

**Tecnologías de desarrollo en el servidor**

*Definición del proyecto Integrador*

**PRESENTA**

Lic. Ingeniería en Nanotecnología: Leonardo Daniel Hernández García

Lic. Ingeniería en Desarrollo de Software: Sergio Isaac Santana Jiménez

Profesora:

Francisco Javier Sevilla Medina

Tlaquepaque, Jalisco, 6 de febrero del 2025

### My-ecommerce Iteso

**Descripción**

Como estudiantes del ITESO, hemos experimentado el proceso de compra y venta a través del grupo de WhatsApp *"Ventas ITESO"*, donde los usuarios pueden ofertar y adquirir productos mediante mensajes. Sin embargo, este método presenta diversas limitaciones. Los compradores deben desplazarse constantemente por el chat para encontrar publicaciones anteriores sin garantía de que los productos sigan disponibles. Por su parte, los vendedores deben republicar con frecuencia para evitar que sus ofertas se pierdan entre tantas publicaciones. Una aplicación más estructurada optimizaría este proceso, brindando una experiencia más eficiente y accesible para todos.

**Integrantes:**

* Leonardo Daniel Hernández García
* Sergio Isaac Santana Jiménez

**Stack de tecnologías a utilizar:**

* La base de datos escogida es MongoDB, la cual es una base de datos no relacional.
* De framework se utilizaría bootstrap
* Tecnologías Node.js
* Express.js
* Mongoose
* Cors
* Stripr (API para gestionar pagos)
* Shopigy API (gestión de tiendas)

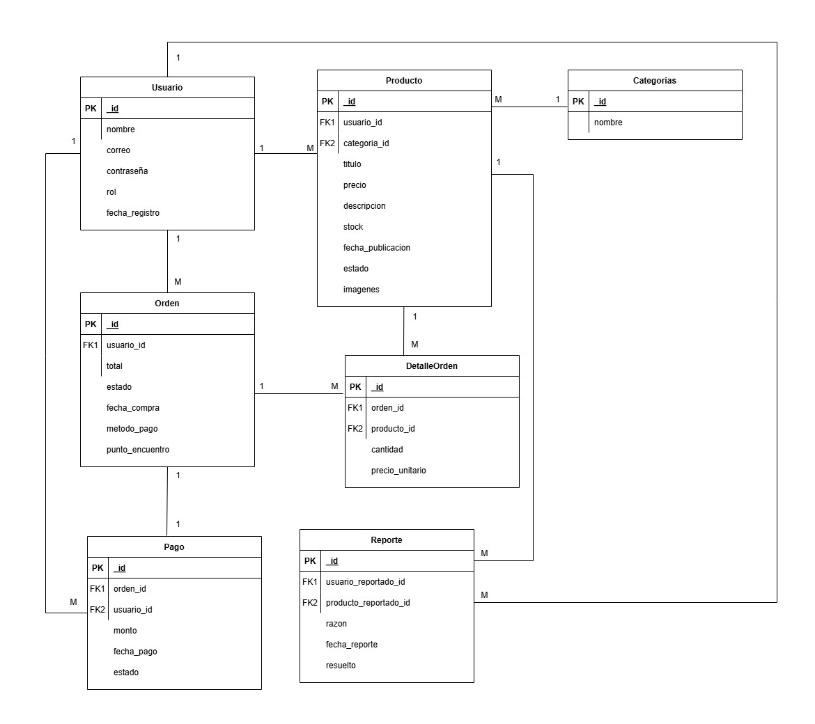
**Lista de roles que pueden registrarse y/o iniciar sesión en la aplicación:**

* Usuarios que cuenten con un correo institucional del ITESO.

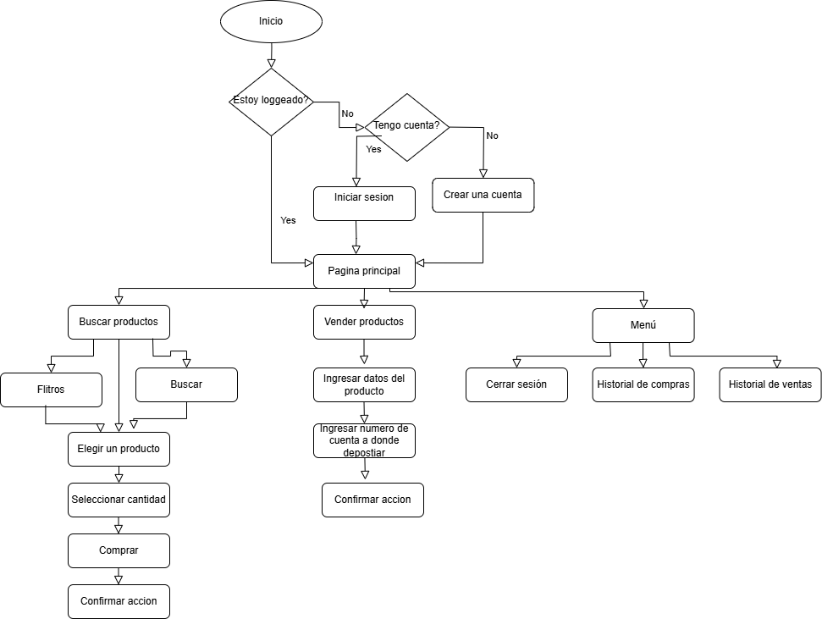
**Listado de usuarios y su rol/tarea:**

* Invitado (usuario que no ha iniciado sesión), solo puede ver contenido de la página, pero no puede comprar ni vender productos.
* Usuario (usuario que inicio sesión), puede agregar contenido a la pagina en forma de productos para su venta o puede comprar de otros usuarios.
* Administrador, puede dar de baja productos no apropiados y gestionar que se cumpla la política de la ecommerce, también puede de dar baja a los usuarios si rompen una norma.

**Diagrama Entidad-Relación:**



**Diagrama de flujo:**

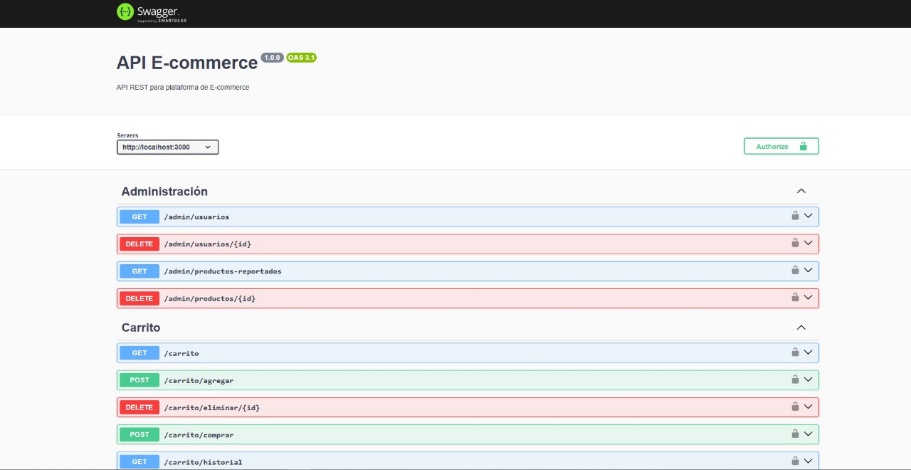


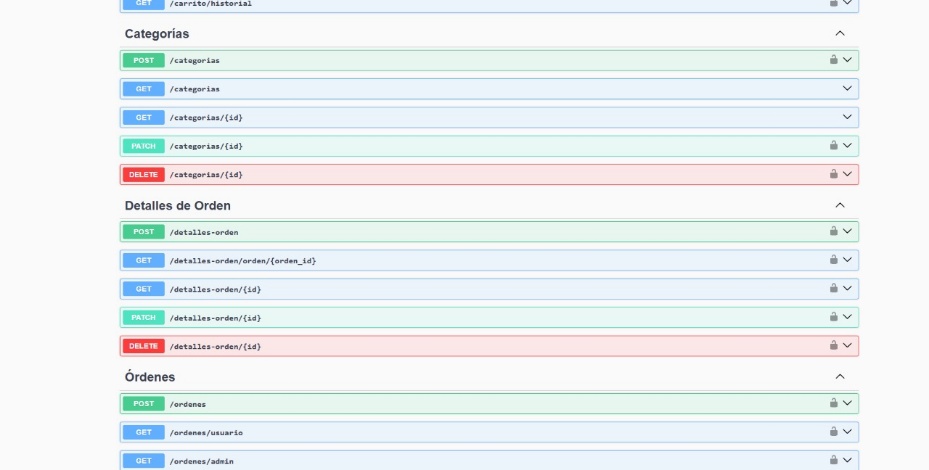
**Documentación:**

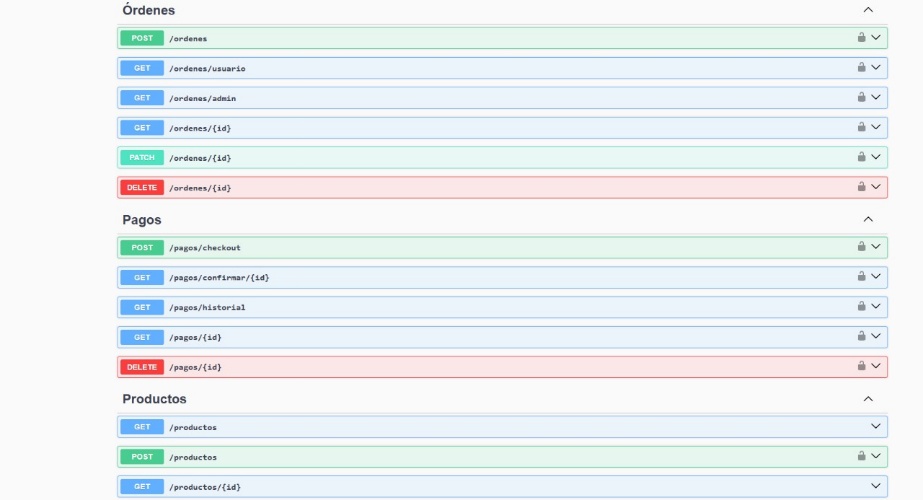
A continuación, se mostrará la configuración Swagger y su visualización, el endpoint creado para entrar a la documentación es localhost:3000/swagger :

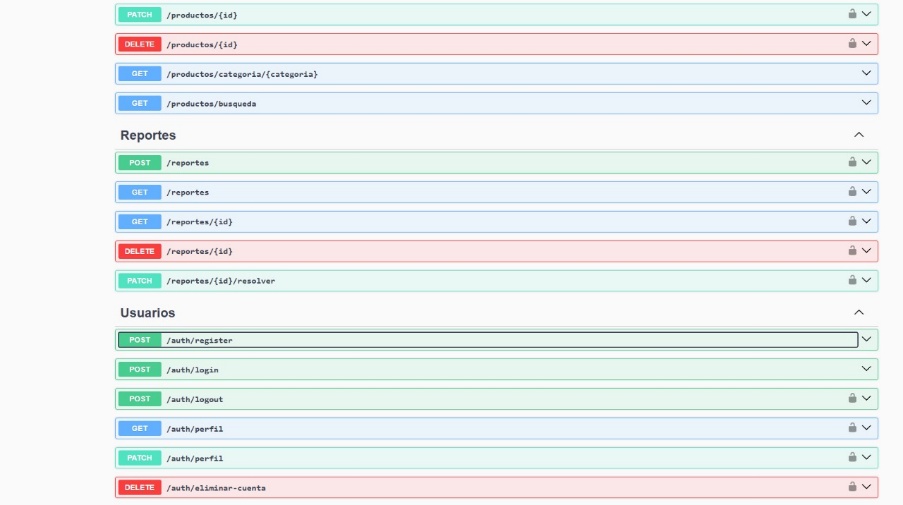


Imagen 1.- Configuración de Swagger.









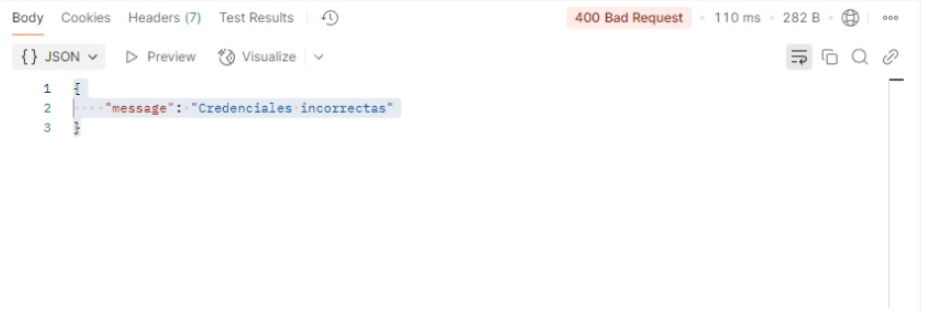
**Autenticación y Permisos**

Para los usuarios, se implementó un *middleware* que verifica la autenticación mediante un token. Este token se genera al iniciar sesión utilizando JWT (JSON Web Tokens). En el caso del administrador, se aplican dos *middlewares*: el primero valida que el usuario esté autenticado, y el segundo verifica su rol. Este último requiere que el usuario tenga un atributo admin para poder acceder a ciertos *endpoints* restringidos.

Los usuarios al tener token pueden tanto comprar productos como dar de alta para su venta, mientras que el administrador puede eliminar productos como usuarios.

Usuarios:

* En el caso de usuario una vez intenta iniciar sesión por medio de su correo, se valida en la base de datos, que este exista, posteriormente se valida la contraseña, si la contraseña es correcta y el usuario también entonces se genera un token por medio de jwt, este mismo token es necesario para el endpoint del perfil y los demás endpoint.
* En caso de no contar con el token:



Administrador:

* En el caso del administrador es muy parecido al del usuario, pero además del token, se valida que tenga un rol de administrador para poder entrar a los demás endpoints.
* Usuario sin rol de admin:

