

Para añadir nuevos datos en la base de datos lo primero que debemos de hacer es realizar las anotaciones en las entidades.

Las anotaciones se podría decir que es definir los parámetros de cada clase en función de la base de datos.

Si es un id, si es una clave ajena, su longitud, etc.

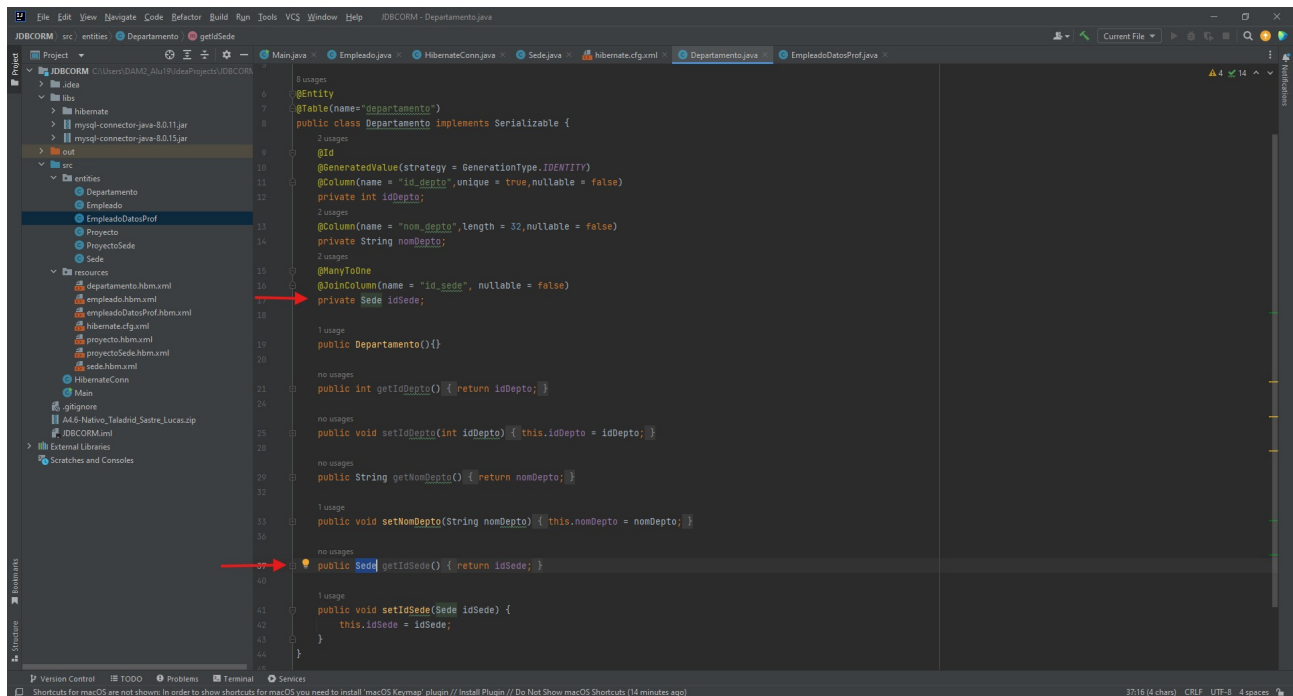


Figura 1: Clase departamento.

En la clase que podemos ver en la Figura 1 tenemos la entidad departamento con sus anotaciones. El id de esta clase por ejemplo es autogenerado, por lo tanto cuando vayamos a añadir nuevos departamentos no será necesario añadir su id.

Por otra parte el nombre del departamento definimos que puede ser null y la longitud máxima que tendrá.

Por último tiene una clave ajena que es el id de una sede. Importante que aunque en la clase sede este id es un integer, aquí es una sede, y al mismo tiempo su getter y setter también tomarán como parámetros una sede.

De la misma forma, cuando vayamos a introducir datos como veremos más adelante en este apartado le debemos de pasar una sede, no su id.

Para más información visitar:

https://docs.jboss.org/hibernate/stable/annotations/reference/en/html_single/

Una vez que tengamos creadas las entidades, añadido sus anotaciones y mapeadas. Debemos de establecer la conexión con la base de datos para poder añadir datos.

```

public class HibernateConn {
    4 usages
    private static SessionFactory sessionFactory;
    6 usages
    private static Session session;

    1 usage
    public static Session openSession() {

        try {
            sessionFactory = new Configuration().configure( resource: "resources/hibernate.cfg.xml").buildSessionFactory();
            session = sessionFactory.openSession();

            return session;
        } catch (ServiceException e) {
            System.err.println("Error de comunicación con la base de datos: " + e.getMessage());
            e.printStackTrace();
        } catch (HibernateException e) {
            System.err.println("Error de Hibernate: " + e.getMessage());
            e.printStackTrace();
        } catch (Exception e) {
            System.err.println("Error desconocido: " + e.getMessage());
            e.printStackTrace();
        }
        return session;
    }

    no usages
    public static void closeSession() {
        if (session != null && session.isOpen()) {
            session.close();
        }
        if (sessionFactory != null) {
            sessionFactory.close();
        }
    }
}

```

Figura 2: Clase para establecer conexión

Como podemos ver en la siguiente Figura estamos estableciendo una conexión para la base de datos.

La línea más importante es la que aparece señalada en la cual creamos un sessionFactory que luego le pasaremos al session para aplicar el método “openSession()”

```

public class Main {

    public static void main(String[] args) {
        Transaction t= null;
        try( Session s= HibernateConn.openSession()){
            t=s.beginTransaction();
            Sede sede= new Sede();
            sede.setNomSede("Málaga");
            s.save(sede);

            Departamento departamento = new Departamento();
            departamento.setNomDepto("INVESTIGACIÓN Y DESRROLLO");
            departamento.setIdSede(sede);
            s.save(departamento);

            Empleado empleado = new Empleado();
            empleado.setDni("56789012B");
            empleado.setNomEmp("SAMPER");
            empleado.setIdDepto(departamento);
            s.save(empleado);
            t.commit();

            System.out.println("Commit exitoso");
        }catch (Exception e){
            e.printStackTrace();
            System.out.println("No se ha podido ejecutar el commit");
            if(t != null){
                t.rollback();
            }
        }
    }
}

```

Figura 3: Clase main

En la Figura de la clase main podemos ver que se abre la conexión y se aplica a una transacción. Luego es necesario crear un objeto del dato que queremos añadir y luego usar los setters para añadir información.

Hay que recordar que si estamos trabajando con claves primarias incrementales o auto generadas no es necesario añadir esa información.

Por último después de cada dato se debe de ejecutar un “save” y al final de la transacción ejecutar un commit.