



# Enunciado Reto 4

Una nueva semana, unos nuevos requerimientos. Al parecer el equipo de iMaster ha avanzado en el proceso y destacado entre sus competidores, aunque ha sido una muy buena noticia y cada vez esta más cerca de ganarse el contrato del proyecto, el 4 y 5 reto, van a filtrar de una vez por todas las empresas que siguen y las que no.

Los retos siguientes están orientados al **backend**, a la lógica del servidor, a las bases de datos, rutas, controladores, middlewares, una gran cantidad de términos con los que el equipo ha quedado un tanto inquieto. Sin embargo, saben que cuentan con usted para cumplir con honores el reto propuesto; el contexto es el siguiente:

- Se ha dividido la verificación del **backend** en 2 etapas, una será el reto 4 y otra será el reto 5.
- Para lo correspondiente a la etapa 1, la interventoría solicitó entregar resultados en cuanto a la creación y autenticación de usuarios pero es de vital importancia que las transacciones obtengan unos códigos **http** de estado correctos.
- El equipo agregó diferentes códigos a las respuestas de las peticiones pero no cree que en todos este usando el correcto y si la lógica esta bien pero el código de estado se envía incorrecto, la prueba falla.
- El sistema integra un entorno de pruebas llamado "**test**", el cual ejecuta una base de datos **sqlite** dentro del proyecto, por lo que si requiere ejecutar pruebas del funcionamiento del aplicativo (levantar el servidor y hacer peticiones manualmente), deberá crear y configurar la base de datos en un entorno de desarrollo "**development**".
- Las configuraciones de la base de datos podrá encontrarlas en la ruta **config -> config.json** y solo deberá modificar, si así lo desea, la configuración para "**development**". Para verificar las pruebas utilizando el comando **npm run test**, no es necesario que cree la base de datos.
- Por ahora solo se probarán las funcionalidades de la ruta **/api/usuario** tales como **add, list, login, update, actívale y deactivate**.
- Finalmente el proyecto debe aprobar todas las pruebas para poder continuar a la siguiente etapa, sobra indicar que el proyecto debe conservar sus funcionalidades y estructura.



Ahora que ya conoce el estado inicial del proyecto y algunas generalidades de su funcionamiento, es hora de entrar en materia y conocer los requerimientos que se deben ajustar:

- El equipo a configurado la ruta **/api/usuario** en el archivo inicial del servidor, **index.js**, pero cada hacen una petición obtienen un error, al parecer ya hay un manejador configurado para la ruta **/.../usuario** y cree que les esta sobrando algo o duplicando una parte de la ruta en el manejador principal. El equipo recuerda que la forma para revisar sería manejador principal, manejador de rutas, ruta y controlador pero no entienden muy bien del tema. Usted debe detectar el error y corregirlo para poder realizar peticiones correctamente.
- En los métodos del controlador usuario están correctamente definidas las funciones con los nombres indicados como **add, list, login, update, activate y deactivate**. Pero al parecer las rutas no están ejecutando el método correcto del controlador, al parecer hay algunos nombres mal escritos.
- Finalmente se entrega un listado de códigos de estado de http que se deben aplicar a la respuesta de cada petición, debido a que el equipo desconocía del tema, puso códigos **404 Not Found** en todas las respuestas que implementó. La especificación es la siguiente:
  - Toda petición bien procesada y exitosa debe retornar un código de estado **200 OK**.
  - Toda petición descartada por algún problema en el procesamiento debe tener un código de estado **500 Internal Server Error**.
  - Los errores de contraseña incorrecta o usuario sin permisos deben retornar un código de estado **401 Unauthorized**.
  - Las peticiones que no encuentren un resultado, ya sea un usuario o una página deben retornar un código de estado **404 Not Found**.
- Los comandos, ayuda e indicaciones iniciales de los proyectos anteriores siguen vigentes:
  - Usted puede ejecutar las pruebas todas las veces que considere necesarias, tanto de manera local como en la plataforma. La plataforma solo guarda la nota del último intento realizado sea un cero, un cinco o cualquier valor intermedio.
  - La plataforma realiza la evaluación automática del proyecto que usted esta presentando, tarda unos minutos en ejecutar cada una de las pruebas por lo que podrá ver el resultado de la evaluación pasados cerca de 20 minutos, en caso de que no sea visible en el libro de calificaciones puede que se haya presentado algún error en el proceso de evaluación, en este caso la



recomendación es intentar nuevamente enviar el **.zip** a la plataforma.

- El contenido de su proyecto previo a la creación del **.zip**, debe estar guardado en una carpeta nombrada con su número de documento de identidad y tener una estructura como la siguiente, pero con todas las correcciones ya aplicadas y sin la carpeta **node\_modules**:

```
> config
> controllers
> middlewares
> migrations
> models
> node_modules
> routes
> seeders
> services
> test
> .gitignore
> database.sqlite3
> index.js
> package-lock.json
> package.json
> README.md
```

Recuerde que el reto puede ser subido por un único integrante del equipo, indicando el número de documento de identidad de los demás compañeros que hayan trabajado en el proyecto, separados por comas y sin puntos u otra clase de separadores, ej.

1111111,2222222,3333333,4444444

La carpeta del proyecto debe ser nombrada solo con el número de documento de identidad del usuario que hace la carga a la plataforma así la presente por todo su equipo, ej.



El futuro digital  
es de todos

MinTIC

tes

Archivo Inicio Compartir Vista

tes

Buscar en tes

Nombre	Fecha de modificación	Tipo	Tamaño
11111111.zip	1/10/2021 4:23 p. m.	Carpeta compri...	1 KB

Acceso rápido

- Escritorio
- Documentos
- Descargas
- Imágenes
- img
- js
- src
- Usuarios
- OneDrive
- Este equipo
- Descargas
- Documentos
- Escritorio
- Imágenes
- Música
- Objetos 3D
- Videos
- Acer (C:)

1 elemento