**Entrega Final**

**Consigna:**

Mejorar la arquitectura del servidor desarrollado durante el curso, enfocándose en la implementación de patrones de diseño, manejo de roles y autorización, así como en la mejora de la lógica de negocio del ecommerce.

Aspectos a Incluir:

1. **Patrón Repository:**
   * Implementar el patrón Repository para trabajar con el DAO (Data Access Object) dentro de la lógica de negocio.
2. **Modificación de la Ruta**

**/current**

**:**

* + Evitar enviar información sensible del usuario. Enviar un DTO (Data Transfer Object) que contenga solo la información necesaria y no sensible.

1. **Sistema de Recuperación de Contraseña:**
   * Implementar un sistema de recuperación de contraseña que envíe un correo con un botón para restablecer la contraseña.
   * El enlace del correo debe expirar después de una hora de ser enviado.
   * Evitar que el usuario pueda restablecer la contraseña a la misma que tenía anteriormente.
2. **Middleware de Autorización:**
   * Crear un middleware que trabaje junto con la estrategia “current” para limitar el acceso a ciertos endpoints:
     + Solo el administrador puede crear, actualizar y eliminar productos.
     + Solo el usuario puede agregar productos a su carrito.
3. **Arquitectura Profesional:**
   * Aplicar una arquitectura más profesional en el servidor, utilizando patrones de diseño, manejo de variables de entorno y técnicas avanzadas como mailing.
4. **Mejora en la Lógica de Compra:**
   * Profundizar en los roles de los usuarios y las autorizaciones aplicables a cada rol en el contexto de las compras dentro del ecommerce.

Formato de Entrega:

* Link al repositorio de GitHub con el proyecto completo, excluyendo la carpeta

node\_modules

.

* Incluir el archivo

.env

 necesario para la configuración de las variables de entorno.

Esta entrega final busca consolidar todos los conocimientos adquiridos durante el curso, enfocándose en la mejora de la arquitectura, seguridad y profesionalización del servidor, preparándote para desarrollar aplicaciones backend robustas y bien estructuradas.

**Criterios:**

* **Implementación de DAO y DTO en Capa de Persistencia**:

Los DAOs y DTOs están adecuadamente estructurados y separados, siguiendo buenas prácticas de diseño y arquitectura. La transferencia de datos entre capas es eficiente y se minimiza el uso de consultas redundantes a la base de datos.

* **Patrón Repository y Lógica de Negocio:**

El patrón Repository se aplica de manera ejemplar, separando claramente la lógica de acceso a datos de la lógica de negocio. Las operaciones de negocio se realizan de manera eficiente y coherente utilizando los Repository.

* Middleware de Autorización y Seguridad de Endpoints:

El middleware de autorización se integra perfectamente con la estrategia "current", permitiendo delimitar el acceso a los endpoints según los roles de usuario de manera segura y eficiente.

* **Modelo de Ticket y Lógica de Compra:**

El modelo Ticket se crea correctamente con todos los campos necesarios y se implementa una lógica de compra robusta que verifica el stock de los productos, genera tickets y maneja compras completas e incompletas de manera eficiente. [Criterios de evaluación](https://drive.google.com/file/d/1cCLF7IUlV17AG9prMh7FYFOa9jqr_IVY/view?usp=drive_link)