Documento de requerimientos DE SOFTWARE

KIBEN

# Historial de Versiones

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Fecha | Versión | Autor | Organización | Descripción |
| 05.09.25 | 1.0 | Sergio Moreno y Vicente Quintana | M&Q Technology | Creación del documento |
| 24.10.25 | 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 | 24.10.25 |  |
| Sergio Moreno | 24.10.25 |  |
| Vicente Quintana | 24.10.25 |  |

# Propósito

## 1.1 Identificación del Software

Nombre: KIBEN

Versión: 1.1

Tipo: Sistema de Gestión Farmacológica para Instituciones Geriátricas

## 1.2 Alcance del Documento

Este ERS cubre la totalidad del sistema KIBEN:

* Aplicación Web (servidor)
* Aplicación Móvil (Android)
* Aplicación Escritorio (Electron)
* Backend API REST
* Base de Datos Relacional
* Integraciones (Telegram, API medicamentos CIMA)

# Alcance del producto / Software

## 2.1 Descripción General

KIBEN es un sistema integral para gestión de inventario y administración de medicamentos en hogares de ancianos. Gestiona el ciclo completo desde ingreso a bodega hasta administración al residente.

## 2.2 Objetivos

Objetivo general: Mejorar seguridad del paciente y eficiencia operativa mediante digitalización del proceso farmacológico.

Objetivos específicos:

* Trazabilidad completa del ciclo de medicamentos
* Reducir errores de administración
* Automatizar alertas de stock crítico
* Optimizar distribución de carga de trabajo
* Generar registros auditables
* Facilitar prescripción con API de medicamentos frecuentes

## 2.3 Beneficios

Para el Hogar:

* Reducción 70% errores médicos
* Ahorro 40% tiempo administrativo
* Reducción pérdidas por vencimiento
* Cumplimiento normativo

Para Personal:

* Interfaz intuitiva
* Acceso remoto móvil
* Alertas automáticas
* Distribución equitativa de trabajo

Para Residentes:

* Mayor seguridad
* Continuidad de tratamiento
* Trazabilidad completa

# Referencias

## 3.1 Documentos del Proyecto

|  |  |  |  |
| --- | --- | --- | --- |
| Documento | Versión | Fecha | Ubicación |
| Project Charter | 1.1 | 17.10.25 | Github |
| DAS Arquitectura 4+1 | 1.1 | 29.10.25 | Github |
| Plan de Pruebas | 1.1 | 05.11.25 | Github |
| Matriz de Riesgos | 1.1 | 24.10.25 | Github |

## 3.2 Normativas

* Ley 19.628: Protección vida privada (Chile)
* Ley 20.584: Derechos y deberes paciente (Chile)
* Norma Técnica N°151: Gestión medicamentos (MINSAL)
* OWASP Top 10: Seguridad web
* ISO/IEC 25010: Calidad de software
* ISO/IEC 27001: Seguridad de los datos

## 3.3 Tecnologías

* Python 3.10+, Django 4.2+
* Django REST Framework 3.14+
* Android SDK API 28+
* Electron 27+
* PostgreSQL 14+ / SQLite 3.x
* Bootstrap 5.3
* Python Telegram Bot API 20+

# Funcionalidades del producto

* F-01: Gestión de Autenticación y Seguridad
* F-02: Gestión de Residentes
* F-03: Gestión de Personal
* F-04: Gestión de Medicamentos
* F-05: Gestión de Recetas Médicas
* F-06: Gestión de Inventario (Stock)
* F-07: Registro de Administración de Medicamentos
* F-08: Asignación Automática de Carga
* F-09: Dashboard y Visualización
* F-10: Generación de Reportes
* F-11: Sistema de Notificaciones
* F-12: Gestión de Perfil de Usuario
* F-13: Trazabilidad del Ciclo de Medicamentos

# Clases y características de usuarios

## 5.1 ENFERMERA/ENFERMERO (Super Admin)

Características:

* Rol administrativo principal
* Mayor nivel de permisos (CRUD completo)
* Uso: Diario, múltiples veces/día
* Nivel técnico: Medio

Funcionalidades: F-01 a F-04 y F-06 a F-13

Responsabilidades:

* Crear cuentas de usuarios (única con permiso)
* Gestionar residentes y personal
* Controlar inventario general
* Reposición de stock
* Supervisión de alertas
* Generación reportes

Dispositivos: Computador, tablet, smartphone

## 5.2 DOCTOR/DOCTORA

Características:

* Rol especializado prescripción
* Uso: Diario, sesiones cortas
* Nivel técnico: Medio

Funcionalidades: F-05, F-10, F-12

Responsabilidades:

* Crear/modificar recetas
* Ajustar tratamientos
* Revisar historial administración
* Desactivar recetas

Dispositivos: Computador, tablet, smartphone

## 5.3 TENS

Características:

* Rol operativo intermedio
* Uso: Diario, intensivo
* Nivel técnico: Medio

Funcionalidades: F-07 a F-12

Responsabilidades:

* Preparar medicación
* Descontar stock
* Asignar carga a cuidadoras
* Administrar medicamentos
* Registrar CHECK/NO CHECK
* Notificar cuidadoras

Dispositivos: Computador, tablet, smartphone

## 5.4 CUIDADORA

Características:

* Rol operativo básico
* Uso: Diario, intensivo
* Nivel técnico: Bajo-Medio

Funcionalidades: F-07, F-11, F-12

Responsabilidades:

* Recibir medicamentos de TENS
* Administrar a residentes asignados
* Registrar CHECK/NO CHECK
* Responder notificaciones

Dispositivos: Computador, tablet, smartphone

# Entorno operativo

## 6.1 Hardware

Servidor:

* VPS: 2+ vCPU, 8 GB RAM
* Storage: 50 GB SSD
* Proveedor: Hostinger/AWS

Clientes:

* Web: PC dual-core 2GHz+, 4GB RAM
* Móvil: Android 10.0+, 2GB RAM
* Escritorio: Dual-core 2GHz+, 4GB RAM

## 6.2 Sistema Operativo

Servidor: Linux Ubuntu 20.04 LTS+ / CentOS 8+

Clientes:

* Web: Independiente (navegador)
* Móvil: Android 10.0+
* Escritorio: Windows 10/11, macOS 10.15+, Linux

## 6.3 Navegadores

* Chrome 90+
* Firefox 88+
* Edge 90+
* Safari 14+

## 6.4 Componentes

* Backend: Django 4.2+, DRF 3.14+, PostgreSQL/SQLite
* Frontend Web: Bootstrap 5.3, JavaScript ES6+, Chart.js
* Móvil: Android SDK, Retrofit, OkHttp, Room
* Escritorio: Electron 27+, Node.js 18+
* Integraciones: Python Telegram Bot, Django REST

# Requerimientos funcionales

## 7.1 F-01: AUTENTICACIÓN Y SEGURIDAD

Descripción: Acceso seguro según credenciales y roles

Prioridad: ALTA (MUST HAVE)

* REQ-001: Sistema solicita username y contraseña
* REQ-002: Valida existencia de username
* REQ-003: Valida contraseña con hash bcrypt
* REQ-004: Mensaje genérico "Credenciales inválidas" si falla
* REQ-005: Redirección a dashboard según rol
* REQ-006: Sesión activa 8 horas inactividad
* REQ-007: Contraseña: mínimo 8 caracteres
* REQ-008: NO permitir registro público (solo Enfermera crea cuentas)

## 7.2 F-02: GESTIÓN DE RESIDENTES

Descripción: Registrar y gestionar residentes

Prioridad: ALTA (MUST HAVE)

* REQ-009: SOLO Enfermera crea residentes en el sistema.
* REQ-010: Crear residentes con: RUT, nombres, apellidos, fecha nacimiento, ingreso
* REQ-011: Campos opcionales: contacto emergencia, alergias, condiciones médicas
* REQ-012: Validar RUT según algoritmo chileno
* REQ-013: Prevenir residentes duplicados (mismo RUT)
* REQ-014: Calcular edad automáticamente
* REQ-015: Editar información de residentes
* REQ-016: Inactivar residentes sin eliminar historial
* REQ-017: Mantener medicamentos asociados post-inactivación
* REQ-018: Búsqueda por: nombre, apellido, RUT
* REQ-019: Indicador visual de residentes con alertas
* REQ-020: Doctores/TENS visualizan (solo lectura)
* REQ-021: Cuidadoras solo ven asignados

## 7.3 F-03: GESTIÓN DE PERSONAL

Descripción: Gestionar cuentas de usuarios

Prioridad: ALTA (MUST HAVE)

* REQ-022: SOLO Enfermera crea cuentas
* REQ-023: Crear con: RUT, nombres, apellidos, email, rol, username
* REQ-024: Generar contraseña temporal
* REQ-025: Enviar email con credenciales
* REQ-026: Email único
* REQ-027: RUT único
* REQ-028: Editar información usuarios
* REQ-029: Desactivar usuarios
* REQ-030: NO eliminar con registros asociados
* REQ-031: Reactivar usuarios desactivados

## 7.4 F-04: GESTIÓN DE MEDICAMENTOS

Descripción: Catálogo general de medicamentos

Prioridad: ALTA (MUST HAVE)

* REQ-032: Agregar con: nombre, concentración, forma.
* REQ-033: Opcionales: vía, contraindicaciones
* REQ-034: Combinación nombre+concentración+forma única
* REQ-035: Búsqueda por nombre
* REQ-036: Autocompletar nombre al escribir
* REQ-037: Editar medicamentos
* REQ-038: Desactivar medicamentos descontinuados
* REQ-039: NO eliminar con recetas activas
* REQ-040: Doctores/TENS visualizan (lectura)

## 7.5 F-05: GESTIÓN DE RECETAS

Descripción: Prescripciones médicas para residentes

Prioridad: ALTA (MUST HAVE)

* REQ-041: SOLO Doctor crea recetas o tratamientos médicos
* REQ-042: Agregar con: residente, medicamento, dosis, frecuencia, vía, fecha inicio, duración
* REQ-043: API sugiere medicamentos más frecuentes
* REQ-044: Calcular fecha fin automáticamente
* REQ-045: Doctores editan recetas activas
* REQ-046: Historial de cambios (auditoría)
* REQ-047: Desactivar recetas finalizadas
* REQ-048: Alertar alergias conocidas
* REQ-049: Campo observaciones
* REQ-050: Enfermeras/TENS visualizan (lectura)
* REQ-051: Estados: Activa, Pausada, Finalizada
* REQ-052: Generar horarios según frecuencia

## 7.6 F-06: GESTIÓN DE INVENTARIO

Descripción: Control stock por residente

Prioridad: ALTA (MUST HAVE)

* REQ-053: SOLO Enfermera gestiona inventario
* REQ-054: Ingresar con: residente, medicamento, cantidad, stock crítico
* REQ-055: Calcular stock actual automáticamente
* REQ-056: TENS descuenta al preparar
* REQ-057: Registrar movimientos: usuario, fecha/hora, cantidad, tipo
* REQ-058: Alerta cuando stock ≤ crítico
* REQ-059: NO permitir descuento > stock actual
* REQ-060: Ajuste manual con justificación
* REQ-061: Historial de movimientos
* REQ-062: Reposición por Enfermera
* REQ-063: Dashboard stock crítico

## 7.7 F-07: REGISTRO DE ADMINISTRACIÓN

Descripción: Registro digital administración medicamentos

Prioridad: ALTA (MUST HAVE - CORE)

* REQ-064: Lista administraciones pendientes por turno
* REQ-065: CHECK con 1 clic/tap
* REQ-066: NO CHECK con motivo obligatorio
* REQ-067: Motivos: Rechazo, Ausencia, Condición no apta, Pérdida
* REQ-068: Registro auto: usuario, fecha/hora, estado, motivo
* REQ-069: Alertar retraso ≤ 1 hora
* REQ-070: Confirmación visual clara
* REQ-071: NO editar después 1 hora
* REQ-072: Historial por medicamento/residente
* REQ-073: Notificación Telegram si no administrado > 2 horas
* REQ-073: Color verde: administrado
* REQ-075: Color rojo: no administrado
* REQ-076: Color amarillo: pendiente

## 7.8 F-08: ASIGNACIÓN DE CARGA

Descripción: Distribución equitativa de residentes

Prioridad: MEDIA (SHOULD HAVE)

* REQ-077: Crear con: fecha, turno, personal, horas
* REQ-078: Calcular residentes con medicamentos programados
* REQ-079: Distribuir equitativamente
* REQ-080: Considerar: cantidad residentes, medicamentos, complejidad
* REQ-081: Ajuste manual permitido
* REQ-082: Previsualización antes confirmar
* REQ-083: Notificación telegram a asignados
* REQ-084: Permitir reasignación
* REQ-085: Historial de asignaciones

## 7.9 F-09: DASHBOARD

Descripción: Paneles personalizados por rol

Prioridad: ALTA (MUST HAVE)

* REQ-086 : Dashboard según rol al login
* REQ-087: Enfermera: alertas, stock crítico, total administraciones del día, puede reponer stock desde dashboard.
* REQ-088: Doctor: residentes, acceso rápido crear receta
* REQ-089: TENS: stock crítico, total administraciones del día.
* REQ-090: Cuidadora: administraciones asignadas
* REQ-091: Actualización auto al refrescar
* REQ-092: Botón "Refrescar" manual
* REQ-093: Código colores: Verde/OK, Amarillo/Atención, Rojo/Crítico
* REQ-094: Clic en indicadores ver detalle

## 7.10 F-10: REPORTES

Descripción: Generación y exportación PDF

Prioridad: ALTA (MUST HAVE)

* REQ-095: Registro Mensual Administración: datos residente, período, lista administraciones, estados, motivos, usuarios, timestamps
* REQ-096: Exportar a PDF
* REQ-097: PDF con: logo, fecha generación, usuario, páginas, footer
* REQ-098: Filtros: residente, fechas, medicamento, estado
* REQ-099: Generar PDF < 5 segundos
* REQ-100: Previsualización disponible
* REQ-101: Doctores/Enfermeras/TENS visualizan sin editar

## 7.11 F-11: NOTIFICACIONES

Descripción: Alertas automáticas vía Telegram

Prioridad: MEDIA (SHOULD HAVE)

* REQ-102: Notificar stock crítico, dosis no administrada > 1h, nueva asignación
* REQ-103: Latencia máxima de 1 minuto
* REQ-104: Incluir tipo, residente, medicamento, acción, timestamp
* REQ-105: Vincular cuenta Telegram (QR/token)
* REQ-106: Stock crítico a Enfermera
* REQ-107: Dosis no administrada a TENS supervisor
* REQ-108: Configurar umbrales alertas
* REQ-109: NO duplicar en <1 hora
* REQ-110: Log notificaciones enviadas

## 7.12 F-12: PERFIL DE USUARIO

Descripción: Gestión información personal

Prioridad: ALTA (MUST HAVE)

* REQ-111: TODOS acceden a perfil
* REQ-112: Editar: nombre, apellido, email, contraseña
* REQ-113: NO editar: RUT, username, rol
* REQ-114: Validar formato email
* REQ-115: Email no duplicado
* REQ-116: Solicitar contraseña actual para cambio
* REQ-117: Nueva contraseña cumple requisitos
* REQ-118: Email confirmación al cambiar email
* REQ-119: Cerrar sesiones al cambiar contraseña
* REQ-120: Mensaje "Cambios guardados"

## 7.13 F-13: TRAZABILIDAD

Descripción: Historial completo ciclo medicamento

Prioridad: ALTA (MUST HAVE)

* REQ-121: Registrar auto: Prescripción, Ingreso, Preparación, Administración
* REQ-122: Consultar por: residente, medicamento, fechas
* REQ-123: Línea tiempo visual
* REQ-124: Cada etapa: usuario, fecha/hora, acción, datos
* REQ-125: Exportar a PDF
* REQ-126: Mantener si inactivados
* REQ-127: Registros inmutables
* REQ-128: Enfermeras acceso completo
* REQ-129: Doctores/TENS lectura
* REQ-130: Cuidadoras sin acceso

# Reglas de negocio

## 8.1 Generales del sistema

* RN-001: Solo usuarios autenticados con rol válido acceden
* RN-002: Un rol único activo por usuario
* RN-003: Solo Enfermera crea/edita/desactiva cuentas
* RN-004: Sistema cerrado: NO auto-registro
* RN-005: Acciones críticas auditadas (timestamp + usuario)
* RN-006: Datos sensibles: confidencialidad Ley 19.628

## 8.2 Residentes

* RN-007: Residente activo en 1 sucursal a la vez
* RN-008: Inactivo mantiene historial completo
* RN-009: NO eliminar con recetas activas o inventario

## 8.3 Personal

* RN-010: Usuario en 1 sucursal a la vez
* RN-011: Desactivado no puede login pero mantiene registros
* RN-012: NO eliminar con prescripciones/administraciones asociadas
* RN-013: Solo Doctores crean/editan recetas
* RN-014: Solo TENS o Enfermeras crean asignaciones
* RN-015: Solo Enfermeras gestionan inventario (ingreso/reposición)
* RN-016: TENS y Cuidadoras registran administración

## 8.4 Recetas

* RN-017: Receta asociada a residente + medicamento catálogo
* RN-018: NO duplicar recetas activas (residente + medicamento)
* RN-019: Solo Doctor creador edita/desactiva
* RN-020: Receta desactivada no se reactiva
* RN-021: Obligatorio: dosis, frecuencia, vía, fecha inicio

## 8.5 Inventario

* RN-022: Stock nunca negativo
* RN-023: NO descontar más que disponible
* RN-024: Registrar movimientos: tipo, cantidad, usuario, fecha/hora
* RN-025: Alerta auto cuando stock ≤ crítico
* RN-026: NO administrar sin stock
* RN-027: Solo TENS descuenta stock

## 8.6 Administración

* RN-028: Toda administración: CHECK o NO CHECK
* RN-029: NO CHECK obliga motivo
* RN-030: Registro auto: usuario, fecha/hora
* RN-031: Solo Enfermeras/TENS/Cuidadoras registran
* RN-032: Cuidadoras solo asignados
* RN-033: Alerta si no administrado >1h

## 8.7 Asignación

* RN-034: Especificar: fecha, turno, personal, residentes
* RN-035: Desviación máxima 10%
* RN-036: Solo Enfermeras/TENS crean o modifican
* RN-037: Una persona múltiples turnos días diferentes, NO mismo día/turno

## 8.8 Trazabilidad

* RN-038: Cambios críticos auditables
* RN-039: Trazabilidad inmutable
* RN-040: Incluir: usuario, fecha/hora, acción, datos

## 8.9 Seguridad

* RN-041: Contraseña: mínimo 8 chars, 1 mayúscula, 1 minúscula, 1 número
* RN-042: Contraseñas hash bcrypt
* RN-043: 5 intentos fallidos = bloqueo 15 min
* RN-044: Sesión expira 8h inactividad
* RN-045: Datos sensibles encriptados en BD
* RN-046: Producción solo HTTPS

# Requerimientos de interfaces externas

## 9.1 Interfaces de Usuario

* REQ-INT-001: Seguir Material Design o Bootstrap
* REQ-INT-002: Responsive: desktop, laptop, tablet, smartphone
* REQ-INT-003: Código colores: Verde/éxito, Amarillo/advertencia, Rojo/crítico, Azul/info, Gris/deshabilitado
* REQ-INT-004: Formularios con validación y mensajes claros
* REQ-INT-005: Navegación consistente

Web:

* REQ-INT-006: Compatibilidad navegadores modernos
* REQ-INT-007: Menú navegación según rol
* REQ-INT-008: Barra superior: logo, atajos según rol, usuario, rol, perfil, logout
* REQ-INT-009: Tablas: filtro, búsqueda, ordenamiento

Móvil:

* REQ-INT-010: Material Design Android
* REQ-INT-011: Bottom nav o drawer
* REQ-INT-012: Teclado contextual
* REQ-INT-013: Portrait y landscape
* REQ-INT-014: Elementos táctiles ≥48dp separación

Escritorio:

* REQ-INT-015: Apariencia nativa SO
* REQ-INT-016: Menú estándar: Archivo, Editar, Ver, Ayuda
* REQ-INT-017: Mantener estado ventana

## 9.2 Interfaces de Hardware

* REQ-INT-018: Soporte impresoras SO para PDF
* REQ-INT-019: Móvil accede: notificaciones
* REQ-INT-020: Web funciona: desktop, laptop, tablets
* REQ-INT-021: Móvil funciona: smartphones ≥5", tablets 7-10"
* REQ-INT-022: NO requiere dispositivos especializados

## 9.3 Interfaces de Software

Base de Datos:

* REQ-INT-023: PostgreSQL 14+ / SQLite 3.x vía Django ORM
* REQ-INT-024: Queries parametrizadas (anti SQL injection)
* REQ-INT-025: Migraciones Django
* REQ-INT-026: Backup automático diario

API REST:

* REQ-INT-027: Exponer vía Django REST Framework
* REQ-INT-028: Autenticación JWT o Session
* REQ-INT-029: Convenciones RESTful: GET/POST/PUT/PATCH/DELETE
* REQ-INT-030: Respuestas JSON
* REQ-INT-031: Códigos HTTP: 200, 201, 400, 401, 403, 404, 500
* REQ-INT-032: Paginación listados grandes
* REQ-INT-033: Documentación OpenAPI/Swagger

Bot Telegram:

* REQ-INT-034: Integración Telegram Bot API
* REQ-INT-035: Autenticación token único
* REQ-INT-036: Formato Markdown
* REQ-INT-037: Manejo errores sin afectar core

Sistema Operativo:

* REQ-INT-038: Backend Linux (Ubuntu 20.04+ / CentOS 8+)
* REQ-INT-039: Systemd o supervisor para procesos
* REQ-INT-040: Logging Python logging module

## 9.4 Interfaces de Comunicación

HTTP/HTTPS:

* REQ-INT-041: Producción solo HTTPS
* REQ-INT-042: SSL válido
* REQ-INT-043: Rechazar HTTP no seguro
* REQ-INT-044: CORS para móvil/escritorio

Email:

* REQ-INT-045: SMTP para nuevos usuarios, confirmaciones
* REQ-INT-046: HTML con fallback texto plano
* REQ-INT-047: Footer con info sistema
* REQ-INT-048: Plantillas consistentes

Seguridad:

* REQ-INT-049: Encriptar datos sensibles
* REQ-INT-050: Tokens con expiración y renovación
* REQ-INT-051: NUNCA transmitir contraseñas sin encriptar

# Requerimientos no funcionales

## 10.1 Rendimiento

* REQ-NF-001: CRUD <3 segundos (95% casos)
* REQ-NF-002: Carga inicial web <5 segundos (5 Mbps)
* REQ-NF-003: Dashboard refresh <2 segundos
* REQ-NF-004: PDF mensual <15 segundos
* REQ-NF-005: Soportar 500 usuarios concurrentes
* REQ-NF-006: Queries BD <1 segundo (90% casos)
* REQ-NF-007: App móvil inicio <3 segundos
* REQ-NF-008: Optimizar imágenes/assets

## 10.2 Disponibilidad

* REQ-NF-009: Uptime mínimo 99% mensual (≤7.3h downtime)
* REQ-NF-010: Objetivo 99.5% (≤3.6h downtime)
* REQ-NF-011: Backup BD automático diario
* REQ-NF-012: Backups en ubicación separada
* REQ-NF-013: Recuperación <4 horas
* REQ-NF-014: Logs de errores para diagnóstico
* REQ-NF-015: Retry para operaciones críticas fallidas

## 10.3 Escalabilidad

* REQ-NF-016: Arquitectura permite escalado horizontal
* REQ-NF-017: Soportar hasta 1,000 residentes sin cambios
* REQ-NF-018: Soportar hasta 2,000 usuarios sin cambios
* REQ-NF-019: Migrar SQLite→PostgreSQL sin pérdida
* REQ-NF-020: Sucursales ilimitadas técnicamente

## 10.4 Seguridad

* REQ-NF-021: Bcrypt factor costo ≥12
* REQ-NF-022: ORM protege SQL injection
* REQ-NF-023: Escape HTML anti XSS
* REQ-NF-024: Tokens CSRF
* REQ-NF-025: Rate limit: máx 10 requests/min login
* REQ-NF-026: Validar/sanitizar inputs
* REQ-NF-027: HTTPS producción
* REQ-NF-028: Tokens sesión expiran 8h
* REQ-NF-029: RBAC todos endpoints
* REQ-NF-030: Log accesos fallidos/sospechosos
* REQ-NF-031: Cumplir Ley 19.628
* REQ-NF-032: Encriptar datos sensibles BD

## 10.5 Usabilidad

* REQ-NF-033: Intuitivo: 2h capacitación tareas principales
* REQ-NF-034: Mensajes error claros español
* REQ-NF-035: Tooltips ayuda contextual
* REQ-NF-036: Confirmar acciones destructivas
* REQ-NF-037: Feedback visual inmediato
* REQ-NF-038: Navegación consistente
* REQ-NF-039: Formularios agrupados lógicamente
* REQ-NF-040: Recordar preferencias usuario
* REQ-NF-041: Interfaz Cuidadoras MUY SIMPLE ≤3 pasos

## 10.6 Mantenibilidad

* REQ-NF-042: Código PEP 8, docstrings
* REQ-NF-043: Módulos responsabilidades claras
* REQ-NF-044: Tests cobertura ≥70% módulos críticos
* REQ-NF-045: Git commits descriptivos
* REQ-NF-046: Documentación técnica actualizada
* REQ-NF-047: Variables entorno (no hardcode)
* REQ-NF-048: Evitar duplicación (DRY)

## 10.7 Portabilidad

* REQ-NF-049: Web sin plugins
* REQ-NF-050: Móvil Android 9.0+ (API 28+)
* REQ-NF-051: Escritorio Windows 10/11, macOS 10.15+, Linux Ubuntu 20.04+
* REQ-NF-052: Despliegue múltiples proveedores hosting
* REQ-NF-053: Formatos estándar: PDF, CSV, JSON

## 10.8 Compatibilidad

* REQ-NF-054: Navegadores: Chrome 90+, Firefox 88+, Safari 14+, Edge 90+
* REQ-NF-055: Python 3.10+
* REQ-NF-056: PostgreSQL 14+ y SQLite 3.x
* REQ-NF-057: API REST estándar

## 10.9 Localización

* REQ-NF-058: Español chileno
* REQ-NF-059: Fechas DD/MM/YYYY
* REQ-NF-060: Números: punto miles, coma decimal
* REQ-NF-061: Moneda CLP
* REQ-NF-062: Zona horaria CLT/CLST auto

## 10.10 Cumplimiento Legal

* REQ-NF-063: Ley 19.628 Protección Privacidad
* REQ-NF-064: Ley 20.584 Derechos Pacientes
* REQ-NF-065: Derechos ARCO (Acceso, Rectificación, Cancelación, Oposición)
* REQ-NF-066: Registros auditables MINSAL
* REQ-NF-067: Política privacidad y términos uso
* REQ-NF-068: Consentimiento informado datos personales

# Otros requerimientos

## 11.1 Base de Datos

* REQ-OT-001: Integridad referencial (foreign keys)
* REQ-OT-002: Índices en campos frecuentes (RUT, email, username)
* REQ-OT-003: Constraints validación nivel BD
* REQ-OT-004: Fecha/hora timezone aware
* REQ-OT-005: Soft delete (campo activo)
* REQ-OT-006: Auditoría: created\_at, updated\_at, created\_by, updated\_by
* REQ-OT-007: Migraciones reversibles

## 11.2 Logging y Monitoreo

* REQ-OT-008: Logs archivo rotación diaria
* REQ-OT-009: Niveles: DEBUG, INFO, WARNING, ERROR, CRITICAL
* REQ-OT-010: Producción INFO+ (no DEBUG)
* REQ-OT-011: Excepciones con stack trace
* REQ-OT-012: Error tracking (Sentry)
* REQ-OT-013: Access logs servidor web
* REQ-OT-014: Monitorear recursos (CPU, RAM, disco)

## 11.3 Deployment

* REQ-OT-015: WSGI
* REQ-OT-016: Reverse proxy (Nginx)
* REQ-OT-017: Servidor web para archivos estáticos
* REQ-OT-018: Deployment automatizado/semi
* REQ-OT-019: Variables entorno por ambiente
* REQ-OT-020: Script inicialización BD con datos prueba

## 11.4 Testing

* REQ-OT-021: Tests unitarios lógica crítica
* REQ-OT-022: Tests integración endpoints API
* REQ-OT-023: Tests funcionales (E2E) flujos principales
* REQ-OT-024: Cobertura ≥70% módulos core
* REQ-OT-025: Tests auto pre-commit hooks
* REQ-OT-026: Fixtures datos prueba

## 11.5 Documentación

* REQ-OT-027: README instalación/configuración
* REQ-OT-028: Docs API (OpenAPI)
* REQ-OT-029: Manual usuario por rol
* REQ-OT-030: Guía deployment producción
* REQ-OT-031: Comentarios secciones complejas
* REQ-OT-032: Docstrings funciones públicas

## 11.6 Accesibilidad

* REQ-OT-035: WCAG 2.1 nivel AA
* REQ-OT-036: HTML5 semántico
* REQ-OT-037: Labels asociados inputs
* REQ-OT-038: alt="" decorativo, alt descriptivo informativo
* REQ-OT-039: Navegación teclado completa
* REQ-OT-040: Estados foco visibles

## 11.7 Reúso

* REQ-OT-041: Componentes formularios CRUD reutilizables
* REQ-OT-042: Componente tablas paginación reutilizable
* REQ-OT-043: Componente modales confirmación reutilizable
* REQ-OT-044: Componente generación PDF reutilizable
* REQ-OT-045: Documentar componentes reutilizables

# Glosario

Términos Médicos

* Administración: Dar/aplicar medicamento según prescripción
* Concentración: Cantidad principio activo (ej: 500mg)
* Contraindicación: Situación desaconseja medicamento
* Dosis: Cantidad en una toma
* Forma Farmacéutica: Presentación física (comprimido, jarabe, etc.)
* Frecuencia: Periodicidad administración (ej: cada 8h)
* Inventario: Stock disponible por residente
* Lote: Número identificación fabricación
* Medicamento: Sustancia propiedades curativas/preventivas
* Prescripción/Receta: Orden médica escrita
* Principio Activo: Componente efecto terapéutico
* Stock: Cantidad unidades disponibles
* Stock Crítico: Nivel ≤ mínimo, requiere reposición
* Stock Mínimo: Cantidad mínima antes reponer
* Trazabilidad: Seguir recorrido completo medicamento
* Vía Administración: Ruta (oral, IM, IV, tópica, etc.)

Roles

* Cuidadora: Auxiliar administra medicamentos según asignación
* Doctor: Médico prescribe tratamientos
* Enfermera: Profesional Super Admin gestión integral
* TENS: Técnico prepara medicación, descuenta stock, asigna carga
* Residente: Adulto mayor en hogar recibe atención
* Sucursal: Sede física hogar ancianos

Sistema

* API: Application Programming Interface
* API REST: Arquitectura API con HTTP
* Backend: Servidor, lógica negocio, BD
* CHECK: Medicamento SÍ administrado
* CRUD: Create, Read, Update, Delete
* Dashboard: Panel control información resumida
* Frontend: Interfaz gráfica usuario
* NO CHECK: Medicamento NO administrado (requiere motivo)
* ORM: Object-Relational Mapping
* PDF: Portable Document Format
* RBAC: Role-Based Access Control
* Soft Delete: Marcar inactivo sin eliminar físicamente
* Token: Cadena autenticación usuario
* Timestamp: Marca fecha/hora evento
* Uptime: Tiempo sistema operativo

Técnicos

* Bcrypt: Hash criptográfico contraseñas
* Bootstrap: Framework CSS responsive
* CORS: Cross-Origin Resource Sharing
* Django: Framework web Python
* Electron: Framework apps escritorio web tech
* HTTPS: HTTP seguro encriptado
* JWT: JSON Web Token
* PostgreSQL: SGBD relacional open source
* SQLite: BD relacional archivo único
* SSL: Secure Sockets Layer
* XSS: Cross-Site Scripting

Gestión Proyecto

* MVP: Minimum Viable Product
* MoSCoW: Must, Should, Could, Won't have
* Stakeholder: Persona/grupo interés proyecto

Acrónimos

* API: Application Programming Interface
* CRUD: Create, Read, Update, Delete
* DAS: Documento Arquitectura Sistema
* ERS: Especificación Requisitos Software
* HTTPS: HyperText Transfer Protocol Secure
* ISP: Instituto Salud Pública (Chile)
* JSON: JavaScript Object Notation
* MINSAL: Ministerio Salud (Chile)
* ORM: Object-Relational Mapping
* PDF: Portable Document Format
* QA: Quality Assurance
* RBAC: Role-Based Access Control
* REST: Representational State Transfer
* RUT: Rol Único Tributario (Chile)
* SCRUM: Marco trabajo ágil
* SMS: Short Message Service
* SQL: Structured Query Language
* SSL: Secure Sockets Layer
* TENS: Técnico Enfermería Nivel Superior
* UI: User Interface
* URL: Uniform Resource Locator
* UX: User Experience
* VPS: Virtual Private Server
* WCAG: Web Content Accessibility Guidelines
* WSGI: Web Server Gateway Interface
* XSS: Cross-Site Scripting