





Tarjeta de Crédito				
numero	nombre	fechaVencimiento	codigoSeguridad	cedulaUsuario
PK, NN, ND	NN	NN, CK[hoy...n]	NN, CK[000...999]	FK(UsuariodeServicio.cedula)
123456789	Usuario1	dic-28	918	10293847
987654321	Usuario2	sep-26	485	65748392

Usuario de Servicios	
cedula	correo
PK, FK(Usuario.cedula)	FK(Usuario.correo)
10293847	usuario1@gmail.com
65748392	usuario2@gmail.com

Usuarios Conductor	
cedula	correo
PK, FK(Usuario.cedula)	FK(Usuario.correo)
123456780	conductor1@gmail.com
987654321	conductor2@gmail.com

Relación Usuario Conductor - Vehículo	
usuarioID	vehiculoID
FK(Usuario.cedula), NN	FK(Vehiculo.placa)
95310298	C9821
27495821	C3851

Rating				
revisor	revisado	comentario	rating	idComentario
NN, FK(Usuario.cedula)	NN, FK(Usuario.cedula)	UA	NN, UA, CK[0...5]	PK, NN
129481284	348248859	excelente	5	X39400BnmsO-20
298593283	839412302	decente	4.1	2993XN-34Mppe39

Usuario					
nombre	correo	celular	calificacion	cedula	tipo
NN	NN, UA	NN, UA	NN, CK[0...5]	PK, NN, UA	NN, CK(U.Conductor, U.Servicios)
John Doe	johnD@gmail.com	3094829131	4.5	98420139	CONDUCTOR
Jane Doe	janeD@gmail.com	3751094099	4.7	85410323	CLIENTE

Relación Usuario - Comentario	
idUsuario	idComentario
FK(Usuario.cedula), NN	FK(Rating.idComentario)
95310298	C9821
27495821	C3851

Ciudad	
idCiudad	nombre
PK, NN, SA, ND, NC	NN, NC
1392018TA	Bogota
391910CTG	Cartagena

Punto Geografico				
nombre	direccion	latitud	longitud	ciudad
NN	PK, NN, ND, NC	NN, CK[0...90]	NN, CK[0...180]	NN, FK(Ciudad.idCiudad)
CityU	Cl. 19 #2a - 10	4.6032	74.0672	Bogota
C.C. Gran Estacion	Av. ClI 26 No 62 – 47	46.474	74.1019	Bogota

Relación Usuario - Servicios	
serviciosOfrece	idUsuario
NN	FK(Usuario.cedula)
domicilio	98420139
mercancias	98420139

Vehículo							
tipo	marca	modelo	color	placa	capacidadPasajeros	ciudadExpedicion	registrado
NN	NN	NN	NN	PK, NN, UA, ND	NN	NN	NN
Van	Chevrolet	N400-Cargo	Blanco	FJI938	6	Bogota	True
Motocicleta	Yamaha	Yzf R1 2015	Amarillo	POC301	1	Villavicencio	False

Relación Vehículo - Disponibilidad	
idVehiculo	idDisponibilidad
FK(Vehiculo.placa)	FK(Disponibilidad.idDisponibilidad)
FJI938	D2451
POC301	D4810

Disponibilidad				
idDisponibilidad	día	horaInicio	horaFin	tipoServicio
PK, NN	NN	NN, CK[00:00...23:59]	NN, CK[00:00...23:59]	NN
D2451	Viernes	7:30	15:00	MERCANCIA
D4810	Lunes	9:00	17:15	PASAJERO

Transporte de Mercancias			
tarifaMercancias	PuntoPartida	PuntoLlegada	idServicio
NN, SA	NN, UA, NC	NN, UA, NC	NN, PK, FK(Usuario.idServicio)
10000	Calle 200, 60-00	Ac. 26 #62-47	3069WXTTRO-30d
10000	Cra. 68a #24 B-10	Cra. 11 #82-71	1002RTODP-56d

Entrega comida a domicilio			
tarifaDomicilio	ubicacionRestaurante	puntoUsuario	idServicio
NN, SA	NN, UA, NC	NN, UA, NC	NN, PK, FK(Usuario.idServicio)
5600	Av. americas # 44-23	Cra 1 E #19a-70	8530LAPxsA-45f
5600	Cl. 140 & Ak 9	Cl. 19 #2a-10	2921JCXrwo-22f

Transporte de Pasajeros				
tarifaPasajeros	puntoPartida	puntoLlegada	nivelTransporte	idServicio
NN, SA	NN, UA	NN, UA	NN, SA, CK(Estandar, Confort, Large)	NN, PK, FK(Usuario.idServicio)
3400	Calle 63 # 59A-06	Cra. 68b #24-39	Comfort	2950RiDOabd-05p
3400	Cl. 83	Cra 1 E #19a-70	Estándar	3912cmDMKOi-87p

Servicio						
idServicio	tipo	distancia	valorServicio	horaInicio	horaFinalización	vehiculo
PK, NN	NN	NN	NN	NN, CK[00:00...23:59]	NN, CK[00:00...23:59]	NN, FK(vehiculo.placa)
8530LAPxsA-45f	Entrega comida a domicilio	4km	43000	12:23	1:14	vehiculo
3912cmDMKOi-87p	Transporte de Pasajeros	13km	25000	9:27	10:09	vehiculo

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.