1: Inyección por dependencias para poyectos que no se usa spring se puede usar este concepto pasando la rerefencia de la clase concreta ( Interfaz → implementación ) mediante el constructor de la clase hará uso de esta ejemplo

class ClaseDeUso {

private InterfaceEjemplo ejemplo ;

public ClaseDeUso(InterfaceEjemplo ejemplo )

{

this. ejemplo = ejemplo;

}

}

En spirng podria ser de la siguiente manera

@Service

class ClaseDeUso {

@AutoWired

private InterfaceEjemplo ejemplo ;

}

Un concepto usado para el mamjeo de implementaciones

2: Chain Of Responsability : Patron que uso para delegar responsabilidad entre capas de servicio e ir validando entre cada una de estas un ejemplo claro de esto es cla clase de spirng OncePerRequestFilter

Fractory : patrón que uso para crear uno o más objetos para crear una funcionalidad en particular regresando una instancia única para el objeto de utilidad

Singleton: Es básicamente generar una sola instancia de una clase dejando principalmente sus atributos como inmutables es decir finales y asignando su valor solo en el constructor

CallBack: Es un patron que uso para delegar responsabilidad para diferentes estados de una solicitud a algun servicio ejemplo

interface CallBack {

void execute();

void onError();

void OnSuccess();

}

Básicamente la clase que implementa esta interfaz será la encargada de manejar el comportamiento de negocio o funcionalidad de la clase seccionando los diferentes estados .

3: Spring JPA: Capa de spring que ayuda a la persistencia de los datos donde podemos administrar Entidades,Repositorios y diferentes utilidades con Hibernate ya que hace uso del framework internamente , unidades de persistencia .

Spring Web: Capa de spring que administra los endpoints (put,get,delete,post) que será publicados como tal por el servidor de aplicaciónes web

Spring Security: Capa de spring que ayuda a la seguridad de nuestro servicio web y de igual manera nos ayuda a administrar nuestros accesos por medio de (login,tokens ) que son autenticados por esta capa y los diferentes componentes de filtrado .

Spring Batch : Es una capa del framework que esta dedicada a los procesos por lotes y tiene varias funcionalidades entre ellas muy parecido a lo que hace una ETL ya que cuenta con JOBS , STEPS y admnistración de manera periodica como lo son expresiones CRON

4:@Autowired, : Sirve para inyectar una implementación

@Qualifier,Sirve para indicar la implementación en especifica

@PostConstruct, Sirve para indicar acciones que hacer antes de cargar el componente de spirng

@Required , : Sirve para indicar que el objeto es requerido para su validación (Javax validation)

@Bean , :Sirve para definir un objeto administrado por el framework

@Configuration, : Sirve para definir una o varias configuraciones del proyecto

@Transactional : Sirve para indicar que lo que vive en el metodo de ser administrado por un bloque begin → end → Close por la parte de JPA generalmente usado para actualizaciones e inserciones , adicional se pueden indicar otras cosas como el rollback al disparar alguna excepción

@ManagedResource, : Sirve para que spirng identifique un fichero más de valores para el proyecto

5: Si se habla de convertidores de objetos a algun otro tipo de lenguaje tipo JSON es basicamnete el parseo de atrubutos y valores a llaves y valor

6: Un aspecto al menos en spring sirve para agregar una funcionalidad a algún servicio / clase o paquete de clases (servicios) funcionando como un proxy escuchando cuado la clase o clases terminen alguna acción en spring se usa @Aspect para definir el metodo y se anota el paquete en los valores a de anotación y como tal lo que tiene que hace dentro del metodó

.