



ANÁLISIS Y PLANTEAMIENTO DEL PROBLEMA

3.1. Análisis del proceso actual de transferencia de CDRs.

Actualmente el Sistema de Mediación de la compañía, se encuentra a su máxima capacidad de procesamiento, darle mantenimiento resulta muy costoso, presenta diversos problemas operativos y de funcionamiento, es necesario invertir una gran cantidad de horas hombre en su operación.

Los tiempos de entrega de la información hacia sus sistemas cliente no son los ideales, lo que en casi la totalidad de casos atenta directamente con la captación de ingresos de forma oportuna a la compañía.



Figura 3.1.1 Proceso actual de transferencia de CDRs



En la figura 3.1.1 se muestra en un cronograma el procesamiento actual de CDRs. El Sistema de Mediación actualmente en operación no se ajusta a las políticas tecnológicas definidas por la compañía además de que se incrementará considerable la base de datos de clientes activos provocando una mayor demanda de procesamiento de CDRs y que el sistema de mediación quede obsoleto, puesto que de no tener esta capacidad de procesamiento se estarían limitando las posibilidades de crecimiento de la compañía.

Las implicaciones que tiene el proceso de transferencia así como el sistema de mediación actual se describen brevemente a continuación:

- Evitar la manipulación y posible alteración de los datos.

Debido a la alta intervención humana en el proceso, este se presta a que los datos puedan sufrir alteraciones sea por error o intencionalmente, lo que nos deja con un sistema vulnerable desde el punto de vista operativo.

- El tiempo de recolección de la información es muy alto.

En razón de que los protocolos de transporte no están unificados y de que el grado de automatización es pobre, el tiempo que se lleva la recolección de los datos es mucho más alto del deseable

- El período medio de pago es muy largo.

Al no contar con una plataforma que nos permita contar con la información de manera oportuna, el periodo medio de pago se alarga porque en lugar de contabilizar las llamadas el mismo día que se realizan, a veces pasan semanas y para cuando el cliente recibe su facturación, tal vez ya no se le cobran todas las llamadas del periodo y si las que quedan pendientes se le cobran en el periodo siguiente, es mucho más probable que no las reconozca en su totalidad.

- No se obtienen los datos para tarifar NRT (Near Real Time). En particular, para la facturación a otros operadores.



Abundando el punto anterior, se ha hecho más que evidente con el sistema actual que la disponibilidad de los datos de facturación de manera casi inmediata es vital para la competitividad de la operadora, pues, otras operadoras competidoras ya lo hacen o están por hacerlo y esto les reporta beneficios visibles para el cliente y para la liquidez de la operadora, además de que al contar con la información al momento, es posible supervisar los procesos de manera oportuna.

- No se tiene un buen control de fraude de celulares.

Como actualmente la supervisión no puede hacerse día a día por la lentitud en el proceso y la vulnerabilidad de los datos es muy difícil detectar patrones de fraude en un corto lapso de tiempo para tomar acciones correctivas.

- No se consolidan en un formato común de CDR la información de las diversas plataformas que componen la red celular.

Debido a que la operadora cuenta con equipos de diferentes marcas y especificaciones en los diversos puntos de servicio final y a que estos equipos cuentan con diferentes formatos de CDR, no se cuenta con un formato único de CDR, por lo que al momento es necesario realizar laboriosas tareas semiautomáticas para dar proceso a los CDRs con un consumo excesivo de horas hombre.

- No se garantiza el correcto transporte y almacenamiento de los CDR's generados en las plataformas de conmutación.

Dado que el sistema es semiautomático en algunos de sus procesos, el transporte y almacenamiento de los datos se ve comprometido por la posibilidad de que existan errores u omisiones de parte del personal operativo; además de que las tareas de transporte y almacenamiento no son fácilmente rastreables. Lo anterior, se agrava si tomamos en consideración que contamos con al menos un punto de llegada por cada una de las centrales nacionales y que el personal está de alguna manera rebasado



- No se provee a los clientes del sistema de mediación con una mayor disponibilidad de información.

Como el sistema en su forma actual no genera registros adecuados de seguimiento de las tareas, no existen las interfaces de servicio adecuadas que provean a los clientes de mayor información respecto a los procesos y a sus resultados.

- No se facilita el acceso y procesamiento de la información colocándola en un formato estándar disponible en una base de datos.

Como no se cuenta con un formato estándar de CDRs, los registros de la base de datos tampoco son estándares del todo y deben contar con un dinamismo excesivo que entorpece el proceso y complica las tareas de seguimiento y documentación de la operación.

- El sistema actual se encuentra a su máxima capacidad de procesamiento.

La operadora está expandiendo su oferta de servicios a nivel nacional y está creciendo de manera significativa en su número de clientes y solicitudes de servicios por lo que el sistema en su forma actual se está quedando muy corto para atender a la creciente demanda.

3.2. Recolección y análisis de la información.

El Sistema de Mediación de la compañía actualmente obtiene los archivos AMA que contienen los CDRs generados en los MSC por medio de una conexión vía FTP replicándolos dentro de los servidores de mediación. Estos archivos cumplen con la nomenclatura establecida, lo que permite mantener un control de estos archivos.

Estos archivos pueden rotar ya sea cumpliendo con el tamaño permitido o con un tiempo de duración configurado.



En la figura 3.2.1 se muestra cómo está organizado el sistema de mediación actual.

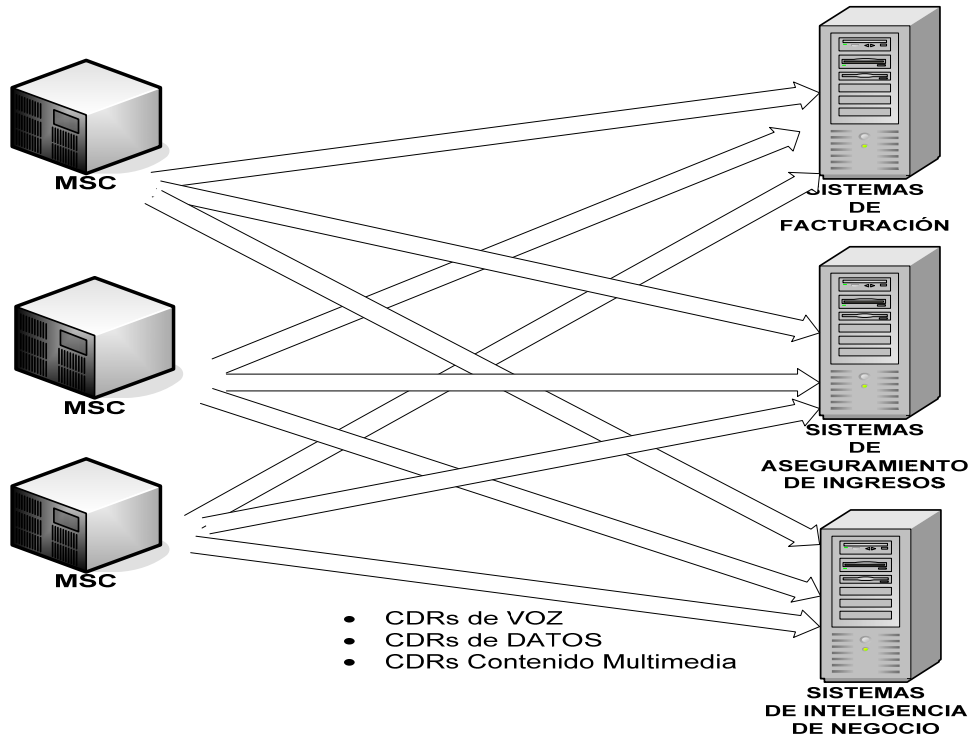


Figura 3.2.1 Sistema de Mediación actual

Este sistema realiza solamente la recolección, validación y distribución de CDR's, maneja solamente archivos y los entrega en su formato original (BCD) a los diferentes sistemas cliente.

Uno de los sistemas cliente de mayor importancia para la compañía es el sistema de facturación de interconexión, a continuación se describe como se realiza el análisis de la información de CDRs para esta facturación.

Principios y tipos de interconexión.

El campo de la interconexión de redes de telecomunicación es vasto e interdisciplinar, pues en él confluyen elementos económicos y regulatorios que determinan la viabilidad del servicio, además de los elementos propiamente



técnicos (topología, calidad, seguridad,..) dichos elementos han de configurarse en base a una serie de principios, que conforman la interconexión de redes de telecomunicación.

Sin duda, el primero de tales principios es el de la universalidad, es decir, que cualquier usuario pueda comunicarse con cualquier otro usuario, con independencia del operador de telecomunicaciones contratado por cada uno de ellos y con independencia, asimismo, de la ubicación geográfica de ambos. Otros principios, básicos, que conforman la interconexión de redes de telecomunicación son la privacidad que garantiza el secreto de las comunicaciones y la neutralidad que avala un tratamiento no-discriminatorio y transparente a los distintos operadores.

Se distinguen dos tipos de tarifas involucradas en la interconexión de redes: las tarifas de interconexión propiamente dichas, que amparan el coste (equipos de conmutación, medios de transmisión,..) de la interconexión de las redes, y las tarifas de interconexión por tráfico cursado (por llamadas telefónicas completadas) entre dichas redes. Estas últimas podrían definirse como el precio que un operador A debe abonar a un operador B por la utilización de la red de este último.

Interconexión de redes fijas.

Primeramente, se establece el plano topológico de la interconexión: el operador dominante u operador con poder significativo en el mercado ofrece a los demás operadores la posibilidad de acceder, al menos, a un punto de interconexión en cada provincia, punto que será elegido por el operador que solicita la interconexión.

Por otra parte, se detallan también los servicios a interconectar: el servicio telefónico básico: metropolitano, nacional, e internacional.



Tarifas de interconexión por tráfico cursado

Dichas tarifas se facturaban sobre tráfico efectivo cursado, midiendo la duración de cada llamada en minutos redondeados por exceso.

Bajo las anteriores premisas, se estipularon las tarifas de conexión, expresadas entonces en pesos/minuto, que tenían carácter de máximas (es decir, que podrían ser sustituidas por otras, inferiores, que acordaran los operadores de las redes a interconectar), para cada tipo de conexión: metropolitana, nacional e internacional.

Facturación entre operadores, y a los usuarios

Como premisa de partida, se establece que cada llamada individual deberá ser facturada al usuario por una única entidad habilitada; dicha entidad dependerá del tipo de llamada. Así, por ejemplo, en llamadas metropolitanas y nacionales de telefonía básica, se facturará únicamente al usuario origen de la llamada, lo cual hará el operador al cual tiene contratado el acceso dicho usuario. Y, por otra parte, dicho operador tendrá que abonar la correspondiente tasa de conexión al operador que sustenta el acceso del usuario destino de la llamada. Vea figura 3.2.1

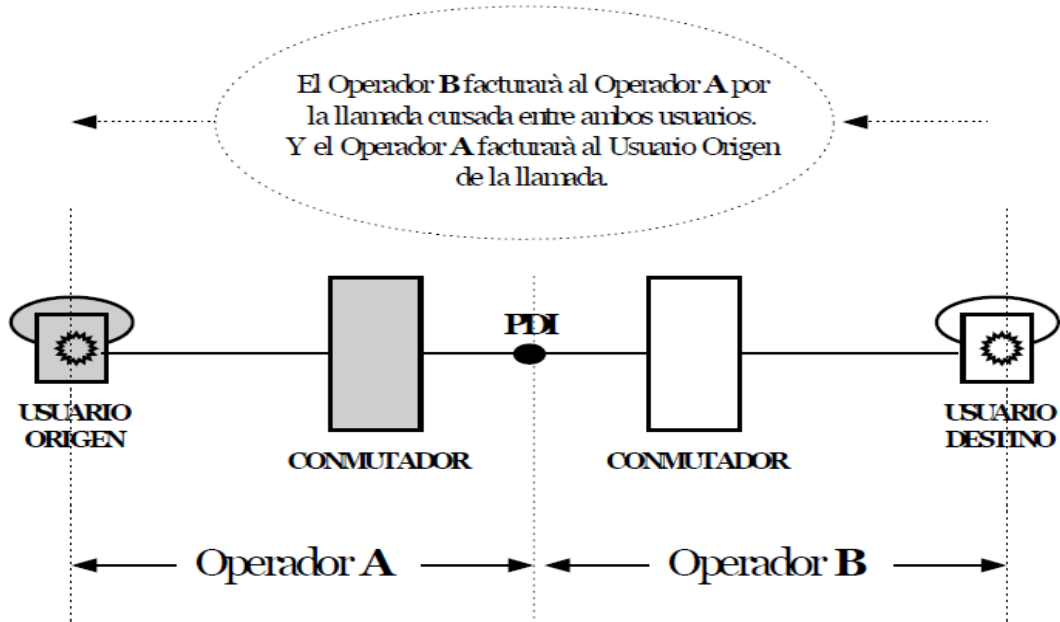


Figura 3.2.1 Facturación de llamadas caso de dos operadores

En caso de llamadas nacionales es aplicable el mismo procedimiento anterior. No obstante, en este caso cabe la posibilidad, además, de que el usuario origen de la llamada decida cursar ésta por la red de un operador distinto de aquel con el que tiene contratado el acceso. Vea figura 3.2.2

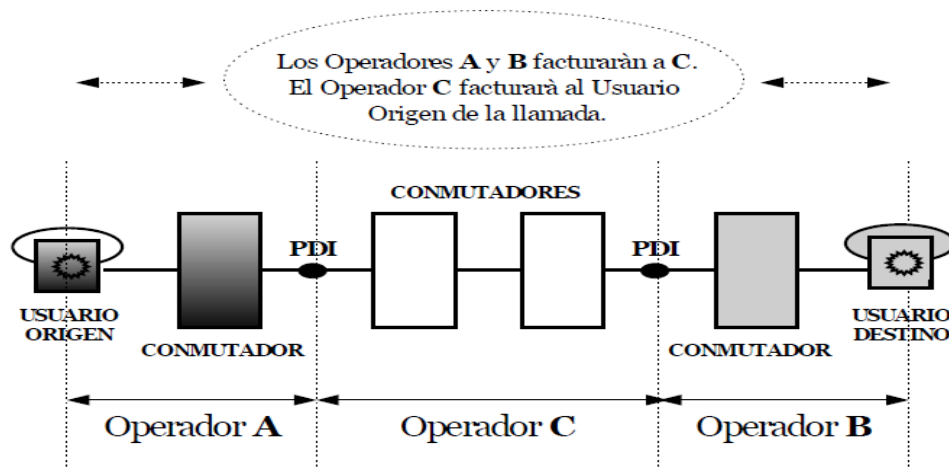


Figura 3.2.2 Facturación de llamadas caso de dos operadores nacional.



En el caso de llamadas internacionales, el operador seleccionado por el usuario origen de la llamada será el responsable de facturar a dicho usuario por la totalidad del servicio, en función de sus precios. Y, por otra parte, dicho operador tendrá que abonar la correspondiente tarifa de interconexión a la entidad con la cual tiene contratado el acceso el usuario en cuestión.

Conexión de redes de móviles.

Los concesionarios nacionales del servicio GSM deberán interconectarse entre sí, interconexión obligatoriamente de tipo directo e igualmente, dichos concesionarios podrán conectarse directamente con otras redes fijas o móviles, públicas o privadas, nacionales o extranjeras. En el primero de los casos se requerirá una propuesta conjunta de tarifas, y en el segundo, obviamente, éstas deberán ser concertadas entre los distintos operadores.

Tarifas de interconexión por tráfico cursado.

A título de referencia, a continuación se reflejan la tarifas, procediendo a señalar que las mismas sólo válidas. Vea figura 3.2.3

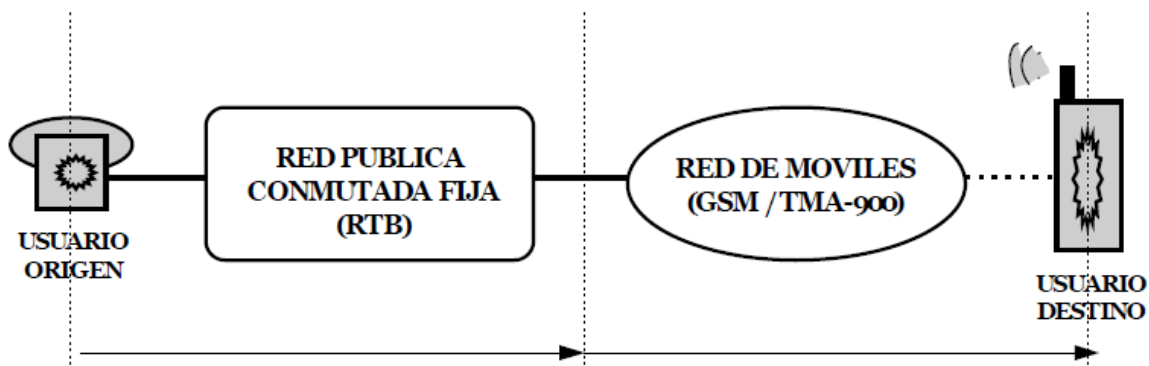


Figura 3.2.3 Llamada nacional, de fijo a móvil

(1) Tarifa (sin IVA) "normal": de lunes a viernes, de 08.00 a 22.00 horas, y sábados, de 08.00 a 14.00 horas.



(2) Tarifa (sin IVA) “reducida”: de lunes a viernes (de 22.00 a 08.00 horas), sábados (a partir de las 14.00 horas), y domingos y festivos de ámbito nacional durante todo el día.

Las tarifas a aplicar a los abonados de la red de móviles se establecerán libremente por los operadores del servicio GSM con la limitación de que, en ningún caso, podrán ser inferiores a las establecidas para los abonados de la red fija. Vea figura 3.2.4

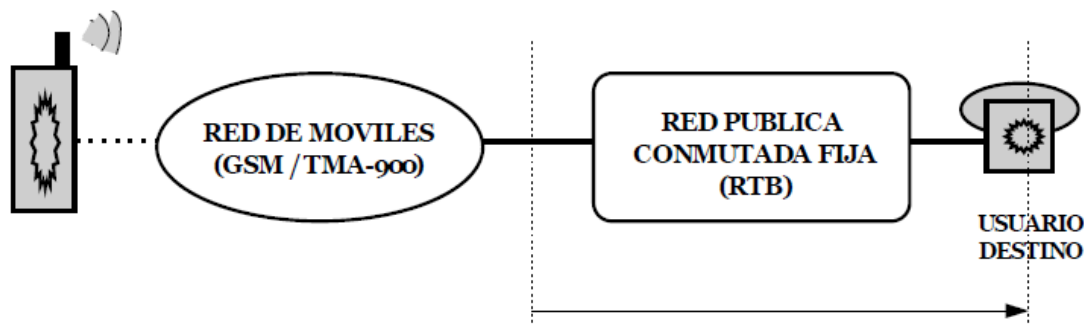


Figura 3.2.4 Llamada Nacional, de Móvil a Fijo.

(1) Tarifa (sin IVA) “normal”: de lunes a viernes, de 08.00 a 22.00 horas, y sábados, de 08.00 a 14.00 horas.

(2) Tarifa (sin IVA) “reducida”: de lunes a viernes (de 22.00 a 08.00 horas), sábados (a partir de las 14.00 horas), y domingos y festivos de ámbito nacional durante todo el día.

La facturación entre ambos operadores se realizará con la periodicidad y la forma en que ambos hayan acordado.

3.3. Requerimientos generales y particulares para el nuevo sistema.

Un requerimiento es un conjunto de propiedades o restricciones definidas con precisión, que un sistema de software debe satisfacer.

Los requerimientos son la pieza fundamental en un proyecto de desarrollo de software, ya que es la base para:



- Planear el proyecto y los recursos que se usarán en él.
- Especificar el tipo de pruebas que se habrán de realizar al sistema.
- Planear la estrategia de prueba a la que habrá de ser sometido el sistema.
- Son el fundamento del ciclo de vida del proyecto.

El Sistema de Mediación recopilará los archivos de voz, datos y de multimedia que sean enviados a través de la red para su tasación, facturación y estadística.

Debemos de puntualizar que el cliente no requiere de un front end porque la información se enviará para su procesamiento a diferentes sistemas de facturación o estadísticas internos, de esta forma sólo se mencionarán los requerimientos generales y particulares acerca de la funcionalidad del mismo y no sobre como el usuario interactuará con él.

Por lo anterior, a continuación se explicarán los requerimientos para el back end.

Requerimientos generales.

Los requerimientos generales para el SIMED son los siguientes:

- El sistema tendrá que unificar la información a partir de distintas fuentes de datos y distintas formas de comunicación con éstas.
- El sistema tendrá que complementar los CDRs, agregando la siguiente información a partir del CDR Original.
 - IRM, agregara el correspondiente IRM (International Roaming MIN) a los respectivos MIN (Mobile Identification Number) que aparezcan en el CDR.
 - Indicará si alguno de los números que intervienen en la llamada corresponden a operadores con convenios especiales con la compañía.
 - Indicará toda la información útil para las troncales y celdas que aparezcan en el CDR, indicará si es una troncal de interconexión, de LD, LDI. Etc.



- Indicará si algunos de los números que intervienen en el CDR corresponden a los clientes corporativos de la compañía.
- Indicará si algunos de los números que intervienen en la llamada es un cliente de pospago.
- Indicará a que operador corresponden todos los números que intervienen en la llamada.
- Indicará si algunos de los números que intervienen en la llamada corresponde a números de carriers internacionales
- Deberá tener la capacidad de procesar aproximadamente 4 millones de CDRs (es decir, la cantidad de CDRs prevista para un día) en menos de medio día.
- Se requiere que tenga la capacidad de procesar un mes de CDRs en una semana o menos.
- En la base de datos se requiere tener almacenados tres meses de información de CDR de voz, aproximadamente 360 millones de CDRs.
- Se solicita una confiabilidad de un 99.9% en el procesamiento de CDRs.
- No debe haber intervención manual para la clasificación y homologación de los CDRs a procesar.
- Debe de implementar los protocolos dispuestos por los proveedores de las MSC para transferir CDRs por bloques y además mantener la transferencia de archivos para conciliación.
- Un CDR escrito por la MSC deberá encontrarse en la base de datos de CDRs, clasificado, homologado, enriquecido y distribuido a los diferentes sistemas cliente.
- Se requiere que el sistema sea de alta disponibilidad de al menos un 99.9% mensual.



Requerimientos particulares.

Los requerimientos particulares para este sistema son:

- Sistema de facturación a otros operadores (Interconexión)
 - Se deberán entregar solamente los CDRs, homologados, clasificados y enriquecidos que sean necesarios para la facturación de interconexión según los criterios que se definan, se estima que se entregará únicamente un 30% de la totalidad de CDRs al sistema de facturación y deberá de realizarse en tiempo real, es decir, prácticamente en el momento que se inserta en la base de datos del SIMED se deberá también de estar entregando a este sistema de facturación.
- Para el sistema de facturación a clientes el sistema deberá disponer de un mecanismo para enviar los CDRs con los criterios necesarios. Este mecanismo deberá ser fácilmente configurable y deberá disponer de mecanismos de reintentos y manejo de errores.
- Para los sistemas de aseguramientos de ingresos y de inteligencia de negocios se enviarán reportes diarios y mensuales, y se depositarán en los servidores especificados por cada aplicación, así como los formatos y contenido solicitado por cada uno.
- Para los sistemas de facturación a clientes de telefonía fija el sistema deberá disponer de un mecanismo para enviar los CDRs con los criterios necesarios hacia los sistemas de facturación de clientes de telefonía fija. Este mecanismo deberá ser fácilmente configurable y deberá disponer de mecanismos de reintentos y manejo de errores.
- Un proceso independiente conciliará los CDRs procesados por el protocolo a bloques y los archivos recolectados, para en su caso realizar los envíos necesarios al nuevo Sistema de Mediación SIMED.



3.4. Posibles módulos del nuevo sistema.

De acuerdo al análisis realizado para el SIMED se construirán los módulos representados en la figura 3.4.2 y que se describen a continuación:

- Módulo de recolección
- Módulo de parseo
- Módulo de catálogos
- Base de datos
- Módulo de explotación de base de datos

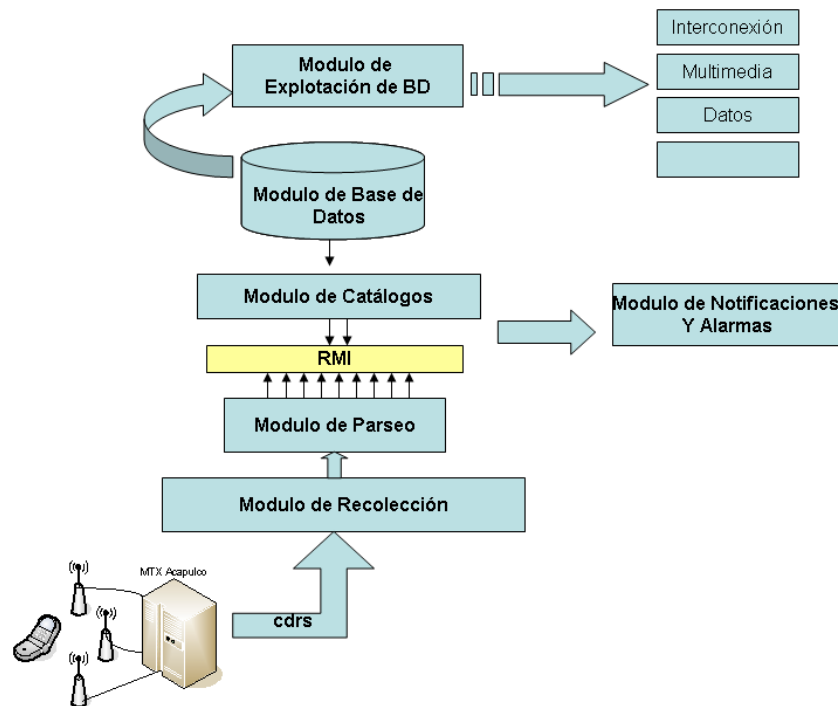


Figura 3.4.1. Módulos de la aplicación.

Módulo de recolección

Es el módulo de la aplicación encargado de establecer conectividad con las plataformas de conmutación celular y de implementar el protocolo de la central para recolección por bloques de CDRs en lugar de archivos.



El proveedor de la central de conmutación celular o MSC, dispuso de librerías de Java para programar su protocolo propietario, dentro de las responsabilidades de este módulo está la de garantizar el correcto transporte de los CDRs validando la consistencia de la información transferida así como su formación correcta.

Este módulo estará programado en Java de acuerdo con las políticas tecnológicas de la compañía.

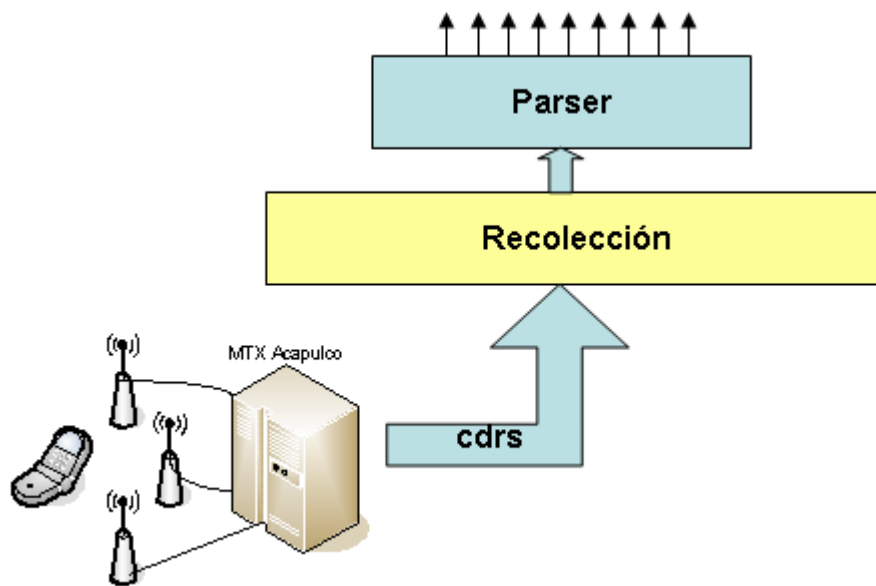


Figura 3.4.2. Módulo de recolección.

En la figura 3.4.2 se resalta el módulo de recolección dentro del Sistema de Mediación por bloques de CDRs y no por archivos. Como se observa el módulo de recolección establecerá una conexión dedicada con la MSC y se comunicará con esta mediante el protocolo propietario del proveedor de la MSC, se establecerá una sesión dedicada y estará enviando bloques de CDRs al módulo de parseo y a su vez los estará almacenando los bloques en archivos, manejando cifras de control.

Posteriormente transferirá por FTP el archivo de CDRs ya cerrado por la MSC y lo comparará con el archivo escrito por el protocolo a bloques, notificando cualquier



anormalidad e incluso seleccionando los bloques faltantes y enviándolos al módulo de parseo para su reproceso.

Módulo de parseo

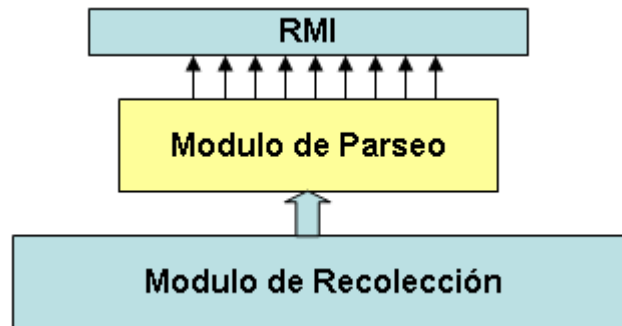


Figura 3.4.3. Módulo de parseo.

Como se observa en la figura 3.4.3 el módulo de Parseo es el que va a recibir los bloques de CDRs desde el módulo de recolección, será el encargado de leer cada CDR, clasificarlo, formatearlo y enriquecerlo con información de negocio, para su posterior inserción en la base de datos para su explotación y entrega manual y/o automática.

Este módulo de parseo necesitará información de catálogos de base de datos para complementar e enriquecer la información de cada CDR para ello implementará un cliente que consulte un servicio publicado por el módulo de catálogos mediante RMI.

En la figura 3.4.4 se muestra a alto nivel los submódulos que van a componer el módulo de parseo:

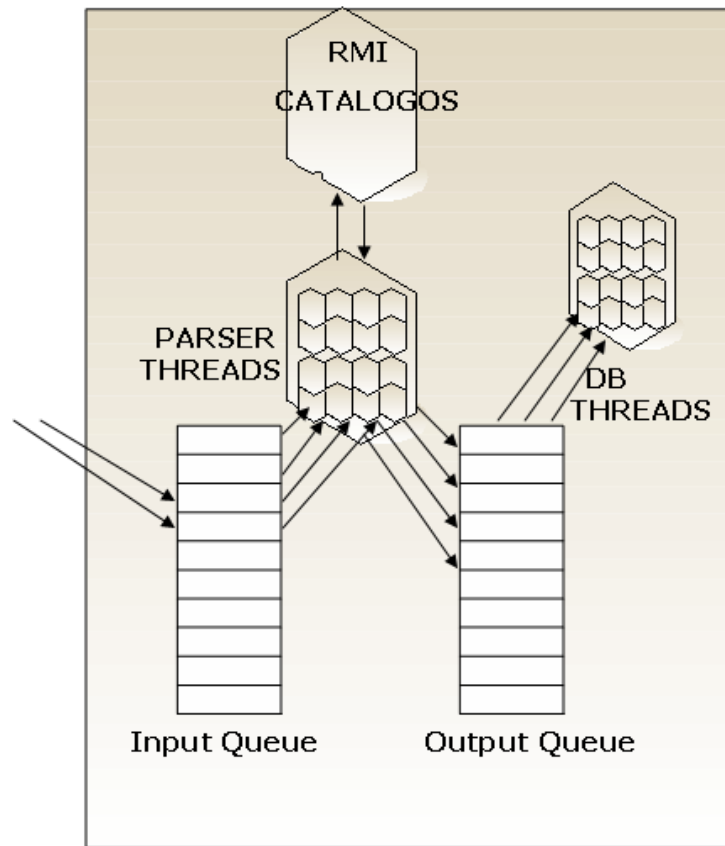


Figura 3.4.4. Submódulos que van a componer el módulo de parseo.

Este módulo también contendrá las reglas de negocio necesarias para enriquecer y complementar la información de cada CDR, utilizará el módulo de catálogos para consultar información respecto a numeración y respecto a troncales.

RMI

RMI (Java Remote Method Invocation) es una funcionalidad ofrecida por Java para compartir procesos o métodos de forma remota, ofreciendo un mecanismo simple para la comunicación de procesos en ambientes distribuidos basados en Java exclusivamente.

A través de RMI un programa Java puede exportar un objeto que estará disponible en la red y el programa permanece a la espera de peticiones en un puerto TCP, pudiéndose conectar un cliente e invocar los métodos publicados.



Se decidió hacerlo con RMI dado que las políticas definidas por la compañía obligaban a compartir funcionalidades a otras aplicaciones que pudieran utilizarlas con el propósito de reutilizar código agilizando los futuros desarrollos en Java para la compañía.

Módulo de catálogos

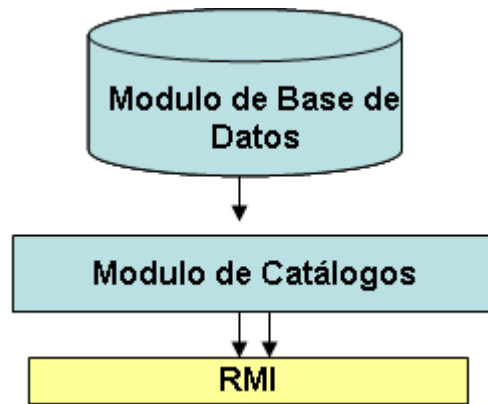


Figura 3.4.5. Módulo de catálogos.

En la figura 3.4.5 se muestra el módulo de catálogos, cuya finalidad será la de proporcionar al módulo de parseo y cualquier aplicación que lo requiera información de la base de datos. Publicará mediante RMI dos métodos, uno asociado a numeración y otro asociado a troncales o celdas, el módulo que consumirá estos métodos, tendrá que invocar estos colocando un número telefónico o bien un número de troncal y su respectiva MSC, con esto el módulo de catálogos entregará toda la información relacionada con ese número o troncal.



Módulo de base de datos y de explotación de BD.

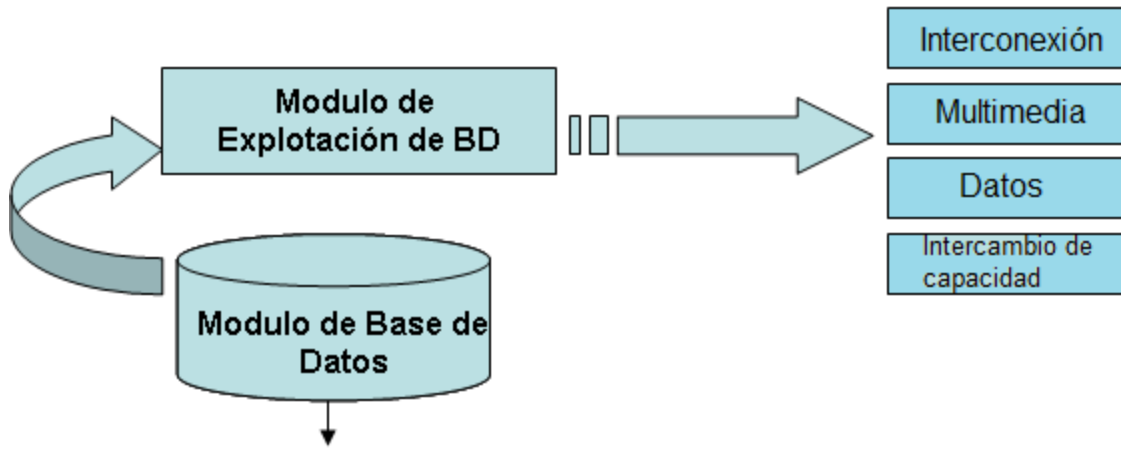


Figura 3.4.6. Módulo de base de datos y de explotación de BD.

La base de datos de la aplicación será donde se almacenen los catálogos necesarios para la clasificación de la información así como el repositorio donde se almacenarán los CDRs clasificados, homologados y enriquecidos, tendrá disparadores y Links hacia las BDs de otra aplicaciones que recibirán estos CDRs o el extracto que de ellos requieran, a su vez filtraran solamente la información necesaria para cada aplicación lo que permitirá entregar la información a las diferentes aplicaciones clientes en tiempo cercano al real.

En la figura 3.4.7 se muestra a un alto nivel como se organizará la Base de datos así como módulo de explotación de esta, uno de los requerimientos para la aplicación es que se tengan 3 meses de CDRs on line, es decir, aproximadamente 360 millones de CDRs almacenados y disponibles en la base de datos. Aprovechando la funcionalidad del manejador se particionarán las tablas de CDRs físicamente, siendo las llaves de partición los campos de mes y de día. Adicionalmente se agregó un campo numérico que almacena la hora lo que permite realizar búsquedas por mes, día y hora.



El diseño de la base incluirá 1440 particiones repartidas en 360 tablespaces, además para la explotación de la base se desarrollaran paquetes (Packages) así como triggers escritos en PL/SQL que estarán entregando, clasificando y transformando la información necesaria de CDRs para cada aplicación cliente que así lo requiera.

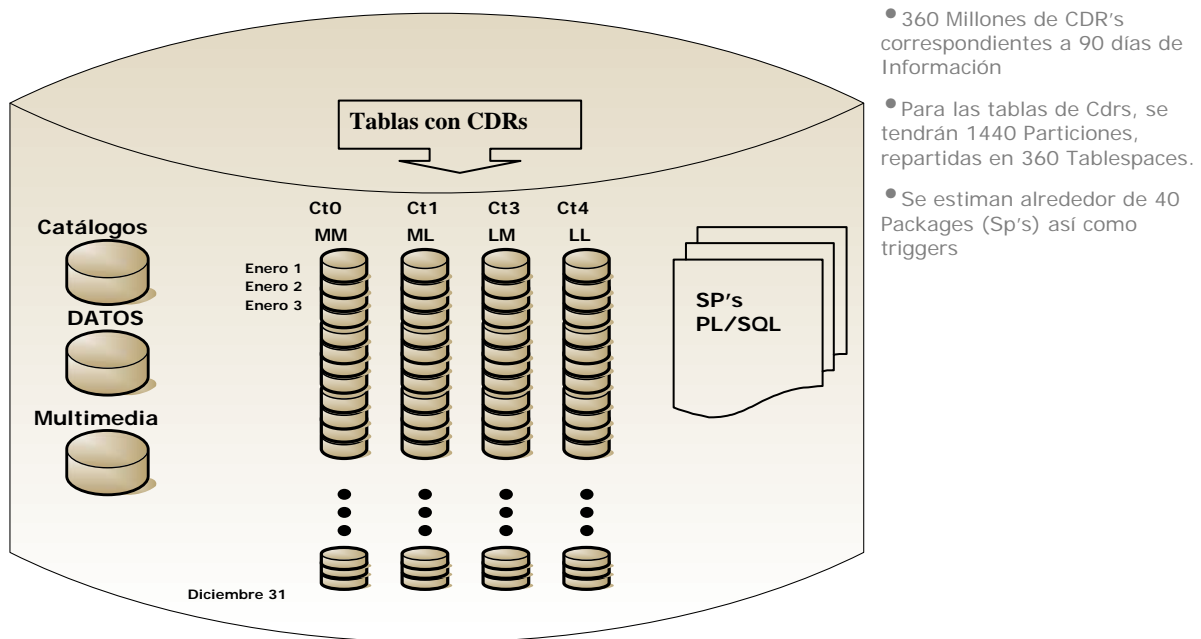


Figura 3.4.7. Organización de la base de datos

3.5. Justificación de la metodología y del software a utilizar.

Para la realización de este proyecto, se nos han especificado la tecnología y la metodología a utilizar para la construcción del SIMED. Estos estándares son determinados por las áreas de tecnología de la compañía cuya función es alinear los desarrollos así como definir la política tecnológica que se seguirá a futuro.

Dada la importancia del desempeño de la nueva aplicación, haremos un breve comparativo entre algunas herramientas existentes en el mercado y que se piensa podrían dar buenos resultados para la aplicación sin que esto signifique que podamos modificar las herramientas a utilizar ya definidas, puesto que uno de los



requerimientos es que el sistema se encuentre alineado a la política tecnológica de la compañía, a continuación se compararan los siguientes RDBMS:

- Informix.
- Progress.
- Oracle 10g.

INFORMIX

Es creado por Informix Software Inc., en el año de 1980, después fue adquirida por IBM en 2001. Informix es un lenguaje de cuarta generación, es considerado como uno de los manejadores de datos más flexibles y potentes. Posee como base para el desarrollo de aplicaciones, el manejo de bases de datos relacionales. Algunas de las características que posee son:

- Utiliza plataforma NT y UNIX.
- Se especializa en aplicaciones tipo GIS (datos geográficos).
- Dispone de herramientas gráficas.
- Gestiona múltiples bases de datos remotas de una consola centralizada.
- Capacidad de relación de datos en múltiples lugares físicos.
- Conecta datos relacionales en páginas web.
- Ocupa menos memoria y recursos que Oracle.
- Se integra con Linux y Oracle.
- Ofrece herramientas para crear menús, formularios de entrada de datos y generadores de listados.
- Soporte para Datawarehouse y Datamining.

PROGRESS

Es un potente sistema de gestión de base de datos relacional, de la compañía de software, llamada "Progress Software Corporation" (o PSC). Progress tiene un



lenguaje de programación 4GL con todas las funciones para trabajar con los datos y la programación lógica. El mismo lenguaje se utiliza para la programación de la interfaz de usuario (GUI y terminal). Se alcanza un mejor rendimiento cuando se trabaja con los datos, en comparación con SQL, ya que la manipulación de datos se hace en un nivel inferior, y con más comandos sin problemas de base de datos de acceso.

Algunas de las características que posee son:

- Solución altamente escalable que proporciona un rendimiento óptimo.
- Hasta 10,000 usuarios concurrentes.
- Soporte para SMP.
- Servidor multienlazado; múltiples servidores por cada base de datos (hasta 256).
- Soporte para múltiples sistemas operativos y plataformas de hardware.
- Conformidad con el ANSI SQL-92.
- Características de seguridad, incluyendo recuperación automática de caídas y transacciones y capacidades de avance "roll-forward".
- Respaldo en línea, soporte a prueba de fallas y reorganización de tablas e índices para alta disponibilidad
- Almacenamiento virtual de datos ilimitado, su única limitante será el hardware.

En la tabla 3.5.1 se muestran las comparaciones de los diferentes RDBMS.



Característica	Informix	Progress	Oracle
Variedad de plataforma en las que opera (Multiplataforma)	Excelente	Excelente	Excelente
Robustez	Excelente	Excelente	Excelente
Facilidad de uso	Bueno	Excelente	Excelente
Compartir los datos con otras bases de datos	Bueno	Bueno	Excelente
Disponibilidad de herramientas	Bueno	Excelente	Excelente
Restauración del sistema	Bueno	Bueno	Excelente
Capacidad de herramientas de información	Bueno	Bueno	Excelente
Cantidad de clientes que pueden atender (conurrencia)	Bueno	Bueno	Excelente
Niveles de seguridad	Bueno	Excelente	Excelente
Costo	Bueno	Bueno	Bueno

Tabla 3.5.1. Comparativa de manejadores de bases de datos.

Algunas de las características que destacan del RDBMS Oracle a utilizar son:

- Facilidad para el uso de la herramienta.
- Multiplataforma.
- Rapidez de la aplicación.
- Concurrencia.
- Seguridad.

Como en el caso de los RDBMS también se compararán algunos lenguajes de programación, asumiendo de igual manera que las políticas de la compañía ya nos indicaron el lenguaje a utilizar, los lenguajes a comparar son los siguientes:

- PHP
- ASP
- Java



PHP (Hiptertext Preprocessor)

Fue creado en 1994 por Rasmus Lerdorf. Es un lenguaje de programación interpretado de alto nivel que puede ser incluido en las páginas HTML, así como ser ejecutado por un servidor web.

Algunas de las características que posee son:

- Es un lenguaje multiplataforma.
- Completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de programación orientada a objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Tiene manejo de excepciones (desde PHP5).



- El programador puede aplicar cualquier técnica de programación y/o desarrollo que le permita escribir código ordenado, estructurado y manejable.

ASP (Active Server Pages)

Es una tecnología creada por Microsoft para la creación dinámica de páginas web ofrecida junto a su servidor IIS (Internet Information Services). Apareció por primera vez en diciembre de 1996. No se trata de un lenguaje de programación sino de un marco en donde construir aplicaciones basadas en internet. La tecnología ASP se emplea principalmente para crear aplicaciones interactivas que funcionan en internet.

Algunas de las características que posee son:

- Es totalmente gratuito para Microsoft Windows NT o Windows 95/98.
- El código ASP se puede mezclar con el código HTML en la misma página.
- El código ASP se puede escribir con un simple editor de textos como el bloc de notas de Windows o UltraEdit.
- Como el código ASP se ejecuta en el servidor, y produce como salida código HTML puro, su resultado es entendible por todos los navegadores existentes.
- Mediante ASP se pueden manipular bases de datos (consultas, actualizaciones, borrados, etc.) de prácticamente cualquier plataforma, con tal de que proporcione un driver OLEDB u ODBC.
- Permite usar componentes escritos en otros lenguajes (C++, Visual Basic, Delphi), que se pueden llamar desde los guiones ASP.
- Sin modificar la instalación, los guiones ASP se pueden programar en JScript o VBScript, pero también existen otros lenguajes, como Perlscript y Rexx, que se pueden emplear para programar ASP.
- Se ha portado a la plataforma Java por Chili!Soft y Halcyon Software, lo que permite que ASP sea usado en casi cualquier sistema operativo.



- Permite acceder a bases de datos de una forma sencilla y rápida.
- Las páginas se generan dinámicamente mediante el código de scripts, (guiones).
- El código de script se ejecuta en el servidor, y no se depende del navegador que se emplee.
- Desde una página ASP se pueden ejecutar servidores OLE en el servidor de web, lo que abre un abanico de nuevas posibilidades sólo accesibles previamente usando CGI y filtros ISAPI: acceso a base de datos, acceso a ficheros, logging en el sistema, envío de correo, etc.

En la tabla 3.5.2 se muestran las comparaciones de las diferentes herramientas de desarrollo.

Característica	PHP	ASP	Java
Facilidad para manejo de interfaz de usuario	Bueno	Bueno	Bueno
Rapidez	Bueno	Bueno	Excelente
Componentes Active X	Excelente	Excelente	Bueno
Manejo de bases de datos	Bueno	Bueno	Excelente
Herramientas para el desarrollo de servicios web	Bueno	Excelente	Excelente
Elaboración de pantallas	Bueno	Bueno	Excelente
Costo	Bueno	Bueno	Bueno
Desempeño	Bueno	Bueno	Excelente
Facilidad de aprendizaje	Bueno	Bueno	Bueno
Documentación	Excelente	Bueno	Excelente

Tabla 3.5.2. Comparativa de herramientas de desarrollo.



Al tener en cuenta las características de las diferentes herramientas de desarrollo, Java definitivamente cumple con los requisitos necesarios para el proyecto.

Algunas de las características que destacan de Java son las siguientes:

- Es un lenguaje independiente de la plataforma, ya que puede funcionar en cualquier ordenador.
- Posee gran variedad de documentación en internet,
- No requiere licencia para poder utilizarlo.
- Facilidad de manejo de interfaz de usuario.
- Desempeño.