

UNIVERSIDAD TECNOLÓGICA DE CANCÚN



Nombres: Briceño Huchim Josue Misael

Materia: ARQUITECTURA DE SOFTWARE

ACCIÓN DE MEJORA

CASO DE ESTUDIO

Caso de estudio: Aquicomio.

Una de las mayores problemáticas de todo viajero, explorador y amante de las aventuras, es la hora de la comida, al no encontrar un restaurante de calidad o a gusto del comensal, esto se deriva de una mala experiencia, dónde muchas veces te llevas un amargo sabor de boca, el turismo es increíble, conocer nuevos paisajes, nuevos lugares, nuevas experiencias, lo cual luego de un largo día o para comenzar un largo día, lo que uno busca es una comida excelente y de calidad, conectar con los amantes de la comida, conectar con eventos culturales de comida, asistir a los mejores restaurantes del área y la mejor calidad de la comida, por lo cual muchas veces al no conocer el área, se deriva de una mala decisión de dónde comer.

Tipo: Tecnología y Alimentación.

Objetivo: Mejorar la experiencia gastronómica mediante el uso de tecnología de los turistas para conectar a los amantes de la comida con restaurantes de calidad.

Aquicomio es una tecnología que busca revolucionar la forma en que las personas descubren, eligen y disfrutan de la comida, la aplicación surgió de la idea de que la tecnología puede ser una herramienta poderosa para conectar a los amantes de la comida con restaurantes excepcionales, proporcionando una experiencia gastronómica única y personalizada, una experiencia inigualable para todo turista que visite una región que guste.

Objetivos:

- Crear una plataforma digital que permita a los usuarios descubrir restaurantes de calidad.
- Utilizar la inteligencia artificial para personalizar las recomendaciones gastronómicas según las preferencias individuales.
- Facilitar la reserva y el pedido en línea para mejorar la conveniencia y la eficiencia de este mismo.
- Fomentar la colaboración con restaurantes para ampliar la oferta de opciones gastronómicas buscando la mejor opción para el comensal.

Requerimientos funcionales:

Exploración de opciones de restaurantes:

- Los usuarios deben poder buscar restaurantes por ubicación, tipo de cocina, calificación y disponibilidad.

Reservas de restaurantes:

- Funcionalidad que permite a los usuarios realizar reservas en restaurantes, indicando el número de personas y preferencias alimenticias si es necesario al igual que alergias.

Seguimiento de rutas gastronómicas:

- Capacidad para crear y seguir rutas gastronómicas recomendadas, que incluyan varios restaurantes con temáticas específicas.

Calificaciones y reseñas:

- Los usuarios pueden calificar y dejar reseñas sobre los restaurantes visitados, contribuyendo a la comunidad de Aquicomio para la expansión de este mismo.

Notificaciones personalizadas:

- Sistema de notificaciones que alerta a los usuarios sobre ofertas especiales, nuevos restaurantes en la zona y eventos gastronómicos, así como de ofertas y promociones.

Requerimientos No Funcionales:

Usabilidad:

- La interfaz debe ser intuitiva, fácil de navegar y accesible para todo tipo de usuarios.

Rendimiento:

- La aplicación debe tener tiempos de carga rápidos, especialmente al buscar restaurantes y rutas gastronómicas para la comodidad del usuario.

Seguridad:

- Garantizar la seguridad de la información del usuario, como datos de pago y preferencias alimenticias, para evitar cualquier fuga de información.

Escalabilidad:

- La aplicación debe ser escalable para manejar un crecimiento significativo de usuarios y datos que esta aplicación genera por la expansión de su conocimiento y popularidad.

Disponibilidad:

- La plataforma debe estar disponible las 24 horas, los 7 días de la semana, para usuarios en diferentes zonas horarias dependiendo del alcance del proyecto.

Funcionalidades claves del sistema:

Perfil de usuario:

- Los usuarios pueden crear perfiles personalizados con sus preferencias gastronómicas y registrar su historial de restaurantes visitados, así como de sus posibles alergias.

Integración de redes sociales:

- Permitir a los usuarios compartir sus experiencias gastronómicas en redes sociales directamente desde la aplicación, para una mayor publicidad y crecimiento posible del proyecto.

Recomendaciones:

- Opción que sugiere restaurantes y rutas gastronómicas en función de las preferencias de los usuarios

Eventos gastronómicos:

- Funcionalidad para listar y mostrar eventos gastronómicos, como festivales de comida o degustaciones, con detalles y opciones de compra de boletos, así como la ubicación y los lugares donde se realizará.

Funcionales:

Exploracion de restaurantes:

Funcionalidad: Búsqueda de restaurantes por ubicación, tipo de cocina, calificación y disponibilidad.

Criterios de calidad: Respuesta de búsqueda en segundos, presentación clara de información sobre cada restaurante.

Reservas:

Funcionalidad: Sistema de reservas que permita a los usuarios seleccionar fecha, hora, número de personas y especificar preferencias alimenticias y alergias.

Criterios de calidad: Proceso de reserva en tres pasos o menos, confirmación instantánea y notificación al usuario y al restaurante.

Rutas gastronómicas:

Funcionalidad: Creación y seguimiento de rutas gastronómicas recomendadas con varios restaurantes.

Criterios de calidad: Posibilidad de personalizar rutas, indicaciones claras en el mapa y la capacidad de compartir rutas con otros usuarios

Calificaciones y reseñas:

Funcionalidad: Posibilidad de calificar y dejar reseñas sobre restaurantes visitados.

Criterios de calidad: Sistema de calificación, reseñas visibles para otros usuarios y restricciones para evitar contenido inapropiado.

Notificaciones personalizadas:

Funcionalidad: Sistema de notificaciones que alerta a los usuarios sobre ofertas, nuevos restaurantes y eventos gastronómicos.

Criterios de Calidad: Notificaciones personalizadas basadas en las preferencias y el historial del usuario, con opciones de gestión.

No funcionales:

Usabilidad: Criterios de Calidad: Evaluación positiva en pruebas de usabilidad, interfaz intuitiva y accesible para usuarios de diferentes edades y habilidades.

Rendimiento:

Criterios de Calidad: Tiempos de carga inferiores a 3 segundos, capacidad para manejar simultáneamente al menos 10,000 usuarios activos.

Seguridad:

Criterios de Calidad: Protección de datos personales y de pago, implementación de cifrado SSL, y cumplimiento de regulaciones de privacidad.

Escalabilidad:

Criterios de Calidad: La aplicación debe ser capaz de manejar un crecimiento del 50% en usuarios y datos sin degradación significativa del rendimiento.

Disponibilidad:

Criterios de Calidad: Disponibilidad del servicio del 99.9%, con mantenimientos programados comunicados con anticipación.

Funcionalidades clave:

Perfil de usuario:

Funcionalidad: Crear un usuario a la personalización del usuario deseado

Criterios de Calidad: Interfaz de perfil fácil de usar, capacidad para editar preferencias y revisar historial de restaurantes visitados.

Integración de Redes Sociales:

Funcionalidad: Compartir experiencias a través de este mismo en redes sociales para la mayor expansión del conocimiento de la aplicación.

Criterios de Calidad: Conexión fácil con cuentas de redes sociales, opción de compartir automáticamente al agregar una reseña o reservar un restaurante.

Recomendaciones:

Funcionalidad: Sistemas de recomendaciones donde los usuarios pueden puntuar al restaurante y así realizar un método de posicionamientos para verse en primera página.

Criterios de Calidad: Recomendaciones precisas basadas en el historial y preferencias del usuario, con opciones para personalizar la configuración de recomendaciones.

Eventos gastronómicos:

Funcionalidad: Apartado donde se enseñarán los eventos cercanos y lejanos de la aplicación y restaurantes para conocimiento de los usuarios.

Criterios de Calidad: Lista clara de eventos gastronómicos, opción de compra de boletos directamente desde la aplicación, y notificaciones sobre eventos cercanos.

Arquitecturas a utilizar:

- Micro servicios:

Justificación: Este estilo de arquitectura es perfecto para proyectos que requieren escalabilidad, flexibilidad y mantenimiento independiente de los componentes, en el contexto de Aquicomio, cada funcionalidad clave, como la gestión de usuarios, la exploración de restaurantes y la generación de rutas gastronómicas, puede ponerse como un microservicio independiente, esto facilita la escalabilidad, el mantenimiento y la actualización de cada microservicio se pueden realizar de forma independiente, sin afectar otras partes del sistema por lo cual es perfecto para este proyecto, llevando a cabo la implementación de micro-servicios.

- N Capas:

Justificación: La arquitectura de N Capas divide la aplicación en capas distintas, como presentación, lógica de negocio y acceso a datos, este enfoque es el indicado para proyectos que buscan una separación de responsabilidades y un diseño en módulos.

En el caso de Aquicomio, la capa de presentación manejaría la interfaz de usuario, la capa de lógica de negocio maneja las reglas del negocio, y la capa de acceso a datos manejaría la de la manipulación de la base de datos y la gestión de la información, esto proporciona una estructura organizada que facilita el mantenimiento, la escalabilidad y la adaptación del sistema a medida que se actualiza con nuevas funcionalidades y requisitos.

Estas arquitecturas contienen 3 características por las cuales fueron seleccionados:

MICROSERVICIOS:

Escalabilidad: La arquitectura de micro-servicios permite la escalabilidad de componentes de manera independiente, lo que va de la mano con el requerimiento de escalabilidad del sistema, si hay un aumento en la carga de trabajo, como la gestión de reservas o la exploración de restaurantes, se pueden escalar los microservicios correspondientes sin afectar el sistema en general.

Mantenimiento independiente: Aquicomio es una aplicación compleja con varias funcionalidades, la arquitectura de microservicios facilita el mantenimiento independiente de cada servicio, esto es importante para asegurar que actualizaciones y correcciones se puedan implementar de manera rápida y eficiente sin afectar otras partes de esta misma.

Flexibilidad: Los microservicios permiten una mayor flexibilidad en el desarrollo, cada función principal puede ser un microservicio, el equipo de desarrollo puede trabajar en diferentes aspectos de la aplicación, acelerando el proceso de desarrollo y complementación de los módulos.

N CAPAS:

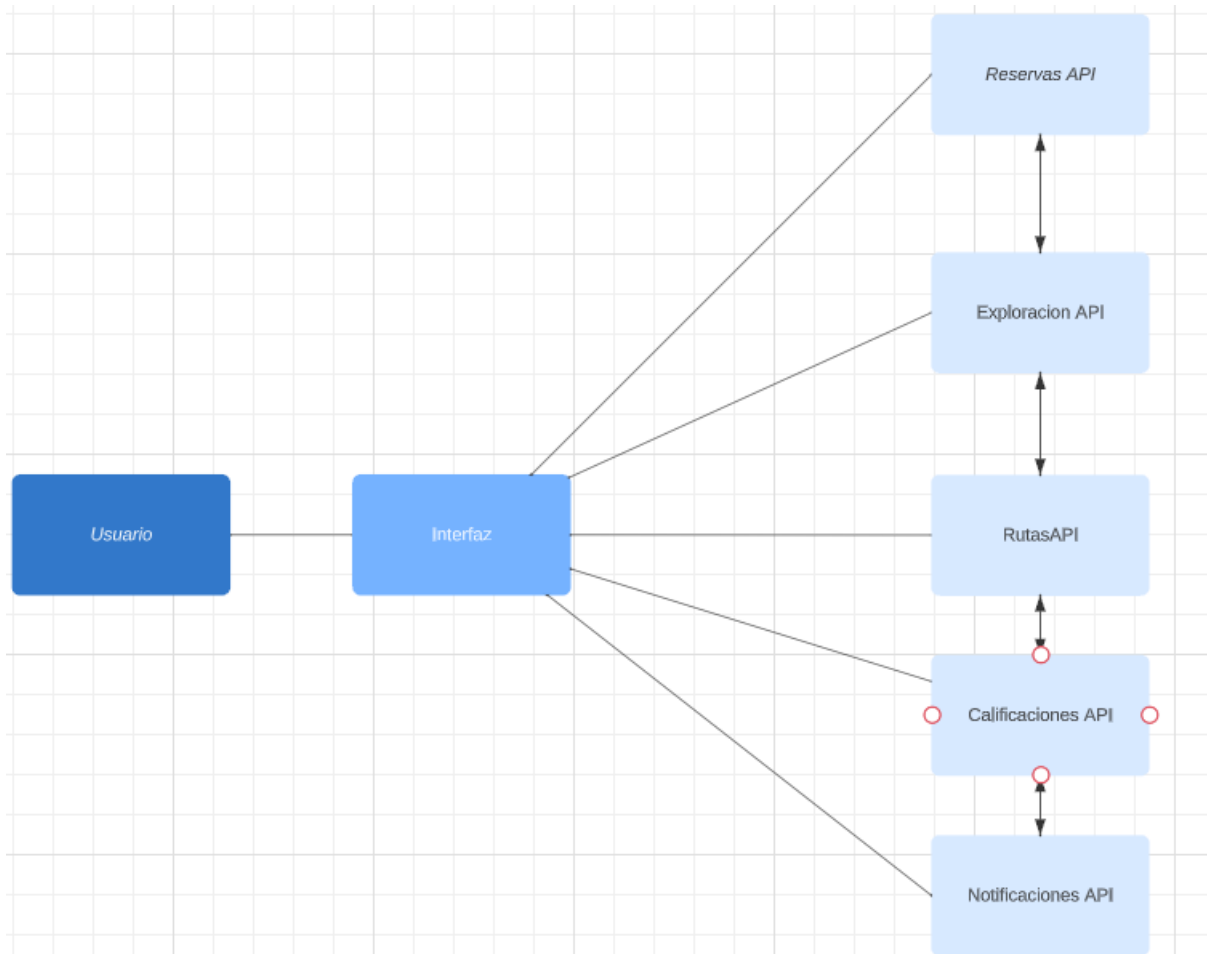
Separación de responsabilidades: La arquitectura de N Capas nos da una clara separación de responsabilidades entre la presentación, la lógica de negocio y el acceso a datos, esto mismo va de la mano con el objetivo de tener un sistema organizado y modular.

Adaptabilidad y mantenimiento: La estructura en capas facilita el mantenimiento y la adaptación de la aplicación a medida que esta se actualiza con nuevos requerimientos, los cambios en la interfaz de usuario, la lógica de negocio o la capa de acceso a datos se pueden realizar de forma independiente, lo que es importante para la evolución continua de la aplicación.

Modularidad: La estructura de N Capas favorece la modularidad, lo que significa que cada capa puede actualizarse de forma independiente, esto facilita la implementación de nuevas funcionalidades e ideas y la mejora de componentes específicos sin afectar otras partes de la aplicación misma.

En conjunto con estas dos arquitecturas, la aplicación puede ser actualizada de manera independiente y sin riesgo de fallas en el sistema a la hora de lanzar actualizaciones y mejoras en este mismo, por lo cual la mejor opción de arquitecturas a utilizar son estas dos para el sistema que se busca implementar.

DIAGRAMAS:



Componentes:

1. Interfaz: La interfaz comunica al usuarios con las diversas APIS a utilizar, dependiendo del servicio que este requiera.
2. Reservas API: Microservicio que gestiona las reservas de los restaurantes dependiendo de la elección del usuarios y sus requerimientos que este proporcione.
3. Exploración API: Microservicio que brinda las opciones y detalles de rutas gastronómicas a elección del usuario y sobre las preferencias de este mismo, de igual manera según las calificaciones de los usuarios
4. Rutas API: Rutas dadas y elegidas por el usuario según sus preferencias, donde enseña cada restaurante y sus opciones de alimentos, así como la distancia y ubicación.
5. Calificaciones API: Microservicio de calificaciones, donde los usuarios pueden puntuar al restaurante y dependiendo de la calificación este se posiciona en un lugar mejor o peor.

6. Notificaciones API: Microservicio de notificaciones personalizadas del usuario, donde se le muestran los detalles de reservas, platillos nuevos y eventos próximos, así mismo como de la popularidad de cada restaurante visitado o reservado.

Conexiones:

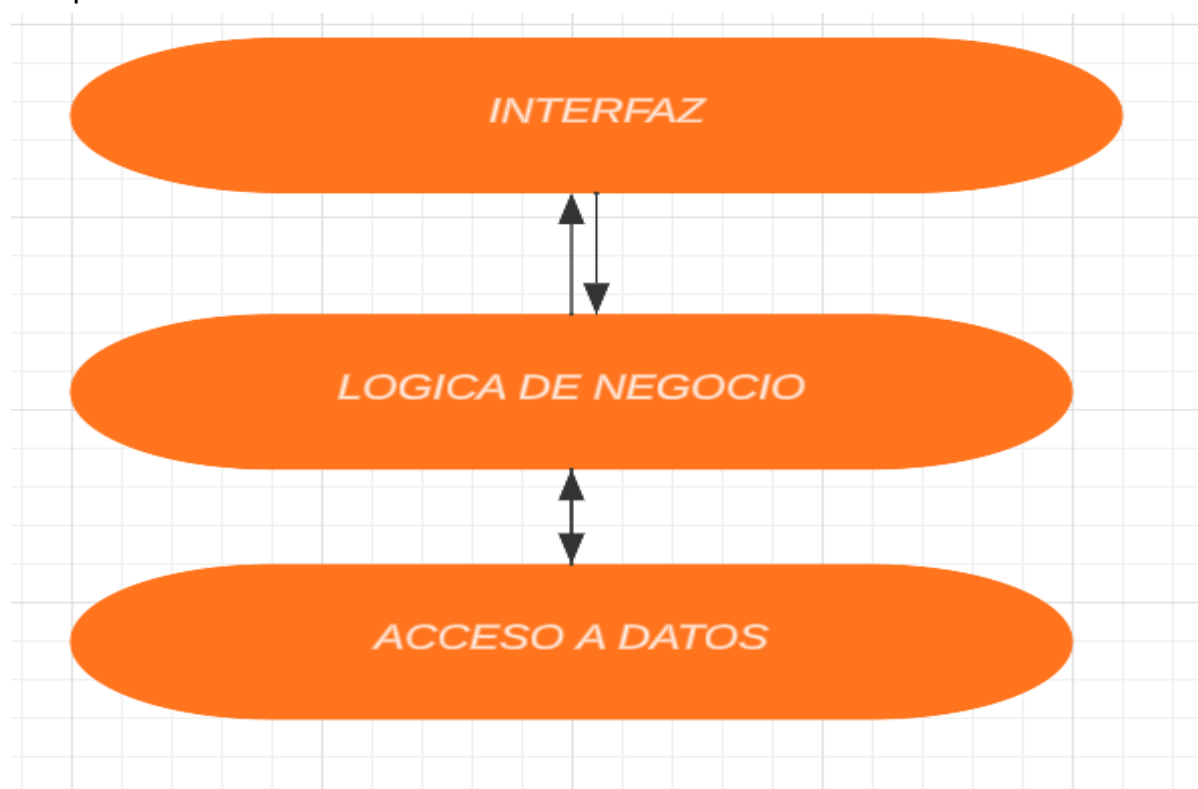
La interfaz se conecta con cada micro servicio el cual le genera al usuario un apartado especial el cual se encarga de brindar la información necesaria o funcionalidad necesaria requerida.

Los microservicios se comunican entre sí, haciendo de esto un programa completo y funcional el cual permite la interacción entre componentes y servicios de este mismo.

Como se aplica:

Cada funcionalidad clave se implementa como un microservicio independiente, permitiendo escalabilidad y mantenimiento independiente.

N capas:



Componentes:

Interfaz: Capa de presentación que interactúa con el usuario.

Lógica de Negocio: Capa que gestiona la lógica de la aplicación, como las reglas de negocio y la gestión de eventos.

Acceso a Datos: Capa encargada de la conexión y manipulación de la base de datos.

Conexiones:

1. La Interfaz se comunica con la Lógica de Negocio para solicitar y presentar información.
2. La Lógica de Negocio se comunica con la Capa de Acceso a Datos para realizar operaciones en la base de datos.

Como se aplica:

La separación de capas permite una clara separación de responsabilidades, facilitando el mantenimiento y la actualización continua de la aplicación, la lógica de negocio y el acceso a datos están claramente definidos, lo que favorece la modularidad y la adaptación del sistema.

Documentación detallada de cada función y requerimiento (MICROSERVICIOS).

1. Interfaz: Permite a los usuarios explorar restaurantes, reservar, calificar y recibir notificaciones personalizadas, cumpliendo con los requisitos clave de la aplicación.
 - a. Función: Interactuar directamente con el usuario, presentar información de manera amigable y enviar solicitudes a los microservicios correspondientes.
 - b. Requerimientos: Permite a los usuarios explorar restaurantes, reservar, calificar y recibir notificaciones personalizadas, cumpliendo con los requisitos clave de la aplicación.
2. Reservas API (Microservicio):
 - a. Función: Gestionar la funcionalidad de reservas en restaurantes, almacenar información sobre reservas y manejar la confirmación de reservas.
 - b. Requerimientos: Satisface la funcionalidad clave de reservas en restaurantes, asegurando confirmaciones rápidas y eficientes.
3. Exploración API (Microservicio):
 - a. Función: Facilitar la búsqueda y exploración de restaurantes, proporcionando información detallada y actualizada.

- b. Requerimientos: Permite a los usuarios explorar restaurantes según ubicación, tipo de cocina y calificaciones, cumpliendo con el requisito de exploración.
- 4. Rutas API (Microservicio):
 - a. Función: Gestionar la creación, edición y seguimiento de rutas gastronómicas recomendadas.
 - b. Requerimientos: Permite a los usuarios seguir rutas gastronómicas personalizadas, cumpliendo con el requisito de rutas gastronómicas.
- 5. Calificaciones API (Microservicio):
 - a. Función: Manejar la funcionalidad de calificaciones y reseñas de restaurantes, almacenando y mostrando información relevante.
 - b. Requerimientos: Permite a los usuarios calificar y dejar reseñas, cumpliendo con el requisito de calificaciones y reseñas.
- 6. Notificaciones API (Microservicio):
 - a. Función: Enviar notificaciones personalizadas a los usuarios sobre ofertas, nuevos restaurantes y eventos gastronómicos.
 - b. Requerimientos: Garantiza que los usuarios estén informados sobre eventos y ofertas relevantes, cumpliendo con el requisito de notificaciones personalizadas.

Documentación detallada de cada función y requerimiento (NCapas).

- 1. Interfaz:
 - Función: Presentar la interfaz de usuario y manejar la interacción con el usuario.
 - Requerimiento: Facilita una interfaz intuitiva y accesible para usuarios, cumpliendo con el requisito de usabilidad
- 2. Lógica de Negocio:
 - Función: Gestionar la lógica de la aplicación, como las reglas de negocio, la generación de rutas gastronómicas y la gestión de eventos.
 - Requerimiento: Permite la adaptación y evolución continua del sistema, cumpliendo con los requisitos de adaptabilidad y mantenimiento.

3. Acceso a Datos:

- Función: Manejar la conexión y manipulación de la base de datos, almacenando y recuperando información según sea necesario.
- Requerimientos: Garantiza la seguridad de la información del usuario y permite un acceso eficiente a los datos, cumpliendo con los requisitos de seguridad y rendimiento.