

UNIVERSIDAD “TÉCNICA DEL NORTE”



FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
ESCUELA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

TESIS PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERA EN
SISTEMAS COMPUTACIONALES.

TEMA:

“FRAGMENTACIÓN DE TABLAS EN BASES DE DATOS
DISTRIBUIDAS”

APLICATIVO:

“IMPLEMENTACIÓN DE UNA BASE DE DATOS DISTRIBUIDA EN
EL SISTEMA ACADÉMICO DE LA FICA”

AUTORA:

LÓPEZ TULCANAZA VERÓNICA ALEXANDRA

DIRECTOR:

Ing. MIGUEL ORQUERA.

Ibarra – Ecuador
2010



CERTIFICACIÓN

Certifico que la Tesis **“FRAGMENTACIÓN DE TABLAS EN BASES DE DATOS DISTRIBUIDAS”** ha sido realizada en su totalidad por la señorita:

Verónica Alexandra López Tulcanaza.

Ing. Miguel Orquera.



DEDICATORIA

A mis Padres.

Eduardo y Fanny que con su infinito amor supieron guiarme en el camino del estudio, para alcanzar una profesión y ser mujer de bien y útil a la sociedad.

A mi hija Hilary por su tierno amor, a ellos dedico este trabajo fruto de su sacrificio y esfuerzo constante.

Verónica A. López.



AGRADECIMIENTO

Mi eterna gratitud para quienes me apoyaron en todo momento, de manera especial a mis Padres y a mi hija, debido a que fueron mi apoyo incondicional.

A mi esposo Saulo por su comprensión y abnegado sacrificio.

A mi Director de Tesis Ing. Miguel Orquera del cual llevo las mejores enseñanzas.

Verónica A. López.



ÍNDICE

CERTIFICACIÓN.....	2
DEDICATORIA.....	3
AGRADECIMIENTO.....	4
RESUMEN.....	9
INTRODUCCION.....	10
CAPITULO I: BASE DE DATOS DISTRIBUIDA.....	12
1.1. INTRODUCCIÓN.....	12
1.1.1 ALMACENAMIENTO DISTRIBUIDO.....	13
1.2. SISTEMA DE GESTIÓN DE BASE DE DATOS DISTRIBUIDAS.....	14
1.2.1. FUNCIONES Y ARQUITECTURA DE UN SGBDD.....	15
1.3. DOCE REGLAS DE LAS BASES DE DATOS DISTRIBUIDAS.....	17
1.4. VENTAJAS E INCONVENIENTES DE LAS BASES DE DATOS DISTRIBUIDAS.....	19
1.5. BASES DE DATOS DISTRIBUIDAS VRS. CENTRALIZADA.....	21
1.5.1. BASE DE DATOS CENTRALIZADA (BDC).....	21
1.5.2. VENTAJAS E INCONVENIENTES.....	22
CAPITULO II: FRAGMENTACIÓN DE TABLAS.....	26
2.1. INTRODUCCIÓN.....	26
2.1.1. DEFINICIÓN.....	26
2.2. TIPOS DE FRAGMENTACIÓN.....	29
2.2.1. FRAGMENTACIÓN HORIZONTAL.....	30
2.2.2. FRAGMENTACIÓN VERTICAL.....	33
2.2.3. FRAGMENTACIÓN MIXTA.....	36
2.2.4. GRADOS DE FRAGMENTACIÓN.....	37
CAPITULO III: PARTICIONAMIENTO EN ORACLE.....	46
3.1. INTRODUCCIÓN.....	46
3.2. ESTRUCTURA LÓGICA Y FÍSICA DE LA BASE DE DATOS ORACLE.....	48
3.3. ESTRATEGIAS BÁSICAS DE PARTICIONAMIENTO.....	54
3.3.1. PARTICIONAMIENTO POR RANGO.....	54
3.3.2. PARTICIONAMIENTO POR LISTA.....	57
3.3.3. PARTICIONAMIENTO POR HASH.....	59
3.3.4. PARTICIÓN POR COMPOSICIÓN.....	62
3.3.5. EXTENSIONES DE PARTICIONAMIENTO.....	66
3.4. BASES DE DATOS DISTRIBUIDAS EN ORACLE.....	73
3.5. CREACIÓN Y USO DE DBLINKS EN ORACLE.....	74
3.6. VISTAS EN ORACLE.....	77
3.7. REPLICACIÓN DE BASE DE DATOS EN ORACLE.....	83
CAPITULO IV: ARQUITECTURA DE DISTRIBUCIÓN DE DATOS.....	88
4.1. TRANSPARENCIA.....	96
4.1.1. TRANSPARENCIA DE LOCALIZACIÓN.....	98
4.1.2. TRANSPARENCIA DE FRAGMENTACIÓN.....	99
4.1.3. TRANSPARENCIA DE REPLICA.....	100
4.2. FORMAS DE RÉPLICAS.....	102
CAPITULO V: ADMINISTRADOR DE LA BASE DE DATOS.....	107
5.1. CONCEPTOS GENERALES.....	107
5.1.1. ADMINISTRAR LA ESTRUCTURA DE LA BASE DE DATOS.....	109
5.1.2. ADMINISTRACIÓN DE LA ACTIVIDAD DE DATOS.....	110
5.1.3. ADMINISTRAR EL SISTEMA MANEJADOR DE BASE DE DATOS.....	110
5.1.4. CONFIRMAR LA SEGURIDAD DE LA BASE DE DATOS.....	112
5.2. OBJETIVOS DEL ADMINISTRADOR DE LA BASE DE DATOS.....	113
5.2.1. INTEGRIDAD DE LOS DATOS.....	114
5.2.2. SEGURIDAD DE DATOS.....	114
5.2.3. DISPONIBILIDAD DE DATOS.....	114



5.3.	FUNCIONES BÁSICAS.....	115
5.3.1.	CREACIÓN DE USUARIOS	115
5.3.2.	CREACIÓN DE BASES DE DATOS Y TABLAS.....	117
5.3.3.	RESTRICCIONES DE INTEGRIDAD.....	117
5.3.4.	PROCEDIMIENTOS DE RESPALDO Y RECUPERACIÓN.....	118
	CAPITULO VI: DESARROLLO DE LA APLICACIÓN	122
6.1.	INTRODUCCIÓN.....	122
6.2.	GESTIÓN DEL PROYECTO	123
6.3.	MODELADO DEL NEGOCIO.....	143
6.4.	REQUISITOS	147
6.5.	ANÁLISIS DE DISEÑO.....	165
6.6.	IMPLEMENTACIÓN	168
6.7.	PRUEBAS.....	171
6.8.	DOCUMENTACIÓN.....	175
	CAPITULO VII: CONCLUSIONES Y RECOMENDACIONES	186
7.1.	Verificación de Hipótesis	186
7.1.1.	Hipótesis	186
7.1.2.	Verificación.....	186
7.2.	CONCLUSIONES	188
7.3.	RECOMENDACIONES	190
7.4.	POSIBLES TEMAS DE TESIS	192
7.5.	BIBLIOGRAFÍA	193



ÍNDICE DE FIGURAS

1.	Fragmentación Horizontal [IMAG.01]	30
2.	Fragmentación Vertical [IMAG.02]	33
3.	Fragmentación Mixta [IMAG.03]	36
4.	Particionamiento por Rango [IMAG.04]	54
5.	Particionamiento por Lista [IMAG.05]	57
6.	Particionamiento por Hash [IMAG.06]	59
7.	Particionamiento por Composición [IMAG.07]	62
8.	Ejemplo de Dblink Y Vistas en Oracle	78
9.	Ejemplo de Replica en Oracle	84
10.	Arquitectura ANSI/X3/SPARC para BDD [IMAG.10]	89
11.	Arquitectura de un SMBDD homogéneo	91
12.	Arquitectura de los esquemas de un SMBDD homogéneo	92
13.	Arquitectura de un sistema multi-bases de datos	93
14.	Arquitectura basada en componentes de un SMBD distribuido	94
15.	Arquitectura basada en componentes sistema multi-bases de datos	95
16.	Organización en Capas de niveles de Transparencia. [IMAG.16]	98
17.	Muestra la Transparencia de Fragmentación [IMAG.17]	99
18.	Principal utilidad de la Transparencia de Replicación [IMAG.18]	101
19.	Replica Periódica [IMAG.19]	102
20.	Replica Continua [IMAG.20]	103
21.	Replica Check – IN/OUT [IMAG.21]	103
22.	División del DBMS [IMAG.22]	108
23.	Relación de Usuarios [IMAG.23]	116
24.	Interacciones del Proyecto	139
25.	Diagrama de caso de Uso del Administrador. Fuente Propia	144
26.	Diagrama de caso de Uso del Usuario. Fuente Propia	145
27.	Diagrama de caso de Uso del Estudiante. Fuente Propia	145
28.	Diagrama de caso de Uso del Docente. Fuente Propia	145
29.	Diagrama de caso de Uso de la Secretaria de la Escuela. Fuente Propia	146
30.	Diagrama de Clase del Sistema Académico	166
31.	Diagrama Relacional del Sistema Académico	167
32.	Diagrama de Componentes del Proyecto	170
33.	Diagrama de Despliegue del Proyecto	171
34.	Arquitectura MVC separa limpiamente la interfaz de usuario, lógica de negocio y Navegación de página [IMAG.34]	177
35.	División de Capa de Modelo en un Servicio de Negocio [IMAG.35]	178
36.	Arquitectura ADF [IMAG.36]	180
37.	ADF Business Components [IMAG.37]	180
38.	Componentes del dominio del negocio [IMAG.38]	183



ÍNDICE DE TABLAS

Comparación de Base de Datos Distribuida vr Centralizada.....	24
Privilegios de Oracle.....	53
Roles y Responsabilidades	136
Plan de Fase del Proyecto	137
Hitos de las Fases del Proyecto	138
Calendario del proyecto. Fase de Inicio.....	140
Calendario del Proyecto. Fase de Elaboración.....	141
Resumen de los Stakeholders del Proyecto.....	149
Resumen de los Usuarios del Proyecto	150
Representante del Área Técnica del Proyecto	150
Usuario Administrador del Proyecto	151
Usuario Secretaria del Proyecto	151
Usuario Docente del Proyecto	151
Resumen de las características del Proyecto.....	152
Matriz de atributos de los Stakeholders del Proyecto.....	162
Matriz de atributos de los Actores del Proyecto.....	162
Matriz de atributos de las características del Proyecto	164
Matriz de atributos de los Casos de Uso del Proyecto	164
Matriz de atributos de las Clases del Proyecto	165



RESUMEN

Este tema trata de simplificar las consultas, con el fin de que un usuario pueda obtener acceso a los datos, tal como si todos los datos estuvieran almacenados en el sitio propio del usuario, ya que cada sitio es un sistema de base de datos en sí, cuyos datos se almacenan físicamente en varias bases de datos reales pero ubicadas en diferentes sitios, donde son utilizados con mayor frecuencia de tal manera que la mayor parte de las operaciones sean sólo locales lo cual reducirá el tráfico en la red

SUMMARY

This topic tries to simplify the consultations, in order which a user could obtain access to the information, such as if all the information was stored in the own site of the user, since every site is a system of database in yes, whose information is stored physically in several databases royal but located in different sites, where they are used by major frequency in such a way that most of the operations is only local which will reduce the traffic in the network



INTRODUCCION

Las Bases de Datos en la actualidad son muy importantes debido a que en ellas se guarda gran información de suma importancia.

Se debe construir sistemas distribuidos de bases de datos, ya que se están utilizando cada vez más al igual que se usan las arquitecturas de cliente-servidor.

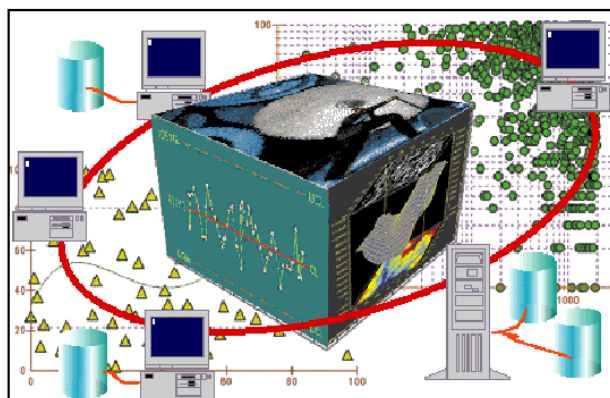
Cada sitio es un sistema de base de datos en sí mismo, con el fin de que un usuario de cualquier sitio de la red pueda obtener acceso a los datos, tal como si todos los datos estuvieran almacenados en el sitio propio del usuario, es en realidad una especie de objeto virtual, cuyos datos se almacenan físicamente en varias bases de datos reales pero ubicadas en diferentes sitios, el mayor reto es identificar la fragmentación ideal para los datos en sí.

La fragmentación es viable debido a que los datos pueden almacenarse en localidades donde son utilizados con mayor frecuencia, como por ejemplo: cada carrera tendrá en su servidor los datos solo de ésta, tanto de estudiantes como profesores, materias y las que sean necesarias, de tal manera que la mayor parte de las operaciones sean sólo locales lo cual reducirá el tráfico en la red



1

BASE DE DATOS DISTRIBUIDOS



Contenido:

Base de Datos Distribuidos

- 1.1. Introducción
 - 1.1.1 Almacenamiento Distribuido
- 1.2. Sistemas de gestión de bases de datos distribuidos
 - 1.2.1 Funciones y Arquitectura de un SGBDD
- 1.3. Las Doce Reglas
- 1.4. Ventajas e Inconvenientes
- 1.5. Base de Datos Distribuidas Vs Centralizada
 - 1.5.1 Base de Datos Centralizada.
 - 1.5.2 Ventajas e Inconvenientes.



CAPÍTULO I

1. BASE DE DATOS DISTRIBUIDA

1.1. INTRODUCCIÓN

Las Bases de Datos en la actualidad son muy importantes debido a que en ellas se guarda gran información de suma importancia.

Este tema trata de simplificar las consultas de una base de datos, para poder acceder desde cualquier sitio, en cualquier punto de la red tal como si todos los datos estuvieran almacenados en el sitio mismo.

Para esto se debe construir sistemas distribuidos de bases de datos, ya que se están utilizando cada vez más al igual que se usan las arquitecturas de cliente-servidor. Los principales problemas que se generan por el uso de la tecnología de bases de datos distribuidas son en lo referente a duplicidad de datos y su integridad al momento de realizar actualizaciones a los mismos.



1.1.1 ALMACENAMIENTO DISTRIBUIDO

BASES DE DATOS DISTRIBUIDAS.

Es una colección de datos que pertenecen lógicamente a un sólo sistema, pero se encuentra físicamente esparcido en varios "sitios" de la red. Un sistema de base de datos distribuidos se compone de un conjunto de sitios, conectados entre sí mediante algún tipo de red de comunicaciones, en el cual:

- Cada sitio es un sistema de base de datos en sí mismo.
- Los sitios trabajan en conjunto si es necesario con el fin de que un usuario de cualquier sitio pueda obtener acceso a los datos de cualquier punto de la red tal como si todos los datos estuvieran almacenados en el sitio propio del usuario.

OBJETIVOS DE LAS BASES DE DATOS DISTRIBUIDAS

Es una especie de objeto virtual, cuyos componentes se almacenan físicamente en varias bases de datos “reales” distintas, ubicadas en diferentes sitios.

Cada sitio tiene sus propias bases de datos “reales” locales, sus propios usuarios locales, sus propios DBMS y programas para la administración de transacciones y su propio administrador local de comunicación de datos.



Un usuario puede realizar operaciones sobre los datos en su propio sitio local exactamente como si ese sitio no participara en absoluto en el sistema distribuido.

1.2. SISTEMA DE GESTIÓN DE BASE DE DATOS DISTRIBUIDAS (SGBDD)

Es necesaria una aplicación que actúe de interfaz entre el usuario, los modelos y el sistema físico. Esta es la función que desempeñan los SGBD. Una base de datos permite al usuario acceder a datos almacenados sin que él deba saber cómo están organizados esos datos en el sistema, garantiza que ningún usuario sin autorización pueda ver o manipular los datos.

Las principales funciones que debe cumplir un SGBD se relacionan con la creación y mantenimiento de la base de datos, el control de accesos, la manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, evitar redundancias e inconsistencias y mantener la integridad.



1.2.1. FUNCIONES Y ARQUITECTURA DE UN SGBDD

OBJETIVOS DE LOS SGBDD

- **Permitir la definición de todos los datos.** Es decir, incorporar a las estructuras todos los objetos necesarios para completarlas e incluir los atributos necesarios para definir a los objetos.
- **Permitir la manipulación de los datos:** Operaciones de intercambio de datos entre las tablas que pueden ser de consulta o de puesta al día (inserción, modificación, supresión)
- **Establecer controles de seguridad:** Garantizando que sólo los usuarios autorizados puedan efectuar operaciones correctas bien sobre toda la base de datos o sobre algunas tablas.
- **Permitir los accesos concurrentes:** El principal objetivo de la implantación de una base de datos es poner a disposición de un gran número de usuarios un conjunto integrado de datos y que estos datos puedan ser manipulados por los diferentes usuarios. El SGBD debe garantizar que esos datos seguirán siendo coherentes después de las diversas manipulaciones.

Una mala concepción a la hora de diseñar una base de datos puede dar origen a:

- ✓ La redundancia de datos
- ✓ Incoherencia en los datos
- ✓ Pérdida de datos



- ✓ Estado de la tabla

COMPONENTES DE UN SISTEMA GESTOR DISTRIBUIDO

Hardware

- Los nodos de procesamiento pueden estar formados por múltiples y distintos tipos de hardware.

Software

- El programa principal que necesitamos considerar en un sistema de base de datos distribuida es el DDBMS.
- En un SGBD distribuido, la base de datos se almacena en varios computadores que se pueden comunicar a su vez por distintos medios de comunicación. Se denomina con el término de emplazamientos o nodos a todos aquellos computadores que pertenecen a un sistema distribuido.

Datos

- Una base de datos distribuida puede no estar duplicada, parcialmente duplicada o totalmente duplicada. Una base de datos parcialmente duplicada contiene ciertos elementos de datos que están duplicados y algunos que no lo están.

Procedimientos

- Los sistemas de base de datos distribuidos contienen una multitud de componentes de procesamiento. En los sistemas distribuidos más sencillos, los datos no están duplicados, y sólo el nodo que almacena los datos puede actualizarlos.



Personal

- Los sistemas con un DDBMS complejo y poderoso imponen pocas demandas especiales sobre los usuarios. De hecho, los usuarios no saben que están procesando datos distribuidos.

1.3. DOCE REGLAS DE LAS BASES DE DATOS DISTRIBUIDAS

Principio fundamental: Para el usuario un sistema distribuido debe ser igual que uno centralizado.

1. **Autonomía Local:** Los sitios distribuidos deben ser autónomos, es decir que todas las operaciones en un sitio dado se controlan en ese sitio.
2. **No dependencia de un sitio central:** No debe de haber dependencia de un sitio central para obtener un servicio. La dependencia de un sitio sería indeseable por las siguientes razones: ese sitio podría ser un cuello de botella. El sistema sería vulnerable; si el sitio sufriera un desperfecto, todo el sistema dejaría de funcionar.
3. **Operación Continua:** Nunca debería haber necesidad de apagarse a propósito para que se pueda realizar alguna función, como añadir un nuevo sitio, o instalar una versión mejorada
4. **Independencia con respecto a la localización:** No debe de ser necesario que los usuarios sepan dónde están almacenados



físicamente los datos, el usuario debe de ver como si solo existiera un sitio local.

- 5. Independencia con respecto a la fragmentación:** La fragmentación es deseable por razones de desempeño, los datos, pueden almacenarse en la localidad donde se utilizan con mayor frecuencia de manera que la mayor parte de las operaciones sean sólo locales y se reduzca el tráfico en la red.
- 6. Independencia de réplica:** Si una relación dada, es decir, un fragmento dado de una relación se puede presentar en el nivel físico mediante varias copias almacenadas o réplicas, en muchos sitios distintos. La réplica es viable por dos razones: las aplicaciones pueden operar sobre copias locales en vez de tener que comunicarse con sitios remotos; una mejor disponibilidad. La desventaja principal de las réplicas es, cuando se pone al día un cierto objeto copiado, deben ponerse al día todas las réplicas de ese objeto.
- 7. Procesamiento Distribuido de Consultas:** El objetivo es convertir transacciones de usuario en instrucciones para manipulación de datos, y así reducir el tráfico en la red, esto implica que el proceso mismo de optimización de consultas debe ser distribuido.
- 8. Manejo Distribuido de Transacciones:** Tiene dos aspectos principales, el control de recuperación y el control de



conurrencia, cada uno de los cuales requiere un tratamiento más amplio en el ambiente distribuido.

9. Independencia con respecto al equipo: El SGBDD (Sistema de Gestión de Base de Datos Distribuidos) debe ser ejecutable en diferentes plataformas hardware.

10. Independencia con respecto al Sistema Operativo: El sistema debe ser ejecutable en diferentes SO.

11. Independencia con respecto a la red: El sistema debe poder ejecutarse en diferentes redes.

12. Independencia con respecto al DBMS: No se requiere sino que los DBMS en los diferentes sitios manejen todos la misma interfaz; no necesitan ser por fuerza copias del mismo sistema.

1.4. VENTAJAS E INCONVENIENTES DE LAS BASES DE DATOS DISTRIBUIDAS

VENTAJAS DE LAS BASES DE DATOS DISTRIBUIDAS

- **Compartimiento de datos.** Los usuarios de un nodo son capaces de acceder a los datos de otro nodo.
- **Autonomía.** Cada nodo tiene cierto grado de control sobre sus datos, en un sistema centralizado, hay un administrador del sistema responsable de los datos a nivel global. Cada administrador local puede tener un nivel de autonomía local diferente.



- **Disponibilidad.** Si en un sistema distribuido falla un nodo, los nodos restantes pueden seguir funcionando. Si se duplican los datos en varios nodos, la transacción que necesite un determinado dato puede encontrarlo en cualquiera de los diferentes nodos.

INCONVENIENTES DE LAS BASES DE DATOS DISTRIBUIDAS

- **Coste de desarrollo del software.** La complejidad añadida que es necesaria para mantener la coordinación entre nodos hace que el desarrollo de software sea más costoso.
- **Mayor probabilidad de errores.** Como los nodos que constituyen el sistema funcionan en paralelo, es más difícil asegurar el funcionamiento correcto de los algoritmos, así como de los procedimientos de recuperación de fallos del sistema.
- **Mayor sobrecarga de procesamiento.** El intercambio de mensajes y ejecución de algoritmos para el mantenimiento de la coordinación entre nodos supone una sobrecarga que no se da en los sistemas centralizados.



1.5. BASES DE DATOS DISTRIBUIDAS VRS. CENTRALIZADA

Los sistemas de bases de datos centralizados, residen en una sola localidad.

Los sistemas de bases de datos distribuidos se encuentran en varias localidades.

1.5.1. BASE DE DATOS CENTRALIZADA (BDC).

Es una base de datos almacenada en su totalidad en un solo lugar físico, es decir, es una base de datos almacenada en una sola máquina y una sola CPU, en donde los usuarios trabajan en terminales que sólo muestran resultados.

Son aquellos que se ejecutan en un único sistema informático sin interaccionar con ninguna otra computadora.

CARACTERÍSTICAS DE LAS BASES DE DATOS CENTRALIZADAS.

- Se almacena completamente en una localidad central, es decir, todos los componentes del sistema residen en un solo computador o sitio.



- No posee múltiples elementos de procesamiento ni mecanismos de intercomunicación como las Bases de Datos Distribuidas.
- Los componentes de la Bases de Datos Centralizadas son los datos, el software del SGBD (Sistema de Gestión de Base de Datos) y los dispositivos de almacenamiento secundario asociados, como disco de almacenamiento en línea de la base de datos y cintas para las copias de seguridad.
- El problema de seguridad es inherentemente fácil en estos sistemas de bases de datos.

1.5.2. VENTAJAS E INCONVENIENTES

VENTAJAS DE LAS BASES DE DATOS CENTRALIZADAS.

- Se evita la redundancia. En sistemas que no usan Bases de Datos Centralizadas, cada aplicación tiene sus propios archivos privados o se encuentran en diferentes localidades. Esto a menudo origina enorme redundancia en los datos almacenados, así como desperdicio resultante del espacio de almacenamiento.
- Se evita la inconsistencia. Ya que si un hecho específico se representa por una sola entrada, la no-concordancia de datos no puede ocurrir.



- Pueden hacerse cumplir las normas establecidas. Con un control central de la base de datos, el Administrador de Base de Datos (DBA) puede garantizar que se cumplan todas las formas aplicables a la representación de los datos.
- Pueden aplicarse restricciones de seguridad. El Administrador de Base de Datos (DBA) puede:
 - ✓ Asegurar que el único medio de acceder la base de datos sea a través de los canales establecidos.
 - ✓ Definir controles de autorización para que se apliquen cada vez que se intente el acceso a datos sensibles
- Puede conservarse la integridad. Que los datos se encuentren en una sola máquina ayuda a evitar la inconsistencia de los datos, por el mismo hecho de encontrarse en una sola máquina.
- El procesamiento de los datos ofrece un mejor rendimiento y resulta más confiable que en los sistemas distribuidos.

DESVENTAJAS DE LAS BASES DE DATOS CENTRALIZADAS.

- Los mainframes (computadora central) no ofrecen mejor proporción precio/rendimiento que los microprocesadores de los sistemas distribuidos.



- Cuando un sistema de Base de Datos Centralizada falla, se pierde toda la disponibilidad de procesamiento y sobre todo de la información confiada al sistema.
- En caso de un desastre o catástrofe, la recuperación es difícil de sincronizar.
- Las cargas de trabajo no se pueden difundir entre diferentes computadoras, ya que los trabajos siempre se ejecutarán en la misma máquina.
- No se puede añadir poder de cómputo en pequeños incrementos, debido a lo complicado de esta operación.

COMPARACIÓN

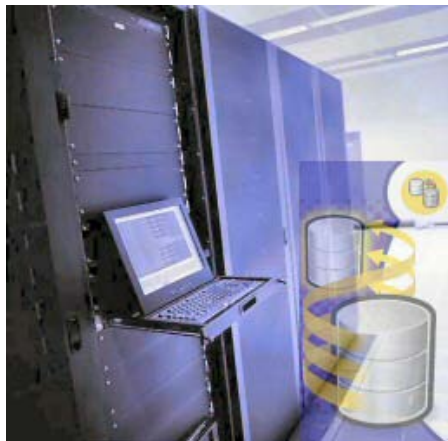
CENTRALIZADO	DISTRIBUIDO
Control centralizado: Un solo DBA	Control jerárquico: DBA global y DBAs locales
Independencia de Datos: Organización de los datos es transparente para el programador	Transparencia en la Distribución: Localización de los datos es un aspecto adicional de independencia de datos
Reducción de redundancia: Una sola copia de datos que se comparta	Replicación de Datos: Copias múltiples de datos que incrementa la localidad y la disponibilidad de datos
Estructuras físicas complejas para accesos eficientes	No hay estructuras intersitios. Uso de optimización global para reducir transferencia de datos
Seguridad	Problemas de seguridad Intrínsecos

Comparación de Base de Datos Distribuida vr Centralizada



2

FRAGMENTACIÓN DE TABLAS



Contenido:

Fragmentación de tablas

- 2.1. Introducción
 - 2.1.1. Definición.
- 2.2. Tipos de Fragmentación
 - 2.2.1. Fragmentación Horizontal
 - 2.2.2. Fragmentación Vertical
 - 2.2.3. Fragmentación Mixta
 - 2.2.4. Grados de Fragmentación



CAPITULO II

2. FRAGMENTACIÓN DE TABLAS

2.1. INTRODUCCIÓN

Es preciso tomar decisiones respecto a los sitios en los que se almacenarán las porciones de la base de datos, debemos determinar las unidades lógicas de la base de datos que van a ser distribuidas, las unidades lógicas se llamarán: fragmentos.

El mayor reto es identificar la fragmentación ideal para los datos, además de resolver los grados de fragmentación.

Los datos pueden almacenarse en localidades donde son utilizados con mayor frecuencia, de tal manera que la mayor parte de las operaciones sean sólo locales lo cual reducirá el tráfico en la red.

2.1.1. DEFINICIÓN

FRAGMENTACIÓN

Es una relación que corresponde a una tabla, consiste en dividirla en fragmentos menores, cada fragmento se guarda en sitio diferente, tiene como objetivo buscar alternativas para dividir una las tablas o instancias en otras más pequeñas. Para ello hay las siguientes alternativas lógicas, como son: La fragmentación se



puede realizar por tuplas individuales (fragmentación horizontal), por atributos individuales (fragmentación vertical) o una combinación de ambas (fragmentación híbrida).

Fragmentos: Cada relación global puede ser dividida en porciones llamados fragmentos. El mapa resultante se denomina esquema de fragmentación. Una relación global puede dividirse en n fragmentos y un fragmento sólo puede pertenecer a una relación global. Los fragmentos se referencian por un nombre de relación global y un subíndice

RAZONES PARA FRAGMENTAR

- Encontrar unidad de distribución más adecuada.
- Disminuir cantidad de accesos remotos.
- Incrementar el nivel de concurrencia.

VENTAJAS DE LA FRAGMENTACIÓN DE TABLAS

- Mejorar el rendimiento de las aplicaciones al trabajar con subconjuntos de relaciones
- Da una respuesta eficiente a aplicaciones que trabajan con los mismos datos en diferentes nodos
- Los fragmentos permiten aumentar el número de ejecuciones concurrentes.



DESVENTAJAS DE LA FRAGMENTACIÓN DE TABLAS

- Disminuye la eficiencia en las aplicaciones que trabajan con varios fragmentos
- La comprobación de las restricciones de integridad puede ser más costosa.

CONDICIONES PARA DEFINIR FRAGMENTOS

Se deben cumplir tres reglas durante el proceso de fragmentación. Los cuales se explicara más detalladamente en los grados de fragmentación.

- ✓ Completitud
- ✓ Reconstrucción
- ✓ Disyunción

Completitud: La descomposición de una relación R en los fragmentos R_1, R_2, \dots, R_n es completa si y solamente si cada elemento de datos en R se encuentra en algún fragmento. [www.03]¹

Reconstrucción: Si la relación R se descompone en los fragmentos R_1, R_2, \dots, R_n , entonces debe existir algún operador que permita reconstruir la Relación Original R . [www.03]¹

¹ http://www.google.com.ec/url?sa=t&source=web&cd=3&ved=0CBwQFjAC&url=http%3A%2F%2Fwww.pascualbravo.edu.co%2Fbuzon%2FTRABAJO%2520DE%2520BASES%2520DE%2520DATOS.ppt&rct=j&q=Completitud+%2B+reglas+de+fragmentacion+de+tablas&ei=d-4jTK_KF4KB8gbwv62_BQ&usg=AFQjCNF-ZylYOaCw_MezqcebMaKRYG3QWQ



Disyunción: Si la relación R se descompone en los fragmentos R_1 , R_2 , ..., R_n , y el dato di está en R_j , entonces, no debe estar en ningún otro fragmento. [www.03]⁴

Repetición y Fragmentación

Las técnicas de réplica y fragmentación se pueden aplicar sucesivamente a la misma relación de partida. Un fragmento se puede replicar y a su vez esa réplica ser fragmentada, para luego replicar alguno de esos fragmentos.

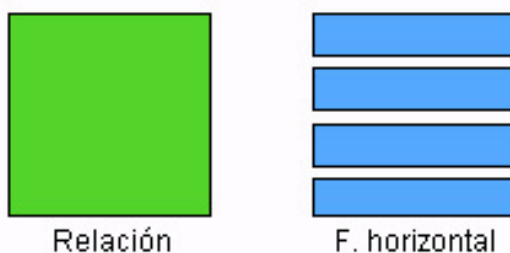
2.2. TIPOS DE FRAGMENTACIÓN.

La forma en cómo se pueden extraer los datos al ser consultados en un ambiente distribuido, se puede hacer una fragmentación de distintas tablas pertenecientes a diversas Bases de Datos localizadas en diversos servidores. Dado que una relación que corresponde esencialmente con una tabla esta se la puede dividir en fragmentos menores, inmediatamente surgen alternativas lógicas para llevar a cabo el proceso:

Existen tres tipos de fragmentación:

1. Fragmentación horizontal
2. Fragmentación vertical
3. Fragmentación híbrida

2.2.1. FRAGMENTACIÓN HORIZONTAL.



1. Fragmentación Horizontal [IMAG.01]²

Se realiza sobre las tuplas de la relación, es decir que cada fragmento será un subconjunto de las tuplas de la relación.

Una tabla T se divide en subconjuntos, T_1, T_2, \dots, T_n . Los fragmentos se definen mediante una operación de selección. Su reconstrucción se realizará mediante la unión de los fragmentos componentes. [www.04]³

Existen dos tipos de fragmentación como es: fragmentación horizontal primaria y la fragmentación horizontal derivada.

La fragmentación horizontal primaria. Consiste en particionar las tuplas (columnas) de una relación global en subconjuntos, donde cada subconjunto tenga propiedades comunes, se puede definir expresando cada fragmento como una operación de selección sobre la relación global.

² http://cablemodem.fibertel.com.ar/bucanero/facultad/Lab4_TP6.doc

³ <http://www.oei.eui.upm.es/Asignaturas/BD/DYOB/DISTRIBUIDAS.pdf>



Ejemplo: Considere la Relación J

Jno	NOMBRE	NOTA	ESCUELA
J1	LUIS YANEZ	8	CIME
J2	ERIKA QUIROZ	8	CIME
J3	DANIEL MURILLO	9	EISIC
J4	MARIA JOSE MENDEZ	10	EISIC
J5	ADONIS PABON	9	EISIC

J1 Escuela = CIME

Jno	NOMBRE	NOTA	ESCUELA
J1	LUIS YANEZ	8	CIME
J2	ERIKA QUIROZ	8	CIME

J2 Escuela = EISIC

Jno	NOMBRE	NOTA	ESCUELA
J3	DANIEL MURILLO	9	EISIC
J4	MARIA JOSE MENDEZ	10	EISIC
J5	ADONIS PABON	9	EISIC

La fragmentación horizontal derivada Consiste en dividir una relación partiendo de los predicados definidos sobre alguna otra, debido a que la relación R depende de la relación Q, sobre cuyos atributos está definido el predicado de la fragmentación

Las tres entradas necesarias para desarrollar la fragmentación horizontal derivada son las siguientes: el conjunto de particiones de la relación propietaria, la relación miembro y el conjunto de predicados resultados de aplicar el semi-yunto entre la propietaria y la miembro.



Es decir La fragmentación de la tabla primaria, se aplica a tablas secundarias, o bien se debe partir de una fragmentación horizontal primaria.

Nota: La relación al final de la conexión es llamada *propietario* de la conexión y la relación que está en la cabeza de la conexión es llamada *miembro*.

Ejemplo: Considere las Relaciones

ID	MATERIA	ESCUELA	CRÉDITOS	NIVEL
1	Análisis Matemático	EISIC	4	1
2	Sistemas Operativos	EISIC	6	3
3	Programación II	CIME	6	2
4	Tecnología Eléctrica	EISIC	4	2
5	Técnicas de Aprendizaje	CIME	4	1
6	Dibujo Mecánico	CIME	6	3

Jno	NOMBRE	MATERIA	NOTA
J1	LUIS YANEZ	Sistemas Operativos	8
J2	ERIKA QUIROZ	Dibujo Mecánico	8
J3	DANIEL MURILLO	Técnicas de Aprendizaje	9
J4	MARIA JOSE MENDEZ	Análisis Matemático	10
J5	ADONIS PABON	Programación II	9

Esta relación puede ser fragmentada horizontalmente basada en la escuela en la cual el estudiante está matriculado. La escuela sin embargo no es un atributo del estudiante.

Escuela = CIME (Fragmentación Horizontal primaria)

ID	MATERIA	ESCUELA	CRÉDITOS	NIVEL
1	Análisis Matemático	EISIC	4	1
2	Sistemas Operativos	EISIC	6	3
4	Tecnología Eléctrica	EISIC	4	2

Fragmentación Horizontal derivada

Jno	NOMBRE	MATERIA	NOTA
J1	LUIS YANEZ	Sistemas Operativos	8
J4	MARIA JOSE MENDEZ	Análisis Matemático	10



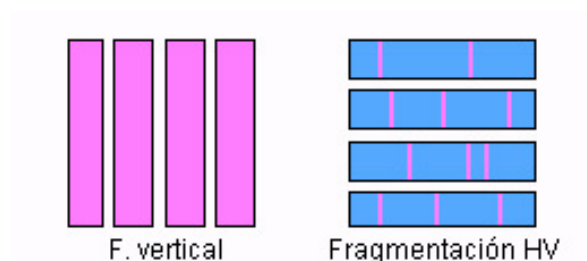
Escuela = EISIC (Fragmentación Horizontal primaria)

ID	MATERIA	ESCUELA	CRÉDITOS	NIVEL
3	Programación II	CIME	6	2
5	Técnicas de Aprendizaje	CIME	4	1
6	Dibujo Mecánico	CIME	6	3

Fragmentación Horizontal derivada

Jno	NOMBRE	MATERIA	NOTA
J2	ERIKA QUIROZ	Dibujo Mecánico	8
J3	DANIEL MURILLO	Técnicas de Aprendizaje	9
J5	ADONIS PABON	Programación II	9

2.2.2. FRAGMENTACIÓN VERTICAL



2. Fragmentación Vertical [IMAG.02]⁴

El objetivo de la fragmentación vertical consiste en dividir la relación en un conjunto de relaciones más pequeñas tal que algunas de las aplicaciones de usuario sólo hagan uso de un fragmento. Sobre este marco, una fragmentación óptima es aquella que produce un esquema de división que minimiza el tiempo de ejecución de las aplicaciones que emplean esos fragmentos.

⁴ http://cablemodem.fibertel.com.ar/bucanero/facultad/Lab4_TP6.doc



La fragmentación vertical se basa en los atributos de la relación para realizar la división, es decir: la subdivisión de atributos en grupos. La fragmentación es correcta si cada atributo se mapea en al menos un atributo del fragmento.

La partición vertical resulta más complicada que la horizontal. Esto se debe al aumento del número total de alternativas que tenemos disponibles.

Existen dos enfoques heurísticos para la fragmentación vertical de relaciones:

- ✓ **Agrupación:** Comienza asignando cada atributo a un fragmento, y en cada paso, junta algunos de los fragmentos hasta que satisface un determinado criterio. La agrupación surgió en principio para bases de datos centralizadas y se usó posteriormente para las bases de datos distribuidas.

- ✓ **Escisión:** A partir de la relación se deciden que fragmentos resultan mejores, basándose en las características de acceso de las aplicaciones a los atributos. Esta técnica se presentó, también, para bases de datos centralizadas. Posteriormente, se extendió al entorno distribuido.



La escisión genera fragmentos no solapados mientras que la agrupación normalmente produce fragmentos solapados. Dentro del contexto de los sistemas de bases de datos distribuidos, son preferibles los fragmentos no solapados por obvias razones. Los fragmentos no solapados se refieren únicamente a atributos clave no primarios.

Ejemplo: Considere la Relación J

Jno	NOMBRE	NOTA	ESCUELA
J1	LUIS YANEZ	8	CIME
J2	ERIKA QUIROZ	8	CIME
J3	DANIEL MURILLO	9	EISIC
J4	MARIA JOSE MENDEZ	10	EISIC
J5	ADONIS PABON	9	EISIC

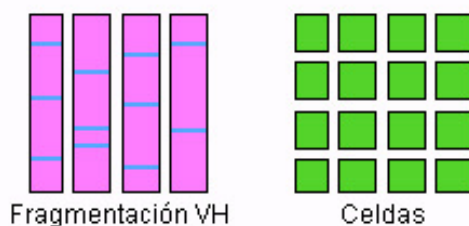
J1 Información de Notas

Jno	NOTA
J1	8
J2	8
J3	9
J4	10
J5	9

J2 Información de Nombres y Escuelas

Jno	NOMBRE	ESCUELA
J1	LUIS YANEZ	CIME
J2	ERIKA QUIROZ	CIME
J3	DANIEL MURILLO	EISIC
J4	MARIA JOSE MENDEZ	EISIC
J5	ADONIS PABON	EISIC

2.2.3. FRAGMENTACIÓN MIXTA



3. Fragmentación Mixta [IMAG.03]⁵

La fragmentación mixta puede llevarse a cabo de tres formas diferentes: desarrollando primero la fragmentación vertical y, posteriormente, aplicando la fragmentación horizontal sobre los fragmentos verticales (denominada partición VH), o aplicando primero una división horizontal para luego, sobre los fragmentos generados, desarrollar una fragmentación vertical (llamada partición HV), o bien, de forma directa considerando la semántica de las transacciones.

Ejemplo: Considere la Relación J

Jno	NOMBRE	NOTA	ESCUELA
J1	LUIS YANEZ	8	CIME
J2	ERIKA QUIROZ	8	CIME
J3	DANIEL MURILLO	9	EISIC
J4	MARIA JOSE MENDEZ	10	EISIC
J5	ADONIS PABON	9	EISIC

⁵ http://cablemodem.fibertel.com.ar/bucanero/facultad/Lab4_TP6.doc



Fragmentación Horizontal Previa

Escuela = EISIC

Jno	NOMBRE	NOTA	ESCUELA
J3	DANIEL MURILLO	9	EISIC
J4	MARIA JOSE MENDEZ	10	EISIC
J5	ADONIS PABON	9	EISIC

Información de Notas (Fragmentación Vertical)

Jno	NOTA
J3	9
J4	10
J5	9

2.2.4. GRADOS DE FRAGMENTACIÓN

El grado de fragmentación puede considerarse en el que cada tupla o atributo forme un fragmento, o una ausencia de la división, considerando a las relaciones como unidades de fragmentación

Se debe buscar un compromiso intermedio, en el que se debería establecerse sobre las características de las aplicaciones que hacen uso de la base de datos, sobre estas una serie de parámetros, se podrá establecer el grado de fragmentación del banco de datos de acuerdo con sus valores.



REGLAS DE CORRECCIÓN DE LA FRAGMENTACIÓN.

Asegurarán la ausencia de cambios semánticos en la base de datos, estas son las tres reglas que se debe cumplir en el proceso de fragmentación.

- 1. Compleción.** Si una relación R se descompone en una serie de fragmentos R1, R2, ..., Rn, cada elemento de datos que pueda encontrarse en R deberá poder encontrarse en uno o varios fragmentos Ri.

$$1 \leq i \leq n \text{ (No hay pérdida de información)}$$

Asegura que los datos de la relación global se proyectan sobre los fragmentos sin pérdida alguna.

En el caso de fragmentación horizontal el elemento de datos, normalmente, es una tupla, mientras que en el caso de fragmentación vertical es un atributo. [www.05]⁶

- 2. Reconstrucción.** Si una relación R se descompone en una serie de fragmentos R1, R2, ..., Rn, puede definirse un operador relacional ∇ tal que el operador será diferente dependiendo de las diferentes formas de fragmentación.

$$R = \nabla R_i \quad \forall R_i \in F_R$$

⁶ http://html.rincondelvago.com/bases-de-datos-distribuidas_1.html



La reconstrucción de la relación a partir de sus fragmentos asegura la preservación de las restricciones definidas sobre los datos en forma de dependencias. [www.06]º

3. Disyunción. Si una relación R se descompone horizontalmente en una serie de fragmentos R1, R2, ..., Rn, y un elemento de datos di se encuentra en algún fragmento Rj, entonces no se encuentra en otro fragmento Rk (k ≠ j). Esta regla asegura que los fragmentos horizontales sean disjuntos. Si una relación R se descompone verticalmente, sus atributos primarios, clave normalmente, se repiten en todos sus fragmentos. [www.07]º

Ejemplo:

Considere la Relación J

Jno	NOMBRE	NOTA	ESCUELA
J1	LUIS YANEZ	8	CIME
J2	ERIKA QUIROZ	8	CIME
J3	DANIEL MURILLO	9	EISIC
J4	MARIA JOSE MENDEZ	10	EISIC
J5	ADONIS PABON	9	EISIC

J1 Escuela = CIME

Jno	NOMBRE	NOTA	ESCUELA
J1	LUIS YANEZ	8	CIME
J2	ERIKA QUIROZ	8	CIME

J2 Escuela = EISIC

Jno	NOMBRE	NOTA	ESCUELA
J3	DANIEL MURILLO	9	EISIC
J4	MARIA JOSE MENDEZ	10	EISIC
J5	ADONIS PABON	9	EISIC



Compleción: No hay pérdida de datos ya que todas las tuplas se encuentran fragmentadas en J1 y J2

Reconstrucción: Se puede reconstruir mediante la unión de todos los fragmentos, es decir $J1 \cup J2$

Disyunción: En la fragmentación $J1$ Escuela = CIME no se va a encontrar EISIC, y en $J2$ Escuela = EISIC no se va a encontrar CIME, entonces J_k ($k \neq j$).

ALTERNATIVAS DE ASIGNACIÓN.

Una vez que se ha realizado la fragmentación correctamente se debe decidir donde serán asignados cada uno de los fragmentos.

Los datos pueden replicarse para mantener una copia al momento que son asignados. La seguridad y a la eficiencia de las consultas de lectura son una de las razones para la réplica. En caso de fallo en el sistema se podría acceder a esos datos ubicados en sitios distintos, además las consultas que acceden a los mismos datos pueden ejecutarse en paralelo, ya que habrá copias en diferentes sitios.

La ejecución de actualizaciones, de escritura, implicará la actualización de todas las copias que existan, este proceso puede ser complicado y problemático, esto se puede resolver calculando la cantidad de consultas de lectura y el número de consultas de



escritura que se llevarán a cabo. Donde las consultas que se llevan a cabo sean la mayor parte de lectura, se podría alcanzar un alto grado de réplica, no así en el caso contrario. Una base de datos fragmentada es aquella donde no existe réplica alguna.

A lo largo de la red los fragmentos se alojan en sitios donde únicamente existe una copia de cada uno de ellos.

Una base de datos se puede considerarla totalmente replicada, cuando existe una copia de todo el banco de datos en cada sitio, o parcialmente replicada cuando existen copias de los fragmentos ubicados en diferentes sitios.

INFORMACIÓN NECESARIA PARA LA FRAGMENTACIÓN

En el diseño de distribución, un aspecto importante es la cantidad de factores, como es: las características de acceso de las aplicaciones a la base de datos, las características del sistema en cada sitio, la organización lógica de la base de datos, la localización de las aplicaciones, éstas tienen una influencia sobre la distribución.

Se divide en las siguientes categorías la información necesaria para el diseño de la distribución.

- La información del banco de datos.
- La información de la aplicación.



- La información sobre la red de ordenadores
- La información sobre los ordenadores en sí.

La información sobre la red de ordenadores y la información sobre los ordenadores en sí, son de carácter cuantitativo y servirán, principalmente, para desarrollar el proceso de asignación.

INFORMACIÓN NECESARIA PARA LA FRAGMENTACIÓN

HORIZONTAL

- **Información sobre la base de datos.**

Es importante saber cómo están conectadas las relaciones de la base de datos una con otra, es decir se refiere a la información del esquema conceptual de la base de datos.

La relación al final de la conexión (relación de origen) es llamada propietario de la conexión y la relación que está en la cabeza de la conexión (relaciones destino) es llamada miembro.

- **Información sobre la aplicación.**

Es Necesario información cualitativa y cuantitativa.

La información cualitativa guiará la fragmentación, mientras que la cuantitativa es requerida en los modelos de asignación.

Referente a la información cuantitativa, se necesita definir dos conjuntos de datos que son:



- ✓ Selectividad mintérmino. Es el número de tuplas de una relación a las que accede una consulta de acuerdo a un predicado mintérmino dado.
- ✓ Frecuencia de acceso. Es la frecuencia con la que un usuario accede a los datos.

INFORMACIÓN NECESARIA PARA LA FRAGMENTACIÓN VERTICAL

La principal información para la fragmentación vertical, es la relacionada a las aplicaciones. Debido a que en el particionamiento vertical coloca en un fragmento aquellos atributos los cuales son accesados juntos.

Los principales datos requeridos relacionados a las aplicaciones son sus frecuencias de acceso.

INFORMACIÓN NECESARIA PARA LA FRAGMENTACIÓN MIXTA

En la etapa de asignación necesitamos los datos cuantitativos acerca de la base de datos, las comunicaciones de la red, las aplicaciones que funcionan sobre ella, el limitantes de almacenamiento para cada sitio en la red y las capacidades de procesamiento.

- **Información de la base de datos.**

Para llevar a cabo una fragmentación horizontal definimos la selección de mintérminos, ahora necesitamos extender la



definición de fragmentos, y definir la selección de un fragmento **F_j** con respecto a una consulta query **q_i**. Este es el número de tuplas de **F_j** que se necesitan ser accesadas para procesar **q_i**.

- **Información de la aplicación.**

Durante la actividad de la fragmentación, la mayoría de la información relacionada a la aplicación ya ha sido compilada, pero en algunos de los casos es requerido por el modelo de asignación

- **Información del sitio.**

Necesitamos conocer la capacidad de almacenamiento y procesamiento de cada sitio, estos valores pueden calcularse por medio de funciones o por estimaciones.

- **Información de la red.**

En nuestro modelo asumimos la existencia de una simple red donde el costo de comunicación es definido en términos de un marco de datos.



3

PARTICIONAMIENTO EN ORACLE



Contenido:

Particionamiento en Oracle

- 3.1. Introducción
- 3.2. Estructura Lógica y Física de la Base de Datos Oracle
- 3.3. Estrategias Básicas de Particionamiento
 - 3.3.1. Particionamiento por Rango
 - 3.3.2. Particionamiento por Lista
 - 3.3.3. Particionamiento por Hash
 - 3.3.4. Partición por Composición
 - 3.3.5. Extensiones de Particionamiento
- 3.4. Bases de Datos Distribuidas en Oracle
- 3.5. Creación y uso del DBLINK en Oracle
- 3.6. Vistas en Oracle
- 3.7. Replicación de Base de Datos en Oracle.



CAPITULO III

3. PARTICIONAMIENTO EN ORACLE

3.1. INTRODUCCIÓN

Oracle brinda una colección de comandos SQL para administrar las tablas particionadas.

El particionado es una técnica de optimización para el mejoramiento de los tiempos de respuesta de las consultas, donde las tablas son muy grandes.

Al crear una tabla se puede elegir qué rangos de datos van almacenados en un tablespace (segmento lógico) u otro, se puede elegir dónde estarán ubicados, es decir en qué discos se guardaran los datafiles (segmentos físicos) de esos tablespace. Se tiene algunas ventajas como:

- **Segmentos de datos más pequeños:** Oracle sabe en qué partición debe buscar cuando hace referencia a la tabla particionada. Esto influye directamente en el rendimiento de las búsquedas porque cada partición es tratada como si fuera una tabla diferente.
- **Índices más pequeños:** Es posible crear índices individuales para cada partición mediante la partición por rangos.



- **Respaldo más rápido:** El respaldo puede hacerse en paralelo, ya que los datos se encuentran en segmentos separados.

BENEFICIOS DEL PARTICIONAMIENTO

El particionamiento puede brindar grandes beneficios, como es una amplia variedad de aplicaciones al mejorar la capacidad de administración, el desempeño y la disponibilidad.

FUNDAMENTOS DEL PARTICIONAMIENTO

Una tabla particionada es idéntica a una tabla no particionada, no se necesitan modificaciones cuando se accede a una tabla particionada utilizando comandos SQL. El particionamiento permite subdividir una tabla, un índice o una tabla organizada por índices en partes más pequeñas. Cada partición tiene su propio nombre, y sus propias características de almacenamiento.

TIPOS DE ÍNDICES PARTICIONADOS.

- **Índices locales:**

Cada partición de un índice local corresponde a una y solo una partición de la tabla subyacente. Un índice local es un índice en una tabla particionada de la misma forma de la tabla



particionada subyacente de tal manera que hereda la partición de la tabla.

- **Índices Particionados Globales:**

Los índices globales solo pueden particionarse utilizando la partición por rango. Un índice particionado global es un índice en una tabla particionada o no particionada que se particiona utilizando una clave de particionamiento de la tabla.

- **Índices Globales No Particionados:**

Un índice global no particionado es esencialmente idéntico a un índice en una tabla no particionada. La estructura del índice no está particionada y no se acopla con la tabla subyacente.

3.2. ESTRUCTURA LÓGICA Y FÍSICA DE LA BASE DE DATOS ORACLE

NIVELES DE ALMACENAMIENTO

La Base de Datos Oracle está constituida a por los siguientes niveles que son:

- Físico: De ficheros
- Lógico: De tablespaces



ESTRUCTURA LÓGICA. Indica la composición y distribución teórica de la base de datos. La estructura lógica sirve para que las aplicaciones puedan utilizar los elementos de la base de datos sin saber realmente cómo se están almacenando. Se divide en unidades de almacenamiento lógicas como son los Tablespaces. Cada Base de Datos estará formada por uno o más tablespaces. Cada tablespace se corresponde con uno o más ficheros de datos. El Tablespace SYSTEM se crea automáticamente al hacer la instalación de Oracle, o al crear una Base de Datos, este Tablespace contiene el diccionario de datos.

ESTRUCTURA FÍSICA. Es la estructura de los datos tan cual se almacenan en las unidades de disco. La correspondencia entre la estructura lógica y la física se almacena en la base de datos es decir en los metadatos.

Una B.D. tiene uno o más ficheros de datos. Estos ficheros son de tamaño fijo y se establecen en el momento en que se crea la base de datos o en el momento en el que se crean tablespaces.

Los datos del fichero de datos son leídos cuando se necesitan y situados en una caché de memoria compartida para que el próximo acceso a los mismos sea más rápido.



CREACIÓN DEL TABLESPACES

Una base de datos está formada por una o varias unidades lógicas llamadas tablespaces, y cada tablespaces está formado por uno o varios ficheros físicos que son los datafiles. Un datafile solamente puede pertenecer a un tablespace.

Cuando se crea una base de datos, hay que crear al menos un tablespace, por lo que durante el proceso de creación de la base de datos siempre se indica el tablespace principal de ésta, que se llama SYSTEM.

Es recomendable crear otro tablespace distinto al SYSTEM de modo que todos los nuevos usuarios, tablas e índices se almacenarán en un tablespace diferente a SYSTEM, para evitar que se bloquee toda la base de datos si ocurre algo en el tablespace SYSTEM.

COMO CREAR UN TABLESPACE.

Debemos iniciar la sesión en la base de datos con un usuario con permisos de administración, y este es el usuario SYSTEM. Esto lo podemos hacer desde el SQLPLUS o desde el Toad. Recordemos que la contraseña del system fue cambiada a oracle al momento de crear la base de datos oracle.

Vamos a crear un tablespace llamado TACADEMICO para nuestra base de datos, la sentencia es la siguiente:



```
CREATE TABLESPACE TACADEMICO DATAFILE 'C:\ORACLE\ACADEMICO\TACADEMICO.DBF'  
SIZE 300M;
```

Con esta sentencia estamos creando en nuestra base de datos un tablespace nuevo llamado "TACADEMICO" y que está formado físicamente por un fichero (datafile) llamado TACADEMICO.dbf de 300 Mbytes y que está en el directorio "C:\oracle\academico\", la carpeta academico fue creada previamente. Esta sentencia crea físicamente el fichero.

Ahora vamos a crear los diferentes tablespaces que utilizaremos para nuestras particiones:

```
CREATE TABLESPACE TEISIC DATAFILE 'H:\ORACLE\EISIC\TEISIC.DBF' SIZE 300M;  
CREATE TABLESPACE TEITEX DATAFILE 'H:\ORACLE\EITEX\TEITEX.DBF' SIZE 300M;  
CREATE TABLESPACE TCIME DATAFILE 'H:\ORACLE\CIME\TCIME.DBF' SIZE 300M;  
CREATE TABLESPACE TCIERCOM DATAFILE 'H:\ORACLE\CIERCOM\TCIERCOM.DBF' SIZE  
300M;  
CREATE TABLESPACE TESDYM DATAFILE 'H:\ORACLE\ESDYM\TESDYM.DBF' SIZE 300M;  
CREATE TABLESPACE TESIIN DATAFILE 'H:\ORACLE\ESIIN\TESIIN.DBF' SIZE 300M;  
CREATE TABLESPACE TDOCENTE1 DATAFILE 'H:\ORACLE\DOCENTE1\TDOCENTE1.DBF' SIZE  
300M;  
CREATE TABLESPACE TDOCENTE2 DATAFILE 'H:\ORACLE\DOCENTE2\TDOCENTE2.DBF' SIZE  
300M;  
CREATE TABLESPACE TDOCENTE3 DATAFILE 'H:\ORACLE\DOCENTE3\TDOCENTE3.DBF' SIZE  
300M;  
CREATE TABLESPACE TDOCENTE4 DATAFILE 'H:\ORACLE\DOCENTE4\TDOCENTE4.DBF' SIZE  
300M;
```

La creación de estos tablespaces no es obligatoria, pero sí recomendable, así cada usuario de la Base de Datos tendrá su propio espacio de datos.

BORRANDO UN TABLESPACE.

Para eliminar un tablespace de la base de datos se debe utilizar la sentencia:

```
Drop tablespace TEISIC;
```



CREACIÓN DE UNA BASE DE DATOS Y TABLAS

Es sencillo crear un nuevo esquema-usuario de Oracle. Para poder realizar estos pasos es necesario iniciar la sesión en la base de datos con un usuario con permisos de administración, y este es el usuario SYSTEM:

Ahora vamos a crear el usuario que va a trabajar sobre el tablespace creado previamente el cual contendrá a toda la base de datos.

```
CREATE USER academico IDENTIFIED BY academico DEFAULT TABLESPACE TACADEMICO;
```

PERMISOS

```
GRANT dba, connect, resource TO academico;  
GRANT CREATE ANY VIEW TO academico WITH ADMIN OPTION;
```

Si no se especifica un tablespace, la Base de Datos le asignará el tablespace USERS, que es el tablespace que se utiliza por defecto para los nuevos usuarios.

ELIMINAR UN USUARIO DE LA BASE DE DATOS

Para eliminar un usuario de la Base de Datos se hace uso de la clausula DROP USER y opcionalmente se puede utilizar CASCADE, para decirle que también elimine todos los objetos creados por ese usuario.

```
DROP USER academico CASCADE;
```

PERMISOS

Se debe asignar permisos necesarios para poder trabajar con nuestra Base de datos.



Oracle incluye tres roles de sistema: CONNECT, RESOURCE y DBA, cuyos privilegios son:

Rol	Privilegios
CONNECT	alter session, create session, create cluster, create table, create view, create synonym, create sequence, create database link
RESOURCE	create cluster, create table, create procedure, create sequence, create trigger
DBA	todos los privilegios de sistema con la opcion with admin option

Privilegios de Oracle

Ahora el usuario ya puede conectarse y comenzar a trabajar sobre su esquema.

Recordemos que para ingresar a nuestro esquema debemos acordarnos del nombre y su contraseña.

Nombre: academico

Contraseña: academico

Una vez ingresado al esquema podemos crear las tabas necesarias, la sentencia es la siguiente:

```
create table ESTUDIANTES (  
  CEDULA          VARCHAR2(10)          not null,  
  NOMBRES         VARCHAR2(40),  
  APELLIDOS       VARCHAR2(40),  
  SEXO            CHAR(1),  
  ID_ESCUELA      INTEGER,  
  constraint PK_ESTUDIANTES primary key (CEDULA)  
)
```

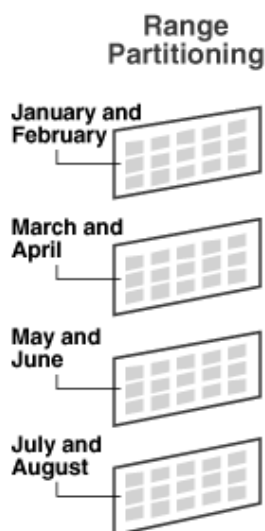


3.3. ESTRATEGIAS BÁSICAS DE PARTICIONAMIENTO

TIPOS DE PARTICIONADO EN ORACLE

- **Oracle 8.0:** Particionamiento por Rango.
- **Oracle 8i:** Particionamiento por Rango, Hash y Composite.
- **Oracle 9iR2/10g:** Particionamiento por Rango, Hash, Composite y el tipo List
- **Oracle 11g:** Columnas virtuales para particionar (que no existen físicamente en la tabla), particionado de Sistema (donde podemos gestionar directamente en que partición de la tabla se insertan los registros) y el particionado por Intervalos.

3.3.1. PARTICIONAMIENTO POR RANGO



4. Particionamiento por Rango [IMAG.04]⁷

⁷ <http://www.dataprix.com/blogs/respinosamilla/particionado-tablas-oracle>



Los datos se distribuyen de acuerdo con el rango de valores de la clave de particionamiento. La distribución de datos es continua.

Se requiere que los registros estén identificado por un “partition key” relacionado por un predefinido rango de valores, el valor de las columnas “partition key” determina la partición a la cual pertenecerá el registro.

Se deben considerar las siguientes reglas:

- Cada partición se define con la clausula VALUES LESS THAN, la que indica el límite superior no inclusive para las particiones, cualquier valor de la clave de la partición igual o superior, es añadida a la próxima partición.
- Todas las particiones, excepto la primera, tienen un límite inferior, especificado en la clausula VALUES LESS THAN de la partición previa.
- Un literal MAXVALUE puede ser definido para la última partición; representa un valor virtual de infinito.

Ejemplo

```
create table ESTUDIANTES (  
  CEDULA          VARCHAR2(10)          not null,  
  NOMBRES         VARCHAR2(40),  
  APELLIDOS       VARCHAR2(40),  
  SEXO            CHAR(1),  
  ID_ESCUELA      INTEGER,  
  constraint PK_ESTUDIANTES primary key (CEDULA)  
)  
PARTITION BY RANGE (id_escuela)  
  
(PARTITION EISC VALUES LESS THAN (2) TABLESPACE TEISIC,  
PARTITION EITEX VALUES LESS THAN (3) TABLESPACE TEITEX,  
PARTITION CIME VALUES LESS THAN (4) TABLESPACE TCIME,  
PARTITION CIERCOM VALUES LESS THAN (5) TABLESPACE TCIERCOM;
```



TABLA ESTUDIANTES (sin partición)

CEDULA	NOMBRES	APELLIDOS	SEXO	ID_ESCUELA
1001766284	JUANA NARCISA	BUITRON DOMINGUEZ	F	2
1002222915	MARTHA CENEIDA	LOMAS NIETO	F	1
0401067871	FLOR DEL ROCIO	PABON POZO	F	3
1002673828	MARIA TATIANA	ACOSTA CAICEDO	F	2
1001645009	LOURDES ZENEIDA	VALLES FIERRO	F	4
1704464633	CARLOS VICENTE	PINEDA PALACIOS	M	2
1001607702	LUIS ALFREDO	AVENDAÑO DIAZ	M	4
0801518648	ELISA ESIL	ZAMBRANO DEL VALLE	F	4
1201467113	ARTURO NARCISO	UBE LUCES	M	1
1302112907	EUCLIDES OJILVIE	MENENDEZ LOOR	M	4
1001537321	SILVIA SUSANA	RIVERA GOMEZ	F	3
1600355794	BLADIMIR RUBEN	TERAN NARVAEZ	M	2
0802729202	MARIUXI BETSY	BONE GARRIDO	F	1
0801303348	WILVER ALEXANDER	SALAS GUERRERO	M	4
1002828588	ANGELICA CAROLA	MORAN ACOSTA	F	2
1302871817	PAULA HENOES	SANCHEZ MEZA	F	3
1304148750	JOSE RENSON	ZAMBRANO MOREIRA	M	1
1305584227	SONIA ESTHER	MACIAS LOOR	F	2
1306687565	ANTONIO GEOVANNY	ALCIVAR CHAVEZ	M	2
1307507135	GIANI GUILLERMO	CEDEÑO GARCIA	M	1

Tabla Estudiantes Particionada por Rango (EISIC)

CEDULA	NOMBRES	APELLIDOS	SEXO	ID_ESCUELA
1002222915	MARTHA CENEIDA	LOMAS NIETO	F	1
1201467113	ARTURO NARCISO	UBE LUCES	M	1
0802729202	MARIUXI BETSY	BONE GARRIDO	F	1
1304148750	JOSE RENSON	ZAMBRANO MOREIRA	M	1
1307507135	GIANI GUILLERMO	CEDEÑO GARCIA	M	1

Tabla Estudiantes Particionada por Rango (EITEX)

CEDULA	NOMBRES	APELLIDOS	SEXO	ID_ESCUELA
1001766284	JUANA NARCISA	BUITRON DOMINGUEZ	F	2
1002673828	MARIA TATIANA	ACOSTA CAICEDO	F	2
1704464633	CARLOS VICENTE	PINEDA PALACIOS	M	2
1600355794	BLADIMIR RUBEN	TERAN NARVAEZ	M	2
1002828588	ANGELICA CAROLA	MORAN ACOSTA	F	2
1305584227	SONIA ESTHER	MACIAS LOOR	F	2
1306687565	ANTONIO GEOVANNY	ALCIVAR CHAVEZ	M	2

Tabla Estudiantes Particionada por Rango (CIME)

CEDULA	NOMBRES	APELLIDOS	SEXO	ID_ESCUELA
0401067871	FLOR DEL ROCIO	PABON POZO	F	3
1001537321	SILVIA SUSANA	RIVERA GOMEZ	F	3
1302871817	PAULA HENOES	SANCHEZ MEZA	F	3

Tabla Estudiantes Particionada por Rango (CIERCOM)

CEDULA	NOMBRES	APELLIDOS	SEXO	ID_ESCUELA
1001645009	LOURDES ZENEIDA	VALLES FIERRO	F	4
1001607702	LUIS ALFREDO	AVENDAÑO DIAZ	M	4
0801518648	ELISA ESIL	ZAMBRANO DEL VALLE	F	4
1302112907	EUCLIDES OJILVIE	MENENDEZ LOOR	M	4
0801303348	WILVER ALEXANDER	SALAS GUERRERO	M	4



Se puede realizar este particionamiento cuando se tiene datos que tienen rango lógicos y que pueden ser distribuidos por este, como por ejemplo: mes del año o un valor numérico.

3.3.2. PARTICIONAMIENTO POR LISTA



5. Particionamiento por Lista [IMAG.05]⁸

El particionamiento por lista no soporta claves de particionamiento formada por varios atributos.

La clave de particionado es una lista de valores, que determina cada una de las particiones, la distribución de datos se define por el listado de valores de la clave de partición. El valor DEFAULT sirve para definir la partición donde iran el resto de registros que no cumplen ninguna condición de las diferentes particiones.

⁸ <http://www.dataprix.com/blogs/respinosamilla/particionado-tablas-oracle>



Ejemplo:

```

create table MATERIAS (
  ID_MATERIA      INTEGER not null,
  MATERIA        VARCHAR2(50),
  ID_ESCUELA     INTEGER,
  MATERIA_ANTERIOR INTEGER,
  ID_NIVEL       INTEGER,
  constraint PK_MATERIAS primary key (ID_MATERIA)
)
PARTITION BY LIST (id_escuela)
(
  PARTITION EISIC values(1)TABLESPACE TEISIC,
  PARTITION EITEX values(2)TABLESPACE TEITEX,
  PARTITION CIME values(3)TABLESPACE TCIME,
  PARTITION CIERCOM values(4)TABLESPACE TCIERCOM,
  PARTITION ESDYM values(5)TABLESPACE TESDYM,
  PARTITION ESIIN values(6)TABLESPACE TESIIN)
  
```

TABLA MATERIAS (sin partición)

ID_MATERIA	MATERIA	ID_ESCUELA	MATERIA_ANTERIOR	ID_NIVEL
1	Analisis Matematico I	1		1
2	Analisis Matematico II	1	1	2
3	Ecuaciones Diferenciales	1	2	3
4	Matematicas Aplicadas	1	3	4
5	Modelacion y Simulacion de Computadoras	1		5
6	Fibrologia	2		1
7	Introduccion al Hilado	2		2
8	Quimica Organica I	2		3
9	Quimica Organica II	2	8	4
10	Control de Procesos I	2		5
11	Introduccion a la Ingenieria	3		1
12	Dibujo Mecanico I	3		2
13	Dibujo Mecanico II	3	12	3
14	Maquinas Herramientas	3	13	4
15	Emprendimiento e Innovacion Tecnologica	3		5
16	Tecnicas de Aprendizaje	4		1
17	Expresion Oral y Escrita	4		2
18	Sistemas Operativos	4		3
19	Base de Datos	4	18	4
20	Programacion de Sistemas Multimedia	4	19	5

Tabla Materias Particionada por LIST (EISIC)

ID_MATERIA	MATERIA	ID_ESCUELA	MATERIA_ANTERIOR	ID_NIVEL
1	Analisis Matematico I	1		1
2	Analisis Matematico II	1	1	2
3	Ecuaciones Diferenciales	1	2	3
4	Matematicas Aplicadas	1	3	4
5	Modelacion y Simulacion de Computadoras	1		5

Tabla Materias Particionada por LIST (EITEX)

ID_MATERIA	MATERIA	ID_ESCUELA	MATERIA_ANTERIOR	ID_NIVEL
6	Fibrologia	2		1
7	Introduccion al Hilado	2		2
8	Quimica Organica I	2		3
9	Quimica Organica II	2	8	4
10	Control de Procesos I	2		5



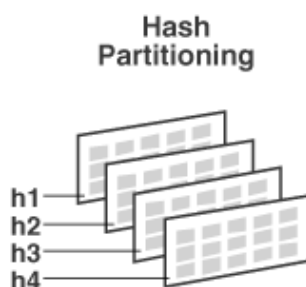
Tabla Materias Particionada por LIST (CIME)

ID MATERIA	MATERIA	ID ESCUELA	MATERIA ANTERIOR	ID NIVEL
11	Introduccion a la Ingenieria	3		1
12	Dibujo Mecanico I	3		2
13	Dibujo Mecanico II	3	12	3
14	Maquinas Herramientas	3	13	4
15	Emprendimiento e Innovacion Tecnologica	3		5

Tabla Materias Particionada por LIST (CIERCOM)

16	Tecnicas de Aprendizaje	4		1
17	Expresion Oral y Escrita	4		2
18	Sistemas Operativos	4		3
19	Base de Datos	4	18	4
20	Programacion de Sistemas Multimedia	4	19	5

3.3.3. PARTICIONAMIENTO POR HASH



6. Particionamiento por Hash [IMAG.06]⁹

La correspondencia entre las filas y las particiones se realiza a través de una función de hash. Es una opción útil cuando:

- Se desconoce la correspondencia en función de los rangos o no hay unos criterios de particionado claros.
- El rango de las particiones difiere sustancialmente o es difícil balancearla manualmente.

⁹ <http://www.dataprix.com/blogs/respinosamilla/particionado-tablas-oracle>



La clave de particionado es una función hash, aplicada sobre una columna, que tiene como objetivo realizar una distribución equitativa de los registros sobre las diferentes particiones.

La función hash devuelve un valor automático que determina a qué partición irá el registro. Es una forma automática de balancear el particionado.

Se puede definir la partición sin indicar los nombres de las particiones, solo poniendo el número de particiones deseadas.

Ejemplo:

```
create table DOCENTES (  
  CEDULA_DOCENTE  VARCHAR2(12)          not null,  
  NOMBRES         VARCHAR2(40),  
  APELLIDOS       VARCHAR2(40),  
  ID_TITULOS      INTEGER,  
  constraint PK_DOCENTES primary key (CEDULA_DOCENTE)  
)  
PARTITION BY HASH (cedula_docente)  
  partitions 4 store in (tdocente1, tdocente2,tdocente3,tdocente4);
```

El sistema creará automáticamente los nombres asignando cada partición a un diferente tablespace.

Igualmente, se pueden indicar los nombres de cada partición individual o los tablespaces donde se localizarán cada una de ellas:



Ejemplo:

```
create table DOCENTES (
  CEDULA_DOCENTE  VARCHAR2(12)          not null,
  NOMBRES         VARCHAR2(40),
  APELLIDOS       VARCHAR2(40),
  ID_TITULOS      INTEGER,
  constraint PK_DOCENTES primary key (CEDULA_DOCENTE)
) PARTITION BY HASH (cedula_docente)
(
  PARTITION DOCENT1 TABLESPACE tdocente1,
  PARTITION DOCENT2 TABLESPACE tdocente2,
  PARTITION DOCENT3 TABLESPACE tdocente3,
  PARTITION DOCENT4 TABLESPACE tdocente4);
```

TABLA DOCENTES (sin partición)

CEDULA_DOCENTE	NOMBRES	APELLIDOS	ID_TITULOS
1001	JAIME	AGUAS	1
1002	HUMBERTO	BRAVO	1
1003	CATALINA	RAMIREZ	2
1004	ENDERSON	LARA	2
1005	PAUL	ANDRADE	2
1006	IRVING	REASCOS	1
1007	JORGE	CARAGUAY	1
1008	RODRIGO	NARANJO	1
1009	MARCELO	JURADO	1
1010	HUGO	IMBAQUINGO	4
1011	WIDMAR	AGUILAR	1
1012	DANIEL	JARAMILLO	1
1013	MIGUEL	ORQUERA	1
1014	JORGE	PORRAS	1
1015	JAIME	ALVARADO	1
1016	NANCY	CERVANTES	1
1017	JOSE	HUACA	1
1018	LUIS	ROMAN	5
1019	IVAN	GARCIA	1
1020	CARPIO	PINEDA	1

Tabla Docentes Particionada por HASH (DOCENTE1)

CEDULA_DOCENTE	NOMBRES	APELLIDOS	ID_TITULOS
1001	JAIME	AGUAS	1
1002	HUMBERTO	BRAVO	1
1003	CATALINA	RAMIREZ	2
1004	ENDERSON	LARA	2
1005	PAUL	ANDRADE	2

Tabla Docentes Particionada por HASH (DOCENTE2)

CEDULA_DOCENTE	NOMBRES	APELLIDOS	ID_TITULOS
1006	IRVING	REASCOS	1
1007	JORGE	CARAGUAY	1
1008	RODRIGO	NARANJO	1
1009	MARCELO	JURADO	1
1010	HUGO	IMBAQUINGO	4



Tabla Docentes Particionada por HASH (DOCENTE3)

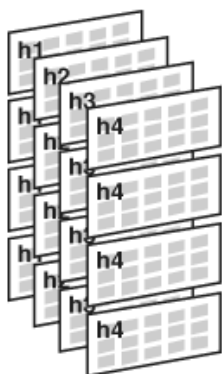
CEDULA_DOCENTE	NOMBRES	APELLIDOS	ID_TITULOS
1011	WIDMAR	AGUILAR	1
1012	DANIEL	JARAMILLO	1
1013	MIGUEL	ORQUERA	1
1014	JORGE	PORRAS	1
1015	JAIME	ALVARADO	1

Tabla Docentes Particionada por HASH (DOCENTE4)

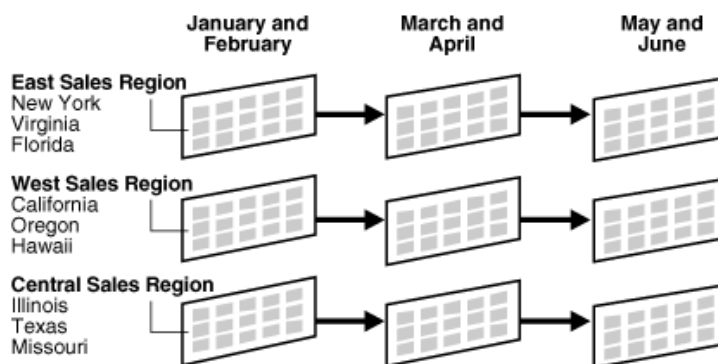
CEDULA_DOCENTE	NOMBRES	APELLIDOS	ID_TITULOS
1016	NANCY	CERVANTES	1
1017	JOSE	HUACA	1
1018	LUIS	ROMAN	5
1019	IVAN	GARCIA	1
1020	CARPIO	PINEDA	1

3.3.4. PARTICIÓN POR COMPOSICIÓN

Composite Partitioning Range-Hash



Composite Partitioning Range - List



7. Particionamiento por Composición [IMAG.07]¹⁰

Oracle nos permite utilizar métodos de particionado compuestos, ya que se conjuga el uso de dos particionados a la vez. Primero la tabla se particiona con un primer método de distribución de datos

¹⁰ <http://www.dataprix.com/blogs/respinosamilla/particionado-tablas-oracle>



y luego cada partición se vuelve a dividir en subparticiones utilizando un segundo método de distribución de datos.

Las técnicas de partición compuesta disponibles son:

- Rango - Hash
- Rango - List
- Rango - Rango
- List - Rango
- List - List
- List - Hash

Las particiones más generales se hacen con el método de rango, cada partición se sub-particiona con el método de hash o por lista.

Ejemplo:

```
/*=====*/
/* Table: ESTUDIANTES PARTITION COMPOSICION RANGE Y HASH */
/*=====*/

create table ESTUDIANTES (
  CEDULA          VARCHAR2(10) not null,
  NOMBRES         VARCHAR2(40),
  APELLIDOS       VARCHAR2(40),
  SEXO            CHAR(1),
  ID_ESCUELA      INTEGER,
  constraint PK_ESTUDIANTES primary key (CEDULA)
)
PARTITION BY RANGE (id_escuela)
  SUBPARTITION BY HASH (SEXO)
  SUBPARTITION TEMPLATE (
    SUBPARTITION MASCULINO,
    SUBPARTITION FEMENINO)

(PARTITION EISIC VALUES LESS THAN (2) TABLESPACE TEISIC,
PARTITION EITEX VALUES LESS THAN (3) TABLESPACE TEITEX,
PARTITION CIME VALUES LESS THAN (4) TABLESPACE TCIME,
PARTITION CIERCOM VALUES LESS THAN (5) TABLESPACE TCIERCOM,);
```



TABLA ESTUDIANTES (sin partición)

CEDULA	NOMBRES	APELLIDOS	SEXO	ID_ESCUELA
1001766284	JUANA NARCISA	BUITRON DOMINGUEZ	F	2
1002222915	MARTHA CENEIDA	LOMAS NIETO	F	1
0401067871	FLOR DEL ROCIO	PABON POZO	F	3
1002673828	MARIA TATIANA	ACOSTA CAICEDO	F	2
1001645009	LOURDES ZENEIDA	VALLES FIERRO	F	4
1704464633	CARLOS VICENTE	PINEDA PALACIOS	M	2
1001607702	LUIS ALFREDO	AVENDAÑO DIAZ	M	4
0801518648	ELISA ESIL	ZAMBRANO DEL VALLE	F	4
1201467113	ARTURO NARCISO	UBE LUCES	M	1
1302112907	EUCLIDES OJILVIE	MENENDEZ LOOR	M	4
1001537321	SILVIA SUSANA	RIVERA GOMEZ	F	3
1600355794	BLADIMIR RUBEN	TERAN NARVAEZ	M	2
0802729202	MARIUXI BETSY	BONE GARRIDO	F	1
0801303348	WILVER ALEXANDER	SALAS GUERRERO	M	4
1002828588	ANGELICA CAROLA	MORAN ACOSTA	F	2
1302871817	PAULA HENOES	SANCHEZ MEZA	F	3
1304148750	JOSE RENSON	ZAMBRANO MOREIRA	M	1
1305584227	SONIA ESTHER	MACIAS LOOR	F	2
1306687565	ANTONIO GEOVANNY	ALCIVAR CHAVEZ	M	2
1307507135	GIANI GUILLERMO	CEDEÑO GARCIA	M	1

Tabla Estudiantes Particionada por Rango (EISIC)

CEDULA	NOMBRES	APELLIDOS	SEXO	ID_ESCUELA
1002222915	MARTHA CENEIDA	LOMAS NIETO	F	1
1201467113	ARTURO NARCISO	UBE LUCES	M	1
0802729202	MARIUXI BETSY	BONE GARRIDO	F	1
1304148750	JOSE RENSON	ZAMBRANO MOREIRA	M	1
1307507135	GIANI GUILLERMO	CEDEÑO GARCIA	M	1

Tabla Estudiantes Sub-Particionada por Hash (Masculino-EISIC)

CEDULA	NOMBRES	APELLIDOS	SEXO	ID_ESCUELA
1201467113	ARTURO NARCISO	UBE LUCES	M	1
1304148750	JOSE RENSON	ZAMBRANO MOREIRA	M	1
1307507135	GIANI GUILLERMO	CEDEÑO GARCIA	M	1

Tabla Estudiantes Sub-Particionada por Hash (Femenino-EISIC)

CEDULA	NOMBRES	APELLIDOS	SEXO	ID_ESCUELA
1002222915	MARTHA CENEIDA	LOMAS NIETO	F	1
0802729202	MARIUXI BETSY	BONE GARRIDO	F	1

Tabla Estudiantes Particionada por Rango (EITEX)

CEDULA	NOMBRES	APELLIDOS	SEXO	ID_ESCUELA
1001766284	JUANA NARCISA	BUITRON DOMINGUEZ	F	2
1002673828	MARIA TATIANA	ACOSTA CAICEDO	F	2
1704464633	CARLOS VICENTE	PINEDA PALACIOS	M	2
1600355794	BLADIMIR RUBEN	TERAN NARVAEZ	M	2
1002828588	ANGELICA CAROLA	MORAN ACOSTA	F	2
1305584227	SONIA ESTHER	MACIAS LOOR	F	2
1306687565	ANTONIO GEOVANNY	ALCIVAR CHAVEZ	M	2



Tabla Estudiantes Sub-Particionada por Hash (Masculino-EITEX)

CEDULA	NOMBRES	APELLIDOS	SEXO	ID_ESCUELA
1704464633	CARLOS VICENTE	PINEDA PALACIOS	M	2
1600355794	BLADIMIR RUBEN	TERAN NARVAEZ	M	2
1306687565	ANTONIO GEOVANNY	ALCIVAR CHAVEZ	M	2

Tabla Estudiantes Sub-Particionada por Hash (Femenino-EITEX)

CEDULA	NOMBRES	APELLIDOS	SEXO	ID_ESCUELA
1001766284	JUANA NARCISA	BUITRON DOMINGUEZ	F	2
1002673828	MARIA TATIANA	ACOSTA CAICEDO	F	2
1002828588	ANGELICA CAROLA	MORAN ACOSTA	F	2
1305584227	SONIA ESTHER	MACIAS LOOR	F	2

Tabla Estudiantes Particionada por Rango (CIME)

CEDULA	NOMBRES	APELLIDOS	SEXO	ID_ESCUELA
0401067871	FLOR DEL ROCIO	PABON POZO	F	3
1001537321	SILVIA SUSANA	RIVERA GOMEZ	F	3
1302871817	PAULA HENOES	SANCHEZ MEZA	F	3

Tabla Estudiantes Sub-Particionada por Hash (Masculino-CIME)

CEDULA	NOMBRES	APELLIDOS	SEXO	ID_ESCUELA
--------	---------	-----------	------	------------

Tabla Estudiantes Sub-Particionada por Hash (Femenino-CIME)

CEDULA	NOMBRES	APELLIDOS	SEXO	ID_ESCUELA
0401067871	FLOR DEL ROCIO	PABON POZO	F	3
1001537321	SILVIA SUSANA	RIVERA GOMEZ	F	3
1302871817	PAULA HENOES	SANCHEZ MEZA	F	3

Tabla Estudiantes Particionada por Rango (CIERCOM)

CEDULA	NOMBRES	APELLIDOS	SEXO	ID_ESCUELA
1001645009	LOURDES ZENEIDA	VALLES FIERRO	F	4
1001607702	LUIS ALFREDO	AVENDAÑO DIAZ	M	4
0801518648	ELISA ESIL	ZAMBRANO DEL VALLE	F	4
1302112907	EUCLIDES OJILVIE	MENENDEZ LOOR	M	4
0801303348	WILVER ALEXANDER	SALAS GUERRERO	M	4



Tabla Estudiantes Sub-Particionada por Hash (Masculino-CIERCOM)

CEDULA	NOMBRES	APELLIDOS	SEXO	ID_ESCUELA
1001607702	LUIS ALFREDO	AVENDAÑO DIAZ	M	4
1302112907	EUCLIDES OJILVIE	MENENDEZ LOOR	M	4
0801303348	WILVER ALEXANDER	SALAS GUERRERO	M	4

Tabla Estudiantes Sub-Particionada por Hash (Femenino-CIERCOM)

CEDULA	NOMBRES	APELLIDOS	SEXO	ID_ESCUELA
1001645009	LOURDES ZENEIDA	VALLES FIERRO	F	4
0801518648	ELISA ESIL	ZAMBRANO DEL VALLE	F	4

```

/*=====*/
/* Table: ESTUDIANTES PARTITION COMPOSICION RANGE Y LIST */
/*=====*/

```

```

create table ESTUDIANTES (
  CEDULA          VARCHAR2(10) not null,
  NOMBRES         VARCHAR2(40),
  APELLIDOS       VARCHAR2(40),
  SEXO            CHAR(1),
  ID_ESCUELA      INTEGER,
  constraint PK_ESTUDIANTES primary key (CEDULA)
)

```

```

PARTITION BY RANGE (ID_ESCUELA)
SUBPARTITION BY LIST (SEXO)
SUBPARTITION TEMPLATE
(SUBPARTITION MASCULINO values('M'),
SUBPARTITION FEMENINO values('F'))

```

```

(PARTITION EISIC VALUES LESS THAN (2) TABLESPACE TEISIC,
PARTITION EITEX VALUES LESS THAN (3) TABLESPACE TEITEX,
PARTITION CIME VALUES LESS THAN (4) TABLESPACE TCIME,
PARTITION CIERCOM VALUES LESS THAN (5) TABLESPACE TCIERCOM,
PARTITION ESDYM VALUES LESS THAN (6) TABLESPACE TESDYM,
PARTITION ESIIN VALUES LESS THAN (7) TABLESPACE TESIIN );

```

3.3.5. EXTENSIONES DE PARTICIONAMIENTO

Oracle Database 11g brinda extensiones de particionamiento, su principal objetivo es:

- Mejorar significativamente la capacidad de administración de una tabla particionada.
- Extender la flexibilidad para definir una clave de particionamiento.



Las extensiones son:

- Particionamiento por Intervalos
- Particionado System
- Particionamiento basado en Columnas Virtuales

PARTICIONAMIENTO POR INTERVALOS

El particionado Interval es complementario a las técnicas de particionado vistas anteriormente.

El particionamiento por intervalos mejora notablemente la capacidad de administración de una tabla particionada.

Oracle extiende las capacidades del método de rangos para definir los rangos igualmente particionados utilizando una definición de intervalo, en lugar de especificar manualmente los rangos individuales.

Oracle creará automáticamente las particiones a medida que sea necesario, cuando se inserta un nuevo registro en la base de datos.

Las técnicas de particionamiento son:

- Interval
- Interval - List
- Interval - Range
- Interval - Hash



Ejemplo:

```

CREATE TABLE interval_tab (
  id      NUMBER,
  code    VARCHAR2(10),
  description VARCHAR2(50),
  created_date DATE
)
PARTITION BY RANGE (created_date)
INTERVAL (NUMTOYMINTERVAL(1,'MONTH'))
(
  PARTITION part_01 values LESS THAN (TO_DATE('01-NOV-2007','DD-MON-YYYY'))
);

```

Mientras se ingresa valores menores se ingresaran a la partición existente caso contrario se crean nuevas particiones

```

INSERT INTO interval_tab VALUES (1, 'ONE', 'One', TO_DATE('16-OCT-2007', 'DD-MON-YYYY'));
INSERT INTO interval_tab VALUES (2, 'TWO', 'Two', TO_DATE('31-OCT-2007', 'DD-MON-YYYY'));

```

Como son fechas menores no tenemos ningún inconveniente

ID	CODE	DESCRIPCIÓN	CREATED_DATE
1	One	One	16/10/2007
2	two	two	31/10/2007

Si agregamos datos fuera del alcance de la partición existente, se creara una nueva partición.

```

INSERT INTO interval_tab VALUES (3, 'THREE', 'Three', TO_DATE('01-NOV-2007', 'DD-MON-YYYY'));
INSERT INTO interval_tab VALUES (4, 'FOUR', 'Four', TO_DATE('30-NOV-2007', 'DD-MON-YYYY'));

```

ID	CODE	DESCRIPCIÓN	CREATED_DATE
1	One	One	16/10/2007
2	two	two	31/10/2007
3	three	three	01/11/2007
4	four	four	30/11/2007

Las particiones quedan de la siguiente manera:

La partición que nosotros creamos se llama partition(part_01)

ID	CODE	DESCRIPCIÓN	CREATED_DATE
1	One	One	16/10/2007
2	two	two	31/10/2007



El sistema creó una nueva partición con el nombre partition(sys_p41)

ID	CODE	DESCRIPCIÓN	CREATED_DATE
3	three	three	01/11/2007
4	four	four	30/11/2007

PARTICIONADO SYSTEM

El particionado system es útil para aplicaciones nosotros gestionemos la forma en la que se realiza el particionado.

Oracle no realiza la gestión del lugar donde se almacenaran los registros, ya que seremos nosotros los que debemos indicar en qué partición se hacen las inserciones.

Ejemplo:

```
create table ESCUELAS (  
  ID_ESCUELA      INTEGER not null,  
  ID_FACULTAD     INTEGER,  
  ESCUELA         VARCHAR2(100),  
  SIGLAS          VARCHAR2(10),  
  constraint PK_ESCUELAS primary key (ID_ESCUELA)  
)  
  PARTITION BY SYSTEM (  
    partition p1,  
    partition p2,  
    partition p3,  
    partition p4,  
    partition p5,  
    partition p6);
```

Al momento de realizar un insert sobre la tabla se debería hacer lo siguiente:

```
INSERT INTO ESCUELAS PARTITION (P1) VALUES (1, 1, 'Ingenieria en Sistemas', 'EISIC');  
INSERT INTO ESCUELAS PARTITION (P2) Values (2, 1, 'Ingenieria Textil', 'EITEX');  
INSERT INTO ESCUELAS PARTITION (P3) Values (3, 1, 'Ingenieria Mecatronica', 'CIME');  
INSERT INTO ESCUELAS PARTITION (P4) Values (4, 1, 'Ingenieria en Electronica y Redes de  
Comunicacion', 'CIERCOM');  
INSERT INTO ESCUELAS PARTITION (P5) Values (5, 1, 'Ingenieria Diseño Textil y Moda', 'ESDYM');  
Insert into ESCUELAS PARTITION (P6) Values (6, 1, 'Ingenieria Industrial', 'ESIIN');  
COMMIT;
```



Oracle nos permite definir sentencias SQL haciendo referencia a las particiones, como SELECT, INSERT, UPDATE, DELETE, LOCK TABLE :

```
SELECT * FROM schema.table PARTITION(part_name);
```

Esta sintaxis nos proporciona una forma simple de acceder a las particiones individuales como si fueran tablas, y utilizarlas,

PARTICIONAMIENTO BASADO EN COLUMNAS VIRTUALES

El Particionamiento basado en columnas virtuales es soportado con las estrategias básicas de particionamiento, se pueden definir en las tablas columnas virtuales.

Las columnas virtuales pueden ser utilizadas para realizar particionamiento sobre ellas.

Anteriormente una tabla solo podía ser particionada si la clave de partición existía físicamente en la tabla. Las columnas virtuales, eliminan esa restricción y permite que la clave de particionamiento se defina por una expresión, utilizando una o más columnas existentes de una tabla, y almacenando la expresión como metadatos solamente.

Ejemplo:

Se crea una tabla con una columna virtual que represente la primera letra del nombre de usuario de la tabla, para lo que se va a



particionar mediante la primera letra del nombre, si está entre las letras de la “a” a la “g” se insertara en la partición part_a_g, y asi sucesivamente.

```
CREATE TABLE users (
  id NUMBER,
  username VARCHAR2(20),
  first_letter VARCHAR2(1)
  GENERATED ALWAYS AS
  (
    UPPER(SUBSTR(TRIM(username), 1, 1))
  ) VIRTUAL
)
PARTITION BY LIST (first_letter)
(
  PARTITION part_a_g VALUES ('A','B','C','D','E','F','G'),
  PARTITION part_h_n VALUES ('H','I','J','K','L','M','N'),
  PARTITION part_o_u VALUES ('O','P','Q','R','S','T','U'),
  PARTITION part_v_z VALUES ('V','W','X','Y','Z')
);
```

Ingresando los datos de la tabla, debemos tomar en cuenta que la columna virtual no se ingresa ya que ella la obtiene de la misma tabla.

```
INSERT INTO users (id, username) VALUES (1, 'Andy Pandy');
INSERT INTO users (id, username) VALUES (1, 'Burty Basset');
INSERT INTO users (id, username) VALUES (1, 'Harry Hill');
INSERT INTO users (id, username) VALUES (1, 'Iggy Pop');
INSERT INTO users (id, username) VALUES (1, 'Oliver Hardy');
INSERT INTO users (id, username) VALUES (1, 'Peter Pervis');
INSERT INTO users (id, username) VALUES (1, 'Veruca Salt');
INSERT INTO users (id, username) VALUES (1, 'Wiley Cyote');
COMMIT;
```

ID	USERNAME	FIRST_LETTER
1	Andy Pandy	A
1	Burty Basset	B
1	Harry Hill	H
1	Iggy Pop	I
1	Oliver Hardy	O
1	Peter Pervis	P
1	Veruca Salt	V
1	Wiley Cyote	W

Entonces las particiones quedarían de la siguiente manera

PARTICION (PART_A_G)

ID	USERNAME	FIRST_LETTER
1	Andy Pandy	A
1	Burty Basset	B



PARTICION (PART_H_N)

ID	USERNAME	FIRST_LETTER
1	Harry Hill	H
1	Iggy Pop	I

PARTICION (PART_O_U)

ID	USERNAME	FIRST_LETTER
1	Oliver Hardy	O
1	Peter Pervis	P

PARTICION (PART_V_Z)

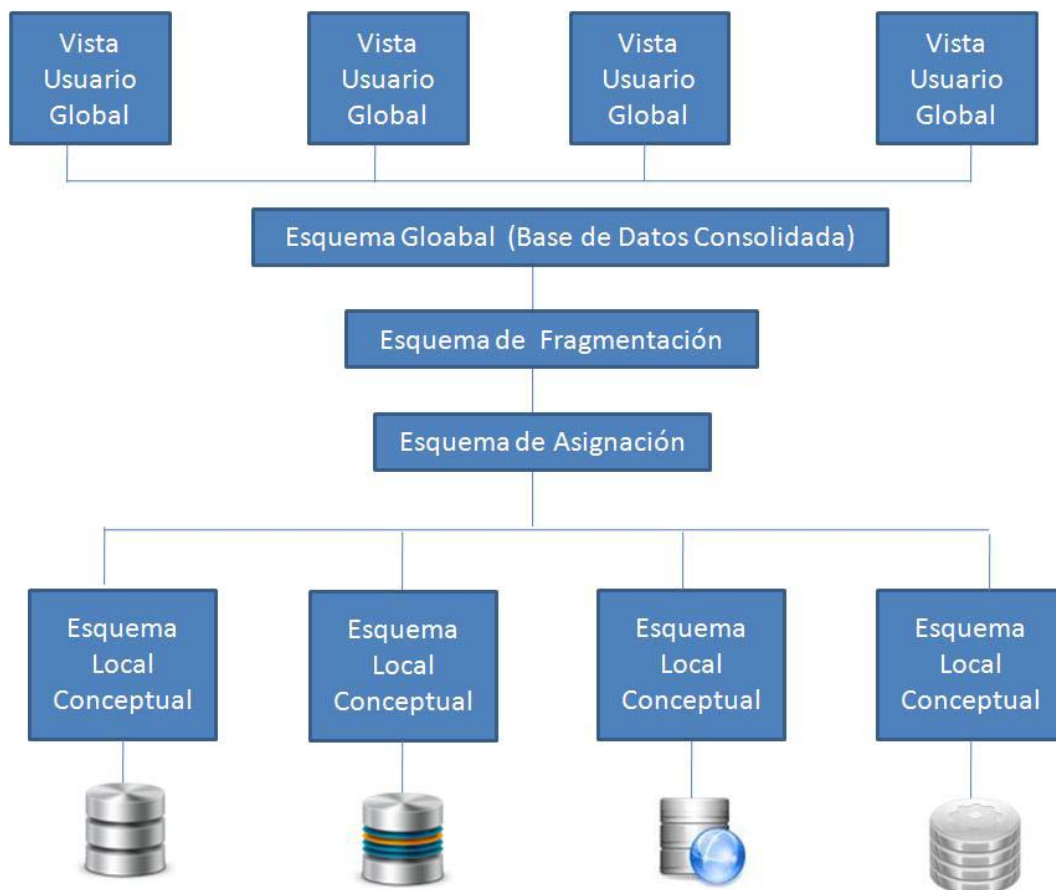
ID	USERNAME	FIRST_LETTER
1	Veruca Salt	V
1	Wiley Cyote	W



3.4. BASES DE DATOS DISTRIBUIDAS EN ORACLE

Un sistema (homogéneo) de bases de datos distribuidas en Oracle es una red de Bases de Datos Oracle que residen en uno o más servidores de modo que es posible acceder a sus datos como si de una única BD se tratara.

El software de red Oracle Net debe ejecutarse en todos los servidores para hacer posible la comunicación entre las Bases de Datos.





3.5. CREACIÓN Y USO DE DBLINKS EN ORACLE

Un Database Link (DBLink) en Oracle permite realizar una conexión desde una base de datos a otra, esta es la manera más sencilla de acceder a tablas y vistas (*views*) de otra base de datos Oracle.

El objetivo principal del Dblink es ocultar los detalles de conexión, facilitando el acceso a los recursos en otras bases de datos, independientemente de que éstas se encuentren instaladas en el mismo servidor o no.

Los DBLinks se crean en la base de datos local utilizando el comando PL/SQL o SQL **CREATE DATABASE LINK**.

El usuario que ejecute el comando anterior debe tener los permisos necesarios para poder hacerlo. La sintaxis del comando es el sigue:

```
CREATE DATABASE LINK Nombre_dblink CONNECT TO Nombre_usuario IDENTIFIED BY  
Contraseña  
USING 'Cadena_conexion';
```

Donde:

Nombre_dblink: Es el nombre del Dblink.

Nombre_usuario y **Contraseña:** Son los identificadores que utilizará el Dblink para conectarse a la base de datos remota.

Cadena_conexion: Identifica a la base de datos remota, este puede ser el nombre de la instancia, esta se define en el archivo `tnsnames.ora` de la base de datos origen.



create database link LINKHILARY connect to academico identified by academico using 'DBHILARY';

BORRAR UN DBLINK

DROP DATABASE LINK nombreLink;

DBLINK COMO PUBLIC

Una vez creado el DBLink, todos los usuarios tendrán acceso al mismo, para referenciar una tabla o vista de la base de datos remota se debe indicar el nombre de la tabla o vista, junto con el carácter "@" y el nombre del DBLink.

Las tablas y vistas remotas pueden usar consultas es decir una sentencia SELECT, se podrán ejecutar también sentencias del tipo DELETE, INSERT, UPDATE o LOCK TABLE.

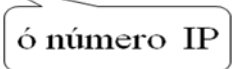

NOMBRE DE SERVICIO

Cada Base de Datos es identificada únicamente en una Base de Datos Distribuida por un nombre global de Base de Datos, éste consta del nombre de la Base de Datos junto con el nombre del host en la red en la que esta Bases de Datos está ubicada, el nombre se hace transparente al usuario mediante el uso de nombres de servicio como es (service names) en la definición de los enlaces como links.

Los nombres de servicio se definen en el archivo tnsnames.ora de Oracle, cuya ubicación depende del cada ordenador:

```
D:\oracle\product\10.2.0\db_1\NETWORK\ADMIN  
\tnsnames.ora
```



```
NombreServiceName =  
(DESCRIPTION =  
  (ADDRESS_LIST =  
    (ADDRESS = (PROTOCOL = TCP)(HOST =  
      NombreOrdenadorEnRed)(PORT = 1521))  
  )  
(CONNECT_DATA =  
  (SERVER = DEDICATED)   
  (SERVICE_NAME = NombreBD)  
)  
)  
) 
```

```
DBHILARY =  
(DESCRIPTION =  
  (ADDRESS_LIST =  
    (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.1.1)(PORT = 1521))  
  )  
(CONNECT_DATA =  
  (SERVICE_NAME = ORCL)  
)  
)
```

CONSULTA A BASES DE DATOS DISTRIBUIDAS REMOTAS Y OBJETOS REMOTOS VIA LINKS

El nombre de un objeto en una Base de Datos es único dentro del esquema. En una Base de Datos remota puede existir un esquema con el mismo nombre y un objeto con el mismo nombre.

El acceso a través de un link a un objeto remoto de un determinado propietario en una Base de Datos remota se realiza mediante la siguiente sentencia:

Propietario.nombreObjeto@nombreLink

ACADEMICO.ESTUDIANTES@LINKHILARY



Para realizar consultas en una Base de Datos Distribuida podemos utilizar objetos situados en una Base de Datos remota, con la siguiente sentencia.

```
select * from MATERIAS@LINKHILARY WHERE ID_ESCUELA=6;
```

3.6. VISTAS EN ORACLE

Una vista es una tabla derivada de otras tablas, estas pueden ser básicas o virtuales. Una vista se caracteriza porque, se considera que forma parte del esquema externo, es una tabla virtual la cual no tiene una correspondencia a nivel físico, las vistas se pueden consultar como cualquier tabla básica, sus actualizaciones se realizan a la o las tablas originales

Se puede utilizar vistas por las siguientes razones:

- Para restringir el acceso a la B.D.
- Para realizar consultas complejas de manera fácil.
- Para obtener una independencia de los datos
- Para presentar diferentes vistas de los mismos datos.

La sintaxis para la creación de vistas en SQL es la siguiente:

```
CREATE VIEW vista AS expresión_tabla
```

Donde:

CREATE VIEW es la orden que permite la creación de la vista.

vista es el nombre de la tabla virtual que se va a crear.

expresión_tabla es una consulta SQL cuyo resultado será el contenido de la vista.

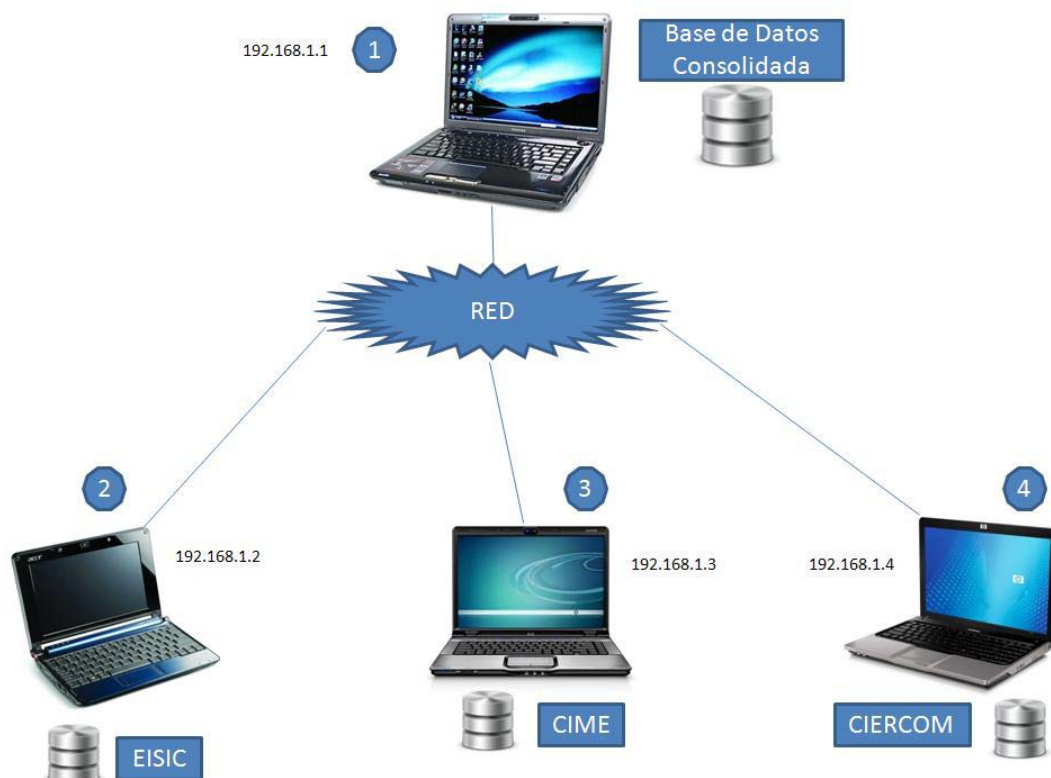
```
CREATE VIEW MATERIAS AS select * from MATERIAS@LINKHILARY WHERE ID_ESCUELA=1;
```

Para la eliminación de una vista se utiliza la instrucción:

```
DROP VIEW vista [restrict | cascade];
```



EJEMPLO DE DBLINKS Y VISTAS EN ORACLE



8. Ejemplo de Dblink Y Vistas en Oracle

ORDENADOR **1**

Aquí se encuentra la Base de Datos Consolidada en la cual tendremos las diferentes particiones.

Creamos el esquema de nuestra base con su respectivo tablespace, la sentencia es la siguiente:

```
CREATE TABLESPACE TACADEMICO DATAFILE 'C:\oracle\academico\TACADEMICO.DBF' SIZE 300M;
```

```
CREATE USER academico IDENTIFIED BY academico DEFAULT TABLESPACE TACADEMICO;
```

```
GRANT dba, connect, resource TO academico;
```

```
GRANT CREATE ANY VIEW TO academico WITH ADMIN OPTION;
```



No debemos olvidar la creación de los diferentes tablespace en los que serán almacenados las diferentes particiones. Las sentencias son las siguientes:

```
CREATE TABLESPACE TEISIC DATAFILE 'H:\oracle\eisic\TEISIC.DBF' SIZE 300M;
CREATE TABLESPACE TEITEX DATAFILE 'H:\oracle\eitex\TEITEX.DBF' SIZE 300M;
CREATE TABLESPACE TCIME DATAFILE 'H:\oracle\cime\TCIME.DBF' SIZE 300M;
CREATE TABLESPACE TCIERCOM DATAFILE 'H:\oracle\ciercom\TCIERCOM.DBF' SIZE 300M;
CREATE TABLESPACE TESDYM DATAFILE 'H:\oracle\esdym\TESDYM.DBF' SIZE 300M;
CREATE TABLESPACE TESIIN DATAFILE 'H:\oracle\esiin\TESIIN.DBF' SIZE 300M;
CREATE TABLESPACE TDOCENTE1 DATAFILE 'H:\oracle\docente1\TDOCENTE1.DBF' SIZE 300M;
CREATE TABLESPACE TDOCENTE2 DATAFILE 'H:\oracle\docente2\TDOCENTE2.DBF' SIZE 300M;
CREATE TABLESPACE TDOCENTE3 DATAFILE 'H:\oracle\docente3\TDOCENTE3.DBF' SIZE 300M;
CREATE TABLESPACE TDOCENTE4 DATAFILE 'H:\oracle\docente4\TDOCENTE4.DBF' SIZE 300M;
```

Una vez creado el esquema, podemos ingresar a nuestra base de datos para crear las tablas necesarias, recordemos que el nombre es academico y su contraseña academico.

Ahora vamos a crear las diferentes tablas, tablas particionadas y paquetes que necesitamos para el funcionamiento de nuestra base de datos. El scrip es el siguiente, como es un ejemplo no hemos incluido todas las tablas :

```
/*=====*/
/* Table: PARTITION DOCENTES */
/*=====*/
create table DOCENTES (
  CEDULA_DOCENTE VARCHAR2(12) not null,
  NOMBRES VARCHAR2(40),
  APELLIDOS VARCHAR2(40),
  TELEFONO VARCHAR2(15),
  CELULAR VARCHAR2(15),
  ESTADO_CIVIL CHAR(1),
  DIRECCION VARCHAR2(80),
  MAIL VARCHAR2(80),
  ESCALAFON INTEGER,
  USUARIO VARCHAR2(20),
  CLAVE VARCHAR2(20),
  ID_TITULOS INTEGER,
  constraint PK_DOCENTES primary key (CEDULA_DOCENTE)
)PARTITION by hash (cedula_docente)
(PARTITION DOCENT1 TABLESPACE tdocente1, PARTITION DOCENT2 TABLESPACE tdocente2,
PARTITION DOCENT3 TABLESPACE tdocente3, PARTITION DOCENT4 TABLESPACE
tdocente4);

/
/*=====*/
/* Table: PARTITION ESCUELAS */
/*=====*/
create table ESCUELAS (
  ID_ESCUELA INTEGER not null,
```



```

ID_FACULTAD      INTEGER,
ESCUELA          VARCHAR2(100),
SIGLAS           VARCHAR2(10),
constraint PK_ESCUELAS primary key (ID_ESCUELA)
)PARTITION BY RANGE (id_escuela)

```

```

(PARTITION EISIC VALUES LESS THAN (2) TABLESPACE TEISIC,
PARTITION EITEX VALUES LESS THAN (3) TABLESPACE TEITEX,
PARTITION CIME VALUES LESS THAN (4) TABLESPACE TCIME,
PARTITION CIERCOM VALUES LESS THAN (5) TABLESPACE TCIERCOM,
PARTITION ESDYM VALUES LESS THAN (6) TABLESPACE TESDYM,
PARTITION ESIIN VALUES LESS THAN (7) TABLESPACE TESIIN );

```

```

/
/*=====*/
/* Table: PARTITION ESTUDIANTES */
/*=====*/

```

```

create table ESTUDIANTES (
CEDULA          VARCHAR2(10)          not null,
NOMBRES         VARCHAR2(40),
APELLIDOS       VARCHAR2(40),
SEXO            CHAR(1),
FECHA_NACIMIENTO  TIMESTAMP,
NOTA_GRADO      NUMERIC(4, 2),
ESPECIALIDAD    VARCHAR2(50),
TIPO_COLEGIO    CHAR(1),
EXTRANJERO      CHAR(1),
ESTADO_CIVIL    CHAR(1),
DOMICILIO       VARCHAR2(50),
COLEGIO         VARCHAR2(70),
TELEFONO        VARCHAR2(15),
MAIL            VARCHAR2(100),
CLAVE           VARCHAR2(20),
ESTADO          VARCHAR2(2),
ID_ESCUELA      INTEGER,
constraint PK_ESTUDIANTES primary key (CEDULA)
)

```

```

PARTITION BY RANGE (id_escuela)
SUBPARTITION by hash (SEXO)
SUBPARTITION TEMPLATE
(SUBPARTITION MASCULINO,
SUBPARTITION FEMENINO)

```

```

(PARTITION EISIC VALUES LESS THAN (2) TABLESPACE TEISIC,
PARTITION EITEX VALUES LESS THAN (3) TABLESPACE TEITEX,
PARTITION CIME VALUES LESS THAN (4) TABLESPACE TCIME,
PARTITION CIERCOM VALUES LESS THAN (5) TABLESPACE TCIERCOM,
PARTITION ESDYM VALUES LESS THAN (6) TABLESPACE TESDYM,
PARTITION ESIIN VALUES LESS THAN (7) TABLESPACE TESIIN );

```

```

/
/*=====*/
/* Table: FACULTADES */
/*=====*/

```

```

create table FACULTADES (
ID_FACULTAD     INTEGER          not null,
FACULTAD        VARCHAR2(100),
SIGLAS          VARCHAR2(10),
constraint PK_FACULTADES primary key (ID_FACULTAD)
)

```

Una vez creadas nuestras tablas procedemos a configurar el nombre del servicio para poder luego acceder desde nuestras bases remotas mediante el dblink.



Recordemos que para configurar el nombre del servicio debemos cambiar el archivo tnsnames.ora que se encuentra en la siguiente dirección, dependiendo de cada ordenador de la siguiente manera:

```
D:\oracle\product\10.2.0\db_1\NETWORK\ADMIN \tnsnames.ora
ORCLHP =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.1.2)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = ORCL)
    )
  )
```

Ahora creamos las vistas que deseamos que nuestros diferentes nodos tengan, en las siguientes sentencias estamos creando las vistas de cada partición según cada carrera (ID_ESCUELA=1).

VISTA DE ESTUDIANTES

```
Create view ESTUDIANTES_EISIC AS SELECT *FROM ESTUDIANTES WHERE ID_ESCUELA=1;
Create view ESTUDIANTES_EITEX AS SELECT *FROM ESTUDIANTES WHERE ID_ESCUELA=2;
Create view ESTUDIANTES_CIME AS SELECT *FROM ESTUDIANTES WHERE ID_ESCUELA=3;
Create view ESTUDIANTES_CIERCOM AS SELECT *FROM ESTUDIANTES WHERE ID_ESCUELA=4;
Create view ESTUDIANTES_ESDYM AS SELECT *FROM ESTUDIANTES WHERE ID_ESCUELA=5;
Create view ESTUDIANTES_ESIIN AS SELECT *FROM ESTUDIANTES WHERE ID_ESCUELA=6;
```

VISTA DE MATERIAS

```
Create view MATERIAS_EISIC AS SELECT *FROM MATERIAS WHERE ID_ESCUELA=1;
Create view MATERIAS_EITEX AS SELECT *FROM MATERIAS WHERE ID_ESCUELA=2;
Create view MATERIAS_CIME AS SELECT *FROM MATERIAS WHERE ID_ESCUELA=3;
Create view MATERIAS_CIERCOM AS SELECT *FROM MATERIAS WHERE ID_ESCUELA=4;
Create view MATERIAS_ESDYM AS SELECT *FROM MATERIAS WHERE ID_ESCUELA=5;
Create view MATERIAS_ESIIN AS SELECT *FROM MATERIAS WHERE ID_ESCUELA=6;
```

ORDENADOR



Aquí se encuentra la Base de Datos Remota en la cual tendremos la partición de esta carrera como es EISIC

Antes de nada debemos configurar el nombre del servicio para poder luego acceder desde nuestra base remota mediante el dblink.



Recordemos que para configurar el nombre del servicio debemos cambiar el archivo tnsnames.ora que se encuentra en la siguiente dirección, dependiendo de cada ordenador de la siguiente manera:

```
F:\oracle\product\10.2.0\db_1\NETWORK\admin \tnsnames.ora
```

```
DBHILARY =  
(DESCRIPTION =  
(ADDRESS_LIST =  
(ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.1.1)(PORT = 1521))  
)  
(CONNECT_DATA =  
(SERVICE_NAME = ORCL)  
)  
)
```

Ahora procedemos a creamos el esquema de nuestra base con su respectivo tablespace, la sentencia es la siguiente:

```
CREATE TABLESPACE TACADEMICO DATAFILE 'H:\oracle\academico\TACADEMICO.DBF' SIZE  
300M;  
  
CREATE USER academico1 IDENTIFIED BY academico1 DEFAULT TABLESPACE TACADEMICO;  
  
GRANT dba, connect, resource TO academico1;  
  
GRANT CREATE ANY VIEW TO academico1 WITH ADMIN OPTION;
```

Una vez creado el esquema, podemos ingresar a nuestra base de datos para crear las diferentes vistas de las tablas necesarias, recordemos que el nombre es academico1 y su contraseña academico1.

La manera más sencilla de acceder a tablas y vistas (*views*) de otra base de datos Oracle es creando un DBLink, en la siguiente sentencia estamos accediendo al ordenador 1, al esquema que tiene por nombre académico y su contraseña es académico, para poder acceder a las vistas creadas en ese ordenador.

```
create database link LINKHILARY connect to academico identified by academico using 'DBHILARY';
```






Recordemos que las tablas y vistas remotas pueden usar consultas es decir una sentencia SELECT, como se muestra.

```
select * from MATERIAS@LINKHILARY WHERE ID_ESCUELA=1;  
select * from ESTUDIANTES@LINKHILARY WHERE ID_ESCUELA=1;
```

En esta sentencia estamos listando a las materias y estudiantes que tienen un id_escuela=1, esto es decir solo material y estudiantes de la carrera EISIC.

Ahora vamos a proceder a crear las vistas, las que serán como una tabla normal en el esquema academico1, pudiendo así realizar una inserción, borrado o edición. La sentencia es la siguiente:

```
CREATE VIEW ESTUDIANTES AS select * from ESTUDIANTES@LINKHILARY WHERE  
ID_ESCUELA=1;  
CREATE VIEW MATERIAS AS select * from MATERIAS@LINKHILARY WHERE ID_ESCUELA=1;
```

En el ordenador  y  se realizan los mismos paso que en el ordenador  teniendo en cuenta el nombre del servicio que será distinto según el ordenador y al momento de crear las vistas debemos tener claro que id_escuela de escuela le corresponde.

3.7. REPLICACIÓN DE BASE DE DATOS EN ORACLE

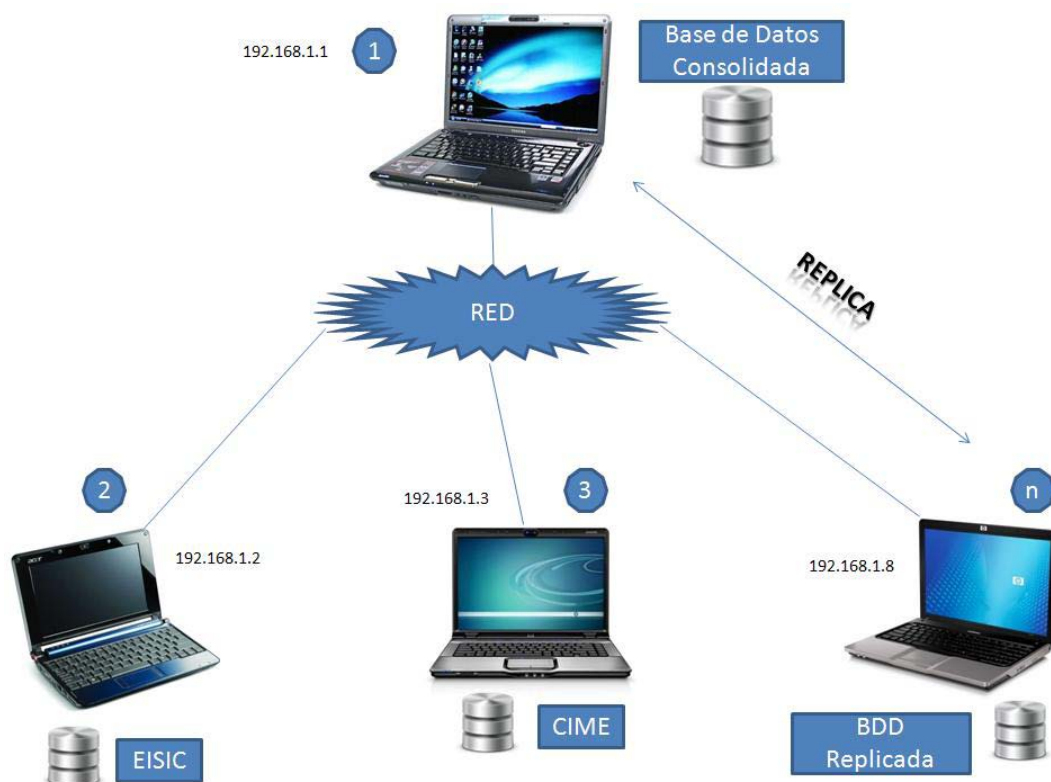
Es el proceso de copiar y mantener objetos de bases de datos como tablas que constituyen una base de datos distribuida.

Los cambios aplicados en un sitio son almacenados localmente para posteriormente ser enviados y aplicados al sitio remoto.



En una base de datos distribuida, existen datos disponibles en muchos lugares, pero un objeto en particular (una tabla) solo existe en un nodo de la BD.

En la replicación sincrónica los cambios que ocurran en un master site son replicados inmediatamente a los otros participantes del sistema. Si una transacción no puede ser procesada por un master site, se producirá un rollback de la transacción en todos los otros master sites.



9. Ejemplo de Replica en Oracle

A continuación se muestra la forma de replicar de manera sencilla los datos de una base de datos en oracle hacia otro servidor oracle, mediante triggers.



La replicación permite tener una copia exacta de una base de datos alojada en un servidor (maestro) que se guardará en otro servidor (esclavo). Todas las modificaciones que se hagan en la base de datos del servidor maestro se actualizarán inmediatamente en el servidor esclavo.

Esto no es una copia de seguridad, ya que si borramos una fila en la base de datos maestra, también se borrará en la base de datos esclava.

Estos son los pasos para realizar una réplica.

Recordemos que previamente debe estar configurado del servicio en el archivo tnsnames.ora para poder realizar un dblink

- Una vez creada nuestra base de datos (esquema) en el servidor (maestro) procedemos a crear cada uno de los triggers, esto es INSERT, DELETE, UPDATE para que cuando hagamos una inserción, borrado o edición se refleje inmediatamente en nuestro servidor (esclavo). La sentencia es la siguiente

INSERT TRIGGER

```
CREATE OR REPLACE TRIGGER INSERT_MATERIAS
AFTER INSERT ON MATERIAS
FOR EACH ROW
BEGIN
    INSERT INTO MATERIAS@LINKHP
VALUES(:new.id_materia,:new.materia,:new.creditos,:new.horas_programadas,:new.fecha_creacion,
:new.estado,:new.id_escuela,:new.materia_anterior,:new.id_nivel);
    NULL;
END;
```

UPDATE TRIGGER

```
CREATE OR REPLACE TRIGGER UPDATE_MATERIAS
AFTER UPDATE ON MATERIAS
FOR EACH ROW
BEGIN
```



```
UPDATE MATERIAS@LINKHP SET
id_materia=:new.id_materia,materia=:new.materia,creditos=:new.creditos,horas_programadas=:new.
horas_programadas,fecha_creacion=:new.fecha_creacion,

estado=:new.estado,id_escuela=:new.id_escuela,materia_anterior=:new.materia_anterior,id_nivel=:ne
w.id_nivel
WHERE id_materia=:new.id_materia;
NULL;
END;
```

DELETE TRIGGER

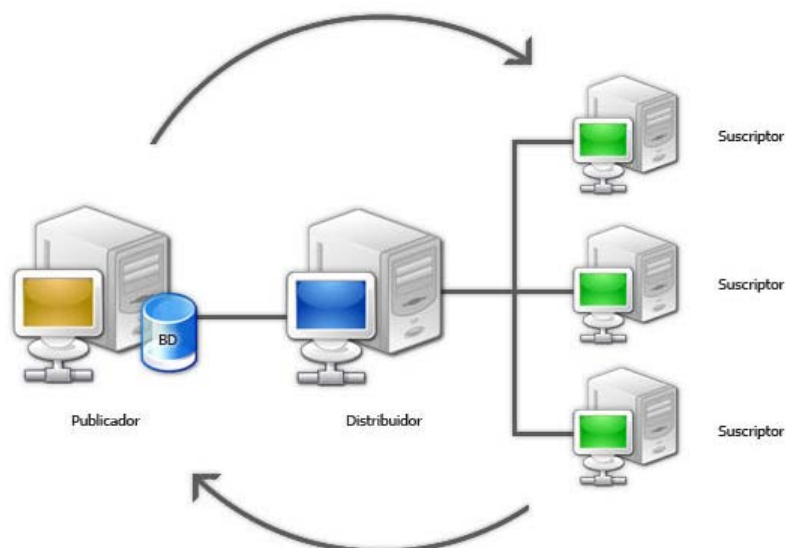
```
CREATE OR REPLACE TRIGGER DELETE_MATERIAS
AFTER DELETE ON MATERIAS
FOR EACH ROW
BEGIN
DELETE MATERIAS@LINKHP WHERE id_materia=:old.id_materia;
NULL;
END;
```

- Ahora de la misma manera debemos crear en nuestro servidor (esclavo) los triggers para que cuando realizamos cualquier cambio aquí también ser reflejen en nuestro servidor (maestro).



4

ARQUITECTURA DE DISTRIBUCIÓN DE DATOS



Contenido:

Arquitectura de Distribución de Datos

- 4.1. Transparencia
 - 4.1.1 Transparencia de Localización
 - 4.1.2 Transparencia de Fragmentación
 - 4.1.3 Transparencia de Replica
- 4.2. Formas de Replica.



CAPITULO IV

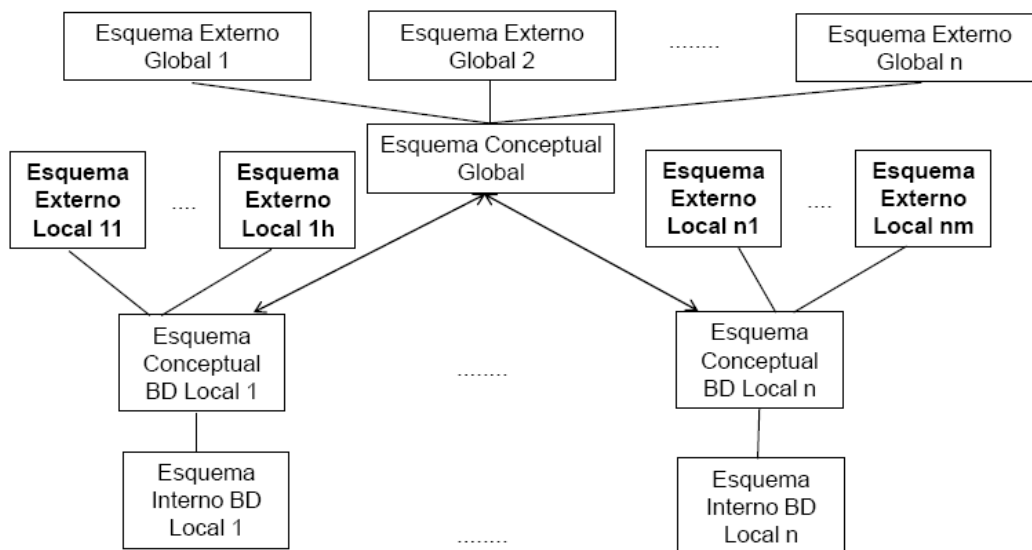
4. ARQUITECTURA DE DISTRIBUCIÓN DE DATOS

ARQUITECTURA DE UNA BASE DE DATOS DISTRIBUIDA

Se debe tomar en cuenta varios factores para la definición de la arquitectura de un sistema:

- 1. Distribución:** Los componentes del sistema están localizados en la misma computadora o en diferente computador.
- 2. Heterogeneidad:** Es cuando existen en él componentes que se ejecutan en diversos sistemas operativos.
- 3. Autonomía:** Se puede presentar en diferentes niveles, como son:
 - **Autonomía de diseño:** Está relacionadas a su propio diseño.
 - **Autonomía de comunicación:** Es cómo y cuándo comunicarse con otros SMBD.
 - **Autonomía de ejecución:** Ejecutar operaciones locales como quiera.

ARQUITECTURA ANSI/SPARC



10. Arquitectura ANSI/X3/SPARC para BDD [IMAG.10]¹¹

La arquitectura ANSI / SPARC se divide en 3 niveles:

1. **EL NIVEL INTERNO.** Es el que se ocupa de la forma como se almacenan físicamente los datos.
2. **EL NIVEL EXTERNO.** Es el que se ocupa de la forma como los usuarios individuales perciben los datos.
3. **EL NIVEL CONCEPTUAL.** Es un nivel de mediación entre los otros dos, es decir define las estructuras de almacenamientos el Administrador de Base de Datos.

No existe un equivalente de una arquitectura estándar para sistemas de manejo de bases de datos distribuidas, cada sistema ha adoptado su propia arquitectura.

¹¹ http://www.kybele.etsii.urjc.es/docencia/DBD/2009-2010/Material/%5BDBD-2009-10%5DBD_Distribuidas.pdf



Se debe definir un modelo de referencia para un esquema de estandarización en bases de datos distribuidas, cuyo propósito es dividir el trabajo en piezas y esas piezas se relacionan unas con otras. Se sigue los siguientes enfoques:

- 1. Basado en componentes.** Se definen las componentes del sistema junto con las relaciones entre ellas.
- 2. Basado en funciones.** Se identifican las diferentes clases de usuarios junto con la funcionalidad que el sistema ofrecerá para cada clase.
- 3. Basado en datos.** Se identifican los diferentes tipos de descripción de datos y se especifica un marco de trabajo arquitectural el cual define las unidades funcionales que realizarán y/o usarán los datos de acuerdo con las diferentes vistas. Este es el enfoque seguido por el modelo ANSI/SPARC.

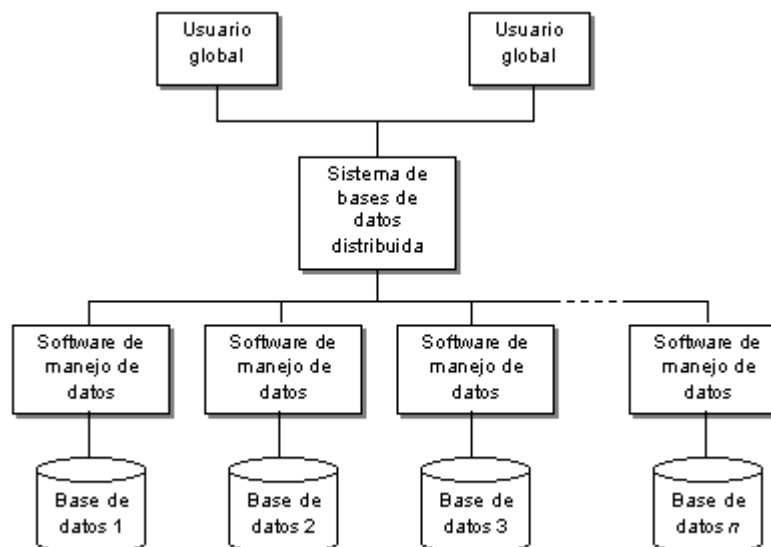
ARQUITECTURA DE UN SISTEMA MANEJADOR DE BASES DE DATOS DISTRIBUIDAS

Los sistemas de datos distribuidos están divididos en dos clases:

1. Sistemas de manejo de bases de datos distribuidos homogéneos
2. Sistemas de manejo de bases de datos distribuidos heterogéneos



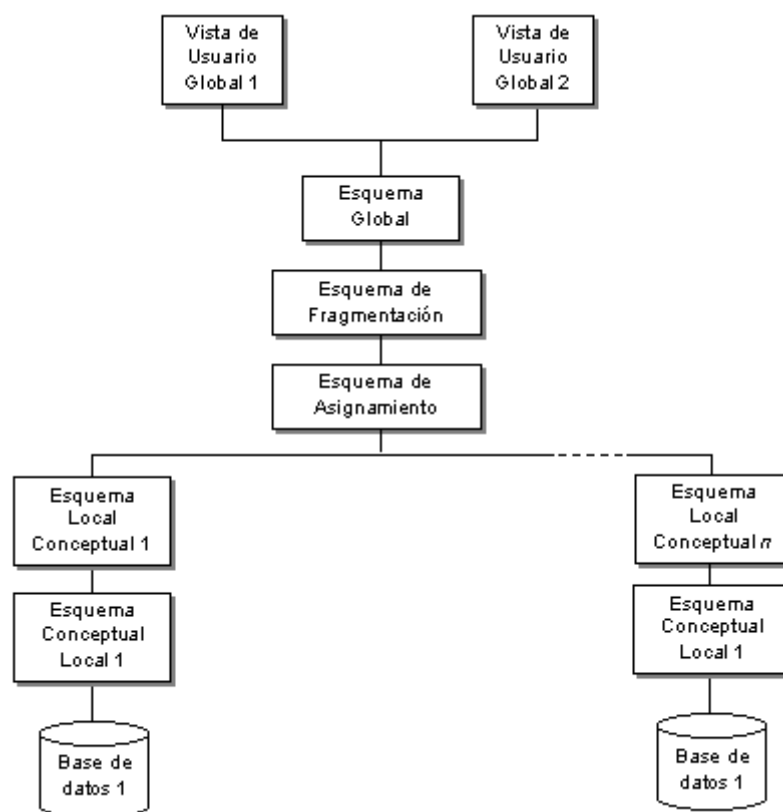
ARQUITECTURA DE UN SISTEMA DE MANEJO DE BASES DE DATOS DISTRIBUIDOS HOMOGÉNEOS



11. Arquitectura de un SMBDD homogéneo.

Los sistemas homogéneos se parece a un sistema centralizado, a diferencia que estos sus datos se distribuyen en varios sitios comunicados por la red. No existen usuarios locales y todos ellos accesan la base de datos a través de una interfaz global.

Para manejar los aspectos de la distribución, se deben agregar dos niveles a la arquitectura estándar ANSI-SPARC, de la siguiente manera, como se muestra en la Figura



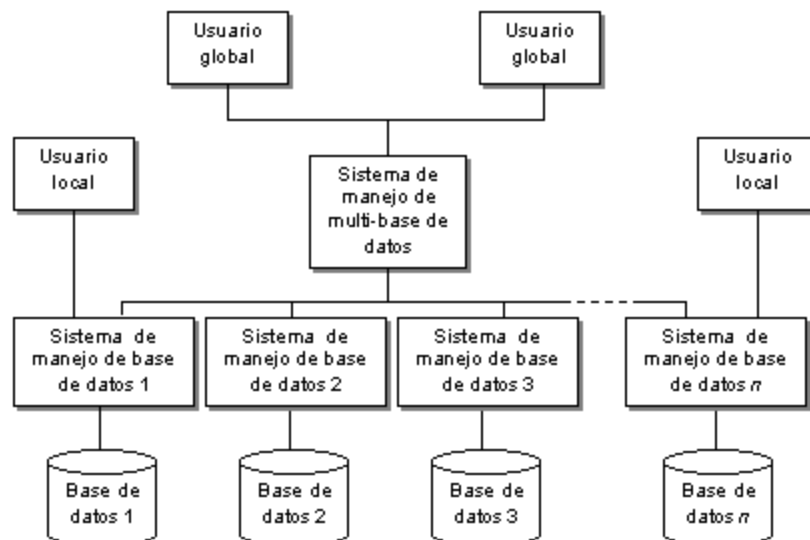
12. Arquitectura de los esquemas de un SMBDD homogéneo

El esquema de fragmentación describe la forma en que las relaciones globales se dividen entre las bases de datos locales.

El esquema de asignamiento especifica el lugar en el cual cada fragmento es almacenado. De aquí, los fragmentos pueden migrar de un sitio a otro en respuesta a cambios en los patrones de acceso.



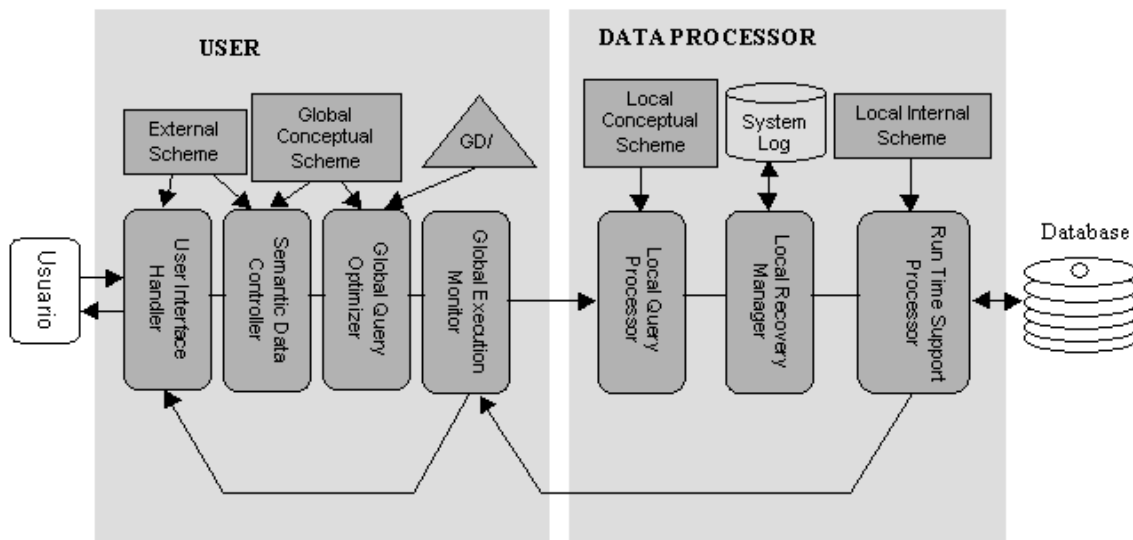
ARQUITECTURA DE UN SISTEMA DE MANEJO DE BASES DE DATOS DISTRIBUIDOS HETEROGÉNEOS



13. Arquitectura de un sistema multi-bases de datos.

Un sistema multi-bases de datos tiene múltiples SMBDs, que pueden ser de tipos diferentes, y múltiples bases de datos existentes. Existen usuarios locales y globales.

ARQUITECTURA BASADA EN COMPONENTES DE UN SISTEMA DE MANEJO DE BASES DE DATOS DISTRIBUIDOS



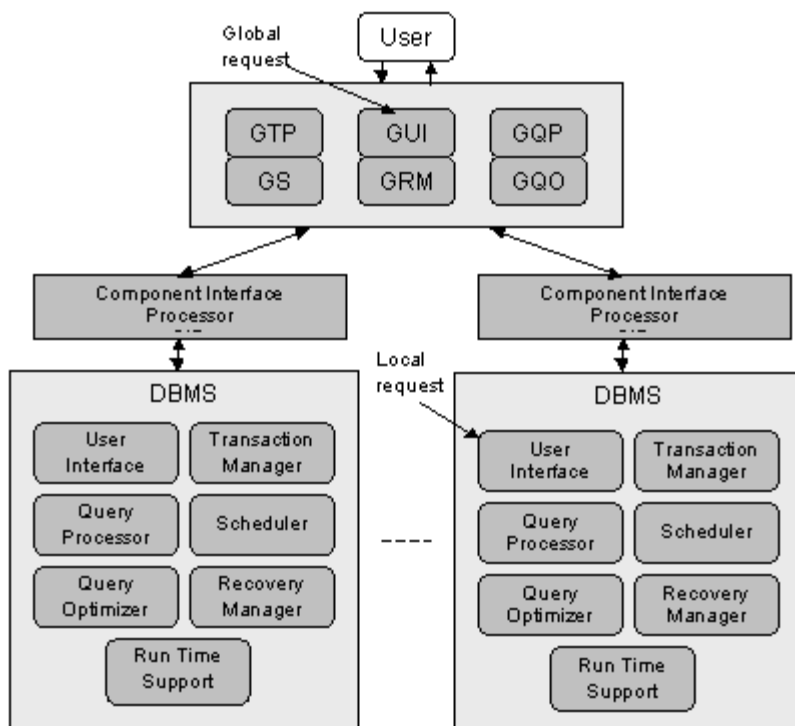
14. Arquitectura basada en componentes de un SDBM distribuido

Consiste en dos partes como son: el procesador de datos y el procesador de usuario.

- El procesador de usuario es el encargado de procesar las solicitudes del usuario, consiste en cuatro partes: un manejador de la interfaz con el usuario, un controlador semántico de datos, un optimizador global de consultas y un supervisor de la ejecución global.
- El procesador de datos existe en cada nodo de la base de datos distribuida. Utiliza un esquema local conceptual y un esquema local interno, el procesador de datos consiste de tres partes: un procesador de consultas locales, un

manejador de recuperación de fallas locales y un procesador de soporte para tiempo de ejecución.

ARQUITECTURA BASADA EN COMPONENTES DE UN SISTEMA MULTI-BASES DE DATOS.



15. Arquitectura basada en componentes sistema multi-bases de datos.

Consta de un sistema de manejo de bases datos para usuarios globales y usuarios locales. Para comunicar el sistema global con los sistemas locales se define una interfaz común entre componentes mediante la cual, las operaciones globales se convierten en una o varias acciones locales.



4.1. TRANSPARENCIA

Se refiere a la división del nivel semántico y la implementación del sistema.

El objetivo es ocultar al usuario los detalles de diseño, es decir, el usuario no tiene que saber que se encuentra trabajando sobre un sistema distribuido, por ejemplo la independencia de los datos es una forma de transparencia

El propósito fundamental de la transparencia en el ambiente distribuido es la independencia de datos.

La transparencia la podemos encontrar en:

- ✓ Manejo de la red de comunicación.
- ✓ Manejo de copias repetidas
- ✓ En la distribución o fragmentación de la información.

INDEPENDENCIA DE DATOS

Es la inmunidad de las aplicaciones de usuarios a los cambios en la definición y organización de los datos y viceversa. La independencia de datos se puede darse en dos aspectos: lógico y físico. [www.08]¹²

1. **Independencia lógica de datos.** Se refiere a la inmunidad de las aplicaciones de usuario a los cambios en la estructura lógica de la base de datos. Esto permite que un cambio en la

¹² <http://basedatosavanzadaselem.obolog.com/unidad-1-base-datos-distribuidas-523685>



definición de un esquema no debe afectar a las aplicaciones de usuario. Por ejemplo, el agregar un nuevo atributo a una relación, la creación de una nueva relación, el reordenamiento lógico de algunos atributos. [www.09]¹³

2. Independencia física de datos. Se refiere al ocultamiento de los detalles sobre las estructuras de almacenamiento a las aplicaciones de usuario. Esto es, la descripción física de datos puede cambiar sin afectar a las aplicaciones de usuario. Por ejemplo, los datos pueden ser movidos de un disco a otro, o la organización de los datos puede cambiar. [www.10]¹⁴

NIVELES DE TRANSPARENCIA.

El propósito de establecer una arquitectura de un sistema de bases de datos distribuidas es ofrecer un nivel de transparencia adecuado para el manejo de la información.

Primer Nivel. Se soporta la transparencia de red.

Segundo Nivel. Se permite la transparencia de replicación de datos.

Tercer Nivel. Se permite la transparencia de la fragmentación.

Cuarto Nivel. Se permite transparencia de acceso (por medio de un lenguaje de manipulación de datos)

¹³ <http://basedatosavanzadaselem.obolog.com/unidad-1-base-datos-distribuidas-523685>

¹⁴ <http://basedatosavanzadaselem.obolog.com/unidad-1-base-datos-distribuidas-523685>



16. Organización en Capas de niveles de Transparencia. [IMAG.16]¹⁵

4.1.1. TRANSPARENCIA DE LOCALIZACIÓN.

La transparencia de localización le permite acceder al usuario a los datos sin tener en cuenta la ubicación de estos, es decir debe ser transparente al usuario, ya que este no necesita saber dónde está el dato para utilizarlo. Se consigue cuando los administradores de transacciones distribuidas pueden determinar la localización de los datos y emitir acciones a los administradores apropiados, esto se puede ejecutar cuando los administradores de transacciones tienen acceso a los directorios de localizaciones de los datos.

Los administradores de transacciones necesitan conocer si los datos cambian de lugar, ya que las transacciones ignoran la modificación en la localización.

¹⁵ http://usuarios.multimania.es/admin2master/documentos/descargas/ppt/BASES_DE_DATOS_DISTRIBUIDAS.PPT

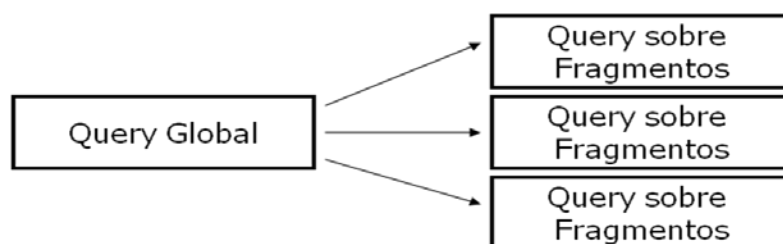
4.1.2. TRANSPARENCIA DE FRAGMENTACIÓN

El acceso a una base de datos distribuida debe hacerse en forma transparente. Los usuarios deben comportarse, como si los datos en realidad no estuvieran fragmentados, lo cual es necesario por razones de rendimiento.

El programador no necesita saber si la base de datos es distribuida o está fragmentada.

Los accesos a la base de datos es de forma global, es decir el usuario no necesita especificar los nombres de los fragmentos ni la ubicaciones de los datos.

El sistema maneja la conversión de consultas de usuario definidas sobre relaciones globales a consultas definidas sobre fragmentos. Las respuestas a consultas fragmentadas para obtener una sola respuesta a una consulta global. [www.11]¹⁶



17. Muestra la Transparencia de Fragmentación [IMAG.17]¹⁷

¹⁶ <http://www.google.com.ec/url?sa=t&source=web&cd=4&ved=0CB8QFjAD&url=http%3A%2F%2Fdircompucv.ciens.ucv.ve%2Fgenerador%2Fsites%2Fbases-de-datos-dist%2Farchivos%2FBDD%2520-2520Clase%2520.pps&rct=j&q=%BFQue+es+independencia+de+datos%3F++Es+la+inmunidad+de+las+aplicaciones+de+usuarios+a+los+cambios+en+la+definici%F3n+y+organizaci%F3n+de+los+datos+y+viceversa.&ei=00YmTKXaI8KAlAeq7P21Aw&usg=AFQjCNH8JW6hh0LX5fMedB9AwofatAvO-A>

¹⁷ <http://www.google.com.ec/url?sa=t&source=web&cd=4&ved=0CB8QFjAD&url=http%3A%2F%2Fdircompucv.ciens.ucv.ve%2Fgenerador%2Fsites%2Fbases-de-datos-dist%2Farchivos%2FBDD%2520-2520Clase%2520.pps&rct=j&q=%BFQue+es+independencia+de+datos%3F++Es+la+inmunidad+de+las+aplicaciones+de+usuarios+a+los+cambios+en+la+definici%F3n+y+organizaci%F3n+de+los+datos+y+viceversa.&ei=00YmTKXaI8KAlAeq7P21Aw&usg=AFQjCNH8JW6hh0LX5fMedB9AwofatAvO-A>



4.1.3. TRANSPARENCIA DE REPLICA

La transparencia sobre replicación de datos se refiere a que si existen réplicas de objetos de la base de datos, esta debe ser controlada por el SMBDD, más no por el usuario.

El usuario no necesita saber sobre la replicación de los datos, la función principal de la transparencia de replicación es la de mantener la consistencia entre las copias, esta funciona en forma transparente a las aplicaciones.

La replicación puede mejorar el funcionamiento y proteger la disponibilidad de las aplicaciones, porque alterna opciones de acceso de los datos existente.

Existe una copia principal y varias copias secundarias, las que se extienden a lo largo de las modificaciones en forma asíncrona.

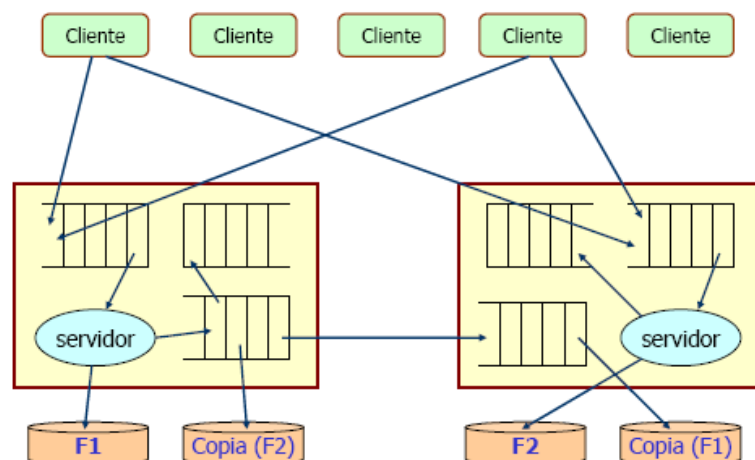
Existen dos tipos de propagación de modificaciones: [www.11]¹⁸

- ✓ Incremental: La información que se envía desde la copia principal a las secundarias son las variaciones en los datos.
- ✓ Total: Se envía toda la copia principal completa.

Su principal utilidad es hacer que el sistema sea menos sensible a los fallos, ya que si la copia principal no está disponible se puede seguir usando alguna de las copias secundarias.

de+usuarios+a+los+cambios+en+la+definici%F3n+y+organizaci%F3n+de+los+datos+y+viceversa.&ei=00YmTKXaI8KA
lAeq7P2lAw&usg=AFQjCNH8JW6hh0LX5fMedB9AwofatAvO-A

¹⁸ <http://alarcos.inf-cr.uclm.es/doc/aplicabdd/Documentos/teoria/arquitecturas%20para%20bases%20de%20datos.pdf>



18. Principal utilidad de la Transparencia de Replicación [IMAG.18]¹⁹

La replicación es necesaria por las siguientes razones:

- ✓ Mayor rendimiento, debido a que se dispone de copias locales.
- ✓ Mayor disponibilidad, ya que los datos son accesibles siempre al tenerse varias copias.

La principal desventaja, es que hay que mantener actualizadas todas las copias de ese objeto o dato replicado. Esto nos lleva al problema de la “propagación de las actualizaciones”.

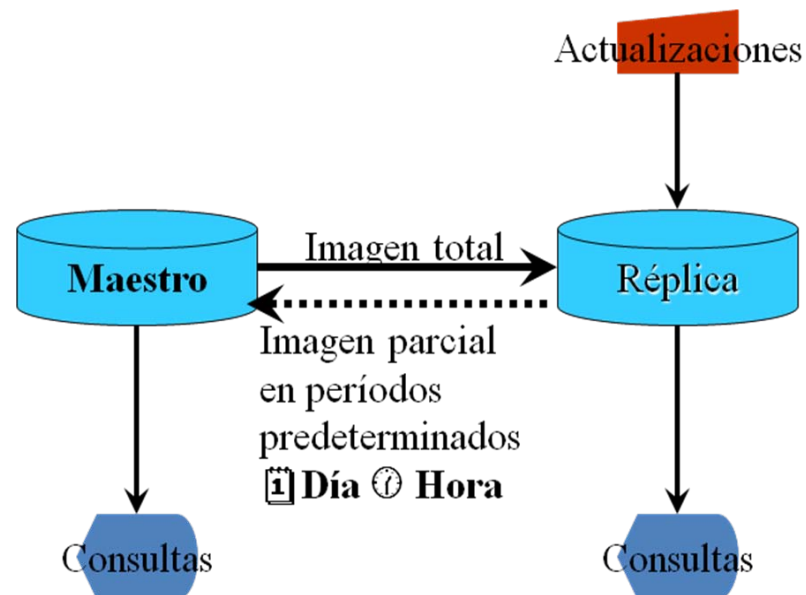
¹⁹ <http://alarcos.inf-cr.uclm.es/doc/aplicabdd/Documentos/teoria/arquitecturas%20para%20bases%20de%20datos.pdf>

4.2. FORMAS DE RÉPLICAS

Las réplicas pueden asumir los siguientes formatos:

- ✓ Periódica
- ✓ Continua
- ✓ Check-in / Check-out

RÉPLICA PERIÓDICA

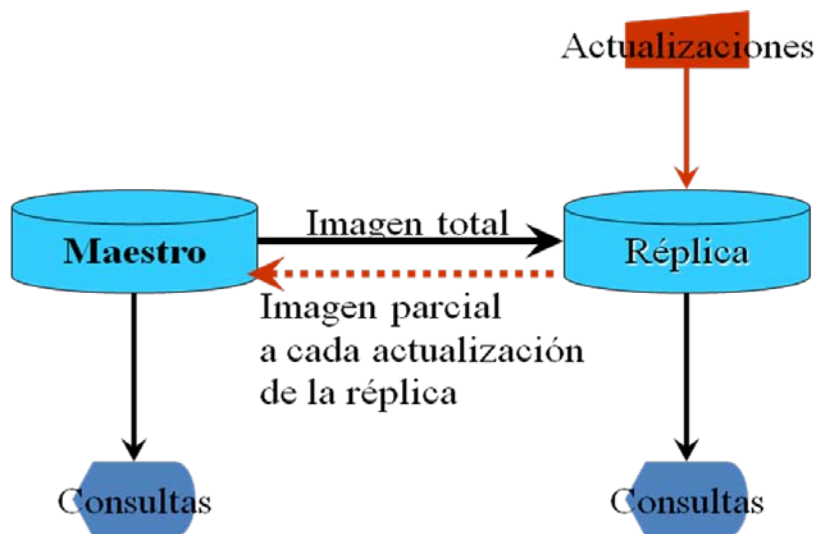


19. Replica Periódica [IMAG.19]²⁰

²⁰ http://www.cyta.com.ar/elearn/bd/curso_archivos/bddistribuida.ppt

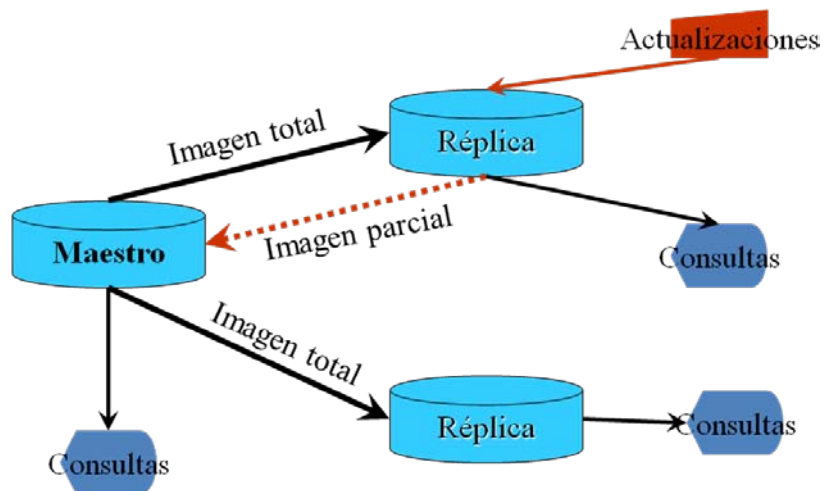


RÉPLICA CONTINUA



20. Replica Continua [IMAG.20]²¹

RÉPLICA CHECK - IN/OUT



21. Replica Check - IN/OUT [IMAG.21]²²

²¹ http://www.cyta.com.ar/elearn/bd/curso_archivos/bddistribuida.ppt

²² http://www.cyta.com.ar/elearn/bd/curso_archivos/bddistribuida.ppt



CREACIÓN DE RÉPLICAS

Es posible crear réplicas completas o parciales:

Réplica completa: Contiene todos los documentos y elementos de diseño de la base de datos como son: formularios y agentes.

Réplica parcial: Contiene documentos abreviados, resúmenes de documentos o documentos cuyos campos contienen información especificada, documentos preseleccionados, documentos de carpetas o vistas especificadas.

Este tipo de réplica le resultará útil cuando sólo necesite una parte de la base de datos y quiere ahorrar espacio en el disco.

NOTA: Las réplicas que no contienen documentos se denominan réplicas vacías. Si intenta abrir una réplica vacía, se mostrará un mensaje diciendo que la base de datos no está inicializada y debe replicarse.

CARACTERÍSTICAS

VENTAJAS

- ✓ Disponibilidad frente a fallos de la red.
- ✓ Paralelismo, las peticiones se pueden procesar en varios nodos en paralelo.
- ✓ Transferencia de datos reducida



INCONVENIENTES

- ✓ Se eleva el coste de la actualizaciones
- ✓ Se complica el control de la concurrencia



5

ADMINISTRACIÓN DE BASES DE DATOS DISTRIBUIDAS



Contenido:

Administración de Bases de Datos Distribuidas

- 5.1 Conceptos Generales
 - 5.1.1 Administración la Estructura de la Base de Datos
 - 5.1.2 Administración de la Actividad de Datos
 - 5.1.3 Administrar el Sistema Manejador de la Base de Datos
 - 5.1.4 Confirmar la Seguridad de la Base de Datos
- 5.2 Objetivos del Administrador de la Base de Datos
 - 5.2.1 Integridad de Datos
 - 5.2.2 Seguridad y Disponibilidad de Datos
- 5.3 Funciones del Administrador.
 - 5.3.1 Creación de Usuarios
 - 5.3.2 Creación de Bases de Datos y Tablas
 - 5.3.3 Restricciones de Integridad
 - 5.3.4 Respaldos y recuperaciones



CAPITULO V

5. ADMINISTRADOR DE LA BASE DE DATOS

5.1. CONCEPTOS GENERALES.

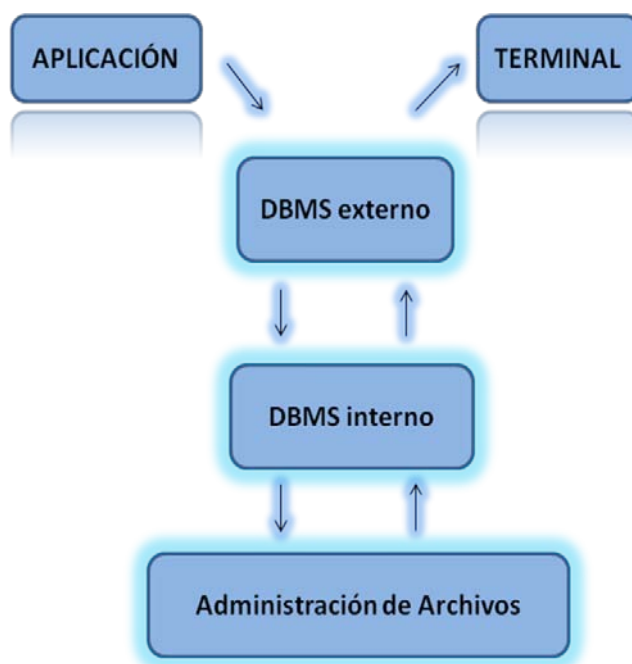
La información es lo más valioso que se tiene, puesto que es indispensable contar con una persona, y esta sería el Administrador de Base de Datos (DBA). La labor del Administrador de Datos es: decidir qué datos deben almacenarse en la base de datos, y establecer políticas para mantener y manejar los datos si ya fueron almacenados.

El Administrador de Bases de Datos es quien posee los conocimientos sobre el lenguaje estructurado de consultas (SQL). Además posee las habilidades y destrezas necesarias para la implementación, configuración y puesta a punto del motor de la base de datos. Es también quien desarrolla una metodología de análisis y evaluación de la estructura de Bases de Datos Relacionales. Entre sus responsabilidades se encuentra el planear y crear bases de datos, administrar el acceso, los recursos y estructuras (tanto físicas como lógicas) de las mismas. Por último también tiene la responsabilidad de administrar usuarios y sus privilegios.



La administración de bases de datos se realiza con un sistema llamado DBMS (Database management system) o (Sistema de administración de bases de datos). El DBMS es un conjunto de servicios como son las aplicaciones de software. Este permite:

- Un fácil acceso a los datos
- El acceso a la información por parte de múltiples usuarios
- La manipulación de los datos encontrados en la base de datos (insertar, eliminar, editar)



22. División del DBMS [IMAG.22]²³

²³ Ilustración propia



El DBMS puede dividirse en tres subsistemas:

- **El sistema de administración de archivos:** Para almacenar información en un medio físico
- **El DBMS interno:** Para ubicar la información en orden
- **El DBMS externo:** Representa la interfaz del usuario

El DBA es responsable de:

- Administrar la estructura de la Base de Datos.
- Administrar la actividad de los datos.
- Administrar el Sistema Manejador de Base de Datos.
 - ✓ Establecer el Diccionario de Datos.
 - ✓ Asegurar la confiabilidad de la Base de Datos.
- Confirmar la seguridad de la Base de Datos.

5.1.1. ADMINISTRAR LA ESTRUCTURA DE LA BASE DE DATOS.

El DBA debe participar en el diseño inicial, diseño general, puesta en marcha, control y administración de requerimientos de la base de datos incluyendo los DBMS (Sistema de Administración de Base de Datos)



El DBA participa en el desarrollo de procedimientos y control de calidad e integridad de la BD.

Al terminar el diseño de las bases de datos, se la pone en práctica la creación de los datos utilizando el DBMS. Todas las modificaciones en las estructuras o procedimientos de base de datos requieren de una cuidadosa administración.

5.1.2. ADMINISTRACIÓN DE LA ACTIVIDAD DE DATOS.

El DBA administra la actividad de datos; protege los datos, proporciona estándares, guías de acción, procedimientos de control y documentación para que los usuarios trabajen en forma conjunta al procesar los datos en la base de datos, puesto que el DBA no es usuario del sistema, no administra valores de datos; ni los procesa.

5.1.3. ADMINISTRAR EL SISTEMA MANEJADOR DE BASE DE DATOS.

El DBA debe asegurarse que los procesos de operación lleguen a quienes han solicitado, debido la concurrencia de varios usuarios.

La base de datos se basa en estándares esto es: tipo de dato, longitud, formato, como es procesada y presentada, como sé



accesa a un archivo, como se determinan los índices primarios y auxiliares, registros, etc.

La función del DBA es de revisar los estándares varias veces para ver su operatividad, ajustarlos, ampliarlos o cancelarlos y hacer que se cumplan

El DBA debe procurar que las aplicaciones beneficien también a los usuarios, privilegiando siempre la optimización en la operación del DBMS.

ESTABLECER EL DICCIONARIO DE DATOS.

Se debe registrar en el diccionario de datos, los estándares de la estructura de la base de datos previamente definidos. A la que todos los usuarios puedan acceder a dicha información, este debe poseer información que indique sus limitantes de seguridad y el tipo de dato que será utilizado.

ASEGURAR LA CONFIABILIDAD DE LA BASE DE DATOS

Se deben utilizar gestores para la reparación de posibles errores que pueden sufrir, como puede ser un corte de luz.

Debe ser capaz de recuperarse frente a errores o usos inadecuados.



5.1.4. CONFIRMAR LA SEGURIDAD DE LA BASE DE DATOS.

Organizar propuestas para realizar ajustes en los accesos a datos compartidos, en caso de haber problemas cuando dos o más usuarios quedan autorizados a acceder a los mismos datos. Uno de los conflictos es el la actualización perdida; es decir cuando el trabajo de un usuario queda sobrescrito por un segundo usuario.

El DBMS debe ser capaz de implementar las restricciones aplicables al acceso concurrente, se hace indispensable el apego a los estándares el seguimiento de instructivos, manuales y las reglas establecidas para los diversos procesamientos y procedimientos que se llevan a cabo.

Existen varias alternativas que el DBA debe utilizar para resolver los siguientes problemas:

- Restringir el acceso a los procedimientos para ciertos usuarios.
- Restringir al acceso a los datos para ciertos usuarios procedimientos y/o datos.
- Evitar la coincidencia de horarios para usuarios que comparten.



El DBA debe prever las fallas, definir procedimientos, estándares de operación; los usuarios deben saber que hacer cuando el sistema este caído y cuando el sistema este puesto en marcha nuevamente. La persona hecho cargo debe saber cómo iniciar el proceso de recuperación de la base de datos que copias de seguridad utilizar; como programar la re ejecución del tiempo perdido y de las tareas pendientes. El DBA debe encargarse del mantenimiento y de la documentación con relación a la actividad de los datos, teniendo en cuenta estándares de la base de datos, derechos de recuperación y de acceso a la base de datos, estándares para la recuperación de caídas del sistema.

5.2. OBJETIVOS DEL ADMINISTRADOR DE LA BASE DE DATOS.

Los objetivos del Administrador de la Base de Datos son los siguientes:

- Integridad de los Datos.
- Seguridad de los Datos.
- Disponibilidad de los Datos.



5.2.1. INTEGRIDAD DE LOS DATOS.

La integridad de los datos, es el encargado de asegurarse que las operaciones ejecutadas por los usuarios sean correctas y se mantenga la consistencia de la base de datos.

Una parte del objetivo es garantizar la integridad de los datos ya que se puede ocasionar inexactitudes en los datos.

La base de datos se debe protegerse de accidentes tales como: errores en la entrada de los datos o en la programación, el uso mal intencionado de la base de datos, de los fallos del hardware o del software que corrompen los datos, fallos durante el procesamiento de las transacciones, acceso concurrente en la base de datos.

5.2.2. SEGURIDAD DE DATOS

Esta se encarga de limitar o restringir a los usuarios a realizar solo las operaciones permitidas. Se denomina seguridad de los datos a la protección de la base de datos de uso mal intencionado o no autorizado.

5.2.3. DISPONIBILIDAD DE DATOS

Significa que los usuarios autorizados tengan acceso a los datos cuando lo necesiten, esto es administrar la actividad de la base de datos. La posibilidad de fallos de hardware o de software requiere



procedimientos de recuperación de la base de datos, se debe proporcionar medios para el restablecimiento de las bases de datos.

5.3. FUNCIONES BÁSICAS.

Las funciones del administrador son las siguientes:

- ✓ Creación de usuarios
- ✓ Creación de base de datos y tablas
- ✓ Restricción de integridad
- ✓ Respaldos y recuperaciones

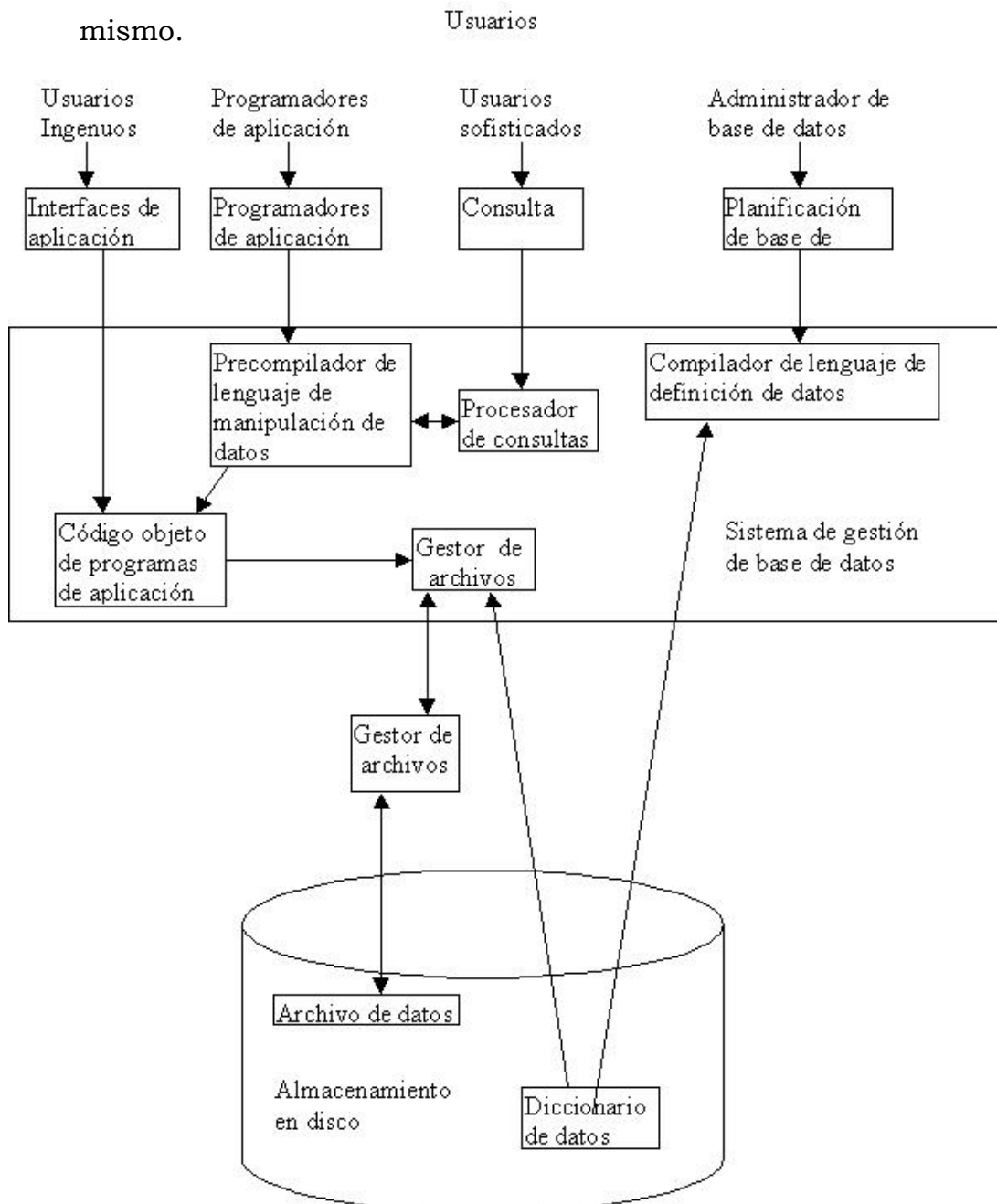
5.3.1. CREACIÓN DE USUARIOS

Usuarios es toda persona que tenga contacto con el sistema de base de datos desde que este se diseña, elabora, termina y se usa.

Los usuarios se pueden clasificarse como:

- **Programadores de aplicaciones:** Son los profesionales que interactúan con el sistema, mediante el DML (Lenguaje de Manipulación de Datos).
- **Usuarios sofisticados:** Son los que interactúa con el sistema sin escribir programas, pero escriben sus preguntas en un lenguaje de consultas de base de datos.

- **Usuarios especializados:** Algunos usuarios escriben aplicaciones de base de datos especializadas que no encajan en el marco tradicional de procesamiento de datos.
- **Usuarios ingenuos:** Son los usuarios finales que utiliza el sistema de base de datos sin saber nada del diseño interno del mismo.



23. Relación de Usuarios [IMAG.23]²⁴

²⁴ http://sistemas.itlp.edu.mx/tutoriales/basedat1/tema1_10.htm



5.3.2. CREACIÓN DE BASES DE DATOS Y TABLAS.

Creando Bases de Datos:

- Localización de las bases de datos.
- Tipo de base de datos (modo de direccionamiento).

Creando Tablas:

- Seleccionando tipos de datos.
- Tablas fragmentadas o no fragmentadas.
- Localización de la tabla.
- Determinación del espacio en disco.
- Modo de aseguramiento de candados.

5.3.3. RESTRICCIONES DE INTEGRIDAD

Cada vez que se tiene lugar una actualización en el sistema, se mantiene una estructura especial que consulta el gestor de la base de datos.

Existen métodos para asegurar la integridad de los datos y estos son:

- Privilegios:
 - ✓ Base de datos.
 - ✓ Tabla.



✓ Columna

- Integridad de identidad, semántica y referencial.
- Vistas.

5.3.4. PROCEDIMIENTOS DE RESPALDO Y RECUPERACIÓN.

El DBA debe definir un plan de recuperación que incluya descargar o vaciar periódicamente la base de datos en un medio de almacenamiento de respaldo, para cargar otra vez la base de datos a partir del vaciado más reciente, en caso ser necesario.

Es posible generar periódicamente copias de respaldo y conservar una serie de versiones anteriores. Cada copia de respaldo estará identificada por tiempo y fecha y por la última transacción incluida. Una copia de respaldo debe generarse mientras la base de datos esta en reposo, ya que las actualizaciones durante el copiado pueden provocar que la copia se inconsistente.

RECUPERACIÓN.

Pasos cuando se presente una falla en el sistema y se desee recuperar la información:

- **Detección del error.** El proceso de recuperación se inicia al detectar la existencia de un error. Se considerarán fallas de



sistemas detectadas por falta de acción del sistema o por verificaciones irrecuperables de redundancia y salida incorrecta observada por un usuario.

- **Determinación de la fuente del error.** Para decidir la acción correctora es necesario determinar la extensión del daño. Es necesario determinar aquellas áreas del archivo de datos que sean sospechosas como cuál fue la transacción que no se concluyó.
- **Ubicación de errores secundarios.** Cuando se ha detectado un error que provocó una modificación inadecuada a un archivo, un rastreo a través de las listas de actividad encontrara aquellas transacciones que emplearon el bloque correcto. Es posible volver a introducir automáticamente el bloque correcto de las transacciones afectadas y producir resultados correctos.
- **Aplicación de correcciones.** Si la extensión del daño es limitada, puede utilizarse un proceso de volver a enrollar. Las porciones dañadas del archivo se restauran aplicando primero aquellas imágenes anteriores a los bloques en error reemplazando después de las transacciones incompletas. La salida proveniente de estas transacciones se suprime de ser posible, para evitar duplicar resultados que previamente se hayan enviado a los usuarios.



COMPARACIÓN

CENTRALIZADO	DISTRIBUIDO
Existe un control total por parte del administrador de la base de datos global. Por razones de seguridad, hay gran énfasis en el control y este debe centralizarse.	Hay un alto grado de autonomía por parte de los administradores locales. En bases de datos distribuidas es posible identificar una estructura de control jerárquica basada en el administrador de la bases de datos global y en los administradores de las bases de datos locales. Estos pueden llegar a tener mucha autonomía.
Deben preservar la atomicidad de las transacciones en presencia de fallas. La recuperación debe hacerse hacia atrás o hacia delante, dependiendo de si la transacción se ha comprometido o no.	Deben preservar la atomicidad de las transacciones aun en casos de fallas en algunos sitios involucrados. La recuperación debe hacerse basado en protocolos complejos de compromiso en dos o en tres fases.
El DBA tiene un control centralizado a través del acceso autorizado a los datos. Pero es más vulnerable a las violaciones de seguridad y privacidad que en archivos separados, desde el punto de vista de la globalidad de la información	Los administradores locales tienen el mismo problema de vulnerabilidad y privacidad de los administradores globales. Por la autonomía de cada sitio, los propietarios de los datos tienen sus propias protecciones, a veces buenas, a veces no tanto. Las bases de datos distribuidas presentan los problemas intrínsecos a los sistemas distribuidos ya que las redes representan un punto débil en protección.



6

DESARROLLO DE LA APLICACIÓN



Contenido:

Desarrollo de la Aplicación

- 6.1 Introducción
- 6.2 Gestión del Proyecto
- 6.3 Modelo del Negocio
- 6.4 Requisitos
- 6.5 Análisis de Diseño
- 6.6 Implementación
- 6.7 Documentación
- 6.8 Documentación



CAPITULO VI

6. DESARROLLO DE LA APLICACIÓN

6.1. INTRODUCCIÓN

En la sección de **Gestión del Proyecto** se muestran las planificaciones de desarrollo del proyecto, así como el cronograma de ejecución del proyecto, de construcción de la aplicación y cumplimiento de los plazos estimados.

En la sección de **Modelado del Negocio** se encuentran los artefactos utilizados de la metodología RUP para definir un modelo del negocio, modelos de objetos del negocio y el modelo del dominio.

En la sección **Requisitos** se encuentra los artefactos definidos según la metodología RUP, es decir, el documento plan de desarrollo de software, el documento visión, el documento glosario, matrices de atributos de todos los requerimientos, los casos de uso y sus especificaciones.

En la sección **Análisis/Diseño** se muestran tanto el modelo de análisis/diseño (diagrama de clases) como el modelo de datos (modelo entidad – relación).

En la sección **Implementación** se muestran los prototipos de interfaces de usuario de la aplicación.



A continuación se detalla las Tecnologías y Herramientas usadas en el Desarrollo del Sistema.

- ✓ **Base de Datos:** Oracle Data Base 10g versión 10.2.0.1.0
- ✓ **Plataforma de Desarrollo:** Java con JDK 1.5
- ✓ **IDE de Desarrollo:** JDeveloper 11g.
- ✓ **Diseñador de Reportes:** iReport-3.0.2
- ✓ **Framework de Desarrollo:** Oracle ADF
- ✓ **Servidor de Aplicaciones:** Web logic.

6.2. GESTIÓN DEL PROYECTO

En esta sección se detalla la planificación inicial del proyecto para la fase de inicio y la fase de elaboración (según la definición de la metodología RUP).

PLAN DE DESARROLLO DEL SOFTWARE

INTRODUCCIÓN

Este Plan de Desarrollo de Software es una versión preliminar preparada para ser incluida en la propuesta elaborada como respuesta al trabajo final de grado previo a la obtención del Título de Ingeniera en Sistemas Computacionales de la Facultad de Ingeniería en Ciencias Aplicadas de la Universidad Técnica del Norte.



El enfoque de desarrollo propuesto constituye una configuración del proceso RUP de acuerdo a las características del proyecto, seleccionando los roles de los participantes, las actividades a realizar y los artefactos (entregables) que serán generados. Este documento es a su vez uno de los artefactos de RUP.

PROPÓSITO

El propósito del Plan de Desarrollo de Software es proporcionar la información necesaria para controlar el proyecto. En él se describe el enfoque de desarrollo de software.

Los usuarios del Plan de Desarrollo de Software son:

- ✓ El jefe de proyecto lo utiliza para organizar la agenda y necesidades de recursos, y para realizar su seguimiento.
- ✓ Los miembros del equipo de desarrollo lo usan para entender lo qué deben hacer, cuándo deben hacerlo y que otras actividades dependen de ello.

ALCANCE

El Plan de Desarrollo de Software describe el plan global usado para el desarrollo de la **“IMPLEMENTACIÓN DE UNA BASE DE DATOS DISTRIBUIDA EN EL SISTEMA ACADÉMICO DE LA FICA”**. Durante el proceso de desarrollo en el artefacto “Visión” se definen las características del producto a desarrollar, lo cual constituye la base para la planificación de las iteraciones.



Para la versión 1.0 del Plan de Desarrollo de Software nos hemos basado en la captura de requisitos por medio del stakeholder representante para hacer una estimación aproximada, una vez comenzado el proyecto y durante la fase de Inicio se generará la primera versión del artefacto “Visión”, el cual se utilizará para refinar este documento.

Posteriormente, el avance del proyecto y el seguimiento en cada una de las iteraciones ocasionará el ajuste de este documento produciendo nuevas versiones actualizadas.

RESUMEN

Después de esta introducción, el resto del documento está organizado en las siguientes secciones:

Vista General del Proyecto: Proporciona una descripción del propósito, alcance y objetivos del proyecto, estableciendo los artefactos que serán producidos y utilizados durante el proyecto.

Organización del Proyecto: Describe la estructura organizacional del equipo de desarrollo.

Gestión del Proceso: Explica los costos y planificación estimada, define las fases e hitos del proyecto y describe cómo se realizará su seguimiento.

Planes y Guías de aplicación: Proporciona una vista global del proceso de desarrollo de software, incluyendo métodos, herramientas y técnicas que serán utilizadas.



VISTA GENERAL DEL PROYECTO

PROPÓSITO, ALCANCE Y OBJETIVOS

La información que a continuación se incluye a sido extraída de las diferentes reuniones que se han celebrado con el stakeholder desde el inicio del proyecto.

El proyecto debe proporcionar una respuesta para el desarrollo de todos los módulos implicados en la **“IMPLEMENTACIÓN DE UNA BASE DE DATOS DISTRIBUIDA EN EL SISTEMA ACADÉMICO DE LA FICA”**. Estos módulos se pueden diferenciar en:

- Roles
- Usuarios
- Estudiantes
- Facultades
- Docentes
- Parámetros
- Escuelas
- Matriculas
- Ingreso de Notas
- Periodos Académicos
- Crear Paralelos



SUPOSICIONES Y RESTRICCIONES

Las suposiciones y restricciones respecto del sistema, y que se derivan directamente de las entrevistas con el stakeholder son:

- a) Debe contemplarse las implicaciones de los siguientes puntos críticos:
 - Sistemas seguros: protección de información, seguridad en las transmisiones de datos, etc.
 - Gestión de flujos de trabajo, seguridad de las transacciones e intercambio de información.
- b) La automatización de la gestión interna del registro debe ajustarse a la legislación vigente.
- c) El módulo de matriculación debe ser desarrollado como un sistema independiente para ser utilizado por todas las Facultades.

Como es natural, la lista de suposiciones y restricciones se incrementará durante el desarrollo del proyecto, particularmente una vez establecido el artefacto “Visión”.

ENTREGABLES DE PROYECTO

A continuación se indican y describen cada uno de los artefactos que serán generados y utilizados por el proyecto y que constituyen los entregables. Esta lista constituyen la configuración de RUP desde la perspectiva de artefactos, y que proponemos para este proyecto.



Es preciso destacar que de acuerdo a la filosofía de RUP (y de todo proceso iterativo e incremental), todos los artefactos son objeto de modificaciones a lo largo del proceso de desarrollo, con lo cual, sólo al término del proceso podríamos tener una versión definitiva y completa de cada uno de ellos. Sin embargo, el resultado de cada iteración y los hitos del proyecto están enfocados a conseguir un cierto grado de completitud y estabilidad de los artefactos. Esto será indicado más adelante cuando se presenten los objetivos de cada iteración.

1) Plan de Desarrollo del Software

Es el presente documento.

2) Modelo de Casos de Uso del Negocio

Es un modelo de las funciones de negocio vistas desde la perspectiva de los actores externos (Agentes de registro, solicitantes finales, otros sistemas etc.). Permite situar al sistema en el contexto organizacional haciendo énfasis en los objetivos en este ámbito. Este modelo se representa con un Diagrama de Casos de Uso usando estereotipos específicos para este modelo.

3) Modelo de Objetos del Negocio

Es un modelo que describe la realización de cada caso de uso del negocio, estableciendo los actores internos, la información que en términos generales manipulan y los flujos de trabajo asociados al caso de uso del negocio. Para la representación de este modelo se utilizan Diagramas de Colaboración (para mostrar actores



externos, internos y las entidades (información) que manipulan, un Diagrama de Clases para mostrar gráficamente las entidades del sistema y sus relaciones, y Diagramas de Actividad para mostrar los flujos de trabajo.

4) Glosario

Es un documento que define los principales términos usados en el proyecto. Permite establecer una terminología consensuada. .

5) Modelo de Casos de Uso

El modelo de Casos de Uso presenta las funciones del sistema y los actores que hacen uso de ellas. Se representa mediante Diagramas de Casos de Uso.

6) Visión

Este documento define la visión del producto desde la perspectiva del cliente, especificando las necesidades y características del producto. Constituye una base de acuerdo en cuanto a los requisitos del sistema.

7) Especificaciones de Casos de Uso

Para los casos de uso que lo requieran (cuya funcionalidad no sea evidente o que no baste con una simple descripción narrativa) se realiza una descripción detallada utilizando una plantilla de documento, donde se incluyen: precondiciones, post-condiciones, flujo de eventos, requisitos no-funcionales asociados. También, para casos de uso cuyo flujo de eventos sea complejo podrá



adjuntarse una representación gráfica mediante un Diagrama de Actividad.

8) Especificaciones Adicionales

Este documento capturará todos los requisitos que no han sido incluidos como parte de los casos de uso y se refieren requisitos no-funcionales globales. Dichos requisitos incluyen: requisitos legales o normas, aplicación de estándares, requisitos de calidad del producto, tales como: confiabilidad, desempeño, etc., u otros requisitos de ambiente, tales como: sistema operativo, requisitos de compatibilidad, etc.

9) Prototipos de Interfaces de Usuario

Se trata de prototipos que permiten al usuario hacerse una idea más o menos precisa de las interfaces que proveerá el sistema y así, conseguir retroalimentación de su parte respecto a los requisitos del sistema. Estos prototipos se realizarán como: dibujos a mano en papel, dibujos con alguna herramienta gráfica o prototipos ejecutables interactivos, siguiendo ese orden de acuerdo al avance del proyecto. Sólo los de este último tipo serán entregados al final de la fase de Elaboración, los otros serán desechados. Asimismo, este artefacto, será desechado en la fase de Construcción en la medida que el resultado de las iteraciones vayan desarrollando el producto final.



10) Modelo de Análisis y Diseño

Este modelo establece la realización de los casos de uso en clases y pasando desde una representación en términos de análisis (sin incluir aspectos de implementación) hacia una de diseño (incluyendo una orientación hacia el entorno de implementación), de acuerdo al avance del proyecto.

11) Modelo de Datos

Previendo que la persistencia de la información del sistema será soportada por una base de datos distribuida, este modelo describe la representación lógica de los datos persistentes, de acuerdo con el enfoque para modelado relacional de datos. Para expresar este modelo se utiliza un Diagrama de Clases (donde se utiliza un profile UML para Modelado de Datos, para conseguir la representación de tablas, claves, etc.).

12) Modelo de Implementación

Este modelo es una colección de componentes y los subsistemas que los contienen. Estos componentes incluyen: ficheros ejecutables, ficheros de código fuente, y todo otro tipo de ficheros necesarios para la implantación y despliegue del sistema. (Este modelo es sólo una versión preliminar al final de la fase de Elaboración, posteriormente tiene bastante refinamiento).



13) Modelo de Despliegue

Este modelo muestra el despliegue la configuración de tipos de nodos del sistema, en los cuales se hará el despliegue de los componentes.

14) Casos de Prueba

Cada prueba es especificada mediante un documento que establece las condiciones de ejecución, las entradas de la prueba, y los resultados esperados. Estos casos de prueba son aplicados como pruebas de regresión en cada iteración. Cada caso de prueba llevará asociado un procedimiento de prueba con las instrucciones para realizar la prueba, y dependiendo del tipo de prueba dicho procedimiento podrá ser automatizable mediante un script de prueba.

15) Solicitud de Cambio

Los cambios propuestos para los artefactos se formalizan mediante este documento. Mediante este documento se hace un seguimiento de los defectos detectados, solicitud de mejoras o cambios en los requisitos del producto. Así se provee un registro de decisiones de cambios, de su evaluación e impacto, y se asegura que éstos sean conocidos por el equipo de desarrollo. Los cambios se establecen respecto al último estado del conjunto de los artefactos en un momento determinado del proyecto.



16) Plan de Iteración

Es un conjunto de actividades y tareas ordenadas temporalmente, con recursos asignados, dependencias entre ellas. Se realiza para cada iteración, y para todas las fases.

17) Evaluación de Iteración

Este documento incluye la evaluación de los resultados de cada iteración, el grado en el cual se han conseguido los objetivos de la iteración, las lecciones aprendidas y los cambios a ser realizados.

18) Lista de Riesgos

Este documento incluye una lista de los riesgos conocidos y vigentes en el proyecto, ordenados en orden decreciente de importancia y con acciones específicas de contingencia o para su mitigación.

19) Manual de Instalación

Este documento incluye las instrucciones para realizar la instalación del producto.

20) Material de Apoyo al Usuario Final

Corresponde a un conjunto de documentos y facilidades de uso del sistema, incluyendo: Guías del Usuario, Guías de Operación, Guías de Mantenimiento.

21) Producto

Los ficheros del producto empaquetados y almacenados en un CD con los mecanismos apropiados para facilitar su instalación. El producto, a partir de la primera iteración de la fase de



construcción es desarrollado incremental e iterativamente, obteniéndose un nuevo reléase al final de cada iteración.

Los artefactos 19, 20 y 21 se generarán a partir de la fase de construcción, con lo cual se han incluido aquí sólo para dar una visión global de todos los artefactos que se generarán en el proceso de desarrollo.

EVOLUCIÓN DEL PLAN DE DESARROLLO DEL SOFTWARE

El Plan de Desarrollo del Software se revisará semanalmente y se refinará antes del comienzo de cada iteración.

ORGANIZACIÓN DEL PROYECTO

PARTICIPANTES EN EL PROYECTO.

El personal participante en el proyecto se encuentra formado por los siguientes puestos de trabajo y personal asociado:

Jefe de Proyecto: Ing. Miguel Orquera.

Arquitecto de Software: Egda. Verónica Alexandra López.

Ingeniero de Software: Egda. Verónica Alexandra López.

Programador: Egda. Verónica Alexandra López.



INTERFACES EXTERNAS

La empresa definirá los participantes del proyecto que proporcionarán los requisitos del sistema, y entre ellos quiénes serán los encargados de evaluar los artefactos de acuerdo a cada módulo y según el plan establecido.

El equipo de desarrollo interactuará activamente con los participantes de la empresa para especificación y validación de los artefactos.

ROLES Y RESPONSABILIDADES

A continuación se describen las principales responsabilidades de cada uno de los puestos en el equipo de desarrollo durante las fases de Inicio y Elaboración, de acuerdo con los roles que desempeñan en RUP

PUESTO	RESPONSABILIDADES
JEFE DE PROYECTO	El jefe de proyecto asigna los recursos, gestiona las prioridades, coordina las interacciones con los clientes y usuarios, y mantiene al equipo del proyecto enfocado en los objetivos. El jefe de proyecto también establece un conjunto de prácticas que aseguran la integridad y calidad de los artefactos del proyecto. Además, el jefe de proyecto se encargará de supervisar el establecimiento de la arquitectura del sistema. Gestión de riesgos. Planificación y control del proyecto.
ARQUITECTO DE SOFTWARE	Se encargará de supervisar el establecimiento de la Arquitectura del Sistema, es decir, definir la vista arquitectónica, los estilos arquitectónicos, el patrón de arquitectura y la arquitectura Tecnológica a utilizar.
INGENIERO DE SOFTWARE	Gestión de requisitos, gestión de configuración y cambios, elaboración del modelo de datos, preparación de las pruebas funcionales, elaboración de la documentación. Elaborar



	modelos de implementación y despliegue. Captura, especificación y validación de requisitos, interactuando con el cliente y los usuarios mediante entrevistas. Elaboración del Modelo de Análisis y Diseño. Colaboración en la elaboración de las pruebas funcionales y el modelo de datos. Encargado además de la puesta en producción.
PROGRAMADOR	Construcción de prototipos. Colaboración en la elaboración de las pruebas funcionales, modelo de datos y en las validaciones con el usuario.
TESTER	Se encargará de la realización de las pruebas funcionales, de conectividad y rendimiento del sistema.

Roles y Responsabilidades

GESTIÓN DE PROCESOS

ESTIMACIONES DEL PROYECTO

El presupuesto del proyecto y los recursos involucrados se adjuntan en un documento por separado.

PLAN DEL PROYECTO

En esta sección se presenta la organización en fases e iteraciones y el calendario del proyecto.



PLAN DE FASES

El desarrollo se llevará a cabo en base a fases con una o más iteraciones en cada una de ellas. La siguiente tabla muestra la distribución de tiempos y el número de iteraciones de cada fase.

FASE	Nro INTERACIONES	DURACIÓN
Fase de Inicio	1	4 semanas
Fase de Elaboración	2	3 semanas
Fase de Construcción	3	9 semanas
Fase de Transición	2	4 semanas

Plan de Fase del Proyecto

Los hitos que marcan el final de cada fase se describen en la siguiente tabla:

DESCRIPCIÓN	HITOS
Fase de Inicio	En esta fase se desarrollará los requisitos del producto desde la perspectiva del usuario, los cuales serán establecidos en el artefacto Visión. Los principales casos de uso serán identificados y se hará un refinamiento al Plan de Desarrollo de Software. La aceptación del cliente/usuario del artefacto Visión y el Plan de Desarrollo de Software marcan el final de esta fase.
Fase de Elaboración	En esta fase se analizan los requisitos y se desarrolla un prototipo de arquitectura (incluyendo las partes más relevantes y/o críticas del sistema). Al final de esta fase, todos los casos de uso correspondientes a requisitos que serán implementados en la primera fase de Construcción deben ser analizados y diseñados (en el Modelo de Análisis/Diseño). La revisión y aceptación del prototipo de la arquitectura del sistema marca el final de esta fase.



Fase de Construcción	Durante la fase de construcción se terminan de analizar y diseñar todos los casos de uso, refinando el Modelo de Análisis/Diseño. El producto se construye en base a dos iteraciones, cada una produciendo un realce a la cual se le aplican las pruebas y se valida con el cliente/usuario. Se comienza la elaboración del material de apoyo al usuario. El hito que marca el fin de esta fase es la versión de la release 3.0, con la capacidad operacional parcial del producto que se haya considerado como crítica, lista para ser entregada a los usuarios para la prueba beta.
Fase de Transición	En esta fase se prepararán dos releases para distribución, asegurando una implantación y cambio del sistema previo de manera adecuada, incluyendo el entrenamiento de los usuarios. El hito que marca el fin de esta fase incluye, la entrega de toda la documentación del proyecto con los manuales de instalación y todo el material de apoyo al usuario, la finalización del entrenamiento de los usuarios y el empaquetamiento del producto

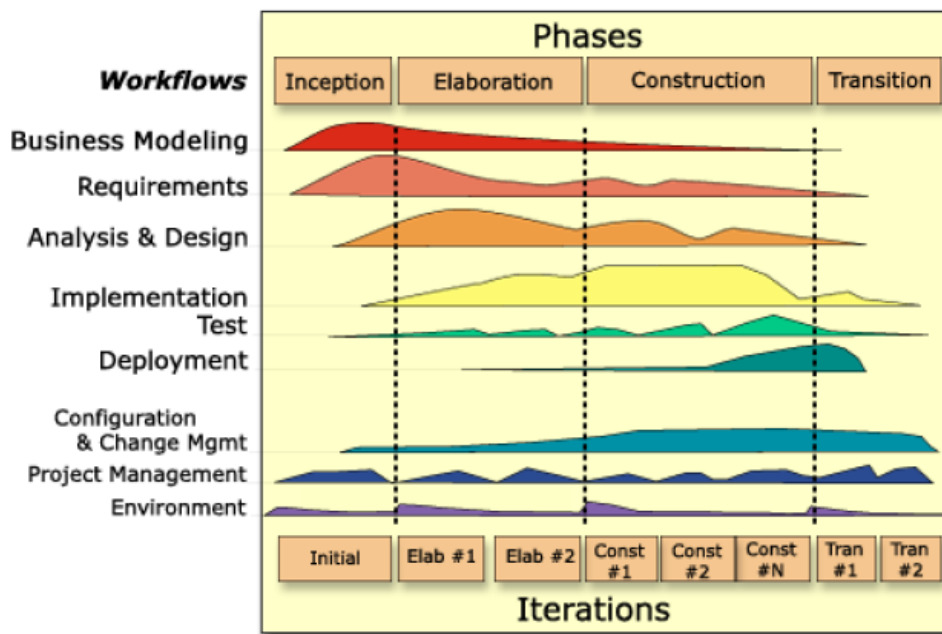
Hitos de las Fases del Proyecto

CALENDARIO DEL PROYECTO

A continuación se presenta un calendario de las principales tareas del proyecto.

Como se ha comentado, el proceso iterativo e incremental de RUP está caracterizado por la realización en paralelo de todas las disciplinas de desarrollo a lo largo del proyecto, con lo cual la mayoría de los artefactos son generados muy tempranamente en el proyecto pero van desarrollándose en mayor o menor grado de acuerdo a la fase e iteración del proyecto.

La siguiente figura ilustra este enfoque, en ella lo ensombrecido marca el énfasis de cada disciplina (workflow) en un momento determinado del desarrollo.



24. Interacciones del Proyecto

Para este proyecto se ha establecido el siguiente calendario. La fecha de aprobación indica cuándo el artefacto en cuestión tiene un estado de completitud suficiente para someterse a revisión y aprobación, pero eso no quita la posibilidad de su posterior refinamiento y cambios

Disciplinas / Artefactos generados o modificados durante la Fase de Inicio	Comienzo	Aprobación
Modelado del Negocio		
Modelo de Casos de Uso del Negocio y Modelo de Objetos del Negocio	Semana 1	Semana 3
Requisitos		
Glosario	Semana 1	Semana 3
Visión	Semana 2	Semana 3
Modelo de Casos de Uso	Semana 3	siguiente fase
Especificación de Casos de Uso	Semana 3	siguiente fase
Especificaciones Adicionales	Semana 3	siguiente fase
Análisis / Diseño		



“FRAGMENTACIÓN DE TABLAS EN BASES DE DATOS DISTRIBUIDAS”

Modelo de Análisis / Diseño	Semana 2	siguiente fase
Modelo de Datos	Semana 2	siguiente fase
Implementación		
Prototipos de Interfaces de Usuario	Semana 3	siguiente fase
Modelo de Implementación	Semana 3	siguiente fase
Pruebas		
Casos de Pruebas Funcionales	Semana 3	siguiente fase
Despliegue		
Modelo de Despliegue	Semana 3	siguiente fase
Gestión de Cambios y Configuración	Durante todo el proyecto	
Gestión del proyecto		
Plan de Desarrollo del Software en su versión 1.0 y planes de las Iteraciones	Semana 1	Semana 3
Ambiente	Durante todo el proyecto	

Calendario del proyecto. Fase de Inicio

Disciplinas / Artefactos generados o modificados durante la Fase de Elaboración	Comienzo	Aprobación
Modelado del Negocio		
Modelo de Casos de Uso del Negocio y Modelo de Objetos del Negocio	Semana 1	Aprobado
Requisitos		
Glosario	Semana 1	Aprobado
Visión	Semana 2	Aprobado
Modelo de Casos de Uso	Semana 3	Semana 5
Especificación de Casos de Uso	Semana 3	Semana 5
Especificaciones Adicionales	Semana 3	Semana 5
Análisis / Diseño		
Modelo de Análisis / Diseño	Semana 2	Revisar en cada iteración
Modelo de Datos	Semana 2	Revisar en cada iteración
Implementación		
Prototipos de Interfaces de Usuario	Semana 3	Revisar en cada iteración
Modelo de Implementación	Semana 3	Revisar en cada iteración
Pruebas		
Casos de Pruebas Funcionales	Semana 3	Revisar en cada iteración



Despliegue		
Modelo de Despliegue	Semana 3	Revisar en cada iteración
Gestión de Cambios y Configuración	Durante todo el proyecto	
Gestión del proyecto		
Plan de Desarrollo del Software en su versión 2.0 y planes de las Iteraciones	Semana 4	Revisar en cada iteración
Ambiente	Durante todo el proyecto	

Calendario del Proyecto. Fase de Elaboración

SEGUIMIENTO Y CONTROL DEL PROYECTO

Gestión de Requisitos

Los requisitos del sistema son especificados en el artefacto Visión. Cada requisito tendrá una serie de atributos tales como importancia, estado, iteración donde se implementa, etc. Estos atributos permitirán realizar un efectivo seguimiento de cada requisito. Los cambios en los requisitos serán gestionados mediante una Solicitud de Cambio, las cuales serán evaluadas y distribuidas para asegurar la integridad del sistema y el correcto proceso de gestión de configuración y cambios.

Control de Plazos

El calendario del proyecto tendrá un seguimiento y evaluación semanal por el jefe de proyecto.



Control de Calidad

Los defectos detectados en las revisiones y formalizados también en una Solicitud de Cambio tendrán un seguimiento para asegurar la conformidad respecto de la solución de dichas deficiencias. Para la revisión de cada artefacto y su correspondiente garantía de calidad se utilizarán las guías de revisión y checklist (listas de verificación) incluidas en RUP.

Gestión de Riesgos

A partir de la fase de Inicio se mantendrá una lista de riesgos asociados al proyecto y de las acciones establecidas como estrategia para mitigarlos o acciones de contingencia. Esta lista será evaluada al menos una vez en cada iteración.

Gestión de Configuración

Se realizará una gestión de configuración para llevar un registro de los artefactos generados y sus versiones. También se incluirá la gestión de las Solicitudes de Cambio y de las modificaciones que éstas produzcan, informando y publicando dichos cambios para que sean accesibles a todo los participantes en el proyecto. Al final de cada iteración se establecerá una baseline (un registro del estado de cada artefacto, estableciendo una versión), la cual podrá ser modificada sólo por una Solicitud de Cambio aprobada.

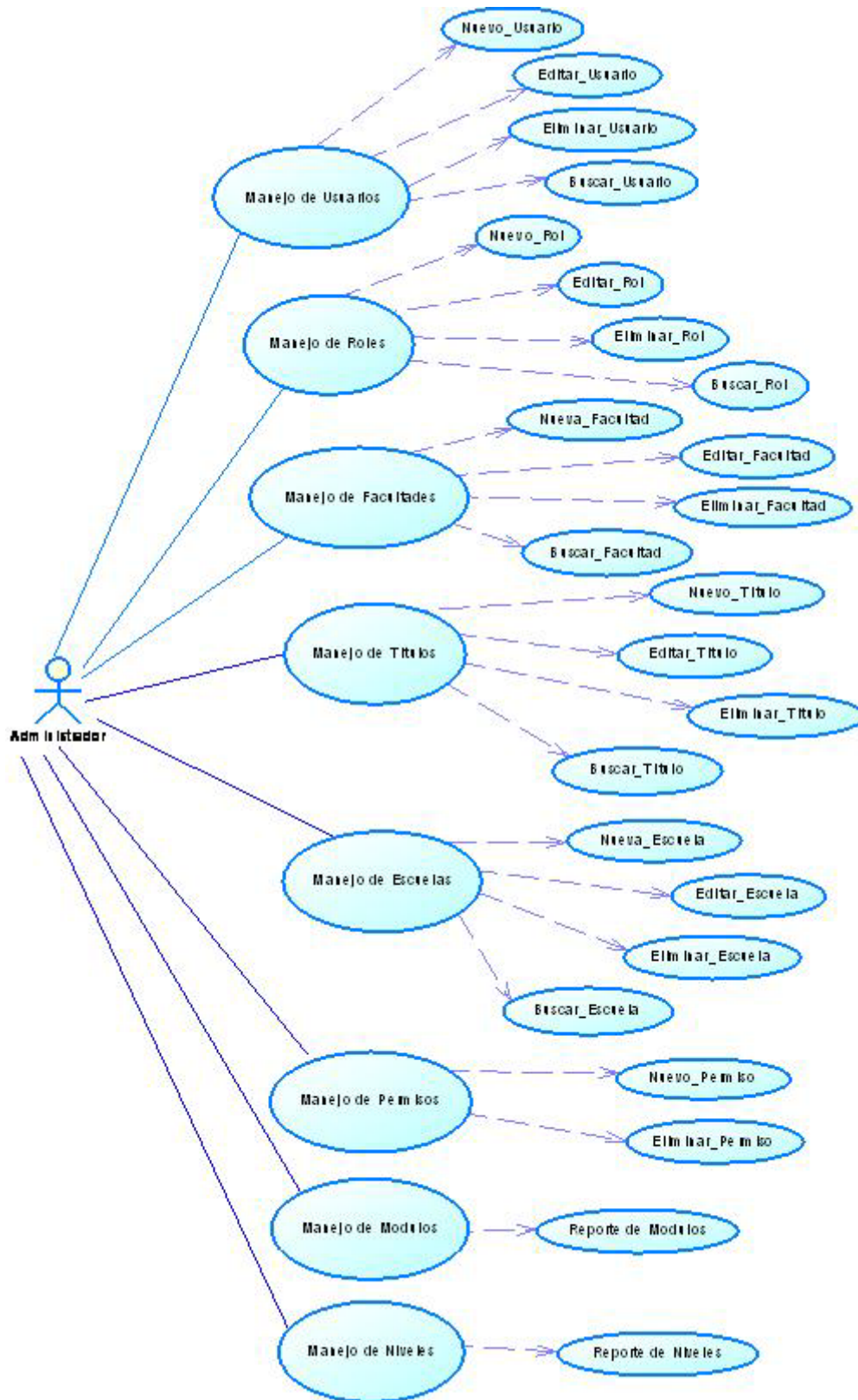


6.3. MODELADO DEL NEGOCIO

A continuación se presentan los modelos definidos en RUP como modelo del negocio, modelo de datos y modelo de análisis y diseño

- Administrador
- Usuario
- Docente
- Secretaria Escuela

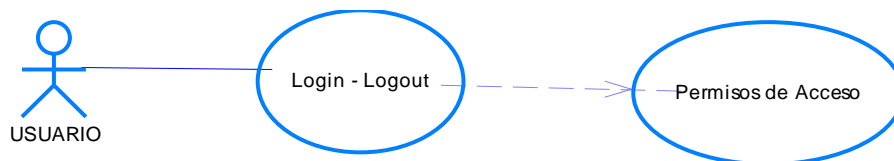
ADMINISTRADOR



25. Diagrama de caso de Uso del Administrador. Fuente Propia

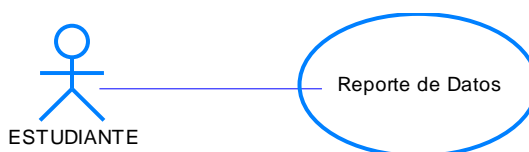


USUARIO



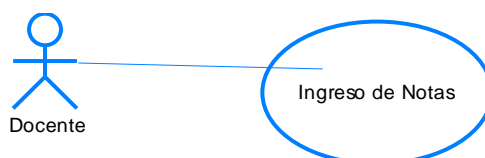
26. Diagrama de caso de Uso del Usuario. Fuente Propia

ESTUDIANTE



27. Diagrama de caso de Uso del Estudiante. Fuente Propia

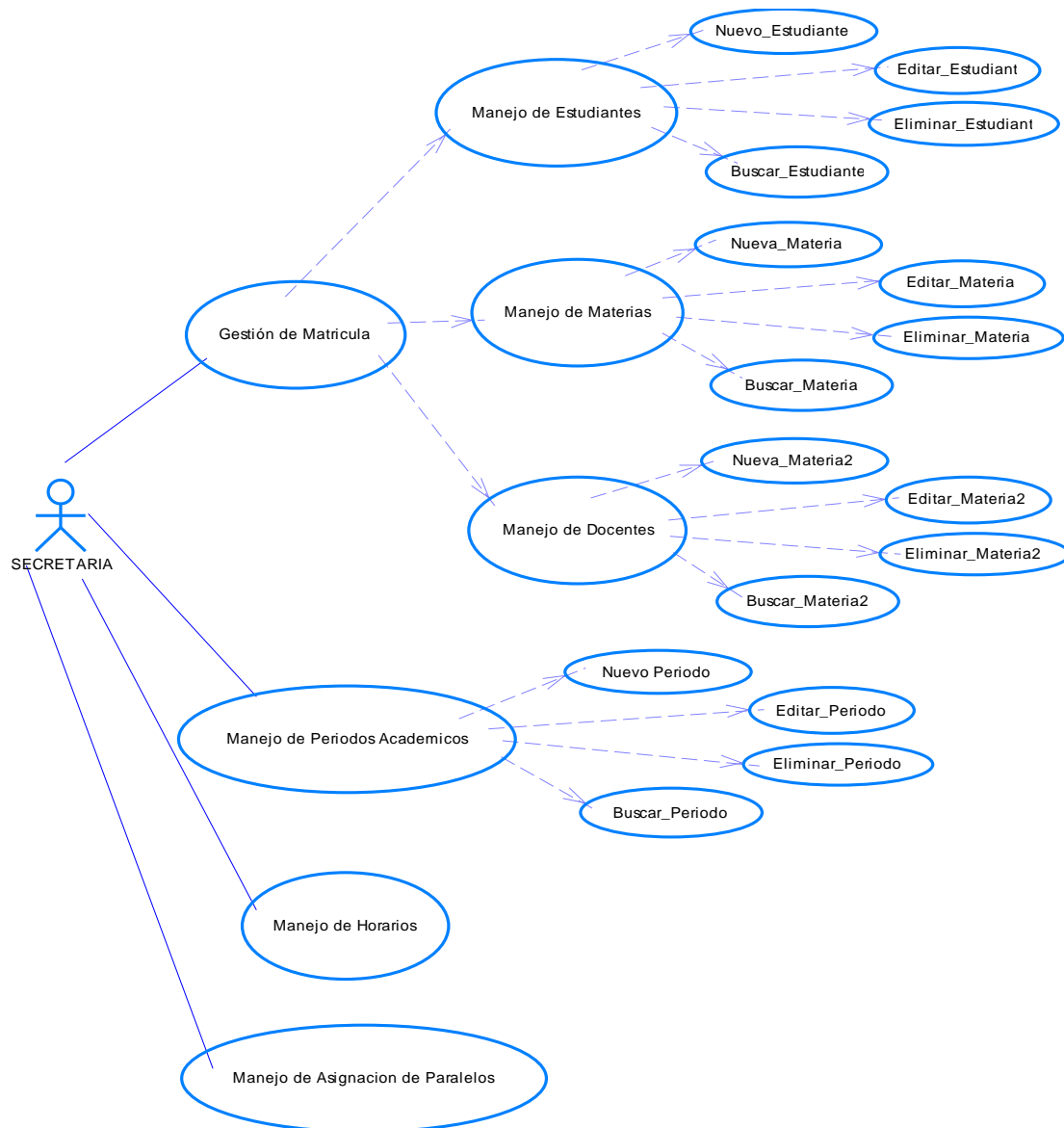
DOCENTE



28. Diagrama de caso de Uso del Docente. Fuente Propia



SECRETARIA



29. Diagrama de caso de Uso de la Secretaria de la Escuela. Fuente Propia



6.4. REQUISITOS

A continuación se presentan las herramienta utilizadas para declarar los requisitos software, es decir, el documento plan de desarrollo software, el documento visión, el documento glosario y tanto las especificaciones de los casos de uso.

VISIÓN

INTRODUCCIÓN

Este documento define la visión del producto desde la perspectiva del cliente, especificando las necesidades y características del producto. Constituye una base de acuerdo a los requisitos del sistema

PROPÓSITO

El propósito de éste documento es recoger, analizar y definir las necesidades de alto nivel y las características del sistema de la “Implementación de una Base de Datos Distribuida en el Sistema Académico e la FICA”.

Los detalles de cómo el sistema cubre los requerimientos se pueden observar en la especificación de los casos de uso y otros documentos adicionales.



ALCANCE

El documento Visión se ocupa, como ya se ha detallado, el sistema de “Implementación de una Base de Datos Distribuida en el Sistema Académico e la FICA”, dicho sistema será desarrollado por Verónica Alexandra López Tulcanaza.

El sistema permitirá a los encargados de la empresa controlar todo lo relativo al Sistema Académico (ingreso de estudiantes, docentes, materias, crear periodos académicos, matriculación etc).

POSICIONAMIENTO

OPORTUNIDAD DE NEGOCIO

Este sistema permitirá a la empresa informatizar el control de sus actividades (ingresos de datos como estudiantes, docentes y materias, matriculas, reportes, etc.), lo cual supondrá un acceso rápido y sencillo a los datos, gracias a interfaces gráficas sencillas y amigables. Además los datos accedidos estarán siempre actualizados, lo cual es un factor muy importante.



DESCRIPCIÓN DE STAKEHOLDERS (PARTICIPANTES EN EL PROYECTO) Y USUARIOS

Para proveer de una forma efectiva productos y servicios que se ajusten a las necesidades de los usuarios, es necesario identificar e involucrar a todos los participantes en el proyecto como parte del proceso de modelado de requerimientos. También es necesario identificar a los usuarios del sistema y asegurarse de que el conjunto de participantes en el proyecto los representa adecuadamente. Esta sección muestra un perfil de los participantes y de los usuarios involucrados en el proyecto, así como los problemas más importantes que éstos perciben para enfocar la solución propuesta hacia ellos. No describe sus requisitos específicos ya que éstos se capturan mediante otro artefacto. En lugar de esto proporciona la justificación de por qué estos requisitos son necesarios.

RESUMEN DE STAKEHOLDERS

Nombre	Descripción	Responsabilidades
Jefe del Proyecto	Representante Global	El stakeholder realiza: Representa a todos los usuarios posibles del sistema. Seguimiento del desarrollo del proyecto. Aprueba requisitos y funcionalidades

Resumen de los Stakeholders del Proyecto.



RESUMEN DE USUARIOS

Nombre	Descripción	Responsabilidades
Administrador	Responsable de la gestión de usuarios como es: roles y permisos.	Seguridad
Secretaria	Responsable del correcto ingreso de datos como de estudiante, docentes, materias y matriculación.	Integridad de Datos
Docente	Encargado de ingreso de notas de los estudiantes	Ingreso notas

Resumen de los Usuarios del Proyecto

ENTORNO DE USUARIOS

Los usuarios entrarán al sistema mediante un navegador web, identificándose de acuerdo al tipo de usuario con su login y password, tras este paso accederán a la aplicación diseñada de acuerdo al los requerimientos de cada usuario. Los reportes serán generados en pdf como en Microsoft Excel, lo cual le resultará familiar.

PERFIL DE LOS STAKEHOLDERS

REPRESENTANTE DEL ÁREA TÉCNICA Y SISTEMAS

DE INFORMACIÓN

Representante	Jefe del proyecto
Descripción	Representante Global
Tipo	Experto de Sistemas.
Responsabilidades	Encargado de mostrar las necesidades de cada usuario del sistema. Además, lleva a cabo un seguimiento del desarrollo del proyecto y aprobación de los requisitos y funcionalidades del sistema
Criterio de Éxito	A definir por el cliente
Grado de participación	Revisión de requerimientos, estructura del sistema

Representante del Área Técnica del Proyecto



PERFILES DE USUARIOS

ADMINISTRADOR

Representante	Seguridad
Descripción	Administrador
Tipo	Experto de Sistemas.
Responsabilidades	Responsable de la gestión de usuarios como es: roles y permisos.
Criterio de Éxito	A definir por el cliente
Grado de participación	A definir por el cliente

Usuario Administrador del Proyecto

SECRETARIA

Representante	Integridad de Datos
Descripción	Secretaria de cada Escuela
Tipo	Usuario experto
Responsabilidades	Responsable del correcto ingreso de datos como de estudiante, docentes, materias y matriculación
Criterio de Éxito	A definir por el cliente
Grado de participación	A definir por el cliente

Usuario Secretaria del Proyecto

DOCENTE

Representante	Ingreso de Notas
Descripción	Docente
Tipo	Usuario Experto
Responsabilidades	Encargado de ingreso de notas de los estudiantes
Criterio de Éxito	A definir por el cliente
Grado de participación	A definir por el cliente

Usuario Docente del Proyecto



DESCRIPCIÓN GLOBAL DE PRODUCTO

PERSPECTIVA DEL PRODUCTO

El producto a desarrollar es “Implementación de una Base de Datos Distribuida en el Sistema Académico de la FICA”, con la intención de agilizar su funcionamiento. Las áreas a tratar por el sistema son: Ingreso de estudiantes, docentes, materias y la gestión de matriculación.

RESUMEN DE LAS CARACTERÍSTICAS

A continuación se mostrará un listado de los beneficios que obtendrá el cliente a partir del producto.

Beneficio del cliente	Características que lo apoyan
Mayor agilidad y rapidez para ver datos y la posibilidad de hacerlo vía servicios web.	Aplicación web.
Mayor control de información.	Base de datos distribuida ya que desde cualquier sitio pueden realizar sus consultas como si fuera en el sitio mismo.
Seguridad.	El ingreso del sistema se controla por medio de un usuario y contraseña, se controla el acceso a las opciones a través de permisos y roles.

Resumen de las características del Proyecto

DESCRIPCIÓN DE PRODUCTO

SEGURIDAD

El administrador del sistema será encargado del manejo del la Seguridad.



SECRETARIA ESCUELA

La Secretaria de cada Escuela tendrá acceso a todo el modulo de matriculación, como también de ingreso de estudiantes, docentes y materias.

DOCENTE

El Docente se encarga del ingreso de notas de los estudiantes por materia que el imparte en cada escuela.

RESTRICCIONES

OTROS REQUISITOS DEL PRODUCTO

ESTÁNDARES APLICABLES

REQUISITOS DE SISTEMA

SERVIDOR

Windows XP.

Espacio libre en Disco Duro de 5 GB o mayor.

Procesador Pentium IV o mayor.

Memoria RAM 1G o mayor.

Mouse / Teclado

Monitor

REQUISITOS DE DESEMPEÑO

El mayor requisito de desempeño es la facilidad y rapidez para el acceso de datos debido a que sus datos se encuentran distribuidos evitando los cuellos de botellas. Y el sistema es agradable a la vista del usuario.



REQUISITOS DE ENTORNO

El hardware debe estar conectado de manera adecuada. Se debe mantener siempre los dispositivos de red y servidores web para el perfecto ingreso al sistema.

REQUISITOS DE DOCUMENTACIÓN

MANUAL DE USUARIO

El manual de usuario se encuentra en los anexos el cual contendrá información como: instalación del sistema, modo de acceder a cada rol.

GUÍAS DE INSTALACIÓN, CONFIGURACIÓN Y

FICHERO LÉAME

Las guías de instalación, configuración y el fichero léame se encuentra en los anexos del proyecto



CASOS DE USO

A continuación se presentan los diagramas de casos definidos. Cabe destacar que los casos de uso que no se incluyeron en la fase de construcción sólo figuran en estado de propuestos.

- **USUARIO**

En este caso de uso permite al usuario poder ingresar al sistema según el rol y los privilegios asignados por el administrador. Este puede ser: Usuario del Sistema, Docente o Estudiante.

FLUJO DE EVENTOS

FLUJO BÁSICOS

LOGIN

1. El sistema solicita al usuario su cuenta este puede ser:
Usuario del Sistema, Docente o Estudiante.
2. El usuario ingresa a su cuenta con su respectivo nombre de usuario, ingresando también su contraseña mediante la interfaz del sistema (“Autenticación de Usuarios”).
3. Una vez ingresado su login y contraseña debe seleccionar la opción “*Entrar*” para poder ingresar al sistema.



FLUJOS ALTERNATIVOS

ERROR DE CUENTA O CONTRASEÑA DE USUARIO

Según el punto dos. Si el usuario ingresa de manera errónea su cuenta o clave el sistema se encarga de mostrar un mensaje de error, que dice lo siguiente: **“Usuario no Valido”**

PRECONDICIONES

El usuario debe tener previamente creado su cuenta. En caso de no tenerla deberá hablar con el administrador para que le proporcione una.

POSTCONDICIONES

El sistema estará conectando mediante la sesión y el perfil que el usuario ingrese, una vez ingresado podrá acceder a todos los privilegios asignados para su cuenta.

- **USUARIO DEL SISTEMA (SECRETARIA ESCUELA)**

La Secretaria inicia el caso de uso. Una vez ingresado al sistema este le muestra las diferentes opciones que ella puede administrar. Por ejemplo, Estudiante, Docente, Ingreso notas, Matricula, Periodo Académico, Crear Paralelo etc. Ella podrá crear, editar y eliminar según sea el criterio o las necesidades. Para su entendimiento vamos a explicar cómo crear un Estudiante



FLUJO DE EVENTOS

FLUJO BÁSICO

El sistema le permite manejar diferentes operaciones como ya lo mencionamos. Puede buscar a un estudiante mediante: cedula, nombre, apellido, fecha nacimiento, etc. Además puede crear, editar, eliminar y exportar xls.

CREAR ESTUDIANTE.

1. El caso de uso comienza cuando la secretaria en el sistema académico hace clic en Estudiante.
2. El sistema académico muestra todos los datos que debe ingresar para crear un nuevo Estudiante.
3. Una vez ingresado todos los datos debe elegir la opción de “*Guardar*” para guardar al nuevo Estudiante.
4. El sistema se encarga de guardar al estudiante.
5. Una vez guardado el estudiante se debe elegir la opción “*Regresar*” para regresar al menú donde partimos inicialmente.

EDITAR ESTUDIANTE

1. La secretaria define la búsqueda ya sea mediante cedula, nombre, apellido etc.
2. Una vez localizado el estudiante que se desea editar, procedemos a editar los campos que se crea conveniente.



3. Una vez actualizados los datos debe elegir la opción de “*Guardar*” para guardar los datos modificados del Estudiante.
4. El sistema se encarga de guardar los datos editados del estudiante.
6. Una vez guardado el estudiante se debe elegir la opción “*Regresar*” para regresar al menú donde partimos inicialmente.

ELIMINAR ESTUDIANTE

1. La secretaria define la búsqueda ya sea mediante cedula, nombre, apellido etc.
2. Una vez localizado el estudiante que se desea eliminar.
3. Ahora se debe presiona el botón “*Eliminar*”. Y el estudiante se eliminara automáticamente.
4. El sistema elimina al estudiante de la base de datos.

NO COLOCA CRITERIO DE BÚSQUEDA

Si en editar y eliminar estudiante la secretaria no ingresa nada en la opción de búsqueda, el sistema mostrará todos los estudiantes ingresados en el sistema hasta el momento.

PRECONDICIONES

La secretaria debe ingresar con su cuenta y contraseña y perfil para poder acceder a los diferentes módulos.



POSTCONDICIONES

El sistema estará conectando mediante la sesión y el rol que el usuario ingrese, una vez ingresado podrá acceder a todos los privilegios asignados para su cuenta

- **DOCENTE**

El docente inicia el caso de uso ingresando al sistema según el rol y los privilegios asignados por el administrador. Este puede ingresar notas de estudiantes de cada materia que el imparte en las diferentes escuelas.

FLUJO DE EVENTOS

FLUJO BÁSICOS

1. El sistema solicita al usuario su perfil este puede ser:
Usuario del Sistema, Docente o Estudiante.
2. El Docente debe ingresar su nombre de usuario, su clave y elegir su perfil, en este caso es el de Docente.
3. Debe elegir la escuela a la que pertenece la materia que va ingresar la nota
4. Ahora se debe elegir la materia.
5. Y por ultimo elige Ingreso de Notas, una vez ingresada las notas de los estudiante matriculados para esta materia tiene que guardar.



FLUJOS ALTERNATIVOS

ERROR DE CUENTA O CONTRASEÑA DE USUARIO

Según el punto dos. Si el usuario ingresa de manera errónea su cuenta o clave el sistema se encarga de mostrar un mensaje de error, que dice lo siguiente: **“Usuario no Valido”**

PRECONDICIONES

El usuario debe tener previamente creado su cuenta. En caso de no tenerla deberá hablar con el administrador par que le proporcione una.

POS CONDICIONES

El sistema estará conectando mediante la sesión y el rol que el usuario ingrese, una vez ingresado podrá acceder a todos los privilegios asignados para su cuenta.

- **ESTUDIANTE**

El estudiante inicia el caso de uso ingresando al sistema según el rol y los privilegios asignados por el administrador. Este puede consultar sus notas en el determinado periodo académico.

FLUJO DE EVENTOS

FLUJO BÁSICOS

1. El sistema solicita al usuario su perfil este puede ser:
Usuario del Sistema, Docente o Estudiante.



2. El Estudiante debe ingresar su nombre de usuario, su clave y elegir su perfil, en este caso es el de Estudiante.
3. Enseguida le aparece el reporte de todas las materias que se encuentra matriculado en dicho periodo académico con sus respectivas notas, el estudiante si desea puede imprimir su reporte, simplemente haciendo click en imprimir se bajara el archivo en pdf, el cual podrá imprimirlo.

FLUJOS ALTERNATIVOS

ERROR DE CUENTA O CONTRASEÑA DE USUARIO

Según el punto dos. Si el usuario ingresa de manera errónea su cuenta o clave el sistema se encarga de mostrar un mensaje de error, que dice lo siguiente: **“Usuario no Valido”**

PRECONDICIONES

El usuario debe tener previamente creado su cuenta. En caso de no tenerla deberá hablar con el administrador para que le proporcione una.

POS CONDICIONES

El sistema estará conectando mediante la sesión y el rol que el usuario ingrese, una vez ingresado podrá acceder a todos los privilegios asignados para su cuenta.



REQUERIMIENTOS

STAKEHOLDERS

Los representantes de los usuarios y portavoces de las necesidades son los stakeholders. En este proyecto solamente se ha tratado con un stakeholder como representante de los usuarios y necesidades, sin embargo se han dividido representativamente.

La matriz de atributos de los stakeholders es la siguiente:

Requerimientos	Representante	Ubicación
STK 1 Administrador	Responsable de la gestión de usuarios como es: roles y permisos.	Documento Visión
STK 2 Secretaria	Responsable del correcto ingreso de datos como de estudiante, docentes, materias y matriculación.	Documento Visión
STK 3 Docente	Encargado de ingreso de notas de los estudiantes	Documento Visión

Matriz de atributos de los Stakeholders del Proyecto

ACTORES

Se define este requerimiento para listar los usuarios potenciales del sistema, en este proyecto se han definido los siguientes actores:

Administrador, Usuario del Sistema (Secretaria Escuela), Docente y Estudiante.

Requerimientos	Ubicación	Modulo
ACT1: Administrador	Documento Visión	Seguridad
ACT2: Secretaria Escuela	Documento Visión	Ingreso de Datos y Matriculas
ACT3: Docente	Documento Visión	Ingreso de Notas
ACT4: Estudiante	Documento Visión	Reportes de Notas

Matriz de atributos de los Actores del Proyecto



CARACTERÍSTICAS DE SOFTWARE

Las características de software son las necesidades de los usuarios propuestas por los stakeholders, son los requisitos que debe cumplir el sistema para satisfacer las necesidades.

Las características definidas son las que aparecen en la matriz de atributos, siendo las indicadas como subcaracterísticas las derivadas según una clasificación jerárquica.

Requerimientos	Asignado a
CSW 1: Seguridad Seguridad	
CSW1.1: Manejo de Usuarios Manejo de Usuarios	Equipo completo de análisis, desarrollo e implementación
CSW1.2: Manejo de Roles Manejo de Roles	Equipo completo de análisis, desarrollo e implementación
CSW1.3: Manejo de Permisos Manejo de Permisos	Equipo completo de análisis, desarrollo e implementación
CSW1.4: Manejo de Niveles Manejo de Niveles	Equipo completo de análisis, desarrollo e implementación
CSW1.5: Manejo de Facultades Manejo de Facultades	Equipo completo de análisis, desarrollo e implementación
CSW1.6: Manejo de Escuelas Manejo de Escuelas	Equipo completo de análisis, desarrollo e implementación
CSW1.7: Manejo de Títulos Manejo de Títulos	Equipo completo de análisis, desarrollo e implementación
CSW 2: Usuario del Sistema Secretaria Escuela	
CSW 2.1: Gestión de Matricula Secretaria	Equipo completo de análisis, desarrollo e implementación
CSW 2.1.1: Manejo de Estudiante Manejo de Estudiante	Equipo completo de análisis, desarrollo e implementación
CSW 2.1.2: Manejo de Materia Manejo de Materia	Equipo completo de análisis, desarrollo e implementación
CSW 2.1.3: Manejo de Docente Manejo de Docente	Equipo completo de análisis, desarrollo e implementación
CSW 2.2: Manejo de Periodo Académico Manejo de Periodo Académico	Equipo completo de análisis, desarrollo e implementación
CSW 2.3: Manejo Crear Paralelo Manejo Crear Paralelo	Equipo completo de análisis, desarrollo e implementación
CSW 2.4: Manejo de Horarios Manejo de Horarios	Equipo completo de análisis, desarrollo e implementación
CSW 3: Docente Docente	
CSW 3.1: Manejo de Ingreso de Notas Manejo de Ingreso de Notas	Equipo completo de análisis, desarrollo e implementación
CSW 4: Manejo De Estudiante	



Manejo De Estudiante	
CSW 4.1: Manejo de Reporte Manejo de Reportes	Equipo completo de análisis, desarrollo e implementación

Matriz de atributos de las características del Proyecto

CASOS DE USO

Derivados de las características de software, son el resultado del análisis de las necesidades de los usuarios. La matriz de atributos es la siguiente:

Requerimientos	Asignado a
ECU1: Gestión de Usuarios	Equipo de desarrollo
ECU2: Buscar Usuarios	Equipo de desarrollo
ECU3: Manejo de Roles	Equipo de desarrollo
ECU4: Buscar Roles	Equipo de desarrollo
ECU5: Manejo de Permisos	Equipo de desarrollo
ECU6: Buscar Permisos	Equipo de desarrollo
ECU7: Manejo de Niveles	Equipo de desarrollo
ECU8: Buscar Niveles	Equipo de desarrollo
ECU9: Manejo de Facultades	Equipo de desarrollo
ECU10: Buscar Facultades	Equipo de desarrollo
ECU11: Manejo de Escuelas	Equipo de desarrollo
ECU12: Buscar Escuelas	Equipo de desarrollo
ECU13: Manejo de Títulos	Equipo de desarrollo
ECU14: Buscar de Títulos	Equipo de desarrollo
ECU15: Gestión de Matricula	Equipo de desarrollo
ECU16: Manejo de Estudiante	Equipo de desarrollo
ECU17: Buscar Estudiante	Equipo de desarrollo
ECU18: Eliminar Estudiante	Equipo de desarrollo
ECU19: Reporte de Estudiante	Equipo de desarrollo
ECU20: Manejo de Materia	Equipo de desarrollo
ECU21: Buscar Materia	Equipo de desarrollo
ECU22: Eliminar Materia	Equipo de desarrollo
ECU23: Reporte Materia	Equipo de desarrollo
ECU24: Manejo de Docente	Equipo de desarrollo
ECU25: Buscar Docente	Equipo de desarrollo
ECU26: Eliminar Materia	Equipo de desarrollo
ECU27: Reporte Materia	Equipo de desarrollo
ECU28: Manejo de Periodo Académico	Equipo de desarrollo
ECU29: Buscar Periodo Académico	Equipo de desarrollo
ECU30: Eliminar Periodo Académico	Equipo de desarrollo
ECU27: Reporte de Periodo Académico	Equipo de desarrollo
ECU28: Manejo Crear Paralelo	Equipo de desarrollo
ECU29: Manejo de Horarios	Equipo de desarrollo

Matriz de atributos de los Casos de Uso del Proyecto



CLASES

Las clases son requerimientos derivados de los casos de uso como necesidad de representación del modelo de datos.

Requerimientos	Ubicación
CLS1: Docentes	Base de Datos
CLS2: Escuelas	Base de Datos
CLS 3: Estudiantes	Base de Datos
CLS 4: Facultades	Base de Datos
CLS 5: horarios	Base de Datos
CLS 6: Materias	Base de Datos
CLS 7: Materias_Matriculas	Base de Datos
CLS 8: Matriculas	Base de Datos
CLS 9: Modulos	Base de Datos
CLS 10: Niveles	Base de Datos
CLS 11: Paralelos_Materias	Base de Datos
CLS 12: Periodo_Academico	Base de Datos
CLS 13: Permisos	Base de Datos
CLS 14: Roles	Base de Datos
CLS 15: Títulos	Base de Datos
CLS 16: Usuarios	Base de Datos

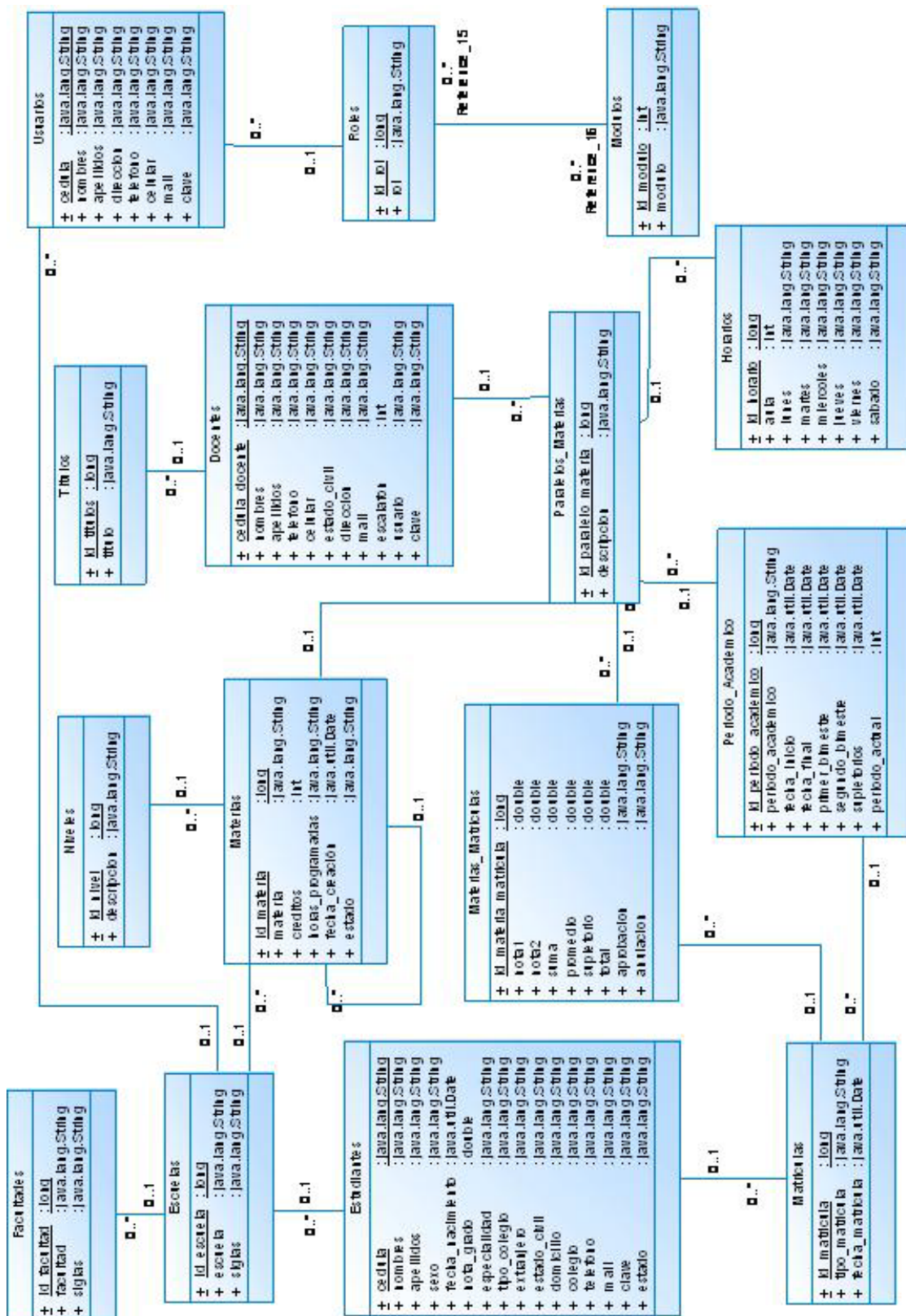
Matriz de atributos de las Clases del Proyecto

6.5. ANALISIS DE DISEÑO

A continuación se presentan los modelos definidos en RUP como modelo de datos y modelo de análisis / diseño. Constará de un diagrama de clases, y de un modelo de datos (modelo relacional).



DIAGRAMA DE CLASES



30. Diagrama de Clase del Sistema Académico



6.6. IMPLEMENTACIÓN

A continuación se presentan los modelos definidos en RUP como prototipos de interfaces gráficas de usuario diseñados para la aplicación final.

INTERFACES DE USUARIOS

INICIO



MENÚ PRINCIPAL





Listado de Estudiantes

http://127.0.0.1...tate=6xk75bm6d_4

Sistema Académico

Administración ▾ Generales ▾ Académico ▾ Guardar Cancelar Regresar Sesión de Administrador null

Estudiantes

Ver ▾ Nuevo Editar Eliminar Exportar XLS Separar

Cedula	Nombres	Apellidos	FechaNacimiento	Domicilio	Telefono	Mail	Escuela	Facultad
0400697496	TIRSO ATILO	ERAZO RUEDA	14/05/1988	AZAYA	2602564	tilo_erazo@yahoo.co	Ingenieria Mecatronica	Facultad de Ingenieri
0400775235	LUIS EDMUNDO	BRACHO GONZALEZ	12/03/1990	SAN FRANCISCO	2967143	lucho_bracho@yahoo	Ingenieria Diseño Tex	Facultad de Ingenieri
0500217773	CESAR ANTONIO	SANTO MENA	19/06/1990	GUAYAQUIL DE ALPA	2608726	cesr_nacho19@yahoo	Ingenieria Diseño Tex	Facultad de Ingenieri
0500799168	YOLANDA CONCEPCI	SALAS ECHEVERRIA	19/09/1990	AV. EL RETORNO	2609456	concep_salas17@yah	Ingenieria Diseño Tex	Facultad de Ingenieri
0702388075	FATIMA MARIBEL	TORRES ROMAN	10/04/1988	LA VICTORIA	2604132	globmari_78@yahoo	Ingenieria Diseño Tex	Facultad de Ingenieri
1704589264	CLARA EMPERATRIZ	ESPINOZA PEREZ	18/09/1990	CARANQUI	2605398	clara_20_perez@yah	Ingenieria Diseño Tex	Facultad de Ingenieri
1800985077	LUIS EDGAR	CHILUIZA CARRILLO	28/11/1990	LOS CEIBOS	2609376	luchito_27_ed@yah	Ingenieria Diseño Tex	Facultad de Ingenieri
1001692365	LUIS RAMIRO	LIMA MOROCHO	30/08/1990	OTAVALO	2601562	lu_ramiro@yahoo.co	Ingenieria Diseño Tex	Facultad de Ingenieri
0400912036	JHANET EREDMITA	OVIEDO PANTOJA	25/06/1990	ATUNTAQUI	2608726	ered_ovie_990@yah	Ingenieria Diseño Tex	Facultad de Ingenieri
1001284023	WASHINGTON GUSTA	PROAÑO PAEZ	18/09/1990	LA VICTORIA	2956569	gus_paez@yahoo.co	Ingenieria Diseño Tex	Facultad de Ingenieri
1001298635	AMANDA GERMANIA	CAPELO ALBUJA	22/07/1990	ATUNTAQUI	2608574	amand_gema187@y	Ingenieria Diseño Tex	Facultad de Ingenieri
1002877270	SILVIA MAGALY	CHANDI IMBAQUING	25/01/1990	AV. 17 DE JULIO	2955674	silvi_maga35@yahoo	Ingenieria Diseño Tex	Facultad de Ingenieri

CREACIÓN Y EDICIÓN

http://127.0.0.1...tate=6xk75bm6d_4

Sistema Académico

Administración ▾ Generales ▾ Académico ▾ Guardar Cancelar Regresar Sesión de Administrador null

* Cedula: 0400697496

Nombres: TIRSO ATILO

Apellidos: ERAZO RUEDA

Sexo: Masculino Femenino

FechaNacimiento: 14/05/1988

NotaGrado: 17

Especialidad: QUIMICO BIOLOGO

TipoColegio: Particular

Extranjero: Si No

EstadoCivil: Soltero

Domicilio: AZAYA

Colegio: SAN FRANCISCO

Telefono: 2602564

Mail: tilo_erazo@yahoo.com

Clave: 0400697496

Estado: Activo Deshabilitado

IdFacultadBind: Facultad de Ingenieria en Ciencias Aplicadas

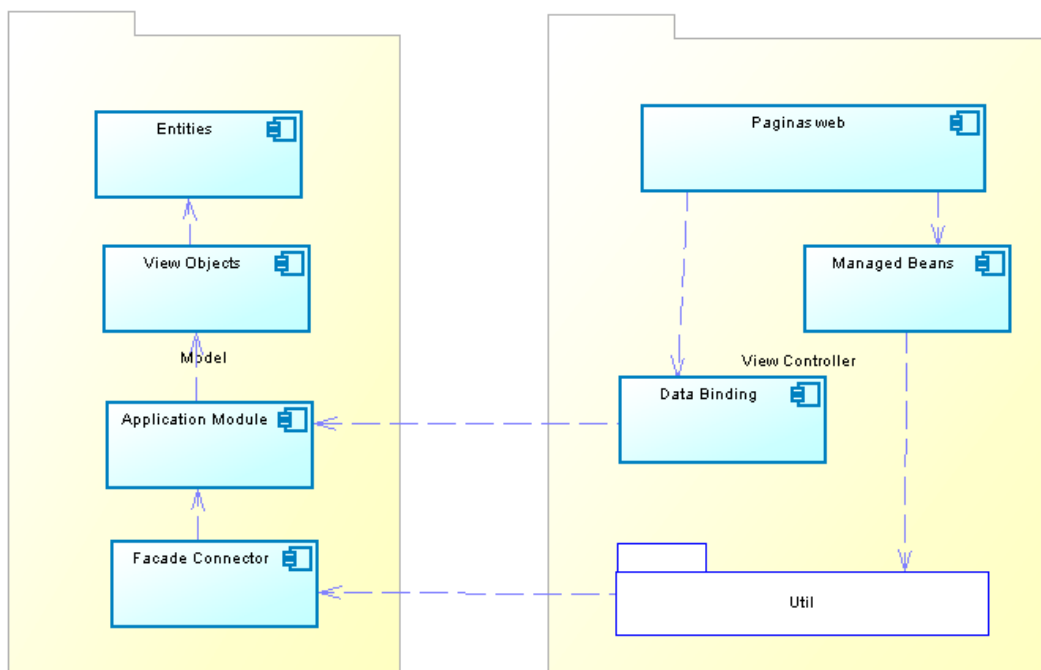
IdEscuela: Ingenieria Mecatronica

Escuela: Ingenieria Mecatronica

Facultad: Facultad de Ingenieria en Ciencias Aplicadas



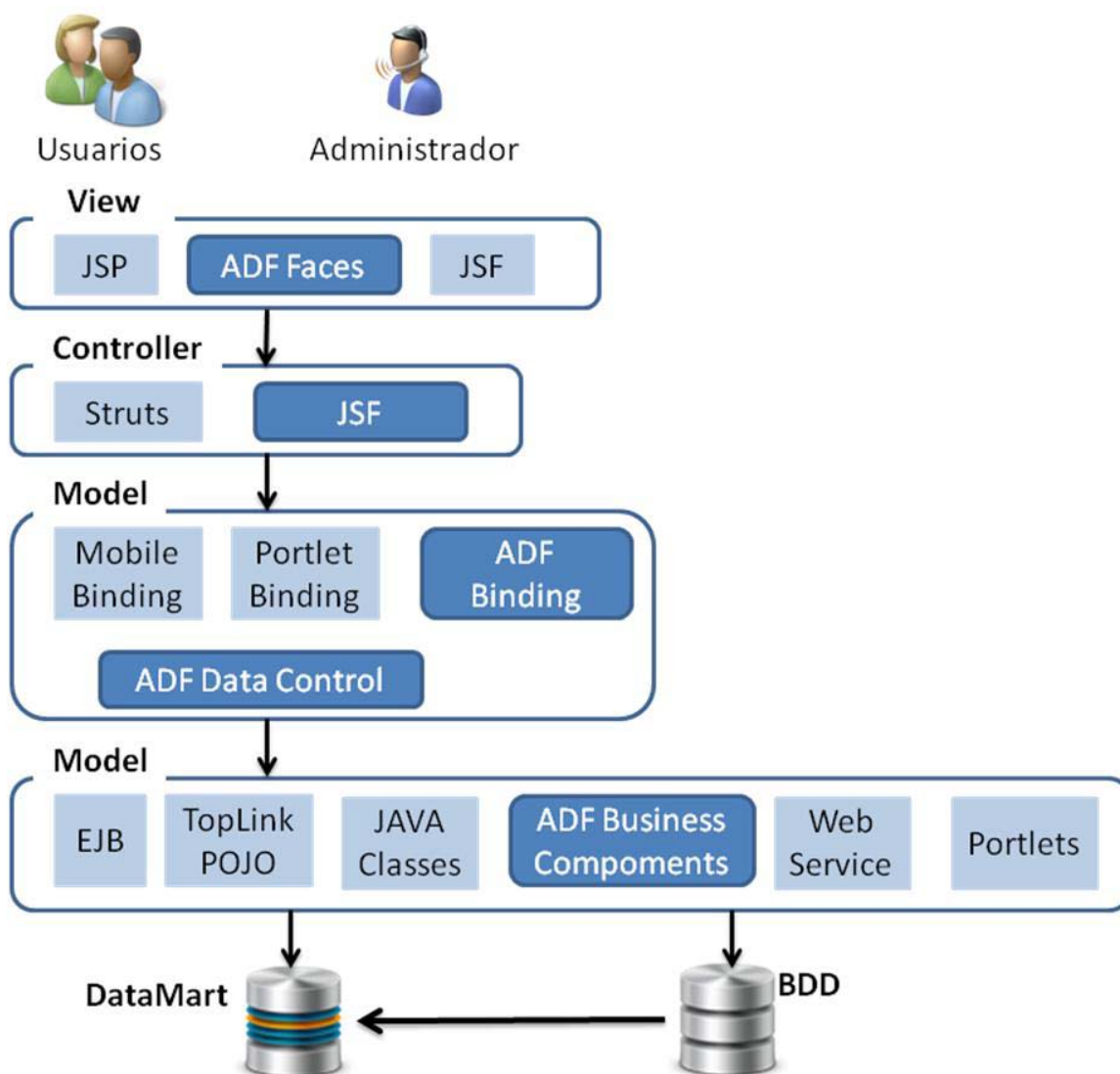
DIAGRAMA DE COMPONENTES



32. Diagrama de Componentes del Proyecto



DIAGRAMA DE DESPLIEGUE



33. Diagrama de Despliegue del Proyecto

6.7. PRUEBAS

A continuación se muestran las especificaciones de casos de prueba funcionales de los casos de uso incluidos en el proyecto de desarrollo de software.



ESPECIFICACIÓN DEL CASO DE PRUEBA: “LOGIN - LOGOUT”

INTRODUCCIÓN

En la presente sección se detallan las pruebas a realizarse para los distintos escenarios del caso de uso “*Login – Logout*”.

PROPÓSITO

Probar que el caso de uso “*Login – Logout*” está correctamente implementado y que se cumplen las especificaciones funcionales y no funcionales.

ALCANCE

Sólo se prueban los escenarios mencionados en el documento de Especificación de Caso de Uso: “*Login – Logout*”.

DEFINICIONES, ACRÓNIMOS Y ABREVIACIONES

Para la presente sección tener en cuenta los siguientes términos:

Autenticar: Se refiere al hecho de haber ingresado sus datos para que la aplicación pueda identificar que la persona que intenta acceder a su contenido es quien dice ser. Esta acción se lo denomina técnicamente Login.

Sesión: Período de tiempo de actividad que un usuario pasa en el sistema, desde que hace el Login hasta que hace Logout o sale del sistema.

ESCENARIOS DE PRUEBA

El presente documento contiene los distintos escenarios que se detallaron en la especificación del caso de uso “*Login – Logout*” y cada escenario tiene una breve descripción de lo que trata, el flujo



de actividades que contiene los pasos a realizarse para cumplir el escenario y los puntos de control que indican los pasos donde evaluar exhaustivamente.

ESCENARIO: FLUJO BÁSICO

Validar que el usuario pueda iniciar y terminar su sesión de manera correcta.

Descripción: Es el escenario ideal del caso de uso, no deberían presentarse errores.

PRECONDICIONES

El usuario no debe tener su sesión activa en el sistema. Se deben haber creado el usuario y sus datos indicando como dato de entrada.

DATOS DE ENTRADA

Se accederá al sistema con el usuario “100” cuya contraseña es “*admin*”. Su perfil es Administrador.

ESCENARIO: ERROR DE CUENTA DE USUARIO

Verificar el reconocimiento de las cuentas de usuario validando su existencia en la base de datos.

Descripción: Comprobar que una cuenta se encuentra o no registrada en la base de datos.

PRECONDICIONES

La cuenta de usuario no ha sido registrada en la base de datos.

DATOS DE ENTRADA

Un usuario cualquiera “*xxx*” con clave “*xxx*”.



FLUJO DE ACTIVIDADES

Instrucción

El usuario ingresa su cuenta de usuario.

Resultado esperado

El sistema muestra mensaje de error “*Usuario no valido*” y retorna a la interfaz que lo solicito.

PUNTO DE CONTROLADOR

Al final de dicha acción muestra el mensaje correspondiente.

PUNTOS DE REVISIÓN

Paso Punto de Control

Muestra: mensaje de error.

Validación a realizar

Verificar que los datos sean los correctos, comprobar que el sistema muestre el mensaje de acuerdo al error



6.8. DOCUMENTACIÓN

ORACLE ADF

ADF (Application Development Framework) es un entorno de trabajo, un framework de aplicaciones que se basa en la plataforma Java EE (Enterprise Edition).

Simplifica y acelera la implementación de aplicaciones orientadas a servicios y el desarrollo de aplicaciones empresariales que buscan visualizar, crear, modificar y validar datos.

Reduce la cantidad de código ya que ADF proporciona implementaciones de patrones de diseño hasta la implementación, con los datos de arrastrar y soltar vinculante, el diseño visual de interfaz de usuario.

Las aplicaciones desarrolladas con Oracle ADF pueden instalarse en cualquier servidor de aplicaciones Java EE compatible.

La única herramienta que trabaja con ADF es JDeveloper 10g. y 11g.

ARQUITECTURA

ADF implementa el patrón MVC

Oracle ADF está basado en el patrón de diseño Modelo/Vista/Controlador (MVC). Este modelo separa las aplicaciones en tres capas por cuestiones de rendimiento,



productividad del desarrollador o escalabilidad entre otras, delegando la responsabilidad de funciones de la siguiente manera:

1) Capa Modelo: Implementa los objetos, encapsulando la lógica de negocio y es la responsable de la persistencia y la interacción con la base de datos, guardando el estado de la aplicación, está compuesta por dos tipos de componentes

- **ADF Data Controls:** Implementa una abstracción de los servicios de negocio para el cliente. Definen los datos devueltos por el servicio de negocio, estos son: atributos, colecciones y operaciones. Estos se crean arrastrando el servicio de negocio directamente al panel “data control palette” de JDeveloper. Los data control no muestran ningún detalle de la implementación, permitiendo el intercambio de implementaciones de manera rápida y limpia.
- **ADF Binding:** Define cómo trabajan los clientes con los data control. Los bindings son creados de manera automática cuando arrastramos un atributo del data control a la página.

2) Capa Vista: Implementa la interfaz de usuario. Pide datos al modelo a través del controlador. Envía eventos al controlador.

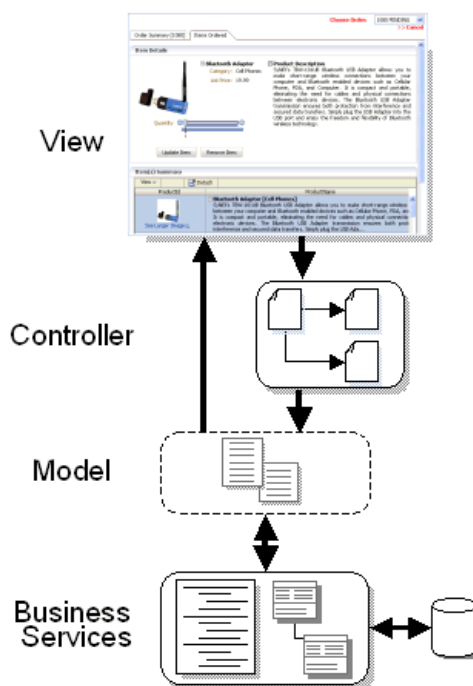
Pasa el control al Controlador para que decida la siguiente vista a mostrar (“enrutado”).



3) Capa Control: Es la responsable de controlar el flujo de navegación y de redireccionar las peticiones del usuario. Obviamente en una aplicación J2EE que siga el patrón MVC la comunicación entre capas es esencial. La capa control en este aspecto juega un papel fundamental puesto que es la encargada de recibir las peticiones de la interfaz de usuario y llamar al modelo para ejecutar reglas de negocio e interactuar con la base de datos.

[www.02]²⁵

ARQUITECTURA MVC



34. Arquitectura MVC separa limpiamente la interfaz de usuario, lógica de negocio y Navegación de página [IMAG.34]²⁶

²⁵ <http://oracleadffaces.blogspot.com/2006/11/un-poco-de-teora-el-patrn-mvc-model.html>

²⁶ http://download.oracle.com/docs/cd/E15051_01/web.1111/b31974/intro.htm



- La capa del modelo representa los valores de datos relacionados con la página actual
- La capa vista contiene las páginas de interfaz de usuario utiliza para ver o modificar los datos
- La capa de controlador de los procesos de entrada del usuario y determina navegación de la página
- La capa de servicios de negocio maneja acceso a datos y encapsula la lógica empresarial

ARQUITECTURA ADF

La Arquitectura de ADF está basada en MVC. No obstante de las tres capas definidas en MVC, incorpora una capa denominada Business Service (“Servicios de Negocio”) que es el resultado de la división de la capa Modelo original.

La capa de Servicio de Negocios se encarga del acceso a las fuentes de datos es decir a la base de datos.

MVC	ADF
Modelo	Modelo
	Servicios de Negocio
Vista	Vista
Controlador	Controlador

35. División de Capa de Modelo en un Servicio de Negocio [IMAG.35]²⁷

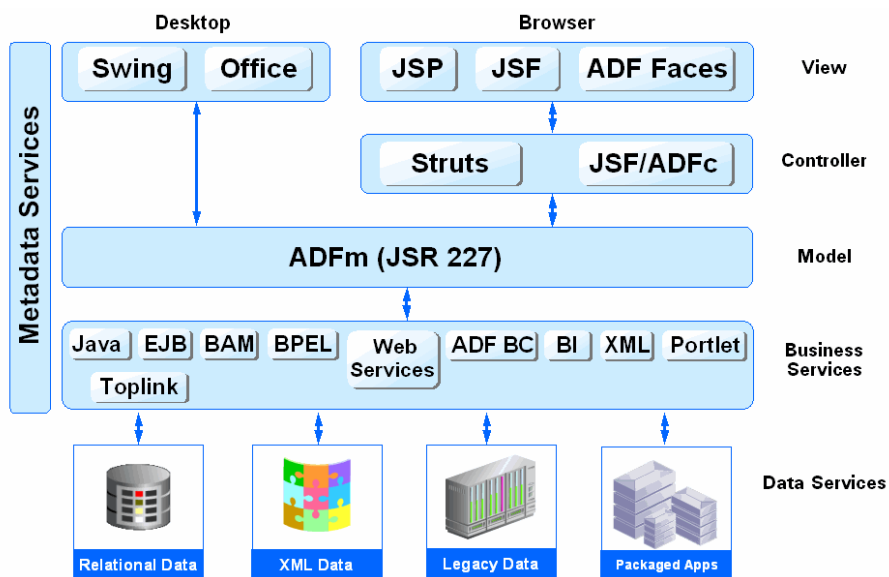
²⁷ http://www.cuore.es/otros/seminarios/seminario_adf.pdf



La capa de Oracle ADF modelo es la principal, debido a que implementa la especificación JSR-227, esta provee una API para acceder a datos de metadatos. Esta capa permite un enfoque de obligar a cualquier interfaz de usuario, sin la necesidad de escribir código.

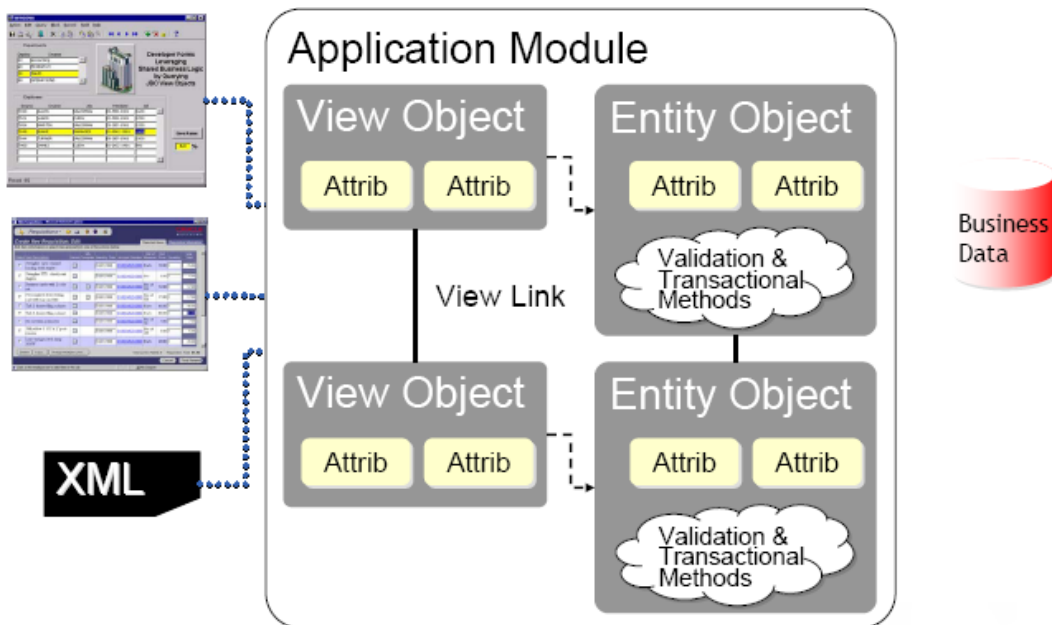
Los otros módulos de tecnología de aplicaciones son:

- Oracle ADF Business Components, que simplifica la creación de servicios de negocios.
- Oracle ADF Faces, que ofrece una amplia biblioteca de componentes AJAX interfaz de usuario para aplicaciones web construidas con JavaServer Faces (JSF).
- Oracle ADF Contralor, que integra JSF con Oracle ADF Modelo. El Contralor ADF extiende su estándar de controlador de JSF, proporcionando una funcionalidad adicional, como los flujos de trabajo reutilizables que pasan a controlar no sólo entre las páginas JSF, sino también entre otras actividades, para las llamadas de método de instancia o de otros flujos de tareas.



36. Arquitectura ADF [IMAG.36]²⁸

ADF BUSINESS COMPONENTS (BC)



37. ADF Business Components [IMAG.37]²⁹

²⁸ http://download.oracle.com/docs/cd/E15051_01/web.1111/b31974/intro.htm

²⁹ http://www.bbr.cat/presentaciones/PDF/Noticias_EventosBbr/BBR+Forms+Java_v1.pdf



Surge la necesidad de tener un framework que abstraiga la complejidad de la plataforma J2EE.

Oracle ADF Business Components es un framework de desarrollo que provee de un conjunto de librerías con funcionalidades que permiten crear componentes de negocios, ayudándonos a mantener la plataforma J2EE, convirtiéndose en un puente entre la base de datos y la aplicación encargándose así de la función de validar los datos.

ADF incrementa la productividad de los desarrolladores ya que implementan componentes reusables.

Sus componentes cubren los siguientes aspectos:

- Persistencia
- Reglas de negocio
- Seguridad
- Validaciones
- Transacciones

ADF BC está compuesta por los siguientes componentes:

- Componentes del dominio del negocio.
- Componentes del modelo de datos.



COMPONENTES DEL DOMINIO DEL NEGOCIO

Representan características asociados a los objetos de la base de datos, y estos son: Entity Objects y Associations

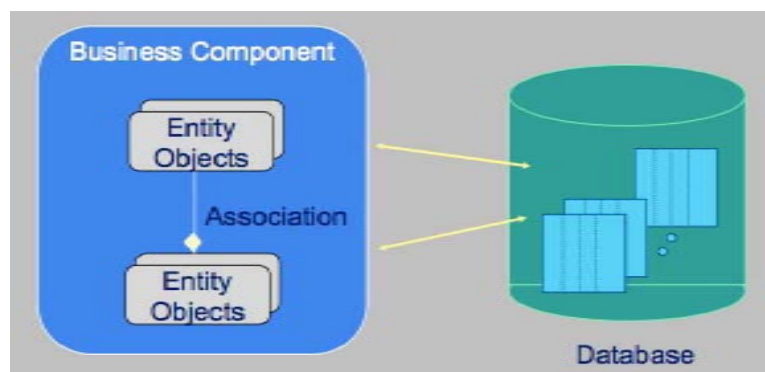
- **Entity Object**

Son los primeros componentes que se debemos crear con Jdeveloper.

Un Entity Object representa un objeto o entidad, cada de ellos mapean una fuente de datos, generalmente una tabla o una vista de una base de datos. Estos contienen atributos, validaciones e información sobre la persistencia. Manejan las interacciones con la base de datos, es decir cada cambio es visto en la entidad y posteriormente se debe hacer un “commit” en la base de datos para que dichos cambios se persistan.

- **Associations**

Está compuesto de una entidad origen (master) y una entidad destino (detail). Son los mecanismo para relacionar dos Entity Objects. Su implementación es mediante XML que es generado por JDeveloper automáticamente.



38. Componentes del dominio del negocio [IMAG.38]³⁰

COMPONENTES DEL MODELO DE DATOS

Recogen los datos de los componentes del dominio del negocio para mostrarlos al resto de las capas (vista y controlador) por medio del modelo, estos son: View Objects y View Links.

- **View Objects**

Generalmente son usados para hacer joins, filtrados y ordenaciones de los datos. Los View Objects representan una vista de los datos, estos pueden construirse directamente o usando cualquier número de Entity Objects mediante sentencias SQL.

³⁰ http://www.cuore.es/otros/seminarios/seminario_adf.pdf



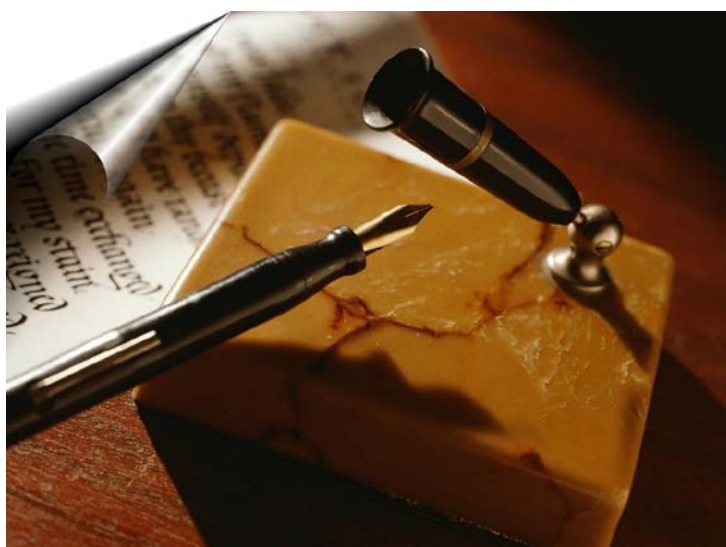
PROCESO DE DESARROLLO CON ADF EN JDEVELOPER

1. Crear el workspace de la aplicación
2. Crear las capas de Servicios de Negocio (Business Services) y Modelo.
3. Crear las capas Controlador y Vista.
4. Testear y depurar.
5. Desplegar en un servidor de aplicaciones.



7

CONCLUSIONES Y RECOMENDACIONES



Contenido:

Conclusiones y Recomendaciones

- 7.1 Verificación de la Hipótesis
 - 7.1.1 Hipótesis
 - 7.1.2 Verificación
- 7.2 Conclusiones
- 7.3 Recomendaciones
- 7.4 Posibles Temas de Tesis
- 7.5 Bibliografía



CAPITULO VII

7. CONCLUSIONES Y RECOMENDACIONES

7.1. Verificación de Hipótesis

7.1.1. Hipótesis

La implementación de una Base de Datos Distribuidas mejora el rendimiento y la accesibilidad de los datos, debido a una disminución de tráfico de la red y a un trabajo colaborativo de varios servidores como si fuera un solo.

7.1.2. Verificación

Por el estudio realizado durante la elaboración de este trabajo de grado podemos decir que la hipótesis ha sido comprobada, ya que se construyó una aplicación de base de datos distribuidas para el sistema académico de la FICA, en la que cada carrera tendrá en su servidor los datos solo de ésta, tanto de estudiantes como, materias y las que sean necesarias ya que se encuentra físicamente esparcido en varios sitios de la red, con el fin de que un usuario de cualquier sitio pueda obtener acceso a los datos, tal como si todos los datos estuvieran almacenados en el sitio propio del usuario.



La fragmentación fue viable debido a que los datos fueron almacenarse en fragmentos donde son utilizados con mayor frecuencia, de tal manera que la mayor parte de las operaciones son locales lo cual redujo el tráfico en la red.



7.2. CONCLUSIONES

- La poca información existente dificultó el normal desarrollo del presente trabajo.
- La falta de orientación de un profesional, experto en la materia debido a que tuve algunos inconvenientes sobre la distribución de datos.
- Al culminar este trabajo, pude constatar que, la implementación de bases de datos distribuidas resulta importante ya que se puede distribuir y replicar en diferentes sitios cuando ciertos sitios locales tienen capacidades de almacenamiento y procesamiento limitadas, pero tienen la ventaja de poder integrarse a otros sitios remotos con mejores recursos por medio de una red de comunicación.
- Las redes de comunicación son un punto fundamental para que las bases de datos pasen de un escenario centralizado a uno distribuido
- A medida que se fue avanzando el desarrollo del trabajo se puede dar una visión global de las características que conlleva el diseño de una base de datos distribuida.
- En este documento indicamos la localización del dato, la replicación y tipos fragmentación, con el fin de dar al lector interesado en las bases distribuidas, un punto de referencia de los temas actuales.



- Se busca simplificar de un modo entendible las técnicas de fragmentación horizontal, vertical y mixta.
- Se ha visto que a lo largo Oracle ha mejorado continuamente la funcionalidad de particionamiento, ya que incorpora nuevas técnicas en cada versión, mejorando las capacidades de mantenimiento, administración, desempeño y disponibilidad de cualquier aplicación de base de datos.
- Las técnicas no son complicadas y se ha incluido ejemplos para facilitar el entendimiento
- El particionamiento es transparente para la aplicación ya que se puede implementar fácilmente, debido a que no requieren cambios en las aplicaciones.
- Las técnicas de particionamiento no son complicadas, lo que si debemos tomar en cuenta cómo se las va a realizar.
- Oracle ADF proporciona un entorno RAD (Rapid Application Development), es decir podemos desarrollar aplicaciones rápidas y eficientes y en muchos de los casos sin escribir una sola línea de código Java.
- ADF está basado en el MVC (Modelo Vista Controlador) y nos permitirá trabajar con aplicaciones web o de escritorio ya que nuestros ADF Business Components se centrarán en la lógica de negocios lo cual nos da la ventaja de la facilidad del mantenimiento de nuestra aplicación.
- ADF es el framework de desarrollo estratégico para Oracle.



7.3. RECOMENDACIONES

- Se debe tener en cuenta la existencia de fragmentación más complejas y eficientes, ya sea las técnicas de fragmentación vertical que se basa en grafos, que genera en un solo paso fragmentos verticales.
- Se debe ampliar el estudio de Oracle ADF, ya que es un framework de desarrollo innovador que provee de un conjunto de librerías que permiten crear componentes de negocios específicos convirtiéndose en un puente entre la base de datos y la aplicación.
- Si una base de datos es demasiado grande, se debe hacer un rediseño de base de datos centralizada, para que esta sea una base de datos distribuida ya que se pueden almacenar los datos en localidades donde son utilizados con mayor frecuencia, de tal manera que la mayor parte de las operaciones sean sólo locales lo cual reduce el tráfico en la red.
- Se recomienda a la Facultad de Ingeniería en Ciencias Aplicadas que incremente la creación de la Matria de Base de Datos Distribuidas en vista que es un tema muy amplio y que en la actualidad se estan usando con mayor frecuencia debido al crecimiento de información almacenada en sus tablas.



- Es recomendable cambiar de un sistema centralizado a un sistema distribuido siempre y cuando sus tablas superen las 2 GB de información, ahí es cuando se debe realizar una fragmentación.
- No se puede especificar en que caso se debe realizar la fragmentación horizontal o vertical debido a que las tablas no son estandar, mas bien ya depende del criterio o del uso frecuente de ciertos datos.



7.4. POSIBLES TEMAS DE TESIS

- Estudio de las diferentes estrategias de particionamiento de Adaptive Server Enterprise 15.0, como son:
 - Particionamiento hash (semántico)
 - Particionamiento por lista (semántico)
 - Particionamiento round- robin
 - Particionamiento por rango (semántico)
- El estudio de Replication Server: Una herramienta para replicación asincrónica entre sistemas heterogéneos.
- Estudio de Tablas e Índices particionados en SQL Server 2005
- Estudio del particionamiento de un índice o una tabla organizada por índices.



7.5. BIBLIOGRAFÍA

INTERNET.

- **Título:** Base de Datos Distribuidas
Url:
http://cablemodem.fibertel.com.ar/bucanero/facultad/Lab4_TP6.doc
Fuente: Universidad de Palermo
- **Título:** Diseño y Optimización de Bases de Datos: Base de Datos Distribuidas
Url:
<http://www.oei.eui.upm.es/Asignaturas/BD/DYOBD/distribuidas.pdf>
Fuente: Universidad Politécnica Madrid
- **Título:** Bases de Datos Distribuidas
Url:
http://html.rincondelvago.com/bases-de-datos-distribuidas_1.html
Fuente: rincondelvago.com
- **Título:** Unidad 1 Base de Datos Distribuidas
- **Url:**
<http://basedatosavanzadaselem.obolog.com/unidad-1-base-datos-distribuidas-523685>
Fuente: Ogolog.com
- **Título:** Particionamiento de Tablas en Oracle
- **Url:**
<http://www.dataprix.com/blogs/respinosamilla/particionado-tablas-oracle>
Fuente: Dtaprix.com
- **Título:** Base de Datos Distribuidas
Url:
http://usuarios.multimania.es/admin2master/documentos/descargas/ppt/BASES_DE_DATOS_DISTRIBUIDAS.PPT
Fuente:
- **Título:** Arquitecturas de las BDD
Url:
<http://www.google.com.ec/url?sa=t&source=web&cd=4&ved=0CB8QFjAD&url=http%3A%2F%2Fdircompucv.ciens.ucv.ve%2Fgenerador%2Fsites%2Fbases-de-datos-dist%2Farchivos%2FBDD%2520-%2520Clase%25202.pps&rct=j&q=%BFQue+es+independencia+de+datos%3F++Es+la+inmunidad+de+las+aplicaciones+de+usuarios+a+los+cambios+en+la+definici%F3n+y+organizaci%F3n+de+los+datos+y+viceversa.&ei=00YmTKXal8KAlAeq7P2lAw&usg=AFQjCNH8JW6hh0LX5fMedB9AwofatAvO-A>
Fuente: Sistemas de Bases de Datos Distribuidas - UCV
- **Título:** Arquitecturas de Sistemas de Bases de Datos
Url:
<http://alarcos.inf-cr.uclm.es/doc/aplicabbdd/Documentos/teoria/arquitecturas%20para%20bases%20de%20datos.pdf>
Fuente: DASGBD Curso 2007/2008
- **Título:** Conceptos de Extracto y Réplica
Url:
http://www.cyta.com.ar/elearn/bd/curso_archivos/bddistribuida.ppt
Fuente: cyta.com



- **Título:** Administrador de Bases de Datos
Url:
http://sistemas.itlp.edu.mx/tutoriales/basedat1/tema1_10.htm
Fuente: sistemas.itlp.edu
- **Título:** Bases de Datos Distribuidas
Url:
http://bd.eui.upm.es/DYOBD/DISTRIBUIDAS_texto.pdf
Fuente: Universidad Politécnica de Madrid
- **Título:** Base de Datos Distribuidas
Url:
<http://www.iessanvicente.com/colaboraciones/BBDDdistribuidas.pdf>
Fuente: I.E.S. San Vicente, de San Vicente del Raspeig, Alicante
- **Título:** Matria de Base de datos Distribuidas
Url:
http://base-de-datos0.tripod.com/unidad_3.htm
Fuente: INSTITUTO TECNOLÓGICO DE CULIACÁN
- **Título:** Bases de Datos Distribuidas
Url:
<http://alarcos.inf-cr.uclm.es/doc/bbddavanzadas/08-09/presentacionDistribucion1.PDF>
Fuente: Escuela Superior de Informática de Ciudad Real
- **Título:** Trabajo de Arquitectura Cliente / servidor
Url:
http://html.rincondelvago.com/bases-de-datos-distribuidas_2.html
Fuente: rincondelvago.com
- **Título:** Funciones del Administrador de Bases de datos
Url:
<http://www.angelfire.com/nf/tecvirtual/cursos/admonbd/DBA1.htm>
Fuente: angelfire.com
- **Título:** Arquitectura de Oracle
Url:
<http://www.scribd.com/doc/14269897/Estructura-de-Datos-Oracle-10G>
Fuente: scribd.com
- **Título:** Apuntes de Administración de Bases de datos
Url:
<http://www.monografias.com/trabajos19/administracion-base-datos/administracion-base-datos.shtml>
Fuente: monografias.com
- **Título:** Base de Datos Avanzada
Url:
<http://basedatosavanzadaselem.obolog.com/unidad-1-base-datos-distribuidas-523685>
Fuente: obolog.com
- **Título:** Arquitectura de Base de Datos Distribuida
Url:
http://www.cs.cinvestav.mx/SC/prof_personal/adiaz/Disdb/Cap_2.html
Fuente: Centro de Investigación y de Estudios Avanzados del IPN. Departamento de Computación
- **Título:** Replicación oracle
Url:
<http://vimeo.com/954517>
Fuente: vimeo.com



- **Título:** Iniciación a Oracle
Url:
http://www.emagister.com/manual/frame.cfm?id_centro=62420110061369504954524948484569&id_curso=1000003342&id_user=63154963109513951094652278081857&id_segmento=5&id_categ=118&id_búsqueda=2439854
Fuente: emagister.com
- **Título:** Replicación de Base de Datos con Oracle
Url:
http://www.scribd.com/doc/17721896/Replicar-Datos-en-Oracle-Usando-Vistas-Materializadas?secret_password=&autodown=docx
Fuente: scribd.com
- **Título:** Administración de Base de datos Oracle
Url:
http://www.emagister.com/manual/frame.cfm?id_centro=62420110061369504954524948484569&id_curso=1000003343&id_user=63154963109513951094652278081857&id_segmento=5&id_categ=118&id_búsqueda=2439856
Fuente: emagister.com
- **Título:**
Url: Introduction to Building Fusion Web Applications with Oracle ADF
http://download.oracle.com/docs/cd/E15051_01/web.1111/b31974/intro.htm
Fuente: download.oracle.com
- **Título:** Oracle Application Development Framework
Url:
http://www.cuore.es/otros/seminarios/seminario_adf.pdf
Fuente: cuore.es
- **Título:** El Patrón MVC
Url:
<http://oracleadffaces.blogspot.com/2006/11/un-poco-de-teora-el-patrn-mvc-model.html>
Fuente: blogspot.com
- **Título:** Oracle ADF Business Components
Url:
http://cafejojano.files.wordpress.com/2007/04/oracle_adf_bc.pdf
Fuente: wordpress.com

LIBROS

- Fundamentos de Bases de Datos, 4ª edición Silberschatz, A., Korth, H.F. y Sudarshan, S. McGraw-Hill. 2002
- Sistemas de bases de datos. Conceptos fundamentales 2ª edición Elmasri, R. y Navathe, S.B. Addison-Wesley. 1997
- George Coulouris. Sistemas Distribuidos. Tercera Edición. Addison Wesley. Madrid. 2001.
- Principles of Distributed Database Systems Ozsu, T. y Valduriez, P. Prentice Hall. 1991.
- Oracle Guía de aprendizaje Abbey, M., M.J. Corey e I. Abramson McGraw-Hill. 2000.
- Öszu, Tamar and Valduriez, P. Principles of Distributed Database Systems 2nd Ed. Prentice Hall, 1998.