

RUBRICA EXPERIENCIA 3 - ET

DESCRIPCIÓN DEL PROYECTO

En esta tercera etapa, continuamos trabajando en la preparación de nuestra tienda online y optimizando procesos ya elaborados de la experiencia 2.

Objetivos de Aprendizaje

- Implementar servicios REST con Spring Boot
 - Desarrollar interfaces de usuario con React
 - Integrar frontend y backend
 - Implementar testing en FrontEnd
 - Implementar carrito de compra
-

REQUERIMIENTOS FUNCIONALES

- ❖ Implementar Login con JWT:
 - Generación de token: JWT implementado correctamente con payload apropiado, expiración configurada, secret seguro.
 - Autenticación: Validación completa del token.
 - Manejo de sesión: manejo de logout, refresh de token, expiración manejada correctamente.
 - Protección de rutas: rutas protegidas correctamente, verificación de permisos.
 - ❖ Hashing de Contraseñas:
 - Algoritmo utilizado: Bycrypt apropiado con salt rounds apropiados (10-12) o Argon2.
 - Implementación: código limpio, async/await, manejo de excepciones.
 - Verificación: comparación segura de contraseñas, protección contra timings attacks.
 - ❖ Perfil Cliente: Carrito de compra
 - Modelo de datos: perfil completo con todos los campos necesarios. Relaciones bien definidas, gestión de carrito.
 - Funcionalidades CRUD: implementación de peticiones correctamente, validaciones apropiadas.
 - Integración con carrito: enlace entre usuario, carrito y pedidos. Persistencia de datos.
 - Interfaz de usuario: UI intuitiva, responsive, completa para gestionar perfil.
 - Seguridad: Validación de datos, protección de información sensible.
 - ❖ Emisión y generación de boleta:
 - Estructura de boleta: todos los campos requeridos y solicitados. Numeración correlativa.
 - Cálculos: montos calculados correctamente, incluyendo IVA.
 - Persistencia: almacenamiento de boleta y detalle de boleta.
 - Integración: generación automática al finalizar la compra.
 - Historial de compras: el cliente podrá tener acceso a las compras realizadas.
 - ❖ Testing en React:
 - Cobertura de tests: implementar 5 pruebas unitarias/integración con RTL / Jest.
 - Configuración de testing: configurar correctamente con Jest/Vitest, scripts de testing, CI/CD opcional.
 - ❖ Dashboard Admin:
 - Acceso a listado de compras realizadas por parte de los clientes.
-

ENTREGABLES: enviar por AVA c/copia al correo electrónico.

1. Github:
 - Url Repositorio FrontEnd.
 - Instrucciones de instalación
 - Instrucciones de ejecución
 - Url Repositorio Backend.
 - README con:
 - Instrucciones de instalación
 - Instrucciones de ejecución
 - Credenciales de prueba
 - Documentación de API (Swagger/Postman collection)
2. Archivo comprimido Backend.
3. Archivo comprimido FrontEnd.
4. Documento ERS en formato pdf.
5. Video Tutorial de la aplicación.
6. Documento de Testing.
7. Base de Datos
 - Script SQL de creación de tablas
 - Script con datos de prueba (seed data)

Puntaje:

- ❖ Implementar Login con JWT: 20 pts.
- ❖ Hashing de contraseñas: 10 pts.
- ❖ Perfil cliente – carrito de compra: 20 pts.
- ❖ Emisión de boleta: 25 pts.
- ❖ Testing en React: 20 pts.
- ❖ Envío - Entregables: 10 pts.
- ❖ Documento Testing: 10 pts.
- ❖ Video tutorial: 15 pts.

Criterios adicionales

Bonificación

- Implementación de paginación en listados (+2 pts)
- Backend en Cloud (+10 pt)

Fechas a considerar:

- Fecha de Entrega y evaluación: semana del 24 de noviembre de acuerdo a su fecha por sección.

Anexos:

Estructura de video tutorial:

- Introducción: 30-45 segundos
- Registro – Login: 2-3 minutos
- Navegación catálogo de productos: 1-2 minutos
- Perfil Admin:
 - a. Gestión de usuarios: 2-3 minutos.
 - b. Gestión de productos: 2-3 minutos.
 - c. Compras: 1-2 minutos
- Carrito de compras: 2-3 minutos
- Generación de boleta: 1-2 minutos
- Historial de compras: 1 minuto.
- Funciones de seguridad: 30 segundos.
- Otros puntos: 30 segundos.
- Cierre y conclusión: 30 segundos.

Estructura del documento testing:

- Introducción y propósito.
- Herramientas y configuración.
- Plan de testing
 - Componentes a testear: cada equipo seleccionara 5 componentes/pages a testear
 - Ejemplos de código
 - Resultados de los testing (consola, interfaz gráfica).
- Conclusión