# Requisitos del Sistema para los ****Servicios Médicos UPTAG****

Este conjunto de módulos está diseñado especialmente para cubrir las necesidades del área de Servicios Médicos de la UPTAG (Universidad Politécnica Territorial Alonso Gamero), donde los insumos médicos se administran internamente y no se comercializan. Cada módulo incluye ejemplos para facilitar su comprensión durante tu entrevista con la comunidad.

## 1. Autenticación y Gestión de Usuarios

**Requisitos Funcionales**

* Registro y edición de usuarios del sistema
* Inicio/cierre de sesión
* Asignación de roles: Administrador, Operador, Auditor
* Restablecimiento de contraseñas:
* Al registrar el usuario, le pides:
  + Nombre
  + Correo
  + Contraseña
  + Pregunta 1 + Respuesta 1
  + Pregunta 2 + Respuesta 2
* Si olvida la contraseña, el sistema le solicita:
  + Correo o nombre de usuario
  + Responder correctamente ambas preguntas
  + Introducir nueva contraseña y confirmarla

**Ventajas:**

* Completamente local
* No depende de emails ni SMS
* Protección suficiente si las preguntas son personalizadas

**Recomendaciones técnicas:**

* Guarda las respuestas encriptadas (Hash::make()), igual que la contraseña
* Ofrece preguntas variadas (familiares, fechas, gustos) para que el usuario elija
* Asegura que el usuario no use respuestas fácilmente adivinables ("123", "admin", etc.

## 2. Gestión de Categorías y Subcategorías

**Requisitos Funcionales**

* Crear y administrar categorías principales (Medicamentos, Material Quirúrgico, Consumibles)
* Crear subcategorías (Tabletas, Jarabes, Inyectables)
* Filtrar productos por categoría y subcategoría

**Ejemplo** El operador registra el insumo “Ibuprofeno 400 mg” en la categoría Medicamentos y subcategoría Tabletas. Luego puede listar todos los insumos tipo “Jarabe” dentro de Medicamentos.

## 3. Gestión de Productos

**Requisitos Funcionales**

* Registro, edición y eliminación de productos
* Asignación de categoría/subcategoría
* Definición de stock mínimo

**Ejemplo** Se crea el producto “Alcohol al 70%”, se asigna a Material Quirúrgico > Líquidos, y de Farmacia Interna con stock mínimo de 10 frascos.

## 4. Inventario / Stock

**Requisitos Funcionales**

* Visualización del stock actual de cada insumo
* Ajuste inicial de inventario según conteo físico
* Alerta automática por stock bajo
* Registro de cambios de stock

**Ejemplo** Después de una revisión, la encargada ajusta el inventario de “Guantes estériles” de 180 a 200 unidades. El sistema ahora mostrará el nuevo stock y marcará en rojo si baja de 50.

## 5. Control de Movimientos

**Requisitos Funcionales**

* Registrar entradas (donaciones, adquisiciones internas)
* Registrar salidas (uso en consultas, emergencias, talleres)
* Validar existencia de stock
* Consultas por fecha, producto o usuario

**Ejemplo** Se registra la entrada de 300 jeringas vía presupuesto institucional. Luego, se realiza una salida de 25 jeringas usadas durante una jornada médica en campus UPTAG.

## 6. Reportes y Análisis

**Requisitos Funcionales**

* Listado de insumos disponibles
* Informe de productos con stock crítico
* Gráficas de consumo mensual
* Exportación en PDF o Excel

**Ejemplo** El administrador genera un reporte PDF del consumo mensual de “Paracetamol” y detecta que en julio hubo un uso inusualmente alto en Consulta Externa.

## 7. Notificaciones y Alertas

**Requisitos Funcionales**

* Aviso automático por stock mínimo alcanzado
* Envío de correo interno institucional
* Registro de alertas activas y resueltas

**Ejemplo** Cuando el stock de “Alcohol” llega a 10 unidades, el sistema envía un correo a la Coordinación para que se planifique la reposición.

## 8. Configuración y Parámetros

**Requisitos Funcionales**

* Definición de bodegas internas (Farmacia UPTAG, Almacén General, Consulta Externa)
* Gestión de proveedores institucionales
* Configuración de unidades de medida (unidad, frasco, caja)
* Códigos internos para cada producto (códigos UPTAG)
* Gestión de fechas de vencimiento

**Ejemplo** Se crea la bodega “Farmacia Servicios Médicos” y se asigna el insumo “Yodo” al estante A. Se define alerta 30 días antes de su fecha de vencimiento. El producto se codifica como **UPTAG-FAR-0012**.

## 9. Auditoría y Seguridad

**Requisitos Funcionales**

* Registro histórico de cambios en productos, stock y movimientos
* Monitoreo de accesos y fallos de autenticación
* Control de acceso a funciones sensibles

**Ejemplo** La auditora revisa que el operador “EnfermeroLuis” realizó 12 salidas de insumos entre el 5 y el 10 de julio, todas desde Farmacia UPTAG. Se consulta el log para confirmar que ninguna salida superó el stock disponible.

## MEJORAS PROFESIONALES DEL SISTEMA

## 🔐 1. Autenticación y Gestión de Usuarios

**Mejora profesional:**

* Implementa Laravel Breeze o Fortify para autenticación rápida y segura.
* Agrega **validaciones** sólidas en formularios de usuario.
* Usa **middleware por roles** para restringir acceso según perfil.

**Extra que agradará:**

* Activar y desactivar usuarios sin eliminarlos.
* Mostrar fecha del último acceso para control interno.

## 📁 2. Gestión de Categorías y Subcategorías

**Mejora profesional:**

* Usa modelos con relaciones bien definidas (hasMany, belongsTo).
* Haz validaciones de unicidad y longitud para evitar nombres duplicados.
* Permite ordenar categorías/subcategorías por nombre o prioridad.

**Extra que agradará:**

* Filtrar productos desde el dashboard por categoría/subcategoría.
* Añadir íconos o colores según tipo para visualización rápida.

## 💊 3. Gestión de Productos

**Mejora profesional:**

* Usa **Form Requests** para validación modular.
* Crear un componente reutilizable para formularios (Blade o Livewire).
* Indexar el campo nombre en DB para búsquedas rápidas.

**Extra que agradará:**

* Mostrar estado del producto: Activo/Inactivo/Caducado.
* Adjuntar ficha técnica o imagen del insumo si aplica.

## 📦 4. Inventario / Stock

**Mejora profesional:**

* Utilizar transacciones para garantizar consistencia en ajustes.
* Registrar quién hizo cada cambio de stock.
* Mostrar cambios recientes en un panel tipo “últimas acciones”.

**Extra que agradará:**

* Dashboard con semáforo de stock (Verde = Ok, Rojo = Crítico).
* Opción de imprimir listado físico de inventario.

## 🔄 5. Control de Movimientos

**Mejora profesional:**

* Crear un MovementService que gestione entradas/salidas.
* Validar que cantidad no sea negativa ni exceda el disponible.
* Agrupar movimientos por día/mes para mejorar el rendimiento.

**Extra que agradará:**

* Historial completo de cada producto (como una “ficha clínica”).
* Campo para **motivo** del movimiento (uso en emergencia, práctica, etc.).

## 📊 6. Reportes y Análisis

**Mejora profesional:**

* Usa Laravel Excel para exportar datos fácilmente.
* Implementa filtros dinámicos por fecha, categoría o ubicación.
* Visualiza gráficas con Chart.js o Laravel Charts.

**Extra que agradará:**

* Generar reportes automáticos semanales.
* Enviar resumen por correo al finalizar cada mes.

## 🔔 7. Notificaciones y Alertas

**Mejora profesional:**

* Crear un NotificationService para centralizar alertas.
* Permitir activar/desactivar alertas por insumo.
* Enviar alertas en tiempo real (WebSocket/Livewire si te animas).

**Extra que agradará:**

* Visualizar alertas en un panel dedicado.
* Escalar la alerta si no se atiende en 3 días (recordatorio automático).

## ⚙️ 8. Configuración y Parámetros

**Mejora profesional:**

* Usa una tabla settings para parámetros globales editables.
* Crea sección de gestión de proveedores con contacto y dirección.
* Validar fechas de vencimiento al ingresar productos.

**Extra que agradará:**

* Mostrar próximos vencimientos en el dashboard.
* Permitir crear formatos de código como “UPTAG-FARM-0001”.

## 🕵️ 9. Auditoría y Seguridad

**Mejora profesional:**

* Implementa Laravel Auditing o crea tu propia tabla de logs.
* Registrar IP, navegador y ubicación aproximada.
* Mostrar historial de sesión de cada usuario.

**Extra que agradará:**

* Exportar log de auditoría para respaldos.
* Enlace directo a ver detalle de cambios por usuario o por producto.

## 💡 Bonus: Ideas para Comenzar con Laravel

* Usa comandos Artisan como php artisan make:model con -mcr para generar modelo, migración, controlador y recurso juntos.
* Comienza con Bootstrap para el diseño hasta que te sientas listo para Tailwind.
* Crea un proyecto sencillo de ejemplo antes de integrar todos los módulos.