Licenciatura en Sistemas de Información
Bases de Datos NSQL

PRACTICA #2 - Node JS y Redis

Usando Redis en NodeJS
Ahora que ya sabemos cómo trabajar en Docker con un contenedor de Redis veamos cómo nos podemos conectar desde NodoJS:

1. Lo primero que haremos será crear un proyecto en Node JS de la siguiente manera
   mkdir api
   cd mkdir
   npm init

2. Luego agregamos las dependencia que necesitaremos *express* y *redis* desde la consola
   npm install redis –save-dev
   npm install express –save-dev

3. O las agregamos en el package.json y las instalamos todas desde la consola
   npm install
4. Luego creamos un script que llamaremos app.js

```
//incluimos redis a nuestro script
var redis = require('redis');

//creamos un cliente
var redisClient = redis.createClient();

redisClient.on('connect', function() {
    console.log('Conectado a Redis Server');
});
```

5. Ahora ejecutamos el script anterior con el contenedor de redis en docker activo, debería mostrar por consola que nos hemos conectado al servidor de redis
6. Probemos ahora almacenar algunos datos
7. Ahora intentemos recuperarlos para ver si todo va bien
8. Ahora carguemos una lista y mostremos su contenido
9. Muestre los resultados del listado anterior en el localhost

Usando Docker Compose (Haciendo una Receta)

10. Primero vamos a crear el archivo Dockerfile

```
FROM node:latest

WORKDIR /api
COPY api/ .
```

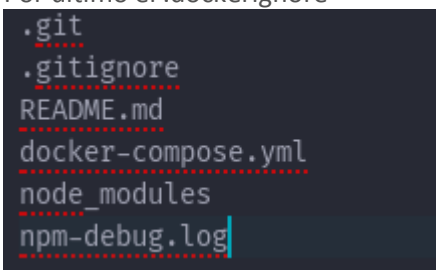
11. Segundo creamos el docker-compose.yml

```yaml
web:
  build: .
  command: sh -c 'npm install; npm start'
  ports:
    - '3000:3000'
  volumes:
    - /home/walter/nodeProjects/star_wars/api:/api
  links:
    - "db:redis"
db:
  image: redis
  ports:
    - "6379:6379"
```

12. Ahora modifiquemos un poco el script

```javascript
var redis = require('redis')
var express = require('express')
var app = express()
var port = 3000


var cliente = redis.createClient(6379, 'redis')
app.set('port', port)

cliente.on('connect', function(){
  console.log('conectado a redis');
})

cliente.lpush("I", "luke", "yoda", "han solo", "chewbacca", redis.print)
cliente.lrange("I", 0, -1, function(err, value){
  console.log(value)
  for (var i in value){
    console.log(value[i]);
  }
});

app.listen(app.get('port'), (err) => {
  console.log(`Server running on port ${app.get('port')}`)
})
```

13. Por último el .dockerignore



Ahora que ya maneja los conocimientos básicos genere una lista para cada uno de los episodios de la saga de Star Wars, en los cuales deberá poder cargar los correspondientes personajes.

1. Genere un ruta agregar personajes, la cual reciba como parámetro el número episodio y el nombre del personaje.
2. Genere una ruta para quitar personajes, ídem anterior.
3. Genere una ruta para listar los personajes de un episodio, la cual reciba como parámetro el número episodio.
4. Realice las misma actividades con componentes gráficos y añádale estilos (para no ser tan rustico).