

Esta clase va a ser

- grabada

Certificados oficialmente por

 PedidosYa

CODERHOUSE

Clase 44. PROGRAMACIÓN BACKEND

Práctica integradora

Certificados oficialmente por



CODERHOUSE

Temario

43

Frameworks de desarrollo: Nestjs (Parte II)

- ✓ Conexión de Nestjs a Mongo
- ✓ Autenticación con JWT

44

Segunda Práctica Integradora

- ✓ [Skills](#)
- ✓ [Práctica integradora](#)

45

Proceso principal del servidor + Global & Child process

- ✓ Despliegue de nuestro aplicativo
- ✓ Configuración de pipeline en Railway.app

Objetivos de la clase

- Hacer una integración práctica de todos los conceptos vistos hasta el momento, bajo el desarrollo de un proyecto paralelo a nuestro proyecto final.

¿Cómo organizar la clase?

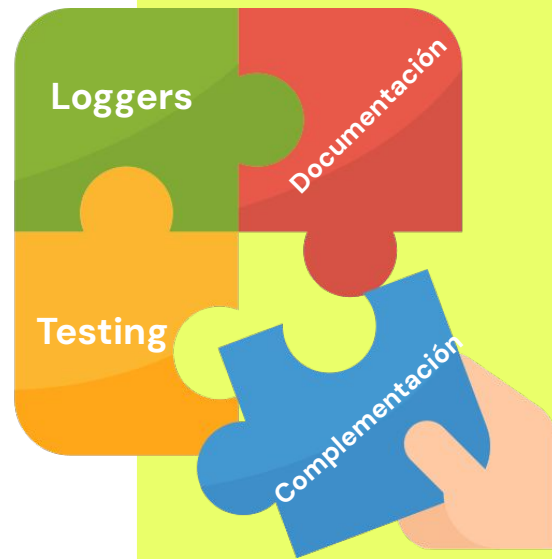
Estructura general de la clase	Tiempo de dedicación	Enfoques
Skills	10 min	<ul style="list-style-type: none">✓ No detenerse a explicar cada skill✓ Utilízalo como motivador, no como stopper.
Práctica integradora	1hr 20 min (contemplando break a tu elección)	<ul style="list-style-type: none">✓ Recuerda que estás retomando un proyecto de práctica integradora previo. No pierdas tiempo repasando cosas de la práctica integradora anterior.✓ Esta práctica integradora supone un caso muy particular, y es que es de carácter más relajado, ya que las cosas a integrar son bastante sencillas, cosas como logging y documentación, haciendo que el resto de los temas sea bastante subjetivo según sea el caso.
Presentación del desafío	10 min	
Espacio de dudas y consultas	15 min	<ul style="list-style-type: none">✓ Puedes traer dudas con respecto a contenidos vistos en el curso o bien relacionadas con el terreno profesional.

¿Qué estamos por ver?

Práctica integradora

La mejor forma de repasar los temas vistos hasta el momento, es hacer un repaso integrado de todos los elementos.

Si bien es correcto repasar el código parte por parte, es importante que como desarrolladores comencemos a trabajar en nuestra **lógica de integración**, es decir, tenemos que tener siempre contemplado el cómo vamos a juntar todo lo aprendido, para tener un proyecto sólido



Elementos a integrar

Elementos a integrar

En el desarrollo de esta práctica de integración, repasarás y conectarás los siguientes conceptos:

- ✓ Logging
- ✓ Documentación
- ✓ Complementación

Skills para esta práctica integradora

Skills para Logging

- ✓ Comprender la importancia de utilizar un logger
- ✓ Comprender el uso de Winston Logger y aplicarlo
- ✓ Entender los diferentes tipos de transportes
- ✓ Entender sobre los niveles de logging
- ✓ Configurar nuestros propios niveles de logging



Skills para documentación



**SWAGGER
UI**

- ✓ Comprender la importancia de documentar
- ✓ Comprender el uso de Swagger para documentación
- ✓ Realizar la Swaggerización por archivos de cada Módulo
- ✓ Comprender sobre los elementos que compone un módulo Swaggerizado (schemas, inputs, requestBodies, responses, etc)

Skills para testing

- ✓ Comprender sobre módulos de Testing
- ✓ Conocimientos de realización de Testing unitario
- ✓ Conocimiento sobre Test de integración
- ✓ Uso de Mocha
- ✓ Uso de Chai
- ✓ Uso de SuperTest



**SUPER
TEST**

Importante

La práctica hace al maestro... si consideras que aún necesitas practicar algún aspecto de los skills mencionados, siempre puedes repasar las clases previas. Sin embargo, esto no significa que no puedas avanzar.

¡Esta práctica integradora te ayudará a sentirte más seguro en estos temas!

Desarrollo de esta práctica integradora



1

Aplicación de un logger que utilice múltiples transportes

2

Documentación elemental del proyecto en cuanto a sus elementos principales

3

Testeo de funcionalidades principales de los módulos por separado e integrados.

¡Comenzamos!



Práctica de integración sobre tu ecommerce



DESAFÍO COMPLEMENTARIO

Consigna

Con base en el proyecto que venimos desarrollando, toca solidificar algunos procesos

Aspectos a incluir

- ✓ Mover la ruta suelta ***/api/users/premium/:uid*** a un router específico para usuarios en ***/api/users/***
 - ✓ Modificar el modelo de User para que cuente con una nueva propiedad "documents" el cual será un array que contenga los objetos con las siguientes propiedades
 - name: String (Nombre del documento).
 - reference: String (link al documento).
- No es necesario crear un nuevo modelo de Mongoose para éste.
- ✓ Además, agregar una propiedad al usuario llamada "last_connection", la cual deberá modificarse cada vez que el usuario realice un proceso de login y logout



DESAFÍO COMPLEMENTARIO

Aspectos a incluir

- ✓ Crear un endpoint en el router de usuarios **`api/users/:uid/documents`** con el método POST que permita subir uno o múltiples archivos. Utilizar el middleware de Multer para poder recibir los documentos que se carguen y actualizar en el usuario su status para hacer saber que ya subió algún documento en particular.
- ✓ El middleware de multer deberá estar modificado para que pueda guardar en diferentes carpetas los diferentes archivos que se suban.
 - Si se sube una imagen de perfil, deberá guardarlo en una carpeta **`profiles`**, en caso de recibir la imagen de un producto, deberá guardarlo en una carpeta **`products`**, mientras que ahora al cargar un documento, multer los guardará en una carpeta **`documents`**.
- ✓ Modificar el endpoint **`/api/users/premium/:uid`** para que sólo actualice al usuario a premium si ya ha cargado los siguientes documentos:
 - Identificación, Comprobante de domicilio, Comprobante de estado de cuenta



DESAFÍO COMPLEMENTARIO

Aspectos a incluir

En caso de llamar al endpoint, si no se ha terminado de cargar la documentación, devolver un error indicando que el usuario no ha terminado de procesar su documentación.
(Sólo si quiere pasar de user a premium, no al revés)

Formato

- ✓ Link al repositorio de GitHub con el proyecto completo (no incluir node_modules).

Sugerencias

- ✓ Corrobora que los usuarios que hayan pasado a premium tengan mayores privilegios de acceso que un usuario normal.

**¿Dudas, preguntas,
consultas?**

Muchas gracias.

Opina y valora
esta clase

#DemocratizandoLaEducación