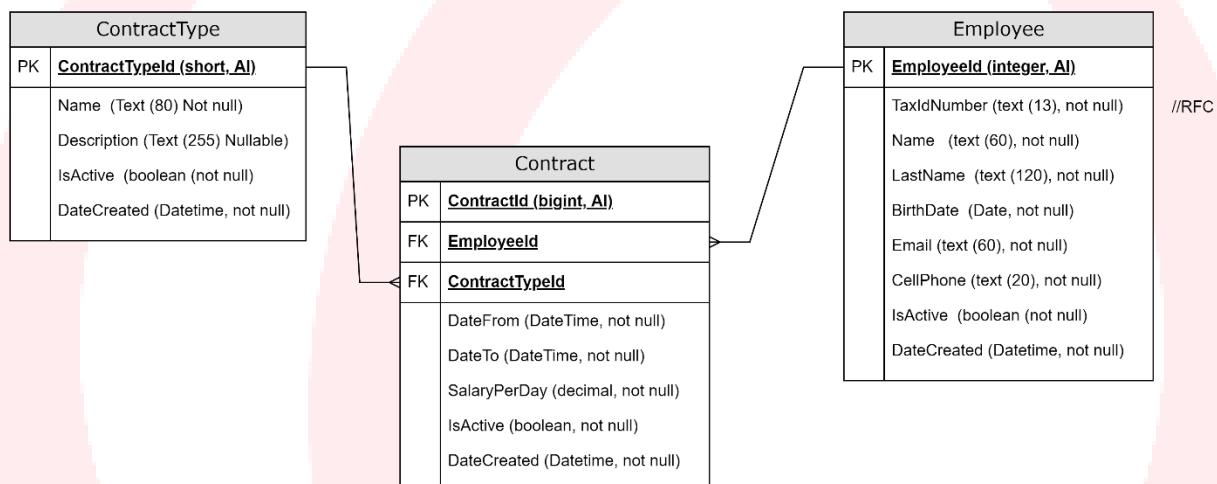


## EJERCICIO TÉCNICO DESARROLLADOR WEB JAVA L2

Metaphorce está en un proceso para el desarrollo de una solución de software en inglés de Recursos Humanos y está solicitando un primer entregable que tenga como alcance el desarrollo de **5 servicios web** para la gestión de sus empleados y la generación de sus contratos, considerando información básica por el momento, de tal forma que posteriormente una aplicación web y/o aplicación móvil pueda gestionar dicha información.

Considere el siguiente schema de una base de datos llamada MetaphorceBd para realizar la implementación del mismo y la API correspondiente que realice las operaciones indicadas en el punto 4 del requerimiento.

### Schema de base de datos:



- El catálogo ContractType deberá tener los siguientes 3 registros activos precargados: "Permanent", "Fixed-Term", "External" con fecha de creación del día en el que se realiza la prueba.

### REQUERIMIENTO

- Utilizar java 1.8 de preferencia.
- Crear un proyecto en **Eclipse / Netbeans / IntelliJ** o similar, utilizando **Java con Spring Framework / Spring Boot** para generar la **API** correspondiente al requerimiento expuesto.
- Utilizar **Hibernate** para crear la conexión entre base de datos, mapeo del schema y las operaciones hacia cualquiera de los siguientes motores: MySQL u Oracle o PostgreSQL (a su elección, elegir solo uno).
- Utilizar **Repositorios** para realizar las consultas a la base de datos como al igual las inserciones y/ actualizaciones
- Generar **únicamente** los siguientes 5 endpoints:
  - Obtener el listado de empleados activos extrayendo **solo los siguientes campos**:
    - Name y LastName** (campo concatenado).

- *TaxIdNumber* (RFC).
  - *Email*.
  - *Nombre del tipo de contrato* (ContractType.Name) en el que se encuentre actualmente activo. Si no cuenta con contrato activo enviar valor NULL.
  - *Fecha inicio del contrato* en el que se encuentre actualmente activo (Contract.DateFrom). Si no cuenta con contrato activo enviar valor NULL.
  - *Fecha final del contrato* en el que se encuentre actualmente vigente (Contract.DateTo), Si no cuenta con contrato activo enviar valor NULL.
  - *Salario diario* (Contract.SalaryPerDay), si no cuenta con contrato activo enviar valor NULL.
- 2) Agregar un contrato nuevo.
    - Si el empleado no tiene un contrato lo agregará con los datos enviados.
    - Si el empleado tiene un contrato actual vigente en la base de datos, deberá dar de baja el contrato actual actualizando la fecha DateTo a la de la operación y posteriormente ingresar el nuevo contrato con los datos enviados. **Un empleado no debe tener 2 contratos activos.**
  - 3) Agregar un nuevo empleado.
    - Deberá validar que el formato del *TaxIdNumber* (RFC) sea correcto a través de una regex. En caso de ser incorrecto no guardará el registro enviando un error.
    - Debe validar que el *TaxIdNumber* (RFC) no esté registrado, en caso de existir deberá mandar un error indicando que ya existe el empleado.
  - 4) Actualizar los datos de un empleado existente a través de su folio.
    - Si el *TaxIdNumber* (RFC) se modifica, debe validar previamente que el nuevo tenga el formato correcto y que de igual manera no exista.
  - 5) Eliminar un contrato vigente de un empleado seleccionado.
    - Deberá desactivar un contrato actualizando la fecha DateTo a la actual.
    - Si el empleado no cuenta con un contrato vigente o el proceso no encuentra ninguno deberá mandar un error.
  6. Implementar **autenticación básica** (usuario: metaphorce, password: m3t4Ph0rc3) para consumir los endpoints.
  7. Documentar a través de OpenAPI (swagger o documento de Word manual) como es que se deben acceder a los endpoints por parte de un cliente (aplicación web, móvil, etc) y las respuestas que arrojará.
    - 1) Especificar el controller/method-name, verbo a utilizar, header y body.
    - 2) Especificar todas las respuestas posibles (Ok, Created, Bad Request, Server Error, etc) de cada endpoint con su estructura Json y su código de error.

## Generales:

1. El proyecto deberá estar codificado en idioma inglés, así como el modelo de base de datos.
2. Deberá utilizar buenas convenciones del lenguaje Java.
  - a. Respetar notaciones camel case, pascal case, etc. para clases y métodos.
  - b. Cuidar el uso de excepciones.
  - c. Modular el proyecto de tal forma que el código sea legible.
  - d. Reutilizar código y funciones dará valor a la prueba.
3. Proveer todas las instrucciones para la ejecución correcta de la solución, incluyendo recursos de tablas y población de datos (scripts, instalación de paquetes, etc.).
4. Deberá utilizarse una arquitectura N-Layers, a 3 capas como mínimo (Controladores, Negocio y Persistencia de datos) agregar capas de manera justificada dará valor a la prueba.
5. Considerar que todas las entidades comparten los campos IsActive y DateCreated.

Domicilio: Plaza de la paz 102, Suite 705 en piso 7, Col. Guanajuato Puerto Interior C.P. 36275 Silao Gto.  
Teléfono: +52 (477) 274 8866 Ext. 705  
Email: contacto@forteinnovation.mx

6. Generar Pruebas Unitarias dará valor a la prueba.

### Tiempo:

Duración: 28 horas naturales a partir de su entrega. Si se termina el tiempo entrega lo que se haya finalizado hasta ese momento, **no te sientas frustrado si no lograste terminar.**

### Repositorio:

1. Debe usar un repositorio personal público como GitHub, Bitbucket, Gitlab, Azure DevOps o similar, subir el proyecto y enviarnos la URL del repositorio junto con la documentación antes de la entrevista de revisión.
2. Al momento de enviar la URL describir los módulos de Spring que se utilizaron en el proyecto, como se utilizó Hibernate / JPA y una breve descripción de cómo se estructuró el proyecto.

## EJERCICIO BASE DE DATOS T-SQL

1. Dada la siguiente base de datos de un comercio de venta de productos por internet realizar las consultas solicitadas a través de Transact-SQL a una tabla llamada Clientes.

Nombre Tabla:		Clientes				
Id	Nombre	Apellidos	Sexo	Edad	Estado	Ciudad
1	Oscar	Huerta Sánchez	M	25	Guanajuato	León
2	Octavio	Sánchez Ortega	M	30	Guanajuato	Irapuato
3	Julio	Morales Camarena	M	23	Jalisco	Guadalajara
4	María	Díaz Arellano	F	25	Michoacán	La Piedad
5	Lorena	López Ortiz	F	30	Nuevo León	Monterrey
6	Fernando	Huerta Mireles	M	28	Jalisco	Lagos de Moreno
7	Cassandra	Contreras Pérez	F	22	Guanajuato	León

Ejemplo:

Sentencia que arroje todos los registros con todas sus columnas

**R = SELECT \* FROM Clientes;**

- 1) Sentencia que arroje aquellos registros cuyo sexo sea masculino y la edad sea superior a 25

**R=**

- 2) Escribir una sentencia que actualice el registro con Id 6 indicando que la Ciudad es "San Juan de los Lagos" y Apellidos "Herrera Huerta"

**R=**

- 3) Escribir una sentencia que arroje aquellas personas que están entre 20 y 29 años y sean sexo femenino.

**R=**

4) Escribir una sentencia que arroje los registros cuyos Apellidos contienen el apellido "Huerta"

R=

5) Sentencia que arroje el conteo de los registros que están en el estado de Guanajuato.

R=

6) Escribir una sentencia que inserte un nuevo registro a dicha tabla considerando que todos los campos son obligatorios.

R=

7) Escribir una sentencia que seleccione la cantidad de personas que hay por estado dando como resultado la siguiente tabla.

<b>Resultado</b>	Guanajuato	3
	Jalisco	2
	Michoacán	1
	Nuevo León	1

R=

8) Escribir una sentencia que consulte las personas que son del sexo Masculino y las ordene de manera descendente por su apellido y posteriormente por su Nombre.

R=

¡Mucho éxito! Nos vemos en la siguiente. 😊