



ESCUELA POLITÉCNICA DEL EJÉRCITO

SEDE LATACUNGA

CARRERA DE SISTEMAS E INFORMÁTICA

TEMA: ELABORAR EL MANUAL PARA LA CONFIGURACIÓN DE UN SERVIDOR GLASSFISH, UTILIZANDO EL SISTEMA OPERATIVO GNU/LINUX.

PROYECTO DE GRADUACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO EN COMPUTACIÓN

**ELABORADO POR:
TORRES COLLAGUAZO JOSÉ LIZANDRO
VILLAGOMEZ CEVALLOS JINSON OSWALDO**

**DIRECTOR:
ING. RAÚL CAJAS**

**CODIRECTOR:
ING. SANTIAGO JÁCOME**

LATACUNGA, MARZO DEL 2009
ESCUELA POLITÉCNICA DEL EJÉRCITO

SEDE LATACUNGA

CARRERA DE SISTEMAS E INFORMÁTICA

CERTIFICADO

Los suscritos Ing. Raúl Cajas e Ing. Santiago Jácome certifican que el presente trabajo titulado: **“ELABORAR EL MANUAL PARA LA CONFIGURACIÓN DE UN SERVIDOR GLASSFISH, UTILIZANDO EL SISTEMA OPERATIVO GNU/LINUX”** Fue desarrollado íntegramente por los señores: Torres Collaguazo José Lizandro y Villagomez Cevallos Jinson Oswaldo, bajo nuestra supervisión.

Ing. Raúl Cajas
DIRECTOR DE TESIS

Ing. Santiago Jácome
CODIRECTOR DE TESIS

AGRADECIMIENTO

Nuestro principal agradecimiento a Dios por su infinito amor incondicional nos a bendecido por siempre, a nuestros queridos y muy estimados padres por su amor, apoyo, comprensión y darnos lo más hermoso que es la vida, a nuestras esposas por su apoyo incondicional en todos los momentos, a nuestros hijos por ser la fortaleza en nuestras vidas, a los amigos y compañeros por el buen tiempo compartido en el transcurso de la vida estudiantil.

A la Escuela Politécnica del Ejército Sede Latacunga en especial a la Carrera de Sistemas e Informática y a sus Docentes, por sus conocimientos brindados e impartidos para culminar nuestros estudios.

A nuestro Director, Codirector y a la Fuerza Terrestre por la oportunidad que nos brindaron de fomentar nuestros conocimientos por medio de la Educación Superior.

DIOS LOS BENDIGA Y LOS CUIDE A TODOS LOS LECTORES DE ESTE PROYECTO

DEDICATORIA

Este trabajo se lo ofrendo a Dios, mi querida y abnegable esposa, mi inocente hija y mis padres, hermanos y familiares quienes han sido mi soporte a lo largo de mi vida.

Lizandro

DEDICATORIA

Primero quiero agradecer y dedicar este trabajo muy infinitamente a Dios por haberme dado la oportunidad de existir y vivir en este mundo y por haberme dado una familia tan linda thank you God.

Con mucho amor a mis dos más preciados tesoros Jinson y Nayely, los cuales son el motor de mi vida a quienes amo y protegeré siempre. Y con mucho cariño para esa mujer tan hermosa que Dios puso en mi camino, por todo su amor incondicional por haber luchado conmigo en las buenas y en las malas, por haberme alegrado la vida dándome estos dos maravillosos babys, lo cual son la razón de mi existir. Que te puedo decir, muchas gracias por estos 13 años de amor que me has brindado, hemos pasado tantas cosas y hoy nos toca compartir una mas, gracias por estar conmigo en este día tan importante para mí, gracias por todo el apoyo que me has dado para continuar y seguir con mi camino, gracias por estar conmigo y recuerda que eres muy especial para mí.

También lo dedico con mucho amor a ese ser maravilloso que un día me vio nacer gracias Madre por guiarme, por tu enseñanza y amor. Por tu herencia: mi educación, por darme una carrera para mi futuro, por creer y confiar en mí, a mi querido Padre como todos le dicen: Don Cuero, por haber fortalecido mi carácter. Porque siempre me han brindado su apoyo sin esperar nada a cambio de mi, aunque hemos pasado momentos difíciles siempre han estado apoyándome y brindándome todo su amor, por todo esto les agradezco de todo corazón, el que estén conmigo a mi lado.

A mi hermana Cindy y a mi suegra que día a día levantan sus oraciones ante Dios para que nos guíe y nos llene de sus bendiciones.

A mi padre quien un día estuvo a mi lado, pero por razones de la vida tuvo que alejarse, un agradecimiento especial y recuerde que siempre lo llevo en mis pensamientos

DEFINITIVAMENTE: debo dedicar mi tesis a todos aquellos que estuvieron, los que están y continúan a mi lado.

A Dios... por el camino recorrido...

A mis hijos... por ser mi fuerza y templanza...

A mi esposa... amiga fiel y sincera...

A mis padres... por su amor y apoyo...

A la vida... Por todo lo aprendido...

A Dios, mi familia, mi Patria y mi Bandera.

Jinson

PRESENTACIÓN

Dentro de el vivir cotidiano, pocas son los estamentos sociales que se preocupan por informarse de los cambios que se dan día a día, en el amplio campo del software y de la computación, ya que la computación, no es solo un poco de aparatos electrónicos o de bellas interfaces en los monitores de su computadora, va mas allá de aquello. También a sido de lo mejor introducirnos en cuanto a los servicios y utilidades que podemos obtener de los objetos computacionales.

Para este proyecto nos hemos adentrado al fabuloso mundo de los servidores de aplicaciones, de tal manera que dominemos estos servidores con un solo fin en común, orientarnos hacia el software libre, con sistemas operativos de fácil manejo y gratuita distribución. Así como lo son la mayoría de las distribuciones de GNU/Linux y algunas de las versiones de Sun Java, que en esta ocasión a puesto a nuestro servicio el servidor de aplicaciones J2EE o mas conocido como GLASSFISH, el mismo que en nuestro medio todavía no son muy comunes ya que tiene unos dos años desde su creación y esta a disposición de todo el público a cero costos y con un soporte técnico, el mismo que cuenta con una grande e interesante documentación, distribuida desde los mismos productores de este servidor como es SUN MYCROSYSTEM.

El servidor de aplicaciones J2EE, esta hecho para los diferentes sistemas operativos que existen, en Ecuador ya existen instalados servidores de este tipo, en los sistemas operativos Windows y Opensolaris en sus distintas versiones, pero recién se empezara a implementar servidores de aplicaciones bajo el sistema operativo Linux, por lo que es de muchísimo valor la elaboración de este manual, para ser los precursores de este proyecto, para ahorrar muchos recursos a quienes deseen tener este servicio y para satisfacer sus necesidades tecnológicas.

CONTENIDO

I.- CAPÍTULO I.....	15
I.1.- HISTORIA DE LOS SISTEMAS OPERATIVOS.....	16
I.1.1.- INTRODUCCIÓN.....	16
I.2.- DEFINICIÓN.....	18
I.3.- OBJETIVOS DE UN SISTEMA OPERATIVO.....	18
I.3.1.- OBJETIVO PRINCIPAL.....	19
I.3.2.- OBJETIVO SECUNDARIO	19
I.4.- CARACTERÍSTICAS DE LOS SISTEMAS OPERATIVOS.....	19
I.4.1.- CONVENIENCIA.....	19
I.4.2.- EFICIENCIA.....	19
I.4.3.- HABILIDAD PARA EVOLUCIONAR.....	19
I.4.4.- ENCARGADO DE ADMINISTRAR EL HARDWARE.....	20
I.4.5.- ORGANIZAR DATOS PARA ACCESO RÁPIDO Y SEGURO.....	20
I.4.6.- PROCESAMIENTO POR BYTES DE FLUJO A TRAVÉS DEL BUS DE DATOS	20
I.4.7.- TÉCNICAS DE RECUPERACIÓN DE ERRORES.....	20
I.4.8.- GENERACIÓN DE ESTADÍSTICAS.....	20
I.4.9.- EL SOFTWARE DE BASE ESTÁ COMPUESTO POR.....	21
I.5.- TIPOS DE SISTEMAS OPERATIVOS.....	21
I.5.1.- SISTEMAS OPERATIVOS POR SERVICIOS (VISIÓN EXTERNA).....	21
I.5.2.- POR NÚMERO DE USUARIOS.....	21
I.5.2.1.- Sistema Operativo Monousuario.....	22
I.5.2.2.- Sistema Operativo Multiusuario.....	22
I.5.3.- POR EL NÚMERO DE TAREAS.....	22
I.5.3.1.- Sistema Operativo Monotarea.....	22
I.5.3.2.- Sistema Operativo Multitarea.....	23
I.5.4.- POR EL NÚMERO DE PROCESADORES.....	24
I.5.4.1.- Sistema Operativo de Uniproseso.....	24
I.5.4.2.- Sistema Operativo de Multiproseso.....	24
I.5.4.2.1.- Asimétrica.....	24
I.5.4.2.2.- Simétrica.....	24
I.5.5.- SISTEMAS OPERATIVOS POR SU ESTRUCTURA (VISIÓN INTERNA)	25
I.5.5.1.- Estructura Monolítica.....	26
I.5.5.2.- Estructura Jerárquica.....	26
I.5.5.3.- Máquina Virtual.....	27
I.5.5.4.- Cliente-Servidor (Microkernel).....	27
I.5.6.- SISTEMAS OPERATIVOS POR LA FORMA DE OFRECER SUS SERVICIOS.....	28
I.5.6.1.- Sistema Operativo de Red.....	28
I.5.6.2.- Sistemas Operativos Distribuidos.....	29
I.6.- PROCESO DE LOS SISTEMAS OPERATIVOS.....	30
I.6.1.- PLANIFICACIÓN DEL PROCESADOR.....	31
I.6.2.- CARACTERÍSTICAS A CONSIDERAR DE LOS PROCESOS.....	31

I.6.2.1.- Cantidad de Entrada/Salida.....	31
I.6.2.2.- Cantidad de Uso de CPU.....	31
I.6.2.3.- Procesos de Lote o Interactivos	31
I.6.2.4.- Procesos en Tiempo Real.....	32
I.6.2.5.- Longevidad de los Procesos.....	32
I.6.2.6.- Bloque de control de procesos.....	32
I.6.2.6.1.- Los objetivos del bloque de control de procesos.....	32
I.6.2.6.2.- La información contenida en el bloque de control:.....	32
I.6.3.- ESTADO DE LOS PROCESOS.....	33
I.6.3.1.- Estados activos	34
I.6.3.1.1.- Ejecución	34
I.6.3.1.2.- Preparado.....	34
I.6.3.1.3.- Bloqueado	34
I.6.3.2.- Estados inactivos	34
I.6.3.2.1.- Suspendido bloqueado.....	34
I.6.3.2.2.- Suspendido programado	35
I.6.4.- OPERACIONES SOBRE PROCESOS.....	35
I.6.4.1.- Crear el proceso	35
I.6.4.1.1.- Jerárquica.....	35
I.6.4.1.2.- No jerárquica	35
I.6.4.2.- Destruir un proceso	36
I.6.4.3.- Suspender un proceso	36
I.6.4.4.- Reanudar un proceso	36
I.6.4.5.- Cambiar la prioridad de un proceso.....	36
I.6.4.6.- Despertar un proceso.....	36
I.6.5.- PRIORIDADES DE LOS PROCESOS.....	36
I.7.- EL NÚCLEO DEL SISTEMA OPERATIVO.....	37
I.8.- NÚCLEO O KERNEL Y NIVELES DE UN SISTEMA OPERATIVO.....	38
I.8.1.- GESTIÓN DE MEMORIA.....	38
I.8.2.- PROCESADOR.....	38
I.8.3.- ENTRADA/SALIDA.....	38
I.8.4.- INFORMACIÓN O APLICACIÓN O INTÉRPRETE DE LENGUAJES	38
I.8.4.1.- Control de Archivos	38
I.8.5.- EL NÚCLEO Y LOS PROCESOS.....	39
I.8.5.1.- El núcleo	39
I.8.5.2.- Los procesos.....	39
I.9.- DISPOSITIVOS DE ENTRADA Y SALIDA.....	39
I.9.1.- TIPOS DE DISPOSITIVOS DE ENTRADA Y SALIDA.....	40
I.9.1.1.- Orientados a Bloques.....	40
I.9.1.2.- Orientados a Caracteres.....	40
I.9.2.- PRINCIPIOS DEL SOFTWARE DE ENTRADA Y SALIDA.....	40
I.9.2.1.- Manejadores de Interrupciones.....	41
I.9.2.2.- Manejadores de Dispositivos.....	41
I.9.2.3.- Software que sea independiente de los dispositivos.....	41
I.9.2.4.- Software para Usuarios.....	41
I.9.3.- MANEJO DE LOS DISPOSITIVOS DE E/S.....	42

I.9.3.1.- Buffering (uso de memoria intermedia).....	42
I.9.3.2.- Spooling.....	43
I.10.- ADMINISTRACIÓN DE ARCHIVOS.....	44
I.11.- SEGURIDAD DE UN SISTEMA OPERATIVO.....	45
I.11.1.- SEGURIDAD FÍSICA.....	46
I.11.2.- SEGURIDAD DE ADMINISTRACIÓN.....	46
I.12.- DEFINICIÓN DE RED HAT.....	46
I.12.1.- HISTORIA DE RED HAT.....	46
I.12.2.- DISTRIBUCIONES DE RED HAT.....	47
I.12.2.1.- Red Hat Enterprise Linux AS.....	47
I.12.2.2.- Red Hat Enterprise Linux ES.....	48
I.12.2.3.- Red Hat Enterprise Linux WS.....	48
I.12.2.4.- Red Hat Desktop.....	48
I.12.3.- CARACTERÍSTICAS ESPECIALES DE RED HAT.....	49
I.13.- GLASSFISH DEFINICIÓN.....	49
I.13.1.- ¿QUÉ ES UN SERVIDOR?	50
I.13.2.- DIFERENCIAS ENTRE SERVIDORES HTTP, SERVIDORES WEB Y SERVIDORES DE APLICACIONES.....	51
I.13.3.- GLASSFISH ¿POR QUÉ? ¿PARA QUÉ? ¿PARA QUIÉN?	51
I.13.3.1.- ¿Por qué?	51
I.13.3.2.- ¿Para qué?.....	52
I.13.3.3.- ¿Para quién?	52
I.13.4.- ADMINISTRANDO GLASSFISH (USUARIOS, SEGURIDAD Y DESPLIEGUE DE APLICACIONES).....	52
I.13.5.- SEGURIDAD	53
I.13.6.- CONFIGURACIÓN XML	54
I.14.- DESPLIEGUE DE APLICACIONES	55
I.15.- CONFIGURANDO GLASSFISH (DATASOURCES, JMS, JAAS, WEBSERVICES, ETC.)	56
I.16.- DATASOURCES	56
I.16.1.- SEGURIDAD.....	57
I.16.2.- PRODUCTIVIDAD.....	57
I.16.3.- CENTRALIZACIÓN.....	57
I.17.- CONFIGURAR UN DATASOURCE O FUENTE DE DATOS.....	57
I.17.1.- JMS	59
I.18.- CARACTERÍSTICAS DE GLASSFISH.....	59
I.19.- CREACIÓN DE COLAS Y TÓPICOS JMS EN GLASSFISH	60
I.19.1.- FACTORÍA DE COLAS	61
I.19.2.- FACTORÍA DE TÓPICOS	61
I.19.2.1.- Cola	61
I.19.2.2.- Tópico	61
I.20.- JAAS	62
I.21.- WEBSERVICES	64
I.21.1.- TUNNING DEL SERVIDOR	64
I.21.2.- CLUSTERING	67
I.21.3.- CONFIGURANDO EL CLUSTER EN GLASSFISH	67

I.21.3.1.- BALANCEO DE CARGA	69
I.22.- MAIL.....	69
II.- CAPÍTULO II.....	70
II.1.- REQUISITOS DE SISTEMA Y PLATAFORMAS COMPATIBLES.....	70
II.1.1.- REQUISITOS DE SOFTWARE.....	70
II.1.2.- PLATAFORMAS COMPATIBLES.....	71
II.1.3.- REQUISITOS DEL HOST DEL SERVIDOR	71
II.1.4.- REQUISITOS DEL HOST DE ADMINISTRACIÓN	71
II.1.5.- REQUISITOS DEL HOST DEL CLIENTE	71
II.1.6.- OTROS REQUISITOS.....	72
II.1.7.- ESPACIO LIBRE.....	72
II.1.8.- USO DEL PROGRAMA DE DESINSTALACIÓN	72
II.1.9.- PUERTOS LIBRES	72
II.1.10.- INICIO DE SERVIDORES PREVIAMENTE INSTALADOS (UNIX)....	73
II.1.11.- CIERRE DEL SERVIDOR DE SEGURIDAD.....	73
II.2.- CONOCIMIENTOS BÁSICOS DE LAS HERRAMIENTAS DE GLASSFISH. 73	
II.2.1.- ACCESO A LA PLATAFORMA GLASSFISH.....	73
II.2.2.- CONFIGURACIÓN DE LA BASE DE DATOS.....	74
II.2.3.- CONFIGURACIÓN DE ECLIPSE.....	74
II.2.4.- CONFIGURACIÓN DE LA APLICACIÓN.....	74
II.2.5.- INDICAR LA LISTA DE LAS LIBRERÍAS DEL DRIVER CONNECTOR/J DE MYSQL.....	76
II.2.6.- CREAR UN NUEVO RECURSO JDBC (DATA SOURCE).....	80
II.2.7.- CREAR UN NUEVO POOL DE CONEXIONES JDBC.....	82
II.3.- VENTAJAS Y DESVENTAJAS DE GLASSFISH.....	90
III.- CAPÍTULO III.....	92
III.1.- INSTALACIÓN DE RED HAT LINUX.....	92
III.1.1.- SELECCIÓN DE UN MÉTODO DE INSTALACIÓN.....	93
III.1.1.1.- Cd-rom	93
III.1.1.2.- Disco duro	93
III.1.1.3.- Imagen NFS	93
III.1.1.4.- Ftp	93
III.1.1.5.- Http	94
III.1.2.- INSTALACIÓN DESDE UN CD-ROM.....	94
III.1.2.1.- ¿Qué hacer si no se detecta el CD-ROM IDE?.....	94
III.1.3.- BIENVENIDO A RED HAT LINUX.....	95
III.1.4.- SELECCIÓN DEL IDIOMA.....	95
III.1.5.- CONFIGURACIÓN DEL TECLADO.....	96
III.1.6.- CONFIGURACIÓN DEL RATÓN.....	96
III.1.7.- SELECCIONAR ACTUALIZAR O INSTALAR.....	98
III.1.8.- OPCIONES DE INSTALACIÓN.....	99
III.1.9.- CONFIGURACIÓN DEL PARTICIONAMIENTO DEL DISCO.....	100
III.1.10.- PARTICIONAMIENTO AUTOMÁTICO.....	101
III.1.10.1.- Eliminar todas las particiones Linux del sistema	101
III.1.10.2.- Eliminar todas las particiones del sistema	101
III.1.10.3.- Mantener todas las particiones y usar el espacio libre existente	102

III.1.11.- PARTICIONAMIENTO DEL SISTEMA.....	103
III.1.11.1.- Punto de montaje:	104
III.1.11.2.- Tipo de sistema de archivos:.....	104
III.1.12.- CONFIGURACIÓN DEL GESTOR DE ARRANQUE.....	106
III.1.13.- GRUB (GRAND UNIFIED BOOTLOADER).....	106
III.1.14.- LILO (LINUX LOADER).....	106
III.1.15.- CONFIGURACIÓN AVANZADA DEL GESTOR DE ARRANQUE....	108
III.1.15.1.- El master boot record (mbr)	108
III.1.16.- CONFIGURACIÓN DE RED.....	109
III.1.17.- CONFIGURACIÓN DEL CORTAFUEGOS.....	110
III.1.17.1.- ALTO	111
III.1.17.2.- Respuestas de DNS.....	111
III.1.17.3.- Transferencias de archivos IRC DCC.....	111
III.1.18.- SELECCIÓN DEL SOPORTE DEL IDIOMA.....	111
III.1.19.- CONFIGURACIÓN DEL USO HORARIO.....	112
III.1.20.- CONFIGURACIÓN DE LA AUTENTICACIÓN.....	114
III.1.20.1.- HABILITAR CONTRASEÑAS MD5.....	115
III.1.20.2.- HABILITAR CONTRASEÑAS SHADOW.....	115
III.1.20.3.- HABILITAR NIS.....	115
III.1.20.3.1.- Dominio NIS.....	115
III.1.20.3.2.- Usar broadcast para encontrar el servidor NIS	115
III.1.20.3.3.- Servidor NIS	115
III.1.21.- SELECCIÓN INDIVIDUAL DE PAQUETES.....	117
III.1.22.- PREPARACIÓN PARA LA INSTALACIÓN.....	118
III.1.23.- Instalación de paquetes.....	118
III.1.24.- CREACIÓN DE UN DISQUETE DE ARRANQUE.....	119
III.1.25.- CONFIGURACIÓN DE LA TARJETA DE VÍDEO.....	120
III.1.26.- CONFIGURACIÓN DEL MONITOR.....	121
III.1.27.- FIN DE LA INSTALACIÓN.....	123
III.2.- INSTALACIÓN DE UBUNTU V8.10.....	125
III.2.1.- CARACTERÍSTICAS.....	125
III.2.2.- PASOS PARA LA INSTALACIÓN DE UBUNTU.....	125
III.2.2.1.- Probar Ubuntu sin alterar su equipo	127
III.2.2.2.- Instalar Ubuntu.....	127
III.2.3.- SELECCIÓN DE RED.....	128
III.2.4.- AUTENTICACIÓN DE RED INALÁMBRICA.....	128
III.2.5.- CREACIÓN PREDETERMINADA PARA ANILLOS DE CLAVE.....	129
III.2.6.- SELECCIÓN DEL IDIOMA.....	130
III.2.7.- SELECCIÓN DE ZONA HORARIA.....	131
III.2.8.- SELECCIÓN DE DISTRIBUCIÓN DEL TECLADO	132
III.2.9.- MANEJO DE PARTICIONES	133
III.2.10.- PREPARACIÓN DE PARTICIONES.....	133
III.2.11.- CREACIÓN DE NUEVA PARTICIÓN.....	134
III.2.11.1.- Selección de la partición.....	135
III.2.12.- PREPARAR PARTICIONES.....	137
III.2.13.- COBERTURA DE DATOS DE USUARIO.....	138

<u>III.2.14.- INSTALANDO EL SISTEMA EN SI.....</u>	<u>139</u>
<u>III.3.- INSTALAR GLASSFISH.....</u>	<u>141</u>
<u>III.3.1.- PASOS PARA LA INSTALACIÓN.....</u>	<u>141</u>
<u>III.4.- CARGAR UNA APLICACIÓN WEB.....</u>	<u>143</u>
<u>IV.- CAPÍTULO IV.....</u>	<u>150</u>
<u>IV.1.- CONCLUSIONES:.....</u>	<u>150</u>
<u>IV.2.- RECOMENDACIONES:.....</u>	<u>150</u>

TABLA DE ILUSTRACIONES

Figura 2-1. Pagina de Inicio Glassfish.....	74
Figura 2-2. Configuración de la Aplicación (Tareas comunes).....	75
Figura 2-3. Configuración del Modulo EJB.....	76
Figura 2-4. Configuración de la Aplicación Web.....	76
Figura 2-5. Configuración del servidor de Aplicaciones.....	77
Figura 2-5. Configuración de la Maquina Virtual de Java.....	78
Figura 2-7. Comprobación de la configuración	79
Figura 2-8. Configuración de los Recursos de JDBC.....	80
Figura 2-9. Habilitación de los Recursos del JDBC.....	81
Figura 2-10. Crear el JDBC.....	82
Figura 2-11. Crear un Nuevo Pool de Conexiones.....	83
Figura 2-12. Configuración del pool de conexiones (paso 1).....	84
Figura 2-12. Configuración del pool de conexiones (Paso 2).....	84
Figura 2-14. Configuración de del pool de conexiones.....	85
Figura 2-15. Validación del Pool de Conexiones.....	86
Figura 2-16. Transacción del Pool de Conexiones.....	86
Figura 2-17. Añadir propiedades al Pool de Conexiones.....	87
Figura 2-18. Tabla del Pool Conexiones.....	87
Figura 2-19. Editar el Pool de Conexiones.....	88
Figura 2-20. Verificación de la edición del Pool de Conexiones	89
Tabla 2-4. Ventajas y desventajas de Glassfish.....	90
Figura 3-1. Selección del idioma.....	96
Figura 3-2. Configuración del teclado.....	96
Figura 3-3. Configuración del ratón.....	98
Figura 3-4. Comprobación de actualización.....	99
Figura 3-5. Elegir el tipo de instalación.....	100
Figura 3-6. Configuración del particionamiento del disco.....	101
Figura 3-7. Particionamiento automático.....	102
Figura 3-8. Particionamiento con Disk Druid.....	103
Figura 3-9. Creación de una nueva partición.....	104
Figura 3-10. Configuración del gestor de arranque.....	107
Figura 3-11. Configuración del gestor de arranque.....	108
Figura 3-12. Configuración de red.....	109
Figura 3-13. Modificación del dispositivo de red.....	110
Figura 3-14. Configuración del firewall.....	111
Figura 3-15. Selección del soporte adicional del idioma.....	112
Figura 3-16. Configuración del uso horario.....	113
Figura 3-17. Configuración de la Contraseña de root.....	114
Figura 3-18. Configuración de la autenticación.....	115
Figura 3-19. Selección de grupos de paquetes.....	116
Figura 3-20. Selección de los detalles para editores.....	117
Figura 3-21. Selección individual de los paquetes.....	118
Figura 3-22. Instalación de los paquetes.....	119

Figura 3-23. Creación del disquete de arranque.....	120
Figura 3-24. Configuración de la interfaz grafica (X).....	121
Figura 3-25. Selección del monitor.....	122
Figura 3-26. Personalización de X.....	123
Figura 3.2-1. Selección de tipo de instalación.....	126
Figura 3.2-2. Interfaz de Ubuntu.....	127
Figura 3.2-3. Selección de red.....	128
Figura 3.2-4. Autenticación de red inalámbrica.....	129
Figura 3.2-5. Creación de Anillos de claves.....	130
Figura 3.2-6. Selección de Idioma.....	130
Figura 3.2-7. Selección de la zona horaria.....	131
Figura 3.2-8. Distribución del teclado.....	132
Figura 3.2-9. Preparación del espacio del disco.....	133
Figura 3.2-10. Preparación de particiones.....	134
Figura 3.2-11. Creación de nueva Partición.....	134
Figura 3.2-12. Asignación de espacio de disco.....	136
Figura 3.2-13. Preparar particiones.....	137
Figura 3.2-14. Cobertura de los datos del Usuario.....	138
Figura 3.2-15. Instalando el sistema.....	139
Figura 3.2-16. Instalando el sistema.....	139
Figura 3.2-17. Instalando el sistema.....	140
Figura 3.2-18. Fin de la instalación.....	140
Figura 3.2-19. Escritorio de Ubuntu.....	140
Figura 3.4-1. Ventana de Acceso a Glassfish.....	144
Figura 3.4-2. Manejo de Aplicaciones pasó 1.....	145
Figura 3.4-3. Manejo de Aplicaciones pasó 2.....	145
Figura 3.4-4. Manejo de Aplicaciones pasó 3.....	146
Figura 3.4-5. Manejo de Aplicaciones pasó 4.....	147
Figura 3.4-6. Manejo de Aplicaciones pasó 5.....	148
Figura 3.4-7. Manejo de Aplicaciones pasó 6.....	148
Figura 3.4-8. Manejo de Aplicaciones pasó 7.....	149
Figura 3.4-8. Aplicación Ejecutada.....	149

CAPÍTULO I

En este capítulo, revisaremos fundamentos teóricos de los sistemas operativos como historia, funcionalidad y características que nos ayudaran a fortalecer los conocimientos, además trataremos temas de interés y que son el objetivo principal de este proyecto como lo es GLASSFISH, REDHAT y WEBSERVICES.

I.1.- HISTORIA DE LOS SISTEMAS OPERATIVOS

I.1.1.- INTRODUCCIÓN

Pocos pueden discutir sobre los diferentes sistemas operativos y sus diferentes beneficios y como han cambiado nuestras vidas, mentes, sociedades y lo más importante, han facilitado nuestras actividades diarias. Cada día es mas común ver y operar maquinas mas modernas con más funcionalidad y con más beneficios para nosotros pero pocos nos preguntamos a que se debe los movimientos o las actividades que hacen dichas máquinas. Es así que, durante las últimas décadas hemos asistido al crecimiento vertiginoso del software y más de los sistemas operativos que cada día son más accesibles, más sofisticados, y más complejos.

Afortunadamente, dicha complejidad se convierte en beneficio para cada usuario de uno o varios sistemas operativos, los hacen mas amigable, garantizan la calidad, fiabilidad pero nos llevan a una dependencia a nivel social, económico y funcional.

Para tratar de comprender los requisitos de un Sistema Operativo y el significado de las principales características de un Sistema Operativo contemporáneo, es útil considerar como han ido evolucionando éstos con el tiempo.

A finales de los 40's el uso de computadoras estaba restringido a aquellas empresas o instituciones que podían pagar su alto precio, y no existían los sistemas operativos. En su lugar, el programador debía tener un conocimiento y contacto profundo con el hardware, y en el infortunado caso de que su programa fallara, debía examinar los valores de los registros y paneles de luces indicadoras

del estado de la computadora para determinar la causa del fallo y poder corregir su programa, además de enfrentarse nuevamente a los procedimientos de apartar tiempo del sistema y poner a punto los compiladores, ligadores, etc.; para volver a correr su programa, es decir, enfrentaba el problema del procesamiento serial (serial processing).

La importancia de los sistemas operativos nace históricamente desde los 50's, cuando se hizo evidente que el operar una computadora por medio de tableros enchufables en la primera generación y luego por medio del trabajo en lote en la segunda generación se podía mejorar notoriamente, pues el operador realizaba siempre una secuencia de pasos repetitivos, lo cual es una de las características contempladas en la definición de lo que es un programa. Es decir, se comenzó a ver que las tareas mismas del operador podían plasmarse en un programa, el cual a través del tiempo y por su enorme complejidad se le llamó "Sistema Operativo". Así, tenemos entre los primeros sistemas operativos al Fortran Monitor System (FMS) e IBSYS.

Posteriormente, en la tercera generación de computadoras nace uno de los primeros sistemas operativos con la filosofía de administrar una familia de computadoras: el OS/360 de IBM. Fue este un proyecto tan novedoso y ambicioso que enfrentó por primera vez una serie de problemas conflictivos debido a que anteriormente las computadoras eran creadas para dos propósitos en general: el comercial y el científico. Así, al tratar de crear un solo sistema operativo para computadoras que podían dedicarse a un propósito, al otro o ambos, puso en evidencia la problemática del trabajo en equipos de análisis, diseño e implantación de sistemas grandes.

Surge también en la tercera generación de computadoras el concepto de la multiprogramación, porque debido al alto costo de las computadoras era necesario idear un esquema de trabajo que mantuviese a la unidad central de procesamiento más tiempo ocupada, así como el encolado (spooling) de trabajos para su lectura hacia los lugares libres de memoria o la escritura de resultados.

Sin embargo, se puede afirmar que los sistemas durante la tercera generación siguieron siendo básicamente sistemas de lote.

En la cuarta generación la electrónica avanza hacia la integración a gran escala, pudiendo crear circuitos con miles de transistores en un centímetro cuadrado de silicón y ya es posible hablar de las computadoras personales y las estaciones de trabajo. Surgen los conceptos de interfaces amigables intentando así atraer al público en general al uso de las computadoras como herramientas cotidianas. Se hacen populares el MS-DOS y UNIX en estas máquinas. También es común encontrar clones de computadoras personales y una multitud de empresas pequeñas ensamblándolas por todo el mundo.

Para mediados de los 80's, comienza el auge de las redes de computadoras y la necesidad de sistemas operativos en red y sistemas operativos distribuidos. La red mundial Internet se va haciendo accesible a toda clase de instituciones y se comienzan a dar muchas soluciones (y problemas) al querer hacer convivir recursos residentes en computadoras con sistemas operativos diferentes. Para los 90's el paradigma de la programación orientada a objetos cobra auge, así como el manejo de objetos desde los sistemas operativos. Las aplicaciones intentan crearse para ser ejecutadas en una plataforma específica y poder ver sus resultados en la pantalla o monitor de otra diferente (por ejemplo, ejecutar una simulación en una máquina con UNIX y ver los resultados en otra con DOS). Los niveles de interacción se van haciendo cada vez más profundos.

I.2.- DEFINICIÓN

Un Sistema Operativo es un programa que actúa como intermediario entre el usuario y el hardware de un computador y su propósito es proporcionar un entorno en el cual el usuario pueda ejecutar programas.

I.3.- OBJETIVOS DE UN SISTEMA OPERATIVO

El objetivo del sistema operativo es gestionar y administrar eficientemente los recursos hardware de la computadora, permitiendo ejecutar concurrentemente varios programas sin que haya conflictos en el acceso de cada uno de ellos a cada uno de los recursos que necesita y sin que ningún programa monopolice un recurso determinado, se subdivide en dos:

- Objetivo Principal
- Objetivo Secundario

I.3.1.- OBJETIVO PRINCIPAL

Es, lograr que el Sistema de computación se use de manera cómoda.

I.3.2.- OBJETIVO SECUNDARIO

Es que el hardware del computador se emplee de manera eficiente.

Un Sistema Operativo es una parte importante de cualquier sistema de computación. Un sistema de computación puede dividirse en cuatro componentes: el hardware, el Sistema Operativo, los programas de aplicación y los usuarios. El hardware (Unidad Central de Procesamiento (UCP), memoria y dispositivos de entrada/salida (E/S)) proporciona los recursos de computación básicos. Los programas de aplicación (compiladores, sistemas de bases de datos, video juegos y programas para negocios) definen la forma en que estos recursos se emplean para resolver los problemas de computación de los usuarios.

I.4.- CARACTERÍSTICAS DE LOS SISTEMAS OPERATIVOS

En general, se puede decir que un Sistema Operativo tiene las siguientes características:

I.4.1.- CONVENIENCIA

Un Sistema Operativo hace más conveniente el uso de una computadora.

I.4.2.- EFICIENCIA

Un Sistema Operativo permite que los recursos de la computadora se usen de la manera más eficiente posible.

I.4.3.- HABILIDAD PARA EVOLUCIONAR

Un Sistema Operativo deberá construirse de manera que permita el desarrollo, prueba o introducción efectiva de nuevas funciones del sistema sin interferir con el servicio.

I.4.4.- ENCARGADO DE ADMINISTRAR EL HARDWARE

El Sistema Operativo se encarga de manejar de una mejor manera los recursos de la computadora en cuanto a hardware se refiere, esto es, asignar a cada proceso una parte del procesador para poder compartir los recursos.

Relacionar dispositivos (gestionar a través del kernel). El Sistema Operativo se debe encargar de comunicar a los dispositivos periféricos, cuando el usuario así lo requiera.

I.4.5.- ORGANIZAR DATOS PARA ACCESO RÁPIDO Y SEGURO

Manejar las comunicaciones en red. El Sistema Operativo permite al usuario manejar con alta facilidad todo lo referente a la instalación y uso de las redes de computadoras.

I.4.6.- PROCESAMIENTO POR BYTES DE FLUJO A TRAVÉS DEL BUS DE DATOS

Facilitar las entradas y salidas. Un Sistema Operativo debe hacerle fácil al usuario el acceso y manejo de los dispositivos de Entrada/Salida de la computadora.

I.4.7.- TÉCNICAS DE RECUPERACIÓN DE ERRORES

Evita que otros usuarios interfieran. El Sistema Operativo evita que los usuarios se bloqueen entre ellos, informándoles si esa aplicación está siendo ocupada por otro usuario.

I.4.8.- GENERACIÓN DE ESTADÍSTICAS

Permite que se puedan compartir el hardware y los datos entre los usuarios.

El software de aplicación son programas que se utilizan para diseñar, tal como el procesador de palabras, lenguajes de programación, hojas de cálculo, etc.

El software de base sirve para interactuar el usuario con la máquina, son un conjunto de programas que facilitan el ambiente plataforma, y permite el diseño del mismo.

I.4.9.- EL SOFTWARE DE BASE ESTÁ COMPUESTO POR

- Cargadores
- Compiladores
- Ensambladores
- Macros.

I.5.- TIPOS DE SISTEMAS OPERATIVOS

Actualmente los sistemas operativos se clasifican en tres clasificaciones:

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

I.5.1.- 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:

I.5.2.- POR NÚMERO DE USUARIOS

Un sistema operativo por el número de Usuarios se clasifica en dos tipos:

- Sistema operativo Monousuario
- Sistema operativo Multiusuario

I.5.2.1.- Sistema Operativo 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 nada más puede atender a un solo usuario, gracias a las limitaciones creadas por el hardware, los programas o el tipo de aplicación que se este ejecutando.

I.5.2.2.- Sistema Operativo 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 mismos recursos. Este tipo de sistemas se emplean especialmente en redes. En otras palabras consiste en el fraccionamiento del tiempo (timesharing).

I.5.3.- POR EL NÚMERO DE TAREAS

Un sistema operativo por el número de Tareas se clasifica en dos tipos:

- Sistema operativo Monotarea
- Sistema operativo Multitarea

I.5.3.1.- Sistema Operativo 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 que solo puede ejecutar las tareas de una en una.

I.5.3.2.- Sistema Operativo Multitarea

Un sistema operativo multitarea es aquél 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 multitareas. 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 esta 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 procesado durante los tiempos muertos de la tarea que se encuentra en primer plano (por ejemplo, cuando esta aplicación esta 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.

I.5.4.- POR EL NÚMERO DE PROCESADORES

Un sistema operativo por el número de Procesadores se clasifica en dos tipos:

- Sistema operativo de Uniproceto
- Sistema operativo Multiproceto

I.5.4.1.- Sistema Operativo de Uniproceto

Un sistema operativo uniproceto 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.

I.5.4.2.- Sistema Operativo de Multiproceto

Un sistema operativo multiproceto 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.

I.5.4.2.1.- 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.

I.5.4.2.2.- Simétrica

Cuando se trabaja de manera simétrica, los procesos o partes de ellos (threads) son enviados indistintamente a cual quiera 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.

I.5.5.- 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.

I.5.5.1.- 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.

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.

I.5.5.2.- 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, del sistema operativo, donde una parte del sistema contenía subpartes y esto 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 una clara 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.

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".

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.

I.5.5.3.- Máquina Virtual

Se trata de un tipo de sistemas operativos que presentan una interface 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

I.5.5.4.- 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 muy básicas de memoria, entrada/salida, archivos y procesos, dejando a los servidores proveer lo 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.

I.5.6.- 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 accesa a los servicios. Bajo esta clasificación se pueden detectar dos tipos principales: sistemas operativos de red y sistemas operativos distribuidos.

I.5.6.1.- 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 sin fin de otras actividades. El punto crucial de estos sistemas es que el usuario debe saber la sintaxis de un conjunto de comandos o llamadas al sistema para ejecutar estas operaciones, además de la ubicación de los recursos 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

I.5.6.2.- Sistemas Operativos Distribuidos

Los sistemas operativos distribuidos abarcan los servicios de los de red, logrando integrar recursos (impresoras, unidades de respaldo, memoria, procesos, unidades centrales de proceso), en una sola máquina virtual que el usuario accesa 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 mas 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 distribuir las 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. El sistema integrador de los microprocesadores que hacer 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 ni cuenta se da de estas fallas, es decir, obtiene acceso a recursos en forma transparente. Los sistemas distribuidos deben de ser muy confiables, ya que si un componente del sistema se compone 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.

I.6.- PROCESO DE LOS SISTEMAS OPERATIVOS

Uno de los conceptos más importantes que gira entorno a un sistema operativo es el de proceso. Un proceso es un programa en ejecución junto con el entorno asociado (registros, variables, etc.).

El corazón de un sistema operativo es el núcleo, un programa de control que reacciona ante cualquier interrupción de eventos externos y que da servicio a los procesos, creándolos, terminándolos y respondiendo a cualquier petición de servicio por parte de los mismos.

I.6.1.- PLANIFICACIÓN DEL PROCESADOR

La planificación del procesador se refiere a la manera o técnicas que se usan para decidir cuánto tiempo de ejecución y cuando se le asignan a cada proceso del sistema. Obviamente, si el sistema es monousuario y monotarea no hay mucho que decidir, pero en el resto de los sistemas esto es crucial para el buen funcionamiento del sistema.

I.6.2.- CARACTERÍSTICAS A CONSIDERAR DE LOS PROCESOS

No todos los equipos de cómputo procesan el mismo tipo de trabajos, y un algoritmo de planificación que en un sistema funciona excelente puede dar un rendimiento pésimo en otro cuyos procesos tienen características diferentes. Estas características pueden ser:

I.6.2.1.- Cantidad de Entrada/Salida

Existen procesos que realizan una gran cantidad de operaciones de entrada y salida (aplicaciones de bases de datos, por ejemplo).

I.6.2.2.- Cantidad de Uso de CPU

Existen procesos que no realizan muchas operaciones de entrada y salida, sino que usan intensivamente la unidad central de procesamiento. Por ejemplo, operaciones con matrices.

I.6.2.3.- Procesos de Lote o Interactivos

Un proceso de lote es más eficiente en cuanto a la lectura de datos, ya que generalmente lo hace de archivos, mientras que un programa interactivo espera mucho tiempo (no es lo mismo el tiempo de lectura de un archivo que la velocidad en que una persona teclea datos) por las respuestas de los usuarios.

I.6.2.4.- Procesos en Tiempo Real

Si los procesos deben dar respuesta en tiempo real se requiere que tengan prioridad para los turnos de ejecución.

I.6.2.5.- Longevidad de los Procesos

Existen procesos que típicamente requerirán varias horas para finalizar su labor, mientras que existen otros que solo necesitan algunos segundos.

Un proceso es una actividad que se apoya en datos, recursos, un estado en cada momento y un programa.

I.6.2.6.- Bloque de control de procesos

Un proceso se representa desde el punto de vista del sistema operativo, por un conjunto de datos donde se incluyen el estado en cada momento, recursos utilizados, registros, etc., denominado Bloque de Control de Procesos (PCB).

I.6.2.6.1.- Los objetivos del bloque de control de procesos

- Localización de la información sobre el proceso por parte del sistema operativo.
- Mantener registrados los datos del proceso en caso de tener que suspender temporalmente su ejecución o reanudarla.

I.6.2.6.2.- La información contenida en el bloque de control:

- Estado del proceso. Información relativa al contenido del controlador del programa (Program Counter, PC), estado de procesador en cuanto a prioridad del proceso, modo de ejecución, etc., y por último el estado de los registros internos de la computadora.
- Estadísticas de tiempo y ocupación de recursos para la gestión de la planificación del procesador.
- Ocupación de memoria interna y externa para el intercambio (swapping).
- Recursos en uso (normalmente unidades de entrada/salida).
- Archivos en uso
- Privilegios

Estas informaciones se encuentran en memoria principal en disco y se accede a ellas en los momentos en que se hace necesaria su actualización o consulta. Los datos relativos al estado del proceso siempre se encuentran en memoria principal. Existe un Bloque de Control de Sistema (SCB) con objetivos similares al anterior y entre los que se encuentra el enlazado de los bloques de control de procesos existentes en el sistema.

El cambio de contexto se producirá en caso de ejecución de una instrucción privilegiada, una llamada al sistema operativo o una interrupción, es decir, siempre que se requiera la atención de algún servicio del sistema operativo.

I.6.3.- ESTADO DE LOS PROCESOS

Los bloques de control de los procesos se almacenan en colas, cada una de las cuales representa un estado particular de los procesos, existiendo en cada bloque, entre otras informaciones. Los estados de los procesos son internos del sistema operativo y transparente al usuario.

Los estados de los procesos se pueden dividir en dos tipos:

- Activos e
- Inactivos.

I.6.3.1.- Estados activos

Son aquellos que compiten con el procesador o están en condiciones de hacerlo. Se dividen en:

I.6.3.1.1.- Ejecución

Estado en el que se encuentra un proceso cuando tiene el control del procesador. En un sistema monoprocesador este estado sólo lo puede tener un proceso.

I.6.3.1.2.- Preparado

Aquellos procesos que están dispuestos para ser ejecutados, pero no están en ejecución por alguna causa (Interrupción, haber entrado en cola estando otro proceso en ejecución, etc.).

I.6.3.1.3.- Bloqueado

Son los procesos que no pueden ejecutarse de momento por necesitar algún recurso no disponible (generalmente recursos de entrada/salida).

I.6.3.2.- Estados inactivos

Son aquellos que no pueden competir por el procesador, pero que pueden volver a hacerlo por medio de ciertas operaciones.

En estos estados se mantiene el bloque de control de proceso aparcado hasta que vuelva a ser activado. Se trata de procesos que no han terminado su trabajo que lo han impedido y que pueden volver a activarse desde el punto en que se quedaron sin que tengan que volver a ejecutarse desde el principio.

Son de dos tipos:

I.6.3.2.1.- Suspendido bloqueado

Es el proceso que fue suspendido en espera de un evento, sin que hayan desaparecido las causas de su bloqueo.

I.6.3.2.2.- Suspendido programado

Es el proceso que han sido suspendido, pero no tiene causa para estar bloqueado.

I.6.4.- OPERACIONES SOBRE PROCESOS

Los sistemas operativos actuales poseen una serie de funciones cuyo objetivo es el de la manipulación de los procesos. Las operaciones que se pueden hacer sobre un proceso son las siguientes:

I.6.4.1.- Crear el proceso

Se produce con la orden de ejecución del programa y suele necesitar varios argumentos, como el nombre y la prioridad del proceso. Aparece en este momento el PCB, que será insertado en la cola de procesos preparados.

La creación de un proceso puede ser de dos tipos:

I.6.4.1.1.- Jerárquica

En ella, cada proceso que se crea es hijo del proceso creador y hereda el entorno de ejecución de su padre. El primer proceso que ejecuta un usuario será hijo del intérprete de comandos con el que interactúa.

I.6.4.1.2.- No jerárquica

Cada proceso creado por otro proceso se ejecuta independientemente de su creador con un entorno diferente. Es un tipo de creación que no suele darse en los sistemas operativos actuales.

I.6.4.2.- Destruir un proceso

Se trata de la orden de eliminación del proceso con la cual el sistema operativo destruye su PCB.

I.6.4.3.- Suspender un proceso

Es un proceso de alta prioridad que paraliza un proceso que puede ser reanudado posteriormente. Suele utilizarse en ocasiones de mal funcionamiento o sobrecarga del sistema.

I.6.4.4.- Reanudar un proceso

Trata de activar un proceso que ha sido previamente suspendido.

I.6.4.5.- Cambiar la prioridad de un proceso

Temporizar la ejecución de un proceso. Hace que un determinado proceso se ejecute cada cierto tiempo (segundos, minutos, horas...) por etapas de una sola vez, pero transcurrido un periodo de tiempo fijo.

I.6.4.6.- Despertar un proceso

Es una forma de desbloquear un proceso que habrá sido bloqueado previamente por temporización o cualquier otra causa.

I.6.5.- PRIORIDADES DE LOS PROCESOS

Todo proceso por sus características e importancia lleva aparejadas unas determinadas necesidades de ejecución en cuanto a urgencia y asignación de recursos.

Las prioridades según los sistemas operativos se pueden clasificar del siguiente modo:

Asignadas por el sistema operativo.

Asignadas por el propietario.

- Estáticas
- Dinámicas

I.7.- EL NÚCLEO DEL SISTEMA OPERATIVO

Todas las operaciones en las que participan procesos son controladas por la parte del sistema operativo denominada núcleo (nucleus, core o kernel, en inglés). El núcleo normalmente representa sólo una pequeña parte de lo que por lo general se piensa que es todo el sistema operativo, pero es tal vez el código que más se utiliza. Por esta razón, el núcleo reside por lo regular en la memoria principal, mientras que otras partes del sistema operativo son cargadas en la memoria principal sólo cuando se necesitan.

Los núcleos se diseñan para realizar “el mínimo” posible de procesamiento en cada interrupción y dejar que el resto lo realice el proceso apropiado del sistema, que puede operar mientras el núcleo se habilita para atender otras interrupciones. El núcleo de un sistema operativo normalmente contiene el código necesario para realizar las siguientes funciones:

- Manejo de interrupciones
- Creación y destrucción de procesos
- Cambio de estado de los procesos
- Despacho
- Suspensión y reanudación de procesos
- Sincronización de procesos
- Comunicación entre procesos
- Manipulación de los bloques de control de procesos
- Apoyo para las actividades de entrada/salida
- Apoyo para asignación y liberación de memoria
- Apoyo para el sistema de archivos
- Apoyo para el mecanismo de llamada y retorno de un procedimiento
- Apoyo para ciertas funciones de contabilidad del sistema

I.8.- NÚCLEO O KERNEL Y NIVELES DE UN SISTEMA OPERATIVO

El Kernel consiste en la parte principal del código del sistema operativo, el cual se encargan de controlar y administrar los servicios y peticiones de recursos y de hardware con respecto a uno o varios procesos, este se divide en 5 capas:

I.8.1.- GESTIÓN DE MEMORIA

Que proporciona las facilidades de bajo nivel para la gestión de memoria secundaria necesaria para la ejecución de procesos.

I.8.2.- PROCESADOR

Se encarga de activar los cuantums de tiempo para cada uno de los procesos, creando interrupciones de hardware cuando no son respetadas.

I.8.3.- ENTRADA/SALIDA

Proporciona las facilidades para poder utilizar los dispositivos de E/S requeridos por procesos.

I.8.4.- INFORMACIÓN O APLICACIÓN O INTÉRPRETE DE LENGUAJES

Facilita la comunicación con los lenguajes y el sistema operativo para aceptar las ordenes en cada una de las aplicaciones. Cuando se solicitan ejecutando un programa el software de este nivel crea el ambiente de trabajo e invoca a los procesos correspondientes.

I.8.4.1.- Control de Archivos

Proporciona la facilidad para el almacenamiento a largo plazo y manipulación de archivos con nombre, va asignando espacio y acceso de datos en memoria.

I.8.5.- EL NÚCLEO Y LOS PROCESOS

I.8.5.1.- El núcleo

El núcleo (Kernel) de un sistema operativo es un conjunto de rutinas cuya misión es la de gestionar el procesador, la memoria, la entrada/salida y el resto de procesos disponibles en la instalación. Toda esta gestión la realiza para atender al funcionamiento y peticiones de los trabajos que se ejecutan en el sistema.

Los procesos relacionados con la entidad básica de los sistemas operativos actuales:

I.8.5.2.- Los procesos

El esquema general del mismo es el siguiente:

- Definición y concepto de proceso.
- El Bloque de Control de Proceso (PCB) como imagen donde el sistema operativo ve el estado del proceso.

- Estados por los que pasa un proceso a lo largo de su existencia en la computadora.
- Operaciones que se pueden realizar sobre un proceso.
- Clasificación de los procesos según su forma de ejecución, de carga, etc.

I.9.- DISPOSITIVOS DE ENTRADA Y SALIDA

El código destinado a manejar la entrada y salida de los diferentes periféricos en un sistema operativo es de una extensión considerable y sumamente complejo. Resuelve las necesidades de sincronizar, atrapar interrupciones y ofrecer llamadas al sistema para los programadores.

I.9.1.- TIPOS DE DISPOSITIVOS DE ENTRADA Y SALIDA

- Dispositivos orientados a bloques y
- Dispositivos orientados a caracteres.

I.9.1.1.- Orientados a Bloques

Los dispositivos orientados a bloques tienen la propiedad de que se pueden direccionar, esto es, el programador puede escribir o leer cualquier bloque del dispositivo realizando primero una operación de posicionamiento sobre el dispositivo. Los dispositivos más comunes orientados a bloques son los discos duros, la memoria, discos compactos y, posiblemente, unidades de cinta.

I.9.1.2.- Orientados a Caracteres

Los dispositivos orientados a caracteres son aquellos que trabajan con secuencias de bytes sin importar su longitud ni ninguna agrupación en especial. No son dispositivos direccionables. Ejemplos de estos dispositivos son el teclado, la pantalla o display y las impresoras.

La clasificación anterior no es perfecta, porque existen varios dispositivos que generan entrada o salida que no pueden englobarse en esas categorías. Por ejemplo, un reloj que genera pulsos. Sin embargo, aunque existan algunos periféricos que no se puedan categorizar, todos están administrados por el sistema operativo por medio de una parte electrónica - mecánica y una parte de software.

I.9.2.- PRINCIPIOS DEL SOFTWARE DE ENTRADA Y SALIDA

Los principios de software en la entrada - salida se resumen en cuatro puntos:

- manejadores de interrupciones,
- manejadores de dispositivos,
- software que sea independiente de los dispositivos y
- software para usuarios.

I.9.2.1.- Manejadores de Interrupciones

El primer objetivo referente a los manejadores de interrupciones consiste en que el programador o el usuario no debe darse cuenta de los manejos de bajo nivel para los casos en que el dispositivo está ocupado y se debe suspender el proceso o sincronizar algunas tareas. Desde el punto de vista del proceso o usuario, el sistema simplemente se tardó más o menos en responder a su petición.

I.9.2.2.- Manejadores de Dispositivos

El sistema debe proveer los manejadores de dispositivos necesarios para los periféricos, así como ocultar las peculiaridades del manejo interno de cada uno de ellos, tales como el formato de la información, los medios mecánicos, los niveles de voltaje y otros. Por ejemplo, si el sistema tiene varios tipos diferentes de discos duros, para el usuario o programador las diferencias técnicas entre ellos no le deben importar, y los manejadores le deben ofrecer el mismo conjunto de rutinas para leer y escribir datos.

I.9.2.3.- Software que sea independiente de los dispositivos

Este es un nivel superior de independencia que el ofrecido por los manejadores de dispositivos. Aquí el sistema operativo debe ser capaz, en lo más posible, de ofrecer un conjunto de utilerías para acceder a periféricos o programarlos de una manera consistente. Por ejemplo, que para todos los dispositivos orientados a bloques se tenga una llamada para decidir si se desea usar 'buffers' o no, o para posicionarse en ellos.

I.9.2.4.- Software para Usuarios

La mayoría de las rutinas de entrada - salida trabajan en modo privilegiado, o son llamadas al sistema que se ligan a los programas del usuario formando parte de sus aplicaciones y que no le dejan ninguna flexibilidad al usuario en cuanto a la apariencia de los datos. Existen otras librerías en donde el usuario si tiene poder de decisión (por ejemplo la llamada a "printf" en el lenguaje "C"). Otra facilidad ofrecida son las áreas de trabajos encolados (spooling áreas), tales como las de impresión y correo electrónico.

I.9.3.- MANEJO DE LOS DISPOSITIVOS DE E/S

En el manejo de los dispositivos de E/S es necesario, introducir dos nuevos términos:

I.9.3.1.- Buffering (uso de memoria intermedia)

El buffering trata de mantener ocupados tanto la CPU como los dispositivos de E/S. La idea es sencilla, los datos se leen y se almacenan en un buffer, una vez que los datos se han leído y la CPU va a iniciar inmediatamente la operación con ellos, el dispositivo de entrada es instruido para iniciar inmediatamente la siguiente lectura. La CPU y el dispositivo de entrada permanecen ocupados. Cuando la CPU esté libre para el siguiente grupo de datos, el dispositivo de entrada habrá terminado de leerlos. La CPU podrá empezar el proceso de los

últimos datos leídos, mientras el dispositivo de entrada iniciará la lectura de los datos siguientes.

Para la salida, el proceso es análogo. En este caso los datos de salida se descargan en otro buffer hasta que el dispositivo de salida pueda procesarlos.

Este sistema soluciona en forma parcial el problema de mantener ocupados todo el tiempo la CPU y los dispositivos de E/S. Ya que todo depende del tamaño del buffer y de la velocidad de procesamiento tanto de la CPU como de los dispositivos de E/S.

El manejo de buffer es complicado. Uno de los principales problemas reside en determinar tan pronto como sea posible que un dispositivo de E/S haya finalizado una operación. Este problema se resuelve mediante las interrupciones. Tan pronto como un dispositivo de E/S acaba con una operación interrumpe a la CPU, en ese momento la CPU detiene lo que está haciendo e inmediatamente transfiere el control a una posición determinada. Normalmente las instrucciones que existen en esta posición corresponden a una rutina de servicio de interrupciones. La rutina de servicio de interrupción comprueba si el buffer no está lleno o no está vacío y entonces inicia la siguiente petición de E/S. La CPU puede continuar entonces el proceso interrumpido.

Cada diseño de computadora tiene su propio mecanismo de interrupción, pero hay varias funciones comunes que todos contemplan.

El buffering puede ser de gran ayuda pero pocas veces es suficiente.

I.9.3.2.- Spooling

El problema con los sistemas de cintas es que una lectora de tarjetas no podía escribir sobre un extremo mientras la CPU leía el otro. Los sistemas de disco eliminaron esa dificultad, moviendo la cabeza de un área del disco a otra.

En un sistema de discos, las tarjetas se leen directamente desde la lectora sobre el disco. La posición de las imágenes de las tarjetas se registra en una tabla mantenida por el sistema operativo. En la tabla se anota cada trabajo una vez leído. Cuando se ejecuta un trabajo sus peticiones de entrada desde la tarjeta se

satisfacen leyendo el disco. Cuando el trabajo solicita la salida, ésta se copia en el buffer del sistema y se escribe en el disco. Cuando la tarea se ha completado se escribe en la salida realmente.

Esta forma de procesamiento se denomina spooling, utiliza el disco como un buffer muy grande para leer tan por delante como sea posible de los dispositivos de entrada y para almacenar los ficheros hasta que los dispositivos de salida sean capaces de aceptarlos.

La ventaja sobre el buffering es que el spooling solapa la E/S de un trabajo con la computación de otro. Es una característica utilizada en la mayoría de los sistemas operativos.

Afecta directamente a las prestaciones. Por el costo de algo de espacio en disco y algunas tablas, la CPU puede simultanear la computación de un trabajo con la E/S de otros. De esta manera, puede mantener tanto a la CPU como a los dispositivos de E/S trabajando con un rendimiento mucho mayor.

Además mantiene una estructura de datos llama job spooling, que hace que los trabajos ya leídos permanezcan en el disco y el sistema operativo puede seleccionar cual ejecutar, por lo tanto se hace posible la planificación de trabajos.

I.10.-ADMINISTRACIÓN DE ARCHIVOS

Un archivo es un conjunto de información, que se encuentra almacenada o guardada en la memoria principal del computador, en el disco duro, en el disquete flexible o en los discos compactos (Cd-Rom).

Antes de que un archivo pueda leerse o escribirse en él, debe abrirse, momento en el cual se verifican los permisos. Estos archivos se abren especificando en el computador la ruta de acceso al archivo desde el directorio raíz, que es la unidad principal del disco del computador, este puede ser un disco duro o disco flexible. Entonces el sistema operativo visualiza el entorno al abrir un archivo.

Uno de los problemas mas frecuentes en el manejo de archivos son los deadlock, un deadlock es una situación no deseada de espera indefinida y se da cuando en un grupo de procesos, dos o más procesos de ese grupo esperan por llevar a

cabo una tarea que será ejecutada por otro proceso del mismo grupo, entonces se produce el bloqueo. Los bloqueos se dan tanto en los sistemas operativos tradicionales como en los distribuidos, solo que en estos últimos es más difícil de prevenirlos, evitarlos e incluso detectarlos, y si se los logra detectar es muy complicado solucionarlos ya que la información se encuentra dispersa por todo el sistema.

Una vez que un deadlock se detecta, es obvio que el sistema está en problemas y lo único que resta por hacer es una de dos cosas: tener algún mecanismo de suspensión o reanudación que permita copiar todo el contexto de un proceso incluyendo valores de memoria y aspecto de los periféricos que esté usando para reanudarlo otro día, o simplemente eliminar un proceso o arrebatárle el recurso, causando para ese proceso la pérdida de datos y tiempo.

I.11.-SEGURIDAD DE UN SISTEMA OPERATIVO

En los sistemas operativos se requiere tener una buena seguridad informática, tanto del hardware, programas y datos, previamente haciendo un balance de los requerimientos y mecanismos necesarios. Con el fin de asegurar la integridad de la información contenida.

Dependiendo de los mecanismos utilizados y de su grado de efectividad, se puede hablar de sistemas seguros e inseguros. En primer lugar, deben imponerse ciertas características en el entorno donde se encuentra la instalación de los equipos, con el fin de impedir el acceso a personas no autorizadas, mantener un buen estado y uso del material y equipos, así como eliminar los riesgos de causas de fuerza mayor, que puedan destruir la instalación y la información contenida.

En la actualidad son muchas las violaciones que se producen en los sistemas informáticos, en general por acceso de personas no autorizadas que obtienen información confidencial pudiendo incluso manipularla. En ocasiones este tipo de incidencias resulta grave por la naturaleza de los datos; por ejemplo si se trata de datos bancarios, datos oficiales que puedan afectar a la seguridad de los estados, etc.

El software mal intencionado que se produce por diversas causas, es decir pequeños programas que poseen gran facilidad para reproducirse y ejecutarse, cuyos efectos son destructivos nos estamos refiriendo a los virus informáticos.

Para esto, se analizan cuestiones de seguridad desde dos perspectivas diferentes la seguridad externa y la seguridad interna.

Todos los mecanismos dirigidos a asegurar el sistema informático sin que el propio sistema intervenga en el mismo se engloban en lo que podemos denominar seguridad externa.

La seguridad externa puede dividirse en dos grandes grupos:

I.11.1.-SEGURIDAD FÍSICA

Engloba aquellos mecanismos que impiden a los agentes físicos la destrucción de la información existente en el sistema; entre ellos podemos citar el fuego, el humo, inundaciones descargas eléctricas, campos magnéticos, acceso físico de personas con no muy buena intención, entre otros.

I.11.2.-SEGURIDAD DE ADMINISTRACIÓN

Engloba los mecanismos más usuales para impedir el acceso lógico de personas físicas al sistema.

Todos los mecanismos dirigidos a asegurar el sistema informático, siendo el propio sistema el que controla dichos mecanismos, se engloban en lo que podemos denominar seguridad interna.

Referencia Bibliográfica:

Tomado de la página Web: www.monografias.com

I.12.-DEFINICIÓN DE RED HAT

Es una distribución Linux creada por Red Hat, que fue una de las más populares en los entornos de usuarios domésticos. Es la compañía responsable de la creación y mantenimiento de una distribución del sistema operativo GNU/Linux que lleva el mismo nombre: Red Hat Enterprise Linux.

I.12.1.-HISTORIA DE RED HAT

Es una de las distribuciones Linux de "mediana edad". La versión 1.0 fue presentada el 3 de noviembre de 1994. Fue la primera distribución que usó RPM*, como su formato de paquete, y en un cierto plazo ha servido como el punto de partida para varias otras distribuciones

Red Hat es famoso en todo el mundo por los diferentes esfuerzos orientados a apoyar el movimiento del software libre. No sólo trabaja en el desarrollo de una de las distribuciones más populares de Linux, sino también en la comercialización de diferentes productos y servicios basados en software de código abierto. Asimismo, poseen una amplia infraestructura en la que se cuentan más de 2.000 empleados en 28 lugares del mundo.

Red Hat garantiza el óptimo funcionamiento de sus productos. Ofrece un amplio rango de servicios, tales como consultoría e ingeniería haciendo exitosa cualquier implementación de Código Abierto. Ofrece un amplio rango de soluciones desde el escritorio hasta servidores de misión crítica.

Red Hat combina la innovadora tecnología del código abierto y la estabilidad de una verdadera plataforma de clase empresarial.

* *RPM Package Manager* (originalmente llamado *Red Hat Package Manager*) es una herramienta de administración de paquetes pensada básicamente para Linux. Es capaz de instalar, actualizar, desinstalar, verificar y solicitar programas.

I.12.2.-DISTRIBUCIONES DE RED HAT

Red Hat es famoso en todo el mundo por los diferentes esfuerzos orientados a apoyar el movimiento del software libre. No sólo trabajan en el desarrollo de una de las distribuciones más populares de Linux, sino también en la comercialización de diferentes productos y servicios basados en software de código abierto, entre las cuales tenemos:

I.12.2.1.-Red Hat Enterprise Linux AS

Red Hat Enterprise Linux AS es la solución Open Source de más alto nivel de la familia de sistemas operativos Red Hat Enterprise, soportando servidores de misión crítica y alta disponibilidad. Además, es la solución open source de más alto nivel de los sistemas.

Red Hat Enterprise. Se encuentra certificada por los más importantes ISV del mercado y cuenta con la mayor cantidad de certificaciones de Hardware a nivel mundial.

I.12.2.2.-Red Hat Enterprise Linux ES

Red Hat Enterprise Linux ES. Es la solución perfecta de la familia de sistemas operativos para cubrir las necesidades de las soluciones de negocios de nuestros días. Provee la infraestructura de sistema operativo y conectividad necesaria para un gran rango de servidores entry-level y servidores de aplicaciones departamentales. Es ideal para conectarse, para servidor de archivos, servidor de impresión, servidor de mail, servidor Web y paquetes de aplicaciones comerciales.

I.12.2.3.-Red Hat Enterprise Linux WS

Es el socio ideal para Enterprise Linux AS y para Enterprise Linux ES. Está diseñado para aplicaciones de productividad, desarrollo de software y

aplicaciones dirigidas a clientes ISV. Es la solución ideal para estaciones de trabajo, y desarrollo de software.

Los desarrolladores encontrarán una gran herramienta para el diseño e implementación de sus aplicaciones. Red Hat Enterprise Linux WS es totalmente compatible con otros productos de la familia de Red Hat Enterprise y provee tecnología y servicios complementarios.

I.12.2.4.-Red Hat Desktop

Red Hat Desktop es un sistema de cliente de alta calidad, muy completo y conveniente para implementaciones del desktop, donde la seguridad y las posibilidades de gestión son claves. Proporciona un sistema de cliente de alta calidad, totalmente completo y que es conveniente para el empleo en una amplia gama de implementaciones de escritorio.

I.12.3.-CARACTERÍSTICAS ESPECIALES DE RED HAT

Red Hat es instalado con un ambiente gráfico llamado Anaconda, diseñado para su fácil uso por novatos. También incorpora una herramienta llamada Lokkit para configurar las capacidades de Cortafuegos. Red Hat Linux carece de muchas características debido a posibles problemas de copyright y patentes. Por ejemplo, el soporte al formato MP3 está desactivado tanto en Rhythmbox como en XMMS; en su lugar, Red Hat recomienda usar Ogg Vorbis, que no tiene patentes. Sin embargo, el soporte para MP3 puede ser instalado luego, aunque se requiere el pago de regalías en los Estados Unidos. El soporte al formato NTFS también está ausente, pero también puede ser instalado libremente.

Referencia Bibliográfica:

Tomado de la página Web:

<http://www.tu-chemnitz.de/docs/lindocs/RH9/RH-DOCS/rhl-ig-x86-es-9/index.html>

I.13.-GLASSFISH DEFINICIÓN.

GlassFish es un servidor de aplicaciones que implementa las tecnologías definidas en la plataforma Java EE y permite ejecutar aplicaciones que siguen esta especificación. Es gratuito y de código libre, se distribuye bajo la licencia CDDL* y la GNU GPL*.

*Desarrollo Común y Licencia de Distribución (CDDL) Es una licencia de código abierto (OSI) y libre, producida por Sun Microsystems, basada en la Mozilla Public License (MPL).

*Licencia Pública General de GNU. Es una licencia creada por la Free Software Foundation a mediados de los 80, y está orientada principalmente a proteger la libre distribución, modificación y uso de software.

GlassFish tiene como base al servidor Sun Java System Application.

- GlassFish es la implementación de referencia (RI) de Java EE 5.
- GlassFish es un proyecto Open Source modular que permite incluir sus librerías como parte de otros frameworks, toolkits y productos.
- GlassFish es la base de código de las distribuciones estables, certificadas y con opción de contratar soporte y mantenimiento del Servidor de Aplicaciones de Sun: Sun Java System Application Server.

- GlassFish es una comunidad que contribuye mediante código, detección de bugs, foros de discusión, feedback, documentación, wikis, blogs y otros medios a la difusión y éxito del proyecto. La comunidad GlassFish colabora en la compatibilidad de otros frameworks JAVA Open Source.

I.13.1.-¿QUÉ ES UN SERVIDOR?

Un servidor es un ordenador dedicado a ejecutar una aplicación servidora y, como aplicación servidora, se conoce al software preparado para aceptar conexiones de otras maquinas con peticiones de servicio y resolver dichas peticiones enviando respuestas de vuelta a las maquinas que originaron las conexiones.

I.13.2.-DIFERENCIAS ENTRE SERVIDORES HTTP, SERVIDORES WEB Y SERVIDORES DE APLICACIONES

1. Servidor HTTP

Es aquel software capaz de procesar peticiones http (Ej. Apache HTTP Server, Oracle HTTP Server).

2. Servidor WEB

Es aquel software que, además de ser un Servidor HTTP es un “contenedor de servlets” Ej. (Apache Tomcat)

3. Servidor de Aplicaciones

Es aquel software que, además de ser un Servidor WEB, implementa el resto de especificaciones de J2EE (ahora Java EE); EJB, JMS, Web Services para Java, etc. Ej. (Weblogic, Websphere, JBoss, Geronimo, Glassfish, etc.)

I.13.3.-GLASSFISH ¿POR QUÉ? ¿PARA QUÉ? ¿PARA QUIÉN?

Dentro de la importancia de Glassfish es conveniente recalcar el ¿por qué?, ¿para qué? y ¿Para quién? Está destinado este servidor de aplicaciones.

I.13.3.1.-¿Por qué?

Porque está hecho/supervisado por SUN y esta es la compañía que siempre va un paso por delante en todo lo relativo al mundo JAVA y a las especificaciones de Java EE.

Porque tiene soporte comercial disponible y esto es un requisito indispensable para alguna de las empresas para las que pueden desarrollar software

Porque es la implementación de referencia de las especificaciones de Java EE.

Lo que funciona en Glassfish debe funcionar en cualquier sitio y, si no funciona en otro servidor de aplicaciones, probablemente sea un problema del servidor, no de la aplicación.

Es fácil de usar y no consume muchos recursos.

I.13.3.2.-¿Para qué?

Para aplicaciones Web hechas en JAVA de gran complejidad, necesaria escalabilidad y alta disponibilidad y con uso exhaustivo de las tecnologías de Java EE 1.5 (EJB3, JMS, etc.)

I.13.3.3.-¿Para quién?

Para desarrolladores Java EE, como entorno de pruebas, integración y producción. Así como Arquitectos o Jefes de Proyecto que quieran comprobar que su aplicación Java EE es compatible al 100% con la especificación Java EE 5.

I.13.4.-ADMINISTRANDO GLASSFISH (USUARIOS, SEGURIDAD Y DESPLIEGUE DE APLICACIONES)

Lo primero que vamos a hacer es añadir el subdirectorio /bin, del directorio de instalación en el PATH del sistema para que este accesible desde cualquier ruta.

Una vez realizado el paso anterior, abrimos una terminal en:

(Aplicaciones>accesorios>terminal).

Recordemos que debemos acceder como administrador para realizar estas operaciones... y luego escribimos

```
asadmin start-domain domain1
```

Con lo que se inicializará el servidor utilizando el usuario y clave por defecto (admin/adminadmin).

Para comprobar que el servidor esta arrancado y funciona, podemos introducir esta URL en el navegador: <http://localhost:8080/>

Para acceder a la consola de administración, la URL es <http://localhost:4848/> y aquí introduciremos los antes mocionados usuario y clave por defecto admin y adminadmin.

Antes de nada, es deseable crear una copia del dominio para poder hacer pruebas y comprobar cómo se pueden guardar varias configuraciones en una única instancia de servidor. Se detiene el servidor con el siguiente comando:

```
asadmin stop-domain domain1
```

Y se copia el directorio glassfish/domains/domain1 a un nuevo directorio denominado glassfish/domains/domain2. Se inicializa de nuevo el servidor pero, esta vez, indicando que se inicia el dominio domain2

```
asadmin start-domain domain2
```

Tener en cuenta que si se tiene un solo dominio se puede acceder (start), con solo escribir domain. De lo contrario especificar al tipo de dominio que se quiere acceder Ej.: domain1, domain2, etc.

I.13.5.-SEGURIDAD

Una de las primeras cosas que debe hacer es cambiar la contraseña de Administrador, para hacerlo, se conecta a la consola de administración (<http://localhost:4848/>) y, una vez más, se consigue la autenticación mediante la clave por defecto admin/adminadmin.

En el árbol de navegación de la izquierda se elegirá la opción de DOMINIO y, dentro de la misma, la pestaña de cambio de contraseña de administrador.

Una vez modificada la contraseña en este ejemplo considerada como, "sistemas" se cerrará la sesión de la consola de administración y se procederá de nuevo a la autenticación para comprobar que el cambio es efectivo.

Se detendrá el servidor para que se apliquen los cambios.

```
asadmin stop-domain domain2
```

Y se volverá a iniciar el domain2.

```
asadmin start-domain domain2
```

Aparece un fallo de autenticación ya que se ha cambiado la contraseña del usuario administrador y la contraseña por defecto ya no vale.

Glassfish no permite en su versión Enterprise el paso dinámico de contraseñas por línea de comandos. Así, es obligatorio crear un fichero de claves, el que copiaremos en el directorio misclavecitas.txt, en el c:\. En este fichero se debe escribir la contraseña para el usuario administrador por defecto:

```
AS_ADMIN_PASSWORD=sistemas
```

Aunque proporciona más seguridad, la clave sigue en crudo y cualquiera la puede ver. Por eso, otra de las cosas que es conveniente establecer para asegurar el dominio es el master password, una clave que da acceso a toda la funcionalidad de Glassfish. Hasta que no se establezca, su valor es changeit y cualquiera la puede modificar, así que, con el servidor detenido, se ejecutará este comando:

```
asadmin change-master-password <nombre de dominio>
```

y estableceremos esta Master Password (en el ejemplo de este tutorial, sistemas). Si se intenta modificarla de nuevo, ahora pide la antigua. Ya está asegurada y no se guarda en crudo, sino encriptada como parte de la Keystore de Java.

Ahora, al arrancar el servidor con el dominio modificado, deberemos incluir dos parámetros al arrancar un dominio; --user <nombre de usuario> y --passwordfile <ruta_absoluta_fichero_claves> para poder autenticarnos en un dominio.

```
asadmin start-domain --user admin --passwordfile c:\misclavecitas.txt domain2
```

Además, como ya no tiene el valor por defecto, va a pedir la Master Password configurada.

I.13.6.-CONFIGURACIÓN XML

Se puede comprobar que el nombre de dominio que aparece en la consola de administración es domain1 en vez de domain2. Esto ocurre porque se ha copiado el primero pero se puede modificar directamente en el fichero de configuración.

glassfish/domains/domain2/config/domain.xml (al final del fichero).

Otra de las cosas que se pueden configurar en este fichero son los puertos de escucha del servidor, aunque también se puede hacer a través de la consola de Administración (Configuraciones > Server Config > Servicios http > Escuchas HTTP)

La mayoría de estos cambios requieren el reinicio del servidor, parando y arrancando de nuevo el dominio.

I.14.-DESPLIEGUE DE APLICACIONES

El despliegue de aplicaciones se puede hacer de dos maneras; a través de la consola de administración y de forma manual, copiando las aplicaciones en la carpeta /autodeploy del dominio de trabajo.

Lo ideal es desplegar las aplicaciones a través de la consola de administración porque se pueden hacer despliegues en remoto –sin tener acceso al filesystem del servidor- y porque los posibles errores de despliegue son más evidentes. Lo malo de la consola es que es mucho más lenta que copiar la aplicación

directamente y que, a veces, cuando se ha “machacado” el servidor con despliegues y repliegues de la misma aplicación... no se entera (por ejemplo, depurando aplicaciones con módulos EJBs). En ese caso, hay que borrar a mano los ficheros generados en la carpeta /generated y /applications relacionados con la aplicación corrupta, y que siguen allí aunque haya sido replegada del servidor. Como ejemplo, se va a desplegar una aplicación muy sencilla, simpleapp.war4, de las dos maneras que se han comentado, para aprender a hacerlo de una forma y de otra.

A través de la consola de Administración, se autentica en la aplicación y, en la pantalla principal, se pulsa el enlace “Implementar aplicación Web (.war)” que lleva a la página donde se “subirá” el war al servidor.

Aunque, el fichero y el servidor estén en la misma máquina, es recomendable utilizar la opción de uploading porque sino habrá que navegar hasta el fichero, mediante una pesada navegación Web.

Se selecciona el .war, Glassfish utiliza el nombre del fichero como nombre y contexto por defecto de la aplicación.

Se cambia el nombre a “simpleapp1”, se comprueba que la aplicación esté disponible mediante el checkbox correspondiente, se acepta y se despliega la aplicación. Para comprobar que todo está bien, introducimos la siguiente ruta en el navegador: <http://localhost:8080/simpleapp/simpleservlet>.

Ahora se va a cambiar el contexto a la aplicación ya desplegada:

Se navega a Aplicaciones > Aplicaciones Web y allí se selecciona simpleapp, se cambia el contexto a simpleapp1 y se guardan los cambios. No hace falta reiniciar la aplicación. Ahora se puede acceder a la aplicación a través de:

<http://localhost:8080/simpleapp1/simpleservlet>

Ahora, se selecciona el mismo war y se copia directamente sobre la carpeta /domain2/autodeploy y se comprueba que la aplicación se ha desplegado y el servidor ha utilizado de nombre de aplicación y de contexto el nombre del war.

Si no se hubiera cambiado el nombre de la aplicación que desplegué a través de la consola de administración, el servidor habría interpretado que era una nueva versión de la misma y la habría sobrescrito.

Se puede comprobar que todo ha ido bien leyendo los logs del servidor, además de lanzar la aplicación mediante la invocación desde un navegador.

Para replegar o eliminar aplicaciones, también se puede utilizar la consola de administración o mediante línea de comandos. Ej.

```
asadmin undeploy simpleapp
```

I.15.-CONFIGURANDO GLASSFISH (DATASOURCES, JMS, JAAS, WEBSERVICES, ETC.)

En este punto se va enseñar como configurar Glassfish para que proporcione los servicios necesarios a nuestras aplicaciones J2EE. Un servidor de aplicaciones Java EE tiene que implementar un conjunto de tecnologías que deben ser accesibles a todas las aplicaciones desplegadas en el mismo.

I.16.-DATASOURCES

Los Datasources o fuentes de datos permiten obtener conexiones con una BBDD, de manera transparente, sin ocuparse de la gestión de las mismas. Esto proporciona varias ventajas respecto a las conexiones gestionadas directamente desde las aplicaciones.

I.16.1.-SEGURIDAD

La configuración de conexión (incluido usuario y contraseña) no tiene porque estar viajando por 1000 descriptores o ficheros de propiedades de 2000 aplicaciones.

I.16.2.-PRODUCTIVIDAD

Es preferible tener una fuente de datos que gestione 50 conexiones a 4 que gestionen 20.

I.16.3.-CENTRALIZACIÓN

La configuración del acceso a datos se tiene que configurar en un único sitio, no en cada aplicación que acceda a dichos datos.

El único problema que puede suponer la delegación de la configuración y gestión del acceso a datos de una aplicación es que se la hace dependiente de esta configuración externa, por lo que debe tenerse en cuenta en el caso de aplicaciones distribuidas.

Glassfish tiene soporte nativo para los principales proveedores de BBDD del mercado (Oracle, DB2, MySQL, PostgreSQL, Microsoft SQL Server, etc...).

Para configurar un Datasource, primero hay que hacer accesible al servidor el driver JDBC necesario para trabajar con la BBDD que se vaya a utilizar al dominio que estamos utilizando. Para conseguirlo, se copian los jar en la carpeta /lib del dominio correspondiente o en la carpeta glassfish/lib, haciéndolos disponibles para todos los dominios.

Solo hay que tener en cuenta que hay que reiniciar el servidor para que la librería esté disponible.

I.17.-CONFIGURAR UN DATASOURCE O FUENTE DE DATOS

Para hacer esto, se utiliza el árbol de navegación; Recursos > JDBC > Conjunto de Conexiones donde se puede comprobar que hay alguno ya creado por defecto por el servidor. Son conexiones de Derby, la BBDD embebida. Se pulsa el botón de "Nuevo" y se introducen los siguientes datos:

Nombre: "Mipool"

Tipo de recurso: javax.sql.ConnectionPoolDataSource

Proveedor: ORACLE

En la siguiente interfaz se especifican las características del Datasource y al final están las variables que el driver espera.

Para Oracle en concreto, solo hay que completar obligatoriamente las de user, password y la cadena de conexión que se puede repartir entre varios parámetros (protocolo, serverName, etc.) o proporcionando un único y nuevo parámetro denominado URL. Así:

User: jinson

Password: sistemas

URL: jdbc:oracle:thin:@mihost:1521:MISID

Aparece el nuevo DataSource en la lista y, si se pulsa sobre él, además de poder comprobar y editar sus características esta disponible también un botón muy interesante, en español "Sondeo", que permite comprobar si todo está bien configurado.

Al pulsarlo, el servidor advierte que no encuentra la clase de implementación ¿Por qué? Porque no se ha reiniciado el dominio. Así:

```
asadmin stop-domain domain2
```

```
asadmin start-domain --user admin --passwordfile c:\misclavecitas.txt domain2
```

Ahora al autenticar en la consola de Administración y volver a probar la fuente de datos... da otro, error, porque el servidor no puede conectarse con la BBDD que he configurado pero, la clase de implementación ya la ha encontrado.

Por último, hay que hacer visible la fuente de datos a las posibles aplicaciones que lo utilicen otorgándole un nombre JNDI con el que puedan encontrarlo. Para hacer esto se navega hasta la opción Recursos >JDBC > Recursos JDBC y se pulsará sobre la opción de nuevo. Se crea un nuevo nombre JNDI para el Datasource, indicando que Datasource va a atender a dicho nombre, si esta activado y el contexto del mismo (el servidor o clúster donde se va a publicar dicha entrada JNDI) y ya hay fuente de datos. Se utilizará este nombre JNDI para trabajar con Datasource desde JAVA, bien mediante un lookup directo (initialContext.lookup(nombreJNDI)), bien mediante una anotación Resource o

mediante una persistenceUnit a la que se le pasa el nombre JNDI y que utilizaremos con JPA, por ejemplo, con EJBs.

I.17.1.- JMS

La API de JMS (Java Message Service) proporciona un mecanismo a las aplicaciones Java EE para mandarse mensajes unas a las otras. Las aplicaciones JMS no se comunican directamente sino que algunas escriben mensajes a un destino y otras recogen mensajes de dicho destino. Es decir, permite comunicación asíncrona.

En JMS se trabaja con dos tipos de destinos: colas de mensajes (queues) o tópicos (topics) dependiendo del tipo de mensajería utilizada.

Se utilizan las colas de mensajes, o mensajería de tipo punto a punto, en el caso de que se quiera tener un único destinatario para el mensaje. En este caso, el productor del mensaje y el consumidor no tienen dependencias temporales, es decir, el consumidor del mensaje recibiría el mensaje estuviera o no conectado en el momento en el que este fue enviado a la cola de mensajes.

Se utilizan los tópicos, o mensajería de tipo publicación/subscripción, en el caso de que cada mensaje pueda tener cero, uno o muchos destinatarios. En este caso, el productor y el consumidor si tienen dependencias temporales. El consumidor solo recibirá los mensajes publicados en el tópico después de que el se haya suscrito al mismo y necesita estar conectado para seguir recibiendo mensajes.

I.18.-CARACTERÍSTICAS DE GLASSFISH

Una de las características de GlassFish V2. Es que ya no es necesario elegir entre las capacidades de código abierto y las empresariales. A través de GlassFish V2 se puede acceder a las funcionalidades principales que se necesitan para implantaciones escalables y de misión crítica.

- 4. Un excelente nivel de rendimiento.** Esto queda demostrado en el resultado del benchmark SPECjAppServer realizado en Julio de 2007,

en el que GlassFish V2 es el servidor de aplicaciones Java EE 5 de código abierto más rápido de la actualidad.

5. **Clustering.** A través de esta característica permite a las empresas agrupar servidores para lograr un mayor nivel de escalabilidad y replicar datos en memoria para su protección y alta disponibilidad en caso de caídas del sistema.
6. **Administración Centralizada.-** Esta nueva versión permite gestionar clusters de servidores de aplicaciones y despliegues de aplicaciones desde una consola de administración centralizada.
7. **Proyecto Metro.-** Permite la interoperabilidad entre los servicios Web alojados en la tecnología Java y entornos Windows.
8. **ESB Abierto.-** Pensado para brindar una fácil integración de los servicios Web y los recursos empresariales existentes.
9. **Java Business Integration (JBI)-** Esta característica permite proporcionar un enfoque centralizado para suministrar SOA* utilizando servicios Web.
10. **Integración con el Entorno de Desarrollo Integrado NetBeans(TM) IDE.-** Permite a los desarrolladores desplegar aplicaciones SOA diseñando workflows de BPEL (Business Process Execution Language).

I.19.-CREACIÓN DE COLAS Y TÓPICOS JMS EN GLASSFISH

Primero hay que crear Factorías de Conexión o Connection Factories para hacer esto se navega por el árbol de opciones hasta Recursos > Recursos JMS > Fabricas de Conexión

Aquí se van a crear dos factorías una para tópicos y otra para colas.

I.19.1.-FACTORÍA DE COLAS

Nombre JNDI: jms/QueueConnectionFactory

Usuario y Password por defecto.

I.19.2.-FACTORÍA DE TÓPICOS

Nombre JNDI: `jms/TopicConnectionFactory`

Usuario y Password por defecto.

`imqDefaultUsername=guest`

`imqDefaultPassword=guest`

`ClientId=<loquedeseesee>`

Una vez configuradas las factorías, hay que crear los destinos de los que se ha hablado anteriormente. Una vez más, se navega hasta Recursos > Recursos JMS > Recursos de Destino y se crea una cola y un tópico:

I.19.2.1.-Cola

Nombre JNDI: `jms/TestQueue`

Nombre de destino físico: `testQueue`

Tipo de Recurso: `javax.jms.Queue`

I.19.2.2.-Tópico

Nombre JNDI: `jms/TestTopic`

Nombre de destino físico: `testTopic`

Tipo de Recurso: `javax.jms.Topic`

Por último, hay que navegar hasta la configuración del servidor o clúster que va a proporcionar el servicio y configurar el trabajo del mismo con JMS. Así, en Configuraciones > `<loquedeseesee>-config` > Servicio JMS se indicará si el servicio JMS se va a proporcionar en local o de forma remota y donde se añaden los destinos físicos de los que antes se habló. Se dan de alta en Configuraciones > `<loquedeseesee>-config` > Servicio JMS > Destinos Físicos y ahí se crean dos: uno de tipo `java.xml.Queue` y con nombre `testQueue` otro de tipo `java.xml.Topic` y con nombre `testTopic`

I.20.-JAAS

JAAS significa Java Authentication and Authorization Service es una librería de seguridad basada en usuario cuyo objetivo es aumentar la seguridad de las aplicaciones escritas en JAVA. Antes se proporcionaba como una extensión pero, desde la versión 1.4, está integrada con el JRE.

JAAS está basado en Security Realms, “reinos de seguridad” o dominios. Básicamente, un Realm está compuesto por usuarios que pertenecen a grupos a los que se les asigna permisos los cuales le permiten o no hacer cosas.

Glassfish viene configurado con tres realms predefinidos (el certificate-realm, el file-realm y el admin.-realm) que utiliza para controlar el acceso a la consola Web de administración y que no se deben usar para ninguna otra aplicación.

Para verlos, podemos navegar hasta Configuraciones > <loquedesees>-config > Seguridad > Dominios.

Se va a añadir un usuario más al grupo de administradores de Glassfish para que aprender cómo funciona el sistema de dominios, usuarios y grupos. Seleccionamos el admin.-realm, se pulsa el botón de “Administrar Usuarios” y se comprueba que solo está creado el admin, que pertenece al grupo asadmin.

Se crea uno nuevo, con nuestro nombre y contraseña y se indica que también pertenece al grupo asadmin.

Ahora, cuando se reinicie el servidor, será posible la autenticación con el nuevo usuario y contraseña, que además debe tener acceso a todas las opciones que tiene el usuario admin.

Se pueden definir nuevos realms -a parte de los 3 ya creados- de alguno de los tipos que nos proporciona Glassfish: por ejemplo de tipo FileRealm, donde le indicaríamos por parámetro el fichero donde guardaríamos la configuración; o el de tipo JDBC, que nos permite guardar la configuración de usuarios, roles y grupos en una BBDD, siempre que respete una estructura predefinida, por ejemplo, la columna password tiene que tener 32 caracteres, porque espera que

este encriptada mediante el algoritmo MD5 y ya sabéis que las claves generadas por MD5 siempre tienen una longitud fija de 32 posiciones.

También se puede trabajar con realms customizados, como se explicará a continuación.

Primero hay que implementar, al menos dos clases: una que implemente `com.sun.enterprise.security.auth.realm.IASRealm` –el Realm en sí- y otra que implemente `com.sun.appserv.security.AppservPasswordLoginModule` –puesto que tiene que haber una clase que sea la que autentique en el Realm recién creado (y en mas, si así lo deseamos).

Una vez implementadas, hay que copiar las clases bien en la carpeta `/lib/classes` del dominio donde se quiera configurar el Realm personalizado, bien en la carpeta `/lib` si van paquetizadas en un fichero jar.

Además, para que el servidor las tenga en cuenta, hay que navegar hasta Configuraciones > <loquedesea>-config > Configuración JVM > Configuración de ruta y añadir la ruta a las clases o al jar en la casilla de sufijo de ruta de clase o Classpath Suffix.

Por último, dar de alta el nuevo Realm y su configuración, a mano, modificando los ficheros de configuración, algo más sencillo que modificar en varios sitios dentro de la consola de Administración:

En el directorio `/glassfish/<dominioquesea>/config` en el fichero `login.conf`, añadir esta sentencia al final:

```
<nombre de vuestro realm>{<clase que implementa el LoginModule> required};
```

Como se puede ver, un mismo LoginModule puede utilizarse para autenticar a varios Realms.

En el directorio `/glassfish/<dominioquesea>/config`, en el fichero `domain.xml`, añadir el nodo

```
<auth-realm classname="<clase que implementa el Realm>" name="<nombre del Realm>">
```

```
<property name="jaas-context" value="sixservixCustomRealm"/>
```

```
<property name="auth-type" value="<nombre del Realm>" />
```

</auth-realm>

Evidentemente, el valor <nombre del Realm> es el que tiene que devolver el método getJAASContext del Realm y el valor el realm-name en el web.xml, etc.

I.21.-WEBSERVICES

Podemos comprobar el estado de los Webservices que hayamos desplegado en el servidor mediante la opción Servicios Web del árbol de navegación. Para comprobarlo, desplegaremos la aplicación calculatorservice.war 6y, después de hacerlo, comprobaremos como el Webservices que la misma contiene se encuentra disponible en la opción de menú antes mencionada. Mediante el botón de “Prueba” podremos comprobar el comportamiento del servicio Web ante distintas solicitudes con distintos parámetros, también podremos obtener el WSDL para generar código JAVA que pueda invocar al Webservices.

I.21.1.-TUNNING DEL SERVIDOR

Se puede mejorar radicalmente el rendimiento de Glassfish y de las aplicaciones desplegadas en el haciendo tunning de todos los elementos que relacionados con el mismo. En teoría, un proceso de tunning siempre tiene que seguir los mismos pasos:

1. Tunning de la aplicación
2. Tunning del Servidor
3. Tunning de la BBDD
4. Tunning de la JVM
5. Tunning del Sistema Operativo.

Mediante el tunning del servidor, se puede confeccionar una pequeña lista de instrucciones sencillas y buenas prácticas que cualquiera puede implementar.

Para mejorar el tiempo de carga de clases, evitar poner muchos directorios en el CLASSPATH del servidor. Es preferible empaquetar las clases relacionadas en ficheros .jar.

Si una aplicación no contiene ningún EJB es más rápido desplegarla como WAR no como EAR.

Deshabilitar el modo de depuración, cuando se desea depurar una aplicación que se despliega contra Glassfish hay que tener este modo activado (en Configuraciones > <loquedeseesee>-config> > Configuración JVM)

Conceder más memoria a las instancias de servidor. Por defecto, se les otorga un máximo de 512MB a cada instancia y, en producción, se debería subir este valor al menos a 1GB. En Configuraciones > <loquedeseesee>-config> > Configuración JVM > Opciones JVM, cambiar el valor de la variable -Xmx de -Xmx512 a -Xmx1024

Si estamos seguros de que en el servidor no se ejecuta código malicioso y, además, no se trabaja con autenticación por JAAS, deshabilitar el SecurityManager, que gasta recursos para llamar al método doPrivilege y comprueba el fichero server.policy. Esto se consigue eliminando de Configuraciones > <loquedeseesee>-config> > Configuración JVM), el parámetro que contenga -Djava.security.manager

Deshabilitar el auto-despliegue, aunque puede venir muy bien en un entorno de desarrollo en producción, el dominio que comprueba el directorio de auto despliegue puede afectar al rendimiento. Evidentemente, solo hay auto despliegue en los servidores no clusterizados y para desactivarlo, se selecciona la opción Servidores Independientes del árbol de navegación y en la pestaña de opciones avanzadas de cada uno de los servidores, se desactivará la implementación automática.

Compilar una JSP consume tiempo y recursos. Al desplegar una aplicación, seleccionar la opción de precompilación de JSPs. Cuando se utiliza, se compilan todas las JSPs y solo se despliegan las clases resultantes no las JSP en sí.

Deshabilitar además la recarga dinámica de JSPs esto se consigue modificando el fichero <instancia>/config/default-web.xml en la configuración del servlet jsp (también se especifica que el texto generado lo haga como un array de caracteres, para mejorar el rendimiento).

```
<servlet>
<servlet-name>jsp</servlet-name>
<servlet-class>org.apache.jasper.servlet.JspServlet</servlet-class>
<init-param>
<param-name>development</param-name>
<param-value>>false</param-value>
</init-param>
<init-param>
<param-name>xpoweredBy</param-name>
<param-value>>true</param-value>
</init-param>
<init-param>
<param-name>genStrAsCharArray</param-name>
<param-value>>true</param-value>
</init-param>
<load-on-startup>3</load-on-startup>
</servlet>
```

Cambiar la configuración de la replicación de sesiones cuando se trabaja en cluster. La configuración por defecto es la de web-method que replica todos los datos cada vez que hay una petición HTTP, se puede hacer que la replicación se haga por tiempo, en aquellos sistemas donde es más importante el rendimiento que la estabilidad.

I.21.2.-CLUSTERING

Un cluster de servidores es un conjunto de instancias de servidores, implementadas físicamente en la misma máquina o en varias que, a nivel lógico,

funcionan y se configuran como un único servidor. El clustering se utiliza por dos motivos principales:

Conseguir alta disponibilidad (si una maquina deja de estar en servicio, el resto siguen funcionando)

Mejora en el rendimiento (repartir las distintas peticiones de servicio entre los servidores que componen el cluster).

Para que el clustering sea transparente para los usuarios se suele utilizar conjuntamente con un balanceador de carga que actúa como pasarela que distribuye el trabajo entre los distintos servidores.

A continuación se explica la configuración de clustering con Glassfish y de cómo hacer balanceo de carga con un servidor Apache.

I.21.3.-CONFIGURANDO EL CLUSTER EN GLASSFISH

- 1.- Se crea un agente de Nodo (no se puede crear desde la consola de administración, solo desde la CLI con el comando `asadmin create-node-agent <nombre de nodo>` .Si no se especifica el nombre de nodo se utiliza el hostname de la maquina. Si se está creando el node agent en un servidor distinto se debe especificar el host y nombre de la maquina con los parámetros `--host <hostname> --port <portname>`
- 2.- Se iniciará el agente de nodo con el comando `asadmin start-node-agent <nodeagentname> --user <user> --passwordfile <passwordfile>`. Se solicitará la contraseña maestra.
- 3.- Si todo va bien, en la opción del menú “Agentes de nodo” se verá el agente de nodo creado y ejecutándose.
- 4.- Se crea un Cluster, seleccionando la opción de Clusters en el árbol de navegación: se le da un nombre, se aplica la configuración por defecto y se añaden dos nodos al cluster, seleccionando los agentes de nodo de las maquinas donde se desea que se ejecuten las instancias. Como solo hay un agente de nodo, las dos instancias correrán en la misma máquina. En el

taller, se denomina al cluster MICLUSTER y a las instancias, instancia1 e instancia2.

Se pulsa OK y el cluster y las dos instancias se crearán. Esto tardara más o menos dependiendo de la maquina.

En la siguiente página, se elegirá MICLUSTER y se iniciará. Ahora las dos instancias ya están ejecutándose.

- 5.- Para comprobar que el clúster funciona, se utilizará una aplicación de prueba, muy sencilla que ya trae el Glassfish como prueba. Se navegará hasta la página de aplicaciones empresariales y seleccionaremos la aplicación.

</glassfish/samples/quickstart/clusterjsp/clusterjsp.ear>.

Se subirá normalmente pero hay que tener especial cuidado con dos cosas; una que esté disponible activando el checkbox correspondiente, y dos, el destino, que ya no será el servidor DAS en si sino el clúster que se ha creado.

- 6.- Por el puerto 8080 continua sirviendo el DAS, así que, para comprobar porque puertos sirve cada una de las instancias hay que visitar de nuevo la opción de clusters del árbol de navegación, dentro de MICLUSTER al pulsar sobre cada una de sus instancias se puede comprobar porque puertos http están sirviendo, normalmente por el 38080 y el 38081. Se puede acceder a la nueva aplicación desplegada en el cluster a través de <http://localhost:38080/clusterjsp> y <http://localhost:38081/clusterjsp>.

- 7.- Ahora se prueba como funciona el cluster, aunque uno de los nodos del mismo se caiga y como se vuelve a replicar la sesión del usuario cuando volvemos a arrancar el nodo con:

```
asadmin stop-instance --user admin --passwordfile c:\misclavecitas.txt
instancia2
```

```
asadmin start-instance --user admin --passwordfile c:\misclavecitas.txt instancia2.
```

Cuando se despliega una aplicación en un cluster, se despliega del mismo modo en todas sus instancias pero, podemos engañar al servidor tocando directamente

los ficheros físicos de cada una de las instancias. Si se despliega la aplicación hello.war, que solo consta de una JSP, podemos comprobar cómo se puede modificar dicha JSP en una instancia y cada una servirá una JSP distinta.

I.21.3.1.-BALANCEO DE CARGA

El cluster no muestra toda su potencia hasta que no lo complementamos con un balanceador de carga. La versión v2 de Glassfish cuenta, según sus creadores, con un estupendo soporte de balanceo de carga. La realidad es que el único balanceador de carga soportado es el servidor SUN Java System Web Server.

I.22.-MAIL

Al igual que con las fuentes de datos, se pueden configurar sesiones de trabajo con servidores de correo desde el servidor de aplicaciones, para que la configuración de acceso a cuentas de correo no tenga que distribuirse a las aplicaciones cliente. Esto es fácil con la API JavaMail, parte de especificación Java EE 5.

Para configurar una Sesión de JavaMail, se utiliza el árbol de navegación hasta Recursos > Sesiones JavaMail donde, mediante el botón "Nuevo", se pueden configurar nuevas sesiones.

Si las aplicaciones cliente solo van a enviar mensajes (lo más habitual) no es necesario rellenar siquiera la parte de guardado, solo el de transporte.

En cualquier caso, se enumeran las clases de SUN que hay que indicar en el caso de que queramos cambiar el protocolo de envío o guardado:

- 1.- IMAP: com.sun.mail.imap.IMAPStore
- 2.- IMAPS: com.sun.mail.imap.IMAPSSLStore
- 3.- POP3: com.sun.mail.pop3.POP3Store
- 4.- POP3S: com.sun.mail.ipop3.POP3SSLStore
- 5.- SMTP: com.sun.mail.smtp.SMTPTransport
- 6.- SMTPS: com.sun.mail.smtp.SMTPSSLTransport

Referencia bibliográfica:

Tomado de la página Web: David@sixservis.com

CAPÍTULO II

La información que contiene este capítulo, es necesaria para la instalación de glassfish, porque nos da referencias del tipo de software y hardware que necesitamos, así como las principales características, ventajas y desventajas de uso del servidor glassfish que ya esta disponible para su implementación en distintas plataformas.

II.1.-REQUISITOS DE SISTEMA Y PLATAFORMAS COMPATIBLES

Además de los requisitos que aparecen en Requisitos de hardware y software, compruebe que su sistema cumpla los requisitos que se indican a continuación para ejecutar.

- Requisitos de Software
- Plataformas compatibles
- Requisitos del host del servidor
- Requisitos del host de administración
- Requisitos del host del cliente

II.1.1.- REQUISITOS DE SOFTWARE

- Necesitamos tener instalado Java 1.5 o superior
- Necesitamos tener instalado Netbeans 6.5

II.1.2.- PLATAFORMAS COMPATIBLES

- **Solaris (SPARC)** – Solaris 8 MU7, Solaris 9 MU7 y Solaris 10 RR.

- **Solaris (x86)** – Solaris 9 MU7 y Solaris 10 RR.
- **RedHat Enterprise Linux** - 2.1 U5 (sólo se admite el sistema de archivos ext2, no ext3), 3.0 U4 (se admiten ext2 y ext3).
- **Microsoft Windows** – Microsoft Windows 2000 Advanced Server Service Pack 4 y Microsoft Windows 2003 Enterprise Edition.

II.1.3.- REQUISITOS DEL HOST DEL SERVIDOR

- **Memoria mínima:** 320 MB por nodo.
- **Cantidad mínima de espacio libre en disco:** 70 MB para binarios por host. Además, se necesita espacio de disco para los dispositivos de datos; 512 MB para una instalación de prueba por cada nodo.
- **Memoria recomendada:** 512 MB por nodo.
- **Espacio libre en disco recomendado:** 70 MB para binarios por host. Además, se necesita espacio de disco para los dispositivos de datos; 1200 MB para una instalación de prueba por cada nodo:

II.1.4.- REQUISITOS DEL HOST DE ADMINISTRACIÓN

- **Memoria mínima:** 128 MB.
- **Cantidad mínima de espacio libre en disco:** 70 MB para binarios por nodo.

II.1.5.- REQUISITOS DEL HOST DEL CLIENTE

Memoria mínima: 120 MB.

Cantidad mínima de espacio libre en disco: 20 MB

II.1.6.- OTROS REQUISITOS

Los siguientes requisitos adicionales se deben cumplir para poder instalar el software de Sun Java System Application Server.

II.1.7.- ESPACIO LIBRE

El directorio temporal debe tener un mínimo de 35 MB libres para la instalación de Sun Java System Application Server y 250 MB para la de SDK.

II.1.8.- USO DEL PROGRAMA DE DESINSTALACIÓN

Si necesita eliminar Application Server del sistema, es fundamental que utilice el programa de desinstalación que se incluye con el software. Si intenta utilizar cualquier otro método, surgirán problemas cuando intente reiniciar la misma versión o cuando desee instalar una versión nueva.

II.1.9.- PUERTOS LIBRES

Debe disponer de 7 puertos libres. El programa de instalación detecta automáticamente los puertos e indica aquellos que no se están utilizando en ese momento para el dominio predeterminado. Por defecto, los puertos predeterminados desde el inicio son:

8080 para HTTP

8181 para HTTPS

3700 para IIOP

3820 para IIOP/SSL

3920 para IIOP/SSL con Autenticación Mutua

4848 (HTTP) para Administration Server

El puerto 8686 (RMI) para los clientes JMX

II.1.10.-INICIO DE SERVIDORES PREVIAMENTE INSTALADOS (UNIX)

A menos que desee sustituir el servidor instalado con anterioridad, deberá iniciarlo antes de comenzar el proceso de instalación de Sun Java System Application Server 9.1. Esto permite que el programa de instalación detecte los puertos que están en uso y no los asigne para otros usos.

II.1.11.-CIERRE DEL SERVIDOR DE SEGURIDAD

debe detener cualquier tipo de software de servidor de seguridad antes de instalar Sun Java System Application Server porque algunos servidores de seguridad desactivan todos los puertos de forma predeterminada. El programa de instalación debe determinar con precisión qué puertos están disponibles.

Referencia bibliográfica:

Tomado de la página Web:

<http://docs.sun.com/app/docs/doc/820-1800/6ndeqhogd?l=es&a=view>

II.2.-CONOCIMIENTOS BÁSICOS DE LAS HERRAMIENTAS DE GLASSFISH

Para empezar a manipular la configuración ya sea de Glassfish o de cualquier servidor, es conveniente tener un conocimiento básico de sus herramientas a fin de poder interactuar de la mejor manera con nuestro servidor. A continuación se presenta las principales herramientas de configuración de Glassfish.

II.2.1.- ACCESO A LA PLATAFORMA GLASSFISH

Para acceder al tablero de control del servidor vaya a la url <http://localhost:4848>.

Las credenciales de autenticación son:

Usuario: admin

Contraseña: adminadmin



Figura 2-1. Pagina de Inicio Glassfish

II.2.2.- CONFIGURACIÓN DE LA BASE DE DATOS

- 1.- Descargue el archivo de instalación de MySQL
- 2.- Ejecute el archivo descargado y siga las instrucciones del wizard.
- 3.- Cree una base de datos según los datos expuestos en el archivo mysql-ds.xml.
- 4.- Ejecute el script scriptDefault.sql que se encuentra en la ruta QualdevSecurityMDA.EJB\test\scripts

II.2.3.- CONFIGURACIÓN DE ECLIPSE

Para utilizar el proyecto QualdevSecurityMDA.Generation descargue el ambiente de generation del repositorio de qualdev. El eclipse que se encuentra con nombre eclipse-mds, contiene toda la configuración necesaria para realizar las transformaciones de los modelos y la generación del código.

Descargar el archivo eclipse-mds.zip del repositorio.

II.2.4.- CONFIGURACIÓN DE LA APLICACIÓN

Para ejecutar la aplicación en el servidor siga las siguientes instrucciones:

- 1.- Levante el servidor de base de datos de MySql habiendo ejecutado el script expuesto en la sesión Configuración de la Base de Datos

- 2.- En la carpeta QualdevSecurityMDA.WEB encontrará el archivo ant build.xml. Ejecute la opción package.ejb y package.webapp. I
- 3.- En la carpeta deploy encontrará el archivo .jar y .war de la aplicación.
- 4.- Acceda al tablero de control del servidor de aplicaciones Glassfish

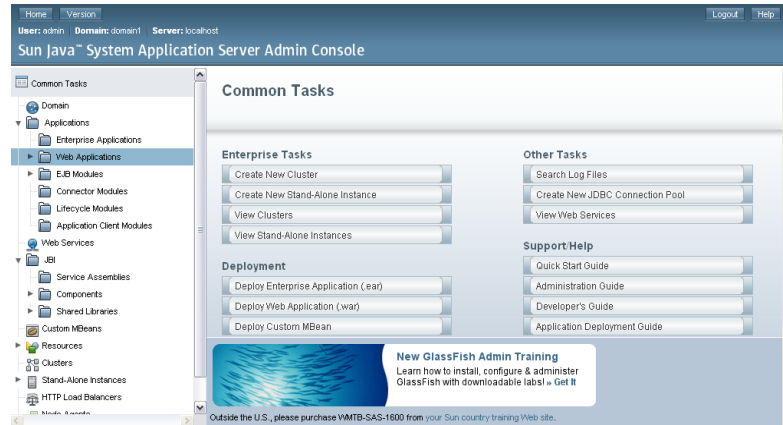


Figura 2-2. Configuración de la Aplicación (Tareas comunes)

- 5.- En la opción EJB Modules del tablero de control de Glassfish puede hacer deploy del qualdevSecurityEJB.jar.

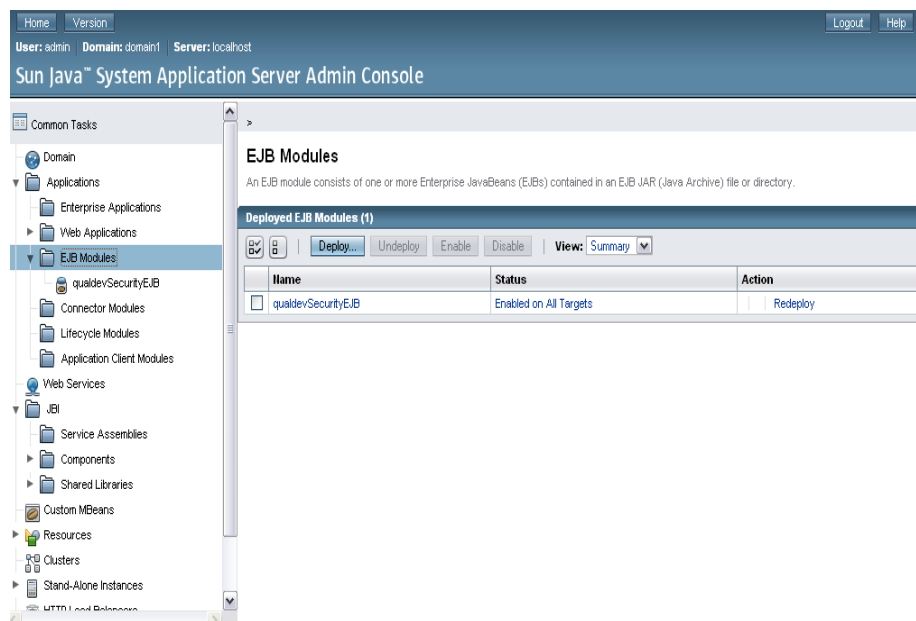


Figura 2-3. Configuración del Modulo EJB

6.- En la opción WebApplications del tablero de control de Glassfish haga deploy de qualdevSecurityWEB.war. Ejecute la opción Launch enseguida a este para lanzar la aplicación.

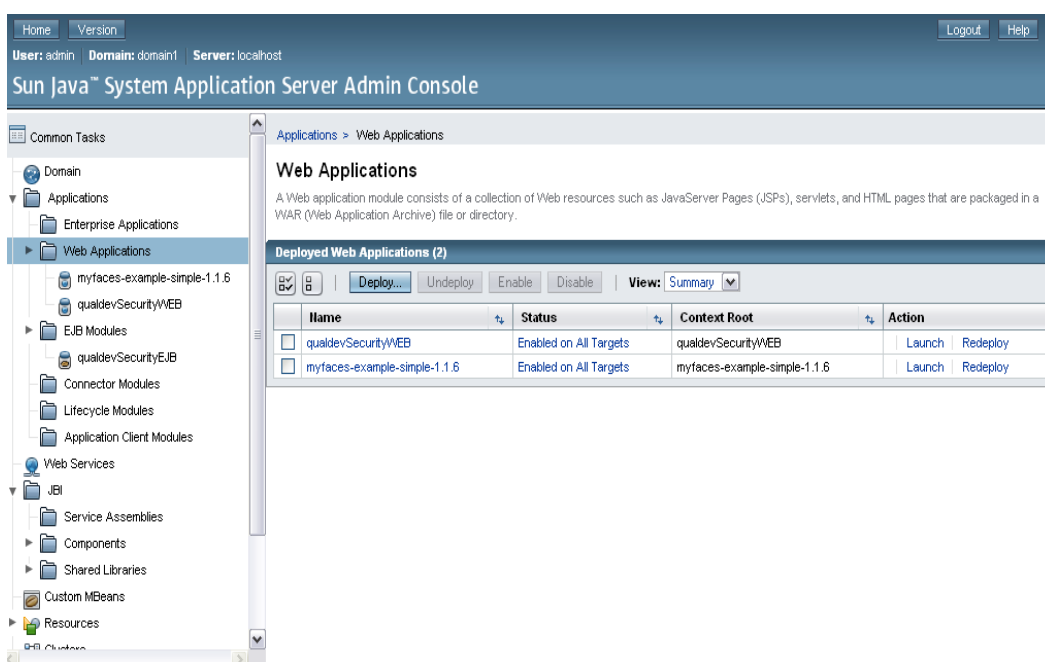


Figura 2-4. Configuración de la Aplicación Web

II.2.5.- INDICAR LA LISTA DE LAS LIBRERÍAS DEL DRIVER CONNECTOR/J DE MYSQL

Para que nuestras aplicaciones Java EE que están alojadas en un Servidor Java EE puedan acceder a las librerías del Driver Connector/J de MySQL, tenemos que seguir los siguientes pasos

En el árbol de navegación, seleccionamos Application Server y seguidamente seleccionamos las pestañas JVM Settings | Path Settings

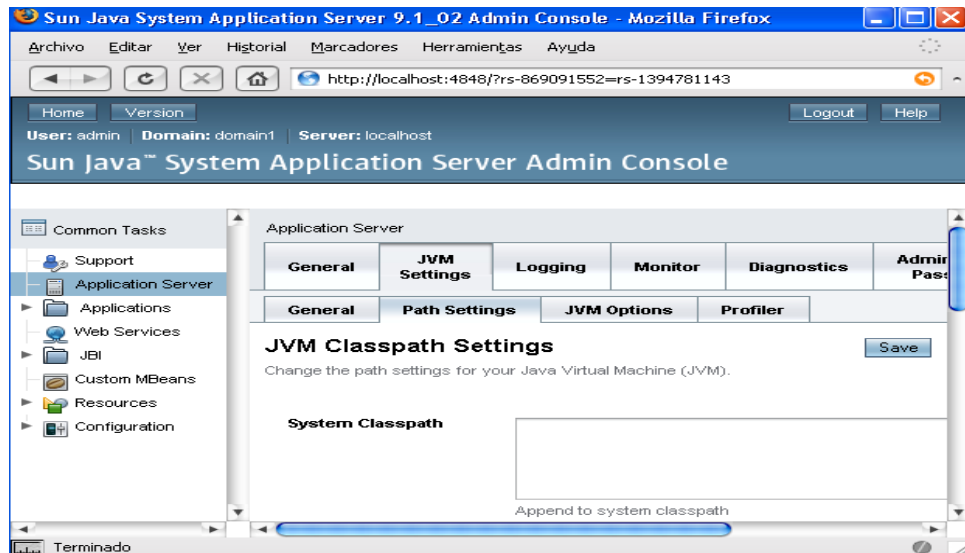


Figura 2-5. Configuración del servidor de Aplicaciones

Nos desplazamos hasta el campo de texto Classpath Prefix e indicamos la localización en el disco duro del Servidor del fichero jar mysql-connector-java-5.1.5-bin.jar.

Si tenemos instalada la versión 6.1 de NetBeans este fichero jar lo encontraremos en la siguiente estructura de directorios C:\Archivos de programa\NetBeans 6.1\ide9\modules\ext

Si no tenemos instalada la versión 6.1 de NetBeans nos podemos descargar el driver siguiendo el siguiente vínculo [Descarga e instalación del Driver MySQL Connector/J](#)

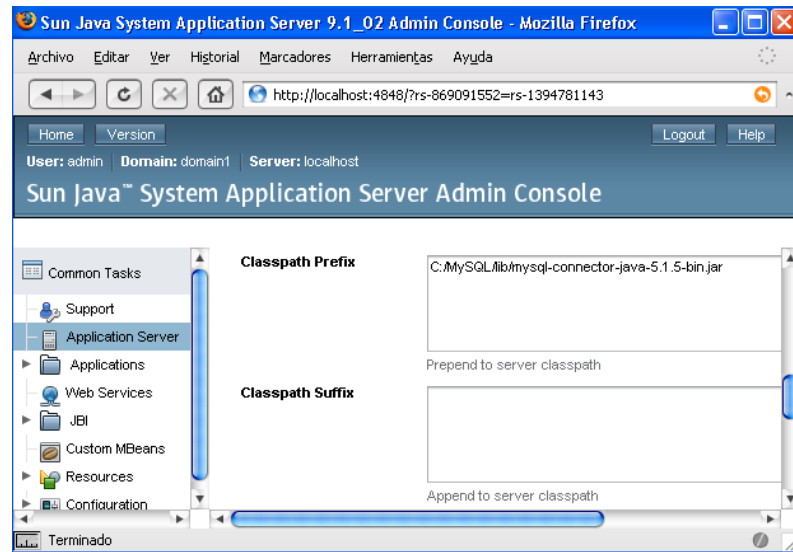



Figura 2-5. Configuración de la Máquina Virtual de Java

Finalmente hacemos clic sobre el botón **Save** y bajamos y subimos GlassFish para que los cambios tengan efecto. Si todo esta bien obtendremos el siguiente

dialogo  **New values successfully saved.**

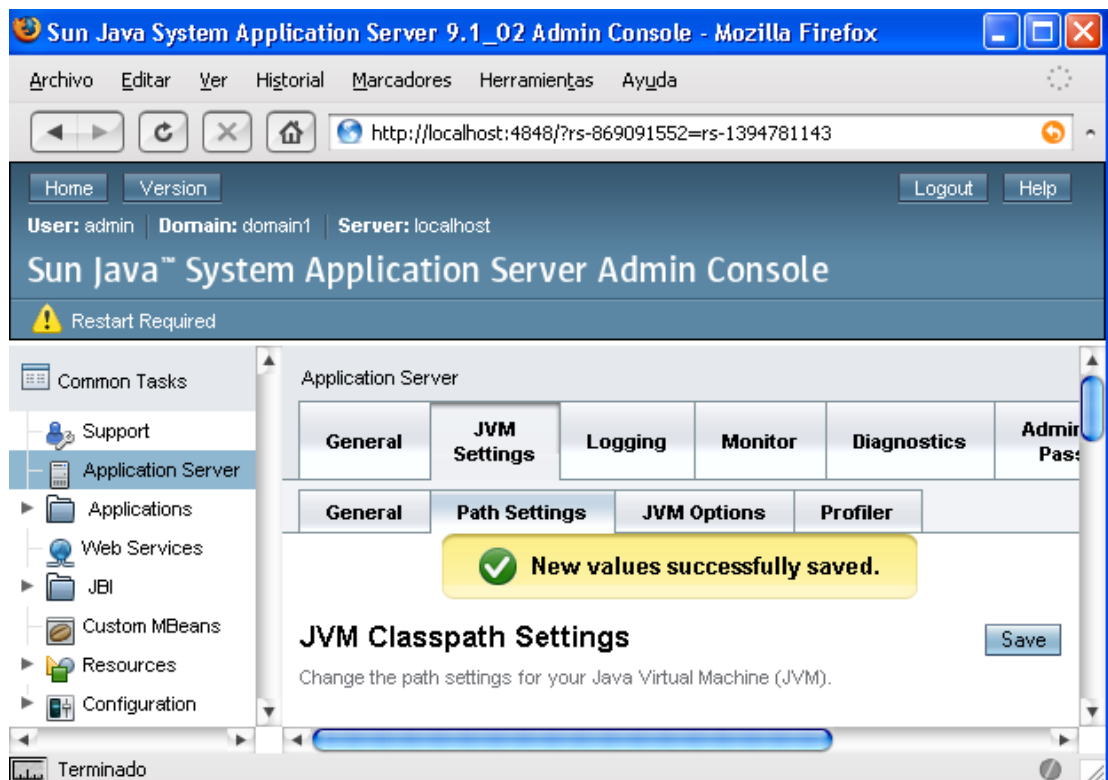


Figura 2-7. Comprobación de la configuración

1. Configurar un recurso JDBC que enlaza con un Pool de conexiones
2. Para guardar y acceder a datos la mayoría de las aplicaciones utilizan Bases de datos relacionales. Las aplicaciones Java EE acceden a las Bases de Datos relacionales a través de la API JDBC.
3. Para que una aplicación pueda acceder a una Base de Datos es obligatorio tener una conexión a la Base de Datos.

Estos son los pasos que una aplicación realiza cuando conecta a una Base de Datos

1. La aplicación obtiene un recurso JDBC (data source) que está asociado a la Base de Datos realizando una llamada a través de la API JNDI.
2. Una vez se facilita el nombre del directorio JNDI, éste localiza el recurso JDBC.
3. Cada recurso JDBC especifica un Pool de conexiones
4. A través del recurso JDBC, la aplicación obtiene una conexión a la Base de Datos de forma transparente,
5. El Servidor de aplicaciones obtiene una referencia de una conexión física del Pool de conexiones que corresponde a su Base de Datos correspondiente

El Pool define atributos de conexión cómo:

1. El nombre de la Base de Datos,
2. La localización y el puerto de la Base de Datos
3. El login y la contraseña para acceder a la Base de Datos
4. Una vez que ya hemos obtenido la conexión a la Base de Datos, la aplicación puede leer, modificar y añadir datos a la Base de Datos
5. La aplicación accede a la Base de Datos realizando llamadas a la API JDBC
6. El driver JDBC traduce las llamadas en el protocolo del Motor de Base de Datos específico

7. Cuando se ha finalizado el acceso a la Base de Datos, la aplicación cierra la conexión
8. El Servidor de aplicaciones entonces retorna la conexión al Pool de conexiones

Y esta conexión física esta lista para ser utilizada por la misma aplicación u otras aplicaciones

Para que una aplicación pueda obtener un recurso JDBC y posteriormente una conexión a una Base de Datos MySQL, primeramente se tiene que configurar Glassfish siguiendo los siguientes pasos:

II.2.6.- CREAR UN NUEVO RECURSO JDBC (DATA SOURCE)

Para crear un nuevo recurso JDBC tenemos que seguir los siguientes pasos:

En el árbol de navegación, seleccionamos Resources | JDBC | JDBC Resources y seguidamente hacemos clic sobre el botón **New...** de la tabla Resources



Figura 2-8. Configuración de los Recursos de JDBC

En la página New JDBC Resource introducimos los siguientes datos:

JNDI Name: jdbc/ABM

Pool Name: mySQLPoolABM

Tal como se muestra en la siguiente figura



Figura 2-9. Habilitación de los Recursos del JDBC

Al hacer clic sobre el botón podemos observar que el nuevo recurso JDBC jdbc/ABM ha sido creado.

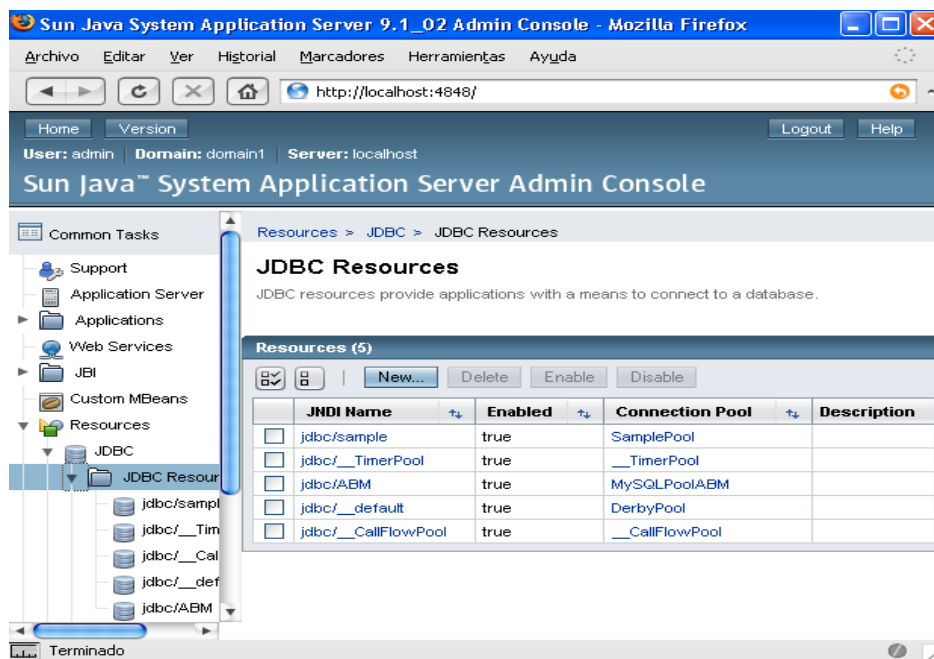


Figura 2-10. Crear el JDBC

II.2.7.- CREAR UN NUEVO POOL DE CONEXIONES JDBC

Para crear un nuevo Pool de conexiones JDBC tenemos que seguir los siguientes pasos

En el árbol de navegación, seleccionamos Resources | JDBC | Connection Pool y seguidamente hacemos clic sobre el botón de la tabla Pools.

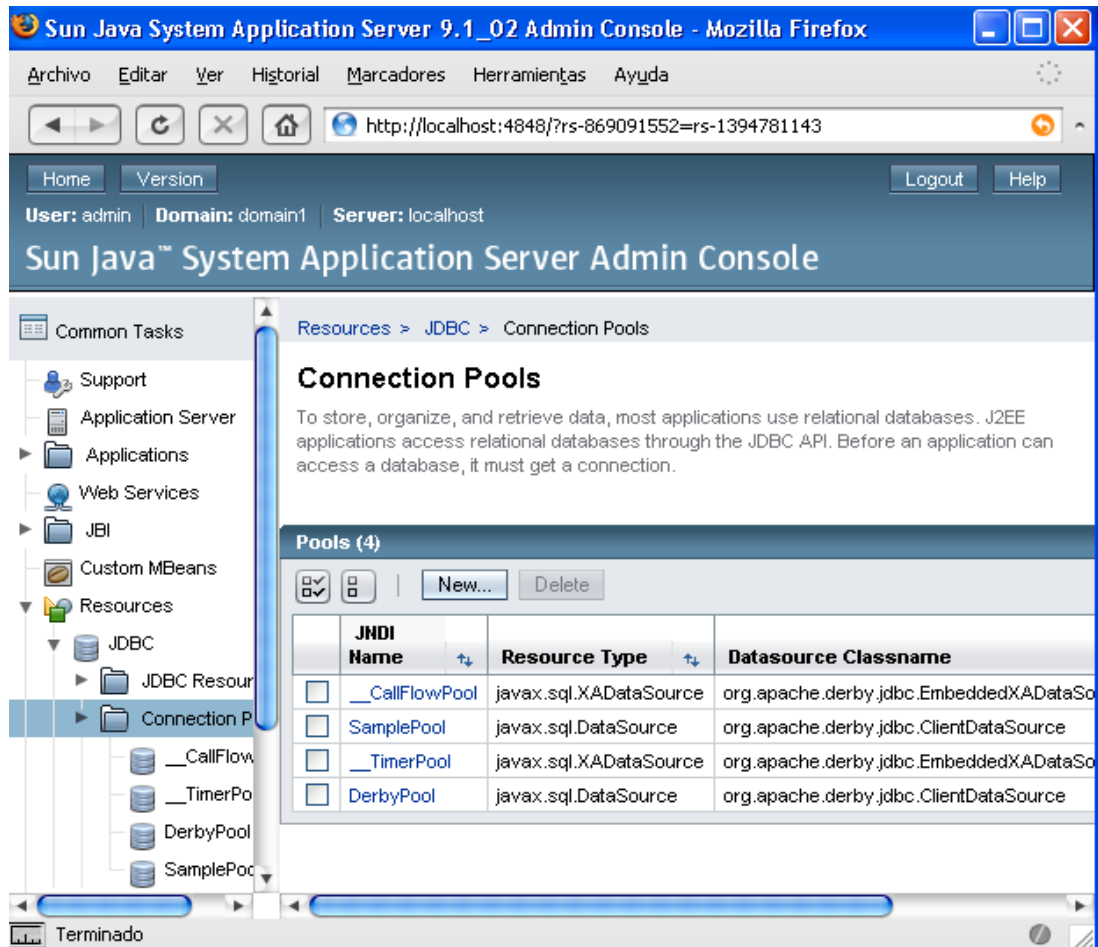


Figura 2-11. Crear un Nuevo Pool de Conexiones

En el apartado New JDBC Connection Pool (Step 1 of 2) introducimos los siguientes datos

Name: MySQLPoolABM

Resource Type: javax.sql.DataSource

Database Vendor: MySQL

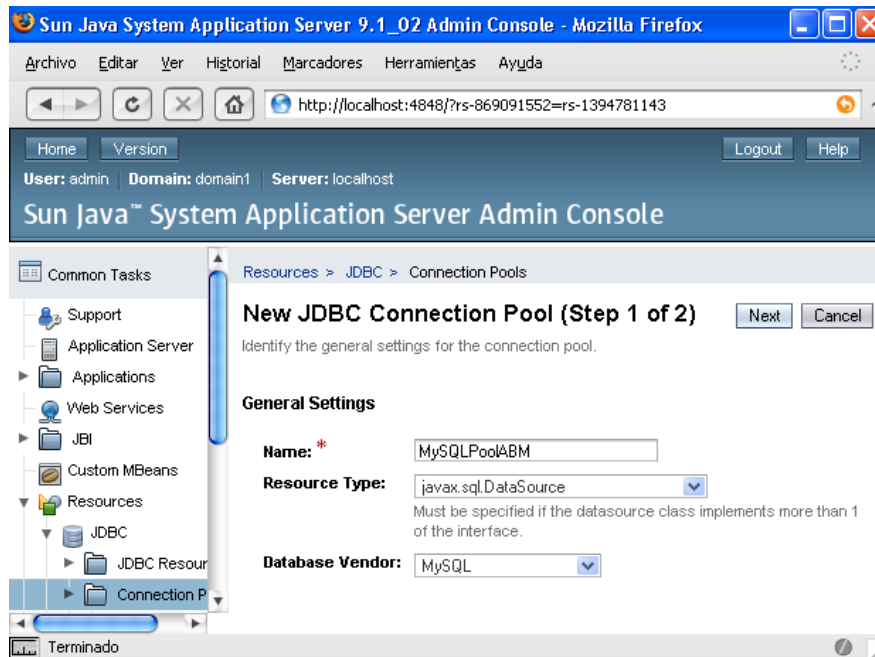


Figura 2-12. Configuración del pool de conexiones (paso 1)

Al hacer clic sobre el botón **Next** obtenemos la página New JDBC Connection Pool (Step 2 of 2). En el campo de texto Datasource Classname: observamos que el asistente ha escrito `com.mysql.jdbc.jdbc2.optional.MysqlDataSource`

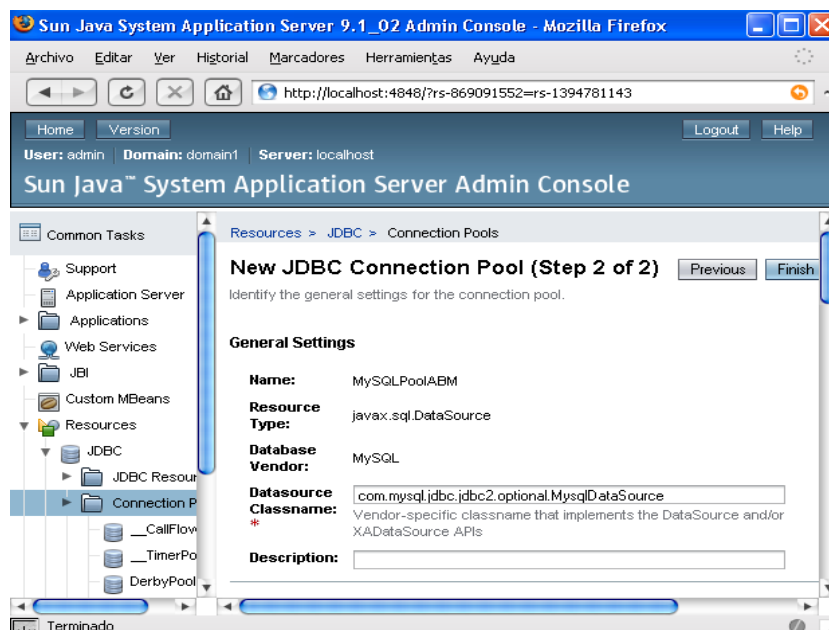


Figura 2-12. Configuración del pool de conexiones (Paso 2)

En el apartado Pool Settings mantenemos todos los valores que hay escritos por defecto.

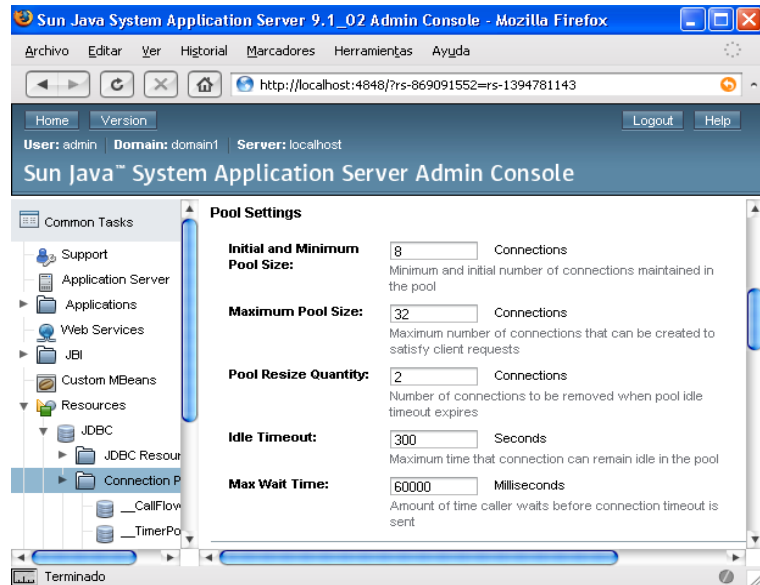


Figura 2-14. Configuración de del pool de conexiones

En el apartado Connection Validation mantenemos todos los valores que hay escritos por defecto.

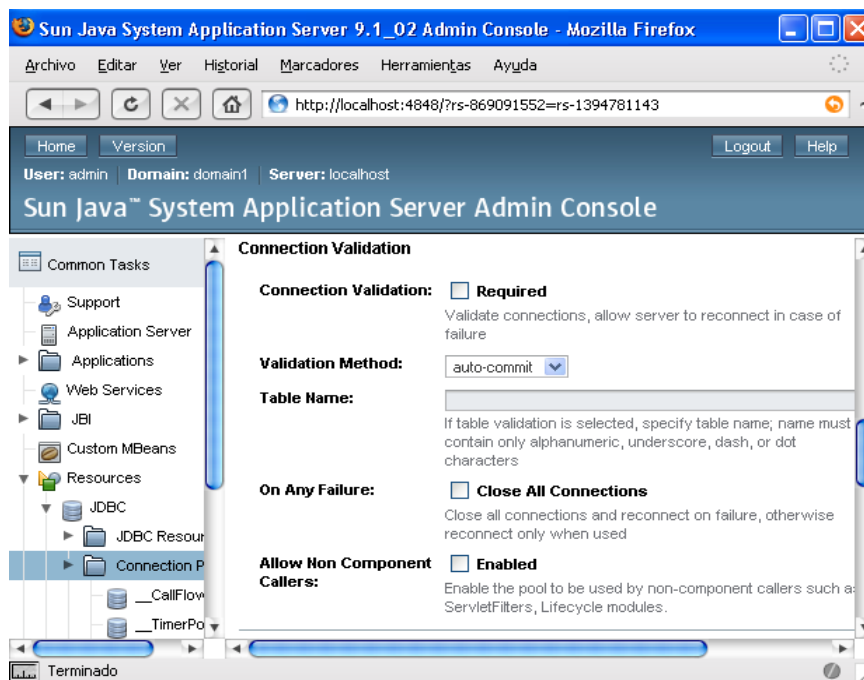


Figura 2-15. Validación del Pool de Conexiones

En el apartado Transaction mantenemos todos los valores que hay escritos por defecto.

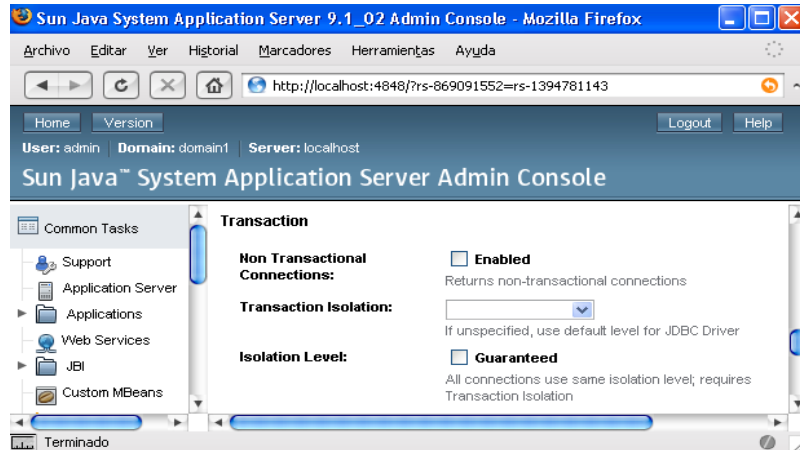


Figura 2-16. Transacción del Pool de Conexiones

En la tabla Additional Properties introducimos los siguientes valores

databaseName: altasbajasmods

password: javadude

portNumber: 3306

serverName: localhost

user: javauser

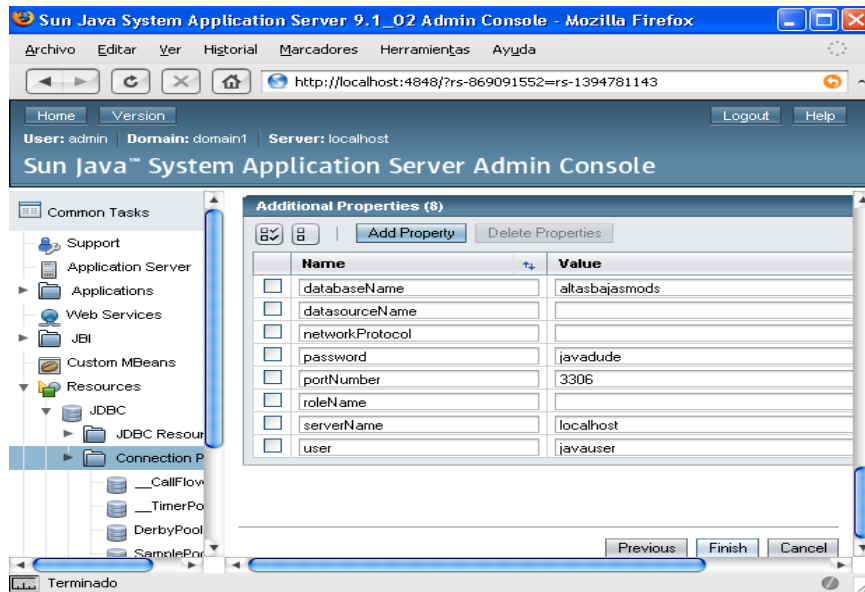


Figura 2-17. Añadir propiedades al Pool de Conexiones

Seguidamente hacemos clic sobre el botón **Finish**. Como podemos observar en la tabla Pools se ha añadido el nuevo Pool de conexiones MySQLPoolABM. Ahora hacemos clic sobre su vínculo.

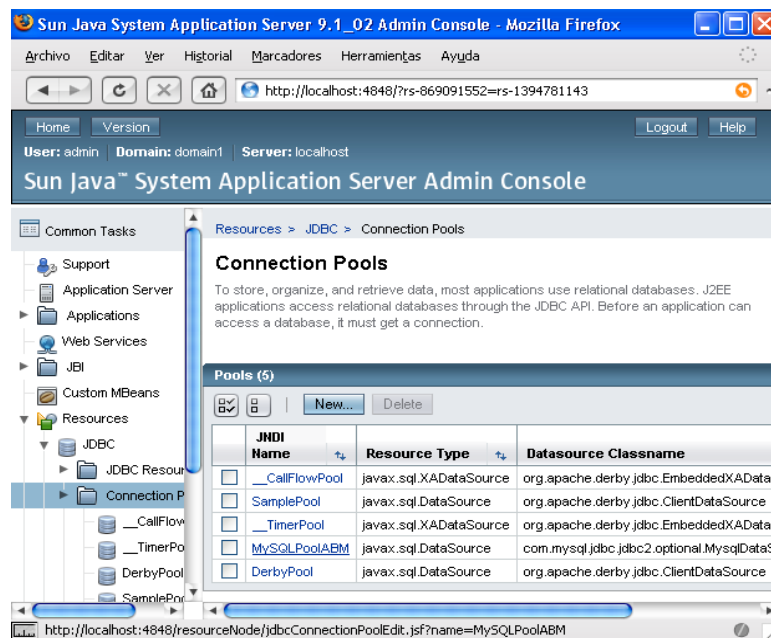



Figura 2-18. Tabla del Pool Conexiones

Para comprobar que la configuración del Pool de conexiones ha sido correcta hacemos clic sobre el botón 

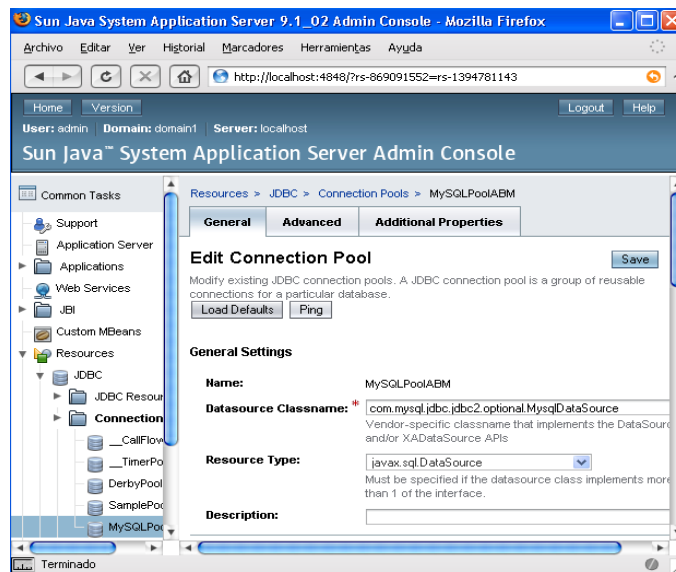
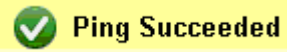


Figura 2-19. Editar el Pool de Conexiones

Si todo ha ido bien, nos aparece el mensaje



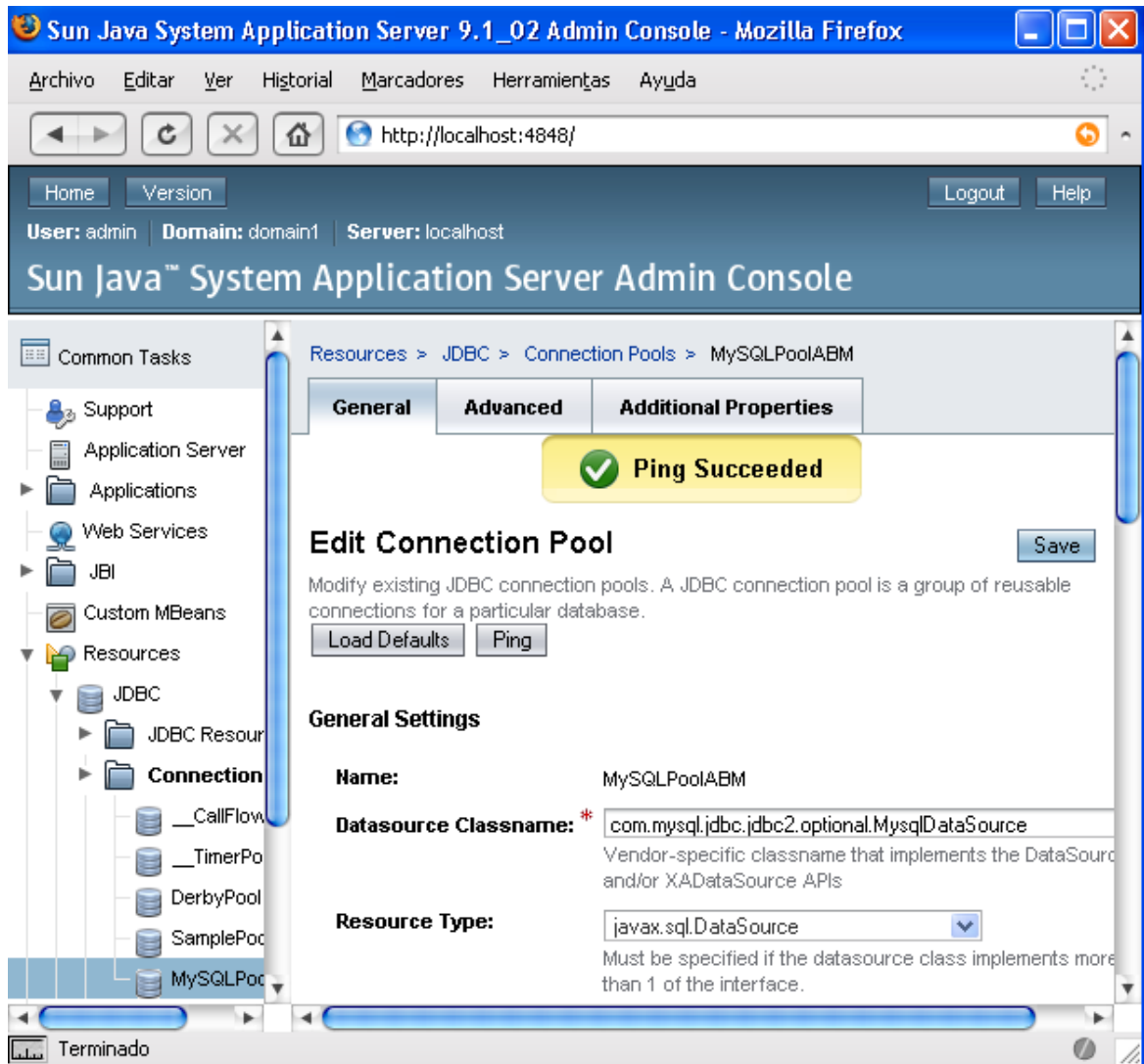


Figura 2-20. Verificación de la edición del Pool de Conexiones

Referencia bibliografica:

Tomado de la página Web:

<http://www.infosintesis.net/apensintesis/glassfish/mapa/v2/crearPoolConexiones.html>

<http://www.infosintesis.net/apensintesis/glassfish/mapa/v2/configurarRecursosJDBCpoolConexiones.html>

http://weblogs.javahispano.org/lasterra/es/entry/glassfish_clustering

II.3.-VENTAJAS Y DESVENTAJAS DE GLASSFISH

Las siguientes descripciones explican mejor algo de lo que proporciona el poder de Glassfish, en la relación al cuadro antes mencionado.

Características	JBoss 4.2	Geronimo 2	Tomcat 6	GlassFish V2
Java EE 5 compliance	Partial	Yes	No	Yes
EJB 3.0 capable	Yes	Yes	Available	Yes
JSP 2.1 and 2.5 capable	Yes	Yes	Yes	Yes
JavaServer Faces 1.2 support	Yes	Yes	Available	Yes
Custom plug-in support	Yes	Yes	No	?
Business-rules engine support	Available	Available	Available	Available
Hibernate 3.x support	Yes	Available	Available	Yes, based on below description
JBoss Seam support	Yes	Yes	Available	Yes
Clustering support	Yes	Yes	Partial	Yes
Eclipse IDE connector support	Yes	Yes	Yes	Yes

Tabla 2-4. Ventajas y desventajas de Glassfish

La ayuda del clustering de GlassFish y la gerencia del cluster fuera del rectángulo, un cluster se pueden configurar de CLI y de la consola de administración.

La consola de administración de GlassFish permite que usted configure su balanceador de la carga, por ejemplo usted puede configurar un web server de Sun Java que trabaje como balanceador de la carga para agregar o quitar un caso de su lista de servidores, manualmente o automáticamente si un nuevo nodo ensambló el cluster o quitado del cluster

GlassFish permite que usted maneje los recursos para el cluster entero inmediatamente en vez de aplicarlos para cada caso, por ejemplo usted puede desplegar una aplicación del Web en un cluster de 10 instancias en vez de desplegarlo por separado para cada caso.

GlassFish tiene una documentación de la gama muy amplia de Sun Microsystems (para libre) y de la comunidad de GlassFish.

La instalación de GlassFish es tan fácil como ejecutando 2 comandos.

Desplegar aplicaciones en GlassFish o aún un cluster entero de los casos del glassfish es apenas 2 clicks.

La calidad de los componentes de GlassFish está fuera de cualquier pregunta, metro utiliza bien los nuevos estándares de WS, esencial de Toplink de las aplicaciones de la ayuda de EJB, MQ Server es Sun Open Sourced MQ, etc.

GlassFish tiene interoperabilidad muy buena con algunos otros proyectos de la fuente como, OpenESB y OpenSSO lo que permite que usted tenga lo que usted necesita, de esta manera usted obtendrá su aplicación de J2EE sin mirar ninguna configuración adicional.

El funcionamiento es ciertamente algo que cada uno debe tener en mente antes de considerar otras características,

GlassFish tiene conectores para eclipse y Netbeans, aunque otros servidores mencionados tienen un conector en Netbeans y eclipse.

La ayuda de motor de la regla de negocio está disponible de la integración del proyecto de OpenESB.

GlassFish tiene un centro de la actualización, que permite que usted ponga al día su servidor de la aplicación de un depósito alejado.

GlassFish se ejecuta en todas las plataformas mencionadas, de Windows a AIX (actualización 1 de Glassfish V2 se ejecuta en AIX) y no hay restricción para que usted la ejecute en su plataforma de la opción.

CAPÍTULO III

En esta instancia del proyecto, aprenderemos a instalar dos versiones del sistema operativo linux y hemos tomado como referencia las versiones redhat 9.0 y ubuntu 8.1, además aprenderemos a instalar el servidor de aplicaciones GLASSFISH, así como también, instalaremos REDHAT y UBUNTU. Se realizara un ejemplo de cómo cargar y poner en funcionamiento una aplicación en dicho servidor.

III.1.-INSTALACIÓN DE RED HAT LINUX

Este apartado explica cómo realizar una instalación personalizada de Red Hat Linux desde el CD-ROM, utilizando el programa de instalación gráfico basado en el ratón.

Interfaz gráfica de usuario del programa de instalación

Use el ratón para navegar por las pantallas, pulse los botones o rellene los campos de texto. Puede también utilizar las teclas [Tab] e [Intro] para navegar.

Inicio del programa de instalación

Para iniciar la instalación, tiene que iniciar el programa de instalación. Asegúrese de disponer de todos los medios (CDs o similar) necesarios para la instalación.

Arranque del programa de instalación

Se puede arrancar el programa de instalación usando cualquiera de los siguientes medios (en función del medio compatible con su sistema).

1. **CD-ROM de arranque.**- Si su ordenador soporta una unidad de CD-ROM de arranque y quiere realizar una instalación de red o disco duro.
2. **Disquete de arranque.**-Puede utilizar esta opción si su ordenador no soporta CD ROM para el arranque y quiere efectuar una instalación desde un CD-ROM, red o disco duro.

Introduzca el disquete de arranque en la primera unidad de disco y vuelva a arrancar (o arranque usando el CD-ROM, si su ordenador admite arrancar desde la unidad de CD-ROM). La configuración de su BIOS puede que se tenga que cambiar para que sea posible arrancar desde un disco o un CD-ROM.

III.1.1.-SELECCIÓN DE UN MÉTODO DE INSTALACIÓN

¿Qué tipo de método de instalación desea utilizar? Están disponibles los siguientes métodos de instalación:

III.1.1.1.-Cd-rom

Si posee un lector de CD-ROM y tiene el CD-ROM de Red Hat Linux, puede utilizar este método. Necesitará un disquete de arranque o un CD-ROM para arrancar.

III.1.1.2.-Disco duro

Si ha copiado las imágenes ISO de Red Hat Linux en el disco duro local, puede utilizar este método. Necesitará un disquete de arranque.

III.1.1.3.-Imagen NFS

Si está realizando la instalación desde un servidor NFS utilizando imágenes ISO o una imagen réplica de Red Hat Linux, puede utilizar este método. Necesitará un disquete de arranque de red.

III.1.1.4.-Ftp

Si está realizando la instalación directamente desde un servidor FTP, utilice este método. Necesitará un disquete de arranque de red.

III.1.1.5.-Http

Si está realizando la instalación directamente desde un servidor Web HTTP, utilice este método. Necesitará un disquete de arranque de red.

En este caso nosotros escogimos instalar desde un CD-ROM

III.1.2.-INSTALACIÓN DESDE UN CD-ROM

Para instalar Red Hat Linux desde un CD-ROM, escoja CD-ROM y seleccione OK. Cuando el programa se lo indique, inserte el CD de Red Hat Linux en el lector de disco (si no arrancó desde una unidad de CD). Una vez que el CD esté en la unidad de CD-ROM, seleccione OK, y presione ENTER.

III.1.2.1.-¿Qué hacer si no se detecta el CD-ROM IDE?

Si tiene un CD-ROM IDE (ATAPI), pero el programa de instalación no lo detecta, le preguntará qué tipo de CD-ROM tiene. En este caso reinicie el programa de instalación y en el intérprete de comandos boot: escriba **linux hdX=cdrom**. Reemplace la **X** con una de las siguientes letras, dependiendo de la interfaz a la que está conectada la unidad y de si está configurado como maestro o esclavo (también conocido como primario y secundario):

- Primer controlador IDE, maestro
- Primer controlador IDE, esclavo
- Segundo controlador IDE, maestro
- Segundo controlador IDE, esclavo

Si tiene un tercer o un cuarto controlador, continúe simplemente asignando las letras en orden alfabético, pasando de controlador a controlador, de maestro a esclavo).

III.1.3.-BIENVENIDO A RED HAT LINUX

Por favor lea el texto de ayuda en el panel de la izquierda para instrucciones adicionales e información sobre el registro de su producto Red Hat Linux.

Haga click en siguiente para continuar

III.1.4.-SELECCIÓN DEL IDIOMA

Elija el idioma que quiere usar por defecto para la instalación y para el sistema. La selección del idioma apropiado le ayudará también a seguir los pasos correctos para configurar la zona horaria (huso horario). El programa de instalación intentará definir el huso horario adecuado basándose en su configuración.



Figura 3-1. Selección del idioma

Una vez que haya seleccionado el idioma, haga click en siguiente para continuar.

III.1.5.-CONFIGURACIÓN DEL TECLADO

Elija el tipo de teclado que mejor se adapte a su sistema

Haga click en siguiente para continuar.



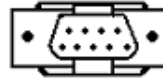
Figura 3-2. Configuración del teclado

III.1.6.-CONFIGURACIÓN DEL RATÓN

Elija el ratón adecuado a su sistema. Si no encuentra el tipo exacto, elija el que crea que será compatible con el suyo.

Si está instalando Red Hat Linux en un ordenador portátil, en la mayoría de los casos el dispositivo en cuestión será compatible con PS/2.

Si su ratón es serial, el puerto será parecido a:



Si su ratón es un ratón PS/2, el puerto será parecido a:



Si su ratón es un ratón USB, el puerto será parecido a:



Si su ratón es un ratón AT (Advanced Technology), el puerto será parecido



Si no encuentra un ratón del que esté seguro que es compatible con su sistema, seleccione una de las entradas Generic, basadas en el número de botones de su ratón y de su interfaz.

Si tiene un ratón de scroll, seleccione la entrada Generic - Wheel Mouse (con el puerto del ratón correcto) como un tipo de ratón compatible.

Si tiene un ratón PS/2, USB, o Bus, no necesita seleccionar un puerto y dispositivo. Si tiene un ratón serial, seleccione el puerto correcto y dispositivo en el que su ratón está conectado.

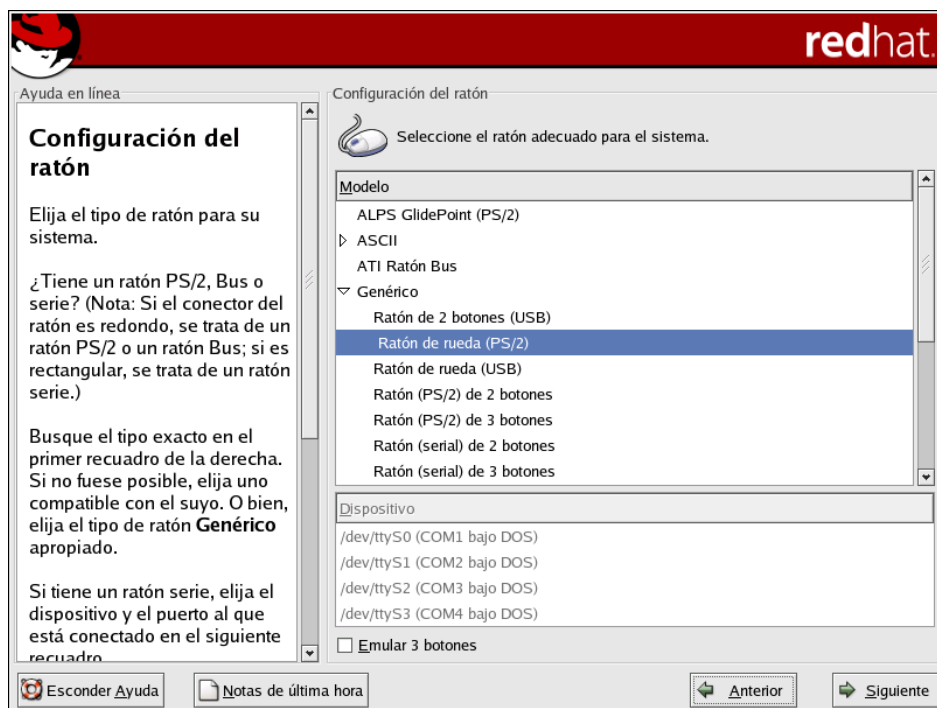


Figura 3-3. Configuración del ratón

III.1.7.-SELECCIONAR ACTUALIZAR O INSTALAR

La pantalla Examinar la actualización aparece automáticamente si el programa de instalación detecta una versión previa de Red Hat Linux en su sistema.

Si desea llevar a cabo una actualización, seleccione Actualizar una instalación existente. Asegúrese de seleccionar Personalizar los paquetes a actualizar si desea tener mayor control sobre cuáles paquetes serán actualizados en su sistema.

Para realizar una nueva instalación de Red Hat Linux en su sistema, seleccione Realizar una nueva instalación de Red Hat Linux y haga click en siguiente

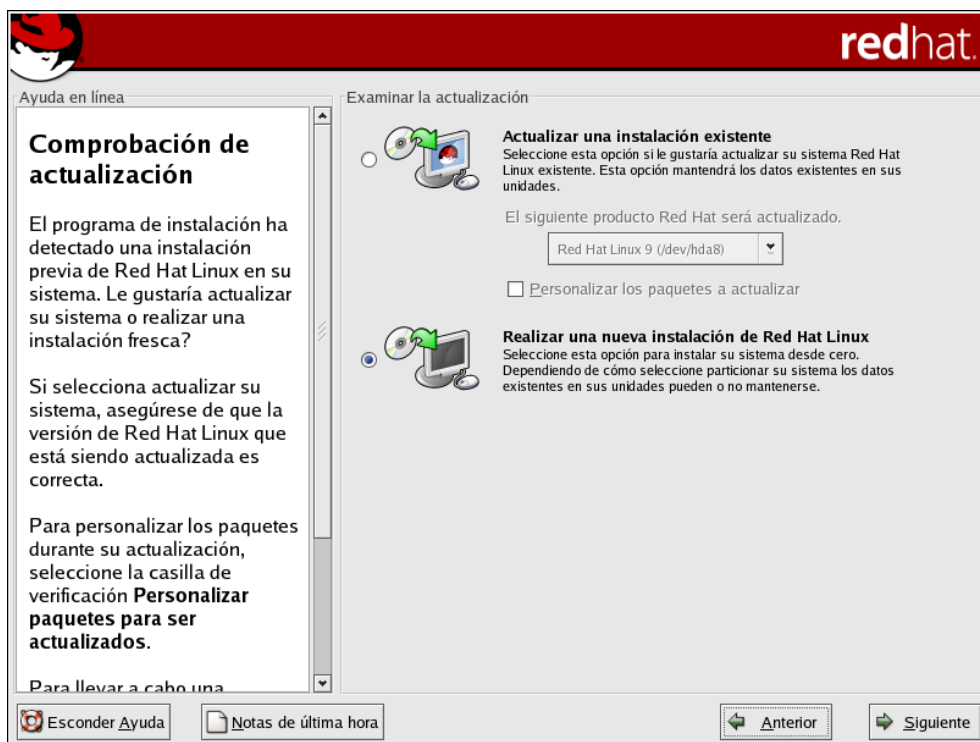


Figura 3-4. Comprobación de actualización

III.1.8.-OPCIONES DE INSTALACIÓN

Elija qué tipo de instalación desea realizar. El sistema Red Hat Linux le permitirá elegir el tipo de instalación que mejor se ajuste a sus necesidades. Las opciones disponibles son:

1. Estación de trabajo,
2. Servidor,
3. Portátil,
4. Personalizada y
5. Actualización.



Figura 3-5. Elegir el tipo de instalación

III.1.9.-CONFIGURACIÓN DEL PARTICIONAMIENTO DEL DISCO

El particionamiento es especialmente útil si ejecuta más de un sistema operativo. En esta pantalla, puede elegir entre realizar un particionamiento automático o un particionamiento manual con Disk Druid.

El particionamiento automático le permite realizar una instalación sin tener que particionar los discos usted mismo. Si no se siente seguro durante la partición manual de su disco, se aconseja elegir la partición automática, a fin de que el proceso de instalación decida por ud. qué tipo de partición adoptar.

Para particionar de forma manual, escoja la herramienta de particionamiento Disk Druid.

Si particiona el sistema manualmente, y crea una partición /var, asegúrese que crea la partición lo suficientemente grande para poder descargar todas las actualizaciones del paquete.

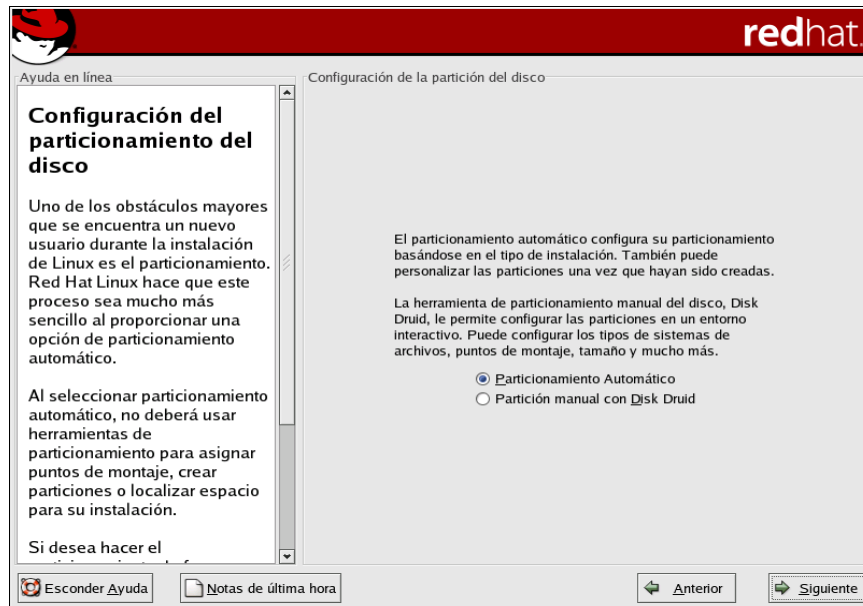


Figura 3-6. Configuración del particionamiento del disco

III.1.10.-PARTICIONAMIENTO AUTOMÁTICO

El particionamiento automático le permite tener control de los datos que se han eliminado en su sistema. Tiene las siguientes opciones:

III.1.10.1.-Eliminar todas las particiones Linux del sistema

Seleccione esta opción para eliminar tan sólo las particiones Linux (particiones creadas en una instalación Linux previa). No borrará el resto de particiones que tenga en el disco(s) duro(s) (tal como VFAT o particiones FAT32).

III.1.10.2.-Eliminar todas las particiones del sistema

Seleccione esta opción para eliminar todas las particiones de su disco duro (esto incluye las particiones creadas por otros sistemas operativos tales como Windows 95/98/NT/2000).

III.1.10.3.-Mantener todas las particiones y usar el espacio libre existente

Seleccione esta opción para conservar los datos y las particiones actuales, presumiendo que tiene suficiente espacio disponible en los discos duros.

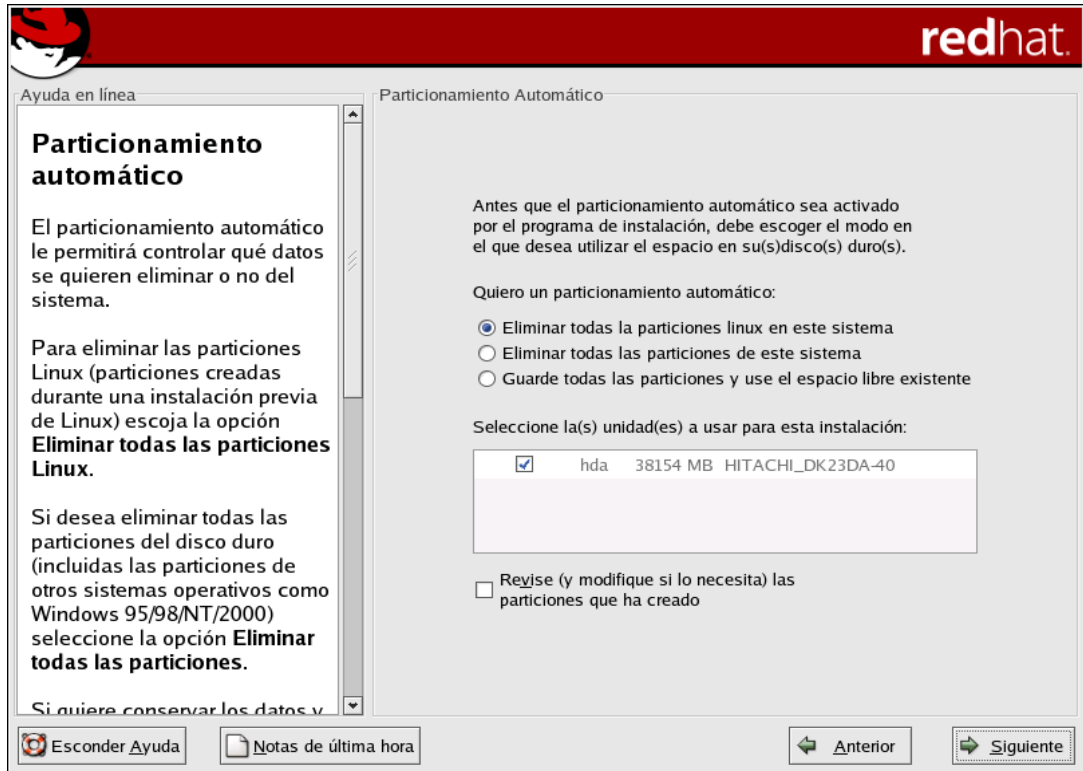


Figura 3-7. Particionamiento automático

Si tiene dos o más discos duros, puede escoger qué disco duro debería contener esta instalación. Esto no repercutirá a los discos duros no seleccionados ni a ninguno de sus datos.

Para revisar y realizar los cambios necesarios en las particiones creadas con el particionamiento automático, seleccione la opción Revisar. Después de seleccionar Revisar y hacer click en siguiente para continuar, verá las particiones creadas en la aplicación Disk Druid. Haga click en siguiente una vez que haya hecho sus selecciones para continuar.

III.1.11.-PARTICIONAMIENTO DEL SISTEMA

Si ha escogido particionamiento automático y seleccionado Revisar, puede aceptar la configuración de las particiones actuales (haga click en siguiente) o modificar la configuración mediante el uso de Disk Druid, la herramienta de particionamiento manual.

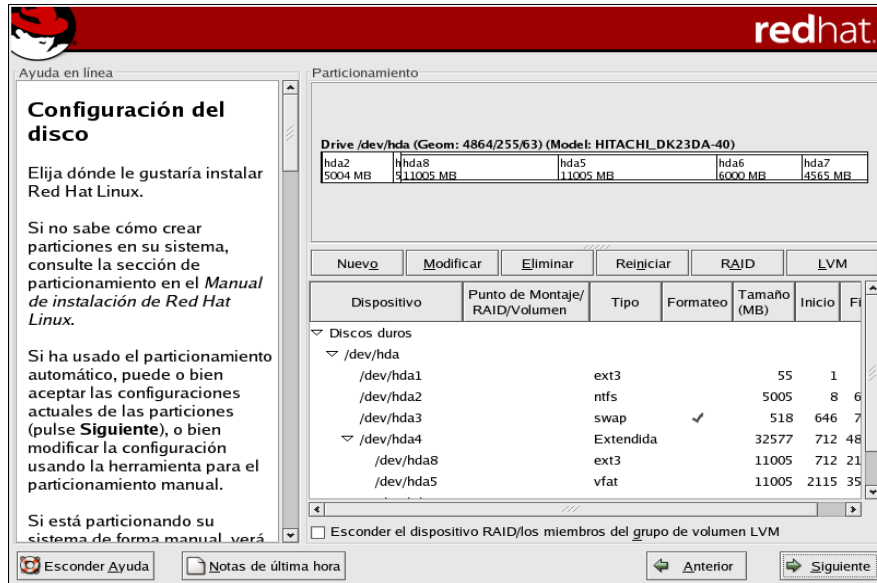


Figura 3-8. Particionamiento con Disk Druid

La herramienta de particionamiento usada en Red Hat Linux 9 será el Disk Druid. Con la excepción de ciertas situaciones "ocultas", el Disk Druid normalmente mantiene los requisitos de particionamiento de una instalación normal de Red Hat Linux.

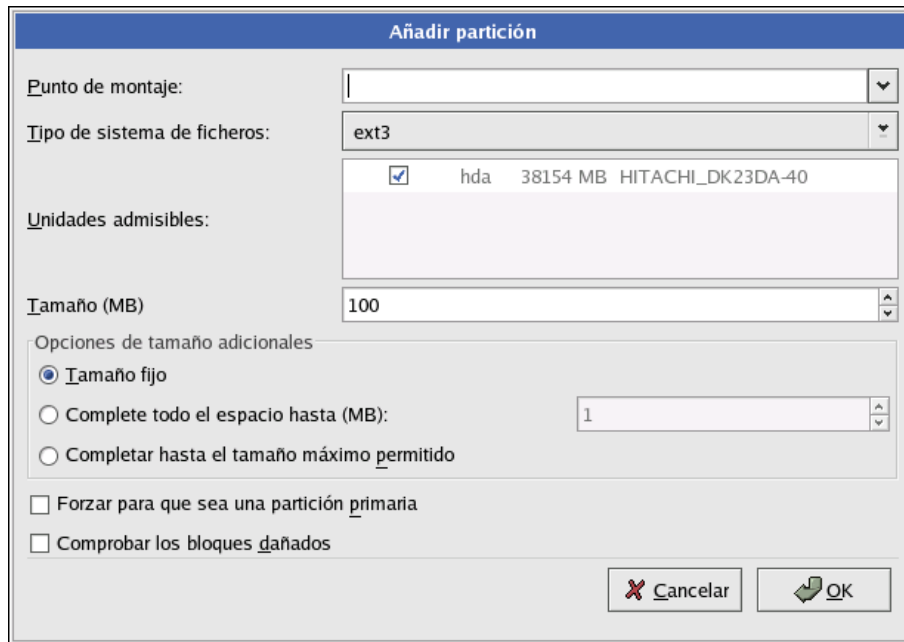


Figura 3-9. Creación de una nueva partición

III.1.11.1.-Punto de montaje:

Introduzca el punto de montaje de la partición. Por ejemplo, si la partición va a ser la partición raíz, introduzca /; o introduzca **/boot** para la partición **/boot**, etc.

III.1.11.2.-Tipo de sistema de archivos:

Usando el menú desplegable, seleccione el tipo de sistema de archivo apropiado para esta partición...

Tipos de sistemas de archivos

Red Hat Linux le permite crear diferentes tipos de particiones, basadas en el sistema de archivos que usarán. A continuación tiene una breve descripción de los diferentes sistemas de archivos disponibles y de cómo puede utilizarlos.

1. Ext2

Un sistema de archivos ext2, soporta tipos de archivo estándar Unix (archivos regulares, directorios, enlaces simbólicos, etc.). Proporciona la habilidad de asignar nombres de archivos largos, hasta 255 caracteres. Las versiones anteriores a Red Hat Linux 7.2 utilizaban el sistema de archivos ext2 por defecto.

2. Ext3

El sistema de archivos ext3, está basado en el sistema de archivos ext2 y tiene una ventaja principal — journaling. El uso de un sistema de archivos journaling reduce el tiempo de recuperación tras una caída al sistema de archivos. El sistema de archivos ext3 está seleccionado por defecto y su uso es bien recomendado.

3. Volumen físico (LVM)

Mediante la creación de una o más particiones LVM le permite crear un volumen lógico

4. Software RAID

La creación de dos o más particiones de software RAID le permite crear un dispositivo RAID. (***Redundant Array of Independent Disks***)

5. Swap

Las particiones swap se usan para soportar memoria virtual. En otras palabras, los datos se escriben en una partición swap cuando no hay suficiente RAM para guardar los datos que su sistema está procesando.

6. Vfat

El sistema de archivos VFAT es un sistema de archivos Linux compatible con Windows 95/NT en el sistema de archivos FAT de nombres largos.

III.1.12.-CONFIGURACIÓN DEL GESTOR DE ARRANQUE

Para poder arrancar su sistema Red Hat Linux sin el disquete de arranque, necesitará instalar un gestor de arranque. El gestor de arranque es el primer software que se ejecuta cuando se arranca el ordenador. Es responsable de la carga y de la transferencia del control al software del sistema operativo del kernel. El kernel, por otro lado, inicializa el resto del sistema operativo.

El programa de instalación le ofrece dos gestores de arranque, GRUB y LILO.

III.1.13.-GRUB (GRAND UNIFIED BOOTLOADER)

El Gran Unificador de Arranque se instala por defecto y es un gestor de arranque muy potente ya que puede cargar una gran variedad de sistemas operativos gratuitos así como sistemas operativos de propietarios, con el sistema de cargado en cadena (el mecanismo para cargar sistemas operativos no soportados mediante la carga de otro gestor de arranque, tal como DOS o Windows).

III.1.14.-LILO (LINUX LOADER)

Es un gestor de arranque versátil para Linux. No depende de un determinado sistema de ficheros y puede arrancar las imágenes del kernel de Linux de los disquetes y de los discos duros e incluso arrancar otros sistemas operativos.

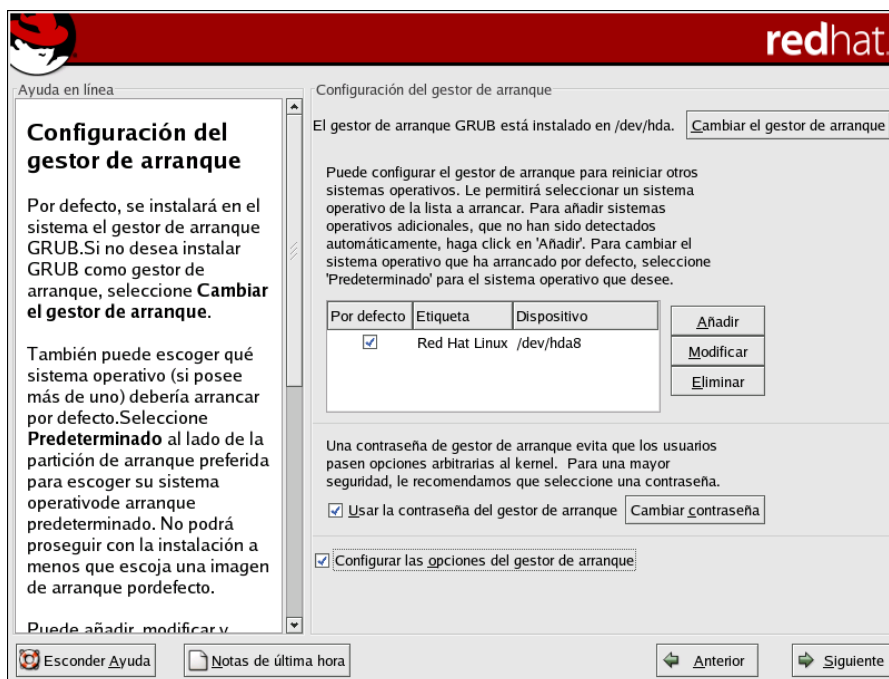


Figura 3-10. Configuración del gestor de arranque

Si no desea instalar GRUB como su gestor de arranque, haga click en cambiar el gestor de arranque. Puede escoger entre instalar LILO o no instalar ninguno. Si ya tiene otro gestor de arranque que arranque Linux y no desea sobrescribir el gestor de arranque actual o si desea arrancar el sistema con disquetes, elija No instalar gestor de arranque haciendo click en el botón cambiar el gestor de arranque.

NOTA:

Si decide no instalar LILO o GRUB no podrá arrancar el sistema directamente y necesitará usar otro método de arranque (como el disquete de arranque). ¡Use esta opción solamente si conoce otra manera de arrancar el sistema!.

III.1.15.-CONFIGURACIÓN AVANZADA DEL GESTOR DE ARRANQUE

Ahora que ha decidido cuál gestor de arranque instalar, tiene que decidir dónde quiere instalarlo. Puede instalar el gestor de arranque en uno de los dos sitios siguiente:

III.1.15.1.-El master boot record (mbr)

Este es el sitio recomendado para instalar un gestor de arranque, a no ser que el MBR esté ocupado por el gestor de arranque de otro sistema operativo, como System Commander. El MBR es un área especial del disco duro que la BIOS de su ordenador carga automáticamente y el punto más próximo en el que el gestor de arranque puede tomar el control de la secuencia de arranque. Si lo instala en el MBR, al arrancar su máquina, GRUB (o LILO) presentará un indicador de comandos de arranque. Podrá entonces iniciar Red Hat Linux o cualquier otro sistema operativo que le haya indicado al gestor de arranque.

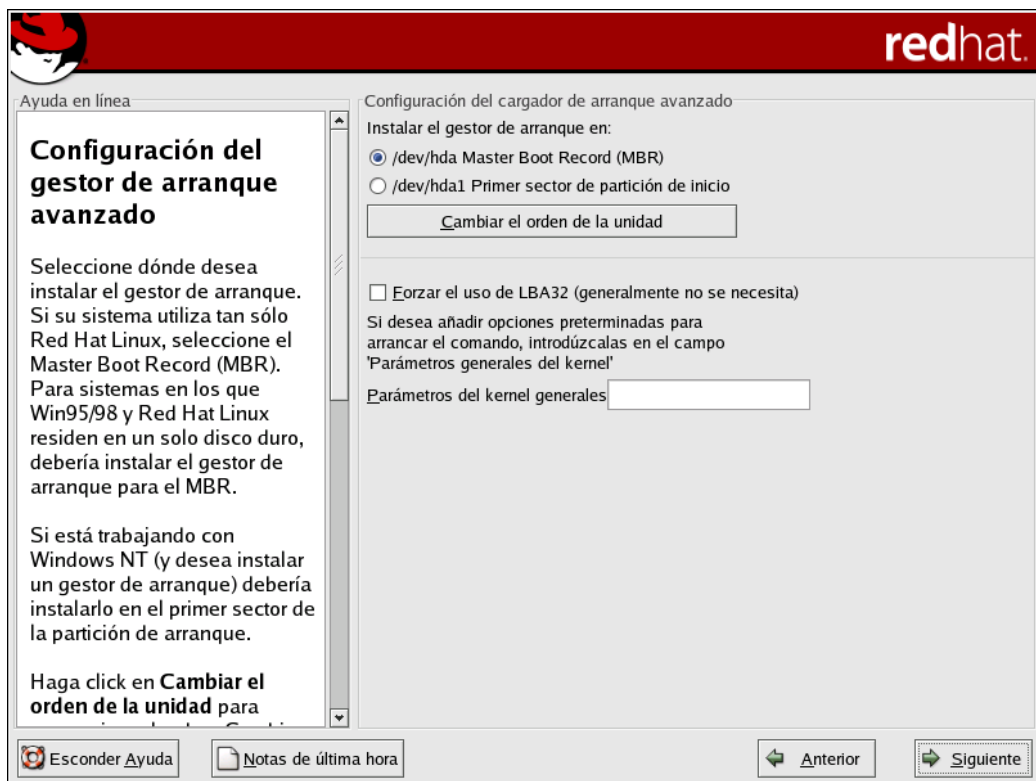


Figura 3-11. Configuración del gestor de arranque

III.1.16.-CONFIGURACIÓN DE RED

Si tiene un dispositivo de red y no ha configurado todavía su red (como por ejemplo proporcionar un disco de arranque de red que haya creado y entrar en la información de red como se indica), tiene la oportunidad de hacerlo (como se muestra en la Figura 3-12).

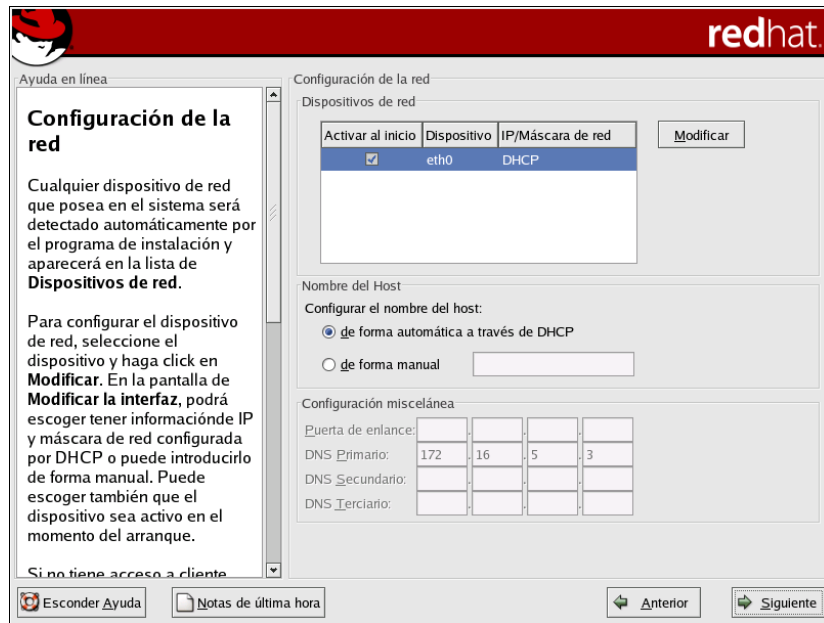


Figura 3-12. Configuración de red

El programa de instalación automáticamente detecta los dispositivos de red que tiene y los muestra en la lista Dispositivos de red.

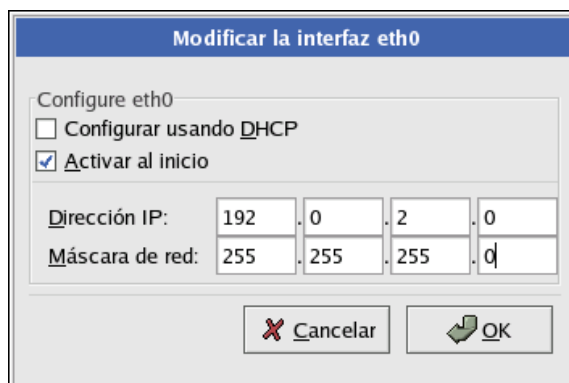


Figura 3-13. Modificación del dispositivo de red

Si tiene un nombre de dominio para el dispositivo de red, introdúzcalo en el campo Nombre del host o puede elegir tener el DHCP automáticamente.

Para finalizar, introduzca la Puerta de enlace y el DNS primario (si los conoce, también el DNS secundario y DNS terciario).

III.1.17.-CONFIGURACIÓN DEL CORTAFUEGOS

Red Hat Linux también le ofrece protección vía cortafuegos (firewall) para una seguridad mejorada del sistema. Un cortafuego se dispone entre su ordenador y la red y determina qué recursos de su equipo están accesibles para los usuarios remotos de la red. Un cortafuego bien configurado puede aumentar significativamente la seguridad de su sistema



Figura 3-14. Configuración del firewall

Seleccione el nivel de seguridad apropiado del sistema.

III.1.17.1.-ALTO

Si elige Alto, su sistema no aceptará más que las conexiones (además de las definidas por defecto) que hayan sido explícitamente definidas por usted. Por defecto, sólo se aceptan las siguientes conexiones:

III.1.17.2.-Respuestas de DNS

1. DHCP para que cualquier interfaz de red que use DHCP puedan ser configuradas correctamente

Si elige Alto, su cortafuego no permitirá:

2. FTP en modo activo (el modo pasivo de FTP, usado por defecto en la mayoría de los clientes debería funcionar)

III.1.17.3.-Transferencias de archivos IRC DCC

RealAudio

Clientes del Sistema X Windows remotos

III.1.18.-SELECCIÓN DEL SOPORTE DEL IDIOMA

Puede instalar y soportar múltiples idiomas para usar en su sistema.

Debe instalar un idioma para usarlo como idioma por defecto. El idioma por defecto será usado por el sistema una vez que la instalación se haya completado.

Si escoge instalar otros idiomas, puede cambiar su idioma por defecto tras la instalación.



Figura 3-15. Selección del soporte adicional del idioma

III.1.19.-CONFIGURACIÓN DEL USO HORARIO

Puede elegir su uso horario o bien seleccionando la localización física de su ordenador o bien especificando su uso horario en función del Universal Time Coordinated (UTC).

Fíjese en las dos pestañas en lo alto de la pantalla (vea la [Figura 3-16](#)). La primera pestaña le permite configurar el uso horario en función de su localización. En el mapa interactivo, puede hacer click en una ciudad específica, que están marcadas con un punto amarillo; una X roja aparecerá indicando su selección. Puede también desplazarse en la lista y seleccionar un uso horario.

La segunda pestaña le permite especificar un desplazamiento UTC. Esta pestaña muestra una lista de desplazamientos para elegir, así como las opciones para el cambio del horario de invierno/verano.



Figura 3-16. Configuración del uso horario

En ambas pestañas, puede seleccionar El reloj del sistema usa UTC. Elija esta opción si sabe que su equipo está en hora UTC.

Configuración de la contraseña de root

La configuración de la cuenta y la contraseña root es uno de los pasos más importantes durante la instalación. Su cuenta root es similar a la cuenta del administrador usada en las máquinas Windows NT. La cuenta root es usada para instalar paquetes, actualizar RPMs y realizar la mayoría de las tareas de mantenimiento del sistema. Conectándose como root le da control completo sobre el sistema.

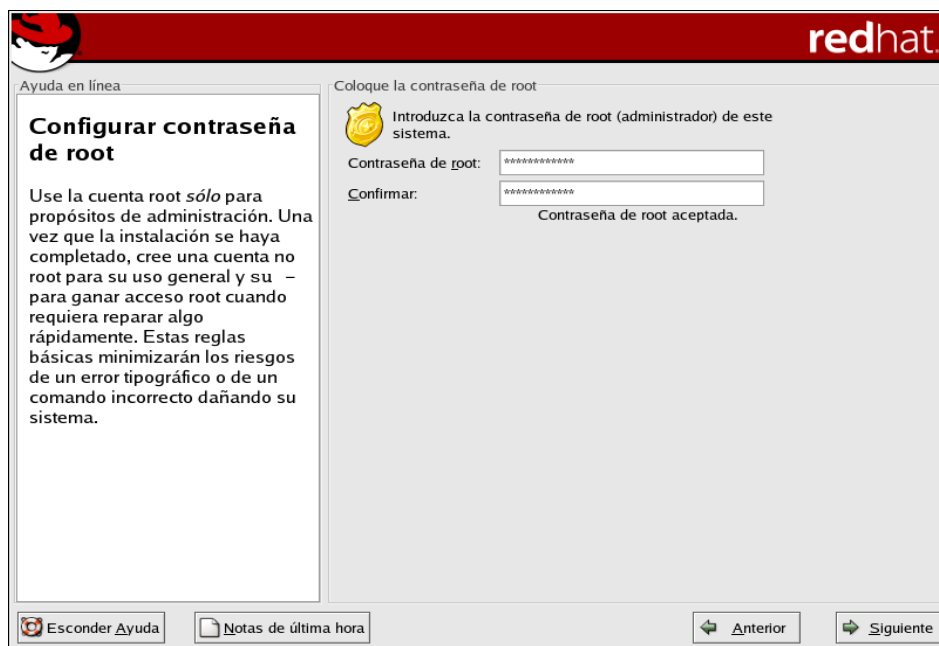


Figura 3-17. Configuración de la Contraseña de root

III.1.20.-CONFIGURACIÓN DE LA AUTENTICACIÓN

Este recuadro nos permite realizar la configuración de la autenticación para lo cual se debe seguir los siguientes pasos

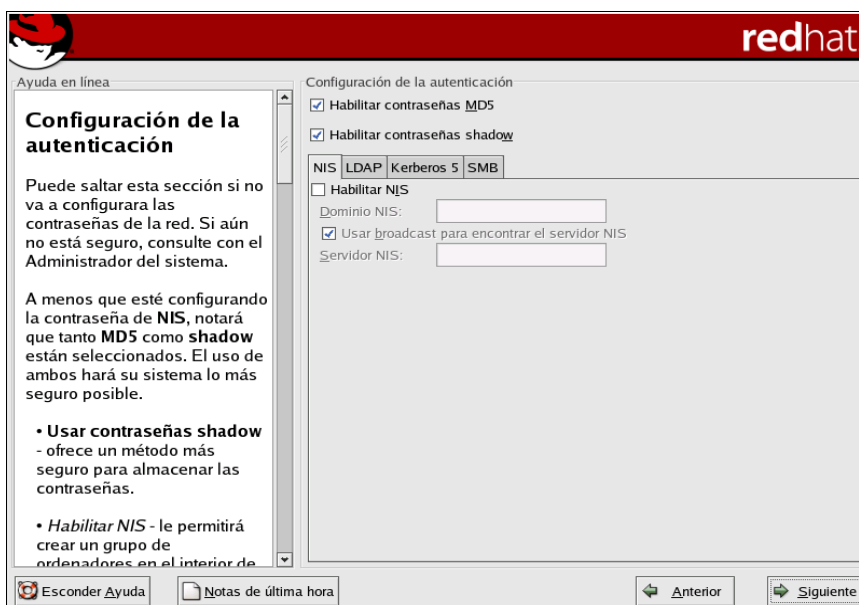


Figura 3-18. Configuración de la autenticación

III.1.20.1.-HABILITAR CONTRASEÑAS MD5

Le permite usar una contraseña larga (de hasta 256 caracteres), en vez de las ocho letras o menos estándar.

III.1.20.2.-HABILITAR CONTRASEÑAS SHADOW

Proporciona un método seguro para conservar contraseñas. Las contraseñas se almacenan en /etc/shadow, al que tan sólo se puede acceder como root.

III.1.20.3.-HABILITAR NIS

Permite ejecutar un grupo de ordenadores en el mismo dominio Network Information Service con una contraseña común y un fichero de grupo. Puede escoger desde las siguientes opciones:

III.1.20.3.1.-Dominio NIS

Le permitirá especificar el dominio o grupo de ordenadores al que pertenece su sistema.

III.1.20.3.2.-Usar broadcast para encontrar el servidor NIS

Le permitirá hacer un broadcast en su red local para encontrar el servidor NIS.

III.1.20.3.3.-Servidor NIS

Hace que su ordenador utilice un servidor NIS específico, en vez de enviar un broadcast a toda la red para encontrar el servidor NIS disponible.

Selección de grupos de paquetes

Para seleccionar paquetes individualmente, compruebe la casilla de verificación Personalizar el conjunto de paquetes a instalar. Puede seleccionar grupos de paquetes, los cuales agrupan componentes de acuerdo a una función (por

ejemplo, Sistema X Windows y Editores), paquetes individuales, o una combinación de los dos.

Para seleccionar un componente, haga click en la casilla de verificación (vea la Figura 3-19).

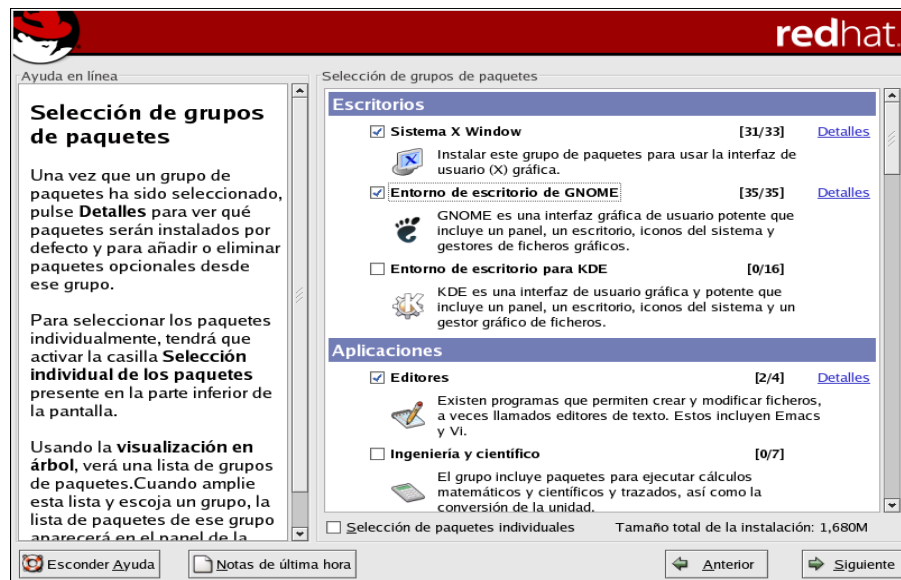


Figura 3-19. Selección de grupos de paquetes

Seleccione los componentes que desea instalar. Al seleccionar Todo (al final de la lista de componentes) en una instalación tipo personalizado se instalarán todos los paquetes incluidos con Red Hat Linux.

Una vez seleccionado el grupo de paquetes, haga click en Detalles para visualizar los paquetes que se instalarán por defecto y los paquetes que desee eliminar o añadir a ese grupo.

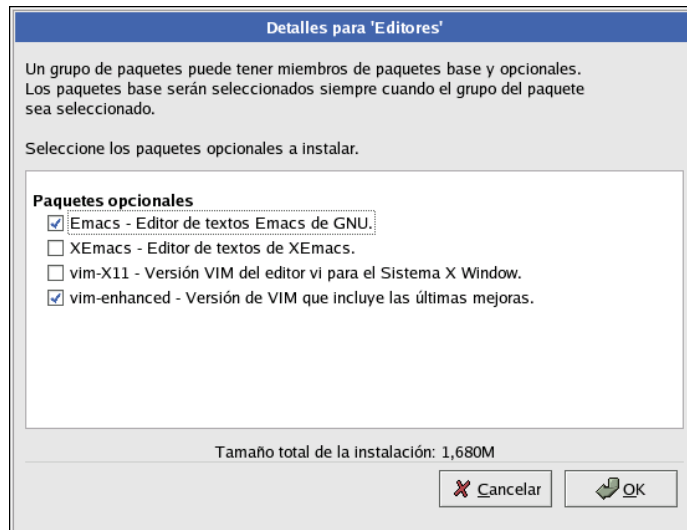


Figura 3-20. Selección de los detalles para editores

III.1.21.-SELECCIÓN INDIVIDUAL DE PAQUETES

Tras haber seleccionado los paquetes que quiera instalar, podrá seleccionar o anular la selección de los paquetes individualmente usando su ratón (vea la Figura 3-21).

Puede escoger visualizar los paquetes individuales en Vista de árbol o Vista plana.

La Vista de árbol le permite ver los paquetes agrupados según el tipo de aplicación.

La Vista plana le permite ver todos los paquetes listados en orden alfabético en la parte derecha de la pantalla.

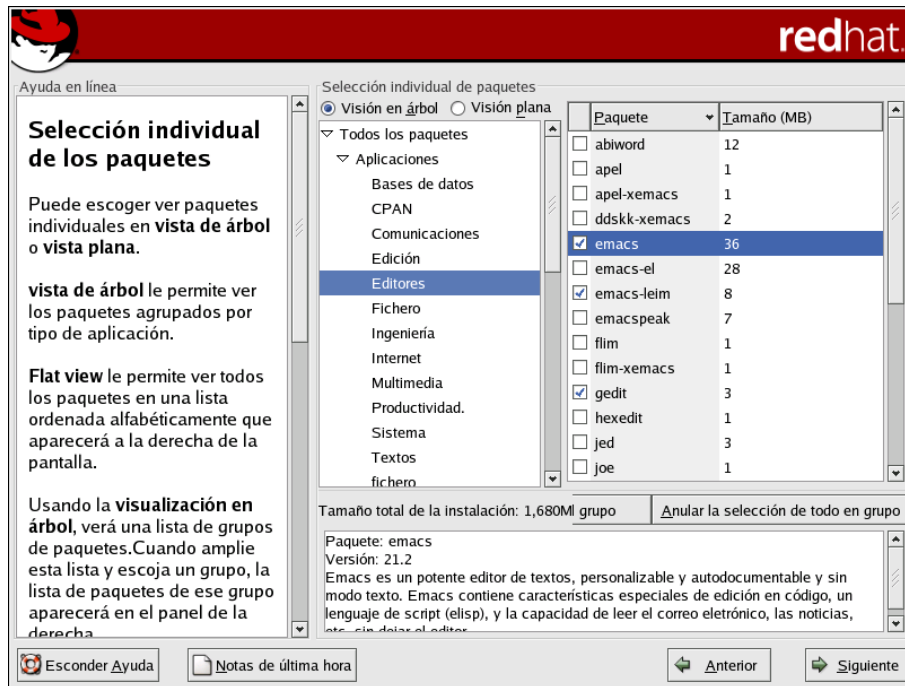


Figura 3-21. Selección individual de los paquetes

Para clasificarlos alfabéticamente, haga click en la pestaña Paquete. Para clasificarlos por tamaño, haga click en la pestaña Tamaño (MB).

Para seleccionar un paquete individual, haga doble click en la casilla de verificación situada junto al nombre del paquete. Si la casilla de verificación aparece marcada, esto quiere decir que será instalado.

III.1.22.-PREPARACIÓN PARA LA INSTALACIÓN

Debería ver una pantalla de preparación para su instalación de Red Hat Linux.

Para cancelar este proceso de instalación, pulse el botón Reset de su ordenador o use la combinación de teclas [Control]-[Alt]-[Suprimir] para reiniciar su máquina.

III.1.23.-Instalación de paquetes

En este momento, no podrá hacer nada hasta que todos los paquetes hayan sido instalados (vea la Figura 3-22). La rapidez de este proceso dependerá del número de paquetes que haya seleccionado y de la velocidad de su ordenador.



Figura 3-22. Instalación de los paquetes

III.1.24.-CREACIÓN DE UN DISQUETE DE ARRANQUE

Para crear un disco de arranque, introduzca un disco en blanco, formateado en su unidad de disco (vea la Figura 3-23) y haga click en siguiente.

Se le recomienda que cree un disquete de arranque. Si, por alguna razón, su sistema no fuera capaz de arrancar adecuadamente usando GRUB, LILO, u otro cargador externo, un disco de arranque le permitirá arrancar de forma adecuada su sistema Red Hat Linux.

Si no desea crear un disco de arranque, asegúrese de que selecciona la opción adecuada antes de pulsar siguiente.

Si arranca el sistema con el disquete de arranque (en vez de GRUB o LILO), asegúrese de crear un nuevo disco de arranque siempre que haga cambios en el kernel (incluyendo la instalación de un nuevo kernel).

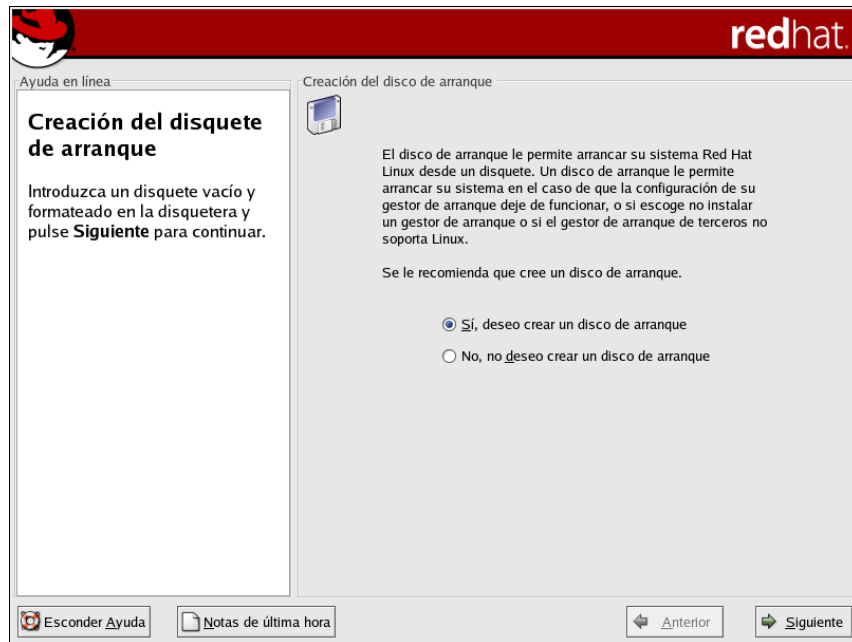


Figura 3-23. Creación del disquete de arranque

III.1.25.-CONFIGURACIÓN DE LA TARJETA DE VÍDEO

El programa de instalación a continuación le proporcionará una lista de tarjetas de vídeo entre las que escoger.

Si decide instalar los paquetes del Sistema X Windows, tendrá la oportunidad de configurar un servidor X para su sistema.

Si su tarjeta de vídeo no aparece en la lista, X puede que no la soporte. No obstante, si posee conocimiento técnico sobre su tarjeta, puede escoger Tarjeta no listada e intentar configurarla al hacer corresponder su chipset de tarjeta de vídeo con uno de los servidores X disponibles.



Figura 3-24. Configuración de la interfaz grafica (X)

Configuración de X Control y personalización

Para completar la configuración de X, deberá configurar su monitor y personalizar la configuración de X.

III.1.26.-CONFIGURACIÓN DEL MONITOR

El programa de instalación le presentará una lista de monitores de la que seleccionar. Desde esta lista, puede usar el monitor que se detecte de forma automática o escoger otro monitor.

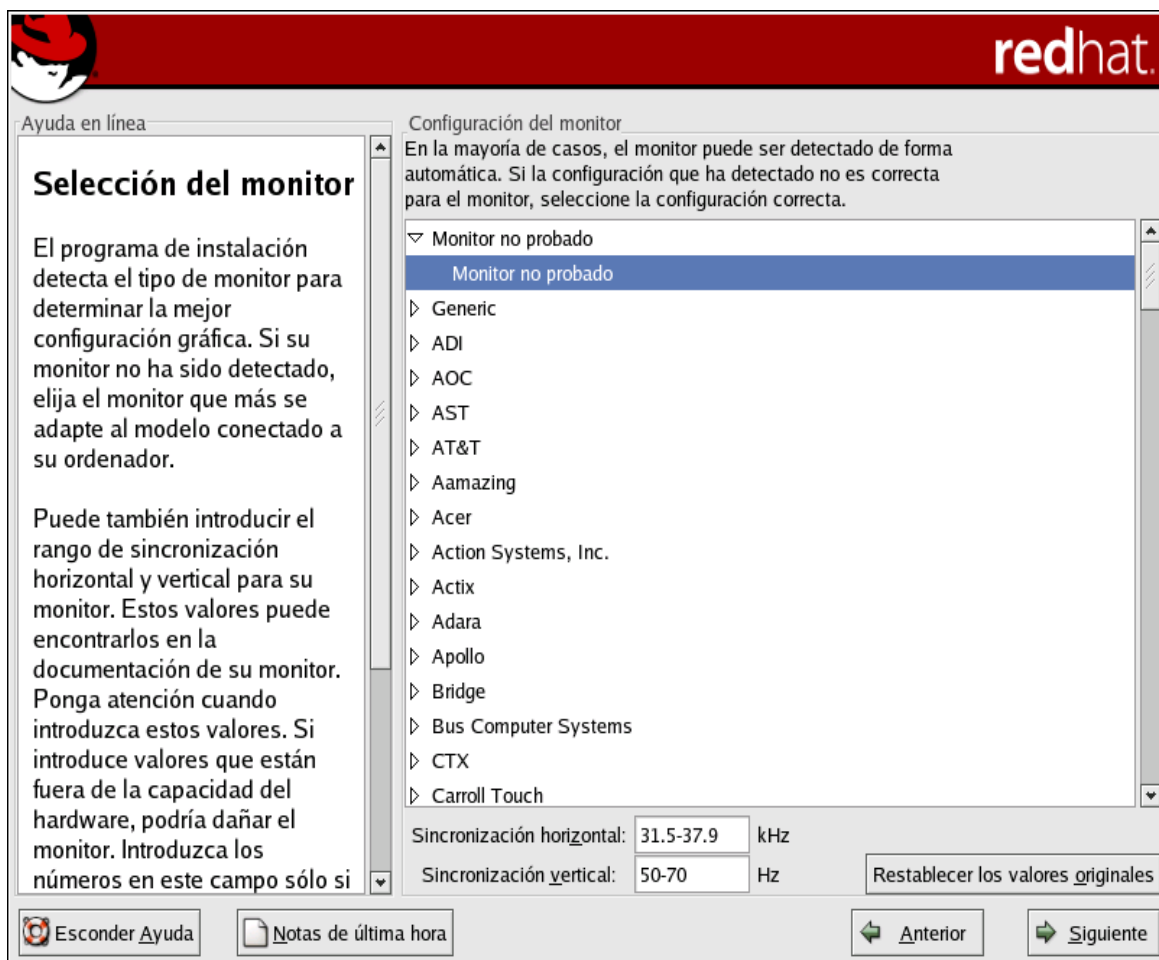


Figura 3-25. Selección del monitor

Los rangos de sincronización horizontal y vertical que el programa de instalación sugiere para su monitor se mostrarán debajo de la lista de monitores.

Haga click en siguiente cuando haya terminado de configurar su monitor.

Configuración personalizada

Elija la densidad del color y la resolución para su configuración de X.

Las instalaciones de escritorio personal y de estación de trabajo automáticamente arrancarán en ambiente gráfico.



Figura 3-26. Personalización de X

III.1.27.-FIN DE LA INSTALACIÓN

El programa de instalación le pedirá que prepare su sistema para reiniciarse. No se olvide de sacar cualquier disco de las disqueteras y CD de la unidad de CD-ROM.)

Si no tiene un gestor de arranque instalado y configurado, necesitará usar el disco de arranque que ha creado durante la instalación.

Después de que la secuencia de encendido se haya terminado, debería visualizar el intérprete de comandos del gestor de arranque gráfico en el que puede hacer cualquiera de las siguientes cosas:

Pulse Enter - se reiniciará la entrada de inicio por defecto.

Seleccionar una etiqueta de arranque seguida de Enter - provocará que el gestor de arranque inicie el sistema operativo correspondiente a la etiqueta de arranque. (Pulse [?] o [Tab] en el intérprete de comandos del cargador de arranque en modo texto para una lista de etiquetas de arranque válidas.)

No hacer nada tras un período de espera, LILO, (por defecto, cinco segundos) inicializará la primera partición automáticamente.

Haga lo que sea necesario para arrancar Red Hat Linux. Debería ver una o más ventanas de mensajes en las que hacer el scroll. También debería ver un intérprete de comandos login: o una pantalla gráfica de login (si ha instalado el sistema X Windows y ha escogido arrancarlo en modo gráfico por defecto).

La primera vez que arranque su máquina con Red Hat Linux, se le presentará el Agente de configuración, la cual lo guiará a través de la configuración de Red Hat Linux. Usando esta herramienta, puede configurar la fecha y hora, instalar software, registrar su máquina con Red Hat Network, y mucho más. El Agente de configuración le permite configurar su entorno al comienzo, para que pueda comenzar a utilizar su sistema Red Hat Linux rápidamente.

Referencia bibliográfica:

Tomado de la página Web:

<http://www.tu-chemnitz.de/docs/lindocs/RH9/RH-DOCS/rhl-ig-x86-es-9/index.html>

III.2.-INSTALACIÓN DE UBUNTU V8.10

Esta nueva versión, la V8.10 Intrepid Ibex, es una versión que contiene grandes cambios, mejorando rendimiento, aspecto y facilidad de uso. Las nuevas características que contiene Intrepid Ibex son las siguientes:

III.2.1.-CARACTERÍSTICAS

1. Gnome 2.24
2. X.Org 7.4
3. Linux Kernel 2.6.27
4. Directorio privado encriptado
5. Sesión de Invitado
6. Network Manager 0.7
7. DKMS (Dynamic Kernel Module Support)
8. Samba 3.2
9. Plugin Totem de la BBC
10. Inclusión de nuevos programas en los repositorios principales de Ubuntu tales como “ClamAV”, “Spam Assassin”, “Java OpenJDK 1.6” y “Tomcat Apache 6”.

Estas son las nuevas características más importantes de la nueva versión de Ubuntu, la 8.10 y también llamada Intrepid Ibex.

III.2.2.-PASOS PARA LA INSTALACIÓN DE UBUNTU

Para empezar, lo primero que tenemos que hacer, es descargarnos “Ubuntu 8.10 Intrepid Ibex” desde la página oficial, eligiendo ahí, la versión 8.10 Intrepid Ibex y

el tipo de procesador que utilices (32 o 64 bits). Por ejemplo, en mi caso elegiré la versión de 64 bits. Una vez tengamos Intrepid Ibex descargado en nuestro sistema, quemaremos la imagen del CD con nuestro programa habitual de grabación, ejemplo: "Nero", etc. Pues como ya tenemos nuestra imagen grabada en un CD, ahora solo tendríamos que reiniciar nuestro ordenador y una vez reiniciado, ya empezamos a utilizar Ubuntu. En primer lugar, nos saldría el menú de inicio de Ubuntu 8.10 Intrepid Ibex, en primer lugar tendremos la opción de elegir el idioma y en segundo lugar, las opciones de instalación:



Figura 3.2-1. Selección de tipo de instalación.

Ahora solo tenemos que elegir el tipo de inicio que nos gustaría para instalar Ubuntu. Pueden ser, como vemos en la imagen, los dos siguientes: **Probar Ubuntu sin alterar su equipo** e **Instalar Ubuntu**. Pero antes de elegir, mira con atención lo que te explicaremos a continuación:

III.2.2.1.-Probar Ubuntu sin alterar su equipo

Esto, significa, que podemos instalar Ubuntu de una manera de la que podamos probar el Sistema para ver si nos gusta, analizar sus funciones e inclusive para iniciar Ubuntu, las acciones realizadas en Ubuntu serán borradas. Estas características son, las que forman esencialmente, el Live CD.

III.2.2.2.-Instalar Ubuntu

Esta segunda forma de inicio significa, que una vez iniciado Ubuntu, solo te sirve para completar la Instalación del sistema, sin brindarte la posibilidad de probar Ubuntu. Esta opción no se suele usar mucho.

Elegimos la primera opción: **Probar Ubuntu sin alterar su equipo**

Una vez Ubuntu esté iniciado, nos aparece el nuevo escritorio de Ubuntu 8.10 Intrepid Ibex:

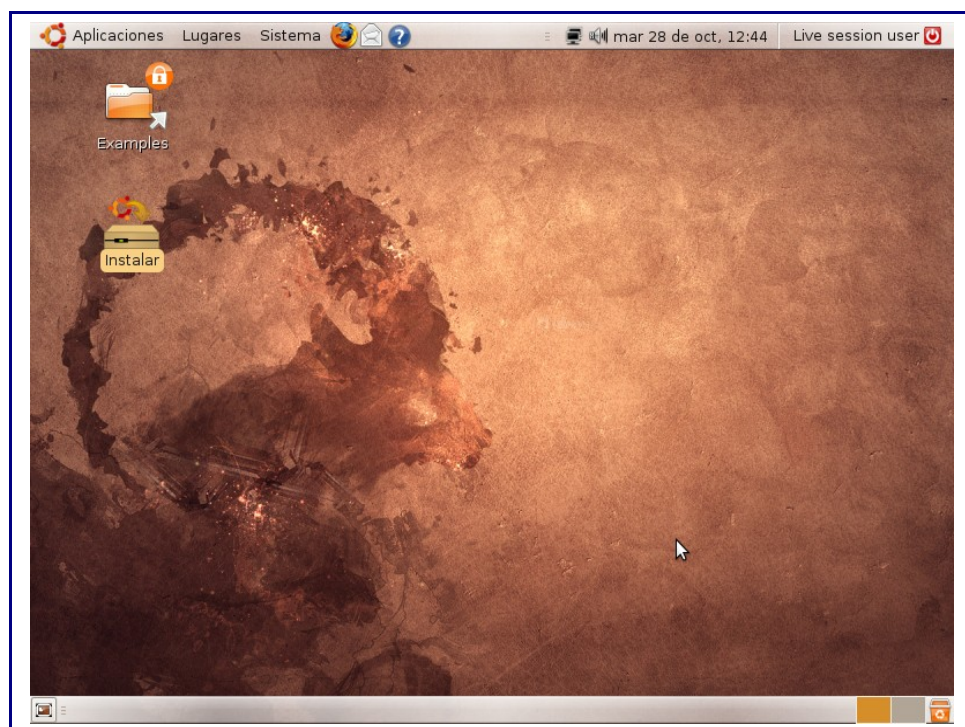


Figura 3.2-2. Interfaz de Ubuntu

III.2.3.-SELECCIÓN DE RED

Después de que el escritorio se halla cargado, si se nos da el caso de que lo vayamos a instalar a través de nuestra Wi-Fi, tendríamos que conectarnos mediante el icono de Network Manager, que son esos dos monitores que salen en la esquina superior derecha, y una vez listadas las redes, seleccionar la nuestra:

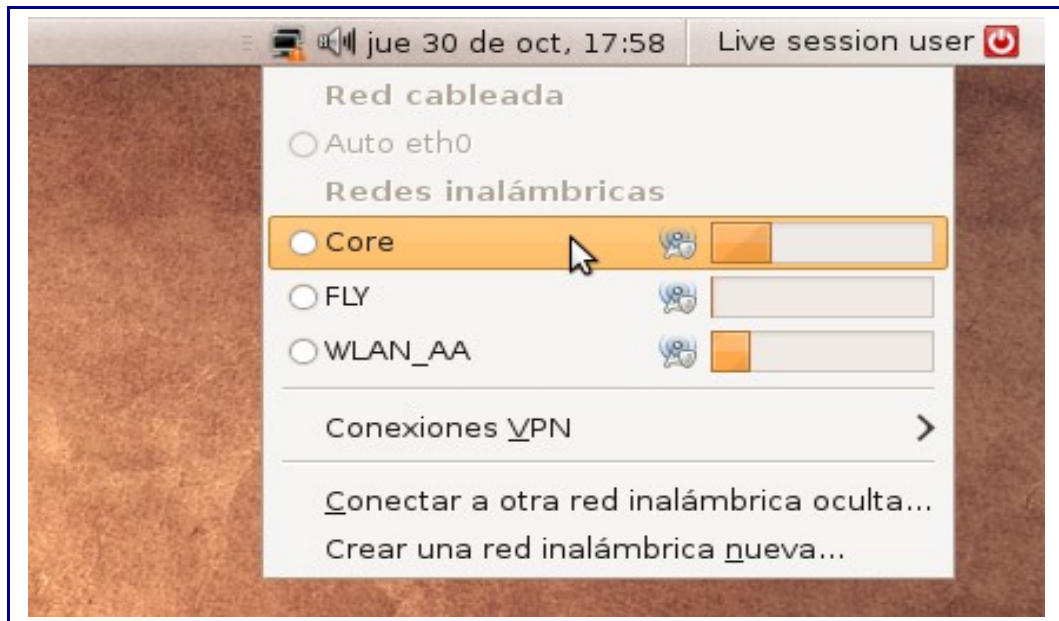


Figura 3.2-3. Selección de red

III.2.4.-AUTENTICACIÓN DE RED INALÁMBRICA

Y una vez seleccionada la red tendremos que introducir su clave para poder conectarnos (siempre y cuando la red tenga la seguridad activada)...

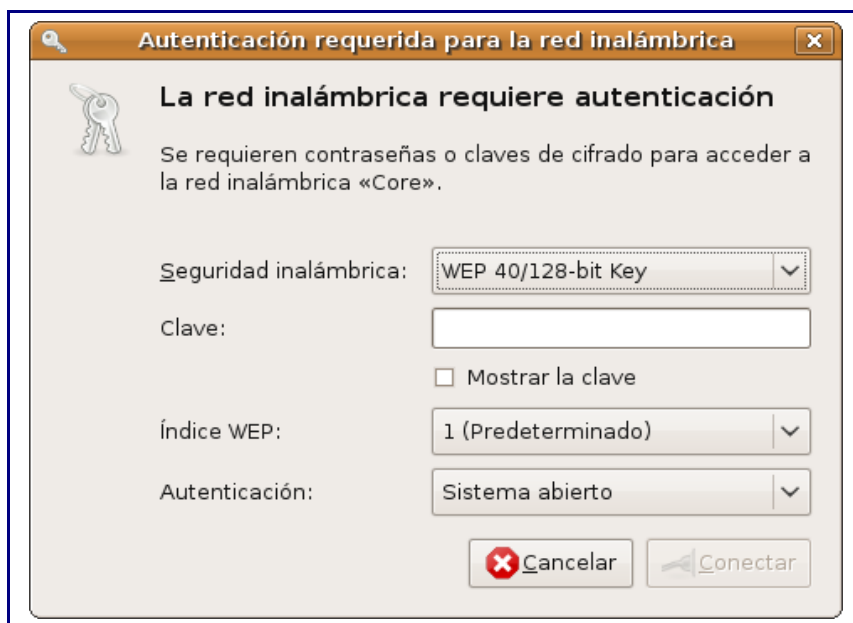


Figura 3.2-4. Autenticación de red inalámbrica

III.2.5.-CREACIÓN PREDETERMINADA PARA ANILLOS DE CLAVE

Segundos después, introducir la contraseña para crear el anillo de claves predeterminado:

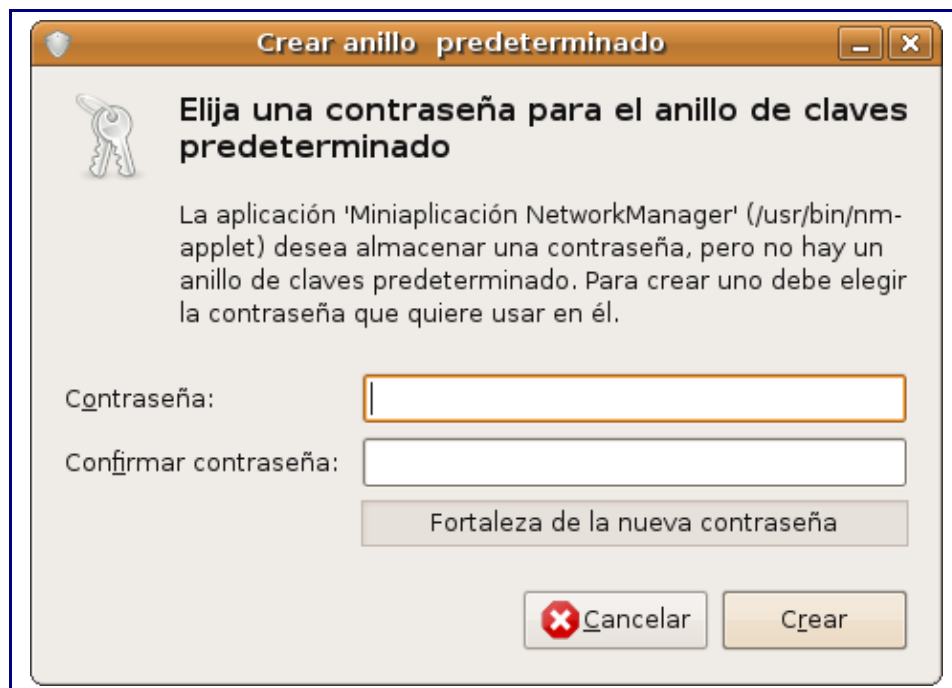


Figura 3.2-5. Creación de Anillos de claves

Pues bien, ahora como ya estamos con nuestro LiveCD de Ubuntu 8.10 Intrepid Ibex iniciado y conectados a internet a través de nuestra red inalámbrica, tan solo nos queda comenzar con el proceso general de instalación, en el cual nos introduciremos ahora.

III.2.6.-SELECCIÓN DEL IDIOMA

Para empezar, tenemos que dar doble click en “Instalar”, que es el enlace que se encuentra en el escritorio para poder Instalar Ubuntu. Una vez dado doble click, nos tendría que salir la ventana de bienvenida, donde por segunda vez, nos brindará la posibilidad de elegir nuestro idioma:

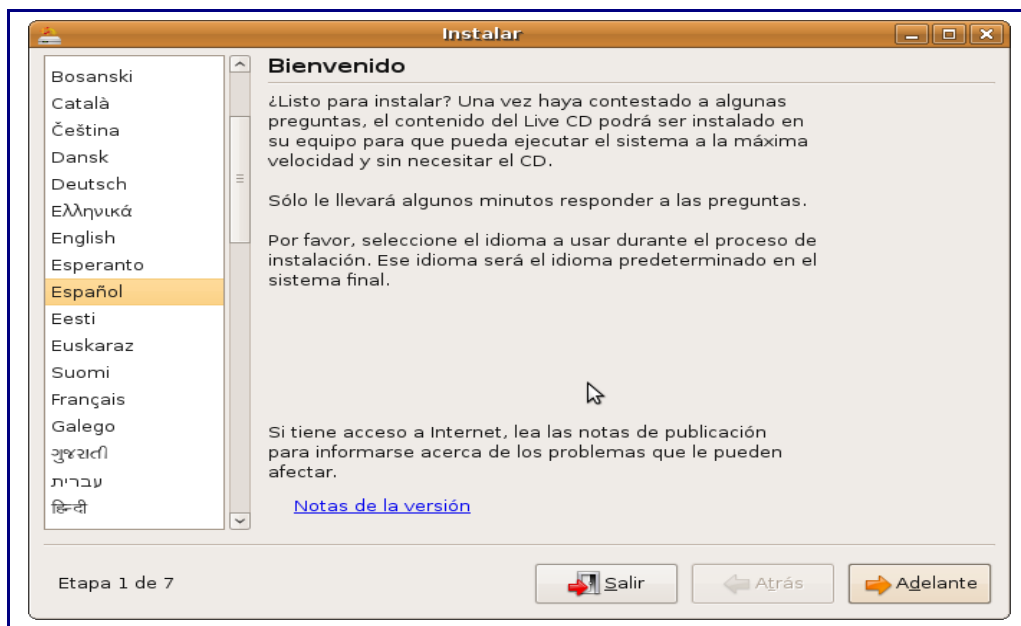


Figura 3.2-6. Selección de Idioma

En esta ventana tan solo damos clic en **Adelante** y pasamos al siguiente paso de instalación, que sería la selección de la zona horaria:

III.2.7.-SELECCIÓN DE ZONA HORARIA



Figura 3.2-7. Selección de la zona horaria

Damos clic en la pestaña **Ciudad seleccionada: Madrid** y escogemos la ciudad a la cual pertenecemos. Pues este paso es también muy sencillo y cuando localicemos nuestra zona y la seleccionemos, damos click en **Adelante** y pasamos al siguiente paso, que sería elegir la distribución del teclado:

III.2.8.-SELECCIÓN DE DISTRIBUCIÓN DEL TECLADO

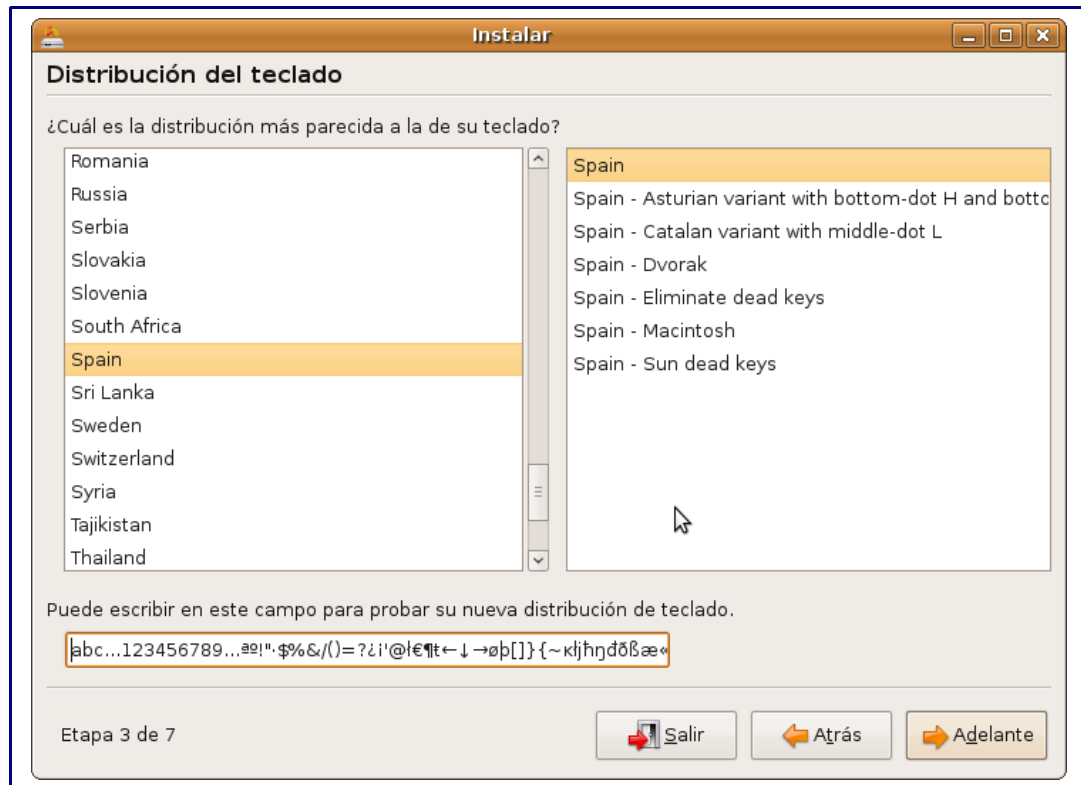


Figura 3.2-8. Distribución del teclado

Ahora, elegimos de acuerdo a nuestro teclado. Entonces, una vez elegido, el siguiente paso sería ya uno de los pasos más importantes de la instalación, que sería el manejo de particiones para poder instalar Ubuntu:

III.2.9.-MANEJO DE PARTICIONES

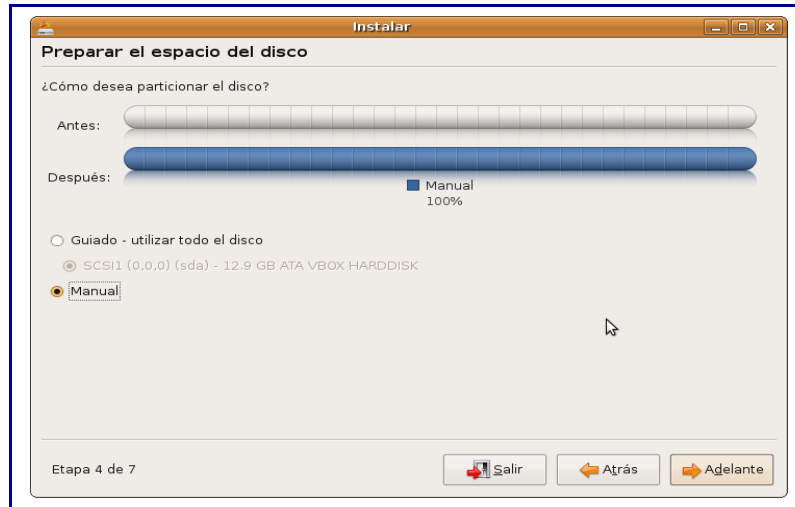


Figura 3.2-9. Preparación del espacio del disco

Tenemos que elegir manual para poder instalar Ubuntu sin problemas y sin ningún fallo, creando así con seguridad la memoria swap y el sistema de archivos ext3, con el directorio raíz (/) como punto de montaje. Para esto, una vez elegido manual, pasamos al siguiente paso, que sería crear las particiones:

III.2.10.-PREPARACIÓN DE PARTICIONES

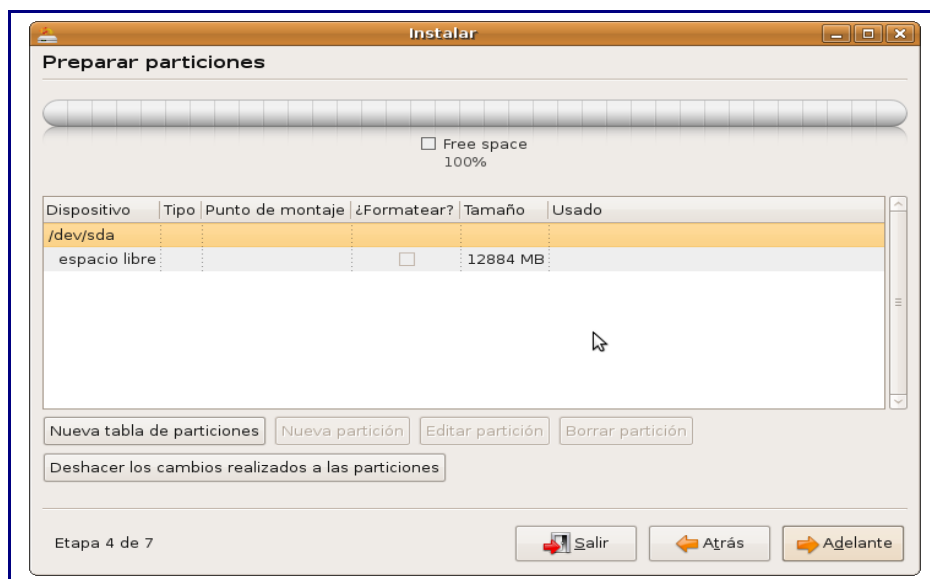


Figura 3.2-10. Preparación de particiones

Como vemos (esto es solo un ejemplo), tenemos un espacio libre de 12,8 GB de disco duro, lo que nos serviría para instalar Ubuntu y trabajar sobre el perfectamente. Para empezar, comenzaremos creando la memoria swap que le asignaremos a nuestro Ubuntu. Para esto, seleccionamos el espacio libre y damos click en Nueva partición, luego vamos al siguiente recuadro para completarlo con los datos del área de intercambio o también llamada memoria swap. Pues bien, el recuadro es el siguiente:

III.2.11.-CREACIÓN DE NUEVA PARTICIÓN

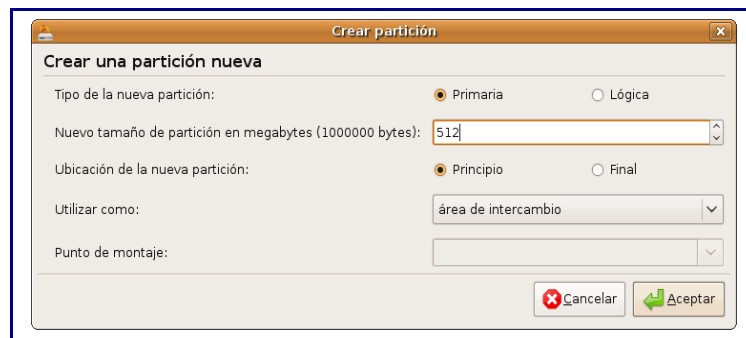


Figura 3.2-11. Creación de nueva Partición

Como vemos, yo se ha asignado una cantidad de 512mb de memoria Swap utilizando un equipo con 1 GB de memoria RAM. A continuación, se muestra una pequeña tabla de cantidades, sobre la elección de la cantidad de swap que debemos asignar a nuestro ordenador y, la tabla de cantidades es la siguiente:

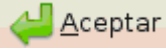
- Con una cantidad de memoria RAM de 4 GB - asignar 2 GB de memoria Swap
- Con una cantidad de memoria RAM de 2GB - asignar 1GB de memoria Swap
- Con una cantidad de memoria RAM de 1GB - asignar 512mb de memoria Swap

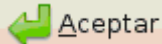
- Con una cantidad de memoria RAM de 512mb - asignar 1GB de memoria Swap
- Con una cantidad de memoria RAM de 256mb - asignar 1,5GB de memoria Swap

Estas son unas buenas cantidades de memoria Swap que le podemos asignar a nuestro Ubuntu pero, no tiene por que ser obligatoriamente dicho y echo si no que la cantidad puede variar dependiendo de el consumo de nuestro ordenador. De hecho, el tema de la memoria Swap, como vemos, no es tan complicado.

III.2.11.1.-Selección de la partición

Luego, tenemos que elegir en el apartado Utilizar como: área de intercambio. También se le puede llamar Swap o Linux-swap, dependiendo de la aplicación elegida para crear las particiones.

Ahora, damos click en  y volver a la ventana de preparar las particiones, eligiendo ahora el otro espacio libre que nos quede, el cuál será nuestro espacio para poder instalar Ubuntu 8.10 Intrepid Ibex. Entonces, damos click en la pestaña que esta libre y creamos una nueva partición y damos click en



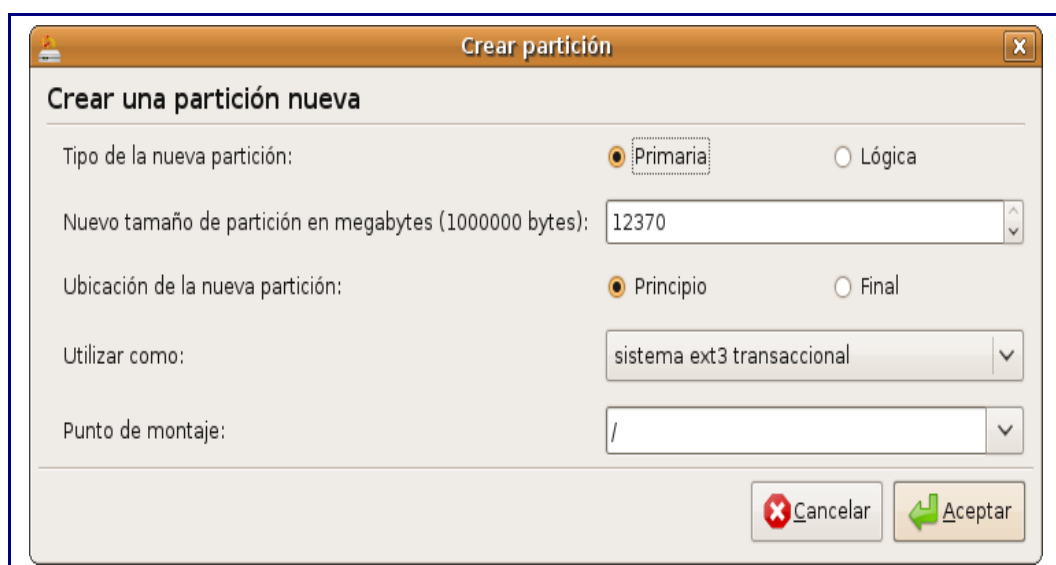


Figura 3.2-12. Asignación de espacio de disco

En esta parte no nos detengamos tanto porque no es demasiado importante. Tan solo tenemos que:

Elegir la cantidad de espacio que le asignaremos al disco duro para poder instalar Ubuntu.

- El tipo de la nueva partición deberá de ser de tipo: "Primaria".
- La ubicación de la nueva partición será: "Principio".
- Utilizaremos la partición como: sistema de ext3 transaccional.
- Y el punto de montaje será el directorio raíz nombrado antes: /

Estos son los datos necesarios para que todo salga en perfectas condiciones. Una vez creadas las particiones, la estructura del disco nos quedaría de la siguiente manera...

III.2.12.-PREPARAR PARTICIONES

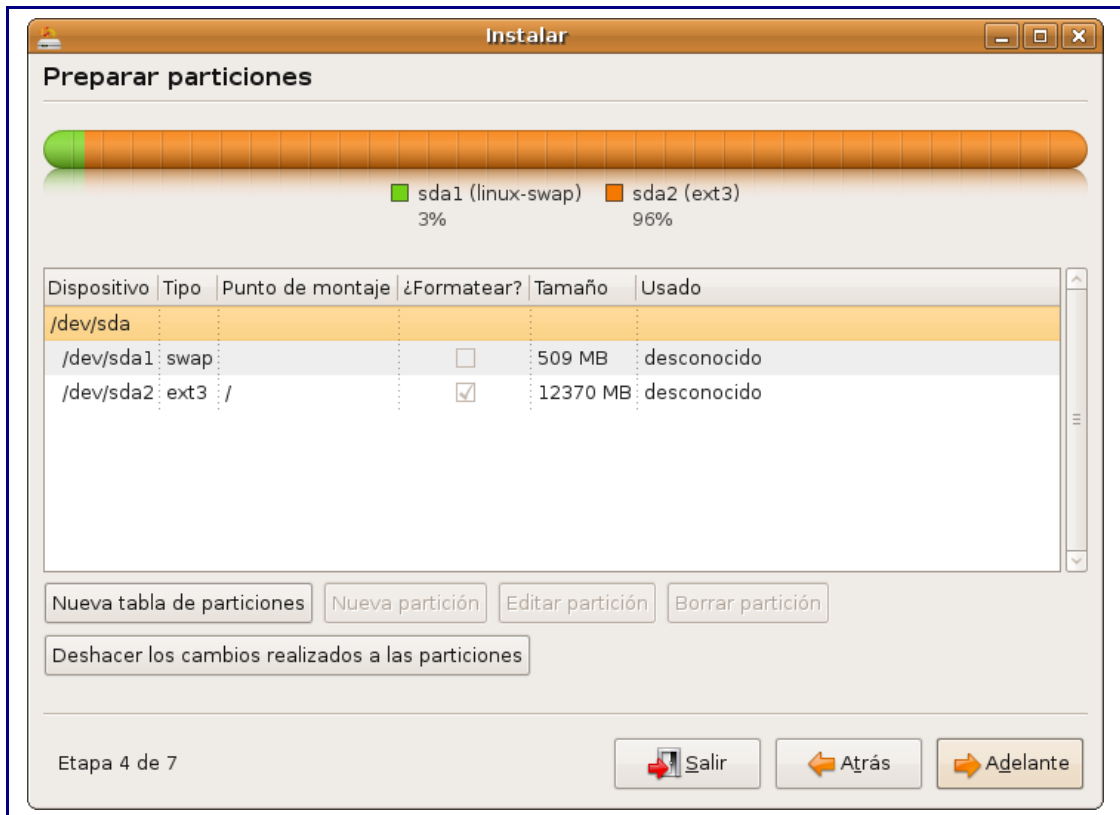


Figura 3.2-13. Preparar particiones

Damos click en **Adelante** para poder pasar al último paso de la instalación, que sería cubrir los datos del nuevo usuario:

III.2.13.-COBERTURA DE DATOS DE USUARIO

Instalar

¿Quién es usted?

¿Cuál es su nombre?

Ubuntu

¿Qué nombre desea usar para iniciar sesión?

ubuntu

Si este equipo va a ser usado por más de una persona, deberá configurar varias cuentas después de la instalación.

Escoja una contraseña para mantener su cuenta segura.

●●●●●● ●●●●●●

Introduzca la misma contraseña dos veces, para así verificar posibles errores de teclado.

¿Cuál es el nombre de este equipo?

intrepid-ibex

Este nombre se usará para hacer visible este equipo a otros equipos de la red.

Entrar automáticamente

Etapa 5 de 7

Salir Atrás Adelante

Figura 3.2-14. Cobertura de los datos del Usuario

Como vemos, un apartado bastante fácil pero necesario. Ahora paso a explicar lo que se necesita en cada pregunta cuestionada en este proceso:

¿Cuál es su nombre?: Aquí escribiremos nuestro nombre.

¿Qué nombre desea usar para iniciar sesión?: Especialmente vale cualquier nombre pero si es muy recomendable que el usuario se escriba en minúsculas.

Escoja una contraseña para mantener su cuenta segura: Aquí solo introduciremos nuestra contraseña.

¿Cuál es el nombre de este equipo?: Se puede poner el que mas nos guste. Solo hay que ponerle un nombre a nuestro ordenador.

Casilla Entrar automáticamente: Esto lo marcaremos si no quisiéramos loguearnos al inicio del sistema. También es muy recomendable desmarcarla por su seguridad.

Pues bien, una vez todo cubierto, damos click en “Adelante” y pasamos al repaso de la instalación para ver si todo está bien:

III.2.14.-INSTALANDO EL SISTEMA EN SI

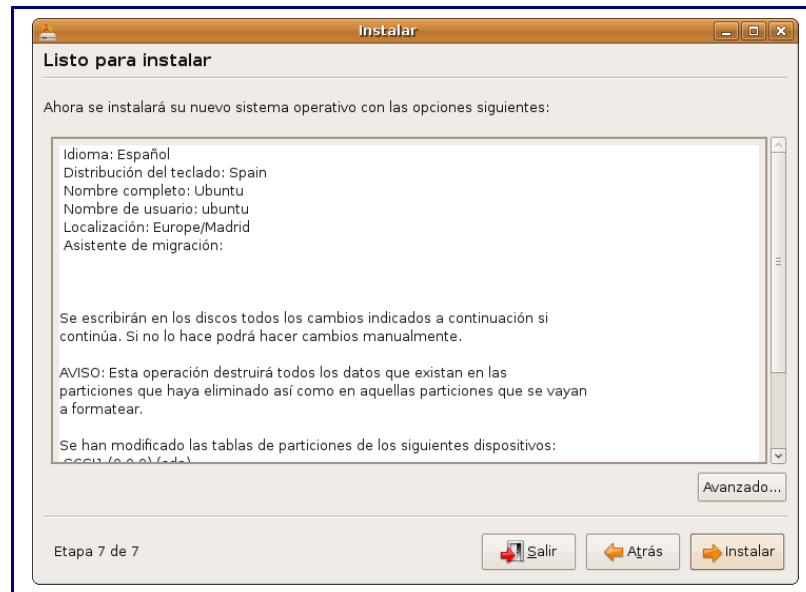



Figura 3.2-15. Instalando el sistema

Pues bien, ya llegamos, la última fase de la instalación. Ahora tan solo nos queda dar click en  y comenzará el proceso de instalación:

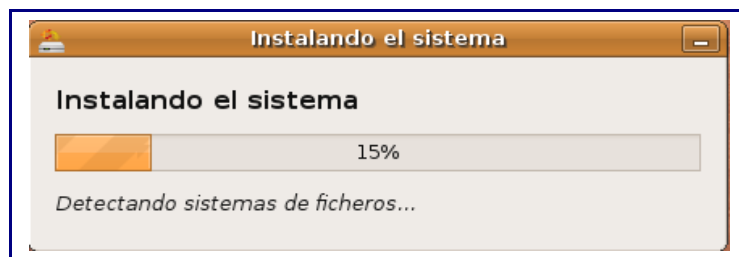


Figura 3.2-16. Instalando el sistema

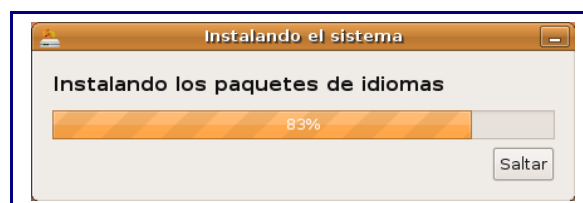


Figura 3.2-17. Instalando el sistema

Y ahora, una vez terminada la instalación, nos saldrá el siguiente recuadro **Reiniciar ahora**, lo cual haremos para poder empezar a utilizar el nuevo Ubuntu 8.10 Intrepid Ibex:

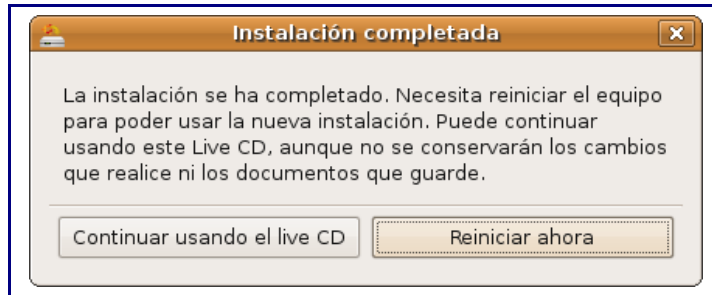


Figura 3.2-18. Fin de la instalación

¡Listo!. Una vez reiniciado, quitamos el CD de Ubuntu y damos click en Enter. La siguiente pantalla será ya la interfaz de Ubuntu.

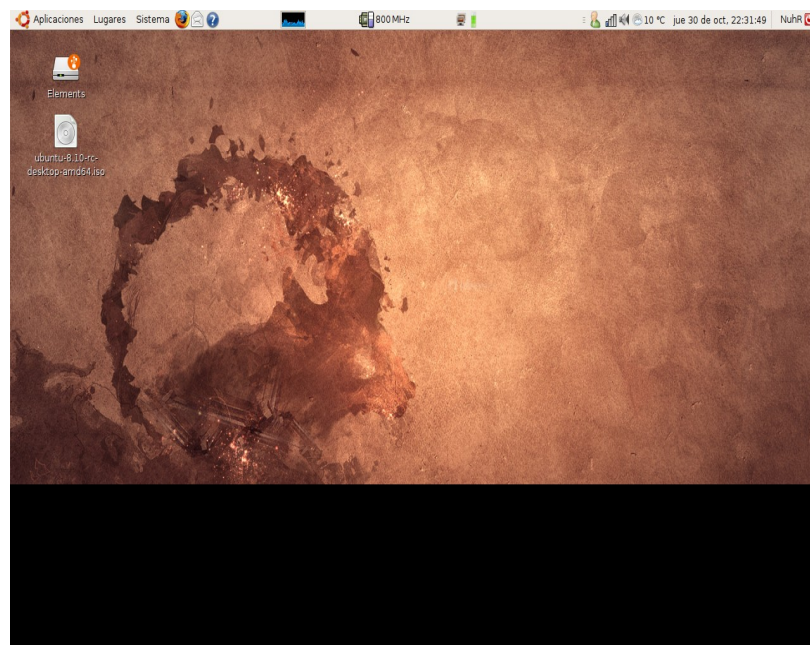


Figura 3.2-19. Escritorio de Ubuntu

Referencia bibliografica:

Tomado de la página web:

http://guia_ubuntu.org/index.php?title=Instalaci%C3%B3n_est%C3%A1ndar

III.3.-INSTALAR GLASSFISH.

Glassfish puede contar con múltiples configuraciones denominadas dominios.

Además, el servidor se puede instalar con 3 perfiles distintos:

Un primer perfil conocido como “Developer” que contiene la consola de administración y nuestras aplicaciones y es el indicado para la típica instalación que hace cada desarrollador en su máquina para depurar código.

Un segundo perfil, denominado “Enterprise” que contiene la DAS o Domain Administration Server y que permite administrar el resto de instancias del dominio.

Un tercer y último perfil que permite crear clústeres de GLASSFISH.

En este tutorial instalaremos el perfil de clúster para poder crearlos posteriormente. Para conseguirlo, desde la misma Terminal, se navegara hasta el nuevo directorio creado:

III.3.1.-PASOS PARA LA INSTALACIÓN

1. Navegamos hasta el nuevo directorio creado:

```
cd glassfish
```

2. Necesitamos asegurarnos de que tenemos java 1.5 o superior para ello:

```
$> java -versión
```

En caso de tener instalado una versión de java inferior se recomienda el siguiente código para instalar

```
$> apt-get install sun-java5-jdk
```

(También se puede usar Automatix2 o Synaptic o lo que se disponga si les resulta más simple)

3. Instalar ant

```
$> apt-get install ant
```

4. Descargarse glassfish

(web oficial) <https://glassfish.dev.java.net/>

5. Copiar el fichero al /opt e instalar glassfish

```
$> cd /opt
```

```
$> java -jar -Xmx256M glassfish-installer-v2.1-b60e-linux-ml.jar
```

(En mi caso dispongo de la versión v2.1-b60e)

6. Reconstruimos el proyecto con ant

```
$> cd glassfish
```

```
$> ant -f setup.xml
```

(Nota. También existe un setup-cluster.xml por si se quiere desplegar el servidor en un cluster).

7. Arrancamos glassfish

```
$> cd glassfish/bin
```

```
$> ./asadmin start-domain domain1
```

Tras esto glassfish se despliega ocupando el puerto 8080 como puerto de publicaciones web y el 4848 para la administración del servidor con un completísimo backend.

8. Acceder al backend de administración

En un navegador ponemos: <http://localhost:4848> (si lo hemos instalado en local), nos aparecerá la pantalla de acceso y por defecto el usuario y el password de glassfish es:

```
user: admin
```

```
password: adminadmin
```

Nota. (Cambiar el password será una de las primeras cosas que tendremos que hacer).

Referencia bibliográfica:

Tomado de la página Web:

http://neosoft.bligoo.com/content/view/201739/Glassfish_Instalacion.html

III.4.-CARGAR UNA APLICACIÓN WEB

Un módulo de aplicación Web está formado por una recopilación de recursos Web como, por ejemplo, páginas JavaServer (JSP), servlets y páginas HTML empaquetadas en un archivo o directorio .WAR (Archivo de aplicación Web).

1. ingresamos a la consola de administración poniendo nuestro usuario y contraseña.
login= admin
login=adminadmin

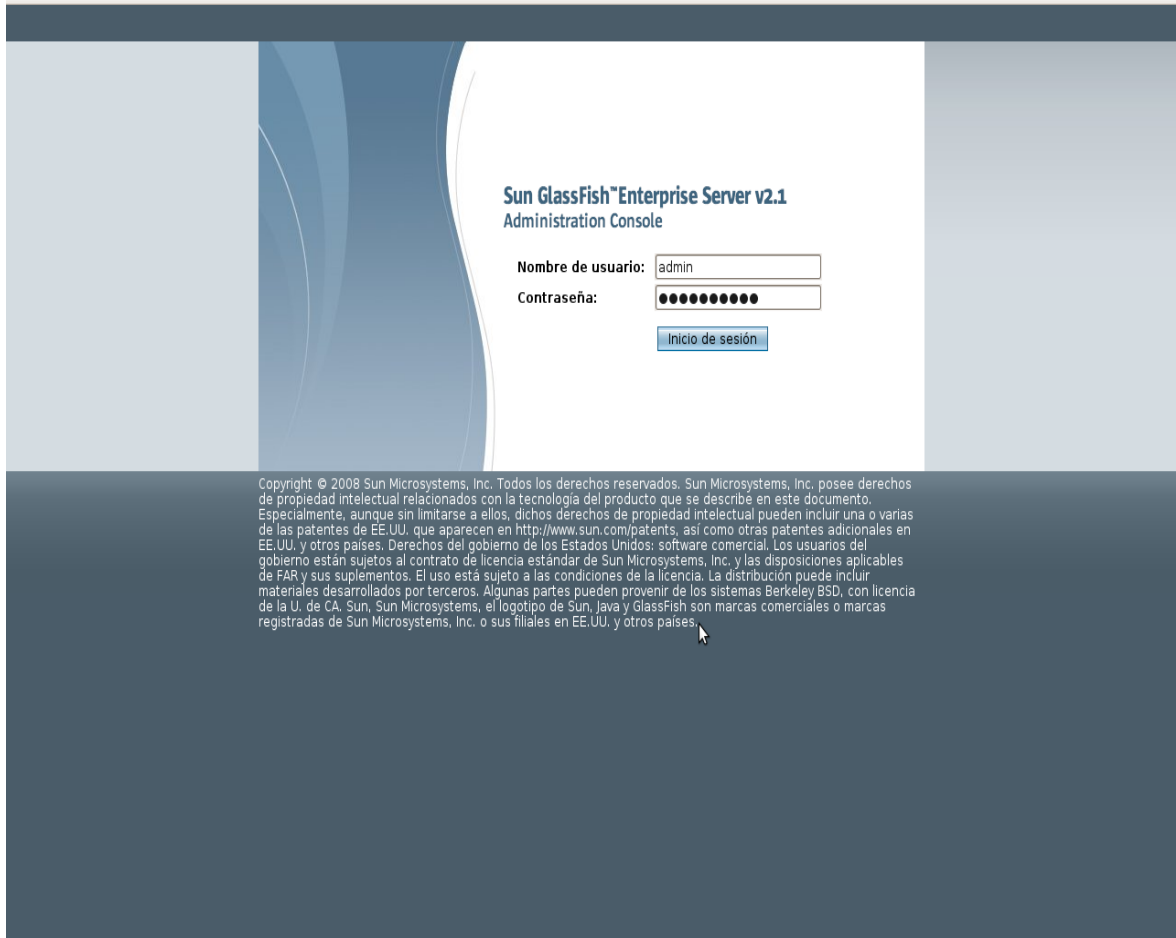


Figura 3.4-1. Ventana de Acceso a Glassfish

2. En el panel de administración damos click en aplicaciones Web.

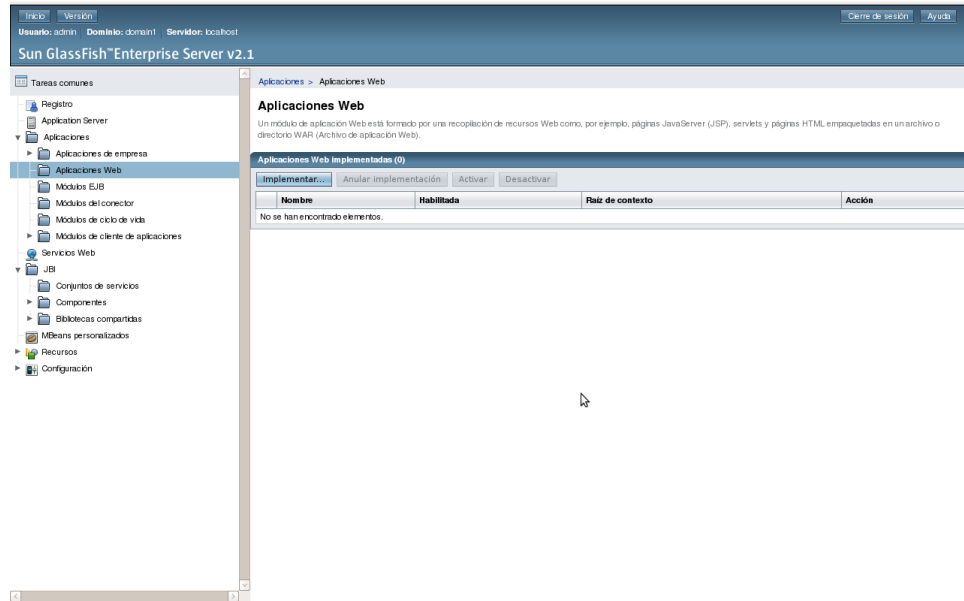


Figura 3.4-2. Manejo de Aplicaciones pasó 1

3. Damos click en implementar para ir a la siguiente página que nos permite cargar una aplicación Web.

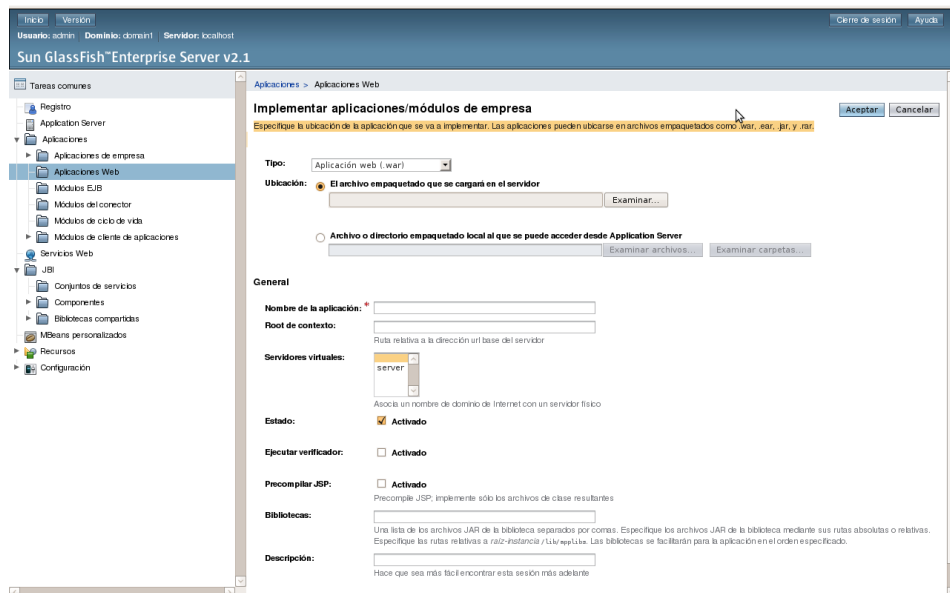


Figura 3.4-3. Manejo de Aplicaciones pasó 2

- Se escoge la opción Archivo o directorio empaquetado local al que se puede acceder desde Application Server y luego damos click en examinar archivos.

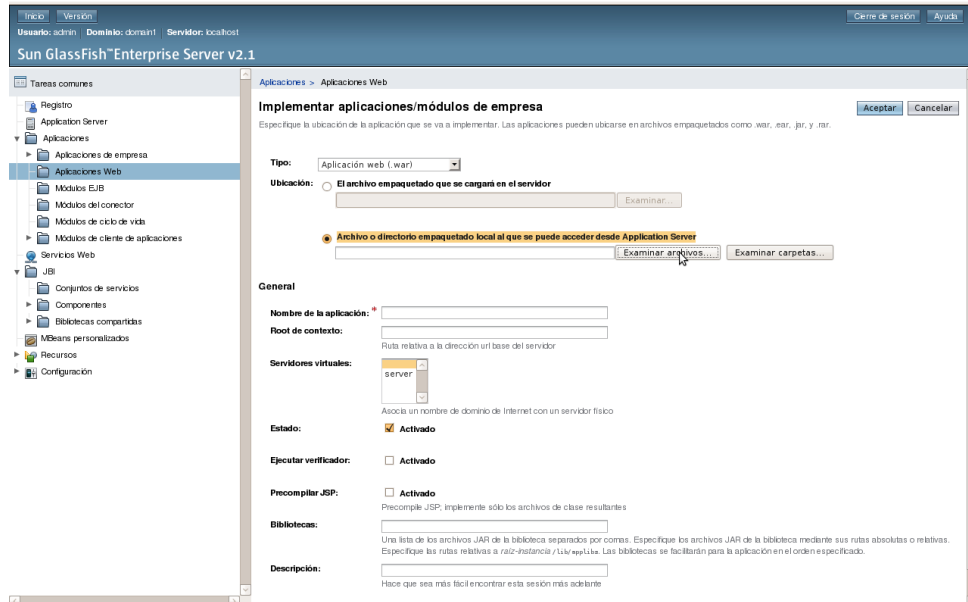


Figura 3.4-4. Manejo de Aplicaciones pasó 3

- Se despliega otra ventana para escoger la dirección donde estén guardado las aplicaciones para cargar en el servidor.

http://localhost:4848/applications/fileChooser.jsf?dirPathId=form:title:sheet1:section1:prop1:

Servidor del navegador

Nombre del servidor: ubuntu

Buscar en: /

Filtro de archivos: *

Para aplicar un valor nuevo en el campo, pulse Intro en ese campo.

Ordenar por: Tamaño (mayor primero)

Nombre	Tamaño	Fecha y hora
lost+found/	16384	15/03/2009 07:35
dev/	14080	17/03/2009 20:08
etc/	12288	18/03/2009 07:34
lib/	12288	15/03/2009 08:24
bin/	4096	15/03/2009 08:29
boot/	4096	15/03/2009 08:48
cdrom/	4096	15/03/2009 07:36
home/	4096	15/03/2009 07:43
host/	4096	17/03/2009 20:38
media/	4096	17/03/2009 20:09
mnt/	4096	20/10/2008 08:27
opt/	4096	16/03/2009 01:06

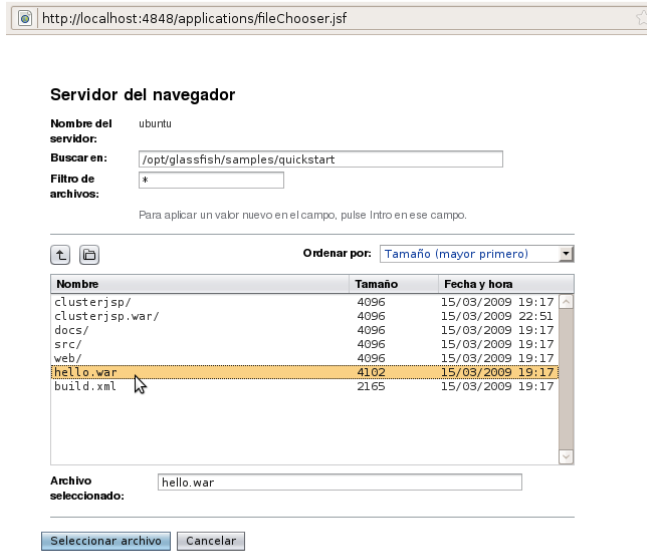
Archivo seleccionado:

Seleccionar archivo Cancelar

Listo

Figura 3.4-5. Manejo de Aplicaciones pasó 4

6. para nuestro ejemplo la aplicación a cargar es la Hello.war, que esta en el directorio opt/glassfish/samples/quickstart donde escogeremos el archivo Hello.war, seguidamente damos click en **Seleccionar archivo** como se muestra en la figura 3.4-6.



Listo

Figura 3.4-6. Manejo de Aplicaciones pasó 5

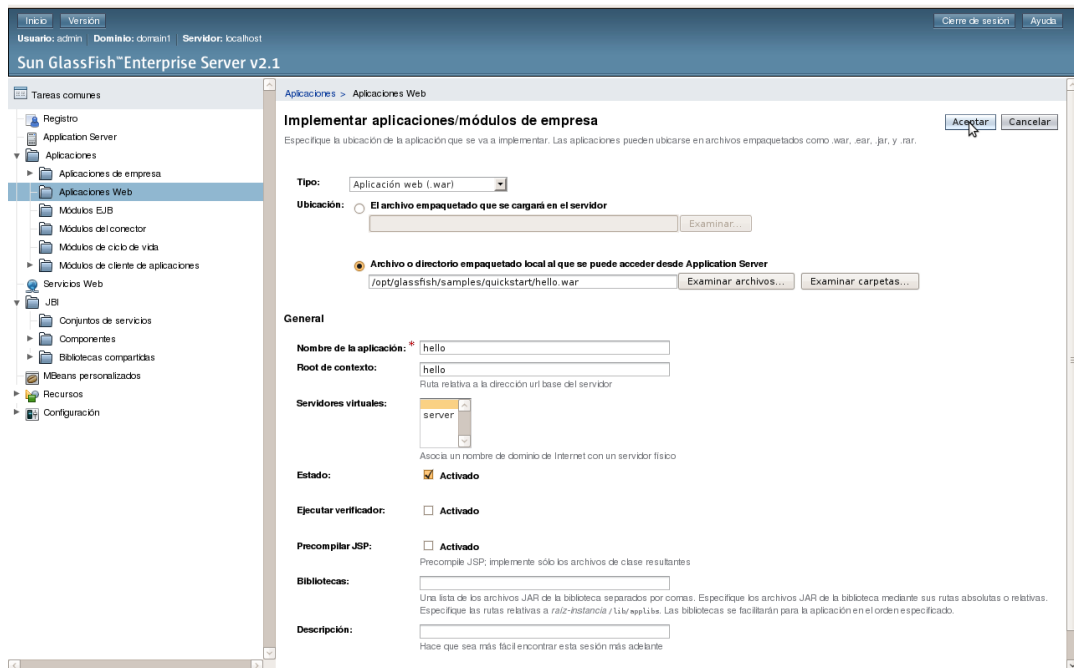



Figura 3.4-7. Manejo de Aplicaciones pasó 6

7. Verificar que este activada la casilla de estado y luego damos click en 
8. Si todo esta bien debería salir la aplicación cargada como se muestra en la figura 3.4-8

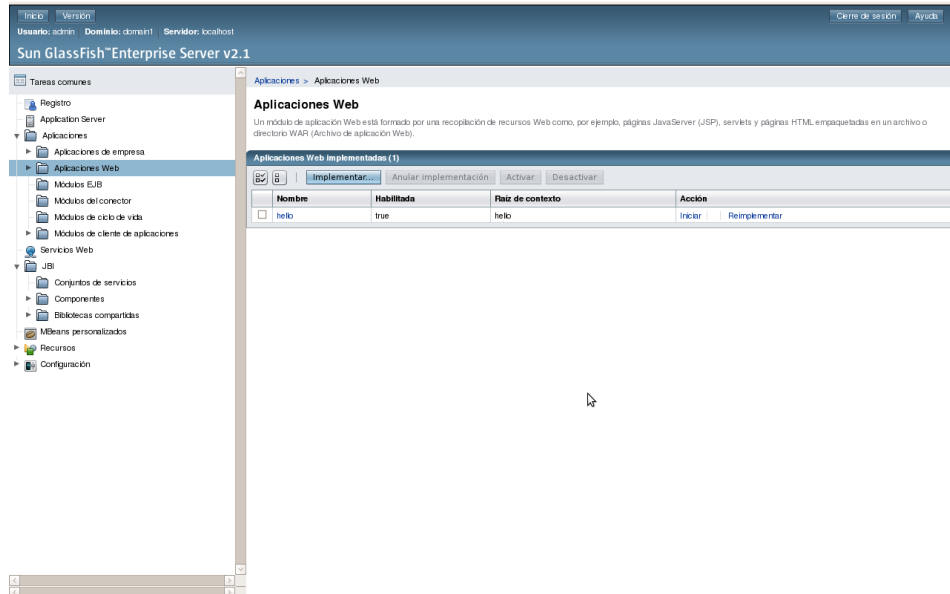


Figura 3.4-8. Manejo de Aplicaciones pasó 7

9. Procedemos a verificar nuestra aplicación dando click en iniciar

10. A continuación podemos ver nuestra aplicación ejecutada.

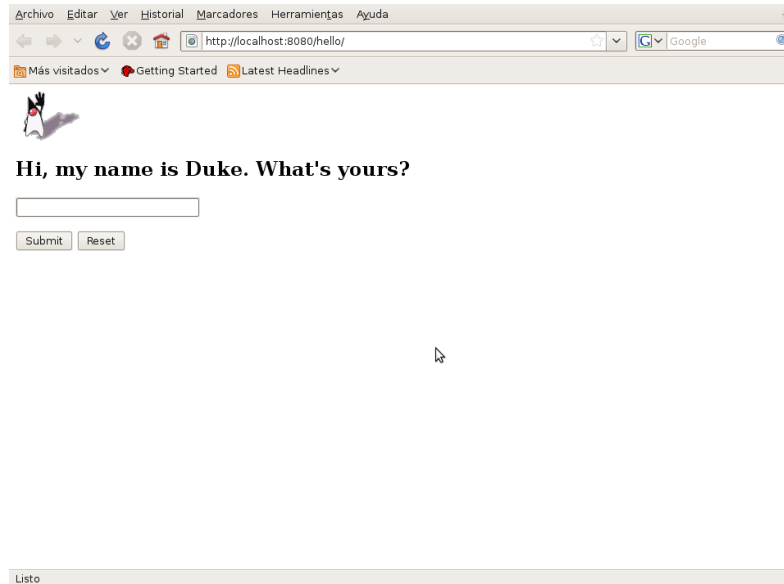


Figura 3.4-8. Aplicación Ejecutada

CAPÍTULO IV

IV.1.-CONCLUSIONES:

El software libre ahorra costos a toda la sociedad y nos brinda facilidades de obtención en cualquier lugar donde lo necesitemos.

El servidor de aplicaciones J2EE GLASSFISH esta bien soportado y es de fácil uso e implementación.

En cada distribución de Linux brinda las facilidades para ser implementado el servidor de aplicaciones J2EE GLASSFISH.

Cada uno de nosotros somos responsables de los cambios que se den dentro del ámbito del desarrollo de software y administración de servicios de red y en común de los servidores.

IV.2.-RECOMENDACIONES:

Se recomienda brindar mas apoyo al desarrollo de software dentro de las instituciones de educación superior para contribuir con el desarrollo del software en el Ecuador.

Se recomienda el uso del software libre dentro de los estamentos estatales para ahorrar costos al estado y apoyar al desarrollo educacional.

Es necesario volcarnos a la investigación y difusión de nuevos proyectos software que contribuyan a facilitar la vida de la humanidad.

REFERENCIAS BIBLIOGRÁFICAS:

www.monografias.com

http://guia_ubuntu.org/index.php?title=Instalaci%C3%B3n_est%C3%A1ndar

<http://www.tu-chemnitz.de/docs/lindocs/RH9/RH-DOCS/rhl-ig-x86-es-9/index.html>

http://neosoft.bligoo.com/content/view/201739/Glassfish_Instalacion.html

<http://www.infosintesis.net/apensintesis/glassfish/mapa/v2/crearPoolConexiones.html>

<http://www.infosintesis.net/apensintesis/glassfish/mapa/v2/configurarRecursosJDBCpoolConexiones.html>

http://weblogs.javahispano.org/lasterra/es/entry/glassfish_clustering

GLOSARIO

- SOA** Arquitectura Orientada a Servicios
- BPEL** Business Process Execution Language
- FMS** Fortran Monitor System
- UCP** Unidad Central de Procesamiento
- DOS** Disk Operating System (sistema operativo de disco).
- MacOS** Macintosh Operating System (Sistema Operativo de Macintosh)
- THE** Technische Hogeschool, Eindhoven
- RCP** Remote Copy
- Hadb** High Availability Database Managemen
- RPM** Package Manager (originalmente llamado Red Hat Package Manager)
- RAID** Redundant Array of Independent Disks
- GRUB** Grand Unified Bootloader
- LILO** Linux Loader
- UTC** Universal Time Coordinated
- MPL** Mozilla Public License

DECLARACIÓN DE AUTENTICIDAD Y RESPONSABILIDAD EXPRESADA

Quien suscribe CBOP. DE COM. Villagomez Cevallos Jinson Oswaldo, portador de la C.I. 091728148-7, libre y voluntariamente declaro que el presente tema de investigación: **“ELABORAR EL MANUAL PARA LA CONFIGURACIÓN DE UN SERVIDOR GLASSFISH, UTILIZANDO EL SISTEMA OPERATIVO GNU/LINUX”**, su contenido, ideas, análisis, conclusiones, y propuestas son auténticos y personales.

En tal virtud son para efectos legales y académicos que se desprenden de la presente tesis es y será de mi exclusiva responsabilidad legal y académica, como autor de esta tesis de grado.

Para el derecho intelectual del autor, de la información obtenida y utilizada a manera de bibliografía se muestra en el documento la referencia de donde fue obtenida.

El resto de la información de este documento es soporte intelectual adquirido mediante las practicas realizadas y lo aprendido en el transcurso de mi carrera universitaria.

Atentamente;

Jinson O. Villagomez C.

EL AUTOR

DECLARACIÓN DE AUTENTICIDAD Y RESPONSABILIDAD EXPRESADA

Quien suscribe CBOS. DE I. Torres Collaguazo José Lizandro, portador de la C.I. 1715277370, libre y voluntariamente declaro que el presente tema de investigación: **“ELABORAR EL MANUAL PARA LA CONFIGURACIÓN DE UN SERVIDOR GLASSFISH, UTILIZANDO EL SISTEMA OPERATIVO GNU/LINUX”**, su contenido, ideas, análisis, conclusiones, y propuestas son auténticos y personales.

En tal virtud son para efectos legales y académicos que se desprenden de la presente tesis es y será de mi exclusiva responsabilidad legal y académica, como autor de esta tesis de grado.

Para el derecho intelectual del autor, de la información obtenida y utilizada a manera de bibliografía se muestra en el documento la referencia de donde fue obtenida.

El resto de la información de este documento es soporte intelectual adquirido mediante las practicas realizadas y lo aprendido en el transcurso de mi carrera universitaria.

Atentamente;

José L. Torres C.

EL AUTOR

ESCUELA POLITÉCNICA DEL EJÉRCITO SEDE LATACUNGA

CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

Este proyecto fue elaborado por:

José L. Torres C.
CI: 1715277370

Jinson O. Villagomez C.
CI: 0917281487

El Coordinador de Carrera

El Secretario Académico

Ing. Edison Espinoza

Dr. Rodrigo Vaca

Latacunga, Marzo del 2009