Fecha: 17 de Julio de 2025

Informe Final del Proyecto: Sistema Distribuido de Gestión de Flotillas y Seguimiento Logístico "LogiTrack"

Integrantes:
- Joel Morales
- Emilio Muñoz
- Jhon Guadalupe
- Alejandro Chamba

1. Objetivo General

Diseñar, implementar y validar un prototipo funcional de un sistema de base de datos distribuida heterogénea para gestionar eficientemente las operaciones logísticas de la empresa ficticia "AndesExpress", demostrando la aplicación práctica de los principios de fragmentación, replicación y procesamiento de consultas distribuidas.

2. Objetivos Específicos

- **Modelar** la base de datos a nivel conceptual y lógico para representar los procesos de negocio de la empresa.
- **Definir** una estrategia de distribución de datos, aplicando técnicas de fragmentación horizontal y vertical para optimizar la localización y el rendimiento de las consultas.
- **Implementar** el esquema de la base de datos en un entorno heterogéneo, utilizando los sistemas gestores de bases de datos MySQL y Microsoft SQL Server.
- **Poblar** la base de datos con datos de ejemplo consistentes a través de los nodos replicados y fragmentados.
- **Probar** la funcionalidad del sistema mediante la ejecución de consultas distribuidas que requieran la combinación de datos de múltiples nodos.

3. Desarrollo

El proyecto se ejecutó siguiendo un ciclo de vida estructurado, abarcando desde el diseño conceptual hasta la implementación y prueba del sistema.

3.1. Diseño Conceptual

Se modelaron las entidades clave del negocio de "AndesExpress" para crear una estructura de datos robusta y normalizada. Las entidades definidas fueron: Centro_Operacion, Conductor, Vehiculo, Mantenimiento, Ruta y Entrega.

El siguiente Diagrama Entidad-Relación (E-R) representa la estructura final y las

relaciones entre las entidades:

3.2. Diseño Lógico

Se tradujo el modelo conceptual a un diseño lógico detallado, especificando los tipos de datos y las restricciones para cada atributo. Este diseño sirvió como el plano técnico para la implementación.

Diccionario de Datos:

Nombre de la Tabla	Nombre del Atributo	Tipo de Dato	Restricciones / Notas
Centro_Operacion	id_centro_op	INT	**PK**, NOT NULL, AUTO_INCREMENT
	nombre_centro	VARCHAR(100)	NOT NULL, UNIQUE
	ciudad	VARCHAR(50)	NOT NULL
Conductor	id_conductor	INT	**PK**, NOT NULL, AUTO_INCREMENT
	cedula	VARCHAR(10)	NOT NULL, UNIQUE
	Conductor_id_centro_op	INT	**FK** -> Centro_Operacion, NOT NULL
Vehiculo_Operativo	id_vehiculo	INT	**PK**, NOT NULL, AUTO_INCREMENT
	placa	VARCHAR(8)	NOT NULL, UNIQUE
	Vehiculo_id_centro_op	INT	**FK** -> Centro_Operacion, NOT NULL
Vehiculo_Mantenimiento	id_vehiculo	INT	**PK**, **FK** -> Vehiculo_Operativo
	marca	VARCHAR(50)	NOT NULL
	modelo	VARCHAR(50)	NOT NULL
Ruta	id_ruta	INT	**PK**, NOT NULL,

			AUTO_INCREMENT
	Ruta_id_vehiculo_asignado	INT	**FK** -> Vehiculo_Operativo, NOT NULL
	Ruta_id_conductor_asignado	INT	**FK** -> Conductor, NOT NULL
Entrega	id_entrega	INT	**PK**, NOT NULL, AUTO_INCREMENT
	Entrega_id_ruta_asignada	INT	**FK** -> Ruta, NOT NULL
	ciudad_destino	VARCHAR(50)	NOT NULL (Atributo de fragmentación)

(Nota: Se han omitido atributos no clave para brevedad en esta tabla resumen).

3.3. Diseño de la Distribución

Se definió una arquitectura distribuida con tres nodos geográficos (Quito, Guayaquil, Cuenca) y se aplicaron las siguientes estrategias de fragmentación:

- **Fragmentación Horizontal (Tabla Entrega):**
  - **Justificación:** El objetivo fue mejorar la localización de los datos, almacenando la información de las entregas en el nodo de su ciudad de destino para acelerar las consultas locales.
  - **Atributo de Fragmentación:** ciudad_destino.
  - **Predicados Simples:** Se definieron tres predicados mutuamente excluyentes, formando un conjunto completo y mínimo:
    - p1: ciudad_destino = 'Quito'
    - p2: ciudad_destino = 'Guayaquil'
    - p3: ciudad_destino = 'Cuenca'
  - **Asignación:** Cada fragmento resultante (Entrega_QTO, Entrega_GYE, Entrega_CUE) se asignó a su respectivo nodo.
- **Fragmentación Vertical (Tabla Vehiculo):**
  - **Justificación:** El objetivo fue mejorar el rendimiento separando los datos según su frecuencia de acceso. Los datos operativos (placa, ubicación) se consultan constantemente, mientras que los datos descriptivos (marca, modelo, año) se usan con menos frecuencia.
  - **Fragmentos:**

    1. Vehiculo_Operativo: Contiene los atributos de alta frecuencia.
    2. Vehiculo_Mantenimiento: Contiene los atributos de baja frecuencia.
  - **Asignación:** Para garantizar la transparencia y la disponibilidad total de la información de la flota, ambos fragmentos se **replicaron** en los tres nodos.

3.4. Implementación

El sistema se implementó en un entorno **Ubuntu Linux**, utilizando las siguientes tecnologías:

- **MySQL Server** para los nodos de Guayaquil y Cuenca.
- **Docker** para ejecutar un contenedor de **Microsoft SQL Server**, simulando el nodo heterogéneo de Quito.
- **MySQL Workbench** y **Azure Data Studio** como clientes para la administración de las bases de datos.

Se ejecutaron scripts SQL adaptados para cada gestor (MySQL y T-SQL) para crear los esquemas, implementar la fragmentación mediante tablas y vistas, y finalmente poblar la base de datos con un conjunto de datos consistente.

4. Resultados

El resultado principal del proyecto es un prototipo funcional del SBDD. La efectividad del sistema se validó mediante la ejecución de una consulta distribuida que simulaba un requerimiento de negocio real.

Escenario de Prueba: "Obtener el manifiesto de carga completo del vehículo con placa PBA-1111."

- **Paso 1 (Consulta en MySQL):** Se consultó la tabla Entrega en el nodo MySQL para encontrar los paquetes asignados a la ruta del vehículo.

```
SELECT e.* FROM AndesExpress.Entrega e
JOIN AndesExpress.Ruta r ON e.Entrega_id_ruta_asignada = r.id_ruta
JOIN AndesExpress.Vehiculo_Operativo vo ON r.Ruta_id_vehiculo_asignado = vo.id_vehiculo
WHERE vo.placa = 'PBA-1111';
```

- **Paso 2 (Consulta en SQL Server):** Se realizó la misma consulta en el nodo SQL Server.

```
SELECT e.* FROM AndesExpress.dbo.Entrega e
JOIN AndesExpress.dbo.Ruta r ON e.Entrega_id_ruta_asignada = r.id_ruta
JOIN AndesExpress.dbo.Vehiculo_Operativo vo ON r.Ruta_id_vehiculo_asignado = vo.id_vehiculo
WHERE vo.placa = 'PBA-1111'
```

Fecha: 17 de Julio de 2025

Resultado Obtenido:
La combinación de los resultados de ambos nodos arrojó el manifiesto completo, demostrando que el sistema es capaz de resolver consultas que requieren datos de múltiples fragmentos distribuidos en un entorno heterogéneo.

- **Resultado en MySQL (después de la sincronización):**

```
mysql> SELECT e.* FROM AndesExpress.Entrega e
    -> JOIN AndesExpress.Ruta r ON e.Entrega_id_ruta_asignada = r.id_ruta
    -> JOIN AndesExpress.Vehiculo_Operativo vo ON r.Ruta_id_vehiculo_asignado = vo.id_vehiculo
    -> WHERE vo.placa = 'PBA-1111';
+------------+--------------------------+---------------------+---------+--------------+---------------+-------------------------+----------------+--------------
------+
| id_entrega | Entrega_id_ruta_asignada | descripcion_paquete | peso_kg | ciudad_origen | ciudad_destino | direccion_destino       | estado_entrega | fecha_creacion
      |
+------------+--------------------------+---------------------+---------+--------------+---------------+-------------------------+----------------+--------------
------+
|          3 |                        1 | Suministros Médicos |  300.00 | Quito         | Guayaquil      | Malecón 2000, Local 5   | En bodega      | 2025-07-12 17:
54:39 |
|          5 |                        1 | Libros y Papelería  |  120.00 | Quito         | Cuenca         | Av. Remigio Crespo Toral | En bodega      | 2025-07-12 17:
54:39 |
|          6 |                        4 | Equipos de Sonido   |  440.00 | Quito         | Guayaquil      | Av. Emilito Estrada 312 | En tránsito    | 2025-07-12 18:
49:00 |
+------------+--------------------------+---------------------+---------+--------------+---------------+-------------------------+----------------+--------------
------+
3 rows in set (0.03 sec)
```

- **Resultado en SQL Server (después de la sincronización):**

```
select
    c.nombre AS NombreConductor,
    c.apellido as ApellidoConductor,
    vo.placa as PlacaVehiculo,
    r.fecha_ruta as FechaDeRuta
from
    AndesExpress.dbo.Ruta as r
JOIN
    AndesExpress.dbo.Conductor as c On r.Ruta_id_conductor_asignado = c.id_conductor
join
    AndesExpress.dbo.Vehiculo_Operativo as vo on r.Ruta_id_vehiculo_asignado = vo.id_vehiculo;
```

Results | Messages

		Entrega_id_ruta_a…	descripcion_paquete	peso_kg	ciudad_origen	ciudad_destino	direccion_destino	estado_entrega	fecha_creacion
1	3	1	Suministros Médicos	300.00	Quito	Guayaquil	Malecón 2000, Local 5	En tránsito	2025-07-12 22:57:36
2	5	1	Libros y Papelería	120.00	Quito	Cuenca	Av. Remigio Crespo Toral	En tránsito	2025-07-12 22:57:36
3	6	4	Equipos de Sonido	450.00	Quito	Guayaquil	Av. Victor Emilio Estrada 712	En bodega	2025-07-13 00:33:32

La sincronización manual de los datos (insertando el último registro en SQL Server) demostró el concepto de consistencia en datos replicados, un desafío clave en sistemas distribuidos.

5. Conclusiones

El desarrollo del proyecto "LogiTrack" permitió aplicar de manera exitosa los fundamentos teóricos de las bases de datos distribuidas en un caso práctico y realista. Se concluye que:

1. El diseño de una estrategia de **fragmentación** (horizontal y vertical) es crucial para optimizar el rendimiento y la localización de los datos, reduciendo la latencia y el tráfico de red en operaciones comunes.
2. La implementación en un **entorno heterogéneo** (MySQL y SQL Server) es viable pero introduce complejidades, como la necesidad de adaptar la sintaxis SQL y

utilizar diferentes herramientas de gestión, lo cual refleja los desafíos de integración en el mundo real.

3. La **replicación de datos** es una estrategia efectiva para garantizar la alta disponibilidad y la transparencia, aunque exige mecanismos para mantener la **consistencia** de los datos entre los nodos, como se evidenció durante las pruebas.
4. El prototipo final cumplió con todos los objetivos propuestos, demostrando ser un sistema funcional capaz de gestionar y consultar datos distribuidos de manera efectiva.

Fecha: 17 de Julio de 2025

6. Referencias

- Ceri, S., & Pelagatti, G. (1984). *Distributed Databases: Principles and Systems*. McGraw-Hill.
- Microsoft. (s.f.). *Documentación de Azure Data Studio*. Microsoft Learn. Recuperado el 17 de julio de 2025, de https://learn.microsoft.com/es-es/sql/azure-data-studio/
- Microsoft. (s.f.). *Inicio rápido: Ejecución de imágenes de contenedor de SQL Server con Docker*. Microsoft Learn. Recuperado el 17 de julio de 2025, de https://learn.microsoft.com/es-es/sql/linux/quickstart-install-connect-docker
- Oracle Corporation. (s.f.). *MySQL Documentation*. dev.mysql.com. Recuperado el 17 de julio de 2025, de https://dev.mysql.com/doc/
- Ozsu, M. T., & Valduriez, P. (2011). *Principles of Distributed Database Systems* (3.ª ed.). Springer.