

# Instituto Politécnico Nacional

## Escuela Superior de Cómputo

Web App Development.

Tarea 5 : Documentar Línea a Línea la Configuración de un DBCP.

*Profesor: M. en C. José Asunción Enríquez Zárate*

*Alumno: Aarón Antonio García González*

*aarongarcia.ipn.escom@gmail.com*

*3CM9*

19 de diciembre de 2020

# Índice

<b>1. Introducción</b>	<b>2</b>
1.1. <a href="#">Tendencias de futuro alrededor de las Bases de Datos</a> . . . . .	2
<b>2. Big data</b>	<b>3</b>
2.1. <a href="#">Connection Pool</a> . . . . .	3
<b>3. Desarrollo</b>	<b>4</b>
<b>4. Conclusión</b>	<b>5</b>
<b>5. Referencias Bibliográficas</b>	<b>6</b>

## Lista de figuras

Lista de tablas

1. Data Source . . . . . 4

# 1. Introducción

Las Bases de Datos tienen una gran relevancia a nivel personal, pero más si cabe, a nivel empresarial, y se consideran una de las mayores aportaciones que ha dado la informática a las empresas. En la actualidad, cualquier organización que se precie, por pequeña que sea, debe contar con una Base de Datos, pero para que sea todo lo efectiva que debe, no basta con tenerla: hay que saber cómo gestionarla.

Las principales utilidades que ofrece una base de datos a la empresa son las siguientes:

- Agrupar y almacenar todos los datos de la empresa en un único lugar.
- Facilitar que se compartan los datos entre los diferentes miembros de la empresa.
- Evitar la redundancia y mejorar la organización de la agenda.
- Realizar una interlocución adecuada con los clientes.

Si una Base de Datos se gestiona adecuadamente, la organización obtendrá diferentes ventajas. Aumentará su eficacia, habrá trabajos que se realicen con mayor rapidez y agilidad debido a la simplificación de los mismos, podremos mejorar la seguridad de los datos que almacenamos, y con todos estos factores, maximizaremos los tiempos y por tanto, se producirá una mejora en la productividad.[DataCentric, 2019]

## 1.1. Tendencias de futuro alrededor de las Bases de Datos

En cuanto al futuro de las Bases de Datos hay dos tendencias que suenan con mucha fuerza entre los expertos del sector.

En primer lugar cabe hacer referencia a su tipología, en la actualidad. Las Bases de Datos relacionales o SQL son las más utilizadas, pero presentan varias desventajas como que son muy rígidas o que impiden un crecimiento constante del archivo. Por su parte las NoSQL se postulan como alternativa en el futuro más inmediato.

Sin embargo, hay dos grandes inconvenientes que hacen que las Bases de Datos NoSQL o no relacionales no avancen al ritmo que deberían, y ambos están relacionados con la formación de las personas que las utilizan.

El primer inconveniente es que muchos expertos se muestran reacios a utilizarlas, en muchas ocasiones, porque no conocen todas las posibilidades que ofrecen; el segundo (relacionado con el primero) es que este tipo de Bases de Datos tienen una gran diversidad, y hay que tener conocimientos sobre todas ellas para saber cuál es la adecuada para cada caso ( es imprescindible tener muy definido el tipo de proyecto al que se asociará, la cantidad esperada de usuarios y la tecnología que se va a utilizar para el desarrollo).

En este aspecto, hay defensores y detractores de cada tipo; los defensores de las Bases de Datos SQL apuestan por la estructuración que ofrece esta tipología, ya que consideran que las NoSQL carecen de estructura. Esta creencia es falsa, y se debe principalmente a la desinformación: las NoSQL sí que poseen una estructura definida denominada schema-free, la diferencia es que es más flexible que la que poseen las SQL o relacionales.

Aparte de la evolución del tipo de Bases de Datos también hay que tener en cuenta los cambios en cuanto a manipulación de las mismas, dada la creciente importancia que tienen en la vida cotidiana los dispositivos móviles como smartphones y tablets. El cambio de dispositivos en el que se manejan las Bases de Datos, obligará a que en el futuro se simplifiquen las interfaces y se facilite el uso de estos sistemas lo que mejorará su rendimiento y ampliará su campo de uso, por ejemplo, en el sector médico o militar que requieren de un acceso más inmediato.

Otro factor importante que esta marcando el futuro de las bases de datos esta siendo la privacidad de los mismos, con la que la sociedad esta cada vez más sensibilizada y para la que se están utilizando nuevas formas de entender las bases de datos, como el blockchain, que tendrá un papel fundamental en el Database Marketing del futuro.

## 2. Big data

El concepto Big Data engloba muchas ideas y se asocia a otros términos como Data Science, Data Analytics o Data Mining, pero el objetivo fundamental de todos ellos es extraer valor de los datos, información útil para las decisiones de negocio.

Es también conocida la definición de Big Data como las cuatro V, que representan el gran Volumen de datos que debe ser capaz de tratar, la Velocidad con la que puede procesar esos datos, y la Variedad de formas que pueden tomar los mismos. Debemos hacer énfasis en el objetivo del Big Data añadiendo una cuarta V, la del Valor que se obtiene por la información extraída de los datos.

El big data está formado por conjuntos de datos de mayor tamaño y más complejos, especialmente procedentes de nuevas fuentes de datos. Estos conjuntos de datos son tan voluminosos que el software de procesamiento de datos convencional sencillamente no puede administrarlos. Sin embargo, estos volúmenes masivos de datos pueden utilizarse para abordar problemas empresariales que antes no hubiera sido posible solucionar.[IIC UAM Madrid, 2020]

### 2.1. Connection Pool

En computación, se denomina connection pool (agrupamiento de conexiones) al manejo de una colección de conexiones abiertas a una base de datos de manera que puedan ser reutilizadas al realizar múltiples consultas o actualizaciones.

Cada vez que un programa cliente necesita comunicarse con una base de datos, establece una conexión, generalmente utilizando un protocolo especializado. Esta conexión, generalmente se mantiene abierta el tiempo que dura la ejecución del programa y sólo es cerrada al finalizar el trabajo de la aplicación con la base de datos

En java, un pool de conexiones es una clase java que tiene abiertas varias conexiones a base de datos. Cuando alguien necesita una conexión a base de datos, en vez de abrirla directamente con `DriverManager.getConnection()`, se la pide al pool usando su método `pool.getConnection()`. El pool coge una de las conexiones que ya tiene abierta, la marca como que alguien la está usando para no dársela a nadie más y la devuelve. La siguiente llamada a este método `pool.getConnection()`, buscará una conexión libre para marcarla como ocupada y la devolverá ... y así sucesivamente.

Cuando el que ha pedido la conexión termina de usarla, normalmente después de una transacción con la base de datos o varias seguidas, llama al método `connection.close()`. Esta conexión que nos ha sido entregada por el pool, realmente no se cierra con esta llamada. El método `close()` únicamente avisa al pool que ya hemos terminado con la conexión, de forma que sin cerrarla, la marca como libre para poder entregársela a otro que lo pida.

### 3. Desarrollo

Elemento	Atributo	Valor	Descripción
Resource			
	name	jdbc/3cm9	El nombre del recurso que se va a crear, relativo al contexto java:comp/env
	global	jdbc/3cm9	
	type	javax.sql.DataSource	El nombre de la clase Java calificada que espera la aplicación web cuando busca este recurso
	auth	Container	Especifica si el código de la Aplicación Web inicia sesión como el administración de recursos correspondiente o si el contenedor inicia sesión como administrador de recursos en nombre de la aplicación. El valor de este atributo debe ser Application o Container. Es obligatorio si la aplicación utilizará un elemento del tipo resource-ref en el descriptor de la implementación(development descriptor), de lo contrario es opcional.
	factory	org.apache.tomcat.jdbc.pool.DataSourceFactory	Indica la clase Java que utiliza la interfaz Factory, que crea un objeto sin usar una entrada o un parámetro.
	username	root	Nombre de usuario para la conexión
	password	tucontrasenia	Contraseña para la conexión
	driver Class-name	com.mysql.cj.jdbc.Driver	El controlador que se utilizará para poder llevar a cabo la conexión.
	description	Alguna descripción	Descripción para hacer el recurso más entendible para el ser humano.
	url	jdbc:mysql://localhost:3306/3CM9	URL del recurso
	initialSize	10	El número inicial de conexiones que se crean cuando inicia el pool.
	maxActive	10	Indica el número máximo de conexiones activas al mismo tiempo.
	maxWaitMillis		Es el tiempo máximo en milisegundos que va a esperar para que la conexión sea exitosa.
	remove Abandoned-Timeout	300	Tiempo en milisegundos que se esperan para que una conexión (activa) sin actividad pueda ser removida.
	defaultAuto Commit	true	Es una bandera que indica el estado del Auto-Commit, si no se asigna ninguno por default se asigna JDBC.

Tabla 1: Data Source

## 4. Conclusión

El Data Source es un recurso que nos facilita realizar conexiones a los servicios de bases de datos, ya que no se pasa de usar distintas librerías en las clases de Java tales como Connection.

Algunos de los tags más importantes y que son cruciales son name (nombre que se le asigna al recurso y con el cual se identifica), driverClassname (este apunta al controlador que se utiliza para poder llevar a cabo la conexión), url (Ya que es la dirección del recurso).

De los demás la mayoría tienen un valor por default por lo que no se tienen que definir de manera obligatoria, aunque algunos atributos como masActive y removeAbandonedTimeout pueden ayudar a que la conexión tenga un funcionamiento más óptimo.

*Aarón Antonio García González*



## 5. Referencias Bibliográficas

### Referencias

[IIC UAM Madrid, 2020] Oracle. *Big Data, Aprovecha los datos para extraer información de valor para tu negocio* IIC UAM.

[DataCentric, 2019] Oracle. *¿Qué importancia tienen las bases de datos a nivel empresarial?* DataCentric.

[Platzi, 2015] Oracle. *Qué es PostgreSQL y cuáles son sus ventajas* Platzi.