



# UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

## FACULTAD 2

### TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS INFORMÁTICAS

Título: Diseño y Configuración de la Base de Datos del Sistema de Atención a Emergencias del Centro de Información y Mando de la Unidad Provincial de Patrullas.

Autores: Abel Alberto Pérez González.  
Lázaro César Sánchez Vázquez.

Tutores: Ing. José Rolando Lafaurié Olivares.  
Ing. Rasiel Aponcio Borges.

La Habana, Cuba. 2012  
"Año 54 de la Revolución"

# *Declaración de Autoría*

---

## DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los 19 días del mes de junio del año 2012.

\_\_\_\_\_  
Abel Alberto Pérez González

Autor

\_\_\_\_\_  
Lázaro César Sánchez Vázquez

Autor

\_\_\_\_\_  
José Rolando Lafaurié Olivares

Tutor

\_\_\_\_\_  
Rasiel Aponcio Borges

Tutor

# *Dedicatoria*

---

*La presente realización se la dedico a mis padres, Alberto y Anabel, y a mi segundo papá, René, pues son las personas que además de mí, siempre soñaron con este momento. Por el cariño, respeto y confianza que siempre me han depositado,*

*A ellos, con eterno reconocimiento, se la dedico,*

*Abel Alberto Pérez González.*

# Dedicatoria

---

*Le dedico este título a mi familia. Siempre atenta y exigente. Tanto, mis padres, tíos, abuelos, primos y hermanos, más que darme aliento con un “adelante que te ayudaremos”, se han evitado ese proceso protocolar de hacer promesas y verdaderamente me han ayudado. Nunca de su parte faltó el “¿cómo van los estudios?”, “¿tienes problemas con alguna asignatura?”, “mi amigo fulano tiene a su hijo(a) que es profesor allá, llámalo y dile que te ayude en tal o mas cual asignatura”, “¿niño, qué te hace falta?”, “niño, cualquier cosa que necesites, dílo con tiempo”, “deja eso para después y estudia”, “vamos, que te llevaré a casa de un profesor que fue alumno mío y sabe cantidad de esa asignatura”. Es preciso que vuelva a mencionar aquí a mis padres porque no han sido pocos los dolores de cabeza que les he dado, ni las veces que han demostrado mucho interés en escuchar mis criterios, llevándolos a análisis aunque reconozco que muchas veces me dejé llevar por la exaltación del momento antes de hablar.*

*Lázaro César Sánchez Vázquez.*

# Agradecimientos

---

*Para la realización de este trabajo fue indispensable el apoyo de muchas personas a las cuales quiero agradecer:*

*En primer lugar a mis padres, Alberto y Anabel, y a mi segundo papá, René, por apoyarme siempre, educarme y enseñarme la importancia de ser un profesional.*

*A Maovys, por estar presente en todo momento y hacer hasta lo imposible por ayudarme siempre.*

*A mis compañeros de aula, por compartir sus conocimientos y experiencias, en especial a aquellos (as) que durante la carrera fueron como hermanos (as) para mí.*

*A todos los profesores que contribuyeron, con su entrega, dedicación y labor de cada día, a que hoy esté en esta etapa de un camino que sólo está comenzando.*

*A mis tutores, por su apoyo en el trabajo diario y por ser guías fundamentales en la realización de esta investigación.*

*A mi compañero de tesis, por su participación en la elaboración conjunta de este trabajo.*

*Y en general, a los miembros del proyecto y otras personas involucradas que de una forma u otra asistieron el desarrollo de este trabajo.*

*A todos, muchas gracias,*

*Abel Alberto Pérez González.*

# Agradecimientos

---

*Debo comenzar agradeciendo a mis amigos, tanto los que he tenido desde niño, que por su puesto, ahora permanecen para dar y aceptar consejos además de compartir gustos y desacuerdos, como los que han nacido en el devenir de estos cinco años de estudios. Ellos han sido mi apoyo en todo momento clave que los he necesitado. Inolvidables para mí serán también, muchos profesores que desde primer año me han ido enseñando cómo solucionar los problemas, ya sean de la vida como del estudio. He conocido a muchas personas de las que he aprendido y aunque no los tenga como mis amigos ni sean profesores, es meritorio mencionarlos.*

*También está el alto mando, es decir, mis jefes, tanto cadetes como Oficiales, que han sabido escuchar las problemáticas que les he planteado y me han dado siempre, una posible solución a las mismas, confiando en mi responsabilidad y buen juicio para tomar las decisiones finales.*

*También se han colado en mi vida un grupo de personas que están dispuestas a convertirse en parte de mi familia y ellos son los padres, tíos, primos y abuelos de mi novia, además de mi propia novia claro. Ellos han tenido en mí una confianza inmensa al punto de permitirme robarles su tesoro. Estoy seguro que mi novia ha sido la responsable de esa confianza que me he ganado.*

*Por último quisiera mencionar el único nombre que aparecerá en mis agradecimientos, cumpliendo así con una promesa realizada desde el año pasado. Es fácil imaginar que siempre hay un gran culpable de todo y en este caso es Fidel Castro Ruz, la persona que ostenta ese título, dado que de otra manera, me hubiese graduado de informático “tal vez”, pero en la Universidad de Pinar del Río o quién sabe dios donde, pero si, fue él quien llevó a cabo esta idea que ha dado fruto y dará más.*

*Lázaro César Sánchez Vázquez.*

## RESUMEN

El Centro de Información y Mando de la Unidad Provincial de Patrullas (UPP) de La Habana, ante la necesidad de brindar servicios cada vez más eficientes en beneficio de la población, solicita a la Universidad de las Ciencias Informáticas (UCI) el perfeccionamiento de su sistema informático. Así surge el Sistema de Atención a Emergencias (SAEM) y como parte de su desarrollo se crea este trabajo, el cual tiene como objetivo principal realizar el diseño y la configuración de la base de datos del área de Atención a Llamadas. En este documento se referenciaron conceptos importantes dentro del mundo de las bases de datos, exponiéndose temas relacionados con el Sistema Gestor de Base de Datos (SGBD) empleado. Se incluyó una descripción del submodelo atención a llamadas, así como sus elementos más representativos. Se brindaron las vías para lograr una pronta recuperación del servidor en situaciones de fallas, así como las políticas de salvos y medidas de seguridad definidas para garantizar la seguridad de los datos. También fueron descritos los resultados de las pruebas realizadas a la solución para su validación. Se obtuvo un diseño de base de datos que garantiza la persistencia de la información y posibilita un acceso fácil a la misma, aplicando patrones de diseño que permitieron modelar distintas situaciones empleando buenas prácticas y reduciendo el tiempo de modelado. Finalmente, se recomendaron algunos aspectos a tener en cuenta para próximas versiones de la base de datos.

**Palabras clave:** SGBD, base de datos, SAEM.

## ÍNDICE

INTRODUCCIÓN .....	1
CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.....	5
1.    Introducción .....	5
2.    Centros de Atención a Emergencias.....	5
3.    Sistemas de Atención a Emergencias .....	6
3.1.    Bases de Datos en los Sistemas de Atención a Emergencias .....	6
4.    Base de Datos .....	8
4.1.    Base de Datos Relacional.....	8
5.    Metodología de Diseño de Base de Datos.....	9
5.1.    Diseño Conceptual.....	9
5.2.    Diseño Lógico.....	10
5.3.    Diseño Físico .....	10
6.    Sistemas Gestores de Bases de Datos (SGBD).....	11
6.1.    Sistema Gestor de Bases de Datos Oracle .....	12
6.2.    Novedades de Oracle 11g.....	12
7.    Arquitectura de Bases de Datos .....	14
8.    Lenguajes.....	15
8.1.    Lenguaje Estructurado de Consulta (SQL) .....	16
8.1.1.    Lenguaje de Definición de Datos (DDL) .....	16
8.1.2.    Lenguaje de Control de Datos (DCL).....	16
8.1.3.    Lenguaje de Manipulación de Datos (DML) .....	16
8.1.4.    Lenguaje de Control de Transacciones (TCL).....	17
8.2.    Lenguaje de programación (PL/SQL) .....	17
9.    Herramientas y Tecnologías .....	17
9.1.    Herramientas de Modelado .....	18
9.1.1.    Embarcadero ER/Studio 8.....	18
9.2.    Herramientas de Desarrollo .....	18
9.2.1.    SQL*Plus .....	19



9.2.2.    PL/SQL Developer 9 .....	19
9.3.    Sistema Operativo del Servidor .....	19
9.3.1.    SUSE Linux Enterprise Server 10.....	19
10.    Conclusiones.....	20
CAPÍTULO II: DISEÑO E IMPLEMENTACIÓN.....	21
1.    Introducción .....	21
2.    Diseño .....	21
2.1.    Submodelo Atención a Llamadas .....	21
3.    Patrones de diseño .....	36
3.1.    Llaves subrogadas.....	36
3.2.    Árboles.....	37
3.3.    Generalizaciones.....	40
4.    Pautas y restricciones del diseño de la Base de Datos .....	40
4.1.    Integridad de datos.....	40
4.2.    Nomenclatura y descripción.....	41
4.3.    Normalización.....	42
5.    Optimización.....	42
5.1.    Particionado.....	43
5.2.    Procedimientos almacenados .....	46
6.    Conclusiones.....	47
CAPÍTULO III: CONFIGURACIÓN Y SEGURIDAD DEL SERVIDOR.....	48
1.    Introducción .....	48
2.    Almacenamiento .....	48
2.1.    Espacios de tablas ( <i>Tablespaces</i> ).....	48
2.2.    Espacios de tablas para el SAEM .....	48
3. <i>Flashback</i> y <i>Undo</i> .....	49
4.    Mecanismos de Salva ( <i>Backup</i> ) .....	49
4.1.    Tipos de <i>Backups</i> .....	50
4.2.    Estrategia de <i>Backup</i> de la solución .....	51

5.	Métodos de Recuperación .....	51
5.1.	Recuperación ante pérdida de archivo de control .....	51
5.2.	Recuperación ante pérdida de archivo de reconstrucción.....	52
5.3.	Recuperación ante pérdidas de archivos de datos.....	53
6.	Implementación de la Seguridad.....	54
6.1.	Usuario de la Base de Datos .....	54
6.2.	Métodos de Autenticación .....	55
6.2.1.	Autenticación por la Base de Datos de Oracle .....	55
6.3.	Políticas de seguridad para contraseñas .....	56
6.4.	Perfiles de usuario .....	56
6.5.	Roles y privilegios .....	57
7.	Conclusiones.....	57
CAPÍTULO IV: VALIDACIÓN DE LA PROPUESTA.....		58
1.	Introducción .....	58
2.	Prueba de normalización al modelo de datos .....	58
3.	Pruebas de integridad .....	58
3.1.	Integridad referencial .....	58
3.2.	Integridad de dominio .....	60
3.3.	Integridad de entidad .....	61
4.	Prueba de carga.....	62
5.	Conclusiones.....	64
CONCLUSIONES GENERALES.....		66
RECOMENDACIONES .....		67
REFERENCIAS BIBLIOGRÁFICAS.....		68

## INTRODUCCIÓN

La información es uno de los recursos más importantes de cualquier organización. Con el desarrollo y la integración de la informática, las telecomunicaciones y las técnicas para el tratamiento de datos, se han facilitado los procesos de recopilación, almacenamiento y procesamiento para su posterior utilización.

En la actualidad, la información generalmente se organiza en soportes informáticos que facilitan su localización y manipulación. El volumen de información que se genera cada vez es más grande y los datos son más complejos, por este motivo se requiere de un mecanismo de almacenamiento flexible y que admita operaciones complejas. Una de las formas más utilizadas de almacenar la información es mediante las Bases de Datos, las cuales son gestionadas a través de aplicaciones denominadas Sistemas Gestores de Bases de Datos (SGBD).

En Cuba existe el Centro de Información y Mando de la Unidad Provincial de Patrullas (UPP) de La Habana, en el cual, para proporcionar efectividad y agilidad en la atención a las distintas solicitudes de emergencia de la población, se cuenta con un sistema informático, denominado “Sistema Automatizado de Puesto de Mando de Seguridad Pública”. La UPP ante la necesidad de brindar servicios cada vez más eficientes en beneficio de la población, solicita a la Universidad de las Ciencias Informáticas (UCI) el perfeccionamiento de dicho sistema informático. Así surge el desarrollo del Sistema de Atención a Emergencias (SAEM), cuyo objetivo es perfeccionar y actualizar en materia de software todos los procesos con los que actualmente se labora en el Centro. (1)

Como parte del desarrollo del SAEM surgen las siguientes necesidades:

- ✓ Guardar y manipular un volumen elevado de información que se maneja en tiempo real.
- ✓ Registrar la información histórica generada como resultado de las distintas solicitudes de emergencia realizadas por la población, datos que posteriormente podrán ser modificados o eliminados.
- ✓ Interactuar con la información en el mínimo de tiempo teniendo en cuenta que se trata de un sistema de atención a emergencias.
- ✓ Guardar la configuración de los distintos subsistemas y los datos de los usuarios, de manera que

permita la disponibilidad y confidencialidad de los datos.

Teniendo en cuenta la situación anteriormente expuesta, se define como **problema a resolver**: ¿Cómo garantizar la persistencia y la recuperación de los datos del Centro de Información y Mando de la Unidad Provincial de Patrullas cumpliendo con los requisitos definidos por el cliente?

A partir del problema a resolver se define como **objeto de estudio** para la investigación: Diseño y configuración de bases de datos para Sistemas Informáticos de Gestión de Emergencias, teniendo como **campo de acción**: Diseño y configuración de bases de datos del área de Atención a Llamadas del Centro de Información y Mando de la Unidad Provincial de Patrullas.

Para dar solución al problema planteado, se propone como **objetivo general**: Realizar el diseño y la configuración de la base de datos del área de Atención a Llamadas del Centro de Información y Mando de la Unidad Provincial de Patrullas.

Una vez definido el objetivo general y en aras de darle cumplimiento, se plantean los siguientes **objetivos específicos**:

- ✓ Diseñar la base de datos para lograr la persistencia de la información.
- ✓ Configurar la base de datos para garantizar la seguridad y disponibilidad de los datos.
- ✓ Implementar los procedimientos almacenados subyacentes a los subsistemas del SAEM para optimizar la respuesta ante consultas complejas.

De acuerdo con el objetivo general y los objetivos específicos, se plantea la siguiente **idea a defender**: Con el diseño y configuración de la base de datos del área de Atención a Llamadas, se provee al Centro de Información y Mando de la Unidad Provincial de Patrullas, de un sistema cuya base de datos permite la persistencia y recuperación de la información que se maneja, y garantiza todos los requisitos definidos por el cliente.

Para desarrollar satisfactoriamente la investigación se han trazado las siguientes **tareas investigativas**:

- ✓ Obtención de las clases persistentes relativas a los procesos del negocio que serán automatizados por el Sistema de Atención a Emergencias.

- ✓ Investigación de las estructuras de almacenamiento y privilegios de usuarios en el servidor de base de datos que se utiliza.
- ✓ Investigación de los patrones de diseño de bases de datos.
- ✓ Definición de la distribución de almacenamiento y los usuarios en el servidor de base de datos.
- ✓ Diseño de la base de datos del Sistema de Atención a Emergencias.
- ✓ Instalación del servidor y montaje de la base de datos.
- ✓ Creación de los scripts de inserción para los datos de las tablas nomencladores.
- ✓ Estudio de los métodos de salvallas y recuperación de bases de datos, y selección del más adecuado.
- ✓ Definición de las políticas de salvallas (*backup*) y recuperación (*recovery*) de la base de datos.
- ✓ Implementación de funciones, vistas, disparadores (*triggers*) y secuencias que permitan manipular la información dentro del Sistema Gestor de base de datos.

En la presente investigación se utilizaron dos de los tipos existentes de **Métodos de Investigación Científica**, los Métodos Teóricos y los Empíricos, dentro de ellos se pueden encontrar diferentes especificaciones:

### **Métodos teóricos:**

**Analítico-Sintético:** Se utilizó para analizar y resumir la documentación recopilada, a través de los diferentes medios bibliográficos, para de esta forma desarrollar un mejor diseño y configuración de la base de datos que se propone.

**Modelación:** Se empleó para modelar el problema planteado, creando el modelo de datos que cumple con el objetivo de dar una solución a la situación presente. Se crean abstracciones con el objetivo de explicar la realidad.

### **Método Empírico:**

Observación: Se recurrió a este método para la recolección de la información; permitió, partiendo de la situación presente, llegar a nuevos conocimientos para lograr una solución práctica del problema planteado, llevando a cabo el registro visual del fenómeno en cuestión.

## **El desarrollo del presente trabajo de diploma está estructurado en 4 capítulos:**

- ✓ **CAPÍTULO I: “Fundamentación Teórica”.** En este capítulo se señala la fundamentación teórica, conceptos y definiciones, que apoyan este trabajo de diploma y que resultan de interés en la presente investigación. Se realiza además un estudio de las bases de datos relacionales, las metodologías de diseño de bases de datos, así como las distintas herramientas y lenguajes empleados para el desarrollo de esta solución.
- ✓ **CAPÍTULO II: “Diseño e Implementación”.** Capítulo donde se plantea el modelo de datos propuesto; la normalización y los patrones de diseño aplicados a éste, así como otros aspectos referentes al modelado de la base de datos. También se trata el tema de los procedimientos almacenados implementados en el servidor.
- ✓ **CAPÍTULO III: “Configuración y Seguridad del Servidor”.** Se tratan los aspectos generales de la configuración del servidor, el modelo de almacenamiento de los datos en éste, el control de transacciones, así como, los mecanismos de salva y recuperación. Se expone además, lo referente a la seguridad del servidor: métodos de autenticación, perfiles de usuario, políticas de seguridad para contraseñas, roles y privilegios.
- ✓ **CAPÍTULO IV: “Validación de la propuesta”.** En este capítulo se muestran los resultados de las pruebas de normalización al modelo de datos. Se exponen, además, los resultados obtenidos de las pruebas de integridad y pruebas de carga realizadas a la base de datos.

# Capítulo I. Fundamentación Teórica

---

## CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.

### 1. Introducción

En el capítulo se presenta una serie de conceptos y aspectos teóricos que constituyen un punto de partida para el desarrollo de la presente investigación. Se realiza un estudio de temas importantes como son los modelos de bases de datos existentes, así como las herramientas y tecnologías a utilizar, para seleccionar, respetando los lineamientos tecnológicos del cliente, las que cumplan con los intereses de la actual solución.

### 2. Centros de Atención a Emergencias

Los Centros de Atención a Emergencias son entidades que comprenden un conjunto de órganos coordinados (policiales, bomberos, salud pública) que para la realización de sus tareas se auxilian de un centro integral de video monitoreo, bases de datos y aplicaciones informáticas de inteligencia, e integra información para la toma de decisiones de viabilidad, seguridad pública, medio ambiente, protección civil, servicios a la comunidad y urgencias médicas (2).

Los Centros de Atención a Emergencia deben:

- Poseer un número telefónico para ser contactado.
- Contestar rápido las llamadas.
- Atender con calidad el servicio.
- Enviar apoyo de inmediato.
- Brindar asistencia o asesoría al momento.
- Lograr que el apoyo llegue rápidamente al lugar del suceso.
- Brindar auxilio de manera eficiente.
- Verificar la calidad del servicio proporcionado. (3)

Con el fin de lograr una mayor efectividad en las respuestas ante las situaciones de urgencia, estos centros se caracterizan por contar con un número único de tres dígitos, gratuito y disponible todos los días

# Capítulo I. Fundamentación Teórica

---

del año, ejemplo de ello son el 911 en Estados Unidos, 112 en Europa, 123 de Colombia, 171 en Venezuela (4) y el 105 y 106 de Cuba que representan el Centro de Bomberos y a la Policía Nacional Revolucionaria, respectivamente.

### 3. Sistemas de Atención a Emergencias

Los sistemas de Atención a Emergencia constituyen la estructura tecnológica que permite realizar coordinaciones entre los diferentes órganos destinados a la atención de las emergencias. En muchos casos posibilitan ubicar la dirección de las emergencias ocurridas en un mapa, así como los recursos asociados a las mismas (5).

Se utilizan para automatizar procesos y dar apoyo a otras acciones, con el objetivo de mejorar el tiempo de respuesta a las necesidades de los clientes, para acceder concurrentemente a documentos e información, minimizar el tiempo de acceso a información crucial, así como al análisis de estadísticas y resultados de investigaciones para la toma de decisiones (6). Muchos de estos sistemas poseen hoy en día tecnologías avanzadas (telefonía, radio, sistemas de cómputo, GPS, mapificación en tiempo real y grabación de llamadas) para aumentar la rapidez y confiabilidad de las respuestas ante cualquier emergencia.

#### 3.1. Bases de Datos en los Sistemas de Atención a Emergencias

En los sistemas de gestión de emergencias, las bases de datos juegan un papel fundamental. Éstas constituyen el centro de almacenamiento de la información que se maneja, la cual está asociada a denuncias y solicitudes de emergencia de la población, registrando datos de las llamadas de emergencia, trabajadores involucrados en su atención y el desempeño de los mismos, así como recursos humanos y materiales con los que se cuenta.

La información histórica registrada en estas bases de datos constituye un punto de partida para la prevención sucesos y hechos delictivos, así como para la elaboración de planes de respuesta ante acciones combatidas con anterioridad. También, con el empleo de bases de datos, se contribuye a elevar los niveles de confidencialidad y disponibilidad de la información.

Constituyen dos ejemplos de estas bases de datos, la del Sistema de Gestión de Emergencias de Seguridad Ciudadana (desarrollado en la UCI para el Ministerio del Poder Popular para Relaciones



# Capítulo I. Fundamentación Teórica

---

Interiores y Justicia de Venezuela) y la del Sistema Automatizado de Puesto de Mando de Seguridad Pública:

## ➤ **Base de datos del Sistema de Gestión de Emergencias de Seguridad Ciudadana (171)**

La base de datos del Sistema de Gestión de Emergencias de Seguridad Ciudadana (SIGESC), cumple las siguientes misiones:

- ✓ 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.
- ✓ Garantizar la confidencialidad y seguridad de la información manipulada. (7)

A partir del estudio que se realizó a la bibliografía encontrada sobre esta base de datos, se logró identificar elementos de configuración y seguridad fundamentales para una solución como la que se requiere, entre éstos se encuentran: estrategia de salva, establecer seguridad a nivel de objetos, los privilegios del sistema a asignar a los usuarios y la creación de perfiles de usuario con el fin de limitar los recursos que son asignados a los mismos.

## ➤ **Base de datos del Sistema Automatizado de Puesto de Mando de Seguridad Pública (106)**

El Sistema Automatizado de Puesto de Mando de Seguridad Pública interactúa con una base de datos. En ésta se almacena la información que se maneja a través del sistema: datos de las llamadas que se reciben, circulaciones, hechos, medios con que se cuenta, incidencias de desempeño, ordenes de radio y otros conceptos propios del negocio.

Constituye además, una fuente de información para la generación de los reportes que se realizan con el sistema y que tienen como objetivo contribuir a la toma de decisiones. Basándose en las estadísticas registradas de suceso y delitos anteriores, es posible planificar la distribución de los recursos de la Unidad (hombres, carros de patrulla, móviles de auxilio) en días y horarios específicos.

# Capítulo I. Fundamentación Teórica

---

El estudio realizado a esta base de datos permitió identificar la información que se persiste en la misma como resultado del trabajo que se lleva a cabo en la UPP. Esto constituye un punto de partida para el diseño del modelo de datos de la solución, pues permite conocer las entidades y relaciones existentes, la cuales pudieran mantenerse en el modelo de datos resultante del presente trabajo, reduciendo el tiempo de modelado y tributando a dar una solución lo más robusta posible.

## 4. Base de Datos

Ante la necesidad de gestionar (recopilar, almacenar y procesar) los crecientes volúmenes de información y la búsqueda de una manera de resolver esta problemática, surgieron las bases de datos, que se definen como un conjunto exhaustivo de datos estructurados, fiables y homogéneos, organizados independientemente de su utilización e implementación en una computadora, accesibles en tiempo real, que pueden compartir varios usuarios con necesidades de información diferentes y no predecibles en el tiempo (8).

Las bases de datos responden a diferentes modelos como son: modelo jerárquico, de red, relacional, relacional/objeto, y de objeto. De éstos, el Modelo Relacional es uno de los más utilizados, cuenta con una fundamentación teórica que lo respalda, por ejemplo las dos versiones del documento El Modelo Relacional, escritos por el creador del modelo Edgar Frank Codd<sup>1</sup> (9).

### 4.1. Base de Datos Relacional

Las bases de datos se usan desde los años 60, pero no fue hasta finales de 1960 y principios de los 70, cuando Edgar Frank Codd, inventó el Modelo Relacional. Este modelo ha sido precedido por los modelos de red y jerárquico, que a diferencia de ellos, su fundamentación teórica fue creada primero por Codd y luego nacieron los productos que se adaptaron a la misma, en otras palabras, los modelos anteriores (jerárquico y de red) representan un atraso tecnológico con respecto al modelo relacional, porque primero se usaron y luego se pensó en que esa era su arquitectura, además de estar muy relacionados con la forma en que se almacena la información, es decir, su fundamentación surge a partir de un análisis de cómo se estaba guardando la información en el sistema (10).

---

<sup>1</sup> También definió las 12 reglas con las que deben cumplir los Sistemas Gestores de Bases de Datos para definirse como Relacionales

# Capítulo I. Fundamentación Teórica

---

El Modelo Relacional difiere además, de los modelos que le anteceden, en que la información es almacenada y trabajada sobre representaciones de tablas (a grandes rasgos se puede ver así, pero lo cierto es que se trabaja con relaciones, de ahí el nombre de Modelo Relacional), ya sean las entidades que son los objetos de los que se guardará información, como los vínculos entre ellas y los resultados de las operaciones que se realizan sobre los datos, también dan como resultado otras tablas (10). De esta manera, basándose en dicho paradigma, el usuario sólo interactúa con los datos mediante la representación en forma de tablas bidimensionales.

## 5. Metodología de Diseño de Base de Datos

Una metodología es un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de un producto software (11). La metodología de diseño de bases de datos surge ante la necesidad de una guía que ayude al diseñador a crear estructuras correctas y fiables, minimizando los tiempos de diseño y explotando todos los 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 independientemente, utilizando técnicas específicas. Así el diseño de una base de datos se descompone en diseño conceptual, diseño lógico y diseño físico (7).

### 5.1. Diseño Conceptual

El proceso de diseño de la base de datos parte del Diseño Conceptual, que es la transformación de las especificaciones de requisitos de usuario en el Esquema Conceptual. Este esquema es la descripción de alto nivel de la estructura de la base de datos, independiente del Sistema Gestor de Base de Datos que sea utilizado para la manipulación de la misma. (7)

En este diseño se realizan las tareas de:

- Identificación de las entidades y las relaciones entre las mismas.
- Determinación de los dominios de los atributos de las entidades.
- Realización del Diagrama Entidad-Relación entre otras actividades.

# Capítulo I. Fundamentación Teórica

---

Las entidades son los objetos de los que el usuario desea guardar información y pueden estar relacionadas entre sí. Además poseen atributos, que son las características o cualidades que describen a estos objetos.

En el Modelo Relacional, antes mencionado, las entidades son vistas como tablas, luego los atributos constituyen los campos de estas tablas. Cada atributo debe pertenecer a uno de los siguientes dominios que se consideran los básicos: cadena de texto, fecha, número o valores lógicos (10).

El proceso de Diseño Conceptual culmina con la revisión del Esquema Conceptual<sup>2</sup>, con el usuario.

## 5.2. Diseño Lógico

El Diseño Lógico es un paso importante para la obtención de una base de datos óptima o al menos poco redundante, debido a que en este diseño, que parte del Esquema Conceptual y en el cual se obtiene un Esquema Lógico, es seleccionado el modelo de datos sobre el cual se trabajará.

El Diseño Lógico es el proceso de construir un nuevo modelo de datos que utiliza la empresa, basándose en un modelo de bases de datos específico (modelo jerárquico, de red, relacional, orientado a objetos) y en el esquema de información creado en el diseño conceptual, independiente del Sistema Gestor de Base de Datos concreto que se vaya a utilizar y de cualquier otra consideración física (7).

En este diseño se realiza la normalización de la base de datos. El proceso de normalización, propuesto por el propio creador del Modelo Relacional, E.F. Codd, tiene como objetivo eliminar la redundancia hasta el punto de hacerla mínima, resolver los problemas de actualización de las bases de datos (inserción, eliminación y modificación), así como mejorar y facilitar las consultas. Codd desarrolló cuatro niveles de normalización, conocidas como Formas Normales (Primera, Segunda y Tercera Forma Normal, y Forma Normal de Boyce/Codd, conocida así por el nombre de los autores). (10)

## 5.3. 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. (12)

---

<sup>2</sup> Todos los diseños culminan con un esquema que es utilizado por el siguiente nivel

# Capítulo I. Fundamentación Teórica

---

Este diseño es dependiente del Sistema Gestor de Bases de Datos, porque cada esquema a este nivel debe adaptarse al gestor a utilizar.

En el diseño físico se debe:

- Obtener un conjunto de relaciones (tablas) y las restricciones que se debe 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. (7)

Este diseño, como los anteriores, es importante porque permite especificar una forma eficiente para guardar los datos.

## 6. Sistemas Gestores de Bases de Datos (SGBD)

Para lograr la independencia, integridad, seguridad y manipulación de los datos, así como una mayor rapidez y efectividad en su gestión, se necesita de Sistemas Gestores de Bases de Datos (SGBD), que son aplicaciones que interactúan con los usuarios de los programas de aplicación y la base de datos, permitiendo a los usuarios definir, crear y mantener las bases de datos, así como proporcionar un acceso controlado a éstas (13). Con su empleo se logra:

- Garantizar la independencia entre los datos y los programas que los utilizan, siendo independientes la representación física y lógica de los datos.
- Minimizar la redundancia de la información para disminuir el tiempo de acceso a los datos.
- Garantizar la seguridad e integridad, permitiendo que los usuarios accedan sólo a los datos a los que están autorizados e impidiendo que se almacene o exista en la base de datos información con errores.
- Lograr la integración y sincronización de las bases de datos. Aunque el SGBD almacene la información con cierta estructura y cierto tipo de representación, debe garantizar entregar al programa de aplicación datos que solicita y en la forma en que lo solicita, permitiendo el acceso múltiple y simultáneo a la BD, de modo que los datos puedan ser compartidos por diferentes

# Capítulo I. Fundamentación Teórica

---

usuarios a la vez.

- Permitir la recuperación de la información mediante métodos que garanticen la restauración de la base de datos al producirse una falla técnica, como por ejemplo, una interrupción de la energía eléctrica. (14)

Dentro de los SGBD más conocidos se encuentran DB2 de IBM, Postgres de Postgres Inc., MySQL, comprado recientemente por Oracle Inc., y Oracle de Oracle Inc.

## 6.1. Sistema Gestor de Bases de Datos Oracle

Oracle forma parte de los gestores relacionales dominantes del mercado y además es multiplataforma. La versión 2 de Oracle fue el primer SGBD relacional que soportase SQL<sup>3</sup> (15).

Hasta la versión 11g, se cuenta un elevado número de prestaciones:

- Reúne las características de *ACID* (*atomicidad, consistencia, aislamiento y durabilidad*).
- Controla la integridad referencial.
- Se pueden almacenar indistintamente números, textos, formatos de fecha, imágenes, sonidos, videos y datos espaciales, así como la definición de dominios.
- Permite la administración de usuarios y roles.
- Soporta funciones disparadoras, cursores, procedimientos almacenados y funciones.
- Permite el particionado de tablas con varias extensiones y fuertes capacidades para su administración.

En el 2007 es que se desarrolla esta versión, la cual incluye novedades y amplía las capacidades existentes.

## 6.2. Novedades de Oracle 11g

### Oracle Real Application Testing

---

<sup>3</sup> Lenguaje Estructurado de Consulta, SQL por sus siglas en inglés.

# Capítulo I. Fundamentación Teórica

---

Oracle 11g es la primera base de datos del mundo en incluir funcionalidades que permiten hacer pruebas de cambios en aplicaciones simulando las cargas reales generadas por los usuarios en los entornos de producción. Real Application Testing permite reducir de manera drástica los tiempos, riesgos y costes derivados de la implantación de cambios, asegurando que las aplicaciones se comportarán de manera adecuada y predecible tras las modificaciones. Con Real Application Testing los clientes ganan en flexibilidad puesto que pueden responder de manera más efectiva a los requerimientos cambiantes del negocio y hacer una gestión del cambio más efectiva (16). Esta funcionalidad se agrega en la edición empresarial y es compatible con versiones anteriores de Oracle, para que los clientes que utilizan Oracle 9i y Oracle 10g puedan emplearla para acelerar la actualización de sus bases de datos. (17).

## **Oracle Data Guard**

Oracle Data Guard permite que los clientes utilicen su base de datos *standby* para mejorar el desempeño de su entorno de producción y también para brindar protección contra las fallas del sistema y desastres en el sitio (18).

## **Particionado**

El particionado de tablas consiste en dividir horizontalmente las tuplas entre unidades que pueden repartirse por más de un grupo de archivos de una base de datos, permitiendo que todas las particiones de una misma tabla se traten como una sola entidad lógica cuando se realicen consultas o actualizaciones en los datos (19). Es una técnica de optimización utilizada cuando las tablas son muy grandes. Esta práctica mejora la capacidad de administración, desempeño y disponibilidad (20). Oracle brinda varias estrategias de particionado: por rango, *hash*, lista y por composición (combina los tipos anteriores, utilizando un primer particionado de un tipo determinado, y luego, para cada partición, realizar un segundo nivel de particionado utilizando otro método). Se introduce en versión 11g la combinación *hash-hash* en una extensión del particionado por composición, así como el particionado basado en columnas virtuales (que no existen físicamente en la tabla), el de Sistema (donde se puede gestionar directamente en que partición de la tabla se insertan los registros) y el particionado por Intervalos (21).

## **Oracle Total Recall**

# Capítulo I. Fundamentación Teórica

---

Esta funcionalidad permite a los administradores consultar datos en tablas designadas en fechas anteriores, y es una manera fácil y práctica de agregar una dimensión de tiempo al rastreo de cambios, auditoría y cumplimiento de datos (18).

## Disponibilidad

Como nueva característica de disponibilidad se incluye Oracle Flashback Transaction, para revertir transacciones realizadas con errores, así como cualquier transacción relacionada; Parallel Backup and Restore para mejorar el desempeño de salva y restauración de bases de datos muy grandes; y *hotpatching* para aplicar parches a las bases de datos sin la necesidad de cerrarlas (18).

## 7. Arquitectura de Bases de Datos

Los SGBD pueden desarrollarse sobre cualquier arquitectura que se proponga en la empresa, pero uno de los estándares existentes de arquitectura de base de datos propuesto por la ANSI/SPARC<sup>4</sup>, es el modelo arquitectónico en tres capas o niveles (10). Este modelo, busca separar en capas las transformaciones a las que debe estar sujeta la información (lo que se conoce como mapeo) para hacer llegar de forma cómoda lo que necesita ver el usuario, aislando a éste de la manera en que está almacenada la información.

Las capas o niveles que propone el estándar ANSI/SPARC son: nivel externo, nivel interno y nivel intermedio entre las dos anteriores, conocido como lógico o conceptual.

### Nivel externo

Es el nivel más alto, representado por el esquema externo, visto por el programador de aplicaciones o el usuario, para quienes, en esta vista, sólo son de interés algunas porciones de la base de datos. Para cada usuario, su vista externa o sus vistas externas, debido a que puede acceder a varias, representarán toda la base de datos. (10)

---

<sup>4</sup> Su nombre completo es ANSI/X3/SPARC, establecido en 1972 por el SPARC (Comité de Planeación de Estándares y Requerimientos) del Comité sobre Computadores y Procesamiento de Información (X3) del ANSI (Instituto Nacional Americano de Estándares)



# Capítulo I. Fundamentación Teórica

---

En este nivel, el usuario trabaja sobre los sub-lenguajes de definición y manipulación de datos, contenidos los mismos dentro de algún lenguaje de Cuarta Generación como lo es Java o PHP, por citar sólo dos de los más utilizados.

## **Nivel Lógico o Conceptual**

En este nivel es donde se describe toda la información de la base de datos, pero de una forma más abstracta comparada con la manera en que por lo general se almacenan los datos físicamente. Cuenta también con un lenguaje de definición de datos específico del nivel con el que se representa la base de datos como relaciones sencillas (teniendo en cuenta que se ha elegido el modelo relacional). En estas definiciones pueden estar presentes restricciones de seguridad y de integridad. Es decir, que en este nivel no sólo se describen los datos como tal, sino también la forma en que son utilizados. (10)

## **Nivel Interno**

Es el nivel donde el lenguaje de definición de datos describe cómo están almacenados los datos físicamente, pero no forma parte de los datos en sí. Es la representación de muchos registros internos<sup>5</sup> aunque está distante del nivel físico porque no tiene que ver con los registros físicos (bloques o páginas<sup>6</sup>) ni con ninguna consideración específica de los dispositivos. En el esquema interno se describen los diversos tipos de registros almacenados, los índices existentes, cómo están representados los campos almacenados y en qué secuencia están dichos campos (10).

## **Transformaciones**

Existen dos tipos de transformaciones necesarias para lograr pasar la información de un nivel a otro, la transformación externo/conceptual y la transformación conceptual/interno. La primera define la correspondencia de los datos en los niveles externo y conceptual y garantiza la independencia lógica, mientras que la segunda define la correspondencia entre la vista conceptual y la base de datos almacenada, y tiene el fin de preservar la independencia física de los datos (10).

## **8. Lenguajes**

---

<sup>5</sup> Término utilizado por ANSI/SPARC para los registros almacenados. (10)

<sup>6</sup> Unidad menor de información que representa la cantidad de datos transferidos entre el almacenamiento secundario y la memoria principal.

# Capítulo I. Fundamentación Teórica

---

Para interactuar con las bases de datos se hace necesario el empleo de lenguajes que pueden estar compuestos por sub-lenguajes.

## **8.1. Lenguaje Estructurado de Consulta (SQL)**

El SQL es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre éstas. Reúne las características del álgebra y el cálculo relacional permitiendo lanzar consultas con el fin de recuperar información de una base de datos de una forma sencilla. (22)

Es además el lenguaje estándar para trabajar con bases de datos relacionales, pues su filosofía se basa en el trabajo sobre tablas (entidades) y sus relaciones, de forma similar a como funciona el Modelo Relacional.

Es válido destacar que su función no sólo se limita a realizar consultas a la base de datos, sino que incluye otras funciones. Atendiendo a esta característica, estas funciones se clasifican según su finalidad para dar origen a cuatro sub-lenguajes: control de transacciones, definición, manipulación y control de datos.

### **8.1.1. Lenguaje de Definición de Datos (DDL)**

El DDL es un sub-lenguaje que se encuentra dentro del SQL. Es utilizado para definir y administrar objetos de la base de datos, y como se conoció en el modelo de arquitectura en tres capas, para que los SGBD logren mantener la independencia física de los datos y la manera en que son visualizados por el usuario, por cada uno de los niveles de la arquitectura ANSI/SPARC, existe un lenguaje de definición de datos que se adapta a las características propias del nivel. Se hace uso de este lenguaje al crear las tablas, vistas y procedimientos almacenados en el sistema, así como cuando se definen los tipos de datos de los campos de las tablas creadas. (23)

### **8.1.2. Lenguaje de Control de Datos (DCL)**

Los comandos GRANT y REVOKE son funciones utilizadas para el control de acceso a los objetos por los diferentes usuarios de la base de datos, es decir, asignan y eliminan privilegios sobre éstos y forman parte del Lenguaje de Control de Datos. (10)

### **8.1.3. Lenguaje de Manipulación de Datos (DML)**

# Capítulo I. Fundamentación Teórica

---

El Lenguaje de Manipulación de Datos, a diferencia del DDL, solo trabaja en los niveles superiores de la arquitectura mencionada, es decir, en el Nivel Externo y en el Nivel Conceptual y como su nombre lo indica es para manipular los datos que se encuentran en los diferentes objetos de la Base de Datos. Se hace uso del mismo, cuando se trabaja con instrucciones de consulta (SELECT), de inserción (INSERT), de modificación (UPDATE) o de eliminación (DELETE). (23)

## 8.1.4. Lenguaje de Control de Transacciones (TCL)

Dado que las consultas a la base de datos, son atómicas, se necesitan comandos para ordenar que se realice la consulta y son los comandos COMMIT, ROLLBACK, SAVEPOINT, los que permiten ejecutar la consulta, regresar al punto anterior en caso de fallo de la misma o que los resultados devueltos no sean los esperados como es el caso del último comando presentado.

## 8.2. Lenguaje de programación (PL/SQL)

PL/SQL es considerado un lenguaje de tercera generación, pues incluye el procedimiento (el programador indica lo que se debe hacer y cómo se debe hacer) (24). Se utiliza para complementar las consultas a las bases de datos en Oracle, ya que resuelve muchas de las limitaciones de SQL. (25)

Como lenguaje de tercera generación, permite el manejo de variables, el uso de estructuras de control (bifurcaciones y bucles) y controlar las excepciones. Además incorpora un completo soporte para la programación orientada a objetos (POO) (25).

Los programas creados con PL/SQL se pueden almacenar en la base de datos como un objeto, quedando disponibles para los usuarios. Además de tenerse como imprescindible para la creación de disparadores, otra característica de PL/SQL es que está incluido en el servidor y algunas herramientas del cliente, además soporta todos los comandos de consulta y manipulación de datos, aportando al lenguaje SQL las estructuras de control y otros elementos propios de los lenguajes de programación de tercera generación.

## 9. Herramientas y Tecnologías

Para el diseño y posterior trabajo con bases de datos existen herramientas que auxilian a los usuarios en estas tareas. Existen algunas que permiten visualizar, mediante diagramas, el producto que se desea crear. También hay otras que facilitan la ejecución de tareas como la generación de consultas y trabajo con los datos de manera cómoda e intuitiva.

# Capítulo I. Fundamentación Teórica

---

## 9.1. Herramientas de Modelado

El desarrollo de la Ingeniería de Software como disciplina para lograr terminar los productos en tiempo y con la calidad requerida, ha traído consigo la creación de herramientas que automaticen el proceso de conversión de los diagramas utilizados en dicha disciplina, facilitando el entendimiento del negocio por las partes implicadas en la implementación de la aplicación. Estas herramientas se conocen como herramientas CASE (Computer Aided Software Engineering) o de modelado como también son conocidas. Entre las herramientas de modelado de bases de datos se tienen: EasyCase, DBDesigner, IBM Rational System Architect, Power Designer, Oracle Data Modeler y Embarcadero ER/Studio.

### 9.1.1. Embarcadero ER/Studio 8

ER/Studio es una herramienta para el modelado exclusivo de diagramas Entidad-Relación. Permite el modelado lógico y físico de los datos, así como modelar a partir de un script SQL. La notación puede verse según las notaciones IDEF 1X o alguna de las tres notaciones de Information Engineering (IE) y una vez realizado el diseño lógico, el ER/Studio permite la transformación a modelo físico librando al usuario de la realización manual de los cambios de sintaxis que dicha transformación conlleva y a su vez genera las entidades necesarias de las relaciones mucho a mucho modeladas por el usuario (26).

Como características a destacar del ER/Studio están:

- Presenta fuertes capacidades para el trabajo con el modelo lógico.
- Presenta sincronización bidireccional entre los modelos lógico y físico.
- Construye automáticamente la base de datos.
- Permite la ingeniería inversa exacta de la base de datos.
- Permite crear un repositorio de modelado de datos para el trabajo en equipo. (26)

## 9.2. Herramientas de Desarrollo

Para la creación de los procedimientos almacenados, la confección de consultas, definición de las restricciones de las tablas o la base de datos, y la configuración del servidor, es necesario interactuar con herramientas como SQL\*Plus y PL/SQL Developer, de las que a continuación se dan características.

# Capítulo I. Fundamentación Teórica

---

## 9.2.1. SQL\*Plus

El SQL\*Plus es el editor de consultas que se instala con el producto Oracle con el que se esté trabajando, pues viene con la distribución tanto del cliente como del servidor. Otra de sus características es que, todo el trabajo es a través de una consola, por lo que puede ser engorroso para usuarios que se estén familiarizando con el gestor Oracle.

## 9.2.2. PL/SQL Developer 9

PL/SQL Developer es un Entorno Integrado de Desarrollo, o IDE (Integrated Developer Environment) utilizado para desarrollar unidades de programas almacenados en bases de datos Oracle. Este IDE permite realizar tareas como editar, compilar, corregir, probar, Debugging (buscar errores sintácticos) optimizar y consultar, además de permitir visualizar diagramas, gráficas sobre las consultas realizadas y reportes en formato HTML. (27)

## 9.3. Sistema Operativo del Servidor

### 9.3.1. SUSE Linux Enterprise Server 10

A pesar de que el gestor de bases de datos Oracle es multiplataforma, se ha optado por el sistema operativo SUSE Linux Enterprise Server 10, que como la mayoría de las distribuciones Linux, brinda alta seguridad, además de poseer varios asistentes gráficos para la realización de diversas tareas. También una de las ventajas de este Sistema Operativo sobre otros sistemas Linux, además de poseer una vasta documentación, SUSE Linux Enterprise Server proporciona una base abierta para una gran variedad de cargas de trabajo de los servidores, como los procesos de cómputos periféricos y de infraestructura, implantación de bases de datos empresariales, aplicaciones de línea comercial, entre otras características (28).

Otras características de SUSE Linux Enterprise Server 10 son:

- Ofrece avanzadas características de gestión de memoria tales como scheduler<sup>7</sup> de I/O que posibilitan una rápida optimización de las aplicaciones.
- Incorpora soporte de clustering<sup>8</sup> para la recuperación automática de caídas del sistema.

---

<sup>7</sup> Planificador de acceso a la memoria por los procesos.

<sup>8</sup> Técnicas para aprovechamiento de las capacidades de procesamiento de dispositivos conectados.

# Capítulo I. Fundamentación Teórica

---

- Proporciona servicio de hotplug<sup>9</sup> para la sustitución de hardware, evitando que los sistemas sean detenidos. (29)

Dentro de los servicios con los que cuenta SUSE Linux están DNS<sup>10</sup>, DHCP<sup>11</sup>, Proxy, Samba y SMTP<sup>12</sup>.

## 10. Conclusiones

En el capítulo se presentaron características de los Centros de Atención a Emergencia y los Sistemas de Atención a Emergencias, así como aspectos asociados a las bases de datos y a los SGBD, fundamentales para la comprensión de la solución a la problemática planteada. Se realizó un estudio de los modelos de datos existentes y se eligió para el desarrollo del trabajo el Modelo Relacional por las facilidades de uso que presenta para el desarrollo rápido de aplicaciones; se explicó además, la arquitectura de bases de datos en tres niveles. También se profundizó en las características de Oracle Database 11g, gestor de base de datos, elegido por estar contemplado en los lineamientos de las herramientas a utilizar de la Dirección de Tecnologías y Sistemas (DTS) del Ministerio del Interior (MININT), además de poseer características que cumplen con las expectativas del proyecto. Como herramienta de modelado, se eligió el Embarcadero ER/Studio, por presentar las características necesarias para el diseño del modelo de datos y presentar el equipo de trabajo experiencia en el trabajo con esta herramienta. El servidor elegido es SUSE Linux Enterprise Server 10, por estar contemplado en los lineamientos de la DTS, así como por poseer características que lo habilitan para la solución que se requiere.

---

<sup>9</sup> Quitar en caliente, con el equipo conectado a la electricidad, funcionando.

<sup>10</sup> Domain Name Server: Servicio de resolución de nombres (43).

<sup>11</sup> Dynamic Host Protocol Configuration: protocolo de configuración dinámica del número de red de las máquinas (número IP) (43).

<sup>12</sup> Simple Message Transfer Protocol: protocolo de transferencia de mensaje (43).

# Capítulo II. Diseño e Implementación

---

## CAPÍTULO II: DISEÑO E IMPLEMENTACIÓN.

### 1. Introducción

En el capítulo se definen conceptos referentes al modelo de datos. Son abordadas las características generales del diseño de la base de datos que se propone, así como las características propias del área de Atención a Llamadas y las entidades correspondientes al modelo de datos asociado a la misma. Para facilitar la escalabilidad de la base de datos se adoptan pautas para el diseño de ésta en cuanto a temas como la integridad de los datos y la nomenclatura. Además, en aras de satisfacer las expectativas del cliente, garantizando una mayor eficiencia en el tratamiento de los datos, se exponen técnicas usadas para la optimización de la base de datos.

### 2. Diseño

Con el SAEM se informatizan procesos de negocio que se agrupan en cuatro áreas de trabajo en la UPP, los cuales, a su vez, se componen por varios módulos. El diseño de BD que se propone cumple con los requisitos definidos por el cliente, posibilitando la persistencia y recuperación de la información que se maneja. Consiste en una sola base de datos para la aplicación, con la cual se establecerá la conexión mediante un único usuario; esta última disposición es un requisito del cliente. Ésta contiene una numerosa colección de entidades (tablas), las cuales se encuentran relacionadas entre sí. Varias de las entidades son nomencladores que son definidos a nivel ministerial o son propios del negocio y son definidos por el cliente.

En una agrupación lógica de las tablas, y en correspondencia con los subsistemas de la aplicación, se definen cuatro submodelos, cuyas entidades son utilizadas por las funcionalidades implementadas en los distintos módulos. Muchas de las entidades, además, son utilizadas por más de un módulo.

#### 2.1. Submodelo Atención a Llamadas

En el área de atención a llamadas las operadoras reciben todas las llamadas que realizan la población u otro sector, enviándolas al oficial de mando correspondiente para darle atención y seguimiento inmediato. También se supervisa el trabajo que realizan las operadoras para la atención de las emergencias y se controla a partir de estadísticas e indicadores el funcionamiento del turno.

## Capítulo II. Diseño e Implementación

El submodelo de datos asociado a esta área agrupa las entidades con las que interactúan tres de los subsistemas del SAEM: Operadora, Supervisor de Operadora y Jefe de Turno de Operadora. Se compone de 66 entidades, de las cuales, 34 son nomencladores. En la tabla 1 se muestran las entidades que conforman el submodelo, especificando las entidades nomencladores y las de negocio o datos.

Nombre del Submodelo	Atención a Llamadas	
Objetivo	Describir los procesos de atención a las llamadas, el trabajo y la supervisión de los trabajadores del área, así como los indicadores del turno de trabajo	
Entidades que lo componen	Negocio o datos	Nomencladores
	dAreaAtencion	nAreaTurnoTrabajo
	dAreaPatrullaje	nCategoriaSuceso
	dCirculacion	nClaseTipo
	dCirculacionMatricula	nClasificacionIncidencia
	dCirculacionObjeto	nClasificacionObjeto
	dCirculacionPersona	nColor
	dCirculacionVehiculo	nColorOjos
	dDatoInteres	nColorPiel
	dDireccion	nEmpresa
	dEmergencia	nEstadoPersonaCirculada
	dIncidenciaDesempeno	nEstadoSuceso
	dIncidenciaSistema	nEstadoTecnico
	dIncidenciaTurno	nEstadoUsuario
	dInformativa	nHorarioTurno
	dLlamada	nLugar
	dManual	nLugarEspecifico
	dPersona	nLugarGeneral
	dRol	nMarcaModelo
	dServicioWeb	nMotivoCirculacion
	dSubsistema	nMunicipio
	dTrabajador	nOcupacion
	dTurnoTrabajo	nPais
dUsuario	nParticipante	
dUsuarioTurnoPuesto	nQuienInforma	
rCategoriaSucesoManual	nRangoEstatura	



## Capítulo II. Diseño e Implementación

	rEmergenciaParticipante	nSector
	rRolSubsistema	nSexo
	rTipoSucesoManual	nTipoIncidencia
	rTurnoTrabajoDatoInteres	nTipoLlamada
	rUsuarioTurnoEstadoUsuario	nTipoLlamadaInformativa
	rUsuarioTurnoReconocimiento	nTipoReconocimiento
	rUsuarioTurnoTrabajo	nTipoSuceso
		nTipoTelefono
		nUnidadMedida

Tabla 1. Entidades que forman el submodelo del Área de Atención a Llamadas.

A continuación se describen las entidades más importantes del submodelo con su definición, atributos y descripciones. El resto de las entidades se pueden ver detalladamente en el documento *Diccionario de Datos\_v1.0.doc* del expediente del proyecto *Perfeccionamiento de los Sistemas Informáticos del Centro de Información y Mando de la Unidad Provincial de Patrullas*.

### ➤ Llamada

<b>Nombre de la entidad</b>		Llamada.					
<b>Descripción de la entidad</b>		Entidad que contiene todos los datos correspondientes de una llamada registrada en la Unidad Provincial de Patrulla.					
Nombre del atributo	Descripción	Tipo	Puede ser nulo	Restricciones		Criterio de Selección	
				Clases válidas	Clases no válidas	Múltiple	Única
idLlamada	Identificador de la llamada.	Varchar.	No.	["A" – "Z"   "a"-"z"   "0"-"9"].	[caracteres especiales].	No.	Sí.
Número de teléfono.	Número del teléfono que realiza la llamada.	Varchar.	Sí.	["0"-"9"] solo son válidos hasta 11	[caracteres especiales   "A"	No.	Sí.

## *Capítulo II. Diseño e Implementación*

				dígitos. –” Z”   “a”-“z”].			
Dirección del teléfono.	Dirección donde está ubicado el teléfono.	Varchar.	Sí.	[caracteres especiales   “A” –” Z”   “a”-“z”].	No aplica.	No.	Sí.
idTipo de teléfono.	Identificador que relaciona la entidad Tipo de Teléfono (que realiza la llamada) y Llamada.	Identificador. (Nomenclado).	Sí.	No aplica.	No aplica.	No.	Sí.
Descripción.	Descripción de la llamada.	Varchar.	No.	[caracteres especiales   “A” –” Z”   “a”-“z”].	No aplica.	No.	Sí.
Hora inicio de llamada.	Hora en que se recibe la llamada.	Datetime.	No.	[caracteres especiales] y [“0”-“9”] con formato de fecha.	[“A” –” Z”   “a”-“z”].	No.	Sí.
Hora de fin de llamada.	Hora en que se termina la	Datetime.	No.	[caracteres	[“A” –” Z”	No.	Sí.

## *Capítulo II. Diseño e Implementación*

---

	llamada.			especial s] y ["0"- "9"] con formato de fecha.	"a"- "z"].		
Hora de inicio de ficha.	Hora en que se comienza a registrar la ficha.	Datetime.	No.	[caracteres especiales] y ["0"- "9"] con formato de fecha.	["A" -"Z"   "a"- "z"].	No.	Sí.
Hora de fin de ficha.	Hora en que se cierra la ficha.	Datetime.	No.	[caracteres especiales] y ["0"- "9"] con formato de fecha.	["A" -"Z"   "a"- "z"].	No.	Sí.
idLlamada asociada.	Identificador que relaciona la entidad Llamada (llamada anterior a la que se asocia) y Llamada (Llamada asociada).	Identificador.	Sí.	No aplica.	No aplica.	No.	Sí.

## Capítulo II. Diseño e Implementación

idTipo de llamada.	Identificador que relaciona la entidad Tipo de llamada (motivo de la llamada) y Llamada.	Identificador (Nomenclado).	No.	No aplica.	No aplica.	No.	Sí.
idUsuario conectado.	Identificador que relaciona la entidad Usuario conectado y Llamada.	Varchar.	No.	["A" – "Z"   "a" – "z"   caracteres especiales ].	["0" – "9"].	No.	Sí.
Estado.	Representa el estado de la ficha que se generó con la llamada.	Varchar.	No.	["A" – "Z"   "a" – "z"   caracteres especiales ].	["0" – "9"].	No.	Sí.

Tabla 2. Descripción de la entidad Llamada.

### ➤ Emergencia

<b>Nombre de la entidad</b>		Emergencia.					
<b>Descripción de la entidad</b>		Situación de riesgo que puede generar víctimas o daños materiales si no se toman medidas inmediatas.					
Nombre del atributo	Descripción	Tipo	Puede ser nulo	Restricciones		Criterio de Selección	
				Clases válidas	Clases no válidas	Múltiple	Única
idEmergenci	Identificador	Identificador.	No.	No aplica.	No	No.	Sí.

## *Capítulo II. Diseño e Implementación*

---

a.	que relaciona las entidades Emergencia y Llamada (que reporta la emergencia).				aplica.		
Prioridad.	Asigna a la orden de radio una prioridad de relevante para su atención inmediata.	Bit.	No.	No aplica.	No aplica.	No.	Sí.
Número de orden de radio.	Representa el número que oficializa la emergencia.	Varchar.	Sí.	["0"- "9"] ["A" -" Z"   "a"- "z"   caracteres especiales] .	No aplica.	No.	Sí.
idEstado del suceso.	Identificador que relaciona las entidades Emergencia y Estado de suceso (en el que se encuentra el suceso).	Identificador. Varchar (Nomenclado).	No.	Actos preparatorios, en ejecución o consumado.	["0"- "9"   caracteres especiales].	No.	Sí.
idTipo probable de suceso.	Identificador que relaciona las entidades	Identificador. Varchar (Nomenclado).	No.	["A" -" Z"   "a"- "z"].	["0"- "9"   caracteres]	No.	Sí.

## Capítulo II. Diseño e Implementación

	Emergencia y Tipo de suceso (al que pertenece el suceso).				especiales].		
idLugar.	Identificador que relaciona las entidades Emergencia y Lugar (donde ocurre el suceso).	Identificador. Varchar (Nomenclado).	Sí.	["0"- "9"]  caracteres especiales] ["A" -" Z"   "a"- "z"].	["0"- "9"   caracteres especiales].	No.	Sí.
idLugar específico.	Identificador que relaciona las entidades Emergencia y Lugar específico (donde ocurre el suceso).	Identificador. Varchar (Nomenclado).	Sí.	["0"- "9"]  caracteres especiales] ["A" -" Z"   "a"- "z"].	["0"- "9"   caracteres especiales].	No.	Sí.
idQuien informa.	Identificador que relaciona las entidades Quien informa (persona denuncia el suceso) y Emergencia.	Identificador. (Nomenclado).	No.	No aplica.	No aplica.	No.	Sí.

## Capítulo II. Diseño e Implementación

idDirección.	Identificador que relaciona las entidades Dirección (ubicación geográfica de donde ocurrió el incidente) y Emergencia.	Identificador.	No.	No aplica.	No aplica.	No.	Sí.
idÁrea de patrullaje.	Identificador que relaciona las entidades Emergencia y Área de patrullaje.	Identificador. Varchar (Nomenclado).	No.	["A" - "Z"   "a" - "z"] ["0" - "9"].	[caracteres especiales].	Sí.	No.

Tabla 3. Descripción de la entidad Emergencia.

### ➤ Persona

<b>Nombre de la entidad</b>		Persona.					
<b>Descripción de la entidad</b>		Datos de las personas.					
Nombre del atributo	Descripción	Tipo	Puede ser nulo	Restricciones		Criterio de Selección	
				Clases válidas	Clases no válidas	Múltiple	Única
Carné de identidad.	Número de identificación de la persona.	Varchar.	Sí.	["0" - "9"].	["A" - "Z"   "a" - "z"].	No.	Sí.
Primer nombre.	Nombre de la persona.	Varchar.	Sí.	["A" - "Z"	["0" - "9"], otros caracteres especiales.	No.	Sí.

## Capítulo II. Diseño e Implementación

Segundo nombre.	Segundo nombre de la persona.	Varchar.	Sí.	"a"- "z".	["0"- "9"], otros caracteres especiales.	No.	Sí.
Primer apellido.	Primer apellido de la persona.	Varchar.	Sí.	["A"- "Z"	["0"- "9"], otros caracteres especiales.	No.	Sí.
Segundo apellido.	Segundo apellido de la persona.	Varchar.	Sí.	["A"- "Z"   "a"- "z"].	["0"- "9"], otros caracteres especiales.	No.	Sí.
Discriminante.	Representa los distintos tipos de persona.	Varchar.	No.	["A"- "Z"   "a"- "z"].	["0"- "9"], otros caracteres especiales.	No.	Sí.

Tabla 4. Descripción de la entidad Persona.

### ➤ Usuario

<b>Nombre de la entidad</b>		Usuario.					
<b>Descripción de la entidad</b>		Trabajadores que tienen permiso sobre las aplicaciones del sistema.					
Nombre del atributo	Descripción	Tipo	Puede ser nulo	Restricciones		Criterio de Selección	
				Clases válidas	Clases no válidas	Múltiple	Única
NombreUsuario.	Usuario del trabajador.	Varchar.	No.	No aplica.	No aplica.	No.	Sí.
idUsuario.	Identificador del usuario.	Identificador.	No.	No aplica.	No aplica.	No.	Sí.
Contraseña.	Contraseña del usuario.	Varchar.	No.	[caracteres especiales   "0"- "9"   "A"- "Z"	No aplica.	No.	Sí.



## Capítulo II. Diseño e Implementación

				"a"- "z"].			
idRol.	Identificador que relaciona las entidades Estado del usuario y Usuario.	Identificador.	No.	No aplica.	No aplica.	No.	Sí.

Tabla 5. Descripción de la entidad Usuario.

### ➤ Turno de Trabajo

<b>Nombre de la entidad</b>		Turno de trabajo.					
<b>Descripción de la entidad</b>		Composición de un turno de trabajo.					
Nombre del atributo	Descripción	Tipo	Puede ser nulo	Restricciones		Criterio de Selección	
				Clases válidas	Clases no válidas	Múltiple	Única
idTurno de trabajo.	Código del turno.	Identificador.	No.	["0"- "9"].	[caracteres especiales], ["A"- "Z"   "a"- "z"].	No.	Sí.
Fecha.	Fecha del turno de trabajo.	Datetime.	No.	[caracteres especiales] y ["0"- "9"] con formato de fecha.	["A"- "Z"   "a"- "z"].	No.	Sí.
idArea del turno de trabajo.	Identificador que relaciona las entidades Turno de	Identificador.	No.	["A"- "Z"   "a"- "z"   "0"- "9"].	[ caracteres especiales].	No.	Sí.

## Capítulo II. Diseño e Implementación

	trabajo y Área del turno de trabajo.						
idHorario del turno.	Identificador que relaciona las entidades Turno de trabajo y Horario del turno.	Identificador.	No.	["A"-"Z"   "a"-"z"   "0"-"9"].	[ caracteres especiales].	No.	Sí.
idJefeTurno.	Identificador del usuario jefe de turno.	Identificador.	No.	No aplica.	No aplica.	No.	Sí.

Tabla 6. Descripción de la entidad Turno de Trabajo.

### Elementos representativos del diseño

- En el submodelo se destaca la representación de relaciones del tipo generalización-especialización:

- ❖ Llamada

Las llamadas que se reciben en el centro se clasifican en circulación, emergencia o informativa. Éstas tienen atributos en común y otros que son propios del tipo de llamada. En la figura 1 se muestra el modelo realizado para representar esta situación.

# Capítulo II. Diseño e Implementación

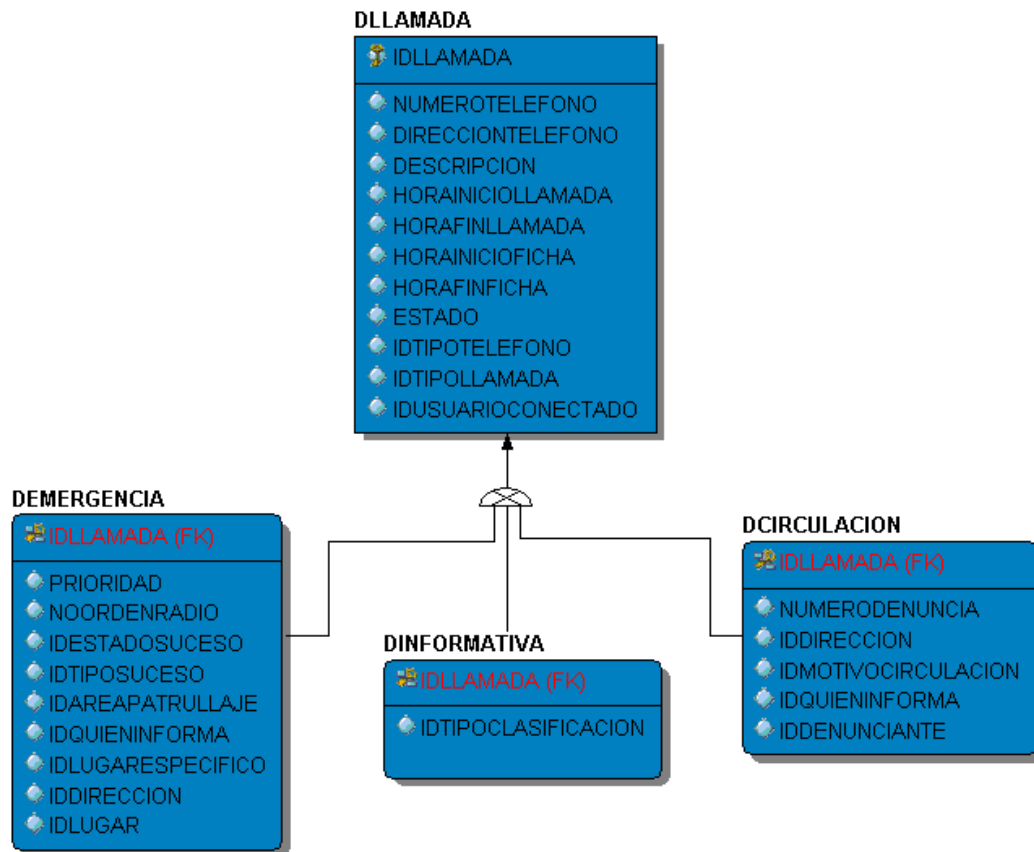


Figura 1. Representación de la generalización Llamada.

## ❖ Circulación

La circulación es un tipo de llamada y, a su vez, se clasifica en circulación de matrícula, objeto, persona y vehículo. Se genera cuando a través de la llamada se recibe información de una persona, objeto, vehículo o matrícula que se encuentra extraviado o implicado en algún delito. Todas las circulaciones comparten atributos y poseen otros que las diferencian. La situación expuesta se representa a través del modelo presentado en la figura 2.

# Capítulo II. Diseño e Implementación

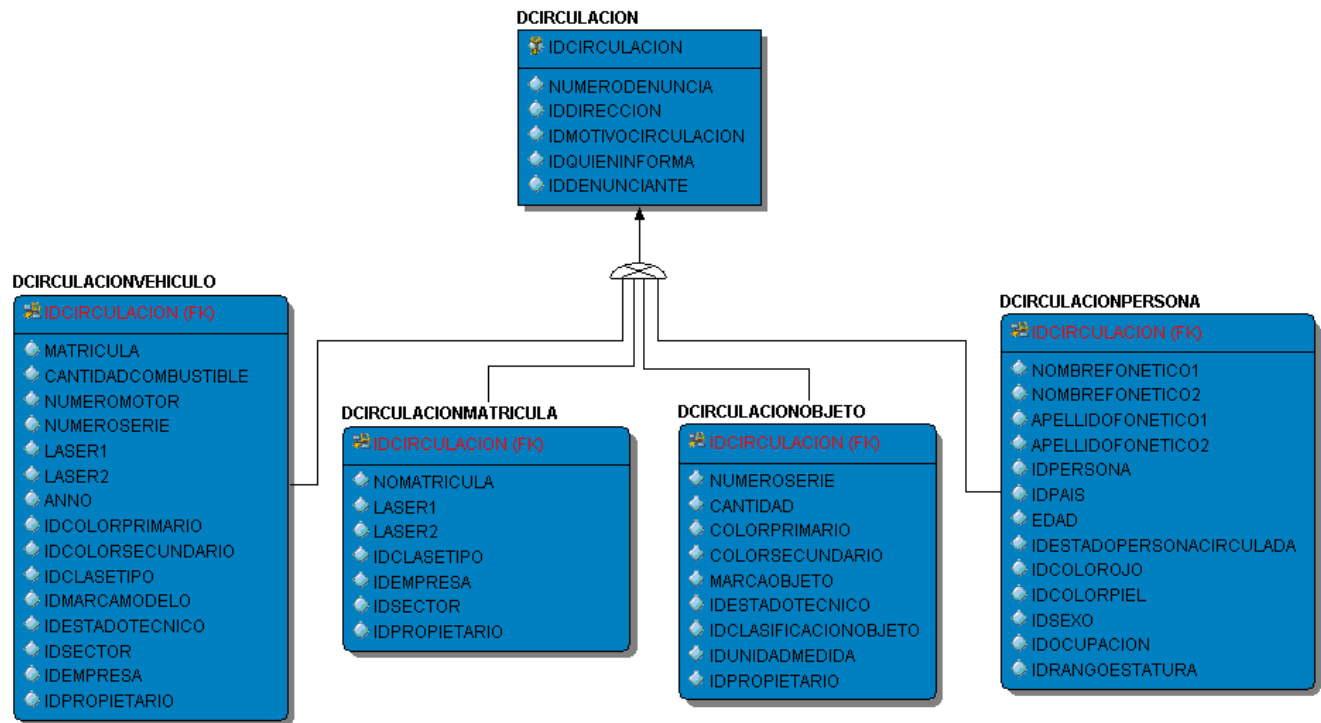


Figura 2. Representación de la generalización Circulación.

## ❖ Trabajador

En la entidad Trabajador se registra la información de todas las personas que laboran en la UPP. Sin embargo, no todos los trabajadores son usuarios del sistema, por lo que se representa la especialización Trabajador<-Usuario para registrar los atributos de estos usuarios (ver figura 3).

# Capítulo II. Diseño e Implementación

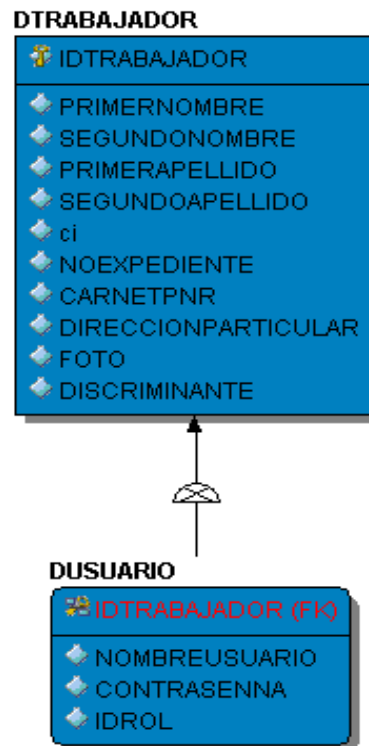


Figura 3. Representación de la generalización Trabajador.

- En las entidades que poseen atributos del contexto que son capaces de identificarlas, se definen llaves alternativas sobre estos campos, permitiendo estos puedan ser usados en algún momento como llaves primarias a pesar de no serlo:
  - ❖ En la tabla dUsuario, donde se almacena la información referente a los trabajadores de la unidad que son usuarios del sistema, se establece el atributo que registra el nombre del usuario (nombreusuario) como llave alternativa especificándole la restricción *UNIQUE*, la cual garantiza que el valor del atributo es único en la entidad..
  - ❖ En la tabla dPersona se tiene el atributo carnet de identidad, el cual es único para las personas, o en todo caso tiene muy poca probabilidad de repetirse. Se define este atributo como llave alternativa, pero en este caso sin especificar la restricción *UNIQUE*, debido a que no existe la seguridad de que los números de carnet de identidad sean únicos en el país.
  - ❖ El atributo número de serie de la entidad dCirculacionObjeto, constituye una llave alternativa

# Capítulo II. Diseño e Implementación

---

para una circulación de objeto. Este atributo se define como tal, sin especificar la restricción *UNIQUE*.

- ❖ En las entidades *dCirculacionVehiculo* y *dCirculacionMatricula*, se cuenta con el atributo matrícula. Este consiste en una sucesión de letras y números que son únicos para cada vehículo. Este atributo se define como llave alternativa especificándole la restricción *UNIQUE*.
- ❖ En la tabla *dRol*, se tiene el atributo nombre de rol, el cual registra el nombre que se define para los roles que se asignarán a los usuarios del sistema. Este campo se define como llave alternativa y se le especifica la restricción *UNIQUE* para asegurar que su valor sea único en la entidad.

### 3. Patrones de diseño

Un patrón de diseño se define como un fragmento de modelo que es profundo y recurrente. Es una solución a un problema específico que se ha mantenido a pesar del tiempo. El uso de los patrones tiene muchos beneficios: proporcionan un enriquecimiento del lenguaje de modelado, documentación mejorada, reducen las dificultades de modelado y hacen de éste un proceso más rápido, posibilitan un resultado correcto y robusto, además permiten la reutilización mediante el uso de los modelos existentes, en lugar de reinventar soluciones. (30)

Es importante mencionar la diferencia entre patrones de diseño y modelos de semillas (*seeds models*), pues en gran parte de la bibliografía de bases de datos se confunden estos términos. Estos últimos son específicos para un contexto dado, proporcionan un punto de partida para las soluciones en el mismo contexto y son realmente útiles, pues aceleran el desarrollo, reducen errores y aportan perspectivas profundas. Ahora, si se trabaja en un contexto diferente, es necesario encontrar modelos de semillas relevantes, comprenderlos, extraer los patrones implícitos en ellos y por último aplicar dichos patrones al trabajo en cuestión (30).

En el diseño de la base de datos, para la elaboración de un modelo fortalecido y asegurar un resultado satisfactorio, se usaron patrones de diseño.

#### 3.1. Llaves subrogadas

# Capítulo II. Diseño e Implementación

---

Con este patrón se plantea que se genere una llave primaria (Primary Key) única para cada entidad, en lugar de utilizar un atributo identificador del contexto en cuestión. En este caso se usa un identificador del tipo GUID, el cual es generado por el gestor Oracle y brinda, por su tamaño y combinación, una alta seguridad, integridad y unicidad; además, está demostrado que no se repite o, en caso contrario, con una probabilidad extremadamente baja, debido a que se genera un valor hexadecimal de 32 caracteres ( $2^{128}$  combinaciones) (31). Esto posibilita además, que las tablas sean más fáciles de consultar a partir del identificador, debido a que todos tienen el mismo tipo en todas las tablas de la base de datos. El identificador se define asignando como valor por defecto del campo llave, la función `rawtohex(sys_guid())`, ejemplo:

```
CREATE TABLE NLUGARGENERAL(  
  
    IDLUGARGENERAL    VARCHAR2(32) DEFAULT    rawtohex (sys_guid())    NOT NULL,  
  
    CODIFICADOR      VARCHAR2(10),  
  
    DESCRIPCION      VARCHAR2(100)          NOT NULL,  
  
    CONSTRAINT PKLUGARGENERAL PRIMARY KEY (IDLUGARGENERAL)  
  
    USING INDEX  
  
);
```

## 3.2. Árboles

### ➤ Árbol fuertemente codificado (*Hardcoded tree*)

Se emplea este patrón para representar las jerarquías y estructuras organizacionales, las cuales generalmente constituyen relaciones de uno a muchos (1-M) y los cambios en la estructura a representar son poco probables. En esta representación es asociada una entidad a cada nivel del árbol, permitiendo tantos niveles como requiera la jerarquía a representar. En la figura 4 se muestra un ejemplo de aplicación del patrón, usado en la representación de las incidencias de desempeño: éstas están agrupadas por clasificaciones de incidencias, a su vez cada clasificación posee un conjunto de tipos de incidencias, las cuales son cometidas por un usuario en un turno dado.

# Capítulo II. Diseño e Implementación

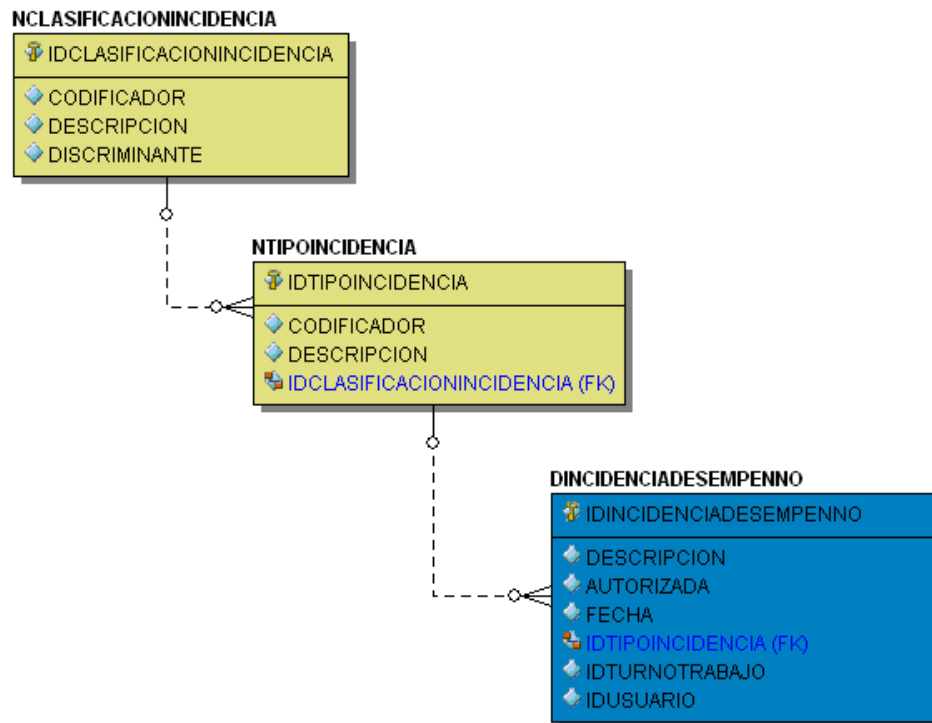


Figura 4. Ejemplo del patrón Hardcoded Tree en la representación de la incidencia de desempeño.

## ➤ Árbol simple (*Simple Tree*)

Se utiliza en los casos donde el árbol se limita a representar una estructura de datos, teniendo en cuenta que los registros a almacenar son del mismo tipo y pueden ser almacenados en la misma entidad, en otras palabras, se establece una relación recursiva sobre la entidad. En la figura 5 se ilustra un ejemplo de cómo se aplica el patrón, empleado para modelar la persistencia de la información relacionada con las llamadas, resolviendo la situación de que las llamadas pueden ser repetidas o tener una llamada asociada. De esta manera se persisten los datos en la misma entidad, estableciéndose una relación recursiva sobre ella misma.



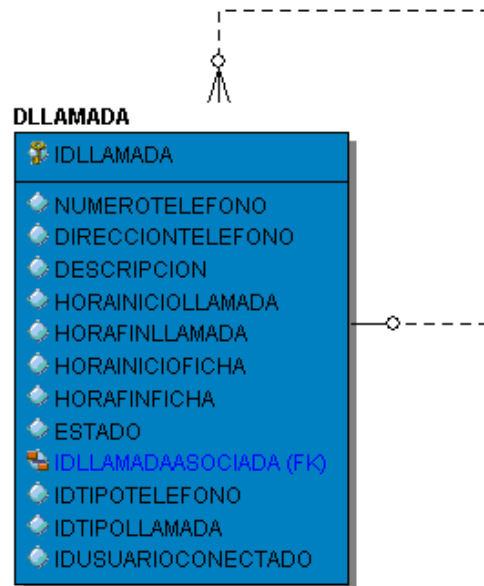


Figura 5. Representación de la aplicación del patrón Simple Tree.

## ➤ **Árbol superpuesto (Overlapping Tree)**

Este patrón representa la pertenencia de un nodo a más de un árbol, donde para cada nodo existe un padre excepto para el nodo raíz (30). Se aplica este patrón en la solución para modelar la relación existente entre los usuarios del sistema y el turno de trabajo. Cada turno tiene un “jefe de turno” y una relación del resto de los usuarios (supervisores, operadoras), los cuales tienen además un jefe asociado (ver figura 6).

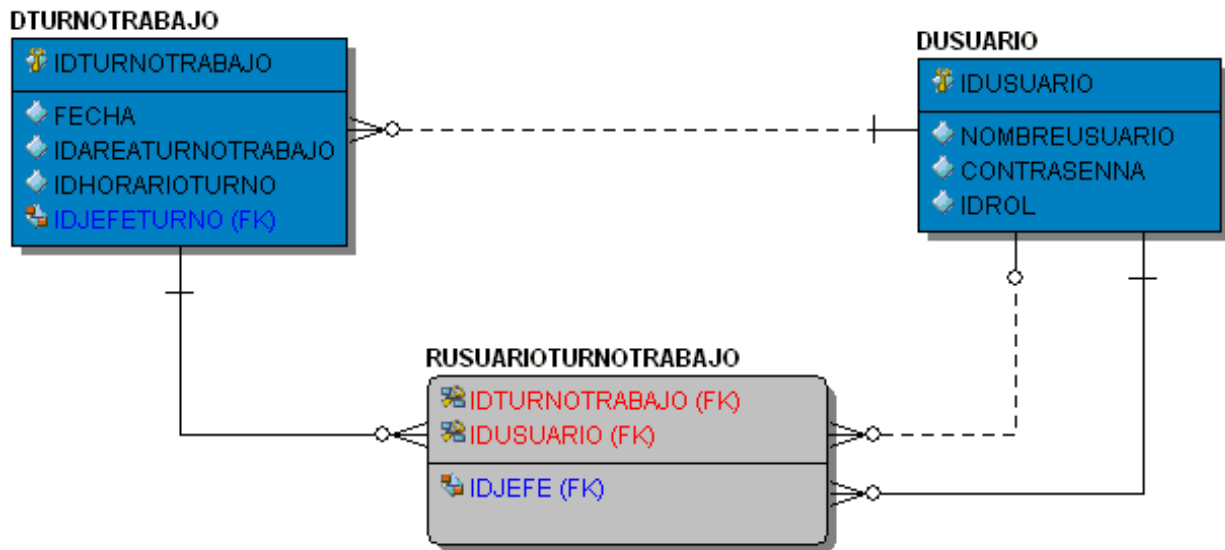


Figura 6. Aplicación del patrón Overlapping Tree en el modelado del turno de trabajo.

### 3.3. Generalizaciones

Se utiliza en la representación de las herencias<sup>13</sup>, usando la variante de crear una entidad para el padre (supertipo) y una para cada hijo (subtipo), estas últimas recibiendo el id de la entidad padre, que además será el suyo propio (figuras 1, 2 y 3). Este patrón es uno de los propuestos para representar la herencia, debido a que las bases de datos relacionales sólo admiten parcialmente la generalización (30).

## 4. Pautas y restricciones del diseño de la Base de Datos

Con el objetivo de alcanzar un mejor entendimiento de los objetos y las relaciones que se establecen en la base de datos, en el diseño propuesto se adoptan una serie de pautas y restricciones, lo que además tributa a la integridad de los datos.

### 4.1. Integridad de datos

Para mantener la precisión y consistencia de los valores de los datos, en la solución todos los atributos hacen uso de los dominios, con tipos de datos estándar y donde se agrupan aquellos que son comunes para varios atributos. En todas las entidades se establece además, que ninguna de las llaves primarias puede tomar valores nulos y en conjunto se tiene la restricción UNIQUE para garantizar que el valor es

<sup>13</sup> Herencia: uno de los conceptos fundamentales de la programación orientada a objetos.

# Capítulo II. Diseño e Implementación

único en la entidad. También se habilita la integridad referencial, esto con el objetivo de garantizar que toda relación se establece entre entidades válidas, que no existan datos perdidos y relaciones mal resueltas, garantizando que estas relaciones no varíen o dejen de existir cuando se modifiquen los datos.

## 4.2. Nomenclatura y descripción

En el diseño del modelo de datos se ha cuidado muy bien la nomenclatura de los objetos para garantizar una organización de éstos en la base de datos. Para lograr este objetivo, se usa una notación clara en las entidades y sus relaciones, con identificadores simples, para así distinguir bien cada elemento dentro de la base de datos.

- Todos los dominios existentes comienzan con la letra “t”. Constituye un ejemplo:
  - ❖ tIdentificadorUnico: es un dominio con tipo de dato GUID, el cual es generado por el gestor Oracle y brinda, por su tamaño y combinación, una alta seguridad, integridad y unicidad. Por este motivo, este dominio se crea para los identificadores de las distintas entidades.
- Para todas las entidades se define como primera letra del nombre: “d” para las tablas de datos, “n” para las tablas nomencladoras y “r” para las tablas que se crean como resultado de las relaciones muchos a muchos (M:N) y las relaciones que involucran más de dos tablas (figura 7).

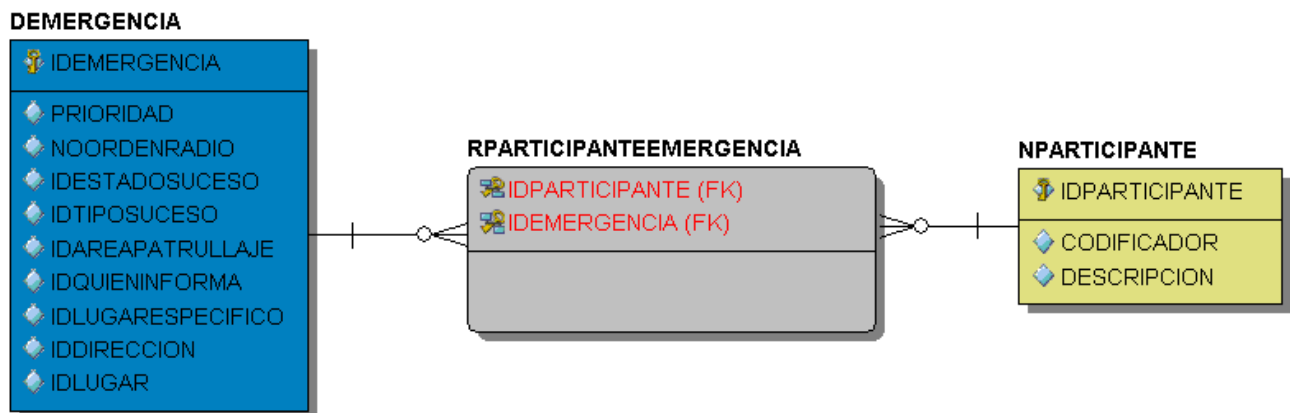


Figura 7. De izquierda a derecha: entidad de datos, relación y nomenclador.

- Las llaves primarias se nombran empezando con las letras “PK” (por las siglas en inglés de *Primary Key*) y concatenándola, sin espacio u otro caracter, al nombre de la entidad que la

# Capítulo II. Diseño e Implementación

---

contiene. Ejemplo: el nombre de la llave de la entidad dLlamada es PKLlamada.

- Las llaves alternativas se nombran comenzando con las letras “AK” (por las siglas en inglés de *Alternate Key*) y concatenándola, sin espacio u otro caracter, al nombre de la entidad que la contiene. Por ejemplo: el nombre de la llave alternativa de la entidad dUsuario es AKUSUARIO.
- Las relaciones se nombran usando el prefijo “FK” (por las siglas en inglés de *Foreign Key*) seguido, sin espacio u otro caracter, primero por el nombre de la entidad origen y luego el nombre de la entidad destino. Ejemplo: el nombre de la relación de nLugarGeneral a nLugar es FKLUGARGENERALLUGAR.
- En todas las entidades primero aparecen las llaves primarias, seguidas por los campos y luego las llaves foráneas.

## 4.3. Normalización

Para lograr la persistencia y recuperación de los datos, sin la existencia de éstos en estado redundante o inconsistente, es fundamental tener en cuenta la normalización. Este proceso consiste en aplicar las formas normales (mencionadas en el capítulo anterior), las cuales proporcionan los criterios para determinar el grado de vulnerabilidad de una tabla a inconsistencias y anomalías lógicas. (32)

El modelo de datos de la solución se normaliza casi en su totalidad hasta tercera forma normal. La ocurrencia de excepciones a esta disposición es intencional y persigue ganar en rendimiento en la gestión de la información; además, no afecta en gran medida el espacio requerido para almacenar los datos ni la explicación de la base de datos.

## 5. Optimización

De acuerdo con las características de los sistemas informáticos de gestión de emergencias, y atendiendo a que la base de datos que se propone será la encargada de soportar las funcionalidades de un sistema de este tipo, es evidente que el rendimiento de la base de datos y la disponibilidad de los datos constituyen factores importantes en la presente solución. En aras de optimizar el tiempo de respuesta ante consultas, se decide utilizar técnicas que reducen estos tiempos en tablas con grandes volúmenes de información o que son consultadas frecuentemente.

# Capítulo II. Diseño e Implementación

## 5.1. Particionado

En el anexo 1 se muestra un resumen de las estrategias de particionado y extensiones que brinda Oracle, además de la forma en que se organizan los datos en cada caso.

En la base de datos que se propone se utiliza el particionado en tablas que almacenan o almacenarán gran cantidad de registros. En la tabla 7 se muestran las tablas que fueron particionadas, especificando la estrategia de particionado utilizada en cada caso y su descripción.

Tablas particionadas en la solución		
Entidad	Técnica de Particionado	Descripción
dLlamada	Compuesto ( <i>interval-list</i> )	Esta entidad es crítica en la base de datos en cuanto a su crecimiento, pues es la tabla donde quedan registradas todas las llamadas que se reciben en el área de atención a llamadas. Ante esta situación se define un particionado compuesto con lo cual se pretende elevar los niveles de rendimiento ante las consultas a esta entidad. Se realiza primero un particionado por intervalo sobre la columna HORAINICIO LLAMADA, subparticionando luego por lista sobre la columna IDTIPO LLAMADA. El intervalo definido para cada partición es de un mes, mientras que los parámetros para crear las particiones por lista son los tipos de llamadas que se reciben.

## Capítulo II. Diseño e Implementación

dTurnoTrabajo	Intervalo	Es la entidad donde se registran los turnos de trabajo confeccionados. Se realiza un particionado por intervalo sobre la columna fecha. El intervalo definido para este caso es de un año. De esta forma, cuando se registre un turno en el que la fecha no coincida con ninguna de las particiones existentes, se creará la partición necesaria automáticamente.
---------------	-----------	---

Tabla 7. Tablas particionadas en la solución.

El script de creación de estas tablas queda de la siguiente manera:

➤ Tabla dLlamada

```
CREATE TABLE DLLAMADA(  
    IDLLAMADA          VARCHAR2(32) DEFAULT rawtohex(sys_guid()) NOT NULL,  
    HORAINICIO LLAMADA  TIMESTAMP(0) NOT NULL,  
    IDTIPOLLAMADA      VARCHAR2(32),  
    ...  
    "resto de los atributos"  
    ...  
    CONSTRAINT PKLLAMADA PRIMARY KEY (IDLLAMADA)  
    USING INDEX  
)  
PARTITION BY RANGE (HORAINICIO LLAMADA) INTERVAL (NUMTOYMINTERVAL(1,'MONTH'))  
SUBPARTITION BY LIST (IDTIPOLLAMADA)
```

# Capítulo II. Diseño e Implementación

## SUBPARTITION TEMPLATE

```
(SUBPARTITION INFORMATIVA VALUES ('6C9554AA1D756266E040240A6A137F65'),
SUBPARTITION CIRC_VEHICULO VALUES ('6C9554AA1D796266E040240A6A137F65'),
SUBPARTITION CIRC_PERSONA VALUES ('6C9554AA1D776266E040240A6A137F65'),
SUBPARTITION CIRC_OBJETO VALUES ('6C9554AA1D786266E040240A6A137F65'),
SUBPARTITION CIRC_MATRICULA VALUES ('6E8EF65032CB69A5E040240A6A131460'),
SUBPARTITION EMERGENCIA VALUES ('6C9554AA1D766266E040240A6A137F65'),
SUBPARTITION OTRAS VALUES (DEFAULT)
)

(PARTITION P0902 VALUES LESS THAN (TO_DATE('2009-03-01 00:00:00','YYYY-MM-DD HH24:MI:SS')));
```

Para definir el intervalo sobre la columna seleccionada, que es de tipo “fecha”, se utiliza la función NUMTOYMINTERVAL (n,'char'). Ésta convierte el número n a un intervalo de mes o año, dependiendo del literal (MONTH, YEAR) que se le pase como segundo parámetro. El valor especificado como parámetro de las subparticiones es el identificador correspondiente a los distintos tipos de llamadas; el valor “DEFAULT” es el parámetro de la subpartición que se crea para aquellos valores que no se corresponden con los especificados en las anteriores. De esta manera, si posterior a la creación de la entidad, surge algún valor nuevo, se asegura que pueda ser almacenado en la entidad.

### ➤ Tabla dTurno

```
CREATE TABLE DTURNOTRABAJO(
    IDTURNOTRABAJO    VARCHAR2(32) DEFAULT    rawtohex(sys_guid())    NOT NULL,
    FECHA              TIMESTAMP(6) NOT NULL,
    ...
    “resto de los atributos”
    ...,
```

# Capítulo II. Diseño e Implementación

---

```
CONSTRAINT PKTURNOTRABAJO PRIMARY KEY (IDTURNOTRABAJO)

USING INDEX

)

PARTITION BY RANGE (FECHA) INTERVAL (NUMTOYMINTERVAL(1,'YEAR'))

(PARTITION P0902 VALUES LESS THAN (TO_DATE('2009-03-01 00:00:00','YYYY-MM-DD HH24:MI:SS')));
```

La columna por la cual se particiona también es tipo “fecha”. Para definir el intervalo de las particiones se utiliza la misma función que en el caso anterior pero en este caso con un intervalo de un año.

## 5.2. Procedimientos almacenados

Los procedimientos almacenados son bloques reutilizables de código PL/SQL almacenados en la base de datos, los cuales son ejecutados por las aplicaciones. Pueden mejorar la seguridad de los datos, pues el usuario tiene permiso de ejecución del procedimiento, en lugar de acceder directamente a las tablas. Elevan los niveles de eficiencia de la base de datos, pues se ejecutan directamente en el lado del servidor de base de datos, el cual garantiza el acceso directo a los datos que necesita manipular, enviando solamente el resultado final al usuario y disminuyendo el tráfico de red (26).

En el SAEM se realizan consultas a la base de datos que reducen el rendimiento de la aplicación por involucrar varias tablas o por el volumen de datos a enviar como resultado de la consulta para luego ser procesados en la capa de negocio.

En estos casos para optimizar la respuesta ante estas consultas se definieron los siguientes procedimientos almacenados:

- `coor_cercanas`: posee tres parámetros de entrada de tipo numérico (coordenada x, coordenada y, radio) y uno de salida del tipo cursor (`cercanas`). Consiste en, dadas las coordenadas de una emergencia, devolver las emergencias que se encuentran alrededor de dichas coordenadas y dentro del radio entrado por parámetro.
- `estadisticas_oper_turno`: posee dos parámetros de entrada de tipo texto (identificador de una operadora, identificador de un turno) y seis de salida de tipo numérico. Al recibir los identificadores de la operadora y del turno, devuelve en los parámetros de salida las estadísticas referentes a



## Capítulo II. Diseño e Implementación

---

éstos: cantidad de llamadas recibidas, cantidad de emergencias, cantidad de circulaciones, cantidad de informativas, cantidad de incidencias de desempeño y cantidad de reconocimientos.

- `list_subordinados`: tiene dos parámetros de entrada de tipo texto (identificador de un jefe, identificador de un turno) y uno de salida de tipo cursor (lista de subordinados). Cuando recibe los identificadores por parámetro, retorna en el parámetro de salida todos los subordinados de un jefe en el turno dado.

### 6. Conclusiones

En el presente capítulo se realizó una descripción del funcionamiento del área de negocio a modelar, lo que permitió sentar las bases para conformar el diseño de base de datos del Sistema de Atención a Emergencias. A partir de un conjunto de pautas y patrones de diseño se logró conformar el submodelo de datos asociado al área de Atención a Llamadas, cumpliéndose con los requerimientos del cliente, así como con las restricciones y condiciones presentes dentro del alcance definido. En el diseño propuesto cuando las tablas manejan grandes volúmenes de información se optimiza el tiempo de respuesta ante consultas haciendo uso de la técnica de particionado y, con el mismo fin de ganar en rendimiento, se implementan procedimientos almacenados para optimizar la respuesta ante consultas complejas.

# Capítulo III. Configuración y Seguridad del Servidor

---

## CAPÍTULO III: CONFIGURACIÓN Y SEGURIDAD DEL SERVIDOR.

### 1. Introducción

Una vez definido el modelo de datos que permitirá el almacenamiento de la información que se maneja en la UPP, ocupa un papel fundamental decidir la distribución de la misma, así como mecanismos que permitan salva-guardar los datos. En el presente capítulo se muestra cómo se encuentra representada lógicamente la información que se posee en la base de datos, además de las políticas de salva y recuperación aplicadas para conservar los archivos de datos. También se trata el aspecto de la seguridad en la base de datos, tomándose como referencia al usuario, su método de autenticación así como los permisos que poseerá entre otros aspectos. En general, son propuestas de configuración que persiguen aumentar los niveles de disponibilidad y seguridad de la base de datos.

### 2. Almacenamiento

#### 2.1. Espacios de tablas (*Tablespaces*)

Como un mecanismo de organización de la información en un nivel superior, se tienen los espacios de tablas, que son divisiones lógicas u objetos de la base de datos que representan la agrupación de uno o más archivos físicos. Toda base de datos Oracle contiene al menos dos espacios de tablas: SYSTEM y SYSAUX (33). El espacio en disco que se le asigna a un espacio de tablas cuando se crea, es reservado inmediatamente se use o no. Oracle puede además, gestionar el crecimiento dinámico de los espacios de tablas en caso de que el espacio asignado pueda ser insuficiente. Los parámetros para el redimensionamiento se especifican en el momento de su creación.

#### 2.2. Espacios de tablas para el SAEM

En la solución propuesta, teniendo ciertas consideraciones que están encabezadas por el factor eficiencia, se definen tres espacios de tablas:

- TS\_DATOS: Se define para las tablas de datos o negocio, las cuales son constantemente modificadas por las distintas funcionalidades de la aplicación.
- TS\_NOMENCLADORES: Se crea para almacenar los nomencladores, tablas que sólo son modificadas por un módulo de administración, mientras que el resto de los subsistemas sólo las utilizarán para extraer información. Teniendo esta situación, se decide tenerlas en un espacio de

# Capítulo III. Configuración y Seguridad del Servidor

---

tablas aparte, para separarlas del monto de información que van a almacenar las tablas de datos.

- TS\_INDICES: En este espacio de tablas se guardarán los índices. Esta decisión cumple además con requisitos definidos por el cliente en la política de lineamientos tecnológicos de la DTS.
- TS\_ACTIVO: Se define para albergar las tablas tanto de datos como nomencladoras más activas de la base de datos, posibilitando que el acceso a las mismas sea mucho más rápido, pues existe un menor volumen de información en el espacio de tablas.

Esta división persigue eliminar la fragmentación, así como minimizar la competencia por el disco al separar los grupos de segmentos de datos sus segmentos de índices correspondientes; favorece las copias de seguridad y la recuperación, contribuyendo además a la limpieza y claridad de la base de datos.

### 3. *Flashback y Undo*

Oracle provee la tecnología *flashback*, la cual permite manipular objetos de la base de datos y es útil para ver estados pasados de los datos sin la necesidad de la existencia de una copia de seguridad. Las características del *flashback* permiten deshacer los cambios realizados accidentalmente por el usuario a la base de datos, de manera rápida y con menor costo de disponibilidad de la base de datos que una recuperación real (la recuperación que se realiza es lógica).

La información sobre los cambios realizados en la base de datos, efectos de cada actualización y los datos que fueron sobrescritos, se conserva en registros de acción de transacciones conocidos como *undo*. Cuando una declaración *undo* es establecida, los cambios realizados por la última transacción son ignorados, quedando la base de datos en el estado en que estaba anterior a ésta. Oracle provee un mecanismo completo para la administración automática de los archivos *undo*, administrando sus segmentos y retención el mayor tiempo posible, así como el espacio en dispositivos físicos requeridos por éstos (33).

### 4. Mecanismos de Salva (*Backup*)

Realizar *backups*, consiste en hacer una salva de la información relevante que poseemos en la base de datos. Como información relevante se puede considerar los archivos de datos, los datos de los usuarios y los archivos de parámetros del servidor, así como los archivos de control que contienen la configuración del servidor (33).

# Capítulo III. Configuración y Seguridad del Servidor

---

El principal objetivo de realizar copias de seguridad es resguardar la información previendo pérdidas inesperadas de información y son necesarios porque es la única forma de salvar el contenido de los archivos de datos<sup>14</sup>.

## 4.1. Tipos de *Backups*

Oracle permite realizar salvallas que atienden a tres criterios básicos:

- I). *Backup* en caliente o frío (conocido como abierto o cerrado). La salva en caliente se realiza mientras la base de datos está abierta y funcionando en modo ARCHIVELOG, debe realizarse cuando la carga de la base de datos sea pequeña. Las salvallas en frío implican parar la base de datos en modo normal y copiar todos los ficheros sobre los que se asienta. Antes de pararla hay que parar también todas las aplicaciones que la estén utilizando y una vez realizada la copia de los ficheros, puede volver a arrancarse.
- II). Total o parcial. Se entiende como salva total de la base de datos a aquella que se le ha hecho a todos los archivos de control y de datos, mientras que se hace una salva parcial cuando sólo se le realiza a un archivo, o a una parte que no conformen la totalidad de los existentes.
- III). Completa o incremental. Cuando se realiza una salva completa, se copian todos los archivos seleccionados y se marcan como copiados. Si se realiza una copia de seguridad incremental sólo se copian los archivos creados o modificados desde la última copia completa o incremental. La combinación de copias de seguridad normal e incremental utiliza el mínimo espacio de almacenamiento posible y es el método de copia de seguridad más rápido.

Como en diversas ocasiones se necesita que la base de datos esté disponible las 24 horas del día, todos los días del año, es muy útil la realización de *backup* en caliente, tarea que puede realizar mediante la consola de administración o a través del sitio que publica el servidor de base de datos Oracle.

Un ejemplo de realización de *backup* de un espacio de tablas mediante la consola es el siguiente:

```
SQL> alter tablespace TS_DATOS begin backup;
```

```
SQL> alter tablespace TS_DATOS end backup;
```

---

<sup>14</sup> Los archivos de control y de reconstrucción pueden multiplexarse, poniendo las copias en unidades de discos diferentes (34).

# Capítulo III. Configuración y Seguridad del Servidor

---

## 4.2. Estrategia de *Backup* de la solución

Aprovechando las posibilidades que brinda Oracle para la realización automática de *backup* y teniendo en cuenta las consideraciones del cliente, se propone la estrategia siguiente:

- Realizar copias completas e incrementales.
- Las copias completas se realizarán una vez a la semana, en el horario y día de la semana que menos demanda tenga la base de datos. Las incrementales por su parte se realizarán el resto de los días de la semana, también en los horarios de menor demanda de la base de datos.
- Se realizarán dos copias de seguridad y se guardarán en dispositivos (discos) diferentes.
- Las salvas, en cumplimiento con políticas definidas por el cliente, se conservarán por un período de cinco años.

La configuración de la salva automática se puede realizar a través del sitio de administración que publica el servidor Oracle.

## 5. Métodos de Recuperación

Se entiende por recuperación a la acción de extraer la información desde una copia de seguridad hacia el lugar donde fue creado (34). En dependencia del tipo de archivo que se haya dañado o perdido por cualquier motivo, se realizan distintas acciones que a continuación se exponen.

### 5.1. Recuperación ante pérdida de archivo de control<sup>15</sup>

En los archivos de control se referencian los archivos de reconstrucción (nombre y dirección física), así como los datos de la instancia en uso. Este archivo por su importancia debe estar multiplexado y para comprobar que así es, se ejecuta en la consola de administración (SQL\*PLUS), el siguiente código:

```
SQL> select name from v$controlfile;
```

---

<sup>15</sup>En los archivos de control es donde se encuentran los datos de los archivos de reconstrucción , así como los datos de la instancia en uso

# Capítulo III. Configuración y Seguridad del Servidor

---

Se mostrará la lista de archivos de control que en caso de ser sólo uno, el elemento que aparezca como resultado, nos está indicando que es necesario crear copias del archivo de control, modificando el archivo `init.ora`. El siguiente código es un ejemplo de multiplexación del archivo de control.

```
#####  
  
# File Configuration  
  
#####  
  
CONTROL_FILES = (/u01/oracle/prod/control01.ctl,  
  
/u02/oracle/prod/control02.ctl,  
  
/u03/oracle/prod/control03.ctl)
```

Para restaurar la base de datos ante la pérdida o fallo de un archivo de control se siguen los siguientes pasos:

- Detener la base de datos.
- Sobre escribir el archivo de control dañado con uno nuevo.
- Inicializar la instancia de la base de datos en estado *nomount*.
- Cambiar los parámetros de los archivos de control incluyendo en ellos el nuevo archivo de control copiado.
- Detener de nuevo la base de datos.
- Inicializar la base de datos en modo abierto para su uso normal. (34)

Es fácil de detectar cuando ha ocurrido un error en los archivos de control, pues la instancia de la base de datos se detiene inmediatamente y cuando se intenta inicializar nuevamente, comienza a inicializar la instancia en modo *nomount* y se detiene.

## 5.2. Recuperación ante pérdida de archivo de reconstrucción

# Capítulo III. Configuración y Seguridad del Servidor

---

Si se tienen los archivos de reconstrucción multiplexados, como es el caso, la pérdida o fallo de uno de ellos tiene muy poca o ninguna repercusión sobre la base de datos, Oracle seguirá trabajando normalmente, pero si notificará la pérdida.

La forma de saber cuántos archivos de reconstrucción se tienen, es similar a la de los archivos de control:

```
SQL> select * from v$logfile;
```

Si algún archivo de reconstrucción presenta estado INVALID, es que está fallando, y la forma de arreglarlo es utilizando el comando CLEAR, a continuación se explica el procedimiento completo.

```
SQL> alter database clear logfile group #_del_archivo_con_fallos;
```

Si el archivo a reparar está activo, ejecutando las siguientes líneas de código se resuelve el problema:

```
SQL>alter system switch logfile;
```

```
SQL>alter system checkpoint;
```

```
SQL>alter system archive log all; (34)
```

## 5.3. Recuperación ante pérdidas de archivos de datos

Dentro de las formas de recuperación de archivos de datos se encuentran la recuperación en modo No Archive Log, donde se recomienda borrar el espacio de tabla dañado (en caso de que no se tenga un *backup* de los archivos de reconstrucción del mismo) y crear los archivos con la siguiente sentencia:

```
SQL> alter database open resetlogs;
```

Mientras que en modo Archive Log el procedimiento es el siguiente:

- Buscar que archivo está dañado mediante el comando:

```
SQL> select name, online_status, error from v$datafile join v$recover_file using (file#); (34)
```

- En caso de existir algún archivo dañado:
- Se monta la base de datos y, estando en estado *closed*, se retiran los archivos defectuosos.
- Se abre la base de datos. Luego se colocan y recuperan los archivos dañados y se ponen online

# Capítulo III. Configuración y Seguridad del Servidor

---

los mismos. (34)

## 6. Implementación de la Seguridad

Dentro de los consejos referentes a la seguridad brindados por la compañía Oracle en su Guía Oficial, Oracle Database Security Guide, se encuentran los siguientes:

- Habilitar la Auditoría por defecto.
- Utilizar contraseñas fuertes para los usuarios de la base de datos.
- No utilizar el privilegio CREATE EXTERNAL JOB como público.
- Realizar cambios en los parámetros de inicialización que tiene por defecto nuestra base de datos al crearla (ver anexo 2). (35)

### 6.1. Usuario de la Base de Datos

Para conectarse a la base de datos Oracle, es necesario contar con al menos un usuario. Para la interacción entre la aplicación y la base de datos, se define un usuario en esta última que tendrá sólo los privilegios necesarios sobre los objetos de la misma. El usuario se define a través del código SQL siguiente:

```
CREATE USER "nombre de usuario"  
  
IDENTIFIED BY "contraseña de usuario"  
  
DEFAULT TABLESPACE USERS  
  
TEMPORARY TABLESPACE TEMP  
  
PROFILE "perfil de usuario";  
  
GRANT "rol de usuario" TO "nombre de usuario";
```

Con la última línea de código se asigna un rol al usuario. El rol es quien tiene asignados los permisos sobre los objetos y las transacciones que puede realizar. De esta manera es más fácil el manejo de los privilegios de los usuarios, sólo es necesario asignar el rol al usuario(s) al que se quiera asignar los permisos. En la versión actual de la aplicación, por políticas del cliente solo existe un usuario, pero si en



# Capítulo III. Configuración y Seguridad del Servidor

---

versiones posteriores se definieran nuevos usuarios, y éstos compartieran los mismos permisos que el usuario actual, simplemente se le asignará a los nuevos usuarios el rol existente en lugar de tener que asignar cada uno de los permisos directamente a éstos.

## 6.2. Métodos de Autenticación

A través del proceso de autenticación los disímiles usuarios que pueda tener la base de datos Oracle podrán conectarse y realizar sus funciones. La autenticación significa verificación de identidad de alguien (usuario, dispositivo, u otra entidad) que quiere usar datos, recursos o aplicaciones. La autenticación de identidad del usuario es imperativa en ambientes distribuidos, sin la cual puede considerarse poca la confianza en la seguridad de red. (36)

El proceso de autenticación se puede realizar de varias maneras, por ejemplo, a través de los usuarios definidos en el Sistema Operativo, autenticación a través de la red por SSL o usando servicios terceristas como Kerberos<sup>16</sup>, Radius<sup>17</sup> o servicios basados en directorios. También se puede realizar a través de la base de datos Oracle.

### 6.2.1. Autenticación por la Base de Datos de Oracle

La autenticación por la BD previene el uso desautorizado de ésta, ya que la conexión se negará si el usuario proporciona una contraseña incorrecta (37). Este proceso es apoyado además por varias opciones que incluye este tipo de autenticación:

- Encriptación de la contraseña durante la conexión. Esta protección siempre está presente por defecto.
- Bloqueo de cuenta.
- Control de tiempo de vida de la contraseña y expiración.
- Historial de contraseñas.
- Comprobación de complejidad de contraseñas. (38)

---

<sup>16</sup> Sistema de la autenticación tercerista confiable que confía en los secretos compartidos. Proporciona las capacidades simples de encendido de señales, almacenamiento de contraseñas centralizado, la autenticación de database links y refuerza la seguridad de las PC. (36)

<sup>17</sup> Protocolo ligero estándar usado para la autenticación de usuario, autorización y manejo de cuentas. (36)

# Capítulo III. Configuración y Seguridad del Servidor

---

El usuario antes creado se autenticará a través de la base de datos utilizando una contraseña definida en el script de creación del mismo.

## 6.3. Políticas de seguridad para contraseñas

La contraseña del único usuario con el que se contará debe tener un elevado nivel de fortaleza, pues la debilidad de la misma comprometería la seguridad de la información registrada en la base de datos y como consecuencia el funcionamiento del trabajo que se realiza en la UPP. Considerando esta situación se recomienda:

- Crear contraseñas cuyo tamaño oscile entre los 8 y los 30 caracteres.
- Crear contraseñas haciendo uso de caracteres como el guión bajo (\_), el signo de peso (\$) y el signo de número (#).
- Incluir al menos un dígito, un carácter alfanumérico y un signo de puntuación en la contraseña.
- No comenzar las contraseñas con números.
- No usar palabras reservadas de Oracle en las contraseñas. (35)

Para asegurar algunas de estas recomendaciones, como parte de la solución se implementa una función de verificación de complejidad de contraseñas. Ésta comprueba que la contraseña cumpla los siguientes requisitos:

- La contraseña no puede ser igual que el nombre de usuario.
- La longitud mínima debe ser de ocho caracteres.
- La contraseña no puede estar en la lista definida como contraseñas comunes.
- La contraseña debe tener al menos una letra, un dígito y un signo de puntuación.

Si se intenta violar alguna de las restricciones anteriores, el gestor prohíbe la asignación de la nueva contraseña y lanza un error informando al usuario.

## 6.4. Perfiles de usuario

# Capítulo III. Configuración y Seguridad del Servidor

---

Los perfiles de usuario tienen el objetivo de limitar los recursos del sistema a los usuarios de la base de datos, así como las secciones concurrentes que el usuario puede establecer, el tiempo de procesamiento del CPU disponible a la sección del usuario y la cantidad de entradas/salidas lógicas disponibles a cada usuario (38), es decir, actúan como medida para aumentar la disponibilidad de la base de datos.

En el caso de la solución, se define un perfil para el usuario del SAEM. En este perfil se limita, entre otros aspectos, el número de sesiones concurrentes por usuario, tiempo en que una sesión puede permanecer inactiva y el número de intentos fallidos de autenticación del usuario.

## 6.5. Roles y privilegios

Los privilegios son reglas asociadas a los objetos de un equipo o red, como archivos y carpetas. Determinan si se puede obtener acceso a un objeto y lo que se puede hacer con él (39). Se le pueden asignar a un usuario individual o a grupos de usuarios y a roles.

Los roles son un conjunto de permisos que se unen para mayor comodidad. De esta forma se acota el alcance de un usuario concreto, y por tanto el daño que puede llegar a hacer (40).

En la base de datos se define un rol para el SAEM donde se permite realizar las operaciones *insert*, *select*, *update* y *delete* sobre todas las entidades, además del permiso de *execute* sobre paquetes, procedimientos y funciones.

## 7. Conclusiones

En el capítulo se han expuesto aspectos de configuración del servidor. Se reflejó la política de salva propuesta y las formas de recuperación ante distintos tipos de fallos o daños, así como varios aspectos de seguridad manejados en el servidor para garantizar la disponibilidad y seguridad de los datos tales como: la autenticación, los recursos y privilegios asignados a los usuario, y políticas de seguridad para las contraseñas. Las propuestas de configuración realizadas, en todos los casos se decidieron teniendo en cuenta requisitos no funcionales expuestos por el cliente, los elementos reflejados en los lineamientos tecnológicos del mismo y recomendaciones que hace el proveedor del gestor utilizado en la solución.

# Capítulo IV. Validación de la Propuesta

---

## CAPÍTULO IV: VALIDACIÓN DE LA PROPUESTA

### 1. Introducción

La validez de un estudio es la cualidad que lo hace creíble y da testimonio del rigor con que se realizó (41). En este capítulo se describen los aspectos tenidos en cuenta para la validación de la solución. Se realiza mediante pruebas teóricas al modelo de datos, así como funcionales a la base de datos.

### 2. Prueba de normalización al modelo de datos

Para validar el modelo de datos se tuvo en cuenta la normalización, comprobándose en cada una de las entidades el cumplimiento de reglas relacionadas con los atributos y la dependencia funcional de éstos con respecto a la llave primaria, las cuales se definen en la primera, segunda y tercera forma normal.

Después de realizar esta tarea se comprobó que la base de datos propuesta cumple con lo expresado en el capítulo 2, epígrafe 4.3).

### 3. Pruebas de integridad

En aras de comprobar que el modelo de datos cumple con las restricciones de integridad expresadas en el epígrafe 4.1 del capítulo 2, se realizaron pruebas que evidencian su cumplimiento. Estas pruebas verifican que las operaciones básicas en la base de datos (insertar, modificar y eliminar), se realizan correctamente y sin ocasionar corrupción en los datos, que en cada campo se puede insertar sólo el tipo de dato definido en cada caso, validando además, que cada entidad tiene una llave que la identifica y cuyo valor es único en la entidad.

#### 3.1. Integridad referencial

Para controlar la integridad referencial, en todas las relaciones se especificó la opción *NO ACTION* en los siguientes casos:

- ✓ Si en una entidad se elimina una tupla cuyo identificador es llave foránea de al menos una tupla en otra entidad, o se modifica el identificador de una tupla, que es llave foránea de al menos una tupla en otra entidad, la eliminación o modificación se prohíbe y el gestor emite un error como el que se muestra en la figura 8.

# Capítulo IV. Validación de la Propuesta

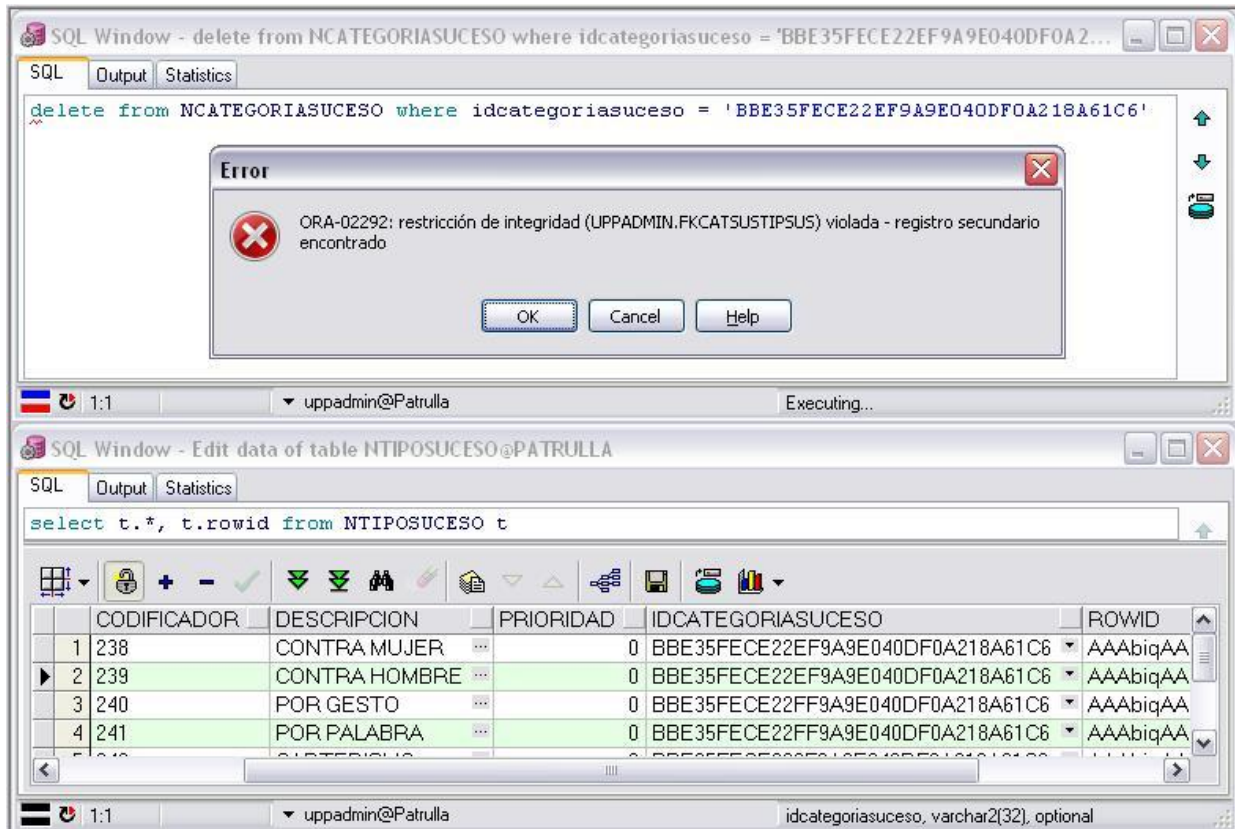


Figura 8. Error al tratar de eliminar una tupla que tiene un registro secundario.

- ✓ Cuando se inserta una tupla que tiene una llave foránea o se modifica este valor en una tupla existente, el valor tiene que existir en la tabla relacionada por esta llave, de lo contrario la inserción o modificación se prohíbe y el gestor lanza un error como el que se muestra en la figura 9.

# Capítulo IV. Validación de la Propuesta

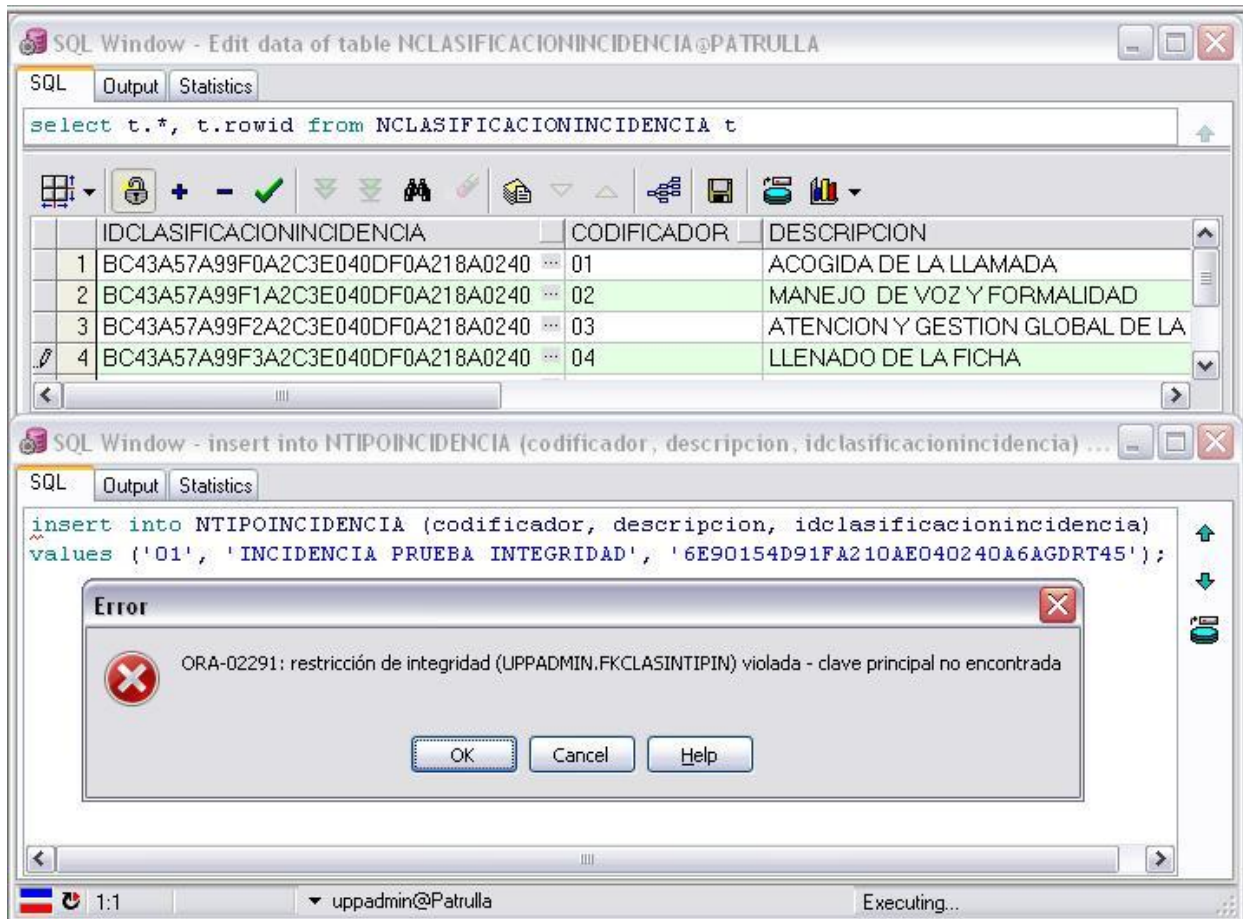


Figura 9. Error al tratar de insertar un registro con un valor de llave foránea que no existe.

## 3.2. Integridad de dominio

- ✓ Utilizando la funcionalidad del ER/Studio que permite validar el modelo de datos, se comprobó que todos los atributos hacen uso de dominios. Éstos restringen el tipo de dato para cada campo, restricciones DEFAULT, si es un dato requerido o no, y otras comprobaciones.
- ✓ El dominio tEntero por ejemplo, define que los campos que lo utilizan son de tipo de dato numérico y que es un dato requerido. Si se intenta insertar datos que violen estas restricciones se emite un error por parte del gestor (ver figura 10).

# Capítulo IV. Validación de la Propuesta

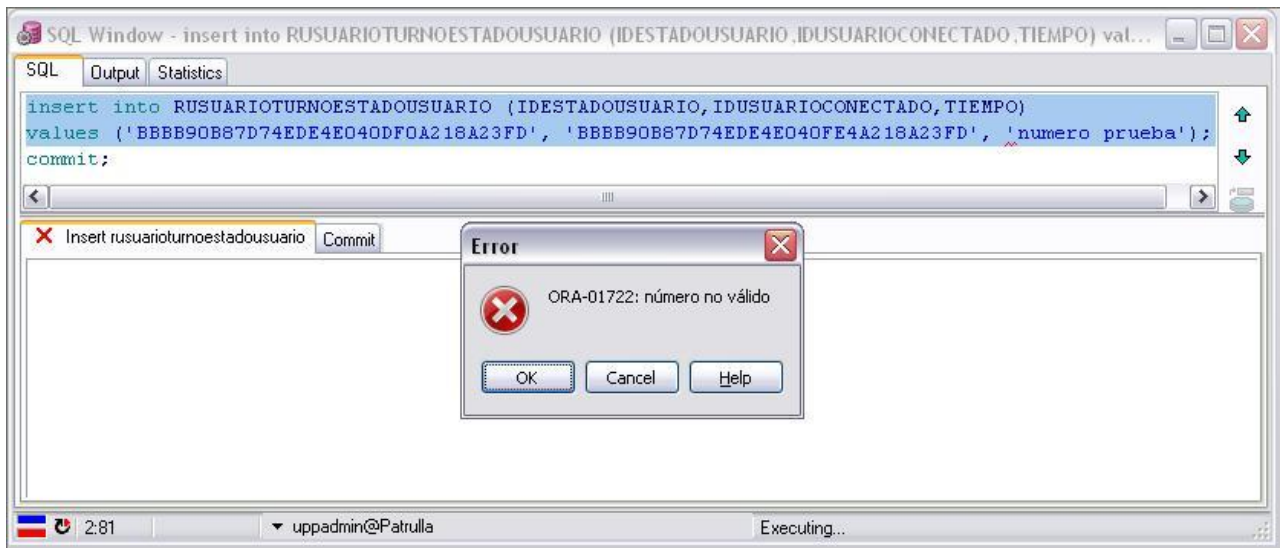


Figura 10. Error al intentar insertar un valor no numérico en un campo que usa el dominio tEntero.

### 3.3. Integridad de entidad

- ✓ Si se intenta ingresar algún valor nulo en cualquier llave primaria, el gestor prohíbe la transacción y lanza un error notificándolo (ver figura 11).



Figura 11. Error emitido al querer insertar el valor nulo en una llave primaria.

- ✓ Si se intenta duplicar un valor de llave primaria en cualquiera de las entidades, el gestor evita la

# Capítulo IV. Validación de la Propuesta

transacción y emite un error para notificarlo (ver figura 12).

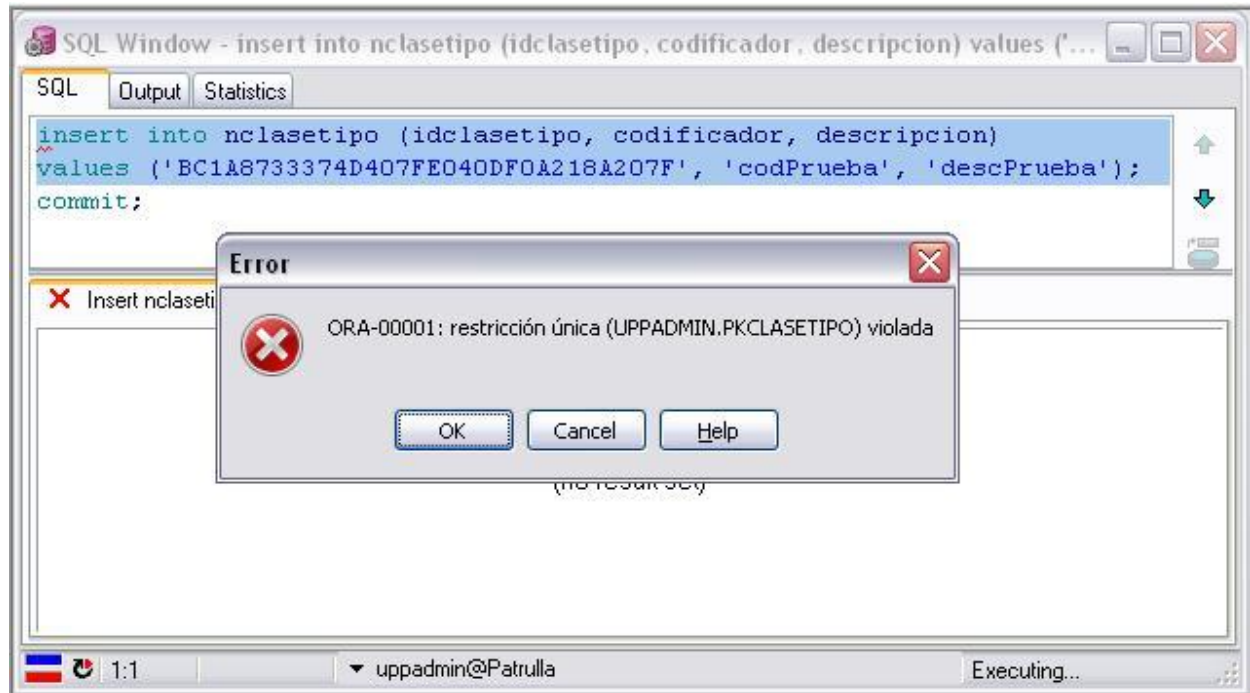


Figura 12. Error que lanza el gestor al tratar de insertar un valor de llave primaria duplicado.

## 4. Prueba de carga

El propósito principal de una prueba de carga es simular el acceso de muchos usuarios a un servidor al mismo tiempo. Se realizó esta prueba al servidor de base de datos con el objetivo de evaluar su comportamiento ante una carga de usuarios concurrentes esperada y luego ante una carga mayor que la esperada. Se efectuó en un escenario donde el servidor de base de datos cuenta con una memoria RAM de 2Gb y un procesador de dos núcleos, que son los recursos de hardware con que se cuenta en el laboratorio de desarrollo y con menores prestaciones que los reales en donde se va a desplegar la solución.

### ➤ Prueba 1

La prueba consistió en realizar tres consultas por una cantidad de 100 usuarios concurrentes para cada una, lo cual se traduce en 300 peticiones concurrentes al servidor de base de datos. A continuación se exponen los resultados proyectados por esta prueba:



# Capítulo IV. Validación de la Propuesta

Funcionalidades	Peticiones concurrentes (PC)	Tiempo de respuesta (TR) en segundos	TR/PC
Buscar emergencia por Id	100	0,72	0,0072
Buscar emergencias asociadas	100	1,35	0,0135
Buscar todas las emergencias	100	0,69	0,0069

Tabla 8. Resultados de la prueba 1.

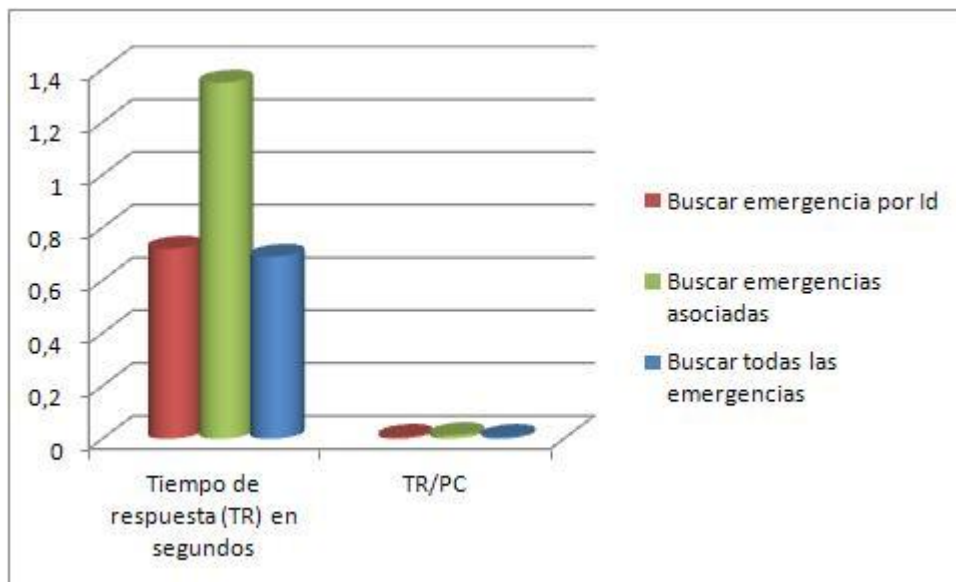


Figura 13. Gráfico que muestra los resultados de la prueba 1.

## ➤ Prueba 2

La prueba consistió en ejecutar tres consultas por una cantidad de 200 usuarios concurrentes para cada una, lo cual se traduce en 600 transacciones concurrentes con el servidor de base de datos. Los resultados arrojados por esta prueba se muestran continuación:

# Capítulo IV. Validación de la Propuesta

Consultas	Peticiones concurrentes (PC)	Tiempo de respuesta (TR) en segundos	TR/PC
Buscar emergencia por Id	200	2,05	0,0103
Buscar emergencias asociadas	200	3,93	0,0197
Buscar todas las emergencias	200	2,07	0,0104

Tabla 9. Resultados de la prueba 2.

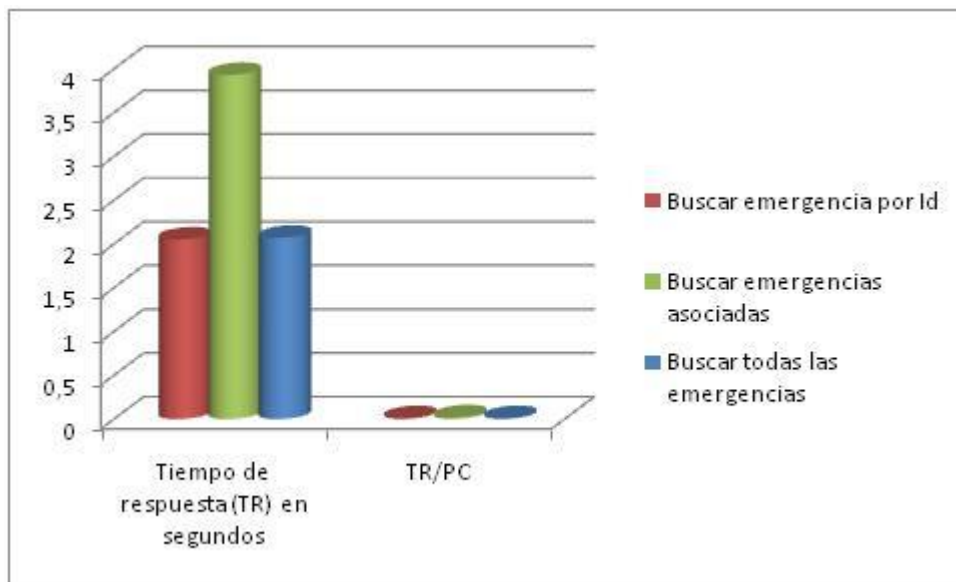


Figura 14. Gráfico que representa los resultados de la prueba 2.

Como resultado de las pruebas de carga realizadas, se puede afirmar que el servidor es capaz de responder al número esperado de usuarios concurrentes y también a una carga mucha mayor, la cual se reflejó en la segunda prueba. Además, el tiempo de respuesta a las peticiones estuvo por debajo del límite definido en los requisitos por parte de la arquitectura y del cliente.

## 5. Conclusiones

En el capítulo se reflejaron pruebas efectuadas a la base de datos, las cuales se realizaron satisfactoriamente. En cada caso se obtuvo los resultados esperados y se concluye afirmando que se

## *Capítulo IV. Validación de la Propuesta*

---

realizó un modelo de datos normalizado casi en su totalidad hasta tercera forma normal, con algunas excepciones para ganar en rendimiento, garantizando la existencia controlada de información redundante en la base de datos; se garantiza la integridad de los datos a través de restricciones que evitan errores y corrupción de los datos en las operaciones de inserción, modificación y eliminación; además, la respuesta a las peticiones se realiza en el tiempo requerido y se probó que el servidor no colapsa ante un número elevado de peticiones en situaciones de stress del sistema.

## CONCLUSIONES GENERALES

A partir de la problemática inicial, utilizando los métodos relativos al diseño de bases de datos y a la investigación científica se dio cumplimiento a los objetivos propuesto y se obtuvieron los siguientes resultados:

- ✓ Se realizó el diseño del modelo de datos del área de Atención a Llamadas, con lo cual se logra la persistencia y recuperación de los datos que se manejan en esa área en el Centro de Información y Mando de la Unidad Provincial de Patrullas.
- ✓ En el diseño del modelo de datos se aplicaron patrones de diseño de bases de datos, los cuales permitieron modelar distintas situaciones, reduciendo el tiempo de modelado y tributando a la limpieza y organización de los objetos en la base de datos.
- ✓ Se propuso el esquema de configuración y seguridad, considerando las políticas del cliente y las recomendaciones que realiza el proveedor del gestor en explotación, para asegurar la disponibilidad y seguridad de la información.
- ✓ Se implementaron procedimientos almacenados para optimizar la respuesta ante consultas que, dada su complejidad por el número de tablas a consultar o por el volumen de datos necesario para obtener los resultados deseados, pudieran responder de manera ineficiente.

## RECOMENDACIONES

Luego de concluir el presente trabajo se recomienda:

- ✓ Aplicar a los nuevos submodelos de datos patrones de diseño que resuelvan la problemática existente empleando buenas prácticas y que además brinden una solución lo más robusta posible.
- ✓ Hacer un estudio a profundidad de las medidas y herramientas de seguridad de Oracle 11g que no se tratan en la solución, para incluirlas en ésta y así garantizar el máximo nivel de seguridad posible.
- ✓ Luego de implementar la base de datos, mantener una perspectiva de perfeccionamiento continuo de la presente solución, partiendo de negocios similares en otros contextos.

## REFERENCIAS BIBLIOGRÁFICAS

1. **Navarro Ferrer, David Gerardo, et al.** *Proyecto Técnico*. La Habana : s.n., 2011.
2. **Notimex.** <http://www.publimetro.com.mx>. *www.publimetro.com.mx*. [Online] Junio 22, 2009. [Cited: Enero 6, 2011.] <http://www.publimetro.com.mx/noticias/crea-gdf-centro-de-atencion-a-emergencias-y-proteccion-ciudadana/nifv!cXOyc2b7IRCRtX8WXd1D4w/>.
3. **Comunidad Contact Center.** *Las 8 actividades que un Sistema de Atención a Emergencias debe cumplir*. [Documento] Mexico : ComunidadCC, 2011.
4. **de la Cruz, Yaritza and Cárdenas Martínez, Alberto.** *Perfeccionamiento de los Sistemas del Centro de Información y Mando de la Unidad Provincial de Patrullas. Subsistema Atención a Emergencias*. La Habana, Cuba : UCI, 2010.
5. **Granados Arozarena, Ana Belkis and Gonzales Mengana, Francis.** *Proyecto de Modernización Sistema Informático del Centro de Información y Mando de la Unidad Provincial de Patrullas: Procesos de atención de llamada e iniciación del turno de la solución Sistema de Atención a Emergencia (SAEM)*. [Documento] La Habana : Universidad de las Ciencias Informáticas, 2011.
6. **Pérez Acuña, Eleandro Emilio.** Universidad de las Ciencias Informáticas Biblioteca. <http://biblioteca.uci.cu>. [En línea] junio de 2011. [http://bibliodoc.uci.cu/TD/TD\\_04038\\_11.pdf](http://bibliodoc.uci.cu/TD/TD_04038_11.pdf).
7. **Rodríguez Fernández, Adonis and Vargas Allegue, Daniel E.** *Sistema de Gestión de Emergencias de Seguridad Ciudadana (171) Diseño y Administración de Base de Datos*. La Habana : Universidad de Ciencias Informáticas, 2007.
8. **Núñez Camallea, Noel L.** *Gestión de Base de Datos con ADO.NET*. La Habana : Editorial Científico-Técnica, 2004. ISBN 959-05-0374-8.
9. **Jorge, Sánchez.** *Principio sobre Bases de Datos Relacionales*. [Documento] California : Creative Commons, 2004.
10. **Date, C. J.** *Sistemas de Bases de Datos*. Naucalpan de Juarez (Mexico) : Prentice Hall (Cámara Nacional de la Industria Editorial Mexicana), 2001.
11. **Iglesias, Ana.** Universidad Carlos III de Madrid OpenCourseWare. <http://ocw.uc3m.es/>. [Online] 2009. [Cited: mayo 25, 2012.] <http://ocw.uc3m.es/ingenieria-informatica/diseño-y-administración-de-bases-de-datos/teoría/Tema1%28UnaMetodologíaDesarrolloBD%29.pdf>.
12. **Núñez Arroyo, Ing. Rosa Martha.** *Esquema Físico de BD*. 2011.
13. **Gil, Fidel, Javier, Albrigo y Do Rosario, Javier.** *Sistemas de Gestión de Bases de Datos*. Valencia : Universidad de Carabobo, 2005.

14. **Mato García, Rosa María.** *Diseño de Bases de Datos*. [Documento] La Habana : s.n., 2005.
15. **Oracle Inc.** *www.oracle.com. www.oracle.com*. [Online] 5 20, 2007. [Cited: 11 17, 2011.] <http://www.oracle.com/us/corporate/history/index.html>.
16. **MasHardware.** *http://www.mashardware.com*. [En línea] 2 de octubre de 2007. [Citado el: 1 de junio de 2012.] <http://www.mashardware.com/index.php?/Ultimas/Oracle-lanza-la-base-de-datos-Oracle-11g-el-desarrollo-mas-innovador-de-toda-su-historia.html>.
17. **Heurtel, Olivier.** *Oracle 11g Administración*. Barcelona : Ediciones ENI, 2009.
18. **Pino, Moira and Fehrmann, Alejandra.** *Oracle Database 11g Press Kit*. [Documento] Miami, Florida : Oracle, 2007.
19. **Microsoft.** Microsoft Developer Network. *http://msdn.microsoft.com*. [En línea] 2012. [Citado el: 1 de junio de 2012.] <http://msdn.microsoft.com/es-es/library/ms190787.aspx>.
20. **Baer, Hermann.** *www.oracle.com*. [Online] junio 2007. [Cited: febrero 10, 2012.] <http://www.oracle.com/technetwork/es/documentation/317489-esa.pdf>.
21. **Espinosa, Roberto.** DATAPRIX. *http://www.dataprix.com*. [Online] 04 24, 2010. [Cited: 01 10, 2012.] <http://www.dataprix.com/blogs/respinosamilla/particionado-tablas-oracle>.
22. **Aponcio Borges, Rasiel.** *Migración de la Capa de Acceso a Datos del Sistema de Gestión de Emergencias de Seguridad Ciudadana (171)*. Universidad de las Ciencias Informáticas. La Habana : s.n., 2009.
23. **Rozic, Sergio Ezequiel.** *Bases de Datos*. Buenos Aires, Argentina : MP Ediciones, 2004.
24. **Educacion Informatica.** *http://www.educacioninformatica.com. educacioninformatica*. [Online] diciembre 20, 2009. [Cited: junio 4, 2012.] <http://www.educacioninformatica.com/modules/lexikon/entry.php?entryID=86>.
25. **Álvarez, Sara.** *www.desarrolloweb.com*. [Online] Febrero 10, 2009. [Cited: 12 9, 2011.] <http://www.desarrolloweb.com/articulos/introduccion-al-lenguaje-plsql.html>.
26. **Embarcadero technologies Inc.** *ER/Studio*. [Documento] Oakland : Embarcadero technologies Inc., 2006.
27. **allroundautomation.** *PL/SQL 7.0 User's Guide*. [Digital] 2006.
28. **MasterMagazine.** *www.MasterMagazine.info*. [Online] agosto 13, 2006. [Cited: enero 16, 2012.] <http://www.MasterMagazine.info/area/pc/10487.php>.
29. **Novell.** *es.opensuse.org*. [Online] 2011. [Cited: ] [http://es.opensuse.org/Bienvenidos\\_a\\_openSUSE.org.htm](http://es.opensuse.org/Bienvenidos_a_openSUSE.org.htm).

# Referencias Bibliográficas

---

30. **Blahe, Michael.** *PATTERNS OF DATA MODELING*. New York : Taylor and Francis Group, LLC, 2010. ISBN 978-1-4398-1989-0.
31. **Microsoft.** Microsoft Developer Network. <http://msdn.microsoft.com>. [Online] mayo 31, 2007. [Cited: junio 05, 2012.] <http://msdn.microsoft.com/en-us/library/aa373931%28VS.85%29.aspx>.
32. **Bueno Cuadra, Alberto.** ABCSoft. <http://www.abcsoftperu.com/>. [Online] 2010. [Cited: enero 25, 2012.] <http://www.abcsoftperu.com/descargas/curso02.html#formas>.
33. **Oracle Inc.** *Oracle Database Documentation Library 11g*. [HTML] California : Oracle, 2007.
34. **Watson, John.** *OCA Oracle Database 11g Administration I Exam Guide*. United States of America : The Macgraw-Hill Companies, 2008.
35. **Oracle.** *Oracle Database Security Guide*. [Documento] 2007.
36. **Horrutiner Calzado, Iriobe and Ruiz Vielza, Yamel.** *Propuesta de estrategia de seguridad para la protección de las bases de datos Oracle del Centro de Datos Ministerial*. 2010.
37. **Sánchez Arenas, Osvaldo José.** *Propuesta de un esquema de seguridad para las Bases de*. 2008.
38. —. *Propuesta de un esquema de seguridad para las Bases de Datos Oracle del MININT*. La Habana : s.n., 2008.
39. **Micrisoft.** Windows. <http://windows.microsoft.com>. [Online] 2012. [Cited: abril 12, 2012.] <http://windows.microsoft.com/es-ES/windows-vista/What-are-permissions>.
40. **Castellanos, Inma.** *acens*. [www.acens.com](http://www.acens.com). [Online] Deusto, julio 3, 2007. [Cited: abril 12, 2012.] <http://www.acens.com/articulos/bases-de-datos-seguras.html>.
41. **Yacuzzi, Enrique A.** *EL ESTUDIO DE CASO COMO METODOLOGÍA DE INVESTIGACIÓN: TEORÍA, MECANISMOS CAUSALES, VALIDACIÓN*. Buenos Aires : s.n., 2005.
42. **Oracle Corporation.** *Oracle SQL Developer Data Modeler 2.0.0*. [Documento] California : Oracle Corporation, 2009.
43. **cusiglas.** <http://www.cusiglas.com>. *cusiglas.com*. [Online] [Cited: 3 2, 2012.] <http://www.cusiglas.com/significadode/dhcp.php>.
44. **Oracle Inc.** Oracle . <http://www.oracle.com>. [Online] [Cited: junio 4, 2012.] <http://www.oracle.com/lad/products/database/options/real-application-testing/index.html>.