

METODOLOGIA DAO

1.- IEmpleadoService

```
public interface IEmpleadoService {  
    public ArrayList<String> getEmpleadoNamesByJob(String job);  
}
```

2.- EmpleadoService

```
public class EmpleadoService implements IEmpleadoService{  
    4 usages  
    private IEmpleadoDao empleadoDao;  
  
    1 usage  
    public EmpleadoService(IEmpleadoDao empleadoDao) {  
        this.empleadoDao = empleadoDao;  
    }  
}
```

```
public ArrayList<String> getEmpleadoNamesByJob(String job) {  
    return this.empleadoDao.getEmpleadoNamesByJob(job);  
}
```

3.- IEmpleadoDao

```
public interface IEmpleadoDao extends IGenericDao<Empleado>{  
    1 usage 1 implementation new *  
    ArrayList<String> getEmpleadoNamesByJob(String job);  
}
```

4.- EmpleadoSQLServerDao

```
public class EmpleadoSQLServerDao extends AbstractGenericDao<Empleado> implements IEmpleadoDao {  
  
    8 usages  
    private DataSource dataSource;  
  
    mfernandez*  
    public EmpleadoSQLServerDao(DataSource dataSource) {  
        this.dataSource = dataSource;  
    }  
  
    1 usage new*  
    public EmpleadoSQLServerDao() {  
        this.dataSource = DBCPDataSourceFactory.getDataSource();  
    }  
}
```

```
public ArrayList<String> getEmpleadoNamesByJob(String job) {  
    String nombreEmpleado;  
    ArrayList<String> nombres = new ArrayList<>();  
  
    try {  
        Connection conexion = this.dataSource.getConnection();  
        PreparedStatement pstmt = conexion.prepareStatement("SELECT ENAME " +  
            "FROM EMP WHERE JOB= ? ORDER BY ENAME");  
  
        pstmt.setString(1, job);  
        ResultSet result = pstmt.executeQuery();  
  
        while (result.next()) {  
            nombreEmpleado = result.getString(1);  
            nombres.add(nombreEmpleado);  
        }  
    } catch (SQLException ex) {  
        ex.printStackTrace();  
        System.err.println("Ha ocurrido una excepción: " + ex.getMessage());  
    }  
  
    return nombres;  
}
```

5.- EmpleadoController

```
public class EmpleadoController {  
    2 usages  
    private IEmpleadoDao empleadoDao;  
    3 usages  
    private IEmpleadoService empleadoService;  
  
    2 usages  
    public EmpleadoController() {  
        this.empleadoDao=new EmpleadoSQLServerDao();  
        this.empleadoService=new EmpleadoService(empleadoDao);  
    }  
  
    public ArrayList<String> getEmpleadoNamesByJob(String job){  
        return this.empleadoService.getEmpleadoNamesByJob(job);  
    }  
}
```

6.- Main

```
public static void main(String[] args) {  
  
    getEmpleadoNamesByJob("SALESMAN");  
}  
  
1 usage new*  
private static void getEmpleadoNamesByJob(String job) {  
    EmpleadoController controlador=new EmpleadoController();  
    ArrayList<String> nombres=controlador.getEmpleadoNamesByJob(job);  
    for (String nom:nombres) {  
        System.out.println(nom);  
    }  
}
```