Diseño de Arquitectura del Sistema - TaskManager

**Fecha:**

**Autor:**

**Versión:** 1.0

# 1. Introducción

Este documento describe la arquitectura del sistema **AYNI**, una aplicación web para la gestión de pacientes y control de la dispensación de leche en el Banco de Leche Humana del Hospital Regional del Cusco. Define la estructura de alto nivel, componentes, interacciones, tecnologías, requisitos no funcionales críticos (seguridad, disponibilidad, rendimiento) y el plan general de despliegue, alineado con la Especificación de Requisitos del Software.

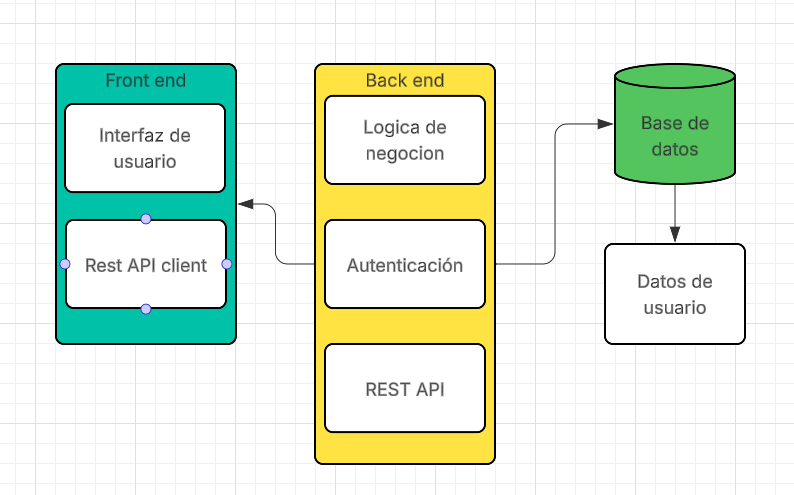
# 2. Visón General de la Arquitectura

## 2.1. Estilo Arquitectónico

Arquitectura en capas (cliente/servidor) con separación clara de presentación, lógica de negocio y persistencia. API RESTful entre frontend y backend. Posible adopción de microservicios en fases futuras (si la carga y la complejidad lo requieren), empezando por un monolito modular en Spring Boot.

* Capa de Presentación (Frontend): Aplicación en Angular, responsiva para escritorio y móviles.
* Capa de Negocio (Backend): Servicios REST implementados en Spring Boot (Java). Control de seguridad, reglas de negocio y validaciones.
* Capa de Persistencia: Base de datos relacional MySQL.

## 2.2. Diagrama General de Arquitectura:



# 3. Descripción de Componentes Principales

## 3.1. Frontend (Presentación)

* **Tecnologías:** Angular 19, Angular Material, HTML5, CSS3.

**Responsabilidades:**

* Formularios para registro de pacientes, madres y donantes.
* Registro de pase de visita diario y programación de dispensaciones.
* Gestión de inventario (entrada/salida de frascos), movimientos y visualización de lotes.
* Módulo de reportes (generación y exportación: PDF/CSV).
* Autenticación de usuarios y control de sesiones (JWT almacenado de forma segura).
* Validaciones de entrada y mensajes de error amigables.

## 3.2. Backend (Lógica de negocio)

* **Tecnologías:** Java + Spring Boot, Spring Security, Spring Data JPA (Hibernate).

Responsabilidades:

* API RESTful segura para todas las operaciones.
* Validación de datos y reglas de negocio (por ejemplo: asignación de leche según pase de visita, cálculo de cantidades por toma).
* Gestión de usuarios y roles (roles mínimos: ADMIN, MEDICO, ENFERMERA, ALMACEN).
* Registro de auditoría para acciones críticas (quién, cuándo, qué cambió).
* Envío de notificaciones (por ejemplo: notificar frascos próximos a vencer, alertas de stock bajo).
* Servicios para generación de reportes y exportación.

### 3.2.1. Módulos del Backend

* AuthController(gestión de usuarios y roles).
* PacienteController (CRUD filiación y datos clínicos).  
  MadreController / DonanteController (registro, consentimiento).
* PaseVisitaController (registro diario, horarios).
* DispensacionController (crear, actualizar, anular dispensaciones).
* InventarioController (gestión de frascos, lotes, entrada/salida).
* ReporteController (generación de reportes individuales y generales).
* RegistropasteurizadaController (registro de leche pasteurizada)
* RegistroautologaController (registro de leche autologa)
* CunaController(registro de cunas disponibles)
* DiagnosticoController(diagnosticos para cada paciente)

## 3.3. Base de Datos (Persistencia)

* **DBMS:** MySQL. Diseñado para integridad referencial y transaccionalidad.
* **Acceso:** Spring Data JPA.
* Esquema de Datos:
  + **Tabla cuna:**  
     Id\_Cuna, Estado\_Cuna
  + **Tabla datos\_extra:**  
     id\_datos\_extra, num\_pre\_inscripciones, higiene\_manos, higiene\_mamas,  
     tecnica\_extraccion\_leche\_materna\_intrahospitalaria, tecnica\_extraccion\_leche\_materna\_extrahospitalaria,  
     lavado\_mamas, calostro, transicion, madura, tasa\_descarte, cantidad\_antonio\_lorena, cantidad\_otros,  
     numero-donantes, numero-donantes-externos, calostro\_al, transicion\_al, madura\_al, tasa\_descarte\_al, fecha
  + **Tabla diagnostico\_paciente:**  
     Id\_Diagnostico\_Paciente, Observacion\_Enfermedad
  + **Tabla dispensacion:**  
     Id\_Dispensacion, Id\_Pase\_Visita, Id\_Paciente, Toma\_1 … Toma\_16,  
     Leche\_Pasteurizada, LDM, Leche\_autologa\_formula, Leche\_Formula\_termino,  
     Leche\_autologa\_pasteurizada, Leche\_autologa, Leche\_pasteurizada\_formula,  
     Leche\_Formula\_pretermino, Fecha
  + **Tabla donadora:**

Id\_Donadora, Nombre\_Donadora, Apellido\_Paterno\_Donadora, Apellido\_Materno\_Donadora, Fecha\_Nacimiento\_Donadora,

Telefono\_Donadora, Talla\_Donadora, Departamento, Provincia, Distrito, Direccion\_Actual\_Donadora,

Centro\_Salud\_Control\_Procedencia, Numero\_Controles, Ocupacion, Transfusion\_Sangre\_Madre, Consumo\_Cigarros,Consumo\_Drogas, Consumo\_Medicamentos, Enfermedades, Prueba\_Serologicos, Prueba\_Sifilis, Prueba\_Hepatitis,Prueba\_VIH, Examen\_Hemoglobina, Enfermedad\_Actual, Donar\_Leche, Apta\_Para\_Donar, Consentimiento\_Donadora

* + **Tabla enfermedad:** Id\_Enfermedad, Nombre\_Enfermedad, Categoria\_Enfermedad
  + **Tabla enfermedades\_paciente:**  
     Id\_Enfermedad\_Paciente, Id\_Diagnostico\_Paciente, Id\_Enfermedad
  + **Tabla madre:**  
     Id\_Madre, Nombre\_Madre, Apellido\_Paterno\_Madre, Apellido\_Materno\_Madre, Fecha\_Nacimiento\_Madre,  
     Telefono\_Madre, Talla\_Madre, Departamento, Provincia, Distrito, Direccion\_Actual\_Madre,  
     Centro\_Salud\_Control\_Procedencia, Numero\_Controles, Ocupacion, Peso\_Inicial\_Madre\_Gestante,  
     Peso\_Final\_Madre\_Gestante, Transfusion\_Sangre\_Madre, Consumo\_Cigarros, Consumo\_Drogas, Consumo\_Medicamentos,  
     Enfermedades, Prueba\_Serologicos, Prueba\_Sifilis, Prueba\_Hepatitis, Prueba\_VIH, Examen\_Hemoglobina,  
     Enfermedad\_Actual, Donar\_Leche, Apta\_Para\_Donar, Menor\_de\_Edad, Consentimiento\_Madre, Id\_Paciente
  + **Tabla apoderado:**  
     Id\_Apoderado, Parentesco, Nombre\_Apoderado, Apellido\_Paterno\_Apoderado, Apellido\_Materno\_Apoderado, Id\_Madre
  + **Tabla capacitación:**  
     Id\_Capacitacion, Id\_Madre, Higiene\_de\_Manos, Tecnica\_de\_Lactancia\_Materna, Higiene\_de\_Mamas,  
     Tecnica\_de\_Extraccion\_de\_Leche\_Extrahospitalaria, Tecnica\_de\_Extraccion\_de\_Leche\_Intrahospitalaria,  
     Masaje\_de\_Mamas, Lavado\_de\_Mamas
  + **Tabla paciente:**  
     Id\_Paciente, Nombre\_Paciente, Apellido\_Paterno\_Paciente, Apellido\_Materno\_Paciente, Fecha\_Nacimiento\_Paciente,  
     Sexo, Edad\_Gestacional, Peso, Talla, Id\_Cuna, Id\_Diagnostico\_Paciente
  + **Tabla pase\_de\_visita:** Id\_Pase\_Visita, Id\_Paciente, Id\_Cuna, Fecha\_Pase, Observaciones
  + **Tabla pasteurizada\_dispensada:**  
     Id\_Pasteurizada\_Dispensada, Id\_Paciente, Codigo\_Leche, Cantidad
  + **Tabla registro\_formula:**  
     Codigo\_leche\_formula, Tipo\_leche, Cantidad\_formula, Kcal
  + **Tabla registro\_leche\_cruda:**  
     Id\_Leche\_Cruda, Cantidad, Hora, Id\_Madre
  + **Tabla registro\_leche\_cruda\_donadora:**  
     Id\_Leche\_Cruda, Cantidad, Hora, Id\_Donadora, Fecha
  + **Tabla registro\_leche\_pasteurizada:**  
     Codigo\_Leche, Tipo\_Leche, Cantidad\_Leche, Kcal, Crema, Grasa, a\_dornix, Contenido\_Energetico
  + **Tabla reporte\_paciente:**  
     Id\_Reporte\_Paciente, Leche\_Autologa, LDM, Leche\_Pasteurizada, Leche\_Formula, Fecha, Id\_Paciente
  + **Tabla slot\_config:**  
     id, patient\_id, fecha, config\_data
  + **Tabla usuarios:**  
     Id\_Usuario, Profesion, Nombre, Apellido, Dni, Contrasenia,  
     Permiso\_Registrar\_Donadora, Permiso\_Registrar\_Paciente, Permiso\_Dispensacion,  
     Permiso\_Reportes, Permiso\_Gestion\_Usuarios, Permiso\_Cunas

# 4. Integraciones Externas (Opcionales)

* **Almacenamiento de archivos:** Reforzamiento del sistema del hospital para que se cuente con un controlado acceso restringido.
* **Autenticación externa (opcional):** OAuth2 con Google
* **Monitoreo/Logs:** Uso de Papertrail (SolarWinds) para la captura de errores y análisis.
* **Backups y replicación:** Servicio de backup con retención definida por el hospital.

# 5. Seguridad

* Encriptación de contraseñas
* Registro de accesos.
* Control de versiones y gestión de releases con entornos: desarrollo → staging → producción.
* Revisiones de seguridad (pruebas de aceptacion)antes de pasar a producción.
* Políticas de contraseñas y expiración de tokens JWT con refresh tokens.

# 6. Escalabilidad y Despliegue

Posibles problemas que pueden surgir durante el desarrollo.

* **Frontend**: Desplegado Nginx o integrado en el mismo servidor.
* **Backend**: Deploy integrado en el mismo servidor.
* **Base de Dato**s: MySql gestionada en una instancia dedicada del servidor
* La arquitectura permite escalar horizontalmente el backend y separar la base de datos en instancias dedicadas

# 7. Conclusiones

La arquitectura que se sugirió para AYNI fue creada con el propósito de satisfacer las necesidades tanto funcionales como no funcionales, asegurando la escalabilidad, la seguridad y la disponibilidad requeridas para un sistema de administración clínica.

Utilizando una arquitectura por capas (MySQL para la persistencia, Spring Boot para la lógica empresarial y Angular para la presentación), se logra una división de responsabilidades muy definida, lo cual hace más fácil el mantenimiento y garantiza que el sistema sea ampliable en versiones posteriores.

La confidencialidad de los datos sensibles, la auditoría de las acciones y la trazabilidad de las operaciones son elementos fundamentales que fortalecen la confianza y fiabilidad del sistema en el Banco de Leche Humana.