

## Historias Técnicas

### Historia Técnica 1

#### 1. Título:

Configuración de Autenticación JWT

#### 2. Descripción:

Implementar y configurar autenticación basada en JWT en el backend Node.js, para permitir el control seguro de sesiones de usuarios autenticados. Esto asegura que cada petición posterior al login esté validada y protegida.

#### 3. Criterios de Aceptación:

- Se genera un token JWT al iniciar sesión exitosamente.
- El token contiene el id, rol y expiración.
- Las rutas protegidas validan el token antes de procesar solicitudes.
- El token expira correctamente tras el tiempo definido.

#### 4. Tareas:

- Instalar librería jsonwebtoken.
- Crear middleware de verificación.
- Proteger rutas con middleware.
- Añadir expiración configurable.

#### 5. Pruebas:

- Unitarias para generación y validación de tokens.
- Pruebas de integración para rutas protegidas.

#### 6. Dependencias:

- **Historia técnica:** Diseño de modelo de usuarios.
- **Historia de usuario:** HU-AD-01 (inicio de sesión y aprobación).

#### 7. Estimación de Esfuerzo:

3 puntos

#### 8. Prioridad:

Alta

## **Historia Técnica 2**

### **1. Título:**

Integración con Oracle Database

### **2. Descripción:**

Establecer conexión entre el backend Node.js y Oracle Database utilizando controladores compatibles (oracledb). Esto es esencial para manejar el inventario y pedidos integrados con el ERP actual.

### **3. Criterios de Aceptación:**

- Se establece conexión estable y segura.
- Se ejecuta al menos una consulta de prueba de lectura/escritura.
- Manejo de errores implementado.
- La conexión soporta consultas asincrónicas.

### **4. Tareas:**

- Instalar e importar el driver oracledb.
- Configurar parámetros de conexión.
- Crear archivo de conexión reutilizable.
- Ejecutar pruebas de lectura/escritura.

### **5. Pruebas:**

- Unitarias para conexión.
- De integración con scripts de prueba.

### **6. Dependencias:**

- Acceso a credenciales del ERP y esquema.
- Historia técnica: Diseño de repositorios y modelos de datos.

### **7. Estimación de Esfuerzo:**

4 puntos

### **8. Prioridad:**

Alta

## **Historia Técnica 3**

### **1. Título:**

Diseño e implementación de API RESTful

### **2. Descripción:**

Diseñar la estructura inicial de la API siguiendo el patrón REST para exponer los servicios requeridos por el frontend, chatbot y sistema ERP. Debe incluir rutas base para usuarios, pedidos, entregas y reportes.

### **3. Criterios de Aceptación:**

- La API sigue buenas prácticas REST (verbos HTTP correctos).
- Rutas: /usuarios, /pedidos, /entregas, /reportes.
- La documentación de rutas está disponible (Postman o Swagger).
- Gestión de errores y estados HTTP adecuada.

### **4. Tareas:**

- Crear estructura modular de Express.
- Implementar controladores iniciales.
- Documentar endpoints.
- Añadir middlewares globales (logs, CORS, bodyParser).

### **5. Pruebas:**

- Postman para pruebas manuales.
- Unitarias para controladores básicos.

### **6. Dependencias:**

- Historia técnica: Middleware JWT.
- Historia técnica: Conexión a base de datos.

### **7. Estimación de Esfuerzo:**

5 puntos

### **8. Prioridad:**

Alta

## **Historia Técnica 4**

### **1. Título:**

Implementación de Validaciones en Backend

### **2. Descripción:**

Desarrollar validaciones en backend para asegurar integridad de datos ingresados por usuarios, choferes y administradores. Uso de middleware como express-validator para validar formularios.

### **3. Criterios de Aceptación:**

- Se validan campos obligatorios y formatos (correo, contraseña, etc.).
- Se evitan registros duplicados (correo, placa de camión).
- El sistema devuelve mensajes claros de error.

### **4. Tareas:**

- Instalar express-validator.
- Crear validadores para cada entidad.
- Añadir sanitización de entradas.
- Manejo de errores centralizado.

### **5. Pruebas:**

- Unitarias para cada validación.
- Casos con datos maliciosos (XSS, SQL Injection).

### **6. Dependencias:**

- Historia técnica: API RESTful
- Historia de usuario: HU-AD-05, HU-CL-01

### **7. Estimación de Esfuerzo:**

3 puntos

### **8. Prioridad:**

Media

## Historia Técnica 5

### 1. Título:

Integración del chatbot con WhatsApp Business API

**2. Descripción** Integrar el backend del sistema con la API oficial de WhatsApp Business (por Twilio o Zenvia) para recibir mensajes de clientes, registrar pedidos y responder con confirmaciones.

### 3. Criterios de Aceptación:

- El chatbot responde a mensajes de texto con flujo guiado.
- Registra pedidos con éxito.
- Devuelve código de seguimiento.
- Se maneja asincronía y errores de red.

### 4. Tareas:

- Configurar cuenta y API Key de proveedor.
- Crear webhook en backend para recibir mensajes.
- Definir flujo de conversación.
- Validar datos y registrar pedidos.

### 5. Pruebas:

- Pruebas manuales con un número de prueba.
- Pruebas de integración de flujos.

### 6. Dependencias:

- Historia técnica: Conexión a Oracle
- Historia técnica: Validaciones de pedidos
- Historia de usuario: HU-CL-01

### 7. Estimación de Esfuerzo:

5 puntos

### 8. Prioridad:

Alta