Ejercicio final:

1. Diseñar una REST API en Node

2. Implementar la REST API usando express o su propia libreria request / response

3. Integrar la solución a un mecanismo de almacenamiento NO SQL (via mongoose o mongodb)

4. Tener unit tests de la solución; con cobertura de código > 80 %.

La solución propuesta por los candidatos debe ser:

1. Escalable

2. Performante

Cree una REST API que en Node que permita:

* Operaciones CRUD para todos los recursos
* Registro de usuarios para acceder a la API
* Token de acceso para acceder a la API (login)
* Todos los endpoints requieren autenticación, con excepción del root endpoint.
* Existen dos niveles de usuarios de la API:
  + Operador
  + Administrador
* El nivel de usuario se dá a la hora de registrarlo.
* Un usuario administrador puede hacer todo, un usuario operador no puede realizar ninguna acción destructiva.
* Todas las acciones de los usuarios deberán logearse a un archivo, con la siguiente información:
  + Fecha y hora de la acción
  + Nombre del usuario e id del mismo
  + Acción realizada.
* Todos los requests deberán ser logueados a consola, con la siguiente información:
  + Fecha y hora
  + URL
  + Método

**Si queda alguna duda, pueden consultar, pero son bienvenidos a decidir por Uds cual es la mejor alternativa siempre que puedan justificar esa decisión.**

Consideraciones:

* La respuesta a un request no puede ser mayor a 100ms
* Se debe proporcionar un diagrama de alto nivel de la solución propuesta.
* Todo módulo implementado debe contar con test unitarios.
* Se deberá contar con procesos de chequeo del código (jslint como mínimo)
* Usando grunt o gulp, automatizar los tests y los procesos de chequeo del código cada vez que se inicie el servidor.