CASO DE USO

Integración de Datos Multi-fuente para Cadena Hotelera

Alonso Gómez

Andrews Dos Ramos

Lucian Ciusa

Mario García

Sergio Jiménez

1. Análisis de Requerimientos

1.1. Resumen ejecutivo

Hospitality Excellence Group necesita centralizar datos de 200 hoteles (40 países) que usan sistemas heterogéneos —PMS variados, plataformas de reservas, reseñas, ERPs, sensores IoT y archivos locales— para proporcionar análisis corporativos y reportes ejecutivos diarios. El reto principal es integrar >50 fuentes con ventanas nocturnas limitadas, orquestar >100 pipelines, manejar dependencias complejas y garantizar calidad, disponibilidad y escalabilidad.

1.2. Requerimientos funcionales clave

- **Conectar y extraer datos desde >50 fuentes**: Azure SQL, PostgreSQL, MySQL, Oracle, APIs REST, SFTP, CSV/Excel, servicios SaaS (Booking, Expedia, TripAdvisor, etc.).
- **Soportar modos de extracción**: exportaciones nocturnas, APIs en tiempo real y cargas de archivos manuales.
- **Transformar y limpiar datos**: normalización de esquemas, deduplicación, enriquecimiento (*lookups*), y homologación de formatos.
- **Cargar a un *data warehouse* central** (Azure Synapse) con particionado y manejo de cargas incrementales y *full refresh*.
- **Orquestación de >100 pipelines** con dependencias, reintentos, *backfills* y ejecución condicional.
- **Monitoreo y alertas automáticas** (correo, Slack, Teams) en fallos críticos y métricas de SLA.
- **Soporte para transformaciones complejas** (agregaciones, *joins* entre fuentes, correlaciones por cliente/reserva).
- **Exponer datos a herramientas de BI** (Power BI/Tableau) y APIs internas para consumos ad-hoc y reportes ejecutivos.
- **Gestión de parámetros dinámicos** (fechas, hotel, entorno) y control de versiones de pipelines/transformaciones.

1.3. Requerimientos no funcionales

- **Performance**: ETL/ELT diarios deben completarse dentro de la ventana operativa (4 horas) para la mayoría de las cargas críticas.
- **Escalabilidad**: Capacidad de escalar horizontalmente ingestión y el procesamiento para soportar crecimiento (más hoteles, sensores).
- **Disponibilidad**: 99.9% para la capa de orquestación y monitorización; 99.95% para procesos críticos durante ventana.
- **Consistencia de Datos**: Garantizar idempotencia, exactitud y consistencia eventual entre sistemas fuente y destino.
- **Seguridad y Cumplimiento**: Cifrado en tránsito y reposo, control de acceso (RBAC), registro de auditoría, cumplimiento GDPR/PCI según aplique.
- **Observabilidad**: Trazabilidad completa de linaje de datos, logs estructurados, métricas y dashboards de salud.
- **Mantenibilidad**: Pipelines parametrizables, módulos reutilizables, pruebas automatizadas y CI/CD para despliegues.

1.4. Restricciones y consideraciones

- **Ventana de procesamiento nocturno limitada** (4 horas) para muchas fuentes; requiere optimización y priorización.
- **Fuentes *legacy*** con exportaciones **solo nocturnas o archivos manuales** (CSV/Excel) y **limitada conectividad**.
- **Variabilidad en calidad de datos entre hoteles**: valores nulos, formatos de fecha distintos, duplicados.
- **Cumplimiento regulatorio y restricciones de residencia de datos** (*data residency*) en ciertos países.
- **Costes asociados a transferencia e ingestión masiva** hacia Azure (*ingress*/*egress*) y **almacenamiento** en Synapse.
- **Dependencia de APIs externas** (Booking, Expedia) con límites de *rate-limit* y SLAs fuera de nuestro control.
- **Necesidad de garantizar re-ejecución eficiente** (*retries*, idempotencia) y capacidades de ***backfill***.

2. Arquitectura Propuesta

2.1. Diagrama de Arquitectura

2.1.1. Componentes Principales y Flujo de Datos

El diagrama incluirá:

- Fuentes de Datos:
  - Bases de datos locales en cada hotel (Oracle, PostgreSQL, MySQL, Azure SQL)
  - Sistemas PMS variados (Oracle Hospitality, Protel, sistemas propietarios)
  - Plataformas externas (Booking.com, Expedia, Airbnb)
  - Plataformas de opinión y satisfacción (TripAdvisor, Google Reviews, encuestas)
  - Redes sociales (menciones de marca)
  - Sistemas ERP (SAP, Oracle Financials, sistemas locales)
  - Sensores IoT (termostatos, cerraduras, consumo energético)
  - Archivos CSV/Excel
- Servicios de Azure utilizados y otras herramientas sugeridas:
  - Azure Data Factory (ADF): Orquestación y pipelines ETL/ELT
  - Azure Data Lake Storage Gen2: Almacenamiento de datos raw y transformados
  - Azure Synapse Analytics: *Data warehouse* para almacenamiento centralizado y análisis
  - Azure Functions: Para procesamiento de transformaciones personalizadas o *trigger* de eventos
  - Azure Logic Apps / Power Automate: Para integraciones con APIs REST externas y flujos automáticos
  - Azure Monitor + Log Analytics: Para monitoreo y alertas
  - Azure Key Vault: Gestión segura de credenciales y secretos
  - Azure SQL Database: Para *staging* o bases intermedias en caso necesario

- Flujo general de datos:
    - Ingesta: Datos se extraen desde las fuentes, mediante conectores de ADF (bases, APIs, SFTP, archivos).
    - Almacenamiento Raw: Datos ingieren en Azure Data Lake Storage Gen2 en formato raw (sin procesar).
    - Procesamiento/Transformación: ADF ejecuta pipelines para limpieza, homologación, agregaciones, y *joins*; puede usar Azure Functions para lógica específica.
    - Carga: Datos transformados se cargan a Azure Synapse Analytics para análisis y generación de reportes.
    - Consumo: Herramientas de BI (Power BI), reportes ejecutivos, analítica avanzada acceden a Synapse.
    - Monitoreo y Orquestación: ADF orquesta pipelines, Azure Monitor supervisa procesos y activa alertas.

2.1.2. Capas de la Arquitectura

Capa	Descripción
Ingesta	Extracción de datos desde bases, APIs, archivos y sensores usando Azure Data Factory y Logic Apps
Almacenamiento	Azure Data Lake Storage Gen2 como repositorio central para datos raw y transformados
Procesamiento	Transformación y limpieza con Azure Data Factory + Azure Functions para lógica compleja
Almacenamiento final	Azure Synapse Analytics para consolidación y análisis empresarial
Consumo	Herramientas BI (Power BI), dashboards ejecutivos, análisis avanzado
Monitoreo y Seguridad	Azure Monitor, Log Analytics, Azure Key Vault para gestión de credenciales y seguridad

2.2. Descripción de Componentes

2.2.1. Azure Data Factory (ADF)

- **Rol en la solución**: Orquestación de pipelines ETL/ELT que extraen datos de múltiples fuentes, los transforman y cargan en el *data warehouse*.
- **Por qué se eligió**:
  - Integración nativa con múltiples orígenes y destinos (Azure SQL, Synapse, APIs, SFTP, archivos).
  - Soporta parametrización dinámica (fechas, configuraciones por hotel).
  - Capacidad para ejecutar pipelines en paralelo o en secuencia, con dependencias complejas.
  - Manejo integrado de monitoreo y alertas.
  - Fácil escalabilidad y mantenimiento.
- **Alternativas y razones para no elegir**:
  - Apache Airflow: Muy flexible, pero requiere infraestructura propia y mayor gestión operativa.
  - SSIS: Menos escalable y no nativo para la nube Azure.
  - Databricks Jobs: Mejor para procesamiento intensivo en Spark, pero no óptimo para orquestación multi-fuente heterogénea.
- **Configuración relevante**:
  - Pipelines configurados con *triggers* horarios (ventanas nocturnas de 4h).
  - Parámetros dinámicos para configuración por hotel y fechas.
  - Uso de *linked services* para conexión segura a fuentes.

2.2.2. Azure Data Lake Storage Gen2

- Rol en la solución: Almacena datos en estado raw (sin transformar) y también datos procesados intermedios para auditoría y recuperación.
- Por qué se eligió:
    - Compatible con formatos optimizados (Parquet, CSV)
    - Alta escalabilidad y costo eficiente para grandes volúmenes
    - Integración nativa con ADF y Synapse
    - Seguridad avanzada con Azure AD y control granular de acceso
- Alternativas y razones para no elegir:
    - Blob Storage clásico: Menos orientado a análisis y gestión de archivos grandes
    - Bases de datos SQL: Costosas para grandes volúmenes y no óptimas para datos sin estructura
- Configuración relevante:
    - Estructura de carpetas organizada por fuente, fecha, hotel
    - Control de acceso basado en roles
    - Versionado y retención para trazabilidad

2.2.3. Azure Synapse Analytics

- **Rol en la solución**: *Data warehouse* corporativo para almacenamiento centralizado y consultas analíticas, base para reportes ejecutivos.
- **Por qué se eligió**:
  - Escalabilidad y rendimiento para grandes volúmenes de datos.
  - Integración nativa con ADF, Data Lake y Power BI.
  - Soporta SQL Server, Spark *pools*, y *pipelines* propios.
  - Opciones de almacenamiento separado y cómputo dedicado para optimización.
- **Alternativas y razones para no elegir**:
  - Azure SQL Database: No está diseñado para *data warehouse* de alta escala.
  - Databricks: Orientado a análisis avanzado, pero no reemplaza un DW corporativo tradicional.
  - Redshift o BigQuery: No nativos en Azure, aumento de complejidad operativa.
- **Configuración relevante**:
  - Particionado de tablas por hotel y fecha para acelerar consultas.
  - Pools de cómputo escalables configurados para cargas nocturnas.
  - Seguridad y auditoría habilitada.

2.2.4. Azure Functions

- **Rol en la solución**: Implementar transformaciones personalizadas o tareas específicas que no se pueden realizar fácilmente en ADF (p. ej. llamadas a APIs externas, procesamiento de JSON complejo).
- **Por qué se eligió**:
  - *Serverless*, escalable y flexible.
  - Fácil integración con ADF mediante *triggers* y actividades personalizadas.
  - Reduce carga en pipelines principales.
- **Alternativas y razones para no elegir**:
  - Logic Apps: Mejor para *workflows* simples, no para lógica pesada o cálculos.
  - Azure Batch: Más orientado a procesos *batch* grandes y programados, no eventos inmediatos.
- **Configuración relevante**:
  - *Timeouts* configurados según ventana de procesamiento.
  - Monitoreo mediante Application Insights.

2.2.5. Azure Logic Apps

- **Rol en la solución**: Automatización de integración con APIs REST externas (Booking, Expedia, plataformas sociales) y flujos de trabajo simples.
- **Por qué se eligió**:
  - Diseñado para integración con sistemas SaaS y APIs sin desarrollo complejo.
  - Bajo mantenimiento y fácil de modificar.
  - Integración con ADF y otros servicios Azure.
- **Alternativas y razones para no elegir**:
  - Funciones Azure: Más flexibles, pero requieren programación.
  - Herramientas de terceros: Aumentan la complejidad y costos.
- **Configuración relevante**:
  - Flujos con control de errores y reintentos.
  - Conexiones autenticadas mediante Key Vault.

2.2.6. Azure Monitor y Log Analytics

- **Rol en la solución**: Supervisión integral, alertas y diagnóstico de pipelines, funciones y servicios Azure.
- **Por qué se eligió**:
  - Integración nativa con servicios Azure.
  - Visualización avanzada y configuración de alertas personalizadas.
  - Centralización de logs y métricas para análisis.
- **Alternativas y razones para no elegir**:
  - Soluciones *on-premises*: Mayor complejidad e integración limitada.
  - Herramientas externas (Datadog, Splunk): Costos adicionales y menor integración directa.
- **Configuración relevante**:
  - Dashboards personalizados para monitoreo en tiempo real.
  - Alertas configuradas para fallas y retrasos en pipelines.

2.2.7. Azure Key Vault

- **Rol en la solución**: Gestión segura de credenciales, claves API, contraseñas y secretos.
- **Por qué se eligió**:
  - Seguridad avanzada y cumplimiento normativo.
  - Integración con ADF, Functions, Logic Apps para autenticación segura.
  - Control de acceso granular y auditoría.
- **Alternativas y razones para no elegir**:
  - Almacenamiento en texto plano o configuración manual: Riesgos de seguridad.
  - Herramientas externas: Menos integración y complejidad añadida.
- **Configuración relevante**:
  - Rotación periódica de secretos.
  - Políticas de acceso estrictas.

Esta arquitectura aprovecha al máximo los servicios nativos de Azure para gestionar de manera eficiente la integración de datos multi-fuente, garantizando escalabilidad, seguridad, mantenimiento sencillo y cumplimiento con ventanas de procesamiento estrictas. La combinación de ADF para orquestación, Data Lake para almacenamiento raw, Synapse para análisis y servicios *serverless* para lógica específica, permite una solución robusta y flexible.

3. Patrones de Arquitectura

3.1. Patrón de ingesta de datos

Híbrido (*Batch* + *Streaming*):

- ***Batch***: para fuentes *legacy*, exportaciones nocturnas, archivos CSV/Excel y sincronizaciones programadas. Permite procesamiento por lotes optimizado dentro de la ventana de 4 horas.
- ***Streaming* (*micro-batch*/*near-real-time*):** para reservas vía APIs, sensores IoT y redes sociales donde la baja latencia aporta valor. Se emplean colas/*streaming* (Event Hub / Kafka) para absorber picos y desacoplar productores y consumidores.

3.2. Patrón de procesamiento

ELT (preferible) **con enfoque modular** — combinado con patrones Lambda/Kappa según el caso:

- **ELT principal**: extraer crudo a un Data Lake (*raw zone*), realizar transformaciones en Synapse o Spark (*compute*) y cargar modelos curados en Synapse (*warehouse*). Reduce movimiento de datos y aprovecha la capacidad de procesamiento distribuido.
- **Lambda (cuando se requiere)**: combinar *batch* (*historical*/*complete reprocessing*) y *streaming* (capa de velocidad) para casos que requieren bajas latencias
- **Kappa (simplificado)**: usar *streaming* como fuente única para *pipelines* que puedan ser procesados en modo *streaming* continuo (por ejemplo, telemetría IoT y menciones sociales).

3.3. Patrón de almacenamiento

***Lakehouse* híbrido (*Data Lake* + *Data Warehouse*)**:

• ***Raw zone* en Azure Data Lake Storage Gen2** (*parquet*/*Delta*) para datos sin transformar y auditoría.

• ***Staging / Cleansed zone***: datos transformados, particionados y versionados (Delta Lake) para permitir *time travel* y ACID en transformaciones.

• ***Serving / Curated zone***: tablas y vistas optimizadas en Azure Synapse Analytics para consumo por BI y cargas analíticas.

• **Archivos históricos y *backups* fríos** en almacenamiento coste-eficiente (*cool*/*archive tiers*).

3.4. Patrón de consumo

Multicanal:

- **BI (Power BI / Tableau)**: Conexiones directas/semánticas a las capas *curated*/*serving* en Synapse, con modelos tabulares y agregados para reportes ejecutivos.
- **ML**: *Datasets* preparados en la zona *curated* o en un *feature store* para modelos de predicción (*churn*, demanda, precios dinámicos).
- **APIs internas**: Exponer *endpoints* para consultas ad-hoc o integraciones con sistemas operacionales (p. ej. *dashboards* operativos).
- **Self-service**: *Data products* y catálogos con documentación y *data lineage* para facilitar consumo por equipos de negocio.

3.5. Decisiones arquitectónicas clave

- Adoptar **Delta Lake en ADLS Gen2** para versionado y cargas incrementales eficientes.
- Usar **Azure Data Factory** / **Synapse Pipelines** o un orquestador como **Apache Airflow** / **Azure Data Factory** para **orquestación y dependencias complejas**.
- Reservar **Event Hubs** o **Kafka** para **ingestión de *streaming*** (IoT, social, APIs en tiempo real).
- Implementar **observabilidad** con **Azure Monitor**, **Log Analytics**, y **soluciones de *metadata*/*lineage*** (Microsoft Purview o similar).
- Configurar **CI/CD** para **pipelines y notebooks** (GitOps) y **políticas de acceso** (RBAC) y **secreto** (KeyVault).

3.6. Recomendaciones operativas

- **Priorizar *pipelines* críticos** dentro de la ventana de 4 horas y ejecutar el resto en micro-ventanas o en modo *streaming*.
- **Particionar y compactar archivos *Parquet*/*Delta*** para optimizar lecturas y reducir tiempo de procesamiento.
- **Implementar pruebas automáticas** (*unit*/*integration*) para transformaciones y alertas con *runbooks* para recuperación.
- **Definir SLAs por *pipeline* y reportar métricas** de éxito/fallo y duración por ejecución.
- **Plan de *backfill* y retención de datos** con políticas claras y control de costes.

4. Flujo de Datos de Extremo a Extremo

4.1. Origen: ¿De dónde vienen los datos?

Los datos provienen de **múltiples fuentes**, incluyendo:

- **Sistemas PMS** (Sist. Gest. Propiedades) **locales** (Oracle Hospitality, Protel).
- **Plataformas de reservas online**: Booking.com, Expedia, Airbnb.
- **Plataformas de satisfacción del cliente**: TripAdvisor, Google Reviews, encuestas internas.
- **Redes sociales**: Twitter, Instagram, Facebook (menciones de marca).
- **Sistemas financieros / ERP**: SAP, Oracle Financials, soluciones locales.
- **IoT en hoteles inteligentes**: termostatos, cerraduras electrónicas, sensores de consumo energético.

4.2. Ingesta: ¿Cómo se capturan los datos?

Tecnologías y métodos de ingesta empleados:

- **Conectores nativos de ETL/ELT** para bases de datos (Azure SQL, PostgreSQL, MySQL, Oracle).
- **APIs REST** para plataformas de reservas, opiniones y redes sociales.
- **Web scraping controlado** (si no existen APIs disponibles en plataformas de reseñas).
- **Gateways IoT o Azure IoT Hub** para dispositivos inteligentes.
- **Extracción por lotes nocturnos** para sistemas que no permiten conexión en tiempo real.

Herramientas de ingesta:

- **Azure Data Factory** (ADF).
- **Azure Logic Apps** / **Functions** para procesos *event driven*.
- ***Dataflows*** para ingesta semiestructurada o no estructurada.

4.3. Procesamiento: ¿Qué transformaciones se aplican?

Transformaciones clave:

- **Homologación de formatos**: Unificación de estructuras y tipos de datos (fechas, divisas, códigos de país, idiomas, etc.).
- **Normalización**: Convertir estructuras jerárquicas o anidadas a modelos tabulares.
- ***Join/Lookup***: Cruzar información entre fuentes (e.g., *matching* reservas con opiniones post-estadía).
- **Agregaciones**: Por hotel, país, tipo de habitación, canal de venta.
- **Cálculo de métricas**: Ocupación, ingresos promedio, Net Promoter Score, consumo energético por huésped.
- **Carga incremental o *full refresh*** según tipo de fuente y ventana disponible.

Tecnologías de transformación:

- **Azure Data Factory** (*Data Flows*).
- **Azure Synapse Pipelines y Notebooks Spark** para cargas complejas.
- **Azure Functions** para lógica personalizada o procesamiento ligero.

4.4. Almacenamiento: ¿Dónde se guardan los datos?

Destino final:

- **Azure Synapse Analytics** (*Data Warehouse* corporativo).

Almacenamientos intermedios:

- **Azure Data Lake** (*staging* y *raw zones*).
- **Azure Blob Storage** (archivos sin estructurar).
- Azure SQL Database (para *staging* de estructuras relacionales).

Modelo de almacenamiento:

- **Esquema en estrella o snowflake** para BI.
- **Particionado** por fecha, país, hotel.
- **Versionado de datos** para auditoría y trazabilidad.

4.5. Consumo: ¿Cómo se utilizan los datos?

- **Reportes ejecutivos y dashboards** en Power BI integrados con Azure Synapse.
- **Alertas e indicadores de desempeño automatizados** por hotel, región, canal.
- **Modelos predictivos** sobre ocupación, mantenimiento preventivo, y análisis de sentimiento.
- **Exportaciones automáticas** a Excel o PDF para *stakeholders* regionales.
- **Integración con sistemas de decisión corporativa** (CRM, Revenue Management Systems).

5. Justificación de Integración entre Servicios

5.1. ¿Cómo se integran los servicios entre sí?

Los componentes de la arquitectura están orquestados principalmente mediante **Azure Data Factory**, que sirve como motor central de *pipelines*. ADF coordina:

- Extracción desde APIs o bases de datos.
- Transformaciones vía Data Flows o Synapse.
- Ejecución de Azure Functions para lógica personalizada.
- Llamadas a Logic Apps para tareas automatizadas.
- Carga en Azure Synapse.

Además, **Azure Event Grid** puede emplearse para disparar eventos cuando nuevos datos llegan a un *blob storage* (por ejemplo, archivos CSV mensuales).

5.2. ¿Qué protocolos o conectores se usan?

- **ODBC / JDBC**: Conexión a bases de datos (PostgreSQL, MySQL, Oracle, SQL Server).
- **REST / HTTP**: Para APIs de terceros (Booking.com, TripAdvisor, redes sociales).
- **MQTT / AMQP**: Protocolos para ingestión de datos IoT.
- **Blob / Data Lake API**: Para lectura y escritura en almacenamiento intermedio.

5.3. ¿Cuáles son los puntos de integración críticos?

- **Integración con sistemas *legacy* PMS**: Algunos no tienen APIs modernas, por lo que se requieren soluciones personalizadas o extracciones por lotes.
- **Conexión con plataformas de reservas y *reviews***: Alta variabilidad en APIs y límites de tasa.
- **Sincronización entre pipelines dependientes**: Ejecución secuencial y paralela según el tipo de dato.
- **Transformaciones de alto volumen y complejidad**: Deben completarse dentro de una ventana de 4 horas.
- **Carga a Synapse y conexión con Power BI**: Deben garantizar consistencia de datos y disponibilidad diaria.

5.4. ¿Qué consideraciones de seguridad existen?

- **Autenticación segura** mediante OAuth 2.0 para APIs públicas (Booking, Google Reviews).
- **Azure Key Vault** para gestión de secretos, claves de API y credenciales de bases de datos.
- **Cifrado en tránsito y en reposo** (TLS/SSL para transporte, AES-256 para almacenamiento).
- **RBAC (Role-Based Access Control)** en Azure para controlar accesos por rol.
- **Auditoría y monitoreo** de accesos y procesos ETL.
- **Redes privadas y *endpoints*** seguros para conectividad entre servicios internos.

5.5. Resumen Ejecutivo

Hospitality Excellence Group enfrenta una integración de datos compleja debido a la gran cantidad de fuentes heterogéneas, diferentes tecnologías y limitaciones operativas. Una arquitectura de datos moderna basada en Azure permite:

- Capturar datos de más de 50 fuentes usando conectores nativos, APIs y archivos.
- Procesar datos mediante pipelines orquestados con lógica condicional, paralela y secuencial.
- Almacenar de forma centralizada en Azure Synapse con modelos optimizados para BI.
- Proveer datos limpios, actualizados y consistentes a la capa de consumo (Power BI, análisis avanzado).

Todo esto bajo una estrategia de seguridad robusta y con alta capacidad de monitoreo y recuperación ante fallos.

6. Presupuesto Estimado

Servicio	Configuración Típica Asumida	Costo Mensual Estimado (USD)
Azure Synapse Analytics (Data Warehouse)	Pool Dedicado: 500 DWUs (Gen2) activos 8 horas/día (ventana de procesamiento nocturno)	$14,500 - $22,000
Azure Data Factory (ADF)	100 flujos de Control/Datos, Ejecuciones diarias (2,000+ actividades)	$800 - $1,500
Azure Databricks (Transformación ETL Pesada)	Cluster Estándar: 3 máquinas virtuales D8ds v4 (32 vCores) activas 4 horas/día	$3,500 - $5,500
Azure Data Lake Storage Gen2 (ADLS Gen2)	50 TB de almacenamiento (Datos Crudos, Staging, y Logs). Redundancia LRS.	$1,000 - $1,500
Azure Managed Workflows for Apache Airflow (Orquestación)	1 entorno (3 nodos, 100+ DAGs)	$1,800 - $2,800
Azure SQL Database (Metadatos/Gobernanza)	Base de datos de Uso General, 8 vCores, 250 GB	$400 - $650

Servicio	Configuración Típica Asumida	Costo Mensual Estimado (USD)
Transferencia de Datos / *Egress* (Salida)	10 TB de salida de Azure a BI/otros servicios (después de la capa gratuita)	$500 - $800
Backup y Disaster Recovery (B&DR)	Snapshots y geo-redundancia para ADLS y Synapse (costo incremental)	$700 - $1,200
Azure Monitor / Log Analytics	Ingesta y retención de logs de 30 días para todos los servicios	$300 - $500
TOTAL	Estimación Mensual Base	$23,500 - $36,450

7. Consideraciones de Implementación

7.1. Seguridad y Cumplimiento

- **Aislamiento de Red**: Implementar la solución dentro de una Azure Virtual Network Usar Azure Private Link para que los servicios de Azure se comuniquen de forma privada sin exponerlos a la internet pública.
- **Gestión de Credenciales**: Almacenar todas las credenciales de los 50+ sistemas fuente en Azure Key Vault y acceder a ellas solo en tiempo de ejecución a través de Azure Data Factory o Airflow.
- **Identidad y Acceso**: Utilizar Azure Active Directory para gestionar el acceso a todos los servicios de la plataforma. Implementar el Principio del Mínimo Privilegio y Autenticación Multifactor.
- **Cumplimiento Normativo**: Garantizar que el manejo de datos de clientes cumpla con normativas como GDPR, implementando enmascaramiento o anonimización para datos sensibles.

7.2. Monitoreo y Observabilidad

- **Registro Centralizado**: Usar Azure Monitor y Log Analytics para consolidar logs de ADF, Synapse, Databricks y Airflow.
- Monitoreo de Pipelines:
    - **Airflow UI**: Proporciona la mejor vista del estado de la orquestación, dependencias y tiempos de ejecución de los 100+ pipelines.
    - **Alertas Críticas**: Configurar alertas en Azure Monitor para fallas de Airflow DAGs o cuando el tiempo de ejecución exceda la ventana de 4 horas.
- **Monitoreo de Performance**: Monitorear el uso de DWU de Synapse y la utilización de vCores de Databricks para optimizar costos y asegurar que el rendimiento cumpla con los SLAs diarios.

7.3. Escalabilidad y Performance

Cómputo Elástico:

- **Azure Synapse**: Usar la característica de pausa y reanudación o escalado automático de DWUs para pagar solo durante la ventana de procesamiento nocturno.
- **Azure Databricks**: Implementar el *auto-scaling* de *clusters* para manejar picos de carga durante las transformaciones complejas.
- **Optimización de Carga**: Priorizar la carga incremental. Para las fuentes que requieren *full refresh*, utilizar comandos PolyBase o COPY de Synapse para ingesta masiva de datos desde ADLS Gen2, maximizando la velocidad de carga.
- **Paralelismo**: Diseñar los DAGs de Airflow para explotar al máximo el paralelismo en la ingesta, especialmente para los datos de los 200 hoteles.

7.4. Disaster Recovery y Backup

- **Synapse DR**: Habilitar Geo-Redundancia para el *data warehouse* (*Synapse*) para tener copias de seguridad de las bases de datos en una región secundaria.
- **ADLS Gen2**: Configurar Geo-Redundant Storage (GRS) o Zone-Redundant Storage (ZRS) para los datos crudos y *staging*, garantizando la disponibilidad incluso en caso de fallas regionales.
- **Metadatos y Código**: Hacer *backup* regular de la base de datos de metadatos de Airflow y usar Azure DevOps/GitHub para el control de versiones (Git) de todo el código de las *pipelines* (Python, SQL, JSON de ADF).

7.5. Gobernanza de Datos

- **Catálogo de Datos**: Implementar Azure Purview para rastrear y catalogar los metadatos de todas las fuentes y transformaciones. Esto ayuda a los analistas a comprender el linaje de los datos (de qué PMS provienen, qué transformaciones se aplicaron) y facilita el cumplimiento.
- **Calidad de Datos**: Integrar chequeos de calidad de datos (ej. valores nulos, formatos inconsistentes, *outliers*) como tareas específicas dentro de los DAGs de Airflow/Databricks antes de cargar en la capa final de Synapse.
- **Definiciones Centralizadas**: Crear un Glosario Empresarial centralizado (en Purview) para homologar los términos clave (ej. 'Reserva', 'RevPAR', 'NPS') utilizados por los diferentes sistemas PMS y financieros, facilitando la creación de reportes ejecutivos consistentes.

8. Glosario de términos

8.1. Servicios y tecnologías de Azure

- **Azure Data Factory (ADF)**: Servicio de integración de datos que permite crear pipelines ETL/ELT para mover, transformar y cargar datos entre múltiples fuentes y destinos.
- **Azure Synapse Analytics**: Plataforma de análisis que combina almacenamiento de datos empresariales con procesamiento distribuido (SQL y Spark), ideal para BI y análisis avanzado.
- **Azure Data Lake Storage Gen2 (ADLS Gen2)**: Almacenamiento escalable para datos estructurados y no estructurados, compatible con formatos como Parquet y Delta Lake.
- **Azure Functions**: Servicio *serverless* que ejecuta código bajo demanda, útil para tareas específicas como transformaciones personalizadas o llamadas a APIs.
- **Azure Logic Apps**: Plataforma de automatización que permite integrar sistemas y APIs mediante flujos de trabajo visuales sin necesidad de escribir código.
- **Azure Monitor**: Servicio para supervisar el rendimiento y estado de recursos en Azure, con alertas configurables y dashboards personalizados.
- **Log Analytics**: Motor de análisis de logs que permite consultar, correlacionar y visualizar eventos de múltiples servicios Azure.
- **Azure Key Vault**: Servicio para almacenar y gestionar secretos, claves y credenciales de forma segura, con control de acceso granular.
- **Azure SQL Database**: Base de datos relacional como servicio, utilizada en este caso para *staging* o almacenamiento intermedio de datos estructurados.
- **Azure Event Grid**: Servicio de enrutamiento de eventos que permite activar flujos de trabajo o funciones en respuesta a cambios en recursos como *blobs*.
- **Azure IoT Hub**: Plataforma para conectar, monitorear y administrar dispositivos IoT, facilitando la ingestión de datos en tiempo real.
- **Azure Purview**: Solución de gobernanza de datos que permite catalogar, rastrear linaje y definir glosarios empresariales.

8.2. Componentes arquitectónicos

- **Pipeline ETL/ELT**: Flujo de trabajo que extrae, transforma y carga datos desde múltiples fuentes hacia un destino analítico.
- **DAG (Directed Acyclic Graph)**: Estructura de dependencias usada en orquestadores como Airflow para definir el orden de ejecución de tareas.
- **Lakehouse**: Arquitectura híbrida que combina Data Lake (almacenamiento flexible) y Data Warehouse (consultas optimizadas) en una sola solución.
- **Raw Zone**: Área de almacenamiento donde se guardan los datos tal como llegan desde las fuentes, sin transformaciones.
- **Staging Zone / Cleansed Zone**: Área intermedia donde los datos son transformados, normalizados y preparados para consumo.
- **Serving / Curated Zone**: Área final donde los datos están listos para ser consultados por herramientas de BI o APIs.
- **Feature Store**: Repositorio de variables o características utilizadas en modelos de Machine Learning, derivadas de datos curados.
- **Star Schema / Snowflake Schema**: Modelos de diseño de bases de datos para BI, donde los datos se organizan en tablas de hechos y dimensiones.

8.3. Patrones y procesos

- **Batch Processing**: Procesamiento por lotes, ideal para cargas nocturnas o fuentes *legacy*.
- **Streaming / Microbatch**: Procesamiento en tiempo real o casi real, útil para sensores IoT, redes sociales y APIs.
- **Lambda Architecture**: Combina procesamiento *batch* y *streaming* para ofrecer vistas históricas y en tiempo real.
- **Kappa Architecture**: Variante simplificada que usa solo *streaming* como fuente de datos.
- **ELT (Extract, Load, Transform)**: Patrón donde los datos se cargan primero y luego se transforman en el destino, aprovechando el poder de cómputo del *data warehouse*.
- **Backfill**: Reprocesamiento de datos históricos que no fueron correctamente cargados en su momento.
- **Carga incremental**: Técnica que solo procesa los datos nuevos o modificados desde la última ejecución.
- **Full refresh**: Reprocesamiento completo de una fuente de datos, reemplazando todo el contenido anterior.
- **Runbook**: Guía operativa para ejecutar tareas manuales o resolver incidencias en *pipelines*.

8.4. Métricas y gobernanza

- **SLAs (Service Level Agreements)**: Acuerdos que definen niveles mínimos de servicio como disponibilidad, tiempo de ejecución y éxito de procesos.
- **RBAC (Role-Based Access Control)**: Modelo de seguridad que asigna permisos según roles definidos en la organización.
- **Data Lineage**: Trazabilidad del recorrido de los datos desde su origen hasta su destino, incluyendo transformaciones aplicadas.
- **Data Catalog**: Repositorio que documenta las fuentes, estructuras y definiciones de los datos disponibles.
- **Glosario Empresarial**: Conjunto de definiciones estandarizadas para términos clave usados en reportes y análisis.
- **Data Quality Checks**: Validaciones aplicadas a los datos para detectar nulos, inconsistencias, duplicados o valores fuera de rango.
- **Auditoría**: Registro de accesos, cambios y ejecuciones para garantizar cumplimiento normativo y trazabilidad.

8.5. APIs y conectividad

- **REST API**: Interfaz de comunicación basada en HTTP, ampliamente usada para integrar sistemas SaaS como Booking o TripAdvisor.
- **ODBC / JDBC**: Protocolos estándar para conectar aplicaciones con bases de datos relacionales.
- **MQTT / AMQP**: Protocolos ligeros para comunicación entre dispositivos IoT y plataformas de ingestión.
- **OAuth 2.0**: Protocolo de autenticación segura usado para acceder a APIs públicas sin compartir credenciales directamente.
- **Linked Service**: Configuración en ADF que define cómo conectarse a una fuente o destino de datos.

8.6. Costos y rendimiento

- **DWU (Data Warehouse Unit)**: Unidad de medida del poder de cómputo en Azure Synapse; afecta el rendimiento y costo.
- **vCore**: Unidad de procesamiento virtual usada en servicios como Azure SQL o Databricks.
- **Auto-scaling**: Capacidad de ajustar automáticamente los recursos de cómputo según la carga de trabajo.
- **Geo-redundancia**: Técnica de replicación de datos en múltiples regiones para garantizar disponibilidad ante fallos.
- **Cool/Archive Tier**: Niveles de almacenamiento en Azure con menor costo, usados para datos históricos o poco accedidos.