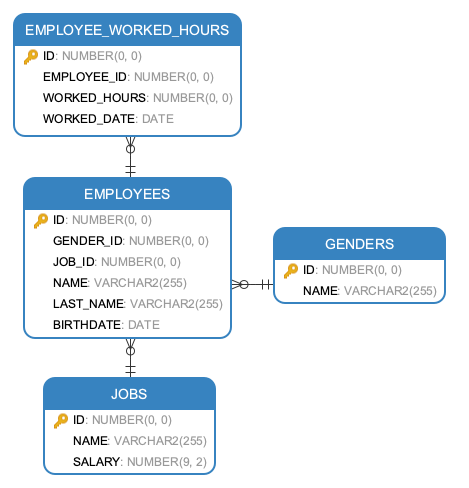
Examen Práctico

**Instrucciones**: Observa el siguiente diagrama de BD y realiza los ejercicios que se piden, los contratos de interfaz se encuentran en el apartado **Contratos De Interfaz**.

**Requisitos**:

* Enviar el link de cualquier gestor de repositorios (Ejemplo: GitHub, Gitlab).
* El proyecto deberá contener un archivo con los scripts de BD para la creación de las tablas.

**Nota:** El acceso a base de datos puede ser utilizando un ORM(Hibernate).



### Ejercicio 1. Realiza un web service que permita agregar un nuevo empleado.

### Se debe validar que el nombre y apellido del empleado no existan, que el empleado sea mayor de edad y que el género y puesto existan en sus tablas correspondientes.

### Ejercicio 2. Realiza un web service que permita agregar horas trabajadas de un empleado (ver anexo 1.2).

### Se debe validar que el empleado exista, que el total de horas trabajadas no sea mayor a 20 horas y que la fecha de trabajo sea menor o igual a la actual y no se duplique por empleado (un empleado sólo puede tener un registro de horas trabajadas por día).

### Ejercicio 3. Realiza un web service que permita consultar los empleados por puesto.

### Se debe validar que el puesto exista.

### Ejercicio 4. Realiza un web service que permita consultar el total de horas trabajadas de un empleado por rango de fechas.

### Se debe validar que el empleado exista y que la fecha de inicio sea menor a la fecha de fin.

### Ejercicio 5. Realiza un web service que permita consultar cuanto se le pagó a un empleado en un rango de fechas.

### Se debe validar que el empleado exista y que la fecha de inicio sea menor a la fecha de fin.

### Realiza al menos 2 validaciones de respuesta en cualquiera de los ejercicios anteriores con ResponseEntity en el cual se retorna el Estatus 404 o 500 con mensajes de error.

Contratos De Interfaz

### Contrato 1.1

|  |  |
| --- | --- |
| Request:  {  "gender\_id": 1, // Id del catálogo **genders**  "job\_id": 1, // Id del catálogo **jobs**  "name": "Juan", // Nombre del empleado  "last\_name": "Pérez", // Apellido del empleado  "birthdate": "1983-01-01" // Fecha de nacimiento del empleado  } | Response:  {  "id": 10, // Id insertado o null en caso de error.  "success": true // true si se insertó el registro o false en caso de error  } |

### Contrato 1.2

|  |  |
| --- | --- |
| Request:  {  "employee\_id": 1, // Id del empleado  "worked\_hours": 10, // Horas trabajadas  "worked\_date": "2019-01-01" // Fecha que trabajó el empleado  } | Response:  {  "id": 100, // Id insertado o null en caso de error.  "success": true // true si se insertó el registro o false en caso de error  } |

### Contrato 1.3

|  |  |
| --- | --- |
| Request:  {  "job\_id": 1, // Id del puesto  } | Response:  {  “employees”: [ //Array de empleados  {  “id”: 1,  “name”: “Juan”,  “last\_name”: “Pérez”,  “birthdate”: “01-02-1978”,  “job”: {  “id”: 3,  “name”: “Gerente”,  “salary”: 100  },  “gender”: {  “id”: 1,  “name”: “Hombre”  }  },  {  “id”: 2,  “name”: “José”,  “last\_name”: “López”,  “birthdate”: “03-12-1980”,  “job”: {  “id”: 3,  “name”: “Gerente”,  “salary”: 100  },  “gender”: {  “id”: 1,  “name”: “Hombre”  }  }  ],  "success": true // true si se logró obtener los datos, false en caso de error  } |

### Contrato 1.4

|  |  |
| --- | --- |
| Request:  {  "employee\_id": 1, // Id del empleado  "start\_date": “2019-01-01”, // Fecha de inicio  "end\_date": "2019-06-30", // Fecha de fin  } | Response:  {  "total\_worked\_hours": 100, // Total de horas o null en caso de error  "success": true // true si se logró obtener los datos, false en caso de error  } |

### Contrato 1.5

|  |  |
| --- | --- |
| Request:  {  "employee\_id": 1, // Id del empleado  "start\_date": “2019-01-01”, // Fecha de inicio  "end\_date": "2019-06-30", // Fecha de fin  } | Response:  {  "payment": 100, // Cantidad pagada al empleado o null en caso de error  "success": true // true si se logró obtener los datos, false en caso de error  } |