Java EE | Jakarta : El lado empresarial de Java

Escoger entre jpa e hibernate

Hibernate es la implementación y jpa la especificación

JPA : Es una especificación es como una norma técnica hibernate (proveedor o vendor que adopte estas especificaciones ) es un producto de jpa s una norma técnica en hibernate se toma una especificación de JPA se escogería hibernate puedo hacer sw es una implementación de la especificación JPA

Especificación JDBC sus implementaciones son: los conectores para postgresql , conectores para MySQL para Oracle y SQL SERVER La especificación puede ser una Api Java

Cuando se hable de JSF si se quiere hablar de solo spring puro seria MVC

Si se quiere hablar de CDI en spring seria DEPENDECIE INYECTION del spring IOC container si se quiere hablar de Managed Beans en spring serian clases como steriotipo como componentes, repositorios jsr 330 es una norma técnica de como hacer las inyecciones de dependencias, Proveedores de terceros han implementado esta especificación

Spring Vs

Spring Boot te permite trabajar de forma productiva con el framework spring

Spring security forma parte de spring framework? No forma parte de spring framework es un proyecto independiente que se puede acoplar al sistema de spring framework

Spring data acceso a bd de datos relacionales y no relacionales

Spring cloud temas de microservicios

Servicios rest permiten comunicar backend con front-end

Cuál es el beneficio de usar la anotación @SpringBootApplication es una anotación que resume otras anotaciones la más importante es @ComponentScan es una anotación importante porque va a buscar por más anotaciones de spring para hacer las configuraciones del proyecto

Inyección de dependencias (repo = dao)

Cliente: Ejemplo Angular

**CONTROLLER**

**Model**

**REPO/DAO**

**SERVICE se dispone de a lógica de negocio del proyecto**

DAO Acceso a datos

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

En el controlador se toma la recepción de datos y envíos de datos en el service se suele hacer la lógica de negocios es decir validaciones que son propias de un requerimiento como mayor de edad, esta vigente o no la cláusula, validaciones, lógica y en la parte de acceso de datos directamente las consultas que implican a extraer información como de un repositorio de bd relacional y no relacional .

Un servicio rest es un componente de sw que me permite comunicarme entre más componentes no necesariamente estos componentes tienen que estar en el mismo lenguaje pero se van a comunicar sobre el protocolo HTTP

@RequestMapping() se pone una dirección al recurso quede 26:45 video 2

@GetMapping dice que cuando entren a la url en spring se trabaja con el concepto de inyección de dependencias sirve para abstraer el código o desacoplar el código y gestionar mejor las instancias a estas instancias en spring se le conoce como Beans (new Clase()) son como objetos manejados por un contenedor de spring. Spring e su memoria tiene algo conocido como Spring IoC Container

Spring IoC Container

Spring IoC Container es un contenedor donde en la memoria del framework va estar prácticamente alojándose todos los vean o todos los objetos que uno va a crear a utilizar a lo largo del proyecto

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Tiene alcance singleton

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Spring pegunta internamente existe alguna implementación de eta interface que me piden con @Autowired

Cual es la diferencia entre controller es un esteriotipo y restcontroller no lo es por lo tanto controller genera un bean y con restcontroller preparas una clase para un servicios Rest puedas acceder a ella por un punto de acceso o endpoint quede minuto @Component cuando

Resumen : Los @Service se usa para lógica de negocios, los @Repository para acceso de datos y los controller se utilizaban más cuando se trabajaba con jsp , los components son como utilitarios se usan cuando no se sabe como catalogar una clase ver video 3 del modulo 1.

Application.properties este archivo hace las configuraciones de spring

Tomcat embebido JNDI

**Primer bloque a que motor de bd me voy a conectar**

**Se indica que motor de bd voy a utilizar**

spring.jpa.database=postgresql

Es para expresar que al momento de ejecutar el proyecto el código interno se muestren las consultas sql

spring.jpa.show-sql=false

Es una abstracción que el mismo spring ha personalizado para trabajar de una forma más productiva pero internamente la implementación de esa abstracción es hibernate core . La parte de ddl-auto cuando se genera un proyecto desde cero

spring.jpa.hibernate.ddl-auto=update

**Segundo bloque url de conexión**

Establece esa conexión a la clase para conectar con postgres

spring.datasource.driver-class-name=org.postgresql.Driver

spring.datasource.url=jdbc:postgresql://localhost/mediappbd13

spring.datasource.username=postgres

spring.datasource.password=a1b2c3d4e5

VER 23:23 video 4

El update detecta solo nuevas tablas, columnas y nuevas relaciones

JpaRepository es una interfaz propia de spring data proporciona métodos para hacer inserciones, modificaciones, eliminaciones y algunas consultas adicionales

Optional es una clase especial de java que prácticamente va evitar que se devuelva el null a secas ver video 7 modulo 1

Con @RequestBody el Json (De postman ) sea transportado a un objeto Java

Con @PathVariable capturamos el valor o id que venga en la dirección de la solicitud

Interfaz de usuario gráfica

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación, Sitio web, Teams

Descripción generada automáticamente

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

No es un IDE, es un proyecto a parte para trabajar con Spring Framework

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

Depende de la implementación para la manipulación de una bd (creación)

Interfaz de usuario gráfica

Descripción generada automáticamente

No, ya que los WSDL son para el tema de servicios soap

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

No, solo la puede heredar

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

No, ya que el mínimo es de 8 en adelante

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

Abstraer código, gestionar instancias es un tema de desacoplar el código

Interfaz de usuario gráfica, Aplicación, Sitio web, PowerPoint

Descripción generada automáticamente

Ver cual es el repositorio remoto

Interfaz de usuario gráfica, Sitio web

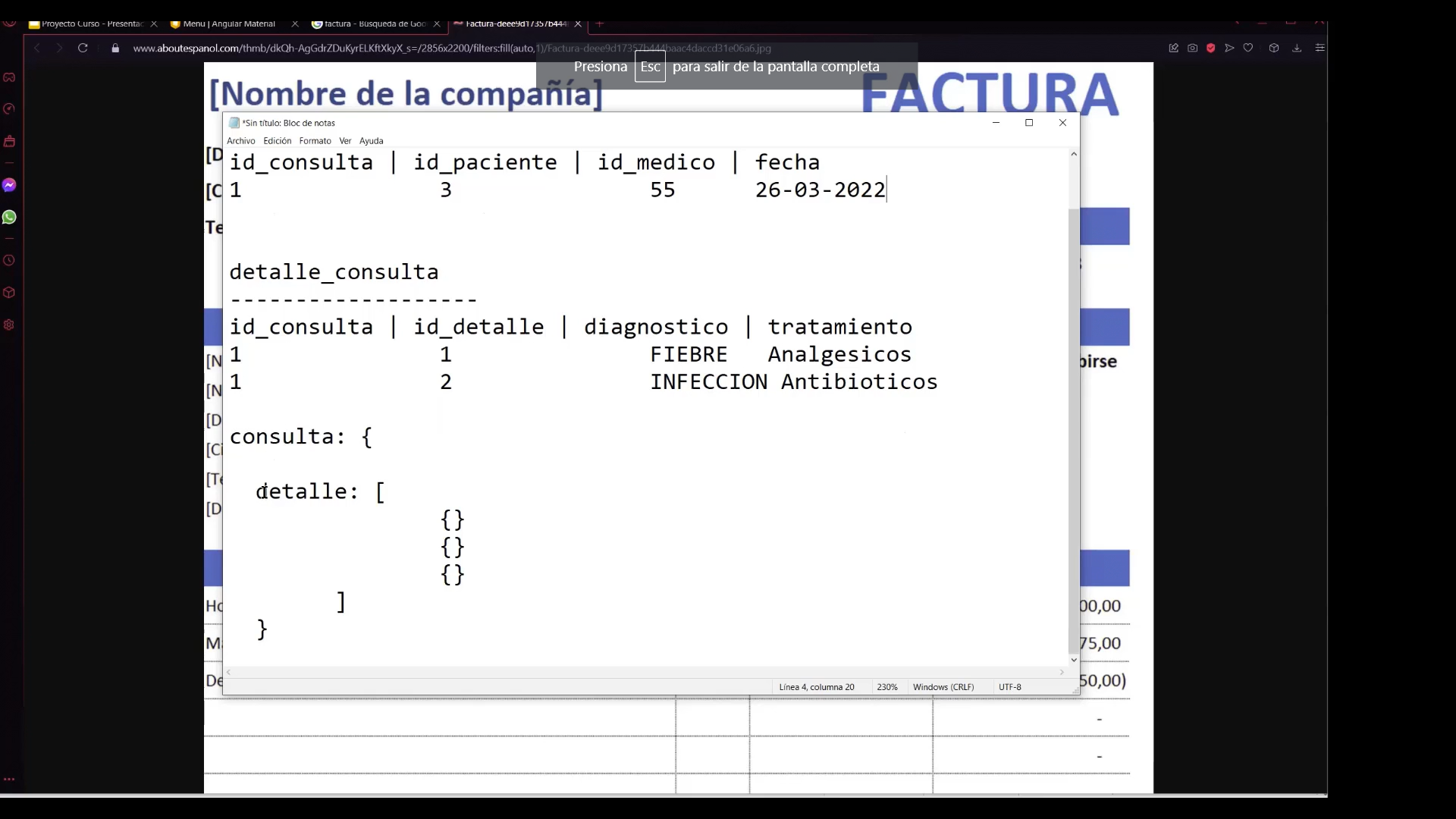
Descripción generada automáticamente

La 5???

**Spring DATA JPA - Mapeo de Tablas simples**

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente



06:35 video 2 modulo 2

Cuando se usa private Integer edad y private int edad; dependerá del escenario planteado en base de datos si fuera NULL en edad seria integer ósea si el valor en la bd acepta NULOS SE USA Integer los procedimientos s pueden consumir en JPQL

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

@JoinColumn() enlaza dos tablas por id dice lo siguiente indicame en la tabla actual el nombre de la columna que quieres crear para representar la relación que tiene con otra

Quede en 38:47 video 2 modulo 2

La diferencia entre un List y Set es la duplicidad de los datos , al utilizar set es que interesa que los elementos que se van guardando en esta colección no tienen que ser duplicados

@OneToMany sobre la tabla consulta no hace que se crea una columna en base de datos es simplemente como apoyo a los trabajos como maestro detalle

Texto

Descripción generada automáticamente

La entidad Consulta Examen es una tabla intermedia entre consulta y examen de muchos a muchos update detecta nuevas tablas, nuevas relaciones y nuevos campos. @IdClass(ConsultaExamen.class) @IdClass va a buscar por una clase (definición de la clase primaria) que este esta definición del contenido de las llaves primarias compuestas de una tabla intermedia de muchos a muchos. Para que la clase public class ConsultaExamenPk sea leída por la clase public class ConsultaExamen se debe agregar la anotación en public class ConsultaExamenPk @Embeddable.

De objeto java a base de datos serializable, Permite traer valores de otras relaciones con fetch = FetchType.EAGER.

Ver video 4 modulo 2

DRY

Dont

Repeat

Yourself

Los genéricos deben estar en la definición de la interfaz , en un conceptos de genéricos las letras que se coloquen no importan

Primero refactorizar el tema de los service después service impl y después repo

Ver video 5 de modulo 2

Modelo de madurez de Richarson hateoas documentación de respuestas

Quede en video 5 de modulo 2 13:31

En el nivel 2 hay que especificar el recurso

Ver el siguiente video

Uso de DTO con ModelMapper, Validaciones y Manejo de idiomas (i18n) DTO

El escenario actual alto acoplamiento entre modelo de bd y lo que el cliente necesita

Clase independiente que converse siempre con el cliente al margen que el modelo cambie o no. El DTO se comunica con el frontend para que este sea el contrato de respuesta esta es la estructura que el frontend va a leer validar edades y otro atributos se evaluarían en service Los DTO conversan con la capa controller

El mapper es para transformar de un tipo A a un tipo B

Quede video 6 modulo 2 13:47 hablan de ventaja de usar DTO estas desacoplando el código del cliente que va a consumirte

Importaciones agregadas :

implementation group: 'org.modelmapper', name: 'modelmapper', version: '3.1.0'

implementation group: 'org.springframework.boot', name: 'spring-boot-starter-validation', version: '2.7.0'

quede en el minuto 32:58 video Uso de DTO con ModelMapper, Validaciones y Manejo de idiomas (i18n) modulo 2 ver modulo 3 video 1 02:52

quede en 10:53 modulo 3 video 1

Maneja los errores que se den durante la aplicación ResponseExceptionHandler

@ControllerAdvice busca o intercepta estas anotaciones como: @ExceptionHandler

Clase abstracta para declarar métodos y que en otro lugar los implementemos

VER MODULO 3 VIDEO 2 quede 05:07

Spring HATEOAS

implementation group: 'org.springframework.hateoas', name: 'spring-hateoas', version: '1.5.1'

La idea es generar una respuesta donde dentro de esa respuesta hay información complementaria esa información es libre a la que se quiera colocar

Ver modulo 3 video 3

Swagger Open API & Restful Stress documentar api

**Springdoc OpenAPI UI**

<http://localhost:8080/v3/api-docs>

[**http://localhost:8080/swagger-ui/index.html#/**](http://localhost:8080/swagger-ui/index.html#/)

Estresar servicios, el estrés de un servicio es cuando mediante un sw se puede simular demasiados clientes. Quede en minuto 7:19 en modulo 3 video 3.