2007
- Microsoft Office Visio 2007
- Microsoft Office Word 2007

Esta API requiere los siguientes parámetros:

- **Sistemas Operativos Soportados:** Windows 2000 Service Pack 4; Windows Server 2003; Windows XP Service Pack 2.
- **Microsoft .Net Framework 1.1** (o superiores).²⁶

Con esto se determina el **TBD-3 “Para los Reportes”** del **Título “Interfaces con el software., página 32.**

²⁶ FUENTE: <http://www.microsoft.com/downloads/details.aspx?FamilyID=59daebaa-bed4-4282-a28c-b864d8bfa513&displaylang=en>

3.2.5. RESUMEN DE LAS HERRAMIENTAS SELECCIONADAS.

A continuación presentamos un gráfico representativo de la estructura final de la aplicación a realizar.

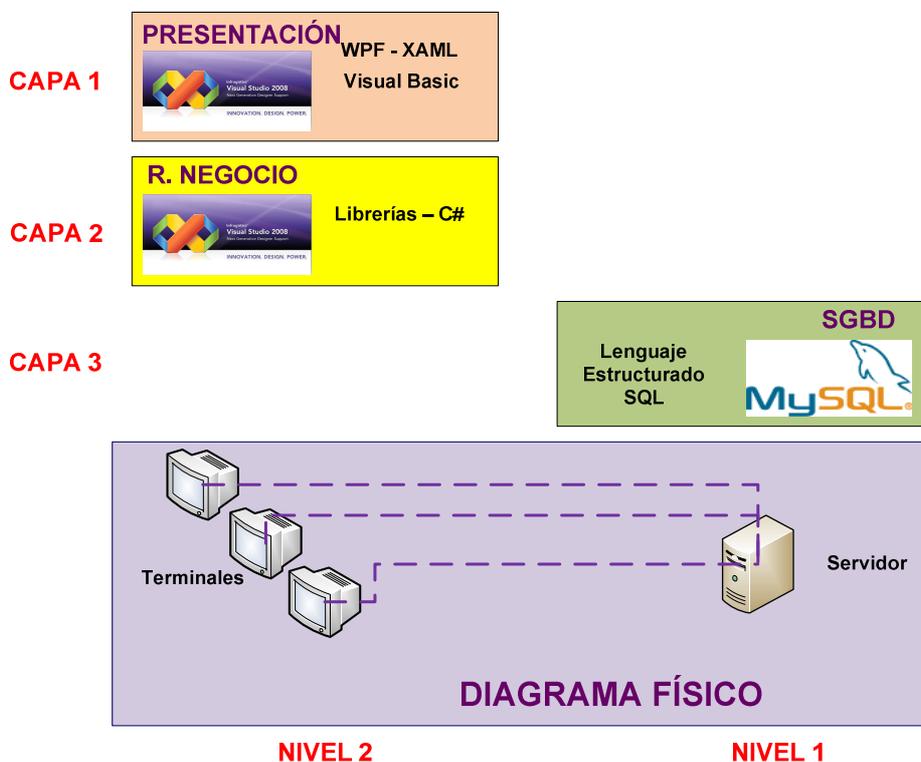


Figura 3.16: Arquitectura de POINTSOFT.

POINTSOF se constituye en un programa Cliente-Servidor junto con una arquitectura en tres capas; Inicialmente se hace constancia de los lugares físicos donde se plantará las diferentes capas:

- **Terminales:** Presentación (Primera capa) y Reglas del Negocio (Segunda capa).
- **Servidor:** Sistema de Gestor de Base de Datos (Tercera capa).

Se ha escogido usar el Servidor **SERVER** de TELCOLSAM S.A., puesto que cumple los requisitos de hardware y software del SGBD a utilizar (MySQL 5.1), y a su vez el dispositivo es el más potente en la empresa.

Por otra parte, se obtuvo la previa autorización del Gerente Técnico para la instalación del SGBD en dicho dispositivo.

La primera capa correspondiente a **PRESENTACIÓN**, es la interfaz que interactuará con el usuario final, situados en cada uno de los terminales de cada departamento. Esta será programada con ayuda del API Microsoft Visual Studio 2008, y dentro del mismo, se encuentra el foco de nuestro trabajo: WPF con un lenguaje de marca llamado XAML y por otro lado, la programación de eventos en el lenguaje subyacente Visual Basic .NET.

La segunda capa es llamada **REGLAS DEL NEGOCIO**, encargada de la implantación de las normas a cumplir en el programa para un correcto funcionamiento. De la misma manera, éste será programada con el API Microsoft Visual Studio 2008, en uno de los lenguajes soportados más potentes: C#.

La tercera y última capa es la llamada **CAPA DE DATOS**, que contiene el **SGBD**, el cual como se muestra en la figura, se mantendrá físicamente en el Servidor SERVER. Este Gestor será programado con un lenguaje SQL (Lenguaje de Consultas Estructurado), para MySQL 5.1.

CAPITULO 4



4. DISEÑO E IMPLANTACIÓN DEL SISTEMA.

4.1. METODOLOGÍA DE DESARROLLO DE SOFTWARE.

Una metodología de desarrollo de software es un plantilla para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información, definiendo **Quién** debe hacer **Qué**, **Cuándo** y **Cómo** debe hacerlo.



El objetivo del proyecto es proporcionar un programa modular y flexible que agilice la gestión en el Departamento de Cobranzas y, para esto, debemos emplear una metodología de desarrollo de software. La metodología que se empleará usa UML (Lenguaje Unificado de Modelado) ORIENTADA A OBJETOS (OO) diseñado por Grady Booch, también conocido como Análisis y Diseño Orientado a Objetos (OOAD).

La metodología OO ha dado un gran avance en la forma de diseño, desarrollo y mantenimiento de software, por lo que hemos decidido tomar esta metodología como referencia por dos razones de peso:

- Brinda la solución al problema de que el código no podía ser reusado ni modificado, o sea la falta de portabilidad de código.

El programa a realizar se enfoca en la solución a problemas del Departamento de Cobranzas, actualmente. Sin embargo, las condiciones ya evaluadas pueden cambiar a lo largo del tiempo, por condiciones propias de legislatura o del mercado, por lo que el sistema debe ser realizado necesariamente de manera modular; dado que si en un futuro, TELCOLSAM S.A. requiere la incorporación de nuevas funcionalidades o módulos, éstos deben ser incorporados sin necesidad de desechar el programa anterior.

La metodología OO nos permite realizar un programa totalmente modular y por consiguiente la reutilización y modificación del código.

- La característica principal del proyecto de titulación radica en la utilización de la nueva herramienta WPF (Windows Presentation Foundation) la cual se encuentra basado en clases (u objetos)²⁷.

Por tales razones, escogemos la metodología llamada **RUP (Rational Unified Process)**, la cual no solamente nos une con UML sino presenta una guía para todo el proceso de desarrollo (diseño e implementación):

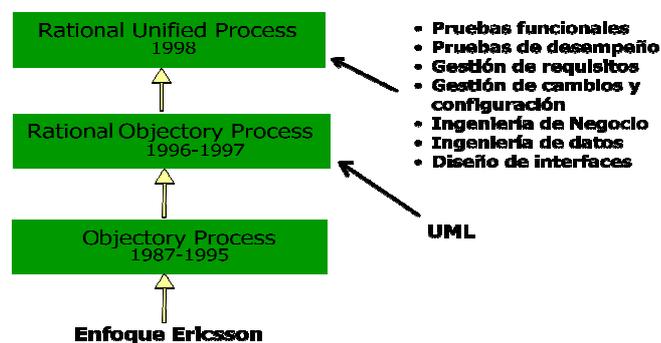


Figura 4.1: Aproximación UML.

Este es un proceso de ingeniería de Software muy bien estructurado, que a la vez permite que la aplicación se adapte a las necesidades y características específicas de la empresa.

Ahora, definiremos esta metodología con más de detalle:

4.1.1. UML (LENGUAJE DE MODELADO UNIFICADO).

“Para la mayoría de las personas, la forma de pensar OO es más natural que las técnicas del análisis y diseño estructurado. Después de todo, el mundo está formado por objetos. Comenzamos a aprender sobre ellos en la infancia y descubrimos que tienen determinados tipos de comportamiento. Si se piensa un poco, esto tiene sentido. Desde una etapa muy temprana categorizamos los

²⁷ FUENTE: <http://www.cristalab.com/tutoriales/programacion-orientada-a-objetos-en-visual-basic-.net-c273/>

objetos y descubrimos su comportamiento. Los usuarios finales y las personas de las empresas piensan de manera natural en términos de objetos, eventos y mecanismos de activación. Podemos crear diagramas OO que les parezcan familiares, mientras tengan dificultades con los diagramas de relación entre entes, tablas de estructura y diagramas de flujo de datos.”²⁸

A continuación, se presenta un cuadro detallando las principales ventajas al usar una metodología basada en Objetos, y a su vez, sustentado nuestra elección de esta metodología:

FACTOR	DETALLE
Reutilización de Código	El Código mediante objetos y clases permite la reutilización de código.
Estabilidad	Las clases al ser reutilizadas hacen que el programa se comporte de manera más estable.
Complejidad	Términos como encapsulamiento, herencia, polimorfismo hace que el diseño de algunas clases resulte más sencillo.
Calidad	Los diseños suelen tener mejor calidad, puesto que se construyen a partir de componentes probados, que han sido verificados y pulidos varias veces.
Sencillez	Los programas se elaboran a partir de piezas pequeñas las cuales ayudan en gran magnitud al desarrollo del mismo.
Mantenimiento	Cada clase efectúa sus funciones independientemente de las demás. Por lo cual los errores serán de fácil ubicación. Los objetos al ser independientes, varios programadores pueden corregir errores al mismo tiempo.
Tiempo	Al poder reutilizar el código es un ahorro significativo de tiempo.
Independencia	Las clases están diseñadas para ser independientes del ambiente de plataformas, hardware y software.

Tabla 4.1: Ventajas de una Metodología Orientada a Objetos.

²⁸ FUENTE: MARTIN- J. J. ODELL

UML es un estándar que nace en 1994, el cual permite especificar, visualizar y construir los artefactos de los sistemas de Software. Actualmente esta metodología agrupa varios enfoques correspondientes a distintos desarrolladores:

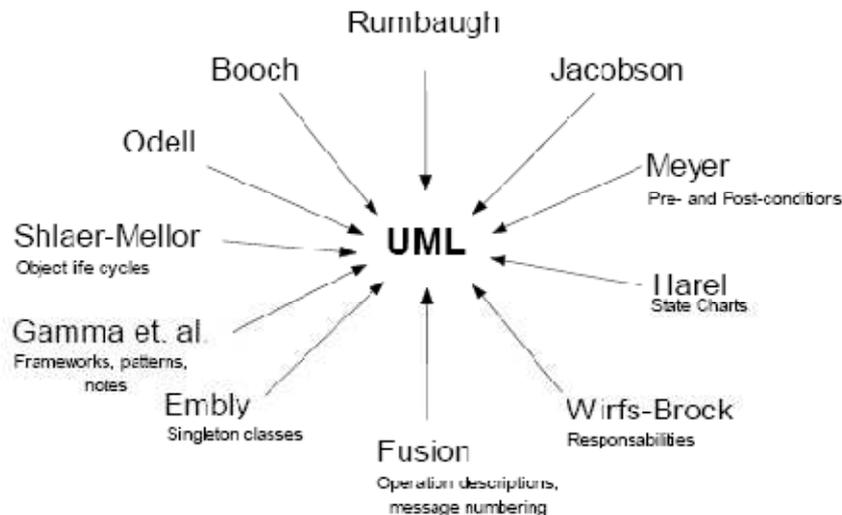


Figura 4.2: Principales aportes a UML.

4.1.1.1. Definiciones en UML.

- **Modelo:** captura una vista de un sistema del mundo real. Es una abstracción de dicho sistema, considerando un cierto propósito. Así, el modelo describe completamente aquellos aspectos del sistema que son relevantes al propósito del modelo, y a un apropiado nivel de detalle.
- **Diagrama:** una representación gráfica de una colección de elementos de modelado, a menudo dibujada como un grafico con vértices conectados por arcos.
- **Paquete:** Ofrece un mecanismo general para la organización de los modelos/subsistemas agrupando elementos de modelado. Un paquete puede tener sub-paquetes sin límite de anidamiento.
- **Casos de Uso:** Es una técnica para capturar información de cómo el sistema trabaja, o como desea que trabaje.

Estos términos son de importancia al entender que un sistema está compuesto de un conjunto de modelos que permiten expresar a un producto desde cada una de las perspectivas de interés.



El código fuente es el modelo más detallado del sistema, sin embargo es necesario tener otras visiones del sistema.

UML se encuentra conformado de los siguientes diagramas:

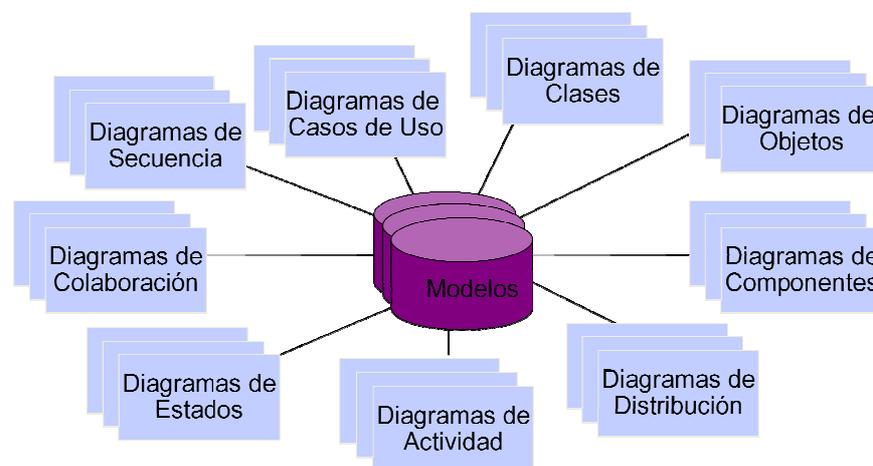


Figura 4.3: Principales diagramas de UML.

“Para modelar un sistema es suficiente utilizar una parte de UML, el 80% de la mayoría de los problemas pueden modelarse usando alrededor del 20% de UML” [Grady Booch].

Es por esta razón que no se realizarán todos los diagramas que constan en UML, solamente los más importantes.

4.1.2. RUP.

4.1.2.1. Historia de RUP.

A continuación, se presenta una figura, la cual detalla la historia que ha venido siguiendo RUP:

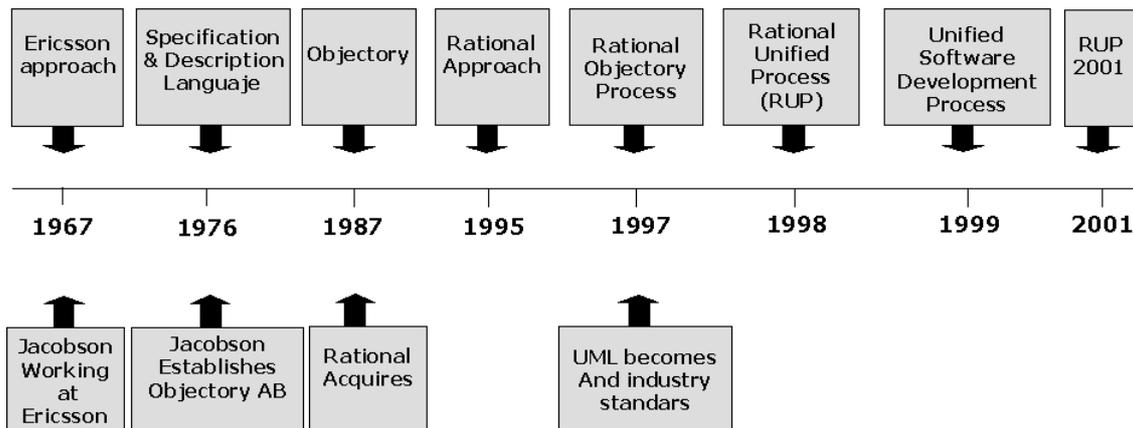


Figura 4.4: Historia de RUP.

El antecedente más importante se ubica en 1967 con la Metodología Ericsson (*Ericsson Approach*), elaborada por Ivar Jacobson, que es una aproximación de desarrollo basada en componentes que introdujo el concepto de Caso de Uso. Entre los años de 1987 a 1995, Jacobson fundó la compañía *Objectory AB* y lanza el proceso de desarrollo *Objectory* (abreviación de *Object Factory*).

Posteriormente en 1995, *Rational Software Corporation* adquiere *Objectory AB* y, entre 1995 y 1997, se desarrolla *Rational Objectory Process* (ROP) a partir de *Objectory 3.8* y del Enfoque Rational (*Rational Approach*), adoptando UML como lenguaje de modelado.

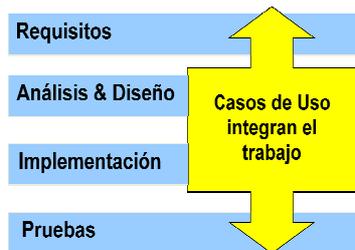
Desde ese entonces y a la cabeza de Grady Booch, Ivar Jacobson y James Rumbaugh, Rational Software desarrolló e incorporó diversos elementos para expandir ROP, destacándose especialmente el flujo de trabajo conocido como modelado del negocio. En junio del 1998 se lanza *Rational Unified Process*.²⁹

²⁹ FUENTE: <http://www.scielo.org.pe/pdf/iigeo/v10n19/a05v10n19.pdf>

4.1.2.2. Características Esenciales de RUP.

RUP tiene seis características esenciales:

- **Dirigido por Casos de Uso:** La gran ventaja que presenta al usar Casos de Uso es que permiten integrar el trabajo.

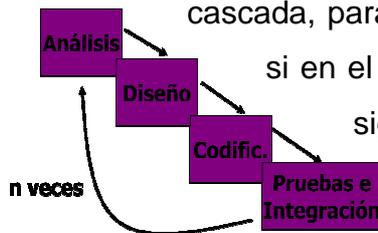


Primeramente, captura, valida y define los Casos de Uso (Requisitos).

Segundo, realiza los Casos de Uso (Análisis, Diseño e implementación).

Por último verifica que los Casos de uso se comporten como lo esperado (pruebas de carga).

- **Proceso Iterativo e Incremental:** Las actividades se encadenan en una cascada, para luego volverse a repetir con el objetivo de que si en el último escalón se produce un error, se lo corrija siguiendo el mismo proceso.



- **Adaptar el Proceso:** Las características propias del proyecto o empresa (en este caso TELCOLSAM S.A.) deben ser totalmente cubiertas por el proceso. El tamaño del mismo, así como su tipo o las regulaciones que lo condicionen, influirán en su diseño específico.
- **Equilibrar Propiedades:** Los requerimientos pueden variar dependiendo de la persona que lo solicite, por tanto, debe encontrarse un equilibrio que satisfaga los deseos de todos. Gracias a este equilibrio, se podrán corregir desacuerdos que surjan en el futuro.
- **Nivel de Abstracción:** Se deben escoger las herramientas adecuadas, y utilizarlas adecuadamente para que el usuario no perciba nada más que su interfaz gráfica.
- **Calidad:** Debe asegurarse que cada fase lleve un control de Calidad y no al final. El aseguramiento de la calidad forma parte del proceso de desarrollo y no de un grupo independiente.

4.2. POINTSOFT.

4.2.1. REQUISITOS.

4.2.1.1. Objetivo del Proyecto POINTSOFT.

PROYECTO	Desarrollo de un Sistema de Software para la Gestión de Cobranzas, para la empresa TELCOLSAM S.A., utilizando la tecnología Windows Presentation Foundation (WPF).
DESCRIPCIÓN	Diseñar e implementar un sistema computacional, modular, en una arquitectura Cliente-Servidor conjunto con la arquitectura en tres Capas, para la Gestión de Puntos de Venta (PDV) y Gestión de Cobranzas para la empresa TELCOLSAM S.A.

Tabla 4.2: Objetivo y Descripción de POINTSOFT.

4.2.2. DIAGRAMAS RUP PARA POINTSOFT.

4.2.2.1. Diagrama de Actividades.

Este diagrama define las actividades del Sistema, siendo útil para esquematizar los flujos de trabajo (workflows) del mismo.

Este punto ya fue previamente tratado en el **Título “DETERMINACIÓN DEL FLUJO DE TRABAJO.”**, **página 10**, el cual servirá para los propósitos del diagrama de actividades.

4.2.2.2. Diagrama de Casos de Uso.

Es una técnica para capturar información sobre los servicios que un sistema proporciona a su entorno, desde el punto de vista del usuario. Especifica los requisitos independientemente de la implementación, definiendo los límites del sistema y las relaciones dentro el sistema y el entorno.

Es importante recordar que los Casos de Uso se realizan desde la visión de la Clase de Usuario, y no de los programadores.

Al modelar el Diagrama de Casos de Uso no buscamos analizar el detalle, y mucho menos los flujos. Todo ese detalle lo podremos plasmar en otro tipo de diagramas, como los diagramas de secuencia, de estados, o simplemente un texto en una especificación.

Hay que tomar en cuenta que entre estos diagramas de uso se encuentran dos tipos de relaciones:

- **<<include>>** En términos muy simples, cuando relacionamos dos casos de uso con un “include”, estamos diciendo que el primero (el caso de uso base) incluye al segundo (el caso de uso incluido). Es decir, el segundo es parte esencial del primero. Sin el segundo, el primero no podría funcionar bien, pues no podría cumplir su objetivo.
- **<<extend>>** Una de las diferencias básicas es que en el caso del “extend” hay situaciones en que el caso de uso de extensión **no es indispensable que ocurra**, y cuando lo hace ofrece un valor extra (extiende) al objetivo original del caso de uso base. En cambio en el “include” es necesario que ocurra el caso incluido, tan sólo para satisfacer el objetivo del caso de uso base.

El objetivo de estos tipos de relaciones **NO** consiste en motivar la división de los casos de uso en la mayor cantidad de pedazos, pues esto enreda el diagrama. Entendiendo esto y siendo congruentes, obtendremos un beneficio real para el proyecto.

4.2.2.2.1. *Requerimientos e Identificación de Actores.*

La primera etapa de RUP es la obtención de Requerimientos, sin embargo no presentamos ya que en el **Capítulo 1, Título “ESPECIFICACIÓN DE REQUERIMIENTOS DEL SOFTWARE.”, página 23**, se centra en este tema.

De igual manera, los actores han sido detallados en el primer capítulo, razón por la cual, solamente se presenta una tabla descriptiva de los actores que intervienen en el proceso.

Tipos de Actores:

- **PRINCIPALES:** Personas que usan el sistema.
- **SECUNDARIOS:** Personas que mantienen o administran el sistema.
- **MATERIAL EXTERNO:** Dispositivos materiales imprescindibles que forman parte del ámbito de la aplicación y deben ser utilizados.
- **OTROS SISTEMAS:** Sistemas con los que el sistema interactúa.

ACTORES EN POINTSOFT	
Empleado Instalaciones	Actor principal (Usuario Final).
Empleado Cobranzas	Actor principal (Usuario Final).
Administrador	Actor Principal. Es una generalización de “Empleado Instalaciones” y “Empleado Cobranzas”.
Base de Datos	Sistema. SGBD que forman parte del ámbito de la aplicación y debe ser utilizado.

Tabla 4.3: Actores presentes en POINTSOFT.

4.2.2.2.2. Identificación de Casos de Uso.

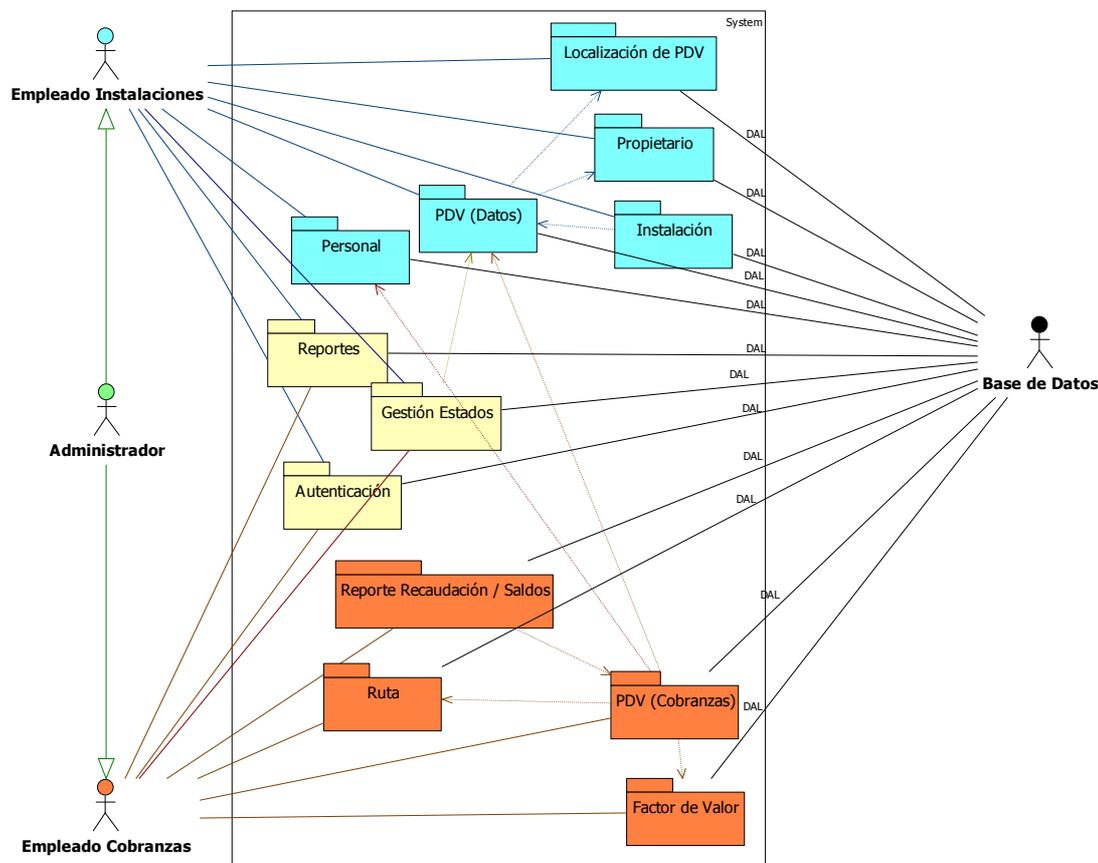


Figura 4.5: Casos de Uso Global en POINTSOFT.

La **Figura 4.5**, representa los Casos de Uso de POINTSOFT de la manera más general, donde cada agrupación de casos de uso (paquete), tendrá un subconjunto de paquetes (Sub-casos de Uso) detallando su comportamiento.

En las siguientes páginas, se detallará cada uno de los elementos que contiene cada paquete, mediante una ilustración y una tabla.

4.2.2.2.1. Sub-Casos de Uso en paquete Autenticación.

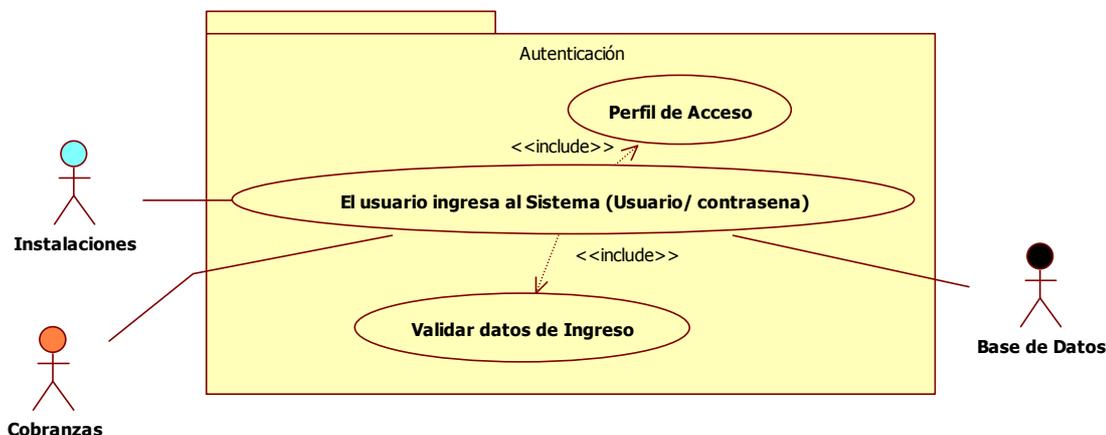


Figura 4.6: Sub-Casos de Uso en el paquete Autenticación.

DESCRIPCIÓN SUB-CASOS DE USO EN PAQUETE AUTENTICACIÓN	
ACTORES	Empleado Cobranzas, Empleado Instalaciones, Base de Datos.
OBJETIVO	Ingresar al Sistema POINTSOFT.
DESCRIPCIÓN	Para poder navegar en el sistema es necesario que el usuario se logue, identificándose como usuario válido del sistema.
PRECONDICIÓN	Ninguna.
SECUENCIA	<ul style="list-style-type: none"> • El usuario digita tanto su clave como su contraseña. • El sistema valida la información. • El sistema brinda un perfil de acceso de ser los datos válidos.
ALTERNATIVA	Ninguna.

Tabla 4.4: Descripción Sub-Casos de Uso en paquete Autenticación.

4.2.2.2.2. *Sub-Casos de Uso en paquete Localización de PDV.*

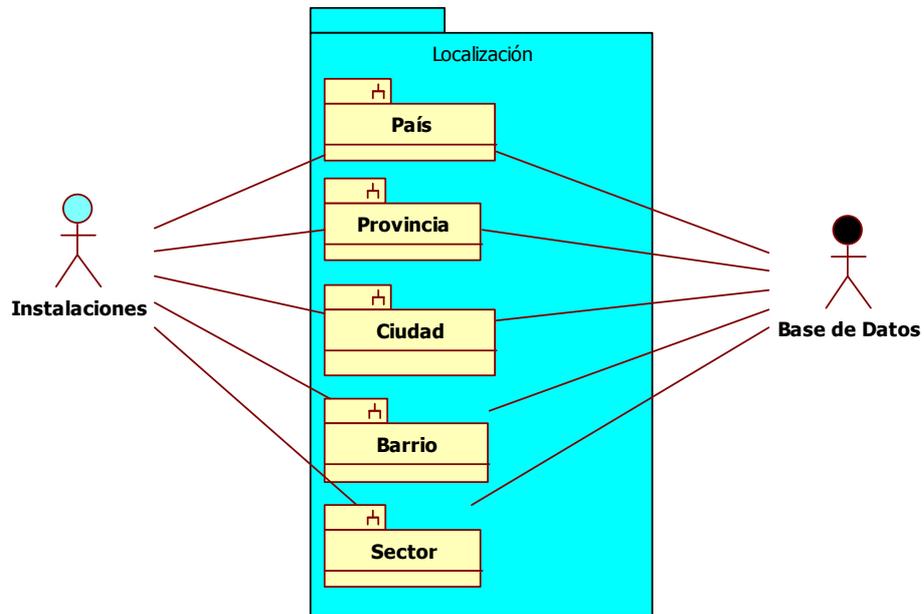


Figura 4.7: Sub-Casos de Uso en el Paquete Localización.

- **Sub-Casos de Uso en los Paquetes País; Provincia; Ciudad; Sector; y, Barrio.**

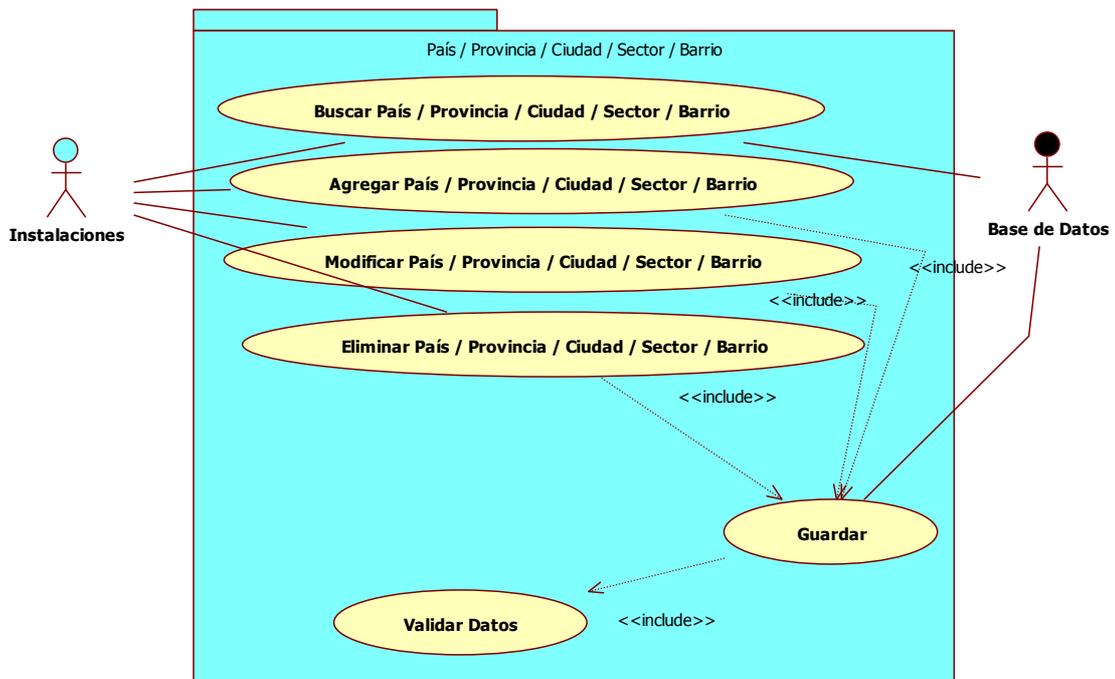


Figura 4.8: Sub-Casos de Uso de País, Provincia, Ciudad, Sector, y Barrio.

DESCRIPCIÓN SUB-CASOS DE USO EN PAQUETE LOCALIZACIÓN	
ACTORES	Empleado Instalaciones, Base de Datos
OBJETIVO	Buscar, ingresar, modificar y eliminar todo tipo de localizaciones: País, Provincia, Ciudad, Sector, Barrio.
DESCRIPCIÓN	<p>El usuario creará, modificará y eliminará las diferentes ubicaciones dependiendo de sus necesidades para poder ubicar detalladamente a un PDV.</p> <p>El sistema es el encargado de realizar las validaciones necesarias según las reglas del negocio.</p>
PRECONDICIÓN	<ul style="list-style-type: none"> • Para País: Ninguna. • Para Provincia: Que exista al menos un (1) país. • Para Ciudad: Que exista al menos una (1) provincia. • Para Sector: Ninguna. • Para Barrio: Que exista al menos una (1) ciudad y (1) sector.
SECUENCIA	<ul style="list-style-type: none"> • El usuario digita su opción (Insertar, modificar, eliminar). • Ingresar los datos si lo necesita. • Selecciona el botón Guardar. • Para la opción Buscar, se ingresa el parámetro de búsqueda, y se obtienen los registros solicitados.
ALTERNATIVA	Ninguna.

Tabla 4.5: Descripción de Sub-Casos de Uso en paquete Localización.

NOTA: Todos los paquetes dentro de Localización, tienen la misma secuencia por tanto se representa con un solo paquete por simplicidad.

4.2.2.2.3. Sub-Casos de Uso en paquete Propietario.

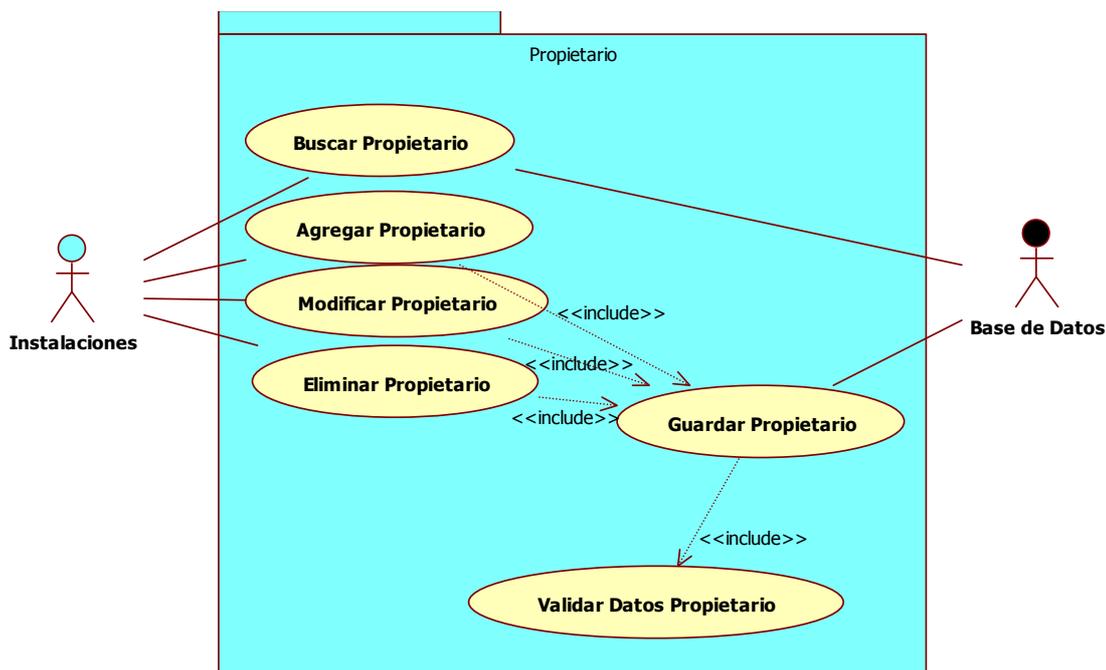


Figura 4.9: Sub-Casos de Uso en el paquete Propietario.

DESCRIPCIÓN SUB-CASO DE USO EN EL PAQUETE PROPIETARIO	
ACTORES	Empleado Instalaciones, Base de Datos
OBJETIVO	Buscar, crear, modificar y eliminar Propietarios/Clientes correspondientes a un Punto de Venta
DESCRIPCIÓN	El usuario es el encargado de operar a Propietario/Clientes identificando todas sus características. El sistema es el encargado de realizar las validaciones necesarias según las reglas del negocio.
PRECONDICIÓN	Ninguna.
SECUENCIA	<ul style="list-style-type: none"> • El usuario digita su opción (Insertar, modificar, eliminar). • Ingresa los datos si lo necesita. • Selecciona el botón Guardar. • Para la opción Buscar, se ingresa el parámetro de búsqueda, y se obtienen los registros solicitados.
ALTERNATIVA	Ninguna.

Tabla 4.6: Descripción de Sub-Casos de Uso en paquete Propietario.

4.2.2.2.4. Sub-Casos de Uso paquete Punto de Venta.

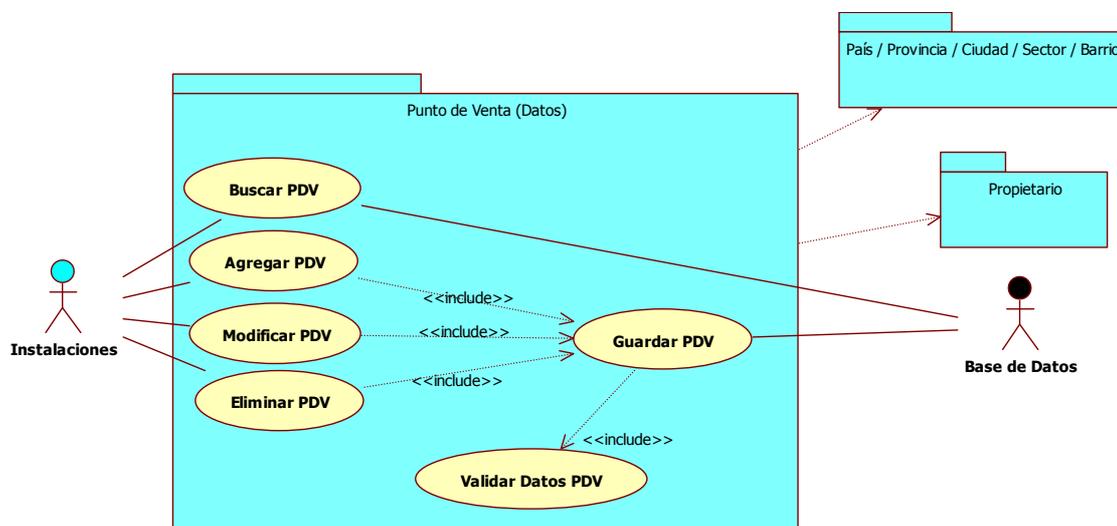


Figura 4.10: Sub-Casos de Uso en el paquete Punto de Venta (Datos).

DESCRIPCIÓN SUB-CASOS DE USO EN PAQUETE PUNTO DE VENTA (DATOS)	
ACTORES	Empleado Instalaciones, Base de Datos.
OBJETIVO	Buscar, crear, modificar y eliminar Punto de Venta
DESCRIPCIÓN	El usuario es el encargado de operar Puntos de Venta identificando todas sus características, incluyendo Localización y Sociedad (Propietario). El sistema es el encargado de realizar las validaciones necesarias según las reglas del negocio.
PRECONDICIÓN	<ul style="list-style-type: none"> • Debe existir un Propietario. • Debe existir una Localización específica (Barrio).
SECUENCIA	<ul style="list-style-type: none"> • El usuario digita su opción (Insertar, modificar, eliminar). • Ingresar los datos si lo necesita. • Selecciona el botón Guardar. • Para la opción Buscar, se ingresa el parámetro de búsqueda, y se obtienen los registros solicitados.
ALTERNATIVA	Ninguna.

Tabla 4.7: Descripción Sub-Casos de Uso en paquete Punto de Venta (Datos).

4.2.2.2.5. Sub-Casos de Uso en el paquete Gestión Estados.

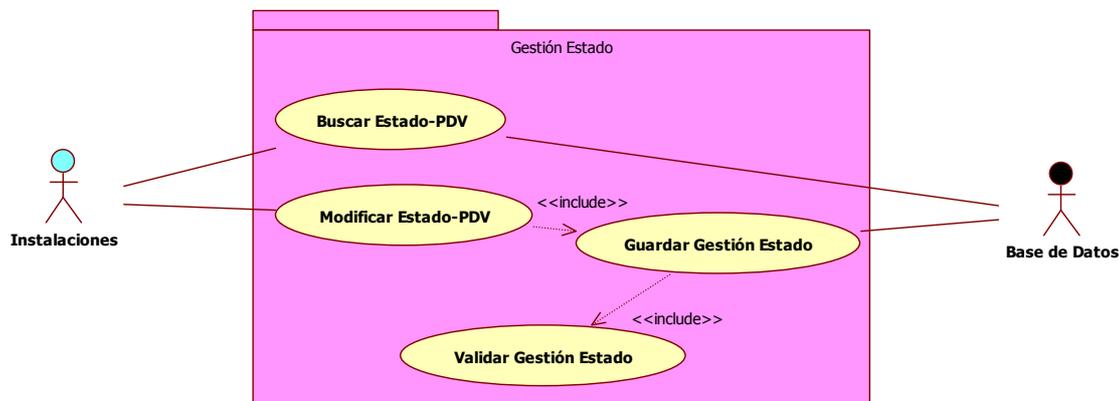


Figura 4.11: Sub-Casos de Uso en el paquete Gestión Estados.

DESCRIPCIÓN SUB-CASOS DE USO EN PAQUETE GESTIÓN ESTADO	
ACTORES	Empleado Instalaciones, Base de Datos
OBJETIVO	Buscar y modificar el Estado(s) Actual(es) que tiene un PDV
DESCRIPCIÓN	El usuario es el encargado de operar los Estados actuales para un PDV. Identificando todas sus características. El sistema es el encargado de realizar las validaciones necesarias.
PRECONDICIÓN	<ul style="list-style-type: none"> • Debe existir un Punto de Venta.
SECUENCIA	<ul style="list-style-type: none"> • El usuario digita su opción (modificar). • Ingresa los datos si lo necesita. • Selecciona el botón Guardar. • Para la opción Buscar, se ingresa el parámetro de búsqueda, y se obtienen los registros solicitados.
ALTERNATIVA	Ninguna

Tabla 4.8: Descripción Sub-Casos de Uso en el paquete Gestión Estados.

4.2.2.2.6. Sub-Casos de Uso paquete Instalación.

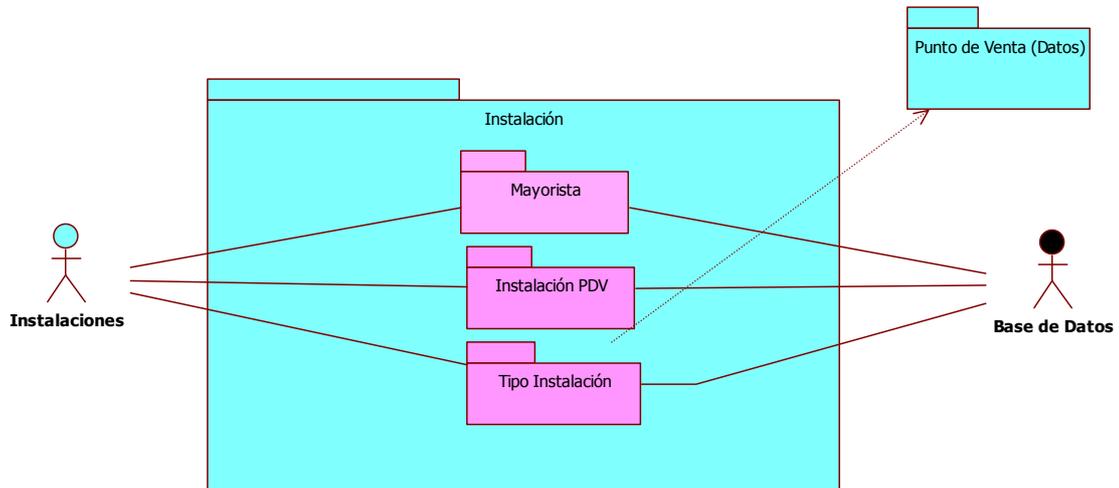


Figura 4.12: Sub-Casos de Uso en el paquete Instalación.

- Sub-Casos de Uso en Mayorista.

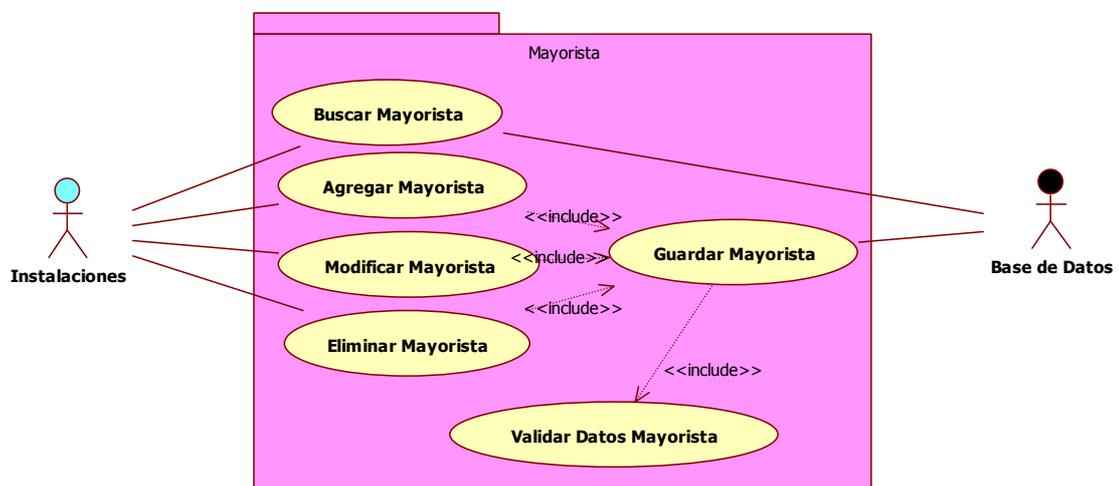


Figura 4.13: Sub-Casos de Uso en Mayorista.

DESCRIPCIÓN SUB-CASO DE USO MAYORISTA	
ACTORES	Empleado Instalaciones, Base de Datos
OBJETIVO	Buscar, crear, modificar y eliminar Mayoristas.
DESCRIPCIÓN	El usuario es el encargado de operar los Mayoristas con los que trabaja TELCOLSAM, identificando todas sus características. El sistema es el encargado de realizar las validaciones necesarias.
PRECONDICIÓN	Ninguna.
SECUENCIA	<ul style="list-style-type: none"> • El usuario digita su opción (insertar, modificar, eliminar). • Ingresa los datos si lo necesita. • Selecciona el botón Guardar. • Para la opción Buscar, se ingresa el parámetro de búsqueda, y se obtienen los registros solicitados.
ALTERNATIVA	Ninguna.

Tabla 4.9: Descripción Sub-Casos de Uso en Mayorista.

- **Sub-Casos de Uso en Tipo Instalación.**

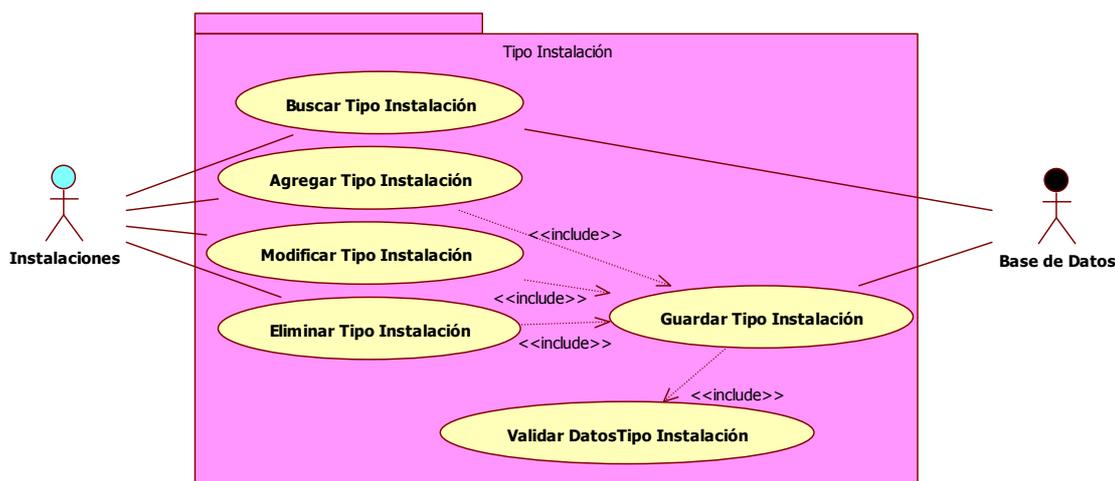


Figura 4.14: Sub-Casos de Uso en Tipo Instalación.

DESCRIPCIÓN SUB-CASO DE USO TIPO INSTALACIÓN	
ACTORES	Empleado Instalaciones, Base de Datos
OBJETIVO	Buscar, crear, modificar y eliminar Tipo Instalación.
DESCRIPCIÓN	El usuario es el encargado de operar el Tipo de Instalaciones que se pueden realizar a un PDV, identificando todas sus características. El sistema es el encargado de realizar las validaciones necesarias.
PRECONDICIÓN	Ninguna
SECUENCIA	<ul style="list-style-type: none"> • El usuario digita su opción (insertar, modificar, eliminar). • Ingresa los datos si lo necesita. • Selecciona el botón Guardar. • Para la opción Buscar, se ingresa el parámetro de búsqueda, y se obtienen los registros solicitados.
ALTERNATIVA	Ninguna

Tabla 4.10: Descripción Sub-Caso de Uso Tipo Instalación.

- Sub-Casos de Uso en Instalación PDV.

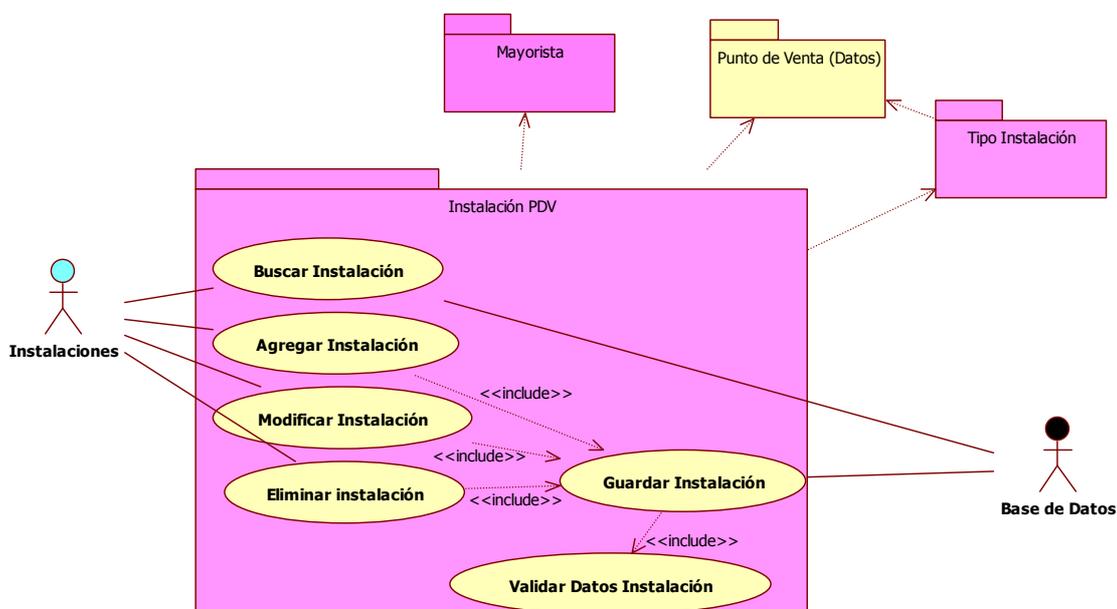


Figura 4.15: Sub-Casos de Uso en Instalación PDV.

DESCRIPCIÓN SUB-CASOS DE USO EN INSTALACIÓN PDV	
ACTORES	Empleado Instalaciones, Base de Datos
OBJETIVO	Buscar, crear, modificar y eliminar Instalaciones
DESCRIPCIÓN	El usuario es el encargado de operar las Instalaciones a realizar a un PDV, identificando todas sus características. El sistema es el encargado de realizar las validaciones necesarias.
PRECONDICIÓN	<ul style="list-style-type: none"> • Debe existir un Punto de Venta. • Debe existir un Mayorista. • Debe existir un Tipo Instalación.
SECUENCIA	<ul style="list-style-type: none"> • El usuario digita su opción (insertar, modificar, eliminar). • Ingresa los datos si lo necesita. • Selecciona el botón Guardar. • Para la opción Buscar, se ingresa el parámetro de búsqueda, y se obtienen los registros solicitados.
ALTERNATIVA	Ninguna.

Tabla 4.11: Descripción Sub-Casos de Uso en Instalación de PDV.

4.2.2.2.2.7. *Sub-Casos de Uso paquete Personal.*

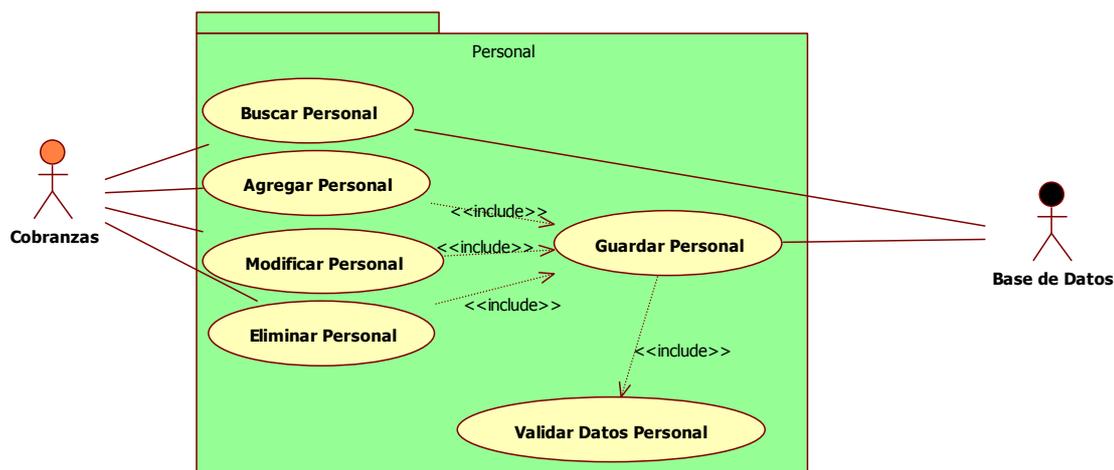


Figura 4.16: Sub-Casos de Uso en el paquete Personal.

DESCRIPCIÓN SUB-CASOS DE USO EN PERSONAL	
ACTORES	Empleado Cobranzas, Base de Datos.
OBJETIVO	Buscar, crear, modificar y eliminar Empleados en TELCOLSAM S.A.
DESCRIPCIÓN	El usuario es el encargado de operar los empleados que existe en TELCOLSAM S.A., identificando todas sus características. El sistema es el encargado de realizar las validaciones necesarias.
PRECONDICIÓN	Ninguna.
SECUENCIA	<ul style="list-style-type: none"> • El usuario digita su opción (insertar, modificar, eliminar). • Ingresa los datos si lo necesita. • Selecciona el botón Guardar. • Para la opción Buscar, se ingresa el parámetro de búsqueda, y se obtienen los registros solicitados.
ALTERNATIVA	Ninguna.

Tabla 4.12: Descripción Sub-Casos de Uso en Personal.

4.2.2.2.2.8. *Sub-Casos de Uso paquete Ruta.*

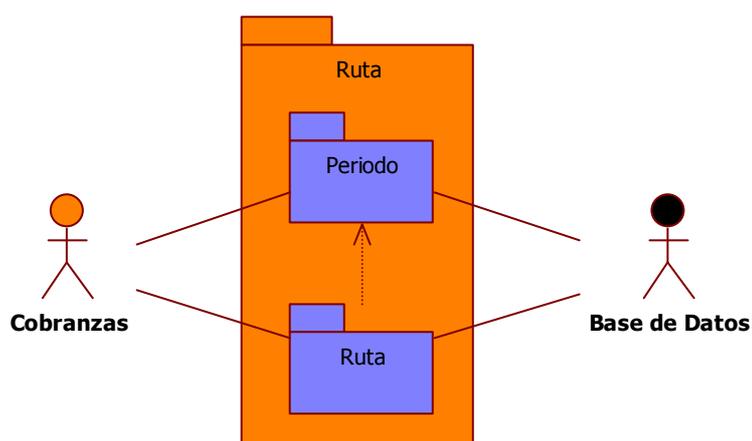


Figura 4.17: Sub-Caso de Uso en el Paquete Ruta.

- **Sub-Casos de uso en Periodo.**

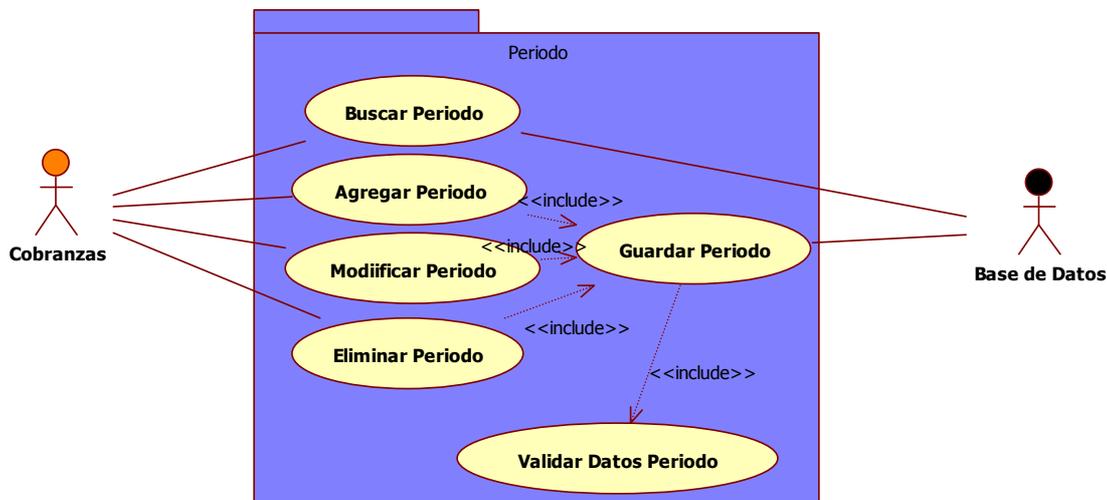


Figura 4.18: Sub-Casos de Uso en Periodo.

DESCRIPCIÓN SUB-CASOS DE USO EN PERIODO	
ACTORES	Empleado Cobranzas, Base de Datos.
OBJETIVO	Buscar, crear, modificar, eliminar Periodos
DESCRIPCIÓN	El usuario es el encargado de operar los periodos para ser asignados a diversas rutas, identificando todas sus características. El sistema es el encargado de realizar las validaciones necesarias.
PRECONDICIÓN	Ninguna.
SECUENCIA	<ul style="list-style-type: none"> • El usuario digita su opción (Insertar, modificar, eliminar). • Ingresa los datos si lo necesita. • Selecciona el botón Guardar. • Para la opción Buscar, se ingresa el parámetro de búsqueda, y se obtienen los registros solicitados.
ALTERNATIVA	Ninguna.

Tabla 4.13: Descripción Sub-Caso de Uso Periodo.

4.2.2.2.9. Sub-Casos de Uso paquete Factor de Valor.

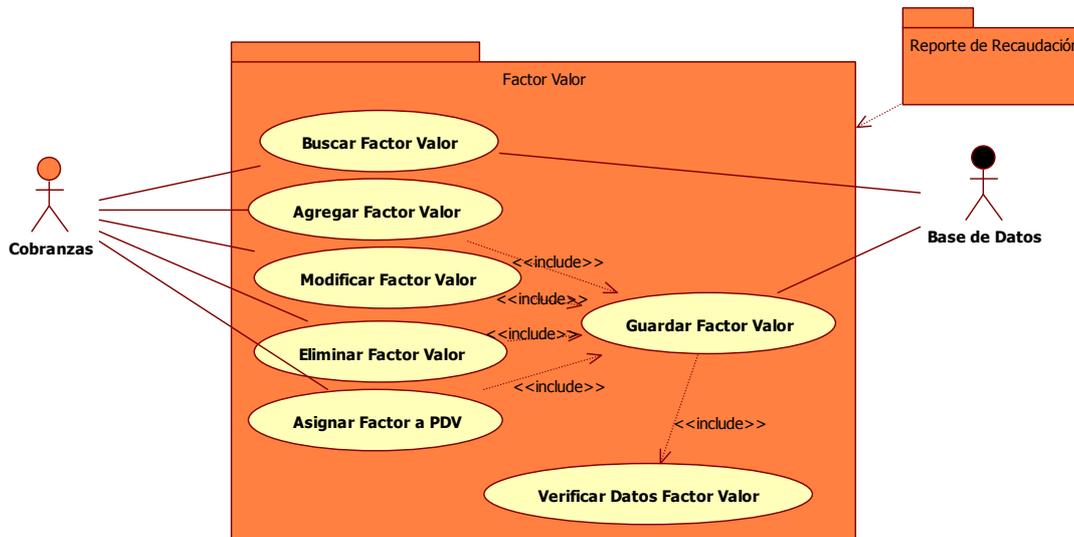


Figura 4.20: Sub-Casos de Uso en el paquete Factor Valor.

DESCRIPCIÓN SUB-CASOS DE USO EN EL PAQUETE FACTOR VALOR	
ACTORES	Empleado Cobranzas, Base de Datos.
OBJETIVO	Buscar, crear, modificar, eliminar Factores de Valor.
DESCRIPCIÓN	El usuario es el encargado de operar los Factores de Valor, identificando todas sus características. El sistema es el encargado de realizar las validaciones necesarias según las reglas del negocio.
PRECONDICIÓN	Ninguna.
SECUENCIA	<ul style="list-style-type: none"> • El usuario digita su opción (Insertar, modificar, eliminar y asigna). • Ingresa los datos si lo necesita. • Selecciona el botón Guardar. • Para la opción Buscar, se ingresa el parámetro de búsqueda, y se obtienen los registros solicitados.
ALTERNATIVA	Ninguna

Tabla 4.15. Descripción Sub-Caso de Uso Factor de Valor.

4.2.2.2.10. *Sub-Casos de Uso paquete PDV (Cobranzas).*

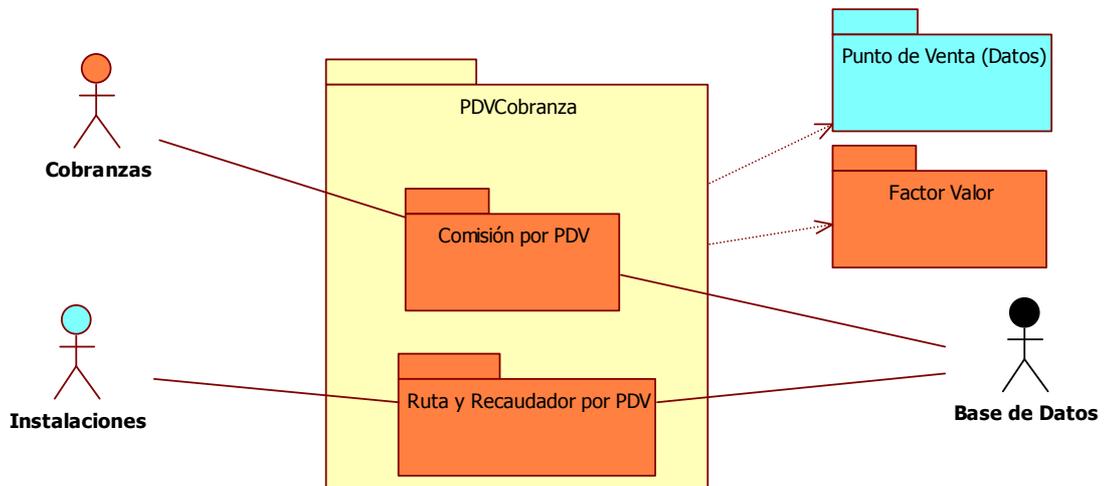


Figura 4.21: Sub-Casos de Uso en el paquete PDV (Cobranzas).

- **Sub-Casos de Uso en Comisión por PDV.**

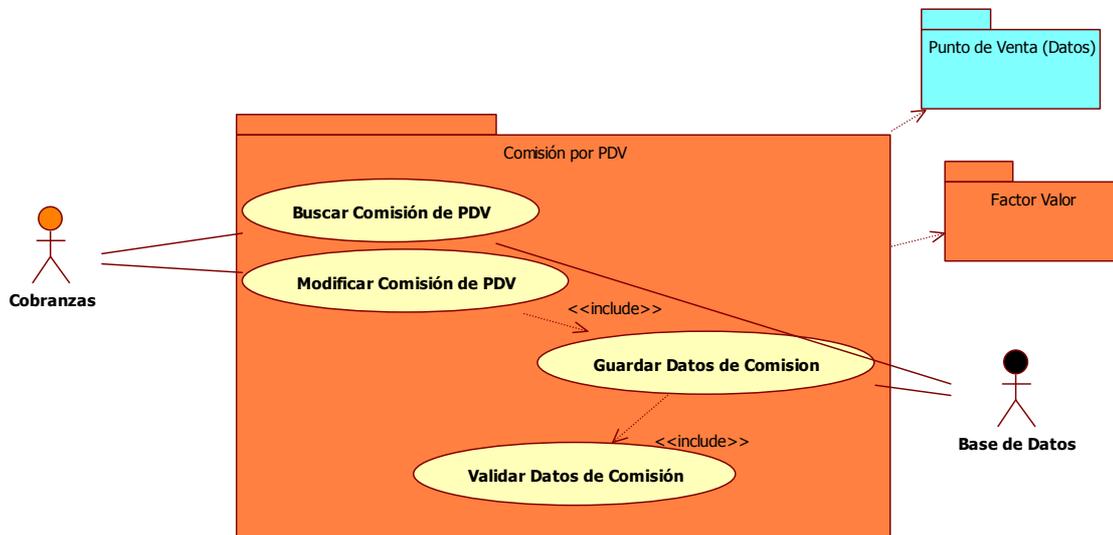


Figura 4.22: Sub-Casos de Uso en Comisión por PDV.

DESCRIPCIÓN SUB-CASOS DE USO EN COMISIÓN POR PDV.	
ACTORES	Empleado Cobranzas, Base de Datos.
OBJETIVO	Buscar y modificar Comisiones para un Punto de Venta.
DESCRIPCIÓN	El usuario es el encargado de modificar comisiones para un Punto de Venta. Identificando todas sus características. El sistema es el encargado de realizar las validaciones necesarias según las reglas del negocio.
PRECONDICIÓN	<ul style="list-style-type: none"> • Debe existir una Comisión. • Debe existir un Punto de Venta.
SECUENCIA	<ul style="list-style-type: none"> • El usuario digita su opción modificar. • El ingresa los datos necesarios. • Selecciona el botón Guardar. • Para la opción Buscar, se ingresa el parámetro de búsqueda, y se obtienen los registros solicitados.
ALTERNATIVA	Ninguna

Tabla 4.16: Sub-Casos de Uso en Comisión por PDV.

- Sub-Casos de Uso en Ruta-Recaudador por PDV.

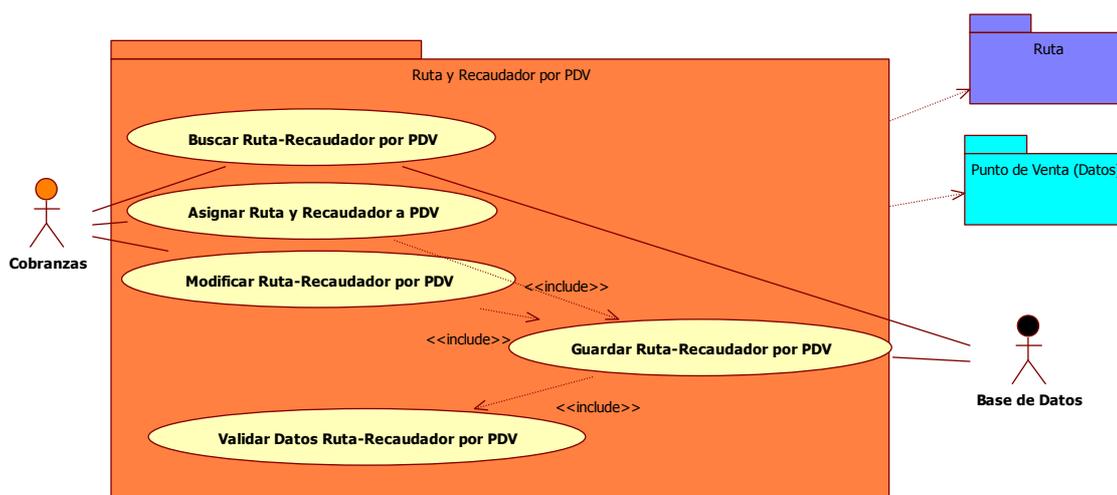


Figura 4.23: Sub-Casos de Uso en Ruta-Recaudador por PDV.

DESCRIPCIÓN SUB-CASOS DE USO RUTA Y RECAUDADOR POR PDV.	
ACTORES	Empleado Cobranzas, Base de Datos.
OBJETIVO	Buscar, asigna y Rutas y Recaudadores por Punto de Venta.
DESCRIPCIÓN	El usuario es el encargado de modificar Ruta - Recaudador para un Punto de Venta. Identificando todas sus características. El sistema es el encargado de realizar las validaciones necesarias según las reglas del negocio.
PRECONDICIÓN	<ul style="list-style-type: none"> • Debe existir una Ruta. • Debe existir un Punto de Venta. • Debe existir un Recaudador.
SECUENCIA	<ul style="list-style-type: none"> • El usuario digita su opción modificar. • El ingresa los datos necesarios. • Selecciona el botón Guardar. • Para la opción Buscar, se ingresa el parámetro de búsqueda, y se obtienen los registros solicitados.
ALTERNATIVA	Ninguna.

Tabla 4.17: Sub-Casos de Uso en Ruta-Recaudador por PDV.

4.2.2.2.2.11. *Sub-Casos de Uso en el paquete Reporte Recaudación / Saldos.*

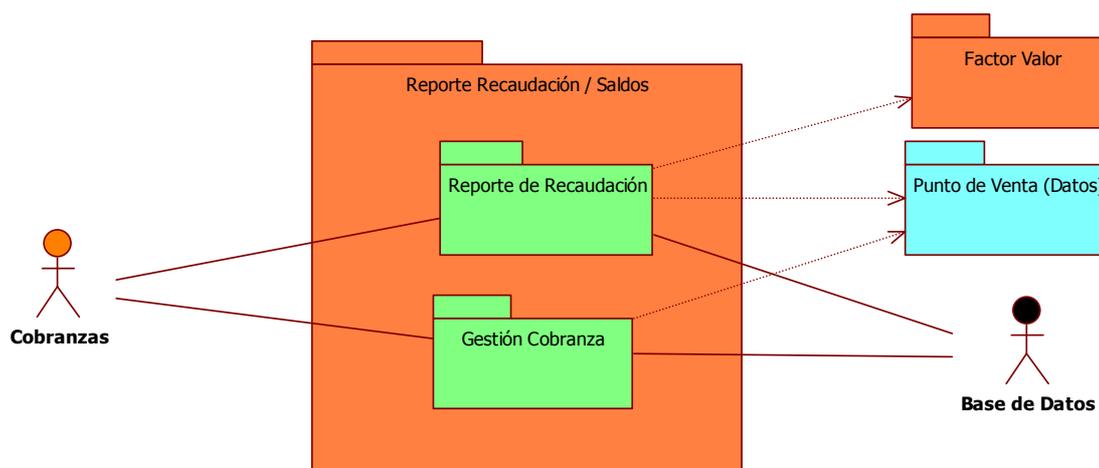


Figura 4.24: Sub-Casos de Uso en el paquete Reporte Recaudación / Saldos.

- **Sub-Casos de Uso en Gestión de Cobranzas.**

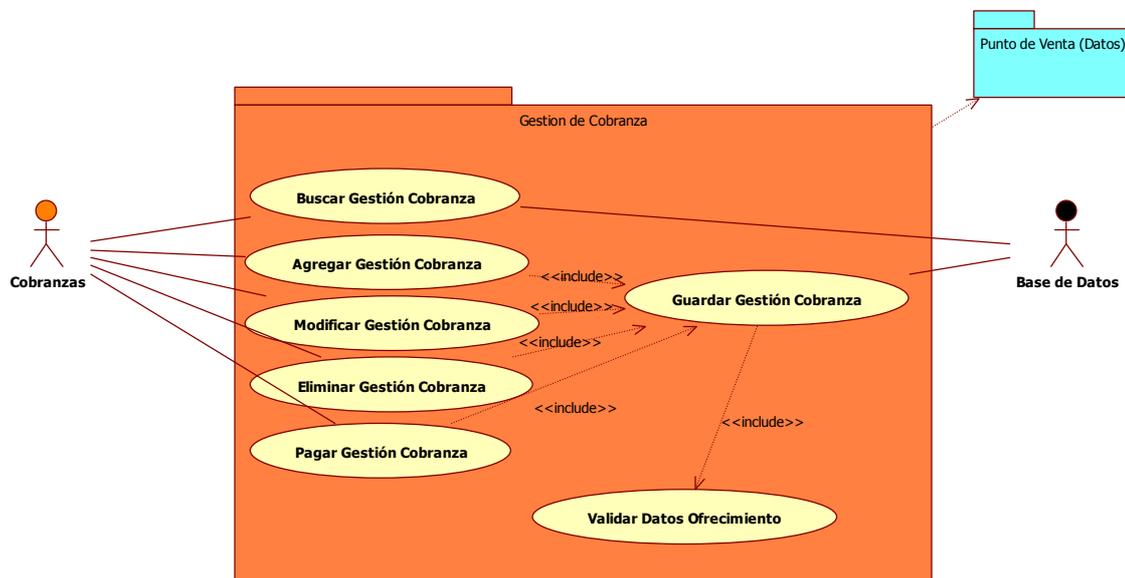


Figura 4.25: Sub-Casos de Uso en Gestión de Cobranzas.

DESCRIPCIÓN SUB-CASOS EN GESTIÓN DE COBRANZAS	
ACTORES	Empleado Cobranzas, Base de Datos.
OBJETIVO	Buscar, crear, modificar, eliminar y Pagar Ofrecimientos de Pago
DESCRIPCIÓN	El usuario es el encargado de operar una lista de Ofrecimientos de Pago que haya hecho un PDV, identificando todas sus características. El sistema es el encargado de realizar las validaciones necesarias según las reglas del negocio.
PRECONDICIÓN	<ul style="list-style-type: none"> • Debe existir un Punto de Venta.
SECUENCIA	<ul style="list-style-type: none"> • El usuario digita su opción (insertar, modificar, eliminar, pagar). • Ingresa los datos si lo necesita. • Selecciona el botón Guardar. • Para la opción Buscar, se ingresa el parámetro de búsqueda, y se obtienen los registros solicitados.
ALTERNATIVA	Ninguna.

Tabla 4.18: Descripción Sub-Casos en Gestión de Cobranzas.

- **Sub-Casos de uso en Reporte de Recaudación.**

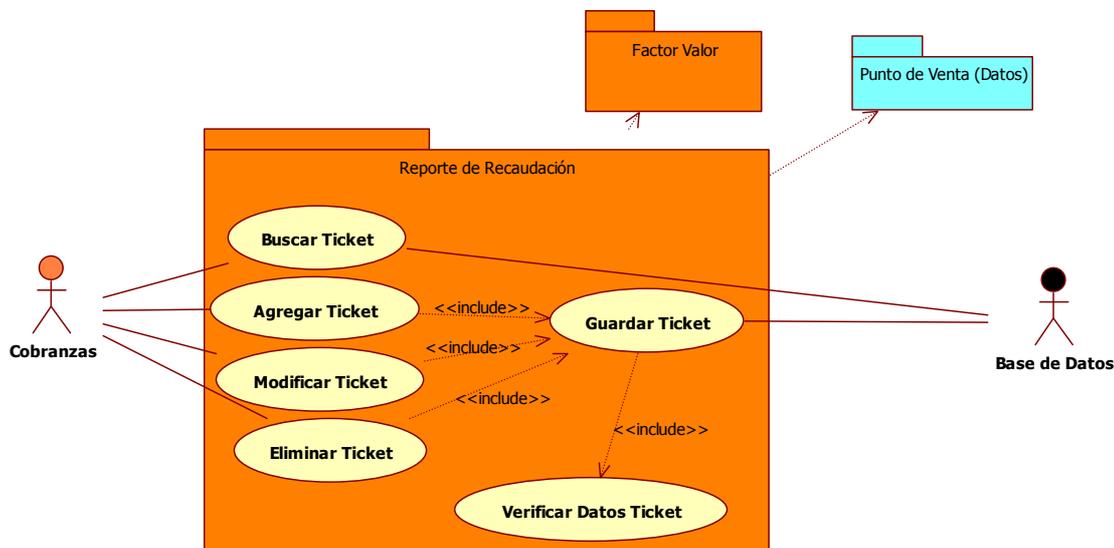


Figura 4.26: Sub-Casos de Uso en Reporte de Recaudación.

DESCRIPCIÓN SUB-CASO DE USO REPORTE DE RECAUDACIÓN	
ACTORES	Empleado Cobranzas, Base de Datos.
OBJETIVO	Buscar, crear, modificar, eliminar Tickets o (Reporte de Recaudación)
DESCRIPCIÓN	<p>El usuario es el encargado de operar los reportes de Recaudación por cada PDV, identificando todas sus características.</p> <p>El sistema es el encargado de realizar las validaciones necesarias según las reglas del negocio.</p>
PRECONDICIÓN	<ul style="list-style-type: none"> • Debe existir un Punto de Venta. • El PDV debe tener asignada una Ruta. • El PDV debe tener asignada una Comisión.
SECUENCIA	<ul style="list-style-type: none"> • El usuario digita su opción (insertar, modificar, eliminar). • Ingresa los datos si lo necesita. • Selecciona el botón Guardar. • Para la opción Buscar, se ingresa el parámetro de búsqueda, y se obtienen los registros solicitados.
ALTERNATIVA	Ninguna

Tabla 4.19: Descripción Sub-Casos de Uso en Reporte de Recaudación.

4.2.2.2.12. Sub-Casos de Uso paquete Reportes

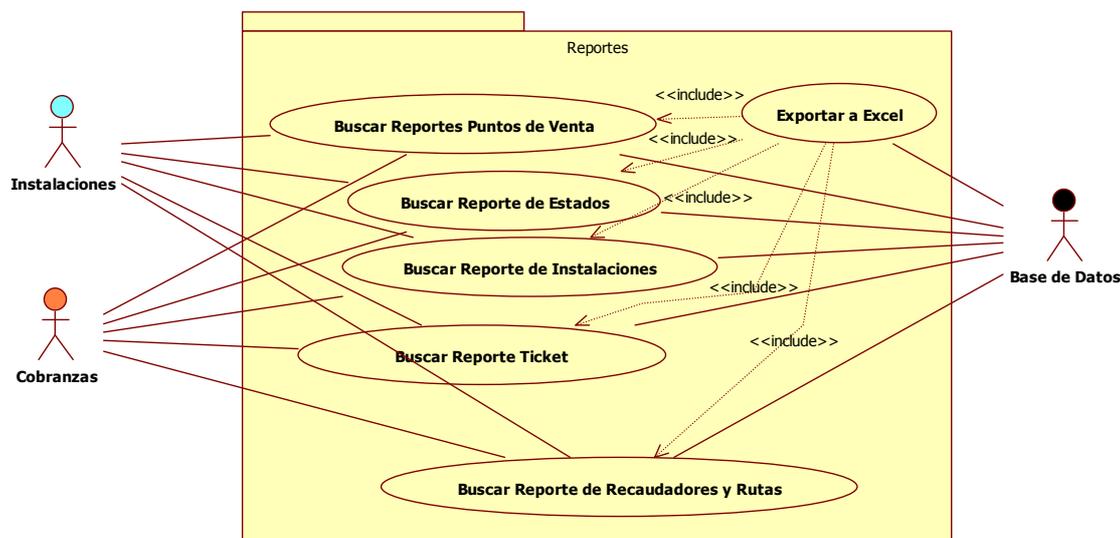


Figura 4.27: Sub-Casos de Uso en el paquete Reportes.

DESCRIPCIÓN SUB-CASO EN EL PAQUETE REPORTES	
ACTORES	Empleado Cobranzas, Empleado Instalaciones, Base de Datos.
OBJETIVO	Buscar los diferentes Reportes de Datos a petición del Usuario. Exportar a Excel información requerida por el Usuario.
DESCRIPCIÓN	El usuario es el encargado buscar reportes según su interés Recalcamos que en este caso cada Caso de Uso es una generalización de los posibles Reportes existentes. El sistema es el encargado entregar datos actuales y verdaderos
PRECONDICIÓN	Ninguna
SECUENCIA	<ul style="list-style-type: none"> El usuario digita su opción en cualquiera de los diferentes reportes existentes. Si así lo requiere exporta a Excel.
ALTERNATIVA	Ninguna

Tabla 4.20: Descripción Sub-Casos de Uso en el paquete Reportes.

4.2.2.3. Diagrama de Estructura de Estática (de Clases).

Una clase define el ámbito de definición de un conjunto de objetos, por tanto cada objeto pertenece a una clase.

El Diagrama de Clases es el diagrama principal para el diseño del sistema, en el que presenta las clases con sus respectivas relaciones estructurales y de herencia. La definición de clases incluye definiciones para atributos y operaciones.

4.2.2.3.1. Especificación de Clases de Diseño.

Puesto que el diagrama de clases es muy extenso, lo dividiremos en secciones para poder especificarlas todas.

Notamos también que varias clases implementan la clase `IDataErrorInfo`, que brinda propiedades para poder realizar las validaciones respectivas.

[La Figura del Diagrama se encuentra en la página siguiente]

4.2.2.3.1.1. Localización.

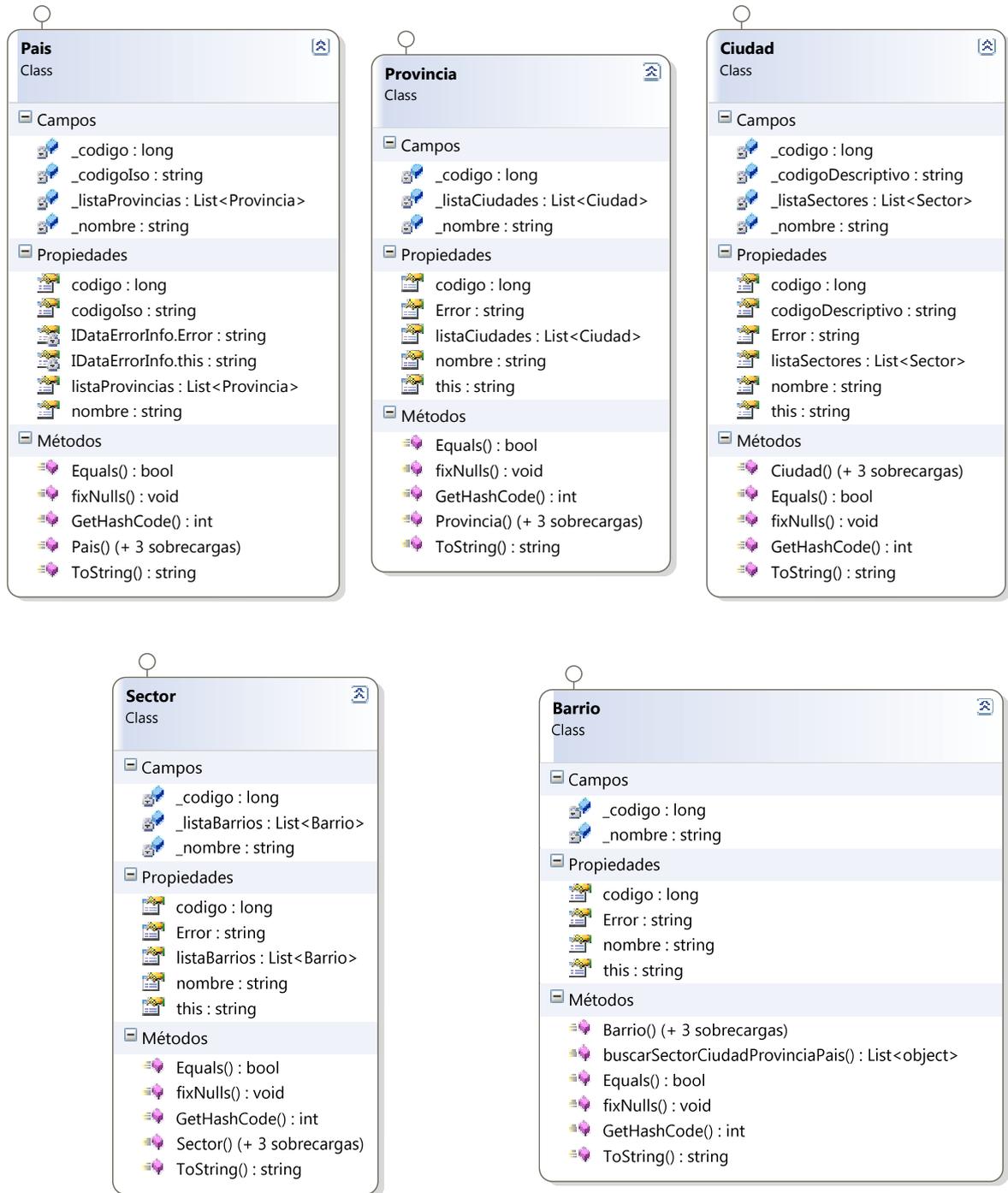


Figura 4.29: Clases de Localización POINTSOFT.

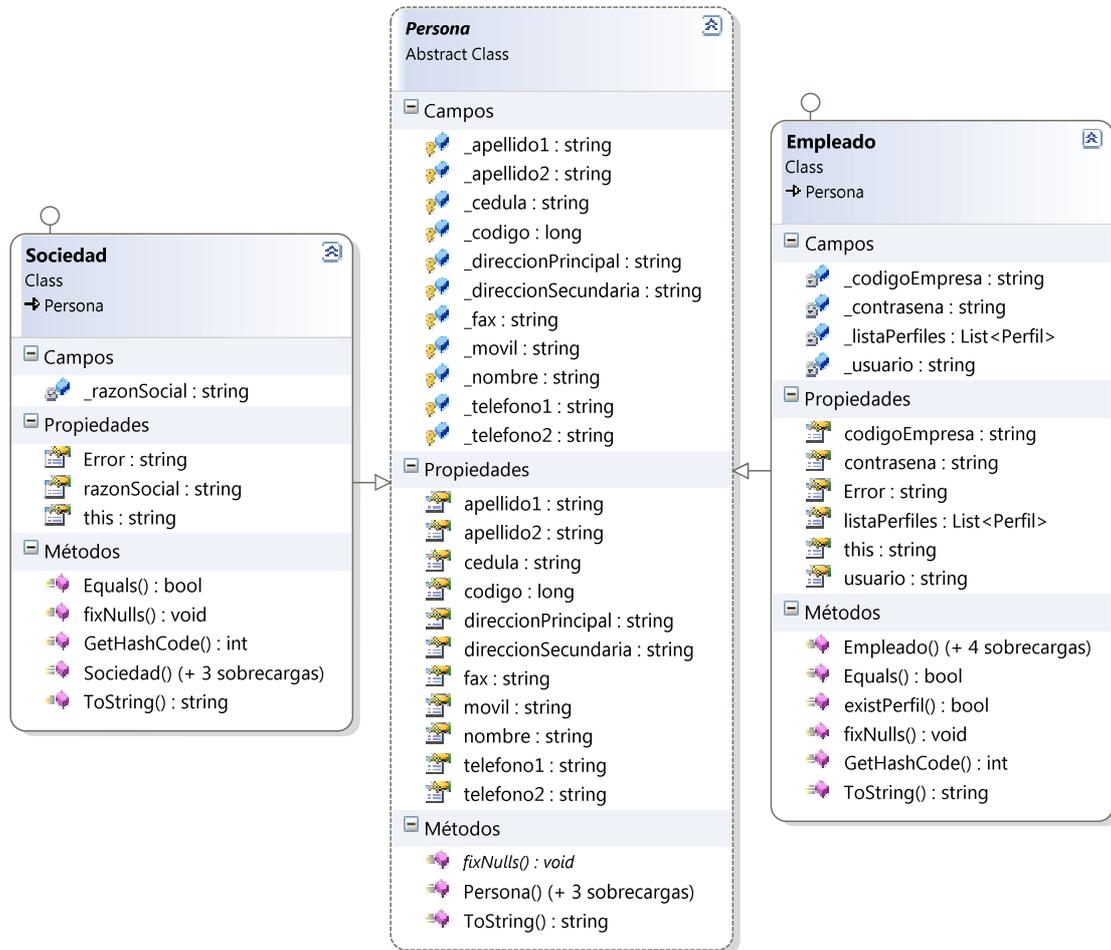
4.2.2.3.1.2. *Persona.*

Figura 4.30: Clases Persona POINTSOFT

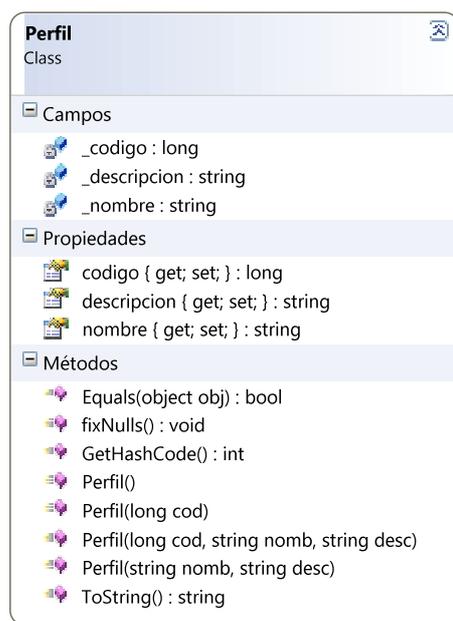


Figura 4.31: Clase Perfil POINTSOFT.

4.2.2.3.1.3. Punto de Venta.

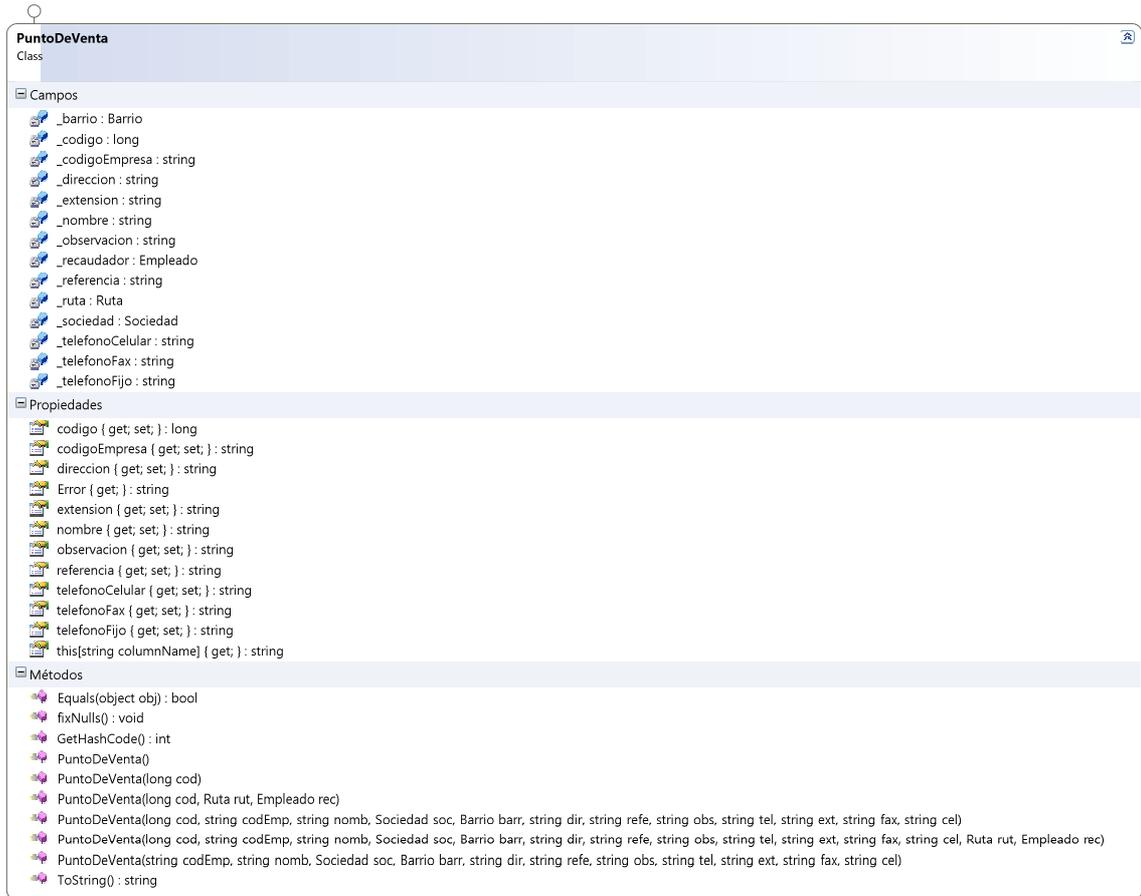


Figura 4.32: Clase Punto de Venta POINTSOFT.

4.2.2.3.1.4. Instalación.

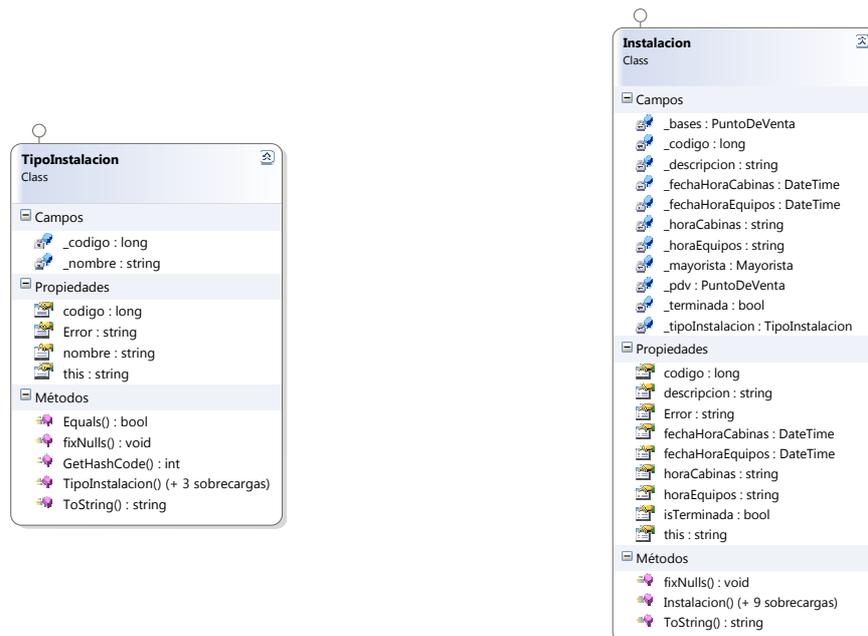


Figura 4.33: Clases Instalación POINTSOFT.

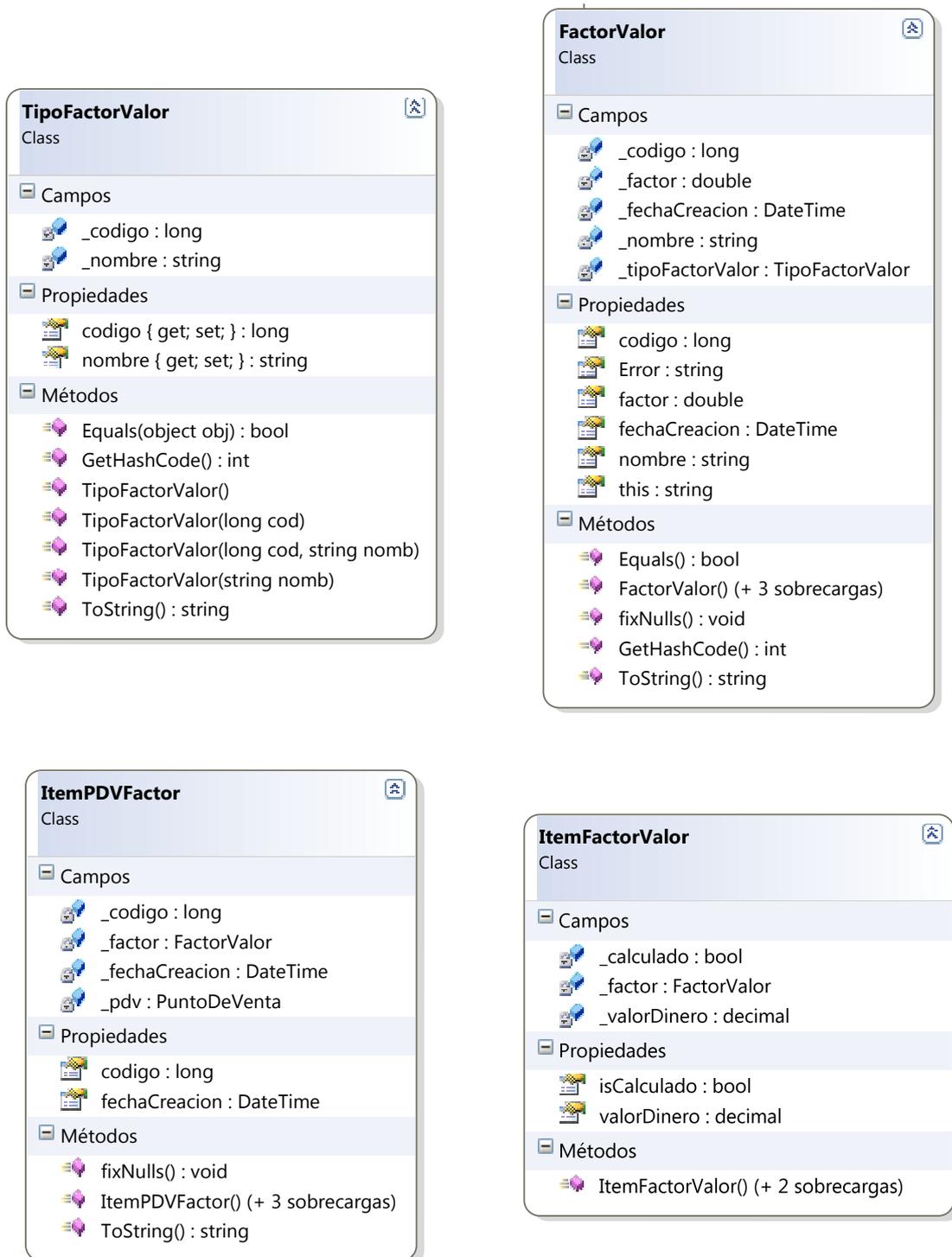
4.2.2.3.1.5. *Factor de Valor.*

Figura 4.34: Clases Factor Valor POINTSOFT.

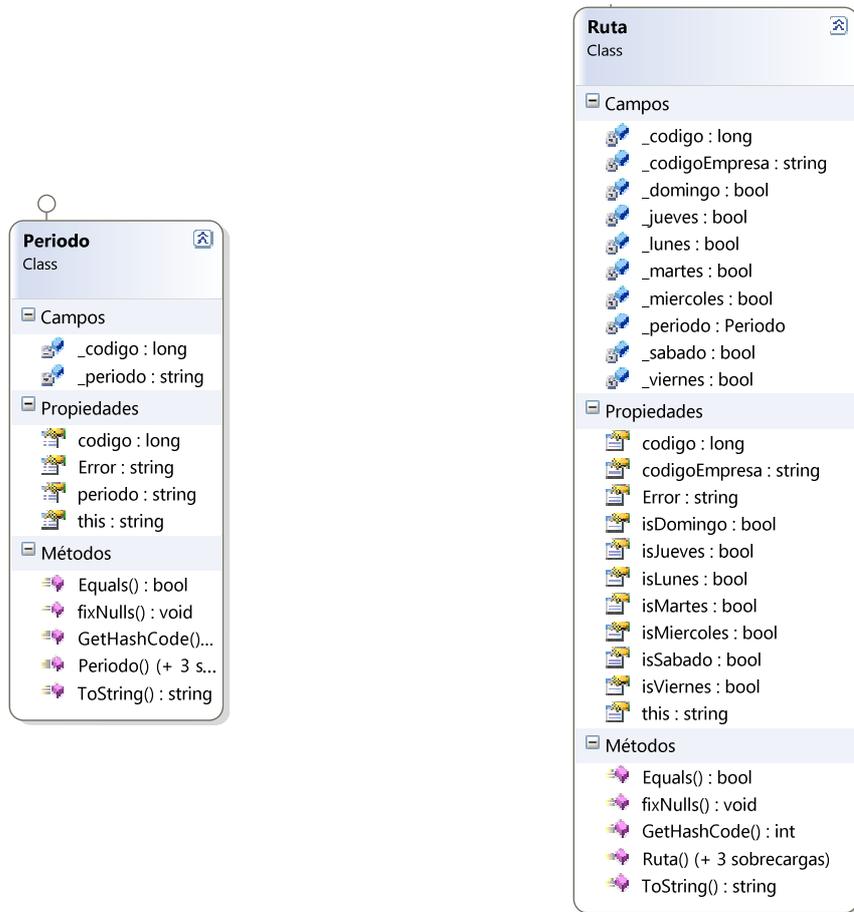
4.2.2.3.1.6. *Ruta.*

Figura 4.35: Clases Ruta-Periodo POINTSOFT.

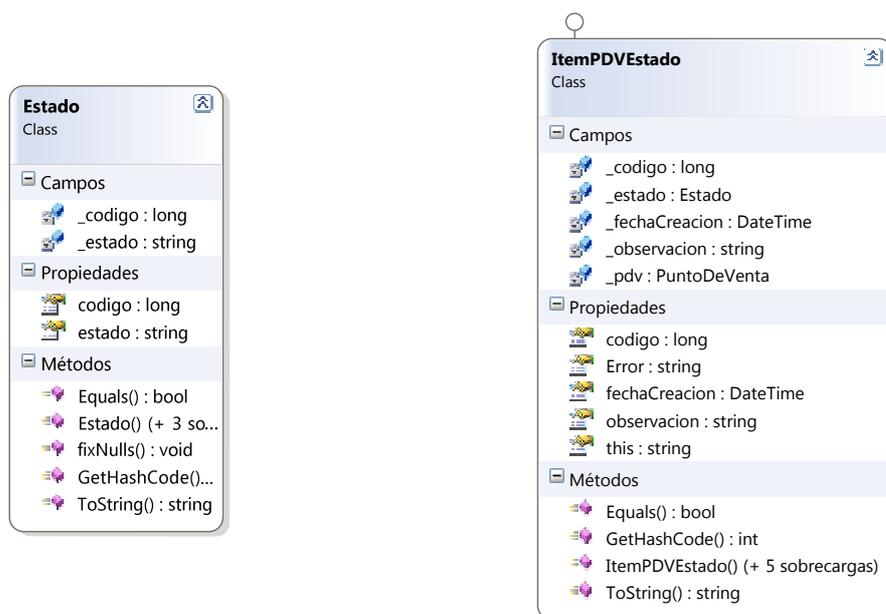
4.2.2.3.1.7. *Estado-PDV.*

Figura 4.36: Clases Estado POINTSOFT.

4.2.2.3.1.8. *Mayorista.*

The screenshot displays the 'Mayorista' class structure in IntelliJ IDEA. The class is categorized as 'Class'. It is organized into three main sections: 'Campos', 'Propiedades', and 'Métodos'.

- Campos:**
 - `_codigo : long`
 - `_codigoEmpresa : string`
 - `_nombre : string`
- Propiedades:**
 - `codigo { get; set; } : long`
 - `codigoEmpresa { get; set; } : string`
 - `Error { get; } : string`
 - `nombre { get; set; } : string`
 - `this[string columnName] { get; } : string`
- Métodos:**
 - `Equals(object obj) : bool`
 - `fixNulls() : void`
 - `GetHashCode() : int`
 - `Mayorista()`
 - `Mayorista(long cod)`
 - `Mayorista(long cod, string codEmp, string nomb)`
 - `Mayorista(string codEmp, string nomb)`
 - `ToString() : string`

Figura 4.37: Clase Mayorista POINTSOFT.

4.2.2.3.1.9. *Gestión de Cobranzas (Ofrecimientos de Pago).*

The screenshot displays the 'GestionCobranzas' class structure in IntelliJ IDEA. The class is categorized as 'Class'. It is organized into three main sections: 'Campos', 'Propiedades', and 'Métodos'.

- Campos:**
 - `_codigo : long`
 - `_descripcion : string`
 - `_fecha : DateTime`
 - `_pagado : bool`
 - `_pdv : PuntoDeVenta`
 - `_tipoGestionCobranzas : TipoGestionCobranzas`
 - `_valorPagar : decimal`
- Propiedades:**
 - `codigo : long`
 - `descripcion : string`
 - `Error : string`
 - `fecha : DateTime`
 - `isPagado : bool`
 - `this : string`
 - `valorPagar : decimal`
- Métodos:**
 - `Equals() : bool`
 - `GestionCobranzas() (+ 4 sobrecargas)`
 - `GetHashCode() : int`
 - `ToString() : string`

Figura 4.38: Clase Gestión Cobranzas.

4.2.2.3.1.10. Reporte de Recaudación.

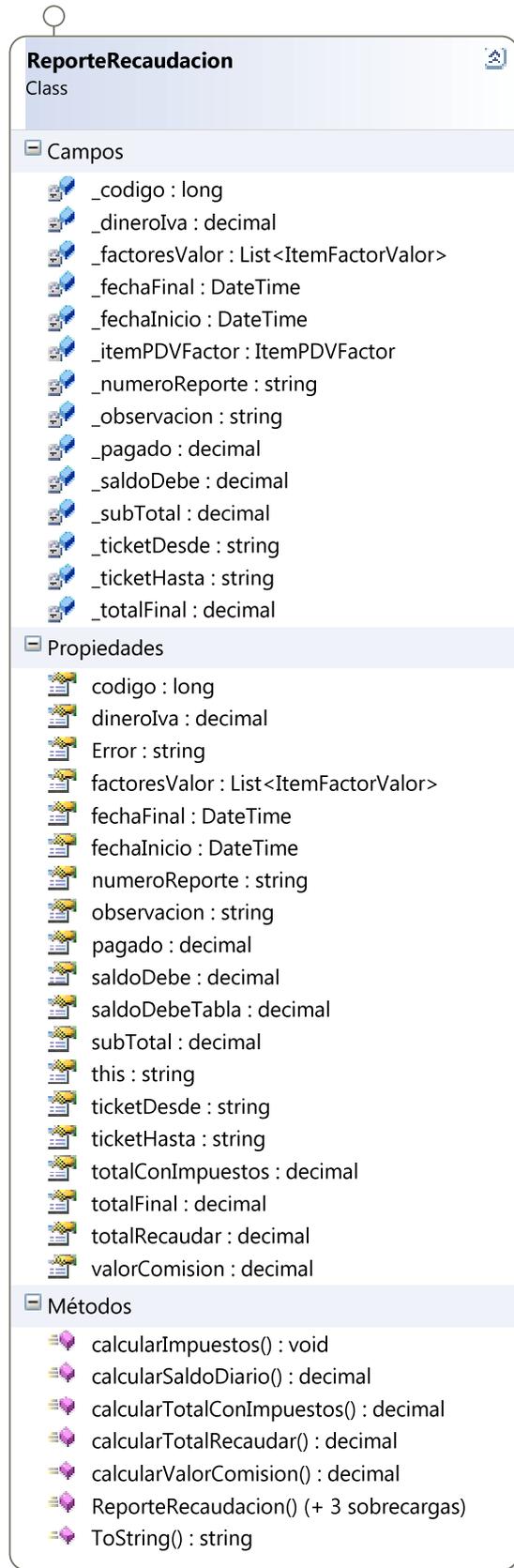


Figura 4.39: Clase Reporte de Recaudación POINTSOFT.

4.2.2.4. Diagrama de Eventos.

En este formato se establecen los eventos que pueden ser generados por el actor y van a ser atendidos por cada Caso de Uso. Por evento entendemos la interacción que tiene un actor con la aplicación a través de la interfaz gráfica, tal como el clic de un ratón, el ingreso de un texto a un componente, el movimiento de un elemento de la interfaz, etc. Todos los eventos van numerados en orden de acuerdo a la secuencia lógica como ocurrirían en la aplicación (ciclo de vida del caso de uso).

NOTA: Si se presenta una ventana por cada Caso de Uso, y en cada ventana por lo menos se tendrían tres botones para las operaciones básicas (ingresar, modificar y eliminar), sería excesivo un Diagrama de Eventos (pensando solamente eventos “clic”) por cada uno de ellos. Por esta razón, solamente se realizará un ejemplo de un caso de uso:

CONTROL DE PROYECTOS	FORMATO DE EVENTOS
Nombre del Caso de Uso:	UBICACIÓN PAÍS.
Actor Responsable:	Empleado de Instalaciones.
EVENTO	RESPUESTA DEL SISTEMA
<p>1. Selecciona abrir la pantalla País.</p> <p>3. Clic en botón Buscar.</p> <p>6. Clic en el botón Insertar.</p> <p>8. Ingreso de Datos de País (Código ISO, Nombre de País).</p>	<p>2. El sistema abre la pantalla correspondiente a país.</p> <p>4. Capturar la Selección</p> <p>5. Mostrar los países existentes en la base de datos.</p> <p>7. Capturar la Selección</p>

Continúa en la página 159.

<p>10. Clic en el botón Guardar.</p> <p>13. Clic en el botón Modificar.</p> <p>15. Escoge un ítem para ser modificado.</p> <p>16. Modifica la información requerida de dicho País.</p> <p>17. Clic en el botón Guardar.</p> <p>20. Clic en el botón Eliminar.</p> <p>22. Escoge un ítem para ser eliminado.</p> <p>23. Clic en el botón Guardar.</p>	<p>9. Capturar la Selección y prepara el sistema para ingresar datos.</p> <p>11. Captura la Selección</p> <p>12. Guarda los Datos de País.</p> <p>14. Captura la selección y prepara el Sistema para modificar datos.</p> <p>18. Captura la Selección</p> <p>19. Guarda los Datos de País.</p> <p>21. Captura la selección y prepara el Sistema para eliminar datos.</p> <p>24. Captura la Selección</p> <p>25. Guarda los Datos de País.</p>
--	---

Tabla 4.21: Diagrama de Eventos, Sub-Caso de Uso Localización.

4.2.2.5. Diagrama de Interacción.

Este diagrama muestra las interacciones de un usuario con el sistema. Una Interacción es una cadena de mensajes enviados en respuesta a un evento generado por el usuario sobre la aplicación.

El responsable o ACTOR es quien inicia el ciclo interactuando inicialmente con la interfaz de usuario: GUI; en seguida se inician todos los objetos que intervienen

en el funcionamiento del aplicativo. En este diagrama se comienza a observar el comportamiento del sistema a partir de los eventos generados por los actores.

De igual manera se realizará solamente un ejemplo (País):

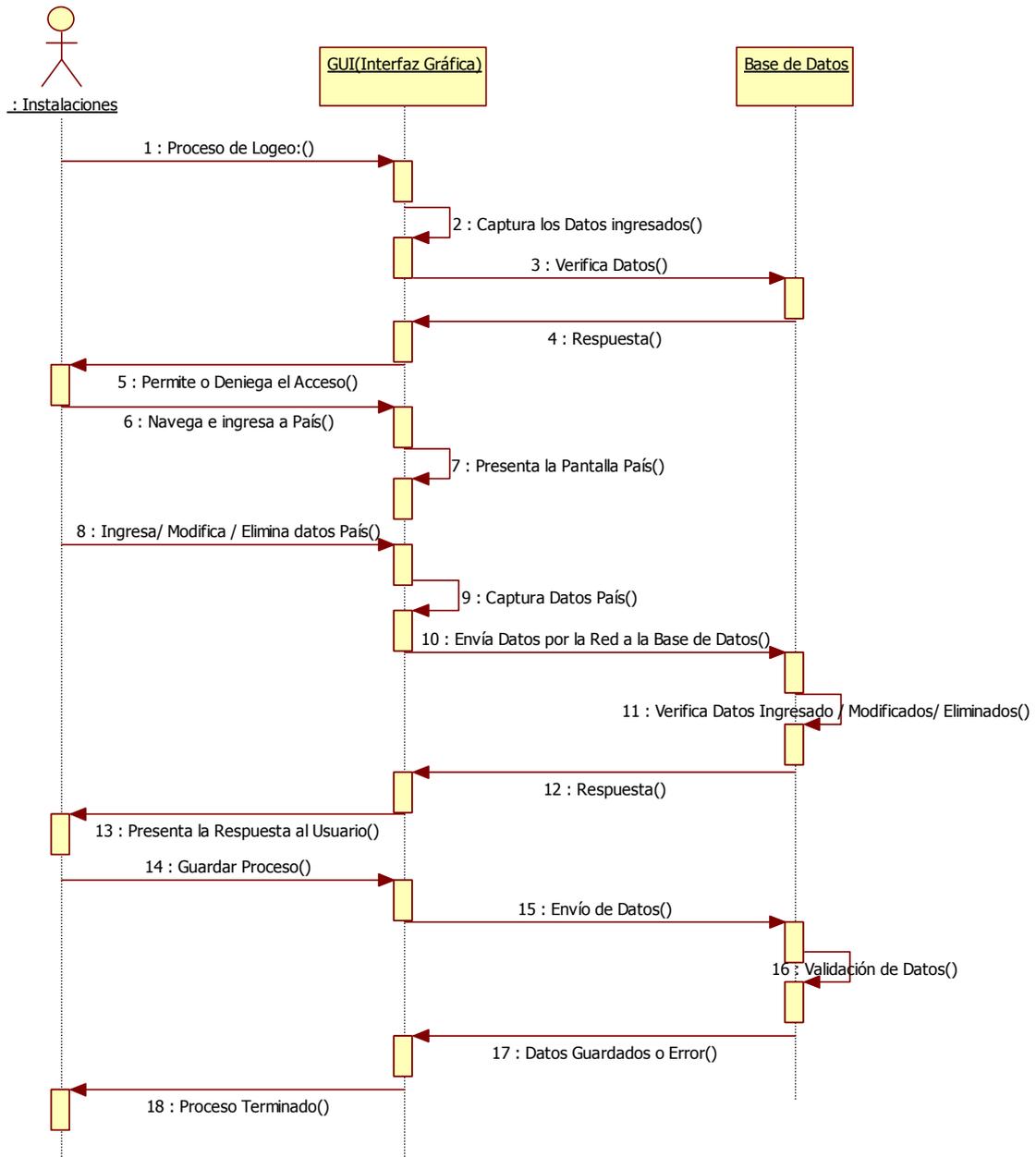


Figura 4.40: Diagrama de Interacción País, POINTSOFT.

4.2.3. DISEÑO DE LA ARQUITECTURA TRES CAPAS.

4.2.3.1. Capa de Base de Datos.

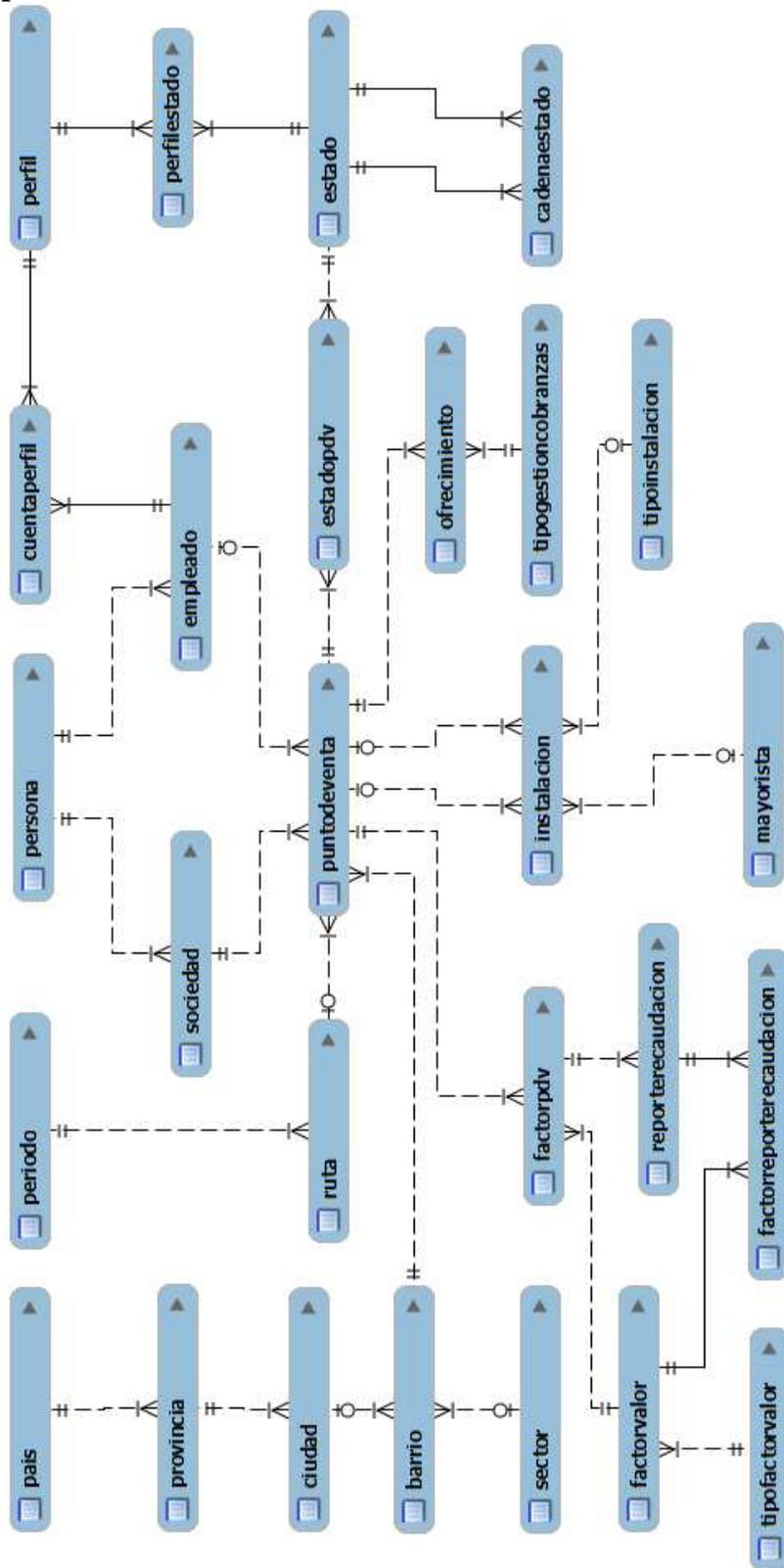


Figura 4.41: Diagrama de Base de Datos, POINTSOFT.

4.2.3.1.1. Definición de Tablas POINTSOFT.

A continuación se detallan solamente las tablas principales de POINTSOFT, ordenadas alfabéticamente:

- **Tabla Barrio:**

Columna	Tipo de Dato	Detalle	Descripción
bar_codigo	INTEGER	Primary Key. Not Null. Auto-increment.	Código de Barrio.
bar_nombre	VARCHAR(45)	Not Null.	Nombre de Barrio.
bar_seccod	INTEGER		Código de Sector.
bar_ciucod	INTEGER		Código de Ciudad.

Tabla 4.22: Descripción Tabla Barrio.

- **Tabla Ciudad:**

Columna	Tipo de Dato	Detalle	Descripción
ciu_codigo	INTEGER	Primary Key. Not Null. Auto-increment.	Código de Ciudad.
ciu_nombre	VARCHAR(45)	Not Null.	Nombre de Ciudad.
ciu_coddes	CHAR(3)		Código Descriptivo.
ciu_procod	INTEGER	Not Null.	Código Provincia.

Tabla 4.23: Descripción Tabla Ciudad.

- **Tabla Empleado:**

Columna	Tipo de Dato	Detalle	Descripción
emp_codigo	INTEGER	Primary Key. Not Null. Auto-increment.	Código de Empleado.
emp_percod	INTEGER	Not Null.	Código de Persona.
emp_codemp	VARCHAR(5)	Not Null.	Código del Empleado usado por la empresa.
emp_usuari	VARCHAR(10)		Nombre de usuario/ alias.
emp_contra	VARCHAR(40)		Contraseña.

Tabla 4.24: Descripción Tabla Empleado.

- **Tabla Estado:**

Columna	Tipo de Dato	Detalle	Descripción
est_codigo	INTEGER	Primary Key. Not Null. Auto-increment.	Código de Estado.
est_estado	VARCHAR(5)	Not Null.	Nombre de Estado.
est_restri	INTEGER	Not Null. Valor Defecto = 0.	Restricciones de Seguridad: 0= Borrar/Mostrar. 1=NoBorrar/Mostrar. 2=NoBorrar/NoMostrar.

Tabla 4.25: Descripción Tabla Estado.

- **Tabla Factor Valor:**

Columna	Tipo de Dato	Detalle	Descripción
fav_codigo	INTEGER	Primary Key. Not Null. Auto-increment.	Código de Factor Valor.
fav_tfvcod	BIGINT(20)	Not Null.	Tipo de Factor Valor: Comisión = 2. Impuesto = 1.
fav_nombre	VARCHAR(45)	Not Null.	Nombre de Factor Valor.
fav_factor	DOUBLE	Not Null.	Numero (Porcentaje) Factor Valor.
fav_fechac	DATETIME	Not Null.	Fecha de creación de Factor Valor.

Tabla 4.26: Descripción Tabla Factor Valor.

- **Tabla Instalación:**

Columna	Tipo de Dato	Detalle	Descripción
ins_codigo	INTEGER	Primary Key Not Null. Auto-increment.	Código de la Instalación.
ins_pdvcod	INTEGER	Not Null.	Código del PDV a instalar.
ins_tpicod	INTEGER	Not Null.	Código del Tipo de Instalación.
ins_maycod	INTEGER		Código del Mayorista.
ins_bascod	INTEGER		Código del PDV Bases.

Continúa en la página 164.

ins_fehoca	DATETIME	Not Null.	Fecha 1 (Cabinas).
ins_fehoeq	DATETIME		Fecha 2 (Equipos).
ins_termin	TINYINT	Not Null. Valor Defecto = 0.	Instalación terminada.
ins_descri	VARCHAR(500)		Descripción de la Instalación.

Tabla 4.27: Descripción Tabla Punto de Venta.

- **Tabla Mayorista:**

Columna	Tipo de Dato	Detalle	Descripción
may_codigo	INTEGER	Primary Key. Not Null. Auto-increment.	Código de Mayorista.
may_codemp	VARCHAR(10)	Not Null.	Código de Empresa.
may_nombre	VARCHAR(100)	Not Null.	Nombre de Mayorista.

Tabla 4.28: Descripción Tabla Mayorista.

- **Tabla País:**

Columna	Tipo de Dato	Detalle	Descripción
pai_codigo	INTEGER	Primary Key. Not Null. Auto-increment.	Código de País.
pai_nombre	VARCHAR(40)	Not Null.	Nombre de País.
pai_codiso	CHAR(2)		Código ISO de País.

Tabla 4.29: Descripción Tabla País.

- **Tabla Perfil:**

Columna	Tipo de Dato	Detalle	Descripción
prf_codigo	INTEGER	Primary Key. Not Null. Auto-increment.	Código de País.
prf_nombre	VARCHAR(45)	Not Null.	Nombre de País.
prf_descri	VARCHAR(45)	Not Null.	Código ISO de País.
prf_restri	INTEGER	Not Null. Valor Defecto = 0	Restricciones de Seguridad: 0= Borrar/Mostrar. 1=NoBorrar/Mostrar. 2=NoBorrar/NoMostrar.

Tabla 4.30: Descripción Tabla Perfil.

- **Tabla Periodo:**

Columna	Tipo de Dato	Detalle	Descripción
prd_codigo	INTEGER	Primary Key. Not Null. Auto-increment.	Código de periodo o frecuencia de Ruta.
prd_period	VARCHAR(45)	Not Null.	Nombre del Periodo.

Tabla 4.31: Descripción Tabla Periodo.

- **Tabla Persona:**

Columna	Tipo de Dato	Detalle	Descripción
per_codigo	INTEGER	Primary Key. Not Null. Auto-increment.	Código de Persona.
per_cedruc	CHAR(3)	Not Null.	Cédula o RUC de Persona.
per_nombre	VARCHAR(45)	Not Null.	Nombre de la Persona
per_apell1	VARCHAR(45)	Not Null.	Primer Apellido de Persona.
per_apell2	VARCHAR(45)		Segundo Apellido de Persona.
per_dirpri	VARCHAR(100)	Not Null.	Dirección primaria.
per_dirsec	VARCHAR(20)		Dirección Secundaria.
per_telef1	VARCHAR(20)	Not Null.	Teléfono Primario.
per_telef2	VARCHAR(20)		Teléfono Secundario.
per_tmovil	VARCHAR(20)		Teléfono móvil o celular
per_telfax	VARCHAR(20)		Fax.

Tabla 4.32: Descripción Tabla Persona.

- **Tabla Provincia:**

Columna	Tipo de Dato	Detalle	Descripción
pro_codigo	INTEGER	Primary Key. Not Null. Auto-increment.	Código de Provincia.
pro_nombre	VARCHAR(45)	Not Null.	Nombre de la Provincia.
pro_paicod	INTEGER	Not Null.	Código de País.

Tabla 4.33: Descripción Tabla Provincia.

- **Tabla Punto de Venta:**

Columna	Tipo de Dato	Detalle	Descripción
pdv_codigo	INTEGER	Primary Key. Not Null. Auto-increment.	Código del PDV.
pdv_soccod	INTEGER	Not Null.	Código de Sociedad (Propietario).
pdv_barcod	INTEGER	Not Null.	Código Barrio.
pdv_rutcod	INTEGER		Código de Ruta.
pdv_reccod	INTEGER		Código de Recaudador.
pdv_codemp	VARCHAR(7)	Not Null.	Código de Empresa.
pdv_nombre	VARCHAR(50)	Not Null.	Nombre del PDV.
pdv_direcc	VARCHAR(500)	Not Null.	Dirección Primaria.
pdv_refere	VARCHAR(500)		Dirección Secundaria.
pdv_telefo	VARCHAR(20)		Número de Teléfono.
pdv_telxt	VARCHAR(10)		Número de la Extensión.
pdv_telfax	VARCHAR(20)		Número de Fax.
pdv_tmovil	VARCHAR(20)		Número de móvil o celular.
pdv_observ	VARCHAR(500)		Observaciones.

Tabla 4.34: Descripción Tabla Punto de Venta.

- **Tabla de Reporte de Recaudación:**

Columna	Tipo de Dato	Detalle	Descripción
rep_codigo	INTEGER	Primary Key. Not Null. Auto-increment.	Código del Reporte de Recaudación.
rep_fpvcod	BIGINT(20)	Not Null.	Código del Factor Valor.
rep_fecha_i	DATETIME	Not Null.	Fecha de Inicio del Reporte.
rep_fecha_f	DATETIME	Not Null.	Fecha Final del Reporte.
rep_numrep	VARCHAR(20)	Not Null.	Número de Reporte.
rep_tickde	VARCHAR(20)	Not Null.	Ticket desde.
rep_tickha	DECIMAL(8,2)	Not Null.	Ticket Hasta.
rep_subtot	DECIMAL(8,2)	Not Null.	Subtotal.
rep_totalf	DECIMAL(8,2)	Not Null.	Total Final.
rep_pagad	DECIMAL(8,2)	Not Null.	Dinero Pagado.

Continúa en la página 167.

rep_saldod	DECIMAL(8,2)	Not Null.	Saldo Debe.
rep_sactua	DECIMAL(8,2)		Saldo Actual.
rep_observ	VARCHAR(500)		Observaciones.

Tabla 4.35: Descripción Tabla Reporte de Recaudación.

- **Tabla Ruta:**

Columna	Tipo de Dato	Detalle	Descripción
rut_codigo	INTEGER	Primary Key. Not Null. Auto-increment.	Código de Ruta.
rut_codemp	CHAR(3)	Not Null.	Código de la empresa.
rut_prdcod	INTEGER	Not Null.	Código de Periodo/ Frecuencia.
rut_dlunes	TINYINT(1)		Día lunes.
rut_dmarte	TINYINT(1)		Día martes.
rut_dmierc	TINYINT(1)		Día miércoles.
rut_djueve	TINYINT(1)		Día jueves.
rut_dviern	TINYINT(1)		Día viernes.
rut_dsabad	TINYINT(1)		Día sábado.
rut_ddomin	TINYINT(1)		Día domingo.

Tabla 4.36: Descripción Tabla Ruta.

- **Tabla Sector:**

Columna	Tipo de Dato	Detalle	Descripción
sec_codigo	INTEGER	Primary Key. Not Null. Auto-increment.	Código de Sector.
sec_nombre	VARCHAR(45)	Not Null.	Nombre de Sector.

Tabla 4.37: Descripción Tabla Sector.

- **Tabla Tipo Factor Valor:**

Columna	Tipo de Dato	Detalle	Descripción
tfv_codigo	BIGINT(20)	Primary Key. Not Null. Auto-increment.	Código de Tipo de Factor Valor.
tfv_nombre	VARCHAR(45)	Not Null.	Nombre de Factor Valor.

Tabla 4.38: Descripción Tipo Factor Valor.

- **Tabla Tipo Instalación:**

Columna	Tipo de Dato	Detalle	Descripción
tpi_codigo	INTEGER	Primary Key. Not Null. Auto-increment.	Código de Tipo de instalación.
tpi_nombre	VARCHAR(100)	Not Null.	Nombre de Tipo Instalación.

Tabla 4.39: Descripción Tipo Instalación.

4.2.3.1.2. Definición de Procedimientos Almacenados de POINTSOFT.

POINTSOFT, en su gran mayoría lleva procedimientos almacenados para efectuar las operaciones básicas que son: Buscar, Insertar, Actualizar, Eliminar.

Ya que existe una gran cantidad de procedimientos almacenados para estas operaciones, solamente se mostrará un ejemplo por cada operación básica en la tabla País:

PROCEDIMIENTO ALMACENADO	VALORES DE ENTRADA	DESCRIPCIÓN
sp_actualizarPais	vcodigo BIGINT vcodlso VARCHAR(45) vcodNombre CHAR(2)	Actualiza los datos de un País en donde el código del mismo sea igual al valor ingresado como parámetro de entrada, vcodigo .
sp_buscarPais_codigo	vparam BIGINT	Busca un país donde el código de País sea igual al valor de entrada vparam .
sp_buscarPais_nombre	vparam VARCHAR(45)	Buscar un país donde el nombre de país sea parecido al valor de entrada vparam .
sp_buscarPais_codigoProvincia	vparam BIGINT	Busca el país donde el parámetro de entrada vparam sea igual al código de su provincia correspondiente.

Continúa en la página 169.

sp_eliminarPais	vcodigo BIGINT	Elimina el país donde el parámetro de entrada vcodigo sea igual al código del país.
sp_insertarPais	vcodlso VARCHAR(45) vcodNombre CHAR(2)	Inserta el país con los valores de entrada. El código de País es auto-numérico.

Tabla 4.40: Descripción de Procedimientos Almacenados de País.

4.2.3.1.3. Definición de Funciones de POINTSOFT.

Hemos realizado una gran variedad de funciones para este proyecto. Nos valemos de éstas para obtener datos calculados de cualquier tabla en la base de datos. De igual manera, solamente daremos un breve resumen de las principales funciones realizadas:

FUNCIÓN	DESCRIPCIÓN
fn_cuentaOfrecimientos_codigoPDV:int(2)	Determina el número de ofrecimientos que tiene un PDV, dado el código.
fn_estadoActualPDV_codigo:varchar(100)	Determina el estado actual en el que se encuentra un Punto de Venta, dado el código.
fn_estadoMaxFecha_codigoPDV:bigint(20)	Determina el estado en la fecha más reciente para un PDV, dado el código
fn_fechaMaxInstalacion_codigoPDV:date	Determina la última fecha de Instalación de un PDV, dado el código
fn_fechaUltimoPago_codigoPDV:date	Determina la fecha del último pago del Punto de Venta, dado el código.
fn_fechaUltimoReporteRecaudacion_codigoPDV:date	Determina la fecha del último reporte de Recaudación (ticket), dado el código del PDV.
fn_nombreBasesUltimaInstalacion:varchar(100)	Determina el nombre de las Bases (PDV) que se utilizaron en la última instalación.

Continúa en la página 170.

fn_nombreMayoristaUltimaInstalacion:varchar(100)	Determina el nombre del mayorista que se utilizó en la última instalación.
fn_recaudador_codigoPDV:varchar(100)	Determina que empleado es el recaudador encargado de visitar el PDV, dado el código.
fn_ruta_codigoPDV:bigint(10)	Determina la ruta en la que es visitado un PDV, dado el código.
fn_ultimaComision_codigoPDV	Determina la última comisión (factor Valor), de un PDV, dado el código.
fn_ultimoReporte_codigoPDV:bigint(20)	Determina el último reporte de Recaudación de un PDV, dado el código.

Tabla 4.41: Descripción de Funciones en POINTSOFT.

4.2.3.1.4. Migración de datos a MySQL.

Para TELCOLSAM S.A. es muy importante poder utilizar el programa con datos actuales; por tanto, la migración es de real importancia para acoplar su información a esta nueva aplicación.

Ya que los datos de la empresa solamente se encuentran en archivos planos, éstos tenían un gran desorden, siendo éste el primer problema que tuvimos que enfrentar.

Para esto, se seleccionaron los datos más relevantes que se necesitaban migrar, y se completaban sus campos con valores con defecto, procurando que esta inconsistencia genere el menor impacto al momento de la utilización de los datos migrados.

Cabe destacar que previo a la migración, se realizaron pruebas del proceso en sí, de tal manera que cuando ya se trabaje con los datos reales, cualquier inconsistencia en éstos (sea de tipo de dato, conversiones, truncamientos, y otros) no interrumpa el proceso sino tan solo se registre en un log.

Luego de analizar mucho este inconveniente, llegamos a un acuerdo con la empresa en el que se comprometían a entregar la información en un único archivo

de Excel, en el orden que se necesitaba. El orden se definió, por columnas, de la siguiente manera:

(Columna A) Cédula / RUC Propietario;	(Columna N) Días de Ruta;
(Columna B) Razón Social;	(Columna O) Frecuencia de Ruta;
(Columna C) Apellidos Propietario;	(Columna P) Estado Actual de PDV;
(Columna D) Nombres Propietario;	(Columna Q) País;
(Columna E) Dirección Propietario;	(Columna R) Provincia;
(Columna F) Teléfono Propietario;	(Columna S) Ciudad;
(Columna G) Código de PDV;	(Columna T) Sector;
(Columna H) Nombre de PDV;	(Columna U) Barrio;
(Columna I) Barrio de PDV;	(Columna V) Comisión;
(Columna J) Dirección de PDV;	(Columna W) Fecha Final de Reporte de Recaudación;
(Columna K) Teléfono de PDV;	(Columna X) Ticket Hasta; y
(Columna L) Recaudador;	(Columna Y) Saldo
(Columna M) Código de Ruta;	

Puesto que TELCOLSAM S.A. en todo su tiempo de vida nunca ha tenido para su ingreso de datos una validación, el archivo acordado efectivamente presentaba datos inconsistentes como: cédulas con valores no reales, e incluso con valor “**default**” o, también, toda clase de campos vacíos. Esta situación se solucionó con una validación previa, o adaptación de los datos, de ser necesario.

La migración se realiza mediante dos clases:

En Migracion.cs: Esta clase se encuentra en la librería LibUtil, posteriormente detallada, que es la encargada de la recolección de datos del archivo en Excel y su validación para mandarlos a ingresar en la Base de Datos.

NOTA: Aclaremos que en la región “Parámetros”, existen todos los campos del archivo de datos, pero en este detalle solamente se observan los datos de Localización.

```
public static string migrarExcelToMysql(String path, int filaInicio,
BDConexion conn)
```

Método de Migración, con tres valores de inicio:
(path: que es correspondiente a la ruta del archivo para ser buscado)
(filaInicio: La fila con la que va a comenzar leyendo el archivo de Excel)
(conn: Conexión a la Base de Datos).

```
Excel.Application oXL;

Excel._Workbook oWB;
Excel._Worksheet oSheet;
Excel.Range oRng;
oXL = new Excel.Application();
oXL.Visible = false;
oWB = (Excel._Workbook)oXL.Workbooks.Open( path,...)
```

Para poder ingresar los datos, es necesario declarar variables para poder dividir el archivo de la siguiente manera:

oXL: Es una Aplicación Excel;
oWB: Un Archivo/ Libro de Excel;
oSheet: Es una hoja de Cálculo;
oRng: Rango de celdas.

Abre la hoja de Cálculo, e inicializa la fila actual.

Abre el Archivo en el Path correspondiente

```
oSheet = (Excel._Worksheet)oWB.ActiveSheet;
int filaActual = filaInicio;
```

Para almacenar los datos, necesitaremos definir variables para cada campo (en este caso solamente se realiza para Localización), definiendo dos variables, el primero para recoger el dato, y el segundo, representa el dato arreglado listo para ser ingresado a la base de datos:

```
String pais;
String provincia;
String ciudad;
String sector;
String barrio;
```

Variables para Valores recogidos del Archivo en Excel.

```
String _pais;
String _provincia;
String _ciudad;
String _sector;
String _barrio;
```

Variables para almacenar Valores arreglados.

```

/*Localizacion*/
pais = (String)oSheet.get_Range(oSheet.Cells[filasActual, columnaActual],
oSheet.Cells[filasActual, columnaActual]).Text;

    columnaActual++;

provincia = (String)oSheet.get_Range(oSheet.Cells[filasActual,
columnaActual], oSheet.Cells[filasActual, columnaActual]).Text;
    columnaActual++;
ciudad = (String)oSheet.get_Range(oSheet.Cells[filasActual,
columnaActual], oSheet.Cells[filasActual, columnaActual]).Text;

    columnaActual++;
sector = (String)oSheet.get_Range(oSheet.Cells[filasActual,
columnaActual], oSheet.Cells[filasActual, columnaActual]).Text;

    columnaActual++;
barrio = (String)oSheet.get_Range(oSheet.Cells[filasActual,
columnaActual], oSheet.Cells[filasActual, columnaActual]).Text;

    columnaActual++;

```

La manera que recorrer el archivo es mediante un lazo **While** donde para pasar a otra fila de lectura, debe haber leído todas las columnas de esa fila.

columnaActual: Aumenta en uno luego de leer la columna actual.

Como podemos ver, también se crea un log en el cual se van guardando las diversas notificaciones de éxito o error, para poder saber si los datos han sido bien ingresados; o caso contrario, encontrar sin problemas el lugar de los datos truncados o sin corrección.

Luego de almacenar en cada variable respectiva, es necesario arreglar el número obtenido:

Arregla cada campo, verificando que cada valor se convierta en el formato especificado en la base de datos.

```

if (!fixTexto(pais, 45, out _pais))
    { log.AppendLine(DateTime.Now.ToLongTimeString() + " -->
" + "Datos truncados PAIS: " + pais);
    val = false;
    }
if (!fixTexto(provincia, 45, out _provincia))
    {
    log.AppendLine(DateTime.Now.ToLongTimeString() + " --
> " + "Datos truncados PROVINCIA: " + provincia);
    val = false;
    }
if (!fixTexto(ciudad, 45, out _ciudad))
    {
    log.AppendLine(DateTime.Now.ToLongTimeString() + " --
> " + "Datos truncados CIUDAD: " + ciudad);
    val = false;
    }

```

```

        if (!fixTexto(sector, 45, out _sector))
        {
            log.AppendLine(DateTime.Now.ToLongTimeString() + " --
> " + "Datos truncados SECTOR: " + sector);
            val = false;
        }
        if (!fixTexto(barrio, 45, out _barrio))
        {
            log.AppendLine(DateTime.Now.ToLongTimeString() + " --
> " + "Datos truncados BARRIO: " + barrio);
            val = false;
        }
    }
}

```

Un solo método para ingresar todos los datos recopilados, este método se encuentra en **MigracionDAL.cs**.

```

dal.insert(_cedulaPropietario, _razonPropietario,
_apellidosPropietario[0],_apellidosPropietario[1], _nombrePropietario,
_direccionPropietario,_fonosPropietario[0], _fonosPropietario[1],
_celularPropietario[0],_codigoPDV, _nombrePDV, _direccionPDV,
_fonosPDV[0], _fonosPDV[1], _celularPDV[0],_codigoRecaudador,
_codigoRuta, _diasRuta[0], _diasRuta[1], _diasRuta[2], _diasRuta[3],
_diasRuta[4],_diasRuta[5], _diasRuta[6], _frecuenciaRuta,
_estado[0], _estado[1], _estado[2], _estado[3],_pais, _provincia,
_ciudad, _sector, _barrio,_comision,_fechaFinal, _ticketHasta, _saldo);

```

```

pais = String.Empty;

provincia = String.Empty;
ciudad = String.Empty;
sector = String.Empty;
barrio = String.Empty;

```

Reinicio de las variables para recorrer la próxima fila.

Existe un método **actualizarExcelToMySQL** que es el encargado de insertar los datos de instalaciones.

En MigracionDAL: Esta clase es la encargada del proceso de ingreso de datos, y tiene una estructura muy similar a los demás DAL correspondientes a cada Clase:

Para detallar su código, de igual manera, se tomará como ejemplo Localización:

```

public static MySqlParameter parameterPais(String obj)
{
    MySqlParameter param = new MySqlParameter();
    param.Value = obj;
    param.ParameterName = "vpais";
    param.DbType = System.Data.DbType.String;

    return param;
}

```

Se ingresa un "String" para convertirlo y que retorne un MySql Parameter.

```

public static MySqlParameter parameterProvincia(String obj)
{
    MySqlParameter param = new MySqlParameter();
    param.Value = obj;
    param.ParameterName = "vprovincia";
    param.DbType = System.Data.DbType.String;

    return param;
}

```

```

public static MySqlParameter parameterCiudad(String obj)
{
    MySqlParameter param = new MySqlParameter();
    param.Value = obj;
    param.ParameterName = "vciudad";
    param.DbType = System.Data.DbType.String;

    return param;
}

```

```

public static MySqlParameter parameterSector(String obj)
{
    MySqlParameter param = new MySqlParameter();
    param.Value = obj;
    param.ParameterName = "vsector";
    param.DbType = System.Data.DbType.String;

    return param;
}

```

```

public static MySqlParameter parameterBarrio(String obj)
{
    MySqlParameter param = new MySqlParameter();
    param.Value = obj;
    param.ParameterName = "vbarrio";
    param.DbType = System.Data.DbType.String;

    return param;
}

```

Método para insertar los datos, llamando a las funciones anteriores para su previa conversión a parámetros

```

public void insert(String vcedruc,
                  String vrazsoc,
                  String vapell1, String vapell2, String vnomsoc,
                  String vdirsoc,
                  String vtelsol, String vtelso2, String vmovsoc,
                  String vcodpdv, String vnompdv,
                  String vdirpdv,
                  String vtelpdv, String vfaxpdv, String vmovpdv,
                  String vcodrec,
                  String vrutcod,
                  bool vdlunes, bool vdmarte, bool vdmierc, bool vdjueve,
                  bool vdviern, bool vdsabad, bool vddomin,
                  String vfrecue,
                  Int16 vcodes1, Int16 vcodes2,
                  Int16 vcodes3, Int16 vcodes4,
                  String vpais, String vprovincia, String vciudad,
                  String vsector, String vbarrio,
                  Double vfactor,
                  DateTime vfecha,
                  String vtickha,
                  Decimal vsaldo)
{
    List<MySqlParameter> parametros = new List<MySqlParameter>();

    parametros.Add(parameterPais(vpais));

    parametros.Add(parameterProvincia(vprovincia));

    parametros.Add(parameterCiudad(vciudad));

    parametros.Add(parameterSector(vsector));

    parametros.Add(parameterBarrio(vbarrio));

    try
    {
        conx.exeNonQuery("sp_migrar", parametros);
    }
    catch (Exception ex)
    {
        throw new ArgumentException("Error en la BDD");
    }
}

```

La Conexión llama al procedimiento almacenado para migrar, ingresando una lista con los parámetros.

Atrapa posibles excepciones, enterando al programador que ha existido un error en la base de datos.

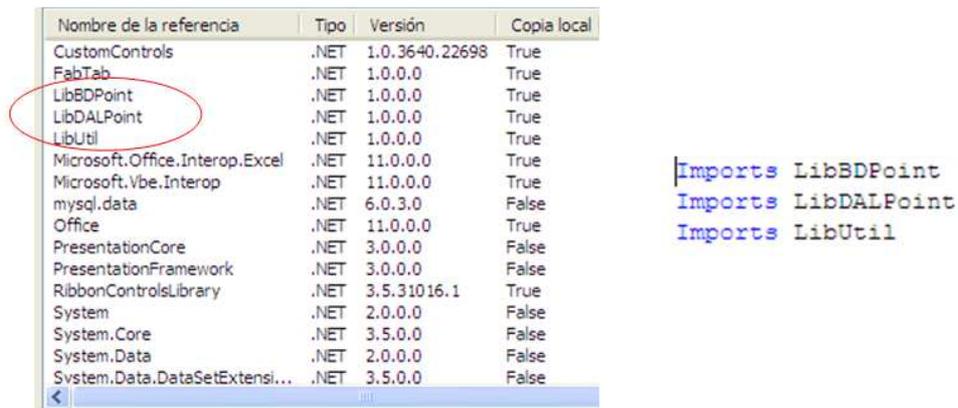
4.2.3.2. Capa de Reglas del Negocio.

Las Reglas del Negocio caracterizan a la aplicación en la manera que ésta va a comportarse.

4.2.3.2.1. Especificación de Librerías.

Para este proyecto, ha sido necesaria la programación de tres librerías. Como se definió en el **Capítulo 3, Título “REGLAS DEL NEGOCIO (SEGUNDA CAPA)”**, **página 85**, estas librerías han sido programadas bajo el lenguaje C#.

Para que estas librerías puedan unirse al proyecto final, **“PointWPF”**, éstas deben ser ingresadas a PointWPF mediante referencias, y en el código importar dichas librerías:



En la Arquitectura en tres capas, la utilizada en POINTSOFT, todas estas librerías se conformarían para representar a la capa intermedia (CAPA DE NEGOCIOS), aunque **LibDAL** también podría considerarse parte de la CAPA DE DATOS pues carece de la lógica del negocio. El objetivo de cada una de ellas se explicará a continuación:

4.2.3.2.1.1. *LibUtil.dll*.



Esta librería, como su nombre lo indica, proporciona utilidades para ser usadas en el proyecto, y que no son específicas de las Reglas del Negocio de POINTSOFT. Por tal motivo, esta librería será utilizada también en otros proyectos que requieran las utilidades que posee.

Consta de cinco secciones, que serán detalladas a continuación:

- **BDConexion.cs**

Tiene todos los métodos necesarios para poder realizar de manera correcta una conexión a la base de datos. Realiza los métodos para usar en los **DAL LibDALPoint.dll** de tal manera que el trabajo sea transparente y desglosado.

- **CedulaRUC.cs**

Esta clase se centra en la validación del ingreso de cédulas y RUC en cualquier parte del Programa.

```
public static int calcularDigitoVerificador(String numero)
```

Todo número de identificación se encuentra sujeto a una regla, donde el último dígito no es puesto al azar sino, más bien, es calculado respecto a los dígitos restantes.

Esta función devuelve este último número llamado Dígito Verificador

```
public static bool validarCedulaRUC(string cedRUC)
```

Determina el número de dígitos correctos, para una cédula (10 dígitos), para RUC (13 dígitos); luego, con la ayuda del método `calcularDigitoVerificador`, determina si el número ingresado (cédula o RUC) es correcto o no.

- **Singleton.cs**

El patrón de diseño singleton (instancia única) está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto.

Su intención consiste en garantizar que una clase sólo tenga una instancia y proporcionar un punto de acceso global a ella.

Su mayor aplicación se da en programas con múltiples hilos de ejecución, tal como trabaja .NET Framework. Si dos hilos de ejecución intentan crear la instancia al mismo tiempo y ésta no existe todavía, sólo uno de ellos debe lograr crear el objeto.

- **Validacion.cs**

Provee métodos para realizar validaciones a distintos datos de entrada, generalmente utilizando “expresiones regulares” o RegEx.

- **BaseExcel.cs**

Esta es la clase que utilizamos para poder realizar los reportes y migrarlos a Excel a partir de un DataTable previamente construido. Construye las columnas y va recorriendo filas para poder agregarlos a un nuevo archivo de Excel que puede, si se requiere, ser modificado y trabajado como cualquier archivo normal

Además, el formato de cada celda ha sido arreglado para aliviar el trabajo del usuario, siendo éste agradable y muy entendible.

	A	B	C
1	DETALLES DE PUNTO DE VENTA		
2			
3	CREADO POR: ROOT ROOT		
4	FECHA DE CREACION: lunes, 11 de enero de 2010		
5	HORA: 12:26 p.m.		
6			
7	CODIGO EMPRESA	NOMBRE	DIRECCION
8	001	PORTA LINK	H FERRIVIA S11-294 Y M. ALFARO
9	002	ESTUDIO FOTOGRAFICO	EL CANELO OE5-69 Y LEXO BRUIS
10	004	CONFITERIA QUINTA ESENCIA	ATUCUCHO CALLE I Y CALLE 16
11	005	FARMA AHORRO N° 30	AV PRENSA 2834 Y AV FLORIDA
12	006	IVONNE VIVERES II	IMBABURA N8-13 Y MANABI
13	009	JORGITO VIVERES II	CIUDADELA SEÑOR DEL ARBOL CALLE RAFAEL AULESTIA Y VALENZUELA
14	010	COMERCIAL LIDER - EGUSQUIZA	JOSE EGUSQUIZA OE6-44
15	011	JG CELL II	CALLE REINO DE QUITO S/N Y LA CURIA
16	012	VIVERES PAOLA	JOSE ANDRADE OE1-497 Y JOAQUIN MANCHENO
17	013	CARMITA PAPELERIA Y VARIEDADES II	CALLE 13 DE JUNIO LOTE 1220 LOCAL 4
18	014	AKIT VIVERES II	CONJUNTO MARIANITAS 3 LOCAL 2
19	015	MICRO ZAMURA	CALLE PURUHA CASA 2 Y URB ZARUMA
20	017	EL CIJAL DELI MARKET II	AV. GEOVANNY 1138 E INDEPENDENCIA
21	019	LA ECONOMIA MICRO II	CALLE COHAVI N69-148 Y MACHALA
22	020	DIMADI	MALVAS S26-142 Y PASAJE 32
23	021	MATEO VIVERES II	CALLE RIO TULIPE N70-07 Y MACHALA

4.2.3.2.1.2. LibDALPoint.dll.



En POINTSOFT, se crea esta librería con el objetivo de realizar la interconexión a la base de datos MySQL.

Cada **DAL** es una clase en la cual se presentan operaciones básicas que se realizan a cada objeto, respecto a la Base de Datos.

Ya que todas las **DALs** tienen un esquema muy similar, detallaremos un ejemplo para el entendimiento del código:

PaisDAL.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using LibUtil;
using MySql.Data.MySqlClient;
using LibBDPoint;
using System.Data;
```

Referencias a usar del sistema o ingresadas manualmente como **LibBDPoint**

```
#region Campos
private BDConexion _conx;

public enum Parametros
{
    Codigo, Nombre, Iso
}
#endregion
```

Los campos que tiene la clase pública PaisDAL son 2:

- Una conexión llamada **_conx** del tipo BDConexion referenciada de la librería **LibUtil**.
- Una lista de parámetros de búsqueda que posee la base de datos: Código del País (de la base), Nombre del País, Código Iso.

Clase estática para crear un parámetro de búsqueda de un País en base a dos criterios, devolviendo un MySqlParameter llamado **tipo**

```
public static MySqlParameter parameterSearch(Object searchParameter,
Parametros criterio)
{
    MySqlParameter tipo = new MySqlParameter();

    switch (criterio)
```

El criterio de búsqueda puede ser por Nombre o Código de la Base de Datos.

```

case Parametros.Nombre:
    if (searchParameter is String)
    {
        tipo.Value = "%" + (String)searchParameter + "%";
        tipo.ParameterName = "vparam";
        tipo.DbType = System.Data.DbType.String;
        return tipo;
    }
    else
    {
        throw new ArgumentException("El parámetro ingresado no
es del tipo System.Data.DbType.String", "searchParameter");
    }

```

Nombre: Primeramente verifica si se ha ingresado un tipo de dato "String".

Devuelve un **tipo** que es un **MySqlParameter** con el valor recogido de **searchParameter** con nombre **vparam**

```

case Parametros.Codigo:
    if (searchParameter is Int64 ||
        searchParameter is UInt64)
    {
        tipo.Value = Convert.ToInt64(searchParameter);
        tipo.ParameterName = "vparam";
        tipo.DbType = System.Data.DbType.Int64;
        return tipo;
    }
    else if (searchParameter is String)
    {
        tipo.Value = (String)searchParameter;
        tipo.ParameterName = "vparam";
        tipo.DbType = System.Data.DbType.String;
        return tipo;
    }
    else
    {
        throw new ArgumentException("El parámetro
ingresado no es del tipo System.Data.DbType.Int64 ó
System.Data.DbType.String", "searchParameter");
    }

```

De la misma manera que Nombre. **Código:** Verifica si se ha ingresado un "Int64" o un "String".

Recoge los datos ingresados para crear un **MySqlParameter** llamado **tipo**.

```

default:
    throw new NotImplementedException();
}
}

```

Devuelve un `MySQLParameter` (Nombre) al ingresar un **obj** del tipo `Pais`.
De igual manera existe uno para cada parámetro: Código ISO, Código.

```
public static MySQLParameter parameterNombre(Pais obj)
{
    MySQLParameter param = new MySQLParameter();
    param.Value = obj.nombre;
    param.ParameterName = "vnombre";
    param.DbType = System.Data.DbType.String;

    return param;
}
```

```
public void insert(Pais obj)
{
    List<MySQLParameter> parametros = new List<MySQLParameter>();

    parametros.Add(parameterNombre(obj));

    parametros.Add(parameterIso(obj));
```

Del **obj (Pais)** ingresado determina sus campos (CodigoIso y Nombre) y los convierte en parámetros.

Llama a su propiedad "conexión" para ejecutar en la base de datos el procedimiento almacenado `sp_insertarPais`.

De la misma manera similar, se programaron las operaciones de actualizar (**update**), y eliminar (**delete**).

```
try
{
    conx.exeNonQuery("sp_insertarPais", parametros);
}
```

```
catch (MySQLException ex)
{
    switch (ex.Number)
    {
        case 1062:
            if (ex.Message
                .ToLower()
                .Contains("un_nombre_pais"))
            {
                throw new ArgumentException("El país ' +
                    obj.nombre + "' ya existe.", "Nombre");
            }
            else
            {
                throw new ArgumentException("Error en la base
                    de datos\n" + ex.Message);
            }
    }
}
```

Se detectan errores específicos de MySQL para presentarlos al Usuario, y se controla si se violó alguna regla de la Base de Datos.

```

        default:
            throw new ArgumentException("Error en la base de
                datos\n" + ex.Message);
    }
}
}

```

Realiza una consulta devolviendo un País. Busca el País en donde pertenece esa provincia.

De la misma manera se presentan otras consultas como:

buscarLista(): Busca la lista de Países con o sin un Criterio (Nombre, códigoISO)

buscarListaProvincias(): Busca las Provincias que tiene un País

```

public Pais getPais(Provincia pro)
{
    Pais obj;

    MySqlCommand comm = new MySqlCommand();
    comm.CommandType = CommandType.StoredProcedure;
    comm.Connection = conx.conn;
    comm.CommandText = "sp_buscarPais_codigoProvincia";

    comm.Parameters.Add(parameterSearch(pro.codigo,
Parametros.Codigo));

    if (conx.isOpen())
    {
        MySqlDataReader reader = comm.ExecuteReader();
        reader.Read();

        if (reader.HasRows)
        {
            obj = createFromReader(reader);
            reader.Close();
        }
        else
        {
            reader.Close();
            throw new NullReferenceException("No existe Pais");
        }
    }
    else
    {
        throw new DataException("La base de datos esta cerrada");
    }

    return obj;
}

```

Determina si la conexión se encuentra abierta o cerrada.

Determina si la consulta ha devuelto algún valor.

4.2.3.2.1.3. *LibBDPoint.dll.*



LibBDPoint.dll
1.0.0.0
LibBDPoint

En POINTSOFT se crea la librería con el nombre **LibBDPoint.dll**, la cual lleva toda la definición de Clases del proyecto.

Estas clases ya se detallaron en el **Título “Especificación de Clases de Diseño.”**, **página 149.**

4.2.3.2.2. *Secuencia de Estados de Puntos de Venta.*

Actualmente, TELCOLSAM S.A. trabaja con pocos estados (Activo, Inactivo, Bloqueado y Bases); pero para tratar de evitar trabajo y equivocaciones, hemos definido un diagrama de estados que restrinja el proceso de acuerdo a la Clase de usuario.

4.2.3.2.2.1. *Definición de Estados de Punto de Venta.*

- **Preparando Instalación:** Cuando el PDV es recientemente creado, automáticamente se pondrá en este estado. Ya que aún no tiene instalación alguna.
- **Instalación Activo:** El PDV se encuentra en este estado cuando ya se ha instalado las cabinas/cubículos que ha solicitado.
- **Ruta-Recaudador:** Cuando al PDV se le asigna una Ruta y un Recaudador para poder ser visitado y poder recaudar el dinero respectivo. **NO** se le considera un Estado en sí, solamente se lo toma como un requisito para poder pasar al Estado Activo.
- **Activo:** Cuando al PDV ya se le ha asignado una Ruta, un Recaudador y ya tiene instalados sus equipos.

- **Bloqueado:** Cuando el PDV ha dejado de incumplir con su deber de entregar el dinero. Generalmente, se bloquea a un PDV cuando el mismo llega a tres ofrecimientos de pago.
- **Inactivo:** Cuando la empresa ha tomado la decisión de retirar los equipos.
- **Bases:** Cuando el PDV ya ha sido retirado sus equipos y dichos equipos se encuentran a disposición para que puedan ser usados en un nuevo PDV.
- **Legal:** Cuando la empresa ha decidido llevar un proceso legal a los dueños del PDV. Lo más común es seguir este proceso porque el PDV no ha saldado a totalidad sus deudas.
- **Cobranco:** Cuando el PDV ofrece pagar el dinero adeudado, y se encuentra en proceso de recaudación.
- **Terminado:** Cuando el proceso de Legal o Cobranco en que se encontraba el PDV ha llegado a su fin.

4.2.3.2.2.2. *Diagrama de Estados de Punto de Venta.*

El empleado de Instalaciones es el encargado de crear el PDV. Automáticamente, sin ningún proceso del empleado, pasa al estado de “**Preparando Instalación**”. Estando en este estado, su único posible cambio de estado es “**Instalación Activo**”, ya que “**Ruta/Recaudador**” no es un estado.

Para poder pasar a un estado de “**Activo**”, es necesario que el PDV ya tenga asignada una Ruta y un Recaudador, y además que ya le hayan instalado los equipos pedidos a la empresa (Estado Instalación Activo). Por ejemplo, si el PDV

solamente ha sido instalado, pero no se le ha asignado una Ruta ni un Recaudador, no podrá pasar al estado Activo.

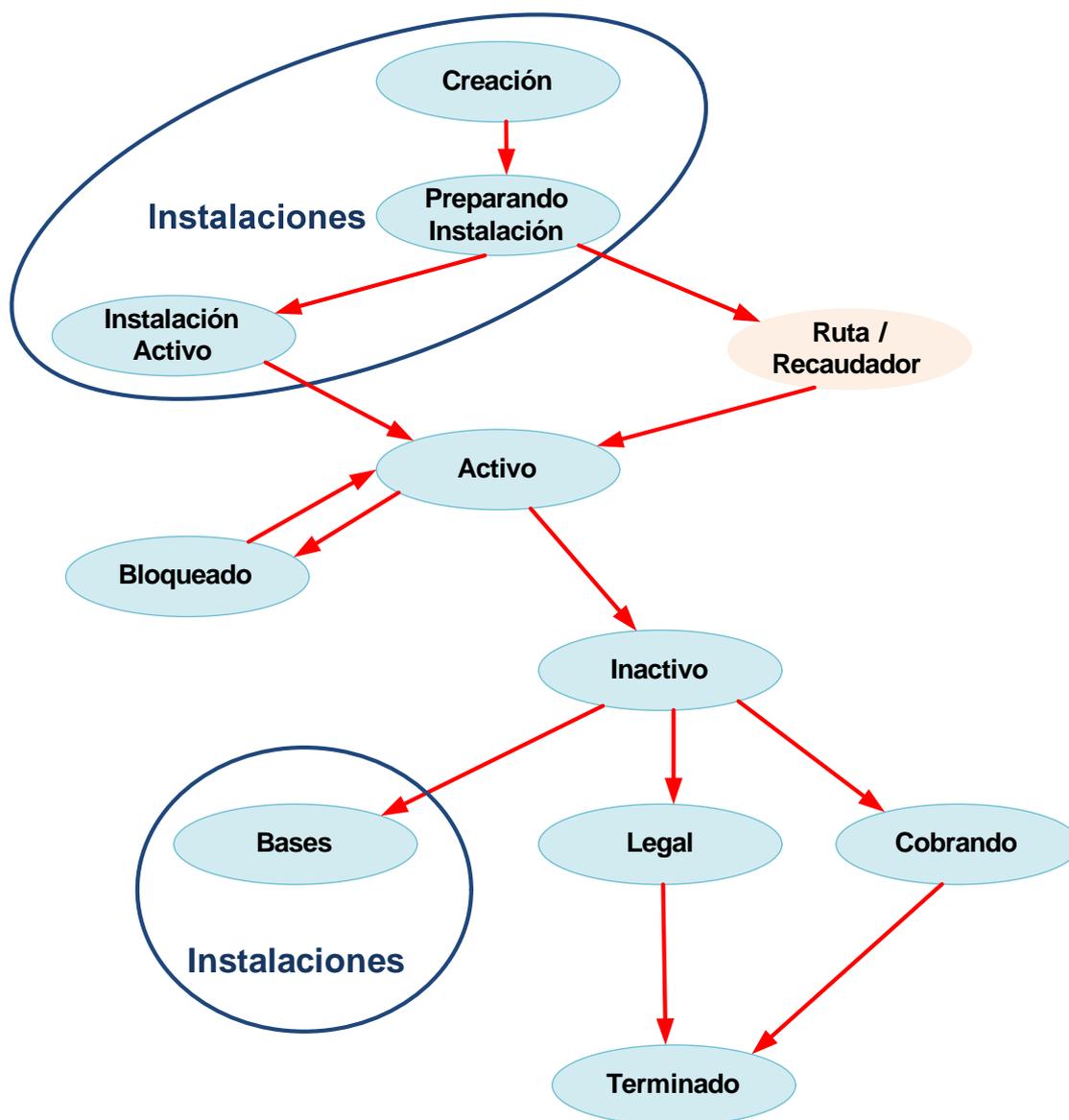


Figura 4.42: Diagrama de Estados de un Punto de Venta.

Si el PDV se encuentra en un estado **“Activo”** se le presentan dos opciones: **“Bloqueado”** e **“Inactivo”**. Un PDV puede pasar cuantas veces sean necesarias del estado **“Activo”** a **“Bloqueado”**, pero ahora la empresa podrá saber cuántas

veces el PDV ha sido bloqueado en su tiempo de vida, puesto que antes no podían hacer ese seguimiento.

Estando en el estado “**Bloqueado**” solamente se presentará la opción de “**Activo**”, y de igualmente pasará al otro estado cuantas veces sean requeridas.

Cuando el PDV se encuentre en estado “**Inactivo**” se presentarán tres opciones: “**Bases**”, “**Legal**”, “**Cobrando**”. Siendo para el estado “**Bases**” su último estado posible.

Estando en los estados “**Legal**” o “**Cobrando**”, su único cambio posible de estado es “**Terminado**”.

Por motivos de corrección de errores de tipo humano, un Administrador puede establecer Estados a un PDV sin seguir la secuencia lógica, pero con ciertas restricciones.

4.2.3.3. Capa Interfaz Gráfica.

4.2.3.3.1. Diseño de Interfaces del Sistema.

Hoy por hoy, los sistemas se caracterizan por dar mejores interfaces al usuario, y hacer que el usuario se sienta seguro en un sistema intuitivo y amigable es, sin duda, uno de los principales objetivos de nuestro proyecto.

Ya que TELCOLSAM S.A. se desarrolla bajo un SO Windows, y piensa mantenerse en ese mismo sistema, solamente con un cambio de versión (Windows Vista o Windows 7), hemos creído conveniente adoptar muchas de las ventajas que éste proporciona. Por tanto, la interfaz gráfica se basará en Ribbon, anteriormente explicado en el **Capítulo 3, Título “Ribbon”, página 111.**

4.2.3.3.2. Navegación.

A continuación, se presentan los diagramas de flujo que representan la manera de navegar bajo POINTSOFT, detallando índices y subíndices.

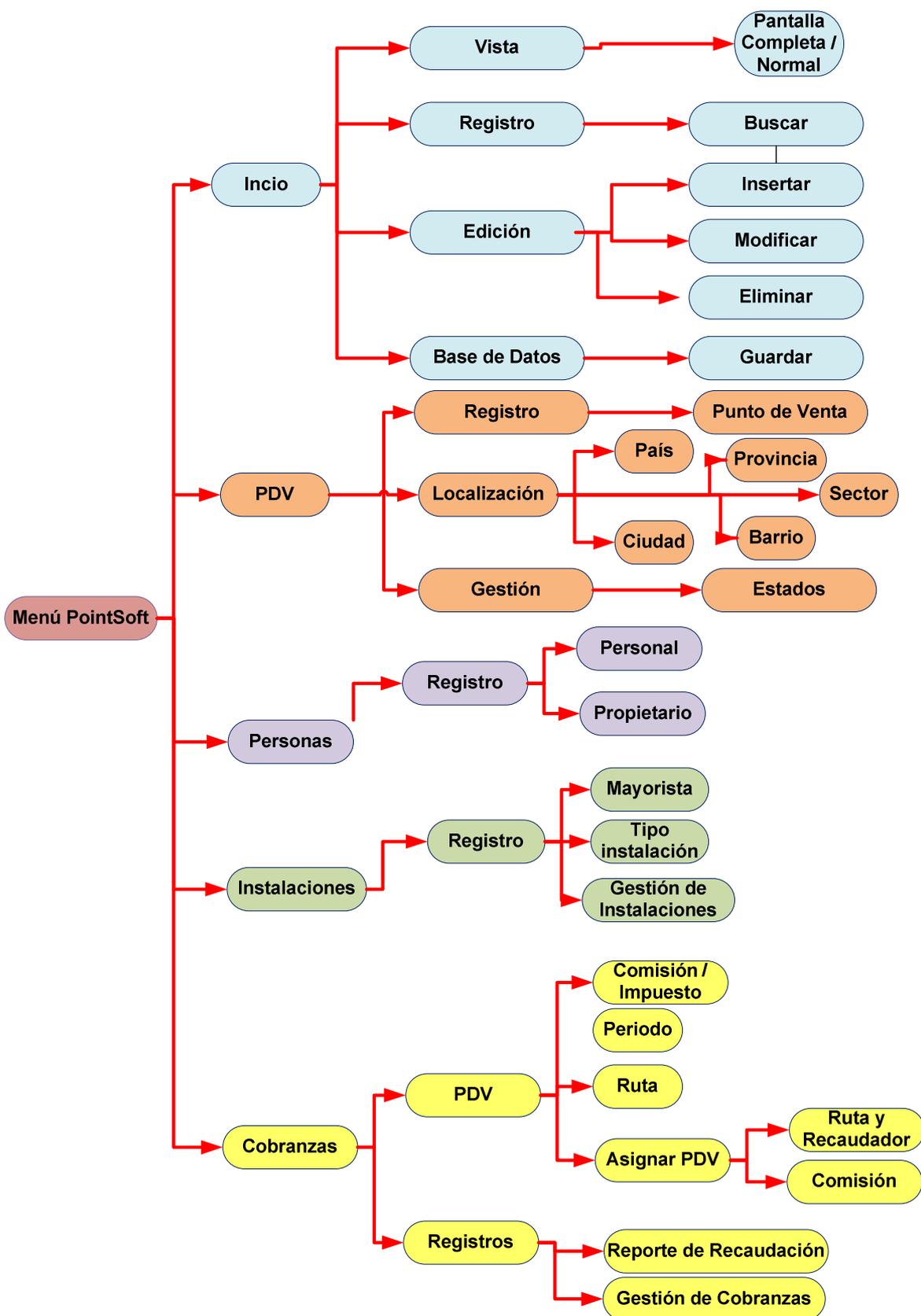


Figura 4.43: Diagrama de Navegación, POINTSOFT.

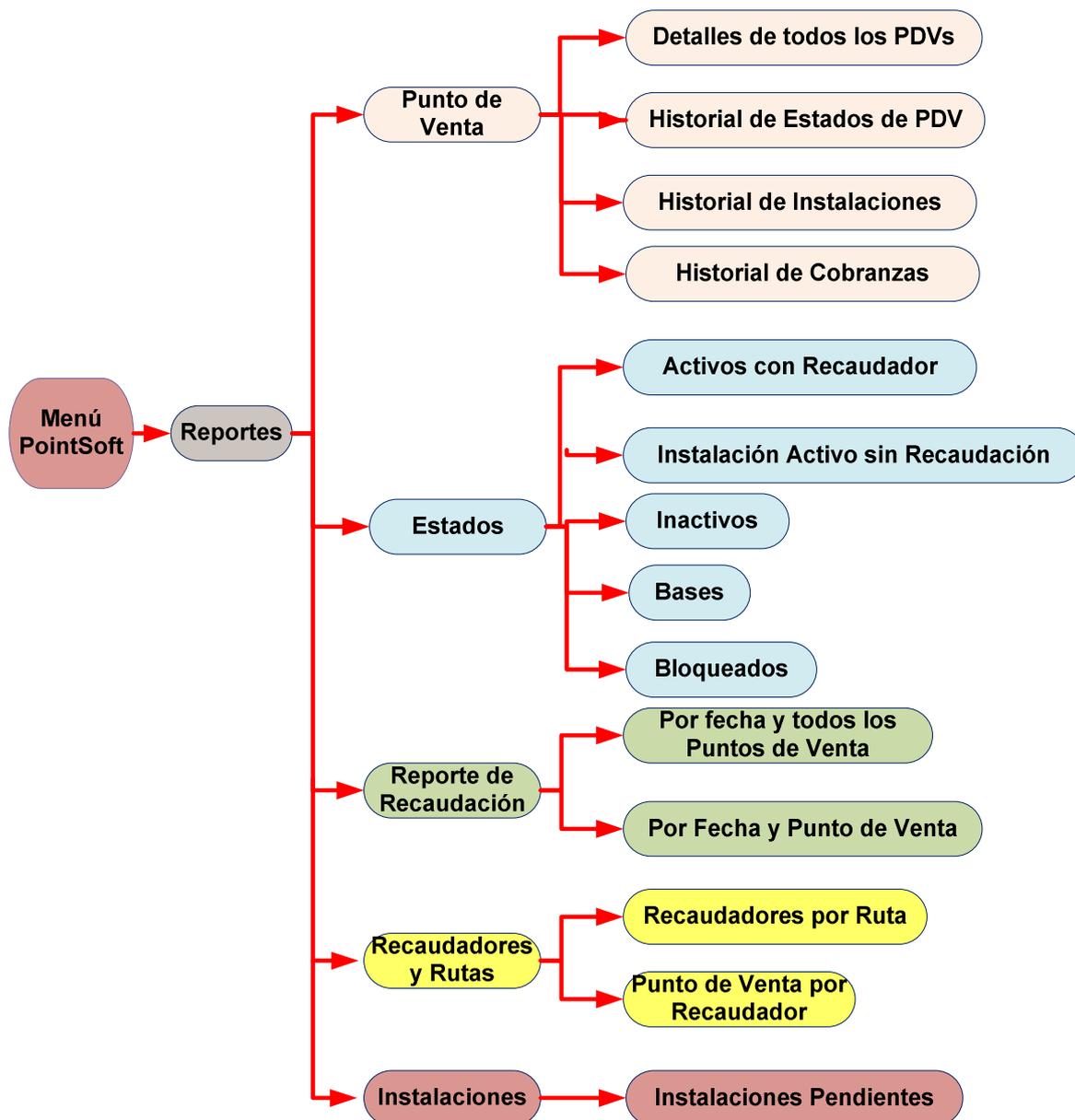


Figura 4.44: Diagrama de Navegación de Reportes, POINTSOFT.

4.2.3.3.3. Descripción de Botones Principales.

POINTSOFT se presenta bajo una pantalla muy similar al Office 2007 de Microsoft. Cada índice se encuentra en una pestaña diferente, y cuando se ingrese a cada una de éstas, los subíndices aparecerán bajo las pestañas divididos en Temas:



Cada subíndice, dependiendo de sus requerimientos, presentará cinco botones básicos: Buscar, Insertar, Modificar, Eliminar, y Guardar:



Por tanto, cada pestaña permitirá el uso de tres a cinco botones (BT), según se especifica, que realizarán las siguientes tareas:

- BT-B:** Botón Buscar.- Inicializará la pestaña y actualizará los campos de selección, con los datos más recientes de la base de datos central.
- BT-I:** Botón Insertar.- Inicializará los campos para permitir el ingreso de un nuevo registro. El botón permanecerá chequeado mientras se encuentre en el estado "Insertar".
- BT-M:** Botón Modificar.- Permitirá la modificación del registro actual. El botón permanecerá chequeado mientras se encuentre en el estado "Modificar".
- BT-E:** Botón Eliminar.- Permitirá la eliminación del registro actual. El botón permanecerá chequeado mientras se encuentre en el estado "Eliminar".
- BT-G:** Botón Guardar.- Se comunicará con la base de datos y ejecutará las acciones de los botones BT-A, BT-M, Y BT-E.

NOTA: Con el fin de prevenir errores, estos botones siguen las siguientes reglas:

- El botón Buscar siempre se encontrará habilitado para todas las pestañas.
- El botón Guardar, será usado para guardar datos que hayan sido ingresados, modificados o eliminados. Sin embargo, para todos los casos, el botón se habilitará siempre y cuando la validación de la pestaña haya sido exitosa.

4.2.3.3.4. Descripción de Pantallas.

La ventana principal, **Figura 4.45**, contiene un TabContainer que albergará cada una de las pestañas donde se realizará la gestión de los paquetes.

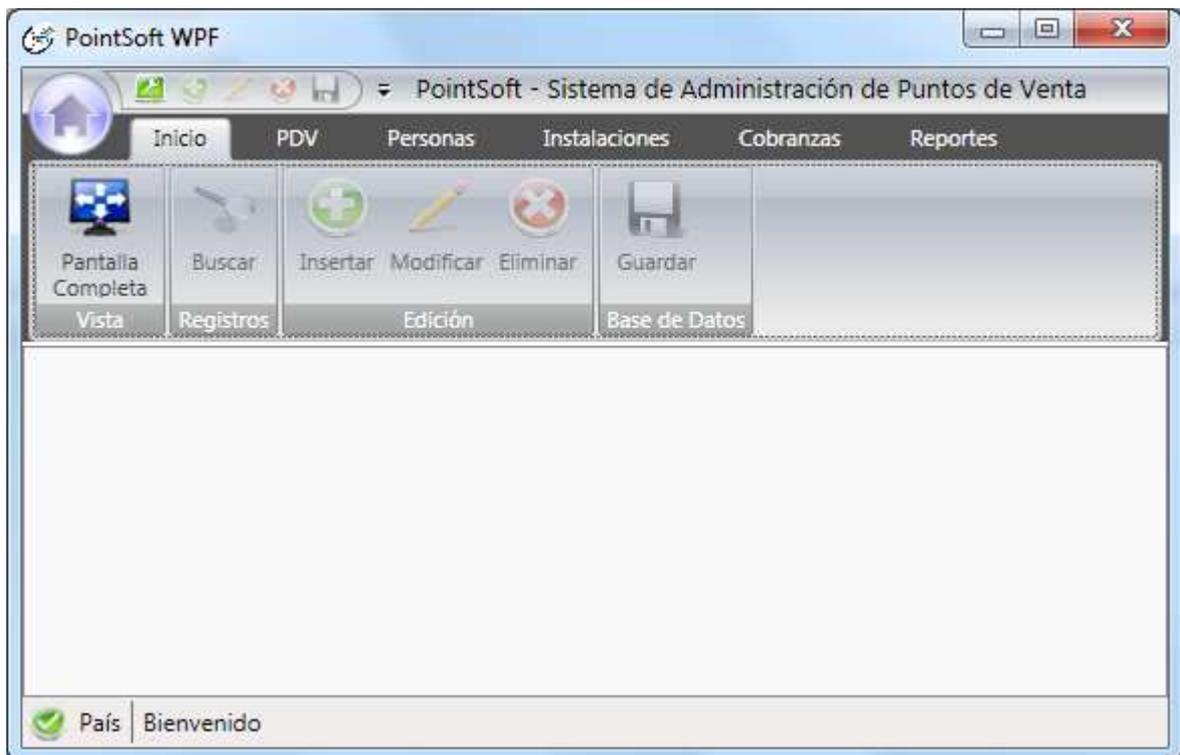


Figura 4.45: Ventana principal de POINTSOFT.

País:
ECUADOR

Código ISO:

Nombre:

Todas las pantallas, correspondientes a un subíndice (Ej. País), tendrán campos validados con el fin de evitar posibles errores de digitación. Entre los campos más importantes validados se encuentran:

- Cédulas y Ruc.
- Teléfonos y Fax
- Códigos.
- Contraseñas.
- Horas y Fechas, etc.

A continuación presentamos la descripción de cada una de las ventanas correspondientes a POINTSOFT.

- **Ventana Principal.**

VENTANA PRINCIPAL	
Propiedad	Características
Tipo de Interfaz	Ventana
Menús	Menú de Inicio
Barras	Barra Ribbon Barra de Estado
Botones Activos	BT-B; BT-I; BT-M; BT-E; BT-G.
Resolución mínima de pantalla	1024x768 pixeles.
Características de color	A criterio del desarrollador.
Consideraciones adicionales	Validaciones de ingreso dependiendo del perfil que haya ingresado.

Tabla 4.42: Descripción Gráfica Ventana Principal.

- Pestaña “País”.

PESTAÑA PAÍS	
Propiedad	Características
Tipo de Interfaz	UserControl sobre un TabItem
Menús	Ninguno.
Botones Activos	BT-B; BT-I; BT-M; BT-E; BT-G.
Resolución mínima de pantalla	1024x768 pixeles.
Características de color	A criterio del desarrollador.
Consideraciones adicionales	Validaciones de ingreso y modificación de datos.

Tabla 4.43: Descripción Gráfica Pestaña País.

Campo	Descripción	Características	Obligatorio ³⁰
Código ISO	Código asignado por la ISO.	Dos caracteres alfabéticos.	No
Nombre	Nombre del país.	Caracteres alfabéticos.	Sí.

Tabla 4.44: Descripción de campos en la Pestaña País.

- Pestaña “Provincia”.

PESTAÑA PROVINCIA	
Propiedad	Características
Tipo de Interfaz	UserControl sobre un TabItem
Menús	Ninguno.
Botones Activos	BT-B; BT-I; BT-M; BT-E; BT-G.
Resolución mínima de pantalla	1024x768 pixeles.
Características de color	A criterio del desarrollador.
Consideraciones adicionales	Validaciones de ingreso y modificación de datos.

Tabla 4.45: Descripción Gráfica Pestaña Provincia.

Campo	Descripción	Características	Obligatorio
Nombre	Nombre de la provincia.	Caracteres alfabéticos.	Sí.

Tabla 4.46: Descripción de campos en la Pestaña Provincia.

³⁰ Se refiere a que si es necesario el ingreso de ese campo, por el usuario, para efectos de ingreso o modificación de registros.

- Pestaña “Ciudad”.

PESTAÑA CIUDAD	
Propiedad	Características
Tipo de Interfaz	UserControl sobre un TabItem
Menús	Ninguno.
Botones Activos	BT-B; BT-I; BT-M; BT-E; BT-G.
Resolución mínima de pantalla	1024x768 pixeles.
Características de color	A criterio del desarrollador.
Consideraciones adicionales	Validaciones de ingreso y modificación de datos.

Tabla 4.47: Descripción Gráfica Pestaña Ciudad.

Campo	Descripción	Características	Obligatorio
Código Descriptivo	Código que identifica de manera corta a la ciudad.	Tres caracteres alfabéticos.	No
Nombre	Nombre de la Ciudad.	Caracteres alfabéticos.	Sí.

Tabla 4.48: Descripción de campos en la Pestaña Ciudad.

- Pestaña “Sector”.

PESTAÑA SECTOR	
Propiedad	Características
Tipo de Interfaz	UserControl sobre un TabItem
Menús	Ninguno.
Botones Activos	BT-B; BT-I; BT-M; BT-E; BT-G.
Resolución mínima de pantalla	1024x768 pixeles.
Características de color	A criterio del desarrollador.
Consideraciones adicionales	Validaciones de ingreso y modificación de datos.

Tabla 4.49: Descripción Gráfica Pestaña Sector.

Campo	Descripción	Características	Obligatorio
Nombre	Nombre del Sector.	Caracteres alfabéticos.	Sí.

Tabla 4.50: Descripción de campos en la Pestaña Sector.

- Pestaña “Barrio”.

PESTAÑA BARRIO	
Propiedad	Características
Tipo de Interfaz	UserControl sobre un TabItem
Menús	Ninguno.
Botones Activos	BT-B; BT-I; BT-M; BT-E; BT-G.
Resolución mínima de pantalla	1024x768 pixeles.
Características de color	A criterio del desarrollador.
Consideraciones adicionales	Validaciones de ingreso y modificación de datos.

Tabla 4.51: Descripción Gráfica Pestaña Barrio.

Campo	Descripción	Características	Obligatorio
Nombre	Nombre del barrio.	Caracteres alfanuméricos.	Sí.

Tabla 4.52: Descripción de campos en la Pestaña Barrio.

- Pestaña “Propietario”.

PESTAÑA PROPIETARIO	
Propiedad	Características
Tipo de Interfaz	UserControl sobre un TabItem
Menús	Ninguno.
Botones Activos	BT-B; BT-I; BT-M; BT-E; BT-G.
Resolución mínima de pantalla	1024x768 pixeles.
Características de color	A criterio del desarrollador.
Consideraciones adicionales	Validaciones de ingreso y modificación de datos.

Tabla 4.53: Descripción Gráfica Pestaña Propietario.

Campo	Descripción	Características	Obligatorio
Cédula / R.U.C.	Número de cédula asignado por el Registro Civil, o número de R.U.C. asignado por el S.R.I.	Código numérico de 10 ó 13 caracteres. Requiere validación.	Sí.
Razón Social	Nombre comercial con el que se asocia al cliente.	Caracteres alfanuméricos.	Sí.
Nombres	Nombres del propietario.	Caracteres alfabéticos.	Sí.

Primer Apellido	Primer apellido del propietario.	Caracteres alfabéticos.	Sí.
Segundo Apellido	Segundo apellido del propietario.	Caracteres alfabéticos.	No.
Dirección 1	Dirección de residencia.	Caracteres alfanuméricos.	Sí.
Dirección 2	Dirección alterna.	Caracteres alfanuméricos.	No.
Teléfono 1	Teléfono fijo.	Caracteres numéricos.	Sí.
Teléfono 2	Teléfono fijo alterno.	Caracteres numéricos.	No.
Teléfono Celular	Teléfono móvil.	Caracteres numéricos.	No.
Fax	Teléfono fax.	Caracteres numéricos.	No.

Tabla 4.54: Descripción de campos en la Pestaña Propietario.

- **Pestaña “Punto de Venta”.**

PESTAÑA PUNTO DE VENTA	
Propiedad	Características
Tipo de Interfaz	UserControl sobre un TabItem
Menús	Ninguno.
Botones Activos	BT-B; BT-I; BT-M; BT-E; BT-G.
Resolución mínima de pantalla	1024x768 pixeles.
Características de color	A criterio del desarrollador.
Consideraciones adicionales	Validaciones de ingreso y modificación de datos.

Tabla 4.55: Descripción Gráfica Pestaña Punto de Venta.

Campo	Descripción	Características	Obligatorio
Código	Código de empresa del PDV	Es un código numérico de 3 o 4 caracteres, proporcionado por el Servicio de Rentas Internas (SRI).	Sí.
Nombre	Nombre comercial del PDV.	No identifica de manera unívoca al local.	Sí.
Propietario	Listado de propietarios.	Lista desplegable.	Sí.
País	País de locación del PDV.	Lista desplegable.	Sí.
Provincia	Provincia dentro del país de locación del PDV.	Lista desplegable.	Sí.

Continúa en la página 197.

Ciudad	Ciudad dentro de la provincia de locación del PDV.	Lista desplegable.	Sí.
Sector	Sector de locación del PDV.	Lista desplegable.	Sí.
Barrio	Barrio dentro de la ciudad de locación del PDV.	Lista desplegable.	Sí.
Dirección	Calle principal, número, e intersección del PDV.	Caracteres alfanuméricos.	Sí.
Referencia	Datos adicionales para ubicar al PDV.	Caracteres alfanuméricos.	No.
Teléfono	Teléfono fijo del PDV.	Caracteres numéricos.	Sí.
Fax	Teléfono fax del PDV.	Caracteres numéricos.	No.
Teléfono Celular	Teléfono móvil del PDV.	Caracteres numéricos.	No.
Observaciones	Observaciones adicionales sobre el PDV.	Caracteres alfanuméricos y especiales.	No.

Tabla 4.56: Descripción de campos en la Pestaña Punto de Venta.

- **Pestaña “Mayorista”.**

PESTAÑA MAYORISTA	
Propiedad	Características
Tipo de Interfaz	UserControl sobre un TabItem
Menús	Ninguno.
Botones Activos	BT-B; BT-I; BT-M; BT-E; BT-G.
Resolución mínima de pantalla	1024x768 pixeles.
Características de color	A criterio del desarrollador.
Consideraciones adicionales	Validaciones de ingreso y modificación de datos.

Tabla 4.57: Descripción Gráfica Pestaña Mayorista.

Campo	Descripción	Características	Obligatorio
Código	Abreviatura del mayorista.	Dos caracteres alfabéticos.	Sí.
Nombre	Nombre del mayorista.	Caracteres alfabéticos.	Sí.

Tabla 4.58: Descripción de campos en la Pestaña Mayorista.

- Pestaña “Tipo de Instalación”.

PESTAÑA TIPO DE INSTALACIÓN	
Propiedad	Características
Tipo de Interfaz	UserControl sobre un TabItem
Menús	Ninguno.
Botones Activos	BT-B; BT-I; BT-M; BT-E; BT-G.
Resolución mínima de pantalla	1024x768 pixeles.
Características de color	A criterio del desarrollador.
Consideraciones adicionales	Validaciones de ingreso y modificación de datos.

Tabla 4.59: Descripción Gráfica Pestaña Tipo Instalación.

Campo	Descripción	Características	Obligatorio
Nombre	Tipo de instalación.	Caracteres alfabéticos.	Sí.

Tabla 4.60: Descripción de campos de la Pestaña Tipo Instalación.

- Pestaña “Instalación”.

PESTAÑA INSTALACIÓN	
Propiedad	Características
Tipo de Interfaz	UserControl sobre un TabItem
Menús	Ninguno.
Botones Activos	BT-B; BT-I; BT-M; BT-E; BT-G.
Resolución mínima de pantalla	1024x768 pixeles.
Características de color	A criterio del desarrollador.
Consideraciones adicionales	Validaciones de ingreso y modificación de datos.

Tabla 4.61: Descripción Gráfica Pestaña Instalación.

Campo	Descripción	Características	Obligatorio
Tipo de Instalación.	Listado de tipos de instalaciones.	Lista desplegable.	Sí.
Bases a usar.	Listado de PDVs que estén en estado de bases.	Lista desplegable.	No.
Mayorista	Listado de mayoristas.	Lista desplegable.	Sí.
Fecha de Cabina	Fecha de instalación de la cabina.	Tipo fecha.	Sí.

Hora de Cabina	Hora de instalación de la cabina, en la fecha especificada.	Caracteres numéricos. Requiere validación.	No.
Fecha de Equipos	Fecha de instalación de los equipos, una vez instalada la cabina.	Tipo fecha.	No.
Hora de Equipos	Hora de instalación de los equipos, en la fecha especificada.	Caracteres numéricos. Requiere validación.	No.
Descripción	Descripción de la Instalación.	Caracteres alfanuméricos y especiales.	No.

Tabla 4.62: Descripción de campos de la Pestaña Instalación.

- Pestaña “Gestión de Cobranzas”.

PESTAÑA GESTIÓN DE COBRANZAS	
Propiedad	Características
Tipo de Interfaz	UserControl sobre un TabItem
Menús	Ninguno.
Botones Activos	BT-B; BT-I; BT-M; BT-E; BT-G.
Resolución mínima de pantalla	1024x768 pixeles.
Características de color	A criterio del desarrollador.
Consideraciones adicionales	Validaciones de ingreso y modificación de datos.

Tabla 4.63: Descripción Gráfica Pestaña Gestión de Cobranzas.

Campo	Descripción	Características	Obligatorio
Lista de Puntos de Venta	Listado de los PDV existentes	Lista desplegable.	Sí.
Lista de Gestiones	Listado de las Gestiones pertenecientes al PDV escogido	Lista Desplegable.	Sí.
Descripción	Descripción de la Gestión realizada	Caracteres alfanuméricos y/o especiales.	No
Fecha	Fecha de la Gestión realizada	Tipo Fecha.	Sí.
Valor a Pagar	Cantidad de dinero por la que PDV tiene una Gestión	Carácter Numérico	No.

Tabla 4.64: Descripción de campos de la Pestaña Gestión.

- Pestaña “Factor de valor”.

PESTAÑA FACTOR DE VALOR	
Propiedad	Características
Tipo de Interfaz	UserControl sobre un TabItem
Menús	Ninguno.
Botones Activos	BT-B; BT-I; BT-M; BT-E; BT-G.
Resolución mínima de pantalla	1024x768 pixeles.
Características de color	A criterio del desarrollador.
Consideraciones adicionales	Validaciones de ingreso y modificación de datos.

Tabla 4.65: Descripción Gráfica Pestaña Factor Valor.

Campo	Descripción	Características	Obligatorio
Tipo de factor de valor.	Listado de tipos de factor de valor. Puede ser únicamente: Impuesto ó Comisión.	Lista desplegable.	Sí.
Fecha de Creación	Fecha de creación del factor de valor.	Tipo fecha.	Sí.
Nombre	Nombre identificativo del factor de valor.	Caracteres alfanuméricos.	Sí.
Factor	Factor de multiplicación en porcentaje.	Caracteres numéricos	Sí.

Tabla 4.66: Descripción de campos de la Pestaña Factor Valor.

- Pestaña “Período”.

PESTAÑA PERIODO	
Propiedad	Características
Tipo de Interfaz	UserControl sobre un TabItem
Menús	Ninguno.
Botones Activos	BT-B; BT-I; BT-M; BT-E; BT-G.
Resolución mínima de pantalla	1024x768 pixeles.
Características de color	A criterio del desarrollador.
Consideraciones adicionales	Validaciones de ingreso y modificación de datos.

Tabla 4.67: Descripción Gráfica Pestaña Periodo.

Campo	Descripción	Características	Obligatorio
Frecuencia	Nombre de la frecuencia con la que se realizará el recorrido.	Caracteres alfabéticos.	Sí.

Tabla 4.68: Descripción de campos de la Pestaña Periodo.

- Pestaña “Ruta”.

PESTAÑA RUTA	
Propiedad	Características
Tipo de Interfaz	UserControl sobre un TabItem
Menús	Ninguno.
Botones Activos	BT-B; BT-I; BT-M; BT-E; BT-G.
Resolución mínima de pantalla	1024x768 pixeles.
Características de color	A criterio del desarrollador.
Consideraciones adicionales	Validaciones de ingreso y modificación de datos.

Tabla 4.69: Descripción Gráfica Pestaña Ruta.

Campo	Descripción	Características	Obligatorio
Código	Código identificativo de la ruta.	Tres caracteres numéricos	Sí.
Frecuencia	Listado de frecuencias.	Lista desplegable.	Sí.
Días de la semana.	Días de la semana, comprendidos desde el lunes hasta el viernes.	Selección múltiple.	Sí.

Tabla 4.70: Descripción de campos de la Pestaña Ruta.

- Pestaña “Comisión por PDV”.

PESTAÑA COMISIÓN POR PDV	
Propiedad	Características
Tipo de Interfaz	UserControl sobre un TabItem
Menús	Ninguno.
Botones Activos	BT-B; BT-M; BT-G.
Resolución mínima de pantalla	1024x768 pixeles.
Características de color	A criterio del desarrollador.
Consideraciones adicionales	Validaciones de ingreso y modificación de datos.

Tabla 4.71: Descripción Gráfica Pestaña Comisión por PDV.

Campo	Descripción	Características	Obligatorio
Punto de Venta	Listado de Puntos de Venta	Lista. Requiere filtros para agilizar el acceso.	Sí.
Comisión	Listado de factores de valor que sean comisiones.	Lista desplegable.	Sí.
Fecha de asignación.	Fecha en la que se le asigna el factor de valor al PDV.	Tipo fecha.	Sí.

Tabla 4.72: Descripción de campos en la Pestaña Comisión por PDV.

- Pestaña “Ruta y Recaudador por PDV”.

PESTAÑA RUTA Y RECAUDADOR POR PDV	
Propiedad	Características
Tipo de Interfaz	UserControl sobre un Tabltem
Menús	Ninguno.
Botones Activos	BT-B; BT-M; BT-G.
Resolución mínima de pantalla	1024x768 pixeles.
Características de color	A criterio del desarrollador.
Consideraciones adicionales	Validaciones de ingreso y modificación de datos.

Tabla 4.73: Descripción Gráfica Pestaña Ruta y Recaudador por PDV.

Campo	Descripción	Características	Obligatorio
Punto de Venta	Listado de Puntos de Venta	Lista. Requiere filtros para agilizar el acceso.	Sí.
Recaudador	Listado del personal de TELCOLSAM S.A. que sea recaudador.	Lista desplegable.	Sí.
Ruta.	Listado de rutas.	Lista desplegable.	Sí.

Tabla 4.74: Descripción de campos de Pestaña Ruta y Recaudador por PDV.

- Pestaña “Reporte de Recaudación”.

PESTAÑA REPORTE RECAUDACIÓN	
Propiedad	Características
Tipo de Interfaz	UserControl sobre un Tabltem
Menús	Ninguno.
Botones Activos	BT-B; BT-I; BT-M; BT-E; BT-G.
Resolución mínima de pantalla	1024x768 pixeles.
Características de color	A criterio del desarrollador.
Consideraciones adicionales	Validaciones de ingreso y modificación de datos. Acceso rápido de teclado al botón guardar.

Tabla 4.75: Descripción Gráfica Pestaña Reporte de Recaudación.

Campo	Descripción	Características	Obligatorio
Saldo del PDV	Es el saldo que el PDV adeuda a TELCOLSAM S.A.	Caracteres numéricos. Es auto-calculado.	Sí.
Fecha Inicio	Es la fecha del último cobro de un reporte.	Tipo fecha. Es auto-calculado.	Sí.
Fecha Final	Es la fecha en la que se cobró este reporte.	Tipo fecha.	Sí.
Número de Reporte	Es un código propio de los reportes que lo da el sistema del SAR-II.	Caracteres alfanuméricos.	Sí.
Ticket desde	Es el último ticket que se cobró.	Caracteres numéricos. Es auto-calculado.	Sí.
Ticket hasta	Es ticket hasta el que se cobra actualmente.	Caracteres numéricos.	Sí.
Subtotal	Es el dinero adeudado antes de aplicarle factores de valor.	Caracteres numéricos.	Sí.
Dinero IVA	Es el dinero por concepto de Impuesto al Valor Agregado.	Caracteres numéricos.	Sí.
Total	Es el subtotal más el dinero IVA.	Caracteres numéricos. Es auto-calculado.	Sí.
Comisión	Es el dinero por concepto de comisión que recibe el PDV.	Caracteres numéricos. Es auto-calculado.	Sí.
Total a recaudar	Es el total menos la comisión.	Caracteres numéricos. Es auto-calculado.	Sí.
Pagado	Es el dinero que pagó el PDV.	Caracteres numéricos.	Sí.
Observaciones	Campo para ingresar información adicional sobre el reporte.	Caracteres alfanuméricos.	No.

Tabla 4.76: Descripción de Campos de la Pestaña Reporte de Recaudación.

Cabe destacar que el Dinero IVA no es un campo auto-calculado. Esto se debe a que el equipo que emite los reportes de recaudación contabiliza cada una de las llamadas que se realiza en las cabinas. Dado el hecho de que pueden existir

llamadas cuyo costo final sea menor a 0,09 dólares americanos (USD), estas generarían un IVA inferior a 0,01 USD y cuyo IVA se contabilizaría como 0,00 USD. En el caso extremo de que el PDV sólo receipte llamadas con un costo menor a 9,00 USD, la suma total del IVA a recaudar sería 0,00 USD. El equipo realiza un ajuste al almacenar como IVA cantidades inferiores a 0,01 USD, por lo que el resultado final del IVA puede ser mayor que el cálculo convencional del IVA al subtotal, como se especifica en la **Fórmula 1**:

$$D_{IVA} \geq S * f_{IVA}$$

Fórmula 1: Relación entre el IVA facturado y el IVA calculado.

Donde:

D_{IVA}: Dinero IVA que factura el equipo.

S: Subtotal que factura el equipo.

f_{IVA}: Factor de multiplicación para el IVA.

- Pestaña "Personal".

PESTAÑA PERSONAL	
Propiedad	Características
Tipo de Interfaz	UserControl sobre un TabItem
Menús	Ninguno.
Botones	BT-B; BT-I; BT-M; BT-E; BT-G.
Resolución mínima de pantalla	1024x768 pixeles.
Características de color	A criterio del desarrollador.
Consideraciones adicionales	Validaciones de ingreso y modificación de datos.

Tabla 4.77: Descripción Gráfica Pestaña Personal.

Campo	Descripción	Características	Obligatorio
Tipo	Listado con los tipos de personal. Se definieron cuatros tipos: Instalaciones, Cobranzas, Recaudador, y Administrador.	Lista desplegable.	Sí
Código	Código identificativo dentro de la empresa.	Tres caracteres alfabéticos.	Sí
Cédula	Número de cédula asignado por el Registro Civil.	Código numérico de 10 caracteres. Requiere validación.	Sí.
Nombres	Nombres del propietario.	Caracteres alfabéticos.	Sí.
Primer Apellido	Primer apellido del propietario.	Caracteres alfabéticos.	Sí.
Segundo Apellido	Segundo apellido del propietario.	Caracteres alfabéticos.	No.
Dirección 1	Dirección de residencia.	Caracteres alfanuméricos.	Sí.
Dirección 2	Dirección alterna.	Caracteres alfanuméricos.	No.
Teléfono 1	Teléfono fijo.	Caracteres numéricos.	Sí.
Teléfono 2	Teléfono fijo alterno.	Caracteres numéricos.	No.
Teléfono Celular	Teléfono móvil.	Caracteres numéricos.	No.
Fax	Teléfono fax.	Caracteres numéricos.	No.

Tabla 4.78: Descripción de campos de la ventana Personal.

- **Pestaña “Reportes”.**

PESTANA REPORTES	
Propiedad	Características
Tipo de Interfaz	UserControl sobre un Tabltem
Menús	Ninguno.
Botones	BT-B; BT-I; BT-M; BT-E; BT-G.
Resolución mínima de pantalla	1024x768 pixeles.
Características de color	A criterio del desarrollador.
Consideraciones adicionales	Validaciones de ingreso y modificación de datos.

Tabla 4.79: Descripción Gráfica Pestaña Personal.

NOTA: Dependiendo del Tipo de Reporte a abrir, habrá diversos métodos de búsqueda como por ejemplo:

- Por código y nombre de Punto de Venta.
- Por fecha.
- Por rango de Fecha.
- Por Recaudador.

4.2.3.3.5. Descripción del Controles ausentes en WPF.

Uno de los problemas que se encontró en WPF, es la falta de algunos controles que para Windows Forms resultaban muy importantes. Para nuestra aplicación, dos controles se implementaron con clases de terceros: DatePicker y MaskedTextBox.

4.2.3.3.5.1. MaskedTextBox³¹.

Este control es de real importancia ya que necesitamos que se validen los datos de entrada como cédulas o teléfonos.

En la comunidad Microsoft, ya se había encontrado este error, por tanto se descargó una solución realizada en Visual Basic que hereda del control TextBox, aunque un control de este tipo debería heredar de BaseTextTBox.

Sin embargo, esta solución presentaba algunos problemas, como el de no controlar validación de espacios o guiones bajos (_); por tanto, muchas veces los valores recopilados no correspondían a los que el usuario ingresaba.

Por esta razón, los MaskedTextBox se encuentran además validados con la ayuda de cada clase IDataErrorInfo + DataBindings.

VER ANEXO 3

³¹ FUENTE: Pro WPF with VB 2008 Windows Presentation Foundation with NET 3. 5

4.2.3.3.5.2. *DatePicker.*

Este control es muy útil en el momento de requerir validación de fechas, permitiendo realizar cambios dependiendo de cada necesidad.

A pesar de la falta de este control, WPF presenta una solución parecida llamada **DatePicker** que, como podemos deducir de su nombre, solamente nos brinda la validación para fechas.

Ya que necesitamos obligatoriamente la validación de horas para campos como Hora_Cabina en la Gestión de Instalación, se decidió usar la unión de un **DatePicker** con un **MaskedTextBox** creado.

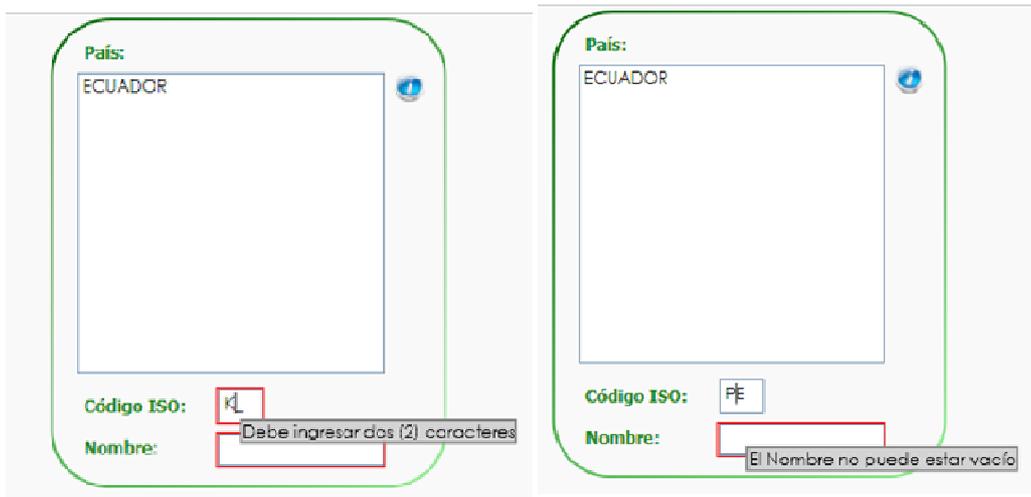


4.2.3.3.6. *Validaciones en WPF.*

POINTSOF se ha desarrollado de tal manera que el usuario se sienta satisfecho con el nuevo programa, promoviendo que se marquen los errores dinámicamente según se vayan ingresando los datos.

El tipo de validación que se usará es: **IDataErrorInfo** y **DataBindings** donde:

- **IDataErrorInfo:** Es una interfaz que proporciona la funcionalidad para facilitar información de error personalizada a la que puede enlazar una interfaz de usuario.
- **DataBindings:** Es una propiedad que provee una simple y consistente manera para que los controles presentes interactúen con los datos.



Con la ayuda de ambos factores, logramos que los controles puedan cambiar físicamente para que el usuario interprete estos cambios como posibles errores.

Para entender este modo de validación, se continuará aplicando el ejemplo de la clase País. Para esto, lo primero es que la clase implemente la interfaz IDataErrorInfo, la cual nos obliga a implementar dos métodos:

```
public class Pais : IDataErrorInfo
{
    #region Miembros de IDataErrorInfo
        string IDataErrorInfo.Error
        {
            get { return null; }
        }
        string IDataErrorInfo.this[string columnName]
        {
            get
            {
                string mensajeError = null;

                switch (columnName)
```

Por defecto error "null".

Columna **codigoIso**:
Si el String, es vacío, retorna "null".
Caso contrario, si su longitud es de dos dígitos y diferente de vacío, retorna el mensaje de error.

```
        case "codigoIso":
            if (String.IsNullOrEmpty(codigoIso))
            {

            }else if (codigoIso.Replace("_", "").Length != 2
                && codigoIso.Replace("_", "").Length != 0)
            {
                mensajeError = "Debe ingresar dos (2) caracteres";
            }

            break;
```

La clase País implementa la interfaz **IDataErrorInfo**, para lo cual debemos importar:

```
using System.ComponentModel;
```

Ésta proporciona clases que se utilizan para implementar el comportamiento en tiempo de ejecución y tiempo de diseño de los componentes y controles.

Método para controlar cada columna y devolver un "String" como un error.

```

        case "nombre" :
            if (String.IsNullOrEmpty(nombre))
            {
                mensajeError = "El Nombre no puede estar
                               vacío";
            }
            break;
        }
        return mensajeError;
    }
}
#endregion }

```

Columna **nombre**:
Si el campo es vacío, retorna el mensaje de error.

NOTA: Recalamos que esto hace que el programa sea modular, ya que estos errores, al estar incorporados en cada Clase, da la posibilidad de no solamente usar WPF sino, también, otras herramientas como Web o Windows Forms.

Es importante destacar que, ya que todos los controles TextBox serán validados, definimos en la aplicación global un estilo que todos estos controles usarán. En nuestro caso hemos decidido que cada vez que se produzca un error estos se pinten de color rojo (por defecto) y muestren en su ToolTip el error asociado:

En Application.xaml:

```

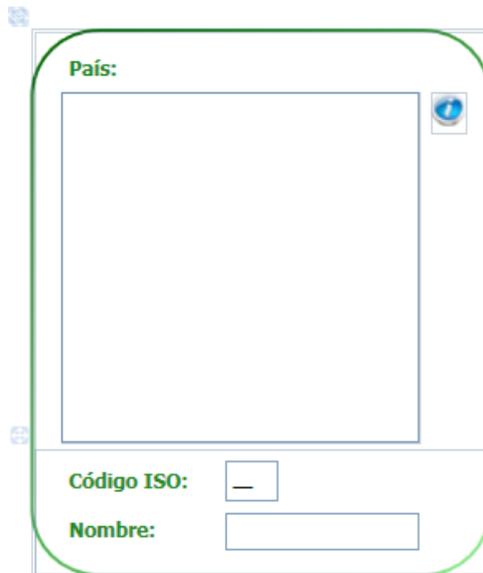
<Application x:Class="Application"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:lib="clr-namespace:CustomControls;assembly=CustomControls"
    StartupUri="Login.xaml">
    <Application.Resources>
        <Style x:Key="txtError" TargetType="{x:Type TextBox}">
            <Style.Triggers>
                <Trigger Property="Validation.HasError" Value="true">
                    <Setter Property="ToolTip"
                        Value="{Binding RelativeSource=
                            {x:Static RelativeSource.Self},
                            Path=(Validation.Errors)[0].ErrorContent}"/>
                </Trigger>
            </Style.Triggers>
        </Style>
    </Application.Resources>
</Application>

```

Modifica el estilo de un control **TextBox** siempre que exista error.
Lanza un ToolTip con el mensaje de error de la clase asociada (**IDataErrorInfo**).

Para que estos errores puedan ser recogidos, y vistos por el Usuario en la interfaz gráfica, es necesario cambiar la propiedad DataBinding de ese control:

En `uscPaís.xaml`:



```
<UserControl x:Class="uscPaís"
```

```
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" Width="278"
  Height="336"
  xmlns:lib="clr-namespace:CustomControls;assembly=CustomControls"
  xmlns:lbd="clr-namespace:LibBDPoint;assembly=LibBDPoint" >
```

```
<UserControl.Resources>
  <lbd:Pais x:Key="Pais"/>
</UserControl.Resources>
```

Referencia a País

Añadimiento de la definición de Clases **LibBDPoint**.

```
<Border Margin="0" Name="Border1" CornerRadius="40,40,40,40"
  BorderThickness="2" Width="278">
```

```
<Border.BorderBrush>
  <LinearGradientBrush StartPoint="0,0" EndPoint="1,1" >
    <GradientStop Color="DarkGreen" Offset="0" />
    <GradientStop Color="LightGreen" Offset="1.0" />
  </LinearGradientBrush>
</Border.BorderBrush>
```

Definición Visual del Control **uscPaís**.

```
<Grid>
```

```
<!--Definicion de los elementos del Grid-->
```

```
<!--Únicamente se especificará el código referente a los TextBox-->
```

```

<TextBox Style="{StaticResource txtError}"
  FontFamily="Century Gothic" FontSize="12" Height="23"
  Margin="115,0,42,14" MaxLength="45" Name="txtNombre"
  VerticalAlignment="Bottom" TabIndex="2">

  <TextBox.Text>
    <Binding Path="nombre" Source="{StaticResource Pais}"
      ValidatesOnDataErrors="True"
      UpdateSourceTrigger="PropertyChanged">
      <Binding.ValidationRules>
        <ExceptionValidationRule/>
      </Binding.ValidationRules>
    </Binding>
  </TextBox.Text>
</TextBox>

```

Definición de **DataBinding**:

Nombre del recurso País agregado.

Se Actualiza cada vez que la propiedad **Text** cambie.

Manda el error con valor True y llama a los métodos correspondientes de **IDataErrorInfo** de País.

MaskedTextBox codigolso:

Estilo igual que el de los TextBox.

DataBinding: codigolso del Recurso País.

Se actualiza cada vez que la propiedad cambie.

Manda el error con valor True y llama a los métodos correspondientes de **IDataErrorInfo** de País.

```

<lib:MaskedTextBox Style="{StaticResource txtError}"
  FontFamily="Century Gothic"
  FontSize="12" Foreground="Black" Mask="&gt;LL"
  x:Name="txtCodigoIso" Margin="115,0,127,44.148"
  Height="25.005" VerticalAlignment="Bottom"
  MaxLength="1" TabIndex="1">

  <lib:MaskedTextBox.Text>
    <Binding Path="codigoIso"
      Source="{StaticResource Pais}"
      ValidatesOnDataErrors="True"
      UpdateSourceTrigger="PropertyChanged">
      <Binding.ValidationRules>
        <ExceptionValidationRule/>
      </Binding.ValidationRules>
    </Binding>
  </lib:MaskedTextBox.Text>
</lib:MaskedTextBox>

</Grid>

</Border>

</UserControl>

```

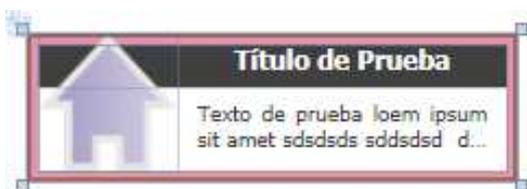
Si nos fijamos en el Binding, la propiedad UpdateSourceTrigger tiene como valor PropertyChanged. UpdateSourceTrigger definirá de que manera los datos serán enlazados al objeto origen:

Las opciones de UpdateSourceTrigger son:

- “Cuando el elemento pierde el foco” (LostFocus)
- “Cuando la propiedad cambia” (PropertyChanged)
- “Especificado vía código” (Explicit)

4.2.3.3.7. Respuestas de Éxito ó Error.

El usuario luego de realizar una de las operaciones posibles en la pantalla seleccionada y el sistema, siempre tendrá una respuesta (éxito ó error) para la operación seleccionada por el usuario.



En WPF no existen algunas de las herramientas que encontramos al trabajar con Windows Forms, pero podemos crear dichas herramientas a través de un

UserControl, para crear un **PopUp**³² enriquecido.

Función que crea con sus valores respectivos a un userControl llamado “**uscAviso**” para que sea contenido en un PopUp.

```
Private Sub mostrarPopGenerico(ByVal titulo As String,
                               ByVal mensaje As String,
                               ByVal tamanoFuente As Double,
                               ByVal imagenPopUp As BitmapImage,
                               ByVal imagenIcono As BitmapImage,
                               ByVal anchoBorde As Double,
                               ByVal colorBorde As Color)
```

³² Un PopUp es una ventana emergente, no solicitada, para dar información o anuncios publicitarios. Como las ventanas de un cliente de mensajería cuando alguien se conecta.

```

Dim popChild As New uscAviso
popChild.tbkTitulo.Text = titulo
popChild.tbkAviso.Text = mensaje
popChild.tbkAviso.FontSize = tamanoFuente
popChild.imgAviso.Source = imagenPopUp
popChild.brdAviso.Width = anchoBorde
popChild.brdAviso.BorderBrush = New SolidColorBrush(colorBorde)

popAviso.Child = popChild
popAviso.IsOpen = True

imgOperacion.Source = imagenIcono
imgOperacion.ToolTip = mensaje

```

End Sub

```

Private Sub mostrarPopExito(ByVal titulo As String,
                          ByVal mensaje As String)

    mostrarPopGenerico(titulo, mensaje, 12, imgPopOk, imgIconoOk,
225, Colors.DarkSeaGreen)

```

End Sub

Respuesta Exitosa para un evento llamando al proceso mostrarPopGenerico

```

Private Sub mostrarPopError(ByVal titulo As String,
                           ByVal mensaje As String)

    mostrarPopGenerico(titulo, mensaje, 10, imgPopError,
imgIconoError, 275, Colors.PaleVioletRed)

```

End Sub

Respuesta Fallida para un evento llamando al proceso mostrarPopGenerico

```

Public Sub mostrarPopInfo(ByVal titulo As String,
                         ByVal mensaje As String,
                         ByVal tamanoFuente As Double,
                         ByVal anchoBorde As Double)

    mostrarPopGenerico(titulo, mensaje, tamanoFuente, imgPopInfo,
imgIconoInfo, anchoBorde, Colors.CornflowerBlue)

```

End Sub

Información al Usuario usando el proceso mostrarPopGenerico.

Entonces veremos en la parte inferior izquierda del programa, las diferentes respuestas que puedan ocurrir. Por ejemplo se agrega el País Perú:

The screenshot shows a form titled 'País' with a list box containing 'ECUADOR' and 'PERÚ'. Below the list box are three input fields: 'Código ISO:' with a dropdown menu, and 'Nombre:' with a text box. To the right, a notification box displays a green checkmark and the text 'País Insertado correctamente' and 'Bienvenido Insertando País'.

Intentamos ingresar nuevamente ingresar País, por tanto muestra una respuesta de Error.

The screenshot shows the same 'País' form. The 'Código ISO:' field now contains 'PE' and the 'Nombre:' field contains 'PERÚ'. To the right, a notification box displays a red exclamation mark and the text 'País Error en la base de datos. Duplicata entry PERÚ for key 2' and 'Bienvenido Insertando País'.

Para visualizar la última respuesta de Información, se toma el ejemplo de Buscar Persona (18 registros encontrados):

The screenshot shows a search interface for 'Persona'. On the left, a list of search results is displayed, including names and identification numbers. On the right, a form titled 'Personal' contains fields for 'Perfil:', 'Código:', 'Cédula:', 'Nombre:', 'Primer Apellido:', 'Segundo Apellido:', 'Dirección:', 'Dirección 2:', 'Teléfono:', and 'Celular:'. A notification box on the right displays a blue information icon and the text 'Personal 18 registros encontrados' and 'Bienvenido Personal'.

4.2.4. IMPLANTACIÓN DE POINTSOFT.

En este apartado, todos los componentes antes diseñados, características y requisitos deben ser implementados, integrados y probados en su totalidad. Como resultado tendremos el programa POINTSOFT completo.

4.2.4.1. Servidor SERVER (Base de Datos).

Para esta aplicación, como primera instancia, el motor de base de datos (MySQL) debe ser instalado en el servidor escogido.

En la instalación, asignamos un usuario con su respectiva contraseña, con la cual nos conectaremos a la base de datos desde los terminales. Siguiendo, procedemos a “restaurar”³³ la base de datos **pointdb** previamente diseñada y probada en fase de desarrollo.

Si en el proceso de instalación no se produjeron errores y, a su vez, en la restauración de la base de datos, **pointdb** se encontrará listo para ser utilizado.

El SGBD instalado podrá ser accedido desde cualquiera de los terminales ya que se ubica en un punto de la red con permisos de acceso, siempre que las computadoras se encuentren dentro del dominio.

Todos los terminales de TELCOLSAM S.A. están dentro del dominio, ya que se utiliza autenticación centralizada. Esto quiere decir que aún cuando se tenga el software en un nuevo terminal, si éste no es parte del dominio, no importa si está conectado a la red simplemente no podrá acceder al SGBD. La autenticación centralizada brinda la seguridad suficiente como para obviar el proceso de encriptación, que únicamente ralentizaría el proceso innecesariamente.

³³ Pasar la información de respaldo a un servidor.

4.2.4.2. Terminales

En cada una de las computadoras donde vaya a ser ocupada la aplicación, debemos añadir los siguientes archivos:

ARCHIVO	DESCRIPCIÓN
 Pointsoft.exe PointSoft SAM	POINTSOFT.exe: Archivo ejecutable, correspondiente a GUI para el usuario.
 LibUtil.dll 1.0.0.0 LibUtil	LibUtil.dll: Librería con validaciones necesarias para datos y otras utilidades.
 LibBDPoint.dll 1.0.0.0 LibBDPoint	LibBDPoint.dll: Librería con las clases del POINTSOFT necesarias para su correcto funcionamiento.
 LibDALPoint.dll 1.0.0.0 LibDALPoint	LibDALPoint.dll: Librería con las clases necesarias para el intercambio de datos entre las Reglas del Negocio y la Base de Datos.
 mysql.data.dll 6.0.3.0 MySql.Data.dll	mysql.data.dll: Librería para la correcta conexión con la Base de datos en MySql.
 CustomControls.dll 1.0.3663.36911 CustomControls	CustomControls.dll: Librería para la inclusión en el proyecto de Controles Importantes.
 RibbonControlsLibrary.dll 3.5.31016.1 RibbonControlsLibrary	RibbonControlsLibrary.dll: Librería para la inclusión en el proyecto de Controles Ribbon Importantes.
 WPFToolkit.dll 3.5.31016.1 WPF Toolkit	WPFToolKit.dll: Librería para la inclusión en el proyecto de Controles Importantes.
 app.config	app.config: Archivo de Configuración, con información del Servidor de la base de datos.

Tabla 4.80: Archivos para la Instalación en los Terminales.

Es recomendable copiar estos archivos en un directorio con los permisos suficientes de acuerdo a la cuenta de usuario de Windows.

Ya que el programa será utilizado diariamente, la manera más sencilla de utilizarlo será enviar un acceso directo hacia el escritorio del archivo **POINTSOFT.exe**.

4.2.4.3. Conexión TCP/IP.

Debemos recordar que TELCOLSAM S.A. trabaja bajo una arquitectura TCP/IP y además bajo una arquitectura Cliente-Servidor.

Para que los terminales puedan acceder al servidor donde se deposita la base de datos (**pointdb**), es necesario un archivo de configuración llamado **app.config** en cada uno de los terminales, tal como detalla la **Figura 4.46**:

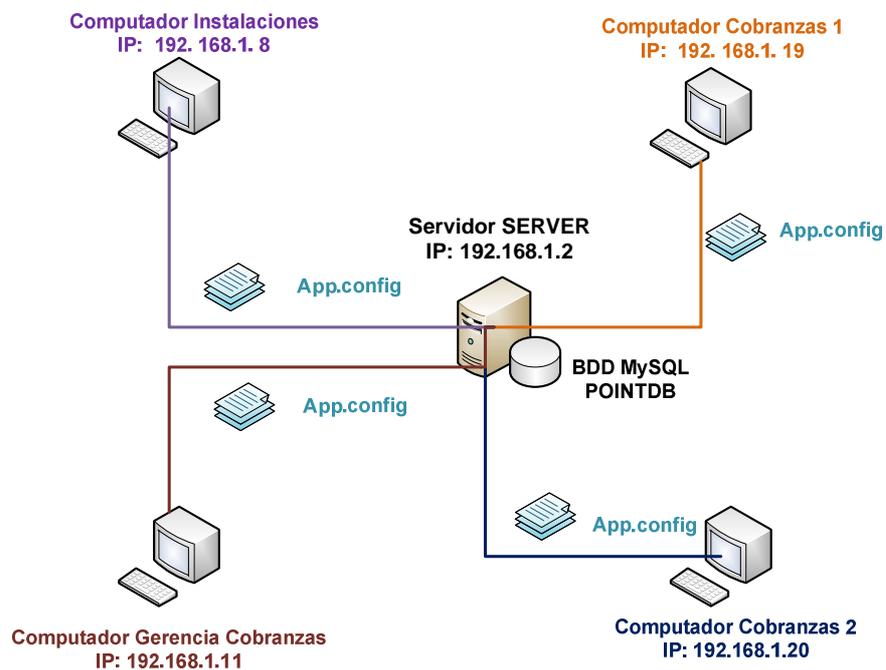


Figura 4.46: Diagrama de Conexión Actual.

Este archivo se encuentra especificado en el lenguaje XML, detallando valores importantes:

```

<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <configSections>
  </configSections>
  <connectionStrings>
    <add name="PointWPF.Settings.MySqlRoot"
      connectionString="server=192.168.1.2;
        user id=userwpf;password=tel15619;persist
        security info=True;database=pointdb" />
  </connectionStrings>
  <system.diagnostics>
  ...

```

Servidor IP: 192.168.1.2

Id_Base: userwpf **password_Base:** tel15619

Nombre_Base: pointdb **security:** Habilitado

4.2.4.4. Guía de Instalación POINTSOFT para el Usuario.

Si el caso a merita, podría ser necesario la instalación de la aplicación en otro terminal. El administrador de la red o técnico encargado, tendrá simplemente que seguir los pasos de la implementación en los terminales, detallados en el apartado anterior.

4.2.5. PRUEBAS Y RESULTADOS.

4.2.5.1. Realización de Pruebas diarias.

Luego de haber concluido la implantación de la aplicación, esperamos que los usuarios empiecen con la interacción con el sistema.

Las pruebas se realizan en el periodo de una semana, durante el horario normal de trabajo: del 9 de noviembre de 2009 hasta el 13 de noviembre de 2009, en el horario de 9:00 a 17:30.

Las pruebas se realizaron siguiendo el trabajo diario de cada uno de los usuarios. A partir de este proceso, se esperó reparar posibles errores que se puedan producir y, a su vez, mejorar en base a sugerencias brindadas.

4.2.5.2. Resultados de las Pruebas.

GESTIÓN INSTALACIONES		
PRUEBA	RESULTADOS	OBSERVACIONES
Creación de Perfiles	El sistema guarda, modifica, inserta y elimina sin presentar inconvenientes.	Recordamos que existe un perfil de SuperUsuario , el cual no puede ser borrado o modificado. SuperUsuario tiene todos los accesos al Sistema sin restricción. SuperUsuario tiene una contraseña solamente revelada a la Gerente General.
Creación de Usuarios/Empleados	Se produjo un error en el sistema, en modificar la contraseña de un empleado	El error fue reprogramado y solucionado posteriormente.
Creación de Propietario/Cliente	El sistema guarda, modifica, inserta y elimina sin presentar inconvenientes.	
Creación País	El sistema guarda, modifica, inserta y elimina sin presentar inconvenientes.	Aunque TELCOLSAM S.A. solamente se extiende en Ecuador, sus tablas de datos presentan este campo.
Creación Provincia	El sistema guarda, modifica, inserta y elimina sin presentar inconvenientes.	
Creación Ciudad	El sistema guarda, modifica, inserta y elimina sin presentar inconvenientes.	
Creación Sector	El sistema guarda, modifica, inserta y elimina sin presentar inconvenientes.	
Creación Barrio	El sistema guarda, modifica, inserta y elimina sin presentar inconvenientes.	

Continúa en la página 220.

Creación de PDVs	Presenta una falla, al no realizar una correcta validación en los datos de ingreso.	La falla fue solucionada de manera inmediata.
Instalación a un PDV	El sistema guarda, modifica, inserta y elimina sin presentar inconvenientes.	Recordar que el estado del PDV es "PREPARANDO INSTALACIÓN". Y luego de finalizada llega a "ACTIVO"
Cambios de Estado a un PDV	El sistema guarda, modifica, inserta y elimina sin presentar inconvenientes.	Ya que los empleados no trabajaban con este sistema de cambios de estados a un PDV, en un inicio presentaron confusión en el proceso. Sin embargo, en el transcurso de las pruebas el trabajo se torno muy eficiente.

Tabla 4.81: Resultado de Pruebas en el Departamento de Instalaciones.

GESTIÓN COBRANZAS		
PRUEBA	RESULTADOS	OBSERVACIONES
Crear Factor Valor	El sistema guarda, modifica, inserta y elimina sin presentar inconvenientes. Aunque el sistema funciona perfectamente. El campo factor muestra un valor incorrecto	La falla fue solucionada de manera inmediata.
Crear Mayorista	El sistema guarda, modifica, inserta y elimina sin presentar inconvenientes.	
Crear Gestión de Cobranzas	El sistema guarda, modifica, inserta y elimina sin presentar inconvenientes.	La empresa sugiere un campo más: al iniciar que muestre el saldo total de cada PDV. Información del PDV es añadida tanto a éste como al Reporte de Recaudación.

Continúa en la página 221.

Asignar Factor Valor a PDV	El sistema guarda, modifica, inserta y elimina sin presentar inconvenientes.	
Crear Periodo/Frecuencia	El sistema guarda, modifica, inserta y elimina sin presentar inconvenientes.	
Crear Ruta	El sistema guarda, modifica, inserta y elimina sin presentar inconvenientes.	
Asignar Ruta – Recaudador al PDV	El sistema guarda, modifica, inserta y elimina sin presentar inconvenientes.	
Ingresar Reporte de Recaudación / Ticket	<p>Al iniciar el formulario se cargaban todos los PDV en una lista; sin embargo, al cargar 500 PDVs el proceso empieza a tardar unos segundos.</p> <p>De todas maneras, se cumple con un tiempo inferior a 5 segundos para los 500 registros.</p> <p>La Gerente General pide desbloquear todos los campos que en un inicio se encontraban bloqueados por motivos de pruebas.</p>	<p>Los problemas son solucionados.</p> <p>El usuario debe buscar por número de PDV o nombre para evitar cargar todos los PDVs.</p> <p>Este es el formulario con más validaciones, ya que pueden existir muchas inconsistencias por fallas humanas. Sin embargo, los empleados presentaban inconvenientes al no acostumbrarse a esta inserción de datos.</p> <p>Por otro lado, acordamos que el beneficio que daba las validaciones sobresale a la falta de práctica de los empleados.</p>

Tabla 4.82: Resultado de Pruebas en el Departamento de Cobranzas.

GESTIÓN GERENCIA GENERAL		
PRUEBA	RESULTADOS	OBSERVACIONES
Reportes	<p>El sistema guarda, modifica, inserta y elimina sin presentar inconvenientes.</p> <p>Aunque el Sistema de Reportes funciona bien, toma demasiado tiempo al realizar la migración a Excel.</p>	<p>El problema de tiempo fue solucionado, cambiando algunos procedimientos almacenados en la base de datos.</p>

Tabla 4.83: Resultado de Pruebas en la Gerencia General.

4.2.5.3. Evaluación Final.

Concluida la etapa de pruebas, luego de obtener los resultados requeridos y pedidos por TELCOLSAM S.A., podemos definir la siguiente evaluación.

PARÁMETRO	EVALUACIÓN
Herramienta	<p>El uso de la metodología conocida como UML permite una comprensión del funcionamiento interno del sistema. Por tanto, se obtiene un sistema modular y escalable.</p> <p>La utilización de MySQL permite tener una base robusta, segura, rápida, y estable.</p>
Documentación	<p>La aplicación consta con su respectiva documentación para un correcto uso y comprensión del mismo.</p>
Administración y mantenimiento	<p>La utilización de estándares de programación permite que el mantenimiento y administración del sistema sea sencillo.</p> <p>Además al usar herramientas conocidas y actuales se encuentra mucha ayuda por parte de la comunidad en Internet.</p>

Continúa en la página 223.

Seguridad	<p>La aplicación permite un control de acceso en la parte de administración dependiendo del perfil de cada usuario, de acuerdo al cual se aplica las restricciones necesarias para el ingreso de datos, garantizando así, la integridad de los datos.</p> <p>Además las contraseñas no se almacenan directamente, sino sólo su hash. Los datos no pasan por un proceso encriptación ya que la red es lo suficientemente segura, primero, por una autenticación centralizada y, segundo, por buenas políticas en la empresa.</p>
Desempeño	<p>El desempeño es la eficiencia desde el punto de vista de operadores y administradores del sistema.</p> <p>Los directivos de la empresa se encuentran muy satisfechos, ya que el programa ha cumplido sus expectativas:</p> <ul style="list-style-type: none"> • En tiempo: Toma de decisiones a tiempo. • Reportes: Información organizada real y actual. • Económico: Anteriormente la toma de decisiones atrasadas, tenía un coste económico en contra.
Utilización	<p>Los resultados de la fase de pruebas indican que el funcionamiento de la aplicación ha cumplido totalmente con el alcance del proyecto.</p>
Manejo	<p>El sistema fue desarrollado pensando en la comodidad del usuario, de tal manera que sea fácil de manejar.</p> <p>Los operadores se encuentran trabajando con la aplicación diariamente. Comentan que el trabajo que realizan, en comparación con el anterior, se ha tornado más sencillo y ágil, aunque aún se hallan en la fase de adaptación.</p>
Contribución	<p>La aplicación POINTSOFT contribuye a la empresa brindando un servicio: ágil, rápido y eficiente.</p>

Tabla 4.84: Evaluación por parámetros de POINTSOFT.

CAPITULO 5



5. CONCLUSIONES Y RECOMENDACIONES.

5.1. CONCLUSIONES.

- Windows Presentation Foundation, es una herramienta utilizada actualmente por grandes empresas a nivel mundial. WPF puede brindar una interfaz gráfica realmente amigable con el usuario, ayudando a que su experiencia con el programa sea satisfactoria.

POINTSOF ha sido realizado pensando en el usuario, por lo que le hemos dado una perspectiva muy similar a la de Windows Vista o Windows 7 en programas como la Suite Office 2007, Paint 6.1, o posteriores. Con esto se logró que el usuario, desde un inicio, no se sienta confundido y trabaje con facilidad como lo hace con los programas que usualmente trabaja (Office, Excel); y al final, con el uso del sistema realizado, logró agilizar sus procedimientos, ahorrando tiempo y dinero en la empresa.

- La ventaja más relevante en POINTSOFT es la modularidad, por dos razones: la primera es que redujimos el tiempo de trabajo al poder programar dos módulos independientemente a la vez, Interfaz Gráfica y Reglas del Negocio, y un tercer sistema independiente, la Base de Datos; segundo, la gran ventaja de poder seguir añadiendo módulos cuantos estos sean necesarios. Actualmente, se tiene propuestas de TELCOLSAM S.A. para incrementar un módulo al Sistema POINTSOFT para Gestión del Departamento Técnico.
- Otra gran ventaja que POINTSOFT brinda a TELCOLSAM S.A. es la posibilidad de que varios empleados trabajen con el programa simultáneamente (conurrencia), dejando atrás los días de trabajo con un único archivo de Excel compartido, donde solamente una persona podía usar los datos al mismo tiempo.

- El mayor problema de TELCOLSAM S.A. consistía en la toma de decisiones tardías ya que esto hacía que la empresa pierda dinero. POINTSOFT fue diseñado principalmente para brindar solución a este inconveniente, sin aumentar ni dificultar el trabajo de los empleados. Por una parte, los usuarios pueden seguir trabajando en Excel para los reportes y órdenes de trabajo, como ya lo tenían acostumbrado; y por otro lado, los reportes serán actuales y reales para una toma de decisiones eficientes y a tiempo.
- POINTSOFT fue diseñado en una arquitectura en Tres Capas, donde dos de ellas (Reglas del Negocio e Interfaz gráfica) fueron realizadas con Herramientas de Microsoft, siendo éstas herramientas privativas. La tercera capa (Base de Datos), por el contrario, fue diseñada con ayuda de MySQL 5.1, o sea una solución pública.
Por lo citado anteriormente, podemos decir concluyentemente que un proyecto puede ser resuelto sin ningún inconveniente de conexión y comunicación uniendo diferentes soluciones, con diferentes licencias, siempre que éstas se adapten a los requerimientos de la empresa.
- La realización del proyecto POINTSOFT ha alcanzado a cabalidad los objetivos trazados en un comienzo y, lo más importante, los empleados y directivos de TELCOLSAM S.A. se sienten satisfechos con el trabajo entregado.
- Aunque TELCOLSAM S.A. anteriormente no brindaba ninguna clase de seguridad, POINTSOFT pudo incluir seguridades básicas como: Perfiles de Usuario y Encriptación de Contraseñas en la Base de Datos mediante SHA1. Esto niveles de seguridad se consideran suficientes puesto que la Intranet de TELCOLSAM S.A. brinda seguridades a nivel de autenticación centralizada con Kerberos. La añadidura de encriptación sólo ralentizaría el proceso para el trabajo dentro de la Intranet.

- Para que un programa de desarrollo de Software pueda cumplir el mayor objetivo que existe en cualquier proyecto, o sea **complacer al cliente**, concluimos que la etapa más importante es la de “Obtención de Requerimientos”, puesto que si no se entiende lo que el Cliente necesita y requiere, el programa no será usado por los usuarios y solamente quedará en papel su realización.

El uso de estándares para este paso facilita en gran parte la clasificación y presentación de los datos obtenidos, lo que en gran medida reducen el tiempo de desarrollo del sistema.

- Aunque los equipos (PCs) actuales de TELCOLSAM S.A., no son del todo potentes, recordamos que POINTSOFT ha sido instalado sin problemas sobre estos equipos, cumpliendo muy bien su objetivo. Sin embargo, la empresa tiene pensado mejorar su equipamiento en un futuro cercano. Por lo tanto, concluimos que cuando se realice dicho cambio, el programa realizado trabajará aún más con mayor velocidad y eficiencia puesto que a medida que aumentan las prestaciones de hardware, WPF se adapta para mejorar su rendimiento.
- Los estándares de XAML, no son difíciles de entender ni de emplear. Al igual que los demás lenguajes que Visual .NET nos proporciona, XAML es muy intuitivo. Se debe empezar especificando sus características generales y luego seguir buscando información para ir mejorando nuestra aplicación.
- La Arquitectura en Tres Capas satisfizo completamente el desarrollo del sistema. Aún cuando WPF puede trabajar tanto con 3-Capas como con MVC, esta segunda metodología incurría en programación excesiva para dar una solución completa a las necesidades de TELCOLSAM S.A., como es la programación del Controlador. WPF, en aplicaciones de escritorio, brinda soporte mediante código subadyacente para la interconexión con XAML, por lo que la solución presentada garantiza la comunicación mediante esta Arquitectura.

5.2. RECOMENDACIONES.

- El proceso de recaudación manual podría solventarse aprovechando nuevas tecnologías que permitan pagos remotos, como transferencias electrónicas, para lo cual se podrían utilizar las mismas librerías y únicamente cambiar la Capa de Presentación para crear una aplicación Web. WPF, utilizando XAML, soporta aplicaciones Web, por lo que el proceso de desarrollo sería similar al ya especificado en este proyecto de titulación.
- Para mejorar la seguridad de la aplicación para trabajar en ambientes más inseguros, como Internet, se recomienda la encriptación de los datos en los DAL, tanto para transmisión como desde la Base de Datos hacia la Capa de Negocios. Esta encriptación debe realizarse dada la sensibilidad de los datos a transmitirse. Incluso, si se implementara un sistema de transferencia electrónica, el manejo de dinero hace aún más crítica la implementación del concepto de Transferencia Segura.
- Recordemos que POINTSOFT solamente es una pequeña parte de lo que se podría realizar con la herramienta Windows Presentation Foundation. Para mejorar este proyecto se pueden incluir ayudas auditivas y navegación mediante herramientas en 3D. De todas maneras, hay que considerar que WPF es más productivo cuando se tienen buenas prestaciones de hardware, por lo que el incremento de estas características en hardware de bajas prestaciones no mejoraría la experiencia del usuario sino causaría el efecto contrario.
- Queremos aclarar que las herramientas escogidas en este proyecto no son las mejores de todas. Estas herramientas son las mejores para la empresa TELCOLSAM S.A. Recomendamos hacer un análisis mucho más complejo de cada herramienta para garantizar que las escogidas funcionarán tal como lo requiere una empresa.

- Es importante entender que ninguna red ni programa nunca asegurará 100% de seguridad. Hasta ahora TELCOLSAM S.A. no ha tenido problemas de manipulación de datos, a pesar de trabajar con archivos planos. Por tanto, recomendamos siempre tener presente este problema, tratando de incorporar cada vez mejores seguridades, pero guiados por la sensibilidad de la información y no únicamente por el avance tecnológico.
- Ahora todos los datos se encuentran almacenados en una sola Base de Datos, en un determinado servidor. Recomendamos siempre estar pendiente en cuidar estos datos, sacando respaldos cada cierto tiempo y almacenarlos en otro dispositivo, si es posible. Para mejorar este funcionamiento, se pueden implementar Servidores de Datos de Backup, donde se respalde automáticamente la información, pudiendo hacerse con MySQL para evitar la migración de datos.
- Si se requiere instalar el Software en otro computador, para un posible nuevo empleado, solamente será necesario realizar una copia local de la carpeta **PointWPF** ubicada en una carpeta de la nueva computadora. Por último, se requerirá enviar al escritorio un acceso directo del ejecutable **PointWPF** que se encuentra ubicado dentro de dicha carpeta.
Esto debido a que la programación es independiente de la versión del SO por trabajar con un Framework. Sin embargo, hay que tener presente que para que el programa funcione correctamente, el terminal debe cumplir con los requerimientos mínimos, entre los que se encuentra .NET Framework 3.1 SP1.

Bibliografía



6. BIBLIOGRAFÍA.

6.1. LIBROS.

Autor: Ivar Jacobson, Grady Booch, Jamer Rumbaugh

Nombre: El Proceso Unificado de Desarrollo de Software

Editorial: Addison Wesley, España

Año: 2000

Autor: Roger Pressman

Nombre: Ingeniería de Software (Un enfoque práctico)

Editorial: McGraw Hill, Madrid España

Año: 2002

Autor: Craig Larman

Nombre: UML y Patrones – Una introducción al análisis y diseño orientado a objetos y al proceso unificado

Editorial: Pearson Education, España

Año: 2003

Autor: Matthew MacDonald

Nombre: Pro WPF with VB 2008 Windows Presentation Foundation with NET 3. 5

Año: 2008

Sitio de Descarga: <http://www.free-ebooks-download.org/free-ebook/dotnet/VB.NET/pro-wpf-with-vb-2008-windows-presentation-foundation-with--net-3-5.php>

6.2. PÁGINAS WEB.

Microsoft SQL Server vs. MySQL a comparison

<http://www.isql.org/2007/02/mysql-vs-sybase-ms-sql-server.html>

Características de SQL Server 2005

<http://www.microsoft.com/spain/sql/productinfo/features/compare-features.msp>

Información de Oracle 10g Express Edition

http://www.oracle.com/lang/es/database/Express_Edition.html

MySQL vs. SQLServer

http://www.tometasoftware.com/mysql_vs_sqlserver.asp

Modelo Visual con UML

<http://www.utm.mx/~caff/doc/ModeladoVisualconUML.pdf>

Un ejemplo con UML

<http://msdn.microsoft.com/es-es/library/bb972214.aspx>

Historia de UML

http://eisc.univalle.edu.co/materias/Material_Desarrollo_Software/UML_Historia_Intro.pdf

WPF

<http://windowsclient.net/WPF>