Fundamentos de Bases de Datos.
Práctica 9.

Profesor: M.I. Gerardo Avilés Rosas
gar@ciencias.unam.mx
Laboratorio: Luis Eduardo Castro Omaña
lalo_ castro@ciencias.unam.mx

18 de mayo de 2020

Se dan a conocer especificaciones de entrega para la práctica 9.

1. JDBC

Existen sistemas que necesitan tener comunicación con alguna base de datos, ya sea para poder obtener información de la base para poder manipular datos, insertar nuevos datos, eliminar información de la base o modificar información contenida en la base de datos.

Actualmente existen una infinidad de software que hace uso de bases de datos. Teniendo en cuenta la cantidad de lenguajes de programación que existen para construir esta cantidad de software, surge la necesidad de tener un conjunto de bibliotecas que nos permita crear un vinculo entre alguna base de datos con algún software que quiera hacer uso de la base, sin importar el lenguaje de programación con el que dicho software se codificó.

Java Data Base Connectivity (JDBC) API es el estandar de industria para la conectividad independiente de base de datos entre el lenguaje de programación Java y una amplia gama de bases de datos SQL y otras fuentes de datos tabulares, como hojas de cálculos o archivos planos. El JDBC proporciona un nivel de llamada API para el acceso de base de datos SQL.

PostgreSQL provee un driver que permite a JDBC crear una conexión con cualquier software codificado con esta tecnología. Lo cual nos permite explotar los datos de manera indirecta de la información concentrada en una base de datos.

2. Servicio Web

Un servicio web es definido como un sistema de software designado para dar soporte a la interacción de máquina a máquina interoperativa a través de una red.

Un servicio web realiza una tarea específica o un conjunto de tareas, y se describe mediante una descripción de servicio en una notación XML estándar llamada WSDL (Web Services Description Language). La descripción de servicio proporciona todos los detalles necesarios para interactuar con el servicio, incluidos los formatos de mensaje (que detallan las operaciones), los protocolos de transporte y la ubicación.

Otros sistemas utilizan mensajes SOAP para interactuar con el servicio web, normalmente utilizando HTTP con una serialización XML conjuntamente con otros estándares relacionados con la web.

La interfaz WSDL oculta los detalles de cómo se implementa el servicio, y el servicio se puede utilizar independientemente de la plataforma de hardware o software en la que se implementa e independientemente del lenguaje de programación en el que está escrito.

Las aplicaciones basadas en servicios web son implementaciones en todas las tecnologías, con acoplamientos flexibles y orientadas a componentes. Los servicios web se pueden utilizar individualmente o junto con otros servicios web, para llevar a cabo una agregación completa o una transacción empresarial.

2.1. Servicio REST

El término REST (Representational State Transfer) se refiere a un estilo arquitectónico propuesto en el año 2000 por Roy Fielding. REST no es un estándar. No existe una especificación formal de REST, y tampoco encontrarás un kit de desarrollo para REST. Simplemente es un estilo arquitectónico, y por lo tanto no se puede embotellar. Simplemente hay que entenderlo para poder diseñar aplicaciones que cumplan el estilo REST.

Los fundamentos de REST son los siguientes:

- Los recursos son expuesto de manera sencilla en forma de URL.
- Las representaciones transfieren JSON O XML para representar objetos de datos y atributos.
- Los mensajes utilizan explícitamente los métodos HTTP.
- No mantienen estado, esto hace que los servicios sean independientes del cliente.

2.2. CRUD

CRUD hace referencia a un acrónimo en el que se reúnen las primeras letras de las cuatro operaciones fundamentales de aplicaciones persistentes en sistemas de bases de datos:

- Create (Crear registros)
- Read bzw. Retrieve (Leer registros)
- Update (Actualizar registros)
- Delete bzw. Destroy (Borrar registros)

En pocas palabras, CRUD resume las funciones requeridas por un usuario para crear y gestionar datos. Varios procesos de gestión de datos están basados en CRUD, en los que dichas operaciones están específicamente adaptadas a los requisitos del sistema y de usuario, ya sea para la gestión de bases de datos o para el uso de aplicaciones.

Operación CRUD	**SQL**	**RESTful**
Create	INSERT	POST,PUT
Read	SELECT	GET,HEAD
Update	UPDATE	PUT, PATCH
Delete	DELETE	DELETE

Cuadro 1: Implementaciones utilizadas para cada operación.

2.3. Actividad

Para esta práctica es necesario:

1. Configurar ambiente para poder consumir la base de datos, para esto es necesario instalar:
   - Java 11: Amazon Corretto es una distribución sin costo, multiplataforma y lista para producción de Open Java Development Kit (OpenJDK). `https://docs.aws.amazon.com/es_es/corretto/latest/corretto-11-ug/what-is-corretto-11.html`
   - Maven: Maven es una herramienta de software para la gestión y construcción de proyectos Java `https://maven.apache.org/index.html`
   - IntelliJ IDEA: Entorno de desarrollo integrado escrito en Java para desarrollar software. `https://www.jetbrains.com/idea/` `https://www.jetbrains.com/community/education/#students`
   - Postman: Plataforma de colaboración para desarrollo de APIs. `https://www.postman.com/`

2. Ingresar a la carpeta del proyecto y compilar el proyecto maven para obtener las dependencias necesarias. Aquí se descargará el JDBC de postgreSQL para poder realizar la conexión a la base de datos.

```
> {path}\practica9\northwind
> mvn clean install
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building northwind 1.0-SNAPSHOT
[INFO] ------------------------------------------------------------------------
.
.
.
.
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 22.743 s
[INFO] Finished at: 2019-11-11T12:15:46-06:00
[INFO] Final Memory: 35M/981M
[INFO] ------------------------------------------------------------------------
```

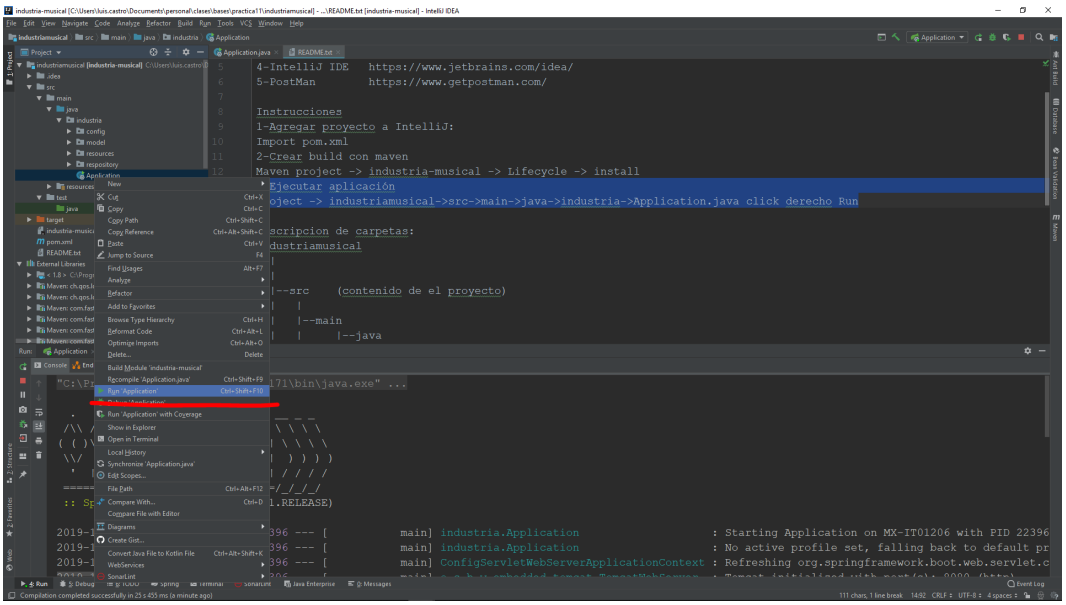
3. Abrir IDE (se recomienda IntelliJ) y cambiar los valores de de conexión a la base de datos en el archivo application.properties.

```
{path}\northwind\src\main\resources\application.properties

application.properties.url -> URL, puerto y base de datos a la que
 se va a conectar el proyecto.
application.properties.databaseName -> nombre de la base de datos.
application.properties.username -> usuario de la base de datos.
application.properties.password -> password de la base de datos.
```

   Se deben conectar a la siguiente base de datos:
   jdbc:postgresql://fbd2020-2.chcvtg6ofmsr.us-east-2.rds.amazonaws.com:5432/northwind
   con el usuario: *fundamentosbd20202*
   y contraseña: *fundamentosbases*

4. Ejecutar aplicación Project ->northwind->src->main->java->northwind->Application.java click derecho Run



Deberán obtener los siguiente para validar que está corriendo correctamente:

```
2020-05-17 21:25:53.634  INFO 7615 --- [main] northwind. Application:
 Started Application in 1.764 secondsm (JVM running for 2.422)
```

5. Hacer una llamada al servicio para verificar que la conexión funciona correctamente:

   `http://localhost:8080/provedor`

   `http://localhost:8080/provedor/1`

6. Una vez que se tenga corriendo correctamente la aplicación, se deben realizar los siguientes ejercicios:

   - Realizar una petición POST para registrar un nuevo proveedor. El nombre de la empresa debe ser el número de cuenta del alumno que envía la práctica. El registro debe ser insertado en la base de datos que se encuentra en AWS.
   - Realizar las operaciones CRUD para los empleados. Se debe considerar la información de la dirección, la cual se encuentra dentro de la tabla region, como parte de las peticiones. Para esto, se deben crear los siguientes endpoints:

- Operación GET *http://localhost:8080/empleado*
  Response:

```
[{
  "idEmpleado":1,
  "nombre":"Jefferson",
  "apellido":"Gutierritos",
  "titulo":"Computólogo",
  "tituloDeCortesia":"joven",
  "fechaNacimiento":"25-04-1980",
  "fechaContratacion":"25-04-2012",
  "telefonoCasa":"5512345678",
  "extension":"55",
  "email":"jeff.gut@mail.com",
  "foto":"foto",
  "notas":"nota nota nota nota nota",
  "reportaAEmpleado":1,
  "pathFoto":"path",
  "region": {
              "idRegion":1,
              "direccion":"calle Z",
              "ciudad":"México",
              "codigoPostal":"03300",
              "region":"norte",
            }
}]
```

  Nótese que el resultado es una lista de empleados.
- Operación GET *http://localhost:8080/empleado/{idEmpleado}*
  Response:

```
{
  "idEmpleado":1,
  "nombre":"Jefferson",
  "apellido":"Gutierritos",
  "titulo":"Computólogo",
  "tituloDeCortesia":"joven",
  "fechaNacimiento":"25-04-1980",
  "fechaContratacion":"25-04-2012",
  "telefonoCasa":"5512345678",
  "extension":"55",
  "email":"jeff.gut@mail.com",
  "foto":"foto",
  "notas":"nota nota nota nota nota",
  "reportaAEmpleado":1,
  "pathFoto":"path",
```

```
  "region": {
              "idRegion":1,
              "direccion":"calle Z",
              "ciudad":"México",
              "codigoPostal":"03300",
              "region":"norte",
            }
}
```


Nótese que el resultado es un solo registro de empleado.

- Operación POST *http://localhost:8080/empleado/*
  Body:


```
{
  "nombre":"Jefferson",
  "apellido":"Gutierritos",
  "titulo":"Computólogo",
  "tituloDeCortesia":"joven",
  "fechaNacimiento":"25-04-1980",
  "fechaContratacion":"25-04-2012",
  "telefonoCasa":"5512345678",
  "extension":"55",
  "email":"jeff.gut@mail.com",
  "foto":"foto",
  "notas":"nota nota nota nota nota",
  "reportaAEmpleado":1,
  "pathFoto":"path",
  "region": {
              "direccion":"calle Z",
              "ciudad":"México",
              "codigoPostal":"03300",
              "region":"norte",
            }
}
```


Response:


```
{
  "idEmpleado":1,
  "nombre":"Jefferson",
  "apellido":"Gutierritos",
  "titulo":"Computólogo",
  "tituloDeCortesia":"joven",
  "fechaNacimiento":"25-04-1980",
  "fechaContratacion":"25-04-2012",
  "telefonoCasa":"5512345678",
  "extension":"55",
```

```
  "email":"jeff.gut@mail.com",
  "foto":"foto",
  "notas":"nota nota nota nota nota",
  "reportaAEmpleado":1,
  "pathFoto":"path",
  "region": {
              "idRegion":1,
              "direccion":"calle Z",
              "ciudad":"México",
              "codigoPostal":"03300",
              "region":"norte",
            }
}
```


Nótese que el body no incluye identificadores ni de empleado ni de region.

- Operación PUT *http://localhost:8080/empleado/{idEmpleado}*
  Body:


```
{
  "nombre":"Jefferson",
  "apellido":"Gutierritos",
  "titulo":"Computólogo",
  "tituloDeCortesia":"joven",
  "fechaNacimiento":"25-04-1980",
  "fechaContratacion":"25-04-2012",
  "telefonoCasa":"5512345678",
  "extension":"55",
  "email":"jeff.gut@mail.com",
  "foto":"foto",
  "notas":"nota nota nota nota nota",
  "reportaAEmpleado":1,
  "pathFoto":"path",
  "region": {
              "direccion":"calle Z",
              "ciudad":"México",
              "codigoPostal":"03300",
              "region":"norte",
            }
}
```


  Response:


```
{
  "idEmpleado":1,
  "nombre":"Jefferson",
  "apellido":"Gutierritos",
```

```
  "titulo":"Computólogo",
  "tituloDeCortesia":"joven",
  "fechaNacimiento":"25-04-1980",
  "fechaContratacion":"25-04-2012",
  "telefonoCasa":"5512345678",
  "extension":"55",
  "email":"jeff.gut@mail.com",
  "foto":"foto",
  "notas":"nota nota nota nota nota",
  "reportaAEmpleado":1,
  "pathFoto":"path",
  "region": {
              "idRegion":1,
              "direccion":"calle Z",
              "ciudad":"México",
              "codigoPostal":"03300",
              "region":"norte",
            }
}
```

Nótese que el body no incluye identificadores ni de empleado ni de region.

- Operación DELETE *http://localhost:8080/empleado/{idEmpleado}*

2.4. Punto Extra

Se obtendrán dos puntos extra sobre la práctica, si agregan interfaz gráfica para consumir los endpoints que construyeron en la práctica.

3. Entregable

Se deberá agregar los endpoints solicitados en la actividad a la aplicación existente, de manera que se puedan consumir tanto los endpoints existentes como los que agregaron. Su código debe ser conciso y lo mas claro posible.

La fecha de entrega es el día Lunes 1 de Junio de 2020.