

Sistema Integral de Gestión para Taller Mecánico Multi Sucursal

El presente sistema modela el funcionamiento de una empresa dedicada al mantenimiento y reparación de vehículos automotores. La organización comenzó como un taller mecánico que gestiona su información mediante registros manuales en papel. Con el crecimiento sostenido de la demanda y la incorporación de personal, se volvió necesario implementar un sistema informático basado en una base de datos relacional que permitiera centralizar, automatizar y asegurar la integridad de la información.

Inicialmente, el taller necesitaba registrar clientes, vehículos, reparaciones realizadas y facturación. Sin embargo, la gestión manual generaba inconvenientes tales como pérdida de información, duplicación de registros, falta de control del stock de repuestos, demoras en la emisión de facturas y ausencia de estadísticas para la toma de decisiones. Ante esta situación, se desarrolló un sistema computarizado que permite administrar de forma integral las operaciones técnicas, logísticas y administrativas del negocio.

1. Gestión de Clientes y Vehículos

El sistema registra clientes, quienes pueden poseer uno o varios vehículos. Cada vehículo se identifica de manera única y está asociado a una marca. Se almacena el historial completo de intervenciones realizadas sobre cada vehículo, permitiendo consultar reparaciones previas, servicios efectuados y facturación asociada.

2. Organización de Turnos

Para optimizar la atención y evitar sobrecarga operativa, el sistema permite gestionar turnos. Un turno es solicitado por un cliente para un vehículo específico en una sucursal determinada. El turno organiza el flujo de trabajo previo al ingreso formal del vehículo al taller.

3. Presupuestos

Antes de realizar un servicio, el taller puede generar un presupuesto estimativo. El presupuesto se asocia a uno o varios servicios ofrecidos por la empresa y queda vinculado al cliente. Este documento representa una estimación económica previa al ingreso del vehículo, sin estar todavía asociado a una orden de trabajo.

4. Órdenes de Trabajo

Cuando el vehículo ingresa efectivamente al taller, se genera una orden de trabajo. La orden:

- Está asociada a un vehículo
- Se asigna a un mecánico
- Pertenece a una sucursal
- Posee un estado (pendiente, en reparación, finalizada, etc.)

- Cada orden puede incluir múltiples servicios realizados, detallados individualmente.

5. Facturación y Pagos

Una vez finalizada la orden de trabajo, se emite una factura correspondiente.

El sistema registra los pagos realizados por el cliente, permitiendo pagos parciales y control de estados de facturación.

6. Gestión de Reclamos

El sistema contempla la posibilidad de registrar reclamos realizados por los clientes.

Un reclamo está asociado exclusivamente al cliente y al vehículo, independientemente de la orden de trabajo.

Esto permite documentar quejas, observaciones o inconvenientes posteriores al servicio.

7. Estructura Multisucursal

Con el crecimiento del negocio, la empresa amplió sus operaciones abriendo múltiples sucursales.

Cada sucursal:

- Posee su propio personal
- Atiende vehículos
- Genera órdenes de trabajo
- Administra su propio stock local de repuestos
- Registra sus propios costos operativos
- El sistema permite identificar y diferenciar todas las operaciones por sucursal.

8. Gestión de Almacén Central y Stock Distribuido

Para optimizar la logística de repuestos, la empresa implementó un almacén central encargado de concentrar el stock general.

El modelo contempla:

- Stock en almacén central
- Stock en cada sucursal
- Solicitudes de reposición desde sucursal hacia el almacén
- Descuento automático del stock al confirmar una transferencia

De esta manera, el sistema permite controlar inventario distribuido y trazabilidad de movimientos.

9. Proveedores y Pedidos de Repuestos

La empresa adquiere repuestos a distintos proveedores.

Un proveedor puede ofrecer múltiples repuestos y un repuesto puede estar disponible a través de varios proveedores.

El sistema permite:

- Registrar proveedores
- Asociar proveedores a repuestos
- Generar pedidos de compra al proveedor
- Registrar detalle de cantidades y precios
- Incrementar el stock del almacén al recibir la mercadería
- Esto permite mantener el historial de compras y analizar costos de abastecimiento.

Normalización

Primera Forma Normal (1FN)

Una tabla se encuentra en Primera Forma Normal (1FN) cuando todos sus atributos contienen valores atómicos, es decir, indivisibles; no existen grupos repetitivos ni atributos multivaluados, y cada registro puede identificarse de manera única mediante una clave primaria.

En el modelo del sistema de garage, la tabla **CLIENTES** cumple con la 1FN. Esta tabla está compuesta por los atributos: id_cliente (clave primaria), nombre, apellido, dni, telefono, email y fecha_registro. Cada campo almacena un único valor por registro; por ejemplo, el atributo teléfono contiene un solo número y no una lista de números. Además, no existen columnas que agrupen múltiples datos en un mismo campo, como podría ser “servicios_realizados” en formato de lista. Cada cliente se identifica de forma única mediante id_cliente, lo que garantiza la correcta individualización de los registros. Por lo tanto, la tabla CLIENTES respeta completamente los requisitos de la Primera Forma Normal.

Segunda Forma Normal (2FN)

Una tabla se encuentra en Segunda Forma Normal (2FN) cuando, además de cumplir la 1FN, todos los atributos no clave dependen completamente de la clave primaria. Esto significa que no deben existir dependencias parciales, especialmente en tablas cuya clave primaria sea compuesta.

En el modelo del garage, la tabla **DETALLE_PEDIDO_REPUESTO** es un buen ejemplo para analizar la 2FN. Esta tabla posee una clave primaria compuesta por id_pedido e id repuesto, y contiene además los atributos cantidad y precio_unitario. Ambos atributos dependen de la combinación completa de la clave compuesta y no solo de una parte de ella. Es decir, la cantidad solicitada no depende únicamente del pedido ni únicamente del repuesto, sino de la relación específica entre ambos. Si, por ejemplo, se almacenará el nombre del proveedor en esta tabla, existiría una dependencia parcial (ya que dependería sólo del pedido o del proveedor y no de toda la clave). Sin embargo, en el modelo actual dicha información se encuentra correctamente separada en su tabla correspondiente. Por lo tanto, DETALLE_PEDIDO_REPUESTO cumple con la Segunda Forma Normal.

Tercera Forma Normal (3FN)

Una tabla se encuentra en Tercera Forma Normal (3FN) cuando, además de cumplir la 2FN, no existen dependencias transitivas. Esto implica que ningún atributo no clave debe depender de otro atributo no clave, sino únicamente de la clave primaria.

Un ejemplo claro dentro del modelo es la tabla **ORDENES_TRABAJO**, que contiene los atributos: id_orden (clave primaria), fecha_ingreso, fecha_egreso, observaciones, id_vehiculo, id_mecanico, id_sucursal e id_estado. En este diseño, la descripción del estado de la orden no se encuentra almacenada en la misma tabla, sino en la tabla independiente **ESTADO_ORDEN**, que posee id_estado y descripción. De esta manera, se evita una dependencia transitiva del tipo: $\text{id_orden} \rightarrow \text{id_estado} \rightarrow \text{descripcion_estado}$. Si la descripción estuviera dentro de **ORDENES_TRABAJO**, cada cambio en el nombre del estado implicaría modificar múltiples registros, generando redundancia e inconsistencias. Al separar la información en tablas distintas y vincularlas mediante claves foráneas, se garantiza que todos los atributos no clave dependan únicamente de la clave primaria. Por lo tanto, la tabla **ORDENES_TRABAJO** cumple con la Tercera Forma Normal.

Conclusión

La realización del presente trabajo nos permitió aplicar de manera práctica los conceptos fundamentales del modelado de datos, la normalización y el diseño de bases de datos relacionales. A través del análisis del dominio del problema y la construcción del Diagrama Entidad-Relación, se logró comprender la importancia de estructurar la información de forma lógica, coherente y eficiente antes de su implementación en un sistema gestor de bases de datos.

El

proceso de normalización fue muy importante, ya que permitió identificar y eliminar redundancias, dependencias parciales y dependencias transitivas, garantizando así la integridad y consistencia de los datos. Comprender estos conceptos no solo mejora la calidad del diseño, sino que también previene problemas futuros como anomalías de actualización, inserción o eliminación, que pueden comprometer el correcto funcionamiento de un sistema real.

Asimismo, este trabajo fortaleció nuestra capacidad de análisis, ya que fue necesario interpretar un contexto organizacional, detectar entidades relevantes, definir relaciones y establecer reglas de negocio claras. Esto nos ayuda en nuestra formación profesional, dado que en el ámbito laboral los sistemas de información deben responder a necesidades concretas y bien definidas.