

Aplicación de gestión hospitalaria

Informe N°1

Desarrollo a cargo de:

Juan Diego Pérez Navarro
Diederik Antonio Montaña Burbano
Alan Castellón García

Arquitectura de Software – 1589

Profesor Jairo Enrique Serrano Castañeda

Universidad Tecnológica de Bolívar
Facultad de ingeniería en sistemas y computación
Cartagena de Indias D.T. y C., Bolívar

2024

Tabla de contenido

Caso de estudio	3
Arquitectura del software	4
Arquitectura centrada en los datos	4
Arquitectura de microservicios	4
Tareas actuales	6
Tareas en progreso	6
Actualización del SRS	6
Construcción de la interfaz visual	6
Desarrollo de API's	6
Tareas en revisión	6
Creación del SRS	7
Desarrollo del mockup de la interfaz visual	7
Tareas completadas	7
Elección de motores de bases de datos	7
Construcción del modelo relacional	7
Construcción de diagrama E-R	7
Revisión de infraestructura Docker	7
Construcción del esquema de API	8

Caso de estudio

Un hospital en constante expansión cuenta con una capacidad de más de 300 camas y atiende alrededor de 1.000 pacientes al día. Actualmente, el hospital almacena las historias clínicas de los pacientes en archivos físicos que se guardan en el área de Archivo. Cuando un médico necesita consultar la historia de un paciente, debe solicitar el archivo a esta área y esperar a que le sea entregado. Esto genera una serie de problemas, tales como: demoras en la atención, ya que se debe esperar la entrega física de la historia clínica, que en promedio se tarda 20 minutos; riesgos de extravío de historias clínicas, lo cual sucede aproximadamente en el 5% de las solicitudes; dificultades para compartir en tiempo real información de un paciente entre médicos de diferentes especialidades; imposibilidad de generar reportes y estadísticas a partir de los datos de las historias clínicas.

El hospital desea desarrollar un sistema de información que permita digitalizar y centralizar las historias clínicas de los pacientes, de forma que estén disponibles en línea para los médicos autorizados y se pueda obtener información para la gestión hospitalaria.

El producto mínimo viable que se requiere es: gestionar historia clínica electrónica de cada paciente, generar reportes de indicadores tales como; porcentaje de ocupación hospitalaria, promedios de estancia por servicio, cantidad de admisiones y altas por servicio.

Los datos que se le solicitarán a los pacientes ingresados son los siguientes:

- Datos del paciente (documento, nombre, sexo, fecha nacimiento)
- Signos vitales (presión arterial, temperatura, saturación O₂, frecuencia respiratoria)
- Notas de evolución
- Imágenes diagnósticas
- Resultados de exámenes de laboratorio

Los beneficios que se logran con la implementación del software van desde el acceso rápido a la información de los pacientes hasta una mejor coordinación entre especialidades médicas, reducción del riesgo de pérdida de historias clínicas y facilidad en la toma de decisiones gerenciales basada en datos.

Arquitectura del software

Teniendo en cuenta el contexto del cliente, resultó fundamental comprender sus necesidades y prioridades para la selección de la arquitectura y la futura implementación de la aplicación. Con este objetivo, fueron priorizados los siguientes principios:

1. **Disponibilidad:** una alta disponibilidad resulta fundamental, teniendo en cuenta que el funcionamiento eficiente del hospital dependerá completamente del software, este debe ser accesible durante la mayor cantidad de tiempo posible.
2. **Resiliencia:** el sistema debe ser capaz de recuperarse rápidamente de fallos de cualquier tipo y mantener su funcionalidad a pesar de caídas parciales o problemas externos tales como los errores de red o apagones imprevistos.
3. **Seguridad y confidencialidad:** comprendiendo que se trabaja constantemente con datos sensibles de un alto volumen de pacientes, resulta prioritario proteger la integridad de la información y mantener la confidencialidad de los datos.
4. **Escalabilidad:** el flujo de pacientes crece constantemente y la cantidad de información almacenada nunca disminuye, por lo que es necesario que el software pueda crecer fácilmente según las necesidades del hospital.
5. **Monitoreo y pruebas:** es necesario que cada sección de la aplicación pueda ser accedida, monitoreada y probada de forma individual, con el objetivo de detectar vulnerabilidades y prevenir la propagación de errores.
6. **Análisis de datos:** la información almacenada debe ser analizable para la producción de indicadores fundamentales para el funcionamiento del hospital.

Teniendo en cuenta estas premisas, se decidió la creación de una arquitectura de microservicios centrada en los datos.

Arquitectura centrada en los datos

Una arquitectura centrada en los datos prioriza la información como el activo más valioso dentro del software, lo que concuerda a cabalidad con el caso abordado, en el que la información de los pacientes funge como activo principal para la toma de decisiones corporativas y el análisis del funcionamiento del hospital, además de su funcionamiento básico.

Asegurar la permanencia y accesibilidad de los datos es fundamental para que el personal hospitalario sea capaz de llevar a cabo sus actividades con normalidad y eficiencia, por lo que una arquitectura centrada en los datos resulta la mejor de las opciones disponibles.

Arquitectura de microservicios

La arquitectura centrada en microservicios permite la construcción de aplicaciones flexibles, resilientes y altamente escalables que además son fáciles de monitorear y probar, todas estas

propiedades ayudan a maximizar la disponibilidad del software, abarcando cuatro de las seis premisas fundamentales para el desarrollo.

Actualmente, dentro de la arquitectura se comprenden los siguientes microservicios:

Gestión de usuarios: en esta sección se almacena y gestiona la información fundamental de cada usuario (nombre de usuario, contraseña, nombre del paciente, fecha de nacimiento, etc.). Cumple todas las funciones relacionadas con los usuarios, incluida la creación de todos los tipos (administrador, paciente y médico), eliminación, actualización de datos, etc.

Autenticación: en este servicio se gestionan los tokens de sesión de los usuarios, que les permiten o no acceder a ciertas funcionalidades e información, protegiendo así la privacidad de los pacientes al prevenir los accesos indeseados.

Análisis de datos: con este microservicio se analizan los datos almacenados para la producción de indicadores relevantes para la toma de decisiones gerenciales del hospital.

Con estos criterios, actualmente se planea que la arquitectura de software acaba por presentar la siguiente estructura:

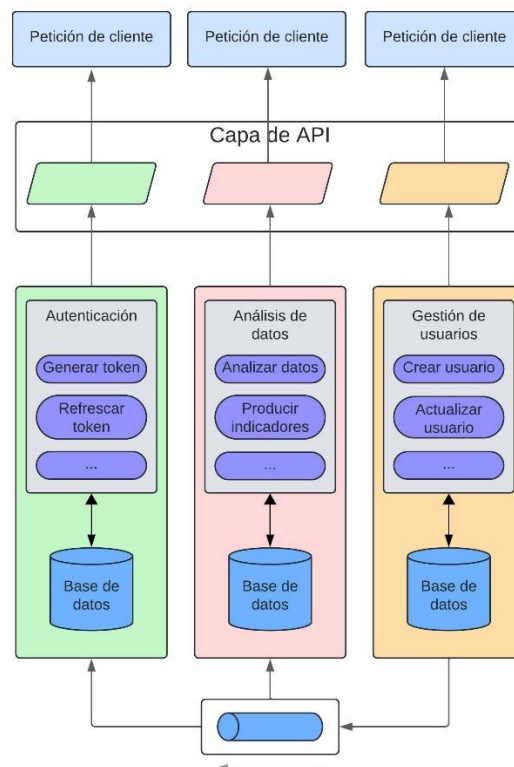


Imagen 1: boceto de la arquitectura de software.

Tareas actuales

En este momento, el proyecto se encuentra en desarrollo activo y posee un total de 22 tareas, de las cuales 12 se encuentran en espera, 4 en progreso, 2 en revisión y 5 completadas.

Tareas en progreso

Actualmente se está desarrollando el código del software a nivel de backend, frontend y desarrollo de API's, además de múltiples actualizaciones de documentación. Se encuentran en desarrollo las siguientes tareas:

Actualización del SRS

Se está actualizando la documentación de requerimientos del software en vista de que su primera versión ha quedado obsoleta a razón de múltiples cambios en la elección de entornos de desarrollo, bases de datos, infraestructura de software y otras propiedades. Además, se busca crear una versión más completa que comprenda también requerimientos de usuario y del sistema.

Esta tarea se encuentra a cargo de Diederik Montaña.

Construcción de la interfaz visual

Se comienza con el desarrollo del frontend mediante la construcción de todas las interfaces visuales planteadas en el mockup de la aplicación. Esta tarea comprende única y exclusivamente la creación de las vistas, pues la integración de API's y otras actividades se encuentran comprendidas en otras tareas del repositorio.

Esta tarea se encuentra a cargo de Diederik Montaña

Desarrollo de API's

Las conexiones entre las bases de datos, el backend y el frontend en la arquitectura seleccionada se realizarán mediante múltiples API's, una para cada microservicio. Esta tarea comprende el desarrollo de todas las API's planteadas hasta el momento.

Esta tarea se encuentra a cargo de Juan Pérez.

Tareas en revisión

Algunas de las tareas ya realizadas se encuentran en estado de revisión, a la espera de que se confirme que cumplen con los requisitos y calidad necesarios para ser aceptadas en el proyecto final. Se encuentran en revisión las siguientes tareas:

Creación del SRS

Es la construcción de la primera versión del documento de especificación de requisitos del software, que ya se encuentra actualmente en el repositorio almacenado en la carpeta de documentación, nombrada “docs”.

Esta tarea fue realizada por Juan Pérez y Alan Castellón

Desarrollo del mockup de la interfaz visual

Comprende el maquetado de todas las vistas que se esperan necesarias para el funcionamiento de la aplicación, su objetivo es facilitar el desarrollo en código de las interfaces visuales.

Esta tarea fue realizada por Jesús Guzmán (un colaborador externo) y Diederik Montaña.

Tareas completadas

Actualmente 5 de las tareas del proyecto se encuentran completadas, al haber pasado el proceso de revisión y considerarse aptas para ser aceptadas dentro del proyecto final; estas tareas son:

Elección de motores de bases de datos

En esta tarea se eligieron los motores de bases de datos que serían utilizados para almacenar la información que produce y solicita cada uno de los microservicios, dividiendo los motores en uno SQL y otro NoSQL. En la primera categoría fue elegido el motor PostgreSQL, mientras que en la segunda se seleccionó MongoDB.

Esta tarea estuvo a cargo de Juan Pérez y Diederik Montaña.

Construcción del modelo relacional

Comprende la construcción del diagrama relacional, que describe las entidades dentro del sistema y la forma en que estas interactúan.

Esta tarea fue llevada a cabo por Juan Pérez.

Construcción de diagrama E-R

Comprende la construcción del diagrama entidad – relación, que describe las conexiones entre distintos actores dentro del sistema y sus propiedades, así como las llaves primarias y foráneas que establecen dichas relaciones.

Esta tarea fue llevada a cabo por Juan Pérez.

Revisión de infraestructura Docker

En esta tarea se realiza la investigación de las imágenes Docker disponibles y la selección de las necesarias para utilizar las tecnologías seleccionadas dentro de contenedores de Docker.

Esta tarea fue llevada a cabo por Alan Castellón.

Construcción del esquema de API

Comprende la consolidación y formalización de los métodos de interacción con la API, incluyendo sus puntos de acceso, operaciones disponibles, tipos de datos, formatos, puntos de acceso y otra información técnica fundamental para el desarrollo y uso de una API.

Esta tarea fue llevada a cabo por Juan Pérez.