**Spring**

////////////////////////////////////////

POJO Plain Old Java Object. Se usa para enfatizar el uso de clases simples y que no dependen de un framework en especial

Spring is an open source Java platform

Enables a POJO-based programming model

Using only POJOs is that you do not need an EJB container product such as an application server but you have the option of using only a robust servlet container such as Tomcat or some commercial product

Its modular

Se integra con frameworks existentes

By using JavaBeanstyle POJOs, it becomes easier to use dependency injection for injecting test data

Provides a convenient API to translate technology-specific exceptions (thrown by JDBC, Hibernate, or JDO, for example) into consistent, unchecked exceptions

Lightweight IoC containers

Transaction management interface can scale down to a local transaction (using a single database, for example) and scale up to global transactions (using JTA, for example)

///////////////////////////////////////////////////

Aspect Oriented Programming (AOP)

The functions that span (abarcar) multiple points of an application are called cross-cutting concerns and these cross-cutting concerns are conceptually separate from the application's business logic. There are various common good examples of aspects including logging, declarative transactions, security, caching, etc

The unit of modularity is the aspect

Helps you decouple cross-cutting concerns from the objects that they affect

////////////////////////////////////////////////////////////////////////////////////////

Core. IoC and dependency injection

Beans. BeanFactory

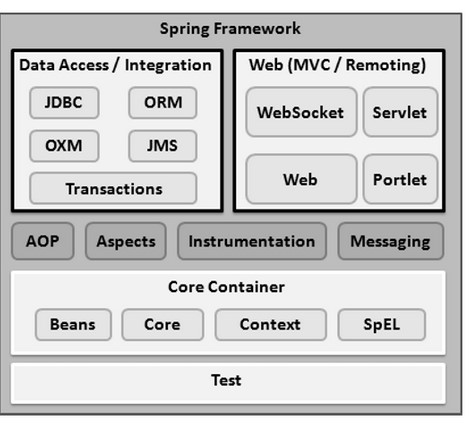
Context. Medium to accesss any objects defined and configured: ApplicationContext

SpEL. Querying and maipulation an object graph at runtime

ORM. Integration layers for object-relational mapping apis: JPA, JDO, Hibernate and iBatis

OXM. Object/XML, mapping implementations for JAXB, Castor, XMLBeans, JiBX and Xstream

Transactions. Transaction management for classes that implement special interfaces and POJOs



Web. Basic web-oriented integration features such as multipart file-upload functionality and the initialization of the IoC container using servlet listeners and a web-oriented application context

AOP. Allows you to define method-interceptors and pointcuts to cleanly decouple code that implements functionality that should be separated

Aspects. Provides integration with AspectJ, which is again a powerful and mature AOP framework

Instrumentation. Provides class instrumentation support and class loader implementations to be used in certain application servers

Messaging module provides support for STOMP as the WebSocket sub-protocol to use in applications. It also supports an annotation programming model for routing and processing STOMP messages from WebSocket clients.

Test (abajo) supports the testing of Spring components with JUnit or TestNG frameworks

//////////////////////////////////////////////////////////////////////////////////////////////////////

Descargar <https://commons.apache.org/proper/commons-logging/download_logging.cgi>

descomprimirlo y ponerlo en C:\

Descargar <https://repo.spring.io/release/org/springframework/spring/>

Descomprimierlo y ponerlo en C:\