

# Definición de Requisitos del Backend

## 1. Introducción

El backend del Sistema de Gestión de Inventario (SGI) gestiona la lógica de negocio, el acceso a la base de datos, y proporciona interfaces de programación para la interacción con el frontend y otras partes del sistema. A continuación, se definen las funcionalidades necesarias para el backend, el tipo de datos que manejará, la forma en que se integrará con el frontend y otros componentes, así como las pruebas, documentación que se le harán y la arquitectura que utilizará.

## 2. Funcionalidades del Backend

### 2.1 Gestión de Usuarios

- **Autenticación y autorización:** Implementar un sistema de autenticación seguro que permita a los usuarios iniciar sesión. Los usuarios deben tener roles definidos (administrador, usuario estándar) con diferentes niveles de acceso.
- **Registro de usuarios:** Permitir a los administradores registrar nuevos usuarios en el sistema.
- **Gestión de perfiles:** Funcionalidades para la visualización, edición y eliminación de perfiles de usuario.
- **Gestión de sesiones:** Manejar sesiones de usuario para mantener la seguridad y el estado de la aplicación.

### 2.2 Gestión de Proveedores

- **Alta de proveedores:** Permitir a los administradores agregar nuevos proveedores con detalles como nombre, dirección, contacto, etc.
- **Modificación de proveedores:** Funcionalidades para editar la información de los proveedores existentes.
- **Baja de proveedores:** Permitir la eliminación de proveedores del sistema.
- **Visualización de proveedores:** Permitir la consulta de la lista de proveedores y sus detalles.

### 2.3 Gestión de Insumos

- **Alta de insumos:** Permitir a los administradores agregar nuevos insumos al inventario con detalles como nombre, descripción, cantidad, proveedor, etc.
- **Modificación de insumos:** Funcionalidades para editar la información de los insumos existentes en el inventario.
- **Baja de insumos:** Permitir la eliminación de insumos del inventario.
- **Visualización de Inventario:** Permitir la consulta del inventario, visualizando la lista de insumos y sus detalles.
- **Control de stock:** Implementar funcionalidades para el control y la actualización del stock de insumos en el inventario.

## 2.4 Seguridad

- **Protección contra inyecciones SQL:** Implementar prácticas de codificación segura para prevenir ataques de inyección SQL.
- **Encriptación de datos:** Utilizar técnicas de encriptación para proteger datos sensibles, como contraseñas e información personal.
- **Validación de entradas:** Validar todas las entradas del usuario para prevenir vulnerabilidades y fallos.

## 2.5 Manejo de errores y mensajes de estado

- **Manejo de errores:** Implementar mecanismos para manejar errores y excepciones en el backend, proporcionando mensajes de error claros y útiles.
- **Mensajes de estado:** Proveer mensajes de estado que informen al frontend sobre el éxito o fracaso de las operaciones realizadas.

## 3. Datos a manejar

### 3.1 Usuarios

- **Datos personales:** Nombre, apellido, dirección de correo electrónico, número de teléfono.
- **Credenciales:** Contraseña (encriptada), roles de usuario.

### 3.2 Proveedores

- **Información de contacto:** Nombre, dirección, número de teléfono, correo electrónico, persona de contacto.
- **Detalles adicionales:** Forma de pago, código, cuit y categorías.

### 3.3 Insumos

- **Detalle del insumo:** Nombre, descripción, categoría, precio unitario, fecha de vencimiento
- **Inventario:** Cantidad en stock, cantidad mínima, zona de depósito, unidad de referencia, código de almacenaje.
- **Proveedor Asociado:** Información del proveedor que suministra el producto.

## 4. Integración con Frontend y Otros Componentes

### 4.1 Endpoints y Autenticación

- **Endpoints:** Crear endpoints para todas las funcionalidades del sistema, permitiendo al frontend interactuar con el backend mediante solicitudes HTTP (GET, POST, PUT, DELETE).
- **Autenticación:** Mediante la validación que provee la api de Google se realizará la autenticación para acceder a la app.

### 4.2 Comunicación Frontend-Backend

- **Solicitudes HTTP:** El frontend enviará solicitudes HTTP al backend para realizar operaciones como la gestión de usuarios, proveedores e insumos.

#### 4.3 Integración con la base de datos

- **ORM:** Utilizar PDO(PHP Data Object) para interactuar con la base de datos de manera eficiente y segura, mapeando las estructuras de datos a tablas en la base de datos.
- **Conexiones seguras:** Asegurar que todas las conexiones a la base de datos sean seguras y que los datos estén protegidos contra accesos no autorizados.

### 5. Pruebas y Documentación

#### 5.1 Pruebas

- **Pruebas unitarias:** Realizar pruebas unitarias para verificar que cada componente del backend funcione correctamente.
- **Pruebas de integración:** Asegurar que todas las partes del backend se integren correctamente entre sí y con el frontend.

#### 5.2 Documentación

- **Comentarios de código:** Incluir comentarios claros y descriptivos en el código para facilitar su comprensión y mantenimiento.
- **Documentación adicional:** Proporcionar documentación adicional, como manuales de uso y guías de desarrollo, para apoyar a los desarrolladores y usuarios del sistema.

### 6. Arquitectura

**6.1 Arquitectura MVC:** Uso de la arquitectura MVC el cual permite una aplicación escalable, mantenible y fácil de expandir.

**6.2 Modelo:** El backend que contiene toda la lógica de datos, para el SGI, elegir como lenguaje a PHP, el cual es un lenguaje de programación interpretado del lado del servidor y de uso general que se adapta especialmente al desarrollo web, en conjunto con la utilización de una base de datos relacional.

**6.3 Vista:** El front end o interfaz gráfica de usuario (GUI), utiliza un desarrollo pensado en la web, mediante HTML, CSS, JavaScript. Mencionar la utilización del framework Bootstrap, para facilitar un diseño visual responsivo.

**6.4 Controlador:** El cerebro de la aplicación que controla como se muestran los datos, para el SGI, se mantiene el desarrollo de este componente mediante el lenguaje de programación PHP.

**6.5 Uso del patrón DAO:** Propone separar por completo la lógica de negocio de la lógica para acceder a los datos, de esta forma, el DAO proporcionará los métodos necesarios para insertar, actualizar, borrar y consultar la información.