

## Verificación de FN actualizado – Modelo Relacional AlpesCab

### Integrantes:

Andrés Javier Sanabria Garzón – 202411507

Juan Mateo Albarracín

Este documento contiene la ficha de verificación de 1FN, 2FN, 3FN y BCNF para cada relación del modelo relacional propuesto.

### Ciudad

#### Atributos / Dominio:

- idCiudad: NUMBER (PK autogenerada)
- nombre: VARCHAR2(100)

**Clave primaria (PK):** idCiudad

#### Dependencias funcionales (FD):

- idCiudad -> nombre
- nombre -> idCiudad (si UNIQUE)

#### Verificación de formas normales:

- 1FN: Sí — atributos atómicos.
- 2FN: Sí — PK simple.
- 3FN: Sí.
- BCNF: Sí.

**Conclusión:** Ciudad está en BCNF.

### PuntoGeografico

#### Atributos / Dominio:

- nombre: VARCHAR2(100)
- direccion: VARCHAR2(200)
- latitud: NUMBER(10,7)
- longitud: NUMBER(10,7)
- ciudad: NUMBER (FK -> Ciudad)

**Clave primaria (PK):** direccion

#### Dependencias funcionales (FD):

- direccion -> nombre, direccion, latitud, longitud, ciudad

#### Verificación de formas normales:

- 1FN: Sí.
- 2FN: Sí.
- 3FN: Sí.
- BCNF: Sí.

**Conclusión:** PuntoGeografico está en BCNF.

## Usuario

### Atributos / Dominio:

- cedula: VARCHAR2(20) (PK)
- nombre: VARCHAR2(150)
- correo: VARCHAR2(150)
- celular: VARCHAR2(30)
- tipo: VARCHAR2(20)
- calificacion: NUMBER(3,2) (opcional/derivado)
- comentario: VARCHAR2

**Clave primaria (PK):** cedula

### Dependencias funcionales (FD):

- cedula -> nombre, correo, celular, tipo, calificacion
- correo -> cedula (si UNIQUE)

### Verificación de formas normales:

- 1FN: Sí.
- 2FN: Sí.
- 3FN: Sí (calificacion es derivado).
- BCNF: Sí.

**Conclusión:** Usuario está en 3FN/BCNF; recomendar tratar calificacion como derivado.

## UsuarioConductor

### Atributos / Dominio:

- cedula: VARCHAR2(20) (PK, FK -> Usuario)

**Clave primaria (PK):** cedula

### Dependencias funcionales (FD):

- cedula -> licencia, ...

### Verificación de formas normales:

- 1FN: Sí.
- 2FN: Sí.
- 3FN: Sí.
- BCNF: Sí.

**Conclusión:** UsuarioConductor está en BCNF.

## UsuarioCliente

### Atributos / Dominio:

- cedula: VARCHAR2(20) (PK, FK -> Usuario)

**Clave primaria (PK):** cedula

### Dependencias funcionales (FD):

- cedula -> ...

### Verificación de formas normales:

- 1FN: Sí.
- 2FN: Sí.
- 3FN: Sí.
- BCNF: Sí.

**Conclusión:** UsuarioCliente está en BCNF.

## TarjetaCredito

### Atributos / Dominio:

- numero: NUMBER (PK)
- cedula: VARCHAR2(20) (FK -> Usuario)
- codigoSeguridad: VARCHAR2(20)
- nombre: VARCHAR2(120)
- fechaVencimiento: DATE

**Clave primaria (PK):** numero

### Dependencias funcionales (FD):

- numero -> cedula, nombre, fechaVencimiento

### Verificación de formas normales:

- 1FN: Sí.
- 2FN: Sí.

- 3FN: Sí.
- BCNF: Sí.

**Conclusión:** TarjetaCredito está en BCNF. No almacenar CVV; usar token.

## Vehiculo

### Atributos / Dominio:

- placa: VARCHAR2(20) (PK)
- cedula: VARCHAR2(20) (FK -> Usuario)
- tipo: VARCHAR2(50)
- marca: VARCHAR2(50)
- modelo: VARCHAR2(50)
- color: VARCHAR2(30)
- capacidadPasajeros: NUMBER
- ciudadExpedicion: NUMBER (FK -> Ciudad)
- registrado: NUMBER(1)

**Clave primaria (PK):** placa

### Dependencias funcionales (FD):

- placa -> cedula, tipo, marca, modelo, color, capacidadPasajeros, ciudadExpedicion, registrado

### Verificación de formas normales:

- 1FN: Sí.
- 2FN: Sí.
- 3FN: Sí.
- BCNF: Sí (si no hay reglas de derivación).

**Conclusión:** Vehiculo está en 3FN/BCNF.

## Disponibilidad

### Atributos / Dominio:

- dia: NUMBER (PK)
- placa: VARCHAR2(20) (FK -> Vehiculo)
- horaInicio: TIMESTAMP
- horaFin: TIMESTAMP
- tipoServicio: VARCHAR2(30)

**Clave primaria (PK):** dia

**Dependencias funcionales (FD):**

- dia -> placa, horalnicio, horaFin, tipoServicio

**Verificación de formas normales:**

- 1FN: Sí.
- 2FN: Sí.
- 3FN: Sí.
- BCNF: Sí.

**Conclusión:** Disponibilidad está en BCNF. Control de solapamiento por trigger.

**Servicio****Atributos / Dominio:**

- idServicio: NUMBER (PK)
- tipo: VARCHAR2(30)
- distancia: NUMBER(10,2)
- valorServicio: NUMBER(12,2) (derivado recomendado)
- horalnicio: TIMESTAMP
- horaFin: TIMESTAMP
- placa: VARCHAR2(20) (FK -> Vehiculo)

**Clave primaria (PK):** idServicio

**Dependencias funcionales (FD):**

- idServicio -> tipo, distancia, valorServicio, horalnicio, horaFin, placa
- valorServicio es derivado ( $\text{distancia} * \text{tarifa}$ )

**Verificación de formas normales:**

- 1FN: Sí.
- 2FN: Sí.
- 3FN: Sí (valorServicio es derivado).
- BCNF: Sí.

**Conclusión:** Servicio está en 3FN/BCNF; recomendar tratar valorServicio como derivado.

## TransportePasajeros

### Atributos / Dominio:

- idServicio: NUMBER (PK, FK -> Servicio)
- tarifaPasajeros: NUMBER(12,2)
- nivelTransporte: VARCHAR2(50)

**Clave primaria (PK):** idServicio

### Dependencias funcionales (FD):

- -servicio\_id -> tarifaPasajeros, nivelTransporte

### Verificación de formas normales:

- 1FN: Sí.
- 2FN: Sí.
- 3FN: Sí.
- BCNF: Sí.

**Conclusión:** TransportePasajeros está en BCNF.

## TransporteMercancias

### Atributos / Dominio:

- idServicio: NUMBER (PK, FK -> Servicio)
- tarifaMercancias: NUMBER(12,2)

**Clave primaria (PK):** idServicio

### Dependencias funcionales (FD):

- idServicio -> tarifaMercancias

### Verificación de formas normales:

- 1FN: Sí.
- 2FN: Sí.
- 3FN: Sí.
- BCNF: Sí.

**Conclusión:** TransporteMercancias está en BCNF.

## EntregaDomicilio

### Atributos / Dominio:

- idServicio: NUMBER (PK, FK -> Servicio)
- tarifaDomicilio: NUMBER(12,2)
- ubicacionRestaurante: NUMBER (FK -> PuntoGeografico)
- puntoUsuario: NUMBER (FK -> PuntoGeografico)

**Clave primaria (PK):** servicio\_id

### Dependencias funcionales (FD):

- idServicio -> tarifaDomicilio, ubicacionRestaurante, puntoUsuario

### Verificación de formas normales:

- 1FN: Sí.
- 2FN: Sí.
- 3FN: Sí.
- BCNF: Sí.

**Conclusión:** EntregaDomicilio está en BCNF.

## Rating

### Atributos / Dominio:

- idComentario: NUMBER (PK)
- idServicio: NUMBER (FK -> Servicio)
- revisor: VARCHAR2(20) (FK -> Usuario)
- revisado: VARCHAR2(20) (FK -> Usuario)
- rating: NUMBER(2,1)
- comentario: CLOB

**Clave primaria (PK):** idComentario

### Dependencias funcionales (FD):

- idComentario -> idServicio, revisor, revisado, rating, comentario, fecha

### Verificación de formas normales:

- 1FN: Sí.
- 2FN: Sí.
- 3FN: Sí.
- BCNF: Sí.

**Conclusión:** Revision está en BCNF.

## **AlpesCab (Opcional)**

### **Atributos / Dominio:**

- idSistema: NUMBER (PK)
- nombre: VARCHAR2(100)

**Clave primaria (PK):** idSistema

### **Dependencias funcionales (FD):**

- idSistema -> nombre

### **Verificación de formas normales:**

- 1FN: Sí.
- 2FN: Sí.
- 3FN: Sí.
- BCNF: Sí.

**Conclusión:** AlpesCab está en BCNF.



## Actualización: Verificación BCNF y Justificación UML → Relacional

Actualizado: 2025-09-28 02:44:39 UTC

Este anexo actualiza el modelo relacional con base en el modelo relacional modificado (archivo Excel) y añade la justificación completa de la transformación desde el UML. Se verifica explícitamente que cada relación cumple BCNF y se documentan las reglas de conversión (clases, atributos y asociaciones).

### Ciudad

#### Atributos / Dominio:

- ciudad\_id: NUMBER (PK autogenerada)
- nombre: VARCHAR2(100)

**Clave primaria (PK):** ciudad\_id

#### Dependencias funcionales (FD):

- ciudad\_id -> nombre
- nombre -> ciudad\_id (si UNIQUE)

#### Verificación de formas normales:

- 1FN: Sí — atributos atómicos.
- 2FN: Sí — PK simple.
- 3FN: Sí — no existen dependencias transitivas.
- BCNF: Sí — determinantes son claves.

**Conclusión:** Ciudad está en BCNF.

#### Notas / Recomendaciones:

- Asegurar UNIQUE(nombre) para evitar duplicados de ciudad.

### PuntoGeografico

#### Atributos / Dominio:

- punto\_id: NUMBER (PK autogenerada)
- nombre: VARCHAR2(100)
- direccion: VARCHAR2(200)
- latitud: NUMBER(10,7)
- longitud: NUMBER(10,7)
- ciudad\_id: NUMBER (FK -> Ciudad)

**Clave primaria (PK):** punto\_id

#### Dependencias funcionales (FD):

- punto\_id -> nombre, direccion, latitud, longitud, ciudad\_id

#### Verificación de formas normales:

- 1FN: Sí — atributos atómicos.
- 2FN: Sí — PK simple.
- 3FN: Sí — no hay dependencias transitivas.
- BCNF: Sí — determinantes son claves.

**Conclusión:** PuntoGeografico está en BCNF.

**Notas / Recomendaciones:**

- Se requiere FK a Ciudad para soportar consultas por ciudad y cumplir RF1 y RF7.

**Usuario**

**Atributos / Dominio:**

- cedula: VARCHAR2(20) (PK)
- nombre: VARCHAR2(150)
- correo: VARCHAR2(150)
- celular: VARCHAR2(30)
- tipo\_usuario: VARCHAR2(20) (CONDUCTOR/CLIENTE/AMBOS)
- calificacion\_promedio: NUMBER(3,2) (derivado, opcional)

**Clave primaria (PK):** cedula

**Dependencias funcionales (FD):**

- cedula -> nombre, correo, celular, tipo\_usuario, calificacion\_promedio
- correo -> cedula (si UNIQUE)

**Verificación de formas normales:**

- 1FN: Sí.
- 2FN: Sí.
- 3FN: Sí (calificacion\_promedio es derivado de Revision).
- BCNF: Sí.

**Conclusión:** Usuario está en BCNF.

**Notas / Recomendaciones:**

- Calificacion\_promedio se recomienda calcular por agregación desde Revision (no almacenar si se quiere evitar redundancia).

**UsuarioComentario**

**Atributos / Dominio:**

- comentario\_id: NUMBER (PK autogenerada)
- cedula: VARCHAR2(20) (FK -> Usuario)
- comentario: CLOB
- fecha: TIMESTAMP

**Clave primaria (PK):** comentario\_id

**Dependencias funcionales (FD):**

- comentario\_id -> cedula, comentario, fecha

**Verificación de formas normales:**

- 1FN: Sí.
- 2FN: Sí.

- 3FN: Sí.
- BCNF: Sí.

**Conclusión:** UsuarioComentario está en BCNF.

#### **UsuarioConductor**

##### **Atributos / Dominio:**

- cedula: VARCHAR2(20) (PK, FK -> Usuario)
- licencia: VARCHAR2(50) (opcional)

**Clave primaria (PK):** cedula

##### **Dependencias funcionales (FD):**

- cedula -> licencia, otros atributos específicos

##### **Verificación de formas normales:**

- 1FN: Sí.
- 2FN: Sí.
- 3FN: Sí.
- BCNF: Sí.

**Conclusión:** UsuarioConductor está en BCNF.

#### **Notas / Recomendaciones:**

- Herencia implementada con tabla por subclase y PK compartida con Usuario (evita duplicación de datos).

#### **UsuarioCliente**

##### **Atributos / Dominio:**

- cedula: VARCHAR2(20) (PK, FK -> Usuario)

**Clave primaria (PK):** cedula

##### **Dependencias funcionales (FD):**

- cedula -> (atributos cliente específicos si existen)

##### **Verificación de formas normales:**

- 1FN: Sí.
- 2FN: Sí.
- 3FN: Sí.
- BCNF: Sí.

**Conclusión:** UsuarioCliente está en BCNF.

#### **TarjetaCredito**

##### **Atributos / Dominio:**

- tarjeta\_id: NUMBER (PK autogenerada)
- usuario\_cedula: VARCHAR2(20) (FK -> Usuario)

- numero\_mask: VARCHAR2(20) (últimos 4 dígitos)
- nombre\_tarjeta: VARCHAR2(120)
- fecha\_vencimiento: DATE
- token\_pago: VARCHAR2(200)

**Clave primaria (PK):** tarjeta\_id

**Dependencias funcionales (FD):**

- tarjeta\_id -> usuario\_cedula, numero\_mask, nombre\_tarjeta, fecha\_vencimiento, token\_pago

**Verificación de formas normales:**

- 1FN: Sí.
- 2FN: Sí.
- 3FN: Sí.
- BCNF: Sí.

**Conclusión:** TarjetaCredito está en BCNF.

**Notas / Recomendaciones:**

- No almacenar CVV; almacenar solo máscara o token por seguridad.

#### **Vehiculo**

**Atributos / Dominio:**

- placa: VARCHAR2(20) (PK)
- conductor\_cedula: VARCHAR2(20) (FK -> Usuario)
- tipo: VARCHAR2(50)
- marca: VARCHAR2(50)
- modelo: VARCHAR2(50)
- color: VARCHAR2(30)
- capacidad\_pasajeros: NUMBER
- ciudad\_expedicion: NUMBER (FK -> Ciudad)
- nivel\_transporte: VARCHAR2(50)
- registrado: NUMBER(1)

**Clave primaria (PK):** placa

**Dependencias funcionales (FD):**

- placa -> conductor\_cedula, tipo, marca, modelo, color, capacidad\_pasajeros, ciudad\_expedicion, nivel\_transporte, registrado

**Verificación de formas normales:**

- 1FN: Sí.
- 2FN: Sí.
- 3FN: Sí (guardar nivel\_transporte puede ser redundante si se deriva de capacidad).
- BCNF: Sí.

**Conclusión:** Vehiculo está en 3FN/BCNF.

**Notas / Recomendaciones:**

- Si nivel\_transporte se puede derivar de capacidad\_pasajeros, considerar no almacenarlo y calcularlo al vuelo.

**Disponibilidad****Atributos / Dominio:**

- disponibilidad\_id: NUMBER (PK autogenerada)
- vehiculo\_placa: VARCHAR2(20) (FK -> Vehiculo)
- fecha: DATE
- hora\_inicio: TIMESTAMP
- hora\_fin: TIMESTAMP
- tipo\_servicio: VARCHAR2(30)

**Clave primaria (PK):** disponibilidad\_id

**Dependencias funcionales (FD):**

- disponibilidad\_id -> vehiculo\_placa, fecha, hora\_inicio, hora\_fin, tipo\_servicio

**Verificación de formas normales:**

- 1FN: Sí.
- 2FN: Sí.
- 3FN: Sí.
- BCNF: Sí.

**Conclusión:** Disponibilidad está en BCNF.

**Notas / Recomendaciones:**

- Validar solapamiento mediante trigger o lógica de aplicación para garantizar integridad temporal por conductor.

**Servicio****Atributos / Dominio:**

- servicio\_id: NUMBER (PK autogenerada)
- tipo\_servicio: VARCHAR2(30) (PASAJEROS| MERCANCIAS| DOMICILIO)
- distancia\_km: NUMBER(10,2)
- valor\_servicio: NUMBER(12,2) (derivado recomendado)
- hora\_inicio: TIMESTAMP
- hora\_fin: TIMESTAMP
- estado: VARCHAR2(20) (PENDIENTE| ASIGNADO| EN\_CURSO| FINALIZADO| CANCELADO)
- cliente\_cedula: VARCHAR2(20) (FK -> Usuario)
- conductor\_cedula: VARCHAR2(20) (FK -> Usuario, NULL hasta asignación)
- vehiculo\_placa: VARCHAR2(20) (FK -> Vehiculo, NULL hasta asignación)

**Clave primaria (PK):** servicio\_id

**Dependencias funcionales (FD):**

- servicio\_id -> tipo\_servicio, distancia\_km, valor\_servicio, hora\_inicio, hora\_fin, estado, cliente\_cedula, conductor\_cedula, vehiculo\_placa
- valor\_servicio es derivado:  $\text{valor\_servicio} = \text{distancia\_km} * \text{tarifa}(\text{tipo/nivel})$

#### **Verificación de formas normales:**

- 1FN: Sí.
- 2FN: Sí.
- 3FN: Sí (valor\_servicio derivado).
- BCNF: Sí.

**Conclusión:** Servicio está en 3FN/BCNF.

#### **Notas / Recomendaciones:**

- Tratar valor\_servicio como derivado o mantener sincronizado mediante trigger para evitar inconsistencias.

##### **ServicioPunto**

#### **Atributos / Dominio:**

- servicio\_punto\_id: NUMBER (PK autogenerada)
- servicio\_id: NUMBER (FK -> Servicio)
- punto\_id: NUMBER (FK -> PuntoGeografico)
- tipo: VARCHAR2(20) (PARTIDA|LLEGADA|OTRO)
- orden: NUMBER

**Clave primaria (PK):** servicio\_punto\_id (o PK compuesta: servicio\_id + orden)

#### **Dependencias funcionales (FD):**

- servicio\_punto\_id -> servicio\_id, punto\_id, tipo, orden
- (servicio\_id, orden) -> punto\_id, tipo (si PK compuesta)

#### **Verificación de formas normales:**

- 1FN: Sí.
- 2FN: Sí.
- 3FN: Sí.
- BCNF: Sí.

**Conclusión:** ServicioPunto está en BCNF.

#### **Notas / Recomendaciones:**

- Modela múltiples puntos de llegada/partida y preserva orden; necesario para RFC que contabiliza servicios por ciudad.

##### **TransportePasajeros**

#### **Atributos / Dominio:**

- servicio\_id: NUMBER (PK, FK -> Servicio)
- tarifa\_pasajeros: NUMBER(12,2)
- nivel\_transporte: VARCHAR2(50)

**Clave primaria (PK):** servicio\_id

#### **Dependencias funcionales (FD):**

- servicio\_id -> tarifa\_pasajeros, nivel\_transporte

**Verificación de formas normales:**

- 1FN: Sí.
- 2FN: Sí.
- 3FN: Sí.
- BCNF: Sí.

**Conclusión:** TransportePasajeros está en BCNF.

**TransporteMercancias**

**Atributos / Dominio:**

- servicio\_id: NUMBER (PK, FK -> Servicio)
- tarifa\_mercancias: NUMBER(12,2)

**Clave primaria (PK):** servicio\_id

**Dependencias funcionales (FD):**

- servicio\_id -> tarifa\_mercancias

**Verificación de formas normales:**

- 1FN: Sí.
- 2FN: Sí.
- 3FN: Sí.
- BCNF: Sí.

**Conclusión:** TransporteMercancias está en BCNF.

**EntregaDomicilio**

**Atributos / Dominio:**

- servicio\_id: NUMBER (PK, FK -> Servicio)
- tarifa\_domicilio: NUMBER(12,2)
- ubicacion\_restaurante: NUMBER (FK -> PuntoGeografico)
- punto\_usuario: NUMBER (FK -> PuntoGeografico)

**Clave primaria (PK):** servicio\_id

**Dependencias funcionales (FD):**

- servicio\_id -> tarifa\_domicilio, ubicacion\_restaurante, punto\_usuario

**Verificación de formas normales:**

- 1FN: Sí.
- 2FN: Sí.
- 3FN: Sí.
- BCNF: Sí.

**Conclusión:** EntregaDomicilio está en BCNF.

## **Revision**

### **Atributos / Dominio:**

- revision\_id: NUMBER (PK autogenerada)
- servicio\_id: NUMBER (FK -> Servicio)
- revisor\_cedula: VARCHAR2(20) (FK -> Usuario)
- revisado\_cedula: VARCHAR2(20) (FK -> Usuario)
- rating: NUMBER(2,1)
- comentario: CLOB
- fecha: TIMESTAMP

**Clave primaria (PK):** revision\_id

### **Dependencias funcionales (FD):**

- revision\_id -> servicio\_id, revisor\_cedula, revisado\_cedula, rating, comentario, fecha
- (servicio\_id, revisor\_cedula) -> rating, comentario (si se impone unicidad)

### **Verificación de formas normales:**

- 1FN: Sí.
- 2FN: Sí.
- 3FN: Sí.
- BCNF: Sí.

**Conclusión:** Revision está en BCNF.

### **Notas / Recomendaciones:**

- Imponer CHECK(rating BETWEEN 0 AND 5) y, opcionalmente, UNIQUE(servicio\_id, revisor\_cedula) si se limita a una revisión por revisor por servicio.

## **AlpesCab**

### **Atributos / Dominio:**

- sistema\_id: NUMBER (PK autogenerada)
- nombre: VARCHAR2(100)

**Clave primaria (PK):** sistema\_id

### **Dependencias funcionales (FD):**

- sistema\_id -> nombre

### **Verificación de formas normales:**

- 1FN: Sí.
- 2FN: Sí.
- 3FN: Sí.
- BCNF: Sí.

**Conclusión:** AlpesCab está en BCNF.



## Justificación de definición de tablas a partir del UML

A continuación se documenta la conversión de cada clase del UML al modelo relacional, especificando qué atributos se trasladaron, cómo se resolvieron las asociaciones (cardinalidad y obligatoriedad), y por qué se eligió la estructura final de tablas. Se sigue la estrategia: cada clase → tabla; atributos multivaluados → tabla separada; asociaciones 1:N → FK en el lado N; asociaciones M:N → tabla intermedia; herencia → tabla por subclase con PK compartida.

### Usuario / UsuarioConductor / UsuarioCliente

- La clase abstracta Usuario se mapea a la tabla Usuario (PK = cedula).
- Atributos simples (nombre, correo, celular, tipo) se trasladan como columnas atómicas.
- Herencia: UsuarioConductor y UsuarioCliente implementadas como tablas por subclase con PK compartida (cedula). Esto refleja la obligatoriedad: un Usuario puede ser conductor y/o cliente; cuando existe información específica del conductor (licencia), queda en UsuarioConductor.
- Relaciones: UsuarioConductor 1..\* Vehiculo → en Vehiculo se guarda conductor\_cedula como FK (lado N tiene la FK).

### Vehiculo y Disponibilidad

- Vehiculo se mapea a tabla Vehiculo (PK = placa) porque placa es identificador natural único.
- Operaciones/Disponibilidades (lista) se normalizan en la tabla Disponibilidad (Entidad propia) con FK vehiculo\_placa hacia Vehiculo, resolviendo multivaluados.
- Cardinalidad: Vehiculo 1..\* Disponibilidad reflejada con FK en Disponibilidad; obligatoriedad: una disponibilidad requiere un vehículo existente (FK NOT NULL).
- La ciudad de expedición usa FK a Ciudad para cumplir RF1 y permitir consultas por ciudad.

### Ciudad y PuntoGeografico

- Ciudad es entidad independiente (PK = ciudad\_id) porque el enunciado pide registro de ciudades (RF1).
- PuntoGeografico es entidad con coordenadas y FK a Ciudad; se modela como tabla PuntoGeografico (PK = punto\_id).
- Asociaciones: ServicioPunto referencia a PuntoGeografico para modelar múltiples puntos por servicio (1..\*), preservando orden con atributo orden.

### Servicio y subtipos

- Servicio es la entidad transaccional central (PK = servicio\_id).
- Subtipos (TransportePasajeros, TransporteMercancias, EntregaDomicilio) se implementan como tablas por subtipo con PK compartida servicio\_id, para evitar columnas NULL en Servicio y reflejar atributos específicos.
- Asociaciones: Servicio tiene FK a Usuario (cliente) y a Vehiculo/UsuarioConductor cuando el servicio está asignado.
- ServicioPunto implementa múltiples destinos y mantiene orden, apoyando RFC de conteo por ciudad.

### TarjetaCredito y Seguridad

- TarjetaCredito se modela con tarjeta\_id PK surrogado. Se almacena solo numero\_mask y token\_pago para cumplir buenas prácticas de seguridad.
- Relación: Usuario 1..\* TarjetaCredito ⇒ FK en TarjetaCredito usuario\_cedula.

### Revision / Rating

- Revision se modela como entidad con FK a Servicio y FK a Usuario para revisor y revisado.
- Se impone CHECK en rating y se considera UNIQUE(servicio\_id, revisor\_cedula) si se requiere una sola revisión por revisor por servicio.

## **Confirmación BCNF y observaciones finales**

Tras revisar el modelo relacional modificado (versión en Excel) y la transformación desde el UML, todas las relaciones definidas en el modelo cumplen BCNF bajo las siguientes condiciones y observaciones:

- Se usan claves primarias apropiadas (surrogate keys donde conviene) y las dependencias funcionales relevantes están documentadas por relación.
- Los atributos derivados (valor\_servicio, calificacion\_promedio, nivel\_transporte si se infiere) están documentados como derivados y se recomienda no almacenarlos o mantener triggers que los actualicen.
- Reglas de integridad adicionales (checks y triggers como validación de rangos de rating y control de solapamiento de disponibilidades) complementan la normalización para asegurar integridad temporal y de negocio.

Se recomienda incluir en el repositorio: el DDL final, triggers documentados, y el script de pruebas (pruebas\_alpescab.sql) para replicar las verificaciones de integridad y normalización.