

¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
Ese apoyo que siempre quiso tener...

1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,
HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
Gestor de contenidos (Alfresco)
Aplicaciones híbridas

Tareas programadas (Quartz)
Gestor documental (Alfresco)
Inversión de control (Spring)

Control de autenticación y
acceso (Spring Security)
UDDI
Web Services
Rest Services
Social SSO
SSO (Cas)

JPA-Hibernate, MyBatis
Motor de búsqueda empresarial (Solr)
ETL (Talend)

Dirección de Proyectos Informáticos.
Metodologías ágiles
Patrones de diseño
TDD

BPM (jBPM o Bonita)
Generación de informes (JasperReport)
ESB (Open ESB)



[Home](#) | [Quienes Somos](#) | [Empleo](#) | [Tutoriales](#) | [Contacte](#)

Autor: Javier Antoniucci

jantoniucci@deloitte.es

Descargar este documento en formato PDF [iBatis.pdf](#)

[Firma en nuestro libro de Visitas](#)

[Master Experto® Oracle 10](#)

Certificados Oracle. Prácticas en empresas.
Fórmate para trabajar.
www.grupoatrium.com

[Curso Superior Java J2EE](#)

¿Quieres? Puedes. Fórmate en Instituto
Novatech 913952875
www.institutonovatech.com

[Master Élite Oracle 10g](#)

Master certificado Oracle 10g con prácticas en
empresas. Becas.
www.exes.es

Manual Básico de Apache iBatis

Introducción

¿Qué es iBatis?

Apache iBatis está constituido por dos frameworks independientes que generalmente se usan juntos: DAO y sqlMaps. El primero simplifica la implementación el patrón de diseño Direct Access Objects (DAO) y el segundo simplifica la persistencia de objetos en bases de datos relacionales.

iBatis NO es un ORM (Object Relational Mapper), por lo que se pueden utilizar modelos de datos existentes o poco normalizados y, finalmente, no es completamente transparente (el programador programa el SQL)

Ambos frameworks de iBatis han sido portados a .NET con notable éxito y popularidad. Ambas versiones ya son releases estables y maduras.

¿Para qué sirve?

El patrón DAO nos ayuda a organizar las tareas de persistencia (guardado, búsqueda, recuperación, borrado, etc.) de los objetos y nos permite definir múltiples implementaciones para un mismo objeto mediante la definición de una interfaz. Por ejemplo, la interfaz UsuarioDao tiene los metodos guardarUsuario(Usuario usuario), obtenerUsuario(String loginId), etc y dos implementaciones: UsuarioMySqlDao y UsuarioLdapDao donde implementamos las operaciones para MySql y Ldap respectivamente. Con iBatis DAO podremos configurar cuándo usar una u otra implementación sin necesidad de modificar código. Además, agregar un UsuarioOracleDao será muy fácil y no implicará modificar código del resto de la aplicación.

Cuando hacemos un análisis y diseño orientado a objetos obtenemos nuestro modelo de clases y también diseñamos el modelo de datos donde se almacena la información. Y siempre nos queda la tarea de conectarnos con una base de datos, crear statements, construir selects/updates/etc, recorrer resultsets y setear atributos de objetos, etc. para guardar, buscar, recuperar, etc. los valores de los atributos de un objeto. iBatis sqlMap simplifica esta tarea resumiéndola a la configuración de ficheros XML, con SQL ANSI o propietario y funciona con prácticamente cualquier base de datos con driver JDBC.

¿Cuándo utilizar iBatis?

Como toda herramienta, no vale para todo. iBatis sqlMap es muy válido cuando:

- Se requiere una curva de aprendizaje rápida y pocos conocimientos previos, y no requiere aprender un lenguaje de consultas como en Hibernate o EJB CMP.
- Se necesita manipular el SQL (para utilizar SQL propietario, optimizarlo, etc.) o se necesita llamar a procedimientos almacenados.
- El modelo de datos existe previamente y no está muy normalizado (aunque lógicamente se puede utilizar con modelos nuevos y normalizados)
- Alto rendimiento y optimización

...pero no para cuando:

- Se requiere una automatización total y transparente de la persistencia
- Se requiere soporte multi-RDBMS (motor de base de datos) transparente y automática

Hay una comparación constructiva con Hibernate en

http://raibledesigns.com/page/rd?entry=hibernate_vs_ibatis

iBatis DAO es válido cuando:

- Se sabe que el RDBMS (motor de base de datos) puede cambiar en el futuro
- La implementación del acceso a datos puede requerir cambios sustanciales en el futuro
- La implementación del acceso a datos puede variar entre un entorno de producción y otro (software comercial en distintos clientes)

iBatis DAO soporta sqlMap, Hibernate, etc.

¿Cómo funciona esto internamente?

Básicamente, programación declarativa y extensión del framework. Es decir, se

configuran ficheros XML y, en iBatis DAO, se extienden clases donde se implementa la interfaz y el comportamiento específico.

Ejemplo:

1. iBatis DAO (opcional)

1. Crear un DAO.XML

```
<!DOCTYPE daoConfig PUBLIC "-//iBATIS.com//DTD DAO Configuration 2.0//EN" "http://www.ibatis.com/dtd/dao-2.dtd">

<daoConfig>

<context>

<transactionManager type="SQLMAP">

<property name="SqlMapConfigResource"

value="es/dxd/km/dao/sqlMap/sqlMapConfig.xml" />

</transactionManager>

<dao interface="es.dxd.km.dao.UsuarioDao"

implementation="es.dxd.km.dao.sqlMap.UsuarioSqlMapDao" />

</context>

</daoConfig>
```

2. Crear una utilidad de configuración tipo DaoConfig

```
public class DaoConfig {

private static final String DAO_XML = "es/dxd/km/dao/dao.xml";

private static final DaoManager daoManager;

static {

try {
```

```
daoManager = newDaoManager();

} catch (Exception e) {

throw new RuntimeException("Description. Cause: " + e, e);

}

}

public static DaoManager getDaoManager() {

return daoManager;

}

public static DaoManager newDaoManager() {

try {

Reader reader = Resources.getResourceAsReader(DAO_XML);

return DaoManagerBuilder.buildDaoManager(reader, null);

} catch (Exception e) {

throw new RuntimeException(

"Could not initialize DaoConfig. Cause: " + e, e);

}

}

}
```

3. Crear la interfaz del Dao para Usuarios, UsuariosDao:

```
public interface UsuarioDao extends Dao {

public int updateUsuario(Usuario usuario);

public int insertUsuario(Usuario usuario);

public int deleteUsuario(String idUsuario);

public Usuario getUsuario(String idUsuario);

public Usuario getUsuarioValidado(String idUsuario, String password);

public List getUsuariosByExample(Usuario usuario);

}
```

4. Crear su implementación:

```
public class UsuarioSqlMapDao extends SqlMapDaoTemplate implements
es.dxd.km.dao.UsuarioDao {

    public UsuarioSqlMapDao(DaoManager arg0) {
        super(arg0);
    }

    public int updateUsuario(Usuario usuario) {
        try {
            return getSqlMapExecutor().update("updateUsuario", usuario);
        } catch (SQLException e) {
            throw new DaoException("Error actualizando usuario. Cause: " + e, e);
        }
    }

    public int insertUsuario(Usuario usuario) {
        try {
            getSqlMapExecutor().insert("insertUsuario", usuario);
        } catch (SQLException e) {
            throw new DaoException("Error insertando usuario. Cause: " + e, e);
        }
        return 1;
    }

    public int deleteUsuario(String idUsuario) {
        try {
            return getSqlMapExecutor().delete("deleteUsuario", idUsuario);
        } catch (SQLException e) {
            throw new DaoException("Error actualizando usuario. Cause: " + e, e);
        }
    }
}
```

```
public Usuario getUsuario(String idUsuario) {  
    try {  
        Usuario usuario = (Usuario) getSqlMapExecutor().queryForObject(  
            "getUsuarioById", idUsuario);  
        usuario.setRoles((List) getSqlMapExecutor().queryForList(  
            "getRolesUsuario", usuario));  
        return usuario;  
    } catch (SQLException e) {  
        throw new DaoException("Error recuperando usuario. Cause: " + e, e);  
    }  
}
```

```
public List getUsuariosByExample(Usuario usuario) {  
    try {  
        Usuario usuarioExample = new Usuario();  
        if (usuario.getApellidos() != null) {  
            usuarioExample.setApellidos("%"  
                + usuario.getApellidos().toLowerCase() + "%");  
        }  
        if (usuario.getNombre() != null) {  
            usuarioExample.setNombre("%"  
                + usuario.getNombre().toLowerCase() + "%");  
        }  
        usuarioExample.setIdUsuario(usuario.getIdUsuario());  
        List usuarios = (List) getSqlMapExecutor().queryForList(  
            "getUsuariosByExample", usuarioExample);  
        // Asignar los roles  
        for (Iterator iter = usuarios.iterator(); iter.hasNext();) {  
            Usuario usuario2 = (Usuario) iter.next();  
            usuario2.setRoles((List) getSqlMapExecutor().queryForList(  
                "getRolesUsuario", usuario2));  
        }  
    }  
}
```

```
return usuarios;

} catch (SQLException e) {

throw new DaoException("Error recuperando usuarios. Cause: " + e, e);

}

}

public Usuario getUsuarioValidado(String idUsuario, String password) {

Usuario usuario = getUsuario(idUsuario);

return usuario.getPassword().equals(password) ? usuario : null;

}

}
```

2. iBatis sqlMap

1. Crear la configuración sqlMapConfig.xml

```
<?xml version="1.0" encoding="UTF-8" ?>

<!DOCTYPE sqlMapConfig

PUBLIC "-//iBATIS.com//DTD SQL Map Config 2.0//EN"

"http://www.ibatis.com/dtd/sql-map-config-2.dtd">

<sqlMapConfig>

<properties

resource="properties/SqlMapConfig.properties" />

<settings cacheModelsEnabled="true" enhancementEnabled="true"

lazyLoadingEnabled="true" maxRequests="32" maxSessions="10"

maxTransactions="5" useStatementNamespaces="false" />

<transactionManager type="JDBC">

<dataSource type="SIMPLE">
```



```
<property name="JDBC.Driver" value="${driver}" />
<property name="JDBC.ConnectionURL" value="${url}" />
<property name="JDBC.Username" value="${username}" />
<property name="JDBC.Password" value="${password}" />
</dataSource>

</transactionManager>

<sqlMap
resource="es/dxd/km/dao/sqlMap/Usuario.xml" />

</sqlMapConfig>
```

2. **La configuración de contexto es preferible sacarla afuera del xml para, por ejemplo, cambiar fácilmente de entornos, en el sqlMapConfig.xml se indicó que estuviera en properties/SqlMapConfig.properties:**

```
driver=org.apache.derby.jdbc.ClientDriver
url=jdbc:derby://localhost:1527/kmdb
username=derby
password=derby
```

3. **Finalmente, se configuran las operaciones y su SQL en Usuario.xml:**

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE sqlMap PUBLIC "-//IBATIS.com//DTD SQL Map 2.0//EN" "http://www.ibatis.com/dtd/sql-map-2.dtd">
<sqlMap namespace="Usuario">

<select id="getUsuarioById" resultClass="es.dxd.km.model.Usuario">
SELECT rtrim(ID_USUARIO) as idUsuario, rtrim(PASSWORD) as
password, rtrim(NOMBRE) as nombre, rtrim(APELLIDOS) as apellidos
FROM APP.USUARIOS WHERE id_usuario = #value#
</select>

<select id="getRolesUsuario"
```

```
parameterClass="es.dxd.km.model.Usuario"

resultClass="es.dxd.km.model.Rol">

SELECT rtrim(R.ID_ROL) as idRol, rtrim(R.NOMBRE) as nombre,
rtrim(R.DESCRIPCION) as descripcion FROM APP.ROLES R,
APP.USUARIOS_ROLES U WHERE R.ID_ROL = U.ID_ROL AND U.ID_USUARIO
= #idUserario#

</select>
```

```
<select id="getUsuariosByExample"
parameterClass="es.dxd.km.model.Usuario"
resultClass="es.dxd.km.model.Usuario">

SELECT rtrim(ID_USUARIO) as idUsuario, rtrim(PASSWORD) as
password, rtrim(NOMBRE) as nombre, rtrim(APELLIDOS) as apellidos
FROM APP.USUARIOS

<dynamic prepend="WHERE">

<isNotNull prepend="AND" property="nombre">

lower(NOMBRE) like #nombre#

</isNotNull>

<isNotNull prepend="AND" property="apellidos">

lower(APELLIDOS) like #apellidos#

</isNotNull>

<isNotNull prepend="AND" property="idUsuario">

ID_USUARIO = #idUserario#

</isNotNull>

</dynamic>

</select>
```

```
<insert id="insertUsuario"
parameterClass="es.dxd.km.model.Usuario">

INSERT INTO APP.USUARIOS (ID_USUARIO, PASSWORD, NOMBRE,
APELLIDOS) VALUES (#idUserario#, #password#, #nombre#,
#apellidos#)

</insert>
```

```
<update id="updateUsuario"

parameterClass="es.dxd.km.model.Usuario">

UPDATE APP.USUARIOS SET PASSWORD = #password#, NOMBRE =

#nombre#, APELLIDOS = #apellidos# WHERE ID_USUARIO = #idUser#

</update>


<delete id="deleteUsuario" parameterClass="string">

DELETE FROM APP.USUARIOS WHERE ID_USUARIO = #value#

</delete>


</sqlMap>
```

¿Licencia?

iBatis es un proyecto que pertenece y utiliza la licencia de Apache Software Foundation, por lo que es Open Source disponible, modificable y comercializable libremente. (todo asesoramiento en este tema será bienvenido)

¿Y para qué este manual básico?

Para simplificar el "first touch" y para explicar iBatis a alto nivel.

No pretende ser un tutorial, una guía del usuario ni un manual de referencia. Ellos ya se han escrito y están disponibles en las direcciones que se mencionan en la sección "¿Dónde encuentro más info?"

¿Dónde encuentro más info?

En la Página Oficial <http://ibatis.apache.org> encontrarán en la sección Documentación unos tutoriales muy buenos aunque extensos, una guía de referencia y el JavaDoc de la API. También tienen una sección de preguntas frecuentes muy completa.

Desde esta página también se puede descargar (recomiendo) el ejemplo oficial de utilización de iBatis: el JPetStore. Viene a ser como el PetStore de J2EE reconvertido a Struts+iBatis.

¿Cómo pongo en funcionamiento iBatis?

Básicamente hay que incluir las librerías en el path (en las aplicaciones web, en

/WEB-INF/lib)

Lo ideal es desplegar el war del JPetStore, descargable desde <http://ibatis.apache.org>, y ver cómo funciona. Hacerle alguna modificación. Comenzar un prototipo desde cero y seguir los tutoriales.

Consejo: ser muy meticuloso con los XML porque la info de debugging es escasa y confusa.



[Puedes opinar sobre este tutorial aquí](#)

Recuerda

que el personal de [Autentia](#) te regala la mayoría del conocimiento aquí compartido ([Ver todos los tutoriales](#))

¿Nos vas a tener en cuenta cuando necesites consultoría o formación en tu empresa?

¿Vas a ser tan generoso con nosotros como lo tratamos de ser con vosotros?

info@autentia.com

Somos pocos, somos buenos, estamos motivados y nos gusta lo que hacemos

Autentia = Soporte a Desarrollo & Formación



[Autentia S.L.](#) Somos expertos en:

J2EE, Struts, JSF, C++, OOP, UML, UP, Patrones de diseño ..
y muchas otras cosas

Nuevo servicio de notificaciones

Si deseas que te enviemos un correo electrónico cuando introduzcamos nuevos tutoriales, inserta tu dirección de correo en el siguiente formulario.

Subscribirse a Novedades	
e-mail	<input type="text"/>
	<input type="button" value="Enviar"/>

Otros Tutoriales Recomendados ([También ver todos](#))

Nombre Corto	Descripción
Patrones de diseño J2EE	Os mostramos una interpretación particular de los patrones de diseño J2EE
Optimización de Serialización Java	Os mostramos una sencilla técnica para mejorar el rendimiento de la serialización de objetos en Java, a través de Streams asociados a buffers en memoria.
Soporte de Asserts en Java 1.4.x	Os mostramos como utilizar los asserts en Java (disponibles a partir de la versión 1.4)
Añadir Scripting a programas Java	Os mostramos como añadir a vuestras aplicaciones soporte de Scripting, gracias a BeansShell. Esto nos permite ejecutar Java generado dinámicamente.
Aplicación de Patrones de Diseño en Java	En este tutorial os mostramos como las técnicas avanzadas de diseño (como patrones de diseño) contribuyen a la contrucción de aplicaciones profesionales en Java.
Introducción a los portales Java con Jahia	Os mostramos como comenzar con el desarrollo de portales (portlets) Java con Jahia
Patrones de GRASP	Os presentamos una introducción a los patrones de asignación de responsabilides y su relación con el proceso unificado.
Analizar ejecución de programa Java	Os mostramos como investigar el comportamiento de vuestros programas Java, en ejecución, a través del profiling.
Aplicaciones con el framework de Microsft .NET	Ejemplo de desarrollo de una aplicación con el framework de Microsft .NET (creación de un servicio deEncuestas Web)
Introducción al Java Web Start	Os mostramos como podéis distribuir aplicaciones de consola de un modo sencillo a través de Java Web Start. Es el mismo principio de los applets aplicado a aplicaciones.

Nota: Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento.

Los contenidos y comentarios de los tutoriales son responsabilidad de sus respectivos autores.

En algún caso se puede hacer referencia a marcas o nombres cuya propiedad y derechos es de sus respectivos dueños. Si algún afectado desea que incorporemos alguna reseña específica, no tiene más que solicitarlo.

Si alguien encuentra algún problema con la información publicada en este Web, rogamos que informe al administrador rcanales@adictosaltrabajo.com para su resolución.

[Patrocinados por enredados.com Hosting en Castellano con soporte Java/J2EE](#)

