**Proyecto Integrador N°2**

Para finalizar la **Primera Etapa** del curso de programación **Backend** de **Argentina programa 4.0**, deberás desarrollar una aplicación CRUD integrando MongoDB en tu proyecto Node.js.

En el proyecto deberás contemplar los siguientes puntos:

1) Crear un servidor HTTP local con **Node JS** que integre las dependencias de **Express JS, MondoDB** y **DotEnv**. Además, agregar como dependencia de desarrollo a **Eslint**.
2) Especificar en el archivo **package.json** tu nombre y apellido en la key **author**. Puedes descargar el archivo desde aquí.
3) Declarar las **variables de entorno** requeridas para tu proyecto.
4) Segurizar las **variables de entorno** en un archivo *.env.dist* e ignorar el archivo *.env* en **Git**.
5) Crear una base de datos en **MongoDB Cloud** denominada ***muebleria*** y una colección llamada ***muebles***.
6) Importar en dicha colección**,** los datos desde el archivo *data.json*. Puedes descargar el archivo desde aquí.
7) Crear un archivo denominado *connection_db.js* que administre la conexión con **MongoDB Cloud**.
8) Crear los siguientes **endpoints** para:
    a) Obtener los registros de todos los muebles (sin ordenar)
    b) Obtener los registros de los muebles filtrados y ordenados por una categoría
    c) Obtener los registros de los muebles por precio mayor o igual que un valor y ordenados por el precio de forma ascendente
    d) Obtener los registros de los muebles por precio menor o igual que un valor y ordenados por el precio de forma descendente
    e) Obtener el registro de un mueble por su código
    f) Crear un nuevo registro de un mueble (contemplar la validación del esquema)
    g) Actualizar el registro de un mueble por su código (contemplar la validación del esquema)
    h) Eliminar el registro de un mueble por su código
    i) Controlar las rutas inexistentes
9) Agregar las **validaciones** correspondientes en cada **endpoint**.

Restricciones y recomendaciones:

- La arquitectura de la plataforma es una API RESTful y por tal motivo, debe cumplir las restricciones y recomendaciones de REST.
- Implementar buenas prácticas de programación.
- Los endpoints se tienen que declarar y definir dentro del mismo archivo *server.js*.
- Emplear try/catch dentro de cada endpoint.
- Diseñar un generador de códigos robusto.

Requisitos de entrega:

A. La api debe cumplir con todas las **especificaciones definidas** en la **documentación del proyecto**. Puedes descargar el archivo desde aquí.
B. La api debe pasar los **casos de pruebas** diseñados en el archivo *proyecto2.test.js*. Puedes descargar el archivo desde aquí.
C. La api debe pasar la **comprobación de errores y formatos** por medio de **Eslint**. Puedes descargar el archivo de configuración desde aquí.

Instrucciones de entrega:

- Revisar detenidamente el código fuente de tu proyecto
- Verificar que funcione correctamente y que cumpla con los requisitos
- Comprimir el proyecto en formato **zip** (NO incluir la node_modules)
- Enviar el archivo comprimido por medio de form de google (verificar el mensaje de recepción)

Consideraciones:

Si desarrollaste junto al profe el proyecto de ejemplo que realiza un CRUD con MongoDB, ya tienes gran parte del código desarrollado, incluyendo la conexión con tu base de datos MongoDB resuelta. Aprovecha las bases de dicho código para adaptarlo a este proyecto integrador.

¡Éxitos en el desarrollo de tu proyecto!