Estrategias de Persistencia

Primer Cuatrimestre 2020.

Clase Nro 3.

En esta clase Nro. 3, vamos a ejecutar códigos en javascript con la tecnología Nodejs, para esta ejecución de códigos e instalación de dependencias sugiero hacer grupos de no más de 3 personas de manera virtual, a modo que entre todos puedan solucionar los inconvenientes que les puedan aparecer en el período de instalación de dependencias.

Esta clase durará 2 semanas ya que hay muchas dependencias e instalaciones que hacer para poder empezar a realizar los ejercicios en Nodejs.

En esta clase vamos a ver los siguientes puntos.

1) Que es un ORM
2) Instalacion de paquetes y dependencias.
3) Ejecucion de codigos simples con Nodejs y Sequelize.

1) ¿Qué es un ORM?

En el siguiente link vamos a encontrar un video interesante sobre qué es y cómo se utiliza un ORM.

https://www.youtube.com/watch?v=pS1nrxDj3yM

2) Para instalar Nodejs y sus dependencias vamos a realizar las acciones que figuran a continuacion:

   IMPORTANTE: La instalación siguiente, se realizará en un entorno LINUX, en el caso que haya alumnos que no posean este sistema operativo, sugiero instalarse un linux Ubuntu 18.04, la distribución Linux es gratuita y simple de instalar.

   a) Installar CURL esto es un paquete que nos va a permitir realizar un request y asi bajar las dependencias yversion que querramos.

   Para lo cual vamos a ejecutar lo siguiente:

   apt install curl

b) Una vez que temenos instalado CURL, vamos a realizar un request a la siguiente ruta para descargar Nodejs.

curl -sL https://deb.nodesource.com/setup_9.11 | sudo -E bash –

c) Ahora vamos a descargar gcc que es una dependencia necesaria para nuestro codigo.

sudo apt-get install gcc g++ make

d) Ahora vamos a ejecutar la instalacion de Nodesj

sudo apt-get install -y nodejs

e) Ahora ejecutaremos la instalacion de NPM, NPM es un producto que nos va a permitir installar dependencias dentro de nuestro directorio de trabajo.

sudo npm install -g npm

f) Con los siguientes commandos verificamos la correcta instalacion de Nodejs y NPM

- nodejs –version
- npm -v

g) Ahora vamos a instalar sequelize que es un ORM propio de Nodejs.
npm install --save sequelize

h) Ahora instalaremos mariadb-server:
sudo apt-get install mariadb-server

i) Paso siguente instalaremos el conector de mariadb dentro de nuestro proyecto.
npm install --save mariadb

3) En esta parte vamos a ejecutar los ejercicios que voy a compartir en la carpeta del campus, con el nombre “Ejemplos”.

Lo primero que debemos hacer es conocer la sintaxis de lo que queremos ejecutar, así que a continuación voy a explicar lo que significa cada una de las líneas de nuestro programa.

Una cosa a tener en cuenta, es que este programa va a crear una tabla llamada users dentro de la base de datos prueba, si se quiere correr por segunda vez el programa, se deberá borrar a mano la tabla users.

Explicacion:

Ejemplo de creacion de un conexión:

```
//En esta linea se hace referencia a que el programa va a requerir el ORM sequelize
const Sequelize = require('sequelize');
```

```
////////////////////////////////////////////////////////////////////////
// En esta seccion se coloca, la base de datos, el usuario y el password, a la que vamos a acceder.
// IMPORTANTE: CREAR LA BASE DE DATOS ANTES DE CORRER EL EJERCICIO. EN ESTE CASO SE LLAMA "prueba"
// Tambien deben verificar los datos de acceso, si su base de datos no tiene usuario y clave
// coloquen -> new Sequelize('prueba', 'root', '', {
const sequelize = new Sequelize('prueba', 'root', 'root', {
  host: 'localhost',
  dialect: 'mariadb' /* aqui va el dialecto que va a interpretar nuestro orm,
                        para el caso creo que todos tienen instalado mysql o mariadb
                        one of 'mysql' | 'mariadb' | 'postgres' | 'mssql' */
});

////////////////////////////////////////////////////////////////////////
```

```
////////////////////////////////////////////////////////////////////////////////
// En esta seccion lo que vamos a realizar es la conexion a la base de datos,
//en el caso de no poder autenticar la conexion a la base de datos
//arrojara el error que detallemos.
sequelize
  .authenticate()
  .then(() => {
    console.log('Connection has been established successfully.');
  })
  .catch(err => {
    console.error('Unable to connect to the database:', err);
  });
////////////////////////////////////////////////////////////////////////////////
```

Ejemplo de creacion de un modelo:

```
////////////////////////////////////////////////////////////////////////
// En la siguiente seccion vamos a realizar una clase de tipo Modelo.
// los modelos son los que llevan la estructuras de nuestras tablas, es decir, que
// si quisieramos poner en nuestra tabla un campo "nombreYApellido" y que este
// sea un STRING lo pondriamos de la siguiente manera: nombreYApellido: Sequelize.STRING,

class Cars extends Sequelize.Model {}
Cars.init({
  firstName: Sequelize.STRING,
  lastName:Sequelize.STRING
}, { sequelize, modelName: 'users' });
////////////////////////////////////////////////////////////////////////
```

Ejemplo de una insercion de un registro en nuestra tabla:

```
////////////////////////////////////////////////////////////////////////////
// En esta seccion vamos a hacer mencion al modelo creado arriba y vamos a insertar un regitro.
sequelize.sync()
  .then(() => Cars.create({
    firstName: 'Pedro',
    lastName: 'Rodriguez'
  }))
  .then(jane => {
    console.log(jane.toJSON());
  });
////////////////////////////////////////////////////////////////////////////
```

Ejemplo de actualizacion de un registro:

```
// Si quisieramos actualizar el registro creado podriamos
// ejecutar las siguientes lineas.
//
// En donde: se busca en la tabla users  un registro en donde
// lastName es igual a 'Marcelli' y se actualiza el firstName con "XXXXXXXXXX"
User.update({ firstName: "XXXXXXX" }, {
  where: {
    lastName: 'Marcelli'
  }
}).then(() => {
  console.log("Done");
});
```

Actividad práctica.
La idea de esta clase Nro. 3 y Nro. 4, es que puedan ejecutar los ejercicios que figuran en la carpeta ejemplos y puedan entender que es lo que sucede en una de las lineas.