INTRODUCCIÓN A SPRING DATA

1.

PERSISTENCIA DE DATOS

PERSISTENCIA DE LA INFORMACIÓN

- La mayoría de las aplicaciones necesitan *persistir* la información.
- Significa que esta sobreviva más allá de una sesión o un determinado tiempo.
- La solución nos la aportan las **bases de datos.**

BASES DE DATOS

- Dos grandes modelos: **relacional** y **NoSQL**

Relacional

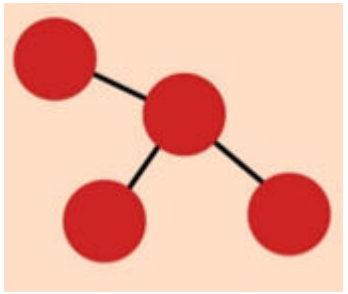
- Tablas, columnas, filas
- Garantía frente a duplicidad.
- Integridad referencial
- Dificilmente escalables

NoSQL

- Diferentes modelos de datos.
- Velocidad, flexibilidad.
- Escalabilidad
- Volumen de datos.

DESFASE OBJETO - RELACIONAL

- En Java usamos objetos (n-dimensiones)
- Las base de datos, tablas (2-dimensiones)
- ¿Cómo salvar este desfase?



Objetos en memoria

Tablas en la base de datos relacional

DESFASE OBJETO - RELACIONAL

```
PreparedStatement ps = con
  .prepareStatement("SELECT * FROM empleados WHERE id = ?");
ps.setInt(1, id);
ResultSet rs = ps.executeQuery();
Empleado result = null;
if (rs.next()) {
        result = new Empleado(rs.getInt("id"),
             rs.getString("nombre"),
rs.getString("apellidos")

rs.getDate("fecha_nacimiento").toLocalDate(),
         rs.getFloat("sueldo"));
}
rs.close();
ps.close();
return result;
```

SOLUCIÓN AL DESFASE OBJETO - RELACIONAL: ORM

- *Object-Relational Mapping*
- Una pieza de software se encarga de traducir objetos en filas y viceversa.

UNA PALABRA SOBRE JPA

- Java Persistence API
- Desde Java se propone un API estándar para tratar la persistencia de datos.
- Java no ofrece ninguna implementación concreta.
- La mayoría de ORMs comerciales, sí lo hacen.

Si JPA es el baile, Hibernate es el bailarín.

UNA PALABRA MÁS SOBRE JPA

- Una facilidad para especificar cómo nuestros objetos Java se relaciona con el esquema de una base de datos
- Una api sencillo para realizar las operaciones CRUD
- Un lenguaje y un API para realizar consultas sobre los datos (JPQL).
- Elementos de optimización (actualización de entidades, captura de asociaciones, caché, …)

2.

SPRING DATA

Y a todo esto,
¿qué hay de
Spring?

¿QUÉ HACEMOS DESDE SPRING?

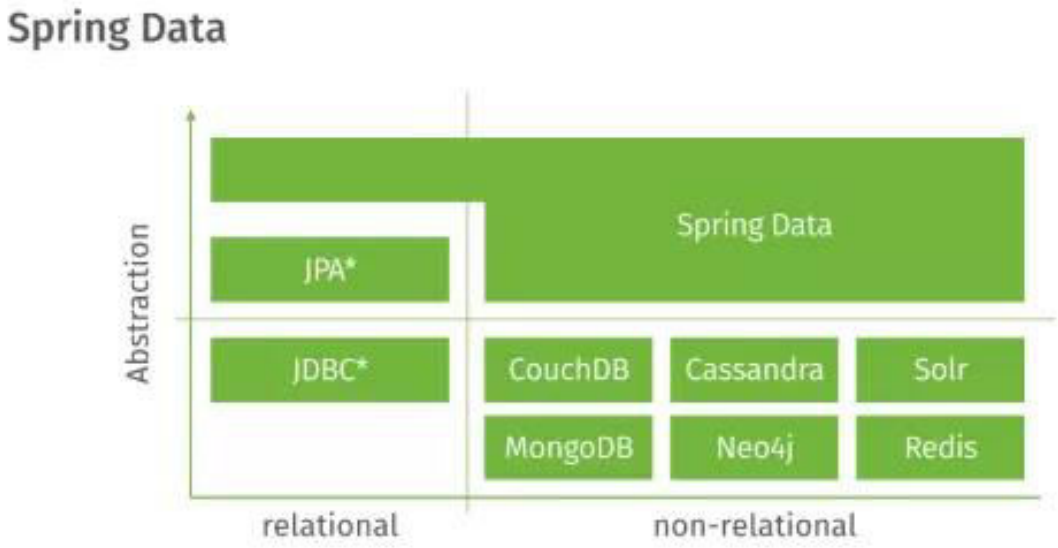
- Se puede usar directamente JDBC
- Se puede utilizar directamente Hibernate
- Podemos usar JPA sobre Hibernate
- También tenemos disponible **Spring Data**.

Spring Data

SPRING DATA

- Ofrece un modelo consistente de programación para acceder a datos.
- Facilita el uso de bases de datos relacionales y NoSQL
- Proyecto *paraguas* para muchos subproyectos.

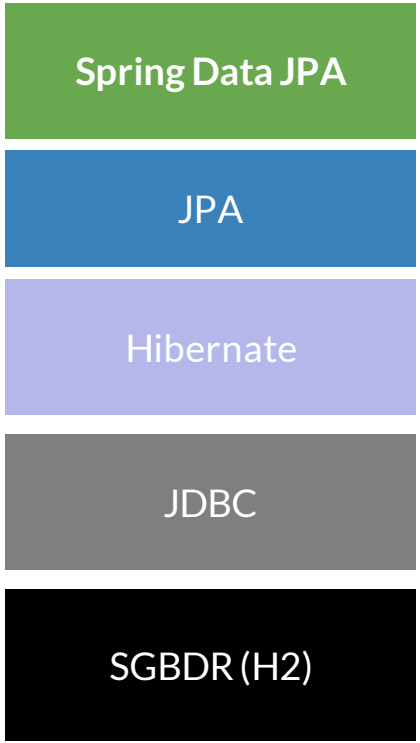
SPRING DATA



https://spring.io/projects/spring-data

SPRING DATA JPA

- Integra todas las funcionalidades de JPA con Spring Data
- Será con el subproyecto que trabajemos.
- Nuestra SGBD será H2.
  - Embebible (Maven)
  - ACID
  - Ligero



DEPENDENCIAS

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>

<dependency>
  <groupId>com.h2database</groupId>
        <artifactId>h2</artifactId>
        <scope>runtime</scope>
</dependency>
```