Índice

1 Revisión de conceptos	2
2 Creación y configuración de dominio	2
2.1 Arranque de servidores	3
2.2 Definición de servidores dentro del dominio	3
2.3 Definición de máquinas	18
3 Administración desde línea de comandos	23

1. Revisión de conceptos

Vamos a revisar algunos de los conceptos ya vistos con anterioridad. Dentro de un dominio debemos disponer de un servidor de administración. Este servidor se encargará, entre otras, de las siguientes tareas:

- Mantenimiento del fichero de log. Todos los mensajes de los distintos servidores así como los propios del dominio se almacenarán en el fichero log del dominio.
- Leer y escribir en el fichero *config.xml*. Este fichero contiene toda la configuración del dominio (nombre de los servidores, puertos, aplicaciones disponibles, etc.).
- Proporciona información de configuración cuando arrancan los servidores administrados. Por ello, el servidor de administración debe estar en marcha cuando arranquen el resto de servidores. Una vez arrancados todos podemos parar el de administración.
- Permite monitorizar el comportamiento de los servidores y cambiar su configuración.



2. Creación y configuración de dominio

Como ya hemos visto anteriormente, WebLogic incorpora una herramienta para crear el dominio, definiendo los servidores que formarán parte del dominio. Si necesitamos añadir nuevos servidores veremos más adelante que podemos hacer uso de la consola de administración.

2.1. Arranque de servidores

Una vez creado el dominio junto con sus servidores, cuando arrancamos un servidor nos va a pedir el nombre y la contraseña del usuario de administración. El servidor de administración se arranca simplemente ejecutando *startWebLogic.cmd*. Para los servidores administrados debemos indicar la dirección del servidor de administración:

bin/startManagedServer.cmd servidor2 http://localhost:7001

donde *servidor2* es el nombre del servidor que queremos arrancar y a continuación indicamos la dirección y el puerto de escucha del servidor de administración.

2.2. Definición de servidores dentro del dominio

Un servidor de aplicaciones no es más que una instancia de la clase *weblogic.Server*. Vamos a ver cómo podemos crear nuevos servidores desde la consola de administración. Pinchamos en la opción *servers* (desde la estructura de dominio, dentro de *environment* o desde la página inicial) y nos aparecerá una página como la siguiente donde tenemos los servidores definidos en nuestro dominio y podemos definir nuevos.

A se confi	rver is an instance of WebL iguration.	ogic Server th	at runs in its o	wn Java Virtua	l Machine (J∨N	4) and has its owr
۲his	page summarizes each se	rver that has b	peen configure	d in the currer	nt WebLogic S	erver domain.
DO	ustomize this table					
Ser	vers					
Ser Cli	vers ck the <i>Lock & Edit</i> buttor	in the Change	e Center to ac	tivate all the b	uttons on this	page.
Ser Cli	vers ck the <i>Lock & Edit</i> buttor	i in the Change	e Center to ac	tivate all the b	uttons on this	page.
Ser Cli	vers ck the <i>Lock & Edit</i> buttor ew Clone Delete	in the Change	e Center to ac	tivate all the b	uttons on this	page. f 2 Previous Ne
Ser Cli	vers ck the <i>Lock & Edit</i> buttor ew Clone Delete Name &	i in the Change Cluster	e Center to act	tivate all the b Sl State	uttons on this nowing 1 - 2 o Health	s page. f 2 Previous Ne Listen Port
Ser Cli	vers ck the <i>Lock & Edit</i> buttor ew Clone Delete Name & AdminServer(admin)	in the Change	e Center to act	tivate all the b Sl State RUNNING	uttons on this nowing 1 - 2 o Health OK	f 2 Previous Ne Listen Port 7001

Podemos hacer tres cosas: crear un nuevo servidor, clonar o borrar uno ya existente (seleccionándolo antes). Recordad que debemos pinchar en *Lock and Edit* para editar.

Si pinchamos en crear uno nuevo (New) nos aparece la siguiente ventana:

Create a New Server	
Back Next Finish	Cancel
Server Properties The following properties wil	Il be used to identify your new server.
* Indicates required fields	
What would you like to	name your new server?
*Server Name:	Servidor3
Where will this server li	sten for incoming connections?
Server Listen Address:	localhost
Server Listen Port:	8001
Should this server belor	ng to a cluster?
[⊙] No, this is a stand	alone server.
C _{Yes} , create a new	cluster for this server.
Back Next Finish	Cancel

Aquí damos el nombre, dirección y puerto de escucha. Dejamos para más adelante lo referente al cluster. El servidor nos aparecerá en la lista de servidores y podemos pasar a configurarlo. Esto lo podemos hacer para cualquier servidor en nuestro dominio. Pinchamos sobre el nombre del servidor y nos aparecerá esta ventana:

Configuration Protocols	Logging Debug	Monitorir	ng Control	Deployments	Services	Security	Notes			
General Cluster S	ervices Keystore	s SSL	Deployment	Migration	Tuning	Overload	Health Monitoring	Server Sta		
Save										
Use this page to conf	igure general feati	ures of this	server such a	as default netv	vork comm	nunications				
View JNDI Tree 🔎										
Name:	servidor3		An alph	anumeric nam	ne for this	server insta	ance. More Info			
Machine:	(None)	J	The Web to run.	Logic Server More Info	host comp	uter (mach	nine) on which this s	erver is mea		
Cluster:	(Standal	one) 🗾	The clus belongs	The cluster, or group of WebLogic Server instances, to which this server belongs. More Info						
🐐 Listen Address:	localhost		The IP a connect	The IP address or DNS name this server uses to listen for incoming connections. More Info						
🐐 🔽 Listen Port Ena	bled		Specifie (non-SS	Specifies whether this server can be reached through the default plain-text (non-SSL) listen port. More Info						
Listen Port:	8001		The definition	The default TCP port that this server uses to listen for regular (non-SSL) incoming connections. More Info						
🖣 🥅 SSL Listen Port	Enabled		Indicate port. M	Indicates whether the server can be reached through the default SSL lister port. More Info						
🕼 SSL Listen Port:	7002		The TCP More Inf	/IP port at whi	ch this ser	ver listens	for SSL connection	requests.		
🦉 🥅 Client Cert Prox	y Enabled		Specifie	s whether the neader. More	HttpClust	erServlet p	roxies the client cer	tificate in a		

Las opciones disponibles son:

- Name: el nombre del servidor. Debe ser único en el dominio.
- **Machine**: si ya tenemos definidas máquinas podemos asignar este servidor a una de ellas. Esta operación la podemos hacer más adelante.
- Cluster: ídem al anterior pero con los clusters.
- Listen Address: la dirección (IP o DNS) de escucha del servidor.
- Listen Port Enabled: al activar esta opción permitimos la escucha por un puerto.
- Listen Port: el puerto de escucha del servidor. Dos servidores en la misma máquina no pueden tener el mismo puerto de escucha.
- **SSL Listen Port Enabled:** nos permite activar el puerto de escucha con protocolo seguro.
- **SSL Listen Port**: El puerto de escucha seguro.
- **Client Cert Proxy Enabled**: con esta opción especificamos que se envíe un encabezado especial con el certificado del cliente.
- Java Compiler: indica el compilador de java a emplear en todas las aplicaciones de este servidor.

Java Compiler:	javac	The Java compiler to use for all applications hosted on this server that need to compile Java code. More Info
♥ Advanced		
툍 厂WebLogic Plug-In E	Enabled	Specifies whether this server uses the proprietary WL-Proxy-Client-IP header, which is recommended if the server instance will receive requests from a proxy plug-in. More Info
🐐 Prepend to classpath	1:	The options to prepend to the Java compiler classpath when compiling Java code. More Info
🐐 Append to classpath	:	The options to append to the Java compiler classpath when compiling Java code. More Info
Extra RMI Compiler Options:		The options passed to the RMIC compiler during server-side generation.
Extra EJB Compiler Options:		The options passed to the EJB compiler during server-side generation. More Info
External Listen Address:		The external IP address or DNS name for this server. More Info
4 Local Administration Port Override:	9002	Overrides the domain-wide administration port and specifies a different listen port on which this server listens for administrative requests. Valid only if the administrative channel is enabled for the domain. More Info
Startup Mode:	Running	The state in which this server should be started. If you specify STANDBY, you must also enable the domain-wide administration port. More Info
4 JDBC LLR Table Name		The table name for this server's Logging Last Resource (LLR) database table(s). WebLogic Server creates the table(s) and then uses them during transaction processing for the LLR transaction optimization. This setting must be unique for each server. The default table name is WL_LLR_SERVERNAME. More Info

Sin pinchamos en opciones avanzadas nos aparecen las opciones mostradas en la anterior figura (comentamos las más utilizadas):

- **Startup Mode:** Podemos arrancar un servidor en dos modos: RUNNING en el cual se ejecuta normalmente y STANDBY en el que permite comandos del servidor de administración pero no reponde a peticiones de sus aplicaciones.
- **Prepend/append to classpath:** directorios a añadir antes o después del CLASSPATH actual.
- External Listen Address: es la dirección de escucha tal como se ve desde fuera de nuestro sistema.

Nos saltamos las siguientes solapas hasta Health Monitoring:

- Auto Restart: indica que el servidor se intentará reiniciar después de un fallo.
- Auto Kill if Failed: si la activamos, el *node manager* matará al servidor si este ha tenido algún fallo.
- Max Restarts Within Interval: número de veces que se intentará reiniciar al servidor durante un intervalo de tiempo predefinido.

• **Restart Delay Seconds:** numero de segundos a esperar antes de intentar reiniciar.

					1				
ieneral Cluster Services	Keystores	SSL D	eployment	Migration	luning	Overload	Health M	4onitoring	Server Sta
Save									
WebLogic Server provides a Server domain. Selected sub Use this page to configure th application (optional) checks stops and restarts the serve	self-health m osystems wit ne frequency the server's r if the serve	onitoring ca hin each se of this sen health staa r reaches t	apability to rver monitiver's autom te. You can he "failed" l	improve the or their healt hated health also use this health state.	reliability h status b checks, a s page to s	and availab based on cri nd the frequ specify whet	ility of serve teria specifi iency with w her Node M	ers in a WebL c to the subs /hich the Noo 1anager auto	ogic system. le Manager matically
🔽 Auto Restart			Specifi crashe	es whether tl s or otherwis	ne Node N e or goes	1anager car down unexp	automatic	ally restart th ore Info	nis server if i
⊤Auto Kill If Failed			Specifi health	es whether tl state is faile	ne Node N d. <mark>More Ir</mark>	1anager sho fo	uld automa	atically kill thi	s server if it:
Max Restarts Within Interval:	2		The nui interva	mber of time I specified in	s that the RestartInt	Node Mana erval. <mark>More</mark>	ager can re: Info	start this ser	ver within th
Health Check Interval:	180		The number of seconds that defines the frequency of this server's self-healt monitoring. The server monitors the health of it's subsystems every HealthCheckIntervalSeconds and changes the Server's overall state if required. More Info						s self-health ery ate if
Restart Delay Seconds:	0		The nui server.	mber of seco More Info	onds the N	lode Manag	er should w	ait before re	starting this
Sava									

Pasamos a la solapa de protocolos. En la subsolapa de HTTP tenemos las siguientes opciones:

- **Post Timeout/Max Post Size:** se utiliza para prevenir los ataques DoS. Un ataque DoS (*Denial of Service*: denegación de servicio) tiene como consecuencia la caída del servidor. Lo suelen utilizar *hackers* para echar abajo un sistema. El ataque puede ser llevado a cabo de dos maneras:
 - Por el envío de sucesivos paquetes incompletos. Cuando enviamos un paquete incompleto, el servidor queda esperando el resto del paquete. El envío de muchos de estos paquetes puede colapsar el servidor. Para combatir este tipo de ataque debemos configurar **Post Timeout** que es el tiempo de espera máximo para recibir el siguiente paquete.
 - Por el envío de un paquete de datos demasiado grande que sobrepasa la memoria del sistema. La protección que podemos realizar es indicar en la opción **Max Post Size** el tamaño máximo de paquete a recibir. Por defecto está en -1 que indica que no existe limitaciónd e tamaño.
- Enable Keepalives: permite el mantenimiento de una conexión permanente al servidor.

- Duration: tiempo transcurrido antes de cerrar una conexión inactiva.
- HTTPS Duration: ídem conexión segura.

Configuration	Protocols	Logging	Debug	Monitoring	Control	Deployments	Services	Security	Notes
General H	ттр јсом	1 IIOP	Channel	s					
Save									

Web-based clients communicate with WebLogic Server using HTTP (HyperText Transfer Protocol).

Use this page to define the HTTP settings for this server.

Default WebApp Context Root:	
🚑 Post Timeout:	30
🍇 Max Post Size:	-1
🏭 🗁 Enable Keepaliv	25
🍇 Duration:	30
4 HTTPS Duration:	60
🚑 Frontend Host:	
Frontend HTTP Port:	0
Frontend HTTPS	0
4 🗍 🖓 WAP Enabled	

CAccept Context	Path In Get Real Path	
HTTP Max Message Size:	-1	
Enable Tunnelin		
Tunneling Client Ping:	45	
Tunneling Client Timeout:	40	

En la solapa *Logging* podemos configurar las opciones del fichero que contendrá los mensajes del servidor. En la siguiente figura se muestran los datos más genéricos, ya explicados en el apartado del log de dominio. La siguiente solapa, *Rotation Type*, permite modificar el tipo de rotación del fichero log. Las opciones también se detallaron anteriormente. La última opción, *Rotate file rotation on startup*, indica que nada más arrancar el servidor se inicializa el fichero log.

onfiguration Protocols	Logging	Debug	Monitoring	Control	Deployments	Services	Security	Note
General HTTP								
Save								
Use this page to define t	the gene	ral loggin	g settings fo	r this serv	/er.			
🝇 Log file name:	logs/se	vidor3.lo	g					
— Rotation ———								
Rotation type:	By Size	•						
Maximum file size:	500							
Begin rotation	00:00							
time:								
Rotation interval:	24							
∏Limit number of I	retained	files						
Files to rotain								
Flies to retain.	7							
Log file rotation directory:								
🖲 🗔 Detete leg file en								

Las opciones avanzadas del logging nos permiten monitorizar los mensajes de error, según el tipo y el nivel de seguridad: *Stdout severity threshold* permite cambiar el nivel de error que muestra los mensajes. Por defecto se mostrará un mensaje si existe un error. Podemos bajar o subir el nivel de error para que muestre un mensaje, por ejemplo, cuando existe una advertencia.

▼ Advanced	
Logging implementation:	JDK _
— Message destinat	ion(s)
Log file :	
Severity level:	Debug
Filter:	None
∏Redirect stdout	logging enabled
Standard out :	
Severity level:	Notice
Filter:	None -
Domain log broadcas	ster :
Severity level:	Notice
Filter:	None -
Memory buffer :	
Severity level:	Debug
Filter:	None -

En *Debug* se nos permite seleccionar qué opciones (servicios, elementos, etc.) de nuestro sistema y servidor queremos que se monitorice. Advertencia: hay que tener mucho cuidado con seleccionar JDBC: el nivel de información puede desbordar nuestro disco.

onfig	uration	Protocols	Logging	Debug	Monitoring	Control	Deployments	Services	Security	Note
lse th Debu	nis page ti ug settin	o define de igs for th	ebug settii iis Servei	ngs for th r	nis server.					
Ena	ible Di	isable	Clear				Showir	ig 1 - 1 of 1	. Previous	Next
Г	Debug S	Scopes a	nd Attrib	utes 🚕				Sta	ite	
	⊞ default	2						Disabled		
Г	🗆 weblog	ic						Disabled		
Г	⊞ prot	ocol						Disabled		
Г	⊞ tran	saction						Disabled		
Г	⊞ iiop							Dis	abled	
Г	⊞ core							Disabled		
Г	⊞ depl	оу						Disabled		
Г	⊞ conr	nector						Dis	abled	
Г	⊞ man	agement						Dis	abled	
Г	⊞ defa	ult						Dis	abled	

En *Monitoring*, pinchamos en las opciones avanzadas y nos proporciona toda la información tanto del servidor de aplicaciones como del sistema operativo y Java (el servidor tiene que estar activo para que nos proporcione esta información).

Settings for AdminServer	
Configuration Protocols Log General Health Channel	gging Debug Monitoring Control Deployments Services Security Not els Performance Threads Timers Workload Security Default Sto
This page provides genera	al runtime information about this server.
State:	RUNNING
ActivationTime:	Thu Mar 08 09:22:35 CET 2007
Weblogic Version:	WebLogic Server 9.2 MP1 Sun Jan 7 00:56:31 EST 2007 883308
Java Vendor:	Sun Microsystems Inc.
Java Version:	1.5.0_06
OSName:	Windows XP
OSVersion:	5.1
JACC Enabled:	false

La opción *Health* nos muestra el funcionamiento de distintos servicios. Para cada uno nos proporciona información indicando si algo ha fallado.

and a second sec		Jointi of L	epioymenus	Services	Security	Note	•
eneral Health Channels Performance	Threads	Timers	Workload	Security	Default	Store	JMS
page allows you to monitor health information	n for this s	server.					
Server Health: OK							
Reason: (No value specified)							
Health information details			sł	nowing 1 - 1	10 of 21	Previou	s Ne
Health information details Subsystem 🗞	Health	Reason	sł	nowing 1 - 1	10 of 21	Previou	s N€
Health information details Subsystem	Health OK	Reason	sł	nowing 1 - 1	10 of 21	Previou	is Ne
Health information details Subsystem AdminServer.jms AdminServer.saf	Health OK OK	Reason	Sł	nowing 1 - 1	10 of 21	Previou	s Ne
Health information details Subsystem AdminServer.jms AdminServer.saf bea_wls_deployment_internal(Application)	Health OK OK OK	Reason	Sł	nowing 1 - 1	10 of 21	Previou	s N€
Health information details Subsystem AdminServer.jms AdminServer.saf bea_wls_deployment_internal(Application) bea_wls_diagnostics(Application)	Неаlth ОК ОК ОК ОК	Reason	Sł	nowing 1 - 1	10 of 21	Previou	s∣Ne
Health information details Subsystem AdminServer.jms AdminServer.saf bea_wls_deployment_internal(Application) bea_wls_diagnostics(Application) bea_wls_internal(Application)	 Health ОК ОК ОК ОК ОК 	Reason	Sł	nowing 1 - 1	10 of 21	Previou	s Ne
Health information details Subsystem AdminServer.jms AdminServer.saf bea_wls_deployment_internal(Application) bea_wls_diagnostics(Application) bea_wls_internal(Application) bea_wls_management_internal2(Application)	Health OK OK OK OK OK OK OK	Reason	Sł	nowing 1 - 1	10 of 21	Previou	is Ne
Health information details Subsystem AdminServer.jms AdminServer.saf bea_wls_deployment_internal(Application) bea_wls_diagnostics(Application) bea_wls_internal(Application) bea_wls_management_internal2(Application) bea_wls_remote_deployer(Application)	Неаlth ОК ОК ОК ОК ОК ОК ОК	Reason	Sł	nowing 1 - 1	10 of 21	Previou	IS Ne
Health information details Subsystem AdminServer.jms AdminServer.saf bea_wls_deployment_internal(Application) bea_wls_diagnostics(Application) bea_wls_internal(Application) bea_wls_management_internal2(Application) bea_wls_remote_deployer(Application) consoleapp(Application)	Неаlth ОК ОК ОК ОК ОК ОК ОК ОК	Reason	Sł	nowing 1 - 3	10 of 21	Previou	s Ne
Health information details Subsystem AdminServer.jms AdminServer.saf bea_wls_deployment_internal(Application) bea_wls_diagnostics(Application) bea_wls_internal(Application) bea_wls_management_internal2(Application) bea_wls_remote_deployer(Application) consoleapp(Application) JDBC	Неаlth ОК ОК ОК ОК ОК ОК ОК ОК	Reason	Sł	nowing 1 - 1	10 of 21	Previou	IS Ne

Workload nos indica el trabajo realizado y el pendiente por realizar de algunas clases.

orniguration Protocols Logging Debug Monitorin	ig control Deployments	Services	Security Not	es	
General Health Channels Performance Thre	ads Timers Workload	Security	Default Stor	e JMS	Л
his page displays statistics for the work managers, co Work Managers	nstraints, and request class	es that are	configured for	this ser	ver.
Name	Pending Requests	Co	mpleted Req	uests	
direct	0	0			
weblogic.kernel.System	0	101	.94		
weblogic.logging.DomainLogBroadcasterClient	0	14			
weblogic.admin.RMI	0	26			
JmsAsyncQueue	0	0			
JmsDispatcher	0	0			
weblogic.logging.LogBroadcaster	0	126	l .		
weblogic.kernel.Non-Blocking	0	0			
weblogic.Rejector	0	0			
SafDispatcher	0	0			
WatchManagerEvents	0	0			
weblogic.kernel.Default	0	150	11		
	1				

La solapa *Security* nos permite monitorizar los intentos de entrada al sistema inválidos y el número de usuarios bloqueados. Puede servir para monitorizar intentos de acceso inválidos.

	THOLOCOID	Logging	Debug	Moniconing	Control	Deployments	Services	Security	Note
General He	alth Cha	nnels P	erformar	nce Thread	s Timers	Workload	Security	Default	Store
This page al	lows you to	monitor	user lock	out manager	ment stati	stics for this se	rver.		
Total Inva	lid Logins	:: 0							
User Lock Count:	cout Total	0							
Total Logi while Loc	n Attemp ked:	ts 0							
Invalid Lo	gins High	i: 0							
Locked U	sers:	0							
Total User	s	0							

En la solapa *Control->Start/Stop* podemos arrancar y parar el servidor en distintos modos. La opción *Graceful Shutdown timeout* indica el tiempo que esperaremos antes de forzar el apagado del servidor (siempre con el apagado *Graceful* que espera a que todas las sesiones HTTP hayan terminado). Indicando el valor cero estamos diciendo que esperará indefinidamente. Si activamos la *Ignore Sessions During Shutdown*, no se tendrán en cuenta las sesiones. También podemos asignar un tiempo de espera antes del arranque. Más abajo tenemos las opciones de arranque y parada. La primera que nos aparece (*Start*) arranca el servidor (para hacerlo necesitaremos el Node Manager). La segunda reinicia el servidor. *Suspend* puede suspender la ejecución del servidor ahora mismo o dejarlo terminar la tarea actual. *Shutdown* para el servidor, permitiendo igual que antes forzarlo o esperar a que termine.

onfiguration	Protocols	Logging	Debug	Monitoring	Control	Deployments	Services	Security	Not
Start/Stop	Remote S	tart Outpu	it Migr	ation					
Save									
Jse this page t Ir view the curr Iomain-wide a	.o change t rent status dministratio	he state o of this ser on port.)	f the cur ver. (So	rrent server me operatio	. You can a ons require	so specify parti the Node Mana	cular shuto ager and th	lown settin e	gs
∏lgnore	Sessions	During S	hutdow	'n					
Graceful Timeout:	Shutdowr	י ס							
🗐 Startup T	imeout:	0							
Save	this table								
Server Statu	sume S	iuspend 🔻	Shut	down 👻	Restart SSL	Showing 1 -	l of l Pre	evious Ne>	đ
Server «	*	Machi	ine	State	S	tatus of Last	Action		
☐ AdminSe	rver			RUNNING	G N	one			
			1000						

2.3. Definición de máquinas

Una máquina es un reflejo de una máquina física que soporta nuestro sistema. Weblogic se puede instalar en multitud de plataformas (Unix, Windows, Solaris, etc.). Una de las principales ventajas de Weblogic es que se pueden utilizar distintos tipos de máquinas en el mismo sistema. Por ejemplo, podemos tener una máquina Linux con un par de servidores y otra máquina Windows NT con otro par, funcionando todo dentro del mismo dominio y dando el mismo servicio. Cuando pinchamos en la opción *Machines* nos permite visualizar las máquinas que tiene nuestro sistema y también nos permite configurar nuevas máquinas.

A machine is the logical representation of the co (servers). WebLogic Server uses configured mac to which certain tasks, such as HTTP session rep machine definition in conjunction with Node Mar	omputer that hosts one or more WebLogic Server instances chine names to determine the optimum server in a cluster plication, are delegated. The Administration Server uses the nager to start remote servers.
ïhis page displays key information about each r Server domain.	machine that has been configured in the current WebLogic
Customize this table	
Customize this table Machines	
Customize this table Machines New Clone Delete	Showing 0 - 0 of 0 Previous Next
Customize this table Machines New Clone Delete Name	Showing 0 - 0 of 0 Previous Next
Customize this table Machines Clone Delete Name There ar	Showing 0 - 0 of 0 Previous Next Type re no items to display

Cuando pinchamos en *New* nos aparece la figura siguiente donde podemos dar nombre a la máquina. Este nombre es propio de Weblogic, es decir, no es nombre DNS ni IP. Podemos configurar una máquina de tipo Unix (Linux, etc.) o un tipo distinto de máquina.

Create a New Machin		
OK Cancel		
Machine Properties		
The following propertie	will be used to identify your new Machine.	
What would you like Name:	o name your new Machine? Machine-0	
Specify the type of	nachine operating system.	
Machine OS:	Unix –	
OK Cancel		

La diferencia es que en las máquinas de tipo Unix nos permite configurar opciones adicionales (en las otras sólo nos deja el nombre). Las siguientes opciones, que sólo aparecen en máquinas Unix, son el nombre de usuario y de grupo con los que se ejecutarán los

Copyright © 2006-2007 Depto. CCIA All rights reserved.

procesos instanciados en esa máquina.

Save	
This page allows you t	to define the general configuration for this Unix machine.
🕼 Name:	Machine-0
🐐 🦵 Enable Post-Bi	nd UID
Post-Bind UID:	nobody
툏 🦵 Enable Post-Bi	nd GID
Post-Bind GID:	nobody

Para asignar servidores a una máquina, nos iremos a cada servidor y en la pestaña de *Configuration->General* asociamos ese servidor a esa máquina.

Configuration P	Protocols	Logging	Debug	Monitoring	Control	Deployments
General Clu	uster Se	ervices	Keystores	SSL D	eployment	Migration
Save						
Lica this page	o to confi		ral faatuur	oc of this co	n ior au ch	aa dafayih nat
Use this pade	e to confi	gure gene	raiteatur	es of this se	erver such	as default het
, ,	\$					
View JNDI Tre	e ≠=	-				
View JNDI Tre	e ,=	 se	rvidor2			
View JNDI Tre Name: Machine:	e ,=	se	rvidor2 1achine-0			
View JNDI Tre Name: Machine: Cluster:	€ ,=	 se [№	rvidor2 1achine-0 Standalo	ne) –		

De vuelta a las máquinas, los servidores asociados a esta máquina aparecerán en la pestaña *Configuration->servers*. Desde este punto también podemos añadir servidores a la máquina.

Settings for maquina1						
Configuration Monitoring	g Notes					
General Node Manag	er Servers					
This page displays the set of available servers. Customize this table Servers	rvers that have be	en assigned to this M	lachine. You ca	n select a serve	r to con	
Add Remove	Add Remove Showing 0 - 0 of					
🗖 Name 🐟	Cluster	Machine	State	Health	Lister	
		There are no item	ns to display			
Add Remove				Showing	0 - 0 of	

Para añadir servidores, pinchamos en Add y seleccionamos los servidores que queramos añadir.

Add a Server to Machine	
Back Next Finish	Cancel
Identify Server Identify the server to be add	led
How would you like to pr	roceed?
Select an existing s	erver, and associate it with this machine
Select a server:	AdminServer
C _{Create} a new serve	er and associate it with this machine
Back Next Finish	Cancel

3. Administración desde línea de comandos

La administración desde línea de comandos es una herramienta útil para el control del sistema. Nos permite realizar varias tareas de monitorización y comprobación del sistema. También es útil en modo producción, pues normalmente se deshabilita la consola de administración por motivos de seguridad.

El comando a ejecutar es el siguiente:

```
java -cp $BEA_HOME/weblogic92/server/lib/weblogic.jar weblogic.Admin -url
URL
```

-username usuario -password contraseña COMANDO argumentos

El comando no es más que una llamada a la clase *weblogic.Admin* que se encuentra en el fichero *weblogic.jar* proporcionado por BEA. El parámetro URL especifica la dirección URL del servidor de administración o bien del servidor contra el que vayamos a realizar el comando. Debemos especificar también el puerto en la dirección. El usuario y su contraseña asociada deben ser válidos para el comando que vamos a ejecutar. Los posibles comandos a utilizar pueden ser algunos de los siguientes:

CONNECT	Realiza el especificado número de conexiones y devuelve el tiempo (en milisegundos) total y medio de conexión.
ejemplo	java weblogic.Admin -url t3://localhost:7001 -username system -password weblogic CONNECT 25
	Connection: 0 - 18 ms Connection: 1 - 7 ms Connection: 2 - 7 ms Connection: 3 - 7 ms Connection: 4 - 9 ms Connection: 5 - 7 ms Connection: 6 - 10 ms
DISCOVERMANAG EDSERVER	Provoca que el servidor de administración reestablezca el control administrativo sobre los servidores administrados.
ejemplo	java weblogic.Admin -url t3://localhost:7001 -username system -password weblogic DISCOVERMANAGEDSERVER -serverName servidor2 -listenPort 6001 -listenAddress localhost
FORCESHUTDOWN	Termina de forma inmediata un proceso servidor pasado como argumento.
ejemplo	java weblogic.Admin -url t3://localhost:7001 -username system -password weblogic FORCESHUTDOWN servidor2 Server "servidor2" was force shutdown successfully
GETSTATE	Devuelve el estado del servidor pasado como argumento.
ejemplo	java weblogic.Admin -url t3://localhost:7001 -username system -password weblogic GETSTATE servidor2 Current state of "servidor2" : RUNNING
HELP	Muestra ayuda de un comando o de todos (ALL).
ejemplo	java weblogic.Admin -url t3://localhost:7001 -username system -password weblogic HELP GETSTATE

LICENSES	Lista las licencias instaladas en la máquina.
ejemplo	java weblogic.Admin -url t3://localhost:7001 -username system -password weblogic LICENSES
PING	Envía un mensaje para verificar que un servidor está disponible y aceptando peticiones. Opcionalmente podemos pasarle dos argumentos: el número de veces intentos y el tamaño de cada paquete.
ejemplo	java weblogic.Admin -url t3://localhost:7001 -username system -password weblogic PING 10 1000 Sending 10 pings of 1.000 bytes.
	RTT = ~185 milliseconds, or ~18 milliseconds/packet
SERVERLOG	Muestra el fichero log de un servidor. Se puede especificar un intervalo de tiempo a mostrar. En el ejemplo se muestra desde las 14:00 horas del 24 de diciembre a las 10:00 horas del 31 de diciembre.
ejemplo	java weblogic.Admin -url t3://localhost:7001 -username system -password weblogic SERVERLOG "2002/12/24 14:00" "2002/12/31 10:00"
SHUTDOWN	Para la ejecución de un servidor (<i>Graceful</i>). Podemos especificar los mismo parámetros que desde la consola: que ignore las sesiones abiertas y el <i>timeout</i> en segundos que esperará antes de pararlo.
ejemplo	java weblogic.Admin -url t3://localhost:7001 -username system -password weblogic SHUTDOWN -ignoreExistingSessions -timeout 10 servidor1
START	Arranca un servidor si tenemos disponible el Node Manager.
ejemplo	java weblogic.Admin -url t3://localhost:7001 -username system -password weblogic START servidor1
THREAD_DUMP	Muestra información de los hilos de ejecución que se están ejecutando en el servidor, por la salida estándar de este.
ejemplo	java weblogic.Admin -url t3://localhost:7001 -username system -password weblogic THREAD_DUMP servidor1

VERSION	Muestra la versión del software Weblogic.
ejemplo	java weblogic.Admin -url t3://localhost:7001 -username system -password weblogic VERSION

Existen otros comandos y utilidades adicionales que se detallan a continuación. En todos ellos hace falta incluir el fichero *weblogic.jar* en el classpath.

dbping	Realiza una conexión a la base de datos especificada utilizando una clase de las proporcionadas por WebLogic.
	<pre>java utils.dbping ORACLE_THIN system oracle localhost:1521:j2eebd **** Success!!! **** You can connect to the database in your app using: java.util.Properties props = new java.util.Properties(); props.put("user", "system"); props.put("password", "oracle");</pre>
system	Obtiene información del sistema: la versión y el desarrollador de Java, el classpath, el nombre, arquitectura y versión del sistema operativo
ejemplo	java utils.system ****** java.version ************************************
туір	Obtiene la dirección IP y el nombre DNS de la máquina
ejemplo	java utils.myip Host portatil-miguel is assigned IP address: 169.254.100.36
Deployer	Controla el despliegue de aplicaciones
ejemplo: desplegar una aplicación	java weblogic.Deployer -adminurl t3://localhost:7001 -username system

Copyright © 2006-2007 Depto. CCIA All rights reserved.

	-password weblogic -name faqs -targets adminServer, servidor2 -deploy c:/myapps/faqs.war
ejemplo: eliminar una aplicación	java weblogic.Deployer -adminurl t3://localhost:7001 -username system -password weblogic -name faqs -targets adminServer -undeploy
ejemplo: reactivar una aplicación	java weblogic.Deployer -adminurl t3://localhost:7001 -username system -password weblogic -name faqs -targets adminServer -redeploy