CAPÍTULO

4

DISEÑO Y CONSTRUCCIÓN DE LA APLICACIÓN

4.1 Arquitectura de la aplicación.

Modelo cliente/servidor.

El modelo cliente/servidor es el procesamiento cooperativo de la información por medio de un conjunto de procesadores, en el cual múltiples clientes, solicitan requerimientos a uno o más servidores centrales, como una arquitectura distribuida que permite a los usuarios finales obtener acceso a la información de forma transparente aún en entornos multiplataforma.

En el modelo usual, un servidor (daemon, se traduce como “demonio”), se activa y espera las solicitudes de los clientes. En este modelo se intenta proveer usabilidad, flexibilidad, interoperabilidad y escalabilidad en las comunicaciones.

En este modelo, el cliente envía un mensaje solicitando un determinado servicio a un servidor (hace una petición), y este envía uno o varios mensajes con la respuesta (provee el servicio), figura 4.1.1, En un sistema distribuido cada máquina puede cumplir el rol de servidor para algunas tareas y el rol de clientes para otras.

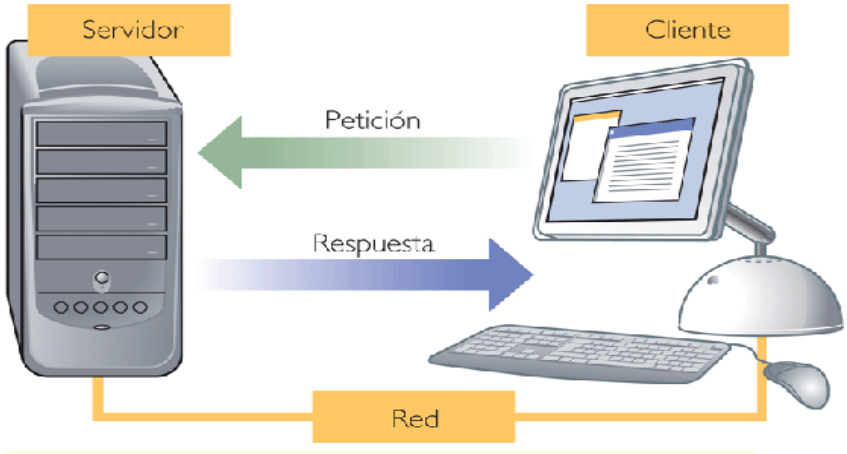

Figura 4.1.1 Modelo cliente/servidor

De aquí se deduce que el proceso cliente es quien inicia el diálogo, el proceso servidor es quien pasivamente espera a que lleguen peticiones de servicio y el middleware corresponde a la interfaz que provee la conectividad entre el cliente y el servicio para poder intercambiar mensajes.

Elementos Principales.

- Cliente

  Es el proceso que permite al usuario formular requerimientos y pasarlos al servidor, se le conoce con el nombre de front-end.

  Es el que maneja todas las funciones relacionadas con la manipulación y despliegue de datos, por lo que están desarrollados sobre plataformas que permiten construir interfaces gráficas de usuario (GUI), además de acceder a los servicios distribuidos en cualquier parte de la red. Sus funciones son:

  - Administrar la interfaz de usuario.
  - Interactuar con el usuario.
  - Procesar la lógica de la aplicación y hacer validaciones locales.
  - Generar requerimientos de base de datos.
  - Recibir resultados del servidor.
  - Formatear resultados.

- Servidor

  Es todo proceso que proporciona un servicio a otros. Es el proceso encargado de atender a múltiples clientes que hacen peticiones de algún recurso administrado por él. Al proceso servidor se le conoce con el término back-end. El servidor normalmente maneja todas las funciones relacionadas con la mayoría de las reglas del negocio y los recursos de datos. Sus principales funciones son:

  - Aceptar los requerimientos de bases de datos que hacen los clientes.
  - Procesar los requerimientos de bases de datos.
  - Formatear datos para transmitirlos a los clientes.
  - Procesar la lógica de la aplicación y realizar validaciones a nivel de bases de datos.

- Middleware

  Es un módulo intermedio que actúa como conductor entre sistemas permitiendo a cualquier usuario de sistemas de información comunicarse con varias fuentes de información que se encuentran conectadas por una red, siendo así el intermediario entre el cliente y el servidor y se ejecuta en ambas partes.

  Sus principales características son:

  - Simplifica el proceso de desarrollo de aplicaciones al independizar los entornos propietarios.
  - Permite la interconectividad de los sistemas de información del organismo.
  - Proporciona mayor control del negocio al poder contar con información procedente de distintas plataformas sobre el mismo soporte.
  - Facilita el desarrollo de sistemas complejos con diferentes tecnologías y arquitecturas.

Características de la arquitectura cliente/servidor.

Las características básicas de una arquitectura cliente/servidor son:

- Combinación de un cliente que interactúa con el usuario, y un servidor que interactúa con los recursos compartidos. El proceso del cliente proporciona la interfaz entre el usuario y el resto del sistema. El proceso del servidor actúa como un motor de software que maneja recursos compartidos tales como bases de datos, impresoras, módems, etc.
- Las tareas del cliente y del servidor tienen diferentes requerimientos en cuanto a recursos de cómputo como velocidad del procesador, memoria, velocidad y capacidades del disco.
- Se establece una relación entre procesos distintos, los cuales pueden ser ejecutados en la misma máquina o en máquinas diferentes distribuidas a lo largo de la red.
- Existe una clara distinción de funciones basada en el concepto de “servicio”, que se establece entre clientes y servidores.
- La relación establecida puede ser de muchos a uno, en la que un servidor puede dar servicio a muchos clientes, regulando su acceso a recursos compartidos.
- Los clientes corresponden a procesos activos en cuanto a que son éstos los que hacen peticiones de servicio a los servidores. Estos últimos tienen un carácter pasivo ya que esperan las peticiones de los clientes.
- No existe otra relación entre clientes y servidores que no sea la que se establece a través del intercambio de mensajes entre ambos. El mensaje es el mecanismo para la petición y entrega de solicitudes de servicio.
- La plataforma de hardware y el sistema operativo del cliente y del servidor no son siempre la misma. Una de las principales ventajas de esta arquitectura es la posibilidad de conectar clientes y servidores independientemente de sus plataformas.

- El concepto de escalabilidad tanto horizontal como vertical es aplicable a cualquier sistema cliente/servidor. La escalabilidad horizontal permite agregar más estaciones de trabajo activas sin afectar significativamente el rendimiento. La escalabilidad vertical permite mejorar las características del servidor o agregar múltiples servidores.

Ventajas de la arquitectura cliente/servidor.

Entre las principales ventajas del sistema cliente/servidor están:

- La existencia de plataformas de hardware cada vez más baratas. Esta constituye a su vez una de las más palpables ventajas de este esquema, la posibilidad de utilizar máquinas considerablemente más baratas que las requeridas por una solución centralizada, basada en sistemas grandes. Además se pueden utilizar componentes, tanto de hardware como de software, de varios fabricantes lo cual contribuye a la reducción de costos y favorece la flexibilidad en la implantación y actualización de soluciones.
- Facilita la integración entre sistemas diferentes y comparte información permitiendo, que las máquinas ya existentes puedan ser utilizadas, pero utilizando interfaces más amigables al usuario. De esta manera podemos integrar PC´s con sistemas medianos y grandes, sin necesidad de que todos tengan que utilizar el mismo sistema operacional.
- Al favorecer el uso de interfaces graficas interactivas, los sistemas construidos bajo este esquema tienen mayor interacción con el usuario. En el uso de interfaces gráficas, no es siempre necesario transmitir información gráfica por la red pues esta puede residir en el cliente, lo cual permite aprovechar mejor el ancho de banda de la red.
- Es más rápido el mantenimiento y el desarrollo de aplicaciones, pues se pueden emplear las herramientas existentes (servidores de SQL).

- Facilita la integración de nuevas tecnologías y el crecimiento de la infraestructura computacional, favoreciendo así la escalabilidad de las soluciones.
- Proporciona a los diferentes departamentos de una organización, soluciones locales pero permitiendo la integración de la información relevante a nivel global.

Desventajas de la arquitectura cliente/servidor.

Entre las principales desventajas están:

- El mantenimiento de los sistemas es más difícil pues implica la interacción de diferentes partes del hardware y del software, distribuidas por diferentes proveedores, lo cual dificulta el diagnostico de fallas.
- Se cuenta con muy escasas herramientas para la administración y ajuste del desempeño de los sistemas.
- Es importante que los clientes y servidores utilicen el mismo mecanismo, lo cual implica que deben tener mecanismos generales que existan en diferentes plataformas.
- Hay que tener estrategias para el manejo de errores y para mantener la consistencia de los datos.

Arquitectura cliente/servidor de dos capas.

Consiste en una capa de presentación y lógica de la aplicación; y la otra de la base de datos figura 4.1.2. Normalmente esta arquitectura se utiliza en las siguientes situaciones:

- Cuando se requiera poco procesamiento de datos en la organización.
- Cuando se tiene una base de datos centralizada en un solo servidor.
- Cuando la base de datos es relativamente estática.
- Cuando se requiere un mantenimiento mínimo.

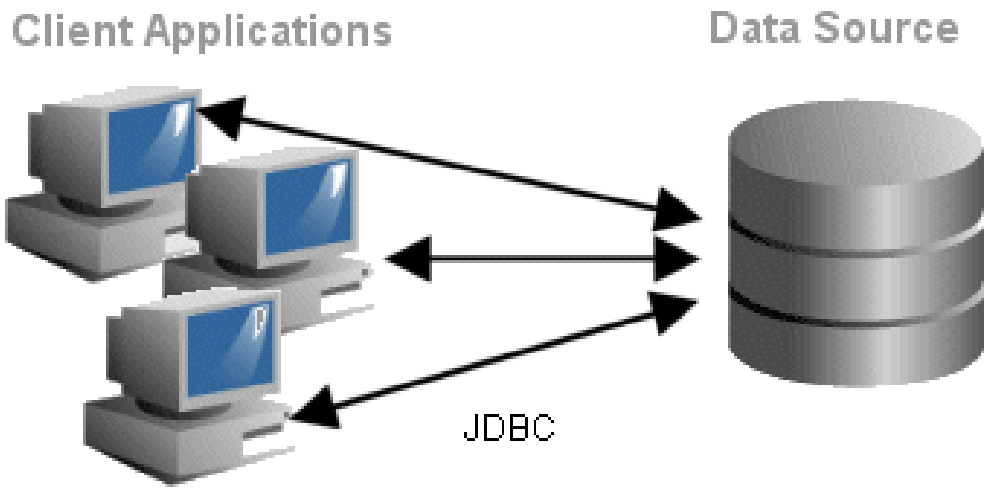

Figura 4.1.2 Cliente/servidor de dos capas.

UML (Lenguaje unificado de modelado).

Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un leguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, también aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los elementos o componentes en el sistema y para documentar y construir. Es el lenguaje en el que esta descrito el modelo, figura 4.1.3.

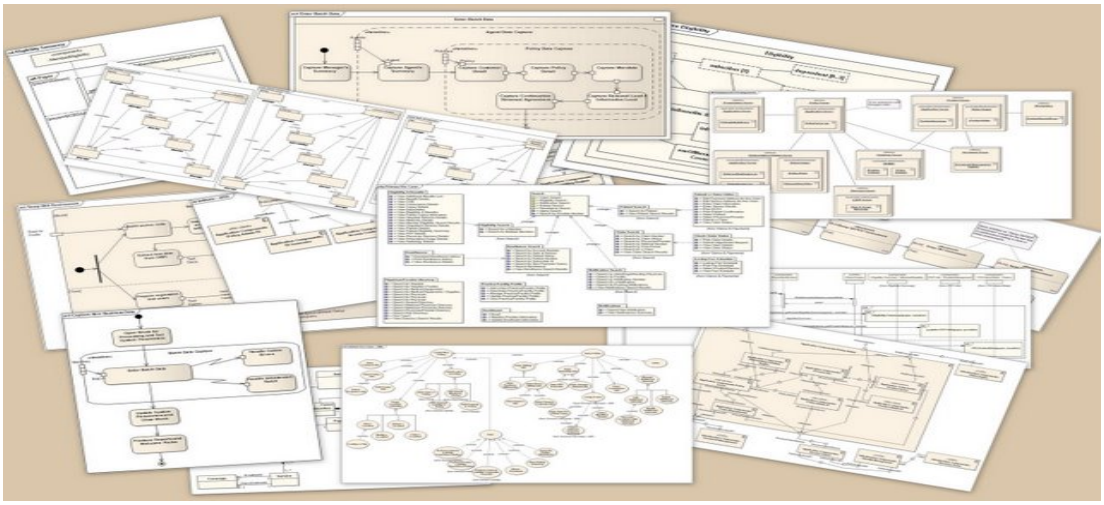

Figura 4.1.3 Diagramas UML.

Se puede aplicar en el desarrollo de software, entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software, pero no especifica qué metodología o proceso usar.

Los diagramas a representar dependerán del sistema a desarrollar. Está recomendaciones se deberán adaptar a las características de cada desarrollo.

En UML hay trece tipos de diagramas diferentes, los cuales son:

- Los diagramas de estructura: Enfatizan en los elementos que deben existir en el sistema modelado.
    - Diagrama de clases.
    - Diagrama de componentes.
    - Diagrama de objetos.
    - Diagrama de estructura compuesta.
    - Diagrama de despliegue.
    - Diagrama de paquetes.
- Los diagramas de comportamiento: Enfatizan en lo que debe suceder en el sistema modelado.
    - Diagrama de actividades.
    - Diagrama de casos de uso.
    - Diagrama de estados.

- Los diagramas de interacción: Son un subtipo de diagramas de comportamiento, que enfatiza sobre el flujo de control y de datos entre los elementos del sistema modelado:
    - Diagrama de secuencia.
    - Diagrama de comunicación.
    - Diagrama de tiempos.
    - Diagrama global de interacciones o diagrama de vista de interacción.

Diagrama de casos de uso.

Un caso de uso es la descripción de la secuencia de interacciones que se realizan entre los actores y el sistema. Presenta las siguientes características:

- Es siempre iniciado por un actor y provee de valores a un actor.
- Es una tarea específica que se realiza tras una orden de algún agente externo, ya sea desde una petición de un actor o bien desde la invocación desde otro caso de uso.
- El nombre del caso de uso siempre esta expresado desde el punto de vista del actor y no desde el punto de vista del sistema.
- Cada caso de uso está acotado al uso de una determinada funcionalidad en el sistema.

Sus elementos (figura 4.1.4), son:

- Actores: Es algo con comportamiento, como una persona, un sistema informático u organización y que realiza algún tipo de interacción con el sistema. Se representa mediante una figura humana.
- Casos de Uso: Es una descripción de la secuencia de interacciones que se producen entre un actor y un sistema, cuando el actor usa el sistema para llevar a cabo una tarea específica. Expresa una unidad coherente de funcionalidad y se representa mediante una elipse con el nombre del caso de uso en su interior.

- Inclusión (include o use): Es una forma de interacción o creación, un caso de uso dado puede “incluir” otro. El primer caso de uso a menudo depende del resultado del caso de uso incluido. Esto es útil para extraer comportamientos verdaderamente comunes desde múltiples casos de uso a una descripción individual. Se puede decir que va desde padre a hijo.
- Extensión (Extend): Es otra forma de interacción, un caso de uso dado, puede extender a otro. Esta relación indica que el comportamiento del caso de la extensión se utiliza en el caso de uso. Un caso de uso a otro caso siempre debe tener extensión o inclusión.
- Generalización: Es la actividad de identificar elementos en común entre conceptos y definir las relaciones de una superclases y subclase. Es una relación de generalización/especialización.

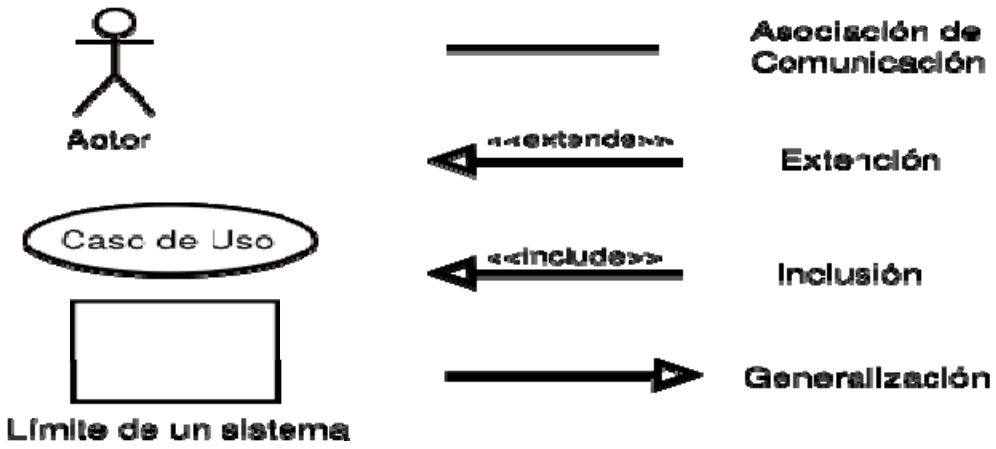

Figura 4.1.4 Componentes de un diagrama de caso de uso.

Justificaciones.

El Sistema de Mediación como ya se ha dicho, utiliza los CDRs que se extraen de las centrales telefónicas, los cuales entran en una serie de procesos que no son tangibles, ni visibles para el usuario. Simplemente es una aplicación que tiene como objetivo procesar los CDRs, estandarizarlos y guardarlos en una base de

datos, sin que el usuario intervenga, además de que no arroja resultados que el usuario pueda manipular, ya que los CDRs procesados solamente se guardan en una base de datos para después ser utilizados por otros sistemas, para fines propios.

Es por esto que se utilizará la arquitectura cliente/servidor de dos capas simplemente porque el sistema no tiene interacción con el usuario de ningún tipo. Se podría decir que el sistema es automático, ya que el usuario en ningún momento introduce datos ni peticiones, en este sistema el cliente no existe, solo existe comunicación entre el servidor y el sistema, alimentados de las centrales y aterrizados en una base de datos.

Para modelar el sistema de mediación, se utilizará UML, con el diagrama de casos de uso, ya que este diagrama nos permitirá visualizar un entorno general de cómo interactúan los actores del sistema, de esta forma, se puede conocer cómo se comporta cierta parte del sistema, mas no, como está implantada la estructura que lo define. Indica lo que cierta parte del sistema debe de hacer cuando ocurra cierto proceso. También permitirá centrarse en lo que se espera lograr al utilizar el sistema así como una estimación más exacta para determinar tiempo, recursos y prioridades en la dosificación de esfuerzo de desarrollo y un mayor control para mantener las sucesivas revisiones de los programas.

4.2 Diagramación

4.2.1 Diagrama de casos de uso.

Estos son los casos de uso más relevantes:

- Recepción de CDRs
- Análisis de CDRs
- Balanceador de tramas.

- Parser SIMED.
- Monitoreo de disponibilidad.
- Procesamiento facturación a clientes.
- Generar información para facturación a clientes corporativos.
- Generar información para facturación a otros operadores.
- Carga información externa.
- Diagrama General.

Recepción de CDRs.

La conexión con las MSC es necesaria para recolectar los CDRs que se generan en el momento de realizar las llamadas entre equipos celulares o teléfonos fijos, es importante obtener la información mediante el protocolo que permite la transferencia por bloques, así como la recolección de los archivos mediante FTP Ver figura 4.2.1

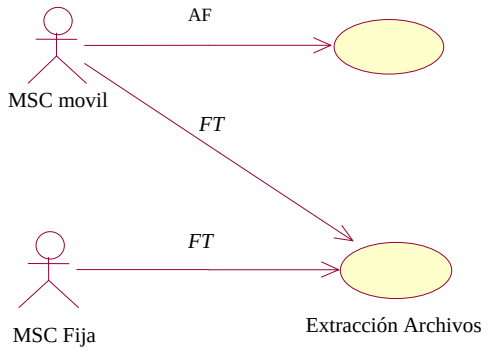

Figura 4.2.1 Diagrama de recepción de CDRs

Análisis de CDRs.

Es necesario identificar el tipo de CDR, para en su caso poder informar a otras compañías de telefonía sobre algún CDR de un equipo que utiliza nuestra red. Se verifica que cumpla con el requerimiento de la versión que establece el MSC. En la figura 4.2.2 se muestra el diagrama de caso de uso para el análisis de CDRs.

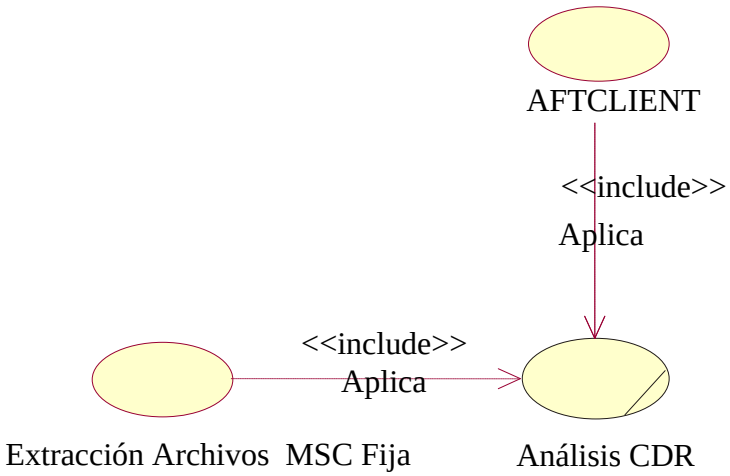

Figura 4.2.2 Diagrama de análisis de CDRs

Balanceador de tramas.

Esta herramienta funciona manteniendo la conexión con los diferentes servidores y también detecta automáticamente cuando un socket destino se encuentra sin poder recibir transacciones, distribuyendo los CDRs a los otros destinos disponibles permitiendo que se mantenga el servicio. La figura 4.2.3 muestra el diagrama de caso de uso para el balanceador de tramas.

Figura 4.2.3 Diagrama de balanceador de tramas

Parser SIMED.

El parser SIMED compone el proceso medular del Sistema de Mediación, debido a que realiza las operaciones de analizar el CDR que se obtuvo de los diferentes medios de conmutación. El parser SIMED analiza los CDRs agregando información adicional al CDR permitiendo clasificar la trama para su inserción en la base de datos. Véase la figura 4.2.4 con el diagrama de parser SIMED.

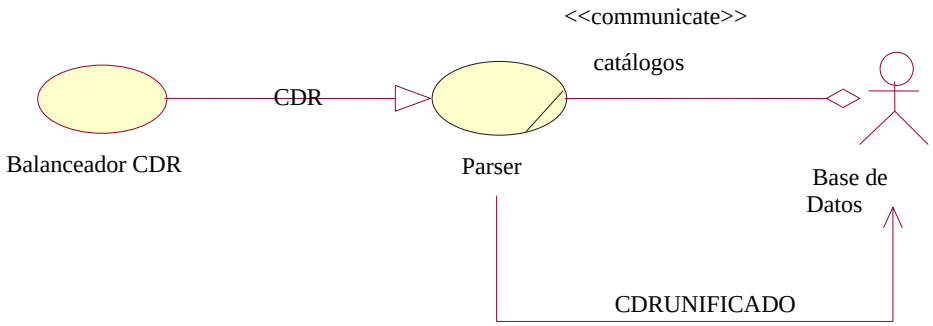

Figura 4.2.4 Diagrama de parser SIMED

Monitoreo de disponibilidad.

Existen diversas herramientas propias que permiten realizar verificaciones constantes con el fin de mantener la disponibilidad de los diversos objetos que intervienen en el flujo del CDR, desde su llegada al AFTCLIENT hasta su impacto a la base de datos. La figura 4.2.5 muestra el diagrama de caso uso de este caso.

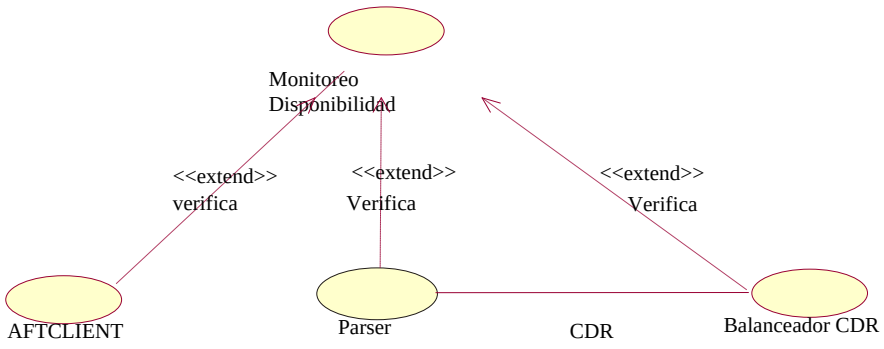

Figura 4.2.5 Diagrama de monitoreo de disponibilidad

Procesamiento facturación a clientes.

La facturación a clientes tiene un formato especificado por otras compañías telefónicas derivado del Intercambio de capacidad y de la interconexión, para efecto de conciliación y facturación, es necesario traducir este formato, a un formato propio y en una base de datos. En la figura 4.2.6 se muestra su diagrama de caso de uso.

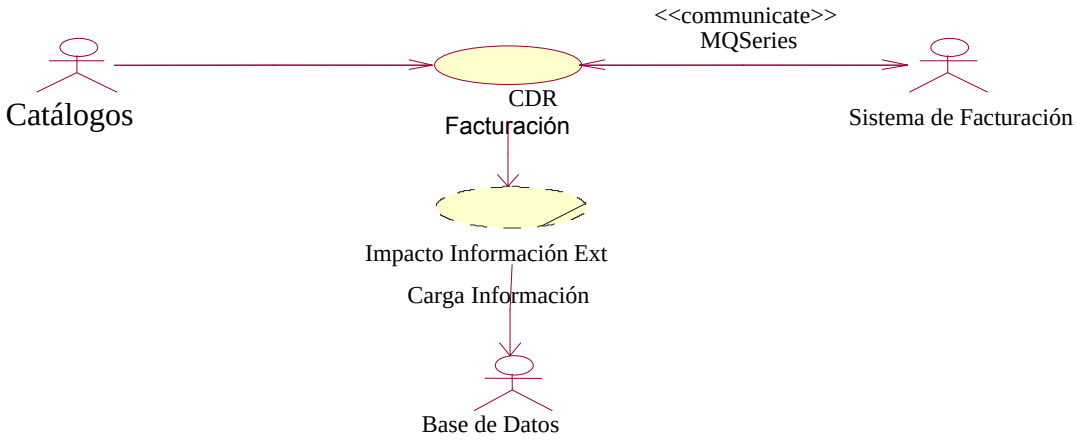

Figura 4.2.6 Diagrama de procesamiento facturación a clientes

Generar Información para facturación a clientes corporativos

Existen registros de CDRs los cuales no son informados a la caja de prepago y es necesario que se le reporten estas llamadas. La información que es necesario impactar de manera mensual es para aquellas series que correspondan al servicio de telefonía fija o post pago móvil. Ver figura 4.2.7 con el diagrama de caso de uso.

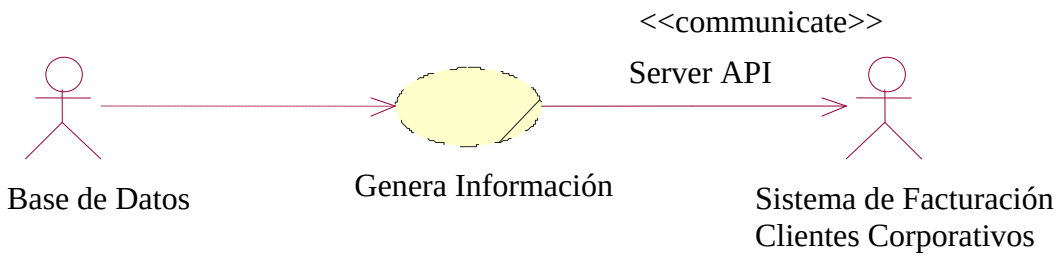

Figura 4.2.7 Diagrama de información facturación a clientes corporativos

Generar información para facturación a otros operadores

Con el fin de cumplir con los requerimientos para la interconexión e intercambio de tecnología cubriendo el servicio de red celular, es necesario que la información que se genera de los diferentes mecanismos de conmutación de servicios, sea de utilidad para la generación de facturas las cuales se utilizan para el cobro o conciliación con otros operadores. La figura 4.2.8 muestra el diagrama correspondiente.

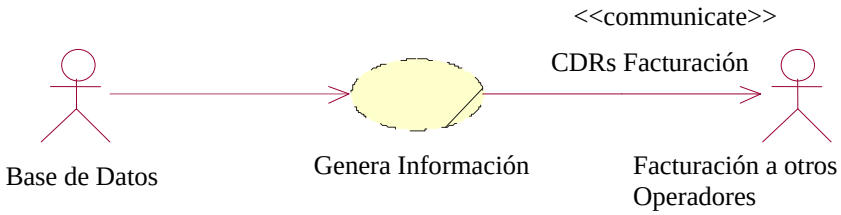

Figura 4.2.8 Diagrama de facturación a otros operadores

Carga de información externa

Con el objetivo de cumplir los requerimientos de información para los distintos usuarios que tiene el Sistema de Mediación es necesario obtener esta información de diferentes plataformas como puede ser el proceso de facturación de clientes con otras compañías telefónicas o las plataformas de envío de datos o multimedia, esto puede seguir creciendo dependiendo de los diferentes equipos que soportan

los nuevos productos que se ofrecen. La figura 4.2.9 muestra el diagrama de carga de información externa.

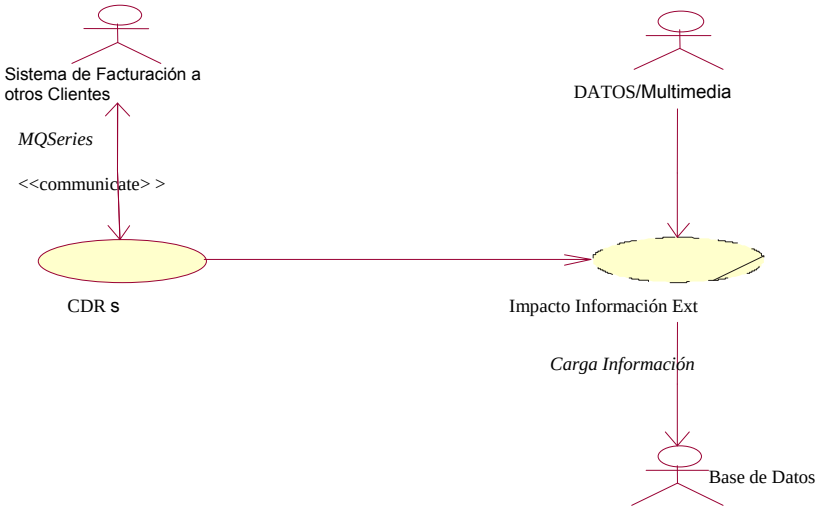

Figura 4.2.9 Diagrama de carga de información externa

Diagrama general.

A continuación la figura 4.2.10 muestra el diagrama unificado con todos los casos de uso mencionados.

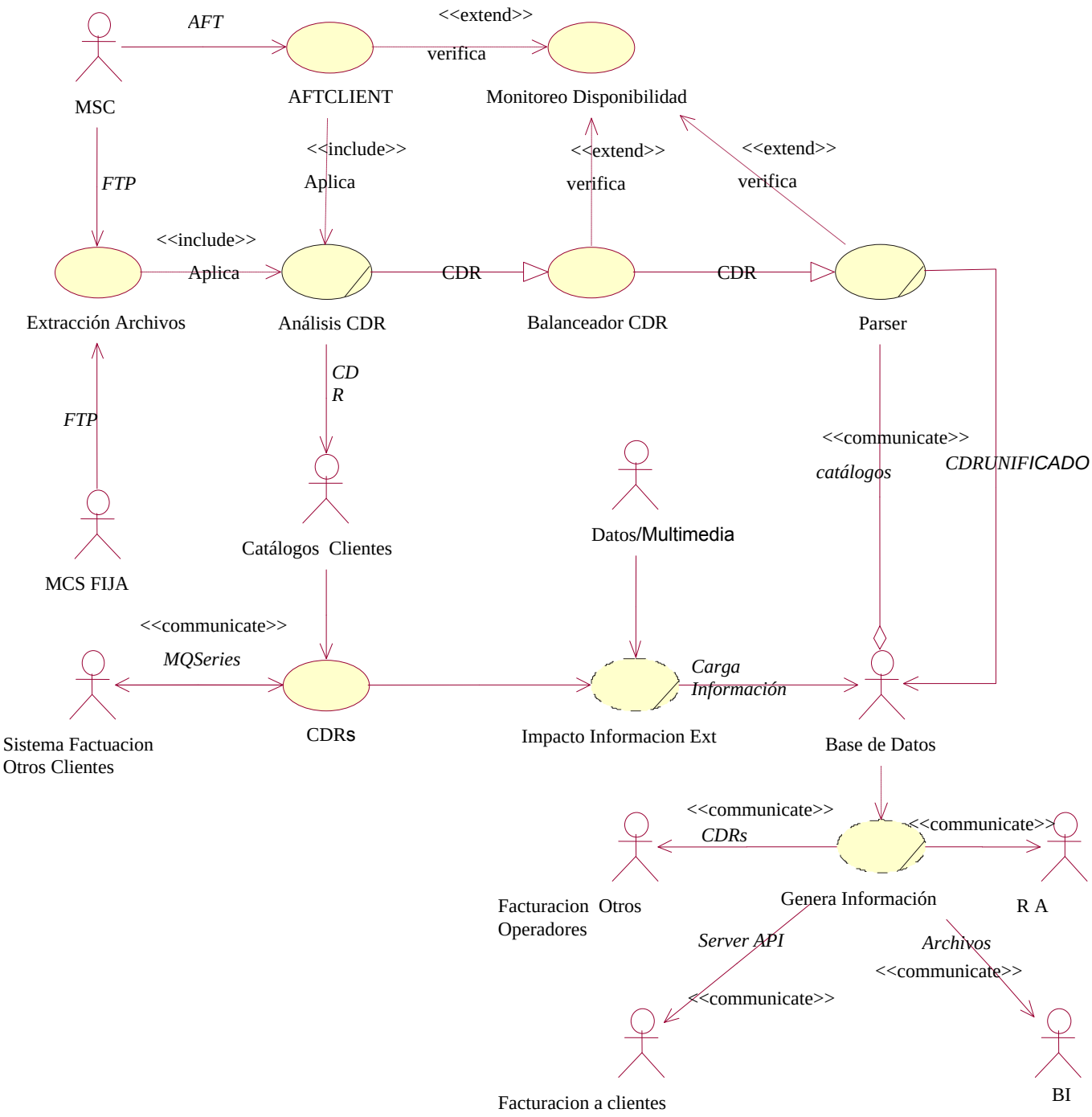

Figura 4.2.10 Diagrama general

4.2.2 Diagramas de secuencia.

Nombre del caso de uso: Recepción de trama de CDR.

El caso de uso comienza cuando se ejecuta la aplicación AFTCLIENT especificando el MSC Móvil con el cual va a interactuar posteriormente establece una conexión permanente mediante un socket y comienza a verificar cual es el archivo AMA que está procesando. En la figura 4.2.3.1 se muestra el diagrama de secuencia de la recepción de trama de CDR.

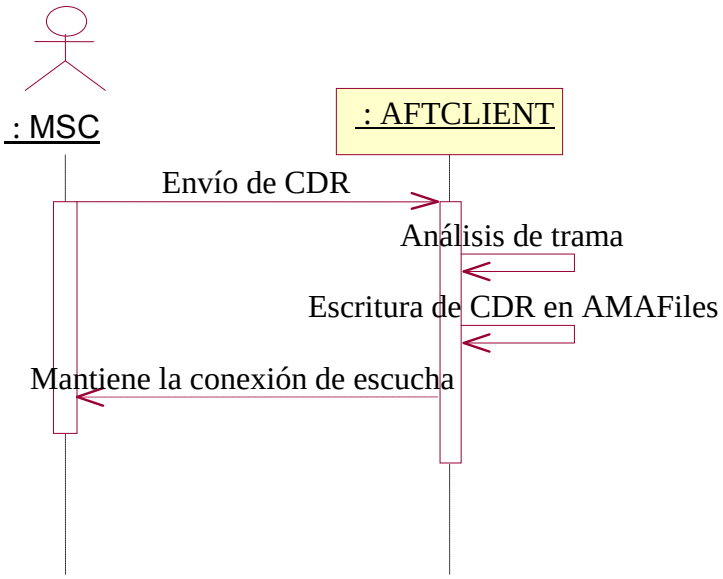

Figura 4.2.2.1 Diagrama de recepción de trama CDR

Nombre del caso de uso: Recepción de CDR MSC fija.

El caso de uso comienza cada hora cuando se realiza la conexión por FTP con el depositario de archivos del MSC fija, el proceso verifica cual es el último archivo extraído en la anterior conexión y verifica que exista un nuevo archivo para extraerlo por FTP. Revisa el formato de los registros que contiene el archivo y solo se analizan aquellos que tienen estatus de “STOP”, debido a que es el último

registro que se genera cuando se conmuta una llamada. En la figura 4.2.2.2 se muestra el diagrama de secuencia.

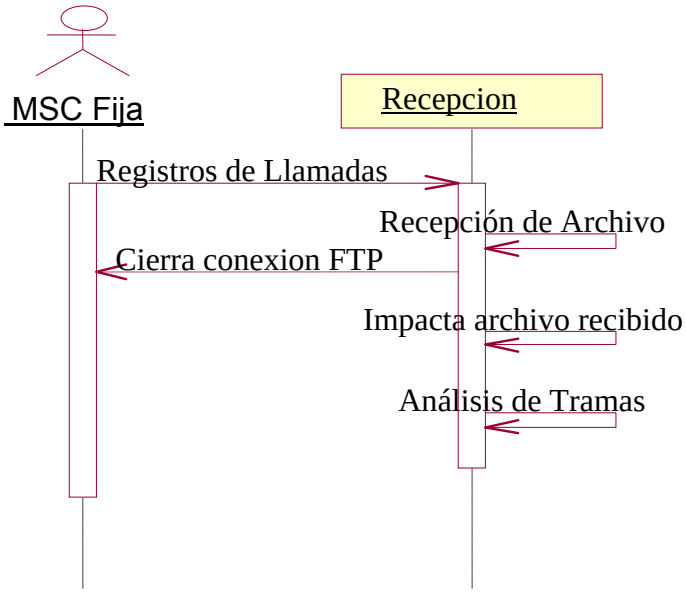

Figura 4.2.2.2 Diagrama de secuencia de recepción de CDRs MSC fija

Nombre del caso de uso: Balanceador de trama CDR.

Levanta la conexión del proceso hacia el parser, el análisis que está incluido en el proceso del AFTCLIENT envía la trama de CDR al balanceador, posteriormente el algoritmo de balanceador toma la trama de CDR en el formato enviado. El balanceador toma una dirección de destino para su envío y almacena la trama de manera temporal en una tabla asignada exclusiva para el parser SIMED destino.
Cuando el balanceador reciba respuesta de recepción por parte del parser SIMED elimina la trama de la tabla temporal, cuando el balanceador detecta que un parser SIMED se encuentra no disponible realiza otra revisión de parser SIMED disponibles para reenviar la trama CDR. Ver el diagrama de secuencia de la figura 4.2.2.3

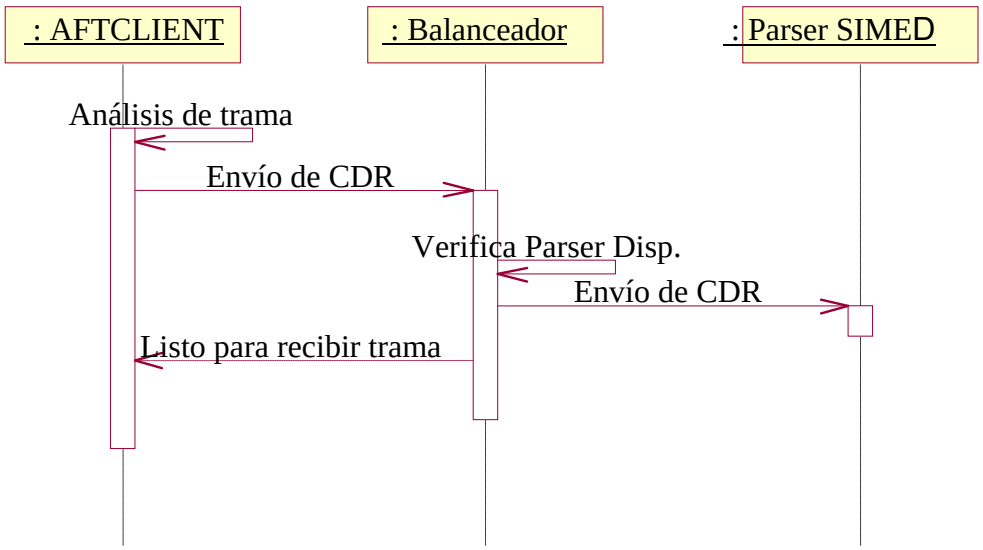

Figura 4.2.2.3 Diagrama de balanceador de trama CDR

Nombre del caso de uso: Parser SIMED.

Al levantar el sistema se cargan los catálogos necesarios para completar la información de los CDRs que ayudan en la clasificación del mismo posteriormente recibe la trama de CDR por parte del balanceador, la trama CDR de entrada se almacena en la cola de entrada El sistema del parser SIMED tiene configurados 50 hilos de conexión para la cola de entrada.

El proceso del parser SIMED descompone la trama de CDR para realizar el análisis de aquellos CDR que correspondan a llamadas completadas analizando el campo de duración el cual deberá de ser 0 o mayor. Después de realizar el proceso de incrementar la información con los catálogos y de realizar la clasificación del CDR, se deposita en la cola de salida para su impacto en la base de datos. En la figura 4.2.2.4 se muestra el diagrama de secuencia del parser SIMED.

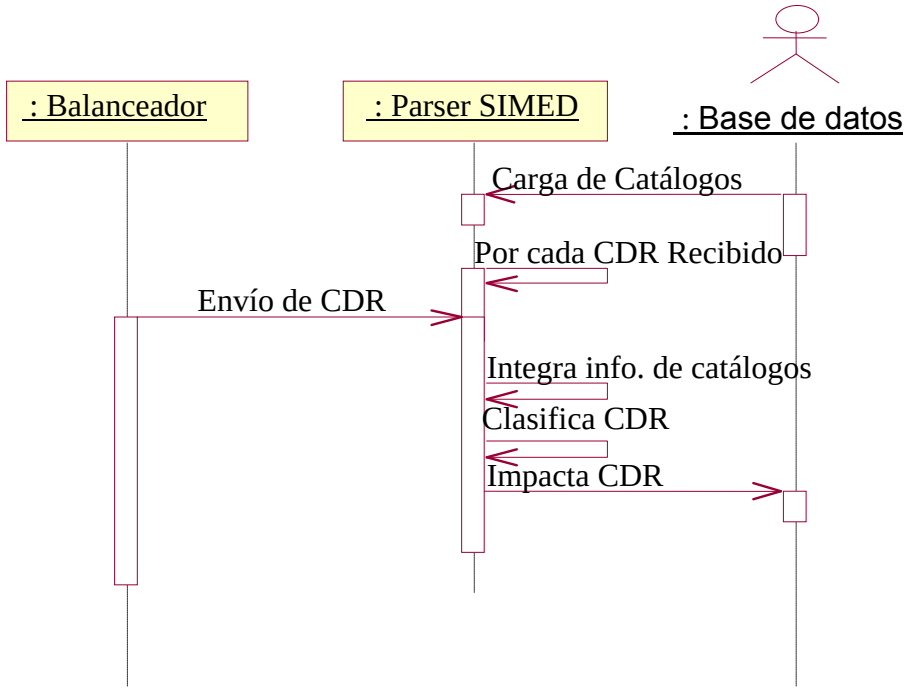

Figura 4.2.2.4 Diagrama de parser SIME

Nombre del caso de uso: Monitoreo de disponibilidad.

Las herramientas que monitorean el adecuado funcionamiento y disponibilidad del proceso de AFTCLIENT verificaran que se encuentre en ejecución el proceso. La salida se deposita en un archivo log para su análisis y el proceso analiza el archivo para verificar que se encuentre en ejecución el AFTCLIENT asociado a cada uno de los MSC fija. El monitoreo realiza el borrado de los archivos AMA para mantener el espacio disponible, solo se mantiene un periodo de 10 días de histórico.

La herramienta de monitoreo para el balanceo de las tramas CDR es necesario verificar que se encuentre en ejecución, comienza cuando analiza el archivo log de salida para verificar si se encuentra actualmente en ejecución el balanceador.

La herramienta de monitoreo de disponibilidad para el parser SIMED es necesario asegurar que se encuentre en ejecución, así como verificar el número de registros encolados en las colas de entrada y salida del parser SIMED.

La herramienta tendrá que analizar que los tres procesos se encuentren en ejecución, en caso contrario deberá de levantar el proceso faltante. Ver figura 4.2.2.5 con el diagrama de secuencia de este proceso.

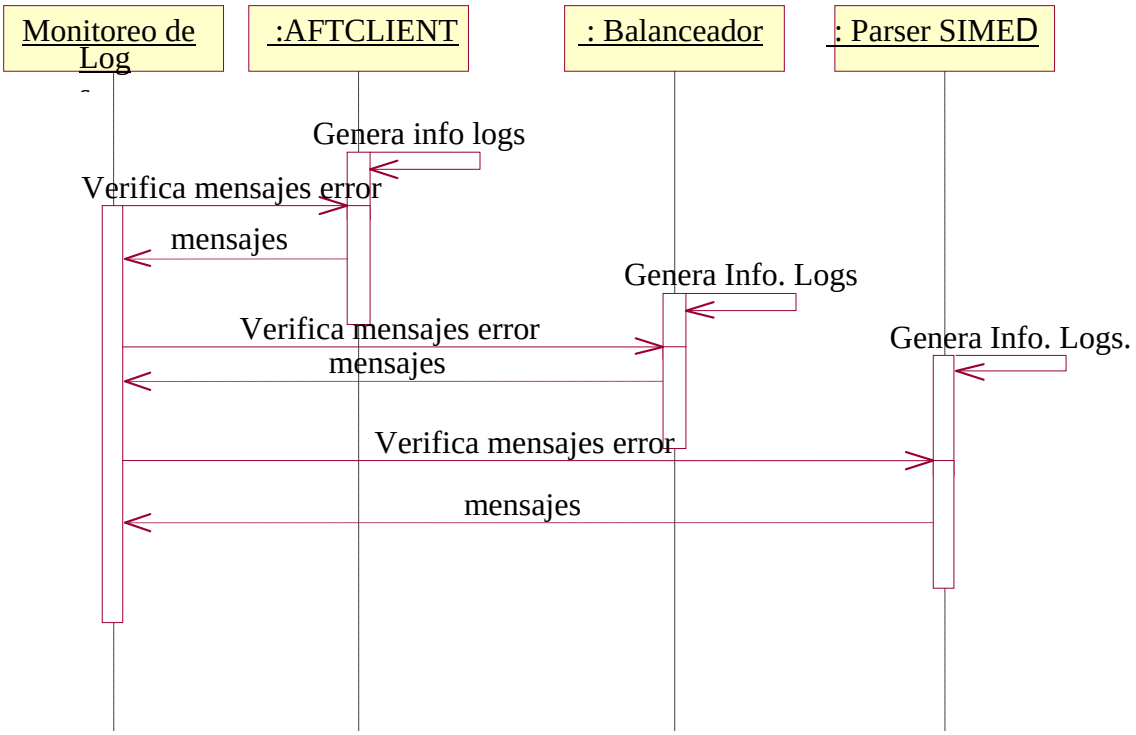

Figura 4.2.2.5 Diagrama de monitoreo de disponibilidad

Nombre del caso de uso: Carga de información externa.

El caso de uso comienza cuando se cuenta con los archivos correspondientes al proceso de facturación a clientes, estos contienen la información que se genera en nuestra red y la de otras compañías telefónicas. Estos archivos son depositados en el servidor en donde se encuentra la base de datos, el proceso tiene objetos de la base de datos indexados a archivos con nombres genéricos para su lectura. Por lo que los archivos recibidos se renombran para que se puedan acceder por la base de datos, el caso de uso termina con la inserción de datos en la base de datos. Para el caso de los datos y multimedia comienza con la recepción de los

archivos que corresponden a la información generada por estas plataformas, el formato que se deben de entregar deberá cumplir con el contrato de formato. El proceso realiza un análisis de los registros para ser insertados en tablas de la base de datos. La figura 4.2.2.6 muestra el diagrama de secuencia para la carga de información externa.

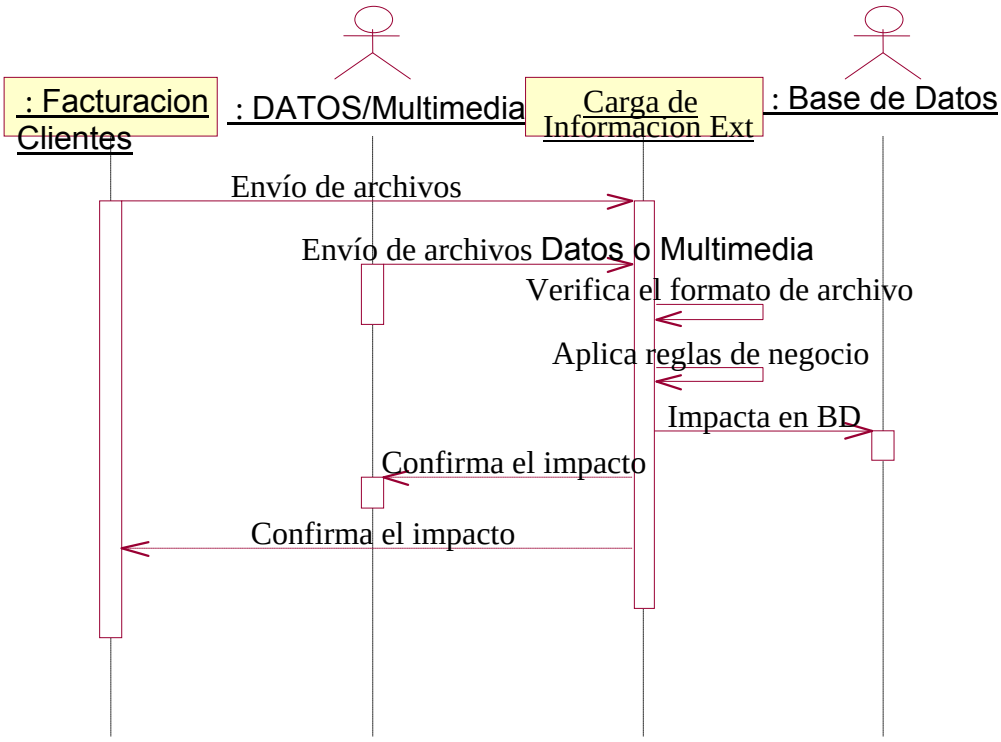

Figura 4.2.2.6 Diagrama de carga de información externa

Nombre del caso de uso: Generación de Información para facturación a clientes corporativos.

Se verifica cifras de la información impactada en la base de datos de los archivos de MSC fija, a partir de la información de los corporativos que se tienen registrados en la base de datos se extraen las llamadas generadas por los mismos, después se crean las tramas con el formato específico para su impacto al servidor, las tramas se depositan en la BD del SIMED en donde se clasifica por mes. Las tramas se generan de entrada y de salida de llamadas una vez que crea

la información, se forma un archivo con las tramas que se impactaran al servidor. El archivo se divide en tantos hilos de impacto que haya definido el área de facturación a clientes corporativos, para no afectar a la producción diaria. Se programa el socket que envía las tramas e impacta al servidor. Para la generación de las tramas de pospago, es necesario verificar cifras de ingresos en la base de datos de los archivos recibidos de los CDRs, con la información que se tiene registrada de los usuarios de pospago, se extraen todas las llamadas realizadas en el periodo del mes a procesar. Ver figura 4.2.2.7 con el diagrama de secuencia de este proceso.

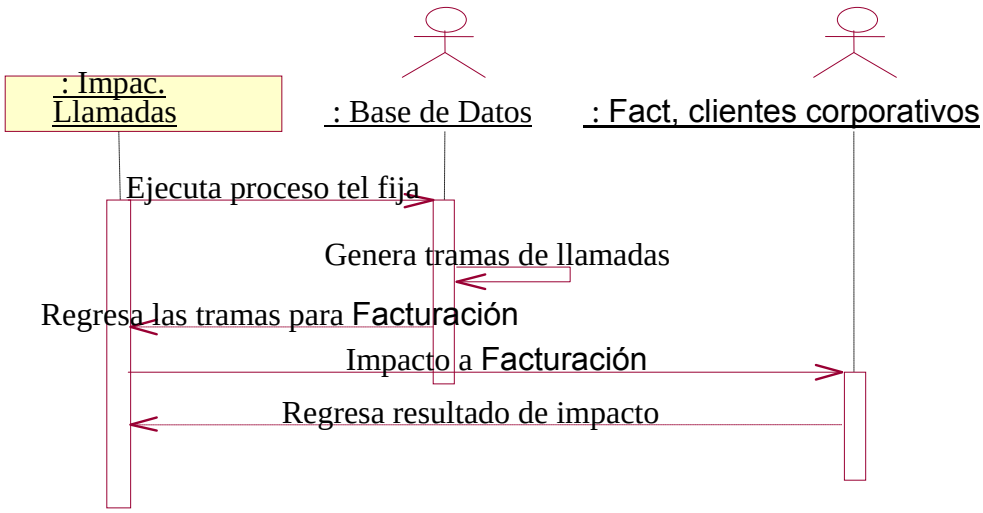

Figura 4.2.2.7 Diagrama de Generación de Información para Pospago

Nombre del caso de uso: Generación de información para facturación a otros operadores.

La información que se inserta en las tablas de mediación se replica aplicando reglas de negocio a las tablas de facturación a otros operadores e Intercambio de capacidad. La información es clasificada basándose en las reglas definidas por cada una de las áreas, la información se clasifica y deposita directamente en sus BD, teniendo la información generada el proceso deposita en archivos los diferentes anexos. Los archivos deberán de cumplir con el formato establecido en los contratos que se hayan firmado. Finaliza el proceso generando todos los

anexos necesarios por el área de interconexión. La figura 4.2.2.8 muestra el diagrama de secuencia para la carga de información externa.

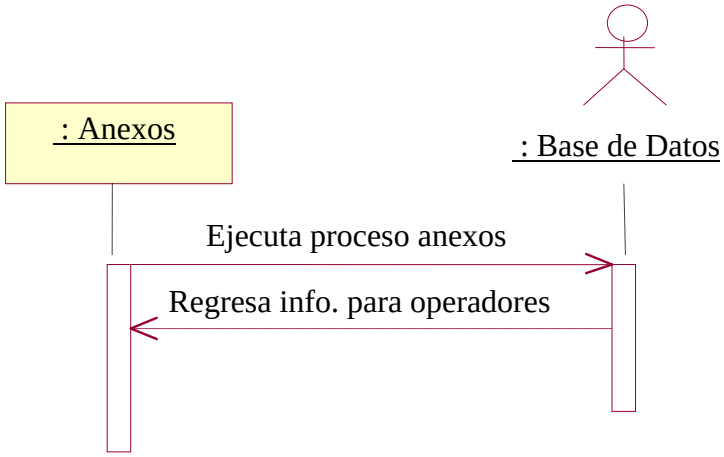

Figura 4.2.2.8 Diagrama de generación información para interconexión.

4.2.3. Diagrama entidad- relación.

El modelo entidad-relación es el modelo conceptual más utilizado para el diseño conceptual de bases de datos. Fue introducido por Peter Chen en 1976. El modelo entidad-relación está formado por un conjunto de conceptos que permiten describir la realidad mediante un conjunto de representaciones gráficas y lingüísticas.

Originalmente, el modelo entidad-relación sólo incluía los conceptos de entidad, relación y atributo. Más tarde, se añadieron otros conceptos, como los atributos compuestos y las jerarquías de generalización, en lo que se ha denominado modelo entidad-relación extendido, figura 4.2.3.1.

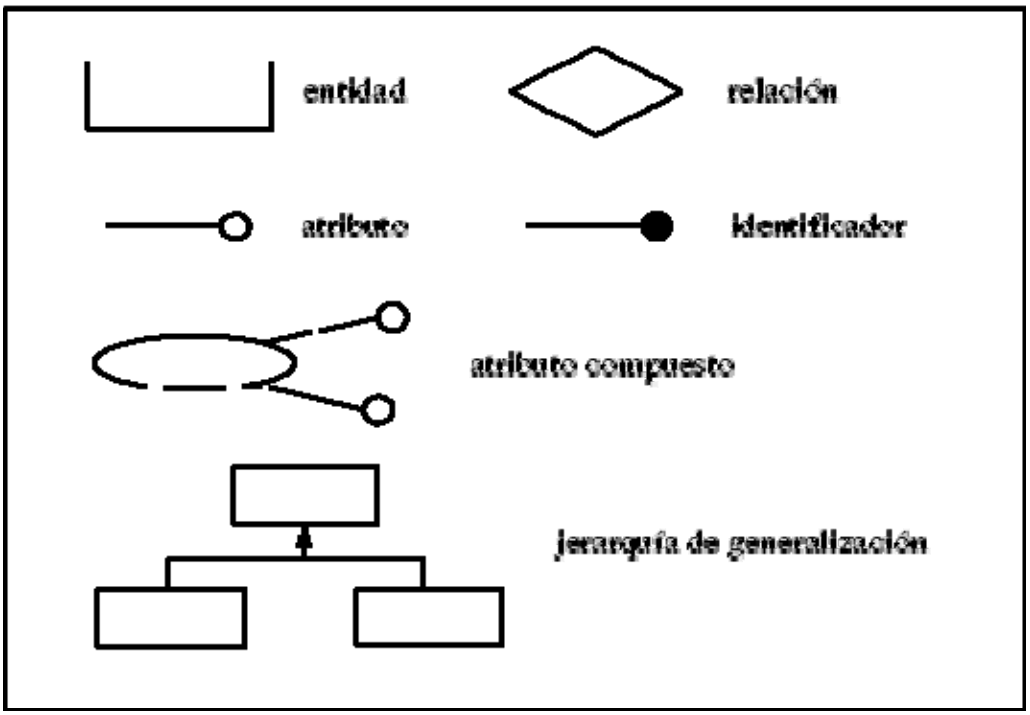

Figura 4.2.3.1 Conceptos del modelo entidad-relación extendido.

Entidad.

Cualquier tipo de objeto o concepto sobre el que se recoge información: cosa, persona, concepto abstracto o suceso. Las entidades se representan gráficamente mediante rectángulos y su nombre aparece en el interior. Un nombre de entidad sólo puede aparecer una vez en el esquema conceptual.

Hay dos tipos de entidades: fuertes y débiles. Una entidad débil es una entidad cuya existencia depende de la existencia de otra entidad. Una entidad fuerte es una entidad que no es débil.

Relación (interrelación).

Es una correspondencia o asociación entre dos o más entidades. Cada relación tiene un nombre que describe su función. Las relaciones se representan gráficamente mediante rombos y su nombre aparece en el interior.

Las entidades que están involucradas en una determinada relación se denominan entidades participantes. El número de participantes en una relación es lo que se denomina grado de la relación. Por lo tanto, una relación en la que participan dos entidades es una relación binaria; si son tres las entidades participantes, la relación es ternaria; etc.

Una relación recursiva es una relación donde la misma entidad participa más de una vez en la relación con distintos papeles. El nombre de estos papeles es importante para determinar la función de cada participación.

La cardinalidad con la que una entidad participa en una relación especifica el número mínimo y el número máximo de correspondencias en las que puede tomar parte cada ocurrencia de dicha entidad. La participación de una entidad en una relación es obligatoria (total) si la existencia de cada una de sus ocurrencias requiere la existencia de, al menos, una ocurrencia de la otra entidad participante. Si no, la participación es opcional (parcial). Las reglas que definen la cardinalidad de las relaciones son las reglas de negocio.

A veces, surgen problemas cuando se está diseñado un esquema conceptual. Estos problemas, denominados trampas, suelen producirse a causa de una mala interpretación en el significado de alguna relación, por lo que es importante comprobar que el esquema conceptual carece de dichas trampas. En general, para encontrar las trampas, hay que asegurarse de que se entiende completamente el significado de cada relación. Si no se entienden las relaciones, se puede crear un esquema que no represente fielmente la realidad.

Una de las trampas que pueden encontrarse ocurre cuando el esquema representa una relación entre entidades, pero el camino entre algunas de sus ocurrencias es ambiguo. El modo de resolverla es reestructurando el esquema para representar la asociación entre las entidades correctamente.

Otra de las trampas sucede cuando un esquema sugiere la existencia de una relación entre entidades, pero el camino entre una y otra no existe para algunas de sus ocurrencias. En este caso, se produce una pérdida de información que se puede subsanar introduciendo la relación que sugería el esquema y que no estaba representada.

Atributo.

Es una característica de interés o un hecho sobre una entidad o sobre una relación. Los atributos representan las propiedades básicas de las entidades y de las relaciones. Toda la información extensiva es portada por los atributos. Gráficamente, se representan mediante bolitas que cuelgan de las entidades o relaciones a las que pertenecen.

Cada atributo tiene un conjunto de valores asociados denominado dominio. El dominio define todos los valores posibles que puede tomar un atributo. Puede haber varios atributos definidos sobre un mismo dominio.

Los atributos pueden ser simples o compuestos. Un atributo simple es un atributo que tiene un solo componente, que no se puede dividir en partes más pequeñas que tengan un significado propio. Un atributo compuesto es un atributo con varios componentes, cada uno con un significado por sí mismo. Un grupo de atributos se representa mediante un atributo compuesto cuando tienen afinidad en cuanto a su significado, o en cuanto a su uso. Un atributo compuesto se representa gráficamente mediante un óvalo.

Los atributos también pueden clasificarse en monovalentes o polivalentes. Un atributo monovalente es aquel que tiene un solo valor para cada ocurrencia de la entidad o relación a la que pertenece. Un atributo polivalente es aquel que tiene varios valores para cada ocurrencia de la entidad o relación a la que pertenece. A estos atributos también se les denomina multivaluados, y pueden tener un número máximo y un número mínimo de valores. La cardinalidad de un atributo indica el número mínimo y el número máximo de valores que puede tomar para cada ocurrencia de la entidad o relación a la que pertenece. El valor por omisión es (1,1).

Por último, los atributos pueden ser derivados. Un atributo derivado es aquel que representa un valor que se puede obtener a partir del valor de uno o varios atributos, que no necesariamente deben pertenecer a la misma entidad o relación.

Identificador.

Un identificador de una entidad es un atributo o conjunto de atributos que determina de modo único cada ocurrencia de esa entidad. Un identificador de una entidad debe cumplir dos condiciones:

1. No pueden existir dos ocurrencias de la entidad con el mismo valor del identificador.
2. Si se omite cualquier atributo del identificador, la condición anterior deja de cumplirse.

Toda entidad tiene al menos un identificador y puede tener varios identificadores alternativos. Las relaciones no tienen identificadores.

Jerarquía de generalización.

Una entidad *E* es una generalización de un grupo de entidades E^1, E^2, ... E^n, si cada ocurrencia de cada una de esas entidades es también una ocurrencia de *E*.

Todas las propiedades de la entidad genérica *E* son heredadas por las subentidades.

Cada jerarquía es total o parcial, y exclusiva o superpuesta. Una jerarquía es total si cada ocurrencia de la entidad genérica corresponde al menos con una ocurrencia de alguna subentidad. Es parcial si existe alguna ocurrencia de la entidad genérica que no corresponde con ninguna ocurrencia de ninguna subentidad. Una jerarquía es exclusiva si cada ocurrencia de la entidad genérica corresponde, como mucho, con una ocurrencia de una sola de las subentidades. Es superpuesta si existe alguna ocurrencia de la entidad genérica que corresponde a ocurrencias de dos o más subentidades diferentes.

Un subconjunto es un caso particular de generalización con una sola entidad como subentidad. Un subconjunto siempre es una jerarquía parcial y exclusiva.

En la figura 4.2.3.2, se muestra el diagrama entidad-relación del SIMED.

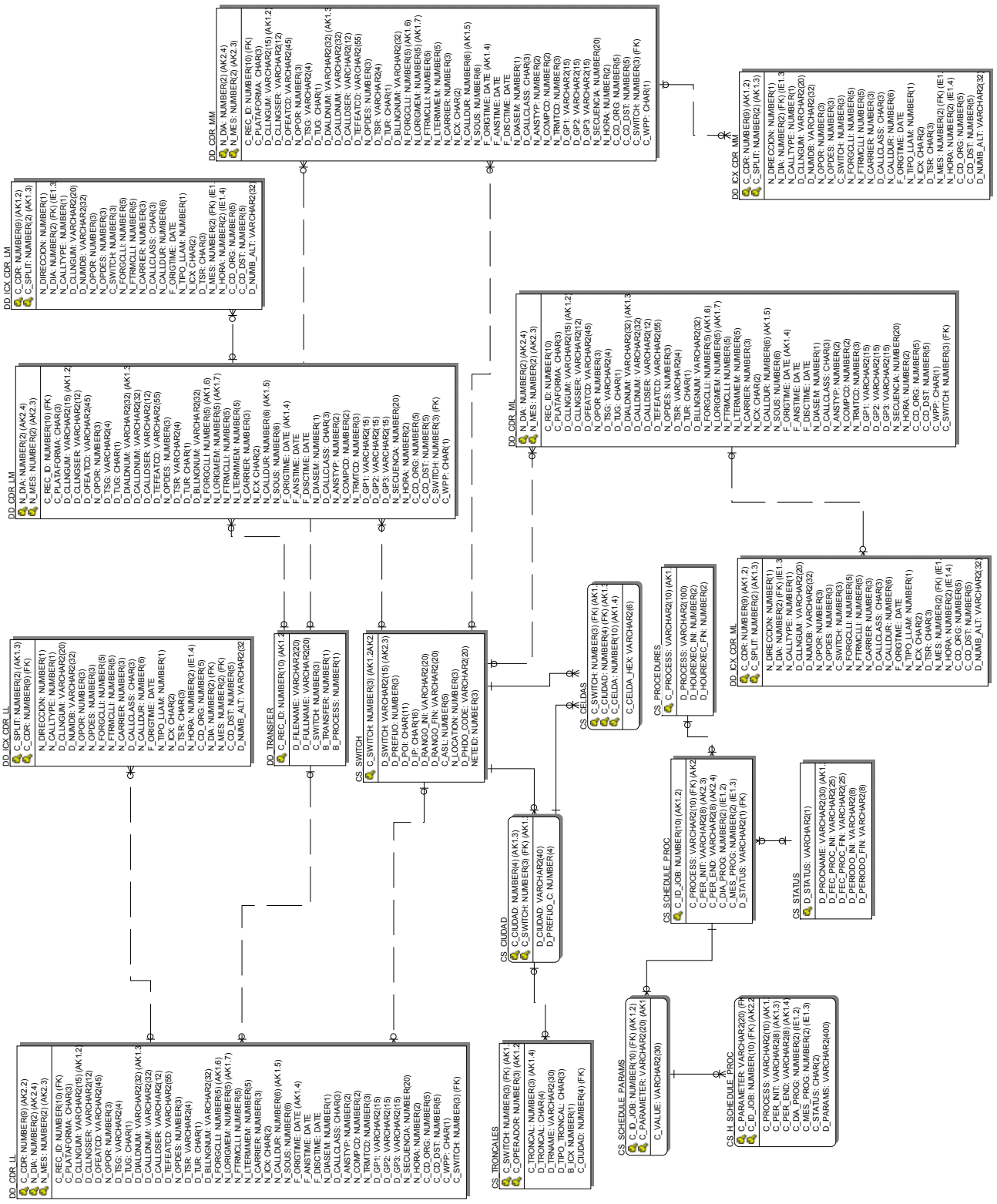

Figura 4.2.3.2 Diagrama entidad-relación

A continuación, se hace una breve descripción de las tablas contenidas en el diagrama entidad relación.

TABLA	DESCRIPCION
CS_ASIT_PHDO	Catálogo que contiene las claves necesarias para el envío de trafico de intercambio de capacidad
CS_CALLCLASS	Catalogo que almacena los diferentes tipos de llamadas a clasificar
CS_CARRIER	Catalogo que contiene todos los carriers de entrega local como de larga distancia con sus respectivas claves
CS_CARRIER_TARIFAS	Catálogo que contiene los cobros de cada carrier
CS_CELDAS	Catálogo que contiene todas las celdas de la compañía, su clave hexadecimal y su ubicación geográfica
CS_CIUDAD	Catálogo que contiene todas las ciudades donde la compañía tiene presencia, identificados por un ID único en toda la compañía
CS_CIUDAD_DAPLACE	Catálogo con las diferentes claves de región para todas las ciudades donde la compañía tiene presencia
CS_CORPORATIVOS	Catálogo que contiene a todos los clientes corporativos o empresas de la compañía, con su respectiva serie manejada así como su cabeza de serie necesaria para la facturación
CS_CORPORATIVOS_VRT	Catálogo que contiene otra modalidad de clientes corporativos de la compañía
CS_EXTRACTOS	Catálogo que almacena los diferentes extractos que se deben generar para otras aplicaciones
CS_H_SCHEDULE_PROC	Tabla para programar reportes y/o entregas periódicas
CS_INT_CARRIER	Catálogo que contiene todos los carriers internacionales con los que tiene convenio la compañía

CS_IRM_ASIT	Catálogo que contiene toda la numeración de IRM para el envío de intercambio de capacidad a facturación
CS_IRM_IU	Catálogo que contiene toda la numeración de IRM para intercambio de capacidad
CS_IRM_MIN	El catálogo de todos los IRM's y su respectivo MIN de todos los clientes de la compañía
CS_IRM_RANGE	El catálogo que contiene todos los rangos de IRM's manejados por la compañía
CS_IRM_UN	El catálogo que contiene todos los IRM's para una de las marcas de la compañía
CS_MIN_POST	Catálogo que contiene todos los mins de todos los clientes de pospago de la compañía
CS_PREFIX_REG	Catálogo con los prefijos para cada región
CS_PROCEDURES	Catálogo de los procedimientos almacenados existentes en la aplicación
CS_REPORTE_ROAMING	Tabla donde se almacenan los reportes de roaming entregados mensualmente
CS_REPORTE_TELFIJ	Tabla que contiene el reporte de telefonía fija entregado mensualmente
CS_REPORTE_TELFIJ_DETALLE	Tabla que contiene el reporte de telefonía fija a detalle
CS_REPORTE_TELFIJ_RESUMEN	Tabla que contiene el reporte de telefonía fija en resumen
CS_RETRO_PHDO	Tabla donde se almacenan todos los CDR's que van a ser enviados para su cobro a la plataforma de tarificación
CS_ROAMING	Tabla que almacena todas las tarifas de roaming para la compañía
CS_ROAMING_TMP	Tabla temporal para el proceso de roaming mensual

CS_RPM_NACIONAL	Tabla que contiene toda la numeración nacional entregada por COFETEL
CS_SCHEDULE_PARAMS	Catálogo con los parámetros necesarios para cada proceso de ejecución programado
CS_SCHEDULE_PROC	Catálogo de procesos programados
CS_SONUS	Catálogo que contiene información respecto al SoftSwitch SONUS
CS_SPECIAL_NUMBERS	Catálogo que contiene números especiales de los clientes de la compañía
CS_STATUS	Catálogo que contiene todos los posibles estatus manejados durante el proceso
CS_SWITCH	Catálogo que contiene todos los MSC de la compañía
CS_TELEFONIA_FIJA	Catálogo que contiene todos los clientes de telefonía fija de la compañía
CS_TRONCALES	Catálogo que contiene todas las troncales que utiliza la compañía para la entrega de trafico
DD_CDR_ASIT_LL	Tabla que contiene todo el tráfico de intercambio de capacidad de la compañía, que sea de calltype 4 LL o Land to Land
DD_CDR_ASIT_ML	Tabla que contiene todo el tráfico de intercambio de capacidad de la compañía, que sea de calltype 1 ML o Mobile to Land
DD_CDR_ASIT_MM	Tabla que contiene todo el tráfico de intercambio de capacidad de la compañía, que sea de calltype 0 MM o Mobile to Mobile
DD_CDR_LL	Tabla que contiene todos los CDR's recolectados de todas las MSC de la compañía de calltype 4, LL o Land to Land
DD_CDR_LM	Tabla que contiene todos los CDR's recolectados de

	todas las MSC de la compañía de calltype 3, LM o Land to Mobile
DD_CDR_ML	Tabla que contiene todos los CDR's recolectados de todas las MSC de la compañía de calltype 1, ML o Mobile to Land
DD_CDR_MM	Tabla que contiene todos los CDR's recolectados de todas las MSC de la compañía de calltype 0, MM o Mobile to Mobile
DD_ICX_CDR_LL	Tabla que contiene todos los CDR's recolectados de las MSC de la compañía de calltype 4, LL o Land to Land y que corresponden a interconexión
DD_ICX_CDR_LM	Tabla que contiene todos los CDR's recolectados de las MSC de la compañía de calltype 3, LM o Land to Mobile y que corresponden a interconexión
DD_ICX_CDR_ML	Tabla que contiene todos los CDR's recolectados de las MSC de la compañía de calltype 1, ML o Mobile to Land y que corresponden a interconexión
DD_ICX_CDR_MM	Tabla que contiene todos los CDR's recolectados de las MSC de la compañía de calltype 0, MM o Mobile to Mobile y que corresponden a interconexión
DD_TRANSFER	Tabla que almacena todos los archivos de CDR's de todas las centrales de la compañía
DD_UDR	Tabla que almacena todos los CDR's de datos
DD_UDR_TRANSFER	Tabla que almacena todos los archivos de CDR's de datos generados por la central

Como se puede observar, la mayoría de las tablas son catálogos necesarios para la clasificación de CDRs (CS_XX) en las tablas operativas (DD_XX). Esta validación y/o referencia se realiza con el módulo de catálogos de la aplicación, para su posterior inserción en la base de datos.

Estas tablas operativas contienen un índice UNIQUE definido sobre los campos D_CLLNGUM, D_DIALDNUM, F_ORIGTIME, N_CALLDUR, N_FORGCLLI, y N_LORIGMEM.. La combinación de estos campos asegura que no podrá insertarse un CDR duplicado. Adicionalmente la llave primaria de estas tablas será una secuencia de Oracle.

- D_CLLNGUM = Número que origina la llamada
- D_DIALDNUM = Número que marca el usuario
- F_ORIGTIME = Fecha y hora de origen de la llamada
- N_CALLDUR = Duración de la llamada expresada en segundos
- N_LORIGMEM = Campo de banderas de Features de la MSC utilizados en la llamada.

Dado que los campos que se insertan en las tablas operativas son complementados y determinados en el módulo de parseo, se decidió que esta validación fuera la única a realizarse y omitir una validación por integridad referencial dentro de la base de datos, cuidando solo la integridad de entidad.

4.2.4 Diagrama de clases.

Diagrama de clases para el módulo de parseo.

La clase principal es la clase ParseThread que a su vez utiliza las clases phone y Trunk para clasificar cada uno de los CDR's, estas clases son las que implementan las interfaces de intercambio mediante RMI entre el módulo de catálogos y el de parseo.
A su vez la clase status control le permite a la clase ParseThread enviar a base de datos el CDR final para su inserción, mediante una queue (cola) de salida, de igual manera la clase CmdServer permite interactuar con las queue's de la aplicación

tanto del módulo de parseo como de la de base de datos, e incluso compartir la configuración con la que se encuentra funcionando, figura 4.2.4.1.

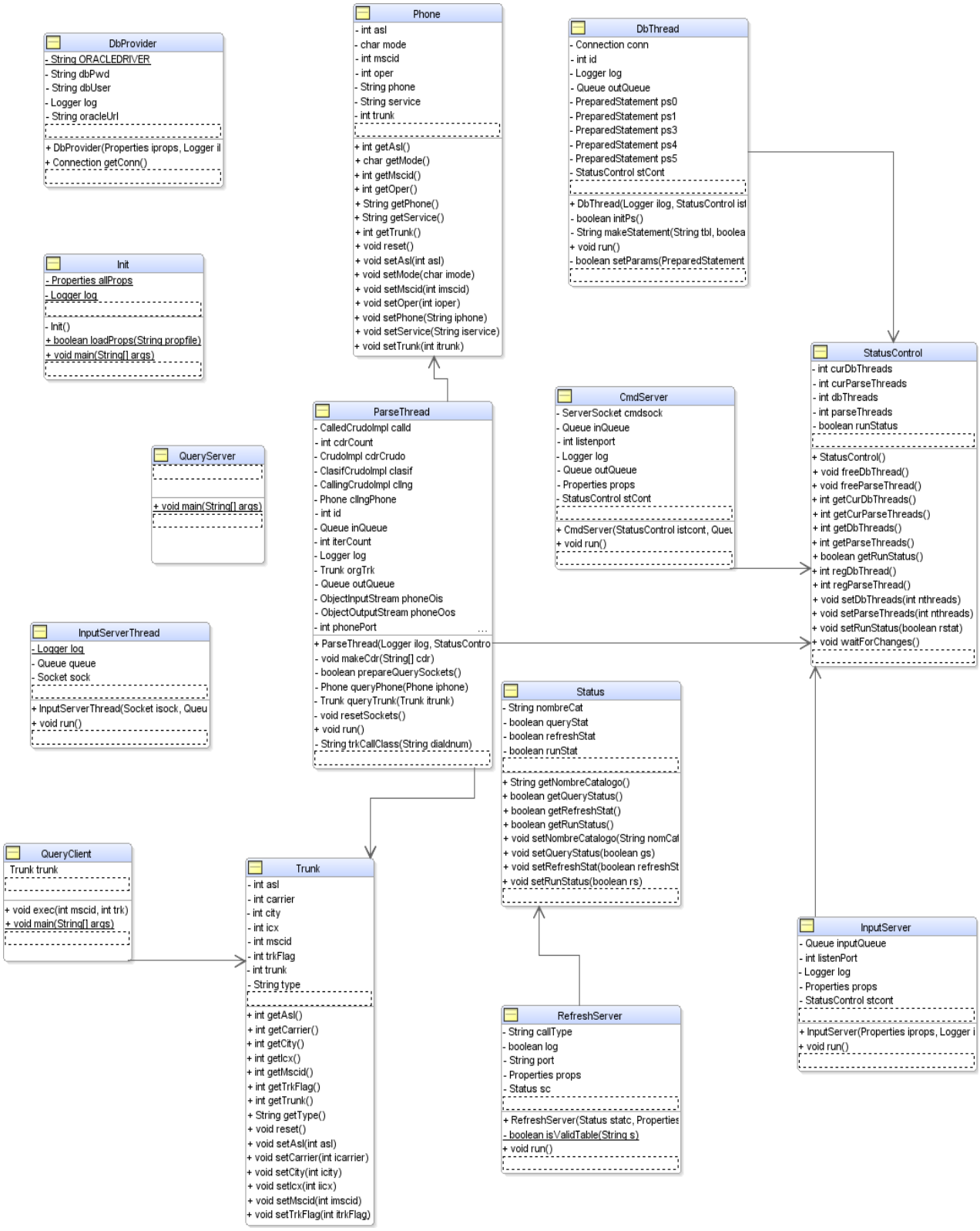

Figura 4.2.4.1 Diagrama de clases para el módulo de parseo.

Diagrama de clases para el módulo de catálogos.

La clase principal InitCat, es la encargada de establecer la conexión a la base de datos e implementa clases genéricas como catálogos y DBProvider, publica las interfaces de los objetos phone y trunk mediante RMI, figura 4.2.4.2.

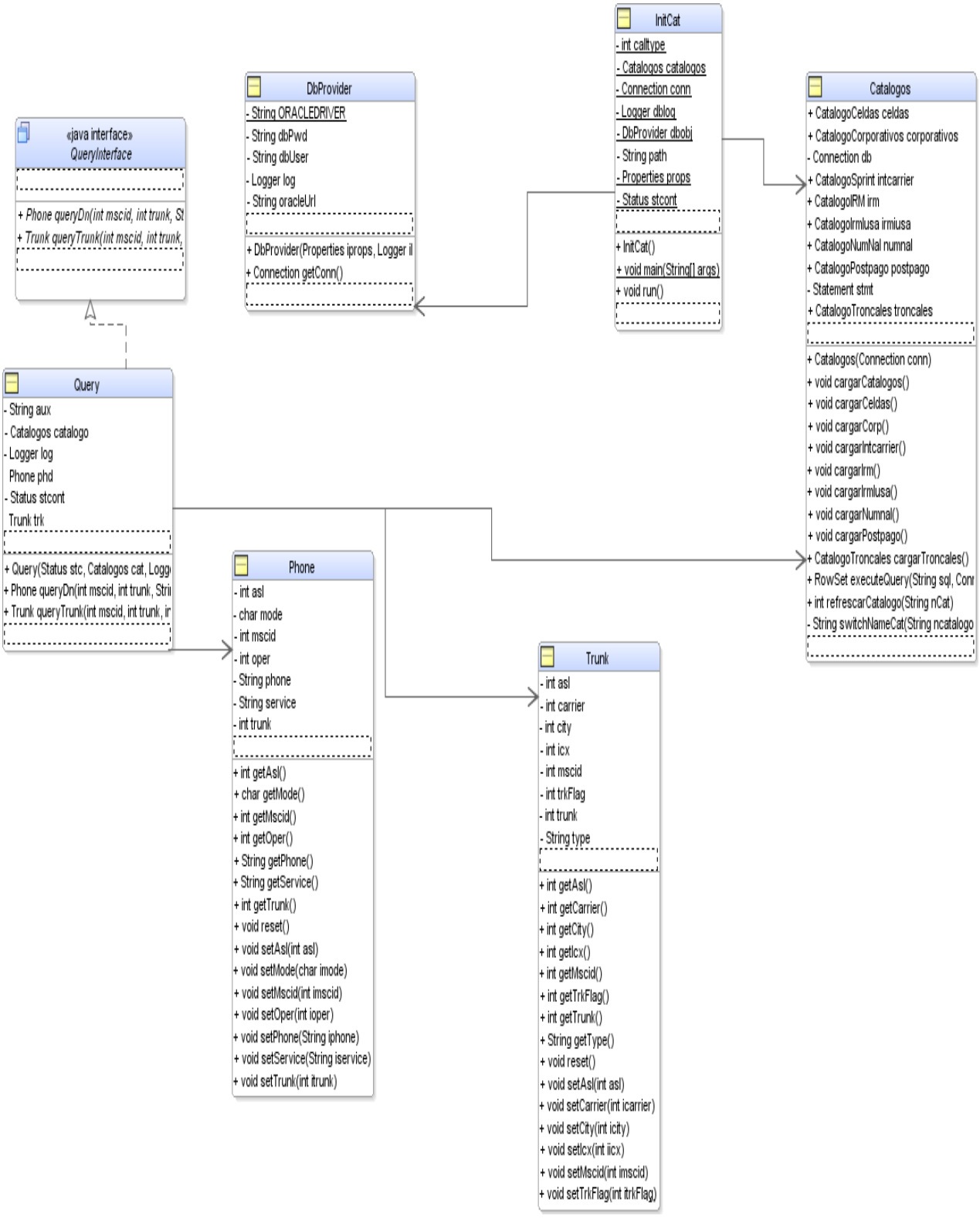

Figura 4.2.4.2 Diagrama de clases para el módulo de catálogos.

4.2.5 Diccionario de datos.

Un diccionario de datos es un conjunto de metadatos que contiene las características lógicas y puntuales de los datos que se van a utilizar en el sistema que se programa, incluyendo nombre, descripción, alias, contenido y organización.

Identifica los procesos donde se emplean los datos y los sitios donde se necesita el acceso inmediato a la información, se desarrolla durante el análisis de flujo de datos y auxilia a los analistas que participan en la determinación de los requerimientos del sistema, su contenido también se emplea durante el diseño.

En un diccionario de datos se encuentra la lista de todos los elementos que forman parte del flujo de datos de todo el sistema. Los elementos más importantes son flujos de datos, almacenes de datos y procesos. El diccionario de datos guarda los detalles y descripción de todos estos elementos.

TABLA: CS_CORPORATIVOS_VRT
Catálogo que contiene otra modalidad de clientes corporativos de la compañía.

Columna	Tipo de dato Longitud	Acepta Nulos	PK o FK	Tabla de Referencia	Breve Descripción
C_COMP	VARCHAR2(10)	NOT NULL	PK		
D_INT_TRUNK	VARCHAR2(20)	NOT NULL	PK		
D_HEADER	VARCHAR2(10)	NOT NULL			
C_SWITCH	NUMBER(3)	NOT NULL			

Tabla 4.2.5.1. CS_CORPORATIVOS_VRT

TABLA: CS_CORPORATIVOS
Catálogo que contiene a todos los clientes corporativos o empresas de la compañía, con su respectiva seria manejada así como su cabeza de serie necesaria para la facturación.

Columna	Tipo de dato Longitud	Acepta Nulos	PK o FK	Tabla de Referencia	Breve Descripción
C_COMP	VARCHAR2(10)	NOT NULL	PK		

D_INT_TRUNK	VARCHAR2(50)	NOT NULL			
D_HEADER	VARCHAR2(10)	NOT NULL			
C_SWITCH	NUMBER(3)	NOT NULL			

Tabla 4.2.5.2. CS_CORPORATIVOS

TABLA: CS_INT_CARRIER
Catálogo que contiene todos los carriers internacionales con los que tiene convenio la compañía.

Columna	Tipo de dato Longitud	Acepta Nulos	PK o FK	Tabla de Referencia	Breve Descripción
D_COMPANY	VARCHAR2(25)	NOT NULL	PK		
D_REGION	VARCHAR2(50)	NOT NULL	PK		
N_INICIO	VARCHAR2(10)	NOT NULL			
N_FIN	VARCHAR2(10)	NOT NULL			

Tabla 4.2.5.3. CS_INT_CARRIER

TABLA: DD_CDR_LL
Tabla que contiene todos los CDRs recolectados de todas las MSC de la compañía de calltype 4, LL o Land to Land.

Columna	Tipo de dato Longitud	Acepta Nulos	PK o FK	Tabla de Referencia	Breve Descripción
C_CDR	NUMBER(9)	NOT NULL	PK		
N_DIA	NUMBER(2)	NOT NULL	PK		
N_MES	NUMBER(2)	NOT NULL	PK		
C_REC_ID	NUMBER(10)	NOT NULL			
C_PLATAFORMA	CHAR(3)	NOT NULL			
D_CLLNGUM	VARCHAR2(15)	NOT NULL			
D_CLLGSER	VARCHAR2(12)	NOT NULL			

Tabla 4.2.5.4. DD_CDR_LL

TABLA: CS_CIUDAD
Catálogo que contiene todas las ciudades donde la compañía tiene presencia, identificados por un ID único en toda la compañía.

Columna	Tipo de dato Longitud	Acepta Nulos	PK o FK	Tabla de Referencia	Breve Descripción
CS_CIUDAD	NUMBER(4)	NOT NULL	PK		
CS_SWITCH	NUMBER(3)	NOT NULL	PK/FK		
D_CIUDAD	VARCHAR2(40)	NOT NULL			
D_PREFIJO_C	NUMBER(4)	NOT NULL			

Tabla 4.2.5.5. CS_CIUDA

TABLA: CS_ROAMING
Tabla que almacena todas las tarifas de roaming para la compañía.

Columna	Tipo de dato Longitud	Acepta Nulos	PK o FK	Tabla de Referencia	Breve Descripción
C_CDR	NUMBER(9)	NOT NULL	PK		
N_MES	NUMBER(2)	NOT NULL			
N_DIA	NUMBER(2)	NOT NULL			
D_CLIENTE	CHAR(10)	NOT NULL			
D_STATUS	CHAR(1)	NOT NULL			
D_STATUS_PROC	CHAR(1)	NOT NULL			
C_ROAMING	NUMBER(1)	NOT NULL			

Tabla 4.2.5.6. CS_ROAMING

TABLA: CS_RETRO_PHDO
Tabla donde se almacenan todos los CDRs que van a ser enviados para su cobro a la plataforma de tarificación.

Columna	Tipo de dato Longitud	Acepta Nulos	PK o FK	Tabla de Referencia	Breve Descripción
C_CDR	NUMBER(9)	NOT NULL	PK		

N_MES	NUMBER(2)	NOT NULL			
N_DIA	NUMBER(2)	NOT NULL			
D_CHAIN	VARCHAR2(160)	NOT NULL			
D_STATUS	CHAR(1)	NOT NULL			
D_STATUS_PROC	CHAR(1)	NOT NULL			
C_RESPUESTA	NUMBER(6)	NOT NULL			
C_SWITCH	NUMBER(3)	NOT NULL			

Tabla 4.2.5.7. CS_RETRO_PHDO

TABLA: CS_SCHEDULE_PROC
Catálogo de procesos programados.

Columna	Tipo de dato Longitud	Acepta Nulos	PK o FK	Tabla de Referencia	Breve Descripción
C_ID_JOB	NUMBER(10)	NOT NULL	PK		
C_PROCESS	VARCHAR2(10)	NOT NULL			
C_PER_INIT	VARCHAR2(8)	NOT NULL			
C_PER_END	VARCHAR(8)	NOT NULL			
C_DIA_PROG	NUMBER(2)	NOT NULL			
C_MES_PROG	NUMBER(2)	NOT NULL			

Tabla 4.2.5.8. CS_SCHEDULE_PROC

TABLA: DD_ICX_CDR_LL
Tabla que contiene todos los CDRs recolectados de las MSC de la compañía de calltype 4, LL o Land to Land y que corresponden a interconexión.

Columna	Tipo de dato Longitud	Acepta Nulos	PK o FK	Tabla de Referencia	Breve Descripción
C_CDR	NUMBER(9)	NOT NULL	PK/FK		
C_SPLIT	NUMBER(2)	NOT NULL	PK		

N_DIRECCION	NUMBER(1)	NOT NULL			
N_CALLTYPE	NUMBER(1)	NOT NULL			
D_CLLNGUM	VARCHAR2(20)	NOT NULL			
D_NUMDB	VARCHAR2(32)	NOT NULL			
N_OPOR	NUMBER(3)	NOT NULL			
N_OPDES	NUMBER(3)	NOT NULL			

Tabla 4.2.5.9. DD_ICX_CDR_L

TABLA: CS_SWITCH
Catálogo que contiene todos los MSC de la compañía.

Columna	Tipo de dato Longitud	Acepta Nulos	PK o FK	Tabla de Referencia	Breve Descripción
C_SWITCH	NUMBER(3)	NOT NULL	PK		
D_SWITCH	VARCHAR2(15)	NOT NULL			
D_PREFIJO	NUMBER(3)	NOT NULL			
D_POI	CHAR(11)	NOT NULL			
D_IP	CHAR(16)	NOT NULL			
D_RANGO_INI	VARCHAR2(20)	NOT NULL			
D_RANGO_FIN	VARCHAR2(20)	NOT NULL			
C_ASL	NUMBRE(5)	NOT NULL			

Tabla 4.2.5.10. CS_SWITCH

TABLA: CS_CELDAS
Catálogo que contiene todas las celdas de la compañía, su clave hexadecimal y su ubicación geográfica.

Columna	Tipo de dato Longitud	Acepta Nulos	PK o FK	Tabla de Referencia	Breve Descripción
C_SWITCH	NUMBER(3)	NOT NULL	PK/FK		
C_CIUDAD	NUMBER(4)	NOT NULL	PK/FK		
C_CELDA	NUMBER(10)	NOT NULL	PK		
C_CELDA_HEX	VARCHAR2(5)	NOT NULL			

Tabla 4.2.5.11. CS_CELDAS

TABLA: DD_TRANSFER
Tabla que almacena todos los archivos de CDRs de todas las centrales de la compañía.

Columna	Tipo de dato Longitud	Acepta Nulos	PK o FK	Tabla de Referencia	Breve Descripción
C_REC_ID	NUMBER(10)	NOT NULL	PK		
D_FILENAME	VARCHAR2(20)	NOT NULL			
D_FULLNAME	VARCHAR2(20)	NOT NULL			
C_SWITCH	NUMBER(3)	NOT NULL			
B_TRANSFER	NUMBER(1)	NOT NULL			
B_PROCESS	NUMBER(1)	NOT NULL			

Tabla 4.2.5.12. DD_TRANSFER

TABLA: CS_TRONCALES
Catálogo que contiene todas las troncales que utiliza la compañía para la entrega de tráfico.

Columna	Tipo de dato Longitud	Acepta Nulos	PK o FK	Tabla de Referencia	Breve Descripción
C_SWITCH	NUMBER(3)	NOT NULL	PK		
C_OPERADOR	NUMBER(3)	NOT NULL	PK		
C_TRONCAL	NUMBER(3)	NOT NULL			
D_TRONCAL	CHAR(4)	NOT NULL			

D_TRNAME	VARCHAR2(30)	NOT NULL			
D_TIPO_TRONCAL	CHAR(3)	NOT NULL			
B_ICX	NUMBER(3)	NOT NULL			
C_CIUDAD	NUMBER(3)	NOT NULL			

Tabla 4.2.5.13. CS_TRONCALES

TABLA: IRM_MIN.
El catálogo de todos los IRM´s y su respectivo MIN de todos los clientes de la compañía.

Columna	Tipo de dato Longitud	Acepta Nulos	PK o FK	Tabla de Referencia	Breve Descripción
MIN	CHAR(10)	NOT NULL	PK		
IRM	CHAR(10)	NOT NULL			

Tabla 4.2.5.14. IRM_MIN.

TABLA: CS_CARRIER
Catálogo que contiene los cobros de cada carrier.

Columna	Tipo de dato Longitud	Acepta Nulos	PK o FK	Tabla de Referencia	Breve Descripción
CARRIER	VARCHAR2(5)	NOT NULL	PK		
TARIFA	NUMBER(4,2)	NOT NULL			

Tabla 4.2.5.15. CS_CARRIER

TABLA: CS_PROCEDURES
Catálogo de los procedimientos almacenados existentes en la aplicación.

Columna	Tipo de dato Longitud	Acepta Nulos	PK o FK	Tabla de Referencia	Breve Descripción
C_PROCESS	VARCHAR2(10)	NOT NULL	PK		
D_PROCESS	VARCHAR2(100)	NOT NULL			
D_HOUREXEC_INI	NUMBER(2)	NOT NULL			

D_HOUREEXEC_FIN	NUMBER(2)	NOT NULL			

Tabla 4.2.5.16. CS_PROCEDURES

TABLA: CS_RPM_NACIONAL
Tabla que contiene toda la numeración nacional entregada por COFETEL.

Columna	Tipo de dato Longitud	Acepta Nulos	PK o FK	Tabla de Referencia	Breve Descripción
MIN	CHAR(25)	NOT NULL	PK		
CITY	CHAR(25)	NOT NULL			

Tabla 4.2.5.17. CS_RPM_NACIONAL

TABLA: CS_PREFIX_REG
Catálogo con los prefijos de cada región.

Columna	Tipo de dato Longitud	Acepta Nulos	PK o FK	Tabla de Referencia	Breve Descripción
C_REGION	VARCHAR2(50)	NOT NULL	PK		
D_PREFIX	NUMBER(5)	NOT NULL			
C_TARIFA	NUMBER(4,2)	NOT NULL			
D_CIUDAD	VARCHAR2(55)	NOT NULL			

Tabla 4.2.5.18. CS_PREFIX_REG

TABLA: CS_MIN_POST
Catálogo que contiene todos los MIN´s de todos los clientes de postpago de la compañía.

Columna	Tipo de dato Longitud	Acepta Nulos	PK o FK	Tabla de Referencia	Breve Descripción
N_NUM_POST	CHAR(10)	NOT NULL	PK		
N_CUENTA	NUMBER(10)	NOT NULL			

Tabla 4.2.5.19. CS_MIN_POST

TABLA: CS_REPORTE_TELFIJ
Tabla que contiene el reporte de telefonía fija entregado mensualmente.

Columna	Tipo de dato Longitud	Acepta Nulos	PK o FK	Tabla de Referencia	Breve Descripción
N_MES	NUMBER(2)	NOT NULL	PK		
N_DIA_INI	NUMBER(2)	NOT NULL	PK		
N_DIA_FIN	NUMBER(2)	NOT NULL	PK		
C_RESPUESTA	NUMBER(10)	NOT NULL	PK		
D_CHAIN	VARCHAR2(1000)	NOT NULL			

Tabla 4.2.5.20. CS_REPORTE_TELFIJ

TABLA: CS_H_SCHEDULE_PROC
Tabla para programar reportes y/o entregas periódicas.

Columna	Tipo de dato Longitud	Acepta Nulos	PK o FK	Tabla de Referencia	Breve Descripción
C_ID_JOB	NUMBER(10)	NOT NULL			
C_PROCESS	VARCHAR2(10)	NOT NULL			
C_PER_INIT	VARCHAR2(8)	NOT NULL			
C_PER_END	VARCHAR2(8)	NOT NULL			
C_DIA_PROG	NUMBER(2)	NOT NULL			
C_MES_PROG	NUMBER(2)	NOT NULL			
C_STATUS	CHAR(2)	NOT NULL			
D_PARAMS	VARCHAR2(400)	NOT NULL			

Tabla 4.2.5.21. CS_H_SCHEDULE_PROC

TABLA: CS_SCHEDULE_PARAMS
Catálogo con los parámetros necesarios para cada proceso de ejecución programado.

Columna	Tipo de dato Longitud	Acepta Nulos	PK o FK	Tabla de Referencia	Breve Descripción
C_ID_JOB	NUMBER(10)	NOT NULL	PK		

C_PARAMETER	VARCHAR2(20)	NOT NULL	PK		
C_VALUE	VARCHAR2(30)	NOT NULL			

Tabla 4.2.5.22. CS_SCHEDULE_PARAMS

TABLA: CS_STATUS
Catálogo que contiene todos los posibles estatus manejados durante el proceso.

Columna	Tipo de dato Longitud	Acepta Nulos	PK o FK	Tabla de Referencia	Breve Descripción
D_PROCNAME	VARCHAR2(30)	NOT NULL	PK		
D_STATUS	VARCHAR2(25)	NOT NULL			
D_FEC_PROC_INI	VARCHAR2(25)	NOT NULL			
D_FEC_PROC_FIN	VARCHAR2(25)	NOT NULL			

Tabla 4.2.5.23. CS_STATUS

TABLA: DD_CDR_LM
Tabla que contiene todos los CDRs recolectados de todas las MSC de la compañía de calltype 3, LM o Land to Mobile.

Columna	Tipo de dato Longitud	Acepta Nulos	PK o FK	Tabla de Referencia	Breve Descripción
N_DIA	NUMBER(2)	NOT NULL	PK		
N_MES	NUMBRE(2)	NOT NULL	PK		
C_REC_ID	NUMBER(10)	NOT NULL			
C_PLATAFORMA	CHAR(3)	NOT NULL			
D_CLLNGUM	VARCHAR2(15)	NOT NULL			
D_CLLNGSER	VARCHAR2(12)	NOT NULL			
D_OFEATCD	VARCHAR2(45)	NOT NULL			
N_OPOR	NUMBER(3)	NOT NULL			

Tabla 4.2.5.24. DD_CDR_LM

TABLA: DD_CDR_ML
Tabla que contiene todos los CDRs recolectados de todas las MSC de la compañía de calltype 1, ML o Mobile to Land.

Columna	Tipo de dato Longitud	Acepta Nulos	PK o FK	Tabla de Referencia	Breve Descripción
N_DIA	NUMBER(2)	NOT NULL	PK		
N_MES	NUMBRE(2)	NOT NULL	PK		
C_REC_ID	NUMBER(10)	NOT NULL			
C_PLATAFORMA	CHAR(3)	NOT NULL			
D_CLLNGUM	VARCHAR2(15)	NOT NULL			
D_CLLNGSER	VARCHAR2(12)	NOT NULL			
D_OFEATCD	VARCHAR2(45)	NOT NULL			
N_OPOR	NUMBER(3)	NOT NULL			

Tabla 4.2.5.25. DD_CDR_ML

TABLA: CS_SPECIAL_NUMBERS
Catálogo que contiene números especiales de los clientes de la compañía.

Columna	Tipo de dato Longitud	Acepta Nulos	PK o FK	Tabla de Referencia	Breve Descripción
C_NUMBER	VARCHAR2(10)	NOT NULL	PK		
D_NUMBER	VARCHAR2(500)	NOT NULL			

Tabla 4.2.5.26. CS_SPECIAL_NUMBERS

TABLA: CS_TELEFONIA_FIJA
Catálogo que contiene todos los clientes de Telefonía fija de la compañía.

Columna	Tipo de dato Longitud	Acepta Nulos	PK o FK	Tabla de Referencia	Breve Descripción
C_CDR	NUMBER(9)	NOT NULL	PK		
N_MES	NUMBRE(2)	NOT NULL			

N_DIA	NUMBER(2)	NOT NULL			
D_CLIENTE	CHAR(10)	NOT NULL			
D_STATUS	CHAR(1)	NOT NULL			
D_STATUS_PROC	CHAR(1)	NOT NULL			
C_ROAMING	NUMBER(1)	NOT NULL			
C_RESPUESTA	NUMBER(5)	NOT NULL			

Tabla 4.2.5.27. CS_TELEFONIA_FIJA

TABLA: DD_CDR_MM
Tabla que contiene todos los CDRs recolectados de todas las MSC de la compañía de calltype 0, MM o Mobile to Mobile.

Columna	Tipo de dato Longitud	Acepta Nulos	PK o FK	Tabla de Referencia	Breve Descripción
N_DIA	NUMBER(2)	NOT NULL	PK		
N_MES	NUMBRE(2)	NOT NULL	PK		
C_REC_ID	NUMBER(10)	NOT NULL			
C_PLATAFORMA	CHAR(3)	NOT NULL			
D_CLLNGUM	VARCHAR2(15)	NOT NULL			
D_CLLNGSER	VARCHAR2(12)	NOT NULL			
D_OFEATCD	VARCHAR2(45)	NOT NULL			
N_OPOR	NUMBER(3)	NOT NULL			

Tabla 4.2.5.28. DD_CDR_MM

TABLA: CS-REPORTE_ROAMING
Tabla donde se almacenan los reportes de roaming entregados mensualmente.

Columna	Tipo de dato Longitud	Acepta Nulos	PK o FK	Tabla de Referencia	Breve Descripción
N_MES	NUMBER(2)	NOT NULL	PK		

N_DIA_INI	NUMBRE(2)	NOT NULL	PK		
N_DIA_FIN	NUMBER(2)	NOT NULL	PK		
C_RESPUESTA	NUMBER(10)	NOT NULL	PK		
D_CHAIN	VARCHAR2(1000)	NOT NULL			

Tabla 4.2.5.29. CS-REPORTE_ROAMING

TABLA: CS_CARRIER
Catálogo que contiene todos los carriers de entrega local como de larga distancia con sus respectivas claves.

Columna	Tipo de dato Longitud	Acepta Nulos	PK o FK	Tabla de Referencia	Breve Descripción
C_CARRIER	VARCHAR2(4)	NOT NULL	PK		
D_CARRIER	VARCHAR2(50)	NOT NULL			
C_CUENTA	VARCHAR2(7)	NOT NULL			
D_DN	VARCHAR2(10)	NOT NULL			
D_MIN	VARCHAR2(10)	NOT NULL			
D_RATEPLAN	VARCHAR2(10)	NOT NULL			

Tabla 4.2.5.30. CS_CARRIER

TABLA: DD_ICX_CDR_ML
Tabla que contiene todos los CDRs recolectados de las MSC de la compañía calltype 1, ML o Mobile to Land y que corresponden a interconexión.

Columna	Tipo de dato Longitud	Acepta Nulos	PK o FK	Tabla de Referencia	Breve Descripción
C_CDR	NUMBER(9)	NOT NULL	PK		
C_SPLIT	NUMBRE(2)	NOT NULL	PK		
N_DIRECCION	NUMBER(1)	NOT NULL			
N_DIA	NUMBER(2)	NOT NULL	FK		

N_CALLTYPE	NUMBER(1)	NOT NULL			
D_CLLNGUM	VARCHAR2(20)	NOT NULL			
D_NUMDB	VARCHAR2(32)	NOT NULL			
N_OPOR	NUMBER(3)	NOT NULL			

Tabla 4.2.5.31. DD_ICX_CDR_ML

TABLA: DD_ICX_CDR_MM
Tabla que contiene todos los CDRs recolectados de todas las MSC de la compañía de calltype 0, MM o Mobile to Mobile y que corresponden a interconexión.

Columna	Tipo de dato Longitud	Acepta Nulos	PK o FK	Tabla de Referencia	Breve Descripción
C_CDR	NUMBER(9)	NOT NULL	PK		
C_SPLIT	NUMBRE(2)	NOT NULL	PK		
N_DIRECCION	NUMBER(1)	NOT NULL			
N_DIA	NUMBER(2)	NOT NULL	FK		
N_CALLTYPE	NUMBER(1)	NOT NULL			
D_CLLNGUM	VARCHAR2(20)	NOT NULL			
D_NUMBD	VARCHAR2(32)	NOT NULL			
N_OPOR	NUMBER(3)	NOT NULL			

Tabla 4.2.5.32. DD_ICX_CDR_MM

TABLA: DD_ICX_CDR_LM
Tabla que contiene todos los CDRs recolectados de todas las MSC de la compañía de calltype 3, LM o Land to Mobile y que corresponden a interconexión.

Columna	Tipo de dato Longitud	Acepta Nulos	PK o FK	Tabla de Referencia	Breve Descripción
C_CDR	NUMBER(9)	NOT NULL	PK		
C_SPLIT	NUMBRE(2)	NOT NULL	PK		

N_DIRECCION	NUMBER(1)	NOT NULL			
N_DIA	NUMBER(2)	NOT NULL	FK		
N_CALLTYPE	NUMBER(1)	NOT NULL			
D_CLLNGUM	VARCHAR2(20)	NOT NULL			
D_NUMBD	VARCHAR2(32)	NOT NULL			
N_OPOR	NUMBER(3)	NOT NULL			

Tabla 4.2.5.33. DD_ICX_CDR_LM

4.3 Diseño y construcción del back end

A continuación haremos una breve descripción del proceso que se sigue para el diseño y construcción de una base de datos en Oracle SQL Developer 1.2.1 de agosto de 2007. Describiremos como crear una tabla, una consulta y los triggers necesarios considerando que la aplicación en cuestión está conceptualizada como un daemon.

El objetivo fundamental es proporcionar una interfaz más amigable para la consulta y programación de la base de datos Oracle. La funcionalidad disponible es sólo parte de la disponibilidad a través de comandos en SQL*Plus, pero se corresponde con las tareas más habituales de interacción, programación y depuración de código sobre la base de datos.

Ejecución de sentencias SQL.

Escribir las sentencias SQL en la ventana de edición SQL (SQL Worksheet), figura 4.3.1.

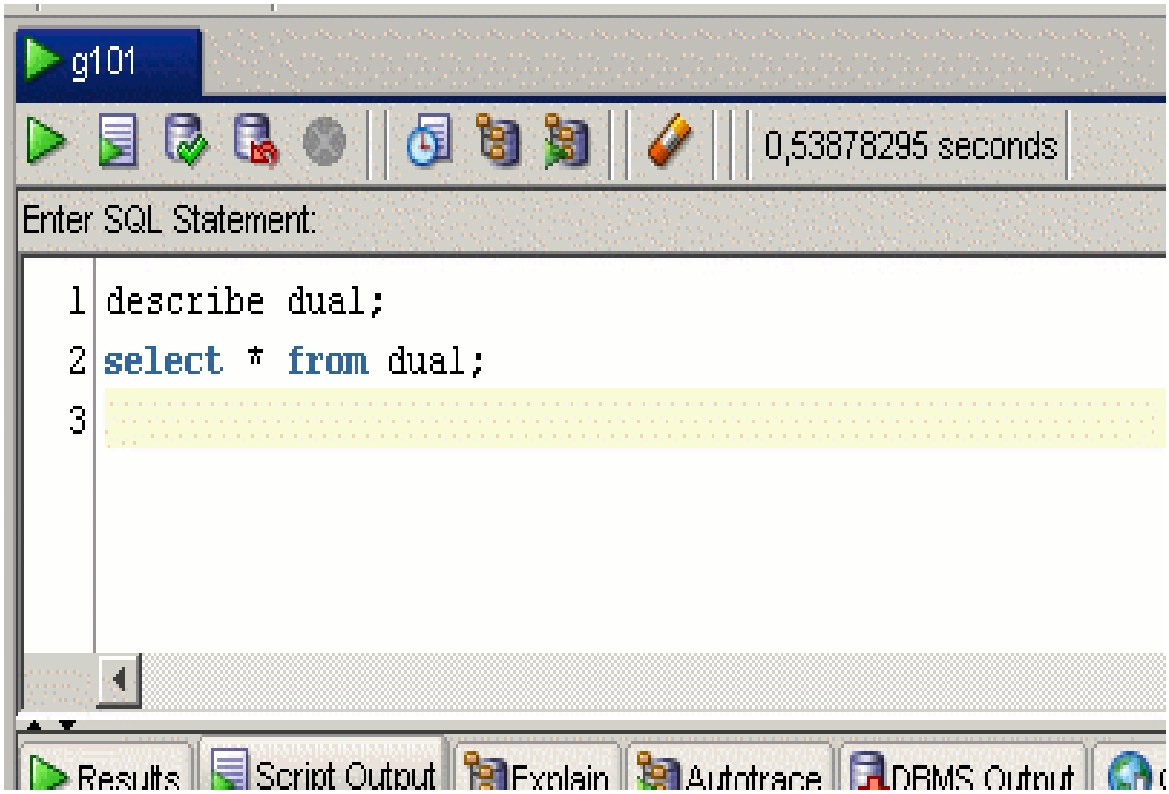

Figura 4.3.1 Ejecución de sentencias.

Para ejecutar sólo una sentencia, se sitúa el cursor sobre la sentencia y se pulsa el icono o la tecla F9.

Para ejecutar todas las sentencias, se pulsa el icono o la tecla F5.

Los resultados de la ejecución de las sentencias SQL se muestran en las pestañas “Results” y “Script Output”, figura 4.3.2.

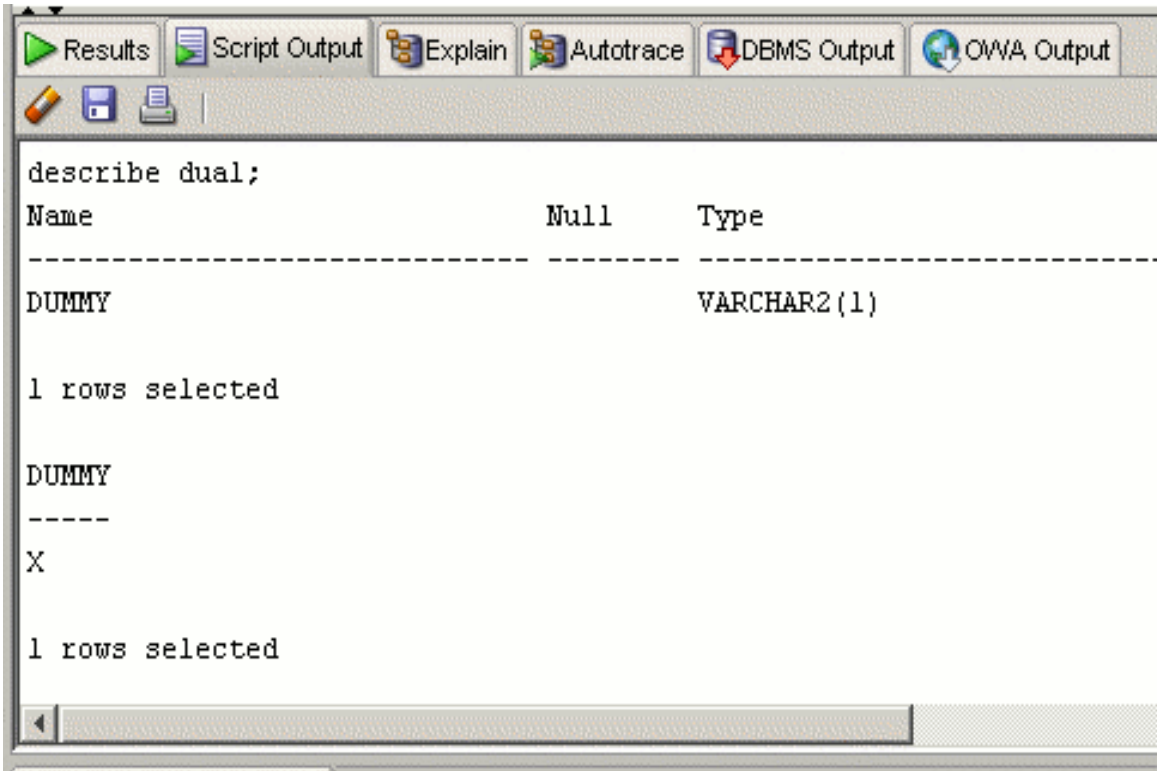

Figura 4.3.2 Pestañas “Results” y “Script Output”.

El icono permite acceder a un histórico de sentencias SQL ejecutadas. Para cargar una sentencia del histórico sobre el editor SQL se hace doble-clik sobre la sentencia.

El icono borra el contenido del editor SQL.

Para ver el número de línea en el editor SQL hay que activar Tools □□Preferentes □□Code Editor □□Line Gutter □□Show Line Numbers.

Para grabar a un fichero .SQL el contenido del editor SQL se utiliza la opción File □□Save o el icono.

Para abrir un fichero .SQL en el editor SQL se utiliza la opción File □□Open o el icono.

Para abrir un nuevo editor SQL se utiliza la opción Tools □□SQL Worksheet o el icono.

Para crear y editar un nuevo fichero SQL se utiliza la opción File □□New □□SQL File

IMPORTANTE: Las sentencias SQL que modifican la base de datos (INSERT INTO, UPDATE, DELETE, ...) no se realizan (ejecutan) en la base de datos hasta que se pulsa el icono .

Si se quiere que las sentencias SQL se realicen automáticamente después de ejecutarlas hay que activar la opción Tools □□Preferences □□Database □□Worksheet Parameters □□Autocommit in SQL Worksheet

Para que los cambios realizados por sentencias SQL de creación de objetos (DDL) se reflejen en el navegador de objetos, es necesario pulsar el icono "Refresh".

Creación de tablas.

Pulsar el botón derecho sobre el icono “Tables” de la conexión, figura 4.3.3.

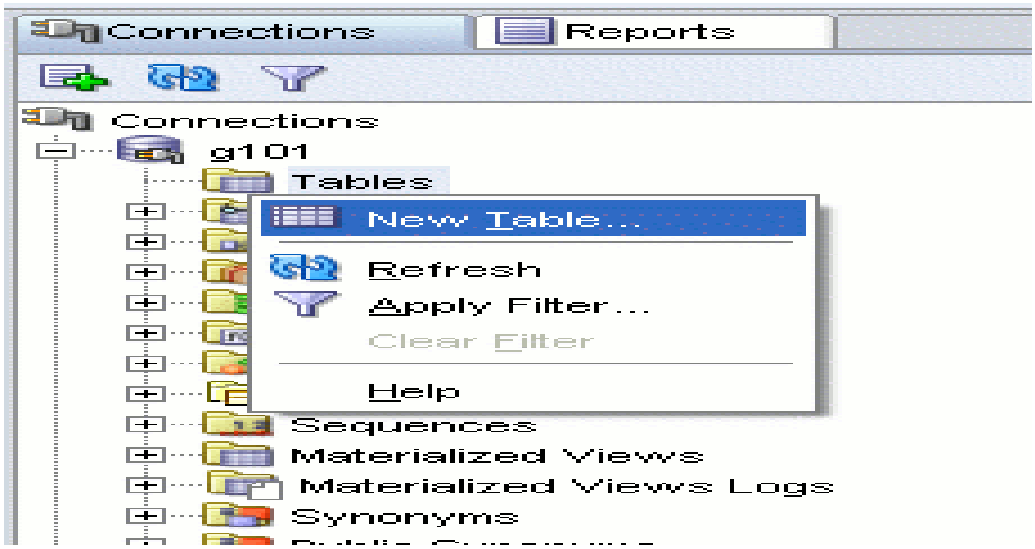

Figura 4.3.3 Pulsar “New Table...”.

Nos despliega un menú, en el cual, se selecciona la opción “New Table”, devolviéndonos la siguiente pantalla, figura 4.3.4.

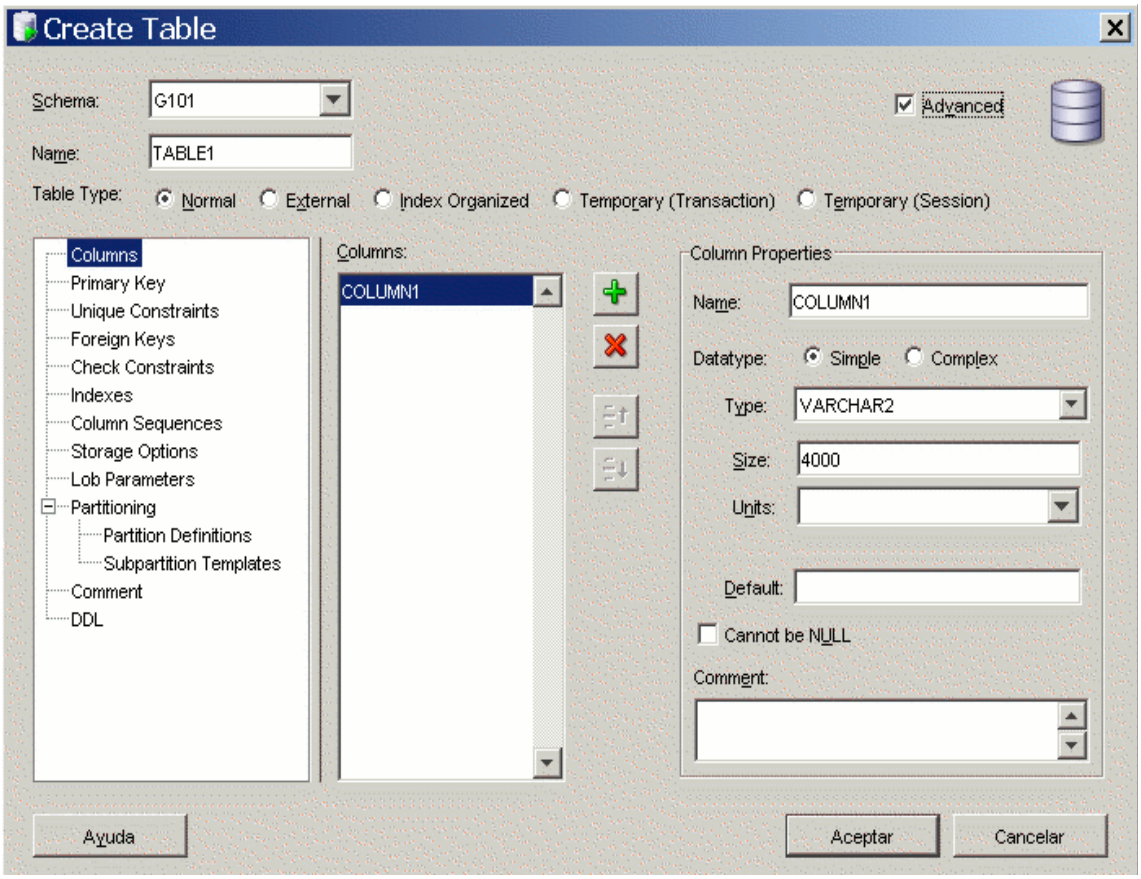

Figura 4.3.4 Creación de tablas.

Modificación de la definición de una tabla.

1. Seleccionar la tabla (doble-click sobre su icono)
2. Elegir la pestaña “Columns”
3. Pulsar sobre el icono “Edit”

También se puede hacer pulsando con el botón derecho sobre el icono de la tabla que se quiere modificar y eligiendo la opción “Edit...”

Inserción de tuplas.

Para insertar tuplas en una tabla, se selecciona la tabla, y se pulsa la pestaña “Data”, figura 4.3.5.

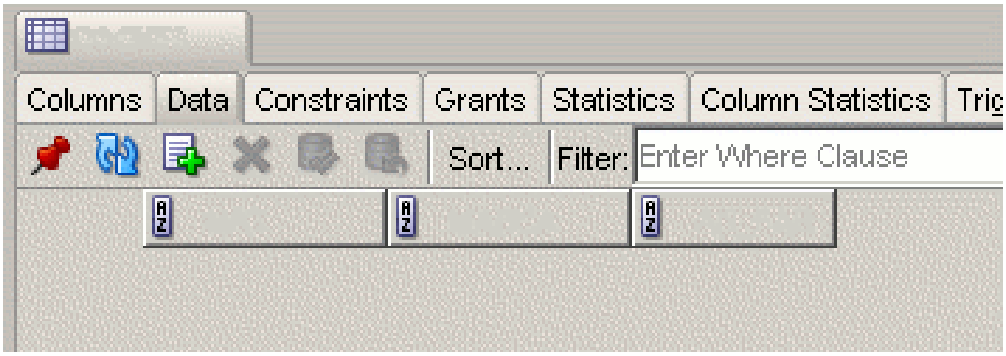

Figura 4.3.5 Inserción de tuplas “data”.

Para introducir una nueva tupla se pulsa sobre el icono, y se escriben los valores de cada atributo, figura 4.3.6.

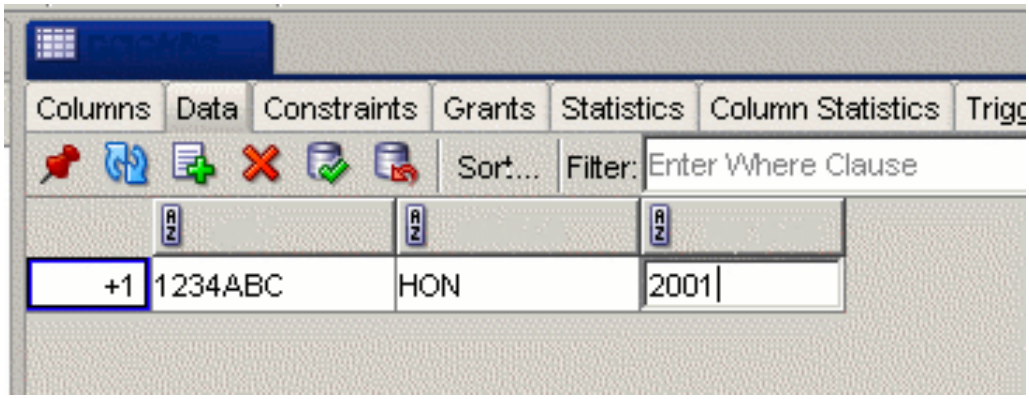

Figura 4.3.6 Inserción de tuplas “Icono”.

Para grabar la(s) tupla(s) en la tabla se pulsa el icono (commit).

El icono permite borrar una tupla.

Para grabar la(s) tupla(s) en la tabla se pulsa el icono (commit).

El icono permite borrar una tupla.

El icono permite fijar la pestaña de la tabla actual de manera que si se selecciona otra tabla en el navegador de objetos se abrirá otra pestaña nueva y no se reutilizará la pestaña fijada.

Generación de código SQL.

Generación del código SQL de un único objeto (tabla, secuencia, procedimiento, disparador), figura 4.3.7.

Pulsar botón derecho sobre el objeto y seleccionar “Export DDL”

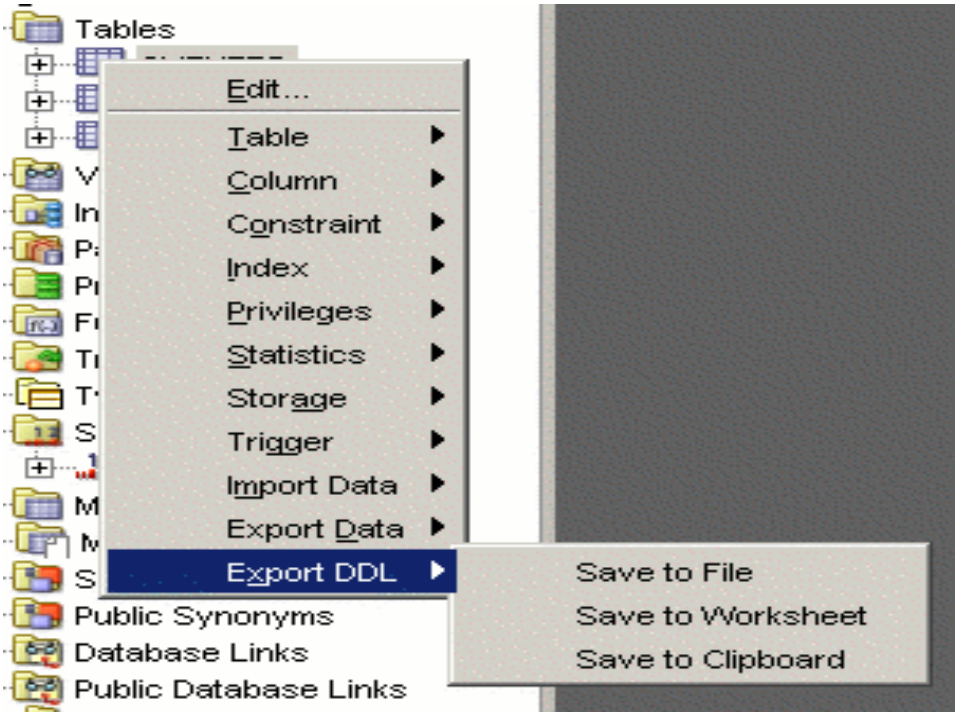

Figura 4.3.7 Generación del Código de Toda una Conexión.

Ir a “Tools” --> “Export DDL (and data)”, figura 4.3.8.

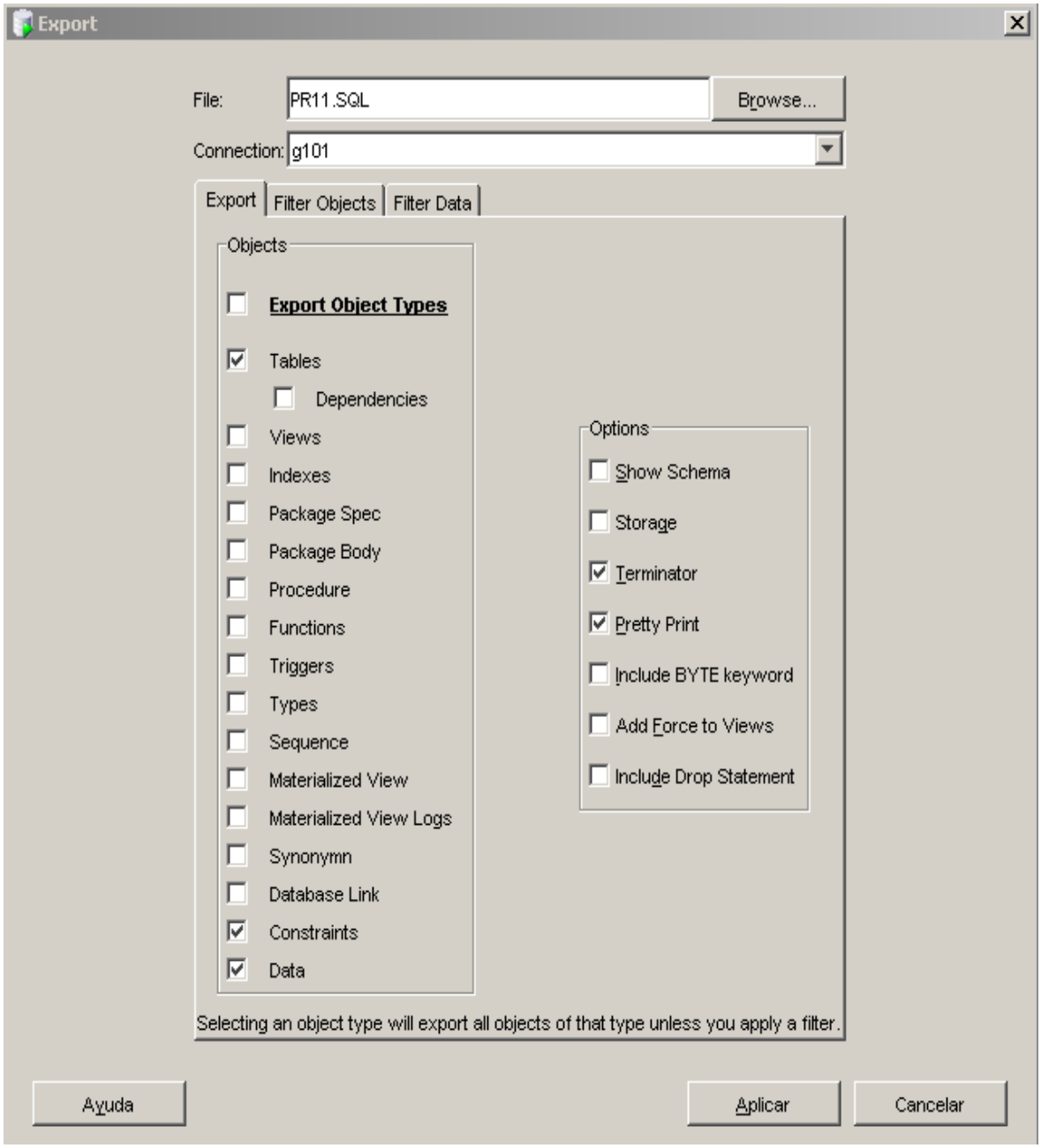

Figura 4.3.8 “Export DDL (and data)”.

Creación y edición de secuencias.

Para crear una nueva secuencia se pulsa el botón derecho sobre icono “Sequences” de la conexión, figura 4.3.9.

Figura 4.3.9 "Creación y edición de secuencias".

Elegir "New sequence...", figura 4.3.10.

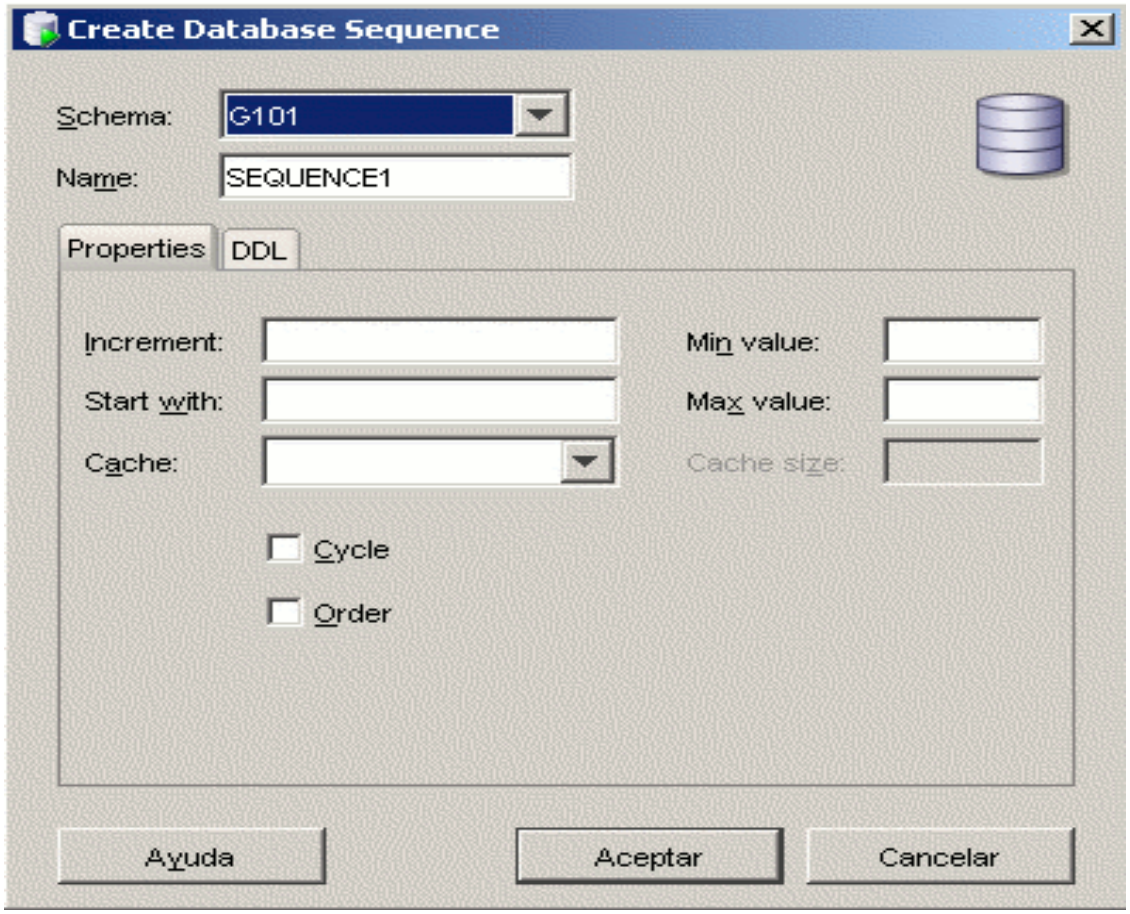

Figura 4.3.10 New Sequence.

Una vez creada la secuencia, se puede acceder a su definición y valor actual haciendo doble-click sobre su icono, figura 4.3.11.

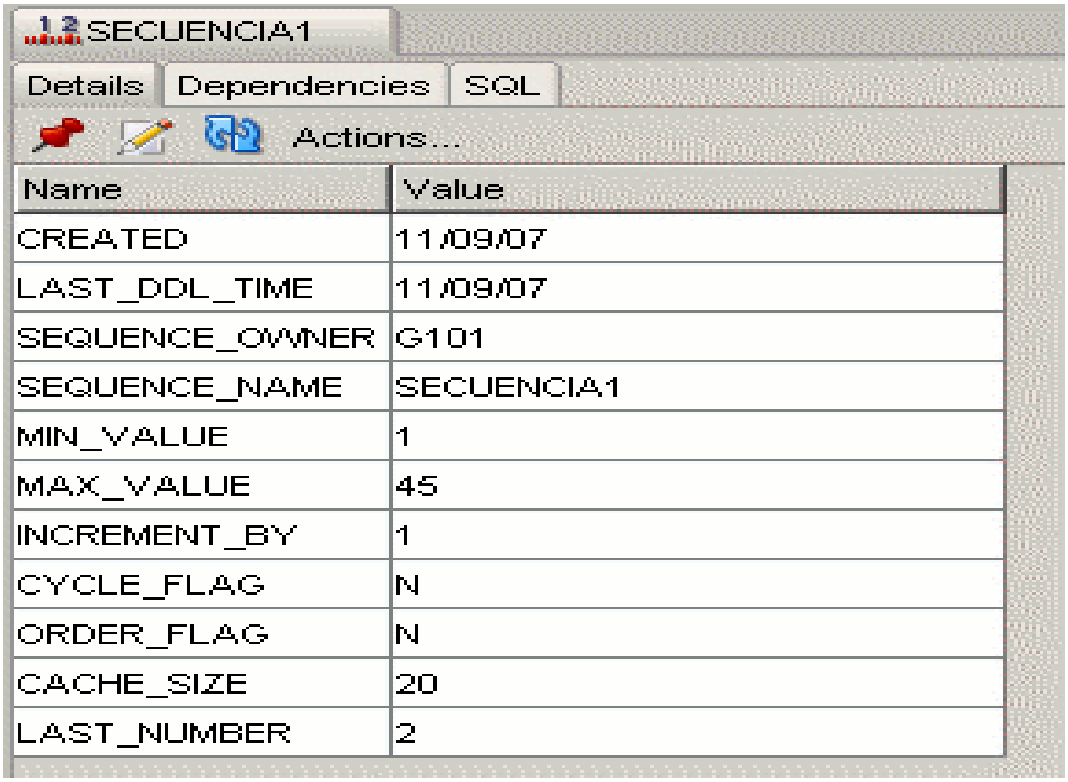

Figura 4.3.11 Definición y valor actual.

Como cualquier otro objeto, para modificar la definición de una secuencia se puede optar por:

1. Seleccionar la secuencia (doble-click sobre su icono) y elegir la pestaña “Details” y pulsar sobre el icono “Edit”
2. Pulsar con el botón derecho sobre el icono de la secuencia que se quiere modificar y elegir la opción “Edit...”, figura 4.3.12.

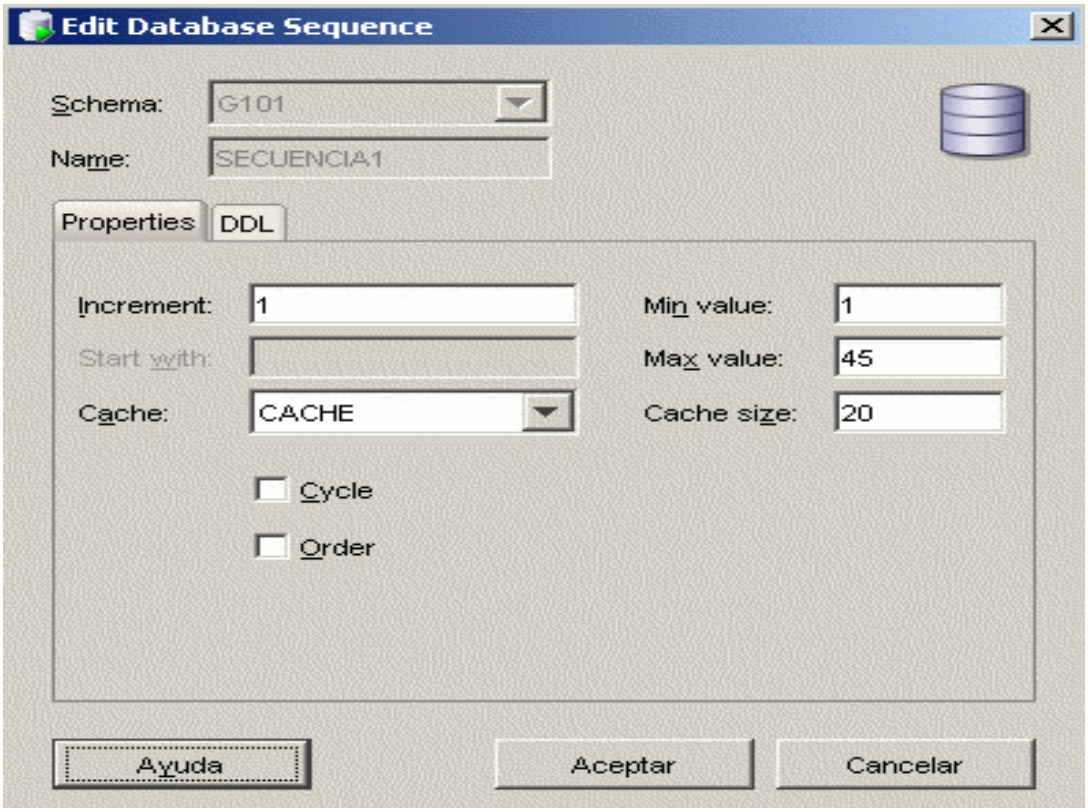

Figura 4.3.12 Definición y valor actual.

Creación, compilación y ejecución de funciones y procedimientos.

Para crear una función o procedimiento se pulsa con botón derecho sobre el icono "Functions" o "Procedures" del navegador de objetos y se elige la opción "New function" o "New Procedure" respectivamente, figura 4.3.13.

Figura 4.3.13 New Function, Procedure.

Se introduce el nombre del procedimiento o función, los nombres de los parámetros, sus tipos de datos, el modo del parámetro y los valores por defecto. Para el caso de las funciones también hay que especificar el tipo del resultado de la función (parámetro <Return>), figura 4.3.14 y figura 4.3.15.

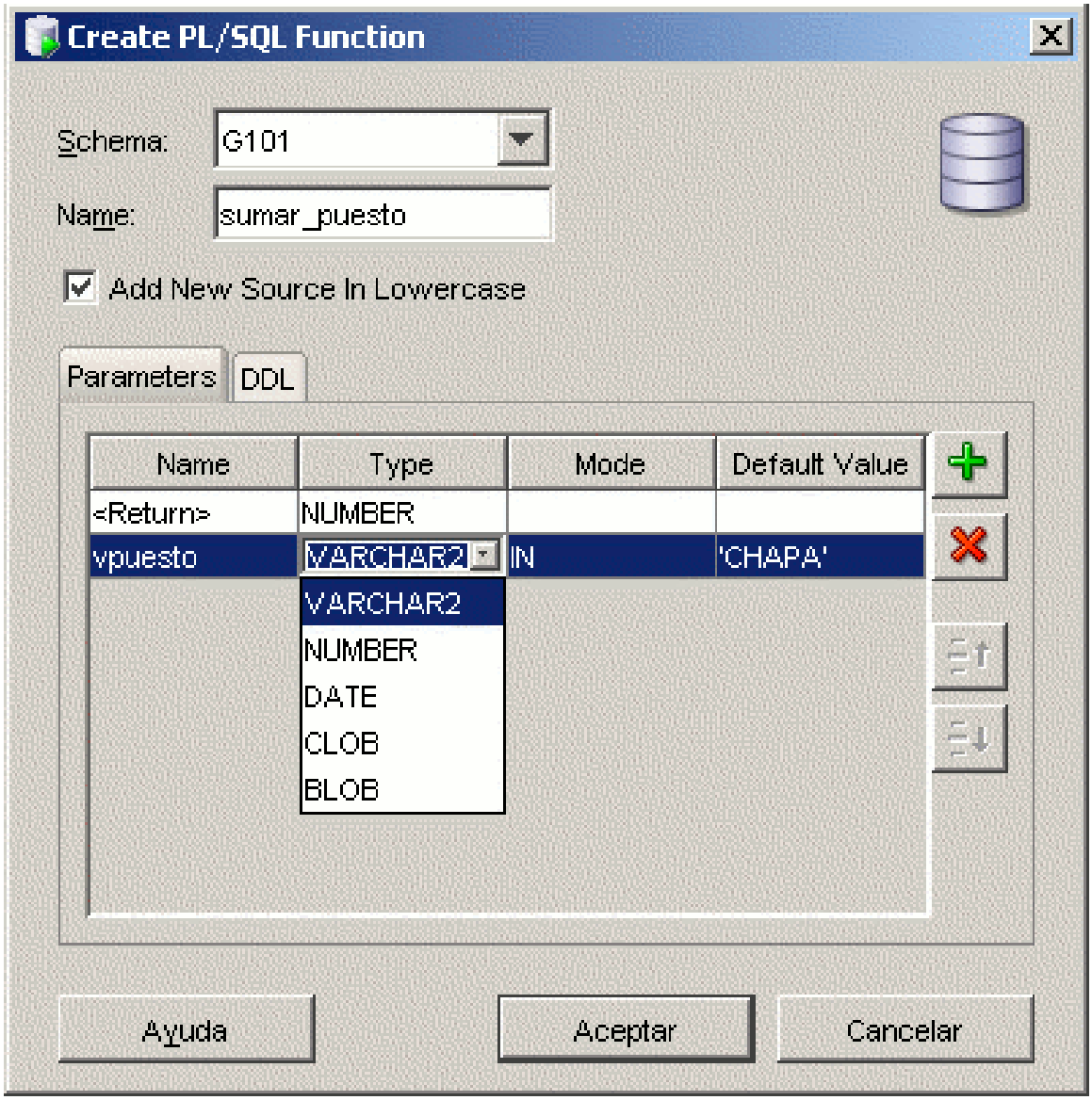

Figura 4.3.14 Parametros New Function Procedure.

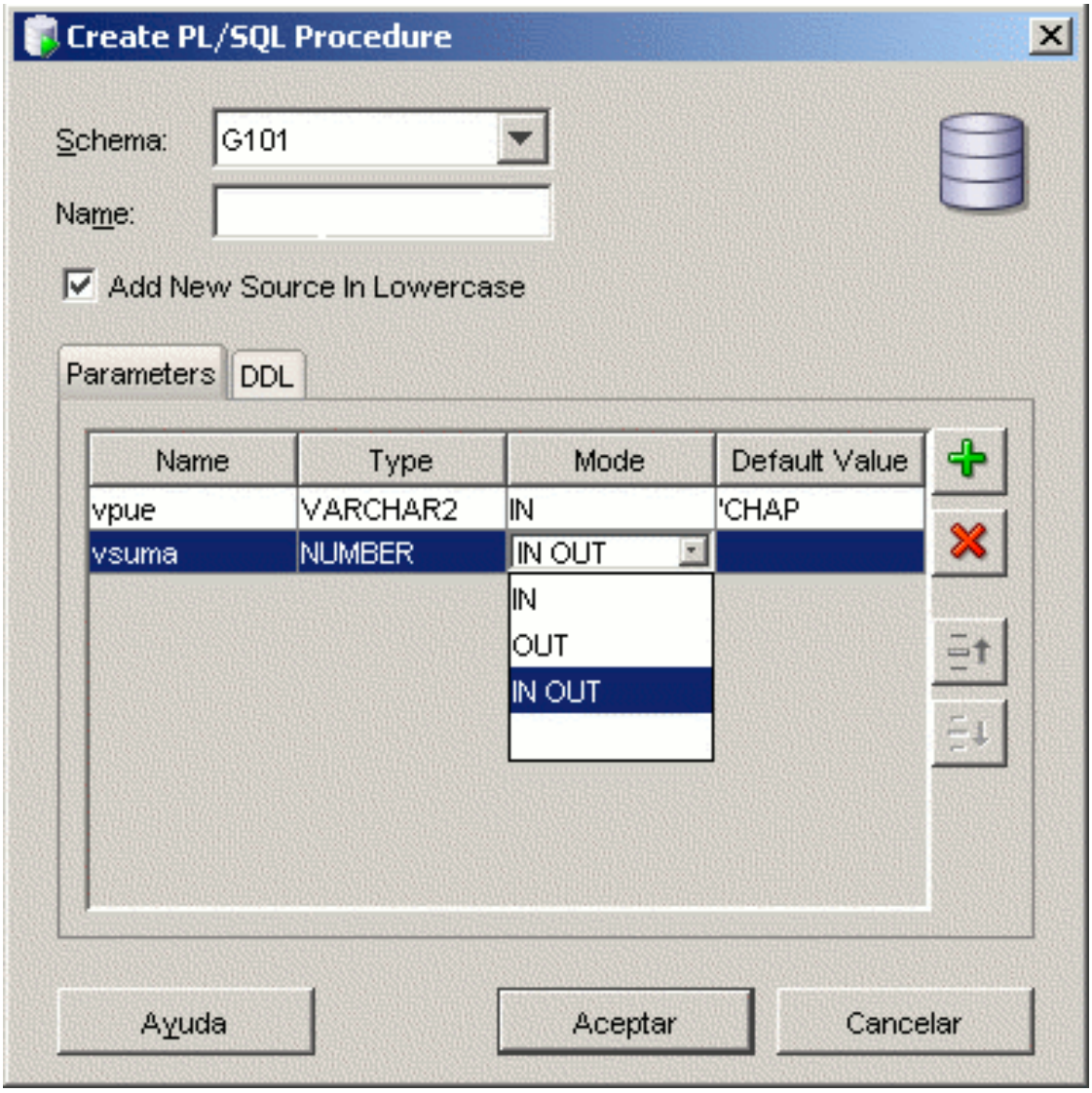

Figura 4.3.15 Parametros New Function , Procedure.

El asistente abre una pestaña de edición con el código generado para la función o el procedimiento con la cabecera especificada y el cuerpo vacío, figura 4.3.16 y figura 4.3.17.

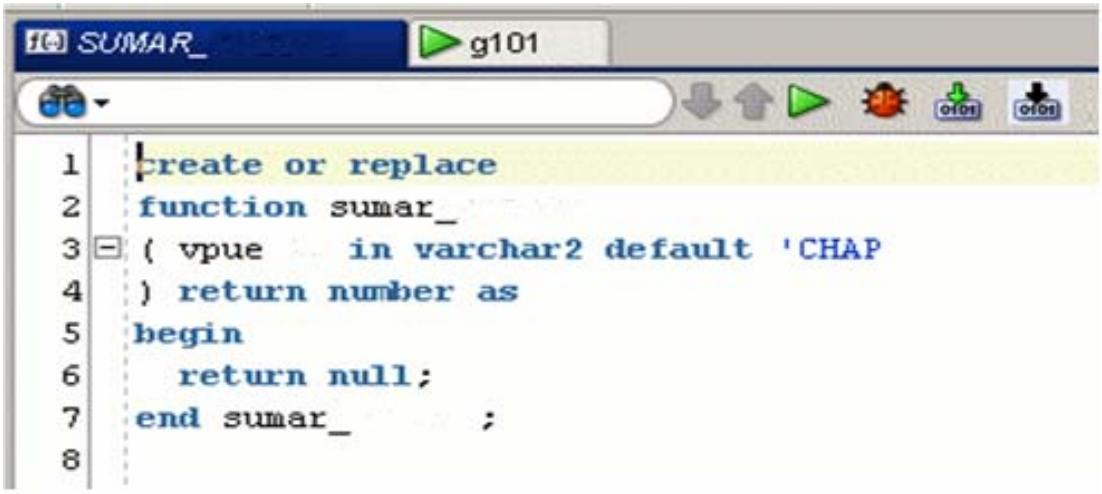

Figura 4.3.16 Pestaña de edición.

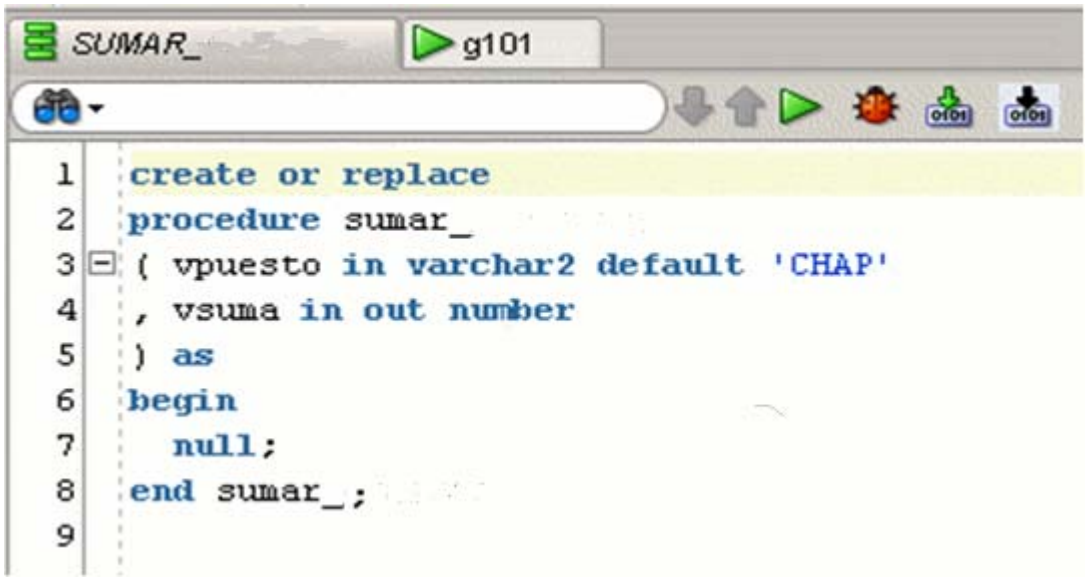

Figura 4.3.17 Pestaña de edición.

Para compilar se pulsa el icono. También se compila automáticamente cuando se almacena el procedimiento o función en la base de datos (icono). Los errores y warnings aparecen en el panel “Log” en la pestaña “Compiler”. Junto a la palabra error o warning se indica entre paréntesis la línea y la columna en la que se ha producido el error. Las sentencias erróneas aparecen subrayadas en rojo y los warnings subrayados en amarillo en la ventana de edición, figura 4.3.18.

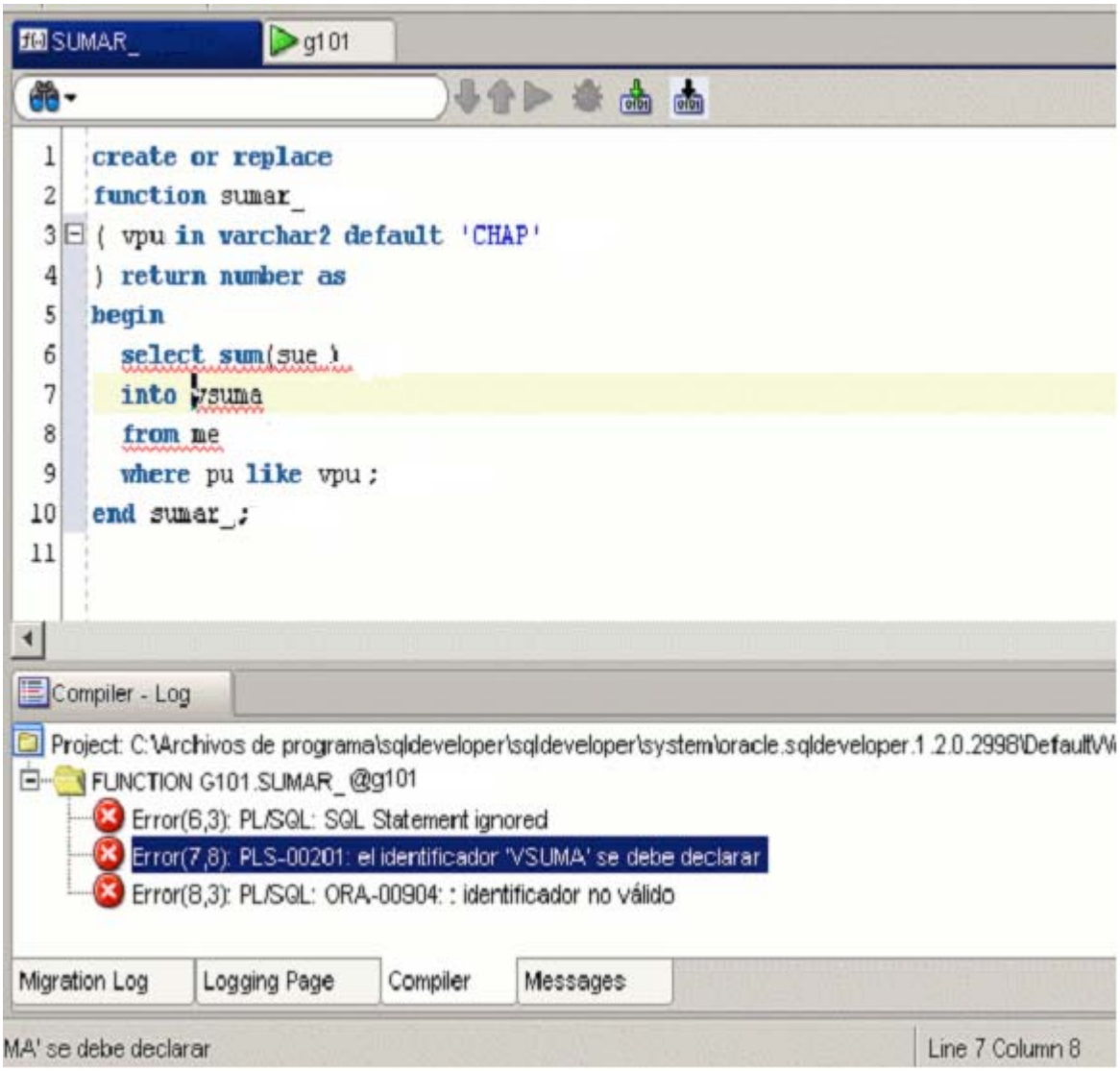

Figura 4.3.18 Errores y Warnings

Para ejecutar un procedimiento o función se pulsa el icono de la ventana de edición o se elige la opción “Run...” que aparece tras pulsar con el botón derecho sobre el icono de la función o procedimiento en el navegador de objetos, figura 4.3.19.

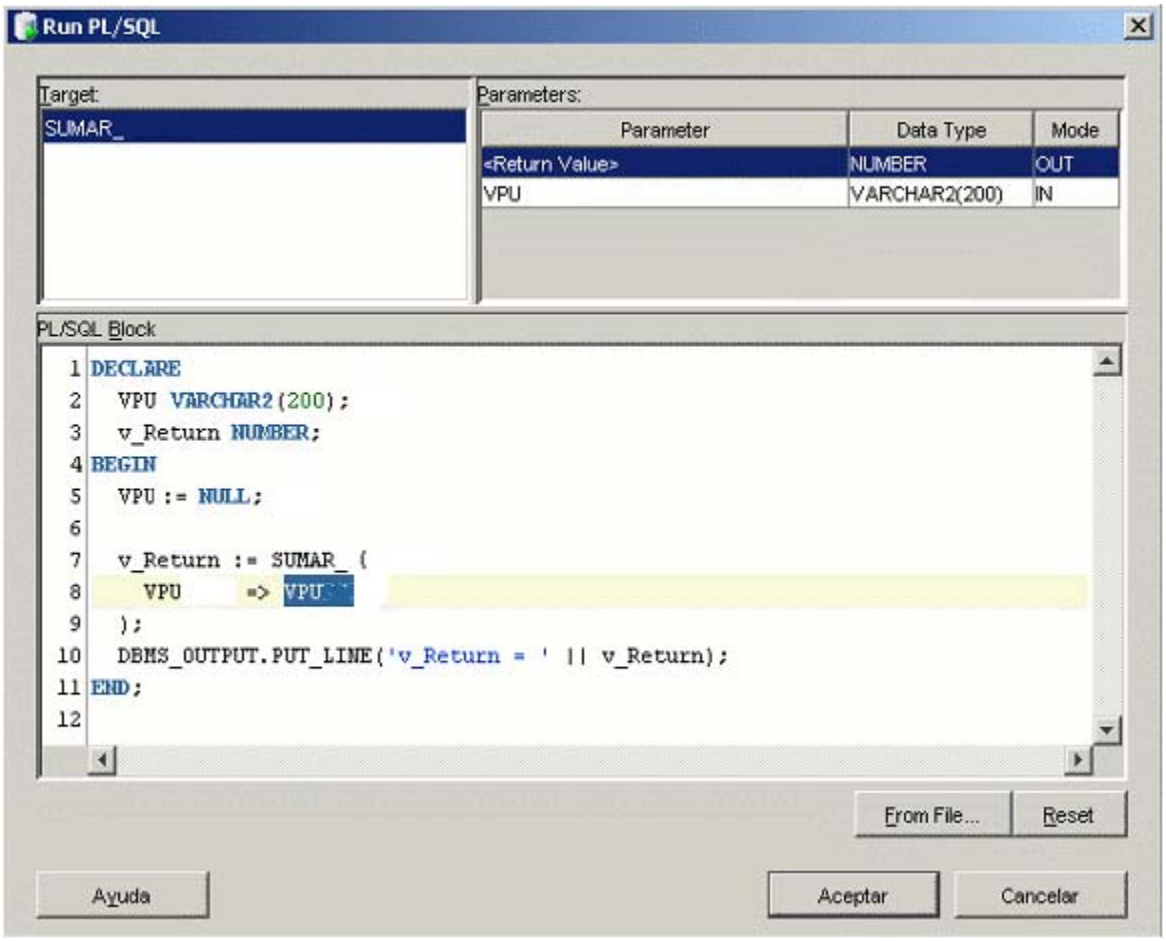

Figura 4.3.19 Ejecución de procedimiento o función.

Para poder ejecutar una función o procedimiento, SQL Developer crea un bloque con las variables necesarias para pasar los parámetros en la llamada a la función o procedimiento, debiéndose sustituir los valores por defecto predefinidos, por el valor actual que se le quiere dar al parámetro para la ejecución:

Inicialmente: VARIABLE => VARIABLE

Se sustituye por: VARIABLE => valor_actual, figura 4.3.20.

```
4  BEGIN
5    VPU:= NULL;
6
7    v_Return := SUMAR_(
8      VPU => 'CHAP'
9    );
10   DBMS_OUTPUT.PUT_LINE('v_Return = ' || v_Return);
```

Figura 4.3.20 Bloque con variables.

Tras pulsar el botón “Aceptar”, el bloque que contiene la llamada a la función o el procedimiento se ejecuta y se muestran los resultados en el panel “Log” en la pestaña “Running”, figura 4.3.21.

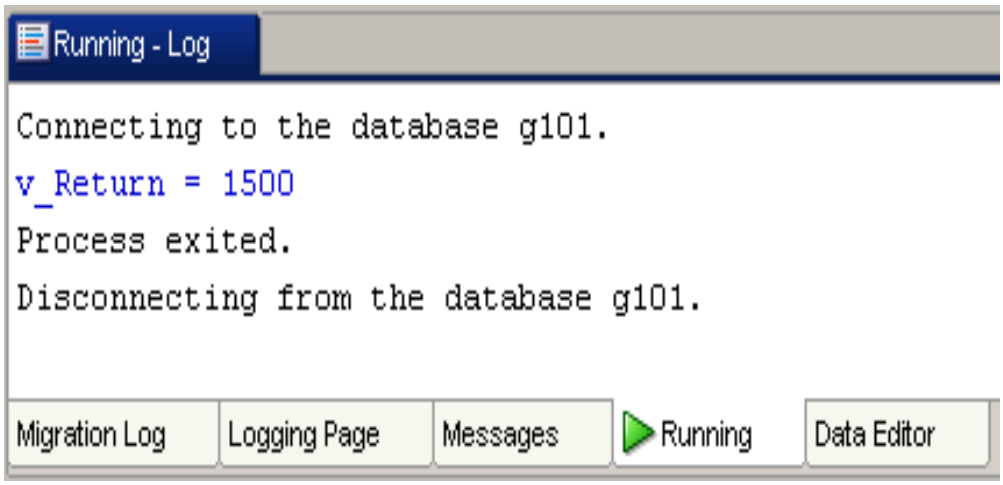

Figura 4.3.21 Resultados en panel log.

Depuración de funciones y procedimientos.

Para poder depurar es necesario que el usuario tenga los privilegios:

- DEBUG ANY PROCEDURE
- DEBUG CONNECT SESSION

El Administrador ya tiene estos privilegios asignados. Para asignarlos en una instalación local de Oracle Express debe consultar el apartado “Asignación de Privilegios y Roles” más abajo.

Para comenzar la depuración, en la ventana de edición del procedimiento o función (botón derecho sobre el icono del objeto y elegir “Edit...”) introducir los puntos de ruptura deseados dentro del cuerpo del procedimiento o función, como mínimo uno para que la ejecución del depurador se interrumpa y se pueda avanzar paso a paso viendo los valores de las distintas variables. Los puntos de ruptura se especifican pulsando con el ratón sobre el número de la línea donde se quiere introducir (el número de línea se sustituye por un círculo rojo), figura 4.3.22.

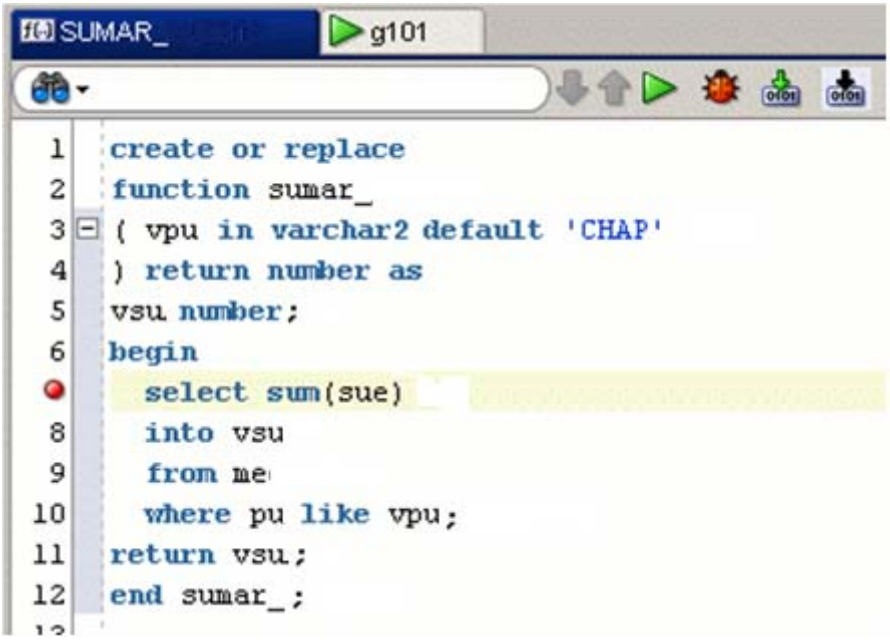

Figura 4.3.22 Puntos de ruptura.

Antes de iniciar la depuración es necesario compilar el procedimiento o función de manera especial para que pueda ser depurado. Esto se realiza pulsando el icono .

Para iniciar el depurador, se pulsa sobre el icono de la ventana de edición del procedimiento o función. A continuación se mostrará una ventana similar a la que aparece cuando se ejecuta un procedimiento o función, en la que hay que establecer los valores actuales de los parámetros como se describió

anteriormente. A continuación el flujo de control (indicado por una flecha roja) se detiene en el primer punto de ruptura establecido, pudiéndose ver los valores de las distintas variables en las pestañas “Data” y “Watches” del depurador, figura 4.3.23.

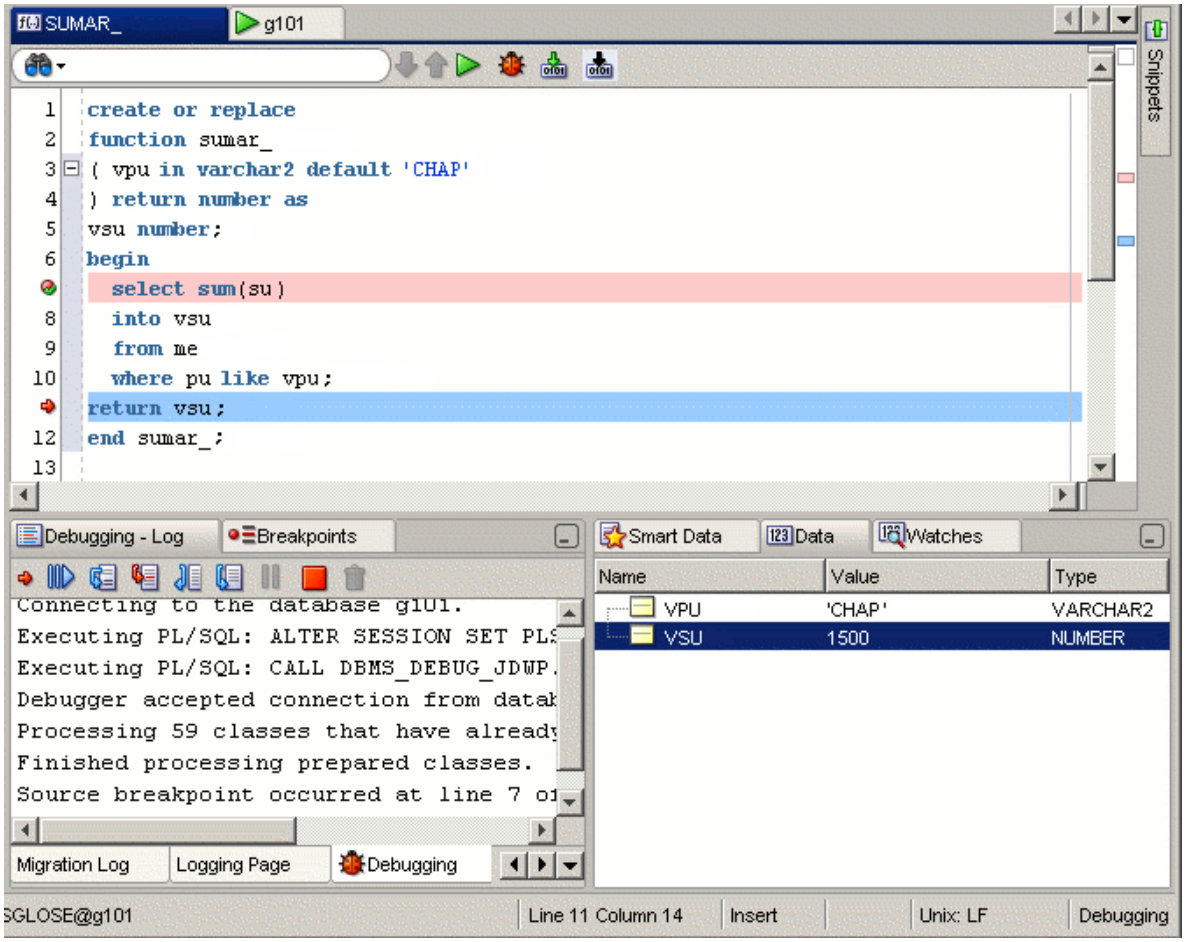

Figura 4.3.23 Inicio del depurador.

Las posibles acciones del depurador se encuentran en el menú “Debug”, las más típicas son:

- Avanzar sin entrar: F8
- Avanzar entrando: F7
- Avanzar hasta el cursor: F4
- Avanzar hasta el próximo punto de ruptura: F9

Creación y compilación de disparadores.

Para crear un disparador se pulsa con botón derecho sobre el icono “Triggers” del navegador de objetos y se elige la opción “New Trigger...”, figura 4.3.24.

Figura 4.3.24 New Trigger.

Se introduce:

- El nombre del disparador
- El tipo de disparador
- La tabla asociada al disparador
- Si el disparador es de sentencia (“Statement Level”) o de tupla (“Row Level”)
- El momento del disparo (“Before” o “After”)
- Los eventos de disparo (“Insert”, “Delete” o “Update”)
- Para el caso del evento “Update” se pueden especificar sobre qué columnas debe ser la actualización
- Para el caso de disparadores de tupla, se puede especificar una condición para la cláusula “When” y cambiar en “Referencing” el nombre de las variables de referencia de tupla por defecto (old y new), figura 4.3.25.

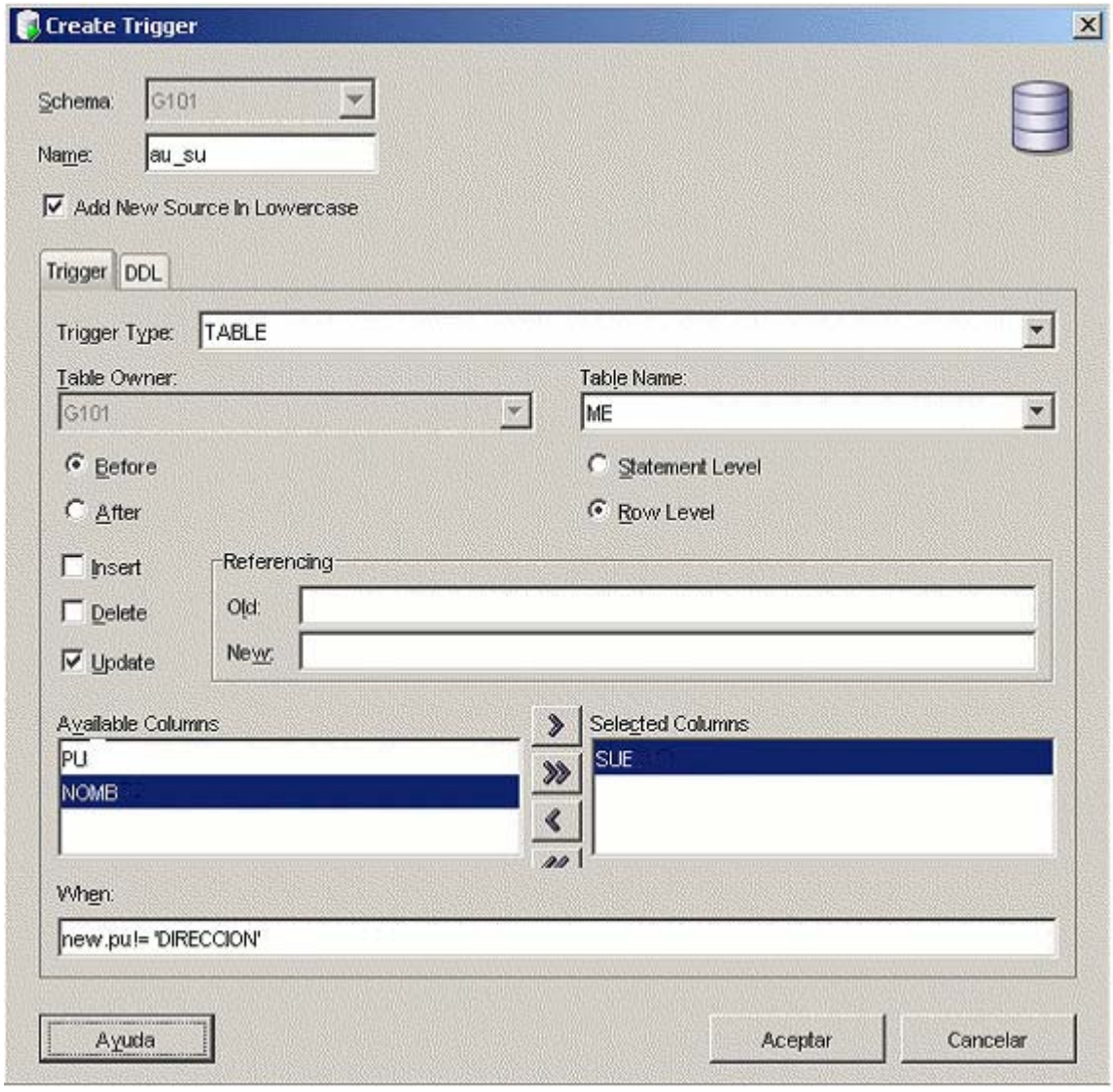

Figura 4.3.25 Create Trigger.

El asistente abre una pestaña de edición con el código generado para el disparador con la cabecera especificada y el cuerpo vacío, figura 4.3.26.

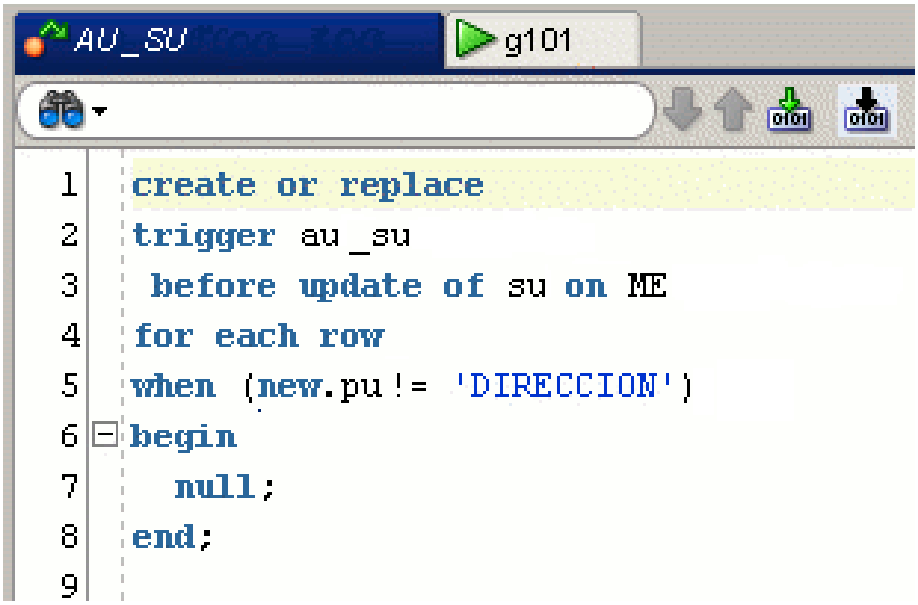

Figura 4.3.26 Código de Trigger.

Para compilar se pulsa el icono. También se compila automáticamente cuando se almacena el disparador en la base de datos (icono). Al igual que para el caso de los procedimientos y las funciones, los errores y warnings aparecen en el panel "Log" en la pestaña "Compiler". Junto a la palabra error o warning se indica entre paréntesis la línea y la columna en la que se ha producido el error. Las sentencias erróneas aparecen subrayadas en rojo y los warnings subrayados en amarillo en la ventana de edición del disparador.

Depuración de disparadores.

La versión actual de SQL Developer no permite la depuración mediante traza del código de los disparadores. La manera tradicional de trazar los disparadores consiste en mostrar mensajes en pantalla.

Para mostrar un mensaje desde un bloque PL/SQL, por ejemplo desde el cuerpo de un disparador, se utiliza la función:

DBMS_OUTPUT.PUT_LINE (cadena);

Por ejemplo: DBMS_OUTPUT.PUT_LINE ('El valor de var es: ' || var);

Para que los mensajes aparezcan por pantalla es necesario activar la salida del servidor (SERVEROUTPUT). Desde SQL*PLUS se realiza mediante la sentencia:
SQL> SET SERVEROUTPUT ON
En SQL Developer, la salida del servidor se establece activando el icono que se encuentra en la pestaña "DBMS Output" de las ventanas de edición de SQL (SQL Worksheet).

Asignación de privilegios y roles.

Para asignar privilegios hay crear una conexión en SQL Developer para el administrador “SYSTEM”, abrir la categoría “Other Users” en el navegador de objetos y elegir la opción “Edit User” al pulsar con el botón derecho sobre el usuario al que se le quieren dar los privilegios, figura 4.3.27.

Figura 4.3.27 Asignación de Privilegios.

En la pestaña “System Privileges” activar la casilla “Granted” para los privilegios que se quieren conceder al usuario y pulsar aplicar, figura 4.3.28.

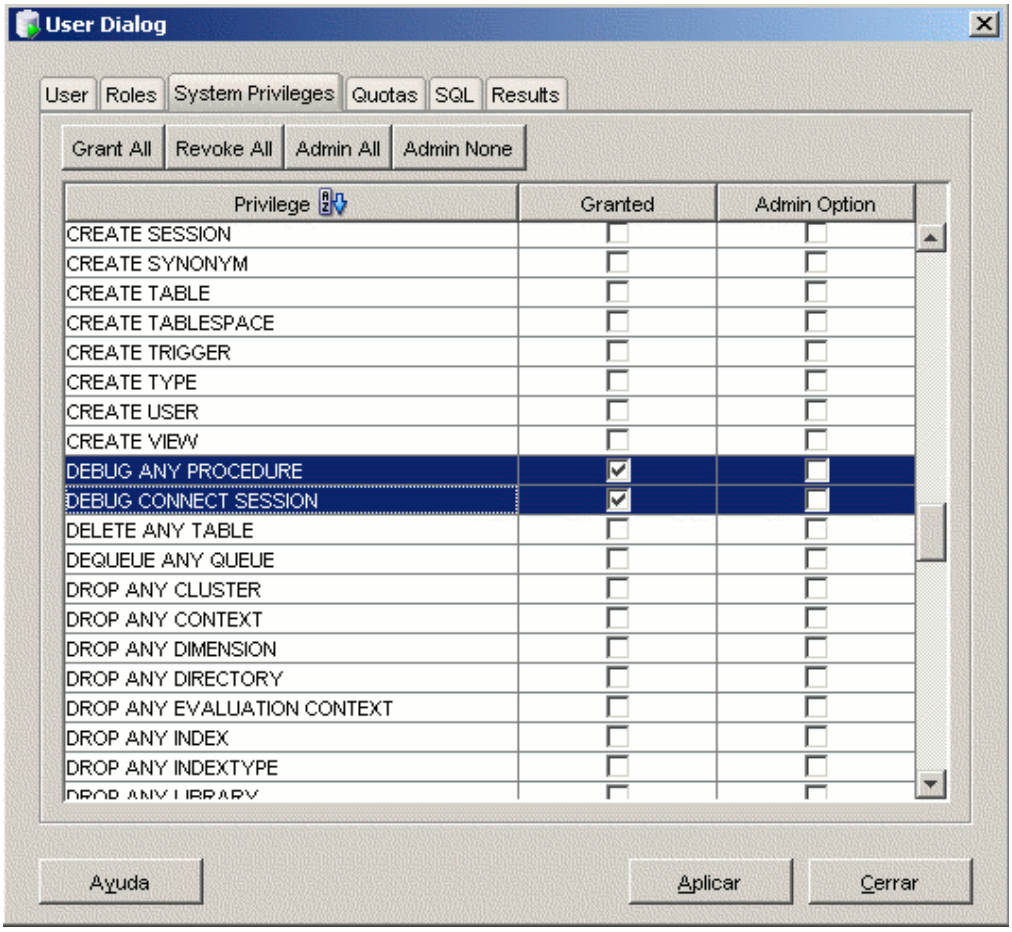

Figura 4.3.28 Concesión de privilegios.

Para poder conceder roles, el usuario debe tener previamente concedido por el administrador el privilegio “CREATE ROLE”.

4.4 Integración, pruebas y mantenimiento.

Integración.

Es el proceso de construir un sistema de software combinando componentes a una unidad de trabajo. La integración de componentes debe proceder de una manera ordenada, siguiendo una secuencia de función en función.

Esto permite que las capacidades operacionales del software sean mostradas tempranamente, dando la confianza de que el proyecto está progresando de manera satisfactoria.

El sistema de mediación se integró a partir del módulo de recolección, dado que este tiene la tarea de establecer la conectividad con las plataformas de conmutación celular (transferencia AFT), para después enviar bloques de CDRs hacia el módulo de parseo y su posterior extracción de archivos entre equipos FTP (File Transfer Protocol).

El módulo de parseo recibe los bloques de CDRs para procesarlos y por medio del método remoto de invocación RMI (Java Remote Method Invocation), consulta la información que requiere del módulo de catálogos, para almacenar los CDRs en la base de datos y de esta forma puedan estar disponibles para otras aplicaciones.

La conexión con la base de datos hacia el módulo de catálogos se realiza mediante un lenguaje de programación java de esta forma se manipulan los datos de un módulo a otro y se filtra a su vez la información necesaria a las diferentes aplicaciones clientes.

Pruebas.

Las pruebas de software, son los procesos que permiten verificar y revelar la calidad de un producto de software. Son utilizadas para identificar posibles fallos de implementación, calidad o usabilidad de un programa. Básicamente es una fase en el desarrollo de software consistente en probar las aplicaciones construidas.

Las pruebas de software se integran dentro de las diferentes fases del ciclo del software dentro de la ingeniería de software. Para determinar el nivel de calidad se deben efectuar las pruebas que permitan comprobar el grado de cumplimiento respecto a las especificaciones iniciales del sistema.

Para el caso en particular al Sistema de Mediación, las pruebas que se aplicaron para verificar el correcto funcionamiento fueron:

- La prueba de la caja blanca. Se introdujeron varios CDRs de prueba para determinar el comportamiento del código del programa en cada una de las etapas o módulos del sistema de mediación.
- La prueba de caja negra. Se verificó que no existiera errores en establecer la conexión con las plataformas de conmutación celular, para poder asegurar el correcto ingreso de bloques de CDRs de entrada, así como verificar que los CDRs de salida tuvieran la estructura adecuada y se almacenaran de forma correcta en la base de datos.
- La prueba de integración. Se probó cada módulo por separado y también al momento de unir dos módulos para verificar su perfecto ensamblaje.
- La prueba de validación. Se probó el sistema completamente ensamblado, verificando que los resultados arrojados en fechas, datos, tamaños, etc., fueran los correctos y cumplieran con los requerimientos y objetivos que se plantearon al inicio del desarrollo.
- La prueba de stress. Como el sistema de mediación va a procesar grandes volúmenes de CDRs diarios, es necesaria esta prueba, para verificar el comportamiento del sistema al procesar 4 millones de CDRs en 2 horas.

Tipos de pruebas.

Prueba de la caja blanca.

Su objetivo principal es asegurar el correcto funcionamiento de las interfaces, o flujo de datos entre componentes.

Su diseño está fuertemente ligado al código fuente. El verificador escoge distintos valores de entrada para examinar cada uno de los posibles flujos de ejecución del programa para cerciorarse de que se devuelven los valores de salida adecuados.

Aunque las pruebas de caja blanca son aplicables a varios niveles (unidad, integración y sistema), habitualmente se aplican a las unidades del software. Se

comprueban los flujos de ejecución dentro de cada unidad (función, clase, modulo, etc.), pero también pueden verificar los flujos entre unidades durante la integración, e incluso entre subsistemas, durante las pruebas del sistema.

Las pruebas de caja blanca intentan garantizar que:

- Se ejecutan al menos una vez todos los caminos independientes de cada módulo.
- Se utilizan las decisiones en su parte verdadera y en su parte falsa.
- Se ejecuten todos los bucles en sus límites.
- Se intenta usar todas las estructuras de datos internas.

El principal factor que se debe considerar al inicio de las pruebas es el tamaño del módulo a probar, se debe considerar si el tamaño del módulo permitirá probar adecuadamente toda su funcionalidad de manera sencilla y rápida. También es importante separar los módulos de acuerdo a su funcionalidad, si los módulos son muy grandes y contienen muchas funcionalidades, estos se volverán más complejos de probar y al encontrar algún error será más difícil ubicar la funcionalidad defectuosa y corregirla.

En el sistema, cada componente debe ser probado en su totalidad (óvalos) y también sus interfaces o comunicaciones con los demás componentes (flechas), figura 4.4.1.

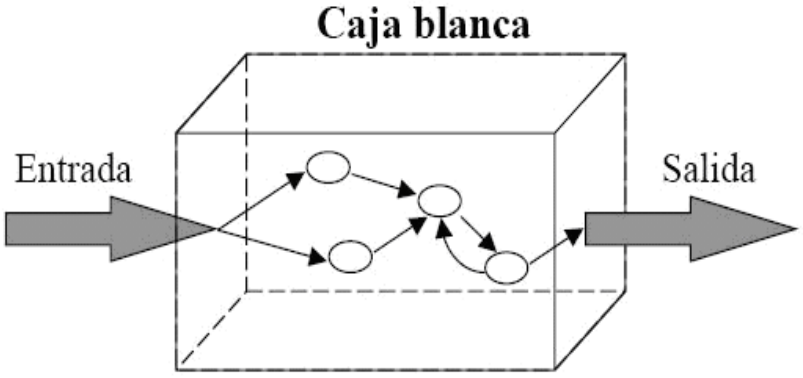

Figura 4.4.1 Diagrama de la caja blanca.

Prueba de caja negra.

Es la prueba que estudia al sistema desde el punto de vista de las entradas que recibe y las salidas o respuestas que produce, sin tener en cuenta su funcionamiento interno. Solo interesa su forma de interactuar con el medio que le rodea, entendiendo ¿Qué es lo que hace?, pero sin dar importancia en ¿Cómo lo hace?. Por lo que debe de estar bien definidas sus entradas y salidas, es decir, su interfaz, pero no es importante conocer los detalles internos de su funcionamiento.
Esta prueba se centra principalmente en los requisitos funcionales del software, permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. En ellas se ignora la estructura de control, concentrándose en los requisitos funcionales del sistema y ejercitándolos, figura 4.4.2.

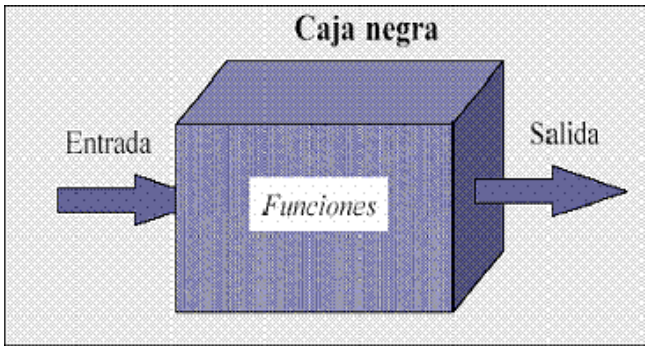

Figura 4.4.2 Diagrama de la caja negra.

Prueba de Integración.

Es una técnica sistemática para construir la estructura del programa mientras al mismo tiempo, se lleva a cabo pruebas para detectar errores asociados con la interacción. Consiste en verificar que un gran conjunto de partes de software funcionan juntos.
Se llevan a cabo durante la construcción del sistema, involucran a un número creciente de módulos y terminan probando el sistema como un conjunto.

Estas pruebas cubren todo el sistema y pretenden cubrir plenamente la especificación de requisitos del usuario. Estas pruebas se pueden plantear desde un punto de vista estructural o funcional.

Los tipos fundamentales de integración son los siguientes:

- Integración incremental: Se combina el siguiente módulo que se debe probar con el conjunto de módulos que ya están probados y se va incrementando progresivamente el número de componentes a probar.
- Integración no incremental: Se prueba cada módulo por separado y posteriormente se integran todos de una vez realizando las pruebas pertinentes.

Pruebas de regresión.

El objetivo es comprobar que los cambios sobre un componente del sistema, no introducen un comportamiento no deseado o errores adicionales en otros componentes no modificados.

Son cualquier tipo de pruebas de software que intentan descubrir las causas de nuevos errores, carencias de funcionalidad, o divergencias funcionales con respecto al comportamiento esperado del software, incluidos por cambios recientemente realizados en partes de la aplicación que anteriormente no eran propensas a errores. Esto implica que el error se presenta como consecuencia inesperada del cambio en el sistema.

Este tipo de cambio puede ser debido a prácticas no adecuadas de control de versiones, falto de consideración acerca del ámbito o contexto de producción final y extensibilidad del error que fue corregido, o simplemente una consecuencia del rediseño de la aplicación.

Estas pruebas se deben llevar a cabo cada vez que se hace un cambio en el sistema, tanto para corregir un error como para realizar una mejora. No es suficiente probar solo los componentes modificados o añadidos, o las funciones que en ellos se realizan, sino que también es necesario controlar que las

modificaciones no produzcan efectos negativos sobre el mismo u otros componentes.

Las pruebas de regresión pueden incluir:

- La repetición de los casos de pruebas que se han realizado anteriormente y están directamente relacionados con la parte del sistema modificado.
- La revisión de los procedimientos manuales preparados antes del cambio, para asegurar que permanecen correctamente.
- La obtención impresa del diccionario de datos de forma que se compruebe que los elementos de datos que han sufrido algún cambio son correctos.

Pruebas de validación.

Son las pruebas realizadas sobre un software completamente integrado para evaluar el cumplimiento de los requisitos especificados. Estas empiezan tras la culminación de las pruebas de integración, cuando se han ejercitado los componentes individuales. Se ha terminado de ensamblar el software como paquete y se han descubierto y corregido los errores de interfaz.

La prueba se concentra en las acciones visibles para el usuario y en la salida del sistema que este puede reconocer.

La validación se define de una forma simple, cuando el software funciona de tal manera que satisface las expectativas razonables del cliente. La validación del software se logra mediante una serie de pruebas que demuestren que se cumplen los requisitos. Un plan de prueba delinea la clase de pruebas que se aplicarán y un procedimiento de prueba define los casos de prueba específicos.

La revisión de la configuración, es un elemento importante del proceso de validación. Su objetivo, es asegurar que todos los elementos de la configuración del software se hayan desarrollado apropiadamente, estén catalogados y tengan el detalle suficiente para reforzar la fase de soporte del ciclo de vida del software.

Prueba de stress.

Consiste en la simulación de grandes cargas de trabajo para observar de qué forma se comporta la aplicación ante situaciones de uso intenso.

- Prueba de stress de componentes: Se aíslan los servicios y componentes que conforman el sistema, se infieren los métodos de navegación, de funcionamiento y de interfaz de estos servicios y componentes y se crea un cliente de prueba que llame a dichos métodos.

  Para aquellos métodos que tienen acceso a un servidor de base de datos o a cualquier otro componente, se puede crear un cliente que proporcione datos simulados en el formato previsto.
  La idea es forzar cada componente de forma aislada, más de lo que la aplicación podría experimentar en condiciones normales.
- Prueba de stress de integración: Después de forzar cada componente individual, se deberá someter a una situación de stress a toda la aplicación con todos sus componentes y servicios.
  Estas pruebas están íntimamente relacionadas con las interacciones con otras estructuras de datos, procesos y servicios tanto de los componentes internos como de otros servicios externos de la aplicación.

Prueba alfa.

Se lleva a cabo por un cliente, en el lugar de desarrollo. Se usa el software de forma natural con el desarrollador como observador del usuario. Las pruebas alfa se llevan a cabo en un entorno controlado. Para que tenga validez, se debe primero crear un ambiente con las mismas condiciones que se encontrarán en las instalaciones del cliente. Una vez logrado esto, se procede a realizar las pruebas y a documentar los resultados.

Prueba beta.

Se lleva a cabo por los usuarios finales del software en los lugares de trabajo de los clientes. A diferencia de la prueba alfa, el desarrollador no está presente normalmente. Así, la prueba beta es una aplicación en vivo" del software en un entorno que no puede ser controlado por el desarrollador. El cliente registra todos los problemas (reales o imaginarios) que encuentra durante la prueba beta e informa a intervalos regulares al desarrollador.

Como resultado de los problemas informados durante la prueba beta, el desarrollador del software lleva a cabo modificaciones y así prepara una versión del producto de software para toda clase de clientes.

Mantenimiento.

El mantenimiento de software es el proceso de mejorar y optimizar el software, así como de remediar los defectos, con el fin de asegurar que se continúen desempeñando las funciones deseadas. Esta fase involucra cambios al software en orden de corregir defectos y dependencias encontradas durante su uso, tanto en la adición de nueva funcionalidad para mejorar la usabilidad y en la aplicabilidad del software.

A la entrega del software al usuario final, dentro del contrato o el acuerdo, también se puede establecer como realizar el mantenimiento. El mantenimiento puede ser variado ya que depende de las necesidades de cada usuario, por esta razón se pueden tener varios tipos de mantenimiento.

Tipos de mantenimiento.

Mantenimiento preventivo.

Consiste en usar el producto de software antes de su entrega, para detectar y corregir fallos latentes antes de que se conviertan en fallos efectivos. El mantenimiento preventivo proporciona mejoras para los usuarios, mejora de la documentación del sistema y la recodificación para mejorar el rendimiento del software, su mantenibilidad u otros atributos. Se incluyen sentencias para comprobar la validez de los datos de entrada, reestructuración de los programas para aumentar su legibilidad o incluir nuevos comentarios.

Este tipo de mantenimiento se diferencia del resto de los tipos de mantenimiento, en que, mientras que el resto se produce generalmente tras una petición por parte del cliente o del usuario final, el preventivo se produce tras un estudio de posibilidades de mejora en los diferentes módulos del sistema.

Uno de sus principales objetivos en este tipo de mantenimiento es evitar los fallos antes de que estos ocurran.

Mantenimiento correctivo.

Tiene por objeto corregir defectos y fallas detectados en la operación del producto de software después de su entrega. Un defecto en un sistema es una característica del sistema con el potencial de provocar un fallo. Un fallo se produce cuando el comportamiento del sistema difiere con respecto al comportamiento definido en la especificación. Los fallos en un sistema de software pueden ser de:

- Procesamiento (salidas incorrectas de un programa).
- Rendimiento (tiempo de respuesta demasiado alto).
- Programación (inconsistencias en el diseño).

- Documentación (inconsistencias entre la funcionalidad de un programa y el manual del usuario).

Se puede dividir el mantenimiento correctivo en:

- Programado: Es el que se efectúa cuando la falla no es urgente, difiriendo de la ejecución para el momento más oportuno y con la reparación más adecuada.
- Crítico: Es cuando la falla es urgente, de la manera más directa en el menor tiempo posible y con la mejor preparación que permitan las circunstancias.
- Normal: Se aplica a los sistemas que al fallar no afectan la seguridad ni la producción, por lo que su reparación puede ser programada y resuelta con los recursos normales.
- Emergente: Se produce cuando las fallas ponen en peligro la seguridad o integridad de la producción.

Mantenimiento perfectivo.

Consiste en modificar el producto de software, después de su entrega, para mejorar su rendimiento o su mantenimiento. Este mantenimiento proporciona mejoras para los usuarios, mejora de la documentación del sistema y la recodificación para mejorar el rendimiento del software u otros atributos.
Los requisitos pueden cambiar con el tiempo para ajustarse a nuevas necesidades o para mejorar las prestaciones actuales, por lo que pueden ir desde una pequeña modificación en un módulo, hasta la adición de nuevos módulos. Este mantenimiento no está únicamente enfocado a mejorar técnicamente una solución, sino que también incluye un proceso continuo de optimización a nivel funcional y de procesos. Este mantenimiento se enfoca en:

- La optimización constante del rendimiento de las aplicaciones mediante análisis técnicos.

- La adaptación de las aplicaciones a las nuevas necesidades del cliente en función de los análisis funcionales.
- La detección de posibles puntos a mejorar en el diseño y uso de las bases de datos mediante el análisis de la misma.

Por lo tanto el principal objetivo del mantenimiento perfectivo es llevar a cabo las tareas y procesos necesarios para identificar aquellos puntos susceptibles de mejora. Este tipo de mantenimiento se divide en:

- Mantenimiento de ampliación (incorpora nuevas funcionalidades).
- Mantenimiento de eficiencia (mejora la eficiencia de ejecución).

Este mantenimiento es derivado de nuevos requisitos en cuanto a funcionalidad y como consecuencia de una posible optimización de rendimiento. Estas actividades están motivadas por cambios introducidos por el usuario, más allá del alcance y objetivos iniciales del sistema.

Mantenimiento tecnológico.

Este tipo de mantenimiento está pensado para aquellas acciones que están relacionadas con el software instalado. Está pensado para dar una cobertura tecnológica a las aplicaciones instaladas en el cliente.

Cuando se instala una aplicación, se instala para un sistema operativo y una base de datos determinada. A lo largo del tiempo estás tecnologías cambian por parte del fabricante o por parte del cliente. Este cambio tecnológico no entra en el mantenimiento estándar.

Este mantenimiento consiste en la disponibilidad de descarga de nuevas actualizaciones específicas del software suministrado, con objeto de garantizar la compatibilidad ante el cliente.

Este mantenimiento permite que la aplicación siga funcionando correctamente cuando surjan en el mercado nuevas versiones de los productos que conforman el entorno tecnológico requerido por la aplicación.

Las nuevas versiones específicas de la aplicación, incorporan nuevas funcionalidades que optimizan y mejoran el uso de la aplicación con el nuevo entorno tecnológico del cliente, con objeto de aprovechar, en la medida de lo posible, las prestaciones que ofrecen las nuevas versiones del entorno tecnológico.

Este mantenimiento se suministra a:

- Sistemas operativos.
- Herramientas.
- Bases de datos.
- Servidores.

Mantenimiento adaptativo.

Consiste en la modificación de un programa debido a cambios en el entorno (hardware o software), en el cual se ejecuta. Estos cambios pueden afectar al sistema operativo, a la arquitectura física del sistema informático o al entorno de desarrollo del software (incorporación de nuevos elementos o herramientas). La envergadura del cambio necesario puede ser muy diferente, desde un pequeño retoque en la estructura de un módulo hasta tener que reescribir prácticamente todo el programa para su ejecución en un ambiente distribuido en una red.

Los cambios en el entorno del software pueden ser de dos clases:

- En el entorno de los datos.
- En el entorno de los procesos.

4.5 Generación de reportes

La información que se obtiene de los CDRs se puede revisar realizando consultas en la base de datos ya que el sistema no tiene un front-end y los archivos se

envían al área de facturación para su procesamiento, en estos reportes se puede observar muchos campos que nos sirven para tener un control sobre el tipo de llamada, la forma de pago, duración de la llamada etc. A continuación se describen los campos más comunes al realizar una consulta.

- Tipo: Muestra si la llamada corresponde a la propia red o si es de interconexión
- Dato: Informa si se enviaron voz, datos o multimedia
- Número que llama: Número que realiza la llamada
- Número al que se llama: Número que recibe la llamada
- Zona: Lugar del país donde se realizó la llamada(Zona 1 norte, Zona 2 centro y Zona 3 sur)
- Forma de pago: Indica si fue en renta mensual o pospago

Se presenta los datos de una consulta.

TIPO	DATO	NUM. QUE LLAMA	NUM. AL QUE SE LLAMA	ZONA	FORMA DE PAGO
Outgoing	Voice	5585089671	7444851920	1	POSTPAID
Outgoing	Voice	5550300001	4777914431	2	POSTPAID
Outgoing	Voice	5550300001	4499749524	2	POSTPAID
Outgoing	Voice	5550300001	4491193288	2	POSTPAID
Outgoing	Voice	5550300001	4499155103	2	POSTPAID
Outgoing	Voice	5550300001	4499121578	2	POSTPAID
Outgoing	Voice	5550300001	4499145757	2	MONTHPAID
Incoming	Voice	5550300001	4777914431	2	POSTPAID
Outgoing	Voice	5550300001	4771006921	2	POSTPAID
Outgoing	MMS	5550300001	4771006921	2	POSTPAID
Outgoing	Voice	5550300001	4771006921	2	POSTPAID
Outgoing	Voice	5585077140	3334901761	3	POSTPAID
Outgoing	Voice	5519750121	3331900576	3	POSTPAID
Incoming	Voice	3331900576	5519750121	3	POSTPAID

TIPO	DATO	NUM. QUE LLAMA	NUM. AL QUE SE LLAMA	ZONA	FORMA DE PAGO
Outgoing	Voice	5519750121	3331900576	3	POSTPAID
Outgoing	Voice	5585077140	3338462145	3	POSTPAID
Outgoing	Voice	5585080288	3311724845	3	POSTPAID
Outgoing	Voice	5585077140	3336721642	3	POSTPAID
Outgoing	Voice	5585080288	3336788888	3	POSTPAID
Outgoing	Voice	5519750121	3331441008	3	POSTPAID
Outgoing	Voice	5514584531	3336027016	3	POSTPAID
Outgoing	Voice	5585077134	3333942593	3	POSTPAID
Outgoing	Voice	5591429718	3573840998	3	POSTPAID
Outgoing	Voice	5550300001	3336753319	3	POSTPAID
Outgoing	Voice	5585077134	3338121867	3	MONTHPAID
Outgoing	Voice	5585077124	3338256401	3	MONTHPAID
Incoming	Voice	5514584531	3334815141	3	POSTPAID
Outgoing	Voice	5585077134	3338121867	3	POSTPAID
Outgoing	Voice	5585080288	3919161846	3	POSTPAID
Outgoing	Voice	5550300001	3337321956	3	POSTPAID
Outgoing	Voice	5519750121	3757584304	3	POSTPAID
Outgoing	Voice	5514584531	3477884834	3	POSTPAID
Outgoing	Voice	5585077124	3336984114	3	POSTPAID
Outgoing	Voice	5519750121	3757588009	3	POSTPAID
Incoming	Voice	5585077134	3336458439	3	POSTPAID
Outgoing	Voice	5519750121	3757588009	3	POSTPAID
Outgoing	Voice	5550300001	3336044533	3	POSTPAID
Outgoing	Voice	5519750121	3757584304	3	POSTPAID
Outgoing	Voice	5585080288	9535410599	3	POSTPAID
Outgoing	Voice	5585077134	3335863589	3	MONTHPAID
Incoming	Voice	5585080288	3333427811	3	POSTPAID
Outgoing	Voice	5519750121	3757584304	3	POSTPAID

TIPO	DATO	NUM. QUE LLAMA	NUM. AL QUE SE LLAMA	ZONA	FORMA DE PAGO
Outgoing	Voice	5591429718	3333662767	3	POSTPAID
Outgoing	Voice	5519750121	3336007707	3	POSTPAID
Outgoing	Voice	5550300001	3310574657	3	POSTPAID
Outgoing	Voice	5585077134	3336075446	3	POSTPAID
Outgoing	Voice	3338394107	3311442184	3	MONTHPAID
Outgoing	Voice	5591429718	3331829030	3	MONTHPAID
Incoming	Voice	5519750121	3336325139	3	MONTHPAID
Outgoing	MMS	5591429718	3335700198	3	POSTPAID

Ahora se muestra el reporte que genera el área de facturación sobre las llamadas realizadas en el mes de enero de telefonía celular.

Facturación enero:

- No. de abonados: 17728

Llamadas nacionales e internacionales:	802296
Llamadas locales:	2723325
Llamadas prepago:	2053
Llamadas entrantes:	1215536
Llamadas salientes de otros operadores:	159464
Total de llamadas procesadas:	4902674

- Tamaño de la base de datos generada: 415 MB
- Tiempo aproximado del proceso de facturación: 20 minutos