TP3 – Servidores en Node.js

Desarrollar un servidor en Node.js con express que realice las siguientes tareas:
(se mostrará por consola cuando el servidor esté listo para operar y en que puerto lo está haciendo)

1) Responda en la ruta raíz un mensaje de acuerdo a la hora actual: si dicha hora se encuentra entre las 6 y las 12hs será 'Buenos dias!', entre las 13 y las 19hs 'Buenas tardes!' y de 20 a 5hs 'Buenas noches!'.

2) Así mismo, dispondrá de otra ruta get '/random' la cuál iniciará un cálculo de 10000 números aleatorios en el rango del 1 al 20. Luego de dicho proceso, el servidor retornará un objeto cuyas claves sean los números salidos y el valor asociado a cada clave será la cantidad de veces que salió dicho número.

3) Definir otra ruta get llamada '/info' que sea capaz de leer el archivo package.json y devuelva un objeto con el siguiente formato y datos:

```
let info = {
   contenidoStr: (contenido del archivo leído en formato string),
   contenidoObj: (contenido del archivo leído en formato objeto),
   size: (tamaño en bytes del archivo)
}
```

Esta ruta será capaz de:

- Mostrar por consola el objeto info luego de leer el archivo.
- Guardar el objeto info en un archivo llamado info.txt dentro de la misma carpeta de package.json, preservando el formato de representación del objeto en el archivo (tabuladores, saltos de línea, etc)
- Utilizar la lectura y escritura de archivos en modo asincrónico con async await.

4) Por último, declarar una ruta get '/operaciones', que reciba por query-params dos números y la operación a realizar entre ellos. Ejemplo: …./operaciones?num1=5&num2=6&operacion=suma
Las operaciones válidas serán: suma, resta, multiplicación y división.
Si no se ingresa alguno de estos parámetros, si los tipos de datos no corresponden ó si operación no es válida, se devolverá un error mediante un objeto con la siguiente estructura:

```
{
   error:  {
            num1: { valor: x, tipo: y },
            num2: { valor: x, tipo: y },
            operacion: { valor: x, tipo: y }
   }
}
```

Si todo está correcto, devolver un objeto que contenga los dos números ingresados, la operación y el resultado.

- Utilizar import (ES Modules) para todos los procesos y separar en módulos el desarrollo.
- Considerar lo necesario para que este proyecto puede funcionar de forma local o hosteado en glitch.com.
- Subir el ejercicio a github (ignorar la subida de node_modules) y hacer el deploy en glitch desde dicho repo. Fijar la versión mínima de Node.js para que glitch instale la versión correcta de node y funcionen los import de ES Modules.