



PERSISTENCIA CON MYBATIS



PERSISTENCIA CON MYBATIS

PRIMEROS PASOS

Eric Gustavo Coronel Castillo
gcoronelc.blogspot.com
gcoronelc@gmail.com



Temas



- ¿Qué es MyBatis?
- Arquitectura
- Objetos Principales
- Funcionamiento
- Parámetros de Conexión
- Configuración
- El Bean
- La Especificación
- El Mapeo
- Prueba

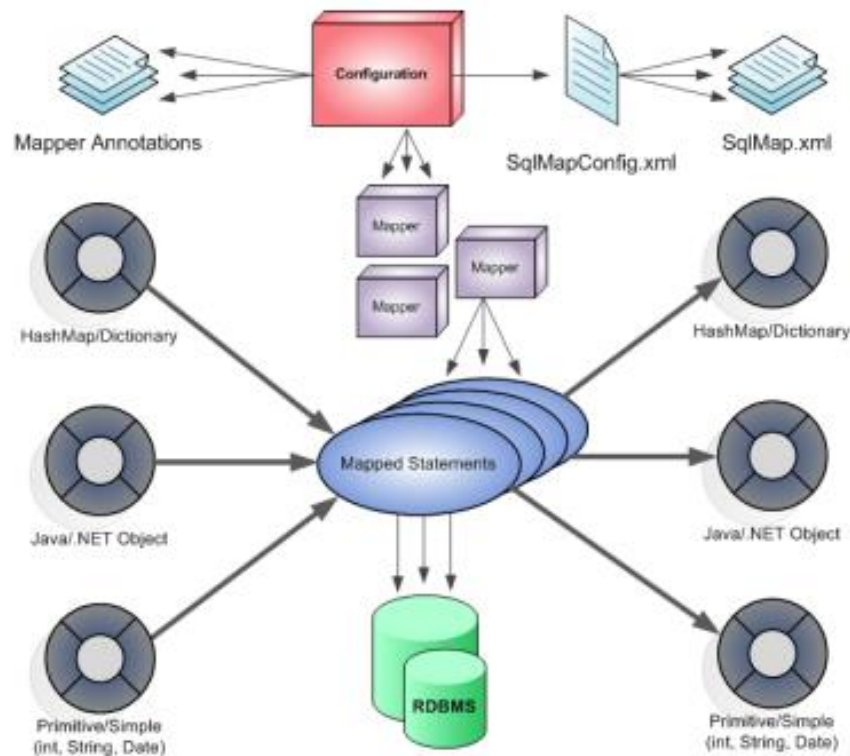


¿Qué es MyBatis?

- MyBatis es un framework de persistencia que soporta SQL, procedimientos almacenados y mapeos avanzados.
- MyBatis elimina casi todo el código JDBC, el establecimiento manual de los parámetros y la obtención de resultados.
- MyBatis puede configurarse con XML o anotaciones y permite mapear MAPs y POJOs (Plain Old Java Objects) con consultas de base de datos.



Arquitectura





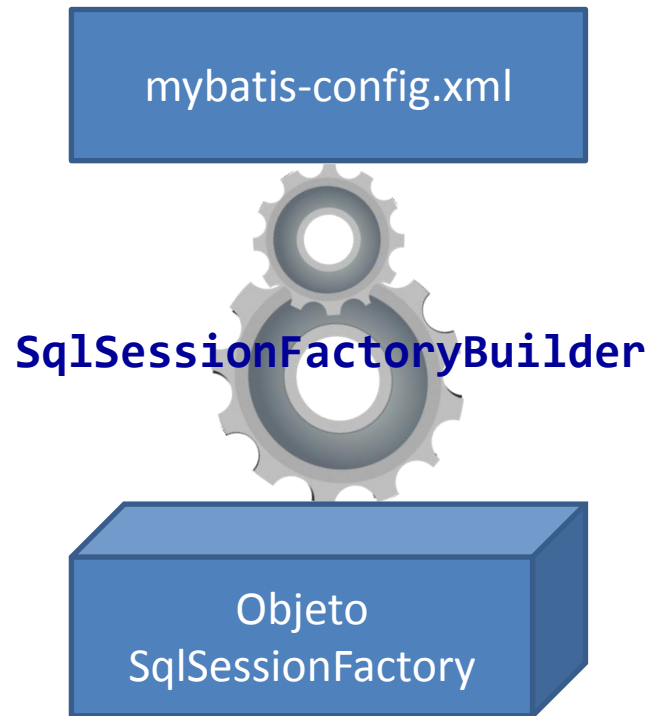
Objetos Principales

■ SqlSessionFactory

Una aplicación que usa MyBatis deberá utilizar una instancia de SqlSessionFactory.

La instancia de SqlSessionFactory se puede obtener mediante el SqlSessionFactoryBuilder.

Un SqlSessionFactoryBuilder puede construir una instancia de SqlSessionFactory a partir de un fichero de configuración XML o de una instancia personalizada de la clase Configuration.



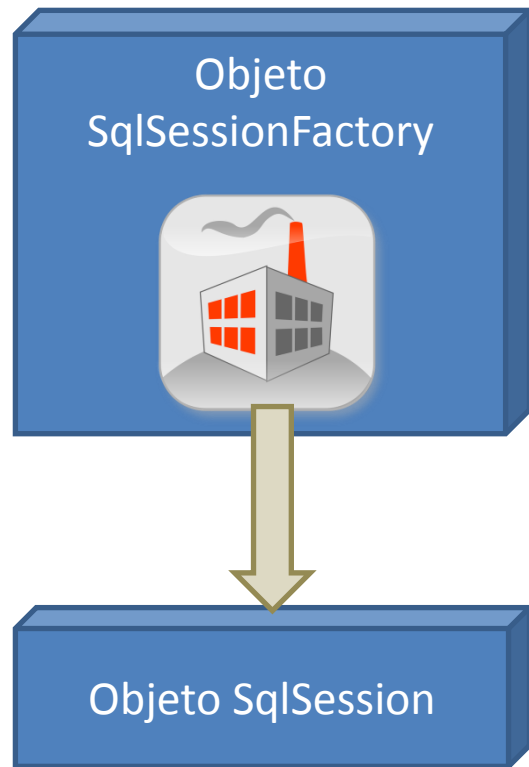


Objetos Principales

- **SqlSession**

Un SqlSessionFactory para crear un SqlSession.

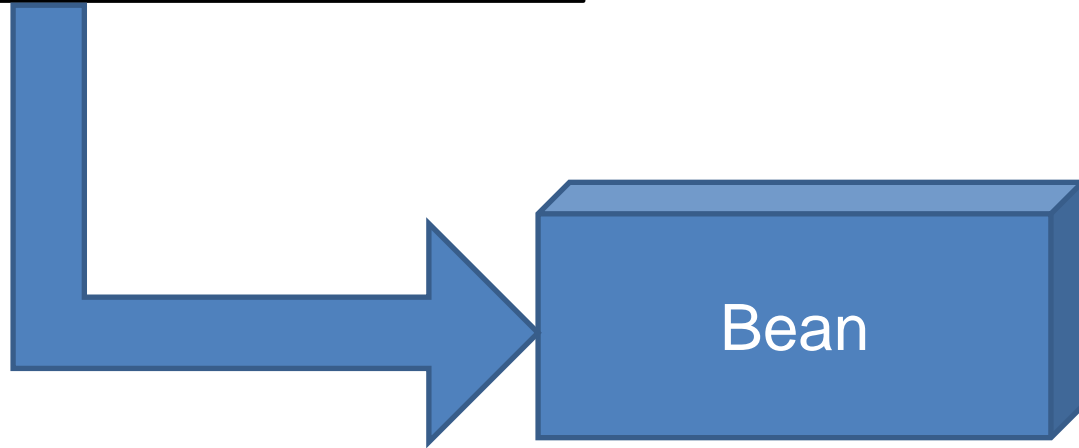
Un objeto SqlSession se utiliza para ejecutar las sentencias mapeadas, hacer commit o rollback, y finalmente, cuando no se necesita se debe cerrar.





Funcionamiento

```
SELECT ...  
FROM empleado  
WHERE chr_emplcodigo='0001'
```





```
driver=oracle.jdbc.driver.OracleDriver  
url=jdbc:oracle:thin:@localhost:1521:XE  
username=eureka  
password=admin
```



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE configuration PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
"http://mybatis.org/dtd/mybatis-3-config.dtd">
<configuration>
  <properties resource="pe/egcc/mybatis/db/configuracion.properties" />
  <typeAliases>
    <package name="pe.egcc.app.model" />
  </typeAliases>
  <environments default="development">
    <environment id="development">
      <transactionManager type="JDBC" />
      <dataSource type="POOLED">
        <property name="driver" value="${driver}" />
        <property name="url" value="${url}" />
        <property name="username" value="${username}" />
        <property name="password" value="${password}" />
      </dataSource>
    </environment>
  </environments>
  <mappers>
    <mapper resource="pe/egcc/mybatis/mapper/EmpleadoMapper.xml" />
  </mappers>
</configuration>
```



```
package pe.egcc.app.model;

public class EmpleadoBean {

    private String codigo;
    private String paterno;
    private String materno;
    private String nombre;
    private String ciudad;
    private String direccion;
    private String usuario;

    public EmpleadoBean() {
    }

    // Métodos setters y getters

}
```



```
package pe.egcc.mybatis.espec;

import java.util.List;
import pe.egcc.app.model.EmpleadoBean;

public interface EmpleadoMapper {

    List<EmpleadoBean> getTodosEmpleados();

}
```



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

<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
"http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="pe.egcc.mybatis.espec.EmpleadoMapper">

  <!-- Consulta de todos los empleados -->

  <select id="getTodosEmpleados" resultType="EmpleadoBean">
    select
      chr_emplcodigo
      vch_emplpaterno
      vch_emplmaterno
      vch_emplnombre
      vch_emplciudad
      vch_empldireccion
      vch_emplusuario
    from empleado
  </select>

</mapper>
```

codigo,
paterno,
materno,
nombre,
ciudad,
direccion,
usuario

Según las
propiedades
del bean.

`EmpleadoBean`



```
SqlSessionFactory sqlSessionFactory = null;
SqlSession sqlSession = null;
EmpleadoMapper empleadoMapper;
List<EmpleadoBean> lista;
try {
    String resource = "pe/egcc/mybatis/db/mybatis-config.xml";
    Reader reader = Resources.getResourceAsReader(resource);
    sqlSessionFactory = new SqlSessionFactoryBuilder().build(reader);
    sqlSession = sqlSessionFactory.openSession();
    empleadoMapper = sqlSession.getMapper(EmpleadoMapper.class);
    lista = empleadoMapper.getTodosEmpleados();
    for (EmpleadoBean empleadoBean : lista) {
        System.out.println(empleadoBean.getCodigo() +
            " - " + empleadoBean.getNombre());
    }
} catch (Exception e) {
} finally {
    . . .
}
```



Gracias..