

**PROYECTO INGENIERIA DE DATOS**

**EMPRESA INVERTRANS**

**INTEGRANTES:**

Juan Nicolas Arévalo Lemus

William Alejandro Ramírez Montaña

Juan Sebastián Arroyo Zúñiga

**Bogotá D.C**

**25/02/2025**

**TABLA DE CONTENIDOS**

**Resumen Ejecutivo** 4

**Necesidad y contextualización de la necesidad 4**

**Definición del problema** 4

**Objetivos** 5

Objetivo General

Objetivos Específicos

**Alcance de la aplicación**  5

**Alternativas de Solución**

Soluciones existentes

Posibles soluciones:

* Solución por Juan Arroyo
* Solución por Juan Arévalo
* Solución por William Ramirez

Alternativa de solución seleccionada

**Metodología**

**Limitaciones**

**Levantamiento de información y RQF’s**

**Mapa StakeHolders**

**Modelamiento de base de datos SQL**

* Diagrama de clases
* Modelo conceptual
* Modelo Lógico
* Modelo Físico
* Diccionario de datos
* Scripts DDL y DML en MySQL

**Modelamiento de Base de datos NoSQL**

* Modelo de arquitectura
* Modelo de componentes
* Script de las 3 formas de manipulación de bases de datos NoSQL

**Conclusiones por fases de desarrollo del proyecto.**

**Referencias**

**Anexos**

**Evaluación de soluciones propuestas y selección de opción definitiva**

**Evidencias de Gerencia**

* Evidencia de comunicación con el cliente
* Carta del cliente con percepción general del proyecto
* Bitácora de cumplimiento de cronograma – hitos del proyecto (Scrums u otro elemento de medida de desarrollo)
* Discusión sobre las alternativas de solución que exploraron y los criterios de selección para el diseño final.

**Evidencias de Diseño**

* Requerimientos implementados
* Modelos
* Descripción de las clases o elementos más relevantes

**Evidencias de Documentación**

* Manual de usuario
* Listado de funcionalidades
* Ejemplo guiado de ejecución
* Proceso de instalación

**Evidencias de Pruebas**

* Plan de pruebas propuesto
* Evidencias de aplicación de plan de pruebas

**Evidencias de Experiencia de usuario e Interfaz gráfica**

* Mapa de navegación de la aplicación (interfaces, pantallas, etc.)
* Descripción detallada de los elementos gráficos de la aplicación
* Encuesta de usabilidad con el cliente.

**RESUMEN EJECUTIVO**

*Somos estudiantes de MACC e Ingeniería de Sistemas de la Universidad del Rosario. Nuestro proyecto de semestre consiste en crear una base de datos relacional funcional, así como una no relacional. El cliente necesita esto específicamente porque tiene una compañía de transporte y hasta el día de hoy maneja toda la información relevante (facturas, conductores, historial de viajes) en un conjunto de hojas de Excel, lo cual es muy ineficiente ya que la información esta guardada en etiquetas no estandarizadas que hacen que sea muy difícil almacenar y acceder a la información de manera rápida y eficiente. Todo esto es muy perjudicial para el buen manejo de una empresa.*

*Por lo que planeamos hacer las bases de datos -relacional y no relacional como alternativa- desde cero con los conceptos aprendidos durante todo el semestre en la clase de ingeniería de datos. Nuestro objetivo es brindarle al cliente una base de datos funcional al final del semestre para que reemplace el anterior sistema de manejo de información dentro de la empresa y mejore el funcionamiento de esta. La información que manejaremos será la misma de las hojas de Excel, haciendo un ETL sobre esta para quitar variables repetidas y hacer la carga más manejable. Esto combinado con un workshop -un espacio de diálogo entre los ingenieros y el cliente en donde el cliente comparte que es lo que necesita y espera de la futura base de datos- son la base del trabajo. El proyecto será monitoreado por los propios miembros y la profesora haciendo uso de metodologías agiles como Kanban para mejorar la carga de trabajo y la eficiencia.*

**NECESIDAD Y CONTEXTUALIZACION DE LA NECESIDAD**

Nuestro cliente posee una empresa de transporte, y actualmente almacena toda la información como gastos, ganancias, nombre de clientes, conductores y placa de los vehículos en un Excel lo cual es un problema ya que no permite realizar comparaciones que determinarían decisiones como escoger entre uno u otro cliente, despedir conductores etc. Además, las búsquedas en Excel no son precisas ya que el registro de información da lugar a duplicados que significan lo mismo, un ejemplo de esto es la variación de la palabra “gastos parqueadero” que también se puede encontrar como “gasto parqueadero” “gastos parqueo” o simplemente “parqueo”. Lo que queremos brindarle a nuestro cliente es una base de datos que le permita realizar consultas y comparaciones con datos limpios para que así pueda generar reportes que le ayudaran a tomar decisiones clave.

**DEFINICIÓN DEL PROBLEMA**

Con esta en cuenta, el equipo de trabajo formuló la siguiente pregunta problema ¿Cómo podemos transformar la base de datos actual del cliente, construida en hojas de Excel con datos duplicados y etiquetas no estandarizadas, en un sistema eficiente y confiable que permita realizar consultas claras y tomar decisiones estratégicas a partir de datos limpios y organizados?

**OBJETIVOS**

**OBJETIVO GENERAL**

Construir una base de datos que facilite la comparación y consulta de información clave de una empresa de transporte, como viajes más rentables, vehículos con mayor cantidad de reparaciones o conductores con mayor número de infracciones de transito

**OBJETIVOS ESPECIFICOS**

* Construir el diagrama de clases para la base de datos de una empresa de transporte
* Proporcionar una base de datos limpia y organizada que centralice la información de la empresa de transporte
* Garantizar la capacidad de modificación y actualización de la base de datos
* Habilitar la consulta de facturas por datos esenciales como ganancias o cliente.
* Facilitar la consulta de conductores por cedula de ciudadanía
* Brindar la capacidad de consulta de vehículos por placa
* Autorizar la consulta de clientes por Id
* Conceder la consulta de conductores por número de multas
* Permitir consultar el valor promedio de un gasto por Id

**ALCANCE DE LA APLICACIÓN**

El presente proyecto tiene como objetivo diseñar e implementar una base de datos relacional y una base de datos no relacional para una empresa de transporte que actualmente gestiona su información en hojas de cálculo de Excel. A lo largo del semestre, se desarrollarán distintos módulos de trabajo, siguiendo metodologías ágiles (Kanban), y estimando los tiempos con base en **puntos de esfuerzo**, asignados conforme a la complejidad de cada tarea. A continuación, está la tabla mostrando lo que se pretende realizar en cada módulo a desarrollar:

|  |  |  |
| --- | --- | --- |
| **Módulo** | **Descripción** | **Puntos de Esfuerzo** |
| Workshop inicial | Reunión con el cliente para identificar necesidades, recopilar requerimientos y establecer criterios de éxito. | 8 (1 – 1.5 días) |
| Proceso ETL | Extracción de datos desde archivos Excel, limpieza de duplicados, estandarización de etiquetas y carga de datos estructurados. | 5 (1 día) |
| Modelado de base relacional | Diseño del diagrama entidad-relación, implementación en MySQL, inserción y validación de datos. | 13 (2 – 3 días) |
| Modelado de base no relacional | Diseño de estructura en MongoDB con documentos JSON, definición de colecciones y relaciones anidadas. | 13 (2 – 3 días) |
| Implementación de consultas | Desarrollo de consultas clave, triggers y procedimientos para facturación, vehículos, conductores, clientes y gastos promedio, tanto en SQL como en MongoDB. | 13 (2 – 3 días) |
| Pruebas y validación | Revisión funcional con el cliente, verificación de resultados esperados y ajustes necesarios | 7 (1 – 1.5 días) |
| Documentación y entrega | Elaboración de manual de uso, documentación técnica del modelo de datos, despliegue en GitHub y presentación final. | 13 (2- 3 días) |

***TABLA 1.*** ***Módulos para desarrollar***

El promedio de puntos de esfuerzo utilizados será de 72. Cada módulo incluirá revisiones periódicas, permitiendo retroalimentación tanto del cliente como de la profesora, para al final del semestre entregar un sistema funcional y documentado, con bases de datos que permitan a la empresa de transporte tomar decisiones estratégicas basadas en datos limpios y estructurados.

**Luego, como recursos humanos se necesitó de un equipo de desarrollo, conformado por** 4 estudiantes de MACC e Ingeniería Electrónica, encargados del proceso total del proyecto para entregarle a la cliente Sandra Zúñiga. Asimismo, se tuvo un acompañamiento y s**upervisión de la docente Tatiana Cabrera, encargada de la revisión constante del trabajo desarrollado y dar retroalimentaciones claves en búsqueda de mejoras en el trabajo del equipo.** En la siguiente tabla se encuentran las tecnologías y herramientas utilizadas:

|  |  |
| --- | --- |
| Base de datos relacional | MySQL |
| Base de datos no relacional | MongoDB |
| ETL | Excel |
| Backend | Node.js |
| Gestión ágil | Trello |
| Actualizaciones | GitHub |
| Comunicación con cliente | Zoom |

***TABLA 2. Tecnologías del proyecto junto a la herramienta usada por cada una***

**ALTERNATIVAS DE SOLUCIÓN**

**SOLUCIONES EXISTENTES**

A lo largo de múltiples estudios y proyectos, se ha evidenciado la importancia de contar con sistemas robustos de bases de datos, especialmente en sectores logísticos y empresariales donde el flujo de información es constante y crítico. Las bases de datos permiten estructurar, depurar y consultar la información de manera eficaz, lo que facilita la toma de decisiones, la generación de reportes y el análisis de desempeño. Como lo destacan varios artículos, entre ellos "La importancia de contar con información precisa, confiable y oportuna en las bases de datos", disponer de información bien organizada y actualizada no solo optimiza procesos internos, sino que también reduce errores, mejora la productividad y contribuye al crecimiento estratégico.

En nuestro caso, al trabajar con una empresa de transporte de carga que actualmente registra su información a través de hojas de Excel, nos enfrentamos a una problemática relacionada con la duplicación de datos, falta de estandarización y dificultad en el acceso y visualización de la información, esto ha dificultado tareas tan esenciales como la comparación de gastos por vehículo o la evaluación del rendimiento de conductores.

**POSIBLES SOLUCIONES**

Diversas posibles soluciones se plantearon durante las sesiones de trabajo grupal y en el workshop realizado con el cliente, todas con el objetivo de ofrecer una herramienta que centralice la información crítica de la empresa de manera clara y útil. A continuación, se presentan las principales alternativas evaluadas:

1. **Primera solución por Juan Arroyo**

La primera propuesta consistía en desarrollar un sistema enfocado únicamente en la gestión de conductores y vehículos. En esta alternativa se contemplaba la creación de dos módulos:

* **Conductores**: Permitía la búsqueda de conductores por documento, mostrando información como el número de multas y la cantidad de viajes realizados. Sin embargo, no se especificaban fechas ni detalles de las infracciones.
* **Vehículos**: Mostraba la cantidad de reparaciones realizadas por cada vehículo, sin detallar cuáles fueron.

Diagrama

El contenido generado por IA puede ser incorrecto.

**Ventaja:** Permitía tomar decisiones como despedir conductores o vender vehículos con base en su historial general.  
**Limitación:** No permitía hacer una correlación directa entre los gastos e ingresos asociados a estos elementos.

1. **Segunda solución por William Ramirez**

La segunda alternativa se enfocó en el seguimiento de viajes y el control de gastos asociados. Se propusieron dos módulos:

* Viajes: Describía el punto de origen y destino de cada viaje, además del número de escalas (sin detallar los lugares de escala).
* Gastos: Permitía consultar el precio promedio de los gastos por su ID. Este promedio podía actualizarse periódicamente, pero no se adaptaba a cada región del país.

Diagrama

El contenido generado por IA puede ser incorrecto.

**Ventaja:** Facilitaba el análisis general del costo promedio por viaje.  
**Limitación:** No permitía calcular con precisión la rentabilidad, ya que los gastos no eran contextualizados.

1. **Tercera solución por Juan Arévalo**

La tercera alternativa se basó en centralizar toda la información en un módulo de facturación, que relacionara los siguientes elementos: conductor, vehículo, cliente y gastos.  
Este módulo permitía:

* Calcular una utilidad estimada de cada viaje (restando gastos al pago inicial).
* Consultar información relevante para la toma de decisiones estratégicas como fijación de precios.

Diagrama

El contenido generado por IA puede ser incorrecto.

Ventaja: Permitía tener una visión global de las operaciones.  
Limitación: Dependía de que existieran datos provenientes de los otros módulos (conductores, vehículos, viajes, gastos).

**ALTERNATIVA DE SOLUCIÓN SELECCIONADA**

Se escogió como solución una aplicación que solo será de uso para administradores y supervisores, esta contará con 6 módulos, siendo estos: cliente, usuario, factura, gastos, conductor, vehículo, viaje. Cada módulo tiene información relevante que se relaciona en factura, donde estará registrada información de viajes como el documento del conductor, la placa del vehículo, el identificador del cliente y lo más relevante para nuestro cliente las utilidades y los gastos. La idea de esta aplicación es que los usuarios puedan consultar información como multas de un conductor, valor de un Soat, reparaciones de un vehículo etc. Con el fin de tomar decisiones importantes tales como despedir un conductor, vender un vehículo o fijar el precio de determinado viaje basándose en los gastos que este conlleva

Sobre los diferentes módulos: conductores, debe permitir su consulta por documento, accediendo a información como numero de multas, numero de viajes, sin embargo no se tendrá en cuenta la información específica de la fecha de multas; viajes, describe la información del lugar de origen y destino, además de la cantidad de escalas, sin embargo, no especifica los lugares de las escalas; gastos, permite consultar el precio por Id, el precio es promedio y actualizado con la frecuencia que el usuario desee pero al ser un promedio no se adecua a cada zona del país en específico; vehículos, tendrá información sobre el número de reparaciones pero no describirá cuales son. El usuario principalmente consultara el módulo facturas, pues es el que brinda información relacionando vehículos, conductores clientes y gastos, sin embargo, la utilidad total de un viaje se calculara restándole gastos al pago inicial, por lo que se obtendrá un aproximado

**LIMITACIONES**

* No se contempla la localización en tiempo real de vehículos ni pasajeros
* La información de cada viaje es definida por lugar de salida y llegada junto con duración, no se especificará cual es la ruta a tomar
* El cliente de la empresa de transporte no tendrá acceso a la aplicación

**METODOLOGIA**

Se uso la metodología kanban, y le evidencia esta en el siguiente **link**: <https://trello.com/invite/b/67a13ac3a9133e5b4ebd7466/ATTIa544d982ce8edae08ba40752628b4cd491A900F2/ing-datos>

FOTO DEL TABLERO

ALTERNATIVAS PROPUESTAS

ALTERNATIVA ELEGIDA

**LEVANTAMIENTO DE INFORMACIÓN Y REQUISITOS FUNCIONALES**

|  |  |
| --- | --- |
| **CODIGO** | **REQUISITO FUNCIONAL** |
| **RQF001** | **Nombre:** Cambiar estado del conductor |
| **Descripción:** La base de datos debe permitir al administrador inactivar o reactivar un conductor para cuando este no esté disponible porque está realizando un viaje, tiene incapacidad etc. |
| **Usuarios:** Administrador |

|  |  |
| --- | --- |
| **CODIGO** | **REQUISITO FUNCIONAL** |
| **RQF002** | **Nombre:** Agregar conductor |
| **Descripción:** La base de datos debe permitir al administrador agregar conductores para tener información importante como su número de contacto |
| **Usuarios:** Administrador |

|  |  |
| --- | --- |
| **CODIGO** | **REQUISITO FUNCIONAL** |
| **RQF003** | **Nombre:** Consultar Conductor por id único |
| **Descripción:** La base de datos debe permitir al administrador o supervisor consultar los conductores por documento para poder ver si están activos o conocer su número de viajes actuales |
| **Usuarios:** Administrador |

|  |  |
| --- | --- |
| **CODIGO** | **REQUISITO FUNCIONAL** |
| **RQF004** | **Nombre:** Consultar EPS conductor |
| **Descripción:** La base de datos debe permitir al administrador o supervisor consultar la EPS de un conductor por si llega a ocurrir algún accidente |
| **Usuarios:** Administrador |

|  |  |
| --- | --- |
| **CODIGO** | **REQUISITO FUNCIONAL** |
| **RQF005** | **Nombre:** Actualizar EPS conductor |
| **Descripción:** La base de datos debe permitir que el administrador actualice la EPS de un conductor por si este cambia de prestadora de salud en determinado momento |
| **Usuarios:** Administrador |

|  |  |
| --- | --- |
| **CODIGO** | **REQUISITO FUNCIONAL** |
| **RQF006** | **Nombre:** Agregar viaje |
| **Descripción:** La base de datos debe permitir al administrador agregarle 1 al número de viajes de un conductor cuando se realice la facturación para llevar un registro actualizado de su número de viajes |
| **Usuarios:** Administrador |

|  |  |
| --- | --- |
| **CODIGO** | **REQUISITO FUNCIONAL** |
| **RQF007** | **Nombre:** Actualizar número del conductor |
| **Descripción:** La base de datos debe permitir al administrador actualizar el número de contacto del conductor en caso de que este haya cambiado su número telefónico ya sea por robo perdida etc. |
| **Usuarios:** Administrador |

|  |  |
| --- | --- |
| **CODIGO** | **REQUISITO FUNCIONAL** |
| **RQF008** | **Nombre:** Consultar Viajes conductor |
| **Descripción:** La base de datos debe permitir que el administrador o supervisor consulte los viajes de un conductor para conocer su rendimiento |
| **Usuarios:** Administrador |

|  |  |
| --- | --- |
| **CODIGO** | **REQUISITO FUNCIONAL** |
| **RQF009** | **Nombre:** Consultar número multas |
| **Descripción:** La base de datos debe permitir al administrador o supervisor consultar las multas de un conductor para saber cuántos errores ha cometido mientras maneja en la empresa |
| **Usuarios:** Administrador |

|  |  |
| --- | --- |
| **CODIGO** | **REQUISITO FUNCIONAL** |
| **RQF010** | **Nombre:** Actualizar multas |
| **Descripción:** La base de datos debe permitir que el administrador actualice el número de multas para llevar un registro actualizado y evitar que se agreguen algunas que fueron desestimadas |
| **Usuarios:** Administrador |

|  |  |
| --- | --- |
| **CODIGO** | **REQUISITO FUNCIONAL** |
| **RQF011** | **Nombre:** Consultar duración viaje |
| **Descripción:** La base de datos debe permitir al administrador o supervisor consultar la duración de un viaje para poder saber por cuanto no estará disponible cierto vehículo y conductor |
| **Usuarios:** Administrador |

|  |  |
| --- | --- |
| **CODIGO** | **REQUISITO FUNCIONAL** |
| **RQF012** | **Nombre:** Actualizar duración viaje |
| **Descripción:** La base de datos debe permitir que el administrador actualice la duración de un viaje para llevar un correcto registro por si se cierran o abren nuevas vías |
| **Usuarios:** Administrador |

|  |  |
| --- | --- |
| **CODIGO** | **REQUISITO FUNCIONAL** |
| **RQF013** | **Nombre:** Cambiar escalas |
| **Descripción:** La base de datos debe permitir al administrador añadir o reducir la cantidad de escalas (si el viaje las posee) para llevar un registro actualizado de los itinerarios de viaje |
| **Usuarios:** Administrador |

|  |  |
| --- | --- |
| **CODIGO** | **REQUISITO FUNCIONAL** |
| **RQF014** | **Nombre:** Crear viaje |
| **Descripción:** La base de datos debe permitir al administrador crear viajes para poder conocer información clave como la duración estimada o el número de escalas |
| **Usuarios:** Administrador |

|  |  |
| --- | --- |
| **CODIGO** | **REQUISITO FUNCIONAL** |
| **RQF015** | **Nombre:** Cambiar escalas |
| **Descripción:** La base de datos debe permitir al administrador añadir o reducir la cantidad de escalas (si el viaje las posee) para llevar un registro actualizado de los itinerarios de viaje |
| **Usuarios:** Administrador |

|  |  |
| --- | --- |
| **CODIGO** | **REQUISITO FUNCIONAL** |
| **RQF016** | **Nombre:** Agregar vehículo |
| **Descripción:** La base de datos debe permitir al administrador agregar vehículos para tener información como la placa el color, valor de los impuestos etc. |
| **Usuarios:** Administrador |

|  |  |
| --- | --- |
| **CODIGO** | **REQUISITO FUNCIONAL** |
| **RQF017** | **Nombre:** Cambiar estado vehículo |
| **Descripción:** La base de datos debe permitir que el administrador cambie el estado de un vehículo entre activo o inactivo para saber si este está disponible para un viaje actual |
| **Usuarios:** Administrador |

|  |  |
| --- | --- |
| **CODIGO** | **REQUISITO FUNCIONAL** |
| **RQF018** | **Nombre:** Consultar descripción vehículo |
| **Descripción:** La base de datos debe permitir al administrador o supervisor consultar la descripción de un vehículo para que un cliente pueda identificarlo más fácil |
| **Usuarios:** Administrador, supervisor |

|  |  |
| --- | --- |
| **CODIGO** | **REQUISITO FUNCIONAL** |
| **RQF019** | **Nombre:** Consultar valor impuestos |
| **Descripción:** La base de datos debe permitir al administrador o supervisor consultar el valor del impuesto de un vehículo para saber cuánto se deberá pagar por el vehículo para cuando se acabe el año |
| **Usuarios:** Administrador, supervisor |

|  |  |
| --- | --- |
| **CODIGO** | **REQUISITO FUNCIONAL** |
| **RQF020** | **Nombre:** Actualizar valor impuestos |
| **Descripción:** La base de datos debe permitir que el administrador cambie el valor de los impuestos para llevar un registro actualizado ya que estos cambian anualmente |
| **Usuarios:** Administrador |

|  |  |
| --- | --- |
| **CODIGO** | **REQUISITO FUNCIONAL** |
| **RQF021** | **Nombre:** Consultar estado SOAT |
| **Descripción:** La base de datos debe permitir al administrador o supervisor consultar el estado del SOAT para saber si el vehículo requiere adquirir el seguro obligatorio |
| **Usuarios:** Administrador |

|  |  |
| --- | --- |
| **CODIGO** | **REQUISITO FUNCIONAL** |
| **RQF022** | **Nombre:** Consultar valor impuestos |
| **Descripción:** La base de datos debe permitir al administrador o supervisor consultar el valor del impuesto de un vehículo para saber cuánto se deberá pagar por el vehículo para cuando se acabe el año |
| **Usuarios:** Administrador |

|  |  |
| --- | --- |
| **CODIGO** | **REQUISITO FUNCIONAL** |
| **RQF023** | **Nombre:** Actualizar valor impuestos |
| **Descripción:** La base de datos debe permitir que el administrador cambie el valor de los impuestos para llevar un registro actualizado ya que estos cambian anualmente |
| **Usuarios:** Administrador |

|  |  |
| --- | --- |
| **CODIGO** | **REQUISITO FUNCIONAL** |
| **RQF020** | **Nombre:** Consultar estado SOAT |
| **Descripción:** La base de datos debe permitir al administrador o supervisor consultar el estado del SOAT para saber si el vehículo requiere adquirir el seguro obligatorio |
| **Usuarios:** Administrador |

|  |  |
| --- | --- |
| **CODIGO** | **REQUISITO FUNCIONAL** |
| **RQF021** | **Nombre:** Cambiar estado SOAT |
| **Descripción:** La base de datos debe permitir al administrador cambiar el estado del SOAT para tener información actualizada en caso de que este se haya vencido o adquirido nuevamente |
| **Usuarios:** Administrador |

|  |  |
| --- | --- |
| **CODIGO** | **REQUISITO FUNCIONAL** |
| **RQF022** | **Nombre:** Consultar número de reparaciones |
| **Descripción:** La base de datos debe permitir al administrador o al supervisor consultar el número de reparaciones, para saber si vale la pena seguir reparando el vehículo o es mejor adquirir uno nuevo |
| **Usuarios:** Administrador |

|  |  |
| --- | --- |
| **CODIGO** | **REQUISITO FUNCIONAL** |
| **RQF023** | **Nombre:** Agregar reparaciones |
| **Descripción:** La base de datos debe permitir que el administrador agregue a un vehículo las reparaciones que se le realizaron para llevar un registro actualizado |
| **Usuarios:** Administrador |

|  |  |
| --- | --- |
| **CODIGO** | **REQUISITO FUNCIONAL** |
| **RQF024** | **Nombre:** Consultar número de reparaciones |
| **Descripción:** La base de datos debe permitir al administrador o al supervisor consultar el número de reparaciones, para saber si vale la pena seguir reparando el vehículo o es mejor adquirir uno nuevo |
| **Usuarios:** Administrador |

|  |  |
| --- | --- |
| **CODIGO** | **REQUISITO FUNCIONAL** |
| **RQF025** | **Nombre:** Consultar valor gasto |
| **Descripción:** La base de datos debe permitir que el administrador o el supervisor consulte el valor de cierto gasto por ID para poder ver si el valor estimado se asemeja al que menciona el conductor |
| **Usuarios:** Administrador |

|  |  |
| --- | --- |
| **CODIGO** | **REQUISITO FUNCIONAL** |
| **RQF026** | **Nombre:** Consultar descripción gasto |
| **Descripción:** La base de datos debe permitir al administrador o supervisor consultar la descripción de un gasto para saber más específicamente que significa un id en palabras |
| **Usuarios:** Administrador |

|  |  |
| --- | --- |
| **CODIGO** | **REQUISITO FUNCIONAL** |
| **RQF027** | **Nombre:** Consultar facturas completas |
| **Descripción:** La base de datos debe permitir que el administrador consulte las facturas completas, mostrando toda la información relevante como valor, utilidad Lugar origen, destine conductor cliente y placa del vehículo |
| **Usuarios:** Administrador |

|  |  |
| --- | --- |
| **CODIGO** | **REQUISITO FUNCIONAL** |
| **RQF028** | **Nombre:** cambiar estado vehículo |
| **Descripción:** La base de datos debe permitir que el administrador alterne entre los estados "activo" e "inactivo" para evitar planear viajes con vehículos que están en viajes, en proceso de reparación ETC |
| **Usuarios:** Administrador |

|  |  |
| --- | --- |
| **CODIGO** | **REQUISITO FUNCIONAL** |
| **RQF029** | **Nombre:** consultar mejores cliente |
| **Descripción:** La base de datos debe permitir que el administrador consulte cuales son los clientes que mayor han pagado a lo largo del tiempo |
| **Usuarios:** Administrador |

|  |  |
| --- | --- |
| **CODIGO** | **REQUISITO FUNCIONAL** |
| **RQF030** | **Nombre:** Consultar el origen de las utilidades |
| **Descripción:** La base de datos debe permitir que el administrador consulte toda la tabla débil GastoFactura de una factura para poder evidenciar como se pasa de "Valor viaje" a "utilidades viaje" |
| **Usuarios:** Administrador |

|  |  |
| --- | --- |
| **CODIGO** | **REQUISITO FUNCIONAL** |
| **RQF031** | **Nombre:** consultar las multas registradas en el sistema |
| **Descripción:** La base de datos debe permitir que el administrador consulte cuales son los gastos que hacen referencia a multas |
| **Usuarios:** Administrador |

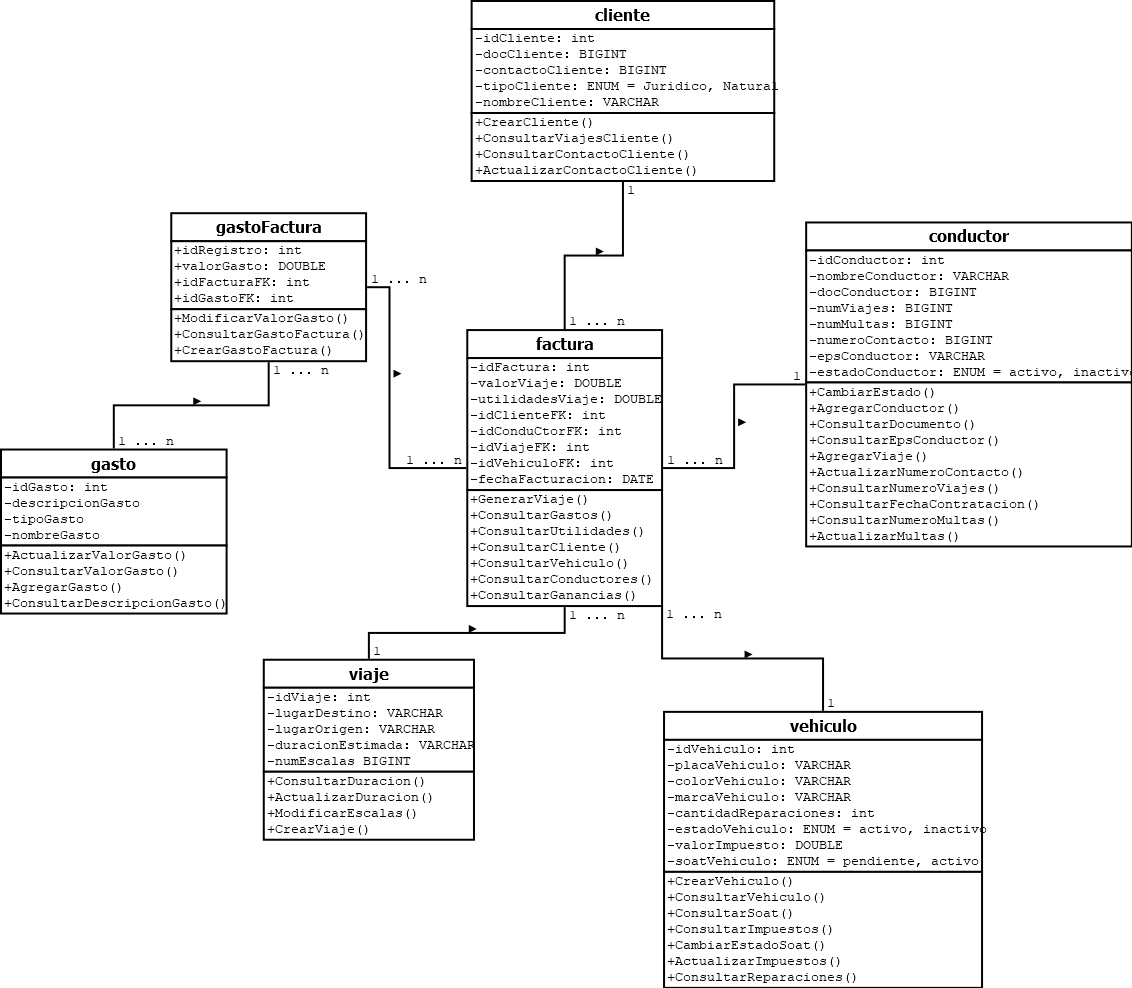
|  |  |
| --- | --- |
| **CODIGO** | **REQUISITO FUNCIONAL** |
| **RQF032** | **Nombre:** consultar mejores viajes |
| **Descripción:** La base de datos debe permitir consultar cuales son los viajes que en promedio dejan mayor utilidad |
| **Usuarios:** Administrador |

|  |  |
| --- | --- |
| **CODIGO** | **REQUISITO FUNCIONAL** |
| **RQF033** | **Nombre:** Consultar los lugares de salida más comunes |
| **Descripción:** La base de datos debe permitir que el administrador consulte cual es el lugar de origen más común de los viajes para organizar de manera correcta la localización de sus vehículos y conductores |
| **Usuarios:** Administrador |

|  |  |
| --- | --- |
| **CODIGO** | **REQUISITO FUNCIONAL** |
| **RQF034** | **Nombre:** consultar el número de cierto conductor |
| **Descripción:** La base de datos debe permitir que el administrador consulte cual es el número de contacto de cierto conductor en caso de imprevistos |
| **Usuarios:** Administrador |

|  |  |
| --- | --- |
| **CODIGO** | **REQUISITO FUNCIONAL** |
| **RQF035** | **Nombre:** consultar mejores viajes |
| **Descripción:** La base de datos debe permitir consultar cuales son los viajes que en promedio dejan mayor utilidad |
| **Usuarios:** Administrador |

**DIAGRAMA** **DE CLASES**

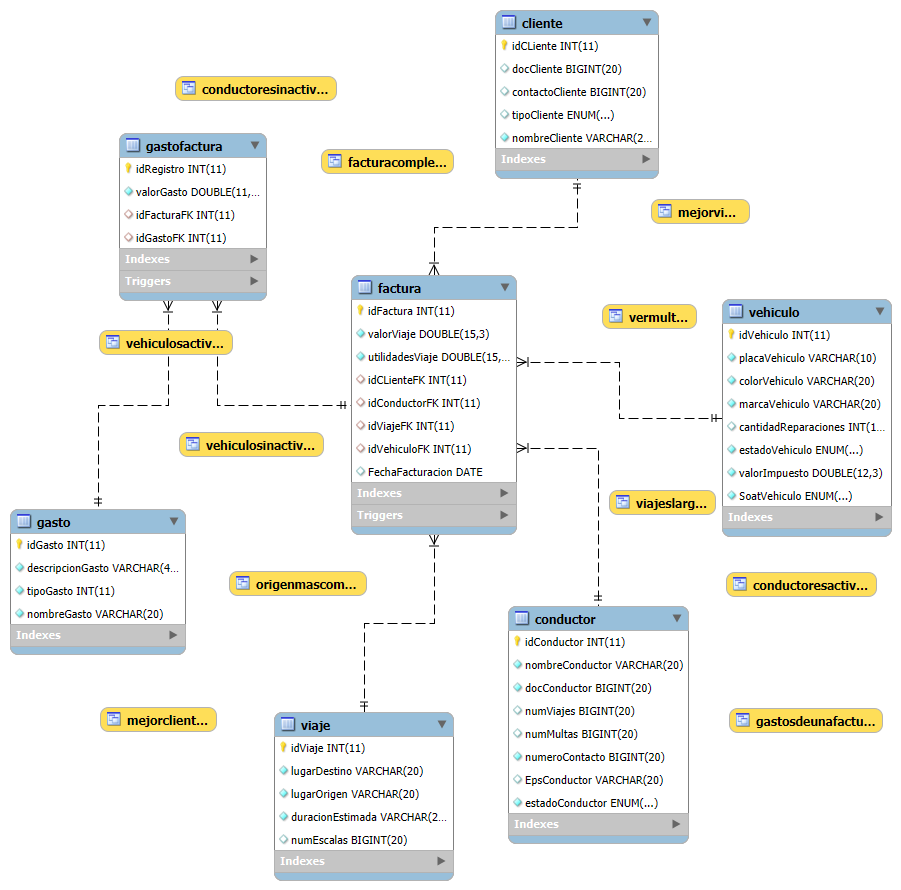
****

**MODELO CONCEPTUAL**

**Diagrama

El contenido generado por IA puede ser incorrecto.**

**MODELO LÓGICO**

****

**MAPA STAKEHOLDERS**

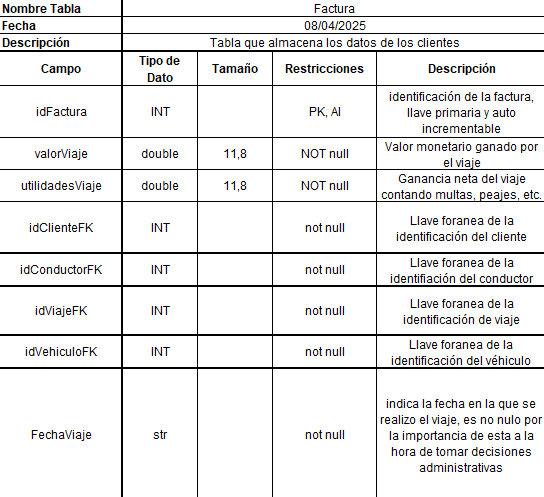
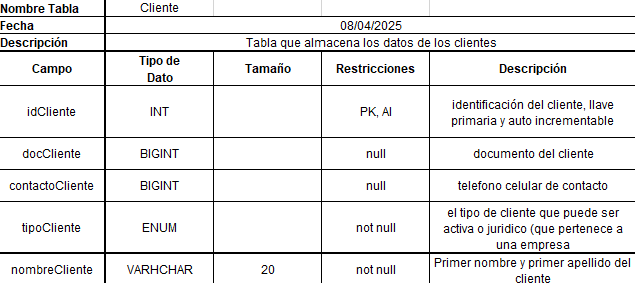
Imagen que contiene Interfaz de usuario gráfica

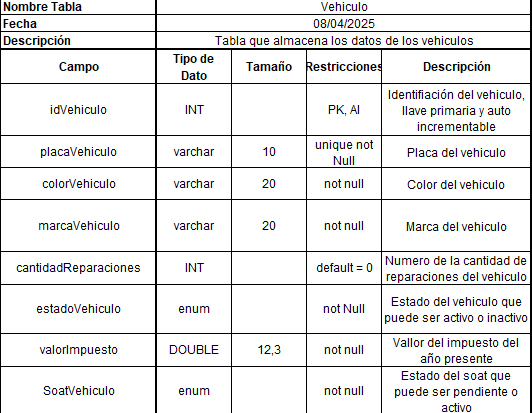
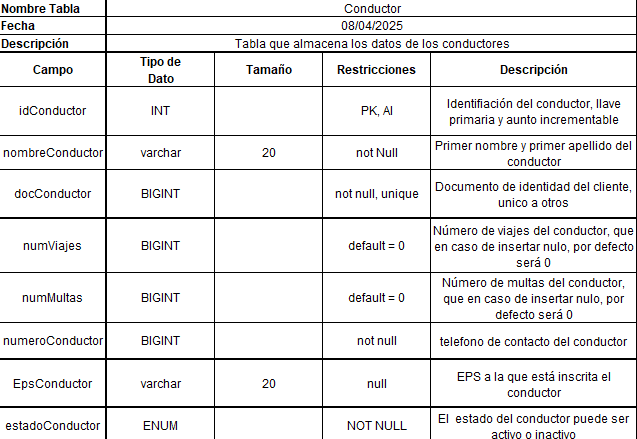
El contenido generado por IA puede ser incorrecto.

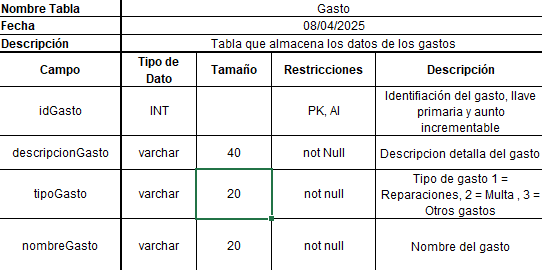
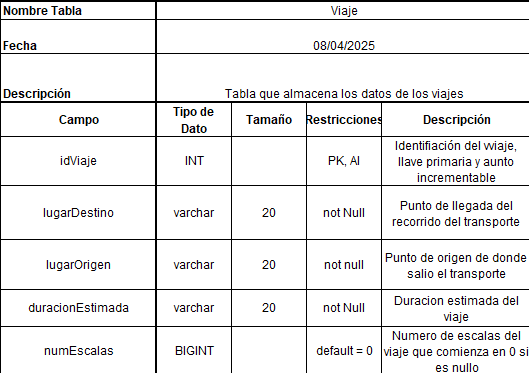
EXPLICACION

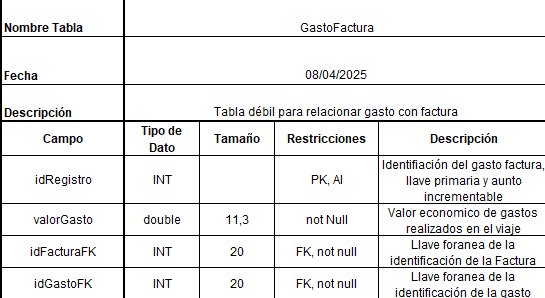
**ACTUALIZAR EL QUE TENEMOS**

**DICCIONARIO DE DATOS**









**IMPLEMENTACION DE LA SOLUCION EN MYSQL**

Para implementar la solución creamos las tablas descritas en el diagrama de clases, una vez se realizó la inserción de 50 datos en cada una de ellas se verifico que este proceso se hubiera realizado de manera correcta realizando las consultas básicas a todas las tablas, el resultado de esto fue el siguiente:

Tabla conductor:

Tabla

El contenido generado por IA puede ser incorrecto.

Tabla Vehículo:

Tabla

El contenido generado por IA puede ser incorrecto.

Tabla Cliente

Tabla

El contenido generado por IA puede ser incorrecto.

Tabla Viaje

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Tabla Gasto

Texto, Tabla

El contenido generado por IA puede ser incorrecto.

Tabla factura

Tabla

El contenido generado por IA puede ser incorrecto.

Tabla GastoFactura

Tabla

El contenido generado por IA puede ser incorrecto.

Sobre las consultas especificas a continuación se mostraran las 7 más relevantes que son: consultar clientes que sean de tipo jurídico, consultar conductores con alguna multa, consultar el viaje con mayor número de escalas, consultar gastos correspondientes a multas, consultar gastos correspondientes a reparaciones del vehículo, consultar conductores inactivos y consultar los 5 conductores con mayor número de viajes.

script consulta 1:

Texto

El contenido generado por IA puede ser incorrecto.

Resultado consulta 1

Interfaz de usuario gráfica, Aplicación, Tabla

El contenido generado por IA puede ser incorrecto.

script consulta 2:



Resultado consulta 2:

Interfaz de usuario gráfica, Tabla

El contenido generado por IA puede ser incorrecto.

script consulta 3:



Resultado Consulta 3:



Script consulta 4:

Texto

El contenido generado por IA puede ser incorrecto.

Resultado consulta 4:

Texto

El contenido generado por IA puede ser incorrecto.

Script 5:

Texto

El contenido generado por IA puede ser incorrecto.

Resultado consulta 5:

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Script 6:

Imagen de la pantalla de un celular con letras

El contenido generado por IA puede ser incorrecto.

Resultado consulta 6:

Tabla

El contenido generado por IA puede ser incorrecto.

Script 7:

Texto

El contenido generado por IA puede ser incorrecto.

Resultado Script 7:

Interfaz de usuario gráfica, Tabla

El contenido generado por IA puede ser incorrecto.

MODIFICACIONES

En el caso de las modificaciones la primera que surgió fue la de agregar el campo Eps a la tabla conductores puesto que fue un campo sugerido por la docente que nos pareció bastante correcto pero que no estaba en la tabla creada inicialmente

Script:



Ocurrió algo similar con la tabla gasto, donde estaba registrado un nombre conciso del gasto para poder identificarlo de manera sencilla pero en caso de necesitar una mayor profundidad, como por ejemplo el nombre “frenos” era fácil de identificar como un gasto referente a vehículo pero difícil saber si era un cambio total o un ajuste, para esto se agregó el campo “descripción gasto”



Las demás modificaciones que surgieron se realizaron porque había registros realizados de manera incorrecta y era mas eficiente realizar un update a volver a realizar las inserciones desde 0.

Script 3:

Imagen que contiene Escala de tiempo

El contenido generado por IA puede ser incorrecto.

Script 4:

Texto

El contenido generado por IA puede ser incorrecto.

Script 5:

Texto

El contenido generado por IA puede ser incorrecto.

Además, se realizó la eliminación de 2 registros que estaban duplicados:

Texto

El contenido generado por IA puede ser incorrecto.

Para las consultas multitablas la mayoría de estas se realizaron en la tabla factura, ya que como se menciono anteriormente esta es la que relaciona la mayor cantidad de clientes, sin embargo, otro parte se realizo en la tabla GastoFactura, Tabla que surgió durante la normalización y contenía información específica sobre el valor de los gastos que surgian en determinado viaje

Script 1:

Texto

El contenido generado por IA puede ser incorrecto.

Script 2:

Texto

El contenido generado por IA puede ser incorrecto.

Resultado script 2:



Además de estas consultas se realizaron procedimientos almacenados, vistas y triggers que agilizaban consultas, actualizaciones o modificaciones automáticas que serían parte del día a día de nuestro cliente.

Script 1, procedimiento para consultar cual es el gasto que mayor peso tuvo en la factura.

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Script 2, procedimiento para consultar toda la informacion de un conductor especifico.

Texto

El contenido generado por IA puede ser incorrecto.

Script 3,procedimiento para actualizar la eps de determinado conductor.

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Script 4, procedimiento para actualizar el estado de un conductor. Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Script 5, procedimiento para cambiar el estado del SOAT de determinado vehiculo

Texto

El contenido generado por IA puede ser incorrecto.

Script 6, vista que desde factura permite consultar la informacion completa, con nombre del conductor los clientes, placa del vehiculo etc.

Texto

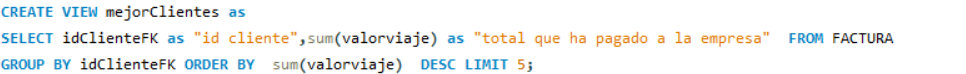
El contenido generado por IA puede ser incorrecto.

Resultado script 6:

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Script 7, vista que permite consultar cuales son los clientes que han aportado más dinero a la empresa

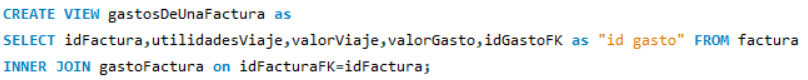


Resultado script 7:

Tabla

El contenido generado por IA puede ser incorrecto.

Script 8, vista que permite consultar todos los gastos de una factura incluyendo su valor, descripcion y tipo



Resultado script 8:

Tabla

El contenido generado por IA puede ser incorrecto.

Script 9, trigger que de manera automática añade 1 al numero de reparaciones a un vehículo una vez se realice una inserción con tipo de gasto 1 en gastoFactura

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Script 9, trigger que de manera automática añade 1 al numero de multas de un conductor una vez se realice una inserción con tipo de gasto 2 en gastoFactura

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

**REFERENCIAS**