

ACCESO A DATOS

UD05: PRÁCTICA 2 - HIBERNATE

Realiza un proyecto Java denominado **adt5_practica2** que contenga todo lo que se pide a continuación. Para cada proyecto crearemos los paquetes necesarios con las clases necesarias con su correspondiente clase App.java donde estará su método *main()*.

Para esta práctica vamos a trabajar con la estructura que hemos utilizado en los ejemplos de la unidad.

Leer toda la práctica para saber lo que se pide incluida la ayuda final para organizarla mejor.

Ejercicio 1

Realiza un pequeño programa en Java utilizando Hibernate con MySQL para interactuar con una tabla simple de Seguro con la siguiente información:

Seguro
int IdSeguro
String nif
String nombre
String ape1
String ape2
int edad
int numHijos
Date fechaCreacion

<<Table>> Seguro
INTEGER IdSeguro
VARCHAR nif
VARCHAR nombre
VARCHAR ape1
VARCHAR ape2
INTEGER edad
INTEGER numHijos
TIMESTAMP fechaCreacion

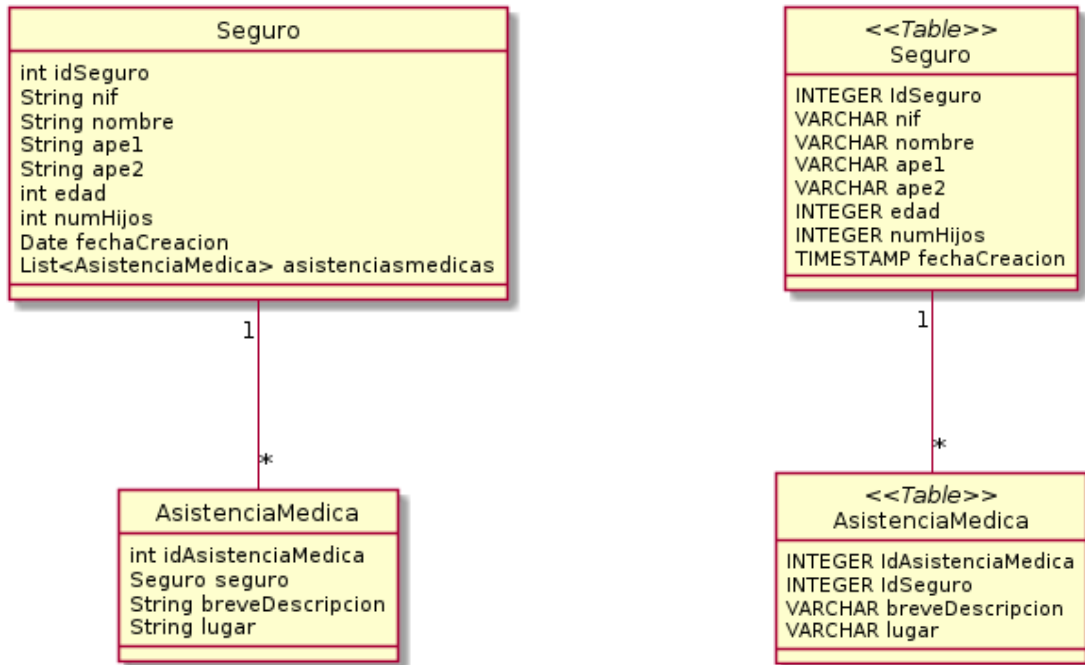
El programa debe realizar las operaciones de insertar, leer, actualizar y borrar de la entidad Seguro usando las anotaciones de JPA con Hibernate.

Ten en cuenta, que la creación de la tabla se hará de forma automática mediante Hibernate.

NOTA: '*java.util.Date*' ha sido reemplazado por las clases '*java.time*' en versiones más recientes de Java (a partir de Java 8). Por lo tanto, para representar solamente una fecha (sin hora), podemos utilizar la clase **LocalDate**. Para sacar la fecha de hoy se utiliza **LocalDate.now()** .

Ejercicio 2

En este segundo ejercicio vamos a crear relaciones entre diferentes entidades.



Modifica el ejercicio anterior para incluir las siguientes tareas:

- Añade la clase Java (entidad) `AsistenciaMedica`
- Realiza una relación uno a muchos entre `Seguro` y `AsistenciaMedica`.

Para ello usa las anotaciones de JPA.

La clase `Main` deberá incluir el siguiente código de creación de los distintos objetos:

```

Seguro seguro =
    new Seguro(321, "12345678Z", "Juan", "Cháfer", "Bellver", 66, 3, LocalDate.now());

AsistenciaMedica asistenciaMedica1 =
    new AsistenciaMedica(1, "Médico de cabecera", "Mislata", seguro);

AsistenciaMedica asistenciaMedica2 =
    new AsistenciaMedica(2, "Operación de bypass", "Sevilla", seguro);
  
```

Ahora añade a la clase `Main` el código de hibernate necesario para que al guardar el objeto `seguro` se guarde todo en la base de datos.

Ejercicio 3

NOTA: Para poder hacer los siguientes ejercicios debes añadir un campo a la tabla de AsistenciaMedica con el importe que cuesta cada una de ellas.

```
Seguro seguro5_1 =
    new Seguro(321, "12345678Z", "Juan", "Chafer", "Bellver", 66, 3, LocalDate.now());
AsistenciaMedica asistenciaMedica1 =
    new AsistenciaMedica(1, "Médico de cabecera", "Mislata", 800, seguro5_1);
AsistenciaMedica asistenciaMedica2 =
    new AsistenciaMedica(2, "Operación de bypass", "Sevilla", 4500, seguro5_1);

Seguro seguro5_2 =
    new Seguro(654, "48573562T", "Amparo", "Martí", "López", 26, 0, LocalDate.now());
AsistenciaMedica asistenciaMedica3 =
    new AsistenciaMedica(3, "Médico de cabecera", "Valencia", 700, seguro5_2);
AsistenciaMedica asistenciaMedica4 =
    new AsistenciaMedica(4, "Operación de miopía", "Valencia", 4500, seguro5_2);
AsistenciaMedica asistenciaMedica5 =
    new AsistenciaMedica(5, "Operación estética", "Madrid", 14500, seguro5_2);
```

En este ejercicio vamos a realizar consultas con el lenguaje de consultas HQL, todos los resultados se deben ver por la consola:

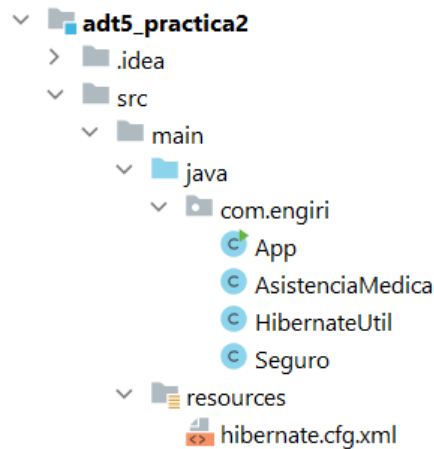
1. Lanza una consulta que nos retorne todos los seguros que hay en la base de datos.
2. Lanza una consulta que nos retorne sólo las columnas NIF y Nombre de todos los seguros que hay en la base de datos.
3. Lanza una consulta que nos retorne sólo el NIF para el seguro con nombre "Juan Chafer Bellver". Usa el método `uniqueResult()` y 3 parámetros con nombre para el nombre y los apellidos.
4. Lanza una consulta que retorna el `idAsistenciaMedica` de todas las asistencias cuyo importe esté entre 2.000 y 5.000 euros (utilizando parámetros por posición para los valores).
5. Lanza una consulta que calcule la suma de todos los importes de todas las asistencias médicas.
6. Lanza una consulta que calcule el saldo medio de todas las asistencias médicas.
7. Lanza una consulta que calcule cuántos seguros hay.
8. Lanza una consulta que muestre para cada seguro cuántas asistencias médicas posee.
9. Lanza una consulta sobre la tabla seguro pero usando una SQL Nativa de MySQL.

Para más información:

- <https://hibernate.org/orm/documentation/6.3/>
- [http://www.cursohibernate.es/doku.php?id=unidades:05 hibernate query language](http://www.cursohibernate.es/doku.php?id=unidades:05+hibernate+query+language)
 - Algunas instrucciones están obsoletas en este último enlace.

AYUDA PARA ORGANIZAR MEJOR LA PRÁCTICA

Estructura de carpetas del proyecto:



Hemos realizado el siguiente menú que engloba a los tres ejercicios:

```

public static void menu() {
    System.out.println("=====");
    System.out.println("===== MENÚ =====");
    System.out.println("1.- Ejercicio1 - Insertar en la entidad Seguro.");
    System.out.println("2.- Ejercicio1 - Actualizar en la entidad Seguro.");
    System.out.println("3.- Ejercicio1 - Borrar en la entidad Seguro.");
    System.out.println("4.- Ejercicio1 - Leer de la entidad Seguro.");
    System.out.println("5.- Ejercicio2 - Creación entidad AsistenciaMedica.");
    System.out.println("6.- Ejercicio3 - Consultas a la base de datos.");
    System.out.println("7.- Salir.");
    System.out.println("=====");
    System.out.print("Opción? ");
}

```

Base de datos con la opción 5 que se corresponde al ejercicio 2:

seguro								
edad	fechaCreacion	idSeguro	numHijos	ape1	ape2	nif	nombre	
66	2023-11-13	321	3	Cháfer	Bellver	12345678Z	Juan	
26	2023-11-13	654	0	Martínez	López	48573562T	Amparo	

asistenciamedica						
idAsistenciaMedica	id_seguro	importe	breveDescripcion		lugar	
1	321	800	Médico de cabecera		Mislata	
2	321	4.500	Operación de bypass		Sevilla	
3	654	700	Médico de cabecera		Valencia	
4	654	4.500	Operación de miopía		Valencia	
5	654	14.500	Operación estética		Madrid	