

```
/* Trabajo de investigación */
```

Spring DAO {

```
<DWF="Desarrollo de Aplicaciones  
con Web Frameworks  
"/>
```

```
}
```



Introducción {

- ¿Qué es Spring DAO?
- Objetivos y beneficios.
- JdbcTemplate.
- Pruebas y Testeo.
- Aspectos de Seguridad.
- Ejemplos Práctico.
- Gracias

`/* Puntos a tratar*/`

}

Spring DAO {

¿Qué es Spring DAO?

```
/*SpringDAO = "Spring Data Access Object" */
```

- Es una herramienta esencial en el desarrollo de aplicaciones que facilita el acceso a bases de datos. Actúa como una capa de abstracción entre el código y las tecnologías de persistencia como Hibernate, JPA o JDBC, permitiendo trabajar con datos sin tener que preocuparse por los detalles técnicos de cada tecnología.

¿De qué forma nos ayuda Spring DAO?

Nos ayuda a simplificar el acceso a bases de datos al servir de capa intermedia entre el código y las tecnologías de persistencia. Facilita la gestión de datos, evita el uso de código SQL complejo y contribuye a desarrollar aplicaciones más eficientes y escalables.

```
/* Abunda mas sobre el punto*/
```

}

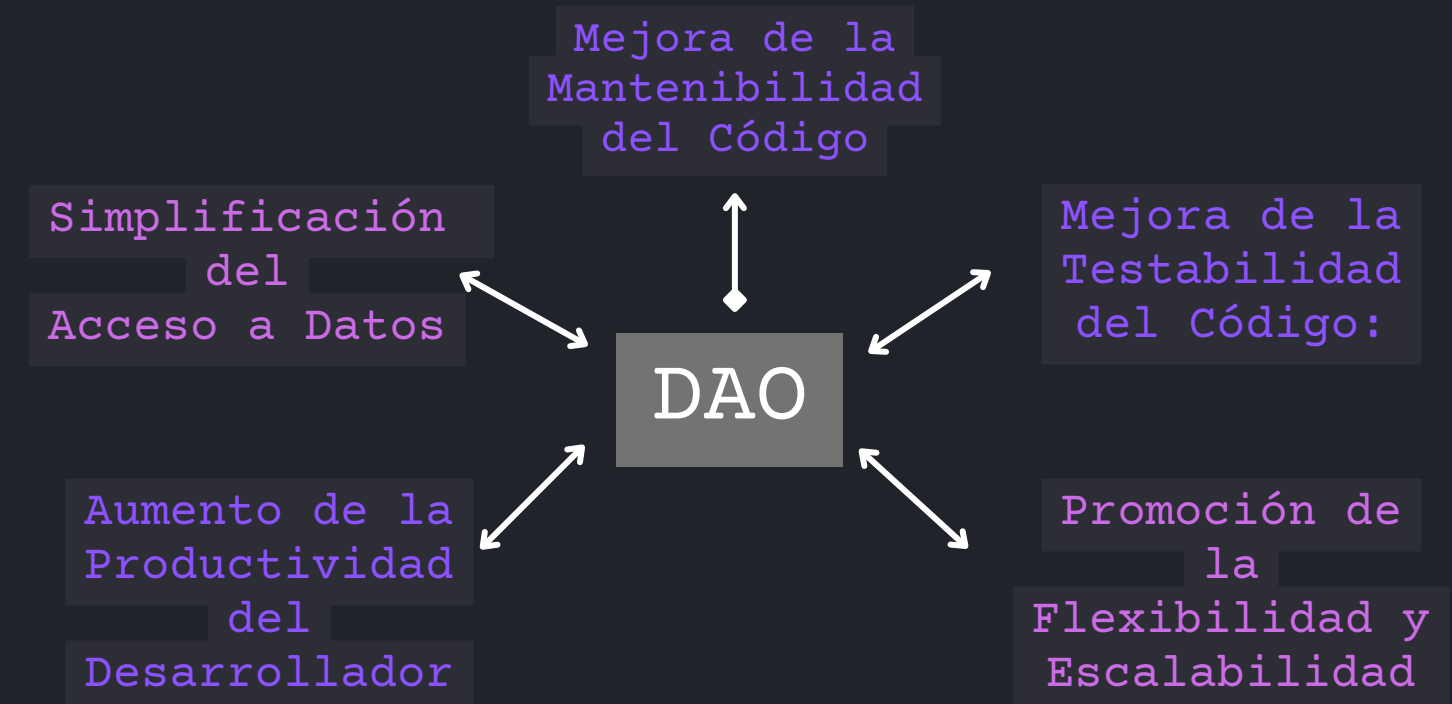
ObjetivosYBeneficios {

Objetivo

`/* Este es el objetivo */`

El objetivo principal es proporcionar una capa de abstracción que simplifica las tecnologías de acceso a datos como JDBC, Hibernate o JPA. Esto permite que los desarrolladores se concentren en la lógica de negocio sin preocuparse por los detalles de la implementación del acceso a datos.

`/* Y estos son los beneficios */`



`/* Y aquí también mas información, por favor
Y habla sobre ORM (Objeto de Mapeo Relacional)
*/`

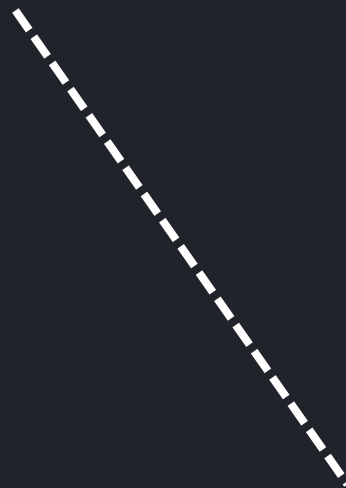
}

```
EstructuraDelPatronDAO
{
```

BusinessObject

DAO

DataSource



TransferObject

```
}
```

JdbcTemplate {

¿Qué es JdbcTemplate?

```
/* JDBC = "Java Database Connectivity  
Template" */
```

- Es una clase central en Spring Framework que simplifica la ejecución de consultas SQL y el manejo de datos en bases de datos relacionales, proporcionando una capa de abstracción sobre JDBC y reduciendo el código repetitivo.

Manejo de excepciones y transacciones

- Manejo de Excepciones: Traduce excepciones JDBC a excepciones Spring, simplificando su manejo.
- Transacciones: Permite gestionar transacciones de base de datos con inicio, confirmación y reversión usando el patrón de plantilla de método.

```
/* No se te olviden los beneficios de  
JdbcTemplate*/
```

}

PruebasYTesteo{

Mocking:

Crea un objeto simulado que imita una dependencia real, permitiendo configurarlo para retornar valores específicos o ejecutar acciones concretas cuando se le llama.

Stubbing:

Similar al mocking, emula un comportamiento específico de una dependencia, programándolo para retornar valores o realizar acciones determinadas al ser invocado con ciertos métodos.

}

AspectosDeSeguridad {

Spring Security es un marco de seguridad para aplicaciones Java que asegura aplicaciones web y empresariales, manejando autenticación, autorización y protección contra amenazas. Al integrarlo con Spring DAO, se garantiza que solo usuarios autorizados accedan y operen sobre datos. Además, Spring Security proporciona anotaciones como `@PreAuthorize` y `@PostAuthorize` para controlar el acceso a métodos DAO, permitiendo definir expresiones de seguridad para verificar los permisos de los usuarios.

`@PreAuthorize`

Define condiciones previas para ejecutar un método DAO, como requerir un rol específico.

`@PostAuthorize`

Realiza verificaciones después de la ejecución del método DAO, como validar permisos de escritura sobre un objeto.

}

Ejemplo practico {

```
String ejemplo = ""; // aquí se  
explica el ejemplo
```

}

```
public class  
Integrantes {
```

```
String AshleyAguilar;  
String FranciscoBayona;  
String NancyCabrera;  
String DavidRoríguez;
```

```
}
```

/* GRACIAS */