



INFORME DE PROYECTO FINAL

I. PORTADA

Tema:	Requerimientos y criterios del proyecto final
Unidad de Organización Curricular:	PROFESIONAL
Nivel y Paralelo:	5 A
Alumnos participantes:	Analuiza Castillo Jimmy Sebastián Gordillo Guevara Luis Josué Manobanda Chango Ana Patricia Peñaloza Narváez Johnny Alexander
Asignatura:	Sistemas de Base de Datos Distribuidos
Docente:	Ing. José Caiza

II. INFORME

1. Introducción

1.1 Propósito

Este documento especifica los requisitos funcionales y no funcionales para el desarrollo de un sistema hospitalario distribuido que implementa arquitectura MVC y bases de datos distribuidas avanzadas, permitiendo la gestión integral de múltiples centros médicos.

1.2 Alcance

El sistema es una aplicación web distribuida que gestiona información hospitalaria mediante APIs RESTful, con replicación y sincronización de datos en bases de datos distribuidas, interfaces de administración y gestión médica, desplegada en infraestructura cloud con contenedores Docker.

1.3 Definiciones, Acrónimos y Abreviaturas

- **MVC:** Modelo-Vista-Controlador
- **API:** Interfaz de Programación de Aplicaciones
- **REST:** Transferencia de Estado Representacional
- **VM:** Máquina Virtual
- **SGBD:** Sistema Gestor de Bases de Datos
- **CI/CD:** Integración Continua/Despliegue Continuo
- **VNet:** Red Virtual
- **NSG:** Grupos de Seguridad de Red

1.4 Referencias

- IEEE Std 830-1998 - Estándar para Especificación de Requisitos Software
- Documentación del Proyecto - Universidad Técnica de Ambato

1.5 Visión General

El documento organiza los requisitos en funcionales, no funcionales, restricciones de diseño, características de usuarios y atributos de calidad.

2. Descripción General



2.1 Perspectiva del Producto

Sistema cliente-servidor distribuido con backend en microservicios, frontend en frameworks modernos, bases de datos distribuidas replicadas, desplegado en Azure con contenedores Docker y gestión mediante Azure DevOps.

2.2 Funciones del Producto

- Gestión distribuida de centros médicos
- Administración de médicos, especialidades y empleados
- Registro y seguimiento de consultas médicas
- Replicación y sincronización de datos entre nodos
- Interfaces web para administración y uso hospitalario
- APIs RESTful documentadas

2.3 Características de los Usuarios

- **Administrador del Sistema:** Gestión global de centros médicos y configuración
- **Administrador de Centro Médico:** Gestión local de médicos, empleados y especialidades
- **Médico:** Registro y consulta de consultas médicas
- **Personal Administrativo:** Gestión de información del centro

2.4 Restricciones

- Debe implementar arquitectura MVC
- Uso de bases de datos distribuidas con replicación
- Despliegue en Azure con máquinas virtuales
- Implementación de contenedores Docker para nodos distribuidos
- Gestión del proyecto mediante Azure DevOps
- APIs RESTful documentadas con Postman.
- Comunicación segura entre nodos mediante VNet y NSG

2.5 Suposiciones y Dependencias

- Los centros médicos tienen conectividad a internet estable
- Azure DevOps disponible para gestión del proyecto
- Servicios cloud de Azure operativos
- Personal técnico capacitado en administración de infraestructura cloud

3. Requisitos Específicos

3.1 Requisitos Funcionales

3.1.1 Gestión de Infraestructura



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL
CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN
CICLO ACADÉMICO: AGOSTO 2025 – ENERO 2026



Código	Descripción
RF001	El sistema debe permitir la creación y configuración de al menos 3 máquinas virtuales en Azure
RF002	El sistema debe configurar redes virtuales (VNet) y grupos de seguridad (NSG) para comunicación segura
RF003	El sistema debe implementar SGBD (PostgreSQL) en cada VM
RF004	El sistema debe configurar replicación de bases de datos distribuidas

3.1.2 Gestión de Datos Distribuidos

Código	Descripción
RF005	El sistema debe implementar fragmentación horizontal de datos entre nodos
RF006	El sistema debe implementar fragmentación vertical de datos entre nodos
RF007	El sistema debe soportar replicación activa/pasiva de datos
RF008	El sistema debe soportar replicación síncrona/asíncrona
RF009	El sistema debe configurar nodos distribuidos en contenedores Docker

3.1.3 Backend y APIs (Laravel)

Código	Descripción
RF010	El sistema debe implementar arquitectura MVC en el backend
RF011	El sistema debe exponer APIs RESTful para operaciones CRUD de centros médicos
RF012	El sistema debe exponer APIs RESTful para operaciones CRUD de médicos
RF013	El sistema debe exponer APIs RESTful para operaciones CRUD de especialidades
RF014	El sistema debe exponer APIs RESTful para operaciones CRUD de empleados
RF015	El sistema debe exponer APIs RESTful para operaciones CRUD de consultas médicas
RF016	El sistema debe documentar APIs con Postman.

3.1.4 Frontend y Interfaces (React)



Código	Descripción
RF017	El sistema debe proporcionar interfaz web para administración de centros médicos
RF018	El sistema debe proporcionar interfaz web para gestión hospitalaria de consultas
RF019	Las interfaces deben consumir APIs RESTful de forma segura
RF020	El sistema debe implementar autenticación y control de acceso por centro médico

3.1.5 Despliegue y Operación

Código	Descripción
RF021	El sistema debe gestionarse mediante Azure DevOps con control de versiones
RF022	El sistema debe implementar pipelines CI/CD para despliegue automático
RF023	El sistema debe configurar Azure Load Balancer para alta disponibilidad
RF024	El sistema debe incluir pruebas de integración y funcionales

3.2 Requisitos No Funcionales

Código	Descripción
RNF001	El sistema debe garantizar disponibilidad del 99.9% para servicios críticos
RNF002	La replicación de datos debe mantener consistencia eventual entre nodos
RNF003	El tiempo de respuesta de APIs debe ser menor a 2 segundos
RNF004	El sistema debe soportar al menos 25 usuarios concurrentes
RNF005	Las interfaces web deben ser responsive y compatibles con navegadores modernos
RNF006	El sistema debe implementar autenticación segura y autorización por roles
RNF007	La comunicación entre nodos debe ser cifrada mediante SSL/TLS
RNF008	El sistema debe permitir escalado horizontal de nodos
RNF009	La documentación técnica debe incluir diagramas de arquitectura y flujos de datos

3.3 Interfaces Externas



3.3.1 Interfaces de Usuario

- **Interfaz de Administración:** Web-based, framework moderno (React)
- **Interfaz Hospitalaria:** Web-based, framework moderno (React)
- **Responsive Design:** Compatible con dispositivos móviles y tablets

3.3.2 Interfaces de Hardware

- Servidores cloud Azure (VMs)
- Dispositivos cliente con navegadores web modernos

3.3.3 Interfaces de Software

- APIs RESTful para integración
- Servicios de bases de datos distribuidas
- Contenedores Docker para orquestación
- Azure DevOps para CI/CD

3.4 Restricciones de Diseño

- Arquitectura MVC para separación de concerns
- Bases de datos distribuidas con estrategias de fragmentación definidas
- Uso de contenedores con Docker para consistencia de entornos

4. Atributos del Producto de Software

4.1 Fiabilidad

- Replicación de datos para tolerancia a fallos
- Mecanismos de recuperación ante caídas de nodos
- Backup automático de configuraciones y datos

4.2 Seguridad

- Autenticación JWT para APIs
- Cifrado de datos
- Control de acceso basado en roles

4.3 Usabilidad

- Interfaces intuitivas con curva de aprendizaje baja
- Navegación consistente entre módulos
- Feedback visual para operaciones de usuario
- Documentación de usuario accesible

4.4 Eficiencia

- Estrategias de caché para datos frecuentes



- Balanceo de carga para distribución de peticiones
- Consultas optimizadas a bases de datos distribuidas

4.5 Mantenibilidad

- Código modular con documentación adecuada
- Logs estructurados para monitoreo

4.6 Portabilidad

- Contenedores Docker para independencia de plataforma