Prueba Técnica: Node.JS Developer - Bquate

¡Hola!

Felicidades por haber llegado hasta esa etapa del proceso de selección para la vacante de desarrollador Node.JS en Bquate (https://bquate.com/home).

Abajo vas a encontrar los 3 ejercicios que elaboramos para analizar cómo resuelves problemas y presenta los resultados. No es obligatorio desarrollar los 3 ejercicios, pero recomendamos que lo hagas para que puedas alcanzar una mayor nota en la evaluación.

En cuanto tengamos los resultados de la evaluación se lo compartiremos contigo.

¡Suerte!

CHALLENGE 1

Recommendations:

1. Consistency and cohesion of the code
2. Following of available best practices for the language being used
3. Understanding of object oriented programming principles
4. Assets and UI/UX are implemented based on requirements

Write a program that prints all the numbers from 1 to 100. However, for multiples of 3, instead of the number, print "Multi". For multiples of 5 print "IT". For numbers which are multiples of both 3 and 5, print "Multipli".

But here's the catch: you can use only one `if`. No multiple branches, ternary operators or `else`.

Requirements

- 1 if
- You can't use `else`, `else if` or ternary
- Unit tests
- We are big fans of JS, Node.JS and React.

Submission

You can create a public repository on your GitHub account and send the link to us, or just send us a zip file.

CHALLENGE 2

Using the api from https://musicbrainz.org/ which you can get documentation here

https://musicbrainz.org/doc/Development/XML_Web_Service/Version_2

Build a frontend interface to search artist, discs (releases) and tracks (recordings), the tables for listing must have:

- Image
- Name
- Little description
- On click name show element with mayor description

1. You must think this is for music search engine so be creative.
2. To get images for releases you can use https://coverartarchive.org/release/**<mbid>**/front which it return jpg or png files, more information here https://musicbrainz.org/doc/Cover_Art_Archive/API.
3. Don't be afraid, we won't you create a music player, just read data.
4. We are big fans of NodeJS, you can use Gulp, Grunt or any other task processor, any toolkit, sass, whatever you have in javascript and ecma6, also you can use **React, we are big fans to.**
5. Must be an MLP (Minimum Lovable Project).
6. Put your code in github and share us.
7. Deploy it online using any services like heroku, netlify or other.
8. Let us know your MVC skill.

CHALLENGE 3

SQL

Using the file bd_musica.sql that we have provided you, using Nodejs and MySql, you must create the database needed for the following query exercises:

1. Create a View that lists all tracks with id, title, title of the album to which it belongs, email and country of origin of the user.
2. List all albums from Peru that which genre is ROCK.
3. Find any track that does not have artist and update it with the user name, only tracks without artist if you modify other tracks you screw up.
4. Set status 0 to any album which not have tracks and show them
5. Set status 0 to any user which not have track or albums and show them
6. List all tracks that here genre is different to genre of here album
7. Create a Web API, REST and CRUD for all the tables and for consult views. the API most be able to filter, paginate and sort for any column. PLUS: multiple filters, multiple sorting