República de Cuba

Universidad de las Ciencias Informáticas

Facultad 2

Sistema de Gestión de Emergencias de Seguridad Ciudadana (171)

"Diseño y Administración de Base de Datos"

Trabajo de Diploma

Presentado para optar por el título de
Ingeniero en Ciencias Informáticas

Autor: Adonis Rodríguez Fernández.
Daniel Ernesto Vargas Allegue.

Tutor: Ing. Yordanis Tornés Medina.

"Año 49 de la Revolución"
Ciudad de la Habana, Cuba.
5 de Julio de 2007.

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas para que haga el uso que estime pertinente con el mismo.

Para que así conste firmo la presente a los _5_ días del mes de _julio_ del año _2007_.

Adonis Rodríguez Fernández	**Daniel Ernesto Vargas Allegue**	**Ing. Yordanis Tornes Medina**
_____	_____	_____
Firma del Autor	Firma del Autor	Firma del Tutor

OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA

Título: **Sistema de Gestión de Emergencias de Seguridad Ciudadana (171).**

Diseño y Administración de Base de Datos.

Autores: **Adonis Rodríguez Fernández**

Daniel Ernesto Vargas Allegue

Los estudiantes que hoy discuten este Trabajo de Diploma son ejemplo a seguir para las futuras generaciones de Ingenieros de esta Universidad. Inmersos en disímiles de tareas, las que les correspondían en función de su área de trabajo y las que cada día se les fueron asignando por la confianza que inspiran, seriedad y responsabilidad con que acuden donde se les necesita; las convocadas por la Faculta, la Universidad, las distintas organizaciones de masas y las convocadas por la gloriosa dirección de esta Revolución; simultaneando todas estas tareas fueron capaces de diseñar la estructura de datos que hoy sirve de base al Sistema de Gestión de Emergencias de Seguridad Ciudadana (171), sistema que contribuye eficiente y eficazmente a mejorar la seguridad ciudadana en la República Bolivariana de Venezuela. Desarrollaron esta investigación con gran independencia y creatividad; implementando y creando soluciones eficaces a los problemas que se presentaban cada día y que afectaban el rendimiento y vitalidad del Sistema.

Estos Diplomantes que desde ya podemos considerar “Colegas”, desde que comenzaron a trabajar en el proyecto demostraron tener condiciones, reafirmadas cada día mas, para ejercer como Ingenieros en Ciencias Informáticas.

Por todo lo anteriormente expresado considero que los Diplomantes están aptos para ejercer como Ingenieros en Ciencias Informáticas; y propongo que se le otorgue al Trabajo de Diploma la calificación de **5 puntos**. Considero además que el trabajo obtenido posee valor para ser publicado una vez se culmine su diseño y configuración y esté en producción, hasta tanto debe quedar archivado y consultado solamente por el equipo de desarrollo.

Ing. Yordanis Tornés Medina

5 de Julio de 2007

Dedicatoria.

…A nuestros Padres y familiares.

Agradecimientos.

Ante todo deseamos expresar nuestro más sincero agradecimiento a la Revolución Cubana por darnos la oportunidad de convertirnos en hombres de ciencia y realizar nuestros sueños.

A la Universidad de las Ciencias Informáticas, por abrirnos sus puertas y darnos la posibilidad de crecer como informáticos comprometidos, intransigentes y revolucionarios.

A nuestro tutor Yordanis, por su entrega y dedicación para que este trabajo tuviera la mejor calidad posible.

A los profesores del proyecto, Adrian, Yisel, Wilfredo, por brindarnos sus conocimientos y paciencia en muchos momentos.

A nuestros amigos inseparables Alejandro y Roig que siempre han estado a nuestro lado en esta larga batalla.

A todos nuestros compañeros de estudios que nos han acompañado a lo largo de estos cinco años y que siempre serán recordados.

En fin, a todos los que de una forma u otra han influido en la culminación exitosa de este trabajo.

Resumen.

La seguridad ciudadana es uno de los principales deberes desde el punto de vista social que todo gobierno debe garantizar, una forma de lograr tal propósito es hacer que las demandas de emergencias efectuadas por la población sean atendidas de manera rápida y efectiva por los órganos de seguridad ciudadana.

La actualidad de la República Bolivariana de Venezuela nos arroja que aún existen problemas con la atención de las emergencias de la población, existen varios números para la atención de emergencias, los centros que existen no automatizan todas las actividades que pueden ser automatizadas, entre otros problemas.

Para una mejor protección de la población ante hechos delictivos, situaciones de emergencia y desastres naturales y para dar respuesta eficiente a cualquier irregularidad, el Ministerio del Poder Popular para Relaciones Interiores y Justicia de Venezuela, promueve la formulación y puesta en marcha del ***Centro de Gestión de Emergencias de Seguridad Ciudadana (171).***

Como parte del desarrollo del Sistema de Gestión de Emergencias de Seguridad Ciudadana se crea este trabajo el cual tiene como objetivo principal analizar y describir los diseños de base de datos propuestos para algunos de los módulos del Sistema de Gestión de Emergencia de Seguridad Ciudadana, así como la definición de algunos elementos de administración y configuración del gestor de base de datos del Sistema (Oracle) en pos de obtener un rendimiento óptimo del mismo, sin sacrificar la disponibilidad e integridad de los datos.

Este trabajo dará paso a lograr un diseño de base de datos que soporte las necesidades de almacenamiento de información del sistema antes mencionado. Brindará algunas de las vías para lograr un funcionamiento óptimo del servidor Oracle a través de elementos de administración y configuración, proporcionando rendimiento y disponibilidad.

Índice.

Introducción.

Cada estado tiene la responsabilidad de la seguridad de sus habitantes. *"Todo individuo tiene derecho a la vida, a la libertad y a la seguridad de su persona"[1], "derecho a la protección por parte del estado, a través de los órganos de seguridad ciudadana, regulados por ley, frente a situaciones que constituyan amenazas, vulnerabilidad o riesgos para la integridad física de las personas, sus propiedades, el disfrute de sus derechos y el cumplimiento de sus deberes"[2]*.

Para una mejor protección de la población ante hechos delictivos, situaciones de emergencia y desastres naturales y para dar respuesta eficiente a cualquier irregularidad, el Ministerio del Poder Popular para Relaciones Interiores y Justicia atendiendo a su misión institucional de garantizar la seguridad ciudadana, promueve la formulación y puesta en marcha de los **Centros de Gestión de Emergencias de Seguridad Ciudadana (171)**, en lo adelante **Centro 171**, partiendo de la premisa que la seguridad ciudadana es una condición necesaria para el desarrollo humano.

Hoy la República Bolivariana de Venezuela cuenta con un número reducido, en todo el territorio nacional, de Centros de Gestión de Emergencias. Estos se encargan de coordinar los diferentes órganos de seguridad ciudadana para atender situaciones que constituyan amenazas para la seguridad de los ciudadanos.

Por otra parte actualmente en Venezuela existen varios números telefónicos para reportar emergencias, ofrecidos por empresas privadas de telecomunicaciones, gobernaciones y alcaldías. Algunos de estos centros funcionan las 24 horas de los 365 días del año, la mayoría sólo responde en horas de oficina, ninguno está interconectado entre sí y actúan como centros independientes, lo que demora la respuesta y repercute en el bienestar y seguridad de la población.

Aunque existen normativas de coordinación entre los órganos de seguridad ciudadana, han resultado insuficientes para la protección de la población ante hechos delictivos, situaciones de emergencia y desastres naturales.

El estado del Táchira en la República Bolivariana de Venezuela, al igual que muchos otros, en la actualidad cuenta con un **Centro 171**, mediante el cual da solución a un número de solicitudes de emergencia que reporta la población, se ha podido comprobar que el software de gestión con el que hoy cuentan no tiene un rendimiento óptimo y no automatiza todas las tareas que se realizan para así disminuir los tiempo de despacho de las solicitudes. Además de que no cuentan con un control automatizado de las tareas, del

personal y de los recursos del centro y de los que los órganos de seguridad ciudadana le asignan para sus tareas de coordinación.

El **Centro 171,** de manera general pretende resolver los siguientes problemas y necesidades de la población de la República Bolivariana de Venezuela:

- ✓ Fortalecer las acciones de coordinación entre los órganos de seguridad ciudadana y la unificación de criterios en la toma de decisiones.
- ✓ Disminuir el tiempo de respuesta a las demandas de emergencias formuladas por la población.
- ✓ Mantener un registro, en tiempo real e histórico, de información de hechos delictivos, emergencias y desastres.
- ✓ Medir la eficiencia y eficacia de los programas sociales en la disminución del delito dentro de las comunidades, y el uso de la información para el desarrollo de estrategias.

Un **Centro 171**, prioriza los siguientes servicios y funcionalidades:

- ✓ Capacidad de recepción de la información y denuncias de la población u otras fuentes.
- ✓ Despachar la atención de las emergencias hacia los centros de atención directa, y garantizar su seguimiento hasta lograr la asistencia solicitada.
- ✓ Mantener un registro de información que permita a los órganos de seguridad ciudadana, consultar y realizar análisis de la situación delictiva y de emergencias con carácter histórico.
- ✓ Posibilitar el análisis geográfico de la actividad delictiva y de emergencias, así como la localización en tiempo real de los recursos en servicio.
- ✓ Disponer de un sistema de radiocomunicaciones TETRA que facilite la intercomunicación entre los diferentes cuerpos policiales de la región y el Centro 171.
- ✓ Acceder desde el centro a información de interés disponible en otros organismos.
- ✓ Gestionar los recursos con que cuenta el centro para la solución de emergencias.
- ✓ Crear un sistema de gestión de emergencias que cumpla con todas las funcionalidades, hasta aquí expuestas, de la mejor forma y en el menor tiempo posible.

El desarrollo de un Sistema de Gestión de Emergencias de Seguridad Ciudadana (171) (SIGESC) incluye la implementación de operaciones que permitan desarrollar las actividades que anteriormente se enumeran de una forma óptima.

Debido a la diversidad de actividades desarrolladas en el Centro 171 para la atención de las emergencias**,** se realizó una división del sistema informático en varios módulos para así efectuar una rápida implementación del software.

Los módulos que se implementarán son los siguientes:

- ✓ **Módulo de Recepción de Llamadas (Operador).**
- ✓ **Módulo de Despacho (Despachador)**
- ✓ Módulo de Mapificación.
- ✓ **Módulo de Supervisión de Operadores.**
- ✓ **Módulo de Supervisión de Despachadores.**
- ✓ Módulo de Supervisión General.
- ✓ Módulo de Estadísticas.
- ✓ Módulo de AVL.
- ✓ **Módulo de Administración y Control de Recursos.**
- ✓ Módulo de Configuración de Operaciones.
- ✓ Módulo de Administración.

De estos módulos se tratarán en este trabajo solamente los que se resaltan en negritas, con el objetivo de poder profundizar en los detalles del diseño de la base de datos de los mismos, algo que no es posible en todos los módulos por estar en una etapa temprana de desarrollo del sistema.

Para el cumplimiento satisfactorio de las funcionalidades que en la primera etapa se quieren priorizar y en todo el desarrollo del sistema, es de vital importancia la base de datos. Esta cumple importantes misiones dentro del sistema tales como:

- ✓ Funcionar como centro de almacenamiento de datos.
- ✓ Establecer un punto de intercambio de información entre los diferentes módulos.
- ✓ Servir como fuente de generación de reportes estadísticos.
- ✓ Mantener la disponibilidad e integridad de la información.
- ✓ Servir como fuente de datos para otros sistemas del Ministerio como: CTAISC.
- ✓ Garantizar la confidencialidad y seguridad de la información manipulada.

En el servidor se almacenarán todos los datos de configuración de las aplicaciones y de los usuarios, necesaria para que el sistema funcione correctamente. Además se hospedarán los datos históricos que se generan como resultado de las solicitudes realizadas por la población.

De esta forma las aplicaciones buscarán información en el servidor de base de datos la mayor parte del tiempo para su funcionamiento, datos que asincrónicamente han sido insertados, actualizados o eliminados por estas mismas aplicaciones, estableciendo así un medio de comunicación entre ellas.

De la base de datos se generaran importantes reportes útiles para la solución de problemas, para prevenir sucesos y emitir criterios que permitan dar una estadística de cómo están determinados índices como pueden ser, qué meses del año tienen los números más elevados de emergencias.

En relación con las políticas de seguridad implementadas dentro del sistema se integrarán los mecanismos de seguridad del servidor Oracle como parte de la solución de seguridad conjunta. De forma que exista seguridad desde las aplicaciones y desde el servidor. Así se aumentan los niveles de confidencialidad y disponibilidad ya que accederán a la información solo los usuarios autorizados, desde los lugares autorizados y en los momentos establecidos.

Como resultado de lo analizado se resume que nuestro **problema científico** se centra en ¿cómo lograr el diseño y administración de servidores de base de datos para el SIGESC?

Nuestro **objeto de investigación** versa sobre el "diseño y administración de bases de datos para el Sistema de Gestión de Emergencias de Seguridad Ciudadana (171)", determinando el **campo de acción** en el "diseño, administración y configuración de base de datos para el SIGESC en la República Bolivariana de Venezuela".

Las preguntas de investigación que se plantean son las siguientes:

- ✓ ¿Cómo diseñar una base de datos sin inconsistencia y redundancia en la información?
- ✓ ¿Cómo configurar los parámetros del servidor para lograr mejor rendimiento y disponibilidad de la información?
- ✓ ¿Qué tareas administrativas se deben implementar para garantizar seguridad, confidencialidad de la información y un buen funcionamiento del servidor?

Dando respuesta a estas preguntas se puede cumplir con el **objetivo general** de diseñar y administrar la base de datos del Sistema de Gestión de Emergencias de Seguridad Ciudadana (171) para la República Bolivariana de Venezuela, y los **objetivos específicos** son diseñar la base de datos, administrar el servidor de base de datos, configurar el servidor de base de datos para mayor rendimiento.

Estos elementos nos conllevan a generar un grupo de tareas de investigación como las que a continuación relacionamos:

1. Estudiar y aplicar la metodología de análisis y diseño de base de datos.

2. Estudiar temas de administración y configuración de Oracle.
3. Estudiar herramientas que faciliten el trabajo de administración.
4. Diseñar la base de datos.
5. Configurar el servidor.
6. Administrar el servidor.

Con estas se propone dar respuesta a las interrogantes planteadas y solucionar los problemas antes comentados.

El documento está estructurado en: resumen, introducción y desarrollo, el cual está compuesto por 3 capítulos:

- ✓ Capítulo I denominado "**Fundamentación Teórica**", se incluyen todos los aspectos teóricos que soportan este proyecto y se realiza un estudio del estado del arte de diferentes sistemas de gestión de emergencia de seguridad ciudadana que existen en el mundo.
- ✓ Capítulo II denominado "**Diseño de la Base de Datos**" donde planteamos la problemática de cada uno de los módulos, el diseño propuesto y la justificación del mismo.
- ✓ Capítulo III denominado "**Administración del Servidor**", donde se analizarán los parámetros de configuración, la seguridad del servidor, las reglas básicas de backup y recovery y clasificación del servidor en cuanto a la disponibilidad de la información.

Se concluye el trabajo con un resumen del resultado obtenido durante el transcurso del desarrollo del mismo, las recomendaciones y los aspectos a profundizar. Se incluye un glosario de términos y la bibliografía que tuvimos como referencia.

Capítulo I: Fundamentación Teórica.

1.1 Introducción.

En este capítulo se presentan un grupo de conceptos teóricos a los que se hará referencia en el resto del trabajo. Se abordan aspectos relacionados con el uso de las nuevas tecnologías de la informática y las comunicaciones. Se tocan conceptos importantes dentro del mundo de las bases de datos y temas relacionados propiamente con el servidor Oracle. Otros temas más relacionados con el SIGESC de forma general como son seguridad ciudadana y centros de gestión de emergencias.

Se presenta además la metodología a utilizar para el análisis, diseño e implementación de la base de datos y las características más importantes que influyeron en la selección de Oracle como gestor de Base de Datos.

1.2 Seguridad Ciudadana.

Debe entenderse como Seguridad Ciudadana el grado de respeto que se otorga al conjunto de derechos de los ciudadanos, no solo por parte del Estado, sino también de parte de las personas e instituciones públicas y privadas, que tienen que garantizar el bienestar de todos los componentes de las estructuras a las cuales están vinculadas, a las cuales prestan servicios o de las que depende su seguridad.

1.3 Centros de Gestión de Emergencias.

Los centros de gestión de emergencias son un servicio que se brinda a la población con el objetivo de ofrecer soluciones efectivas a las situaciones problemáticas de la sociedad, y con esto brindar mayor confianza en la seguridad ciudadana de cada individuo. Para lograr tales expectativas estos centros deben ser un ente integrador de los organismos de seguridad y emergencias, encargado de recibir las llamadas de emergencia de la ciudadanía las 24 horas del día durante los 365 días del año, manteniendo un servicio de comunicaciones que permita garantizar la adecuada supervisión y capacidad de respuesta de los organismos.

En la actualidad esto se realiza con la ayuda de sistemas automatizados que se encargan de realizar la mayoría de las funciones de manera inmediata, estos se les ha llamado Sistemas de Gestión de Emergencias de Seguridad Ciudadana.

Estos sistemas cuentan con diferentes subsistemas de cómputo, telefonía, radio e información operativa, ofreciendo a la población beneficios como la disminución del tiempo de respuesta a las demandas de emergencias formuladas, detectar la localización de las llamadas e informar a la(s) institución(es) del estado que corresponde atender el incidente. También visualizan toda la región que se atiende, los hechos que ocurren, los lugares donde están ocurriendo las emergencias solicitadas, cuáles móviles están en servicio, entre otras. Así se brinda un mejor servicio y contribuye de una forma u otra a que los ejecutivos tomen mejores decisiones para el bienestar y la seguridad del pueblo.

1.3.1 Centros de Gestión de Emergencias en Venezuela.

Los centros de gestión de emergencias son un servicio que se brinda a la población con el objetivo de ofrecer soluciones efectivas a las situaciones problemáticas de la sociedad. En la República Bolivariana de Venezuela existen hoy varios centros de este tipo que poseen características que los hacen ineficientes. Existen un grupo de estos centros que brindan servicios por empresas y que solamente laboran en horarios de oficina, en estos centros no se coordinan correctamente a todos los órganos de seguridad ciudadana sino que queda a consideración de ellos informar del hecho a la policía u otros órganos que serían los encargados de solucionar los problemas en las calles. Esto disminuye el impacto que tiene la atención por parte de los diferentes órganos ante las situaciones que se presentan en el día a día de los ciudadanos.

Existen otros centros que funcionan correctamente, son de prestación de servicio 24x7 y los 365 días del año, coordinan correctamente a los órganos de seguridad ciudadana y dan un seguimiento de la solicitud desde que la reciben hasta que es solucionada y almacenan toda la información relacionada.

A continuación se muestran el vínculo a la página principal de algunos de estos sistemas:

- ✓ http://www.monagas.gov.ve/171.asp
- ✓ http://www.e-171.gob.ve/
- ✓ http://www.171tachira.gob.ve/
- ✓ http://www.sel171.gob.ve/

- ✓ http://www.171aragua.org.ve/

1.4 Sistemas de Gestión de Seguridad Ciudadana en Venezuela y el mundo.

1.4.1 Centro Coordinador de Emergencias y Seguridad 1-1-2, "Canarias"

El Centro Coordinador de Emergencias y Seguridad 1-1-2 de Canarias comenzó a funcionar con su anterior denominación -Teléfono Único de Urgencias 1-1-2- el 30 de mayo de 1998, con el fin de garantizar una respuesta rápida y eficaz a todas las llamadas de emergencia que se producen en las Islas. A este sistema pueden acceder todos los ciudadanos o visitantes que requieran, en caso de urgente necesidad, la asistencia de los dispositivos públicos competentes en materia de atención de urgencias sanitarias, seguridad ciudadana, extinción de incendios, salvamento y rescate.

Para ello, basta con llamar al 1-1-2, un número de teléfono gratuito, de cobertura regional, que funciona las 24 horas del día y que es atendido por personal especializado en los idiomas inglés, alemán y español.

Con la creación de este servicio, el Gobierno de Canarias se integra en el Sistema Europeo de Atención de Urgencias y Emergencias, cumpliendo así la Decisión de la Unión Europea 91/396, por la que se recomendaba a los estados miembros la puesta en marcha de un único número de teléfono, el 1-1-2, para acceder a todos los servicios de urgencia de cada uno de los países comunitarios.

En Canarias, la titularidad de este servicio recae en la Dirección General de Seguridad y Emergencias, centro directivo de la Consejería de Presidencia y Justicia. La gestión del mismo ha sido encomendada a la empresa pública Gestión de Servicios para la Salud y Seguridad en Canarias. [6]

1.4.2 Centro Automático de Despacho – CAD 123 "Santiago de Cali".

En una ciudad como Santiago de Cali se registran todas las expresiones de conflicto que pueden afectar una gran urbe, principalmente las relacionadas con problemas de convivencia, seguridad ciudadana y atención de emergencias .Por eso la Administración Municipal a través de la Secretaría de Gobierno, Convivencia y Seguridad Ciudadana, ha tomado la iniciativa de implementar El Centro Automático de Despacho (C.A.D. Cali). El Centro Automático de Despacho (CAD) fue concebido para proporcionar atención y respuesta a llamadas de emergencias, basado en el uso del Número Telefónico Único 123. Surge ante la necesidad de contribuir a la disminución de los niveles de inseguridad y violencia de la

ciudad de Santiago de Cali, fortaleciendo la eficiencia de las acciones de atención, tendientes a prevenir, contrarrestar y controlar factores identificados como asociados con hechos delictivos de violencia, accidentes y desastres.

Este número único que funciona bajo el código 1-2-3, está en la capacidad de proveer a todos los ciudadanos la posibilidad de comunicarse de una manera rápida y efectiva con los organismos que lo componen entre los cuales se encuentran la Policía Metropolitana, Transito Municipal, Bomberos, Salud, Comité Local para la Prevención y Atención de Desastres (CLOPAD), Cuerpo Técnico de Investigaciones (CTI), Departamento Administrativo de Seguridad (DAS) y la Policía Militar.

El C.A.D. permite y facilita coordinar de una manera óptima todos los recursos de atención de emergencia del área metropolitana de la ciudad de Santiago de Cali, asegurando que sus ciudadanos contarán con una atención confiable, oportuna y segura los 365 días del año.

Teniendo en cuenta que las personas más susceptibles de presentar emergencias y requerir ayuda urgente, son niños y ancianos, se ha concebido el 1-2-3 como la marcación telefónica más sencilla de realizar y de fácil recordación. [7]

1.4.3 Gobernación del Estado Bolívar.

El Centro 171 de Bolívar es uno de los centros que realiza el seguimiento de la solicitud de emergencia desde su surgimiento hasta que es atendido el ciudadano que solicito la ayuda. Entre sus objetivos se encuentran:

Atender las emergencias de la ciudad, para garantizar la vida y la salud del ciudadano, por ello trabajan durante las 24 horas del día y los 365 días del año sin interrupción, en todo el Estado Bolívar y parte del estado Monagas. Emergencias Bolívar 1-7-1, es un organismo Integrador de todos los cuerpos de Seguridad del Estado Bolívar, como consecuencia de ello la sala de Operaciones atiende un promedio de 5.000 llamadas diarias

Misión.

- ✓ Coordinar los servicios de atención de auxilio y emergencias, dispuestos por el Estado a fin de garantizar la vida, salud y atención del ciudadano.
- ✓ Fomentar, estimular y desarrollar la idoneidad, eficiencia y efectividad de los servicios de atención, auxilio y emergencia del Estado, mediante la implementación de programas de equipamiento, actualización y entrenamiento.

- ✓ Coordinar con los organismos de seguridad pública nacional y de los Estados vecinos, a fin de unificar criterios de información y comunicación.
- ✓ Lograr la autogestión para aminorar las cargas del Estado.

Visión

- ✓ Coordinar con los gobiernos municipales y la sociedad civil organizada en materia de seguridad y emergencias.
- ✓ Establecer convenios institucionales, para lograr alianzas estratégicas en beneficio de la Seguridad y Emergencia del Estado Bolívar.
- ✓ Apoyar las iniciativas y proyectos que se propongan a través del Gobierno Nacional.
- ✓ Supervisar las comunicaciones policiales y de emergencias del Estado Bolívar. [8]

1.4.4 Servicio 066 Coahuila, México.

La misión de este centro es ser un equipo que a través de acciones y estrategias, está comprometido en preservar la seguridad, el orden y la paz social.

Visión:

- ✓ Ser una Secretaría confiable que garantice un Estado Seguro.

Este centro tiene como objetivos:

- ✓ Proporcionarle a la ciudadanía una vía para la atención de sus emergencias reportadas. Dotar a todos los municipios de la infraestructura y herramientas adecuadas para la atención de emergencias reportadas por la ciudadanía.
- ✓ Darle a varios centros de atención, la posibilidad de comunicarse entre ellos.
- ✓ Contar con herramientas y procesos de monitoreo, control, administración y seguimiento que permitan a los centros poder realizar su trabajo de forma más sencilla y eficaz.
- ✓ Proporcionar a las instancias receptoras de los reportes para su atención, tales como Cruz Roja, Bomberos, Protección Civil, entre otros, de una herramienta de coordinación y atención rápida a los requerimientos recibidos por parte de la ciudadanía.

Metas:

- ✓ Que la ciudadanía cuente con un número telefónico en el que confíe para pedir ayuda.
- ✓ Que todos los municipios cuenten con la infraestructura necesaria para cubrir todas y cada una de las emergencias de la entidad.

- ✓ Ser un vehículo que proporcione de manera oportuna y eficaz, información para una mejor toma de decisiones a las corporaciones.[9]

Estos sistemas de Gestión de Emergencias de Seguridad Ciudadana tienen que ser soportados por una base de datos que es la encargada de almacenar la información que es generada como resultado de las operaciones que se realizan en el centro. Esta base de datos brinda seguridad a los datos, confidencialidad, integridad y garantiza disponibilidad de la información.

1.5 Base de Datos.

En los sistemas informáticos hoy es muy importante la presencia de bases de datos que almacenen la información.

Una base de datos es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su uso posterior. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta [3].

Estos sistemas se clasifican según la forma en que se diseña la estructura de almacenamiento, pueden ser jerárquicas, de red, relacional, orientada a objetos, entre otras clasificaciones.

1.5.1 Base de Datos Jerárquica.

Éstas son bases de datos que, como su nombre indica, almacenan su información en una estructura jerárquica. En este modelo los datos se organizan en una forma similar a un árbol (visto al revés), en donde un nodo padre de información puede tener varios hijos. El nodo que no tiene padres es llamado raíz, y a los nodos que no tienen hijos se los conoce como hojas.

Las bases de datos jerárquicas son especialmente útiles en el caso de aplicaciones que manejan un gran volumen de información y datos muy compartidos permitiendo crear estructuras estables y de gran rendimiento.

Una de las principales limitaciones de este modelo es su incapacidad de representar eficientemente la redundancia de datos [3].

1.5.2 Base de Datos de Red.

Éste es un modelo ligeramente distinto del jerárquico su diferencia fundamental es la modificación del concepto de nodo: se permite que un mismo nodo tenga varios padres (posibilidad no permitida en el modelo jerárquico).

Fue una gran mejora con respecto al modelo jerárquico, ya que ofrecía una solución eficiente al problema de redundancia de datos; pero, aún así, la dificultad que significa administrar la información en una base de datos de red ha significado que sea un modelo utilizado en su mayoría por programadores más que por usuarios finales [3].

1.5.3 Base de Datos Relacional.

Una base de datos relacional es una base de datos basada en un modelo relacional. Estrictamente hablando el término se refiere a una colección específica de datos, pero a menudo es usado como sinónimo del software utilizado para gestionar esa colección de datos. Ese software se conoce como sistema gestor de base de datos relacional o RDBMS (Relational Database Management System) [3].

Éste es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Tras ser postulados sus fundamentos en 1970 por Edgar Frank Codd, de los laboratorios IBM en San José (California), no tardó en consolidarse como un nuevo paradigma en los modelos de base de datos. Su idea fundamental es el uso de "relaciones". Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados "tuplas". Pese a que ésta es la teoría de las bases de datos relacionales creadas por Edgar Frank Codd, la mayoría de las veces se conceptualiza de una manera más fácil de imaginar. Esto es pensando en cada relación como si fuese una tabla que está compuesta por registros (las filas de una tabla), que representarían las tuplas, y campos (las columnas de una tabla).

En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia. Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario esporádico de la base de datos. La información puede ser recuperada o almacenada mediante "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información.

El lenguaje más habitual para construir las consultas a bases de datos relacionales es SQL, Structured Query Language o Lenguaje Estructurado de Consultas, un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales.

Durante su diseño, una base de datos relacional pasa por un proceso al que se le conoce como normalización de una base de datos [3].

1.5.4 Base de Datos Orientada a Objetos.

Este modelo, bastante reciente, y propio de los modelos informáticos orientados a objetos, trata de almacenar en la base de datos los *objetos* completos (estado y comportamiento).

Una base de datos orientada a objetos es una base de datos que incorpora todos los conceptos importantes del paradigma de objetos y que los implementa en sus operaciones, estos conceptos son los siguientes:

- ✓ Encapsulación - Propiedad que permite ocultar la información al resto de los objetos, impidiendo así accesos incorrectos o conflictos.
- ✓ Herencia - Propiedad a través de la cual los objetos heredan comportamiento dentro de una jerarquía de clases.
- ✓ Polimorfismo - Propiedad de una operación mediante la cual puede ser aplicada a distintos tipos de objetos [10].

En bases de datos orientadas a objetos, los usuarios pueden definir operaciones sobre los datos como parte de la definición de la base de datos. Una operación (llamada función) se especifica en dos partes. La interfaz (o signatura) de una operación incluye el nombre de la operación y los tipos de datos de sus argumentos (o parámetros). La implementación (o método) de la operación se especifica separadamente y puede modificarse sin afectar la interfaz. Los programas de aplicación de los usuarios pueden operar sobre los datos invocando a dichas operaciones a través de sus nombres y argumentos, sea cual sea la forma en la que se han implementado. Esto podría denominarse independencia entre programas y operaciones.

Se está trabajando en SQL3 [11], que es el estándar de SQL92 [3] ampliado, que soportará los nuevos conceptos orientados a objetos y mantendría compatibilidad con SQL92.

1.5.5 Bases de Datos Documentales.

Permiten la indexación a texto completo, y en líneas generales realizar búsquedas más potentes. Tesaurus es un sistema de índices optimizado para este tipo de bases de datos.

1.5.6 Bases de Datos Deductivas.

Un sistema de **base de datos deductivos**, es un sistema de base de datos pero con la diferencia de que permite hacer deducciones a través de inferencias. Se basa principalmente en reglas y hechos que son almacenados en la base de datos. También las bases de datos deductivas son llamadas base de datos lógica, a raíz de que se basan en lógica matemática.

1.5.7 Bases de Datos Distribuidas.

La base de datos está almacenada en varias computadoras conectadas en red. Surgen debido a la existencia física de organismos descentralizados. Esto les da la capacidad de unir las bases de datos de cada localidad y acceder así a distintas universidades y sucursales de tiendas [3].

1.6 Metodología de Diseño de Base de Datos.

El diseño de una base de datos es un proceso complejo que abarca decisiones a muy distintos niveles. La complejidad se controla mejor si se descompone el problema en subproblemas y se resuelve cada uno de estos subproblemas de forma independientemente, utilizando técnicas específicas. Así, el diseño de una base de datos se descompone en diseño conceptual, diseño lógico y diseño físico.

1.6.1 Diseño Conceptual.

En esta etapa se debe construir un esquema de la información que se usa en la empresa, independientemente de cualquier consideración física. A este esquema se le denomina *esquema conceptual*. Al construir el esquema, los diseñadores descubren la semántica (significado) de los datos de la empresa: encuentran entidades, atributos y relaciones. El objetivo es comprender:

- ✓ La perspectiva que cada usuario tiene de los datos.
- ✓ La naturaleza de los datos, independientemente de su representación física.

- ✓ El uso de los datos a través de las áreas de aplicación.

El esquema conceptual se puede utilizar para que el diseñador transmita a la empresa lo que ha entendido sobre la información que ésta maneja. Para ello, ambas partes deben estar familiarizadas con la notación utilizada en el esquema. La más popular es la notación del modelo entidad-relación y la que proponemos se utilice en el diseño de este sistema.

El esquema conceptual se construye utilizando la información que se encuentra en la especificación de los requisitos de usuario. El diseño conceptual es completamente independiente de los aspectos de implementación, como puede ser el SGBD que se vaya a usar, los programas de aplicación, los lenguajes de programación, el hardware disponible o cualquier otra consideración física. Durante todo el proceso de desarrollo del esquema conceptual éste se prueba y se valida con los requisitos de los usuarios. El esquema conceptual es una fuente de información para el diseño lógico de la base de datos.

1.6.2 Diseño Lógico.

El diseño lógico es el proceso de construir un nuevo modelo de datos que utiliza la empresa, basándose en un modelo de base de datos específico (modelo jerárquico, de red, relacional, orientado a objetos) y en el esquema de información creado en el diseño conceptual, independiente del SGBD concreto que se vaya a utilizar y de cualquier otra consideración física.

En esta etapa, se transforma el esquema conceptual en un esquema lógico que utilizará las estructuras de datos del modelo de base de datos en el que se basa el SGBD que se vaya a utilizar, como puede ser el modelo relacional, el modelo de red, el modelo jerárquico o el modelo orientado a objetos. Conforme se va desarrollando el esquema lógico, éste se va probando y validando con los requisitos de usuario.

La *normalización* es una técnica que se utiliza para comprobar la validez de los esquemas lógicos basados en el modelo relacional, ya que asegura que las relaciones (tablas) obtenidas no tienen datos redundantes.

El esquema lógico es una fuente de información para el diseño físico. Además, juega un papel importante durante la etapa de mantenimiento del sistema, ya que permite que los futuros cambios que se realicen sobre los programas de aplicación o sobre los datos, se representen correctamente en la base de datos.

Tanto el diseño conceptual, como el diseño lógico, son procesos iterativos, tienen un punto de inicio y se van refinando continuamente. Ambos se deben ver como un proceso de aprendizaje en el que el diseñador va comprendiendo el funcionamiento de la empresa y el significado de los datos que maneja. El diseño conceptual y el diseño lógico son etapas clave para conseguir un sistema que funcione correctamente. Si el esquema no es una representación fiel de la empresa, será difícil, sino imposible, definir todas las vistas de usuario (esquemas externos), o mantener la integridad de la base de datos. También puede ser difícil definir la implementación física o el mantener unas prestaciones aceptables del sistema. Además, hay que tener en cuenta que la capacidad de ajustarse a futuros cambios es un sello que identifica a los buenos diseños de bases de datos. Por todo esto, es fundamental dedicar el tiempo y las energías necesarias para producir el mejor esquema que sea posible.

1.6.3 Diseño Físico.

El diseño físico es el proceso de producir la descripción de la implementación de la base de datos: estructuras de almacenamiento y métodos de acceso que garanticen un acceso eficiente a los datos.

Para llevar a cabo esta etapa, se debe haber decidido cuál es el SGBD que se va a utilizar, ya que el esquema físico se adapta a él. Entre el diseño físico y el diseño lógico hay una realimentación, ya que algunas de las decisiones que se tomen durante el diseño físico para mejorar las prestaciones, pueden afectar a la estructura del esquema lógico.

En general, el propósito del diseño físico es describir cómo se va a implementar físicamente el esquema lógico obtenido en la fase anterior. Concretamente, en el modelo relacional, esto consiste en:

- ✓ Obtener un conjunto de relaciones (tablas) y las restricciones que se deben cumplir sobre ellas.
- ✓ Determinar las estructuras de almacenamiento y los métodos de acceso que se van a utilizar para conseguir unas prestaciones óptimas.
- ✓ Diseñar el modelo de seguridad del sistema.

1.7 Sistemas Gestores de Bases de Datos (SGBD).

En 1964, se conciben los primeros SGBD, con estos se pretendió dar un viraje a los Sistemas de Archivos, los cuales se limitaban a la estructuración del almacenamiento físico de los datos. Con los SGBD se logró por medio de actividades integradas verlos físicamente en un solo almacenamiento, pero lógicamente se manipulan a través de esquemas compuestos por estructuras donde se establecen vínculos de integridad, métodos de acceso y organización física sobre los datos, permitiendo así obtener valores agregados de utilización.

Viendo la necesidad de mejorar la forma de gestionar la información a través de SGBD se desarrollaron los Sistemas Gestores de Base de Datos Relacionales (SGBDR) cuyas características hacen al sistema mucho más eficiente que los sistemas de manejos de archivos. Esto esta basado en la tesis propuesta por el Dr. Edgar F. Codd en 1970 del Modelo de Datos Relacional, este modelo es el que ha marcado la línea de investigación por muchos años. Representa al mundo real mediante tablas relacionadas entre sí por columnas comunes.

Un SGBDR es un conjunto de datos relacionados entre sí y un grupo de programas para tener acceso a esos datos, permitiendo concurrencia y recuperación. La persistencia de un SGBDR hace referencia a la conservación de los datos después de la finalización del proceso que los creó y la concurrencia se refiere a la capacidad del sistema para gestionar a múltiples usuarios interactuando al mismo tiempo. Entre los SGBDR más comunes se encuentran SQL Server, PostgreSQL, MySQL, DB2 y Oracle.

El sistema gestor de base de datos seleccionado para el desarrollo e implementación del SIGESC fue Oracle en su versión 10g por sus potencialidades y comprobada eficiencia y calidad como software de manejo de datos.

1.7.1 Oracle.

En 1979 la empresa Relational Software Incorporated (RSI) sacó al mercado su producto ORACLE versión 2 y se convirtió en la primera base de datos comercial relacional del mundo. Para 1985 Oracle tenía más de 1,000 bases de datos instaladas. IBM no pudo hacer comercial su tecnología relacional sino hasta 1983.

Ese mismo año RSI fue renombrado como Oracle Corporation para evitar confusión con un competidor llamado RTI. Para ese entonces ya estaba el Oracle versión 3 y ya no solo corría en sistemas de Digital

VAX/VMS sino también en UNIX y otras plataformas. Para 1985 Oracle podía correr sobre 30 distintas Plataformas hasta llegar a más de 70 hoy en día.

Algunas plataformas son curiosidades históricas, pero otras permanecen hasta nuestros días. Actualmente Oracle puede ser usado en plataformas Windows NT/2000/XP y Linux para captar un importante segmento del mercado en franco crecimiento. A continuación se presenta un resumen de lo que pasó después:

- ✓ 1986 Oracle presenta la base de datos Cliente/Servidor
- ✓ 1987 Presenta los programas de desarrollo de cuarta generación (Form y Reports hoy conocidos como Developer).
- ✓ En 1988 Oracle hace programas de aplicaciones financieras
- ✓ 1989 Oracle versión 6
- ✓ 1991 Oracle puede ser corrido en plataformas masivas y paralelas
- ✓ 1993 Oracle versión 7 con optimizadores sugeridos por el usuario y programación a nivel de base de datos.
- ✓ 1997 Oracle versión 8 con orientación a objetos y capacidad masiva de almacenamiento.
- ✓ 1999 Oracle versión 8i con orientación a servicios de Internet, incluyendo programación en Java. 2000 Oracle 9i. Aplicaciones con servicios de tres capas.
- ✓ 2001 Oracle 9i con Cluster reales para servicios críticos.
- ✓ 2002 Oracle 91 release 2 con mejoras de rendimiento.
- ✓ 2003 Oracle 10g orientado a la tecnología grid computing.

Oracle es un manejador de BD relacional que hace uso de los recursos del sistema informático en todas las arquitecturas de hardware, para garantizar su aprovechamiento al máximo en ambientes cargados de información. También proporciona la capacidad de almacenar y acude a los datos de forma consecuente con un modelo definido como relacional (…). Además es una suite de productos que ofrece una gran variedad de herramientas [4].

Es el mayor y más usado Sistema Manejador de Base de Datos Relacional (RDBMS) en el mundo, e incluye cuatro generaciones de desarrollo de aplicación, herramientas de reportes y utilitarios [4].

Entre las características de Oracle se destaca su escalabilidad y alta disponibilidad, aportando un sistema de administración completo para gestionar todas las situaciones críticas de una Base de Datos, por ejemplo

presenta: sistema de seguridad basados en usuarios, grupos y roles, alertas, backups y restauración de datos. Oracle corre en computadoras personales (PC), mainframes y computadoras con procesamiento paralelo masivo. Soporta unos 17 idiomas, corre automáticamente en más de 80 arquitecturas de hardware y software distintos sin tener la necesidad de cambiar una sola línea de código.

La tecnología Oracle se encuentra prácticamente en todas las industrias alrededor del mundo. Oracle es la primera compañía de software que desarrolla e implementa software para empresas 100 por ciento activado por Internet a través de toda su línea de productos: base de datos, aplicaciones comerciales y herramientas de desarrollo de aplicaciones y soporte de decisiones. Oracle es el proveedor mundial líder de software para administración de información, y la segunda empresa de software independiente más grande del mundo.

Oracle posee interacción con todas las plataformas (Windows, Unix, Macintosh y Mainframes). Esto es porque más del 80% de los códigos internos de Oracle son iguales a los establecidos en todas las plataformas de Sistemas Operativos.

Oracle soporta bases de datos de todos los tamaños, desde severas cantidades de bytes hasta gigabytes en tamaño, además provee salvar con seguridad la información y asegurar de los errores vistos en el monitor y la información de acceso y uso.

Oracle soporta un verdadero ambiente cliente servidor. Este establece un proceso entre bases de datos del servidor y el cliente para la aplicación de programas.

Uno de los problemas en comprender un producto masivo como lo es Oracle, es tratar de entender como funciona el producto sin perderse en los miles de detalles que contiene cada solución específica. Oracle ha crecido desde su humilde origen, como una de los tantos gestores de base de datos existentes en los años setentas, hasta convertirse en el mayor líder de este segmento del mercado.

El gestor de base de datos 10g de Oracle es la primera base de datos diseñada para grid computing, la manera más flexible y rentable para manejar la información de la empresa.

Este potente servidor de base de datos ha podido extenderse a muchas áreas de las bases de datos como por ejemplo los clusters, exhibiendo muy buenos resultados en indicadores como rendimiento, disponibilidad, seguridad, confidencialidad. También muestra buenos mecanismos de balance de carga entre sus nodos e implementa excelentes políticas de backup y recovery de los datos.

Es un producto con muy buenas políticas de soporte y capacitación sobre sus herramientas, contando con diferentes métodos y herramientas para mantener a sus clientes actualizados y seguros de sus programas. Brindan soporte desde sitios en Internet de disponibilidad 24x7, con especialistas en las diferentes áreas y mecanismos de solución comprobados para los posibles errores disponibles para sus usuarios.

Es un RDBMS que desde su núcleo incluye la implementación de almacenes de datos, tecnología OLAP, OLTP y minería de datos, asociado a esto incluye un grupo de productos de la llamada capa media o Middleware que nos permiten realizar muchas operaciones sobre los datos almacenados con una amplia perspectiva de inteligencia de negocio, además de lo incorporado por cada una de sus versiones del servidor.

1.7.2 Oracle Edición Enterprise

Está dirigido a implementaciones a gran escala y funciona en más plataformas que la Estándar e incluye mejoras en el manejo de redes, administración, características de Data Warehousing. También tiene otras opciones para funciones especiales tales como integración con datos a sistemas de información geográfica, sonido y video.

1.7.3 Oracle Edición Estándar

Esta versión fue conocida como Servidor de grupos de trabajo (Workgroups). Este producto está considerado base de datos multiusuario pero con un número limitado de usuarios. Actualmente existe para Windows, Unix y Linux.

1.7.4 Personal Oracle

Esta versión es para un solo usuario y es usada normalmente por desarrolladores que trabajan individualmente en sus máquinas. Como "personal" Oracle comparte las mismas características que el Enterprise, las aplicaciones pueden ser transportadas al área de producción real sin ningún problema.

Algunas compañías lo utilizan para aplicaciones móviles o donde requieren un sólo usuario aunque para ello es mejor y más económico utilizar el "Oracle Lite".

1.7.5 Oracle Lite

Fue conocido como el "Oracle Móvil" y esta diseñado para usuarios que utilizan dispositivos móviles inalámbricos. Este producto difiere de los demás porque no utiliza el mismo núcleo del resto de la familia. En lugar de ello Oracle desarrolló un nuevo núcleo que requiere de muy poca memoria para hacerlo compatible con las computadoras portátiles incluyendo las agendas electrónicas basadas en los sistemas operativos Windows CE y Palm OS. (Requiere menos de un megabyte de memoria).

Obviamente es posible intercambiar datos (replicación) entre este producto con cualquier otra base de datos de la familia. Además como Oracle Lite funciona con el mismo SQL que sus hermanos mayores, los programas diseñados para los otros miembros pueden funcionar también con este producto.

1.8 Otros Productos Oracle

1.8.1 Oracle AS 10g (Application Server)

Oracle Application Server 10g proporciona una plataforma para desarrollar y ejecutar aplicaciones empresariales, integrando muchas funciones por ejemplo un entorno de ejecución para Web Services J2EE, complementos de Business Intelligence o una Web caché, aparte de características especialmente enfocadas al grid.

1.8.2 Oracle Forms Developer

Provee de una poderosa herramienta basada en SQLForms para el desarrollo de aplicaciones tradicionales cliente-servidor o para la arquitectura de tres capas utilizando Oracle AS. Es considerada de

4ta generación y permite construir aplicaciones transaccionales muy robustas. La versión 6 de este producto tiene una máquina virtual de Java para su presentación en Internet.

1.8.3 Oracle Reports Developer

Está diseñada para el desarrollo y producción de reportes para ser publicados vía Internet (con Oracle AS) o en el concepto tradicional cliente-servidor.

1.8.4 Oracle JDeveloper

Fue introducido en 1998 para desarrollar aplicaciones en Java. Tiene muchas ayudas para evitar que el programador tenga que escribir mucho código de bajo nivel. El código de Java es más flexible para el concepto de Internet pero desafortunadamente, es menos productivo que el concepto tradicional de 4GL como Oracle Forms. Actualmente se están haciendo esfuerzos para mejorarlo ya que tiene la ventaja de programación orientada a objetos.

1.8.5 Oracle Designer

Provee una interfase gráfica para realizar aplicaciones muy rápidas eliminando la programación tradicional. Esta herramienta recibe los requerimientos y los convierte en programas de Forms, Reports, HTML y C++. Todo esto suena fantástico pero tiene un requisito: los requerimientos deben estar completamente establecidos antes de usarlo, de lo contrario puede requerir más tiempo implementación que con la programación tradicional.

1.8.6 Oracle Discoverer

Es una herramienta para el usuario final que desea generar su propia información a partir de los datos existentes sin depender de un programador. Su objetivo es realizar análisis de negocios: Ver tendencias, creación de escenarios etc.

1.8.7 Oracle AS Portal

Como producto fue introducido como WebDB en 1999 y provee una herramienta para desarrollar páginas HTML en Internet con capacidad de utilizar otros productos de Oracle como Reporte.

La gran mejora de este producto respecto a su antecesor es que se puede usar "portlets" que permite a una página de Internet dividirse entre diferentes áreas que pueden independizar la información desplegada e interactuar con el usuario.

1.8.8 10g Grids: en la Empresa

Con Oracle Database 10g, es la primera base de datos diseñada para grid computing, usted puede automatizar los servidores en cluster y la administración para asignar recursos en forma dinámica. Las críticas editoriales elogian el valor, la escalabilidad y capacidad de administración de Oracle, y los analistas colocan a Oracle como el líder en amplitud de visión y capacidad de ejecución.

Oracle ofrece los beneficios de clustering alta disponibilidad y escalamiento horizontal ha pedido—con Oracle Real Application Clusters [5].

10.8.9 Real Application Cluster (RAC)

Oracle Real Application Clusters permite que una única base de datos se expanda por múltiples todos en un grid o red, uniendo los recursos de varias máquinas. Esto que requería un proceso en versiones anteriores del servidor se puede hacer inmediatamente en Oracle 10g, y se puede empezar a balancear el flujo de trabajo hacia la nueva máquina que se incorpora al grid, a la vez que abandonarla cuando ya no es necesario. Otros Sistemas de Bases de Datos no pueden hacer esto dinámicamente cuando la base de datos se encuentra ejecutándose. El nuevo software de cluster en Oracle 10g simplifica el proceso eliminando la necesidad de adquirir, instalar y configurar estas herramientas de terceros. Se pueden añadir servidores a la vez que eliminarlos en un cluster Oracle sin tiempo de inactividad, es decir, sin detener la base de datos, sin importar tampoco la plataforma donde se encuentra instalado el servidor.

1.9 Conclusiones.

Después de realizado el estudio de los conceptos más importantes que formarán parte de este trabajo, se han podido apreciar elementos importantes como son la amplia cantidad de productos que facilita el servidor de bases de datos Oracle, las características más importantes que justifican la elección de dicho gestor de base de datos. Además se exponen conceptos como seguridad ciudadana, que aclaran la importancia que tiene este aspecto para la población de Venezuela. Se explica qué son los centros de gestión de emergencia, servicio que indiscutiblemente, por su objetivo y las cosas que permite lograr, tendrá incidencia directa en los problemas que presenta actualmente la República Bolivariana de Venezuela en relación con la seguridad de sus ciudadanos. De forma general se explica, sin entrar en detalles, la metodología que será utilizada para realizar el análisis, diseño e implementación de la base de datos. De esta forma queda el camino despejado de dudas acerca de los elementos más importantes que soportarán el posterior desarrollo del trabajo.

Capítulo II: Diseño de Base de datos.

2.1 Introducción.

En este capítulo se pretende ilustrar la problemática de cada uno de los módulos principales del Sistema de Gestión de Emergencia de Seguridad Ciudadana (171), **Módulo de Recepción de Llamadas (Operador), Módulo de Despacho (Despachador), Módulo de Supervisión de Operadores**, **Módulo de Supervisión de Despachadores**, **Módulo de Administración y Control de Recursos.** Asociado cada uno de ellos se describirá el diseño de base de datos propuesto como solución a las necesidades de almacenamiento de información operacional o de configuración según corresponda.

2.2 Desarrollo.

2.2.1 Módulo de Recepción de Llamadas.

Descripción del Módulo.

Un Centro de Gestión de Emergencia organiza su personal en diferentes áreas de trabajo, por ejemplo operadores, supervisores de operadores, despachadores, supervisores de despacho, entre otros.

Al **Módulo de Recepción de Llamadas** tienen acceso los operadores del Centro 171. Utiliza dos pantallas por posición, una para la mapificación de puntos de referencias, solicitudes y recursos, y otra para registrar en el sistema las llamadas recibidas.

Tiene como función principal la recepción de llamadas telefónicas de la población, permite determinar solicitudes repetidas de otra solicitud que esté registrada en el Sistema, muestra las preguntas que se deben realizar al usuario y las recomendaciones a indicar según el motivo de la solicitud, muestra las áreas de despacho que pueden atender una solicitud permitiéndole al operador añadir o eliminar áreas de la lista propuesta. Permite obtener la dirección exacta del lugar donde está ocurriendo el hecho según lo que informa el usuario, si éste no conoce la dirección exacta permite obtener la dirección por Puntos de Referencia o ayudándose del mapa digital.

Las llamadas se pueden clasificar como: informativa, falsa, de agradecimiento, abandonada y queja.

Se registra fecha, hora inicio y tiempo de duración de las llamadas recibidas, permitiendo calcular la fecha y hora fin las llamadas telefónicas, factor que influye en la determinación de la eficiencia en el servicio de atención al ciudadano.

Se permite transferir llamadas por sistema a otros operadores, al supervisor que supervisa al operador autenticado o al supervisor general.
El operador puede registrar además de solicitudes: eventos, personas desaparecidas, autos robados y traslados médicos.
El operador puede brindar información a la población de solicitudes que estén registradas en el Centro 171 y de organismos que trabajan para la seguridad ciudadana.

Elementos Representativos del Diseño.

Entre los elementos significativos en el diseño de la base de datos de este módulo se encuentra la creación de una generalización de llamada y cámaras de video vigilancia como fuentes de ingreso de solicitudes de emergencias al SIGESC como muestra la Fig. 1.
Este diseño permite no solo recepcionar las solicitudes de emergencia por las vías identificadas hasta este momento, sino que al surgir una nueva se adiciona a través del módulo de administración al nomenclador **nTipoFuente** e inmediatamente el sistema podrá almacenar solicitudes de emergencias que no provienen de llamadas telefónicas, ni de cámaras de video vigilancia. Con esto desarrollamos un software garantizando la escalabilidad en el tiempo.

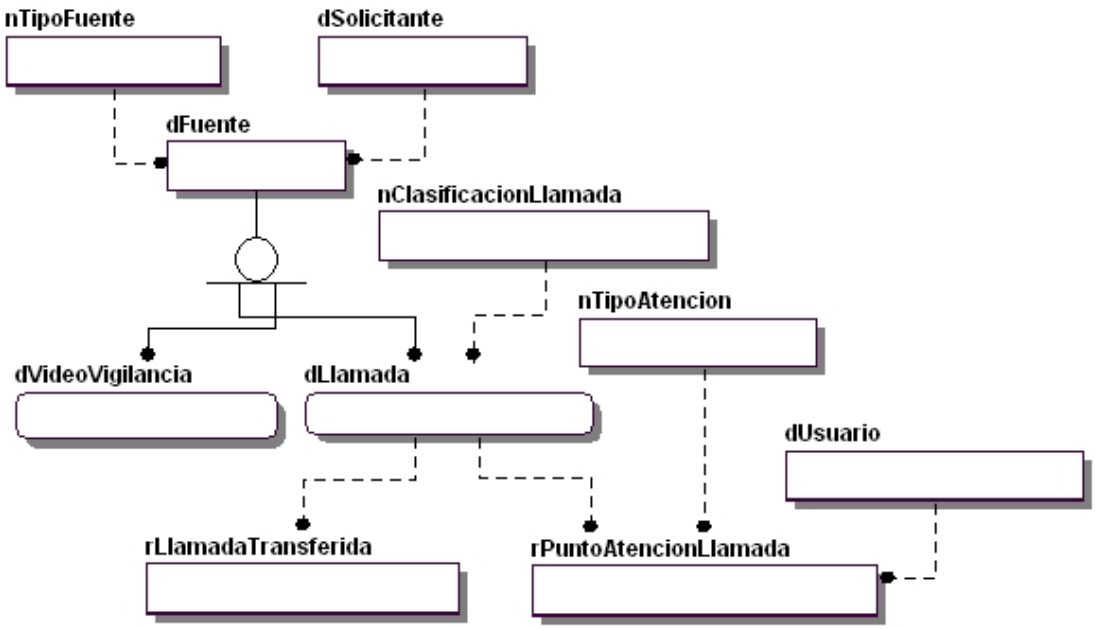


Fig. 1 Generalización de Llamada.

Los operadores en su trabajo pueden durante la atención de una llamada trasferirla a otros operadores, a los supervisores de operadores y al supervisor general. En la base de datos quedan almacenados los datos asociados a la transferencia, donde está incluida la información de la solicitud y de la llamada que se había recuperado hasta el momento de la transferencia, la Fig. 2 nos muestra el diseño.

Con esta estructura podemos almacenar desde que punto[1] se transfiere una llamada y que punto la recibe, se almacena el identificador del Punto y no el de **PuntoOperador**, **PuntoSupervisorOperador**, **PuntoSupervisorGeneral**, porque de esta forma utilizando restricciones de inserción podemos controlar que solamente se transfiera desde los puntos que la configuración de la aplicación establece, y al mismo tiempo somos consecuentes con la flexibilidad que exige el sistema.

[1] Estación de trabajo del Centro 171.

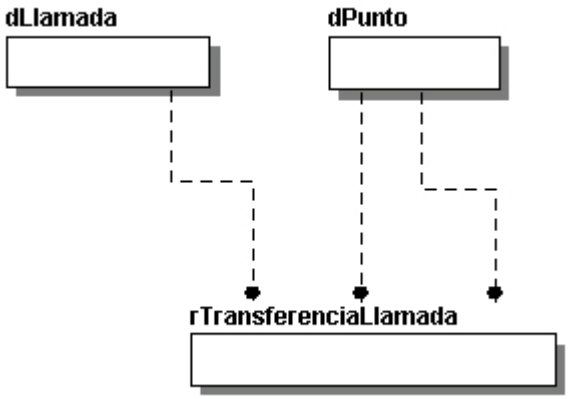


Fig. 2 Transferencia de Llamada.

Además de gestionar las solicitudes de emergencias, el SIGESC también recepciona y da seguimiento a las solicitudes de traslados médicos, reporte de personas desaparecidas y autos robados. La forma de operar para la coordinación de los órganos de seguridad ciudadana es relativamente diferente, el traslado médico genera una solicitud que el centro debe despachar, mientras que los otros dos no, solamente son para que el centro cuente con información y pueda operar con ella. El diseño normalizado se presenta en la Fig. 3.

Este modelo provee de mayores facilidades de almacenamiento de información para el sistema, ya que para cada reporte de auto robado y persona desaparecida almacenamos los datos de uno o más solicitantes y los datos asociados a cada una de las fuentes de información utilizada para comunicarse con el centro.

Las denuncias de personas desaparecidas tienen un denunciante asociado, se decide complementar la información de este denunciante con la información de la persona solicitante.

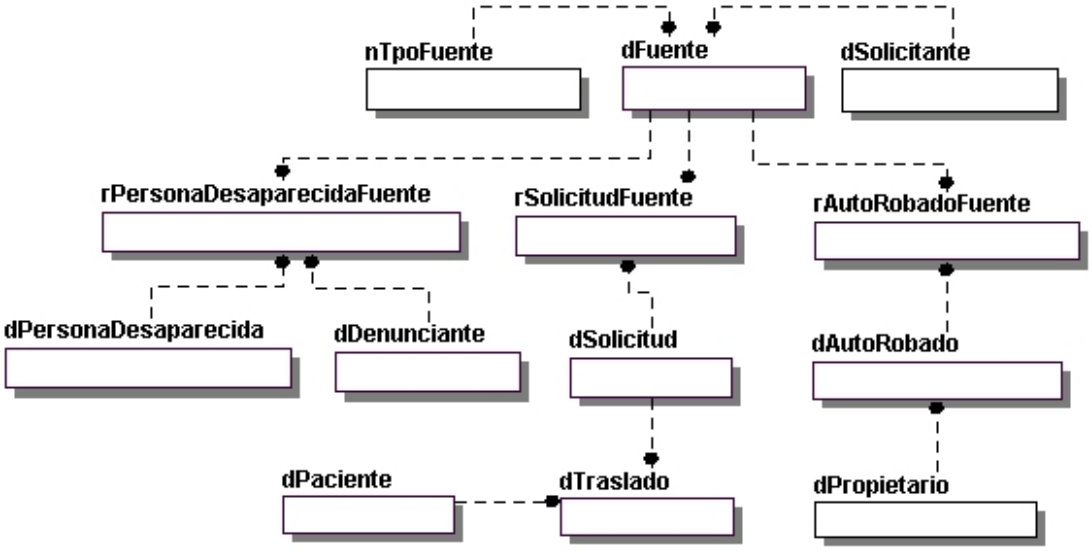


Fig. 3 Persona desaparecida, Auto robado y Traslado médico.

Existen determinados eventos que por su importancia o relevancia necesitan una custodia, seguimiento o simplemente desarrollar un grupo de actividades que garanticen la seguridad de los participantes, el SIGESC recepciona los eventos que las personas, tanto jurídicas como naturales, le solicitan custodia. Para lograr coordinar correctamente los órganos de seguridad ciudadana, el Centro 171 requiere de información como es la fecha del evento, lugar donde se efectuará, que tipo de evento es, entre otros. El diseño propuesto se presenta en la Fig. 4.

Las solicitudes una vez que son registradas el sistema será capaz de recomendar a los operadores si una nueva llamada esta asociada a una solicitud que ya está registrada, para esto se utiliza el motivo y la dirección, y en este caso las llamadas se unen por relación a la misma solicitud, haciendo que la descripción de solicitud de emergencia sea la suma de las descripciones de cada llamada asociada a ella.

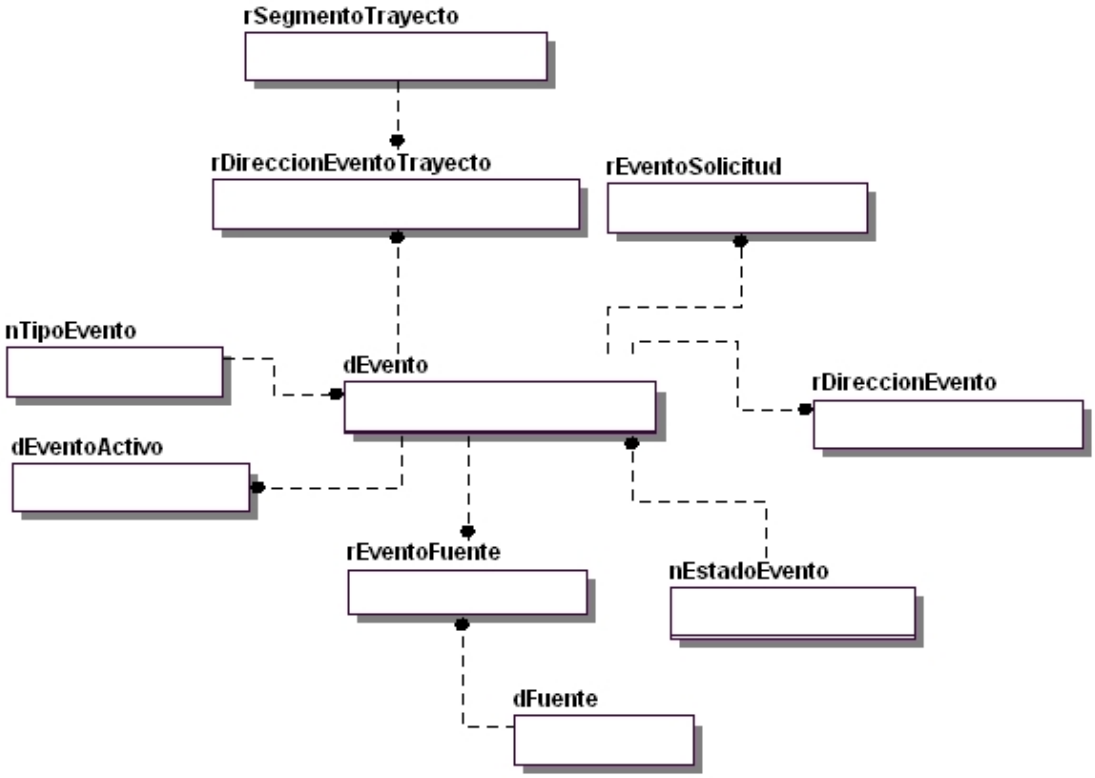


Fig. 4 Evento dirección.

Este modelo considera los diferentes tipos de direcciones que puede tener un determinado evento. Pueden ser puntuales, o una trayectoria, permite almacenar en el caso del trayecto que dirección lo inicia, cual lo termina y que calle une a estos dos puntos, esto desde el punto de vista del sistema facilita la ubicación de los recorridos en el mapa digital y le aporta una mayor claridad al despachador que atienda el evento.

2.2.2 Módulo de Supervisión de Operadores.

Descripción del Módulo.

El Módulo de Supervisión de Operadores es fundamental para lograr el buen funcionamiento del Centro 171. Este está diseñado para los supervisores de operadores y los supervisores generales del Centro 171.

Por medio de este se supervisan las acciones de los operadores usando dos vistas diferentes, en las cuales los datos que se muestran de los operadores, y la forma en que se exponen son distintos. Se muestran además las llamadas y solicitudes atendidas por los operadores supervisados.

Permite observar los datos de un operador en el turno, intervenirle a un operador la atención de una llamada con el registro de solicitud asociado, buscar operadores del centro, llamadas y solicitudes registradas por los operadores asignados al supervisor.

Este módulo brinda la posibilidad a los supervisores de operador de monitorear todas las llamadas que son atendidas por los operadores que ellos atienden.

Permite además, ver los detalles de solicitudes y llamadas y modificar estas últimas, localizar operadores e indicar el estado de bloqueo de los operadores.

Se pueden controlar los tiempos de bloqueo de sesión y de atención de las llamadas y registro de solicitudes para los operadores asignados al supervisor. Se muestran también las estadísticas del turno para los operadores asignados.

El módulo brinda la posibilidad de registrar, mostrar y ver los detalles sobre incidencias.

Con toda la información generada la aplicación muestra estadísticas del turno de trabajo una herramienta más para el control de los supervisores.

Elementos Representativos del Diseño.

Este módulo fundamentalmente extrae mucha de la información generada por el módulo de operador. Sin embargo podemos resaltar algunos elementos relevantes del diseño.

Al punto supervisor de operador se le asignan puntos operadores a supervisar, de los usuarios que se autentican en dichos puntos se extrae un grupo de datos que constituyen las estadísticas del módulo. Algunos de los parámetros medidos son el tiempo de bloqueo de sección, llamadas atendidas y solicitudes registradas por cada uno de los operadores que supervisa, entre otros elementos.

Este diseño brinda mucha flexibilidad y ventajas desde el punto de vista de la aplicación ya que los usuarios supervisores de operadores supervisan a los operadores que están autenticados en los puntos de operador, que el punto de supervisión de operador en el que dicho supervisor está autenticado, supervisa. Es recomendable que los usuarios supervisores de operador y operadores trabajen en los mismos puntos, de forma tal que se propicie la comunicación entre ambos, sino es así el sistema funcionara correctamente.

El modelo de la Fig. 5 pretende almacenar información asociada a un perfil para el usuario supervisor de operadores, donde se guarda qué vista para mostrar la información está utilizando el supervisor de operadores. Además se relacionan a los supervisores con los operadores que están controlando a nivel de usuarios para permitir que los supervisos puedan controlar a sus operador sin importar en que punto este autenticados.

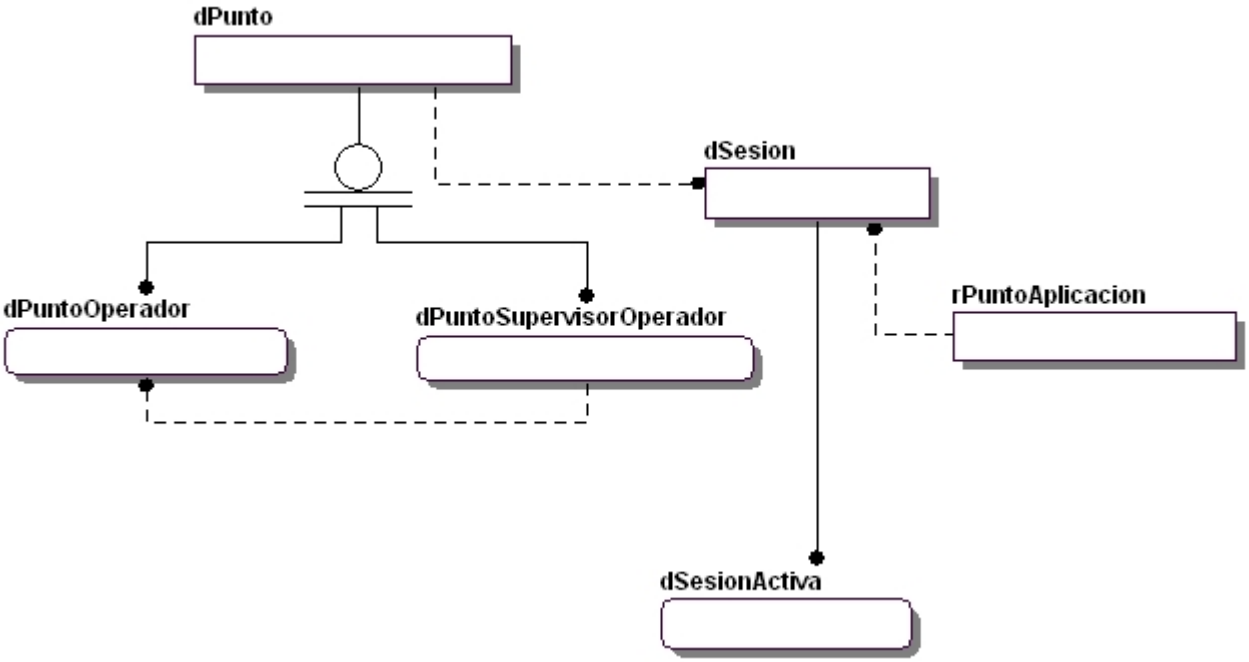


Fig. 5 Supervisión de Operadores por Puntos.

Se utiliza para la solución la entidad **dSesion** para almacenar las sesiones de cada uno de los usuarios, mediante la cual podremos controlar las estadísticas de los operadores en un turno de trabajo, fecha y hora de inicio sesión y fecha y hora de fin de sesión, rol con el que acceden al sistema. Asociado con esto existe en el diseño una entidad **dSesionActiva** con la cual aumentamos el rendimiento de las consultas sobre la entidad **dSesion**, ya que esta última entidad almacenará las sesiones históricas y realizar una búsqueda sobre esta requeriría de muchos más recursos del servidor que si seleccionando primeramente sobre las sesiones que están activas. Las búsquedas en las sesiones históricas normalmente no son necesarias para el trabajo operacional del sistema y no forman parte de los datos que se consulten

constantemente, son consultados mayormente para realizar reportes, cálculos de promedios de tiempo de bloqueos de sesión, entre otros elementos.

2.2.3 Módulo de Despacho de Solicitudes.

Descripción del Módulo.

Encaminado a los usuarios despachadores del Centro, utiliza dos pantallas por posición, una para el mapa digital y otra para el despacho de las solicitudes.

Por medio de este módulo los despachadores pueden asignar unidades a las solicitudes pendientes, la asignación de la primera unidad pone la solicitud en estado en proceso.

Una solicitud de emergencia puede estar siendo despachada simultáneamente en varias áreas de despacho.

Las áreas de despacho se configuran en el Módulo de Configuración de Operaciones del SIGESC, teniendo en cuenta una combinación de organismo-área geográfica del estado.

Los despachadores pueden ordenar, a voluntad, las solicitudes según el motivo o por el tiempo que llevan pendientes.

Existe interactividad con un mapa digital, por medio del cual el despachador puede observar la ubicación geográfica de la solicitud, la posición de las unidades tanto móviles como fijas y obtener direcciones.

Los despachadores pueden transferir solicitudes por sistema hacia el supervisor que lo atiende o al supervisor general.

El sistema muestra ayuda en la determinación de solicitudes repetidas, en cuyo caso permite unir las fichas de dichas solicitudes para observar la descripción de la solicitud de la manera más completa posible.

Entre otras funcionalidades, se pueden: despachar las solicitudes de traslados médicos programados para el turno actual, reportar vehículos y personas encontradas y registrar incidencias sobre el comportamiento de las unidades.

El cierre de un despacho de solicitud permite clasificar la atención de la emergencia en las siguientes formas: efectivo, sin efecto, sabotaje, cancelada, repetida, no atendida por el organismo.

Elementos Representativos del Diseño.

Una solicitud según su motivo y dirección puede ser despachada en varias áreas de despacho, de manera que por cada una de ellas puede tener asignación de recursos, despachadores que la estén atendiendo y estados. Se propone para este módulo la creación de una agregación, donde se relacione la solicitud con las áreas de despacho en las que le corresponde ser atendida.

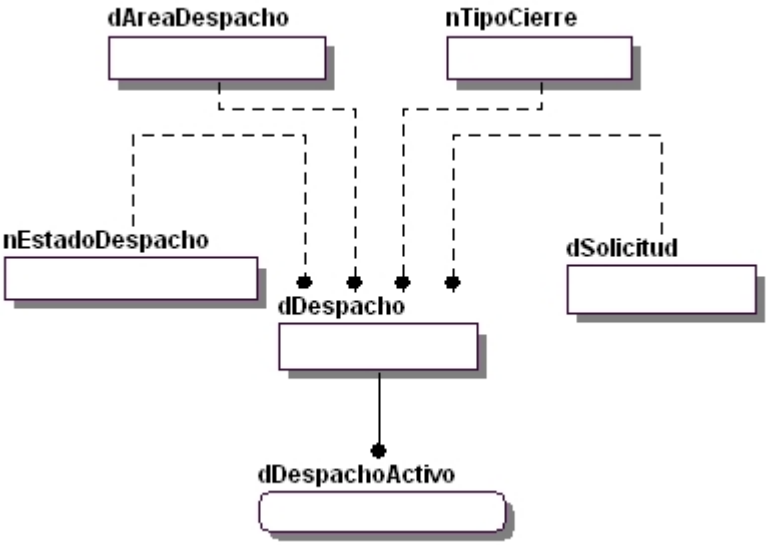


Fig. 6 Despachos.

En este diseño se representa a la agregación **dDespacho** como una entidad en la que se almacena la solicitud, el área en la que está siendo despachada, el estado que tiene esa atención, entre otros elementos. Estas entidades se crean con el objetivo de reducir la redundancia de la información y dejar a operaciones SQL la inserción de un grupo de elementos como es el estado de la solicitud, que es determinada a partir de los estados de los despachos que tenga.

De igual forma se propone la utilización de una entidad **dDespachoActivo**, en la que se almacenarán los despachos con estados pendientes y en procesos, proporcionando mayor rendimiento a la hora de realizar el trabajo operacional, ya que para obtener estadísticas del turno, o datos de un determinado despacho, no tendríamos que realizar la búsqueda sobre la entidad **dDespacho** que almacena los datos de despachos históricos.

Durante la atención de los despachos se asignan recursos, los recursos pueden tener dos estados: libres u ocupados. Al mismo tiempo cada despachador podrá apreciar estados de ocupado de esos mismos recursos, en camino y en el sitio, estos estados le permiten a los despachadores saber qué recursos han sido ocupados por ellos, cuáles ocupados por su supervisor y cuales libres. Un despachador puede liberar solo los recursos que fueron ocupados por el, de forma tal que puede tener en su lista de recursos ocupados, algunos que han sido ocupados por el supervisor que lo atiende.

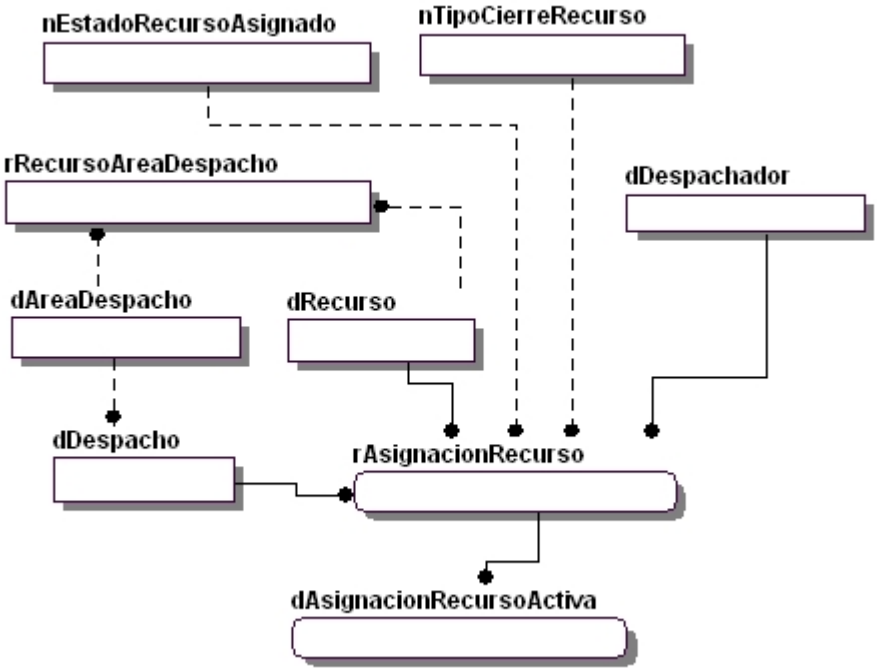


Fig. 7 Asignación de Recursos.

Con este diseño se puede determinar quién, si despachador o supervisor de despacho, tiene ocupado un recurso y permitir o denegar según corresponda. Los usuarios pueden tener más de un rol, como es el caso de los supervisores de despacho que además son despachadores, por lo tanto un supervisor puede ocupar un recurso de un despachador, pero lo hará como despachador de forma tal que estará registrado en la entidad **rAsignacionRecurso**. En la entidad **rRecursoAreaDespacho** se almacena el estado de ocupado, así el despachador no podrá reasignarlo, ni liberarlo ya que no tiene el control del recurso.

Las solicitudes pueden ser detectadas por lo operadores como repetidas, en este caso las fichas se unen por relación y muestran una información conjunta de la solicitud de emergencia.

Los despachos de solicitudes también pueden ser detectados como repetidos por despachadores y supervisores de despacho, y se relacionan para enriquecer la información que se tiene del despacho, y para optimizar el funcionamiento del centro ya que no se asignan recursos a la solución de una misma solicitud dos veces.

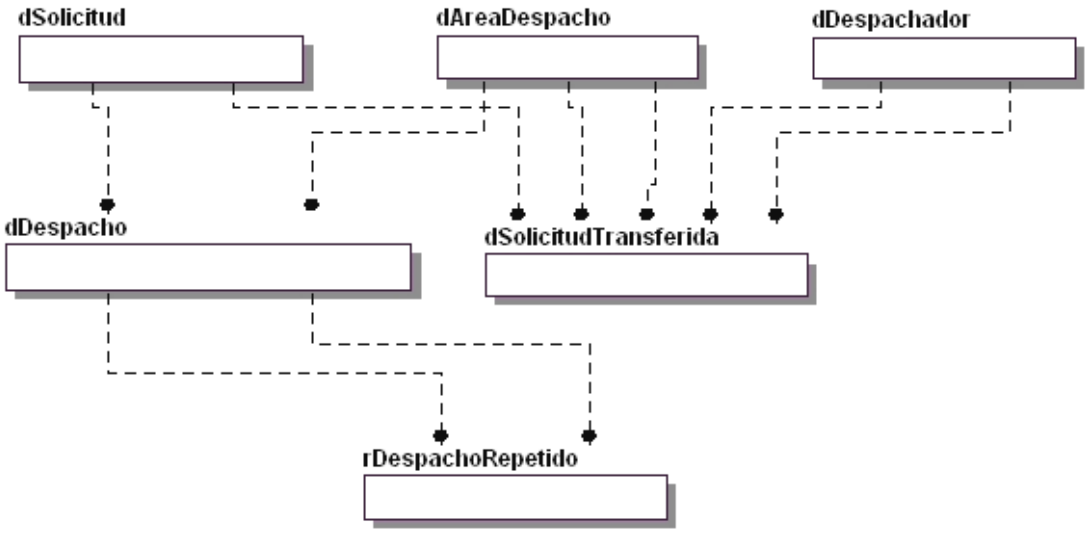


Fig. 8 Despacho de solicitud transferida.

Las solicitudes pueden ser transferidas por los supervisores de despacho o a otros despachadores, ya que al cambiar el motivo o dirección de la misma puede ser enviada a nuevas áreas de despacho.

Los despachos de solicitud también pueden ser transferidos por los despachadores y supervisores de despacho, para autorizar la transferencia primero el sistema determina cuales son las transferencias permitidas almacenadas en los elementos de configuración del centro.

La entidad **dSolicitudTranferida** almacena la información de transferencia, el identificador de solicitud, el área de despacho que envía, la que recibe, el despachador que realiza la transferencia, el que la acepta, entre otros elementos. Este diseño nos permite almacenar la transferencia de una solicitud y la de despacho de solicitud ya que con la información almacenada cubrimos todo lo necesario en cada tipo de transferencia, la transferencia de solicitud no es más que las transferencias de todos sus despachos.

2.2.4 Módulo de Supervisión de Despacho de Solicitudes.

Descripción del módulo.

En el Módulo de Supervisión de Despachadores los supervisores tienen la posibilidad de controlar el trabajo de los despachadores bajo su supervisión. Utiliza dos pantallas por posición, una para el mapa digital y otra para el control de los despachadores.

Un Módulo de Configuración de Operaciones para el Centro 171 permite definir cuáles despachadores quedan supervisados por qué supervisor. Se permite modificar esta configuración desde el Módulo de Supervisión General por parte del Supervisor General del turno, para situaciones excepcionales que requieran la modificación de la asignación supervisor-despachador.

Con respecto a un despachador, se puede retirar el despacho de una solicitud que esté siendo procesada, en cuyo caso se avisa al despachador correspondiente y se elimina la solicitud a ese despachador. También se pueden observar los tiempos en que las solicitudes de cada despachador están en estado *Pendiente* y *En Proceso*.

La interfaz principal muestra las transferencias de despacho de solicitudes recibidas y enviadas entre las diferentes áreas, y el estado de estas en función de la presencia o no del despachador que la atiende.

El Supervisor de Despacho puede despachar cualquier solicitud asignada a cualquiera de las áreas de los despachadores que el supervisa. Se apoya en el Módulo de Despacho y el Módulo de Mapificación que muestra todas las zonas geográficas de las áreas que supervisa. Tiene acceso a las mismas funciones de manejo de unidades con o sin ayuda del mapa.

Este Módulo muestra estadísticas del trabajo de los despachadores en el turno, que permiten tener una mejor visión de la situación en las áreas que supervisa.

Elementos Representativos del Diseño.

El módulo de supervisión de despacho fundamentalmente consulta información almacenada por el Módulo de Despacho, genera mediante operaciones SQL un número de estadísticas para los usuarios de este módulo. Algunas de estas estadísticas son:

- ✓ Cantidad de áreas asignadas.
- ✓ Cantidad de áreas que están sin atención.
- ✓ Cantidad de despachadores autenticados.
- ✓ Cantidad de despachadores inactivos

No almacena mucha información, pero contiene elementos relevantes en su diseño.

Los usuarios supervisores realizan su trabajo en puntos del sistema y controlan el trabajo en las áreas de despacho. La asignación de estas áreas al supervisor es de forma indirecta ya que la configuración del centro se establece teniendo en cuenta el punto como centro de atención, quedando como se muestra en la Fig. 9.

Existe una especialización de puntos de trabajo, llegando a puntos de supervisores de despacho en los que trabajan los usuarios supervisores de despacho. Al punto de supervisor de despacho se le asignan puntos de despachos a supervisar de forma tal que un punto de despacho pueda ser supervisado por un punto de supervisor de despacho y un punto de supervisor de despacho pueda supervisar varios puntos de despacho.

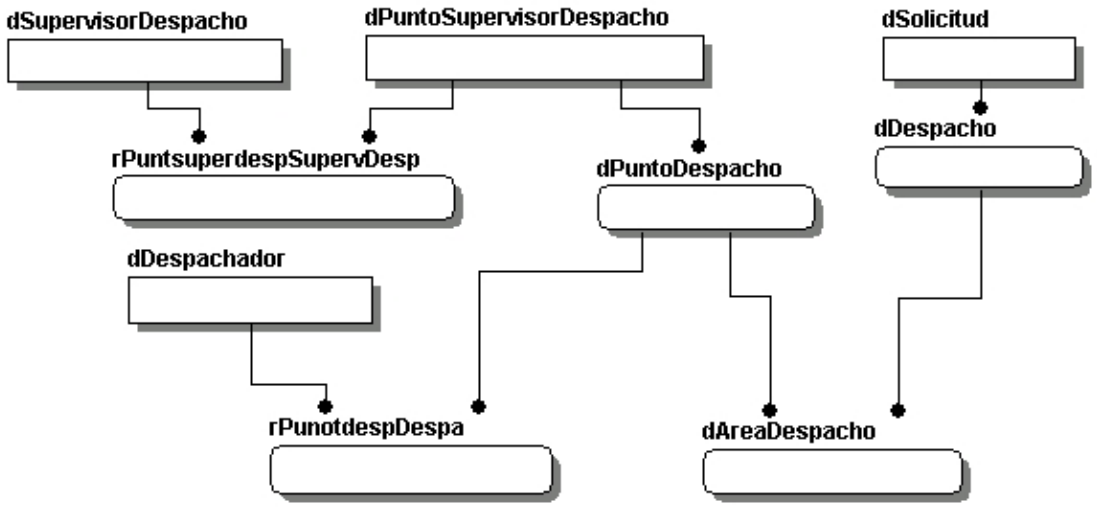


Fig. 9 Control de Áreas de Despacho.

A los puntos de despacho se le asignan áreas de despacho a despachar, además de despachadores que trabajaran en el, estableciendo que los despachadores que trabajen en ese punto tendrán además las áreas de despacho que tiene asignada el punto de despacho. Este control por áreas en puntos y no de despachadores por parte del supervisor responde a que el control por despachador es limitado.

El control por despachadores brinda la flexibilidad de que los usuarios despachadores puedan sentarse a realizar su trabajo en cualquiera de los puntos dedicados a despacho, esto es una ventaja para el usuario, pero una desventaja muy grande para la aplicación ya que la recuperación de la información de configuración de la aplicación y del usuario tendrían que ser cargadas siempre que alguno de los usuarios se autenticara en el sistema, esta operación demoraría el inicio de sesión.

Por otra parte se soluciona el problema de la supervisión por usuarios que consistía en que si no se tenía a un despachador en una área determina, el supervisor no tenía como controlar que no se perdieran dichas solicitudes, ya que el lo que controlaba eran usuarios y por tanto lo que estaría visualizando en su aplicación serían los despachadores autenticados dejando fuera las áreas que no tenían a ningún despachador autenticado, esto queda solucionado con este diseño, puesto que ahora se supervisan áreas tengan o no un despachador que las despache. Esto permite que el supervisor de despacho sepa de las solicitudes que llegan a todas las áreas y pueda tomar dediciones en el caso de que lleguen a áreas donde no existan despachadores autenticados.

Este diseño permite mostrar las dos vistas control de usuarios despachadores y control de áreas de despacho que es la solución que brinda más información al supervisor. Las áreas de despacho pertenecen a un organismo y un organismo puede tener varias áreas dependiendo de sus necesidades, entre ellas cuentan: que el organismo atienda un área geográfica muy extensa y por tanto serían muchos recursos a su disposición para despachar por un solo despachador; que actividad delictiva sea muy fuerte en diferentes zonas del área, esto generaría un número elevado de solicitudes en cortos espacios de tiempo dejando una alta probabilidad de que no se puedan atender todas correctamente.

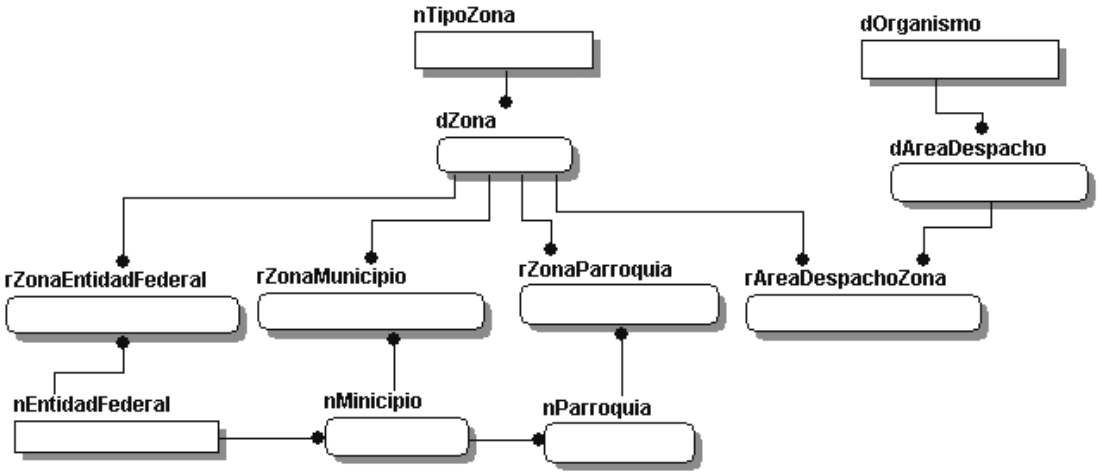


Fig. 10 Zona Geográfica del Área de Despacho.

Este modelo permite definir las situaciones antes planteadas, de forma tal que el área pueda quedar definida solamente por el área geográfica que abarca el organismo, por una parroquia que atienda ese

organismo, por un municipio que atienda ese organismo incluyendo todas las parroquias de este, hasta el nivel de entidad federal con todos los municipios y parroquias que incluyen sucesivamente.

Esta forma de definir las áreas de despacho es muy ventajosa para el sistema ya que los mapas digitales nos permiten definir capas de entidades federales, municipios y parroquias de forma tal que se les pueda dar mayor información visual a los usuarios de la aplicación que utilicen el mapa y es a su vez más sencillo de hacer para el sistema.

2.2.5 Módulo de Administración y Gestión de Recursos.

Descripción del módulo.

El Módulo Administración y Control de Recursos es un módulo Web.

A este módulo tienen acceso los administradores de recursos tanto del Centro 171 como los de los organismos que suministran recursos al Centro.

Por medio de este módulo se definen los recursos que estarán disponibles para la atención de las emergencias.

Los organismos externos definen los tipos de recursos disponibles cada día, con un horario posible para su uso. Un administrador de recursos del centro consulta la disponibilidad de los recursos según su horario publicado, para habilitar o deshabilitar el recurso a los despachadores.

El módulo permite observar los mantenimientos programados a los recursos, como información a tener en cuenta para la habilitación de los recursos.

Así mismo, permite a los administradores de recursos del Centro 171 solicitar por la vía de la Web los recursos que necesita para fechas y propósitos determinados.

Los informáticos del Centro 171 podrán incorporar nuevas categorías y tipos de recursos a manejar, además de los que se consideran como predeterminados y gestionarlos.

El paquete de opciones de Gestión de Organismos permite definir los organismos que interactuarán con el Centro 171.

Se podrán imprimir expedientes de recursos, que contienen los datos del recurso, los mantenimientos que se le han efectuado, así como las incidencias en las que ha estado involucrado.

El módulo permite obtener reportes sobre el uso de los recursos.

Elementos Representativos del Diseño.

Los recursos constituyen uno de los componentes más importantes en la atención de una solicitud de emergencia, una patrulla, un carro de bombero, una ambulancia, entre otros.

La gran variedad de recursos entre organismos, así como su forma tan diversa y la información que se desea almacenar de cada uno, hace más complejo el diseño.

Por esta razón se propone un diseño que abarque todo lo dicho anteriormente y que por consiguiente permita la creación de entidades de recursos dinámicamente.

En este modelo se proponen la creación de un grupo de entidades fijas que nos permitirán almacenar la información necesaria para la creación dinámica de las entidades reales, información como entidad, atributos que tiene, si son capos únicos, criterios de búsqueda, nulos, llaves foráneas de otra entidad, tipo de dato de ese atributo.

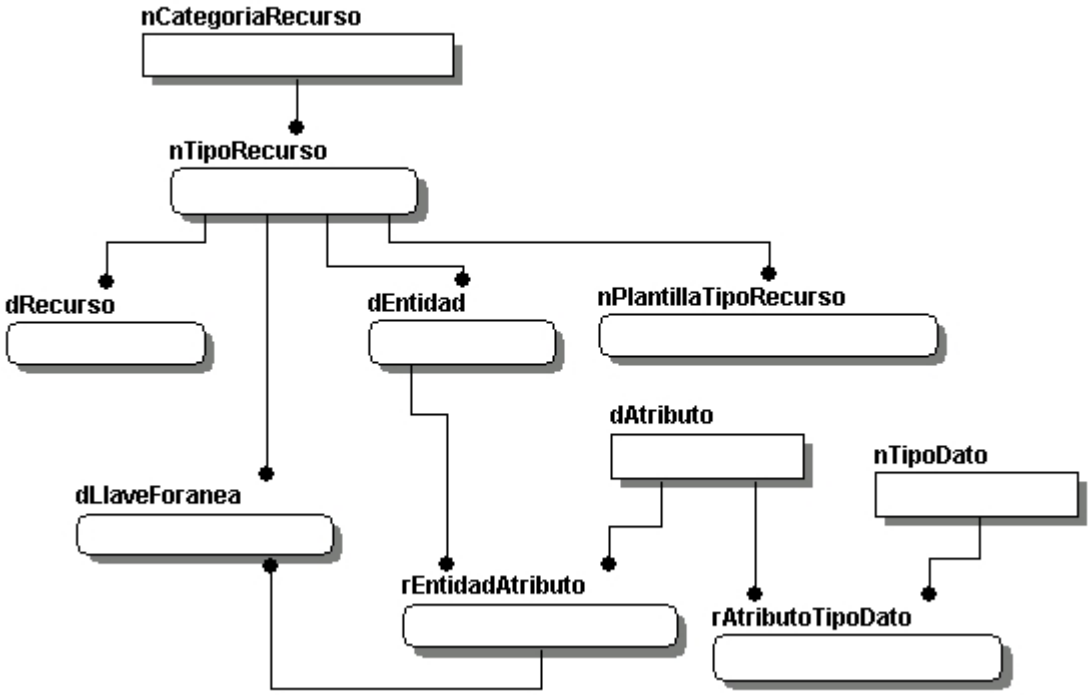


Fig. 11 Meta Dato de Recursos.

Todas las entidades que se creen en el sistema serán almacenadas aquí y tendrán un identificador que se unirá a la entidad **dRecurso** creando una especialización de recurso, por tanto la llave primaria de todas las entidades serán de tipo la llave de **dRecurso**. Las llaves alternas se crearan a partir de la información almacenada en la entidad **rEntidadAtributo** de donde se obtiene si es o no llave alterna, se le pone el

nombre compuesto por el prefijo AK y el nombre de la entidad, y las llaves foráneas almacenamos en la entidad **dLlaveForanea** el par **idEntidad-idAtributo** más el tipo de recurso con el cual se establece se puede relacionar, el tipo de dato será el de **idrecurso** y nombre del atributo será el nombre del tipo. Esto nos permite lograr operaciones más rápidas ya que desde la misma tabla **dAtributo** se puede conocer con que tipo o tipos de recurso se relaciona esta entidad a la que pertenece un determinado atributo.

El tipo de dato igual que el de **idrecurso** permite, a través de la generalización dRecurso, crear relaciones entre los recursos de cualquier tipo, en este caso el sistema dejará a la experticia de los administradores del sistema de administración de recursos, la creación de nuevos tipos de recursos. La generalización nos da mucha flexibilidad ya que mediante ella asignamos recursos a áreas de despacho, realizarle mantenimientos, asignarles imágenes y planos para distintos tipos de recursos según establezca el administrador. La entidad **dEntidad** será la encargada de almacenar todas estas características de los recursos.

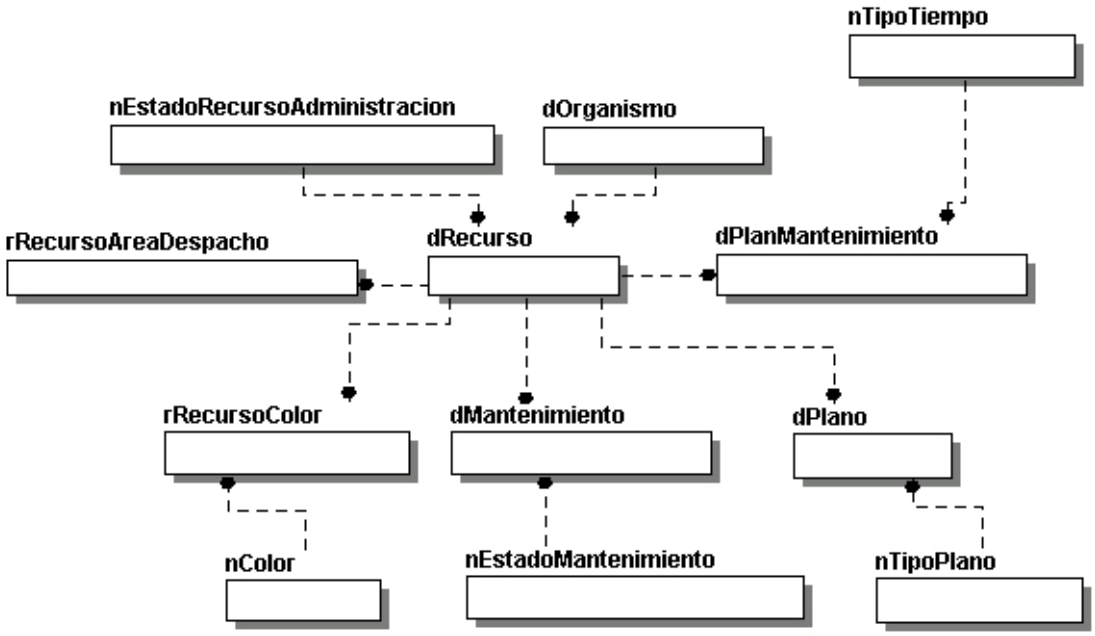


Fig. 12 Recurso.

A los recursos se les planifican mantenimientos y luego entran en mantenimiento, en el diseño se proponen entidades que almacenan el plan del mantenimiento y la información del mantenimiento. La entidad **dPlanMantenimiento** almacena la fecha y hora de inicio, fecha y hora de fin del mantenimiento y

una cantidad de tiempo de ciclo, con esta información podemos almacenar en la entidad **dMantenimiento** cuáles son los mantenimientos pendientes y con la información de fecha y hora de fin y la información de ciclo de la entidad **dPlanMantenimiento** podremos insertar un nuevo registro en **dMantenimiento** que nos permita determinar cuando entra en mantenimiento nuevamente un recurso. Con este diseño logramos que el usuario establezca un plan de mantenimiento una vez, a partir de ese automáticamente se generen el resto de los mantenimientos de ese recurso, teniendo en cuenta que el usuario en cualquier momento puede crear un nuevo plan de mantenimiento para cualquier recurso, permitimos que reajuste el tiempo de mantenimiento de los recursos y el tiempo de ciclo.

Los recursos pertenecen a un organismo y un organismo tiene muchos recursos. En nuestro sistema tenemos a la entidad organismo como un concepto independiente del recurso que representa su edificio o sus locales físicos sin dejar de tener en cuenta que tienen la relación de pertenencia descrita en la oración anterior.

2.3 Conclusiones.

En este capítulo se realizó una descripción del funcionamiento de algunos de los módulos que, para la etapa temprana de desarrollo del sistema, sea han comenzado a implementar. Se describieron elementos significativos del diseño para cada uno de ellos mostrando diagramas para una mejor compresión de estos.

Capítulo III: Administración del servidor.

3.1 Introducción.

El tema de administración de servidores es extenso, por tanto en este capítulo se han descrito solamente algunos elementos que forman parte del gran número de factores que se deben tener en cuenta a la hora de hablar de configuración y administración de servidores de base de datos Oracle. Se hace este estudio teniendo en cuenta que el sistema debe quedar optimizado para un Sistemas de Gestión de Emergencias de Seguridad Ciudadana. Se describen las políticas a seguir para realizar las salvas, en pos de poder recuperar la información en cualquier momento ante cualquier adversidad. Se tratan un grupo de elementos de seguridad importantes, incluyendo dentro de ellos los elementos de auditoria existentes en el RDBMS seleccionado.

3.2 Desarrollo.

3.2.1 Estructuras de almacenamiento.

3.2.1.1 Los espacios de tablas (Tablespaces).

Un espacio de tablas es una división lógica de la base de datos. Cada base de datos tiene al menos uno (*SYSTEM*). Un espacio de tablas puede pertenecer sólo a una base de datos. Los espacios de tablas se utilizan para mantener juntos los datos de usuarios o de aplicaciones para facilitar su mantenimiento y/o mejorar las prestaciones del sistema.

De esta manera, cuando se crea una tabla se debe indicar el espacio de tablas al que se destina. Por defecto se depositan en el espacio de tablas *SYSTEM*, que se crea por defecto. Este espacio de tablas es el que contiene el diccionario de datos, por lo que conviene reservarlo para el uso del servidor, y asignar las tablas de usuario a otro.

Lo razonable y aconsejable es que cada aplicación tenga su propio espacio de tablas.

Hay varias razones que justifican este modo de organización de las tablas en espacios de tablas:

- ✓ Permiten distribuir a nivel lógico/físico los distintos objetos de las aplicaciones.
- ✓ Un espacio de tablas puede quedarse offline debido a un fallo de disco, permitiendo que el SGBD continúe funcionando con el resto.
- ✓ Los espacios de tablas pueden estar montados sobre dispositivos ópticos si son de sólo lectura.
- ✓ Son una unidad lógica de almacenamiento, pueden usarse para aislar completamente los datos de diferentes aplicaciones.
- ✓ Oracle permite realizar operaciones de *backup/recovery* a nivel de espacio de tabla mientras la base de datos sigue funcionando.

Cuando se crean se les asigna un espacio en disco que Oracle reserva inmediatamente, se utilice o no. Si este espacio inicial se ha quedado pequeño Oracle puede gestionar el crecimiento dinámico de los ficheros sobre los que se asientan los espacios de tablas. Esto elimina la posibilidad de error en las aplicaciones por fallos de dimensionamiento inicial. Los parámetros de crecimiento del tamaño de los espacios de tablas se especifican en la creación de los mismos.

Se pueden ver los espacios de tablas definidos en nuestra base de datos con el comando SQL siguiente:

*DB171> select * from user_tablespaces;*

3.2.1.2 Definición de los espacios de tablas para el sistema 171.

Para nuestro proyecto se propone crear 3 espacios de tablas, uno para las tablas más activas o consultadas, uno para los índices y otro para el resto de la información.

Separar las tablas más activas del resto de las tablas del sistema nos brinda una elevada velocidad de respuesta a las peticiones de acceso a los datos, esto se logra debido a la existencia de una menor cantidad de tablas en el espacio de tablas y recursos disponibles para realizar las operaciones de selección, inserción, actualización y eliminación.

Utilizar otro espacio de tablas para el almacenamiento de los índices, ya que estos son independientes a los objetos sobre los cuales actúan, tienen crecimiento independiente algo que es una ventaja porque no afecta el crecimiento de la tabla, pero que se convierte en un elemento importante a tener en cuenta por

los DBAs[2] a la hora de lograr optimización en el sistema porque puede afectar el almacenamiento del espacio de tablas y su funcionamiento.

El resto de la información se almacenará en un espacio de tablas base donde originalmente debió estar almacenado todo lo que si ha distribuido en este sistema.

Se propone que cada espacio de tabla quede almacenado en un disco diferente, esto nos brindará ventajas para la **disponibilidad**, **velocidad de respuesta** y **seguridad** de la información.

Condiciona **disponibilidad** de la información del sistema porque los espacios de tablas pueden quedar fuera de servicio (offline) por falta de capacidad de almacenamiento, por ejemplo: El espacio de tablas de los índices puede llenarse y quedar fuera de servicio que no afecta al de los datos y por tanto se puede continuar accediendo a la información.

Condiciona la **velocidad de respuesta** porque al estar la información en dispositivos diferentes las operaciones de entrada/salida sobre datos almacenados en diferentes espacios de tablas se realzarán con recursos físicos diferentes, la cantidad de información será menos en cada espacio de tablas algo que también brindara mayor velocidad de ejecución.

Condiciona la **seguridad** porque mejora las operaciones de backup y recuperación de los datos, ya que estas operaciones pueden ser hechas a nivel de espacios de tablas y dispositivos. Si alguno de los dispositivos se daña físicamente esto no afecta el funcionamiento del sistema ni en la disponibilidad del resto de los datos.

3.2.2 Seguridad del servidor.

El gestor de bases de datos oracle es reconocido en el mundo por sus mecanismos de seguridad. Se puede decir que utiliza varios métodos de seguridad de la información y a varios niveles.

[2] Data base administrator o administrador de base de datos.

3.2.2.1 Seguridad por usuarios.

Establece seguridad a nivel de cuentas de usuarios en el servidor, solamente autoriza a acceder a los datos a los usuarios que tengan cuenta en la base de datos de la que se quiere la información. Cada cuenta en el servidor se conforma de clave y nombre usuario y conjunto de datos almacenados bajo el nombre de perfil. La clave de las cuentas puede ser cambiada por el DBA o por el mismo usuario. La información asociada a las cuentas se almacena en el diccionario de datos del servidor, en una vista llamada dba_users. Para acceder a vistas como esta se necesitan un grupo de privilegios sobre esos objetos, algo que constituye otra forma de seguridad del servidor.

3.2.2.2 Seguridad a Objetos.

Los objetos de base de datos son igualmente de controlado acceso por el servidor para los usuarios. Estos autoriza la utilización de determinadas operaciones o comandos puedan ser ejecutados sobre objetos en el servidor. Oracle nos brida una utilidad que ayuda mucho a los administradores de base de datos y son los roles, estos no son más que la agrupación de un conjunto de privilegios que luego pueden ser asignado a los usuarios simplificando las tareas se asignación de permisos y control de acceso. Es posible controlar el acceso a los roles con un clave o password, además de que pueden ser activados y desactivados dinámicamente, constituyendo así otro elemento de seguridad en el servidor. Los roles y los privilegios son asignados con el comando GRANT y retirados con REVOKE. Los roles pueden tener privilegios sobre objetos, sobre otros roles y privilegios del sistema.

3.2.2.3 Privilegios del Sistema.

Los privilegios del sistemas tienen gran importancia y deben ser asignados de forma cuidadosa ya que puede darle el control del sistema al usuario que no lo debe llevar. Para ello se proponen la oración de roles que contengan estos privilegios tal y como se ha propuesto anteriormente. Los usuarios para poder iniciar una sesión en el servidor necesitan el privilegio del sistema CONNECT, además existen permisos sobre comandos como son CREATE TABLE, SELECT ANY TABLE y el rol RESOURCE es para aquellos que necesitan crear segmentos. El rol DBA asignado a un usuario le permite manejar todos los objetos y datos del servidor de base de datos. Los privilegios del sistema que existen nos permiten asignar privilegios de forma independiente según operaciones por ejemplo un usuario puede tener rol de CREATE TABLE y no tener ALTER TABLE.

3.2.2.4 Implementación de Seguridad.

Como parte de la implementación de la seguridad estableceremos un grupo de restricciones. En el sistema de software se ha diseñado una seguridad a nivel de roles y usuarios que permitan controlar el acceso a la información y las distintas operaciones del negocio. El servidor oracle, como se ha analizado en los párrafos anteriores, posee mecanismo de seguridad implementado sobre roles y usuarios, compatibles con los intereses de seguridad de la aplicación, de forma tal que asigna y deniega permisos sobre objetos del servidor. Aprovechando las ventajas que la unión de estos dos elementos de seguridad nos brindan se propone como solución de seguridad para la aplicación lo siguiente:

- ✓ crear una relación directa entre los roles de la aplicación y los roles en el gestor de base de datos.
- ✓ crear usuarios que respondan a los usuarios de la aplicación de forma tal que cada usuario de nuestro sistema sea un usuario en el servidor.
- ✓ garantizar que esos usuarios de base de datos tengan los roles equivalentes a los de la aplicación en el servidor y así de esta forma brindan autorización a la información necesaria para la realización de su deberes.

Los roles propuesto para el servidor son los mismo que los de la aplicación, seguidos de módulo al que representan. Los roles son:

- ✓ **Operador** (Módulo de Recepción de Llamadas)
- ✓ **Despachador** (Módulo de Despacho)
- ✓ **SupervisorOperador** (Módulo de Supervisión de Operadores)
- ✓ **SupervisorDespacho** (Módulo de Supervisión de Despachadores)
- ✓ **AdminRecursos** (Módulo de Administración y Control de Recursos)

Con la implementación de esta estrategia de roles se garantiza que los usuarios que inicien la aplicación además de tener todos los privilegios en el negocio tienen que tener todos los privilegios en el servidor, comenzando por tener permiso a iniciar una sesión en el servidor, hasta autorización a ejecutar un determinado procedimiento, tabla, paquete, vista, entre otros objetos que pueden ser creados como solución a una petición de datos desde la aplicación.

Cuando se logre la seguridad del gestor se debe implementar la seguridad del sistema operativo que lo soporta, en este caso Red Hat Linux Enterprise 4. Oracle utiliza una serie de ficheros a los que los

usuarios no tienen porque acceder de manera directa. Por ejemplo, los ficheros de datos o los de *redo Log* son escritos y leídos sólo por los procesos Oracle. Así, sólo los DBAs que han creado estos ficheros necesitan acceder directamente a ellos a nivel del sistema operativo, por tanto se propone la creación de un usuario oracle cuyo password lo dominen solamente los DBAs, aislar los servidores de forma que solamente los especialistas tengan acceso a ellos. Restringir el acceso en el servidor a nivel de sistema operativo utilizando un software firewall, que permita restringir el acceso desde la red de los IPs que solamente sean necesarios. Utilización de un software antivirus, para evitar la contaminación de virus informáticos, prohibir utilizar el servidor como servidor de fichero, dejar su trabajo exclusivamente para el servidor oracle.

3.2.2.5 Perfiles de Usuario

Los perfiles son una herramienta de control que nos brinda el servidor oracle que se utilizan para limitar la cantidad de recursos del sistema y de la BD disponibles para un usuario. Si no se definen perfiles para un usuario se utiliza el perfil por defecto, que especifica recursos ilimitados en todos los parámetros que se configuran.

Los recursos que pueden ser limitados vía perfil son los siguientes:

- ✓ SESSIONES_PER_USER: El número de sesiones concurrentes que un usuario puede tener en una instancia.
- ✓ CPU_PER_SESSION: El tiempo de CPU, en centenas de segundos, que una sesión puede utilizar.
- ✓ CONNECT_TIME: El número de minutos que una sesión puede permanecer activa.
- ✓ IDLE_TIME: El número de minutos que una sesión puede permanecer sin que sea utilizada de manera activa.
- ✓ LOGICAL_READS_PER_SESSION: El número de bloques de datos que se pueden leer en una sesión.
- ✓ LOGICAL_READS_PER_CALL: El número de bloques de datos que se pueden leer en una operación.
- ✓ PRIVATE_SGA: La cantidad de espacio privado que una sesión puede reservar en la zona de SQL compartido de la SGA.

- ✓ COMPOSITE_LIMIT: El número de total de recursos por sesión, en unidades de servicio. Esto resulta de un cálculo ponderado de CPU_PER_SESSION, CONNECT_TIME, LOGICAL_READS_PER_SESSION y PRIVATE_SGA, cuyos pesos se pueden variar con el comando ALTER RESOURCE COST.

Los perfiles establecen restricciones del password del usuario, algunas de las restricciones establecen lo siguiente:

- ✓ FAILED_LOGIN_ATTEMPTS: número máximo de intentos de login fallidos.
- ✓ PASSWORD_LIFE_TIME: tiempo máximo de valides de password.
- ✓ PASSWORD_REUSE_MAX: mínimo de password diferentes del antes de reutilizarse.
- ✓ PASSWORD_REUSE_TIME: número mínimo de días antes de volver a usar un password.
- ✓ PASSWORD_LOCK_TIME: numeró máximo de días de bloqueo después de fallos en los intentos de login.
- ✓ PASSWORD_GRACE_TIME: días para que expire el pass después de la advertencia.
- ✓ PASSWORD_VERIFY_FUNCTION: función de la verificación del password.

Los perfiles se pueden crear vía el comando CREATE PROFILE, y se pueden modificar con la sentencia ALTER PROFILE.

En general, el perfil por defecto debe ser adecuado para los usuarios normales; los usuarios con requerimientos especiales deberían tener perfiles especiales.

El sistema esta aún en proceso de definición y no es posible crear un perfil definitivo para los usuarios del proyecto, incluso pudieran ser más de uno, pero se aclara que con la utilización de perfiles de usuarios en el servidor oracle lograremos controlar un grupo de permisos. Por ejemplo algunos elementos a establecer es el número máximo de sesiones abiertas para un usuario en el servidor, para casi todos los usuarios es uno, ya que se necesita controlar que este autenticado en un solo módulo. Otro elemento seria el número máximo de intentos de autenticarse. Con la creación de ese o esos perfiles ajustados a nuestras necesidades se deja el control de estos parámetros al servidor de base de datos.

3.2.2.6 Auditoría de Seguridad

El sistema gestor de base de datos Oracle tiene la capacidad de auditar todas las acciones que tienen lugar en la base de datos. Se pueden auditar tres tipos de acciones:

- ✓ intentos de entrada en cuentas de la BD.
- ✓ accesos a los objetos de la BD.
- ✓ acciones sobre la BD.

El gestor es capaz de almacenar en la base de datos todos los intentos de acción, tanto los exitosos como los infructuosos, aunque es un parámetro configurable.

Esta opción del gestor de base de datos debe ser habilitada para que se controlen todas las acciones. Para habilitar la capacidad de auditoria, se debe fijar el parámetro AUDIT_TRAIL en el fichero init.ora, que se encuentra en el directorio raíz de la instalación del servidor. Los registros de auditoría se almacenan en la tabla SYS.AUD$ o bien su gestión se deja al SO. Cuando se decide utilizar la tabla SYS.AUD$ es importante que se revise periódicamente, por si hiciera falta truncarla debido a que su aumento de tamaño puede causar problemas de espacio en el *tablespace* SYSTEM, provocando que se dejen de realizar algunas operaciones o limitando el trabajo del gestor. En la tabla a continuación mostramos todos los parámetros de AUDIT_TRAIL:

Valor	***Descripción***
NONE	Deshabilita la auditoría
BD	Habilita la auditoría, escribiendo en la tabla SYS.AUD$.
OS	Habilita la auditoría, dejando al SO su gestión.

3.2.2.6.1 Auditando Conexiones

El gestor permite el almacenamiento de todos los intentos de conexión con el servidor de base. El comando utilizado para iniciar la auditoría es:

SVRMGR> audit session;

Se utilizan comandos que le permiten al usuario definir que tipos de acceso desea almacenar. Para determinar si se deben registrar sólo los éxitos, o sólo los fracasos se pueden utilizar los siguientes comandos:

DB171> audit session whenever successful;
DB171> audit session whenever not successful;

Como los registros de auditoría se almacenan en la tabla del diccionario de datos SYS.AUD$, entonces se pueden acceder a toda la información que contiene a través de la vista DBA_AUDIT_SESSION.

```
select
  os_username,                /* nombre de usuario SO */
  username,                   /* nombre de usuario BD */
  terminal,
  decode (returncode,'0','Conectado',
              '1005','Solo username, sin password',
              '1017','Password incorrecto',
               returncode),  /* comprobación de error */
  to_char (timestamp,'DD-MON-YY HH24: MI: SS'), /* hora de entrada */
  to_char (logoff_time,'DD-MON-YY HH24: MI: SS') /* hora de salida */
from dba_audit_session;
```

Para deshabilitar la auditoria de las conexiones basta con ejecutar la siguiente sentencia:

DB171> noaudit session;

3.2.2.6.2 Auditando Acciones

Una vez estando el usuario conectado con el servidor de base de datos se puede auditar cualquier acción que afecte a cualquier objeto de la base de datos. Para facilitar la gestión, las acciones a auditar se encuentran agrupadas según los grupos que se muestran en la siguiente tabla:

Grupo	**Comandos Auditados**
CLUSTER	Todas las sentencias que afecten a *clusters*.
DATABASE LINK	Todas las sentencias que afecten a enlaces de BD.
EXISTS	Todas las sentencias que fallen porque ya existe un objeto en la BD.
INDEX	Todas las sentencias que afecten a índices.
NOT EXISTS	Todas las sentencias que fallen porque un determinado objeto no existe.
PROCEDURE	Todas las sentencias que afecten a procedimientos.
PROFILE	Todas las sentencias que afecten a perfiles.
PUBLIC DATABASE LINK	Todas las sentencias que afecten a enlaces públicos de BD.
PUBLIC SINONYM	Todas las sentencias que afecten a sinónimos públicos.
ROLE	Todas las sentencias que afecten a roles.
ROLLBACK SEGMENT	Todas las sentencias que afecten a segmentos de *rollback*.
SEQUENCE	Todas las sentencias que afecten a secuencias.
SESSION	Todas las sentencias de acceso a la BD.
SYNONYM	Todas las sentencias que afecten a sinónimos.
SYSTEM AUDIT	Todas las sentencias AUDIT y NOAUDIT.
SYSTEM GRANT	Todas las sentencias afecten a privilegios.
TABLE	Todas las sentencias que afecten a tablas.
TABLESPACE	Todas las sentencias que afecten a espacios de tablas.

TRIGGER	Todas las sentencias que afecten a disparadores.
USER	Todas las sentencias que afecten a las cuentas de usuarios.
VIEW	Todas las sentencias que afecten a vistas.

Teniendo en cuenta todos estos grupos las operaciones se simplifican, por ejemplo, para auditar todas acciones que tienen que ver con las tablas sirve el siguiente comando:

DB171> audit table;

Y para deshabilitar la auditoría se utilizará el siguiente comando:

DB171> noaudit table;

La auditoría es más potente ya que podemos afinar un poco más en la auditoría fijando un usuario concreto al que seguir la pista de las acciones que realiza:

DB171> audit table by perez;

Cada acción auditada recibe un código numérico al que se puede acceder a través de la vista AUDIT_ACTIONS. Una vez que conocemos el código de la acción, podemos utilizarlo para determinar como dicha acción ha afectado a un objeto, consultado la vista DBA_AUDIT_OBJECT.

4.3 Auditando Objetos

Además de la auditoría de acciones sobre los objetos, se puede seguir el rastro a las operaciones de manipulación de tablas: SELECT, INSERT, UPDATE y DELETE. Estas auditorías se pueden hacer por sesión o por acceso.

Un ejemplo de sentencias de auditorías sobre objetos se puede ver en el siguiente grupo de sentencias:

DB171> audit insert on perez.emp;

DB171> audit all on perez.emp by session;
DB171> audit delete on perez.emp by access;

Los registros de auditoría se pueden ver en la misma vista DBA_AUDIT_OBJECT anteriormente mencionada.

3.2.2.6.3 Protegiendo los Registros de Auditoría

Los usuarios siempre tratan de limpiar todo el rastro de las operaciones que han hecho en los lugares por donde han pasado, es por eso que los registros de la tabla SYS.AUD$ pueden ser objeto de intentos de acceso para ser eliminados ya que pueden reflejar acciones no autorizadas en la base de datos. Así, resulta interesante reflejar ese tipo de acciones. Esto se consigue con el siguiente comando:

DB171> audit all on sys.aud$ by access;

De este modo cualquier acción contra la tabla SYS.AUD$ quedará registrado. Además, las acciones contra la tabla SYS.AUD$ sólo pueden ser borradas por los usuarios que puedan conectarse como INTERNAL.

3.2.3 Reglas Básicas de Backup

La capacidad de mantener un sistema constante de datos es probablemente la parte más importante de un trabajo de los administradores de la base de datos. En caso de un fallo del sistema, la necesidad de la recuperación de los datos se convierte en una prioridad superior. Sin un procedimiento de reserva definido, la capacidad de restaurar datos a un estado constante es obstaculizada seriamente o imposible. El propósito epígrafe es contornear los procedimientos de la reserva y de recuperación que se utilizan en lo que respecta a los SGBD Oracle.

Un *backup* válido es una copia de la información sobre la base de datos necesaria para reconstruirla a partir de un estado no utilizable de la misma. Normalmente, si la estrategia de *backup* se basa en la copia de los ficheros de datos y en el archivado de los ficheros *redo log ()*, se han de tener copias de los ficheros de datos, de los ficheros de control, de los ficheros *redo log* activos y también de los archivados. Si se

pierde uno de los ficheros *redo log* archivados se dice que se tiene un agujero en la secuencia de ficheros. Esto invalida el *backup*, pero permite a la BD ser llevada hasta el principio del agujero realizando una recuperación incompleta.

Antes de nada, es muy importante entender ciertas reglas que determinan la situación de los ficheros y otras consideraciones que afectarán al esquema de *backup*:

- ✓ Es recomendable archivar los ficheros *redo log []* en disco, y luego copiarlos a cinta, pero siempre en un disco diferente del que soporta los ficheros de datos y de *redo log []* activos.
- ✓ Los ficheros copias no deben estar en el mismo dispositivo que los originales. No siempre hay que pasar las copias a cinta, ya que si se dejan en disco se acelera la recuperación. Además, si se copian las copias a cinta y se mantienen en el disco, se puede sobrevivir a diversos fallos de dispositivo.
- ✓ Se deberían mantener diferentes copias de los ficheros de control, colocadas en diferentes discos con diferentes controladores.
- ✓ Siempre que la estructura de la BD cambie debido a la inclusión, borrado o renombrado de un fichero de datos o de *redo log*, se debe copiar el fichero de control, ya que almacenan la estructura de la BD. Además, cada fichero añadido también debe ser copiado. El fichero de control puede ser copiado mientras la BD está abierta con el siguiente comando:
  DB171> alter database backup controlfile to 'destino';

Teniendo en cuenta las reglas anteriores, los siguientes puntos pueden considerarse un ejemplo de estrategia de *backup*:

1. Activar el modo ARCHIVELOG.
2. Realizar un *backup* al menos una vez a la semana si la BD se puede parar. En otro caso, realizar *backups* en caliente cada día.
3. Copiar todos los ficheros *redo log* archivados cada cuatro horas. El tamaño y el número de ellos dependerá de la tasa de transacciones.
4. Efectuar un export de la BD semanalmente.

3.2.3.1 Backups Físicos

Los *backups* físicos son aquellos que copian físicamente los ficheros de la BD. Existen dos opciones: en frío y en caliente. Se dice que el *backup* es en frío cuando los ficheros se copian con la BD esta parada. En caliente es cuando se copian los ficheros con la BD abierta y funcionando.

3.2.3.2 Backup en Frío

El primer paso es parar la BD con el comando *shutdown normal*. Si la BD se tiene que parar con *inmediate* o *abort* debe rearrancarse con el modo RESTRICT y vuelta a parar en modo normal. Después se copian los ficheros de datos, los de *redo log* y los de control, además de los *redo log* archivados y aún no copiados.

Una buena idea es automatizar todo este proceso con los scripts correspondientes, de modo que no nos olvidemos de copiar ningún fichero.

Como este tipo de *backup* es una copia de los ficheros de la BD, si estos contienen algún tipo de corrupción, la traspasaremos a la copia de seguridad sin detectarla. Por esto es importante comprobar las copias de seguridad.

3.2.3.3 Backup en Caliente

Cuando la implantación de base de datos requiere disponibilidad de la misma 24h. al día, 7 días a la semana no se pueden realizar *backups* en frío. Para efectuar un *backup* en caliente debemos trabajar con la base de datos en modo ARCHIVELOG. El procedimiento de *backup* en caliente es bastante parecido al frío. Existen dos comandos adicionales: begin backup antes de comenzar y end backup al finalizar el *backup*. Por ejemplo, antes y después de efectuar un *backup* del *tablespace* users se deberían ejecutar las sentencias:

DB171> alter tablespace users begin backup;

DB171> alter tablespace users end backup;

Así como el *backup* en frío permitía realizar una copia de toda la base de datos al tiempo, en los *backups* en caliente la unidad de tratamiento es el *tablespace*. El *backup* en caliente consiste en la copia de los ficheros de datos (por *tablespaces*), el actual fichero de control y todos los ficheros *redo log* archivados creados durante el periodo de *backup*. También se necesitarán todos los ficheros *redo log* archivados después del *backup* en caliente para conseguir una recuperación total.

3.2.3.4 Backups Lógicos

Este tipo de *backups* copian el contenido de la base de datos pero sin almacenar la posición física de los datos. Se realizan con la herramienta *export* que copia los datos y la definición de la base de datos en un fichero en un formato interno de Oracle.

Para realizar un *export* la base de datos debe estar abierta. *Export* asegura la consistencia en la tabla, aunque no entre tablas. Si se requiere consistencia entre todas las tablas de la base de datos entonces no se debe realizar ninguna transacción durante el proceso de *export*. Esto se puede conseguir si se abre la base de datos en modo RESTRICT.

Entre las ventajas de efectuar un *export* están las siguientes:

- ✓ Se puede detectar la corrupción en los bloques de datos, ya que el proceso de *export* fallará.
- ✓ Protege de fallos de usuario, por ejemplo si se borra una fila o toda una tabla por error es fácil recuperarla por medio de un *import*.
- ✓ Se puede determinar los datos a exportar con gran flexibilidad.
- ✓ Se pueden realizar *exports* completos, incrementales y acumulativos.
- ✓ Los *backups* realizados con *export* son portables y sirven como formato de intercambio de datos entre base de datos y entre máquinas.

Pero realizar *backups* lógicos con *export* es que son mucho más lentos que los *backups* físicos.

3.2.3.5 Parámetros de Export

Parámetro	***Defecto***	***Descripción***
USERID	indefinido	El username/password del usuario que efectúa

		el *export*.
BUFFER	dependiente del SO	El tamaño en bytes del buffer utilizado.
FILE	expdat.dmp	El nombre del fichero destino.
GRANTS	Yes	Indica si se exportan también los derechos.
INDEXES	Yes	Indica si se exportan también los índices.
ROWS	Yes	Indica si se exportan también las filas de las tablas, o sólo las definiciones de las tablas.
CONSTRAINTS	Yes	Indica si se exportan también las restricciones.
COMPRESS	Yes	Indica si se exporta en modo comprimido.
FULL	No	Indica si se exporta la BD entera.
OWNER	usuario actual	Una lista de usuarios cuyos objetos se quieren exportar.
TABLES	indefinido	La lista de tablas a exportar.
RECORDLENGTH	dependiente del SO	La longitud en bytes del registro del fichero.
INCTYPE	indefinido	El tipo de *export* incremental.
RECORD	Yes	Indica si se anota el *export* incremental en las tablas SYS.INCVID y en SYS.INCEXP.
PARFILE	indefinido	El fichero de parámetros.

3.2.3.5.1 Modos de Export

Existen tres modos de realizar una exportación de datos:

Modo Tabla:

Exporta las definiciones de tabla, los datos, los derechos del propietario, los índices del propietario, las restricciones de la tabla y los disparadores asociados a la tabla.

Modo Usuario:

Exporta todo lo del modo de Tabla más los *clusters*, enlaces de base de datos, vistas, sinónimos privados, secuencias, procedimientos, etc. del usuario.

Modo de Base de Datos Entera:

Además de todo lo del modo Usuario, exporta los roles, todos los sinónimos, los privilegios del sistema, las definiciones de los *tablespaces*, las cuotas en los *tablespaces*, las definiciones de los segmentos de *rollback*, las opciones de auditoria del sistema, todos los disparadores y los perfiles.

El modo base de datos entera puede ser dividido en tres casos: Completo, Acumulativo e Incremental. Estos dos últimos se toman menos tiempo que el completo, y permiten exportar sólo los cambios en los datos y en las definiciones.

Completo

Exporta todas las tablas de la base de datos e inicializa la información sobre la exportación incremental de cada tabla. Después de una exportación completa, no se necesitan los ficheros de exportaciones acumulativas e incrementales de la base de datos anteriores.

Ver sentencias de ejemplo:

DB171> exp userid=system/manager full=y inctype=complete constraints=Y file=full_export_filename

Acumulativo

Exporta solo las tablas que han sido modificadas o creadas desde la última exportación Acumulativa o Completa, y registra los detalles de exportación para cada tabla exportada. Después de una exportación acumulativa, no se necesitan los ficheros de exportaciones incrementales de la base de datos anteriores.

Ver sentencias de ejemplo:

DB171> userid=system/manager full=y inctype=cumulative constraints=Y file=cumulative_export_filename

Incremental

Exporta todas las tablas modificadas o creadas desde la última exportación Incremental, Acumulativa o Completa, y registra los detalles de exportación para cada tabla exportada. Son interesantes en entornos en los que muchas tablas permanecen estáticas por periodos largos de

tiempo, mientras que otras varían y necesitan ser copiadas. Este tipo de exportación es útil cuando hay que recuperar rápidamente una tabla borrada por accidente.

Ver sentencias de ejemplo:

DB171> exp userid=system/manager full=y inctype=incremental constraints=Y file=incremental_export_filename

3.2.3.6 Política y planes de backup para el sistema 171.

La política de exportación puede ser la siguiente:

Realizar una exportación completa un día a la semana (por ejemplo el domingo), y luego realizar exportaciones incrementales el resto de la semana. De este modo de lunes a sábado sólo se exportarán aquellas tablas o estructuras que se hayan modificado, ahorrando tiempo y recursos en el proceso. Además tener al menos dos copias de todos los ficheros generados durante el proceso completo agrupados por fechas, en dispositivos de almacenamiento diferentes.

3.2.4 Principios de la Recuperación

La recuperación de los datos es el proceso de revertir tanto la estructura como los datos de la base de datos, en cualquier momento hacia un punto anterior en el tiempo. Teniendo así la posibilidad de recuperarse frente a los distintos problemas que puedan provocar la caída del servidor.

3.2.4.1 Definiciones y Conceptos

Para entender los principios de la recuperación, se necesita entender las estructuras de datos subyacentes utilizadas en la recuperación.Los ficheros *redo log* contienen los cambios realizados sobre la base de datos. Conviene presentar algunos conceptos relacionados con ellos.

Vector de Cambio

Describe un cambio simple en un bloque de datos de la base de datos. Entre otros datos, contiene el número de versión, el código de la transacción, y la dirección del bloque afectado.

System Change Number, SCN

Es un dato que define la versión confirmada de la base de datos en este instante de tiempo. Cuando una transacción es confirmada, se le asigna un SCN que la identifica unívocamente. Los ficheros *redo log* son marcados con dos SCN. Cuando se abre un nuevo fichero *redo log* se le

marca con un SCN, *low SCN*, que es uno más que el SCN mayor del anterior fichero *redo log*; y su *high SCN* es puesto a infinito. Los SCN también se asocian al fichero de control, ya que cuando se para una base de datos, un *tablespace* o fichero de datos, se almacena para cada fichero de datos su *stop SCN* en el fichero de control.

Registro Redo log

Es un conjunto de vectores de cambio que describen un cambio atómico sobre la BD. La transacción es también la unidad de recuperación.

Evolución de Redo log por día

Se puede calcular ejecutando el comando archive log list en dos días consecutivos y calculando la diferencia del número de secuencia de los ficheros *redo log*, multiplicado por el tamaño de un fichero *redo log*:

DB171> archive log list;

Database log mode No Archive Mode

Automatic archival Disabled

Archive destination /opt/app/oracle/admin/demo/arch/arch.log

Oldest online log sequence 3

Current log sequence 5

Cambio de redo log

Es el proceso mediante el cual se deja de utilizar un fichero *redo log* y el LGWR cambia al siguiente fichero *redo log* disponible. Se puede hacer con el comando alter system switch logfile.

Checkpoints

Son activados automáticamente durante el funcionamiento normal de la instancia, pero pueden ser activados manualmente con el comando alter system checkpoint local o alter system checkpoint global dependiendo si nos referimos a la instancia en la que estamos, o si queremos que afecte a todas las instancias activas, respectivamente.

Métodos de Recuperación

Existen varios métodos de recuperación, pero todos ellos se basan en la aplicación de los registros de *redo log*.

3.2.4.2 Recuperación Física

La utilización de una copia de *backup* de ficheros de datos siempre necesita de una recuperación física. También es así cuando un fichero de datos se pone *offline* sin un *checkpoint*.

Oracle detecta que se necesita una recuperación física cuando el contador de *checkpoints* de la cabecera del fichero de datos no coincide con el correspondiente contador de *checkpoints* del fichero de control. Entonces se hace necesario el comando recover. La recuperación comienza en el SCN menor de los ficheros de datos en recuperación, aplicando los registros de *redo log* a partir de él, y parando en el SCN de final mayor de todos los ficheros de datos.

Existen tres opciones para realizar una recuperación física. La primera es una recuperación de base de datos donde se restaura entera. La segunda es una recuperación de *tablespace* donde, mientras una parte de la BD está abierta, se puede recuperar un *tablespace* determinado. Esto significa que serán recuperados todos los ficheros de datos asociados al *tablespace*. El tercer tipo es la recuperación de un fichero de datos específico mientras el resto de la BD está abierta.

3.2.4.3 Requisitos para Utilizar Recuperación Física

La primera condición que se ha de poner para poder recuperar físicamente una BD es que ésta se esté utilizando en modo ARCHIVELOG. De otro modo, una recuperación completa puede que no sea posible. Si trabajamos con la BD en modo NOARCHIVELOG, y se hace una copia semanal de los ficheros de la BD, se debería estar preparado para perder, en el peor de los casos, el trabajo de la última semana si sucede un fallo. Ya que los ficheros de *redo log* contendrían un agujero y no se podía avanzar la BD hasta el instante anterior al fallo. En este caso el único medio para reconstruir la BD es hacerlo desde un *export* completo, recreando el esquema de la BD e importando todos los datos.

3.2.4.4 Recuperación de la BD

La BD debe estar montada pero no abierta. El comando de recuperación es el siguiente:

RECOVER [AUTOMATIC] [FROM 'localizacion'] [BD]
[UNTIL CANCEL]
[UNTIL TIME fecha]
[UNTIL CHANGE entero]
[USING BACKUP CONTROLFILE]

Las opciones entre corchetes son opcionales:

- ✓ AUTOMATIC hace que la recuperación se haga automáticamente sin preguntar al DBA por el nombre de los ficheros redo log. También se puede utilizar para este cometido el comando set autorecovery on/off. Los ficheros redo log deben estar en la localización fijada en LOG_ARCHIVE_DEST y el formato del nombre de los ficheros debe ser el fijado en LOG_ARCHIVE_FORMAT.
- ✓ FROM se utiliza para determinar el lugar donde están los ficheros redo log, si es distinto del fijado en LOG_ARCHIVE_DEST.
- ✓ UNTIL sirve para indicar que se desea realizar una recuperación incompleta, lo que implica perder datos. Solo se dará cuando se han perdido redo log archivados o el fichero de control. Cuando se ha realizado una recuperación incompleta la base de datos debe ser abierta con el comando alter database open resetlogs, lo que produce que los redo log no aplicados no se apliquen nunca y se inicialice la secuencia de redo log en el fichero de control. Existen tres opciones para parar la recuperación:
- ✓ UNTIL CANCEL permite recuperar un redo log cada vez, parando cuando se teclea CANCEL.
- ✓ UNTIL TIME permite recuperar hasta un instante dado dentro de un fichero de redo log
- ✓ UNTIL CHANGE permite recuperar hasta un SCN dado.
- ✓ USING BACKUP CONTROLFILE utiliza una copia de seguridad del fichero de control para gobernar la recuperación.

3.2.4.5 Recuperación de un tablespace

La BD debe estar abierta, pero con el *tablespace* a recuperar *offline*. El comando de recuperación es el siguiente:

RECOVER [AUTOMATIC] [FROM 'localización']
TABLESPACE nombre_tablespace [, nombre_tablespace]

3.2.4.6 Recuperación de un Fichero de Datos

La BD debe estar abierta o cerrada, dependiendo del fichero a recuperar. Si el fichero a recuperar es de un *tablespace* de usuario la base de datos puede estar abierta, pero con el fichero a recuperar *offline*. Si el fichero es del *tablespace* SYSTEM la base de datos debe estar cerrada, ya que no puede estar abierta con los ficheros del SYSTEM *offline*. El comando de recuperación es el siguiente:

RECOVER [AUTOMATIC] [FROM 'localización']
DATAFILE nombre_fichero [, nombre_fichero]

3.2.4.7 Creando un Fichero de Control

Si el fichero de control ha resultado dañado y se ha perdido se puede utilizar una copia de seguridad del mismo o crear uno nuevo. El comando de creación de un nuevo fichero de control es *CREATE CONTROLFILE*. Este comando se puede ejecutar sólo con la base de datos en estado *nomount*. La ejecución del comando produce un nuevo fichero de control y el montaje automático de la base de datos.

Un comando interesante que ayuda a mantener los ficheros de control a salvo es el siguiente:

DB171> alter database backup controlfile to trace;

Este produce un *script* que puede ser utilizado para generar un nuevo fichero de control y recuperar la BD, en caso necesario.

3.2.4.8 Recuperación Lógica

Oracle dispone de la herramienta *import* para restaurar los datos de una BD a partir de los ficheros resultados de un *export*. *Import* lee los datos de los ficheros de exportación y ejecuta las sentencias que almacenan creando las tablas y llenándolas de datos.

3.2.4.9 Parámetros del Import

Parámetro	***Defecto***	***Descripción***
USERID	indefinido	El username/password del usuario que efectúa el *import*.
BUFFER	dependiente del SO	El tamaño en bytes del buffer utilizado.
FILE	expdat.dmp	El nombre del fichero de exportación a importar.
SHOW	No	Indica si se muestran los contenidos del fichero de exportación, sin importar ningún dato.
IGNORE	Yes	Indica si ignorar los errores producidos al importar un objeto que ya existe en la BD.
GRANTS	Yes	Indica si se importan también los derechos.
INDEXES	Yes	Indica si se importan también los índices.
ROWS	Yes	Indica si se importan también las filas de las tablas.
FULL	No	Indica si se importan el fichero entero.
FROMUSER	Indefinido	Una lista de los usuarios cuyos objetos se han exportado.
TOUSER	Indefinido	Una lista de los usuarios a cuyo nombre se importan los objetos.
TABLES	indefinido	La lista de tablas a importar.
RECORDLENGTH	dependiente del SO	La longitud en bytes del registro del fichero.

INCTYPE	indefinido	El tipo de *import* incremental (SYSTEM o RESTORE).
COMMIT	No	Indica si se efectúa un commit después de importar cada fila. Por defecto, *import* efectúa un commit después de cargar cada tabla.
PARFILE	indefinido	El fichero de parámetros.

Los planes de recuperación esta sujetos a problemas en un entorno de producción real, ya que estos son aplicables cuando ha surgido un problema, mientras tanto se deben realizar pruebas sistemáticas de las salvas que se han ido haciendo en los backups, para verificar la integridad de las mismas, y prevenir cualquier incidencia sobre estos ficheros, como pueden ser corrupción de los ficheros, deterioro de los dispositivos de almacenamiento entre otros.

3.3 Conclusiones.

En este capítulo se han definido algunas de las actividades a realizar para llevar a cabo la gestión y configuración en el servidor de base de datos de forma tal que permita obtener un sistema estable con una alta disponibilidad y una buena integración. También se estudió la capacidad del servidor Oracle para prevenir fallos haciendo copias de sus estructuras y datos, así como la recuperación completa del servidor.

Conclusiones.

Con este trabajo se ha realizado un estudio del funcionamiento de los centros 171 enfocándolo desde el punto de vista del diseñador de base de datos, que es una parte fundamental en el flujo de los datos así como la seguridad, integridad y disponibilidad de los mismos. Además se ha dado parcial cumplimiento a los objetivos planteados obteniéndose los siguientes resultados:

- ✓ Descripción de algunos de los principales módulos del sistema 171.
- ✓ Esquemas de configuración del servidor de base de datos.
- ✓ Esquemas de seguridad del servidor de base de datos.
- ✓ Plan de políticas de salvas y restauración del sistema.

Con el diseño de base de datos realizado se solucionaron las problemáticas presentadas por los diferentes módulos del Sistema de Gestión de Emergencia de Seguridad Ciudadana (171) SIGESC. Garantizando el almacenamiento, integridad y seguridad de la información con el gestor de bases de datos seleccionado.

Recomendaciones.

Se recomienda continuar con el análisis y diseño de base de datos de los módulos que no se presentan en el trabajo, **Módulo de Configuración de Operaciones**, **Módulo de Supervisión General**, **Módulo de Mapa**, **Módulo de AVL**, y extender hasta estos conceptos como la supervisión a nivel de puntos y no de usuarios, entre otros. Además continuar las recomendaciones siguientes:

- ✓ Distribuir el diseño de creación dinámico de entidades en aras de detectar con mayor rapidez problemas de rendimiento en las operaciones DML y DDL y problemas de seguridad. Fomentando el desarrollo colaborativo en la comunidad de desarrolladores de la universidad.
- ✓ Estudiar la solución de Servicios Web que brinda el gestor de bases de datos propuesto, para el intercambio de información entre Sistemas 171.
- ✓ Estudiar las arquitecturas de servidores para garantizar desde el punto de vista del hardware la seguridad y disponibilidad de los datos, un rendimiento óptimo del servidor y flexibilidad a la hora de realizar los backup sin detener el servicio.
- ✓ Integrar el sistema de auditoría que brinda el gestor de base de datos, con la solución de auditoría que se propone en el Sistema de Gestión de Emergencias de Seguridad Ciudadana.

Referencias Bibliográficas.

[1] Declaración Universal de los Derechos humanos 1948, 2006. Fecha de acceso: 20-10-2006. Disponible en: http://www.un.org/spanish/aboutun/hrights.htm

[2] Constitución de la República Bolivariana de Venezuela 1999, 2006 Fecha de acceso: 24-11-2006. Disponible en: http://www.tsj.gov.ve/legislacion/constitucion1999.htm

[3] Base de datos, 2006. Fecha de acceso: 16-04-2007. Disponible en: http://es.wikipedia.org/wiki/Base_de_datos.

[4] Corabel, S. (2004) "Manejadores de Base de Base de Datos - SQL, ORACLE, INFORMIX" Fecha de acceso: 24-04-2007, disponible en:

http://www.ilustrados.com/publicaciones/EpZVVlyFyAbRDtMKhI.php

[5] Del Valle Brito, A. (2006) "Oracle". Fecha de acceso: 10-05-2007 Disponible en: http://www.monografias.com/trabajos25/oracle/oracle.shtml#mejoras

[6] "Sitio Oficial de la Dirección General de Seguridad y Emergencias Canarias". Fecha de acceso: 20-05-2007. Disponible en: http://www.gobcan.es/dgse/evo/cecoes.html

[7] "Sitio Oficial de Centro Automático de Despacho – CAD, 2007, Santiago de Cali". Fecha de acceso: 20-05-2007. Disponible en:
http://www.cali.gov.co/modules.php?op=modload&name=Corporativo&file=index&id=626

[8] "Gobernación del estado bolívar. Secretaría de seguridad ciudadana. Emergencias bolívar 1-7-1, 2007, Venezuela". Fecha de acceso: 20-05-2007. Disponible en:
http://www.e-171.gob.ve/3srv31.php

[9] Servicio 066, 2007, Coahuila, México. Fecha de acceso: 20-05-2007. Disponible en: http://ssp.sfpcoahuila.gob.mx/modulo11.asp?Id_Contenido=12

[10] Trejo, Janhil Aurora (2006) "Base de datos". Fecha de acceso: 25-05-2007. Disponible en: http://www.monografias.com/trabajos11/basda/basda.shtml

[11] SQL, 2007. Fecha de acceso: 23/06/2007. Disponible en: http://es.wikipedia.org/wiki/SQL

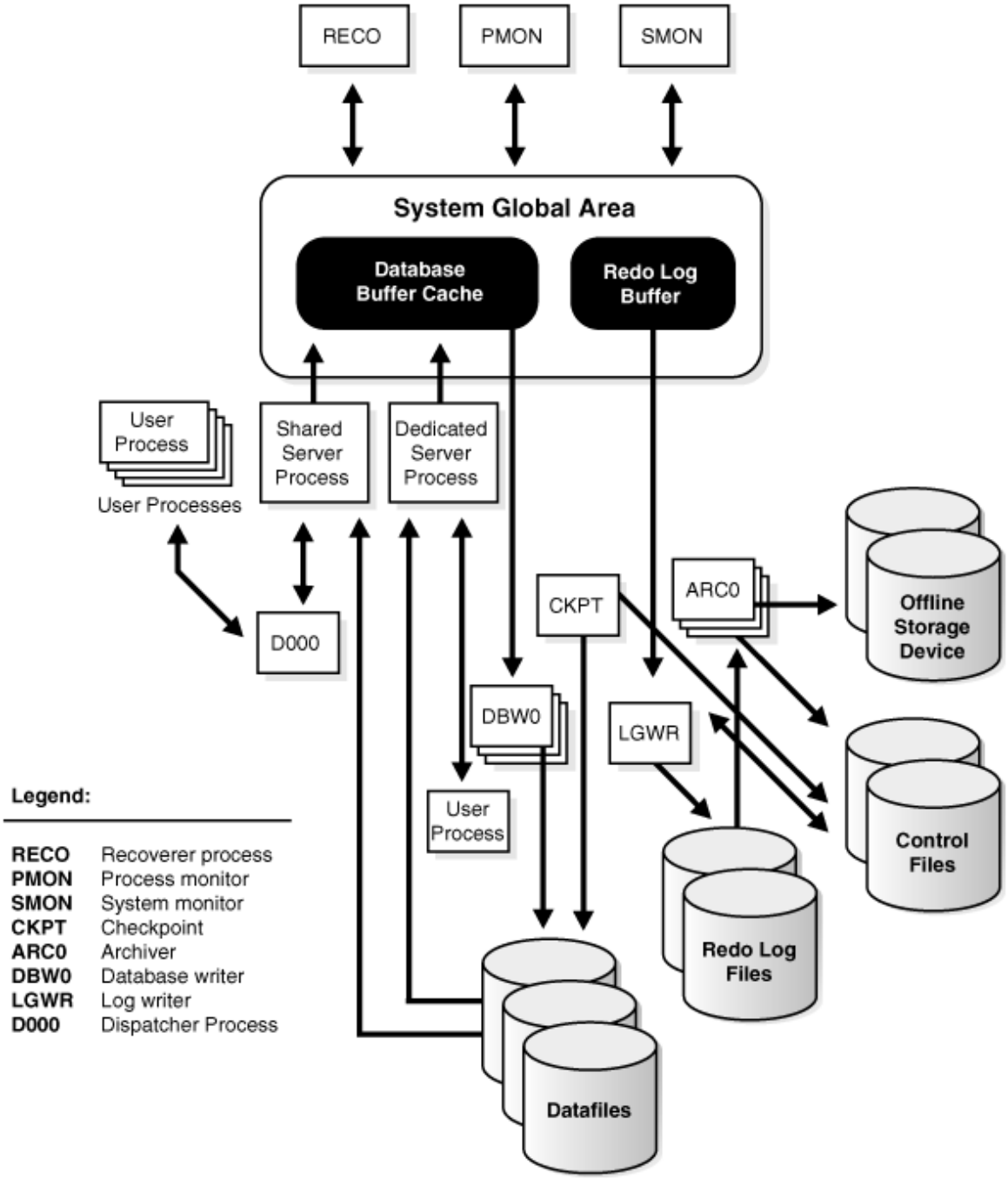
Bibliografía.

1. Grupo Asesor del Convenio de Colaboración Cuba Venezuela, *Proyecto Técnico Económico para la creación del Centro de Gestión de Emergencias y Seguridad Ciudadana 171 de la República Bolivariana de Venezuela*., 2006. p 52.
2. ORACLE. *Oracle Ibérica*, [Consultado el: 12/1/2007]. Disponible en: http://www.oracle.com/global/es/index.html
3. ORACLE. *Oracle Spatial & Oracle Locator: Location Features for Oracle Database 10g*, [Consultado el: 12/4/2006]. Disponible en: http://www.oracle.com/technology/products/spatial/index.html
4. Oracle Database 2 Day DBA 10g Release 1 (10.1) June 2004. [Consultado el: 19/4/2007]. Disponible en: http://download-east.oracle.com/docs/cd/B14117_01/server.101/b10742.pdf
5. Oracle Database Administrator's Guide 10g Release 1 (10.1) December 2003. [Consultado el: 20/4/2007]. Disponible en: http://download-east.oracle.com/docs/cd/B14117_01/server.101/b10739.pdf
6. Oracle Database SQL Reference 10g Release 1 (10.1) December 2003. [Consultado el: 23/4/2007]. Disponible en: http://download-east.oracle.com/docs/cd/B14117_01/server.101/b10759.pdf
7. Oracle Database Recovery Manager Reference 10g Release 1 (10.1) Junio 2004. [Consultado el: 26/4/2007]. Disponible en: http://download-east.oracle.com/docs/cd/B14117_01/server.101/b10770.pdf
8. Descripción de Oracle, 2006. [Consultado el: 2/5/2007]. Disponible en: http://www.monografias.com/trabajos25/oracle/oracle.shtml#mejoras
9. Oracle, 2006. [Consultado el: 3/5/2007]. Disponible en: http://www.alegsa.com.ar/Diccionario/diccionario.php
10. Base de datos, 2006. [Consultado el: 3/5/2007]. Disponible en: http://es.wikipedia.org/wiki/Base_de_datos

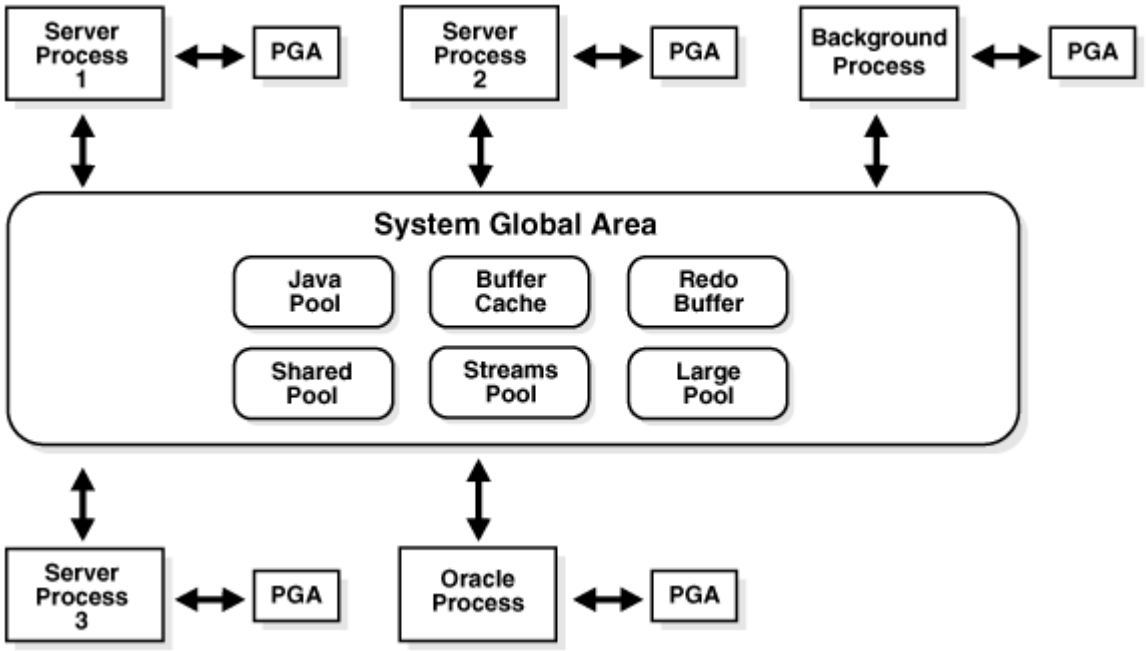
11. URMAN, S. *Oracle 8i. Programación avanzada con PL-SQL*, 2001. [Consultado el: 3/5/2007]. Disponible en: http://bibliodoc.uci.cu/pdf/reg01431.pdf
12. Grid Computing, 2006, [Consultado el: 3/5/2007]. Disponible en: http://es.wikipedia.org/wiki/Grid_computing, wikipedia.org, 16 may. 07.

Anexos.

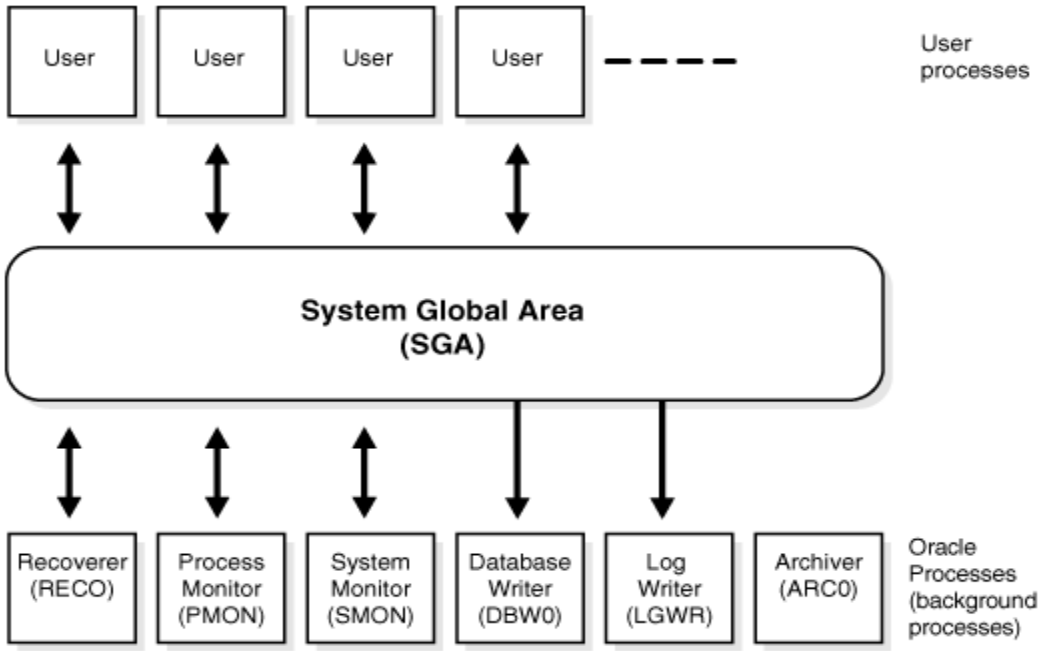
Anexo 1, Procesos de Background en Oracle Múltiple-Proceso.



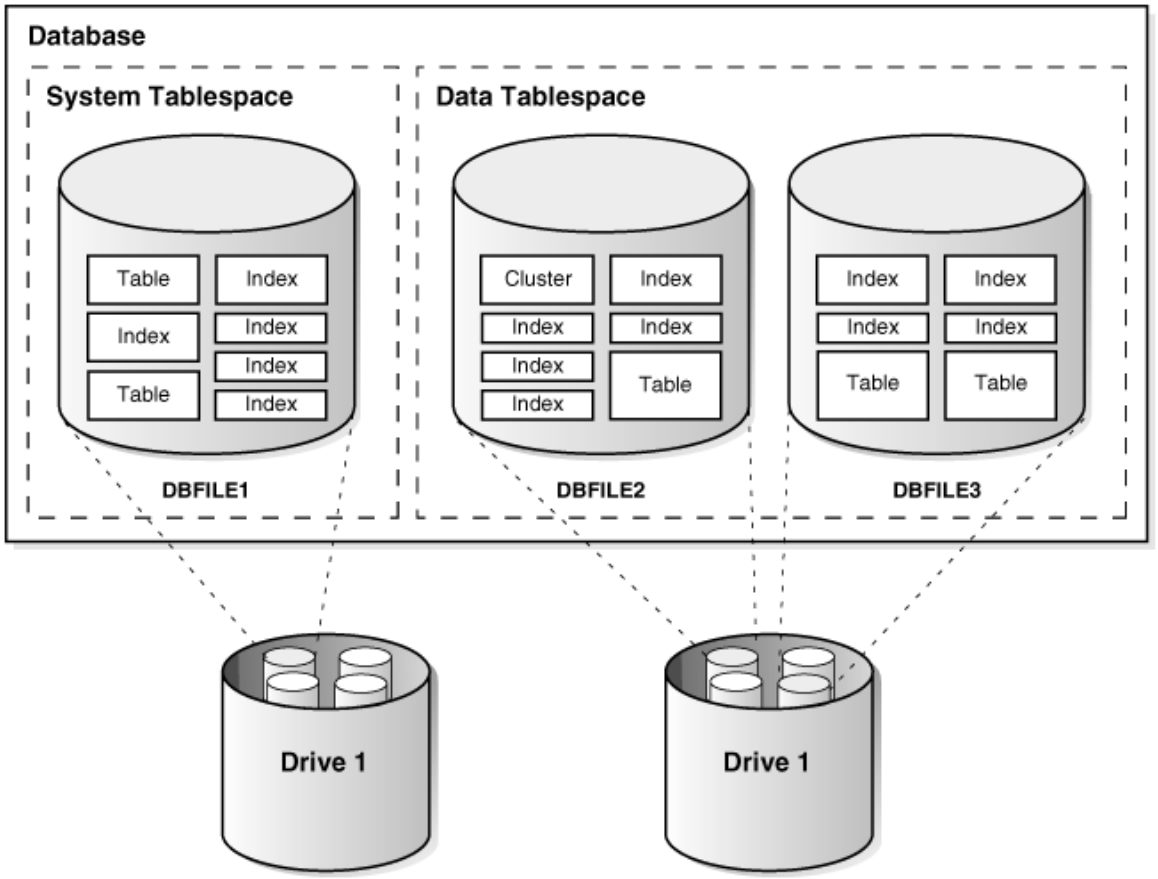
Anexo 2, Distribución de la memoria.



Anexo 3, Distribución de los procesos.



Anexo 4, Objetos, Tablespaces, y Datafiles del esquema.



Glosario de Términos.

Órganos de seguridad ciudadana: Entidades del estado que velan por el bienestar social de cada ciudadano.

- ✓ Las Policías de cada Estado.
- ✓ Las Policías de cada Municipio, y los servicios mancomunados de policías prestados a través de la Policía Metropolitana.
- ✓ El Cuerpo de Investigaciones Científicas, Penales y Criminalísticas.
- ✓ El Cuerpo de Bomberos y Administración de Emergencias de Carácter Civil.
- ✓ La Organización de Protección Civil y Administración de Desastre.

BD: Base de Datos.

PC: Computador Personal.

BDG: Base de Datos Geográfica.

Centro 171: Centro de Gestión de Emergencia de Seguridad Ciudadana (171).

CU: Casos de Uso.

TNS: (Transparent Network Substrate, Sustrato de red transparente). Permite la comunicación entre los clientes y los servidores de bases de datos Oracle con independencia del protocolo de comunicaciones que se utilice.

Tablespace (espacio de tablas): Una base de datos se divide en unidades lógicas denominadas **TABLESPACES**. Un tablespace no es un fichero físico en el disco, simplemente es el nombre que tiene un conjunto de propiedades de almacenamiento que se aplican a los objetos (tablas, secuencias…) que se creen en la base de datos bajo el tablespace indicado (tablas, secuencias…).

Datafile (fichero de datos): Un datafile es la representación física de un tablespace. Son los "ficheros de datos" donde se almacena la información físicamente. Un datafile puede tener cualquier nombre y extensión (siempre dentro de las limitaciones del SO), y puede estar localizado en cualquier directorio del disco duro, aunque so localización típica suele ser $ORACLE_HOME/Database. Un datafile tiene un tamaño predefinido en su creación (por ejemplo 100Mb) y este puede ser alterado en cualquier momento.

Segment (segmento, trozo, sección): Un segment es aquel espacio direccionado por la base de datos dentro de un datafile para ser utilizado por un solo objeto. Así una tabla (o cualquier otro objeto) está dentro de su segment, y nunca podrá salir de él, ya que si la tabla crece, el segment también

crece. Físicamente, todo objeto en base de datos no es más que un segment (segmento, trozo, sección) en un datafile.

Extent (extensión): Para cualquier objeto de base de datos que tenga cierta ocupación en disco, es decir, cualquier objeto que tenga un segment relacionado, existe el concepto de extent. Extent es un espacio de disco que se direcciona de una sola vez, un segment que se direcciona en un momento determinado de tiempo. El concepto de extent es un concepto físico, unos extent están separados de otros dentro del disco.

Data block (bloque de datos): Un data block es el último eslabón dentro de la cadena de almacenamiento. El concepto de Data block es un concepto físico, ya que representa la mínima unidad de almacenamiento que es capaz de manejar Oracle. Igual que la mínima unidad de almacenamiento de un disco duro es la unidad de asignación, la mínima unidad de almacenamiento de Oracle es el data block.

Grid Computing

Es una tecnología innovadora que permite utilizar de forma coordinada todo tipo de recursos (entre ellos cómputo, almacenamiento y aplicaciones específicas) que no están sujetos a un control centralizado. En este sentido es una nueva forma de computación distribuida, en la cual los recursos pueden ser heterogéneos (diferentes arquitecturas, supercomputadores, clusters...) y se encuentran conectados mediante redes de área extensa (por ejemplo Internet). Desarrollado en ámbitos científicos a principios de los años 1990, su entrada al mercado comercial siguiendo la idea de la llamada Utility computing supone una revolución que dará mucho que hablar.

El término grid se refiere a una infraestructura que permite la integración y el uso colectivo de ordenadores de alto rendimiento, redes y bases de datos que son propiedad y están administrados por diferentes instituciones. Puesto que la colaboración entre instituciones envuelve un intercambio de datos, o de tiempo de computación, el propósito del grid es facilitar la integración de recursos computacionales. Universidades, laboratorios de investigación, empresas, etc., se asocian para formar grid para lo cual utilizan algún tipo de software que implemente este concepto.

Las características de esta arquitectura serían:

- ✓ Capacidad de balanceo de sistemas: no habría necesidad de calcular la capacidad de los sistemas en función de los picos de trabajo, ya que la capacidad se puede reasignar desde la granja de recursos a donde se necesite.

- ✓ Alta disponibilidad. Con la nueva funcionalidad, si un servidor falla, se reasignan los servicios en los servidores restantes.
- ✓ Reducción de costes: Con esta arquitectura los servicios son gestionados por "granjas de recursos". Ya no es necesario disponer de "grandes servidores" y podremos hacer uso de componentes de bajo coste. Cada sistema puede ser configurado siguiendo el mismo patrón.

4GL:

No existe consenso sobre lo que es un *lenguaje de cuarta generación* (4GL). Lo que en un lenguaje de tercera generación (3GL) como COBOL requiere cientos de líneas de código, tan solo necesita diez o veinte líneas en un 4GL. Comparado con un 3GL, que es procedural, un 4GL es un lenguaje no procedural: el usuario define qué se debe hacer, no cómo debe hacerse. Los 4GL se apoyan en unas herramientas de mucho más alto nivel denominadas herramientas de cuarta generación. El usuario no debe definir los pasos a seguir en un programa para realizar una determinada tarea, tan sólo debe definir una serie de parámetros que estas herramientas utilizarán para generar un programa de aplicación. Se dice que los 4GL pueden mejorar la productividad de los programadores en un factor de 10, aunque se limita el tipo de problemas que pueden resolver. Los 4GL abarcan:

- ✓ Lenguajes de presentación, como lenguajes de consultas y generadores de informes.
- ✓ Lenguajes especializados, como hojas de cálculo y lenguajes de bases de datos.
- ✓ Generadores de aplicaciones que definen, insertan, actualizan y obtienen datos de la base de datos.
- ✓ Lenguajes de muy alto nivel que se utilizan para generar el código de la aplicación.

Los lenguajes SQL y QBE son ejemplos de 4GL

La instancia

La instancia es la unión de los procesos y de las estructuras de memoria (base de datos), los cuales se hallan en ejecución para el acceso de los usuarios a los datos a través de diferentes aplicaciones como por ejemplo administración, desarrollo y otras aplicaciones de usuario final. También hacen parte de la instancia distintos tipos de procesos: procesos usuario, procesos servidor y procesos de

fondo (background process). Entre ellos cuentan PMON (Monitor de procesos), SMON (Monitor del sistema), DBWR (Proceso de escritura), LGWR (Proceso de registro de operaciones), CKPT (Proceso de Checkpoint).

SGA (System Global Area)

Esta estructura de memoria es creada cuando la instancia se arranca y consiste principalmente de estructuras de memoria más pequeñas y con funcionalidades específicas entre las cuales se encuentran las siguientes estructuras obligatorias:

CTAISC: Centro de Tratamiento y Análisis de Información de Seguridad Ciudadana.