



# Servidores de aplicaciones

Sesión 6:  
JNDI y acceso a bases de datos  
con el servidor de aplicaciones



# Índice

- Programación con JNDI
- WebLogic y JNDI
- Configuración de fuentes de datos
- Uso de fuentes de datos en una aplicación



# JNDI

- JNDI: *Java Naming and Directory Interface*
- Permite enlazar programas Java con: sistemas de ficheros, recursos de red, objetos, etc.
- Principales funciones:
  - Servicio de nombres: asociación de nombres lógicos con recursos
  - Servicio de directorio: sistema de ficheros, de datos



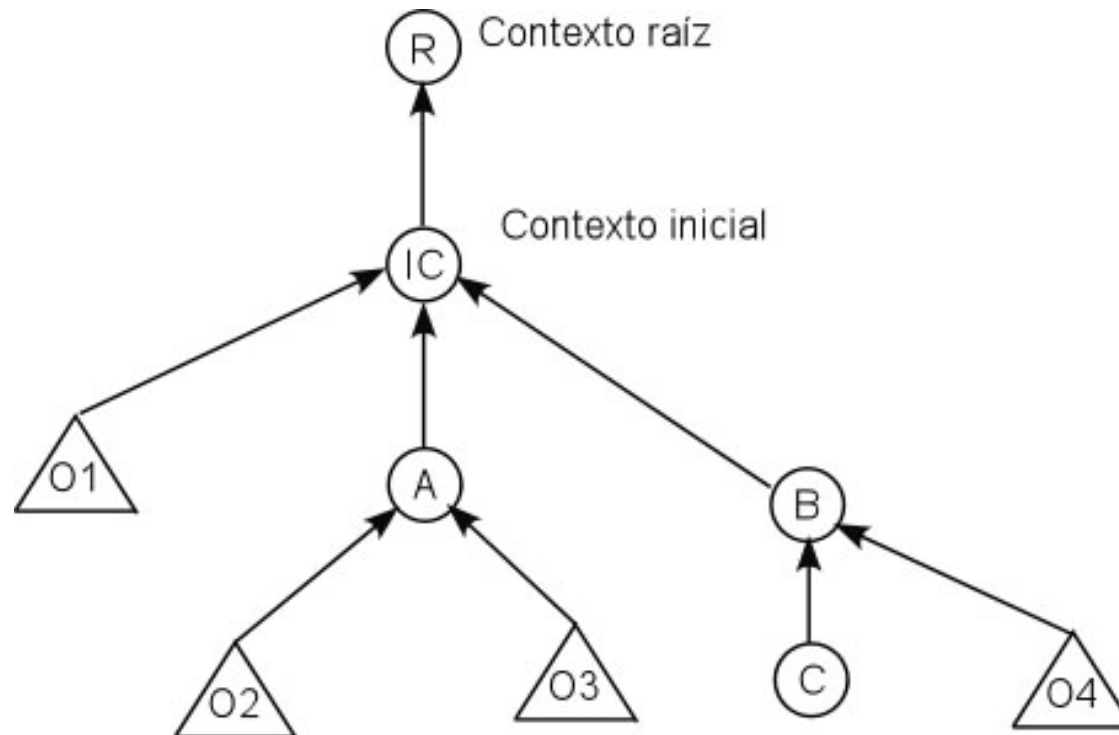
## Búsqueda de objetos mediante su nombre lógico

- Se mapea un objeto con un nombre lógico
- JNDI se encarga de mantener el mapeo
- Nuestro programa, si quiere obtener el objeto, llama al nombre lógico
- Ventaja: si modificamos el objeto (su nombre, por ejemplo) no tenemos que cambiar nuestro programa, sólo el enlace en JNDI



# Definiciones

- Contexto: se puede entender como un directorio. Dentro de un contexto podemos tener objetos o más contextos





# Programación con JNDI

- Context miContexto = null;  
Hashtable ht = new Hashtable ();  
ht.put(Context.INITIAL\_CONTEXT\_FACTORY,  
"weblogic.jndi.WLInitialContextFactory");  
ht.put(Context.PROVIDER\_URL, "t3://localhost:7001");  
miContexto = new InitialContext (ht);
- Hay que capturar la excepción *NamingException*
- Cuando terminemos de utilizarlo llamamos al método *close*  
miContexto.close();



# Programación con JNDI

- Enlace de un objeto
  - ```
Persona persona = new Persona();  
miContexto.bind ("objeto persona", persona);  
// miContexto.rebind ("objeto persona", persona);
```
- Creación de subcontexto
  - ```
Context subcontexto =  
miContexto.createSubContext  
("empleados");  
Persona persona = new Persona();  
subcontexto.bind ("contable", persona);
```
- Recuperación de objetos
  - ```
Persona pers = (Persona) miContexto.lookup  
("empleados/contable");
```



# WebLogic y JNDI

- WebLogic gestiona un árbol JNDI
- Podemos ver el árbol JNDI de ciertos objetos (p.e. servidor)

Configuration | Protocols | Logging | Debug | Monitoring | Control | Deployments | Services | Security

General | Cluster | Services | Keystores | SSL | Deployment | Migration | Tuning | Overview

Click the **Lock & Edit** button in the Change Center to modify the settings on this page.

Use this page to configure general features of this server such as default network communication protocols.


[View JNDI Tree](#)

|                        |                      |
|------------------------|----------------------|
| <b>Name:</b>           | AdminServer          |
| <b>Machine:</b>        | (None)               |
| <b>Cluster:</b>        | (Standalone)         |
| <b>Listen Address:</b> | <input type="text"/> |





# Árbol mostrado

 **WEBLOGIC SERVER**  
ADMINISTRATION CONSOLE

**Domain Structure**

```
AdminServer
├── bea_wls_remote_deployer_bea_wls_remote_deployer_jarDeployerBean_EO
├── com
│   └── bea
│       └── ejecutivos
│           ├── Ejecutivo en jefe
│           ├── Presidente
│           └── Propietario
├── ejb
├── javax
│   └── mejbmejb_jarMejb_EO
└── weblogic
```

**Contents**  
**Settings for com.bea.ejecutivos;Ejecutivo en jefe;AdminServe**  
Binding **Security**  
**General**

This tab allows you to configure a Bound Object

|                      |                             |
|----------------------|-----------------------------|
| <b>Binding Name:</b> | Ejecutivo en jefe           |
| <b>Class:</b>        | arranque.objetos.Jefe       |
| <b>Hash Code:</b>    | 407510                      |
| <b>ToString:</b>     | arranque.objetos.Jefe@637d6 |



## Clases de arranque y parada

- WebLogic permite la ejecución de clases cuando un servidor arranca o para
- Puede servir para “colgar” objetos en el árbol JNDI
- Las clases deben estar disponibles en el CLASSPATH (editar los ficheros *startManagedWebLogic.cmd* y *startWebLogic.cmd*)



# Despliegue de clases de arranque y parada

- Pinchamos en *Environment->Startup&stop classes*

### Summary of Startup and Shutdown Classes

Startup and shutdown classes are Java programs that you create to provide custom, system-wide services for your applications. You add the classes to the WebLogic Server class path and then configure them to load and run when a server starts or shuts down.

This page summarizes the startup and shutdown classes that have been configured in the current domain. You must deploy each class on one or more specific servers.

[Customize this table](#)

#### Startup and Shutdown Classes

New Clone Delete Showing 0 - 0 of 0 Previous | Next

| <input type="checkbox"/>      | Name | Type | Class Name | Deployment Order | Arguments |
|-------------------------------|------|------|------------|------------------|-----------|
| There are no items to display |      |      |            |                  |           |

New Clone Delete Showing 0 - 0 of 0 Previous | Next



# Despliegue de clases de arranque y parada

- Definimos el tipo de clase a desplegar

Configure a New Startup or Shutdown Class

Back Next Finish Cancel

**Class Type**  
Which type of class would you like to define?

Startup Class

Shutdown Class

Back Next Finish Cancel



# Despliegue de clases de arranque y parada

- Definimos el nombre (genérico) asignado a este despliegue y el nombre de la clase

**Configure a New Startup or Shutdown Class**

Back | Next | Finish | Cancel

---

**Startup Class Properties**  
The following properties will be used to identify the class you are configuring.

What name would you like to use to identify the class?

**Name:**

**Class Name:**

Back | Next | Finish | Cancel



# Despliegue de clases de arranque y parada

- Indicamos los servidores donde se desplegará

Configure a New Startup or Shutdown Class

Back Next Finish Cancel

---

**Select Targets**

You can target this new class to any of these servers or clusters. Please select your targets.

Targets

| Servers                                         |
|-------------------------------------------------|
| <input checked="" type="checkbox"/> AdminServer |
| <input type="checkbox"/> servidor2              |
| <input type="checkbox"/> servidor3              |

Back Next Finish Cancel



# Despliegue de clases de arranque y parada

- Activamos los cambios
- Podemos configurar varias opciones más

Settings for MiClaseArranque

Configuration **Targets** Notes

Click the **Lock & Edit** button in the Change Center to modify the settings on this page.

Use this page to modify the configuration of a startup class.

**Name:** MiClaseArranque

**ClassName:**

**Deployment Order:**

**Arguments:**

Failure is Fatal

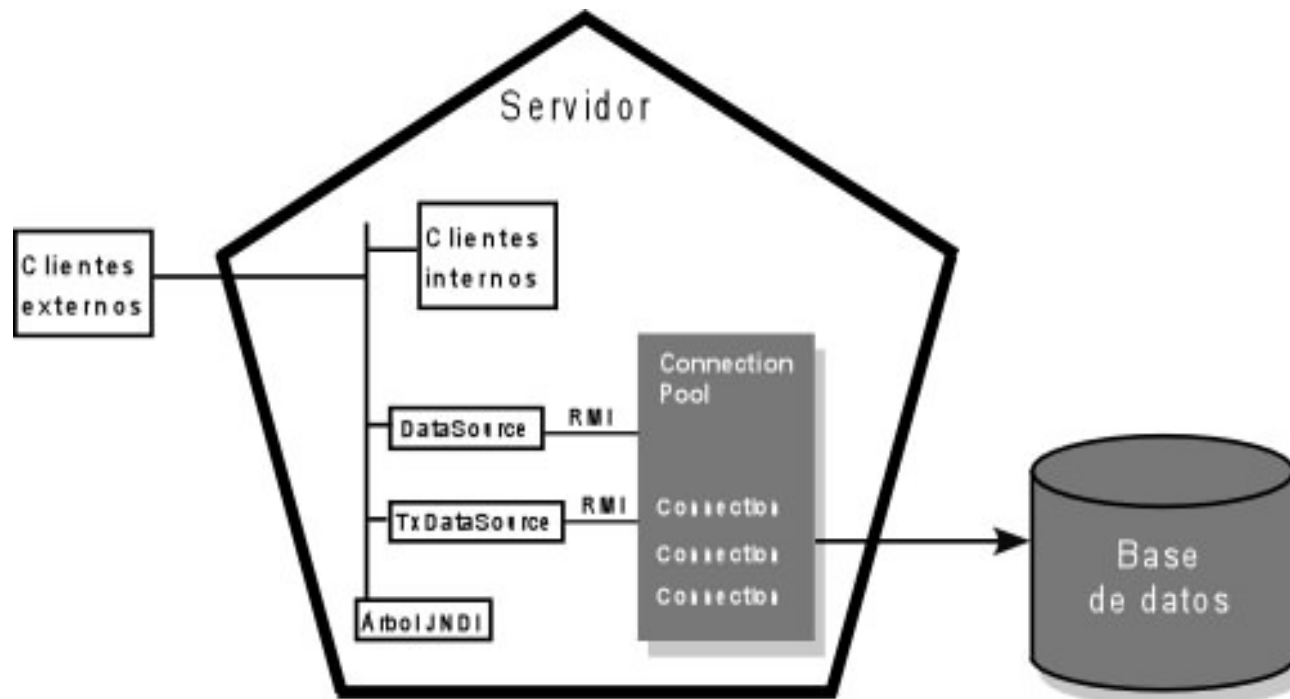
Run Before Application Deployments

Run Before Application Activations

Click the **Lock & Edit** button in the Change Center to modify the settings on this page.



# JDBC y WebLogic







## Cambio de versión

- En versiones anteriores a la 9.0, para trabajar con bases de datos se debía definir primero el pool de conexiones y luego crear una fuente de datos asociada a ese pool
- A partir de la 9.0 sólo es necesario definir la fuente de datos. Ésta lleva asociada internamente un pool.
- Pinchamos en *Services->DataSources*.



# Definición de una fuente de datos (I)

- No incluye *drivers* de todos los SGBD que aparecen
- Debemos incluir el *driver* en el CLASSPATH

Create a New JDBC Data Source

Back Next Finish Cancel

**JDBC Data Source Properties**  
The following properties will be used to identify your new JDBC data source.

What would you like to name your new JDBC data source?

Name:

What JNDI name would you like to assign to your new JDBC Data Source?

JNDI Name:

What database type would you like to select?

Database Type:

What database driver would you like to use to create database connections?

Database Driver:

Back Next Finish Cancel



# Definición de una fuente de datos (II)

### Create a New JDBC Data Source

Back Next Finish Cancel

#### Transaction Options

You have selected non-XA JDBC driver to create database connection in your new data source.

Does this data source support global transactions? If yes, please choose the transaction protocol for this data source.

**Supports Global Transactions**

Select this option if you want to enable non-XA JDBC connections from the data source to participate in global transactions using the Logging Last Resource (LLR) transaction optimization. Recommended in place of Emulate Two-Phase Commit.

**Logging Last Resource**

Select this option if you want to enable non-XA JDBC connections from the data source to emulate participation in global transactions using JTA. Select this option only if your application can tolerate heuristic conditions.

**Emulate Two-Phase Commit**

Select this option if you want to enable non-XA JDBC connections from the data source to participate in global transactions using the one-phase commit transaction processing. With this option, no other resources can participate in the global transaction.

**One-Phase Commit**

Back Next Finish Cancel



# Definición de una fuente de datos (III)

Create a New JDBC Data Source

Back Next Finish Cancel

**Connection Properties**  
Define Connection Properties.

What is the name of database you would like to connect to?

**Database Name:**

What is the name or IP address of the database server?

**Host Name:**

What is the port on the database server used to connect to the database?

**Port:**

What database account user name do you want to use to create database connections?

**Database User Name:**

What is the database account password to use to create database connections?

**Password:**

**Confirm Password:**

Back Next Finish Cancel



# Definición de una fuente de datos (IV)

Test Configuration | Back | Next | Finish | Cancel

**Test Database Connection**  
Test the database availability and the connection properties you provided.

What is the full package name of JDBC driver class used to create database connections in the connection pool? (Note that this driver class must be in the classpath of any server to which it is deployed.)

**Driver Class Name:**

What is the URL of the database to connect to? The format of the URL varies by JDBC driver.

**URL:**

What database account user name do you want to use to create database connections?

**Database User Name:**

What is the database account password to use to create database connections?

**Password:**

**Confirm Password:**

What are the properties to pass to the JDBC driver when creating database connections?

**Properties:**



# Definición de una fuente de datos (final)

**Create a New JDBC Data Source**

Back Next Finish Cancel

**Select Targets**

You can select one or more targets to deploy your new JDBC data source. If you don't select a target, the data source will be created but not deployed. You will need to deploy the data source at a later time.

| Servers                                         |
|-------------------------------------------------|
| <input checked="" type="checkbox"/> AdminServer |
| <input type="checkbox"/> servidor2              |
| <input type="checkbox"/> servidor3              |

Back Next Finish Cancel



## Configuración de opciones del pool

- Pinchamos sobre el nombre de la fuente de datos
- Nos vamos a la solapa *Configuration->Connection pool*
- Podemos definir la capacidad inicial, máxima capacidad y el incremento



# Configuración de opciones del pool

Settings for Mi Data Source

Configuration | Targets | Monitoring | Control | Security | Notes

General | **Connection Pool**

Click the **Lock & Edit** button in the Change Center to modify the settings on this page.

The connection pool within a JDBC data source contains a group of JDBC connections that applications reserve, use, and then return to the pool. The connection pool and the connections within it are created when the connection pool is registered, usually when starting up WebLogic Server or when deploying the data source to a new target.

Use this page to define the configuration for this data source's connection pool.

**URL:** jdbc:mysql://localhost:3306/vuelos

**Driver Class Name:** com.mysql.jdbc.Driver

**Properties:** user=admin

**Password:** [REDACTED]

**Confirm Password:** [REDACTED]

**Initial Capacity:** 1

**Maximum Capacity:** 15

**Capacity Increment:** 1

**Statement Cache Type:** LRU

**Statement Cache Size:** 10

Advanced

Click the **Lock & Edit** button in the Change Center to modify the settings on this page.





## Multi fuentes de datos

- Conjunto de fuentes de datos que permiten dar soporte de balanceo de carga y recuperación ante fallos
- Simplemente es una agrupación de distintas fuentes de datos (previamente creadas)



# Creación de multidatasources

## Create a New JDBC Multi Data Source

Back

Next

Finish

Cancel

### Configure the Multi Data Source

The following properties will be used to identify your new JDBC multi data source.

What would you like to name your new JDBC multi data source?



Name:

Mi Multi DS

What JNDI name would you like to assign to your new JDBC multi data source?



JNDI Name:

miMDS

What algorithm type for this JDBC Multi Data Source would you like to select?



Algorithm Type:

Failover

Back

Next

Finish

Cancel



# Creación de multidatasources

**Create a New JDBC Multi Data Source**

---

**Select Data Source Type**

Please select type (XA or Non-XA) of data source you would like to add to your new JDBC Multi Data Source.

XA Driver

Non-XA Driver

---



# Creación de multidatasources

**Create a New JDBC Multi Data Source**

Back Next Finish Cancel

---

**Add Data Sources**  
What JDBC Data Sources would you like to add to your new JDBC Multi Data Source?

**Data Sources:**

| Available | Chosen         |
|-----------|----------------|
|           | Mi Data Source |

Back Next Finish Cancel



# Uso de fuentes de datos en nuestra aplicación

- Haremos uso de JNDI para acceder a una fuente de datos
- Si queremos ejecutar externamente al servidor (aplicación separada del servidor) debemos incluir en el CLASSPATH la ruta hasta `$HOME_BEA/weblogic92/server/lib/weblogic.jar`



## Código a incluir en nuestro programa

- Importación de clases:

```
import javax.sql.DataSource;  
import javax.naming.*;  
import java.util.Hashtable;
```

- Definición del contexto inicial:

```
Context miContexto = null;  
Hashtable ht = new Hashtable ();  
ht.put(Context.INITIAL_CONTEXT_FACTORY,  
        "weblogic.jndi.WLInitialContextFactory");  
ht.put(Context.PROVIDER_URL, "t3://localhost:7001");  
miContexto = new InitialContext (ht);
```

↖ Servidor que  
contenga la fuente  
de datos



## Código a incluir en nuestro programa

- Obtención de la fuente de datos:  
*DataSource ds = (DataSource)  
miContexto.lookup ("MySQLDataSource");*
- Obtención de la conexión:  
*Connection con = ds.getConnection ();*