Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación
Luis Yael Méndez Sánchez
Control de Calidad de Software
Arquitectura de Software de Proyecto
202240624 | Jhoel Boset Hernández Hernández
202256748 | Haciel Antionio Viveros Reyes
Otoño 2024

Contenido

Arquitectura de Software: Seguimiento de Dieta

Objetivo del Proyecto:

- Desarrollar una aplicación para facilitar el seguimiento de regímenes alimenticios personalizados para pacientes con problemas de ingesta alimenticia.
- Funcionalidades para pacientes y profesionales de salud.

Recolección de Requerimientos

Requerimientos Funcionales:

Para los Profesionales de Salud:

- **Registro del Régimen Alimenticio:**
    - Posibilidad de crear, actualizar y visualizar los planes alimenticios personalizados para cada paciente, basados en consumo diario o por tiempos específicos (desayuno, almuerzo, cena, etc.).
- **Estadísticas de Pacientes:**
    - Visualización de estadísticas de seguimiento de los pacientes, tanto individualmente como de forma agregada, permitiendo evaluar patrones de consumo y adherencia al plan.
    - Herramientas de análisis para hacer sugerencias basadas en el reporte en tiempo real de los pacientes.
- **Revisión y Ajuste de Porciones:**
    - Funcionalidad para revisar y corregir ajustes en los reportes de ingesta de los pacientes, asegurando que el seguimiento sea preciso y alineado con las recomendaciones dietéticas.
- **Comunicación con los Pacientes:**
    - Sistema de mensajería para la comunicación rápida con los pacientes, accesible desde la interfaz web.

Para los Pacientes:

- **Seguimiento del Régimen Alimenticio:**
    - Registro de la ingesta diaria de alimentos de manera sencilla desde la interfaz web o móvil, con opciones para reportar cambios o equivalencias en las porciones.
- **Visualización de Estadísticas:**
    - Acceso a estadísticas de seguimiento semanal y mensual, incluyendo calorías consumidas, recomendadas, porciones consumidas, excedentes, faltantes, y estimaciones de pérdida o ganancia de peso.
- **Corrección y Ajuste de Porciones:**
    - Permitir reportar cambios en la ingesta y recibir retroalimentación automática del sistema respecto a si estos cambios corresponden adecuadamente a su plan.
- **Comunicación con Profesionales de Salud:**
    - Acceso a la herramienta de comunicación rápida para resolver dudas o hacer consultas en tiempo real.

Requerimientos No Funcionales:

- **Seguridad y Privacidad:**
    - Cumplimiento con las normativas y estándares de seguridad y privacidad, como GDPR o HIPAA, para proteger la información personal y médica de los pacientes.
- **Fiabilidad y Recuperación de Errores:**
    - Mecanismos robustos para la recuperación de errores y la continuidad del servicio, asegurando la integridad de los datos en caso de fallas.
- **Disponibilidad y Accesibilidad:**
    - La aplicación debe estar disponible tanto en versión web como móvil, garantizando acceso constante para los usuarios.
- **Escalabilidad:**
    - La arquitectura del sistema debe soportar un número creciente de usuarios y datos sin comprometer el rendimiento.

Casos de Uso

Caso de Uso 1: Registrar Régimen Alimenticio

Actor Primario: Profesional de Salud

Objetivo: Permitir al profesional de salud registrar el régimen alimenticio personalizado para un paciente.

Precondiciones:

- El profesional de salud debe estar autenticado en la aplicación.
- El paciente debe estar registrado en el sistema.

Escenario Principal:

1. El profesional de salud selecciona un paciente de la lista de sus pacientes.
2. El sistema muestra el perfil del paciente y las opciones para gestionar su régimen alimenticio.
3. El profesional de salud ingresa los detalles del régimen alimenticio (consumo diario o por tiempos).
4. El sistema valida los datos ingresados y guarda el régimen alimenticio en el expediente del paciente.
5. El paciente puede consultar el régimen alimenticio desde su interfaz móvil o web.

Postcondiciones:

- El régimen alimenticio está disponible para consulta por parte del paciente en su perfil.

Caso de Uso 2: Reportar Ingesta Diaria

Actor Primario: Paciente

Objetivo: Permitir al paciente registrar la ingesta diaria de alimentos y reportar cualquier cambio o ajuste en las porciones.

Precondiciones:

- El paciente debe estar autenticado en la aplicación móvil o web.
- Debe existir un régimen alimenticio registrado por el profesional de salud.

Escenario Principal:

1. El paciente accede a la sección de seguimiento diario desde la interfaz móvil.
2. El sistema muestra el plan alimenticio recomendado para ese día.
3. El paciente registra los alimentos consumidos, ajustando porciones según lo necesario.
4. Si el paciente no está seguro de los cambios realizados, reporta el ajuste para revisión.
5. El sistema calcula las equivalencias de las porciones y actualiza el seguimiento diario.
6. El profesional de salud revisa y, si es necesario, corrige los ajustes al día siguiente.

Postcondiciones:

- El seguimiento diario del paciente se actualiza en su expediente electrónico.

Caso de Uso 3: Revisar y Ajustar Ingesta del Paciente

Actor Primario: Profesional de Salud

Objetivo: Permitir al profesional de salud revisar los ajustes reportados por el paciente en su ingesta diaria y realizar correcciones.

Precondiciones:

- El paciente ha reportado ajustes en su ingesta diaria.
- El profesional de salud debe estar autenticado en la interfaz web.

Escenario Principal:

1. El profesional de salud accede a la lista de reportes de ajustes de los pacientes.
2. El sistema muestra los ajustes reportados por cada paciente.
3. El profesional de salud revisa los detalles del ajuste y verifica las equivalencias con el régimen alimenticio.
4. Si es necesario, el profesional corrige el ajuste y guarda los cambios en el expediente del paciente.
5. El paciente recibe una notificación de los cambios realizados en su seguimiento.

Postcondiciones:

- El expediente del paciente se actualiza con los ajustes correctos.

Caso de Uso 4: Consultar Estadísticas de Seguimiento

Actor Primario: Paciente

Objetivo: Permitir al paciente consultar sus estadísticas de seguimiento semanal y mensual.

Precondiciones:

- El paciente debe estar autenticado en la aplicación móvil o web.

Escenario Principal:

1. El paciente accede a la sección de estadísticas desde la interfaz web o móvil.
2. El sistema muestra las estadísticas de calorías consumidas, recomendadas, porciones consumidas, faltantes, excedentes y seguimiento del día libre.
3. El paciente puede revisar estimaciones de pérdida o ganancia de peso, considerando su actividad física registrada.

Postcondiciones:

- El paciente tiene una vista clara de su progreso y adherencia al régimen alimenticio.

Caso de Uso 5: Comunicación entre Paciente y Profesional de Salud

Actor Primario: Paciente y Profesional de Salud

Objetivo: Facilitar la comunicación rápida y efectiva entre el paciente y el profesional de salud.

Precondiciones:

- Ambos actores deben estar autenticados en sus respectivas interfaces (web para profesionales y móvil para pacientes).

Escenario Principal:

1. El paciente envía un mensaje al profesional de salud a través de la interfaz móvil.
2. El sistema notifica al profesional de salud del nuevo mensaje al iniciar sesión o en tiempo real si ya está conectado.
3. El profesional de salud revisa el mensaje y responde desde la interfaz web.
4. El paciente recibe la respuesta en su aplicación móvil.

Postcondiciones:

- La comunicación se mantiene fluida y eficiente, permitiendo aclaraciones y ajustes oportunos en el seguimiento del paciente.

Caso de Uso 6: Autenticación y Seguridad

Actor Primario: Paciente y Profesional de Salud

Objetivo: Garantizar el acceso seguro a la aplicación para todos los usuarios.

Precondiciones:

- El usuario debe estar registrado en el sistema.

Escenario Principal:

1. El usuario abre la aplicación e ingresa sus credenciales de acceso (usuario y contraseña).
2. El sistema verifica las credenciales y aplica políticas de seguridad (como verificación en dos pasos, si aplica).
3. Si las credenciales son correctas, el usuario accede a su perfil y funcionalidades disponibles.
4. En caso de intentos fallidos, el sistema ofrece opciones de recuperación de cuenta.

Postcondiciones:

- El acceso al sistema está restringido y protegido, cumpliendo con los requerimientos de seguridad y privacidad.

Estilo Arquitectónico

Arquitectura de Microservicios:

Descripción:

- En este estilo arquitectónico, el sistema se descompone en pequeños servicios independientes que se comunican a través de APIs.
- Cada microservicio es responsable de una funcionalidad específica del sistema, como la gestión de usuarios, la gestión de regímenes alimenticios, la comunicación entre pacientes y profesionales, etc.
- Los microservicios son desarrollados, desplegados y escalados de forma independiente, lo cual facilita el mantenimiento y la evolución del sistema.

Ventajas:

- **Escalabilidad Independiente:** Permite escalar solo las partes del sistema que requieren más recursos, por ejemplo, la gestión de seguimiento de pacientes podría necesitar más escalabilidad que otras funcionalidades.
- **Despliegue Continuo:** Los equipos pueden trabajar y desplegar cambios en microservicios específicos sin afectar todo el sistema, lo cual es ideal para implementaciones ágiles y CI/CD (Integración y Despliegue Continuo).
- **Resiliencia:** Si un microservicio falla, no necesariamente afecta a los demás, mejorando la disponibilidad del sistema.
- **Tecnologías Heterogéneas:** Puedes usar diferentes tecnologías para diferentes microservicios, lo cual es útil si en el futuro decides introducir componentes que no encajen bien con Django o Angular.

Consideraciones:

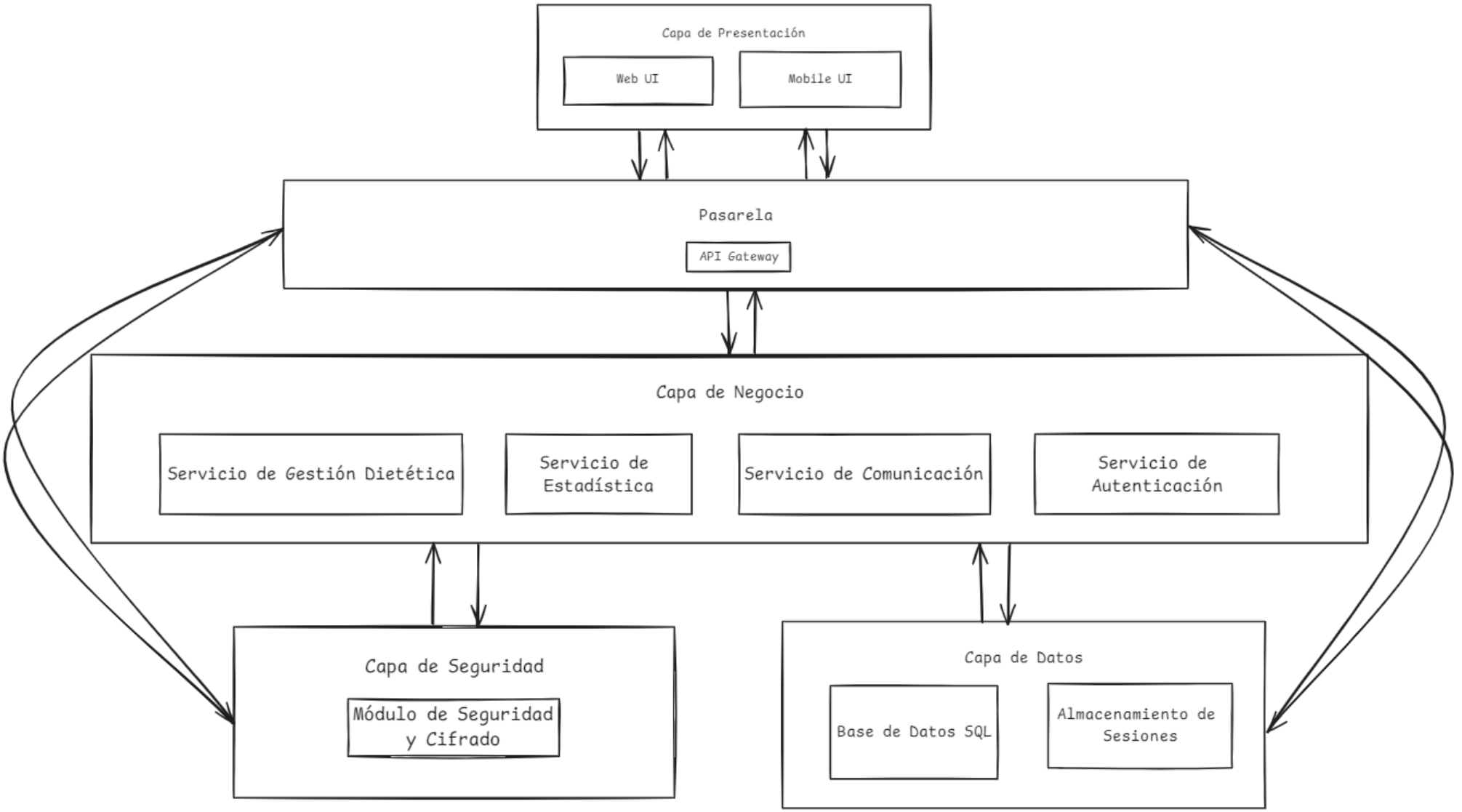
- Requiere una infraestructura más compleja para la orquestación y el despliegue, como Kubernetes para la gestión de contenedores.
- Necesidad de manejar la comunicación entre microservicios y asegurar la consistencia de los datos.

Conclusión:

La arquitectura de microservicios es una elección robusta y moderna para el desarrollo de sistemas complejos, como la aplicación destinada a la gestión de regímenes alimenticios y la comunicación entre pacientes y profesionales. Al dividir el sistema en servicios independientes, se logra una alta escalabilidad y resiliencia, permitiendo que cada microservicio crezca y evolucione de manera autónoma según las necesidades del sistema. Esto no solo facilita el mantenimiento y las actualizaciones continuas, sino que también habilita la adopción de diferentes tecnologías para optimizar cada funcionalidad específica.

Sin embargo, la implementación de microservicios también trae consigo desafíos, principalmente relacionados con la complejidad de la infraestructura y la gestión de la comunicación entre los servicios. Es fundamental contar con una orquestación adecuada y herramientas para asegurar la consistencia de los datos y la integridad del sistema en su conjunto. A pesar de estos retos, los beneficios en términos de escalabilidad, flexibilidad y capacidad de respuesta a los cambios superan las dificultades, haciendo de la arquitectura de microservicios una opción ideal para sistemas que buscan adaptarse rápidamente a las demandas del mercado y ofrecer una experiencia de usuario robusta y confiable.

Diseño de Capas: Arquitectura de Microservicios.



Capa de Presentación:

- **Web UI y Mobile UI:** Corresponde a las interfaces de usuario para profesionales y pacientes, respectivamente.
- **Pasarela:** Actúa como punto de entrada a los servicios, validando las solicitudes y enrutándolas a los microservicios correspondientes.
- **API Gateway:** Gestiona la comunicación entre la capa de presentación y los microservicios, ofreciendo un punto de control centralizado para las API.

Capa de Negocio:

- **Servicio de Gestión Dietética:** Se encarga de crear, actualizar y visualizar los planes alimenticios, así como de calcular las calorías y las porciones.
- **Servicio de Estadística:** Genera reportes y estadísticas de seguimiento para pacientes y profesionales.
- **Servicio de Comunicación:** Facilita la comunicación entre pacientes y profesionales a través de un sistema de mensajería.
- **Servicio de Autenticación:** Gestiona la autenticación y autorización de usuarios, asegurando que solo los usuarios autorizados puedan acceder a los datos y funcionalidades.

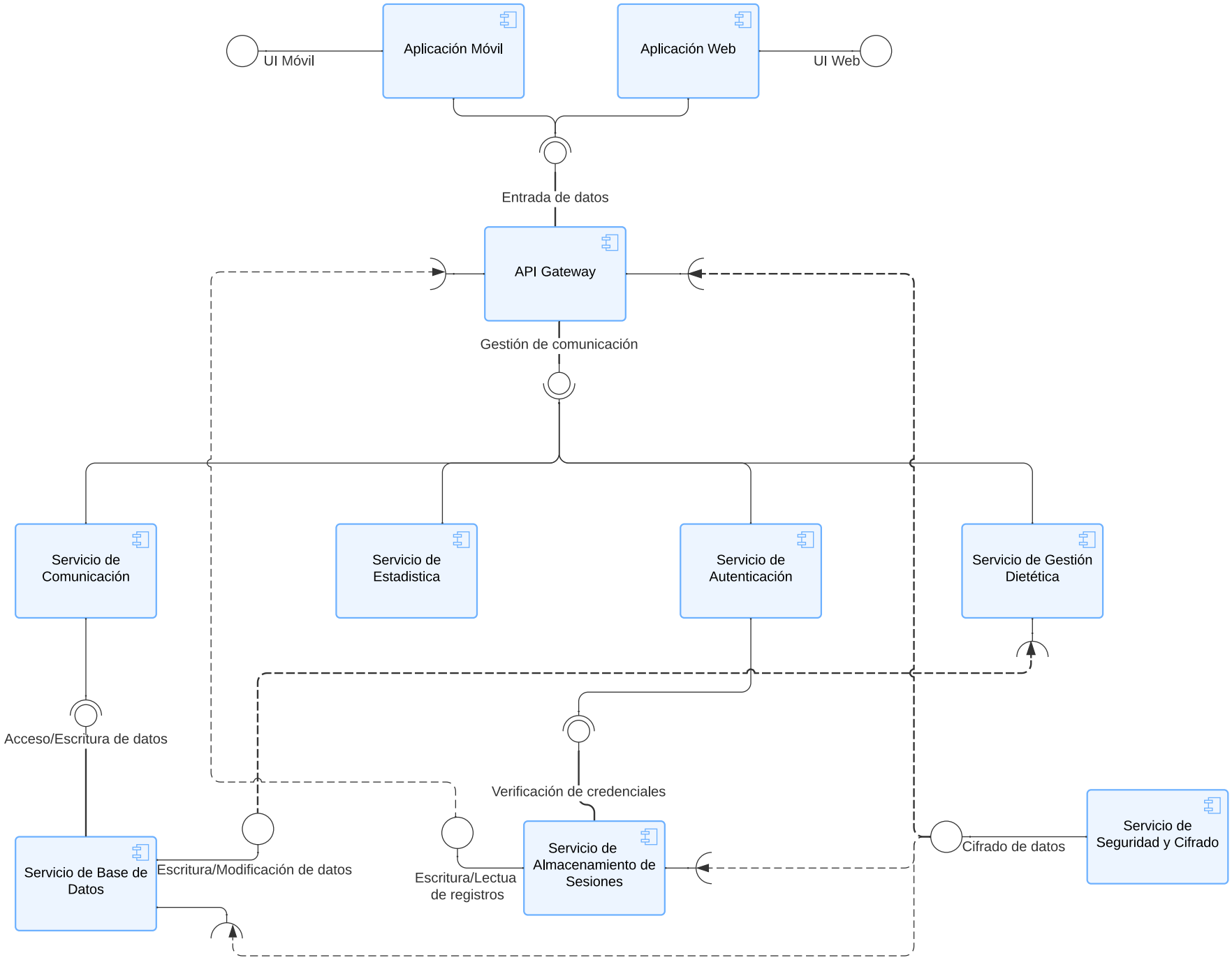
Capa de Seguridad:

- **Módulo de Seguridad y Cifrado:** Implementa mecanismos de seguridad para proteger los datos de los usuarios, como el cifrado de datos en reposo y en tránsito.

Capa de Datos:

- **Base de Datos SQL:** Almacena la información de los usuarios, los planes alimenticios, las estadísticas y los mensajes.
- **Almacenamiento de Sesiones:** Gestiona las sesiones de los usuarios para mantener el estado de la aplicación.

Diagrama de Componentes



Descripción Detallada de los Servicios de la Capa de Negocio

Servicio de Gestión Dietética:

- **Funcionalidades:**
    - **Creación de planes alimenticios:** Permite a los profesionales de la salud definir planes personalizados para cada paciente, incluyendo:
        - Tipos de alimentos permitidos y prohibidos.
        - Porciones recomendadas por comida y por día.
        - Frecuencia de comidas.
        - Objetivos nutricionales específicos (pérdida de peso, ganancia de peso, mantenimiento).
    - **Actualización de planes alimenticios:** Permite modificar los planes existentes en función de la evolución del paciente o de nuevas recomendaciones médicas.
    - **Visualización de planes alimenticios:** Ofrece una interfaz para que los pacientes puedan consultar su plan de alimentación de manera clara y detallada.
    - **Cálculo de calorías y nutrientes:** Realiza cálculos precisos de las calorías y nutrientes contenidos en cada comida, comparándolos con los objetivos del plan.
    - **Gestión de equivalencias:** Permite establecer equivalencias entre diferentes alimentos para facilitar el registro de la ingesta.
- **Datos:**
    - Información detallada sobre los alimentos (calorías, macronutrientes, micronutrientes).
    - Planes alimenticios personalizados para cada paciente.

Servicio de Estadísticas:

- **Funcionalidades:**
    - **Cálculo de métricas:** Calcula métricas clave como:
        - Calorías consumidas vs. calorías recomendadas.
        - Porcentaje de cumplimiento del plan.
        - Tendencia de peso.
        - Análisis de nutrientes (proteínas, carbohidratos, grasas).
    - **Generación de reportes:** Genera reportes personalizados para pacientes y profesionales, incluyendo gráficos y tablas que visualicen los datos de manera clara.
    - **Análisis de tendencias:** Identifica patrones en el comportamiento alimentario de los pacientes (por ejemplo, días de la semana en los que se cumple menos el plan, alimentos que se consumen con más frecuencia).
- **Datos:**
    - Historial de registro de ingesta de los pacientes (por mes o por día).
    - Configuración de los reportes (período, métricas a incluir).
    - Metas nutricionales de cada paciente.

3. Servicio de Comunicación:

- **Funcionalidades:**
    - **Envío de mensajes:** Permite a los pacientes y profesionales intercambiar mensajes de texto y archivos (por ejemplo, informes médicos).
    - **Notificaciones:** Envía notificaciones push para recordar a los pacientes el registro de su ingesta o para informarles de cambios en su plan.
    - **Historial de conversaciones:** Guarda un historial de todas las conversaciones para facilitar la consulta posterior.
- **Datos:**
    - Información de contacto de los usuarios.
    - Contenido de los mensajes.

4. Servicio de Autenticación:

- **Funcionalidades:**
    - **Registro de usuarios:** Permite a los pacientes y profesionales registrarse en la plataforma.
    - **Gestión de credenciales:** Almacena de forma segura las contraseñas de los usuarios.
    - **Autenticación:** Valida las credenciales de los usuarios al iniciar sesión.
    - **Autorización:** Define los permisos de acceso de cada usuario a las diferentes funcionalidades de la aplicación.
- **Datos:**
    - Información de los usuarios (nombre, correo electrónico, contraseña).
    - Roles y permisos de cada usuario.

Tecnologías Por Implementar

Base:

- **Frontend:** Angular Framework
- **Backend:** Django, Django REST Framework
- **Base de Datos:** Relacional MySQL
- **Servicios en la Nube:** Servidores y Herramientas AWS, Docker Containers.

Conexiones entre Componentes:

- **API RESTful:** Para la comunicación entre el frontend y los microservicios del backend.
- **WebSockets:** Para el servicio de comunicación en tiempo real.
- **Autenticación JWT:** Cada solicitud se valida mediante un token de autenticación.

Consideraciones de Seguridad

Autenticación y Autorización

- **Token-based authentication:** Utiliza tokens JWT (JSON Web Tokens) para autenticar a los usuarios. Estos tokens son seguros, fáciles de implementar y permiten una gestión eficiente de sesiones.
- **Roles y permisos:** Define roles para los usuarios (paciente, profesional de la salud, administrador) y asigna permisos específicos a cada rol, limitando el acceso a las funcionalidades y datos según corresponda.

Cifrado

- **Cifrado de datos en reposo:**
    - **Base de datos:** Utiliza el cifrado transparente para proteger los datos almacenados en la base de datos. Configura el cifrado de columnas o tablas sensibles, como contraseñas y datos médicos.
    - **Archivos:** Si almacenas archivos, como imágenes o documentos, encripta los archivos antes de guardarlos en el disco.

Cifrado de datos en tránsito:

- **HTTPS:** Utiliza HTTPS para todas las comunicaciones entre el cliente y el servidor. Esto garantiza que los datos transmitidos estén encriptados y protegidos contra interceptación.
- **TLS/SSL:** Configura correctamente los certificados SSL para asegurar la conexión entre el cliente y el servidor.

Otras Consideraciones de Seguridad

- **Validación de entrada:** Valida y sanitiza todos los datos de entrada para prevenir ataques de inyección (SQL injection, XSS).
- **Protección contra CSRF (Cross-Site Request Forgery):** Implementa tokens CSRF para proteger contra ataques que explotan la confianza del navegador en el sitio web.
- **WAF (Web Application Firewall):** Considera el uso de un WAF para proteger tu aplicación de ataques comunes como DDoS, SQL injection y XSS.
- **Gestión de errores:** Maneja los errores de forma segura, evitando revelar información sensible en los mensajes de error.
- **Auditoria:** Implementa un sistema de auditoría para rastrear las acciones de los usuarios y detectar actividades sospechosas.
- **Escaneo de vulnerabilidades:** Realiza escaneos de vulnerabilidades de forma regular para identificar y corregir cualquier problema de seguridad.

Especificidades de Django y Angular

- **Django:** Utiliza los mecanismos de seguridad integrados de Django, como el middleware de autenticación y autorización, y las herramientas para gestionar usuarios y permisos.
- **Angular:** Implementa la seguridad en el frontend utilizando las mejores prácticas de Angular, como la protección de rutas y la validación de formularios.