

Link GitHub:  
<https://github.com/matiasag1989>

# Entrega N° 1

## Planes de pago (Concesionaria)

**Alumno: Matias Alejandro Guerriero**

**Profesor: Alejandro Daniel Di Stefano**

**Tutor: Leonel Lo Presti**

**Fecha: 12 de agosto de 2025**

**Comisión: 81830**

### Contenido

Entrega N° 1: Planes de pago (Concesionaria) .....	1
1. Introducción .....	2
2. Objetivo.....	2
3. Situación problemática .....	2
4. Modelo de negocio .....	3
5. Diagrama Entidad–Relación .....	3
6. Listado de tablas .....	4

Link GitHub:

<https://github.com/matiasag1989>

## 1. Introducción

El proyecto consiste en el diseño y desarrollo de una base de datos para una concesionaria oficial de la marca Peugeot que comercializa vehículos 0 km, específicamente los modelos **Peugeot 208** y **Peugeot 2008**.

El sistema permitirá gestionar información de clientes, modelos, versiones, planes de financiación, entrega de vehículos usados y la relación entre estos elementos.

---

## 2. Objetivo

El objetivo principal es contar con una base de datos que centralice y organice la información necesaria para:

- Registrar clientes y sus datos de contacto.
  - Gestionar stock de modelos y versiones disponibles.
  - Administrar planes de cuotas (por ejemplo, 70/30 y otros).
  - Controlar operaciones con entrega de usados como parte de pago, incluyendo su valorización.
  - Vincular toda la información para facilitar consultas de estado de ventas, cuotas canceladas y saldos pendientes.
- 

## 3. Situación problemática

Actualmente, muchas concesionarias utilizan hojas de cálculo dispersas o sistemas fragmentados para registrar la información de clientes, autos y planes. Esto genera problemas como:

- Duplicación de datos.
- Errores en el seguimiento de cuotas.
- Falta de control centralizado sobre la entrega de usados.
- Dificultad para obtener reportes integrales de ventas y financiamiento.

Una base de datos relacional permitirá **integrar la información**, mejorar la trazabilidad de cada venta y optimizar los procesos administrativos y comerciales.

---

Link GitHub:

<https://github.com/matiasag1989>

## 4. Modelo de negocio

La concesionaria (de Peugeot) vende autos 0 km y ofrece diferentes planes de pago como posibilidad de financiación para acceder a un modelo nuevo de la marca.

El cliente elige:

1. El modelo y la versión del auto.
2. El plan de financiación.
3. Si entrega o no un vehículo usado como parte de pago.

El sistema registra todas las operaciones, calcula cuotas pendientes y almacena el valor de los usados para impactarlo en la cancelación parcial del plan.

---

## 5. Diagrama Entidad-Relación

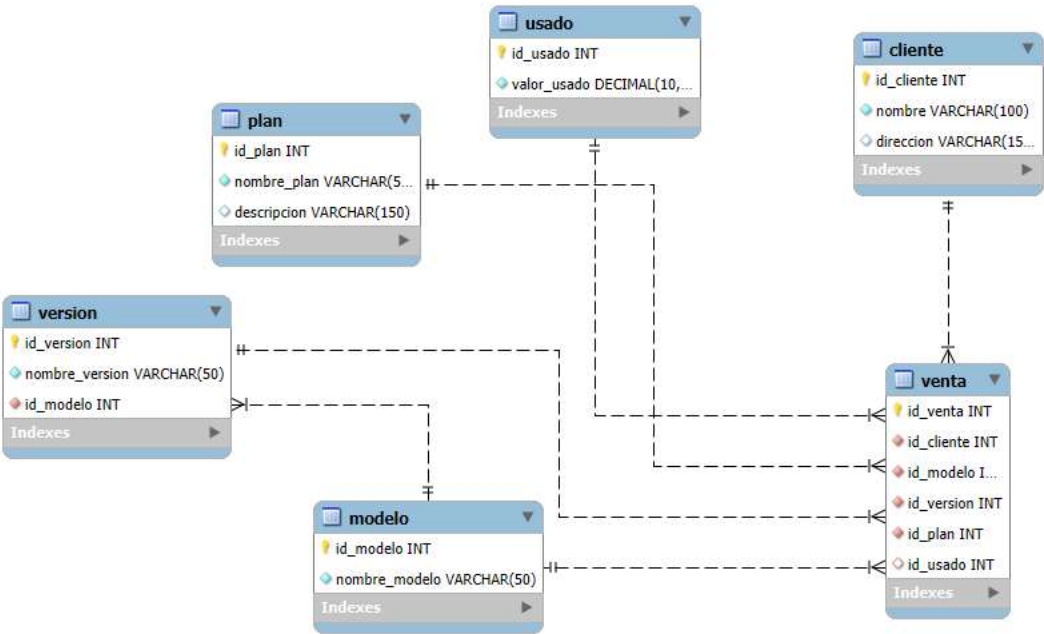
Entidades principales:

- **Cliente**
- **Modelo**
- **Versión**
- **Plan**
- **Venta** (relaciona cliente, modelo, versión y plan)
- **Usado** (opcional, vinculado a la venta)

Relaciones:

- Un cliente puede tener muchas ventas.
- Cada venta está asociada a un único plan, modelo y versión.
- Un usado puede estar o no asociado a la venta.

Link GitHub:  
<https://github.com/matiasag1989>



6. Listado de tablas

Tabla	Descripción	Campo	Nombre completo	Tipo de dato	Clave
Cliente	Datos de clientes	id_cliente	ID Cliente	INT	PK
		nombre	Nombre y apellido	VARCHAR(100)	
		direccion	Dirección del cliente	VARCHAR(150)	
Modelo	Modelos disponibles	id_modelo	ID Modelo	INT	PK
		nombre_modelo	Nombre del modelo (208, 2008)	VARCHAR(50)	
Versión	Versiones disponibles	id_version	ID Versión	INT	PK

Link GitHub:

<https://github.com/matiasag1989>

Tabla	Descripción	Campo	Nombre completo	Tipo de dato	Clave
Plan	Planes de financiación	nombre_version	Nombre de versión (Allure, GT)	VARCHAR(50)	
		id_plan	ID Plan	INT	PK
		nombre_plan	Nombre del plan (70/30)	VARCHAR(50)	
Usado	Vehículos usados entregados	descripcion	Descripción del plan	VARCHAR(150)	
		id_usado	ID Usado	INT	PK
Venta	Operaciones realizadas	valor_usado	Valor del vehículo usado	DECIMAL(10,2)	
		id_venta	ID Venta	INT	PK
		id_cliente	Cliente que realizó la compra	INT	FK
		id_modelo	Modelo elegido	INT	FK
		id_version	Versión elegida	INT	FK
		id_plan	Plan elegido	INT	FK
		id_usado	Vehículo usado (si aplica)	INT	FK (NULL)

Link GitHub:

<https://github.com/matiasag1989>

## Entrega N° 2

Este documento complementa el trabajo de la Entrega 1 y detalla las *Vistas*, *Funciones*, *Stored Procedures* y *Triggers* creados, junto con su objetivo y tablas involucradas. Se adjuntan dos scripts .SQL: uno de creación de objetos y otro de inserción de datos.

### Vistas

Vista	Objetivo	Tablas involucradas
v_ventas_detalle	Consolidar en una sola consulta el detalle de cada venta (cliente, modelo, versión, plan y, si corresponde, usado).	venta, cliente, modelo, version, plan, usado
v_ventas_con_usado	Filtrar ventas que incluyen vehículo usado como parte de pago.	v_ventas_detalle
v_ventas_sin_usado	Filtrar ventas que no incluyen vehículo usado.	v_ventas_detalle
v_ranking_modelos	Ranking de modelos por cantidad de ventas.	venta, modelo
v_ranking_planes	Ranking de planes por cantidad de ventas, con porcentaje sobre el total.	venta, plan

### Funciones Personalizadas

Función	Objetivo	Datos/Tablas
fn_tiene_usado(p_id_usado)	Devolver 1 si la venta incluye usado; 0 si no.	Parametro id_usado (venta)
fn_valor_usado_por_id(p_id_usado)	Obtener el valor del usado asociado a una venta.	Tabla usado
fn_porcentaje(p_parte, p_total)	Utilidad para calcular	N/A (utilidad general)

Link GitHub:

<https://github.com/matiasag1989>

porcentajes con 2  
decimales.

## Stored Procedures

Stored Procedure	Objetivo	Tablas involucradas
sp_registrar_venta(p_id_cliente, p_id_modelo, p_id_version, p_id_plan, p_id_usado)	Registrar una venta validando existencia de entidades y evitar reutilizar un mismo usado.	venta, cliente, modelo, version, plan, usado
sp_ventas_por_cliente(p_id_cliente)	Obtener el detalle de ventas de un cliente.	v_ventas_detalle
sp_resumen_ventas()	Entregar dos sets de resultados: ranking de modelos y ranking de planes.	v_ranking_modelos, v_ranking_planes

## Triggers

Trigger	Objetivo	Tabla/evento
trg_venta_usado_unico (BEFORE INSERT)	Evitar asociar el mismo vehículo usado a más de una venta.	venta (BEFORE INSERT)
trg_cliente_con_ventas_no_delete (BEFORE DELETE)	Impedir borrar clientes con ventas registradas.	cliente (BEFORE DELETE)
trg_cliente_trim_nombre (BEFORE INSERT)	Normalizar el nombre del cliente (TRIM) al registrar uno nuevo.	cliente (BEFORE INSERT)