**MINI PROJECT**:

**Employee Management System**

**FUNCTIONALITIES:**

**1. Add Employee**

**2. Search Employee**

**3. View All Employee**

**4. Update Employee**

**5. Delete Employee**

**6. Quit**

**PROJECT STRUCTURE:**

**Every project** which **we design** following **industry standards** is **divided** into **different layers / tiers**.

**Each layer** plays a **different role** and contains **classes** for that **purpose** only.

In the **software industry**, every **database oriented application** contains **3 layers**:

1.The **Front End** Layer

2.The **Back End** Layer

3.The **Middle** Layer

**1. empmgmt.gui :**

All the **front end code** is written here.

It contains all the **Frames** and **UI elements** our **application** will have.

The **execution** will **begin** from **here**.

**All the calls** to **other methods** will be **made** from **here**.

**2. empmgmt.dao(Data Access Object) :** It **contains** all the **classes** that **interact** with the **database**.

It **contains** all the **methods** like

**addEmploye( )**, **searchEmployee( ), updateEmployee( )** etc are **placed** here.

**3. empmgmt.dbutil:**

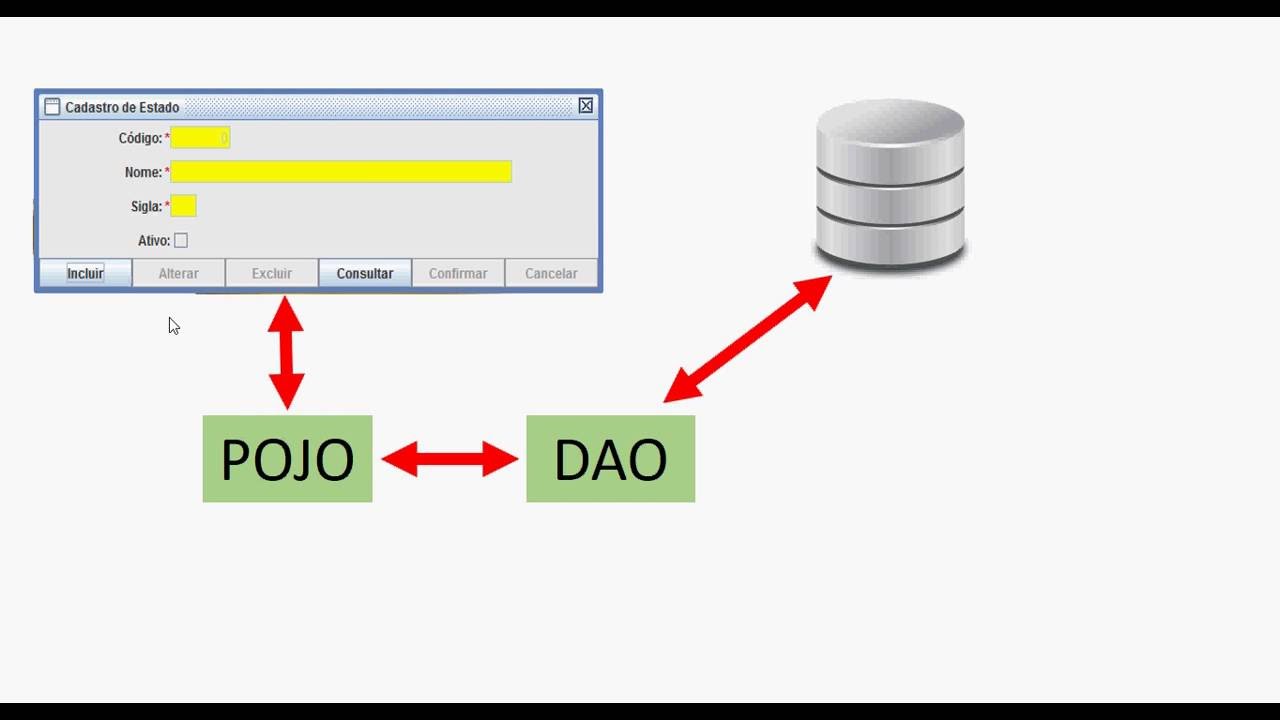
It **contains** the **database connection class**.

It will **contain method** to **load the driver class**, **open the connection to the db** and **close the connection** to the **database**.

**4. empmgmt.pojo(Plain Old Java Objects) :**

This **package** **contains** all the **classes** which **hold** the **data** of our **application** i.e. they **hold** the **data** coming from the **table** or **to be stored** into the **table**.

**POJO classes** are **created** **one per table** and are **also called** as **DTO** (Data Transfer Object) **classes**.



**A More Detailed Diagram**

