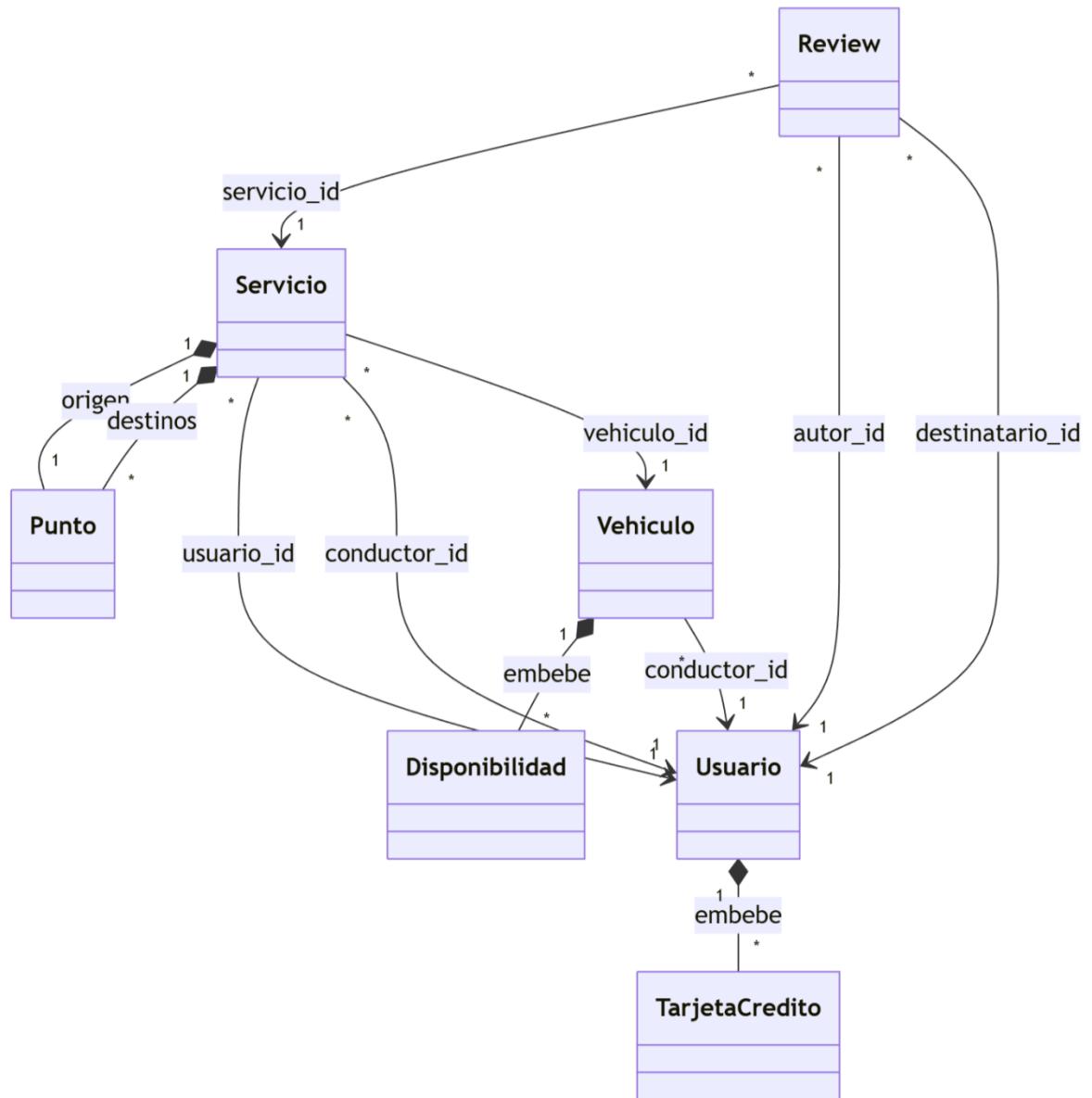


Proyecto 3

Julian Esteban Hernandez Rojas

1.a)



1.b) Entidad Descripción

Usuario Persona que usa la aplicación, tipo servicio o conductor, con información personal y tarjetas embebidas.

Vehículo Vehículo registrado por un conductor. Incluye disponibilidad embebida.

Servicio Viaje o entrega realizada por un conductor. Guarda usuario_id, conductor_id y vehiculo_id por referencia.

Tarjeta Medio de pago del usuario de servicio. Embebida dentro de Usuario.

Punto Coordenadas/dirección de origen y destino. Embebido en Servicio.

Disponibilidad Disponibilidad del vehículo. Embebida en Vehículo.

Relaciones con cardinalidades

Relación	Tipo	Cardinalidad Justificación
Usuario – Servicios	Referencia 1 a N	Un usuario puede tener muchos servicios; el histórico no debe inflar el doc usuario.
Conductor – Vehículos	Referencia 1 a N	Un conductor registra muchos vehículos; no se debe duplicar usuario dentro de cada vehículo.
Vehículo – Servicios	Referencia 1 a N	Histórico de servicios no pertenece al vehículo como documento embebido.
Servicio – Destino(s)	Embebido 1 a N	Destinos pertenecen solo al servicio; no se comparten.

Relación	Tipo	Cardinalidad	Justificación
Servicio – Origen	Embebido	1 a 1	Igual: origen es propio del viaje.
Usuario – Tarjetas	Embebido	1 a N	Parte del usuario, se accede junto al perfil.
Vehículo – Disponibilidad	Embebido	1 a N	Franja horaria es propia del vehículo.

1.c)

Usuario ↔ Servicios

Decisión: REFERENCIADO (Servicio guarda usuario_id)

Justificación:

- ALPES CAB tiene **1.200.000 usuarios activos** → si embebemos servicios dentro del usuario, su documento crecería sin límite.
- Un usuario puede generar cientos de viajes al año → el documento sería demasiado grande.
- Historial de servicios requiere **consultas frecuentes**, filtros por fecha, agrupaciones, cálculos, etc.
- Generación de servicios es **altísima**:
 - Aprox 7 viajes/día × 100.000 conductores × 70% activos ≈ **490.000 servicios nuevos por día**
 - Esto requiere una colección independiente optimizada.
- Servicios se actualizan durante el viaje (1/3), lo cual sería costoso si estuvieran embebidos.

Conclusión:

Los servicios deben ser documentos independientes, enlazados por usuario_id.

2) Conductor ↔ Servicios

Decisión: REFERENCIADO

Justificación:

- Cada conductor realiza entre **6–8 servicios diarios**, durante años → millones de servicios.
- El conductor solo debe guardar su info personal; su historial es demasiado grande para embeberlo.
- Los viajes necesitan consultas frecuentes: “historial de conductor”, “top 20 conductores”, etc.
- Los servicios cambian durante el viaje → actualización frecuente → NO puede estar embebido.

Conclusión:

Relación 1:N con **referencia** conductor_id.

3) Usuario (servicio) ↔ Tarjetas de pago

Decisión: EMBEBIDO

Justificación:

- Los usuarios actualizan el medio de pago **solo 1 vez al año** → muy baja frecuencia.
- Un usuario normalmente tiene **1 o 2 tarjetas** → tamaño pequeño.
- Las tarjetas siempre se consultan junto con el usuario.
- No requieren búsquedas globales ni colecciones aparte.

Conclusión:

Embebido en el documento usuario.

4) Vehículo ↔ Conductor**Decisión: REFERENCIADO (conductor_id)****Justificación:**

- Hay alrededor de 100.000 conductores, cada uno con en promedio **1.1 vehículos**.
- Información del conductor puede cambiar → si se embebe, habría duplicación.
- Vehículos requieren lecturas independientes (filtrado por ciudad, placa, tipo, etc.).

Conclusión:

Vehículo referencia al conductor.

5) Vehículo ↔ Disponibilidad**Decisión: EMBEBIDO****Justificación:**

- La disponibilidad del vehículo se actualiza **mensualmente** → baja frecuencia.
- Las franjas horarias pertenecen exclusivamente al vehículo.
- Siempre se consultan como parte del documento del vehículo.
- No tienen vida independiente.

Conclusión:

Embebido dentro del documento vehiculo.

6) Servicio ↔ Origen / Destinos**Decisión: EMBEBIDO (Punto y lista de Puntos)**

Justificación:

- Es información **histórica** del servicio → debe guardarse congelada.
- No se reutiliza en otros servicios.
- El viaje puede ser modificado (1/3 de los casos según la carga de trabajo).
- Los puntos de origen y destino **solo tienen sentido dentro del viaje.**

Conclusión:

Origen y destinos embebidos.

7) Servicio ↔ Vehículo

Decisión: REFERENCIADO (vehiculo_id)

Justificación:

- Un vehículo participa en muchos servicios (miles a lo largo de su vida).
- Consulta de historial por vehículo es común.
- Embebido haría crecer el documento vehículo de forma ilimitada.

Conclusión:

Usar referencia vehiculo_id.

1.d) Usuario (con tarjetas embebidas)

```
{  
  "_id": "u002",  
  "tipo": "servicio",  
  "nombre": "Paula Pasachova",  
  "correo": "paula@example.com",
```

```
"celular": "3125136383",
"cedula": "100211311",
"tarjetas": [
{
  "numero": "4222221111111111",
  "nombreEnTarjeta": "Paula Pasachova",
  "vencimiento": "2028-05",
  "cvv": "321"
}
]
```

Vehículo (referencia al conductor, disponibilidad embebida)

```
{
  "_id": "v900",
  "conductor_id": "u200",
  "tipo": "carro",
  "marca": "Toyota",
  "modelo": "Corolla 2022",
  "color": "Blanco",
  "placa": "ABC123",
  "ciudad": "Bogotá",
  "capacidad": 4,
```

```
"nivel": "Confort",
"disponibilidad": [
{
  "dia": "lunes",
  "horalnicio": "08:00",
  "horaFin": "12:00",
  "tipoServicio": "transporte_pasajeros"
}
]
}
```

Servicio (referencias + datos embebidos)

```
{
  "_id": "s8000",
  "usuario_id": "u001",
  "conductor_id": "u200",
  "vehiculo_id": "v900",
  "tipoServicio": "transporte_pasajeros",
  "nivel": "Confort",
  "origen": {
    "direccion": "Cra 7 # 72-34",
    "latitud": 4.6532,
    "longitud": -74.059,
```

```
"ciudad": "Bogotá"  
},  
"destino": [  
{  
    "direccion": "Cll 93 # 12-45",  
    "latitud": 4.6767,  
    "longitud": -74.048,  
    "ciudad": "Bogotá"  
}  
,  
"distancia_km": 5.3,  
"costo": 15000,  
"horalnicio": { "$date": "2025-10-10T15:15:00.000Z" },  
"horaFin": { "$date": "2025-10-10T15:30:00.000Z" }  
}  
}
```

2. Se creo los scripts en la parte de Documentos

The screenshot shows the MongoDB Compass interface with the database 'sistransnosql.lphqws4.mongodb.net' selected. Under the 'SistransNoSQL' collection, there are four collections: 'reviews', 'servicios' (which is highlighted in light blue), 'usuarios', and 'vehiculos'. Each collection has a checkbox icon next to it.

- ▼ sistransnosql.lphqws4.mongodb.net
- ▼ SistransNoSQL
 - reviews
 - servicios
 - usuarios
 - vehiculos

3. RF1 Crear usuario servicio

HTTP SistransNoSQL / RF1 Crear usuario servicio

POST http://localhost:8080/usuarios/new/save

Docs Params Authorization Headers (8) **Body** Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL **JSON**

```
1  {
2    "id": "u002",
3    "tipo": "servicio",
4    "nombre": "Paula Pasachova",
5    "correo": "paula@example.com",
6    "celular": "3125136383",
7    "cedula": "100211311",
8
9    "tarjetas": [
10      {
11        "numero": "4222221111111111",
12        "nombreEnTarjeta": "Paula Pasachova",
13        "vencimiento": "asds",
14        "cvv": "321"
15      }
16    ]
17  }
```

Body Cookies Headers (5) Test Results

Raw Preview Visualize

```
1  Usuario creado exitosamente
```

RF2. Crear un usuario conductor

http SistransNoSQL / RF2 Crear usuario conductor

POST http://localhost:8080/usuarios/new/save

Docs Params Authorization Headers (8) Body Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL [JSON](#)

```
1  {
2    "id": "u200",
3    "tipo": "conductor",
4    "nombre": "Juan Hernandez",
5    "correo": "juan@example.com",
6    "celular": "3117798877",
7    "cedula": "11111010",
8    "tarjetas": []
9 }
```

Body Cookies Headers (5) Test Results | ⚙️ 200 OK

[Raw](#) ▾ [Preview](#) [Visualize](#) ▾

```
1 Usuario creado exitosamente
```

RF3. Registrar vehículo para conductor

POST <http://localhost:8080/vehiculos/new/save>

Docs Params Authorization Headers (8) Body Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL [JSON](#)

```
1 {
2   "id": "v800",
3   "conductor_id": "u200",
4
5   "tipo": "camioneta",
6   "marca": "Ford",
7   "modelo": "Raptor 2022",
8   "color": "Negro",
9   "placa": "ABC124",
10  "ciudad": "Bogotá",
11  "capacidad": 4,
12
13  "disponibilidad": [
14    {
15      "dia": "lunes",
16      "horaInicio": "06:00",
17      "horaFin": "07:00",
18      "tipoServicio": "transporte de pasajeros"
19    },
20    {
21      "dia": "lunes",
22      "horaInicio": "12:00",
23      "horaFin": "13:00",
24      "tipoServicio": "transporte de pasajeros"
25    }
]
```

Body Cookies Headers (5) Test Results

[Raw](#) [Preview](#) [Visualize](#)

1 Vehiculo creado exitosamente

200 OK

RF4 Registrar disponibilidad de un usuario conductor

POST <http://localhost:8080/vehiculos/v800/edit/save>

Docs Params Authorization Headers (8) Body Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL [JSON](#)

```
5   "tipo": "prueba",
6   "marca": "Ford",
7   "modelo": "Raptor 2022",
8   "color": "Negro",
9   "placa": "ABC124",
10  "ciudad": "Bogotá",
11  "capacidad": 2,
12
13  "disponibilidad": [
14    {
15      "dia": "lunes",
16      "horaInicio": "08:00",
17      "horaFin": "12:00",
18      "tipoServicio": "transporte_pasajeros"
19    },
20    {
21      "dia": "lunes",
22      "horaInicio": "14:00",
23      "horaFin": "18:00",
24      "tipoServicio": "transporte_pasajeros"
25    }
26  ]
27 }
```

Body Cookies Headers (4) Test Results

[Raw](#) [Preview](#) [Debug with AI](#)

1 Disponibilidad solapada con otro vehiculo (id=v910) para el dia lunes

400 Bad Request

RF5. Modificar disponibilidad

POST <http://localhost:8080/vehiculos/v800/edit/save>

Docs Params Authorization Headers (8) Body Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
5   "tipo": "prueba",
6   "marca": "Ford",
7   "modelo": "Raptor 2022",
8   "color": "Negro",
9   "placa": "ASCI24",
10  "ciudad": "Bogotá",
11  "capacidad": 2,
12
13
14  "disponibilidad": [
15    {
16      "dia": "lunes",
17      "horaInicio": "08:00",
18      "horaFin": "12:00",
19      "tipoServicio": "transporte_pasajeros"
20    },
21    {
22      "dia": "lunes",
23      "horaInicio": "14:00",
24      "horaFin": "18:00",
25      "tipoServicio": "transporte_pasajeros"
26    }
27  ]
28 }
29
```

Body Cookies Headers (4) Test Results

Raw Preview Debug with AI

400 Bad Request

1 Disponibilidad solapada con otro vehículo (id=v910) para el dia lunes

RF6. Solicitar un servicio por parte de usuario

POST <http://localhost:8080/servicios/request>

Docs Params Authorization Headers (8) Body Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL [JSON](#)

```
1 {  
2   "_id": "s6900",  
3   "usuario_id": "u100",  
4   "tipoServicio": "pasajero",  
5   "nivel": "Estándar",  
6   "origen": {  
7     "direccion": "Cra 7 # 72-34",  
8     "lat": 4.6532,  
9     "long": -74.059,  
10    "ciudad": "Bogotá"  
11  },  
12  "destino": [  
13    {  
14      "direccion": "Cl1 93 # 12-45",  
15      "lat": 4.6767,  
16      "log": -74.048,  
17      "ciudad": "Bogotá"  
18    }  
19  ]  
20}
```

Body Cookies Headers (5) Test Results [?](#)

[201 Created](#) 291

{ } JSON [D](#) Preview [Visualize](#)

```
1 {  
2   "id": "692d051022cad49a9002b330",  
3   "usuario_id": "u100",  
4   "conductor_id": "u1000",  
5   "vehiculo_id": "v910",  
6   "tiposervicio": "pasajero",  
7   "nivel": "Estándar",  
8   "origen": {  
9     "direccion": "Cra 7 # 72-34",  
10    "latitud": 4.6532,  
11    "longitud": -74.059,  
12    "ciudad": "Bogotá"  
13  },  
14  "destino": [  
15    {}  
16  ]  
17}
```

A
Ve

Runner

RF7 Finalizar servicio

POST <http://localhost:8080/servicios/692d051022cad49a9002b330/finish>

Docs Params Authorization Headers (7) Body Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL

This request does not have a body

Body Cookies Headers (5) Test Results [?](#)

[200 OK](#)

{ } JSON [D](#) Preview [Visualize](#)

```
1 {  
2   "id": "692d051022cad49a9002b330",  
3   "usuario_id": "u100",  
4   "conductor_id": "u1000",  
5   "vehiculo_id": "v910",  
6   "tiposervicio": "pasajero",  
7   "nivel": "Estándar",  
8   "origen": {  
9     "direccion": "Cra 7 # 72-34",  
10    "latitud": 4.6532,  
11    "longitud": -74.059,  
12    "ciudad": "Bogotá"  
13  },  
14  "destino": [  
15    {}  
16  ]  
17}
```

Ru

RFC1 Usuario servicio histórico

HTTP SistransNoSQL / servicio / RFC1 - Historico de servicio usuario

GET http://localhost:8080/servicios/historico/usuario/u001

Docs Params Authorization Headers (6) Body Scripts Settings

Query Params

Key	Value	Description
Key	Value	Description

ions

Body Cookies Headers (5) Test Results

200 OK

{ } JSON Preview Visualize

```
1 [
2   {
3     "id": "s1000",
4     "usuario_id": "u001",
5     "conductor_id": "v100",
6     "vehiculo_id": "v900",
7     "tiposervicio": "transporte_pasajeros",
8     "nivel": "Confort",
9     "origen": {
10       "direccion": "Cra 7 # 72-34",
11       "latitud": 4.6532,
12       "longitud": -74.059,
13       "ciudad": "Bogotá"
14     },
15     "destino": [

```

Usuario conductor histórico

HTTP SistransNoSQL / servicio / RFC1 - Historico de los servicio usuario

GET http://localhost:8080/servicios/historico/conductor/u100

Docs Params Authorization Headers (6) Body Scripts Settings

Query Params

Key	Value	Description
Key	Value	Description

ions

Body Cookies Headers (5) Test Results

200 OK

{ } JSON Preview Visualize

```
1 [
2   {
3     "id": "s1000",
4     "usuario_id": "u001",
5     "conductor_id": "v100",
6     "vehiculo_id": "v900",
7     "tiposervicio": "transporte_pasajeros",
8     "nivel": "Confort",
9     "origen": {
10       "direccion": "Cra 7 # 72-34",
11       "latitud": 4.6532,
12       "longitud": -74.059,
13       "ciudad": "Bogotá"
14     },
15     "destino": [

```

RFC2 Top 20 mas servicios

SistransNoSQL / RFC2 -Top 20

GET http://localhost:8080/usuarios/conductores/top20

Docs Params Authorization Headers (6) Body Scripts Settings

Query Params

Key	Value	Description
Key	Value	Description

Body Cookies Headers (5) Test Results | 200 OK

{ } JSON ▾ ▶ Preview ▾ Visualize ▾

```
1  [
2    {
3      "totalServicios": 5,
4      "conductor_id": "u200",
5      "nombre": "Juan Hernandez",
6      "correo": "juan@example.com",
7      "celular": "3117798877",
8      "cedula": "11111010"
9    },
10   {
11     "totalServicios": 4,
12     "conductor_id": "u201",
13     "nombre": "Carlos Gomez",
14     "correo": "carlos@example.com",
15     "celular": "3100976543".
```