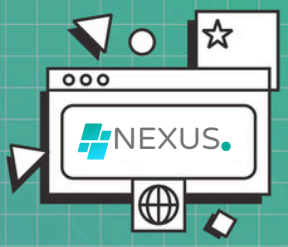
ITBA Escuela de INNOVACIÓN
EVOLUCIÓN CONTINUA



Consigna final | Sprint 7 y 8

Lanzamiento del E-commerce "Mueblería Hermanos Jota"

Plazo de Entrega:
Finalización del Sprint 8

Resumen del Proyecto

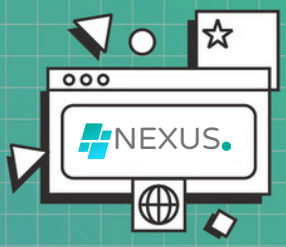
Hemos llegado al sprint final. El objetivo de esta última entrega es transformar nuestra aplicación en una plataforma de e-commerce completa, segura y funcional, lista para el mundo real. Implementaremos el ciclo de vida completo de un usuario: registro, inicio de sesión y acceso a rutas protegidas. Gestionaremos el estado global de la aplicación de forma profesional con la Context API de React y, finalmente, desplegaremos toda la aplicación en la nube para que sea públicamente accesible.

Al finalizar, "Mueblería Hermanos Jota" no será solo un proyecto en tu computadora, sino un producto web vivo y funcional.

Objetivos de Aprendizaje

Al completar este proyecto final, cada estudiante deberá ser capaz de:

ITBA Escuela de INNOVACIÓN
EVOLUCIÓN CONTINUA



Consigna final | **Sprint 7 y 8**

- Implementar un sistema de autenticación full stack seguro usando hashing (bcrypt) y JSON Web Tokens (JWT).
- Proteger rutas de API en el backend con middleware de autorización.
- Gestionar estado global complejo en el frontend (autenticación, carrito de compras) de forma eficiente con la React Context API.
- Crear y gestionar rutas protegidas en el frontend con React Router.
- Realizar un despliegue completo de una aplicación MERN, configurando el frontend, el backend y la base de datos en entornos de producción.
- Configurar variables de entorno para producción en plataformas de hosting.
- Consolidar todas las habilidades del stack MERN en un único proyecto cohesivo.

Requisitos Funcionales y de Negocio

La aplicación debe cumplir con las siguientes funcionalidades para ser considerada completa:

1. Flujo de Autenticación de Usuarios Completo:

"Queremos que nuestros clientes puedan crear una

cuenta y acceder a ella de forma segura. Su información es lo más importante para nosotros."

- Registro: Una página de registro (/registro) con un formulario que envíe los datos a la API. La contraseña debe ser hasheada en el backend antes de guardarse en MongoDB.
- Login: Una página de login (/login) que valide las credenciales del usuario contra la base de datos. Si son correctas, el backend debe generar y devolver un JWT.
- Persistencia de Sesión: El frontend debe almacenar el JWT de forma segura y mantener al usuario autenticado a través de recargas de página.

2. Experiencia de Usuario Dinámica y Segura:

"El sitio debe sentirse inteligente. Debe saber quién soy y mostrarme las opciones relevantes. Las áreas privadas, como mi perfil, deben ser solo para mí."

- UI Condicional: La barra de navegación y otras partes de la UI deben cambiar si el usuario está logueado o no (ej: mostrar "Mi Perfil" y "Logout" en lugar de "Login").
- Logout: La funcionalidad de "Logout" debe

limpiar el token del cliente y actualizar el estado global de la aplicación.

- Rutas Protegidas: Rutas como /perfil o /mis-pedidos deben ser inaccesibles para usuarios no autenticados. Cualquier intento de acceso debe redirigir a la página de login.

3. Carrito de Compras y Proceso de Pedido Funcional:

"El carrito de compras debe funcionar a la perfección. Cuando un cliente decide comprar, el proceso debe ser simple y registrar el pedido a su nombre."

- Gestión con Context API: El estado del carrito (ítems y cantidades) debe ser gestionado globalmente con la Context API de React para ser accesible desde cualquier componente.
- Creación de Pedidos (Protegido): El botón "Finalizar Compra" solo debe funcionar para usuarios logueados. La petición POST al backend para crear un pedido debe estar protegida y usar el JWT para identificar al usuario y asociar el pedido con su cuenta en

la base de datos.

- Limpieza del Carrito: Después de un pedido exitoso, el carrito de compras debe vaciarse automáticamente.

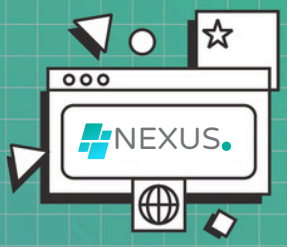
4. Despliegue Público:

"¡El mundo debe ver nuestra mueblería! El sitio final debe estar online y accesible a través de una URL pública."

- La aplicación completa (frontend, backend y conexión a la base de datos) debe estar desplegada y funcionando en internet.

Requisitos Técnicos

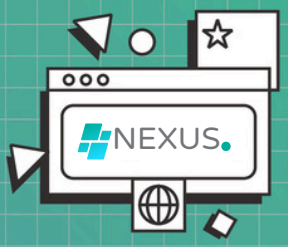
- Backend:
  - Las contraseñas deben ser hasheadas con bcrypt.
  - La autenticación debe gestionarse con JWT.
  - Los endpoints que requieren que un usuario esté logueado (ej: crear un pedido, ver un perfil) deben estar protegidos por un middleware que verifique el JWT.
- Frontend:
  - El estado de autenticación (usuario, token, flag isAuthenticated) y el estado del carrito



deben gestionarse con la React Context API.
    - Las rutas protegidas deben implementarse con un componente <ProtectedRoute> que verifique el estado de autenticación del contexto.
    - El JWT recibido del backend debe enviarse en el encabezado Authorization: Bearer <token> en todas las peticiones a rutas protegidas.
- Despliegue:
    - Backend (API Express): Desplegado en Render.
    - Frontend (App React): Desplegado en Vercel o Netlify.
    - Base de Datos: Cluster de MongoDB Atlas.
    - Todas las claves secretas (URI de la base de datos, secreto de JWT) deben estar configuradas como variables de entorno en las plataformas de hosting, NO hardcodeadas.

Entregables Finales

1. El enlace al repositorio de GitHub con el código fuente final de las carpetas /client y /backend.



Consigna final | **Sprint 7 y 8**

2. El enlace a la aplicación frontend desplegada y pública (ej: en Vercel).
3. El enlace a la API backend desplegada y pública (ej: en Render).
4. Un archivo README.md final y pulido, que sirva como documentación del proyecto, incluyendo los enlaces a los sitios desplegados y una breve descripción de la arquitectura.