ENTREGA DEL PROYECTO FINAL

Deberás entregar tu aplicación eCommerce Backend correspondiente a la última entrega de tu proyecto final.

CODER HOUSE

ENTREGA DEL PROYECTO FINAL

Formato: link a un repositorio en Github con el proyecto cargado.
Sugerencia: no incluir los node_modules

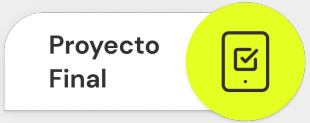
>>Consigna: Para culminar con el proyecto final, vamos a realizar las últimas reformas al desarrollo backend e-Commerce para que quede estructurado de acuerdo a los criterios y mecanismos que fuimos aprendiendo en este último trayecto del curso.

- En primer lugar la aplicación de servidor debe tener sus capas MVC bien definidas y en archivos separados. Debe existir la capa de ruteo, el controlador, la capa de lógica de negocio con los casos de uso y las validaciones y la capa de persistencia con los DAOs/DTOs o Repositories necesarios para soportar el o los sistemas de persistencia elegidos. En caso de ser más de uno, utilizar una factory para que podamos elegir el sistema de almacenamiento al inicio del servidor.

- El servidor debe disponer de configuraciones mediante variables de entorno, que permitan crear un ambiente para desarrollo y otro para producción, elegibles desde la variable de environment NODE_ENV al desplegar la aplicación. Como variables de configuración deberían estar el puerto de escucha del servidor, la persistencia elegida, el string de conexión a la base de datos (si hubiera varios sistemas de persistencia en base de datos considerar todos los casos y sus diferencias), API keys y todo lo que sea necesario que esté en un archivo protegido fuera del código del servidor. Pensar en utilizar bases de datos y servidores locales para la configuración de desarrollo.

ENTREGA DEL PROYECTO FINAL

Formato: link a un repositorio en Github con el proyecto cargado.
Sugerencia: no incluir los node_modules



- Se debe analizar que el hecho de incorporar un caso más de uso en la lógica del servidor, sea un proceso de agregar código y no de modificar el existente.
- Si agregamos un sistema más de persistencia, deberíamos agregar sólo el módulo nuevo y reformar la factory, mientras que resto del proyecto: router, controlador, lógica de negocio, validaciones y otros sistemas de persistencia no deberían sufrir modificaciones para soportar la nueva función.
- El código debe quedar bien tabulado, legible, ordenado y comentado ni por exceso ni por defecto.
- Las funciones o clases que se por sí solas expliquen su misión, no necesitan ser explicadas (salvo que amerite por complejidad).
- Para concluir, subir el desarrollo completo a Heroku o algún PASS de preferencia, seleccionando la configuración a producción de modo de utilizar los parámetros adecuados de funcionamiento y la persistencia en la nube a través de bases de datos como servicio (DBaaS).

👉 Para más detalle, puedes consultar la Consigna Proyecto Final Curso Backend.

CODER HOUSE