

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

**DISEÑO E IMPLEMENTACIÓN DE UNA APLICACIÓN CTI CON
GEORREFERENCIACIÓN DE NÚMEROS TELEFÓNICOS FIJOS
UTILIZANDO EL TAPI DE LA CENTRAL TELEFÓNICA IP OFFICE DE
AVAYA**

TOMO I

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

CHACÓN HERRERA DIEGO JEFFERSON
contact@chacondiego.com

DIRECTOR: ING. HIDALGO LASCANO PABLO WILLIAM
pablo.hidalgo@epn.edu.ec

Quito, Junio 2015

DECLARACIÓN

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

A través de la presente declaración cedo mis 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.

Diego Chacón Herrera

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Diego Jefferson Chacón Herrera, bajo mi supervisión.

Ing. Pablo Hidalgo
DIRECTOR DEL PROYECTO

AGRADECIMIENTOS

En primer lugar, me es casi imposible expresar con palabras la inmensurable gratitud que siento hacia mis padres, Diego Chacón y Hisella Herrera, personas que me han demostrado repetidamente que el valor de un individuo está más allá de sus logros conseguidos y que entregar amor incondicionalmente es algo no sólo natural sino necesario para seguir adelante. Sé que he cometido tantos errores en tan poco tiempo que tengo la certeza de que sin ustedes ni su apoyo no habría culminado esta etapa. Gracias padres queridos por haberme brindado el privilegio de ser su hijo. Espero que algún día puedan sentirse tan orgullosos de mí como yo lo estoy de ustedes.

Seguidamente, no pueden faltar las merecidas palabras de agradecimiento a David González, Johanna Pinargote, Franklin Quiroz, Diego Fernández, y a todas las demás personas a quienes tuve el gusto de conocer en Lynxsource Cía. Ltda. Fueron ellos quienes no sólo me prestaron la idea que terminaría por convertirse en mi proyecto de titulación, sino también me facilitaron todos los recursos necesarios para que el plan saliera adelante. Gracias por haber sido tan buenos conmigo y haberme permitido ser su amigo.

Además, transmito mi sincera gratitud al Ing. Pablo Hidalgo, director de este proyecto, quien me ha enseñado que fuera del aula de clases los profesores pueden comprender y compartir las vicisitudes de sus alumnos. Gracias por su entendimiento y enorme paciencia (no, no exagero) y también lamento mucho las inconveniencias ocasionadas. Estoy tranquilo y seguro de que los estudiantes de redes no pueden estar en mejores manos.

Finalmente, gracias a mis hermanos, Omar y Karolina, quienes han sabido entender mis errores y animar mis esfuerzos. No puedo hacer nada más que desearles lo mejor. Los quiero mucho.

DEDICATORIA

A mis padres, sabios comprensivos y redentores de mi alma.

Diego Jefferson

CONTENIDO

TOMO I

DECLARACIÓN	I
CERTIFICACIÓN	II
AGRADECIMIENTOS	III
DEDICATORIA	IV
CONTENIDO	V
ÍNDICE DE FIGURAS	XIII
ILUSTRACIONES DE ANEXOS.....	XIX
ÍNDICE DE TABLAS	XXIII
RESUMEN.....	XXVI
PRESENTACIÓN.....	XXVII
CAPÍTULO I	
1. MARCO TEÓRICO	1
1.1. GENERALIDADES Y CONCEPTOS BÁSICOS DE TELEFONÍA.....	1
1.1.1. INTRODUCCIÓN	1
1.1.2. TELEFONÍA FIJA.....	1
1.1.3. RED DE TELEFONÍA PÚBLICA CONMUTADA (RTPC ó PSTN)	1
1.1.4. PROCESO DE UNA LLAMADA TELEFÓNICA	2
1.1.5. PBX	3
1.1.5.1. Componentes	4
1.1.5.2. Características.....	5
1.1.6. TELEFONÍA IP	6

1.1.6.1. Definición.....	6
1.1.6.2. Tecnologías de Transmisión de Voz sobre Redes de Datos.....	6
1.1.6.3. Conceptos Adicionales	7
1.1.6.4. Protocolos de Señalización.....	8
1.1.7. PLANES DE NUMERACIÓN	9
1.1.7.1. Plan Técnico Fundamental de Numeración (PTFN) del Ecuador	9
1.1.7.2. Acrónimos.....	10
1.1.7.3. Estructura del Número Telefónico Internacional del Ecuador.....	10
1.1.7.4. Asignación General del Indicativo Nacional de Destino (NDC)	11
1.1.7.5. Número Nacional Telefónico: Numeración Geográfica	12
1.1.7.6. Número de Abonado.....	12
1.1.7.7. Áreas y Códigos de Numeración Geográfica	12
1.1.7.8. Número Nacional Móvil: Numeración no Geográfica.....	13
1.1.7.9. Plan de Marcación	13
1.1.7.10. Series Numéricas	14
1.2. LA TECNOLOGÍA CTI	14
1.2.1. INTRODUCCIÓN.....	14
1.2.2. FUNCIONES Y CARACTERÍSTICAS DE LAS APLICACIONES CTI.....	15
1.2.3. ARQUITECTURAS DE INTEGRACIÓN CTI	16
1.2.3.1. <i>First-Party</i> (Acceso Directo o de Primera Parte)	16
1.2.3.2. <i>Third-Party</i> (Acceso Indirecto o de Tercera Parte).....	17
1.2.4. ESTÁNDARES CTI.....	18
1.2.4.1. Interfaz de Programación de Aplicaciones de Telefonía (TAPI)	19
1.2.4.2. Interfaz de Programación de Aplicaciones de Servicios de Telefonía (TSAPI)	23

1.2.4.3. Interfaz de Programación de Aplicaciones de Telefonía de Java (JTAPI)	23
1.2.4.4. Aplicaciones de Telecomunicaciones con Soporte de Computador (CSTA)	24
1.2.4.5. Interfaz de Programación de Aplicaciones de Telefonía de Linux	25
1.3. AVAYA IP OFFICE 500 V2	25
1.3.1. CARACTERÍSTICAS	26
1.3.2. ARQUITECTURA DE COMUNICACIONES IP OFFICE.....	28
1.3.2.1. Unidad de Control	29
1.3.2.2. Tarjetas Base	30
1.3.2.3. Tarjetas Troncales	32
1.3.2.4. Módulos Externos de Expansión.....	33
1.3.2.5. Teléfonos.....	34
1.3.3. <i>COMPUTER TELEPHONY INTEGRATION</i>	35
1.3.3.1. Herramientas e Interfaces CTI	35
1.3.3.2. Licencias CTI.....	38
1.4. <i>MICROSOFT FOUNDATION CLASSES</i>	38
1.4.1. DEFINICIÓN	38
1.4.2. MODELO DE PROGRAMACIÓN DE WINDOWS	39
1.4.3. ESTRUCTURA DE UNA APLICACIÓN MFC.....	41
1.5. JAVA EE	41
1.5.1. INTRODUCCIÓN	41
1.5.2. APLICACIONES DISTRIBUIDAS MULTINIVEL.....	43
1.5.2.1. Componentes Java EE	43
1.5.3. CONTENEDORES JAVA EE	45

1.5.3.1. Servicios de Contenedor.....	45
1.5.3.2. Tipos de Contenedores.....	45
1.5.4. APIS Y TECNOLOGÍAS JAVA EE 6 & 7.....	46
1.5.4.1. Tecnología <i>Enterprise JavaBeans</i> (EJB)	46
1.5.4.2. Tecnología <i>JavaServer Faces</i> (JSF).....	48
1.5.4.3. <i>Java Persistence API</i> (JPA).....	48
1.5.4.4. <i>Java Database Conectivity API</i> (JDBC)	49
1.5.4.5. <i>Java Naming and Directory Interface</i> (JNDI).....	49
1.5.5. SERVIDORES DE APLICACIONES	50
1.6. LENGUAJE UNIFICADO DE MODELADO	51
1.6.1. INTRODUCCIÓN	51
1.6.2. DIAGRAMAS DEL UML	51
CAPÍTULO II	
2. ANÁLISIS DE REQUERIMIENTOS Y DISEÑO	56
2.1. PLANTEAMIENTO DEL PROBLEMA	56
2.2. METODOLOGÍA DE DESARROLLO DE SOFTWARE	56
2.2.1. INTRODUCCIÓN.....	56
2.2.2. SELECCIÓN DE LA METODOLOGÍA.....	57
2.2.3. VISIÓN GENERAL DE PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE	59
2.2.3.1. Introducción y Características.....	59
2.2.3.2. Fases del Proceso Unificado	60
2.3. FASE DE INICIO (I)	62
2.3.1. INTRODUCCIÓN	62
2.3.2. MODELAMIENTO DE LOS PROCESOS DE NEGOCIO Y CASOS DE	

USO.....	63
2.3.2.1. Agente	63
2.3.2.2. Supervisor	68
2.3.2.3. Administrador	69
2.3.3. REQUISITOS (IR).....	70
2.3.3.1. Requisitos Técnicos-Operativos (IR-TO).....	71
2.3.3.2. Requisitos Funcionales (IR-RF).....	71
2.3.3.3. Requisitos No Funcionales (IR-NF).....	75
2.3.4. ANÁLISIS (IA).....	76
2.3.4.1. Categorización de los Requisitos del Sistema (IA-CA).....	77
2.3.4.2. Análisis de los Requisitos del Sistema (IA-AR).....	81
2.3.4.3. Análisis de la Arquitectura (IA-AA).....	91
2.4. FASE DE ELABORACIÓN (E)	97
2.4.1. REQUISITOS (ER)	98
2.4.1.1. Especificación de Casos de Uso de Requisitos (ER-ECU-R).....	98
2.4.1.2. Refinamiento de Casos de Uso de Funcionalidades (ER-RCU-F) .	108
2.4.2. ANÁLISIS Y DISEÑO (EAD).....	133
2.4.2.1. Comunicaciones (EAD-COM)	134
2.4.2.2. Inicio y Cierre de Sesión en el Sistema – Agentes (EAD-SES).....	141
2.4.2.3. Gestión de Eventos de las Líneas Telefónicas (EAD-EVN)	144
2.4.2.4. Despliegue de la Información de Contacto en una Llamada (EAD-INF).....	152
2.4.2.5. Detalle del Número Telefónico Participante en una Llamada Externa (EAD-NUM)	155
2.4.2.6. Servidor CTI (EAD-CTI).....	156

2.4.2.7. Servidor Principal (EAD-CORE).....	160
2.4.2.8. Cliente de Escritorio (EAD-DSK).....	168
2.4.2.9. Base de Datos (EAD-BDD).....	172

TOMO II

CONTENIDO	I
ÍNDICE DE FIGURAS	IX
ILUSTRACIONES DE ANEXOS.....	XV
ÍNDICE DE TABLAS	XIX
CAPÍTULO III	
3. DESARROLLO, DESPLIEGUE Y PRUEBAS.....	176
3.1. FASE DE CONSTRUCCIÓN (C).....	176
3.1.1. ANÁLISIS (CA)	177
3.1.1.1. Servidor CTI (CA-CTI)	177
3.1.1.2. Servidor Principal (CA-CORE)	178
3.1.1.3. Cliente de Escritorio (CA-DES).....	182
3.1.2. IMPLEMENTACIÓN (CM).....	187
3.1.2.1. Servidor CTI (CM-CTI).....	189
3.1.2.2. Biblioteca Común (CM-BC).....	193
3.1.2.3. Servidor Principal (CM-CORE)	195
3.1.2.4. Cliente de Escritorio (CM-DES)	225
3.1.2.5. Gestión de Eventos (CM-EVN).....	230
3.1.3. PRUEBAS DE LA FASE DE CONSTRUCCIÓN (CP)	238
3.2. FASE DE TRANSICIÓN (T)	239

3.2.1. DESPLIEGUE (TD).....	239
3.2.1.1. Central Telefónica <i>Avaya IP Office</i> 500 (TD-IPO)	239
3.2.1.2. Servidor CTI (TD-CTI).....	241
3.2.1.3. Servidor Principal (TD-CORE)	243
3.2.1.4. Agente de Escritorio (TD-DSK).....	245
3.2.2. PRUEBAS DE LA FASE DE TRANSICIÓN (TP).....	248
3.2.2.1. Escenario de Evaluación de la Solución (TP-EV)	248
3.2.2.2. Cumplimiento General de Requisitos Primarios y Secundarios (TP-CR)	252
3.2.2.3. Cumplimiento de Casos de Uso de Funcionalidades (TP-CU)	258
CAPÍTULO IV	
4. ANÁLISIS COMPARATIVO	261
4.1. COSTO DEL PROYECTO	261
4.1.1. ENFOQUE TEÓRICO: ESTIMACIÓN MEDIANTE EL MODELO CONSTRUCTIVO DE COSTOS	261
4.1.2. ENFOQUE PRÁCTICO: COSTOS FIJOS	264
4.2. OTRAS SOLUCIONES DE GESTIÓN DE <i>CALL CENTERS</i>	266
4.2.1. <i>IP OFFICE CONTACT CENTER</i>	267
4.2.2. <i>EVOLUTION CALL CENTER</i>	270
4.3. ANÁLISIS COMPARATIVO.....	271
4.3.1. TÉCNICO-FUNCIONAL	271
4.3.2. ECONÓMICO	274
CAPÍTULO V	
5. CONCLUSIONES Y RECOMENDACIONES.....	277
5.1. CONCLUSIONES	277

5.2. RECOMENDACIONES	280
REFERENCIAS BIBLIOGRÁFICAS	283
LIBROS, CURSOS, Y MANUALES	283
DIRECCIONES ELECTRÓNICAS	284
ANEXOS	286
ANEXO A: GLOSARIO DE TÉRMINOS Y DEFINICIONES	
ANEXO B: GUÍA DE INSTALACIÓN DE <i>CALL MANAGER</i> : SERVIDOR CTI	
ANEXO C: GUÍA DE DESPLIEGUE DE <i>CALL MANAGER</i> : SERVIDOR PRINCIPAL	
ANEXO D: GUÍA DE INSTALACIÓN DE <i>CALL MANAGER</i> : AGENTE <i>DESKTOP</i>	
ANEXO E: GUÍA DE CREACIÓN Y CARGA DE PLANES DE NUMERACIÓN	
ANEXO F: GUÍA DE INTERCONEXIÓN CON BASES DE DATOS EXTERNAS	
ANEXO G: PROFORMA DE <i>AVAYA IP OFFICE CONTACT CENTER</i>	

ÍNDICE DE FIGURAS

Figura 1.1 Estructura del Número Internacional del Ecuador	10
Figura 1.2 Estructura del Número Nacional (Significativo)	11
Figura 1.3 Composición del Número Nacional Telefónico	12
Figura 1.4 Composición del Número de Abonado	12
Figura 1.5 Ejemplo de Arquitectura <i>First-Party</i>	16
Figura 1.6 Ejemplo de la Arquitectura <i>Third-Party</i>	18
Figura 1.7 Arquitectura de Telefonía de Microsoft.....	20
Figura 1.8 Modelo de Programación de Telefonía de Microsoft	22
Figura 1.9 Entorno de comunicaciones <i>IP Office</i>	28
Figura 1.10 Vista Frontal y Posterior de la Unidad de Control IP500 V2	29
Figura 1.11 Colocación de una Tarjeta Base	31
Figura 1.12 Ubicación de una Tarjeta Troncal en una Tarjeta Base	32
Figura 1.13 Vista Frontal y Posterior del Módulo de Expansión IP500 Digital Station	33
Figura 1.14 Terminales Soportados por IP Office.....	34
Figura 1.15 Modelo de Programación de Windows dirigido por Eventos	39
Figura 1.16 Aplicaciones Multinivel	42
Figura 1.17 Ejemplo de Notación UML de una Clase	52
Figura 1.18 Ejemplo de Representación de un Caso de Uso	52
Figura 1.19 Ejemplo de Representación de Estados de una Llamada Telefónica.	53
Figura 1.20 Ejemplo de Representación de un Diagrama de Comunicaciones	53
Figura 1.21 Ejemplo de Representación de la Interacción entre un Cliente de Correo Electrónico y el Servidor	54

Figura 1.22 Representación de un Diagrama de Distribución.....	55
Figura 1.23 Diagrama de Actividades del Procesamiento de Eventos en Aplicaciones MFC Windows	55
Figura 2.1 Fases y Disciplinas del Proceso Unificado	61
Figura 2.2 Caso de Uso - Agente	64
Figura 2.3 Proceso de Negocio de Agentes para la Recepción de Llamadas.	66
Figura 2.4 Proceso de Negocio de Agentes para la Realización de Llamadas.	67
Figura 2.5 Caso de Uso – Supervisor	68
Figura 2.6 Caso de Uso – Administrador.....	70
Figura 2.7 Información del Plan Técnico Fundamental de Numeración – Resumen..	87
Figura 2.8 Detalle del Anexo 3 (código de área 3) del PTFN.....	88
Figura 2.9 Sistema “Multiplataforma” de Gestión de <i>Call Center</i>	90
Figura 2.10 Arquitectura Candidata.....	92
Figura 2.11 Servidor Principal y Componentes con los que Interactúa.....	96
Figura 2.12 Caso de Uso – Inicio y Cierre de Sesión	108
Figura 2.13 Logotipo de “ <i>Call Manager</i> ”	134
Figura 2.14 Diagrama de Actividades - Inicio de Sesión de Agentes.....	142
Figura 2.15 Diagrama de Actividades – Cierre de Sesión de Agentes	143
Figura 2.16 Gestión de Eventos de Líneas Telefónicas Generados por la Central Telefónica	145
Figura 2.17 Procesamiento de Eventos Generados por la Central Telefónica en el Servidor CTI	148
Figura 2.18 Gestión de Comandos de los Usuarios	149
Figura 2.19 Gestión de Eventos Generados por los Agentes en el Servidor CTI.....	150
Figura 2.20 Notificación de Llamadas y Búsqueda de Información de Contactos....	153

Figura 2.21 Diagrama de Clases del Servidor CTI	157
Figura 2.22 Procesamiento de mensajes en el servidor CTI	159
Figura 2.23 Ventana del módulo CTI de <i>Call Manager</i>	160
Figura 2.24 Principales Componentes de Negocio y Procesamiento de Mensajes .	162
Figura 2.25 Procesamiento de mensajes recibidos en el servidor principal.....	163
Figura 2.26 Respuesta a la solicitud de un cliente según el modelo JSF	164
Figura 2.27 Esquema de Construcción de Páginas Web de <i>Call Manager</i>	168
Figura 2.28 Procesamiento de Mensajes Recibidos en el Agente de Escritorio	169
Figura 2.29 Ventana principal del aplicativo de escritorio del agente	171
Figura 2.30 Diseño de Base de Datos.....	175
Figura 3.1 Servicio de Disponibilidad Geográfica de Telefonía Fija Provisto por la CNT.....	185
Figura 3.2 Carga de la Página Principal.....	186
Figura 3.3 Solicitud POST de Búsqueda de Coordenadas.....	186
Figura 3.4 Respuesta que Contiene la Localización Geográfica del Número Telefónico Fijo.....	186
Figura 3.5 Diagrama de Clases del Servidor CTI	188
Figura 3.6 Código de Arranque del TAPI y Comunicaciones en el Servidor CTI	190
Figura 3.7 Función de Inicialización del TAPI.....	191
Figura 3.8 Función de Retorno en <i>TapiHandler</i> para el Procesamiento de Eventos	191
Figura 3.9 Cuadro de Diálogo Principal del Servidor CTI	192
Figura 3.10 Diagrama de Clases de la Biblioteca <i>CallManagerCommons</i>	194
Figura 3.11 Gestión del Proyectos CMW1 en <i>NetBeans</i>	195
Figura 3.12 Diagrama de Clases del Componente de Negocio del Servidor Principal	197

Figura 3.13 Procedimiento de Reconexión Automática del Servidor Principal.....	198
Figura 3.14 Procesamiento de Llamadas y Búsqueda de Información de Contacto en el Servidor Principal.....	200
Figura 3.15 Utilización de la Biblioteca Apache <i>HttpComponents</i> para la Georreferenciación de Llamadas Telefónicas Utilizando el Servicio de Disponibilidad Geográfica de CNT.....	201
Figura 3.16 Búsqueda de Información en Bases de Datos Externas	203
Figura 3.17 Creación de Cadenas de Búsqueda para las BDD Soportadas	203
Figura 3.18 Diagrama de Clases del Componente Web del Servidor Principal	205
Figura 3.19 Página de Inicio de Sesión	206
Figura 3.20 Procedimiento de Inicio de Sesión	207
Figura 3.21 Página de Inicio del Administrador	208
Figura 3.22 Creación de un Nuevo Puesto de Trabajo	208
Figura 3.23 Página de Gestión de Puestos de Trabajo	209
Figura 3.24 Gestión de Usuarios.....	209
Figura 3.25 Creación de Usuarios.....	210
Figura 3.26 Gestión de Comunicaciones.....	210
Figura 3.27 Gestión de Planes de Numeración	211
Figura 3.28 Plan de Numeración de la Zona 2 Cargado en la BDD.....	211
Figura 3.29 Carga de un Nuevo Plan de Numeración en el Sistema	212
Figura 3.30 Página de Configuración de Variables del Sistema	212
Figura 3.31 Asistente de Integración con un Repositorio de Información Externo ...	213
Figura 3.32 Parámetros de Interconexión Correctamente Configurados	213
Figura 3.33 Página Principal de Supervisión	214
Figura 3.34 Monitoreo del Estado de los Agentes	215

Figura 3.35 Envío de Mensajes Desde el Supervisor Hacia los Agentes	216
Figura 3.36 Establecimiento de Parámetros de Búsqueda de la Sección de Estadísticas	216
Figura 3.37 Diagrama de Secuencias de Búsqueda de Registros de Llamadas.....	217
Figura 3.38 Estadística de Desempeño de los Agentes del <i>Call Center</i>	218
Figura 3.39 Registros de Llamadas Procesadas por el Sistema	219
Figura 3.40 Estadística de Número de Llamadas por Operadora	219
Figura 3.41 Gráfico de Zona de Calor por Provincia	220
Figura 3.42 Página Principal del Agente	221
Figura 3.43 Historial de Llamadas del Agente	221
Figura 3.44 Actualización del Perfil del Agente	222
Figura 3.45 Descarga del Aplicativo de Escritorio del Agente	222
Figura 3.46 Página de Cierre de Sesión.....	223
Figura 3.47 Diagrama Organizacional del Sitio Web de <i>Call Manager</i>	224
Figura 3.48 Diagrama de Clases del Cliente de Escritorio del Agente.....	226
Figura 3.49 Aplicativo de Escritorio en el Transcurso de una Llamada	227
Figura 3.50 Ventana Principal del Aplicativo de Escritorio en Estado Inactivo.....	228
Figura 3.51 Procesamiento Detallado de Comandos en el Aplicativo de Escritorio .	229
Figura 3.52 Procesamiento de Eventos Generados por la Central Telefónica.....	231
Figura 3.53 Implementación del Procesamiento de Eventos Generados por el Agente de <i>Call Center</i>	234
Figura 3.54 Ejemplo de Implementación de la Generación de una Llamada Telefónica en <i>Call Manager</i>	235
Figura 3.55 Diagrama de Secuencias de Inicio de Sesión de los Agentes	236
Figura 3.56 Diagrama de Secuencias de Cierre de Sesión de Agentes	237

Figura 3.57 Licenciamiento <i>CTI Link Pro</i> Activo en IP Office.....	240
Figura 3.58 Creación del Paquete Instalador del Servidor CTI.....	242
Figura 3.59 Comandos Utilizados en la Firma Digital de <i>Call Manager Desktop</i>	247
Figura 3.60 Escenario de Despliegue de la Solución	249
Figura 3.61 Escenario de Pruebas en la EPN	250
Figura 3.62 El Agente Recibe una Llamada Entrante.....	251
Figura 3.63 Central Telefónica Utilizada en las Pruebas	252
Figura 3.64 Configuración de Integración Básica con Aplicaciones Externas.....	256

ILUSTRACIONES DE ANEXOS

Ilustración B-1 Asistente de Instalación del TAPI de Avaya	B-1
Ilustración B-2 Selección del Directorio de Instalación del Controlador	B-1
Ilustración B-3 Tipo de Instalación Personalizado del Controlador TAPI de Avaya .	B-2
Ilustración B-4 Selecciones de Funciones del TAPI de Avaya.....	B-2
Ilustración B-5 Ingreso de Credenciales de Acceso del TAPI de Avaya	B-3
Ilustración B-6 Configuración del Proveedor de Servicio de Telefonía en Windows	B-3
Ilustración B-7 Parámetros de Configuración del Proveedor de Servicios	B-4
Ilustración B-8 Pantalla Inicial del Asistente de Instalación de <i>Call Manager</i> CTI....	B-5
Ilustración B-9 Selección del Directorio de Instalación de <i>Call Manager</i> CTI.....	B-5
Ilustración B-10 Ingreso del Puerto de Interconexión con el Servidor Principal	B-6
Ilustración B-11 Selección de Parámetros de Instalación de <i>Call Manager</i> CTI	B-6
Ilustración C-1 Selección del Directorio de Instalación de <i>PostgreSQL</i>	C-1
Ilustración C-2 Selección del Número de Puerto de <i>PostgreSQL</i>	C-2
Ilustración C-3 Creación de un Nuevo Rol de Inicio de Sesión en <i>PostgreSQL</i>	C-2
Ilustración C-4 Script de Creación de la BDD <i>CallManagerDB</i>	C-2
Ilustración C-5 Ejecución del <i>Script Tablas.sql</i>	C-3
Ilustración C-6 Ejecución del <i>Script InsercionDatos.sql</i>	C-3
Ilustración C-7 Instalación Personalizada de <i>GlassFish Server Open Source</i> <i>Edition</i>	C-4
Ilustración C-8 Especificación del Directorio de Instalación de <i>GlassFish</i>	C-4
Ilustración C-9 Ingreso de Información de Dominio de <i>GlassFish</i>	C-5
Ilustración C-10 Drivers JDBC de las BDD Soportadas Instaladas en <i>GlassFish</i>	C-5

Ilustración C-11	Página de Administración del Servidor de Aplicaciones	C-6
Ilustración C-12	Creación de un Nuevo <i>Pool</i> de Conexiones en <i>GlassFish</i>	C-6
Ilustración C-13	Parámetros de Creación de un Nuevo <i>Pool</i> de Conexiones	C-7
Ilustración C-14	Propiedades Adicionales de conexión a la BDD	C-7
Ilustración C-15	Comprobación de Acceso a la BDD desde el Servidor de Aplicaciones	C-8
Ilustración C-16	Recursos JDBC del Servidor de Aplicaciones	C-8
Ilustración C-17	Asociación del Pool de Conexiones con el Recurso JDBC para la BDD Local	C-9
Ilustración C-18	Sección Aplicaciones de la Consola de Administración de <i>GlassFish</i>	C-9
Ilustración C-19	Pantalla de Despliegue de Aplicaciones en <i>GlassFish</i>	C-10
Ilustración C-20	Carga de la Aplicación <i>Call Manager</i> en <i>GlassFish</i>	C-10
Ilustración C-21	<i>Call Manager</i> Correctamente Desplegado en <i>GlassFish</i>	C-11
Ilustración C-22	Pantalla de Inicio de Sesión de <i>Call Manager</i>	C-11
Ilustración D-1	Asistente de Instalación de <i>Call Manager Desktop</i>	D-1
Ilustración D-2	Ingreso del Directorio de Instalación de <i>Call Manager Desktop</i>	D-1
Ilustración D-3	Instalación de <i>Call Manager Desktop</i> Completada	D-2
Ilustración D-4	Prueba de Conectividad con el Servidor <i>Call Manager</i>	D-2
Ilustración D-5	Pestaña Seguridad del Panel de Control de Java en Ubuntu	D-3
Ilustración D-6	Adición del Servidor Principal en la Lista de Excepciones de Sitios	D-3
Ilustración D-7	Página Aplicación Cliente del Agente de <i>Call Center</i>	D-4
Ilustración D-8	Descarga del archivo JNPL desde el Servidor Principal	D-4
Ilustración D-9	Instalación de <i>Call Manager Desktop</i> vía <i>Java WebStart</i>	D-5
Ilustración D-10	<i>Call Manager Desktop</i> Instalado.....	D-5

Ilustración D-11 Configuración de Parámetros de <i>Call Manager Desktop</i>	D-6
Ilustración D-12 Parámetros de Comunicación de <i>Call Manager Desktop</i>	D-6
Ilustración D-13 Inicio de Sesión en <i>Call Manager Desktop</i>	D-7
Ilustración D-14 Ingreso de Credenciales de Inicio de Sesión en <i>Call Manager Desktop</i>	D-7
Ilustración E-1 Descarga de las Series Numéricas.....	E-1
Ilustración E-2 Selección de Campos del Plan de Numeración, Sección Anexos	E-2
Ilustración E-3 Borrado de Columnas Innecesarias.....	E-3
Ilustración E-4 Formato de Celdas para los Campos de la Serie Asignada.....	E-3
Ilustración E-5 Formato de la Serie Asignada de Tipo Texto.....	E-4
Ilustración E-6 Creación de Filtros en Campos serie, Operadora y Provincia.....	E-4
Ilustración E-7 Selección de Campos Vacíos de la Columna Serie Previa su Eliminación	E-5
Ilustración E-8 Almacenamiento de Archivos CSV	E-7
Ilustración E-9 Verificación de un Archivo de Tipo CSV	E-7
Ilustración E-10 Campos a Copiar del Plan de Numeración – Telefonía Móvil	E-8
Ilustración E-11 Concatenación de Recurso Numérico – Telefonía Móvil	E-9
Ilustración E-12 Rangos Numéricos no Deben Contener Separadores de Miles	E-9
Ilustración E-13 Página de Administración de Planes de Numeración de <i>Call Manager</i>	E-10
Ilustración E-14 Detalles del Plan Numérico – Zona 2.....	E-11
Ilustración E-15 Plan de Numeración a Ser Cargado en el Sistema.....	E-11
Ilustración E-16 Información Correctamente Almacenada en el Sistema	E-12
Ilustración E-17 Planes de Numeración Actualizados	E-12
Ilustración F-1 Asociación del <i>Pool</i> de Conexiones con el Recurso JDBC	F-2

Ilustración F-2	Página de Integración con BDD Externas en <i>Call Manager</i>	F-2
Ilustración F-3	Selección del Tipo de BDD en <i>Call Manager</i>	F-3
Ilustración F-4	Selección del Tipo de Origen de Datos en <i>Call Manager</i>	F-3
Ilustración F-5	Comprobación de los Recursos de Conexión con BDD Externas	F-3
Ilustración F-6	Selección de la Vista o Tabla a Utilizar en <i>Call Manager</i>	F-4
Ilustración F-7	Identificación de Atributos	F-4
Ilustración F-8	Confirmación de Parámetros de Interconexión	F-5
Ilustración F-9	Cambios Almacenados en el Sistema	F-5

ÍNDICE DE TABLAS

Tabla 1.1 Asignación del Indicativo Nacional de Destino	11
Tabla 1.2 Códigos de Numeración Geográficos	13
Tabla 2.1 Principales Metodologías de Desarrollo de Software.....	58
Tabla 2.2 Categorización de los Requisitos del Sistema	79
Tabla 2.3 Requisitos Primarios	80
Tabla 2.4 Requisitos Secundarios.....	80
Tabla 2.5 Versiones CTI de IP Office	82
Tabla 2.6 Comparación entre las Interfaces TAPI 2.0 y 3.0.....	95
Tabla 2.7 Caso de Uso – Gestión y Control de Llamadas Telefónicas	100
Tabla 2.8 Caso de Uso – Despliegue de Información de los Contactos en una Llamada	101
Tabla 2.9 Caso de Uso – Gestión e Ingreso de Observaciones de las Llamadas....	102
Tabla 2.10 Caso de Uso – Detalle del Número Telefónico de Contacto en una Llamada Externa	103
Tabla 2.11 Caso de Uso – Integración con Aplicaciones Externas.....	104
Tabla 2.12 Caso de Uso – Gestionar Usuarios, Servicios y Comunicaciones	105
Tabla 2.13 Caso de Uso – Monitoreo y Supervisión de Agentes	106
Tabla 2.14 Caso de Uso – Gestión de Reportes y/o Estadísticas	107
Tabla 2.15 Caso de Uso – Autenticar Usuario (Inicio de Sesión)	110
Tabla 2.16 Caso de Uso – Cierre de Sesión	110
Tabla 2.17 Caso de Uso – Realizar Llamada	112
Tabla 2.18 Caso de Uso – Colgar Llamada.....	113

Tabla 2.19 Caso de Uso – Contestar Llamada.....	114
Tabla 2.20 Caso de Uso – Retener Llamada	115
Tabla 2.21 Caso de Uso – Recuperar Llamada Retenida	116
Tabla 2.22 Caso de Uso – Transferir Llamada.....	118
Tabla 2.23 Caso de Uso – Recepción de Eventos	119
Tabla 2.24 Caso de Uso – Ingresar Observaciones	120
Tabla 2.25 Caso de Uso – Monitoreo de Agentes	121
Tabla 2.26 Caso de Uso – Envío de Mensajes a los Agentes	122
Tabla 2.27 Caso de Uso – Visualización de Registros, Reportes o Estadísticas	124
Tabla 2.28 Caso de Uso – Administración de Puestos de Trabajo (Extensiones) ...	125
Tabla 2.29 Caso de Uso – Administración de Usuarios.....	127
Tabla 2.30 Caso de Uso – Administración de Comunicaciones	128
Tabla 2.31 Caso de Uso – Administración de los Planes de Numeración	130
Tabla 2.32 Caso de Uso – Administración de Parámetros del Sistema	131
Tabla 2.33 Caso de Uso – Administración de Interconexión con Base de Datos.....	133
Tabla 2.34 Principales Alternativas de Interconexión entre los Componentes del Sistema	135
Tabla 2.35 Números de Puerto por Defecto Utilizados por el Sistema	136
Tabla 2.36 Protocolo de Comunicaciones	140
Tabla 2.37 Estados Soportados por la Central Telefónica Avaya a través de TAPI.	147
Tabla 2.38 Principales Funciones Soportadas por el TAPI 2.0 de Avaya.	151
Tabla 3.1 Características de los Principales Servidores Java EE Gratuitos.	179
Tabla 3.2 Principales Gestores de Bases de Datos Multiplataforma	181
Tabla 3.3 Principales Servicios de Mapas Disponibles.....	183
Tabla 3.4 Puertos Utilizados por <i>Call Manager</i>	244

Tabla 3.5 Subredes Utilizadas por <i>Call Manager</i> en el Escenario de Pruebas	249
Tabla 3.6 Cumplimiento General de Requisitos Primarios	253
Tabla 3.7 Cumplimiento General de Requisitos Secundarios	255
Tabla 3.8 Cumplimiento de Casos de Uso de Funcionalidades.....	259
Tabla 4.1 Valores de Constantes para los Modos del Modelo COCOMO	262
Tabla 4.2 Costo Estimado del Proyecto – Modelo COCOMO.....	263
Tabla 4.3 Número de Líneas de Código Efectivas de <i>Call Manager</i>	264
Tabla 4.4 Costos Fijos de Elaboración de <i>Call Manager</i>	265
Tabla 4.5 Cuadro Comparativo de Alternativas de Gestión de <i>Call Centers</i>	273

RESUMEN

El presente proyecto de titulación describe el diseño y la implementación de un software de control de líneas telefónicas de *Call Center* que incluye funcionalidades de georreferenciación de números telefónicos fijos. Para ello utiliza interfaces de programación basados en *Computer Telephony Integration* (CTI) disponibles para la central telefónica Avaya IP Office. El proyecto está ordenado de la siguiente forma:

El capítulo uno define las principales características de la plataforma de comunicaciones Avaya IP Office. Adicionalmente, se describen los conceptos relacionados a la tecnología de Integración Telefónica por Computador que permiten su combinación con sistemas computacionales; también se incluye una revisión de las tecnologías de desarrollo de aplicaciones utilizadas en la creación de este software.

En el segundo capítulo se detallan los requerimientos funcionales de la aplicación, la metodología de desarrollo de software empleada, el proceso de diseño de la misma y sus correspondientes fases.

El capítulo tres explica detalladamente el proceso de implementación (elaboración) de la aplicación planteada, así como también se describe la fase de despliegue (instalación) y su correspondiente evaluación en un escenario de pruebas.

El cuarto capítulo detalla el costo del proyecto, describe las aplicaciones y sistemas similares al propuesto, y realiza el correspondiente análisis comparativo técnico, funcional y económico del mismo.

El capítulo cinco presenta las conclusiones y recomendaciones desprendidas de la elaboración del proyecto.

Finalmente, se incluye una sección de anexos que contiene información adicional como el glosario, los procedimientos de instalación y configuración de la solución, la creación y carga de los planes de numeración, entre otros.

PRESENTACIÓN

De manera general, los sistemas tradicionales de telefonía pura y las aplicaciones institucionales dentro de las empresas operan de forma separada, lo que significa que en la realización o recepción de las llamadas telefónicas no es posible obtener información relevante de los contactos que participan en el proceso. Lo anterior repercute en mayores tiempos de respuesta por parte de los usuarios del servicio de telefonía en el *Call Center* (llámense agentes) y la utilización de dos (o posiblemente más) aplicaciones que proporcionen los datos necesarios para brindar un adecuado servicio a los clientes.

En vista que la plataforma de comunicaciones Avaya IP Office no es la excepción, y dado que las soluciones propietarias disponibles en el mercado no ofrecen la flexibilidad de ajuste al entorno ecuatoriano ni a los requerimientos de las instituciones nacionales, se ha visto necesaria la elaboración de un sistema que integre los eventos relacionados con las llamadas telefónicas y la información de los contactos almacenada en bases de datos de forma simple y flexible. Dicho sistema, por tanto, será un software que permita no sólo optimizar los procesos de los agentes y aumentar su productividad, sino también ofrecerá una mejor calidad de servicio a los clientes a través de una atención personalizada. Este tópico cobra mucha mayor importancia en situaciones donde la rápida localización de un contacto sea prioritaria, como aquella necesaria en servicios de salud, emergencias, bomberos y policía, aunque es aplicable a cualquier escenario nacional y extranjero.

Este proyecto de titulación está orientado a estudiantes y profesionales que tengan como objetivo crear aplicaciones informáticas distribuidas y/o de telefonía, así como también a todas aquellas instituciones interesadas en la georreferenciación de llamadas telefónicas. En este documento, por lo tanto, se detalla cada uno de los pasos seguidos en la creación de una solución informática que provea las funcionalidades y beneficios señalados en el párrafo previo, con un enfoque en el mercado ecuatoriano para pequeñas y medianas empresas a un costo asequible.

CAPÍTULO I

1. MARCO TEÓRICO

1.1. GENERALIDADES Y CONCEPTOS BÁSICOS DE TELEFONÍA

1.1.1. INTRODUCCIÓN

Las aplicaciones CTI (*Computer Telephony Integration*) están desarrolladas para que operen de forma conjunta con los sistemas telefónicos tradicionales.

En los apartados siguientes se describen los principales fundamentos, componentes y sistemas que hacen posible la creación de estándares y aplicaciones CTI.

1.1.2. TELEFONÍA FIJA ^[3]

La Agencia de Regulación y Control de las Telecomunicaciones¹ (ARCOTEL) define a la telefonía fija como: “*un servicio que permite el transporte de voz en tiempo real entre dos terminales (abonados), estando ambos terminales, o al menos el terminal de origen (que realiza la llamada), conectados a una red de telecomunicaciones en una ubicación fija.*” ^[E-1]

Previo al establecimiento de una llamada telefónica debe crearse un circuito de comunicaciones entre los interesados, con base en un proceso de señalización que comienza tan pronto como el abonado levanta el auricular. El equipo encargado del establecimiento de la comunicación y del enrutamiento de las llamadas a los destinos correspondientes, recibe el nombre de central telefónica o central de conmutación.

1.1.3. RED DE TELEFONÍA PÚBLICA CONMUTADA (RTPC O PSTN) ^{[1][3]}

La Red de Telefonía Pública Conmutada es una red de comunicación conformada por equipos, medios de transmisión y dispositivos que permite brindar servicios de telefonía fija.

¹ Entidad que reemplazó a la desaparecida Superintendencia de Telecomunicaciones (SUPERTEL) según el registro oficial N° 439 de la Ley Orgánica de Telecomunicaciones.

Si bien la RTPC inicialmente fue creada para soportar sólo tráfico de voz, en la actualidad también es utilizada para la transmisión de datos debido a su gran flexibilidad, lo que a su vez proporciona ventajas importantes como calidad, administración y capacidad de ser extendida fácilmente.

Al cableado instalado entre la central telefónica del proveedor de servicios (también llamada *Central Office* o CO) y las dependencias del usuario se denomina bucle local o bucle de abonado. En una RTPC, dos o más COs se interconectarán para que la generación de una llamada sea posible.

1.1.4. PROCESO DE UNA LLAMADA TELEFÓNICA ^[1]

El procedimiento básico para la realización y recepción de llamadas telefónicas ha sufrido pequeños cambios desde su establecimiento a finales del siglo XIX. A continuación se describe el flujo básico de una llamada:

- **Responder una Llamada.** Para la recepción de llamadas en teléfonos analógicos, la central de conmutación envía un voltaje de 70 a 90 voltios al circuito telefónico, haciendo que el teléfono timbre. Cuando el auricular es descolgado, la CO determina que el circuito está cerrado y listo para la recepción de la señal analógica. De forma similar, en los teléfonos digitales e IP la aplicación de telefonía reconoce la llamada entrante, inspecciona el estado del teléfono y recibe la señal digital con base en las acciones del usuario.
- **Identificación del Llamante.** En una llamada, para que sea posible reconocer el número telefónico participante, el operador de telefonía debe soportar el servicio de identificación de llamadas (*Caller ID*) o el Servicio de Identificación del Número Marcado (DNIS²). Una vez identificado el número, las aplicaciones CTI pueden utilizar esta información para presentar ventanas emergentes (o de notificación) con la información recuperada de una base de datos relativa al

² **Dialed Numbered Identification Service**, es un servicio provisto por las compañías de telecomunicaciones a sus clientes corporativos que permite determinar el número que fue marcado por un cliente.

participante; también es posible personalizar la atención de la llamada en función del número marcado por el cliente.

- **Realización de la Llamada.** En líneas analógicas, digitales e IP, tan pronto como se descuelga el auricular del teléfono, el primer paso para la realización de una llamada consiste en el envío de una señal de petición de servicio o de información de actividad de la línea. A esto la central telefónica responde con una señal de retorno que produce el tono de invitación a marcar, indicando que es posible realizar la llamada.
- **Marcación.** La marcación es el envío de señales que identifican el número telefónico a llamar. Para esto existen varias tecnologías que el presente estudio no requiere examinar a profundidad, entre las cuales se pueden mencionar la marcación por pulsos (discado) y la marcación por tonos DTMF (*Dual-Tone Multi-Frequency*).
- **Progreso de la Llamada.** Desde el transcurso hasta la terminación de una llamada, la RTPC y la mayoría de las centrales telefónicas generan tonos audibles que informan al usuario del estado de la misma. Entre los tonos que identifican el estado de una llamada están: marcado, en espera, ocupado, error, timbrado, entre otros.
- **Terminación de la Llamada.** Cuando se cuelga el auricular del teléfono se finaliza la llamada liberando el circuito establecido.

1.1.5. PBX ^{[1][3]}

Un PBX (siglas en inglés de *Private Branch Exchange*), traducido al español como un Ramal Privado de Conmutación, se define como una central telefónica autónoma con la capacidad de gestionar las llamadas internas, entrantes y salientes e interconectada a la RTPC a través de líneas troncales. De forma general, el equipo pertenece a la empresa que lo tiene instalado y no al proveedor de servicios de telefonía (de ahí que sea privado y autónomo).

En la actualidad es posible encontrar centrales telefónicas IP-PBX híbridas que soportan la comunicación entre líneas analógicas, digitales, e IP.

1.1.5.1. Componentes

Entre los principales componentes que conforman una plataforma de comunicaciones y que involucra una PBX están los siguientes.

- **Estaciones (teléfonos).** Son los dispositivos terminales (los cuales pueden ser teléfonos digitales, analógicos e IP) que utilizan los recursos del sistema de comunicaciones. Generalmente, una estación está asociada a un número de extensión previamente configurado en el sistema.
- **Gabinete Principal.** Es el equipo físico que contiene los componentes internos fundamentales de la PBX, entre ellos la red interna de conmutación, CPU, memorias, dispositivos de almacenamiento, fuentes de alimentación, buses de comunicación y de expansión, etc. En la actualidad, la gran mayoría de los sistemas son modulares, permitiendo extender (o ampliar) las capacidades de la central telefónica en caso de ser necesario.
- **Tarjetas Lógicas.** Corresponden al conjunto de tarjetas que tienen funciones especializadas que facilitan la operación de una PBX. Las tarjetas de entrada/salida (I/O) son utilizadas para acceder al equipo con el propósito de enviar o recibir información, en su mayoría instaladas como tarjetas de red de tipo Ethernet. Hay tarjetas especiales que permiten la conexión de las estaciones a la central telefónica, dependiendo de la tecnología (analógica o digital) de cada una de ellas.

Por otro lado, se pueden encontrar módulos que realizan la compresión de canales de voz con el objeto de transformar las señales en formatos más apropiados para su manipulación, tal es el caso de ambientes híbridos donde se pueden realizar llamadas entre teléfonos analógicos e IP.

Finalmente, también existe tarjetería dirigida a la interconexión de la PBX con enlaces troncales (que enlazan redes externas como la RTPC u otras centrales telefónicas), las cuales pueden ser de tipo E1, T1, analógicas, entre otras.

- **Consola o Terminal de Administración del Sistema.** Permite gestionar, configurar y en algunos casos programar la operatividad de la plataforma. Puede ser de tipo gráfico (GUI) o por línea de comandos.
- **Otros Equipos o Servidores.** En la actualidad las centrales telefónicas ofrecen características adicionales que van más allá de la prestación del servicio de telefonía. Muchas de ellas utilizan servidores o equipos adicionales para el efecto.
- **Cableado de Interconexión y Enlaces Troncales.** El cableado interno comunica las estaciones con la PBX, mientras que los enlaces externos troncales enlazan la central telefónica con el proveedor de servicios de telefonía o con otras PBX.
- **Sistemas de Alimentación Eléctrica.** Son el cableado eléctrico y los equipos necesarios para que el sistema de comunicaciones funcione apropiadamente. En la mayoría de casos se cuenta con un Sistema de Alimentación Ininterrumpida (en inglés *Uninterruptible Power Supply*, UPS) que proporcione energía previamente almacenada en baterías en caso de un apagón, manteniendo operativos los servicios de telefonía.

1.1.5.2. Características

Las centrales telefónicas presentan varios conjuntos de características y funcionalidades, los cuales pueden variar para cada tipo de plataforma o según el tipo de licenciamiento. La adición o remoción de alguna característica o funcionalidad particular dependerá de las necesidades de la organización. A continuación se mencionan algunas de las más comunes.

- Distribución automática de llamadas (ACD),
- Identificación de llamadas.
- Bloqueo de llamadas.
- Correo de voz.
- Reenvío de llamadas.
- Captura de llamadas (*call pickup*).

- Clases de servicio.
- Transferencia de llamadas.
- Llamadas en conferencia.
- Traducción de números.
- Marcación directa.
- Último número marcado (*redial*).
- Tono o música en espera.

1.1.6. TELEFONÍA IP ^{[1][3]}

1.1.6.1. Definición

La gran incursión de las redes de datos, y sobre todo su enorme flexibilidad, ha hecho posible la transición de tecnologías que utilizaban recursos analógicos a entornos puramente digitales, como ha sido el caso de la telefonía.

Aunque hay muchas variantes de su definición, la telefonía IP permite la comunicación de voz a través de una red de datos y es flexible respecto al medio físico en el que ocurre.

1.1.6.2. Tecnologías de Transmisión de Voz sobre Redes de Datos

Aunque en general la Telefonía IP y VoIP se consideran intercambiables, son diferentes, ya que la segunda es una estandarización de la primera. Sin embargo, es importante recalcar que existen otras tecnologías que permiten el envío de la voz sobre una red de datos, mismas que serán expuestas rápidamente en los siguientes apartados.

- **Voz sobre Protocolo de Internet (VoIP).** La tecnología VoIP es el conjunto de normas, protocolos y dispositivos que permite la transmisión de señales de voz sobre una red de datos utilizando el protocolo IP (Protocolo de Internet). La utilización de esta tecnología trae sus ventajas, como la utilización y mantenimiento de una sola infraestructura de comunicaciones (red de datos), abaratamiento de costos, posibilidad de crear e integrar aplicaciones

informáticas más fácilmente, entre otras. Por otro lado, el hecho que tengan que fragmentarse los paquetes IP para garantizar una entrega del tráfico de voz con mínimos retardos hace que el *overhead* generado pueda requerir un mayor ancho de banda y un medio de transmisión en buenas condiciones para garantizar una adecuada calidad en la comunicación.

- **Voz sobre ATM (VoATM).** El servicio usado para transmitir paquetes de voz sobre redes ATM recibe el nombre de VoATM. Entre sus ventajas estaba su capacidad para controlar sus mecanismos de variación de retraso y calidad de servicio, así como la reservación de ancho de banda que garantiza la calidad de las comunicaciones. Entre los detalles que potenciaron su desuso, está la costosa implementación de esta tecnología en ambientes de escritorio.
- **Voz sobre Frame Relay.** Es una tecnología rentable y madura, principalmente utilizada en proveedores de servicios o grandes instituciones, que utiliza el protocolo de capa 2 Frame Relay para el transporte común de datos, voz y vídeo. Posee mecanismos de control de tráfico que la convierten en una alternativa interesante, pero que no ha logrado despegar a nivel de usuario final por la gran incursión de la tecnología Ethernet y sus bajos precios.
- **Voz sobre MPLS.** Al igual que el modelo anterior, la tecnología de transmisión de voz sobre MPLS es utilizada en proveedores de servicios o instituciones ingentes que requieran un control de tráfico sofisticado o una creación simplificada de VPNs en sus comunicaciones.

1.1.6.3. Conceptos Adicionales

Si bien es cierto que las principales características de la telefonía convencional se replican en su similar IP, hay algunos componentes y definiciones adicionales que deben ser detallados.

- **Gateway.** En telefonía IP, un *gateway* se coloca entre una red conmutada y una red de telefonía IP, codificando y traduciendo las señales analógicas en digitales y viceversa. Generalmente, los *gateways* que conectan dispositivos

analógicos terminales (como teléfonos o faxes) reciben el nombre de Adaptadores Telefónicos Analógicos (ATA).

- **Teléfonos IP.** Los teléfonos IP se pueden clasificar en dos clases:
 - **Hardphones.** Son teléfonos fijos que se conectan a la red mediante conectores RJ-45, con base inalámbrica o USB.
 - **Softphones.** Son aplicaciones informáticas que simulan el funcionamiento de su equivalente físico y que pueden ser usados desde cualquier ordenador.
- **Centrales Telefónicas IP (IP PBX).** Este tipo de centrales realizan las mismas funciones que las convencionales e incluso implementan otras nuevas; son diseñadas para comunicar tráfico digital de voz (y en algunos casos vídeo) sobre una red de datos. Las centrales híbridas pueden operar en ambientes legados donde funcionen equipos analógicos y digitales con los IP.
- **Gatekeeper.** Con el objeto de controlar y administrar los recursos de telefonía IP, un gatekeeper actúa como una pequeña central telefónica gestionando las comunicaciones entre varios dispositivos IP de la red o con proveedores de servicios. En algunos casos son conocidos como *Session Border Controllers* por las funcionalidades de control de admisión de llamadas y servicios de traslación de números que pueden ofrecer.

1.1.6.4. Protocolos de Señalización

En la telefonía IP pueden utilizarse varios protocolos de señalización, algunos estandarizados y otros propietarios, siendo los más adoptados los protocolos H. 323 y SIP.

- **H.323.** Es un estándar de la ITU-T que determina cómo una sesión es establecida, monitoreada y terminada para tráfico audiovisual a través de una red de datos.
- **Protocolo de Inicio de Sesiones (SIP).** Es un protocolo estándar de la IETF creado para iniciar, modificar y finalizar sesiones interactivas que contienen tráfico multimedia. En los últimos años ha ganado popularidad frente a H. 323

especialmente por la flexibilidad del uso de otros protocolos, lo que aumenta la eficiencia de las comunicaciones.

1.1.7. PLANES DE NUMERACIÓN ¹⁴

Los planes de numeración establecen las bases numéricas y las estructuras de marcación que permiten la comunicación telefónica con otras redes o equipos a través de las redes públicas o externas, pudiendo ser éstas nacionales o internacionales.

1.1.7.1. Plan Técnico Fundamental de Numeración (PTFN) del Ecuador

En el Ecuador, el Consejo Nacional de Telecomunicaciones (CONATEL) como ente regulador elabora y administra los Planes Técnicos Fundamentales (PTF), siendo uno de los más relevantes el PTFN.

El formato del plan de numeración para el Ecuador está detallado en la resolución TEL-068-04-CONATEL-2013 (http://www.regulaciontelecomunicaciones.gob.ec/wp-content/uploads/downloads/2013/07/plan_tecnico_fundamental_numeracion.pdf). En su parte esencial el PTFN declara:

“El presente PTFN diseñado para cubrir las necesidades actuales y futuras de capacidad numérica para números geográficos y números no geográficos tales como bloques identificables para telefonía móvil, números personales (Como UPT Recomendación UIT-T E.168), números de servicios que se asocian a un número particular (como el servicio de RI), otros como servicios nomádicos, M2M, Data card y otros que pueden presentarse en el futuro debido a los adelantos tecnológicos y a la convergencia.

El PTFN define también el sistema de numeración adoptado, la distribución de los indicativos nacionales de destino, las áreas de numeración geográfica, las estructuras de los números para los diferentes servicios y el sistema de administración de la numeración.”

1.1.7.2. Acrónimos

Los acrónimos aquí presentados son aquellos de mayor relevancia e importancia para el desarrollo de este proyecto y mejor comprensión de este apartado.

- **CA:** Código de Acceso.
- **CC:** *Country Code*, indicativo de país para áreas geográficas (593).
- **DN:** *Destination Network*, código de red o indicativo de red destino.
- **NA:** Número de Acceso.
- **NAS:** Número de Abonado al Servicio.
- **NDC:** Indicativo (o Código) Nacional de Destino.
- **N_(s)N:** Número Nacional Significativo.
- **SN:** Número de Abonado.
- **TC:** *Trunk Code*, indicativo interurbano o código de área.

1.1.7.3. Estructura del Número Telefónico Internacional del Ecuador

Considerando la adopción de la estructura de numeración establecida en la recomendación E.164 de la UIT-T, la longitud máxima del número internacional debe ser de 15 cifras, sin incluir el prefijo internacional de acuerdo a la siguiente distribución:



Figura 1.1 Estructura del Número Internacional del Ecuador^[4]

En el Ecuador el número nacional significativo (**N_(s)N**) se compone de 9 dígitos, conformado por el Código Nacional de Destino (**NDC**) de dos dígitos y el número de abonado (**SN**) de 7 dígitos, como se muestra en la figura 1.2.

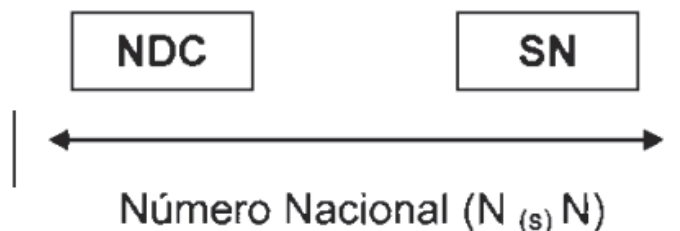


Figura 1.2 Estructura del Número Nacional (Significativo) ^[4]

1.1.7.4. Asignación General del Indicativo Nacional de Destino (NDC)

La tabla 1.1 presenta la estructura y asignación del NDC.

NDC	TIPO	NÚMEROS/CÓDIGOS
2X	TC	Geográficos: 22, 23, 24, 25, 26, 27 y 28
3X		Reserva
4X		Reserva
5X	DN	No geográfico: Telecomunicaciones Personales, Universales UPT: 51, 52, 53, 54 y 55
6X		Reserva
7X	DN	No geográfico acceso a redes: Servicios nómadas: 71, 72 y 73; Servicios M2M: 76, 77, 78 y 79
8X	DN	No geográfico: Categoría de Servicios y otros
9X	DN	No geográfico acceso a redes móviles

Tabla 1.1 Asignación del Indicativo Nacional de Destino ^[4]

De la tabla en mención es importante destacar que el Código Nacional de Destino (NDC) pasa a ser llamado Código de Área (TC) para llamadas de tipo geográficas³, mientras que para llamadas no geográficas de acceso a redes móviles se convierte en código de red (DN).

³ El NDC conformado por dos dígitos para llamadas geográficas será implementado en la fase IV del plan de implementación del PTFN que comenzará el 27 de septiembre de 2015.

1.1.7.5. Número Nacional Telefónico: Numeración Geográfica

El número nacional telefónico tiene una distribución uniforme de 9 dígitos y está conformado por el Indicativo Interurbano o código de área (**TC**) de dos dígitos⁴ y el Número de Abonado (**SN**) de siete.



Figura 1.3 Composición del Número Nacional Telefónico ^[4]

1.1.7.6. Número de Abonado

Siete dígitos conforman el número de abonado, de los cuales tres o cuatro corresponden a la serie o código de central (**SC**) y los restantes al código de abonado (**CA**).

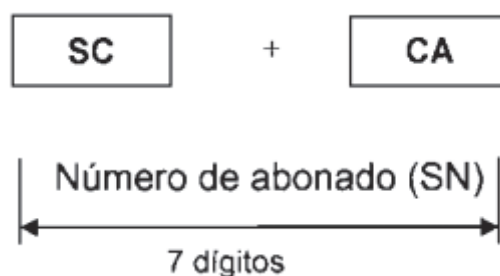


Figura 1.4 Composición del Número de Abonado ^[4]

1.1.7.7. Áreas y Códigos de Numeración Geográfica

Cada área de numeración geográfica se identifica con un código de área (TC) de dos dígitos⁵, cuya asignación se presenta en la tabla 1.2:

⁴ Véase nota 3

Nº	Área de Numeración	Códigos de Área	Cobertura Geográfica (Provincias)
1	22	22	Pichincha y Sto. Domingo de los Tsáchilas.
2	23	23	Tungurahua, Cotopaxi, Pastaza, Bolívar, Chimborazo.
3	24	24	Guayas y Península de Santa Elena.
4	25	25	Manabí, Galápagos, Los Ríos.
5	26	26	Carchi, Imbabura, Esmeraldas, Sucumbíos, Napo, Orellana.
6	27	27	Azuay, Cañar, Morona Santiago.
7	28	28	Loja, Zamora, El Oro.

Tabla 1.2 Códigos de Numeración Geográficos ^[4]

1.1.7.8. Número Nacional Móvil: Numeración no Geográfica

Este número móvil está conformado por el Indicativo de Red (**DN**) de dos dígitos seguido por el número de abonado móvil (**SN**) de siete dígitos.

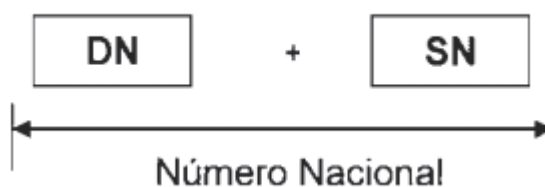


Figura 1.3 Número Nacional Significativo Móvil ^[4]

1.1.7.9. Plan de Marcación

El CONATEL define el plan de marcación como: “Una cadena o combinación de cifras decimales, símbolos e información adicional que definen el método según el cual se utiliza el PTFN. Un Plan de Marcación incluye la utilización de prefijos, sufijos

⁵ Véase nota 3

e información adicional, complementaria al Plan de Numeración y necesaria para completar la llamada.”

Dicho de otra forma un plan de marcación define el procedimiento que debe seguirse para la realización de las llamadas, mediante la utilización del PTFN y la adición de prefijos o números especiales según sea necesario.

1.1.7.10. Series Numéricas ^{[1][4]}

Las series numéricas corresponden a los conjuntos de números permitidos para la telefonía fija, móvil, red inteligente, números especiales y servicios suplementarios. Dichas series están basadas en el Plan Técnico Fundamental de Numeración, y generalmente son organizadas por rangos, códigos, zonas, proveedores, etc.

En el país, las series son actualizadas y publicadas mensualmente en la página web de la ARCOTEL, disponibles desde el enlace: <http://www.arcotel.gob.ec/plan-tecnico-fundamental-de-numeracion-series-numericas/>

En lo que concierne al desarrollo de este proyecto las series numéricas de relevancia son las siguientes:

- Informe series Tel – Fija: comprende la asignación del recurso numérico para el servicio de telefonía fija.
- Informe series SMA: contiene el plan de marcación para servicios móviles avanzados.

Las series numéricas utilizadas en este proyecto, corresponden a las publicadas en el mes de febrero del año 2015.

1.2. LA TECNOLOGÍA CTI ^{[1][2]}

1.2.1. INTRODUCCIÓN

Los sistemas telefónicos tradicionales y las tecnologías de la información nacieron de necesidades distintas, en períodos separados. La idea de combinarlos con el objeto de proporcionar un valor añadido a la experiencia del usuario frente a la plataforma

de comunicaciones, recibe el nombre de Integración Telefónica por Computador o CTI (*Computer Telephony Integration* por sus siglas del idioma inglés).

En otras palabras, CTI es la tecnología que permite la interacción entre teléfonos y computadores de forma recíproca y coordinada.

1.2.2. FUNCIONES Y CARACTERÍSTICAS DE LAS APLICACIONES CTI

Este apartado pretende definir brevemente algunas de las aplicaciones y características de la tecnología CTI, entre las cuales están:

- **Control de llamadas.** Manipulación de llamadas y funcionalidades del teléfono mediante una interfaz visual en el computador.
- **Softphone.** El programa informático reemplaza al teléfono físico, siendo administrado totalmente desde el ordenador.
- **Tarificación.** Obtención de información de utilización y gastos con relación a las llamadas realizadas. También es conocida con el nombre de *tasación*.
- **Despliegue de información de la llamada en pantalla.** Con base en el número telefónico es posible realizar búsquedas en bases de datos y presentarla al agente incluso antes de contestar la llamada.
- **Servicios activados por voz.** Como por ejemplo la conversión de texto a voz (*text-to-speech*) y viceversa.
- **Respuesta de Voz Interactiva (IVR).** Utilizado principalmente para interactuar con seres humanos, se basa en flujos de gestión de las llamadas entrantes, permitiendo el acceso a servicios de información o redirección de llamadas.
- **Control de Call Centers.** Es posible personalizar las colas de las llamadas entrantes para que sean redirigidas a los agentes con base en parámetros tales como la disponibilidad o aptitudes de los receptores de las llamadas.
- **Campañas de Llamadas.** Sistemas que gestionan y automatizan la realización de las llamadas salientes con propósitos de publicidad, actualización de datos, encuestas, entre otros.

- **Mensajería Unificada.** Integración de varias plataformas de comunicación en una sola, como telefonía, correo electrónico, chat, fax, etc.

1.2.3. ARQUITECTURAS DE INTEGRACIÓN CTI ^{[1][2][5]}

La creación de programas CTI puede seguir cualquiera de las dos arquitecturas siguientes, en función de las necesidades de las instituciones interesadas.

1.2.3.1. *First-Party* (Acceso Directo o de Primera Parte)

En este tipo de arquitectura un solo computador controla y manipula las funcionalidades de un teléfono particular, es decir, gestiona los eventos y las llamadas de una sola extensión.

Las aplicaciones *first-party* en su mayoría se comunican directamente con el dispositivo telefónico, sin necesidad de utilizar los recursos de la central telefónica o de un servidor de comunicaciones para el efecto, aunque para ello suele ser necesaria la conexión directa entre el computador y el teléfono, pudiéndose requerir tarjetas o conectores adicionales.

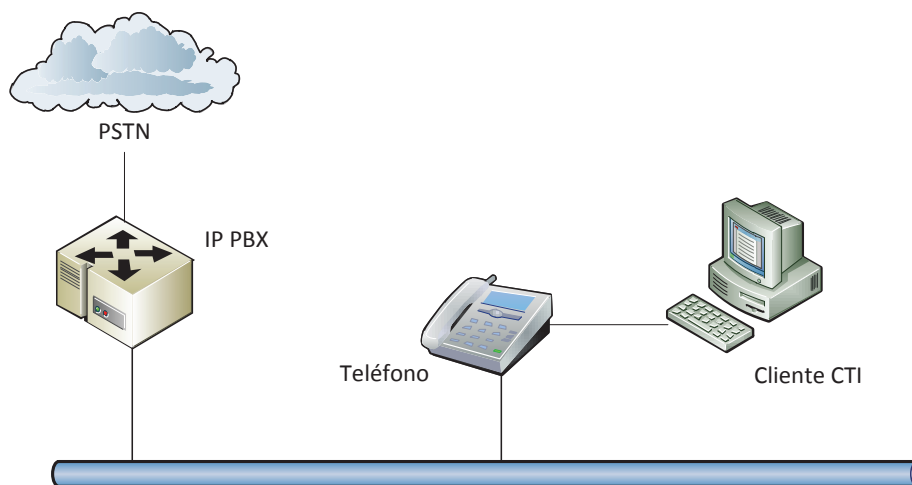


Figura 1.5 Ejemplo de Arquitectura *First-Party* ^[5]

Este esquema se ajusta a entornos de desarrollo de aplicaciones en los que se requieren funcionalidades básicas o están dirigidas a un número pequeño de

usuarios. Entre sus ventajas, algunas centrales telefónicas comerciales ya incluyen en su licenciamiento básico la posibilidad de integración CTI a través de un control de llamadas *first-party*, por lo que no es obligatoria la adquisición de licencias adicionales para su operación.

Por otro lado, la principal desventaja de esta tecnología proviene de su propia definición. *First-party* implica que un único ordenador interactúa solamente con la línea telefónica que tiene asignado, lo que significa que en organizaciones con un gran número de usuarios la complejidad relacionada a la distribución, el mantenimiento y la administración de aplicaciones CTI aumenta enormemente. En este caso es más útil la utilización de la tecnología *Third-party*.

1.2.3.2. *Third-Party* (Acceso Indirecto o de Tercera Parte)

La integración *third-party* permite monitorear, administrar y controlar una o más extensiones desde una sola aplicación CTI.

Debido a que es posible gestionar otras líneas desde un sistema centralizado, esta alternativa suele ser preferida en escenarios donde se requiere administración avanzada (monitoreo, tarifación, reportes) o donde no es posible instalar o ejecutar aplicaciones *first-party* en los computadores de los usuarios.

Entre las principales consideraciones que se deben tomar en cuenta para el desarrollo de este tipo de sistemas, están:

- Aumento del nivel de complejidad de desarrollo, debido a la administración y manipulación de varias líneas (extensiones) a la vez.
- Algunas plataformas de comunicaciones requieren la adquisición de licenciamiento adicional que permita la interacción con aplicaciones CTI.
- Se requiere de un equipo central (servidor) que administre todas las conexiones, pudiendo dejar inoperantes a todos los usuarios en caso de fallas.

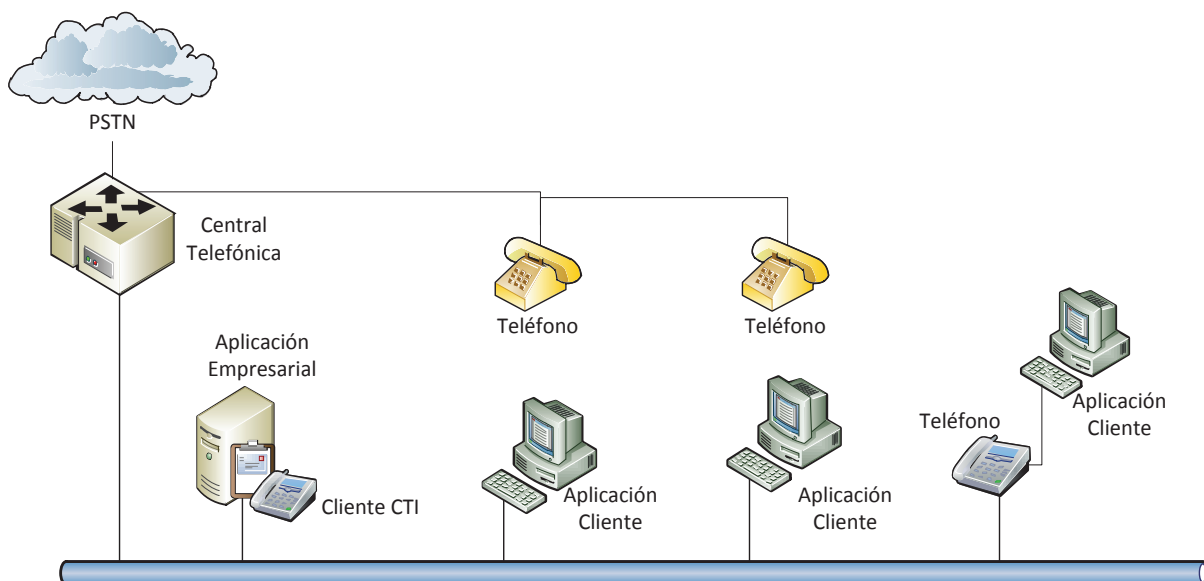


Figura 1.6 Ejemplo de la Arquitectura *Third-Party* [2]

1.2.4. ESTÁNDARES CTI [1]

CTI tiene sus orígenes a principios de los años 80, cuando AT&T creó un sistema que proporcionaba funciones de directorio y centro de mensajes integrado a su plataforma de comunicaciones “*Dimension PBX*”. A partir de ese entonces los fabricantes se enfocaron en crear sus propias tecnologías y plataformas de integración con las centrales telefónicas, obligando a los desarrolladores a crear distintos tipos de programas para cada sistema. Ante tal situación, fue necesaria la elaboración de estándares e interfaces que permitieran la portabilidad de las aplicaciones (con un nivel de esfuerzo razonable) y simplificaran el aprendizaje y desarrollo de las mismas.

La tecnología CTI se basa en Interfaces de Programación de Aplicaciones (o APIs por sus siglas en inglés), que son un conjunto de procedimientos y métodos publicados como una capa de abstracción (interfaz) para la comunicación entre componentes de software. En telefonía existen varias APIs de uso comercial, entre ellas TAPI, TSAPI, JTAPI, CSTA y el API de Telefonía de Linux.

En los subcapítulos posteriores se describirán cada una de las interfaces antes señaladas, con una especial atención al API de Telefonía de Windows (TAPI), por ser de especial interés en el desarrollo de este proyecto.

1.2.4.1. Interfaz de Programación de Aplicaciones de Telefonía (TAPI)

Fue desarrollada por Intel y Microsoft en el año 1994 como alternativa a soluciones CTI dependientes del hardware. Es la interfaz más popular debido a su amplio conjunto de funcionalidades (con el correspondiente aumento de complejidad) y la posibilidad de ejecutar aplicaciones CTI desde cualquier sistema operativo Windows.

Para que las funciones y los comandos de la interfaz puedan comunicarse con el hardware subyacente, es necesario que el fabricante proporcione el código de bajo nivel (o *driver*) que debe ser instalado en el equipo. Estos *drivers* reciben el nombre de Proveedores de Servicios TAPI (TSPs), que serán detallados posteriormente.

Debido a que no todas las plataformas de comunicaciones (hardware) poseen las mismas características, no todas las funciones y los métodos TAPI podrían ser soportados por cada TSP. Para conocer las funcionalidades soportadas será necesario acudir a la información proporcionada por el fabricante.

1.2.4.1.1. Interfaces ^[E-2]

Microsoft define cuatro interfaces de aplicaciones de telefonía descritas a continuación:

- **TAPI 2.x.** Es una interfaz basada en C/C++ que permite la implementación de aplicaciones que van desde el control básico de módems hasta gestión de *call centers* con múltiples agentes. Es la interfaz más utilizada por su rapidez y gran flexibilidad, pero requiere de un mayor grado de conocimiento y de cuidado en el diseño de aplicaciones.
- **TAPI 3.x.** Implementada como un conjunto de objetos COM, permite el desarrollo de aplicaciones TAPI desde lenguajes de programación como C#,

Visual Basic o C/C++. Está enfocado en combinar la telefonía clásica e IP, y provee características y funcionalidades especiales que permiten gestionar desde llamadas de voz sobre la RTPC hasta conferencias multimedia con calidad de servicio.

- **Interfaz de Provisión de Servicios de Telefonía (TSPI).** Es una interfaz que a través de comandos estandarizados soporta la gestión y control de los dispositivos de comunicaciones. Se provee como una librería de enlace dinámico (DLL).
- **Interfaz de Provisión de Servicios Multimedia (MSPI).** Otorga a las aplicaciones un considerable control del medio para un mecanismo de transporte particular. Un MSP siempre se comunica con un TSP.

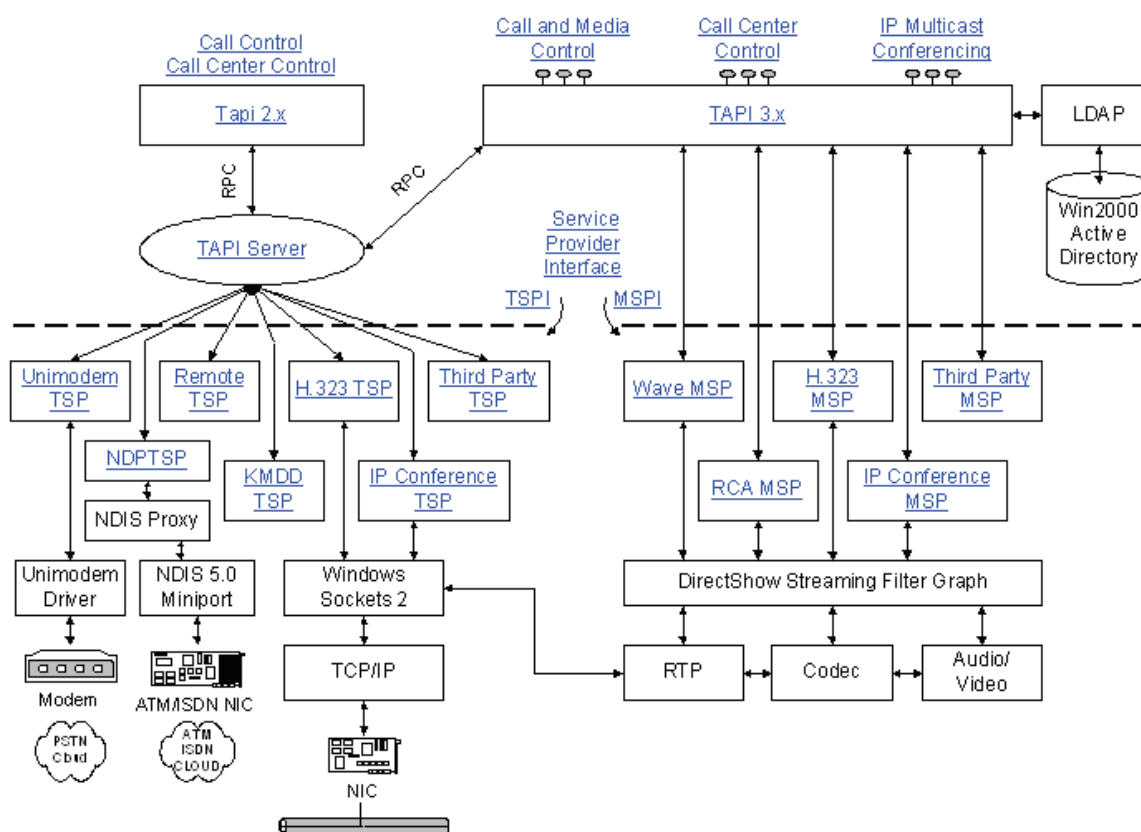


Figura 1.7 Arquitectura de Telefonía de Microsoft [E-3]

1.2.4.1.2. Arquitectura de Telefonía de Microsoft ^[E-3]

En la figura 1.7 se presenta el diagrama correspondiente a la arquitectura de telefonía de Microsoft. Como se puede apreciar, en la parte superior de la arquitectura las aplicaciones CTI acceden a las funcionalidades provistas por las interfaces TAPI 2.x y TAPI 3.x. Éstos a su vez invocan a los servidores de telefonía que a través de las interfaces TSPI (y MSPI en el caso del TAPI 3.x) se conectan a los drivers que comunican el hardware subyacente.

Es notable aclarar que la tecnología TAPI no está limitada a la RPTC, ISDN, o a TCP/IP.

1.2.4.1.3. Modelo de Programación de Telefonía ^{[E-4][E-5]}

Basado en la arquitectura anterior, el modelo de programación de telefonía de Microsoft exige a las aplicaciones CTI de la necesidad de manipular directamente las características de los dispositivos mediante la implementación de un mecanismo jerárquico de abstracción de funcionalidades, como se puede apreciar en la figura 1.8.

Un TAPI DLL, junto al servidor TAPI (Tapisvr.exe) constituyen los componentes principales capaces de abstraer y separar los proveedores del servicio de las aplicaciones CTI. Existen tres tipos de DLLs, Tapi.dll (16 bits, obsoleto), Tapi32.dll (32 bits) y Tapi3.dll, generalmente localizados el directorio de Windows `%SystemRoot%\system32`.

El servidor TAPI rastrea los recursos telefónicos locales y remotos con los TSPs (a través del TSPI). En computadores con Windows NT corre como un proceso separado, mientras que en sistemas operativos Windows 2000, 2003 y XP funciona en el contexto de Svchost.exe. Las comunicaciones que ocurren entre un TSP y un MSP toman lugar utilizando una conexión virtual que atraviesa el TAPI DLL y TAPISRV.

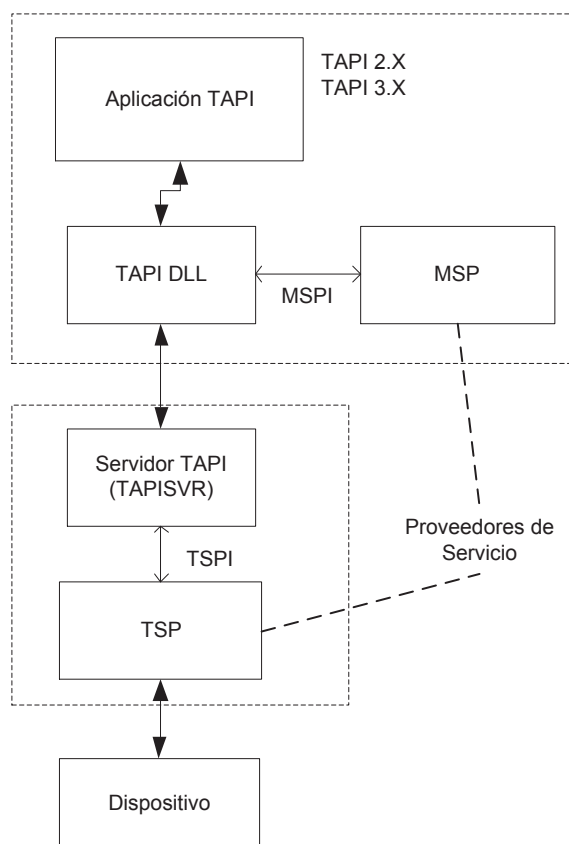


Figura 1.8 Modelo de Programación de Telefonía de Microsoft ^[E-4]

Los Proveedores de Servicio (TSPs) brindan funciones detalladas de los equipos telefónicos. Generalmente, un TSP proporciona controles de llamadas y la provisión de servicios de transmisión a través de un proveedor de servicio multimedia (MSP). El MSP controla el flujo de la información sobre el medio, el cual es ejecutado sobre el espacio de procesamiento de la aplicación (y no del DLL), permitiéndole obtener respuestas más rápidas de los equipos.

Una aplicación TAPI cumple generalmente con dos o más de los siguientes apartados:

- a) **Inicialización.** Obligatorio. Luego de instalar los drivers y componentes necesarios, el programa carga el TAPI DLL (de tipo 2.x ó 3.x) en su espacio de procesamiento. Posteriormente el sistema debe negociar la versión con la

que se trabajará (por ejemplo 2.1 ó 2.2) y se configuran los eventos que el sistema está dispuesto a gestionar.

- b) Control de la sesión.** En esta etapa es posible iniciar, recibir o manipular llamadas.
- c) Control del dispositivo.** Recuperación y configuración de la información del teléfono.
- d) Control del medio.** Detección o generación de tonos o dígitos.
- e) Control del *call center*.** Manejo avanzado de las características de los agentes, flujo de llamadas de los clientes y colas de distribución.
- f) Cierre del TAPI.** Mandatorio. Liberación de los recursos del sistema.

1.2.4.2. Interfaz de Programación de Aplicaciones de Servicios de Telefonía (TSAPI) ^[1]

Fue una de las primeras interfaces, desarrollada por Novell y AT&T y dirigida a la integración de PBX con los equipos de área local NetWare (de Novell). La interfaz cuenta con un abundante conjunto de características y funcionalidades, como el control y enrutamiento de llamadas, monitoreo, servicios de directorio y consulta. Adicionalmente, entre sus características avanzadas permite la gestión de agentes, enrutamiento basado en habilidades y gestión de grupos de interés.

Al ser diseñado en un modelo basado en servidor (de tipo *third-party*), este sistema fue ampliamente requerido en ambientes donde se disponía de un gran número de teléfonos analógicos, lo que representaba un menor costo de implementación. Otra de sus ventajas es que era soportado por múltiples sistemas operativos (a diferencia de TAPI que sólo opera en sistemas Windows).

Sin embargo, la creciente popularidad de TAPI y la decisión de Novell de discontinuar el soporte de la interfaz, han provocado su debilitamiento en el mercado informático.

1.2.4.3. Interfaz de Programación de Aplicaciones de Telefonía de Java (JTAPI) ^{[1][E-6]}

El API de Telefonía de Java es una especificación (no una implementación) que soporta el control de llamadas telefónicas, diseñada para trabajar tanto en

aplicaciones enfocadas a usuario (*first-party*) como en sistemas distribuidos (*third-party*).

La primera especificación, JTAPI 1.1 fue liberada en el año 1996 por Sun Microsystems (posteriormente adquirida por Oracle) en colaboración con varias compañías de telecomunicaciones. La última especificación fue publicada en el año 2002 correspondiente a la versión 1.4, y se espera que salga la versión 2.0 que está en desarrollo.

Entre las ventajas de esta tecnología están:

- Multiplataforma. Los sistemas desarrollados en JTAPI pueden ser trasladados a otras plataformas con un mínimo esfuerzo.
- Desarrollo simplificado, mediante el esquema de orientación a objetos y la reutilización de código.
- Menor tiempo de aprendizaje y amplia documentación.
- Interoperabilidad. Integración con aplicaciones basadas en TAPI y TSAPI.
- Soporte de características avanzadas, como enrutamiento de llamadas.

Entre las principales desventajas de esta plataforma se puede considerar la enorme influencia de TAPI, lo que hace que nuevas funcionalidades en esta última puedan convertirse en estándares *de-facto* en la industria. Adicionalmente, JTAPI no cuenta con el mismo nivel de acceso público que su principal competidor.

1.2.4.4. Aplicaciones de Telecomunicaciones con Soporte de Computador (CSTA)

CSTA, acrónimo del inglés *Computer-Supported Telecommunications Applications*, defendida inicialmente por la empresa Siemens y posteriormente adoptada por la ECMA (*European Computer Manufacturers Association*), provee una capa de abstracción a las aplicaciones de telefonía, independientemente de los protocolos subyacentes que participan en la comunicación.

Con un enfoque similar al TAPI 3.x de Microsoft, ha tenido mucha acogida entre las empresas principalmente por su carácter internacional, que en el año 2000 fue

convertido en un estándar ISO. En la actualidad se encuentra en su tercera fase, y se lo halla detallado en el documento ECMA-269, que puede ser descargado desde la página web:

<http://www.ecma-international.org/activities/Communications/TG11/cstalll.htm>.

Entre las características del esquema se pueden mencionar:

- Soporte de interacciones con y sin voz (e-mail, mensajería instantánea, etc.) con el mismo modelo de llamada.
- Complementa a SIP y permite que los desarrolladores provean características avanzadas.
- Soporte de control de llamadas *first-party* y *third-party*.
- Colaboración y conferencia.
- Enrutamiento y distribución de llamadas.
- Servicios de contabilidad y recolección de datos.

1.2.4.5. Interfaz de Programación de Aplicaciones de Telefonía de Linux^[1]

En los inicios de la creación de plataformas de integración, varias empresas crearon interfaces de telefonía para el sistema operativo Linux, sin mucho éxito. Algunas centrales telefónicas basadas en este SO (como Asterisk) proveen sus propias interfaces, aunque las tecnologías que han prevalecido han sido CSTA y JTAPI.

1.3. AVAYA IP OFFICE 500 V2^[5]

“IP Office es una solución de comunicaciones versátil que combina la confiabilidad y facilidad de uso de un sistema de telefonía tradicional con las aplicaciones y ventajas de una solución de telefonía IP.”^[5]

La solución de comunicaciones modular IP Office es una PBX híbrida con capacidades TDM e IP que pueden ser utilizadas de forma independiente o simultánea. Esto quiere decir que la central telefónica es capaz de brindar servicios de telefonía fija analógica, digital e IP.

1.3.1. CARACTERÍSTICAS ^{[5][6][7]}

Entre las principales características provistas por la plataforma de comunicaciones en cuestión están:

Comunicación de Voz

- Capacidades completas de una central PBX (reenvío de llamadas, conferencias, mensajes de voz, etc.).
- Soporte de enlaces troncales de tipo: E1, T1, *Primary Rate Interface* (PRI), *Integrated Service Digital Network* (ISDN), *Session Initiation Protocol* (SIP) y RTPC.
- Soporte de 2 a 384 extensiones.
- Compatible con teléfonos analógicos, digitales, *hardphones* y *softphones* IP.
- Enrutamiento avanzado de llamadas basado en los requerimientos de las organizaciones.
- Interoperabilidad con otras PBXs basados en estándares.
- *Gateway* y *gatekeeper* (H. 323 y SIP) integrado con la posibilidad de brindar calidad de servicio de las llamadas a través de servicios diferenciados.

Herramientas de Administración

- La central telefónica es administrada desde la aplicación informática *IP Office Manager*, para sistemas operativos Windows.
- Además cuenta con una herramienta de diagnóstico que provee detalles del estado del equipo y los recursos del sistema, llamada *System Status*.

Comunicación de Datos

- Enrutamiento estático o dinámico basado en RIP (I/II).
- Capacidad de Traducción de Direcciones de Red (NAT) a través de un *firewall* integrado, con soporte de IPsec para proteger las comunicaciones.
- Puede actuar como servidor DHCP para equipos y dispositivos locales y remotos.

- Posee 2 interfaces de red Ethernet (1 LAN y 1 WAN).
- Integra un cliente LDAP para la sincronización de directorios basados en estándares.

Plataforma de Comunicaciones

Algunas de las funcionalidades aquí descritas pueden requerir de licenciamiento adicional.

- Consola de gestión de llamadas *SoftConsole*.
- La aplicación web *Avaya One-X Portal* permite a los usuarios controlar y administrar las llamadas mediante un navegador web con posibilidad de integración con servicios de directorio.
- IP Office provee un *Kit* de Desarrollo de Software (SDK) basado en interfaces abiertas (TAPI) que permite la creación e integración de aplicaciones CTI.
- Brinda servicio de correo de voz, con la posibilidad de copiar y transformar los mensajes de voz en mensajes de correo electrónico que puedan ser enviados al sistema de mensajería local.
- Provee Servicios de Mensajería Unificada (UMS) que permitan integrar y sincronizar los mensajes de correo electrónico y los mensajes de voz en un solo repositorio.
- Asistente automático de llamadas (*auto-attendant*) que permite la interacción y navegación de los usuarios con el sistema, sin la asistencia de un operador.
- Respuesta de Voz Interactiva (IVR) y *text-to-speech*.
- Permite rastrear y medir la productividad de los agentes y el servicio a los clientes a través de la aplicación *Customer Call Reporter* o CCR.

Escalabilidad de la Plataforma

La central telefónica IP Office cuenta con un chasis modular con capacidad de hasta:

- 16 enlaces troncales analógicos.
- 16 enlaces troncales ISDN BRI.

- 8 troncales digitales PRI (hasta 240 canales E1).
- 148 canales de compresión de voz.
- Adición de hasta 12 módulos de expansión que proveen una combinación de hasta 384 extensiones analógicas, digitales e IP.

Esquema de Licenciamiento

Algunas aplicaciones y funcionalidades del sistema requieren de licencias para su activación. Cada licencia es generada como una combinación del identificador de la característica a activar con el número de la *Feature Key* (llave de activación) de la central telefónica.

Las licencias son provistas como licencias de prueba (válidas por un tiempo máximo de 60 días) o como licencias indefinidas.



Figura 1.9 Entorno de comunicaciones IP Office ^[6]

1.3.2. ARQUITECTURA DE COMUNICACIONES IP OFFICE ^[6]

La solución de comunicaciones IP Office está compuesta por unidades de hardware y aplicaciones de software.

Cada solución IP Office requiere de una unidad de control de sistema (llamada IP500 V2) que podrá estar interconectada a otras a través de la RTPC o Redes de Área Extendida (WAN), enlaces troncales al proveedor de servicios, módulos de expansión (que brinden telefonía convencional TDM), servidores de aplicaciones propietarias y teléfonos y dispositivos finales que requieran servicios de comunicaciones.

1.3.2.1. Unidad de Control

La unidad de control IP500 V2 tiene 2U de altura que permite montar la estructura en un *rack* estándar de 19 pulgadas.

Posee cuatro ranuras compatibles con una amplia variedad de tarjetas complementarias o tarjetas base que proporcionan conectividad a teléfonos, y con tarjetas secundarias que proporcionan conexión al proveedor del servicio.

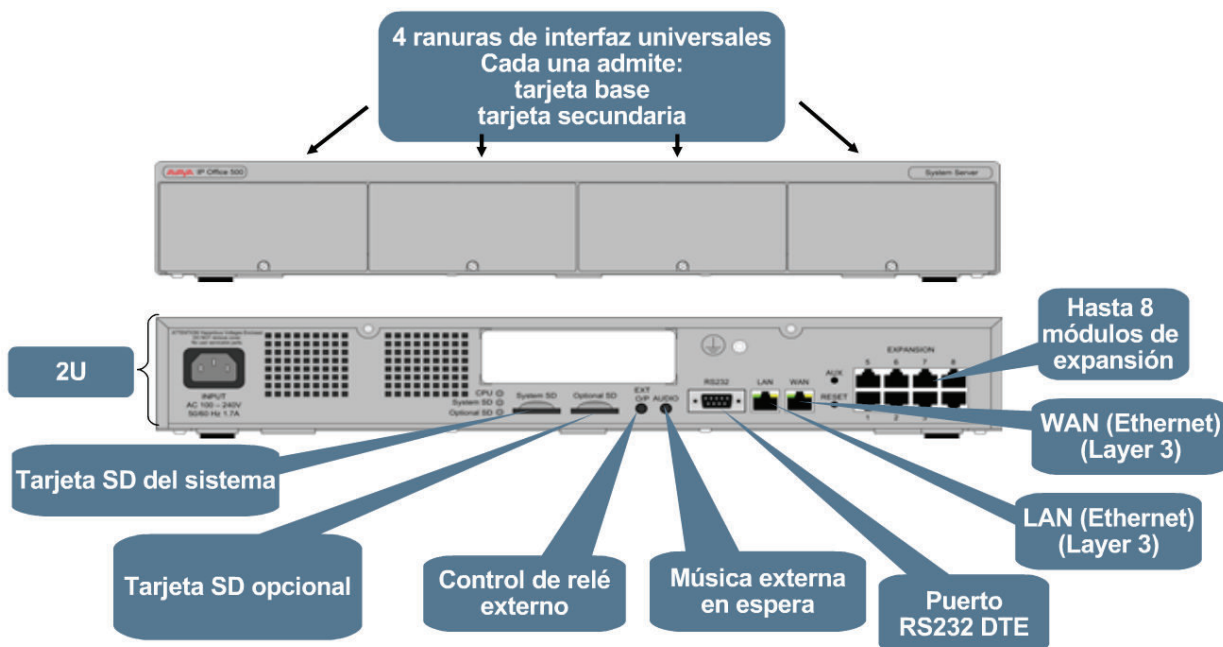


Figura 1.10 Vista Frontal y Posterior de la Unidad de Control IP500 V2 ^[7]

El sistema utiliza tarjetas digitales SD que almacenan el sistema operativo y el archivo de licencias de activación de aplicaciones y funcionalidades.

Añade un control de relé externo que proporciona conexión a dispositivos como porteros telefónicos.

Los puertos Ethernet LAN y WAN se utilizan para la conexión a la red de datos local y a servicios y redes externas respectivamente.

Los ocho puertos de expansión proporcionan la interconexión de módulos externos que amplían el soporte de comunicaciones a teléfonos y dispositivos.

El equipo incluye un puerto de entrada de audio para fuentes externas de música en espera, así como también un puerto RS232 DTE de 9 pines para mantenimiento.

1.3.2.2. Tarjetas Base

La unidad de control IP500 posee cuatro ranuras para la inserción de tarjetas, que pueden ser tarjetas base o tarjetas troncales. Las primeras incluyen un panel frontal y puertos para conexiones a través de cables. Las tarjetas troncales, en cambio, son montadas sobre las tarjetas base con el propósito de proveer comunicaciones adicionales (generalmente conexiones troncales).

Entre las tarjetas base disponibles están:

- **Tarjeta IP500 Digital Station 8.** Posee 8 puertos para la conexión de teléfonos digitales. Tiene 4 puertos adicionales que requieren de una tarjeta troncal para brindar servicios adicionales. Es posible instalar hasta tres tarjetas en una unidad de control.
- **Tarjeta IP500 Digital TCM Station 8.** Una variante del anterior que soporta 8 teléfonos digitales Nortel. Cada unidad de control soporta un máximo de cuatro tarjetas.
- **Tarjetas IP500 Analog Phone 2 y Analog Phone 8.** Proveen 2 y 8 puertos respectivamente para la conexión de teléfonos analógicos. Pueden instalarse hasta cuatro de éstas en una unidad de control.

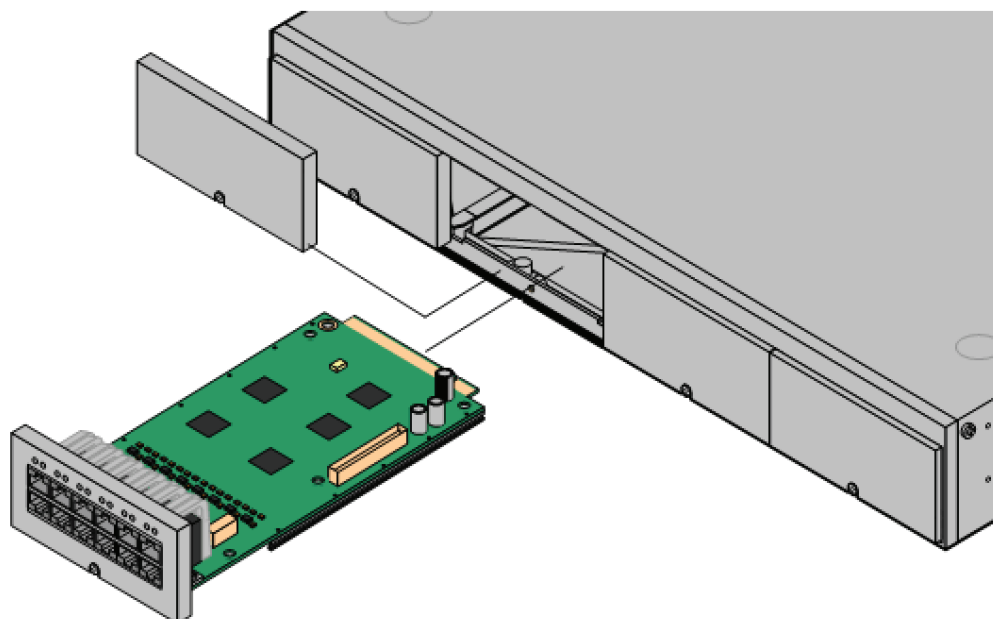


Figura 1.11 Colocación de una Tarjeta Base ^[5]

- **Tarjeta IP500 VCM.** Utilizada en llamadas de tipo VoIP y troncales SIP para la provisión de canales de compresión de voz, en variantes de 32 y 64 canales. IP500 V2 soporta hasta 2 tarjetas de este tipo.
- **Tarjeta de expansión de 4 puertos.** Insertada en la cuarta ranura de la unidad de control, esta tarjeta proporciona conectores adicionales para la adición de 4 módulos de expansión. Sólo puede instalarse una tarjeta por unidad de control.
- **Portadora de Tarjetas Legadas.** Permite que una unidad de control IP500 pueda utilizar una variedad de tarjetas troncales y VCM de su modelo anterior, IP400. Esta tarjeta es útil cuando se actualiza o se reemplaza un sistema de telefonía IP400 ya instalado. Cada unidad de control tolera un máximo de dos tarjetas.
- **Tarjeta telefónica IP500 ETR de 6 puertos.** Provee puertos para la conexión de teléfonos inalámbricos 3910/3920 y teléfonos digitales ETR 6/18/34 D, Refresh y Euro-Style. La unidad de control soporta un máximo de tres tarjetas.

- **Tarjeta combinada con 4 troncales analógicas.** Es una tarjeta que integra cuatro puertos troncales analógicos, dos puertos para la conexión de teléfonos analógicos, seis puertos para la conexión de teléfonos digitales y 10 canales VCM para comunicaciones IP en una. Es posible instalar hasta dos tarjetas en una unidad de control.
- **Tarjeta Combinada con 2 Troncales BRI.** Similar al anterior, posee dos puertos troncales BRI en lugar de los puertos analógicos.

1.3.2.3. Tarjetas Troncales

Estas tarjetas pueden ser instaladas sobre las tarjetas base IP500 Digital Station, IP500 Analog Phone y IP500 VCM para la provisión de puertos y servicios troncales.

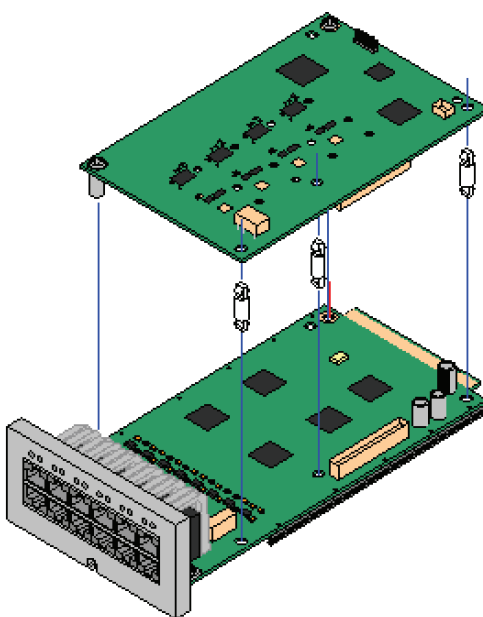


Figura 1.12 Ubicación de una Tarjeta Troncal en una Tarjeta Base ^[5]

Existen las siguientes tarjetas troncales:

- **Tarjeta Troncal IP500 Analógica.** Con un máximo de cuatro por unidad de control, provee o habilita 4 puertos troncales analógicos. Montada sobre una

tarjeta base IP500 Analog Phone 8 soporta fallas de alimentación de una extensión o enlace troncal.

- **Tarjeta Troncal IP500 BRI.** En variantes de 2 puertos (4 canales) y 4 puertos (8 canales), brinda hasta 4 conexiones troncales BRI.
- **Tarjeta Troncal Universal IP500 PRI.** La instalación de esta tarjeta brinda soporte a enlaces troncales digitales de tipo E1, T1 o E1R2, por lo que incorpora un CSU/DSU integrado. Está disponible en variantes individual (hasta 24 canales T1 o 30 canales E1) o dual (48 canales T1 o 60 canales E1), las cuales tienen habilitados por defecto 8 y 16 canales respectivamente para su funcionamiento. Es necesaria la adquisición de licencias para la habilitación de canales adicionales.

1.3.2.4. Módulos Externos de Expansión

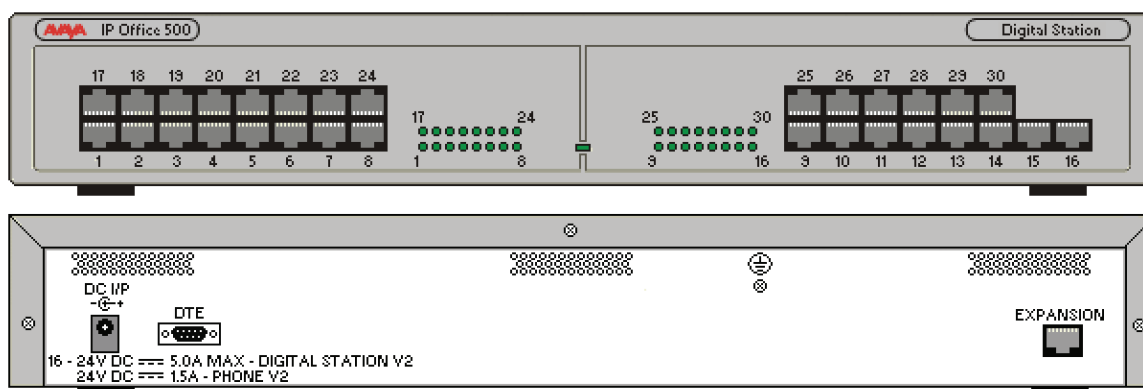


Figura 1.13 Vista Frontal y Posterior del Módulo de Expansión IP500 Digital Station ^[5]

Los módulos de expansión brindan conectividad adicional a troncales analógicas y a teléfonos analógicos y digitales, descritos a continuación:

- **Módulo de Expansión IP500 Phone.** Disponible en variantes de 16 y 30 extensiones; este módulo brinda conectividad a terminales localizadas hasta 1 Km. de distancia de la unidad de control. Proporciona funcionalidades de señalización DTMF, identificación de llamadas, entre otras.

- **Módulo de Expansión IP500 Digital Station.** Provee puertos adicionales para teléfonos digitales en variantes de 16 y 30 extensiones, con soporte de conectores RJ45 y RJ21.
- **Módulo de Expansión IP500 Analog Trunk 16.** Este módulo proporciona dieciséis troncales analógicas de dos hilos de inicio de bucle o inicio básico con tolerancia a fallas de alimentación para dos troncales.

1.3.2.5. Teléfonos



Figura 1.14 Terminales Soportados por IP Office ^[7]

- La central telefónica soporta los teléfonos IP de las series Avaya 1600, 5600, 9600, T3 (IP), teléfonos wireless de la serie 3600 y el *softphone* de vídeo IP Office.
- El equipo además soporta teléfonos digitales de las series Avaya 1400, 2400, 4400, 5400, 9500 y T3 (digital) a través de las tarjetas digitales y módulos de expansión necesarios. También brinda soporte a teléfonos Nortel de la serie T7000 y otros modelos mediante sus módulos de expansión especiales.

- Soporte de teléfonos Avaya ETR para la versión *Essential Edition* – PARTNER solamente.
- La plataforma soporta teléfonos, faxes y módems estándar con funcionalidades de identificación de llamadas e indicación de mensaje en espera.
- A través de los módulos apropiados se pueden añadir teléfonos Avaya DECT R4 3720 y 3725, además de tecnologías de movilidad de Nortel.
- Finalmente, IP Office integra un servidor SIP que le permite interconectar dispositivos finales como teléfonos de escritorio, softphones y altavoces de conferencia compatibles.

1.3.3. COMPUTER TELEPHONY INTEGRATION ^[5]

IP Office ofrece a sus clientes la posibilidad de crear aplicaciones CTI mediante la utilización de IP Office CTI Link, un producto de *middleware* CTI basado en estándares y su correspondiente Kit de Desarrollo de Software (SDK).

En el desarrollo e integración de sistemas y aplicaciones CTI debe tomarse en cuenta que es obligatorio contar con una Red de Área Local (LAN), por lo que serán necesarios equipos y sistemas informáticos externos a la plataforma de comunicaciones de telefonía que deben ser apropiadamente instalados y configurados para el correcto funcionamiento de la solución.

1.3.3.1. Herramientas e Interfaces CTI

La plataforma de comunicaciones IP Office soporta varias interfaces y tecnologías para el desarrollo de aplicaciones CTI, las cuales son descritas a continuación.

a) TAPILink Lite

Es una interfaz que soporta el desarrollo de aplicaciones CTI de tipo first-party mediante el uso del TAPI 2.1 ó TAPI 3.0 de Microsoft. TAPILink Lite ofrece características y funcionalidades básicas de control y monitoreo de los

dispositivos telefónicos, ajustándose perfectamente a ambientes que requieren de aplicaciones sencillas o de pocos recursos.

La principal ventaja del uso de esta herramienta es que se incluye en el CD-ROM de usuario de IP Office libre de cargo, lo que permite su utilización sin la adquisición de licencias.

b) TAPILink Pro

Al igual que la anterior utiliza las interfaces TAPI 2.1 y TAPI 3.0 de Microsoft, cuya principal diferencia es la gestión de varias líneas (extensiones) desde una misma aplicación (o servidor) en un modo *third-party*.

Adicionalmente, TAPILink Pro provee funcionalidades adicionales respecto a la versión gratuita, entre ellas:

- Habilita la posibilidad de monitorear y controlar grupos.
- Inicio y cierre de sesión de los agentes.
- Establecimiento y recuperación de desvío de destinos.
- Establecimiento y recuperación extendidos del estado de desvío (reenviar todas las llamadas, reenviar cuando está ocupado, reenviar cuando no contesta, control del estado no disponible del teléfono).
- Recuperación de configuraciones regionales de la extensión (como idioma).
- Establecer o limpiar el indicador de mensajes en espera.
- Habilitar o deshabilitar la pertenencia a grupos.
- Generación o detección de tonos y dígitos DTMF.

Para la creación de aplicaciones CTI que utilicen *TAPILink Pro* debe comprarse una llave de licencia de tipo *CTI Link Pro RFA* que puede ser adquirida de forma similar a cualquier otro producto o funcionalidad de IP Office.

c) TAPI-WAV Driver

Provee soporte para el procesamiento de voz basado en software, exclusivamente para el uso con el TAPI 2.1 de Windows, cuya funcionalidad sólo puede ser utilizada conjuntamente con *CTI Link Pro*. Para TAPI 3.0, IP Office soporta el Proveedor de Servicio Multimedia (MSP) que cumple similares propósitos.

Para el uso del *TAPI-WAV driver* es necesaria la adquisición de licencias *TAPI Wave User*, cada una de las cuales habilita 4 puertos para el procesamiento de voz.

d) DevLink Pro

Provee una interfaz de programación complementaria de las interfaces *TAPILink Lite* y *TAPILink Pro*. Provee flujos de eventos en tiempo real que contienen información de las actividades de los teléfonos y enlaces troncales. Para su utilización es necesaria la licencia *CTI Link Pro*.

e) IP Office Station Messaging Detail Record (SDMR)

Provee una interfaz para la obtención de eventos SMDR. Cuando una llamada es completada, se emite un registro separado por comas respecto al transcurso de la misma, cuya información puede ser dirigida a un puerto y dirección IP específicos. La interfaz IP Office SDMR se utiliza en el desarrollo de aplicaciones de contabilidad o tarificación de llamadas en las empresas.

f) Kit de Desarrollo de Software

Distribuido en un CD-ROM, contiene la documentación de las interfaces *TAPILink Lite*, *TAPILink Pro*, *DevLink Lite* y *DevLink Pro*.

También incluye programas pre-compilados basados en TAPI 2.1 y TAPI 3.0 junto a su código fuente, con el objeto de familiarizar a los desarrolladores con las interfaces antes mencionadas.

1.3.3.2. Licencias CTI

Considerando que para la creación de aplicaciones CTI es posible utilizar las versiones *TAPILink Lite*, *TAPILink Pro* y *TAPI-WAV Driver* de Avaya IP Office, éste apartado describe las dos últimas licencias que deben ser adquiridas para la provisión de funcionalidades avanzadas.

a) Licencia *CTI Link Pro*

Esta licencia permite al software TAPI de IP Office operar en modo *third-party*. Su número de parte es *IPO LIC CTI RFA LIC*, y se requiere de una licencia por cada servidor que requiera esta funcionalidad.

b) Licencia *WAV User*

Las aplicaciones que funcionan en modo *third-party* también pueden enviar flujos de datos WAV a los usuarios. Para ello es necesaria una licencia cuyo número de parte es *IPO LIC TAPI WAV RFA 4 LIC*, y permite el flujo de hasta 4 usuarios. La combinación de licencias es posible hasta un máximo soportado de 40 usuarios para la unidad de control IP Office 500 V2.

1.4. MICROSOFT FOUNDATION CLASSES^[8]

1.4.1. DEFINICIÓN

La librería de *Microsoft Foundation Classes* (MFC) es un “envoltorio” (del inglés *wrapper*) que encapsula el API de Windows en un esquema orientado a objetos. De forma más específica, la librería MFC es una plataforma para la programación de aplicaciones en Microsoft Windows la cual, escrita en C++, provee el código necesario para la administración de la interfaz gráfica de usuario (ventanas, menús, cuadros de diálogo, etc.), ejecutar operaciones básicas de entrada/salida, almacenamiento y acceso a colecciones de objetos de datos, entre otros.

Su principal objetivo es acortar el tiempo y aumentar la flexibilidad de desarrollo de aplicaciones, ofreciendo un acceso simplificado a tecnologías y elementos difíciles de programar como acceso a objetos de bases de datos (DAO), redes e internet,

portabilidad, múltiples hilos de ejecución, etc.; enfocando los esfuerzos de los desarrolladores en tareas más específicas y relacionadas con la lógica de la aplicación.

1.4.2. MODELO DE PROGRAMACIÓN DE WINDOWS

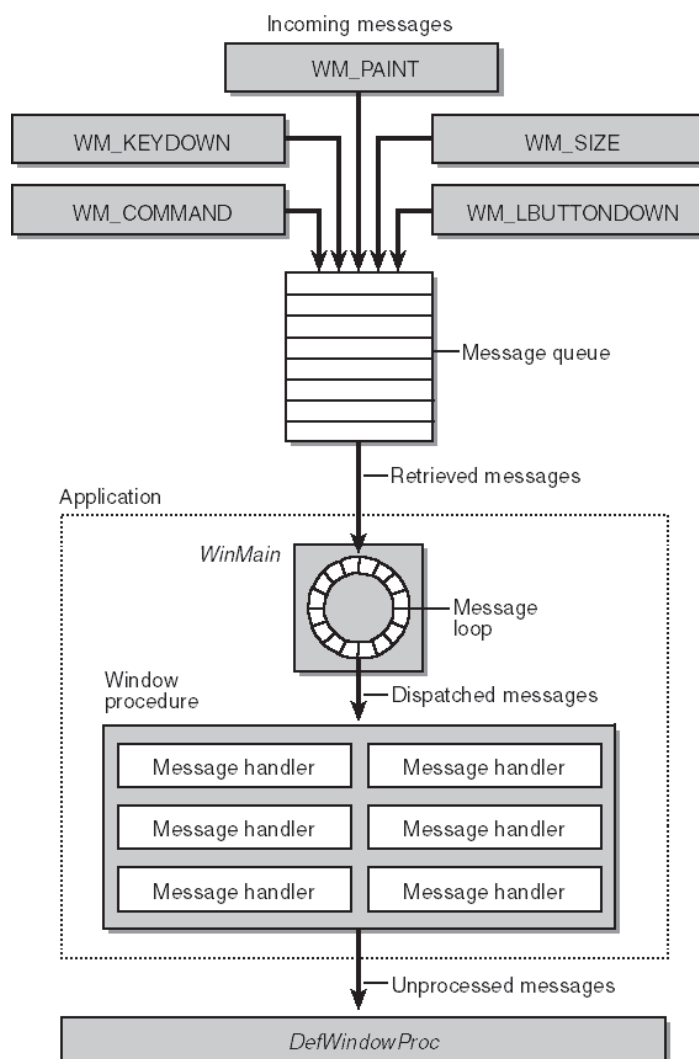


Figura 1.15 Modelo de Programación de Windows dirigido por Eventos ^[8]

A diferencia de las aplicaciones basadas en un paradigma de programación estructurado cuya ejecución se realiza de forma secuencial, en Windows los programas operan bajo un esquema de programación dirigido por eventos, los cuales

responden a sucesos mediante el procesamiento de mensajes enviados por el sistema operativo. La figura 1.15 conceptualiza el procesamiento de mensajes de acuerdo al modelo de programación dirigido por eventos.

La función *WinMain* es el punto de entrada para cualquier aplicación Windows, pero el procesamiento de todos los mensajes que el S.O. generará (como WM_CREATE para crear una ventana, WM_SIZE para asignar el tamaño de la misma, WM_MOVE para moverla, etc.) será realizado en otra función llamada *Windows procedure* (*WinProc* o *WndProc*), conocida también como el manipulador de eventos. En una aplicación que opera bajo este esquema, *WinMain* entra en un bucle de recuperación de mensajes que están almacenados en una cola hasta que son despachados a *WinProc*. Dicho lazo finaliza cuando se recupera el mensaje WM_QUIT (la ventana ha sido cerrada o el proceso ha sido finalizado), en cuyo caso *WinMain* retorna y la aplicación termina.

WinProc comúnmente utiliza otras funciones para el adecuado procesamiento de los mensajes, las cuales pueden ser locales (con respecto a la aplicación) o funciones API provistas por el sistema operativo. Las funciones API están encapsuladas en Bibliotecas de Enlace Dinámico (DLLs) que permiten realizar varias tareas como crear ventanas, dibujar líneas, manipular archivos de entrada y de salida, etc.

Mensajes

Un mensaje es manifestado en la forma de una llamada a una función *WinProc* que incluye entre los más relevantes los siguientes parámetros de entrada:

- ***hwnd***, manipulador de la ventana, es un campo único de identificación de 32 bits.
- ***message***, campo numérico que especifica el tipo de mensaje (los mensajes de Windows comienzan por WM_).
- ***wParam***, contiene información relacionada con el estado de las teclas *Ctrl*, *Shift* y los botones de ratón.

- *IParam*, almacena dos valores de 16 bits que identifican la ubicación del puntero del ratón cuando ocurre un clic.

1.4.3. ESTRUCTURA DE UNA APLICACIÓN MFC

En MFC todas las clases comienzan con la letra “C”. La mayoría ellas derivan directa o indirectamente de *CObject*. La clase *CWnd* es derivada en la clase *CFrameWnd*, que contiene la funcionalidad necesaria para la creación y control de la ventana principal. A su vez, en una aplicación MFC el objeto de ventana principal hereda el comportamiento de *CFrameWnd* y añade el suyo propio correspondiente a las funciones miembro definidas en la clase. Adicionalmente, todo programa MFC debe incluir una clase de aplicación, derivada de la clase *CWinApp*, que contiene la funcionalidad necesaria para establecer y ejecutar el bucle de procesamiento de mensajes (eventos).

Para la creación de un programa, todas las definiciones, clases y componentes principales de MFC están definidas en el archivo de cabecera *<afxwin.h>*. Los símbolos *<>* informan al compilador que éstas deben buscarse en los subdirectorios del compilador en lugar del directorio de trabajo. Toda aplicación MFC, por tanto, deberá incluir la referencia a este archivo de cabecera (y a todas las referencias externas también) para su adecuada generación y funcionamiento.

1.5. JAVA EE ^[9]

1.5.1. INTRODUCCIÓN

Diariamente, las empresas y organizaciones emplean herramientas informáticas que optimicen los procesos y faciliten las tareas. Estas aplicaciones empresariales son las encargadas de proporcionar la lógica de negocio que las instituciones necesitan. Los desarrolladores, por tanto, requieren herramientas y plataformas que permitan crear aplicaciones empresariales distribuidas, transaccionales y portables de una manera fácil, segura y confiable.

Java Platform, Enterprise Edition (Java EE), es una plataforma de programación de desarrollo y ejecución de aplicaciones empresariales en el lenguaje de programación Java. Basada en una arquitectura distribuida multicapa, está compuesta por varios conjuntos de componentes de software modulares que pueden ser ejecutados sobre un servidor de aplicaciones. Su principal objetivo es proporcionar a los desarrolladores un poderoso conjunto de APIs dirigidos a acortar el tiempo de creación de aplicaciones, reduciendo la complejidad de las mismas y con un menor grado de utilización de recursos.

El Proceso de la Comunidad de Java (JCP por sus siglas en inglés) es responsable de la definición de todas las tecnologías Java. Grupos de expertos de entidades interesadas crean las *Java Specification Requests* (JSRs) que definen varias de las tecnologías Java EE, con el objeto de asegurar compatibilidad entre plataformas y estandarizar la tecnología Java.

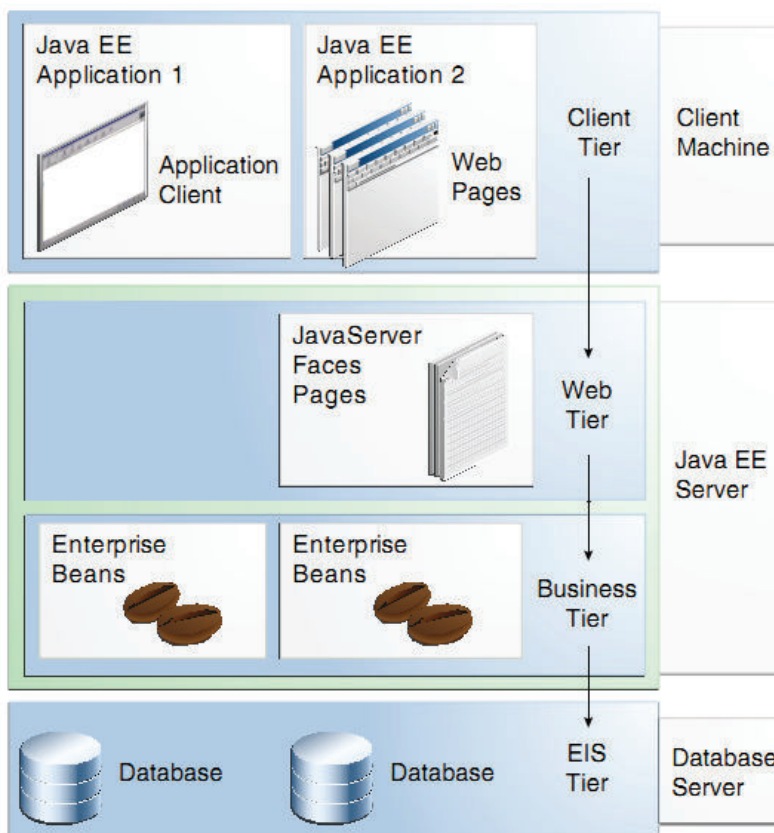


Figura 1.16 Aplicaciones Multinivel ^[9]

1.5.2. APLICACIONES DISTRIBUIDAS MULTINIVEL ^[9]

La plataforma Java EE utiliza un modelo de aplicaciones distribuido multinivel, en el cual los componentes que la conforman están divididos de acuerdo a su función, pudiendo ser instalados en una o varias máquinas dependiendo del grado de complejidad de la aplicación empresarial.

Aunque las aplicaciones Java EE pueden estar compuestas por tres o cuatro niveles, generalmente son consideradas como aplicaciones de tres capas debido a sus posibles localizaciones: máquinas de los clientes, equipo servidor de aplicaciones Java EE y máquinas de almacenamiento y bases de datos; como muestra la figura 1.16.

1.5.2.1. Componentes Java EE

Un componente Java EE puede definirse como una unidad funcional de software auto-contenida, que puede comunicarse con otros componentes y ser ensamblada dentro de una aplicación Java EE.

La especificación define los siguientes componentes.

Clientes Java EE

Los componentes que corren en el cliente pueden ser de dos tipos:

- **Clientes Web.** Son el conjunto de páginas web dinámicas (generadas en el servidor por componentes web) y navegadores web (que muestra las páginas recibidas del servidor). Dentro de esta categoría están incluidos los *applets*.
- **Aplicaciones Cliente.** Son aplicaciones que corren en las máquinas de los clientes que permiten a los usuarios manipular o controlar tareas a través de una poderosa interfaz gráfica. Las aplicaciones de este tipo pueden acceder directamente a los *beans* empresariales de la capa de negocios, aunque también pueden establecer conexiones con componentes de la capa web mediante conexiones HTTP. Adicionalmente, los clientes pueden estar escritos en otros lenguajes de programación o ser de tecnologías legadas y

aun así interactuar con los componentes y recursos de los servidores Java EE.

Componentes Web

Los componentes web pueden ser *servlets* o páginas web creadas mediante las tecnologías *JavaServer Faces* y/o *JavaServer Pages* (páginas JSP). Los *servlets* son clases escritas en lenguaje de programación Java que dinámicamente procesan peticiones y construyen respuestas. Las páginas JSP son documentos basados en texto que se ejecutan como *servlets* y que permiten un enfoque más natural en la creación de contenido estático. La tecnología *JavaServer Faces* (JSF) en cambio, está construida sobre las dos anteriores (*servlets* y JSP) y provee un *framework* de componentes de interfaz de usuario para el desarrollo de aplicaciones web.

Componentes de Negocio (*Business Components*)

La lógica encargada de solucionar o encontrar las necesidades de un dominio de negocios particular, como banca, finanzas o ventas, es conocida como código de negocio. Los componentes de negocio que encapsulan dicha lógica se ejecutan en el servidor y tienen por nombre *Enterprise JavaBeans* (conocidos como *beans* empresariales o EJBs) y pueden interactuar con otros niveles, capas, componentes o sistemas.

La principal ventaja de la utilización de estos componentes es la provisión centralizada de la lógica de negocio que puede ser utilizada por otros componentes EJB, web o aplicaciones cliente por igual.

Nivel de Sistema de Información Empresarial (*Enterprise Information System, EIS*)

En este nivel los sistemas de infraestructura empresarial como los sistemas de Planificación de Recursos Empresariales (ERPs), unidades de procesamiento de transacciones, bases de datos y otros sistemas de información legados son

accedidos por los componentes de aplicación Java EE para su adecuado procesamiento.

1.5.3. CONTENEDORES JAVA EE ^[9]

La construcción de aplicaciones distribuidas multinivel implica administrar y gestionar transacciones, estados, hilos múltiples, distribución de recursos y otros detalles complejos de bajo nivel que aumentan el nivel de dificultad y la propensión a errores de las mismas. La arquitectura Java EE, por tanto, provee servicios subyacentes en forma de contenedores para cada tipo de componente, liberando al desarrollador de las tareas antes mencionadas y concentrando sus esfuerzos en la lógica de negocio.

1.5.3.1. Servicios de Contenedor

Los contenedores son la interfaz entre un componente y la funcionalidad específica de la plataforma de bajo nivel que soporta dicho componente. Para que un componente web, empresarial o de tipo cliente pueda ser ejecutado primero debe ser ensamblado como un módulo Java EE y desplegado en su contenedor correspondiente.

Entre los principales servicios y características personalizables de los contenedores Java EE están: gestión de transacciones, seguridad, búsqueda de APIs mediante la Interfaz de Nombrado y Directorio Java (JNDI) y conectividad remota.

1.5.3.2. Tipos de Contenedores

En el despliegue de una aplicación Java EE los componentes son instalados en sus respectivos contenedores, que pueden ser:

- **Contenedor EJB.** Gestiona la ejecución de *beans* empresariales (o EJBs) ejecutados en el servidor Java EE.
- **Contenedor Web.** Administra la ejecución de páginas web, *servlets* y algunos componentes EJB. Estos componentes y su contenedor operan en el servidor Java EE.

- **Contenedor de Aplicaciones Cliente.** Funciona en el cliente, que gestiona la ejecución de componentes de aplicaciones cliente.
- **Contenedor de *Applets*.** Administra la ejecución de *applets*. Se conforma de un navegador web y un *plug-in* Java que son ejecutados en la máquina del cliente.

1.5.4. APIs Y TECNOLOGÍAS JAVA EE 6 & 7 ^[9]

Este apartado define brevemente las principales tecnologías e interfaces utilizadas en este proyecto de titulación, especificadas al momento, para la plataforma Java EE 6 (o su versión más reciente, 7).

1.5.4.1. Tecnología *Enterprise JavaBeans* (EJB)

El código compuesto de campos y métodos para la implementación de lógica de negocio y encapsulado en objetos Java simples (POJOs) es conocido como un componente JavaBeans Empresarial (EJB), o *enterprise bean*, que es ejecutado en el servidor Java EE y que puede ser utilizado sólo o en interacción con otros componentes similares.

Generalmente, los *enterprise beans* son utilizados en los siguientes escenarios:

- Cuando la aplicación debe ser escalable con el objeto de soportar un número creciente de usuarios.
- Las transacciones u operaciones deben asegurar la integridad de los datos.
- La aplicación tiene varios tipos de clientes.

Entre las ventajas en la utilización de *enterprise beans* está la simplificación del desarrollo de grandes aplicaciones distribuidas. De esta manera, el desarrollador puede concentrarse en resolver los problemas de negocios, mientras que el contenedor EJB es responsable de servicios como la gestión de transacciones, seguridad o consumo de recursos.

Por tal motivo, y dado que la lógica de negocios se encuentra centralizada, el desarrollador de la aplicación cliente puede enfocarse en la presentación, aligerando

y simplificando la creación de este tipo de sistemas; beneficio importante para clientes que operan en dispositivos móviles, por ejemplo.

Adicionalmente, la portabilidad de estos componentes permite la creación de nuevas aplicaciones a partir de *beans* existentes siempre y cuando operen sobre servidores compatibles con Java EE.

1.5.4.1.1. Tipos de Enterprise Beans

Los *enterprise beans* pueden ser:

- **Beans de Sesión (Session Beans)**. Representan una conversación transitoria con el cliente. Estos componentes encapsulan la lógica de negocios que puede ser invocada por los clientes sobre los métodos definidos a través de vistas locales, remotas o servicios web. Pueden ser a su vez de tres tipos:
 - *Beans de Sesión con Estado (Stateful Session Beans)*, representan el estado de una única sesión cliente/*bean*, lo que implica que el componente no es compartido y sólo puede estar relacionado con un cliente particular.
 - *Beans de Sesión sin Estado (Stateless Session Beans)*, que al no mantener un estado conversacional pueden ser compartidos por varios clientes. Son los únicos que pueden ser publicados como servicios web.
 - *Beans de Sesión Singleton (Singleton Session Beans)*, son instanciados sólo una vez en todo el ciclo de vida de la aplicación y compartidos por todos los clientes de la misma.
- **Beans Dirigidos por Mensajes (Message-Driven Beans)**. Son componentes de tipo *listener* (reciben eventos generados por otros componentes) que permiten la recepción de mensajes de forma asíncrona. Comúnmente operan con un sistema de mensajería como JMS (*Java Message Service*).

1.5.4.2. Tecnología *JavaServer Faces* (JSF)

Provee un marco de trabajo (*framework*) para el desarrollo simplificado de interfaces de usuario en aplicaciones web.

Basado en el patrón MVC (Modelo Vista Controlador), JSF separa los datos de una aplicación en tres capas específicas: datos (modelo), interfaz de usuario (vista) y lógica de negocio (controlador), ajustándose perfectamente con otras tecnologías (como EJB) y haciéndole propicio para el desarrollo de aplicaciones web de manera sencilla y organizada.

Esta tecnología provee un API para la representación de componentes GUI y gestión de estados, manipulación de eventos, validación y conversión de datos, configuración del esquema de navegación entre páginas, soporte para accesibilidad e internacionalización y extensibilidad de todas las características mencionadas. Adicionalmente, proporciona una biblioteca de etiquetas para la adición de componentes en páginas web y la conexión de éstos con objetos que se encuentran de lado del servidor.

Dentro de esta especificación se define *facelets* como el lenguaje declarativo usado para la creación de vistas JSF mediante la utilización de plantillas XHTML y la construcción de árboles de componentes.

1.5.4.3. *Java Persistence API* (JPA)

Es una API de la plataforma Java basada en estándares para la gestión de persistencia. La persistencia de objetos está relacionada con el almacenamiento y recuperación de los valores de los atributos de uno o más objetos cuyas fuentes pueden ser tablas, archivos, entre otros. JPA, por tanto, se encarga del mapeo de modelos orientados a objetos (POJOs) con bases de datos relacionales, denominado ORM (*Object Relational Mapping* – Mapeo Relacional de Objetos), pudiendo ser utilizada por aplicaciones Java SE (escritorio) o Java EE (empresariales).

La persistencia de Java se enfoca en las siguientes áreas:

- El API de Persistencia de Java en sí mismo, definido en el paquete *javax.persistence*.
- *Java Persistence Query Language* (JPQL), que es el lenguaje orientado a objetos utilizado para la generación de sentencias de consulta o acceso a la información persistida.
- La metadata de mapeo objeto/relación que puede ser configurada mediante archivos XML o anotaciones. Las anotaciones proveen datos sobre un programa que no son parte de éste en sí mismo. Al no afectar el funcionamiento del código que anotan, son utilizadas para detectar errores, suprimir advertencias, generar código adicional o ser procesadas en tiempo de ejecución.

1.5.4.4. *Java Database Connectivity API (JDBC)*

Es una API que permite la invocación de comandos SQL desde el lenguaje de programación Java. JDBC, al ser diseñado para ser independiente de la plataforma, proporciona acceso a una gran variedad de bases de datos a través de una interfaz de programación común.

El API se compone de dos partes:

- Una interfaz de nivel aplicación utilizada por los componentes para el acceso a la base de datos.
- Una interfaz del proveedor de servicios que vincula el controlador JDBC con plataforma Java EE.

1.5.4.5. *Java Naming and Directory Interface (JNDI)*

La Interfaz de Nombrado y Directorio Java es una API para servicios de directorio que permite el descubrimiento y búsqueda de objetos por su nombre. Al ser independiente de la implementación subyacente, es posible acceder a múltiples servicios de directorio y nombrado como LDAP, DNS, NIS y NDS.

JNDI define una interfaz de proveedor de servicio (*Service Provider Interface* - SPI) que habilita la integración con el *framework* y el acceso a una gran variedad de servicios de nombres y directorios, de forma que puedan ser invocados de forma transparente por parte de los clientes a través de un esquema común.

Un componente Java EE puede acceder a objetos nombrados provistos por el sistema o definidos por el usuario. Éstos pueden ser *Enterprise beans*, objetos de fuente de datos JDBC u objetos de conexiones de mensajes, los cuales deben ser nombrados en un subcontexto que forme parte del sistema de nombrado de acuerdo a su tipo. Por ejemplo, referencias JDBC son nombradas en el subcontexto `java:comp/env/jdbc`.

1.5.5. SERVIDORES DE APLICACIONES

Un servidor de aplicaciones provee un entorno en el que es posible ejecutar aplicaciones multinivel distribuidas. Generalmente está compuesto por conjuntos de componentes accesibles a través de APIs especificados por la plataforma a la que corresponde, con características y funcionalidades avanzadas como son tolerancia a fallos, seguridad, balanceo de carga, escalabilidad e integración con otras plataformas de software, las cuales facilitan las tareas de los desarrolladores y les permiten enfocarse en la lógica de negocio de sus aplicaciones sin preocuparse por tareas de bajo nivel.

De forma habitual este término hace referencia a un servidor de aplicaciones Java EE, en cuyo caso es posible encontrar una gran variedad de alternativas privadas y de código abierto, entre las más importantes:

- *GlassFish server Open Source Edition.*
- *Oracle GlassFish server.*
- *Oracle WebLogic Application Server.*
- *IBM WebSphere Application Server.*
- Apache Geronimo.
- JBoss Application Server, JOnAS, entre otros.

1.6. LENGUAJE UNIFICADO DE MODELADO ^[10]

1.6.1. INTRODUCCIÓN

El Lenguaje Unificado de Modelado, o UML por las siglas en inglés de *Unified Modeling Language*, es un lenguaje gráfico de visualización, especificación, construcción y documentación de sistemas de software ^[E-7].

Así como un automóvil requiere de un plano de diseño donde se detalle cada uno de los elementos que lo integran, el UML pretende definir los componentes, recursos u objetos que conforman un sistema de software y las interacciones que existen entre éstos. Su principal propósito es organizar el proceso de diseño, facilitar el desarrollo de la aplicación y orientar a clientes, analistas y programadores sobre el funcionamiento del sistema o programa desarrollado.

1.6.2. DIAGRAMAS DEL UML

“El UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas.”^[10]

La finalidad de los diagramas UML es detallar las diversas facetas que conforman un sistema, las que en conjunto conforman un modelo. Debe tomarse en cuenta que el modelo describe únicamente el funcionamiento del sistema, mas no su implementación.

A continuación se describirán brevemente los diagramas UML más comunes, sus características y reglas necesarias para su elaboración.

Diagrama de Clases

Entendiéndose una clase como un modelo que agrupa las acciones y atributos comunes de un grupo de cosas, un diagrama de clases tiene por objeto describir las clases y atributos que componen un sistema, así como las relaciones que existen entre ellas.

Una clase generalmente es representada como un rectángulo dividido en tres secciones. En la parte superior se incluye el nombre, la parte central especifica los atributos y la parte inferior contiene las funciones (acciones).

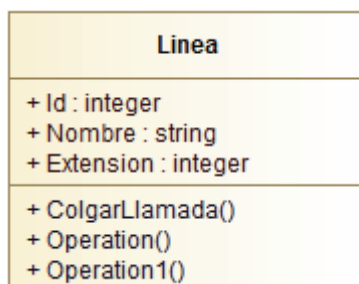


Figura 1.17 Ejemplo de Notación UML de una Clase

Diagrama de Casos de Uso

Los diagramas de casos de uso tienen por finalidad describir las interacciones de los usuarios con el sistema. Su principal objetivo es determinar los requisitos del sistema desde el punto de vista de los usuarios.

Estos diagramas se componen de actores (usuarios) y casos de uso (acciones) como se puede ver en el ejemplo de la figura 1.18.

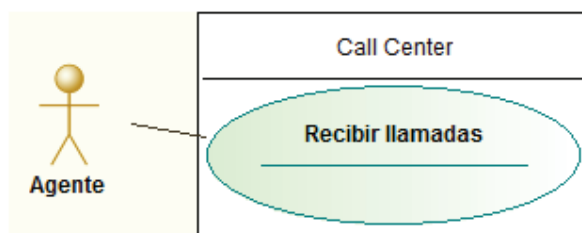


Figura 1.18 Ejemplo de Representación de un Caso de Uso

Diagrama de Estados

Estos diagramas muestran los distintos estados y transiciones por las que pasan los objetos de un sistema con el propósito de describir su comportamiento.

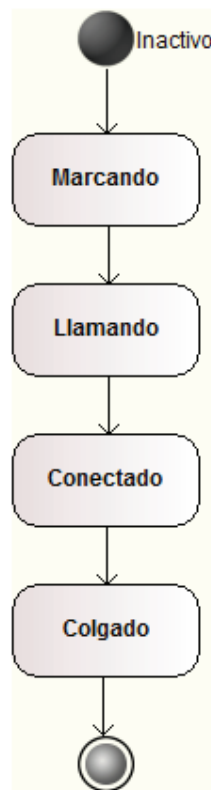


Figura 1.19 Ejemplo de Representación de Estados de una Llamada Telefónica

Diagrama de Comunicaciones

Un diagrama de comunicaciones modela las interacciones entre los objetos de un sistema en términos de mensajes secuenciados. Se diferencia del diagrama de secuencias en la representación espacial (en lugar de temporal) de las interacciones.

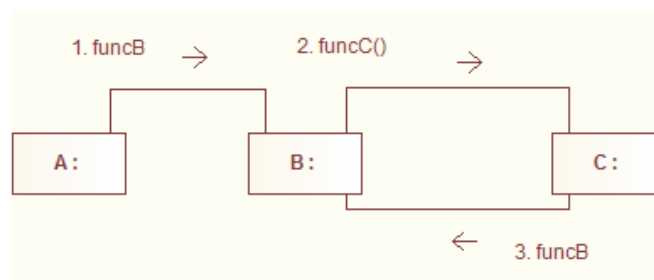


Figura 1.20 Ejemplo de Representación de un Diagrama de Comunicaciones

Diagrama de Secuencias

Los diagramas de secuencias tienen por objeto mostrar el orden y modo de interacción entre los componentes de un sistema (ver figura 1.21).

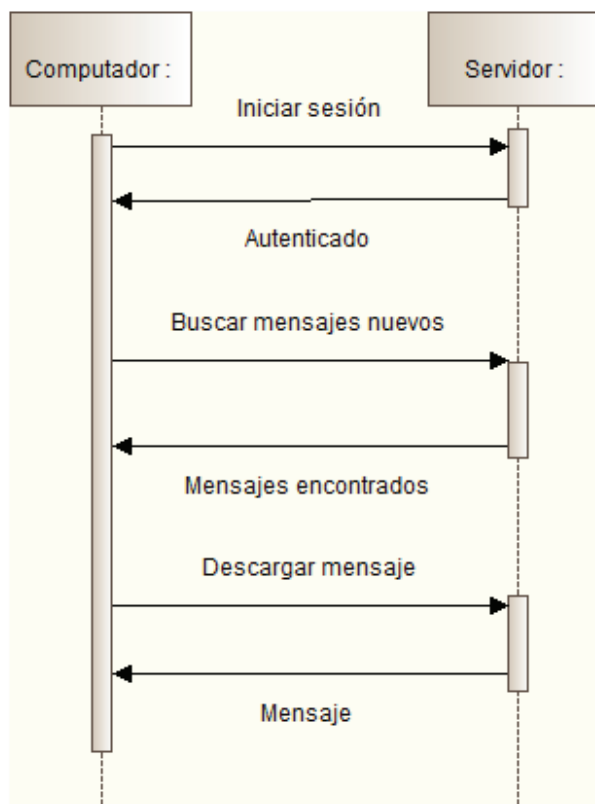


Figura 1.21 Ejemplo de Representación de la Interacción entre un Cliente de Correo Electrónico y el Servidor

Diagrama de Distribución

Su propósito es mostrar la arquitectura física del sistema informático. En algunos casos se combina con el diagrama de componentes para representar la agrupación y despliegue de los nodos que conforman la aplicación.

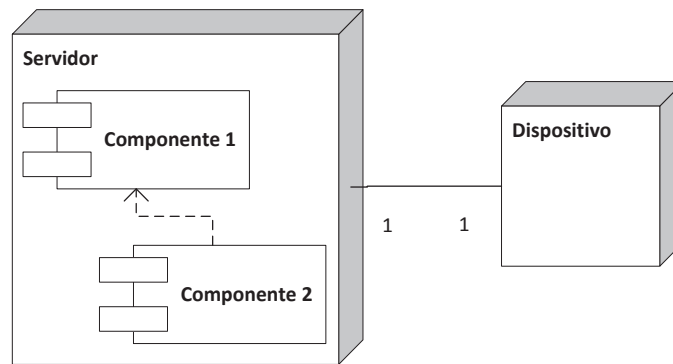


Figura 1.22 Representación de un Diagrama de Distribución

Diagrama de Actividades

Estos diagramas pretenden describir el algoritmo, procedimiento o flujo de trabajo de un sistema (figura 1.23).

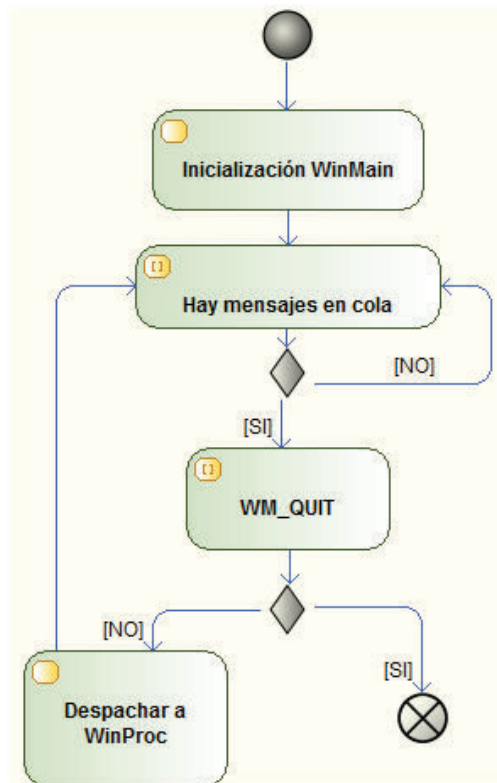


Figura 1.23 Diagrama de Actividades del Procesamiento de Eventos en Aplicaciones MFC Windows ^[8]

CAPÍTULO II

2. ANÁLISIS DE REQUERIMIENTOS Y DISEÑO

2.1. PLANTEAMIENTO DEL PROBLEMA

De forma general, los *Call Centers* tienen como propósito interactuar con los clientes a través de llamadas telefónicas entrantes o salientes para atender sus requerimientos o prestar algún servicio particular. Bajo este concepto el flujo de información entre los agentes y los clientes es urgente y variable, que en algunos casos debe ser almacenado con el objeto de generar estadísticas cuyo análisis permita mejorar los procesos de las organizaciones.

Los sistemas tradicionales de telefonía pura y las aplicaciones institucionales dentro de las empresas, sin embargo, operan de forma separada. Esto significa que en la realización o recepción de las llamadas telefónicas no es posible obtener de forma inmediata y automatizada información relevante de los contactos involucrados en la comunicación, lo que repercute en mayores tiempos de respuesta por parte de los agentes y la utilización de dos (o posiblemente más) aplicaciones que ofrezcan los datos necesarios para brindar un adecuada atención a los clientes.

Dado que la plataforma de comunicaciones Avaya IP Office no es la excepción, y que las soluciones propietarias disponibles en el mercado no ofrecen la personalización ni la flexibilidad de ajuste al entorno ecuatoriano, es necesaria una herramienta informática de tipo empresarial que permita resolver las necesidades anteriormente definidas, con la posibilidad de extenderse a otras plataformas de telefonía similares y a costos asequibles conforme a la realidad nacional.

2.2. METODOLOGÍA DE DESARROLLO DE SOFTWARE ^[11]

2.2.1. INTRODUCCIÓN

La creación de sistemas informáticos complejos, y en este caso particular, de tipo empresarial, requiere del seguimiento de un marco de trabajo organizacional que

estructure, planifique y controle cada una de las etapas (entiéndase tareas) involucradas en su desarrollo. La principal ventaja de la utilización de este tipo de esquemas está en la creación sistematizada de aplicaciones de una forma inteligente y organizada, detalladamente documentada y que cumpla adecuadamente los requerimientos establecidos. En palabras de Franklin Roosevelt: *“Existen muchas formas de ir hacia adelante, pero sólo una de permanecer”*.

La utilización de una metodología en la creación de un sistema no implica el estricto seguimiento de todos y cada uno de los lineamientos que la constituyen. Más bien, representa una guía que comprende las mejores prácticas para la adecuada elaboración de un programa informático. En lo que respecta a este documento, se tomarán en cuenta las técnicas y sugerencias de la metodología elegida en el ámbito que éstas sean de utilidad para su exitosa realización, con el propósito de no exceder el alcance original de este proyecto.

2.2.2. SELECCIÓN DE LA METODOLOGÍA

En la actualidad es posible encontrar una gran variedad de metodologías, algunas tradicionales, enfocadas en la resolución sistematizada y detallada de requerimientos, y otras recientes, encaminadas a crear software de forma rápida y flexible. En la tabla 2.1 se realiza una rápida comparación de las principales alternativas vigentes con el propósito de escoger la opción más favorable.

La metodología de Desarrollo Rápido de Aplicaciones se caracteriza por su enorme rapidez y flexibilidad en el proceso de creación del producto. Es especialmente efectiva en equipos de desarrollo que cuentan con profesionales experimentados que potencian la creación de sistemas en un corto tiempo. Entre las principales variantes de esta metodología están: Programación Extrema (XP), SCRUM, Método de Desarrollo de Sistemas Dinámico (DSDM), Desarrollo Ágil, entre otros.

Un enfoque más tradicional es ofrecido por la metodología de Proceso Unificado, cuyo proceso de desarrollo iterativo e incremental permite crear aplicaciones estables y altamente escalables, especialmente orientada a la creación de sistemas

complejos o de largo plazo. En cada iteración el software es modificado, pulido y corregido según las necesidades y requisitos planteados a cambio del tiempo de culminación del proyecto. La enorme flexibilidad de adaptación en las organizaciones estimuló la creación de variantes y refinamientos de esta metodología, entre las más relevantes: Proceso Unificado de Rational (RUP), Proceso Unificado Ágil (AUP), Proceso Unificado Básico (BUP), etc.

Metodología	Fases, Etapas	Características
Desarrollo Rápido de Aplicaciones (RAD)	<ul style="list-style-type: none"> • Planificación de Requerimientos • Diseño de Usuario • Construcción • Corte 	<ul style="list-style-type: none"> • Metodología de desarrollo ágil. • Mayor flexibilidad que otros modelos. • Participación activa del cliente en todas las etapas del producto. • Menor énfasis en la arquitectura del software. • Escalabilidad reducida.
Proceso Unificado (PU)	<ul style="list-style-type: none"> • Inicio • Elaboración • Construcción • Transición 	<ul style="list-style-type: none"> • Metodología tradicional de desarrollo. • Iterativo e Incremental. • Dirigido por Casos de Uso. • Centrado en la Arquitectura. • Adaptable a cualquier escenario.
ICONIX	<ul style="list-style-type: none"> • Análisis de Requerimientos • Análisis y Diseño Preliminar • Diseño Detallado • Despliegue 	<ul style="list-style-type: none"> • Fuertemente basado en UML • Dirigido por casos de uso. • Más ligero que los paradigmas basados en proceso unificado. • Paradigma de desarrollo ágil.

Tabla 2.1 Principales Metodologías de Desarrollo de Software

Una metodología menos conocida pero no menos notable, ICONIX, combina algunas de las características de las dos anteriores, en las que se prioriza la recolección y

procesamiento de los datos, que son descompuestos en fronteras para la elaboración del modelo de dominio del sistema. Una de las principales desventajas de esta metodología radica en la utilización de únicamente cuatro tipos de diagramas UML, los cuales pueden ser insuficientes para representar a detalle el diseño o funcionamiento de aplicaciones complejas o distribuidas.

Como resultado de la comparación de las tres principales metodologías antes mencionadas, aquella elegida para la creación de este proyecto de titulación corresponde al **Proceso Unificado**, que proporciona una mejor organización y detalle en la creación de aplicaciones informáticas respecto a sus competidoras.

2.2.3. VISIÓN GENERAL DE PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE ^{[11][12]}

2.2.3.1. Introducción y Características

La metodología de Proceso Unificado (PU) no sólo es considerada como un proceso de software, sino como un *framework* flexible que puede ser ajustado según las necesidades de los proyectos o las instituciones. Los creadores del Lenguaje Unificado de Modelado (UML), Ivar Jacobson, Grady Booch y James Rumbaugh en el año 1999 describieron por primera vez esta metodología en el libro “The Unified Software Development Process”, refiriéndose específicamente al Proceso Unificado de Rational (RUP) de IBM, un refinamiento del PU.

El Proceso Unificado se basa fuertemente en el uso de UML, que proporciona un gran nivel de detalle respecto al funcionamiento y la composición de los proyectos de software. Entre las principales características de esta metodología se pueden describir:

- **Iterativa e Incremental.** Cada una de las fases del PU está compuesta por iteraciones que como resultado incluyen mejoras (incrementos) en el producto en desarrollo. De forma general, cada una de las iteraciones está debidamente coordinada y calendarizada, de forma que la efectividad del proceso se maximiza.

- **Dirigida por Casos de Uso.** Los casos de uso son claves en la captura y refinamiento de los requerimientos en cada una de las iteraciones del proyecto. Una de las principales ventajas de la utilización de los casos de uso es la determinación temprana de las necesidades o servicios requeridos por la solución, con lo cual los esfuerzos y recursos del proyecto son enfocados de forma más efectiva.
- **Centrado en la Arquitectura.** Entendiéndose a la arquitectura como el conjunto de atributos estáticos y dinámicos más importantes de la solución, conformada por plataformas de software y hardware, componentes reutilizables, consideraciones de instalación y despliegue, entre otros. También forman parte de ésta los casos de uso recogidos y analizados, así como el esqueleto que determina el funcionamiento de la solución. En el diseño de un proyecto informático, la especificación de la arquitectura es importante puesto que ofrece un marco general de desarrollo y funcionamiento que puede ser interpretado por desarrolladores, directores y clientes.
- **Enfocada en los Riesgos.** En cada una de las fases y sus correspondientes etapas, esta metodología realiza un análisis exhaustivo de los requerimientos, recursos y componentes involucrados, lo que permite determinar y disminuir de forma anticipada los potenciales riesgos implicados en el desarrollo del sistema o simplemente descartar ciertos aspectos innecesarios de la solución.

2.2.3.2. Fases del Proceso Unificado

El Proceso Unificado se compone de cuatro fases: Inicio, Elaboración, Construcción y Transición, cada una de las cuales a su vez está compuesta por disciplinas o procesos que especializan las tareas involucradas en la creación del proyecto.

a) Fase de Inicio

Generalmente la más corta, su principal objetivo es modelar los procesos de negocio de las organizaciones y determinar los principales requisitos del sistema, identificando una arquitectura candidata o los lineamientos generales

de la solución. Además, en esta fase se realiza un análisis e identificación de riesgos, así como una estimación aproximada de tiempo y costos del proyecto.

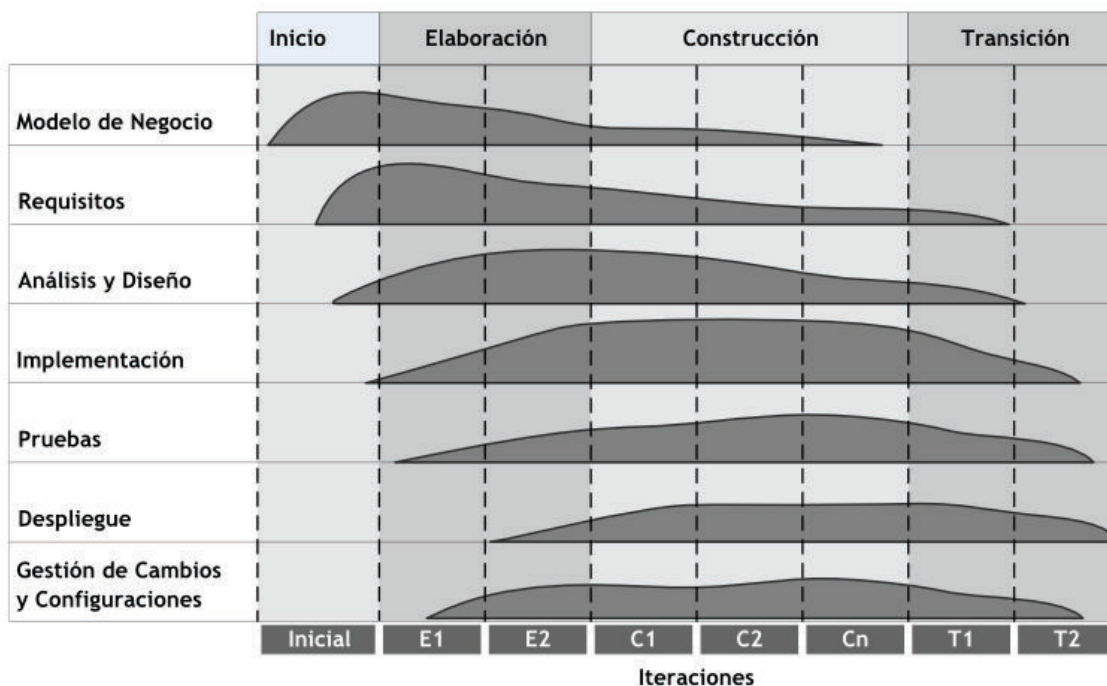


Figura 2.1 Fases y Disciplinas del Proceso Unificado ^[E-8]

b) Fase de Elaboración

En esta fase se realiza un profundo análisis de los requisitos y componentes del sistema, así como de su arquitectura, apoyándose principalmente en diagramas de caso de uso, diagramas de clase y definición de paquetes. Las principales metas de esta fase son, por tanto, el tratamiento de los factores de riesgo claves y la validación y diseño de la arquitectura del sistema.

c) Fase de Construcción

Generalmente es la etapa de desarrollo que lleva más tiempo, basada directamente en los lineamientos definidos en la fase de elaboración. Cada una de las características del sistema es implementada en una serie de iteraciones, al final de las cuales se obtiene una nueva versión ejecutable del

producto. En esta fase comúnmente son utilizados los diagramas de secuencia, actividades, colaboración, estados, etc.

d) Fase de Transición

Esta fase tiene por objeto desplegar la solución en el entorno de los usuarios, cuya retroalimentación favorece el refinamiento y la corrección de problemas de funcionamiento o estabilidad del sistema. Adicionalmente, en esta fase se realiza la capacitación de los usuarios y la creación del material de despliegue, administración y uso de la solución.

2.3. FASE DE INICIO (I)⁶

2.3.1. INTRODUCCIÓN^[12]

De forma general los sistemas informáticos son creados para prestar algún servicio o cubrir alguna necesidad particular. Para ello el primer paso será determinar los requerimientos funcionales, tecnológicos, organizacionales y financieros (de ser el caso) que no sólo justifiquen la realización del proyecto, sino además proporcionen la información necesaria para diseñar y elaborar la solución propuesta.

Esta fase, por tanto, tiene por objeto recabar y estudiar los requisitos anteriormente señalados, cuyo análisis contribuirá en la determinación del ámbito del sistema, la determinación de riesgos y la especificación de un diseño general que serán utilizados en las subsiguientes fases.

Entradas de Esta Fase

Dado que este proyecto está enfocado en la integración con la plataforma de comunicaciones Avaya IP Office 500 V2, la primera entrada de esta fase está constituida por toda la documentación del fabricante correspondiente y una adecuada experiencia en el manejo y configuración de la misma.

⁶ La letra "I" mayúscula tiene por objeto agrupar y representar a los distintos subelementos de la fase de Inicio que posteriormente serán referenciados en otras fases. Así, por ejemplo, los Requisitos No Funcionales especificados en esta fase serán definidos como IR-NF.

En tal virtud, es notable señalar que se dispone de un completo conjunto de información que comprende la instalación, configuración y administración de la central telefónica referida. Además se dispone de una cuenta en Avaya Devconnect⁷, un portal en el que es posible obtener información de los productos, SDKs y aplicaciones de ejemplo de los principales productos de la marca Avaya.

Adicionalmente, con base en la experiencia laboral adquirida en el sector de comunicaciones unificadas y *contact centers*, se tiene una clara perspectiva de los requisitos y las necesidades de los usuarios respecto al sistema que se pretende crear, así como también de los principales alcances y limitaciones de otras soluciones disponibles en el mercado.

2.3.2. MODELAMIENTO DE LOS PROCESOS DE NEGOCIO Y CASOS DE USO

La comprensión del funcionamiento de los procesos organizacionales de las instituciones influye enormemente en la creación de las soluciones informáticas. Es por ello que en el ámbito de este proyecto se definirá al proceso de negocio como una colección de tareas o actividades concretas que producen un servicio o producto específico para los clientes (usuarios).

En la operación, control y administración de un *call center* pueden identificarse tres roles principales (agente, supervisor y administrador) ^[14] cuyos modelos de negocio y casos de uso serán descritos posteriormente y estarán circunscritos a las funciones y responsabilidades de interés para la creación del sistema propuesto.

2.3.2.1. Agente

En el libro “*The Call Center Dictionary*” se define un agente como “*un término general para alguien que manipula (gestiona) llamadas telefónicas en un call center. Otros nombres comunes para el mismo oficio incluyen, aunque no están limitados a: operador(a), asistente, representante, representante de servicio al cliente, CSR, representante de soporte al cliente, representante de soporte técnico, vendedor por*

⁷ <http://www.devconnectprogram.com/>

teléfono, representante de ventas por teléfono, tele-vendedor...” [13] Es importante tomar en cuenta que esta definición de agente está circunscrita a la función que desempeña una persona en un *call center*.

2.3.2.1.1. Funciones y Responsabilidades

Las principales funciones y responsabilidades de los agentes de un *call center* pueden resumirse en el diagrama de caso de uso de la figura 2.2.

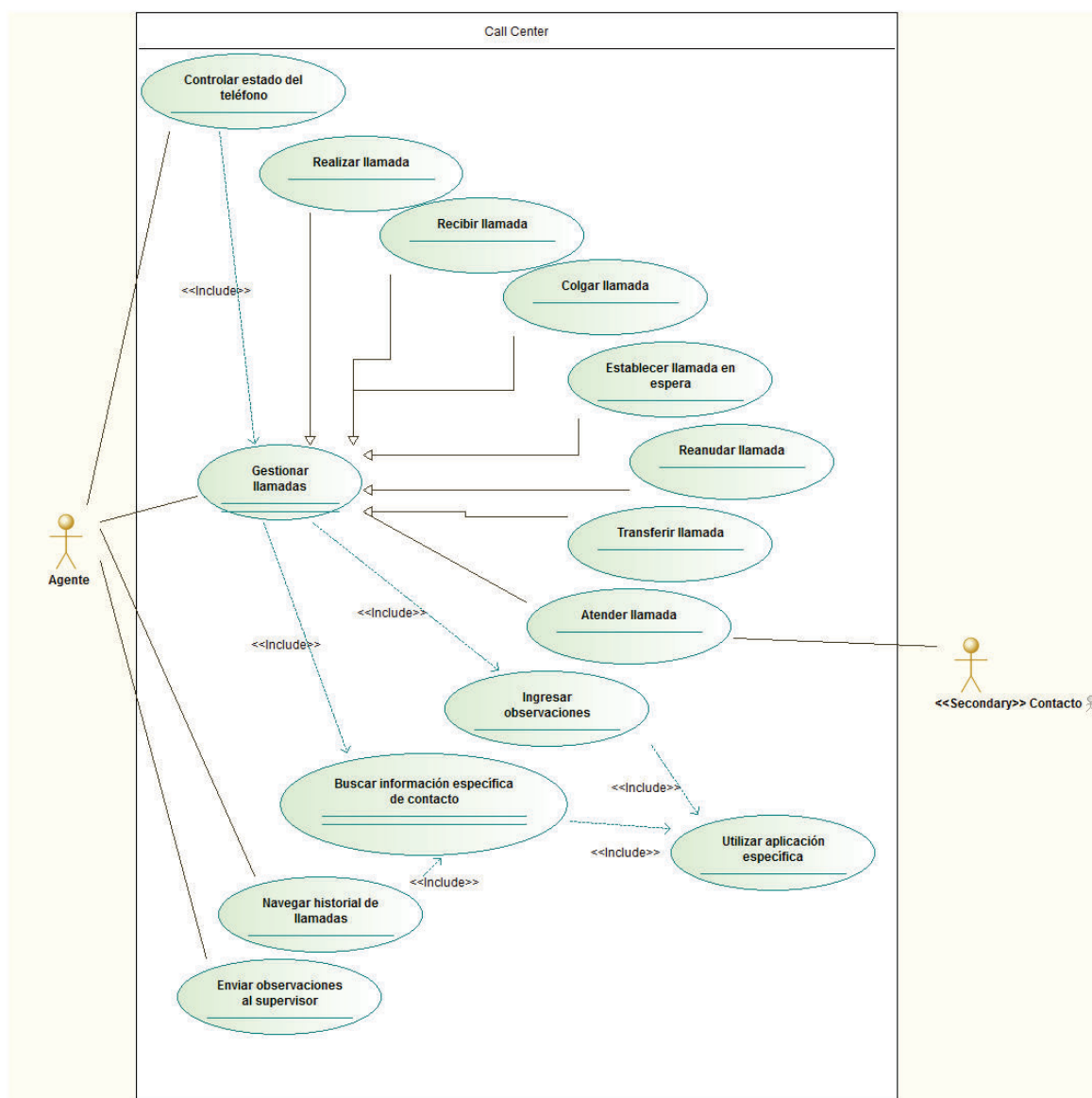


Figura 2.2 Caso de Uso - Agente

De lo anterior se puede determinar que los agentes son encargados de:

- Gestionar y controlar el estado del teléfono.
- Atender las llamadas (entrantes o salientes) e interactuar con los contactos (clientes principalmente):
 - Buscar información de importancia en el transcurso de la llamada en curso mediante la utilización de aplicaciones o sistemas empresariales que almacenan y gestionan la información de los contactos (clientes), como por ejemplo: ERPs, CRMs, etc.
 - Registrar anotaciones, observaciones e información de relevancia respecto al desarrollo de la llamada. Esta tarea puede realizarse de forma manual o mediante la utilización de otras herramientas informáticas. Estas observaciones serán analizadas posteriormente por los supervisores en función de las necesidades de las empresas.
 - De forma implícita, el agente se encarga de repasar las llamadas recibidas en casos especiales, como el caso de la comprobación de llamadas perdidas o de emergencia mediante la utilización de un historial de llamadas.

2.3.2.1.2. Procesos de Negocio (Gestión de Llamadas Telefónicas)

A continuación se mostrarán los modelos de proceso de negocio involucrados en la gestión de llamadas telefónicas de un *call center* por parte de los agentes. Dichos modelos están presentados en forma de diagramas de actividades y desde una perspectiva puramente operacional, cuyo propósito es abstraer las principales tareas involucradas en el procesamiento de llamadas de la forma más simple posible.

- **Llamadas Entrantes**

Para el caso de las llamadas telefónicas entrantes (*inbound*), como puede apreciarse en la figura 2.3, el contacto es el iniciador de la comunicación. La llamada es asignada a una cola que posteriormente será redirigida al primer agente disponible. Una vez contestada, el agente dará el saludo respectivo y preguntará al contacto por el motivo de la llamada (interacción inicial). Si para

atender la llamada el agente requiere de información relevante del contacto, será necesario abrir un aplicativo externo que ofrezca los datos suficientes. Adicionalmente, es posible que el agente tenga que anotar las observaciones desprendidas de la interacción con el contacto, por lo que utilizará una aplicación específica (como un editor de texto, por ejemplo) o escribirá los datos de forma manual, que después serán analizados y procesados por el supervisor del *call center*.

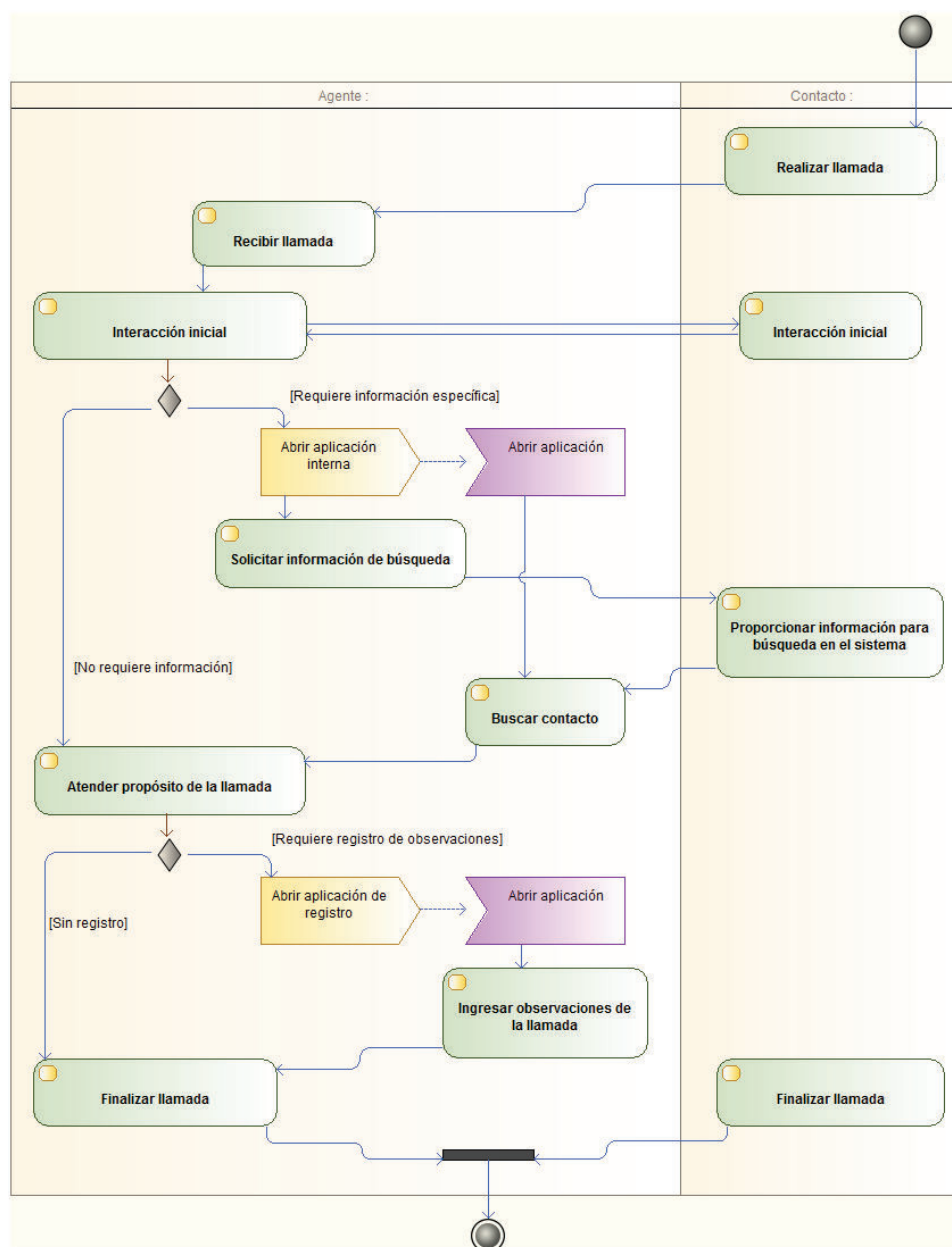


Figura 2.3 Proceso de Negocio de Agentes para la Recepción de Llamadas.

- **Llamadas Salientes**

En la figura 2.4 se presenta el diagrama que representa el proceso de negocio relacionado a las llamadas salientes en un *call center*.

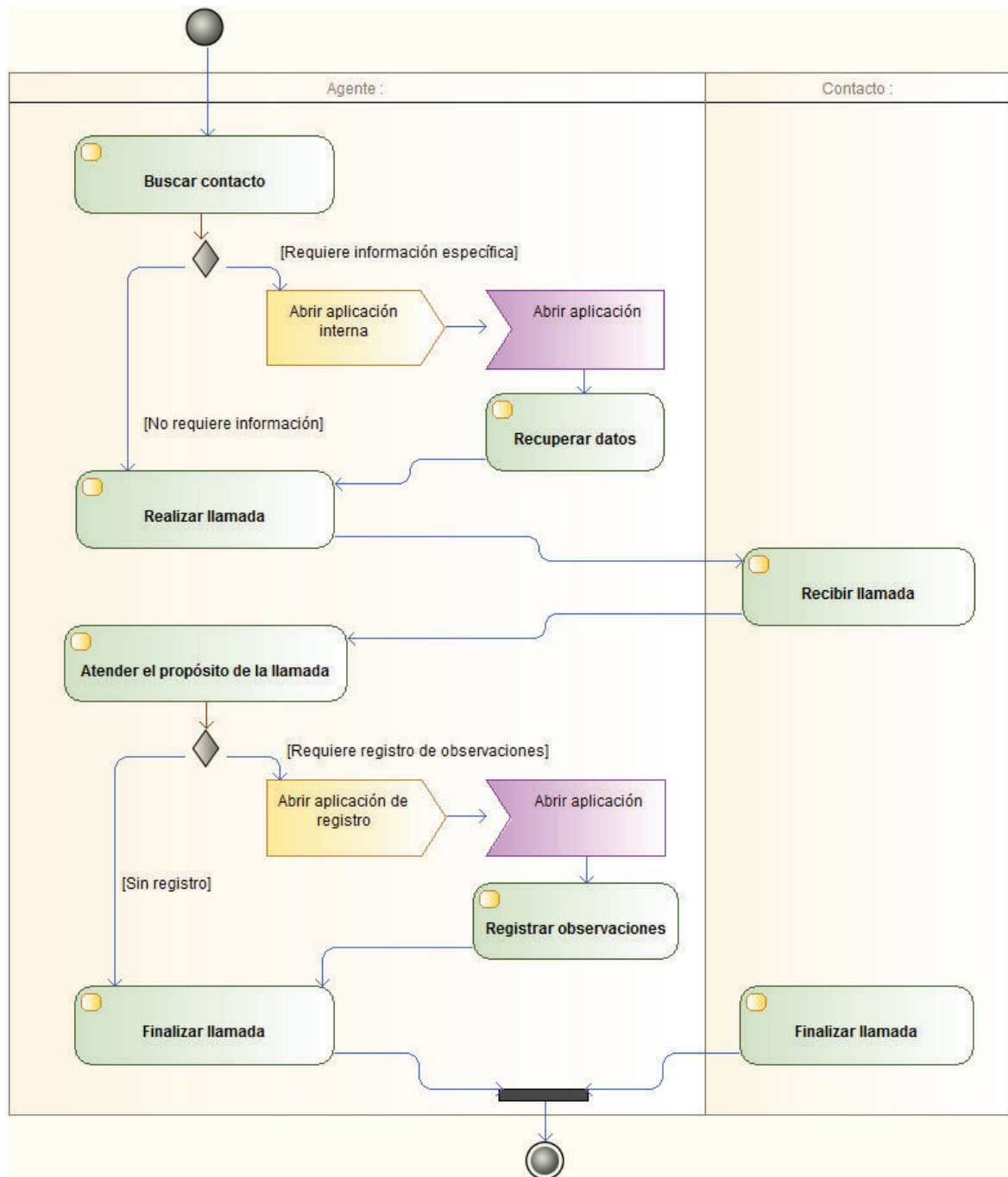


Figura 2.4 Proceso de Negocio de Agentes para la Realización de Llamadas.

Es importante notar que los *call centers* no están enfocados únicamente en recibir y procesar las llamadas entrantes, sino también realizar campañas de llamadas telefónicas salientes (*outbound*) con propósitos publicitarios, de cobranzas, encuestas, entre otros. En este caso, los agentes serán quienes inicien la comunicación, para lo cual pueden requerir de información relevante que les permita gestionar la llamada en proceso (dicha información usualmente es obtenida a través de aplicativos externos como CRMs o ERPs). El registro de observaciones adquiere gran importancia en el tratamiento de este tipo de llamadas en el que, al igual que en el caso anterior, los agentes utilizan aplicativos específicos o crean las anotaciones manualmente para su posterior análisis por parte del supervisor.

2.3.2.2. Supervisor

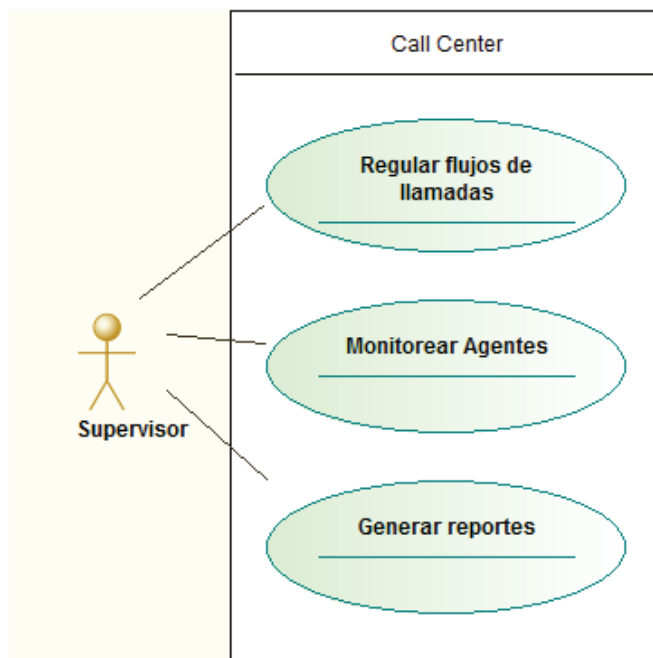


Figura 2.5 Caso de Uso – Supervisor

Un supervisor de *call center* es “la persona responsable de regular el flujo de llamadas entrantes y salientes de un grupo de agentes.”^[13] Además, un supervisor

tiene entre sus responsabilidades monitorear y gestionar el desempeño del equipo, clientes y recursos (tecnología) que forman parte del *call center*.

De forma específica, en el contexto de este proyecto, el supervisor será el encargado de regular el flujo de llamadas (a través del aplicativo de administración de la central telefónica), monitorear el desempeño del equipo de trabajo (agentes) y recuperar y analizar los reportes relacionados con las llamadas telefónicas gestionadas por el *call center*.

2.3.2.3. Administrador

De forma general, administrador es la persona responsable de la gestión, configuración y mantenimiento de los equipos, aplicaciones y servicios implicados en el funcionamiento de las plataformas de red, comunicaciones y sistemas informáticos. En una organización, por tanto, es posible encontrar más de un administrador, cuyas responsabilidades estén asignadas según el grado de especialización o de complejidad de los sistemas bajo su responsabilidad.

Desde la perspectiva de gestión de *call centers* (y específicamente uno que incluya la plataforma de telefonía IP Office) el administrador posee las siguientes tareas y responsabilidades (visualizadas en el diagrama de caso de uso de la figura 2.6):

- Crear, borrar y configurar extensiones (líneas telefónicas) y usuarios. Esta configuración es realizada desde el aplicativo de administración de la central telefónica Avaya IP Office (*Manager*).
- Gestionar la información de los contactos (clientes), la cual puede estar almacenada en bases de datos y/o utilizar aplicativos específicos.
- Administrar el adecuado funcionamiento de equipos, plataformas y servidores (hardware) que garanticen el óptimo desempeño de las aplicaciones que operan dentro de la institución (software).
- Monitorear el estado de operación de aplicaciones y servicios y responder a tiempo ante fallas o imprevistos para mantener los sistemas operativos el mayor tiempo posible para los usuarios.

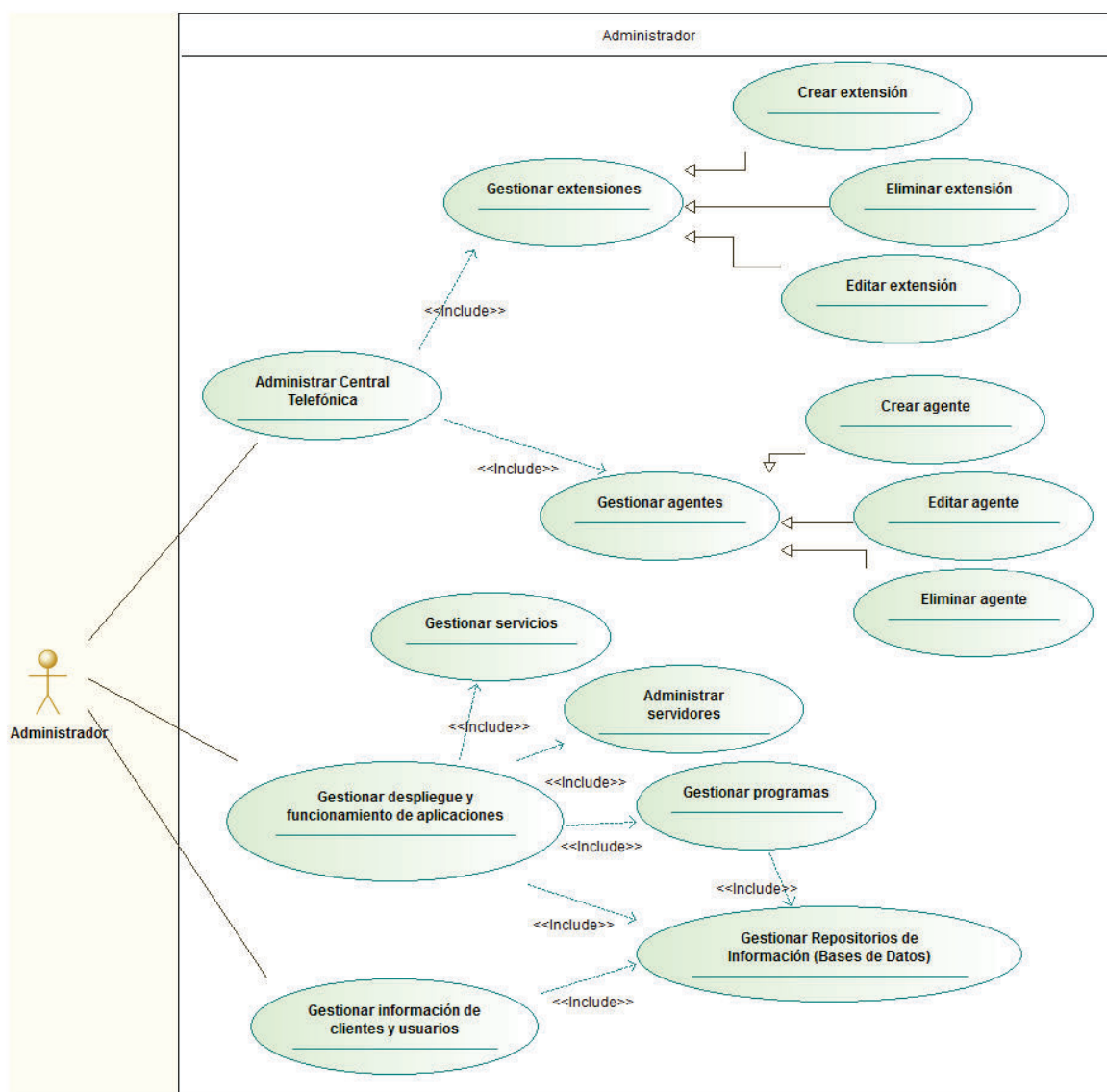


Figura 2.6 Caso de Uso – Administrador

2.3.3. REQUISITOS (IR)

Antes de comenzar con la creación de un sistema informático, primero es necesario determinar los principales requerimientos que justifican la creación del mismo, así como también determinar las posibles complicaciones que demanden un mayor análisis o detalle en su posterior desarrollo.

En este apartado, por lo tanto, serán descritos los requisitos que especificarán no sólo las funcionalidades del proyecto, sino que también limitarán el esquema sobre el cual éste será desarrollado, tomando en cuenta las herramientas y plataformas disponibles para tal efecto.

2.3.3.1. Requisitos Técnicos-Operativos (IR-TO)

Si bien es cierto que la creación de aplicaciones informáticas está centrada en satisfacer las necesidades de los usuarios (clientes), deben tomarse en cuenta las limitaciones técnicas y operativas que confinarán el alcance (y por ende el funcionamiento) de la solución.

En ese sentido (y dado que la metodología está centrada en la arquitectura), es imperativo mencionar el siguiente requisito técnico-operativo:

- ***Uso del TAPI de Avaya IP Office (IR-RT-T1)***

El sistema informático deberá utilizar la Interfaz de Programación de Aplicaciones de Telefonía (**TAPI**) de la central telefónica Avaya IP Office, proporcionada por la empresa Avaya y disponible únicamente para sistemas operativos **Windows**. La utilización de esta interfaz de aplicaciones de telefonía permite la creación de aplicaciones informáticas que puedan recibir y enviar eventos desde y hacia la central telefónica respectivamente. Este requisito forma parte como una de las entradas de la fase de inicio según la metodología de Proceso Unificado.

2.3.3.2. Requisitos Funcionales (IR-RF)

En este apartado se determinarán las necesidades y los intereses de carácter funcional que se desean incluir en el sistema. Dichos requisitos pueden estar basados en el estudio de los modelos de negocio y casos de uso descritos con antelación.

2.3.3.2.1. *Requerimientos Obligatorios (IR-RF-O)*

Comprenden las características técnicas que deben ser soportadas por el sistema y que tienen por objeto cubrir las necesidades básicas o primordiales de las organizaciones.

- ***Gestión y Control de Llamadas Telefónicas (IR-RF-O1)***

El sistema a desarrollar debe permitir la gestión y control de las líneas telefónicas (extensiones) desde un programa informático ejecutado en el computador de los usuarios (agentes). Específicamente, un agente de *call center* desde su computador deberá ser capaz de:

- Realizar llamadas.
- Colgar llamadas.
- Contestar llamadas.
- Poner llamadas en espera.
- Recuperar llamadas sostenidas.
- Transferir llamadas.
- Activar y desactivar del estado de No Molestar de la línea (DND).

- ***Despliegue de Información de los Contactos en una Llamada (IR-RF-O2)***

Los agentes deben poder visualizar la información de contacto de los clientes que participan en una llamada, obtenida desde un repositorio (base de datos) que será proporcionado por las instituciones interesadas.

Lo anterior implica que el sistema debe ser capaz de recuperar los datos de los contactos (clientes) con base en el identificador del número telefónico de la llamada en curso. Si hay más de una coincidencia (es decir, varios contactos poseen el mismo número telefónico registrado) el sistema debe ofrecer al agente la posibilidad de seleccionar el contacto más apropiado. De forma adicional, si se dispone de coordenadas de ubicación geográfica (latitud y longitud) el agente deberá visualizar la localización del contacto en un mapa.

- ***Gestión e Ingreso de Observaciones de las Llamadas (IR-RF-O3)***

El software debe permitir el ingreso de observaciones de las llamadas procesadas por los agentes para su posterior procesamiento y visualización por parte del supervisor.

- ***Detalle del Número Telefónico de Contacto en una Llamada Externa (IR-RF-O4)***

La aplicación deberá mostrar información adicional relacionada con el número telefónico de contacto en llamadas externas (entrantes o salientes) basada en los planes de numeración del Ecuador⁸ que deberán ser cargados y actualizados por el Administrador del sistema. Durante el transcurso de una llamada externa, los agentes visualizarán:

- El nombre de la operadora para todos los números telefónicos.
- El nombre de la provincia y el nodo (nombre asignado a la central telefónica) para los números telefónicos fijos solamente.

Adicionalmente, la información anterior deberá ser utilizada en la generación de reportes y estadísticas por parte del supervisor del sistema.

- ***Multiplataforma (IR-RF-O5)***

Los usuarios del sistema deberán ser capaces de utilizar los servicios y administrar las características de la plataforma desde los principales sistemas operativos disponibles en el mercado. El objetivo primordial de este requisito radica en la creación del menor número de aplicaciones dedicadas de usuario final posibles que cubran las expectativas planteadas.

- ***Monitoreo y Supervisión de Agentes (IR-RF-O6)***

El supervisor deberá ser capaz de visualizar el estado de las extensiones de los agentes en cualquier momento, de ser posible en tiempo real.

⁸ Véase la sección “1.1.7 Planes de Numeración”.

Adicionalmente, el supervisor deberá tener la posibilidad de enviar mensajes o notificaciones a los agentes registrados en el sistema.

- ***Gestión de Reportes y/o Estadísticas (IR-RF-07)***

El sistema deberá almacenar información de todas las llamadas gestionadas para su posterior procesamiento en la generación de reportes y/o estadísticas por parte del supervisor.

- ***Gestión de Usuarios, Servicios y Comunicaciones (IR-RF-08)***

El software permitirá gestionar (crear, eliminar o modificar) los usuarios que utilicen la plataforma, los servicios utilizados (sean externos o internos) y las comunicaciones existentes entre todos los componentes que la conforman. En otras palabras, el sistema debe permitir la configuración de los actores, servicios y características que permitan su adecuada operación.

2.3.3.2.2. Requerimientos Opcionales (IR-RF-P)

Son las características especiales que mejoran, optimizan o facilitan la utilización y el funcionamiento de la solución. Al ser de carácter opcional no siempre son incluidos en las etapas iniciales de los proyectos, cuya implementación (o solución) es realizada a través de un cuidadoso análisis en términos técnicos, operativos y económicos que no excedan los alcances definidos en las fases de inicio y elaboración.

- ***Integración con Aplicaciones Externas (IR-RF-P1)***

La aplicación deberá contemplar la posibilidad de ofrecer alternativas de integración con aplicaciones externas, mediante el paso de parámetros asociados a la llamada en curso. Esto significa que el agente podría optar por invocar aplicaciones externas (entiéndase por externa cualquier otra herramienta informática ajena al proyecto) con el propósito de obtener información significativa de la llamada y de esta manera optimizar la atención de llamadas en los *call centers*.

- ***Fácil Distribución (IR-RF-P2)***

La aplicación de interacción con los usuarios (agentes) será fácilmente distribuida, instalada y configurada en sus computadores, soportando una fácil distribución de actualizaciones y cambios.

- ***Escalable (IR-RF-P3)***

El software debe soportar un crecimiento sostenido y razonable del número de usuarios que puedan utilizar la herramienta.

- ***Flexible (IR-RF-P4)***

La solución deberá contemplar la posibilidad de integración con otras plataformas de comunicaciones, compatibles o no con TAPI y ajenas a la central telefónica Avaya IP Office, de la forma menos compleja posible.

De manera adicional, el sistema podría ofrecer la posibilidad de gestionar los planes de comunicación, parámetros de configuración y bases de datos en cualquier momento de forma que se adapten a las necesidades de las instituciones.

- ***Tolerancia a Fallos (IR-RF-P5)***

El sistema buscará soportar fallos básicos de energía y comunicaciones que puedan alterar perjudicialmente su funcionamiento.

- ***Estable (IR-RF-P6)***

La solución procurará funcionar la mayor cantidad de tiempo sin fallos y con el menor tiempo dedicado a la administración de sus componentes.

2.3.3.3. Requisitos No Funcionales (IR-NF)

Los requisitos no funcionales establecen las características que no tienen una influencia directa sobre la solución y funcionamiento de las necesidades planteadas

en los requisitos funcionales, pero ejercen una gran influencia en la percepción que los clientes y usuarios tienen de la plataforma.

- ***Económico (IR-NF-F1)***

En lo posible se priorizarán los recursos, servicios y tecnologías de libre distribución que minimicen el costo final del proyecto sin afectar la calidad y funcionamiento del sistema.

- ***Atractivo (IR-NF-F2)***

La solución buscará ofrecer una experiencia confortable para agentes, supervisores y administradores, con el propósito de fomentar el uso de la herramienta.

- ***Sencillo (IR-NF-F3)***

En el proyecto se tomará en cuenta la facilidad de uso del software mediante un diseño intuitivo y sencillo de sus interfaces.

- ***Seguro (IR-NF-F4)***

La aplicación intentará incluir el aseguramiento y protección de la información sensible o importante transmitida sobre la red. Si bien este requisito también puede ser considerado como operativo, debe aclararse que la complejidad asociada podría exceder el alcance del proyecto.

2.3.4. ANÁLISIS (IA)

Esta sección tiene por objeto refinar y procesar la información recopilada en el apartado anterior, así como establecer las directrices generales de funcionamiento, composición y distribución del sistema a desarrollar. Cabe aclarar que el análisis de la fase de inicio no define explícitamente la interacción de cada una de las clases o componentes que conforman el sistema, ya que dicha tarea forma parte de la etapa de análisis de la fase de elaboración según la metodología de desarrollo elegida (Proceso Unificado).

2.3.4.1. Categorización de los Requisitos del Sistema (IA-CA)

En este apartado se agruparán los requerimientos anteriormente señalados según la relación que tengan con las actividades realizadas por cada uno de los roles del *call center*. En otras palabras, este apartado pretende definir qué grupo de requisitos corresponde a cada actor. Debe considerarse que dos o más roles pueden compartir un mismo requisito, o que éste tenga que ser dividido (o especializado) para cubrir las necesidades que cada rol requiere.

Para ello serán considerados los siguientes perfiles:

- Global (contendrá las características comunes a todos los perfiles o que pertenecen al sistema).
- Agente.
- Supervisor.
- Administrador.

La tabla 2.2 categoriza y clasifica los requisitos funcionales y no funcionales basándose en su importancia o aporte conforme al cumplimiento de los objetivos de la solución. Para ello se han tomado en cuenta los siguientes aspectos:

- **Perfil.** Determina el perfil al que corresponde el requisito en cuestión.
- **Requisito.** Nombre o descripción corta del requisito.
- **Código.** Código del requisito.
- **Tipo.** Carácter del requisito (funcional, no funcional, obligatorio, optativo).
- **Peso.** El peso de un requisito está relacionado al tipo. Un requisito de carácter técnico recibe el máximo valor (5); uno de tipo obligatorio obtiene un peso de 4 y los requisitos optativos adquieren valores cambiantes entre 1 y 3, respecto a su influencia en el cumplimiento de los objetivos del proyecto.

- **Relevancia**⁹. Determina la medida de importancia del requisito dentro de la categoría a la que pertenece. De un máximo de 5, un requerimiento tendrá mayor significancia mientras mayor sea su valor.
- **Total**. Especifica el valor de importancia del requisito en el tratamiento y análisis del proyecto. Para este trabajo los requisitos con un valor total mayor a doce serán prioritarios.

De lo anterior se infiere la posibilidad de que algunos requisitos optativos tengan una mayor prioridad que los obligatorios, por lo que serán determinantes en el diseño y selección de la arquitectura. Esto no quiere decir que los requisitos obligatorios con una baja calificación no serán implementados, sino que tendrán una menor inversión de tiempo y recursos en su análisis, diseño e implementación.

Perfil	Requisito	Código	Tipo	Peso	Relevancia	Total
Global	TAPI de IP Office	IR-RT-T1	Técnico	5	5	25
	Multiplataforma	IR-RF-O5	Funcional Obligatorio	4	5	20
	Escalable	IR-RF-P3	Funcional Optativo	2	3	6
	Flexible	IR-RF-P4	Funcional Optativo	2	2	4
	Tolerante a fallos	IR-RF-P5	Funcional Obligatorio	3	3	9
	Estable	IR-RF-P6	Funcional Optativo	2	3	6
	Económico	IR-NF-F1	No Func. Optativo	3	4	12
	Atractivo	IR-NF-F2	No Func. Optativo	1	2	2

⁹ El peso y la relevancia de los requisitos son valorados por los creadores del proyecto, por lo que pueden ser subjetivos y/o estar sujetos a cambios que se ajusten a los objetivos del mismo.

	Sencillo	IR-NF-F3	No Func. Optativo	2	2	4
	Seguro	IR-NF-F4	No Func. Optativo	2	2	4
Agente	Gestión y control de llamadas telefónicas	IR-RF-O1	Funcional Obligatorio	4	5	20
	Despliegue de información de los contactos en una llamada	IR-RF-O2	Funcional Obligatorio	4	5	20
	Gestión e ingreso de observaciones de las llamadas	IR-RF-O3	Funcional Obligatorio	4	3	12
	Detalle del número telefónico de contacto en una llamada externa	IR-RF-O4	Funcional Obligatorio	4	3	12
	Integración con aplicaciones externas	IR-RF-P1	Funcional Optativo	3	3	9
	Fácil distribución	IR-RF-P2	Funcional Optativo	1	2	2
Supervisor	Monitoreo y supervisión de agentes	IR-RF-O6	Funcional Obligatorio	4	5	20
	Gestión de reportes y/o estadísticas	IR-RF-O7	Funcional Obligatorio	4	5	20
Administrador	Gestionar usuarios, servicios y comunicaciones	IR-RF-O8	Funcional Obligatorio	4	5	20

Tabla 2.2 Categorización de los Requisitos del Sistema

El análisis de la tabla 2.2 establece que los requisitos que tendrán una influencia importante en la conformación de la arquitectura son los siguientes:

- **Requisitos Primarios (IA-CA-1)**

Los requisitos primarios serán influyentes y primordiales en la determinación de la arquitectura de la solución, cuyo valor total sea superior o igual a doce (12).

Requisito	Código	Total
TAPI de IP Office	IR-RT-T1	25
Gestión y control de llamadas telefónicas	IR-RF-O1	20
Despliegue de información de los contactos en una llamada	IR-RF-O2	20
Monitoreo y supervisión de agentes	IR-RF-O6	20
Gestión de reportes y/o estadísticas	IR-RF-O7	20
Gestionar usuarios, servicios y comunicaciones	IR-RF-O8	20
Multiplataforma	IR-RF-O5	20
Económico	IR-NF-F1	12
Gestión e ingreso de observaciones de las llamadas	IR-RF-O3	12
Detalle del número telefónico de contacto en una llamada externa	IR-RF-O4	12

Tabla 2.3 Requisitos Primarios

Requisito	Código	Total
Tolerante a fallos	IR-RF-P5	9
Integración con aplicaciones externas	IR-RF-P1	9
Escalable	IR-RF-P3	6
Estable	IR-RF-P6	6
Flexible	IR-RF-P4	4
Sencillo	IR-NF-F3	4
Seguro	IR-NF-F4	4
Atractivo	IR-NF-F2	2
Fácil distribución	IR-RF-P2	2

Tabla 2.4 Requisitos Secundarios

- **Requisitos Secundarios (IA-CA-2)**

Los requisitos secundarios (ver tabla 2.4) no serán tomados en cuenta en la determinación de la arquitectura de la solución. Lo anterior quiere decir que la arquitectura será definida según el cumplimiento de los requisitos primarios, mientras que los secundarios tendrán que acomodarse a ella según sea conveniente.

2.3.4.2. Análisis de los Requisitos del Sistema (IA-AR)

En esta sección se realizará un análisis de los requisitos del sistema más importantes, riesgosos o que requieren mayor detalle, con el objeto de encontrar las limitaciones y dificultades del proyecto, así como también establecer un marco inicial de la arquitectura de la solución.

2.3.4.2.1. Utilización de la Interfaz de Aplicaciones de Telefonía de Avaya IP Office (IA-AR-1)

El requisito técnico-operativo IR-TO-T1 especifica que para la creación de un sistema CTI integrado con la central telefónica Avaya IP Office debe utilizarse el TAPI disponible únicamente para sistemas operativos Windows.

El documento *IP Office Product Description* en el capítulo 17, página 357, detalla las características y escenarios de integración y creación de aplicaciones CTI. En resumen, IP Office ofrece seis recursos e interfaces:

- **TAPILink Lite:** provee características CTI básicas para controlar o monitorear el teléfono de un solo usuario. No requiere licenciamiento adicional. Soporta TAPI 2.1 y 3.0.
- **TAPILink Pro:** ofrece funcionalidades CTI avanzadas para el control y monitoreo de múltiples usuarios, para lo cual es necesaria la adquisición de licenciamiento adicional. Soporta TAPI 2.1 y 3.0.
- **TAPI-WAV driver:** soporte de software para el procesamiento de voz. Soporta TAPI 2.1 solamente. Requiere licenciamiento adicional.

- **IP Office SMDR:** Proporciona una interfaz para la obtención de eventos SMDR.
- **DevLink Pro:** Provee flujos de eventos en tiempo real adicionales la interfaz SMDR.
- **IP Office Software Development Kit:** Contiene un conjunto de herramientas con la documentación de las interfaces anteriores, así como programas de ejemplo y código fuente para la familiarización con la plataforma.

Dado que en ninguno de los requisitos planteados (tanto primordiales como secundarios) se solicitan funcionalidades de voz (IVR) o tarificación, los recursos *TAPI-WAV driver*, *IP Office SMDR* y *DevLink Pro* quedan fuera del análisis del sistema.

El siguiente paso en el análisis corresponde a la elección de la versión del TAPI a utilizar, empleando el cuadro comparativo establecido en la tabla 2.5.

Versión	Ventajas	Desventajas	Contribuye en Requisitos
TAPILink Lite	<ul style="list-style-type: none"> • Controla una sola extensión (ideal en aplicaciones <i>stand-alone</i>). • No necesita licenciamiento (más económico). 	<ul style="list-style-type: none"> • Posee funcionalidades básicas. • Requiere que cada usuario posea un equipo con sistema operativo Windows. 	<ul style="list-style-type: none"> • IR-NF-F1: Económico • IR-NF-F3: Sencillo
TAPILink Pro	<ul style="list-style-type: none"> • Control de múltiples extensiones (ideal para arquitecturas cliente-servidor). • Soporta funcionalidades avanzadas (más flexible). 	<ul style="list-style-type: none"> • Requiere licenciamiento adicional. • Mayor complejidad de desarrollo. 	<ul style="list-style-type: none"> • IR-RF-P1: Multiplataforma • IR-RF-P3: Escalable • IR-RF-P4: Flexible

Tabla 2.5 Versiones CTI de IP Office

Tomando en cuenta las principales características, ventajas y desventajas de cada tecnología CTI de Avaya IP Office ha sido posible deducir que la versión más apropiada para la creación de un sistema de gestión y control de llamadas telefónicas es **TAPILink Pro**, ya que a pesar del aumento de la complejidad del desarrollo y la adquisición de licenciamiento adicional, ofrece funcionalidades avanzadas que permiten una mayor flexibilidad y escalabilidad de la solución, así como la posibilidad de crear un sistema multiplataforma.

Este último punto debe analizarse con cuidado puesto que, debido a que TAPI funciona exclusivamente en sistemas operativos Windows, puede verse afectado el propósito de la creación de un sistema multiplataforma. En este sentido, lo que se quiere crear es un aplicativo distribuido de tipo multiplataforma para los **usuarios** del sistema, es decir, agentes, supervisores y administradores (tratando de evitar la creación de aplicativos individuales para los principales sistemas operativos del mercado). El hecho de que sea posible crear una aplicación de tipo *Third-Party* permite la integración de tecnologías multiplataforma con el correspondiente gestor de comunicaciones (TAPI) en Windows.

2.3.4.2.2. Despliegue de la Información de los Contactos en una Llamada (IA-AR-2)

El requisito funcional del agente IR-RF-O2 relacionado con el despliegue de la información de los contactos que participan en una llamada requiere del esclarecimiento de varios puntos:

- ***Integración con Bases de Datos (IA-AR-2a)***

La recuperación de la información de los contactos desde repositorios centralizados de información (bases de datos) es un proceso que puede complicarse en la medida del nivel de detalle de la información que sea presentada. Por tal motivo se vuelve imperativo examinar ciertos puntos:

- **La información de los contactos es variable.** Esto quiere decir que el nivel de detalle de la información desplegada al agente puede ser modificada en función de las metas o necesidades de las instituciones,

o en otras palabras, las instituciones frecuentemente modifican o extienden la información de los clientes, la cual debe ser presentada a los agentes en los *call centers* para la atención de llamadas telefónicas. Lo anterior implica que debe crearse un sistema que permita flexibilidad en la configuración y selección de la información de los clientes que será recuperada desde la base de datos de la institución.

- **La selección de la información debe ser sencilla.** La información relacionada con los usuarios debe ser configurada de la forma más sencilla, y, de ser posible, gestionada dentro de los mismos esquemas de gestión de información de los clientes. La segmentación de responsabilidades facilita el desarrollo de la aplicación y minimiza el tiempo invertido en el aprendizaje innecesario de habilidades con poca significancia o de gran complejidad.
- **La modificación de datos no es posible.** Muchas organizaciones utilizan procesos de negocio, funciones o algoritmos que son ejecutados por las mismas aplicaciones institucionales ante los cambios de los atributos de los usuarios. Por tal motivo, y dado que el sistema que se pretende ofrecer debe abarcar la mayor cantidad de instituciones posible, se concluye que la edición de los datos de los contactos no será incluida. Esto implica que el agente sólo podrá visualizar la información de los clientes, pero no podrá editarla desde la misma interfaz. Para ello será necesaria la utilización de las aplicaciones especializadas de cada empresa.
- **Existen muchas bases de datos.** Cada organización utiliza el (o los) gestor(es) de bases de datos que mejor se ajuste(n) a sus necesidades. Esto significa que la solución propuesta debe contemplar la posibilidad de integración con las principales bases de datos y con el menor tiempo/costo de desarrollo del sistema.

De los puntos tratados anteriormente, se puede concluir que:

- Se requiere de un aplicativo capaz de interconectar varios tipos de bases de datos de la forma más estandarizada posible. Dicho de otro modo, se requiere de una plataforma común de acceso a datos. Está por demás decir que un sistema *general* de acceso a datos es considerablemente complejo y no es el principal objetivo del presente proyecto.
- La solución propuesta no podrá modificar la información de los contactos (clientes).
- La creación (o especificación) de la información que se presentará a los usuarios será manejada fuera de la solución. Desde el sistema a desarrollar será posible elegir el repositorio externo desde el cual se recuperará la información previamente formateada (o especificada).

Ahora bien, el análisis de un escenario simple de integración con bases de datos debería estar determinado por las condiciones técnicas y operativas de cada organización (ya que cada institución maneja su información de manera distinta). Sin embargo, en este estudio se realizará una estimación de los escenarios más probables para el acceso a la información de los clientes.

- ***Registros Repetidos (IA-AR-2b)***

El sistema utilizará el número telefónico del contacto para buscar la información del cliente en la base de datos de la institución. Si se recupera más de una coincidencia, el sistema debe ofrecer al agente la posibilidad de seleccionar el registro apropiado.

- ***Visualización de la Información Geográfica de la Llamada (IA-AR-2c)***

En el transcurso de una llamada, los agentes deberán visualizar la ubicación geográfica de los clientes en un mapa.

Este punto enfrenta dos limitaciones principales:

- La localización debe ser realizada sólo para números telefónicos fijos. No tiene sentido localizar números telefónicos móviles (celulares) por limitaciones tecnológicas y de servicios. Sin embargo, dado el creciente auge y masificación de los teléfonos inteligentes (smartphones), la creación de aplicaciones que permitan localizar estos dispositivos es posible mediante la creación e instalación de aplicaciones dedicadas. Este proyecto no tomará en cuenta tal escenario.
- Las coordenadas geográficas (latitud y longitud) de la localización de los contactos deben ser provistas por las instituciones interesadas. Este sistema es independiente de la información de los contactos por motivos anteriormente expresados en el análisis IA-AR-2a. Por tal razón, el sistema constituye una herramienta de visualización de la información de los contactos, en la que puede estar incluida la localización de los mismos de ser el caso.

Para las instituciones interesadas en la localización geográfica de las llamadas, la Corporación Nacional de Telecomunicaciones en su página web ofrece un servicio gratuito de localización de números telefónicos fijos y disponibilidad de líneas telefónicas en el siguiente enlace: <http://gis.cnt.gob.ec/giscnt/php/dispoapi.php>. En la medida de lo técnicamente posible, deberá considerarse la posibilidad de integrar este servicio gratuito en la plataforma que se desea desarrollar con el objeto de simplificar los procesos de las instituciones y ofrecer una solución unificada.

2.3.4.2.3. Detalle del Número Telefónico de Contacto en una Llamada Externa (IA-AR-3)

La visualización del nombre de la provincia y la operadora en el transcurso de una llamada telefónica implica utilizar los planes de numeración publicados por la ARCOTEL.

Dado el carácter variable de los rangos de números telefónicos publicados, debe contemplarse la posibilidad de actualizar los planes de numeración de forma manual y de la manera más sencilla posible para facilitar la administración del sistema.

Los datos publicados en la página web de la ARCOTEL son descargados como dos archivos en formato de archivo de Excel (extensión .xlsx), cuyo contenido está agrupado en pestañas:

- **Resumen.** Contiene los porcentajes de asignación y reserva de los números telefónicos.

RECURSO NUMÉRICO		PORCENTAJES DE ASIGNACIÓN Y DISPONIBILIDAD							
		2	3	4	5	6	7	8	9
ASIGNADA	CORPORACIÓN NACIONAL TELECOMUNICACIONES CNT EP	14,83%	4,87%	12,63%	3,64%	3,96%	5,46%	-	-
	ETAPA EP	0,04%	0,00%	0,05%	0,01%	0,00%	3,20%	-	-
	LINKOTEL S.A.			0,25%	0,03%			-	-
	ECUADORTELECOM S.A.	0,87%	0,02%	1,24%	0,06%	0,08%	0,09%	-	-
	SETEL S.A.	0,69%	0,07%	0,62%	0,15%	0,05%	0,15%	-	-
	LEVEL 3 ECUADOR LVL3 S.A.	0,25%						-	-
	CORIPAR S.A.	0,06%						-	-
	TOTAL	16,74%	4,98%	14,79%	3,89%	4,09%	8,90%	0,00%	0,00%
DISPONIBLE	83,26%	95,02%	85,21%	96,11%	95,91%	91,10%	100,00%	100,00%	
TOTAL	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%	

Figura 2.7 Información del Plan Técnico Fundamental de Numeración – Resumen

- **Gráficos.** Visualización gráfica de la distribución de los números telefónicos por zonas.
- **Anexos** (2-9 para telefonía fija, 92-99 para telefonía móvil). Contienen la información detallada del plan de numeración, en el que cada anexo corresponde a una zona o grupo (por ejemplo: anexo 3 - código de zona 3 en telefonía fija).

Los datos publicados para cada zona se presentan en una tabla, cuyas columnas son las siguientes:

- Número de registro.
- Nombre de la central.*
- Serie asignada.*
- Capacidad.

- Provincia (sólo telefonía fija). *
- Mes de uso.
- Operadora. *

* Estas columnas ofrecen información de relevancia para el sistema que se quiere crear.

El número de registro, capacidad y mes de uso no son importantes para la gestión de las llamadas por parte de los agentes.

El nombre de la central podría parecer innecesario a primera vista, pero para el caso de los números telefónicos fijos este nombre está asociado al sector en el que se brinda el servicio, como se puede apreciar en la figura 2.8.

Servicio Telefonía Fija Plan Técnico Fundamental de Numeración: Anexo 3 Fecha de publicación: 31 de julio 2014							
NUMERACION CORRESPONDIENTE AL C.A. "3"				PROV. O CIUD.	MES DE USO	OP.	
No.	NOMBRE DE LA CENTRAL	SERIE ASIGNADA		CAP.			
1	BALZAPAMBA	2.200.000	2.200.299	300	BOLI	CNT	
2	SAN LORENZO (NQU1)	2.201.000	2.201.199	200	BOLI	CNT	
3	FACUNDO VELA	2.203.000	2.203.299	300	BOLI	CNT	
4	CUATRO ESQUINAS (NQU1)	2.205.000	2.205.199	200	BOLI	CNT	
5	GUANUJO (NQU1)	2.206.000	2.207.199	1.200	BOLI	CNT	
6	SALINAS	2.210.000	2.210.299	300	BOLI	CNT	
7	SAN JOSÉ DE TAMBO	2.213.000	2.213.599	600	BOLI	CNT	
8	SAN JOSÉ DE ATENAS	2.217.000	2.217.299	300	BOLI	CNT	
9	SAN SIMÓN	2.220.000	2.220.199	200	BOLI	CNT	
10	SIMIÁTUG	2.223.000	2.223.399	400	BOLI	CNT	
11	CASIPAMBA (NQU1)	2.225.000	2.225.299	300	BOLI	CNT	
12	LAS NAVES 1 (NQU1)	2.227.000	2.227.199	200	BOLI	CNT	
13	SANTIAGO (NQU1)	2.229.000	2.229.199	200	BOLI	CNT	
14	JOSÉ GUANGO ALTO (NQU1)	2.230.000	2.230.399	400	COTX	CNT	
15	LATACUNGA GPON (NQU1)	2.233.000	2.236.599	3.600	COTX	CNT	
16	SAN JOSÉ DE YUGSILOMA (NQU1)	2.240.000	2.240.599	600	COTX	CNT	
17	SAN TAN (NQU1)	2.242.000	2.242.499	500	COTX	CNT	
18	LATACUNGA 3 (NQU1)	2.245.000	2.246.499	1.500	COTX	CNT	
19	SICOTO (NQU1)	2.248.000	2.248.099	100	COTX	CNT	
20	TINGO LA ESPERANZA (NQU1)	2.249.000	2.249.299	300	COTX	CNT	
21	SAN CARLOS (NQU1)	2.250.000	2.250.599	600	COTX	CNT	
22	SAN FELIPE (NQU1)	2.252.000	2.253.799	1.800	COTX	CNT	
23	SAN JOSÉ DE PICHUL (NQU1)	2.255.000	2.255.299	300	COTX	CNT	
24	SAN JOSÉ DE POALÓ (NQU1)	2.257.000	2.257.399	400	COTX	CNT	
25	RAMÓN CAMPAÑA (NQU1)	2.258.000	2.258.099	100	COTX	CNT	
26	PUSUCHISI (NQU1)	2.259.000	2.259.199	200	COTX	CNT	
27	ANTONIO JOSÉ HOLGUÍN	2.260.000	2.260.599	600	COTX	CNT	
28	ALAUQUEZ	2.262.000	2.263.599	1.600	COTX	CNT	
29	BELISARIO QUEVEDO	2.266.000	2.267.099	1.100	COTX	CNT	

Figura 2.8 Detalle del Anexo 3 (código de área 3) del PTFN

2.3.4.2.4. Sistema Multiplataforma (IA-AR-4)

La creación de sistemas multiplataforma conlleva una enorme inversión de tiempo y recursos en la determinación del escenario más apropiado para que los usuarios puedan acceder a los servicios de la solución desde los principales sistemas operativos.

Tomando en cuenta los otros requisitos del sistema, el análisis establece los siguientes puntos:

- **Los agentes necesitan gestionar los eventos de las llamadas en tiempo real (IA-AR-4a)**

Debido a que los agentes del *call center* requieren de la notificación asincrónica de eventos de la línea telefónica (un ejemplo de ello sería la recepción de una llamada), es necesario un aplicativo específico que ofrezca tales características. Adicionalmente, si se considera la posibilidad de integración con aplicaciones externas o características de georreferenciación, debe contemplarse la posibilidad de la creación de un cliente de escritorio multiplataforma.

- **Los administradores y supervisores requieren características de gestión de contenido estático (IA-AR-4b)**

Dado que los administradores y supervisores no requieren gestionar eventos en tiempo real, pueden utilizarse métodos multiplataforma de acceso y configuración de las características de la aplicación, como por ejemplo, a través de un navegador web, eliminando la necesidad de instalar y configurar un cliente de escritorio.

- **La base de datos del sistema debe ser multiplataforma (IA-AR-4c)**

De igual forma, el gestor de base de datos propio de la solución debe ser capaz de ser instalado y accedido desde cualquier sistema operativo soportado por el gestor de comunicaciones.

- **Debe considerarse la posibilidad de integración con otras plataformas de telefonía (IA-AR-4d)**

Dado que Avaya IP Office no es la única solución de telefonía que provee aplicativos de integración CTI, debe tomarse en cuenta la interconexión con otras plataformas similares sin afectar grandemente el diseño multiplataforma del sistema.

Por tal razón, es necesaria la segmentación de funcionalidades (roles) que se acoplen al esquema propuesto, estableciéndose el escenario de comunicación siguiente:

- **Servidor CTI.** Encargado de la gestión de los eventos de las líneas telefónicas y la interconexión con las centrales telefónicas, es dependiente de las características definidas por el fabricante (es decir, será dependiente de la plataforma). Constituye el elemento intermediario entre las aplicaciones informáticas y el sistema tradicional de telefonía.
- **Servidor de comunicaciones.** Es el sistema multiplataforma de procesamiento de eventos producidos por los usuarios (agentes, supervisores y administradores) y el servidor CTI.

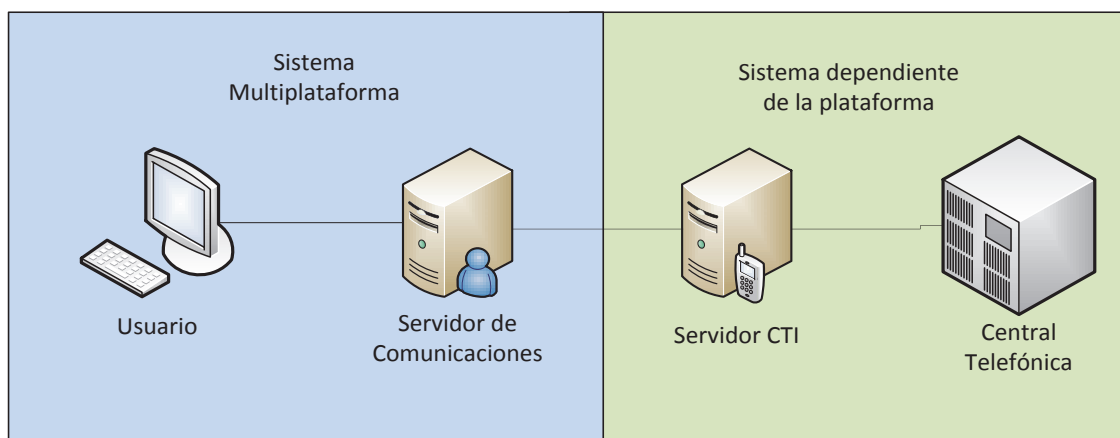


Figura 2.9 Sistema “Multiplataforma” de Gestión de *Call Center*.

- **Integración con Aplicaciones Externas.** Al ser un proceso complicado y de alto riesgo que demanda conocimientos y recursos avanzados que garanticen el éxito de la solución se considerará el escenario de integración más simple y solamente en las etapas finales del desarrollo del sistema con propósitos puramente académicos o de aprendizaje. Lo anterior es determinado por el bajo valor de relevancia del requisito en la sección de categorización.

2.3.4.3. Análisis de la Arquitectura (IA-AA)

Para plantear la arquitectura es importante tomar en cuenta todos los aspectos funcionales, económicos, operativos y técnicos que influyen en la solución. Esta sección pretenderá definir un esqueleto del sistema a construir con base en los requerimientos (técnicos, funcionales y no funcionales) y sus respectivos análisis descritos en apartados anteriores.

2.3.4.3.1. Elementos

El análisis IA-AR-1 concluye que la interfaz de programación de aplicaciones de telefonía a utilizarse en este proyecto es *Avaya TAPILink Pro*, lo que implica que el software estará enfocado en una **arquitectura cliente-servidor**. En esta arquitectura los agentes enviarán los comandos de gestión y control de llamadas telefónicas al servidor central, que a su vez procesará y enviará los comandos necesarios a la central telefónica a través del TAPI. Adicionalmente, el servidor principal será el encargado del establecimiento de la conexión con la base de datos que contiene la información de los clientes que participan en las llamadas telefónicas externas.

De lo anterior puede concluirse que la solución de telefonía estará conformada de varios componentes o elementos mencionados a continuación y que se indican en la figura 2.10.

Central telefónica Avaya IP Office. La central telefónica es un elemento crucial de la plataforma, para lo cual debe adquirirse todo el licenciamiento necesario para el adecuado funcionamiento e integración de la aplicación.

Servidor CTI Windows. Se requiere de un servidor Windows que utilice la interfaz de aplicaciones de telefonía de Avaya, el cual será en el encargado de enviar y recibir comandos y eventos desde/hacia la central telefónica.

Servidor Principal de Gestión de Comunicaciones. Según el análisis de requisitos (IA-AR-4), debe considerarse un sistema centralizado de gestión de eventos multiplataforma, interconectado al servidor CTI. Este servidor de comunicaciones integrará una base de datos multiplataforma que contendrá información de las extensiones, usuarios, planes de numeración, registros de eventos, entre otros.

Repositorio de información de los clientes. La información detallada de los clientes debe estar almacenada en bases de datos, con perfiles definidos para la búsqueda y navegación de registros.

Usuarios. Especificado en el análisis de requerimientos IA-AR-4a e IA-AR-4b, los agentes requieren de un aplicativo de escritorio que gestione los eventos de la línea telefónica en tiempo real, mientras que administradores y supervisores pueden utilizar navegadores web para sus funciones.

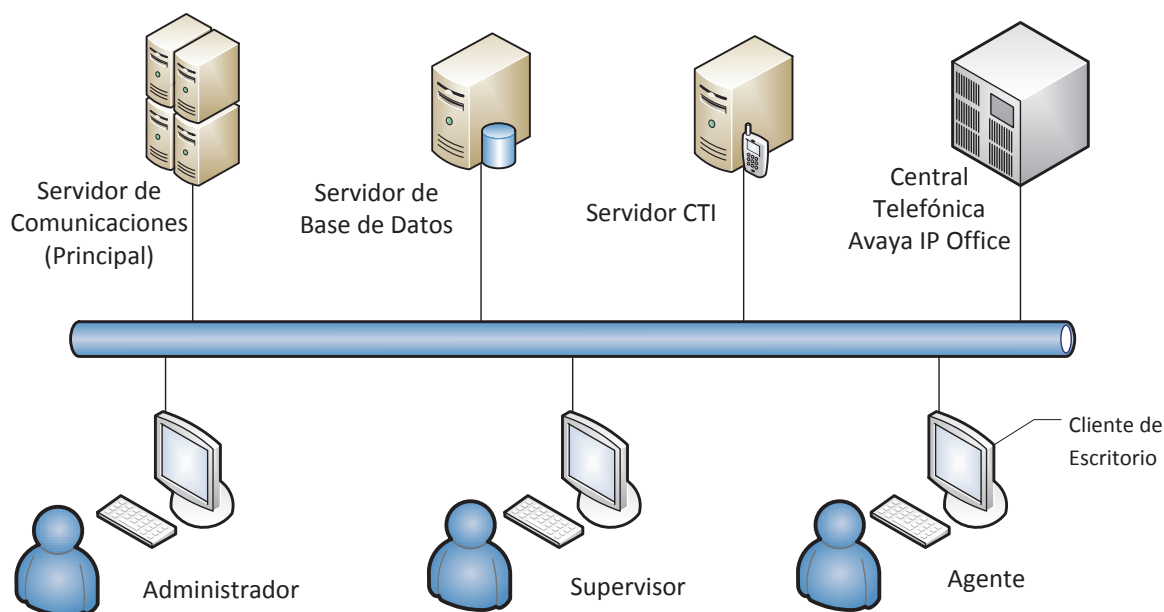


Figura 2.10 Arquitectura Candidata

Las secciones posteriores tratarán en mayor detalle cada uno de los componentes que conforman la arquitectura anteriormente descrita.

2.3.4.3.2. Plataforma de Telefonía Avaya IP Office (IA-AA-IPO)

La central telefónica Avaya IP Office es una “... *solución de comunicaciones versátil que combina la confiabilidad y facilidad de un sistema de telefonía tradicional con las aplicaciones y ventajas de una solución de telefonía IP. Esta solución de comunicaciones convergente puede ayudar a los negocios a reducir costos, incrementar la productividad y mejorar el servicio al cliente.*” [5]

La central telefónica en cuestión está enfocada en brindar soluciones de comunicaciones de telefonía a las pequeñas y medianas empresas, con un soporte máximo de 384 usuarios en un modo *stand-alone* (de una sola central telefónica) o hasta 1000 usuarios en una configuración múltiple SMB (varias centrales telefónicas interconectadas).

Tomando en cuenta los requisitos definidos en el análisis de requisitos, una implementación básica de IP Office en su versión 9.0 requiere de:

- Infraestructura de red (*networking*) y cableado estructurado.
- Chasis IP Office 500 V2.
- Licenciamiento *Essential Edition* (mínimo).
- Tarjetería y licenciamiento de usuarios y enlaces troncales.
- Teléfonos físicos o lógicos (*softphones*) para los usuarios de la plataforma.
- Licenciamiento *CTI Link Pro*.
- Configuración de permisos, códigos o planes de marcación que posibiliten la generación y recepción de llamadas telefónicas en el *call center*.
- Servicio de telefonía (acceso a la PSTN) a través de un proveedor de servicios con la funcionalidad de identificación de llamadas habilitada.

2.3.4.3.3. Servidor CTI (IA-AA-CTI)

Este servidor estará a cargo de la interacción existente entre los usuarios del servicio (agentes) y la central telefónica Avaya IP Office a través del *driver TAPILink Pro*. Sus principales funcionalidades serán:

- Enviar comandos (solicitudes) a la central telefónica Avaya: llamar, colgar, transferir, establecer estado de no molestar, etc. para cada una de las extensiones que componen el *call center*.
- Recibir eventos desde la central telefónica IP Office: cambios de estado de las llamadas (timbrando, conectada, ocupada, etc.) o del estado de las líneas telefónicas que son parte del *call center*.

El servidor CTI (o controlador CTI en este proyecto) utiliza la Interfaz de Programación de Aplicaciones de Telefonía (TAPI) para IP Office provisto gratuitamente por la empresa Avaya en el instalador ***USER4_version_subversion.exe***. El software a utilizarse en este proyecto será: ***USER4_2_43.exe***

Las versiones de la Interfaz de Programación de Aplicaciones de Telefonía soportadas por el fabricante son: TAPI 2.0 y TAPI 3.0, cada una de las cuales incluye ejemplos de programas (y su código fuente) que favorecen el aprendizaje y aceleran el desarrollo de aplicaciones CTI. Tomando en cuenta el análisis de las mencionadas aplicaciones de ejemplo como referencia, los problemas de estabilidad presentados con la versión 3.0 (específicamente en la transferencia de llamadas) fueron determinantes en la elección de la interfaz. Adicionalmente, el hecho de que Microsoft no soporte más la versión 3.0 desde código gestionado (según el artículo oficial <http://support.microsoft.com/default.aspx?scid=kb;en-us;841712>) determina que no es posible utilizar las facilidades que proveen lenguajes de programación como C# o Visual Basic de forma nativa, siendo un punto más en contra de TAPI 3.0.

La tabla 2.6 muestra las ventajas y desventajas de cada una de las versiones de TAPI.

Versión TAPI	Ventajas	Desventajas
2.0	Basada en C/C++ Flexible. Ofrece un mayor conjunto de funcionalidades respecto a la versión 3.0.	Requiere de un mayor grado de conocimiento. Aumento en la complejidad.
3.0	Implementada como un conjunto de objetos COM. Desarrollo desde lenguajes como C#, Visual Basic o C/C++ Sencilla.	Encapsulamiento de funcionalidades que restringen la utilización de características avanzadas o específicas. No soportado desde código gestionado ¹⁰ . Problemas de estabilidad.

Tabla 2.6 Comparación entre las Interfaces TAPI 2.0 y 3.0

Por tal motivo, aún a costa de aumentar la complejidad de desarrollo del módulo CTI del sistema, se ha elegido la versión 2.0 que provee una mayor “granularidad” en el desarrollo de aplicaciones y un mejor soporte y estabilidad de la plataforma.

Los requisitos de esta versión son los siguientes:

- Servidor con un sistema operativo Windows de 32 o 64 bits. Las versiones de sistema operativo soportadas son Windows XP, Windows Vista, Windows 7, Windows Server 2003 y Windows Server 2008 (R2).
- La documentación del fabricante no especifica requisitos de hardware específicos para la utilización del API.

Los pasos de instalación y configuración inicial del *driver* de Avaya están descritos en el documento “*IP Office, TAPI Link Installation*”^[15].

Por otra parte, la información relacionada con el API de Telefonía está detallada en el documento “*IP Office, TAPILink Developer’s Guide*”^[16].

¹⁰ Código gestionado (*managed code*) es un término acuñado por Microsoft para referirse al código de un programa que es ejecutado sobre una máquina virtual CLR (*Common Language Runtime*).

2.3.4.3.4. Servidor Principal (IA-AA-CORE)

Tomando en cuenta lo establecido en el análisis de requisitos de la fase de inicio (IA-AR-4), este servidor multiplataforma es el encargado de recibir e interconectar las bases de datos, servidor CTI y usuarios, constituyéndose como el componente central y de mayor complejidad del sistema. Entre sus principales atribuciones estarán:

- Autenticar y recibir comandos y peticiones de los usuarios (agentes, supervisores y administradores).
- Transferir los comandos relacionados con las llamadas telefónicas o las extensiones al servidor CTI.
- Ingresar y actualizar la información de usuarios, parámetros de sistema, planes de numeración, registros de llamadas, etc. en la base de datos propia del aplicativo.
- Recuperar la información de los contactos desde bases de datos externas.

La figura 2.11 muestra el papel centralizado de este componente en la arquitectura de la solución.

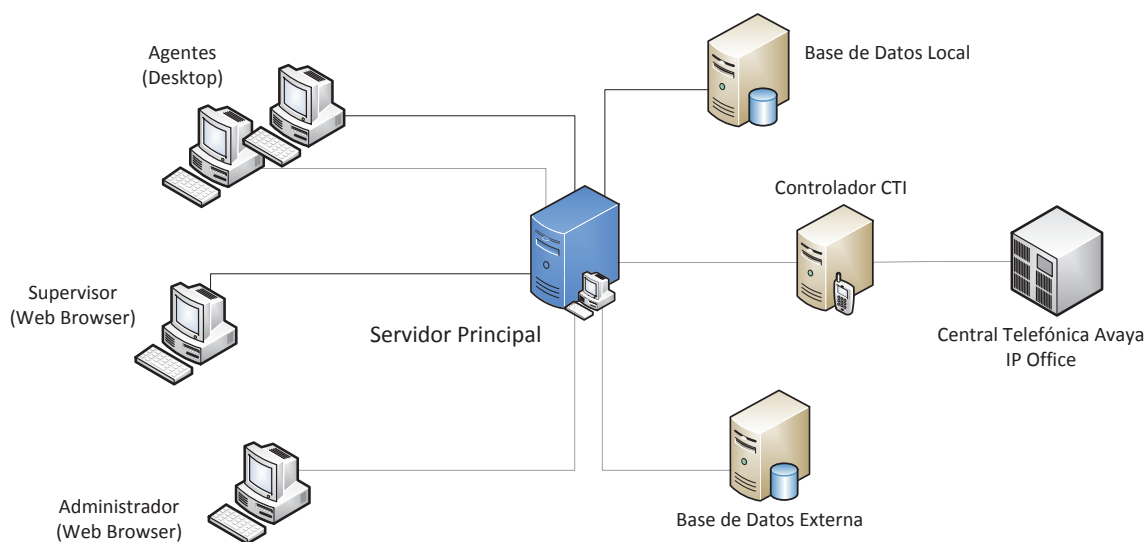


Figura 2.11 Servidor Principal y Componentes con los que interactúa

En la actualidad el *framework* de desarrollo y ejecución de aplicaciones distribuidas multiplataforma más conocido es Java EE, que al momento se encuentra en la versión 7.0 (estable) y cuyas principales características son^[9]:

- **Multiplataforma.** La tecnología Java permite que las aplicaciones escritas en este lenguaje sean ejecutadas desde cualquier sistema operativo compatible (Windows, Linux, Unix, MacOs) a través de la Máquina Virtual de Java (JVM).
- **Plataforma de bajo coste.** La instalación del *framework* es gratuita, así como algunas herramientas de desarrollo y servidores de aplicaciones.
- **Variedad de tecnologías, servicios y componentes.** El enfoque empresarial de esta solución ha contribuido en la creación de APIs (especificaciones) que facilitan el desarrollo de aplicaciones, entre las que se puede mencionar: EJBs, JSFs, JDBC, JPA, etc.
- **Vasta documentación y ejemplos.** La amplia aceptación promueve la creación de material de apoyo, desde tutoriales hasta ejemplos, los cuales pueden ser encontrados fácilmente en Internet.

2.4. FASE DE ELABORACIÓN (E¹¹)

El principal objetivo de la fase de elaboración es el de establecer y validar la arquitectura del sistema, utilizando como guía los requisitos, análisis y arquitecturas planteadas en la fase de inicio (I). Para ello será necesaria la refinación de los requisitos, creación de diagramas, tablas y modelos que permitan describir de forma general la interacción y operación de cada uno de los componentes que conformarán la plataforma.

Entradas de Esta Fase

Los modelos de negocio y casos de uso, así como el análisis de requerimientos procedentes de la fase de inicio serán utilizados como entradas de esta fase:

¹¹ La letra "E" mayúscula tiene por objeto agrupar y representar a los distintos subelementos de la fase de Elaboración que posteriormente serán referenciados en otras fases.

- **Modelos de Negocio y Casos de Uso**

Los modelos (o procesos) de negocio y los diagramas de caso de uso planteados en la fase anterior (IA-AA) son vitales en el diseño de la solución, los cuales serán analizados a profundidad y serán derivados en diagramas, algoritmos o secuencias conforme a las necesidades del proyecto.

- **Análisis de Requerimientos**

El análisis de requerimientos (IA-AR), que en un primer tratamiento realizó un estudio inicial de los riesgos y necesidades del sistema sirve como punto de partida para la definición, diseño y dimensionamiento de los componentes que conformarán la arquitectura de la plataforma.

2.4.1. REQUISITOS (ER)

Si bien en la fase de inicio se determinaron los requisitos primordiales de creación de la plataforma, la fase de elaboración requiere del establecimiento de requisitos que permitan filtrar y refinar la información de la fase anterior para detallarla en casos de uso que pueden ser utilizados como base para el diseño de la solución.

2.4.1.1. Especificación de Casos de Uso de Requisitos (ER-ECU-R)

Esta sección tiene como propósito transformar los requisitos primarios (definidos en IA-CA-1) en casos de uso que posteriormente serán afinados (con el propósito de obtener las funcionalidades del sistema) y utilizados como base para la etapa de análisis. Debe tomarse en cuenta que no todos los requisitos primarios pueden ser convertidos en casos de uso (especialmente los requisitos técnicos o no funcionales), por lo que no serán incluidos en este estudio. De forma adicional, también será incluido y descrito un caso de uso correspondiente a un requisito secundario (funcional opcional) que se ha considerado importante en este proyecto. La principal ventaja de la especificación de los requisitos funcionales como casos de uso está en la determinación de los principales elementos, actores y procedimientos involucrados en la creación y operación del sistema, que serán explicados en las secciones de análisis y diseño de esta fase (EA y ED).

2.4.1.1.1. Agente

Especificación de Caso de Uso: Gestión y Control de Llamadas Telefónicas	
ID	ER-ECU-R-A1
ID Requisito	IR-RF-O1
Nombre	Gestión y Control de Llamadas Telefónicas
Descripción	El agente administrará el estado de la extensión telefónica y podrá controlar las llamadas procesadas.
Autor	Diego Chacón Herrera
Actor	Agente
Condiciones Previas	El actor debe poseer los recursos físicos (teléfono, PC, <i>softphone</i> , etc.) y lógicos (cuentas y credenciales de acceso, permisos de marcación y <i>call center</i> , etc.) necesarios para la recepción y generación de llamadas. El actor debe haber ingresado en el sistema.
Condiciones Resultantes	El agente controla y manipula los eventos de la línea telefónica o del teléfono desde el computador.
Flujo Normal de Eventos	
Recepción de Eventos	
<ol style="list-style-type: none"> 1. La línea telefónica del agente genera un evento (recibe una llamada telefónica, por ejemplo). La central telefónica detecta dichos eventos y los envía al servidor CTI a través de TAPI. 2. El servidor CTI reenvía la notificación al servidor de comunicaciones. 3. El servidor de comunicaciones procesa el evento y reenvía la notificación al aplicativo de escritorio del agente correspondiente. 4. El usuario utiliza el aplicativo del computador según las acciones a tomar sobre la línea (extensión). 	
Generación de Eventos	
<ol style="list-style-type: none"> 1. El agente a través del aplicativo de escritorio envía los comandos al servidor 	

<p>de comunicaciones (realización de una llamada, por ejemplo).</p> <ol style="list-style-type: none"> 2. El servidor de comunicaciones procesa el requerimiento del usuario que posteriormente es reenviado al servidor CTI. 3. El servidor CTI envía la solicitud a la central telefónica a través de TAPI. 4. Si la central telefónica contiene los recursos necesarios, procesa la solicitud relacionada con la línea telefónica en cuestión, cambiando su estado. <ol style="list-style-type: none"> a. Los eventos son procesados y almacenados en el sistema (servidor principal) para su posterior análisis en la generación de reportes o estadísticas. 	
<p>Excepciones</p>	
<p>Error de Comunicación</p> <p>En los pasos 1-3 del flujo normal de recepción y generación de eventos:</p> <ol style="list-style-type: none"> 1. El sistema no puede retransmitir/procesar el evento generado. 2. El aplicativo del agente permanece sin cambios. 	
<p>Evento no Procesado</p> <p>En el 4 del flujo normal de Generación de Eventos:</p> <ol style="list-style-type: none"> 1. La central telefónica no puede procesar la solicitud. 2. Manejo de eventos: <ol style="list-style-type: none"> a. Si la central telefónica responde con un mensaje de error, éste será retransmitido hasta el usuario final. b. En caso de no generarse un mensaje de error, el aplicativo del agente permanece sin cambios. 	
<p>Anotaciones</p>	<p>Los actores podrán realizar las siguientes acciones sobre una llamada: Llamar, colgar, contestar, retener (en espera), recuperar, transferir, no disponible.</p>

Tabla 2.7 Caso de Uso – Gestión y Control de Llamadas Telefónicas

Especificación de Caso de Uso: Despliegue de Información de los Contactos en una Llamada	
ID	ER-ECU-R-A2
ID Requisito	IR-RF-O2
Nombre	Despliegue de Información de los Contactos en una Llamada.
Descripción	El agente visualizará la información del contacto (cliente) recuperada desde una base de datos en el transcurso de una llamada telefónica.
Autor	Diego Chacón Herrera
Actor	Agente
Condiciones Previas	<p>Debe estar habilitado el servicio de identificación de llamadas en la institución.</p> <p>El agente debe disponer del hardware y software necesario para la gestión de llamadas.</p> <p>El actor debe estar registrado y activo en el sistema.</p>
Condiciones Resultantes	El agente visualiza la información del cliente atendido en una llamada.
Flujo Normal de Eventos	
<ol style="list-style-type: none"> 1. El usuario recibe, realiza, retiene, recupera o transfiere una llamada telefónica. 2. Con base en el número telefónico del contacto (cliente) participante en la comunicación, el sistema busca la información desde una base de datos de la organización. 3. Si existen coincidencias, el sistema despliega en la pantalla del agente la información de todas las coincidencias encontradas. 4. El agente procesa la llamada con base en la información previamente obtenida. 	
Flujos Alternos	
<p>En el paso 3 del flujo normal de eventos:</p> <ol style="list-style-type: none"> 1. El sistema no encuentra ninguna coincidencia. 2. El agente atiende la llamada sin la información del cliente. 	

Tabla 2.8 Caso de Uso – Despliegue de Información de los Contactos en una Llamada

Especificación de Caso de Uso: Gestión e ingreso de las observaciones de las llamadas	
ID	ER-ECU-R-A3
ID Requisito	IR-RF-O3
Nombre	Gestión e ingreso de las observaciones de las llamadas.
Descripción	En el transcurso de una llamada, el agente ingresará observaciones que serán procesadas por el supervisor.
Autor	Diego Chacón Herrera
Actor	Agente
Condiciones Previas	El agente debe disponer del hardware y software necesario para la gestión de llamadas. El actor debe estar registrado y activo en el sistema. El agente debe tener una llamada telefónica en proceso.
Condiciones Resultantes	El agente registra las observaciones de una llamada.
Flujo Normal de Eventos	
<ol style="list-style-type: none"> 1. El actor atiende una llamada telefónica. 2. El agente selecciona la opción de ingreso de observaciones para la llamada en curso. 3. El actor acepta los datos anotados. 4. El sistema guarda la información y la almacena para ser utilizada por el supervisor. 	
Flujos Alternos	
<p>En el paso 3 del flujo normal de eventos:</p> <ol style="list-style-type: none"> 1. El agente cancela el ingreso de las observaciones. 	

Tabla 2.9 Caso de Uso – Gestión e Ingreso de Observaciones de las Llamadas

Especificación de Caso de Uso: Detalle del Número Telefónico de Contacto en una Llamada Externa	
ID	ER-ECU-R-A4
ID Requisito	IR-RF-O4
Nombre	Detalle del Número Telefónico de Contacto en una Llamada Externa
Descripción	En el procesamiento de una llamada externa (entrante o saliente), el agente podrá visualizar información detallada respecto al número telefónico, como el nombre de la operadora o la provincia.
Autor	Diego Chacón Herrera
Actor	Agente
Condiciones Previas	El agente debe disponer del hardware y software necesario para la gestión de llamadas. El actor debe estar registrado y activo en el sistema. El actor debe procesar una llamada telefónica.
Condiciones Resultantes	El agente visualiza el nombre de la operadora e información relacionada en el transcurso de una llamada telefónica externa.
Flujo Normal de Eventos	
<ol style="list-style-type: none"> 1. El actor atiende una llamada telefónica. 2. Con base en el número telefónico del contacto se busca información adicional: <ol style="list-style-type: none"> a. Para números telefónicos fijos se visualizarán los nombres de la operadora, provincia y nodo (central telefónica). b. Para números telefónicos móviles se presentará el nombre de la operadora¹². 3. El sistema registra la información procesada para utilizarla en la generación de reportes o estadísticas. 4. El agente utiliza la información desplegada para ofrecer una mejor atención a los usuarios. 	
Flujos Alternos	
<p>En el paso 2 del flujo normal de eventos:</p> <ol style="list-style-type: none"> 1. El sistema no puede recabar información detallada del número telefónico. 2. Ninguna acción se toma al respecto. 	

Tabla 2.10 Caso de Uso – Detalle del Número Telefónico de Contacto en una Llamada Externa

¹² La portabilidad numérica no será incluida ya que esa información no es de acceso público.

Especificación de Caso de Uso: Integración con Aplicaciones Externas	
ID	ER-ECU-R-A5
ID Requisito	IR-RF-O5
Nombre	Integración con Aplicaciones Externas
Descripción	En el transcurso de una llamada, el agente ejecutará (invocará) aplicaciones de terceros que presenten información detallada del cliente.
Autor	Diego Chacón Herrera
Actor	Agente
Condiciones Previas	El agente debe disponer del hardware y software necesario para la gestión de llamadas. El actor debe estar registrado y activo en el sistema. El actor debe tener una llamada en proceso.
Condiciones Resultantes	
Flujo Normal de Eventos	
<ol style="list-style-type: none"> 1. El actor atiende una llamada telefónica. 2. El agente selecciona la opción de invocación de aplicaciones externas desde la interfaz de gestión de llamadas. 3. La aplicación externa recibe los parámetros enviados para buscar y desplegar la información del cliente. 4. El agente utiliza información de la aplicación invocada para atender la llamada. 	
Flujos Alternos	
<p>En el paso 3 del flujo normal de eventos:</p> <ol style="list-style-type: none"> 1. La aplicación externa no entiende o no procesa los parámetros recibidos. 2. Ninguna acción es tomada al respecto. 	

Tabla 2.11 Caso de Uso – Integración con Aplicaciones Externas

2.4.1.1.2. *Administrador*

Especificación de Caso de Uso: Gestionar usuarios, servicios y comunicaciones	
ID	ER-ECU-R-M1
ID Requisito	IR-RF-O8
Nombre	Gestión de Reportes y/o Estadísticas
Descripción	El supervisor monitoreará y comprobará la actividad de los agentes que utilizan el sistema.
Autor	Diego Chacón Herrera
Actor	Supervisor
Condiciones Previas	El actor debe contar con el software y hardware necesario para cumplir su función. El administrador debe estar registrado en el sistema.
Condiciones Resultantes	El actor gestiona los parámetros del sistema según su conveniencia.
Flujo Normal de Eventos	
<ol style="list-style-type: none"> 1. El actor selecciona la opción a gestionar. 2. El sistema despliega la interfaz de administración o configuración. 3. El actor realiza los cambios/configuraciones necesarias. 4. El sistema configura, crea, modifica o elimina los registros solicitados. 	

Tabla 2.12 Caso de Uso – Gestionar Usuarios, Servicios y Comunicaciones

Debe considerarse que el caso de uso anterior, por su carácter muy general no describe flujos alternos o excepciones.

2.4.1.1.3. Supervisor

Especificación de Caso de Uso: Monitoreo y Supervisión de Agentes	
ID	ER-ECU-R-S1
ID Requisito	IR-RF-O6
Nombre	Monitoreo y Supervisión de Agentes
Descripción	El supervisor monitoreará y comprobará la actividad de los agentes que utilizan el sistema.
Autor	Diego Chacón Herrera
Actor	Supervisor
Condiciones Previas	El actor debe contar con el software y hardware necesario para cumplir su función. El supervisor debe estar registrado en el sistema.
Condiciones Resultantes	El actor visualiza los eventos de los agentes del <i>call center</i> .
Flujo Normal de Eventos	
<ol style="list-style-type: none"> 1. El actor selecciona la opción de monitoreo de agentes. 2. El sistema recopila los eventos generados por los agentes. 3. El sistema presenta la información recabada al supervisor. 	
Flujos Alternos	
<p>En el paso 2 del flujo normal de eventos:</p> <ol style="list-style-type: none"> 1. No existen eventos generados en el <i>call center</i>. 2. La pantalla principal no muestra actividad alguna al supervisor. 	

Tabla 2.13 Caso de Uso – Monitoreo y Supervisión de Agentes

Especificación de Caso de Uso: Gestión de Reportes y/o Estadísticas	
ID	ER-ECU-R-S2
ID Requisito	IR-RF-O7
Nombre	Gestión de Reportes y/o Estadísticas
Descripción	El supervisor monitoreará y comprobará la actividad de los agentes que utilizan el sistema.
Autor	Diego Chacón Herrera
Actor	Supervisor
Condiciones Previas	El actor debe contar con el software y hardware necesario para cumplir su función. El supervisor debe estar registrado en el sistema.
Condiciones Resultantes	El supervisor visualiza el reporte o la estadística requerida.
Flujo Normal de Eventos	
<ol style="list-style-type: none"> 1. El actor selecciona la opción de gestión de reportes y/o estadísticas. 2. El actor ingresa los parámetros que delimitan la generación del reporte/estadística. 3. El sistema recupera la información solicitada de su base de datos con base en los parámetros establecidos. 4. La aplicación despliega la información solicitada. 	
Flujos Alternos	
<p>En el paso 3 del flujo normal de eventos:</p> <ol style="list-style-type: none"> 1. El sistema no encuentra información coincidente con los parámetros de búsqueda. <ul style="list-style-type: none"> • Se presenta un reporte vacío al supervisor. 2. El sistema no puede conectarse a la base de datos. <ul style="list-style-type: none"> • Se despliega una notificación de error. 	

Tabla 2.14 Caso de Uso – Gestión de Reportes y/o Estadísticas

2.4.1.2. Refinamiento de Casos de Uso de Funcionalidades (ER-RCU-F)

Si bien en el apartado anterior fueron detallados los requisitos primarios como casos de uso, todavía no ha sido claramente establecida la interacción de la plataforma con los usuarios (y viceversa); es decir, tomando el caso del agente como ejemplo, aún no ha sido descrito apropiadamente el procedimiento (entiéndase flujo) necesario para la realización de una llamada o de una transferencia. En esta disciplina de proceso, por tanto, se especificarán (detallarán) los casos de uso relacionados con las funcionalidades del sistema para cada uno de los roles establecidos, tomando como base la especificación de los casos de uso de la sección anterior (ER-ECU-R) y algunos elementos detallados en la fase de inicio (IR y AR).

2.4.1.2.1. Global

Esta parte detalla los casos de uso comunes para todos los usuarios o aquellos que repercuten en el funcionamiento general de la plataforma.

- **Inicio y Cierre de Sesión en el Sistema**

Dado que la solución planteada gestiona perfiles (datos del usuario) y roles (agente, supervisor o administrador), está implícito el respectivo inicio y cierre de sesión que asignará y liberará recursos del sistema. Es por ello que a pesar de no haber sido definido como un caso de uso funcional o como un requerimiento inicial, debe ser tratado, diseñado y documentado debidamente.

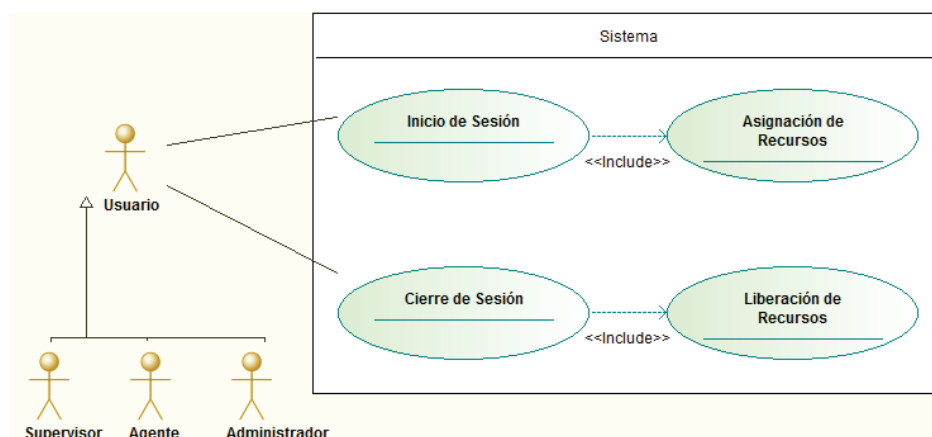


Figura 2.12 Caso de Uso – Inicio y Cierre de Sesión

Refinamiento de Caso de Uso: Inicio de Sesión (Autenticar Usuario)	
ID	ER-RCU-F-G1
Nombre	Autenticar Usuario
Descripción	Autenticar y validar un usuario registrado en el sistema.
Autor	Diego Chacón Herrera
Actor	Todos los usuarios (agentes, supervisores y administradores)
Condiciones Previas	<p>Todos los usuarios deben haber sido creados previamente en el sistema.</p> <p>Agentes</p> <ul style="list-style-type: none"> • El agente tiene instalado el aplicativo de escritorio correspondiente. • Los parámetros de comunicación con el servidor han sido previamente configurados. <p>Supervisores y administradores</p> <ul style="list-style-type: none"> • Poseer navegadores web compatibles.
Condiciones Resultantes	El usuario es validado e inicia sesión en el sistema.
Flujo Normal de Eventos	
<p>Validar:</p> <ol style="list-style-type: none"> 1. El actor ingresa las credenciales de acceso. 2. El sistema valida los datos ingresados por el usuario. Los recursos son asignados en función del rol al que corresponde. <ol style="list-style-type: none"> a. En caso de tratarse de un agente, comprobar si existe una sesión previa activa en el sistema. En caso de haberlo, notificar al usuario que no es posible iniciar sesión por tal motivo. 3. Una vez validado el usuario, la pantalla se actualiza mostrando la información relevante para cada actor. 	
Flujos Alternos	
Nombre de usuario no encontrado	
<ol style="list-style-type: none"> 1. El sistema muestra un mensaje que indica que el usuario no ha sido 	

encontrado.	
Contraseña incorrecta	
1. El sistema informa al actor que la contraseña ingresada no es correcta.	
Excepciones	
Anotaciones	Para los agentes sólo puede haber una sesión activa a la vez (un usuario no puede iniciar sesión si hay otra activa con las mismas credenciales).

Tabla 2.15 Caso de Uso – Autenticar Usuario (Inicio de Sesión)

Refinamiento de Caso de Uso: Cierre de Sesión	
ID	ER-RCU-F-G2
Nombre	Cierre de Sesión
Descripción	Cerrar una sesión activa en el sistema.
Autor	Diego Chacón Herrera
Actor	Todos los usuarios (agentes, supervisores y administradores)
Condiciones Previas	El usuario en cuestión debe haber iniciado sesión en el sistema.
Condiciones Resultantes	Liberación de los recursos asociados a ese usuario y cierre de sesión.
Flujo Normal de Eventos	
<ol style="list-style-type: none"> 1. El actor selecciona la opción de cierre de sesión. 2. El sistema libera los recursos asociados. 3. Se envía una notificación al usuario respecto al evento. 	
Flujos Alternos	
Sesión no Iniciada	
<ol style="list-style-type: none"> 1. El usuario no ha iniciado sesión o no tiene una sesión activa en el sistema. Se desplegará un mensaje informativo. 	

Tabla 2.16 Caso de Uso – Cierre de Sesión

2.4.1.2.2. *Agente*

- **Llamadas Telefónicas**

Refinamiento de Caso de Uso: Realizar Llamadas	
ID	ER-RCU-F-A1
Requisito Base	ER-ECU-R-A1
Nombre	Realizar Llamadas
Descripción	El agente realizará una llamada.
Autor	Diego Chacón Herrera
Actor	Agente
Condiciones Previas	El agente debe haber iniciado sesión.
Condiciones Resultantes	Llamada realizada.
Flujo Normal de Eventos	
<ol style="list-style-type: none"> 1. El agente selecciona la opción (a través del botón o del menú) de realización de llamada. 2. El sistema despliega un cuadro de diálogo para el ingreso del número telefónico. 3. El usuario ingresa el número telefónico al que desea marcar y presiona el botón de llamada. 4. El sistema procesa el número y solicita la generación de la llamada a la central telefónica. 5. Se realiza la llamada. 6. El sistema muestra los eventos de la llamada en pantalla. 	
Flujos Alternos	

<p>Cancelar</p> <p>En el paso 2 del flujo normal de eventos:</p> <ol style="list-style-type: none"> 1. El usuario presiona el botón Cancelar para cerrar el cuadro de diálogo. La llamada no se realiza. <p>Número inválido</p> <p>En el paso 3 del flujo normal de eventos:</p> <ol style="list-style-type: none"> 1. El usuario ingresa un número de teléfono no válido y presiona el botón de llamada. 2. El sistema detecta un número inválido y muestra un mensaje de información del error. 	
<p>Excepciones</p>	
<p>Sin respuesta</p> <p>En el paso 4 del flujo normal de eventos:</p> <ol style="list-style-type: none"> 1. La central telefónica no puede realizar la llamada. 2. No se genera la llamada. <p>Sin conexión</p> <p>En el paso 4 del flujo normal de eventos :</p> <ol style="list-style-type: none"> 1. Pérdida de conectividad de cualquier componente del sistema. 2. La llamada no se realiza. 	
<p>Anotaciones</p>	<p>El número telefónico a marcar debe tener el mismo formato como si fuera digitado en el teléfono, incluidos planes de numeración y códigos de autorización, determinados en la central telefónica y configurados por el administrador.</p>

Tabla 2.17 Caso de Uso – Realizar Llamada

Refinamiento de Caso de Uso: Colgar llamadas	
ID	ER-RCU-F-A2
Requisito Base	ER-ECU-R-A1
Nombre	Colgar llamadas
Descripción	El agente cuelga una llamada en proceso.
Autor	Diego Chacón Herrera
Actor	Agente
Condiciones Previas	El agente debe haber iniciado sesión y debe tener una llamada activa ¹³ en el sistema.
Condiciones Resultantes	La llamada se interrumpe.
Flujo Normal de Eventos	
<ol style="list-style-type: none"> 1. En el transcurso de una llamada activa el agente cuelga la llamada. 2. El sistema detecta el evento generado por el usuario y cuelga la llamada. 3. La llamada se interrumpe. 4. El sistema actualiza la interfaz acorde al cierre de la llamada. 	
Flujos Alternos	
Acción no disponible	
<ol style="list-style-type: none"> 1. La llamada en proceso posee un estado que impide colgarla, por lo que la acción no es habilitada. 2. La llamada permanece activa. 	
Excepciones	
Sin conexión	
En el paso 2 del flujo normal de eventos :	
<ol style="list-style-type: none"> 1. Pérdida de conectividad de cualquier componente del sistema. 2. La llamada permanece activa. 	
Anotaciones	La línea debe estar configurada como estación descolgada (<i>off-hook station</i>) en la central telefónica.

Tabla 2.18 Caso de Uso – Colgar Llamada

¹³ En el contexto de este esquema, una llamada activa corresponde a cualquier estado de la línea telefónica distinto de desconectado o inactivo.

Refinamiento de Caso de Uso: Contestar Llamadas	
ID	ER-RCU-F-A3
Requisito Base	ER-ECU-R-A1
Nombre	Contestar llamadas
Descripción	El agente contesta una llamada entrante.
Autor	Diego Chacón Herrera
Actor	Agente
Condiciones Previas	El agente debe haber iniciado sesión y debe tener una llamada entrante en proceso (timbrando).
Condiciones Resultantes	La interfaz actualiza el estado de la llamada y se levanta el altavoz del teléfono.
Flujo Normal de Eventos	
<ol style="list-style-type: none"> 1. En el timbrado de una llamada entrante (RINGING), el agente elige la opción de contestar llamada. 2. El sistema detecta el evento generado por el usuario y contesta la llamada a través del altavoz del teléfono. 3. La llamada es conectada. 4. El sistema actualiza la interfaz acorde al evento de la llamada. 	
Flujos Alternos	
Acción no disponible	
<ol style="list-style-type: none"> 1. La llamada cambia su estado antes que el agente seleccione la opción adecuada. 2. La llamada no es contestada por el altavoz. 	
Excepciones	
Sin conexión	
En el paso 2 del flujo normal de eventos :	
<ol style="list-style-type: none"> 1. Pérdida de conectividad de cualquier componente del sistema. 2. El estado de la llamada permanece sin cambios. 	
Anotaciones	Sólo se puede contestar por altavoz en teléfonos Avaya.

Tabla 2.19 Caso de Uso – Contestar Llamada

Refinamiento de Caso de Uso: Retener (sostener) llamadas	
ID	ER-RCU-F-A4
Requisito Base	ER-ECU-R-A1
Nombre	Retener (sostener) llamadas
Descripción	El agente retiene una llamada activa.
Autor	Diego Chacón Herrera
Actor	Agente
Condiciones Previas	El agente debe haber iniciado sesión y debe tener una llamada conectada (intercambio de audio) en el sistema.
Condiciones Resultantes	La llamada cambia su estado a retenida (HOLD).
Flujo Normal de Eventos	
<p>En el transcurso de una llamada activa:</p> <ol style="list-style-type: none"> 1. El agente selecciona la opción de retener llamada. 2. El sistema detecta el evento y procesa la petición del usuario. 3. La llamada cambia su estado a retenida (HOLD). 4. La interfaz se actualiza para reflejar los cambios, habilitando la acción de recuperación (UNHOLD) de la llamada. 	
Flujos Alternos	
Acción no disponible	
<ol style="list-style-type: none"> 1. La llamada no está conectada, por lo que la acción no es habilitada. 2. La llamada permanece activa. 	
Excepciones	
Sin conexión	
<p>En el paso 2 del flujo normal de eventos :</p> <ol style="list-style-type: none"> 1. Pérdida de conectividad de cualquier componente del sistema. 2. La llamada permanece activa. 	
Anotaciones	El estado HOLD de la llamada es utilizada para recuperarla (UNHOLD)

Tabla 2.20 Caso de Uso – Retener Llamada

Refinamiento de Caso de Uso: Recuperar Llamadas	
ID	ER-RCU-F-A5
Requisito Base	ER-ECU-R-A1
Nombre	Recuperar llamadas
Descripción	El agente recupera una llamada sostenida.
Autor	Diego Chacón Herrera
Actor	Agente
Condiciones Previas	El agente debe haber iniciado sesión y debe tener una llamada retenida en el sistema.
Condiciones Resultantes	La llamada cambia su estado a conectada (CONNECTED).
Flujo Normal de Eventos	
<ol style="list-style-type: none"> 1. El agente selecciona la acción de recuperación de la llamada retenida. 2. El sistema detecta el evento generado y procesa la solicitud. 3. La llamada cambia su estado a conectada. <ol style="list-style-type: none"> a. Si había otra llamada con estado conectado, ésta pasa a estar retenida. 4. La interfaz se actualiza para reflejar los cambios, habilitando la opción de retener (HOLD) la llamada vigente. 	
Flujos Alternos	
Excepciones	
Sin conexión	
En el paso 2 del flujo normal de eventos :	
<ol style="list-style-type: none"> 1. Pérdida de conectividad de cualquier componente del sistema. 2. La llamada permanece activa. 	
Anotaciones	El agente puede seleccionar entre varias llamadas retenidas y recuperarlas a discreción.

Tabla 2.21 Caso de Uso – Recuperar Llamada Retenida

Refinamiento de Caso de Uso: Transferir llamadas	
ID	ER-RCU-F-A6
Requisito Base	ER-ECU-R-A1
Nombre	Transferir llamadas
Descripción	El agente transfiere una llamada conectada.
Autor	Diego Chacón Herrera
Actor	Agente
Condiciones Previas	El agente debe haber iniciado sesión y debe tener una llamada conectada en el sistema.
Condiciones Resultantes	La llamada es transferida.
Flujo Normal de Eventos	
<ol style="list-style-type: none"> 1. En una llamada conectada, el agente selecciona la opción Transferir llamada. 2. El sistema despliega un cuadro de diálogo para el ingreso del número de extensión al que se transferirá la llamada. <ol style="list-style-type: none"> a. Si la transferencia es ciega¹⁴, el usuario hará clic en el botón correspondiente. b. Si la transferencia es asistida¹⁵, el agente seleccionará esta opción. <ol style="list-style-type: none"> i. La llamada en curso (inicial) cambiará a un estado retenido (HOLD). ii. Se efectuará una llamada (consultiva) al número al que se desea transferir. iii. Si el usuario (consultado) aprueba la transferencia, el agente selecciona el botón que completa el proceso. iv. La llamada consultiva es finalizada. 3. La transferencia se realiza y la llamada original es finalizada (colgada). 4. La interfaz se actualiza para reflejar los cambios. 	

¹⁴ Transferencia ciega es aquella que se realiza de forma directa y sin intervención del usuario al que se transfiere la llamada.

¹⁵ Transferencia asistida es la que requiere de la aprobación de la persona que recibirá la llamada para efectuar la transferencia.

Flujos Alternos	
Usuario no Aprueba la Transferencia	
En el paso 2.b.ii del flujo normal de eventos:	
<ol style="list-style-type: none"> 1. El contacto (consultado) al que se desea realizar la llamada rechaza la transferencia (no contesta, no está disponible, no aprueba la llamada, etc.). 2. La llamada consultiva es finalizada, pero no se realiza ninguna transferencia. 3. El agente selecciona la llamada original cuyo estado es retenido (HOLD) y la recupera mediante el procedimiento de caso de uso correspondiente para notificar los sucesos al cliente. 4. La interfaz es actualizada para notificar los cambios. 	
Excepciones	
Sin conexión	
En el paso 2 del flujo normal de eventos (incluidos los subflujos correspondientes):	
<ol style="list-style-type: none"> 1. Pérdida de conectividad de cualquier componente del sistema. 2. La llamada original permanece activa y la transferencia no se realiza. 	
Anotaciones	Es importante tomar en cuenta que el sistema debe permanecer estable para cualquier número de transferencias que tengan que realizarse.

Tabla 2.22 Caso de Uso – Transferir Llamada

Refinamiento de Caso de Uso: Recepción de Eventos	
ID	ER-RCU-F-A7
Requisito Base	ER-ECU-R-A1
Nombre	Recepción de eventos relacionados con la línea telefónica.
Descripción	Caso de uso que describe el procedimiento de recepción de eventos relacionados con las líneas telefónicas de los agentes de un <i>call center</i> . En el contexto de este proyecto, un evento es el cambio de estado de una línea telefónica (como la recepción de una llamada, por ejemplo).
Autor	Diego Chacón Herrera

Actor	Agente
Condiciones Previas	El agente debe haber iniciado sesión en el sistema.
Condiciones Resultantes	El agente recibe la notificación de un evento relacionado a la línea telefónica.
Flujo Normal de Eventos	
<ol style="list-style-type: none"> 1. La central telefónica recibe, genera o procesa un evento de una línea telefónica particular. 2. A través de TAPI, la central telefónica notifica al servidor CTI el evento generado. 3. Este componente a su vez, transfiere la notificación al servidor principal. 4. Este servidor redirige la notificación al aplicativo de escritorio del agente correspondiente. 5. La interfaz del agente se actualiza para reflejar los cambios en el estado de la línea telefónica. 	
Flujos Alternos	
<p>En el paso 4 del flujo normal de eventos:</p> <ol style="list-style-type: none"> 1. El aplicativo del agente recibe información errónea y/o incompleta. 2. El mensaje es descartado y la interfaz permanece sin cambios. 	
Excepciones	
<p>Sin conexión</p> <p>En cualquiera de los pasos 2-4 del flujo normal de eventos:</p> <ol style="list-style-type: none"> 1. Pérdida de conectividad de cualquier componente del sistema. 2. El evento no es transferido, por lo que el agente desconoce del mismo y la interfaz permanece inalterada. 	

Tabla 2.23 Caso de Uso – Recepción de Eventos

- **Ingreso de Observaciones**

Refinamiento de Caso de Uso: Ingresar observaciones	
ID	ER-RCU-F-A8
Requisito Base	ER-ECU-R-A3
Nombre	Ingresar observaciones
Descripción	El agente ingresa observaciones respecto al transcurso de una llamada.
Autor	Diego Chacón Herrera
Actor	Agente
Condiciones Previas	El agente debe haber iniciado sesión y debe tener una llamada en proceso.
Condiciones Resultantes	Las observaciones deben ser visualizadas posteriormente por el supervisor.
Flujo Normal de Eventos	
<ol style="list-style-type: none"> 1. En el transcurso de una llamada, el agente selecciona la opción de ingreso de observaciones. 2. El sistema despliega un cuadro de diálogo para la anotación de observaciones. 3. El usuario ingresa el mensaje correspondiente. 4. El agente presiona el botón de confirmación y el cuadro de diálogo se cierra. <ol style="list-style-type: none"> a. El sistema recopila las observaciones y las almacena para un futuro procesamiento. 	
Flujos Alternos	
Cancelación	
En el paso 4 del flujo normal de eventos:	
<ol style="list-style-type: none"> 1. El agente presiona el botón de cancelación. 2. El cuadro de diálogo se cierra y el sistema no procesa los cambios. 	
Excepciones	
Sin conexión	
En el paso 4 del flujo normal de eventos (incluidos los subflujos correspondientes):	
<ol style="list-style-type: none"> 1. Pérdida de conectividad de cualquier componente del sistema. 2. Las observaciones no son almacenadas. 	

Tabla 2.24 Caso de Uso – Ingresar Observaciones

2.4.1.2.3. Supervisor

Refinamiento de Caso de Uso: Monitorear Agentes	
ID	ER-RCU-F-S1
Requisito Base	ER-ECU-R-S1
Nombre	Monitorear agentes
Descripción	El supervisor visualiza el estado de la línea de los agentes activos registrados en el sistema.
Autor	Diego Chacón Herrera
Actor	Supervisor
Condiciones Previas	El supervisor y los agentes deben haber iniciado sesión en el sistema.
Condiciones Resultantes	Presentación de estado de agentes activos en la interfaz web.
Flujo Normal de Eventos	
<ol style="list-style-type: none"> 1. En una sesión activa, el supervisor selecciona la opción de monitoreo de agentes. 2. El sistema despliega una tabla con la principal información de los agentes: <ol style="list-style-type: none"> a. Nombre de usuario b. Extensión asociada c. Estado de la línea (marcando, timbrando, conectado, etc.) d. Número llamante e. Número llamado 	
Flujos Alternos	
Sin Agentes	
En el paso 2 del flujo normal de eventos:	
<ol style="list-style-type: none"> 1. No hay agentes conectados al sistema que obvia el monitoreo del <i>call center</i>. 	
Excepciones	
Error de conexión	
En el paso 1 del flujo normal de eventos:	
<ol style="list-style-type: none"> 1. El sistema pierde conectividad con el gestor de llamadas o la central telefónica, interrumpiendo el registro de eventos de los agentes y por lo tanto el monitoreo de <i>call center</i>. 	

Tabla 2.25 Caso de Uso – Monitoreo de Agentes

Refinamiento de Caso de Uso: Envío de Mensajes a los Agentes	
ID	ER-RCU-F-S2
Requisito Base	ER-ECU-R-S1
Nombre	Envío de mensajes a los agentes.
Descripción	El supervisor envía un mensaje o una notificación particular a los agentes del <i>call center</i> .
Autor	Diego Chacón Herrera
Actor	Supervisor
Condiciones Previas	El supervisor y los agentes deben haber iniciado sesión en el sistema. El supervisor debe estar en la pantalla de monitoreo de agentes.
Condiciones Resultantes	El agente debe visualizar el mensaje creado por el supervisor.
Flujo Normal de Eventos	
<ol style="list-style-type: none"> 1. En la pantalla de monitoreo de agentes, el supervisor, para un agente particular, selecciona la opción de envío de mensaje. 2. El sistema despliega un cuadro de diálogo para el ingreso del mensaje. 3. El supervisor ingresa el mensaje y presiona el botón de confirmación. 4. El sistema envía el mensaje al agente concerniente. 	
Flujos Alternos	
Cancelación	
En el paso 3 del flujo normal de eventos:	
<ol style="list-style-type: none"> 1. El supervisor presiona el botón de cancelación. 2. El cuadro de diálogo se cierra sin enviar ningún mensaje al agente en cuestión. 	
Excepciones	
Error de conexión	
En el paso 4 del flujo normal de eventos:	
<ol style="list-style-type: none"> 1. El agente o el supervisor pierden conectividad con el sistema, imposibilitando el envío o la recepción de notificaciones. 	
Anotaciones	Únicamente los supervisores son capaces de enviar mensajes a los agentes.

Tabla 2.26 Caso de Uso – Envío de Mensajes a los Agentes

Refinamiento de Caso de Uso: Visualización de Registros, Reportes o Estadísticas	
ID	ER-RCU-F-S3
Requisito Base	ER-ECU-R-S2
Nombre	Visualización de registros, reportes o estadísticas.
Descripción	El supervisor envía un mensaje o una notificación particular a los agentes del <i>call center</i> .
Autor	Diego Chacón Herrera
Actor	Supervisor
Condiciones Previas	El supervisor y los agentes deben haber iniciado sesión en el sistema. El supervisor debe estar en la pantalla de monitoreo de agentes.
Condiciones Resultantes	Presentación de reporte/estadística en la interfaz web del supervisor.
Flujo Normal de Eventos	
<ol style="list-style-type: none"> 1. El supervisor, dentro de la sección Reportes, selecciona el nombre del reporte o la estadística que desea visualizar. 2. De ser necesario, el supervisor ingresa los parámetros que delimitan la información que será presentada. 3. El supervisor selecciona el botón que creará el reporte/estadística. 4. La interfaz se actualiza para mostrar los datos solicitados. 	
Flujos Alternos	
Cancelación	
En el paso 3 del flujo normal de eventos: <ol style="list-style-type: none"> 1. El supervisor sale de la interfaz seleccionando otro menú/opción. 2. El reporte/estadística no es generado. 	
Excepciones	
Sin información	

<p>En el paso 4 del flujo normal de eventos:</p> <ol style="list-style-type: none"> 1. El sistema no dispone o no puede acceder a la información solicitada. 2. No se muestra información alguna. <p>Error de conexión</p> <p>En el paso 4 del flujo normal de eventos:</p> <ol style="list-style-type: none"> 1. El sistema no puede recuperar la información de la base de datos o el supervisor ha perdido conectividad con el servidor. 2. El reporte/estadística no es generado. 	
Anotaciones	No serán mostrados mensajes personalizados para las situaciones en las que no se ha encontrado información.

Tabla 2.27 Caso de Uso – Visualización de Registros, Reportes o Estadísticas

2.4.1.2.4. Administrador

Refinamiento de Caso de Uso: Administración de Puestos de Trabajo	
ID	ER-RCU-F-A1
Requisito Base	ER-ECU-R-M1
Nombre	Administración de Puestos de Trabajo.
Descripción	El administrador crea o elimina puestos de trabajo (extensiones) en el sistema.
Autor	Diego Chacón Herrera
Actor	Administrador
Condiciones Previas	El administrador debe haber iniciado sesión en el sistema.
Condiciones Resultantes	Los cambios se guardan en el sistema y la interfaz de administración se actualiza.
Flujo Normal de Eventos	
<ol style="list-style-type: none"> 1. El administrador seleccionará la opción de gestión de puestos de trabajo. Se despliega la interfaz correspondiente. 2. El administrador gestionará los puestos de trabajo. 	

<p>a. Crear puestos de trabajo:</p> <ol style="list-style-type: none"> i. El administrador selecciona el botón que permitirá crear los puestos de trabajo. ii. Se despliega una ventana para el ingreso de los datos de la nueva extensión a crear. iii. El administrador acepta los datos para crear el nuevo puesto de trabajo. iv. Se actualiza la interfaz para reflejar los cambios. <p>b. Eliminar puestos de trabajo:</p> <ol style="list-style-type: none"> i. El administrador seleccionará el botón de eliminación para el puesto de trabajo particular. ii. La interfaz se actualiza para reflejar los recientes cambios. 	
Flujos Alternos	
Cancelación de creación de usuarios	
En el paso 2.a.iii del flujo normal de eventos:	
<ol style="list-style-type: none"> 1. El supervisor selecciona la opción de cancelación de la creación del nuevo puesto de trabajo. 2. Se presenta la interfaz anterior. 	
Excepciones	
Puesto de Trabajo Repetido	
En el paso 2.a.iv del flujo normal de eventos:	
<ol style="list-style-type: none"> 1. La interfaz despliega un mensaje que informa que el puesto de trabajo ingresado ya ha sido creado. 	
Error de Eliminación de Puesto de Trabajo	
En el paso 2.b.ii del flujo normal de eventos:	
<ol style="list-style-type: none"> 1. La interfaz despliega un mensaje que informa que el puesto de trabajo seleccionado no se ha podido eliminar junto a la causa del error. 	
Anotaciones	Es posible crear extensiones sin que éstas existan en la central telefónica. Sin embargo, el usuario asociado a este puesto de trabajo no recibirá eventos de ningún tipo.

Tabla 2.28 Caso de Uso – Administración de Puestos de Trabajo (Extensiones)

Refinamiento de Caso de Uso: Administración de Usuarios	
ID	ER-RCU-F-A2
Requisito Base	ER-ECU-R-M1
Nombre	Administración de Usuarios.
Descripción	El administrador crea, modifica o elimina usuarios del sistema.
Autor	Diego Chacón Herrera
Actor	Administrador
Condiciones Previas	El administrador debe haber iniciado sesión en el sistema.
Condiciones Resultantes	Los cambios se guardan en el sistema y la interfaz de administración se actualiza.
Flujo Normal de Eventos	
<ol style="list-style-type: none"> 1. El administrador elige la opción de gestión de usuarios. Se despliega la interfaz correspondiente. 2. El administrador gestionará los puestos de trabajo. <ol style="list-style-type: none"> a. Crear usuarios: <ol style="list-style-type: none"> i. El administrador selecciona el botón que permitirá crear usuarios. ii. Se despliega una ventana para el ingreso de los datos del nuevo usuario a crear. <ol style="list-style-type: none"> 1. Nombre de usuario. 2. Contraseña. 3. Tipo de usuario (agente, administrador, supervisor). 4. Puesto de trabajo (extensión). 5. Información personal. iii. El administrador acepta los datos para crear el nuevo puesto de trabajo. iv. Se actualiza la interfaz para reflejar los cambios. b. Modificar usuarios: <ol style="list-style-type: none"> i. El actor selecciona el botón de modificación del perfil del usuario 	

<p>escogido.</p> <ul style="list-style-type: none"> ii. Se despliega una ventana para la alteración de los datos del usuario. iii. El administrador acepta los cambios. iv. La interfaz se actualiza con base en los cambios realizados. <p>c. Eliminar puestos de trabajo:</p> <ul style="list-style-type: none"> i. El administrador seleccionará el botón de eliminación para un usuario particular. ii. La interfaz se actualiza para reflejar los recientes cambios. 	
Flujos Alternos	
Cancelación de Acciones	
<p>En los pasos <i>2.a.iii</i> o <i>2.b.iii</i> del flujo normal de eventos:</p> <ul style="list-style-type: none"> 1. El administrador selecciona la opción de cancelación de la creación/modificación del usuario. 2. Se presenta la interfaz anterior sin cambios. 	
Excepciones	
Usuario Repetido	
<p>En el paso <i>2.a.iv</i> del flujo normal de eventos:</p> <ul style="list-style-type: none"> 1. La interfaz despliega un mensaje de error que informa que el usuario ingresado ya ha sido creado. 2. Se despliega la interfaz anterior sin cambios. 	
Error de Eliminación de Usuarios	
<p>En el paso <i>2.c.ii</i> del flujo normal de eventos:</p> <ul style="list-style-type: none"> 1. La interfaz despliega un mensaje que informa que el usuario seleccionado no se ha podido eliminar junto a la causa del error. 	
Anotaciones	<p>Para que los agentes puedan utilizar el sistema deben ser creados previamente por el administrador.</p>

Tabla 2.29 Caso de Uso – Administración de Usuarios

Refinamiento de Caso de Uso: Administración de Comunicaciones	
ID	ER-RCU-F-A3
Requisito Base	ER-ECU-R-M1
Nombre	Administración de Comunicaciones
Descripción	El administrador establece los parámetros de comunicación del sistema.
Autor	Diego Chacón Herrera
Actor	Administrador
Condiciones Previas	El administrador debe haber iniciado sesión en el sistema. No debe haber conexiones establecidas entre los componentes de la aplicación.
Condiciones Resultantes	Los parámetros de interconexión entre los componentes del sistema son definidos/actualizados.
Flujo Normal de Eventos	
<ol style="list-style-type: none"> 1. El administrador elige la opción de gestión de comunicaciones. Se despliega la interfaz correspondiente. 2. El actor selecciona la opción de configuración de parámetros de comunicación. 3. Se despliega un <i>wizard</i> de configuración de los parámetros utilizados en la comunicación entre los módulos del sistema. 4. El administrador acepta los datos ingresados y cierra el wizard de configuración. 5. El sistema inicia (arranca) las comunicaciones con base en los nuevos parámetros establecidos. 	
Flujos Alternos	
Cancelación	
En el paso 3 del flujo normal de eventos:	
<ol style="list-style-type: none"> 1. El administrador cierra el cuadro de configuración de parámetros de comunicaciones. 2. Se presenta la interfaz anterior sin cambios. 	
Edición	
Cuando el Administrador desee cambiar los campos de interconexión:	
<ol style="list-style-type: none"> 1. Detener las comunicaciones mediante la selección del botón de desconexión. 2. Ejecutar el paso 3 del flujo normal de eventos. 	
Anotaciones	No debe haber conexiones establecidas entre los componentes de la aplicación para que los parámetros puedan ser configurados.

Tabla 2.30 Caso de Uso – Administración de Comunicaciones

Refinamiento de Caso de Uso: Administración de Planes de Numeración	
ID	ER-RCU-F-A4
Requisitos Base	ER-ECU-R-M1, EA-AR-3
Nombre	Administración de Planes de Numeración
Descripción	El administrador establece los parámetros de comunicación del sistema.
Autor	Diego Chacón Herrera
Actor	Administrador
Condiciones Previas	El administrador debe haber iniciado sesión en el sistema. El administrador cuenta con los archivos que contienen la información de los planes de numeración corregida y simplificada.
Condiciones Resultantes	Los parámetros correspondientes a los planes de numeración del sistema son definidos/actualizados.
Flujo Normal de Eventos	
<ol style="list-style-type: none"> 1. El administrador elige la opción de gestión de planes de numeración. Se despliega la interfaz correspondiente. 2. El actor selecciona la zona numérica que desea administrar. <ol style="list-style-type: none"> a. En caso de no contener registros se presentará la opción de carga de datos (archivo que contiene la información numérica de la zona en cuestión). b. En caso de haber registros previos, éstos serán presentados en una tabla y adicionalmente se presentará la opción de eliminación de los mismos para poder actualizarlos con forme al literal anterior. 3. El administrador carga el archivo para cada zona, comprobando el éxito de la operación mediante la visualización de la información en una tabla. 	
Flujos Alternos	

Cancelación	
En el paso 3 del flujo normal de eventos:	
<ol style="list-style-type: none"> 1. El administrador cancela la operación de subida del archivo requerido. 2. Se presenta la interfaz anterior sin cambios. 	
Excepciones	
Archivo Incorrecto	
En el paso 3 del flujo normal de eventos:	
<ol style="list-style-type: none"> 1. Se presenta un mensaje de error que informa que el archivo no tiene el formato apropiado, excede el tamaño máximo o que contiene errores. 2. El usuario cierra el aviso y se presenta la interfaz anterior sin cambios. 	
Anotaciones	Los archivos deben tener la extensión .csv .

Tabla 2.31 Caso de Uso – Administración de los Planes de Numeración

Refinamiento de Caso de Uso: Administración de los Parámetros del Sistema	
ID	ER-RCU-F-A5
Requisitos Base	ER-ECU-R-M1
Nombre	Administración de Parámetros del Sistema
Descripción	El administrador establece o modifica los parámetros generales de funcionamiento del sistema.
Autor	Diego Chacón Herrera
Actor	Administrador
Condiciones Previas	El administrador debe haber iniciado sesión en el sistema.
Condiciones Resultantes	Los parámetros de la aplicación son exitosamente definidos/actualizados.
Flujo Normal de Eventos	

<ol style="list-style-type: none"> 1. El administrador elige la opción de administración de parámetros. 2. El actor selecciona la variable del sistema que desea editar y presiona el botón correspondiente. 3. Una vez modificado el valor del parámetro, selecciona la opción de guardado de cambios. 4. El administrador reinicia el sistema para que los cambios surjan efecto. 	
Flujos Alternos	
<p>Cancelación</p> <p>En el paso 3 del flujo normal de eventos:</p> <ol style="list-style-type: none"> 1. El administrador cancela la operación de almacenamiento de cambios realizados. 2. Se presenta la interfaz anterior sin cambios. 	
Excepciones	
<p>Parámetro no válido</p> <p>En el paso 3 del flujo normal de eventos:</p> <ol style="list-style-type: none"> 1. El parámetro ingresado no corresponde con los valores permitidos por el sistema. 2. Se muestra un mensaje de error. 	
Anotaciones	La configuración de los parámetros generales del sistema puede requerir del reinicio de todos los componentes que conforman la aplicación.

Tabla 2.32 Caso de Uso – Administración de Parámetros del Sistema

Refinamiento de Caso de Uso: Administración de Interconexión con Base de Datos	
ID	ER-RCU-F-A6
Requisitos Base	ER-ECU-R-M1, IA-AR-2a
Nombre	Administración de Interconexión con Base de Datos
Descripción	El administrador establece o modifica los parámetros de interconexión con la base de datos que contiene la información de los contactos de la institución.
Autor	Diego Chacón Herrera
Actor	Administrador
Condiciones Previas	<p>Debe haberse establecido la interconexión entre el servidor central (servidor de aplicaciones) y la base de datos de la organización a través de JDBC.</p> <p>En dicha base deberá configurarse una vista o tabla que agrupe la información específica de los contactos que será entregada a los agentes.</p> <p>El administrador debe haber iniciado sesión en el sistema.</p>
Condiciones Resultantes	Los parámetros de interconexión con la base de datos de la institución son exitosamente definidos/actualizados.
Flujo Normal de Eventos	
<ol style="list-style-type: none"> 1. El administrador elige la opción de administración de integración con base de datos. 2. El administrador configura los parámetros de interconexión: <ol style="list-style-type: none"> a. En caso de no haber establecido ninguna integración previamente, será desplegado un <i>wizard</i> en el que seleccionarán los siguientes parámetros: tipo de base de datos (Microsoft SQL Server, PostgreSQL, MySQL), origen de datos (vista o tabla) y atributos (columnas). b. En caso de modificar o actualizar la conexión, se permitirá borrar la conexión actual que permita continuar desde el paso 2.a. 	

3. El administrador guarda los cambios que son reflejados en la ventana respectiva.	
Flujos Alternos	
Cancelación	
En el paso 3 del flujo normal de eventos: <ol style="list-style-type: none"> 1. El administrador cancela el almacenamiento de los cambios. 2. Se presenta la ventana inicial de configuración. 	
Excepciones	
Atributo teléfono no definido	
En el paso 2.a del flujo normal de eventos: <ol style="list-style-type: none"> 1. De las columnas pertenecientes a la vista o tabla seleccionada, no fue seleccionada ninguna como número telefónico. 2. Se muestra un mensaje de error. 	
Anotaciones	La base de datos a la que se refiere este caso de uso no corresponde con la base de datos interna del sistema.

Tabla 2.33 Caso de Uso – Administración de Interconexión con Base de Datos

2.4.2. ANÁLISIS Y DISEÑO (EAD)

Hasta el momento ha sido posible determinar qué es lo que se necesita del sistema y los actores involucrados en cada escenario, pero no el cómo operará. Para ello será útil estudiar todos los requerimientos (descritos como casos de uso) para determinar los flujos, elementos e interacciones involucrados en cada uno de ellos.

Para este apartado se ha considerado conveniente la fusión de las disciplinas de análisis y diseño en una sola ya que de esta forma se facilita la comprensión del funcionamiento y dimensionamiento de la solución.

Por otra parte, tomando en cuenta que la documentación oficial del fabricante está escrita en idioma Inglés, con el objeto de mantener una coherencia entre todas las funciones, estructuras, enumeraciones y clases, se ha resuelto mantener dicho idioma como base para el desarrollo del sistema. Es decir, todo el código del sistema

(tanto para clientes, servidor y creación de base de datos) estará escrito en idioma inglés.

Finalmente, el proyecto tomará como nombre clave **Call Manager**, al tratarse precisamente de un sistema encargado de la gestión, control y registro de las llamadas telefónicas. A continuación se presentará el logotipo que utilizará el sistema a lo largo de todo el proyecto.



Figura 2.13 Logotipo de “Call Manager”

2.4.2.1. Comunicaciones (EAD-COM)

Uno de los puntos más importantes en el desarrollo de aplicaciones distribuidas consiste en la especificación de la nomenclatura y el procedimiento de comunicación entre los distintos elementos que conforman las soluciones. En lo que respecta a este proyecto, es necesaria una tecnología que permita intercomunicar componentes escritos en distintos lenguajes de programación, como es el caso del paso de eventos y comandos entre el servidor CTI (C++) y el servidor central (Java). En la tabla 2.34 se mencionan las principales alternativas de interconexión multiplataforma de los componentes de sistema.

La facilidad de uso que ofrecen los *sockets planos* sobre las otras alternativas ha sido el motivo de selección como tecnología de interconexión entre los distintos módulos. Esto no quiere decir que las otras opciones no sean válidas (ni las únicas) para la creación de aplicaciones distribuidas, sino que, conforme a los plazos y

recursos disponibles para la creación de este proyecto, la primera alternativa es la más conveniente. Cabe recalcar, además, que las tecnologías *WebSockets* y Servicios Web han sido agrupadas como una misma alternativa por sus características comunes, aunque su implementación y funcionamiento sean completamente distintos.

Tecnología	Características
Sockets planos	Comunicación de red a bajo nivel. Ofrece una mayor flexibilidad para la transmisión de información. Generalmente consume menos recursos de red. Requiere del diseño e implementación de protocolos, sistemas de autenticación y reconexión a bajo nivel.
WebSockets / Servicios Web	Orientado principalmente a entornos web. Basado en HTTP. Alto nivel. Adición innecesaria de <i>overhead</i> para transmisión de pocos datos.
JMS	Complejidad de desarrollo. Enfocado a Java, requiere de un bróker de mensajería o de JMV y <i>proxies</i> para los clientes escritos en otros lenguajes de programación. Utilizado principalmente en grandes <i>backbones</i> empresariales o para la interconexión con sistemas legados.

Tabla 2.34 Principales Alternativas de Interconexión entre los Componentes del Sistema

Por otra parte, tomando en cuenta que tanto el servidor central como los agentes de escritorio serán desarrollados en el mismo lenguaje de programación (Java), es posible utilizar otras tecnologías de comunicaciones que son específicas de la plataforma, como la Invocación Remota de Métodos (*RMI*) o mediante la invocación de EJBs remotos, las cuales requieren de conocimientos más avanzados y por lo tanto un mayor tiempo de construcción de la solución. Conforme a los objetivos

planteados en el párrafo anterior, y para facilitar el entendimiento y comprensión del sistema, también se escogerá la comunicación mediante *sockets* como alternativa de comunicación entre los agentes de escritorio y el servidor central. La tabla 2.35, por tanto, muestra los números de puerto privados que por defecto serán utilizados para la interconexión de los distintos componentes de la solución, los cuales podrán ser cambiados en cualquier etapa del ciclo de vida de la aplicación. No se incluyen los números de puerto utilizados por el servidor web o las bases de datos.

Componente	Puerto	Uso
Servidor CTI	47915	Interconexión con el servidor de llamadas
Servidor Principal	47925	Escucha de agentes de escritorio

Tabla 2.35 Números de Puerto por Defecto Utilizados por el Sistema

Ahora bien, dado que se ha escogido interconectar los módulos de sistema a un bajo nivel, es obligatorio el intercambio de comandos, eventos y mensajes definidos en un formato común y organizado llamado protocolo. Para el diseño del mismo ha sido necesario el estudio de cada una de las funcionalidades ofrecidas por la central telefónica (descritas más adelante), así como del análisis de los casos de uso tratados con anterioridad.

En términos generales, los mensajes serán transmitidos en el siguiente formato:

Inicio<#>Notificación<#>Clase<#>Comando<#>Campos_variables<#>Final

- **Inicio.** Corresponde a un conjunto de caracteres cuyo propósito es establecer (delimitar) el comienzo de la trama. En este caso particular la cadena de inicio es **<BOF>**. Tanto los delimitadores de inicio como de fin de trama podrán ser cambiados por el administrador del sistema.
- **Separador.** El elemento separador tiene por objeto seccionar los campos que componen la trama; en este caso particular están separados por **<#>**. Este

campo también podrá ser personalizado (cambiado) por el administrador del sistema.

- **Notificación.** El campo de notificación tiene por objeto determinar el propósito del mensaje, que podrá ser cualquiera de las siguientes opciones:
 - **EVENT.** El mensaje transmitido contiene información relacionada con eventos del sistema (como una desconexión entre los componentes) o de las líneas telefónicas (como el cambio del estado de una llamada, por ejemplo).
 - **MESSAGE.** La trama transmitida contiene mensajes de carácter informativo.
 - **REQUEST.** El propósito de transmitir mensajes este tipo es solicitar algún recurso, servicio o estado (como por ejemplo, solicitar la realización de una llamada o recuperar el estado no disponible del teléfono).
 - **RESPONSE.** Generalmente es transmitido como respuesta a un mensaje de tipo *Request*, aunque también puede ser parte de mensajes asincrónicos que no requieren respuesta (como el envío de información de los contactos en una llamada, por ejemplo).
- **Clase.** La clase tiene por objeto catalogar el ámbito o el origen de la trama. En el diseño han sido consideradas las siguientes clases:
 - **CTI.** Un comando CTI implica que el mensaje en cuestión contiene órdenes que sólo podrán ser interpretadas por el servidor CTI que implementa el TAPI 2.0. En otras palabras, este campo únicamente estará relacionado con llamadas o líneas telefónicas.
 - **SESSION.** En la trama están siendo enviadas credenciales o información de sesión que requerirán de la asignación o liberación de recursos.
 - **COMMUNICATION.** Interpretado únicamente por los componentes internos del sistema. Su propósito es intercambiar información

- relevante respecto al estado de la comunicación entre los mismos (como *heartbeats*, por ejemplo).
- **DESKTOP.** En mensaje ha sido creado o va dirigido hacia el aplicativo de escritorio del agente.
 - **GENERAL.** Mensajes no identificados o de carácter general.
 - **Comando.** Este campo especifica el nombre del servicio o de la función requerida por el componente. A su vez está clasificado como:
 - **LOGIN.** Inicio de sesión.
 - **LOGOUT.** Cierre de sesión.
 - **KEEP_ALIVE.** Monitoreo periódico del estado de la comunicación (*heartbeat*).
 - **CALL.** Llamada.
 - **MAKECALL.** Realizar una llamada.
 - **ANSW.** Responder una llamada.
 - **DROP.** Colgar una llamada activa.
 - **HOLD.** Poner una llamada en espera.
 - **UNHOLD.** Recuperar una llamada retenida.
 - **ATRN.** Iniciar una transferencia asistida.
 - **DIAL.** Marcar.
 - **FIN.** Finalizar una transferencia asistida.
 - **BTRN.** Realizar una transferencia ciega.
 - **CONF.** Iniciar una conferencia telefónica.
 - **SETDIVERTSET.** Configurar características de desvío de una línea telefónica.
 - **GETDIVERTSET.** Obtener las características de desvío de una línea telefónica.
 - **DONOTDISTURB.** Establecer el estado de No Molestar (DND) del teléfono.
 - **LINEDEVSTATE.** Recuperar parámetros de una línea particular.
 - **LINEDEVSTATE_DEVSPECIFIC.** Recuperar detalles respecto a una línea particular.

- **LINE_STATUS.** Recuperar el estado de una línea telefónica.
- **READY.** Determinar la disponibilidad de la línea telefónica.
- **CALL_DATA.** Información de la llamada relacionada al número telefónico detectado.
- **CALL_OBS.** Observaciones de una llamada particular.
- **CLIENT_DATA.** El mensaje contiene información de los contactos identificados en la llamada externa.
- **LOCAL_DATA.** Información de los participantes involucrados en una llamada interna.
- **CLIENT_MAPPING.** Información de los parámetros que componen la información los clientes en llamadas externas.
- **LOCAL_MAPPING.** Información de los parámetros de los contactos en llamadas internas.
- **UNKNOWN.** Estado desconocido o sin propósito específico.
- **Final.** Conjunto de caracteres que delimita el fin de la trama. En este proyecto el delimitador está definido por: **<EOF>**.

De lo anterior, puede ejemplificarse el caso en el que un agente cuyo número de extensión es 2040 desee realizar una llamada a la extensión 2050. Para ello, se creará un mensaje de tipo:

```
<BOF><#>REQUEST<#>CTI<#>MAKECALL<#>IP Office Line: 2040<#>2050<#><EOF>
```

La tabla 2.36 detalla las posibles combinaciones entre los campos anteriormente definidos, así como también describe los campos variables correspondientes a cada mensaje transmitido por el *socket*.

NOTIFICACIÓN	CLASE	COMANDO	CAMPOS VARIABLES					
EVENT	CTI	CALL	Nombre de Línea	Id Llamada	Estado	Llamante	Llamado	
	COMMUNICATION	KEEP_ALIVE	Cuenta	Fecha				
	SESSION	LOGIN	Id Usuario	Username	Contraseña			
		LOGOUT	Línea	Username				
	REQUEST	CTI	MAKECALL	Línea	Llamado			
			DROP	Línea	Id Llamada			
			HOLD	Línea	Id Llamada			
			UNHOLD	Línea	Id Llamada			
			ATRN	Línea	Id Llamada			
			DIAL	Línea	Llamado			
FIN			Línea					
BTRN			Línea	Id Llamada	Llamado			
RESPONSE	SESSION	ANSW	Línea	Id Llamada				
		SETDIVERTSET	Línea	Todo	Ocupado	No contesta		
		LINEDEVSTATE_DEVSPECIFIC	Línea	"DONOTDISTURB"	Acción	Valor		
		LINE_STATUS	Línea					
		CLIENT_MAPPING	Línea	Ámbito				
		CALL_OBS	Línea	Id Llamada	Observaciones			
		LOGIN	Resultado	Username	Línea			
MESSAGE	Tipo	LOGOUT	Username	Línea				
		CALL_DATA	Línea	Id Llamada	Origen	Zona	Operadora	
		CLIENT_DATA	Línea	Ámbito	Id Llamada	Valores repetidos	Detalles	
		CLIENT_MAPPING	Línea	Ámbito	Campos	Detalles		
		Origen	Mensaje					

Tabla 2.36 Protocolo de Comunicaciones

2.4.2.2. Inicio y Cierre de Sesión en el Sistema – Agentes (EAD-SES)

En el caso de los agentes, los eventos de inicio y cierre de sesión del aplicativo asociado en la plataforma tienen una gran importancia ya que son los desencadenadores de la asignación o liberación de los recursos (y por lo tanto, la gestión de más eventos) que serán claves en la operación de los programas de escritorio.

Este análisis y diseño cubre los casos de uso definidos en la etapa de requisitos (ER):

- ER-RCU-F-G1: Inicio de Sesión.
- ER-RCU-F-G2: Cierre de Sesión.

Como resultado del estudio de los casos de uso anteriores, han sido diseñados los diagramas de actividades posteriores que marcarán la pauta de funcionamiento no sólo del aplicativo de escritorio de los agentes sino también del servidor central y el controlador CTI.

El diagrama de actividades de la figura 2.14, relacionada con el inicio de sesión de agentes, revela que únicamente se procesarán los eventos de las extensiones telefónicas de los agentes configurados y conectados al sistema, descartando cualquier otro tipo de evento que no cumpla este requerimiento. En otras palabras, aunque posible, no se gestionará la información de otras líneas telefónicas que no hayan sido previamente creadas, habilitadas y activadas. Esta decisión se apoya en la optimización de los recursos disponibles, como por ejemplo el tiempo de procesamiento, transmisión de datos por la red, espacio de memoria, espacio de almacenamiento en disco, etc., evitando así el tratamiento de información poco relevante o innecesaria. La autorización para la recepción de eventos, como es posible apreciar, tiene lugar cuando se comprueba el inicio de sesión del agente en el servidor central, el cual envía un mensaje de monitoreo de eventos al controlador CTI.

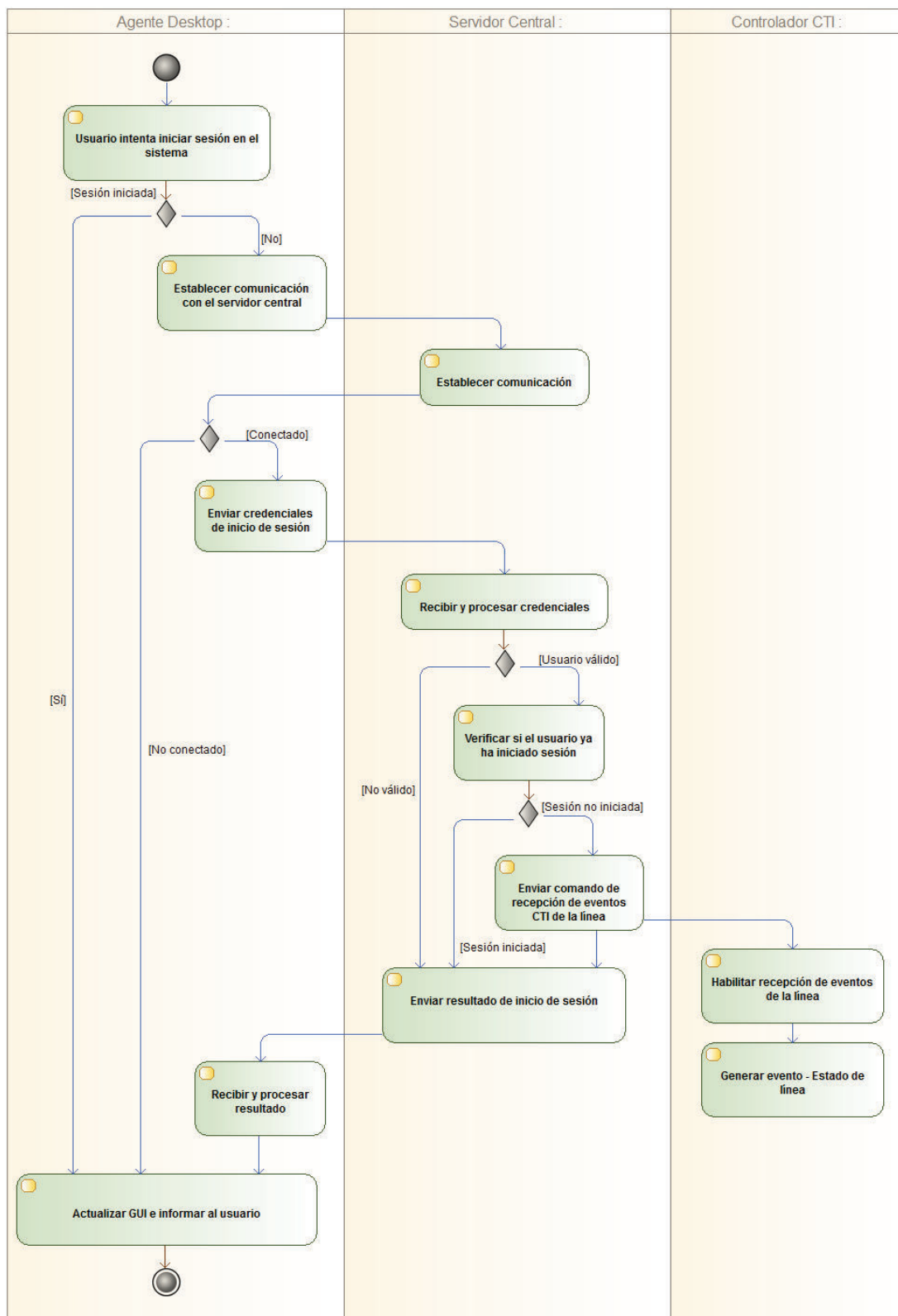


Figura 2.14 Diagrama de Actividades - Inicio de Sesión de Agentes

En lo que respecta al cierre de sesión, el agente envía una notificación al servidor principal informando su deseo de salir del sistema. El servidor en cuestión libera los recursos asignados al agente e inmediatamente notifica al servidor CTI, indicando que no es necesario el monitoreo de los eventos generados por la línea telefónica asociada al agente.

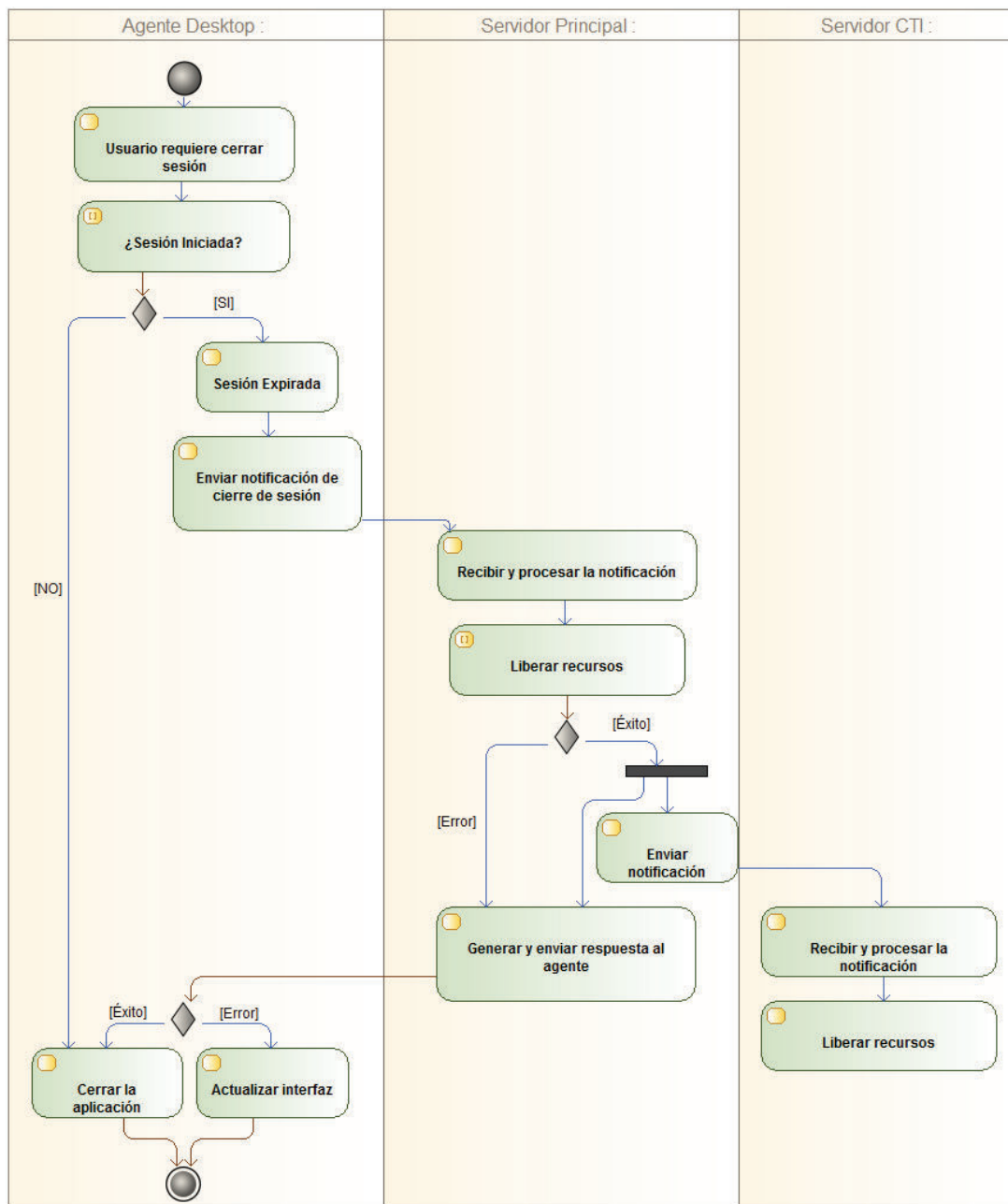


Figura 2.15 Diagrama de Actividades – Cierre de Sesión de Agentes

2.4.2.3. Gestión de Eventos de las Líneas Telefónicas (EAD-EVN)

El siguiente paso en el análisis de la solución consiste en la determinación de las actividades relacionadas con la gestión de los eventos generados por la central telefónica (a través de TAPI) y por los agentes con respecto a las líneas telefónicas del *call center*.

El subsiguiente análisis abarca los casos de uso:

- ER-ECU-R-A1: Gestión y control de llamadas telefónicas. Éste a su vez incluye los casos de uso ER-RCU-F-A1-6.
- ER-ECU-R-A2: Despliegue de información de los contactos en una llamada.
- ER-ECU-R-A4: Detalle del número telefónico de contacto en una llamada externa.
- ER-RCU-F-A7: Recepción de eventos.

2.4.2.3.1. Eventos Generados por la Central Telefónica (EAD-ENV-IPO)

En este caso, el primer componente del sistema implicado en la detección de eventos es el servidor CTI, el cual, luego de determinar si la extensión corresponde a un agente activo, enviará la notificación al servidor central.

El servidor principal a su vez será responsable de buscar información adicional de la llamada con base en los números de origen y destino de la misma (como el nombre de la operadora o la información del contacto de una base de datos) para posteriormente ser enviada al aplicativo de escritorio del agente.

Este último, finalmente, actualizará su interfaz gráfica para reflejar los últimos cambios y eventos producidos según la información obtenida por el servidor principal que no necesariamente llega al mismo tiempo (ya que primero detecta la notificación de la llamada y después la información adicional de contacto).

El diagrama de actividades mostrado de la figura 2.16 representa tal ocurrencia.

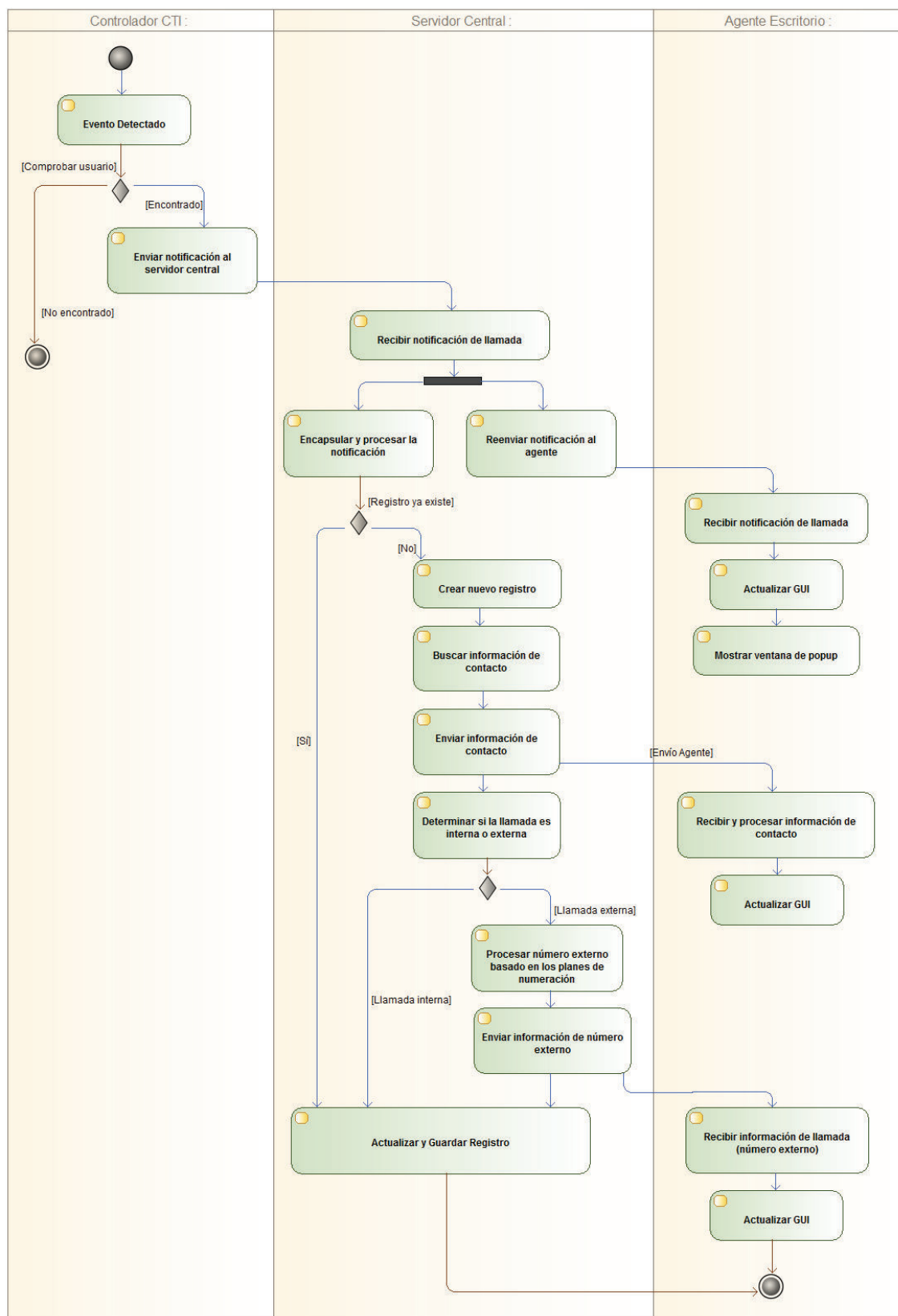


Figura 2.16 Gestión de Eventos de Líneas Telefónicas Generados por la Central Telefónica

Ahora bien, un estudio detallado de las aplicaciones de ejemplo proporcionadas por Avaya en su SDK refleja varios estados, objetos y funciones comunes y necesarias para la gestión de los eventos de las llamadas telefónicas. Con el objeto de reducir el tiempo de desarrollo del controlador CTI de esta solución se ha considerado utilizar un esquema organizacional similar al de los ejemplos analizados. Entre los eventos y objetos más importantes a ser incluidos en la solución están los siguientes:

- **Manipulador de TAPI.** Este objeto es el encargado de encapsular todas las funciones y variables encargadas de la administración del TAPI, así como el conjunto de líneas detectadas.
- **Manipulador de línea.** Gestiona la información específica de cada línea telefónica como el número de extensión, el conjunto de llamadas activas, el estado de la línea, etc.
- **Manipulador de llamada.** Controla y registra la información perteneciente a una llamada activa particular.
- **Eventos (mensajes).** Los eventos que pueden ser producidos y detectados por TAPI están detallados en la página 42 del documento “*TAPILink Developer’s Guide*” de Avaya. Entre los más importantes están:
 - **LINE_APPNEWCALL.** Informa que una nueva llamada ha sido creada.
 - **LINE_CALLINFO.** La información contenida en la estructura LINECALLINFO ha cambiado. De forma general coincide con los cambios producidos en el estado de la línea telefónica que es manejado por LINE_CALLSTATE.
 - **LINE_CALLSTATE.** El estado de la llamada ha cambiado. Los estados soportados se encuentran definidos en la estructura LINEADDRESSCAPS que está detallada a continuación:

ESTADO	DESCRIPCIÓN
LINECALLSTATE_IDLE	La llamada ya no existe / No hay ninguna llamada sobre la línea.
LINECALLSTATE_OFFERING	Aviso de nueva llamada entrante.
LINECALLSTATE_ACCEPTED	La llamada ha sido tomada por la aplicación.
LINECALLSTATE_DIALTONE	El usuario llamante escucha el tono de marcación.
LINECALLSTATE_DIALING	El conmutador está recibiendo la información de marcado (números).
LINECALLSTATE_RINGBACK	El llamante escucha el tono de timbrado.
LINECALLSTATE_BUSY	El llamante escucha el tono de ocupado.
LINECALLSTATE_CONNECTED	La llamada ha sido conectada entre los dos extremos.
LINECALLSTATE_PROCEEDING	La marcación ha sido completada pero la llamada aún no ha sido conectada.
LINECALLSTATE_ONHOLD	La llamada está retenida.
LINECALLSTATE_ONHOLDPENDTRANSFER	La llamada está retenida antes de ser transferida.
LINECALLSTATE_DISCONNECTED	La llamada ha sido colgada desde el otro extremo.
LINECALLSTATE_UNKNOWN	El estado de la llamada es desconocido.

Tabla 2.37 Estados soportados por la central telefónica Avaya a través de TAPI.

El procesamiento y posterior envío de cada uno de los eventos producidos por la central telefónica que son detectados en el servidor CTI se encuentran detallados en el diagrama de actividades de la figura 2.17. Se destaca que cualquier cambio en el estado de la línea es detectado en el evento LINE_CALLSTATE, por lo que es el único utilizado para la construcción de mensajes que luego son enviados al agente.

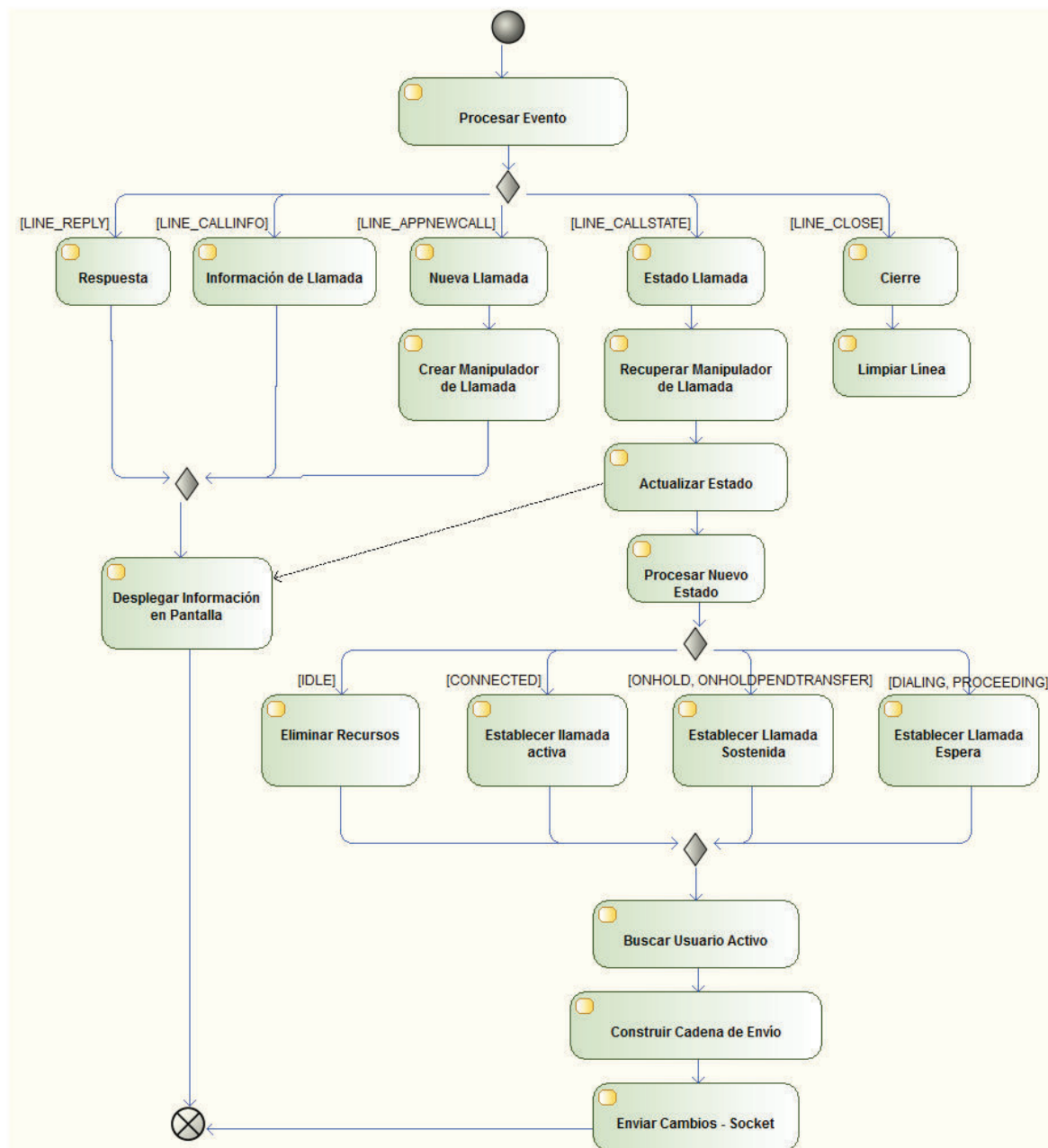


Figura 2.17 Procesamiento de Eventos Generados por la Central Telefónica en el Servidor CTI

2.4.2.3.2. Eventos Generados por el Agente (EAD-ENV-AGT)

En lo que respecta al agente, la interacción de éste con la interfaz del aplicativo con el propósito de cambiar el estado de la línea telefónica (realizar una llamada, colgar, transferir, etc.) se convierte en un requerimiento (comando) que se envía al servidor central. Éste a su vez transfiere la petición al servidor CTI, último responsable del procesamiento y cambio del estado de la línea telefónica.

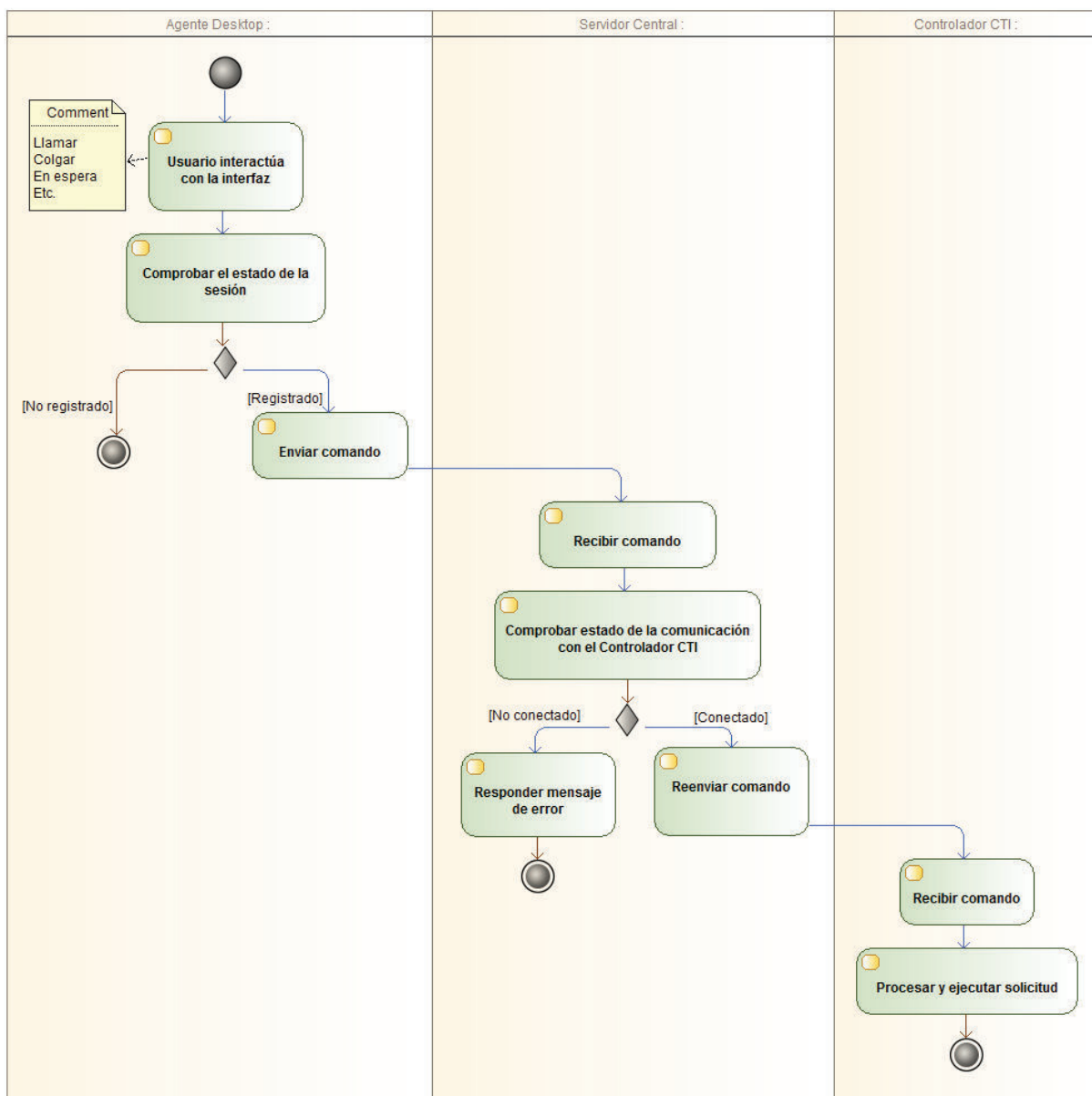


Figura 2.18 Gestión de Comandos de los Usuarios

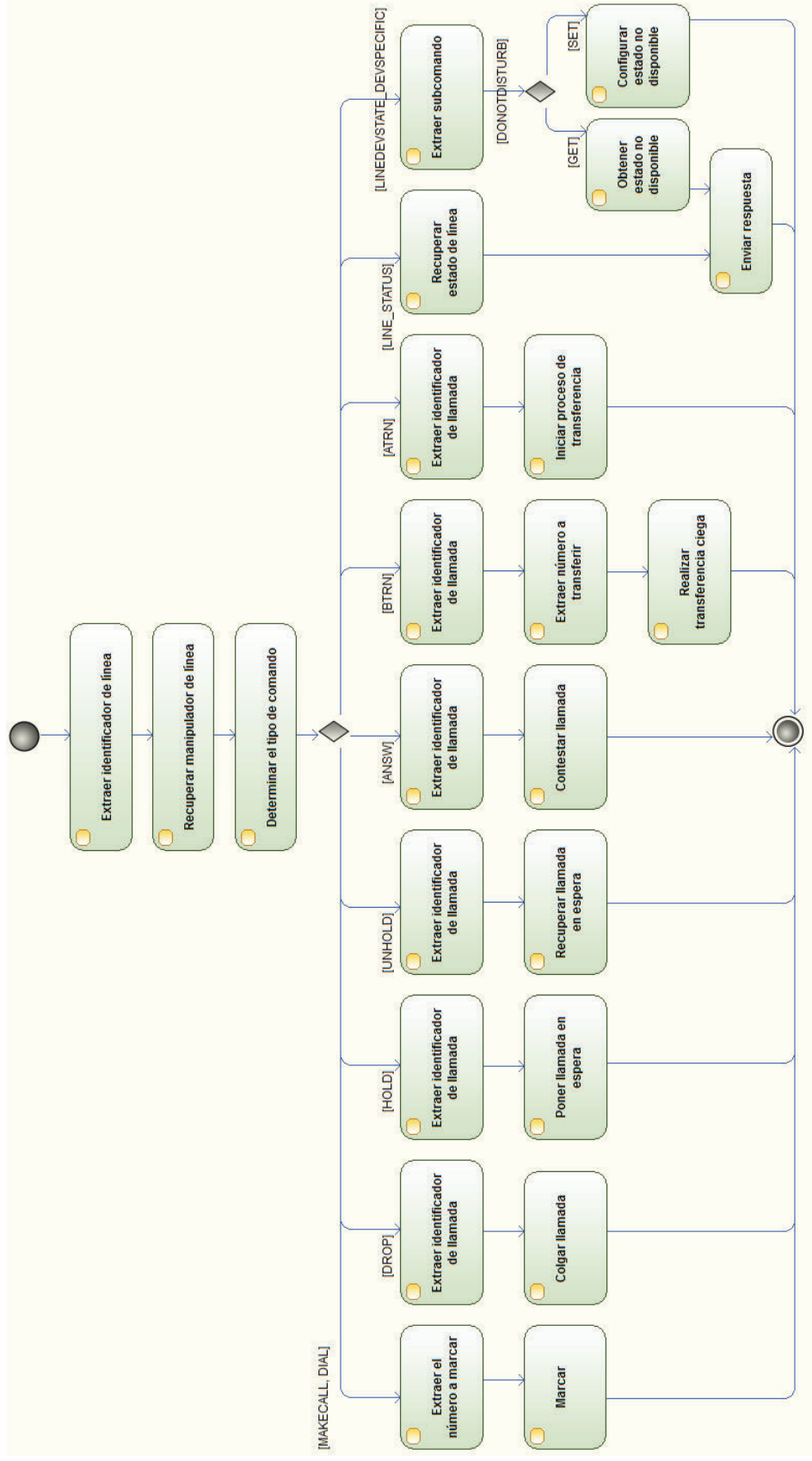


Figura 2.19 Gestión de Eventos Generados por los Agentes en el Servidor CTI

FUNCIÓN	DESCRIPCIÓN
lineAnswer	Responder una llamada entrante.
lineBlindTransfer	Transferir una llamada activa a una tercera línea de forma inmediata (transferencia ciega).
lineCompleteTransfer	Completar o finalizar una transferencia (transferencia asistida).
lineClose	Cerrar una línea.
lineDevSpecific	Establecer funcionalidades extendidas (dependientes del fabricante).
lineDial	Marcar a un número en el transcurso de una llamada activa (utilizada en transferencias asistidas).
lineDrop	Colgar una llamada
lineGetAddressCaps	Obtener las capacidades de una línea particular.
lineGetCallInfo	Obtener la información relacionada con una llamada específica.
lineGetCallStatus	Obtener el estado de una llamada particular.
lineGetLineDevStatus	Utilizada para recuperar información especial de una línea telefónica.
lineHold	Retener una llamada activa.
lineInitializeEx	Es la primera función que debe ser llamada para inicializar el TAPI.
lineMakeCall	Función utilizada para realizar una llamada.
lineOpen	Abrir (habilitar) una línea.
lineSetupTransfer	Crear una transferencia asistida. Para ello deberá haber una llamada activa en proceso.
lineShutdown	Finalizar todas las funciones TAPI sobre la línea.
lineSwapHold	Poner la llamada activa en espera y recuperar la otra llamada sostenida.
lineUnhold	Recuperar una llamada en espera.

Tabla 2.38 Principales Funciones Soportadas por el TAPI 2.0 de Avaya.

Además del procesamiento de los eventos generados por la central telefónica, el controlador CTI también tiene la obligación de gestionar los mensajes producidos por los agentes de escritorio (como por ejemplo la realización de llamadas, transferencias, entre otros). El diagrama de actividades de la figura 2.19 muestra el flujo de tareas que serán ejecutadas cuando se reciban este tipo de eventos.

Adicionalmente, para que el controlador CTI pueda ser capaz de realizar cambios sobre las líneas telefónicas, es obligatorio conocer las funciones soportadas por el TAPI subyacente. La guía de desarrollo *TAPILink* de Avaya ^[16], a partir de la página 16 describe cada una de las funciones soportadas por el *driver* de la versión 2.x. En lo que respecta a los objetivos de este proyecto las funciones más importantes se indican en la tabla 2.38.

2.4.2.4. Despliegue de la Información de Contacto en una Llamada (EAD-INF)

A partir de lo expresado en la fase de inicio **IA-AR-2** en esta parte se diseñará un modelo que permita extraer información de bases de datos externas de una forma común y no intrusiva; esto quiere decir que la creación o modificación de cualquier registro perteneciente a un contacto será tratado fuera de esta solución. En otras palabras *Call Manager* únicamente recuperará (sin modificar) información de una base de datos externa cada vez que se procese una llamada en el *call center*. Los motivos de esta decisión han sido descritos en la fase de inicio **IA-AR-2a** que resumidos están relacionados con la diversidad y complejidad de las bases de datos así como problemas de integridad, consistencia e incluso seguridad.

Con base en lo anterior, es necesaria la elección de una tecnología Java de acceso a datos compatible con la mayor cantidad de gestores de bases de datos disponibles, siendo JDBC la más apropiada para ello. Al ser diseñada para ser independiente de la plataforma (de datos) esta solución proporciona acceso a la información almacenada a través de una interfaz de programación común, ajustada a cada situación particular en caso de así requerirlo. Para este proyecto se tomarán en cuenta únicamente las bases de datos más conocidas en la construcción y realización de pruebas de conectividad debido a la enorme cantidad de alternativas

existentes en el mercado. No queda por demás aclarar que se pretenderá mantener la solución lo más general posible con el objeto de reducir los cambios al mínimo en el caso de utilizar algún otro tipo de bases de datos aquí no especificadas.

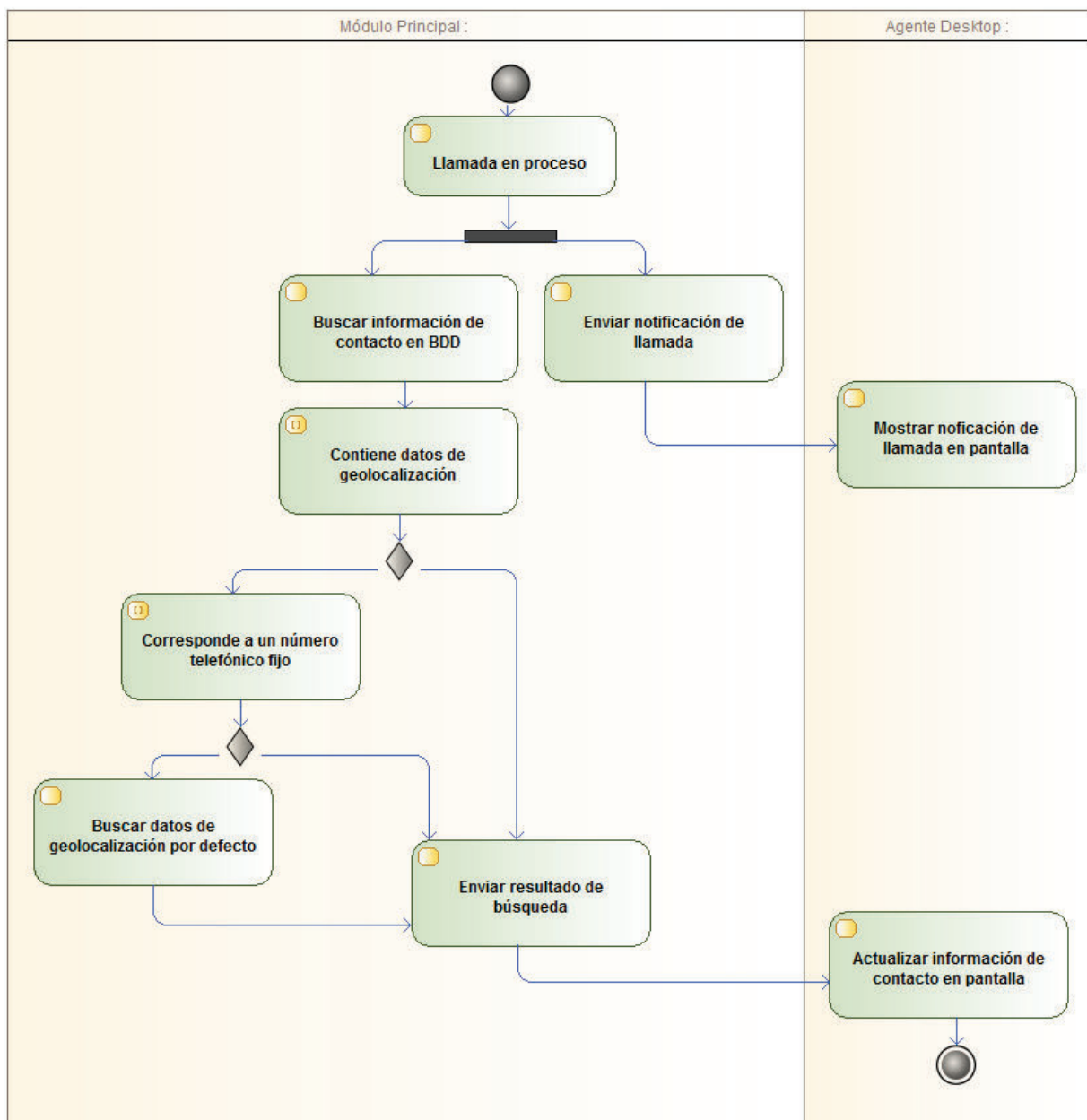


Figura 2.20 Notificación de Llamadas y Búsqueda de Información de Contactos

Como es conocimiento del lector, el evento que desencadene la búsqueda de la información del contacto será la generación o la recepción de una llamada telefónica.

Ahora bien, existe la posibilidad de que los datos del contacto no puedan ser recibidos a tiempo por varias causas (para ejemplificar, por problemas físicos, de seguridad o de comunicación) que aplacen e incluso interrumpen el aviso del cambio de estado de la línea telefónica a los agentes. Bajo esta consideración, por tanto, es conveniente separar la notificación correspondiente a la llamada en curso de la notificación relacionada con la búsqueda de la información del contacto para de esta manera evitar retrasos en la producción de notificaciones. Así pues, tan pronto como sea detectada una llamada telefónica en el sistema, de forma simultánea se realizará el envío del aviso del evento generado y la búsqueda de la información del contacto los cuales actualizarán la interfaz del agente de escritorio, proceso que está esquematizado en el diagrama de la figura 2.20.

En el punto IA-AR-2 también se establece que la información de contacto puede incluir los datos de georreferenciación en el caso de tratarse de números telefónicos fijos. Éstos tienen que haber sido previamente almacenados en las base de datos de las instituciones para poder ser mostrados en un mapa, aunque en el caso de no encontrarse dichos resultados se ha considerado como alternativa la posibilidad de utilizar un servicio público de geolocalización ofrecido por la CNT. Es importante notar que la opción presentada tiene por objeto complementar y mejorar el desempeño del sistema y no está contemplado dentro del alcance original de este proyecto. Como se presenta en el diagrama 2.20, la búsqueda de este tipo de información será efectuada después de haber consultado la base de datos local y de haber confirmado que la llamada ha sido establecida con un número telefónico fijo.

Finalmente, es imperioso definir un marco que estandarice el acceso a la información y establezca límites que determinen y faciliten el proceso de integración de *Call Manager* con el repositorio de datos externo requerido para su funcionamiento. Debido a que las bases de datos difieren en cada organización, el registro “Teléfono” puede ser llamado “Phone” en otra base de datos, por ejemplo. Además, la información de los contactos puede estar distribuida en varias tablas de una misma base de datos (o en más de una en ciertos casos) e incluso ciertas instituciones pueden restringir el acceso a ciertas tablas importantes que guarden datos críticos

que sean compartidos con aquellos solicitados por el *call center*, aumentando considerablemente la complejidad en la creación de un sistema general de navegación y recuperación de información de interés para este proyecto.

Por los motivos señalados en el párrafo anterior, *Call Manager* se integrará con bases de datos externas mediante la lectura de una única tabla o vista que contenga la información de contacto de interés para el *call center*, método que provee las siguientes ventajas:

- Al recuperar los datos de un único lugar, será fácil determinar qué campos corresponden al nombre, número de teléfono, latitud, longitud, entre otros; por lo que no habrá necesidad de navegar por distintas tablas a lo largo de la base de datos para tal propósito, facilitando la administración y configuración del sistema.
- La creación de una vista es especialmente recomendada, ya que ésta puede integrar los datos pertenecientes a diversas tablas y presentarlos como una sola. Cabe aclarar que la creación, configuración y mantenimiento de dicha vista (o tabla de ser el caso) está a cargo de los administradores y DBAs de las instituciones interesadas y es manejada fuera de la solución propuesta en este proyecto.
- La configuración de permisos de acceso en los gestores de bases de datos se simplifica al constar de un único elemento de sólo lectura que es accedido por *Call Manager*.

2.4.2.5. Detalle del Número Telefónico Participante en una Llamada Externa (EAD-NUM)

En la sección EA-AR-3 de este documento se había indicado que se utilizarían los planes de numeración publicados por el ARCOTEL para determinar el nombre de la operadora (números fijos y celulares) y la provincia (números fijos solamente) en el transcurso de una llamada.

En primer lugar, el responsable del mantenimiento de los planes de numeración será el administrador de la solución, el cual tendrá la posibilidad de borrar y actualizar dicha información en caso de ser necesario. Para ello, modificará los archivos de Excel (.xls) descargados de la página oficial y creará archivos de tipo de Valores Separados por Comas .csv (, *Comma Separated Value*) bajo el estándar **RFC 4180**. Cada dígito del plan de numeración (TC para números fijos, DN para móviles) debe ser cargado como un archivo independiente, por lo que existirán ocho archivos en total, los cuales obligatoriamente incluirán los siguientes campos:

- **Nombre de la central.** Campo aprovechado en el tratamiento de números telefónicos fijos.
- **Serie asignada.** Es el rango numérico en el que operan las líneas para cada central.
- **Provincia.** Campo de utilidad para la telefonía fija.
- **Operadora.**

El administrador del sistema deberá cargar los archivos generados uno a uno mediante el empleo del navegador web para almacenar dicha información en la base de datos interna de la solución. El procedimiento de creación de este tipo de archivos está detallado en la sección de anexos E de este documento.

En cuanto a la cronología en la recuperación de los datos se refiere, ésta se realizará luego de primero haberse consultado la información del contacto (por su carácter más general y menos prioritario). Finalmente, el resultado de la búsqueda será reenviado al agente de escritorio conforme a los siguientes campos: nombre de la central (números fijos), provincia (fijos) y operadora (números fijos y celulares).

2.4.2.6. Servidor CTI (EAD-CTI)

Descrito en la fase de inicio (IA-AA), este módulo será el único dependiente de la plataforma (debido al uso del TAPI 2.x proporcionado por el fabricante), cuya principal funcionalidad es la de enviar y recibir comandos y eventos desde/hacia la

central telefónica. Dicho de otro modo, el servidor CTI es una especie de *proxy* entre el sistema informático y el sistema de telefónica (de allí su nombre).

De acuerdo a lo establecido en la sección de análisis EAD-EVN el estudio de los programas de ejemplo incluidos en el Kit de Desarrollo de Software (SDK) proporcionado por Avaya, así como de las necesidades del sistema actual da como resultado el diagrama de clases mostrado en la figura 2.21.

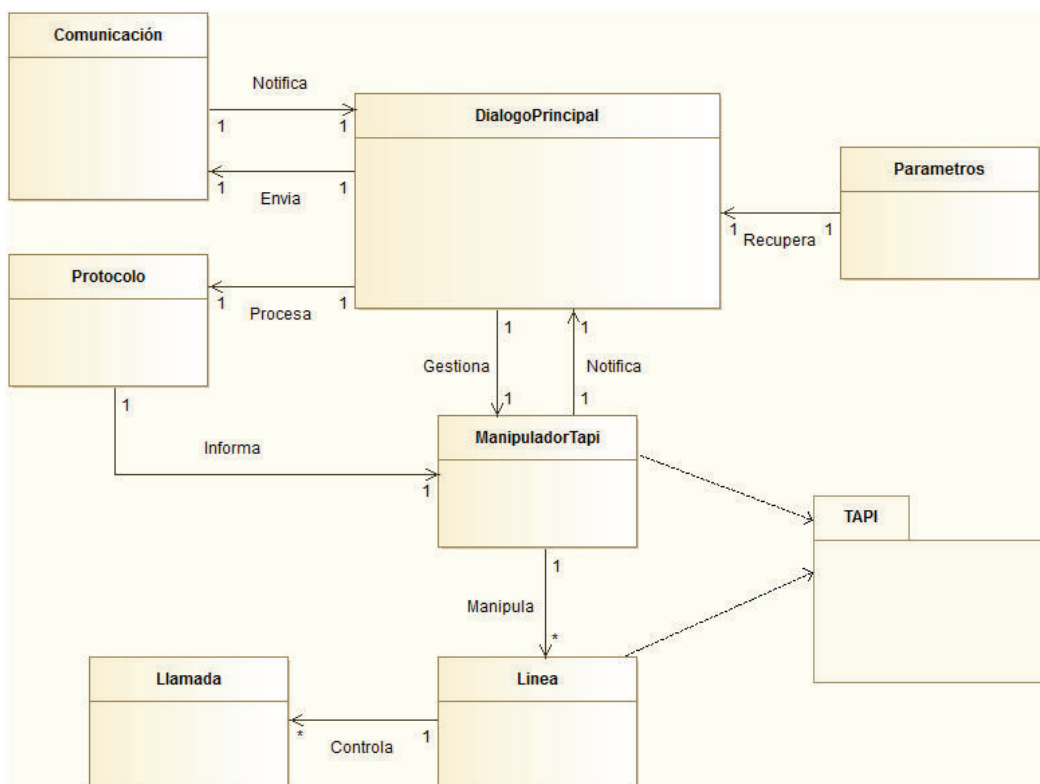


Figura 2.21 Diagrama de Clases del Servidor CTI

Los principales objetos y clases son los siguientes:

- **ManipuladorTapi**. Clase que carga, inicializa y encapsula las funcionalidades de la Interfaz de Programación de Aplicaciones de Telefonía proporcionada por el fabricante de la central telefónica.

- **Línea.** Clase cuyo propósito principal es el de agrupar las características y funciones de las líneas telefónicas para su fácil tratamiento. Esta clase, por tanto, también utilizará TAPI para cumplir su objetivo.
- **Llamada.** Clase encargada de encapsular los atributos relacionados con las llamadas telefónicas para un fácil procesamiento de los eventos generados y las solicitudes recibidas desde el servidor principal.
- **Comunicación.** Clase encargada del control y gestión de la comunicación establecida con el servidor principal a través de *sockets*.
- **Protocolo.** Clase abstracta cuya finalidad es la de proporcionar el formato y los nombres utilizados en el protocolo de comunicaciones descrito con anterioridad.
- **Parámetros.** Clase utilizada para la recuperación de atributos de carácter específico de este componente, como números de puerto, archivos de configuración o imágenes e íconos del sistema.
- **Diálogo Principal.** Hilo principal que muestra en pantalla los eventos e información del módulo mediante la utilización de las principales clases anteriormente especificadas.

En lo que respecta al intercambio de mensajes con el servidor principal, con base en el protocolo definido en la sección “Comunicaciones” (EAD-COM), el diagrama de actividades 2.22 tiene por finalidad detallar el proceso de recepción y procesamiento de mensajes enviados por el servidor principal que serán claves en el funcionamiento de este módulo. Los mensajes en cuestión estarán divididos en tres categorías:

- **CTI**, relacionado con eventos y solicitudes que gestionen o administren directamente el estado de las líneas telefónicas.
- **SESSION**, cuyo propósito es asignar o deshabilitar los recursos concedidos a los usuarios.
- **COMMUNICATION**, encargado de responder los mensajes de *keepalive* que determinan la disponibilidad de este componente en el sistema.

Cabe aclarar que la gestión de los mensajes relacionados con el cambio de estado de las líneas telefónicas ya ha sido detallada en la sección 2.4.2.3.2 (EAD-ENV-AGT), por lo que el diagrama de actividades únicamente describe el flujo de eventos para los mensajes de sesión (SESSION) y comunicaciones (COMMUNICATION).

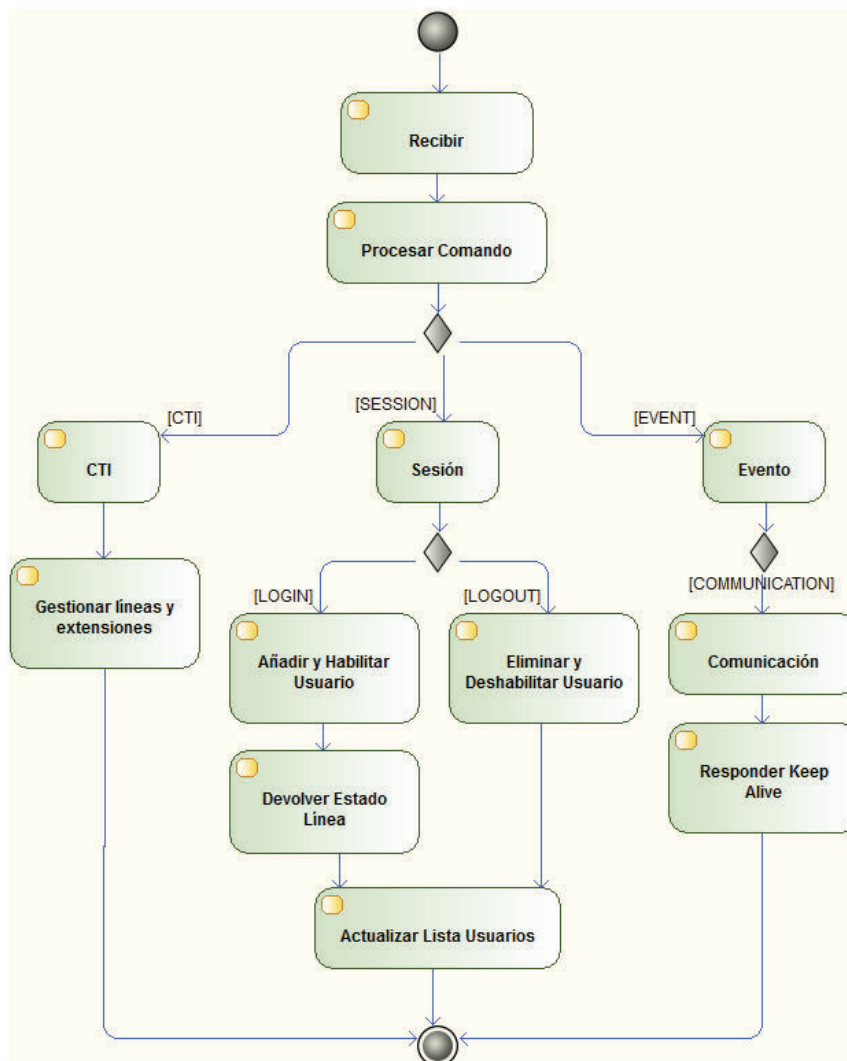


Figura 2.22 Procesamiento de mensajes en el servidor CTI

En lo correspondiente a la interfaz gráfica de usuario, se ha considerado importante que este servidor posea una ventana en la que sea posible visualizar el estado de la comunicación, número de líneas telefónicas detectadas, sesiones activas, eventos del sistema, entre otros. Aunque es verdad que la creación de una interfaz de este

tipo es innecesaria puesto que el componente en cuestión no tiene interacción alguna con los usuarios finales (al estar interconectado con la central telefónica y con el servidor principal de forma automática), razones académicas y de pruebas motivan tal construcción para facilitar la lectura de la información producida y determinar el estado del sistema. Debido a su sencillez, estará conformado por una sola ventana de tipo cuadro de diálogo como se muestra a continuación:

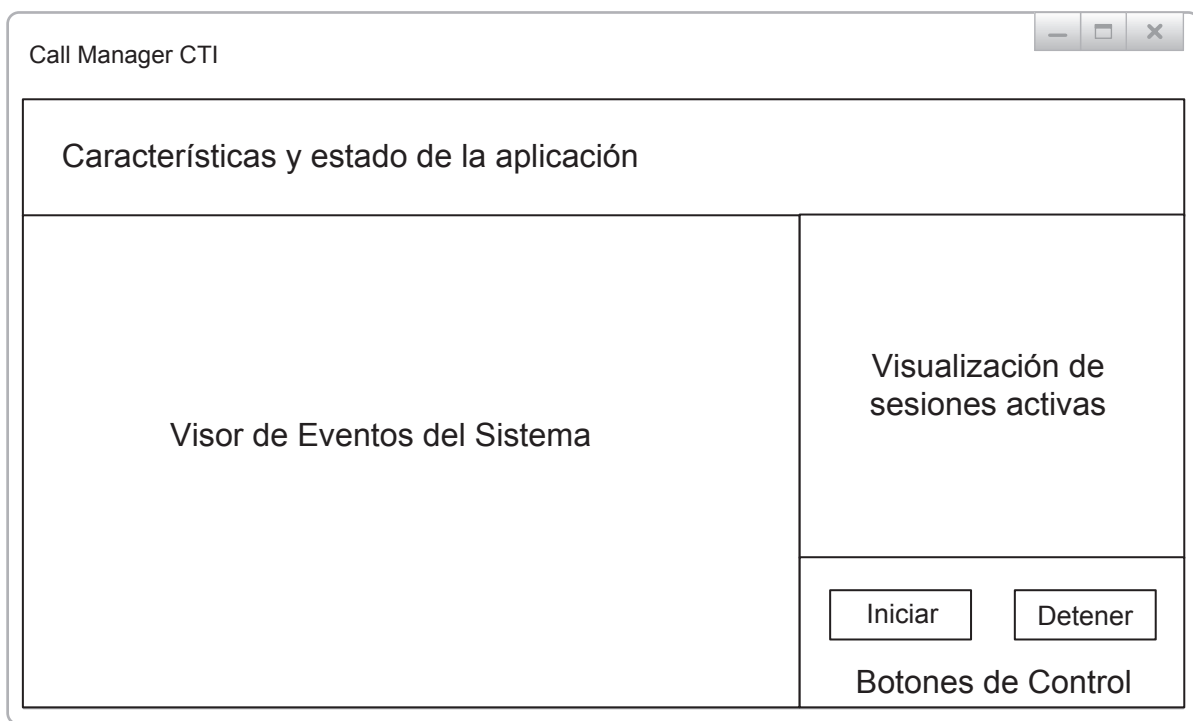


Figura 2.23 Ventana del módulo CTI de *Call Manager*

2.4.2.7. Servidor Principal (EAD-CORE)

Es el componente de mayor complejidad del sistema, punto central de intercambio de información entre los módulos de agente, bases de datos, y aplicativo CTI, así como el encargado de presentar una interfaz de administración y configuración a supervisores y administradores de la plataforma.

Tomando como base lo descrito en la sección 2.3.4.3.4 (IA-AA-CORE) la complejidad de este bloque funcional requiere el seccionamiento de su análisis y diseño para facilitar su comprensión y posterior desarrollo.

2.4.2.7.1. Componentes de Negocio (EAD-CORE-Bss)

Este tipo de componentes es vital en el funcionamiento de cualquier aplicación que sea desarrollada según el modelo de aplicaciones distribuidas multinivel de Java EE. Es en esta capa (o nivel) donde se proporciona de forma centralizada la lógica de negocio de la aplicación, como por ejemplo, insertar o recuperar información de las bases de datos, procesar los comandos transferidos entre los módulos de agente y CTI o de monitorear el estado de los componentes del sistema; todo de forma transparente para el usuario final.

El primer paso en la creación de este módulo consiste en la especificación de un objeto que proporcione un único punto de acceso global a los servicios proporcionados por la plataforma, centralizando la administración y reduciendo la complejidad. Para ello se creará un EJB de tipo sesión concebido bajo el uso del patrón *singleton* con nombre **MainController**, una única instancia que operará durante el ciclo de vida de la aplicación e iniciará de forma automática desde el arranque del servidor de aplicaciones. De forma adicional, al tratarse de un componente distribuido, es importante considerar la posibilidad de interactuar con los demás elementos del sistema mediante una interfaz que abstraiga los principales métodos utilizados por ellos, de tal forma que sea posible modificar o actualizar las tecnologías (o plataformas) implicadas sin necesidad de alterar completamente el código subyacente. Para ello, la interfaz **CommunicationHandler** encapsulará las funciones relacionadas con la sesión de los agentes y el envío y recepción de eventos, comandos y notificaciones.

Por motivos de rendimiento cada mensaje generado por los agentes de escritorio o el módulo CTI tiene que ser procesado fuera de **MainController** (al ser creado bajo el patrón *singleton*). Para ello se utilizará un EJB sin estado (*stateless*) denominado **EventController**, que será responsable de seccionar los mensajes recibidos

conforme al protocolo de comunicaciones definido con anterioridad y prestar los servicios requeridos en caso de ser necesario, todo de forma simultánea e independientes entre sí. En otras palabras, aun cuando los mensajes serán recibidos a través de *MainController*, la carga del procesamiento de éstos será transferida a *EventController*, con lo cual se evitarán las colisiones de los mensajes o un incremento en los tiempos de respuesta del sistema debido al manejo concatenado de las solicitudes desde un mismo hilo de ejecución. Al ser un *bean* sin sesión, *EventController* desconoce de las identidades de los usuarios que utilizan el sistema, por lo que tendrá que usar los métodos proporcionados por *MainController* para transmitir o actualizar la información relevante según se requiera. El diagrama de comunicaciones 2.24 resume lo anteriormente descrito.

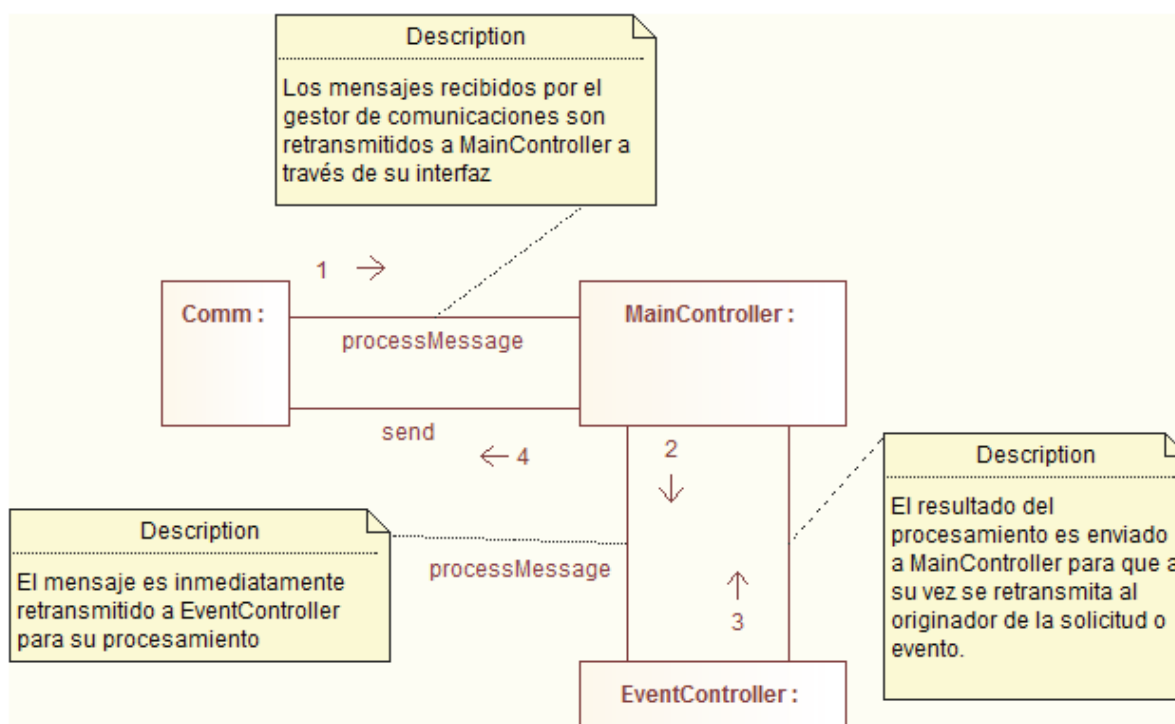


Figura 2.24 Principales Componentes de Negocio y Procesamiento de Mensajes

Por otro lado, teniendo como objetivo proporcionar un rápido funcionamiento de la solución, se considerará que cualquier objeto que forme parte del módulo principal que precise acceder a la información almacenada en las bases de datos lo haga

directamente a través de *beans* de sesión sin estado (*stateless EJB*) dedicados a tal tarea, evitando tener que pasar por *MainController* en cada ocasión para el efecto.

Finalmente, el diagrama de actividades 2.25 detalla cómo serán divididos y procesados todos los mensajes recibidos en este módulo según lo definido en el protocolo de comunicaciones descrito en secciones anteriores.

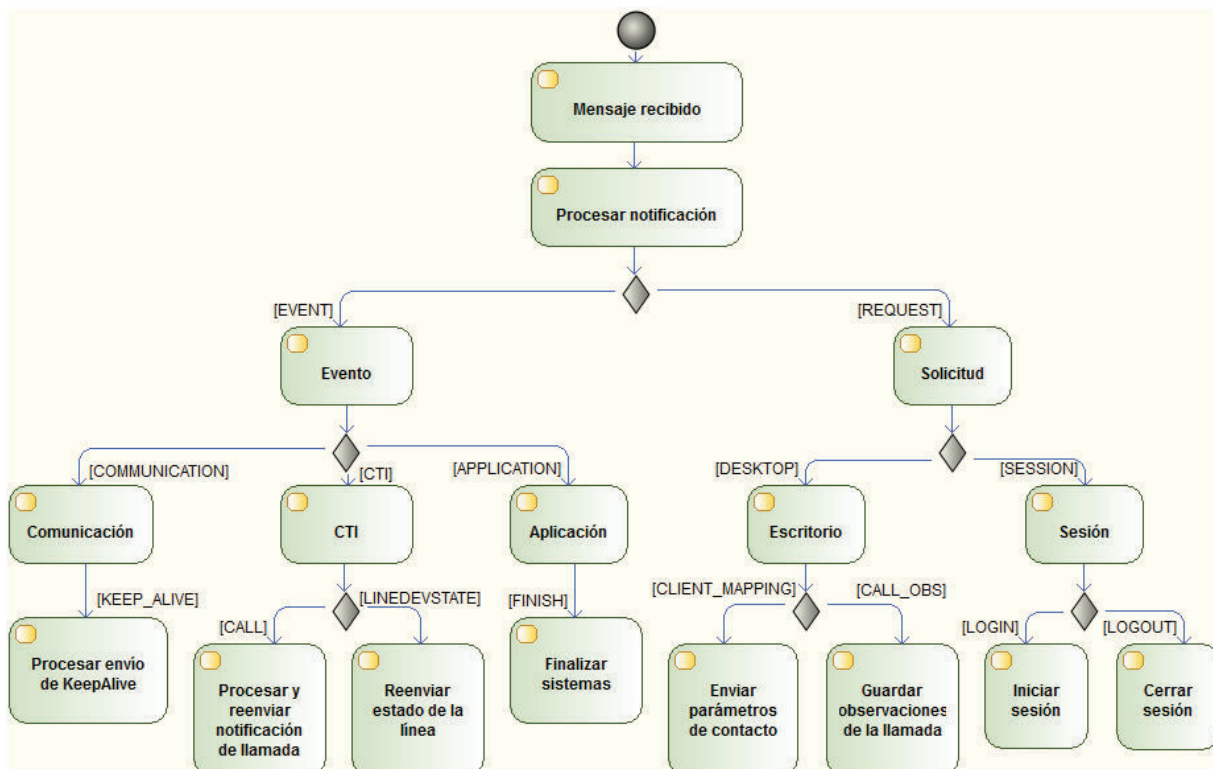


Figura 2.25 Procesamiento de mensajes recibidos en el servidor principal

Como es posible apreciar el servidor principal recibirá dos tipos de mensajes: EVENT y REQUEST. Los primeros están relacionados con los cambios de estado de las líneas telefónicas, notificaciones del estado de los componentes internos de la solución o cualquier tipo de mensaje automatizado en el que no existe intervención del usuario final del aplicativo. Esto quiere decir que los mensajes de *heartbeat* enviados desde el servidor CTI también son procesados como mensajes de tipo EVENT. Por otro lado, las notificaciones de tipo REQUEST están relacionadas directamente con las actividades de los usuarios del sistema (agentes). Es por ello

que en este caso solicitudes de inicio o cierre de sesión así como requerimientos de información de llamada, parámetros de contactos y observaciones de las llamadas serán procesadas separadamente.

2.4.2.7.2. Componentes Web (EAD-CORE-Web)

Los componentes web serán los encargados de proporcionar una interfaz de administración y uso de la plataforma tanto a supervisores como a administradores, que puede ser accedida por medio de un navegador web. Dentro del conjunto de tecnologías disponibles en Java EE, la responsable del desarrollo de aplicaciones web es *JavaServer Faces (JSF)*^[9] que agrupa un diverso conjunto de componentes como son *beans* administrados (*managed beans*), validadores (*validators*), escuchadores (*listeners*), convertidores (*converters*), y otros objetos del lado del servidor que están asociados con las páginas web. Estos elementos y en especial los *beans* administrados interactuarán con bases de datos y otros componentes de negocio (como *MainController* por ejemplo) para recuperar información relevante del sistema como el estado de la comunicación, los agentes conectados o las llamadas realizadas. El diagrama 2.26 resalta lo anteriormente expresado, donde se representa la creación dinámica de páginas web (generadas en el contenedor web) según los datos recuperados desde el nivel de negocio (contenedor EJB) que a su vez podría requerir información de una base de datos.

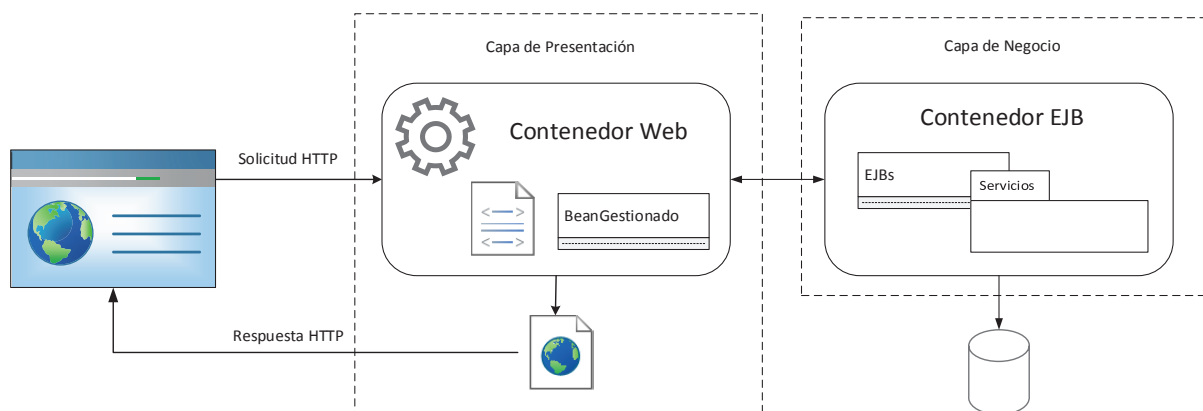


Figura 2.26 Respuesta a la solicitud de un cliente según el modelo JSF

Si bien los *beans* administrados podrían interactuar directamente con las bases de datos a través de EJBs locales ubicados en el mismo contenedor web, se ha considerado apropiado centralizar el acceso a ellas desde el contenedor de negocio con el objetivo de reducir la fragmentación de código y facilitar el desarrollo del sistema. Adicionalmente, cada *bean* administrado puede guardar información cuyo tiempo de vida o alcance (*scope*) dependerá de las necesidades de cada perfil, pudiendo persistir durante el transcurso de la petición (*request*) solamente, una vista o *wizard* de configuración (*view*), la sesión activa del usuario (*session*) o durante la entera ejecución de la aplicación (*application*).

A continuación, considerando que existen tres tipos de roles en el sistema, será conveniente crear páginas de servicio para cada uno de ellos. Aunque podría considerarse innecesaria la opción de proporcionar gestión web para los agentes del *call center* (puesto que utilizan su correspondiente aplicativo dedicado) es importante definir un espacio de administración propio con el propósito de brindar servicios adicionales, facilitar la distribución de software o actualizaciones o proporcionar flexibilidad (respaldo futuro a nuevas características o incluso una posible migración del aplicativo de escritorio a un entorno *Rich Internet Applications* o RIA en donde no exista la necesidad de instalar ni ejecutar un aplicativo dedicado). Por tal razón, cualquier usuario será capaz de ingresar al sistema web a través de una única pantalla de inicio de sesión que luego de validar sus credenciales lo redirigirá de acuerdo a su perfil a una nueva página principal (*home* de cada usuario) que contenga información general o relevante. Por tal motivo, a continuación se presentará un esquema según el cual estará organizada la distribución de las páginas web del sistema a desarrollar:

1. **Pantalla de inicio de sesión.** Todos los usuarios deberán ingresar sus credenciales de acceso para poder ser redirigidos a sus respectivos perfiles.
2. **Administrador**
 - a. **Página principal.** Esta página mostrará el estado de la comunicación y el número de agentes conectados al sistema en ese momento.

- b. Puestos de trabajo.** En esta sección en administrador será capaz de crear, borrar, y editar las extensiones que serán utilizadas por los agentes que forman parte del *call center*.
- c. Usuarios.** Aquí será posible añadir, editar, o borrar usuarios (administradores, supervisores o agentes).
- d. Comunicaciones.** El administrador configurará los puertos de conexión y escucha hacia el servidor CTI y los agentes respectivamente, también será capaz de interrumpir o reiniciar el estado de la comunicación entre los componentes antes mencionados.
- e. Planes de numeración.** En esta sección será posible cargar los planes de numeración del Ecuador con el propósito de identificar llamadas (véase requerimiento 2.3.5.2.3, IA-AR-3).
- f. Variables del sistema.** La plataforma debe permitir el reajuste de parámetros internos y/o externos, como el número máximo de intentos de reconexión o el código de área local, por ejemplo.
- g. Integración con bases de datos.** En esta página se podrán configurar los parámetros de interconexión con bases de datos externas compatibles con la solución.
- h. Ayuda.** Guía de utilización/navegación del sistema web.
- i. Cierre de sesión.**

3. Supervisor

- a. Página principal.** En este lugar el supervisor podrá visualizar todos los agentes conectados al sistema, así como el estado de las líneas telefónicas asociadas a ellos (entiéndase llamadas).
- b. Reportes.** Directamente relacionados con el rendimiento del *call center*. Esta sección agrupa lo siguiente:
 - i. Desempeño de agentes.**
 - ii. Llamadas.** Visualización de las llamadas procesadas por el sistema.
 - iii. Operadora.** Estadística de las llamadas procesadas según la operadora.

iv. **Provincia.** Mapa de calor que destaque la frecuencia de las llamadas para números telefónicos fijos según la provincia.

v. **Ayuda.**

vi. **Cierre de sesión.**

4. Agente

a. **Página principal.** Página que mostrará el historial de las últimas llamadas telefónicas procesadas por el agente.

b. **Perfil.** El agente podrá actualizar su perfil según crea conveniente.

c. **Agente de escritorio.** Detalles de la descarga, instalación, configuración, o uso del aplicativo de escritorio.

d. **Ayuda.**

e. **Cierre de sesión.**

5. **Página de cierre de sesión.** El usuario que haya cerrado sesión será redirigido hacia una página que notifique el resultado.

En lo correspondiente al diseño de las interfaces web de usuario, en la figura 2.27 se concibe la plantilla con base en la cual serán creadas todas las páginas web del sistema para administradores, supervisores y agentes. Las secciones que componen dicha plantilla son:

- **Cabecera.** Contiene el logotipo del proyecto, así como el nombre “Supervisión” o “Administración” según sea el caso.
- **Menús.** Listado que contiene las distintas funciones o características según el rol (por ejemplo, administrar usuarios, puestos de trabajo, comunicación, etc.). La selección de una de ellas actualizará la información desplegada en el área de contenido.
- **Navegación.** Opcional según la situación lo requiera, será utilizado para navegar entre las subcategorías de cada menú.
- **Contenido.** Contendrá la información de administración o supervisión propiamente dicha: *wizards* de configuración, tablas, reportes, etc.

- **Pie.** Desplegará información adicional respecto al creador del proyecto y la universidad.

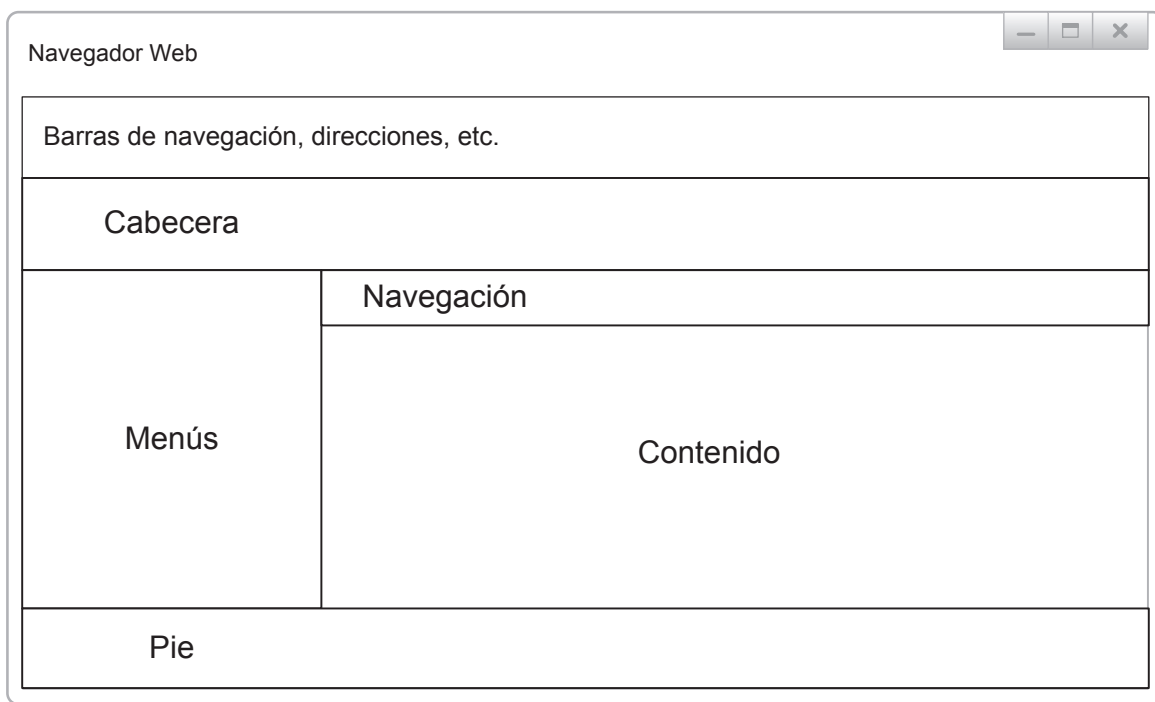


Figura 2.27 Esquema de Construcción de Páginas Web de *Call Manager*

2.4.2.8. Cliente de Escritorio (EAD-DSK)

Conforme a lo definido en la fase de inicio en el análisis de la arquitectura (IA-AA), el cliente de escritorio tiene por objeto enviar y recibir eventos en tiempo real que pueden ser generados por el agente (como por ejemplo, el cierre de sesión), por el sistema (se ha perdido la comunicación) o por la línea telefónica (se ha recibido una nueva llamada telefónica que tiene que ser atendida).

Este módulo, al tener una mayor interacción con el usuario final (que es el agente de *call center*) requiere de un minucioso análisis de las principales características y funcionalidades que dispondrá, presentado a continuación.

2.4.2.8.1. Procesamiento de Mensajes

Los eventos y mensajes serán procesados directamente desde el hilo principal de la aplicación ya que su frecuencia no incide en el desempeño general del aplicativo. En otras palabras, aunque habrá un hilo dedicado exclusivamente al manejo de la comunicación (envío y recepción de mensajes) el procesamiento de éstos será efectuado directamente en el hilo principal de la aplicación, por su carácter inmediato y de fácil ejecución que no influye de forma notable en el rendimiento del programa. Por tal razón se hará uso del patrón *Observer/Observable* según el cual un evento de actualización desencadenado en un objeto observado (que en este caso particular será llamado **DeskComm**) generará una reacción en otro que lo observa (observador llamado **CallManagerDesktop**).

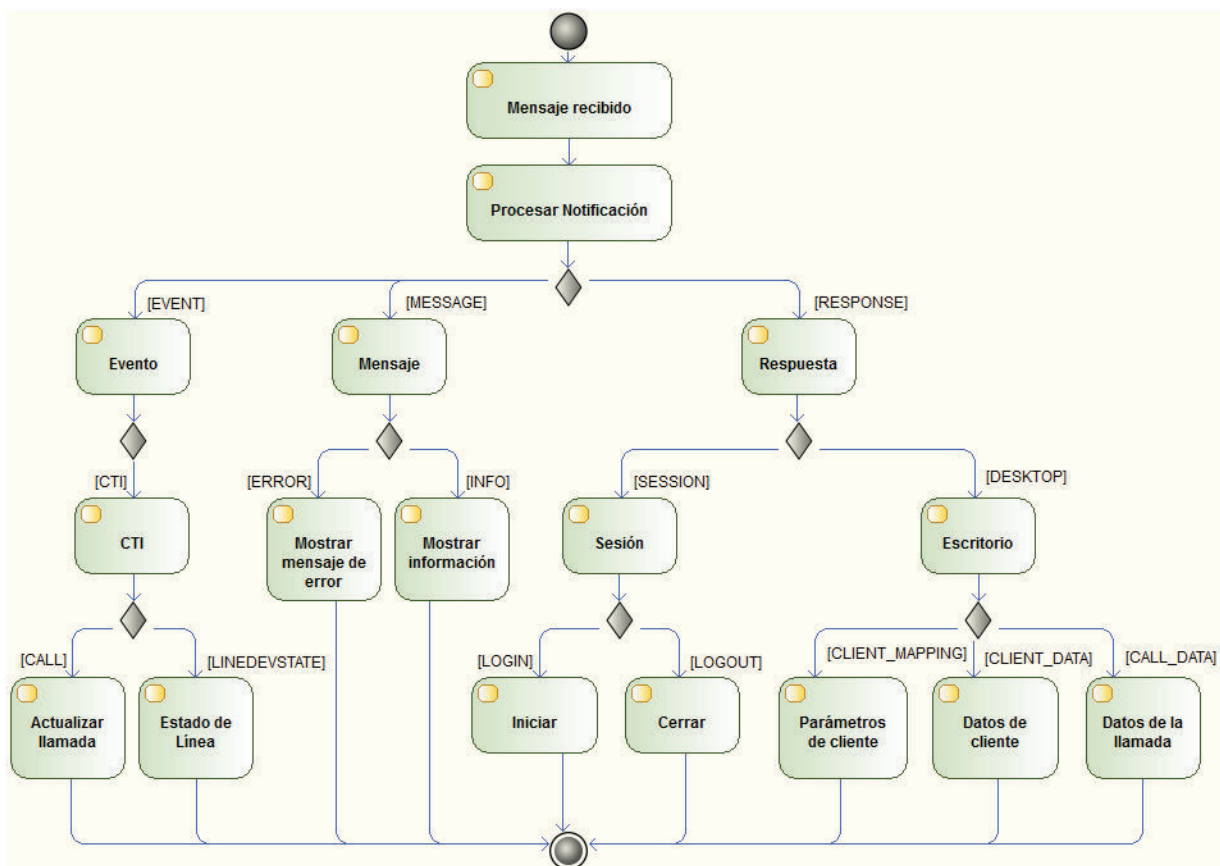


Figura 2.28 Procesamiento de Mensajes Recibidos en el Agente de Escritorio

El árbol de procesamiento de los mensajes recibidos será efectuado de acuerdo al protocolo definido en la sección de análisis y diseño de comunicaciones (EAD-COM) así como dispuesto en el flujo de actividades representado en el diagrama anterior, en el que las cadenas de mensajes serán clasificadas de acuerdo al tipo de notificación como eventos, mensajes y respuestas. Los eventos estarán directamente relacionados con el estado del sistema y de las líneas telefónicas, lo que significa que el ingreso de una llamada será tratada mediante este flujo, por ejemplo. Los mensajes son notificaciones generales como el aviso de desconexión del servidor o un comentario realizado por el supervisor para un agente particular. Las respuestas, en cambio, están asociadas con el flujo activo de información existente entre el agente y el módulo principal, como es el caso del inicio de sesión o la actualización de la información geográfica de una llamada en curso.

2.4.2.8.2. Diseño de la Interfaz

Tomando en cuenta que el aplicativo de escritorio del agente envía y recibe eventos en tiempo real, es importante necesario contar con una ventana que reúna la mayor cantidad de información relevante sin afectar la complejidad de interpretación de la misma, al mismo tiempo que ofrezca permanentemente la posibilidad de controlar y gestionar la línea telefónica. El diagrama 2.29 reúne las características antes mencionadas.

Hay tres áreas a las que se les prestará una especial atención:

- **Control de la línea telefónica.** Similar a una barra de herramientas, esta sección contendrá un grupo de botones, cada uno de los cuales permitirá realizar alguna funcionalidad sobre la línea telefónica: llamar, colgar, transferir, etc.
- **Visualización de llamadas.** Estará compuesta por una tabla en la que se visualizará(n) la(s) llamada(s) activa(s), cuyo orden estará definido por el número de identificación proporcionado por la central telefónica. La información que podrá ser visualizada será:
 - Número de identificación de la llamada.

- Número de origen (llamante).
 - Número de destino (llamado).
 - Estado de la llamada (marcando, timbrando, conectado, etc.)
 - Botón para el ingreso de observaciones.
- **Información de la llamada en proceso.** Esta sección deberá contener el siguiente tipo de información.
 - Información de los contactos recuperados desde la base de datos de la organización (en el caso que existan datos disponibles).
 - Información de georreferenciación de la llamada. En caso de haberse encontrado información de geo-localización, ésta será visualizada en un mapa.
 - Nombre de la operadora y provincia (sólo números fijos) extraída de los planes de numeración.

Este campo, al contener datos variables y dependientes sólo de las llamadas activas, deberá estar oculto todo el tiempo mientras la línea esté inactiva.

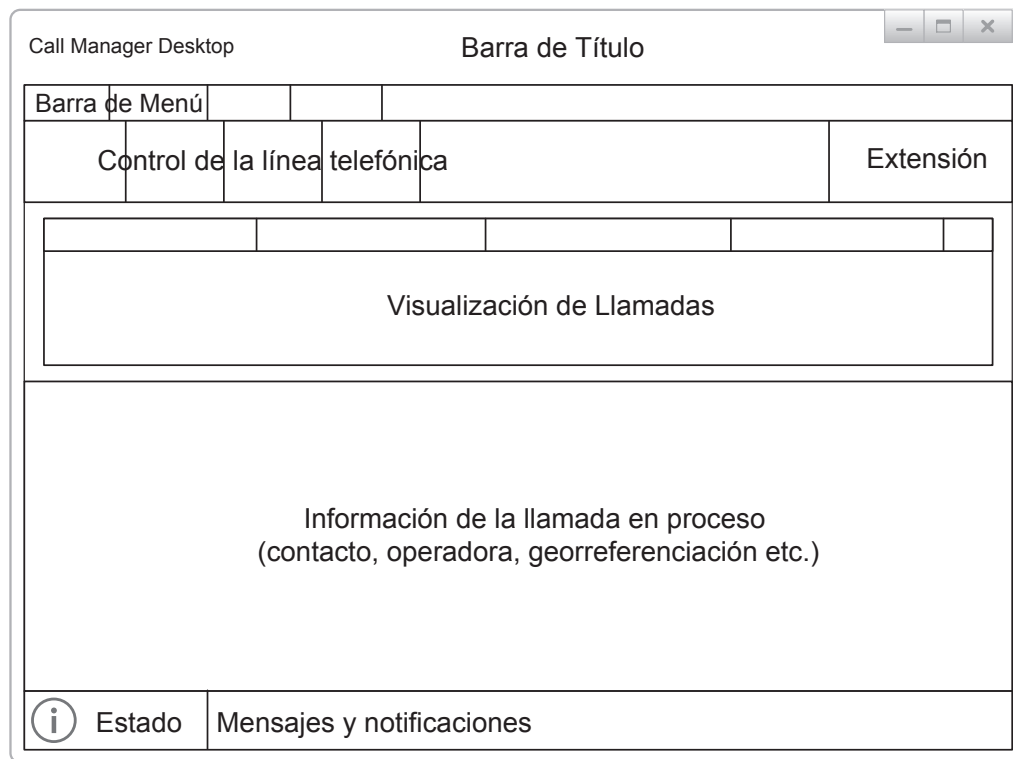


Figura 2.29 Ventana principal del aplicativo de escritorio del agente

Como nota adicional, no se detallarán los cuadros de diálogo adicionales del aplicativo de escritorio debido a su simplicidad o poca relevancia en esta fase de diseño.

2.4.2.9. Base de Datos (EAD-BDD)

En el diseño de la base de datos **CallManagerDB** local han sido considerados varios factores, entre los más importantes los siguientes:

- La tabla **callrecord** será la encargada de almacenar los registros concernientes a las llamadas telefónicas que hayan sido atendidas (conectadas) por el *call center*. El modelo no está enfocado en la creación de registros acumulados tipo *warehouse*.
- La tabla **appuser** es la encargada de asociar los datos personales de los usuarios con el rol correspondiente (administrador, supervisor o agente) y con las líneas telefónicas (tabla **phoneline**) de ser el caso.
- Los rangos numéricos correspondientes a los planes de numeración (tabla **numberedplan**) están estrechamente relacionados con los códigos de zona (tabla **zonecode**) y las provincias (tabla **province**) por separado. También se relacionarán con la tabla **serviceoperator** de la cual será posible obtener el nombre de las operadoras del servicio de telefonía fija y móvil del Ecuador.
- La tabla **contactcenter** contiene un único registro en el que constan los parámetros específicos de la central telefónica Avaya IP Office. La creación de esta tabla fue considerada con propósitos de proveer flexibilidad con otras plataformas de telefonía a futuro.
- La tabla **systemparams** contendrá los parámetros específicos del sistema, como números de puerto, delimitadores, cadenas de interconexión con bases de datos, etc. Esta opción es preferida en lugar de la creación de archivos de configuración por la facilidad que ofrece el API de persistencia de Java.
- La tabla **calldetailinfo** será la responsable de guardar la información adicional de los contactos involucrados en una llamada, en el caso que haya sido posible obtener la información de éstos desde la base de datos interna de la

organización. Para ello obviamente necesariamente tienen que haber sido configurados los parámetros de interconexión con bases de datos desde el sistema de administración.

En lo que respecta al procedimiento de acceso a datos desde el servidor principal deben tomarse en cuenta dos escenarios:

a) Acceso a Información Desde la Base de Datos Local

Para el acceso a la información localizada en la base de datos local del aplicativo se utilizará el API de Persistencia de Java (JPA), el cual tiene por objeto utilizar objetos Java planos (o POJOs) en la interacción con bases de datos relacionales. Las *entidades* representan tablas de la base de datos relacional, cuyas instancias corresponden a las filas (registros) de dichas tablas. Es por ello que se necesitará un paquete llamado **Entities** que contendrá las entidades que mapean las tablas del sistema Call Manager.

Adicionalmente, será creado un paquete denominado **Facade** (fachada) que albergará *beans* sin estado (o *stateless EJBs*) que serán los encargados de proporcionar los métodos necesarios para que otros componentes del sistema *Call Manager* puedan acceder y modificar la información mediante el empleo de JPA. Dichos *beans* heredan de la clase *AbstractFacade* que emplea la unidad de persistencia *EntityManager* para gestionar todas las clases de tipo *Entities* y así junto al empleo del Lenguaje de Consultas de Persistencia de Java (JPQL, *Java Query Persistence Language*) sea posible crear consultas específicas de acuerdo a las necesidades del proyecto en cuestión.

En lo que respecta al procedimiento mismo de recuperación de la información, cualquier componente del servidor principal (sea este de web o de negocio) será capaz de llamar a los métodos ofrecidos por las fachadas de las tablas. Para ejemplificar, supóngase una llamada entrante originada desde un teléfono fijo. Cuando el flujo procesamiento de eventos en el servidor principal llega al punto en el que es necesario recuperar los valores correspondientes al

código de zona del llamante, el EJB *EventController* encargado de dicho procesamiento invocará a otro EJB denominado *DataAccess* del paquete *Facade* para utilizar el método *getZoneCodeByValue()* que incluye una consulta JPQL que a su vez utilizará el *Entity* denominado *ZoneCode* para de esta manera recuperar la información de la base de datos mediante el API subyacente. Del mismo modo los *beans gestionados* y las páginas web del componente web del servidor principal serán capaces de acceder a fachadas y entidades para recuperar la información requerida.

Cabe recalcar que debido a que JPA es una especificación, su respectiva implementación deberá ser previamente instalada y configurada según lo requiera el servidor de aplicaciones. Esta tarea corresponde a la etapa de implementación de la fase de construcción del proyecto descrita en el siguiente capítulo.

b) Acceso a Información Desde Bases de Datos Externas

De acuerdo a lo establecido en sección 2.4.2.4 (EAD-INF) el acceso a información de los contactos de las llamadas procesadas en el *call center* empleará el estándar de la industria de conectividad de bases de datos de Java, es decir el API JDBC (conectividad de bases de datos de Java) para recuperar los datos de una única tabla o vista en modo de sólo lectura. Debe notarse que JDBC es diferente a JPA en el sentido que el primero utiliza directamente el lenguaje SQL para la realización de consultas, mientras que el API de persistencia encapsula las tablas en entidades y realiza las consultas mediante JPQL, motivo por el cual no es adecuado utilizar JPA para interconectar bases de datos de distintos fabricantes ya que las tablas a utilizar pueden variar entre una y otra organización. En este sentido, JDBC es la opción más adecuada aunque de más bajo nivel. De entre la enorme variedad de bases de datos compatibles con JDBC, en este proyecto se soportará interconectividad con aquellas más populares como son: **MySQL**, **Microsoft SQL Server** y **PostgreSQL**.

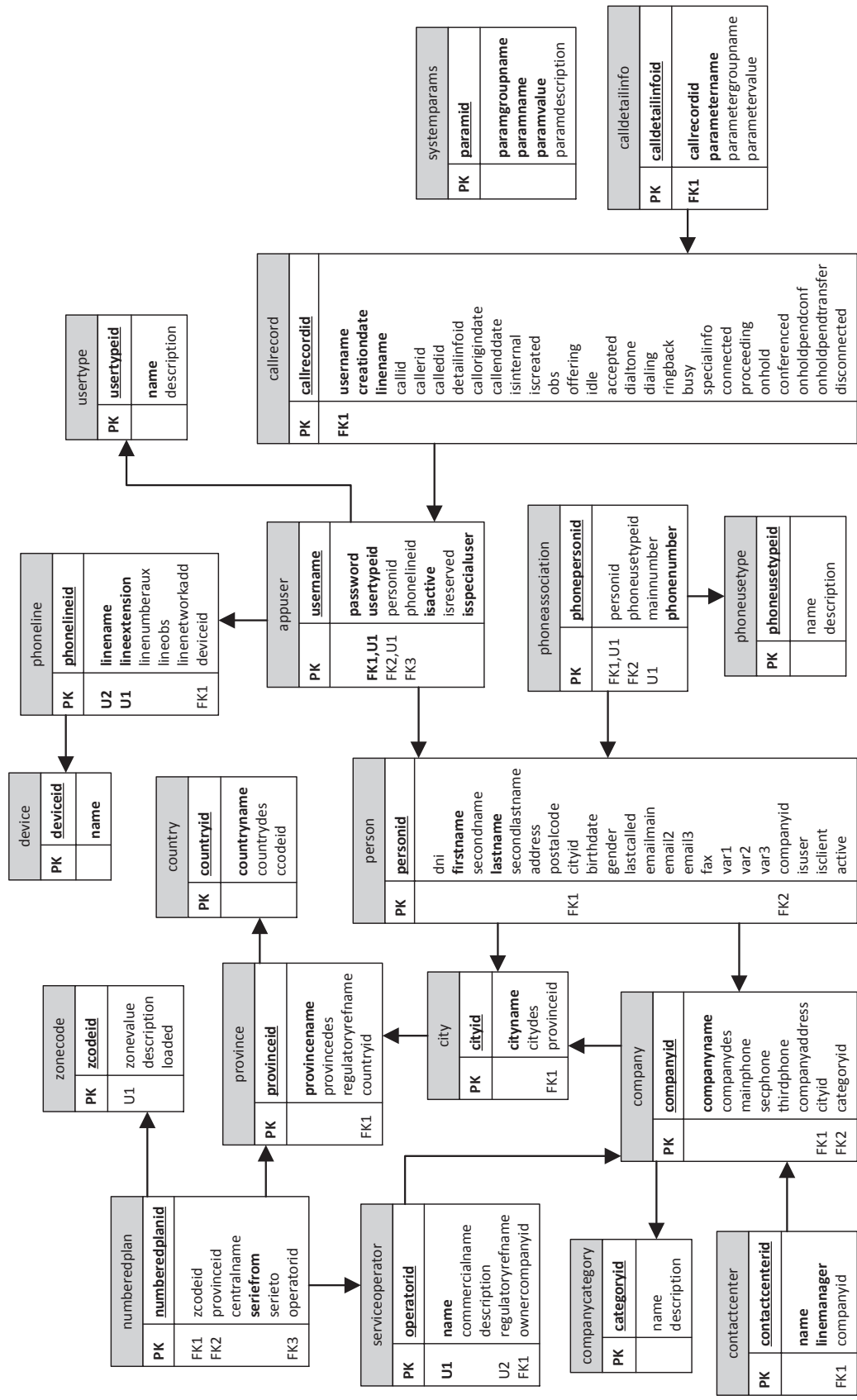


Figura 2.30 Diseño de Base de Datos