Documento de ARQUITECTURA DE SOFTWARE

KIBEN

# Historial de Versiones

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Fecha | Versión | Autor | Organización | Descripción |
| 10.10.25 | 1.0 | Sergio Moreno y Vicente Quintana | M&Q Technology | Creación del documento |
| 29.10.25 | 1.1 | Sergio Moreno y Vicente Quintana | M&Q Technology | Modificación de SIFA a KIBEN |

# Información del Proyecto

|  |  |
| --- | --- |
| Empresa / Organización | M&Q Technology |
| Proyecto | Kiben |
| Fecha de preparación | Octubre 2025 |
| Cliente | Hogares de ancianos |
| Patrocinador principal | M&Q Technology |
| Gerente / Líder de Proyecto | Sergio Felipe Moreno Cepeda |
| Gerente / Líder de Análisis de negocio y requerimientos | Sergio Felipe Moreno Cepeda |

# Aprobaciones

|  |  |  |
| --- | --- | --- |
| Nombre y Apellido | Fecha | Firma |
| Fernando Herrera | 29.10.25 |  |
| Sergio Moreno | 29.10.25 |  |
| Vicente Quintana | 29.10.25 |  |

# Introducción

## 1.1 Propósito

Este Documento de Arquitectura de Software (DAS) describe la arquitectura del Sistema KIBEN utilizando el modelo de arquitectura 4+1 vistas de Kruchten.

## 1.2 Alcance del Documento

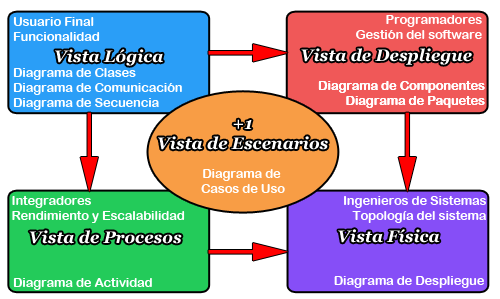
Este DAS cubre la arquitectura completa del sistema KIBEN:

* Aplicación Web (Django + Bootstrap)
* Aplicación Móvil (Android)
* Aplicación de Escritorio (Electron)
* Backend API REST (Django REST Framework)
* Base de Datos (SQLite/PostgreSQL)
* Integraciones externas (Telegram Bot, API Medicamentos Cima)

# Representación arquitectónica

## 2.1 Modelo arquitectónico 4+1 vistas de Philippe Kruchten

KIBEN adopta el modelo de arquitectura 4+1 vistas.



# Objetivos y restricciones

## 3.1 Objetivos arquitectónicos

## Control acceso basado en roles (RBAC)

## Auditoría completa acciones críticas

## Interfaz intuitiva para personal no técnico

## Diseño responsive (desktop/tablet/móvil)

## Satisfacción usuarios ≥4/5

## Respuesta rápida para operaciones críticas

## Sistema accesible 24/7

## Backup automático diario

## Monitoring con alertas

## Multiplataforma (web/móvil/escritorio)

## API REST permite múltiples clientes

## 3.2 Restricciones Arquitectónicas

|  |  |  |
| --- | --- | --- |
| Hito | Fecha | Impacto |
| Desvinculación cliente | 10 oct 2025 | Pérdida 7 semanas (SIFA) |
| Inicio KIBEN | 13 oct 2025 | Solo 8 semanas restantes |
| Fase programación | 3-21 nov | Solo 3 semanas desarrollo |
| Fase pruebas | 21 nov-4 dic | Solo 2 semanas testing |
| Presentación final | 5 dic 2025 | Fecha inamovible |

# Vista lógica

## 4.1 Descripción General

# La vista lógica describe la estructura estática del sistema en términos de clases y objetos.

# Incluye:

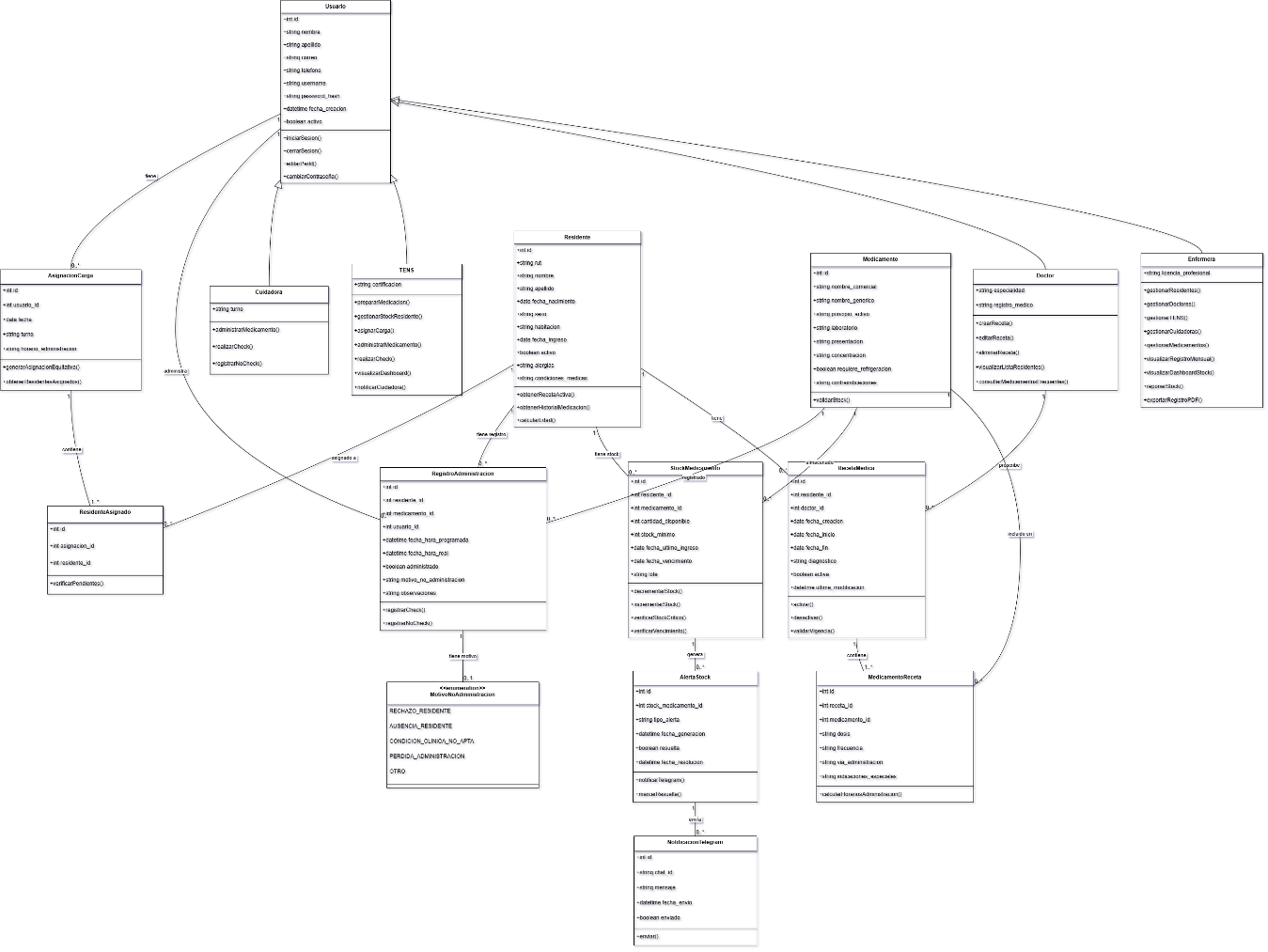
# Entidades del dominio

# Jerarquía de usuarios con herencia

# Relaciones entre entidades

# Métodos principales

## 4.2 Diagrama de Clases Completo

****

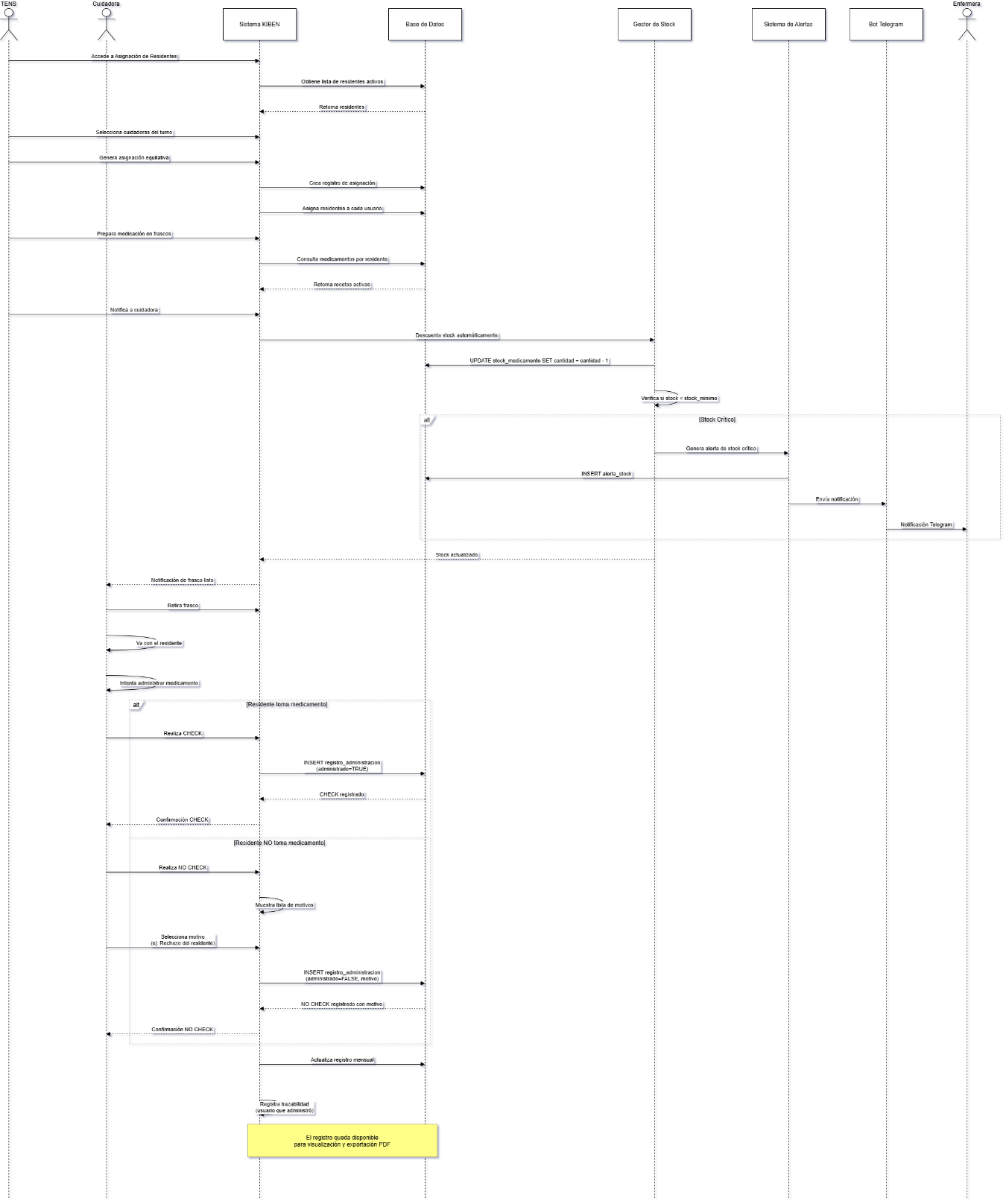
# Modelo completo de clases con herencia de Usuario (Enfermera, Doctor, TENS, Cuidadora)

# Todas las entidades del dominio: Residente, RecetaMedica, Medicamento, StockMedicamento, RegistroAdministracion, etc.

# Relaciones entre clases con multiplicidades

# Métodos principales de cada clase

## 4.3 Diagrama de Secuencia - Administración de Medicamento



Flujo:

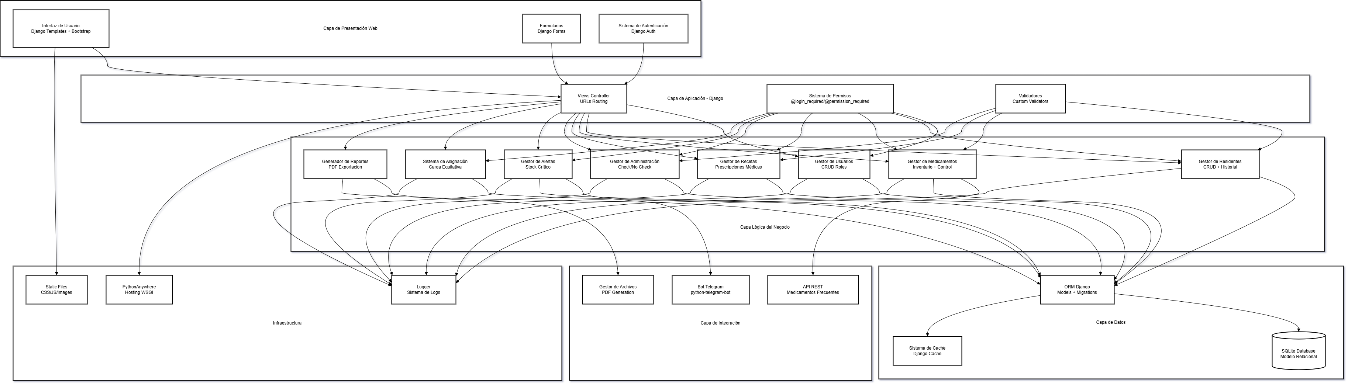
1. Actor (TENS/Cuidadora) inicia sesión
2. Sistema carga asignaciones del turno
3. Actor selecciona residente
4. Actor visualiza medicamentos pendientes
5. Actor administra medicamento físicamente
6. Actor registra resultado:
   1. CHECK: Sistema descuenta stock, marca ADMINISTRADO
   2. NO CHECK: Sistema solicita motivo, NO descuenta stock
7. Sistema genera trazabilidad completa
8. Sistema evalúa alertas (si retraso >1h)
9. Sistema actualiza dashboard tiempo real

# Vista de despliegue

## 5.1 Descripción General

Vista de despliegue muestra la organización del sistema en módulos de software, bibliotecas y componentes.

## 5.2 Diagrama de Componentes



Incluye:

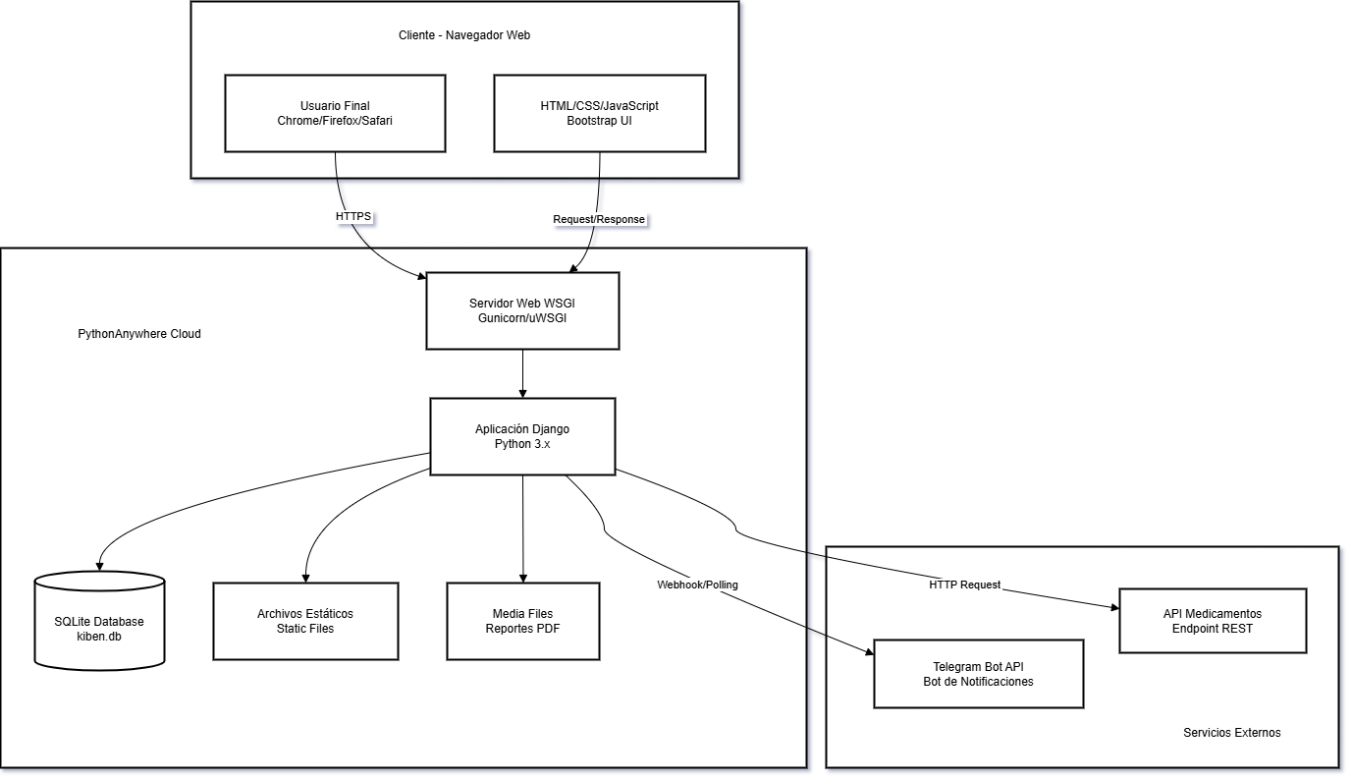
* Capa de Presentación (Templates, Forms, Auth)
* Capa de Aplicación (Views, Permisos, Validadores)
* Capa de Lógica de Negocio (8 gestores principales)
* Capa de Integración (API REST, Bot Telegram, PDF)
* Capa de Datos (SQLite, ORM Django, Caché)
* Infraestructura (PythonAnywhere, Static Files, Logger)

# Vista física

## 6.1 Descripción General

La vista física describe el mapeo del software sobre hardware y distribución física de componentes.

## 6.2 Diagrama de Despliegue



Incluye:

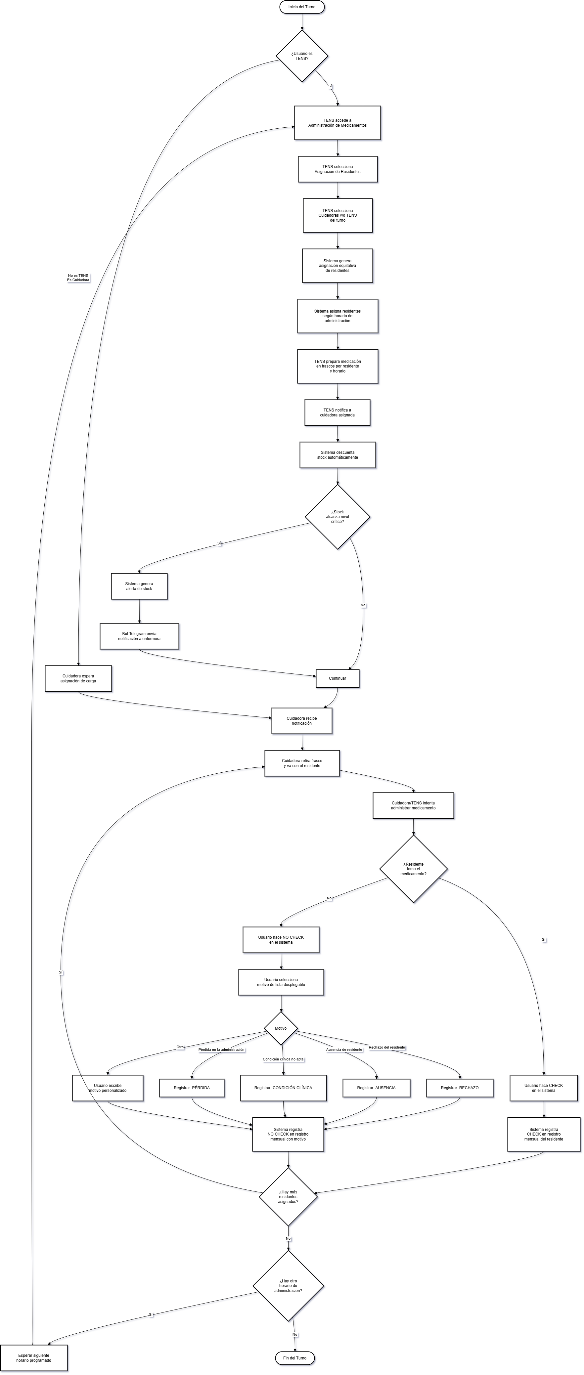
* Arquitectura desplegada en PythonAnywhere Cloud
* Interacción con servicios externos (Telegram Bot API, API Medicamentos)
* Flujo de comunicación HTTPS entre cliente y servidor

# Vista de procesos

## 7.1 Descripción General

La vista de procesos describe aspectos dinámicos del sistema, mostrando flujos de trabajo y procesos críticos.

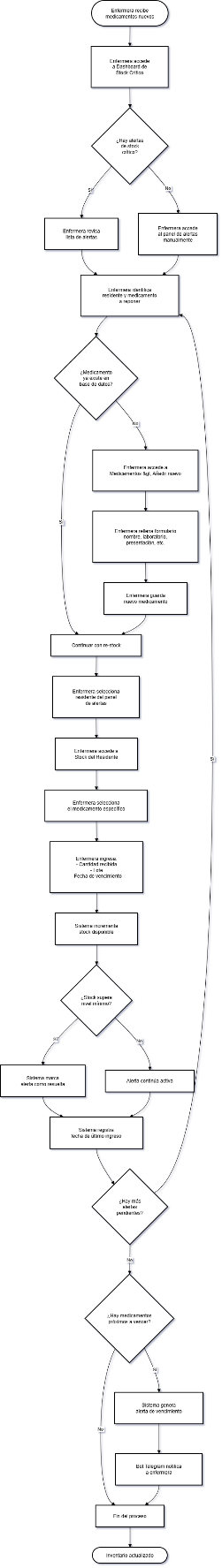
## 7.2 Proceso 1: Administración de Medicamentos



Flujo:

1. TENS crea asignación de carga turno
2. Sistema distribuye residentes equitativamente
3. Sistema notifica personal asignado
4. TENS prepara medicamentos según horarios
5. Sistema descuenta stock automáticamente
6. TENS notifica cuidadora para retirar
7. Cuidadora administra medicamento
8. Registro resultado:
   1. CHECK → registra fecha/hora, actualiza ADMINISTRADO
   2. NO CHECK → solicita motivo, NO descuenta, genera alerta
9. Evaluación retraso (>1h → alerta Telegram)
10. Actualización dashboard tiempo real
11. Actores: TENS, Cuidadora, Sistema, Bot Telegram
12. Entradas: Recetas activas, stock disponible, personal turno
13. Salidas: Registro administración, stock actualizado, alertas, notificaciones

## 7.3 Proceso 2: Gestión Inventario y Re-stock

****

Flujo:

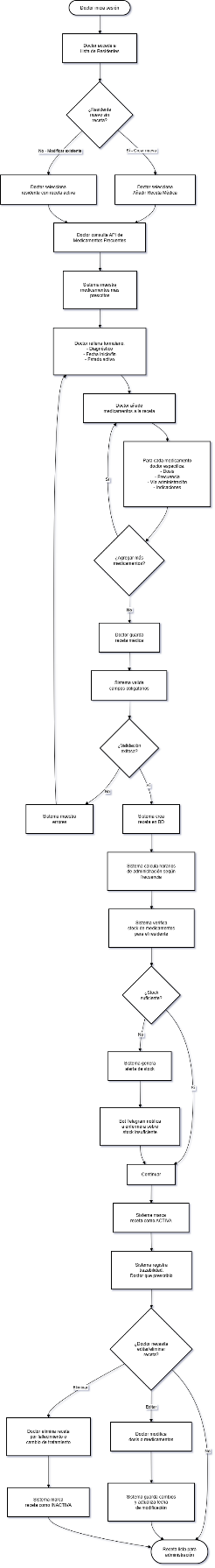
1. Sistema ejecuta tarea programada (cada 1 hora)
2. Evaluación stock todos los StockMedicamento
3. ¿Stock ≤ mínimo? → Alerta STOCK\_CRITICO
4. ¿Vence ≤30 días? → Alerta VENCIMIENTO\_PROXIMO
5. ¿Vencido? → Alerta MEDICAMENTO\_VENCIDO
6. Consolidación alertas por residente/medicamento
7. Dashboard alertas para Enfermera
8. Notificación Telegram si críticas
9. Enfermera visualiza dashboard stock crítico
10. Enfermera selecciona medicamento reponer
11. Enfermera ingresa: cantidad, lote, vencimiento
12. Sistema valida datos
13. Sistema actualiza stock, registra movimiento
14. ¿Stock > mínimo? → Desactivar alerta
15. Sistema registra trazabilidad
16. Fin

Actores: Sistema (automático), Enfermera, Bot Telegram

Entradas: Stock actual, stock mínimo.

Salidas: Alertas, stock repuesto, movimientos registrados

## 7.4 Proceso 3: Creación y Gestión Recetas Médicas



Flujo:

1. Doctor accede módulo Recetas
2. Doctor selecciona residente
3. Doctor busca medicamento:
   1. Opción A: Búsqueda manual catálogo
   2. Opción B: API medicamentos frecuentes (top 20)
4. Sistema muestra sugerencias
5. Doctor selecciona medicamento
6. Sistema valida alergias (alerta si existe)
7. Doctor ingresa detalles: dosis, frecuencia, vía, fecha inicio, duración, observaciones
8. Sistema valida duplicados (error si existe receta activa residente+medicamento)
9. Sistema calcula fecha fin si especificó duración
10. Sistema valida stock (advertencia si no hay, pero permite crear)
11. Sistema guarda receta estado ACTIVA
12. Sistema genera horarios administración
13. Sistema crea RegistroAdministracion próximos 7 días
14. Sistema registra trazabilidad
15. Opciones editar/desactivar:
    1. Editar: Solo doctor creador, registra auditoría, actualiza horarios
    2. Desactivar: Marca INACTIVA, cancela registros pendientes
16. Fin

Actores: Doctor, Sistema, API Medicamentos Frecuentes

Entradas: Residente, medicamento catálogo, datos prescripción

Salidas: Receta creada/editada/desactivada, horarios generados, trazabilidad

# Vista de escenarios

## 8.1 Descripción General

La vista de escenarios documenta casos de uso del sistema, mostrando interacciones de actores.

## 8.2 Diagrama CU - Enfermera (Super Admin)

## 

|  |  |  |
| --- | --- | --- |
| ID | Caso de Uso | Descripción |
| CU-ENF-01 | Gestionar Usuarios | CRUD Doctores, TENS, Cuidadoras |
| CU-ENF-02 | Gestionar Residentes | CRUD residentes |
| CU-ENF-03 | Gestionar Medicamentos | CRUD catálogo medicamentos |
| CU-ENF-04 | Gestionar Inventario | Ingreso y reposición stock |
| CU-ENF-05 | Visualizar Dashboard | Dashboard completo alertas |
| CU-ENF-06 | Generar Reportes | Exportar PDF de administración mensuales/inventario/trazabilidad |
| CU-ENF-07 | Visualizar Trazabilidad | Historial completo medicamentos |
| CU-ENF-08 | Resolver Alertas | Atender stock crítico |
| CU-ENF-09 | Editar Perfil Propio | Cambiar datos/contraseña |

## 8.3 Diagrama CU – Doctor

## 

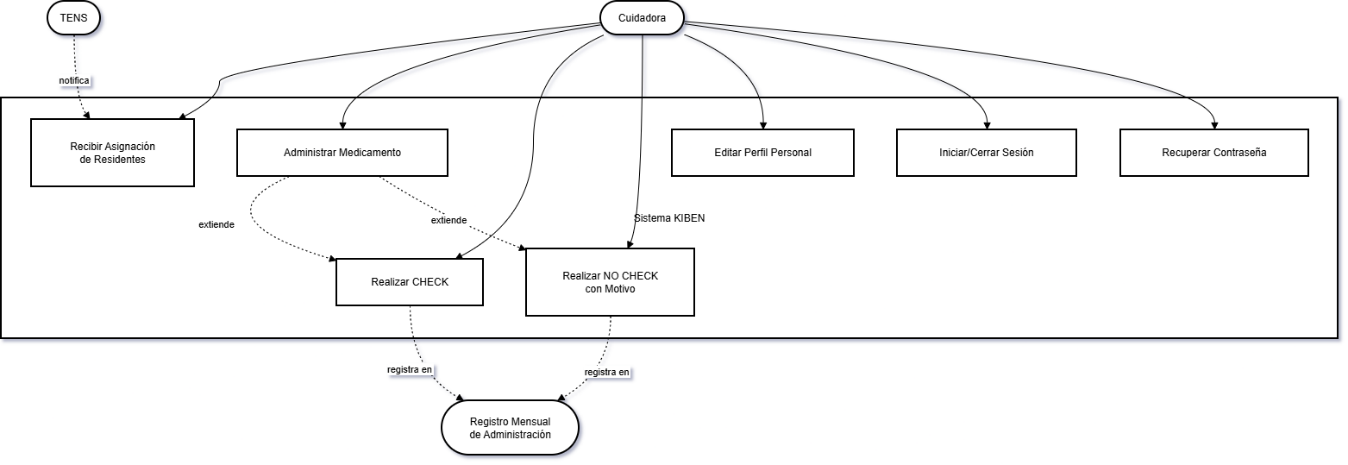
|  |  |  |
| --- | --- | --- |
| ID | Caso de Uso | Descripción |
| CU-DOC-01 | Crear Receta Médica | Prescribir medicamentos para un residente |
| CU-DOC-02 | Editar Receta Médica | Modificar dosis, frecuencia o duración |
| CU-DOC-03 | Desactivar Receta Médica | Suspender tratamiento |
| CU-DOC-04 | Consultar Medicamentos Frecuentes | Usar API externa para búsqueda de medicamentos |
| CU-DOC-05 | Visualizar Residentes (lectura) | Ver lista de residentes (solo lectura) |
| CU-DOC-06 | Visualizar Historial Administración | Ver registro de administración de medicamentos |
| CU-DOC-07 | Editar Perfil Propio | Cambiar datos/contraseña |

## 8.4 Diagrama CU – TENS

## 

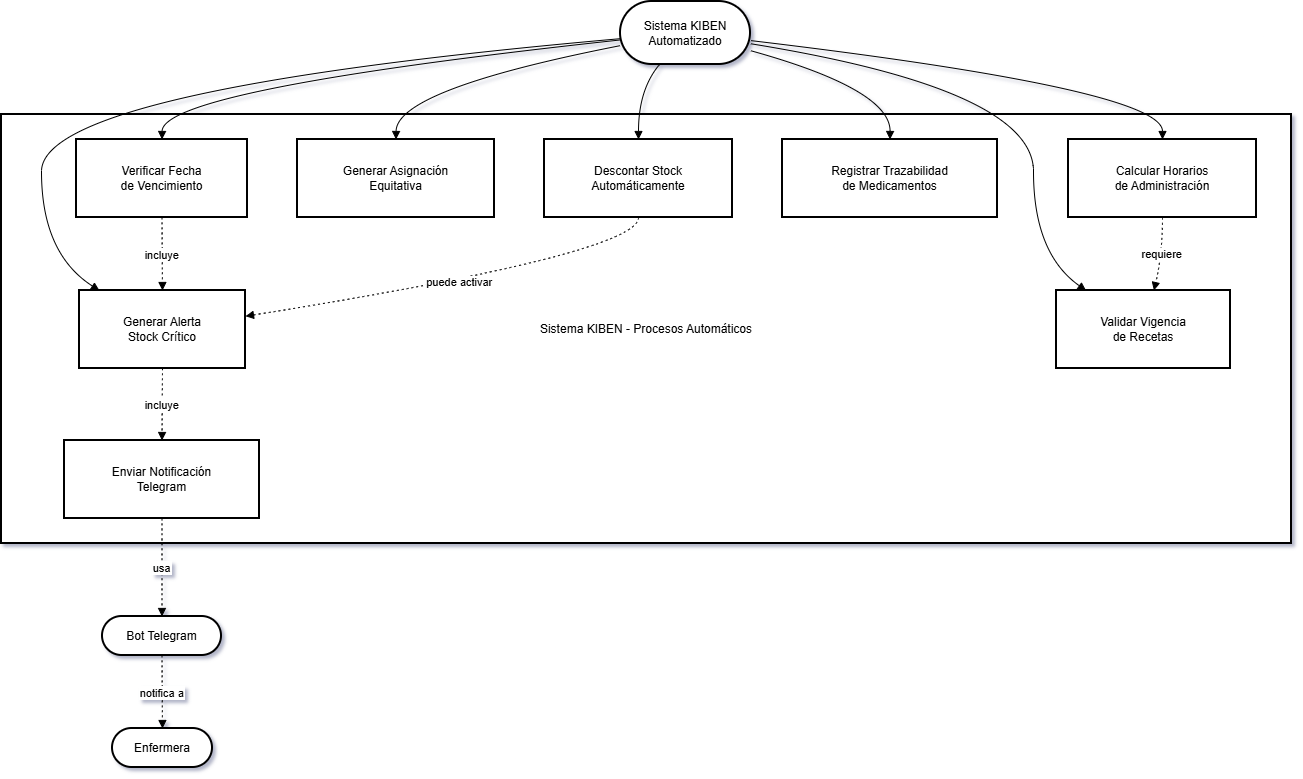
|  |  |  |
| --- | --- | --- |
| ID | Caso de Uso | Descripción |
| CU-TENS-01 | Gestionar Stock Residente | Asignar medicamentos individuales |
| CU-TENS-02 | Crear Asignación Carga | Asignar residentes y turnos a Cuidadoras |
| CU-TENS-03 | Administrar Medicamentos | Registrar administración directa (CHECK) |
| CU-TENS-04 | Registrar CHECK | Confirmar administración exitosa |
| CU-TENS-05 | Registrar NO CHECK | Registrar administración no realizada con motivo |
| CU-TENS-06 | Visualizar Dashboard Stock Crítico | Ver alertas de reabastecimiento |
| CU-TENS-07 | Visualizar Registro Mensual | Ver historial de administraciones |
| CU-TENS-08 | Recibir Notificaciones | Alertas vía Telegram y dashboard |
| CU-TENS-09 | Editar Perfil Propio | Cambiar datos/contraseña |

## 8.5 Diagrama CU - Cuidadora



|  |  |  |
| --- | --- | --- |
| ID | Caso de Uso | Descripción |
| CU-CUID-01 | Ver Asignaciones Turno | Visualizar residentes y medicamentos asignados |
| CU-CUID-02 | Ver Medicamentos Pendientes | Lista de medicamentos por administrar |
| CU-CUID-03 | Administrar Medicamento | Registrar administración (CHECK) |
| CU-CUID-04 | Registrar CHECK | Confirmar administración exitosa |
| CU-CUID-05 | Registrar NO CHECK | Registrar no administración con motivo |
| CU-CUID-06 | Visualizar Dashboard Personal | Ver solo asignaciones propias |
| CU-CUID-07 | Recibir Notificaciones | Alertas vía Telegram |
| CU-CUID-08 | Editar Perfil Propio | Cambiar datos/contraseña |

## 8.6 Diagrama de Casos de Uso - Sistema (Procesos Automáticos)



|  |  |  |
| --- | --- | --- |
| ID | Caso de Uso | Descripción |
| CU-SYS-01 | Generar Alertas Stock Crítico | Notificar stock bajo/agotado |
| CU-SYS-02 | Descontar Stock Automáticamente | Restar unidades al registrar CHECK |
| CU-SYS-03 | Enviar Notificaciones Telegram | Alertas urgentes vía bot |
| CU-SYS-04 | Registrar Trazabilidad | Auditoría completa de movimientos |
| CU-SYS-05 | Generar Reportes Automáticos | ReportesPDF mensuales programados |
| CU-SYS-06 | Detectar Retrasos Administración | Alertar medicamentos no administrados |
| CU-SYS-07 | Sincronizar Dashboard | Actualización en tiempo real |