



INFORME

I. PORTADA

Tema:	Arquitectura MVC y Bases de Datos Distribuidas Avanzadas.
Unidad de Organización Curricular:	PROFESIONAL
Nivel y Paralelo:	5to TI “A”
Alumnos participantes:	Ases Tiban Jeremy Damián Palate Moreta Kevin Damián Poveda Gómez William Alberto Pullupaxi Chango Daniel
Asignatura:	Sistemas De Bases De Datos Distribuidas
Docente:	Ing. José Caiza, Mg.

II. INFORME

2.1 Objetivos

General:

Planificar el desarrollo una aplicación distribuida que utilice la arquitectura MVC para gestionar la información de un sistema hospitalario integrado por múltiples centros médicos. La solución debe exponer servicios mediante API REST ful, garantizar la replicación y sincronización de datos en bases de datos distribuidas y enfrentar desafíos avanzados como la selección del modelo de distribución, configuración en contenedores y esquemas de replicación.

Específicos:

- Investigar las tecnologías necesarias para el desarrollo del proyecto.
- Definir y Acordar la Arquitectura Técnica Completa para asegurar que todo el equipo tiene una visión clara sobre cómo se implementarán.
- Crear un Plan de Iteración Inicial para que el equipo empiece a trabajar en los elementos de mayor valor primero.

•

2.2 Modalidad

Presencial

2.3 Instrucciones

- Definición y Diseño
- Configuración de Azure DevOps y Agile

2.4 Listado de equipos, materiales y recursos

Listado de equipos y materiales generales empleados en la guía práctica:

- Herramientas de Diseño (Draw.io)
- Herramientas de Desarrollo y Programación (IDEs)
- Plataformas en la Nube y Bases de Datos (PostgreSQL)
- Herramientas de Gestión y Automatización (DevOps)
- Herramientas de control de versiones (GitHub)

TAC (Tecnologías para el Aprendizaje y Conocimiento) empleados en la guía práctica:

- ☐ Plataformas educativas
- ☐ Simuladores y laboratorios virtuales
- ☐ Aplicaciones educativas



- ☐ Recursos audiovisuales
☐ Gamificación
☒ Inteligencia Artificial
Otros (Especifique): _____

2.5 Actividades desarrolladas

INTRODUCCIÓN

En el ámbito de la salud, la gestión eficiente de información sobre múltiples centros médicos representa un desafío tecnológico constante. Este proyecto desarrolla un sistema hospitalario distribuido que implementa arquitectura MVC y bases de datos distribuidas para garantizar disponibilidad, escalabilidad y consistencia de datos (Atomicamente). La solución utiliza PostgreSQL con replicación maestro-esclavo, Node.js para APIs RESTful y React en el frontend, desplegándose en Azure. Se implementa fragmentación horizontal para consultas médicas locales mientras se centralizan datos maestros, optimizando así el rendimiento y la confiabilidad del sistema.

- **Definición y Diseño**

ESQUEMA GENERAL

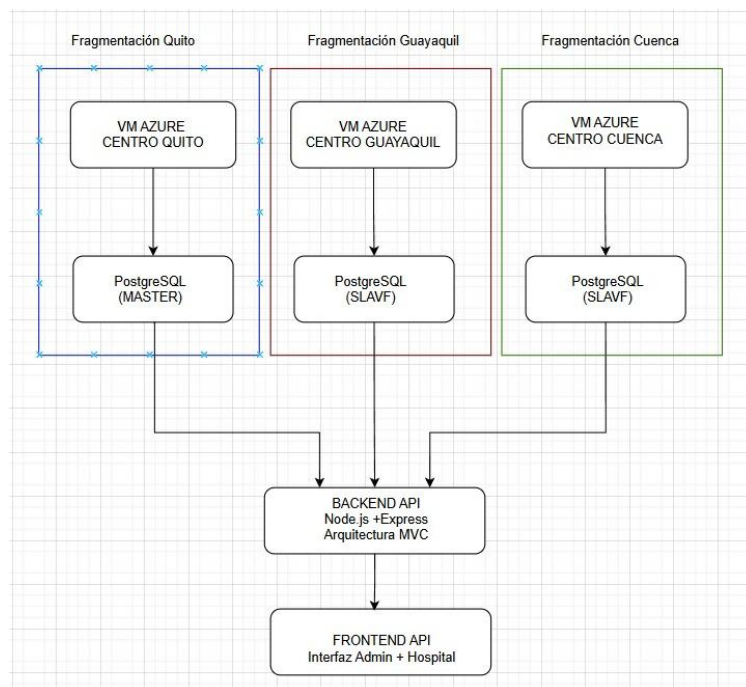


Fig. 1 Diagrama de Arquitectura

El diagrama de la **Fig.1** ilustra una arquitectura distribuida para el sistema hospitalario, enfocada en la fragmentación y replicación. La VM Centro Quito actúa como el nodo principal (PostgreSQL MASTER), mientras que las VMs Guayaquil y Cuenca operan como nodos de réplica (PostgreSQL SLAVE), implementando una Fragmentación Horizontal para manejar datos locales en cada sitio. Todos estos nodos se conectan a un Backend API monolítico desarrollado con Node.js/Express bajo la Arquitectura MVC, el



cual gestiona la lógica de acceso distribuido y expone los servicios a una interfaz de usuario (Frontend API) separada para la gestión administrativa y hospitalaria.

ESQUEMA DE TABLAS

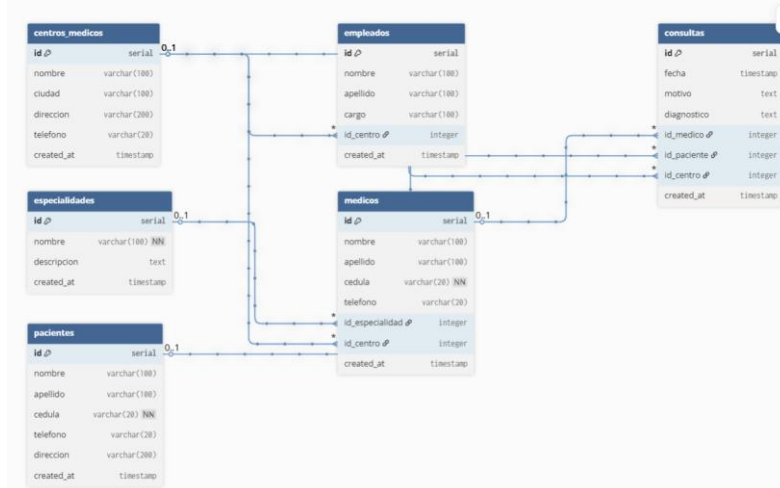


Fig. 2: Esquema de tablas centralizada

En la **Fig.2** se observa una base de datos relacional, la cual servirá como base para la implementación de nuestra base de datos distribuida aplicando los conceptos aprendidos de fragmentación, transparencia y atomicidad de datos.

TECNOLOGIAS A IMPLEMENTAR

Backend y API

Lenguaje: JavaScript con Node.js

Framework: Express.js

Arquitectura: MVC (Modelo vista controlador)

Documentación API: Swagger/OpenAPI

Autenticación: JWT Tokens

El proyecto utilizará Node.js/Express porque será rápido y eficiente para construir la API, permitiendo manejar muchas conexiones concurrentes de los centros médicos. Además, la Arquitectura MVC permitirá mayor organización para la lógica del proyecto.

Base de Datos Distribuida

SGBD: PostgreSQL (Homogéneo)

Modelo: Híbrido (Centralizado + Distribuido)

Replicación: Maestro-Esclavo (Pasiva) + Asíncrona

Fragmentación: Horizontal por ciudad/hospital

Configuración: 3 nodos (1 principal + 2 secundarios)

La Base de Datos Distribuida utilizará PostgreSQL por ser robusto. Se implementará un Modelo Híbrido para mejorar las operaciones, usando Fragmentación Horizontal para dividir las consultas por hospital. Y finalmente, la Replicación Asíncrona se establecerá para garantizar que los datos sean consistentes y que el sistema tenga alto rendimiento sin esperar por cada réplica.

Frontend

Tecnología: React.js

Interfaz Admin: Gestión de centros, médicos, empleados



Interfaz Hospital: Gestión de consultas médicas

El Frontend será desarrollado con React.js para garantizar una interfaz moderna y eficiente.

Infraestructura y DevOps

Cloud: Microsoft Azure

Virtualización: 3 Máquinas Virtuales (VMs)

Control de Versiones: Git + Azure DevOps

CI/CD: Azure Pipelines

La infraestructura se alojará en Microsoft Azure, utilizando tres Máquinas Virtuales (VMs) para simular los centros médicos. La gestión completa del proyecto se hará mediante Azure DevOps y Git para el control de versiones.

CONFIGURACIÓN DE BASE DE DATOS DISTRIBUIDA

Modelo de Distribución

Datos Maestros: Almacenados en nodo central (Quito) y replicados

Datos Locales: Fragmentados por ciudad en cada VM

Entidades Centralizadas: Médicos, Especialidades, Empleados

Entidades Fragmentadas: Consultas Médicas (por centro)

Esquema de Replicación

Nodo Principal (Quito) → Escritura/Lectura

Replicación Asincrónica

Nodo Secundario (Guayaquil) → Solo Lectura

Nodo Secundario (Cuenca) → Solo Lectura

Fragmentación Implementada

Horizontal: Tabla 'consultas' dividida por 'id_cen_med'

Vertical: No aplica (mantenemos simplicidad)

Híbrida: No aplica (complejidad innecesaria)

- **Configuración de Azure DevOps y Agile**

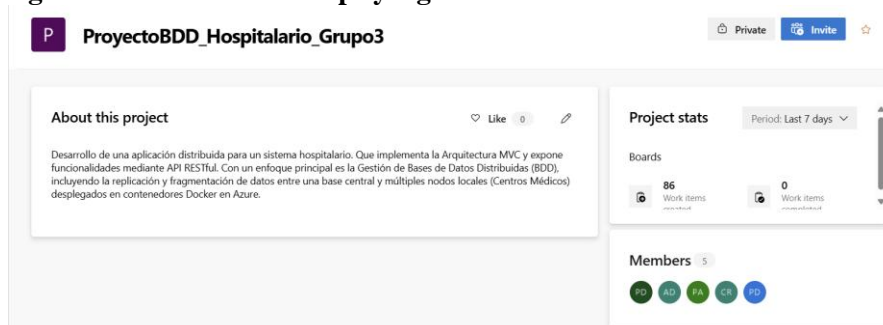


Fig. 3 Pagina inicial del proyecto en Azure DevOps



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



En la **Fig. 3** se muestra la inicialización del proyecto junto a los miembros.

ID	Title	Assigned To	Status	Area/Path	Tags	Comments	Activity Date
7	Configuración de Azure y VMs	Palate Moreta...	To Do	ProyectoRDO_Hospitalario_Gr...	Infraestructura		05/10/2025 23:11:25
8	Instalación y Configuración de PostgreSQL	Palate Moreta...	To Do	ProyectoRDO_Hospitalario_Gr...	Infraestructura		05/10/2025 23:11:34
9	Configuración de Red y Seguridad	Poveda Gome...	To Do	ProyectoRDO_Hospitalario_Gr...	Infraestructura		05/10/2025 23:12:13
10	Configuración Inicial del Proyecto	Poveda Gome...	To Do	ProyectoRDO_Hospitalario_Gr...	Backend		05/10/2025 23:13:12
48	Modelos de Base de Datos	Palate Moreta...	To Do	ProyectoRDO_Hospitalario_Gr...	Backend		05/10/2025 23:13:22
49	Controladores y Rutas	Poveda Gome...	To Do	ProyectoRDO_Hospitalario_Gr...	Backend		05/10/2025 23:13:29
50	Lógica de Base de Datos Distribuida	Poveda Gome...	To Do	ProyectoRDO_Hospitalario_Gr...	Backend		05/10/2025 23:13:35
51	Documentación API	Palate Moreta...	To Do	ProyectoRDO_Hospitalario_Gr...	Backend		05/10/2025 23:13:44
15	Configuración Proyecto React	Pullpaxi Cha...	To Do	ProyectoRDO_Hospitalario_Gr...	Frontend		05/10/2025 23:14:57
20	Módulo de Administración	Ases Tiban Jer...	To Do	ProyectoRDO_Hospitalario_Gr...	Frontend		05/10/2025 23:15:25
34	Módulo de Hospitales	Pullpaxi Cha...	To Do	ProyectoRDO_Hospitalario_Gr...	Frontend		05/10/2025 23:15:40
40	Integración y Servicios	Ases Tiban Jer...	To Do	ProyectoRDO_Hospitalario_Gr...	Frontend		05/10/2025 23:15:49
64	Pruebas de Replicación de Bases de Datos	Ases Tiban Jer...	To Do	ProyectoRDO_Hospitalario_Gr...	Desarrollo y pruebas		05/10/2025 23:16:45
55	Azure DevOps	Pullpaxi Cha...	To Do	ProyectoRDO_Hospitalario_Gr...	Desarrollo y pruebas		05/10/2025 23:18:01
70	Pruebas del Backend y APIs	Pullpaxi Cha...	To Do	ProyectoRDO_Hospitalario_Gr...	Desarrollo y pruebas		05/10/2025 23:18:28
79	Pruebas del Frontend y Interfaces	Ases Tiban Jer...	To Do	ProyectoRDO_Hospitalario_Gr...	Desarrollo y pruebas		05/10/2025 23:18:38

Fig. 4: Asignación de tareas

En la **Fig. 4** se observa parte de las tareas que se llevarán durante la realización del proyecto además de servir como método de inicio, este permitirá tener en cuenta el avance y la implicación que se tiene durante su desarrollo.

2.6 Resultados Obtenidos

Tras la planificación del proyecto, se espera obtener un sistema hospitalario distribuido completamente funcional, que incluirá: una base de datos PostgreSQL configurada con replicación entre los tres nodos en Azure, garantizando la sincronización de datos; un backend con API RESTful documentada que permita la gestión centralizada y distribuida de la información; y un frontend unificado en React.js para las operaciones administrativas y médicas. El resultado final será una plataforma escalable y de alta disponibilidad, automatizada mediante CI/CD, que demuestre la viabilidad de la arquitectura MVC aplicada a un entorno de bases de datos distribuidas.

2.7 Habilidades blandas empleadas en la práctica

- ☐ Liderazgo
- ☒ Trabajo en equipo
- ☐ Comunicación asertiva
- ☐ La empatía
- ☐ Pensamiento crítico
- ☐ Flexibilidad
- ☐ La resolución de conflictos
- ☐ Adaptabilidad
- ☐ Responsabilidad

2.8 Conclusiones

La planificación del sistema hospitalario distribuido bajo arquitectura MVC y bases de datos distribuidas define una solución robusta y escalable para la gestión de múltiples centros médicos. La propuesta integra APIs RESTful, replicación de datos en PostgreSQL y fragmentación horizontal, lo que garantizará alta disponibilidad y consistencia de la información. Este diseño sienta las bases técnicas para un desarrollo eficiente y una futura expansión de la infraestructura médica en un entorno cloud con Azure.



2.9 Referencias

- [1] “Tutorial de DevOps”, *Microsoft.com*. [En línea]. Disponible en: <https://azure.microsoft.com/es-mx/solutions/devops/tutorial>. [Consultado: 06-oct-2025].
- [2] “Planificación y realización de un seguimiento del trabajo en Azure Boards”, *Microsoft.com*. [En línea]. Disponible en: <https://learn.microsoft.com/es-es/azure/devops/boards/get-started/plan-track-work?view=azure-devops&tabs=agile-process>. [Consultado: 06-oct-2025].

2.10 Anexos

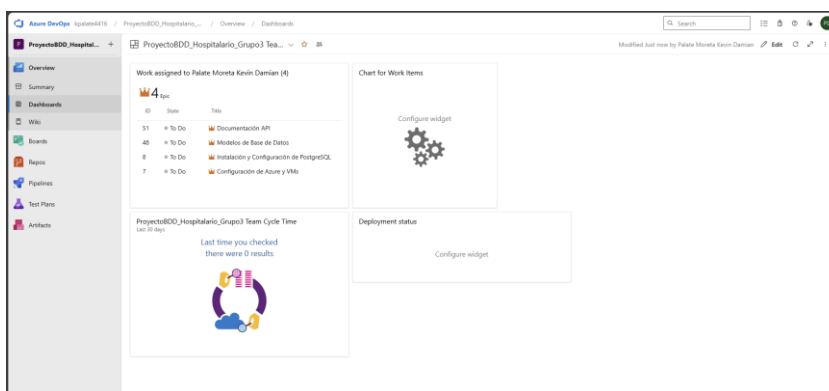


Fig. 6 Dashboard

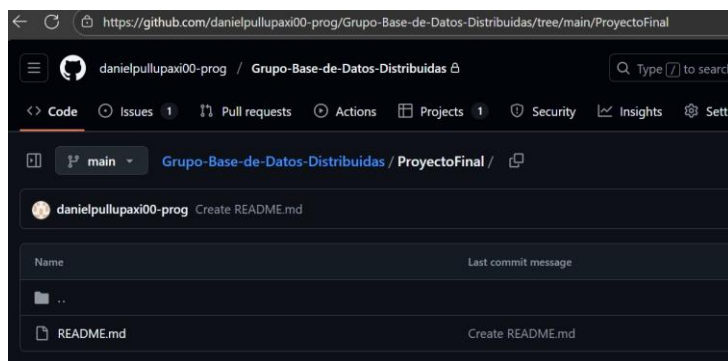


Fig. 5 Repositorio en GitHub

