

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

ADMINISTRACIÓN DE UN CALL CENTER DE COBRANZA

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
INFORMÁTICA
MENCIÓN SOFTWARE**

IRINA ALEXANDRA ORQUERA JÁTIVA
iorquera@hotmail.com

GIOVANNA PATRICIA MUÑOZ CRUZ
giovymunoz@gmail.com

DIRECTOR: ING. MSc. CARLOS ESTALESMIT MONTENEGRO ARMAS
carlos.montenegro@epn.edu.ec

Quito, Mayo 2015

DECLARACIÓN

Yo Giovanna Patricia Muñoz Cruz, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada 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.

Giovanna Patricia Muñoz Cruz

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Irina Alexandra Játiva Orquera y Giovanna Patricia Muñoz Cruz, bajo mi supervisión.

Ing. Carlos Montenegro Armas
DIRECTOR DE PROYECTO

AGRADECIMIENTOS

Agradezco primeramente a Dios, quien me ha guiado con su infinito amor hasta éste punto de mi camino por la vida, dándome la sabiduría y poder de entendimiento de lo que necesito para cumplir con mi misión en éste mundo.

A los profesores, autoridades y compañeros, de la ESCUELA POLITÉCNICA NACIONAL, por la buena educación que me han impartido.

Agradezco también de todo corazón a mis padres, Charito, Anita y Pedrito, a mis hermanos Fernando, Cynthia, Roberto y Pedro, quienes han sido siempre mi respaldo y mi apoyo incondicional.

Y a toda la gente que me incentivó y alentó de alguna manera para seguir adelante en éste proyecto.

DEDICATORIA

Este gran logro en mi vida lo dedico a mis grandes amores Anahí, Ariana, Mayté, Emily y Said.

Lo que más añoro es saber que cada uno de ustedes va crecer aprendiendo todo lo bueno de la vida para que sean personas buenas, valiosas y triunfadoras.

Que Dios les bendiga siempre.

CONTENIDO

1. FUNDAMENTOS TEÓRICOS	13
1.1. INTRODUCCIÓN A CALL CENTER	13
1.1.1. DEFINICIÓN DE CRM (Customer Relationship Management)	13
1.1.2. DEFINICIÓN DE CALL CENTER	17
1.1.3. CALL CENTER CON ORIENTACIÓN CRM	17
1.1.3.1. Retención de clientes	18
1.1.3.2. Adquisición y crecimiento de clientes	19
1.1.3.3. Conocimiento de clientes	19
1.1.4. ANTECEDENTES Y EVOLUCIÓN DEL CALL CENTER	20
1.1.5. OBJETIVOS	22
1.1.6. COMPONENTES	23
1.1.7. TIPOS	25
1.2. TECNOLOGÍA APLICADA A UN CALL CENTER	26
1.2.1. INFRAESTRUCTURA	27
1.2.1.1. Puestos de Trabajo	27
1.2.1.2. Comunicaciones	27
1.2.1.3. Arquitectura de red	32
1.2.2. SOLUCIONES TECNOLÓGICAS	34
1.2.2.1. Conversión de texto a voz (TTS Text to Speech)	34
1.2.2.2. Reconocimiento Automático de voz (ASR)	34
1.2.2.3. Fax a Demanda	35
1.2.2.4. Respuesta Interactiva de voz (IVR - Interactive Voice Response) ...	36
1.2.2.5. Distribución Automatizada de llamadas (ACD - Automatic Call Distributor)	38
1.2.2.6. Integración de Telefonía Computador (CTI – Computer Telephony Integration)	39
1.2.2.7. Mailing Telefónico Automático (MTA)	40
1.2.2.8. Grabación de llamadas	42
1.2.2.9. Middleware	43
1.2.3. SISTEMAS DE MARCACIÓN	44
1.2.3.1. Marcación automática	44
1.2.3.2. Marcación progresiva	45
1.2.3.3. Marcación predictiva	46
1.3. ADMINISTRACIÓN EFECTIVA DE UN CALL CENTER	48
1.3.1. SUPERVISIÓN DE GESTIÓN	48
1.3.1.1. Flujo de una llamada	49
1.3.1.2. Indicadores Clave	52
1.3.1.3. Early Warning Monitor (EWM)	54
1.3.2. AUDITORIA DE GESTIÓN	58
1.3.2.1. Monitoreo	58
1.3.2.2. Balance Score Card	61
2. REQUERIMIENTOS, ANÁLISIS Y DISEÑO	64
2.1. REQUERIMIENTOS	66
2.1.1. REQUERIMIENTOS FUNCIONALES	66
2.1.2. REQUERIMIENTOS DE INTERFACES EXTERNAS	67

2.1.2.1.	Usuarios.....	67
2.1.2.2.	Hardware.....	67
2.1.2.3.	Software	68
2.1.3.	REQUERIMIENTOS DE DESEMPEÑO.....	69
2.2.	ANÁLISIS.....	69
2.2.1.	DESCRIPCIÓN DEL PROBLEMA	69
2.2.2.	MODELO DE OBJETOS.....	70
2.2.2.1.	Identificación de Clases	70
2.2.2.2.	Identificación de Asociaciones y Agregaciones.....	71
2.2.2.3.	Identificación de Atributos.....	72
2.2.2.4.	Diccionario de datos Clases y Atributos	72
2.2.2.5.	Diagrama del modelo de Objetos.....	78
2.2.3.	MODELO DINÁMICO.....	79
2.2.3.1.	Identificación de escenarios	79
2.2.3.2.	Identificación de sucesos entre objetos	79
2.2.3.2.1.	Diagrama de seguimiento de sucesos para una PLANTILLA.....	79
2.2.3.2.2.	Diagrama de seguimiento de sucesos para una EVALUACIÓN	80
2.2.3.2.3.	Diagrama de seguimiento de sucesos para un INDICADOR CLAVE	80
2.2.3.2.4.	Diagrama de seguimiento de sucesos para una CALIFICACIÓN.....	81
2.2.3.2.5.	Diagrama de seguimiento de sucesos para un EWM.....	81
2.2.3.3.	Definición de sucesos.....	82
2.2.3.4.	Diagrama de estados.....	84
2.2.3.4.1.	Diagrama de estados de una PLANTILLA.....	84
2.2.4.	MODELO FUNCIONAL.....	85
2.2.4.1.	Diagrama de Flujo de Datos para el subsistema EWM.....	85
2.2.4.2.	Diagrama de Flujo de Datos para el subsistema EVALUACIÓN	86
2.3.	DISEÑO DEL SISTEMA.....	87
2.3.1.	ORGANIZACIÓN DEL SISTEMA EN SUBSISTEMAS	87
2.3.1.1.	Capas	87
2.3.1.2.	Particiones.....	89
2.3.2.	IDENTIFICACIÓN DE LA CONCURRENCIA.....	90
2.3.3.	ASIGNACIÓN DE SUBSISTEMAS A PROCESADORES Y TAREAS 90	
2.3.4.	ESTRATEGIA PARA ALMACENAMIENTO DE DATOS	91
2.3.5.	MANEJO DE RECURSOS GLOBALES	93
2.3.6.	CONTROL DEL SOFTWARE	94
2.4.	DISEÑO DE OBJETOS	97
2.4.1.	DIAGRAMA FINAL DEL MODELO DE OBJETOS	97
2.4.2.	DISEÑO DE LA BASE DE DATOS.....	98
2.4.2.1.	Reglas de Transformación.....	98
2.4.2.2.	Modelo Físico de la Base de Datos	100
2.4.3.	DISEÑO DE ALGORITMOS	100
2.4.3.1.	Algoritmos para la creación de operaciones	100
2.4.3.2.	Algoritmos para la creación de tablas	115
2.4.3.3.	Estructuras capa lógica del negocio	118
2.4.3.4.	Algoritmos capa lógica del negocio	121
3.	IMPLEMENTACIÓN Y PRUEBAS	128
3.1.	IMPLEMENTACIÓN	128
3.1.1.	HERRAMIENTAS UTILIZADAS	128

3.1.1.1.	FRONT-END	129
3.1.1.1.1.	ASP.NET	129
3.1.1.1.2.	C#	131
3.1.1.2.	BACK-END	141
3.1.2.	MENÚ PRINCIPAL DEL PROTOTIPO	143
3.1.3.	ESTÁNDARES PARA IMPLEMENTACIÓN	144
3.1.3.1.	Estándares de Interfaces	144
3.1.3.1.1.	Formatos de Pantallas	144
3.1.3.1.2.	Nomenclatura en Formas	147
3.1.3.2.	Estándares de Programación	148
3.1.3.2.1.	Variables	148
3.1.3.2.2.	Formas, Clases y Procedimientos Almacenados	148
3.1.3.3.	Estándares de Base de Datos	150
3.1.4.	FUNCIONES Y OPERACIONES (COMPONENTES)	151
3.1.4.1.	Arquitectura	151
3.1.4.2.	Componentes	151
3.1.4.2.1.	Componente CONECTION	151
3.1.4.2.2.	Componente CATALOGOS	152
3.1.4.2.3.	Componente AUDITORIA	152
3.1.4.2.4.	Componente EWM	153
3.2.	PRUEBAS	154
3.2.1	PRUEBAS DE UNIDAD	154
3.2.1.1	Definición de pruebas de unidad	155
3.2.2	PRUEBAS DE INTEGRACIÓN	160
3.2.2.1	Definición de pruebas de integración	160
3.2.3	PRUEBAS DE VALIDACIÓN	161
4.	CONCLUSIONES Y RECOMENDACIONES	163
4.1.	CONCLUSIONES	163
4.2.	RECOMENDACIONES	166
5.	REFERENCIAS BIBLIOGRÁFICAS	167
6.	ANEXOS	168
6.1.	MODELO FÍSICO DE LA BASE DE DATOS	168
6.2.	ANEXO 2: CÓDIGO DEL PROTOTIPO	169
6.2.1.	MÓDULO AUDITORIA DE GESTIÓN	169
6.3.	ANEXO 3: MANUAL DE INSTALACIÓN	177
6.4.	ANEXO 4: MANUAL DE USUARIO	191

ÍNDICE DE FIGURAS

Figura 1.1	Arquitectura de CC en los 90's.....	21
Figura 1.2	Arquitectura de CC a futuro.....	21
Figura 1.3	Componentes de un CC	25
Figura 1.4	Línea Digital Asimétrica ADSL.....	28
Figura 1.5	Red Privada Virtual	30
Figura 1.6	Arquitectura Física.....	32
Figura 1.7	Arquitectura Lógica	33
Figura 1.8	Distribución Automatizada de llamadas.....	39
Figura 1.9	Marcación Automática.....	45
Figura 1.10	Marcación Progresiva.....	46
Figura 1.11	Marcación Predictiva.....	47
Figura 1.12	Flujo del resultado de una Gestión	50
Figura 1.13	Árbol de resultado de Gestión de Cobranza.....	51
Figura 1.14	Early Warning Monitor.....	54
Figura 1.15	Selección de variables para monitorear EWM.....	55
Figura 1.16	Acciones Derivadas Early Warning Monitor	57
Figura 1.17	Interfaz de EWM.....	57
Figura 1.18	Calidad de Gestión (Monitoreo)	58
Figura 1.19	Monitoreo	59
Figura 1.20	Indicadores Balance Score Card.....	61
Figura 1.21	Calificación Agente.....	62
Figura 2.1	Estructura de Capas.....	89
Figura 2.2	Subsistemas	90
Figura 2.3	Arquitectura	91
Figura 2.4	Esquema de Respaldo de Datos	92
Figura 2.5	Esquema de Mantenimiento de BDD.....	93
Figura 2.6	Entorno del sistema	95
Figura 2.7	Rack de Servidor.....	96
Figura 3.1	Plataforma Microsoft Visual .NET.....	128
Figura 3.2	SQL Server.....	141
Figura 3.3	Tipos de BDD SQL Server.....	143
Figura 3.4	Menú Principal del Prototipo.....	144
Figura 3.5.:	Formato Pantalla de Acceso.....	144
Figura 3.6.:	Formato Pantalla de Selección de Módulos.....	145
Figura 3.7.:	Formato Pantalla de Menú y Submenú.....	145
Figura 3.8.:	Formato Pantalla de Mantenimiento de Datos.....	146
Figura 3.9.:	Formato Pantalla de Reportes de Datos	146
Figura 3.10.:	Formato Pantalla de Reportes Estadísticos.....	147

ÍNDICE DE TABLAS

Tabla 2.1	Análisis: Diccionario de Datos Clase INDICADOR CLAVE.....	72
Tabla 2.2	Análisis: Diccionario de Datos Clase EWM.....	73
Tabla 2.3	Análisis: Diccionario de Datos Clase EWM INDICADOR CLAVE	73
Tabla 2.4	Análisis: Diccionario de Datos Clase CALIFICACION.....	74
Tabla 2.5	Análisis: Diccionario de Datos Clase PLANTILLA.....	74
Tabla 2.6	Análisis: Diccionario de Datos Clase PREGUNTA.....	75
Tabla 2.7	Análisis: Diccionario de Datos Clase RESPUESTA.....	75
Tabla 2.8	Análisis: Diccionario de Datos Clase EVALUACION.....	76
Tabla 2.9	Análisis: Diccionario de Datos Clase RESULTADO EVALUACION	76
Tabla 2.10	Análisis: Diccionario de Datos Clase ADMINISTRADOR.....	77
Tabla 2.11	Análisis: Diccionario de Datos Clase GESTION.....	77
Tabla 2.12	Análisis: Escenario Seguimiento de Sucesos para una PLANTILLA.....	82
Tabla 2.13	Análisis: Escenario Seguimiento de Sucesos para una EVALUACION.....	82
Tabla 2.14	Análisis: Escenario Seguimiento de Sucesos para un INDICADOR CLAVE.....	83
Tabla 2.15	Análisis: Escenario Seguimiento de Sucesos para una CALIFICACION.....	83
Tabla 2.16	Análisis: Escenario Seguimiento de Sucesos para un EWM.....	84
Tabla 3.1	Implementación: Nomenclatura en Formas.....	147
Tabla 3.2	Implementación: Nomenclatura Tipos de datos.....	148
Tabla 3.3	Pruebas de Unidad Clase Plantilla	155
Tabla 3.4	Pruebas de Unidad Clase Evaluación	157
Tabla 3.5	Pruebas de Unidad Clase Indicador Clave.....	158
Tabla 3.6	Pruebas de Unidad Clase Calificación	159
Tabla 3.7	Pruebas de Unidad Clase EWM.....	160
Tabla 3.8	Pruebas de Integración	161
Tabla 3.9	Pruebas de Validación	162

RESUMEN

El presente trabajo pretende la construcción del prototipo de un sistema automático que permita brindar soporte a los administradores de un call center de cobranzas. Está conformado por tres capítulos que contienen la información necesaria para que el lector comprenda la teoría de la administración de un call center, y los pasos que se siguieron para la construcción del prototipo, en base a la Metodología orientada a objetos para construcción de Software OMT.

El Capítulo I, contiene los fundamentos teóricos para introducir al lector a los conceptos de Call Center, y describe la Administración del mismo, que es la base del presente trabajo.

El Capítulo II, se refiere a los Requerimientos Funcionales, de interfaces externas y del desempeño del Prototipo. Además, se realiza el Análisis, donde se describe el problema, y se diseña el modelo de Objetos y el respectivo diagrama del modelo de Objetos, el modelo Dinámico formado por diagramas de Seguimiento de sucesos de cada objeto y diagrama de estados y el modelo Funcional, constituido por los diagramas de flujo de datos.

El Capítulo III contiene la implementación del prototipo utilizando Visual Studio .NET y las pruebas realizadas para comprobar el buen desempeño del mismo.

PRESENTACIÓN

El objetivo principal del presente trabajo es presentar el prototipo de un sistema automático para administrar un Call Center de Cobranzas, el cual permita brindar soporte a los administradores para poder supervisar y auditar las gestiones realizadas por los agentes telefónicos. Así como también los resultados de productividad obtenidos por los mismos.

Otro objetivo es introducir al lector en los conceptos de la administración de un Call Center, para una buena comprensión del funcionamiento del prototipo.

Además, el lector puede ver el proceso de análisis, diseño e implementación del prototipo, de acuerdo a la Metodología OMT y el desarrollo de una aplicación web utilizando como herramienta Visual Studio .NET.

1. FUNDAMENTOS TEÓRICOS

1.1. INTRODUCCIÓN A CALL CENTER

1.1.1. DEFINICIÓN DE CRM (Customer Relationship Management)

Es una estrategia de negocios centrada en el cliente, la cual está destinada a lograr identificar y administrar las relaciones en aquellas cuentas más valiosas para una empresa, trabajando diferentemente en cada una de ellas de forma tal de poder mejorar la efectividad sobre los clientes.

Los CRM son las soluciones tecnológicas para conseguir que “la estrategia de negocio” esté centrada en anticipar, conocer y satisfacer las necesidades y los deseos presentes y previsibles de los clientes”.

Los objetivos de las soluciones CRM son:

- Maximizar la información del cliente
- Identificar nuevas oportunidades de negocio
- Mejora del servicio al cliente
- Procesos optimizados y personalizados
- Mejora de ofertas y reducción de costos
- Identificar los clientes potenciales que mayor beneficio generen para la empresa.
- Fidelizar al cliente, aumentando las tasas de retención de clientes

Para implantar una herramienta CRM se deben considerar los cuatro pilares básicos en una empresa:

■ Estrategia:

La implantación de herramientas CRM debe estar alineada con la estrategia corporativa y estar en concordancia con las necesidades tácticas y operativas de la misma.

■ Personas:

La implantación de la tecnología no es suficiente. Al final, los resultados llegarán con el correcto uso que hagan de ella las personas. Se debe gestionar el cambio en la cultura de la organización buscando el total enfoque al cliente por parte de todos sus integrantes.

■ Procesos:

Es necesaria la redefinición de los procesos para optimizar las relaciones con los clientes, consiguiendo procesos más eficientes y eficaces. Al final, cualquier implantación de tecnología redundará en los procesos de negocio, haciéndolos más rentables y flexibles.

■ Tecnología:

La cual en cada caso será diferente en función de las necesidades y recursos de la empresa. {2}

Monica, P. P. (s.f.). *Desarrollo Histórico Del Marketing*. Bogotá: Universidad Libre Colombia.

Conceptualmente, la metodología para desarrollar un proyecto CRM contemplará las siguientes etapas:

- **Definición de objetivos y visión del proyecto CRM**

Es necesario definir una visión ("cómo será la organización tras la implementación del proyecto") así como unos objetivos globales del proyecto para así poder focalizar en estos objetivos y poder hacer un seguimiento de los mismos. La definición ha de ser desarrollada tras un análisis inicial para conocer tanto los puntos fuertes y débiles de la organización, siendo crítico este análisis inicial para el desarrollo posterior del proyecto.

Estos objetivos deben ser bastantes concretos (dentro de unos rangos). Es decir, los objetivos serían por ejemplo "disminuir la tasa de pérdida de clientes en un x%" en lugar de "mejorar las relaciones con los clientes".

- **Definición de la estrategia CRM**

Tras haber definido la visión y objetivos, es indispensable definir una estrategia para alcanzar los objetivos definidos.

En esta estrategia es clave la definición del posicionamiento en cada uno de los segmentos de clientes de la organización, analizando las competencias actuales y necesarias así como un calendario para su implementación.

- **Cambios organizacionales, en los procesos y en las personas**

Es necesario modificar la estructura organizativa y los procesos para conseguir una empresa centrada en el cliente. Los procesos han de ser redefinidos para mejorar su eficacia y eficiencia dando máxima prioridad a los que más impacto tengan en la satisfacción del cliente. En este punto, la tecnología será clave.

Igualmente, es necesario introducir los valores de organización orientada al cliente en la cultura corporativa. Este es uno de los elementos críticos en el éxito de un proyecto CRM: la "pasión por el cliente" integrada en la cultura de la organización.

■ Información

Definición de las correctas prácticas para la gestión de la información.

Esta es una parte muy importante para el desarrollo de la "inteligencia de clientes" (customer intelligence) y consiguiendo de esta manera conocer más a los clientes, paso inicial para el desarrollo de una estrategia completa CRM basada en el conocimiento de los clientes y el desarrollo de productos y servicios a su medida.

Igualmente es importante también la recogida de información para la mejora de los procesos así como para la puesta en marcha de sistemas de mejora continua.

■ Tecnología

En este punto es muy importante destacar que es totalmente necesario conocer exactamente las necesidades de negocio que se tienen para poder escoger la solución tecnológica mejor adaptada a las necesidades concretas.

Es problemático tanto escoger soluciones sobredimensionadas para las necesidades como escoger una solución que tras su implantación se detecta claramente que es insuficiente para las necesidades de la organización.

- Seguimiento y control

Como cualquier proyecto importante, se han de definir unos indicadores (KPI) que sirvan para el control de los resultados así como la toma de decisiones en consecuencia con esos objetivos.

Es importante destacar que aunque tengamos un plan global de desarrollo del proyecto es muy importante dar pasos cortos y seguros, analizando el ROI (Retorno de la inversión) de cada uno de los pasos y así seguir motivando a la organización hacia el camino de ser una "organización centrada en el cliente".

1.1.2. DEFINICIÓN DE CALL CENTER

Un Call Center es el conjunto tecnológico y administrativo que permite unificar el procesamiento de los sistemas informáticos y las facilidades de la conmutación de llamadas telefónicas.

El Call Center es una herramienta que se diseña y construye, "a medida", atendiendo las necesidades que plantean las áreas comerciales. A los requerimientos y necesidades de esas áreas es necesario incorporar criterios de calidad, los que se traducen en requerimientos de equipamiento y recursos humanos, que definen o modelan el Call Center.

1.1.3. CALL CENTER CON ORIENTACIÓN CRM

El Call Center debe apoyar los siguientes objetivos de la estrategia CRM:

- Adquisición

- Retención
- Crecimiento
- Conocimiento

1.1.3.1. Retención de clientes

- Resolver en el primer contacto
 - Del énfasis en la eficiencia al énfasis en la efectividad
 - Requisitos:
 - Disponibilidad de información en línea: facturación, órdenes, información cliente, quejas y reclamos, información producto y políticas, localizador.
 - Conocimiento agente: entrenamiento, enrutamiento por habilidades.
 - Empoderamiento: enfoque solución, procedimientos flexibles, atribuciones.
 - KPI: Líneas de servicio al cliente, líneas técnicas.
- Maximizar el acceso
 - Acceso multicanal (Contact Center); telefónicos (IVR, fax por demanda, Inbound y Outbound) y virtuales (email, chat, call back, VoIP).
 - Cola universal
 - Combinación autoservicio (Internet e IVR) y soporte operadora (agente o web collaboration).
 - Programación de turnos y agentes: inferior a la demanda, superior a la demanda.
 - Horarios y acceso telefónico (combinación de números locales y nacionales)
 - Nuevos requerimientos de capacitación para asesores con capacidad de atención multicanal.

Las estrategias de retención de clientes en el Call Center son: Identificar, clasificar y personalizar al cliente.

1.1.3.2. Adquisición y crecimiento de clientes

■ Pre llamada

- Generar y calificar prospectos
- Convertir las preguntas sobre productos y servicios en potenciales procesos de venta (marcar cliente como potencial).

■ Durante el tiempo de la llamada

- Inbound
 - Identificar y clasificar al cliente según valor y oportunidad
 - Sistema de soporte de decisión para sugerir alternativas de solución o venta
 - Enrutar a agente con habilidades
- Outbound
 - Calidad de base de datos, atractivo oferta, estrategia para realizar contacto.
 - Efectividad

■ Después de la llamada

- Acciones Correctivas
- Auditorías de gestión

1.1.3.3. Conocimiento de clientes

Del cliente en la interacción del Call Center se recopila:

- Inquietudes y Dudas sobre el producto/servicio
- Problemas con el producto/servicio
- Características deseables sobre el producto/servicio

- Comportamiento en el canal
 - Expectativas de trato
 - Retroalimentación a campañas o iniciativas
 - Alertas de deserción
- Tipos de análisis y reportes
- Identificar y clasificar razones para que un cliente decida no cancelar su deuda o no compre el producto.
 - Identificar y clasificar áreas de insatisfacción del cliente con el producto.
 - Identificar y clasificar áreas de insatisfacción del cliente con el Call Center.
 - Identificar y clasificar oportunidades de mejoramiento del producto
 - Identificar y clasificar información no disponible o errada.
- Medios y Herramientas
- Agente: incentivo a ideas aplicadas que mejoren la relación con el cliente.
 - Customer Interaction Software (Especializado o parte de una suite de CRM)
 - Herramientas de análisis: OLAP, Data Mining

1.1.4. ANTECEDENTES Y EVOLUCIÓN DEL CALL CENTER

Los Call Centers surgen a nivel mundial en los años 80 y en Latinoamérica en los años 90.

A continuación un gráfico en los 90's:

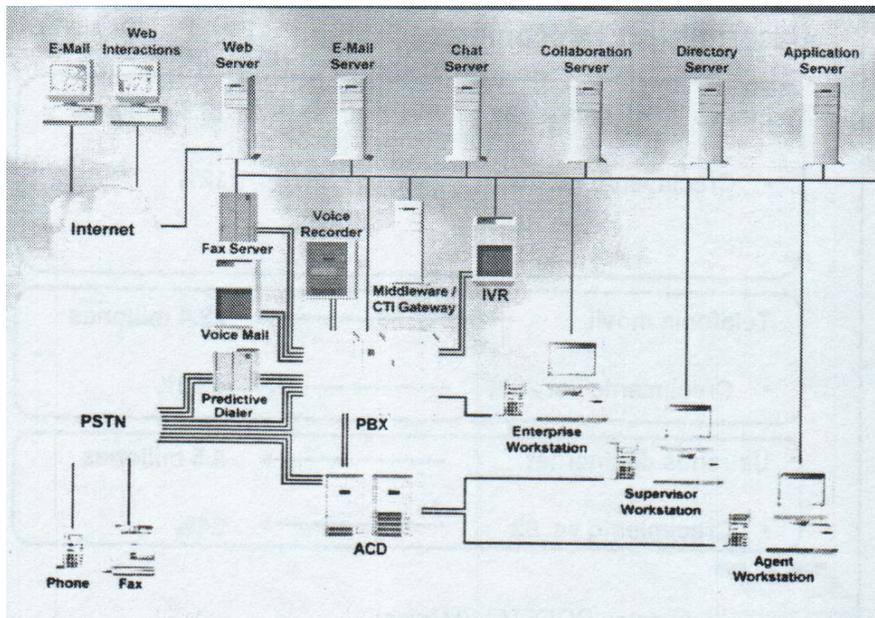


Figura 1.1 Arquitectura de CC en los 90's

Ref.: Desafíos y Oportunidades del Contact Center ante una nueva era

Y el esquema a futuro:

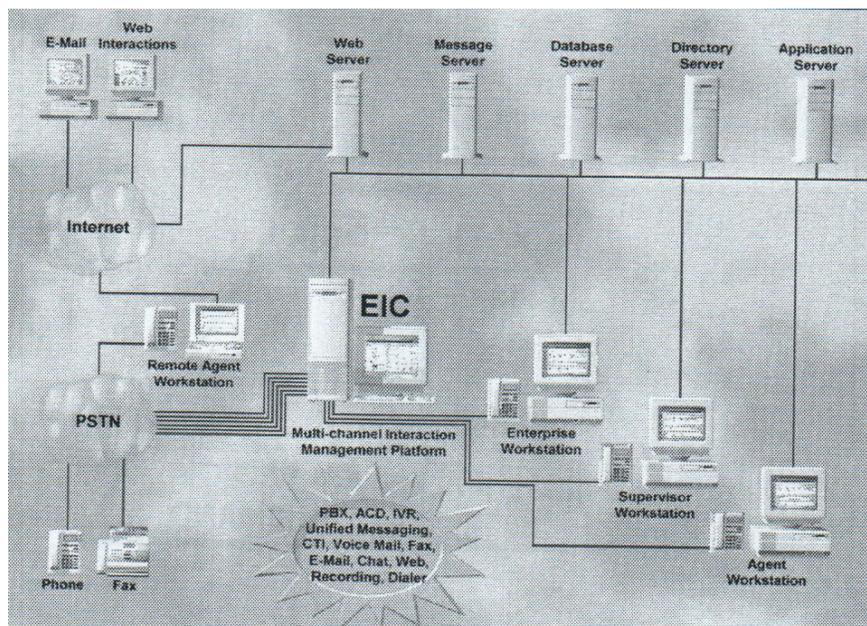


Figura 1.2 Arquitectura de CC a futuro

Ref.: Desafíos y Oportunidades del Contact Center ante una nueva era

La línea de evolución de los Call Centers ha progresado tecnológicamente desde instalaciones primordiales de atención "manual" hasta los llamados agentes universales, pasando por la automatización de respuestas rutinarias, integración de voz y datos, software especializado para aplicaciones de agentes y la marcación predictiva (predictive dialing).

Los sectores en los cuales se usan actualmente Call Centers son:

- Telecomunicaciones
- Servicio Al Cliente
- Aeronáutica
- Tele mercadeo
- Turismo
- Hotelero
- Publicitario
- Financiero
- Salud

1.1.5. OBJETIVOS

Los objetivos principales de un centro de llamadas pueden dividirse en dos grandes grupos:

- Reducir costos
 - Marcación Directa a Interno (DID)
 - Costo del negocio
 - Tiempo de llamada
 - Tiempo de espera
 - Personal
 - Transferencia

- Papel
- Complejidad
- Tiempo de aprendizaje

- Incrementar Ganancias
 - Tiempo de aprendizaje
 - Productividad de agentes
 - Satisfacción de los clientes
 - Incrementar posibilidad de negocios
 - Retención de clientes
 - Funcionalidad
 - Calidad
 - Posición competitiva

1.1.6. COMPONENTES

- **Recurso Humano**

Son todas aquellas personas que intervienen en la operación del Call Center.

 - **Agentes Operadores**

Son aquellas personas que contestan las llamadas en un Centro de Llamadas y también asesorar y atender cualquier inquietud de los usuarios.
 - **Supervisores**

Son las personas encargadas del monitoreo, control y supervisión de los agentes y de la óptima operatividad del Call Center.

- Cliente

Es la persona a la cual o con la cual se realiza la gestión telefónica.

- Tecnología

La adquisición e implantación de la tecnología en un Call Center, juega un papel muy importante, ya que es la base de su funcionamiento.

Este tema se lo tratará ampliamente más adelante.

- Procesos

Para la implantación de un Call Center es necesario evaluar y rediseñar los procesos de la compañía, buscando un mejoramiento integrado, que garantice el cumplimiento de sus objetivos y orientándolos de cara al cliente, teniendo en cuenta:

- Las expectativas y necesidades de los clientes
- Los resultados y sus especificaciones de calidad, enfocándolas a proveer mayor valor al cliente.
- Los proveedores
- El flujo de trabajo, actividades y reglas del negocio, buscando mejorar la productividad.
- Los indicadores de control y desempeño
- Relaciones con los clientes y proveedores internos. Identificación de oportunidades de ventas cruzadas y referencia.

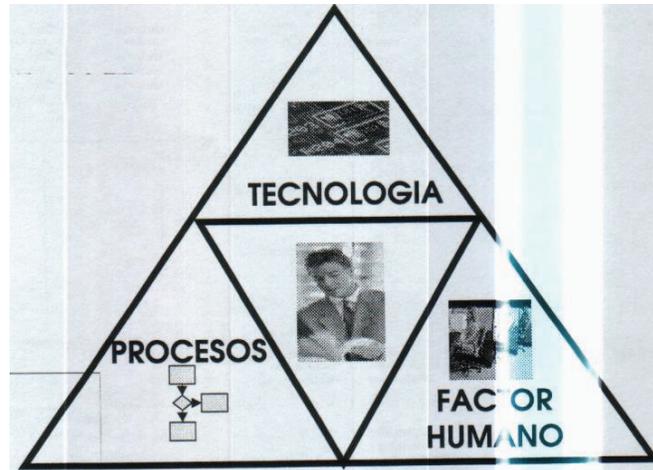


Figura 1.3 Componentes de un CC
Ref.: Factores Clave de desempeño en un CC

1.1.7. TIPOS

Un Call Center puede brindar varios servicios, dependiendo del tipo de llamadas que se realicen, estas pueden ser Inbound, Outbound o ambas.

- Inbound (Llamadas entrantes)
 - Servicio al cliente: información, quejas, reclamos, estudios de mercado, sondeos de opinión, calidad y satisfacción de clientes.
 - Identificación de oportunidades de ventas cruzadas y referencia para nuevos negocios.
 - Toma de pedidos.
 - Oferta de nuevos productos.
 - Realización de ventas telefónicas de servicios.
 - Captura de información de los clientes para ser visitados.
 - Aclaración de dudas sobre la campaña existente.
- Outbound (Llamadas salientes)
 - Verificación de datos (actualización de base de datos).

- Tele mercadeo.
 - Encuestas (investigación del mercado, toma de opiniones).
 - Venta de servicios.
 - Presentación de campañas.
 - Seguimiento a las ventas.
 - Divulgación de información.
 - Oferta de servicios complementarios o nuevos.
 - Recuperación de cartera vencida (gestión de cobranza).
-
- Call Blending (Combinación de llamadas entrantes y salientes)

Durante una campaña puede realizarse una combinación de llamadas entrantes o salientes en el Call Center. En ambos casos se mantienen los niveles de servicio y se maximiza la productividad de los agentes.

1.2. TECNOLOGÍA APLICADA A UN CALL CENTER

Actualmente, el teléfono es el primer punto de contacto y el medio de comunicación preferido entre la empresa y sus clientes. Por tanto, la calidad del servicio ofrecido por teléfono es esencial para iniciar y mantener la relación con los clientes. El concepto de centro de contacto ha evolucionado con el tiempo desde la gestión básica de llamadas telefónicas hasta la información e interacción con el cliente a través de múltiples medios de acceso.

Según su grado de sofisticación, un centro de contacto requiere la integración de recursos de infraestructura, soluciones tecnológicas (aplicaciones dedicadas) y sistemas de marcación.

1.2.1. INFRAESTRUCTURA

1.2.1.1. Puestos de Trabajo

Son los lugares de operación donde se ubican los agentes telefónicos para realizar su trabajo de interacción con los llamantes. Este puesto debe constar de un cubículo, un PC, un head phone y con su respectivo punto de voz y datos. Para esto es necesario establecer el cableado que se utilizará en la organización.

El cableado de red de voz y datos debe soportar más de 100Mbps y los equipos activos también deben soportar estas velocidades, aunque lo recomendable actualmente es un cableado que soporte VOIP.

Esta característica del cableado permitirá ahorrar nodos en el mismo, ya que por una sola vía se tendrá voz y datos; o bien utilizar nodos dobles, teniendo algún respaldo en lugares críticos que así lo ameriten.

Para el cableado que soportará el Call Center, es recomendable utilizar velocidades de Gigabit por segundo en la vía principal, así como nodos “switchados” a 100 Mbps cuando menos.

1.2.1.2. Comunicaciones

■ Enlaces de Datos

Si el Call Center estuviera compuesto de una oficina matriz y sucursales es necesario determinar el tipo de enlace de datos que se utilizará.

- ADSL Línea Digital Asimétrica

Para redes menores a 12 usuarios ADSL.

El ADSL es una tecnología de líneas digitales que sobre las redes telefónicas convencionales, permite alcanzar velocidades de conexión mucho mayores que las actuales hasta 50 veces más que los módems convencionales de 33.6 Kbps.

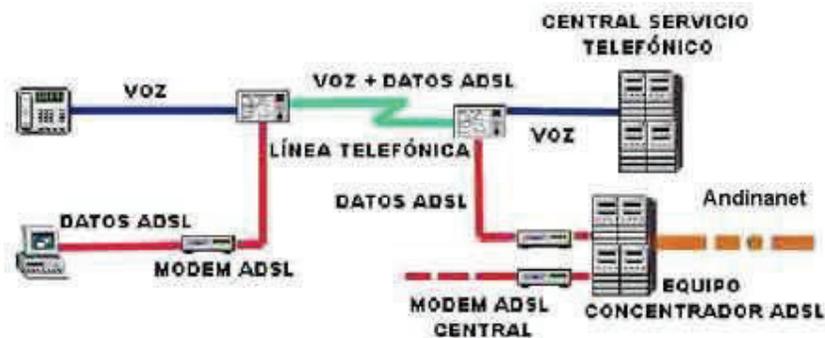


Figura 1.4 Línea Digital Asimétrica ADSL

Ref.: Factores Clave en la toma de decisiones para la inversión de tecnología en CC

En el servicio ADSL, el envío y recepción de datos se establece desde el ordenador del usuario a través de un módem ADSL. Estos datos pasan por un filtro (splitter), que permite la utilización simultánea del servicio telefónico básico (RTC) y del servicio ADSL. Es decir, el usuario puede hablar por teléfono a la vez que está navegando por Internet.

Los canales de datos son asimétricos, es decir, no tienen la misma velocidad de transmisión de datos. El canal de recepción de datos tiene mayor velocidad que el canal de envío de datos.

Esta asimetría, característica de ADSL, permite alcanzar mayores velocidades en el sentido red - usuario, lo cual se adapta perfectamente a los servicios de acceso a información (Ej. Internet) en los que normalmente, el volumen de información recibido es mucho mayor que el enviado.

- Canales Dedicados

Para redes mayores a 12 usuarios un canal dedicado actúa como una conexión permanente. Los canales dedicados se miden por el tamaño de su capacidad y están disponibles en los siguientes anchos de banda: 64K, 128K, 256K, 512K, a mayor ancho de banda mayor capacidad de transmisión.

Los beneficios de un canal dedicado son:

- No existe cargo alguno por llamada telefónica únicamente una cuota de consumo mensual.
- La velocidad de acceso es mayor al de los módems tradicionales

Varios usuarios pueden hacer uso del canal simultáneamente.

- VPN's Redes Privadas Virtuales

Si es que se trabajará con VOIP en la matriz y en las sucursales.

Una red privada virtual es una red de datos que hace uso de la infraestructura pública de telecomunicaciones, manteniendo la privacidad a través del uso de protocolos de túnel y procedimientos de seguridad.

Una VPN puede ser comparada con un sistema de líneas dedicadas usadas por una sola compañía. La idea de una VPN es dar a la empresa las mismas capacidades del sistema anterior a un costo mucho menor utilizando la infraestructura pública compartida en lugar de una privada.

El uso de una VPN involucra la encriptación de los datos antes de enviarlos a través de la red pública y la desencriptación en

el punto de recepción de los mismos. Un nivel de seguridad adicional involucra la encriptación no solo de la información sino de las direcciones de origen y destino de la misma.



Figura 1.5 Red Privada Virtual

Ref.: Factores Clave en la toma de decisiones para la inversión de tecnología en CC

■ Telefonía

- Central Telefónica (PBX, Private Branch Exchange)

Es la Central de conmutación de llamadas telefónicas, la cual puede ser una PC PBX o una central telefónica tradicional.

- Líneas Digitales y análogas

En un Call Center son imprescindibles las líneas telefónicas receptoras o generadoras de llamadas, estas pueden ser análogas o digitales (ej.: E1).

Línea E1 es un servicio totalmente digital brindado sobre la red de fibra óptica. Brinda la posibilidad de configurar y agregar rápidamente otros productos para la empresa. E1 permite conectar una central privada digital a la red telefónica

pública y puede ser configurado para llamadas entrantes, salientes y bidireccionales.

Garantiza total disponibilidad del servicio, aún en condiciones de tráfico pico; brinda privacidad dentro de la red.

- Marcación Directa a Interno (DID): Los internos funcionan como líneas directas.
- Facturación Detallada: Permite reconocer el destino y la duración de las llamadas.
- Análisis de tráfico telefónico: Sirve para determinar la mejor solución a las necesidades de la empresa.

E1 puede utilizarse para sustituir las actuales líneas analógicas expandiendo su capacidad, o reemplazar servicios similares de otros proveedores.

■ **Número Gratis**

Es recomendable que el Call Center disponga de un número gratis, 1800 o similar, con el objetivo de que el cliente pueda contactarse con los agentes telefónicos sin costo alguno, especialmente para gestión de tele mercadeo, autorizaciones, consultas de saldos/ movimientos, etc.

■ **Número Piloto**

El número piloto o número PBX es aquel número telefónico el cual agrupa varias líneas, este es muy utilizado para Call Center que realizan gestión de cobranza.

1.2.1.3. Arquitectura de red

Microsoft recomienda mantener la seguridad a nivel de firewall, en todas las redes, especialmente aquellas que tienen salida hacia el Internet, por esta razón se debe tomar muy en cuenta esta recomendación en el diseño de la red.

■ Arquitectura Física

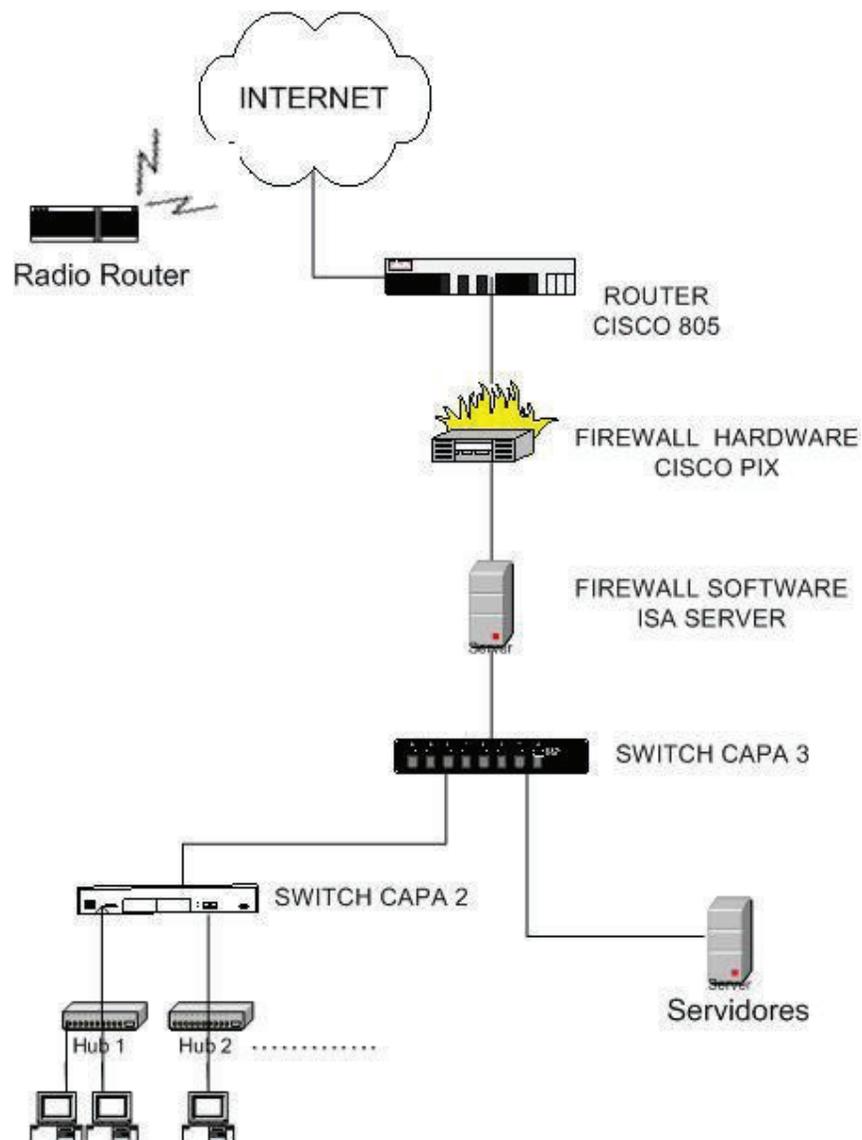


Figura 1.6 Arquitectura Física
Ref.: Autor

Siempre es recomendable tener dos firewalls uno de hardware al que se conecta directamente la red WAN y otro de software donde se conecta la red LAN, ahora si en algún punto la empresa crece al tiempo de tener un Servidor WEB empresarial con un Sitio Web, esta estructura es ideal para colocar dicho equipo entre ambos firewalls.

Para reducir tiempos y reducir las colisiones de la red es factible tomar en cuenta la estructura estrella en la cascada de Hubs o switches capa dos.

■ Arquitectura Lógica

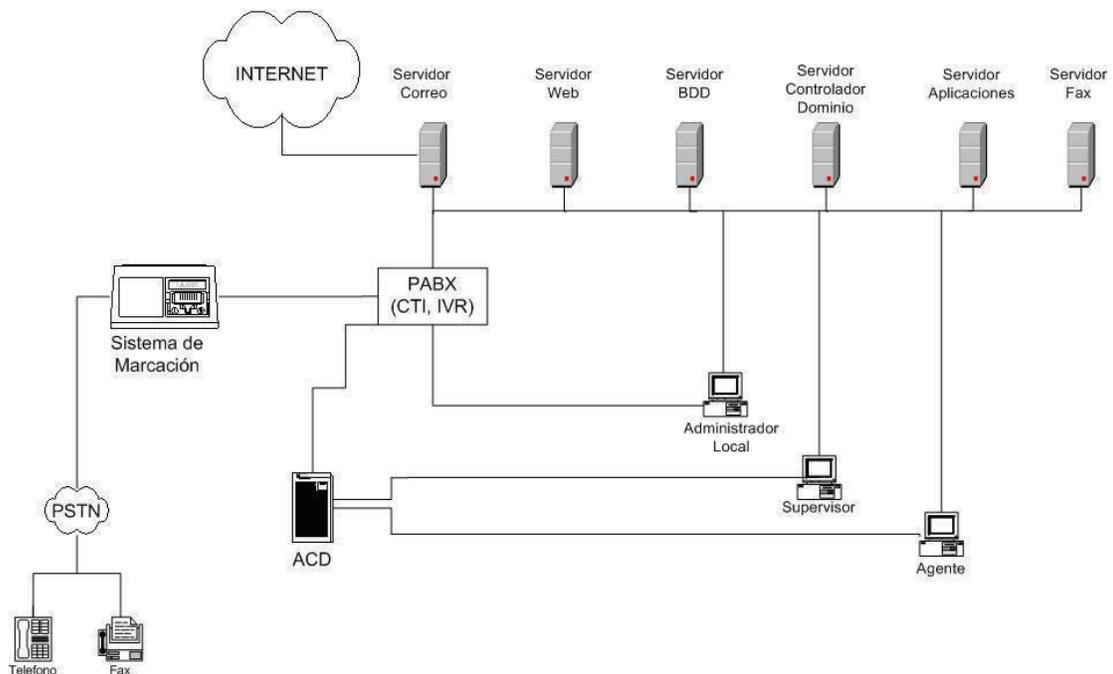


Figura 1.7 Arquitectura Lógica

Ref.: Autor

1.2.2. SOLUCIONES TECNOLÓGICAS

1.2.2.1. Conversión de texto a voz (TTS Text to Speech)

La tecnología de conversión de texto a voz, transforma cualquier texto legible por ordenador en voz sintética con sonido humano. Dado que esta tecnología se basa en software, no se necesita hardware especial.

Las características principales que debe tener una aplicación TTS son:

- Convierte cualquier texto legible por ordenador en salida de voz con sonido natural.
- Permite controlar el volumen, la velocidad y el tono del habla.
- Existen distintas voces (femeninas y masculinas) disponibles.
- Utilizar secuencias de control para personalizar la salida de texto a voz.
- Admitir la entrada fonética con información prosódica para añadir entonación natural a la voz sintetizada.
- El editor del diccionario de excepciones deberá permitir al usuario especificar la pronunciación de abreviaturas y palabras o nombres extranjeros.
- Ofrecer múltiples idiomas

1.2.2.2. Reconocimiento Automático de voz (ASR)

Es una tecnología informática que permite la conversión de información hablada en información textual. El reconocimiento automático de voz puede

ser exclusivamente software o software con integración de un hardware especial.

Actualmente existen 2 tecnologías disponibles para el ASR: Lernout & Hauspie y Nuance.

- **Tecnología para ASR Lernout & Hauspie**

Para estos sistemas de reconocimiento de voz, la base del proceso de reconocimiento se define en el contexto. Un contexto describe qué frases acepta el sistema. En una determinada situación, los contextos dependen por tanto de la aplicación. Un contexto está ligado a un idioma.

- **Tecnología para ASR Nuance (NLU Lenguaje Natural)**

Permite al llamante interactuar con un sistema de procesamiento de llamada con la forma más natural de comunicación: las frases habladas naturalmente.

En lugar de tener que trabajar con capas de menús e instrucciones, como ocurre con los sistemas estándar, una aplicación NLU reconoce varias unidades de información hablada en una frase. De este modo, la NLU permite a los usuarios hablar con la misma naturalidad que si estuvieran dirigiéndose a una persona.

1.2.2.3. Fax a Demanda

Es una herramienta que ofrece la posibilidad de crear y enviar faxes a petición del llamante. La función de fax a petición permite a cualquier usuario con

teléfono de tonos(o a través de ASR) y un aparato de fax, solicitar documentos impresos y otros tipos de información.

1.2.2.4. Respuesta Interactiva de voz (IVR - Interactive Voice Response)

El IVR consiste en el uso del teléfono para establecer comunicación bidireccional entre el llamante y un sistema de procesamiento de datos.

La entrada es hablada o se realiza a través del teclado del teléfono o el fax. La salida es hablada o se envía al llamante a través del fax o el correo electrónico.

Mediante esta tecnología es posible realizar una interacción guiada con el usuario que llama a solicitar una información o servicio. En los casos de información es posible proporcionar completamente una experiencia satisfactoria de servicio sin necesidad de intervención humana.

En otros casos, el proceso guiado le permite al usuario acceder al grupo de agentes indicado para resolver su situación o necesidad.

El IVR es una solución ideal para servicios automáticos que funcionen durante las 24 horas.

Una de las ventajas principales de este sistema consiste en el establecimiento de nuevos modos de interacción con usuarios finales mediante el uso de las tecnologías de voz y conversión texto a voz.

Por su naturaleza, los sistemas de IVR deben consumir/vocalizar información corporativa y realizar transacciones en tiempo real, interactuando dentro de la organización con múltiples plataformas.

Una aplicación de IVR debe contar con las funciones básicas siguientes:

- Contestación, colocación y transferencia de llamadas telefónicas.
- Interpretación de comandos de marcación por tonos.

- Interpretación de voz natural hablada.
- Ensamblaje y reproducción de frases grabadas.
- Grabación de mensajes de voz.
- Recepción, reenvío, respuesta o creación de mensajes desde los paquetes de correo electrónico más usados.
- Acceso local o remoto a bases de datos para consultar, recuperar o actualizar información.
- Envío de fax en la misma llamada o puesta en cola de fax en la base de datos de faxes salientes para su posterior envío (puede realizarse en una segunda llamada en canales definibles).
- Recepción de fax y escritura de toda la información en la base de datos de faxes entrantes.
- Transferencia y devolución de llamadas a otras aplicaciones secundarias.
- Integración de funcionalidad adicional a través de interfaz estándar.
- Conmutación de dos llamadas (conmutación MUIP).
- Flexibilidad y adaptación a los permanentes cambios en las reglas de negocio.
- Por su diseño las aplicaciones IVR no deben estar limitadas en términos de tamaño (número total de opciones de menú), niveles de profundidad de los menús o complejidad en lo referido a acceso a datos.

Las Aplicaciones IVR clásicas son las siguientes:

- Banca Telefónica
- Realización de reservas por teléfono
- Inscripción en cursos por teléfono
- Sistemas de distribución de información
- Sistemas de recogida de información
- Realización automática de pedidos/compra telefónica por catálogo

- Confirmación de transacciones
- Pago de servicios

1.2.2.5. Distribución Automatizada de llamadas (ACD - Automatic Call Distributor)

Esta aplicación permite asignar, de manera totalmente automática, las llamadas que llegan al grupo de agentes capacitados para su atención. El ACD determina la disponibilidad de los agentes para la atención de la llamada y la enruta para su atención. En caso de no haber agentes disponibles, la llamada puede ser colocada en cola, escuchando mensajes institucionales que sugieren la espera hasta que haya algún agente disponible. Tan pronto se produce este evento, automáticamente la llamada es enrutada al agente que ha quedado disponible.

La capacidad de enrutar una llamada al agente más adecuado es un factor importante en la administración de un Call Center. El ACD es el responsable de la distribución precisa y eficiente de las llamadas de los agentes disponibles de acuerdo con la capacidad del agente y el coste relativo.

La distribución de llamadas se basa en el concepto de pilotos. Un piloto corresponde a un punto de entrada en un sistema ACD. Suele haber un piloto para cada tipo de servicio que presta una empresa a sus clientes.

El ACD asocia 3 clases de objetivos distintos: pilotos, colas y recursos (agentes o grupos) por medio de direcciones de prioridad.

Si los tiempos de espera se incrementan en exceso, es posible enrutar todas las llamadas o algunas de ellas a uno o varios emplazamientos remotos (desbordamiento de red).

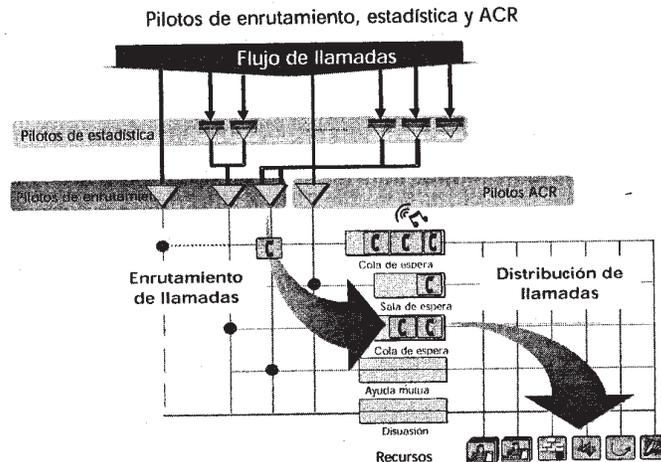


Figura 1.8 Distribución Automatizada de llamadas
Ref.: Soluciones Web-Telephony al alcance de CC

Las ventajas de tener un ACD son:

- Mejora en la calidad del servicio mediante una atención oportuna de la llamada.
- Monitoreo permanente de la productividad de los agentes asignados para la atención en la línea.

1.2.2.6. Integración de Telefonía Computador (CTI – Computer Telephony Integration)

Es un middleware que hace las funciones de "director de orquesta" de todos los componentes hardware y software del Call Center. Es este servidor el que, por ejemplo, define y adscribe a los agentes telefónicos al correspondiente ACD, o imparte órdenes para el envío de información a los diferentes puestos de los agentes, o almacena y estructura la información para los diferentes reportes de operación que se requiera.

Esta tecnología permite proporcionar una atención personalizada al cliente que llama. Cuando se trata de clientes conocidos, es presentarle al agente toda la

información de su relación con el Call Center, en el momento de entregarle la llamada obteniéndola de una base de datos.

La misma situación puede ser conveniente para llamadas generadas desde el Call Center dándole al agente la posibilidad de revisar la información de la persona a la cual está llamando antes de establecer el contacto para decidir el curso de acción a seguir.

Las ventajas de tener un CTI son:

- Mayor eficiencia en la atención.
- Mejor relación con el cliente.
- Mayor exactitud en la información que se obtiene o se proporciona.

1.2.2.7. Mailing Telefónico Automático (MTA)

Es la herramienta de comunicación automática masiva y selectiva. Consiste en la emisión de mensajes de voz digitalizada a listados de teléfonos, con registro completo de resultados de llamadas.

Se usa para complementar las tareas del Call Center, especialmente cuando se requiere enviar miles de mensajes en que la repetitividad, las estresantes tareas telefónicas y el difícil control, producen caída del rendimiento, mala atención y resultados deficientes.

El mailing telefónico puede ser llevado a cabo no sólo por grandes sistemas IVR sino también por prácticos y sencillos sistemas mono línea – monousuario, verdaderos robots telefónicos fácilmente programables, cuyas “manos” toman virtualmente el micro teléfono, sus “ojos” leen la base de datos, sus “dedos” marcan números telefónicos, sus “oídos” identifican tono de línea, tono de ocupado y tono de llamada, y con su “voz” hablan con los receptores de los llamados. Todo ello regido permanentemente por un “cerebro “ que administra

los tiempos y recoge resultados (derivados del análisis automático del Call Progress) que son almacenados en su “memoria”.

La posibilidad de disponer de MTA en los puestos de agentes del Call Center permite convertir en un instante a cada PC en un medio de difusión telefónica automática y a su conjunto en una poderosa Central de Emisión multilínea de mensajes telefónicos automáticos.

Con el MTA el Call Center contará con un medio de comunicación especialmente indicado para gestiones de:

- Tele- cobranza
- Tele- mensajes
- Tele- promociones
- Tele- marketing

Emitiendo grabaciones que combinen voces, sonidos y efectos especiales.

Es conveniente tener un MTA en los siguientes casos:

- Cuando sea necesario enviar miles de comunicados con bajo costo y alta velocidad.
- Cuando la repetitividad de las tareas telefónicas conduce a una baja calidad de gestión. Para no convertir a una persona en un robot, es preferible utilizar MTA, dejando a personas las tareas de personas y a sistemas automáticos el trabajo de máquinas.
- Cuando se desea implementar servicios de promoción telefónica que exigen herramientas empleadas con creatividad y profesionalismo.
- Cuando es necesario llamar la atención sobre situaciones irregulares. La atención que el interlocutor presta al llamado telefónico es mayor que el efecto de otros medios, quien no

dudará en transmitirlo a quien corresponda, si él no fuera el destinatario final del mismo.

- Cuando avisos televisivos, radiales y gráficos que pese a su carácter masivo no alcanzan a veces a sus destinatarios específicos, pueden ser reemplazados por mensajes telefónicos precisos y con un costo que no resiste comparación, a lo que se agrega la ventaja adicional de poder orientar geográficamente las emisiones según los prefijos urbanos, determinados por barrios o zonas.
- Cuando el correo postal resulta costoso para el tipo de mensaje que se necesita enviar.
- Cuando se necesita estrechar vinculación con clientes, socios o abonados, manteniéndolos informados y demostrando preocupación por contarlos en su cartera.

1.2.2.8. Grabación de llamadas

Existen contactos telefónicos que implican el establecimiento de compromisos recíprocos entre el agente y la persona que lo contacta. Para aquellos clientes del Call Center que por razón de su actividad requieren esta capacidad es necesario realizar la grabación de llamadas cursadas y su almacenamiento y custodia para cuando la situación lo amerite. Esta facilidad tecnológica protege al Cliente del Call Center de reclamos posteriores permitiéndole determinar exactamente los componentes adquiridos y la aceptación de los mismos.

Esta tecnología consiste en una grabación digitalizada de las conversaciones del agente en comunicaciones salientes y entrantes, en discos duros de PCS y directorios a elección, según diversos criterios.

Es imprescindible para control de personal, supervisión de calidad de atención, rendimiento y resguardo de información. Es también un recurso de capacitación y perfeccionamiento que permite detectar defectos y errores, manteniendo registros de desempeño en archivos digitales de audio.

Algunos beneficios de esta tecnología es poder escuchar a los agentes al dirigirse a los clientes: conocer su tono de voz, sus errores, aciertos, entre otras variables de interés. De esta manera se conoce dónde se debe reforzar mediante capacitación, dónde a través de correctivos y dónde amerite despido. También se puede conocer si los guiones están bien diseñados, qué sensaciones producen al ser escuchados por los clientes, es decir si se modificarán y están cumpliendo su cometido.

1.2.2.9. Middleware

La tecnología de Middleware M.O.M (Message Oriented Middleware), interactúa en todo momento con programas, actores y recursos del sistema (teléfonos IP, puertos H. 323, servicios de correo, servidor de correo, servidor de fax, etc.), suscribiendo objetos en línea.

Por estar orientado a Mensajería (intercambio de mensajes entre aplicaciones), habilita la comunicación Asíncrona con entrega de mensajes garantizada en plataformas heterogéneas y distribuidas.

Middleware es el modo más confiable y profesional para soportar aplicaciones de Call Center, CRM. Facilita el monitoreo en tiempo real y genera registros de cada evento para posteriores reportes y explotación de información analítica orientada a la toma de decisiones.

Esta tecnología brinda una propuesta amplia para integrar la aplicación a bases de datos relacionales, aplicaciones "Legacy" y procesos de negocio existentes, manteniendo una vista simplificada desde el Call Center hacia la empresa.

Mantener el Centro de Interacciones "conectado" al BACKBONE de INFORMACIÓN CORPORATIVA, no representa solamente una concepción tecnológica de la solución, habitualmente, las bases de datos, procesos e

inteligencia del negocio NO están dentro de la infraestructura de la empresa pero si son requeridas para brindar adecuados niveles de servicio.

Los costos derivados de no integrar los canales de comunicación entre la compañía y clientes (IVR, Telefonía, Colaboración en la Web), a los sistemas de información son incalculables, los clientes advierten rápidamente el hecho de no contar en el Call Center con información sincronizada y es natural que intenten eludir estos canales para contactar con las áreas que puedan procesar y resolver transacciones, incidentes de soporte o simplemente responder consultas.

1.2.3. SISTEMAS DE MARCACIÓN

1.2.3.1. Marcación automática

Es como un verdadero robot telefónico, que disca un número telefónico indicado, evitando al agente este trabajo.

Un servidor de administración de listas distribuye registros de llamadas a la aplicación del agente a petición de este. Utilizando una interfaz el agente obtiene una vista previa de la información del cliente a ser contactado e inicia la llamada saliente. Una vez finalizada o durante la llamada, el agente puede actualizar la información de la gestión o actualización de datos del cliente.

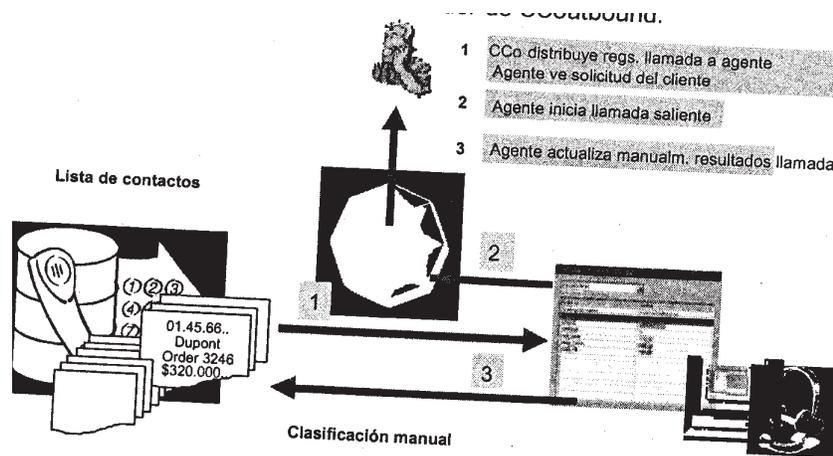


Figura 1.9 Marcación Automática
Ref.: Soluciones Web-Telephony al alcance de CC

1.2.3.2. Marcación progresiva

Cuando los contactos se establecen por medio de llamadas originadas en el Call Center es un factor muy importante y crítico el tiempo asociado con la marcación telefónica. Para disminuir estos tiempos, evitar errores y detectar situaciones de contacto no efectivo:

- No contesta
- Ocupado
- Atendieron
- Contestador Automático
- Fax
- Teléfono dañado

Esta utilización, que representa un aumento de la eficiencia en el proceso de establecimiento del contacto, se traduce en beneficios tangibles para el Cliente del Call Center al aumentarse el número de contactos efectivos por unidad de tiempo y por lo tanto, dando la posibilidad de aumentar la eficiencia de los agentes, factores que se traducen en un mejor retorno de la inversión ROI.

El modo progresivo garantiza que un agente estará disponible para todas las llamadas de clientes activos. El servidor efectúa activamente las llamadas salientes, activa la detección de progresión de todas las llamadas y transfiere las llamadas conectadas a agentes disponibles.

Este modo de marcación no realiza reserva de agentes.

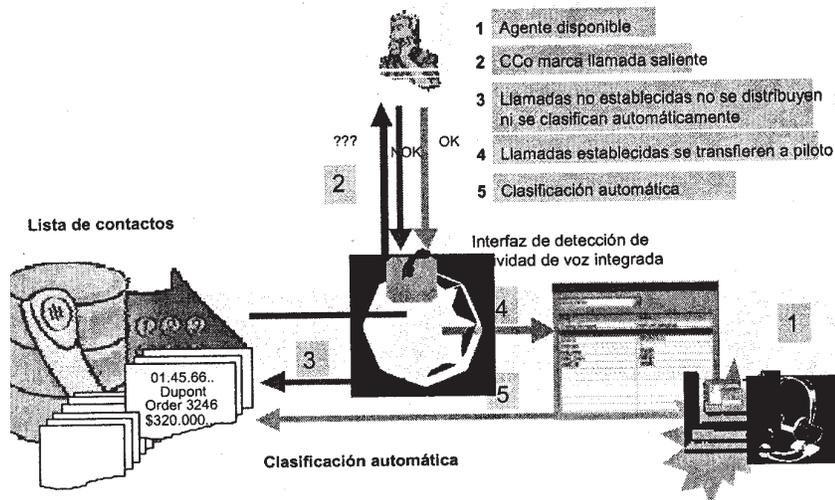


Figura 1.10 Marcación Progresiva
Ref.: Soluciones Web-Telephony al alcance de CC

1.2.3.3. Marcación predictiva

En este modo se activa un algoritmo predictivo. El servidor supervisa la actividad de los agentes, recoge estadísticas y predice el tráfico futuro de llamadas. En este modo de marcación, el servidor puede efectuar más llamadas que agentes disponibles haya en el grupo. En este caso pueden abandonarse algunas llamadas establecidas. El algoritmo se basa en dos parámetros de optimización:

- **Tasa de sobrellamada**

El número de llamadas realizadas se calcula a partir del porcentaje de llamadas establecidas que no se han transferido a un agente libre.

■ Factor de ocupado

El número de llamadas realizadas depende del porcentaje especificado para el tiempo de ocupado del agente.

El marcador predictivo apoya a los CC en la realización de las llamadas salientes para aquellas empresas que tengan un alto número de llamadas por realizar. Si bien este tipo de soluciones no son del todo económicas a comparación de las llamadas programadas y realizadas en forma manual, sí ofrecen un alto beneficio en la eficiencia de la operación, ya que optimizan el tiempo de los agentes en el intento de realizar llamadas exitosas.

Por lo tanto genera acciones consecuentes, sin intervención del agente. Con esta aptitud consigue varios beneficios:

- Reducción de Tiempos.
- Eliminación de acciones improductivas a cargo del agente.

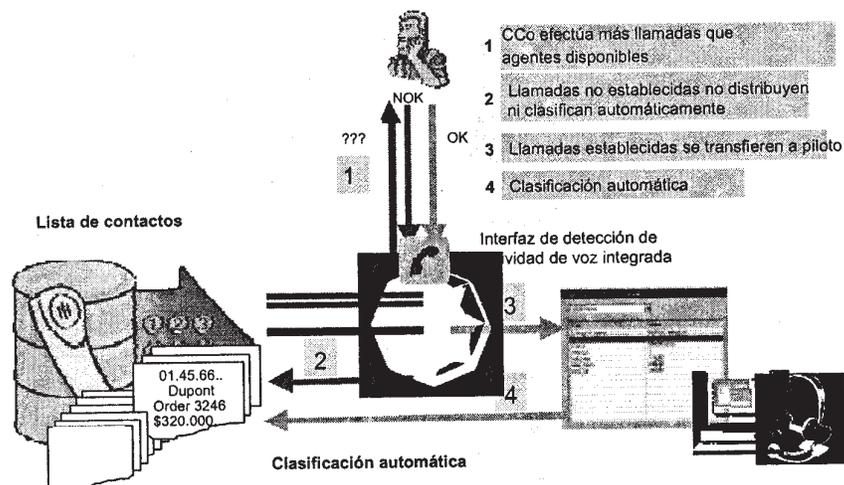


Figura 1.11 Marcación Predictiva

Ref.: Soluciones Web-Telephony al alcance de CC

1.3. ADMINISTRACIÓN EFECTIVA DE UN CALL CENTER

Un Call Center depende de muchos factores que deben funcionar perfectamente en conjunto para tener un rendimiento óptimo. No sólo la tecnología debe cubrir perfectamente sus necesidades, si no que todos los demás elementos del CC deben trabajar para alcanzar la misma meta y cumplir con los objetivos de la organización.

Ante la premisa de que los agentes representan el recurso más importante en los Call Centers, es en este concepto donde se deben canalizar buena parte de los esfuerzos por mejorar los niveles de desempeño, utilizando la auditoría de gestión.

1.3.1. SUPERVISIÓN DE GESTIÓN

Un programa de supervisión de gestión proporciona a los directivos de los Call Center, un sistema estructurado, analítico y permanente de monitoreo y de gestión de resultados. Permitiendo el desarrollo de programas de comparación, teniendo como punto de referencia las mejores prácticas.

Se debe implantar un programa de auditoría de gestión para:

- Contar con indicadores claves de medición.
- Poder evaluar la gestión de los agentes
- Poder comparar
- Todos hablar el mismo idioma
- Tener puntos de referencia claros para mejorar.

Los programas de auditoría de gestión de un Call Center deben medir:

- Costos
- Métricas de ejecución
- Satisfacción de clientes

- Estrategias
- Recurso Humano
- Flujo de procesos de llamadas
- Llamadas y conocimientos de los agentes
- Tecnología
- Hábitat

Las características de un programa de auditoría de gestión son:

- Identificar y documentar el proceso
- Recolectar y almacenar los Indicadores Clave
- Analizar los datos
- Proyectar los resultados de la gestión
- Comunicar los resultados
- Desarrollar un plan de acción
- Implementar y monitorear los resultados

1.3.1.1. Flujo de una llamada

El flujo de una llamada consiste en el recorrido que realiza la llamada desde su origen hasta su finalización. Para conocer este flujo es necesario la utilización de dos conceptos: resultado de gestión y árbol de resultados de gestión.

- Resultado de gestión

Es la última postura del cliente derivada de un evento (llamada telefónica) o en su defecto, el resultado del intento de esta.

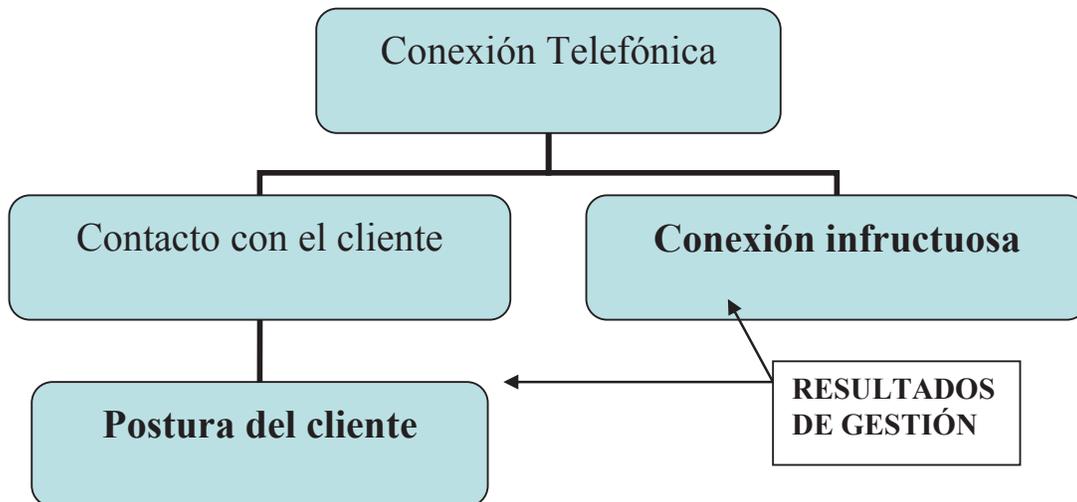


Figura 1.12 Flujo del resultado de una Gestión
Ref.: Autor

Los resultados de gestión son los cimientos del sistema de administración de un Call Center. Cada uno de los resultados que se definen, tendrán el propósito de proporcionar una visión clara y real de lo que está sucediendo y fijar elementos de seguimiento a la estrategia definida.

Se debe garantizar un completo sistema en base a los resultados de gestión que se definan para cada una de las campañas.

Los resultados de gestión se necesitan para:

- Alinear las estrategias
- Tomar decisiones
- Adquirir conocimientos sobre el cliente

Para la elección de los resultados de gestión se debe considerar:

- Un resultado nunca debe suponer algo.
- Los resultados de gestión deben ser apenas suficientes.
- Un resultado de gestión debe tener un objetivo de interpretación.
- Debe ser sencillo de interpretar para el agente telefónico.

- No debe existir el resultado “otros” cuando su proporción sea mayor al 3%.
- En su conjunto deben soportar la óptica de la estrategia.

■ **Árbol de Resultados de gestión**

Consiste en un árbol de decisión, que el agente elige de acuerdo con los sucesos que se den durante la llamada (gestión) que realiza con el cliente. Este árbol debe estar de acuerdo con el negocio de la campaña, de esta forma se deberán diseñar árboles para ventas, cobranzas, atención al cliente, etc. El diseño de este árbol debe contemplar todas las posibilidades factibles de resultados.

El gráfico a continuación muestra el árbol de resultados de gestión que se sugiere para una gestión de cobranza:

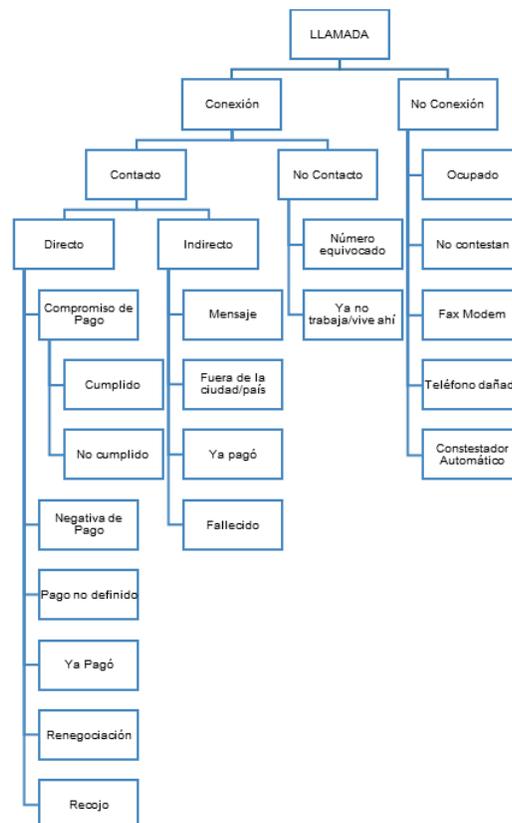


Figura 1.13 **Árbol de resultado de Gestión de Cobranza**
Ref.: Autor

1.3.1.2. Indicadores Clave

Los indicadores clave representan los puntos clave a observar del desempeño de la operación del Call Center, ligados a la estrategia del negocio.

Los indicadores clave sirven para tomar decisiones y para ello es necesario relacionarlos.

Existen 4 indicadores claves, estos son:

- Hit Rate

Representa la relación de éxito entre una conexión y la ubicación de un cliente. Este indicador da la información sobre la calidad de base de datos con la que se está trabajando.

Métrica:

Contactos / Conexiones

- Contact Rate

Indica el porcentaje de contactos que se tiene directamente con el cliente. Con este indicador se puede medir con claridad, la calidad de la estrategia operativa, es decir, se puede ver la eficiencia de la aplicación de best time to call, best day to call y regionalización.

Métrica:

Contactos Directos / Contactos

Este indicador permite evaluar la eficiencia de la gestión telefónica, pero no como estrategia.

- Promise Rate

Identifica el porcentaje de éxito de la gestión telefónica. El agente tiene mediante una buena estrategia operativa a la persona objetivo del otro lado de la línea. Su labor ahora es negociar con una alta calidad el compromiso de pago u otras resoluciones favorables al negocio.

Métrica:

Compromisos de Pago / Contactos Directos

- Efficiency

Representa el cierre efectivo del proceso de gestión. El objetivo final del Call Center es concretar sus gestiones con resultados positivos para el negocio.

La cuantificación del cumplimiento de nuestras promesas de pago o negociaciones acordadas, contra el cumplimiento real indica la efectividad del Call Center.

Métrica:

Compromisos de Pago Cumplidos / Compromisos de Pago Totales

1.3.1.3. Early Warning Monitor (EWM)

Es el medio que permitirá evaluar en línea el desempeño de las estrategias establecidas a través de las variables operativas que impactan directamente en los resultados de operación y el proceso de misión crítica.

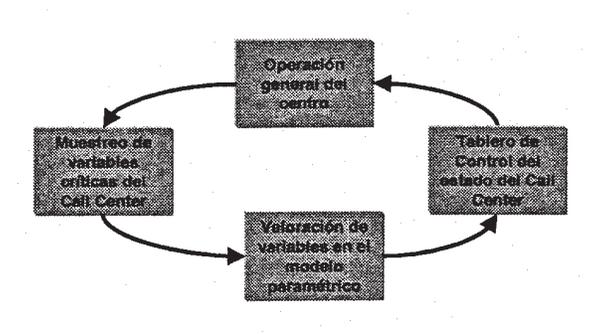


Figura 1.14 Early Warning Monitor
Ref.: Mediciones y Métricas de un CC

El objetivo es mostrar de manera continua el desempeño de la operación a través de información clave y en línea, que permita tomar decisiones eficientemente y detectar posibles desviaciones lo antes posible.

Permitirá implementar acciones preventivas y de mejora, de manera inmediata para encaminar esfuerzos de forma controlada.

Los pasos a seguir para lograr un EWM eficiente son:

- Selección de las variables operativas a monitorear

Se debe tener cuidado de seleccionar única y exclusivamente variables que tengan un impacto significativo en la operación. Estas variables deberán tener una relación directa de interpretación en relación a las estrategias de negocio y operativas definidas y estar orientadas al logro de los objetivos del negocio.

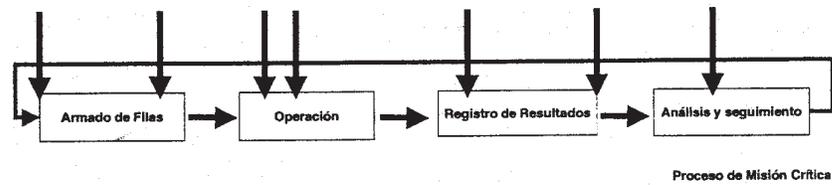


Figura 1.15 Selección de variables para monitorear Early Warning Monitor
Ref.: Mediciones y Métricas de un CC

- Definición de los datos necesarios que brinden información de las variables operativas

Se debe considerar todas las relaciones de datos que pueda tener una variable operativa a monitorear. Así mismo, se debe tener perfectamente ubicado el dato y las características de su disponibilidad. Por ejemplo el Hit Rate.

- Análisis de la relación entre los datos y las variables operativas.

Los datos seleccionados también pueden tener relación con otros, que al combinarlos puedan enriquecer la variable operativas. Por ejemplo Hit Rate del 1 al 30 de Abril.

- Estructuración de la base de conocimiento.
 - Obtención de un esquema de datos históricos, que permitirán establecer parámetros para una medición correcta del EWM.
 - Segmentación de datos para efectos de análisis por año, mes, día e intervalos en el día.
 - Definición de criterios de medición, así como las técnicas y estadísticas utilizadas.

- Definición del proceso de alimentación de la base de conocimiento
 - Qué dato
 - Cuál es su origen
 - Con qué frecuencia
 - En qué formato
 - Con qué otros datos se relaciona
 - Quién lo va a registrar
 - En dónde lo va a registrar
 - Cómo lo va a registrar
 - Cómo se va a validar que sea correcto

- Definición de la forma y medios del despliegue de la información.

La información del EWM deberá estar en un lugar visible en todo momento. No es recomendable que se maneje como un archivo en una computadora, por que dependería de que este archivo fuera leído o desplegado. Siempre deberá tener la última fecha y hora de actualización. Por ejemplo displays, localizadores, tableros, etc.

- Definir acciones derivadas de la información del EWM.

Para la información de cada variable que se definió para ser monitoreada, deberá existir un plan de acción.

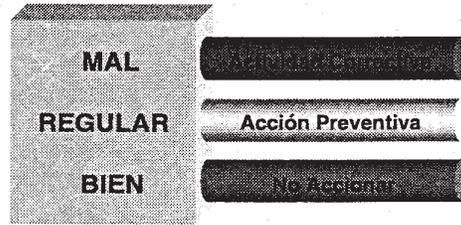


Figura 1.16 Acciones Derivadas Early Warning Monitor
Ref.: Mediciones y Métricas de un CC

El EWM es un instrumento de control que formará una estructura histórica de información respecto a la operación del Call Center. Lo anterior permitirá realizar un análisis de los factores críticos de la operación para dotarnos de las herramientas necesarias para la mejor toma de decisiones.

The screenshot shows the 'Early Warning Monitor' interface. At the top, there are filters for 'Mes' (12/04/02), 'Procesos de Negocio', 'Métrica', and 'Límite de Alerta' (15,000). Below the filters is a table with columns for 'Métrica', 'Valor Actual', 'Valor Objetivo', 'Diferencia', 'Tendencia', 'Alerta', 'Estatus', and 'Acción'. A box labeled 'Calificación' is overlaid on the table, pointing to the 'Estatus' column. A large oval highlights the 'Acción' column.

Métrica	Valor Actual	Valor Objetivo	Diferencia	Tendencia	Alerta	Estatus	Acción	
Cuentas a Gestionar	92,000	95,500	-3,500	5,000	2,500	Acceptable	✓	
Valores de Cuentas	19,500	20,000	-500	1,000	500	Controlado	✓	
Cuentas a Decir	14,500	15,000	-500	1,000	500	Controlado	✓	
Ejecutivos Conectados					2	Controlado	✓	
Tiempo en Línea					11:28:10	No Aceptable	✗	
Llamadas Hechas					4,137	Controlado	✓	
Conexiones					803	Acceptable	✓	
No Conexiones	11,500	12,472	-972	2,072	6,649	3,525	Controlado	✓
Contactos	4,300	12,465	-4,676	-376	1,580	791	Acceptable	✓
No Contactos	950	2,624	-595	305	289	141	No Aceptable	✗
Contactos Directos	1,382	3,642	-1,336	-47	773	367	Controlado	✓
Promesas de Pago	1,128	2,580	-972	216	326	163	Controlado	✓
Llamadas Abandonadas	150	1,580	-232	-102	174	87	Controlado	✓
% Abandona	2.80%		5.24%	-2.35%	5.18%	2.95%	Controlado	✓
HR Rate	82.69%		87.30%	-4.50%	5.88%	2.97%	No Aceptable	✗
Inproductivo Esperes	17.31%		12.80%	4.50%	5.88%	2.97%	No Aceptable	✗
Contact Rate	26.60%		23.65%	2.95%	5.07%	2.91%	Controlado	✓
Promesa Rate	88.79%		70.27%	18.52%	7.22%	3.57%	Controlado	✓

Figura 1.17 Interfaz de EWM
Ref.: Mediciones y Métricas de un CC

1.3.2. AUDITORIA DE GESTIÓN

1.3.2.1. Monitoreo

Una vez identificados los Indicadores Claves y alineada la estrategia, se debe orientar los esfuerzos a la calidad de la gestión telefónica.

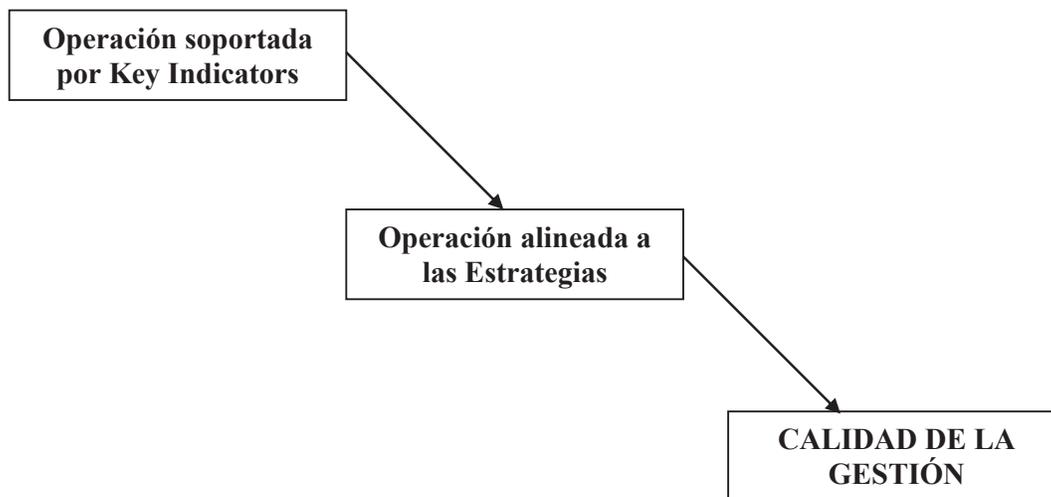


Figura 1.18 Calidad de Gestión (Monitoreo)
Ref.: Autor

Hoy en día ya no es suficiente llevar un control o registro de llamadas y realizar un proceso de supervisión de los agentes que se encuentran ante el cliente.

Ahora la calidad en el proceso de atención, debe incrementarse día con día para el beneficio del Call Center y el de los clientes.

Es por esta situación, que cualquier Call Center debe contar con una medición que identifique las fortalezas y áreas de oportunidad de cada uno de los agentes que representan al CC.

El monitoreo debe identificarse como un área funcional de retroalimentación al agente y apoyo a las áreas operativas, con una visión de mejora continua.

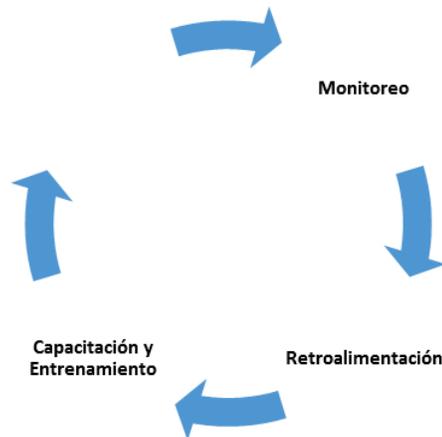


Figura 1.19 Monitoreo
Ref.: Estándar ELTCO

El objetivo del monitoreo es cubrir completamente los siguientes puntos:

- Evaluar continuamente a los agentes
- Identificar problemas con los agentes
- Identificar puntos a mejorar
- Retroalimentar a los agentes
- Retroalimentar a los supervisores
- Medir la calidad de la gestión
- Asegurar el apego a los lineamientos de la estrategia
- Retroalimentar a la gerencia sobre aspectos del mercado

■ **Parámetros a monitorear**

La elección de parámetros se realizará de acuerdo a las áreas de oportunidad y/o perspectivas de nuestro Call Center en cuanto a la atención que brinda.

Los parámetros deberán ser seleccionados minuciosamente para obtener lo que se necesita y hacia dónde se quiere ir con la valoración de los ejecutivos del Call Center.

Los parámetros deberán ser considerados dentro de un check list que será la base de evaluación de cada uno de los ejecutivos.

Es importante tomar en cuenta que debemos tener diferentes check list según sea el tipo de llamada que el agente esté realizando.

Lo anterior vuelve a referenciar al árbol de flujo de llamada, ya que se debe considerar diferentes puntos de medición, según el número de posibilidades que se tiene en el árbol.

- Definición de pesos específicos de los parámetros

Definir los pesos específicos que se necesita para la valoración de los agentes, depende de los resultados que se desea obtener y hacia dónde se quiere ir.

Los pesos específicos dan el poder de direccionar la gestión y empujar al agente a que logre el resultado esperado.

- Tipos de monitoreo

- Remoto

- Es cuando el monitoreo se lo hace utilizando intervención de llamadas, grabaciones, es decir sin estar junto al agente.

- Paralelo

- Es cuando el monitoreo se realiza uno a uno, es decir sentado junto al agente.

1.3.2.2. Balance Score Card

La evolución de los Call Centers ha llevado a diseñar diferentes instrumentos de medición de la operación.

Un instrumento que determine el índice de eficiencia y el comportamiento de la productividad de cada agente en las áreas de atención, ayudará a mejorar los objetivos que se han planteado para el desarrollo y evolución del Call Center.

El balance Score Card mostrará detalladamente los indicadores fundamentales de la operación en relación a las llamadas que realicen los agentes.

	Peso	Estándar	
Productividad			
Conexiones / hr	30%	28	↑
Rango de Negociaciones acordadas	25%	70%	↑
Eficiencia			
Rango de negociaciones cumplidas	20%	65%	
Cobranza	5%	60%	
Desempeño			
Tiempo hablando	5%	95%	
Calidad			
Monitoreo de gestión	15%	100%	

Puntaje - penalización = Puntaje Final

Figura 1.20 Indicadores Balance Score Card
Ref.: Estándar ELTCO

Debe contener todos los puntos de evaluación de los ejecutivos, lo que dotará de resultados reales del desempeño, los cuales se convertirán en el arma principal para poder llevar al agente al cumplimiento de sus metas individuales y en conjunto, al logro de los objetivos de la empresa.

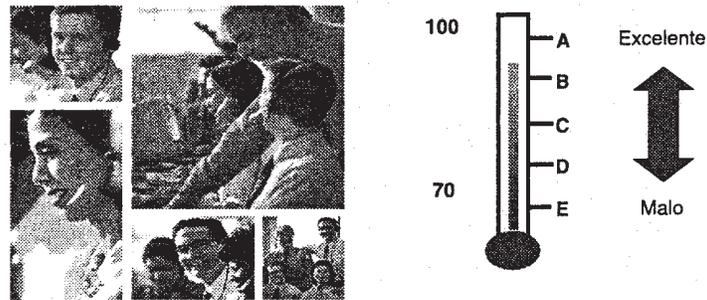


Figura 1.21 Calificación Agente
Ref.: Estándar ELTCO

El objetivo del BSC es medir de forma estructurada y precisa, la participación individual de los agentes en la gestión telefónica en términos de: productividad, eficiencia, desempeño y calidad.

- Políticas y facultades

Para construir el BSC se requiere de un fundamento estructurado que soporte su función.

El BSC debe verse como el medio más robusto en la medición de los agentes:

- Alcance
- Vigencia
- Frecuencia
- Criterios de evaluación
- Roles y responsabilidades
- Definición de pesos y estándares
- Revisión de resultados
- Plan de incentivos

■ Componentes

○ Productividad

Capacidad de procesamiento del CC: volúmenes y tiempo. Con el fin de evaluar la productividad individual del agente telefónico, en este rubro, se consideran variables como:

- Número de conexiones por hora
- Contactos realizados
- Contactos directos
- Compromisos de pago cumplidos

○ Eficiencia

Participación en las metas del negocio. Un aspecto importante se refiere a la eficiencia, en donde se pretende ver el resultado del uso de las técnicas de negociación, asesoría y empatía con el cliente. El resultado se puede ver en metas:

- Individuales
- Grupales

○ Desempeño

Grado de enfoque a las actividades críticas del agente. Esta variable está orientada a la medición de la actividad física y más importante del agente:

- Tiempo que está hablando.
- Tiempo que toma para el registro de gestiones.
- Tiempo invertido en la consulta y ubicación de información y antecedentes.

- Calidad
Cumplimiento de las expectativas mínimas requeridas por el negocio. Uno de los componentes más importantes se refiere a la calidad de la gestión telefónica, ya que en ella se respalda la imagen del negocio ante los clientes y se garantiza el logro de los objetivos:
 - Monitoreo

2. REQUERIMIENTOS, ANÁLISIS Y DISEÑO

Para el análisis y diseño de este prototipo utilizaremos la metodología OMT (Object Modeling Technique), la cual fue creada por James Rumbaugh y Michael Blaha en 1991. OMT es una metodología de análisis y diseño orientadas a objetos, abierta (no propietaria).

Las fases que conforman a la metodología OMT son:

- Análisis

Es una abstracción resumida y precisa de lo que debe de hacer el sistema deseado y no de la forma en que se hará.
- Diseño del sistema

Durante esta fase el sistema se organiza en subsistemas basándose tanto en la estructura del análisis como en la arquitectura propuesta. Se selecciona una estrategia para afrontar el problema.
- Diseño de objetos

Se construye un modelo de diseño basándose en el modelo de análisis, pero incorporando detalles de implementación. El diseño de objetos se

centra en las estructuras de datos y algoritmos que son necesarios para implementar cada clase.

- Implementación

Las clases de objetos y relaciones desarrolladas durante el análisis de objetos se traducen finalmente a una implementación concreta.

La metodología OMT emplea tres clases de modelos para describir el sistema:

- Modelo de objetos

Describe la estructura estática de los objetos del sistema (identidad, relaciones con otros objetos, atributos y operaciones). El objetivo es capturar aquellos conceptos del mundo real que sean importantes para la aplicación. Se representa mediante diagramas de objetos.

- Modelo dinámico

Describe los aspectos de un sistema que tratan de la temporización y secuencia de operaciones (sucesos que marcan los cambios, secuencias de sucesos, estados que definen el contexto para los sucesos) y la organización de sucesos y estados. Captura el control, aquel aspecto de un sistema que describe las secuencias de operaciones que se producen sin tener en cuenta lo que hagan las operaciones, aquello a lo que afecten o la forma en que están implementadas. Se representa gráficamente mediante diagramas de estado.

- Modelo funcional

Describe las transformaciones de valores de datos (funciones, correspondencias, restricciones y dependencias funcionales) que ocurren en el sistema. Captura lo que hace el sistema,

independientemente de cuando se haga o de la forma en que se haga.
Se representa mediante diagramas de flujo de datos

2.1. REQUERIMIENTOS

2.1.1. REQUERIMIENTOS FUNCIONALES

El presente prototipo deberá permitir a los administradores de un Call Center de Cobranza supervisar y auditar la gestión realizada por los agentes telefónicos, así como también los resultados de productividad obtenidos por los mismos, quienes serán calificados en base a indicadores clave. Por lo que el prototipo deberá proveer las siguientes funcionalidades:

- **Administración de Indicadores Clave**
Los cuales medirán los resultados de gestión obtenidos por los agentes telefónicos.

- **Administración de Calificaciones**
Definir calificaciones en base al porcentaje obtenido en los indicadores clave, las cuales permitirán identificar el trabajo realizado por los agentes.

- **Construcción del Informe EWM**
Realizar el proceso de cálculo de los indicadores clave y la calificación obtenida por cada uno de los agentes, en base a las gestiones y llamadas realizadas por ellos en una fecha dada.

- **Consulta del Informe EWM**
El EWM deberá estar disponible siempre para la consulta del Jefe o Supervisor del Call Center, con el objetivo de permitir el monitoreo y control de los agentes.

- **Administración de Plantillas de Evaluación**
Facilitar al auditor de gestión la elaboración de plantillas, las cuales serán utilizadas para la evaluación de la calidad de gestión realizada por un agente.

- **Auditoría de Gestión**
Permitir al auditor de gestión evaluar la calidad de gestión que realiza un agente, empleando para ello las plantillas de evaluación previamente diseñadas. Esta evaluación será aplicada durante la intervención de llamadas que realiza el auditor a un agente.

Todos los procesos anteriores explicados, deberán ser definidos de forma diferenciada dependiendo de la ciudad, producto y segmento donde el Call Center realice la cobranza, además de permitir la generación de reportes de gráficos estadísticos.

2.1.2. REQUERIMIENTOS DE INTERFACES EXTERNAS

2.1.2.1. Usuarios

La interfaz del usuario deber ser orientada a página web y el manejo del prototipo se realizará a través del teclado y el ratón.

2.1.2.2. Hardware

- **Servidor de DBB**
 - Pentium IV 2.6GB
 - 4 GB Ram
 - Disco duro con espacio libre de 2 Gigas
 - Tarjeta de red de 100 mbps
 - Tarjeta de video SVGA
 - Unidad de CD

- Servidor de Aplicaciones
 - Pentium IV 2.6GB
 - 4 GB Ram
 - Disco duro con espacio libre de 2 Gigas
 - Tarjeta de red de 100 mbps
 - Tarjeta de video SVGA
 - Unidad de CD

- Cliente
 - Pentium IV 1.2GB mínimo
 - 1 Gb Ram
 - Disco duro con espacio de 50 Mb mínimo
 - Tarjeta de red de 100 mbps
 - Tarjeta de video SVGA

2.1.2.3. Software

- Servidor de BDD
 - Windows 2008 Server o Superior
 - Microsoft SQL Server 2012 o superior

- Servidor de aplicaciones
 - Windows 2008 Server o Superior
 - .Net Framework 3.5.1 o superior
 - Internet Information Server 7 o superior

- Cliente
 - Windows 2008 Server o Superior, Windows 7 o superior
 - Cualquier navegador WEB versiones actualizadas

2.1.3. REQUERIMIENTOS DE DESEMPEÑO

Se requiere que el prototipo brinde los mejores niveles de calidad, eficiencia y oportunidad, para lo cual se requiere que sea un sistema abierto n-capas que pueda integrarse con el Sistema de Gestión Telefónica del Call Center. Además deberá tener las siguientes características:

- **Autonomía**
Es decir que cada componente sea independiente del resto para facilitar el mantenimiento del software.
- **Confiabilidad**
Facilitar el manejo de la complejidad disminuyendo el riesgo de caídas.
- **Disponibilidad**
Asegurar que las aplicaciones permanezcan siempre accesibles y operativas para los usuarios.
- **Escalabilidad**
Permitir la capacidad de expandir las aplicaciones sin interrumpir o degradar el servicio.

2.2. ANÁLISIS

2.2.1. DESCRIPCIÓN DEL PROBLEMA

El problema consiste en crear un prototipo para la administración automatizada de un call center de cobranza, la cual se concentra en las actividades que realizan: el supervisor, el auditor de gestión y el jefe del call center.

El supervisor controla la gestión y productividad de los agentes telefónicos mediante la consulta de los resultados de gestión obtenidos, así como también el registro de calificaciones de cada uno de ellos, los cuales son valorados con los Indicadores Clave. El prototipo deberá fusionar estas consultas y generar un informe denominado EWM (Early Warning Monitor), el cual permitirá la consulta por diferentes criterios tales como: Empresa, ciudad, producto, rango de fechas, calificación, segmento, agente. En base a estos datos el supervisor analiza qué agente debe ser evaluado por el auditor de gestión.

El auditor de gestión, controla la calidad de la gestión realizada por los agentes telefónicos, consultando resultados de evaluaciones históricas y/o realizando evaluaciones en la intervención de llamadas utilizando para ello plantillas previamente creadas por el mismo auditor a manera de cuestionarios.

El Jefe del call center crea los valores óptimos y adjudica pesos a los indicadores clave que medirán a los agentes telefónicos. Además analiza los resultados de la auditoría de gestión con el objetivo de establecer acciones correctivas para los agentes telefónicos como la capacitación, y asesoría. Y crea o redefine las estrategias del mejor día para llamar o la mejor hora para llamar, de acuerdo a los resultados de productividad obtenidos. El prototipo deberá proporcionar toda la información necesaria para esta gestión a manera de gráficos estadísticos, y no considerará la creación de estrategias.

2.2.2. MODELO DE OBJETOS

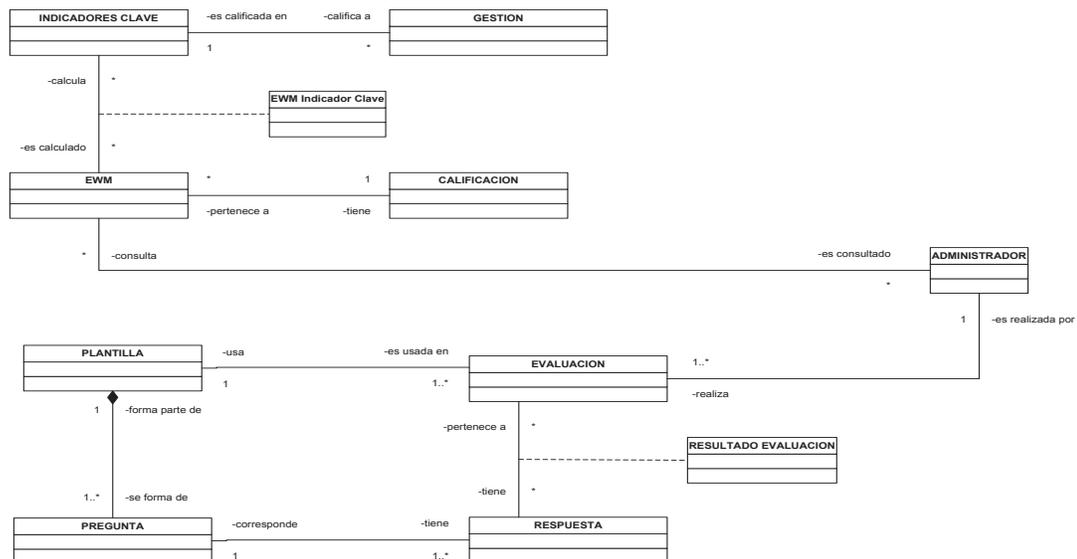
2.2.2.1. Identificación de Clases

En base a la descripción del problema se identificaron las siguientes clases:

- Indicadores Clave
- Calificación
- EWM

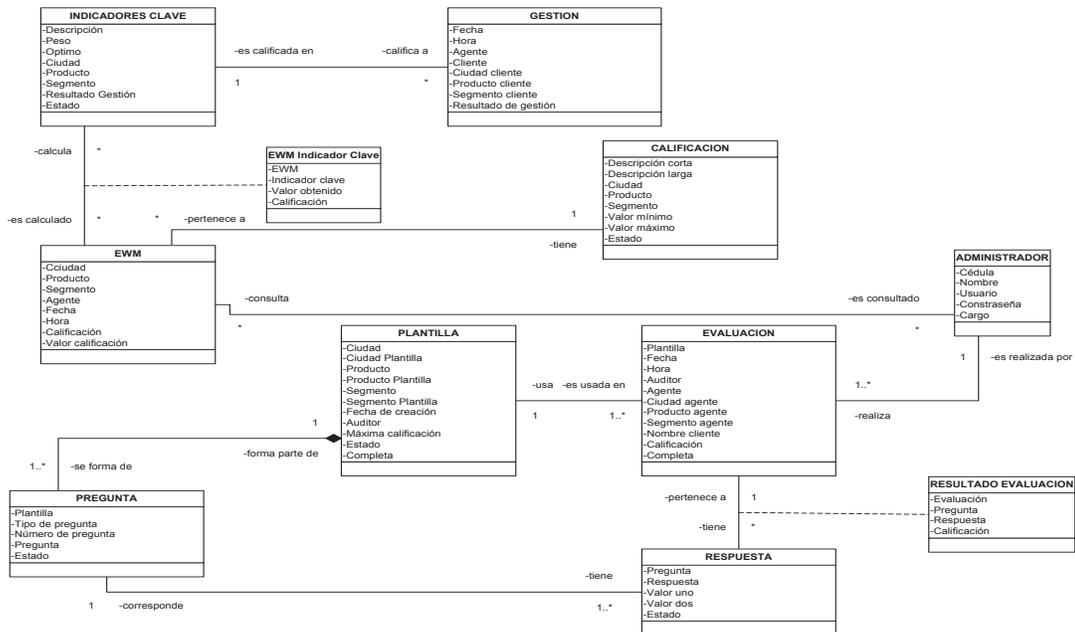
- EWM IC
- Plantilla
- Evaluación
- Pregunta
- Respuesta
- Resultado Evaluación
- Administrador
- Gestión

2.2.2.2. Identificación de Asociaciones y Agregaciones



Ref.: Autor

2.2.2.3. Identificación de Atributos



Ref. Autor

2.2.2.4. Diccionario de datos Clases y Atributos

Tabla 2.1. Análisis: Diccionario de Datos Clase INDICADOR CLAVE

CLASE	INDICADOR CLAVE	Representan los puntos clave de observación del desempeño de la operación del Call Center, ligados a la estrategia del negocio, los cuales son calculados con los resultados de la gestión realizada por el agente. Los mismos que son: Hit Rate, Contact Rate, Promise Rate y Eficiencia.
Atributos	Descripción	Nombre del indicador clave
	Peso	Porcentaje que participará en la calificación del EWM.
	Óptimo	Porcentaje deseado al cual debe llegar el agente telefónico al realizar la gestión de cobro.
	Ciudad	Ciudad en la que se aplicará este indicador.
	Producto	Producto en el que se aplicará este indicador.

	Segmento	Segmento en el que se aplicará este indicador.
	Estado	Indica si el indicador está activo o no.

Tabla 2.2. Análisis: Diccionario de Datos Clase EWM

CLASE	EWM (Early Warning Monitor)	Es el informe que permitirá evaluar en línea el desempeño de los agentes telefónicos a través de las variables operativas que impactan directamente en los resultados de operación como son los indicadores clave.
Atributos	Ciudad	Ciudad que se calificará la gestión.
	Producto	Producto que se calificará la gestión.
	Segmento	Segmento que se calificará la gestión.
	Agente	Agente que se calificará la gestión.
	Fecha	Fecha en la que el agente gestionó.
	Hora	Hora en la que el agente gestionó.
	Calificación	Calificación obtenida por el agente por la gestión realizada, en una fecha y hora determinada.

Tabla 2.3. Análisis: Diccionario de Datos Clase EWM INDICADOR CLAVE

CLASE	EWM INDICADOR CLAVE	Contiene la información del detalle de del valor obtenido y la respectiva calificación en cada indicador clave en un EWM.
Atributos	Indicador clave	Indicador Clave.
	Valor Obtenido	Porcentaje registrado por el agente para un indicador clave.
	Calificación	Calificación obtenida por el agente en un indicador clave en letras.
	Valor Calificación	Calificación obtenida por el agente en un indicador clave en porcentaje.

Tabla 2.4. Análisis: Diccionario de Datos Clase CALIFICACIÓN

CLASE	CALIFICACIÓN	Valor obtenido por cada agente telefónico de acuerdo a los resultados registrados por su gestión de cobro en base a los indicadores claves preestablecidos.
Atributos	Descripción Corta	Un nombre abreviado para una calificación.
	Descripción Larga	Un nombre extenso para una calificación.
	Ciudad	Ciudad en la que se aplicará la calificación.
	Producto	Producto en el que se aplicará la calificación.
	Segmento	Segmento en el que se aplicará la calificación.
	Valor mínimo	Porcentaje mínimo aceptable para obtener la calificación.
	Valor máximo	Porcentaje máximo aceptable para obtener la calificación.
	Estado	Indica si la calificación está activa o no.

Tabla 2.5. Análisis: Diccionario de Datos Clase PLANTILLA

CLASE	PLANTILLA	Son formatos predefinidos de evaluaciones de auditoría de gestión para realizar la evaluación de los agentes.
Atributos	Ciudad	Ciudad a la que se evaluará con la plantilla.
	Ciudad Plantilla	Nombre de ciudad a la que se evaluará con la plantilla, dado por el auditor.
	Producto	Producto al que se evaluará con la plantilla.
	Producto Plantilla	Nombre del producto al que se evaluará con la plantilla, dado por el auditor.
	Segmento	Segmento al que se evaluará con la plantilla.
	Segmento Plantilla	Nombre del segmento al que se evaluará con la plantilla, dado por el auditor.
	Fecha de creación	Fecha en la que se creó la plantilla.
	Auditor	Auditor que creó la plantilla.

	Máxima Calificación	Puntaje máximo que registrará la plantilla.
	Estado	Indica si la plantilla está activa o no.
	Completa	Indica si la plantilla tiene creadas las preguntas o no.

Tabla 2.6. Análisis: Diccionario de Datos Clase PREGUNTA

CLASE	PREGUNTA	
		Es el listado de preguntas que consta en cada una de las plantillas de evaluación, el cual es desarrollado a manera de check list.
Atributos	Plantilla	Plantilla a la cual pertenece esta pregunta.
	Tipo de pregunta	Tipo de pregunta.
	Número de pregunta	Número secuencial de las preguntas por plantilla.
	Pregunta	Es la pregunta como tal.
	Estado	Indica si la pregunta está activa o no.

Tabla 2.7. Análisis: Diccionario de Datos Clase RESPUESTA

CLASE	RESPUESTA	
		Son las posibles respuestas de la pregunta formulada en la plantilla.
Atributos	Pregunta	Pregunta a la que pertenece la respuesta.
	Respuesta	La posible respuesta a la pregunta.
	Valor uno	El puntaje que tendrá por la respuesta.
	Valor dos	El puntaje que tendrá por la respuesta, se utilizará este cuando el tipo de pregunta es de rangos.
	Estado	Indica si la respuesta está activa o no.

Tabla 2.8. Análisis: Diccionario de Datos Clase EVALUACIÓN

CLASE	EVALUACIÓN	Es el registro de la calificación obtenida por cada agente telefónico en las evaluaciones realizadas por el auditor de gestión durante la intervención de llamadas.
Atributos	Plantilla	Plantilla en la que se realizará la evaluación.
	Fecha	Fecha de la evaluación.
	Hora	Hora de la evaluación.
	Auditor	Auditor que realizará la evaluación.
	Agente	Agente a quien se evaluará.
	Ciudad agente	Ciudad a la que gestiona el agente.
	Producto agente	Producto al que gestiona el agente.
	Segmento agente	Segmento al que gestiona el agente.
	Nombre cliente	Nombre del cliente al que está gestionando el agente el momento de la evaluación.
	Calificación	Calificación obtenida por el agente en la evaluación.
	Completa	Indica si la evaluación se ha completado o no.

Tabla 2.9. Análisis: Diccionario de Datos Clase RESULTADO EVALUACIÓN

CLASE	RESULTADO EVALUACIÓN	Contiene la calificación obtenida en cada una de las respuestas durante la evaluación.
Atributos	Evaluación	Evaluación a la que corresponde el resultado.
	Respuesta	Respuesta que se calificará.
	Calificación	Calificación de la respuesta en la evaluación.

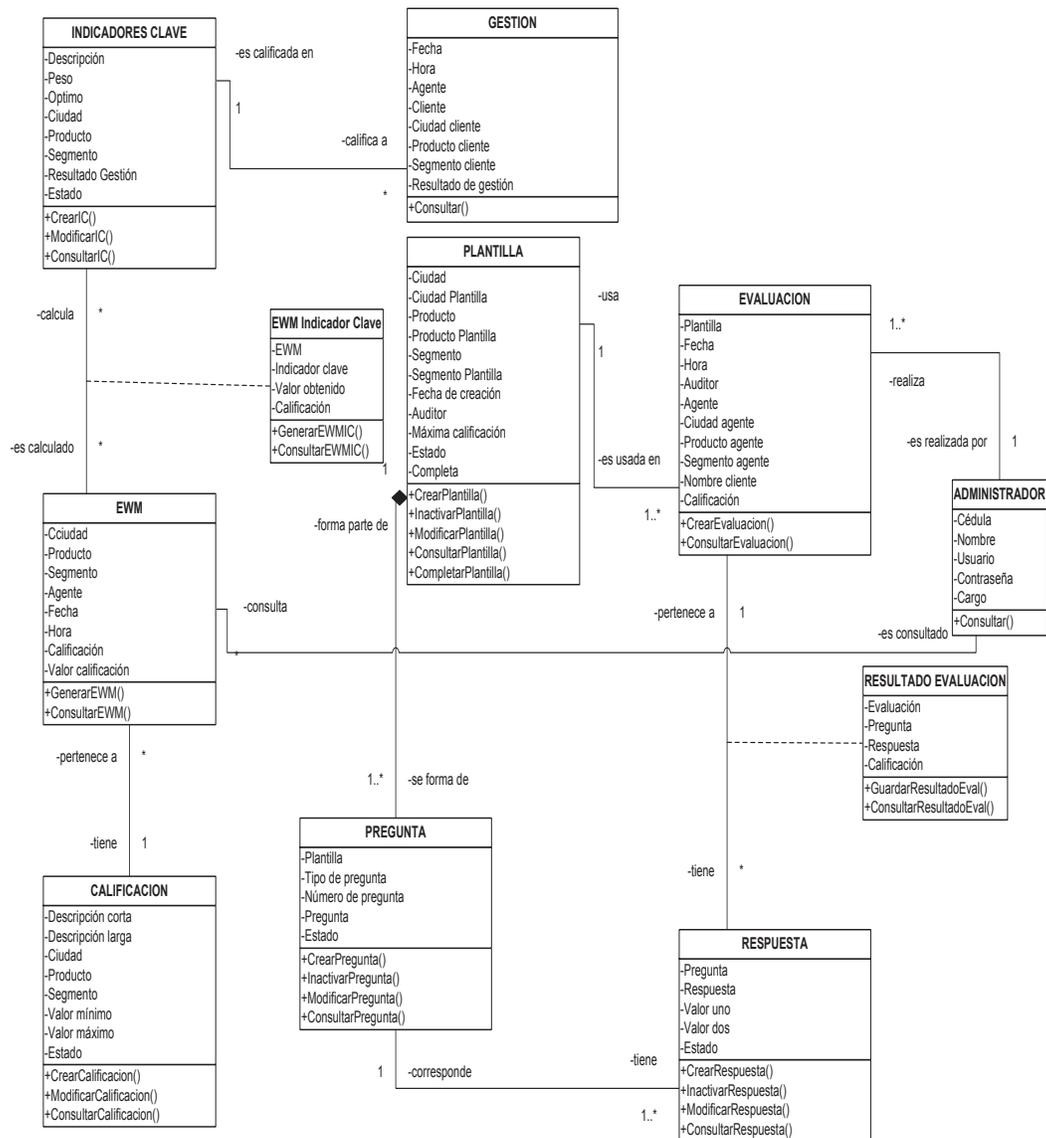
Tabla 2.10. Análisis: Diccionario de Datos Clase ADMINISTRADOR

CLASE	ADMINISTRADOR	Personas que administran el Call Center.
Atributos	Cédula	Cédula de un administrador.
	Nombre	Nombres y apellidos de un administrador.
	Contraseña	Clave con la que el administrador ingresa al sistema.
	Cargo	Cargo al que pertenecen, estos son: Supervisor, Auditor de gestión y Jefe del Call Center.

Tabla 2.11. Análisis: Diccionario de Datos Clase GESTIÓN

CLASE	GESTIÓN	Gestiones realizadas por los agentes a los clientes en mora.
Atributos	Fecha	Fecha en la que se realizó la gestión.
	Hora	Hora en la que se realizó la gestión.
	Agente	Agente que gestionó.
	Cliente	Cliente gestionado.
	Ciudad cliente	Ciudad del cliente gestionado.
	Producto cliente	Producto del cliente gestionado.
	Segmento cliente	Segmento del cliente gestionado.
	Resultado de gestión	Resultado de la gestión realizada.

2.2.2.5. Diagrama del modelo de Objetos



Ref.: Autor

2.2.3. MODELO DINÁMICO

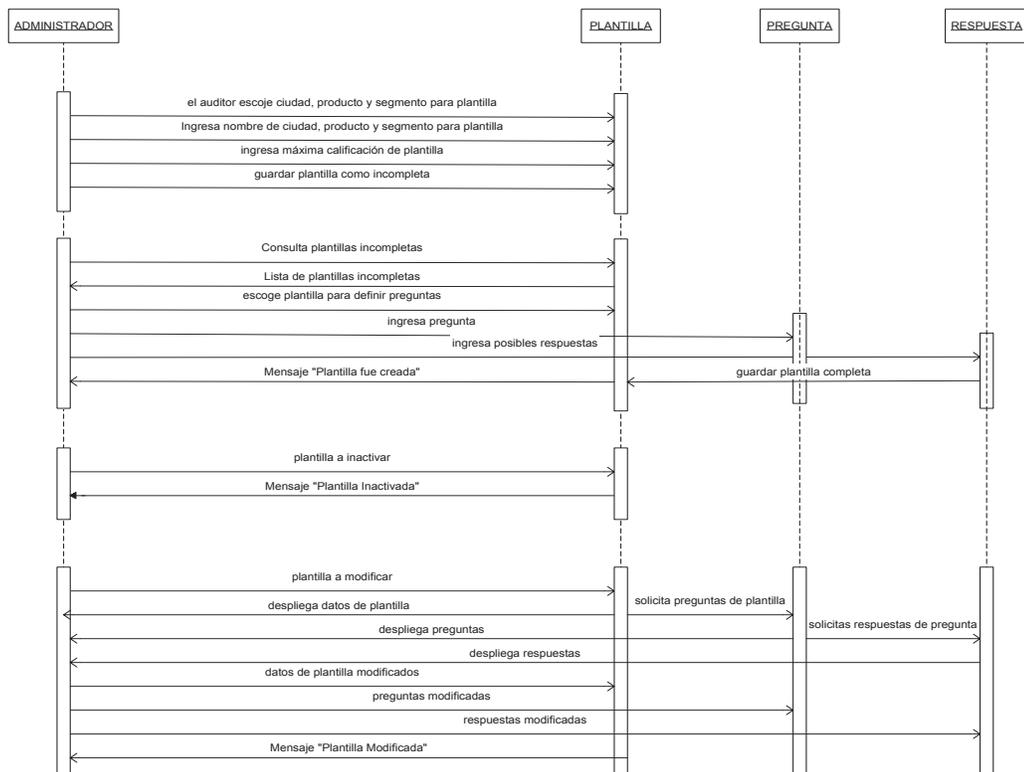
2.2.3.1. Identificación de escenarios

A partir del modelo de objetos se identificaron los siguientes escenarios:

- Seguimiento de sucesos para una plantilla
- Seguimiento de sucesos para una evaluación
- Seguimiento de sucesos para un indicador clave
- Seguimiento de sucesos para una calificación
- Seguimiento de sucesos para un EWM

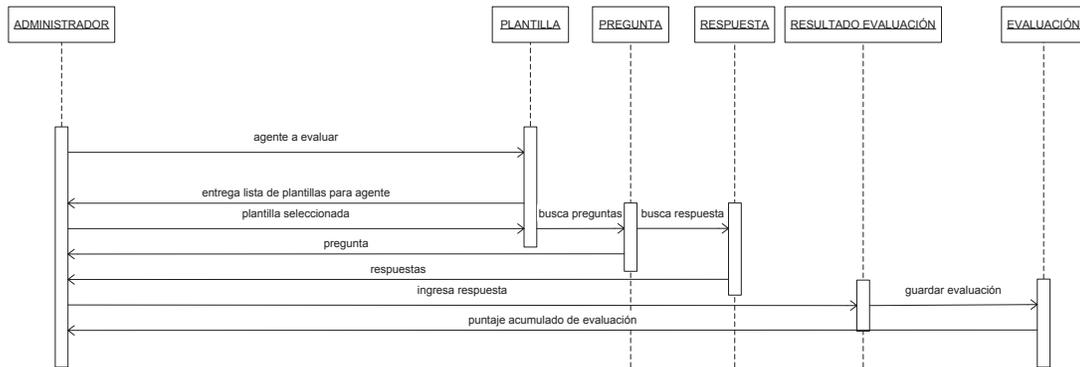
2.2.3.2. Identificación de sucesos entre objetos

2.2.3.2.1. Diagrama de seguimiento de sucesos para una PLANTILLA



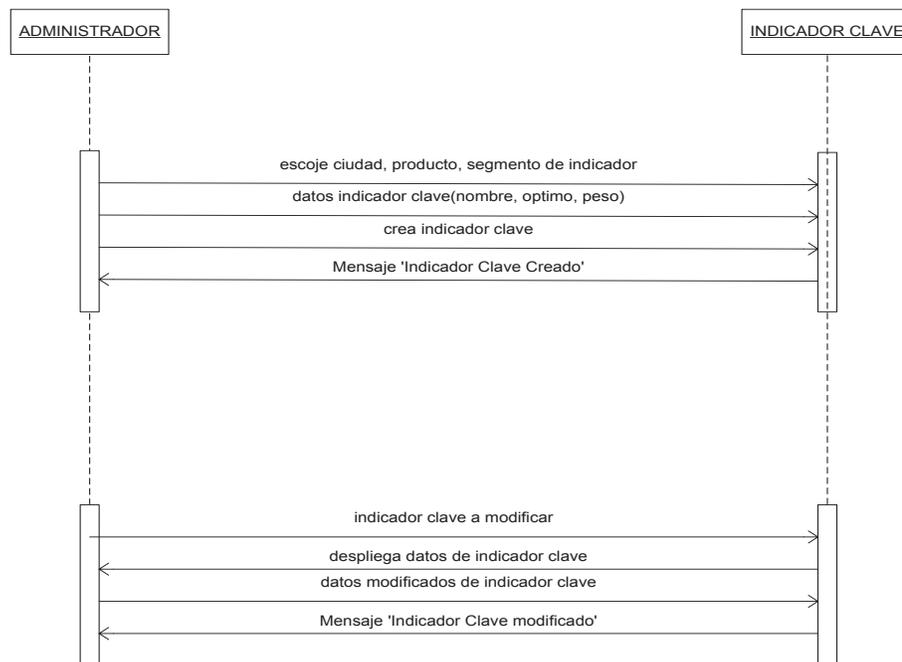
Ref.: Autor

2.2.3.2.2. Diagrama de seguimiento de sucesos para una EVALUACIÓN



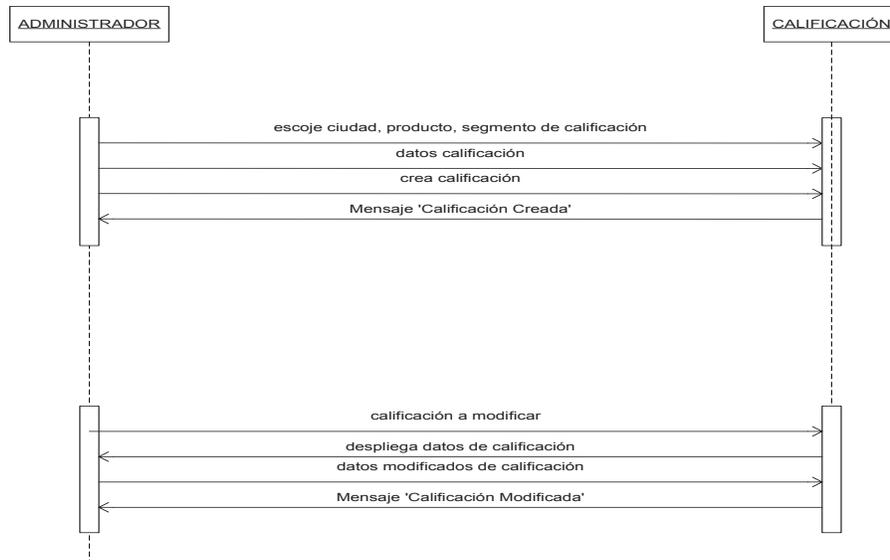
Ref.: Autor

2.2.3.2.3. Diagrama de seguimiento de sucesos para un INDICADOR CLAVE



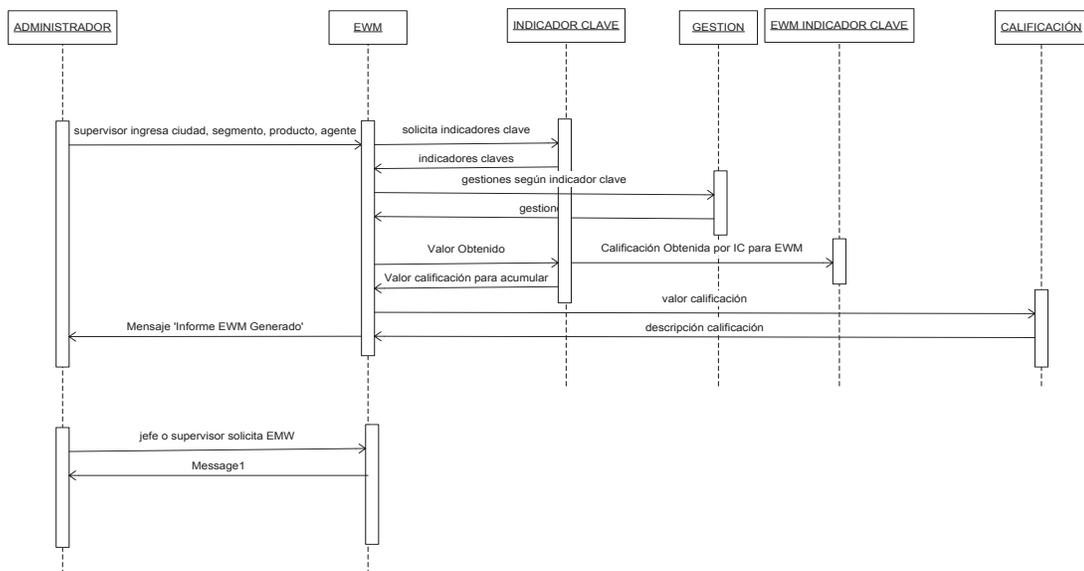
Ref.: Autor

2.2.3.2.4. Diagrama de seguimiento de sucesos para una CALIFICACIÓN



Ref.: Autor

2.2.3.2.5. Diagrama de seguimiento de sucesos para un EWM



2.2.3.3. Definición de sucesos

Tabla 2.12. Análisis: Escenario SEGUIMIENTO DE SUCESOS PARA UNA PLANTILLA

SEGUIMIENTO DE SUCESOS PARA UNA PLANTILLA		
ESCENARIO	Agente	Descripción
Suceso		
Crear Plantilla	Administrador, Plantilla	Guarda una plantilla.
Inactivar Plantilla	Administrador, Plantilla	Cambia el estado de la plantilla a inactiva.
Modificar Plantilla	Administrador, Plantilla	Actualiza los atributos de la plantilla.
Consultar Plantilla	Administrador, Plantilla	Selecciona una plantilla, para obtener sus atributos.
Completar Plantilla	Plantilla, Respuesta, Pregunta	Registra el estado de finalización del proceso de elaboración de la plantilla.
Crear Pregunta	Administrador, Pregunta	Inserta una pregunta.
Consultar Pregunta	Administrador, Pregunta	Selecciona una pregunta, para obtener sus atributos.
Modificar Pregunta	Administrador, Pregunta	Actualiza los atributos de una pregunta.
Crear Respuesta	Administrador, Respuesta	Son las posibles respuestas de la pregunta formulada en la plantilla.
Consultar Respuesta	Administrador, Respuesta	Selecciona una respuesta, para obtener sus atributos.
Modificar Respuesta	Administrador, Respuesta	Actualiza los atributos de una respuesta.

Tabla 2.13. Análisis: Escenario SEGUIMIENTO DE SUCESOS PARA UNA EVALUACIÓN

SEGUIMIENTO DE SUCESOS PARA UNA EVALUACIÓN		
ESCENARIO	Agente	Descripción
Suceso		
Crear Evaluación	Administrador, Evaluación, Plantilla, Pregunta, Respuesta, Resultado Evaluación	Inserta una evaluación.
Consultar Evaluación	Administrador, Evaluación	Busca una evaluación, para obtener sus atributos.

Guardar Eval	Resultado	Resultado Evaluación, Evaluación, Respuesta	Graba la calificación obtenida por pregunta en una evaluación.
Consultar Eval	Resultado	Resultado Evaluación, Evaluación, Respuesta	Selecciona la calificación obtenida por pregunta en una evaluación.

Tabla 2.14. Análisis: Escenario SEGUIMIENTO DE SUCESOS PARA UN INDICADOR CLAVE

SEGUIMIENTO DE SUCESOS PARA UN INDICADOR CLAVE		
ESCENARIO	Agente	Descripción
Suceso	Agente	Descripción
Crear Indicador Clave	Administrador, Indicador Clave	Realiza la inserción de un indicador clave
Modificar Indicador Clave	Administrador, Indicador Clave	Permite cambiar el valor óptimo y el peso de un indicador clave.
Consultar Indicador Clave	Administrador, Indicador Clave	Realiza la consulta de los indicadores clave.

Tabla 2.15. Análisis: Escenario SEGUIMIENTO DE SUCESOS PARA UNA CALIFICACIÓN

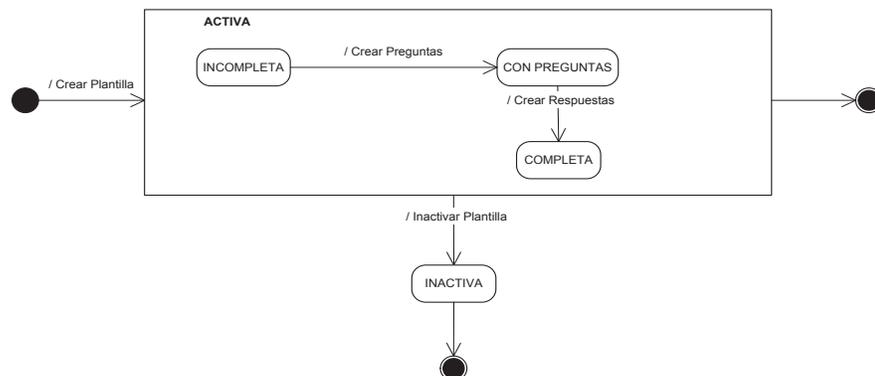
SEGUIMIENTO DE SUCESOS PARA UNA CALIFICACIÓN		
ESCENARIO	Agente	Descripción
Suceso	Agente	Descripción
Crear Calificación	Administrador, Calificación	Permite la inserción de las diferentes calificaciones que podrán existir en un EWM.
Modificar Calificación	Administrador, Calificación	Actualiza los atributos de la calificación.
Consultar Calificación	Administrador, Calificación	Selecciona una calificación creada, para obtener sus atributos

Tabla 2.16. Análisis: Escenario SEGUIMIENTO DE SUCESOS PARA UN EWM

ESCENARIO	SEGUIMIENTO DE SUCESOS PARA UN EWM	
Suceso	Agente	Descripción
Generar EWM	Administrador, EWM, EWM IC	Realiza el proceso de construcción del EWM para un agente en una fecha dada.
Consultar EWM	Administrador, EWM	Desplegar el EWM de acuerdo a parámetros especificados tales como: Fecha, Ciudad, Producto, Segmento.
Generar EWM IC	EWM, Indicador Clave, EWM IC	Calcula la calificación de cada indicador clave en un EWM.
Consultar EWM IC	EWM IC, Indicador Clave	Consulta la calificación de cada indicador clave en un EWM para que esta sea acumulada y posteriormente almacenada en el EWM.
Consultar Gestión	EWM, Indicador Clave, Gestión	Selecciona una gestión, para obtener sus atributos.

2.2.3.4. Diagrama de estados

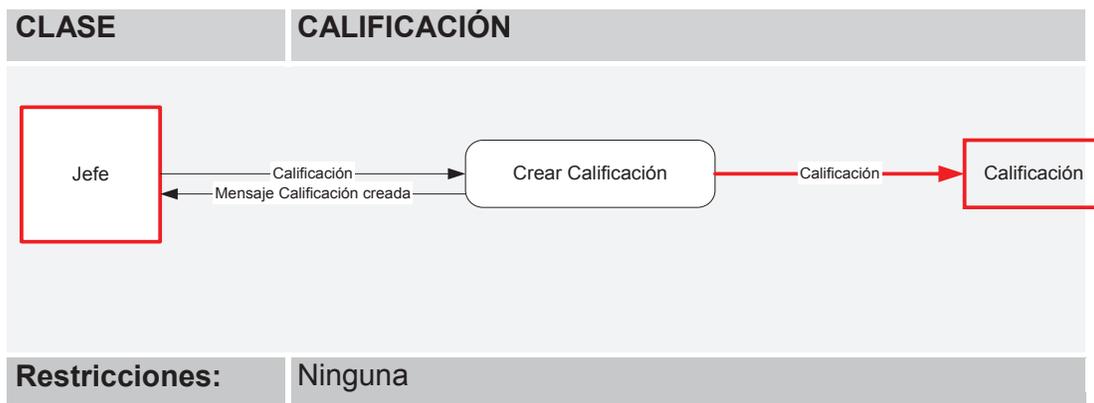
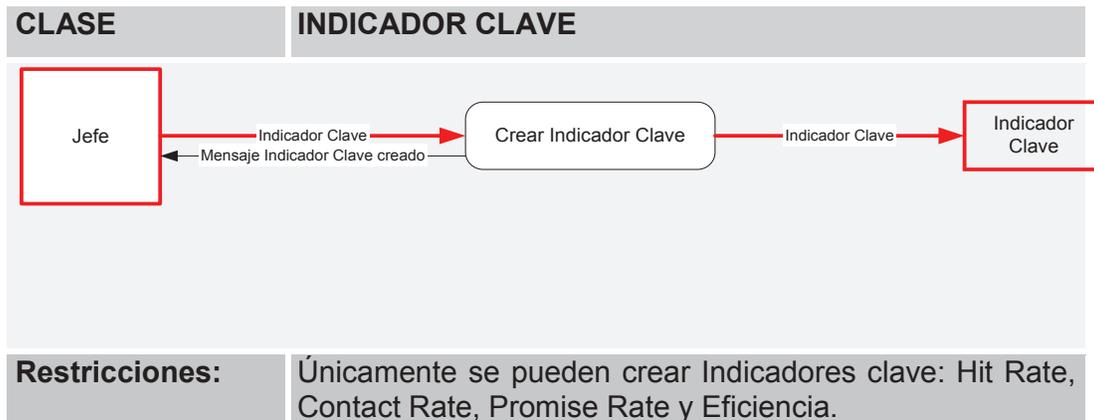
2.2.3.4.1. Diagrama de estados de una PLANTILLA

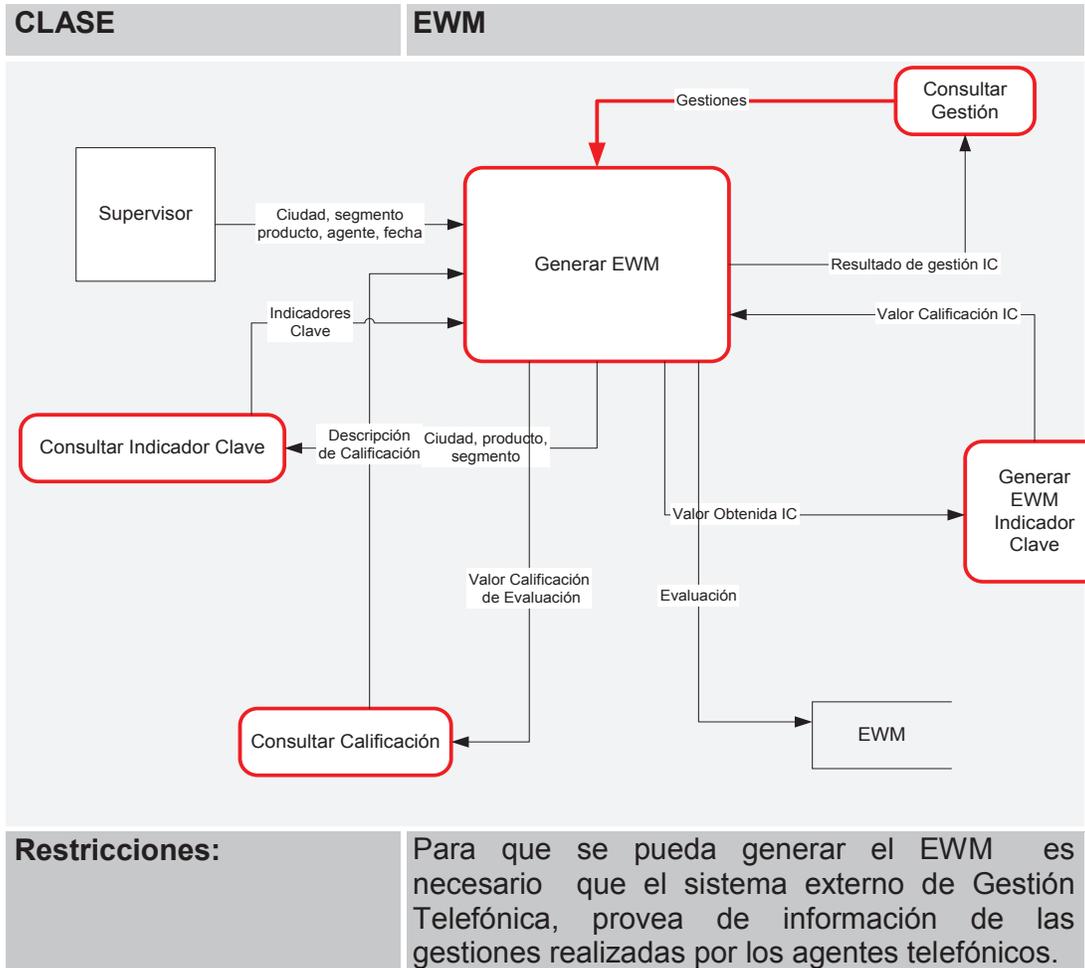


Ref.: Autor

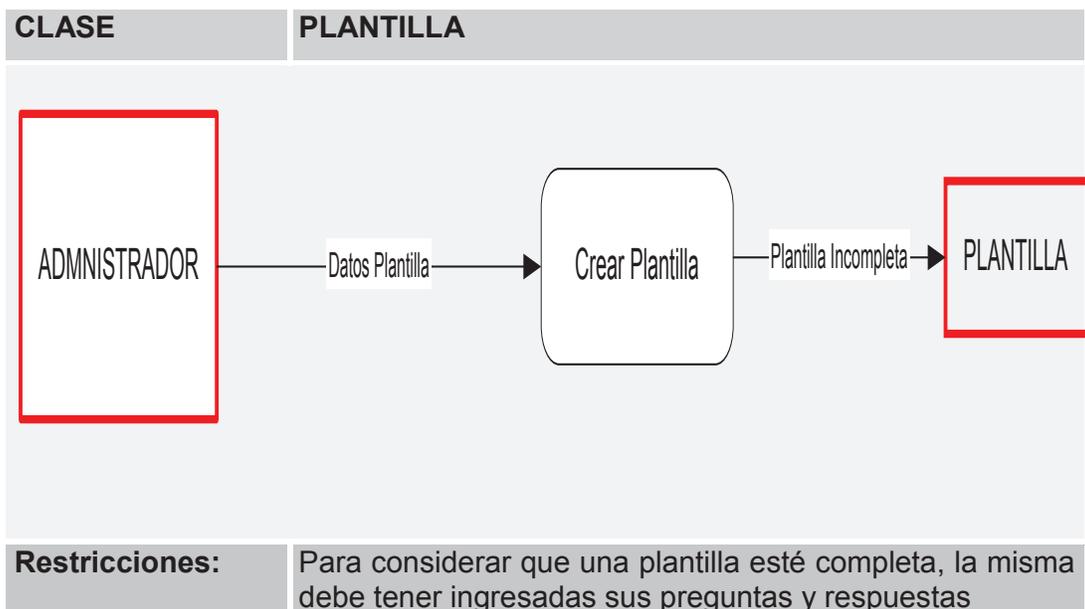
2.2.4. MODELO FUNCIONAL

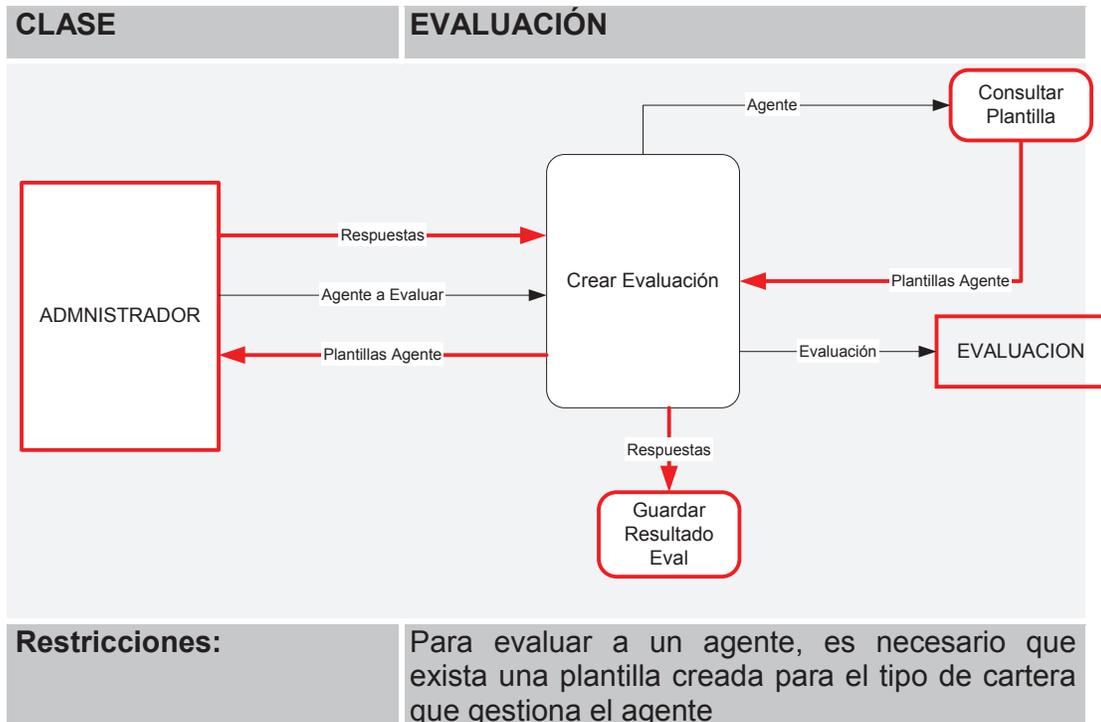
2.2.4.1. Diagrama de Flujo de Datos para el subsistema EWM





2.2.4.2. Diagrama de Flujo de Datos para el subsistema EVALUACIÓN





2.3. DISEÑO DEL SISTEMA

2.3.1. ORGANIZACIÓN DEL SISTEMA EN SUBSISTEMAS

La descomposición de sistemas en subsistemas se puede organizar como una secuencia de *Capas* horizontales y/o en *Particiones* verticales, nuestro prototipo requiere una mezcla de capas y particiones como se explica a continuación.

2.3.1.1. Capas

Se eligió una estructura n-capas, las capas son las siguientes:

- CAPA DE DATOS

La capa de datos estará ubicada en la base de datos, el acceso a los datos se lo manejará a través de procedimientos almacenados (stored procedures). El lenguaje a utilizar será Transact SQL.

- CAPA LÓGICA DEL NEGOCIO

Esta capa estará conformada a su vez por tres sub-capas muy bien identificadas.

- Capa de lógica de conexión a repositorio de datos

Como el repositorio de datos en este caso es una base de datos, a esta capa, se le conoce como motor de persistencia, se utilizará un motor de persistencia propio, ya que se necesita tener completo control sobre esta capa, esta es la razón por la cual no se utilizará Application Blocks, un motor de persistencia desarrollado por Microsoft y de libre distribución para usuarios del .NET.

- Capa de lógica de invocación de procedimientos

Se encargará de formar la sentencia SQL necesaria para invocar al stored procedure.

- Capa de lógica del negocio pura

Se encargará de todos los cálculos necesarios antes de presentar información al usuario o antes de enviar información a la base de datos, es importante poner toda la lógica del negocio en esta parte para no tener desparramada la misma a lo largo del resto de capas.

- CAPA PRESENTACIÓN

Es lo que ve el usuario. Será programado con ASP.NET, lenguaje que procesa código y lo convierte en lenguaje HTML, el que puede ser interpretado por el browser, de cualquier navegador de WEB

Para la validación de datos ingresados por el usuario se utilizará javascript, lenguaje de programación que permite realizar eventos sobre el browser, evitando así el envío de datos que serán rechazados por el servidor de aplicación, con esto se mejora el rendimiento de la aplicación en general, al reducir el trabajo de validación de datos en el servidor.

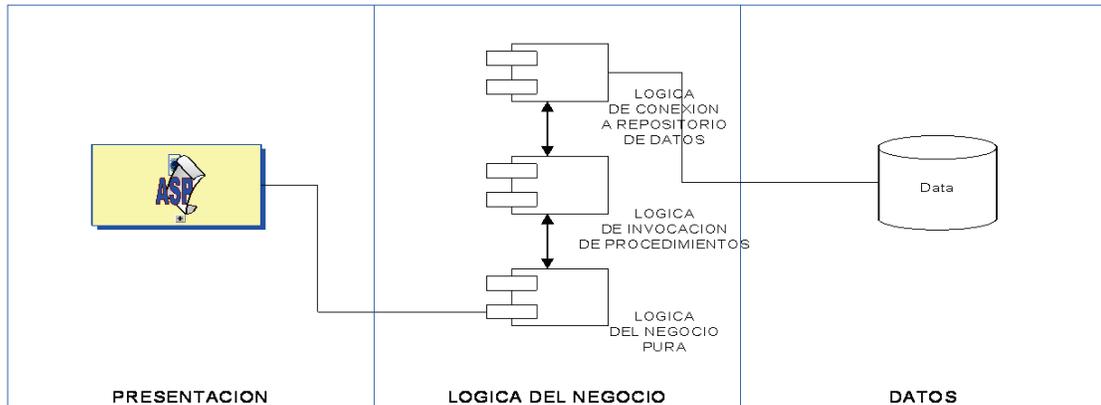


Figura 2.1 Estructura de Capas

Ref.: Autor

2.3.1.2. Particiones

Al diseñar el sistema, se vio la necesidad de subdividirlo en 2 subsistemas:

- SUBSISTEMA DE AUDITORIA DE GESTIÓN

Es donde se llevará a cabo todo el proceso de auditoría de la gestión telefónica de un agente.

- SUBSISTEMA DE EWM

Es el subsistema encargado del monitoreo de estándares de gestión (indicadores clave).



Figura 2.2 Estructura de Subsistemas

Ref.: Autor

2.3.2. IDENTIFICACIÓN DE LA CONCURRENCIA

Un objetivo importante del diseño del sistema es identificar los objetos que deben estar activados concurrentemente y los objetos que tienen actividad que sea mutuamente exclusiva.

En este prototipo no existe concurrencia entre ninguno de los objetos.

2.3.3. ASIGNACIÓN DE SUBSISTEMAS A PROCESADORES Y TAREAS

Los subsistemas de auditoría telefónica y EWM, están ubicados en un mismo servidor, no se vio la necesidad de separar la base de datos del servidor de aplicación, ya que no existe concurrencia en el prototipo.

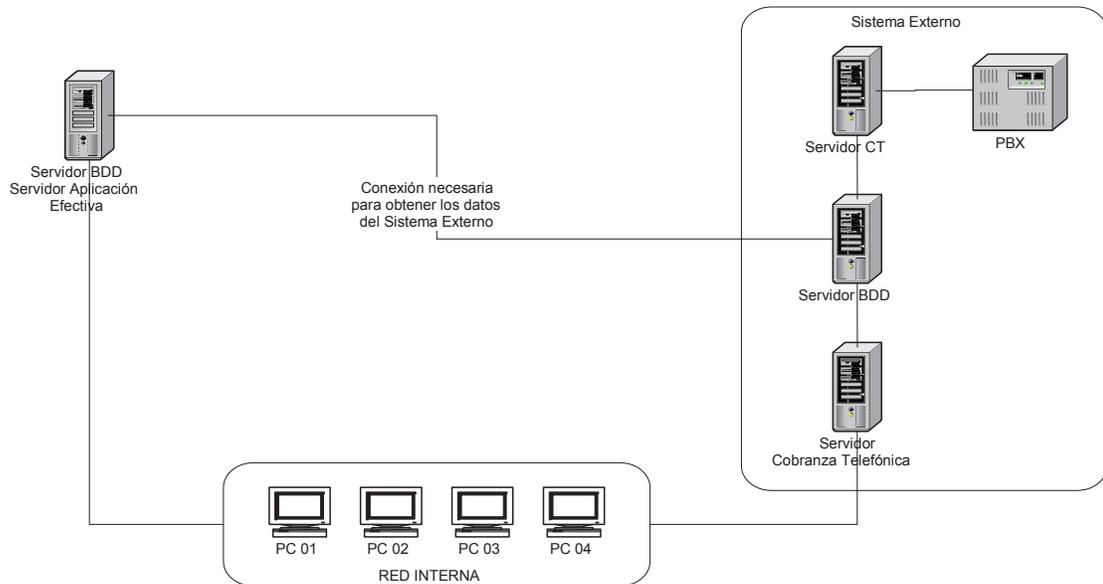


Figura 2.3 **Arquitectura**

Ref.: Autor

Se indica además en la imagen cuál es la arquitectura de equipos que tiene el sistema externo “Gestión Telefónica” del que se toman los datos.

2.3.4. ESTRATEGIA PARA ALMACENAMIENTO DE DATOS

Se sacará respaldos diarios nocturnos de la base de datos, este será un proceso nocturno que empezará a partir de las 10:00 p.m. hora en la que no existen ya usuarios conectados al sistema.

Por ser una aplicación que interactúa solamente con bases de datos, no lo hace con archivos, se han cerrado los puertos del servicio ftp, pues se quiere evitar un posible hueco de seguridad.

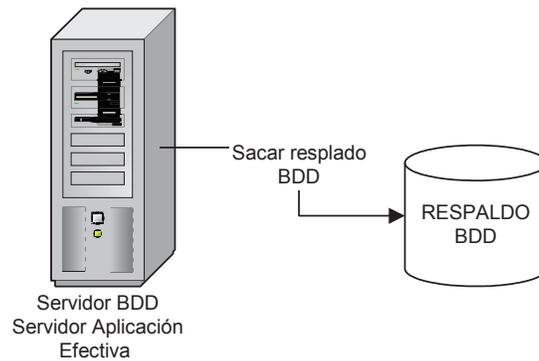


Figura 2.4 Esquema de Respaldo de Datos

Ref.: Autor

La cuenta sa, no será utilizada por el prototipo por razones de seguridad, será necesario, crearse una cuenta propia para el sistema, la misma que no tendrá privilegios de creación, modificación o eliminación de tablas, simplemente podrá realizar cualquier acción sobre los datos.

Habrà un mantenimiento trimestral de la base de datos, en el cual se espera mejorar el rendimiento de la base de datos con la creación de nuevos índices de ser así necesario o el paso de información histórica a otra estructura de datos.

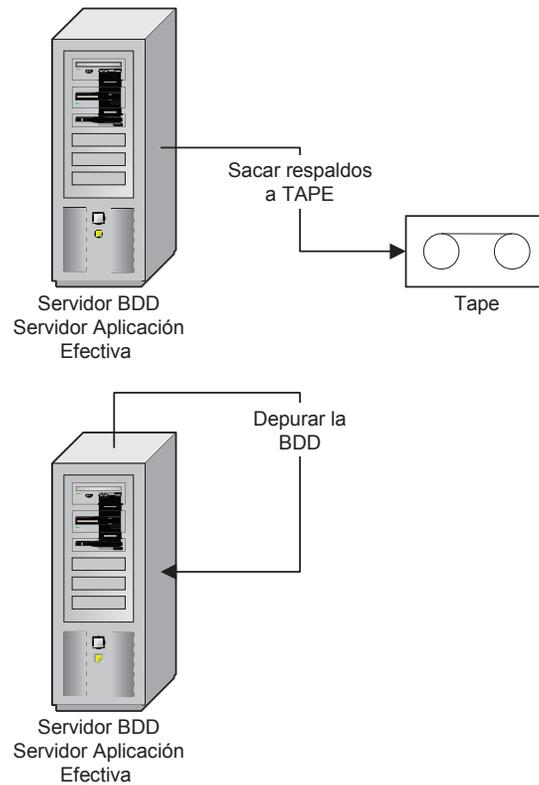


Figura 2.5 Esquema de Mantenimiento de BDD

Ref.: Autor

2.3.5. MANEJO DE RECURSOS GLOBALES

Se considera como único recurso global al servidor de aplicaciones, este servidor se encuentra detrás de un firewall, ya que solamente podrá ser usado por los computadores que forman parte de la red.

El servidor de aplicación no tendrá carpetas compartidas. Tampoco será el servidor de autenticación.

La autenticación al sistema estará a cargo del prototipo y se utilizará el método de autenticación de formularios, un método seguro que provee el framework de .NET.

2.3.6. CONTROL DEL SOFTWARE

- CONTROL INTERNO

Para el control interno del prototipo se realizarán pruebas con datos extremos, es decir donde se pueda producir un error de programación.

- CONTROL EXTERNO

Para esto se realizarán tres tipos de pruebas:

- El primer tipo se basa en una validación de datos, es decir demostrar que el usuario no puede ingresar datos que son inválidos, por ejemplo ingresa letras en la calificación, la misma que es de tipo numérico. Estas pruebas se las realizarán por cada interfaz.
- El segundo tipo de pruebas tendrá que demostrar que los datos que envía el cliente, en este caso el explorador de Internet, llegan perfectamente hasta el servidor.
- El tercer tipo de pruebas serán orientadas a demostrar que el comportamiento del sistema es el adecuado, con los mismos datos de prueba del control interno, se verificará el buen funcionamiento del o los procesos que realiza el sistema, los resultados deberán ser los esperados, ahora no por algoritmo sino por proceso completo.

- CONTROL DEL ENTORNO

El entorno lastimosamente en este tipo de sistemas no es muy controlable, la razón es fundamentalmente la falta de conectividad que puede existir en un momento dado ya sea por parte de nuestro servidor como del cliente.

Hay que recordar que se depende de la red Internet, en esta red puede existir falla por parte de:

- Servidor de acceso a internet que utiliza el servidor.
- Servidor de acceso a internet que utilizan los clientes, pueden ser varios.

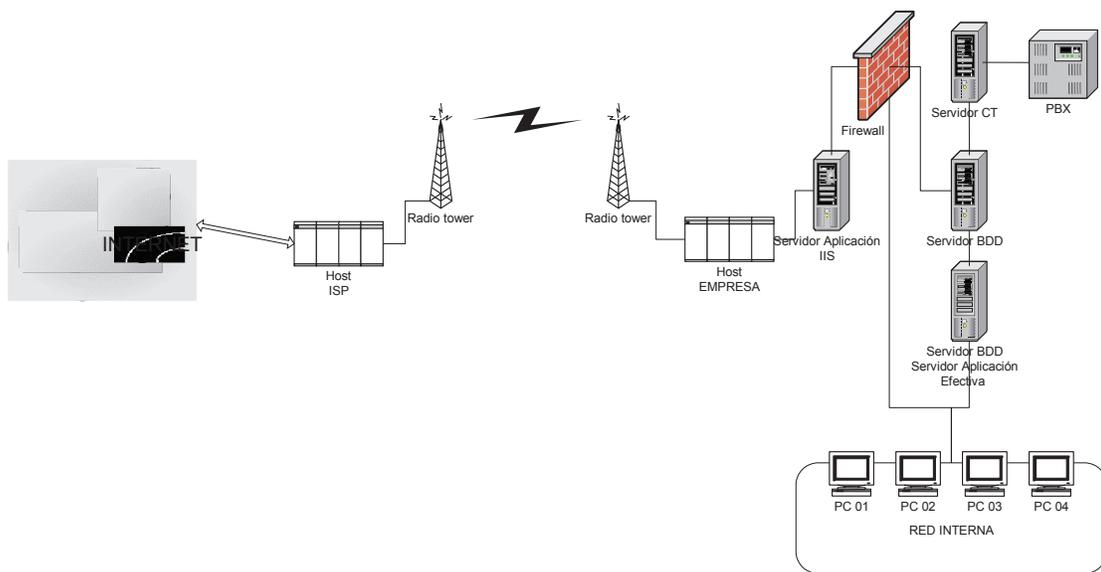


Figura 2.6 Entorno del sistema

Ref.: Autor

Por otro lado existe la posibilidad que nuestro servidor sea el que falle, por diversos motivos, falta de energía eléctrica o una mala operación, para controlar esta posible falla se deberá tomar como medida uno la adquisición de un UPS de 30 minutos de alimentación, tiempo que se considera prudencial para poder realizar cualquier tarea en caso de que no se restablezca el alumbrado. Como segunda medida el servidor ha sido alojado en RACKS que están cerrados y fuera del alcance de

cualquier usuario, solamente el administrador de servidores tiene acceso al mismo, minimizando la posibilidad de una mala utilización del mismo.

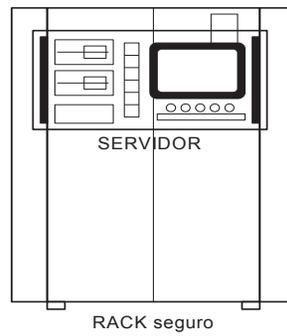
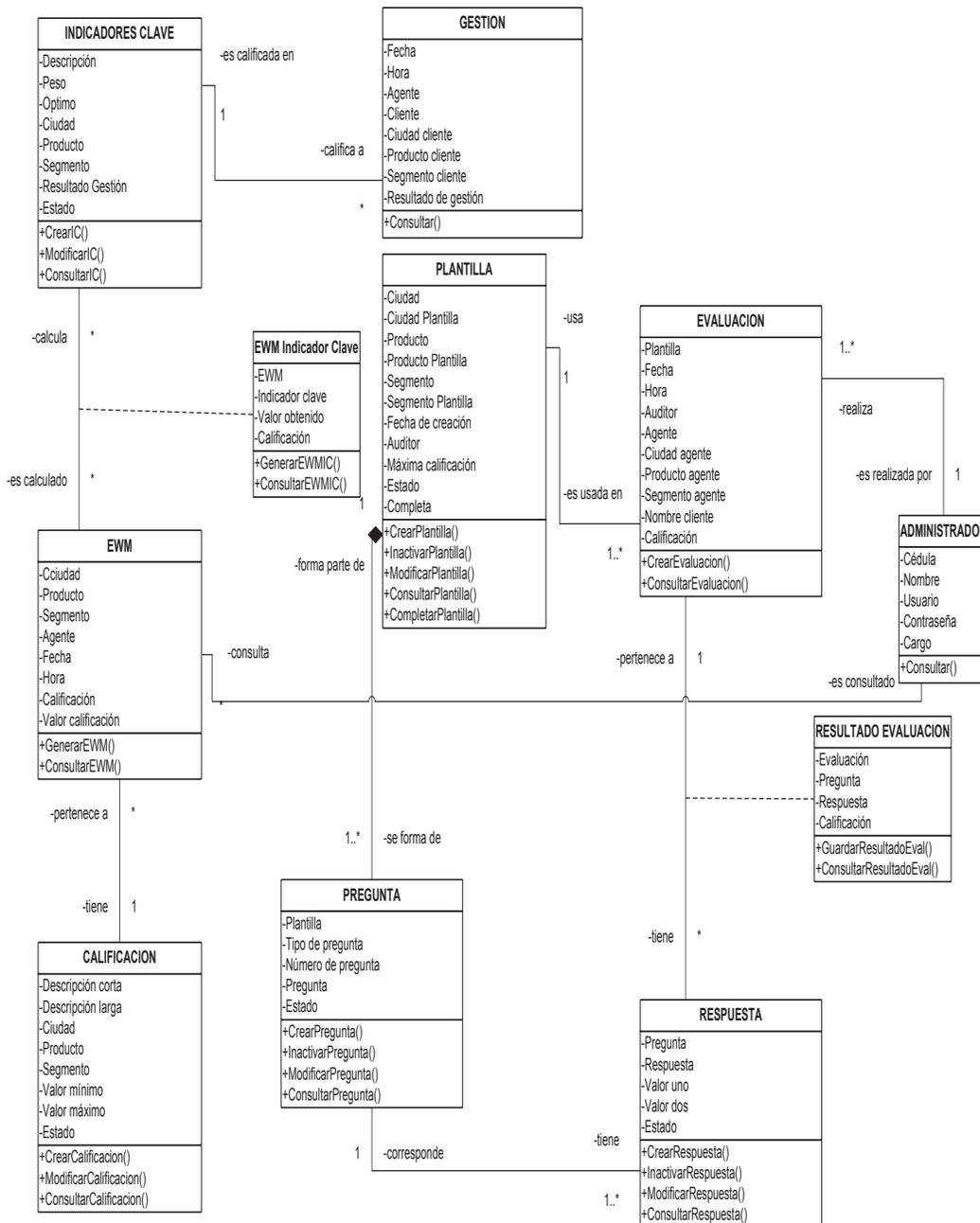


Figura 2.7 Rack de Servidor

Ref.: Autor

2.4. DISEÑO DE OBJETOS

2.4.1. DIAGRAMA FINAL DEL MODELO DE OBJETOS



Ref.: Autor

2.4.2. DISEÑO DE LA BASE DE DATOS

Cada uno de los elementos del modelo de clases se tiene que transformar en un elemento del modelo relacional, el objetivo de transformar es obtener un modelo físico para lo cual es necesario aplicar un conjunto de reglas de transformación.

2.4.2.1. Reglas de Transformación

- Transformación de Clases

Una clase se transforma en una o más tablas. Es posible que 2 clases se transformen en una sola tabla cuando el comportamiento de una de ellas sea irrelevante en la BDD.

- Transformación de atributos de clases

Cada atributo se transforma en una columna de la tabla en la que se transformó la clase a la que pertenece. El identificador único se convierte en clave primaria. Si existen restricciones asociadas a los atributos estas pueden recogerse con algunas cláusulas del lenguaje lógico que se convertirán en disparadores cuando estos sean soportados por el producto.

- Transformación de relaciones

- Relaciones M:N

Se transforman en una tabla, cuya clave primaria es la concatenación de los identificadores de las clases asociadas, siendo cada uno de ellos clave extranjera de la propia tabla. Si la relación tiene atributos, estos se transforman en columnas de la tabla.

- Relaciones 1:N

- Propagar el identificador de la clase de cardinalidad máxima 1 a la que es N, teniendo en cuenta que:

- Si la relación es de asociación la clave propagada es clave extranjera en la tabla a la que se ha propagado.
 - Si la relación es de dependencia, la clave primaria de la tabla correspondiente a la clase débil está formada por la concatenación de los identificadores de ambas clases.

- Si la relación tiene atributos propios y queremos que aparezcan como tales, o cuando se piensa que en un futuro se pueda convertir en N:M, o en el caso de que el número de ocurrencias relacionadas de la clase que propaga su clave sea muy pequeño (y por tanto pueden existir muchos valores nulos), se transforma en una tabla con clave primaria sólo el identificador de la clase de cardinalidad máxima N. Al igual que en el caso N:M, las claves propagadas son claves extranjeras de la nueva tabla creada.

- Relaciones 1:1

Es un caso particular de las 1:N y se puede tanto crear una tabla o propagar la clave, si bien en este último caso, la clave se propaga en las dos direcciones. Para decidir qué solución adoptar se debe analizar la situación intentando recoger la mayor semántica posible y evitar valores nulos. Las relaciones de agregación se transforman del mismo modo que las 1:N.

- Transformación de relaciones exclusivas

Después de haber realizado la transformación según las relaciones 1:N, se debe tener en cuenta que si se han propagado los atributos de las clases, convirtiéndose en claves extranjeras de la tabla que provenía de la clase común a las relaciones, hay que comprobar que una y solo una de esas claves es nula en cada ocurrencia. En caso de no propagarse las claves, estas comprobaciones se deben hacer en las tablas resultantes de transformar las relaciones.

2.4.2.2. Modelo Físico de la Base de Datos

Ver Anexo 1.

2.4.3. DISEÑO DE ALGORITMOS

Al ser elegida una programación n-capas, los algoritmos necesarios para la implementación del sistema estarán distribuidos en las distintas capas según su aplicación.

Todo cálculo se lo realizará única y exclusivamente en la capa de lógica del negocio pura, así como el acceso a los datos siempre pasará por el componente encargado de la conexión y acceso a datos.

2.4.3.1. Algoritmos para la creación de operaciones

- **CLASE PLANTILLA**

- **Operación Crear Plantilla**

```

CREAR PROCEDIMIENTO ALMACENADO pa_ingresa_plantilla
(
  [Ciudad] [varchar] (100),
  [CiudadPlantilla] [varchar] (100),
  [Producto] [varchar] (200),
  [ProductoPlantilla] [varchar] (100),
  [Segmento] [varchar] (100),
  [SegmentoPlantilla] [varchar] (100),

```

```

    [FechaCreación] [date],
    [IdAuditor] [int],
    [PuntajeSobre] [int],
    [Estado] [int],
    [Completa] [char]
)
AS

```

```

INSERT INTO PLANTILLA VALUES (
    [Ciudad],
    [CiudadPlantilla],
    [Producto],
    [ProductoPlantilla],
    [Segmento],
    [SegmentoPlantilla],
    [FechaCreación],
    [IdAuditor],
    [PuntajeSobre],
    [Estado],
    [Completa]
)

```

- **Operación Modificar Plantilla**

```

CREAR PROCEDIMIENTO ALMACENADO pa_modifica_plantilla
(
    [IdPlantilla] [int],
    [Ciudad] [varchar] (100),
    [CiudadPlantilla] [varchar] (100),
    [Producto] [varchar] (200),
    [ProductoPlantilla] [varchar] (100),
    [Segmento] [varchar] (100),
    [SegmentoPlantilla] [varchar] (100),
    [FechaCreación] [date],
    [IdAuditor] [int],
    [PuntajeSobre] [int],
    [Estado] [int],
    [Completa] [char]
)
AS

```

```

ACTUALIZAR PLANTILLA VALUES (
    [Ciudad],
    [CiudadPlantilla],
    [Producto],
    [ProductoPlantilla],

```

```

    [Segmento],
    [SegmentoPlantilla],
    [FechaCreación],
    [IdAuditor],
    [PuntajeSobre],
    [Estado],
    [Completa]
)
DONDE
    [IdPlantilla] = [IdPlantilla]

```

- **Operación Consultar Plantilla**

```

CREAR PROCEDIMIENTO ALMACENADO pa_consulta_plantilla
AS
    SELECCIONAR
        TODO
    FROM
        PLANTILLA
    DONDE
        [IdPlantilla] = [IdPlantilla]

```

- **Operación Inactivar Plantilla**

```

CREAR PROCEDIMIENTO ALMACENADO pa_inactiva_plantilla
(
    [IdPlantilla] [int],
    [Estado] [int]
)
AS

```

```

ACTUALIZAR PLANTILLA VALUES (
    [Estado]
)
DONDE
    [IdPlantilla] = [IdPlantilla]

```

- **Operación Completar Plantilla**

```

CREAR PROCEDIMIENTO ALMACENADO pa_completa_plantilla
(
    [IdPlantilla] [int],
    [Estado] [int]
)

```

```

)
AS

ACTUALIZAR PLANTILLA VALUES (
    [Estado]
)
DONDE
    [IdPlantilla] = [IdPlantilla]

```

- **CLASE EVALUACIÓN**

- **Operación Crear Evaluación**

```

CREAR PROCEDIMIENTO ALMACENADO pa_ingresa_evaluacion
(
    [IdPlantilla] [int],
    [Fecha] [date],
    [Hora] [char] (5),
    [idAuditor] [int],
    [idAgente] [int],
        [idCiudad] [int],
        [idProducto] [int],
        [idSegmento] [int],
    [NomCliente] [varchar] (30),
    [Calificacion] [int],
    [Completa] [char]
)
AS

INSERT INTO PLANTILLA VALUES (
    [IdPlantilla],
    [Fecha],
    [Hora],
    [idAuditor],
    [idAgente],
        [idCiudad],
        [idProducto],
        [idSegmento],
    [NomCliente],
    [Calificación],
    [Completa]
)

```

- **Operación Consultar Evaluación**

```

CREAR PROCEDIMIENTO ALMACENADO pa_consulta_evaluacion
AS
    SELECCIONAR
        TODO
    FROM
        EVALUACION
    DONDE
        [IdEvaluacion] = [IdEvaluacion]

```

- **Operación Completar Evaluación**

```

CREAR PROCEDIMIENTO ALMACENADO pa_completa_evaluacion
(
    [IdEvaluacion] [int],
    [Estado] [int]
)
AS

ACTUALIZAR EVALUACION VALUES (
    [Estado]
)
DONDE
    [IdEvaluacion] = [IdEvaluacion]

```

- **CLASE RESULTADO EVALUACIÓN**

- **Operación Guardar Resultado Eval**

```

CREAR PROCEDIMIENTO ALMACENADO
pa_ingresa_resultadoevaluacion
(
    [idEvaluacion] [int],
    [idPregunta] [int],
    [idRespuesta] [int],
    [puntaje] [decimal]
)
AS

INSERT INTO RESULTADOEVALUACION VALUES (
    [idEvaluacion],
    [idPregunta],
    [idRespuesta],
    [puntaje]
)

```

- **Operación Consultar Resultado Eval**

```

CREAR PROCEDIMIENTO ALMACENADO
pa_consulta_resultadoevaluacion
AS
    SELECCIONAR
        TODO
    FROM
        RESULTADOEVALUACION
    DONDE
        [IdResultadoEvaluacion] = [IdResultadoEvaluacion]

```

- **CLASE TIPO PREGUNTA**

- **Operación Crear Tipo Pregunta**

```

CREAR PROCEDIMIENTO ALMACENADO
pa_ingresa_TipoPregunta
(
    [Descripción] [varchar] (100),
    [Estado] [int]
)
AS

```

```

INSERT INTO TIPOPREGUNTA VALUES (
    [Descripción],
    [Estado]
)

```

- **Operación Modificar Tipo Pregunta**

```

CREAR PROCEDIMIENTO ALMACENADO
pa_actualiza_TipoPregunta
(
    [idTipoPregunta] [int],
    [Descripción] [varchar] (100),
    [Estado] [int]
)
AS

```

```

ACTUALIZAR TIPOPREGUNTA VALUES (
    [Descripción] [varchar] (100),
    [Estado] [int]
)
DONDE
    [idTipoPregunta] = [idTipoPregunta]

```

- **Operación Consultar Tipo Pregunta**

```

CREAR PROCEDIMIENTO ALMACENADO pa_consulta_
TipoPregunta
AS
    SELECCIONAR
        TODO
    FROM
        TIPOPREGUNTA
    DONDE
        [idTipoPregunta] = [idTipoPregunta]

```

- **CLASE PREGUNTA**

- **Operación Crear Pregunta**

```

CREAR PROCEDIMIENTO ALMACENADO pa_ingresa_Pregunta
(
    [idPlantilla] [int],
    [idtipoPregunta] [int],
    [Pregunta] [varchar] (100),
    [Estado] [int]
)
AS

INSERT INTO PREGUNTA VALUES (
    [idPlantilla],
    [idtipoPregunta],
    [Pregunta],
    [Estado]
)

```

- **Operación Inactivar Pregunta**

```

CREAR PROCEDIMIENTO ALMACENADO pa_inactiva_pregunta
(
    [idPregunta] [int],
    [Estado] [int]
)
AS

ACTUALIZAR PREGUNTA VALUES (
    [Estado]
)
DONDE
    [idPregunta] = [idPregunta]

```

- **Operación Modificar Pregunta**

```

CREAR PROCEDIMIENTO ALMACENADO pa_actualiza_pregunta
(
    [idPregunta] [int],
    [idPlantilla] [int],
    [idtipoPregunta] [int],
    [Pregunta] [varchar] (100),
    [Estado] [int]
)
AS

ACTUALIZAR PREGUNTA VALUES (
    [idPlantilla],
    [idtipoPregunta],
    [Pregunta],
    [Estado]
)
DONDE
    [idPregunta] = [idPregunta]
  
```

- **Operación Consultar Pregunta**

```

CREAR PROCEDIMIENTO ALMACENADO pa_consulta_Pregunta
AS
    SELECCIONAR
        TODO
    FROM
        PREGUNTA
    DONDE
        [idPregunta] = [idPregunta]
  
```

- **CLASE RESPUESTA**

- **Operación Crear Respuesta**

```

CREAR PROCEDIMIENTO ALMACENADO pa_ingresa_respuesta
(
    [idPregunta] [int],
    [Respuesta] [varchar] (20),
    [valoruno] [int],
    [valordor] [int],
    [Estado] [int]
  
```

```

)
AS

INSERT INTO RESPUESTA VALUES (
    [idPregunta],
    [Respuesta],
    [valoruno],
    [valordos],
    [Estado]
)

```

- **Operación Inactivar Respuesta**

```

CREAR PROCEDIMIENTO ALMACENADO pa_inactiva_respuesta
(
    [idRespuesta] [int],
    [Estado] [int]
)
AS

ACTUALIZAR RESPUESTA VALUES (
    [Estado]
)
DONDE
    [idRespuesta] = [idRespuesta]

```

- **Operación Modificar Respuesta**

```

CREAR PROCEDIMIENTO ALMACENADO pa_actualiza_respuesta
(
    [idRespuesta] [int],
    [idPregunta] [int],
    [Respuesta] [varchar] (20),
    [valoruno] [int],
    [valordor] [int],
    [Estado] [int]
)
AS

ACTUALIZAR RESPUESTA VALUES (
    [idPlantilla],
    [idtipoPregunta],
    [Pregunta],
    [Estado]
)
DONDE
    [idRespuesta] = [idRespuesta]

```

- **Operación Consultar Respuesta**

```

CREAR PROCEDIMIENTO ALMACENADO pa_consulta_respuesta
AS
    SELECCIONAR
        TODO
    FROM
        RESPUESTA
    DONDE
        [idRespuesta] = [idRespuesta]

```

- **CLASE INDICADORES CLAVE**

- **Operación Crear Indicador Clave**

```

CREAR PROCEDIMIENTO ALMACENADO
pa_ingresa_IndicadoresClave
(
    [idIndicadoresClave] [int],
    [Descripción] [varchar] (100),
    [Peso] [int],
    [Optimo] [int],
    [idCiudad] [int],
    [idProducto] [int],
    [idSegmento] [int],
    [Estado] [int]
)
AS

INSERT INTO INDICADORESCLAVE VALUES (
    [idIndicadoresClave],
    [Descripción],
    [Peso],
    [Optimo],
    [idCiudad],
    [idProducto],
    [idSegmento],
    [Estado]
)

```

- **Operación Inactivar Indicadores Clave**

```

CREAR PROCEDIMIENTO ALMACENADO
pa_inactiva_IndicadoresClave
(
    [idPregunta] [int],
    [Estado] [int]
)
AS

```

```

ACTUALIZAR INDICADORESCLAVE VALUES (
    [Estado]
)
DONDE
    [idPregunta] = [idPregunta]

```

- **Operación Modicar Indicador Clave**

```

CREAR PROCEDIMIENTO ALMACENADO
pa_actualiza_IndicadoresClave
(
    [idIndicadoresClave] [int],
    [Descripción] [varchar] (100),
    [Peso] [int],
    [Optimo] [int],
    [idCiudad] [int],
    [idProducto] [int],
    [idSegmento] [int],
    [Estado] [int]
)
AS

```

```

ACTUALIZAR IndicadoresClave VALUES (
    [Descripción],
    [Peso],
    [Optimo],
    [idCiudad],
    [idProducto],
    [idSegmento],
    [Estado]
)
DONDE
    [idIndicadoresClave] = [idIndicadoresClave]

```

- **Operación Consultar IndicadoresClave**

```

CREAR PROCEDIMIENTO ALMACENADO
pa_consulta_IndicadoresClave
AS
    SELECCIONAR
        TODO
    FROM
        IndicadoresClave
    DONDE
        [idIndicadoresClave] = [idIndicadoresClave]

```

- **CLASE EWM**

- **Operación Generar EWM**

```

CREAR PROCEDIMIENTO ALMACENADO pa_ingresa_EWM
(
    [idCiudad] [int],
    [idProducto] [int],
    [idSegmento] [int],
    [idAgente] [int],
    [Fecha] [date],
    [Hora] [char] (5),
    [Calificación] [char] (1),
    [puntaje] [decimal]
)
AS

INSERT INTO EWM VALUES (
    [idCiudad],
    [idProducto],
    [idSegmento],
    [idAgente],
    [Fecha],
    [Hora],
    [puntaje]
)

```

- **Operación Consultar EWM**

```

CREAR PROCEDIMIENTO ALMACENADO pa_consulta_EWM
AS
    SELECCIONAR
        TODO
    FROM
        EWM
    DONDE
        [idEWM] = [idEWM]

```

- **CLASE EWM IC**

- **Operación Generar EWM IC**

```

CREAR PROCEDIMIENTO ALMACENADO pa_ingresa_EWMIC
(
    [idEWM] [int],
    [idIndicadoresClave] [int],
    [valorobtenido] [decimal],
    [puntaje] [decimal]
)
AS

```

```

INSERT INTO EWMIC VALUES (
    [idEWM],
    [idIndicadoresClave],
    [valorobtenido],
    [puntaje]
)

```

- **Operación Consultar EWM IC**

```

CREAR PROCEDIMIENTO ALMACENADO pa_consulta_EWMIC
AS
    SELECCIONAR
        TODO
    FROM
        EWMIC
    DONDE
        [idEWMIC] = [idEWMIC]

```

- **CLASE CALIFICACIÓN**

- **Operación Crear Calificación**

```

CREAR PROCEDIMIENTO ALMACENADO pa_ingresa_Calificacion
(
  [DescripcionCorta] [varchar] (100),
  [DescripcionLarga] [varchar] (100),
  [idCiudad] [int],
  [idProducto] [int],
  [idSegmento] [int],
  [Valorminimo] [decimal],
  [ValorMaximo] [decimal],
  [Estado] [int]
)
AS

```

```

INSERT INTO Calificacion VALUES (
  [DescripcionCorta],
  [DescripcionLarga],
  [idCiudad],
  [idProducto],
  [idSegmento],
  [Valorminimo],
  [ValorMaximo],
  [Estado] [int]
)

```

- **Operación Inactivar Calificación**

```

CREAR PROCEDIMIENTO ALMACENADO pa_inactiva_Calificacion
(
  [idCalificacion] [int],
  [Estado] [int]
)
AS

```

```

ACTUALIZAR CALIFICACION VALUES (
  [Estado]
)
DONDE
  [idCalificacion] = [idCalificacion]

```

- **Operación Modificar Calificación**

```

CREAR PROCEDIMIENTO ALMACENADO
pa_actualiza_Calificacion
(
    [idCalificacion] [int],
    [DescripcionCorta] [varchar] (100),
    [DescripcionLarga] [varchar] (100),
    [idCiudad] [int],
    [idProducto] [int],
    [idSegmento] [int],
    [Valorminimo] [decimal],
    [ValorMaximo] [decimal],
    [Estado] [int]
)
AS

```

```

ACTUALIZAR Calificacion VALUES (
    [DescripcionCorta],
    [DescripcionLarga],
    [idCiudad],
    [idProducto],
    [idSegmento],
    [Valorminimo],
    [ValorMaximo],
    [Estado] [int]
)
DONDE
    [idCalificacion] = [idCalificacion]

```

- **Operación Consultar Calificación**

```

CREAR PROCEDIMIENTO ALMACENADO pa_consulta_Calificacion
AS
    SELECCIONAR
        TODO
    FROM
        CALIFICACION
    DONDE
        [idCalificacion] = [idCalificacion]

```

2.4.3.2. Algoritmos para la creación de tablas

- **CLASE PLANTILLA**

```
CREAR TABLA [Plantilla] (  
    [IdPlantilla] [int],  
    [Ciudad] [varchar] (100),  
    [CiudadPlantilla] [varchar] (100),  
    [Producto] [varchar] (200),  
    [ProductoPlantilla] [varchar] (100),  
    [Segmento] [varchar] (100),  
    [SegmentoPlantilla] [varchar] (100),  
    [FechaCreación] [date],  
    [IdAuditor] [int],  
    [PuntajeSobre] [decimal],  
    [Estado] [int],  
    [Completa] [char]  
)
```

- **CLASE EVALUACIÓN**

```
CREAR TABLA [Evaluacion] (  
    [idEvaluacion] [int],  
    [IdPlantilla] [int],  
    [Fecha] [date],  
    [Hora] [char] (5),  
    [idAuditor] [int],  
    [idAgente] [int],  
    [idCiudad] [int],  
    [idProducto] [int],  
    [idSegmento] [int],  
    [NomCliente] [varchar] (30),  
    [Calificacion] [int],  
    [Completa] [char]  
)
```

- **CLASE RESULTADO EVALUACIÓN**

```
CREAR TABLA [ResultadoEvaluación] (  
    [id ResultadoEvaluación] [int],  
    [idEvaluacion] [int],  
    [idPregunta] [int],  
    [idRespuesta] [int],
```

```

        [puntaje] [decimal]
    )

```

- **CLASE TIPO PREGUNTA**

```

CREAR TABLA [TipoPregunta] (
    [idTipoPregunta] [int],
    [Descripción] [varchar] (100),
    [Estado] [int]
)

```

- **CLASE PREGUNTA**

```

CREAR TABLA [Pregunta] (
    [idPregunta] [int],
    [idPlantilla] [int],
    [idtipoPregunta] [int],
    [Pregunta] [varchar] (100),
    [Estado] [int]
)

```

- **CLASE RESPUESTA**

```

CREAR TABLA [Respuesta] (
    [idRespuesta] [int],
    [idPregunta] [int],
    [Respuesta] [varchar] (20),
    [valoruno] [int],
    [valordor] [int],
    [Estado] [int]
)

```

- **CLASE INDICADORES CLAVE**

```

CREAR TABLA [IndicadoresClave] (
    [idIndicadoresClave] [int],
    [Descripción] [varchar] (100),
    [Peso] [int],
    [Optimo] [int],
    [idCiudad] [int],
    [idProducto] [int],
    [idSegmento] [int],
    [Estado] [int]
)

```

- **CLASE EWM**

```
CREAR TABLA [EWM] (  
    [idEWM] [int],  
    [idCiudad] [int],  
    [idProducto] [int],  
    [idSegmento] [int],  
    [idAgente] [int],  
    [Fecha] [date],  
    [Hora] [char] (5),  
    [Calificación] [char] (1),  
    [puntaje] [decimal]  
)
```

- **CLASE EWM IC**

```
CREAR TABLA [EWMIC] (  
    [idEWM] [int],  
    [idIndicadoresClave] [int],  
    [valorobtenido] [decimal],  
    [puntaje] [decimal]  
)
```

- **CLASE CALIFICACIÓN**

```
CREAR TABLA [CALIFICACION] (  
    [idCalificacion] [int],  
    [DescripcionCorta] [varchar] (100),  
    [DescripcionLarga] [varchar] (100),  
    [idCiudad] [int],  
    [idProducto] [int],  
    [idSegmento] [int],  
    [Valorminimo] [decimal],  
    [ValorMaximo] [decimal],  
    [Estado] [int]  
)
```

2.4.3.3. Estructuras capa lógica del negocio

- **CLASE PLANTILLA**

```
CLASS Plantilla {  
    int IdPlantilla;  
    string Ciudad;  
    string CiudadPlantilla ;  
    string Producto;  
    string ProductoPlantilla ;  
    string Segmento ;  
    string SegmentoPlantilla ;  
    date FechaCreación;  
    int IdAuditor;  
    decimal PuntajeSobre;  
    int Estado;  
    string Completa;  
}
```

- **CLASE EVALUACIÓN**

```
CLASS Evaluación (  
    int idEvaluación;  
    int IdPlantilla;  
    date Fecha;  
    string Hora;  
    int idAuditor;  
    int idAgente;  
    int idCiudad;  
    int idProducto;  
    int idSegmento;  
    string NomCliente;  
    int Calificación;  
    string Completa;  
)
```

- **CLASE RESULTADO EVALUACIÓN**

```
CLASS ResultadoEvaluación (  
    int idResultadoEvaluación;  
    int idEvaluacion;  
    int idPregunta;  
    int idRespuesta;  
    decimal puntaje;  
)
```

- **CLASE TIPO PREGUNTA**

```
CLASS TipoPregunta (  
    int idTipoPregunta;  
    string Descripción;  
    int Estado;  
)
```

- **CLASE PREGUNTA**

```
CLASS Pregunta (  
    int idPregunta;  
    int idPlantilla;  
    int idtipoPregunta;  
    string Pregunta ;  
    int Estado;  
)
```

- **CLASE RESPUESTA**

```
CLASS Respuesta (  
    int idRespuesta;  
    int idPregunta;  
    string Respuesta;  
    int valoruno;  
    int valordor;  
    int Estado;  
)
```

- **CLASE INDICADOR CLAVE**

```
CLASS IndicadoresClave (  
    int idIndicadoresClave;  
    string Descripción;  
    int Peso;  
    int Optimo;  
    int idCiudad;  
    int idProducto;  
    int idSegmento;  
    int Estado;  
)
```

- **CLASE EWM**

```
CLASS EWM (  
    int idEWM;  
    int idCiudad;  
    int idProducto;  
    int idSegmento;  
    int idAgente;  
    date Fecha;  
    string Hora;  
    puntaje decimal;  
)
```

- **CLASE EWMIC**

```
CLASS EWMIC (  
    int idEWM;  
    int idIndicadoresClave;  
    int valorobtenido;  
    decimal puntaje;  
)
```

- **CLASE CALIFICACIÓN**

```
CLASS CALIFICACION (  
    int idCalificacion;  
    string DescripcionCorta ;  
    string DescripcionLarga ;  
    int idCiudad;  
    int idProducto;  
    int idSegmento;  
    int Valorminimo;  
    Int ValorMaximo;  
    int Estado;  
)
```

2.4.3.4. Algoritmos capa lógica del negocio

- **CLASE PLANTILLA**

- **Operación Crear Plantilla**

```
void crear_plantilla {  
    se llenan los atributos de la clase;  
    se ejecuta pa_ingresa_plantilla;  
}
```

- **Operación Modificar Plantilla**

```
void modificar_plantilla {  
    se llenan los atributos de la clase;  
    se ejecuta pa_modifica_plantilla;  
}
```

- **Operación Consultar Plantilla**

```
Dataview consultar_plantilla {  
    se llenan los atributos de la clase;  
    se ejecuta pa_consulta_plantilla;  
    se devuelve vista con datos;  
}
```

- **Operación Inactivar Plantilla**

```
void modificar_plantilla {  
    se llenan los atributos de la clase;  
    se ejecuta pa_inactiva_plantilla;  
}
```

- **Operación Completar Plantilla**

```
void completar_plantilla {  
    se configura la plantilla;  
    se llenan los atributos de la clase;  
    se ejecuta pa_inactiva_plantilla;  
}
```

- **CLASE EVALUACIÓN**

- **Operación Crear Evaluación**

```
void crear_evaluación {  
    se llenan los atributos de la clase;  
    se ejecuta pa_ingresa_evaluacion;  
}
```

- **Operación Consultar Evaluación**

```
Dataview consultar_evaluación {  
    se llenan los atributos de la clase;  
    se ejecuta pa_consulta_evaluación;  
    se devuelve vista con datos;  
}
```

- **Operación Completar Evaluación**

```
void completar_evaluación {  
    se suman los resultados parciales de la clase  
    RESULTADOEVALUACION por evaluacion;  
    se llenan los atributos de la clase;  
    se ejecuta pa_completa_evaluacion;  
}
```

- **CLASE RESULTADO EVALUACIÓN**

- **Operación Guardar Resultado Eval**

```
void guardar_resultado_eval {  
    se llenan los atributos de la clase;  
    se ejecuta pa_ingresa_resultadoevaluacion;  
}
```

- **Operación Consultar Resultado Eval**

```
Dataview consultar_resultado_eval {  
    se llenan los atributos de la clase;  
    se ejecuta pa_consulta_resultadoevaluacion;  
    se devuelve vista con datos;  
}
```

- **CLASE TIPO PREGUNTA**

- **Operación Crear Tipo Pregunta**

```
void crear_tipopregunta {  
    se llenan los atributos de la clase;  
    se ejecuta pa_ingresa_TipoPregunta;  
}
```

- **Operación Modificar Tipo Pregunta**

```
void modificar_tipopregunta {  
    se llenan los atributos de la clase;  
    se ejecuta pa_actualiza_TipoPregunta;  
}
```

- **Operación Consultar Tipo Pregunta**

```
Dataview consultar_tipopregunta {  
    se llenan los atributos de la clase;  
    se ejecuta pa_consulta_TipoPregunta;  
    se devuelve vista con datos;  
}
```

- **CLASE PREGUNTA**

- **Operación Crear Pregunta**

```
void crear_pregunta {  
    se llenan los atributos de la clase;  
    se ejecuta pa_ingresa_Pregunta;  
}
```

- **Operación Inactivar Pregunta**

```
void inactivar_pregunta {  
    se llenan los atributos de la clase;  
    se ejecuta pa_inactiva_pregunta;  
}
```

- **Operación Modificar Pregunta**

```
void actualizar_pregunta {  
    se llenan los atributos de la clase;  
    se ejecuta pa_actualiza_pregunta;  
}
```

- **Operación Consultar Pregunta**

```
Dataview consultar_pregunta {  
    se llenan los atributos de la clase;  
    se ejecuta pa_consulta_Pregunta;  
    se devuelve vista con datos;  
}
```

- **CLASE RESPUESTA**

- **Operación Crear Respuesta**

```
void crear_respuesta {  
    se llenan los atributos de la clase;  
    se ejecuta pa_ingresa_respuesta;  
}
```

- **Operación Inactivar Respuesta**

```
void inactivar_respuesta {  
    se llenan los atributos de la clase;  
    se ejecuta pa_inactiva_respuesta;  
}
```

- **Operación Modificar Respuesta**

```
void modificar_respuesta {  
    se llenan los atributos de la clase;  
    se ejecuta pa_modifica_respuesta;  
}
```

- **Operación Consultar Respuesta**

```
Dataview consultar_respuesta {
```

```
    se llenan los atributos de la clase;  
    se ejecuta pa_consulta_respuesta;  
    se devuelve vista con datos;  
}
```

- **CLASE INDICADOR CLAVE**

- **Operación Crear Indicador Clave**

```
void crear_indicadoresclave {  
    se llenan los atributos de la clase;  
    se ejecuta pa_ingresa_IndicadoresClave;  
}
```

- **Operación Inactivar Indicador Clave**

```
void inactivar_indicadoresclave {  
    se llenan los atributos de la clase;  
    se ejecuta pa_inactiva_IndicadoresClave;  
}
```

- **Operación Modificar Indicador Clave**

```
void modificar_indicadoresclave {  
    se llenan los atributos de la clase;  
    se ejecuta pa_actualiza_IndicadoresClave;  
}
```

- **Operación Consultar Indicador Clave**

```
Dataview consultar_IndicadoresClave {  
    se llenan los atributos de la clase;  
    se ejecuta pa_consulta_IndicadoresClave;  
    se devuelve vista con datos;  
}
```

- **CLASE EWM**

- **Operación Generar EWM**

```
void generar_ewm {  
    se suma gestiones de la clase GESTION del SUBSISTEMA  
    GESTION TELEFONICA  
    se llenan los atributos de la clase;  
    se ejecuta pa_ingresa_EWM;  
}
```

- **Operación Consultar EWM**

```
Dataview consultar_ewm {  
    se llenan los atributos de la clase;  
    se ejecuta pa_consulta_EWM;  
    se devuelve vista con datos;  
}
```

- **CLASE EWMIC**

- **Operación Generar EWM IC**

```
void generar_ewmic {  
    se suma gestiones de la clase GESTION del SUBSISTEMA  
    GESTION TELEFONICA por INDICADOR CLAVE;  
    se llenan los atributos de la clase;  
    se ejecuta pa_ingresa_EWMIC;  
}
```

- **Operación Consultar EWM IC**

```
Dataview consultar_ewmic {  
    se llenan los atributos de la clase;  
    se ejecuta pa_consulta_EWMIC;  
    se devuelve vista con datos;  
}
```

- **CLASE CALIFICACIÓN**

- **Operación Crear Calificación**

```
void crear_calificación {  
    se llenan los atributos de la clase;  
    se ejecuta pa_ingresa_Calificacion;  
}
```

- **Operación Inactivar Calificación**

```
void inactivar_calificación {  
    se llenan los atributos de la clase;  
    se ejecuta pa_inactiva_Calificacion;  
}
```

- **Operación Modificar Calificación**

```
void modificar_calificación {  
    se llenan los atributos de la clase;  
    se ejecuta pa_actualiza_Calificacion;  
}
```

- **Operación Consultar Calificación**

```
Dataview consultar_calificación {  
    se llenan los atributos de la clase;  
    se ejecuta pa_consulta_Calificacion;  
    se devuelve vista con datos;  
}
```

3. IMPLEMENTACIÓN Y PRUEBAS

3.1. IMPLEMENTACIÓN

3.1.1. HERRAMIENTAS UTILIZADAS

La plataforma elegida para realizar el prototipo es Microsoft Visual Studio .NET, esta plataforma la conforman el framework, y los lenguajes .NET, como se visualiza en la figura a continuación.

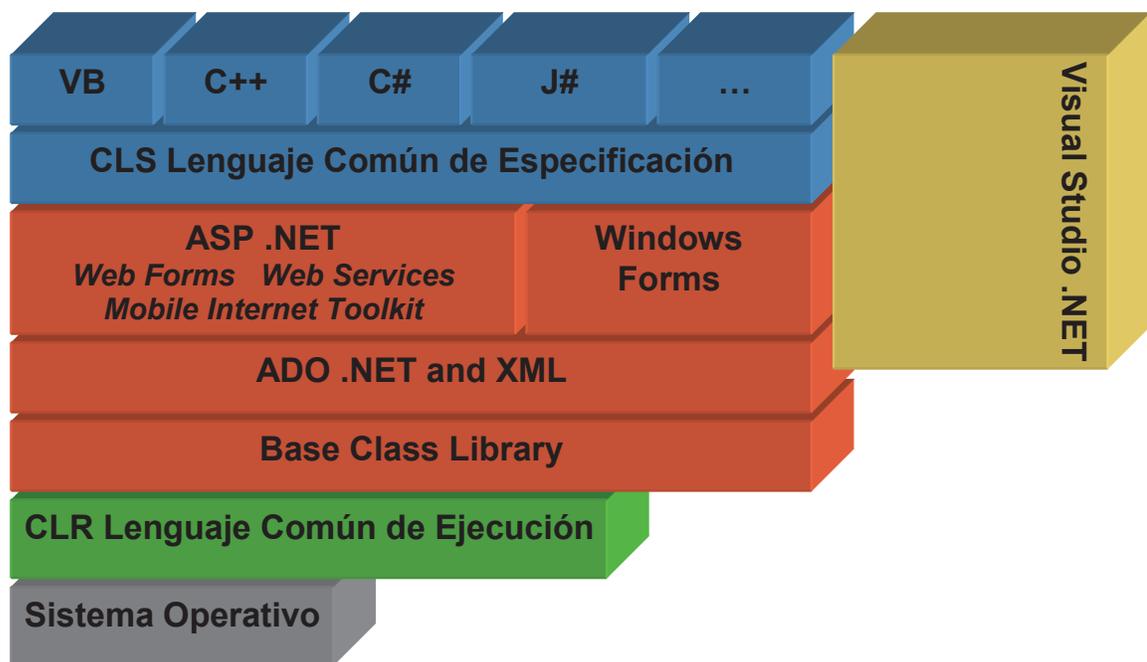


Figura 3.1 Plataforma Microsoft Visual.Net
Ref.: Professional C Sharp

Visual Studio .NET es el ambiente de programación creado exclusivamente para servicios Web XML. Al permitir que las aplicaciones se comuniquen y compartan datos a través de Internet, los servicios Web XML permiten que los desarrolladores ensamblen aplicaciones con código existente y nuevo, independientemente de la plataforma, lenguaje de programación o modelo de objetos.

3.1.1.1. FRONT-END

Visual Studio .NET permite crear aplicaciones que aprovechan la potencia de World Wide Web, por ello se ha escogido trabajar con ASP.NET, estas aplicaciones serán creadas en Visual C#.

3.1.1.1.1. ASP.NET

ASP.NET es un ambiente de programación construido sobre el entorno NGWS (New Generation Windows Services, o sea, "Servicios de la Nueva Generación de Windows"), que permite crear poderosas aplicaciones de Internet.

ASP.NET ofrece varias ventajas importantes sobre los modelos previos de desarrollo para Internet:

- Mejor Eficiencia
 - ASP.NET corre código compilado sobre el entorno NGWS en el servidor. Distinto a sus predecesores interpretados.
 - ASP.NET usa amarres tempranos ("early binding"), así como compilación justo a tiempo ("just-in-time compilation"), optimización nativa, y servicios de caché, sin configuración adicional. Para los desarrolladores, esto significa eficiencia dramáticamente superior antes de escribir la primera línea de código.

- Herramientas superiores de desarrollo
 - ASP.NET tiene una "caja de herramientas" rica :
 - El ambiente de desarrollo integrado de Visual Studio.NET.
 - La edición WYSIWYG
 - La creación de controles mediante "drag-and-drop"
 - La publicación automática son varias ventajas.

- Poder y Flexibilidad

Porque ASP.NET está basado en el Entorno Común de Ejecución de Lenguajes (Common Language Runtime, o "CLR"), el poder y la flexibilidad de la plataforma completa está disponible para los desarrolladores. Las librerías de Clases del CLR, la Mensajería, y las soluciones de Acceso a Datos, son accesibles al través del Internet. ASP.NET permite el uso de una gran variedad de lenguajes de programación y, por tanto, usted puede escoger el mejor lenguaje para su aplicación, o particionar su aplicación en varios lenguajes. Más aún, la interoperabilidad del CLR garantiza que su inversión en el desarrollo de aplicaciones COM es preservada cuando se migra a ASP.NET.

- Simplicidad

ASP.NET hace fácil el ejecutar tareas comunes, desde el simple envío de un formulario o la autenticación de un cliente, hasta el despliegue y la configuración de un Web. Por ejemplo, el entorno de paginado de ASP.NET le permite construir interfaces de usuario que separan limpiamente la lógica de su aplicación del código de su presentación, y maneja eventos con un modelo sencillo de procesamiento de formularios al estilo de Visual Basic. Adicionalmente, el CLR simplifica el desarrollo con servicios de código gerenciado, como el conteo automático de referencias y la limpieza automática de la memoria utilizada por su aplicación.

- Gerenciabilidad

ASP.NET usa un sistema jerárquico de configuración, basado en archivos de texto, que simplifica la aplicación de parámetros de configuración al servidor y sus aplicaciones. Porque la información de configuración es almacenada como texto, nuevos parámetros pueden ser configurados sin recurrir a herramientas de administración locales. Esta filosofía de "cero administración local" también se extiende al despliegue de aplicaciones de ASP.NET. Una aplicación de ASP.NET se

despliega a un servidor simplemente copiando los archivos necesarios al servidor. No hay que reiniciar el servidor, ni siquiera para reemplazar código compilado que ya está en servicio.

- Escalabilidad y Disponibilidad

ASP.NET ha sido diseñado para la escalabilidad con características específicamente dirigidas a mejorar el funcionamiento de servidores racimados (clustered) y de servidores con procesadores múltiples. Los procesos del servidor son vigilados y gerenciados por el entorno del ambiente de ejecución de ASP.NET, así que si algún proceso se entorpece o se detiene, un nuevo proceso puede ser creado para reemplazarlo, lo cual ayuda a mantener la disponibilidad de su aplicación para manejar solicitudes de servicio.

- Personalización y Extensibilidad

ASP.NET entrega una arquitectura bien formada que permite que los desarrolladores "enchufen" su código al nivel apropiado. De hecho, es posible el extender o reemplazar cualquier sub-componente del ambiente de ejecución de ASP.NET con un componente personalizado. La implementación de autenticación personalizada o de servicios de mantenimiento de estado nunca ha sido tan sencilla.

- Seguridad

Con autenticación nativa de Windows y configuración individual por aplicación, usted puede estar tranquilo: sus aplicaciones están seguras.

3.1.1.1.2. C

C# (leído en inglés "C Sharp" y en español "C Almohadilla") es el nuevo lenguaje de propósito general diseñado por Microsoft para su plataforma .NET. Sus principales creadores son Scott Wiltamuth y Anders Hejlsberg, éste último

también conocido por haber sido el diseñador del lenguaje Turbo Pascal y la herramienta RAD Delphi.

Aunque es posible escribir código para la plataforma .NET en muchos otros lenguajes, C# es el único que ha sido diseñado específicamente para ser utilizado en ella, por lo que programarla usando C# es mucho más sencillo e intuitivo que hacerlo con cualquiera de los otros lenguajes ya que C# carece de elementos heredados innecesarios en .NET. Por esta razón, se suele decir que C# es el lenguaje nativo de .NET

La sintaxis y estructuración de C# es muy parecida a la de C++ o Java, puesto que la intención de Microsoft es facilitar la migración de códigos escritos en estos lenguajes a C# y facilitar su aprendizaje a los desarrolladores habituados a ellos. Sin embargo, su sencillez y el alto nivel de productividad son comparables con los de Visual Basic.

Un lenguaje que hubiese sido ideal utilizar para estos menesteres es Java, pero debido a problemas con la empresa creadora del mismo -Sun-, Microsoft ha tenido que desarrollar un nuevo lenguaje que añadiese a las ya probadas virtudes de Java las modificaciones que Microsoft tenía pensado añadirle para mejorarlo aún más y hacerlo un lenguaje orientado al desarrollo de componentes.

En resumen, C# es un lenguaje de programación que toma las mejores características de lenguajes preexistentes como Visual Basic, Java o C++ y las combina en uno solo. El hecho de ser relativamente reciente no implica que sea inmaduro, pues Microsoft ha escrito la mayor parte de la BCL usándolo, por lo que su compilador es el más depurado y optimizado de los incluidos en el .NET Framework SDK.

Las características generales son:

- Sencillez
C# elimina muchos elementos que otros lenguajes incluyen y que son innecesarios en .NET. Por ejemplo:

- El código escrito en C# es autocontenido, lo que significa que no necesita de ficheros adicionales al propio fuente tales como ficheros de cabecera o ficheros IDL
 - El tamaño de los tipos de datos básicos es fijo e independiente del compilador, sistema operativo o máquina para quienes se compile (no como en C++), lo que facilita la portabilidad del código.
 - No se incluyen elementos poco útiles de lenguajes como C++ tales como macros, herencia múltiple o la necesidad de un operador diferente del punto (.) acceder a miembros de espacios de nombres (::)
- Modernidad

C# incorpora en el propio lenguaje elementos que a lo largo de los años ha ido demostrándose son muy útiles para el desarrollo de aplicaciones y que en otros lenguajes como Java o C++ hay que simular, como un tipo básico decimal que permita realizar operaciones de alta precisión con reales de 128 bits (muy útil en el mundo financiero), la inclusión de una instrucción for each que permita recorrer colecciones con facilidad y es ampliable a tipos definidos por el usuario, la inclusión de un tipo básico string para representar cadenas o la distinción de un tipo bool específico para representar valores lógicos.
 - Orientación a objetos

Como todo lenguaje de programación de propósito general actual, C# es un lenguaje orientado a objetos, aunque eso es más bien una característica del CTS que de C#. Una diferencia de este enfoque orientado a objetos respecto al de otros lenguajes como C++ es que el de C# es más puro en tanto que no admiten ni funciones ni variables globales sino que todo el código y datos han de definirse dentro de definiciones de tipos de datos, lo que reduce problemas por conflictos de nombres y facilita la legibilidad del código.

C# soporta todas las características propias del paradigma de programación orientada a objetos: encapsulación, herencia y polimorfismo.

En lo referente a la encapsulación es importante señalar que aparte de los típicos modificadores `public`, `private` y `protected`, C# añade un cuarto modificador llamado `internal`, que puede combinarse con `protected` e indica que al elemento a cuya definición precede sólo puede accederse desde su mismo ensamblado.

Respecto a la herencia -a diferencia de C++ y al igual que Java- C# sólo admite herencia simple de clases ya que la múltiple provoca más quebraderos de cabeza que facilidades y en la mayoría de los casos su utilidad puede ser simulada con facilidad mediante herencia múltiple de interfaces. De todos modos, esto vuelve a ser más bien una característica propia del CTS que de C#.

Por otro lado y a diferencia de Java, en C# se ha optado por hacer que todos los métodos sean por defecto sellados y que los redefinibles hayan de marcarse con el modificador `virtual` (como en C++), lo que permite evitar errores derivados de redefiniciones accidentales. Además, un efecto secundario de esto es que las llamadas a los métodos serán más eficientes por defecto al no tenerse que buscar en la tabla de funciones virtuales la implementación de los mismos a la que se ha de llamar. Otro efecto secundario es que permite que las llamadas a los métodos virtuales se puedan hacer más eficientemente al contribuir a que el tamaño de dicha tabla se reduzca.

- Orientación a componentes

La propia sintaxis de C# incluye elementos propios del diseño de componentes que otros lenguajes tienen que simular mediante construcciones más o menos complejas. Es decir, la sintaxis de C#

permite definir cómodamente propiedades (similares a campos de acceso controlado), eventos (asociación controlada de funciones de respuesta a notificaciones) o atributos (información sobre un tipo o sus miembros)

- Gestión automática de memoria

Como ya se comentó, todo lenguaje de .NET tiene a su disposición el recolector de basura del CLR. Esto tiene el efecto en el lenguaje de que no es necesario incluir instrucciones de destrucción de objetos. Sin embargo, dado que la destrucción de los objetos a través del recolector de basura es indeterminista y sólo se realiza cuando éste se active –ya sea por falta de memoria, finalización de la aplicación o solicitud explícita en el fuente-, C# también proporciona un mecanismo de liberación de recursos determinista a través de la instrucción `using`.

- Seguridad de tipos

C# incluye mecanismos que permiten asegurar que los accesos a tipos de datos siempre se realicen correctamente, lo que permite evita que se produzcan errores difíciles de detectar por acceso a memoria no perteneciente a ningún objeto y es especialmente necesario en un entorno gestionado por un recolector de basura. Para ello se toman medidas del tipo:

- Sólo se admiten conversiones entre tipos compatibles. Esto es, entre un tipo y antecesores suyos, entre tipos para los que explícitamente se haya definido un operador de conversión, y entre un tipo y un tipo hijo suyo del que un objeto del primero almacenase una referencia del segundo (downcasting) Obviamente, lo último sólo puede comprobarlo en tiempo de ejecución el CLR y no el compilador, por lo que en realidad el CLR y el compilador colaboran para asegurar la corrección de las conversiones.

- No se pueden usar variables no inicializadas. El compilador da a los campos un valor por defecto consistente en ponerlos a cero y controla mediante análisis del flujo de control del fuente que no se lea ninguna variable local sin que se le haya asignado previamente algún valor.
 - Se comprueba que todo acceso a los elementos de una tabla se realice con índices que se encuentren dentro del rango de la misma.
 - Se puede controlar la producción de desbordamientos en operaciones aritméticas, informándose de ello con una excepción cuando ocurra. Sin embargo, para conseguirse un mayor rendimiento en la aritmética estas comprobaciones no se hacen por defecto al operar con variables sino sólo con constantes (se pueden detectar en tiempo de compilación)
 - A diferencia de Java, C# incluye delegados, que son similares a los punteros a funciones de C++ pero siguen un enfoque orientado a objetos, pueden almacenar referencias a varios métodos simultáneamente, y se comprueba que los métodos a los que apunten tengan parámetros y valor de retorno del tipo indicado al definirlos.
 - Pueden definirse métodos que admitan un número indefinido de parámetros de un cierto tipo, y a diferencia lenguajes como C/C++, en C# siempre se comprueba que los valores que se les pasen en cada llamada sean de los tipos apropiados.
- Instrucciones seguras
Para evitar errores muy comunes, en C# se han impuesto una serie de restricciones en el uso de las instrucciones de control más comunes. Por ejemplo, la guarda de toda condición ha de ser una expresión condicional y no aritmética, con lo que se evitan errores por confusión

del operador de igualdad (==) con el de asignación (=); y todo caso de un switch ha de terminar en un break o goto que indique cuál es la siguiente acción a realizar, lo que evita la ejecución accidental de casos y facilita su reordenación.

- Sistema de tipos unificado

A diferencia de C++, en C# todos los tipos de datos que se definan siempre derivarán, aunque sea de manera implícita, de una clase base común llamada System.Object, por lo que dispondrán de todos los miembros definidos en ésta clase (es decir, serán “objetos”)

A diferencia de Java, en C# esto también es aplicable a los tipos de datos básicos. Además, para conseguir que ello no tenga una repercusión negativa en su nivel de rendimiento, se ha incluido un mecanismo transparente de boxing y unboxing con el que se consigue que sólo sean tratados como objetos cuando la situación lo requiera, y mientras tanto puede aplicárseles optimizaciones específicas.

El hecho de que todos los tipos del lenguaje deriven de una clase común facilita enormemente el diseño de colecciones genéricas que puedan almacenar objetos de cualquier tipo.

- Extensibilidad de tipos básicos

C# permite definir, a través de estructuras, tipos de datos para los que se apliquen las mismas optimizaciones que para los tipos de datos básicos. Es decir, que se puedan almacenar directamente en pila (luego su creación, destrucción y acceso serán más rápidos) y se asignen por valor y no por referencia. Para conseguir que lo último no tenga efectos negativos al pasar estructuras como parámetros de métodos, se da la posibilidad de pasar referencias a pila a través del modificador de parámetro ref.

- Extensibilidad de operadores

Para facilitar la legibilidad del código y conseguir que los nuevos tipos de datos básicos que se definan a través de las estructuras estén al mismo nivel que los básicos predefinidos en el lenguaje, al igual que C++ y a diferencia de Java, C# permite redefinir el significado de la mayoría de los operadores -incluidos los de conversión, tanto para conversiones implícitas como explícitas- cuando se apliquen a diferentes tipos de objetos.

Las redefiniciones de operadores se hacen de manera inteligente, de modo que a partir de una única definición de los operadores ++ y -- el compilador puede deducir automáticamente como ejecutarlos de manera prefijas y posfija; y definiendo operadores simples (como +), el compilador deduce cómo aplicar su versión de asignación compuesta (+=) Además, para asegurar la consistencia, el compilador vigila que los operadores con opuesto siempre se redefinan por parejas (por ejemplo, si se redefine ==, también hay que redefinir !=)

También se da la posibilidad, a través del concepto de indizador, de redefinir el significado del operador [] para los tipos de dato definidos por el usuario, con lo que se consigue que se pueda acceder al mismo como si fuese una tabla. Esto es muy útil para trabajar con tipos que actúen como colecciones de objetos.

- Extensibilidad de modificadores

C# ofrece, a través del concepto de atributos, la posibilidad de añadir a los metadatos del módulo resultante de la compilación de cualquier fuente información adicional a la generada por el compilador que luego podrá ser consultada en tiempo ejecución a través de la librería de reflexión de .NET . Esto, que más bien es una característica propia de la plataforma .NET y no de C#, puede usarse como un mecanismo para definir nuevos modificadores.

- Versionable

C# incluye una política de versionado que permite crear nuevas versiones de tipos sin temor a que la introducción de nuevos miembros provoquen errores difíciles de detectar en tipos hijos previamente desarrollados y ya extendidos con miembros de igual nombre a los recién introducidos.

Si una clase introduce un nuevo método cuyas redefiniciones deban seguir la regla de llamar a la versión de su padre en algún punto de su código, difícilmente seguirían esta regla miembros de su misma signatura definidos en clases hijas previamente a la definición del mismo en la clase padre; o si introduce un nuevo campo con el mismo nombre que algún método de una clase hija, la clase hija dejará de funcionar. Para evitar que esto ocurra, en C# se toman dos medidas:

- Se obliga a que toda redefinición deba incluir el modificador `override`, con lo que la versión de la clase hija nunca sería considerada como una redefinición de la versión de miembro en la clase padre ya que no incluiría `override`. Para evitar que por accidente un programador incluya este modificador, sólo se permite incluirlo en miembros que tengan la misma signatura que miembros marcados como redefinibles mediante el modificador `virtual`. Así además se evita el error tan frecuente en Java de creerse haber redefinido un miembro, pues si el miembro con `override` no existe en la clase padre se producirá un error de compilación.
- Si no se considera redefinición, entonces se considera que lo que se desea es ocultar el método de la clase padre, de modo que para la clase hija sea como si nunca hubiese existido. El compilador avisará de esta decisión a través de un mensaje de aviso que puede suprimirse incluyendo el modificador `new` en la definición del miembro en la clase hija para así indicarle explícitamente la intención de ocultación.

- Eficiente

En principio, en C# todo el código incluye numerosas restricciones para asegurar su seguridad y no permite el uso de punteros. Sin embargo, y a diferencia de Java, en C# es posible saltarse dichas restricciones manipulando objetos a través de punteros. Para ello basta marcar regiones de código como inseguras (modificador unsafe) y podrán usarse en ellas punteros de forma similar a cómo se hace en C++, lo que puede resultar vital para situaciones donde se necesite una eficiencia y velocidad procesamiento muy grandes.

- Compatible

Para facilitar la migración de programadores, C# no sólo mantiene una sintaxis muy similar a C, C++ o Java que permite incluir directamente en código escrito en C# fragmentos de código escrito en estos lenguajes, sino que el CLR también ofrece, a través de los llamados Platform Invocation Services (PInvoke), la posibilidad de acceder a código nativo escrito como funciones sueltas no orientadas a objetos tales como las DLLs de la API Win32. Nótese que la capacidad de usar punteros en código inseguro permite que se pueda acceder con facilidad a este tipo de funciones, ya que éstas muchas veces esperan recibir o devuelven punteros.

También es posible acceder desde código escrito en C# a objetos COM. Para facilitar esto, el .NET Framework SDK incluye una herramientas llamadas tbimp y regasm mediante las que es posible generar automáticamente clases proxy que permitan, respectivamente, usar objetos COM desde .NET como si de objetos .NET se tratase y registrar objetos .NET para su uso desde COM.

Finalmente, también se da la posibilidad de usar controles ActiveX desde código .NET y viceversa. Para lo primero se utiliza la utilidad aximp, mientras que para lo segundo se usa la ya mencionada regasm.

3.1.1.2. BACK-END

Para la base de datos se eligió Microsoft SQL Server 2000 por tratarse de una base de datos robusta que satisface los requerimientos planteados, esta herramienta posee las siguientes características:

- Funcionamiento en ambiente cliente – servidor
- Adecuada documentación
- Fácil utilización y mantenimiento
- Mecanismos de respaldo y recuperación
- Soporte Transaccional
- Manejo de procedimientos almacenados

SQL Server es un sistema administrador para Bases de Datos relacionales basadas en la arquitectura Cliente / Servidor (RDBMS) que usa Transact-SQL para mandar peticiones entre un cliente y el SQL Server.

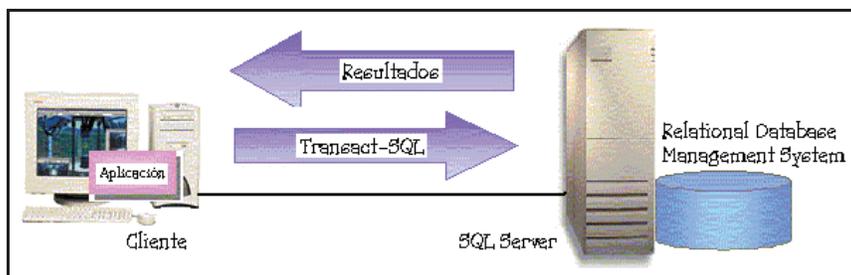


Figura 3.2 SQL Server

Ref.: Autor

SQL Server usa la arquitectura Cliente / Servidor para separar la carga de trabajo en tareas que corran en computadoras tipo servidor y tareas que corran en computadoras tipo cliente:

El Cliente es responsable de la parte lógica y de presentar la información al usuario. Generalmente, el cliente corre en una o más computadoras Cliente, aunque también puede correr en una computadora Servidor con SQL Server.

SQL Server administra Bases de Datos y distribuye los recursos disponibles del servidor (tales como memoria, operaciones de disco, etc.) entre las múltiples peticiones.

La arquitectura Cliente /Servidor permite desarrollar aplicaciones para realizar en una variedad de ambientes.

El RDBMS es responsable de:

- Mantener las relaciones entre la información y la Base de Datos.
- Asegurarse de que la información es almacenada correctamente, es decir, que las reglas que definen las relaciones ente los datos no sean violadas.
- Recuperar toda la información en un punto conocido en caso de que el sistema falle.

El TRANSACT – SQL usado como lenguaje de programación para SQL Server. SQL es un conjunto de comandos que permite especificar la información que se desea restaurar o modificar. Con Transact – SQL se puede tener acceso a la información, realizar búsquedas, actualizar y administrar sistemas de Bases de Datos Relacionales.

SQL Server valida a los usuarios con 2 niveles de seguridad: autenticación del login y validación de permisos en la Base de Datos de cuentas de usuarios y de roles. La autenticación identifica al usuario que está usando una cuenta y verifica sólo la habilidad de conectarse con SQL Server. El usuario debe tener permiso para acceder a las Bases de Datos en el Servidor. Esto se cumple para asignar permisos específicos para la Base de Datos, para las cuentas de usuario y los roles. Los permisos controlan las actividades que el usuario tiene permitido realizar en la Base de Datos del SQL Server.

Cada SQL Server tiene dos tipos de Bases de datos: Bases de Datos del Sistema y Bases de Datos del usuario. Las Bases de Datos del sistema

almacenan información acerca de SQL Server como un total. SQL Server usa la Base de Datos del sistema para operar y administrar al sistema. Las Bases de Datos de usuarios, son Bases de Datos creadas por los usuarios. Una copia del SQL Server puede administra una o más Bases de datos de usuario.

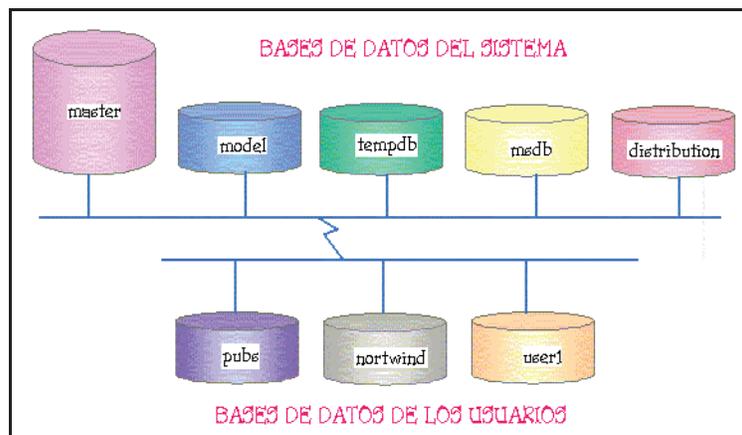


Figura 3.3 Tipos de BDD SQL Server
Ref.: Autor

Cuando SQL Server es instalado, el setup crea 4 bases de datos de sistema y 2 de usuario, de ejemplo. La Base de Datos de distribución es instalada cuando se configura SQL Server para actividades de replicación.

3.1.2. MENÚ PRINCIPAL DEL PROTOTIPO

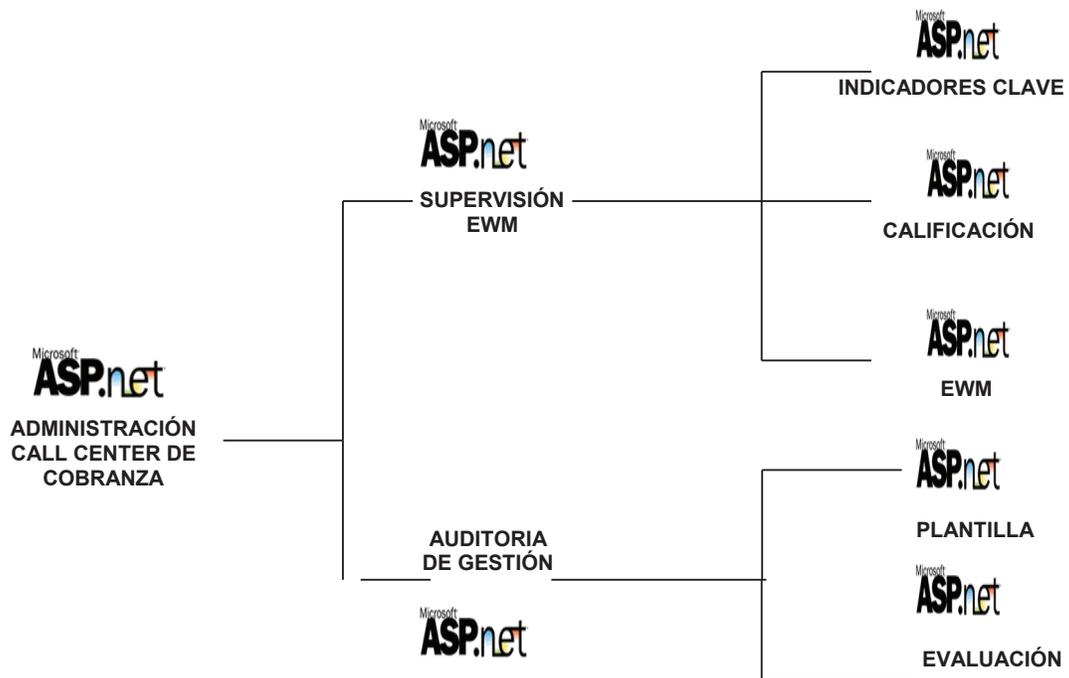


Figura 3.4 Menú Principal del Prototipo
Ref.: Autor

3.1.3. ESTÁNDARES PARA IMPLEMENTACIÓN

3.1.3.1. Estándares de Interfaces

3.1.3.1.1. Formatos de Pantallas

PANTALLA DE ACCESO

USUARIO

CONTRASEÑA

BOTON DE ACCION

Figura 3.5 Formato Pantalla de Acceso

PANTALLA DE SELECCIÓN DE MODULOS

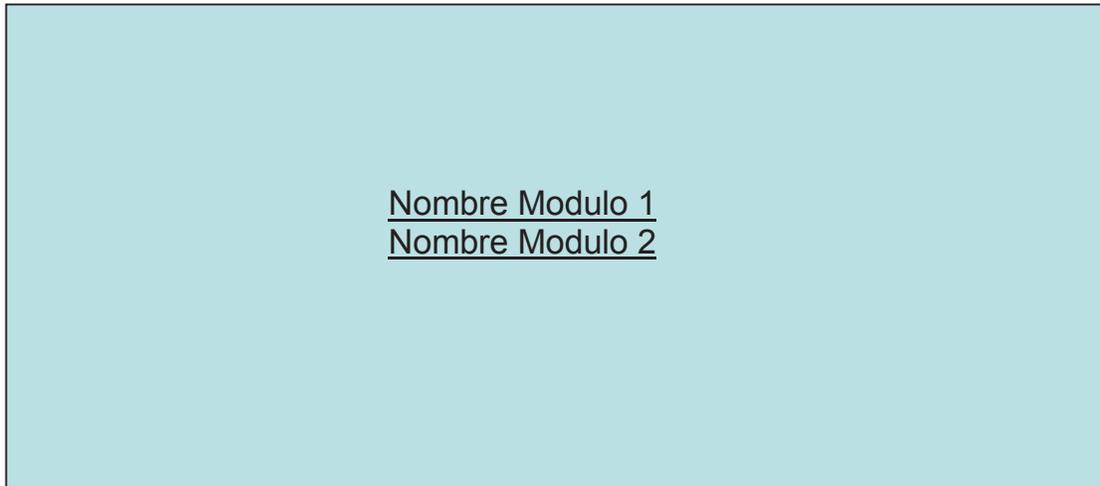


Figura 3.6 Formato Pantalla de Selección de Módulos

PANTALLA DE MENU Y SUBMENU

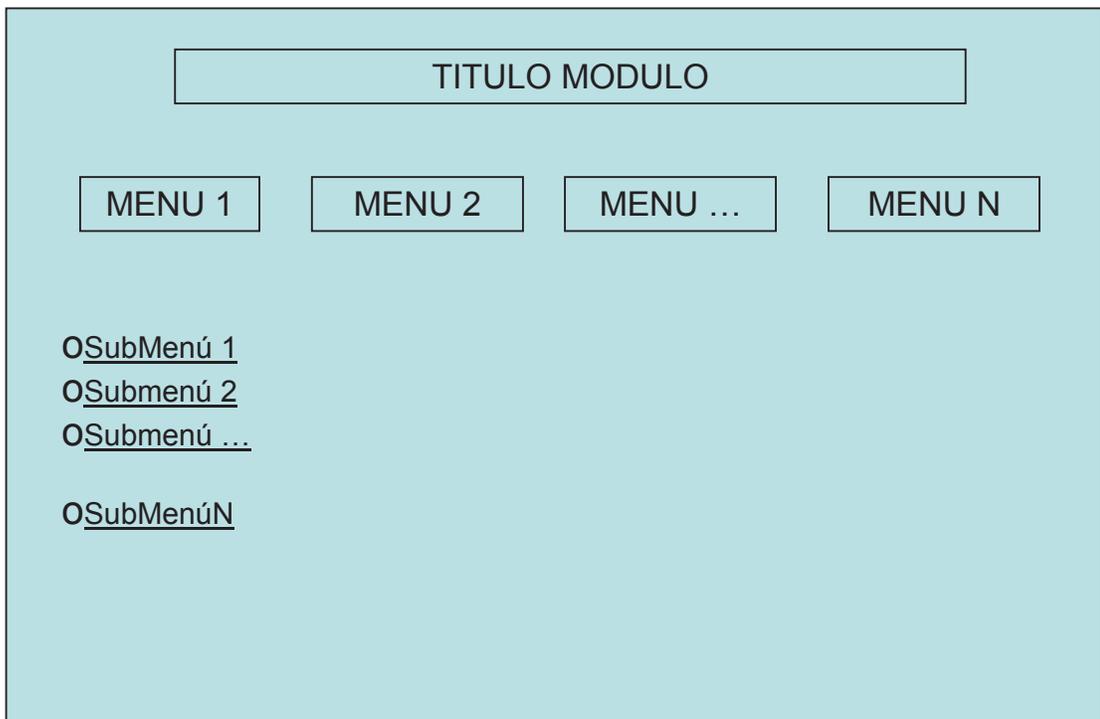


Figura 3.7 Formato Pantalla de Menú y Submenú

PANTALLA DE MANTENIMIENTO DE DATOS

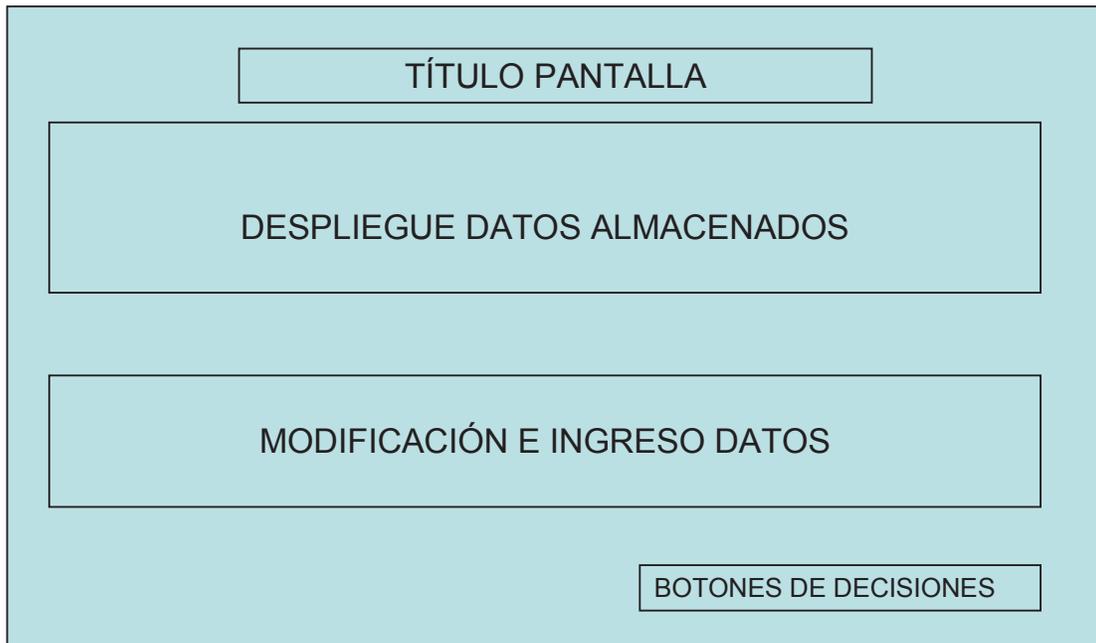


Figura 3.8 Formato Pantalla de Mantenimiento de Datos

PANTALLA DE REPORTES DE DATOS

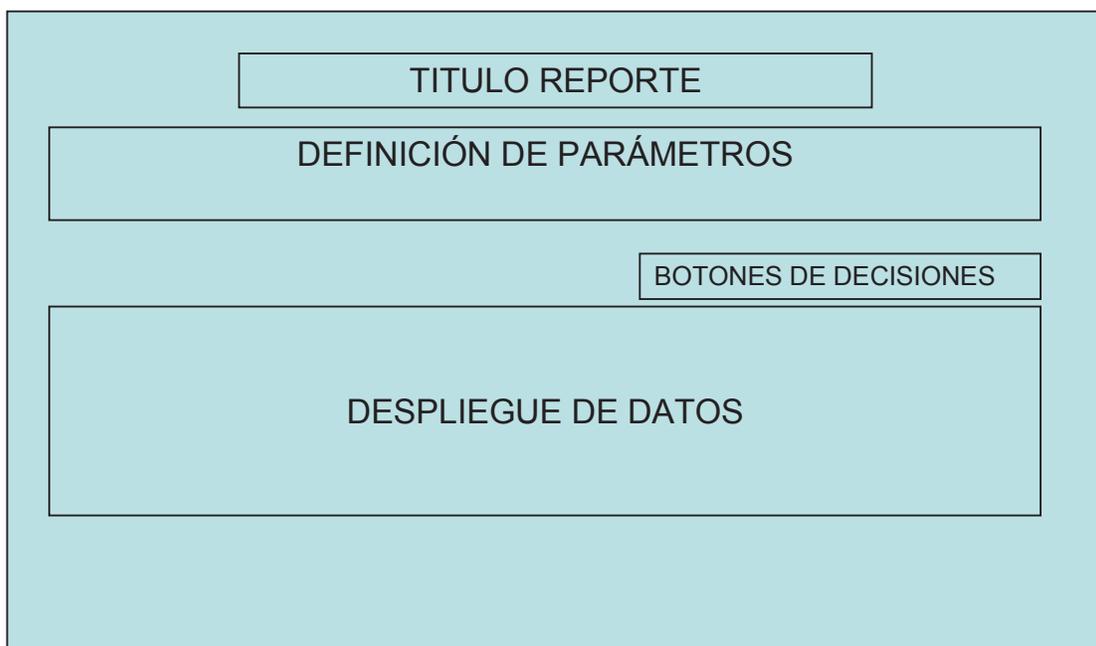


Figura 3.9 Formato Pantalla de Reportes de Datos

PANTALLA DE REPORTES ESTADÍSTICOS



Figura 3.10

Formato Pantalla de Reportes Estadísticos

3.1.3.1.2. Nomenclatura en Formas

Tabla 3.1. Implementación: Nomenclatura en Formas

NOMBRE: NombreWebForm

Objeto de WebForm	Nomenclatura
Texto	txtNombreObjeto
Label	lblNombreObjeto
Botón	cmdNombreObjeto
Grid	dgNombreObjeto
Botón imagen	imgbtnNombreObjeto
Radio Boton	rbtnNombreObjeto
Combo	cboNombreObjeto
Panel	pnlNombreObjeto
Check Box	chkNombreObjeto
Dundas Chart para gráficos	chartNombreObjeto
Required Field Validator	rvNombreObjeto
Custom Validator	cvNombreObejto

3.1.3.2. Estándares de Programación

3.1.3.2.1. Variables

- Cada variable en C# tendrá el siguiente formato:
NombreVariable
- Cada variable en SQL tendrá el siguiente formato:

@Inicial_NombreVariable

Donde Inicial está de acuerdo al cuadro siguiente:

Tabla 3.2: Implementación: Nomenclatura Tipos de datos

Tipo de Dato	Inicial
Bit	b
DateTime	d
Float	f
Smallint	s
Int	i
Char	c
Varchar	v

3.1.3.2.2. Formas, Clases y Procedimientos Almacenados

- Formas
 - El encabezado de los Web Forms tendrá el siguiente formato:

```

/*****
*      NOMBRE:                               *
*      DESCRIPCIÓN:                         *
*      VERSIONES:      Fecha      Autor      Motivo*
*
*****/

```

- Los nombres de las funciones de los Web Forms seguirán el siguiente formato:

NombreFunción_NombreWebForm

- Clases

- Los nombres de las clases seguirán el siguiente formato:

NombreClase

- El encabezado de las clases tendrá el siguiente formato:

```

/*****
*      NOMBRE:                               *
*      DESCRIPCIÓN:                         *
*      VERSIONES:      Fecha      Autor      Motivo*
*
*****/

```

- Los nombres de las funciones de las clases seguirán el siguiente formato:

NombreFunción_nombreclase

- Procedimientos Almacenados

- Los nombres de los procedimientos almacenados seguirán el siguiente formato:

pa_nombre

- El encabezado de los procedimientos almacenados tendrá el siguiente formato:

```

/*****
*      NOMBRE:                               *
*      DESCRIPCIÓN:                          *
*      VALOR DE ENTRADA                       *
*      VALOR DE SALIDA                        *
*      VERSIONES:      Fecha      Autor      Motivo*
*
*****/

```

En todos los procesos se incluirá comentarios explicativos que permitan la comprensión del mismo.

Para la indentación del código en SQL se utilizará tabuladores.

3.1.3.3. Estándares de Base de Datos

- El nombre de la Base de Datos será
AdministraciónCC
- Los nombres de las tablas serán de una o dos palabras juntas, las que identifiquen el contenido de la tabla.
- Los nombres de los campos serán de una o dos palabras juntas, las que describan el campo.
- Los nombres de las vistas tendrán el siguiente formato:
ViewNombreVista
- Los nombres de los índices tendrán el siguiente formato:
PK_NombreIndice

3.1.4. FUNCIONES Y OPERACIONES (COMPONENTES)

En .NET existe la posibilidad de realizar componentes los mismos que contienen funciones que son reutilizables para todo el proyecto.

3.1.4.1. Arquitectura

CAPAS	SUBSISTEMA DE AUDITORÍA DE GESTIÓN	SUBSISTEMA EWM
Capa de Presentación	auditoria.aspx plantilla.aspx, evaluación.aspx	supervisión.aspx indicador.aspx calificación.aspx ewm.aspx
Capa Lógica	LN_AUDITORIA	LN_EWM
Capa de Datos	Plantilla Pregunta Respuesta Evaluación Resultado Evaluación	IndicadorClave Calificación EWM EWMIndicadorClave

3.1.4.2. Componentes

3.1.4.2.1. Componente CONECTION

COMPONENTE	CONNECTION
	MSConexion
Método	Descripción
Conectar()	Conecta a la tabla
Ejecutar()	Ejecuta una transacción
Consultar()	Realiza una consulta
Consultar_Vista()	Consulta y devuelve una vista
ConsultarDR()	Consulta un Data Reader
cerrarDataReader()	Cierra sesión de Data Reader

3.1.4.2.2. *Componente CATALOGOS*

COMPONENTE	CATALOGOS
	LN_CONSULTAS
Método	Descripción
Consulta_ciudad()	Consulta las ciudades
Consulta_producto()	Consulta de productos
Consulta_segmento()	Consulta de segmentos

3.1.4.2.3. *Componente AUDITORIA*

COMPONENTE	AUDITORIA
CLASE: Plantilla, Pregunta, Respuesta, Evaluación, Resultado Evaluación	LN_Auditoria
Método	Descripción
Consulta_plantilla()	Consulta de plantillas
Ingresa_plantilla()	Ingresa datos en tabla plantilla
Modifica_plantilla()	Modifica valores de tabla plantilla
Inactivar_plantilla()	Inactiva registro de tabla plantilla
Consulta_pregunta_plantilla()	Consulta las preguntas asociadas a una plantilla
Consulta_pregunta_plantilla_agente()	Consulta preguntas de una plantilla específica para un agente auditado
Ingresa_pregunta_plantilla()	Ingreso de pregunta para una plantilla
Ingresa_respuesta_pregunta()	Ingresa respuesta para una pregunta
Consulta_respuesta_pregunta ()	Consulta de respuesta que pertenecen a una pregunta
Consulta_respuesta_agente ()	Consulta de respuesta(s) auditadas de un agente
Elimina_respuesta_pregunta()	Elimina una respuesta a una pregunta asociada
Elimina_pregunta_y_respuestas()	Elimina una pregunta y todas sus respuestas asociadas
Actualiza_respuesta_pregunta()	Modifica valores de una respuesta
Actualiza_respuesta_pregunta_SiNo()	Modifica valores de una respuesta tipo SI o NO

Actualiza_respuesta_pregunta_multiple()	Modifica valores de una respuesta tipo Múltiple
Actualiza_solo_pregunta()	Modifica solo la pregunta
Consulta_Agentes()	Lista de todos los agentes (telefónicos) que tiene al menos una auditoria
Consulta_agente_por_cédula (string cédula)	Consulta de agente por cedula
Ingresa_evaluación()	Ingresa la evaluación de un agente
Ingresa_resultado_evaluación()	Ingresa la evaluación de un agente en tabla de resultado evaluación
Consulta_evaluaciones_Anteriores()	Consulta de las evaluaciones hechas a un agente en una fecha dada

3.1.4.2.4. Componente EWM

COMPONENTE	EWM
CLASE: Indicador Clave, Calificación, EWM, EWM	LN_EWM
Indicador Clave, Gestión	
Método	Descripción
Consulta_Tasa_Contactos()	Consulta de indicador tasa de contactos
Consulta_Tasa_Compromisos()	Consulta de indicador tasa de compromisos
Consulta_Tasa_Conexiones()	Consulta de indicador tasa de conexiones
Consulta_Tasa_Eficiencia()	Consulta de indicador tasa de eficiencia
Consulta_Indicador()	Consulta de los Pesos de Indicadores
Consulta_Calificación()	Consulta de las Calificación para los Indicadores
Modifica_Indicador()	Modifica los valores de los indicadores clave
Modifica_Calificación()	Modifica los valores de la tabla calificación
Ingresa_Indicador()	Crea indicador clave
Ingresa_Calificación()	Crea calificación
Consulta_Calificación_agente()	Consulta las calificaciones de los agentes
Consulta_EWM()	Consulta para reporte de EWM
Generar_EWM()	Construye el reporte EWM de acuerdo a parámetros definidos.

3.2. PRUEBAS

Luego de terminada la implementación es necesario garantizar la calidad del prototipo mediante la utilización de mecanismos de prueba de software que permitan detectar errores y verificar el cumplimiento de las especificaciones de análisis y diseño.

La estrategia que se utilizará para probar el software consiste en:

- Pruebas de unidad
Se centra en cada unidad del software
- Pruebas de Integración
Verifican el diseño y construcción de la arquitectura del software
- Pruebas de Validación
Se validan los requisitos establecidos como parte del análisis de requisitos del software, comparándolos con el prototipo construido.

3.2.1 PRUEBAS DE UNIDAD

Las pruebas de unidad centran el proceso de verificación de errores dentro de cada clase u objeto encapsulado. Los errores son localizados mediante un seguimiento de cómo las operaciones de cada clase reciben, procesan y devuelven la información.

El diseño de casos de prueba de unidad comienza luego de haber desarrollado, revisado y verificado la sintaxis del código fuente. Por este motivo se considera a esta prueba como algo adjunto al paso de codificación.

3.2.1.1 Definición de pruebas de unidad

Tabla 3.3: Pruebas de Unidad Clase Plantilla

CLASE	PLANTILLA	
OPERACIONES	Crear Plantilla, Inactivar Plantilla, Modificar Plantilla, Consultar Plantilla	
Casos de Prueba	Resultados esperados	Resultados Obtenidos
<p>Crear Plantilla.</p> <p>En la pantalla Definición de plantillas se selecciona ciudad: Quito, empresa y producto: todos, segmento: Maquinaria y máxima calificación: 100.</p>	<p>Presionar Visualizar y se deben desplegar los nombres contraídos de ciudad, empresa-producto y segmento. Luego de hacer click en el botón Guardar y se debe crear la plantilla.</p>	<p>Se crea la Plantilla correctamente y aparece incluida en la lista de plantillas creadas.</p>
<p>Consultar Plantilla.</p> <p>En la pantalla Definición de plantillas, se selecciona la plantilla: "ciudad: Quito, empresa-producto: todos, segmento: Construcción", presionando el botón Restaurar junto a la plantilla correspondiente.</p>	<p>Consultar los datos de la plantilla: "ciudad: Quito, empresa-producto: todos, segmento: Construcción"</p>	<p>Se despliega toda la información que contiene la plantilla "ciudad: Quito, empresa-producto: todos, segmento: Construcción" y su puntaje máximo.</p>
<p>Asignar Preguntas a una Plantilla.</p> <p>En el menú Definición de Preguntas, se presiona el botón Seleccionar junto a la plantilla creada</p>	<p>Al seleccionar la plantilla en la pantalla Definición de Preguntas, debe aparecer la</p>	<p>Al seleccionar la plantilla aparece la pantalla Asignación de preguntas en los campos</p>

<p>“ciudad: Quito, empresa-producto: todos, segmento: Construcción”. En la pantalla Asignación de preguntas se selecciona la opción Ingresar pregunta, se selecciona el tipo de pregunta “si-no”, se ingresa la pregunta: ¿Contesta amablemente al cliente? En el campo Puntaje se ingresa en valor 20 para “si” y 0 para ‘no’ y se presiona el botón Ingresar.</p>	<p>pantalla Asignación de preguntas y debe aparecer en los campos respectivos los nombres de “ciudad: Quito, empresa-producto: todos, segmento: Construcción”. Al presionar el botón Ingresar deben activarse los campos: tipo de pregunta y pregunta para seleccionar el tipo e ingresar la pregunta. Al seleccionar el tipo de pregunta se deben activar los campos Puntaje para “si” y no” Luego de ingresar los valores de puntajes, al presionar el botón Ingresar, la pregunta debe aparecer en la lista numerada de preguntas y se puede seguir ingresando más preguntas de la misma manera.</p>	<p>respectivos aparecen los nombres de “ciudad: Quito, empresa-producto: todos, segmento: Construcción”. Al presionar el botón Ingresar se activan los campos tipo de pregunta y pregunta. Al seleccionar el tipo de pregunta se activan los campos Puntaje para “si” y no”. Luego de ingresar los valores de puntajes, al presionar el botón Ingresar, la pregunta aparece en la lista numerada de preguntas y permite seguir ingresando más preguntas de la misma manera.</p>
<p>Modificar pregunta. En el menú Definición de preguntas se selecciona la pregunta ¿Contesta el teléfono amablemente? y se modifica a ¿Atiende amablemente al cliente?</p>	<p>Al seleccionar la pregunta que se desea modificar, se deben activar los campos: Pregunta y Puntajes. Luego de modificar la información deseada, se debe presionar el botón Actualizar y se debe grabar la pregunta modificada.</p>	<p>Al seleccionar la pregunta que se desea modificar, se activan los campos: Pregunta y Puntajes. Luego de modificar la información deseada, y presionar el botón Actualizar la pregunta modificada aparece en la lista.</p>

Tabla 3.4: Pruebas de Unidad Clase Evaluación

CLASE	EVALUACION	
OPERACIONES	Crear Evaluación, Consultar Evaluación	
Casos de Prueba	Resultados esperados	Resultados Obtenidos
<p>Crear Evaluación.</p> <p>En el sub menú evaluación Agente, seleccionamos: Ángel Miranda.</p> <p>Cuando aparece la lista de plantillas disponibles para el agente se selecciona la plantilla “GUA-IIA-M-AGR” y se responde las preguntas: ¿Atiende amablemente al cliente? “si”. Se ingresa el nombre del cliente: Fopeca. Aparece el puntaje de 80 para “si” y se presiona el botón Ingresar.</p> <p>A la pregunta “Tiempo para concretar respuesta”, se ingresa el valor 10 y presionamos el botón Ingresar y se suma el puntaje 90 asignado para ese rango al puntaje acumulado. Finalmente a la pregunta “Nivel de Conocimiento del producto” seleccionamos “medio” y se suma el valor de 50 al puntaje acumulado.</p>	<p>Se debe activar el campo agente a evaluar, y debe permitir buscar por cédula, nombre del agente o se presiona el botón buscar para que aparezca el listado de todos los agentes.</p> <p>Luego de seleccionar el agente, se deben desplegar sólo las plantillas que pertenecen a la ciudad a donde está asignado el agente. Al seleccionar una plantilla, deben aparecer las preguntas y sus posibles respuestas para contestarlas de una en una.</p> <p>Cada vez que se ingresa una respuesta debe ir acumulándose el puntaje y debe aparecer la siguiente pregunta y sus posibles respuestas.</p> <p>Cuando se termina de responder todas las preguntas, deberá aparecer el mensaje “Ha contestado la última pregunta. Presione finalizar para finalizar la Auditoría”.</p>	<p>Se activa el campo agente a evaluar, luego de seleccionar el agente, aparecen en la pantalla los datos del agente y se despliegan sólo las plantillas que pertenecen a la ciudad GUA. Al seleccionar una plantilla, aparecen de una en una las preguntas y sus posibles respuestas.</p> <p>Cada vez que se ingresa una respuesta se acumula el puntaje y aparece la siguiente pregunta y sus posibles respuestas.</p> <p>Cuando se termina de responder todas las preguntas, aparece el mensaje “Ha contestado la última pregunta. Presione finalizar para finalizar la Auditoría”.</p> <p>El puntaje acumulado que aparece es de 220.</p> <p>Al presionar el botón Finalizar se graba la evaluación completa.</p>

<p>Aparece el mensaje “Ha contestado la última pregunta. Presione finalizar para finalizar la Auditoría”.</p>	<p>Al presionar el botón Finalizar se debe grabar la evaluación completa y el puntaje acumulado deberá ser 220.</p>	
<p>Consultar Evaluación. En el menú Consultar Evaluación, se ingresa el rango de fechas donde se supone que está la o las evaluaciones al Agente Ángel Miranda. Se selecciona Todos los Auditores y el recurso Ángel Miranda, Puntaje de 0 a 1000 y se presiona el botón Aceptar.</p>	<p>Al seleccionar el agente Ángel Miranda, Se debe poder visualizar la o las evaluaciones de dicho agente en el rango de fechas seleccionado. Debe permitir imprimir el reporte de evaluaciones. Al seleccionar una evaluación específica se debe presentar en la pantalla la información de preguntas, respuestas y puntajes en esa evaluación.</p>	<p>Al seleccionar el agente Ángel Miranda, en un rango de fechas, se visualiza en la pantalla la lista de las evaluaciones de dicho agente en el rango de fechas seleccionado. Ese listado puede imprimirse. Al seleccionar una evaluación específica se presentaren la pantalla la información de preguntas, respuestas y puntajes en esa evaluación.</p>

Tabla 3.5: Pruebas de Unidad Clase Indicador Clave

CLASE	INDICADOR CLAVE	
OPERACIONES	Crear Indicador Clave, Modificar Indicador Clave, Consultar Indicador Clave	
Casos de Prueba	Resultados esperados	Resultados Obtenidos
<p>En la pantalla Crear Indicador clave, se ingresan los campos descripción: Contact Rate, peso 30%, óptimo 80%, ciudad Quito, producto: IIASA-M segmento todos.</p>	<p>Se deben activar los campos descripción, peso, óptimo, ciudad, producto, segmento, para ingresar la información del indicador clave. Se debe activar el botón guardar y se debe crear el indicador clave correctamente</p>	<p>Se crea el indicador correctamente.</p>
<p>En la pantalla Modificar</p>	<p>Debe solicitar descripción del</p>	<p>Se modifican correctamente los</p>

Indicador Clave se selecciona el indicador clave "contact rate", se modifica el campo óptimo a 70%	indicador clave, y desplegar los datos para que sean modificados. Se deben actualizar los campos modificados.	campos.
En la pantalla Consultar Indicador Clave, se selecciona el modificador clave "contact rate"	Debe solicitar descripción del indicador clave, y desplegar los datos.	Se despliega toda la información del Indicador Clave

Tabla 3.6: Pruebas de Unidad Calificación

CLASE	CALIFICACION	
OPERACIONES	Crear Calificación, Modificar Calificación	
Casos de Prueba	Resultados esperados	Resultados Obtenidos
En la pantalla Crear Calificación, se ingresan los campos: descripción corta: A, descripción larga: Excelente, ciudad: Quito, producto: IIASA-M, segmento: todos, valor mínimo 10, valor máximo 100	Se deben activar los campos descripción corta, descripción larga, ciudad, producto, segmento, valor mínimo, valor máximo, para ingresar la información de la calificación. Se debe activar el botón guardar y se debe crear la calificación correctamente.	Se crea correctamente la calificación.
En la pantalla Modificar Calificación se selecciona la calificación A, se modifica el campo valor máximo a 80.	Debe solicitar descripción corta de la calificación, y desplegar los datos para que sean modificados. Se deben actualizar los campos modificados.	Se modifican correctamente los campos de la calificación.
En la pantalla Consultar Calificación, se selecciona la calificación A.	Debe solicitar descripción corta de la calificación, y desplegar los datos.	Aparece toda la información de la Calificación

Tabla 3.7: Pruebas de Unidad EWM

CLASE	EWM	
OPERACIONES	Generar EWM, Consultar EWM	
Casos de Prueba	Resultados esperados	Resultados Obtenidos
En la pantalla Generar EWM, se ingresan los campos: ciudad: Quito, producto: IIASA-M segmento: CONSTRUCCION, agente: Ángel Miranda,	Se deben activar los campos ciudad, producto, segmento, agente, fecha, hora, para ingresar la información del EWM. Se debe activar el botón guardar y se debe generar el EWM	Se genera correctamente el EWM y aparece mensaje "EWM Generado"
En la pantalla Consultar EWM, se ingresan los campos: ciudad, producto, segmento.	Debe, y desplegar los datos.	Se despliega toda la información del EWM

3.2.2 PRUEBAS DE INTEGRACIÓN

Después de realizar las pruebas de unidad se debe realizar las pruebas que indiquen que no existen errores asociados con la interacción, para esto se debe construir una estructura de programa que esté de acuerdo con lo que indica el diseño.

El conjunto de clases debe ser probado primero de forma individual y luego realizar otra construcción que permita integrar otros conjuntos de clases, esta estrategia se denomina integración ascendente.

3.2.2.1 Definición de pruebas de integración

Tabla 3.8: Pruebas de Integración

Casos de Prueba	Clases Asociadas	Resultados Esperados	Resultados Obtenidos
Se modifica el tipo de la pregunta ¿cuál fue el resultado de la negociación? “si-no” a “selección múltiple” y se ingresan las siguientes posibles respuestas: compromiso de pago, negativa de pago.	PLANTILLA – PREGUNTA - RESPUESTA	En la plantilla, se debe permitir cambiar el tipo de pregunta y solicite las posibles respuestas.	Se modifica correctamente el tipo de pregunta en la tabla pregunta, y todos los atributos de la tabla respuesta.
Se genera un EWM, para la ciudad de Quito, todos los segmentos, todos los productos, en la fecha 30/03/2015, para el Agente Ángel Miranda	EWM – EWMINDICAD OR CLAVE	Al crearse el EWM, se deben crear en la tabla EWMIndicadorClave, registros con la calificación de cada indicador clave para este EWM.	Registros de calificación creados para cada indicador clave en la tabla EWMIndicadorClave

3.2.3 PRUEBAS DE VALIDACIÓN

Una vez terminadas las pruebas de integración se procede con las pruebas de validación que se realizan mediante una serie de casos de prueba que demuestren que los requisitos del prototipo especificados estén bien establecidos.

Todos los requerimientos funcionales se encuentran implementados en el prototipo, tal como lo indica la tabla a continuación:

Tabla 3.9: Pruebas de Validación

Requerimiento Funcional	Módulos	Submódulo	Opción
Administración de Indicador Clave	Supervisión de EWM	Indicadores Clave	<ul style="list-style-type: none"> ○ Crear Indicador Clave ○ Modificar Indicador Clave ○ Consultar Indicador Clave
Administración de Calificación	Supervisión de EWM	Calificación	<ul style="list-style-type: none"> ○ Crear Calificación ○ Modificar Calificación ○ Consultar Calificación
Construcción del Informe EWM	Supervisión de EWM	EWM	<ul style="list-style-type: none"> ○ Generar EWM
Consulta del Informe EWM	Supervisión de EWM	EWM	<ul style="list-style-type: none"> ○ Consultar EWM
Administración de Plantillas de Evaluación	Auditoría de Gestión	Plantilla	<ul style="list-style-type: none"> ○ Crear Plantilla ○ Modificar Plantilla ○ Consultar Plantilla
Auditoría de Gestión	Auditoría de Gestión	Evaluación	<ul style="list-style-type: none"> ○ Realizar Evaluación ○ Consultar Evaluación

4. CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES

- Por falta de conocimiento se corre el riesgo de pasar por alto una necesidad de información que ayuda a la óptima administración de un Call Center de Cobranzas, herramienta tecnológica y administrativa que permite unificar el procesamiento de los sistemas informáticos y las facilidades de la conmutación de llamadas telefónicas. Por este motivo es necesario seguir una estrategia de centrada en el cliente, la cual está destinada a lograr identificar y administrar las relaciones en aquellas cuentas más valiosas para una empresa, y cuyos objetivos principales son: Reducir Costos e Incrementar ganancias, aumentando la productividad de los agentes telefónicos y dando mayor satisfacción a los clientes.

- La información básica que debe ser analizada para administrar un Call Center es la que registra un agente telefónico durante una llamada, ya que a partir de estos datos se construyen los indicadores clave: Tasa de Contactos, Tasa de Conexiones, Tasa de compromisos y Eficiencia, los cuales sirven de ayuda al supervisor del Call Center para medir la productividad de sus agentes telefónicos y también la calidad de la base de datos que se está gestionando.

- Los indicadores clave son llevados a un reporte de monitoreo de alerta temprana EWM, el mismo que permitirá al supervisor y jefe del Call Center, evaluar en línea el desempeño de las estrategias establecidas y los resultados logrados por los agentes, lo cual permitirá tomar decisiones eficientemente y detectar posibles

desviaciones lo antes posible, dotándonos de las herramientas necesarias para la mejor toma de decisiones.

- La administración de un Call Center depende de muchos factores como las telecomunicaciones, recursos de información y aplicaciones dedicadas, las cuales deben funcionar perfectamente en conjunto para tener un rendimiento óptimo.

- Un programa de auditoría de gestión proporciona un sistema estructurado, analítico y permanente de evaluación de gestión de resultados. En este programa los indicadores clave representan los puntos clave para observar el desempeño de la operación del Call Center, ligados a la estrategia del negocio. Sirven para tomar decisiones y para ello es necesario relacionarlos. El trabajo que cumple el auditor de gestión es muy importante, ya que es quien evalúa la calidad de gestión en cada uno de los agentes telefónicos, para lo cual utiliza plantillas de evaluación mientras realiza la intervención de llamadas

- El monitoreo debe identificarse como un área funcional de retroalimentación al agente y apoyo a las áreas operativas, con una visión de mejora continua.

- En el Call Center se tiene el problema del control de los descansos, incremento de la demanda en las llamadas entrantes, control en los tiempos de atención, tiempo de conexión y tiempo efectivo del agente, cantidad de llamadas atendidas, tiempo por cada llamada, entre otras. Todas estas variables nos proporcionan elementos de evaluación tanto para el Call Center como para el personal.

- Una aplicación especializada de Work Force Management, nos permite tener la cantidad adecuada de personal de acuerdo a las

cargas de trabajo, tomando en cuenta la demanda de llamadas entrantes y salientes, el tipo de campaña y costos, de tal manera, que nos permite balancear el Call Center de la mejor forma para el negocio.

- Definir los pesos específicos que se necesita para la valoración de los agentes, depende de los resultados que se desea obtener y hacia dónde se quiere ir. Los pesos específicos dan el poder de direccionar la gestión y empujar al agente a que logre el resultado esperado.
- El objetivo del Balance Score Card es medir de forma estructurada y precisa, la participación individual de los agentes en la gestión telefónica en términos de: productividad, eficiencia, desempeño y calidad.
- En la realización de este prototipo se utilizó la metodología OMT (Object Modeling Technique), la cual es orientadas a objetos y abierta y está basada en los datos, al contrario de otras metodologías basadas en los casos de uso.
- La Aplicación del prototipo, se implementó con las herramientas Microsoft Visual Studio .NET como Front End y Microsoft SQL Server 2000 como Back End, por ser las herramientas que permiten implementar el prototipo de tal manera que la aplicación pueda ser ejecutada a través de la WEB. Además que permiten un buen manejo administración y almacenamiento de los datos por su potencialidad.
- La aplicación en sí, es muy sencilla, amigable, y fácil de usar. Además, que cumple con todos los requerimientos del problema y brinda ayuda a los supervisores y jefes de un Call Center de Cobranzas, para administrarlo de una manera rápida y eficiente.

4.2. RECOMENDACIONES

- Para la utilización del Prototipo de Administración de un Call Center, se deben tener claros los conceptos generales y la teoría de Administración de un call Center los cuales se describen en el capítulo I de este proyecto.
- Utilizar el informe Early Warning Monitor, que se obtiene del Prototipo y que constituye una herramienta valiosa para el objetivo principal de la administración de un Call Center que es determinar a tiempo posibles fallas en la gestión de cobro por parte de los agentes y definir las acciones a seguir.
- Para el análisis de cualquier sistema o prototipo, se recomienda, en primer lugar, comprender el problema que se debe resolver, y no construir el modelo de objetos anotando clases, asociaciones y herencias como prioridad.
- Un buen modelo debe poder ser entendido por expertos en el dominio del problema que no tengan conocimientos informáticos. Por lo tanto se debe poner mucho énfasis en las etapas de análisis y diseño de cualquier sistema, tomándose el debido tiempo para realizar un buen modelo del mismo.
- Es muy importante que se definan estándares para la implementación, debido a que se puede llevar un mejor y fácil seguimiento a los programas, lo cual ayuda al programador y al usuario.
- Para utilizar el prototipo, se recomienda seguir el manual de usuario e instalación del mismo, el cual explica claramente y paso a paso su funcionamiento.

5. REFERENCIAS BIBLIOGRÁFICAS

- {1} Espinoza Lizardo, Soluciones WEB- Telephony al alcance de los centros de Contacto, 2002, Alcatel, 1ra. Ed.
 - {2} García Eugenia. Desafíos y Oportunidades del Contact Center ante una nueva era en la gestión directiva. Ed. Instituto Mexicano de Telemarketing, 2002, 1ra Ed.
 - {3} Hernández Ponce Helio, Metodologías orientadas a objetos, 1999, Instituto Tecnológico de Morelia.
 - {4} Lemon Tree, Estándar ELTECO, 2002 1ra. Ed.
 - {5} Peña Joaquín y Castañeda Fernando. Mediciones y Métricas de un Call Center/Contact Center. Ed. Delta Ingenieros, 2002, 1ra Ed.
 - {6} Peña Joaquín y Castañeda Fernando. Factores Claves de desempeño en el Call Center/ Contact Center. Ed. Delta Ingenieros, 2002, 1ra Ed.
 - {7} Ramón Gustavo. Factores Claves en la toma de decisiones para la inversión y optimización de la Tecnología en los Contact Centers 2002, Customer Interaction Consulting, 1ra. Ed.
 - {8} Robinson Simon, Ollie Cornes, Jay Glynn, Burton Harvey, Creag moqueen, Jerod Moemeka, Christian Nagel, Morgan Sjinner, Karli Watson, Proffesional C Sharp, Wrox Press Ltda., 2001, 1ra Ed.
 - {9} Rumbaugh J., Blaha M,, Modelado y Diseño Orientado a Objetos, Ed. Prentice Hall Inc, 1991.1ra Ed.
 - {10} Silberscjat A., H.F.Korth, s. Sudarshan. Fundamentos de Bases de datos Ed. Mc. Gran Hill, 2002.
 - {11} Osorno Luis, Interpretación de Métricas, Factor de éxito en el Call Center/Contact Center 2002, Teleacción, 1ra. Ed.
 - {12} Osuna Luis, Aseguramiento de Calidad, parte fundamental en el proceso CRM, 2002, Dictaphone, 1ra. Ed.
- Direcciones electrónicas:
- {13} [http://technet.microsoft.com/es-es/library/ms130214\(v=sql.90\).aspx](http://technet.microsoft.com/es-es/library/ms130214(v=sql.90).aspx), Libros en pantalla de SQL SERVER.
 - {14} [http://technet.microsoft.com/es-es/library/ms187875\(v=sql.90\).aspx](http://technet.microsoft.com/es-es/library/ms187875(v=sql.90).aspx), Motor de la Base de Datos SQL Server.
 - {15} <http://www.contactforum.com.mx/media/uploads/cm20025C.pdf>. Como asegurar el éxito en los proyectos de Call Center: Factores Clave.
 - {16} <http://www.luxortec.com/blog/metricas-de-atencion-y-productividad-para-call-center/>. Métricas de Atención y Productividad para el call center.
 - {17} <http://blog.dreampbx.com/2014/02/el-valor-de-las-metricas-en-el-call-center/>
 - {18} <http://www.etc-easy.com/services/jobmatch/ccs.htm>. Simulación de un call center.

6. ANEXOS

6.1. MODELO FÍSICO DE LA BASE DE DATOS

6.2. ANEXO 2: CÓDIGO DEL PROTOTIPO

6.2.1. MÓDULO AUDITORIA DE GESTIÓN

PROCEDIMIENTOS ALMACENADOS

```

/*****
*   NOMBRE:      pa_consulta_ciudades      *
*   DESCRIPCIÓN: Consulta de ciudades      *
*   VALOR DE ENTRADA      *
*   VALOR DE SALIDA      *
*   VERSIONES:      *
*   Fecha      Autor      Motivo      *
*   10/10/2004      IO, GM      Creación      *
*****/
CREATE PROCEDURE pa_consulta_ciudades
AS
    SELECT
        CS.idCiudad AS IDCIUDADSIC,
        UPPER(ltrim(rtrim(CS.nombreCiudad)))
    FROM
        ADMINISTRACIONCC.dbo.CIUDAD      CS
    ORDER BY ltrim(rtrim(CS.nombreCiudad))

GO

```

WEB FORM Y C SHARP

```

/*****
*   NOMBRE:      PlantillaGestion.aspx      *
*   DESCRIPCIÓN: Definición de Plantilla      *
*   VALOR DE ENTRADA      *
*   VALOR DE SALIDA      *
*   VERSIONES:      *
*   Fecha      Autor      Motivo      *
*   10/10/2004      IO, GM      Creación      *
*****/

using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;

namespace AdministracionCCSistema.modulos.M28
{
    ///<summary>
    /// Summary description for PlantillaGestion.
    ///</summary>
    publicclass PlantillaGestion : System.Web.UI.Page
    {
        protected System.Web.UI.WebControls.TextBox txtSegmentos;
        protected System.Web.UI.WebControls.TextBox txtProductos;
    }
}

```

```

protected System.Web.UI.WebControls.TextBox txtCiudades;
protected System.Web.UI.WebControls.CheckBoxList chkSegmento;
protected System.Web.UI.WebControls.CheckBoxList chkProducto;
protected System.Web.UI.WebControls.CheckBoxList chkCiudad;
protected System.Web.UI.WebControls.ImageButton cmdArribaSegmento;
protected System.Web.UI.WebControls.ImageButton cmdAbajoSegmento;
protected System.Web.UI.WebControls.ImageButton cmdAbajoProducto;
protected System.Web.UI.WebControls.ImageButton cmdArribaProducto;
protected System.Web.UI.WebControls.ImageButton cmdArribaCiudad;
protected System.Web.UI.WebControls.ImageButton cmdAbajoCiudad;
protected System.Web.UI.WebControls.Panel Panel2;
protected System.Web.UI.WebControls.ImageButton cmdAceptar;

protected AdministracionCCSistema.clases.grid_listado objGrid =
new AdministracionCCSistema.clases.grid_listado();
protected AdministracionCCSistema.clases.cbl objCbl =
new AdministracionCCSistema.clases.cbl();
protected IGS.LN_AdministracionCC.LN_Consultas objCM = new
IGS.LN_AdministracionCC.LN_Consultas();
protected AdministracionCCSistema.clases.tablas objTabla = new
AdministracionCCSistema.clases.tablas();

protected ArrayList ACiudades = new ArrayList();
protected ArrayList AProductos = new ArrayList();
protected ArrayList ASegmentos = new ArrayList();
protected System.Web.UI.WebControls.CheckBox CheckBox1;
protected System.Web.UI.WebControls.CheckBox CheckBox2;
protected System.Web.UI.WebControls.CheckBox CheckBox3;
protected System.Web.UI.WebControls.DataGrid dgPlantilla;
protected System.Web.UI.WebControls.Panel Panel1;
protected System.Web.UI.WebControls.ImageButton Imagebutton2;
protected System.Web.UI.WebControls.ImageButton Imagebutton1;
protected System.Web.UI.WebControls.Label Label1;
protected bool llena;
protected System.Web.UI.WebControls.Label lblIncompleto;
protected System.Web.UI.WebControls.TextBox txtProductos1;
protected System.Web.UI.WebControls.TextBox txtSegmentos1;
protected System.Web.UI.WebControls.Label lblTitulo;
protected System.Web.UI.WebControls.TextBox txtCiudades1;
protected System.Web.UI.WebControls.ImageButton cmdGuardar;
protected System.Web.UI.WebControls.ImageButton cmdEliminar2;
protected System.Web.UI.WebControls.ImageButton cmdCancelar;
protected System.Web.UI.WebControls.ImageButton cmdActualizar;
protected System.Web.UI.WebControls.CustomValidator Customvalidator4;
protected System.Web.UI.WebControls.RequiredFieldValidator RVpundesde;
protected System.Web.UI.WebControls.TextBox txtPuntajeSobre;
protected System.Web.UI.WebControls.TextBox idplantilla1;

private void Page_Load(object sender, System.EventArgs e)
{
    if (!Page.IsPostBack)
    {
        this.lblTitulo.Text = objTabla.tabla_azul("DEFINICION DE
PLANTILLAS");
        this.llena_ciudad();
        this.llena_producto();
        this.llena_segmento();
        this.llena_grid();
    }
}
#region Web Form Designer generated code
override protected void OnInit(EventArgs e)
{
    //
    // CODEGEN: This call is required by the ASP.NET Web Form Designer.
    //

```

```

        InitializeComponent();
        base.OnInit(e);
    }

    ///<summary>
    /// Required method for Designer support - do not modify
    /// the contents of this method with the code editor.
    ///</summary>
    private void InitializeComponent()
    {
        this.cmdAbajoCiudad.Click += new
System.Web.UI.ImageClickEventHandler(this.cmdAbajoCiudad_Click);
        this.cmdArribaCiudad.Click += new
System.Web.UI.ImageClickEventHandler(this.cmdArribaCiudad_Click);
        this.cmdArribaProducto.Click += new
System.Web.UI.ImageClickEventHandler(this.cmdArribaProducto_Click);
        this.cmdAbajoProducto.Click += new
System.Web.UI.ImageClickEventHandler(this.cmdAbajoProducto_Click);
        this.cmdAbajoSegmento.Click += new
System.Web.UI.ImageClickEventHandler(this.cmdAbajoSegmento_Click);
        this.cmdArribaSegmento.Click += new
System.Web.UI.ImageClickEventHandler(this.cmdArribaSegmento_Click);
        this.CheckBox1.CheckedChanged += new
System.EventHandler(this.CheckBox1_CheckedChanged);
        this.CheckBox2.CheckedChanged += new
System.EventHandler(this.CheckBox2_CheckedChanged);
        this.CheckBox3.CheckedChanged += new
System.EventHandler(this.CheckBox3_CheckedChanged);
        this.chkCiudad.SelectedIndexChanged += new
System.EventHandler(this.chkCiudad_SelectedIndexChanged);
        this.cmdAceptar.Click += new
System.Web.UI.ImageClickEventHandler(this.cmdAceptar_Click);
        this.cmdGuardar.Click += new
System.Web.UI.ImageClickEventHandler(this.cmdGuardar_Click);
        this.cmdActualizar.Click += new
System.Web.UI.ImageClickEventHandler(this.cmdActualizar_Click);
        this.cmdCancelar.Click += new
System.Web.UI.ImageClickEventHandler(this.cmdCancelar_Click);
        this.cmdEliminar2.Click += new
System.Web.UI.ImageClickEventHandler(this.cmdEliminar2_Click);
        this.Imagebutton1.Click += new
System.Web.UI.ImageClickEventHandler(this.Imagebutton1_Click);
        this.Imagebutton2.Click += new
System.Web.UI.ImageClickEventHandler(this.Imagebutton2_Click);
        this.dgPlantilla.SelectedIndexChanged += new
System.EventHandler(this.dgPlantilla_SelectedIndexChanged);
        this.Load += new System.EventHandler(this.Page_Load);
    }
    #endregion

    private void cmdAbajoCiudad_Click(object sender, System.Web.UI.ImageClickEventArgs
e)
    {
        this.chkCiudad.Visible = true;
        this.cmdArribaCiudad.Visible = true;
        this.cmdAbajoCiudad.Visible = false;
    }

    private void cmdArribaCiudad_Click(object sender, System.Web.UI.ImageClickEventArgs
e)
    {
        this.chkCiudad.Visible = false;
        this.cmdArribaCiudad.Visible = false;
        this.cmdAbajoCiudad.Visible = true;
    }

```

```

privatevoid llena_ciudad()
{
    objCbl.cblControl = this.chkCiudad;
    objCbl.dvControl = objCM.Consulta_ciudad();
    objCbl.llena_cbl();
}
privatevoid llena_producto()
{
    objCbl.cblControl = this.chkProducto;
    objCbl.dvControl = objCM.Consulta_producto();
    objCbl.llena_cbl();
}
privatevoid llena_segmento()
{
    objCbl.cblControl = this.chkSegmento;
    objCbl.dvControl = objCM.Consulta_segmento();
    objCbl.llena_cbl();
}

privatevoid cmdAbajoProducto_Click(object sender,
System.Web.UI.ImageClickEventArgs e)
{
    this.chkProducto.Visible = true;
    this.cmdArribaProducto.Visible = true;
    this.cmdAbajoProducto.Visible = false;
}

privatevoid cmdArribaProducto_Click(object sender,
System.Web.UI.ImageClickEventArgs e)
{
    this.chkProducto.Visible = false;
    this.cmdArribaProducto.Visible = false;
    this.cmdAbajoProducto.Visible = true;
}

privatevoid cmdAbajoSegmento_Click(object sender,
System.Web.UI.ImageClickEventArgs e)
{
    this.chkSegmento.Visible = true;
    this.cmdArribaSegmento.Visible = true;
    this.cmdAbajoSegmento.Visible = false;
}

privatevoid cmdArribaSegmento_Click(object sender,
System.Web.UI.ImageClickEventArgs e)
{
    this.chkSegmento.Visible = false;
    this.cmdArribaSegmento.Visible = false;
    this.cmdAbajoSegmento.Visible = true;
}

privatevoid cmdAceptar_Click(object sender, System.Web.UI.ImageClickEventArgs e)
{
    this.txtCiudades.Text = barre_checkBox_todo(this.chkCiudad);
    this.txtProductos.Text = barre_checkBox_todo(this.chkProducto);
    this.txtSegmentos.Text = barre_checkBox_todo(this.chkSegmento);

    this.txtCiudades1.Text = barre_checkBox_letra(this.chkCiudad);
    this.txtProductos1.Text = barre_checkBox_letra(this.chkProducto);
    this.txtSegmentos1.Text = barre_checkBox_letra(this.chkSegmento);

    if (this.txtCiudades.Text.Trim()==" " || this.txtProductos.Text.Trim()==" " ||
this.txtSegmentos.Text.Trim()==" ")
    {

```

```

        this.lblIncompleto.Visible = true;
        this.cmdCancelar.Visible = true;
    }
    else
    {
        this.cmdCancelar.Visible = false;
        this.lblIncompleto.Visible = false;
        this.Panel1.Visible = false;
        this.cmdCancelar.Visible = true;
        if(cmdEliminar2.Visible)
            this.cmdActualizar.Visible = true;
        else
            this.cmdGuardar.Visible = true;
    }
}
private string barre_checkBox_letra(System.Web.UI.WebControls.CheckBoxList
checkboxG)
{
    string strArreglo = "";
    string palabra1 = " ";
    string palab1 = " ";
    string palab2 = " ";
    int x2 = 0;
    for (int cont=0; cont<checkboxG.Items.Count; cont++)
    {
        if (checkboxG.Items[cont].Selected == true)
        {
            palabra1 = "";
            palab1 = "";
            palab2 = "";
            x2 = 0;

            palabra1 =
Convert.ToString(checkboxG.Items[cont].Text).Trim();
            int vfin = palabra1.Length;

            for (int e=0; e < vfin; e++)
                if (palabra1.Substring(e,1)==" " && e != vfin)
                {
                    x2 = e+1;
                    break;
                }

            if (x2 == 0)
            {
                palab1 = palabra1.Substring(0,vfin);
                if (palab1.Length < 3)
                    strArreglo +=
palabra1.Substring(0,palab1.Length);

                else
                    strArreglo += palabra1.Substring(0,3);
            }
            else
            {
                palab1 = palabra1.Substring(0,x2-1);
                palab1 = palab1.Trim();
                palab2 = palabra1.Substring(x2,vfin-x2);
                palab2 = palab2.Trim();
                if (palab1.Length > 2)
                    palab1 = palab1.Substring(0,3);

                else
                {
                    palab1 =
palab1.Substring(0,palab2.Length);

                }
                if (palab2.Length > 2)
                    palab2 = palab2.Substring(0,3);
            }
        }
    }
}

```

```

        else
        {
            palab2 =
palab2.Substring(0,palab2.Length);
        }
        strArreglo += palab1+palab2;
    }
    strArreglo = strArreglo + "-";
}
}
int longui = strArreglo.Length;
if (longui > 0)
    strArreglo = strArreglo.Substring(0,longui-1);
return strArreglo;
}

privatevoid CheckBox1_CheckedChanged(object sender, System.EventArgs e)
{
    this.llena_vacia_CheckBox(this.chkCiudad,this.CheckBox1);
}
privatevoid CheckBox2_CheckedChanged(object sender, System.EventArgs e)
{
    this.llena_vacia_CheckBox(this.chkProducto,this.CheckBox2);
}
privatevoid CheckBox3_CheckedChanged(object sender, System.EventArgs e)
{
    this.llena_vacia_CheckBox(this.chkSegmento,this.CheckBox3);
}

privatevoid llena_vacia_CheckBox(System.Web.UI.WebControls.CheckBoxList
ChecListG, System.Web.UI.WebControls.CheckBox ChecBoxG)
{
    if (ChecBoxG.Checked == true)
        for (int cont=0; cont < ChecListG.Items.Count; cont++)
            ChecListG.Items[cont].Selected = true;
    else
        for (int cont=0; cont < ChecListG.Items.Count; cont++)
            ChecListG.Items[cont].Selected = false;
}
privatevoid llena_grid()
{
    objGrid.dgControl = this.dgPlantilla;
    objGrid.dvControl = objCM.Consulta_plantilla();
    if (objGrid.dvControl.Count > 0)
        objGrid.grid();
}

privatevoid cmdGuardar_Click(object sender, System.Web.UI.ImageClickEventArgs e)
{
    objCM.Ingresar_plantilla(Convert.ToString(txtCiudades.Text.Trim()),Convert.ToString(txtCiudades
1.Text.Trim()),Convert.ToString(txtProductos.Text.Trim()),Convert.ToString(txtProductos1.Text.Trim()),Con
vert.ToString(txtSegmentos.Text.Trim()),Convert.ToString(txtSegmentos1.Text.Trim()),Convert.ToInt16(thi
s.txtPuntajeSobre.Text));
    llena_grid();
    this.cmdGuardar.Visible = false;
    this.cmdEliminar2.Visible = false;
    this.txtPuntajeSobre.Text = "";
}

privatevoid cmdEliminar2_Click(object sender, System.Web.UI.ImageClickEventArgs e)
{
    this.cmdActualizar.Visible = false;
    this.cmdCancelar.Visible = false;
    this.cmdEliminar2.Visible = false;
    this.Panel1.Visible = true;
}

```

```

}

privatevoid Imagebutton2_Click(object sender, System.Web.UI.ImageClickEventArgs e)
{
    this.Panel1.Visible = false;
    this.idplantilla1.Text = "";
}

privatevoid Imagebutton1_Click(object sender, System.Web.UI.ImageClickEventArgs e)
{
    objCM.Elimina_plantilla(Convert.ToInt32(this.idplantilla1.Text));
    llena_grid();
    this.Panel1.Visible = false;
    this.idplantilla1.Text = "";
}

privatevoid dgPlantilla_SelectedIndexChanged(object sender, System.EventArgs e)
{
    this.idplantilla1.Text = this.dgPlantilla.SelectedItem.Cells[0].Text;
    string ciudad1 = this.dgPlantilla.SelectedItem.Cells[1].Text.Trim();
    this.verifica_llena_chechList(ciudad1,this.chkCiudad);
    string producto1 = this.dgPlantilla.SelectedItem.Cells[2].Text.Trim();
    this.verifica_llena_chechList(producto1,this.chkProducto);
    string segmento1 = this.dgPlantilla.SelectedItem.Cells[3].Text.Trim();
    this.verifica_llena_chechList(segmento1,this.chkSegmento);
    this.cmdCancelar.Visible = true;

    this.txtCiudades.Text = ciudad1;
    this.txtProductos.Text = producto1;
    this.txtSegmentos.Text = segmento1;

    this.txtCiudades1.Text = this.dgPlantilla.SelectedItem.Cells[4].Text.Trim();
    this.txtProductos1.Text = this.dgPlantilla.SelectedItem.Cells[5].Text.Trim();
    this.txtSegmentos1.Text = this.dgPlantilla.SelectedItem.Cells[6].Text.Trim();
    this.txtPuntajeSobre.Text =
Convert.ToString(this.dgPlantilla.SelectedItem.Cells[7].Text.Trim());

    this.cmdGuardar.Visible = false;
    this.cmdEliminar2.Visible = true;
    this.cmdActualizar.Visible = false;
    this.cmdCancelar.Visible = false;
}

privatevoid verifica_llena_chechList(string
ciudad1, System.Web.UI.WebControls.CheckBoxList checListG)
{
    ciudad1 = ciudad1.Trim()+"-";
    for (int cont=0; cont < checListG.Items.Count; cont++)
        checListG.Items[cont].Selected = false;
    string a = "";
    for (int i=0; i<ciudad1.Length; i++)
        if(ciudad1.Substring(i,1)!="-")
            a += ciudad1.Substring(i,1);
        else/***** marcar CheckBox
            for (int cont=0; cont < checListG.Items.Count; cont++)

if(a==Convert.ToString(checListG.Items[cont].Value).Trim())
    {
        checListG.Items[cont].Selected = true;
        a = "";
        break;
    }
}

privatevoid cmdCancelar_Click(object sender, System.Web.UI.ImageClickEventArgs e)
{
    this.lblIncompleto.Visible = false;
    this.cmdActualizar.Visible = false;
    this.cmdCancelar.Visible = false;
}

```

```

        this.cmdEliminar2.Visible = false;
    }

checkboxG)
privatestring barre_checkBox_todo(System.Web.UI.WebControls.CheckBoxList
{
    string strArreglo = "";
    for (int cont=0; cont<checkboxG.Items.Count; cont++)
    {
        if (checkboxG.Items[cont].Selected == true)
            strArreglo +=
Convert.ToString(checkboxG.Items[cont].Value).Trim()+"-";
    }
    int longui = strArreglo.Length;
    if (longui > 0 )
        strArreglo = strArreglo.Substring(0,longui-1);
    return strArreglo;
}

privatevoid cmdActualizar_Click(object sender, System.Web.UI.ImageClickEventArgs e)
{
    objCM.Modifica_plantilla(Convert.ToInt32(this.idplantilla1.Text),Convert.ToString(txtCiudades.Text.Trim()),Convert.ToString(txtCiudades1.Text.Trim()),Convert.ToString(txtProductos.Text.Trim()),Convert.ToString(txtProductos1.Text.Trim()),Convert.ToString(txtSegmentos.Text.Trim()),Convert.ToString(txtSegmentos1.Text.Trim()),Convert.ToInt16(this.txtPuntajeSobre.Text));
    llena_grid();
    this.cmdActualizar.Visible = false;
    this.cmdCancelar.Visible = false;
    this.cmdEliminar2.Visible = false;
    this.idplantilla1.Text = "";
    this.txtPuntajeSobre.Text = "";
}

privatevoid chkCiudad_SelectedIndexChanged(object sender, System.EventArgs e)
{
}
}
}

```

6.3. ANEXO 3: MANUAL DE INSTALACIÓN

<p>Introducción</p> <p>El Prototipo Efectiva es orientado a la Web y algunos pasos para su configuración se los realiza manualmente. Este documento presenta una forma de realizar la instalación y puesta a punto de manera sencilla.</p>	
<p>Requerimientos</p>	<p>Debido a que el programa es orientado a Web, es necesario que el computador donde será instalado el programa tenga conexión a Internet.</p> <p>Además, para lograr una instalación exitosa se deberá tener previamente instalado en el o los servidores los siguientes programas cargados en el orden que se describen.</p> <p>SERVIDOR DE BASE DE DATOS</p> <ul style="list-style-type: none"> ■ Windows Server 2008 o superior ■ Microsoft SQL Server 2012 o superior <p>SERVIDOR DE APLICACIONES</p> <ul style="list-style-type: none"> ■ Windows Server 2008 o Superior ■ .Net Framework 3.5.1 o superior ■ Internet Information Server 7 o superior <p>O si se desea puede tener un único servidor que contemple ambas funcionalidades:</p> <p>SERVIDOR DE BASE DE DATOS Y APLICACIONES</p> <ul style="list-style-type: none"> ■ Windows 2008 Server o Superior ■ Internet Information Server 7 o superior ■ .Net Framework 3.5.1 o superior ■ Microsoft SQL Server 2012 o superior

	<p>Es importante que el Internet Information Server IIS esté configurado antes de correr el programa, para lo cual debemos seguir los siguientes pasos:</p> <p>En el Server Manager, ubicarse en Internet Information Server (IIS Manager) que se encuentra dentro Roles y de Web Server (IIS). Ver <i>Pantalla 1</i>.</p> <p>En las características del IIS, escoger el ícono Authentication y habilitar Anonymous Authentication, cambiando de disable a enable. Ver <i>Pantalla 2</i>.</p> <p>A continuación volver a la ventana del Server Manager y dentro de Application Pools hacer doble clic en Default AppPool para ingresar a la ventana de edición.</p> <p>En ésta ventalla seleccionar los siguientes parámetros:</p> <p>Name: Default AppPool .Net Framework Version: .Net Framework v 4.0 Managed Pipeline Mode: Integrated Ver <i>Pantalla 3</i>.</p> <p>Hacer clic derecho en Default Application Pool que fue creada y seleccionar la opción Set Application Pool Defaults.</p> <p>Aparece una nueva ventana y nos dirigimos al campo Identity de la sección Process Model.</p> <p>Finalmente en el campo Built in account escogemos LocalSystem. Ver <i>Pantalla 4</i>.</p> <p>Y así queda configurado el Internet Information Server.</p>
--	---

		<p>CLIENTE</p> <p>El cliente únicamente requiere el siguiente software:</p> <ul style="list-style-type: none"> ■ Una versión de Windows actualizada. Pueden ser las versiones 7 u 8. O el mismo Windows Server 2008 o superior. ■ Un requisito fundamental es tener instalado un navegador Web que puede ser: Internet Explorer, Google Chrome, Firefox etc. De preferencia Google Chrome debido a que permite una mejor presentación de las pantallas. 	
1 Instalar Sitio Web		Indica el proceso de instalar y subir el sitio Web	
	Proc.	Actividad	Sig. Proc.
	10	Si se instala desde el CD de instalación, insertarlo en unidad respectiva para lectura y debe aparecer la pantalla del menú interactivo.	20
	20	<p>En el menú seleccionar la opción Instalación y aparece la pantalla del menú Instalación, donde seleccionamos Instalación programa Efectiva, para que se ejecute el archivo setup.exe el cual hace que se ejecute el programa de instalación EfectivaWebInstall.</p> <p>Otra forma es copiar al computador donde será instalado el programa, los archivos setup.exe y EfectivaWebInstall.msi que es el archivo instalador de Windows generado al crear el script en VisualStudio y ejecutar setup.exe</p>	30

	30	<p>Al ejecutar setup.exe, se desplegará el wizard de instalación del sitio web.</p> <p>La primera pantalla es de Bienvenida. Presionar botón Next. Ver <i>Pantalla 5</i>.</p>	40 50
	40	<p>En caso de que el programa de instalación constate que ya existe una versión instalada aparece una pantalla en la que solicita que seleccione si va a reparar o a remover el programa. Escoger la opción Repair y continuar con el proceso presionando el Botón Finish, o puede cancelar y salir del proceso presionando el botón Finish. Ver <i>Pantalla 6</i>.</p>	50 FIN
	50	<p>Aparece la pantalla para seleccionar el destino de la instalación. Aquí confirmamos que se encuentren los siguientes parámetros como se indica a continuación:</p> <p>Site: Default WebSite Virtual directory: Efectiva Application Pool: Default AppPool</p> <p>Hacer clic en Next para continuar. Ver <i>Pantalla 7</i>.</p>	60
	60	<p>Confirmamos el proceso de instalación haciendo clic en Next.</p> <p>A continuación presenta la pantalla para confirmar el inicio de la Instalación. Aceptar presionando Next. Ver <i>Pantalla 8</i>.</p>	70
	70	<p>En la siguiente se pantalla muestra el proceso de instalación. Esperar hasta que se complete y</p>	80

	80	presionar botón Next. Ver <i>Pantalla 9</i> . Finalmente, aparece la ventana donde se indica que la Instalación se completó. Para finalizar este proceso hacemos clic en Close. Ver <i>Pantalla 10</i> .	FIN
2 Personalizar Aplicación	Se realizan algunas modificaciones necesarias para el correcto funcionamiento del sitio.		
	10	Ir al Explorador de Windows y abrir la carpeta C:\Inetpub\wwwroot\Efectiva\	20
	20	Hacer clic con el botón derecho del mouse, para activar el menú emergente, de la carpeta Efectiva, y elegir la opción Propiedades. Aparece la ventana de Propiedades de la carpeta. Debemos hacer clic sobre la pestaña General y desactivar la opción Atributos: Solo Lectura.	30
	30	En la misma ventana hacemos clic en la pestaña Seguridad o Security, y presionamos el botón Editar y aparece una ventana donde añadimos con el botón Añadir o Add el usuario Everyone, y presionamos el botón Aceptar.	40
	40	Volvemos a la pestaña Security donde nos despliegan los permisos para el usuario Everyone, hacemos clic en el casillero Full Control o Control Total. Para finalizar este paso, hacemos clic sobre el botón Aceptar.	50
	50	Volver al Explorador de Windows y abrir la carpeta	60

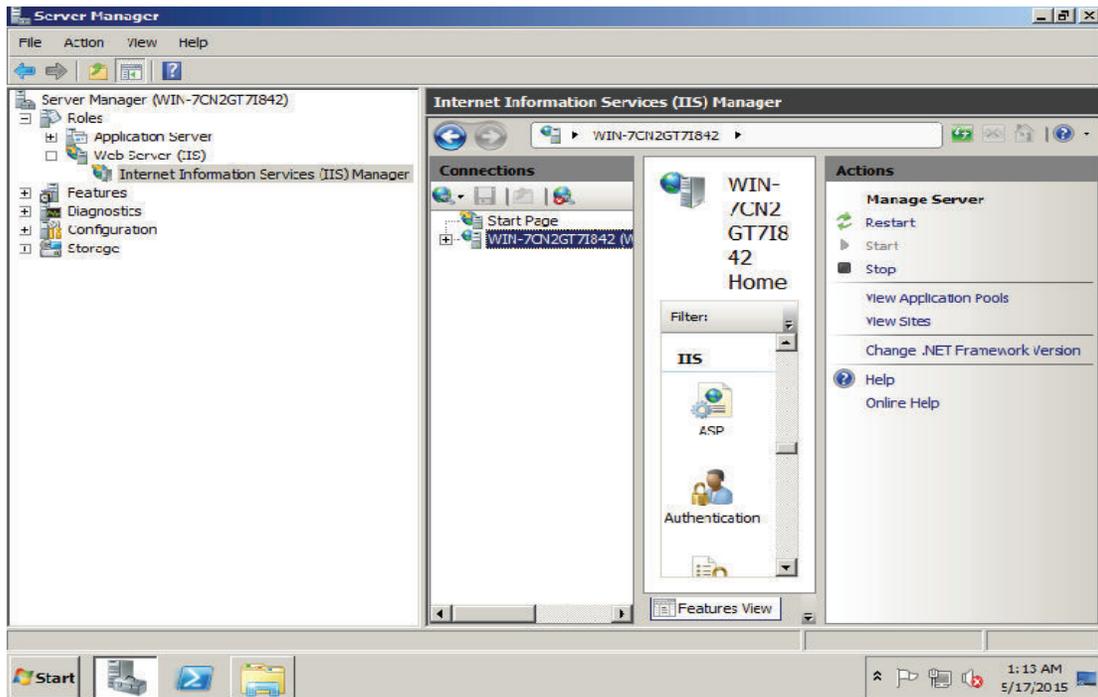
		C:\inetpub\wwwroot\Efectiva\	
60		Hacer clic con el botón derecho del mouse, para activar el menú emergente, del archivo Web.config, y escoger la opción Editar para abrir el archivo en el Editor de Texto Notepad (Block de Notas) o puede abrirlo con un editor similar.	70
70		<p>En el Bloque <fuentesDatos> <Administracioncc> se deberá reemplazar los siguientes parámetros:</p> <p><add key="DataSource" value=".SQLEXPRESS"/></p> <p><add key="userId" value="sa"/>, (donde sa es el nombre de usuario de la base de datos proporcionado por el Administrador).</p> <p><add key="password" value="Efectiva01"/>, (donde Efectiva01 es la contraseña proporcionada por el Administrador).</p>	80
80		Finalizado este proceso se procederá a guardar el archivo reemplazando el archivo original. Y salir del Editor.	FIN

3 Crear e Instalar Base de Datos		Aquí se crear la base de datos con sus respectivas tablas.	
	10	Para realizar el proceso de creación de la Base de Datos, usted debe conocer el nombre del servidor donde está instalando esta aplicación, además deberá poseer un nombre de usuario y contraseña para acceder al SQL Server 2012. Si no los tiene favor no continuar con este proceso y contactarse con su administrador local de base de datos para que le facilite esta información.	20 FIN
	20	Si se instala desde el CD de instalación, insertarlo en unidad respectiva para lectura y debe aparecer la pantalla del menú interactivo.	30
	30	En el menú seleccionar la opción Instalación y aparece la pantalla del menú Instalación, donde seleccionamos Crear Base de Datos, para que se ejecute el archivo crearbd.bat el cual hace que se ejecute el script efectiva.sql que copia los datos de las tablas y genera la Base de datos AdministracionCC. En este caso el archivo crearbd.bat ya está editado como se indica en los pasos 50, 60 y 70.	70
	40	Otra forma de instalar es creando en el directorio raíz la carpeta C:\EFECTIVA y copiar los archivos crearbd.bat y efectiva.sql para ejecutarlos desde el disco duro del computador.	50
	50	A continuación hacer clic en el botón derecho del	60

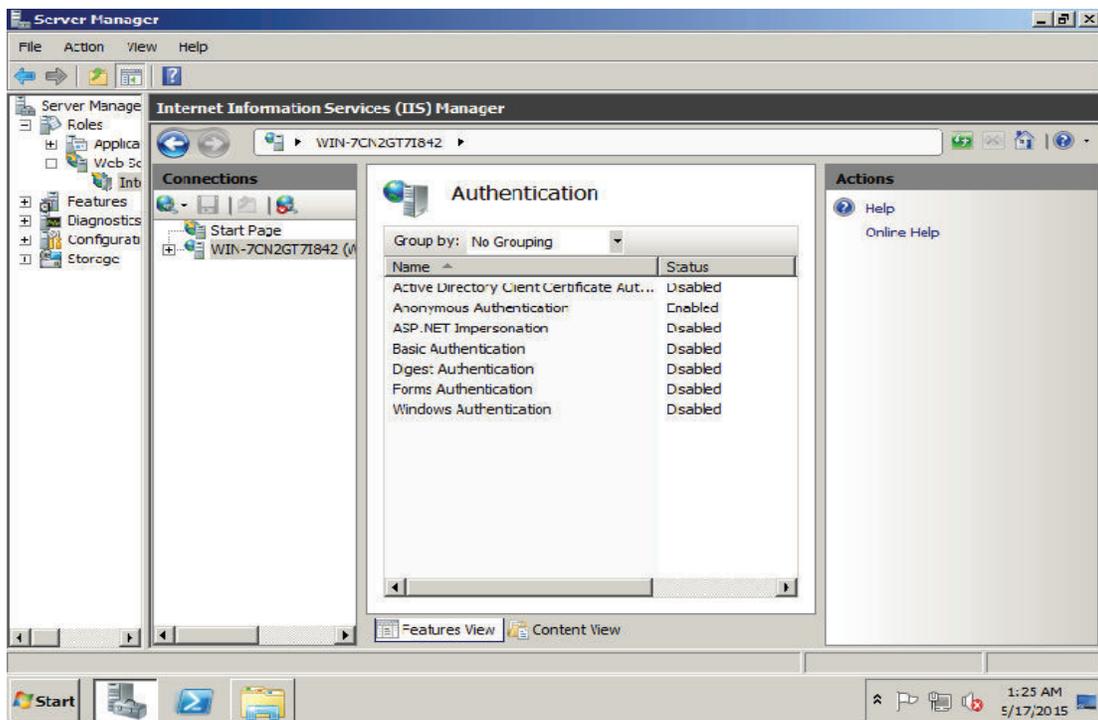
	<p>mouse, para activar el menú emergente, del archivo crearb.bat, y elegir haciendo clic sobre Editar.</p> <p>Se abrirá automáticamente el Editor de Texto Notepad (Bloc de Notas).</p>	
60	<p>Confirmar que la línea de comando esté como se indica a continuación o editarla.</p> <pre>sqlcmd -i efectiva.sql -S .\SQLEXPRESS -U sa -P Efectiva01</pre> <p>Donde: “./SQLEXPRESS” es el nombre del servidor de base de datos. “sa” es el nombre de usuario de la base de datos proporcionada por el Administrador. “Efectiva01” es la contraseña para acceder a SQL proporcionada por el Administrador.</p> <p>Tomar en cuenta que estos parámetros inician con: -U (usuario), -P (contraseña) y -S (dirección) que no deberán ser reemplazados ni borrados.</p>	70
70	<p>Finalizado este proceso se procederá a guardar el archivo reemplazando el archivo original. Y salir del editor.</p>	80
80	<p>Realizar doble clic sobre el archivo modificado, crearb.bat, con lo que dará inicio al proceso de creación de la base de datos, donde se cuenta con</p>	FIN

		<p>datos de prueba, tener paciencia, ya que puede demorar varios minutos y aparecerá una pantalla negra mientras dure este proceso.</p> <p>Nota: Para el correcto funcionamiento del prototipo y pruebas, las tablas generadas en la creación de la Base de Datos contienen datos iniciales debido a que éste prototipo es un módulo del Sistema General del Call Center de donde se obtiene la información de las Gestiones realizadas por los Agentes telefónicos del Call Center.</p>	
4. Acceso al prototipo		Se define como un cliente puede acceder al servidor web.	
	10	<p>Para ingresar al sistema (Sitio Web), debe ingresar al Browser y en la dirección digitar:</p> <p>http://nombreservidor/Efectiva/Sistema/ingreso.aspx</p> <p>Donde:</p> <p>Nombreservidor.- corresponde al nombre del servidor web o a su vez se puede poner la dirección IP.</p> <p>Para el óptimo manejo del prototipo favor consultar el manual de usuario.</p>	FIN

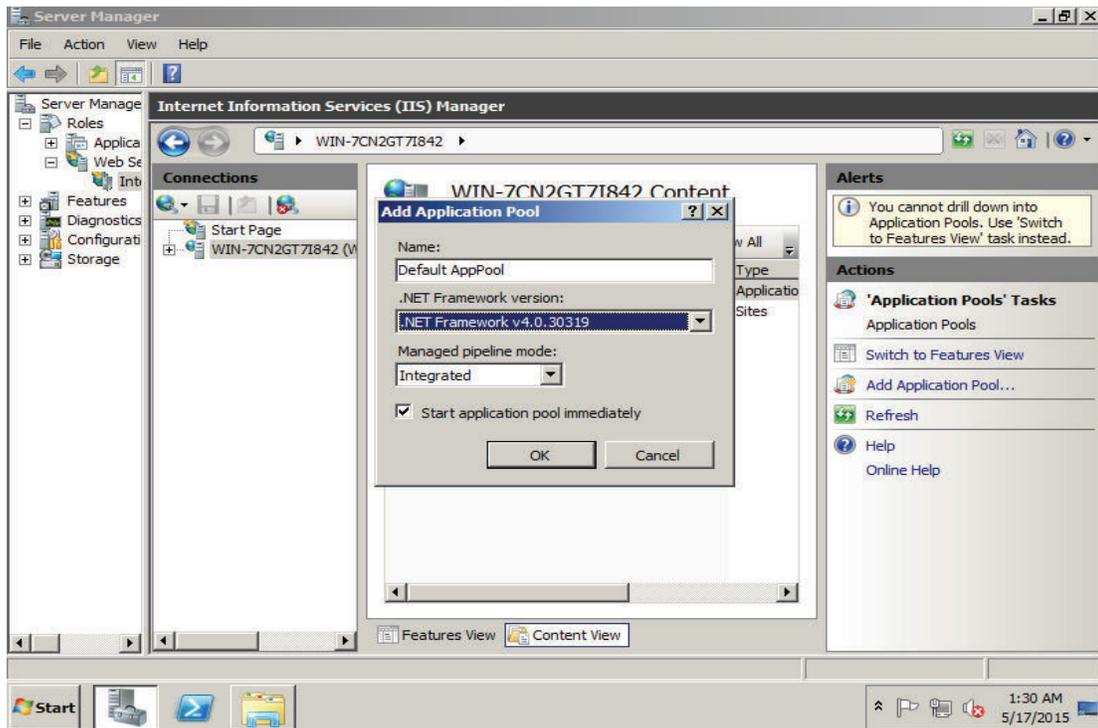
Pantalla 1 Configuración del Internet Information Service IIS



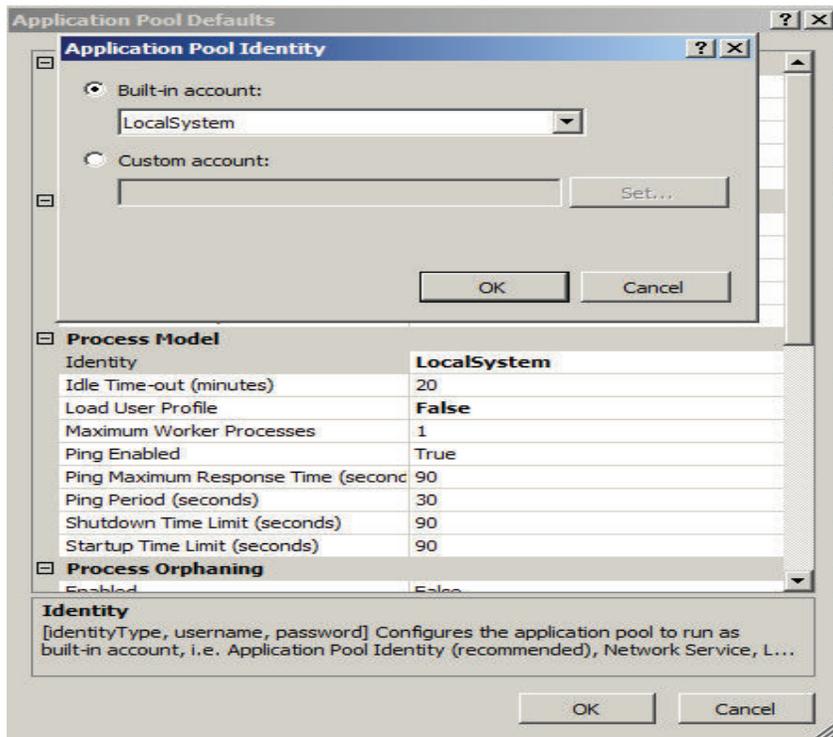
Pantalla 2 Configuración del Internet Information Service IIS



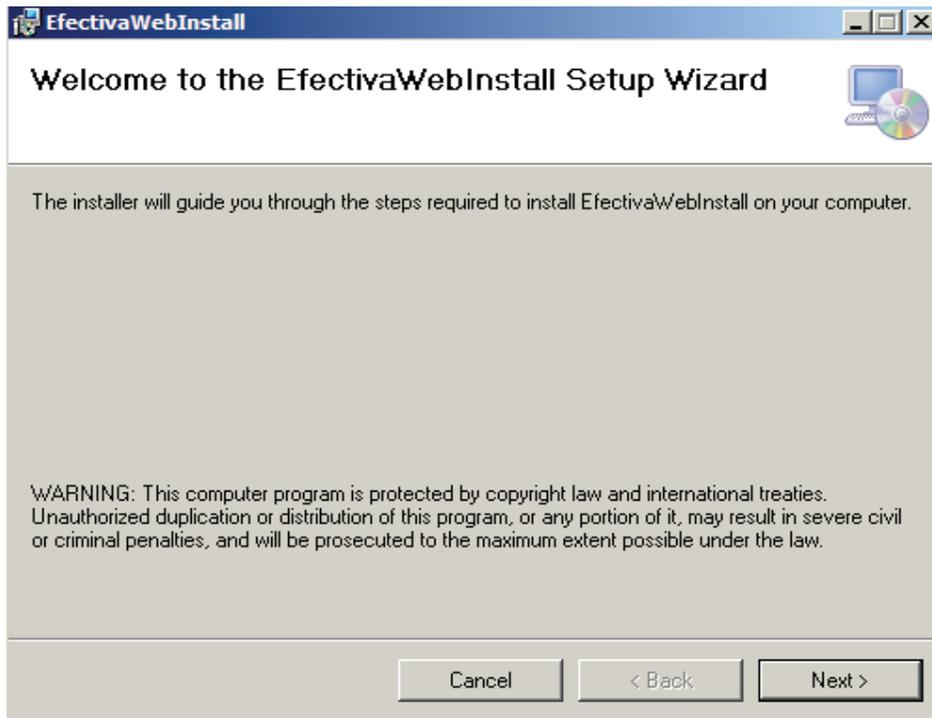
Pantalla 3 Configuración del Internet Information Service IIS



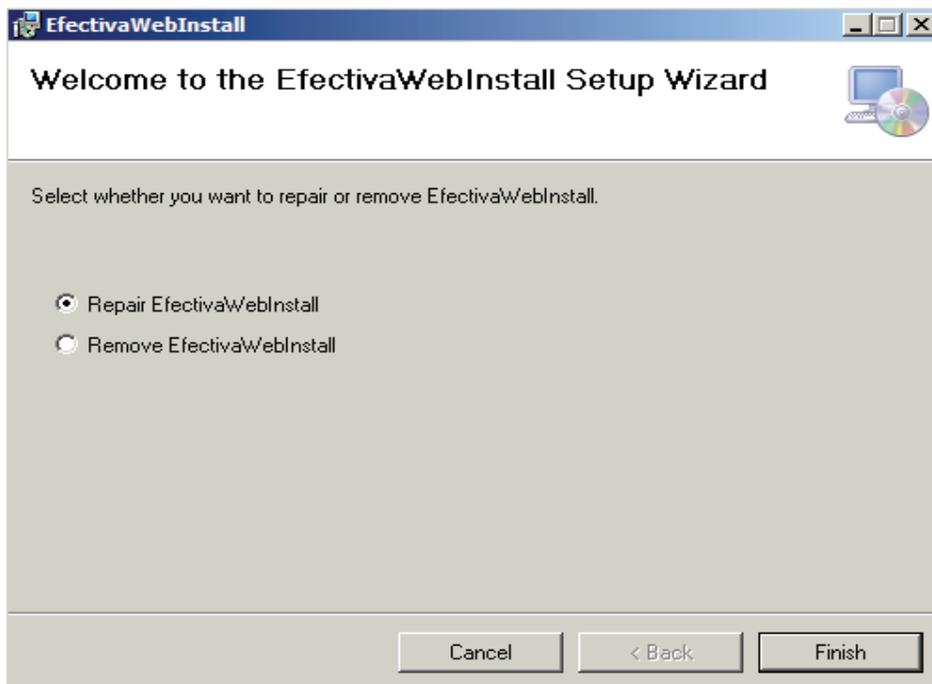
Pantalla 4 Configuración del Internet Information Service IIS



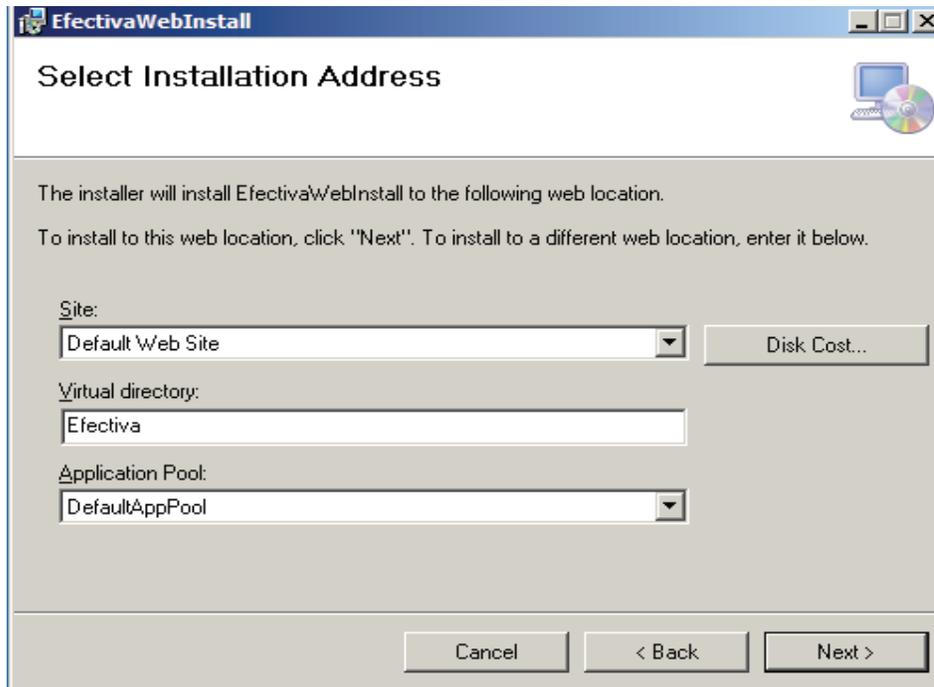
Pantalla 5 Instalación del programa Efectiva



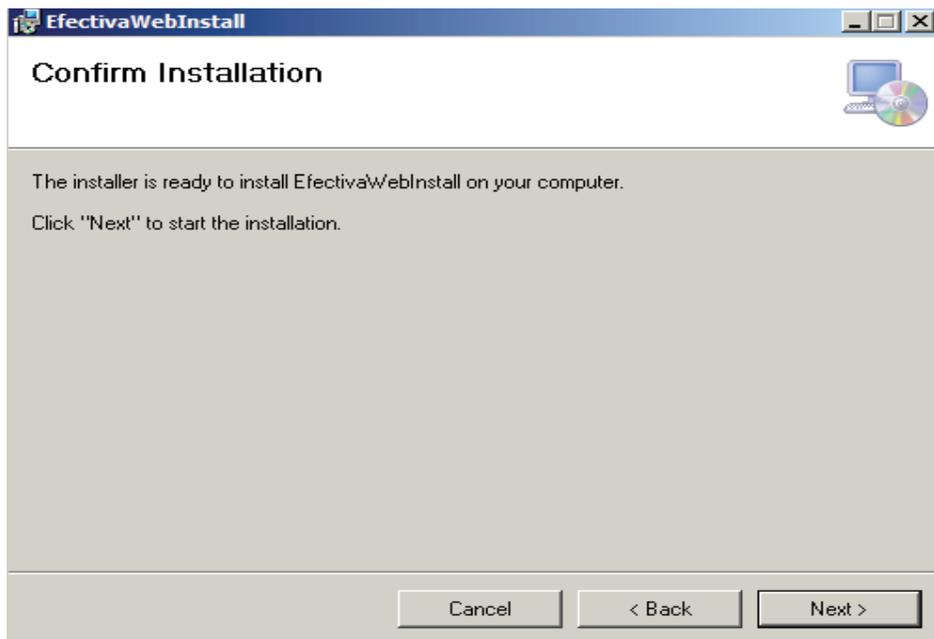
Pantalla 6 Instalación del programa Efectiva



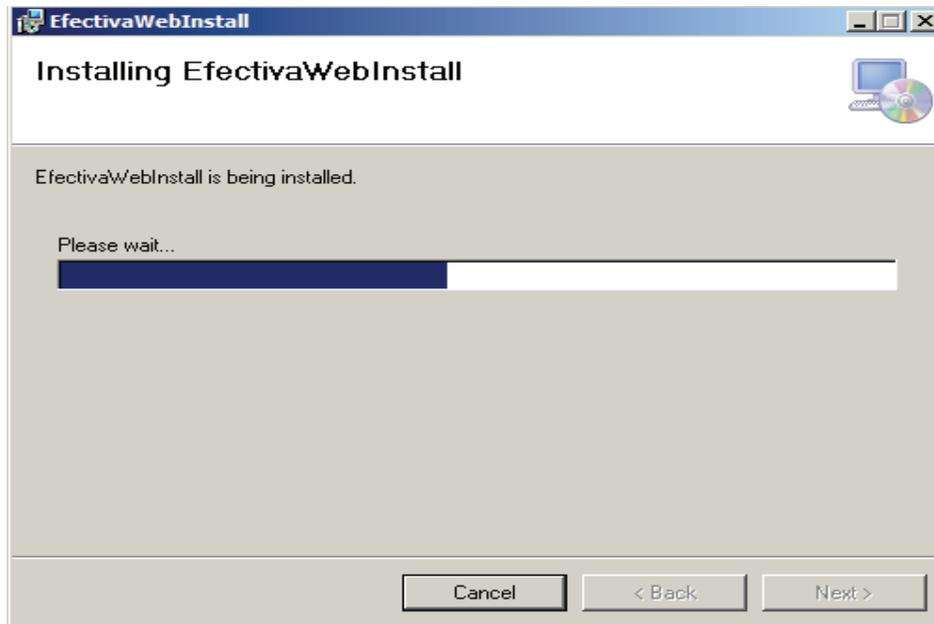
Pantalla 7 Instalación del programa Efectiva



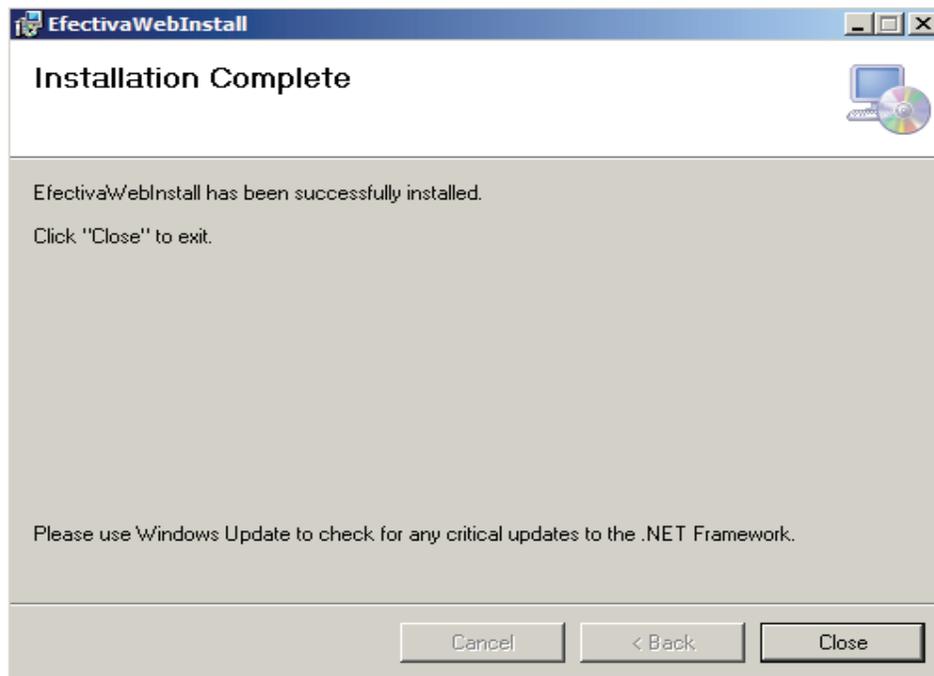
Pantalla 8 Instalación del programa Efectiva



Pantalla 9 Instalación del programa Efectiva



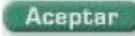
Pantalla10 Instalación del programa Efectiva



6.4. ANEXO 4: MANUAL DE USUARIO

Introducción

Este prototipo permite evaluar la calidad de gestión de los agentes telefónico a través del módulo de Auditoría de Gestión, así como también los resultados de productividad en el módulo de Supervisión EWM.

<p>Ingreso al prototipo</p>	<p>Para ingresar al prototipo debe acceder a un navegador Web y en el Browser digitar lo siguiente:</p> <p><code>http://localhost/Efectiva/Sistema/ingreso.aspx</code></p> <p>Luego se desplegará la pantalla de inicio (Ver Pantalla 0), en la cual deberá ingresar en los cuadros de textos el usuario y su respectiva contraseña proporcionados por el Administrador, y deberá presionar el botón  para acceder al menú principal.</p>
<p>Menú Principal</p>	<p>En esta pantalla (Ver Pantalla 1).se muestran los módulos de los cuales se conforma el prototipo, estos son:</p> <p style="text-align: center;">AUDITORIA DE GESTIÓN</p> <p style="text-align: center;">SUPERVISIÓN EWM</p> <p>Los que contienen dentro de sí diversas opciones las que se explicarán en detalle dentro de este manual.</p>

1. Módulo de Gestión	Auditoria	Este módulo sirve para evaluar la calidad de gestión de los agentes telefónicos (Ver Pantalla 2).		
1.1. Menú Plantilla 		Permite la administración de una plantilla de evaluación, es decir la creación y modificación de plantillas.		
1.1.1. Definición de Plantillas	Proc. 10 20	<p>Esta opción permite la creación y/o consulta de una plantilla de evaluación.</p> <p>Actividad</p> <p>Al escoger la opción Definición de Plantillas se mostrará en pantalla una lista con las plantillas que fueron creadas anteriormente.</p> <p>Creación de Plantilla: (Ver Pantalla 3).</p> <p>Para crear una plantilla se debe escoger en el combo  la o las ciudades sobre la(s) cual(es) se aplicará la plantilla, de igual forma se lo hace con empresa-producto y el segmento en el combo respectivo.</p> <p> </p> <p>Además de esto se debe ingresar el puntaje máximo de dicha plantilla de evaluación. Presionamos  para obtener los datos ingresados a plantilla, si se desea grabar la plantilla presione  y</p>	Respons Supervisor o Auditor de Gestión Supervisor o Auditor de Gestión	Sig. Proc. 20, 30, 40 FIN, 30, 40

	30	<p>quedará creada la pantalla caso contrario, presione Cancelar</p> <p>Modificación de Plantilla: (Ver Pantalla 4).</p> <p>En el caso de querer modificar una de las plantillas existentes se debe presionar el botón Restaurar de la plantilla que se desea modificar y que se visualiza en la lista indicada en el Proc.10, luego de lo cual se desplegarán en la pantalla los datos de la plantilla tales como: ciudad, empresa-producto, segmento y puntaje máximo, los cuales pueden ser modificados presionar el botón Visualizar para observar como quedarían las modificaciones realizadas y si está de acuerdo con los cambios presionar el botón Actualizar para grabar las modificaciones hechas en la plantilla, caso contrario presionar el botón Cancelar, con lo que no se guardarán los cambios realizados.</p>	Supervisor o Auditor de Gestión	FIN, 20, 40
	40	<p>Eliminación de Plantilla: (Ver Pantalla 4).</p> <p>En el caso de querer eliminar una de las plantillas existentes se debe presionar el botón Restaurar, de la plantilla que se desea modificar y que se visualiza en la lista indicada en el</p>	Supervisor o Auditor de Gestión	FIN, 20, 30

	30	<p>puntaje, el cual permitirá posteriormente el cálculo de la calificación en la evaluación realizada al agente telefónico.</p> <p>Puede ingresar cualquier cantidad de preguntas.</p> <p><u>Modificación de Preguntas:</u> (Ver Pantalla 7).</p> <p>En el caso de que desee modificar las preguntas, debe presionar el botón Seleccionar en la lista de plantillas que se explica en Proc.10, donde desplegará las preguntas disponibles para la plantilla seleccionada. Presione el botón Seleccionar sobre la pregunta que desea modificar y le permitirá cambiar los datos de dicha pregunta, si desea guardar los cambios efectuados presione el botón Actualizar y las modificaciones serán grabadas, caso contrario presione el botón Cancelar.</p> <p>También es posible modificar las posibles respuestas de una pregunta, para lo cual se presiona el botón Seleccionar junto a la pregunta que se desea modificar y se despliegan las posibles respuestas. En el caso de que el tipo de respuesta es “Si” o “No” se puede modificar el puntaje.</p>	Supervisor o Auditor de Gestión	FIN, 20, 40
--	----	--	---------------------------------	-------------

	40	<p>En el caso de ser una pregunta de Respuesta Múltiple o Rango Numérico se presiona el botón Restaurar para modificar las respuestas y puntajes. Si desea guardar los cambios efectuados presione el botón Actualizar y las modificaciones serán grabadas, caso contrario presione el botón Cancelar.</p> <p><u>Eliminación de Preguntas:</u> (Ver Pantalla 7).</p> <p>Para eliminar una pregunta, debe presionar el botón Seleccionar en la lista de plantillas que se explica en Proc.10, donde desplegará las preguntas disponibles para la plantilla seleccionada. Presione el botón Seleccionar sobre la pregunta que desea eliminar y le mostrará en pantalla los datos de la pregunta y si desea eliminarla presione el botón Eliminar, con lo cual se eliminará la pregunta y sus posibles respuestas.</p> <p>También es posible eliminar las posibles respuestas de una pregunta de Respuesta Múltiple o Rango Numérico para lo cual se presiona el botón Seleccionar junto a la pregunta que se desea modificar y se</p>	Supervisor o Auditor de Gestión	FIN, 20, 30
--	----	---	---------------------------------	-------------

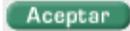
	20	<p>de las evaluaciones. Estos puede ser:</p> <ul style="list-style-type: none"> - Rango de Fechas.- debemos indicar desde que fecha y hasta que fecha deseamos la consulta. - Nombre del Auditor.- es el nombre del auditor que realizó la evaluación. Se puede elegir Todos los Auditores. - Recurso.- es el nombre del agente telefónico de quién se le realizó la evaluación. Se puede elegir Todos los recursos. - Puntaje Total.- Se escoge el rango de los puntajes que se desea consultar y puede ser de 0 hasta 1000. <p><u>Generación de consulta:</u> (Ver Pantalla 11).</p> <p>Luego presione Aceptar y nos despliega en la pantalla la fecha, hora, el nombre del Auditor, el Recurso (nombre agente telefónico) y la puntuación total, de todas las evaluaciones que cumplen con los parámetros seleccionados en el Proc.10.</p> <p>Si deseamos obtener el detalle de una evaluación presione Seleccionar y nos enseña los datos de: nombre del Auditor, el Recurso (nombre agente telefónico), la puntuación total, fecha y hora de la evaluación, y</p>	Supervisor o Auditor de Gestión	FIN
--	----	---	---------------------------------	-----

	FIN	<p>además se listan las preguntas realizadas con su respectiva respuesta y puntaje. (Ver Pantalla 12).</p> <p>Los 2 tipos de Consulta, general y detallada pueden dirigirse a impresora.</p>		
1.2.3. Opción Estadística de Evaluación	10	<p>Esta opción nos ayuda a obtener los resultados de las evaluaciones realizadas a los Agentes Telefónicos.</p> <p><u>Escoger parámetros de consulta:</u> (Ver Pantalla 13).</p> <p>Para realizar las estadísticas procedemos a llenar los siguientes parámetros:</p> <p>-Definición.- en este parámetro se encuentran dos opciones que se puede elegir: por puntaje o por recurso.</p> <p>-Tamaño de Gráfica.- permite indicar el tamaño que se desea presentar en la pantalla.</p> <p>-Fecha Inicial.- es la fecha inicial del rango de fecha que se desea consultar.</p> <p>-Fecha Final.- es la fecha final del rango de fecha que se desea consultar.</p> <p>-Elija el tipo de gráfico.- la representación gráfica puede ser de</p>	Supervisor o Auditor de Gestión	20

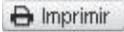
	20	<p>tipo en barras, circular o líneas.</p> <p>-Tipo de gráfico.- este tipo de gráfico puede ser columna, barra y área</p> <p>-Auditor.- es el nombre del auditor que realiza las evaluaciones de las cuales se obtendrá la estadística.</p> <p>-Tipo de Gráfica.- Sirve para seleccionar como desea que la representación gráfica, puede ser 3D o normal.</p> <p><u>Generar consulta:</u> (Ver Pantalla 13).</p> <p>Luego de haber indicado todos los parámetros presionamos  y se despliega en la pantalla un reporte donde se encuentra la referencia, detalle que indica los rangos de puntajes y valor que indica el número de agentes que obtuvieron el puntaje dentro de ese rango, y aparece también el respectivo gráfico.</p> <p>El resultado de la estadística y su gráfico pueden dirigirse a impresora.</p>	Supervisor o Auditor de Gestión	FIN
2. Módulo EWM		Nos permite supervisar la productividad de los agentes telefónicos.		
2.1. Indicadores Clave 	Proc.	<p>Actividad</p> <p>Permite la administración de los indicadores clave, creación,</p>	Respons.	Sig. Proc

		modificación y eliminación.		
2.1.1. Mantenimiento Indicadores Clave	10	<p>Crear Indicador Clave: (Ver Pantalla 14). Para crear un indicador clave se debe presionar el botón Ingresar y luego se debe escoger los parámetros necesarios para aplicar los indicadores, en los combos:</p> <p>CIUDAD ▲ PRODUCTO ▼ SEGMENTO ▼</p> <p>Luego presione el botón Aceptar e Ingresar Key Indicators, para crear los indicadores clave</p> <p>Aparece una pantalla que permite Seleccionar cada uno de los 4 indicadores clave para ingresar el valor del peso y el valor de óptimo de cada uno de ellos, los cuales no pueden exceder de 100:</p> <ul style="list-style-type: none"> ○ Tasa de Conexiones (Hit Rate) ○ Tasa de Contactos (Contact Rate) ○ Tasa de Compromisos (Promise Rate) ○ Tasa de Eficiencia <p>La suma de los pesos no puede exceder a 100.</p> <p>Modificar Indicador Clave:</p>	Jefe	20

	20	(Ver Pantalla 15).En el caso de querer modificar uno de los indicadores existentes se debe presionar el botón Seleccionar del indicador que se desea modificar, luego de lo cual se desplegará en la pantalla los datos del indicador clave, y se puede modificar el peso y el óptimo y si está de acuerdo con los cambios presionar el botón Actualizar para grabar las modificaciones hechas, caso contrario presionar el botón Cancelar , con lo que no se guardarán los cambios realizados.	Jefe	30
	30	Eliminar Indicador Clave: (Ver Pantalla 15).En el caso de querer eliminar uno de los indicadores existentes se debe presionar el botón Seleccionar , del indicador que se desea eliminar, luego de lo cual se desplegará en la pantalla los datos del indicador clave y si lo desea eliminar el indicador clave presione el botón Eliminar	Jefe	FIN
	FIN			
2.2. EWM		Permite la generación de informes necesarios para el alineamiento de las estrategias del Call Center.		

<p>2.2.3. Mejor día para llamar</p>	<p>FIN 10</p>	<p>Segmento Gráfico, que puede ser: barras o líneas. Tipo de gráfico depende del gráfico. Gráfico en: 3D o Normal Tamaño del Gráfico.</p> <p>Luego presionamos  y nos despliega un reporte para saber la hora en la que el agente pudo gestionar de mejor forma y el gráfico correspondiente.</p> <p>(Ver Pantalla 21)</p> <p>Para obtener el informe del mejor día para llamar procedemos a llenar los siguientes parámetros:</p> <p>Fecha Inicial Fecha Final Ciudad Producto Segmento Gráfico, que puede ser: barras o líneas. Tipo de gráfico depende del gráfico. Gráfico en: 3D o Normal Tamaño del Gráfico.</p> <p>Luego presionamos  y nos despliega un reporte para saber el día en el que el agente pudo</p>	<p>Supervisor , Jefe</p>	<p>FIN</p>
---	------------------------	--	------------------------------	------------

	FIN	gestionar de mejor forma, y el gráfico correspondiente.		
3. Informes 3.1. Informe de Comportamiento de Calificación	10	<p>(Ver Pantalla 22)</p> <p>Para emitir el informe Comportamiento de Calificaciones se debe ingresar los siguientes parámetros:</p> <p>Rango de fechas para realizar la consulta:</p> <p>Fecha Inicio</p> <p>Fecha Final</p> <p>Ciudad</p> <p>Producto</p> <p>Segmento</p> <p>Comportamiento: Se selecciona la calificación de la cual se quiere observar el comportamiento en un rango de fechas.</p> <p>Gráfico, puede ser Barras o líneas</p> <p>Gráfico en: Puede ser 3D o normal</p> <p>Tipo de Gráfico: Depende del Gráfico escogido, y</p> <p>Tamaño del Gráfico.</p> <p>Luego presionamos  y nos despliega el gráfico del comportamiento de la calificación seleccionada, en un rango de fechas.</p> <p>El gráfico puede ser dirigido a</p>	Supervisor ; Jefe	FIN

<p>3.2. Informe de Seguimiento de Calificación</p>	<p>10</p> <p>FIN</p>	<p>impresora presionando el botón</p> <p></p> <p>(Ver Pantalla 23)</p> <p>Para emitir el informe Seguimiento de Calificaciones se debe ingresar los siguientes parámetros:</p> <p>Rango de fechas para realizar la consulta:</p> <p> Fecha Inicio</p> <p> Fecha Final</p> <p>Ciudad</p> <p>Producto</p> <p>Segmento</p> <p>Presione  para generar el informe.</p> <p>El informe de Seguimiento de Calificaciones puede ser dirigido a impresora, presionando el botón</p> <p></p>	<p>Supervisor ; Jefe</p>	<p>FIN</p>
--	----------------------	--	--------------------------	------------

Pantalla 0. Ingreso al Prototipo Efectiva



EFECTIVA
Administrador de Call Center para Cobranza

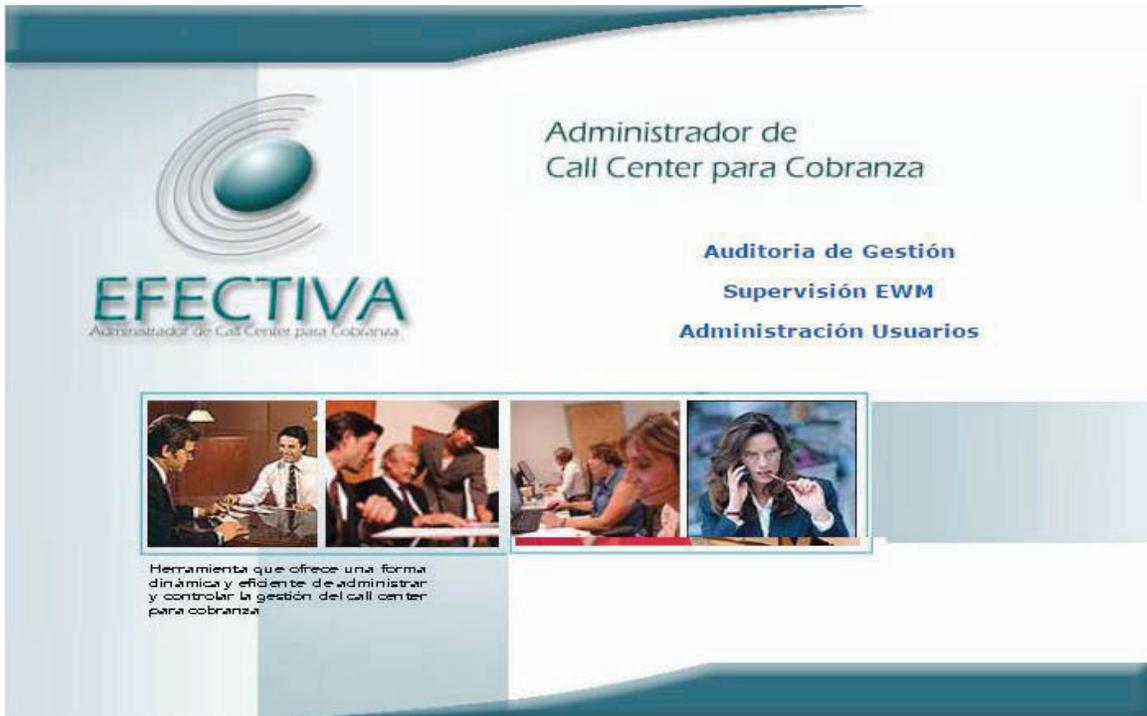
Ingrese usuario y contraseña para acceder al Prototipo

USUARIO

CONTRASEÑA

Aceptar

Pantalla 1. Menú Principal



EFECTIVA
Administrador de Call Center para Cobranza

Administrador de Call Center para Cobranza

Auditoria de Gestión

Supervisión EWM

Administración Usuarios

Herramienta que ofrece una forma dinámica y eficiente de administrar y controlar la gestión del call center para cobranza.

Pantalla 2. Menú Auditoría de Gestión



Pantalla 3. Opción Definición de Plantillas (Crear)

DEFINICIÓN DE PLANTILLAS

DEFINICIÓN DE PREGUNTAS

DEFINICIÓN DE PLANTILLAS				
CIUDAD(s)	PRODUCTO(s)	SECTOR(es)	SECTORES	PUNTOS
QUI	IA_M	AGR		1000
MAN	IA_R	AGR-PET		500
AVE-CUE-GUA-LAG-MAN-POR-QUI	IA_M	AGR		1000
QUI-QUI	IA_M	AGR		1000
AVE-CUE	IA_M	AGR		900
MAN	IA_R	PET		500
AVE-MAN	IA_R MAC_C	AGR CON		700
MAN-QUI	IA_M-TM_S	PET-FUB		800
GUA	IA_R	AGR-CON-PET-PUB		500
GUA	IA_M	AGR		500
QUI	IA_M-IA-R-IA-L-C-IPM_S	CON		100

CIUDAD	EMPRESA_PRODUCTO	SECTOR
<input type="checkbox"/> MARCAR TODOS	<input type="checkbox"/> MARCAR TODOS	<input type="checkbox"/> MARCAR TODOS
<input type="checkbox"/> AMBIENTE <input type="checkbox"/> CUENCA <input checked="" type="checkbox"/> GUAYACIL <input type="checkbox"/> LASCAJON <input type="checkbox"/> MANTA <input type="checkbox"/> PORTOVEJO <input checked="" type="checkbox"/> QUITO	<input type="checkbox"/> IAGA_MAQUNARIA <input type="checkbox"/> IAGA_MPUJOS <input checked="" type="checkbox"/> MACASA_CAMIONES <input type="checkbox"/> TPIASA_SERVICIO	<input type="checkbox"/> AGRICOLA <input checked="" type="checkbox"/> COMERCIO <input type="checkbox"/> PETROLERO <input type="checkbox"/> PUBLICO
<input type="text" value="GUA-QUI"/>	<input type="text" value="MAC_C"/>	<input type="text" value="CON"/>
DE EVALUAR A OBTENER UN PUNTAJE DE:	<input type="text" value="200"/> PUNTOS	
<input type="button" value="Instalar"/> <input type="button" value="Guardar"/> <input type="button" value="Cancelar"/>		

Pantalla 4. Opción Definición de Plantillas (Modificar y/o Eliminar)

CIUDAD(S)	PRODUCTO(S)	SEMENTO(S)	PUNTOS	
QUI	IIA_M	AGR	1000	Restaurar
MAN	IIA_R	AGR-PET	500	Restaurar
AMB-CJE-GUA-LAG-MAN-POR-QUI	IIA_M	AGR	1000	Restaurar
CUE-QUI	IIA_M	AGR	800	Restaurar
AMB-CUE	IIA_M	AGR	900	Restaurar
MAN	IIA_N	PET	500	Restaurar
AMB-MAN	IIA_M-MAC_C	AGR-CON	700	Restaurar
MAN-QUI	IIA_M-TPM_S	PET-SUB	800	Restaurar
GUA	IIA_R	AGR-CON-PET-PUB	200	Restaurar
GUA	IIA_M	AGR	200	Restaurar
QUI	IIA_M-IIA_R-MAC_C-TPM_S	CON	100	Restaurar
GUA-MAN-QUI	NAC_C	CON	250	Restaurar

Pantalla 5. Opción Definición de Preguntas (Seleccionar)

CIUDAD(S)	PRODUCTO(S)	SEMENTO(S)	
QUI	IIA_M	AGR	Seleccionar
MAN	IIA_R	AGR-PET	Seleccionar
AMB-CUE-GUA-LAG-MAN-POR-QUI	IIA_M	AGR	Seleccionar
CUE-QUI	IIA_M	AGR	Seleccionar
AMB-CJE	IIA_M	AGR	Seleccionar
MAN	IIA_N	PET	Seleccionar
AMB-MAN	IIA_M-MAC_C	AGR-CON	Seleccionar
MAN-QUI	IIA_M-TPM_S	PET-SUB	Seleccionar
GUA	IIA_R	AGR-CON-PET-PUB	Seleccionar
GUA	IIA_M	AGR	Seleccionar
QUI	IIA_M-IIA_R-MAC_C-TPM_S	CON	Seleccionar
GUA-MAN-QUI	NAC_C	CON	Seleccionar

Pantalla 6. Opción Definición de Preguntas (Crear)

CIUDAD(S)	PRODUCTO(S)	SEMENTO(S)	Regresar
QUI	IIA_M-IIA_R-MAC_C-TPM_S	CON	

No.	PREGUNTA ASIGNADA PARA ESTA PLANTILLA	Seleccionar
1	REALIZA UN SALUDO INICIAL?	Seleccionar
2	¿CUAL ES EL TIEMPO PROMEDIO PARA CONCEPTAR UNA RESPUESTA?	Seleccionar
3	¿ESTÁ COMPROMETIDO A RESOLVER EL PROBLEMA DEL CLIENTE?	Seleccionar

Tipo de pregunta: EXIGE RESPUESTA (Si o No)

Pregunta: APLICA TÉCNICAS DE ATENCIÓN AL CLIENTE?

SI	NO
PUNTAJE 100	

Pantalla 7. Modificación de Preguntas (Modificar y/o Eliminar)

ASIGNACION DE PREGUNTAS

Ciudad(s)	PRODUCTO(s)	SEGMENTO(s)
AMB-CUE	IIA_M	AGR

REGISTRAR PREGUNTA

RESUESTA	PUNTAJE
ALTO	100
BAJO	25
MEDIO	50

Pantalla 8. Opción Evaluación de Agente (Buscar)

SELECCIONE AGENTE PARA EVALUAR

CEDULA APELLIDO 17

CEDULA	APELLIDOS	NOMBRES	CARGO	CIUDAD	EXTENSION
1710832932	Barba	Carlos	Agente Telefonico	Quito	138
1710832936	Eraso	Mauricio	Agente Telefonico	Quito	140
1710832935	Espin	Juan	Agente Telefonico	Ambato	130
1710832937	Miranda	Angel	Agente Telefonico	Guayaquil	210
1710832930	Muñoz	Giovanna	Agente Telefonico	Quito	120
1710832934	Paez	Carolina	Agente Telefonico	Quito	122
1754181150	Rodriguez	Daniel	Agente Telefonico	Quito	101
1710832941	Venegas	Paulina	Agente telefonico	Luzern	121

Pantalla 9. Opción Evaluación de Agente (Seleccionar Plantilla)

SELECCIONE AGENTE PARA EVALUAR

DATOS PERSONALES

Cédula: 1710832932
 Nombre: Barba Carlos | Tel. Casa: 2888555
 Dirección: Pusuqui | Celular: 099967564
 Ciudad: Quito | Extension: 138

LISTA DE PLANTILLAS

No.	Ciudad(s)	PRODUCTO(s)	SEGMENTO(s)
1	QUI	IIA_M	AGR
2	AMB-CUE-SUA-LAG-MAN-POR-QUI	IIA_M	AGR
3	CUE-QUI	IIA_M	AGR
4	MAN-QUI	IIA_M-TM_S	PET-RUB
5	QUI	IIA_M-IIA_B-MAC_C-TPM_S	CON

Pantalla 10. Opción Evaluación de Agente (Realizar Evaluación)

Auditoria de Gestión

Plantilla Evaluación

AUDITANDO A << Espin Ivan >>

PLANTILLA	REGRESAR		
Ciudades	Productos	Segmentos	
AMB-MAN	TIA_R-MAC_C	AGR-CON	
Ingrese nombre del Cliente:	Constructora Odebrecht	Fecha:	26/04/2015 / 16:18:22

PREGUNTA: 2 De 2

Pregunta: **CUAL ES EL TIEMPO PROMEDIO PARA CONCRETAR UNA RESPUESTA?**

Respuesta:

Puntaje: Ingresar Finalizar Auditoria

PREGUNTAS YA CONTESTADAS	
PUNTAJE ACUMULADO:	80 SOBRE 700
No. RESPUESTA	PUNTAJE
1 CONOCE BIEN LA EMPRESA Y PRODUCTO OFRECIDO	80 Seleccionar

Pantalla 11. Consulta de Evaluación (Consulta General y Reporte)

Auditoria de Gestión

Plantilla Evaluación

CONSULTAS Y REPORTES

QUE ESTEN ENTRE UN RANGO DE FECHAS

Fecha Inicial: 01/01/2015 Fecha Final: 26/04/2015

QUE EL PUNTAJE TOTAL ESTE ENTRE

Desde: 0 Hasta: 1000

AUDITORIAS POR QUE TIPO DE RECURSO

ELIJA TODOS LOS AUDITORES

Recurso: Miranda Angel

[Aceptar](#) [Imprimir](#)

FECHA	HORA	AUDITOR	RECURSO	PUNTAJE TOTAL	Seleccionar
07/04/2015	22:32:31	Daniel	Miranda Angel	120	Seleccionar
07/04/2015	22:33:19	Daniel	Miranda Angel	370	Seleccionar
08/04/2015	15:25:41	Daniel	Miranda Angel	250	Seleccionar
11/04/2015	21:31:06	Daniel	Miranda Angel	370	Seleccionar
12/04/2015	21:58:53	Daniel	Miranda Angel	320	Seleccionar
25/04/2015	21:47:52	Daniel	Miranda Angel	320	Seleccionar
25/04/2015	21:48:42	Daniel	Miranda Angel	250	Seleccionar

Pantalla 12. Consulta de Evaluación (Consulta Detallada y Reporte)

Auditoria de Gestión

Plantilla Evaluación

CONSULTAS Y REPORTES

QUE ESTEN ENTRE UN RANGO DE FECHAS

Fecha Inicial: 01/01/2015 Fecha Final: 26/04/2015

QUE EL PUNTAJE TOTAL ESTE ENTRE

Desde: 0 Hasta: 1000

AUDITORIAS POR QUE TIPO DE RECURSO

ELIJA TODOS LOS AUDITORES

Recurso: Miranda Angel

[Aceptar](#) [Imprimir](#)

Auditor: Daniel fecha: 07/04/2015

Recurso: Miranda Angel Hora: 22:32:31

Puntaje: 130

RESPUESTA	RESPUESTA	PUNTAJE
NIVEL DE CAPACIDAD DE MANEJAR CLIENTES ENVIADOS	MUY BUENA	100
REALIZA UN SALUDO INICIAL?	SI	30

[Cancelar](#)

Pantalla 13. Opción Estadísticas de Evaluación (Consultar)

- EVALUACION AGENTE
- CONSULTA EVALUACION
- ESTADISTICAS EVALUACION

ESTADISTICAS DE EVALUACIONES			
Definición:	Por Puntaje	Tipo Recurso:	ELIJA TODOS
Tamaño Graf.:	50 %	Auditor:	ELIJA TODOS LCS AUDITORES
Elija Grafico:	BARRAS	Tipo Grafico:	Column
3D:	3D		
Fecha Inicial:	01/03/2015	Fecha Final:	31/03/2015
Visualizar			

[Imprimir](#)

ESTADÍSTICA DE EVALUACIÓN		
Ref	DETALLE	VALOR
P1	De 0 a 20 Puntos	2
P3	De 101 a 150 Puntos	3
P4	De 151 a 200 Puntos	1
P5	De 201 a 250 Puntos	3
P6	De 251 a 300 Puntos	1

GRAFICO RESULTANTE

Pantalla 14. Crear Indicadores Clave

MÓDULOS ATRÁS

Supervisión EWM

(Early Warning Monitor)

Indicadores
Mantenimientos
EWM

Governia

- MANTENIMIENTO DE INDICADORES CLAVES

PARAMETROS PARA GENERACION DE INFORME DE INDICADORES DE GESTION			
CUIDAD	PRODUCTO	SECTOR	
Ambato	MAQUINARIA	AGRICOLA	Seleccionar
Ambato	REPUESTOS	PUBLICO	Seleccionar
Cuenca	MAQUINARIA	AGRICOLA	Seleccionar
Cuenca	REPUESTOS	CONSTRUCCION	Seleccionar
Cuenca	SERVICIO	CONSTRUCCION	Seleccionar
Quínez	CAMIONES	CONSTRUCCION	Seleccionar
Quínez	MAQUINARIA	AGRICOLA	Seleccionar
Quínez	REPUESTOS	PETROLERO	Seleccionar
Manabí	SERVICIO	CONSTRUCCION	Seleccionar
Quito	MAQUINARIA	AGRICOLA	Seleccionar
Quito	REPUESTOS	AGRICOLA	Seleccionar
Quito	REPUESTOS	CONSTRUCCION	Seleccionar

CUIDAD	PRODUCTO	SECTOR
AMBATO	REPUESTOS	PUBLICO

[Aceptar e Ingresar Key Indicators](#)

Pantalla 15. Modificar y Eliminar Indicadores Clave

MANTENIMIENTO DE INDICADORES CLAVES

PARAMETROS PARA GENERACION DE INFORME DE INDICADORES DE GESTION				
CIUDAD	PRODUCTO	SECTOR		
Ambato	MAQUINARIA	AGRICOLA		Seleccionar
Ambato	REPUESTOS	PUBLICO		Seleccionar
Cuenca	MAQUINARIA	AGRICOLA		Seleccionar
Cuenca	REPUESTOS	CONSTRUCCION		Seleccionar
Cuenca	SERVICIO	CONSTRUCCION		Seleccionar
Guanoqui	CAMIONES	CONSTRUCCION		Seleccionar
Guanoqui	MAQUINARIA	AGRICOLA		Seleccionar
Guanoqui	REPUESTOS	PETROLIO		Seleccionar
Maipa	SERVICIO	CONSTRUCCION		Seleccionar
Quito	MAQUINARIA	AGRICOLA		Seleccionar
Quito	REPUESTOS	AGRICOLA		Seleccionar
Quito	REPUESTOS	CONSTRUCCION		Seleccionar
Quito	SERVICIO	PUBLICO		Seleccionar

CIUDAD	PRODUCTO	SECTOR
AMBATO	REPUESTOS	PUBLICO

Ingresar Eliminar

MANTENIMIENTO DE KEY INDICATORS

INDICACION	UNDA (%)	OPTIMO (%)	
HIT RATE	15 %	30 %	Seleccionar
CONTACT RATE	20 %	70 %	Seleccionar
PRIME RATE	25 %	55 %	Seleccionar
EFICIENCIA	0 %	0 %	Seleccionar

EFICIENCIA Peso: 40 % Optimo: 00 % Actualizar Cancelar

Pantalla 16. Crear Calificación

MANTENIMIENTO DE CALIFICACIONES

INFORME DE COMPORTAMIENTO DE CALIFICACION

INFORME DE SEGUIMIENTO DE CALIFICACION

PARAMETROS PARA GENERACION DE INFORME DE INDICADORES DE GESTION

CIUDAD	PRODUCTO	SECTOR		
Ambato	MAQUINARIA	AGRICOLA		Seleccionar
Cuenca	MAQUINARIA	AGRICOLA		Seleccionar
Cuenca	REPUESTOS	CONSTRUCCION		Seleccionar
Cuenca	SERVICIO	CONSTRUCCION		Seleccionar
Guanoqui	CAMIONES	CONSTRUCCION		Seleccionar
LagoAgro	REPUESTOS	CONSTRUCCION		Seleccionar
Quito	MAQUINARIA	AGRICOLA		Seleccionar
Quito	REPUESTOS	CONSTRUCCION		Seleccionar
Quito	SERVICIO	PUBLICO		Seleccionar

CIUDAD	PRODUCTO	SECTOR
QUITO	SERVICIO	PUBLICO

Ingresar

INDICADOR DE CALIFICACION

De Que	Para Que	CALIFICACION	DESCRIPCION	
0 %	0 %	A	EXCELENTE	Seleccionar
0 %	0 %	B	MUY BUENO	Seleccionar
0 %	0 %	C	BUENO	Seleccionar
0 %	0 %	D	REGULAR	Seleccionar
0 %	0 %	E	MALO	Seleccionar

De Que: % Para Que: % Actualizar

Calificación: Descripción: Cancelar

Pantalla 17. Modificar y Eliminar Calificación

MANTENIMIENTO DE CALIFICACIONES

INFORME DE COMPORTAMIENTO DE CALIFICACIONES

INFORME DE SEGUIMIENTO DE CALIFICACION

PARAMETROS PARA GENERACION DE INFORME DE INDICADORES DE GESTION			
CUIDAD	PRODUCTO	SEGMENTO	
Ambato	NAQUINARIA	AGRICOLA	Seleccionar
Cuenca	NAQUINARIA	AGRICOLA	Seleccionar
Cuenca	REPUESTOS	CONSTRUCCION	Seleccionar
Cuenca	SERVICIO	CONSTRUCCION	Seleccionar
Guayaquil	CAMIONES	CONSTRUCCION	Seleccionar
LagoAgrio	REPUESTOS	CONSTRUCCION	Seleccionar
Quito	NAQUINARIA	AGRICOLA	Seleccionar
Quito	REPUESTOS	CONSTRUCCION	Seleccionar
Quito	SERVICIO	PUBLICO	Seleccionar

QUITO	REPUESTOS	CONSTRUCCION
Ingresar	Eliminar	

INDICADOR DE CALIFICACION				
>= Que	<= Que	CALIFICACION	DESCRIPCION	
98 %	100 %	A	EXCELENTE	Seleccionar
80 %	97 %	B	MUY BUENO	Seleccionar
70 %	79 %	C	BUENO	Seleccionar
50 %	69 %	D	REGULAR	Seleccionar
0 %	49 %	E	MALO	Seleccionar

>= Que: 98 %	<= Que: 100 %	Actualizar
Calificación: A	Descripción: EXCELENTE	Cancelar

Pantalla 18. Elegir Parámetros para Generar EWM

GENERAR EWM

MEJOR HORA PARA LLAMAR

MEJOR DIA PARA LLAMAR

Supervisión EWM (Early Warning Monitor)

Indicadores | Mantenimientos | EWM

<< INFORME DE INDICADORES CLAVE (E.W.M) >>

Fecha Inicio: 2015/03/02 | Fecha fin: 2015/05/02

Tipo Reporte: DETALLADO RESUMIDO

CUIDAD	PRODUCTO	SEGMENTO
GUAYAQUIL	CAMIONES	CONSTRUCCION

[Visualizar](#)

Pantalla 19. Generar EWM

GENERAR EWM

MEJOR HORA PARA LLAMAR

MEJOR DIA PARA LLAMAR

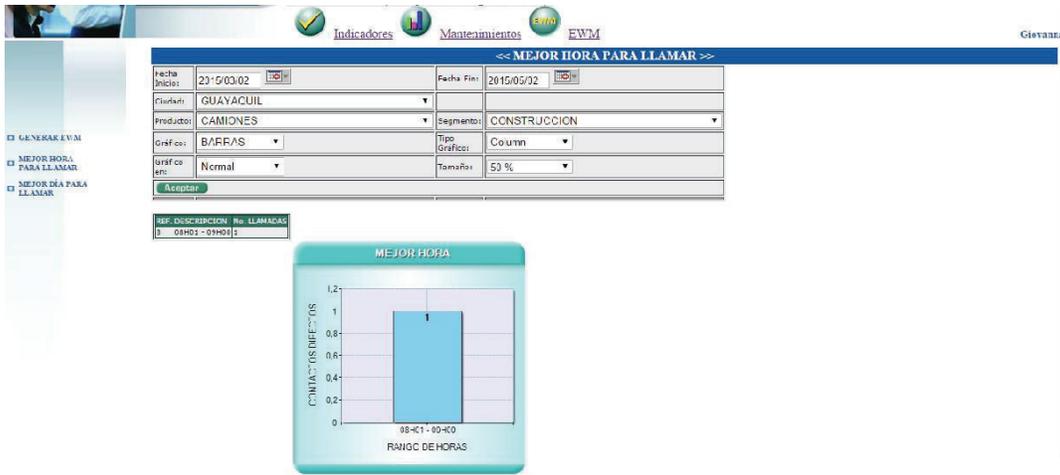
Supervisión EWM (Early Warning Monitor)

Indicadores | Mantenimientos | EWM

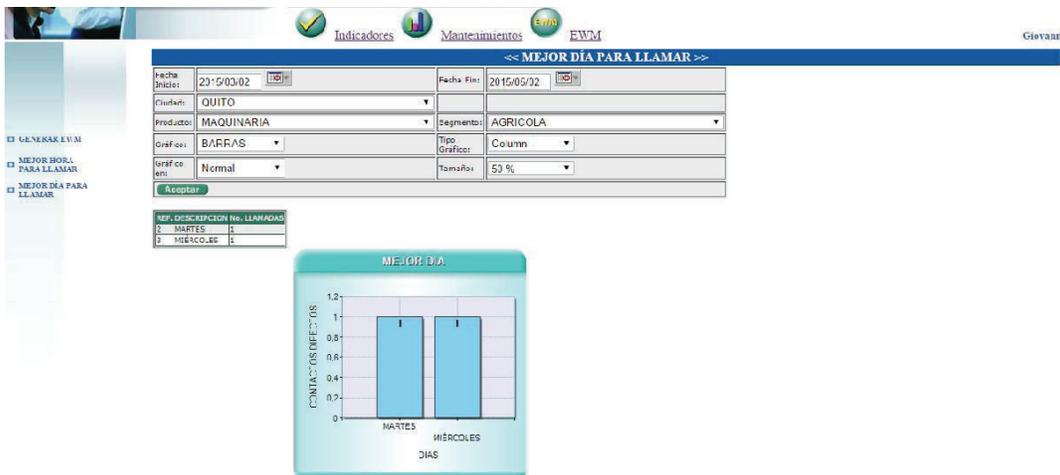
CIUDAD	PRODUCTO	AGENTE TELEFONICO	TASA DE CONEXIONES (HIT RATE)			TASA DE CONTACTOS (CONTACT RATE)			TASA DE COMPROMISOS (PROMISE RATE)			EFICIENCIA		CALIFICACION PORCENTAJE			
			No. LLAMADAS PROM. CONECTADAS HORA	LLAMADAS PROM. CONECTADAS HORA	No. CONTACTOS	No. CONTACTOS DIRECTOS	No. COMPROMISOS PAGOS CUMPLIDOS	OBTENIDO	CALIFICACION	OBTENIDO	CALIFICACION						
GUAYAQUIL	CAMIONES	Angel Miranda	6	0.71	4	3	1	0	80 %	20 %	75 %	25 %	33.33 %	5.55 %	0 %	0 %	50.55 %



Pantalla 20. Consulta Mejor Hora Para Llamar

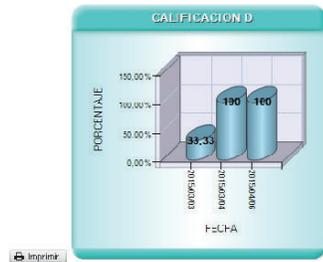


Pantalla 21. Consulta Mejor Día Para Llamar



Pantalla 22. Opción Informe de Comportamiento de Calificaciones

INFORME DE COMPORTAMIENTO DE LA CALIFICACION			
Fecha Inicio:	2015/03/02	Fecha Fin:	2015/05/02
Ciudad:	ELIJA TODAS	Segmento:	ELIJA TODOS
Producto:	ELIJA TODAS	Gráfico:	BARRAS
Comportamiento de:	Calificación D	Tipo Gráfico:	Column
Gráfico en:	3D	<input type="button" value="Aceptar"/>	
Tamaño:	50 %		



Pantalla 23. Opción Informe de Seguimiento Calificaciones

MODULOS: ATRAS

Supervision EWM
 (Early Warning Monitor)

Giovanna

Indicadores
Mantenimientos
EWM

<< INFORME DE SEGUIMIENTO POR CALIFICACION >>

Fecha Inicio:	2015/03/02	Fecha Fin:	2015/05/02
Ciudad:	QUITO	Segmento:	AGRICOLA
Producto:	MAQUINARIA	<input type="button" value="Aceptar"/>	

FECHA	A	B	C	D	E
2015/03/03	50 %		25 %	25 %	
2015/03/04			100 %		