



**ESPE**  
UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE  
CIENCIAS DE LA COMPUTACIÓN  
INGENIERÍA EN  
TECNOLOGÍAS DE LA INFORMACIÓN**

**“HISTORIAS DE USUARIO”**

**Integrantes:**

**Tutor:**  
Ing. Jenny Ruiz

Historias de usuario para el cargo de utilización del sistema.

Historia de Usuario	
Número: 1	Usuario: Administrador
<b>Nombre Historia:</b> Roles Administrativos	
Prioridad en negocio: Alta	Riesgo en desarrollo: alta
<b>Iteración Asignada:</b> 1	
<b>Programador Responsable:</b> Cesar Herrera	
<b>Descripción:</b> <ul style="list-style-type: none"><li>El programa deberá implementar un sistema de reconocimiento exclusivo para administradores, con gestión de perfiles que pueda habilitar, crear o desactivar usuarios según los permisos asignados por el administrador en jefe, garantizando la seguridad y control de accesos dentro del sistema de inventario, evitando manipulaciones no autorizadas y mejorando la gestión administrativa por niveles.</li></ul>	
<b>Validación:</b> <ul style="list-style-type: none"><li>Verificar que solo administradores con credenciales válidas pueden acceder al sistema. Probar inicio de sesión con usuario y contraseña correctos, incorrectos y con un usuario deshabilitado (debe negar el acceso).</li><li>Comprobar que el administrador general puede crear, editar y desactivar perfiles, y que dichas acciones quedan registradas en un historial (usuario, fecha, hora y acción).</li><li>Confirmar que un usuario desactivado no puede volver a acceder ni realizar operaciones en el sistema.</li></ul>	

Historia de Usuario	
Número: 2	Usuario: Administrador
<b>Nombre Historia:</b> Categorización de productos en el inventario	
Prioridad en negocio: Media	Riesgo en desarrollo: alta
<b>Iteración Asignada:</b> 1	
<b>Programador Responsable:</b> Matias Intríago	
<b>Descripción:</b> <ul style="list-style-type: none"><li>El programa deberá organizar los productos mediante categorías claras y filtros de búsqueda avanzados, permitiendo al administrador crear, asignar y modificar categorías, así como registrar nuevos productos, actualizar los existentes y aplicar filtros basados en nombre, categoría, precio, lote o disponibilidad para mejorar la velocidad, precisión y eficiencia en la gestión y localización de productos dentro del inventario.</li></ul>	
<b>Validación:</b>	

- Comprobar que el sistema permita insertar, modificar, buscar y eliminar productos en la base de datos de manera correcta.
- Evaluar que cada producto quede adecuadamente asociado a una categoría válida y que la organización siga los criterios establecidos (categoría, nombre, precio, stock).
- Realizar pruebas de búsqueda y filtrado utilizando diferentes combinaciones de datos y distintos volúmenes de información para validar el rendimiento del sistema.
- Confirmar con el administrador que la categorización y los filtros cumplen con los requerimientos funcionales establecidos.

<b>Historia de Usuario</b>	
Número: 3	Usuario: Administrador
<b>Nombre Historia:</b> Control de caducidad y alertas de stock	
Prioridad en negocio: Baja	Riesgo en desarrollo: alta
<b>Iteración Asignada:</b> 1	
<b>Programador Responsable:</b> Kelly Montalvo	
<b>Descripción:</b>	
<ul style="list-style-type: none"> <li>● El programa deberá registrar las fechas de caducidad y niveles mínimos de stock para cada medicina, comparando estos datos con la información actual del inventario para identificar productos próximos a vencer (dentro de 30 días) o con existencias por debajo del nivel permitido, generando alertas visuales o mensajes que informen al administrador para evitar pérdidas por productos vencidos, asegurar la disponibilidad adecuada y mejorar la gestión del inventario.</li> </ul>	
<b>Validación:</b>	
<ul style="list-style-type: none"> <li>● Registrar productos con fechas de caducidad próximas y niveles de stock inferiores al mínimo establecido, comprobando que el sistema muestre correctamente las alertas visuales o mensajes correspondientes.</li> <li>● Evaluar que, al actualizar la fecha de caducidad o la cantidad disponible, las alertas se ajusten automáticamente según los nuevos valores.</li> <li>● Validar con el administrador que las alertas cumplen con los requerimientos funcionales y permiten gestionar los productos de forma oportuna.</li> </ul>	

<b>Historia de Usuario</b>	
Número: 4	Usuario: Administrador
<b>Nombre Historia:</b> Persistencia de medicinas en MongoDB	
Prioridad en negocio: Alta	Riesgo en desarrollo: alta
<b>Iteración Asignada:</b> 2	
<b>Programador Responsable:</b> Cesar Herrera	

**Descripción:**

- El programa deberá guardar permanentemente las medicinas registradas en una base de datos MongoDB mediante la creación de un cluster, configuración de conexión con pymongo, creación de colección con validaciones básicas, y modificación de las funciones de registro, búsqueda y listado para que interactúen con MongoDB, incluyendo manejo de errores y respaldo básico para mantener los datos aunque se cierre el sistema y dejar listo el soporte para reportes a futuro.

**Validación:**

- Registrar 2 medicinas, cerrar el programa, abrir de nuevo y verificar que siguen apareciendo en el sistema.
- Validar en MongoDB Compass o Atlas que los documentos existen y contienen la información correcta.
- Probar el manejo de errores con casos de conexión fallida o datos inválidos.

Historia de Usuario	
Número: 5	Usuario: Administrador
Nombre Historia: Reporte de medicinas por rango de fecha	
Prioridad en negocio: Media	Riesgo en desarrollo: alta
Iteración Asignada: 2	
Programador Responsable: Matias Intriago	
<b>Descripción:</b>	
<ul style="list-style-type: none"><li>● El programa deberá generar un reporte con todos los datos de la base de datos, filtrable desde una fecha seleccionada mediante un calendario, mostrando un resumen (cantidad de medicinas, stock bajo y próximas a caducar) y una tabla con el detalle completo (nombre, categoría, stock, caducidad, lote), con opción de descarga en PDF o Excel, para obtener un resumen del historial de medicinas y facilitar el control y toma de decisiones.</li></ul>	
<b>Validación:</b>	
<ul style="list-style-type: none"><li>● Seleccionar una fecha "desde" y generar reporte, verificando que solo aparezcan registros desde esa fecha en adelante.</li><li>● Probar con tres casos: (1) hay datos disponibles, (2) no hay datos para el rango seleccionado, (3) fecha inválida o vacía.</li><li>● Validar que los datos del reporte coincidan con la consulta en MongoDB Compass o Atlas.</li><li>● Verificar que la descarga en PDF o Excel funcione correctamente y contenga la información completa.</li></ul>	

Historia de Usuario	
Número: 6	Usuario: Administrador
Nombre Historia: Historia de movimientos de inventario	

<b>Prioridad en negocio: Media</b>	<b>Riesgo en desarrollo: alta</b>
<b>Iteración Asignada: 3</b>	
<b>Programador Responsable: Kelly Montalvo</b>	
<b>Descripción:</b>	
<ul style="list-style-type: none"> <li>El programa deberá registrar automáticamente todos los movimientos realizados en el inventario (altas, bajas y modificaciones) en una colección "historial_movimientos" en MongoDB, guardando fecha, hora, usuario responsable y datos modificados, con una interfaz para consultar el historial filtrando por fecha, tipo de acción o medicina específica, para tener control completo y trazabilidad de todos los cambios, permitiendo realizar auditorías, detectar errores y mantener un seguimiento histórico confiable.</li> </ul>	
<b>Validación:</b>	
<ul style="list-style-type: none"> <li>Registrar una medicina nueva y verificar que aparezca en el historial como "ALTA" con fecha, hora y usuario.</li> <li>Editar una medicina existente (cambiar stock o fecha de caducidad) y confirmar que se registre como "MODIFICACIÓN" mostrando datos anteriores y nuevos.</li> <li>Eliminar una medicina y verificar que aparezca como "BAJA" en el historial.</li> <li>Consultar el historial filtrando por fecha específica, por tipo de acción y por nombre de medicina, confirmando que los filtros funcionen correctamente.</li> <li>Validar en MongoDB Compass que todos los movimientos están siendo almacenados con la información completa.</li> </ul>	