**PRUEBA TÉCNICA**

**INTRODUCCIÓN**

Esta prueba técnica está diseñada para evaluar las habilidades en el desarrollo de servicios web utilizando el framework Quarkus. Los ejercicios involucran la creación y gestión de datos relacionados con empleados y sus horas trabajadas, así como la implementación de operaciones CRUD básicas y consultas específicas en una base de datos Oracle. A continuación se describen los ejercicios y sus soluciones.

**ENTORNO DE DESARROLLO**

* **Framework:** Quarkus 3.12.1
* **Base de Datos:** Oracle
* **IDE:** NetBeans
* **Librerías:** Hibernate ORM, Panache, RESTEasy, Agroal, Narayana JTA

**EJERCICIO 1**

Realiza un web service que permitirá agregar un nuevo empleado. Para que se pueda agregar debe cumplir con lo siguiente:

* Respetar el contrato de interfaz.
* El nombre y apellido del empleado no están registrados en BD.
* Ser mayor de edad.
* El puesto asignado debe existir en la tabla de Jobs.

Contrato 1.1

*{*

*"gender\_id": 1, // Id del catálogo genders*

*"job\_id": 1, // Id del catálogo jobs*

*"name": "Juan", // Nombre del empleado*

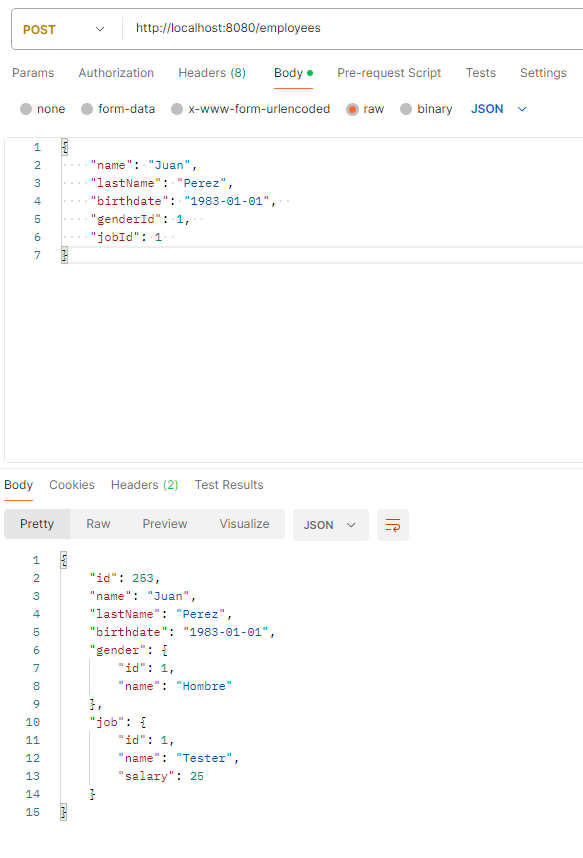
*"last\_name": "Pérez", // Apellido del empleado*

*"birthdate": "1983-01-01" // Fecha de nacimiento del empleado*

*}*

Tabla EMPLOYEES antes de el POST:  


Ejecución en PostMan



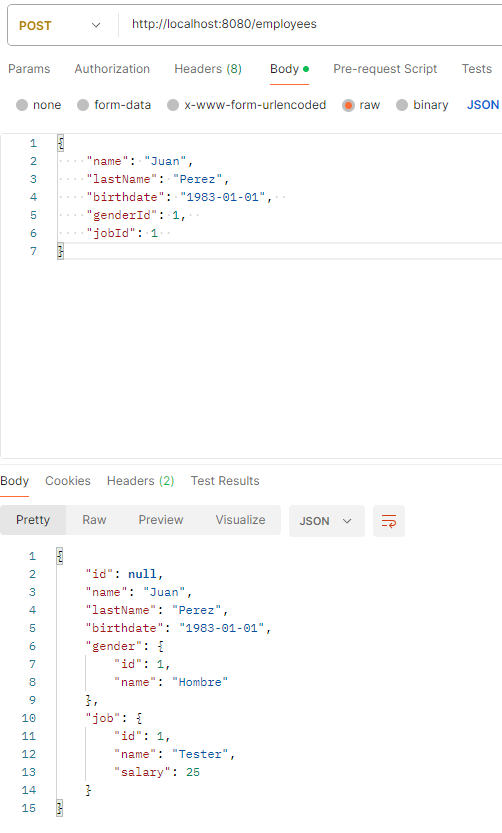
Se valida en Oracle

***select \* from employee;***



***Caso de prueba 1:*** No se puede agregar con el mismo nombre y apellido

Ejecución en PostMan



Se vuelve a correr y nos retorna null en el ID ya que no es válido.

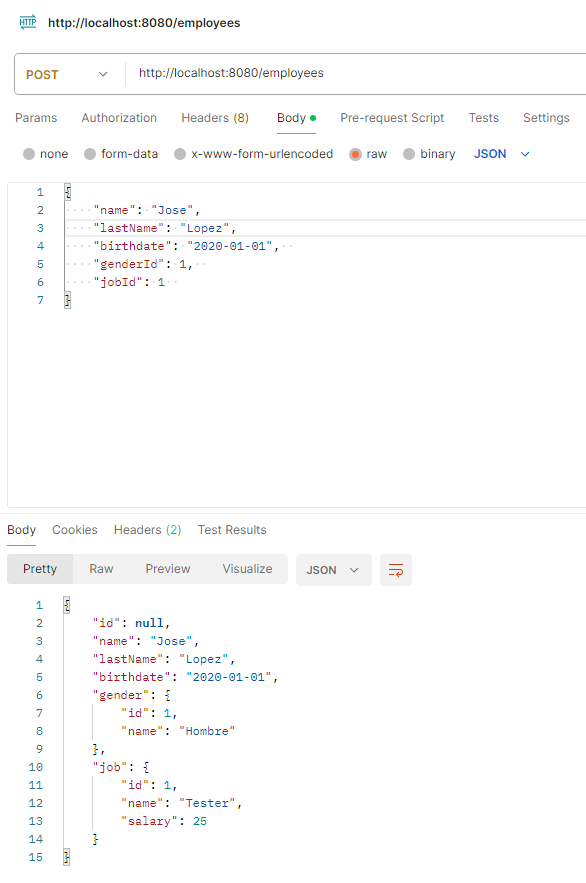
Cuando se valida en Oracle solo tenemos el primero.

Validación en Oracle.



***Caso de prueba 2***: Debe ser mayor de edad:

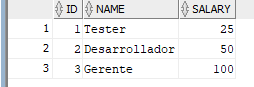
Ejecución en PostMan.



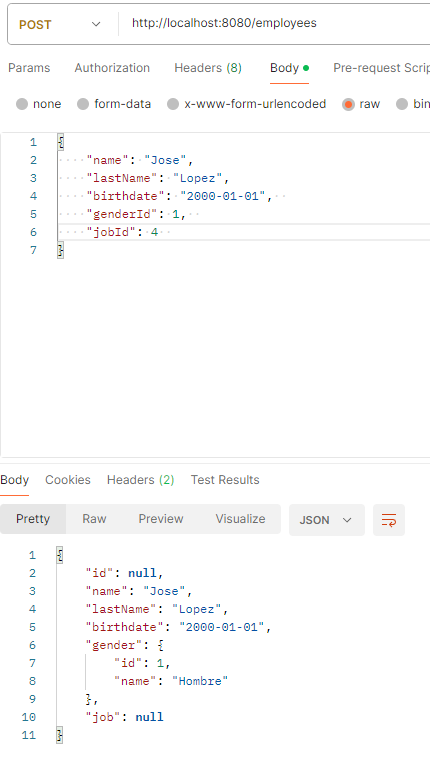
Cambie de nombre y la fecha de nacimiento, y como se puede observar sigue regresando null, señal de que no se agregó a la tabla ya que no cumple con las condiciones.

***Caso de prueba 3***: El puesto debe existir en la tabla Jobs

Validación de datos en Oracle.



Ejecución en PostMan.



Se vuelve a correr con el mismo nombre solo cambiamos a que la fecha de nacimiento sea válida y a su vez un id de job que no existe. Por lo cual no solo regresa id null sino también job.

**Ejercicio 2:**

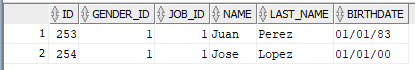
Realizar un web service que permita:

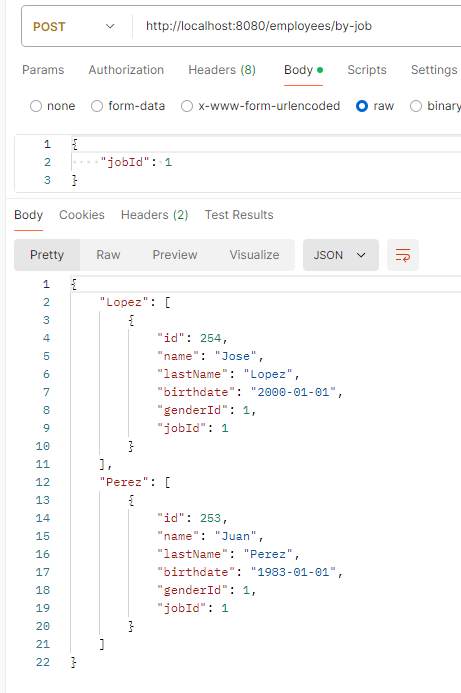
1. Recibir un puesto.
2. Consultar a los empleados por puesto.
3. Filtrar los empleados obtenidos en b) con el puesto recibido en a) ocupando expresiones lambda con filtro.
4. De los empleados obtenidos en c) ordenarlos por apellido materno.
5. De los empleados obtenidos en d) agruparlos con expresión lambda por apellido materno.

Validación de datos de la table Worked (horas trabajadas) en Oracle.



Validación de datos de la tabla Employee (empleados) en Oracle.

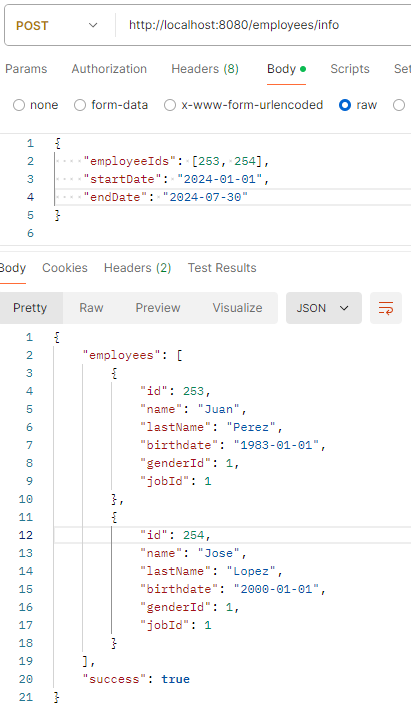


Ejecución en PostMan validando todos los requisitos.  


Se corrió el JSON con el id de Job, para que este mismo retornara los datos y los acomoda en orden alfabético.

***Ejercicio 3.***

Realiza un web service que permite consultar la información recibida de una lista de empleados con proceso multi-hilos.



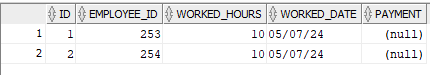
Se corrió el JSON para recibir validar el servicio web y que devuelve correctamente la información solicitada para cada empleado en la lista.

***Ejercicio 4.***

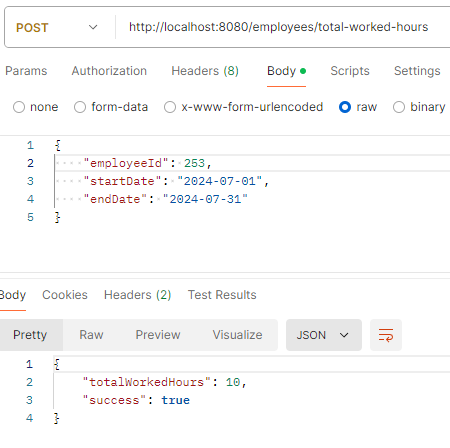
Realiza un web service que permite consultar el total de horas trabajadas de un empleado por rango de fechas.

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

Se consulta la tabla Worked (horas trabajadas) para ver el ID y las horas registradas en la tabla:



Ejecución en PostMan:



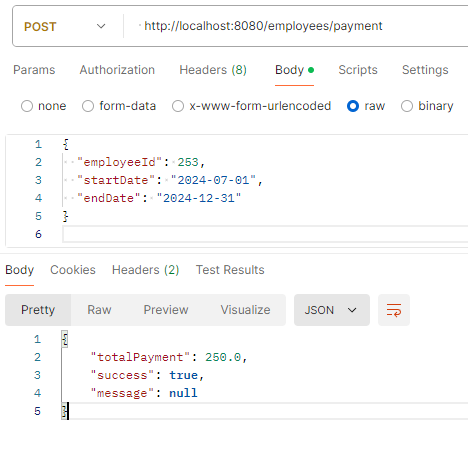
Se valida ya que entra en un rango de fechas permitido, al igual que las horas trabajadas concuerdan con las que están registradas en la tabla.

***Ejercicio 5.***

Realiza un web service que permite consultar cuánto se le pagó a un empleado en un rango de fechas.

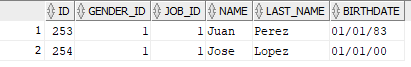
Se debe validar que el empleado exista y que la fecha de inicio sea menor a la fecha de fin.

Ejecución de PostMan:



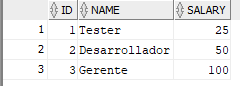
Se valida ya que los datos de prueba existen y son correctos. Por lo cual retorna un pago total de acuerdo con sus horas trabajadas y el sueldo propuesto.

Tabla de empleados:



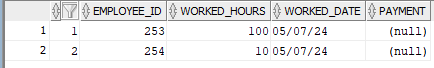
Se muestra que Juan Perez de ID 253 tiene JOB\_ID 1

Tabla de trabajos:



Se muestra que el ID 1 es correspondiente a tester y su salario de 25

Tabla de horas trabajadas:



Se muestra que el ID 253 (Juan Perez) trabajó 100 horas.

Por lo cual Juan Perez con ID 253 trabajo 100 horas de Tester de acuerdo a su JOB\_ID por lo cual su salario sería de 250 (25\*100), validando el retorno de PostMan.