**Análisis Final del sistema**

Se utilizó ChatGPT como herramienta para buscar patrones y tácticas recomendadas en arquitecturas de microservicios, También ayudo a definir requerimientos funcionales y atributos de calidad. Esto permitió identificar soluciones probadas y adaptar las mejores prácticas al diseño de los servicios, asegurando que cumplieran con las necesidades del sistema

Link del formato wiki en GitHub: <https://github.com/CowsmonDev/Software-System-Design-TPE/tree/TP-ADD/TP-ADD>

**ADD01: Brownfield y Migración a Microservicios**

**Template Correspondiente:** Arquitectura de referencia para la migración del monolito

Se planteo hacer una migración tipo Brownfield, manteniendo la página en funcionamiento a medida que se hace una migración progresiva de los servicios del monolito a microservicios.

1. **Decisiones Tecnológicas:**
   * + Uso de Spring Boot para microservicios
     + Docker para contenedorización
     + Kong API Gateway para la gestión de comunicaciones

Estas decisiones nos ayudaran a que el sistema continue activo durante la migración, poder comunicarse con el monolito y los microservicios que se vallan creando, Además, los microservicios podrán funcionar en conjunto con el monolito durante la migración.

1. **Atributos de Calidad:**
   * El ADD01 define atributos que son respaldados por las decisiones del template:
     + Compatibilidad: Soportada por Spring Boot y Kong API Gateway
     + Escalabilidad: Facilitada por Docker y la arquitectura de microservicios
     + Disponibilidad: Garantizada por la infraestructura de microservicios, permitiendo que el sistema siga funcionando si se cae un servicio.

**ADD02: Microservicios de Clientes y Pagos**

**Template Correspondiente:** Decisiones de arquitectura de Clientes y Pagos

Estos microservicios eran un requerimiento critico del sistema, El microservicio Clientes Gestiona y guarda los datos de los clientes y el microservicio de Pagos se encarga de gestionar los pagos de los pedidos a través de mercado Libre/ Mercado Pago.

1. **Soluciones Arquitectónicas:**
   * Para Clientes:
     + ADD02 requiere gestión segura de datos personales
     + El template implementa Event Sourcing y Validación de Datos
   * Para Pagos:
     + ADD02 especifica integración con MercadoLibre
     + El template propone Adapter Pattern para esta integración
     + ADD02 se necesitaba gestionar el estado del pago
     + En el Template se propone Event Sourcing utilizando eventos internos para cambiar el estado del pago
2. **Atributos de Calidad:**
   * Las decisiones del template apoyan directamente los atributos especificados en ADD02:
     + Seguridad: Mediante validación y filtrado
     + Disponibilidad: A través de Event Sourcing
     + Escalabilidad: Soportada por la arquitectura elegida (Microservicios), Mejorando el servicio que de problemas

**ADD03: Microservicios de Pedidos y de Reparto y rutas**

**Template Correspondiente:** Implementación de Microservicios para Pedidos y Reparto de con Rutas

El sistema requiere dos microservicios principales: **Pedidos y Repartos con Rutas**, que deben integrarse y garantizar la continuidad del servicio en tiempo real. El objetivo es optimizar el manejo de pedidos y logística de reparto, implementando tecnológicas que permitan escalabilidad, disponibilidad y rendimiento.

1. **Decisiones Arquitectónicas:**
   * Pedidos:
     + ADD03 requiere procesamiento en tres fases
     + El template implementa pipeline asíncrono con tres fases
   * Reparto y Rutas:
     + ADD03 especifica un algoritmo para generar las rutas de viajes
     + El template propone algoritmo dinámico por demanda que genere las rutas
2. **Consecuencias:**
   * Las decisiones del template abordan los atributos de calidad especificados en ADD03:
     + Disponibilidad mediante procesamiento asíncrono
     + Rendimiento a través de optimización dinámica
     + Escalabilidad con arquitectura desacoplada

**ADD04: Microservicios de Estadísticas e Incidencias**

**Template Correspondiente:** Implementación de microservicios para Estadísticas e incidencias

Se están implementando dos microservicios especializados en manejo de datos para devolver respuestas como el estado de los pedidos o la situación en tiempo real de los camiones, y en el caso de las incidencias pueden notificar, por ejemplo, un camión averiado.

1. **Requisitos Funcionales:**
   * Estadísticas e incidecias:
     + ADD04 requiere generación de informes y análisis
     + El template implementa Aggregation Layer
     + ADD04 especifica gestión en tiempo real y escalabilidad
     + El template propone Database per Service
2. **Atributos de Calidad:**
   * Las decisiones del template soportan los atributos especificados en ADD04:
     + Disponibilidad: Mediante bases de datos independientes
     + Exactitud: A través de la capa de agregación
     + Rendimiento: Optimización de consultas específicas