



UNIVERSIDAD NACIONAL  
AUTÓNOMA DE  
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES  
CAMPUS ARAGÓN

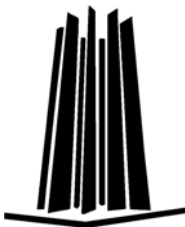
**“DISEÑO Y CONFIGURACIÓN DE SERVICIOS CON  
ALTA DISPONIBILIDAD PARA SISTEMAS DE  
INFORMACIÓN OPERATIVOS CON TECNOLOGÍA  
ORACLE”**

**T E S I S**  
QUE PARA OBTENER EL TÍTULO DE:  
**INGENIERO EN COMPUTACIÓN**

**P R E S E N T A:**

**OSWALDO VALENCIA LÓPEZ**

**ASESOR DE TESIS:**  
**Ing. JOSÉ AGUSTÍN VARGAS PADILLA**



**MÉXICO, 2008.**

## Agradecimientos

- A mi madre por guiarme como una regla mostrándome el camino de la igualdad y las leyes para que con inteligencia me conduzca por el camino de las buenas costumbres.
- A mi padre por mostrarme la regularidad, la constancia y el poder de la razón con el cual logré ser un hombre libre.
- A mi Hermana y Chatito por ayudarme a fortalecer mi voluntad como un martillo el cual he utilizado a lo largo de toda mi carrera profesional y personal.

---

---

Índice

1	Fundamentos de Sistemas Informáticos y Alta disponibilidad .....	7
1.1	Definición de Sistema de Información .....	7
1.1.1	Definición de Información .....	8
1.1.2	Definición de Sistema .....	8
1.1.3	Tipos de Sistemas de Información .....	8
1.1.3.1	Sistemas de Nivel Operativo .....	9
1.1.3.2	Sistemas de Nivel de Conocimientos .....	9
1.1.3.3	Sistemas de Nivel Gerencial .....	9
1.1.3.4	Sistemas de Nivel Estratégico.....	9
1.1.4	Evolución de los Sistemas de Información .....	10
1.1.4.1	Etapa de contagio o expansión.....	10
1.1.4.2	Etapa de control o formalización.....	11
1.1.4.3	Etapa de integración.....	11
1.1.4.4	Etapa de administración de datos .....	11
1.1.4.5	Etapa de madurez.....	12
1.1.5	Arquitectura de un Sistema de Información .....	12
1.1.5.1	Tecnología Cliente Servidor.....	13
1.1.5.1.1	Ventajas de la arquitectura cliente-servidor .....	13
1.1.5.1.2	Arquitectura con gradas.....	13
1.1.5.1.3	Ventajas .....	14
1.1.5.1.4	Desventajas .....	14
1.1.5.2	Sistemas Multicapa .....	14
1.1.5.2.1	Cliente .....	15
1.1.5.2.2	Interface .....	15
1.1.5.2.3	Lógica de Negocio .....	16
1.2	La empresa u Organización.....	16
1.2.1	Arquitectura de la empresa .....	16
1.2.1.1	Arquitectura de una Empresa u Organización .....	17
1.2.1.2	Definición técnica microeconómica de la organización .....	18
1.2.2	Niveles de Servicios de TI en una organización (ITIL).....	19
1.2.2.1	Gestión de Servicios TI.....	19
1.2.2.2	Soporte al Servicio.....	21
1.2.2.3	Organización, clientes y usuarios .....	21
1.2.2.3.1	Service Desk .....	21
1.2.2.3.2	Knowledge Base .....	21
1.2.2.3.3	Gestión de incidentes.....	22
1.2.2.3.4	Gestión de problemas.....	22
1.2.2.3.5	Gestión de Cambios.....	22
1.2.2.3.6	Gestión de Versiones .....	22
1.2.2.3.7	Gestión de configuraciones .....	22
1.2.2.4	Provisión del Servicio .....	22
1.2.2.5	Organización, clientes y usuarios .....	23
1.2.2.6	Gestión del nivel del servicio .....	23
1.2.2.6.1	CMBD (Configuration Managment Database) .....	23
1.2.2.6.2	Gestión de disponibilidad.....	23
1.2.2.6.3	Gestión de Capacidad.....	24
1.2.2.6.4	Gestión Financiera .....	24
1.2.2.6.5	Gestión de la continuidad del servicio .....	24

---

---

1.2.3	Errores Informáticos.....	24
1.2.3.1	Fallas en Sistemas de Software.....	24
1.3	Componentes Tecnológicos de un sistema de Información Actual.....	31
1.3.1	Aplicaciones.....	31
1.3.1.1	Servidores de Aplicaciones.....	31
1.3.1.1.1	Características comunes.....	31
1.3.1.1.2	Usos.....	31
1.3.1.1.3	Arquitectura de Servidor de Aplicaciones.....	31
1.3.1.1.4	MiddleWare.....	32
1.3.1.1.5	Categorías de Middleware.....	33
1.3.1.1.6	Servidores de aplicación J2EE.....	34
1.3.1.2	Servidor de Aplicaciones Oracle.....	34
1.3.1.2.1	Arquitectura.....	34
1.3.1.2.2	El modelo multicapa.....	34
1.3.1.3	Arquitectura funcional de Oracle Application Server 10g.....	37
1.3.1.3.1	Nivel Cliente.....	37
1.3.1.3.2	Nivel Web.....	37
1.3.1.3.3	Oracle http Server (OHS).....	37
1.3.1.3.4	Web Cache.....	38
1.3.1.3.5	Nivel Servidor de Aplicaciones.....	38
1.3.1.3.6	Nivel de Base de Datos.....	39
1.3.2	Bases de Datos.....	39
1.3.2.1	Sistema de gestión de base de datos (SGBD).....	40
1.3.2.2	Estructura de un sistema de base de datos.....	41
1.3.2.3	Estructura lógica de usuario (esquema externo).....	41
1.3.2.4	Estructura lógica global (esquema conceptual).....	41
1.3.2.5	Estrategia de almacenamiento.....	41
1.3.2.6	Tipos de usuarios.....	41
1.3.2.7	Historia de los sistemas de bases de datos.....	43
1.3.2.8	Base de Datos ORACLE.....	44
1.3.2.8.1	Componentes de ORACLE.....	45
1.3.2.8.1.1	Memoria SGA.....	45
1.3.2.8.1.1.1	Buffers de BD, Database Buffer Cache.....	46
1.3.2.8.1.1.2	Buffer Redo Log.....	46
1.3.2.8.1.1.3	Área de SQL Compartido, Shared SQL Pool.....	46
1.3.2.8.1.2	Estructuras de Proceso.....	47
1.3.2.8.1.3	System Monitor, SMON.....	47
1.3.2.8.1.4	Process Monitor, PMON.....	47
1.3.2.8.1.5	Database Writer, DBWR.....	47
1.3.2.8.1.6	Log Writer, LGWR.....	47
1.3.2.8.1.7	Checkpoint, CKPT.....	48
1.3.2.8.1.8	Archiver, ARCH.....	48
1.3.2.8.1.9	Los Espacios de Tablas ( <i>Tablespaces</i> ).....	48
1.3.2.8.1.10	Archivos (Archives).....	50
1.3.2.8.1.11	Estructuras Internas de la BD.....	50
1.3.2.8.1.11.1	Tablas y Columnas.....	50
1.3.2.8.1.11.2	Restricciones de Tablas.....	50
1.3.2.8.1.11.3	Usuarios.....	50
1.3.2.8.1.11.4	Esquemas.....	51
1.3.2.8.1.11.5	Índices.....	51



---

---

1.3.2.8.1.11.6	Clusters.....	51
1.3.2.8.1.11.7	Vistas.....	51
1.3.2.8.1.11.8	Secuencias.....	51
1.3.2.8.1.11.9	Procedimientos y Funciones.....	51
1.3.2.8.1.11.10	Paquetes, <i>Packages</i> .....	51
1.3.2.8.1.11.11	Disparadores, <i>Triggers</i> .....	51
1.3.2.8.1.11.12	Sinónimos.....	51
1.3.2.8.1.11.13	Privilegios y Roles.....	52
1.3.2.8.1.11.14	Segmentos, Extensiones y Bloques.....	52
1.3.2.8.1.11.15	Segmento de <i>Rollback</i> .....	52
1.3.2.8.1.12	Estructuras Externas.....	52
1.3.2.8.1.12.1	Archivos <i>redo log</i> .....	52
1.3.2.8.1.12.2	Archivos de control.....	53
1.3.2.8.1.12.3	El Archivo INITORA.....	54
1.3.2.8.1.12.4	Archivos de Traza.....	54
1.3.2.9	SQL Server.....	54
1.3.3	Sistema Operativo.....	57
1.3.3.1	¿Que es un sistema operativo?.....	57
1.3.3.2	Tipos de Sistemas Operativos.....	57
1.3.3.3	Sistemas Operativos por Servicios (Visión Externa).....	57
1.3.3.3.1	Por el número de Usuarios.....	58
1.3.3.3.1.1	Monousuario.....	58
1.3.3.3.1.2	Multiusuario.....	58
1.3.3.3.2	Por el Número de Tareas.....	58
1.3.3.3.2.1	Monotarea.....	58
1.3.3.3.2.2	Multitarea.....	59
1.3.3.3.3	Por el Número de Procesadores.....	59
1.3.3.3.3.1	Uniproceto.....	59
1.3.3.3.3.2	Multiproceto.....	59
1.3.3.4	Sistemas Operativos por su Estructura (Visión Interna).....	60
1.3.3.4.1	Estructura Monolítica.....	60
1.3.3.4.2	Estructura Jerárquica.....	61
1.3.3.5	Máquina Virtual.....	62
1.3.3.6	Cliente-Servidor (Microkernel).....	62
1.3.3.7	Sistemas Operativos por la Forma de Ofrecer sus Servicios.....	62
1.3.3.8	Sistema Operativo de Red.....	63
1.3.3.9	Sistemas Operativos Distribuidos.....	63
1.3.3.10	Unix.....	64
1.3.3.11	Linux.....	68
1.3.3.12	Red Hat Linux.....	70
1.3.3.13	Solaris.....	72
1.3.3.14	HP-UX.....	74
1.3.3.15	Windows.....	75
1.3.4	Hardware.....	81
1.3.4.1	Tipos de Hardware.....	81
1.3.4.2	Tipos de Procesadores.....	83
1.3.4.2.1	Uniprocetores.....	83
1.3.4.2.2	Multiprocetores.....	83
1.3.4.2.3	Symmetric Multi-Processing SMP.....	84
1.3.4.2.4	Massively paralell procesing (MPP).....	85

---

---

1.3.4.2.5	Non uniform memory access (NUMA)	85
1.3.4.3	Tipos de Servidores	86
1.3.4.4	Tipos de Almacenamiento	87
1.3.4.4.1	Dispositivo de almacenamiento	87
1.3.4.4.2	Unidades de información	88
1.3.4.4.3	Sistema binario	88
1.3.4.4.4	Medidas de almacenamiento de la información	89
1.3.4.4.5	Los sistemas de archivos	89
1.3.4.4.6	Discos Duros	90
1.3.4.4.6.1	Discos Duros IDE	90
1.3.4.4.7	Niveles de Arreglos RAID	91
1.3.4.4.7.1	Implementaciones de RAID	95
1.3.4.4.8	Discos Duros SCSI	95
1.3.4.4.8.1	Ultra-SCSI y Ultra-SCSI-2	95
1.3.4.4.8.2	Serial Storage Architecture	96
1.3.4.4.9	Fibre Channel	96
1.3.4.4.10	Storage area network	96
1.4	Alta Disponibilidad en sistemas de Información	98
1.4.1	Requerimientos de Negocios Modernos y Niveles de Alta Disponibilidad	98
1.4.1.1.1	Confiabilidad	98
1.4.1.1.2	Portabilidad	98
1.4.1.1.3	Recuperación	99
1.4.1.1.4	Seguridad	99
1.4.1.1.5	Revisión	99
1.4.1.1.6	Flexibilidad y Mantenimiento	99
1.4.1.1.7	Escalabilidad	99
1.4.1.1.8	Disponibilidad	100
1.4.1.1.9	Respuesta	101
1.4.2	Definición de Alta Disponibilidad	101
1.4.2.1.1	Cluster	102
1.4.2.1.1.1	Componentes de un Cluster	103
1.4.2.1.1.2	Nodos	103
1.4.2.1.1.3	Sistema Operativo	103
1.4.2.1.1.4	Conexiones de Red	103
1.4.2.1.1.5	Middleware	103
1.4.2.1.1.6	El concepto de cluster	104
1.4.2.1.1.7	Características de un cluster	104
1.4.2.1.1.8	Acoplamiento de un cluster	106
1.4.2.1.1.8.1	Acoplamiento fuerte	106
1.4.2.1.1.8.2	Acoplamiento medio	106
1.4.2.1.1.8.3	Acoplamiento débil	107
1.4.2.1.1.9	Esquema y otras características	107
1.4.2.1.1.10	Homogeneidad de un cluster	107
1.4.2.1.2	Clasificación según el servicio prioritario	108
1.4.2.1.3	Alto rendimiento (HP, high performance)	108
1.4.2.1.4	Alta disponibilidad (HA, high availability)	109
1.4.2.1.5	Alta confiabilidad (HR, high reliability)	109
1.4.2.1.6	Técnicas que utilizan	109
1.4.2.1.7	Clusters HA: alta disponibilidad	109

---

---

---

---

1.4.2.1.7.1	La misión .....	109
1.4.2.1.7.2	Problemas que solucionan.....	110
1.4.2.1.8	Como funciona un Cluster.....	110
1.4.2.1.9	Cluster Fail Over.....	111
1.4.2.2	Alta Disponibilidad en Servidor de Aplicaciones Oracle .....	111
1.4.2.2.1	Alta disponibilidad .....	111
1.4.2.2.2	Detección de procesos muertos y reinicio automático... ..	112
1.4.2.2.3	Clustering.....	112
1.4.2.2.4	Administración de Configuraciones.....	112
1.4.2.2.5	Balanceo de Cargas y Failover .....	113
1.4.2.2.6	Respaldos y Recuperación.....	113
1.4.2.2.7	Recuperación de desastres .....	113
1.4.2.3	Alta Disponibilidad en Bases de Datos .....	113
1.4.2.4	Real Application Cluster (RAC).....	113
1.4.2.4.1	Cache Fusion .....	114
1.4.2.4.2	Sequence Generators.....	114
1.4.2.4.3	System Change Number (SCN).....	114
1.4.2.4.4	FailOver .....	114
1.4.2.4.5	Scalability .....	114
1.4.2.4.6	Procesos de RAC .....	115
1.4.2.4.6.1	Global Cache Service Monitor (LMS).....	117
1.4.2.4.6.2	Global Enqueue Service monitor (LMON).....	117
1.4.2.4.6.3	Global Enqueue Service Daemon (LMD) .....	117
1.4.2.4.6.4	Global Service Daemon (GSD).....	117
1.4.2.4.6.5	Lock Process (LCK).....	117
1.4.2.4.6.6	Diagnosability daemon (DIAG) .....	117
1.4.2.4.6.7	Cache Fusion.....	117
1.4.2.4.6.8	Global Resource Directory (GRD).....	118
1.4.2.4.6.9	Lock Structure .....	118
1.4.2.4.6.10	Global Cache Management (CGS).....	118
1.4.2.4.6.11	Mastering of resources.....	118
1.4.2.4.6.12	Failover.....	118
1.4.2.4.7	Archivos utilizados por Oracle RAC .....	119
1.4.2.4.7.1	<i>Parameter file</i> .....	119
1.4.2.4.7.2	Data files .....	119
1.4.2.4.7.3	Control files .....	119
1.4.2.4.7.4	Online redo log files.....	120
1.4.2.4.7.5	Archived redo log files.....	120
1.4.2.5	Tecnología Grid.....	120
1.4.2.5.1	Sharing.....	121
1.4.2.5.2	Servicios .....	122
1.4.2.5.3	Funcionamiento de la computación grid.....	122
1.4.2.5.4	Ventajas y desventajas y aplicaciones de la Computación Grid	124
1.4.2.5.5	Aplicaciones de la Computación Grid.....	124
1.4.2.5.6	Súper computación distribuida .....	124
1.4.2.5.7	Sistemas distribuidos en tiempo real.....	124
1.4.2.5.8	Servicios puntuales.....	125
1.4.2.5.9	Proceso intensivo de datos .....	125
1.4.2.5.10	Entornos virtuales de colaboración.....	125

---

---

1.4.2.5.11	Algunos Proyectos en desarrollo bajo el sistema Data Grid	125
1.4.2.5.11.1	Globus Project.....	125
1.4.2.5.11.2	TeraGrid .....	126
1.4.2.5.11.3	CrossGrid .....	126
1.4.2.5.11.4	OpenMolGRID .....	126
1.4.2.5.11.5	UK e-Science.....	127
1.4.2.5.11.6	EGEE (Enabling Grids for e-Science in Europe) .....	127
1.4.2.5.12	Arquitectura grid.....	128
1.4.2.5.12.1	Capa de Infraestructura (Fabric) .....	128
1.4.2.5.12.2	Recursos computacionales .....	128
1.4.2.5.12.3	Recursos alojados .....	128
1.4.2.5.12.4	Recursos de red.....	128
1.4.2.5.12.5	Repositorios de recursos.....	129
1.4.2.5.12.6	Catálogos.....	129
1.4.2.5.12.7	Capa de Conectividad (Connectivity).....	129
1.4.2.5.12.8	Capa de Recurso (Resource) .....	129
1.4.2.5.12.9	Protocolo de información.....	129
1.4.2.5.12.10	Protocolos de manejo .....	130
1.4.2.5.12.11	Capa Recursos o Colectiva (Collective) .....	130
1.4.2.5.12.12	Capa de Aplicación (Application).....	131
1.4.2.5.13	Oracle GRID .....	132
1.4.2.5.13.1	Almacenamiento de la información .....	133
1.4.2.5.13.2	Utilización Óptima de los Recursos .....	134
1.4.2.5.13.3	Máximo Nivel de Servicios.....	134
1.4.2.5.13.4	Facilidad en la administración .....	135
1.4.2.5.13.5	Desarrollo de Aplicaciones.....	136
1.4.2.5.13.6	Datawarehousing.....	136
2	Planteamiento de Caso Práctico .....	137
2.1	Descripción de la empresa .....	137
2.2	Descripción del sistema .....	138
2.3	Requerimientos Tecnológicos de la Empresa.....	140
3	Análisis de Caso Práctico .....	141
3.1	Arquitectura Tecnología de la Empresa.....	141
3.2	Niveles de Servicio.....	142
3.3	Aplicaciones Empresariales.....	142
4	Propuesta de Diseño.....	143
4.1	Arquitectura Tecnológica de la empresa .....	143
4.2	Niveles de Servicio.....	144
4.3	Aplicaciones Empresariales.....	145
5	Implementación en Ambiente Corporativo.....	145
5.1	Implementación de Oracle RAC.....	145
5.2	Implementación de Oracle Application Server en Cluster .....	145
6	Conclusión.....	146
7	Bibliografía .....	147

## 1 Fundamentos de Sistemas Informáticos y Alta disponibilidad

### 1.1 Definición de Sistema de Información

La información es uno de los principales tipos de recursos con que cuenta una empresa. La información puede manejarse igual que cualquier otro recurso, el interés en este tema se debe a dos influencias. La primera es que los negocios se han vuelto más complejos y la segunda que las capacidades de las computadoras han aumentado.

La información que las computadoras producen es de utilidad para las empresas, gerentes, usuarios y otro tipo de organizaciones relacionadas con la compañía, así como en todas sus áreas funcionales.

Es un recurso crítico de las organizaciones, tan fundamental como la energía o las máquinas. Es el eslabón indispensable que une a todos los componentes de la organización para una mejor operación y coordinación pero con una gran importancia, tener una supervivencia en un ambiente competitivo. En general se puede pensar que una compañía no podría funcionar sin información.

Un sistema de información es un conjunto de elementos que interactúan entre si con el fin de apoyar a las actividades de una empresa o negocio.

Los componentes que forman todo un sistema de información son:

- **Herramientas tecnológicas:**
  - Hardware: Equipo físico empleado para la alimentación el procesamiento y la salida en un sistema de información.
  - Software: Instrucciones detalladas, previamente programadas que controlan y coordinan los componentes del hardware.
  - Tecnología de Almacenamiento: Medios Físicos y lógicos que gobiernan el almacenamiento y la organización de la información en un sistema de información.
  - Tecnología de telecomunicaciones: Dispositivos lógicos y software que enlazan diversos componentes de hardware de computadora y que transfieren la información de un lugar a otro.
- **Recursos Humanos:** Es el personal que interactúa con el sistema de información.
- **Procedimientos :** Logran la automatización de procesos operativos dentro de una organización, son llamados frecuentemente Sistemas Transaccionales, ya que su función primordial consiste en procesar transacciones tales como pagos, cobros, pólizas, entradas, salidas, etc. Por otra parte, los sistemas de Información que apoyan el proceso de toma de decisiones son los sistemas de Soporte a la Toma de Decisiones, sistemas para la Toma de Decisión de Grupo, sistemas Expertos de Soporte a la Toma de Decisiones y Sistema de Información para Ejecutivos. El tercer tipo de sistema, de acuerdo con su uso u objetivos, es el de los sistemas Estratégicos, los cuales se desarrollan en las organizaciones con el fin de lograr ventajas competitivas, a través del uso de la tecnología de información.

Se puede definir técnicamente como un conjunto de componentes interrelacionados que permiten capturar, procesar, almacenar y distribuir la información para apoyar la toma de decisiones y el control en una institución. Ayuda a los administradores y al personal a analizar problemas, visualizar y resolver problemas más complejos así como crear nuevos productos.

Un sistema de información realiza cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información.

- **Entrada de Información:** Es el proceso mediante el cual el Sistema de Información toma los datos que requiere para procesar la información. Las entradas pueden ser manuales o automáticas. Las manuales son aquellas que se proporcionan en forma directa por el usuario, mientras que las automáticas son datos o información que provienen o son tomados de otros sistemas o módulos, los cuales se denominan interfases automáticas.  
Las unidades típicas de entrada de datos a las computadoras son las terminales, las cintas magnéticas, las unidades de diskette, los códigos de barras, los escáners, la voz, los monitores sensibles al tacto, el teclado y el mouse, entre otras.
- **Almacenamiento de información:** El almacenamiento es una de las actividades o capacidades más importantes que tiene una computadora, ya que a través de esta propiedad el sistema puede recordar la información guardada en la sección o proceso anterior. Esta información suele ser almacenada en estructuras de información denominadas archivos. Las unidades típicas de almacenamiento son los discos magnéticos o discos duros, los discos flexibles o diskettes y los discos compactos.
- **Procesamiento de Información:** Es la capacidad del Sistema de Información para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecida. Estos cálculos pueden efectuarse con datos introducidos recientemente en el sistema o bien con datos que están almacenados. Esta característica de los sistemas permite la transformación de datos fuente en información que puede ser utilizada para la toma de decisiones, lo que hace posible, entre otras cosas, que un tomador de decisiones genere una proyección financiera a partir de los datos que contiene un estado de resultados o un balance general de un año base.
- **Salida de Información:** La salida es la capacidad de un Sistema de Información para sacar la información procesada o bien datos de entrada al exterior. Las unidades típicas de salida son las impresoras, terminales, diskettes, cintas magnéticas, la voz, los graficadores los plotters, entre otros. Es importante aclarar que la salida de un Sistema de Información puede constituir la entrada a otro Sistema de Información o módulo. En este caso, también existe una interfase automática de salida. Por ejemplo, un Sistema de Control de Clientes tiene una interfase automática de salida con el Sistema de Contabilidad, ya que genera las pólizas contables de los clientes.

#### 1.1.1 Definición de Información

La información la componen datos que se han colocado en un contexto significativo y útil, se ha comunicado a un receptor, quien a su vez la utiliza para tomar decisiones. La información implica la comunicación y recepción de inteligencia o conocimiento, evalúa y notifica, sorprende y estimula, reduce incertidumbre y revela alternativas a una solución.

#### 1.1.2 Definición de Sistema

Un sistema es un grupo de elementos que se integran con el propósito común de lograr un objetivo en una organización como una compañía o un área funcional.

#### 1.1.3 Tipos de Sistemas de Información

Las instituciones y los sistemas de información pueden dividirse en niveles estratégicos, administrativos, de conocimientos y operativos. Luego pueden dividirse en cinco áreas funcionales: ventas y mercadotecnia, manufactura, finanzas, contabilidad y recursos humanos. Los sistemas de información sirven a cada uno de los niveles y funciones. Los sistemas de nivel estratégico ayudan a los directivos con la planeación a largo plazo; los de nivel administrativo ayudan a los gerentes medios en el seguimiento y control; los de nivel de conocimiento ayudan a los trabajadores del conocimiento e información a diseñar productos, distribuir información y al manejo de documentación; los de nivel operativo ayudan a los gerentes operativos al seguimiento de las actividades diarias.

### 1.1.3.1 Sistemas de Nivel Operativo

Apoyan a los gerentes operativos al hacer el seguimiento de las actividades y transacciones elementales de la institución, como ventas, recepción de materiales, depósitos en efectivo, nominas, decisiones de crédito y el flujo de materiales de planta. El fin principal del sistema a este nivel es responder a la cuestiones de rutina y seguir el flujo de transacciones a lo largo de la institución.

### 1.1.3.2 Sistemas de Nivel de Conocimientos

Apoyan a los trabajadores del conocimiento y a los de la información en una institución. La ventana sobre organización ilustra como los sistemas de nivel de conocimiento pueden mejorar la productividad de los ingenieros y diseñadores. La finalidad de estos sistemas es ayudar a la empresa de negocios a integrar nuevos conocimientos para el negocio y para que la institución controle el flujo de la documentación. Estos sistemas, en especial bajo la forma de estaciones de trabajo y sistemas de oficina, constituyen las aplicaciones de mayor crecimiento en la actualidad en los negocios.

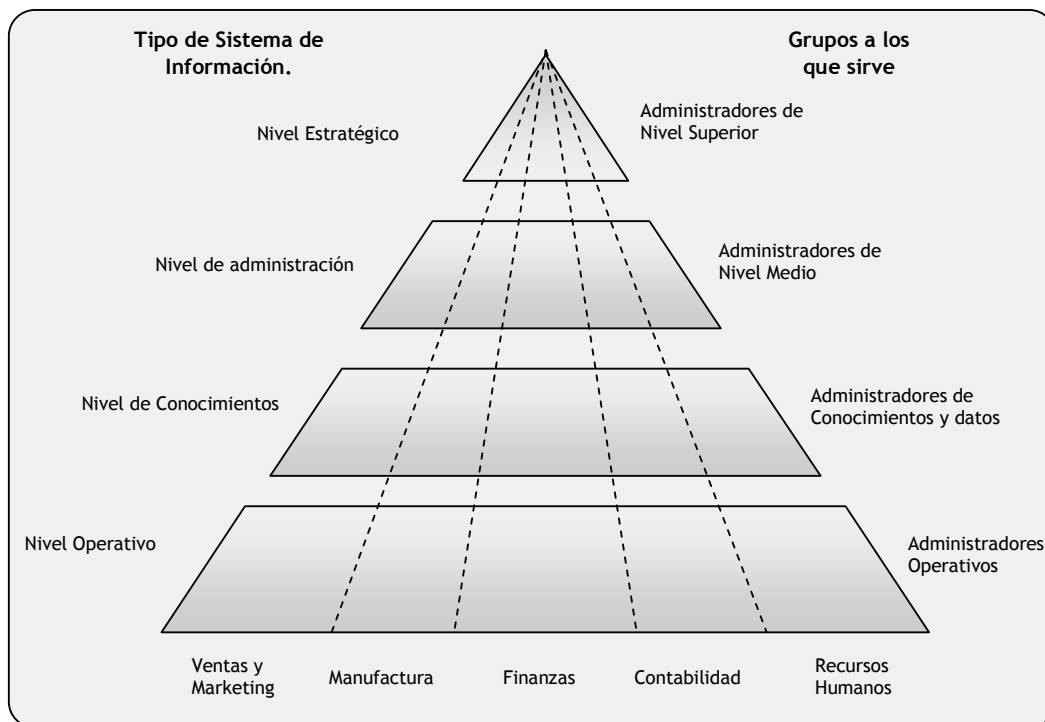
### 1.1.3.3 Sistemas de Nivel Gerencial

Se diseñan para las actividades de seguimiento, control, toma de decisiones y actividades administrativas de los administradores de nivel medio. Estos sistemas comparan los resultados del trabajo del día con los del mes o el año anterior. En general proporcionan reportes periódicos en vez de información instantánea de información periódica.

### 1.1.3.4 Sistemas de Nivel Estratégico

Ayudan a los niveles directivos a atacar y dirigir las cuestiones estratégicas y las tendencias a largo plazo dentro y en el entorno de la institución. Su interés principal es hacer frente a los cambios que ocurren en el entorno con las capacidades con las que se cuentan.

Figura 1: Tipos de Sistemas de Información.



#### 1.1.4 Evolución de los Sistemas de Información

En la década de los setenta, Richard Nolan, un conocido autor y profesor de la Escuela de Negocios de Harvard, desarrolló una teoría que impactó el proceso de planeación de los recursos y las actividades de la informática.

Según Nolan, la función de la Informática en las organizaciones evoluciona a través de ciertas etapas de crecimiento, las cuales se explican a continuación:

- Comienza con la adquisición de la primera computadora y normalmente se justifica por el ahorro de mano de obra y el exceso de papeles.
- Las aplicaciones típicas que se implantan son los Sistemas Transaccionales tales como nóminas o contabilidad.
- El pequeño Departamento de Sistemas depende en la mayoría de los casos del área de contabilidad.
- El tipo de administración empleada es escaso y la función de los sistemas suele ser manejada por un administrador que no posee una preparación formal en el área de computación.
- El personal que labora en este pequeño departamento consta a lo sumo de un operador y/o un programador. Este último podrá estar bajo el régimen de honorarios, o bien, puede recibirse el soporte de algún fabricante local de programas de aplicación.
- En esta etapa es importante estar consciente de la resistencia al cambio del personal y usuario (ciberfobia) que están involucrados en los primeros sistemas que se desarrollan, ya que estos sistemas son importantes en el ahorro de mano de obra.
- Esta etapa termina con la implantación exitosa del primer Sistema de Información. Cabe recalcar que algunas organizaciones pueden vivir varias etapas de inicio en las que la resistencia al cambio por parte de los primeros usuarios involucrados aborta el intento de introducir la computadora a la empresa.

##### 1.1.4.1 Etapa de contagio o expansión

Los aspectos sobresalientes que permiten diagnosticar rápidamente que una empresa se encuentra en esta etapa son:

- Implantación exitosa del primer Sistema de Información en la organización. Como consecuencia de lo anterior, el primer ejecutivo usuario se transforma en el paradigma o persona que se habrá de imitar.
- Las aplicaciones que con frecuencia se implantan en esta etapa son el resto de los Sistemas Transaccionales no desarrollados en la etapa de inicio, tales como facturación, inventarios, control de pedidos de clientes y proveedores, cheques, etc.
- El pequeño departamento es promovido a una categoría superior, donde depende de la Gerencia Administrativa o Contraloría.
- El tipo de administración empleado está orientado hacia la venta de aplicaciones a todos los usuarios de la organización; en este punto suele contratarse a un especialista de la función con preparación académica en el área de sistemas.
- Contratación de personal especializado y nacen puestos tales como analista de sistemas, analista-programador, programador de sistemas, jefe de desarrollo, jefe de soporte técnico, etc.
- Las aplicaciones desarrolladas carecen de interfases automáticas entre ellas, de tal forma que las salidas que produce un sistema se tienen que alimentar en forma manual a otro sistema, con la consecuente irritación de los usuarios.

Los gastos por concepto de sistemas empiezan a crecer en forma importante, lo que marca la pauta para iniciar la racionalización en el uso de los recursos computacionales dentro de la empresa. Este problema y el inicio de su solución marcan el paso a la siguiente etapa.



#### **1.1.4.2 Etapa de control o formalización**

Para identificar a una empresa que transita por esta etapa es necesario considerar los siguientes elementos:

- Esta etapa de evolución de la informática dentro de las empresas se inicia con la necesidad de controlar el uso de los recursos computacionales a través de las técnicas de presupuestos desde cero (partiendo de que no se tienen nada) y la implantación de sistemas de cargos a usuarios (por el servicio que se presta).
- Las aplicaciones están orientadas a facilitar el control de las operaciones del negocio para hacerlas más eficaces, tales como sistemas para control de flujo de fondos, control de órdenes de compra a proveedores, control de inventarios, control y manejo de proyectos, etc.
- El departamento de sistemas de la empresa suele ubicarse en una posición gerencial, dependiendo del organigrama de la Dirección de Administración o Finanzas.
- El tipo de administración empleado dentro del área de Informática se orienta al control administrativo y a la justificación económica de las aplicaciones a desarrollar. Nace la necesidad de establecer criterios para las prioridades en el desarrollo de nuevas aplicaciones. La cartera de aplicaciones pendientes por desarrollar empieza a crecer.
- En esta etapa se inician el desarrollo y la implantación de estándares de trabajo dentro del departamento, tales como: estándares de documentación, control de proyectos, desarrollo y diseño de sistemas, auditoría de sistemas y programación.
- Se integra a la organización del departamento de sistemas, personal con habilidades administrativas y preparadas técnicamente.

#### **1.1.4.3 Etapa de integración**

Las características de esta etapa son las siguientes:

- La integración de los datos y de los sistemas surge como un resultado directo de la centralización del departamento de sistemas bajo una sola estructura administrativa.
- Las nuevas tecnologías relacionadas con bases de datos, sistemas administradores de bases de datos y lenguajes de cuarta generación, hicieron posible la integración.
- En esta etapa surge la primera hoja electrónica de cálculo comercial y los usuarios inician haciendo sus propias aplicaciones. Esta herramienta ayudó mucho a que los usuarios hicieran su propio trabajo y no tuvieran que esperar a que sus propuestas de sistemas fueran cumplidas.
- El costo del equipo y del software disminuyó por lo cual estuvo al alcance de más usuarios.
- En forma paralela a los cambios tecnológicos, cambió el rol del usuario y del departamento de Sistemas de Información. El departamento de sistemas evolucionó hacia una estructura descentralizada, permitiendo al usuario utilizar herramientas para el desarrollo de sistemas.
- Los usuarios y el departamento de sistemas iniciaron el desarrollo de nuevos sistemas, reemplazando a los antiguos, en beneficio de la organización.

#### **1.1.4.4 Etapa de administración de datos**

Entre las características que destacan en esta etapa están las siguientes:

- El departamento de Sistemas de Información reconoce que la información es un recurso muy valioso que debe estar accesible para todos los usuarios.
- Para poder cumplir con lo anterior resulta necesario administrar los datos en forma apropiada, es decir, almacenarlos y mantenerlos en forma adecuada para que los usuarios puedan utilizar y compartir este recurso.
- El usuario de la información adquiere la responsabilidad de la integridad de la misma y debe manejar niveles de acceso diferentes.

#### 1.1.4.5 Etapa de madurez

Entre los aspectos sobresalientes que indican que una empresa se encuentra en esta etapa, se incluyen los siguientes:

- Al llegar a esta etapa, la Informática dentro de la organización se encuentra definida como una función básica y se ubica en los primeros niveles del organigrama.
- Los sistemas que se desarrollan son: Sistemas de Manufactura Integrados por Computadora, Sistemas Basados en el Conocimiento y Sistemas Expertos, Sistemas de Soporte a las Decisiones, Sistemas Estratégicos y, en general, aplicaciones que proporcionan información para las decisiones de alta administración y aplicaciones de carácter estratégico.

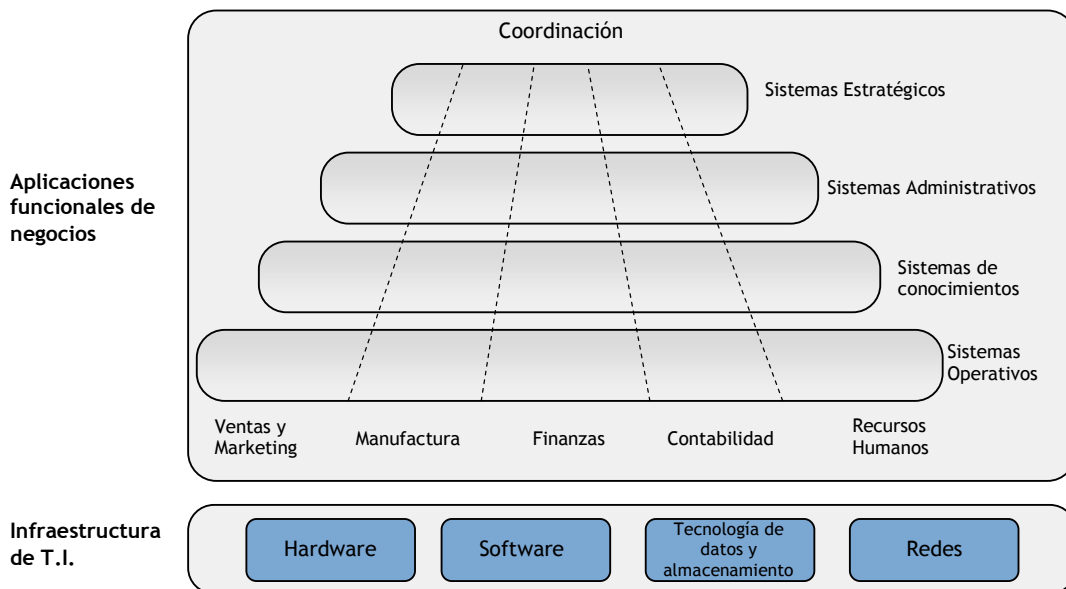
En esta etapa se tienen las aplicaciones desarrolladas en la tecnología de base de datos y se logra la integración de redes de comunicaciones con terminales en lugares remotos, a través del uso de recursos computacionales.

#### 1.1.5 Arquitectura de un Sistema de Información

La arquitectura de información es la forma específica que la tecnología de información adopta en una organización para lograr metas o realizar funciones selectas. Se trata de un diseño para los sistemas de aplicación de negocios que sirven a cada especialidad funcional y nivel de la organización, y la forma específica en que cada organización los usa. La siguiente figura muestra los principales elementos de la arquitectura de información que los administradores necesitan desarrollar.

Ya que los administradores y empleados interactúan directamente con estos sistemas, es vital para el éxito de la organización que su arquitectura de información satisfaga las necesidades del negocio ahora y en el futuro.

Figura 2: Arquitectura de Información de la Organización



La plataforma tecnológica para esta arquitectura se denomina infraestructura de TI, y consiste en el software y el hardware, tecnología de datos y almacenamiento, redes y recursos de TI, compartidos de la compañía y están disponibles para todas sus aplicaciones. Aunque esta plataforma tecnológica por lo regular es operada por personal técnico, la gerencia general debe decidir como repartir los recursos que tiene asignados a hardware, software, almacenamiento de datos y redes de telecomunicaciones, con el fin de efectuar inversiones sensatas en tecnología de información.

### 1.1.5.1 Tecnología Cliente Servidor

Esta arquitectura consiste básicamente en que un programa, el cliente informático realiza peticiones a otro programa en el servidor el cual les da respuesta.

Aunque esta idea se puede aplicar a programas que se ejecutan sobre una sola computadora es más ventajosa en un sistema multiusuario distribuido a través de una red de computadoras. En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa.

Una disposición muy común son los sistemas multicapa en los que el servidor se descompone en diferentes programas que pueden ser ejecutados por diferentes computadoras aumentando así el grado de distribución del sistema.

#### 1.1.5.1.1 Ventajas de la arquitectura cliente-servidor

- Centralización del control: los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema.
- Escalabilidad: se puede aumentar la capacidad de clientes y servidores por separado.
- Se reduce el tráfico de red considerablemente. Idealmente, el cliente se comunica con el servidor utilizando un protocolo de alto nivel de abstracción como por ejemplo SQL

Típicamente, un único servidor atiende a una multitud de clientes, ahorrando a cada uno de ellos el problema de tener la información instalada y almacenada localmente. Los sistemas Cliente-Servidor pueden ser de muchos tipos, dependiendo principalmente de las aplicaciones instaladas en el propio servidor.

Entre otros, existen: servidores de impresión mediante los cuales los usuarios comparten impresoras, servidores de archivos con los que los clientes comparten discos duros, servidores de bases de datos donde existe una única base de datos que es consultada por los clientes y puede o no ser modificada por ellos y servidores Web que utilizan también la tecnología Cliente/Servidor, aunque añaden aspectos nuevos y propios a la misma.

#### 1.1.5.1.2 Arquitectura con gradas

Una arquitectura genérica del cliente/servidor tiene dos tipos de nodos en la red: clientes y servidores. Consecuentemente, estas arquitecturas genéricas se refieren a veces como arquitecturas "de dos niveles".

Algunas redes pueden tener tres diversas clases de nodos: cliente, servidores del uso de datos de proceso para los clientes, y servidores de la base de datos que almacenan los datos para los servidores de uso. A esta configuración se le llama arquitectura de tres gradas.

La ventaja de una arquitectura de n-grada comparada con una arquitectura de dos niveles es que separa los procesos. Esto ocurre para mejorar el balance de la carga en los diversos servidores, por tanto tiene mayor escalabilidad. Las desventajas de las arquitecturas de la n-grada son:

- 1.- Crea más carga en la red
- 2.-Es mucho más difícil programar y probar software que en arquitectura de dos niveles porque más dispositivos tienen que comunicarse para terminar la transacción de un usuario

### 1.1.5.1.3 Ventajas

- Todos los datos se almacenan en los servidores, así que tienen mayor capacidad de control de seguridad. El servidor puede controlar el acceso y el recurso al cerciorarse que dejó solamente esos accesos de usuarios permitidos y cambiar datos.
- Es más flexible que el paradigma del P2P para poner al día los datos u otros recursos.
- Existen tecnologías maduras y diseñadas para el paradigma de C/S que certifica la seguridad al usuario que no entiende la interfaz así como facilitar su empleo.
- Cualquier elemento de la red C/S puede ser aumentado fácilmente.

### 1.1.5.1.4 Desventajas

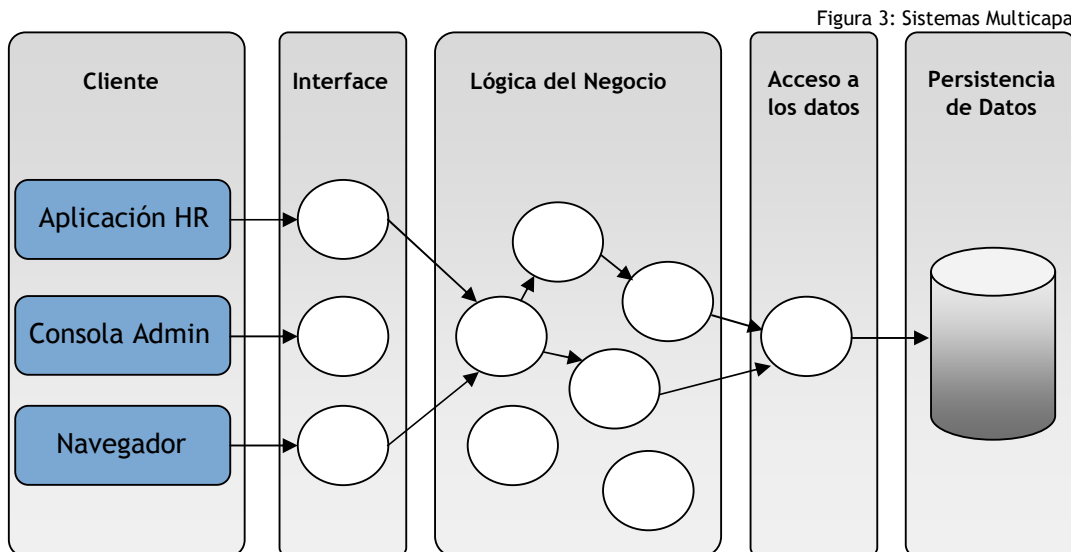
La congestión de tráfico en el paradigma de C/S, ha sido un problema desde su nacimiento, ya que pueden causar congestiones al servidor. Cuando una gran cantidad de clientes envían peticiones al mismo servidor al mismo tiempo, a más clientes, mayor congestión. El ancho de banda de una red P2P se compone de un conjunto de nodos incluidos en al red y por lo tanto a al tener un mayor número de nodos el ancho de banda es mayor.

El paradigma de C/S no tiene una buena robustez como red del P2P [1]. Cuando el servidor está apagado, las peticiones de los clientes no pueden ser satisfechas. En la mayor parte de redes del P2P, los recursos están situados generalmente en nodos por todas partes de la red, aunque algunos nodos salen o abandonan al descargar; otros nodos pueden continuar descargando, consiguiendo datos del resto de los nodos en la red.

El software y el hardware de un servidor son generalmente muy limitados. EL hardware utilizado en una computadora personal puede ser inútil sobre cierta cantidad de clientes, mientras que una edición casera de Windows XP, incluso sin tener IIS, puede trabajar como servidor. Dependiendo de las necesidades del cliente se deberá definir el hardware y el software correcto para una completa satisfacción, entendienddo que afectaría notoriamente el costo.

### 1.1.5.2 Sistemas Multicapa

La Arquitectura Cliente-Servidor con múltiples capas divide a una aplicación en las siguientes capas.



1. Peer to Peer (Punto a Punto). Sistema de red basado en el protocolo TCP/IP utilizado para el intercambio de ficheros entre usuarios de la red.

#### **1.1.5.2.1 Cliente**

Esta capa contiene componentes de fachada, controlador y validación de datos

El objetivo de esta capa es crear un entorno de desarrollo de Diseño Gráfico que no tenga visibilidad con la lógica de negocio ni con la persistencia de datos. Esto conlleva grandes ventajas, por ejemplo:

Optimización de Recursos Humanos

- Los desarrolladores especializados en crear interfaces gráficas para las Aplicaciones Cliente pueden modificar su código con total libertad
- No tienen llamadas a Objetos de otras capas que les estorben como conversaciones de bajo nivel con Bases de Datos o llamadas a Objetos de lógica de negocio

#### **1.1.5.2.2 Interface**

Esta capa contiene componentes de fachada, controlador y validación de datos

El objetivo de esta capa es crear un punto de unión entre la Capa de Cliente y la Capa de Lógica de negocio. Esto conlleva grandes ventajas

Optimización en la validación de los datos

- Los datos introducidos por un usuario desde cualquier tipo de Aplicación Cliente son validados en un punto localizado
  - Si los datos introducidos por el usuario son correctos entonces la Capa de Lógica de Negocio se pone en funcionamiento, si por el contrario hay algún dato de entrada incorrecto la Lógica de negocio no es ejecutada inútilmente
  - El punto anterior conlleva a que una vez que el código se está ejecutando en la Capa de Lógica de negocio no tenemos que validar una y otra vez los mismos datos porque estos ya fueron validados en la Capa de Presentación

Optimización de Recursos Humanos

- Si la Aplicación Cliente es un Navegador y no validamos los datos en la Capa de presentación
  - Le estamos dando a los diseñadores de páginas Web un trabajo extra más allá del diseño de la página Web

Evolución de las Aplicaciones

- En ocasiones hay partes de la lógica de negocio que cambian, pero no siempre este cambio conlleva la modificación de la Capa Cliente y viceversa
  - En estos casos sólo se tienen que realizar los cambios pertinentes en la Capa de Lógica de negocio o en la Capa de Cliente y ajustar los puntos de sutura entre ambas capas
- Si no hay Objetos que realicen funciones de fachada y de controlador en la Capa de Presentación, sucede que los Objetos de la Capa de Presentación están demasiado unidos a los Objetos de la Capa de Lógica de negocio
  - Esto provoca un efecto dominó porque cualquier cambio en cualquiera de las dos capas afecta directamente a la otra

### **1.1.5.2.3 Lógica de Negocio**

Esta capa es el corazón de la Aplicación

El objetivo de esta capa es que toda la lógica de negocio de la aplicación esté bien localizada y no mezclada con otros Objetos de otras capas. Esto conlleva grandes ventajas.

Evitamos que la incorporación de una nueva Aplicación Cliente no sea traumática fomentando la reutilización de código

- Por ejemplo una entidad financiera ofrece ventas de entradas a espectáculos a través de sus ATMs (cajeros automáticos) a través de una aplicación cliente de ventanas
  - Cuando posteriormente la entidad financiera decide ofrecer la venta de entradas por Internet a través de Navegadores el Objeto de la lógica de negocio que gestiona la entrada seguirá siendo el mismo (como la lógica de negocio no está mezclada con la Capa de cliente, no tenemos que volver a escribir la lógica de negocio). A esta práctica se le llama reutilización de código)
  - Si más adelante la entidad financiera decide ofrecer la venta de entradas a través de teléfonos celulares el Objeto de la lógica de negocio que gestiona la entrada seguirá siendo el mismo

## 1.2 La empresa u Organización

### 1.2.1 Arquitectura de la empresa

Para poder entender la arquitectura de una empresa es necesario entender que es una organización.

Una organización es una estructura social formal estable que toma recursos del entorno y los procesa para producir salidas. Esta definición técnica se concentra en diferentes elementos de la organización. El capital y la mano de obra son factores de producción primarios proporcionados por el entorno. La organización (empresa) transforma estas entradas en productos y servicios con una función de producción: un proceso que transforma capital y mano de obra en un producto. Los entornos consumen los productos y servicios a cambio de las entradas que proporcionan.

Una organización es más estable que un grupo informal, en términos de longevidad y de lo rutinario de sus actividades. Las organizaciones son formales porque son entidades legales y deben respetar las leyes; además, tienen reglas y procedimientos internos. Las organizaciones son estructuras sociales porque son conjuntos de elementos sociales, de forma análoga a la estructura que tiene una máquina (un acomodo específico de válvulas, levas, ejes y otras piezas).

Esta definición de organización es sencilla y amplia, pero no la describe ni predice en forma exacta en relación al mundo real al que pertenece la mayoría de la gente. Una definición conductual más realista de una organización es que se trata de un conjunto de derechos, privilegios, obligaciones y responsabilidades que alcanzan un delicado equilibrio con el paso del tiempo, gracias a los conflictos y la resolución de los mismos.

En esta perspectiva conductual de la compañía, la gente que labora en una organización desarrolla formas acostumbradas de trabajar, se apega a relaciones existentes y establece acuerdos con sus subordinados y superiores en cuanto a cómo se efectuará el trabajo, qué tanto trabajo se efectuará y en qué condiciones. Casi ninguno de estos acuerdos y sentimientos se menciona en un reglamento formal.

Figura 4: Arquitectura de la organización

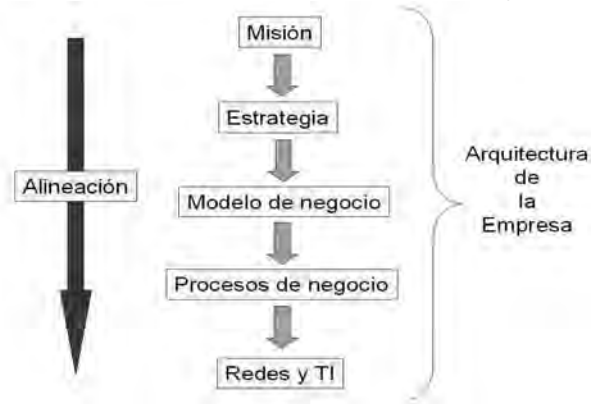


#### 1.2.1.1 Arquitectura de una Empresa u Organización

Es el conjunto de elementos organizacionales (objetivos estratégicos, departamentos, procesos, tecnología, personal, etc.) que describen a la empresa y se relacionan entre sí garantizando la alineación desde los niveles más altos (estratégicos) hasta los más bajos (operativos), con el fin de optimizar la generación de productos y servicios que conforman la propuesta de valor entregada a los clientes.

De esta definición destaca el hecho de que se busca una alineación de los niveles más altos con los más bajos de la empresa. Esto es importante, debido a que todas las áreas de la empresa deben actuar en armonía para conseguir los objetivos definidos por la misma. Esto suena muy obvio, pero en la práctica es frecuente perder este enfoque. La Arquitectura de la Empresa ayuda a conservar la perspectiva y a garantizar esta alineación. Los diferentes niveles que se tienen que modelar en la arquitectura se explican a continuación:

Figura 5: Arquitectura de la empresa



**Misión.** Este es el nivel más alto y explica por qué existe la empresa.

**Estrategia.** En este nivel se define qué es lo que se tiene que hacer para cumplir con la misión de la empresa. Se revisa la empresa y su entorno externo (es frecuente utilizar técnicas como el análisis SWOT<sup>[2]</sup>) para decidir cual es el mejor camino para maximizar las ganancias dados sus recursos actuales y la posición de la competencia. Es importante tener una visión de largo plazo para no caer en la trampa de ganancias a corto plazo que comprometa el futuro de la compañía.

*2. Análisis Swot (Strengths, Weakneses, Oportunities, Threatens) o Foda (Fortalezas, Oportunidades, Debilidades y Amenazas), es una herramienta estratégica que se utiliza para conocer la situación presente de una empresa. Identifica las amenazas y oportunidades que surgen del ambiente y las fortalezas y debilidades internas de la organización (como pie de página o cursivas y sangría)*

**Modelo de negocio.** Este nivel actúa como etapa de conexión entre el nivel Estratégico y el de Procesos de negocio (sin este nivel, la transición entre uno y otro es más difícil debido a la ambigüedad que genera el gran paso en el nivel de abstracción). Se explica cómo es que la empresa va a generar sus utilidades. Se debe documentar cómo es que las diferentes áreas se relacionan entre sí para generar valor para los clientes. Es común describir las etapas de Innovación del producto, Administración de relaciones con los clientes, Administración de la Infraestructura, y por último, los Aspectos Financieros.

**Procesos de negocio.** En este nivel se describen las actividades más importantes de la empresa (el núcleo del negocio). Se sugiere modelar la Cadena de valor del negocio y de ahí obtener los procesos del negocio. Los procesos son un conjunto de actividades relacionadas y agrupadas que en conjunto reciben un insumo y producen una salida. Dichos procesos pueden automatizarse mediante sistemas de cómputo desarrollados a la medida o mediante la compra de sistemas existentes en el mercado como ERP o BPM<sup>[3]</sup>.

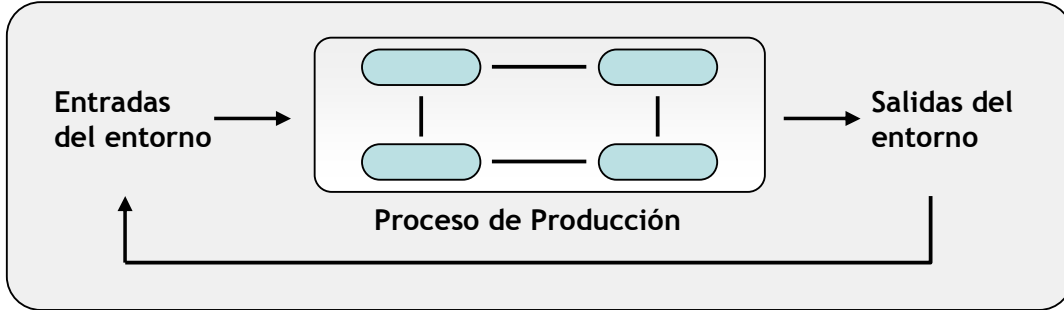
**Redes y Tecnologías de la Información.** Esta etapa se refiere a la tecnología que debe utilizarse para apoyar los procesos de negocio. Claro está que toda la información contenida en cualquier tipo de sistema de cómputo no serviría de mucho sin redes de computadoras que permitieran la comunicación de dicha información entre clientes, proveedores y la empresa misma.



### 1.2.1.2 Definición técnica microeconómica de la organización

- En la definición microeconómica de las organizaciones, la compañía transforma, mediante el proceso de producción, el capital y la mano de obra (los factores de producción primarios que proporciona el entorno), en productos y servicios (salidas al entorno). El entorno consume los productos y servicios, y proporciona capital y mano de obra adicionales como entradas al ciclo de retroalimentación.

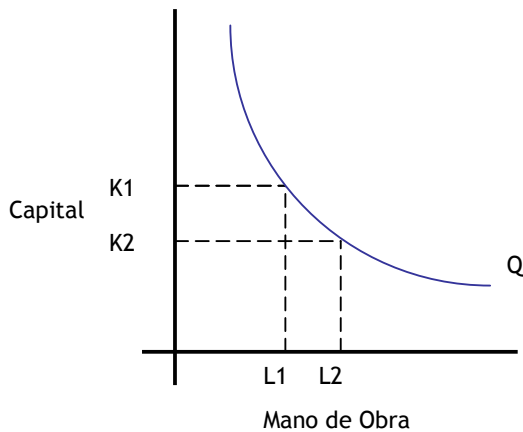
Figura 6: Proceso de Producción



- La perspectiva microeconómica es un modelo técnico de la compañía en el que ésta combina el capital y la mano de obra en una función de producción para generar un solo producto en una cantidad  $Q$ . La compañía puede sustituir libremente el capital por mano de obra en tanto no se salga de la curva  $Q$ . En esta perspectiva, el proceso de producción no se examina y se considera prácticamente como una caja negra.

### 3. ERP (Planeación de Recursos Empresariales) y BPM (Business Process Management) (solo reubicar)

Figura 7: Perspectiva Microeconómica



**Organización (definición técnica):** Estructura social formal estable que toma recursos del entorno y los procesa para producir salidas.

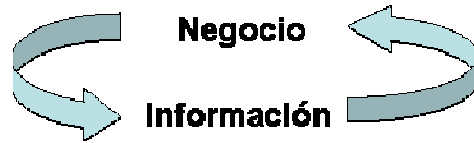
**Organización (definición conductual):** Conjunto de derechos, privilegios, obligaciones y responsabilidades que alcanzan un delicado equilibrio con el paso del tiempo, gracias a los conflictos y la resolución de los mismos.

## 1.2.2 Niveles de Servicios de TI en una organización (ITIL)

### 1.2.2.1 Gestión de Servicios TI

Las tecnologías de la información son tan antiguas como la historia misma y han jugado un importante papel. Sin embargo, no ha sido hasta tiempos recientes que mediante la automatización de su gestión se han convertido en una herramienta imprescindible y clave para empresas e instituciones.

La información es probablemente la fuente principal de negocio en el primer mundo y ese negocio a su vez genera ingentes cantidades de información. Su correcta gestión es de importancia estratégica y no debe considerarse como una herramienta más entre muchas otras.



Hasta hace poco las infraestructuras informáticas se limitaban a dar servicios de soporte y de alguna forma eran equiparables con cualquier otro material de oficina, importantes y necesarios para el correcto funcionamiento de la organización. Sin embargo, en la actualidad esto ha cambiado y los servicios TI representan generalmente una parte sustancial de los procesos de negocio.

Los objetivos de una buena gestión de servicios TI son:

- Proporcionar una adecuada gestión de la calidad
- Aumentar la eficiencia
- Alinear los procesos de negocio y la infraestructura TI
- Reducir los riesgos asociados a los Servicios TI
- Generar negocio

ITIL nace como un código de buenas prácticas dirigidas a alcanzar esas metas mediante:

- Un enfoque sistemático del servicio TI centrado en los procesos y procedimientos
- El establecimiento de estrategias para la gestión operativa de la infraestructura TI

La **Information Technology Infrastructure Library** ('Biblioteca de Infraestructura de Tecnologías de Información'), frecuentemente abreviada **ITIL**, es un marco de trabajo de las mejores prácticas destinadas a facilitar la entrega de servicios de tecnologías de la información (TI) de alta calidad. ITIL resume un extenso conjunto de procedimientos de gestión ideados para ayudar a las organizaciones a lograr calidad y eficiencia en las operaciones de TI. Estos procedimientos son independientes del proveedor y han sido desarrollados para servir de guía para que abarque toda infraestructura, desarrollo y operaciones de TI.

Aunque se desarrolló durante los años 80, ITIL no fue ampliamente adoptada hasta mediados de los años 90. ITIL es considerada junto con otros marcos de trabajo de mejores prácticas tales como la *Information Services Procurement Library* (ISPL, 'Biblioteca de adquisición de servicios de información'), la *Application Services Library* (ASL, 'Biblioteca de servicios de aplicativos'), el método de desarrollo de sistemas dinámicos (DSDM, *Dynamic Systems Development Method*) y el Modelo de Capacidad y Madurez (CMM/CMMI). A menudo se relaciona con los lineamientos tecnológicos de la información mediante COBIT (*Control Objectives for Information and related Technology*).

El concepto de gestión de servicios de TI, aunque relacionado con ITIL, no es idéntico: ITIL contiene una sección específicamente titulada «Gestión de Servicios de TI, sin embargo es importante señalar que existen otros marcos parecidos. La Gestión de Servicio ITIL está actualmente integrada en el estándar ISO 20000 (anteriormente BS 15000).

ITIL se construye en torno a una vista basada en proceso-modelo del control y gestión de las operaciones atribuida a W. Edwards Deming <sup>[4]</sup>. Las recomendaciones de ITIL fueron desarrolladas en la década de los ochentas, por la Central Computer and Telecommunications Agency (CCTA) del gobierno británico como respuesta a la creciente dependencia de las tecnologías de la información y al reconocimiento de que sin prácticas estándar, los contratos de las agencias estatales y del sector privado creaban independientemente sus propias prácticas de gestión de TI y duplicaban esfuerzos dentro de sus proyectos, lo que resultaba en errores comunes y mayores costos.

ITIL fue publicado como un conjunto de libros, cada uno dedicado a un área específica dentro de la Gestión de TI. Los nombres *ITIL* e *IT Infrastructure Library* ('Biblioteca de infraestructura de TI') son marcas registradas de la "Office of Government Commerce" OGC ('Oficina de Comercio Gubernamental'), que es una división del Ministerio de Hacienda del Reino Unido.

En abril de 2001 la CCTA fue integrada en la OGC, desapareciendo como organización separada.

En diciembre de 2005, la OGC emitió un aviso de una actualización a ITIL, conocida comúnmente como ITIL v3, que estaba planificada para ser publicada a finales de 2006. Se esperaba que la publicación de ITIL versión 3 incluya cinco libros principales, concretamente:

Diseño de Servicios de TI, Introducción de los Servicios de TI, Operación de los Servicios de TI, Mejora de los Servicios de TI y Estrategias de los Servicios de TI, consolidando buena parte de las prácticas actuales de la versión 2 en torno al Ciclo de Vida de los Servicios.

Uno de los principales beneficios propugnado por los defensores de ITIL dentro de la comunidad de TI es que proporciona un vocabulario común, consistente en un glosario de términos precisamente definidos y ampliamente aceptados. Un nuevo glosario ampliado ha sido desarrollado como clave de ITIL versión 3.

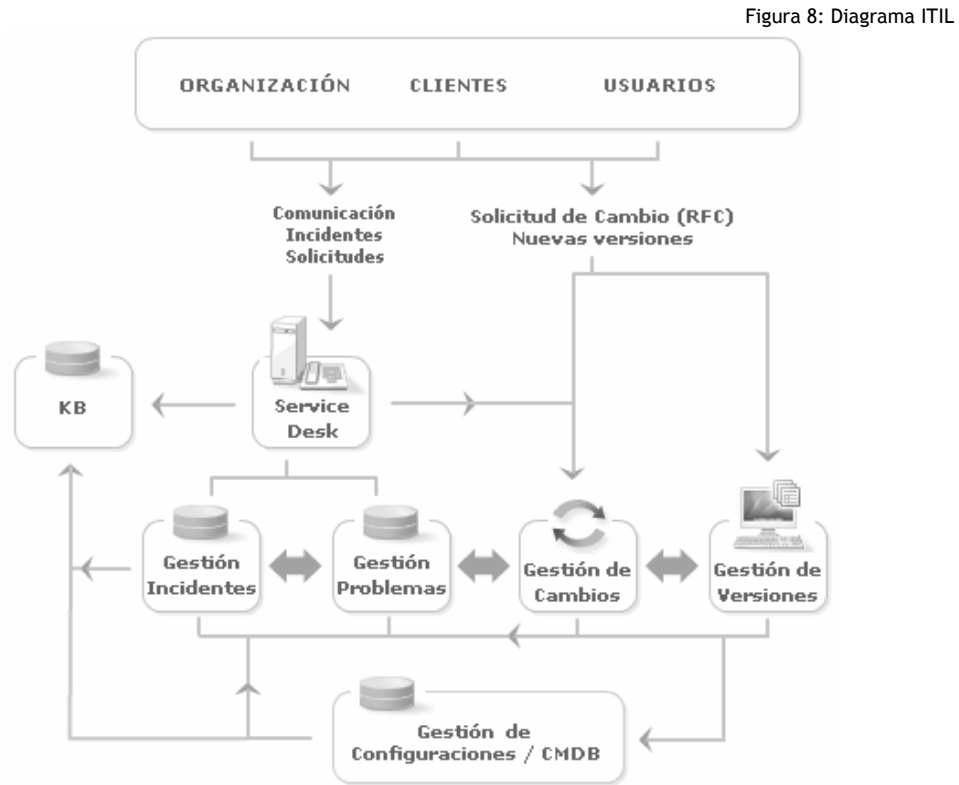
---

4. Estadístico estadounidense, profesor universitario, autor de textos, consultor y difusor del concepto de calidad total.

### 1.2.2.2 Soporte al Servicio

El soporte al servicio se preocupa de de todos los aspectos que garanticen la continuidad, disponibilidad y calidad del servicio prestado al usuario.

El siguiente diagrama resume brevemente los principales aspectos de la metodología de soporte al servicio según los estándares ITIL:



### 1.2.2.3 Organización, clientes y usuarios

- Clientes: Son los encargados de contratar los servicios de TI y a los que hay que rendir cuentas, respecto a los acuerdos y niveles de servicio.
- Usuarios: Son aquellos que utilizan los servicios de TI para llevar a cabo sus actividades.
- Organización: la propia organización de TI debe considerarse como otro cliente/usuario más de los servicios de TI.

#### 1.2.2.3.1 Service Desk

Debe representar el centro neurológico de todos los procesos de soporte al servicio:

- Registrando y monitorizando incidentes.
- Aplicando soluciones temporales a errores conocidos en colaboración con la gestión de problemas.
- Colaborando con la gestión de configuraciones para asegurar las actualizaciones.
- Gestionando cambios solicitados vía peticiones de servicio en colaboración con la gestión de cambios y versiones.

#### 1.2.2.3.2 Knowledge Base

La base de conocimiento debe recoger toda la información necesaria para:

- Ofrecer una primera línea de soporte ágil y eficaz sin necesidad de escalar el problema
- Realizar una tarea comercial y de soporte de negocio.

#### **1.2.2.3.3 Gestión de incidentes**

Tiene como objetivo resolver cualquier incidente que cause interrupción en el servicio de la manera más rápida y eficaz posible.

La gestión de incidentes no debe confundirse con la gestión de problemas, pues a diferencia de esta última, no se preocupa de encontrar y analizar las causas subyacentes a un determinado incidente sino exclusivamente a restaurar el servicio.

#### **1.2.2.3.4 Gestión de problemas**

Sus funciones principales son:

- Investigar las causas subyacentes a toda alteración, real o potencial, del servicio de TI
- Determinar posibles soluciones
- Proponer las peticiones de cambio
- Realizar revisiones post implementación en colaboración con la gestión de cambios.

#### **1.2.2.3.5 Gestión de Cambios**

Sus funciones principales son:

- Evaluar el impacto de los posibles cambios sobre la infraestructura de TI
- Tramitar los cambios mediante proceso y procedimientos estandarizados y consistentes
- Revisar junto a la gestión de problemas y a los usuarios, los resultados post implementación.

#### **1.2.2.3.6 Gestión de Versiones**

Sus funciones principales son:

- Implementar los cambios
- Llevar a cabo reparaciones de emergencia
- Desarrollar planes para el lanzamiento de nuevas versiones, y recuperación de versiones antiguas.

#### **1.2.2.3.7 Gestión de configuraciones**

Sus funciones principales son:

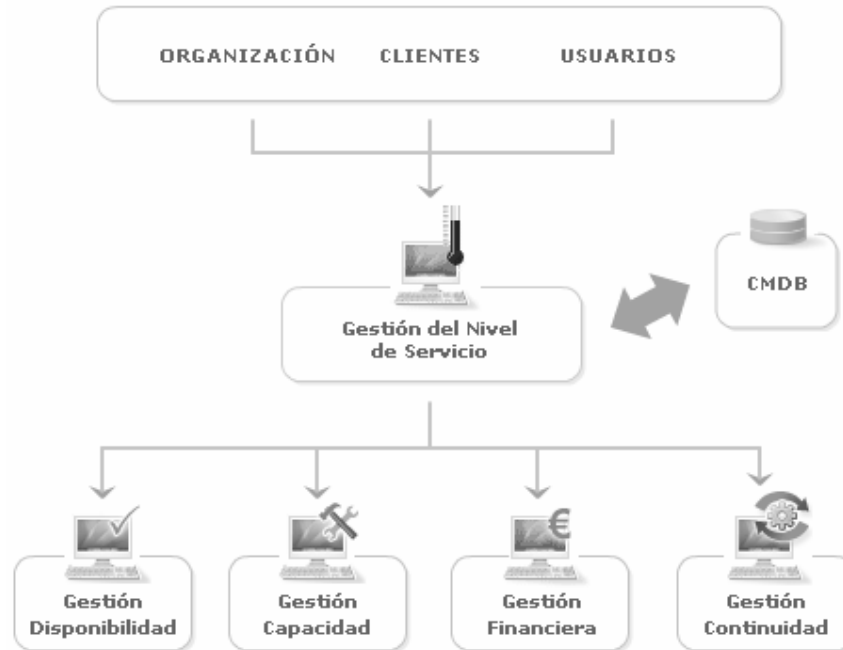
- Llevar el control de todos los elementos de configuración de la infraestructura de TI
- Realizar auditorías periódicas de configuración.
- Proporcionar información precisa sobre la configuración TI a todos los diferentes procesos.

#### **1.2.2.4 Provisión del Servicio**

La provisión del servicio se ocupa de los servicios ofrecidos en sí mismos. En particular de los Niveles de servicio, su disponibilidad, su continuidad, su viabilidad financiera, la capacidad necesaria de la infraestructura TI y los niveles de seguridad requeridos

El siguiente diagrama resume brevemente los principales aspectos de la metodología de provisión del servicio según los estándares ITIL:

Figura 9: Diagrama Servicios ITIL



#### 1.2.2.5 Organización, clientes y usuarios

- Clientes: Son los encargados de contratar los servicios de TI y a los que hay que rendir cuentas, respecto a los acuerdos y niveles de servicio.
- Usuarios: Son aquellos que utilizan los servicios de TI para llevar a cabo sus actividades.
- Organización: la propia organización de TI debe considerarse como otro cliente/usuario más de los servicios de TI.

#### 1.2.2.6 Gestión del nivel del servicio

Es la encargada de definir los servicios TI prestados y formalizados en términos de Acuerdos de Nivel de Servicio y Acuerdos de Nivel operativo que tomen en cuenta tanto las necesidades de los clientes como los costos asociados.

También es la responsable de la creación de planes y emisión de informes sobre la calidad del servicio.

##### 1.2.2.6.1 CMDB (Configuration Management Database)

La base de datos de configuraciones incluye:

- Información detallada de cada elemento de configuración (físicos y lógicos)
- Interrelaciones entre los diferentes elementos de configuración, como por ejemplo relaciones "padre - hijo" o interdependencias tanto lógicas como físicas.

##### 1.2.2.6.2 Gestión de disponibilidad

Hoy en día es esencial que los servicios de TI estén disponibles de continuo y sin interrupciones. El objetivo principal de la gestión de disponibilidad es asegurar que los servicios de TI:

- Estén disponibles siempre que se soliciten
- Sean fiables y tengan un margen operativo
- Estén correctamente mantenidos

#### **1.2.2.6.3 Gestión de Capacidad**

Se encarga de proporcionar la capacidad de TI necesaria para suministrar los servicios previstos. Sus principales actividades son:

- Elaborar planes de capacidad
- Monitorizar el rendimiento de la infraestructura de TI
- Realizar simulaciones de requisitos de capacidad para diferentes escenarios previstos.

#### **1.2.2.6.4 Gestión Financiera**

Es la encargada de medir la relación calidad/costo y tomar las medidas correctivas al respecto para ofrecer buenos servicios a precios razonables.

Una de sus principales tareas es hacer comprender tanto a la organización como a los clientes cuales son los beneficios y costos reales del servicio para que estos puedan asumir los gastos asociados.

#### **1.2.2.6.5 Gestión de la continuidad del servicio**

Sus principales actividades se resumen en:

- Evaluar los riesgos que pongan en peligro la continuidad del servicio
- Elaborar planes de contingencia
- La rápida recuperación del servicio tras fallos graves en la infraestructura TI o desastres naturales.

### **1.2.3 Errores Informáticos**

#### **1.2.3.1 Fallas en Sistemas de Software**

El orden en la presentación de los siguientes ejemplos de desastres ocasionados directa o indirectamente por software y por las computadoras que lo operan, es exclusivamente cronológico sin relación al área de aplicación o las consecuencias que ocasionaron.

**Fracaso del Mariner 1 (1962).** Podemos remontarnos mucho tiempo atrás para descubrir errores ya causados por computadoras. La primera misión del programa Mariner (con costo total de la misión Mariner 1 hasta Mariner 10 de 554 millones de dólares) fracasó por culpa de un carácter incorrecto ('- ') en la especificación del programa de control para el cohete de propulsión Atlas lo cual causó finalmente que el vehículo se saliera de curso. Ambos, el cohete y el vehículo espacial, tuvieron que ser destruidos poco después del lanzamiento. Adicionalmente, se cree que un error de computadora también fue la causa del fracaso del Mariner 8 en 1971.

**Sobregiro del Banco de New York (1985).** En Noviembre de 1985, el Banco de New York (BoNY) tuvo accidentalmente un sobregiro de \$32 billones de dólares (¡una buena suma si consideramos que esto fue hace 15 años!) por culpa de un contador de 16 bits (la mayoría de los demás contadores eran de 32 bits) que se activó ocasionando un "overflow" del contador que nunca fue verificado. BoNY no pudo procesar nuevos créditos de transferencias de "securities", mientras que la Reserva Federal de New York automáticamente hizo un traspaso de \$24 billones de dólares al BoNY para cubrir sus gastos por un día, por lo cual el banco tuvo que pagar \$5 millones de dólares por los intereses diarios, hasta que el software fue arreglado.

**Accidente de un F-18 (1986).** En Abril 1986 un avión de combate F-18 se estrelló por culpa de un giro descontrolado ("unrecoverable spin") atribuido a una expresión "IF-THEN" para la cual no había una instrucción "ELSE" porque se pensó que era innecesaria, resultando en una transferencia fuera de control del programa.

**Muertes por el Therac-25 (1985-1987).** El acelerador lineal médico, Therac-25, producido por AECL (Atomic Energy of Canada Limited), fue diseñado para tratamiento a pacientes por medio de radiación de Rayos X de dos tipos: (i) tratamiento de rayo directo de bajo poder, y (ii) tratamiento de rayo indirecto reflejado de alto poder. Entre 1985 y 1987 este sistema ocasionó la muerte de varios pacientes en diferentes hospitales de USA y Canadá por culpa de radiaciones de alto poder aplicadas de manera incontrolada. La probable causa de los accidentes consistía en que para ciertas secuencias de comandos del operador de la máquina, los controles de la computadora llevaban la máquina a un estado interno erróneo y muy peligroso, generando una sobredosis masiva de radiación al paciente. Después de amplia publicidad de estos accidentes se descubrió que la FDA (Federal Drug Agency) no especificaba requisitos, y no hacía revisiones sobre prácticas de desarrollo de software o control de calidad de software en dispositivos médicos. El FDA informó en Septiembre de 1987 que comenzaría a requerir controles de software integrados a ciertas clases de dispositivos médicos.

**Avión derribado por el USS Vincennes (1988).** En julio de 1988 la fragata USS Vincennes estaba asignada al golfo pérsico. Después de repetidos intentos de comunicación por radio, el USS Vincennes disparó un misil (por error) derribando un avión Airbus comercial iraní matando a todos los 290 pasajeros y tripulantes. Esto ocurrió mientras el Airbus ganaba altura, bajo la suposición incorrecta de que era un avión de combate F-14 que descendería sobre el barco de manera hostil. Aunque la orden de disparo fue dada por el comandante del navío, se culpa como causa contribuyente del incidente al sistema de radar AEGIS, el cual con su sistema de interfase de usuario mostraba únicamente un punto junto a un dato textual representando al avión, en lugar del eco real del radar sobre el avión. Posteriormente se supuso que en algún momento la aerolínea iraní estuvo en la proximidad de un F-14, probablemente durante el despegue del aeropuerto, confundiendo al sistema AEGIS y asociando de manera incorrecta la información transmitida por los "transponders" aire-tierra del F-14 a la aerolínea. Cuando el avión despegó, éste quedó asociado con los datos del F-14 sobre la pantalla. Un despliegue inconveniente y posiblemente confuso de la información de altitud del avión posiblemente confundió aún más a los oficiales del barco, los cuales supusieron que el F-14 estaba descendiendo, aunque en realidad estaba ganando altura. La inclusión de un eco real del radar sobre la pantalla hubiera hecho posible determinar que el eco del radar del avión era del tamaño incorrecto para un avión de combate.

**Falla del software de AT&T (1990).** El 15 de enero de 1990, AT&T (American Telegraph and Telephone) la compañía que controla las redes del mayor sistema de comunicación en el mundo, experimentó una falla masiva que dejó fuera de servicio su sistema de comunicaciones de larga distancia. La falla duró alrededor de nueve horas e interrumpió millones de llamadas de larga distancia. Un error en el software de manejo de excepciones de un tipo particular de sistema de switching telefónico resultó en una falla de switching, que a su vez causó otras fallas de switching en un efecto cascada. Según Neuman, se reportó que la última causa del problema tuvo origen en un programa en el lenguaje "C" que contenía una instrucción "BREAK" dentro de una cláusula "IF" dentro de una cláusula "SWITCH".

**Aberración esférica en el telescopio espacial Hubble (1990).** El 25 de Abril de 1990 se puso en órbita el famoso telescopio espacial Hubble desde el vehículo espacial Discovery. Al poco tiempo, NASA descubrió que el componente más crítico del telescopio de \$4 billones de dólares, en su espejo principal, tenía una gran falla, imposibilitando producir imágenes altamente enfocadas. El problema en su lente es técnicamente conocido como una "aberración esférica". Una investigación de la NASA reveló que el espejo se había construido con la forma incorrecta siendo 2 micrones (1 micron =  $10^{-6}$  metros) más plano a los lados de lo estipulado en el diseño original, un error bastante grande según los estándares de precisión de la óptica moderna. Este fue el error principal encontrado en el telescopio, considerando que hubo otros problemas adicionales, como en sus paneles solares, sus giroscopios, y contactos eléctricos. El problema del lente radicó en que nunca fue realmente probado antes de ser enviado al espacio. En su lugar, una simulación de computadora se usó como método de menor costo para validar el rendimiento del espejo. Por desgracia, malos datos de entrada se utilizaron en la simulación, significando resultados despreciables. Para corregir el error final en el espacio, se agregó al telescopio óptica correctiva a un costo muchas veces mayor que una prueba en tierra del espejo, significando además que el espejo nunca funcionaría tan bien como se planeó. Por lo pronto, la NASA no planea otro telescopio de la magnitud del



Hubble, por lo cual los astrónomos tendrán que limitarse a las restricciones actuales del Hubble, con el cual sólo se pueden ver objetos aproximadamente 20 veces más grandes de lo original.

**Duplicación de solicitudes de transferencias bancarias (1990).** En 1990 un error de software ocasionó que un banco en el Reino Unido duplicara cada solicitud de transferencia de pago por un periodo de media hora, acumulando 2 billones de libras esterlinas adicionales. Aunque el banco expresó haber recuperado todos los fondos, se especuló que los posibles intereses perdidos pudieran haber llegado a medio millón de libras por día.

**Falla del software de los misiles Patriot (1991).** En las primeras etapas de la guerra del golfo pérsico de 1991, el sistema Patriot fue descrito como altamente exitoso. En análisis posteriores, los estimados de su efectividad fueron disminuidos seriamente de 95% a 13% o incluso menos. El sistema fue diseñado para trabajar en un ambiente mucho más limitado y menos hostil que el que había en Arabia Saudita. Según reportó posteriormente el New York Times, una falla en la computadora de tierra del misil Patriot fue responsable de evitar la peor baja americana durante la guerra. Esto resultó en su inoperabilidad, permitiendo que un misil "scud" destruyera unas barracas militares americanas en Dhahran, Arabia Saudita, causando 29 muertos y 97 heridos. Aparentemente el sistema de radar del Patriot nunca vio al misil Scud. Según oficiales del ejército "una combinación imprevista de docenas de variables - incluyendo la velocidad, altura y trayectoria del Scud - causaron la falla del sistema de radar... [este caso fue] una anomalía que nunca apareció durante las horas de pruebas."

El error se atribuye a una acumulación de inexactitudes en el manejo interno del tiempo de la computadora del sistema. Aunque el sistema ejecutaba según las especificaciones, éste debía ser apagado y prendido con la suficiente frecuencia para que el error acumulado nunca fuera peligroso. Como el sistema se usó de manera no planeada, una pequeña inexactitud significó un serio error. Después de 8 horas de uso se detectó el problema del reloj acumulado. La corrección sólo se logró al día siguiente de la catástrofe (Mellor, 1994; Schach 1997).

**Error en el procesador Pentium de Intel (1994).** En 1994 un error de punto flotante en el procesador Pentium le costó US \$475 millones a Intel. El error no fue reconocido públicamente por varios meses por Intel diciendo que el procesador era "suficientemente bueno" además de que sería muy difícil que el error ocurriera. Actualmente, Intel está sufriendo otros problemas similares con sus procesadores, como la unidad MTH (Memory Translator Hub) usado para transferir señales de la memoria a otra unidad de la computadora (Intel 820) que podría significarle un costo similar. Recientemente ha tenido problemas con la última generación del Pentium III de 1 Ghz, donde se ha visto obligada a retirarlo del mercado. ¡Al menos la compañía ya ha aprendido a reconocer sus errores!

**Error en sistema de autenticación de tarjeta de crédito (1995).** Según un artículo del 4 de Noviembre 4 de 1995 del periódico Guardián en UK se relata que los dos sistemas más grandes en UK para la autorización de crédito (Barclay's PDQ y NatWest's Streamline) fallaron el sábado 28 de octubre de 1995 dejando a los negocios sin poder verificar las tarjetas de crédito de sus clientes. En el caso de Barclay, más del 40% de las transacciones fallaron por un "error en el sistema de software". Para NatWest, el problema fue una gran cola de llamadas, por razones desconocidas, las cuales retrasaron la autenticación de tarjetas. Aunque ambos tenían sistemas de contingencia permitiendo a los negocios telefonar para autenticar solicitudes, por el volumen de ventas las líneas se saturaron rápidamente.

**Explosión del cohete Ariane 5 (1996).** El 6 de Junio de 1996 una computadora fue culpada por la explosión del primer vuelo, el 501, del cohete Ariane 5 con un costo de US\$500 millones de dólares. El cohete, que parece que no estaba asegurado, llevaba cuatro satélites, ocasionando pérdidas totales de \$1.8 billones de dólares. El Ariane-5 estaba funcionando perfectamente hasta los 40 segundos iniciales cuando de repente empezó a salirse de su curso y fue destruido remotamente por una explosión solo fracciones de segundo después, ocasionadas por una señal enviada por un controlador de tierra del Ariane. Según ESA (Agencia Espacial Europea), la desviación fuera de curso fue ocasionada por instrucciones de la computadora controlando los escapes de los dos poderosos impulsores del cohete. Incluso se especuló que la instrucción fue generada por la computadora porque creyó que el cohete

se estaba saliendo de su curso y de esta manera estaría corrigiendo el curso de vuelo. Según el reporte final, la causa de la falla fue una excepción de software ocurrida durante la conversión de un número flotante de 64-bits a un número entero de 16 bits. El número flotante siendo convertido tenía un valor mayor del que podía ser representado por un número entero de 16 bits (con signo). Esto resultó en un "error de operando".

Las instrucciones de conversión de datos (código en Ada) no estaban protegidos de causar tal error de operando, aunque otras conversiones de variables similares en el mismo lugar, sí estaban protegidas. El origen del problema parece haber sido en que el Ariane 5 podía llevar un mayor número de satélites que el Ariane-4, incrementando así su peso. Sin embargo el Ariane-5 utilizaba una gran cantidad de software diseñado para el Ariane-4. Las conclusiones finales no oficiales fueron que ningún método formal hubiera detectado el problema, ya que la raíz de tal era a nivel de comunicación entre humanos, en relación a información física aparentemente no relacionada y no un problema de programación.

**Error del sistema de cobranza lleva a una compañía a la quiebra (1996).** Un artículo en la edición de Abril de 1996 de "TVRO Dealer" (una publicación del área de televisión por satélite) describe cómo el intento de un servicio de programación de una gran compañía de televisión por satélite al cambiar a un nuevo sistema de software de cobranza el 28 de Marzo anterior finalmente causó la quiebra de la compañía.

**Error del sistema de cobranza en MCI (1996).** En la edición del 29 de marzo de 1996 del Washington Post, MCI reportó que devolverían aproximadamente 40 millones de dólares a sus clientes por causa de un error de cómputo. El error de cobranza fue descubierto por un reportero investigador de una estación local de televisión en Richmond, VA. Los reporteros encontraron que fueron facturados por 4 minutos después de hacer una llamada de tan solo 2.5 minutos, dando lugar a una profunda investigación.

**Mayor falla de una computadora en la historia de bancos en USA (1996).** El 18 de mayo de 1996 la revista US & World Report, y al siguiente día el diario The Boston Globe, reportaron que aproximadamente 800 clientes del First National Bank of Chicago fueron sorprendidos al ver que sus saldos eran \$924 millones de dólares más de lo que tenían la semana pasada. La causa fue el tradicional "cambio en el programa de la computadora". De acuerdo a la Asociación de Banqueros Americanos, el total de \$763.9 billones fue la cantidad más grande para tal error en la historia bancaria de los Estados Unidos, más de seis veces el total de fondos ("assets") del First Chicago NBD Corp. El problema fue atribuido a un "error de la computadora".

**Falla de la computadora del centro de control de tráfico aéreo de NY (1996).** El 20 de Mayo 1996 falló la computadora del Centro de Control Aéreo de Nueva York (ARTCC - NY Air Route Traffic Control Center) que controla el tráfico aéreo sobre los estados de New York, Connecticut, New Jersey, Pennsylvania y parte del océano Atlántico. La computadora de 7 años de vigencia perdió capacidad de servicio efectivo ("falló") dos veces la tarde del lunes 20 de mayo; la primera vez por 23 minutos y la segunda por alrededor de una hora, una hora más tarde. Parece que 4 días antes se había instalado un nuevo software en el sistema. Se volvió al sistema anterior, con procedimientos de control de tráfico aéreo más ineficientes, ocasionando un límite más bajo de saturación de tráfico y retrasos en los despegues de alrededor de una hora en los aeropuertos principales en el área; junto con un incremento en la carga de trabajo de los controladores y menor seguridad, incluyendo la desactivación de la "alerta automática de conflictos".

**Mala planificación del nuevo sistema de una administradora de servicios de salud (1997).** Según reportó el Wall Street Journal el 11 de diciembre de 1997, Oxford Health Plans Inc., administradora de servicios de salud en USA, de gran crecimiento en los últimos tiempos, anunció que registraría una pérdida de US\$120 millones o más durante ese trimestre, además de una pérdida adicional de US\$78,2 millones, su primera pérdida desde que salió a la bolsa en 1991.

La razón principal fue la larga lista de problemas ocurridos con un sistema informático que se puso en línea en 1996; desde el diseño del sistema y su instalación hasta cómo fue

administrado por los ejecutivos del grupo Oxford. Los problemas ocasionaron que Oxford no pudiera enviar facturas mensuales a miles de cuentas de clientes además de incapacitarla para rastrear los pagos a cientos de médicos y hospitales. En menos de un año, los pagos no cobrados de sus clientes se triplicaron a más de US\$400 millones, mientras que el monto que Oxford debía a los proveedores de servicios médicos aumentó en más del 50%, a una suma superior a los US\$650 millones. La administradora de servicios médicos comenzó a planear su nuevo sistema informático en 1993, cuando sólo tenía 217,000 miembros. El sistema, desarrollado por Oracle, no comenzó a utilizarse hasta Octubre de 1996, cuando el número de abonados a su seguro médico había llegado a 1,5 millones. En ese momento el sistema ya era obsoleto. En lugar de tomar 6 segundo inscribir a un nuevo miembro, tomaba 15 minutos.

A pesar de esto y que la infraestructura administrativa de Oxford no daba abasto, los ejecutivos seguían inscribiendo nuevos clientes, en el último año se incorporó medio millón adicional. A finales de 1993, Oxford trató de ajustar el sistema, además de convertir de una sola vez la mayoría de su base de datos para facturación: unas 43,000 cuentas cubriendo a 1,9 millones de miembros. Esto significó la catástrofe final, ya que la transformación entre base de datos no funcionó, y mientras tanto se suspendió por unos meses la facturación ya que no se contaba con un sistema de seguridad, ni siquiera manual. A pesar de todo esto, Oxford siguió sus prácticas habituales de contabilidad, registrando aquellas facturas que no se habían cobrado como facturación trimestral. Los problemas finales surgieron cuando Oxford comenzó a poner al día las cuentas vencidas contactando a los clientes por primera vez en varios meses. Muchos se negaron a pagar y otros dijeron que hacía mucho que habían cancelado su cuenta.

Por consiguiente, Oxford tuvo que registrar US\$111 millones como deudas incobrables y reconoció que tenía 30,000 afiliados menos de lo que había calculado. El presidente reconoció que debía haber contratado un ejército de trabajadores temporales para que escribieran a máquina las facturas. Por otro lado, lo primero que perdieron fueron los clientes pequeños, pero como luego no resolvieron ningún problema comenzaron a perder a los clientes grandes.

**Error de un controlador de discos de Toshiba (1999).** En noviembre de 1999 Toshiba llegó a un arreglo fuera de corte que le costaría a la compañía más de \$2 billones de dólares para cubrir errores que pudiesen haber significado la pérdida de información por culpa de fallas en los controladores de discos "floppy" producidos en sus computadoras portátiles a partir de 1980. Aunque los controladores fueron diseñados originalmente por NEC, Toshiba producía sus propios componentes y nunca incluyó la modificación hecha por NEC en 1987 lo cual hubiese evitado el problema. Lo más interesante del caso es que realmente nunca se reportó falla alguna. Queda por ver que consecuencias traerá este caso al resto de los fabricantes de computadoras para los cuales este precedente los tiene extremadamente preocupados.

**Actualización de software mal planificada paraliza Nasdaq (1999).** El 17 de noviembre de 1999 los corredores de la bolsa de valores Nasdaq no pudieron comprar ni vender acciones durante 17 minutos cruciales, después de que oficiales de Nasdaq intentaran actualizar sobre la marcha un sistema de software durante la última media hora de la sesión. Algo funcionó mal y los inversionistas tuvieron que pagar el precio.

**Error del milenio (2000).** Concluyo estos ejemplos con un pequeño pero nocivo error que se le conoce como el problema Y2K, "error del milenio", o inclusive "el" problema de software del siglo 20. El problema se remonta a la década de los 60, y radica en que hace mucho tiempo los programadores adoptaron la convención de representar el año con dos dígitos en lugar de cuatro. Esta convención ocasionaría fallas en los sistemas al llegar al año 2000, ya que se alambra el "19" (no se permitía utilizar un número que no fuera el "19"), para generar la fecha lo cual al llegar al año 2000 fallaría por saturar el registro de almacenamiento ("overflow"). Para empeorar las cosas a menudo los dígitos "99" o "00" eran valores reservados ("números mágicos") significando "nunca borrar esto" o "esta es una cuenta de demostración". Además, esto resultaba en el uso de un algoritmo incorrecto para reconocer años bisiestos. No está claro si hay una razón única para haber hecho esto; porque la memoria de las computadoras en esa época era extremadamente cara, o porque no se esperaba que estos sistemas duraran tanto tiempo, o incluso quizás porque no reconocieron el problema. Aunque el problema se activó en este nuevo milenio, se tiene precedentes.

Muy pocos se dieron cuenta que la IBM 360 no podía manejar fechas mayores al 31 de Diciembre de 1969, hasta que estas máquinas empezaron a fallar a la medianoche hora local. IBM recomendó a sus clientes en América y Asia mentirles a las computadoras cambiando a una fecha anterior, mientras IBM empezó a crear una solución al problema. El problema es muy extenso, afecta hardware (BIOS, relojes de tiempo real, "embedded firmware", etc.), lenguajes y compiladores, sistemas operativos, generadores de números aleatorios, servicios de seguridad, sistemas de manejo de bases de datos, sistemas de procesamiento de transacciones, sistemas financieros, hojas de cálculo, conmutadores telefónicos, sistemas telefónicos y más. No es solamente un problema de sistemas de información, todo aquel sistema que use fechas está expuesto: automóviles, elevadores, etc. (Por ejemplo, en cierto momento Visa y MasterCard pidieron a sus bancos asociados que dejen de dar tarjetas que expiren en el 2000 o después.) No solamente es un problema de aplicaciones antiguas: el año pasado el paquete Quicken para manejo de finanzas personales fue corregido para ir más allá de 1999. En enero se reportó que el Instituto Nacional de Salud (NIH) en USA recibió nuevas PCs con tres versiones diferentes de BIOS, dos de las cuales fallaron a la transición Y2K. Las soluciones fueron muchas, consistiendo de diferentes etapas para analizar el problema particular y decidiendo las medidas a tomar, incluyendo no hacer nada y dejar que el problema ocurriera para luego arreglarlo. Existe aún toda una industria alrededor de este problema, con 1,800 compañías asociadas a la organización "year200" (year2000.com, year2000.org) y proporcionando certificaciones. Se dice que llegó a ser tanta la demanda por programadores del lenguaje Cobol (donde el problema fue más significativo) y tan desesperada la situación, que según se reportó, se fue a buscar programadores de Cobol ya retirados en asilos para ancianos. Actualmente se desconoce el costo final al problema de Y2K.

Es difícil estimar cual fue el costo total del problema Y2K. Según la compañía TMR (Technology Management Reports) de San Diego, los costos podrían haber sido superiores a \$1 trillón de dólares. Esto incluye reescribir programas existentes, adquisición e instalación de sistemas que los reemplacen, y productividad perdida por culpa de la interrupción de los sistemas para pruebas y las propias fallas por no ser funcionales en el año 2000. Y esto no incluye demandas por daños ocasionados. Otras compañías predijeron rangos de costos similares, como el grupo Gartner, que predijo un costo de \$600 billones de dólares a nivel mundial. Varias compañías asignaron presupuestos para este problema: Chase Manhattan dijo que costaría \$250 millones de dólares, American Airlines y Hughes Electronics dijeron que costarían \$100 millones de dólares cada una. La oficina de administración de presupuesto de la Casa Blanca (OMB - Office of Management and Budget) calculó sus costos de reparación en 2.8 billones. Esto incluía 4,500 computadoras para defensa nacional, tráfico aéreo, pagos de impuestos y seguridad social. El Grupo Gartner estimó que los costos para el Departamento de Defensa de USA podrían ser superiores a los US \$30 billones.

#### **Sobrecostos, Retrasos y Cancelaciones en los Sistemas de Software**

Lamentablemente los costos de los sistemas de software no se restringen a fallas en el software o los sistemas de computadora. Según una encuesta hecha por el Standish Consulting Group en 1995 compañías y agencias gubernamentales americanas perdieron \$81 billones de dólares por proyectos de software cancelados. Según Rob Thomsett, las causas de esto pueden ser de dos niveles principales: (i) factores que casi garantizan la cancelación del proyecto, como la falta de un dueño del proyecto; y (ii) factores que no resultan en una cancelación inminente del proyecto, pero seguramente ocasionarán reducciones substanciales en su calidad. En esta sección se muestran algunos ejemplos "clásicos" en orden cronológico.

**Sobrecosto y retraso en sistema de Allstate Insurance (1982).** En 1982, Allstate Insurance comenzó a construir un sistema para automatizar su negocio por \$8 millones. El supuesto esfuerzo de 5 años continuó hasta al menos 1993 cuando terminó costando cerca de \$100 millones.

**Sobrecosto, retraso y cancelación en sistema de London Stock Exchange (1983-1988).** El proyecto Taurus de la Bolsa de Valores de Londres estaba originalmente cotizado en 6 millones de libras. Varios años más tarde y más de 100 veces (13,200%) sobre presupuesto el proyecto fue cancelado, costando a la ciudad de Londres al momento de ser abandonado, 800 millones de libras.

**Sobrecosto y retraso en sistema del bombardero B-1 (1985).** El bombardero B-1 en servicio desde 1985 requirió US \$1 billón adicional para mejorar su software de defensa aérea que era inefectivo, aunque problemas de software imposibilitaron alcanzar los objetivos originales.

**Sobrecosto, retraso y cancelación en sistema de Bank of America (1988).** En 1988, Bank of America gastó US \$23 millones en un plan inicial de 5 años para desarrollar MasterNet, un sistema computarizado para contabilidad y reportes de "trust". Luego de abandonar el sistema viejo, gastaron \$60 millones adicionales para lograr que el nuevo sistema funcionara y finalmente terminaron desistiendo. Las cuentas de los clientes perdidos pudieron haber excedido los billones de dólares.

**Sobrecosto y retraso en sistema de control de rastreo por satélite (1989).** El software para la modernización de la Facilidad de Control de Rastreo por Satélite tomó 7 años más de lo previsto y costó \$300 millones adicionales ofreciendo menor capacidad de la requerida.

**Sobrecosto y retraso en sistema Airborne Self-Protection Jammer (1989).** El sistema ASJP (Airborne Self-Protection Jammer), un sistema electrónico de defensa aérea instalado en alrededor de 2,000 aviones de combate y ataque de la Marina Americana, costó US \$1 billón adicional, tomó 4 años adicionales, y sólo fue "efectivo operacionalmente marginalmente y apropiado operacionalmente marginalmente".

**Sobrecosto en sistema del avión de carga C-17 (1989).** El avión de carga C-17 construido por Douglas Aircraft costó \$500 millones adicionales por problemas del software aeronáutico. Un reporte de GAO notó que existieron 19 computadoras a bordo, 80 microprocesadores, y 6 lenguajes diferentes de programación.

### **1.3 Componentes Tecnológicos de un sistema de Información Actual**

#### **1.3.1 Aplicaciones**

##### **1.3.1.1 Servidores de Aplicaciones**

En informática se denomina servidor de aplicaciones a un servidor en una red de computadores que ejecuta ciertas aplicaciones de software.

Usualmente se trata de un dispositivo de software que proporciona servicios de aplicación a las computadoras cliente. Un servidor de aplicaciones generalmente gestiona la mayor parte (o la totalidad) de las funciones de lógica de negocio y de acceso a los datos de la aplicación.

Los principales beneficios de la aplicación de la tecnología de servidores de aplicación son la centralización y la disminución de la complejidad en el desarrollo de aplicaciones. Si bien el término es aplicable a todas las plataformas de software, hoy en día el término servidor de aplicaciones se ha convertido en sinónimo de la plataforma J2EE de Sun Microsystems.

##### **1.3.1.1.1 Características comunes**

Los servidores de aplicación típicamente incluyen también middleware (o software de conectividad) que les permite intercomunicarse con variados servicios, para efectos de confiabilidad, seguridad, etc. Los servidores de aplicación también brindan a los desarrolladores una Interfaz para Programación de Aplicaciones (API), de tal manera que no tengan que preocuparse por el sistema operativo o por la gran cantidad de interfaces requeridas en una aplicación web moderna.

Los servidores de aplicación también brindan soporte a una gran variedad de estándares, tales como HTML, XML, JDBC, SSL, etc., que les permiten su funcionamiento en ambientes web y la conexión a una gran variedad de fuentes de datos, sistemas y dispositivos.

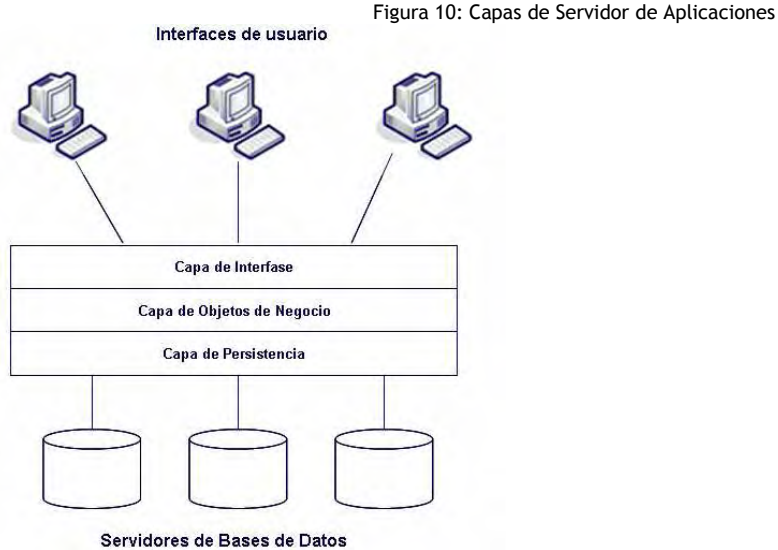
##### **1.3.1.1.2 Usos**

Un ejemplo común del uso de servidores de aplicación son los portales de Internet, que permiten a las empresas la gestión y divulgación de su información, y un punto único de entrada a los usuarios internos y externos. Teniendo como base un servidor de aplicación, dichos portales permiten tener acceso a información y servicios (como servicios Web) de manera segura y transparente, desde cualquier dispositivo.

##### **1.3.1.1.3 Arquitectura de Servidor de Aplicaciones**

Un servidor de aplicaciones contiene capas medias para soluciones cliente/servidor. Las aplicaciones de usuarios a través de interfaces genera requerimientos desde el servidor de aplicaciones, y esos servicios son almacenados y reciben información de la base de datos o de otros servidores de aplicaciones.

La siguiente imagen muestra las diferentes capas que contiene un servidor de aplicaciones.



- Capa de Interfase. Es la puerta frontal del servidor de aplicaciones. Cada interfase del usuario tiene privilegios a un grupo de servicios, o procedimientos remotos, que ocultan todos los detalles de proceso de negocio y los datos persistentes que se almacenan en el servidor de aplicaciones.
- Capa de Objeto de Negocio. Es una colección de muchas aplicaciones que encapsulan el proceso y las reglas de negocio. Un buen diseño de objetos de negocio se describe con los términos del negocio, no a nivel de programación, y generalmente debe ser rehusado por otras aplicaciones.
- La capa de persistencia. Trabaja como un divisor, que almacena objetos del negocio desde almacenamientos permanentes o interfaces con otros sistemas delegados. Cuando los objetos de negocio se necesitan, la capa de persistencia es llamada para recuperar información de la base de datos o de otro medio de almacenamiento, entonces la información recuperada se utiliza para generar un nuevo objeto de negocio, una vez que la es el objeto ya no es necesario, la capa de persistencia es responsable de almacenar los datos y quitar el objeto de memoria.

#### 1.3.1.1.4 MiddleWare

El Middleware es un software de conectividad que ofrece un conjunto de servicios que hacen posible el funcionamiento de aplicaciones distribuidas sobre plataformas heterogéneas. Funciona como una capa de abstracción de software distribuida, que se sitúa entre las capas de aplicaciones y las capas inferiores (sistema operativo y red). El Middleware nos abstraer de la complejidad y heterogeneidad de las redes de comunicaciones subyacentes, así como de los sistemas operativos y lenguajes de programación, proporcionando una API para la fácil programación y manejo de aplicaciones distribuidas. Dependiendo del problema a resolver y de las funciones necesarias, serán útiles diferentes tipo de servicios de middleware.

Por lo general el middleware del lado cliente está implementado por el Sistema Operativo subyacente, el cual posee las librerías que implementan todas las funcionalidades para la comunicación a través de la red.

Los Middleware han aparecido de manera relativamente reciente en el mundo de la informática. Ganaron popularidad en los 80s ya que eran la solución de como integrar las nuevas aplicaciones con los sistemas antiguos (legacy systems<sup>[5]</sup>), en todo caso, el término ha sido usado desde 1968. También facilitaba la computación distribuida, mediante conexión de múltiples aplicaciones para crear una conexión mucho mayor, sobre una red.

5. *Legacy Systems (sistemas heredados): Sistemas antiguos utilizados por las compañías por muchos años, estos sistemas no se puede sustituir por alguna dependencia en el proceso.*

#### **1.3.1.1.5 Categorías de Middleware**

El mercado del middleware esta en crecimiento todavía, pero existen algunas arquitecturas que comienzan a ser dominantes en el mercado. Las diferentes arquitecturas que están en el mercado se incluyen en las siguientes categorías:

- a) Protocolos remotos de bases de datos: Probablemente este protocolo es de los más comunes en las implementaciones de middleware, este protocolo se conoce desde las bases de datos cliente-servidor. Este protocolo permite al cliente que se encuentra en una red, comunicarse con la base de datos sin tener que preocuparse por el lenguaje de bajo nivel para poderse comunicar, algunos ejemplos son los paquetes de Microsoft llamados ODBC (Open Data Base Connectivity) y DAO (Data Acces Objects).

El objetivo principal es esconder los detalles de las llamadas de red y protocolos de comunicación que se utilizan para la conexión a la base de datos, se proveen nombres y ubicación, para localizar fácilmente los servicios disponibles, y después poder traducir los datos en múltiples lenguajes de programación.

- b) Procedimientos remotos: El DCE (Distributed Computing Enviroment) fue una de las primeras industrias con iniciativas para el desarrollo de estándares para la computación distribuida, compañías como IBM, HP, Digital y otras generaron acuerdos para la estandarización en las comunicaciones, los cuales permiten que los programas de cada compañía funcione en los equipos de otras compañías. Las especificaciones de DCE, incluyen estándares para la ejecución remota de procedimientos, seguridad, servicio de directorios, servicios de tiempo y servicios de distribución de archivos.
- c) Objetos distribuidos: Actualmente los programadores utilizan lenguajes Orientados a objetos; las iniciativas de programación distribuida se dirige hacia los objetos distribuidos en lugar de los las funciones y librerías distribuidas.

Un grupo llamado OMG (Object managment Group) creó un paquete de estándares y especificaciones para los objetos distribuidos llamado CORBA (Common Object Request Broker Arquitectura). El estándar CORBA tiene el soporte más grande de la industria, además de ser el más grande de los estándares de objetos distribuidos. Este estándar da un soporte para el cruce de plataformas, desde pc a servidores mainframes, el software que utilice CORBA puede ser escrito en una gran cantidad de lenguajes.

Muchas compañías dan soporte para CORBA por ejemplo Sun Microsystems, da un soporte muy grande a CORBA utilizando su plataforma JAVA, Microsoft también adopto la tecnología de CORBA, utilizando también un propio modelo de objetos distribuidos llamado DCOM (Distributed Component Object Model)

Monitores de procesos transaccionales: El concepto de esta capa está en las transacciones de la base de datos y se aplica a los procesamientos distribuidos. Una serie de operaciones pueden ser empacadas en una sola transacción, pero si la transacción falla en algún punto del proceso, todas las operaciones que se tenían que aplicar desde el principio del proceso, son recuperadas a su estado anterior (roll back), con esto se asegura consistencia en los datos, de manera que éstos siempre son fiables, no importa que problema haya ocurrido en la transacción.



#### **1.3.1.1.6 Servidores de aplicación J2EE**

Como consecuencia del éxito del lenguaje de programación Java, el término servidor de aplicaciones usualmente hace referencia a un servidor de aplicaciones J2EE. WebSphere (IBM), Oracle Application Server (Oracle Corporation) y WebLogic (BEA) están entre los servidores de aplicación J2EE privados más conocidos. EAServer (Sybase Inc.) es también conocido por ofrecer soporte a otros lenguajes diferentes a Java, como PowerBuilder. El servidor de aplicaciones JOnAS, desarrollado por el consorcio ObjectWeb, fue el primer servidor de aplicaciones libre en lograr certificación oficial de compatibilidad con J2EE. Tomcat (Apache Software Foundation) y JBoss son otros servidores de aplicación libres muy populares en la actualidad.

J2EE provee estándares que le permiten a un servidor de aplicaciones servir como "contenedor" de los componentes que conforman dichas aplicaciones. Estos componentes, escritos en lenguaje Java, usualmente se conocen como Servlets, Java Server Pages (JSPs) y Enterprise JavaBeans (EJBs) y permiten implementar diferentes capas de la aplicación, como la interfaz de usuario, la lógica de negocio, la gestión de sesiones de usuario o el acceso a bases de datos remotas.

La portabilidad de Java también ha permitido que los servidores de aplicación J2EE se encuentren disponibles sobre una gran variedad de plataformas, como Unix, Microsoft Windows y GNU/Linux.

Otros servidores de aplicación:

El término servidor de aplicaciones también ha sido aplicado a otros productos no-J2EE. Por ejemplo, con el aumento de la popularidad de .NET, Microsoft califica a su producto Internet Information Server como un servidor de aplicaciones. Adicionalmente, se pueden encontrar servidores de aplicación de código abierto y comerciales de otros proveedores; algunos ejemplos son Base4 Server y Zope.

#### **1.3.1.2 Servidor de Aplicaciones Oracle**

##### **1.3.1.2.1 Arquitectura**

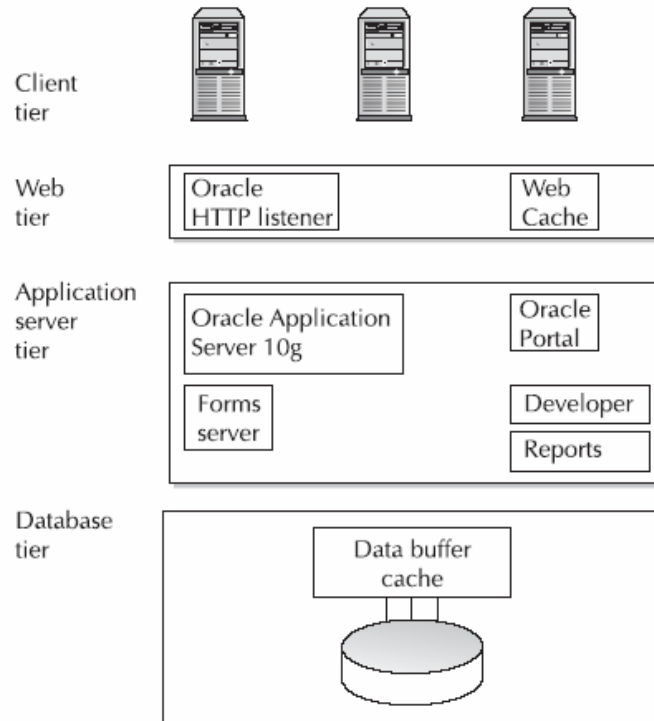
Oracle comenzó a desarrollar servidores de aplicaciones desde los años noventa, y constantemente ha improvisado y mejorado sus productos para dar una solución a las aplicaciones basadas en Web.

Oracle Application Server 10g es la última versión en una larga evolución de productos, comenzando en los noventa con Oracle Web Server, y Oracle Application Server. El servidor de aplicaciones está envuelto en un sofisticado sistema de módulos interrelacionados, los cuales pueden ser configurados de acuerdo a los requerimientos del cliente. Existen dos caminos para analizar la arquitectura del servidor de aplicaciones; desde el nivel de diseño y desde el nivel funcional. Ambos están basados en un modelo multicapa.

##### **1.3.1.2.2 El modelo multicapa**

Los productos de Oracle son envueltos por arquitecturas multicapa, en la imagen siguiente se muestra una arquitectura de los productos Oracle, y el cómo interactúan con Oracle Application Server. OAS tiene la flexibilidad de adoptar diferentes niveles de capas; puede utilizar dos, tres o hasta 4 capas en un solo modelo.

Figura 11: Modelo Multicapa



Las capas Oracle Application Server 10g se componen de:

**Capa Cliente (Client Tier):**

Contiene el navegador Web para usuarios finales

**Capa Web (Web Tier):**

Contiene el servidor Web y el Web Cache

**Capa de Servidor de Aplicaciones (Application Server Tier):**

Contiene todo el paquete del servidor de aplicaciones, además herramientas para desarrollo, administración y minería de datos como Oracle Application Server Portal 10g, Oracle Developer, Oracle Reports y Oracle Forms Server.

**Capa de Base de Datos (Database Tier):**

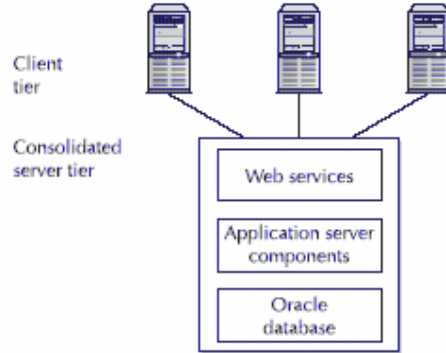
Contiene la base de datos que tiene el repositorio de metadatos del servidor de aplicaciones. Puede ser una sola instancia, o muchas instancias si se utiliza un ambiente de cluster (RAC).

No todos los negocios requieren tener un sistema multicapa de cuatro niveles, hay negocios más pequeños que pueden combinar algunas capas en una sola. Por ejemplo en una arquitectura de 3 capas, la capa Web y la capa del servidor de aplicaciones puede ser combinada. Los sistemas multicapas más grandes son los que utilizan cuatro niveles, esto quiere decir que en la capa de web existirán muchos servidores, docenas de servidores en la capa del servidor de aplicaciones y muchas instancias de base de datos, por cada nodo.

Para negocios pequeños es común que se utilicen modelos de dos capas. En la siguiente imagen se muestra un ejemplo donde sólo se muestran los clientes, y en un servidor está la

combinación de la capa Web, la capa de aplicación y la capa de base de datos. Normalmente las arquitecturas de servidores de aplicaciones se dimensionan con mucha memoria RAM y grandes procesadores y se diseñan de manera que se pueda agregar y quitar hardware de una manera flexible y funcional, pero cuando se utiliza un modelo de dos capas, es mucho más difícil lograrlo.

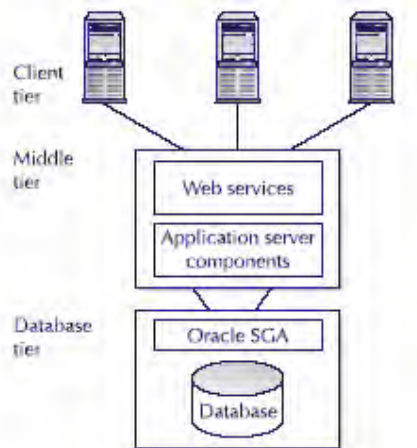
Figura 12: Modelo de dos capas



El modelo que predomina en los negocios medianos es el de 3 capas, en este modelo después de la capa del cliente, se encuentra el servidor Web y el servidor de aplicaciones en servidores separados, la base de datos también se encuentra en servidor por separado.

El modelo de 3 capas tiene algunos beneficios comparado al de dos capas, el primero es el incremento del procesamiento ante las peticiones que se reciben en el servidor de aplicaciones y la base de datos; el segundo beneficio es que se pueden agregar instancias al servidor de aplicaciones Oracle 10g, así como agregar capacidades a la memoria SGA.

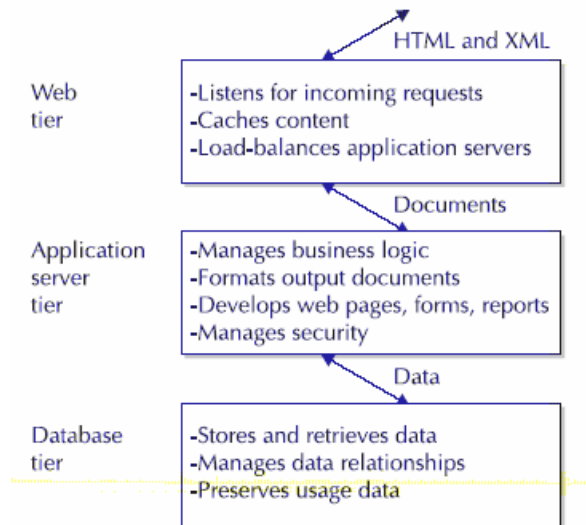
Figura 13: Modelo de 3 capas



### 1.3.1.3 Arquitectura funcional de Oracle Application Server 10g

En la siguiente figura se muestran las capas que puede tener el servidor de aplicaciones Oracle, así como los componentes que se utilizan en cada capa.

Figura 14: Capas de Servidor Aplicaciones Oracle



En la capa Web, el componente principal es el listener, el cual recibe los requerimientos de los clientes por medio de puertos; el componente Web cache, que almacena los componentes de la página Web, es el mecanismo de balanceo de cargas para asegurar una administración óptima de los recursos para el nivel de aplicaciones; el servidor Web utiliza Oracle http Server, el cual está basado en Apache.

El nivel de aplicaciones controla todo el modelo de negocio. Algunos de los componentes que se pueden encontrar en este nivel son: Oracle Portal, el cual es utilizado para publicar sitios Web; Oracle Reports, define especificaciones de contenido y Oracle Single Sign On (sso), el cual controla la seguridad para la capa del servidor de aplicaciones. En el nivel de base de datos se encuentran los componentes e infraestructura necesarios para que funcione el servidor de aplicaciones.

#### 1.3.1.3.1 Nivel Cliente

Este nivel consiste en la interfase que utiliza el usuario para poder ejecutar las aplicaciones del servidor de aplicaciones por medio del navegador Web. De esta forma el cliente siempre tendrá la última versión de la aplicación, además se tiene la gran ventaja de que el cliente puede acceder a la aplicación desde cualquier punto de la red.

#### 1.3.1.3.2 Nivel Web

La capa Web, contiene dos componentes muy importantes, el servidor de aplicaciones, también llamado Oracle http Server (OHS), y el componente llamado Web Cache. Esta capa es responsable de manejar todas las peticiones http, capturando mensajes Web y enviando información XML y HTML a los clientes.

#### 1.3.1.3.3 Oracle http Server (OHS)

Todos los sistemas Web de Oracle deben tener los procesos suficientes para poder atender las peticiones entrantes. El servidor Web http de Oracle es un componente del servidor de aplicaciones que escucha a través de un puerto en específico, después llega una petición de J2EE a través del componente mod\_oc4j al último contenedor OC4J que se cargó en memoria.

Es muy importante que el servidor Web tenga inteligencia para el balanceo de cargas, de esta manera un solo contenedor OC4j no es sobrecargado con el trabajo. Oracle ha resuelto este problema incorporando el servidor Web Apache al Servidor Oracle http, teniendo así la capacidad de balanceo de cargas, permitiendo una configuración más sencilla.

#### **1.3.1.3.4 Web Cache**

Este componente ayuda notablemente al rendimiento del servidor de aplicaciones, reduciendo la necesidad de regenerar la información dinámica o estática. El Web Cache se ubica antes que el servidor Web y almacena el contenido Web, ya sea estático o dinámico. Existen diversas virtudes de este componente, entre ellas la captura de páginas parciales, el balanceo de contenido Web, la capacidad de crear un cluster de Web cache haciendo que funcionen como uno sólo y tiene la capacidad de trabajar con otro tipo de servidores de aplicaciones, como Sub, IBM, BEA, etc.

#### **1.3.1.3.5 Nivel Servidor de Aplicaciones**

El corazón de Oracle Application Server 10g, es la capa del servicio de aplicaciones, este es el componente central del servidor, en este nivel se almacenan los módulos de OC4J de java, en este nivel se da el soporte de la conectividad de las aplicaciones a Java.

Como se mencionó anteriormente, Oracle Application Server 10g, incluye otros componentes independientes que se almacenan en este nivel, estos son:

- **Oracle Portal.** Permite una rápida definición y desarrollo de sitios Web.
- **Oracle Discoverer.** Permite una utilización más sencilla en la creación de queries.
- **Oracle Forms.** Es usado para dar formato y configurar la presentación de las páginas que ve el usuario, esto basado en metadatos <sup>[6]</sup>.
- **Oracle Personalization.** Este componente permite personalizar el sitio Web que el usuario final ve dependiendo de la historia del sitio Web.
- **Oracle Wireless.** Este componente permite la comunicación entre el servidor de aplicaciones y dispositivos móviles, como PDAS o teléfonos celulares. La información se reformatea dinámicamente para que se muestre de manera correcta en las pantallas de la mayoría de estos tipos de dispositivos.
- **Oracle Reports Server.** Este componente permite la creación de reportes, documentos y hojas de cálculo con información de la base de datos.
- **Single Sign On (SSO).** Este componente es un completo sistema de autenticación de usuarios, administrando roles y servicios Web, este componente también funciona con Java y Portal.
- **Oracle Internet Directory (OID).** Esta es una compilación de un directorio LDAP (Lightweight Directory Access Protocol), el cual provee de manera centralizada una gran cantidad de información de los usuarios, las aplicaciones y los recursos de la empresa.
- **Metadata Repository (Infraestructura).** Este componente es muy importante, se refiere a la infraestructura del servidor de aplicaciones. En esta infraestructura se almacenan los metadatos y permite una administración común entre diferentes instancias del servidor de aplicaciones.
- **Oracle Management Server (OMS).** Este componente se refiere a la consola de administración "Enterprise Manager", con la cual se pueden administrar todas las instancias del servidor de aplicaciones, bases de datos y aplicaciones Java.
- **Oracle Application Server TopLink.** Este componente provee persistencia para la información de Java. TopLink contiene el mapeo de las interfaces, y las traduce a estructuras tipo Java en tablas relacionales.

Todos los componentes que se mencionaron anteriormente se encuentran en la capa del servidor de aplicaciones, permitiendo al administrador la flexibilidad de crear múltiples instancias de Oracle Aplicación Server.

6. *Metadato: Se refiere a los datos que contienen otros datos, por ejemplo índices para localizar datos*

#### **1.3.1.3.6 Nivel de Base de Datos**

La base de datos requerida por el servidor de aplicaciones se encuentra en el nivel de base de datos. La función del nivel de base de datos es proveer al servidor de aplicaciones un almacenamiento consistente. Oracle Application Server 10g, también tiene una instancia especial llamada infraestructura, y es utilizada para almacenar los metadatos del servidor de aplicaciones en la base de datos. El esquema que contiene la infraestructura se puede generar y utilizar en cualquier base de datos, pero como buena práctica es recomendable que la infraestructura tenga su propia instancia.

Los componentes del servidor de aplicaciones que se interactúan con la infraestructura son:

- Oracle Application Server Portal.
- Oracle Reports
- Oracle Application Server Discoverer
- Oracle Application Server Personalization

#### **1.3.2 Bases de Datos**

Existen diferentes definiciones de las bases de datos, pero al entenderlas se puede notar que el concepto es el mismo, algunos ejemplos son los siguientes:

- Colección de datos interrelacionados almacenados en conjunto sin redundancias perjudiciales o innecesarias; su finalidad es servir a una aplicación o más de la mejor manera posible, los datos se almacenan de modo que resulten independientes de los programas que los usan, se emplean métodos bien determinados para incluir nuevos datos y para modificar o extraer los datos almacenados.
- Colección o depósito de datos, donde los datos están lógicamente relacionados entre sí, tienen una definición y descripción comunes, y están estructurados de una forma particular. Una base de datos es también un modelo del mundo real y como tal, debe poder servir para toda una gama de usos y aplicaciones.
- Conjunto de datos de la empresa almacenado en una computadora, que es utilizado por numerosas personas y cuya organización está regida por un modelo de datos.
- Conjunto estructurado de datos registrados sobre soportes accesibles por computadoras para satisfacer simultáneamente a varios usuarios de forma selectiva y en tiempo oportuno.
- Colección no redundante de datos que son compartidos por diferentes sistemas de aplicación.
- Colección integrada y generalizada de datos, estructurada atendiendo a las relaciones naturales de modo que suministre todos los caminos de acceso necesarios a cada unidad de datos con objeto de poder atender todas las necesidades de los diferentes usuarios.
- Conjunto de archivos maestros, organizados y administrados de una manera flexible de modo que los archivos puedan ser fácilmente adaptados a nuevas tareas imprevisibles.
- Colección de datos interrelacionados.

La aparición de la expresión base de datos se produce a comienzos de los años sesenta. En 1963 tuvo lugar en Santa Mónica (EEUU) un simposio en cuyo título se encontraba la expresión Data Base. En una de sus sesiones, se propuso una definición de base de datos que según las actas del simposio, no fue universalmente aceptada. Posteriormente en 1967 el grupo de estandarización Codasyl decidió cambiar su primitiva denominación de base de datos por el

de Data base Task Group. Poco a poco, el concepto y la expresión base de datos iba imponiéndose.

Una base de datos es un conjunto, colección o depósito de datos almacenados en un soporte informático no volátil. Los datos están interrelacionados y estructurados de acuerdo con un modelo capaz de recoger el máximo contenido semántico. Dada la relevación que tienen en el mundo las interrelaciones entre los datos, es imprescindible que la base de datos sea capaz de almacenar estas interrelaciones. En el mundo real existen restricciones semánticas a las que se está concediendo una importancia creciente y que en los sistemas actuales, tienden a almacenarse junto con los datos, al igual que ocurre con las interrelaciones. La base de datos se describe y se manipula apoyándose en un modelo de datos.

La redundancia de los datos debe ser controlada, de forma que no existan duplicidades perjudiciales innecesarias, y que las redundancias físicas, convenientes muchas veces a fin de responder a objetivos de eficiencia, sean tratadas por el mismo sistema, de modo que no puedan producirse inconsistencias. Esto podría resumirse diciendo que en la bases de datos no debe existir redundancia lógica, aunque sí se admite cierta redundancia física por motivos de eficiencia.

Las bases de datos pretender servir al conjunto de la organización, manejando los datos como otro recurso que viene a añadirse a los ya tradicionales. Por tanto las bases de datos han de atender a múltiples usuarios y a diferentes aplicaciones, en contraposición a los sistemas de archivos, en los que cada archivo esta diseñado para responder a las necesidades de una determinada aplicación.

Otro aspecto importante de las bases de datos es la independencia, tanto física como lógica entre datos y tratamientos. Esta independencia, objetivo fundamental de las bases de datos, es una característica esencial que distingue las bases de datos de los archivos y que han tenido una enorme influencia en la arquitectura de los SGBD.

La definición o descripción del conjunto de datos contenidos en la base deben ser únicas y estar integradas con los mismos datos. En los sistemas basados en archivos, los datos se encuentran almacenados en los archivos, mientras que su descripción esta separada de los mismos, formando parte de los programas, para lo cual se precisa que los lenguajes faciliten medios para descripción de los datos. Este tipo de organización da origen a infinidad de problemas, ya que a veces no se sabe cuál es la descripción de un determinado archivo, ya sea por perdida de la misma, o porque no se ha actualizado debidamente la correspondiente documentación y tampoco se conoce exactamente el programa que lo utiliza. En las bases de datos la descripción, y en algunos casos también una definición y documentación completas, se almacenan junto con los datos, de modo que estos están auto documentados, cualquier cambio que se produzca en dicha documentación se ha de reflejar y quedar guardado en el sistema, con todas las ventajas que de este hecho se derivan.

La actualización y recuperación de los datos debe realizarse mediante procesos bien determinados, incluidos en el SGBD, el cual ha de proporcionar también instrumentos que faciliten el mantenimiento de la seguridad (confidencialidad, disponibilidad e integridad) del conjunto de datos.

#### **1.3.2.1 Sistema de gestión de base de datos (SGBD)**

Es el conjunto de programas que permiten la implantación, acceso y mantenimiento de la base de datos. El SGBD, junto con la base de datos y con los usuarios, constituye el Sistema de Base de Datos.

El sistema de gestión de base de datos se puede definir como un conjunto coordinado de programas, procedimientos, lenguajes, etc., que suministra a los distintos tipos de usuarios los medios necesarios para describir y manipular los datos almacenados en la base de datos, garantizado su seguridad.

Las operaciones típicas que debe realizar un SGBD pueden resumirse en aquellas que afectan a lo totalidad de los datos (o a todos los registros de un determinado tipo) y las que tienen lugar sobre registros concretos.

Las operaciones típicas sobre una base de datos son:

- a) Sobre el conjunto de la base
  - a. Creación
  - b. Reestructuración
  - c. Consulta a la totalidad
- b) Sobre registros concretos
  - a. Inserción
  - b. Borrado
  - c. Modificación
  - d. Consulta selectiva

#### **1.3.2.2 Estructura de un sistema de base de datos**

En los sistemas de información existen dos estructuras distintas, la lógica (vista del usuario) y la física (forma en que se encuentran los datos almacenados). En las bases de datos aparece un nuevo nivel de abstracción que se ha denominado de diversas maneras: nivel conceptual, lógico global, etc. Esta estructura intermedia pretende una representación global de los datos que se interponga entre las estructuras lógica y física de la arquitectura a dos niveles, siendo independiente, tanto del equipo como de cada usuario en particular.

La estructura lógica de usuario o esquema externo es la visión que tiene de la base de datos, cada usuario en particular; la estructura lógica global (también denominada esquema conceptual) responde al enfoque del conjunto de la empresa y la estructura física (o esquema interno) es la forma en que se organizan los datos en el almacenamiento físico. La estructuración de una base de datos en estos tres niveles de abstracción tiene como principal objetivo conseguir la independencia entre datos y aplicaciones.

#### **1.3.2.3 Estructura lógica de usuario (esquema externo)**

Debido a que un esquema externo es la visión que de la base de datos tiene un usuario en particular, en él deberán encontrarse reflejados solo aquellos datos interrelacionados que necesite el correspondiente usuario. También habrá de especificarse las restricciones de uso, como puede ser el derecho de insertar, borrar determinados datos o el acceso a los mismos, etc. Así mismo, y aunque esto no sea lo mas conveniente, ya que indica una dependencia físico-lógica, puede que aparezcan en este nivel los caminos de acceso a los datos, hecho que dependerá en gran medida del modelo de datos en el que se apoya el correspondiente SGBD; en el modelo relacional los caminos de acceso solo se encuentran en el nivel interno, sin ser visibles por los usuarios.

#### **1.3.2.4 Estructura lógica global (esquema conceptual)**

En el esquema conceptual, por ser la visión global de los datos, deberá incluirse la descripción de todos los datos e interrelaciones entre estos, así como las restricciones de integridad y de confidencialidad.

#### **1.3.2.5 Estrategia de almacenamiento**

Es la asignación de espacios de almacenamiento para el conjunto de datos.

#### **1.3.2.6 Tipos de usuarios**

Los tipos de usuarios de una base de datos pueden clasificarse en usuarios informáticos y usuarios finales.

- A) Usuarios Informáticos
  - Tienen a su cargo las tareas de creación y mantenimiento de la base de datos, así como la realización de los procedimientos y programas que necesiten los usuarios finales. Entre ellos se puede distinguir.
    - a. Diseñadores



Tienen la responsabilidad de identificar los datos que han de estar contenidos en la base de datos, de acuerdo con las necesidades que les manifiesten los usuarios, así como de determinar las estructuras más apropiadas para conseguir satisfacer estas necesidades. Dependiendo de la fase en que intervengan los diseñadores, se pueden dividir en:

**Diseñadores Lógicos:** Son capaces de determinar que tipos de datos han de estar contenidos en la base y conseguir plasmar el punto de vista del conjunto de usuarios en las estructuras lógicas más adecuadas. Para lograrlo deben mantener constantes entrevistas con los usuarios a fin de que la base de datos represente lo más fielmente posible el mundo real que se trata de capturar. Es decir los diseñadores lógicos deben perseguir el objetivo de eficacia de la base de datos.

**Diseñadores Físicos:** Su objetivo es transformar las estructuras lógicas en estructuras físicas que proporcionen la mayor eficiencia de cara a la máquina, minimizando el tiempo de respuesta y el consumo de recursos, es decir, optimizar el ratio costo/beneficio. Muchas veces esta función es asumida por el administrador.

**b. Administradores**

El uso compartido de recursos, propio de una base de datos, obliga a la existencia de un administrador, cuya misión es la vigilancia y gestión de los datos. El principal recurso en una base de datos son los datos, y el administrador debe velar para que estos no se destruyan ni se contaminen, perdiendo su confidencialidad, disponibilidad e integridad. Por ello el administrador debe impedir consultas o actualizaciones no autorizadas y proteger la base de datos contra fallos del equipo lógico o físico. Será el responsable, por lo tanto, de establecer el sistema de autorizaciones de acceso y deberá coordinar y controlar su uso.

También el administrador tendrá a su cargo la gestión de otros recursos distintos de los datos, como pueden ser el SGBD y otras herramientas relacionadas con el mismo. Deberá ocuparse del buen funcionamiento de todo el sistema, sin que se produzcan paradas así como de que se proporcionen los tiempos adecuados de respuesta.

**c. Analistas y programadores**

Los analistas y programadores tienen a su cargo el análisis y la programación de las tareas que no pueden ser llevadas a cabo por los usuarios finales, para lo cual han de desarrollar distintos procedimientos y programas que se ponen a disposición de los usuarios finales a fin de facilitarles su trabajo.

**B) Usuarios Finales.**

Los usuarios finales son aquellos que tienen que acceder a los datos porque los necesitan para llevar a cabo su actividad. A diferencia de los usuarios informáticos, su interés suele estar centrado en el contenido de la base de datos, es decir, en los datos. Existen también distintas clases de usuarios finales.

**C) Habituales**

Suelen hacer actualizaciones en la base de datos como parte habitual de su trabajo. Utilizan en general menús previamente preparados por analistas y/o programadores, de forma que se les facilite su interrelación con la computadora. En otros casos pueden usar lenguajes sencillos para el acceso a la base de datos, así como paquetes de programas.

Es preciso distinguir, dentro de este grupo, a los operadores de entrada de datos, cuya labor consiste en actualizar la base de datos para lo que se les prepara menús de actualización, aunque con la diferencia respecto a otros usuarios habituales que

tienen exigencias muy estrictas respecto a los tiempos de respuesta, además de no estar interesados en la información contenida en la base de datos.

D) **Esporádicos**

Es un tipo de usuario muy parecido al anterior en la medida en que necesitan la computadora a fin de que les preste una ayuda en su trabajo, pero en cambio no la utilizan habitualmente porque el tipo de actividad que realizan no lo exige. Son, por lo tanto, usuarios a los que se les tiene que suministrar herramientas sencillas y, en general, potentes, ya que en bastantes casos así lo exige la clase de tareas que han de llevar a cabo; tareas que, en muchas ocasiones, no son fácilmente formalizables, por lo que no pueden ser atendidas por medio de menús.

### 1.3.2.7 Historia de los sistemas de bases de datos

Los sistemas de bases de datos tienen sus raíces en el proyecto estadounidense "Apolo" (mandar al hombre a la luna) en los años sesenta. En aquella época, no había ningún sistema que permitiera gestionar la inmensa cantidad de información que requería el proyecto. La primera empresa encargada del proyecto, NAA (North American Aviation), desarrolló un software denominado GUAM (General Update Access Method) que estaba basado en el concepto de que varias piezas pequeñas se unen para formar una pieza más grande, y así sucesivamente hasta que el producto final está ensamblado. Esta estructura, que tiene la forma de un árbol, es lo que se denomina una estructura jerárquica. A mediados de los sesenta, IBM se unió a NAA para desarrollar GUAM en lo que ahora se conoce como IMS (Information Management System). El motivo por el cual IBM restringió IMS al manejo de jerarquías de registros fue el de permitir el uso de dispositivos de almacenamiento serie, más exactamente las cintas magnéticas, ya que era un requisito del mercado por aquella época.

A mitad de los sesenta, se desarrolló IDS (Integrated Data Store), de General Electric. Este trabajo fue dirigido por uno de los pioneros en los sistemas de bases de datos, Charles Bachmann. IDS era un nuevo tipo de sistema de bases de datos conocido como sistema de red, que produjo un gran efecto sobre los sistemas de información de aquella generación. El sistema de red se desarrolló, en parte, para satisfacer la necesidad de representar relaciones entre datos más complejos que las que se podían modelar con los sistemas jerárquicos, y en parte, para imponer un estándar de bases de datos. Para ayudar a establecer dicho estándar, CODASYL (Conference on Data Systems Languages), formado por representantes del gobierno de EEUU y representantes del mundo empresarial, formaron un grupo denominado DBTG (Data Base Task Group), cuyo objetivo era definir unas especificaciones estándar que permitieran la creación de bases de datos y el manejo de los datos. El DBTG presentó su informe final en 1971 y aunque éste no fue formalmente aceptado por ANSI (American National Standards Institute), muchos sistemas se desarrollaron siguiendo la propuesta del DBTG. Estos sistemas son los que se conocen como sistemas de red, o sistemas CODASYL o DBTG.

Los sistemas jerárquico y de red constituyen la primera generación de los SGBD. Pero estos sistemas presentan algunos inconvenientes:

- Es necesario escribir complejos programas de aplicación para responder a cualquier tipo de consulta de datos, por simple que ésta sea.
- La independencia de datos es mínima.
- No tienen un fundamento teórico.

En 1970 Codd, de los laboratorios de investigación de IBM, escribió un artículo presentando el modelo relacional. En este artículo, presentaba también los inconvenientes de los sistemas previos, el jerárquico y el de red. Entonces, se comenzaron a desarrollar muchos sistemas relacionales, apareciendo los primeros a finales de los setenta y principios de los ochenta. Uno de los primeros es System R, de IBM, que se desarrolló para probar la funcionalidad del modelo relacional, proporcionando una implementación de sus estructuras de datos y sus operaciones. Esto condujo a dos grandes desarrollos:

- El desarrollo de un lenguaje de consultas estructurado denominado SQL, que se ha convertido en el lenguaje estándar de los sistemas relacionales.
- La producción de varios SGBD relacionales durante los años ochenta, como DB2 y SLQ/DS de IBM, y ORACLE de ORACLE Corporation.

Hoy en día, existen cientos de SGBD relacionales, tanto para microcomputadoras como para sistemas multiusuario, aunque muchos no son completamente fieles al modelo relacional. Otros sistemas relacionales multiusuario son INGRES de Computer Associates, Informix de Informix Software Inc. y Sybase de Sybase Inc. Ejemplos de sistemas relacionales de microcomputadoras son Paradox y dBase IV de Borland, Access de Microsoft, FoxPro y R:base de Microrim.

Los SGBD relacionales constituyen la segunda generación de los SGBD. Sin embargo, el modelo relacional también tiene sus fallos, siendo uno de ellos su limitada capacidad al modelar los datos. Se ha hecho mucha investigación desde entonces tratando de resolver este problema. En 1976, Chen presentó el modelo entidad-relación, que es la técnica más utilizada en el diseño de bases de datos. En 1979, Codd intentó subsanar algunas de las deficiencias de su modelo relacional con una versión extendida denominada RM/T (1979) y más recientemente RM/V2 (1990). Los intentos de proporcionar un modelo de datos que represente al mundo real de un modo más fiel han dado lugar a los modelos de datos semánticos. Como respuesta a la creciente complejidad de las aplicaciones que requieren bases de datos, han surgido dos nuevos modelos: el modelo de datos orientado a objetos y el modelo relacional extendido. Sin embargo, a diferencia de los modelos que los preceden, la composición de estos modelos no está clara. Esta evolución representa la tercera generación de los SGBD.

#### 1.3.2.8 Base de Datos ORACLE

Oracle es un sistema de gestión de base de datos relacional (o RDBMS por el acrónimo en inglés de Relational Data Base Management System), fabricado por Oracle Corporation. Se considera a Oracle como uno de los sistemas de bases de datos más completos, destacando por:

- Soporte de transacciones.
- Estabilidad.
- Escalabilidad.
- Multiplataforma.

Su mayor defecto es su enorme precio, que es de varios miles de pesos (según versiones y licencias). Otro aspecto que ha sido criticado por algunos especialistas es la seguridad de la plataforma, y las políticas de suministro de parches de seguridad, modificadas a comienzos de 2005 y que incrementan el nivel de exposición de los usuarios. En los parches de actualización provistos durante el primer semestre de 2005 fueron corregidas 22 vulnerabilidades públicamente conocidas, algunas de ellas con una antigüedad de más de 2 años.

Aunque su dominio en el mercado de servidores empresariales ha sido casi total hasta hace poco, recientemente sufre la competencia del Microsoft SQL Server de Microsoft y de la oferta de otros RDBMS con licencia libre como PostgreSQL, MySQL o Firebird. Las últimas versiones de Oracle han sido certificadas para poder trabajar bajo Linux

Oracle surge a finales de los 70 bajo el nombre de Relational Software a partir de un estudio sobre SGBD (Sistemas Gestores de Base de Datos) de George Koch. Computer World definió este estudio como uno de los más completos jamás escritos sobre bases de datos. Este artículo incluía una comparativa de productos que erigía a Relational Software como el más completo desde el punto de vista técnico. Esto se debía a que usaba la filosofía de las bases de datos relacionales, algo que por aquella época era todavía desconocido.

En la actualidad, Oracle todavía encabeza la lista. La tecnología Oracle se encuentra prácticamente en todas las industrias alrededor del mundo y en las oficinas de 98 de las 100 empresas Fortune 100. Oracle es la primera compañía de software que desarrolla e

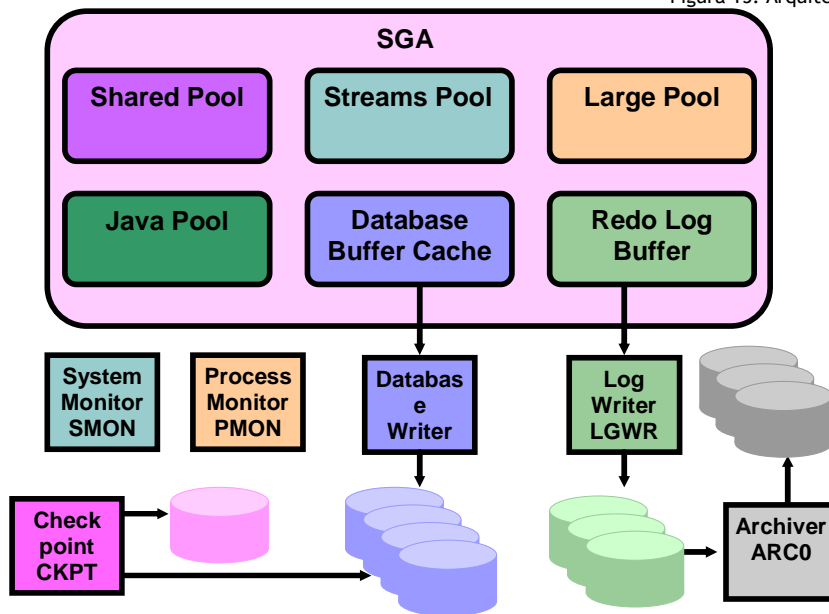
implementa software para empresas 100 por ciento activado por Internet a través de toda su línea de productos: base de datos, aplicaciones comerciales y herramientas de desarrollo de aplicaciones y soporte de decisiones. Oracle es el proveedor mundial líder de software para administración de información, y la segunda empresa de software independiente más grande del mundo.

Una Base de Datos Oracle es un conjunto de datos almacenado y accesible según el formato de tablas relacionales. Los datos están almacenados en filas. Las tablas pueden estar relacionadas con otras.

Una Base de Datos Oracle está almacenada físicamente en archivos, y la correspondencia entre los archivos y las tablas es posible gracias a las estructuras internas de la BD, que permiten que diferentes tipos de datos estén almacenados y físicamente separados. Está división lógica se hace gracias a los espacios de tablas, *tablespaces*.

### 1.3.2.8.1 Componentes de ORACLE

Figura 15: Arquitectura Base de Datos Oracle



#### 1.3.2.8.1.1 Memoria SGA

Todas las bases de datos Oracle están asociadas a un espacio de memoria, este espacio de memoria es apartado previamente por el sistema y es usado por la base de datos, para administrar y controlar los requerimientos y los procesos de la base de datos. A este espacio en memoria se le llama Memoria SGA.

Una vez que la memoria es establecida Oracle inicia una serie de procesos para completar los requerimientos del usuario, a este grupo se le llama PGA. La memoria SGA y los correspondientes procesos de oracle (PGA) conforman una instancia de Base de Datos ORACLE.

Los parámetros que determinan el tamaño y composición de una instancia están almacenados en un archivo llamado *init.ora*. Este archivo es leído durante el arranque de la BD y puede ser modificado por el DBA. Cualquier modificación de este archivo no tiene efecto hasta la siguiente vez que se arranque la BD.

Las estructuras de la BD Oracle pueden ser divididas en tres clases:

- Aquellas que son internas a la BD,

- Aquellas que son internas a las áreas de memoria (incluidas la memoria compartida y procesos),
- Aquellas que son externas a la BD.

La memoria SGA se divide en varias partes, las más importantes son:

#### 1.3.2.8.1.1 Buffers de BD, Database Buffer Cache

Es el cache que almacena los bloques de datos leídos de los segmentos de datos de la BD, tales como tablas, índices y clusters. Los bloques modificados se llaman *bloques sucios*. El tamaño de buffer cache se fija por el parámetro DB\_BLOCK\_BUFFERS del archivo init.ora. Como el tamaño del buffer suele ser pequeño para almacenar todos los bloques de datos leídos, su gestión se hace mediante el algoritmo LRU (Least Recently Used).

#### 1.3.2.8.1.2 Buffer Redo Log

Los registros *Redo* describen los cambios realizados en la BD y son escritos en los archivos *redo log* para que puedan ser utilizados en las operaciones de recuperación hacia adelante, *roll-forward*, durante las recuperaciones de la BD. Antes de ser escritos en los archivos *redo log* son escritos en un caché de la SGA llamado *redo log buffer*. El servidor escribe periódicamente los registros *redo log* en los archivos *redo log*. El tamaño del buffer redo log se fija por el parámetro LOG\_BUFFER.

#### 1.3.2.8.1.3 Área de SQL Compartido, Shared SQL Pool

En esta zona se encuentran las sentencias SQL que han sido analizadas. El análisis sintáctico de las sentencias SQL lleva su tiempo y Oracle mantiene las estructuras asociadas a cada sentencia SQL analizada durante el tiempo que pueda para ver si puede reutilizarlas. Antes de analizar una sentencia SQL, Oracle revisa si encuentra otra sentencia exactamente igual en la zona de SQL compartido. Si es así, no la analiza y pasa directamente a ejecutar la que mantiene en memoria. De esta manera se premia la uniformidad en la programación de las aplicaciones.

La igualdad se entiende que es lexicográfica, espacios en blanco y variables incluidas. El contenido de la zona de SQL compartido es:

- Plan de ejecución de la sentencia SQL.
- Texto de la sentencia.
- Lista de objetos referenciados.

Los pasos de procesamiento de cada petición de análisis de una sentencia SQL son:

- Comprobar si la sentencia se encuentra en el área compartida.
- Comprobar si los objetos referenciados son los mismos.
- Comprobar si el usuario tiene acceso a los objetos referenciados.

Si no, la sentencia es nueva, se analiza y los datos de análisis se almacenan en la zona de SQL compartida.

También se almacena en la zona de SQL compartido el *caché del diccionario*. La información sobre los objetos de la BD se encuentra almacenada en las tablas del diccionario. Cuando esta información se necesita, se leen las tablas del diccionario y su información se guarda en el cache del diccionario de la SGA.

Este cache también se administra mediante el algoritmo LRU. El tamaño del cache está gestionado internamente por el servidor, pero es parte del *shared pool*, cuyo tamaño viene determinado por el parámetro SHARED\_POOL\_SIZE.

#### Área Global de Programa

El *Program Global Area* es un área de memoria utilizada por un proceso Oracle. Esta zona de memoria no se puede compartir.

#### 1.3.2.8.1.2 Estructuras de Proceso

El servidor se vale de una serie de procesos que son el enlace entre las estructuras físicas y de memoria.

#### 1.3.2.8.1.3 System Monitor, SMON

El proceso SMON es el supervisor del sistema y se encarga de todas las recuperaciones que sean necesarias durante el arranque. Esto puede ser necesario si la BD se paró inesperadamente por fallo físico, lógico u otras causas. Este proceso realiza la recuperación de la instancia de BD a partir de los archivos *redo log*. Además limpia los segmentos temporales no utilizados y compacta los huecos libres contiguos en los archivos de datos. Este proceso se despierta regularmente para comprobar si debe intervenir.

#### 1.3.2.8.1.4 Process Monitor, PMON

Este proceso restaura las transacciones no validadas de los procesos de usuario que abortan, liberando los bloqueos y los recursos de la SGA. Asume la identidad del usuario que ha fallado, liberando todos los recursos de la BD que estuviera utilizando, y anula la transacción cancelada. Este proceso se despierta regularmente para comprobar si su intervención es necesaria.

#### 1.3.2.8.1.5 Database Writer, DBWR

El proceso DBWR es el responsable de administrar el contenido de los buffers de datos y del caché del diccionario de datos. Este proceso lee los bloques de los archivos de datos y los almacena en la SGA. Luego escribe en los archivos de datos los bloques cuyo contenido ha variado. La escritura de los bloques a disco es diferida buscando mejorar la eficiencia de la E/S.

Es el único proceso que puede escribir en la BD. Esto asegura la integridad. Se encarga de escribir los bloques de datos modificados por las transacciones, tomando la información del *buffer* de la BD cuando se valida una transacción. Cada validación no se lleva a la BD física de manera inmediata sino que los bloques de la BD modificados se vuelcan a los archivos de datos periódicamente o cuando sucede algún *checkpoint* o punto de sincronización: grabación *diferida*:

- Los bloques del *buffer* de la BD (bloques del segmento de *rollback* y bloques de datos) menos recientemente utilizados son volcados en el disco continuamente para dejar sitio a los nuevos bloques.
- El bloque del segmento de *rollback* se escribe SIEMPRE antes que el correspondiente bloque de datos.
- Múltiples transacciones pueden solapar los cambios en un sólo bloque antes de escribirlo en el disco.

Mientras, para que se mantenga la integridad y coherencia de la BD, todas las operaciones se guardan en los archivos de *redo log*. El proceso de escritura es asíncrono y puede realizar grabaciones multibloque para aumentar la velocidad.

#### 1.3.2.8.1.6 Log Writer, LGWR

El proceso LGWR es el encargado de escribir los registros *redo log* en los archivos *redo log*. Los registros *redo log* siempre contienen el estado más reciente de la BD, ya que puede que el DBWR deba esperar para escribir los bloques modificados desde el buffer de datos a los archivos de datos.

Conviene tener en cuenta que el LGWR es el único proceso que escribe en los archivos de *redo log* y el único que lee directamente los buffers de *redo log* durante el funcionamiento normal de la BD.

Coloca la información de los *redo log buffers* en los archivos de *redo log*. Los *redo log buffers* almacenan una copia de las transacciones que se llevan a cabo en la BD. Esto se produce:

- A cada validación de transacción, y antes de que se comunique al proceso que todo ha ido bien
- Cuando se llena el grupo de *buffers de redo log*
- Cuando el DBWR escribe *buffers* de datos modificados en disco

Así, aunque los archivos de DB no se actualicen en ese instante con los buffers de BD, la operación queda guardada y se puede reproducir. Oracle no tiene que consumir sus recursos escribiendo el resultado de las modificaciones de los datos en los archivos de datos de manera inmediata. Esto se hace porque los registros de *redo log* casi siempre tendrán un tamaño menor que los bloques afectados por las modificaciones de una transacción, y por lo tanto el tiempo que emplea en guardarlos es menor que el que emplearía en almacenar los bloques sucios resultado de una transacción, que ya serán trasladados a los archivos por el DBWR. El LGWR es un proceso único, para asegurar la integridad. Es asíncrono. Además permite las grabaciones multibloque.

#### 1.3.2.8.1.7 Checkpoint, CKPT

Este proceso escribe en los archivos de control los *checkpoints*. Estos puntos de sincronización son referencias al estado coherente de todos los archivos de la BD en un instante determinado, en un punto de sincronización. Esto significa que los bloques sucios de la BD se vuelcan a los archivos de BD, asegurándose de que todos los bloques de datos modificados desde el último *checkpoint* se escriben realmente en los archivos de datos y no sólo en los archivos *redo log*; y que los archivos de *redo log* también almacenan los registros de *redo log* hasta este instante. La secuencia de puntos de control se almacena en los archivos de datos, *redo log* y control. Los *checkpoints* se producen cuando:

- Un espacio de tabla se pone inactivo, *offline*,
- Se llena el archivo de *redo log* activo,
- Se para la BD,
- El número de bloques escritos en el *redo log* desde el último *checkpoint* alcanza el límite definido en el parámetro LOG\_CHECKPOINT\_INTERVAL,
- Cuando transcurra el número de segundos indicado por el parámetro LOG\_CHECKPOINT\_TIMEOUT desde el último *checkpoint*.

Está activo si el parámetro CHECKPOINT\_PROCESS tiene un valor verdadero.

#### 1.3.2.8.1.8 Archiver, ARCH

El proceso archivador tiene que ver con los archivos *redo log*. Por defecto, estos archivos se reutilizan de manera cíclica de modo que se van perdiendo los registros *redo log* que tienen una cierta antigüedad. Cuando la BD se ejecuta en modo ARCHIVELOG, antes de reutilizar un archivo *redo log* realiza una copia del mismo. De esta manera se mantiene una copia de todos los registros *redo log* por si fueran necesarios para una recuperación. Este es el trabajo del proceso archivador.

#### 1.3.2.8.1.9 Los Espacios de Tablas (*Tablespaces*)

Un *tablespace* es una división lógica de la BD. Cada BD tiene al menos uno (SYSTEM). Un *tablespace* puede pertenecer sólo a una BD. Estos se utilizan para mantener juntos los datos de usuarios o de aplicaciones para facilitar su mantenimiento o mejorar las prestaciones del sistema.

De esta manera, cuando se crea una tabla se debe indicar el *tablespace* donde se almacenara la tabla. Por defecto se depositan en el *tablespace* SYSTEM, que se crea por defecto. Este

tablespace es el que contiene el diccionario de datos, por lo que conviene reservarlo para el uso del servidor, y asignar las tablas de los usuarios en otro tablespace.

Lo razonable y aconsejable es que cada aplicación tenga su propio tablespace. Hay varias razones que justifican este modo de organización de los tablespaces:

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

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

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

```
SQL> select * from user_tablespaces;
```

Dentro de cada espacio de tabla se pueden almacenar objetos de distinta naturaleza: tablas, índices, etc., pero no se pueden mezclar. Necesitamos una manera de separarlos, y eso son los *segmentos*.

Se pueden almacenar más de un segmento por espacio de tabla. Un segmento está contenido en su totalidad en un tablespace. Un segmento está constituido por un conjunto de extensiones, que no son más que grupos de bloques de disco ORACLE contiguos. Cuando se borra un segmento, el espacio es devuelto al tablespace.

Todos los datos de la BD están almacenados en segmentos. Existen 5 tipos de segmentos:

- De datos: almacenan las tablas.
- De índices: permiten un acceso rápido a los datos dependiendo de la cantidad de los mismos (árboles B). Las consultas que sólo referencian a columnas indexadas se resuelven en el índice. Establecen un control de unicidad (los índices son automáticos cuando se definen claves primarias). Cada índice ocupa un segmento independiente del segmento de datos y deberían estar en un espacio de tablas distinto al de los datos, para mejorar el rendimiento.
- De *rollback*: son objetos internos de la BD que permiten efectuar la restauración de las transacciones no validadas asegurando la consistencia en lectura. La estructura de los registros de *rollback* es:
  - Identificador de la transacción.
  - Dirección del bloque donde está la tabla.
  - Número de fila.
  - Número de columna.
  - Valor del dato antiguo (antes de ser modificado).

Son tan importantes que una BD no puede iniciar si no puede acceder al menos a un segmento de *rollback*. Si la BD tiene múltiples espacios de tablas, deben existir al menos dos segmentos de *rollback* y cada segmento de *rollback* debe tener al menos dos extensiones, reutilizables



de manera cíclica. Estos segmentos son un objeto compartido de la BD, aunque se puede aginar un segmento de *rollback* particular a una transacción dada.

- Temporales: son creados por Oracle para un uso temporal cuando debe realizar una ordenación que no cabe en memoria, y en las operaciones: *create index*, *order by*, *group by*, *distinct*, *union*, *intersect*, *minus*. Son eliminados cuando la sentencia finaliza.
- De *bootstrap*: Se crea en SYSTEM y contiene definiciones del diccionario para sus tablas, que se cargan al abrir la BD. No requiere ninguna acción por parte del DBA. No cambia de tamaño.

La tabla que guarda la información de los segmentos de usuario es *user\_segments*, y se puede visualizar la información sobre los segmentos con la sentencia SQL siguiente:

```
SQL> select * from user_segments;
```

#### 1.3.2.8.1.10 Archivos (Archives)

Cada tablespace se compone de uno o más archivos en disco. Un archivo puede pertenecer sólo a un tablespace. Los archivos reciben un tamaño fijo en el momento de su creación, y cuando se necesita más espacio se deben añadir más archivos al tablespace.

Dividir los objetos de la BD entre múltiples tablespaces permite que los objetos sean almacenados físicamente en discos separados, dependiendo de donde estén los archivos sobre los que se asientan.

#### 1.3.2.8.1.11 Estructuras Internas de la BD

##### 1.3.2.8.1.11.1 Tablas y Columnas

Los datos son almacenados en la BD utilizando tablas. Cada tabla está compuesta por un número determinado de columnas.

Las tablas propiedad del usuario SYS son llamadas tablas del diccionario de datos. Proveen el catálogo del sistema que permite que la BD se administre a sí misma.

Las tablas se pueden relacionar entre ellas a través de las columnas que las componen. La BD se puede utilizar para asegurar el cumplimiento de esas relaciones a través de la integridad referencial, que se concreta en las restricciones de tablas.

##### 1.3.2.8.1.11.2 Restricciones de Tablas

Una tabla puede tener asociadas restricciones que deben cumplir todas las filas. Entre las restricciones que se pueden fijar algunas reciben nombres especiales.: *clave primaria*, *clave ajena*.

La clave primaria de una tabla está compuesta por las columnas que hacen a cada fila de la tabla una fila distinta.

La clave ajena se utiliza para especificar las relaciones entre tablas, de modo que un conjunto de columnas declaradas como clave ajena de una tabla deben tener valores tomados de la clave primaria de otra tabla.

##### 1.3.2.8.1.11.3 Usuarios

Una cuenta de usuario no es una estructura física de la BD, pero está relacionada con los objetos de la BD: los usuarios poseen los objetos de la BD. Existen dos usuarios especiales: SYS y SYSTEM. El usuario SYS posee las tablas del diccionario de datos que almacenan información sobre el resto de las estructuras de la BD. El usuario SYSTEM posee las vistas que permiten acceder a las tablas del diccionario para el uso del resto de los usuarios de la BD.

Todo objeto creado en la BD se crea por un usuario, en un espacio de tablas y en un archivo de datos determinado. Toda cuenta de la BD puede estar unida a una cuenta del S.O., lo que permite a los usuarios acceder a la cuenta de la BD sin dar la clave de acceso.

Cada usuario puede acceder a los objetos que posea o a aquellos sobre los que tenga derecho de acceso.

#### **1.3.2.8.1.11.4 Esquemas**

El conjunto de objetos de un usuario es conocido como esquema.

#### **1.3.2.8.1.11.5 Índices**

Un índice es una estructura de la BD utilizada para agilizar el acceso a una fila de una tabla. Cada fila tiene un identificador de fila, ROWID, que determina el archivo, bloque y fila dentro del bloque donde está almacenada la fila.

Cada entrada del índice consiste en un valor clave y una ROWID. Cada una de estas entradas se almacena en un árbol B<sup>+</sup>.

Los índices se crean automáticamente cuando se define una restricción UNIQUE o PRIMARY KEY.

#### **1.3.2.8.1.11.6 Clusters**

Las tablas que son accedidas juntas frecuentemente pueden ser almacenadas juntas. Para ello se crea un *cluster*. De este modo se minimiza el número de E/S.

Las columnas que relacionan las tablas de un *cluster* se llaman clave del *cluster*.

#### **1.3.2.8.1.11.7 Vistas**

Conceptualmente, una vista puede considerarse como una máscara que se extiende sobre una o más tablas, de modo que cada columna de la vista se corresponde con una o más columnas de las tablas subyacentes. Cuando se consulta una vista, esta traspasa la consulta a las tablas sobre las que se asienta. Las vistas no se pueden indexar.

Las vistas no generan almacenamiento de datos, y sus definiciones se almacenan en el diccionario de datos.

#### **1.3.2.8.1.11.8 Secuencias**

Las definiciones de secuencias se almacenan en el diccionario de datos. Son mecanismos para obtener listas de números secuenciales.

#### **1.3.2.8.1.11.9 Procedimientos y Funciones**

Un procedimiento es un bloque de código PL/SQL, que se almacena en el diccionario de datos y que es llamado por las aplicaciones. Se pueden utilizar para implementar seguridad, no dando acceso directamente a determinadas tablas sino es a través de procedimientos que acceden a esas tablas. Cuando se ejecuta un procedimiento se ejecuta con los privilegios del propietario del procedimiento. La diferencia entre un procedimiento y una función es que ésta última puede devolver valores.

#### **1.3.2.8.1.11.10 Paquetes, Packages**

Se utilizan para agrupar procedimientos y funciones. Los elementos dentro de los paquetes pueden ser públicos o privados. Los públicos pueden ser llamados por los usuarios, los privados están ocultos a los usuarios y son llamados por otros procedimientos.

#### **1.3.2.8.1.11.11 Disparadores, Triggers**

Son procedimientos que son ejecutados cuando se produce un determinado evento en la BD. Se pueden utilizar para mejorar y reforzar la integridad y la seguridad de la BD.

#### 1.3.2.8.1.11.12 Sinónimos

Para identificar completamente un objeto dentro de una BD se necesita especificar el nombre de la máquina, el nombre del servidor, el nombre del propietario y el nombre del objeto. Para hacer transparente todo esto al usuario se pueden utilizar los sinónimos. Éstos apuntarán a los objetos y si el objeto cambia de lugar o propietario, sólo habrá que modificar el sinónimo.

Existen sinónimos públicos y privados. Los públicos son conocidos por todos los usuarios de una BD. Los privados son locales a un usuario.

#### 1.3.2.8.1.11.13 Privilegios y Roles

Para que un objeto pueda ser accedido por un usuario debe de tener otorgado ese privilegio. Ejemplos de privilegios son INSERT, SELECT, UPDATE, EXECUTE, etc.

Los roles son grupos de privilegios que pueden ser utilizados para facilitar la gestión de los privilegios. Los privilegios se pueden otorgar a un rol, y los roles pueden ser otorgados a múltiples usuarios.

#### 1.3.2.8.1.11.14 Segmentos, Extensiones y Bloques

Los segmentos son los equivalentes físicos de los objetos que almacenan datos. El uso efectivo de los segmentos requiere que el DBA conozca los objetos se que utilizan en una aplicación, cómo los datos son introducidos en esos objetos y el modo en que serán recuperados.

Como los segmentos son entidades físicas, deben estar asignados a tablespaces en la BD y estarán localizados en uno de los archivos de datos del tablespace. Un segmento está constituido por secciones llamadas extensiones, que son conjuntos contiguos de bloques Oracle. Una vez que una extensión existente en un segmento no puede almacenar más datos, el segmento obtendrá del espacio de tabla otra extensión. Este proceso de extensión continuará hasta que no quede más espacio disponible en los archivos del tablespace, o hasta que se alcance un número máximo de extensiones por segmento.

#### 1.3.2.8.1.11.15 Segmento de *Rollback*

Para mantener la consistencia en lectura y permitir deshacer las transacciones, Oracle debe tener un mecanismo para reconstruir la imagen previa a una transacción incompleta. Oracle utiliza los segmentos de *rollback* para esta función.

Los segmentos de *rollback* pueden crecer tanto como sea necesario para soportar las transacciones.

#### 1.3.2.8.1.12 Estructuras Externas

Por estructuras externas se entienden los archivos que utiliza el servidor de BD, de los cuales ya se han contemplado algunos aspectos, y otros se han ido intuyendo. Estos archivos guardan información tanto de los datos almacenados en la BD como la necesaria para gobernar la propia BD.

##### 1.3.2.8.1.12.1 Archivos *redo log*

En ellos se graba toda operación que se efectúe en la BD y sirven de salvaguarda de la misma. Tiene que haber por lo menos 2, uno de ellos debe estar activo, *online*, y se escribe en ellos de forma cíclica. Existe la posibilidad de almacenar los distintos archivos de *redo log* en el tiempo mediante el modo ARCHIVER. Así, se puede guardar toda la evolución de la BD desde un punto dado del tiempo.

Una opción es la utilización de archivos *redo log* multiplexados:

- Permite al LGWR escribir simultáneamente la misma información en múltiples archivos *redo log*.
- Se utiliza para protegerse contra fallos en el disco.
- Da una alta disponibilidad a los archivos *redo log* activos u *online*.

Esto se lleva a cabo definiendo el número de *grupos* y de *miembros* de archivos *redo log* que van a funcionar en paralelo:

- Grupos: funcionan como archivos *redo log* normales, uno de ellos está activo y el resto espera su turno.
  - Su nombre lleva incorporado una numeración.
  - Deben contener todos los mismos números de miembros.
- Miembros: cada escritura de un registro *redo log* se lleva a cabo en todos los miembros del grupo activo en ese momento. Los miembros deben:
  - Tener el mismo tamaño y el mismo número de secuencia.
  - Nombres similares y estar en diferentes discos para proteger contra fallos de una manera efectiva.

Cuando se produce algún fallo en los archivos de *redo log* o en el proceso LGWR:

- Si la escritura en un archivo *redo log* falla pero el LGWR puede escribir al menos en uno de los miembros del grupo, lo hace, ignorando el archivo inaccesible y registrando un fallo en un archivo de traza o alerta.
- Si el siguiente grupo no ha sido archivado (modo ARCHIVELOG) antes del cambio de grupo que lo pone activo, ORACLE espera hasta que se produzca el archivado.
- Si fallan todos los miembros de un grupo mientras el LGWR trata de escribir, la instancia se para y necesita recuperación al arrancar.

Se pueden visualizar los nombres y estado de los archivos de *redo log*:

```
SQL> select group#, status, substr(member,1,60) from v$logfile;
```

También se pueden visualizar estadísticas de los archivos *redo log*:

```
SQL> select group#, sequence#, bytes, members, archived,  
2 status, first_change#, first_time from v$logfile;
```

#### 1.3.2.8.1.12.2 Archivos de control

Mantienen la información física de todos los archivos que forman la BD, camino incluido; así como el estado actual de la BD. Son utilizados para mantener la consistencia interna y guiar las operaciones de recuperación. Son imprescindibles para que la BD se pueda arrancar.

Contienen:

- Información de arranque y parada de la BD.
- Nombres de los archivos de la BD y *redo log*.
- Información sobre los *checkpoints*.
- Fecha de creación y nombre de la BD.
- Estado *online* y *offline* de los archivos.

Debe haber múltiples copias en distintos discos, mínimo dos, para protegerlos de los fallos de disco. La lista de los archivos de control se encuentra en el parámetro CONTROL\_FILES, que debe modificarse con la BD abajo.

Se puede componer una sentencia SQL que nos muestre todos los archivos asociados a una BD., a saber:

```
SQL> select 'control' tipo, substr(name,1,70) nombre from v$controlfile
```

```
2 union all
3 select 'datos' tipo, substr(name,1,70) nombre from v$datafile
4 union all
5 select 'redo log' tipo, substr(name,1,70) nombre from v$logfile
6 /
```

Hasta aquí los tipos de archivos que se suelen considerar fundamentales en la arquitectura del SGBD Oracle. Pero existen otros archivos, que aunque no forman parte de la arquitectura Oracle resultan importantes en el uso del SGBD.

#### 1.3.2.8.1.12.3 El Archivo INITORA

Como parte de la distribución software, Oracle provee de un archivo de parámetros de inicialización llamado init.ora. Este archivo contiene los parámetros del sistema Oracle y debe ser utilizado por el DBA para configurar el SGBD y adecuarlo a una determinada explotación. Oracle lee este archivo durante el proceso de arranque para determinar el tamaño de la SGA y encontrar los archivos de control, entre otros menesteres.

Como el archivo init.ora es fundamental para el arranque de la BD, debería ser copiado frecuentemente para protegerlo de posibles pérdidas.

#### 1.3.2.8.1.12.4 Archivos de Traza

Oracle crea archivos de texto llamados de traza para ayudar en la diagnosis de problemas y en el ajuste del SGBD. Cada proceso del servidor escribe en un archivo de traza asociado cuando es necesario. Los procesos de usuarios también pueden tener asociados archivos de traza. La situación de estos archivos de traza del sistema se especifica por el parámetro BACKGROUND\_DUMP\_DEST, y los de usuario por USER\_DUMP\_DEST. Oracle crea archivos de traza automáticamente cuando ocurre algún error.

Un parámetro muy frecuentemente utilizado por los desarrolladores Oracle es el SQL\_TRACE, que cuando está puesto a TRUE produce que toda sentencia SQL ejecutada genere información en los archivos de traza. Este parámetro se puede variar con el siguiente comando:

```
SQL> alter session set SQL_TRACE=TRUE;
Session Altered.
```

El directorio donde se depositan los archivos de traza debe de examinarse con regularidad para controlar el tamaño de los archivos allí depositados.

#### 1.3.2.9 SQL Server

Microsoft SQL Server es un sistema de gestión de bases de datos relacionales (SGBD) basada en el lenguaje SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea. Así se tienen las ventajas que a continuación se pueden describir.

Entre sus características figuran:

- Soporte de transacciones.
- Seguridad.
- Escalabilidad.
- Soporta procedimientos almacenados.
- Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información.
- Además permite administrar información de otros servidores de datos

Este sistema incluye una versión reducida, llamada MSDE con el mismo motor de base de datos pero orientado a proyectos más pequeños.

Microsoft SQL Server constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son Oracle o Sybase ASE.

Es común desarrollar proyectos completos complementando Microsoft SQL Server y Microsoft Access a través de los llamados ADP (Access Data Project). De esta forma se completa una potente base de datos (Microsoft SQL Server) con un entorno de desarrollo cómodo y de alto rendimiento (VBA Access) a través de la implementación de aplicaciones de dos capas mediante el uso de formularios Windows.

Para el desarrollo de aplicaciones más complejas (tres o más capas), Microsoft SQL Server incluye interfaces de acceso para la mayoría de las plataformas de desarrollo, incluyendo .NET

Microsoft SQL Server, al contrario de su más cercana competencia, no es multiplataforma, ya que sólo está disponible en Sistemas Operativos de Microsoft.

Algunas de las características que tiene SQL Server son las siguientes:

#### **Arquitectura RDBMS.**

- Arquitectura de servidor simétrico y paralelo con balanceo automático de carga en múltiples procesadores.
- Kernel multithread real para mejor rendimiento transaccional y escalabilidad.
- Soporte grandes bases de datos (VLDB) (+1 TB).
- Completo proceso transaccional interactivo con rollback automático y recuperación de roll-forward.
- Optimizador de consultas mejorado basado en costo.
- Checkpointing mejorado para un mejor throughput de datos y tiempo de respuesta.
- Soporte E/S asíncrono para acceso en paralelo a múltiples dispositivos de disco para un mejor throughput.
- Bloqueo a nivel página con escalación de bloqueos; resolución automática de deadlocks.

#### **Datos distribuidos y replicación.**

- Llamadas a procedimientos remotos servidor-a-servidor (procedimientos almacenados remotos).
- Replicación asíncrona o continua basada en registros, o sincronización planificada de tablas point-in-time.
- Configuración de replicación gráfica y características de gestión.
- Configuración de subscriptores ODBC, incluyendo IBM DB2, ORACLE, SYBASE y Microsoft Access.
- Distributed Transaction Coordinator, gestiona transacciones que involucran a dos o más servidores SQL (proceso Two Phase Commit 2PC) transparente.
- Replicación de tipos de datos Texto e Imagen.

#### **Data Warehousing y amplio soporte de datos.**

- Ejecución transaccional en paralelo, carga e indexación de datos, y verificación de integridad.
- Backup/restore en paralelo de alta velocidad que soporta hasta 32 discos o dispositivos de cinta.
- Las extensiones de consulta OLAP, CUBE, y ROLLUP permiten conjuntos de resultados multidimensionales.
- EXEC e INTO ahora permiten almacenar en tablas resultados desde procedimientos almacenados.
- Copias de seguridad en cinta y recuperación de tablas individuales.
- Reconstrucción de índice sin realizar dropping y recreación del índice.

### **Integración Internet y correo electrónico.**

- MAPI, permitiendo aplicaciones de flujo de trabajo y notificación de cambio de datos automática.
- Compatibilidad con Microsoft Internet Information Server y otros servidores Web populares.
- SQL Web Assistant, para el retorno automático de datos en formato HTML.
- Procedimientos almacenados para generar páginas HTML o actualizar datos en plantillas Web.
- Posibilidad de poblar automáticamente carpetas públicas de Microsoft Exchange con datos.

### **Gestión y administración centralizada de bases de datos.**

- SQL Enterprise Manager, una consola de gestión y motorización 32-bit visual basada en Windows.
- Un único punto de configuración y gestión de control de datos remotos.
- SQL Executive, planificador de trabajos y monitor para gestión proactiva de servidores distribuidos.
- Operaciones remotas desatendidas mediante un "agente inteligente" que incluye evento-alerta- respuesta.
- Scripts Visual Basic a través de SQL-Distributed Management Objects (SQL-DMO) basados en OLE.
- DBA Assistant, para el mantenimiento automático rutinario en una única tarea planificada.
- SQL Trace, para monitorizar consultas cliente-servidor mediante SQL almacenadas en archivos de registros.
- Soporte de MIB y traps SNMP para monitorizar SQL Server desde herramientas de gestión basadas en SNMP.

### **Disponibilidad, fiabilidad y tolerancia a fallos.**

- Mirroring de dispositivos de base de datos con failover automático para tolerancia a fallos de dispositivos.
- Copias de seguridad online desatendidas garantizando la consistencia de datos para la más alta disponibilidad.
- Contextos de usuario protegidos, que pueden aislar los fallos a un thread de un único usuario.
- Recuperación point-in-time para restaurar bases de datos o transacción logs en un intervalo de tiempo.
- Tolerancia a fallos de servidor, permitiendo failover automático a un servidor de backup o en espera.

### **Mejoras en programación y lenguaje.**

- Triggers, procedimientos almacenados (autoexec), disparador de eventos antes y después de conexiones.
- Procedimientos almacenados extendidos (funciones definidas por el usuario) utilizando C/C++.
- Cursores basados en el motor con scrolling hacia adelante y atrás; posicionamiento absoluto y relativo.
- Sentencias DLL permitidas dentro de transacciones.
- Transacciones distribuidas dentro de interfaces DB-Library, ODBC, Transact-SQL, XA y OLE Transaction.
- Procedimientos almacenados OLE Automation.

### **Seguridad.**

- Un único ID de login tanto para red como para la DB para mejorar la seguridad y facilitar la administración.
- Password y encriptación de datos en red para mejorar la seguridad.
- Encriptación de procedimientos almacenados para la integridad y seguridad de código de aplicación.
- Interoperabilidad e integración con desktops.
- API estándar DB-Library totalmente soportada: estándar ODBC Nivel 2 totalmente soportado como API nativa.
- Gateway Open Data Services (ODS) programable para acceso transparente a fuentes de datos externas.
- Gateways de Microsoft y de terceros para fuentes de datos relacionales y no-relacionales, incluyendo IBM DB2.

### 1.3.3 Sistema Operativo

Los sistemas operativos aparecieron a finales de los años 50, y durante veinte años experimentaron grandes cambios en conceptos y tecnología. Actualmente la teoría y la práctica de los sistemas operativos parece haber madurado y estabilizado. Los conceptos fundamentales están bien definidos y asentados, aunque indudablemente con la evolución del hardware, se requieren nuevos algoritmos que administren de forma eficiente la memoria, el CPU, interfaz de usuario, etc.

#### 1.3.3.1 ¿Que es un sistema operativo?

Es un programa que actúa como interfaz entre el usuario de un ordenador y el hardware del mismo, ofreciendo el entorno necesario para que el usuario pueda ejecutar programas. Así su principal objetivo es facilitar el uso del sistema informático. Como objetivo secundario es el empleo del hardware del equipo de cómputo de manera eficiente.

El sistema operativo es una parte muy importante en los sistemas computacionales, un sistema computacional se puede dividir en:

- Usuarios, generador de información a resolver por el sistema de cómputo.
- Hardware, el cual facilita los recursos básicos de la computación.
- Programas o Aplicaciones, los cuales definen como utilizar el hardware o recursos para solucionar los problemas de los usuarios.
- Sistema Operativo, el cual controla y coordina el uso del hardware por parte de distintos programas o aplicaciones.

#### 1.3.3.2 Tipos de Sistemas Operativos

Los sistemas operativos se clasifican en diferentes clasificaciones:

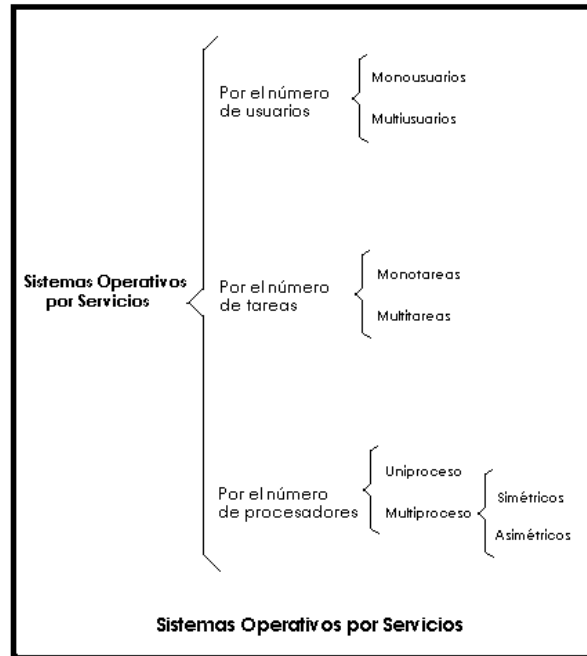
- Sistemas operativos por su estructura (visión interna)
- Sistemas operativos por la forma en que ofrecen sus servicios (visión externa).

#### 1.3.3.3 Sistemas Operativos por Servicios (Visión Externa)

Esta clasificación es la más comúnmente usada y conocida desde el punto de vista del usuario final. Esta clasificación se comprende fácilmente con el cuadro sinóptico que a continuación se muestra:



Figura 16: Sistemas Operativos por Servicios



#### 1.3.3.3.1 Por el número de Usuarios

##### 1.3.3.3.1.1 Monousuario

Los sistemas operativos monousuarios son aquéllos que soportan a un usuario a la vez, sin importar el número de procesadores que tenga la computadora o el número de procesos o tareas que el usuario pueda ejecutar en un mismo instante de tiempo. Las computadoras personales típicamente se han clasificado en este renglón.

En otras palabras los sistemas monousuarios son aquellos que solamente puede atender a un solo usuario, gracias a las limitaciones creadas por el hardware, los programas o el tipo de aplicación que se esté ejecutando.

##### 1.3.3.3.1.2 Multiusuario

Los sistemas operativos multiusuarios son capaces de dar servicio a más de un usuario a la vez, ya sea por medio de varias terminales conectadas a la computadora o por medio de sesiones remotas en una red de comunicaciones. No importa el número de procesadores en la máquina ni el número de procesos que cada usuario puede ejecutar simultáneamente.

En esta categoría se encuentran todos los sistemas que cumplen simultáneamente las necesidades de dos o más usuarios, que comparten los mismos recursos. Este tipo de sistemas se emplean especialmente en redes. En otras palabras consiste en el fraccionamiento del tiempo (timesharing).

#### 1.3.3.3.2 Por el Número de Tareas

##### 1.3.3.3.2.1 Monotarea

Los sistemas monotarea son aquellos que sólo permiten una tarea a la vez por usuario. Puede darse el caso de un sistema multiusuario y monotarea, en el cual se admiten varios usuarios al mismo tiempo pero cada uno de ellos puede estar haciendo solo una tarea a la vez.

Los sistemas operativos monotareas son más primitivos y solo pueden manejar un proceso en cada momento o solo puede ejecutar las tareas de una en una.

#### **1.3.3.3.2 Multitarea**

Un sistema operativo multitarea es aquel que le permite al usuario estar realizando varias labores al mismo tiempo.

Es el modo de funcionamiento disponible en algunos sistemas operativos, mediante el cual una computadora procesa varias tareas al mismo tiempo. Existen varios tipos de multitarea. La conmutación de contextos (context Switching) es un tipo muy simple de multitarea en el que dos o más aplicaciones se cargan al mismo tiempo, pero en el que solo se está procesando la aplicación que se encuentra en primer plano (la que ve el usuario. En la multitarea cooperativa, la que se utiliza en el sistema operativo Macintosh, las tareas en segundo plano reciben tiempo de procesamiento durante los tiempos muertos de la tarea que se encuentra en primer plano (por ejemplo, cuando esta aplicación está esperando información del usuario), y siempre que esta aplicación lo permita. En los sistemas multitarea de tiempo compartido, como OS/2, cada tarea recibe la atención del microprocesador durante una fracción de segundo.

Un sistema operativo multitarea puede estar editando el código fuente de un programa durante su depuración mientras compila otro programa, a la vez que está recibiendo correo electrónico en un proceso en background. Es común encontrar en ellos interfaces gráficas orientadas al uso de menús y el ratón, lo cual permite un rápido intercambio entre las tareas para el usuario, mejorando su productividad.

Un sistema operativo multitarea se distingue por su capacidad para soportar la ejecución concurrente de dos o más procesos activos.

La multitarea se implementa generalmente manteniendo el código y los datos de varios procesos simultáneamente en memoria y multiplexando el procesador y los dispositivos de E/S entre ellos.

La multitarea suele asociarse con soporte hardware y software para protección de memoria con el fin de evitar que procesos corrompan el espacio de direcciones y el comportamiento de otros procesos residentes.

#### **1.3.3.3.3 Por el Número de Procesadores**

##### **1.3.3.3.3.1 Uniproseso**

Un sistema operativo uniproseso es aquél que es capaz de manejar solamente un procesador de la computadora, de manera que si la computadora tuviese más de uno le sería inútil. El ejemplo más típico de este tipo de sistemas es el DOS y MacOS.

##### **1.3.3.3.3.2 Multiproseso**

Un sistema operativo multiproseso se refiere al número de procesadores del sistema, que es más de uno y éste es capaz de usarlos todos para distribuir su carga de trabajo. Generalmente estos sistemas trabajan de dos formas: simétrica o asimétricamente.

#### **Asimétrica.**

Cuando se trabaja de manera asimétrica, el sistema operativo selecciona a uno de los procesadores el cual jugará el papel de procesador maestro y servirá como pivote para distribuir la carga a los demás procesadores, que reciben el nombre de esclavos.

#### **Simétrica.**

Cuando se trabaja de manera simétrica, los procesos o partes de ellos (threads) son

enviados indistintamente a cualquiera de los procesadores disponibles, teniendo, teóricamente, una mejor distribución y equilibrio en la carga de trabajo bajo este esquema.

Se dice que un thread es la parte activa en memoria y corriendo de un proceso, lo cual puede consistir de un área de memoria, un conjunto de registros con valores específicos, la pila y otros valores de contexto.

Un aspecto importante a considerar en estos sistemas es la forma de crear aplicaciones para aprovechar los varios procesadores. Existen aplicaciones que fueron hechas para correr en sistemas monoproceso que no toman ninguna ventaja a menos que el sistema operativo o el compilador detecte secciones de código paralelizable, los cuales son ejecutados al mismo tiempo en procesadores diferentes. Por otro lado, el programador puede modificar sus algoritmos y aprovechar por sí mismo esta facilidad, pero esta última opción las más de las veces es costosa en horas hombre y muy tediosa, obligando al programador a ocupar tanto o más tiempo a la paralelización que a elaborar el algoritmo inicial.

#### 1.3.3.4 Sistemas Operativos por su Estructura (Visión Interna)

Según, se deben observar dos tipos de requisitos cuando se construye un sistema operativo, los cuales son:

Requisitos de usuario: Sistema fácil de usar y de aprender, seguro, rápido y adecuado al uso al que se le quiere destinar.

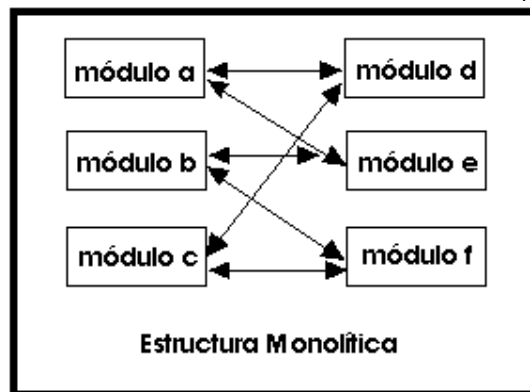
Requisitos del software: Donde se engloban aspectos como el mantenimiento, forma de operación, restricciones de uso, eficiencia, tolerancia frente a los errores y flexibilidad. A continuación se describen las distintas estructuras que presentan los actuales sistemas operativos para satisfacer las necesidades que de ellos se quieren obtener.

##### 1.3.3.4.1 Estructura Monolítica

Es la estructura de los primeros sistemas operativos constituidos fundamentalmente por un solo programa compuesto de un conjunto de rutinas entrelazadas de tal forma que cada una puede llamar a cualquier otra. Las características fundamentales de este tipo de estructura son:

- Construcción del programa final a base de módulos compilados separadamente que se unen a través del ligador.
- Buena definición de parámetros de enlace entre las distintas rutinas existentes, que puede provocar mucho acoplamiento.
- Carecen de protecciones y privilegios al entrar a rutinas que manejan diferentes aspectos de los recursos de la computadora, como memoria, disco, etc.

Figura 17: Estructura Monolítica



Generalmente están hechos a medida, por lo que son eficientes y rápidos en su ejecución y gestión, pero por lo mismo carecen de flexibilidad para soportar diferentes ambientes de trabajo o tipos de aplicaciones.

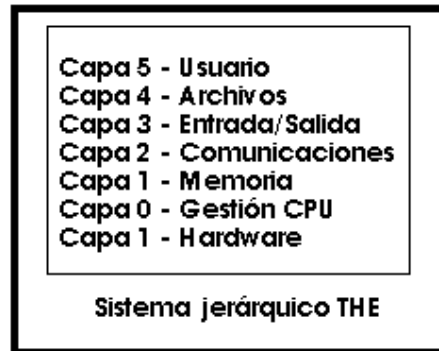
#### 1.3.3.4.2 Estructura Jerárquica

A medida que fueron creciendo las necesidades de los usuarios y se perfeccionaron los sistemas, se hizo necesaria una mayor organización del software y del sistema operativo, donde una parte del sistema contenía subpartes, organizado en forma de niveles.

Se dividió el sistema operativo en pequeñas partes, de tal forma que cada una de ellas estuviera perfectamente definida y con un claro interface con el resto de elementos.

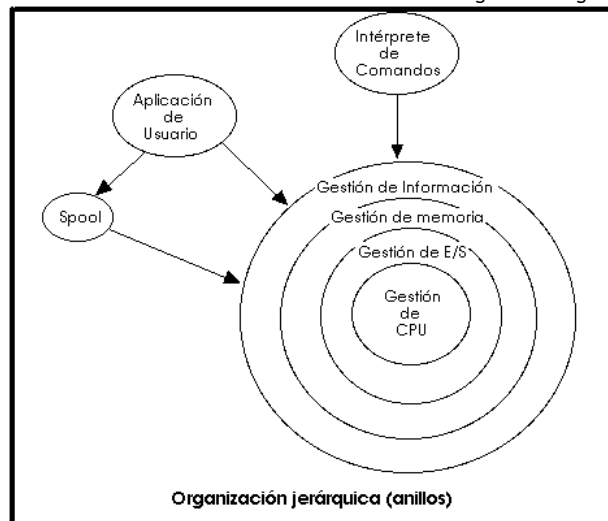
Se constituyó una estructura jerárquica o de niveles en los sistemas operativos, el primero de los cuales fue denominado THE (Technische Hogeschool, Eindhoven), de Dijkstra, que se utilizó con fines didácticos. Se puede pensar también en estos sistemas como si fueran 'multicapa'. Multics y Unix caen en esa categoría.

Figura 18: Sistema Jerárquico THE



En la estructura anterior se basan prácticamente la mayoría de los sistemas operativos actuales. Otra forma de ver este tipo de sistema es la denominada de anillos concéntricos o "rings".

Figura 19: Organización jerárquica (anillos)



En el sistema de anillos, cada uno tiene una apertura, conocida como puerta o trampa (trap), por donde pueden entrar las llamadas de las capas inferiores. De esta forma, las zonas más internas del sistema operativo o núcleo del sistema estarán más protegidas de accesos

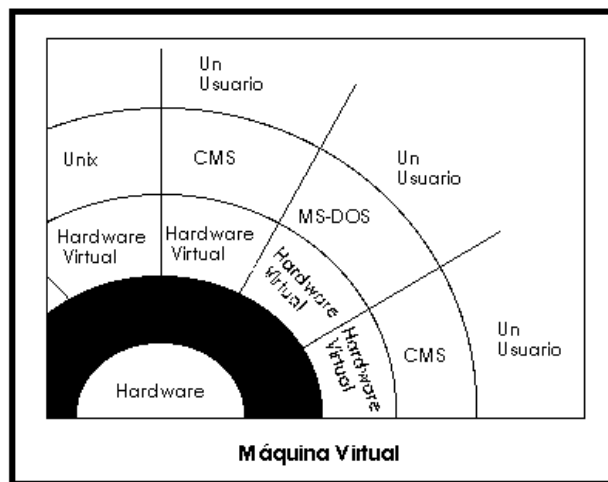
indeseados desde las capas más externas. Las capas más internas serán, por tanto, más privilegiadas que las externas.

### 1.3.3.5 Máquina Virtual

Se trata de un tipo de sistemas operativos que presentan una interfase a cada proceso, mostrando una máquina que parece idéntica a la máquina real subyacente. Estos sistemas operativos separan dos conceptos que suelen estar unidos en el resto de sistemas: la multiprogramación y la máquina extendida. El objetivo de los sistemas operativos de máquina virtual es el de integrar distintos sistemas operativos dando la sensación de ser varias máquinas diferentes.

El núcleo de estos sistemas operativos se denomina monitor virtual y tiene como misión llevar a cabo la multiprogramación, presentando a los niveles superiores tantas máquinas virtuales como se soliciten. Estas máquinas virtuales no son máquinas extendidas, sino una réplica de la máquina real, de manera que en cada una de ellas se pueda ejecutar un sistema operativo diferente, que será el que ofrezca la máquina extendida al usuario

Figura 20: Máquina Virtual



### 1.3.3.6 Cliente-Servidor (Microkernel)

El tipo más reciente de sistemas operativos es el denominado Cliente-servidor, que puede ser ejecutado en la mayoría de las computadoras, ya sean grandes o pequeñas.

Este sistema sirve para toda clase de aplicaciones por tanto, es de propósito general y cumple con las mismas actividades que los sistemas operativos convencionales.

El núcleo tiene como misión establecer la comunicación entre los clientes y los servidores. Los procesos pueden ser tanto servidores como clientes. Por ejemplo, un programa de aplicación normal es un cliente que llama al servidor correspondiente para acceder a un archivo o realizar una operación de entrada/salida sobre un dispositivo concreto. A su vez, un proceso cliente puede actuar como servidor para otro.

Este paradigma ofrece gran flexibilidad en cuanto a los servicios posibles en el sistema final, ya que el núcleo provee solamente funciones básicas de memoria, entrada/salida, archivos y procesos, dejando a los servidores proveer la mayoría que el usuario final o programador puede usar. Estos servidores deben tener mecanismos de seguridad y protección que, a su vez, serán filtrados por el núcleo que controla el hardware. Actualmente se está trabajando en una versión de UNIX que contempla en su diseño este paradigma.

### 1.3.3.7 Sistemas Operativos por la Forma de Ofrecer sus Servicios

Esta clasificación también se refiere a una visión externa, que en este caso se refiere a la del usuario, el cómo accede a los servicios. Bajo esta clasificación se pueden detectar dos tipos principales: sistemas operativos de red y sistemas operativos distribuidos.

### 1.3.3.8 Sistema Operativo de Red

Los sistemas operativos de red se definen como aquellos que tiene la capacidad de interactuar con sistemas operativos en otras computadoras por medio de un medio de transmisión con el objeto de intercambiar información, transferir archivos, ejecutar comandos remotos y un sinfín de otras actividades. El punto crucial de estos sistemas es que el usuario debe conocer la sintaxis de un conjunto de comandos o llamadas al sistema para ejecutar estas operaciones, además de la ubicación de los recursos a los que desee acceder.

El primer Sistema Operativo de red estaba enfocado a equipos con un procesador Motorola 68000, pasando posteriormente a procesadores Intel como Novell Netware.

Los Sistemas Operativos de red más ampliamente usados son: Novell Netware, Personal Netware, LAN Manager, Windows NT Server, UNIX, LANtastic.

### 1.3.3.9 Sistemas Operativos Distribuidos

Los sistemas operativos distribuidos abarcan los servicios de red, logrando integrar recursos (impresoras, unidades de respaldo, memoria, procesos, unidades centrales de proceso) en una sola máquina virtual que el usuario accede en forma transparente. Es decir, ahora el usuario ya no necesita saber la ubicación de los recursos, sino que los conoce por nombre y simplemente los usa como si todos ellos fuesen locales a su lugar de trabajo habitual. Todo lo anterior es el marco teórico de lo que se desearía tener como sistema operativo distribuido, pero en la realidad no se ha conseguido crear uno del todo, por la complejidad que suponen: distribuir los procesos en las varias unidades de procesamiento, reintegrar sub-resultados, resolver problemas de concurrencia y paralelismo, recuperarse de fallas de algunos recursos distribuidos y consolidar la protección y seguridad entre los diferentes componentes del sistema y los usuarios. Los avances tecnológicos en las redes de área local y la creación de microprocesadores de 32 y 64 bits lograron que computadoras más o menos baratas tuvieran el suficiente poder en forma autónoma para desafiar en cierto grado a los mainframes, y a la vez se dio la posibilidad de intercomunicarlas, sugiriendo la oportunidad de partir procesos muy pesados en cálculo en unidades más pequeñas y distribuirlos en los varios microprocesadores para luego reunir los sub-resultados, creando así una máquina virtual en la red que exceda en poder a un mainframe.

Al sistema integrador de los microprocesadores que hacen ver a las varias memorias, procesadores, y todos los demás recursos como una sola entidad en forma transparente se le llama sistema operativo distribuido. Las razones para crear o adoptar sistemas distribuidos se dan por dos razones principales: por necesidad (debido a que los problemas a resolver son inherentemente distribuidos) o porque se desea tener más confiabilidad y disponibilidad de recursos. En el primer caso tenemos, por ejemplo, el control de los cajeros automáticos en diferentes estados de la república. Ahí no es posible ni eficiente mantener un control centralizado, es más, no existe capacidad de cómputo y de entrada/salida para dar servicio a los millones de operaciones por minuto. En el segundo caso, supóngase que se tienen en una gran empresa varios grupos de trabajo, cada uno necesita almacenar grandes cantidades de información en disco duro con una alta confiabilidad y disponibilidad. La solución puede ser que para cada grupo de trabajo se asigne una partición de disco duro en servidores diferentes, de manera que si uno de los servidores falla, no se deje dar el servicio a todos, sino sólo a unos cuantos y, más aún, se podría tener un sistema con discos en espejo (mirror) a través de la red, de manera que si un servidor se cae, el servidor en espejo continúa trabajando y el usuario no se percata de estas fallas, es decir, obtiene acceso a recursos en forma transparente.

Los sistemas distribuidos deben de ser muy confiables, ya que si en un componente del sistema se dispone de otro componente, debe de ser capaz de reemplazarlo.

Entre los diferentes Sistemas Operativos distribuidos que existen tenemos los siguientes: Sprite, Solaris-MC, Mach, Chorus, Spring, Amoeba, Taos, etc.

### 1.3.3.10 Unix

Ken Thompson y Dennis Ritchie **Unix**® (o **UNIX**) es un sistema operativo portable, multitarea y multiusuario; desarrollado en principio por un grupo de empleados de los laboratorios Bell de AT&T, entre los que figuran Ken Thompson, Dennis Ritchie y Douglas McIlroy.

Hoy día, la palabra **UNIX** se utiliza para denotar diferentes conceptos dependiendo del contexto en que es usada. Esto suele dar lugar a confusiones:

- **UNIX - Familia:** Desde el punto de vista técnico, UNIX se refiere a una familia de sistemas operativos que comparten unos criterios de diseño e interoperabilidad en común. Esta familia incluye más de 100 sistemas operativos desarrollados a lo largo de 20 años. No obstante, es importante señalar que esta definición no implica necesariamente que dichos sistemas operativos compartan código o cualquier propiedad intelectual.
- **UNIX - El sistema operativo original:** Desde el punto de vista histórico, UNIX se refiere a la subfamilia de sistemas operativos que descienden de la primera implementación original de AT&T. El término "descendencia" ha de interpretarse como trabajos derivativos que comparten propiedad intelectual con la implementación original.
- **UNIX - La marca:** Desde el punto de vista legal, Unix es una marca de mercado. Dicha marca es propiedad de "The Open Group", una organización de estandarización que permite el uso de dicha marca a cualquier sistema operativo que cumpla con sus estándares publicados. Todo ello independientemente de que el sistema operativo en cuestión sea descendiente o clónico del Unix original. Resumiendo, la marca Unix no es propiedad de ninguna compañía.

### Historia

El inicio del desarrollo de este sistema operativo (uno de los más influyentes en la historia de la computación) fue muy particular, pues nadie habría predicho el éxito de UNIX después de su primera encarnación.

A finales del año 1960, el Instituto Tecnológico de Massachussets, los laboratorios Bell de AT&T y General Electric trabajaban en un sistema operativo experimental llamado Multics (**M**ultiplexed **I**nformation and **C**omputing **S**ystem), desarrollado para ejecutarse en una computadora central (mainframe) modelo GE-645. El objetivo del proyecto era desarrollar un sistema operativo interactivo que contase con muchas innovaciones, entre ellas mejoras en las políticas de seguridad.

El proyecto, por cierto, consiguió dar a luz versiones para producción, pero las primeras versiones contaban con un pobre rendimiento. Los laboratorios Bell de AT&T decidieron desvincularse y dedicar sus recursos a otros proyectos. Uno de los programadores del equipo de los laboratorios Bell, Ken Thompson, siguió trabajando para la computadora GE-645 y escribió un juego llamado *Space Travel* (Viaje espacial). Sin embargo, descubrió que el juego era lento en la máquina de General Electric y resultaba realmente caro, algo así como 75 dólares de EE.UU. por cada partida.

De este modo, Thompson escribió nuevamente el programa, con ayuda de Dennis Ritchie, en lenguaje ensamblador, para que se ejecutase en una computadora DEC PDP-7. Esta experiencia, junto al trabajo que desarrolló para el proyecto Multics, condujo a Thompson a iniciar la creación de un nuevo sistema operativo para la DEC PDP-7. Thompson y Ritchie lideraron un grupo de programadores, entre ellos a Rudd Canaday, en los laboratorios Bell, para desarrollar tanto el sistema de archivos como el sistema operativo multitarea en sí.

A lo anterior, agregaron un intérprete de órdenes (o intérprete de comandos) y un pequeño conjunto de programas. El proyecto fue bautizado UNICS, como acrónimo **U**niplexed **I**nformation and **C**omputing **S**ystem, pues sólo prestaba servicios a dos usuarios (de acuerdo a Andrew Tanenbaum, era sólo a un usuario). La autoría de esta sigla se le atribuye a Brian Kernighan, ya que era un hack de Multics. Dada la popularidad que tuvo un juego de palabras que consideraba a UNICS un sistema MULTICS castrado (pues *eunuchs*, en inglés, es un

homófono de UNICS), se cambió el nombre a UNIX, dando origen al legado que llega hasta nuestros días.

Hasta ese instante, no había existido apoyo económico por parte de los laboratorios Bell, pero eso cambió cuando el Grupo de Investigación en Ciencias de la Computación decidió utilizar UNIX en una máquina superior a la PDP-7. Thompson y Ritchie lograron cumplir con la solicitud de agregar herramientas que permitieran el procesamiento de textos a UNIX en una máquina PDP-11/20, y como consecuencia de ello consiguieron el apoyo económico de los laboratorios Bell. Fue así como por vez primera, en 1970, se habla oficialmente del sistema operativo UNIX ejecutado en una PDP-11/20. Se incluía en él un programa para dar formato a textos (runoff) y un editor de texto.

Tanto el sistema operativo como los programas fueron escritos en el lenguaje ensamblador de la PDP-11/20. Este "sistema de procesamiento de texto" inicial, compuesto tanto por el sistema operativo como de runoff y el editor de texto, fue utilizado en los laboratorios Bell para procesar las solicitudes de patentes que ellos recibían. Pronto, runoff evolucionó hasta convertirse en troff, el primer programa de edición electrónica que permitía realizar composición tipográfica. El 3 de noviembre de 1971 se publicó *The UNIX Programmer's Manual*.

En 1973 se tomó la decisión de escribir nuevamente UNIX, pero esta vez en el lenguaje de programación C. Este cambio significaba que UNIX podría ser fácilmente modificado para funcionar en otras computadoras (de esta manera, se volvía portátil) y así otras variaciones podían ser desarrolladas por otros programadores. Ahora, el código era más conciso y compacto, lo que se tradujo en un aumento en la velocidad de desarrollo de UNIX. AT&T puso a UNIX a disposición de universidades y compañías, así como al gobierno de los Estados Unidos, a través de licencias. Una de estas licencias fue otorgada al Departamento de Computación de la Universidad de California, con sede en Berkeley. En 1975 esta institución desarrolló y publicó su propio sucedáneo de UNIX, conocida como *Berkeley Software Distribution* (BSD), que se convirtió en una fuerte competencia para la familia UNIX de AT&T.

Mientras tanto, AT&T creó una división comercial denominada *Unix Systems Laboratories* para la explotación comercial del sistema operativo. El desarrollo prosiguió, con la entrega de las versiones 4, 5 y 6 en el transcurso de 1975. Estas versiones incluían los *pipes* o *tuberías*, lo que permitió dar al desarrollo una orientación modular respecto a la base del código, consiguiendo aumentar aún más la velocidad de desarrollo. Ya en 1978, cerca de 600 o más máquinas estaban ejecutándose con alguna de las distintas encarnaciones de UNIX.

La versión 7, la última versión del UNIX original con amplia distribución, entró en circulación en 1979. Las versiones 8, 9 y 10 se desarrollaron durante la década de los 80, pero su circulación se limitó a unas cuantas universidades, a pesar de que se publicaron los informes que describían el nuevo trabajo. Los resultados de esta investigación sirvieron de base para la creación de Plan 9, un nuevo sistema operativo portátil distribuido y diseñado para ser el sucesor de UNIX en investigación por los Laboratorios Bell.

AT&T entonces inició el desarrollo de UNIX System III, basado en la versión 7, como una variante de tinte comercial y así vendía el producto de manera directa. La primera versión se lanzó en 1982. A pesar de lo anterior, la empresa subsidiaria Western Electric seguía vendiendo versiones antiguas de Unix basadas en las distintas versiones hasta la séptima. Para finalizar con la confusión con todas las versiones divergentes, AT&T decidió combinar varias versiones desarrolladas en distintas universidades y empresas, dando origen al Unix System V Release 1. Esta versión presentó características tales como el editor Vi y la biblioteca *curses*, desarrolladas por Berkeley Software Distribution en la Universidad de California, Berkeley.

También contaba con compatibilidad con las máquinas VAX de la compañía DEC. En 1993, la compañía Novell adquirió la división *Unix Systems Laboratories* junto con su propiedad intelectual. Esto ocurrió en un momento delicado en el que *Unix Systems Laboratories* disputaba una demanda en los tribunales contra BSD por infracción de los derechos de copyright, revelación de secretos y violación de marca de mercado.



BSD no solamente ganó el juicio sino que descubrieron que grandes porciones del código de BSD habían sido copiadas ilegalmente en UNIX System V. En realidad, la propiedad intelectual de Novell (recién adquirida de Unix Systems Laboratories) se reducía a unos pocos archivos fuente. La correspondiente contra-demanda acabó en un acuerdo extrajudicial cuyos términos permanecen bajo secreto a petición de Novell.

Aproximadamente por esa misma fecha, un estudiante de ciencias de la computación llamado Linus Torvalds desarrolló un núcleo o kernel para computadoras con arquitectura de procesador Intel x86 que mimetizaba muchas de las funcionalidades de UNIX y lo denominó Linux y lo lanzó en forma de código abierto en 1991, el cuál fue unido con los programas desarrollados por el proyecto GNU y crear el SO GNU/Linux. Cientos de miles de desarrolladores incorporaron otras funcionalidades en el núcleo de este sistema operativo derivando, después de unos años, en un sistema multiplataforma capaz de competir con las demás familias de UNIX.

En 1995, Novell vendió su división UNIX comercial (es decir, la antigua Unix Systems Laboratories) a *Santa Cruz Operation* (SCO) reservándose, aparentemente, algunos derechos de propiedad intelectual sobre el software. SCO continúa la comercialización de System V en su producto UnixWare, que durante cierto tiempo se denominó OpenUnix, aunque ha retomado de nuevo el nombre de UnixWare.

## Familias

Como se puede deducir de esta breve reseña histórica, existen varias familias del sistema operativo UNIX que han evolucionado de manera independiente a lo largo de los años. Cada familia se distingue no tanto por sus diferencias técnicas como por sus diferencias en propiedad intelectual. Se observa que todas las familias se han visto contaminadas, directa o indirectamente, por otras familias. Las familias UNIX más significativas son:

- AT&T: La familia que tuvo su origen en el UNIX de AT&T. Considerada la familia UNIX "pura" y original. Sus sistemas operativos más significativos son UNIX System III y UNIX System V.
- BSD: Familia originada por el licenciamiento de UNIX a Berkely. BSD incorpora propiedad intelectual no originaria de AT&T, la primera implementación de los protocolos TCP/IP que dieron origen a Internet.
- AIX: Esta familia surge por el licenciamiento de UNIX System III a IBM.
- Xenix: Familia derivada de la adquisición de los derechos originales de AT&T por parte de SCO.
- GNU/Linux: En 1983, Richard Stallman anunció el Proyecto GNU, un ambicioso esfuerzo para crear un sistema similar a Unix, que pudiese ser distribuido libremente. El software desarrollado por este proyecto -por ejemplo, GNU Emacs y GCC - también han sido parte fundamental de otros sistemas UNIX. En 1991, cuando Linus Torvalds empezó a proponer el *kernel* Linux y a reunir colaboradores, las herramientas GNU eran la elección perfecta. Al combinarse ambos elementos, conformaron la base del sistema operativo (basado en POSIX) que hoy conocemos como GNU/Linux o simplemente Linux. Las distribuciones basadas en el kernel, el software GNU y otros agregados entre las que podemos mencionar a Red Hat Linux y Debian GNU/Linux se han hecho populares tanto entre los aficionados a la computación como en el mundo empresarial. Obsérvese que Linux tiene un origen independiente, por lo que se considera un 'clónico' de UNIX y no un UNIX en el sentido histórico.

Las interrelaciones entre estas familias son las siguientes, aproximadamente en orden cronológico:

- La familia BSD surge del licenciamiento del UNIX original de AT&T.
- Xenix también surge por licenciamiento del UNIX original de AT&T, aunque aún no era propiedad de SCO.
- AIX surge por licenciamiento de UNIX System III, pero también incorpora propiedad intelectual de BSD.

- La familia original AT&T incorpora ilegalmente propiedad intelectual de BSD en UNIX System III r3.
- La familia AIX vuelve a incorporar propiedad intelectual de la familia AT&T, esta vez procedente de UNIX System V.
- Linux incorpora propiedad intelectual de BSD, gracias a que éste también se libera con una licencia de código abierto denominada *Open-source BSD*.
- Según SCO el sistema Linux incorpora propiedad intelectual procedente de AIX, gracias a la colaboración de IBM en la versión 2.4 de su núcleo (kernel), más aún no está demostrado.

### La marca

UNIX es una marca registrada de The Open Group en Estados Unidos y otros países. Esta marca sólo se puede aplicar a los sistemas operativos que cumplen la "Single UNIX Specification" de esta organización y han pagado las regalías establecidas.

En la práctica, el término UNIX se utiliza en su acepción de familia. Se aplica también a sistemas multiusuario basados en POSIX (tales como GNU/Linux, Mac OS X, FreeBSD, NetBSD, OpenBSD), los cuales no buscan la certificación UNIX por resultar cara para productos destinados al consumidor final o que se distribuyen libremente en Internet. En estos casos, el término se suele escribir como "UN\*X", "\*NIX", o "\*N?X".

### Implementaciones más importantes

A lo largo de la historia ha surgido una gran multitud de implementaciones comerciales de UNIX. Sin embargo, un conjunto reducido de productos han consolidado el mercado y prevalecen gracias a un continuo esfuerzo de desarrollo por parte de sus fabricantes. Los más importantes son:

- Solaris de Sun Microsystems. Uno de los sistemas operativos Unix más difundido en el entorno empresarial y conocido por su gran estabilidad. Parte del código fuente de Solaris se ha liberado con licencia de fuentes abiertas.
- AIX de IBM. El UNIX "propietario" de IBM ha cumplido 20 años de vida en el 2006 y continúa en pleno desarrollo, con una perceptible herencia del mainframe en campos como la virtualización o la RAS de los servidores, heredada de sus "hermanos mayores".
- HP-UX de Hewlett-Packard. Este sistema operativo también nació ligado a las computadoras departamentales de este fabricante. También es un sistema operativo estable que continúa en desarrollo.

Adicionalmente, existen distribuciones de Linux muy difundidas en la empresa. Se trata de distribuciones sostenidas por grandes compañías y que, por tanto, pueden ofrecer las mismas garantías de continuidad y mantenimiento existente para los sistemas UNIX más comunes. Las más importantes son:

- Red Hat Linux de Red Hat. La primera compañía que tuvo éxito en la comercialización de Linux, tras el fracaso de Caldera, hoy desaparecida.
- SuSE Linux de Novell. Originalmente liberado por la compañía alemana SuSE. Es popular por sus herramientas de administración centralizada.

Las siguientes implementaciones tienen importancia desde el punto de vista histórico, no obstante, actualmente están en desuso:

- Tru64 UNIX de Compaq (antes Digital Equipment Corporation).
- SCO UnixWare de Santa Cruz Operation.
- UX/4800 de NEC.
- IRIX de Silicon Graphics Inc.

### 1.3.3.11 Linux

Linux es un sistema libre creado por Linus Torvalds en 1991, mientras todavía era estudiante de la universidad de Helsinki, Torvalds empezó Linux escribiendo el núcleo, en parte desde cero y en parte utilizando software que estaba a disposición del público, después Torvalds envió el sistema a sus amigos y a una comunidad hacker de Internet y les pidió que trabajaran con ello y lo mejoraran, en ese momento despegó Linux.

Hoy en día existen cientos de desarrolladores de software por todo el mundo que contribuyen con su software al proyecto Linux. Dado que el código fuente está disponible con toda libertad, cualquiera puede trabajar con él, cambiarlo o mejorarlo. En la cima del proyecto de núcleo de Linux, los creadores de Linux también han recurrido a una gran cantidad de software de sistema y aplicaciones que actualmente están unidas a Linux gracias al proyecto de software GNU, dirigido por la Free Software Foundation (FSF).

Una gran cantidad de veces se escucha que Linux es una versión libre de Unix. A pesar de que Linux comenzó con una gran cantidad de código nuevo, siguió el estándar POSIX, (Interfaz de sistema operativo portable para UNIX), un estándar de sistema operativo de la industria informática al que se le ceñían todas las grandes versiones de UNIX. Dicho de otro modo, si su sistema operativo se ajustaba a POSIX, también era UNIX.

#### Las raíces de Linux en UNIX

Linux creció en una cultura de libre intercambio de ideas y software. Al igual que UNIX, el sistema operativo sobre el que se basa Linux, centraba su interés en mantener abiertos los canales de comunicación con los desarrolladores de software. El objetivo era hacer que funcionara el código, sin prestar demasiada atención en quien era el dueño del código, e Internet era el medio de comunicación primario.

En los años 80 y 90, mientras Microsoft inundaba el mundo con computadoras personales que incorporaban sistemas operativos DOS y Windows, los usuarios avanzados pedían más a un sistema operativo. Suspiraban por sistemas que pudieran funcionar en redes, admitir muchos usuarios a la vez (multiusuario) y ejecutar varios programas al mismo tiempo (multitarea), y para esas fechas Microsoft DOS (Disk Operating System) y Microsoft Windows, no lo hacían.

UNIX por su parte creció en un entorno en el que la tecnología era la estrella y en el que era difícil encontrar personas interesadas en la comercialización. Los laboratorios BELL en Murria Hill, Nueva Jersey, eran un hervidero en el que primero nacían las ideas y después se dejaba a los demás el asunto de los beneficios. Dennis Ritchie, co-creador de Unix y diseñador de lenguaje de programación C, pronunció unas palabras durante una conferencia sobre la evolución de Unix, que resumen el espíritu que dio lugar a Unix. Estaba comentando sus esperanzas y las de sus compañeros sobre el proyecto Unix tras un proyecto similar denominado Multics que acaba de fracasar:

*“Lo que queríamos preservar no era solo un buen ambiente en el que programar, sino un sistema alrededor del cual se pudiera generar compañerismo. Sabíamos gracias a la experiencia, que la esencia de la informática comunitaria tal y como la posibilitan las máquinas de acceso remoto y de tiempo compartido, no es solo teclear programas en una Terminal, en lugar de una tarjeta perforada, sino fomentar una mayor comunicación”.*

Con ese espíritu se distribuyó gratuitamente el primer código fuente de Unix en las universidades. Al igual que ocurre con Linux, la disponibilidad del código fuente de Unix hizo posible que una gran diversidad de desarrolladores de software pudieran realizar sus propias mejoras de Unix y compartirlas con otros.

A principios de la década de los 80, el desarrollo de Unix se mudó desde la organización en Murria Hill a un laboratorio de desarrollo más orientado hacia la comercialización en Summit, Nueva Jersey. Durante este tiempo Unix empezó a cosechar éxito comercial como el sistema informático de elección para aplicaciones como por ejemplo, el equipo de conmutación

telefónica de AT&T, aplicaciones de supercomputadoras, como los modelados de patrones meteorológicos y para controlar los proyectos espaciales de la NASA.

Los principales vendedores de hardware autorizaron el código fuente de UNIX para que funcionara en sus equipos de cómputo y para intentar crear un ambiente de igualdad y comunidad con sus fabricantes de equipos originales.

### Características comunes de Linux

Da igual qué versión de Linux utilice, el fragmento de código común a todas es el núcleo de Linux. El núcleo se puede modificar para incluir el soporte para las utilidades que desee incorporar. Cualquier núcleo de Linux ofrece las siguientes características:

**Multiusuario:** No sólo puede tener muchas cuentas de usuario disponibles en un sistema Linux, sino que también puede tener múltiples usuarios conectados y trabajando en el sistema al mismo tiempo. Los usuarios pueden tener sus propios entornos organizados como ellos deseen: su propio directorio de inicio para almacenar archivos y su propia interfaz de escritorio (con los iconos, menú y aplicaciones que les convengan).

**Multitarea:** En Linux es posible tener muchos programas ejecutados al mismo tiempo, lo que implica que no sólo puede tener muchos programas funcionando al mismo tiempo, sino también que Linux puede ejecutar programas en segundo plano. Muchos de estos procesos del sistema hacen que Linux pueda funcionar como servidor, ya que estos procesos en segundo plano escuchan las peticiones de la red para acceder al sistema, ver una página Web, imprimir un documento o copiar un archivo. Estos procesos en segundo plano se denominan demonios.

**Interfaz Gráfica de Usuario, GUI (Sistema X Window):** El potente marco de trabajo con aplicaciones gráficas en Linux se denomina Sistema X Window (o simplemente X). X maneja las funciones de apertura de aplicaciones GUI basadas en X y de visualización de las mismas en un proceso de servidor X (el proceso que gestiona la pantalla, el ratón y el teclado).

Sobre X utilizará un entorno de escritorio basado en X para generar una metáfora de escritorio y un gestor de ventanas para crear el ver y sentir el específico de su GUI (iconos, marcos de ventana, menú y colores). Hay varios entornos de escritorio y docenas de gestores de escritorio para elegir. (Red Hat proporciona varios gestores de escritorio, pero se centra en los entornos de escritorio Gnome y KDE.)

**Compatibilidad de hardware:** Puede configurar compatibilidad para casi cualquier tipo de hardware que se puede conectar a un equipo. Hay compatibilidad para disquetes, CD-ROM, discos extraíbles (como, por ejemplo, unidades Zip), tarjetas de sonido, dispositivos de cinta, tarjetas de vídeo y casi todo lo que se le ocurra.

**Conectividad a la red:** Para conectar su sistema Linux a una red, Linux ofrece soporte para una serie de tarjetas, módems y dispositivos seriales de Red de Área Local (LAN). Además de los protocolos LAN, como los protocolos Ethernet y Token Ring, se pueden incorporar todos los protocolos de red de nivel superior más populares. El más conocido de estos protocolos es TCP/IP (usado para conectarse a Internet). También están disponibles otros protocolos, por ejemplo, IPX (para las redes Novell) y x.25 (un tipo de red de conmutación de paquetes extendido en Europa).

**Servidores de red:** Lo que Linux hace mejor es ofrecer servicios de red a los equipos clientes en la red o para todo Internet. Se puede tener acceso a una gran variedad de paquetes de software que le permiten utilizar Linux como servidor de impresión, servidor de archivos, servidor FTP, servidor de correo, servidor Web, servidor de noticias o servidor de grupo de trabajo (bootp o NIS).

**Compatibilidad para aplicaciones:** Debido a su compatibilidad con POSIX y varias interfaces de programación de aplicaciones (API), hay una amplia gama de freeware y shareware

disponibles para Linux. La mayor parte del software GNU de la Free Software Foundation funcionará en Linux.

### **Principales-ventajas de Linux**

Al compararlo con otros sistemas operativos disponibles en el mercado, los principales valores de Linux son su precio y su fiabilidad. La mayoría sabe que su precio de salida es gratuito (o al menos inferior a 100 euros cuando viene en una caja o en un libro). Sin embargo, cuando la gente habla de lo asequible que es Linux, normalmente están pensando en su costo total, que incluye la capacidad de utilizar hardware económico. Aunque los sistemas operativos comerciales tienden a fomentar el cambio cualitativo hacia el hardware posterior, Linux no fomenta ese cambio, (aunque resulta atractivo tener hardware más rápido y discos de mayor tamaño).

En términos de fiabilidad, el consenso general es que Linux es comparable a muchos sistemas comerciales UNIX, pero más fiables que Windows NT o Windows 2000. Esta idea se confirma especialmente cuando tienes que confiar en que tu sistema informático no se caiga por ser un servidor Web o de archivos. (No necesita reiniciar el sistema cada vez que cambie algo.)

Otra ventaja de utilizar Linux es que siempre hay ayuda a la que recurrir en Internet. Probablemente hay alguien ahí afuera, en un grupo de noticias de Linux, que está dispuesto a ayudarlo a resolver su problema.

#### **1.3.3.12 Red Hat Linux**

Para los hacker eso de tener directorios con paquetes de software flotando extrañamente por la red era una buena forma de compartir software. Sin embargo, para que Linux fuera aceptable para una población de usuarios de computadora menos familiarizada con cuestiones técnicas, era necesario que fuera fácil de instalar y utilizar. De igual forma, las empresas que estaban considerando la posibilidad de asignar sus aplicaciones críticas a un sistema operativo, querían tener la certeza de que este sistema había sido verificado cuidadosamente.

Con este fin, varias empresas y organizaciones empezaron a reunir y empaquetar el software de Linux para crear formas utilizables denominadas distribuciones. La meta fundamental de una distribución de Linux es conseguir que cientos de paquetes de software inconexos que forman Linux funcionen como conjunto cohesionado. La distribución más popular es Red Hat Linux.

Red Hat Linux es un producto comercial de Red Hat Software, Inc. Puede entrar en una tienda y comprar la caja con los CD y los manuales. También puede conseguir Red Hat Linux bajándolo de la red o instalándolo desde los CD. La caja proporciona asistencia técnica sobre instalación e incluye un CD adicional que contiene versiones limitadas y demo de paquetes de aplicaciones Linux disponibles en el mercado (como procesadores de texto y aplicaciones de bases de datos).

#### **Por qué elegir Red Hat Linux?**

Con el fin de diferenciarse de otras versiones de Linux, cada distribución añade algunas utilidades adicionales. Dado que la mayoría de las utilidades avanzadas ya están incorporadas, la mayoría de las mejoras de una distribución concreta tienen por finalidad facilitar la instalación, la configuración y el uso de Linux. Asimismo, dado que hay diferentes paquetes de software disponibles para hacer los mismos trabajos (como los gestores de ventanas o un tipo específico de servidor), una distribución puede diferenciarse mediante la elección de los paquetes que desea incluir y mostrar.

La elección de Red Hat Linux como su distribución es una buena decisión. Empresas como IBM ofrecen Red Hat Linux en sus servidores de gama alta. Clientes tan importantes como Amazon.com confían en Red Hat Linux como su primer sistema operativo para realizar sus actividades. Se trata del mismo sistema operativo que se puede utilizar para desarrollar un pequeño negocio o gestionar una red doméstica o un servidor Web personal.

Red Hat Linux se ha desmarcado de otras distribuciones de Linux en los siguientes aspectos:

**Empaquetado del software:** Red Hat Software, Inc. creó el método de administración de paquetes de Red Hat (RPM) para empaquetar Linux.

Los RPM permiten que usuarios con menos habilidades técnicas instalen fácilmente el software de Linux. Con las herramientas RPM se puede instalar desde el CD, desde el disco duro, desde su LAN o desde Internet. Resulta sencillo averiguar qué paquetes están instalados o echar un vistazo a los contenidos de un paquete. Como RPM está disponible para la comunidad Linux, éste se ha convertido en un estándar de facto para empaquetar el software de Linux.

**Facilidad de instalación:** El proceso de instalación de Red Hat Linux proporciona pasos sencillos para instalar Linux. Red Hat también le ayuda durante la instalación a dar los primeros pasos de la configuración de Linux. Puede elegir qué paquetes quiere instalar y cómo realizar las particiones del disco duro. Puede, incluso, preparar su escritorio GUI para ser utilizado configurando la tarjeta de vídeo y el monitor.

**Guiones de estilo de nivel de ejecución UNIX System V:** Con el fin de disponer sus servicios de sistema (procesos demonio) de inicio y cierre de forma organizada, Red Hat Linux utiliza el mecanismo UNIX System V para iniciar y finalizar servicios. Los guiones del intérprete de comandos del núcleo (fáciles de leer y cambiar) están contenidos en directorios de / etc. Al cambiar el nivel de ejecución, como por ejemplo cuando el sistema se inicia o cambia al modo usuario único, aparecen mensajes que dicen si los servicios se iniciaron correctamente o si no se ejecutaron debidamente (¡una utilidad muy divertida!). El uso de los guiones de nivel de ejecución se describirá ampliamente en otro capítulo.

**Entornos de escritorio (Gnome y KDE):** Con objeto de facilitar el uso de Linux, Red Hat Linux está preparado para utilizar los entornos de escritorio Gnome y KDE. Gnome se instala por defecto y ofrece utilidades muy interesantes, entre las que podemos destacar los protocolos para arrastrar y soltar y ventanas que actúan como sombras de ventana. KDE es otro conocido gestor de escritorio que incorpora varias herramientas, como el Centro de control KDE para configurar el escritorio y el K Organizer para gestionar las citas.

**Herramientas de administración GUI:** Hay algunas herramientas de configuración que resultan de gran ayuda a la hora de configurar las tareas más complejas de Linux. Las herramientas GUI, como por ejemplo linuxconf y netcfg ofrecen una interfaz gráfica, orientada a formularios para configurar redes, usuarios, sistemas de archivos y servicios de inicialización.

En lugar de crear intrincadas líneas de comando o verse en la necesidad de crear complejos archivos de configuración, linuxconf y netcfg pueden configurar esos archivos automáticamente.

**Comprobación:** La configuración exacta que aparece en una distribución de Red Hat Linux ha pasado por un exhaustivo proceso de comprobación realizado por expertos de todo el mundo. El simple hecho de que un paquete de software se haya incluido en una distribución de Linux es una indicación de haber alcanzado un determinado nivel de calidad.

**Actualización Automática:** los paquetes de software que componen Red Hat Linux se mejoran constantemente y se modifican de diferentes formas. Para facilitar un mecanismo de selección automática, descarga e instalación de paquetes de software actualizado, Red Hat Linux ha creado Red Hat Network. Utilizando este sitio Web o el comando up2date, se puede recibir automáticamente parches de seguridad vitales y nuevos elementos de manera sencilla a través de Internet.

## Historia

Red Hat Software Inc. fue fundada en 1994 por Bob Young y Marc Ewing. En agosto de 1999, Red Hat salió a la Bolsa y sus acciones obtuvieron la octava ganancia de primer día más grande en toda la historia de Wall Street. Cuatro años más tarde, el valor de las acciones de Red Hat es en torno a una centésima parte del máximo valor que llegara a alcanzar antes de la crisis de las puntocom. Aun así, sus comienzos exitosos en el mercado de valores sirvieron para que Red Hat fuera portada en periódicos y revistas no directamente relacionadas con temas informáticos. En cualquier caso, parece ser que Red Hat ha sabido superar los problemas de otras compañías del mundo de los negocios en torno al software libre y anunció números negros por primera vez en su historia en el último cuarto del año 2002.

Otro de los hechos históricos más importantes de Red Hat fue la adquisición en noviembre de 1999 de Cygnus Solutions, una empresa fundada una década antes y que ya había demostrado cómo con una estrategia integral basada en software libre se puede ganar dinero.

En septiembre de 2003, Red Hat ha decidido concentrar sus esfuerzos de desarrollo en la versión corporativa de su distribución y delegó la versión común a Fedora Core, un proyecto abierto independiente de Red Hat.

### 1.3.3.13 Solaris

Solaris es un sistema operativo desarrollado por Sun Microsystems. Es un sistema certificado como una versión de UNIX. Aunque Solaris en sí mismo aún es software propietario, la parte principal del sistema operativo se ha liberado como un proyecto de software libre denominado *Opensolaris*. Solaris puede considerarse uno de los sistemas operativos más avanzados. Sun denomina así a su sistema operativo.

#### Historia

El primer sistema operativo de Sun nació en 1983 y se llamó inicialmente **SunOS**. Estaba basado en el sistema UNIX BSD, de la Universidad de Berkeley, del cual uno de los fundadores de la compañía fue programador en sus tiempos universitarios. Más adelante incorporó funcionalidades del System V, convirtiéndose prácticamente en un sistema operativo totalmente basado en System V.

Esta versión basada en System V fue publicada en 1992 y fue la primera en llamarse **Solaris**, más concretamente *Solaris 2*. Las anteriores fueron llamadas *Solaris 1* con efecto retroactivo. SunOS sólo tendría sentido a partir de ese momento como núcleo de este nuevo entorno operativo Solaris. De esta forma Solaris 2 contenía SunOS 5.0. Desde ese momento se distingue entre el núcleo del sistema operativo (SunOS), y el entorno operativo en general (Solaris), añadiéndole otros paquetes como Apache o DTrace. Como ejemplo de esta función, Solaris 8 contiene SunOS 5.8.

#### Arquitecturas soportadas

Solaris usa una base de código común para las arquitecturas que soporta: SPARC y x86 (incluyendo AMD64/EM64T). También fue portado a la arquitectura PowerPC (en plataforma PReP) en la versión 2.5.1, pero el porte fue cancelado casi tan pronto como fue liberado. En un tiempo se planeó soporte para el Itanium pero nunca se llevó al mercado. Sun también tiene planes de implementar ABIs de Linux en Solaris 10, permitiendo la ejecución de código objeto Linux de forma nativa en la plataforma x86.

Solaris tiene la reputación de ser muy adecuado para el multiprocesamiento simétrico (SMP), soportando un gran número de CPUs. También ha incluido soporte para aplicaciones de 64 bits SPARC desde Solaris 7. Históricamente Solaris ha estado firmemente integrado con la plataforma hardware de Sun, SPARC, con la cual fue diseñado y promocionado como un paquete combinado.

Esto proporcionaba frecuentemente sistemas más fiables pero con un costo más elevado que el del hardware de PC. De todas formas, también ha soportado sistemas x86 desde la versión Solaris 2.1 y la última versión, Solaris 10, ha sido diseñada con AMD64 en mente, permitiendo a Sun capitalizar en la disponibilidad de CPUs de 64 bits commodities basadas en la arquitectura AMD64. Sun ha promocionado intensamente Solaris con sus estaciones de trabajo

de nivel de entrada basadas en AMD64, así como con servidores que en 2006 varían desde modelos dual-core hasta modelos a 16 cores.

### Entornos de escritorio

El primer entorno de escritorio para Solaris fue OpenWindows. Fue reemplazado por CDE en la versión Solaris 2.5. El escritorio Java Desktop System, basado en GNOME, se incluye por defecto con Solaris 10.

### OpenSolaris

El código fuente de Solaris (con unas pocas excepciones) ha sido liberado bajo la licencia CDDL (Licencia Común de Desarrollo y Distribución) como un proyecto de software libre bajo el nombre OpenSolaris.

La licencia CDDL ha sido aprobada por la Open Source Initiative (OSI) como una licencia open source y por la FSF como una licencia de software libre (aunque incompatible con la popular licencia GPL).

La base de OpenSolaris fue alimentada el 14 de junio de 2005 a partir de la entonces actual base de desarrollo de código de Solaris. Es posible descargar y licenciar versiones tanto binarias como en forma de código fuente sin costo alguno.

### Versiones

En orden descendente, las siguientes versiones de Solaris han sido liberadas a 2006:

Versión de Solaris	Versión de SunOS	Fecha de publicación	Descripción
Solaris 10	SunOS 5.10	31 de enero, 2005	Incluye soporte AMD64/EM64T, DTrace, Solaris Containers, Service Management Facility (SMF) para reemplazar al sistema init.d, NFSv4. Modelo de seguridad basado en el menor privilegio. Se ha eliminado soporte para procesadores sun4m y UltraSPARC I a frecuencia menor que 200 Mhz. Se ha añadido Java Desktop System como escritorio por defecto. Se ha añadido GRUB como cargador de arranque para plataformas x86. Se ha añadido soporte SCSI. Se ha añadido soporte para el nuevo sistema de archivos, ZFS (versión 1/06).
Solaris 9	SunOS 5.9	28 de mayo, 2002 (SPARC) 10 de enero, 2003 (x86)	iPlanet Directory Server, Resource Manager, Solaris Volume Manager. Añadida compatibilidad con Linux. Eliminado OpenWindows. Eliminado soporte para sun4d. La actualización más reciente es Solaris 9 9/05.
Solaris 8	SunOS 5.8	Febrero de 2000	Incluye Multipath I/O, IPv6 y IPsec. Introduce RBAC (control de acceso basado en roles). Soporte para sun4c eliminado. La actualización más reciente es Solaris 8 2/04.
Solaris 7	SunOS 5.7	Noviembre de 1998	La primera versión de 64 bits para plataforma UltraSPARC. Añadido soporte nativo para registro de metadatos en el sistema de archivos (UFS logging).
Solaris 2.6	SunOS 5.6	Julio de 1997	Incluye protocolo Kerberos, PAM, TrueType, WebNFS, y soporte de archivos grandes.
Solaris 2.5.1	SunOS 5.5.1	Mayo de 1996	Ésta fue la primera y única versión que soportó la plataforma PowerPC. También se añadió soporte Ultra Enterprise, y los identificadores de usuario (uid_t) se expandieron a 32 bits.
Solaris 2.5	SunOS 5.5	Noviembre de 1995	Primera versión en soportar UltraSPARC e incluir CDE, NFSv3 y NFS/TCP
Solaris 2.4	SunOS 5.4	Noviembre de 1994	Primera versión unificada SPARC/x86. Incluye soporte de ejecución OSF/Motif.
Solaris 2.3	SunOS 5.3	Noviembre de 1993	OpenWindows 3.3 cambia de NeWS a Display PostScript y elimina soporte de SunView. Esta versión fue sólo para SPARC.
Solaris 2.2	SunOS 5.2	Mayo de 1993	Primera en soportar la arquitectura sun4d. Esta versión fue sólo para



			SPARC.
Solaris 2.1	SunOS 5.1	Diciembre de 1992 (SPARC), Mayo de 1993 (x86)	Soporte para arquitectura Sun-4 y sun4m. Primera versión para Solaris x86.
Solaris 2.0	SunOS 5.0	Junio de 1992	Primera versión preliminar, soporte solamente para la arquitectura sun4c.

Solaris 7 ya no se distribuye pero aún está soportado. Las versiones anteriores no están soportadas. Un resumen más comprensivo de algunas versiones de Solaris también está disponible. Las versiones de Solaris también se describen en el FAQ de Solaris 2.

### Versión en desarrollo

La base de código subyacente de Solaris ha estado bajo desarrollo continuo desde que el trabajo empezó a finales de los años 80 en lo que fue con el tiempo liberado como Solaris 2.0. Cada versión como Solaris 10 se basa en una instantánea (*snapshot*) de este tren de desarrollo, tomada cerca del momento de su liberación, que es después mantenida como un proyecto derivado. Las actualizaciones a ese proyecto son construidas y entregadas varias veces al año hasta que sale la siguiente versión oficial. La versión de Solaris bajo desarrollo por Sun al día de hoy se llama Nevada y se deriva de lo que es ahora la base de código OpenSolaris.

En 2003 se inició una adición al proceso de desarrollo de Solaris. Bajo el nombre de programa Solaris Express, una instantánea del tren de desarrollo está ahora disponible para su descarga una vez al mes, permitiendo a cualquiera probar las nuevas características y probar la calidad y estabilidad del sistema a medida que progresa hacia la liberación de la siguiente versión oficial.

Dado que Solaris Express predata la liberación de Solaris como proyecto open source, empezó como un programa solamente en forma de binarios, pero ahora hay una versión llamada Solaris Express: Community Release dirigida especialmente hacia desarrolladores OpenSolaris.

Modelo de negocio: Si bien Solaris en una computadora personal apenas necesita mantenimiento profesional; utilizada en una empresa es posible que el empresario quiera contratar los servicios del equipo de Sun para hacer rendir al máximo su negocio, exprimiendo todas las novedades en seguridad de redes y muchas más cosas. Sun fabrica hardware libre, como lo es la tecnología SPARC.

#### 1.3.3.14 HP-UX

HP-UX es la versión de Unix de Hewlett-Packard. HP-UX es la versión de Unix desarrollada y mantenida por Hewlett-Packard desde 1983, ejecutable típicamente sobre procesadores HP PA RISC y en sus últimas versiones sobre Intel Itanium (arquitectura Intel de 64 bits); a pesar de estar basada ampliamente en System V, incorpora importantes características BSD. En la actualidad la última versión del operativo es la 11.23, también conocida como 11iv2 (2003), aunque existen numerosas instalaciones de sistemas más antiguos, especialmente HP-UX 10.x (1995-97) o incluso 9.x. (1992-95).

A partir de la versión 11.11 (2000) se usa un sistema de numeración doble, así la 11.11 es también conocida como 11i, la 11.20 es 11iv1.5 y así sucesivamente hasta la actual 11.23 (11iv2).

HP-UX es, como la mayor parte de Unix comerciales, un entorno de trabajo flexible, potente y estable, que soporta un abanico de aplicaciones que van desde simples editores de texto a complicados programas de diseño gráfico o cálculo científico, pasando por sistemas de control industrial que incluyen planificaciones de tiempo real.

Durante los últimos años Hewlett-Packard, como muchos otros fabricantes, parece haberse interesado bastante por la seguridad en general, y en concreto por los sistemas de protección para sus plataformas; prueba de ello es la gama de productos relacionados con este campo desarrollados para HP-UX, como el sistema de detección de intrusos IDS/9000 para HP-UX 11.x

corriendo sobre máquinas HP-9000 o la utilidad Security Patch Check, similar al PatchDiag de Sun Microsystems. También es importante destacar las grandes mejoras en cuanto a seguridad del sistema se refiere entre HP-UX 9.x, HP-UX 10.x y muy especialmente HP-UX 11.x.

### **1.3.3.15 Windows**

Microsoft Windows es un sistema operativo gráfico para computadoras personales cuyo propietario es la empresa Microsoft. La primera versión popular, Windows 3.1 es una interfaz gráfica que funciona en MS-DOS. MS Windows, como también se le conoce, es el sistema operativo más utilizado en el mundo.

#### **Historia**

##### **Windows 1.0**

En 1985 Microsoft publicó la primera versión de Windows, una interfaz gráfica de usuario (GUI) para su propio sistema operativo (MS-DOS) que había sido incluido en el IBM PC y ordenadores compatibles desde 1981. Es muy similar a la de Apple, usando la interfaz gráfica, fue creada imitando el MacOS de Apple.

La primera versión de Microsoft Windows Premium nunca fue demasiado potente ni tampoco se hizo popular. Estaba severamente limitada debido a los recursos legales de Apple, que no permitía imitaciones de sus interfaces de usuario. Por ejemplo, las ventanas sólo podían disponerse en mosaico sobre la pantalla; esto es, nunca podían solaparse u ocultarse unas a otras. Tampoco había "papelera de reciclaje" debido a que Apple creía que ellos tenían la patente de este paradigma o concepto. Ambas limitaciones fueron eliminadas cuando el recurso de Apple fue rechazado en los tribunales. Por otro lado, los programas incluidos en la primera versión eran aplicaciones "de juguete" con poco atractivo para los usuarios profesionales.

##### **Windows 2.0**

Apareció en 1987, y fue un poco más popular que la versión inicial. Gran parte de esta popularidad la obtuvo de la inclusión en forma de versión "run-time" de nuevas aplicaciones gráficas de Microsoft, Microsoft Excel y Microsoft Word para Windows. Éstas podían cargarse desde MS-DOS, ejecutando Windows a la vez que el programa, y cerrando Windows al salir de ellas. Windows 2 todavía usaba el modelo de memoria 8088 y por ello estaba limitado a 1 megabyte de memoria; sin embargo, mucha gente consiguió hacerlo funcionar bajo sistemas multitareas como DesqView.

##### **Windows 3.0**

La primera versión realmente popular de Windows fue la versión 3.0, publicada en 1990. Ésta se benefició de las mejoradas capacidades gráficas para PC de esta época, y también del microprocesador 80386, que permitía mejoras en las capacidades multitarea de las aplicaciones Windows. Esto permitiría ejecutar en modo multitarea viejas aplicaciones basadas en MS-DOS. Windows 3 convirtió al IBM PC en un serio competidor para el Apple Macintosh.

##### **OS/2 1**

Durante la segunda mitad de los 80, Microsoft e IBM habían estado desarrollando conjuntamente OS/2 como sucesor del DOS, para sacar el máximo provecho a las capacidades del procesador Intel 80286. OS/2 utilizaba el direccionamiento hardware de memoria disponible en el Intel 80286 para poder utilizar hasta 16 MB de memoria. La mayoría de los programas de DOS estaban por el contrario limitados a 640 KB de memoria. OS/2 1.x también soportaba memoria virtual y multitarea.

Más adelante IBM añadió, en la versión 1.1 de OS/2, un sistema gráfico llamado Presentation Manager (PM). Aunque en muchos aspectos era superior a Windows, su API (Programa de Interfaz de Aplicaciones) era incompatible con la que usaban los programas de este último.

(Entre otras cosas, Presentation Manager localizaba el eje de coordenadas X,Y en la parte inferior izquierda de la pantalla como las coordenadas cartesianas, mientras que Windows situaba el punto 0,0 en la esquina superior izquierda de la pantalla como otros sistemas informáticos basados en ventanas).

A principio de los 90, crecieron las tensiones en la relación entre IBM y Microsoft. Cooperaban entre sí en el desarrollo de sus sistemas operativos para PC y cada uno tenía acceso al código del otro. Microsoft quería desarrollar Windows aún más, mientras IBM deseaba que el futuro trabajo estuviera basado en OS/2. En un intento de resolver estas diferencias, IBM y Microsoft acordaron que IBM desarrollaría OS/2 2.0 para reemplazar a OS/2 1.3 y Windows 3.0, mientras Microsoft desarrollaría un nuevo sistema operativo, OS/2 3.0, para suceder más adelante al OS/2 2.0.

Este acuerdo pronto fue dejado de lado y la relación entre IBM y Microsoft terminó. IBM continuó desarrollando IBM OS/2 2.0 mientras que Microsoft cambió el nombre de su (todavía no publicado) OS/2 3.0 a Windows NT.

(Microsoft promocionó Windows NT con tanto éxito que la mayoría de la gente no se dio cuenta de que se trataba de un OS/2 remozado). Ambos retuvieron los derechos para usar la tecnología de OS/2 y Windows desarrollada hasta la fecha de terminación del acuerdo.

### **OS/2 2.0**

IBM publicó OS/2 versión 2.0 en 1992. Esta versión suponía un gran avance frente a OS/2 1.3. Incorporaba un nuevo sistema de ventanas orientado a objetos llamado Workplace Shell como sustituto del Presentation Manager, un nuevo sistema de archivos, HPFS, para reemplazar al sistema de archivos FAT de DOS usado también en Windows y aprovechaba todas las ventajas de las capacidades de 32 bit del procesador Intel 80386. También podía ejecutar programas DOS y Windows, ya que IBM había retenido los derechos para usar el código de DOS y Windows como resultado de la ruptura.

### **OS/2 3.0 y 4.0**

IBM continuó vendiendo OS/2, produciendo versiones posteriores como OS/2 3.0 (también llamado Warp) y 4.0 (Merlin). Pero con la llegada de Windows 95, OS/2 comenzó a perder cuota de mercado. Aunque OS/2 seguía corriendo aplicaciones de Windows 3.0 carecía de soporte para las nuevas aplicaciones que requerían Windows 95. Al contrario que con Windows 3.0, IBM no tenía acceso al código fuente de Windows 95; y tampoco tenía el tiempo ni los recursos necesarios para emular el trabajo de los programadores de Microsoft con Windows 95, no obstante, OS/2 3.0 (Warp) apareció en el mercado antes que Windows 95 (que se retrasaba respecto a la fecha inicial de lanzamiento). Como mejoras incorporaba una reducción en los requisitos de hardware (de pedir 8 Mb de RAM de su antecesor OS/2 2.1, ahora pedía sólo 4 Mb) y como gran añadido, incorporaba el llamado BonusPack, un conjunto de aplicaciones de oficina, comunicaciones, etc. que ahorraban el tener que comprar software adicional como en el caso de Windows. Todo esto unido a una gran campaña publicitaria y a un muy reducido precio (el equivalente a unos 59.40 € frente a los 100 € de Windows) provocaron que mucha gente se animase a probarlo en lugar de esperar la llegada de Windows 95. Lamentablemente, el posterior abandono por parte de IBM hizo que fuese quedando relegado (aunque sigue siendo utilizado -cada vez menos- en sectores bancarios por su alta estabilidad).

### **Windows 3.1 y Windows 3.11**

En respuesta a la aparición de OS/2 2.0, Microsoft desarrolló Windows 3.1, que incluía diversas pequeñas mejoras a Windows 3.0 (como las fuentes escalables TrueType), pero que consistía principalmente en soporte multimedia. Más tarde Microsoft publicó el Windows 3.11 (denominado Windows para trabajo en grupo), que incluía controladores y protocolos mejorados para las comunicaciones en red y soporte para redes punto a punto.

## Windows NT

Mientras tanto Microsoft continuó desarrollando Windows NT. Para ello reclutaron a Dave Cutler, uno de los jefes analistas de VMS en Digital Equipment Corporation (hoy parte de Compaq que en 2005 fue comprada por HP) para convertir NT en un sistema más competitivo. Cutler había estado desarrollando un sucesor del VMS en DEC (Digital Equipment Corporation) llamado Mica, y cuando DEC abandonó el proyecto se llevó sus conocimientos y algunos ingenieros a Microsoft. DEC también creyó que se llevaba el código de Mica a Microsoft y entabló una demanda. Microsoft finalmente pagó 150 millones de dólares y acordó dar soporte al microprocesador Alpha de DEC en NT.

Siendo un sistema operativo completamente nuevo, Windows NT sufrió problemas de compatibilidad con el hardware y el software existentes. También necesitaba gran cantidad de recursos y éstos estaban solamente disponibles en equipos grandes y caros. Debido a esto muchos usuarios no pudieron pasarse a Windows NT. La interfaz gráfica de NT todavía estaba basada en la de Windows 3.1 que era inferior a la Workplace Shell de OS/2

### Windows NT 3.1

Windows NT 3.1 (la estrategia de marketing de Microsoft era que Windows NT pareciera una continuación de Windows 3.1) apareció en su versión beta para desarrolladores en la Conferencia de Desarrolladores Profesionales de Julio de 1992 en San Francisco. Microsoft anunció su intención de desarrollar un sucesor para Windows NT y Chicago (que aún no había sido lanzada). Este sucesor habría de unificar ambos sistemas en uno sólo y su nombre clave era Cairo. (Visto en retrospectiva Cairo fue un proyecto más difícil de lo que Microsoft había previsto y como resultado NT y Chicago no sería unificados hasta la aparición de Windows XP). Las versiones antiguas de Windows NT se distribuían en disquettes y requerían unos elevados recursos de hardware (además de soportar relativamente poco hardware) por lo que no se difundieron demasiado hasta llegar a Windows NT 4.0 y sobre todo a Windows 2000. Por primera vez daba soporte para el sistema de archivos NTFS.

### Windows NT 3.5/3.51

Cabe destacar que la interfaz gráfica de Windows NT 3.5 y Windows 3.51 era la misma que la de sus predecesores, Windows NT 3.1 y Windows 3.1, con el Administrador de Programas. Por otra parte, Microsoft distribuyó un añadido llamado NewShell, cuyo nombre completo es "Shell Technology Preview Update", que no era otra cosa más que una versión Beta de la nueva interfaz gráfica de Windows 95 y NT 4.0, con el botón y menú inicio, pero para Windows NT 3.5x. Su función principal era que los usuarios de Windows evaluaran el nuevo interfaz gráfico, que iba a ser presentado en Windows 95 y NT 4.0, pero como "daño colateral" le daba a Windows NT 3.5x la nueva interfaz gráfica.

### Windows NT 4.0

Windows NT 4.0 presentaba varios componentes tecnológicos de vanguardia y soporte para diferentes plataformas como MIPS, ALPHA, Intel, etc. Las diferentes versiones como Workstation, Server, Terminal server, Advancer server, permitían poder adaptarlo a varias necesidades. El uso de componentes como tarjetas de sonido, modems, etc, tenían que ser diseñados específicamente para este sistema operativo.

### Windows 95

Microsoft adoptó "Windows 95" como nombre de producto para Chicago cuando fue publicado en Agosto de 1995. Chicago iba encaminado a incorporar una nueva interfaz gráfica que compitiera con la de OS/2. Aunque compartía mucho código con Windows 3.x e incluso con MS-DOS, también se pretendía introducir arquitectura de 32 bits y dar soporte a multitarea preventiva, como OS/2 o el mismo Windows NT. Sin embargo sólo una parte de Chicago comenzó a utilizar arquitectura de 32 bits, la mayor parte siguió usando una arquitectura de

16 bits. Microsoft argumentaba que una conversión completa retrasaría demasiado la publicación de Chicago y sería demasiado costosa.

Microsoft desarrolló una nueva API para remplazar la API de Windows de 16 bits. Esta API fue denominada Win32, desde entonces Microsoft denominó a la antigua API de 16 bits como Win16. Esta API fue desarrollada en tres versiones: una para Windows NT, una para Chicago y otra llamada Win32s, que era un subconjunto de Win32 que podía ser utilizado en sistemas con Windows 3.1., de este modo Microsoft intentó asegurar algún grado de compatibilidad entre Chicago y Windows NT, aunque los dos sistemas tenían arquitecturas radicalmente diferentes

Windows 95 tenía dos grandes ventajas para el consumidor medio. Primero, aunque su interfaz todavía corría sobre MS-DOS, tenía una instalación integrada que le hacía aparecer como un solo sistema operativo (ya no se necesitaba comprar MS-DOS e instalar Windows encima). Segundo, introducía un subsistema en modo protegido que estaba especialmente escrito a procesadores 80386 o superiores, lo cual impediría que las nuevas aplicaciones Win32 dañaran el área de memoria de otras aplicaciones Win32. En este aspecto Windows 95 se acercaba más a Windows NT, pero a la vez, dado que compartía código de Windows 3.x, las aplicaciones podían seguir bloqueando completamente el sistema en caso de que invadiesen el área de aplicaciones de Win16.

Tenía también como novedad el incluir soporte para la tecnología Plug&Play. Windows 95 se convirtió en el primer gran éxito de los de Redmond a nivel mundial. La evolución de Internet y la potencia de los equipos, cada vez más capaces, dio lugar a un binomio en el que Intel y Microsoft dominaban el panorama mundial con solvencia. Los fabricantes comenzaban a volcarse en este sistema a la hora de sacar sus controladores de dispositivos y, aunque con algunos problemas por incompatibilidades inevitables, el éxito de la plataforma fue absoluto.

### **Windows 98**

El 25 de Junio de 1998 llegó Windows 98. Incluía nuevos controladores de hardware y el sistema de archivos FAT32 (también soportado por Windows 95 OSR 2 y OSR 2.5) que soportaba particiones mayores a los 2 GB permitidos por Windows 95. Dio soporte también a las nuevas tecnologías como DVD, FireWire, USB o AGP. Era novedosa también la integración del explorador de Internet en todos los ámbitos del sistema. La principal diferencia de Windows 98 sobre Windows 95 era que su núcleo había sido modificado para permitir el uso de controladores de Windows NT en Windows 9x y viceversa. Esto se consiguió con la migración de parte del núcleo de Windows NT a Windows 98 aunque éste siguiera manteniendo su arquitectura MS-DOS/Windows GUI. Esto permitió la reducción de costos de producción, dado que Windows NT y Windows 98 ahora podían utilizar casi idénticos controladores.

### **Windows 98 Second Edition**

En 1999 Microsoft sacó al mercado Windows 98 Second Edition, cuya característica más notable era la capacidad de compartir entre varios equipos una conexión a Internet a través de una sola línea telefónica. También eliminaba gran parte de los errores producidos por Internet Explorer en el sistema. Esta versión es la más estable de todas las de esta serie, y aún se sigue utilizando en muchos equipos.

### **Windows Millenium Edition**

En 2000 Microsoft introdujo Windows ME que era una copia de Windows 98 con más aplicaciones añadidas. Windows ME fue un proyecto rápido de un año para rellenar el hueco entre Windows 98 y el nuevo Windows XP, y eso se notó mucho en la poca estabilidad de esta versión. En teoría Windows 2000 iba a ser la unificación entre las dos familias de Windows, la empresarial y la de hogar, pero por retrasos, se lanzó este pequeño avance. En esta versión se aceleraba el inicio del sistema y oficialmente ya no se podía distinguir entre el MS-DOS y el entorno gráfico (aunque aparecieron parches que permitían volver a separarlo como se hacía en versiones anteriores).

Esta versión no traía unidad de proceso de 16 bits, centrándose únicamente en la compatibilidad con nuevo hardware de 32 bits. Como consecuencia, sólo funcionaba correctamente con los equipos nuevos que lo traían instalado, ya que si se instalaba sobre un equipo antiguo (mediante una actualización de software) el hardware de 16 bits era más complejo de configurar, o bien no funcionaba en absoluto.

Cabe destacar que este Sistema Operativo fue muy popular por sus continuos errores (bugs).

### **Windows 2000**

En este mismo año vio la luz Windows 2000, una nueva versión de Windows NT muy útil para los administradores de sistemas y con una gran cantidad de servicios de red y lo más importante: admitía dispositivos Plug&Play que habían sido y seguían siendo un problema con Windows NT.

La familia de Windows 2000 estaba formada por varias versiones del sistema: una para las estaciones de trabajo (Windows 2000 Professional) y varias para servidores (Windows 2000 server, advanced server, datacenter server).

Windows 2000 incorporaba importantes innovaciones tecnológicas para entornos Microsoft, tanto en nuevos servicios como en la mejora de los existentes. Algunas de las características que posee son:

Almacenamiento:

- Soporte para FAT16, FAT32 y NTFS.
- Encriptación de archivos (EFS).
- Servicio de indexación.
- Sistema de archivos distribuido (DFS).
- Nuevo sistema de backup (ASR).
- Sistema de tolerancia a fallos (RAID) con discos dinámicos (software).

Comunicaciones:

- Servicios de acceso remoto (RAS, VPN, RADIUS y Enrutamiento).
- Nueva versión de IIS con soporte para HTTP/1.1.
- Directorio Activo.
- Balanceo de carga (clustering)
- Servicios de instalación desatendida por red (RIS).
- Servicios nativos de Terminal Server.

Estos avances marcan un antes y un después en la historia de Microsoft.

### **Windows XP**

La unión de Windows NT/2000 y la familia de Windows 9.x se alcanzó con Windows XP puesto en venta en 2001 en su versión Home y Professional. Windows XP usa el núcleo de Windows NT. Incorpora una nueva interfaz y hace alarde de mayores capacidades multimedia. Además dispone de otras novedades como la multitarea mejorada, soporte para redes inalámbricas y asistencia remota. Inmediatamente después de haber lanzado el último Service Pack (SP2) Microsoft diseñó un sistema orientado a Empresas y Corporaciones llamado Microsoft Windows XP Corporate Edition, algo similar al Windows XP Profesional, solo que diseñado especialmente a Empresas. En el apartado multimedia, XP da un avance con la versión Media Center. Esta versión ofrece una interfaz de acceso fácil a todo lo relacionado con multimedia (TV, fotos, reproductor DVD, Internet...)

## Windows Server 2003

Sucesor de la familia de servidores de Microsoft a Windows 2000 Server. Es la versión de Windows para servidores lanzada por Microsoft en el año 2003. Está basada en el núcleo de Windows XP, al que se le han añadido una serie de servicios, y se le han bloqueado algunas de sus características (para mejorar el rendimiento, o simplemente porque no serán usadas).

## Windows Vista

Windows Vista apareció en el mercado el 30 de enero de 2007. Cabe destacar los continuos retrasos en las fechas de entrega del sistema operativo. Inicialmente se anunció su salida al mercado a inicios-mediados de 2006, posteriormente y debido a problemas durante el proceso de desarrollo, se retrasó su salida hasta finales de 2006. El último retraso trasladó la fecha hasta finales de enero de 2007. Estos continuos retrasos han llevado a Microsoft a tomar diversas medidas para minimizar los gastos extras derivados de los retrasos. Por ejemplo, en Argentina, se podrá comprar Windows Vista con un "ticket" que la persona adquiere al momento de comprar un nuevo equipo PC que no tiene instalado todavía Windows Vista. Podrán canjear el "ticket" por una copia original de Windows Vista y así actualizar su sistema. Las diferentes versiones que se podrán adquirir son tres para el consumidor, Vista Home Basic, Vista Home Premium y Ultimate Edition y dos versiones para las empresas, Vista Business y Vista Enterprise, más Vista Starter, para los mercados emergentes. Los requisitos mínimos para que la versión básica de Windows Vista (Home Basic - Starter) funcione en un equipo son los siguientes:

- Procesador de 32 Bits de por lo menos 800Mhz
- 512 MB de RAM (para trabajar a una velocidad aceptable), se recomienda 1GB
- Tarjeta Gráfica compatible con DirectX 9 con por lo menos 32MB de memoria, se recomienda 128MB, aunque esta Versión sin Aero no requiere una tarjeta muy avanzada
- Disco duro de 40GB con 15GB disponibles
- Grabadora y lectora de DVD-ROM.

Para disfrutar de todas las características como Aero (Home Premium - Ultimate), se requiere de una computadora con estas características:

- Procesador de 32 Bits de por lo menos 1Ghz (se recomienda procesador de doble núcleo como Intel Pentium D (64 bit))
- 1GB de RAM (recomiendan 2GB para el Aero)
- Compatibilidad con gráficos DirectX 9 con un controlador WDDM, 128 MB de memoria gráfica (mínimo)2, Pixel Shader 2.0 y 32 bits por pixel.
- Disco duro de 40GB con 15GB disponibles
- Grabadora y lectora de DVD
- Conexión a internet
- Salida de audio

Cabe destacar que durante su desarrollo fue conocido como Windows Longhorn. Las Ediciones de Windows Vista serán las siguientes:

Windows Vista Starter, destinada a sustituir a Windows XP Starter Edition. Estará dirigido a mercados emergentes, y estará muy limitado.

Windows Vista Home Basic parecido al Windows XP Home Edition, no incluirá "Aero Glass" con efectos translúcidos. Soportará 8 GB de RAM.

Windows Vista Home Premium más similar al Windows XP Media Center Edition. Por ejemplo soportará HDTV y hasta 16 GB de memoria RAM.

Windows Vista Business equivalente al Windows XP Profesional. No incluye características del Media Edition, pero a cambio ofrece herramientas más orientadas hacia los negocios, como

pueden ser soporte de Fax, servidor web IIS, hasta 128 GB. No requerirá activación del producto.

Windows Vista Enterprise Basado en la anterior versión (Windows Vista Business). Tiene todo lo del anterior, más Virtual PC, interfaz multi-idioma, y podrá soportar aplicaciones UNIX. No se venderá a través de los medios tradicionales de venta.

Windows Vista Ultimate combina las características del Home Premium junto con las del Enterprise. Como las versiones para el mercado profesional, no requerirá de activación del producto.

Además de estas versiones, estarán disponibles para el mercado Europeo las ediciones "Home Basic N" y "Business N", idénticas a las anteriores, salvo por no contar con el Windows Media Player.

Las versiones Home, Home Premium y Ultimate se venderán en el mismo DVD, y se podrá pasar a una versión superior simplemente pagando una actualización de la licencia a través del Windows Anytime Upgrade.

También cabe destacar que Windows Vista traerá una nueva interfaz grafica llamada Aero que es una evolución de la interfaz gráfica denominada de Windows XP.

Versiones para Tablet PC's

- Windows XP Tablet PC Edition  
Versiones para pequeños dispositivos
- Windows CE
- Windows Mobile

### Sistemas de archivos

El sistema de archivos utilizado por estos sistemas operativos comenzó siendo FAT16 o FAT "a secas". La primera versión de Windows en incorporar soporte nativo para FAT32 fue Windows 95 OSR2. Por otro lado, los Sistemas Operativos basados en NT emplean los sistemas de archivos NTFS desde el origen y a partir de Windows 2000 se otorgó también soporte para FAT32.

#### 1.3.4 Hardware

Término del inglés (Hardware) que se utiliza generalmente para describir los artefactos físicos de una tecnología. En un sentido más corto, el hardware puede ser equipo militar importante, equipo electrónico, o equipo informático. En la Informática se denomina hardware o soporte físico al conjunto de elementos materiales que componen una computadora. Hardware también son los componentes físicos de una computadora tales como el disco duro, CD-ROM, disquetera (floppy), etc. En dicho conjunto se incluyen los dispositivos electrónicos y electromecánicos, circuitos, cables, tarjetas, armarios o cajas, periféricos de todo tipo y otros elementos físicos.

El hardware se refiere a todos los componentes físicos (que se pueden tocar) de la computadora: discos, unidades de disco, monitor, teclado, ratón (mouse), impresora, placas, chips y demás periféricos. En cambio, el software es intangible, existe como ideas, conceptos, símbolos, pero no tiene sustancia. Una buena metáfora sería un libro: las páginas y la tinta son el hardware, mientras que las palabras, oraciones, párrafos y el significado del texto son el software. Una computadora sin software sería tan inútil como un libro con páginas en blanco.

##### 1.3.4.1 Tipos de Hardware

Se clasifica generalmente en básico y complementario, entendiendo por básico todo aquel dispositivo necesario para iniciar el funcionamiento de la computadora, y el complementario como su nombre lo dice sirve para realizar funciones específicas o más allá de las básicas.



Periféricos de entrada: Son los que permiten que el usuario aporte información exterior. Entre ellos podemos encontrar: teclado, ratón (mouse), escáner, SAI (Sistema de Alimentación Ininterrumpida), micrófono, etc.

Periféricos de salida: Son los que muestran al usuario el resultado de las operaciones realizadas por la computadora. En este grupo podemos encontrar: monitor, impresora, altavoces, etc.

Periféricos de entrada/salida: Son los dispositivos que pueden aportar simultáneamente información exterior a la computadora y al usuario. Aquí se encuentran: módem (Modulador/Demodulador), unidades de almacenamiento (Discos duros, disquetes o floppy), ZIP, Memorias de pequeño tamaño, flash, etc).

Hardware: Aparato electrónico capaz de interpretar y ejecutar comandos programados para operaciones de entrada, salida, cálculo y lógica. Las computadoras:

1. Reciben entradas. La entrada son los datos que se capturan en un sistema de computación para su procesamiento.
2. Producen salidas. La salida es la presentación de los resultados del procesamiento.
3. Procesan información
4. Almacenan información

Todo sistema de cómputo tiene componentes de hardware dedicados a estas funciones:

1. Dispositivos de entrada
2. Dispositivos de salida
3. Unidad central de procesamiento. Es la computadora real, la "inteligencia" de un sistema de computación.
4. Memoria y dispositivos de almacenamiento.

Cada dispositivo de entrada es sólo otra fuente de señales eléctricas; cada dispositivo de salida no es más que otro lugar al cual enviar señales; cada dispositivo de almacenamiento es lo uno o lo otro, dependiendo de lo que requiera el programa; no importa cuales sean los dispositivos de entrada y salida si son compatibles. Los elementos fundamentales que justifican el uso de las computadoras, radican en que las computadoras son:

- Útiles.
- Baratas: tanto con respecto a sí mismas como con respecto al costo de la mano de obra.
- Fáciles de utilizar.

Tecnologías y avances

- 1ª generación: Con tubos de vacío, tubos de vidrio del tamaño de una bombilla que albergaban circuitos eléctricos. Estas máquinas eran muy grandes caras y de difícil operación.
- 2ª generación: Con transistores. Máquinas más pequeñas, confiables y económicas.
- 3ª generación: Con la tecnología que permitió empaquetar cientos de transistores en un circuito integrado de un chip de silicio.
- 4ª generación: Con el microprocesador, que es una computadora completa empaquetada en un solo chip de silicio.

### 1.3.4.2 Tipos de Procesadores

#### 1.3.4.2.1 Uniprocesadores

Este tipo de procesador tiene la arquitectura mas simple, esta compuesto por un solo procesador (interpreta y ejecuta las instrucciones de la computadora), el cache está conectado con la memoria y los dispositivos de entrada y salida a través de un bus compartido.

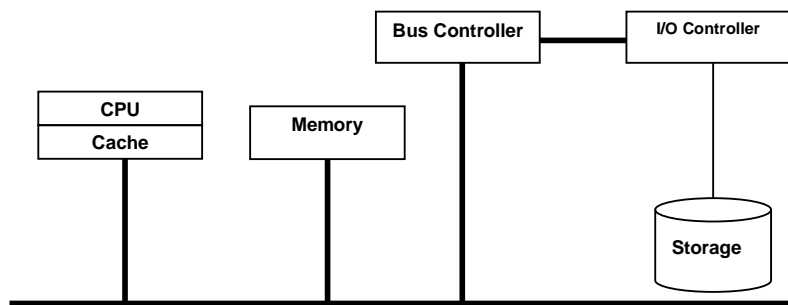
El bus de los dispositivos se comunica con el bus del procesador y la memoria mediante un puente entre los buses. Un ejemplo común es la arquitectura de los procesadores 486 y 586

Debido a la naturaleza de estas arquitecturas, aplicaciones o servidores, no pueden ser implementados en este tipo de procesadores. La mayoría de las aplicaciones o motores bases de datos, necesitan ejecutar muchos procesos al mismo tiempo, y un procesador sencillo no podría ser eficiente.

Un ejemplo muy claro es Oracle, una base de datos Oracle no podría funcionar correctamente en este tipo de procesadores porque Oracle necesita ejecutar varios procesos como: DBWR, LGWR, PMON, SMON, etc.

La siguiente imagen muestra la arquitectura de un procesador simple.

Figura 21: Arquitectura procesador simple



#### 1.3.4.2.2 Multiprocesadores

El concepto de trabajo en equipo, el dividir una tarea entre varias unidades de ejecución para que pueda completarse más rápidamente, fue naturalmente llevado al mundo de las computadoras prácticamente desde sus inicios. David Slotnick, quien fuera colaborador de Von Neumann, le hizo la propuesta de una máquina que contara con varias unidades de procesamiento central; sin embargo, al sentir de Von Neumann, la tecnología de la época no permitía la realización de semejante proyecto. Aun así, Slotnick continuó con sus ideas, lo que eventualmente daría origen a la ILLIAC IV (1964), considerada una de las primeras computadoras masivamente paralelas de la historia. En la actualidad, las computadoras más rápidas del mundo son las máquinas masivamente paralelas.

El cómputo paralelo ofrece una serie de ventajas que lo hacen particularmente atractivo para los requerimientos de capacidad de cómputo, en particular los de la comunidad científica. Una de estas ventajas es económica. El uso de componentes comúnmente disponibles, en grandes cantidades, permite ofrecer mayor rendimiento, a un precio menor que el de máquinas con procesadores especialmente diseñados (como por ejemplo las máquinas de procesadores vectoriales y de propósito específico). Adicionalmente, las computadoras paralelas son inherentemente escalables, permitiendo actualizarlas para adecuarlas a una necesidad creciente. Las arquitecturas "tradicionales" se actualizan haciendo los procesadores existentes obsoletos por la introducción de nueva tecnología a un costo posiblemente elevado. Por otro lado, una arquitectura paralela se puede actualizar en términos de rendimiento simplemente agregando más procesadores.

En ocasiones se menciona también la limitante física; existen factores que limitan la velocidad máxima de un procesador, independientemente del factor económico. Barreras físicas infranqueables, tales como la velocidad de la luz, efectos cuánticos al reducir el tamaño de los elementos de los procesadores, y problemas causados por fenómenos eléctricos a pequeñas escalas, restringen la capacidad máxima de un sistema uniprocador, dejando la opción obvia de colocar muchos procesadores para realizar cálculos cooperativamente.

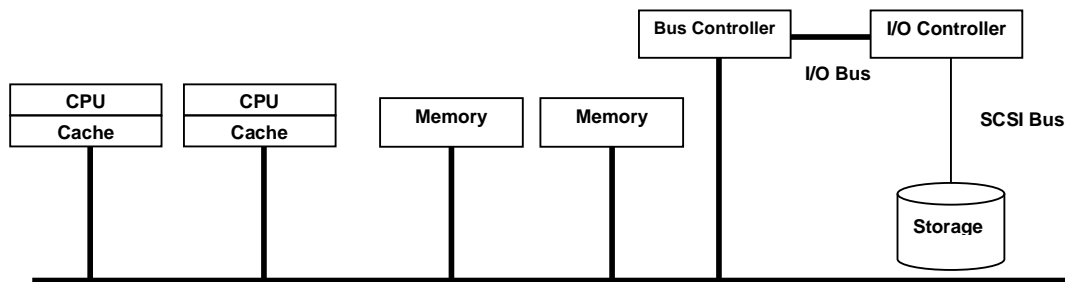
Como toda nueva arquitectura, las máquinas paralelas poseen características, y plantean ventajas y desventajas, que obligan a considerar cuidadosamente su utilización. También, dentro del mundo de las máquinas paralelas, existen dos enfoques con distintas características: la arquitectura SMP y la arquitectura MPP

#### 1.3.4.2.3 Symmetric Multi-Processing SMP

Es el acrónimo de *Symmetric Multi-Processing*, multiproceso simétrico. Se trata de un tipo de arquitectura de computadoras en que dos o más procesadores comparten una única memoria central

El enfoque más sencillo para una máquina con múltiples procesadores es el esquema SMP (*Symmetrical Multiprocessing*, Multiproceso Simétrico). Una arquitectura SMP es básicamente una expansión de una arquitectura tradicional pero con la adición de varios procesadores que comparten todos los demás recursos del sistema (memoria principal, almacenamiento secundario, periféricos de entrada y salida). En esta arquitectura no se establece distinción entre los procesadores; todos son jerárquicamente iguales y pueden ejecutar tareas indistintamente. De esta característica viene el nombre de "simétrica". Un diagrama de una arquitectura SMP genérica se muestra en la figura:

Figura 22: Arquitectura Multiprocesador



En general una arquitectura SMP tiene un equivalente en uniprocador, y naturalmente un sistema SMP puede ejecutar simultáneamente varios programas o aplicaciones, que normalmente podrían ejecutarse en el sistema uniprocador, de manera independiente. Sin embargo, para el uso de aplicaciones que aprovechen los múltiples procesadores para expedir la realización de cálculos, nos interesa que dichos procesos no sean totalmente independientes, buscando entonces que cuenten con manera de comunicarse para distribuirse información, compartir y consolidar resultados.

Ya que en un sistema SMP los procesadores comparten todos los periféricos y recursos, el esquema más obvio para comunicarse en una arquitectura SMP es el uso de memoria compartida. Como el nombre lo indica, en este esquema los procesadores tienen acceso a un espacio de direcciones común; esto puede ser todo el espacio de direcciones o únicamente un área designada para memoria compartida, permitiendo a cada proceso contar con un área exclusiva para sus requerimientos.

La memoria compartida es un esquema conceptualmente simple de utilizar. Sin embargo presenta ciertas limitaciones. Una de ellas, ya que se puede tener a dos o más procesadores manipulando la misma área de memoria, es que se puede caer en inconsistencias donde un procesador espera un dato que ha sido modificado por otro. Esto también puede provocar condiciones de competencia (*"race conditions"*) y atoramientos (*"deadlocks"*), que son problemas clásicos de la sincronización de procesos, pero que no pueden dejar de tomarse en

cuenta en una arquitectura SMP. Estas condiciones pueden resolverse utilizando mecanismos de sincronización de procesos, como semáforos, monitores y secciones críticas.

La limitación más importante de la arquitectura SMP, en términos del rendimiento máximo que puede alcanzarse, es la posibilidad de saturación de los buses del sistema. Ya que todos los procesadores tienen acceso al mismo bus de memoria, y en general a todos los periféricos que se encuentran comunicados comúnmente por buses, conforme se incrementa el número de procesadores se incrementa también el tráfico en dichos buses. Esto causa una saturación que finalmente termina por negar el incremento de rendimiento obtenido teniendo varios procesadores. Por esta razón una arquitectura SMP difícilmente puede escalar más allá de algunas decenas de procesadores.

#### **1.3.4.2.4 Massively parallel processing (MPP)**

Ya que el problema es el hecho de contar con memoria compartida y sus limitaciones, se propuso un esquema de una máquina paralela que consta de varias unidades de procesamiento básicamente independientes. En efecto cada una de estas unidades, conocida como "nodo", es prácticamente una computadora en sí misma, contando con su propio procesador, memoria no compartida, y que se comunica con las demás unidades de procesamiento a través de un canal provisto exclusivamente para este propósito. Este tipo de máquinas se conocen como computadoras masivamente paralelas o máquinas MPP (*Massively Parallel Processing*, procesamiento masivamente paralelo).

Una máquina MPP presenta una serie de consideraciones importantes derivadas de su arquitectura, que se deben tomar en cuenta al escribir programas que pretendan aprovechar su naturaleza multiprocesador. Obviamente la característica más importante es el hecho de que, en cada nodo, cada procesador opera básicamente como una computadora independiente, ejecutando su propio código independiente de los demás procesadores, y teniendo un área de memoria con datos también independientes.

Desde luego, para que esta organización redunde en un mayor desempeño, se requiere colaboración entre los nodos. Como se mencionó, una máquina MPP debe contar con un canal que permita a los nodos comunicarse entre sí, a fin de intercambiar datos y coordinar sus operaciones. Ya que el objetivo principal de una máquina MPP es obtener alto rendimiento, se busca que este canal de comunicaciones sea lo más eficiente posible, en términos tanto de ancho de banda como de tiempo de latencia. En la mayoría de los casos este canal será un bus propietario, diseñado por el fabricante del equipo MPP.

Para tener acceso a información fuera de su propia área de memoria, los nodos se comunican entre sí, regularmente empleando un esquema de paso de mensajes. Esto resuelve el problema de saturación del bus de comunicaciones, pues éste sólo se emplea cuando se está realizando comunicación entre los nodos. De esta manera se tiene una arquitectura que puede escalarse a varios cientos o miles de procesadores (las máquinas MPP más grandes en la actualidad tienen alrededor de 10 mil procesadores).

Sin embargo el tener varias secciones de memoria independientes complica la programación en este tipo de arquitecturas. En una arquitectura MPP la distribución de trabajo entre los nodos es una consideración vital al diseñar cualquier aplicación. Se debe tomar en cuenta la sincronización de datos entre los nodos, y toda comunicación entre ellos debe realizarse explícitamente por medio de llamadas al mecanismo de paso de mensajes.

#### **1.3.4.2.5 Non uniform memory access (NUMA)**

NUMA es el acrónimo de "Non-uniform Memory Access", acceso no uniforme a memoria.

Se trata de un tipo de arquitectura de ordenadores que generalmente se utiliza en servidores de alta gama y servidores particionables debido a su mayor escalabilidad.

Las computadoras con arquitectura NUMA se caracterizan por el hecho de que el tiempo de acceso a la memoria depende del fragmento de memoria accedido. Una computadora NUMA

se compone de módulos o celdas que agrupan en un circuito integrado un conjunto mínimo de recursos de computación, a saber:

- Microprocesadores (generalmente, dos o cuatro).
- Memoria.
- Ranuras de entrada/salida, donde se pueden interconectar tarjetas de red, controladoras de discos, etc.

Varias de estas celdas se pueden unir a través de un backplane (es un componente que facilita la interconexión de los módulos o celdas que la componen) para construir una computadora. De esta manera, un microprocesador puede acceder muy rápidamente a la memoria que está en la misma celda. Si, por el contrario, tiene que acceder a memoria situada en otra celda será necesario atravesar el *backplane*. Por tanto, requerirá un tiempo mayor. Este es el motivo de la denominación NUMA.

#### 1.3.4.3 Tipos de Servidores

En informática, un servidor es un tipo de software que realiza ciertas tareas en nombre de los usuarios. El término servidor ahora también se utiliza para referirse al equipo físico en el cual funciona ese software, una máquina cuyo propósito es proveer datos de modo que otras máquinas puedan utilizar esos datos.

Este uso dual puede llevar a confusión. Por ejemplo, en el caso de un servidor web, este término podría referirse a la máquina que almacena y maneja los sitios web, y en este sentido es utilizada por las compañías que ofrecen hosting u hospedaje. Alternativamente, el servidor web podría referirse al software, como el servidor de http de Apache, que funciona en la máquina y maneja la entrega de los componentes de las páginas web como respuesta a peticiones de los navegadores de los clientes.

Los archivos para cada sitio de Internet se almacenan y se ejecutan en el servidor. Hay muchos servidores en Internet y muchos tipos de servidores, pero comparten la función común de proporcionar el acceso a los archivos y servicios.

Un servidor provee información a las computadoras que se conecten a él. Cuando los usuarios se conectan a un servidor pueden acceder a programas, archivos y otra información del servidor.

En la Web, un servidor web es una computadora que usa el protocolo http para enviar páginas web a la computadora de un usuario cuando el usuario las solicita.

Los servidores web, servidores de correo y servidores de bases de datos son a lo que tiene acceso la mayoría de la gente al usar Internet. Algunos servidores manejan solamente correo o solamente archivos, mientras que otros hacen más de un trabajo, ya que una misma computadora puede tener diferentes programas de servidor funcionando al mismo tiempo. Los servidores se conectan a la red mediante una interfaz que puede ser una red verdadera o mediante conexión vía línea telefónica o digital.

Esta lista agrupa los diversos tipos de servidores del mercado actual:

**Plataformas de Servidor (*Server Platforms*):** Un término usado a menudo como sinónimo de sistema operativo. La plataforma es el hardware o software subyacentes para un sistema, es decir, el motor que dirige el servidor.

**Servidores de Aplicaciones (*Application Servers*):** Designados a veces como un tipo de *middleware* (software que conecta dos aplicaciones). Los servidores de aplicaciones ocupan una gran parte del territorio entre los servidores de bases de datos y el usuario, y a menudo los conectan.

**Servidores de Audio/Video (Audio/Video Servers):** Los servidores de Audio/Video añaden capacidades multimedia a los sitios web permitiéndoles mostrar contenido multimedia en forma de flujo continuo (*streaming*) desde el servidor.

**Servidores de Chat (Chat Servers):** Los servidores de chat permiten intercambiar información a una gran cantidad de usuarios ofreciendo la posibilidad de llevar a cabo discusiones en tiempo real.

**Servidores de Fax (Fax Servers):** Un servidor de fax es una solución ideal para organizaciones que tratan de reducir el uso del teléfono pero necesitan enviar documentos por fax.

**Servidores FTP (FTP Servers):** Uno de los servicios más antiguos de Internet, File Transfer Protocol permite mover uno o más archivos.

**Servidores Groupware (Groupware Servers):** Un servidor groupware es un software diseñado para permitir colaborar a los usuarios, sin importar la localización, vía Internet o vía Intranet corporativo y trabajar juntos en una atmósfera virtual.

**Servidores IRC (IRC Servers):** Otra opción para usuarios que buscan la discusión en tiempo real, Internet Relay Chat consiste en varias redes de servidores separadas que permiten que los usuarios conecten el uno al otro vía una red IRC.

**Servidores de Listas (List Servers):** Los servidores de listas ofrecen una manera mejor de manejar listas de correo electrónico, bien sean discusiones interactivas abiertas al público o listas unidireccionales de anuncios, boletines de noticias o publicidad.

**Servidores de Correo (Mail Servers):** Casi tan ubicuos y cruciales como los servidores web, los servidores de correo mueven y almacenan el correo electrónico a través de las redes corporativas (vía LANs y WANs) y a través de Internet.

**Servidores de Noticias (News Servers):** Los servidores de noticias actúan como fuente de distribución y entrega para los millares de grupos de noticias públicos actualmente accesibles a través de la red de noticias USENET.

**Servidores Proxy (Proxy Servers):** Los servidores proxy se sitúan entre un programa del cliente (típicamente un navegador) y un servidor externo (típicamente otro servidor web) para filtrar peticiones, mejorar el funcionamiento y compartir conexiones.

**Servidores Telnet (Telnet Servers):** Un servidor telnet permite a los usuarios entrar en una computadora huésped y realizar tareas como si estuviera trabajando directamente en esa computadora.

**Servidores Web (Web Servers):** Básicamente, un servidor web sirve contenido estático a un navegador, carga un archivo y lo sirve a través de la red

#### 1.3.4.4 Tipos de Almacenamiento

Una de las necesidades principales, de una empresa es el almacenamiento de la información, es un tanto difícil definir las cosas importantes de una empresa, pero definitivamente, la protección de su información es una de ellas.

##### 1.3.4.4.1 Dispositivo de almacenamiento

Es todo aparato que se utilice para grabar los datos de la computadora de forma permanente o temporal. Una unidad de disco, junto con los discos que graba, es un dispositivo de almacenamiento. A veces se dice que una computadora tiene dispositivos de almacenamiento primarios (o principales) y secundarios (o auxiliares). Cuando se hace esta distinción, el dispositivo de almacenamiento primario es la memoria de acceso aleatorio (RAM) de la computadora, un dispositivo de almacenamiento permanente pero cuyo contenido es

temporal. El almacenamiento secundario incluye los dispositivos de almacenamiento más permanentes, como unidades de disco y de cinta.

La velocidad de un dispositivo se mide por varios parámetros: la velocidad máxima que es capaz de soportar, que suele ser relativa, en un breve espacio de tiempo y en las mejores condiciones; la velocidad media, que es la que puede mantener de forma constante en un cierto período de tiempo, y, por último, el tiempo medio de acceso que tarda el dispositivo en responder a una petición de información debido a que debe empezar a mover sus piezas, a girar y buscar el dato solicitado. Este tiempo se mide en milisegundos (ms), y cuanto menor sea esta cifra más rápido será el acceso a los datos.

#### **1.3.4.4.2 Unidades de información**

Bit (Binary Digit o dígito binario): Adquiere el valor 1 ó 0 en el sistema numérico binario. En el procesamiento y almacenamiento informático un bit es la unidad de información más pequeña manipulada por la computadora y está representada físicamente por un elemento como un único pulso enviado a través de un circuito, o bien como un pequeño punto en un disco magnético capaz de almacenar un 0 o un 1.

La representación de información se logra mediante la agrupación de bits para lograr un conjunto de valores mayor que permite manejar mayor información. Por ejemplo, la agrupación de ocho bits compone un byte que se utiliza para representar todo tipo de información, incluyendo las letras del alfabeto y los dígitos del 0 al 9.

Código ASCII (American Standard Code for Information Interchange o Código Estándar Americano para el Intercambio de Información): Esquema de codificación que asigna valores numéricos a las letras, números, signos de puntuación y algunos otros caracteres. Al normalizar los valores utilizados para dichos caracteres, ASCII permite que las computadoras y programas informáticos intercambien información.

ASCII incluye 256 códigos divididos en dos conjuntos, estándar y extendido, de 128 cada uno. Estos conjuntos representan todas las combinaciones posibles de 7 u 8 bits, siendo esta última el número de bits en un byte. El conjunto ASCII básico, o estándar, utiliza 7 bits para cada código, lo que da como resultado 128 códigos de caracteres desde 0 hasta 127 (00H hasta 7FH hexadecimal). El conjunto ASCII extendido utiliza 8 bits para cada código, dando como resultado 128 códigos adicionales, numerados desde el 128 hasta el 255 (80H hasta FFH extendido).

En el conjunto de caracteres ASCII básico, los primeros 32 valores están asignados a los códigos de control de comunicaciones y de impresora –caracteres no imprimibles, como retroceso, retorno de carro y tabulación– empleados para controlar la forma en que la información es transferida desde una computadora a otra o desde una computadora a una impresora. Los 96 códigos restantes se asignan a los signos de puntuación corrientes, a los dígitos del 0 al 9 y a las letras mayúsculas y minúsculas del alfabeto latino.

Los códigos de ASCII extendido, del 128 al 255, se asignan a conjuntos de caracteres que varían según los fabricantes de computadoras y programadores de software. Estos códigos no son intercambiables entre los diferentes programas y computadoras como los caracteres ASCII estándar. Por ejemplo, IBM utiliza un grupo de caracteres ASCII extendido que suele denominarse conjunto de caracteres IBM extendido para sus computadoras personales. Apple Computer utiliza un grupo similar, aunque diferente, de caracteres ASCII extendido para su línea de computadoras Macintosh. Por ello, mientras que el conjunto de caracteres ASCII estándar es universal en el hardware y el software de los microordenadores, los caracteres ASCII extendido pueden interpretarse correctamente sólo si un programa, computadora o impresora han sido diseñados para ello.

#### **1.3.4.4.3 Sistema binario**

El sistema binario desempeña un importante papel en la tecnología de los ordenadores. Los primeros 20 números en el sistema en base 2 son 1, 10, 11, 100, 101, 110, 111, 1000, 1001,

1010, 1011, 1100, 1101, 1110, 1111, 10000, 10001, 10010, 10011 y 10100. Cualquier número se puede representar en el sistema binario, como suma de varias potencias de dos.

Las operaciones aritméticas con números en base 2 son muy sencillas. Las reglas básicas son:  $1 + 1 = 10$  y  $1 \times 1 = 1$ . El cero cumple las mismas propiedades que en el sistema decimal:  $1 \times 0 = 0$  y  $1 + 0 = 1$ . La adición, sustracción y multiplicación se realizan de manera similar a las del sistema decimal:

Puesto que sólo se necesitan dos dígitos (o bits), el sistema binario se utiliza en los ordenadores o computadoras. Un número binario cualquiera se puede representar, por ejemplo, con las distintas posiciones de una serie de interruptores. La posición "encendido" corresponde al 1, y "apagado" al 0. Además de interruptores, también se pueden utilizar puntos imantados en una cinta magnética o disco: un punto imantado representa al dígito 1, y la ausencia de un punto imantado es el dígito 0. Los biestables —dispositivos electrónicos con sólo dos posibles valores de voltaje a la salida y que pueden saltar de un estado al otro mediante una señal externa— también se pueden utilizar para representar números binarios. Los circuitos lógicos realizan operaciones con números en base 2. La conversión de números decimales a binarios para hacer cálculos, y de números binarios a decimales para su presentación, se realizan electrónicamente.

#### **1.3.4.4.4 Medidas de almacenamiento de la información**

Byte: unidad de información que consta de 8 bits; en procesamiento informático y almacenamiento, el equivalente a un único carácter, como puede ser una letra, un número o un signo de puntuación.

- Kilobyte (Kb): Equivale a 1.024 bytes.
- Megabyte (Mb): Un millón de bytes o 1.048.576 bytes.
- Gigabyte (Gb): Equivale a mil millones de bytes.

En informática, cada letra, número o signo de puntuación ocupa un byte (8 bits). Por ejemplo, cuando se dice que un archivo de texto ocupa 5.000 bytes estamos afirmando que éste equivale a 5.000 letras o caracteres. Ya que el byte es una unidad de información muy pequeña, se suelen utilizar sus múltiplos: kilobyte (Kb), megabyte (MB), gigabyte (GB)... Como en informática se utilizan potencias de 2 en vez de potencias de 10, se da la circunstancia de que cada uno de estos múltiplos no es 1.000 veces mayor que el anterior, sino 1.024 ( $2^{10} = 1.024$ ). Por lo que  $1 \text{ GB} = 1.024 \text{ MB} = 1.048.576 \text{ Kb} =$  más de 1.073 millones de bytes.

#### **1.3.4.4.5 Los sistemas de archivos**

Todo dispositivo que almacene datos ha de ser formateado antes de poder utilizarlo; es decir, hemos de darle la forma para que reconozca cómo ha de almacenar la información. Esta operación la realiza un programa como el FORMAT (formatear) que lo que hace es darle la forma de sectores y pistas para que un sistema operativo concreto reconozca ese espacio y a la vez destruye toda la información que contenga el citado dispositivo.

Cuando guardamos un archivo, instalamos un programa, etc., el ordenador almacena la información en el disco duro en pequeñas áreas llamadas clústeres. Cuanto menor sea el tamaño del clúster que utilicemos más eficazmente se almacenará la información en el disco. El tamaño del clúster depende del tamaño de la partición (cada una de las divisiones lógicas de un disco, que se asemejan a discos duros separados) y el tamaño de la partición depende del sistema de archivos que utilice. Generalmente, la mayoría de los equipos utilizan una sola partición.

Las versiones anteriores de MS-DOS y Windows utilizan exclusivamente el sistema de archivos FAT16, cuestión por la que no podíamos utilizar nombres de archivos que superasen los ocho caracteres. Pero Windows 98 y la última revisión de Windows 95 vienen con FAT32, sistema de archivos ampliado que mejora el rendimiento del disco y aumenta el espacio de éste, es de 32 bits y permite usar nombres de archivos y carpetas largos (de hasta 255 letras).



El sistema de archivos FAT32 presenta las siguientes ventajas con respecto a FAT16:

- Permite que los programas se abran más rápidamente, cerca de un 36% más rápido.
- Utiliza un tamaño de clúster menor, lo que da como resultado un uso más eficaz del espacio del disco, cerca de un 28% más de espacio en disco.

Si un disco duro tiene menos de 2 Gb y utilizamos el sistema de archivos FAT16 y cambiamos a FAT32 no notaremos gran mejoría, pero si nuestro disco duro es mayor de 2 Gb sí notaremos su eficacia y nos ahorraremos tener que crear varias particiones.

Algunos tipos de almacenamiento utilizados en las organizaciones son los siguientes:

#### **1.3.4.4.6 Discos Duros**

El disco duro (hard disk) es una unidad de almacenamiento magnético de la información. Es un disco metálico (normalmente de aluminio) recubierto con una capa de material magnetizable por sus dos caras (usualmente níquel).

El disco duro magnético está dividido en pistas concéntricas. Cada pista se divide en igual número de bloques radiales denominados sectores. La capacidad de almacenamiento en bytes por cada pista es variable, dependiendo del tamaño de la misma y de la densidad de grabación. En todas las pistas de un mismo disco (desde las exteriores hasta las interiores) cabe la misma cantidad de información, lo que se consigue grabando con mayor densidad en las pistas interiores y menor densidad en las pistas exteriores.

El disco duro normalmente permanece fijo dentro del sistema, aunque existen computadores que admiten discos duros separables del sistema, que pueden ser reemplazados.

Por su gran velocidad de acceso aleatorio a la información (puede llegar a una media de 7 milisegundos), y por su elevada capacidad de almacenamiento (hasta 23 Gbytes), se le utiliza como unidad de memoria exterior activa (interviene en el procesamiento), que interactúa constantemente con la memoria electrónica de las computadoras (memoria RAM).

Para grabar o leer la información, el disco está girando constantemente (hasta con velocidades de 10000 revoluciones por minuto). Existe un brazo exterior al extremo del cual se hallan dos cabezas de lectura-grabación (una por cada cara). Según la pista y sector sobre los que se quiera grabar o leer, el brazo se mueve hacia el exterior o interior del disco hasta la pista deseada y el momento de pasar por el sector respectivo, la cabeza apropiada lee o graba la información. La cabeza no toca físicamente al disco, sino que lo sobrevuela; para ello existe permanentemente un colchón de aire entre la cabeza y la superficie del disco. Si no hay contacto no hay rozamiento, y tanto las cabezas como el disco duran más. Un aterrizaje de una cabeza sobre el disco provoca que ambos queden inutilizados, de ahí la conveniencia de obtener copias de seguridad de la información almacenada en el disco.

Las necesidades actuales de almacenamiento masivo de la información han determinado que los discos duros modernos sean en realidad varios discos agrupados en torno a un eje común. Dentro de dicha unidad, hay tantos brazos de acceso como discos existan, con lo que se multiplica considerablemente la capacidad de almacenamiento, sin disminuir la velocidad de acceso a la información.

Existen dos tipos de discos duros empleados en microcomputadores, que aún se mantienen en el mercado: IDE (integrated drive electronics / electrónica integrada al disco) y SCSI (small computer system interface / interfaz del sistema para computadores pequeños).

##### **1.3.4.4.6.1 Discos Duros IDE**

Son discos duros cuya electrónica de manejo está incorporada al propio disco, por lo que son los más económicos. El tiempo medio de acceso a la información puede llegar a 10 milisegundos (mseg). Su velocidad de transferencia secuencial de información puede alcanzar

hasta 3 Mbytes por segundo (Mbps) bajo la especificación estándar y hasta 11 Mbps bajo la especificación mejorada (Enhanced IDE / EIDE). Su capacidad de almacenamiento en discos modernos alcanza hasta 8 Gbytes). Los controladores IDE pueden manejar hasta 2 discos duros en la versión estándar y hasta 4 discos en la versión mejorada EIDE.

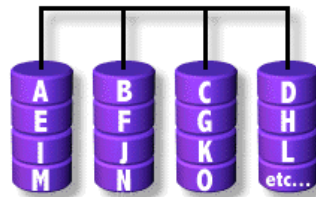
#### 1.3.4.4.7 Niveles de Arreglos RAID

- Existen 6 niveles diferentes los cuales usan múltiples discos duros para mejorar la velocidad, la fiabilidad o ambos.
- Los niveles 1-5 están relacionados con la fiabilidad (redundancia de datos) pero no ofrecen grandes mejoras en lo que se refiere a operaciones de lectura.
- La redundancia permite tener múltiples copias de los mismos datos en varios discos de manera que si alguno falla los datos no se pierden.
- Para que un RAID funcione se requiere invertir en un hardware que permita obtener los beneficios de dicha técnica, de otra manera el maximizar el uso de cache es una buena solución.

#### RAID0:

- Específico para aumentar el performance, no provee redundancia
- Los datos se envían (almacenan) en los diferentes discos
- Para la computadora varios discos se aprecian como si fueran 1 solo.
- Se requieren al menos 2 discos para implementarlo

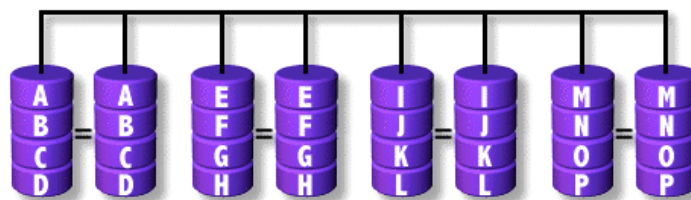
Figura 23: Raid 0



#### RAID1:

- Este nivel produce un "espejo" (mirror) de los datos, los mismos datos son almacenados en 2 discos
- Estos discos pueden o no estar en la misma máquina
- Las lecturas tienden a ser más rápidas que si se hicieran de un solo disco, ya que los datos pueden ser leídos simultáneamente de ambos.
- Las escrituras suelen ser más lentas ya que se están escribiendo 2 copias de los datos.
- Se requieren al menos 2 discos para implementarlo

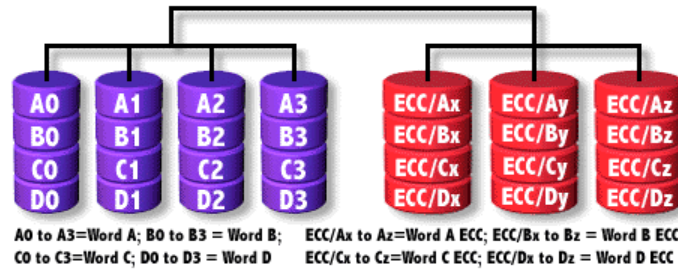
Figura 24: Raid 1



**RAID2:**

- Utiliza códigos de corrección de errores para compensar la carencia de dicha funcionalidad en algunos discos
- Actualmente ya no es muy utilizado debido a que la mayoría de los dispositivos posee dicha característica.

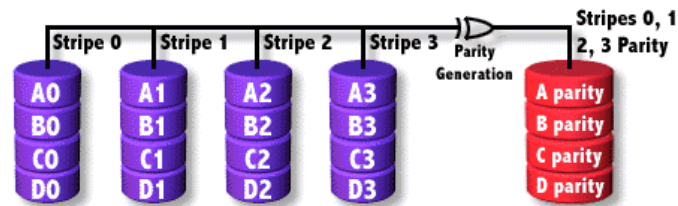
Figura 25: Raid 2



**RAID3:**

- Almacena toda la información en diferentes discos pero a nivel byte.
- Un disco está dedicado exclusivamente a guardar el bit de paridad, de manera que si algún disco falla los datos se puede recuperar gracias a este bit.
- Se requieren al menos 3 discos para implementarlo

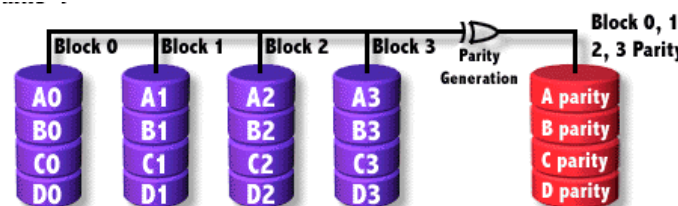
Figura 26: Raid 3



**RAID4:**

- Similar a RAID3 con la diferencia que la paridad se almacena por bloques.
- La paridad también se almacena en un solo disco
- En ambos casos el disco de paridad se puede volver un cuello de botella
- Se requieren al menos 3 discos para implementarlo

Figura 27: Raid 4

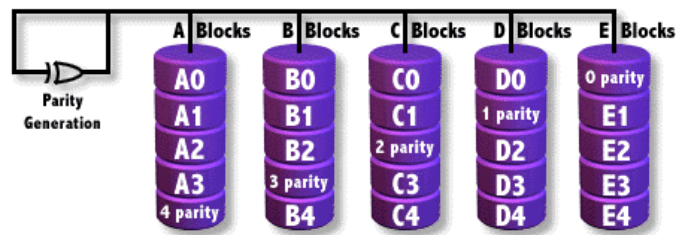


## RAID5:

- Semejante a RAID4 solo que con la diferencia de que la paridad se divide entre todos los discos del arreglo.
- Resiste la caída o falla de un disco
- Las escrituras son lentas debido a que una sencilla operación de escritura requiere leer los datos del bloque así como su paridad, recalcular la paridad con los nuevos datos y escribir entre los diferentes discos del raid.
- Sin embargo la gran mejora en tiempo de lectura minimiza el overhead que se gasta en las escrituras.
- El rendimiento puede llegar a ser igual o mejor que si trabajáramos con un solo disco.
- Se requieren al menos 3 discos para implementarlo

Otros Niveles de RAID, basados en las ideas generales de los 6 niveles básicos (no se encuentran accesibles en todas las implementaciones disponibles) son:

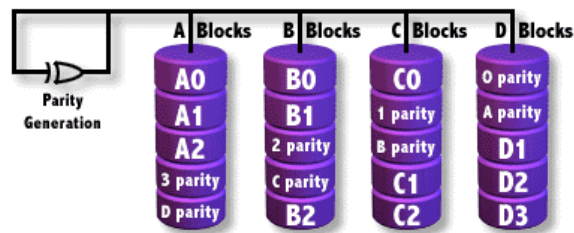
Figura 28: Raid 5



## RAID6:

- Similar a RAID5
- Incluye un esquema de paridad adicional distribuido a través de los diferentes discos
- Se requieren al menos N+2 discos para implementarlo

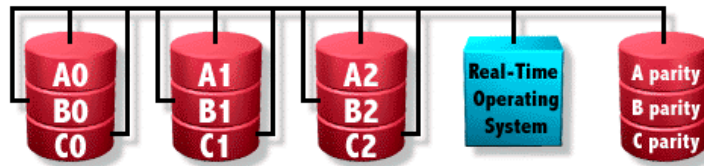
Figura 29: Raid 6



## RAID7:

- Principalmente implementado a nivel hardware
- Aprovecha propiedad de un sistema operativo inmerso (embedded) para tener las ventajas del bus de transferencia
- Permite grandes velocidades en transferencia de datos
- Marca Registrada por Storage Computer Corporation.

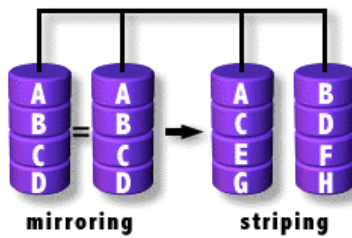
Figura 30: Raid 7



### RAID10

- Es un arreglo de clases (stripes), cada una de ellas compuesta por un arreglo de discos con RAID1
- El costo de esta implementación es demasiado alto, tanto en recursos como en dinero
- Se requieren al menos 4 discos para implementarlo

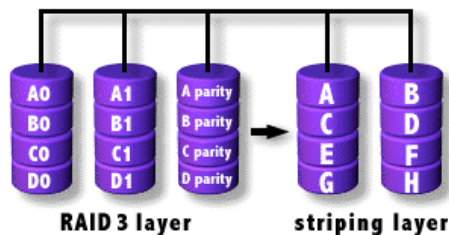
Figura 31: Raid 10



### RAID53:

- Es un arreglo de clases (stripes), cada una de ellas compuesta por un arreglo de discos con RAID3
- Esto ofrece gran velocidad pero el costo es demasiado alto
- Se requieren al menos 5 discos para implementarlo

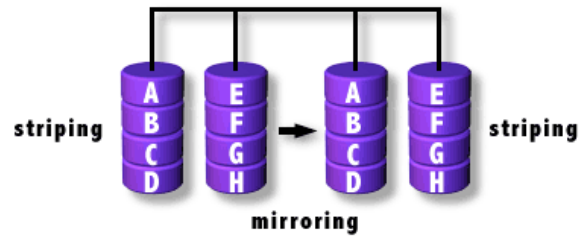
Figura 32: Raid 53



### RAID0+1:

- Es un arreglo de discos espejo
- Cada segmento de discos está configurado como un RAID0

Figura 33: Raid 0+1



#### 1.3.4.4.7.1 Implementaciones de RAID

- Disc Spare (Hot Spare): discos o particiones de respaldo que entran automáticamente si alguno de los discos del arreglo tiene alguna falla
- Dispositivos Hot Plug: máquinas/discos que pueden ser agregados al arreglo en caliente; sin necesidad de reiniciar la máquina.

Un RAID puede estar implementado en Hardware o en Software

- Hardware: Son más rápidos, pero más costosos
  - Sun StorEdge T3 169Tb
- Software:
  - Solaris Volume Manager (Solstice Disk Suite)
  - Linux Kernels
  - Windows Management Console -> Disk Management

#### 1.3.4.4.8 Discos Duros SCSI

Son discos duros de gran capacidad de almacenamiento (desde 5 Gbyte hasta 23 Gbytes). Se presentan bajo tres especificaciones: SCSI Estándar (Standard SCSI), SCSI Rápido (Fast SCSI) y SCSI Ancho-Rápido (Fast-Wide SCSI). Su tiempo medio de acceso puede llegar a 7 mseg y su velocidad de transmisión secuencial de información puede alcanzar teóricamente los 5 Mbps en los discos SCSI Estándares, los 10 Mbps en los discos SCSI Rápidos y los 20 Mbps en los discos SCSI Anchos-Rápidos (SCSI-2).

Un controlador SCSI puede manejar hasta 7 discos duros SCSI (o 7 periféricos SCSI) con conexión tipo margarita (daisy-chain). A diferencia de los discos IDE, pueden trabajar asincrónicamente con relación al microprocesador, lo que los vuelve más rápidos.

En los últimos años han empezado a desarrollarse nuevas tecnologías de discos duros que permiten superar las limitaciones de capacidad de transferencia de información de los discos IDE y SCSI, y que incrementan la capacidad total de almacenamiento. Estas nuevas tecnologías están siendo utilizadas inicialmente en sistemas RISC, minicomputadores y main frames, pero se espera su próxima introducción en servidores tipo PC. La industria de la computación no ha tomado aún partido por ninguna de esas tecnologías, pero las más destacadas son: Ultra-SCSI, Ultra-SCSI-2, Serial Storage Architecture y Fibre-Channel.

#### 1.3.4.4.8.1 Ultra-SCSI y Ultra-SCSI-2

También se las conoce como tecnologías FAST20, siendo consideradas por los expertos como un paso intermedio hacia las interfaces seriales. Ultra-SCSI y Ultra-SCSI-2 representan la última mejora de la tecnología SCSI, que aprovecha las grandes capacidades de los buses locales. Ultra-SCSI y Ultra-SCSI-2 implementan el nuevo protocolo SCSI-3, permitiendo un incremento en la velocidad de transferencia de información hasta 40 MBps para conexiones de 16 bits y hasta 80 Mbps para conexiones de 32 bits. Ultra-SCSI y Ultra-SCSI-2 siguen siendo implementaciones paralelas en las que se ha duplicado la velocidad del reloj del bus, pudiendo coexistir con dispositivos SCSI de tecnologías anteriores, pero por eficiencia es preferible que esos otros dispositivos se conecten a adaptadores independientes.

#### 1.3.4.4.8.2 Serial Storage Architecture

La Arquitectura de Almacenamiento Serial (Serial Storage Architecture / SSA), desarrollada por IBM, es una implementación serial del conjunto de comandos de la tecnología SCSI-2. SSA no ha sido implementada como un bus sino más bien como una serie de pequeños saltos independientes entre 126 dispositivos autoconfigurables (self-configuring) y conectables en caliente (hot-pluggable).

Uno de los atributos más importantes de SSA es su "Reutilización Espacial", que permite la existencia de más tráfico en un bus e incrementa el ancho de banda. La mayor limitación de la tecnología SSA es el ancho de banda máximo de 20 MBps para cualquier componente de la cadena, pero el bus puede soportar hasta 80 MBps. IBM considera a SSA como una solución universal y económica para almacenamiento local.

En un futuro próximo se espera que SSA duplique su velocidad de 20 MBps por nodo a 40 MBps, y de un ancho de banda total del bus de 80 MBps pase a 160 MBps.

#### 1.3.4.4.9 Fibre Channel

Esta tecnología se basa en el trabajo realizado por el Comité de Canales de Fibra (Fiber-Channel Committee) de la IEEE. Fibre Channel (FC) es una interfaz serial que, a pesar de su nombre, no requiere conexiones de fibra óptica (puede utilizar cable de cobre o fibra óptica, indistintamente). Está basada en comandos SCSI-3, que soportan hasta 126 dispositivos autoconfigurables y conectables en caliente, en conexión tipo margarita. Fibre Channel está evolucionando hacia varias topologías que incluyen Punto a Punto (Point-Point), Estructura Conmutable (Fabric), y Cadena Arbitrada (Arbitrated Loop), con diversas velocidades de transferencia, de hasta 100 MBps simultáneamente en cada dirección (hasta 200 MBps en conexiones análogas a full duplex).

- Fibre Channel Punto a Punto establece una conexión entre diferentes dispositivos, proveyendo un ancho de banda total de 100 MBps para cada dispositivo. Sin embargo, solamente un componente puede transmitir o recibir al mismo tiempo sobre una conexión. A pesar de que esto proporciona una mayor velocidad de transmisión que SSA, muchos dispositivos que deseen transmitir o recibir al mismo tiempo deberán esperar que se libere el bus.
- Los dispositivos Fibre Channel de Estructura Conmutable son conocidos como elementos que funcionan de modo similar a switches y ruteadores (sobre redes de dispositivos en lugar de redes de computadores). Pueden consistir de uno o varios elementos que posibilitan que puedan introducirse nuevos componentes o nuevas tecnologías, sin perturbar a los nodos en el extremo exterior de la estructura conmutable ni a la estructura previamente existente. En una estructura conmutable cualquier nodo puede hablar con cualquier otro nodo. La estructura conmutable realiza el ruteo apropiado para proveer un servicio par a par (peer-peer).
- Fibre Channel en Cadena Arbitrada implementa un algoritmo de distribución equitativa, similar a FDDI, para asegurar que todos los dispositivos tengan igualdad de posibilidad de acceso al bus. Sin embargo, se deben configurar apropiadamente los sistemas para evitar congestión.

Los promotores de Fibre Channel Arbitrated Loop (FC-AL) argumentan que esta tecnología tolera mejor la falla de los discos, y debido a sus lazos cercanos con los canales de fibra puede ser utilizada como una interconexión universal tanto para sistemas como para almacenamiento. Empresas como Adaptec, BusLogic, Hewlett-Packard, Q-Logic, Quantum, NCR y Seagate están detrás de su desarrollo. Los partidarios de la tecnología SSA argumentan que los defectos de FC-AL son su alto costo y su alto consumo de energía. IBM apoya ambas interfaces: SSA para almacenamiento y Fiber Channel para interconexión de sistemas.

#### 1.3.4.4.10 Storage area network

Una red SAN se distingue de otros modos de almacenamiento en red por el modo de acceso a bajo nivel. El tipo de tráfico en una SAN es muy similar al de los discos duros como ATA y SCSI. Un servidor preguntaría por el "bloque 6000 del disco 4" en otras modalidades (como SMB o NFS), el servidor solicitaría un determinado archivo como "/home/matias/rocks". La mayoría

de las actuales SAN usan el protocolo SCSI para acceder a los datos de la SAN, aunque no usen interfaces físicas SCSI. Una SAN se puede considerar una extensión de Direct\_Attached\_Storage (DAS). Donde hay en Direct\_Attached\_Storage un enlace punto a punto entre el servidor y su almacenamiento, una SAN permite a varios servidores acceder a varios dispositivos de almacenamiento en una red compartida.

El protocolo SCSI es todavía ampliamente utilizado para conectar los sistemas de cómputo con los dispositivos de almacenamiento como Discos y unidades de cinta, sin embargo, la poca flexibilidad que este provee, así como la distancia que puede existir entre los servidores y los dispositivos de almacenamiento fueron los detonantes para crear un medio de conexión que, permitiera compartir los recursos y a la vez, incrementar las distancias y capacidades de los dispositivos de almacenamiento. Dada la necesidad de compartir recursos, se hizo un primer esfuerzo con los primeros sistemas que compartían el almacenamiento a dos servidores, como el actual HP MSA500G2, pero la corta distancia y la capacidad máxima de 2 servidores, sugirió la necesidad de otra forma de conexión.

#### **Características Principales:**

**Latencia** - Una de las diferencias y principales características de las SAN es que son construidas para minimizar el tiempo de respuesta del medio de transmisión.

**Conectividad** - Permite que múltiples servidores sean conectados al mismo grupo de discos o librerías de cintas, permitiendo que la utilización de los sistemas de almacenamiento y los respaldos sean óptimos.

**Distancia** - Las SAN al ser construidas con fibra óptica heredan los beneficios de ésta, por ejemplo, las SAN pueden tener dispositivos con una separación de hasta 10 Km sin ruteadores.

**Velocidad** - El rendimiento de cualquier sistema de cómputo dependerá de la velocidad de sus subsistemas, es por ello que las SAN han incrementado su velocidad de transferencia de información, desde 1 Gigabit, hasta actualmente 2 y 4 Gigabits por segundo.

**Disponibilidad** - Una de las ventajas de las SAN es que al tener mayor conectividad, permiten que los servidores y dispositivos de almacenamiento se conecten más de una vez a la SAN, de esta forma, se pueden tener rutas redundantes que a su vez incrementarían la tolerancia a fallos.

**Seguridad** - La seguridad en las SAN ha sido desde el principio un factor fundamental, desde su creación se notó la posibilidad de que un sistema accediera a un dispositivo que no le correspondiera o interfiriera con el flujo de información, es por ello que se ha implementado la tecnología de zonificación, la cual consiste en que un grupo de elementos se aíslan del resto para evitar estos problemas. La zonificación puede llevarse a cabo por hardware, software o ambas, siendo capaz de agrupar por puerto o por WWN (World Wide Name), una técnica adicional que se implementa a nivel del dispositivo de almacenamiento que es la Presentación, y consiste en hacer que una LUN (Logical Unit Number) sea accesible sólo por una lista predefinida de servidores o nodos (se implementa con los WWN).

**Componentes** - Los componentes primarios de una SAN son: switches, directores, HBAs, Servidores, Ruteadores, Gateways, Matrices de discos y Librerías de cintas.

**Topología** - Cada topología provee distintas capacidades y beneficios. Las topologías de SAN son: Cascada (cascade) Anillo (ring) Malla (meshed) Núcleo/borde (core/edge) ISL (Inter Switch Link, enlace entre conmutadores). Actualmente las conexiones entre los switches de SAN se hacen mediante puertos tipo "E" y pueden agruparse para formar una troncal (trunk) que permita mayor flujo de información y tolerancia a fallos.

**Arquitectura** - Las SAN Fibre channel actuales funcionan bajo dos arquitecturas básicas, FC-AL (Fibre Channel Arbitrated Loop) y Switched Fabric, ambos esquemas pueden convivir y ampliar las posibilidades de las SAN. La arquitectura FC-AL puede conectar hasta 127 dispositivos, mientras que switched fabric hasta 16 millones teóricamente.



## 1.4 Alta Disponibilidad en sistemas de Información

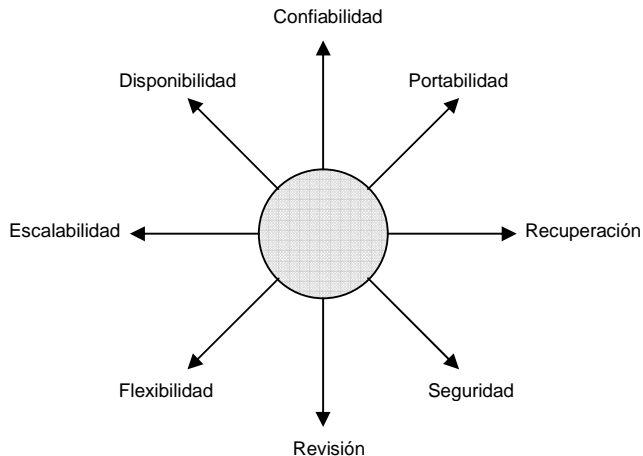
### 1.4.1 Requerimientos de Negocios Modernos y Niveles de Alta Disponibilidad

Los requerimientos de negocios modernos se pueden clasificar por los servicios que una empresa puede ofrecer, como disponibilidad, confiabilidad, etc., en las operaciones y manejo de datos que necesita el usuario cada día.

Los requerimientos se pueden clasificar por capacidades. La siguiente grafica muestra el conjunto de capacidades que un negocio moderno puede ofrecer, sobretodo estas capacidades pueden ser determinadas como requerimientos básicos en todos los sistemas.

Sin embargo con el boom de negocios basados en Internet en estos últimos años, las capacidades como disponibilidad, recuperación, escalabilidad, flexibilidad y seguridad se han vuelto una necesidad. Existen aquellos elementos que no son clasificados bajo las capacidades del sistema, pero continúan siendo requerimientos vitales, por ejemplo el rendimiento del procesador y factores de respuesta del sistema.

Figura 34: Niveles de Alta disponibilidad



#### 1.4.1.1.1 Confiabilidad

El requerimiento de confiabilidad menciona que el sistema debe de ser confiable, es decir, cuando un usuario se conecta al sistema para procesar o ejecutar algo en particular, el sistema debe garantizar la entrega de los resultados esperados o una respuesta razonable de el porque no se entregó.

Este requerimiento aplica para ambos, las herramientas usadas para el desarrollo de la aplicación, y la lógica de la aplicación especificada en el modelo de negocio. Como mencionamos anteriormente existe una gran mayoría de negocios en Internet, y si el sistema no puede ser confiable, es muy probable que el usuario o comprador busque otro sitio que le pueda ofrecer un mejor servicio, y obviamente esto se traduce en pérdidas para la empresa.

#### 1.4.1.1.2 Portabilidad

La portabilidad es un requerimiento importante, es seriamente considerado cuando se establecen nuevos modelos de negocio. El requerimiento de portabilidad asegura que los sistemas de la empresa son fácilmente movibles a otras plataformas, conforme la empresa crezca y utilice un tipo de hardware más grande y eficiente.

Lo que se exige es que el desarrollo de la aplicación, las herramientas de soporte, las capas de productos de terceros e infraestructura sean organizados de tal manera que puedan estar disponibles sin necesidad de modificar algo o si es necesario modificar, que sea algo mínimo. Si la portabilidad del sistema no se considera previamente, es muy probable que se de la necesidad de reprogramar el sistema y esto implica un gran costo para la empresa.

#### **1.4.1.1.3 Recuperación**

La recuperación se refiere a que el sistema se puede recuperar de una caída de sistema, con un mínimo de tiempo de baja. Como nivel básico se tiene el porcentaje de tiempo requerido para resolver la falla en el sistema por ejemplo, una instancia de Oracle.

El requerimiento de recuperación de una base de datos se refiere a la estrategia de respaldos que se realiza en la empresa. Una buena estrategia de respaldos está basada en los requerimientos en tiempos de recuperación, dependiendo del tipo de negocio, el nivel de complejidad que manejan y los acuerdos de niveles de servicio, el tamaño de la base de datos, el intervalo de tiempo que se tiene entre cada respaldo, y el nivel de respaldos que toman al hacer una recuperación de la base de datos.

Seleccionar un buen esquema de recuperación también depende del tiempo de recuperación permitido, o mejor dicho el tiempo medio de recuperación (MTTR Mean Time to Recover)

#### **1.4.1.1.4 Seguridad**

Los días en que las aplicaciones eran usadas por un pequeño grupo de usuarios ya no existe. Bajo el modelo de cliente servidor, las aplicaciones eran usadas por grupos de usuarios identificados que pertenecen a la misma organización.

Los sistemas basados en Internet tienen bases de datos que pueden ser accedidas en cualquier parte del mundo, consecuentemente la seguridad en los datos se ha convertido en un tema muy importante, como un requerimiento de alto nivel. La información es vital para el negocio y debe ser protegida de hackers.

#### **1.4.1.1.5 Revisión**

La revisión de información se refiere a la capacidad de obtener información suficiente con respecto a la creación de la información, por ejemplo: el creado de la información, por qué se creó la información, quién la modificó, cuándo se modificó, etc. Este requerimiento es muy importante ya que existe desde que las computadoras se utilizan en operaciones comerciales.

#### **1.4.1.1.6 Flexibilidad y Mantenimiento**

La flexibilidad es un área extensa con muchos aspectos, las aplicaciones deben ser desarrolladas de tal manera que sean fáciles de administrar para el mantenimiento y deben ser flexibles para cualquier nivel del sistema de la empresa. El mantenimiento se refiere a la administración diaria, la protección del sistema e implementación de cambios funcionales en el sistema. La flexibilidad se refiere al monitoreo, puesta a punto y organización del sistema.

#### **1.4.1.1.7 Escalabilidad**

La escalabilidad normalmente está definida de dos modos, cualquiera como la capacidad de madurar el sistema conforme al crecimiento del negocio, o como la capacidad de la aplicación o el sistema empresarial, para aceptar usuarios adicionales de acuerdo al crecimiento del negocio sin reescribir o rediseñar el sistema.

La escalabilidad puede ser vertical u horizontal (lineal). Cuando se considera el crecimiento del sistema empresarial, la escalabilidad lineal es la mejor opción de configuración cuando es comparada con una escalabilidad vertical.

Mientras que la escalabilidad vertical soporta más usuarios incrementando la capacidad del hardware existente, la escalabilidad lineal soporta más usuarios incrementando el número de sistemas de cómputo (nodos) en la configuración. Desde una perspectiva de sistemas, la solución a este problema es un cluster.

#### 1.4.1.1.8 Disponibilidad

La disponibilidad es otro requerimiento importante que se solicita en los negocios basados en Internet, y a menudo es combinado con la confiabilidad, esta combinación se debe al hecho de que bajo muchas circunstancias la disponibilidad y la confiabilidad son agrupadas como un solo requerimiento.

La disponibilidad se mide por el tiempo que el sistema está funcionando. Normalmente la disponibilidad es calculada por el número de horas que el sistema puede estar funcionando de manera continua al año. Las necesidades del sistema pueden ser libremente definidas como el punto intermedio entre el costo que tiene el sistema disponible y las pérdidas para la empresa por tener el sistema no disponible. La necesidad de tener un sistema disponible tiende a ser del 100%.

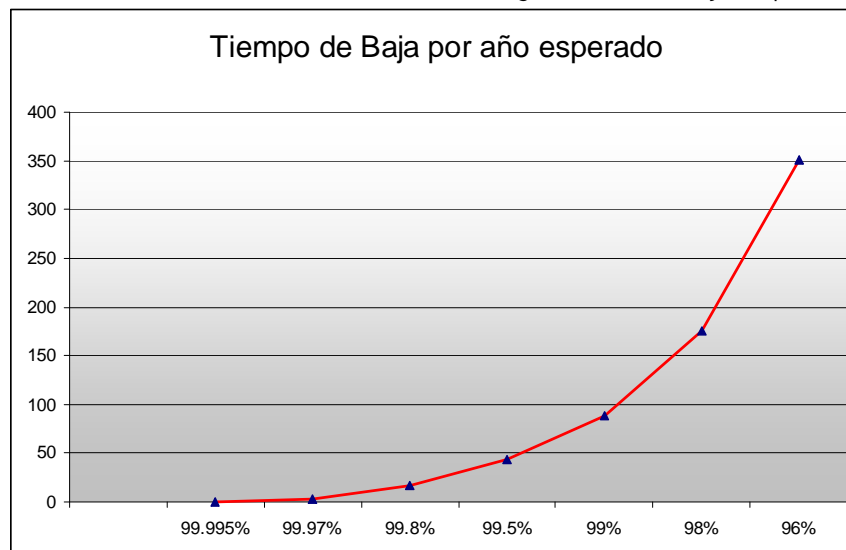
También se mide por la cantidad de tiempo que el sistema está arriba y está disponible para operar. Cuando se menciona la disponibilidad del sistema no solo nos referimos a la aplicación, o la base de datos, el concepto de disponibilidad aplica para todo el sistema de cómputo que se requiere para garantizar la disponibilidad, por ejemplo los servidores, tarjetas de red redundantes, fuentes de poder redundantes, switches redundantes, almacenamiento, etc.

Si la empresa tiene la necesidad de mantener su sistema disponible las 24 horas del día, los 365 días del año es decir, con un nivel de disponibilidad de 99.99%, es necesario trabajar con hardware redundante. Sin embargo si las necesidades de la empresa pueden permitir un tiempo determinado de baja en el sistema, tal vez no sea necesario invertir en un sistema con hardware redundante.

La siguiente tabla muestra los niveles aceptables de bajas en el sistema.

Requerimientos de Disponibilidad	Tiempo de Baja por año esperado
99.995%	0.5 horas
99.97%	2.5 horas
99.8%	17.5 horas
99.5%	43.2 horas o 1.8 días
99%	88.8 horas o 3.7 días
98%	175.2 horas o 7.3 días
96%	350.4 horas o 14.6 días
6*16	Todo el Domingo y 8 horas/noche

Figura 35: Nivele de bajas aceptables en un sistema



#### 1.4.1.1.9 Respuesta

Cuando comparamos las sensibilidades del sistema, los requerimientos de otros negocios pueden ser juzgados como algo menos importante. El tiempo de respuesta que un usuario espera es el tiempo que toma el sistema en responder a un requerimiento hecho por el usuario. Desde la perspectiva del sistema, el tiempo de respuesta es el tiempo de servicio mas el tiempo de espera, es decir el tiempo que toma el sistema para recopilar la información solicitada por el usuario y la cantidad de tiempo experimentada por el sistema debido a otros problemas encontrados mientras se recopila la información, antes de dar una respuesta al usuario.

El tiempo de servicio depende de la capacidad de la aplicación para procesar la información de manera eficaz, que incluye la eficiencia de queries SQL, para obtener la información de la base de datos en forma óptima.

El tiempo de espera es afectado por otros factores, como la intermitencia en la red, la intermitencia en el almacenamiento, etc.

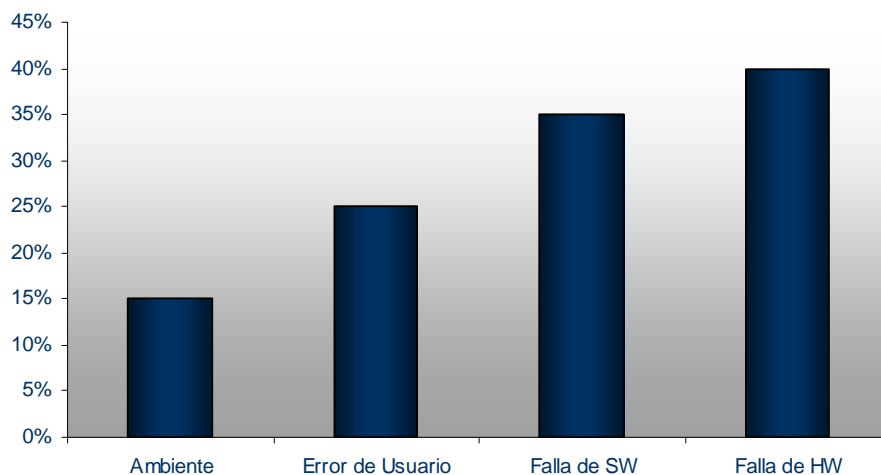
#### 1.4.2 Definición de Alta Disponibilidad

En tecnología informática este término se utiliza para indicar un sistema o un componente que está operacionalmente apto durante un largo período e tiempo.

En la actualidad las organizaciones se miden, en ocasiones, por el servicio que proporcionan a sus clientes, y si en algún momento dicho servicio no está disponible, esa organización llega a perder grandes cantidades de dinero por muy corto que sea el tiempo que dejó de dar ese servicio, incluso puede llegar a perder clientes, ya que éstos buscarían a alguien más que les pueda dar disponibilidad continua en el servicio que requieren.

Existen varias causas de fallas de los sistemas, que provocan interrupción en los servicios que proporciona una organización a sus clientes; las fallas más comunes se ilustran en la siguiente gráfica:

Figura 36: Fallas mas comunes



Para evitar que este tipo de fallas de los sistemas generen pérdidas en la organización, es recomendable contar con la tecnología de información, que soporta a los sistemas del negocio, con cierto grado de alta disponibilidad, dependiendo ésta de la criticidad de las aplicaciones de los negocios.

Alta Disponibilidad se refiere a un sistema o componente que continuamente está en operación por un periodo de tiempo deseable, que está alineado con los objetivos de la organización. La disponibilidad puede ser medida en un porcentaje de operación, por ejemplo los "cinco nueves" (99.999%) de disponibilidad de una solución, la cual involucra Hardware, Software, consultoría, soporte y capacitación,

La alta disponibilidad se puede encontrar desde nivel componente, es decir, un equipo o servidor con componentes redundantes, tales como fuentes de poder, discos en espejo, duplicidad en tarjetas, etc., o llegar a contar con clusters de varios nodos, que pueden estar ubicados localmente en un site o en diferentes localidades.

Los Beneficios que se pueden obtener con una solución de Alta Disponibilidad son:

- Cero caídas no planeadas de los sistemas críticos que soportan los servicios del negocio.
- Porcentaje de disponibilidad requerido por los objetivos de la organización.
- Reducción en costos de administración y soporte de los sistemas de información
- Rápida recuperación de los sistemas críticos en casos de contingencias, para continuar proporcionando los servicios del negocio.

Un cluster de alta disponibilidad es un conjunto de dos o más máquinas, que se caracterizan porque comparten los discos de almacenamiento de datos, y porque están constantemente monitorizándose entre sí. Si se produce un fallo del hardware o de las aplicaciones de alguna de las máquinas del clúster, el software de alta disponibilidad es capaz de reorganizar automáticamente los servicios que han fallado en cualquiera de las otras máquinas del clúster. Y cuando la máquina que ha fallado se recupera, los servicios son nuevamente migrados a la máquina original. Esta capacidad de recuperación automática de servicios nos garantiza la integridad de la información, ya que no hay pérdida de datos, y además evita molestias a los usuarios, que no tienen por qué notar que se ha producido un problema.

No hay que confundir un cluster de alta disponibilidad con un cluster de alto rendimiento. El segundo es una configuración de equipos diseñado para proporcionar capacidades de cálculo mucho mayores que la que proporcionan los equipos individuales (véanse por ejemplo los sistemas de tipo Beowulf), mientras que el primer tipo de cluster está diseñado para garantizar el funcionamiento ininterrumpido de ciertas aplicaciones.

#### **1.4.2.1.1 Cluster**

El término cluster se aplica a los conjuntos o conglomerados de computadoras construidos mediante la utilización de componentes de hardware comunes y que se comportan como si fuesen una única computadora. Juegan hoy en día un papel importante en la solución de problemas de las ciencias, las ingenierías y del comercio moderno.

La tecnología de clusters ha evolucionado en apoyo de actividades que van desde aplicaciones de supercómputo y software de misiones críticas, servidores Web y comercio electrónico, hasta bases de datos de alto rendimiento, entre otros usos.

El cómputo con clusters surge como resultado de la convergencia de varias tendencias actuales que incluyen la disponibilidad de microprocesadores económicos de alto rendimiento y redes de alta velocidad, el desarrollo de herramientas de software para cómputo distribuido de alto rendimiento, así como la creciente necesidad de potencia computacional para aplicaciones que la requieran.

Simplemente, cluster es un grupo de múltiples equipos unidos mediante una red de alta velocidad, de tal forma que el conjunto es visto como una única computadora, más potente que los comunes de escritorio. De un cluster se espera que presente combinaciones de los siguientes servicios:

1. Alto rendimiento (*High Performance*)
2. Alta disponibilidad (*High Availability*)
3. Equilibrio de carga (*Load Balancing*)
4. Escalabilidad (*Scalability*)

La construcción de los servidores en cluster es más fácil y económica debido a su flexibilidad: todos pueden tener la misma configuración de hardware y sistema operativo (cluster homogéneo), diferente rendimiento pero con arquitecturas y sistemas operativos similares (cluster semi-homogéneo), o tener diferente hardware y sistema operativo (cluster heterogéneo).

Para que un cluster funcione como tal, no basta solo con conectar entre sí los servidores, sino que es necesario proveer un sistema de manejo del cluster, el cual se encargue de interactuar con el usuario y los procesos que corren en él para optimizar el funcionamiento.

#### 1.4.2.1.1.1 Componentes de un Cluster

En general, un cluster necesita de varios componentes de software y hardware para poder funcionar. A saber:

- Nodos (servidores)
- Sistemas Operativos
- Conexiones de Red
- Middleware (capa de abstracción entre el usuario y los sistemas operativos)
- Protocolos de Comunicación y servicios.
- Aplicaciones (pueden ser o no ser paralelas)

#### 1.4.2.1.1.2 Nodos

Pueden ser simples computadoras, sistemas multi procesador o estaciones de trabajo (*workstations*).

#### 1.4.2.1.1.3 Sistema Operativo

Debe ser de fácil uso y acceso y permitir además múltiples procesos y usuarios. Ejemplos:

- GNU/Linux
- Unix: Solaris / HP-Ux / Aix
- Windows NT / 2000 / 2003 Server
- Mac OS X
- Cluster OS's especiales
- Otros

#### 1.4.2.1.1.4 Conexiones de Red

Los nodos de un cluster pueden conectarse mediante una simple red Ethernet con placas comunes (adaptadores de red o NICs), o utilizarse tecnologías especiales de alta velocidad como Fast Ethernet, Gigabit Ethernet, Myrinet, Infiniband, SCI, etc.

#### 1.4.2.1.1.5 Middleware

El middleware es un software que generalmente actúa entre el sistema operativo y las aplicaciones con la finalidad de proveer a un cluster lo siguiente:

- Una interfaz única de acceso al sistema, denominada SSI (*Single System Image*), la cual genera la sensación al usuario de que utiliza un único equipo muy potente
- Herramientas para la optimización y mantenimiento del sistema: migración de procesos, *checkpoint-restart* (congelar uno o varios procesos, mudarlos de servidor y continuar su funcionamiento en el nuevo host), balanceo de carga, tolerancia a fallos, etc.

- Escalabilidad: Debe tener la capacidad de detectar automáticamente nuevos servidores conectados al cluster para proceder a su utilización

Existen diversos tipos de middleware, como por ejemplo: MOSIX, OpenMOSIX, Cónдор, OpenSSI, etc.

El middleware recibe los trabajos entrantes al cluster y los redistribuye de manera que el proceso se ejecute más rápido y el sistema no sufra sobrecargas en un servidor. Esto se realiza mediante políticas definidas en el sistema (automáticamente o por un administrador) que le indican dónde y cómo debe distribuir los procesos, por un sistema de monitorización, el cual controla la carga de cada CPU y la cantidad de procesos en él.

El middleware también debe poder migrar procesos entre servidores con distintas finalidades:

- Balancear la carga: Si un servidor está muy cargado de procesos y otro está ocioso, pueden transferirse procesos a este último para liberar de carga al primero y optimizar el funcionamiento
- Mantenimiento de servidores: Si hay procesos corriendo en un servidor que necesita mantenimiento o una actualización, es posible migrar los procesos a otro servidor y proceder a desconectar del cluster al primero
- Priorización de trabajos: En caso de tener varios procesos corriendo en el cluster, siendo uno de ellos de mayor importancia, puede migrarse este proceso a los servidores que posean más o mejores recursos para acelerar su procesamiento

#### 1.4.2.1.1.6 El concepto de cluster

Aunque parezca sencillo de responder no lo es en absoluto. Podría incluirse alguna definición de algún libro, pero aún ni en los expertos en clusters ni en las personas que los implementa existe un acuerdo sobre qué es aquello en lo que trabajan.

#### 1.4.2.1.1.7 Características de un cluster

En este apartado se explican los requisitos que deben cumplir un conjunto de computadoras para ser consideradas cluster, tal y como se conocen hasta el momento.

Para crear un cluster se necesitan al menos dos nodos. Una de las características principales de estas arquitecturas es que exista un medio de comunicación (red) donde los procesos puedan migrar para computarse en diferentes estaciones paralelamente. Un sólo nodo no cumple este requerimiento por su condición de aislamiento para poder compartir información. Las arquitecturas con varios procesadores en placa tampoco son consideradas clusters, bien sean máquinas SMP o mainframes, debido a que el bus de comunicación no suele ser de red, sino interno.

Por esta razón se deduce la primera característica de un cluster:

**Un cluster consta de 2 o más nodos.**

Los nodos necesitan estar conectados para llevar a cabo su misión. Por tanto:

**Los nodos de un cluster están conectados entre sí por al menos un canal de comunicación.**

Por ahora se ha referenciado a las características físicas de un cluster, que son las características sobre las que más consenso hay. Pero existen más problemas sobre las características del centro de control que se ejecuta, pues es el software el que finalmente dotará al conjunto de máquinas de capacidad para migrar procesos, balancear la carga en cada nodo, etc.

**Los clusters necesitan software de control especializado.**

El problema también se plantea por los distintos tipos de clusters, cada uno de ellos requiere un modelado y diseño del software distinto.

Como es obvio las características del cluster son completamente dependientes del software, por lo que no se tratarán las funcionalidades del software sino el modelo general de software que compone un cluster.

Existen varios tipos de software que pueden conformar un cluster:

- Software a nivel de aplicación.

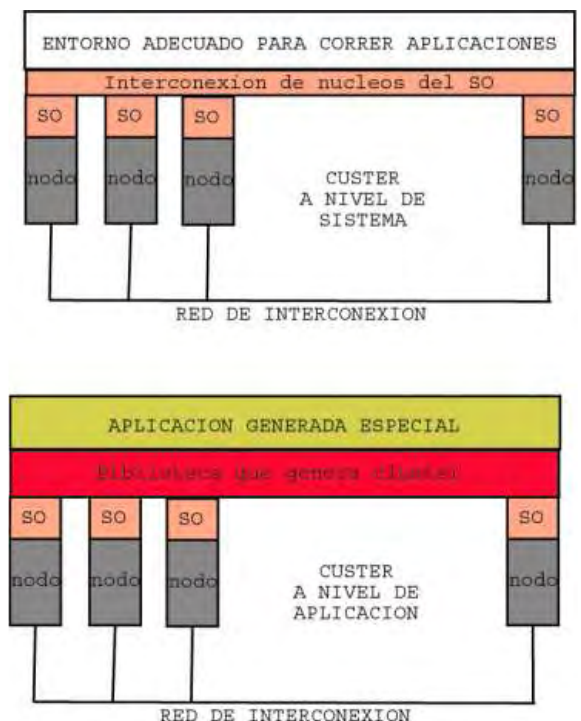
Este tipo de software se sitúa a nivel de aplicación, se utilizan generalmente bibliotecas de carácter general que permiten la abstracción de un nodo a un sistema conjunto, permitiendo crear aplicaciones en un entorno distribuido de la manera más abstracta posible. Este tipo de software suele generar elementos de proceso del tipo rutinas, procesos o tareas, que se ejecutan en cada nodo del cluster y se comunican entre sí a través de la red.

- Software a nivel de sistema.

Este tipo de software se sitúa a nivel de sistema, suele estar implementado como parte del sistema operativo de cada nodo, o ser la totalidad de éste. Es más crítico y complejo, por otro lado suele resolver problemas de carácter más general que los anteriores y su eficiencia, por norma general, es mayor.

A pesar de esta división existen casos en los cuales se hace uso de un conjunto de piezas de software de cada tipo para conformar un sistema cluster completo. Son implementaciones híbridas donde un cluster puede tener instalado a nivel de kernel, parte del sistema mientras que la otra parte estará preparada a nivel de usuario.

Figura 37: Cluster de Sistema y a Nivel de Aplicación





#### 1.4.2.1.1.8 Acoplamiento de un cluster

Dependiendo del tipo de software, el sistema puede estar más o menos acoplado.

Se entiende por acoplamiento del software a la *integración que tengan todos los elementos software que existan en cada nodo*. Gran parte de la integración del sistema la produce la comunicación entre los nodos, y es por esta razón por la que se define el acoplamiento; otra parte es la que implica que tan de crítico es el software y su capacidad de recuperación ante fallos.

Sería importante destacar la diferencia entre los sistemas centralizados o distribuidos. En cualquier caso, el acoplamiento del software es una medida subjetiva basada en la integración de un sistema cluster a nivel general.

Se distingue entre 3 tipos de acoplamiento:

- Acoplamiento fuerte
- Acoplamiento medio
- Acoplamiento débil

##### 1.4.2.1.1.8.1 Acoplamiento fuerte

El software que entra en este grupo es un software cuyos elementos se interrelacionan mucho unos con otros y posibilitan la mayoría de las funcionalidades del cluster de manera altamente cooperativa. El caso de acoplamiento más fuerte que se puede dar es el que solamente haya una imagen del kernel del sistema operativo distribuida entre un conjunto de nodos que la compartirán. Por supuesto algo fundamental es poder acceder a todas las partes de este sistema operativo, estrechamente relacionadas entre sí y distribuidas entre los nodos.

Este caso es el que se considera como el más acoplado, de hecho no está catalogado como cluster, sino como sistema operativo distribuido.

Otro ejemplo son los cluster SSI, en estos clusters todos los nodos ven una misma imagen del sistema, pero todos los nodos tienen su propio sistema operativo, aunque estos sistemas están estrechamente relacionados para dar la sensación a las aplicaciones que todos los nodos son idénticos y se acceda de una manera homogénea a los recursos del sistema total.

Si arranca o ejecuta una aplicación, ésta verá un sistema homogéneo, por lo tanto los kernels tienen que conocer los recursos de otros nodos para presentarle al sistema local los recursos que encontraría si estuviera en otro nodo. Por supuesto se necesita un sistema de nombres único, manejado de manera distribuida o centralizada así como un mapeo de los recursos físicos a este sistema de nombres.

##### 1.4.2.1.1.8.2 Acoplamiento medio

A este grupo pertenece un software que no necesita un conocimiento tan exhaustivo de todos los recursos de otros nodos, pero que sigue usando el software de otros nodos para aplicaciones de muy bajo nivel. Como ejemplos: OpenMosix y Linux-HA.

Un cluster openMosix necesita que todos los kernels sean de la misma versión. Por otro lado no está tan acoplado como el caso anterior, no necesita un sistema de nombres común en todos los nodos, y su capacidad de dividir los procesos en una parte local y otra remota consigue que por un lado se necesite el software del otro nodo donde está la parte del archivo que falta en el nodo local y por otro que no se necesite un SSI para hacer otras tareas.

#### 1.4.2.1.1.8.3 Acoplamiento débil

Generalmente se basan en aplicaciones construidas por bibliotecas preparadas para aplicaciones distribuidas. Es el caso de por ejemplo PVM, MPI o CORBA. Éstos por sí mismos no funcionan en modo alguno con las características que antes se han descrito (como Beowulf) y hay que dotarles de una estructura superior que utilice las capacidades del cluster para que éste funcione.

#### 1.4.2.1.1.9 Esquema y otras características

Las características básicas de un cluster de carácter general podrían resumirse en el siguiente esquema:

- Un cluster consta de 2 o más nodos conectados entre sí por un canal de comunicación funcional.
- En cada nodo es imprescindible un elemento de proceso, memoria y un interfaz para comunicarse con la red del cluster.
- Los clusters necesitan software especializado. Este software y las máquinas conforman el cluster.
- Todos los elementos del cluster trabajan para cumplir una funcionalidad conjunta, sea ésta la que sea. Es la funcionalidad la que caracteriza el sistema.

En general la catalogación de los clusters se hace en base a cuatro factores de diseño ortogonales entre sí:

- Acoplamiento
- Control
- Homogeneidad
- Seguridad

El parámetro de control implica el modelo de gestión que propone el cluster. Este modelo de gestión hace referencia a la manera de configurarlo y es dependiente del modelo de conexión o colaboración que surgen entre los nodos. Puede ser de dos tipos:

- **Control centralizado:** Se hace uso de un nodo maestro desde el cual se puede configurar el comportamiento de todo el sistema. Este nodo es un punto crítico del sistema aunque es una ventaja para una mejor administración del cluster.
- **Control descentralizado:** En un modelo distribuido donde cada nodo debe administrarse en forma independiente, también puede ser administrado mediante aplicaciones de más alto nivel de manera centralizada, pero la mayoría de la gestión que hace el nodo local es leer archivos de configuración de su propio nodo.

#### 1.4.2.1.1.10 Homogeneidad de un cluster

Es homogeneidad de los equipos y recursos que conforman a éste. Los clusters heterogéneos son más difíciles de conseguir ya que se necesitan notaciones abstractas de transferencias e interfaces especiales entre los nodos para que éstas se entiendan, por otro lado los clusters homogéneos obtienen más beneficios de estos sistemas y pueden ser implementados directamente a nivel de sistema.

- **Homogéneos:** Formados por equipos de la misma arquitectura. Todos los nodos tienen una arquitectura y recursos similares, de manera que no existen muchas diferencias entre cada nodo.

- **heterogéneos:** Formados por nodos con distinciones que pueden estar en los siguientes puntos.
  - Tiempos de acceso distintos
  - Arquitectura distinta
  - Sistema operativo distinto
  - Rendimiento de los procesadores o recursos sobre una misma arquitectura distintos

El uso de arquitecturas distintas, o distintos sistemas operativos, impone que exista una biblioteca que tenga la función de interfaz, e incluso una sintaxis de notación abstracta del tipo ASN.1 o XDR en la capa de presentación que utilice la interfaz de comunicación de nuestro sistema distribuido o cluster. Esto hace que este tipo de clusters se consideren implementados a nivel de aplicación.

Existen otros muchos factores de diseño que limitan el comportamiento y modelado de un cluster. La imposibilidad de llegar a clusters que paralelicen cualquier proceso se basa en que la mayoría de las aplicaciones hacen uso, en mayor o menor medida, de algoritmos secuenciales no paralelizables.

#### **1.4.2.1.2 Clasificación según el servicio prioritario**

Generalmente el diseño de un cluster se realiza para solucionar problemas como:

- Mejora de rendimiento
- Abaratamiento de costos
- Distribución de factores de riesgo del sistema
- Escalabilidad

El punto inicial ha sido explicado anteriormente: el costo para incrementar la capacidad de un equipo no suele ser habitualmente a costa de pagar el doble, sino unas cuantas veces más. El modelo de los clusters permite que la mejora de rendimiento sea evidente respecto a grandes mainframes a un precio realmente accesible, lo que explica a su vez el segundo punto, acerca del costo de los clusters, que permite relaciones rendimiento precio que se acercan a un margen lineal dependiendo del cluster implementado.

Por otro lado está la distribución de riesgos. La mayoría de los usuarios tienen sus servicios, aplicaciones, bases de datos o recursos en un solo equipo, o dependientes de un solo equipo. Otro paso más adelante es colocar las bases de datos replicadas sobre sistemas de archivos distribuidos de manera que estos no se pierdan por que los datos son un recurso importante.

Actualmente el mercado de la informática exige no sólo que los datos sean críticos sino que los servicios estén activos constantemente. Esto exige medios y técnicas que permitan que el tiempo en el que una máquina esté inactiva sea el menor posible. La distribución de factores de riesgo a lo largo de un cluster (o la distribución de funcionalidades en casos más generales) permite de una forma única obtener la funcionalidad de una manera más confiable: si una máquina cae otras podrán dar el servicio.

Por último está el factor de escalabilidad. Cuanto más escalable es un sistema menos costará y mejorará el rendimiento, lo cual ayudará a expandir la tecnología con mayor facilidad.

#### **1.4.2.1.3 Alto rendimiento (HP, high performance)**

Los clusters de alto rendimiento han sido creados para compartir el recurso más valioso de una computadora, es decir, el tiempo de proceso. Cualquier operación que necesite altos tiempos de CPU puede ser utilizada en un cluster de alto rendimiento, siempre y cuando se encuentre un algoritmo que sea paralelizable.

Existen clusters que pueden ser denominados de alto rendimiento tanto a nivel de sistema como a nivel de aplicación. A nivel de sistema tenemos openMosix, mientras que a nivel de

aplicación se encuentran otros como MPI, PVM, Beowulf así como otros muchos. En cualquier caso, estos clusters hacen uso de la capacidad de procesamiento que pueden tener varias máquinas.

#### **1.4.2.1.4 Alta disponibilidad (HA, high availability)**

Los clusters de alta disponibilidad son ortogonales como los de alto rendimiento en relación a su funcionalidad. Los clusters de alta disponibilidad pretenden dar servicios 7/24 de cualquier tipo, son clusters donde la principal funcionalidad es estar controlando y actuando para que un servicio o varios se encuentren activos durante el máximo periodo de tiempo posible. En estos casos se puede comprobar como la monitorización de otros es parte de la colaboración entre los nodos del cluster.

#### **1.4.2.1.5 Alta confiabilidad (HR, high reliability)**

Por ultimo, están los clusters de alta confiabilidad. Estos clusters tratan de aportar la máxima confiabilidad en un entorno, en la cual se necesite saber que el sistema se va a comportar de una manera determinada. Puede tratarse por ejemplo de sistemas de respuesta a tiempo real.

#### **1.4.2.1.6 Técnicas que utilizan**

Las técnicas utilizadas dependen del nivel en que trabaje el cluster.

Los clusters implementados a nivel de aplicación no suelen efectuar balanceo de carga. Suelen basar todo su funcionamiento en una política de localización que sitúa las tareas en los diferentes nodos del cluster, comunicándolas mediante las librerías abstractas. Resuelven problemas de cualquier tipo los cuales han sido mencionados en el apartado anterior, pero se deben diseñar y codificar aplicaciones propias para cada tipo.

Por otro lado están los sistemas de alto rendimiento implementados a nivel de sistema. Estos clusters se basan en comunicación y colaboración de los nodos a nivel de sistema operativo, lo que generalmente indica que son clusters de nodos de la misma arquitectura, con ventajas en lo que se refiere al factor de acoplamiento y que basan su funcionamiento compartiendo recursos a cualquier nivel, así como balanceo de la carga de manera dinámica, funciones de planificación especiales y otros tantos factores que componen el sistema.

#### **1.4.2.1.7 Clusters HA: alta disponibilidad**

Son los más solicitados por las empresas ya que están destinados a mejorar los servicios de alto costo que éstas ofrecen a los clientes en las redes a las que pertenecen, tanto en redes locales como en redes como Internet. En este apartado se darán las claves que explican tanto el diseño de estos clusters así como algún factor de implementación.

##### **1.4.2.1.7.1 La misión**

Los clusters de alta disponibilidad han sido diseñados para garantizar su utilización constante y brindar la máxima disponibilidad sobre los servicios que presenta. Este tipo de clusters son la competencia que abarata los sistemas redundantes, de manera que ofrecen una serie de servicios durante el mayor tiempo posible. Para poder dar estos servicios los clusters de este tipo se implementan en base a tres factores.

- Fiabilidad
- Disponibilidad
- Dotación de servicio

Mediante estos tres tipos de actuaciones y los mecanismos que lo implementan se asegura que un servicio esté el máximo tiempo disponible y que éste funcione de una manera fiable. Respecto al tercer punto, la dotación de servicios significa que los clusters pueden en un momento dado proveer servicios a clientes externos.

#### 1.4.2.1.7.2 Problemas que solucionan

La mayoría de estos problemas están ligados a la necesidad de dar servicio continuo de cualquier tipo a una serie de clientes de manera ininterrumpida. En una construcción real se suelen producir fallos inesperados en las máquinas, estos fallos provocan la aparición de dos eventos en el tiempo: el tiempo en el que el servicio está inactivo y el tiempo de reparación del problema.

Entre los problemas que solucionan se encuentran:

- Sistemas de información redundante
- Sistemas tolerantes a fallos
- Balanceo de carga entre varios servidores
- Balanceo de conexiones entre varios servidores

En general todos estos problemas se ligan en dos fuentes de necesidad de las empresas u organizaciones.

- Tener un servicio disponible
- Ahorrar económicamente todo lo que sea posible

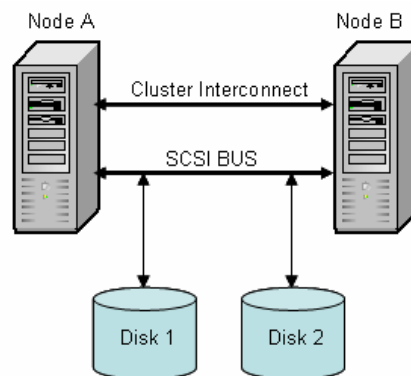
El servicio puede ser diverso. Desde un sistema de archivos distribuidos de carácter módico, hasta grandes clusters de balanceo de carga y conexiones para los grandes portales de Internet. Cualquier funcionalidad requerida en un entorno de red puede ser colocada en un cluster e implementar mecanismos para hacer que ésta obtenga la mayor disponibilidad posible.

#### 1.4.2.1.8 Como funciona un Cluster

Un cluster funciona principalmente a través de un mecanismo llamado Polling (Interrogación), es una actividad similar al ping, donde un mensaje es enviado a una destino, el mensaje es analizado y si no hay problema en el mensaje, el envío del mensaje resulta correcto; de otro modo se entiende que hay un problema en la comunicación, de manera que los servidores se envían mensajes entre ellos, para determinar si están funcionando, si no es así se efectúa un fail over, El polling, depende de la interconexión de los servidores a través de una red, y los dispositivos de conexión.

En la siguiente imagen el esquema de conexión entre dos servidores en cluster, la interconexión del cluster por medio de una red, es usada por el polling para determinar que nodo esta disponible mientras el sistema determina si los nodos están disponibles, por medio del SCSI BUS, basado en un chequeo inicial y la interconexión del cluster.

Figura 38: Esquema de conexión en cluster



El dispositivo de interconexión interroga al otro nodo del cluster para determinar si el otro nodo esta disponible. Si el nodo o servidor no responde en un tiempo de respuesta previamente determinado, el servidor que comenzó la interrogación intenta buscar el otro nodo por medio de la red.

La interconexión por medio de una red LAN se le denomina Heartbeat, en la configuración del heartbeat se define el tiempo de respuesta entre los nodos, si un nodo no responde en ese periodo de tiempo se asume que el nodo no esta disponible y se genera un fail over.

Si el nodo que interroga determina a través del heartbeat que el segundo nodo no está disponible, este nodo ocupara todos los recursos de disco asignados al otro nodo, de manera que siempre estén disponibles en la red.

Cuando el nodo que no funcionó, vuelve a funcionar, el servidor sustituto le entrega nuevamente los recursos de disco. Este es un proceso que la mayoría de los clusters utilizan.

Las condiciones para que se genere un fail over son las siguientes:

1. La interrogación entre nodos por medio del SCSI BUS falla
2. La interrogación entre nodos por medio de la red falla
3. El nodo que comienza el interrogatorio, toma el control de manera satisfactoria de los recursos de disco que tenia el otro nodo.

#### **1.4.2.1.9 Cluster Fail Over**

Es un modo de operación de backup en el cual las funciones de un componente del sistema son asumidas por un segundo componente del sistema cuando el primero no se encuentra disponible debido a un fallo o un tiempo de parada preestablecido. Es usado para hacer a los sistemas más tolerantes a fallos, y de esta forma hacer el sistema permanentemente disponible.

#### **1.4.2.2 Alta Disponibilidad en Servidor de Aplicaciones Oracle**

En capítulos anteriores se mencionaron los conceptos de alta disponibilidad y la necesidad de utilizarla, Oracle Application Server 10g, ofrece soluciones para este tema.

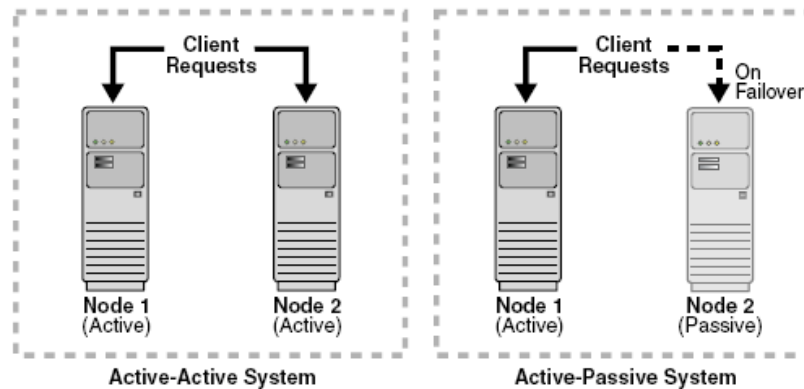
Oracle ofrece Alta disponibilidad y Recuperación de desastres como soluciones para obtener una máxima protección en caso de fallas. Estas soluciones se pueden organizar de la siguiente forma:

1. Alta disponibilidad, con la cual se pretende tener siempre disponibles los datos y las aplicaciones de la empresa, una solución de alta disponibilidad protege de fallas en los procesos, en los nodos, almacenamiento o errores humanos.
2. Recuperación de desastres, el cual se utiliza de una manera geográficamente distribuida, para protegerse de desastres naturales.

##### **1.4.2.2.1 Alta disponibilidad**

Para resolver el problema de la alta disponibilidad, existen diferentes tecnologías y buenas prácticas para lograrlo, pero el concepto más importante es la redundancia. La alta disponibilidad se conforma por sistemas y componentes redundantes. La alta disponibilidad se puede categorizar en dos tipos de redundancia, Servidores Activo - Activo u Servidores Activo - Pasivo, la imagen siguiente muestra un ejemplo.

Figura 39: Tipos de Redundancia



En una solución Activo - Activo se implementan dos o más instancias activas y pueden ser utilizadas para mejorar la escalabilidad del cluster tanto como el hardware lo permita. Todas las instancias manejan las peticiones al mismo tiempo.

Una solución Activo - Pasivo se implementa de manera tal que una sola instancia maneja todas las peticiones y una segunda instancia se encuentra en espera a que se de alguna falla. Un mecanismo de Heartbeat es configurado entre las dos instancias. Este mecanismo lo provee y lo administra el software del cluster que incluye el sistema operativo, lo cual dependerá del proveedor. Normalmente el proveedor genera procesos agentes que están disponibles para monitorear automáticamente los nodos, de manera que cuando la instancia activa falla, un agente da de baja completamente la instancia que falló y activa la instancia pasiva, para que reciba todas las peticiones. Este tipo de configuración también es llamada Cold Failover Cluster.

Oracle Application Server se puede configurar con cualquiera de las dos topologías. Desde el punto de acceso al Servidor de Aplicaciones hasta el almacenamiento se puede configurar redundancia. Algunas de las cualidades más importantes son:

#### 1.4.2.2.2 Detección de procesos muertos y reinicio automático

Los procesos pueden morir inesperadamente debido a problemas de software o una mala configuración. Un proceso de monitoreo adecuado y reinicio del sistema deberá monitorear los procesos del sistema constantemente, y reiniciar los procesos que tengan algún problema. El monitoreo de estos procesos se lleva a cabo por medio del OPMN (Oracle Process Manager and Notifications).

#### 1.4.2.2.3 Clustering

Los componentes clusterizables de un sistema, habilitan los componentes para ser vistos funcionalmente como una entidad única desde la perspectiva de un cliente o procesos en tiempo de ejecución. Un Cluster es un grupo de procesos ejecutados en un nodo independiente que sólo son compartidos a través de la misma carga de trabajo.

#### 1.4.2.2.4 Administración de Configuraciones

Los componentes de un cluster requieren tener configuraciones comunes, una configuración correcta asegura que todos los componentes tengan la misma configuración para recibir las peticiones entrantes, habilita los componentes para sincronizar las configuraciones y provee la administración de la alta disponibilidad para evitar el menor tiempo de baja.

En el servidor de aplicaciones Oracle, existe un proceso llamado DCM (Distributed Configuration Management) para sincronizar las configuraciones en todos los nodos del cluster.

#### 1.4.2.2.5 Balanceo de Cargas y Failover

Cuando múltiples instancias que tienen componentes idénticos están disponibles, las peticiones de los clientes pueden ser balanceadas entre los componentes de la instancia para asegurar que las instancias tengan exactamente la misma carga. Con un balanceo de cargas las instancias son redundantes, y si alguna de las instancias falla, las peticiones que lleguen a la instancia que falló, pueden dirigirse a las instancias que si están funcionando.

#### 1.4.2.2.6 Respaldos y Recuperación

Los errores de los usuarios pueden causar un mal funcionamiento del sistema. En ciertas circunstancias un componente del sistema falla y no es reparable. La generación de respaldos con ciertos intervalos de tiempo puede ayudar a una recuperación de fallas irreparables.

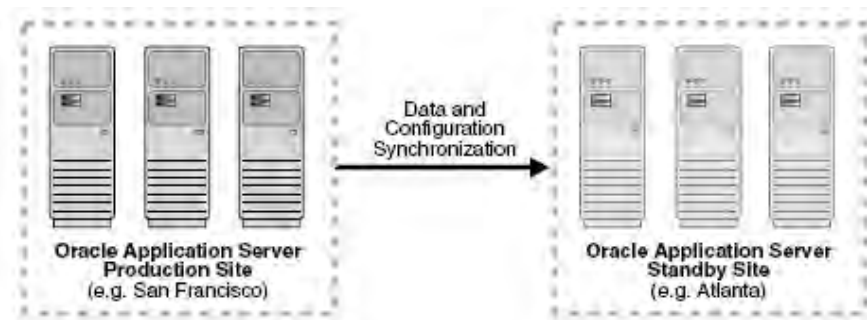
#### 1.4.2.2.7 Recuperación de desastres

Una de las cosas mas importantes para una empresa son los datos que maneja, para ello Oracle provee una solución de recuperación ante algún desastre natural o simplemente una pérdida total de servidores con información. Esta solución se basa en tener un esquema de replicamiento en línea de manera remota, es decir tener un grupo de servidores alternos en modo de espera, que se actualicen los datos cada vez que se haga una nueva transacción en los servidores principales.

La configuración de los servidores alternos deberá ser idéntica a la de los servidores principales. Así mismo, entre ambos sistemas la actualización debe ser constante.

Normalmente los servidores alternos se encuentran alejados de los servidores principales, frecuentemente en otro estado del país, dependiendo de las necesidades de la empresa.

Figura 40: Replicación a sites alternos



#### 1.4.2.3 Alta Disponibilidad en Bases de Datos

En los capítulos anteriores se mencionó el concepto de bases de datos, bases de datos Oracle, alta disponibilidad y la redundancia. En este capítulo veremos una solución de alta disponibilidad de bases de datos con Oracle.

#### 1.4.2.4 Real Application Cluster (RAC)

RAC es una base de datos redundante que utiliza tecnología de clusters y que requiere dos o más nodos, el hardware de los nodos debe ser capaz de trabajar de manera conjunta bajo el concepto de cluster. El hardware del cluster tiene sus propias herramientas de administración, normalmente las provee el distribuidor, el sistema operativo es responsable de proveer el acceso a los medios de almacenamiento.

Oracle RAC permite que múltiples computadoras ejecuten el software de SGBD de Oracle simultáneamente mientras acceden a una base de datos individual. Esto se llama una base de datos en cluster (clustered).



En una base de datos de Oracle no-RAC, una base de datos individual es accedida por una instancia individual. La base de datos se considera la colección de archivos de datos, archivos de control, y archivos redo log localizados en disco. La instancia se considera la colección de procesos del sistema operativo y memoria relacionada de Oracle que están ejecutándose en la computadora.

En Oracle RAC, dos o más computadoras (cada una con una instancia) acceden concurrentemente a una base de datos individual. Esto permite que una aplicación o usuario se conecte a alguno de los computadores y tenga acceso a los mismos datos.

Algunas de las características que tiene ORACLE RAC son las siguientes:

#### **1.4.2.4.1 Cache Fusion**

Dado que todos los computadoras/instancias acceden a los mismos datos, el software de Oracle debe garantizar que los datos cambien en computadoras diferentes de forma coordinada y que cuando una computadora consulta datos recibe la versión actual incluso si los datos fueron modificados recientemente por otra computadora.

Esta funcionalidad de Oracle RAC se llama Cache Fusion. Cache Fusion es el nombre dado a la habilidad de Oracle RAC para tratar los caches de datos en memoria de cada computadora en una cache individual global. Cache Fusion esencialmente funde las caches separadas en una cache global.

Cache Fusion transfiere los bloques de datos (la unidad de transferencia más pequeña en la base de datos) usando la red de interconexión de alta velocidad de la infraestructura. Antes de la fusión de cache, el disco se usa como un medio de transferencia de datos que tiene desventajas evidentes.

Dado que Oracle RAC permite a varias computadoras acceder a una base de datos individual, puede ser usado para dirigir varias áreas de gestión de base de datos. Estas áreas incluyen: Alta disponibilidad, Escalabilidad, Crecimiento Incremental, y Consolidación de Base de datos.

#### **1.4.2.4.2 Sequence Generators**

Este componente permite que todos los objetos de las bases de datos sean accesibles desde una o más instancias al mismo tiempo.

#### **1.4.2.4.3 System Change Number (SCN)**

Este componente es generado en una instancia y se comunica a los otros nodos por medio de la interconexión de red, para proveer al usuario un solo estado de la transacción.

#### **1.4.2.4.4 FailOver**

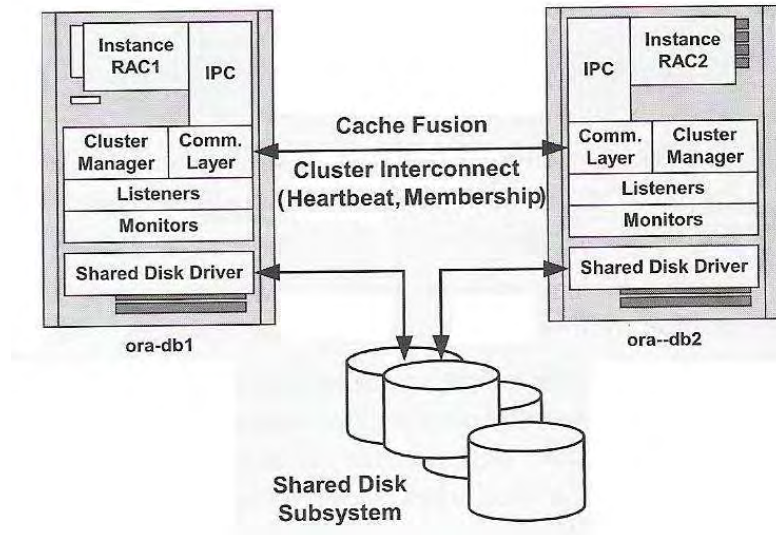
Este componente consiste en dos o más nodos que participan en una configuración colectiva. En las bases de datos clusterizadas permiten hacer un failover permitiendo una reconexión a la base de datos en caso de que una de las instancias falle.

#### **1.4.2.4.5 Scalability**

Este componente permite agregar nodos al cluster con la finalidad de proveer más poder de procesamiento al usuario.

La siguiente imagen muestra una configuración de cluster y sus diferentes componentes.

Figura 41: Configuración de cluster



- Cluster Manager (CM)
- Shared Disk driver (disk group manager)
- Communication layer
- Interprocess communication protocol (IPC)
- Listeners
- Monitors

Cluster manager es parte del cluster de sistema operativo y es el responsable de proveer integridad al cluster. Una interconexión de alta velocidad es usada para la interconexión entre los nodos en el cluster. El CM utiliza la interconexión para generar mensajes que se envían a los otros nodos. La función del heartbeat es determinar que nodos son miembros del cluster, es decir el heartbeat hace que los nodos entiendan y conozcan cuantos miembros están disponibles en el cluster.

La capa de comunicación, administra la comunicación entre los nodos. Es responsable de garantizar la comunicación entre los nodos para que el administrador del cluster pueda enviar mensajes por medio del heartbeat.

El protocolo de intercomunicación entre procesos es el responsable del empacamiento de los mensajes de oracle y de enviarlos de la capa de comunicación al acceso de interconexión.

Los monitores de procesos revisan que todos los procesos para que el heartbeat realice su labor estén funcionando correctamente.

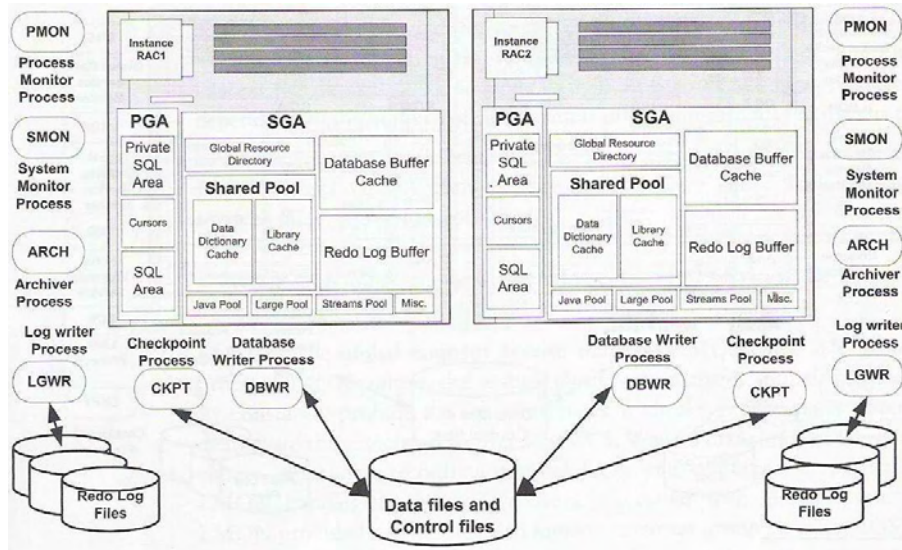
El monitor de Listener verifica que el proceso de listeners y la instancia estén funcionando correctamente.

#### 1.4.2.4.6 Procesos de RAC

En los capítulos anteriores se mencionan los procesos que necesita una instancia de Base de Datos Oracle, RAC utiliza prácticamente los mismos procesos de una sola instancia, pero RAC utiliza más procesos para que puedan comunicarse todas las instancias instaladas en el Cluster. Cada nodo instalado en el cluster tiene sus propios procesos, y memoria SGA asignada pero todas las instancias apuntan a los mismos datos, es decir, una base de datos esta abierta por varias instancias.

La siguiente imagen muestra el esquema de los procesos y memoria SGA que tiene cada instancia en un RAC de dos nodos.

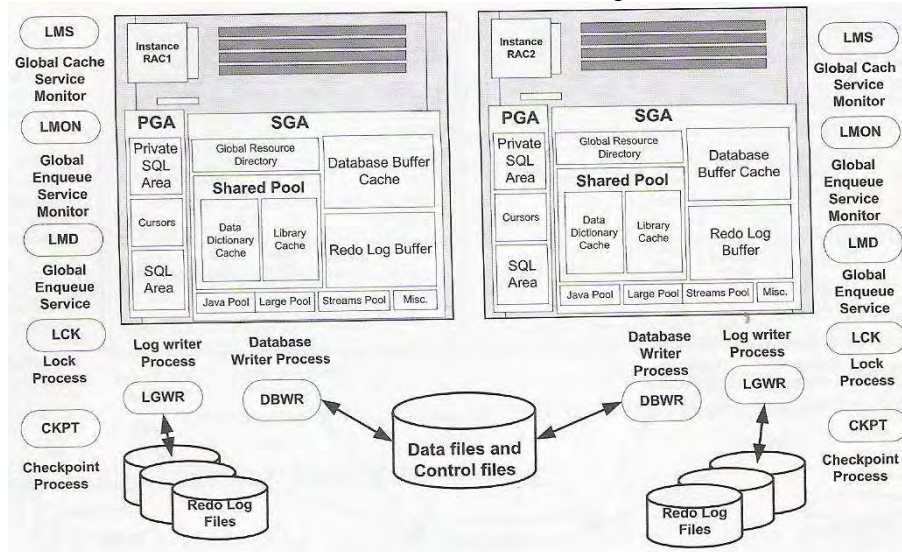
Figura 42: Esquema de procesos y SGA en RAC



- RAC es una configuración con múltiples instancias de Oracle corriendo en varios nodos.
- Múltiples instancias de Oracle comparten una sola base de datos, así como data files y control files.
- Múltiples instancias se alojan en diferentes nodos y se comunican con otros nodos por medio de la interconexión del cluster
- Se pueden agregar y quitar instancias al cluster de manera dinámica, definiendo el número máximo de instancias que utiliza el cluster con el parámetro: MAX\_INSTANCES que se define el archivo de parámetros.

La siguiente imagen muestra los procesos de RAC que utilizan las instancias en cada nodo.

Figura 43: Procesos en una instancia de RAC



- Cada instancia que participa en el cluster tiene sus propios archivos log, segmentos de rollback y tablespaces de Undo.
- Todas las instancias que pertenecen al cluster pueden ejecutar transacciones en la misma base de datos
- Todas las instancias que pertenecen al cluster se comunican por medio de interconexión usando la nueva tecnología llamada Cache Fusion.

#### **1.4.2.4.6.1 Global Cache Service Monitor (LMS)**

Este proceso copia los bloques directamente del buffer cache de la instancia que los está utilizando y los envía como una copia consistente a los procesos en primer plano para que la instancia que necesita esos bloques los pueda leer desde buffer cache.

#### **1.4.2.4.6.2 Global Enqueue Service monitor (LMON)**

Este proceso monitorea todo el cluster para administrar los recursos generales. Lo hace probando constantemente las instancias que integran el cluster, revisa y administra las instancias que no responden, y lo asocia con el Global Cache Service (GCS). Cuando un nodo entra o deja el cluster, este proceso maneja la configuración de los procesos bloqueados por otra instancia. En particular este proceso maneja parte de la recuperación asociada a los recursos generales. LMON provee servicios que son también conocidos como Cluster Group Service (CGS).

#### **1.4.2.4.6.3 Global Enqueue Service Daemon (LMD)**

El proceso es un agente que maneja los requerimientos de los recursos de control para acceder a los bloques y los encolamientos generales. Administra los bloqueos que son requeridos por el servicio GCS y los envía a un servicio que los administra llamado LMSn. El proceso LMD también administra los bloqueos globales (global deadlock) y los requerimientos de otras instancias.

#### **1.4.2.4.6.4 Global Service Daemon (GSD)**

Este proceso es un componente que recibe los requerimientos del control SRVCTL, para ejecutar las tareas administrativas como el inicio y parado de la base de datos. Este comando se ejecuta localmente en cada nodo.

#### **1.4.2.4.6.5 Lock Process (LCK)**

Este proceso administra los requerimientos que se hacen hacia los recursos de la instancia así como las llamadas a los recursos compartidos. Este proceso construye una lista de elementos inválidos o bloqueados, y valida los elementos bloqueados durante la recuperación. Esto es porque el proceso LMS maneja la función primaria de la administración de bloqueos. Un solo proceso de LCK existe en cada instancia.

#### **1.4.2.4.6.6 Diagnosability daemon (DIAG)**

Este proceso monitorea la integridad de la instancia y captura un diagnóstico acerca de los procesos que han fallado con la instancia, la operación de este demonio es automática y actualiza el archivo de alertas en donde guarda sus actividades.

#### **1.4.2.4.6.7 Cache Fusion**

La fusion de cache es la tecnología que utiliza la interconexión entre procesos a una alta velocidad, para poder hacer una transferencia de bloques de datos entre las instancias de bases de datos en el cluster. Esta tecnología para transferir datos a otros nodos a través de la interconexión se vuelve una opción válida ya que las interconexiones de banda ancha han evolucionado, así como los mecanismos de transporte.

La arquitectura del Cache Fusion es un tema novedoso en la industria, ya que maneja la memoria RAM de cada nodo del cluster como una sola, y a su vez una un gran espacio para SGA.

#### 1.4.2.4.6.8 Global Resource Directory (GRD)

El recurso directorio global contiene información acerca de los estatus de los recursos compartidos. Es mantenido por los procesos GCS y GES, los cuales guardan información de los recursos y encolamientos de éstos. GRD se almacena en memoria y es utilizado por los procesos GCS y GES para administrar la actividad general de los recursos. Es distribuido a través del cluster a todos los nodos. Cada nodo participa en la administración de los recursos generales y maneja una porción del GRD.

#### 1.4.2.4.6.9 Lock Structure

Las bases de datos Oracle utilizan un sistema de bloqueo de objetos, para evitar que dos usuarios utilicen el mismo objeto; en el caso de RAC, se mantiene el mismo concepto pero a nivel de instancias, ya que al tener más de una instancia, los bloques de datos pueden ser accedidos por más de una instancia, la forma de manejar los bloques es asignando una bandera binaria.

Si una primera instancia accede a un bloque de datos y la segunda instancia necesita acceder al mismo, se generará una copia de ese bloque en la segunda instancia pero en una versión anterior a la que tiene la primera.

#### 1.4.2.4.6.10 Global Cache Management (CGS)

Cache Fusion provee la transferencia de bloques de datos entre el buffer cache de la primer instancia al buffer cache de la segunda instancia.

Le proceso CGS lleva un registro de la ubicación, tipos y roles de los bloques de datos. El proceso CGS juega un papel muy importante para el Global Cache Management, adquiriendo recursos a nivel del cluster y ayudando a que el cache tenga coherencia cuando un mismo bloque de datos sea requerido tanto por la primera como por la segunda instancia.

#### 1.4.2.4.6.11 Mastering of resources

En base a la demanda de recursos en una instancia, se puede dar el caso en que a una instancia le llegue "n" cantidad de requerimientos para acceder a un bloque de datos, pero al mismo tiempo la segunda instancia requiere tener acceso al mismo bloque de datos, pero con una menor carga de requerimientos.

La instancia que tenga más requerimientos siempre tendrá el control de los recursos, y si la otra instancia requiere acceder, se coordinará con los procesos GCS y GRD para poder recuperar la información a través de la interconexión del cluster.

#### 1.4.2.4.6.12 Failover

Como se ha mencionado en capítulos anteriores, una de las grandes ventajas de tener un cluster de bases de datos, es la posibilidad de que los usuarios continúen accediendo a la base de datos en caso de que un nodo del cluster falle.

La implementación del Failover en el RAC puede ser de dos formas, host-based failover o Server-side failover.

**Host Based Failover.** Este método normalmente se implementa en configuraciones de dos nodos. Cuando se monitorean los servicios y la actividad de un nodo principal, y dicho nodo falla, todos los servicios se traspasan al segundo nodo, este proceso se lleva a cabo por medio de un servidor activo y el otro pasivo.

La transferencia de recursos y servicios se lleva a cabo justo cuando el nodo principal deja de responder.

**Server Side Failover.** Este método es completamente diferente al mencionado anteriormente, esta configuración se puede llevar a cabo por una configuración de servidores activo - activo.



Esta configuración se utiliza cuando todos los nodos del cluster tienen los servicios y recursos ejecutándose al mismo tiempo, así como todos los nodos tienen acceso al mismo disco. Cuando ocurre una falla solamente se hace un cambio de dueño.

El siguiente paso después de una falla en algún nodo, es la recuperación de la base de datos; los procesos necesarios para recuperar las transacciones de la instancia son ejecutados y la información es obtenida de los archivos redo log.

#### 1.4.2.4.7 Archivos utilizados por Oracle RAC

En el ambiente RAC, la mayoría de los archivos se relacionan con la base de datos y son compartidos entre las instancias, sin embargo existen ciertos archivos que no son compartidos y se encuentran en cada una de ellas. Se trata de archivos tales como archivos redo log, parameter file, archive log, entre otros.

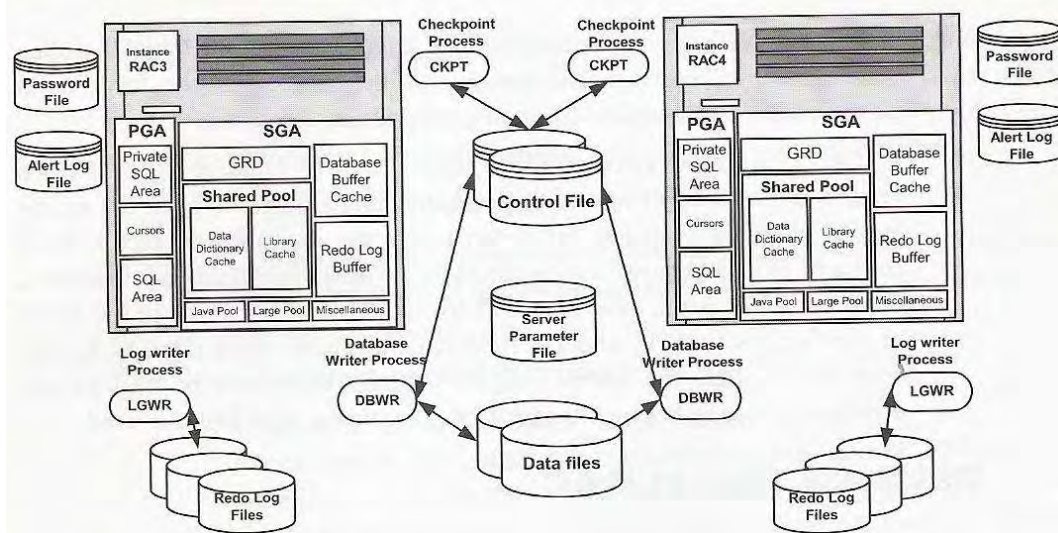
##### 1.4.2.4.7.1 Parameter file

El archivo de parámetros contiene las definiciones requeridas para el funcionamiento de la instancia, por lo tanto todos los nodos del cluster deben tener la misma configuración en el archivo de parámetros; existe una forma de lograr que sólo se tenga un archivo de parámetros para todas las instancias, la cual consiste en crear un archivo de parámetros binario, alojándolo en un almacenamiento común para todos los nodos.

El archivo de parámetros se puede utilizar de dos formas: como un archivo de texto en cada nodo, o como un archivo binario el cual puede ser compartido para todas las instancias.

La siguiente imagen muestra una configuración de RAC, con los archivos que son comunes para las instancias y los archivos que son requeridos en cada instancia. Por ejemplo el archivo de parámetros que se mencionó anteriormente.

Figura 44: Archivos en común para instancias en RAC



##### 1.4.2.4.7.2 Data files

Los data files o archivos de datos son utilizados para almacenar todos los bloques de datos que utiliza la base de datos, y que son accedidos por todas las instancias que se encuentran en el cluster. Estos archivos son almacenados en el sistema de almacenamiento común.

##### 1.4.2.4.7.3 Control files

El archivo de control contiene la información necesaria para la recuperación de la base de datos; estos archivos son comunes para todas las instancias y son utilizados cada vez que se inician las instancias, para poder comenzar a generar un registro de todas las actividades de la base de datos.

#### 1.4.2.4.7.4 Online redo log files

Los archivos de redo, contienen información relacionada con la instancia y se aloja en el almacenamiento compartido; estos archivos ayudan a las operaciones de recuperación de la base de datos cuando una instancia falla.

Las otras instancias tienen acceso a él y lo pueden usar para propósitos de recuperación.

La utilización de estos archivos es igual al de una instancia independiente, ya que cada instancia tiene sus propios archivos de redo.

#### 1.4.2.4.7.5 Archived redo log files

Estos archivos también se crean por cada instancia, y son utilizados para crear una copia de los archivos de redo, para que en caso de que exista alguna falla, la base de datos se pueda recuperar de manera correcta.

### 1.4.2.5 Tecnología Grid

La tecnología actual, y principalmente la informática, ha contribuido de forma única a la resolución de millones de problemas en diferentes ámbitos y disciplinas, constituyendo hoy en día el motor de procesamiento y fuente de recursos absolutamente imprescindible.

Desde sus orígenes, la informática ha visto la luz de su evolución en las actividades científicas, más precisamente en sus necesidades de almacenamiento y procesamiento de datos. Y si bien en la mayoría de los casos la ciencia y otra variedad de disciplinas han visto satisfechos sus requerimientos, aún quedan desafíos abordables que esperan a ser atendidos. Un claro ejemplo es la capacidad de procesamiento requerida en ambiciosos proyectos de investigación científica, simulaciones a gran escala, toma de decisiones a partir de grandes volúmenes de información y cientos de casos imaginables que no encuentran una solución, o quizá parte de ella, en las herramientas disponibles en la tecnología actual. En estos casos, ni aún la supercomputadora más sofisticada disponible en el mercado podría satisfacer la capacidad de procesamiento necesaria en casos donde el volumen de datos manipulado es de varios petabytes. Si bien un gran acercamiento a la resolución de estos problemas de enorme potencia de cálculo fue logrado mediante “clusters” o “granjas” de computadoras, surgidos a principios de los 80, aún seguían conservándose los recursos en forma dedicada. Existen también otras situaciones en que no sólo se persigue gran procesamiento, sino que también se busca lograr niveles de conectividad y cooperación entre proyectos científicos y académicos de gran escala que no han podido lograrse ni siquiera mediante Internet.

Muchas aplicaciones difundidas a través de Internet han ofrecido un modo más descentralizado para lograr gran potencia de cálculo gracias al aprovechamiento del tiempo de CPU inactiva aportado por sus usuarios, pero a pesar de este avance de importancia revolucionaria existen objetivos más cercanos de conectividad entre proyectos que se ven limitados en la red de redes. En realidad, el e-mail y la World Wide Web proveen modos básicos de trabajo conjunto pero no se ha logrado vincular y compartir datos, computadoras, sensores y demás recursos creando una entidad virtual. Es entonces aquí donde surge el concepto de “computación grid” (En castellano: rejilla, tramado, entrelazado, enrejado).

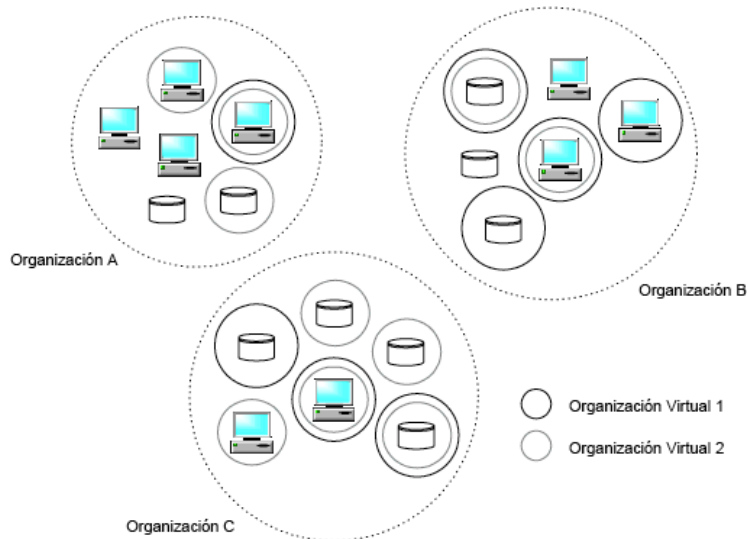
Varios conceptos similares coexisten acerca de qué es un grid. Uno de ellos, elaborado por el Grid Computing Information Centre, una de las asociaciones dedicada exclusivamente al desarrollo de esta tecnología, llama grid a un “tipo de sistema paralelo y distribuido que permite compartir, seleccionar y reunir recursos ‘autónomos’ geográficamente distribuidos en forma dinámica y en tiempo de ejecución, dependiendo de su disponibilidad, capacidad, desempeño, costo y calidad de servicio requerida por sus usuarios”. Según esta definición, se busca aprovechar la sinergia que surge de la cooperación entre recursos computacionales y proveerlos como servicios.

Otra definición más estructurada expuesta por Foster, Kesselman y Tuecke, precursores de la computación grid, plantea la existencia de organizaciones virtuales (OV) como puntos de partida de este enfoque. Una organización virtual es “un conjunto de individuos y/o

instituciones definida por reglas que controlan el modo en que comparten sus recursos". Básicamente, son organizaciones unidas para lograr objetivos comunes. Ejemplos de OV's podrían ser proveedores de servicios de aplicaciones o almacenamiento, equipos de trabajo empresarial realizando análisis y planeamiento estratégico, miembros de una planta de energía evaluando trabajo de campo, universidades involucradas en un proyecto de investigación conjunto, etc. Las OV's varían enormemente en cuanto a sus objetivos, alcance, tamaño, duración, estructura, comunidad y sociología. Sin embargo, existen varios requerimientos y problemas subyacentes tales como la necesidad de relaciones flexibles para compartir recursos, niveles de control complejos y precisos, variedad de recursos compartidos (programas, archivos, datos, sensores y redes, entre otros), modos de funcionamiento (individual, multiusuario), calidad de servicio, etc. Las tecnologías actuales o bien no proveen espacio para la variedad de recursos involucrados o no aportan la flexibilidad y control de las relaciones cooperativas necesarias para establecer las OV's.

Como solución, se propone el grid como un modelo de trabajo para "compartir recursos en forma coordinada y resolver problemas en organizaciones virtuales multi-institucionales de forma dinámica". De esta manera, varias instituciones pueden formar distintas OV's e incluso formar parte de más de una al mismo tiempo, realizando diferentes roles e integrando distintos recursos como se muestra en la siguiente figura:

Figura 45: Como modelo de trabajo



*Una organización real puede participar en una o más OV's compartiendo algunos o todos sus recursos. Se muestran tres organizaciones reales (los círculos de puntos) y dos OV's (conformadas por los recursos compartidos en cada caso). Las políticas que controlan el acceso a los recursos varían de acuerdo a las organizaciones reales, los recursos y las OV's involucradas. Nótese que existen recursos que componen una o más OV's y otros que no se comparten.*

Para lograr este nuevo cometido, se han desarrollado varios protocolos, servicios y herramientas que intentan sustentar organizaciones virtuales escalables. Debido a su enfoque basado en compartir recursos de manera dinámica y multi-organizacional, las tecnologías grid se complementan en vez de competir con las tecnologías de computación distribuida existentes.

#### 1.4.2.5.1 Sharing

El sharing de los recursos es condicional: cada dueño de los recursos hace disponible el mismo sujeto a condiciones de cuándo, dónde, y qué es lo que puede realizarse. Los consumidores de recursos pueden colocar condiciones en propiedades de los recursos para los cuales ellos están preparados para trabajar.



Las relaciones de sharing pueden variar dinámicamente sobre el tiempo en términos de recursos involucrados, la naturaleza de los accesos permitidos y los participantes que tienen accesos permitidos. Estas no son a menudo simplemente de tipo cliente-servidor, por peer to peer: los proveedores pueden ser clientes y las relaciones de sharing pueden existir sobre un subconjunto de participantes. Además pueden combinarse de forma coordinada sobre muchos recursos, cada uno perteneciente a diferentes organizaciones. La habilidad de delegar en vías controladas se torna importante en algunas situaciones como hacer mecanismos para coordinar operaciones a través de múltiples recursos.

Un mismo recurso puede usarse de diversas formas dependiendo de las restricciones situadas en el lugar desde donde se comparten, así como sus objetivos. Por ejemplo una computadora puede usarse solamente para correr piezas de software específicas en un arreglo de sharing mientras que éste puede proveer ciclos genéricos de operación en otra.

En el contexto de una red la interoperabilidad se refiere a protocolos comunes. Nuestra arquitectura Grid es primeramente una arquitectura de protocolos que definen los mecanismos básicos para cada usuario de las OV y la negociación de recursos, establecimiento, manejo y la utilización de las relaciones de sharing. La arquitectura basada en estándares facilita la extensibilidad, interoperabilidad, portabilidad y el sharing de código; los protocolos estándares hacen más fácil definir servicios estándares que provean capacidades. Se pueden construir interfaces de aplicaciones para programación y kits de desarrollo de software que provean abstracciones de programación requeridas para crear una Grid útil.

A través de la interoperabilidad se asegura que las relaciones de sharing puedan iniciarse en partes arbitrarias, acomodándose dinámicamente a nuevos participantes sobre diferentes plataformas, lenguajes y entornos de desarrollo.

La definición de un protocolo especifica cómo los elementos distribuidos de sistemas interactúan en orden de devolver un comportamiento específico y la estructura del intercambio de información durante su interacción.

Las OV tienden a fluir y se notan ciertos cambios: los mecanismos usados localizan recursos y se establecen y reconocen identidades, por ello la determinación de autorizaciones e iniciación del sharing debe ser flexible y liviano de forma que el arreglo de sharing de recursos pueda establecerse y cambiarse fácilmente.

#### **1.4.2.5.2 Servicios**

Un servicio se define por el protocolo con el que habla y el comportamiento que implementa. La definición de servicio estándar (de acceso a computación, datos, descubrimiento de recursos, replicación de datos y más) nos permite mejorar los servicios ofrecidos a participantes de las OV y también abstraer los detalles específicos de los recursos que de otra forma deberían ser posteriores al desarrollo de aplicaciones para OV.

#### **1.4.2.5.3 Funcionamiento de la computación grid**

Uno de los principales problemas que trae aparejada la implementación de la computación grid es la forma en que se administrarán los recursos que son parte de su infraestructura.

Los recursos que utiliza la computación grid se dividen en:

- Recursos que forman parte de la infraestructura, tales como recursos de hardware, de software y de red.
- Recursos que forman parte de los requerimientos; por ejemplo, el tiempo que se dispone de los recursos de infraestructura, los datos requeridos para un proceso dado, etc.

Por ser un sistema de cómputo distribuido, la computación grid presenta problemas típicos de éstos, por ejemplo a la hora de acceder a los recursos de cómputos y a los de red. Como se

trata de un sistema dinámico los recursos cambian, y por lo tanto se deben utilizar sistemas que se encarguen de buscar recursos disponibles. De la misma forma en que se pueden agregar nodos que aporten recursos, más tarde pueden desaparecer, reduciéndose los recursos que aportaban y el software que pudieran tener. También puede haber pérdida de recursos por una falla en las comunicaciones.

Para optimizar las infraestructuras distribuidas que forman el grid y facilitar su utilización, surge la necesidad de una arquitectura global que se encargue de prestar los servicios de asignación y planificación de recursos. Un ejemplo de esto es el proyecto CrossGrid que está orientado a aplicaciones paralelas e interactivas.

Este proyecto grid se centra en la utilización de un middleware que tiene la tarea de planificar y seleccionar eficientemente los recursos. A tal efecto cuenta con un Resource Broker como entidad principal, que presta servicios para la ejecución de trabajos secuenciales. Cuando un usuario envíe un trabajo éste será el encargado de buscar y seleccionar los recursos necesarios.

Para dar respaldo en el control de aplicaciones paralelas interactivas se cuenta con extensiones de los servicios prestados por el Resource Broker. Una de estas extensiones es el Scheduling Agent, una extensión de la funcionalidad que proporciona el Resource Broker como punto de acceso al Grid para el usuario y capaz gestionar los recursos y hacer que los procesos se ejecuten de manera transparente.

En esta arquitectura, los recursos disponibles se identifican como nodos de computación. Estos nodos representan una granja local compuesta por diferentes nodos de trabajo. También existen nodos de almacenamiento desde donde se tendrá acceso a los datos.

Cuando se debe realizar un trabajo, el encargado de verificar que recursos están disponibles es el Resource Selector. La asignación de dichos recursos dependerá si se trata de un trabajo secuencial o paralelo. Si se trata de un trabajo secuencial se deberá ejecutar en una única granja y requerirá un solo nodo de trabajo o CPU. A tal efecto existirá una lista de posibles granjas que podrían realizar el trabajo, estando ordenadas según algunas preferencias (tiempo ocioso de CPU, memoria, etc.). Un trabajo que requiera procesamiento paralelo necesitará varias CPUs y habrá que tener en cuenta que primero se tratarán de seleccionar granjas con todos sus procesadores libres, para evitar tener que establecer una comunicación entre clusters y sufrir la latencia producida por sus comunicaciones. Luego se intentará formar grupos de granjas que reúnan el número de CPUs libres requeridas. Y por último se tendrá un ordenamiento de grupos de menor a mayor cantidad de granjas diferentes y para los que igualen en elementos se utilizará una función de ponderación de sus componentes.

Un problema que surge de la utilización de estas listas de recursos es que puede ocurrir que no se encuentren actualizadas completamente en tiempo real, y puede darse el caso en que el Resource Broker o el Scheduling Agent considere una CPU como libre cuando realmente esté ocupada. Para evitar esto se trabaja con un método de reservas locales temporales. Una vez decidido a qué grupo se asignarán esos recursos, se les reserva por un espacio de tiempo y no serán tenidos en cuenta para trabajos siguientes.

También se utiliza el concepto de prioridades que permite que una aplicación se ejecute antes que otra. En el caso en que todos los recursos estén ocupados, se podrá decidir qué tarea será suspendida temporalmente para dar paso a otra de mayor prioridad. |

Una vez realizadas todas las consideraciones sobre los recursos y prioridades se debe permitir la ejecución del trabajo. El encargado de enviar los ejecutables de forma confiable es el Application Launcher, que hará la distribución a todos los recursos seleccionados que se encuentren implicados en la realización del trabajo.

Hasta aquí se ha dado una aproximación de cómo interactúan los distintos componentes del grid para que un paquete de procesamiento enviado por un usuario se resuelva de la forma más eficiente y transparente posible.

#### **1.4.2.5.4 Ventajas y desventajas y aplicaciones de la Computación Grid**

La Computación Grid está creada con el fin de brindar una solución a determinadas cuestiones, como problemas que requieren de un gran número de ciclos de procesamiento o acceso a una gran cantidad de datos. Encontrar un hardware y un software que permitan brindar estas utilidades comúnmente proporciona inconvenientes de costos, seguridad y disponibilidad. En ese sentido se integran diferentes tipos de máquinas y de recursos, por lo tanto una red grid nunca queda obsoleta, todos los recursos se aprovechan. Si se renuevan todas las PCs de una oficina, se pueden incorporar las antiguas y las nuevas.

Por otra parte, esta tecnología brinda a las empresas el beneficio de la velocidad, lo que supone una ventaja competitiva, con lo cual se provee una mejora de los tiempos para la producción de nuevos productos y servicios.

Facilita la posibilidad de compartir, acceder y gestionar información, mediante la colaboración y la flexibilidad operacional, aunando no sólo recursos tecnológicos dispares, sino también personas y aptitudes diversas. Otro de los aspectos al que se tiende es a incrementar la productividad otorgando a los usuarios finales acceso a los recursos de computación, datos y almacenamiento que necesiten, cuando los necesiten.

Con respecto a la seguridad en la grid, ésta está sustentada con las "intergrids", donde esa seguridad es la misma que ofrece la red Lan sobre la cual se utiliza tecnología grid.

El paralelismo puede estar visto como un problema, ya que una máquina paralela es muy costosa. Pero, si tenemos disponibilidad de un conjunto de máquinas heterogéneas de pequeño o mediano porte, cuya potencia computacional sumada sea considerable, eso permitiría generar sistemas distribuidos de muy bajo costo y gran potencia computacional.

Grid computing necesita, para mantener su estructura, de diferentes servicios como Internet, conexiones de 24 horas, los 365 días, con banda ancha, servidores de capacidad, seguridad informática, VPN, firewalls, encriptación, comunicaciones seguras, políticas de seguridad, normas ISO, y algunas características más... Sin todas estas funciones y características no es posible hablar de Grid Computing.

La tolerancia a fallos significa que si una de las máquinas que forman parte del grid colapsa, el sistema lo reconoce y la tarea se reenvía a otra máquina, con lo cual se cumple el objetivo de crear infraestructuras operativas flexibles y resistentes.

#### **1.4.2.5.5 Aplicaciones de la Computación Grid**

Actualmente, hay cinco aplicaciones generales para la Computación Grid:

##### **1.4.2.5.6 Súper computación distribuida**

Son aquellas aplicaciones cuyas necesidades no pueden ser satisfechas en un único nodo. Las necesidades se producen en instantes de tiempo determinados y consumen muchos recursos.

##### **1.4.2.5.7 Sistemas distribuidos en tiempo real**

Son aplicaciones que generan un flujo de datos a alta velocidad que debe ser analizado y procesado en tiempo real.

#### **1.4.2.5.8 Servicios puntuales**

Aquí no se tiene en cuenta la potencia de cálculo y capacidad de almacenamiento sino los recursos que una organización puede considerar como no necesarios. Grid presenta a la organización esos recursos.

#### **1.4.2.5.9 Proceso intensivo de datos**

Son aquellas aplicaciones que hacen un gran uso del espacio de almacenamiento. Este tipo de aplicaciones desbordan la capacidad de almacenamiento de un único nodo y los datos son distribuidos por todo el grid. Además de los beneficios por el incremento de espacio, la distribución de los datos a lo largo del grid permite el acceso a los mismos de forma distribuida.

#### **1.4.2.5.10 Entornos virtuales de colaboración**

Área asociada al concepto de Teleinmersión, de manera que se utilizan los enormes recursos computacionales del grid y su naturaleza distribuida para generar entornos virtuales 3D distribuidos.

Existen aplicaciones reales que hacen uso de mini-grids, las cuales están centradas en el campo de la investigación en el terreno de las ciencias físicas, médicas y del tratamiento de la información. Además existen diversas aplicaciones en el campo de la seguridad vial. Por ejemplo, este sistema permite traducir el riesgo de herir a un peatón y la resistencia del parachoques de un vehículo en una serie de datos que ayudan a diseñar la solución de protección más adecuada.

Entre los primeros proyectos grid, surge Information Power Grid (IPG), que permite la integración y gestión de recursos de los centros de la NASA. El proyecto SETI@Home a nivel mundial, de investigación de vida extra-terrestre, o búsqueda de vida inteligente en el espacio, puede ser considerado como precursor de esta tecnología. La idea de Grid Computing es mucho más ambiciosa puesto que no sólo se trata de compartir ciclos de CPU para realizar cálculos complejos sino que se busca la creación de una infraestructura de computación distribuida, con interconexión de diferentes redes, de definición de estándares, de desarrollo de procedimientos para la construcción de aplicaciones, etc.

#### **1.4.2.5.11 Algunos Proyectos en desarrollo bajo el sistema Data Grid**

##### **1.4.2.5.11.1 Globus Project**

El Proyecto Globus es una iniciativa multi-institucional para la investigación y el desarrollo de tecnologías fundamentales para Grids, con la activa participación de la empresa IBM, cuya intención principal es crear una plataforma completa donde compartir aplicaciones y recursos informáticos en Internet. Uno de los primeros productos desarrollados por el Proyecto Globus es el Globus Toolkit, que está siendo utilizado en varios proyectos de aplicación y despliegue de Grid en los Estados Unidos, Europa y el resto del mundo. El Proyecto Globus tiene su sede central en el Laboratorio Nacional Argonne y la Universidad del Instituto de Ciencias de Información de California del Sur.

El proyecto permitirá llevar las redes Grid más allá de las habituales aplicaciones técnicas y científicas para que pueda ser de utilidad en aplicaciones reales de negocio, conectando muchas supercomputadoras dispersas geográficamente mediante Internet y unos protocolos específicos de código abierto creados por la organización internacional Globus (globus.org).

La Arquitectura de Servicios Abiertos Grid (OGSA) presenta un conjunto de especificaciones y estándares que combina los beneficios de la informática Grid y los servicios web. Así, los clientes pueden, por primera vez, compartir y acceder a los recursos informáticos que necesitan en Internet, contando con el soporte de una infraestructura muy resistente, con capacidad de autogestión y siempre disponible; pueden integrar aplicaciones y compartir datos y potencia de procesado, consiguiendo unos niveles de eficiencia muy altos, así como muy bajos costos.

Este conjunto de especificaciones OGSA completa los estándares XML, WDSL y SOAP - todos ellos importantes para los servicios web-, con los estándares desarrollados por Globus para tecnologías de redes Grid, utilizados para localizar, planificar y asegurar recursos informáticos.

OGSA cuenta con el apoyo de empresas de diferentes industrias, incluyendo AVAKI, proveedor de soluciones comerciales de software Grid; Entropía, proveedor de informática de redes Grid distribuida basada en PC; Microsoft; y Platform Computing, proveedor de software de informática distribuida.

IBM tiene como objetivo la implantación de OGSA como punto clave en su "Proyecto eLiza". El proyecto eLiza es la iniciativa de informática autónoma de IBM para construir un servidor de infraestructura autogestionable, abierto y heterogéneo para el comercio electrónico y la puesta en práctica de Grids comerciales.

#### **1.4.2.5.11.2 TeraGrid**

Es un proyecto estadounidense, llevado adelante por la Fundación Nacional de Ciencias (NSF). Dedicado a la investigación científica abierta, tiene el objetivo de interconectar instalaciones y centros de investigación académica en puntos distantes geográficamente, está considerado como una de las infraestructuras más grandes y más rápidas del mundo.

A fin de permitir que los investigadores lleven a cabo proyectos de investigación y colaboren en ellos, se archivan datos de varias disciplinas científicas y de ingeniería. Los proyectos de esta magnitud son de valor crítico y fundamental para la comunidad científica y la recompensa que surja de la integración de soluciones para los investigadores a esta escala podría ser enorme, al reducir los costos y el tiempo de desarrollo y al crear productos más seguros.

Mediante el proyecto TeraGrid, los científicos tendrán la capacidad para simular actividades sísmicas en estructuras a fin de diseñar edificios y puentes más seguros, los astrónomos podrán compartir datos desde sus telescopios y los investigadores médicos tendrán la posibilidad de compartir ideas y datos para quizá curar una enfermedad, lo que asegura que las oportunidades que se ofrecen a los científicos son incalculables.

#### **1.4.2.5.11.3 CrossGrid**

Con el fin de poder ejecutar aplicaciones interactivas en un entorno Grid, se modifica el middleware adaptándolo para ello. En este proyecto se han definido cuatro aplicaciones que utilizarán desarrollos Grid comunes, que son: 1) Simulación interactiva y visualización de un sistema biomédico; 2) Sistema de apoyo a un equipo de crisis por inundaciones; 3) Análisis de datos distribuidos en Física de Altas Energías y 4) Previsión meteorológica y modelación de la contaminación atmosférica.

Los proyectos científicos relacionados con áreas como el análisis de la física de partículas, la biología computacional, la medicina, las ciencias medioambientales y la astrofísica, tienen su núcleo en el Reino Unido bajo el nombre de e-Ciencia, la cual sólo puede ser comprendida con el avance de la tecnología Grid o de computación distribuida. Precisamente, en torno a esta tecnología y a la e-Ciencia han surgido numerosos proyectos y múltiples centros de investigación se han centrado en su desarrollo.

Por su parte, CrossGrid, también proyecto de la Unión Europea nacido en 2001, tiene como fin el desarrollo, entre los años 2002 y 2005, de aplicaciones interactivas en entornos Grid y la extensión del banco de ensayo (testbed) de DataGrid.

#### **1.4.2.5.11.4 OpenMolGRID**

Desarrollado por ComGenex Inc., un proveedor de descubrimientos químicos avanzados, tiende a proporcionar información unificada y extensible del entorno para solucionar las cuestiones moleculares de diseño e ingeniería con relación a la química, la farmacia y la bioinformática. El proyecto utiliza un set de aplicación con herramientas orientadas a

establecer servicios centrales Grid y funciones provistas por la infraestructura Eurogrid, y se desarrollan herramientas para aumentar el acceso a las bases de datos heterogéneas y de distribución y para adaptar las herramientas de software existentes.

ComGenex es un proveedor de soluciones químicas integrales para las industrias farmacéutica y biotecnológica. Esta compañía dedicada a la formación de compuestos químicos, facilita el desarrollo de medicamentos en Europa. Cuenta con varias tecnologías propias de síntesis y análisis en las áreas de la química, la producción de instrumentos, la optimización de derivados, el análisis químico con aplicación médica y la bioinformática, y es compatible con ComGenexDirect, el primer sistema de comercio electrónico de la industria en el campo de los descubrimientos de nuevos fármacos.

ComGenex, el único socio comercial entre los cinco grupos de investigación en el proyecto, proporcionará diseño químico, química de altas prestaciones, biología celular y molecular, desarrollo de ensayos, monitorización de altas prestaciones y experiencia química y bioinformática.

#### **1.4.2.5.11.5 UK e-Science**

Un equipo de científicos del Reino Unido dio a conocer en una conferencia de gran magnitud, un elemento clave de la computación Grid que facilitará a los investigadores aprovechar enormes recursos informáticos de todo el mundo para afrontar los desafíos científicos clave en campos como el genoma humano y la física de partículas. Los responsables de la iniciativa elaboraron un conjunto de procedimientos que permitirán que los científicos que utilicen Grid accedan a las bases de datos de los resultados de investigación procedentes de sistemas que se encuentren en cualquier parte del mundo.

La intención es que las nuevas especificaciones sirvan para crear sistemas de prototipos de Grid, así como aplicaciones comerciales y científicas.

#### **1.4.2.5.11.6 EGEE (Enabling Grids for e-Science in Europe)**

El proyecto utilizará la tecnología Grid para interconectar recursos computacionales de veintisiete países europeos, con el objeto primordial de unir los recursos de los equipos informáticos de las instituciones participantes y crear de este modo una supercomputadora virtual, aprovechando la infraestructura de comunicación de banda ancha proporcionada por la Red Europea de Investigación Géant.

En el proyecto se plantea coordinar el uso conjunto de recursos de computación en forma distribuida entre los diferentes centros, constituyendo una plataforma donde aplicaciones de gran importancia en Informática Biomédica y en Física de Altas Energías funcionarán a máximo rendimiento. El centro que dirige el proyecto es el Laboratorio Europeo de Física de Partículas (CERN) que se encuentra en Ginebra (Suiza).

El Grupo de Redes y Computación de Altas Prestaciones de Valencia (GRyCAP) ha coordinado en los últimos años unos diez proyectos, la mayor parte de ellos europeos, orientados al desarrollo de aplicaciones informáticas avanzadas para el diagnóstico por imagen o la planificación quirúrgica. El GRyCAP se encargará en el proyecto EGEE de identificar, seleccionar y apoyar el despliegue de aplicaciones informáticas de apoyo a la asistencia sanitaria de forma que los usuarios médicos puedan tener acceso a las más avanzadas herramientas para el diagnóstico, la prevención o la simulación de la terapia tanto para la investigación clínica como la práctica diaria.

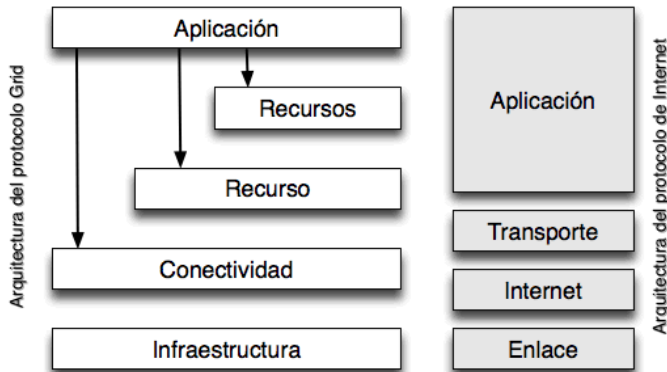
Asimismo, cuenta con una avanzada infraestructura informática formada por un Grupo de computadoras para el Grid y con facilidades de almacenamiento de datos. El objetivo es participar en la puesta a punto de un sistema computacional internacional montado en Grid que cubra las necesidades informáticas generadas por el procesado de datos procedentes de los experimentos de física de partículas. Esta iniciativa consiste en aglutinar los recursos informáticos con el fin de ser compartidos por investigadores de diferentes áreas científicas como la Astrofísica, la Meteorología, la Biología, la Nanofotónica, etc.

#### 1.4.2.5.12 Arquitectura grid

La arquitectura grid estándar propuesta en la figura 46 identifica componentes fundamentales del sistema y además las principales funciones y propósitos que deberían cumplir las aplicaciones, toolkits APIs, para componer un sistema grid.

Dicha arquitectura es una arquitectura abierta, donde los componentes del sistema se organizan en capas dentro de las cuales comparten características similares. La Figura siguiente muestra la arquitectura propuesta, organizada en 5 capas - Infraestructura, Conectividad, Recurso, Recursos y Aplicación - que definen mecanismos básicos que permiten a los usuarios gestionar los recursos compartidos.

Figura 46: Arquitectura de capas de un sistema Grid



##### 1.4.2.5.12.1 Capa de Infraestructura (Fabric)

En la primera de las capas, la capa de Infraestructura (Fabric), se encuentran los recursos computacionales que serán compartidos por las organizaciones virtuales - bases de datos, sistemas de almacenamiento de red, clusters - junto con la infraestructura de red y sus mecanismos de gestión y control, donde un recurso puede ser una entidad lógica (como un sistema de archivos distribuido, o un cluster de computadoras).

La funcionalidad de esta capa permite el sharing de operaciones sofisticadas.

Detalles de los recursos específicos presentes en esta capa:

##### 1.4.2.5.12.2 Recursos computacionales

Mecanismos requeridos para iniciar programas, controlar y monitorear la ejecución de procesos resultantes. El manejo de mecanismos que permite el control sobre los recursos alojados a procesos es usado como mecanismos avanzados de reserva. Se necesitan algunas funciones de evaluación para la determinación de características de hardware y software así como información de estado relevante.

##### 1.4.2.5.12.3 Recursos alojados

Se requieren algunos mecanismos para el guardado y recuperación de archivos. Hay mecanismos para la lectura y escritura de subconjuntos de archivos y/o ejecución remota de una selección de datos o reducción de funciones. El manejo de estos mecanismos permite el control sobre los recursos alojados, para transferencia de datos y otras funciones específicas.

##### 1.4.2.5.12.4 Recursos de red

Se requiere un control sobre los recursos utilizados en la red así como la transferencia de datos que se transmiten a través de ésta.

#### **1.4.2.5.12.5 Repositorios de recursos**

Es una forma especializada en almacenamiento de recursos que requiere mecanismos de manejo de versiones de fuentes y código de objetos.

#### **1.4.2.5.12.6 Catálogos**

Forma especializada de almacenamiento que requiere mecanismos para implementar consultas y actualización de operaciones (ej. BD relacionales).

#### **1.4.2.5.12.7 Capa de Conectividad (Connectivity)**

Dentro de la capa de Conectividad (Connectivity) se encuentran protocolos estándar de seguridad y comunicación para transacciones de red. Los protocolos de comunicación permiten el intercambio de datos entre la capa más inferior y los recursos mientras que los protocolos de seguridad brindan mecanismos de criptografía para identificar usuarios y recursos.

Los Recursos y los protocolos de conectividad mantienen todas las transacciones específicas de Grid entre diferentes computadoras y otros recursos. La red empleada por GRID es Internet, la misma red usada por la web y muchos otros servicios como el correo electrónico. Sin embargo, las tecnologías de Internet direccionan el intercambio de comunicaciones e información entre computadoras pero no provee aproximaciones integradas para el uso coordinado de recursos de diversos sitios de computación.

Esta capa define el centro de las comunicaciones y autenticación de protocolos requeridos para transacciones específicas de Grid dentro de la red. Los protocolos de comunicación proveen el intercambio de datos entre los recursos de la capa Fabric. Los protocolos de autenticación construyen la comunicación de servicios para proveer mecanismos seguros para la verificación e identificación de usuarios y recursos. Las comunicaciones requieren transporte, ruteo, y servicio de nombre (TCP, DNS, IP y ICMP).

Dentro de los protocolos de capa de conectividad se sitúan algunos protocolos correspondientes a la suite TCP/IP - dado que la comunicación implica por ejemplo ruteo, transporte, etc., se utilizan protocolos IP, ICMP, TCP - así como también el protocolo SSL y certificados X.509 como protocolos estándar que brindan seguridad para acceder a los recursos definidos en la capa de Infraestructura.

Las soluciones de autenticación para algunos entornos pueden tener características de Log In, Delegación (de forma que un usuario pueda finalizar un programa con la habilidad de correr con el provecho de otro usuario y que el programa sea capaz de acceder a recursos para los cuales está autorizado) e Integración con varias soluciones de seguridad. Las soluciones de Grid de seguridad deben ser capaces de interoperar con varias soluciones locales. Además, Usuarios basados en las relaciones confiables: en el cual el sistema de seguridad no debe requerir que cada proveedor de recursos coopere o interactúe con cada una de las configuraciones de los entornos de seguridad.

#### **1.4.2.5.12.8 Capa de Recurso (Resource)**

En el nivel correspondiente a la capa de Recurso (Resource) se encuentran los protocolos que permiten obtener la información de un recurso en particular y gestionarlo controlando el acceso, arranque de procesos, gestión, monitorización y auditoría.

Las implementaciones de estos protocolos llaman a funciones de capa Fabric para acceder y controlar recursos locales. Los protocolos de esta capa se refieren a recursos individuales y por ende ignoran resultados de estado global y acciones atómicas a lo largo de colecciones distribuidas, éstos son de consideración para la capa colección.

#### **1.4.2.5.12.9 Protocolo de información**

Para obtener información sobre la estructura y estado de los recursos.



#### **1.4.2.5.12.10 Protocolos de manejo**

Usados para negociar el acceso a un recurso compartido, especificando por ejemplo sus requerimientos (incluyendo reservas avanzadas ó calidad de servicio), y operaciones para optimizar estos recursos, como ser la creación de procesos o acceso a datos. A partir de que el manejo de protocolos es responsable para la inicialización de las relaciones del sharing deben servir como puntos de aplicación de política, asegurando que las operaciones requeridas por el protocolo sean consistentes con las políticas bajo las cuales el recurso es compartido. Los protocolos a menudo pueden soportar el monitoreo del estado de una operación y el control de la misma.

#### **1.4.2.5.12.11 Capa Recursos o Colectiva (Collective)**

Al contrario de la capa de recursos - destinada a gestionar un recurso específico - la siguiente capa definida en la figura 46 denominada Recursos o Colectiva (Collective), contiene los protocolos y servicios que permiten gestionar la interacción de un conjunto de recursos. Algunos ejemplos son los servicios de directorios (que permiten a las organizaciones virtuales descubrir y ubicar recursos compartidos), schedulers distribuidos (que permiten asignar tareas a cada recurso), servicios de monitorización y diagnóstico de recursos ante fallas y servicios de replicación de datos.

Dado que los componentes de capa colectiva construyen sobre capa Recursos y capa Conectividad pueden implementar una amplia variedad de comportamientos para el sharing sin la localización de nuevos requerimientos sobre los recursos almacenados.

**Por ejemplo:**

##### **Servicios de directorio**

Permite a participantes de las OV (Organizaciones Virtuales) descubrir la existencia y/o propiedades de los recursos de OV. Permite a sus usuarios hacer consultas para recursos por el nombre y/o por atributos, como ser tipo, disponibilidad o carga.

##### **Co-Allocation, Scheduling y servicios brokering**

Permiten a recursos de participantes de las OV averiguar sobre más recursos para propósitos específicos y para la programación de tareas sobre los recursos apropiados.

##### **Monitoreo y diagnóstico de servicios**

Soportan el monitoreo de los recursos de las OV para fallas, ataques de adversarios, detección de intrusos, sobrecarga y más.

##### **Servicios de replicación de datos**

Soporta el manejo de almacenamiento de los recursos pertenecientes a las OV para maximizar el rendimiento en los accesos a datos con las respectivas métricas como el tiempo de respuesta, costo, etc.

##### **Sistemas de programación GRID-ENABLED**

Permite usar en los entornos GRID modelos de programación familiar usando varios servicios Grid para localizar recursos, seguridad, etc.

##### **Sistemas de manejo de Workload y Collaboration Frameworks**

Conocidos como entornos de solución de problemas "PSE's", provee para la descripción, uso y manejo de multi-steps, asíncronos, flujos de trabajo multi-componentes, etc.

## **Sistemas de ubicación de software**

Descubre y selecciona las mejores implementaciones de software y ejecución de plataformas basadas en parámetros del problema a resolver. Ejemplo: NetSolve y Ninf.

## **Servidor de autorización de la comunidad**

Refuerza el gobierno de las políticas de acceso a recursos generando capacidades que los miembros de la comunidad pueden usar para el acceso a recursos comunitarios.

## **Servicios de cuentas de la comunidad y pagos**

Reúnen información acerca del uso de la misma, para el propósito de manejo de cuentas, pagos, y/o limitación de usos de recursos a miembros de la comunidad.

## **Servicios de colaboración**

Soportan el intercambio coordinado de información dentro de comunidades de usuarios potencialmente grandes. Mientras los protocolos de capa "Recursos" deben ser generales en naturaleza y desarrollados ampliamente, los protocolos de capa "Collective" expanden su espectro desde propósitos generales a aplicaciones altas o dominios específicos sólo entre OV específicas. Las funciones de esta capa pueden implementarse como servicios persistentes con protocolos asociados o SDK's designados para enlazarse con ciertas aplicaciones. En ambos casos sus implementaciones pueden construir en capa Recurso protocolos y API's.

Los componentes de capa Collective pueden crearse para requerimientos de usuarios de comunidades específicas, OV, o dominios de aplicaciones como por ejemplo un SDK que implementa protocolos de coherencia de aplicaciones específicas, o un servicio co-reservado para un conjunto específico de recursos en la red. Otros componentes de esta capa pueden ser de propósitos más generales por ejemplo, servicios de replicación que manejen una colección internacional de sistemas de almacenamiento para múltiples comunidades o un servicio directorio designado a permitir el descubrimiento de OV's.

Los servicios colectivos (collective services) se basan en protocolos: protocolos de información que obtienen datos sobre la estructura y estado de los recursos, y protocolos de manejo que negocian el acceso a recursos de una forma uniforme.

### **1.4.2.5.12.12 Capa de Aplicación (Application)**

La última capa definida es denominada Aplicación (Application). En esta se encuentran definidos los protocolos que permiten acceso a la estructura Grid. Según la figura 46. Las aplicaciones son construidas en términos de servicios definidos para alguna de las capas antes mencionadas, pudiendo, por ejemplo, comunicarse directamente con una capa en particular. Cada una de estas capas tiene protocolos bien definidos que proveen acceso al uso de servicios: manejo de recursos, acceso a datos, y más.

Si se considera que una aplicación de usuario necesita analizar datos contenidos en archivos independientes, se tendrá que realizar entonces las siguientes tareas básicas:

- Obtener la credencial necesaria de autenticación para abrir los archivos (recursos y protocolos de conectividad)
- Consultar el sistema de información y réplica de catálogos para determinar dónde pueden encontrarse las copias de los archivos en GRID así como también dónde se hallan los recursos más convenientes para hacer el análisis de datos (Collective Services)

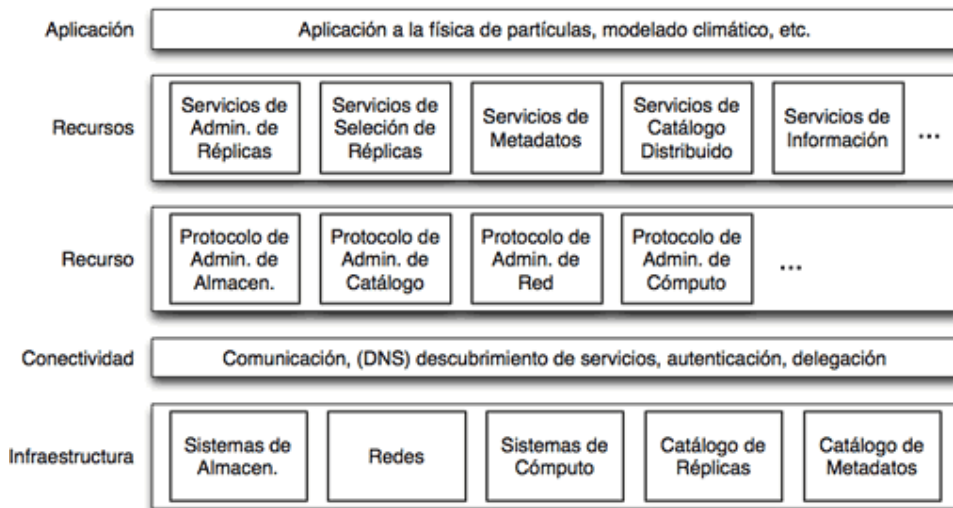
- Pasar los pedidos a la capa de Infraestructura, la computadora apropiada, sistema de almacenamiento y redes, para extraer los datos, iniciar los procesos, y proveer los resultados (recursos y protocolos de conectividad)
- Monitorear el progreso de varios procesos y transferencia de datos, notificando al usuario cuándo el análisis se completa y también detectando y respondiendo ante situaciones de fallas (collective services).

Hoy día hay diversos tipos de redes disponibles las cuales se caracterizan por su tamaño (locales, nacionales e internacionales) y según su rendimiento en términos de throughput (cantidad de datos transferidos desde un lugar a otro en un tiempo determinado). Típicamente el throughput se mide en Kbps, Mbps or Gbps. Grid está construida sobre redes de alto rendimiento tal es el caso de la red Intra-Europe GEANT o la red de UK Super Janet la cual exhibe 10 Gbps de rendimiento en el backbone de red (backbone: se utiliza para nombrar a aquellos enlaces de alta velocidad en las redes que unen grandes nodos).

A medida que crece la velocidad de la red, el poder de GRID es determinado por el rendimiento de los recursos de computación disponibles en los nodos de la red. Los nodos mayores serán recursos de alto rendimiento como un cluster largo de computadoras o cualquier supercomputadora dedicada.

Como ejemplo en la Figura 47 se propone una lista parcial de componentes correspondiente a cada una de las capas antes descritas, que podrían ser utilizados en un sistema grid relacionado a administración de réplicas.

Figura 47: Lista parcial de componentes en Grid

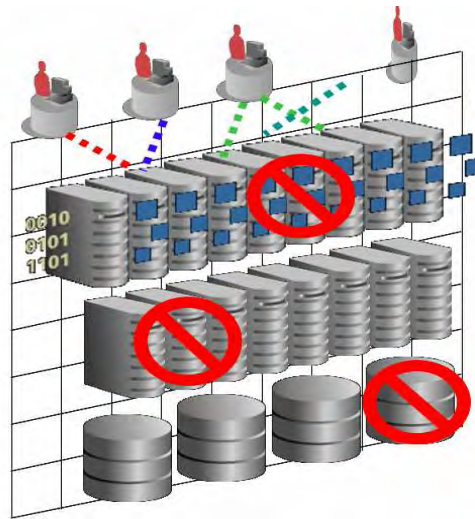


#### 1.4.2.5.13 Oracle GRID

Oracle Grid computing, agrupa en malla y distribuye servidores y recursos de almacenamiento. Cada sistema de malla puede ser configurado siguiendo el mismo patrón y al añadir una gestión centralizada a través de herramientas de gestión se reduce drásticamente la complejidad de administración y la monitorización. Ahora los servidores de aplicaciones, bases de datos y almacenamiento ya no son individuales para cada área de negocio, sino que trabajan a modo de panel.

Con esta infraestructura de grid computing se consigue almacenamiento eficiente para toda la información, utilización optima de recursos, máximo nivel de servicios, facilidad de gestión, mejora del desarrollo de aplicaciones y de las capacidades de datawarehousing.

Figura 48: Oracle Grid

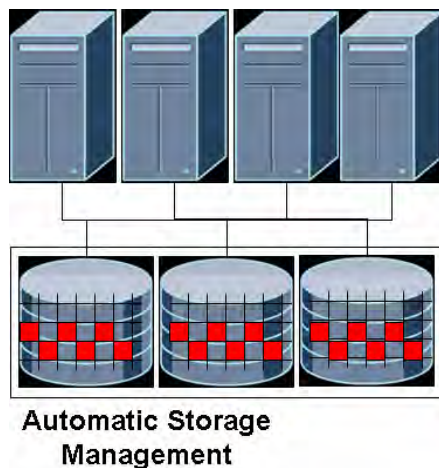


#### 1.4.2.5.13.1 Almacenamiento de la información

Para las bases de datos, el almacenamiento es un recurso clave. En un entorno estándar, distribuir los datos a lo largo de los discos para obtener un rendimiento máximo puede ser una tarea ardua y que consume mucho tiempo, que se ha de repetir cada vez que se añade o se quita un disco en el área de almacenamiento de la base de datos.

El componente Automatic Storage Manager (ASM) de Oracle Database 10g, soluciona este problema permitiendo a la base de datos administrar directamente discos en crudo. Elimina la necesidad de un sistema de archivos y gestor de volúmenes para gestionar los archivos de datos y log. ASM distribuye automáticamente todos los datos de la base de datos a través de todos los discos, proporcionando el mayor número de entradas/salidas sin ningún tipo de tarea de gestión. Según los discos que se añaden o retiran, incluso en caliente, ASM mantiene la distribución automáticamente. Por otra parte, la posibilidad de pérdida de datos debido a errores humanos se elimina, ya que ASM maneja la eliminación de los archivos que dejan de ser parte de la base de datos. ASM incluso detecta automáticamente picos de carga y mueve la información para aliviarlos. A continuación se muestra una imagen para esquematizar el funcionamiento de ASM.

Figura 49: ASM



Oracle 10g gestiona toda la información relacional, email, documentos, multimedia, XML y spatial. Las capacidades XML de Oracle 10g han sido extendidas para gestionar de manera más efectiva grandes cantidades de datos con un rendimiento y escalabilidad óptimos. Hay soporte

para tipos de datos Intermedia nativos, y la restricción de 4GB en LOBs almacenados en la base de datos se ha eliminado, elevándose a un límite de 128 Terabytes permitiendo el almacenamiento de documentos multimedia más grandes.

El límite del tamaño de la base de datos se ha elevado gracias a la instrucción de los archivos "ultra large", columnas LOB de tamaño ilimitado (Terabytes), administración mejorada de ASM, administración automática de espacio y una nueva área de recovery.

Para ayudar a migrar a un esquema de Grid computing, Oracle database 10g, incluye Oracle Data Puma, que permite el movimiento de datos desde una base de datos a otra de una manera rápida, a través de plataformas e incluso versiones de bases de datos.

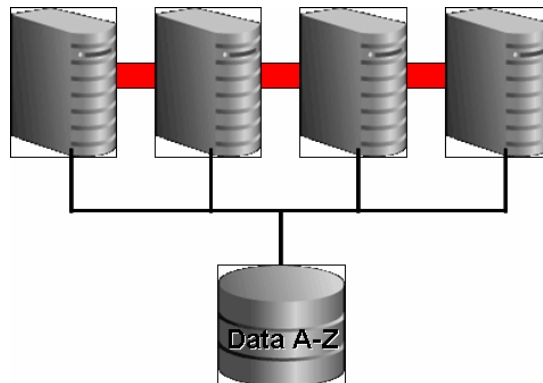
#### 1.4.2.5.13.2 Utilización Óptima de los Recursos

Oracle Real Application Cluster (RAC) ofrece la posibilidad de asignar servidores ya que es la única tecnología de bases de datos que soporta la ejecución de cualquier aplicación en un cluster. Oracle Database 10g hace que la gestión de un cluster sea más fácil con dos nuevas funcionalidades: Automatic Service Provisioning e Integrated Clusterware.

Automatic Service Provisioning ofrece la asignación y reasignación de los servidores a las cargas de trabajo (Servicios). Los clientes hacen login a los servicios y son automáticamente encaminados al servidor apropiado que en ese momento tiene la menor carga de trabajo. Si un servidor falla, los servidores supervivientes se reasignaran a los servicios.

Integrated Clusterware hace que RAC sea fácil de administrar debido a que elimina la necesidad de comprar, instalar, configurar y dar soporte a software clusterware de terceros. La imagen siguiente muestra un esquema de RAC.

Figura 50: Oracle RAC

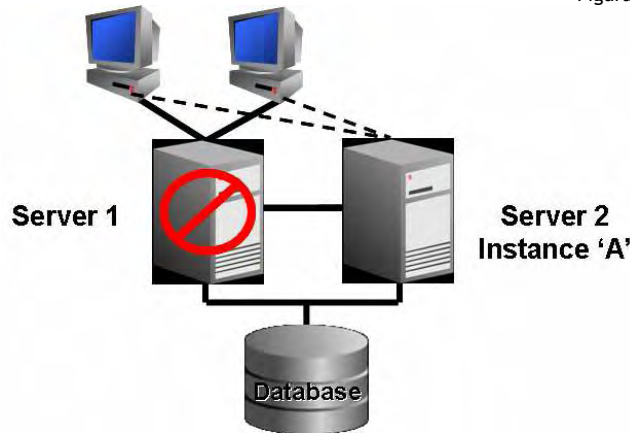


*Los servidores pueden añadirse y eliminarse a un cluster de RAC sin necesidad de inactividad.*

#### 1.4.2.5.13.3 Máximo Nivel de Servicios

Oracle 10g elimina el tiempo de caída no planificada ya sea por fallos en la información o en el servidor. Real Application Cluster protege de fallo de nodo en un Grid, gracias al failover automático de nodo y a la redistribución de la carga en tiempo real. La siguiente imagen muestra esta descripción.

Figura 51: Failover Automatico



Los errores humanos son generalmente la causa más común en caídas del sistema, y Oracle 10g es una revolución en términos de corrección de errores humanos a través de sus capacidades de flashback.

Además, la nueva característica Cluster Workload Management de RAC permite la respuesta rápida de fluctuaciones en los trabajos Grid.

Oracle 10g dispone de Flashback, que es una característica que permite al administrador "regenerar" la actividad en la base de datos y puede ser ejecutado a nivel de fila, transacción, tabla o incluso de base de datos. La recuperación hasta un punto en el tiempo puede ser realizada en pocos minutos, no siendo necesaria una restauración del backup y la consiguiente recuperación. Además, el backup automático y de recuperación puede mantener un área de recuperación en un disco que es actualizado incrementalmente, modificando sólo esos bloques que han cambiado desde el último backup proporcionando una recuperación mucho más rápida y fiable que los métodos tradicionales basados en cinta.

#### 1.4.2.5.13.4 Facilidad en la administración

Oracle Enterprise Grid Control gestiona las operaciones a través del Grid incluyendo la gestión de todo el software, los usuarios, etc. Oracle Database 10g ha dado un gran paso en la simplificación y automatización de todas las tareas en todo el ciclo de vida de la administración, de la base de datos. Se ha simplificado la instalación y configuración de la base de datos haciéndola mucho más rápida y ligera.

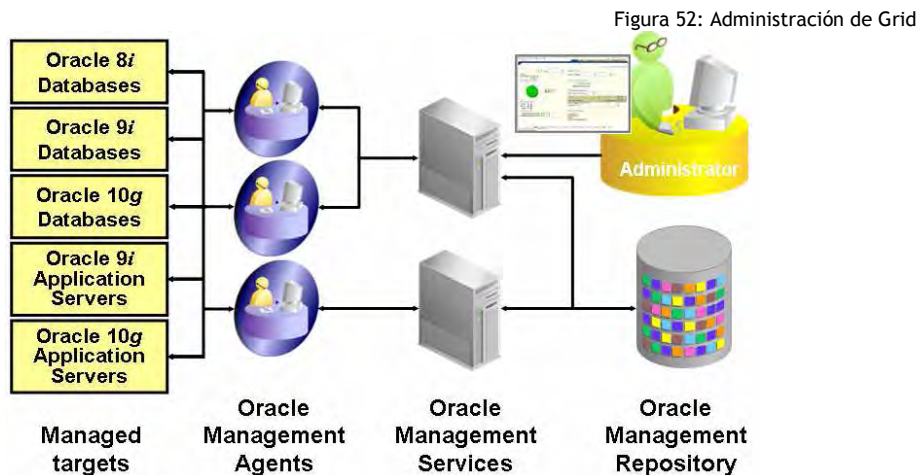


Figura 52: Administración de Grid



## 2 Planteamiento de Caso Práctico

### 2.1 Descripción de la empresa

Valecorp es una empresa dedicada a ofrecer servicios de financiamiento a través de vales de despensa, comida y gasolina a corporativos de todo tipo que desean tener un mayor control y servicios de financiamiento para prestaciones otorgadas a sus empleados.

Una de las metas que tiene Valecorp es ofrecer un servicio de calidad y seguridad a sus clientes.

Para lograr dichas metas Valecorp se interesa en adquirir tecnología de punta para dar el servicio requerido por sus clientes.

Valecorp es una empresa internacional con sucursales en las ciudades más importantes del mundo.

En la ciudad de México se encuentra una de las sucursales más importantes, desde la esta sucursal se distribuyen y administran vales de toda la republica mexicana, así como gran parte de Latinoamérica.

Valecorp considera que el utilizar la experiencia y conocimientos de otros países, las herramientas actuales y la innovación tecnológica mediante soluciones creativas, permiten proponer el financiamiento por medio de vales, sistemas con una nueva visión del futuro orientada a resolver la problemática y mejorar la calidad de vida mediante el uso oportuno de la tecnología.

La meta es la integración de la movilidad, la seguridad y la inter - operabilidad con los esquemas de globalización, intercambio de mercancías, personas e información.

Como resultado, en lo referente a la información, los sistemas de Valecorp cuentan con características que los hacen únicos en el mercado; ya que la información de los sistemas puede ser segmentada por niveles de usuario y por áreas o centros de costos dentro de la misma organización del cliente, permitiendo realizar conexión automática de la información de los servicios operados por Valecorp con los sistemas de administración propios de cada cliente y establecer líneas dedicadas de comunicación.

La solución en concepto es sencilla iniciándose al entregar vales de despensa a los clientes, dichos vales cuentan con la tecnología mas avanzada para evitar falsificaciones y evitar violaciones a las leyes del país. Cuando el cliente está entregando los vales a sus propios clientes, se tiene un registro electrónico de todos los vales que se distribuyen por el país.

El proceso de negocio de Valecorp se divide en diferentes procesos:

1. La fabricación de vales de despensa, y gasolina que se distribuirán con los clientes.
2. Venta y promoción de vales a los clientes.
3. Registro de clientes y afiliados.
4. Administración financiera de vales recibidos por los afiliados
5. Reembolso a los afiliados de vales adquiridos por clientes.
6. Administración de pedidos en línea para clientes y afiliados.

Valecorp dentro de sus diferentes procesos, utiliza diferentes sistemas, uno de los más importantes es el reembolso de los vales que reciben los afiliados, dichos vales se entregaron previamente a los clientes de Valecorp.

La empresa tiene un proceso completo para hacer el reembolso de los vales, es por ello que en este trabajo de tesis solo se mencionará este módulo.

Para este proyecto se implementará el sistema de reembolsos en un servidor de aplicaciones Oracle 10g, y una base de datos en oracle 10g con tecnología RAC.



El modulo de reembolsos interactúa con otras aplicaciones de la empresa utilizando procesos batch y otras bases de datos.

## **2.2 Descripción del sistema**

El proceso comienza cuando un prospecto de afiliado firma un contrato con VALECORP. Cuando el afiliado comienza a recibir vales de los clientes ya está en posición de elaborar una solicitud, ya sea de reembolso o anticipo, canje o devolución con el objetivo de que VALECORP devuelva el monto de esos vales.

La solicitud debe estar acompañada por los vales de manera física cuando ésta sea una solicitud de reembolso, canje o devolución. Para el caso de una solicitud de anticipo, el afiliado debe hablar por teléfono a VALECORP para levantar su solicitud de anticipo, el usuario captura información sobre el monto de los vales que vaya a presentar el afiliado y se le especifica una fecha de pago. En esa fecha el afiliado presenta la solicitud y los vales que se especificaron en la solicitud de anticipo; si por alguna razón el monto de los vales físicos que se presentan no coincide con algún dato de la solicitud de anticipo, su cheque no se le entrega. Cabe aclarar que las solicitudes de canjes y devoluciones las hace el cliente y no el afiliado.

La persona encargada de recibir la solicitud y los vales, en el área de reembolso, también es responsable de agruparlos y meterlos a un contenedor pudiendo ocurrir lo siguiente:

- Que varias remesas (conjunto de vales identificados por un número especificado en una tarjeta maestra) se encuentren en un mismo contenedor;
- Que una sola remesa ocupe varios contenedores o;
- Que una sola remesa este en un solo contenedor.

Ocurrida una de las situaciones anteriores el contenedor se cierra y se le asigna un documento llamado marbete el cual tiene información relacionada a la o las remesas incluidas en él.

Además de lo anterior, en SISRE (sistema de reembolsos), hace la lectura de cada uno de los vales que presentó el afiliado o cliente. La lectura de remesas la realiza un software llamado TRANSDOC. Los archivos resultantes de la lectura son puestos en un servidor de SISRE para posteriormente ser llevados a otro servidor, VALE\_PROD de VALECORP. Una vez que los archivos se encuentran en éste último, pasan por otro proceso denominado lectura.

Con los archivos en el servidor VALE\_PROD y el proceso de lectura ejecutándose, se insertan datos en algunas tablas del sistema REMESASREC. Esto se hace con el fin de que a través del sistema, se puedan detectar aquellas remesas que tengan alguna diferencia, ya sea mínima o significativa. En el caso de las remesas con diferencia mínima, éstas se autorizan para ser confrontadas. Cuando se trata de una remesa con diferencia significativa se puede mandar a releer a través de TRANSDOC hasta que no haya problemas y es entonces cuando la remesa está lista para confrontar. Si por alguna razón la diferencia no desapareció, el usuario del sistema tiene la opción de clasificar a la remesa como lista para confrontar y es hasta el proceso de tratamiento de anomalías donde se resuelven los problemas que haya tenido la remesa.

Los vales físicos que ya pasaron por el proceso de lectura o de análisis de remesa se van a un pre-almacen, el cual se encarga de resguardar y controlar los vales de las remesas en tanto se realiza su análisis y confrontación. De manera manual el encargado del pre-almacén revisa que el número de remesas descritas en el marbete se encuentren en el contenedor. Solo saldrán del pre-almacen las remesas o contenedores para relectura (cuando presenten diferencias o anomalías) y cuando la remesa este lista para ser destruida.

Se ejecuta un par de procesos más que es el de generar rangos y explosión de vales. El primero se encarga de poner a disposición de REMESASREC los rangos de folios de los vales que se van produciendo en SISRE, los que vayan llegando a las bóvedas de VALECORP y los que se hayan vendido. El proceso de explosión de vales se encarga de actualizar el estatus de

cada vale dependiendo si este fue recibido en bóveda o si fue vendido; en el caso de que el vale se haya emitido (producido), se inserta un nuevo registro en la base de datos de REMESASREC. Los procesos de generar rangos y explosión de vales se hacen antes de la confrontación con el objetivo de que este último haga comparaciones con datos actualizados.

El proceso de confrontación se encarga de verificar que los datos de cada vale leído por el sistema TRANSDOC coincidan con los datos que se encuentran dentro de la base de datos. El resultado de la confrontación nos va a permitir clasificar a las remesas como anómala, lista para generar su orden de pago, lista para enviar pedido de vales de un canje a CREDITNOTE o lista para generar una nota de crédito en el caso de que esa remesa pertenezca a una solicitud de devolución.

El proceso de confrontación valida principalmente que el vale no se reporte como robado, que no esté reprocesado, vencido, que no muestre un año de emisión válido, que la equivalencia entre la lectura y la emisión del vale con respecto al valor nominal, producto y dígito verificador sea idéntica. Que el afiliado o cliente tenga un contrato válido para el producto que desean rembolsar, canjear o devolver respectivamente.

Las remesas que no reporten ninguna anomalía están listas para generar una orden de pago y listas para destruir, para lo cual deben pasar por el almacén que es el encargado de recibir todos aquellos contenedores de remesas provenientes del pre-almacén para ser destruidas. La destrucción puede hacerse de dos maneras:

Destrucción Parcial. Sólo se destruirán aquellas remesas que no sean enviadas a resguardo  
Destrucción Total. Se destruye el contenedor completo de las remesas.

Las remesas que salgan con una anomalía son tratadas por un proceso llamado tratamiento de anomalías que es un método a través del cual se decide si se paga o no cada vale anómalo que aparezca dentro de una remesa. Ahora, suele suceder que el monto de lo que se leyó sea mayor a lo que indica la solicitud, en este caso VALECORP ajusta, baja el monto de lo que se leyó y lo paga. También sucede que lo que se leyó sea menor a lo reportado en la solicitud y es entonces cuando VALECORP ajustar o declarar un siniestro parcial.

Cuando la solicitud no reporta algún problema o cuando todas las anomalías ya se trataron se genera una orden de pago.

A partir de las órdenes de pago se generan movimientos contables y estos movimientos contables están controlados por el Sistema SAP, para el cual hay un proceso hecho en JAVA que se encarga de generar un archivo con tal información.

Lo anterior sería un proceso normal dentro del Sistema REMESASREC, los siguientes procesos se dan de manera poco frecuente, pero siguen siendo parte del mismo.

Algunas veces es necesario cancelar algunas órdenes de pago para lo cual también hay procesos dentro de REMESASREC. Estas cancelaciones también generan movimientos contables y en consecuencia hay que generar información para SAP.

En algunas ocasiones hay que generar cargos al afiliado básicamente por mantenimiento. Este proceso también origina movimientos contables y se tiene que generar información para SAP. Este último es uno de los llamados "Procesos Satelitales" que se ejecutan fuera del sistema REMESASREC.

Durante el manejo o transporte de los vales, estos últimos se pueden extraviar por varias razones, entre ellas el robo; dentro de REMESASREC hay un módulo o proceso que se encarga de manejar este tipo de eventos y se llama manejo de siniestros. Hay dos tipos de siniestro

- Cuando se tiene la solicitud y la información de sus remesas registradas, y se pierdan las remesas. En este caso puede haber:
  - SINIESTRO TOTAL: Cuando se pierden todos los vales de la solicitud.
  - SINIESTRO DE REMESA: Aplica a solicitudes por reembolso y sólo se pierde un porcentaje del total de las remesas.
  - Cuando no se tienen los datos de la solicitud y sus remesas.

REMESASREC también se encarga de llevar el control de los reembolsos de solicitudes de Vale Electrónico. Se hace un corte en donde se van acumulando las transacciones por día que vayan realizando los usuarios de las tarjetas y se realiza una confrontación de ellos. Tanto los cortes como la confrontación se hacen en CREDITNOTE y es a través de un proceso automático que los cortes ya confrontados son registrados en el sistema REMESASREC. Cada corte es pasado con un estatus de “no procesado” si no contiene transacciones anómalas o si llega a tener alguna anomalía, el corte se pasa con el estatus de “anómalo pendiente”. Con los cortes en REMESASREC se ejecuta un proceso que genera las solicitudes de reembolso de Vale Electrónico. En este punto el estatus de aquellos cortes sin anomalías pasan a “procesado” y los cortes que tengan anomalías pasan a un estatus de “anómalo”. Las solicitudes que no cuenten con ningún corte anómalo se consideran listas para generar orden de pago. Las solicitudes que contengan algún corte anómalo pasan por un proceso de tratamiento de anomalías en donde se identifican aquellas transacciones con problemas y se decide si se pagan o no. Cuando una solicitud ya se encuentra sin cortes anómalos está lista para generar una orden de pago.

### **2.3 Requerimientos Tecnológicos de la Empresa**

Valecorp, tiene como meta el ofrecer un servicio de calidad y seguridad a sus clientes. Para lograrlo requiere tener siempre tecnología de punta en su infraestructura.

Los requerimientos tecnológicos de la empresa son los siguientes:

- a) Confiabilidad a sus clientes
- b) Portabilidad de las aplicaciones
- c) Recuperación en caso de fallas
- d) Seguridad en el acceso a la información
- e) Revisión de la información de manera concreta y actual
- f) Flexibilidad con las aplicaciones
- g) Escalabilidad del hardware y aplicaciones
- h) Disponibilidad de los servicios

### 3 Análisis de Caso Práctico

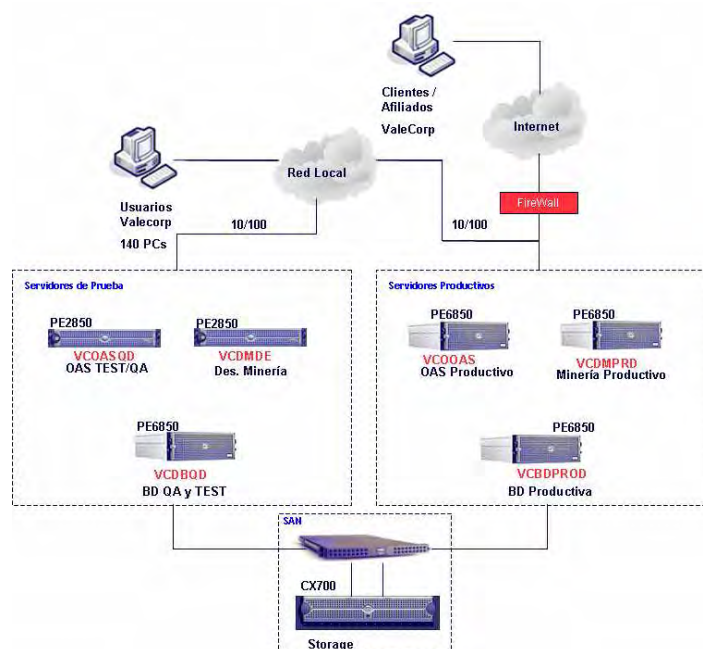
#### 3.1 Arquitectura Tecnología de la Empresa

Actualmente la empresa ValeCorp cuenta con diferentes servidores que almacenan las diferentes aplicaciones requeridas para la ejecución completa del proceso de negocio.

La empresa cuenta con diferentes tecnologías para lograr dar un servicio eficiente a los clientes y usuarios de la empresa.

- a) En oficinas se cuenta con 144 equipos de cómputo, los cuales son utilizados para personal que labora en la empresa, tales como personal administrativo, ventas, contabilidad, sistemas, recursos humanos, etc.
- b) Red interna en oficinas con velocidad 10/100
- c) Los servidores que utiliza la empresa se encuentran en un datacenter, con todas las condiciones que un site requiere para que los servidores funcionen de manera correcta.
- d) Valecorp cuenta con diferentes tipos de servidores:
  - PE2850: Utilizado como servidor de aplicaciones de prueba y desarrollo
  - PE 2850: Utilizado como servidor para minería de datos de prueba y desarrollo
  - PE6850: Utilizado para almacenar las instancias bases de datos de desarrollo y pruebas, así como la base de datos del servidor de aplicaciones
  - PE6850: Utilizado como servidor de aplicaciones productivo
  - PE6850: Utilizado como servidor de minería de datos productivo
  - PE6850: Utilizado para almacenar las instancias de bases de datos productivas, así como la base de datos del servidor de aplicaciones
  - CX700: Storage Area Network, donde se almacenan todas las bases de datos de la empresa, teniendo una redundancia tipo raid 0+1
- e) Para los clientes y proveedores que requieren tener acceso a los servidores desde una forma remota, se puede acceder a la red de ValeCorp por medio de una VPN la cual es autenticada a través de un Firewall.

Figura 54: Diagrama de Servidores



### **3.2 Niveles de Servicio**

Una de las metas que tiene Valecorp es ofrecer un servicio de calidad y seguridad a sus clientes y afiliados.

Actualmente las aplicaciones que se tienen para proveer un servicio de excelencia a los clientes y afiliados funcionan de manera óptima, pero la empresa ha sufrido grandes pérdidas económicas al tener problemas con los servidores de aplicaciones y bases de datos.

Los problemas más comunes que han tenido han sido falla en el hardware, pérdida de energía y problemas con el almacenamiento.

Actualmente la empresa no tiene algún esquema de alta disponibilidad en sus servidores y por lo tanto siempre tiene el riesgo de perder información y como consecuencia dinero.

Se tienen esquemas de respaldo y recuperación, pero dependiendo de la falla que se pueda dar en los servidores, el personal de TI puede llegar a tardar hasta 2 días en poder recuperar todo el servicio para los clientes.

Valecorp requiere cumplir con los requerimientos de negocios modernos, y con la infraestructura con la que cuenta actualmente no puede lograrlo, y por lo tanto no puede cumplir con la meta de ofrecer un servicio de calidad y seguridad para sus clientes y afiliados.

### **3.3 Aplicaciones Empresariales**

Valecorp cuenta con diferentes tecnologías de software, estas son utilizadas para diferentes actividades de la empresa, algunas de ellas son las siguientes:

1. Sistema Operativo Windows XP PRO, para todos los equipos de cómputo que se encuentran en las oficinas.
2. Oracle 10g, para bases de datos Productivas y Desarrollo
3. SAP, para administración contable y Recursos Humanos
4. Oracle Application Server 9i
5. Sistema Operativo Red Hat Enterprise, para servidores de aplicaciones y Bases de Datos.
6. MsOffice xp para todos los equipos administrativos de la empresa
7. Antivirus McAfee, para todos los equipos de cómputo de la empresa
8. VPN 1, como administrador de conexión VPN para clientes y proveedores
9. Toad, como administrador de Bases de Datos Oracle.

## 4 Propuesta de Diseño

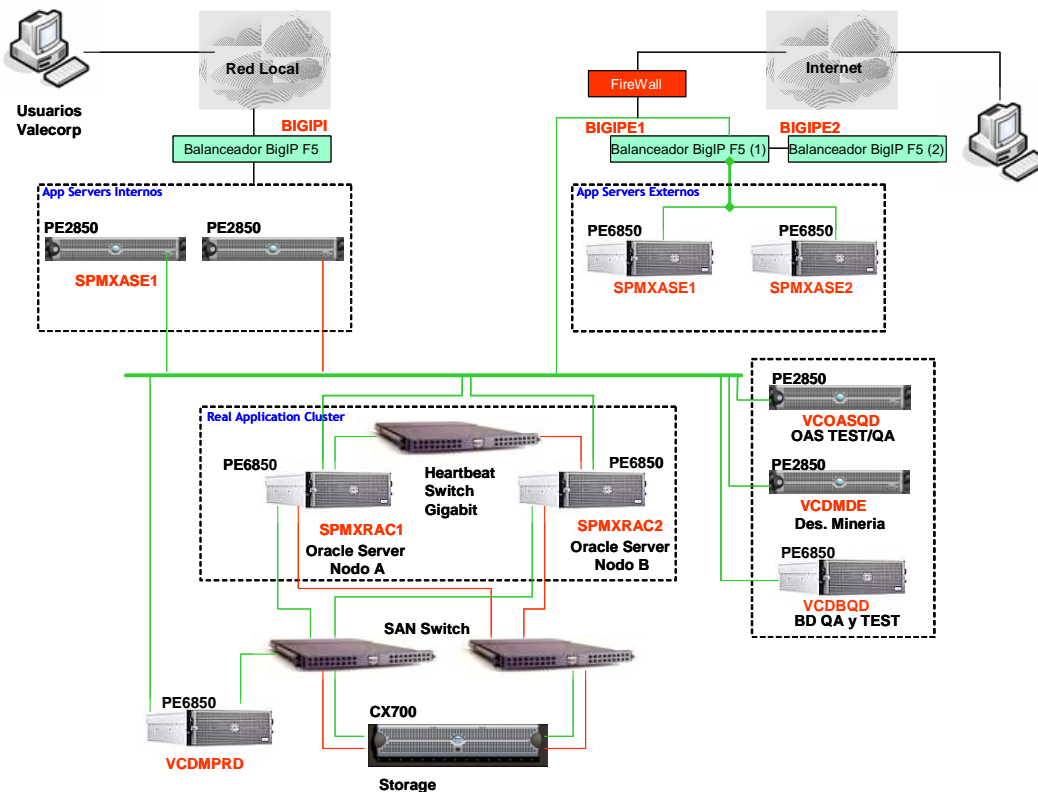
### 4.1 Arquitectura Tecnológica de la empresa

Después del análisis tecnológico de la empresa, se tomó la decisión de implementar un sistema de alta disponibilidad para servidores de aplicaciones y bases de datos, con la finalidad de cumplir con las metas que tiene la empresa.

Después de la implementación la infraestructura tecnológica, la empresa quedará de la siguiente forma:

- a) En oficinas se cuenta con 144 equipos de cómputo, los cuales son utilizados para personal que labora en la empresa, tales como personal administrativo, ventas, contabilidad, sistemas, recursos humanos, etc.
- b) Red interna en oficinas con velocidad 10/100
- c) Los servidores que utiliza la empresa se encuentran en un datacenter, con todas las condiciones que un site requiere para que los servidores funcionen de manera correcta.
- d) Después de la implementación, Valecorp tendrá los siguientes servidores:
  - PE2850 (VCOASQD): Utilizado como servidor de aplicaciones de prueba y desarrollo
  - PE 2850 (VCDMDE): Utilizado como servidor para minería de datos de prueba y desarrollo
  - PE6850 (VCDBQD): Utilizado para almacenar las instancias bases de datos de desarrollo y pruebas, así como la base de datos del servidor de aplicaciones
  - 2 PE 6850 (SPMXASI1 y SPMXASI2): Utilizados como servidores de aplicaciones internas productivas de la empresa, estos servidores se encuentran en un ambiente de cluster.
  - 2 PE6850 (SPMXASE1 y SPMXASE2): Utilizados como servidores de aplicaciones externas productivas de la empresa, estos servidores se encuentran en un ambiente de cluster
  - PE6850 (VCDMPRD): Utilizado como servidor de minería de datos productivo
  - 2 PE6850 (SPMXRAC1, SPMXRAC2): Utilizados para almacenar las instancias de bases de datos productivas, así como la base de datos del servidor de aplicaciones, estas instancias se encuentran en un ambiente de Oracle Real Application Cluster.
  - CX700: Storage Area Network, donde se almacenan todas las bases de datos de la empresa, teniendo una redundancia tipo raid 0+1
  - BigIP F5 (BIGIPI): Como balanceador de carga para servidores de aplicaciones internos.
  - 2 BigIP F5 (BIGIPE1 y BIGIPE2): Como balanceadores de carga y failover automático de servidores de aplicaciones externos. Estos balanceadores son redundantes
  - 1 Switch de Red Cisco: Que tiene la función de Heartbeat para el RAC, este switch es Gigabit
  - 2 Switches de Red Cisco para la SAN: Estos switches tienen la función de interconectar los servidores de base de datos con el storage, estos equipos son redundantes.

Para los clientes y proveedores que requieren tener acceso a los servidores desde una forma remota, se puede acceder a la red de ValeCorp por medio de una VPN la cual es autenticada a través de un Firewall.



#### 4.2 Niveles de Servicio

Después de la implementación los sistemas de valecorp tendrán un nivel de servicio mucho mayor, ya que cumplirán con los todos los requerimientos que un negocio moderno requiere, esto significa que las aplicaciones tendrán:

- Confiability:** Los sistemas de Valecorp podrán ser confiables ya que el tiempo de respuesta será mínimo, al tener tecnología de comunicaciones veloces, y servidores con el hardware suficiente para poder atender todas las solicitudes de los clientes.
- Portability:** Las aplicaciones se encuentran desarrolladas en J2EE, y están desarrolladas con las mejores prácticas propuestas por Sun Microsystems para poder garantizar una portabilidad a versiones futuras o cambios del proceso de negocio.
- Recuperación:** Con este esquema, las bases de datos se encuentran en un ambiente de RAC, lo cual garantiza que las instancias de base de datos siempre estarán disponibles para el usuario, en caso de que falle una siempre se tendrá la otra instancia, este esquema es un servicio activo-activo, de manera que no hay tiempo de espera para hacer un failover entre bases de datos. Los respaldos de bases de datos se siguen efectuando, pero ahora se hacen a nivel de servidor y se hacen en cinta.
- Seguridad:** El acceso a la red de Valecorp de forma remota está limitado por un firewall el cual sólo permitirá acceder a los usuarios que sean correctamente autenticados.
- Revisión:** Los procesos que se han implementado en la empresa para la creación de la información incluyen las reglas básicas para garantizar la revisión.
- Flexibilidad:** La flexibilidad en las aplicaciones se puede dar de manera más sencilla, ya que son administradas por el software Enterprise Manager de OAS, en el cual se puede controlar y administrar todas las aplicaciones.

- g) Escalabilidad: Los sistemas de la empresa pueden crecer de manera horizontal ya que al tener la tecnología de Oracle, se utiliza el Grid computing, esto significa que se pueden agregar nodos a los clusters de aplicaciones y de bases de datos sin necesidad de reprogramar algo.
- h) Disponibilidad: Como tema principal es la disponibilidad de las aplicaciones. Utilizando los clusters de aplicaciones y de bases de datos, se garantiza la disponibilidad de las aplicaciones las 24 horas del día.

Al cumplir con estos requerimientos, en caso de una falla el equipo de TI podrá recuperar el servicio en cuestión de segundos, de hecho dependiendo de la falla la aplicación podrá estar disponible 24 horas al día, 7 días a la semana, es decir tendrá un nivel de disponibilidad de 99.99%.

#### **4.3 Aplicaciones Empresariales**

Valecorp cuenta con diferentes tecnologías de software, estas son utilizadas para diferentes actividades de la empresa, algunas de ellas son las siguientes:

1. Sistema Operativo Windows XP PRO, para todos los equipos de cómputo que se encuentran en las oficinas.
2. Oracle 10g, para bases de datos Productivas y Desarrollo
3. Real Application Cluster para el ambiente de cluster en las bases de datos.
4. SAP, para administración contable y Recursos Humanos
5. Oracle Application Server 10g, con ambiente de cluster.
6. Sistema Operativo Red Hat Enterprise, para servidores de aplicaciones y Bases de Datos.
7. MsOffice xp para todos los equipos administrativos de la empresa
8. Antivirus McAfee, para todos los equipos de cómputo de la empresa
9. VPN 1, como administrador de conexión VPN para clientes y proveedores
10. Toad, como administrador de Bases de Datos Oracle

## **5 Implementación en Ambiente Corporativo**

### **5.1 Implementación de Oracle RAC**

Como archivo anexo a esta tesis, se encuentra la memoria técnica de esta implementación.

### **5.2 Implementación de Oracle Application Server en Cluster**

Como archivo anexo a esta tesis, se encuentra la memoria técnica de esta implementación.



## 6 Conclusión

Al principio de la era de la computación se sabe que solo se utilizaban los equipos de cómputo para algo específico o para cálculos que no eran sencillos de hacer manualmente; a lo largo de la evolución de los sistemas de cómputo, las sociedades y las organizaciones comerciales y no comerciales han llegado al punto en que la necesidad de tener un equipo de cómputo es imprescindible para realizar ciertas actividades.

Los equipos de cómputo los podemos encontrar prácticamente en cualquier lugar, ya sea con una computadora de escritorio, un teléfono celular, en un automóvil, etc. Con los equipos de cómputo las organizaciones han logrado organizar y administrar la información generada de una manera más eficiente y más barata, por lo que en estos tiempos lo más importante no es tener un equipo de cómputo con el que se puedan capturar datos, sino la manipulación de dichos datos y por consiguiente el almacenamiento de estos.

En el primer capítulo comentamos a cerca de las necesidades de una empresa u organización moderna, como las definiciones de los sistemas de información, tecnologías y ejemplos que se pueden utilizar para el manejo y mantenimiento de los datos obtenidos a lo largo de sus procesos, enfocándose en la implementación de los sistemas utilizados con alta disponibilidad.

En el segundo capítulo se llevó a cabo el planteamiento del proyecto especificando la forma en como opera uno de los sistemas más importantes para la empresa, este sistema se utiliza para generar una gran cantidad de la información utilizada. Como se mencionó anteriormente, lo más importante son los datos. Con esta información se concluyó que la empresa requiere tener la tecnología necesaria para lograr una alta disponibilidad en sus bases de datos y aplicaciones, ya que esta información representa dinero para el negocio.

En el tercer capítulo se hizo un análisis del estado actual de la empresa y su estructura tecnológica. En base a esta información es como se puede realizar una propuesta de diseño para llegar al objetivo de tener la información siempre disponible.

En el cuarto capítulo se hizo la propuesta de diseño y de la mejor solución para la empresa. Con esta solución podrá tener su información siempre disponible. Se puede concluir que este diseño además de habilitar una alta disponibilidad en los datos y aplicaciones, podrá contar con tecnología de punta y mejorará en un 40% el rendimiento de los servidores comparados con la estructura tecnológica que tenían anteriormente.

El quinto capítulo muestra las memorias técnicas utilizadas en la implementación de la Alta disponibilidad de las bases de datos y los servidores de aplicaciones que requiere la empresa. Después de realizar la implementación y hacer las pruebas correspondientes en los servidores se llegó a la conclusión de que los sistemas estarán disponibles en un 99.99% del tiempo anual.

Como conclusión final mencionaré que se cumplió el objetivo de la empresa moderna logrando implementar la tecnología necesaria para la alta disponibilidad tanto en bases de datos como aplicaciones, por consecuencia la empresa de vales tiene la confianza de dar un servicio eficiente a sus clientes así como estar a la vanguardia en un nivel tecnológico competitivo en el mercado del mismo ramo.



UNIVERSIDAD NACIONAL  
AUTÓNOMA DE  
MÉXICO

**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES  
CAMPUS ARAGÓN

**“DISEÑO Y CONFIGURACIÓN DE SERVICIOS CON  
ALTA DISPONIBILIDAD PARA SISTEMAS DE  
INFORMACIÓN OPERATIVOS CON TECNOLOGÍA  
ORACLE”**

**“Anexo 5.1, Implementación de Oracle Real Application Cluster”**

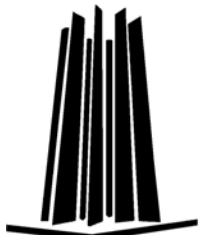
**T E S I S**  
QUE PARA OBTENER EL TÍTULO DE:  
INGENIERO EN COMPUTACIÓN

**P R E S E N T A:**

**OSWALDO VALENCIA LÓPEZ**

**ASESOR DE TESIS:**

**Ing. JOSÉ AGUSTÍN VARGAS PADILLA**



**MÉXICO, 2008.**

## **CAPITULADO**

- 1. Objetivo del documento**
- 2. Situación Actual**
- 3. Procedimientos de Instalación**
  - 3.1. Instalación y configuración hardware**
- 4. Instalación software de cluster de oracle**
- 5. Instalación de software rdbms oracle 10g**

## 1. Objetivo del documento

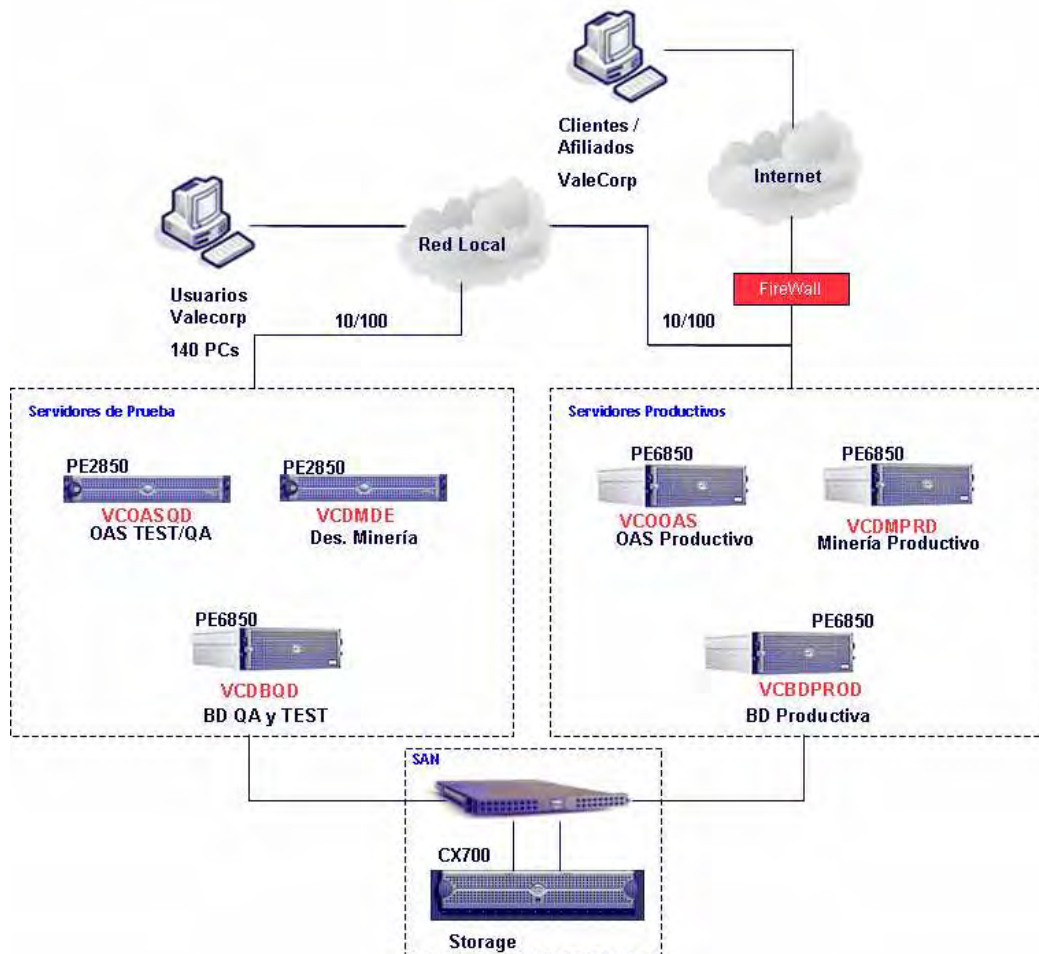
Esta memoria técnica tiene como finalidad documentar la implantación de Oracle RAC para la empresa ValeCorp

Este documento será desarrollado conforme al avance del proyecto, sin embargo al menos contendrá las secciones siguientes:

- Ambiente Actual
- Diseño Propuesto.
- Configuración de Hardware
- Instalación de Sistema Operativo
- Parametrización de Sistema Operativo
- Instalación Clusterware de Oracle
- Instalación de Oracle 10g RDBMS
- Configuración Oracle ASM
- Creación de Base de Datos UNIMAX
- PROD
- IMAG
- Migración de Datos
- Pruebas de funcionalidad.

## 2. Situación Actual

La empresa Valecorp contaba con la siguiente infraestructura de hardware:



Para efectos de la configuración de Oracle RAC se definieron los siguientes requerimientos:

- Adición de Switch Ethernet Gigabit para la interconexión de nodos.
- Actualización de Sistema Operativo RedHat a versión 4 AS
- Soporte vigente en Oracle y Red Hat por parte del cliente.
- Pruebas Preliminares de los aplicativos con Base de datos 10g

### 3. Procedimientos de Instalación

#### 3.1 Instalación y configuración hardware

##### Características de hardware nodos SPMXRAC1 y SPMXRAC2

DELL 6850  
2 Procesador Intel Xeon 3.0  
2 Discos Duros 73 GB (73 - RAID 1)  
8 GB RAM  
2 tarjetas de red Gigabit

##### distribucion de sistema de archivos

/dev/sda			
/dev/sda1	/boot	ext3	102
/dev/sda2	/	ext3	4997
/dev/sda3	/usr	ext3	4997
/dev/sda4			
/dev/sda5	/tmp	ext3	4001
/dev/sda6	swap		2047
/dev/sda7	swap		2047
/dev/sda8	swap		2047
/dev/sda8	swap		2047
/dev/sda9	/var	ext3	2047

##### Configuración física de disco nodos SPMXRAC1 y SPMXRAC2

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	1	13	104391	83	Linux
/dev/sda2		14	650	5116702+	83	Linux
/dev/sda3		651	1287	5116702+	83	Linux
/dev/sda4		1288	8908	61215682+	5	Extended
/dev/sda5		1288	1797	4096543+	83	Linux
/dev/sda6		1798	2058	2096451	82	Linux swap
/dev/sda7		2059	2319	2096451	82	Linux swap
/dev/sda8		2320	2580	2096451	82	Linux swap
/dev/sda9		2581	2841	2096451	83	Linux
/dev/sda10		2842	8908	48733146	83	Linux

##### Configuración para montaje automático oracle "u01" nodos SPMXRAC1 y SPMXRAC2

```
# This file is edited by fstab-sync - see 'man fstab-sync' for details
LABEL=/ / ext3 defaults 1 1
LABEL=/boot /boot ext3 defaults 1 2
none /dev/pts devpts gid=5,mode=620 0 0
none /dev/shm tmpfs defaults 0 0
none /proc proc defaults 0 0
none /sys sysfs defaults 0 0
LABEL=/tmp /tmp ext3 defaults 1 2
LABEL=/usr /usr ext3 defaults 1 2
LABEL=/var /var ext3 defaults 1 2
LABEL=SWAP-sda6 swap swap defaults 0 0
LABEL=SWAP-sda7 swap swap defaults 0 0
LABEL=SWAP-sda8 swap swap defaults 0 0
/dev/sda10 /u01 ext3 defaults 1 2
```

## Instalación sistema operativo Redhat linux 4 as

- 1) Configuración de idioma
- 2) Configuración de teclado
- 3) Asignación de particiones del sistema
- 4) Configuración de parámetros de inicialización
- 5) Asignación de contraseña grub (boot loader)
- 6) Asignación de dirección IP
- 7) Asignación de mascara de subred
- 8) Asignación de gateway
- 9) Asignación de servidor DNS
- 10) Asignación de nombre de host
- 11) Asignación de contraseña para root
- 12) Selección de programas para instalar
- 13) Entorno grafico
- 14) Librerías de compilación de kernel
- 15) Herramientas de desarrollo para compilar aplicaciones
- 16) Configuración de tarjeta de video
- 17) Configuración de parámetros de frecuencia para el monitor
- 18) Programación de inicialización modo texto
- 19) Reinicialización del sistema para entrar en modo funcional

### Software incluido

#### Desktop

X Windows

Gnome

#### Aplicaciones

Text based internet

#### Servidores

Server configuration tools

#### Desarrollo

Development tools

Legacy software

#### Sistema

Administration tools

#### Políticas de seguridad

Configuración de Firewall para aceptar administración remota SSH

```
#iptables -F
```

```
#service iptables save
```

```
#service iptables restart
```

#### Administración remota

Configuración servidor SSH

```
#service sshd restart
```

#### Actualización sistema operativo up2date redhat network

```
#up2date --register
```

```
#up2date -u
```

```
#up2date -p
```

### Inicialización de servicio vnc

```
#su oracle  
$ vncserver : 2 &
```

### Sincronización de tiempo

“tzselect”

```
/etc/cron.daily/ntpdate
```

```
#!/bin/sh
```

```
/usr/sbin/ntpdate clock.fmt.he.net dewey.lib.ci.phoenix.az.us louie.udel.edu
```

```
/usr/sbin/hwclock --systohc
```

### Seguridad para sincronización de tiempo

```
# clock.fmt.he.net
```

```
udp:in:d=123:s=216.218.192.202
```

```
# dewey.lib.ci.phoenix.az.us
```

```
udp:in:d=123:s=148.167.132.200
```

```
# louie.udel.edu
```

```
udp:in:d=123:s=128.4.40.12
```

### Configuración red

```
/etc/sysconfig/network-scripts/ifcfg-eth0
```

```
/etc/sysconfig/network-scripts/ifcfg-eth1
```

### Tabla de direcciones ip

```
/etc/hosts
```

```
127.0.0.1          localhost  
192.168.20.23     spmxcrac1.valecorp.com.mx   spmxcrac1  
192.168.20.24     spmxcrac1-vip.valecorp.com.mx spmxcrac1-vip  
172.16.0.1        spmxcrac1-priv.valecorp.com.mx spmxcrac1-priv  
192.168.20.25     spmxcrac2.valecorp.com.mx   spmxcrac2  
192.168.20.26     spmxcrac2-vip.valecorp.com.mx spmxcrac2-vip  
172.16.0.2        spmxcrac2-priv.valecorp.com.mx spmxcrac2-priv
```

### Parametrización sistema operativo

```
/etc/security/limits.conf
```

```
/etc/pam.d/login
```

```
/etc/profile
```

```
/etc/sysctl.conf
```

```
/etc/rc.d/init.local
```

### Creación de cuenta para usuario oracle

```
/usr/sbin/groupadd oinstall
```

```
/usr/sbin/groupadd dba
```

```
/usr/sbin/useradd -g oinstall -G dba oracle
```

### Configuración acceso entre nodos vía ssh

```
/usr/bin/ssh-keygen -t rsa
```

```
/usr/bin/ssh-keygen -t dsa
```



### Parámetros ajustados en el sistema operativo

```
# /etc/security/limits.conf
#
#Each line describes a limit for a user in the form:
#
#<domain>      <type> <item> <value>
#
#Where:
#<domain> can be:
#   - an user name
#   - a group name, with @group syntax
#   - the wildcard *, for default entry
#   - the wildcard %, can be also used with %group syntax,
#       for maxlogin limit
#
#<type> can have the two values:
#   - "soft" for enforcing the soft limits
#   - "hard" for enforcing hard limits
#
#<item> can be one of the following:
#   - core - limits the core file size (KB)
#   - data - max data size (KB)
#   - fsize - maximum filesize (KB)
#   - memlock - max locked-in-memory address space (KB)
#   - nofile - max number of open files
#   - rss - max resident set size (KB)
#   - stack - max stack size (KB)
#   - cpu - max CPU time (MIN)
#   - nproc - max number of processes
#   - as - address space limit
#   - maxlogins - max number of logins for this user
#   - priority - the priority to run user process with
#   - locks - max number of file locks the user can hold
#
#<domain>      <type> <item>      <value>
#
#*              soft  core      0
#*              hard  rss       10000
#@student       hard  nproc     20
#@faculty       soft  nproc     20
#@faculty       hard  nproc     50
#ftp            hard  nproc     0
#@student       -     maxlogins 4

oracle         soft  nproc     2047
oracle         hard  nproc     16384
oracle         soft  nofile    1024
oracle         hard  nofile    65536

# End of file
```

```
# /etc/pam.d/login
#
#%PAM-1.0
auth    required    pam_securetty.so
auth    required    pam_stack.so service=system-auth
auth    required    pam_nologin.so
account  required    pam_stack.so service=system-auth
password required    pam_stack.so service=system-auth
# pam_selinux.so close should be the first session rule
session  required    pam_selinux.so close
session  required    pam_stack.so service=system-auth
session  required    pam_loginuid.so
session  optional    pam_console.so
# pam_selinux.so open should be the last session rule
session  required    pam_selinux.so multiple open
session  required    pam_limits.so
# End of file

# /etc/profile
# System wide environment and startup programs, for login setup
# Functions and aliases go in /etc/bashrc

pathmunge () {
    if ! echo $PATH | /bin/egrep -q "(^|:)$1($|:)" ; then
        if [ "$2" = "after" ] ; then
            PATH=$PATH:$1
        else
            PATH=$1:$PATH
        fi
    fi
}
# Path manipulation
if [ `id -u` = 0 ]; then
    pathmunge /sbin
    pathmunge /usr/sbin
    pathmunge /usr/local/sbin
fi

pathmunge /usr/X11R6/bin after

# No core files by default
ulimit -S -c 0 > /dev/null 2>&1
USER=`id -un`
LOGNAME=$USER
MAIL="/var/spool/mail/$USER"
HOSTNAME=`/bin/hostname`
HISTSIZE=1000

if [ -z "$INPUTRC" -a ! -f "$HOME/.inputrc" ]; then
    INPUTRC=/etc/inputrc
fi

export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE INPUTRC

for i in /etc/profile.d/*.sh ; do
```

```
if [ -r "$i" ]; then
    . $i
fi
done
unset i
unset pathmunge

if [ $USER = "oracle" ]; then
    if [ $SHELL = "/bin/ksh" ]; then
        ulimit -p 16384
        ulimit -n 65536
    else
        ulimit -u 16384 -n 65536
    fi
fi
# End of file

# /etc/sysctl.conf
# Kernel sysctl configuration file for Red Hat Linux
#
# For binary values, 0 is disabled, 1 is enabled.  See sysctl(8) and
# sysctl.conf(5) for more details.

# Controls IP packet forwarding
net.ipv4.ip_forward = 0

# Controls source route verification
net.ipv4.conf.default.rp_filter = 1

# Do not accept source routing
net.ipv4.conf.default.accept_source_route = 0

# Controls the System Request debugging functionality of the kernel
kernel.sysrq = 0

# Controls whether core dumps will append the PID to the core filename.
# Useful for debugging multi-threaded applications.
kernel.core_uses_pid = 1

kernel.shmall = 2097152
kernel.shmmax = 2147483648
kernel.shmmni = 4096
kernel.sem = 250 32000 100 128
fs.file-max = 65536
net.ipv4.ip_local_port_range = 1024 65000
net.core.rmem_default = 262144
net.core.rmem_max = 262144
net.core.wmem_default = 262144
net.core.wmem_max = 262144
# End of file
```

```
# /etc/hosts
# Do not remove the following line, or various programs
# that require network functionality will fail.

127.0.0.1    localhost

# CONFIGURACION ORACLE CLUSTER 10g r2

192.168.20.23 spmxrac1.valecorp.com.mx    spmxrac1
192.168.20.24 spmxrac1-vip.valecorp.com.mx    spmxrac1-vip
172.16.0.1    spmxrac1-priv.valecorp.com.mx    spmxrac1-priv
192.168.20.25 spmxrac2.valecorp.com.mx    spmxrac2
192.168.20.26 spmxrac2-vip.valecorp.com.mx    spmxrac2-vip
172.16.0.2    spmxrac2-priv.valecorp.com.mx    spmxrac2-priv

/etc/rc.d/init.local

#!/bin/sh
#
# This script will be executed *after* all the other init scripts.
# You can put your own initialization stuff in here if you don't
# want to do the full Sys V style init stuff.

touch /var/lock/subsys/local

# ORACLE CLUSTER NETWORK ROUTING
route add -host 172.16.0.1 dev eth1
route add -host 172.16.0.2 dev eth1
route del -net 172.16.0.0/16
route add -net 172.16.178.0/24 gw 192.168.20.1 dev eth0
route add -net 172.16.1.0/24 gw 192.168.20.1 dev eth0
```

#### Tabla de ruteo actual de ambos nodos

```
[root@spmxrac1 ~]# route -n
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
172.16.0.1 0.0.0.0 255.255.255.255 UH 0 0 0 eth1
172.16.0.2 0.0.0.0 255.255.255.255 UH 0 0 0 eth1
172.16.178.0 192.168.20.1 255.255.255.0 UG 0 0 0 eth0
192.168.20.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0
169.254.0.0 0.0.0.0 255.255.0.0 U 0 0 0 eth1
0.0.0.0 192.168.20.1 0.0.0.0 UG 0 0 0 eth0
```

#### Nota:

La configuración de ruteo se carga de forma automática, en caso de reiniciar el servicio de red manualmente las tablas de ruteo se eliminan. Por lo que es necesario ejecutar los siguientes comandos:

```
route add -host 172.16.0.1 dev eth1
route add -host 172.16.0.2 dev eth1
route del -net 172.16.0.0/16
route add -net 172.16.178.0/24 gw 192.168.20.1 dev eth0
route add -net 172.16.1.0/24 gw 192.168.20.1 dev eth0
```

La tabla de ruteo debe validarse de la siguiente forma:

**Tabla de ruteo actual de ambos nodos**

```
[root@spmxcrac1 ~]# route -n
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
172.16.0.1 0.0.0.0 255.255.255.255 UH 0 0 0 eth1
172.16.0.2 0.0.0.0 255.255.255.255 UH 0 0 0 eth1
172.16.178.0 192.168.20.1 255.255.255.0 UG 0 0 0 eth0
192.168.20.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0
169.254.0.0 0.0.0.0 255.255.0.0 U 0 0 0 eth1
0.0.0.0 192.168.20.1 0.0.0.0 UG 0 0 0 eth0
```

Todos el software empleado en este proyecto esta en ambos nodos bajo el siguiente directorio

/software

```
[root@spmxcrac1 ~]# ls /software/
10201_clusterware_linux32.zip clusterware DELL PROFTPD SCRIPTS
10201_database_linux32.zip database KERNEL route-table.sh
```

#### 4. Instalación software de cluster de oracle

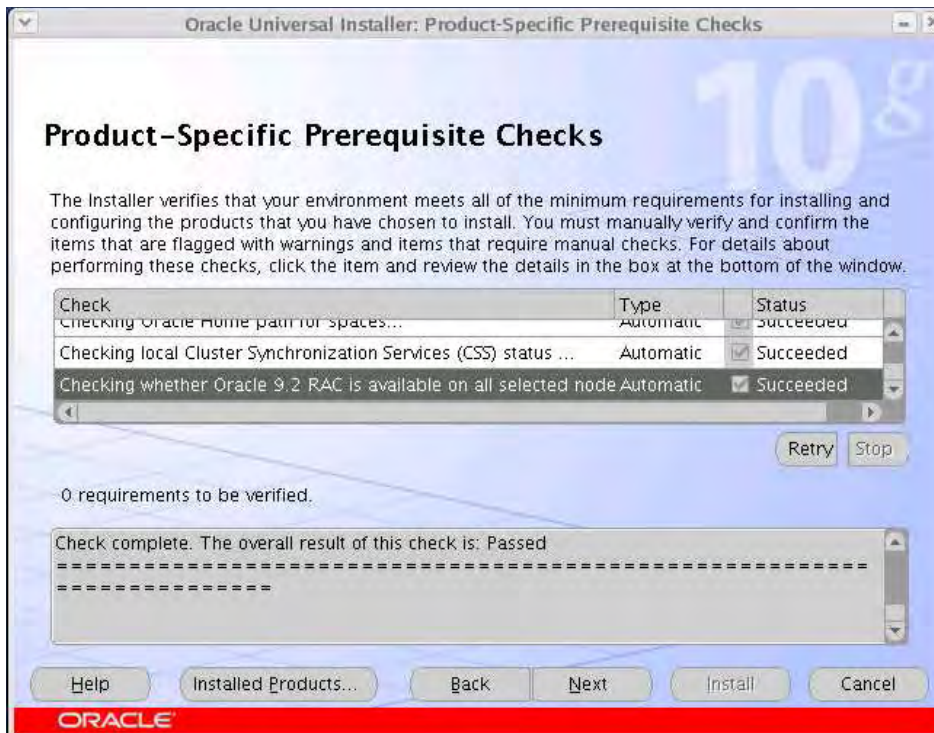
1. Descargar y desempacar el software de "Oracle Clusterware" en una carpeta con permisos para que el usuario Oracle del sistema operativo pueda tener acceso a ella. Ya teniendo el software listo, ejecutamos el comando: runInstaller, el cual iniciara el instalador de la aplicación



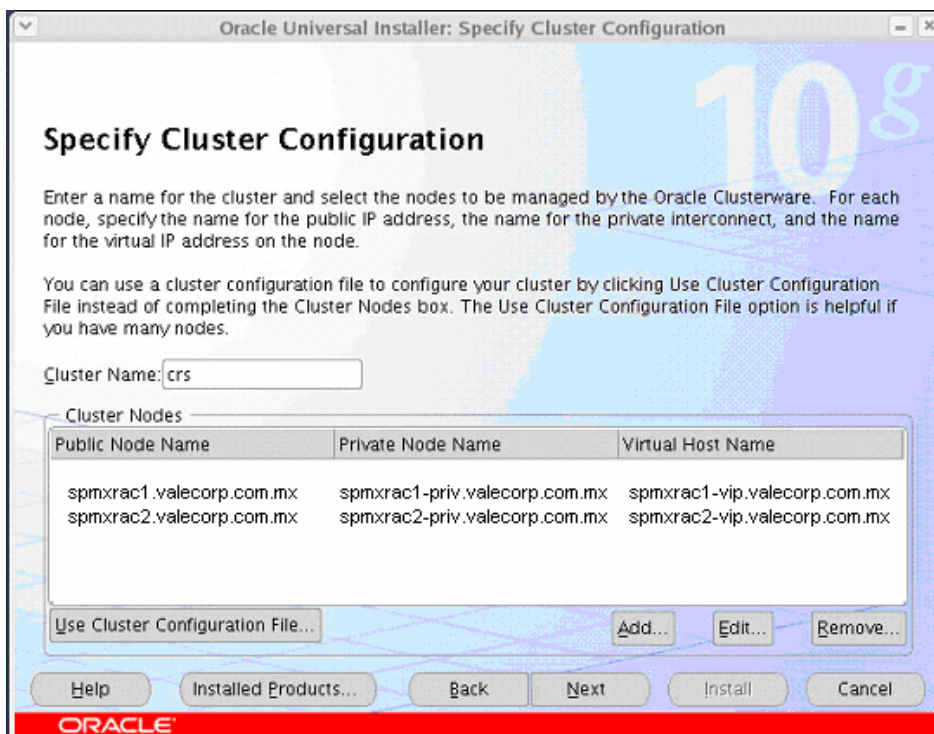
2. Especificamos el origen del software de instalación, el nombre de la instalación y el destino donde se instalara el software.



3. El asistente verifica que se cumplan los requerimientos mínimos del sistema, para poder continuar con la instalación.

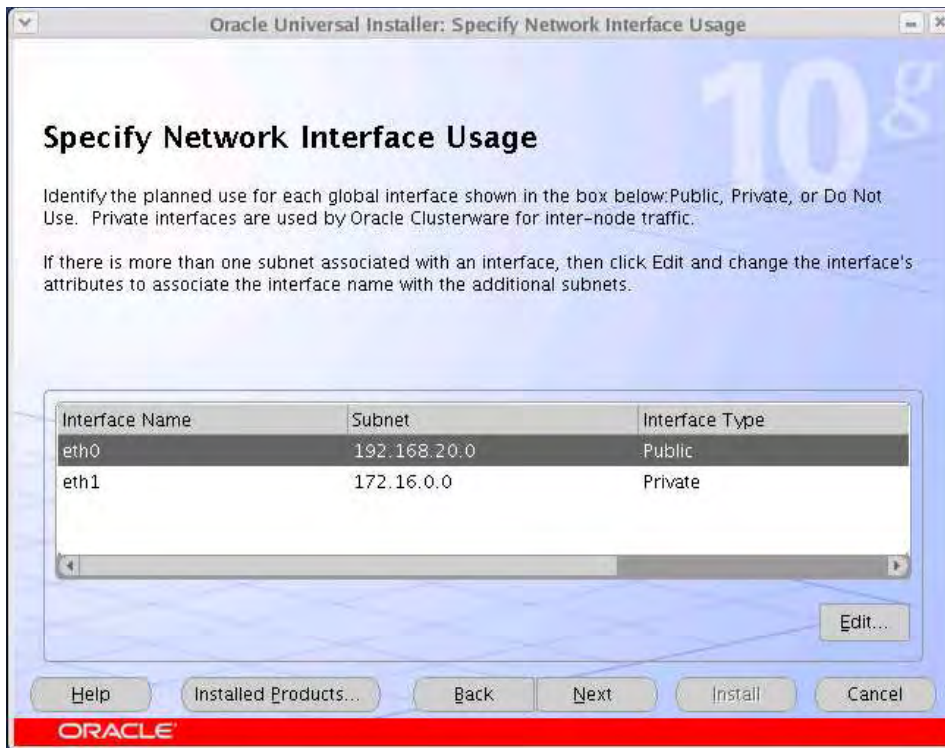


4. Especificamos un nombre para el cluster y agregamos los nodos que lo conformaran.

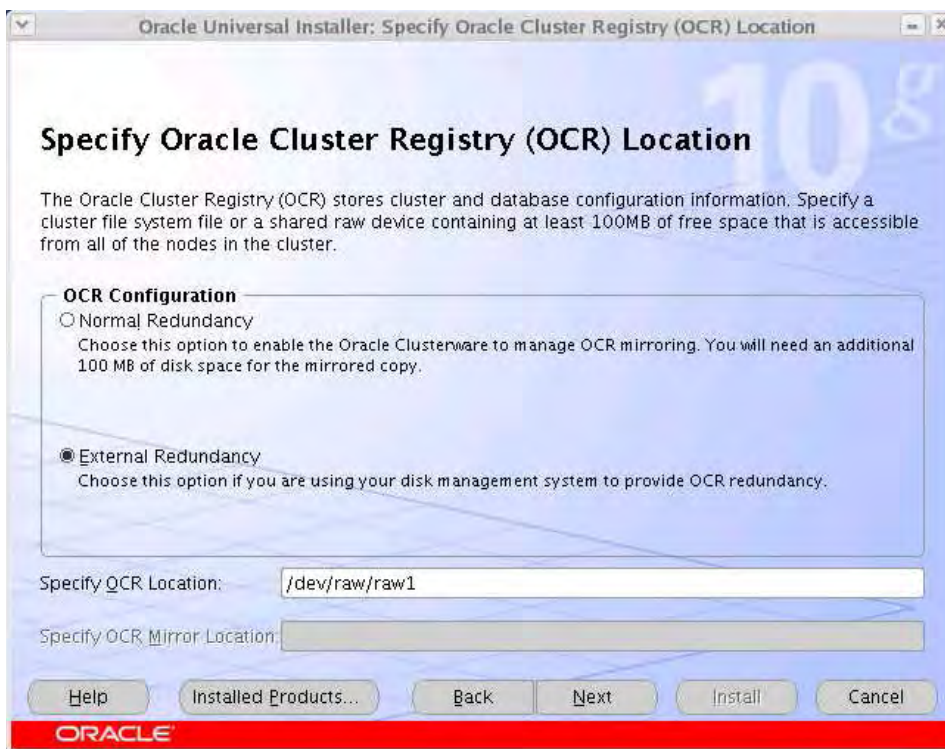




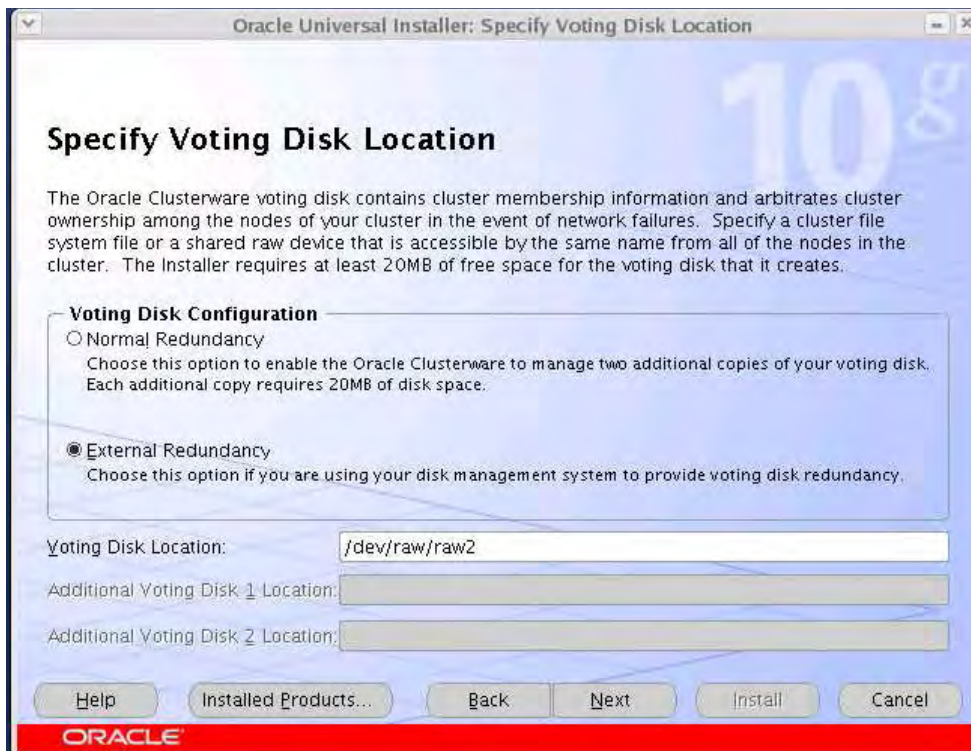
5. Especificamos las tarjetas de red que se utilizaran para la interconexión a la red publica y las que se utilizaran para la red privada (Heartbeat)



6. Especificamos la ubicación para el registro del cluster (OCR)



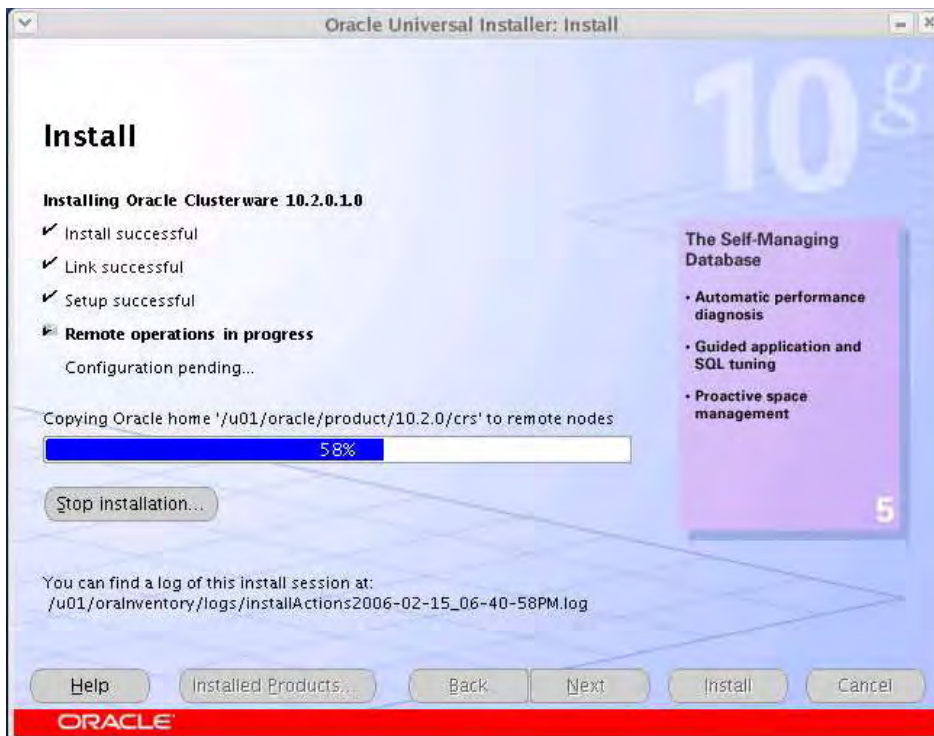
7. Especificamos la ubicación para el voting disk.



8. Se genera un resumen de la instalación y si no hay observaciones, se comienza la instalación.

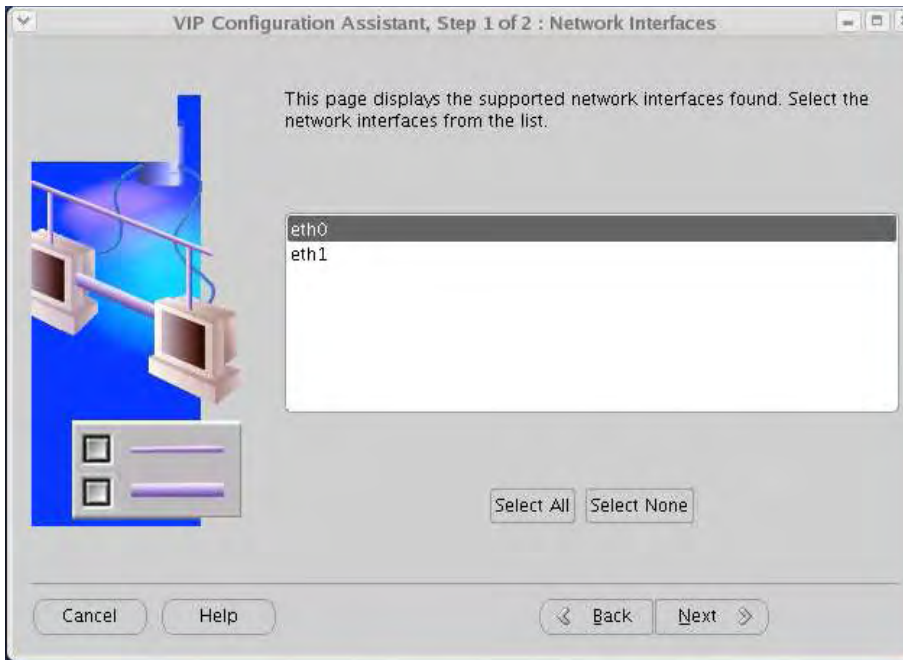


## 9. Progreso de la copia de archivos

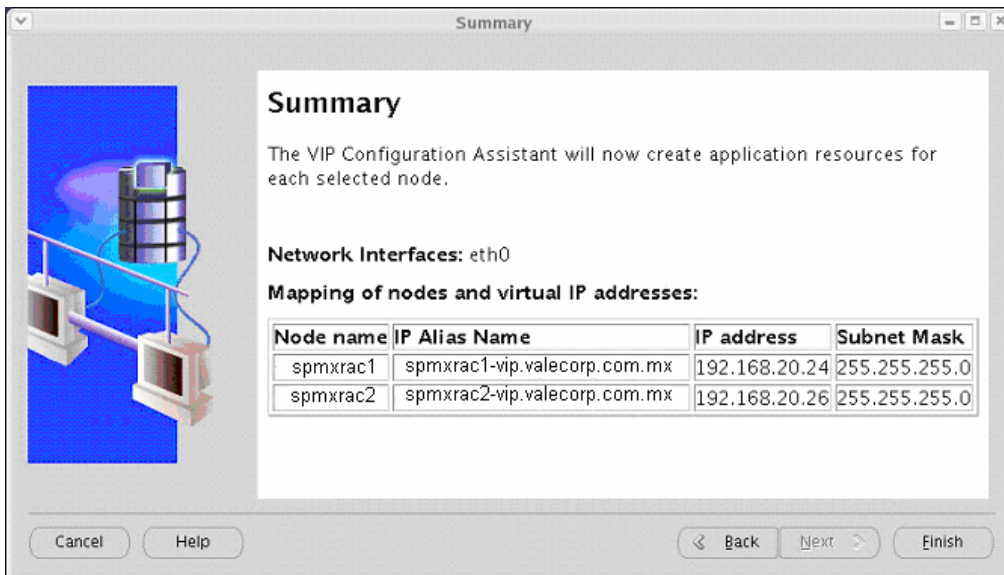




10. Seleccionamos las tarjetas de red virtuales.

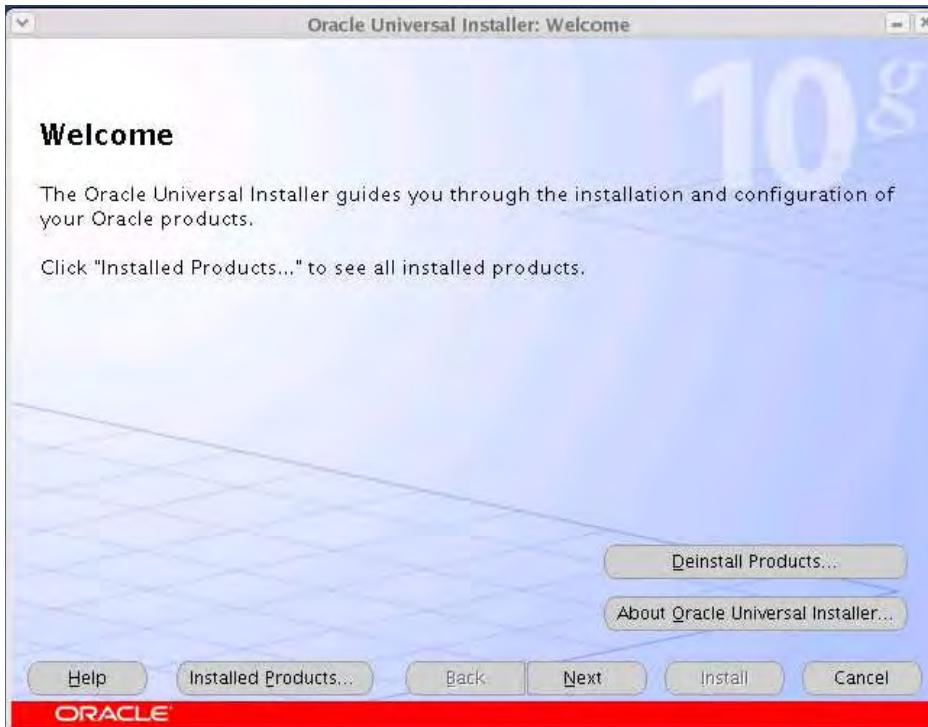


11. Resumen y fin de la instalación.



## 5. Instalación de software rdbms oracle 10g

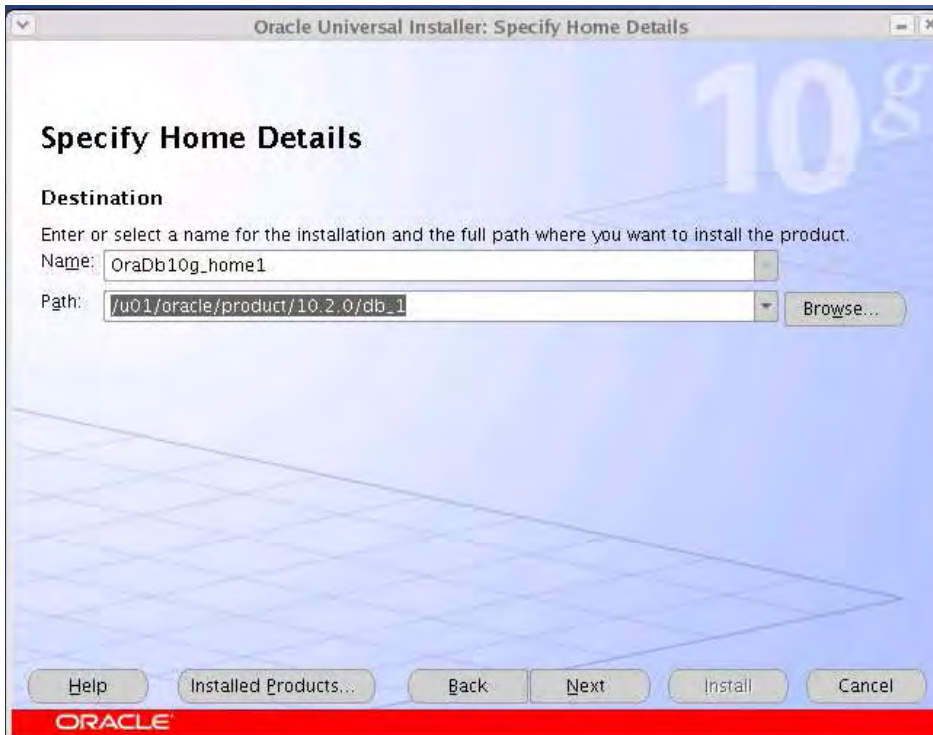
1. Descargar y desempacar el software de "Oracle 10g R2" en una carpeta con permisos para que el usuario Oracle del sistema operativo pueda tener acceso a ella. Ya teniendo el software listo, ejecutamos el comando: runInstaller, el cual iniciara el instalador de la aplicación.



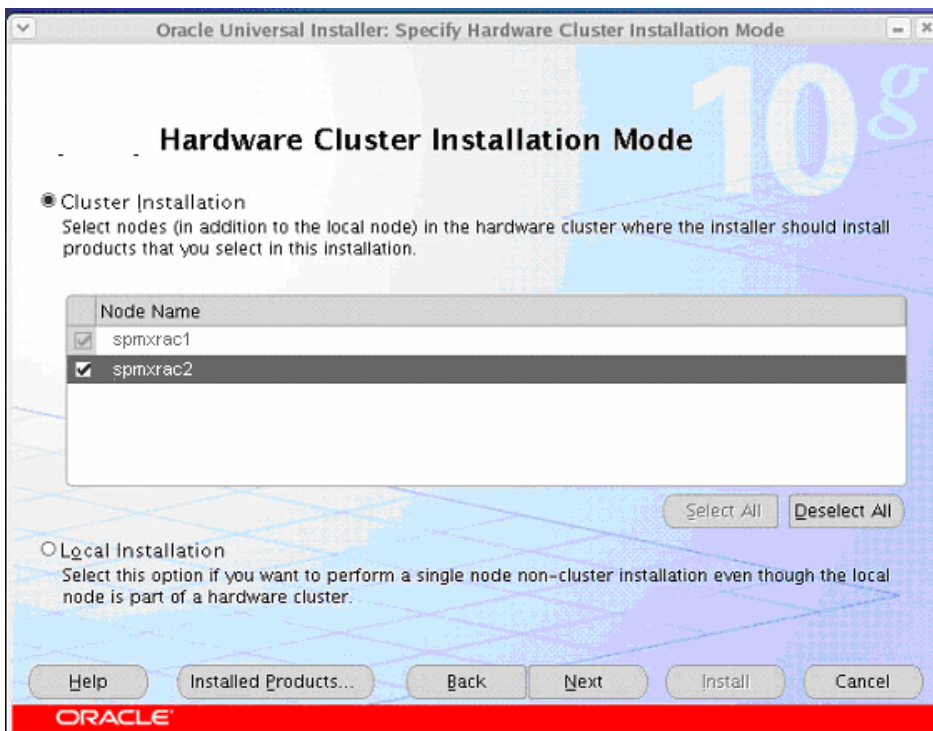
2. Seleccionamos el producto a instalar



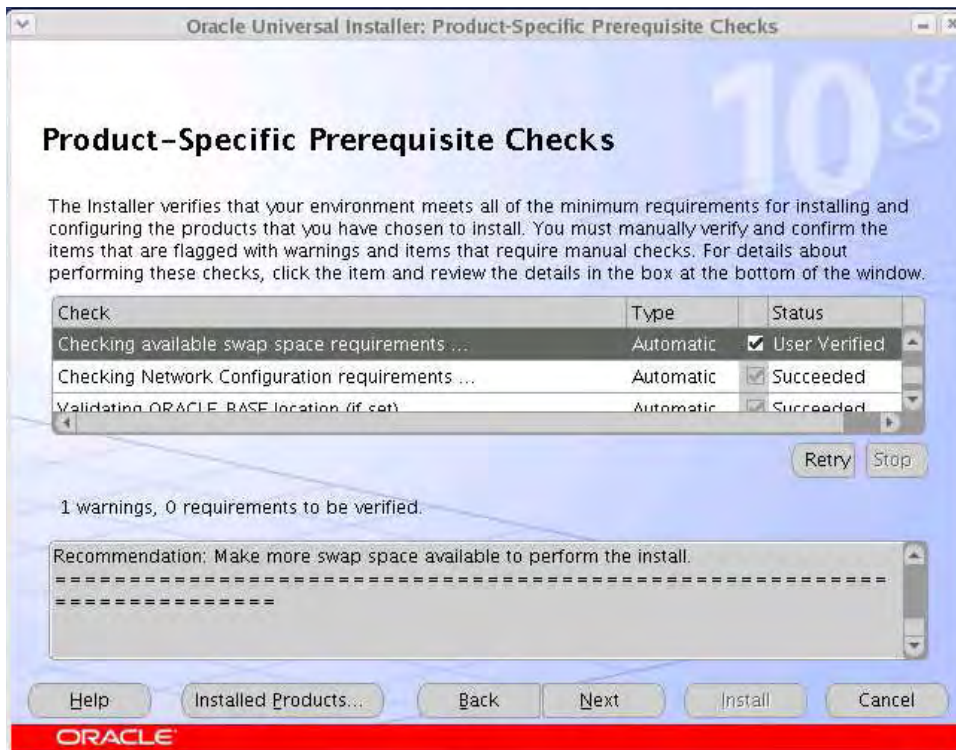
3. Seleccionamos la ubicación donde se instalara el software.



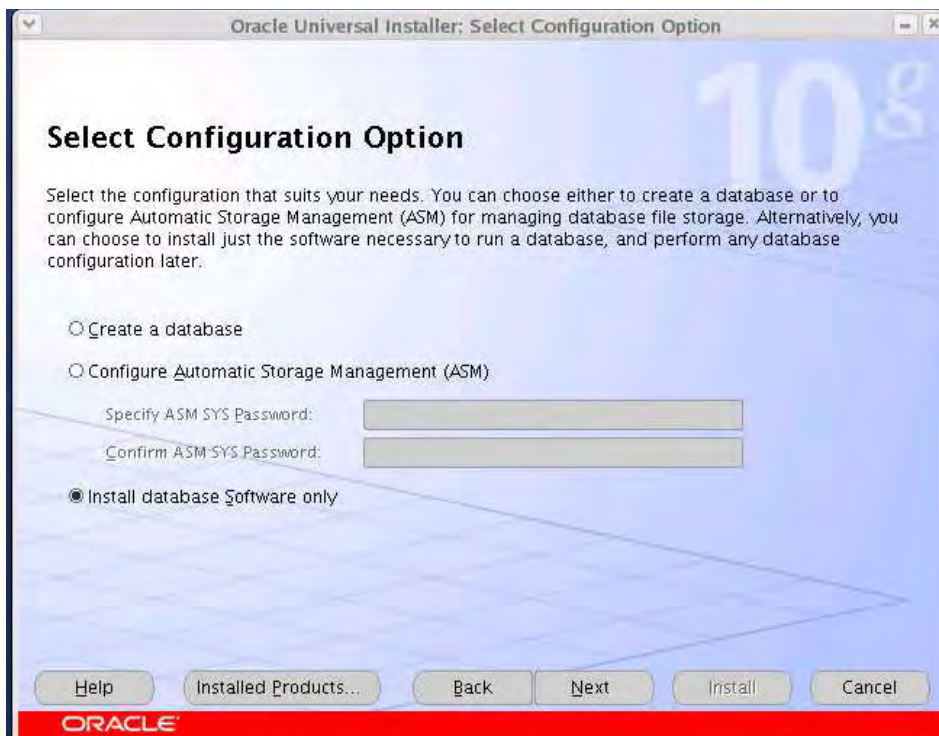
4. Seleccionamos los nodos que serán parte del cluster.



5. El asistente verifica que se cumplan los requerimientos mínimos del sistema, para poder continuar con la instalación.

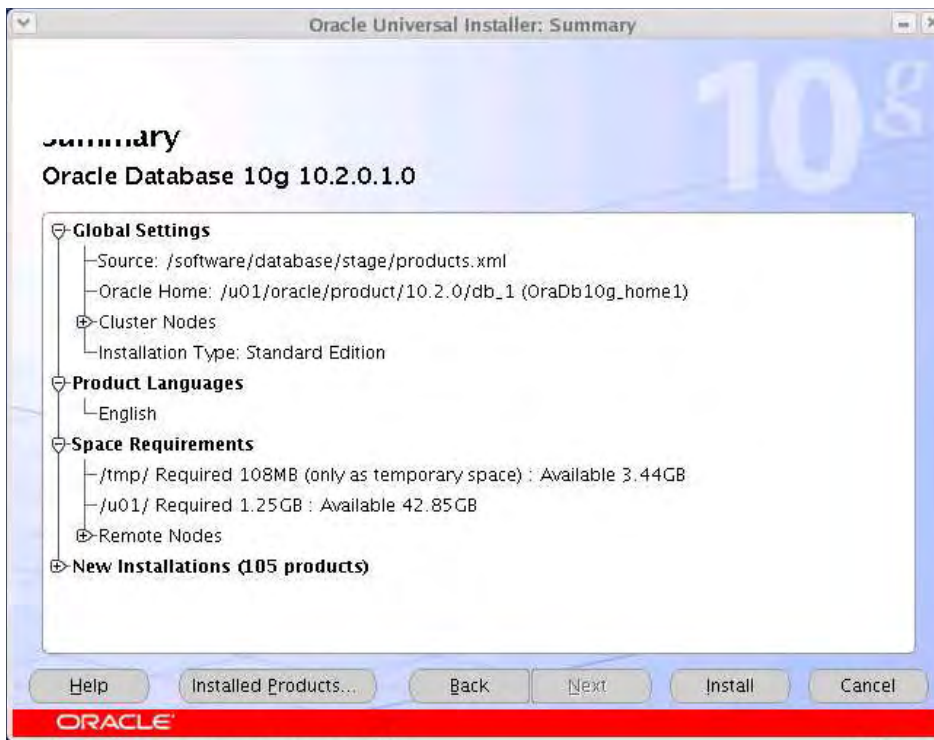


6. El asistente permite crear una base de datos en cuanto termina de instalar el software, para este ejercicio solo cargaremos el software, ya que las bases de datos las crearemos mas adelante.

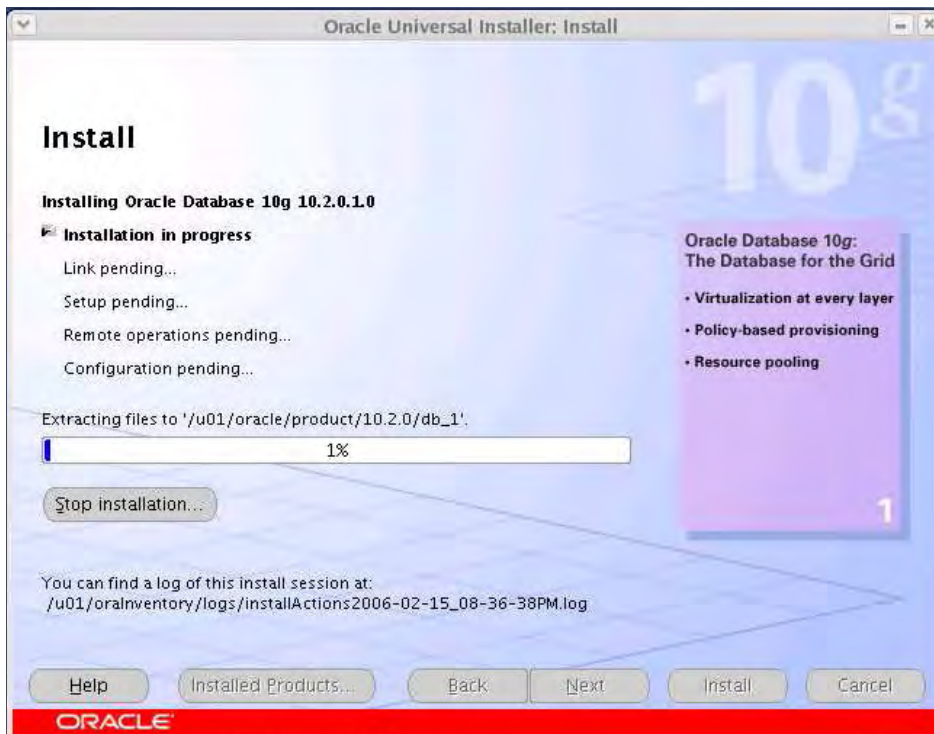




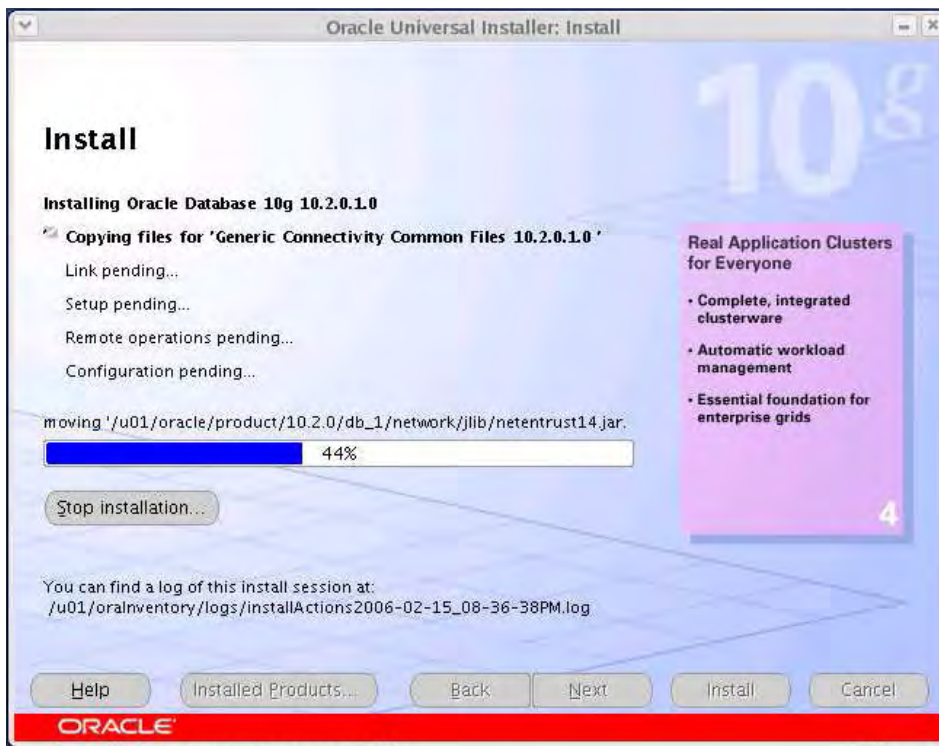
7. Se genera un resumen de la instalación y si no hay observaciones, se comienza la instalación.



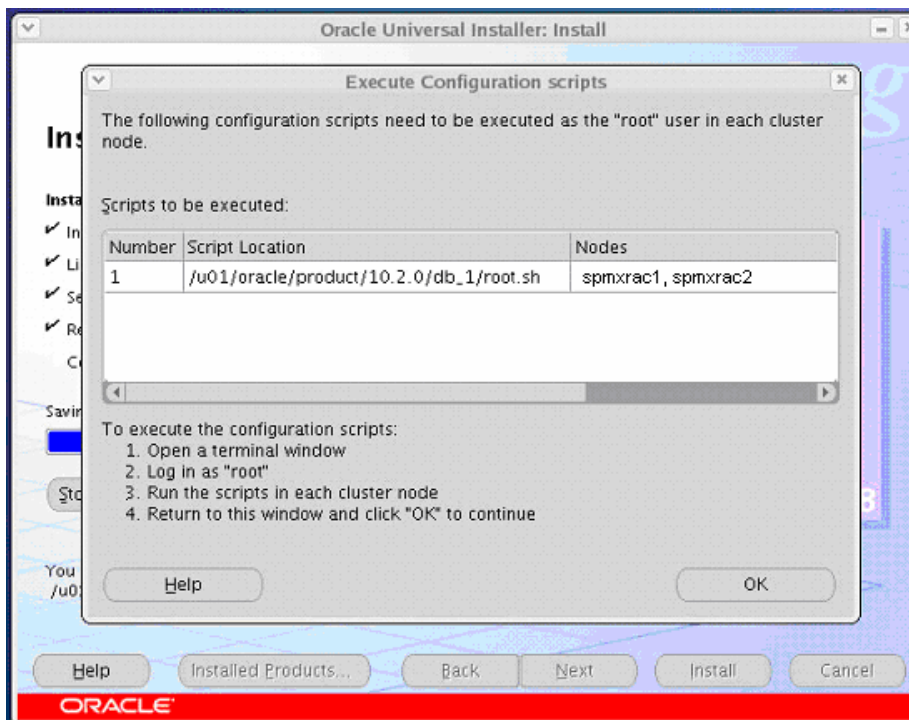
8. Progreso de la copia de archivos







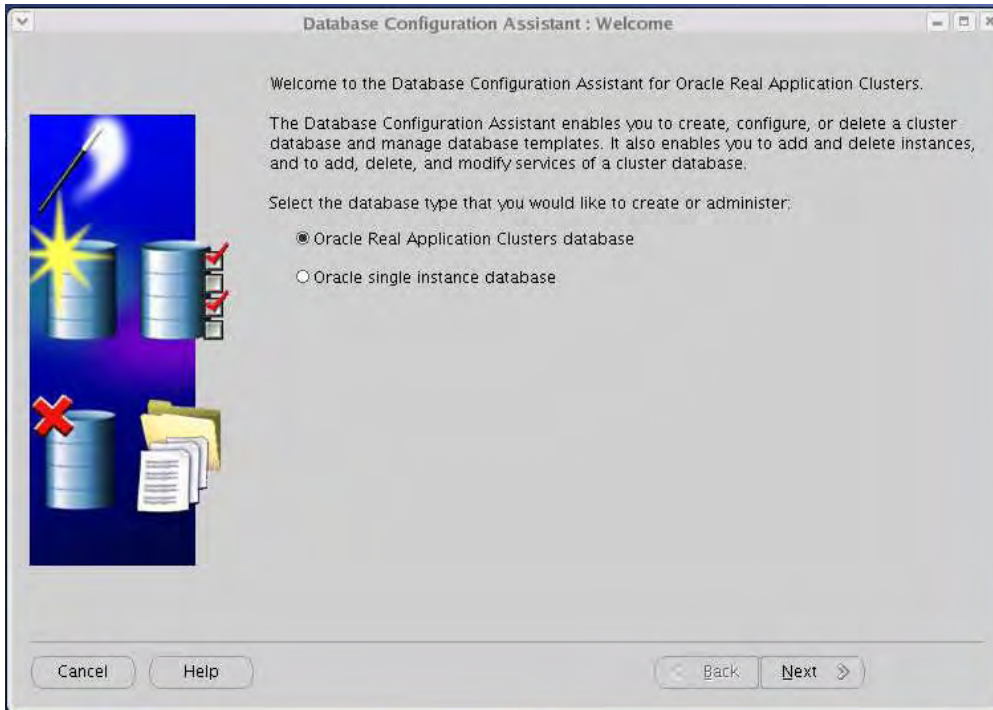
9. Al terminar el copiado de los binarios al disco duro, es necesario ejecutar un script para asignar permisos a Oracle, este script debe ser ejecutado por el usuario root.



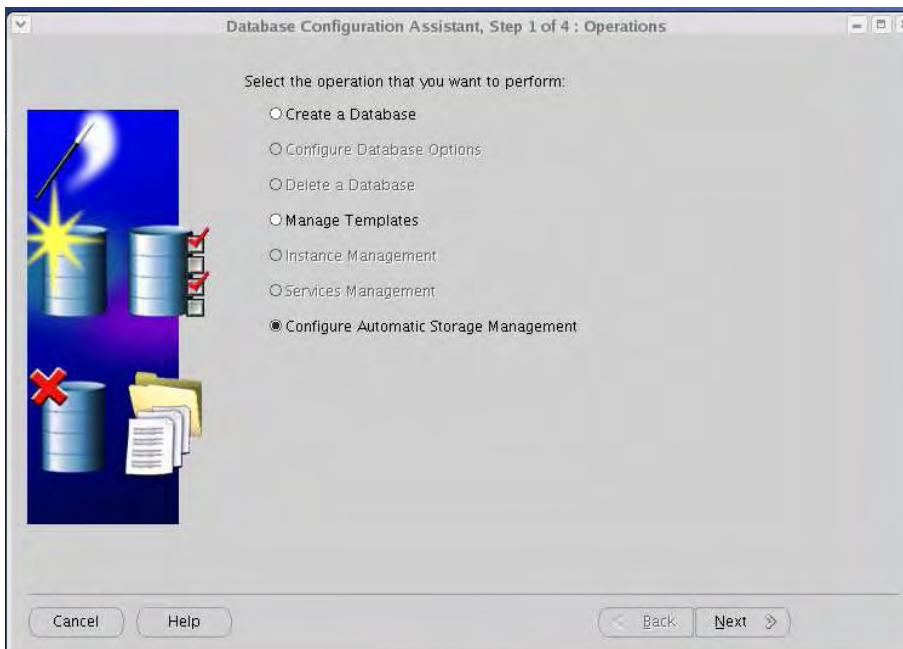
10. Si no hay problemas en la configuración, el asistente muestra el resumen de la instalación y configuración, dando fin a esta.

### Configuración ASM

1. Para la creación del ASM, es necesario ejecutar el asistente para crear bases de datos DBCA.



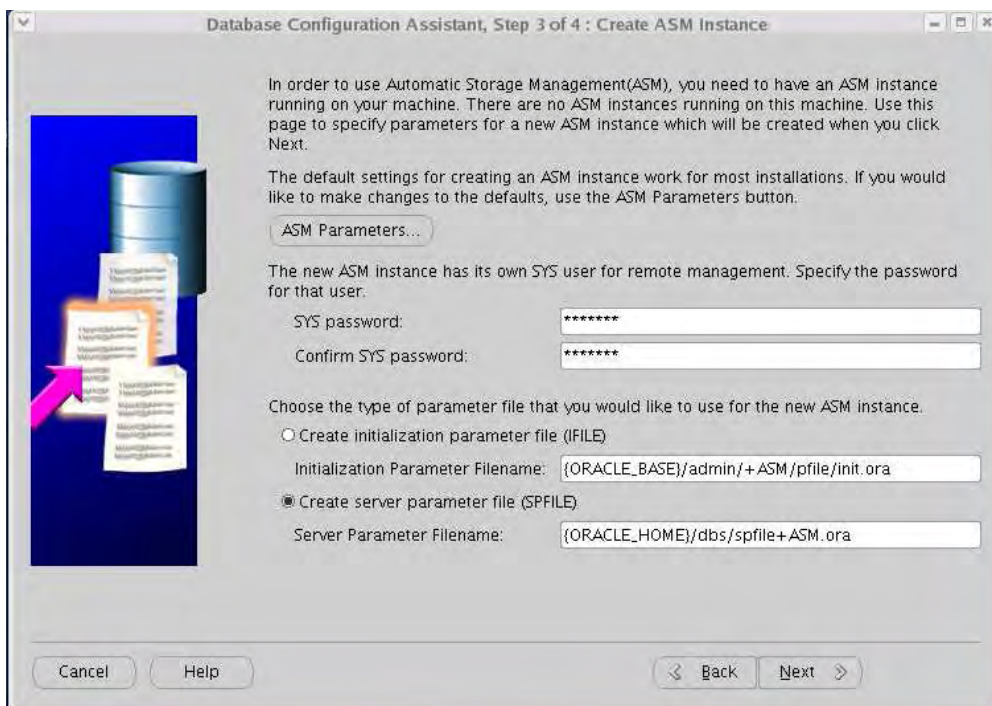
2. Seleccionamos la tercera opción disponible, con la cual estamos indicando al asistente que se va a configurar ASM



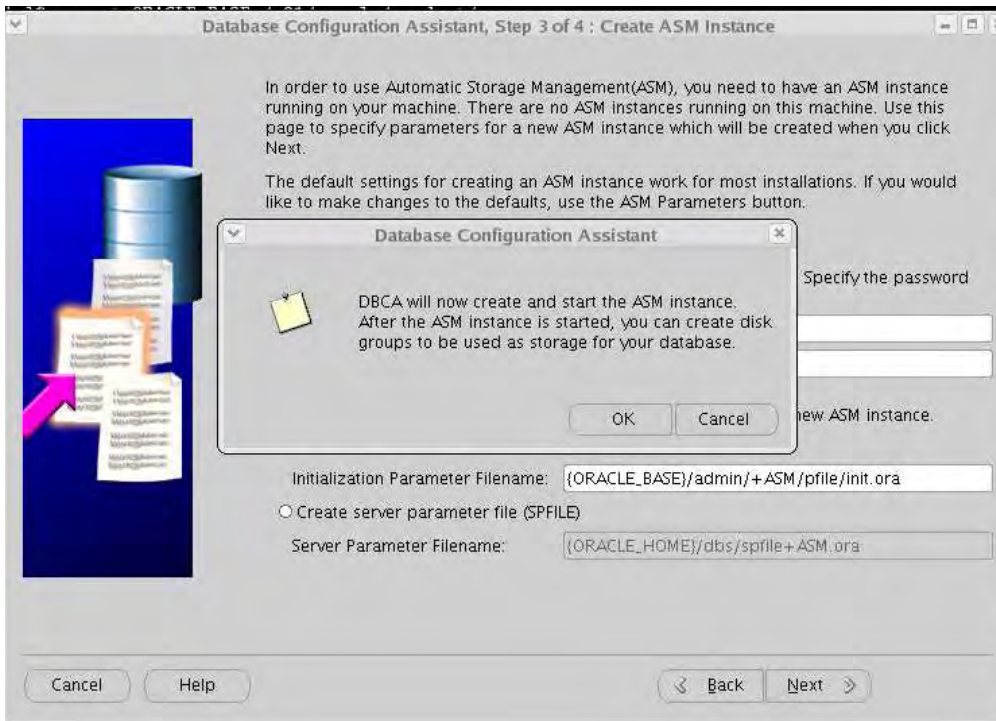
### 3. Seleccionamos los nodos que conforman el Cluster.



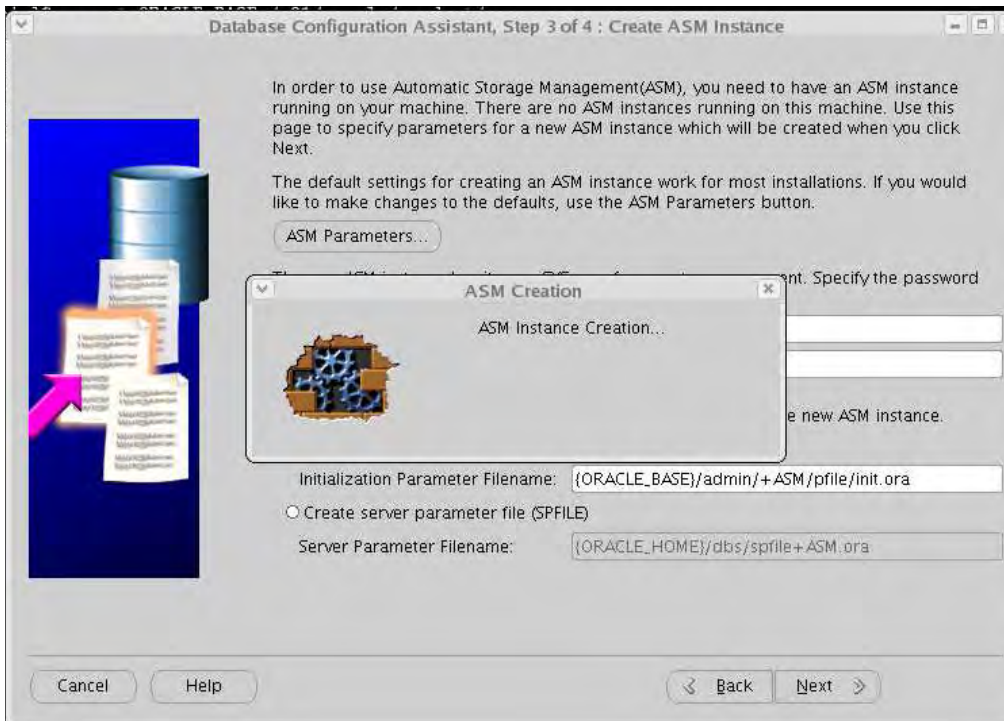
### 4. Especificamos el password que utilizar el usuario sys, y la ubicación del archivo de parámetros de la instancia.



5. El asistente muestra un aviso que comenzara con la creación de la instancia para ASM.

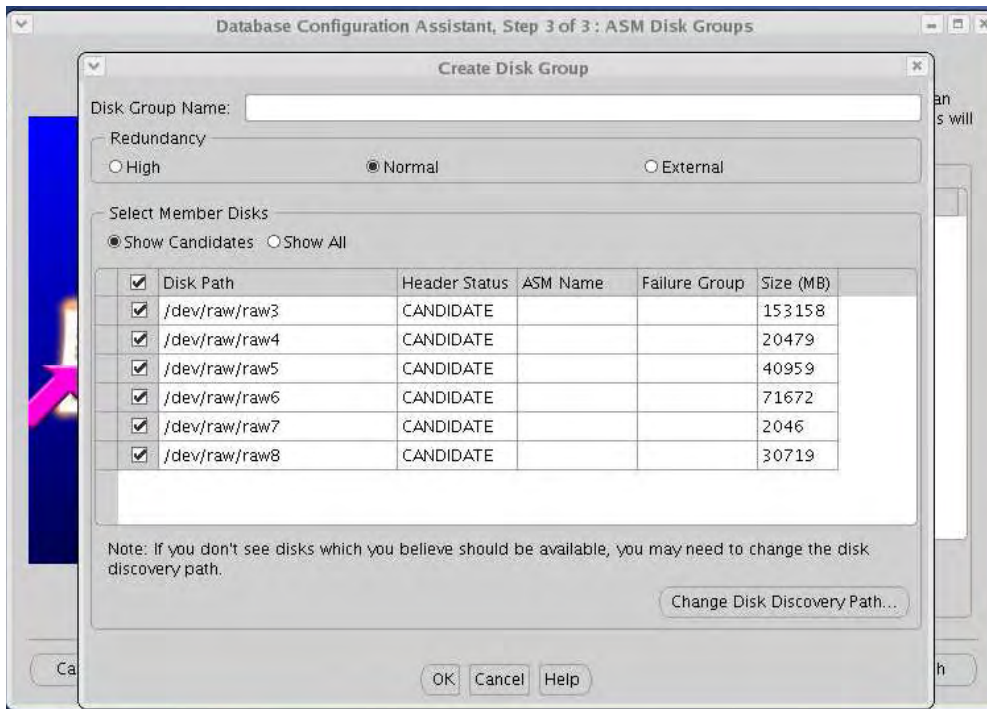


6. Progreso de la configuración.





7. Indicamos las particiones que se configuraran con ASM.





UNIVERSIDAD NACIONAL  
AUTÓNOMA DE  
MÉXICO

**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES  
CAMPUS ARAGÓN

**“DISEÑO Y CONFIGURACIÓN DE SERVICIOS CON  
ALTA DISPONIBILIDAD PARA SISTEMAS DE  
INFORMACIÓN OPERATIVOS CON TECNOLOGÍA  
ORACLE”**

**“Anexo 5.2, Implementación de Oracle Application Server en  
Cluster”**

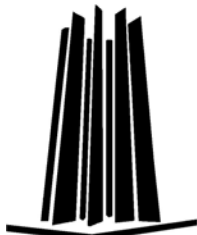
**T E S I S**  
QUE PARA OBTENER EL TÍTULO DE:  
INGENIERO EN COMPUTACIÓN

**P R E S E N T A:**

**OSWALDO VALENCIA LÓPEZ**

*ASESOR DE TESIS:*

*Ing. JOSÉ AGUSTÍN VARGAS PADILLA*



**MÉXICO, 2008.**

## **CAPITULADO**

- 1. Introducción**
- 2. Planteamiento del proyecto**
  - 2.1. Objetivo**
  - 2.2. Situación Actual**
  - 2.3. Alcance**
- 3. Implementación**
  - 3.1. Configuración de Sistema Operativo**
  - 3.2. Creación de Base de Datos “Metadata Repository” en RAC**
  - 3.3. Actualización de Base de Datos con OracleAS Metadata Repository Creation Assistant (REPCA)**
  - 3.4. Instalación de instancia para “Identity Management” y configuración de cluster, en NODO1**
  - 3.5. Instalación de instancia “Identity Management “ en Nodo 2 y unión a cluster NODO1**
  - 3.6. Instalación de Middle Tier en NODO 1**
  - 3.7. Instalación de Middle Tier en NODO 2**
  - 3.8. Configuración de Cluster Middle Tier**
  - 3.9. Instalación de Middle Tier J2EE en nodo1**
  - 3.10. Instalación de Middle Tier J2EE en nodo2**
  - 3.11. Configuración de Cluster J2EE**

## 1. Introducción

Uno de los objetivos del proyecto es la implementación de servidores de aplicaciones, con Oracle Application Server 10g R2, los cuales darán servicio a las aplicaciones REMESASREC y CREDITNOTE, dichas aplicaciones están diseñadas en J2EE y requieren un ambiente de Middle Tier J2EE, así como un servidor de reportes para dichas aplicaciones.

Se requieren dos ambientes diferentes, los cuales son llamados como Servidores de Aplicaciones Internos y Servidores de Aplicaciones Externos. Los servidores de Aplicaciones Internos darán servicio solamente a la Intranet de ValeCorp, y los Servidores de Aplicaciones Externos, darán servicio a los clientes de Internet.

El servidor de aplicaciones se divide en dos partes "Middle Tier" e "Infraestructura", la cual a su vez requiere de dos componentes importantes, "Metadata Repository" e "Identity Management".

Como necesidad del proyecto se requiere tener las instancias antes mencionadas en un ambiente de Cluster, con balanceo de cargas.

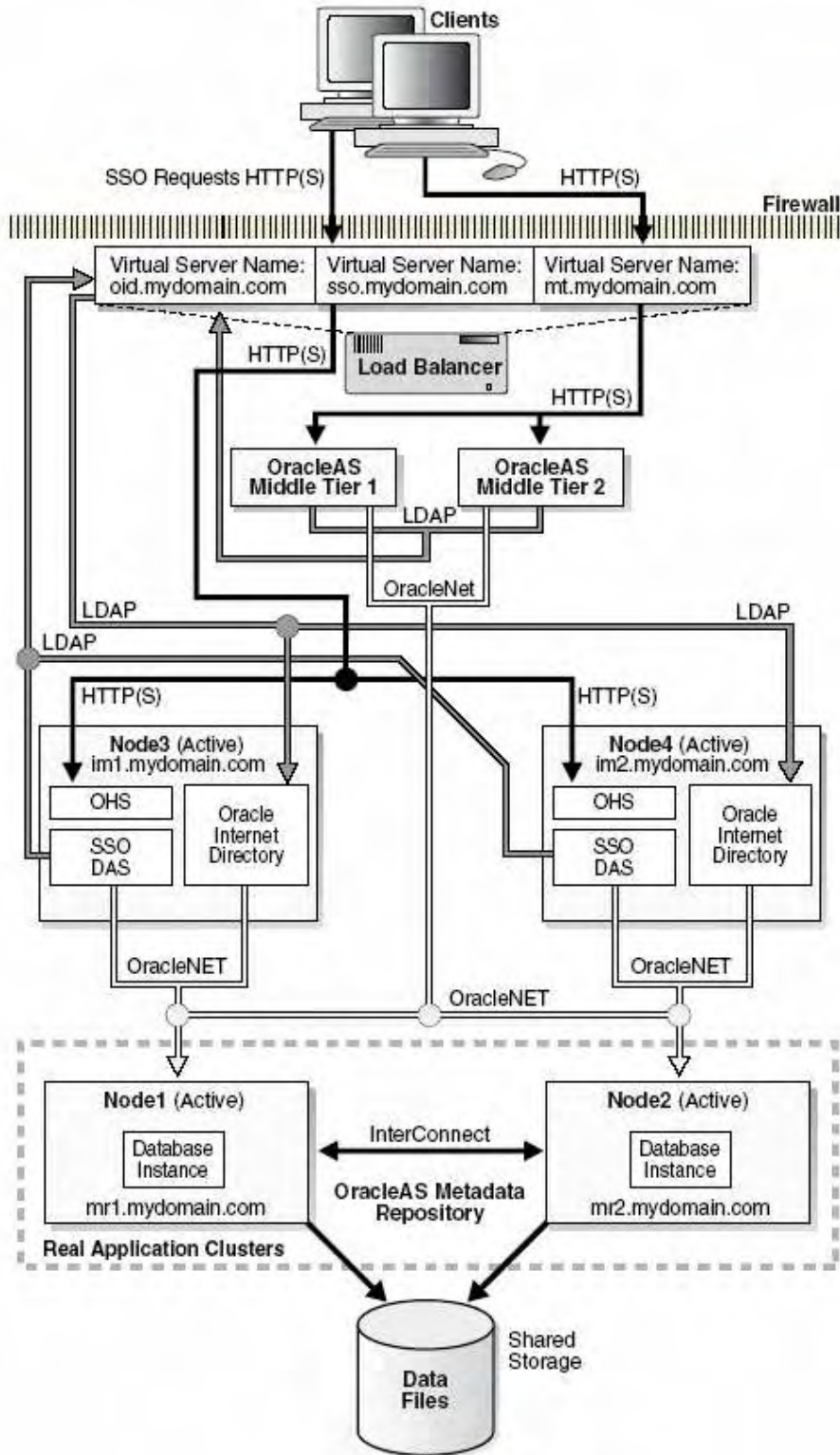
Por lo tanto se requiere tener:

- Una Base de Datos que se utilizara como "Metadata Repository"
- Una Instancia para que del servicio de "Identity Management"
- Una instancia para dar el servicio de las aplicaciones "Middle Tier (J2EE)"
- Una instancia para dar el servicio de "Middle Tier (Portal, Forms y Reports)"
- Un cluster lógico que contenga las instancias de "J2EE"
- Un cluster lógico que contenga las instancias de "Identity Management"
- Un Balanceador de Cargas, el cual será de la marca F5.

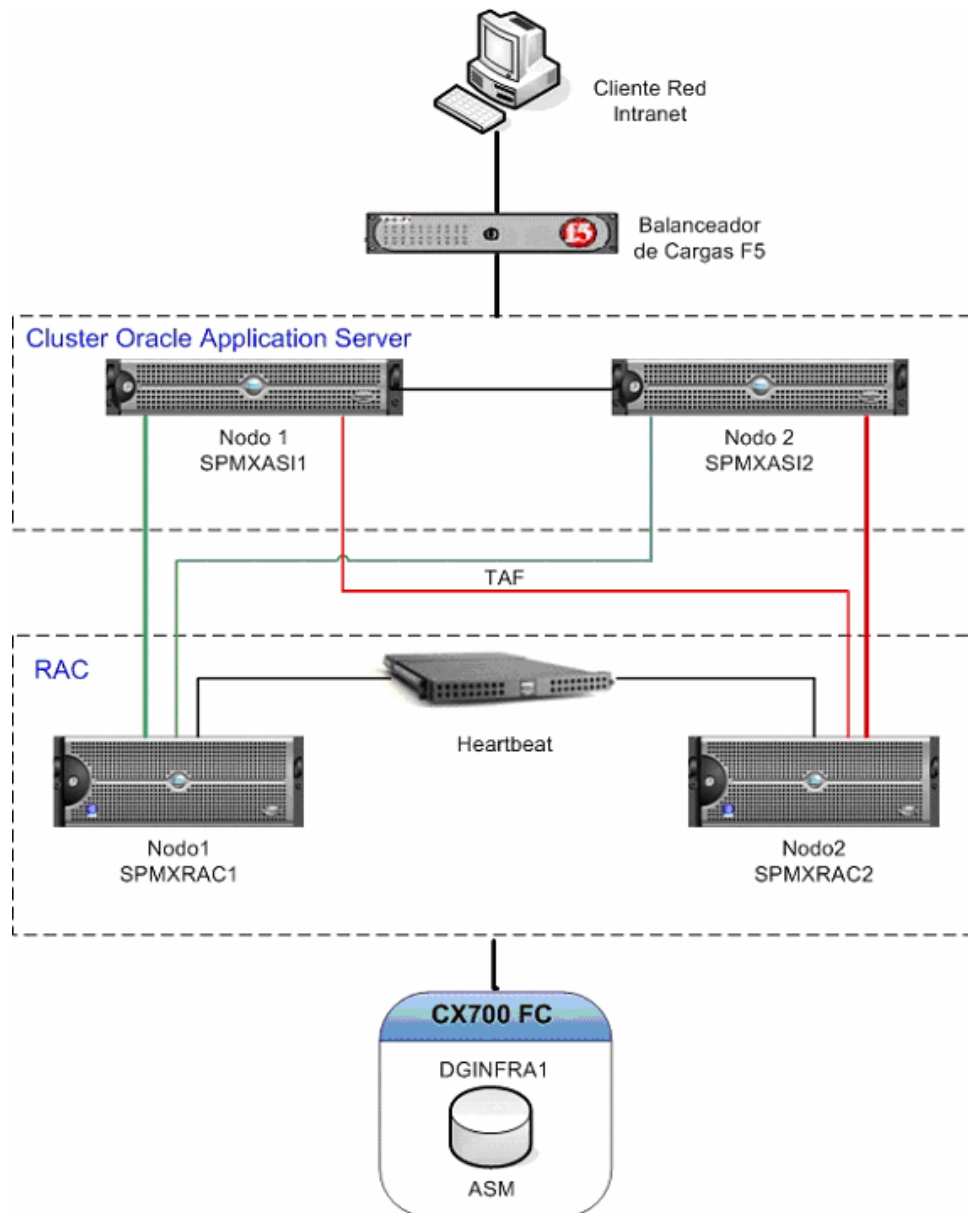
Las bases de Datos requeridas por los servidores de aplicaciones se crearan en un ambiente de RAC, previamente configurado.



### Topología de Cluster Oracle Application Server 10g.



## Servidores de Aplicaciones Internos.

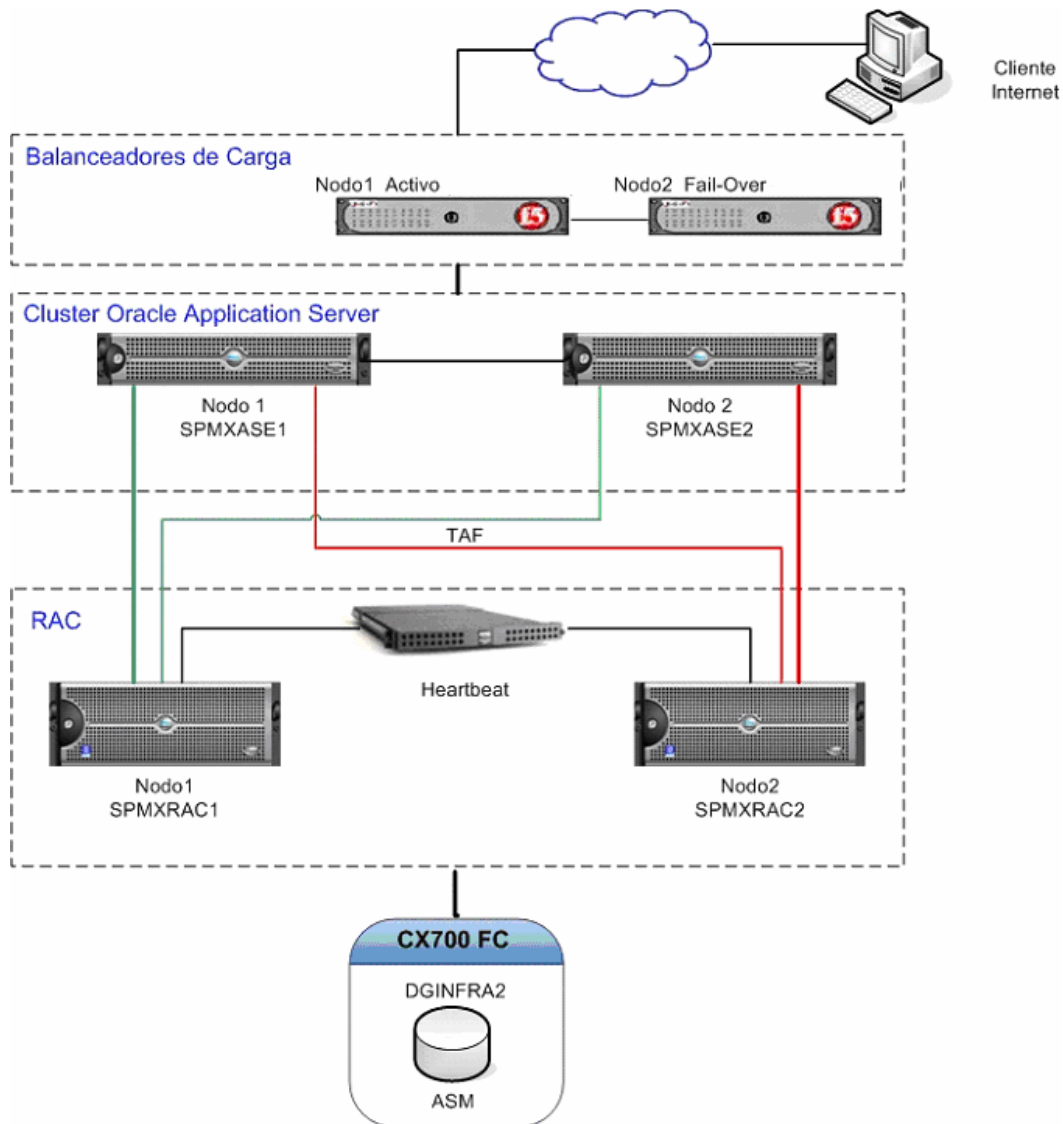


### Diagrama de servidores de aplicaciones Internos.

El ambiente de Servidores de Aplicaciones Internos, esta formado por dos servidores Dell con las mismas características, los cuales están en el segmento de la red privada de ValeCorp. Estos servidores cuentan con un sistema operativo Red Hat ES 4.

Dichos Servidores se encuentran conectados a un balanceador de Cargas F5, el cual recibirá todas las peticiones de los clientes y se encargara de balancear la carga de conexiones, hacia los servidores.

## Servidores de Aplicaciones Externos.



### Diagrama de servidores de aplicaciones Externos.

El ambiente de Servidores de Aplicaciones Externos, esta formado por dos servidores Dell con las mismas características, los cuales están en el segmento independiente de la red de ValeCorp, esta red solo esta formada por estos dos equipos y se encuentra administrada por un balanceador de Carga F5. Estos servidores cuentan con un sistema operativo Red Hat ES 4.

Estos servidores darán servicio a Internet por lo que tendrán la mayor carga de peticiones, por lo que se tendrán dos balanceadores redundantes los cuales se encargaran de recibir todas las peticiones y balancearlas.

Por seguridad los balanceadores F5 serán configurados de manera tal que solo puedan acceder a ellos por medio del balanceador.

NOTA: La implementación de los dos ambientes, tanto servidores internos como servidores externos es muy similar, por lo que solo se utilizarán las imágenes de la implementación de los servidores internos, haciendo notar las diferencias en el ambiente de los servidores externos.

## 2. Planteamiento del proyecto

### 2.1 Objetivo

Creación de Bases de Datos requeridas por los servidores de aplicaciones en RAC.

Instalación de software "Oracle Application Server 10g R2", creación de instancias requeridas para el servicio de las aplicaciones, configurar los Servidores Internos y Externos con ambiente de cluster, preparar servidores para que funcionen con el Balanceador F5.

### 2.2 Situación Actual

Actualmente las aplicaciones se encuentran funcionando de modo autónomo en diferentes servidores a los actuales, es decir los ambientes que se requiere instalar se instalarán en servidores nuevos.

Dichos servidores ya se encuentran configurados con el sistema operativo Red Hat ES 4, así como todas las configuraciones necesarias para poder trabajar en el.

Se cuenta con un ambiente de RAC funcionando, y con dos espacios de ASM reservados para los Servidores de Aplicaciones.

### 2.3 Alcance

Basándose en el objetivo planteado, el alcance es el siguiente:

- Configuración de Sistema Operativo
- Creación de Base de Datos en RAC.
- Actualización de Base de Datos de RAC, dando soporte a los servidores de aplicaciones
- Creación de Instancia "Identity Management" en Nodo 1
- Creación de Instancia "Identity Management" en Nodo 2
- Configuración de Cluster "Identity Management"
- Creación de Instancia "Middle Tier (Portal, Forms y Reports)" en Nodo 1
- Creación de Instancia "Middle Tier (Portal, Forms y Reports)" en Nodo 2
- Configuración de "Middle Tier (Portal, Forms y Reports)" en ambiente de cluster.
- Creación de Instancia "J2EE" en Nodo 1
- Creación de Instancia "J2EE" en Nodo 2
- Configuración de Cluster "J2EE"

### 3. Implementación

#### 3.1 Configuración de Sistema Operativo

Como requerimiento de la implementación de “Oracle Application Server 10g R2” en un ambiente de cluster requiere una configuración previa del sistema operativo. Dichas configuraciones son las siguientes:

##### Requerimientos de Hardware

Para poder llevar acabo la implementación se requiere revisar que se tenga el hardware necesario, así como la capacidad necesaria para un funcionamiento optimo de las aplicaciones de Oracle, los requerimientos mínimos son los siguientes:

- Tarjeta de Red con IP Fija
- Procesador 32 bits o mayor
- Velocidad de Procesador mayor a 450 Mhz
- Memoria RAM :
  - OracleAS Infrastructure:
    - OracleAS Metadata Repository and Identity Management: 1 GB
    - Identity Management only: 1 GB
    - OracleAS Metadata Repository only: 1 GB
  - Oracle Application Server middle tier:
    - J2EE and Web Cache: 512 MB
    - Portal and Wireless: 1 GB
    - Business Intelligence: 1 GB
    - OracleAS Developer Kits: 256 MB, but 512 MB recommended
  - Memoria SWAP, mayor a 1.5 GB
- Espacio en Disco:
  - OracleAS Infrastructure:
    - OracleAS Metadata Repository and Oracle Identity Management: 3.7 GB
    - Oracle Identity Management only: 1.3 GB
    - OracleAS Metadata Repository only: 3.7 GB
  - Oracle Application Server middle tier:
    - J2EE and Web Cache: 900 MB
    - Portal and Wireless: 1.2 GB
    - Business Intelligence and Forms: 2 GB
    - OracleAS Developer Kits: 710 MB
  - /TMP mayor a 400 Mb
- Video, 256 colores

##### Requerimientos de Software

Existen diferentes opciones para poder instalar el servidor de aplicaciones de Oracle, a continuación se listan los componentes que son compatibles con OAS, y los componentes de sistema operativo que se requieren para una instalación satisfactoria.

- Sistema Operativo:
  - Red Hat Enterprise Linux AS/ES 2.1
  - Red Hat Enterprise Linux AS/ES 3.0
  - Red Hat Enterprise Linux AS/ES 4.0
  - Suse Linux ES 8
  - Suse Linux ES 9
- Kernel:
  - kernel-2.6.9-11.EL
  - kernel-smp-2.6.9-11.EL
  - kernel-hugemem-2.6.9-11.EL
- Paquetes obligatorios:
  - glibc-2.3.4-2.9

- glibc-common-2.3.4-2.9
- binutils-2.15.92.0.2-13
- compat-libstdc++-296-2.96-132.7.2
- gcc-3.4.3-22.1
- gcc-c++-3.4.3-22.1
- libstdc++-3.4.3-22.1
- libstdc++-devel-3.4.3-22.1
- openmotif21-2.1.30-11.RHEL4.4
- pdksh-5.2.14-30
- setarch-1.6-1
- make-3.80-5
- gnome-libs-1.4.1.2.90-44.1
- sysstat-5.0.5-1
- compat-db-4.1.25-9
- control-center-2.8.0-12
- xscreensaver-4.18-5.rhel4.2
- Hostname definido en un servidor de DNS o en archivo /etc/hosts
- Browser:
  - Netscape
  - Mozilla
  - FireFox

### Parámetros de Kernel

Oracle requiere ciertos parámetros mínimos de Kernel para poder funcionar de forma correcta, los parámetros que se requieren son los siguientes:

Archivo: /etc/sysctl.conf

```
kernel.shmall = 2097152
kernel.shmmax = 2147483648
kernel.shmmni = 4096
# semaphores: semmsl, semmns, semopm, semmni
kernel.sem = 256 32000 100 142
fs.file-max = 131072
net.ipv4.ip_local_port_range = 10000 65000
kernel.msgmni = 2878
kernel.msgmax = 8192
kernel.msgmnb = 65535
```

Archivo: /etc/security/limits.conf

```
* soft nproc 2047
* hard nproc 16384
* soft nofile 2048
* hard nofile 65536
```

Archivo: /etc/pam.d/login

```
session required /lib/security/pam_limits.so
```

### Revisión de Puertos

Oracle Application Server 10g R2, utiliza una serie de puertos que pueden configurarse de forma automática basándose en los puertos disponibles que tiene el sistema operativo, o se pueden registrar los puertos de forma específica, para esto se requiere revisar que estén disponibles los puertos y reservar puertos para las siguientes aplicaciones.

- # J2EE and Web Cache  
Oracle HTTP Server port = port\_num

Oracle HTTP Server Listen port = port\_num  
Oracle HTTP Server SSL port = port\_num  
Oracle HTTP Server Listen (SSL) port = port\_num  
Oracle HTTP Server Diagnostic port = port\_num  
Java Object Cache port = port\_num  
DCM Java Object Cache port = port\_num  
DCM Discovery port = port\_num  
Oracle Notification Server Request port = port\_num  
Oracle Notification Server Local port = port\_num  
Oracle Notification Server Remote port = port\_num  
Application Server Control port = port\_num  
Application Server Control RMI port = port\_num  
Oracle Management Agent port = port\_num  
Web Cache HTTP Listen port = port\_num  
Web Cache HTTP Listen (SSL) port = port\_num  
Web Cache Administration port = port\_num  
Web Cache Invalidation port = port\_num  
Web Cache Statistics port = port\_num  
Log Loader port = port\_num  
ASG port = port\_num

- # Business Intelligence and Forms  
Reports Services SQL\*Net port = port\_num  
Reports Services discoveryService port = port\_num  
Reports Services bridge port = port\_num
- # Infrastructure  
Oracle Internet Directory port = port\_num  
Oracle Internet Directory (SSL) port = port\_num  
Oracle Certificate Authority SSL Server Authentication port = port\_num  
Oracle Certificate Authority SSL Mutual Authentication port = port\_num  
Ultra Search HTTP port number = port\_num

## Creación de Usuarios y Grupos requeridos por Oracle

Como todo producto de Oracle requiere tener un usuario dueño de las aplicaciones y un grupo al cual pertenezca dicho usuario.

- Usuario de Sistema Operativo: oracle
- Password de Usuario Oracle: oracle
- Grupo principal de usuario Oracle: oinstall
- Grupo secundario de usuario Oracle: dba

## VARIABLES DE AMBIENTE

Las variables de ambiente, no tienen que estar registradas en la instalación, pero al terminar de instalar, se requiere asignar las variables siguientes:

- \$ORACLE\_HOME
- \$ORACLE\_SID
- \$PATH
- \$LD\_LIBRARY\_PATH
- \$DISPLAY
- \$TMP
- \$TNS\_ADMIN
- \$ORA\_NLS
- \$LS\_BIND\_NOW

## **Software de instalación**

La instalación Oracle Application Server 10g R2 requiere que se instale desde el disco duro, por lo que se creara una carpeta en donde se copiaran y desempacaran los discos de OAS y REPCA.

## **Configuración de SSH de forma confiable**

Desde que se instalo el sistema operativo se configuro con el servicio de ssh, el cual sirve para conectarse a otro equipo de forma remota, pero cada vez que se hace una conexión a un equipo remoto, se requiere autenticar por medio de un password, como necesidad de Oracle Application Server, se requiere que los servidores, Nodo 1 y el Nodo 2 se puedan comunicar entre ellos sin necesidad de un password.

## **Configuración de Resolución de Nombres.**

Un requerimiento muy importante es que los servidores se puedan comunicar por medio de nombres y no de IP, por lo que se requiere configurar un DNS para que registre los nombres de todos los equipos, o en su defecto configurar el archivo /etc/hosts, par que contenga toda la información de los servidores de aplicaciones, virtual hosts de los balanceadores y el RAC. Este archivo se debe de configurar en todos los servidores.

## **Sincronización de tiempo entre ambos servidores.**

Las instancias en cluster de Oracle Application Server requieren estar sincronizadas, y utilizan el reloj del sistema, es por eso que se requiere tener sincronizadas las horas en cada Nodo.

## **3.2 Creación de Base de Datos “Metadata Repository” en RAC**

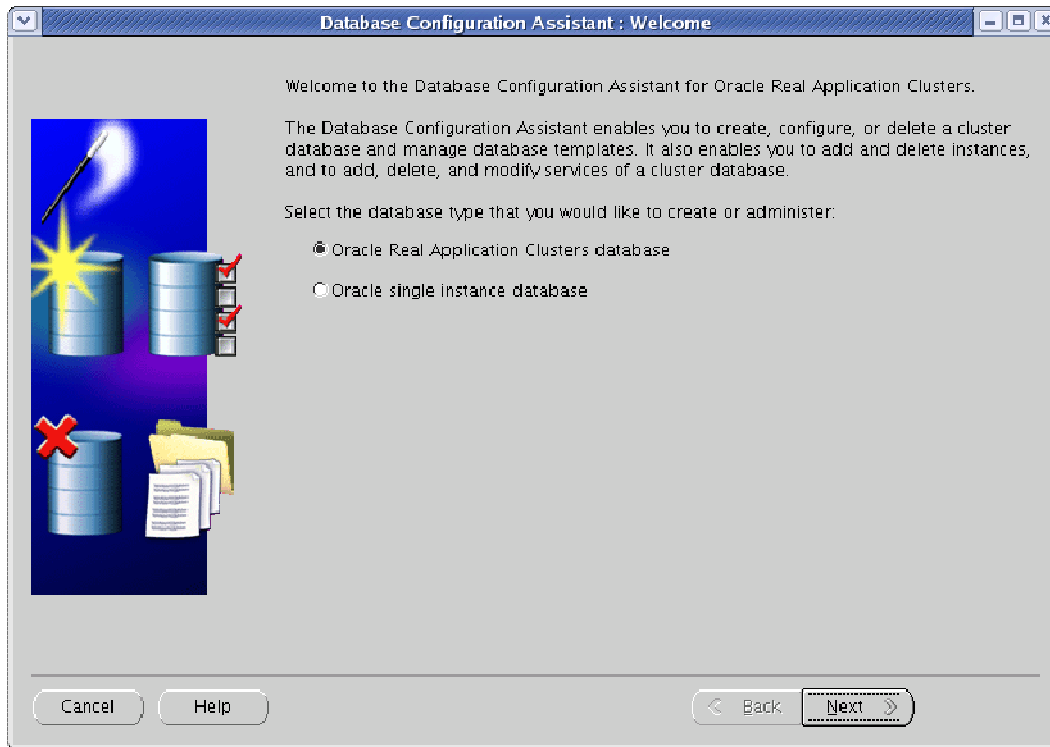
Como se menciona anteriormente, Oracle “Application Server 10g R2”, requiere una base de datos con los componentes necesarios para poder tener la funcionalidad de Repositorio de Metadatos, cabe mencionar que la creación de la base de datos es de forma personalizada y los parámetros que se personalizan son los recomendados por Oracle, dicha información se puede revisar en el manual de REPCA.

Se tiene reservado un grupo de ASM, para cada base de datos requerida por Oracle Application Server 10g R2. Las bases de Datos se configuraran en un ambiente de RAC, por lo que la creación de la base de datos se ejecutara desde el servidor SPMXRAC1 y SPMXRAC2

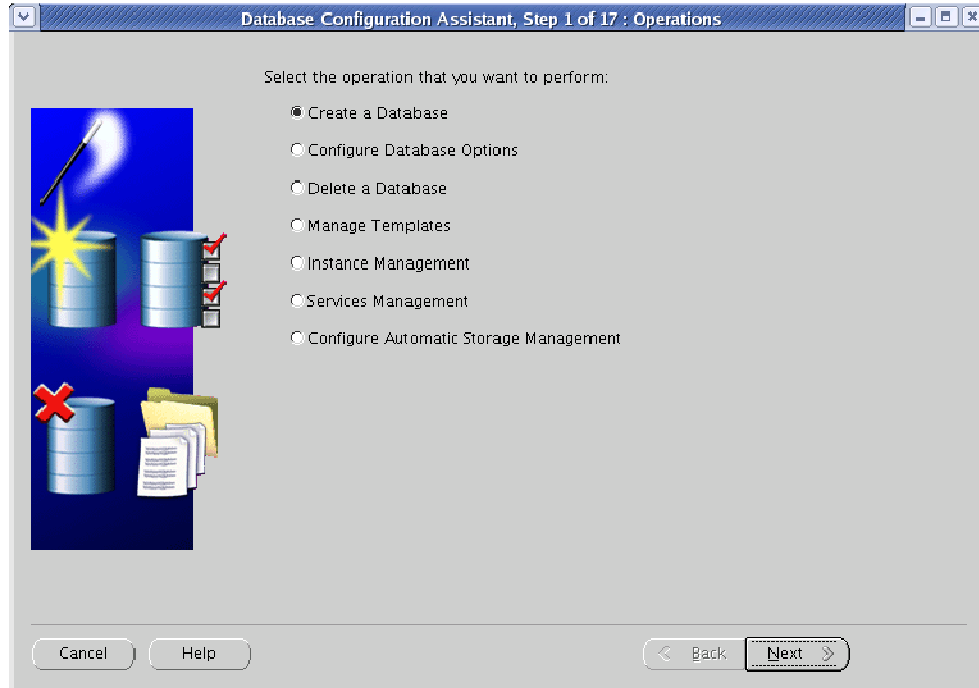
El procedimiento para crear la base de datos para los servidores de aplicaciones internos es el siguiente:

1. Se inicia una sesión del usuario oracle en SPMXRAC1 y desde una terminal, ejecutamos el comando: dbca con el cual se iniciara el asistente de creación de bases de datos. Desde el que se inicia el asistente, comienza a pedir información para poder configurar la base de datos, lo primero que se hace es especificar que tipo de base de datos se va a crear.

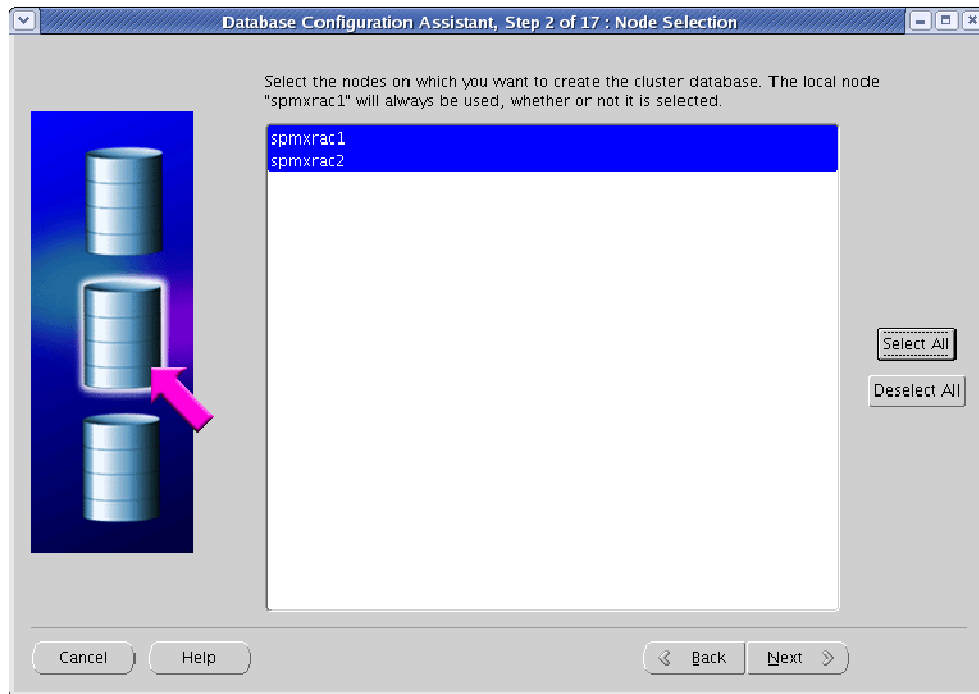




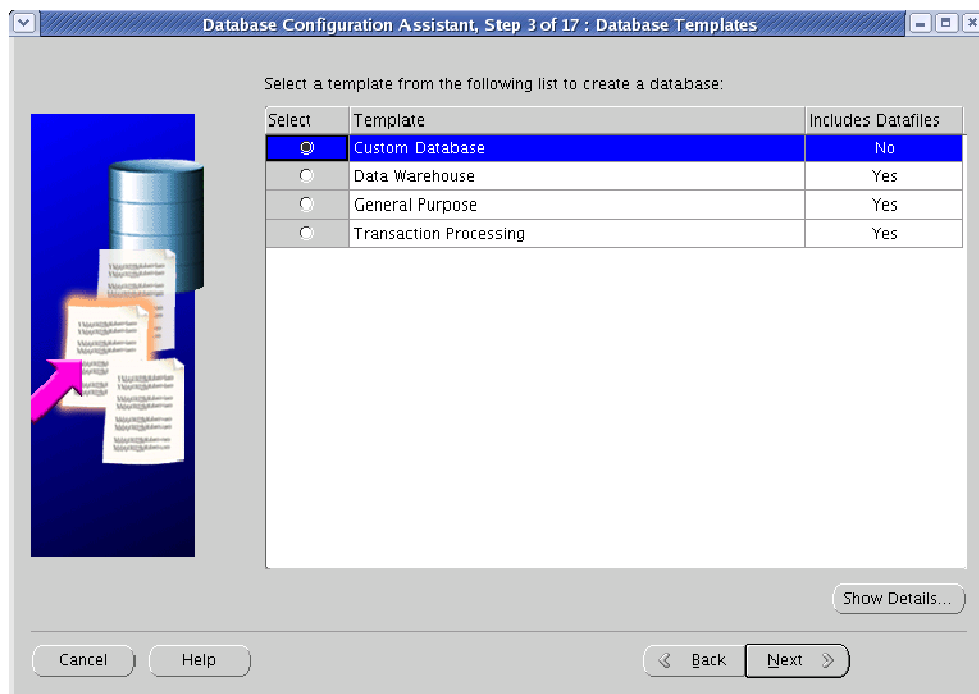
2. Especificamos la operación que se requiere, para este caso especificamos la creación de una nueva base de datos.



3. Seleccionamos los nodos del RAC que contendrán la base de datos.

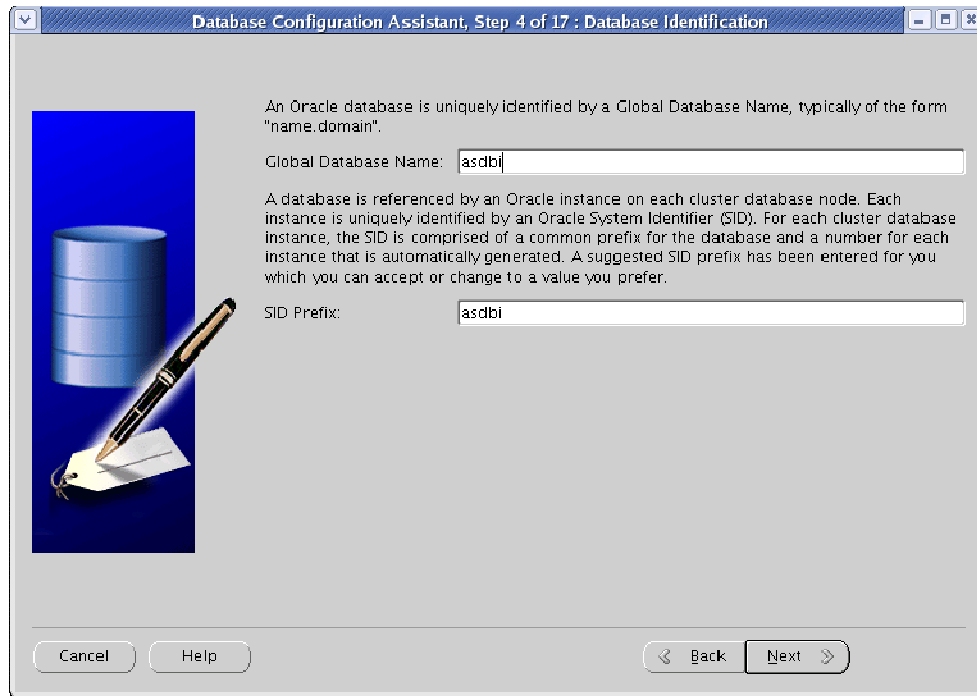


4. El asistente DBCA, utiliza plantillas para diferentes tipos de bases de datos, para el caso de Oracle Application Server, crearemos una base de datos personalizada, ya que se requieren modificar algunos parámetros.

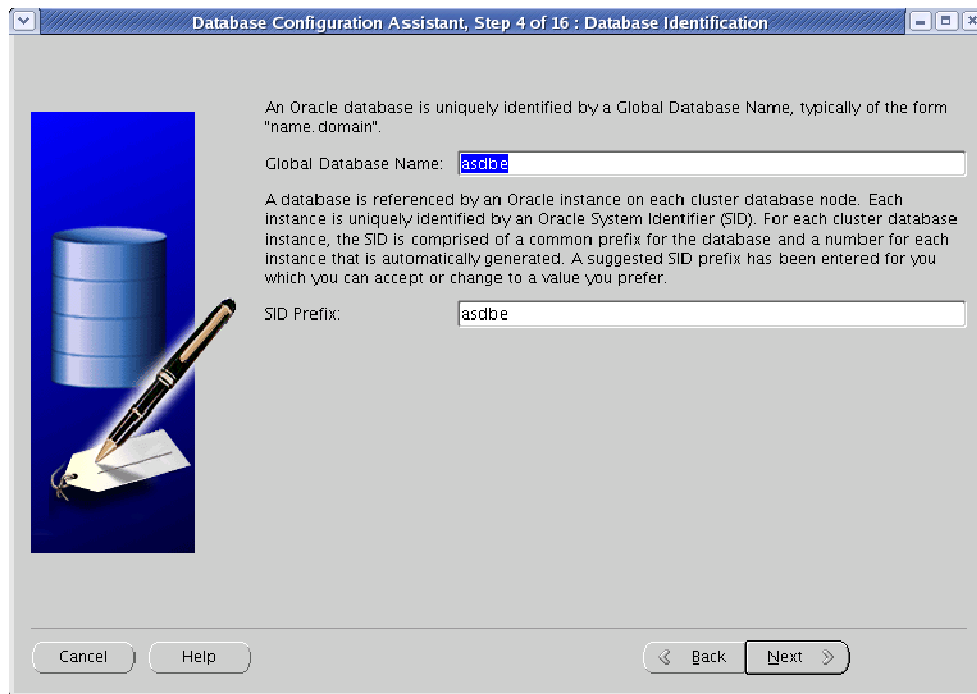


5. Especificamos el nombre de la base de datos, y el SID.

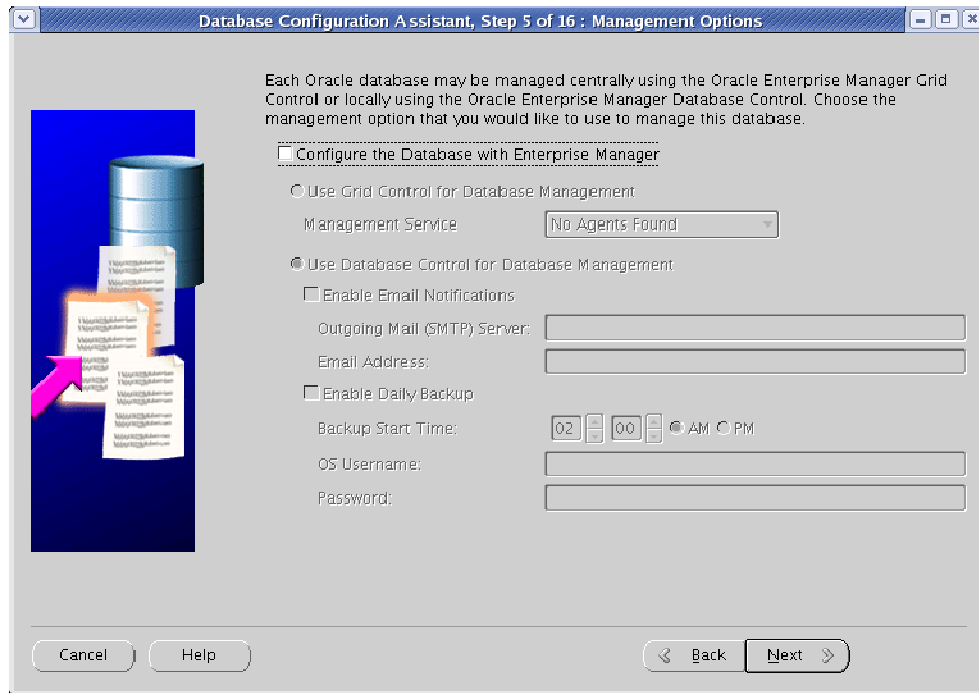
a) Base de datos, para servidores de aplicaciones Internos.



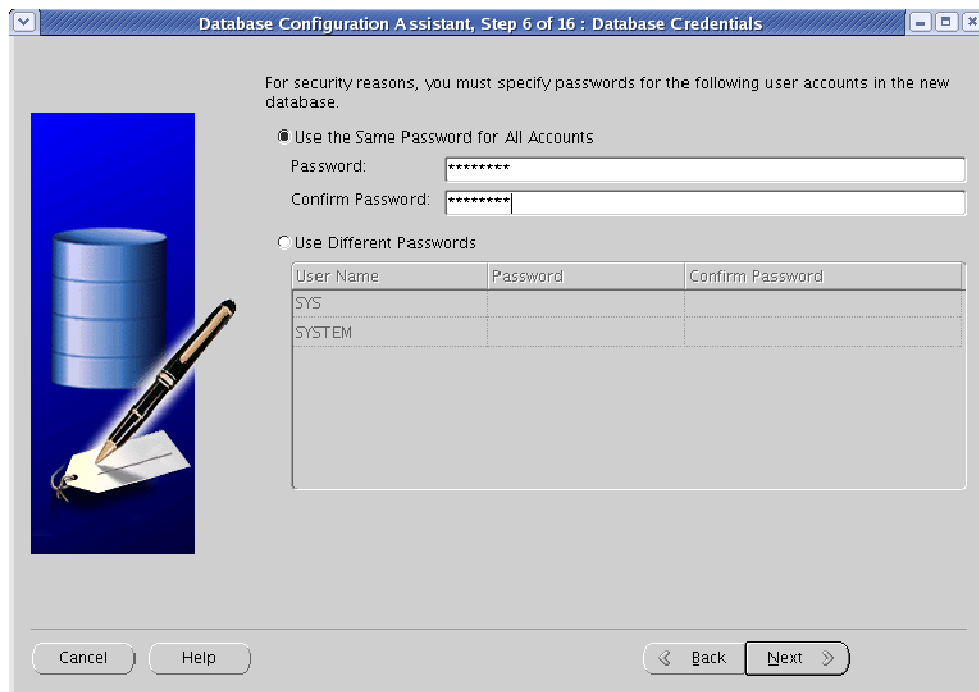
b) Base de datos, para servidores de aplicaciones Externos.



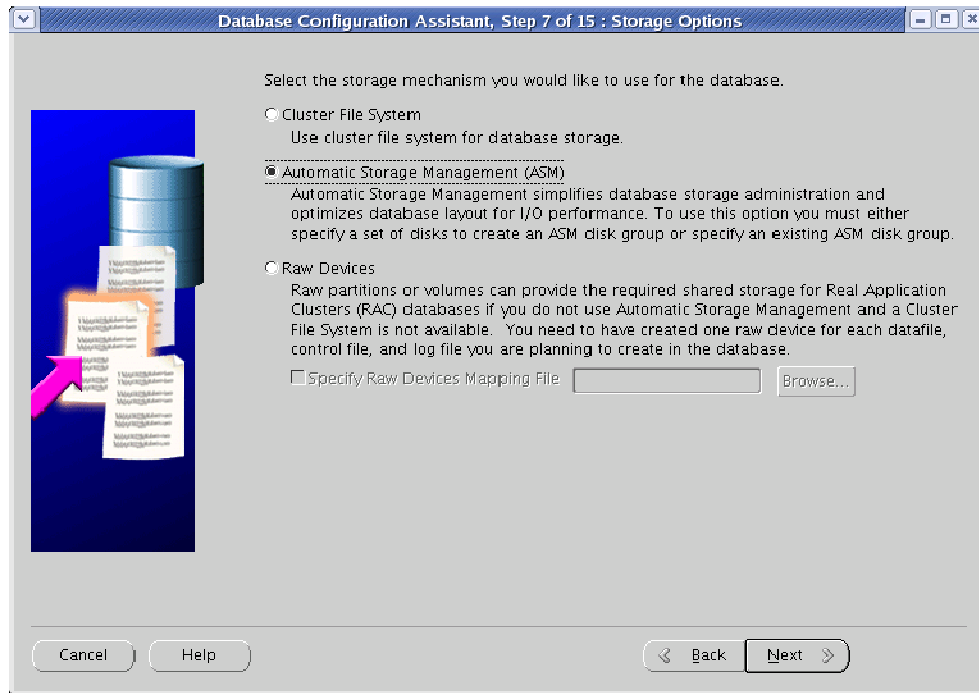
6. Con el asistente se puede configurar el administrador gráfico de la base de datos, para este caso no es necesario ocuparlo, por lo tanto no se configurará.



7. Especificamos un password común para las cuentas de SYS y SYSTEM.

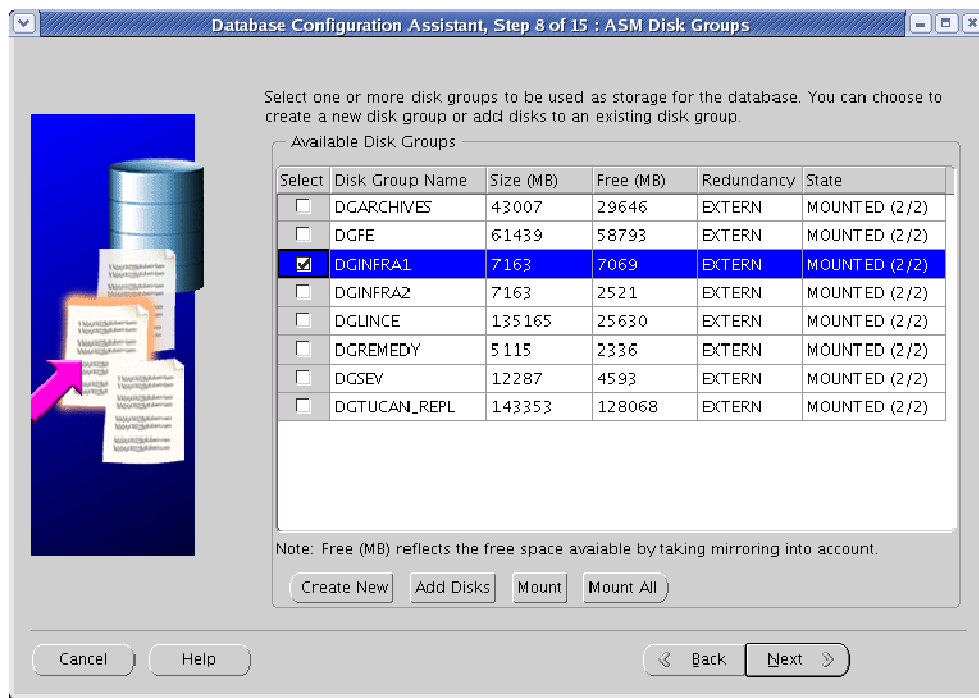


8. Como se menciona anteriormente, el almacenamiento de las bases de datos es por medio de ASM, por lo tanto seleccionamos la opción ASM

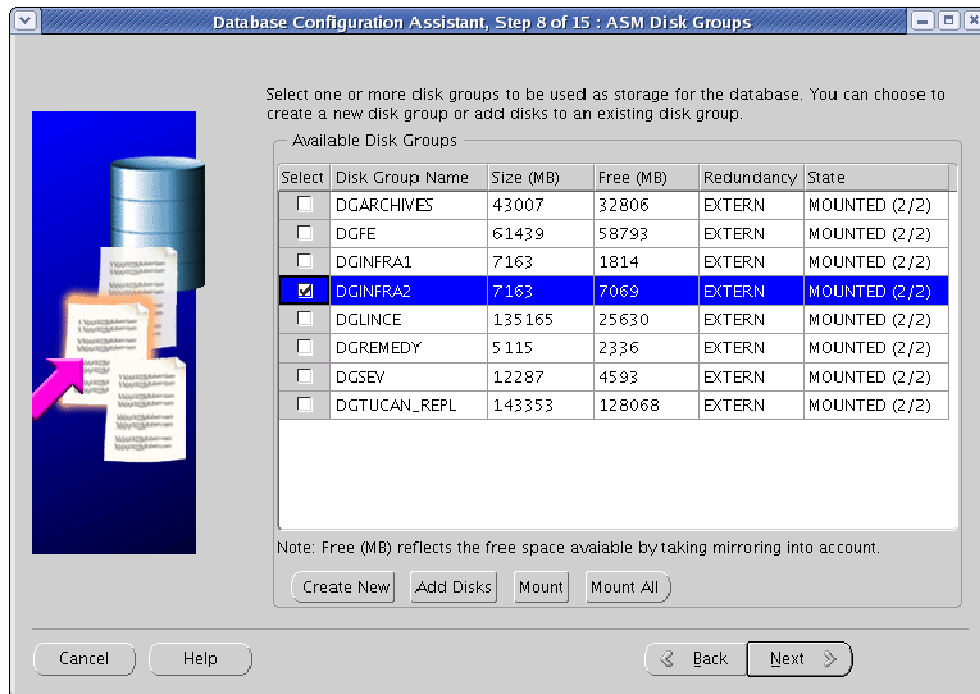


9. Existen dos grupos reservados para los servidores de aplicaciones estos son DGINFRA1 y DGINFRA2.

a) Grupo DGINFRA1, para servidores de aplicaciones Internos.

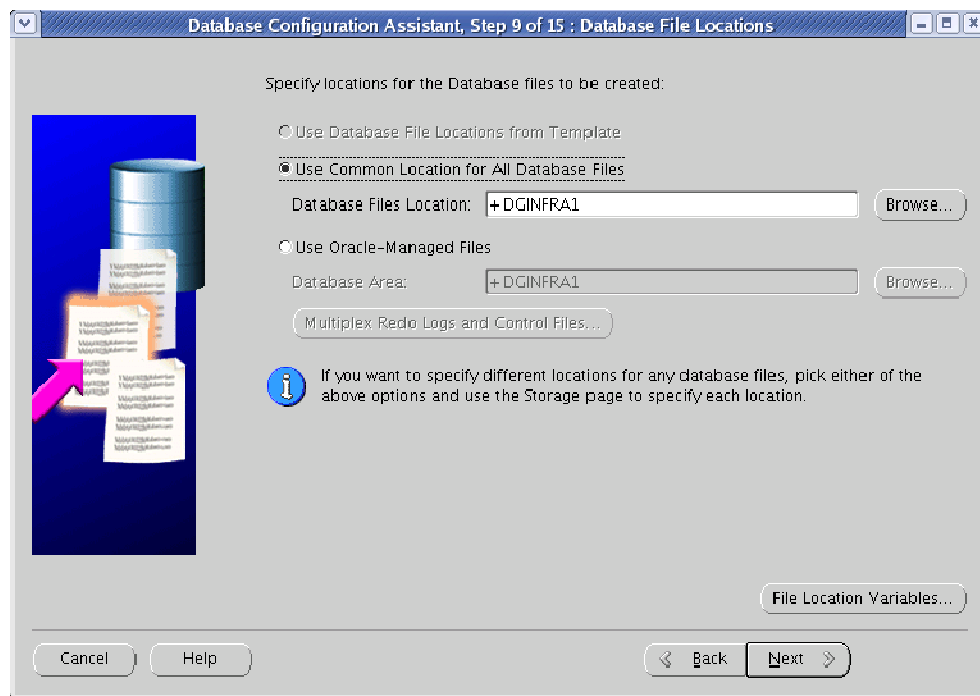


b) Grupo DGINFRA2, para servidores de aplicaciones Externos.

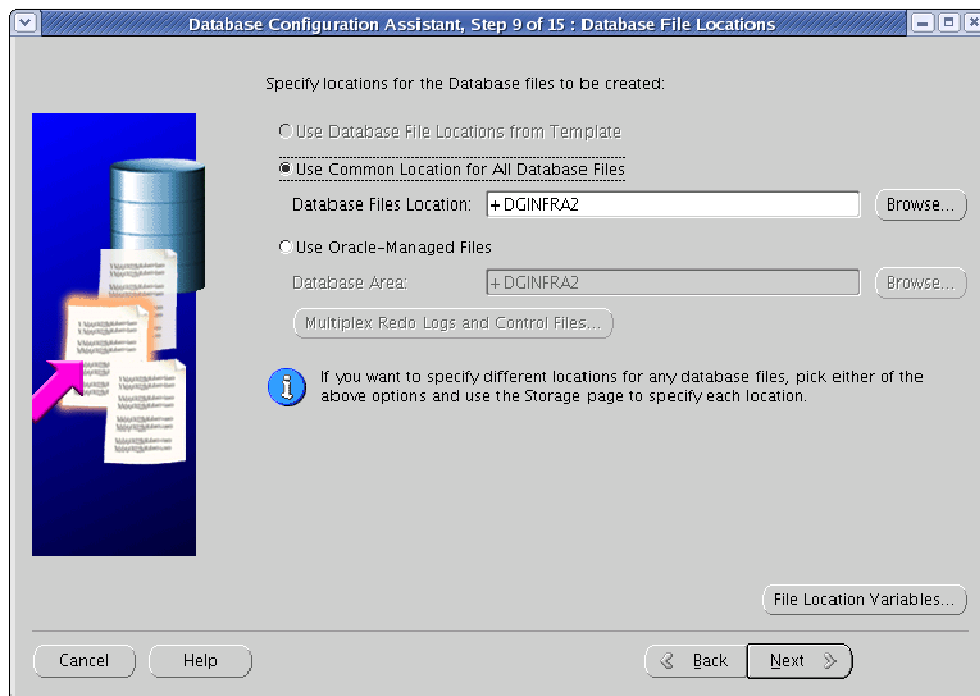


10. Como todas las bases de datos Oracle, utiliza diferentes archivos y diferentes rutas para su almacenamiento, para este caso especificaremos que se administren todos los archivos en una misma ruta.

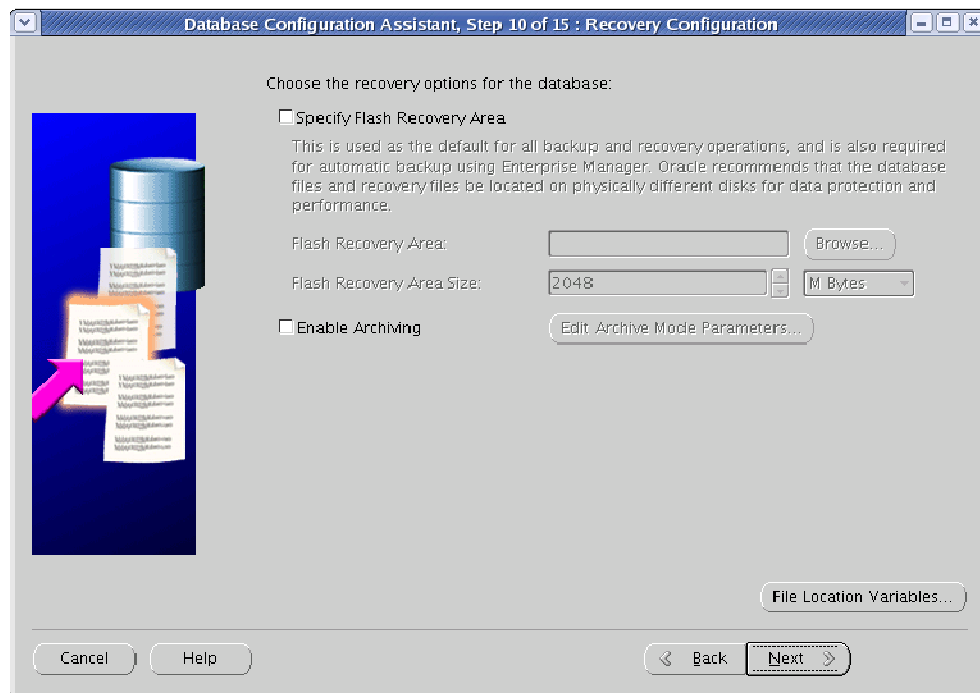
a) Rutas en Grupo DGINFRA1, para servidores de aplicaciones Internos.



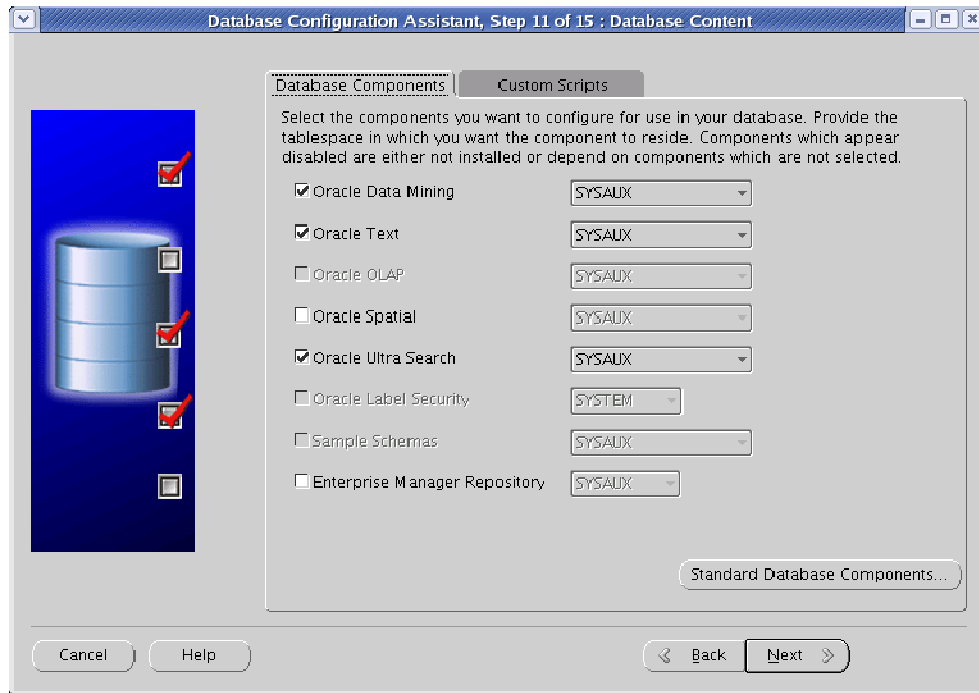
b) Rutas en Grupo DGINFRA2, para servidores de aplicaciones Externos.



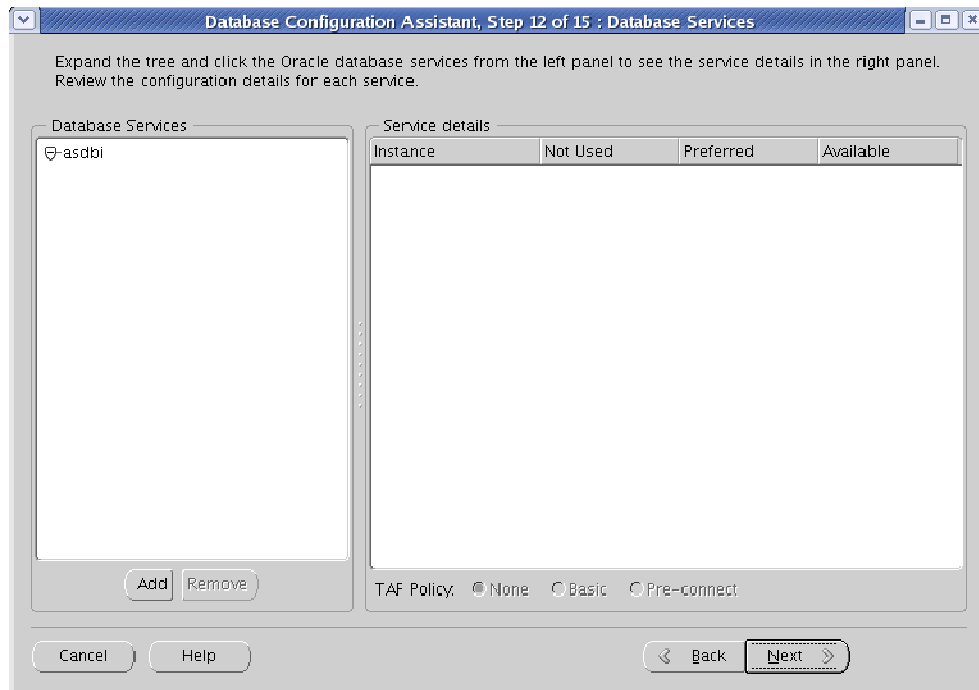
11. Tampoco necesitaremos activar la base de datos en modo archive.



12. El repositorio de metadatos requiere ciertos módulos extras para que pueda funcionar, por lo que seleccionamos los siguientes paquetes:

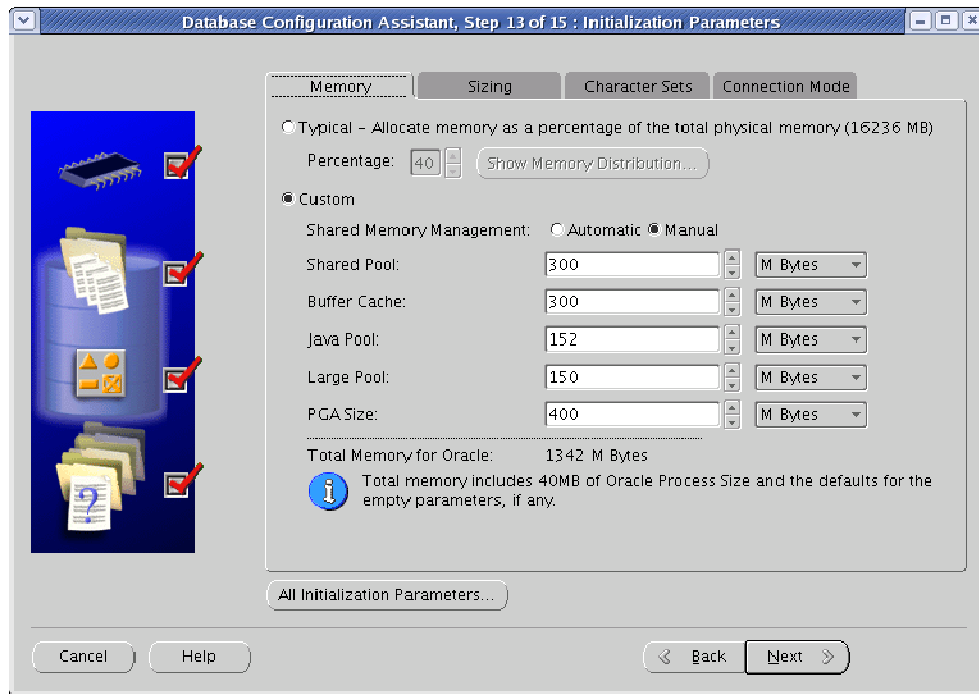


13. No creamos ningún servicio





14. Definimos parámetros de memoria recomendados para un buen funcionamiento.

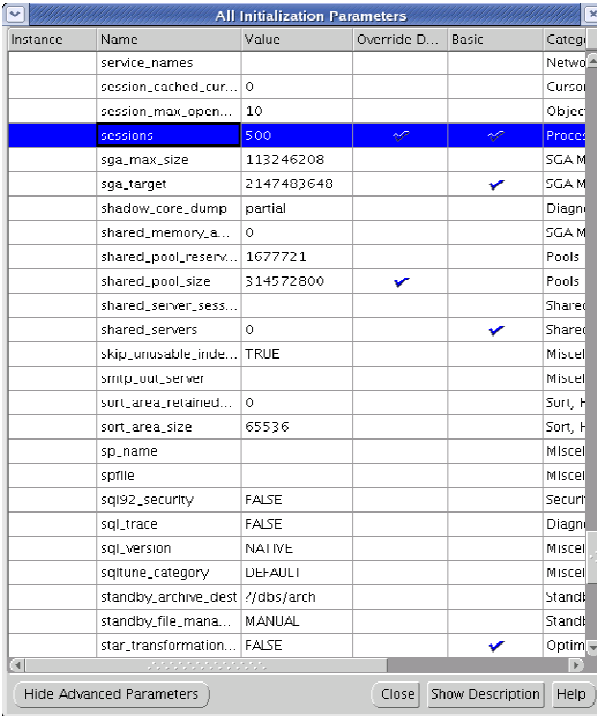


15. Modificamos el parámetro `aq_tm_processes` asignando el valor de 3.

The screenshot shows the 'All Initialization Parameters' window with a table of parameters. The parameter `aq_tm_processes` is highlighted in blue, showing a value of 3. The table includes columns for Instance, Name, Value, Override D..., Basic, and Category.

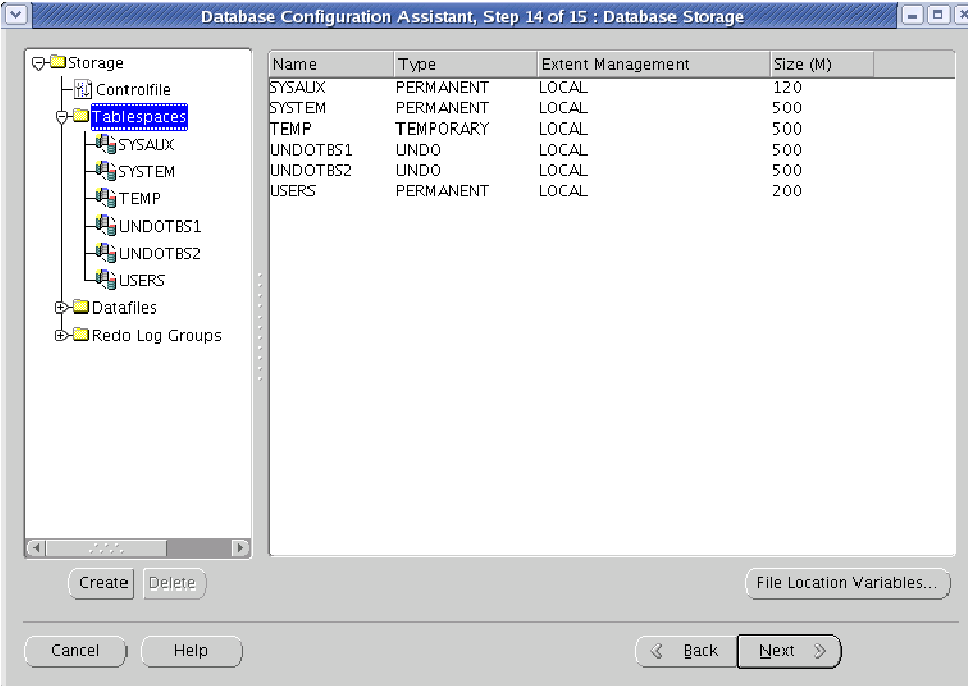
Instance	Name	Value	Override D...	Basic	Categori
	<code>O7_DICTIONARY_A...</code>	FALSE			Securi
	<code>active_instance_count</code>				Cluste
	<code>aq_tm_processes</code>	3	<input checked="" type="checkbox"/>		Miscel
	<code>archive_lag_target</code>	0			Standb
	<code>asm_diskgroups</code>				Autom
	<code>asm_diskstring</code>				Autom
	<code>asm_power_limit</code>	1			Autom
	<code>audit_file_dest</code>	{ORACLE_BAS...}	<input checked="" type="checkbox"/>		Securi
	<code>audit_sys_operations</code>	FALSE			Miscel
	<code>audit_trail</code>	NONE			Securi
	<code>background_core...</code>	partial			Diagn
	<code>background_dump...</code>	{ORACLE_BAS...}	<input checked="" type="checkbox"/>		Diagn
	<code>backup_tape_io_sla...</code>	FALSE			Backu
	<code>bitmap_merge_are...</code>	1048576			Sort, H
	<code>blank_trimming</code>	FALSE			ANSI C
	<code>buffer_pool_keep</code>				Cache
	<code>buffer_pool_recycle</code>				Cache
	<code>circuits</code>				Share
	<code>cluster_database</code>	true	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Cluste
	<code>cluster_database_i...</code>	2	<input checked="" type="checkbox"/>		Cluste
	<code>cluster_interconnects</code>				Cluste
	<code>commit_point stren...</code>	1			Distrib
	<code>compatible</code>	10.2.0.2.0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Miscel
	<code>control file record ...</code>	7			Redo
	<code>control files</code>	{+DGINFRA1...}	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	File Co

## 16. Modificamos el parámetro Sessions, asignando el valor de 500



Instance	Name	Value	Override D...	Basic	Category
	service_names				Netwo
	session_cached_cur...	0			Cursor
	session_max_open...	10			Objec
	<b>sessions</b>	<b>500</b>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Proce
	sga_max_size	113246208			SGA M
	sga_target	2147483648		<input checked="" type="checkbox"/>	SGA M
	shadow_core_dump	partial			Diagn
	shared_memory_a...	0			SGA M
	shared_pool_reserv...	1677721			Pools
	shared_pool_size	314572800	<input checked="" type="checkbox"/>		Pools
	shared_server_sess...				Share
	shared_servers	0		<input checked="" type="checkbox"/>	Share
	skip_unusable_inde...	TRUE			Miscel
	smtp_out_server				Miscel
	sort_area_retained...	0			Sort, H
	sort_area_size	65536			Sort, H
	sp_name				Miscel
	sptfile				Miscel
	sql92_security	FALSE			Secur
	sql_trace	FALSE			Diagn
	sql_version	NATIVE			Miscel
	sqltune_category	DEFAULT			Miscel
	standby_archive_dest	~/dbs/arch			Standb
	standby_file_mana...	MANUAL			Standb
	star_transformation...	FALSE		<input checked="" type="checkbox"/>	Optim

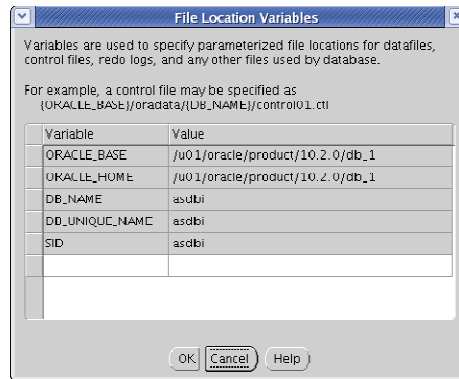
## 17. Modificamos el tamaño de los tablespaces, quedando de la siguiente manera:



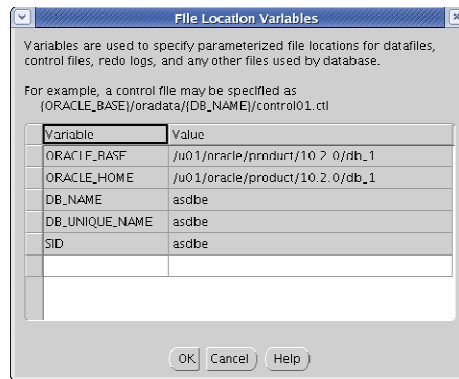
Name	Type	Extent Management	Size (M)
SYSAUX	PERMANENT	LOCAL	120
SYSTEM	PERMANENT	LOCAL	500
TEMP	TEMPORARY	LOCAL	500
UNDOTBS1	UNDO	LOCAL	500
UNDOTBS2	UNDO	LOCAL	500
USERS	PERMANENT	LOCAL	200

18. Revisamos las variables de ambiente requeridas para el funcionamiento de la base de datos.

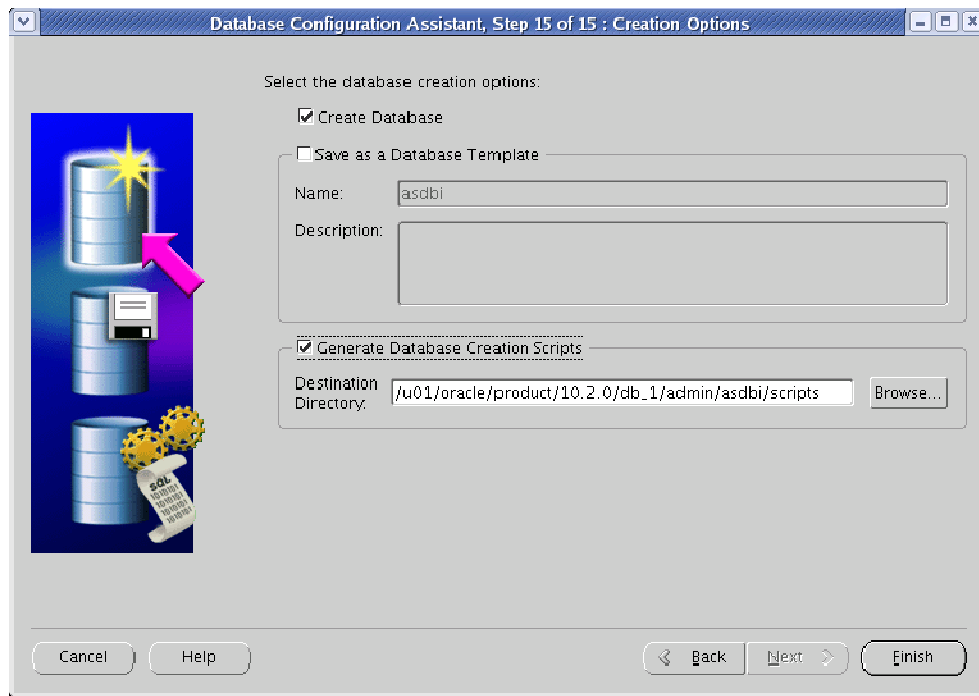
a) Variables Base de Datos asdbi, para servidores de aplicaciones Internos.



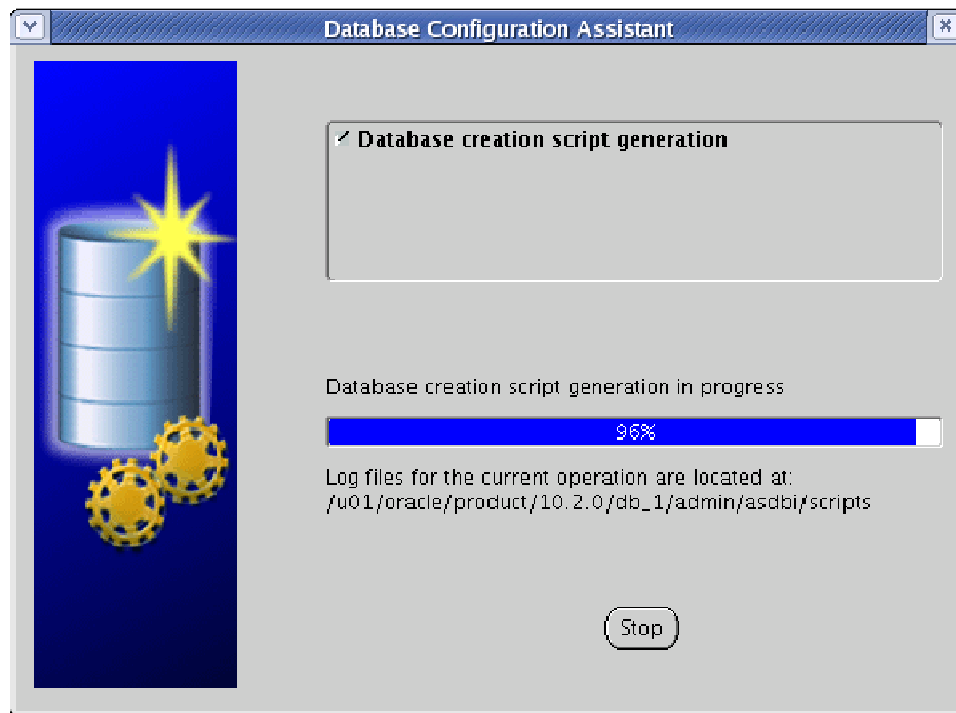
b) Variables Base de Datos asdbe, para servidores de aplicaciones Externos.

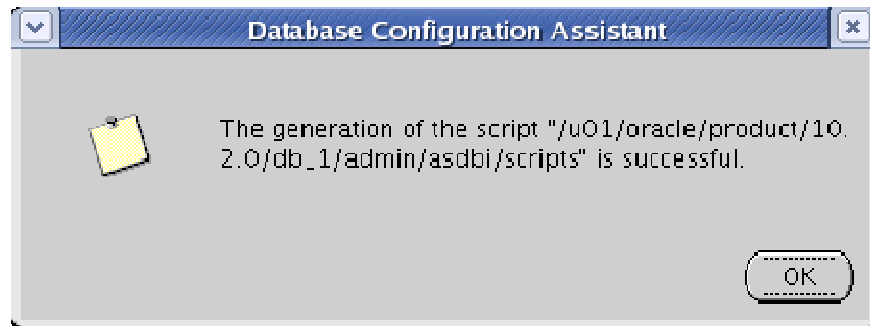


19. Como parte final de configuración del asistente, activamos la opción de crear scripts de la base de datos que se esta creando. A partir de que se da clic en el botón llamado **finish**, el asistente comienza a generar la base de datos.

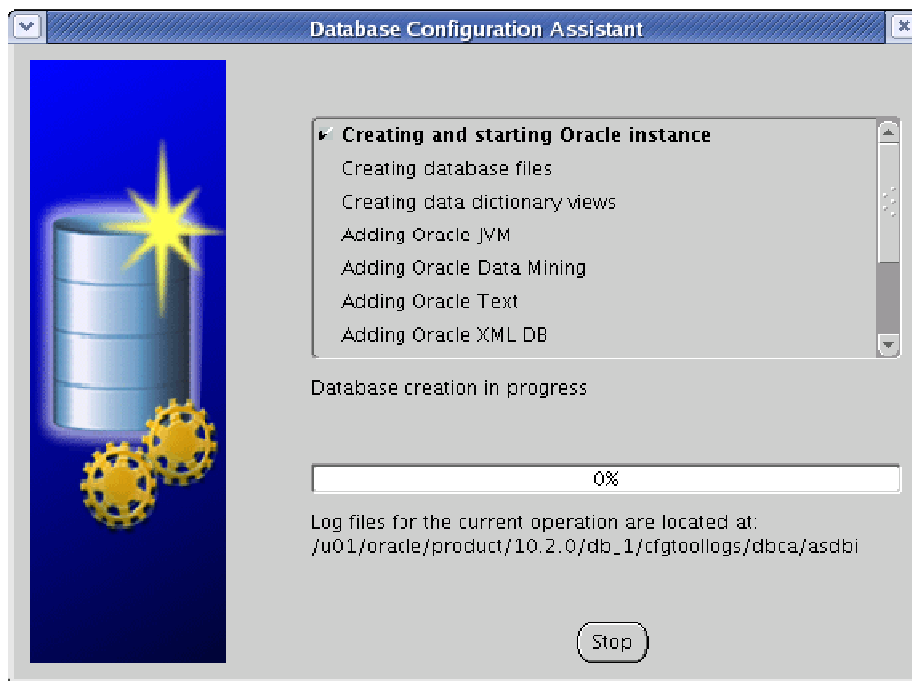


20. Progreso en la creación de scripts

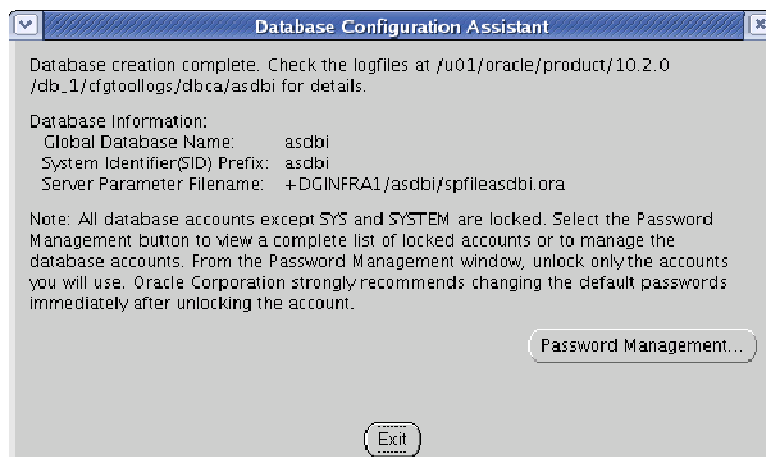




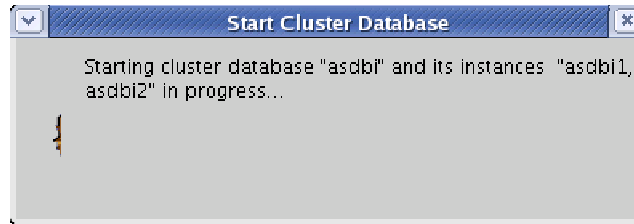
21. Progreso de creación de base de datos, y componentes por configurar.



22. Fin de creación de base de datos, mostrando la información general de la base de datos nueva.



23. Como parte final del proceso de creación de base de datos, el asistente configura las instancias y los servicios de listener en cada nodo del RAC.



Finalmente, después de tener configurada la base de datos, hacemos una prueba de conexión.

Asignamos variable de ambiente: ORACLE\_HOME=/u01/oracle/product/10.2.0/db\_1

Asignamos variable de ambiente: ORACLE\_SID = asdbi1 ó ORACLE\_SID = asdbe1

Hacemos una conexión a las base de datos de forma local

Sqlplus "/as sysdba"

Ya estando dentro de la base de datos, vía sqlplus ejecutamos la siguiente sentencia para revisar que se encuentra abierta la base de datos.

SQL> select status from v\$instance;

A screenshot of a terminal window titled "oracle@spmrac1:-". The terminal shows the following commands and output:

```
[oracle@spmrac1 ~]$ export ORACLE_HOME=/u01/oracle/product/10.2.0/db_1
[oracle@spmrac1 ~]$ export ORACLE_SID=asdbi1
[oracle@spmrac1 ~]$ sqlplus "/as sysdba"

SQL*Plus: Release 10.2.0.2.0 - Production on Thu Sep 28 16:26:46 2006

Copyright (c) 1982, 2005, Oracle. All Rights Reserved.

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Production
With the Partitioning, Real Application Clusters and Data Mining options

SQL> show parameter name

NAME                                 TYPE        VALUE
-----
db_file_name_convert                string      asdbi
db_name                              string      asdbi
db_unique_name                      string      asdbi
global_names                        boolean     FALSE
instance_name                       string      asdbi1
lock_name_space                    string
log_file_name_convert               string
service_names                      string      asdbi

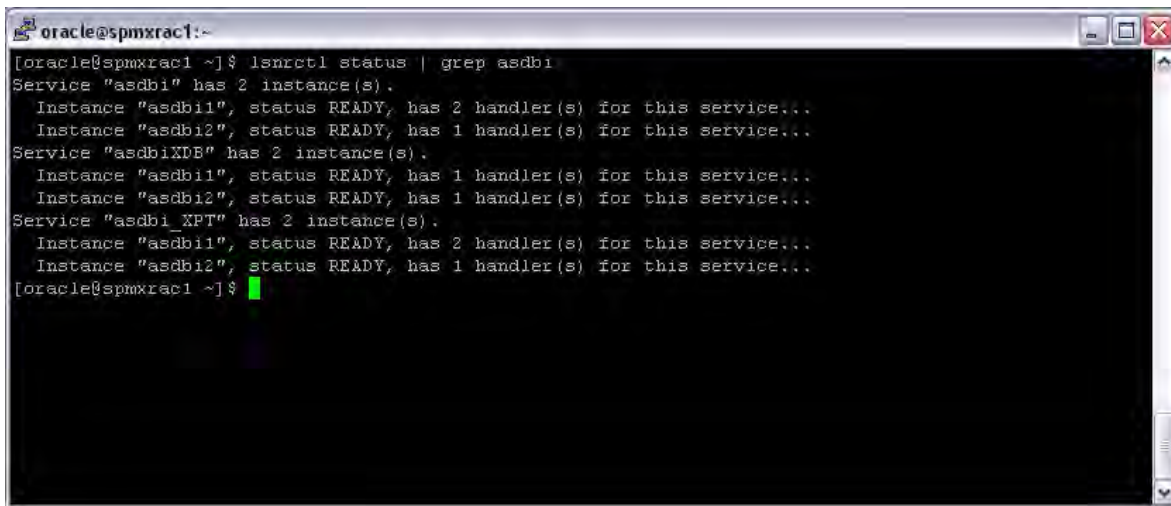
SQL> select status from v$instance;

STATUS
-----
OPEN

SQL>
```

Revisamos que el servicio de listener este activado y de servicio a la nueva base de datos.

\$lsnrctl status | grep asdb



```
oracle@spmxcrac1:~$ lsnrctl status | grep asdbi
Service "asdbi" has 2 instance(s).
  Instance "asdbi1", status READY, has 2 handler(s) for this service...
  Instance "asdbi2", status READY, has 1 handler(s) for this service...
Service "asdbiXDB" has 2 instance(s).
  Instance "asdbi1", status READY, has 1 handler(s) for this service...
  Instance "asdbi2", status READY, has 1 handler(s) for this service...
Service "asdbi_XPT" has 2 instance(s).
  Instance "asdbi1", status READY, has 2 handler(s) for this service...
  Instance "asdbi2", status READY, has 1 handler(s) for this service...
[oracle@spmxcrac1 ~]$
```

### 3.3 Actualización de Base de Datos con OracleAS Metadata Repository Creation Assistant (REPCA)

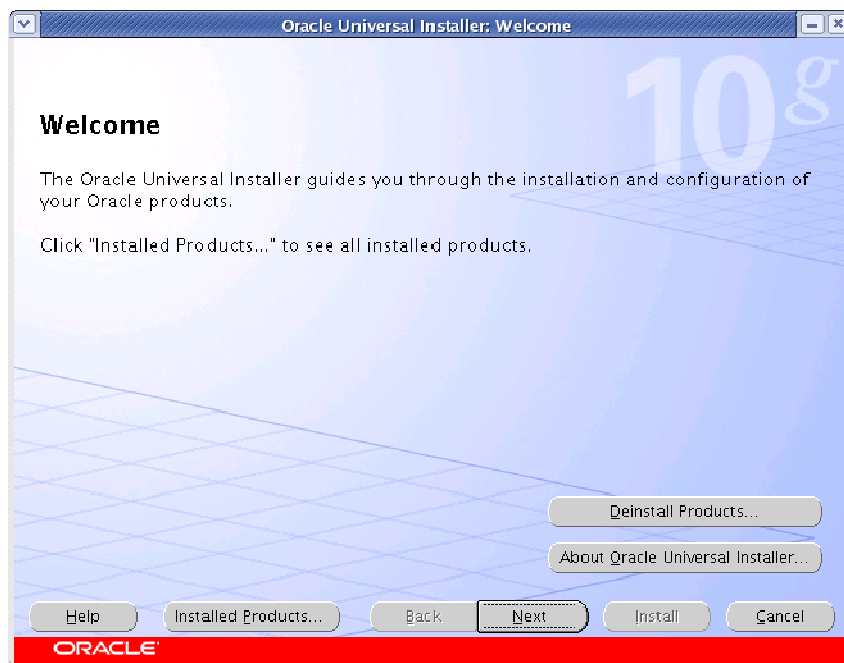
Oracle Application server requiere de una base de datos, para utilizarla como “Metadata Repository”. Cuando se hace una implementación de OAS en un servidor que administrara la base de datos localmente, el mismo software de OAS, puede crear el repositorio, pero en este caso las bases de datos se encuentran en un ambiente de RAC, por lo que se requiere crear la base de datos en el RAC (ver punto 3.2), ya teniendo la base de datos lista, se utiliza un software llamado “OracleAS Metadata Repository Creation Assistant” (REPCA).

Este asistente actualiza la base de datos creada en el RAC, con los componentes necesarios para que pueda funcionar Oracle Application Server 10g R2.

Esta actualización solo se hace una vez, y se hace desde el Nodo 1 de los servidores de aplicaciones.

El procedimiento que se llevo acabo para la actualización de la base de datos es el siguiente:

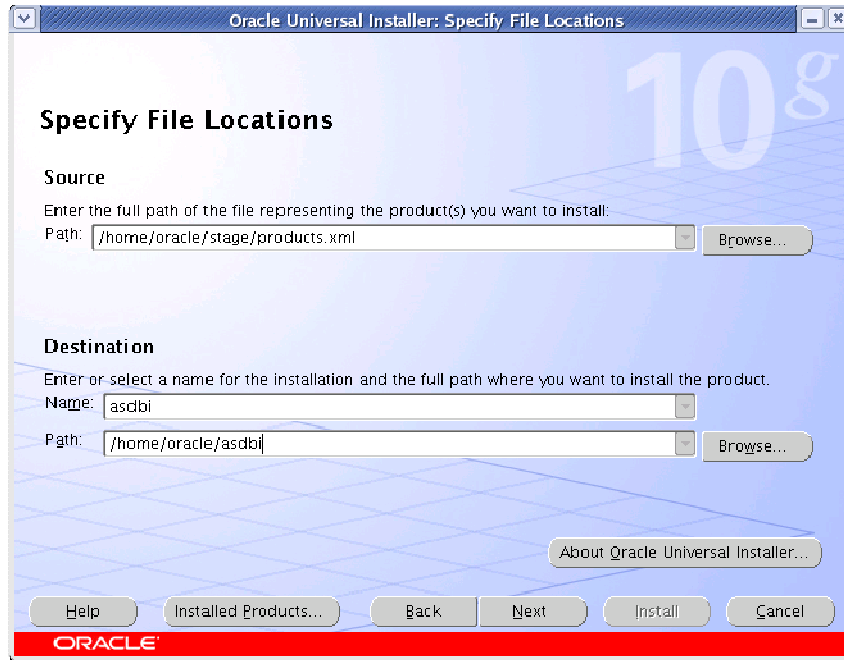
1. Descargar y desempacar el software de REPCA en una carpeta con permisos para que el usuario Oracle del sistema operativo pueda tener acceso a ella. Ya teniendo el software listo, ejecutamos el comando: runInstaller, el cual iniciara el instalador de la aplicación.



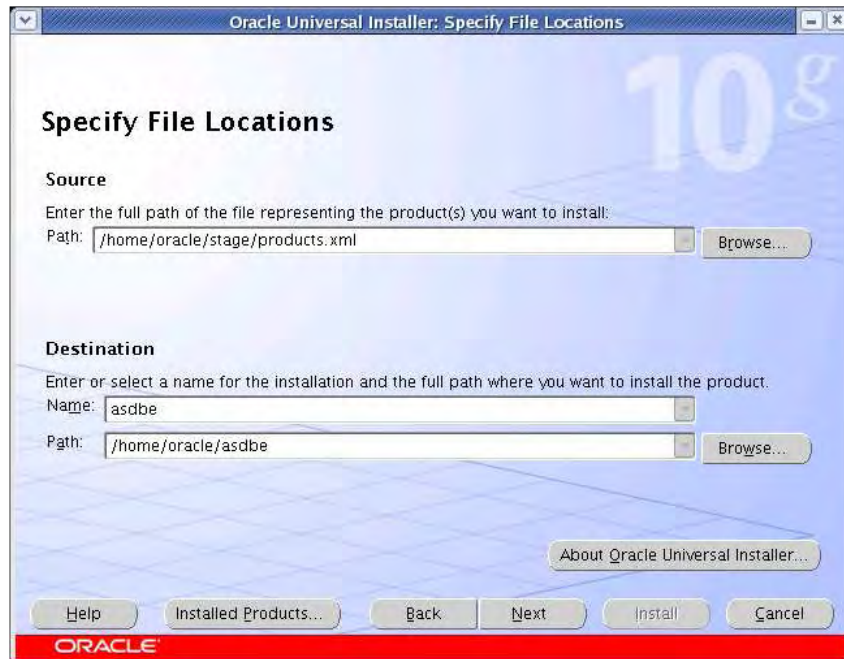


2. Especificamos el origen del software de instalación, el nombre de la instalación y el destino donde se instalara el software.

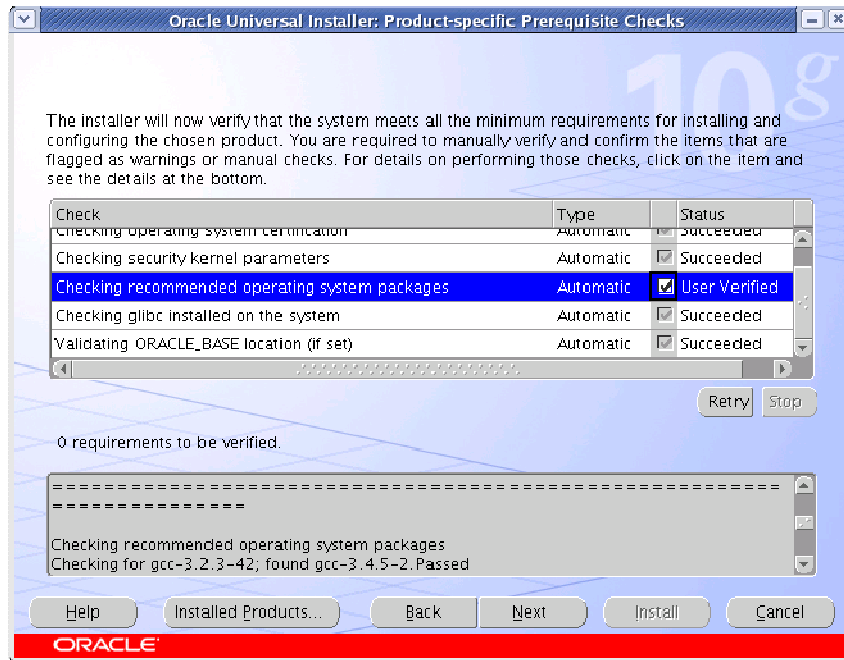
a) Rutas Base de Datos asdbi, para servidores de aplicaciones Internos.



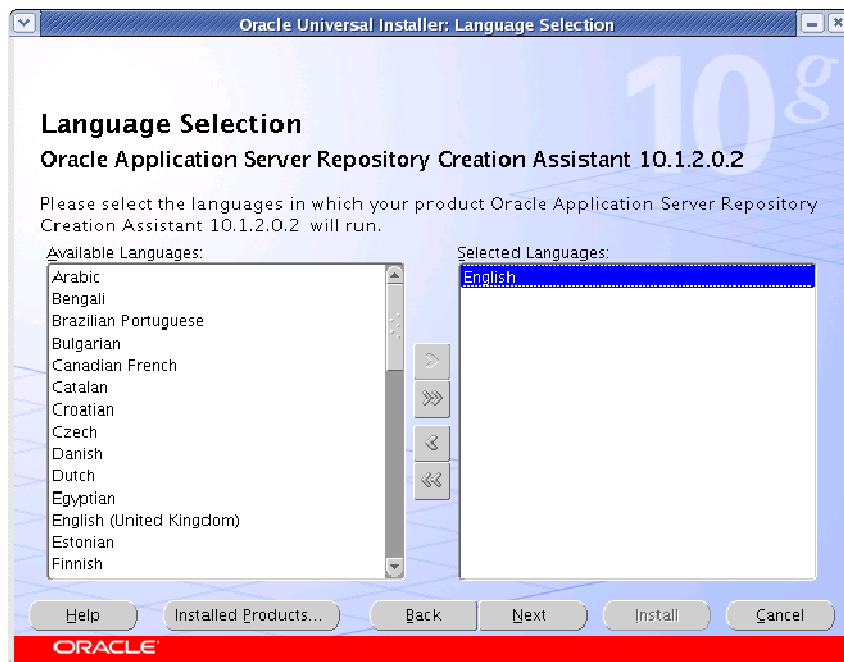
b) Rutas Base de Datos asdbe, para servidores de aplicaciones Externos.



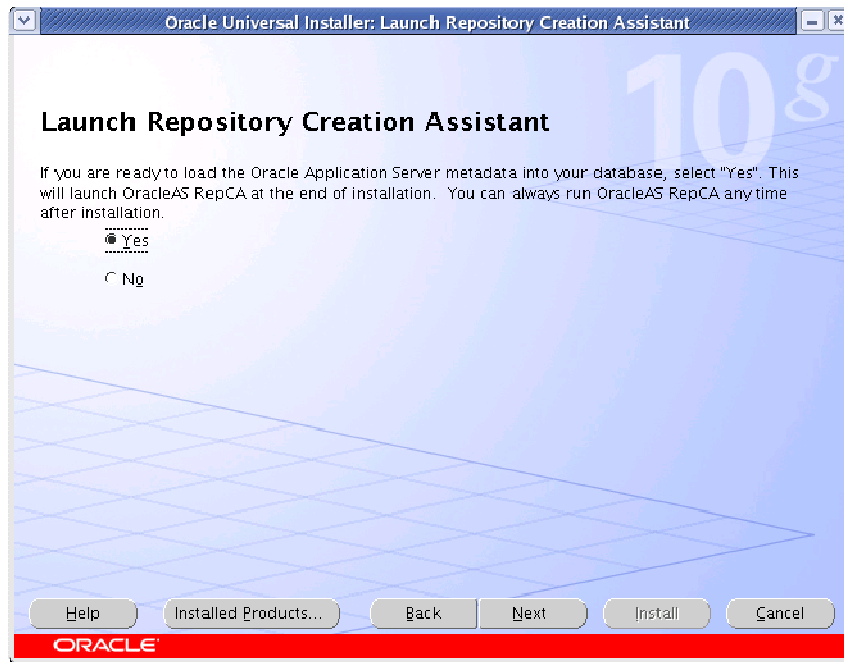
3. El asistente verifica que se cumplan los requerimientos mínimos del sistema, para poder continuar con la instalación.



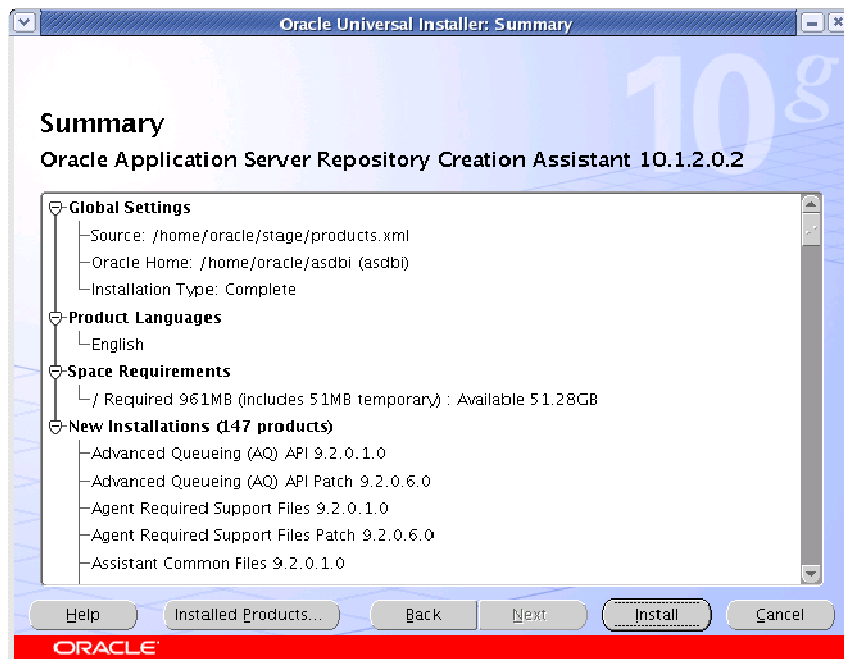
4. Seleccionamos el idioma que se utilizara en esta instalación.



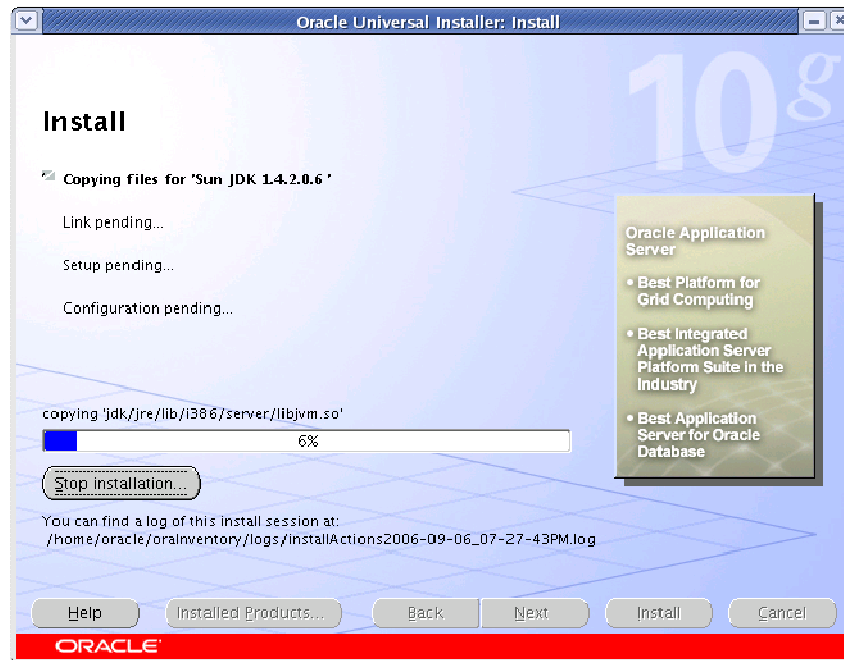
5. El asistente se divide en dos fases, la primera es instalar el software de REPCA, y la otra es el uso del asistente REPCA, esto indica que puedes hacer la instalación del software, y ejecutar el asistente después, pero si se cumple con todos los requisitos y se desea ejecutar el REPCA, seleccionamos la opción YES.



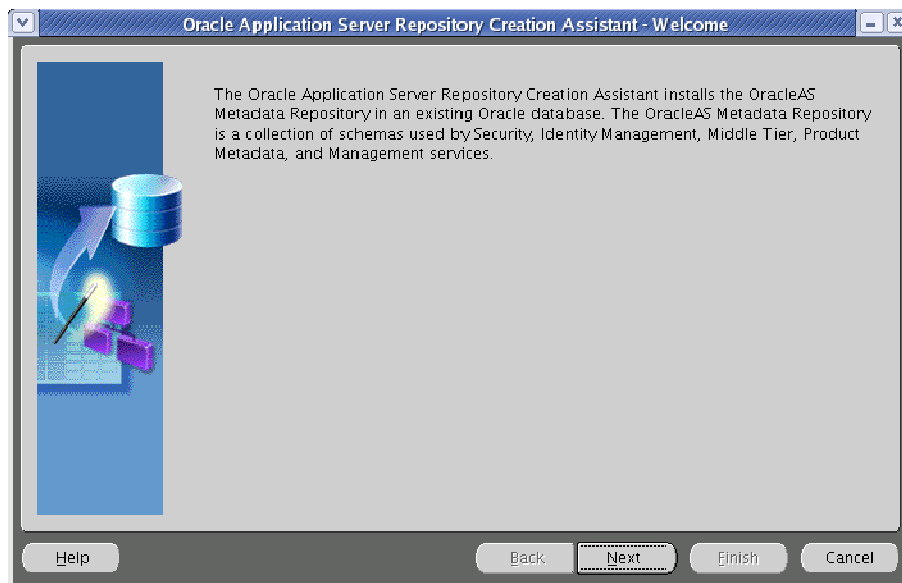
6. Se genera un resumen de la instalación y si no hay observaciones, se comienza la instalación.



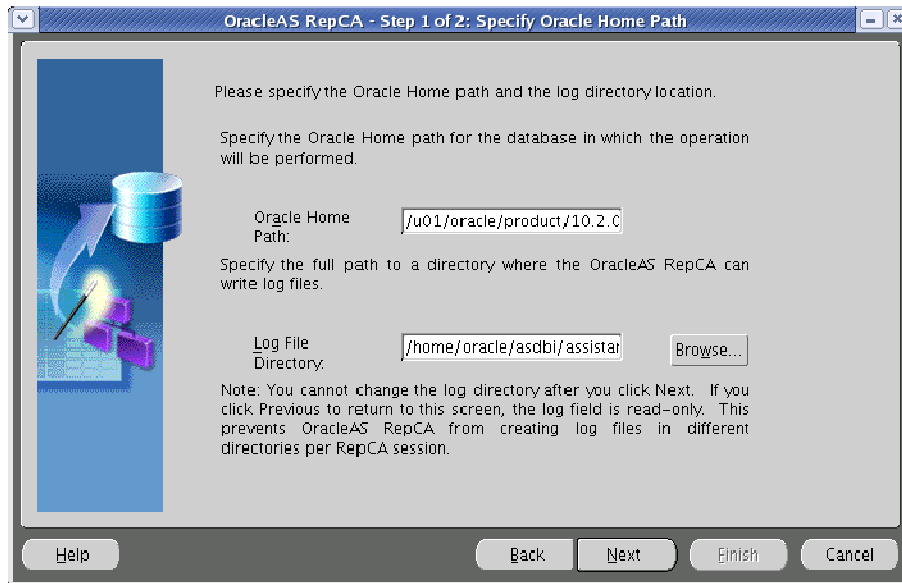
## 7. Progreso de la copia de archivos.



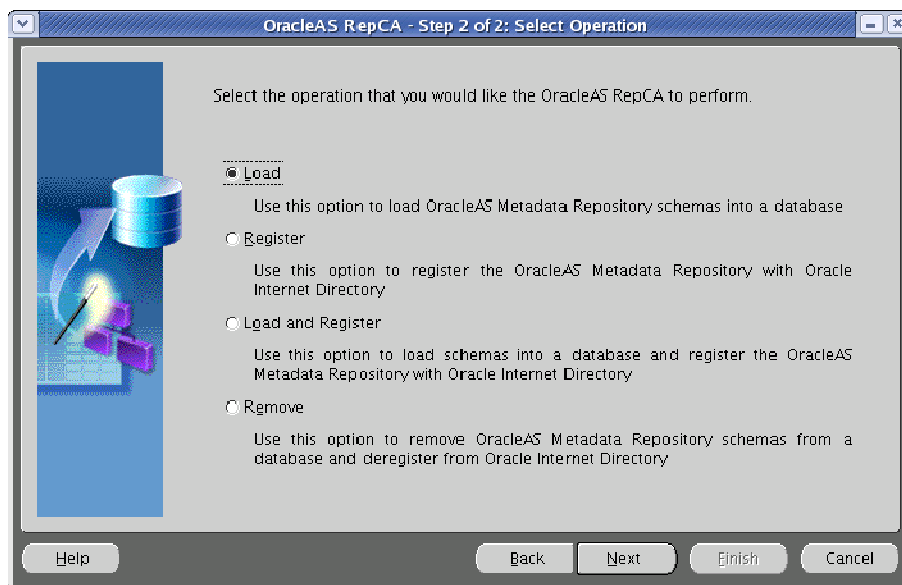
## 8. Al terminar la copia de archivos, inicia automáticamente el asistente REPCA



9. Especificamos la ruta asignada a la variable de ambiente \$ORACLE\_HOME en el servidor SPMXRAC1, así como la ruta en donde se generaran los archivos LOG.

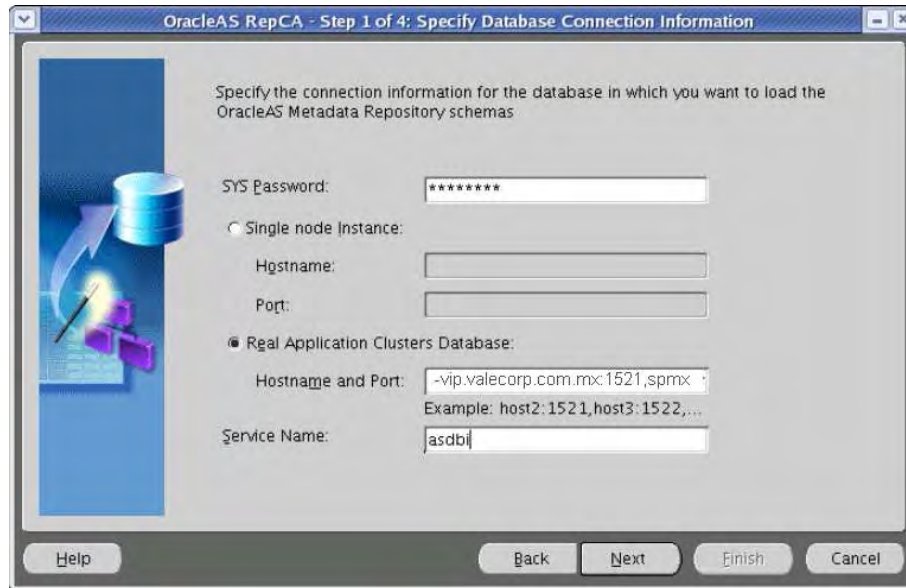


10. Este asistente tiene diferentes utilidades, la carga del repositorio de datos, el registro en una instancia de Identity Management, o ambas cosas y la eliminación de alguna de ellas. Para este caso solo haremos la carga de "Metadata Repository"



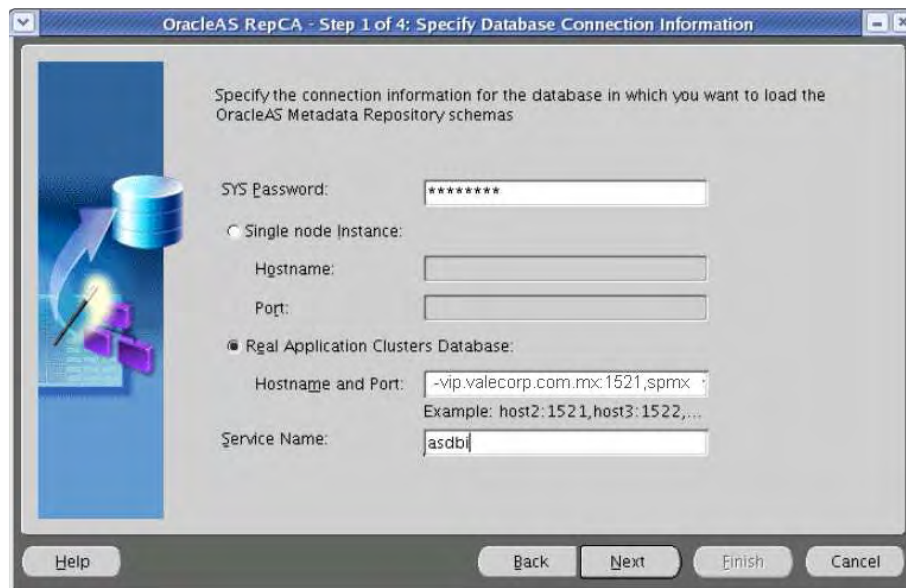
11. Indicamos el password de SYS de la base de datos que se actualizará, los servidores de RAC que contienen la base de datos y el nombre del servicio. Cabe mencionar que tenemos que indicar los nombres virtuales del RAC.

a) Servicio asdbi, para servidores de aplicaciones Internos.



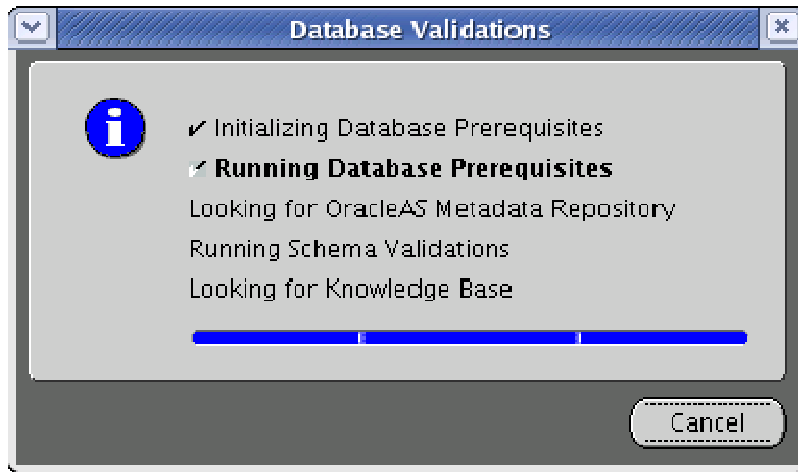
The screenshot shows the 'OracleAS RepCA - Step 1 of 4: Specify Database Connection Information' dialog box. The title bar includes a dropdown arrow, the text 'OracleAS RepCA - Step 1 of 4: Specify Database Connection Information', and standard window control buttons. The main area contains the instruction: 'Specify the connection information for the database in which you want to load the OracleAS Metadata Repository schemas'. On the left is a graphic of a database cylinder with an arrow pointing to a document. The form fields are: 'SYS Password:' with a masked password '\*\*\*\*\*'; 'Single node Instance:' with an unselected radio button; 'Hostname:' and 'Port:' with empty text boxes; 'Real Application Clusters Database:' with a selected radio button; 'Hostname and Port:' with the value '-vip.valecorp.com.mx:1521,spx' and an example 'Example: host2:1521,host3:1522,...'; and 'Service Name:' with the value 'asdbi'. At the bottom are buttons for 'Help', 'Back', 'Next', 'Finish', and 'Cancel'.

b) Servicio asdbe, para servidores de aplicaciones Externos.

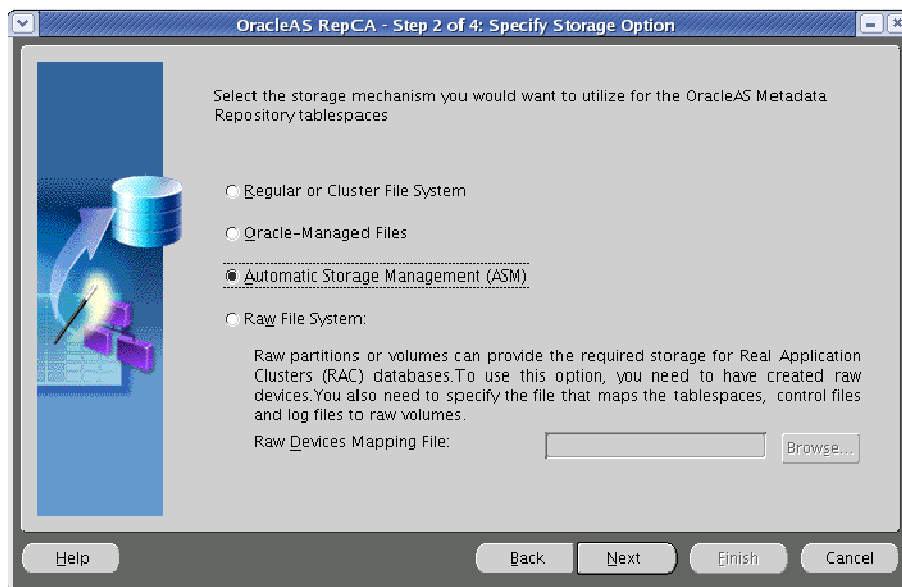


This screenshot is identical to the one above, showing the 'OracleAS RepCA - Step 1 of 4: Specify Database Connection Information' dialog box. The 'Service Name' field is filled with 'asdbi'.

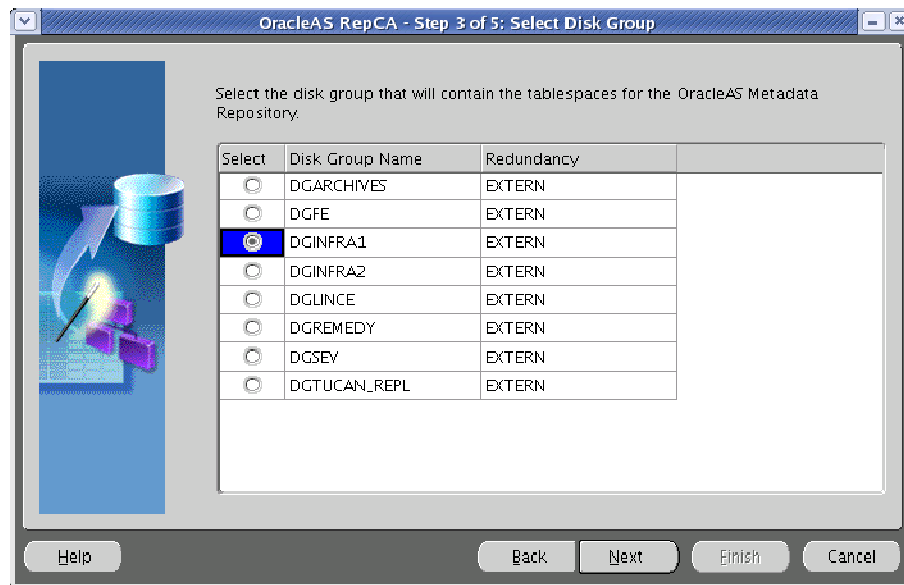
12. El asistente comienza a analizar la base de datos y revisa que cumpla con los requerimientos mínimos.



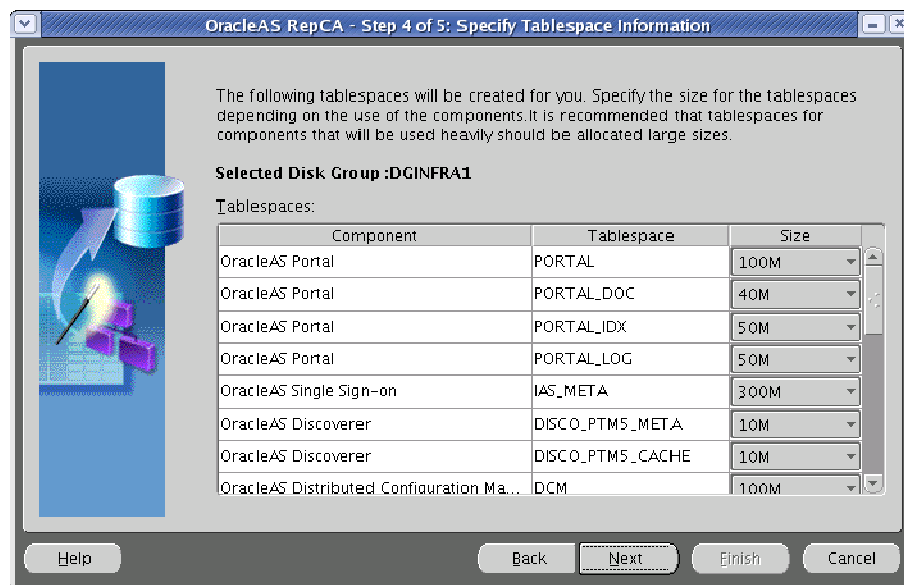
13. Después de revisar la base, tenemos que indicar, como es el almacenamiento de la base.



14. Indicamos el grupo de ASM que se le asigno a la instalación de los servidores de aplicaciones, DGINFRA1 para servidores internos y DGINFRA2 para servidores externos.

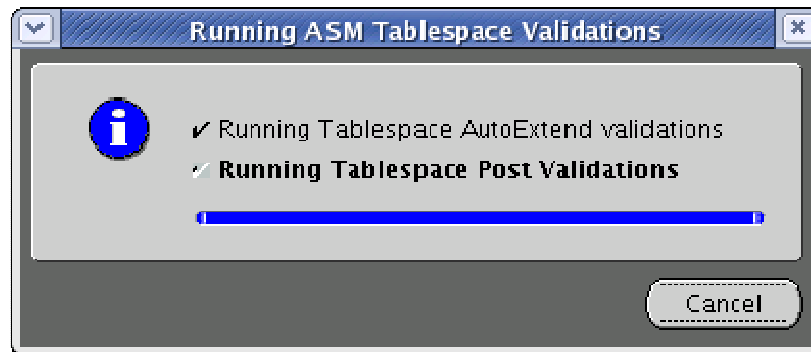


15. A continuación se muestran los componentes que se agregaran a la base de datos, para que pueda funcionar como repositorio de metadatos. Si se requiere modificar el tamaño de algún componente se puede hacer, en este caso se utilizaron los valores por default

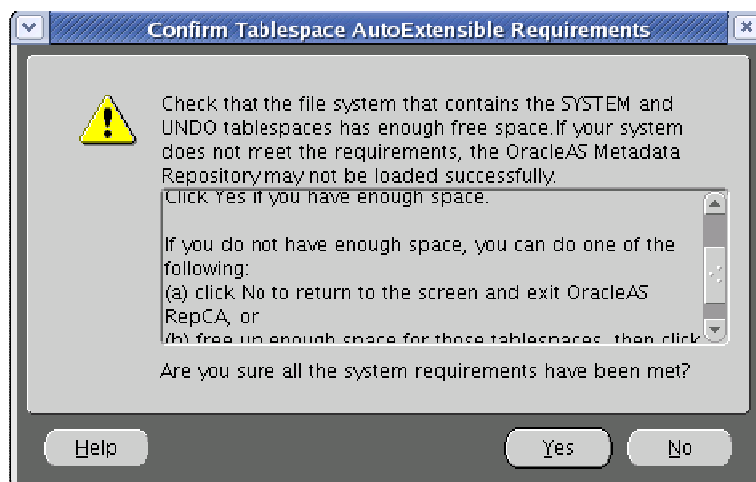




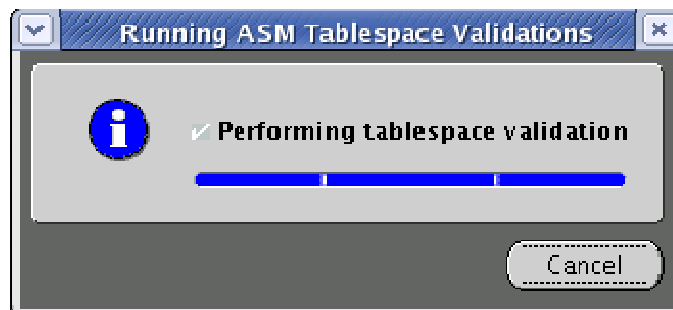
16. Antes de hacer la carga de los componentes comienza una revisión de los tablespaces



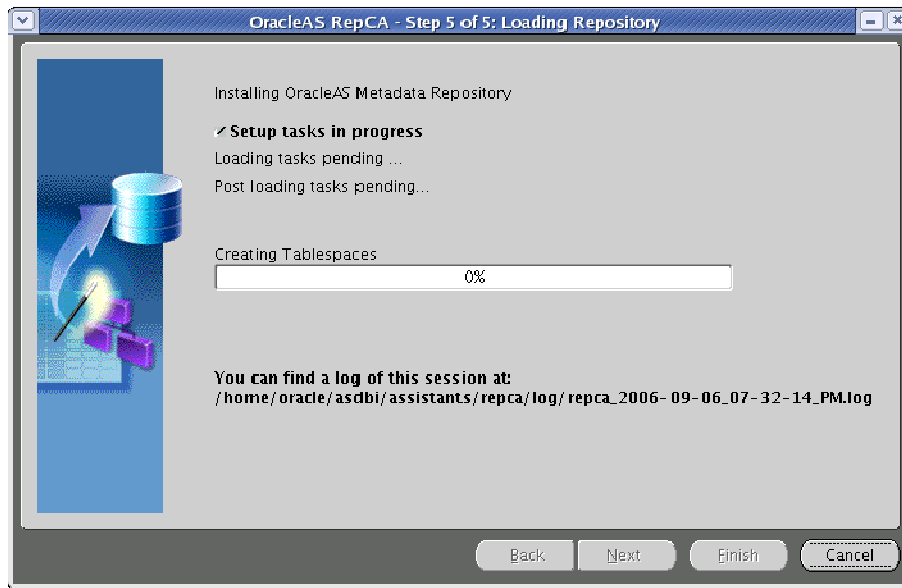
17. El asistente nos da una advertencia, requiriendo suficiente espacio para el tablespace de SYSTEM y UNDO, como estamos seguros que tenemos suficiente espacio, continuamos con la instalación.



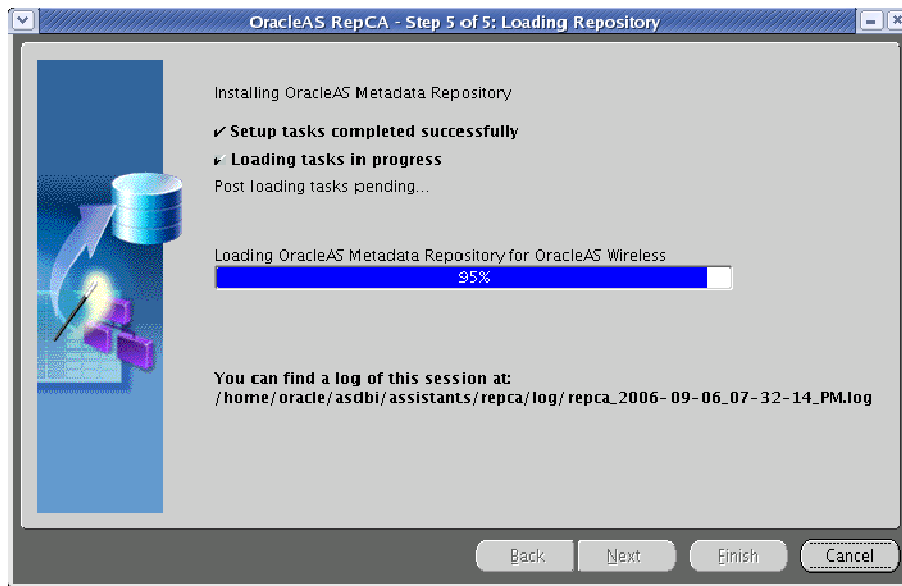
18. Se inicia una validación de tablespaces



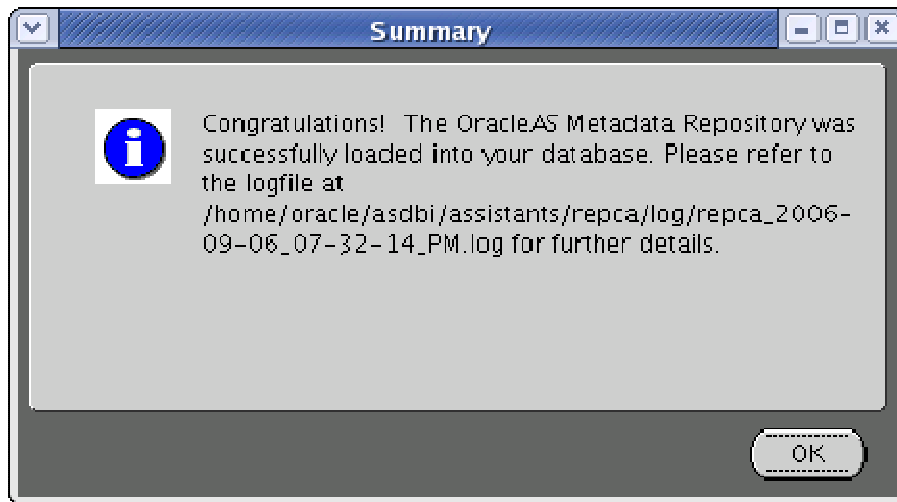
19. Comienza la carga de componentes a la base de datos y la configuración de ella.



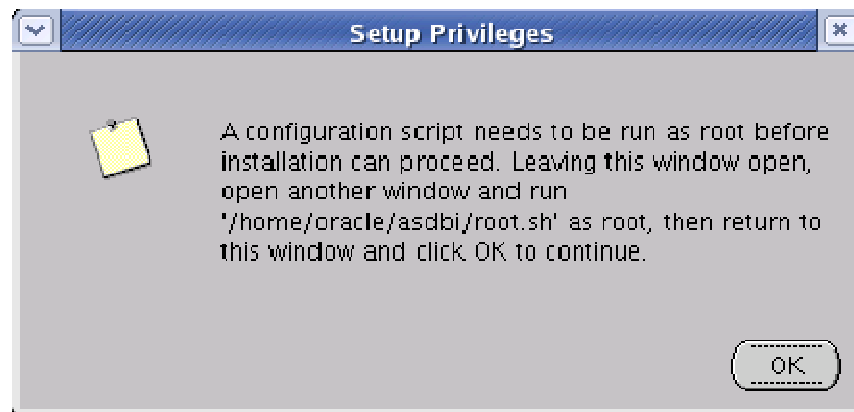
20. Progreso de la carga.



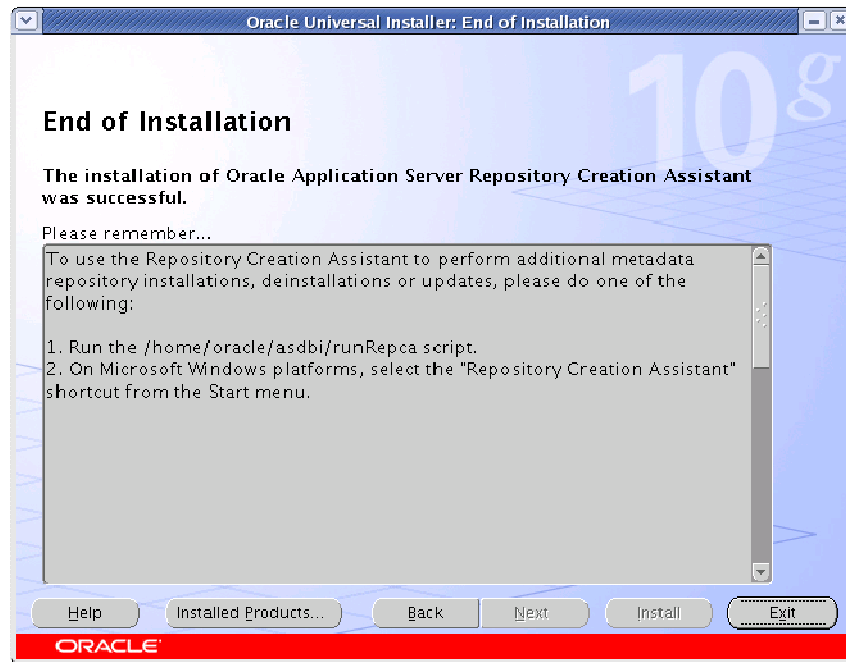
21. Después de la carga se muestra un aviso de término de carga.



22. Se requiere ejecutar un script con los permisos de root.



23. Al terminar se muestra el resumen de la instalación, dando fin a esta.



### 3.4 Instalación de instancia para “Identity Management” y configuración de cluster, en NODO1

Las aplicaciones Middle Tier (Portal, Forms y Reports) de Oracle Application Server 10g R2, requieren como parte de su infraestructura, una instancia de Identity Management, la cual consta principalmente de SSO, e Internet Directory.

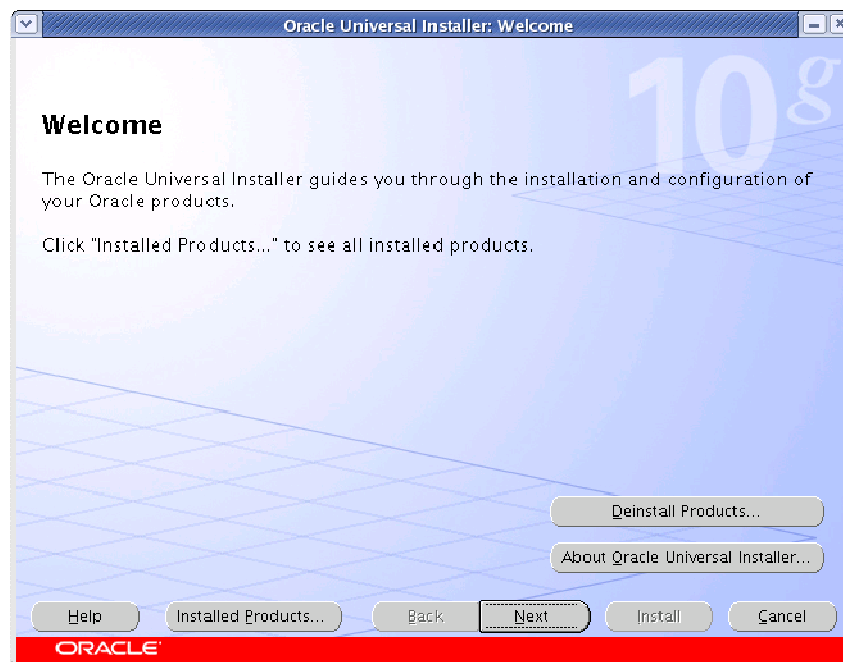
Como se ha mencionado antes, se requiere que estas instancias se configuren en un ambiente de cluster.

Oracle Application Server utiliza un concepto llamado granja, el cual consiste en tener n cantidad de instancias registradas en el mismo “Metadata Repository” e “Identity Management”, y a su vez, hay instancias que se pueden unir mediante un cluster lógico creado en la granja.

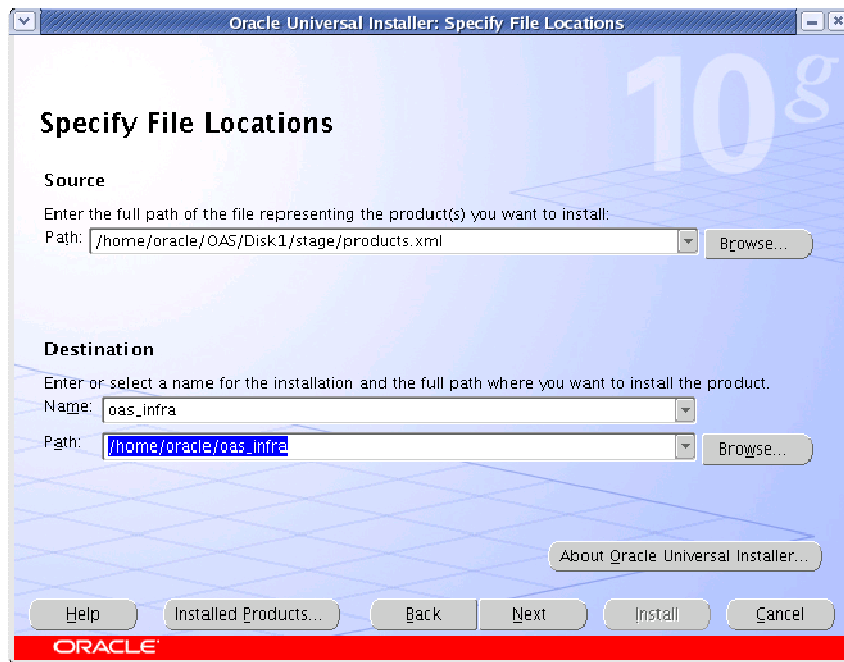
Es muy importante tener en cuenta que en este ambiente se va a utilizar un balanceador de cargas externo, por lo tanto se requiere que el DNS o el archivo /etc/hosts, contenga la información del balanceador, de manera que las instancias registraran la IP del balanceador y no la de un Nodo del cluster. La topología se puede apreciar en la introducción.

A continuación se muestra el procedimiento que se lleva a cabo para crear la instancia de “Identity Management” en el Nodo 1, y la creación del cluster para estas instancias.

1. Descargar y desempacar el software de “Oracle Application Server 10g R2” en una carpeta con permisos para que el usuario Oracle del sistema operativo pueda tener acceso a ella. Ya teniendo el software listo, ejecutamos el comando: runInstaller, el cual iniciara el instalador de la aplicación



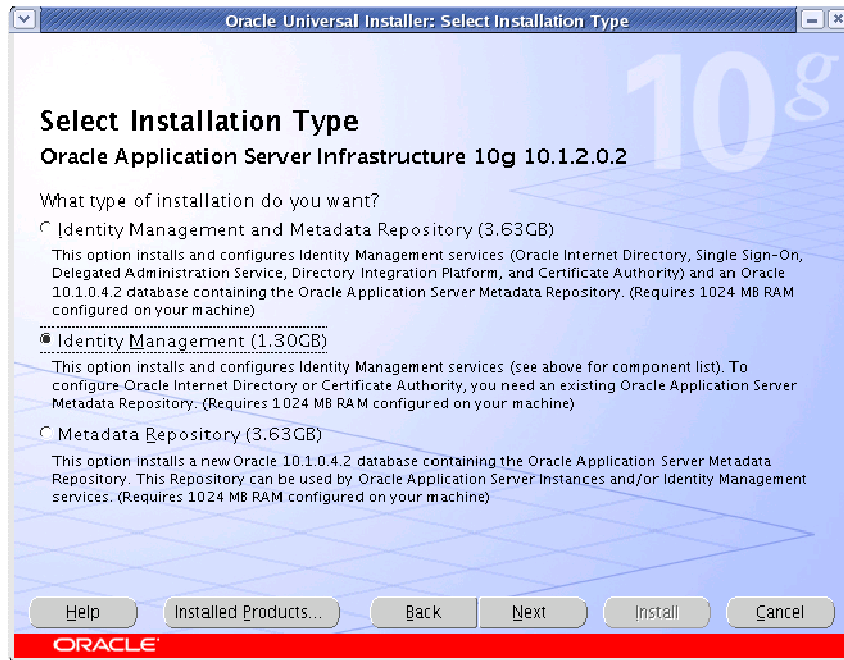
2. Especificamos el origen del software de instalación, el nombre de la instalación y el destino donde se instalara el software.



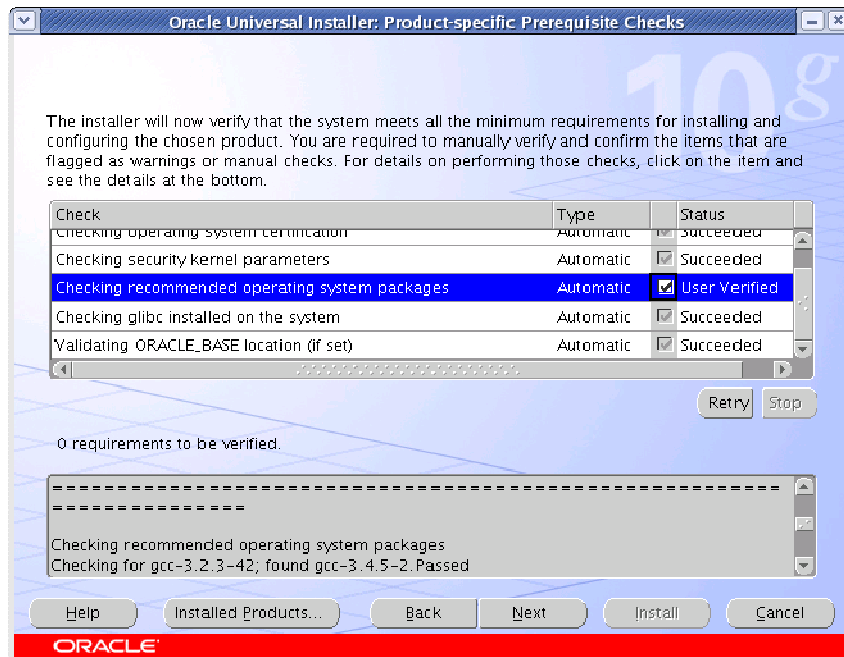
3. Seleccionamos el producto a instalar, para poder comenzar, seleccionamos Infraestructura.



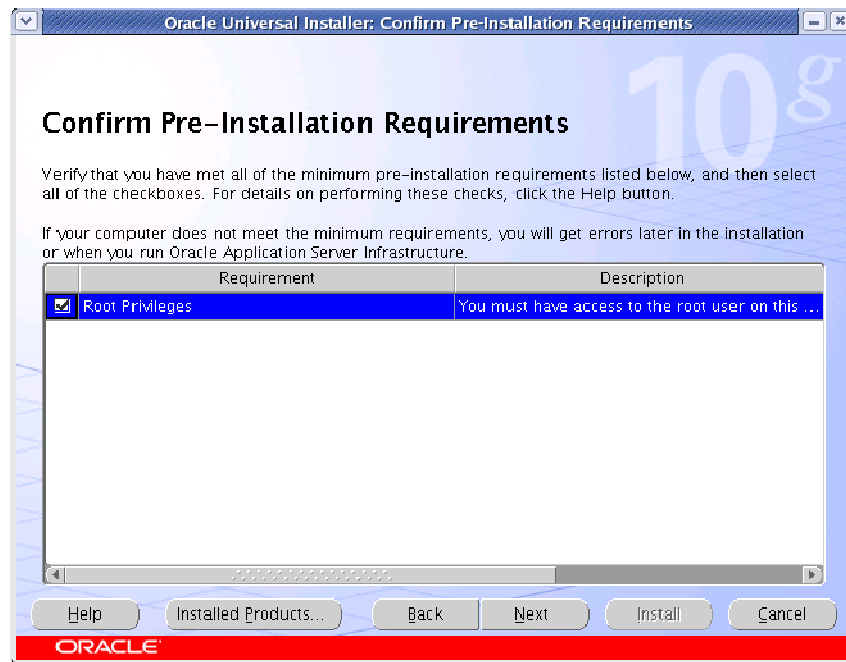
4. Recordemos que anteriormente se creó la base de datos "Metadata Repository", por lo tanto solo necesitamos instalar "Identity Management".



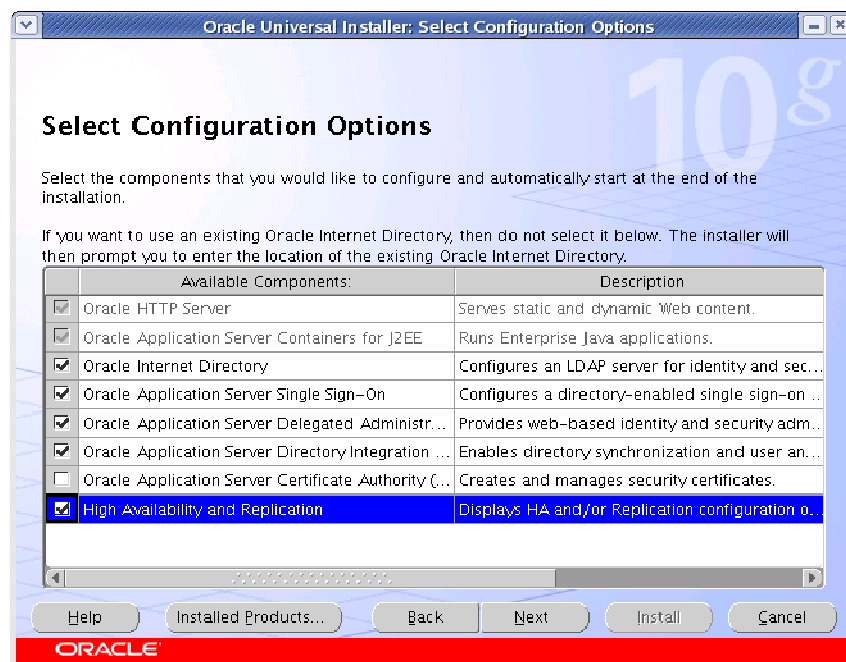
5. El asistente verifica que se cumplan los requerimientos mínimos del sistema, para poder continuar con la instalación.



6. Como prerequisite el asistente nos informa que necesitamos tener acceso a la cuenta de root, para ejecutar dos scripts.

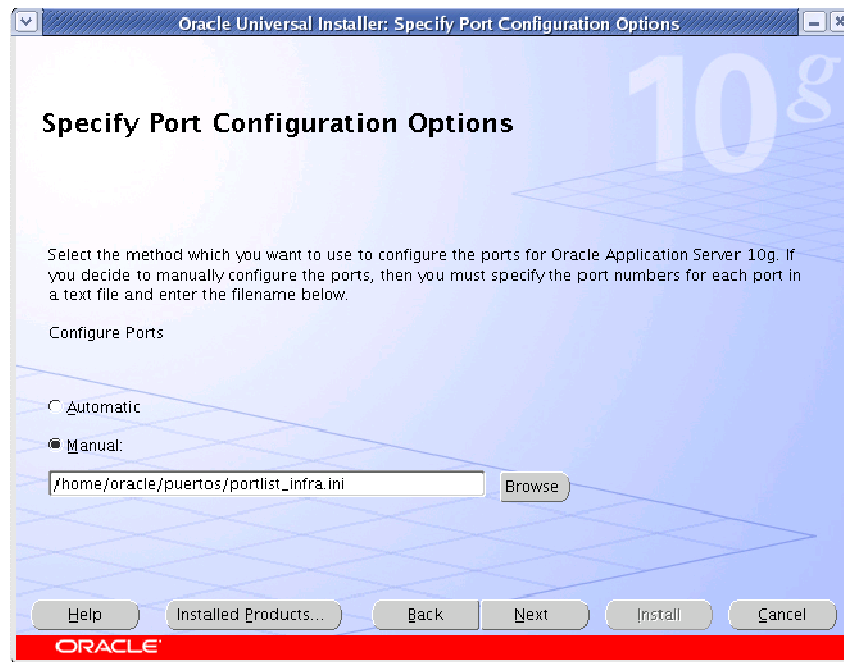


7. Seleccionamos los componentes que se desean instalar.





8. Especificamos los puertos que utilizaremos para la instancia "Identity Management", el asistente puede crear los puertos de forma automática, pero como se requiere tener un control de ello, se genero anteriormente un archivo de puertos llamado portlist\_infra.ini.



9. El asistente requiere tener acceso a la base de datos, para poder hacer la carga y configuración de esta instancia, por lo que tenemos que indicar la siguiente información:

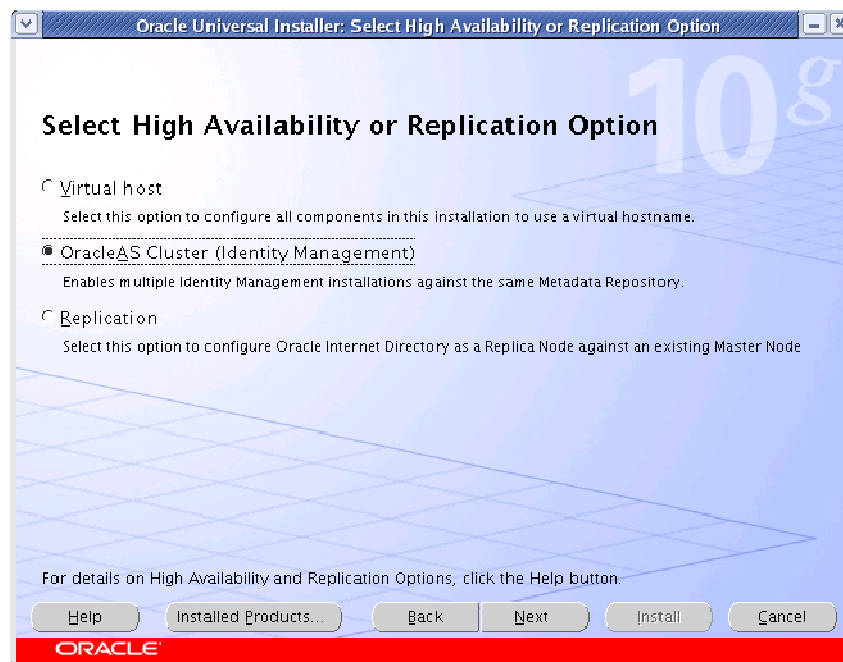
a) Servicio Base de Datos asdbi, para servidores de aplicaciones Internos.



b) Servicio Base de Datos asdbe, para servidores de aplicaciones Externos.



10. De las opciones que tenemos para alta disponibilidad, seleccionamos la creación de un cluster por medio de "Identity Management"



11. Especificamos el nombre del nuevo cluster.

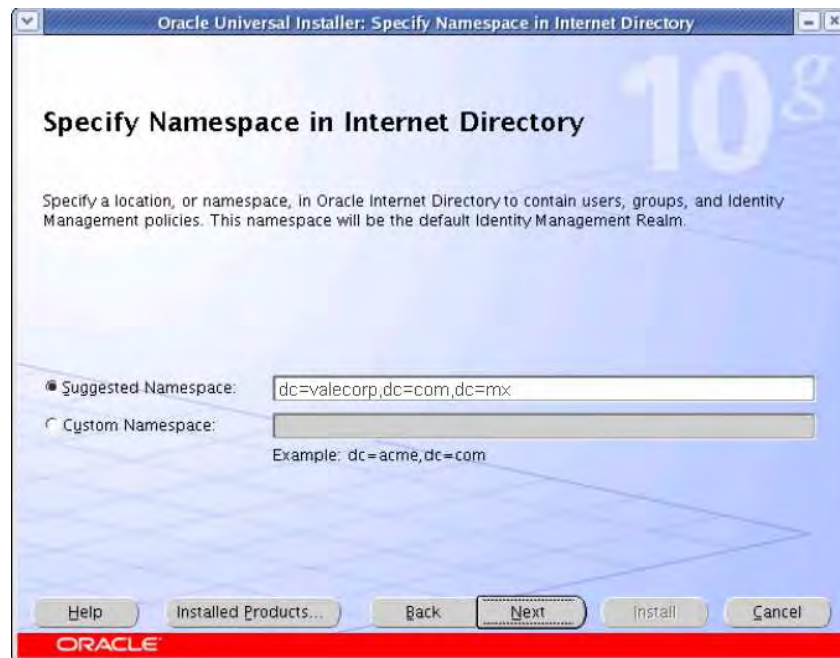
a) Nombre de cluster para instancia sso de servidores de aplicaciones Internos.



b) Nombre de cluster para instancia sso de servidores de aplicaciones Externos.

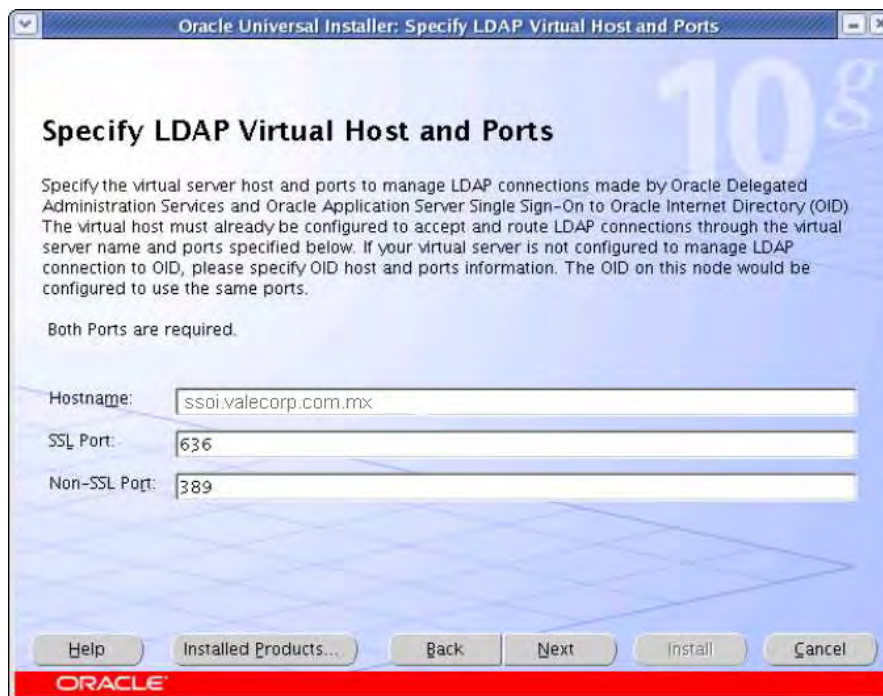


12. Especificamos el espacio de nombre para Internet Directory.



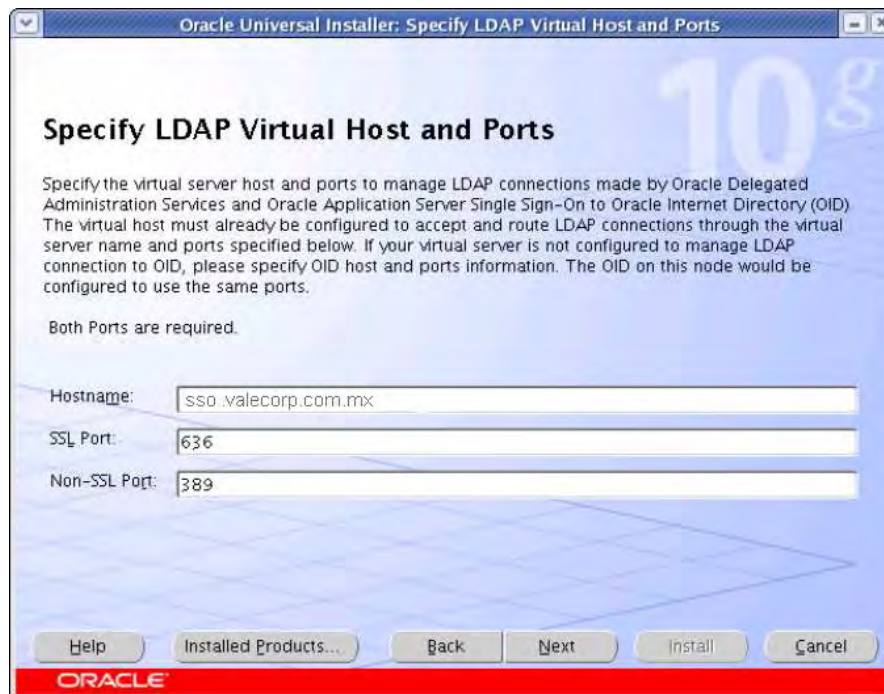
13. Especificamos el nombre del equipo por el que se tendrá acceso a la instancia de single sign on (sso), este nombre tiene que estar registrado en el DNS o en el archivo /etc/hosts, apuntando a la IP del balanceador y especificamos el puerto que se utilizaran.

a) Host Virtual para sso de servidores de aplicaciones Internos.





b) Host Virtual para sso de servidores de aplicaciones Externos.

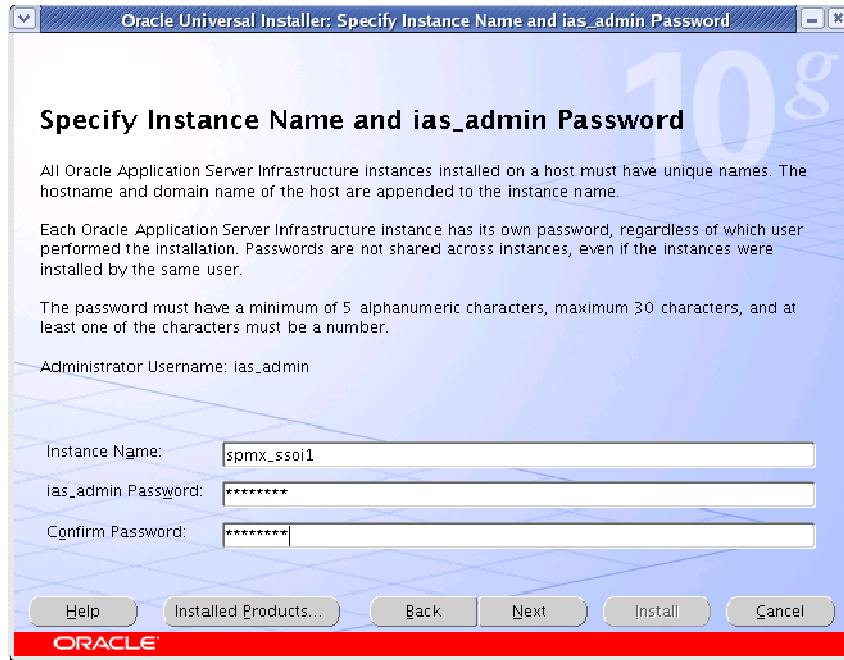


14. Especificamos el nombre del equipo por el que se tendrá acceso al servidor Web, y sus puertos.



15. Especificamos el nombre de la instancia para "Identity Management" del Nodo 1 y el password para la administración de esta.

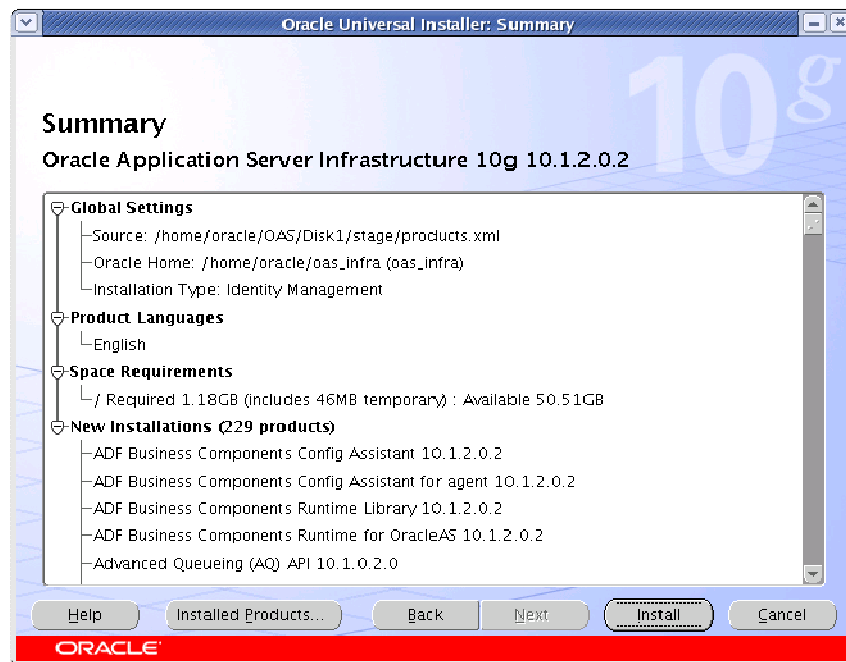
a) Nombre de instancia para servidores de aplicaciones Internos.



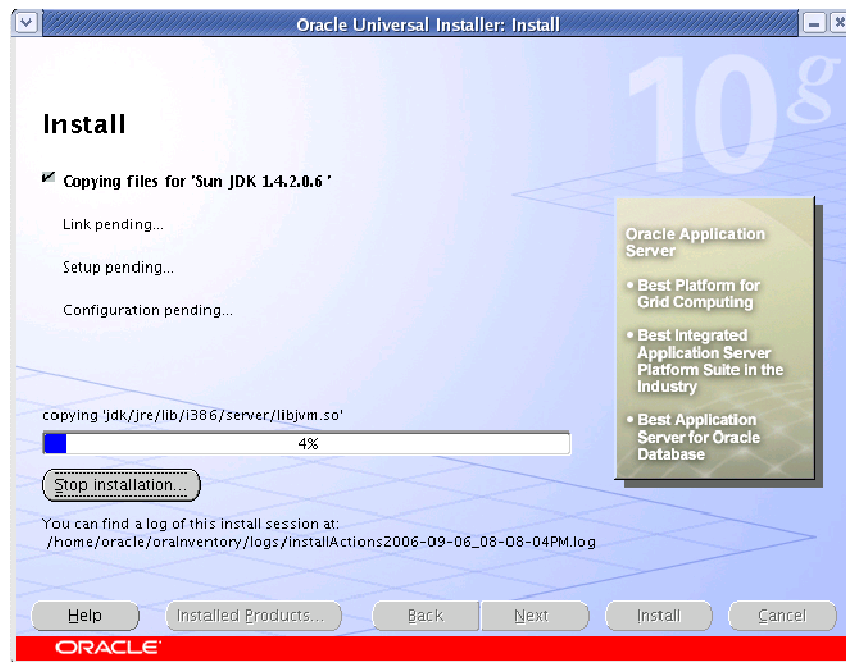
b) Nombre de instancia para servidores de aplicaciones Externos.



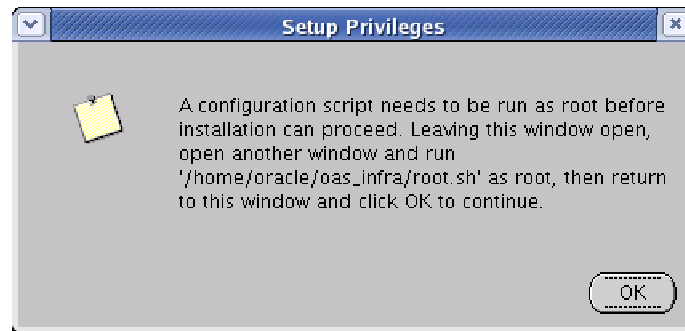
16. Se genera un resumen de la instalación y si no hay observaciones, se comienza la instalación.



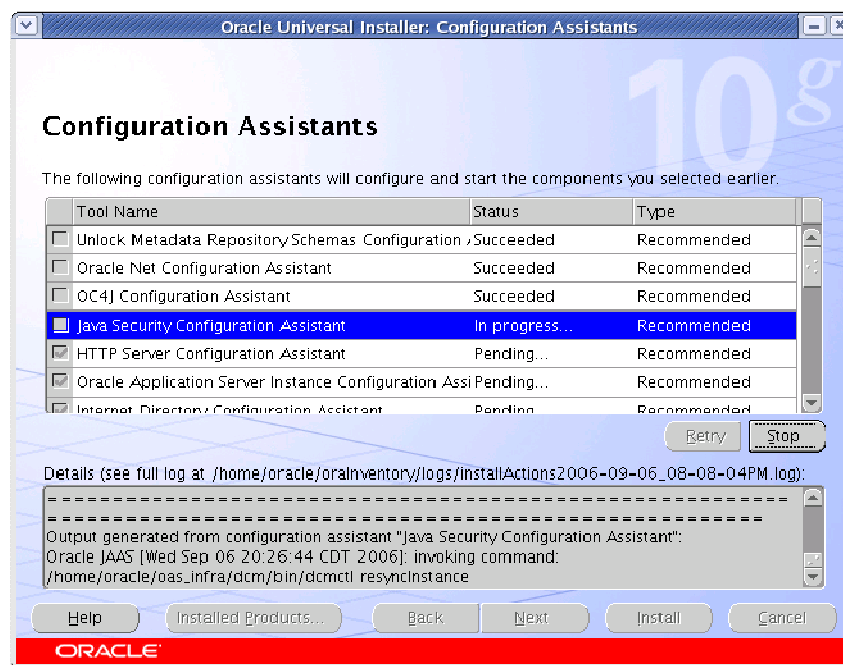
17. Progreso de la copia de archivos.



18. Se requiere ejecutar un script con los permisos de root.

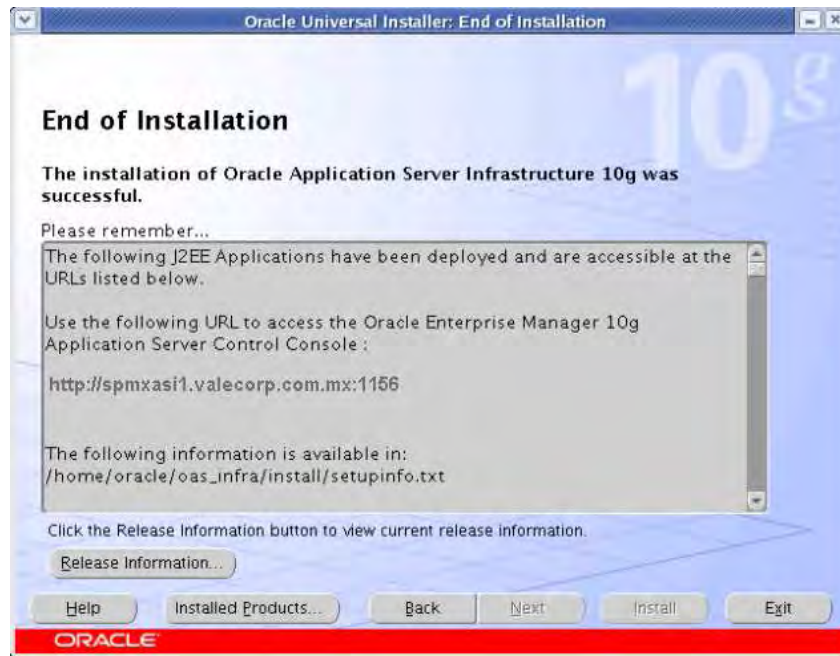


19. Después de copiar todos los archivos, se inicia la configuración de los componentes seleccionados.





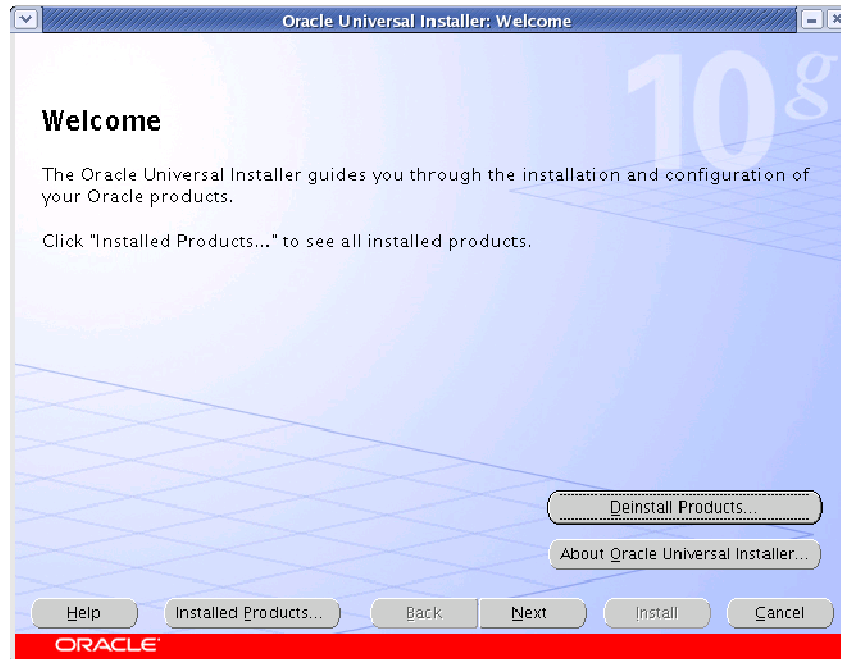
20. Si no hay problemas en la configuración, el asistente muestra el resumen de la instalación y configuración, dando fin a esta.



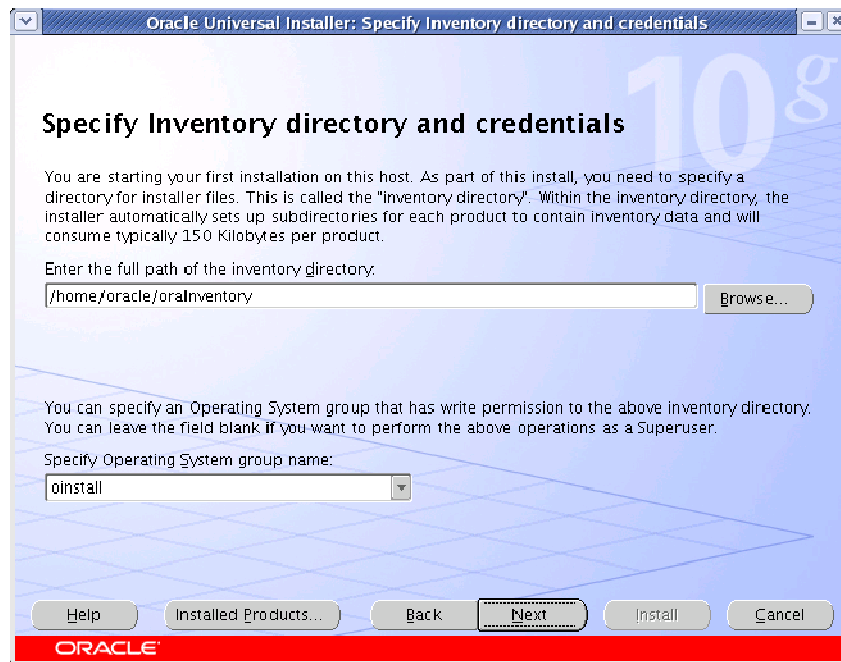
### 3.5 Instalación de instancia "Identity Management " en Nodo 2 y unión a cluster Nodo 1

En el Nodo 2 se hace una instalación con los mismos componentes de Identity Management, esta instalación es similar a la de el Nodo 1, la diferencia es que el cluster se creó desde el Nodo 1, y ahora solo se agregará esta instancia al cluster creado. El procedimiento para crear la instancia es el siguiente:

1. Descargar y desempacar el software de "Oracle Application Server 10g R2" en una carpeta con permisos para que el usuario Oracle del sistema operativo pueda tener acceso a ella. Ya teniendo el software listo, ejecutamos el comando: runInstaller, el cual iniciará el instalador de la aplicación



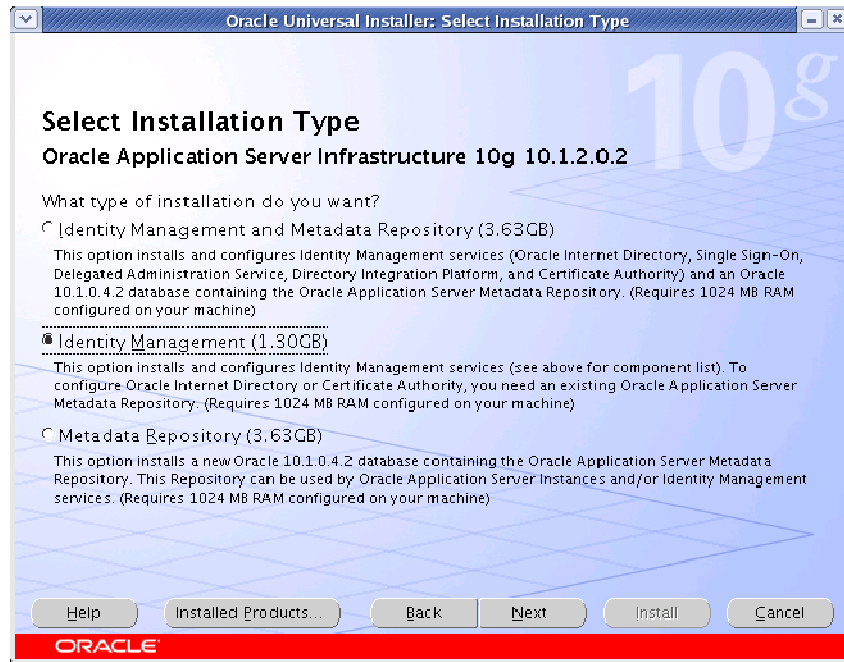
2. Especificamos el origen del software de instalación, el nombre de la instalación y el destino donde se instalara el software.



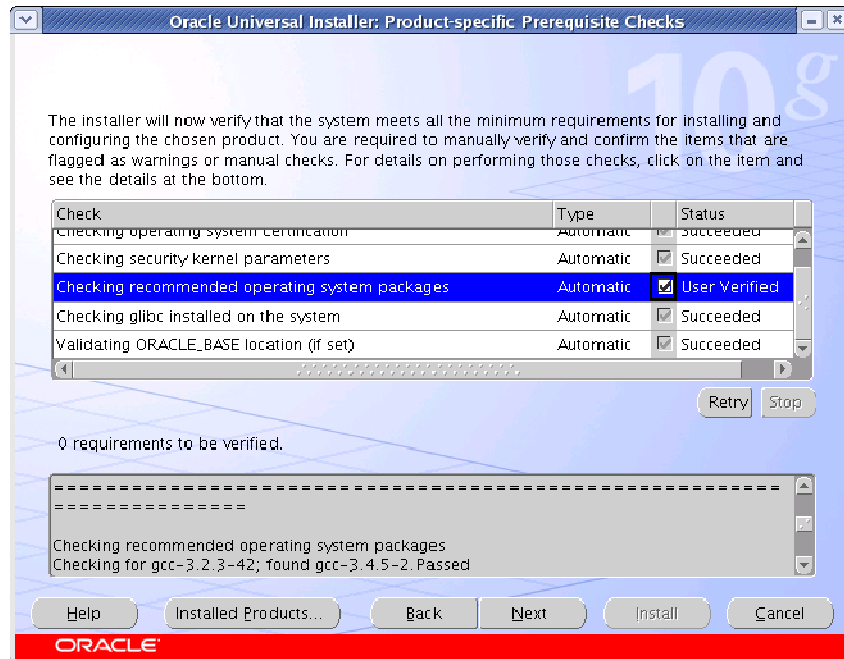
3. Seleccionamos el producto a instalar, para poder comenzar, seleccionamos Infraestructura.



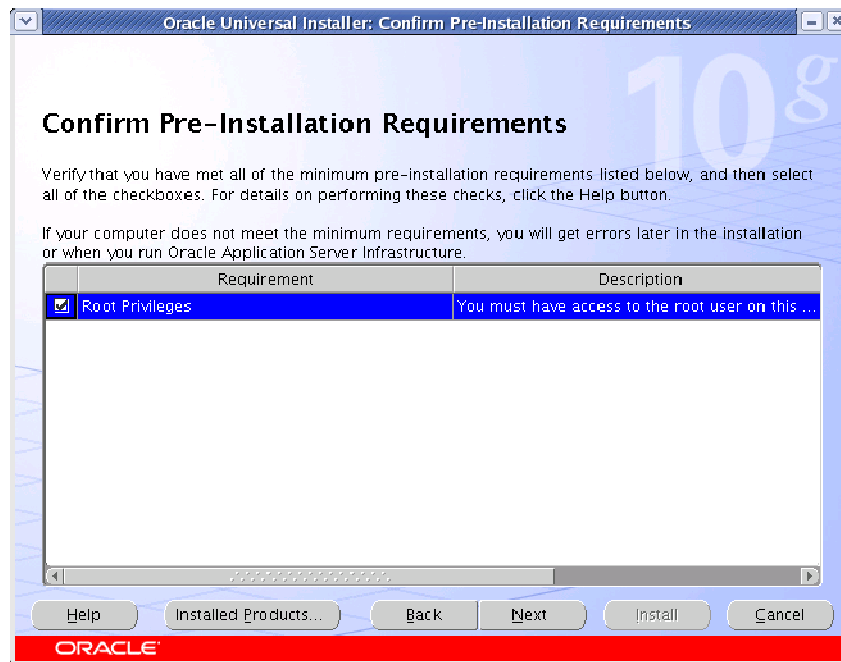
4. Recordemos que anteriormente se creó la base de datos "Metadata Repository", por lo tanto solo necesitamos instalar "Identity Management".



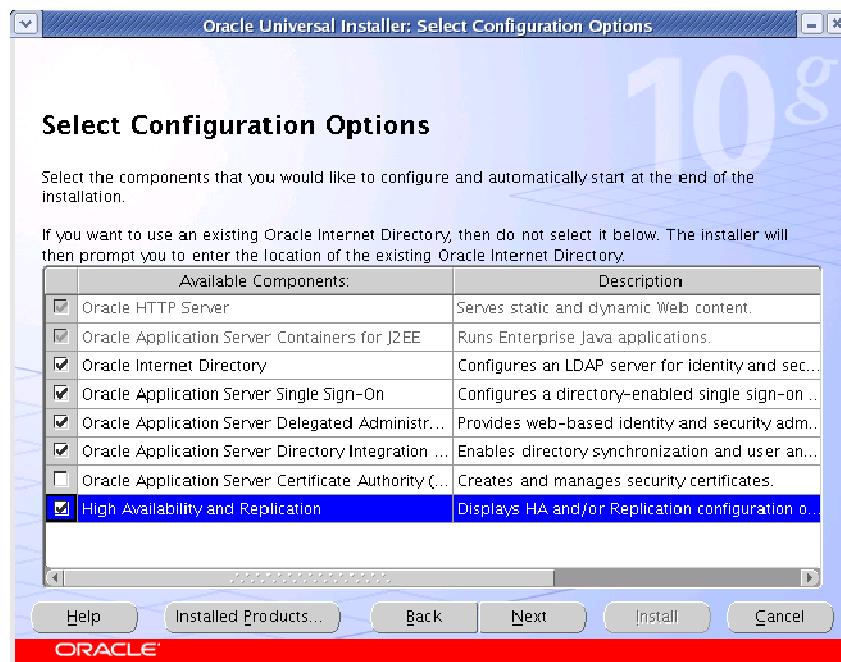
5. El asistente verifica que se cumplan los requerimientos mínimos del sistema, para poder continuar con la instalación.



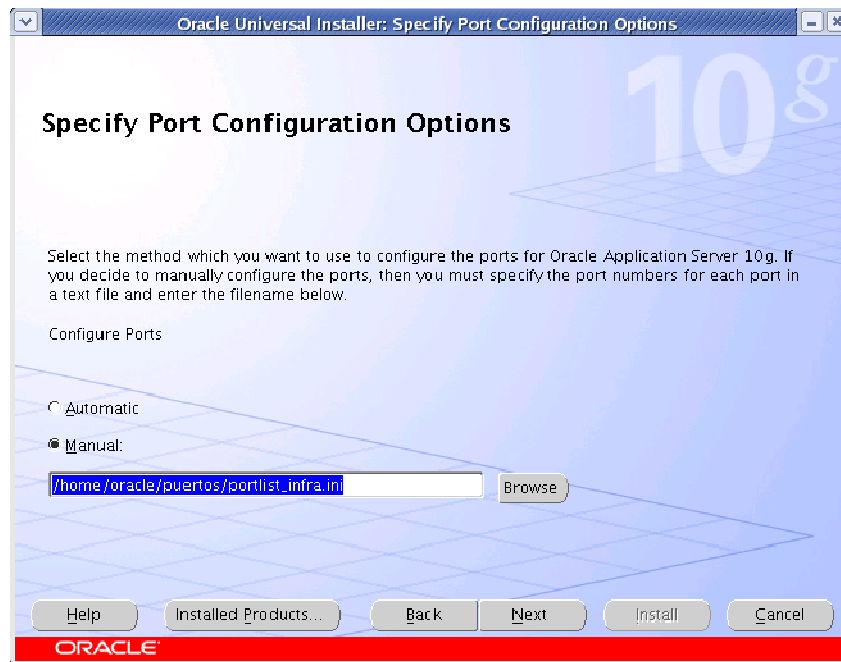
6. Como prerequisite el asistente nos informa que necesitamos tener acceso a la cuenta de root, para ejecutar dos scripts.



7. Seleccionamos los componentes que se desean instalar.

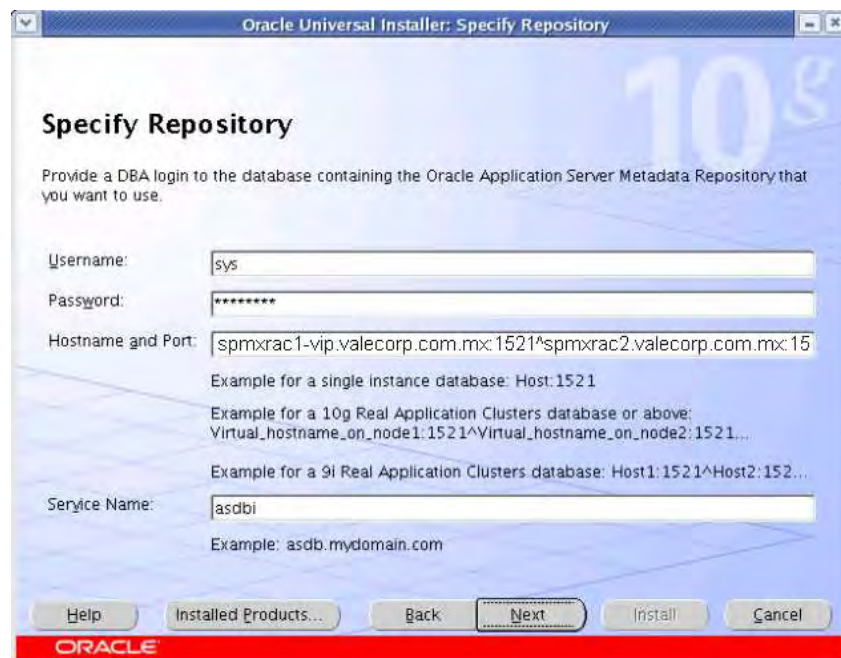


8. Especificamos los puertos que utilizaremos para la instancia "Identity Management", el asistente puede crear los puertos de forma automática, pero como se requiere tener un control de ello, se genero anteriormente un archivo de puertos llamado portlist\_infra.ini.

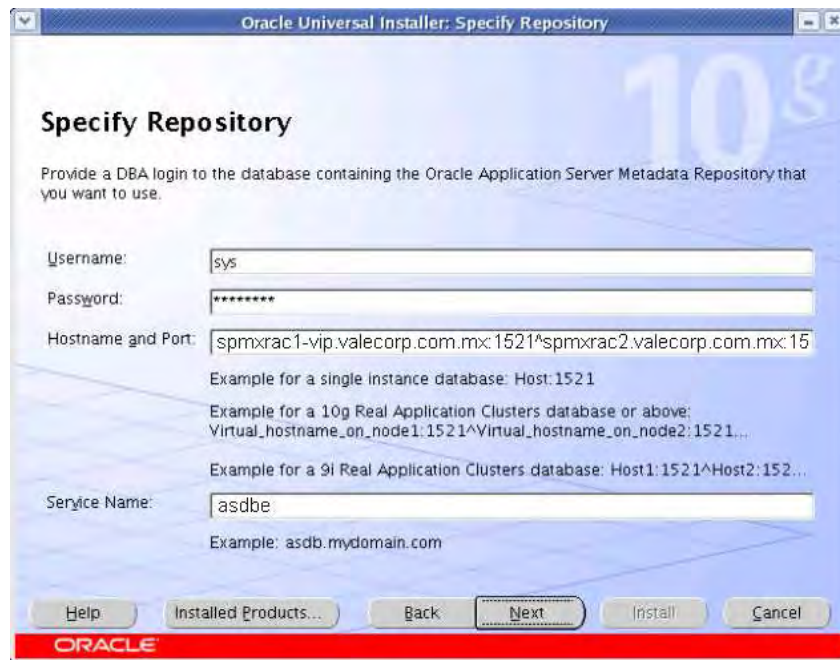


9. El asistente requiere tener acceso a la base de datos, para poder hacer la carga y configuración de esta instancia, por lo que tenemos que indicar la siguiente información:

a) Servicio Base de Datos asdbi, para servidores de aplicaciones Internos.



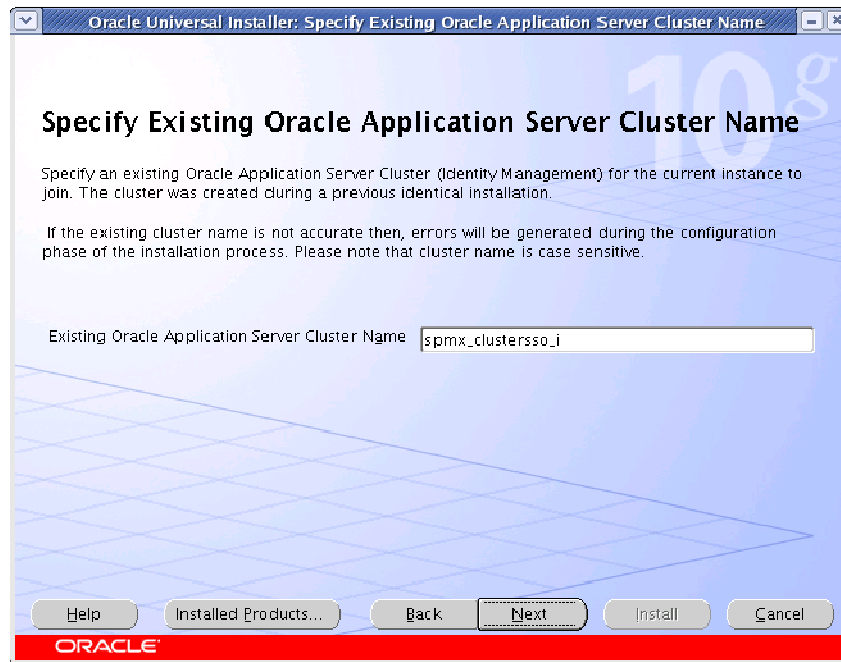
b) Servicio Base de Datos asdbe, para servidores de aplicaciones Externos.



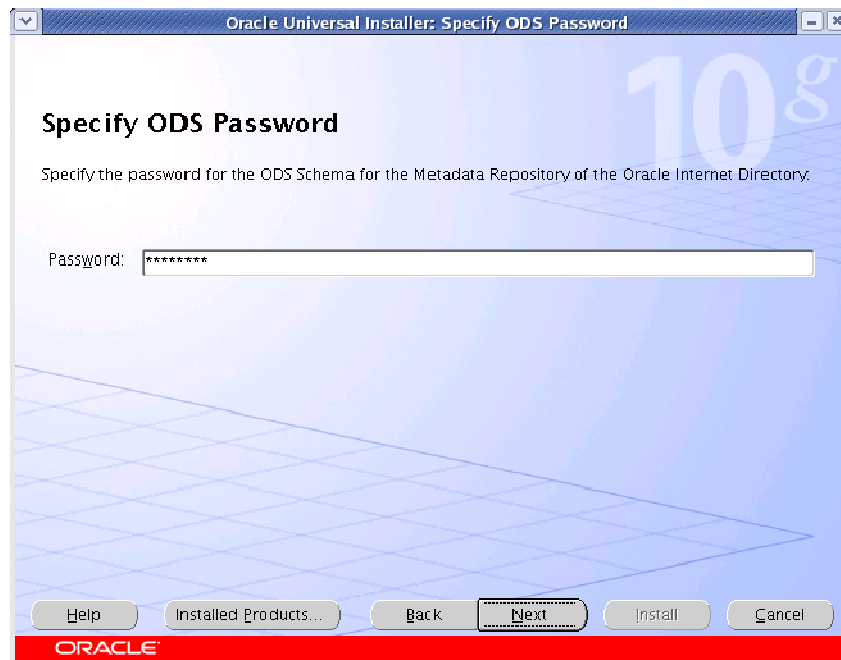
10. El asistente muestra un aviso, acerca de la sincronización del tiempo, la sincronización del tiempo la hicimos desde la configuración del sistema operativo, por lo tanto



11. Especificamos el nombre del cluster que se creó en la instalación anterior.



12. Especificamos el password del cluster creado.



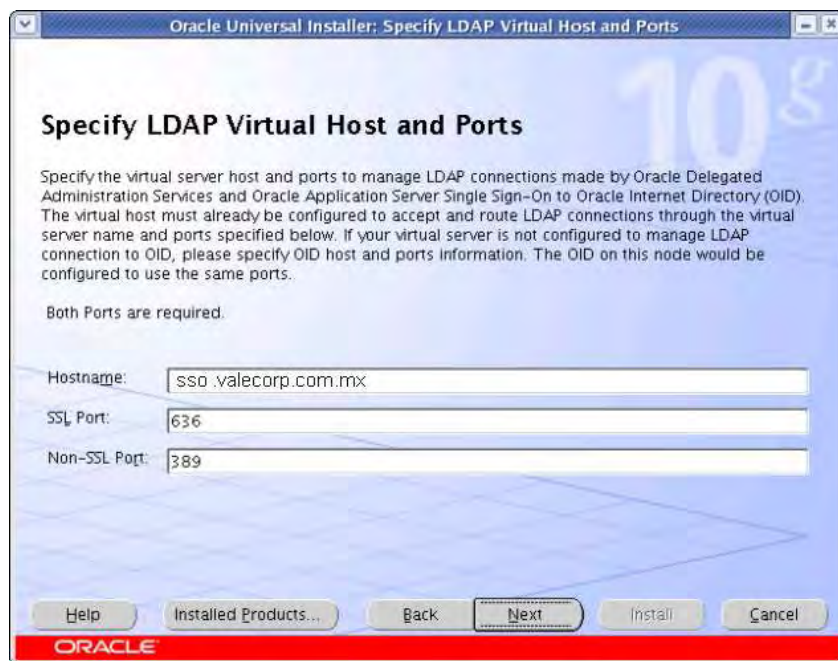


13. Especificamos el nombre del equipo por el que se tendrá acceso a la instancia de single sign on (sso), este nombre tiene que estar registrado en el DNS o en el archivo /etc/hosts, apuntando a la IP del balanceador y especificamos el puerto que se utilizaran.

a) Host Virtual para sso de servidores de aplicaciones Internos.



b) Host Virtual para sso de servidores de aplicaciones Externos.



## 15. Especificamos el usuario y password de Internet Directory



The screenshot shows a window titled "Oracle Universal Installer: Specify Oracle Internet Directory Login". The main heading is "Specify Oracle Internet Directory Login". Below the heading, there is instructional text: "Enter your username and password to connect/login to the Oracle Internet Directory at the hostname and port ssoi.valecorp.com.mx:389. You need to be the Oracle Internet Directory Superuser or a Single Sign-On user. Use cn=orcladmin as the username if you are the Oracle Internet Directory Superuser. Use your Single Sign-On username if you are a Single Sign-On user with the appropriate install privileges." There are two input fields: "Username:" with the value "cn=orcladmin" and "Password:" with the value "\*\*\*\*\*". At the bottom, there are buttons for "Help", "Installed Products...", "Back", "Next", "Install", and "Cancel". The Oracle logo is visible at the bottom left.

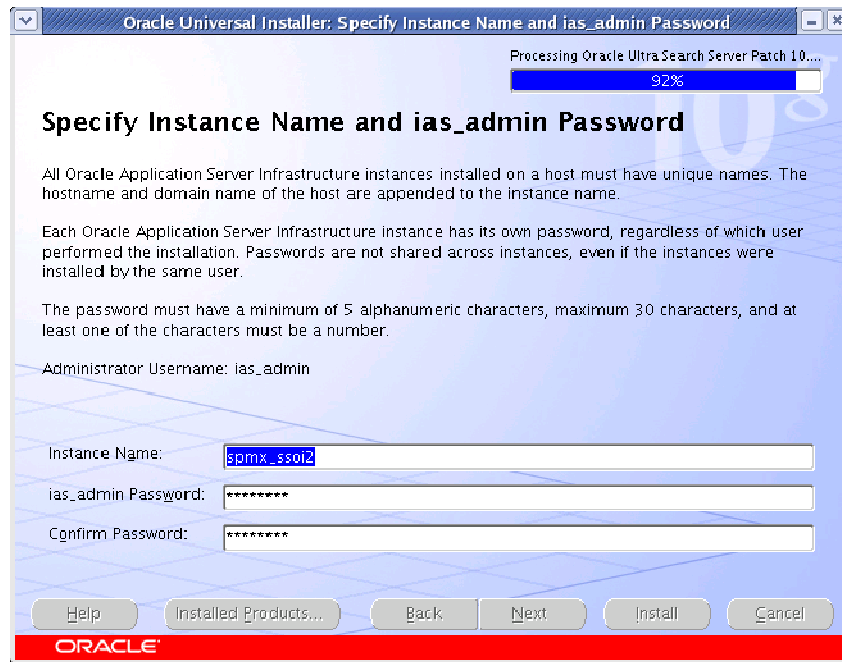
## 16. Especificamos el nombre del equipo por el que se tendrá acceso al servidor Web, y sus puertos.



The screenshot shows a window titled "Oracle Universal Installer: Specify HTTP Listen Port, Load Balancer Host and Port". The main heading is "Specify HTTP Listen Port, Load Balancer Host and Port". Below the heading, there is instructional text: "Specify HTTP Listen Port, Load Balancer Host and Port to manage HTTP connections made by client applications to Oracle Delegated Administration Services and Oracle Application Server Single Sign-On. Note that when you enable SSL (Secure Socket Layer) for the HTTP Listen port, the HTTP load balancer port will need to be also SSL enabled. The HTTP Load Balancer Hostname and port number must be the same for all the OracleAS Cluster (Identity Management) installations that use the same Metadata Repository." There are two sections: "HTTP Listener:" with a "Port:" field containing "7777" and an unchecked "Enable SSL" checkbox; and "HTTP Load Balancer:" with a "Hostname:" field containing "ssoi.valecorp.com.mx", a "Port:" field containing "7777", and an unchecked "Enable SSL" checkbox. At the bottom, there are buttons for "Help", "Installed Products...", "Back", "Next", "Install", and "Cancel". The Oracle logo is visible at the bottom left.

17. Especificamos el nombre de la instancia para "Identity Management" del Nodo 1 y el password para la administración de esta.

a) Nombre de instancia para servidores de aplicaciones Internos.



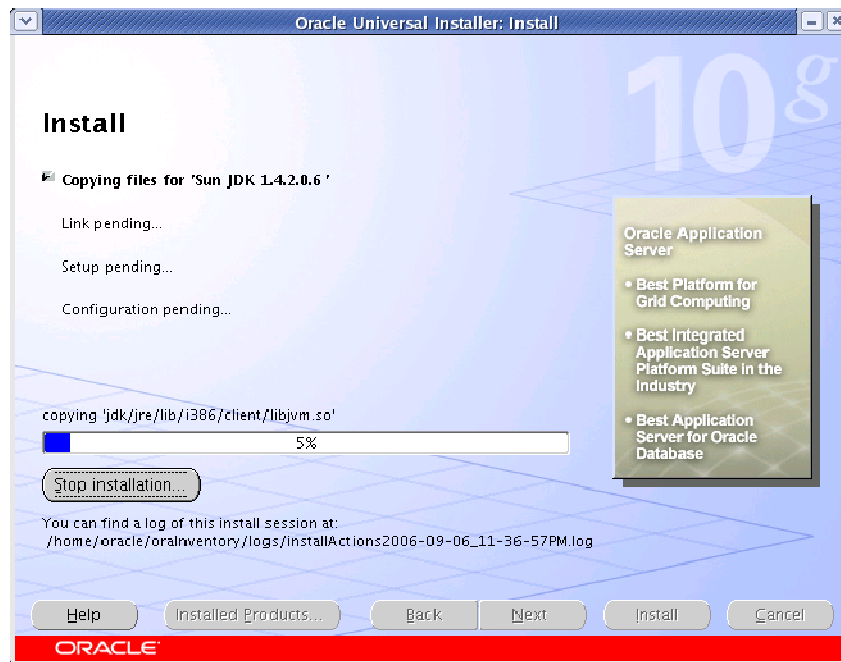
b) Nombre de instancia para servidores de aplicaciones Externos.



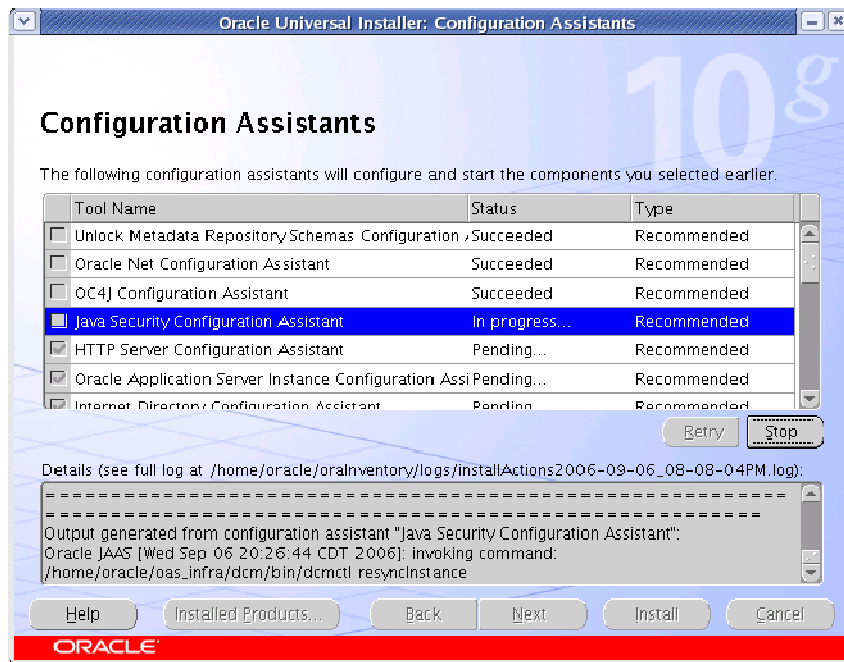
18. Se genera un resumen de la instalación y si no hay observaciones, se comienza la instalación.



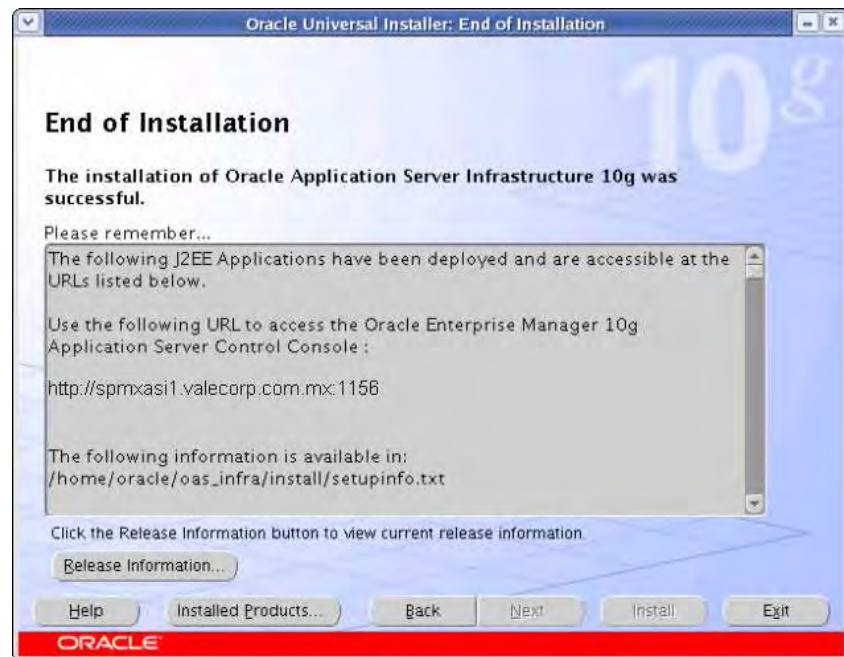
19. Progreso de la copia de archivos



20. Después de copiar todos los archivos, se inicia la configuración de los componentes seleccionados



21. Si no hay problemas en la configuración, el asistente muestra el resumen de la instalación y configuración, dando fin a esta.



### 3.6 Instalación de Middle Tier en NODO 1

Como se menciona en la introducción, Oracle Application Server 10g se divide en dos, Infrastructure y Middle Tier, en el capítulo anterior se hizo la instalación de "Infrastructure", ahora instalaremos "Middle Tier".

Middle Tier, consta de diferentes componentes, en esta instalación solo utilizaremos: Portal, Report Services, y Forms Services.

EL concepto de cluster en Middle Tier, es un poco diferente a la configuración que se hizo en la instalación de "Identity Management", ya que no se puede crear un cluster lógico, por lo que se configuran las aplicaciones para que compartan un mismo repositorio de metadatos, y por medio del cluster de "Identity Management" y el balanceador F5, los clientes pueden usar cualquiera de los dos nodos.

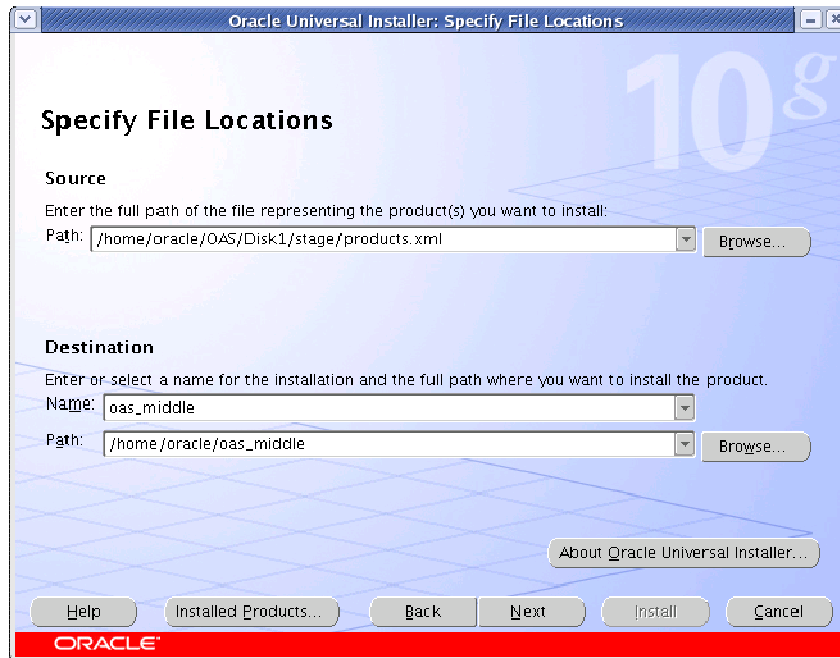
El procedimiento que se utilizó para la instalación de Middle Tier es el siguiente:

1. Ejecutamos nuevamente en comando: runInstaller desde la ruta que contiene los discos desempaquetados de Oracle Application Server 10g R2.





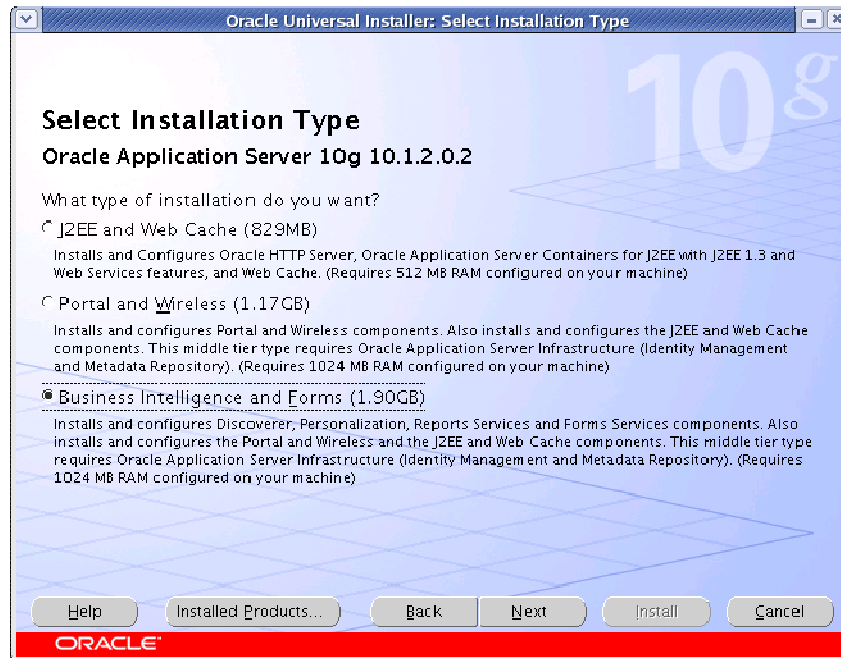
2. Especificamos el origen del software de instalación, el nombre de la instalación y el destino donde se instalara el software.



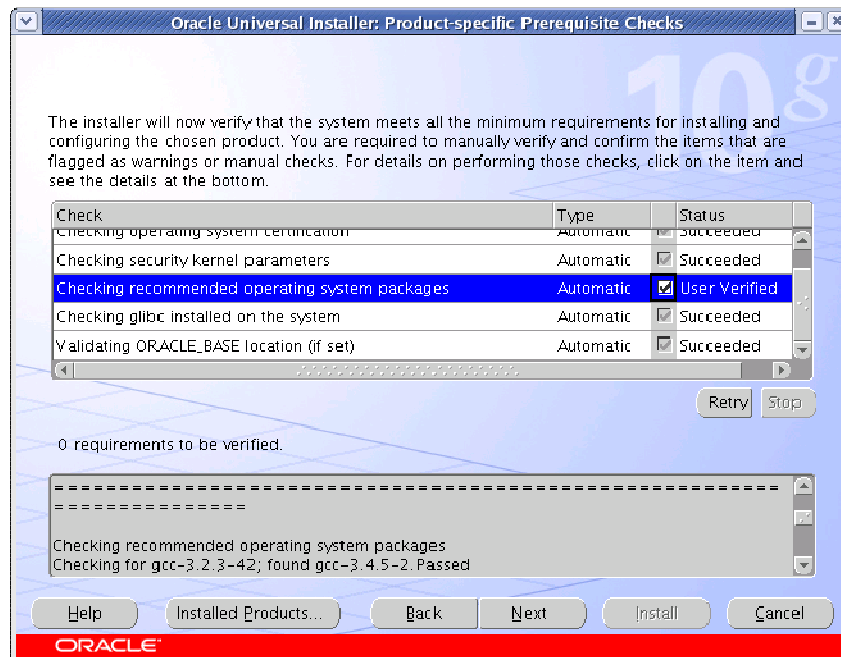
3. Seleccionamos el producto a instalar, como ya contamos con infraestructura, seleccionamos Oracle Application Server 10g 10.1.2.0.2



4. Seleccionamos el tipo de instalación, para este caso se requiere instalar Business Intelligence and Forms.

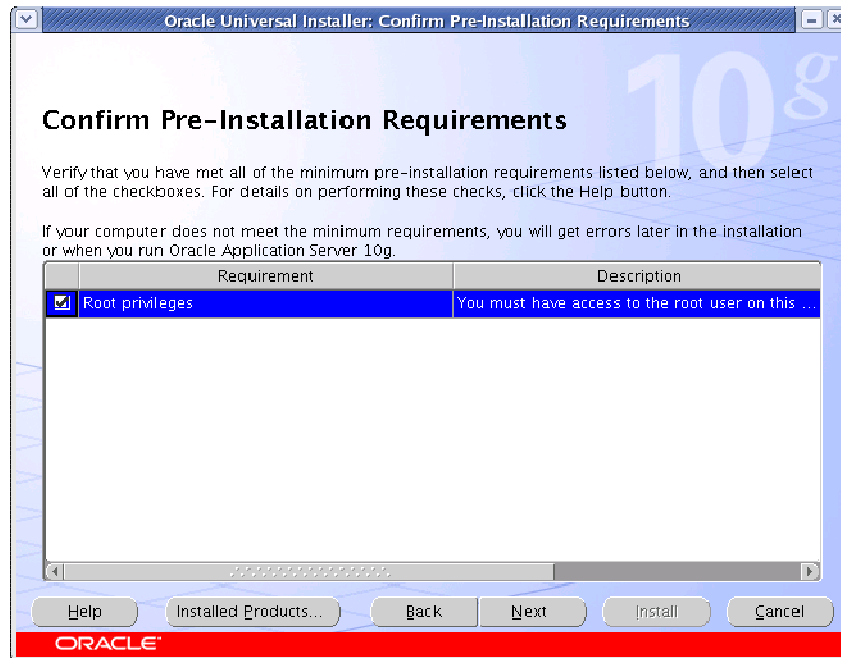


5. El asistente verifica que se cumplan los requerimientos mínimos del sistema, para poder continuar con la instalación.

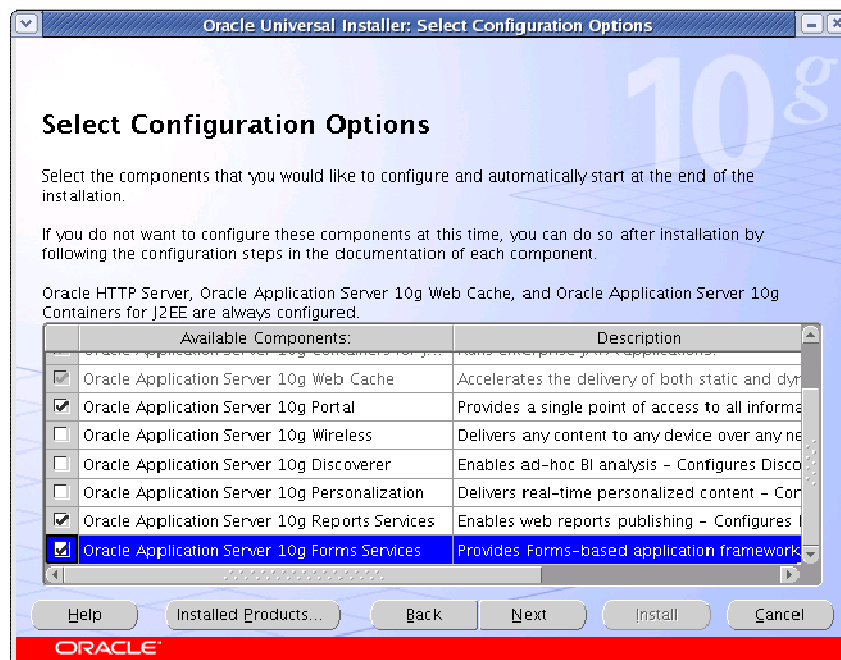




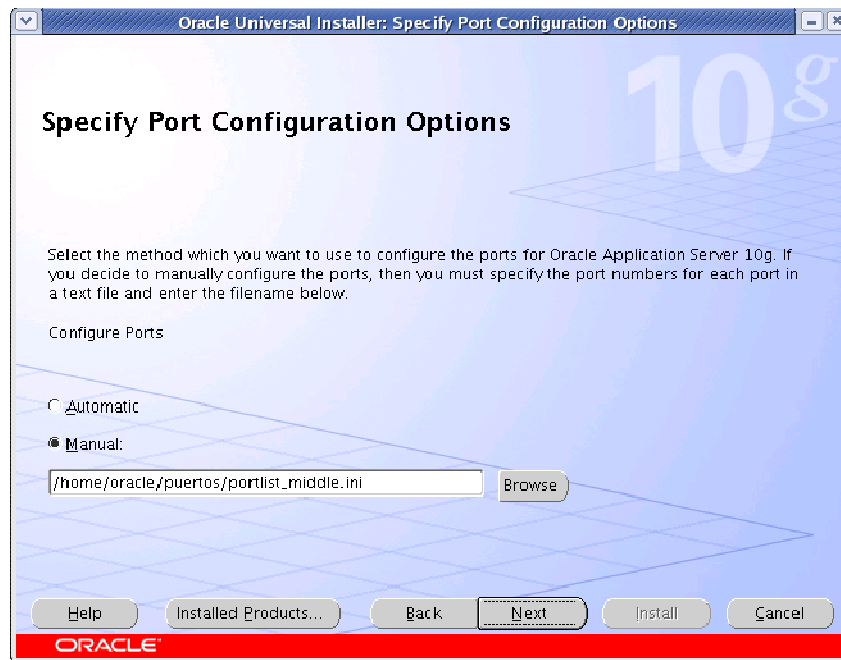
6. Como prerequisite el asistente nos informa que necesitamos tener acceso a la cuenta de root, para ejecutar dos scripts.



7. Seleccionamos los componentes que se desean instalar.



8. Especificamos los puertos que utilizaremos para la instancia "Identity Management", el asistente puede crear los puertos de forma automática, pero como se requiere tener un control de ello, se genero anteriormente un archivo de puertos llamado portlist\_middle.ini.



9. Especificamos el nombre del equipo por el que se tendrá acceso a la instancia de single sign on (sso), este nombre tiene que estar registrado en el DNS o en el archivo /etc/hosts, apuntando a la IP del balanceador y especificamos el puerto que se utilizaran.

a) Host Virtual para sso de servidores de aplicaciones Internos.



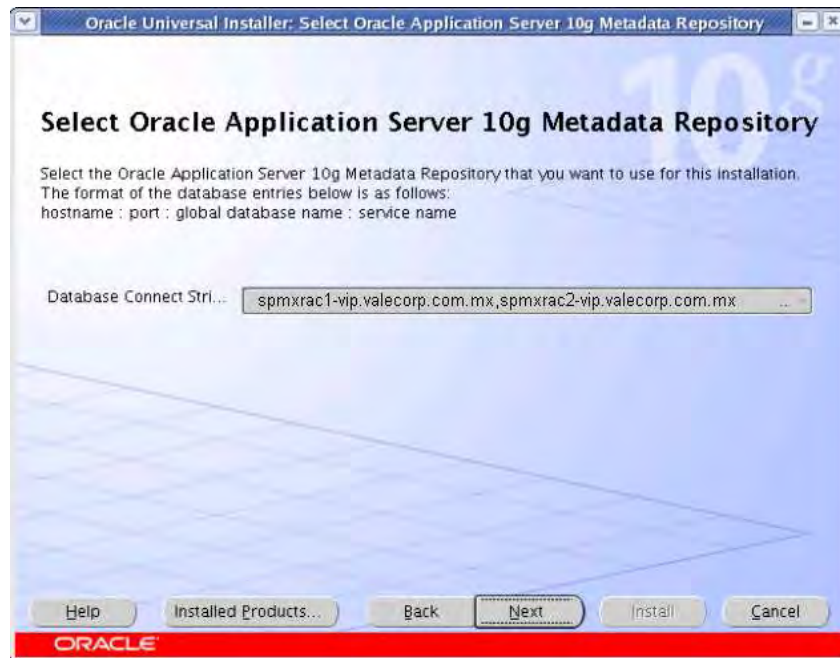
b) Host Virtual para sso de servidores de aplicaciones Externos.



10. Hacemos el registro de OID.



11. Automáticamente nos aparece la cadena de conexión que se utilizara para conectarse a la base de datos.

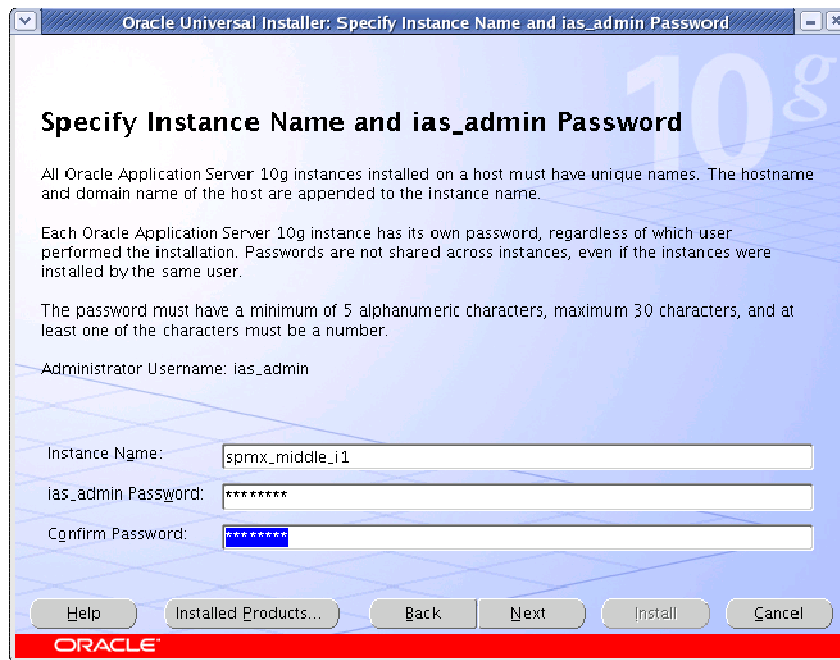


12. Opcionalmente podemos especificar la información de un servidor de correos, en este caso no utilizaremos ninguno.



13. Especificamos en nombre de la instancia y su password.

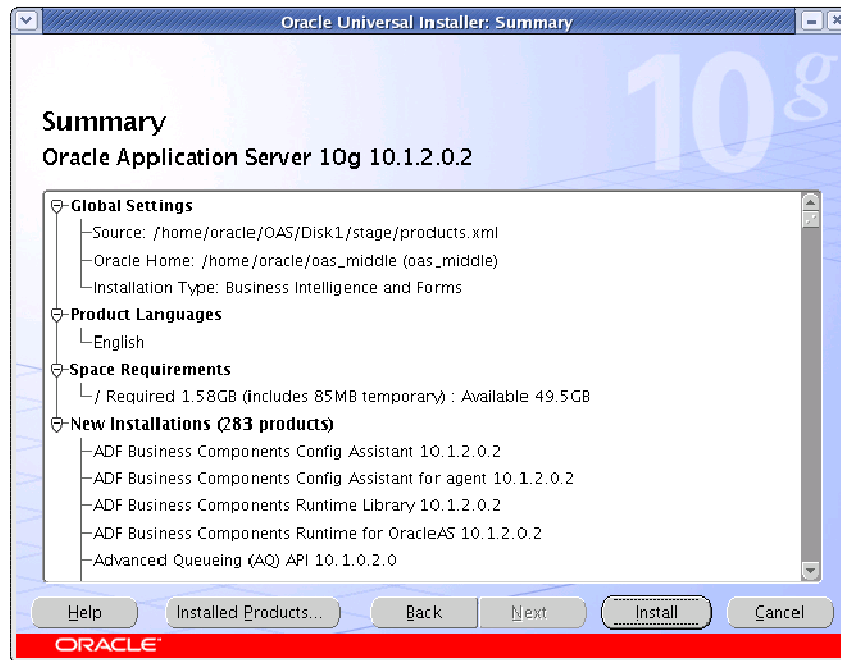
a) Nombre de instancia para servidores de aplicaciones Internos.



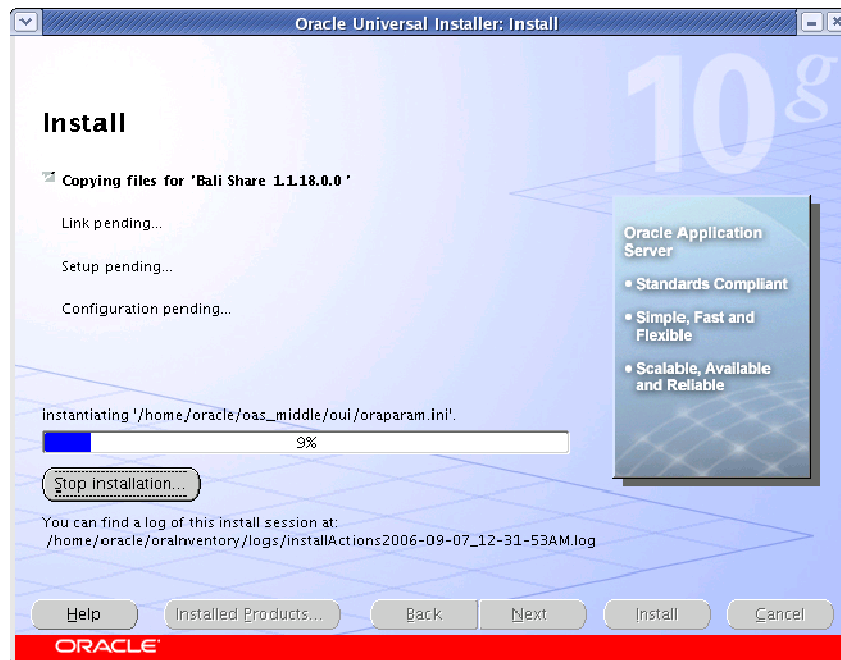
b) Nombre de instancia para servidores de aplicaciones Externos.



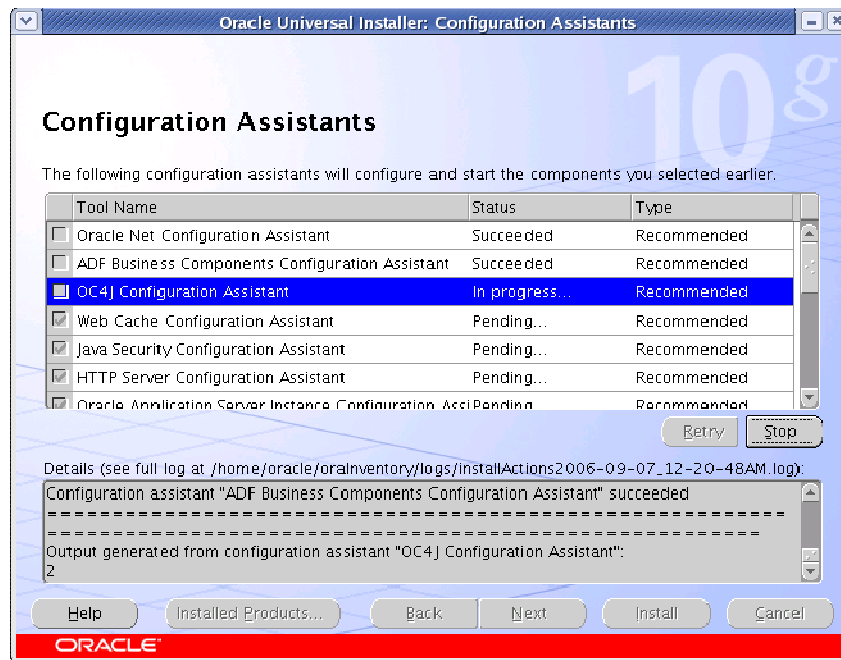
14. Se genera un resumen de la instalación y si no hay observaciones, se comienza la instalación.



15. Progreso de la copia de archivos.



16. Después de copiar todos los archivos, se inicia la configuración de los componentes seleccionados.



16. Si no hay problemas en la configuración, el asistente muestra el resumen de la instalación y configuración, dando fin a esta.

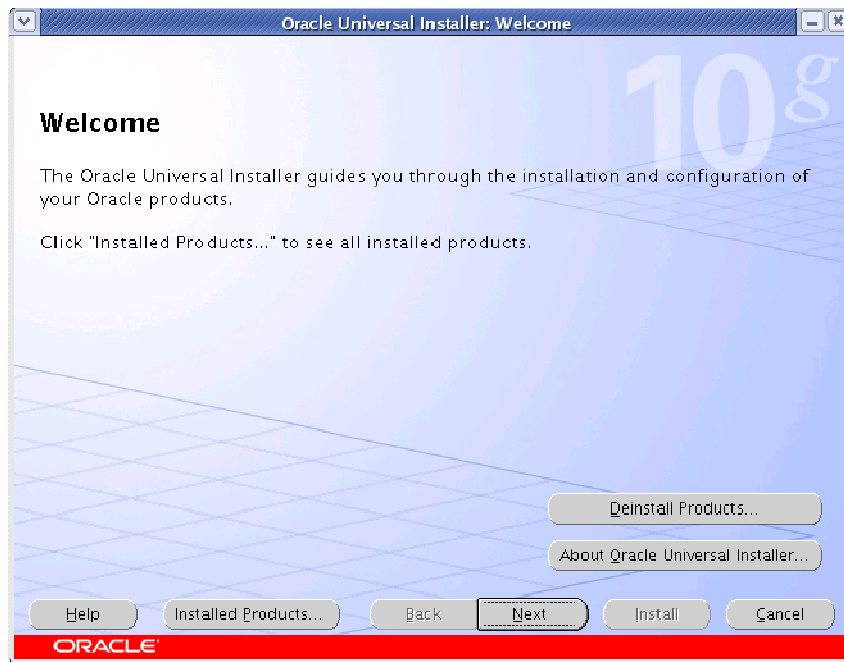




### 3.7 Instalación de Middle Tier en NODO 2

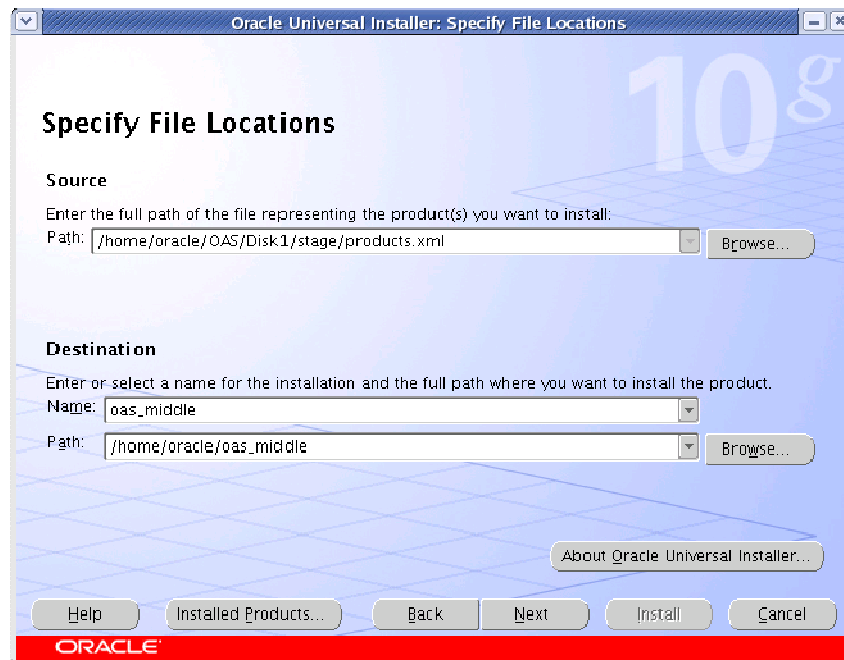
Como se menciona anteriormente, la instalación de Middle Tier es un poco diferente; en el tema anterior se instaló Middle Tier en el NODO1, ahora se instalará en el NODO 2, pero con la diferencia que solo se instalará el software, es decir sin activar ningún componente, esto es porque se configurarán más adelante para que puedan funcionar de forma unida los dos nodos.

1. Ejecutamos nuevamente en comando: runInstaller desde la ruta que contiene los discos desempacados de Oracle Application Server 10g R2.





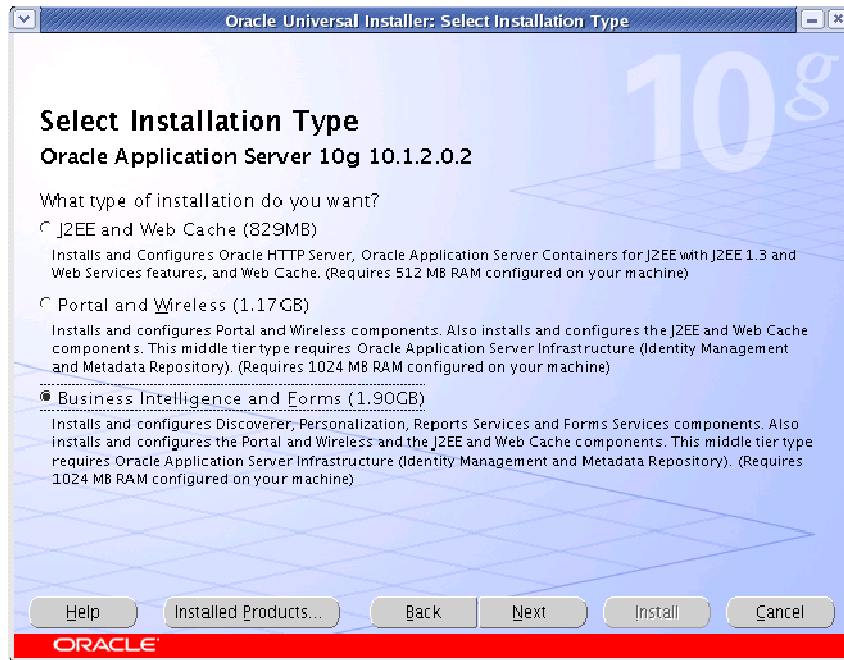
2. Especificamos el origen del software de instalación, el nombre de la instalación y el destino donde se instalara el software.



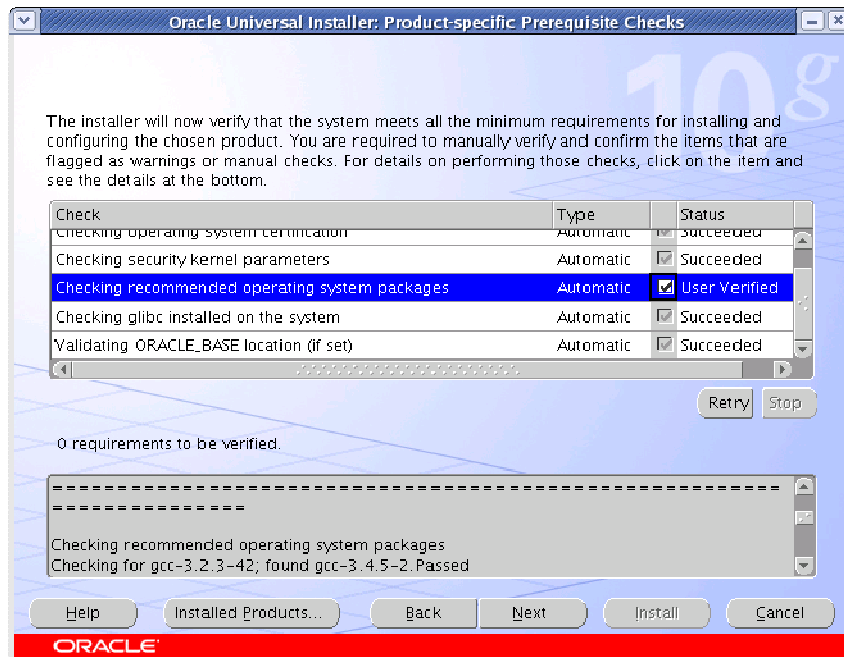
3. Seleccionamos el producto a instalar, como ya contamos con infraestructura, seleccionamos Oracle Application Server 10g 10.1.2.0.2



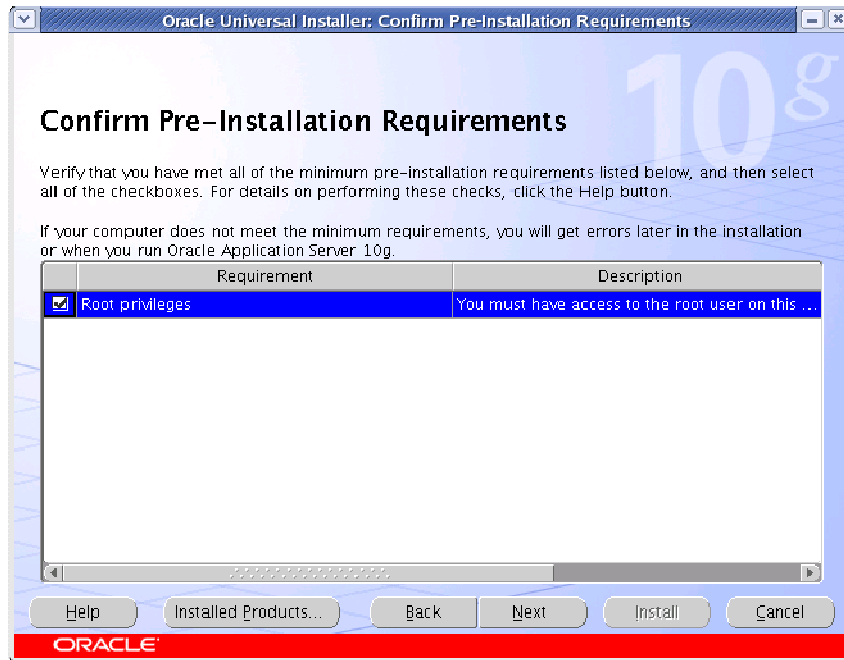
4. Seleccionamos el tipo de instalación, para este caso se requiere instalar Business Intelligence and Forms.



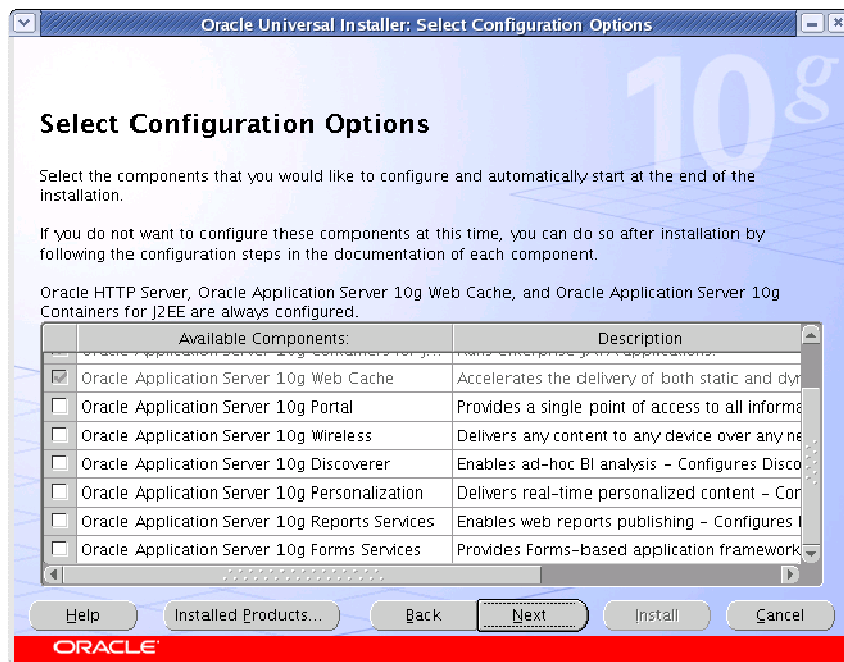
5. El asistente verifica que se cumplan los requerimientos mínimos del sistema, para poder continuar con la instalación.



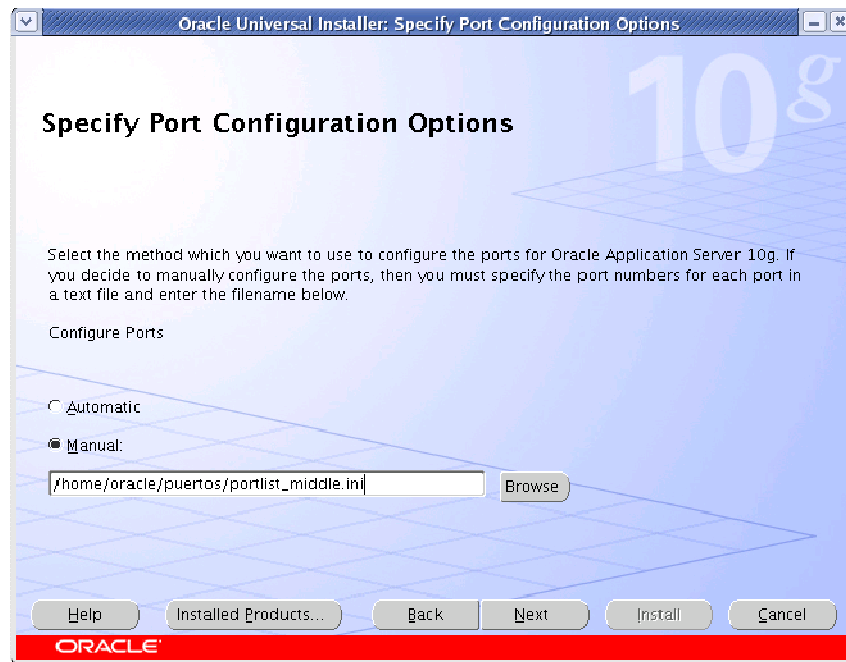
6. Como prerequisite el asistente nos informa que necesitamos tener acceso a la cuenta de root, para ejecutar dos scripts.



7. No seleccionamos componentes, ya que se agregaran mas adelante.



8. Especificamos los puertos que utilizaremos para la instancia "Identity Management", el asistente puede crear los puertos de forma automática, pero como se requiere tener un control de ello, se genero anteriormente un archivo de puertos llamado portlist\_middle.ini.



9. Especificamos el nombre del equipo por el que se tendrá acceso a la instancia de single sign on (sso), este nombre tiene que estar registrado en el DNS o en el archivo /etc/hosts, apuntando a la IP del balanceador y especificamos el puerto que se utilizaran.

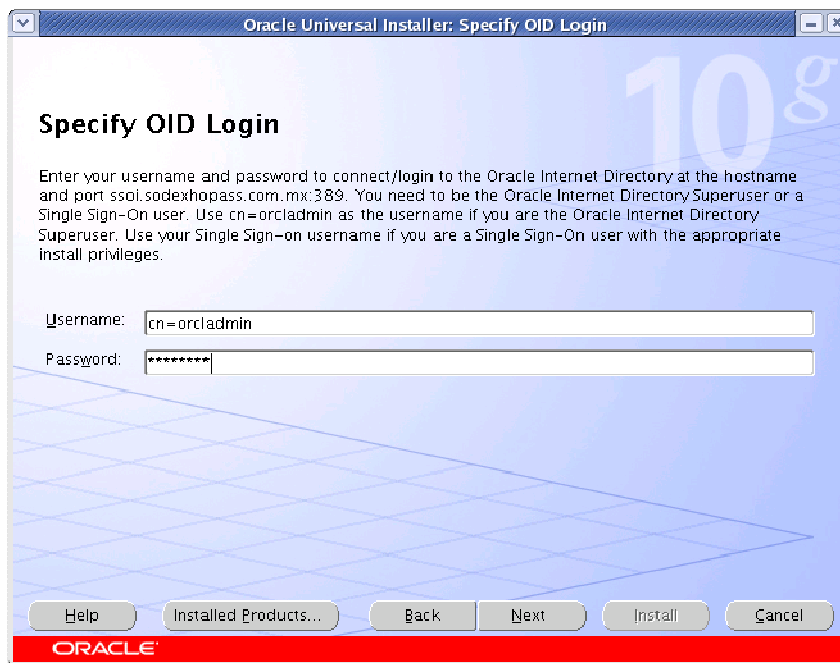
a) Host Virtual para sso de servidores de aplicaciones Internos.



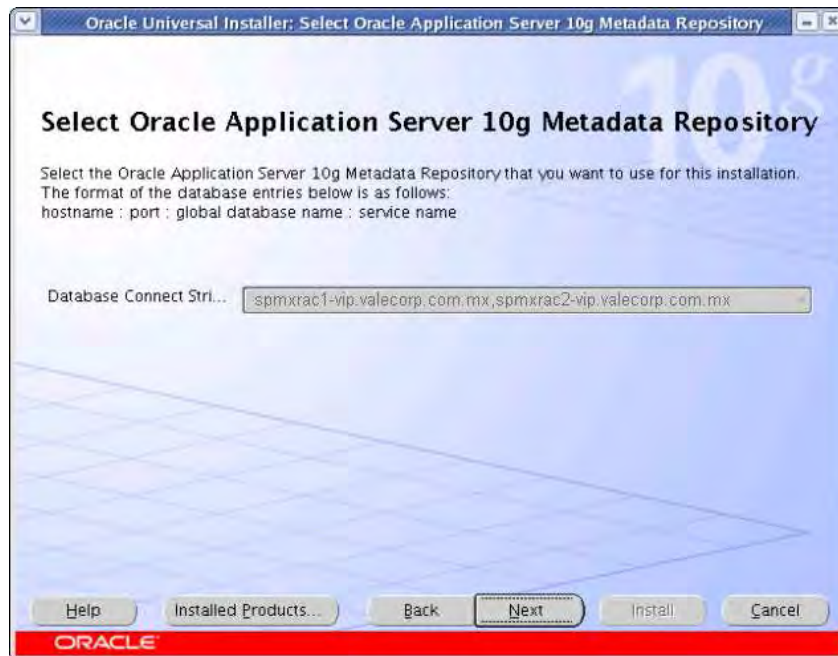
b) Host Virtual para sso de servidores de aplicaciones Externos.



10. Hacemos el registro de OID.

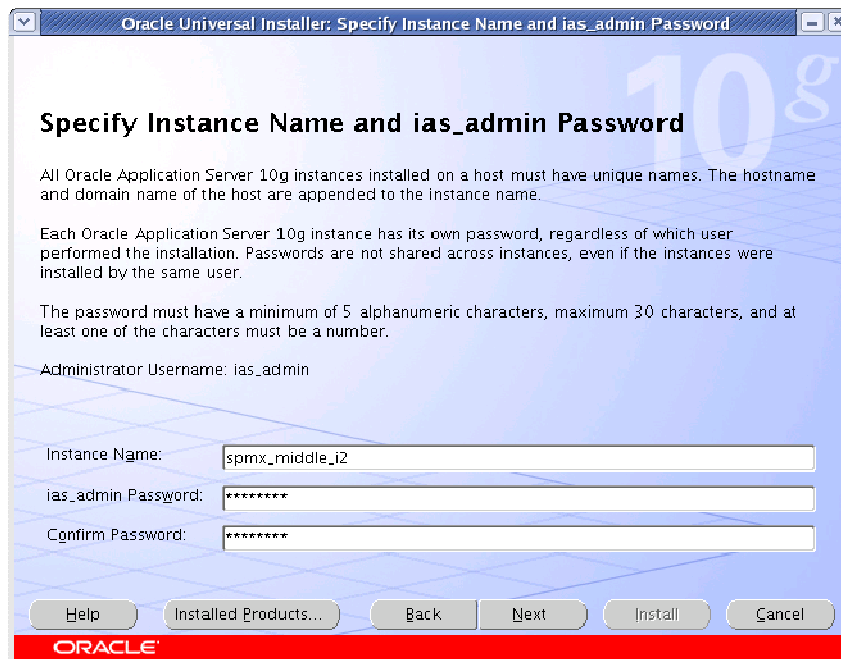


11. Automáticamente nos aparece la cadena de conexión que se utilizara para conectarse a la base de datos



12. Especificamos en nombre de la instancia y su password.

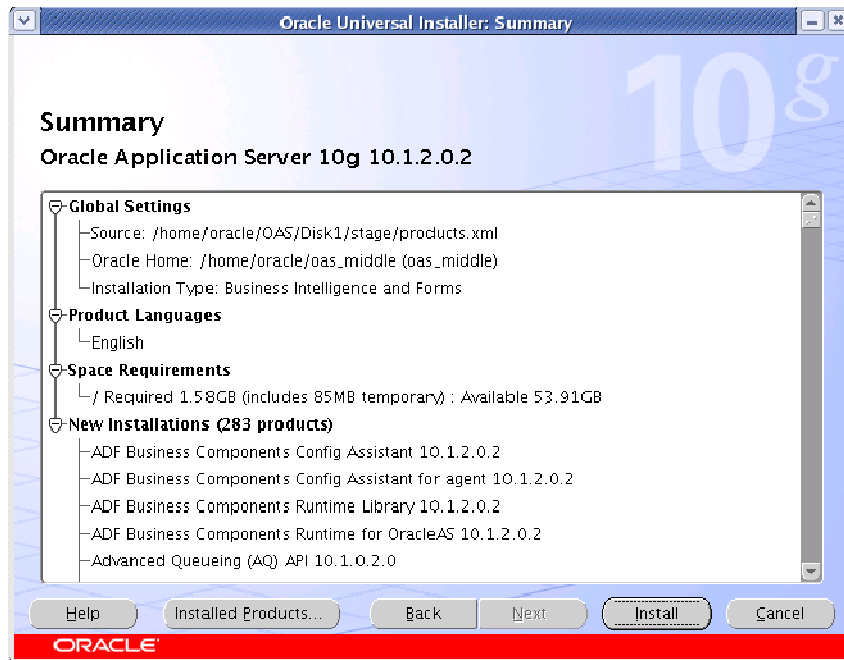
a) Nombre de instancia para servidores de aplicaciones Internos.



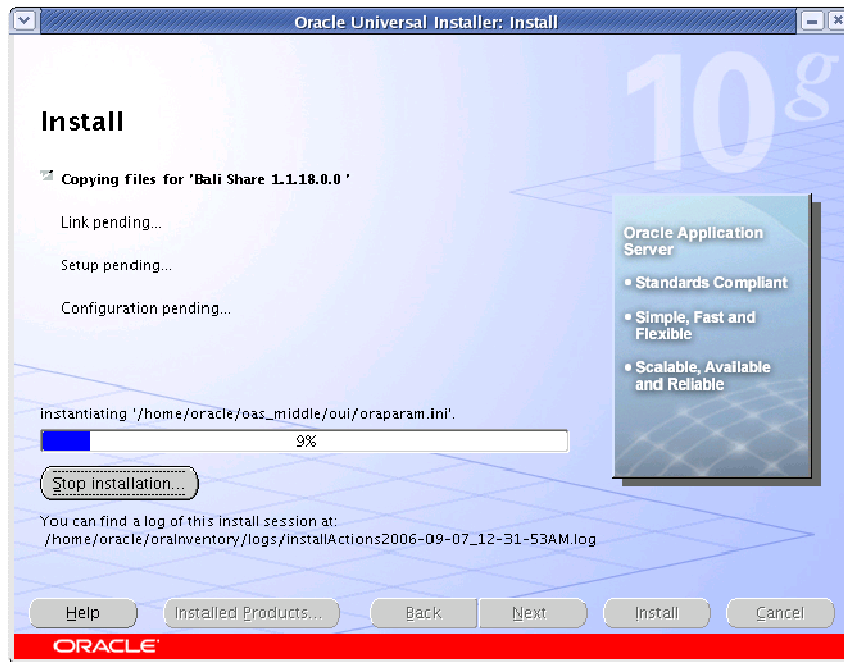
b) Nombre de instancia para servidores de aplicaciones Externos.



13. Se genera un resumen de la instalación y si no hay observaciones, se comienza la instalación.



14. Progreso de la copia de archivos.

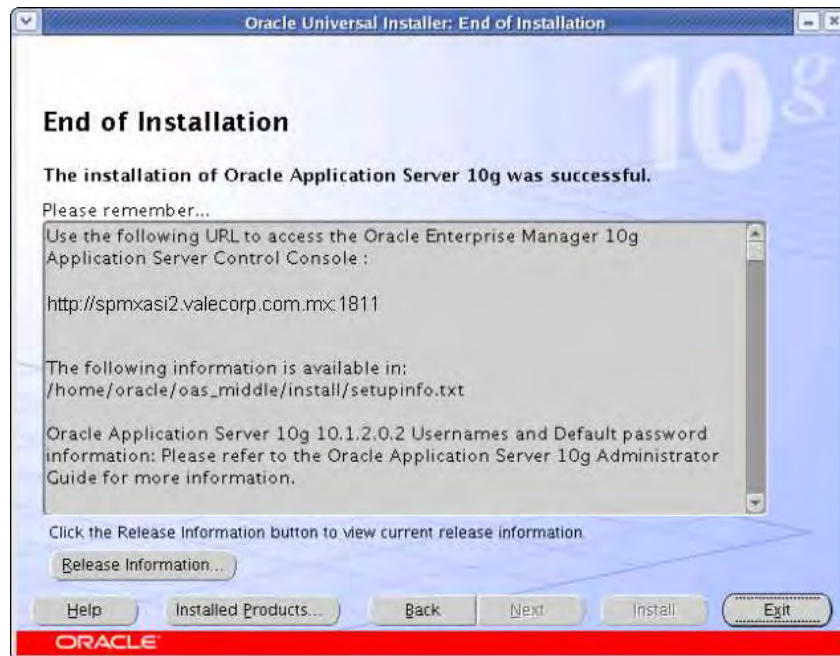


15. Después de copiar todos los archivos, se inicia la configuración de los componentes seleccionados





16. Si no hay problemas en la configuración, el asistente muestra el resumen de la instalación y configuración, dando fin a esta



### 3.8 Configuración de Cluster Middle Tier

Después de que se instalaron las instancias de Middle Tier en los dos nodos, se requiere hacer algunas configuraciones para que puedan funcionar las instancias de los dos nodos, en un mismo repositorio de metadatos. El procedimiento que se hizo para la configuración es el siguiente:

1. Desde el administrador de OAS del Nodo 2, agregamos el componente Portal, con esto el sistema registra que existen dos instancias de Portal en un mismo repositorio.

2. A pesar de que las instancias ya se encuentran en el mismo repositorio es necesario copiar algunos archivos desde el nodo1 y ejecutar algunas instrucciones para registrar la nueva instancia de Portal en Internet Directory.

En la carpeta /home/oracle/scripts se encuentra un archivo llamado copia\_archivos.sh el cual ejecuta lo necesario para que puedan funcionar las dos instancias. Las instrucciones que se requieren ejecutar son las siguientes:

#desde el nodo 2

```
scp oracle@spmxasi1:/home/oracle/oas_middle/Apache/modplsql/conf/dads.conf  
/home/oracle/oas_middle/Apache/modplsql/conf/.
```

```
scp oracle@spmxasi1:/home/oracle/oas_middle/Apache/oradav/conf/oradav.conf  
/home/oracle/oas_middle/Apache/oradav/conf/.
```

```
scp oracle@spmxasi1:/home/oracle/oas_middle/Apache/modplsql/conf/cache.conf  
/home/oracle/oas_middle/Apache/modplsql/conf/.
```

#desde el nodo 2

```
/home/oracle/oas_middle/dcm/bin/dcmctl updateconfig -ct ohs
```

#desde el nodo 2

```
scp oracle@spmxasi1:/home/oracle/oas_middle/portal/conf/iasconfig.xml  
/home/oracle/oas_middle/portal/conf/.
```

#desde el nodo 1

```
scp  
oracle@spmxasi1:/home/oracle/oas_middle/j2ee/OC4J_Portal/applications/portalTools/omniPortlet/WEB-  
INF/providers/omniPortlet/provider.xml  
/home/oracle/oas_middle/j2ee/OC4J_Portal/applications/portalTools/omniPortlet/WEB-  
INF/providers/omniPortlet/provider.xml
```

```
scp  
oracle@spmxasi1:/home/oracle/oas_middle/j2ee/OC4J_Portal/applications/portalTools/webClipping/WEB-  
INF/providers/webClipping/provider.xml  
/home/oracle/oas_middle/j2ee/OC4J_Portal/applications/portalTools/webClipping/WEB-  
INF/providers/webClipping/provider.xml
```

```
scp oracle@spmxasi1:/home/oracle/oas_middle/j2ee/OC4J_Portal/config/data-sources.xml  
/home/oracle/oas_middle/j2ee/OC4J_Portal/config/data-sources.xml
```

```
scp oracle@spmxasi1:/home/oracle/oas_middle/j2ee/OC4J_Portal/config/jazn-data.xml  
/home/oracle/oas_middle/j2ee/OC4J_Portal/config/jazn-data.xml
```

#desde el nodo 2

```
/home/oracle/oas_middle/opmn/bin/opmnctl restartproc process-type=HTTP_Server
```

#desde el nodo 2

```
scp oracle@spmxfasi1:/home/oracle/oas_middle/Apache/Apache/conf/osso/osso.conf  
/home/oracle/oas_middle/Apache/Apache/conf/osso/.
```

#desde el nodo 2

```
/home/oracle/oas_middle/portal/conf/ptlconfig -dad portal -sso -host spmxfasi2.ValeCorp.com.mx -port  
7780
```

```
/home/oracle/oas_middle/sso/bin/ssoreg.sh -site_name spmxfasi2.ValeCorp.com.mx -mod_osso_url  
http://spmxfasi2.ValeCorp.com.mx:7780 -config_mod_osso TRUE -oracle_home_path  
/home/oracle/oas_middle/ -config_file /home/oracle/oas_middle/Apache/Apache/conf/osso/osso.conf -  
admin_info cn=orcladmin
```

3. Después de ejecutar el script anterior, se requiere reiniciar todos los servicios, cuando todos los servicios se han iniciado, se pueden probar las aplicaciones de el Nodo 1 y podemos notar que funcionan sin problemas, desde el nodo 2 probamos las instancias y además de que ya se muestra el componente portal, si accedemos a el, funciona sin problema alguno.

4. De la forma que se agrego el componente Portal, agregamos los dos componentes que faltan, Forms y Reports.

5. Después de agregar los componentes, se reinician los servicios de todos los nodos, y podemos ver que todo funciona sin problemas y que no importa por que nodo se tenga acceso, ya que la información es la misma.

### 3.9 Instalación de Middle Tier J2EE en nodo1

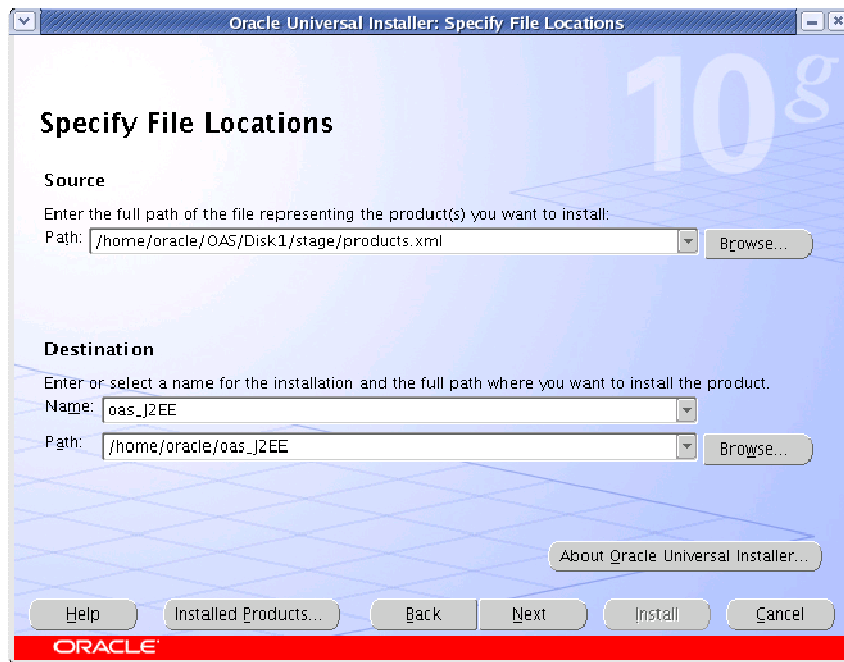
La implementación de Middle Tier J2EE en cluster, resulta un poco más sencilla que las instalaciones anteriores, ya que el procedimiento es de instalar el Middle Tier en ambos nodos, agregar las instancias a una misma granja, crear un cluster lógico y agregar las instancias a ese cluster.

El procedimiento para instalar las instancias de J2EE es el siguiente.

1. Ejecutamos nuevamente en comando: runInstaller desde la ruta que contiene los discos desempacados de Oracle Application Server 10g R2.



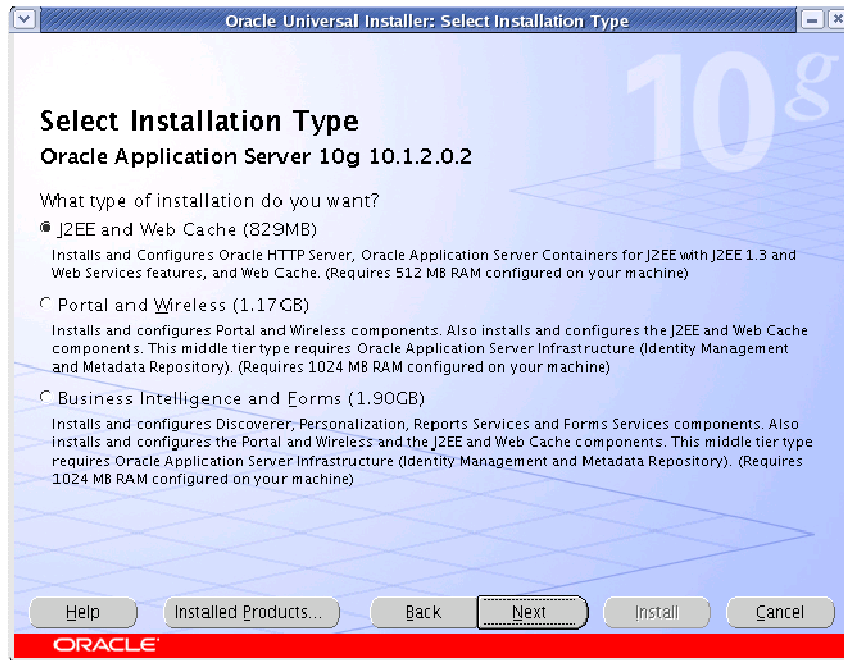
2. Especificamos el origen del software de instalación, el nombre de la instalación y el destino donde se instalara el software.



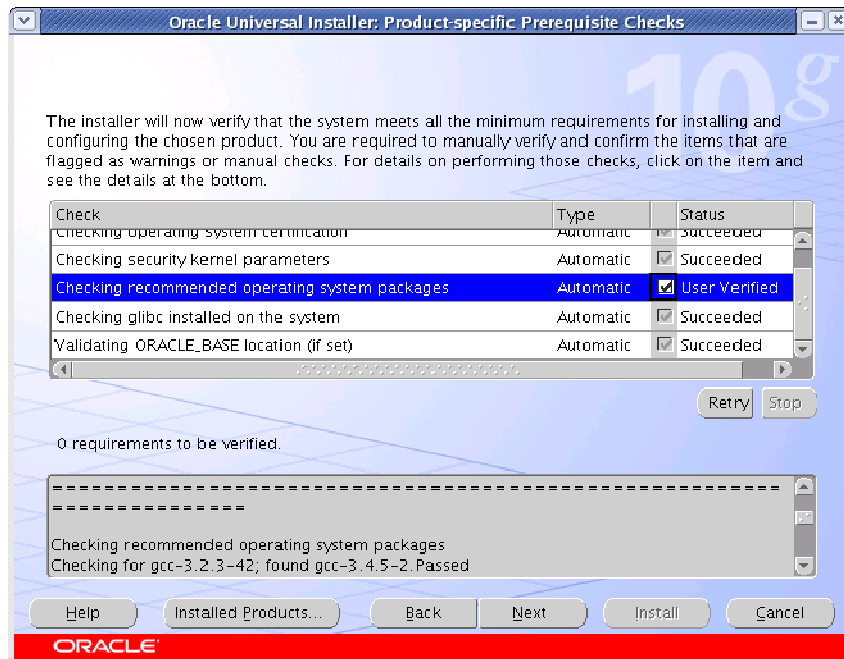
3. Seleccionamos el producto a instalar, como ya contamos con infraestructura, seleccionamos Oracle Application Server 10g 10.1.2.0.2



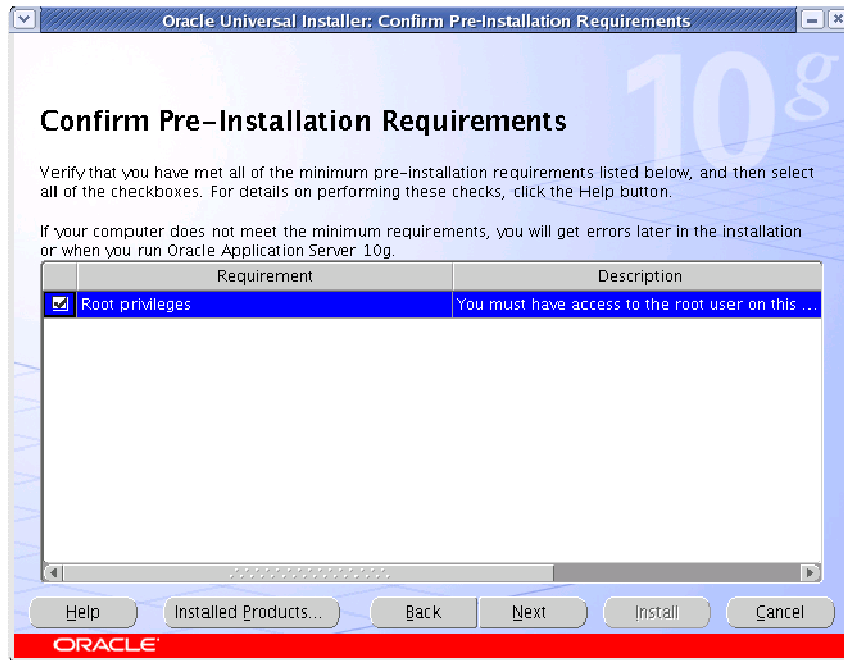
4. Seleccionamos el producto a instalar, estas instancias solo serán contenedoras de J2EE, por lo tanto seleccionamos J2EE and Web Cache



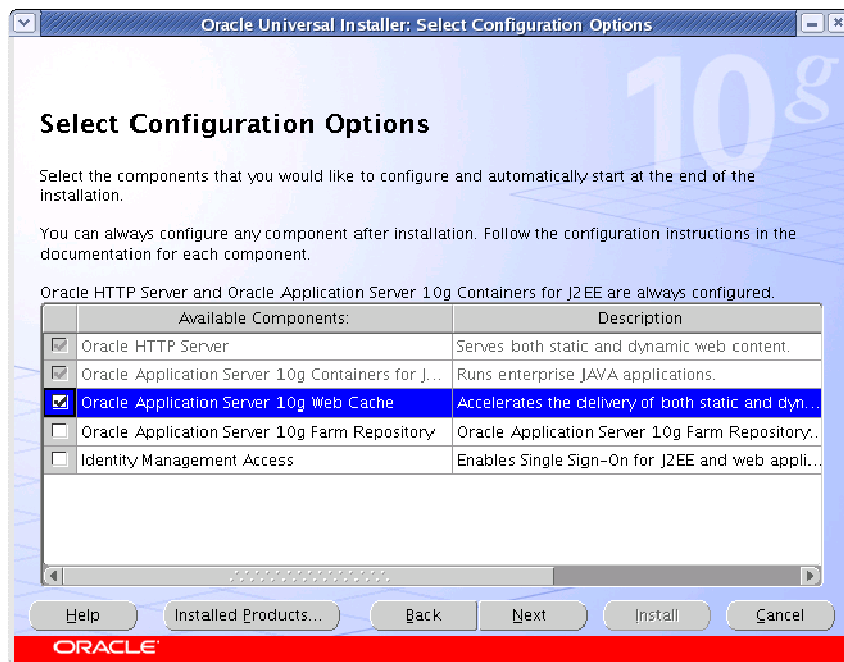
5. El asistente verifica que se cumplan los requerimientos mínimos del sistema, para poder continuar con la instalación



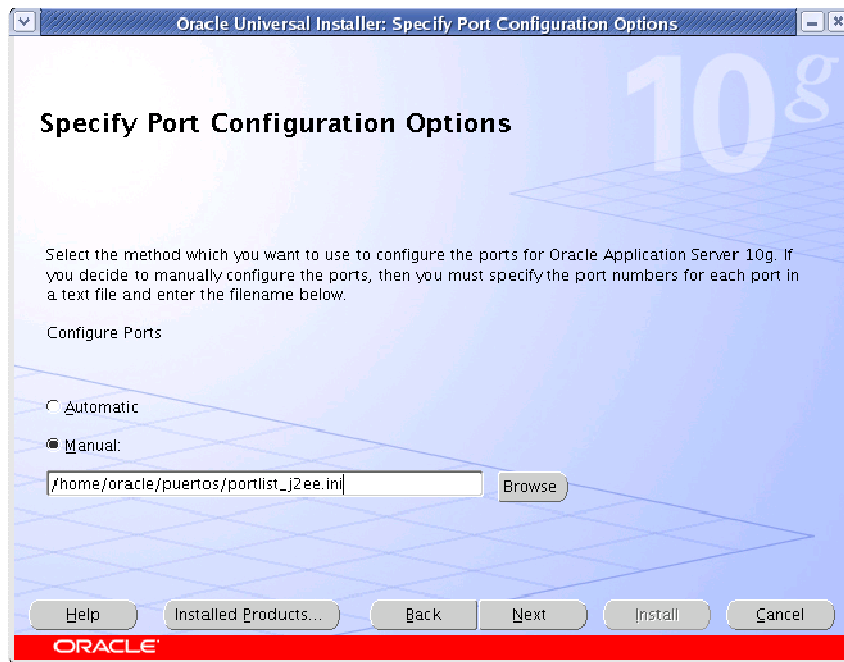
6. Como prerequisite el asistente nos informa que necesitamos tener acceso a la cuenta de root, para ejecutar dos scripts.



7. Seleccionamos los componentes que se desean instalar.

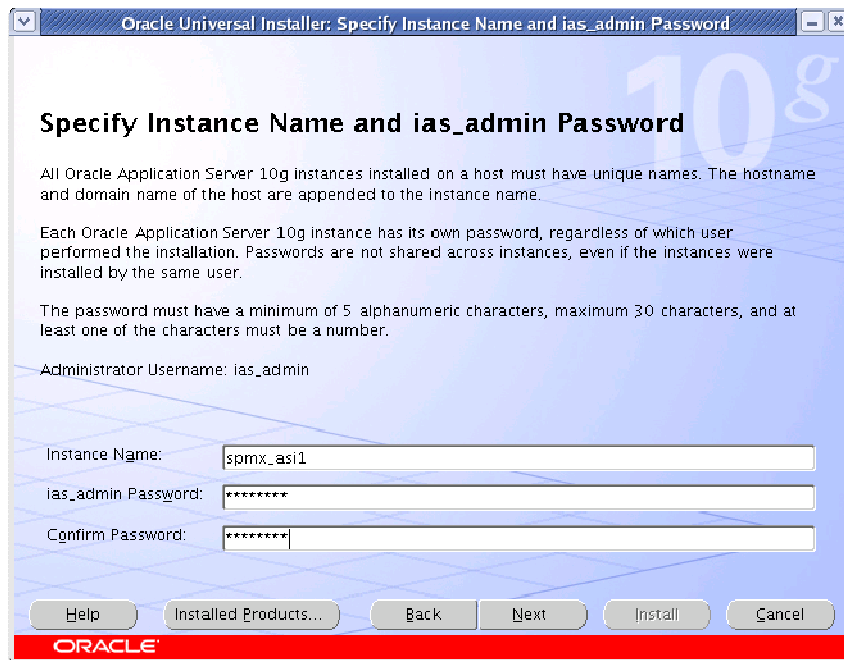


8. Especificamos los puertos que utilizaremos para la instancia "J2EE", el asistente puede crear los puertos de forma automática, pero como se requiere tener un control de ello, se genero anteriormente un archivo de puertos llamado portlist\_j2ee.ini.



9. Especificamos el nombre de la instancia para "J2EE" del Nodo 1 y el password para la administración de esta.

a) Nombre de instancia para servidores de aplicaciones Internos.

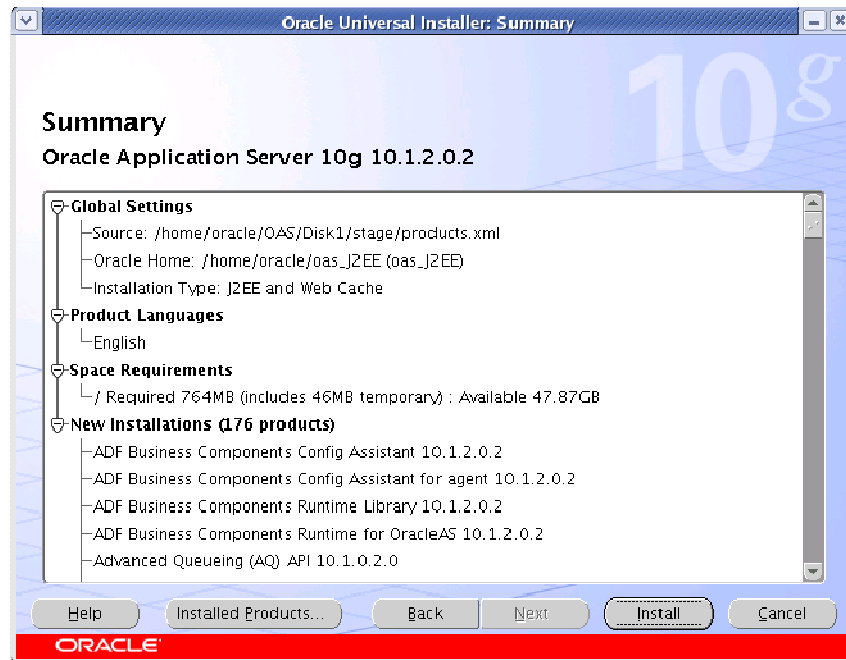




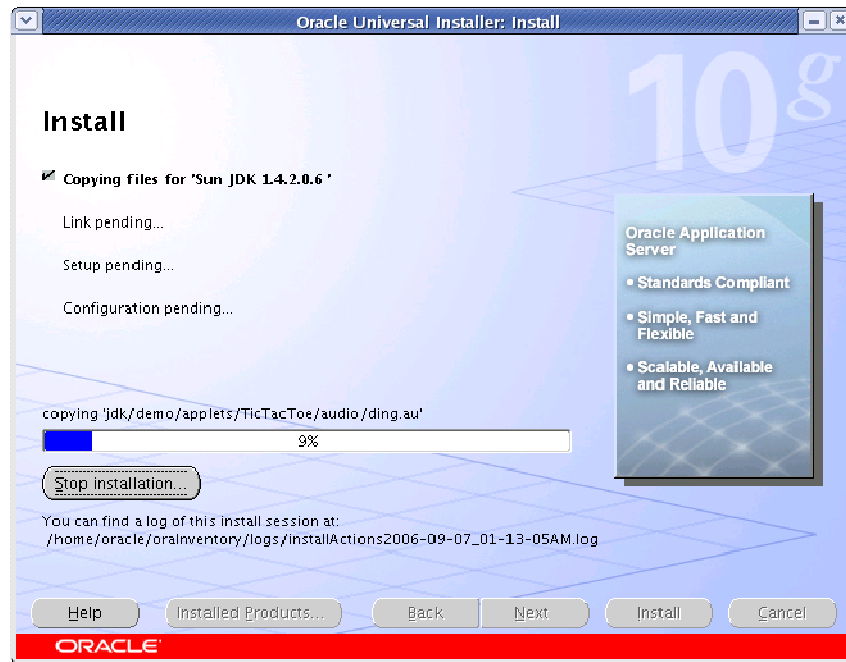
b) Nombre de instancia para servidores de aplicaciones Externos.



10. Se genera un resumen de la instalación y si no hay observaciones, se comienza la instalación.



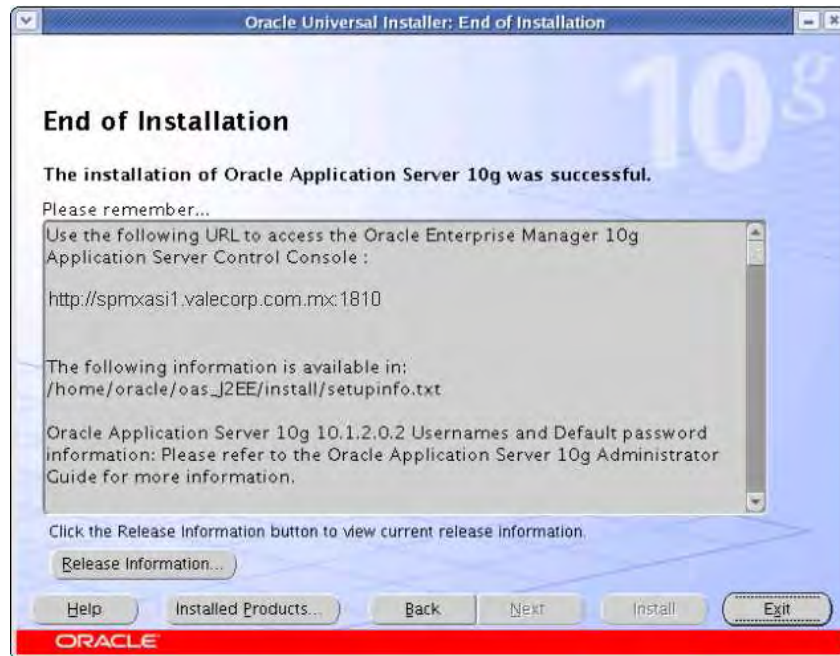
## 11. Progreso de la copia de archivos.



## 12. Después de copiar todos los archivos, se inicia la configuración de los componentes seleccionados.



13. Si no hay problemas en la configuración, el asistente muestra el resumen de la instalación y configuración, dando fin a esta.

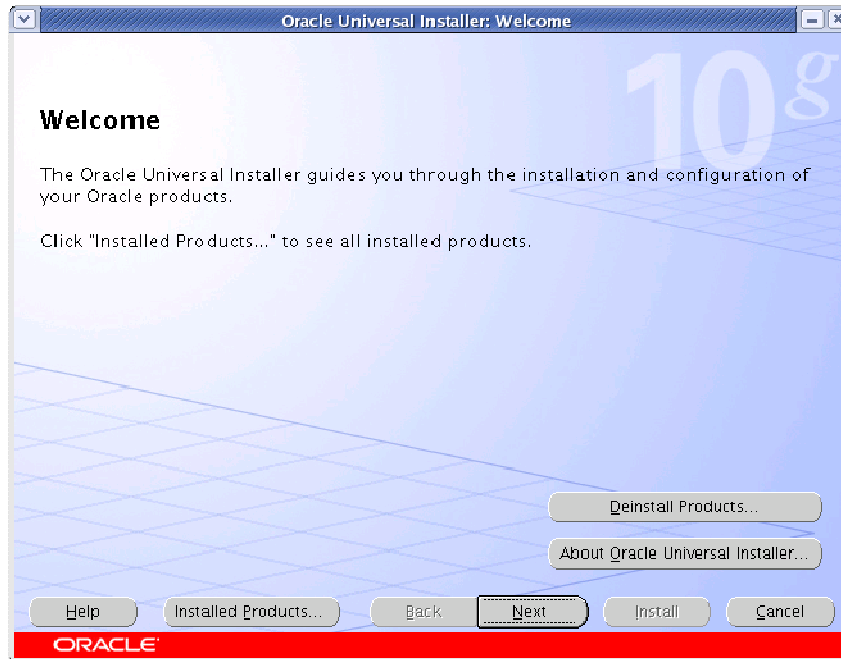


### 3.10 Instalación de Middle Tier J2EE en nodo2

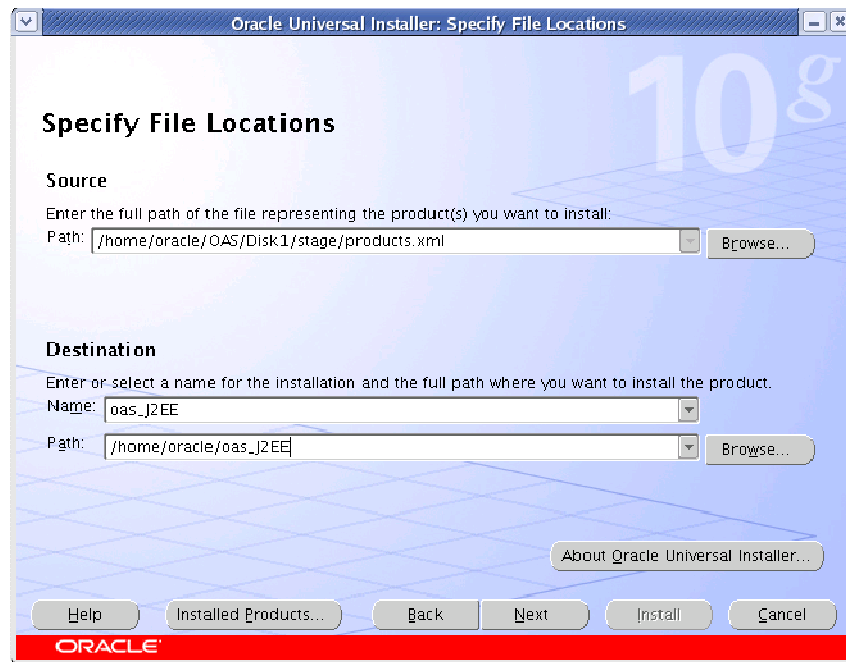
La instalación de J2EE en el nodo 2 es igual a la del nodo 1, claro que con las rutas del nodo2 son diferentes, pero el procedimiento es el mismo.

El procedimiento para instalar las instancias de J2EE es el siguiente.

1. Ejecutamos nuevamente en comando: runInstaller desde la ruta que contiene los discos desempacados de Oracle Application Server 10g R2.



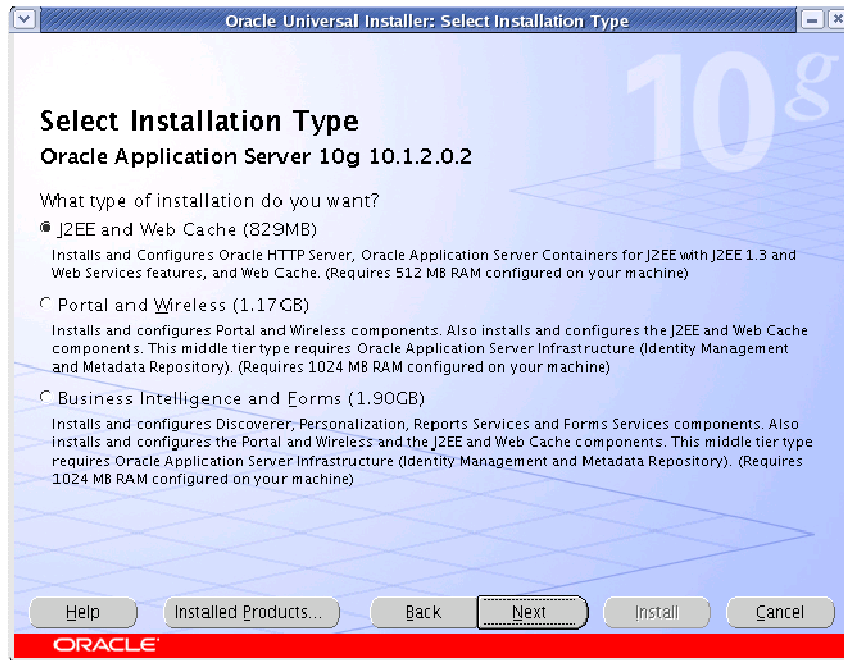
2. Especificamos el origen del software de instalación, el nombre de la instalación y el destino donde se instalara el software.



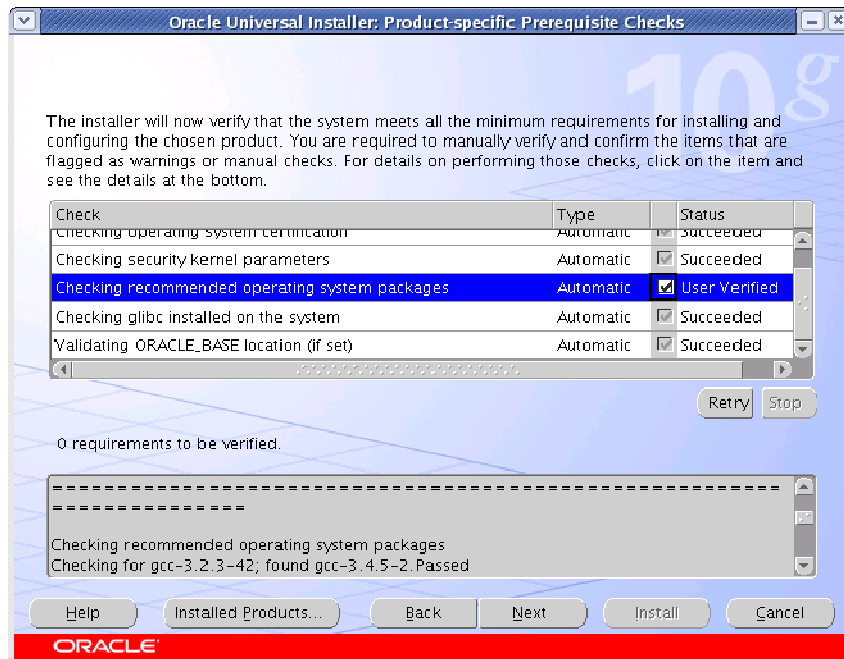
3. Seleccionamos el producto a instalar, como ya contamos con infraestructura, seleccionamos Oracle Application Server 10g 10.1.2.0.2



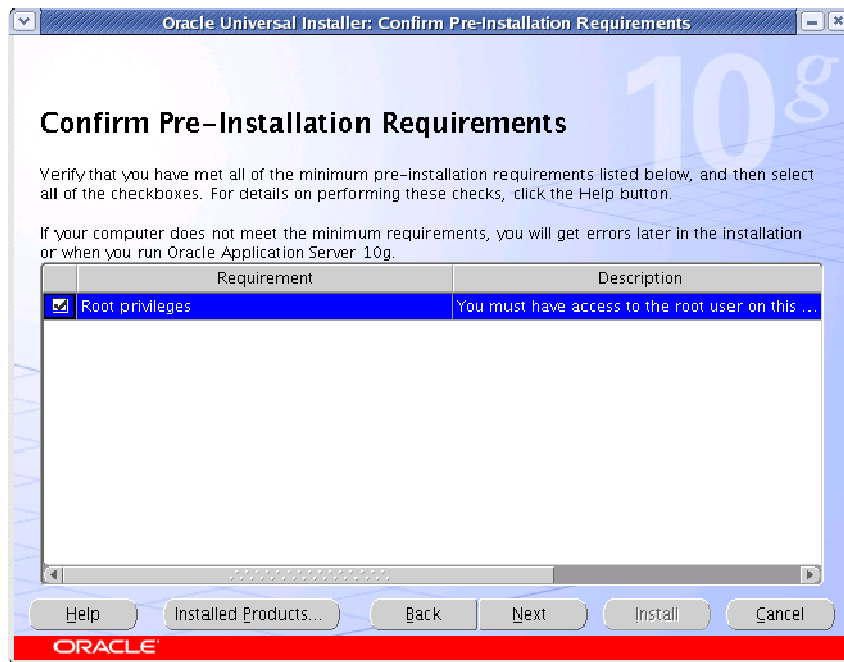
4. Seleccionamos el producto a instalar, estas instancias solo serán contenedoras de J2EE, por lo tanto seleccionamos J2EE and Web Cache



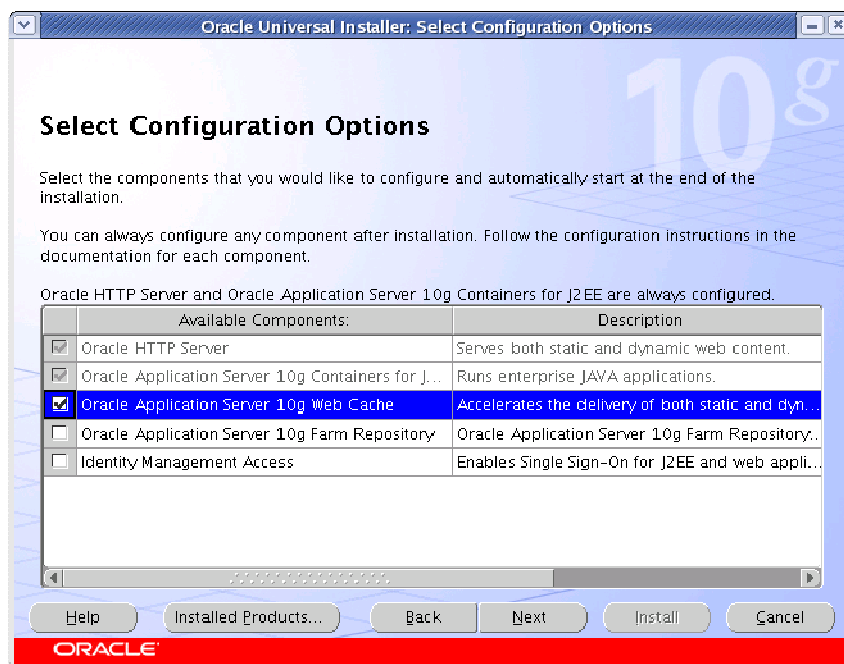
5. El asistente verifica que se cumplan los requerimientos mínimos del sistema, para poder continuar con la instalación



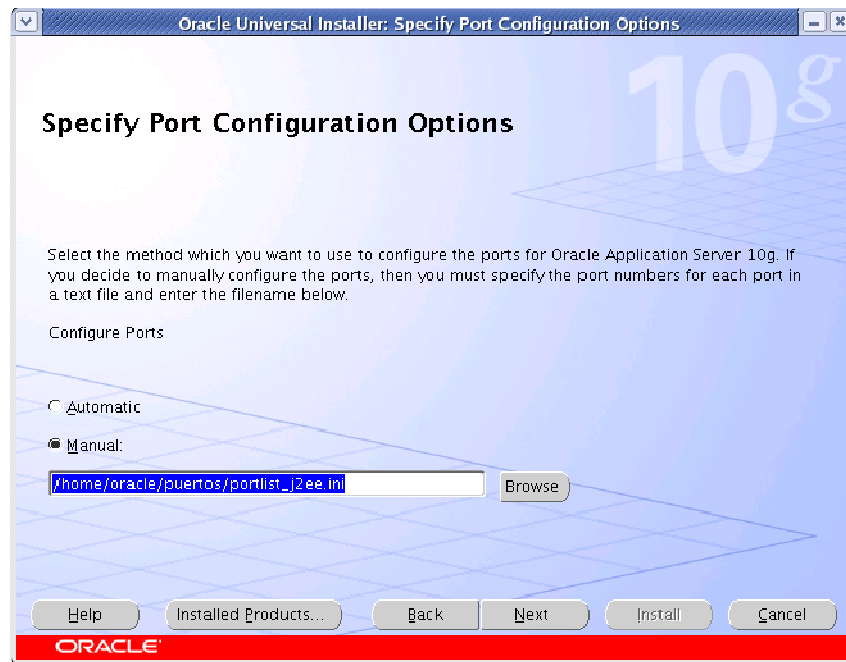
6. Como prerequisite el asistente nos informa que necesitamos tener acceso a la cuenta de root, para ejecutar dos scripts.



7. Seleccionamos los componentes que se desean instalar.



8. Especificamos los puertos que utilizaremos para la instancia "J2EE", el asistente puede crear los puertos de forma automática, pero como se requiere tener un control de ello, se genero anteriormente un archivo de puertos llamado portlist\_j2ee.ini.



9. Especificamos el nombre de la instancia para "J2EE" del Nodo 1 y el password para la administración de esta.

a) Nombre de instancia para servidores de aplicaciones Internos.

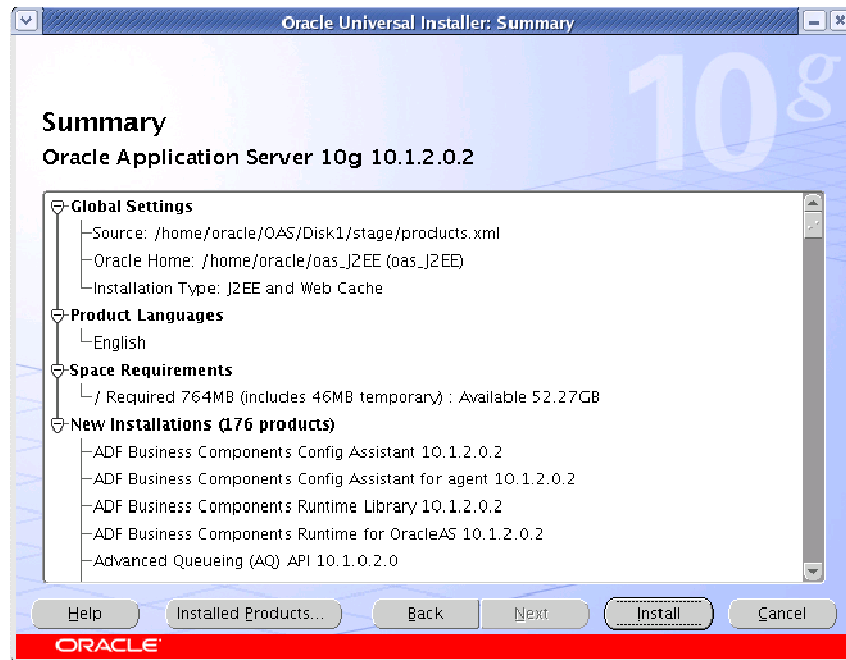




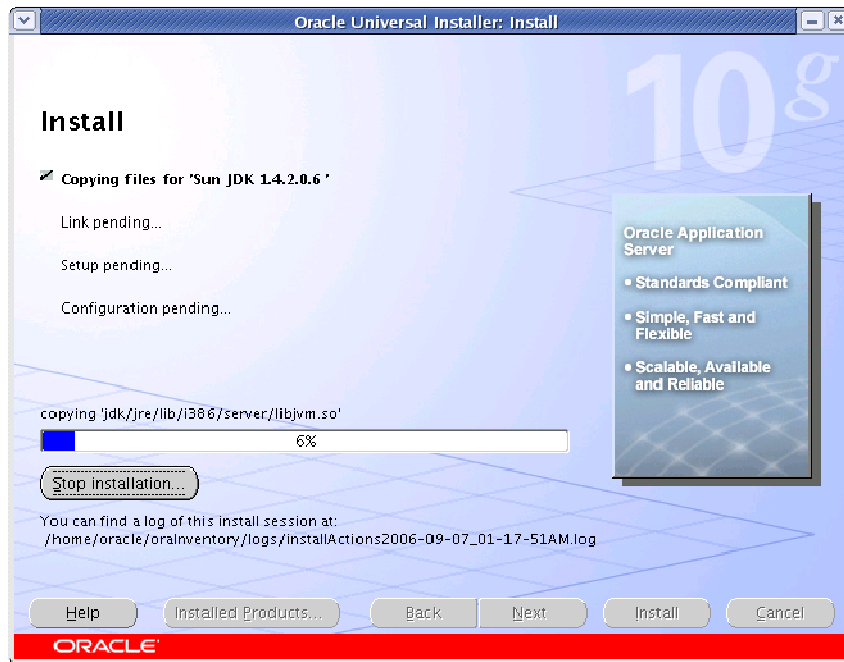
b) Nombre de instancia para servidores de aplicaciones Externos.



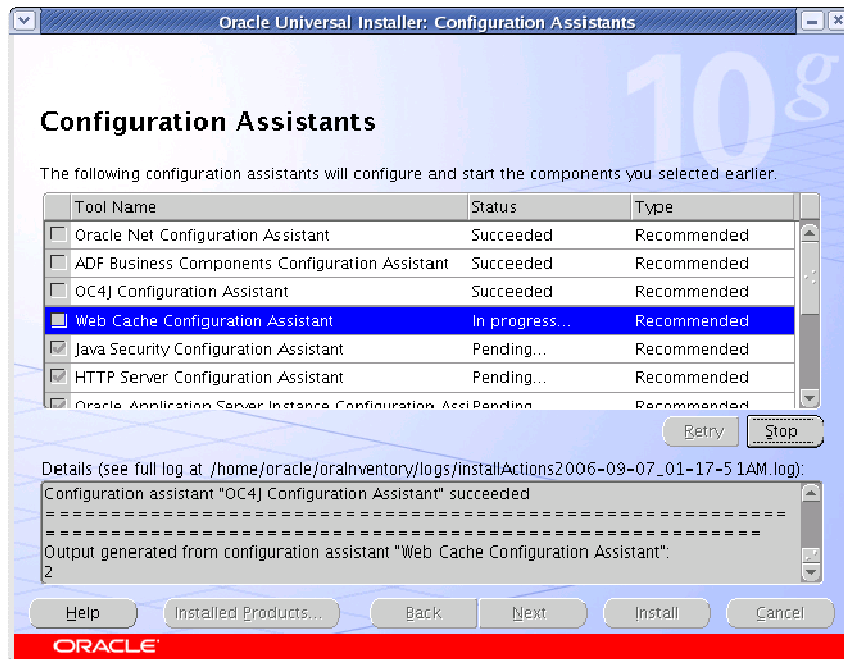
10. Se genera un resumen de la instalación y si no hay observaciones, se comienza la instalación.



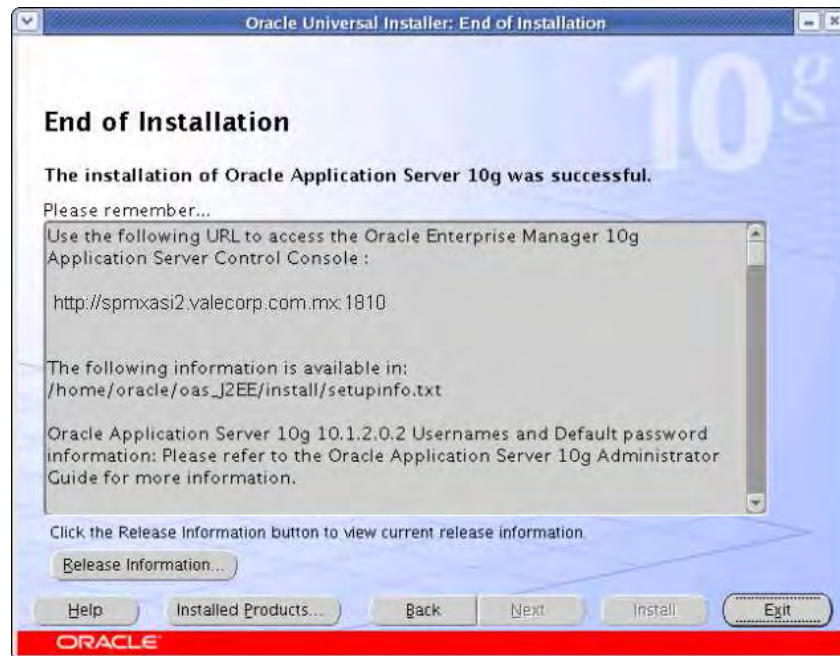
## 11. Progreso de la copia de archivos.



## 12. Después de copiar todos los archivos, se inicia la configuración de los componentes seleccionados.



13. Si no hay problemas en la configuración, el asistente muestra el resumen de la instalación y configuración, dando fin a esta.



### 3.11 Configuración de Cluster J2EE

Para poder crear el cluster de J2EE se requiere que las instancias antes instaladas se unan por medio de la infraestructura, es decir, estar dentro de la misma granja, y después crear un cluster en el cual se agregaran las instancias.

El procedimiento es el siguiente:

1. Abrir administrador de cada instancia
2. Configurar la instancia, indicándole la infraestructura.
3. Abrir administrador general de OAS, y ahora podemos ver que se encuentran dos nuevas instancias.
4. Creamos un nuevo cluster J2EE
5. Agregamos las instancias de J2EE al cluster.
6. Reiniciamos todos los servicios
7. Fin.

## 7 Bibliografía

- **Sistemas de información:** herramientas practicas para la gestión empresarial / Álvaro Gómez Vieites, Carlos Suárez Rey
- **Sistemas de información gerencial:** organización y tecnología de la empresa conectada en red / Kenneth C. Laudon, Jane P. Laudon; traducción Roberto L. Escalona García
- **Sistemas de información basados en computadoras para la administración moderna:** Por. r. g. murdick y j. e. ross. tr. j. meza nieto
- **Computadoras y sistemas de información en los negocios:** George j. brabb
- **Oracle 10g:** administración y análisis de bases de datos / César Pérez López
- **Oracle 10g certification books:** Oracle University
- **Sistemas de bases de datos:** diseño, implementación y administración / Peter Rob, Carlos Coronel; traducción Rodolfo Navarro Salas
- **Physical database design using Oracle:** Donald K. Burleson
- **Expert Oracle database architecture:** 9i and 10g programming techniques and solutions / Thomas Kyte
- **Expert Oracle database 10g administration:** Sam R. Alapati.
- **Oracle application server 10g essentials:** / Rick Greenwald, Robert Stackowiak, and Donald Bales
- **Database modeling for industrial data management :** emerging technologies and applications / Zongmin Ma, editor
- **Oracle application server 10g :** J2EE deployment and administration / Michael Wessler
- **Oracle9i : servidor de aplicaciones, red y programación:** Cesar Pérez López
- **Sistemas operativos:** diseño e implementación / Andrew S. Tanenbaum, Albert S. Woodhull; traducción, Roberto Escalona
- **Operating systems:** Gary Nutt
- **Introduction to operating systems : behind the desktop:** John English
- **The Industrial Information Technology Handbook:** Richard Zurawski
- <http://www.oracle.com/technology/documentation/index.html>
- [http://www.redhat.com/docs/manuals/csgfs/Oracle\\_GFS-en-US/index.html](http://www.redhat.com/docs/manuals/csgfs/Oracle_GFS-en-US/index.html)
- <http://www.gridcomputing.com/>