

# 5

## Accesing of Data from RESTful Service

Java Backend Developer I

# Objetivos

Comprender los conceptos:

- Trabajando con Spring Data Repository
- JDBC
- JPA
- MongoDB
- Redis



# Agenda

Revisión de los siguientes conceptos:

- Spring Data Repository
- JDBC
- JPA
- MongoDB
- Redis



# Spring Data Repository

## Concepto

Spring Data Repository es un proyecto de Spring que contiene muchos sub proyectos que son específicos de una base de datos determinada. Los proyectos se desarrollan trabajando en conjunto con muchas de las compañías que están detrás de estas tecnologías.

Spring Data nos proporciona un modelo de programación familiar y consistente basado en Spring para el acceso a los datos, al tiempo que conserva los rasgos especiales del almacén de datos.

Facilita el uso de tecnologías de acceso a datos, bases de datos relacionales y no relacionales, marcos de reducción de mapas y servicios de datos basados en la nube.

Spring Data ofrece también soporte para acceso a datos de forma reactiva relacionales, no SQL, mensajería, seguridad.



# Spring Data Repository

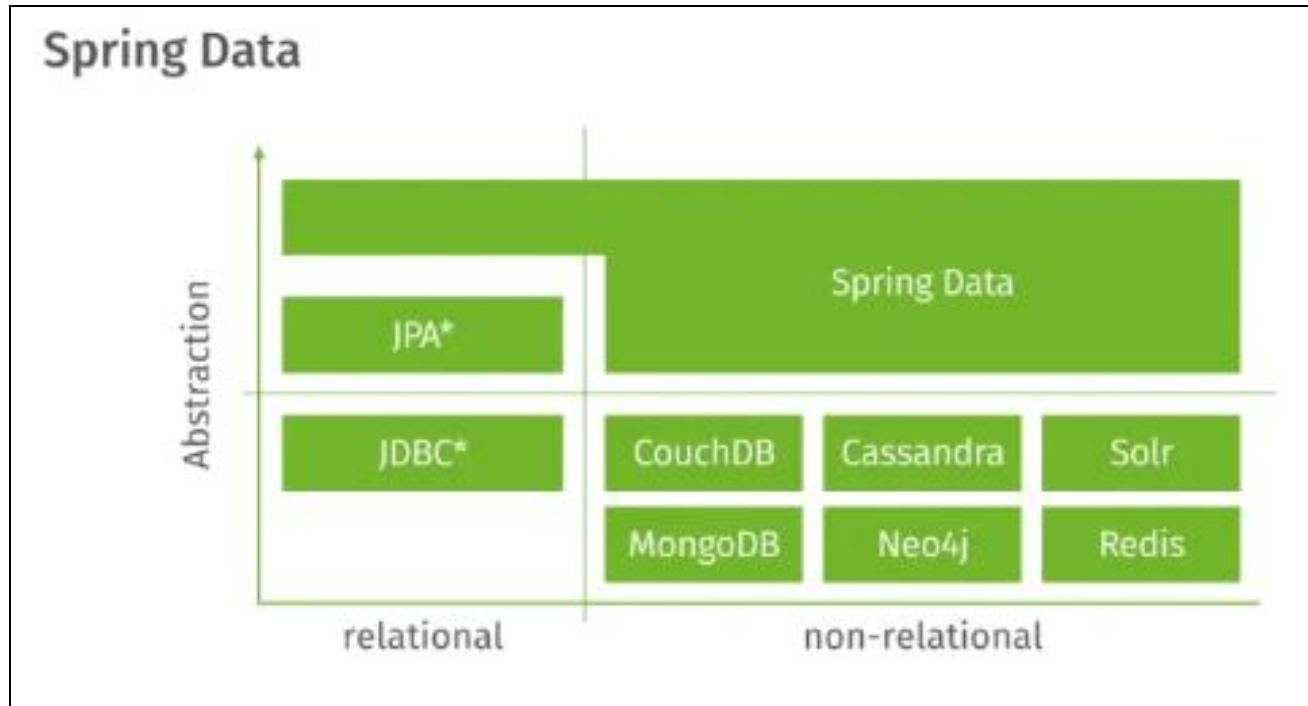
## Características

- Potente repositorio y abstracciones personalizadas de mapeo de objetos
- Consulta dinámica derivada de nombres de métodos de repositorio
- Implementación de clases base de dominio que proporcionan propiedades básicas
- Soporte para auditorías transparentes (creado, modificado por última vez)
- Posibilidad de integrar código de repositorio personalizado
- Fácil integración de Spring a través de JavaConfig y espacios de nombres XML personalizados
- Integración avanzada con controladores Spring MVC.



# Spring Data Repository

## Diagrama



# Trabajando con Spring Data Repository

## Principales módulos

- Spring Data JDBC: soporte del repositorio Spring Data para JDBC.
- Spring Data JDBC Ext: soporte para extensiones específicas de la base de datos para JDBC estándar, incluido soporte para conmutación por error de conexión rápida Oracle RAC, soporte para AQ JMS y soporte para el uso de tipos de datos avanzados.
- Spring Data JPA: soporte del repositorio de Spring Data para JPA.



# Trabajando con Spring Data Repository

## Principales módulos

- Spring Data MongoDB: soporte y repositorios de documentos de objetos basados en Spring para MongoDB.
- Spring Data Redis: configuración y acceso sencillos a Redis desde las aplicaciones Spring.
- Spring Data REST: exporta los repositorios de Spring Data como recursos RESTful impulsados por hipermedia.





# Spring Data JDBC

## Concepto

Spring Data JDBC, parte de la familia Spring Data, facilita la implementación de repositorios basados en JDBC. Este módulo trata con soporte mejorado para capas de acceso a datos basadas en JDBC. Facilita la creación de aplicaciones con tecnología Spring que utilizan tecnologías de acceso a datos.

Spring Data JDBC pretende ser conceptualmente fácil. Para lograr esto, NO ofrece almacenamiento en caché, carga diferida, escritura detrás o muchas otras características de JPA. Esto hace que Spring Data JDBC sea un ORM simple.



# Spring Data JDBC

## Características

- Operaciones CRUD para agregados simples con NamingStrategy personalizable.
- Soporte para anotaciones @Query.
- Soporte para consultas MyBatis.
- Eventos.
- Configuración de repositorio basada en JavaConfig mediante la introducción de @EnableJdbcRepositories.
- Spring proporciona una plantilla JdbcTemplate que facilita el trabajo con bases de datos relacionales SQL y JDBC. La mayoría del código JDBC está sumido en la adquisición de recursos, la administración de conexiones, el manejo de excepciones y la verificación general de errores. JdbcTemplate se encarga de todo ello.



# Spring Data JPA

## Concepto

- Spring Data JPA, hace que sea fácil implementar repositorios basados en JPA. Este módulo se ocupa del soporte mejorado para capas de acceso a datos basadas en JPA. Facilita la creación de aplicaciones con tecnología Spring que utilizan tecnologías de acceso a datos.
- Spring Data JPA tiene como objetivo mejorar significativamente la implementación de las capas de acceso a datos al reducir el esfuerzo a la cantidad que realmente se necesita. Como desarrollador, se escribe sus interfaces de repositorio, incluidos los métodos de búsqueda personalizados, y Spring proporcionará la implementación automáticamente.



# Spring Data JPA

## Características

- Soporte sofisticado para construir repositorios basados en Spring y JPA
- Soporte para predicados Querydsl y, por lo tanto, consultas JPA de tipo seguro
- Soporte de paginación, ejecución dinámica de consultas, capacidad de integrar código de acceso a datos personalizado
- Validación de consultas comentadas de @Query en el momento de arranque
- Soporte para mapeo de entidades basado en XML
- Configuración de repositorio basada en JavaConfig introduciendo @EnableJpaRepositories.



# Spring Data JPA

## Características

- Spring Data JPA API proporciona la clase JpaTemplate para integrar la aplicación Spring con JPA.
- JPA (Java Persistent API) es la especificación de Sun para los objetos persistentes en la aplicación empresarial. Actualmente se utiliza como reemplazo de beans de entidad complejos.
- La implementación de la especificación JPA es proporcionada por muchos proveedores como
  - Hibernate
  - Toplink
  - iBatis
  - OpenJPA etc.



# MongoDB

## Concepto

Es un sistema de base de datos NoSQL, orientado a documentos y de código abierto.

En lugar de guardar los datos en tablas, tal y como se hace en las bases de datos relacionales, MongoDB guarda estructuras de datos BSON (una especificación similar a JSON) con un esquema dinámico, haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida.

## Características

- Consultas ad hoc
- Indexación y Replicación
- Balanceo de carga
- Almacenamiento de archivos



# MongoDB

## Características

- Persistencia y mapeo de eventos del ciclo de vida
- Mapeo de bajo nivel usando abstracciones MongoReader / MongoWriter
- Consulta basada en Java, criterios y DSL de actualización
- Implementación automática de interfaces de repositorio que incluye soporte para métodos de consulta personalizados.
- Integración de QueryDSL para admitir consultas de tipo seguro. Integración geoespacial
- Integración Map-Reduce
- Administración y monitoreo de JMX
- Soporte de GridFS



# MongoDB

## Cuando usar

Su flexibilidad al definir esquemas de datos, nos va a permitir entre otras cosas:

- Insertar documentos (registros) sin tener una estructura rígida al realizarlo.
- Cambiar la naturaleza de los registros, pueden ser pares de llave-valor, estructuras basadas en grafos, documentos o columnas
- Dinamismo pudiendo agregar o quitar propiedades de una entidad a medida que se van agregando registros nuevos.





# Trabajando con Spring Data Repository

## Spring Data y la relación con MongoDB

- Spring Data para MongoDB es parte del proyecto general Spring Data que tiene como objetivo proporcionar un modelo de programación basado en Spring y consistente para nuevos almacenes de datos mientras se conservan las características y capacidades específicas.
- El proyecto Spring Data MongoDB proporciona integración con la base de datos de documentos MongoDB. Las áreas funcionales clave de Spring Data MongoDB son un modelo centrado en POJO para interactuar con una colección de DB MongoDB y escribir fácilmente una capa de acceso a datos de estilo de repositorio.



# Trabajando con Spring Data Repository

## Spring Data y la relación con MongoDB

### Beneficios

- Soporte de configuración de Spring utilizando clases de @Configuration basadas en Java o por XML para una instancia de controlador Mongo y conjuntos de réplica
- Clase auxiliar MongoTemplate que aumenta la productividad realizando operaciones comunes de Mongo. Incluye mapeo de objetos integrado entre documentos y POJOs.
- Traducción de excepciones a la jerarquía portátil de excepciones de acceso a datos de Spring
- Asignación de objetos ricos en funciones integrada con el servicio de conversión de Spring
- Metadatos de mapeo basados en anotaciones pero extensibles para admitir otros formatos de metadatos



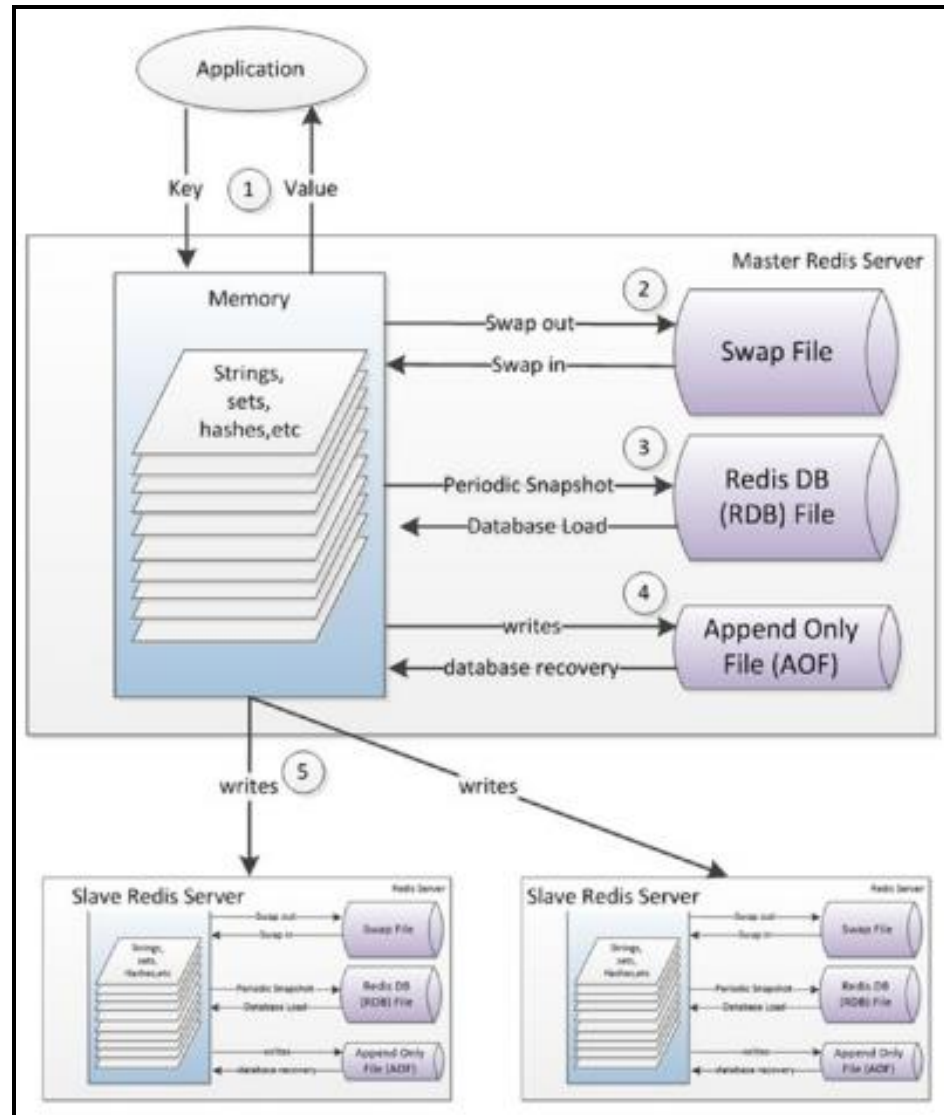
# Redis

## Concepto

- Es un motor de base de datos en memoria, basado en el almacenamiento en tablas de hashes (clave/valor) pero que opcionalmente puede ser usada como una base de datos durable o persistente.
- Está escrito en ANSI C.
- Está liberado bajo licencia BSD por lo que es considerado software de código abierto.



# Redis Arquitectura



# Spring Data Redis

## Características

- Paquete de conexión como abstracción de bajo nivel en múltiples controladores de Redis (Lettuce y Jedis).
- Traducción de excepciones a la jerarquía de excepciones de acceso a datos portátil de Spring para excepciones de controladores de Redis.
- RedisTemplate que proporciona una abstracción de alto nivel para realizar diversas operaciones de Redis, traducción de excepciones y soporte de serialización.
- Soporte de Pubsub (como un MessageListenerContainer para POJOs basados en mensajes).
- Soporte de Redis Sentinel y Redis Cluster.
- API reactiva con el controlador Lettuce.



# Spring Data Redis

## Características

- Serializadores de mapeo JDK, String, JSON y Spring Object / XML.
- Implementaciones de la Colección JDK sobre Redis.
- Clases de soporte atómico.
- Funcionalidad de clasificación y canalización.
- Soporte dedicado para el patrón SORT, SORT / GET y valores masivos devueltos.
- Implementación de Redis para la abstracción de caché Spring 3.1.
- Implementación automática de interfaces de repositorio que incluye soporte para métodos de consulta personalizados utilizando @EnableRedisRepositories.
- Soporte CDI para repositorios.



# Spring Data Redis

## Características

- Proporciona una configuración fácil y acceso a Redis desde las aplicaciones Spring. Ofrece abstracciones de bajo y alto nivel para interactuar, liberando al usuario de preocupaciones de infraestructura.
- RedisTemplate proporciona una abstracción de alto nivel para realizar diversas operaciones de Redis, traducción de excepciones y soporte de serialización.



# Lecturas adicionales

Para obtener información adicional, puede consultar los siguientes enlaces:

<https://spring.io/projects/spring-data-jpa>

<https://spring.io/projects/spring-data-jdbc>

<https://spring.io/projects/spring-data-mongodb>

<https://spring.io/projects/spring-data-redis>





# Resumen

En este capítulo, usted aprendió:

- Spring Data Repository
- JDBC
- JPA
- MongoDB
- Redis

