

PARCIAL 1

Desarrollar un servidor API REST que cumpla con los siguientes requisitos:

- El servidor debe estar desarrollado con Node.js, preferentemente usando el framework express.
- La ruta raíz del servidor debe responder con un texto plano que diga “Servidor ok”. Es decir que al ingresar a la URL del servidor, este me debe responder con ese mensaje.
- Un método GET para poder obtener datos de la base de datos.
- Un método GET para poder obtener un elemento por id de la base de datos.
- Un método POST para poder agregar un elemento a la base de datos.
- Un método PUT para poder editar un elemento a la base de datos.
- Un método DELETE para poder eliminar un elemento a la base de datos.
- Respetar la arquitectura de carpetas (Rutas, Modelos, Controladoras)
- Los datos necesarios para conectar la base de datos con el servidor deben estar configurados desde una variable de entorno, no pueden estar directamente escritos en el código del servidor.
- El proyecto del servidor debe estar subido a un repositorio de Github a nombre del alumno (o uno de los alumnos del grupo).
- El repositorio debe contener código prolijo, segmentado en commits.
- El servidor debe estar deployado en Heroku, se debe poder acceder desde una URL desde cualquier navegador web con conexión a internet.
- Debe estar conectado a una base de datos noSQL hosteada en la nube usando MongoDB Atlas.
- Creación de un readme donde se especifiquen los pasos a seguir para la correcta ejecución de la aplicación, url del repositorio y nombre del o los alumnos.
- Creación de una colección de Postman con todos los métodos del servidor documentados, listos para ser testeados.

El proceso de evaluación será:

1. Abrir la URL que provee heroku y mostrar la leyenda que refiere a que el servidor se encuentra corriendo correctamente.
2. En Postman, se revisará que los endpoints estén contenidos en una colección.
3. Proba cada uno de los endpoints solicitados, los cuales deberán funcionar correctamente.
4. Revisar la calidad del código del backend en Github y que **el alumno tenga commits a su nombre**.
5. Revisar el correcto entendimiento de las funcionalidades desarrolladas, revisando el código y charlando sobre el flujo de datos en la aplicación.