Universidad Nacional Autónoma de Honduras
Facultad de Ingeniería
Departamento de Ingeniería en Sistemas

Ejercicios Unidad II

Lenguajes de Programación

1. Desarrolle un api rest que permita crear operaciones de mantenimiento sobre la siguiente tabla:

Usuarios
codigoUsuario - varchar(20)
nombre - varchar(50)
apellido - varchar(50)
departamento - varchar(100)
correo - varchar(90)
telefono - varchar(20)

   a. Método para crear usuarios
   b. Método para eliminar usuarios
   c. Método para buscar usuarios por código
   d. Método para obtener todos los usuarios

2. Modifique el esquema anterior para asociarle al usuario un perfil, de manera que el esquema quede de la siguiente manera. Considere que un usuario solo puede tener un perfil.

Usuarios
codigoUsuario - varchar(20)
nombre - varchar(50)
apellido - varchar(50)
departamento - varchar(100)
correo - varchar(90)
telefono - varchar(20)
idPerfil - int

Perfil
idPerfil - int, identity
rol - varchar(20)
descripcion - varchar(20)

Cree los siguientes métodos, considerando la inclusión del perfil:

a. Método para crear usuarios
b. Método para eliminar usuarios
c. Método para buscar usuarios por código
d. Método para obtener todos los usuarios

3. Desarrolle una base de datos que permita manejar los pagos históricos que se hacen a los empleados, para lo cual deberá diseñar las siguientes tablas:

Empleado
dni - varchar(20)
nombre - varchar(50)
apellido - varchar(50)
fechaIngreso - date

HistoricoPagos
FechaPago - date
dni - varchar(20)
HorasTrabajadas - int
PrecioPorHora - decimal (10,2)
TotalSueldo - decimal(10,2)

Con base en lo anterior usted deberá crear: Modelos, Repositorios, servicios y controladores que permitan realizar estas tareas:

a. Método para crear empleados
Parámetros: Json con las propiedades del empleado
b. Método para eliminar empleados
Parámetros: id de empleado a eliminar
c. Método para buscar Empleado (deberá devolver el historial de pagos)
Parámetros: id de empleado a buscar
d. Método para obtener todos los empleados (deberá incluir el historial de pagos de todos los empleados)
Parámetros: Ninguno
e. Método para obtener los pagos de un empleado en una fecha específica
Parámetros: Json Request

```
{"idEmpleado: "idEmpleado",
 "fechaInicio": "fechaInicio",
 "fechaFin": "fechaFin"}
```

f. Método para crear pago de empleado:
Parámetros: Json request

{
"idEmpleado" : "idEmpleado",
"horasTrabajadas" : "999",
"precioPorHora" : "999.99"
}

Usted deberá validar que el empleado exista, de igual forma que tanto horas trabajadas como precio por hora sean mayor que 0.
El método deberá calcular el total (horasTrabajadas * precioPorHora) y la fecha deberá tomarla del sistema.

4. Desarrolle una base de datos que permita manejar la facturación de la energía eléctrica de los abonados, la cual sería como a continuación:

Abonado
dni - varchar(20)
nombre - varchar(20)
apellido - varchar(20)
telefono - varchar(20)
idTipoResidencia - int

TiposResidencia
idTipoResidencia - int
descripcion - varchar(20)
precioKw - decimal(10,2)

Facturacion
fechaFacturacion - date
dni - varchar(20)
kwsConsumidos - decimal(10,2)
TotalFacturacion - decimal(10,2)

Con base en lo anterior deberá crear los siguientes métodos:

a. Método para crear tipos de residencias
Parámetros: Json request {"descripción":"descripción", "precioKw" : 5.467}
Deberá insertar 3 tipos: residencial, 4.56. bodega, 8.98, edificio, 5.48.

b. Método crear abonados:
Parámetros: Json request {"abonadoId": "0801199018499", "nombre":"Karla", "apellido":"Mendoza", "teléfono": "999-999-99", "idTipoResidencia":1}

c. Método para obtener Abonado.
Parámetros: dni

d. Método para eliminar Abonado
Parámetros: dni

e. Método para facturar
Parámetros: Json Request
{
"dni": "0801199804965",

"kwConsumido" : 32.65

}

Con lo anterior usted deberá crear un registro en la tabla facturación, la fechaFacturacion la tomará del sistema y el total lo calculará multiplicando los kwConsumidos * precioKw el cual deberá obtener a partir del idTipoResidencia del abonado por el dni.