**Sesión 4: “Preparando el modelo de datos”**

* Definición del modelo de datos
* Definición de las tablas y las entidades
* Definición de clases y relaciones

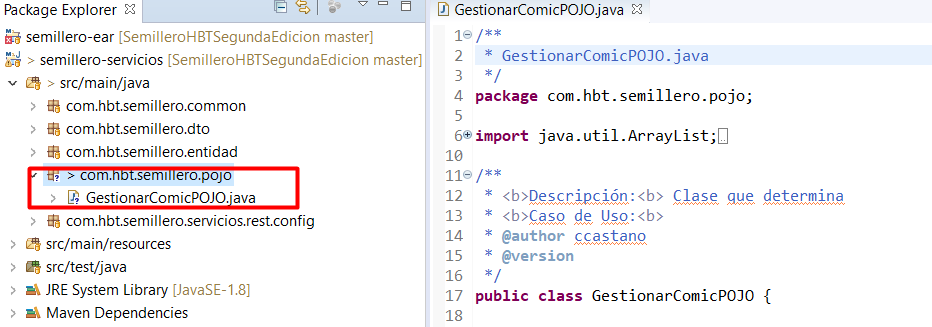
**Objetivos de la sesión:**

1. Realizar un POJO para poder manipular las estructuras de datos creados.
2. Realizando lógica de negocio

**Desarrollo de la sesión**

1. Realizar un POJO para poder manipular las estructuras de datos creados.

Un pojo es una clase java sencilla que no implementa ni extiende ninguna otra clase. En el paquete com.hbt.semillero.pojo crear la clase GestionarComicPOJO.java



Definiendo una estructura de datos para almacenar los comics se requiere utilizar una lista que permite ir almacenando los comics en memoria, esto se hace de la siguiente manera:

private List<ComicDTO> listaComics = null; //Agregar los get y set de la lista

La anterior sentencia define el modificador de acceso private que determina que solo los métodos de la misma clase pueden usar la lista. También determina que es un java.util.List que va a contener solo objetos de tipo ComicDTO y hace su correspondiente inicialización.

Se debe crear el método public void crearComicDTO(){} con el objetivo de llenar uno a uno los datos relacionados con el comic

ComicDTO comic = new ComicDTO();

comic.setId("100");

comic.setNombre("EL EJÉRCITO DEL CAPITÁN AMÉRICA");

comic.setEditorial("Panini Comics ");

comic.setTematica("AVENTURA");

comic.setColeccion("BIBLIOTECA MARVEL ");

comic.setNumeroPaginas(128);

comic.setPrecio(new BigDecimal(50));

comic.setAutores("Phillippe Briones, Roger Stern ");

comic.setColor(Boolean.TRUE);

comic.setFechaVenta(LocalDate.now());

comic.setEstado("ACTIVO");

comic.setCantidad(200L);

Después adicionar el nuevo comic a la lista listaComics

¿Como se puede modificar el nombre a un comic ya almacenado en la listaComics?

Crear el método: public void modificarComicDTO(ComicDTO comicModificar) {}

Validar que la lista tenga elementos

Utilizar un ciclo que permita recorrer los elementos de listaComics (while)

Validar que el elemento a modificar sea el elemento que se está iterando

Realizar la modificación

Salir del ciclo.

¿Como se puede eliminar un comic ya almacenado en la listaComics?

Crear el método: public void eliminarComicDTO(ComicDTO comicEliminar) {}

Validar que la lista tenga elementos

Utilizar un ciclo que permita recorrer los elementos de listaComics (for each)

Validar que el elemento a eliminar sea el elemento que se está iterando

Realizar la eliminación

Salir del ciclo.

¿Como se pueden ver toda la información un comic ya almacenado en la listaComics?

Crear el método: public ComicDTO consultarComicDTO(String idComic) {}

Validar que la lista tenga elementos

Utilizar un ciclo que permita recorrer los elementos de listaComics (for)

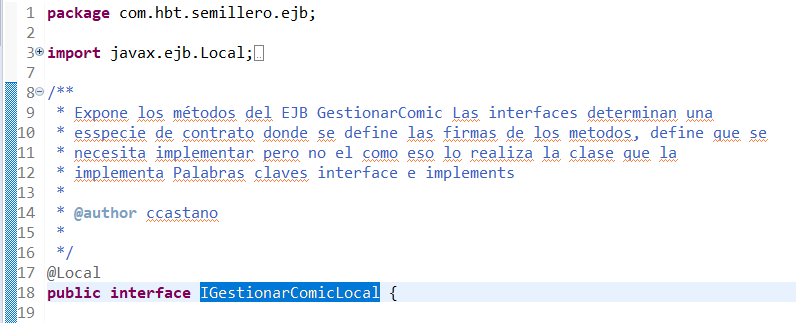
Validar que el elemento a consultar sea el elemento que tenga el mismo idComic que se ingresó por parámetros

Retornar el elemento

Salir del ciclo

1. Realizando lógica de negocio

Crear una interface local de negocio, en el paquete “com.hbt.semillero.ejb” y crear la interface IGestionarComicLocal.java en donde se van a definir que métodos de negocio se requieren para resolver la gestión del comic



Solo se requieren las firmas de los métodos

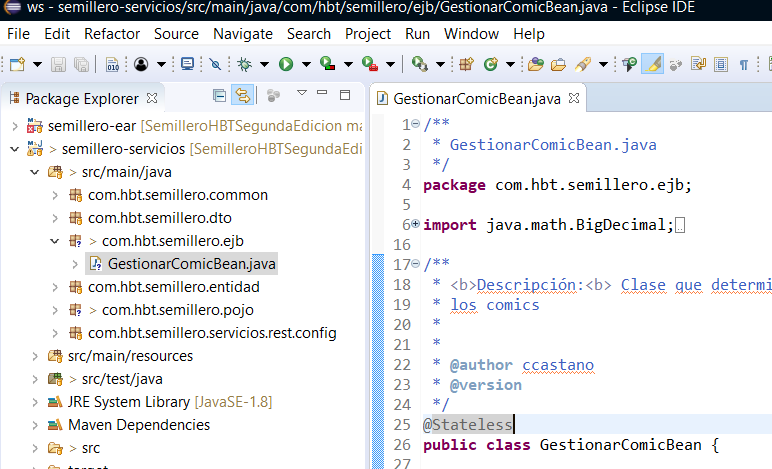
public void crearComic(ComicDTO comicNuevo);

public void modificarComic(Comic comicModificar);

public void eliminarComic(Comic comicEliminar);

public ComicDTO consultarComic(String idComic);

Crear un bean de negocio @Stateless en el mismo paquete GestionarComicBean.java que implemente la interface IGestionarComicLocal



Se requiere una forma de interactuar con la entidad Comic y que permita hacer el manejo de esta. Para ello se requiere crear un objeto javax.persistence.EntityManager

@PersistenceContext

private EntityManager em;

¿Como se puede crear un comic y almacenarlo en la Base de datos?

¿Como se puede modificar el nombre a un comic ya almacenado en la Base de datos?

¿Como se puede eliminar un comic ya almacenado en la Base de datos?

¿Como se puede obtener los comics ya almacenado en la Base de datos?

**Ejercicio sesión 3**

* Crear dos comics más en la listaComics con la información de la tabla COMIC
* Realizar los casos de prueba para verificar que se agregan los 3 comic, se modifica el comic con id 2 y se elimina el comic con id 1
* Lectura “Conceptos básicos de java.docx”

**Conceptos adicionales**

**Enlaces de interés**

* Documento “Rest vs WebServices.pdf”