Camilo Aranda – Desarrollador de Software

Actividad 3

1. Organización de paquetes y código
2. ¿Qué son los servicios de dominio?

Son aquellos donde se agregan funciones que sirven para calcular algún valor y que serán utilizados por las clases del dominio.

1. ¿ Qué representan los gateways?

Por el lado del core representan las especificaciones de las funciones que se van a usar en el controlador de la aplicación. Por el lado de la infraestructura son las implementaciones que van a conectarse con la base de datos o también llamados repositorios.

1. ¿Cuáles son las partes que componen un Gateway?

Por el lado del core un Gateway está compuesto por las interfaces que que van a utilizar en la implementación del Gateway del lado de la infraestructura. Está compuesto por funciones como crear, listar, editar etc.

1. ¿Qué tipo de clases están dentro del paquete Core de la aplicación?

Están los paquetes de dominio, gateways y servicios. Las clases de dominio son las que respresentan las validaciones necesarias de la aplicación, en los gateways se encuentran las interfaces que conectarán a la base de datos y en los servicios, estarán funciones que calculan datos.

1. ¿Cuál es la diferencia entre el paquete Core y el paquete Infrastructure?

El paquete Core se compone de los paquetes de dominio, gateways y servicios, mientras que el paquete Infrastructure, comprende los paquetes de configuración, controlador y gateways de implementación . Básicamente en el paquete core es más interno que el de infrastucture, ya que infrastructure contiene las clases y funciones necesarias para conectarse a la base de datos, luego usa el paquete core para pasar por el dominio para luego ir al controlador que conecta con los dto para exteriorizar los resultados, o de forma inversa para ingresar información a la base de datos.

1. ¿ Cuál es el fin del paquete share?

Se compone de clases que contienen validaciones extra, para ser utilizadas en la clase controlador.

1. ¿Cómo se diferencia un servicio de aplicación y un servicio de dominio?

El servicio de dominio son funciones para calcular operaciones, los servicios de dominio sirven para retornar u obtener los valores por medio del protocolo http.

1. ¿Dónde debe ir la configuración de la conexión de una base de datos?

Debe ir en la clase DatabaseConfiguration.java que se encuentra en el paquete Configuration, que a la vez se encuentra dentro del paquete de Infraestructura.

1. Estructuras de control
2. Para que sirve la estructura de control “try-with-resources” y como se diferencia de un try simple?

Sirve para nombrar de forma más simple los llamados a otras funciones y en caso de tener un error de respuesta, se encargue de responder con una excepción del catch que contiene, mientras que son un try simple se debe crear un finally que a la vez dentro de él se crea una excepción por cada llamado de esas funciones, siendo un proceso más largo con el try simple.

1. Java y SQL
2. Para que sirve un DataSource?

Para configurar una conexión a una base de datos

1. ¿Que desventajas existen de utilizar el datasource nativo de java vs una librería de hikariCP?

La configuración es mas compleja si se utiliza un datasource nativo de java que un librería de hikariCP

1. En términos simples ¿Qué es un Resultset?

Es un cursor que apunta a cada registro de la base de datos para obtener la información.

1. ¿ Por que es mala practica concatenar cadenas de texto y variables para generar una sentencia SQL?

Porque se podrían eliminar la información de los registros de esa consulta SQL.

1. Spring
2. ¿En cuales casos es recomendable utilizar la anotación @Configuration?

Para configurar la conexión a una base de datos

1. ¿Se puede utilizar una anotación @Bean en una clase?

Si en la clase de configuración de una base de datos o datasource

1. ¿Es un Bean de Spring lo mismo que un Bean de java?

No, un Bean de java es una clase con un constructor, getters y setters, mientras que un Bean de Spring es como convertir un método en una dependencia.

1. Cositas varias

¿Qué es un java Bean?

Es un Bean de java es una clase con un constructor, getters y setters.