PRUEBA DE CONOCIMIENTOS
CARGO: ARQUITECTO DE SOFTWARE
FECHA: NOVIEMBRE 11 DE 2021
NOMBRE: ANGIE LIZETH MORA LANCHEROS

DOCUMENTO DE ARQUITECTURA.

1. ATRIBUTOS DE CALIDAD

Los atributos de calidad permiten alcanzar las motivaciones técnicas indicadas por los interesados, basándose en las decisiones tomadas en el diseño arquitectónico de la solución. A continuación, se relacionan los atributos de calidad, su justificación, prioridad e indicadores clave de rendimiento:

Atributo	Justificación	Prioridad	KPI	Indicador
Escalabilidad	La arquitectura garantizará escalabilidad horizontal, teniendo en cuenta que tiene un API de interoperabilidad la cual permite que cualquier tercero pueda consumir los servicios de consulta sin importar la tecnología. Por otra parte, dado que se propone el uso de ECS de Amazon, que permitirá dicha elasticidad basada en nube.	1	Con el fin de mantener un cumplimiento con este atributo de calidad se establece como indicador de rendimiento (KPI) el no impactar los tiempos de respuestas de los servicios cada vez que se aumenten o disminuyan éstos en el core.	Tiempo máximo de respuesta esperado en los servicios: 200 ms
Seguridad	La arquitectura se basará en implementar medidas que contrarresten los fallos de seguridad que se nombran en el top 10 que describe OWASP, adicional, tener en cuenta que se contempla un API de autenticación que permitirá el acceso con token.	2	Con el fin de dar cumplimiento y seguimiento con el atributo de calidad de seguridad en la arquitectura solución, se establece como KPI, el número de ataques resistidos	Porcentaje resistido del 90%

			mediante la seguridad que se establece en el API de autenticación, esto con el fin de dar cumplimiento con la efectividad de los protocolos de seguridad informática.	
Disponibilidad	La arquitectura garantizará la alta disponibilidad, por dos razones principales: su core basado en nube y la implementación de balanceadores y patrón *publisher and subscribe.*	3	Con el fin de mantener un cumplimiento con este atributo de calidad se establece como indicador de rendimiento (KPI) el tiempo máximo para detectar la falla en la disponibilidad del API de afiliación el cual debe ser menor a 52 minutos y 34 segundos durante un año para garantizar una disponibilidad del 99%	Porcentaje de disponibilidad: 99%
Modificabilidad	La arquitectura se plantea con el atributo de ser modificable y bajo nivel de acople, siendo así	4	Con el fin de mantener un cumplimiento con este	Tiempo máximo de respuesta de los servicios: 200 ms

	tener la posibilidad de ajustar y actualizar sin afectar al usuario final, dado que se plantea un dominio de servicios.		atributo de calidad se establece como indicador de rendimiento (KPI) el reducir el tiempo de latencia en los servicios cada vez que se realice un cambio la cual debe ser menor a 200 ms	
Interoperabilidad	La arquitectura garantizará la interoperabilidad entre los distintos servicios, adicional se implementará un API interoperabilidad que permitirá la comunicación entre distintos tipos de webservices (SOAP o APIRest) de los distintos terceros.	5	Con el fin de dar cumplimiento y seguimiento con el atributo de calidad de interoperabilidad en la arquitectura solución, se establece como KPI, el porcentaje de peticiones exitosas, una vez se adiciona una nueva firma o contrato sin conocerse el origen de su API	Porcentaje de peticiones exitosas: 85%
Capacidad de prueba	La arquitectura se garantizará mediante la realización de pruebas E2E de integración, carga y estrés.	6	Con el fin de mantener un cumplimiento con este atributo de calidad, se	Porcentaje mínimo de pruebas exitosas: 80%

			establece como indicador de rendimiento (KPI) que el porcentaje de las pruebas sea el exitoso en los flujos end-to-end de integración sea superior al 80%	

2. ACUERDOS DE NIVEL DE SERVICIOS

a. OBJETIVO

Presentar los Acuerdos de Niveles de Servicio con el fin de dar claridad los procesos, alcance, responsabilidades, tiempos de respuesta y demás acuerdos relacionados con la prestación del servicio sobre la arquitectura propuesta.

b. SEVERIDAD

El nivel de severidad indica el tipo de incidente y el impacto sobre el negocio. Basado en esta información se establece la prioridad de resolución y el plan de escalamiento.

Nivel de severidad	Descripción
Crítica	Caída total no permite la operación
Alta	Caída parcial de la operación
Media	Incidentes sin detener la operación.
Baja	Dudas de la operación de la herramienta y/o nuevas funcionalidades

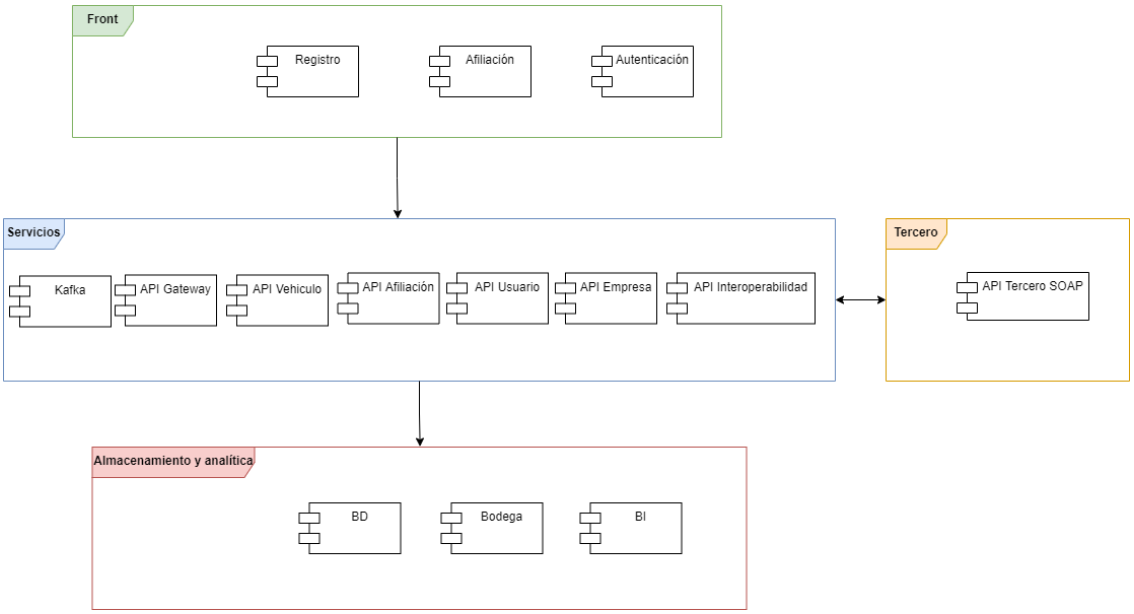
c. NIVEL DE ATENCIÓN Y SOLUCIÓN

A continuación, se presentan los niveles de atención y solución por severidad:

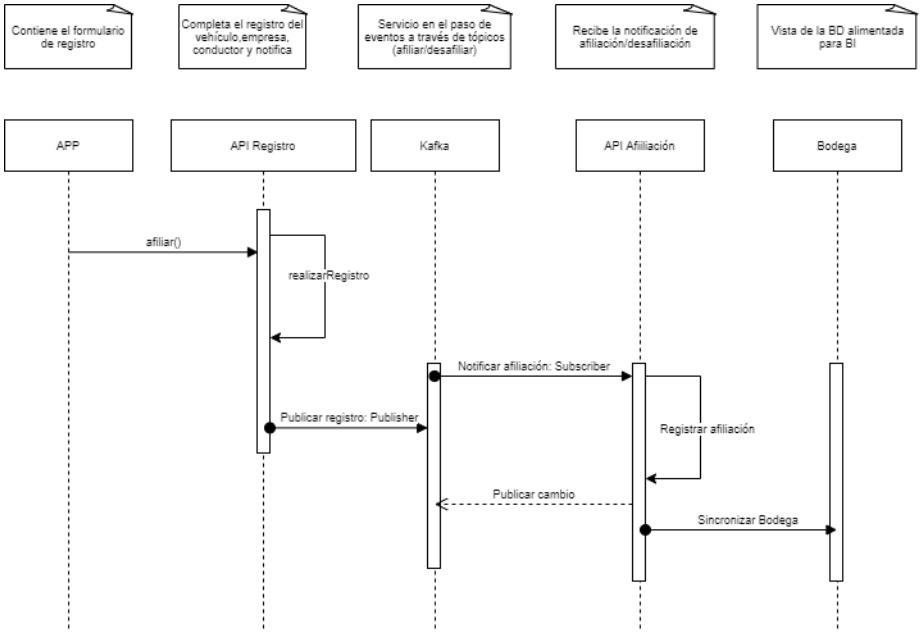
Nivel de severidad	Tiempo máximo de atención	Tiempo máximo de solución
Crítica	20 minutos	5 horas
Alta	35 minutos	7 horas
Media	1 hora	16 horas
Baja	1 día	32 horas

3. ARQUITECTURA SOLUCIÓN (4+1)

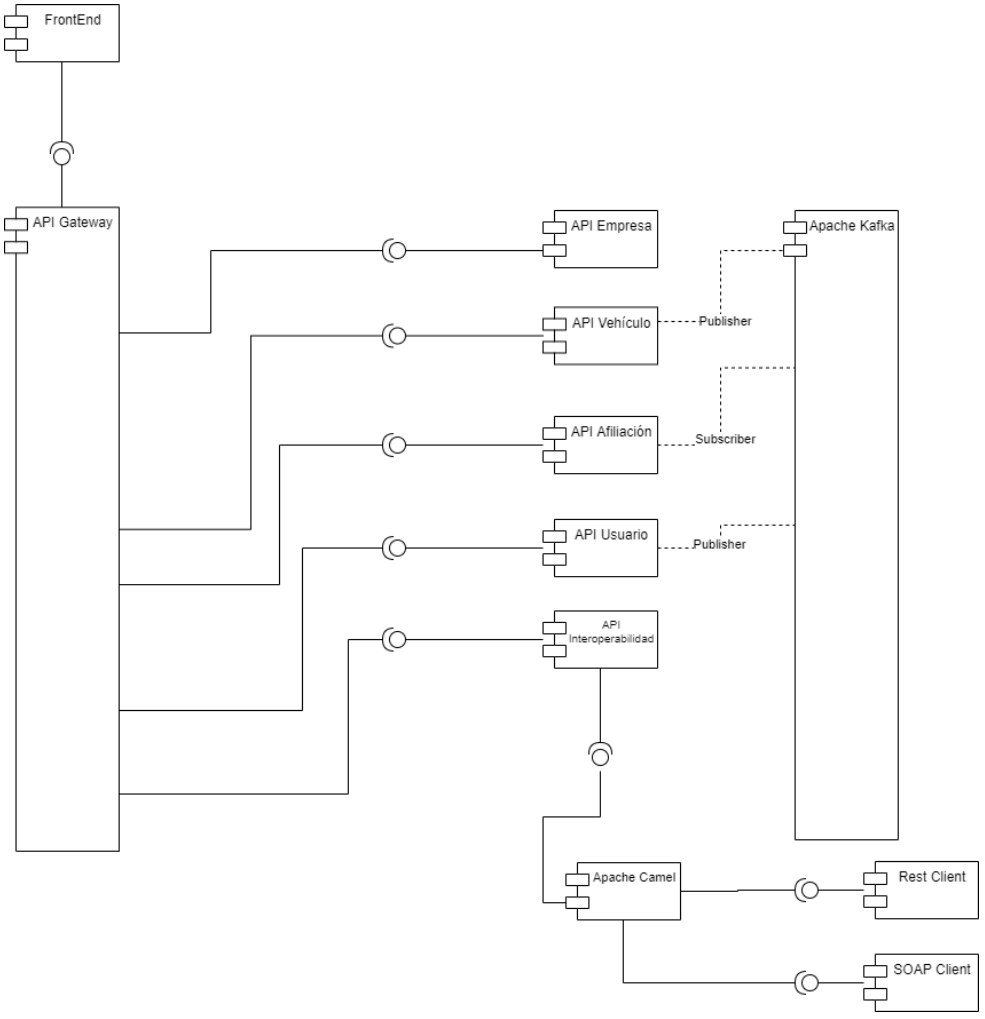
a. PANORAMA



b. VISTA LÓGICA

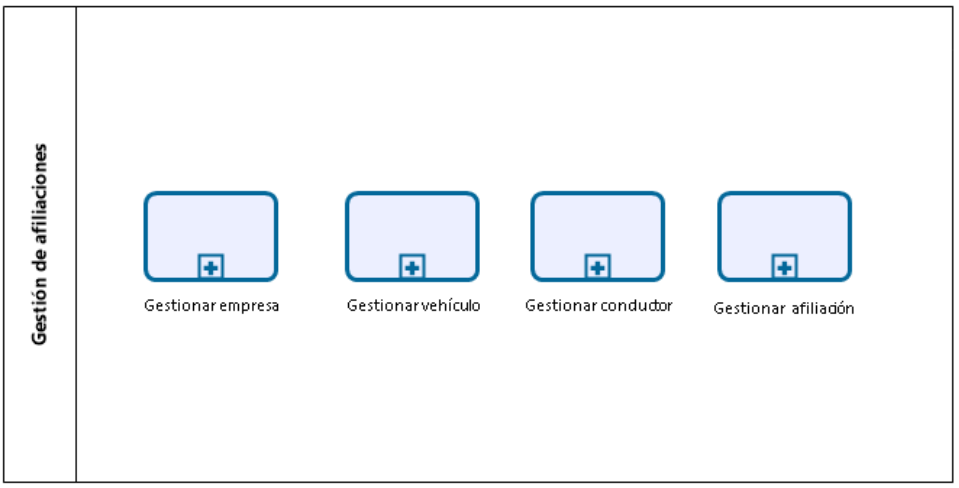


c. VISTA DESARROLLO

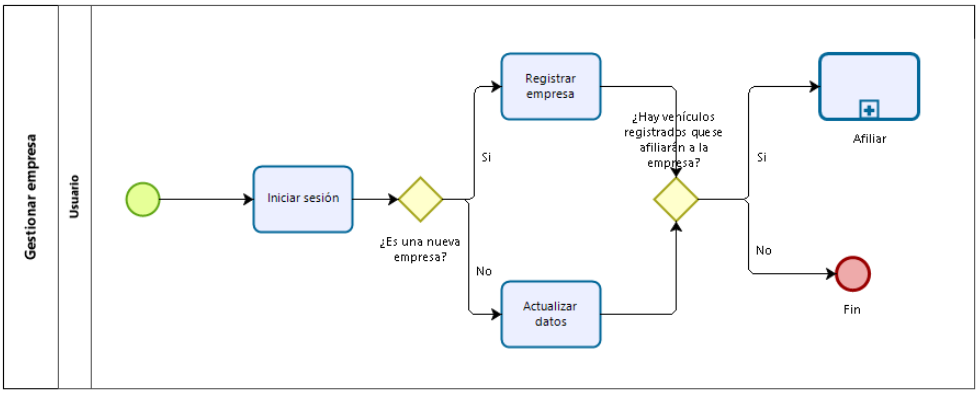


d. VISTA PROCESO

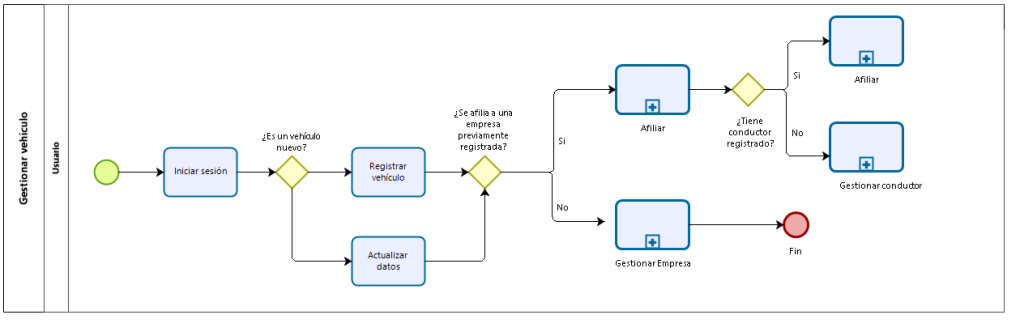
i. Panorama del proceso



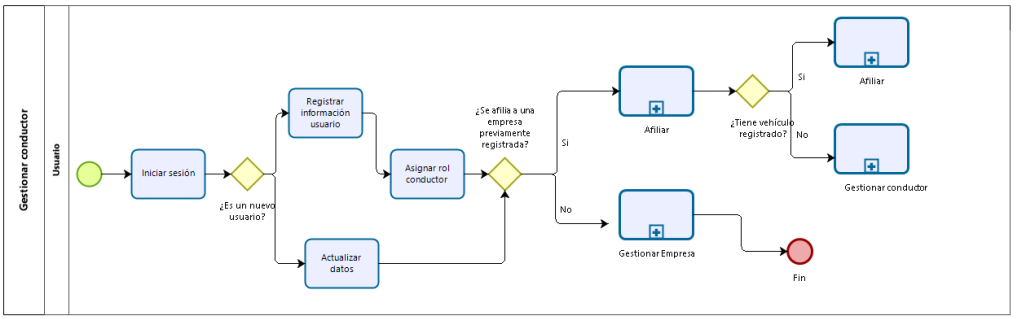
ii. Gestionar empresa



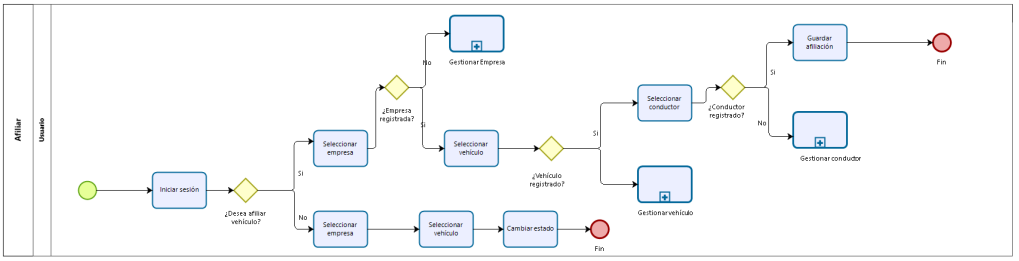
iii. Gestionar vehículo



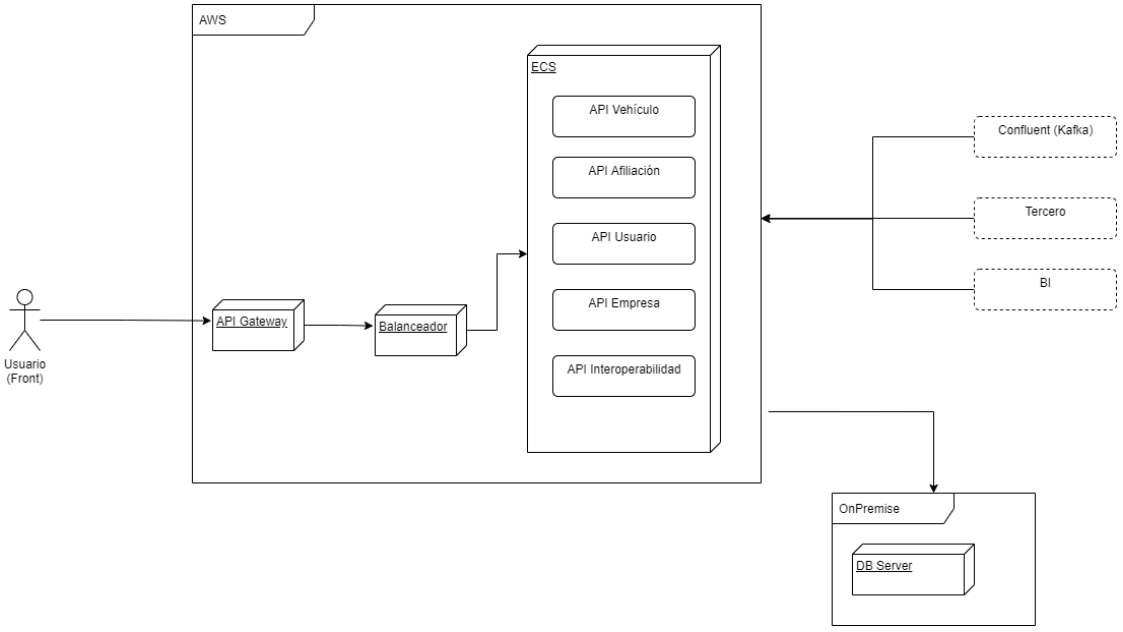
iv. Gestionar conductor



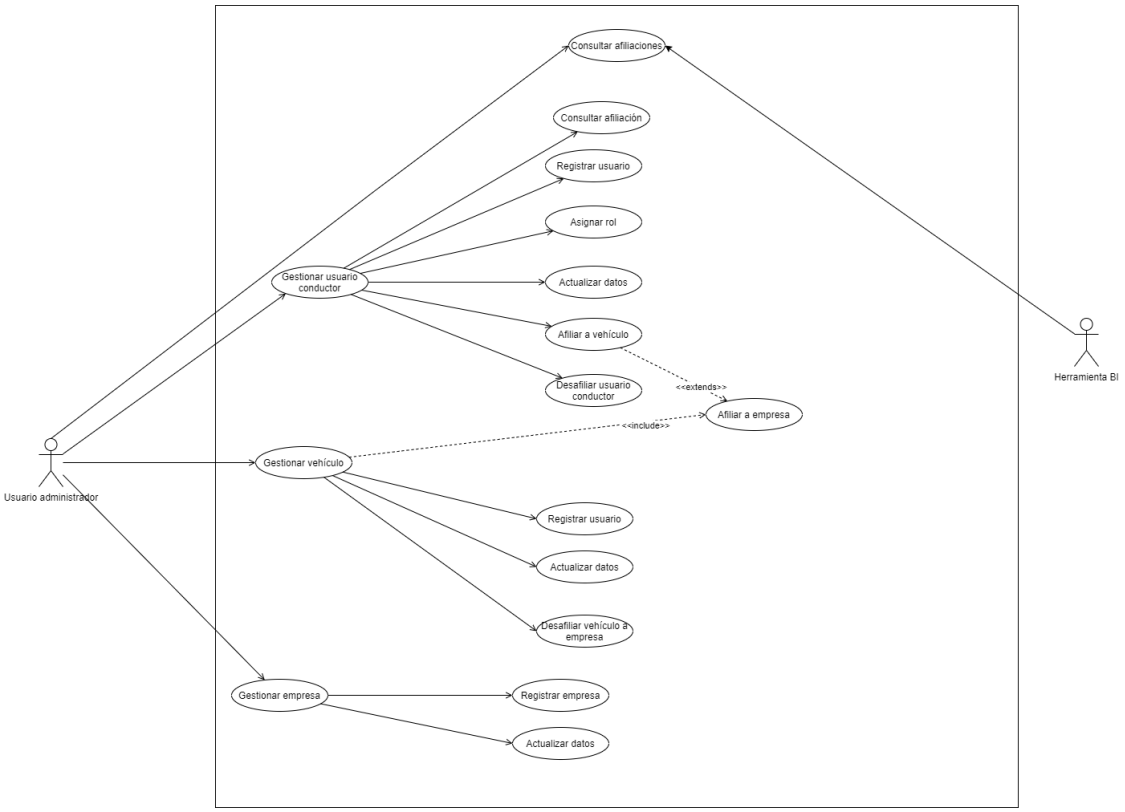
v. Afiliar



e. VISTA FÍSICA

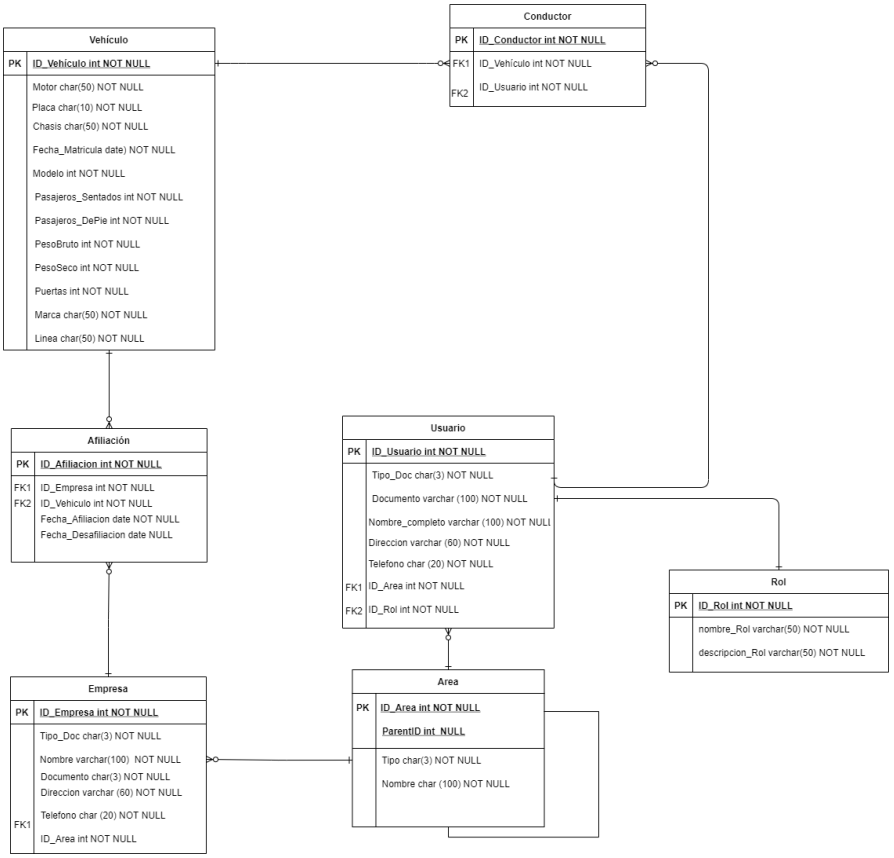


f. VISTA ESCENARIOS



g. OTROS DIAGRAMAS

i. DIAGRAMA ENTIDAD RELACIÓN



4. SUPUESTOS

Los supuestos que se asumen para la propuesta de la arquitectura son:

- La compañía actualmente no cuenta con un sistema de directorio activo, LDAP.
- La compañía cuenta con servicios en nube en Amazon Web Services
- La compañía no cuenta actualmente con un balanceador de carga.
- La compañía no tiene restricciones de presupuesto.

5. TECNOLOGÍAS SUGERIDAS

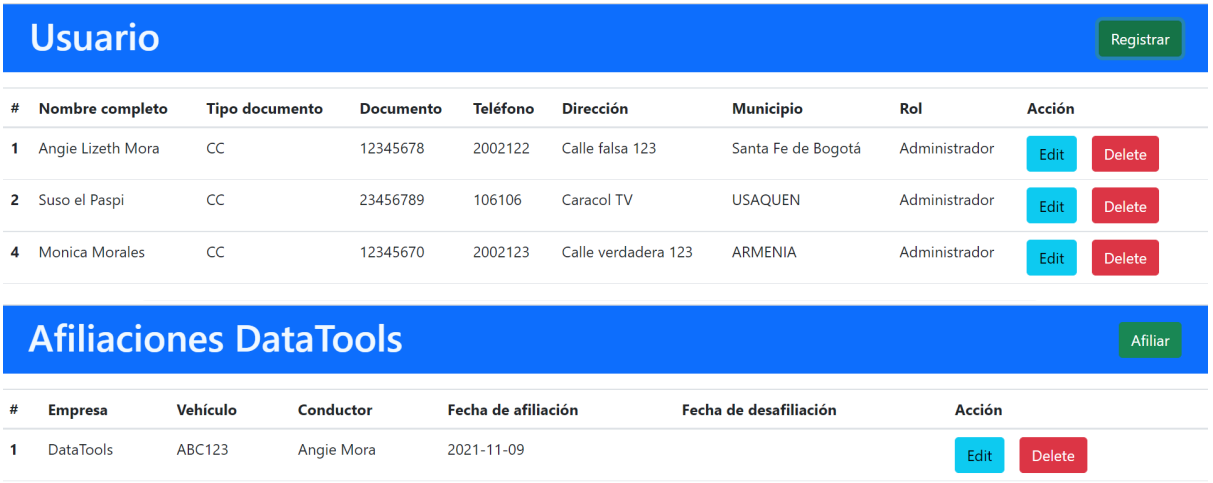
Se sugiere una arquitectura híbrida, en la cual la capa de servicios esté en la nube, así como los componentes tipo balanceador y API Gateway, y para la capa de datos sea *on premise* dado que el costo de licencias puede ser menor que mantener disponible en la nube.

6. REPOSITORIO GIT

https://github.com/almoralit/dataTools

En cuanto al desarrollo del aplicativo, se usó el framework spring boot a nivel de backend y angular a nivel de frontend.

FrontEnd:



Para representar el uso del api gateway, se desplegaron las APIs en swagger:

Api Documentation 1.0

[Base URL: 192.168.0.22:8080/]
http://192.168.0.22:8080/v2/api-docs

Api Documentation

Terms of service

Apache 2.0

afiliacion-controller Afiliacion Controller

GET /afiliacion obtenerAfiliaciones

POST /afiliacion afiliarVehiculo

DELETE /afiliacion/{id} desafiliarVehiculoPorId

empresa-controller Empresa Controller

GET /empresas obtenerEmpresas