

	ACTA DE PROYECTO – CloudFleet	
Versión 1	1/4/2016	

ACTA DE PROYECTO

Nombre / Titulo del Proyecto:

CloudFleet.Com

“Plataforma integral de servicios para la gestión de flotas de transporte y logística”

Sumario Ejecutivo:

Con la aparición y continuo abaratamiento de costos de las tecnologías IoT, los Smartphones, las conexiones móviles a internet, los servicios informáticos en la nube y la revolución de la minería de datos (Big Data). Cada vez más industrias van haciendo su transición al mundo digital con innovaciones que hacen más eficientes y rentables los negocios al tiempo que abaratan costos para los usuarios y mejoran la experiencia.

Cada uno de estos cambios, ha sido disruptivo para esas industrias, generando grandes oportunidades para los nuevos jugadores y grandes desafíos para los tradicionales.

CloudFleet es una empresa que provee un servicio monitoreo de flotas. Funciona con un dispositivo a bordo del vehículo que obtiene la información del viaje y envía al servidor la información en tiempo real.

Hay varias diferencias en el servicio de CloudFleet y otros proveedores de monitoreo. En primer lugar el control es mucho mas detallado ya que se conecta con la interfaz digital de los vehículos OBDI para obtener información mecánica, de consumo y utilización del vehículo y cuenta con un acelerómetro. Pero además ofrece un marketplace de servicios del automóvil como Seguros, Garantías extendidas, Seguridad, servicio técnico y combustibles.

“La gran propuesta de valor de CloudFleet es generar ofertas de servicios superadores para el cliente al tiempo que se maximizan los márgenes de los proveedores mediante el uso de Big Data, logrando el mejor fitness entre lo que el usuario necesita y lo que el proveedor brinda.”

También permitirá a los clientes, tener un control detallado de las erogaciones por vehículo, conductor y zona. Entendiendo por ejemplo con gran detalle quienes son los choferes que incurren en mayores riesgos y por lo tanto en mayores costos de los servicios.

ACTA DE PROYECTO – CloudFleet		
Versión 1	1/4/2016	

Objetivos del Proyecto

- Brindar un servicio de monitoreo de transporte superior a un costo menor.
- Entender mediante la explotación de los datos, cual es la mejor propuesta de servicios para cada cliente.
- Ser una plataforma integral para la gestión de las flotas: Monitoreo de vehículos y choferes, optimización de los costos y su gestión.
- Reducir dramáticamente el fraude a los proveedores de servicios.
- Generar una disruptión en el mercado de servicios para flotas.

Entregables del Proyecto

- Tesis del trabajo impresa.
- Repositorio github del proyecto con historial de cambios y documentación detallada.

ACTA DE PROYECTO – CloudFleet

Versión 1

1/4/2016

Organización del Proyecto y Gerente de Proyecto Designado

La empresa estará organizada de la siguiente manera:



Los Miembros de la **Junta Directiva**, Mg. Gastón Addatti y Mg. Mario Moreno serán los veedores del funcionamiento de la empresa y supervisarán las decisiones del Presidente Fundador. Harán sugerencias de alto nivel respecto al avance del negocio y las grandes decisiones sobre el rumbo del negocio.

Representan a los socios accionistas y tienen el poder de hacer un llamado a asamblea para remover al Presidente en funciones en caso de considerarlo necesario.

El **Presidente Fundador**, el Sr Darío Simonassi, tiene como objetivo el lanzamiento exitoso del negocio en función al plan aquí descripto. Para eso cuenta con un equipo de tres ejecutivos de distintas especialidades, a los cuales deberá guiar para lograr los objetivos y construir los equipos necesarios para tal fin.

El **Director de Tecnología**, el Sr Ramiro Colognini tiene como responsabilidad principal la coordinación general y definición del producto así como su construcción e implementación. Su primer objetivo será elaborar las definiciones, establecer vínculos con proveedores de hardware y software y elegir con quienes se trabajará.

El **Director de Finanzas**, el Sr Pedro Ramirez, tendrá a su cargo la construcción de las políticas de riesgo, la optimización y planificación impositiva y financiera, armar la estructura de cobros y de pagos, la gestión de recursos humanos y la relación con los socios. Su primer objetivo será armar un equipo de actuarios, un equipo de rrhh, un equipo contable y la búsqueda de más socios capitalistas.

El **Director de Operaciones**, el Sr Mariano Tolda, tendrá a su cargo el crecimiento del negocio. En primer lugar, cumplir con un agresivo plan de colocación de dispositivos para conseguir una masa crítica de clientes, lo suficientemente atractiva para seducir a los proveedores de servicios a involucrarse y hacer las integraciones necesarias para ofrecer productos diferenciales a los clientes de la plataforma.

ACTA DE PROYECTO – CloudFleet

Versión 1

1/4/2016

Entendimiento del alcance del Proyecto

Introducción:

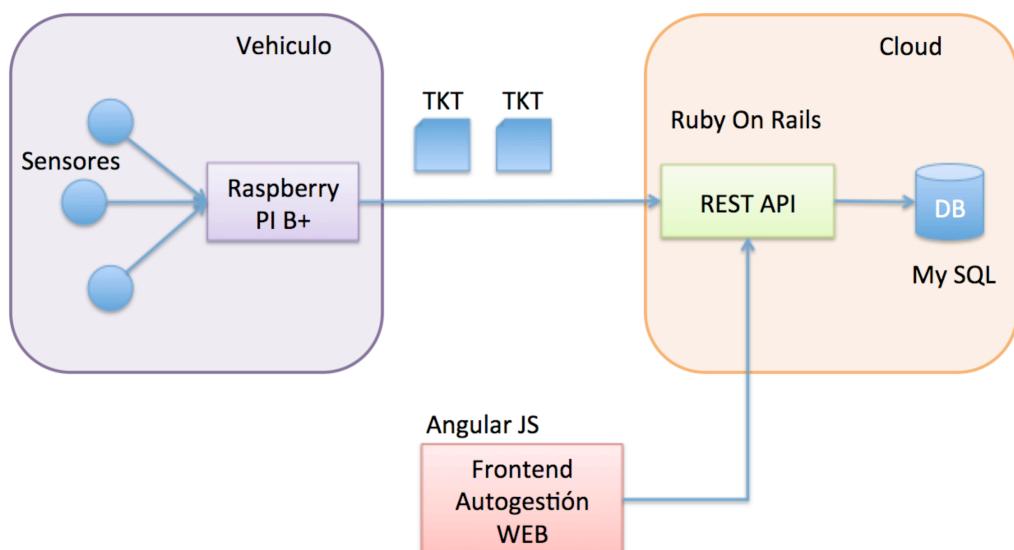
Cloud Fleet es una empresa que brindará plataforma tecnológica de soporte necesario para que las empresas propietarias de flotas de vehículos puedan ejercer el control de sus unidades y sus choferes, así como proveer una herramienta que centralice el control de los costos, los servicios y demás aspectos de la gestión de la flota.

El objetivo principal de CloudFleet es obtener información en tiempo real a gran escala sobre el comportamiento de las flotas para servir como ente de mediación y optimización entre los dueños de dichos vehículos y los proveedores de servicios para los mismos mediante el uso de tecnologías de BigData.

La información que se registrará en tiempo real es:

- Tiempo de uso del vehículo en cuestión.
- Lugares por los que se conduce.
- Velocidades altas, aceleraciones, y frenadas bruscas.
- Condiciones mecánicas del vehículo.

El producto consta de componentes tanto de software como de hardware. El dispositivo de abordo es una pieza de hardware que opera en cada vehículo asegurado junto con un software corriendo en forma local que le permite tasar la prima de riesgo en periodos cortos de 1 minuto. Este envía la información a la central de forma periódica para su almacenamiento verificación y en caso de que haya servicios contratados, el cobro de los mismos.



	ACTA DE PROYECTO – CloudFleet	
Versión 1	1/4/2016	

Estrategia comercial:

CloudFleet es una plataforma que integra varios servicios, por un lado están las empresas que gestionan sus flotas y por el otro las empresas que venden sus servicios para flotas de vehículos. Ambos son nuestros clientes, y el propósito de nuestra empresa es brindar una experiencia comercial superior a ambos.

GRÁFICO DE ACTORES DE LA PLATAFORMA

Clientes del servicio de control de flotas:

- **Remiserías y Mandatarias de Taxis:** Grandes y medianas empresas de remises pueden beneficiarse mucho del servicio, no solo por las posibilidades que ofrece en términos de seguridad y planificación sino también en la eficiencia: la posibilidad de entender la rentabilidad de un horario, zona de cobertura o conductor por ejemplo.
- **Flotas de transporte y logística:** La posibilidad de contar con servicios integrales de seguridad, seguro para cargas, seguro vehicular y laboral, servicio mecánico y garantía extendida así como un control de gestión centralizado y con la posibilidad de hacer integraciones por software con los sistemas ERP de las empresas.
- **Gobierno:** La posibilidad de controlar subsidios, y cumplimiento de servicios. Por ejemplo: subsidios a empresas de omnibus podrían aplicarse a través de la plataforma. También se podría ejercer control sobre entidades de control de tránsito, remolque de vehículos, unidades policiales y de otras dependencias.
- **Flotas de omnibus y líneas de colectivos:** Aplicación para subsidios, justificación de quebrantos, control de choferes, eficiencia en los costos, estimación certera de duración de recorridos.
- **Usuarios domésticos:** Finalmente existe una oportunidad muy grande en términos de usuarios domésticos, sobre todo trabajando con las automotrices y con los bancos para brindar un servicio integral de movilidad que incluya mediante un pago único todos los servicios y garantías:
 - Seguros.
 - Garantía.
 - Auxilio mecánico.
 - Amortización del activo (leasing).
 - Patentes.
 - Venta y reposición.

Clientes como canal de venta de servicios:

- **Compañías de seguros:** Las compañías de seguros podrán ofrecer seguros altamente competitivos con rentabilidades muy superiores. Considerando para el costo de la prima dos aspectos fundamentales:
 - **El comportamiento del usuario:** zonas, cantidad de uso, forma de conducir.
 - **La dificultad del fraude:** El fraude en una situación de observación tan profunda es virtualmente imposible.
 - **La posibilidad de ofrecer paquetes multiproducto:** Robo, accidentes, laborales, responsabilidad civil, pasajeros, cargas, garantía mecánica extendida.
- **Compañías de auxilio mecánico:** Las compañías de servicio mecánico podrán ofrecer un servicio diseñado a la medida del beneficiario y hacer una mejor gestión de sus propias flotas

	ACTA DE PROYECTO – CloudFleet	
Versión 1	1/4/2016	

para beneficio de sus clientes mediante el pleno conocimiento de los mismos. También podrán gozar de un feedback ordenado de sus servicios.

- **Empresas de seguridad privada:** Ademas de poder controlar sus propias flotas, se podrá facturar los servicios de escolta de forma automática y alertar ante la falta de servicio. Ademas de solicitar apoyo o brindar servicios de guardia pasiva o puntos de encuentro.
- **Combustibles:** Las tarjetas de control de combustibles e insumos para flotas (Por ejemplo: YPF en Ruta) podrán integrar sus informes al sistema de CloudFleet que controlará la consistencia con los recorridos y consumos de las unidades y centralizará la cobranza del servicio.

Ejecución:

La estrategia comercial de CloudFleet tendrá tres grandes etapas:

GRAFICO DE ETAPAS

- **Líderes en monitoreo de flotas:** Queremos masificar el producto de trackeo e instalarlo como líder del mercado. Para el cumplimiento de esta primera etapa, consideramos estratégicos los siguientes aspectos:
 - Construcción de un producto superior.
 - Precios agresivos.
 - Gran equipo comercial.
 - Ausencia de ganancia o perdida.
 - Construir producto a partir de la información.
- **Proveer servicios directos a través de partners:** Con una gran cartera de clientes, podremos trabajar en la integración de algunos servicios altamente acoplados con partners estratégicos que nos permitan demostrar la oferta de valor. Ofrecer servicios con precios altamente competitivos mediante el scoring en real time y el manejo de carteras de gran volumen sumado a los beneficios de una gestión unificada de costos, empleados y vehículos. Para esta etapa es clave:
 - Construcción de integraciones.
 - Elección de partners en servicios claves (Seguros, Servicios mecánicos, Combustibles)
 - Gran cartera de clientes para ofrecer.
 - Pequeñas o nulas ganancias.
 - Aprender de la información.
- **Marketplace de servicios líder:** Una vez que el servicio gane tracción, es importante dejar de ser el único proveedor de servicios. El negocio más eficiente y rentable es la intermediación y no generación de estos servicios.
 - Armar un marketplace API integrable y escalable.
 - Explotar los datos.
 - Construir un producto de primer nivel para proveedores de servicios.
 - Competir con:
 - Gran cartera de clientes
 - Plataforma sólida
 - Disponibilidad de información
 - Conocimiento del cliente

	ACTA DE PROYECTO – CloudFleet	
Versión 1	1/4/2016	

Desarrollo del sistema informático:

Para iniciar la operación de la compañía es preciso contar con la plataforma funcionando con solidez en lo que respecta a los sistemas de trackeo dado que la estrategia es comenzar a comercializar los servicios de monitoreo de forma inmediata. Para lograr un impacto mayor, se ha decidido invertir en la construcción del mismo usando un proveedor externo que dispone de los recursos con la formación suficiente para iniciar el desarrollo de inmediato. Es necesario avanzar mientras armamos nuestro propio equipo de tecnología.

Hemos decidido utilizar Ruby on Rails como lenguaje y framework respectivamente para poder avanzar rápidamente en funcionalidades para la plataforma Web.

En lo que respecta a la interfaz gráfica, se utilizará Bootstrap para lograr un resultado reactivo a todos los dispositivos de diferentes tamaños. Para el soporte cartográfico se utilizará Leaflet.js y los mapas de OpenStreetMap. Para charts de reporte se utilizará la librería Charts.js.

El equipo de trabajo constará de dos perfiles:

- **Senior Full Stack Engineer:** 5 Años de experiencia en desarrollo de aplicaciones móviles de alta complejidad.
 - Excluyente:
 - Ruby On Rails.
 - Javascript, DOM y AJAX.
 - HTML 5.
 - MySQL.
 - Amazon Web Services.
 - Deseable
 - Conocimientos de cartografía: Leaflet.js, OpenStreetMap, Google Maps.
 - Conocimientos de HTML reactivo: Bootstrap.
- **Project Manager:** Gerente de proyectos con mas de 10 años de experiencia total, y al menos 3 años en la coordinación de proyectos.
 - Excluyente:
 - Comunicación.
 - Liderazgo.
 - Gran capacidad de análisis y aprendizaje.
 - Disponibilidad para visitas de campo (clientes).
 - Inglés.
 - Deseable:
 - Experiencia en el rubro.
 - Dominio avanzado de Inglés.

	ACTA DE PROYECTO – CloudFleet	
Versión 1	1/4/2016	

	Costo x hora
Senior Full Stack Engineer	ARS 750.00
Project Manager	ARS 1,050.00

Tarea	Horas
Desarrollo y población de las estructuras de datos básicas	
Países, provincias y ciudades	
Utilización de la API de mercadolibre para obtener la información y bajarla a JSON	16
Seed inicial de la información a las estructuras de datos	16
Marcas, modelos, versiones, precios	
Utilización de la API de mercadolibre en grafo a JSON	32
Seed inicial de información	24
Construcción de entidades principales	
Home Page, Menues y estilos	24
Usuarios y autenticación	
Alta de usuario	8
Login, sesiones y logout	16
Vehículos	
Alta de vehículo	16
AJAX p gestión de modelos	8
Modificación de vehículos	8
Listado y gestión de vehículos	12
Conductores	
Alta de conductores	8
Modificación de conductores	8
Listado y gestión de conductores	12
Construcción de motor de tracking	
Dispositivos de trackeo tipos y unidades	16
Trackeos de dispositivo y ubicaciones	
Construcción de entidades	16
Construcción de API	16
Simulación de conducción de vehículos	
Utilización de API de Google Maps y ubicaciones azarosas	32
Motor de física (Aceleraciones, velocidades y trayectorias)	32
Mockeo de compañía de Taxis, Unidades y Nombres	24
Mockeo de trayectos en el conurbano y ciudad de buenos aires	32
Dashboard de monitoreo de vehículos en tiempo real	
Integración de leaflet js open street map	32
Agregado de componentes dinámicos y layers	24
Actualización en background x pooling AJAX	16
Construcción del motor de alertas	
Creación de la entidad y creación de API de generación de alertas	24
Mocks de alertas para la flota de taxis	16
Integración de las alertas al dashboard de monitoreo de vehículos en tiempo real	16
Integración del refresco AJAX	8
Vínculo entre las alertas y los layers del mapa	4
Construcción de motor de reporting	
Estructura del reporte general + links y gestión de las fechas	8
Integración de estilos y librerías de Charts.js	
Reporte de velocidad, aceleración y distancia del conductor	24
Reporte de utilización y recorrido por vehículo	16
Total Hs Desarrollo	596
Total HS Gerenciamiento	119
Costo total del proyecto	ARS 572,160.00

(*)El desarrollo aqui cotizado y detallado es el que se entrega como parte del presente trabajo.

	ACTA DE PROYECTO – CloudFleet	
Versión 1	1/4/2016	

Especificación funcional y metodológica

Tanto el gerente del proyecto como los desarrolladores trabajarán en oficinas del cliente (CloudFleet) para tener una mejor integración entre las ideas del producto y el desarrollo de software.

Si bien el marco general del proyecto tiene una organización en cascada tradicional y un plan muy detallado de ejecución, en las implementaciones particulares hay gran variedad de incertidumbres que todavía no se han bajado a tierra. Por este motivo es preciso que el equipo de desarrolladores senior trabaje muy cerca del equipo de producto, resolviendo estas incertidumbres e innovando en equipo.

Además hay incertidumbre en términos tecnológicos sobre todo en los siguientes módulos

- Simulación de recorridos
- Implementación de cartografía
- Descarga de inventarios de vehículos

Es muy importante establecer una metodología flexible, se trabajará con las siguientes premisas:

- Se aplicará una metodología de desarrollo ágil: Scrum.
- Se aceptarán pequeños fracasos y se entenderán como aprendizajes del equipo.
- Se realizarán reuniones diarias (Standups Meetings) para el seguimiento.
- Cuando alguien del equipo se bloquee con alguna funcionalidad específica, otro lo asistirá con la metodología Pair Programming. Y si el problema persiste se organizará una reunión de brainstorming con todo el equipo para poder avanzar.
- Se trabajará en sprints de 2 semanas, para los cuales se planificará tener algún resultado funcionando, para luego evolucionar a partir de ahí.
- Al comienzo de cada Sprint hará una reunión de Planning con el equipo de producto, el Project Manager y los Desarrolladores para elegir las tareas que serán realizadas durante el próximo sprint.
- Las tareas que no sean elegidas para el Sprint quedarán registradas en la lista de Backlog con tareas que servirán de fuente para la próxima planning.
- Al cierre de cada Sprint el equipo hará una reunión de cierre para el análisis de los resultados y otra de retrospectiva para analizar los errores cometidos en el período y posibles formas de remediarlos.
- El equipo trabajará muy enfocado en el Sprint, cualquier requerimiento nuevo o idea, quedará asentado en el backlog.
- En caso de urgencia, o necesidad de cambiar el rumbo drásticamente, el Project Manager será la única persona habilitada para hacerlo.

Como entrada inicial, los equipos de producto han trabajado en esta especificación de casos de uso para tener en el backlog. A su mejor entender, al final de los sprints son las funcionalidades que si o si deberían estar implementadas.

Historias de casos de uso en el backlog (Dia 1): (Use case stories)

01: Registrar usuario.

	ACTA DE PROYECTO – CloudFleet	
Versión 1	1/4/2016	

Propósito: Un cliente nuevo que entra a la plataforma debe poder registrarse para poder usar los servicios.

Precondición:

- El usuario no debe estar logueado.

Campos de entrada:

- Nombre
- Apellido
- E-Mail
- Nickname
- Password
- Confirmación de password

Éxito: Usuario registrado en la base de datos.

Flujos alternativos:

- **01A:** Email invalido
- **01B:** Email repetido
- **01C:** Nickname invalido
- **01D:** Nickname repetido
- **01E:** Password invalida
- **01F:** Password no es igual
- **01G:** Nombre blanco
- **01H:** Apellido blanco

TODO MOCK REGISTRACION

02: Login

Propósito: Un usuario registrado debe poder ingresar al sitio para operar y ningún otro sujeto debería poder ingresar en su nombre y realizar operaciones.

Precondición: El usuario debe estar deslogueado, haberse registrado y recordar su usuario y contraseña.

Éxito: usuario logueado correctamente.

Campos de entrada:

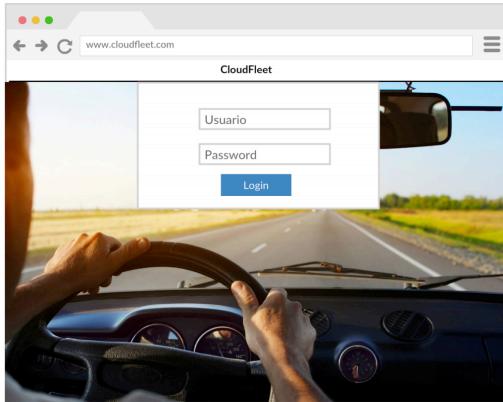
- Usuario
- Clave

Flujos alternativos:

- **01A:** Usuario o password invalido

	ACTA DE PROYECTO – CloudFleet	
Versión 1	1/4/2016	

Mock:



03: Logout

Propósito: El usuario debe poder desconectarse e impedir que alguien actue en su nombre una vez que abandona la computadora.

Precondiciones:

- El usuario debe estar registrado y logueado.

Éxito: usuario deslogueado.

Campos: no hay campos.

No hay flujos alternativos

TODO MOCK LOGOUT

04: Alta de Vehículo

Propósito: El usuario debe poder registrar un vehículo para poder monitorearlo y recibir sus alertas.

Precondiciones:

- El usuario esta registrado.
- El usuario esta logueado.
- El usuario adquirió e instaló un dispositivo de trackeo.
- El usuario cuenta con el número de serie que figura la caja del trackeador.

Campos:

- **Nombre del vehículo:** Identificador interno del vehículo (EJ: Taxi 112)
- **Marca:** Marca del vehículo, se elige de un combo de marcas. (Ej: Volkswagen)
- **Modelo:** Modelo del vehículo, se elige de un combo de modelos. (Ej: Voyage)

	ACTA DE PROYECTO – CloudFleet	
Versión 1	1/4/2016	

- **Versión:** Versión, se elige del combo (Ej: Trendline 1.6 MT)
- **Año:** Año de fabricación del vehículo, se elige de combo y contiene el precio (ej: 2014 - \$190.000)
- **Código del dispositivo de trackeo:** El número de serie del dispositivo de trackeo adquirido.(Ej: AAAA1238).
- **Patente:** Patente del vehículo (Ej: MST 123)
- **Número de chasis:** Número de chasis del vehículo.
- **Número de motor:** Número de motor del vehículo.

Éxito: El vehículo fue registrado

Flujos alternativos:

- **04A:** Nombre de vehículo repetido para el usuario
- **04B:** Código de dispositivo de trackeo inexistente.
- **04C:** Código de dispositivo de trackeo ya esta en uso.

TODO MOCK ALTA DE VEHICULO

TODO ACTIVITY DIAGRAM Visualización AJAX de Modelos, Versiones y Precios

05: Lista de vehículos

Propósito: El usuario debe poder listar todos sus vehículos y buscar en ellos para ver la información de los mismos o bien para poder editarlos o eliminarlos.

Precondiciones:

- El usuario está registrado
- El usuario esta logueado
- El usuario tiene vehículo/s registrado/s

Campos de entrada:

- Busqueda: Busqueda de texto client side para filtrar vehículos.
- Ordenamientos: Ordenamiento por cualquiera de las columnas.

Éxito: se listan todos los vehículos paginados con posibilidad de búsqueda y ordenamiento. Las columnas a listar serán:

- Marca
- Modelo
- Versión
- Año
- Precio
- Patente
- # Chasis
- # Motor
- Dispositivo de trackeo

	ACTA DE PROYECTO – CloudFleet	
Versión 1	1/4/2016	

- Acción: Eliminar
- Acción: Modificar

No hay flujos alternativos

Mock:

TODO MOCK LISTA VEHICULOS

06: Modificación de vehículo

Proposito: El usuario debe poder modificar un vehículo ya cargado en el sistema, para remediar un error u omisión o bien reutilizar un dispositivo de trackeo.

Precondiciones:

- El usuario está registrado
- El usuario está logueado
- El usuario tiene vehículo/s registrado/s
- El usuario identificó un vehículo para modificar

Campos de entrada:

- **Nombre del vehículo:** Identificador interno del vehículo (Ej: Taxi 112)
- **Marca:** Marca del vehículo, se elige de un combo de marcas. (Ej: Volkswagen)
- **Modelo:** Modelo del vehículo, se elige de un combo de modelos. (Ej: Voyage)
- **Versión:** Versión, se elige del combo (Ej: Trendline 1.6 MT)
- **Año:** Año de fabricación del vehículo, se elige de combo y contiene el precio (ej: 2014 - \$190.000)
- **Código del dispositivo de trackeo:** El número de serie del dispositivo de trackeo adquirido.(Ej: AAAA1238).
- **Patente:** Patente del vehículo (Ej: MST 123)
- **Número de chasis:** Número de chasis del vehículo.
- **Número de motor:** Número de motor del vehículo.

Éxito: El vehículo fue modificado

Flujos alternativos:

- **06A:** Nombre de vehículo repetido para el usuario
- **06B:** Código de dispositivo de trackeo inexistente.
- **06C:** Código de dispositivo de trackeo ya está en uso.

Mock:

TODO MOCK MODIFICACION DE VEHICULO

	ACTA DE PROYECTO – CloudFleet	
Versión 1	1/4/2016	

07: Baja de vehículo:

Propósito: El usuario debe poder eliminar un vehículo en caso de error, venta u otros.

Precondiciones:

- El usuario esta registrado
- El usuario esta logueado
- El usuario tiene vehículo/s registrado/s
- El usuario identificó un vehículo para eliminar

Éxito: El vehículo fue eliminado

No hay flujos alternativos.

08: Alta de Conductor

Propósito: Para sacar provecho a la herramienta y sus posibilidades de reporte. El usuario debe registrar sus conductores en la plataforma.

Precondiciones:

- El usuario esta registrado
- El usuario esta logueado

Campos:

- **Nombre:** Nombre del conductor que se desea registrar (Ej: Roberto Fernandez)
- **Identificador interno:** Un código de identificación interna (Ej: Legajo/DNI)
- **Passphrase:** Una clave para el conductor (Ej: fangio123). Esta clave deberá ser utilizada para abordar al vehículo e iniciar el sistema de trackeo. La ausencia de la misma generará una alerta.

Éxito: El conductor es registrado con éxito en la cuenta del usuario.

Flujos alternativos:

- **08A:** El nombre esta repetido para este usuario.
- **08B:** El identificador interno esta repetido para este usuario.
- **08C:** Alguno de los campos es nulo o vacío.

Mock:

TODO MOCK PANTALLA ALTA CONDUCTOR

09: Lista de conductores:

Propósito: El usuario debe poder visualizar y buscar en su nómina de conductores, así como contar con un punto de acceso para eliminar o modificar la información de alguno de ellos.

	ACTA DE PROYECTO – CloudFleet	
Versión 1	1/4/2016	

Precondiciones:

- El usuario esta registrado
- El usuario esta logueado
- El usuario tiene conductor/es registrado/s

Éxito: Se listan los conductores del usuario logueado con opción de buscar y ordenar clientside. Las columnas que se muestran son:

- Nombre
- Identificador interno
- Passphrase
- Acciones (Eliminar, Editar, Reporte)

No hay flujos alternativos.

Mock:

TODO MOCK LISTAR CONDUCTORES

10: Modificar Conductores:

Propósito: El usuario debe poder modificar la información de sus conductores registrados. Sea porque un conductor ya no cumple sus funciones, o bien cambiaron los identificadores internos o las claves de acceso.

Precondiciones:

- El usuario esta registrado
- El usuario esta logueado
- El usuario tiene conductor/es registrado/s

Campos:

- **Nombre:** Nombre del conductor que se desea registrar (Ej: Roberto Fernandez)
- **Identificador interno:** Un código de identificación interna (Ej: Legajo/DNI)
- **Passphrase:** Una clave para el conductor (Ej: fangio123). Esta clave deberá ser utilizada para abordar al vehículo e iniciar el sistema de trackeo. La ausencia de la misma generará una alerta.

Éxito: El conductor es modificado con éxito en la cuenta del usuario.

Flujos alternativos:

- **10A:** El nuevo nombre esta repetido para este usuario.
- **10B:** El nuevo identificador interno esta repetido para este usuario.
- **10C:** Alguno de los campos nuevos es nulo o vacío.

Mock:

TODO MOCK PANTALLA MODIFICACIÓN DE CONDUCTOR

	ACTA DE PROYECTO – CloudFleet	
Versión 1	1/4/2016	

11: Reporte de gestión del conductor:

Propósito: El usuario debe poder ver un informe de gestión que le permita evaluar el desempeño de un conductor en particular. Así como su nivel de actividad y horarios.

Precondiciones:

- El usuario debe estar registrado.
- El usuario debe estar logueado.
- El usuario tiene conductor/es registrado/s
- El usuario tiene vehículo/s registrado/s
- El conductor manejó alguno de los vehículos en el periodo a evaluar.

Éxito:

Se muestran los siguientes gráficos interactivos:

- Velocidades x período.
- Aceleraciones x período.
- Distancias recorridas x período.
- Vehículos utilizados y proporción de la distancia recorrida
- Horarios en los que el conductor manejó.
- Repaso general del período y números del conductor.
- Alertas generadas durante el período evaluado.

No hay flujos alternativos.

Mock:

TODO MOCK PANTALLA REPORTING

12: Baja de conductor:

Propósito: El usuario debe poder eliminar conductores en caso de error o desvinculación.

Precondiciones:

- El usuario esta registrado
- El usuario esta logueado
- El conductor a eliminar esta registrado

Éxito: El conductor es eliminado.

No hay flujos alternativos.

13: Visualizar Dashboard:

Propósito: El usuario debe poder ver la situación de su flota y tener un panorama claro de las alertas involucradas. Todo esto actualizado en tiempo real.

	ACTA DE PROYECTO – CloudFleet	
Versión 1	1/4/2016	

Precondiciones:

- El usuario debe estar registrado.
- El usuario debe estar logueado.
- El usuario tiene conductor/es registrado/s
- El usuario tiene vehículo/s registrado/s
- Algun conductor maneja alguno de los vehículos.

Éxito:

- El sistema muestra todos los vehículos en uso actualmente y su recorrido de los últimos minutos

No hay flujos alternativos.

Mock:

TODO MOCK DEL DASHBOARD

14: Detalle de alerta

Propósito: Si el usuario ve una alerta, debe poder clickear en ella y ver el lugar donde se encuentra el vehículo, su nombre, el de su conductor y su situación actual.

Precondiciones:

- El usuario debe estar registrado.
- El usuario debe estar logueado.
- El usuario tiene conductor/es registrado/s
- El usuario tiene vehículo/s registrado/s
- Algun conductor maneja alguno de los vehículos.
- El usuario debe estar viendo el dashboard.
- Hay una alerta para alguno de los vehículos en los últimos 10'

Éxito:

- Se marca automáticamente el vehículo alertado en el mapa.

No hay flujos alternativos:

Mock:

##TODO MOCK CLICK ALERTA ##

15: Detalle de vehículo

Propósito: Dado el mapa de vehículos, debo poder seleccionar uno para seguirlo de cerca, ver información de velocidades, conductor y posibles alertas.

		ACTA DE PROYECTO – CloudFleet	
Versión 1		1/4/2016	

Precondiciones:

- El usuario debe estar registrado.
- El usuario debe estar logueado.
- El usuario tiene conductor/es registrado/s
- El usuario tiene vehículo/s registrado/s
- Algun conductor maneja alguno de los vehículos.
- El usuario debe estar viendo el dashboard.
- Hay una alerta para alguno de los vehículos en los últimos 10'

Éxito:

- Se sigue al vehículo con la etiqueta de información correspondiente.

No hay flujos alternativos

Mock:

TODO ACTIVITY DIAGRAM Refresco AJAX del Mapa y las Alertas

Las APIs del sistema:

El sistema cuenta con cuatro grupos de APIs:

- **Trackeos del dispositivo:** Es la API utilizada por el dispositivo de trackeo para publicar la información nueva respecto a un dispositivo de trackeo durante uno o varios minutos:
 - **Número de serie:** es el número de serie del dispositivo. Cada trackeador vendido tendrá un código diferente que lo diferencia de los demás. El Número de serie es informado, y es requisito, al momento de registrar un vehículo en la plataforma. El mismo no se puede repetir entre dos vehículos.
 - **Identificador de conductor:** es el identificador del conductor dentro de la flota de un usuario. Se utilizará para posteriores controles. En caso de no informarlo se disparará una alerta avisando que el vehículo se está desplazando sin informar el conductor responsable.
 - **Data:** Es un vector de datos de varios pedidos, se puede usar para enviar varios POSTs en uno solo en caso de que el trackeador se haya atrazado.
 - **Período:** el periodo informado correspondiente al minuto en formato 'YYYYMMDDHH24MI'
 - **Ubicaciones del dispositivo:** Se envían 6 puntos de latitud y longitud cada, uno para cada 10 segundos de la trayectoria. El formato es en grados radianes decimales.
 - **Velocidad:** se envía la velocidad máxima, mínima y los percentiles 25 y 75 del minuto en cuestión. La unidad es en kilómetros por hora.
 - **Aceleración:** se envían las aceleraciones máximas del minuto evaluado en dos sentidos de las tres dimensiones. Si el dispositivo no cuenta con sensores en todas las dimensiones puede omitir aquellas donde no se tenga información. La unidad es g (gravedades terrestres: 9.8m/s²).

ACTA DE PROYECTO – CloudFleet

Versión 1

1/4/2016

```
# HTTP Method: POST
# Resource URL: /tracks
# Payload:
{
  "serial_no": "AAAA19",
  "driver_internal_id": "4",
  "data": [
    {
      "period": "20160701234000",
      "speed": {
        "max": 100,
        "avg": 80,
        "min": 50,
        "p75": 90,
        "p25": 55
      },
      "acceleration": {
        "up": 0.1,
        "down": 0.3,
        "forward": 0.4,
        "backward": 0.1
      },
      "locations": [
        {
          "lat": 22.1231, "long": 44.23243},
          {"lat": 22.1231, "long": 44.23243},
          {"lat": 22.1231, "long": 44.23243},
          {"lat": 22.1231, "long": 44.23243},
          {"lat": 22.1231, "long": 44.23243}
        ]
      }
    }
  ]
}
```

- **Alertas:** La API de alertas es el mecanismo mediante el cual, cualquier dispositivo o sistema externo puede generar una alerta para un vehículo determinado. La intención es poder subir alertas que son generadas desde el dispositivo de trackeo en forma directa o alertas que son generados con sistemas externos, por ejemplo un servicio de recepción telefónica de denuncias. Los datos recibidos son:
 - **Número de serie:** es el número de serie del dispositivo. Cada trackeador vendido tendrá un código diferente que lo diferencia de los demás. El Número de serie es informado, y es requisito, al momento de registrar un vehículo en la plataforma. El mismo no se puede repetir entre dos vehículos. Se proveerán stickers para el vehículo con QR Codes con una URL con este código incluido para la recepción de quejas de otros conductores.
 - **Identificador de conductor:** es el identificador del conductor dentro de la flota de un usuario. Se utilizará para posteriores controles. En caso de no informarlo se disparará una alerta avisando que el vehículo se está desplazando sin informar el conductor responsable.
 - **Tipo de alerta:** String identificador del tipo de alerta.
 - **SPEEDING:** Exceso de velocidad
 - **COMPLAINT:** Queja de otro conductor
 - **CRASH:** Posible accidente
 - **ZONE:** Fuera de zona
 - **LOW_FUEL:** Combustible bajo
 - **FUEL_CONS:** Consumo de combustible alto
 - **NO_DRIVER:** No hay conductor definido
 - **OBDI:** Alerta mecánica de OBDI
 - **Datos adicionales:** Detalles del motivo de la alerta. Ejemplos:
 - **SPEEDING:**
 - Supero el máximo de 120Km/h
 - **COMPLAINT:**

ACTA DE PROYECTO – CloudFleet

Versión 1

1/4/2016

- Maneja ebrio
- Maneja a alta velocidad
- Maneja imprudentemente
- Me agredió
- **CRASH:**
 - Posible siniestro (Desaceleración)
- **ZONE**
 - Fuera de la zona operativa establecida
- **LOW_FUEL:**
 - Menos de 30Km de autonomía de combustible,
- **FUEL_CONS:**
 - Consumo exagerado de combustible
- **NO_DRIVER:**
 - El conductor no se identifico
- **OBDI:**
 - Service vencido
 - Falla de motor
 - Advertencia de motor
 - Neumaticos desinflados
 - Verificar aceite
 - Calentamiento de motor
- **Período:** el periodo informado correspondiente al minuto en formato 'YYYYMMDDHH24MI'
- **Latitud:** la latitud en la cual fue generada la alerta si la misma esta disponible. Fromato decimal radianes.
- **Longitud:** la longitud en la cual fue generada la alerta si la misma esta disponible. Fromato decimal radianes.

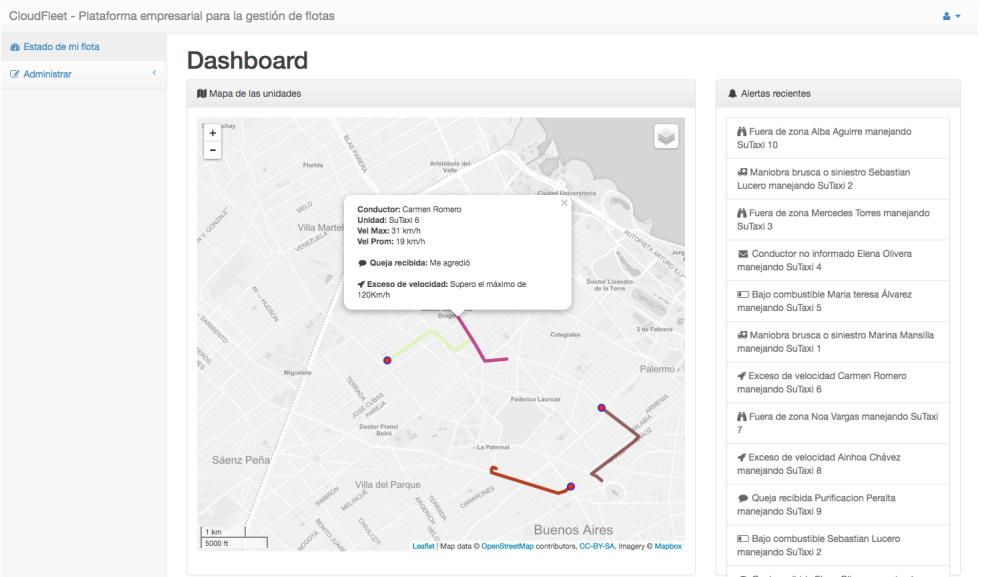
```
# HTTP Method: POST
# Resource URL: /alerts
# Payload:
{
    "serial_no": "AAAA19",
    "driver_internal_id": "4",
    "alert_type": "SPEEDING",
    "additional_data": "SPEEDING",
    "period": "20160701234000",
    "latitude": 22.1231,
    "longitude": 44.23243
}
```

- **Dashboard:** La api de dashboard esta pensada para responder los pedidos AJAX del frontend de monitoreo. Incluye la información de los vehículos respecto a posiciones, velocidades, aceleraciones y alertas del vehículo durante los últimos 20 minutos. Esta funcionalidad es utilizada desde el frontend para refrescar la información de los vehículos en pantalla y las alertas para un usuario determinado.
 - **La pantalla:**

ACTA DE PROYECTO – CloudFleet

Versión 1

1/4/2016



- **Información:** La respuesta a este pedido de API retorna información sobre todos los vehículos activos del usuario en cuestión. La información que retorna es la siguiente para cada vehículo:
 - **ID del Vehículo:** identificador único del vehículo en consideración, es utilizado para matchear los contenidos actualizados con las estructuras de rendering ya existentes en el DOM HTML.
 - **Nombre del vehículo:** es un nombre interno del usuario, conveniente para identificar los miembros de su flota. Por ejemplo, si tengo taxis puede ser el código de la unidad.
 - **Número de serie:** es el número de serie del dispositivo. Cada trackeador vendido tendrá un código diferente que lo diferencia de los demás. El Número de serie es informado, y es requisito, al momento de registrar un vehículo en la plataforma. El mismo no se puede repetir entre dos vehículos.
 - **Nombre del conductor:** El nombre del conductor que esta al frente de la unidad. Si nadie se autenticó en el dispositivo de trackeo, entonces el campo será nulo.
 - **Velocidad máxima en el último minuto:** La velocidad máxima a la que transito la unidad en el último minuto evaluado.
 - **Velocidad promedio:** Velocidad promedio de circulación del último minuto evaluado.
 - **Aceleración adelante:** valor en g (aceleraciones gravitatorias terrestres: 9.8 m/s²) de aceleración hacia adelante durante el último minuto evaluado de conducción. Valores altos indican una conducción agresiva.
 - **Aceleración hacia atrás:** valor en g de las frenadas del vehículo. Misma unidad que el anterior. Valores altos pueden indicar un accidente, un frenazo o una conducción agresiva dependiendo el valor.
 - **Aceleración hacia arriba y abajo:** Mismas unidades que las otras aceleraciones, estos valores sirven para detectar empedrados, lomos de burro y otras cuestiones relacionadas con el tipo de camino y la agresividad de la conducción.
 - **Ubicaciones:** Son las ubicaciones de los últimos 10 minutos, sirven para dibujar

ACTA DE PROYECTO – CloudFleet

Versión 1

1/4/2016

la estela del camino del vehículo en el mapa. Están en coordenadas decimales en radianes. Organizadas en forma de vector.

- **Alertas:** Es un vector con las alertas para cada unidad en los últimos 20'. Cada alerta contiene la siguiente información:

- **Período:** El período en el que ocurrió la alerta en formato 'YYYYMMDDHH24MI'.
- **Ubicación:** Latitud y longitud en radianes decimales del lugar en el mapa en el que ocurrió la situación.
- **Tipo de alerta:** String identificador del tipo de alerta. Ver listado (ref: API de alertas)
- **Datos adicionales:** Detalles de la alerta en particular.
- **Descripción:** Descripción del tipo de alerta en cuestión, pero en general.

```
# HTTP Method: GET
# Resource URL: /users/4/dash_data
# Response:
{
  "3": {
    "id": 3,
    "name": "SuTaxi_1",
    "tracking_serial_no": "BBBB1",
    "driver_name": "Marina Mansilla",
    "speed_max": 71,
    "speed_avg": 56,
    "acceleration_forward": 0.28,
    "acceleration_backward": -0.35,
    "acceleration_up": 0.0,
    "acceleration_down": 0.0
    "locations": [
      [
        -34.606389721377234,
        -58.52377600295732
      ],
      [
        -34.60532244804244,
        -58.52349119645121
      ],
      ...
    ]
  },
  "4": {
    "id": 4,
    "name": "SuTaxi_2",
    "tracking_serial_no": "BBBB2",
    "driver_name": "Sebastian Lucero",
    "speed_max": 73,
    "speed_avg": 34,
    "acceleration_forward": 0.28,
    "acceleration_backward": -0.35,
    "acceleration_up": 0.0,
    "acceleration_down": 0.0
    "locations": [
      [
        -34.77331662960827,
        -58.28190532623246
      ],
      [
        -34.773676632455576,
        -58.281761659356086
      ],
      ...
    ],
    "alerts": [
      {
        "period": "20160615001500",
        "latitude": -34.78691646850799,
        "longitude": -58.33633611880325,
        "alert_type": "CRASH",
        "description": "Posible accidente",
        "additional_data": "Posible siniestro (Desaceleración)"
      }
    ]
  }
}
```

	ACTA DE PROYECTO – CloudFleet	
Versión 1	1/4/2016	

- **Modelos, versiones y precios:** Las APIs de modelos versiones y precios existen para asistir a los frontends de “Alta de vehículo” y “Modificación de vehículos”. Existen debido a que en la base de datos hay 95.000 tipos de vehículos distintos entre Paises, Marcas, Modelos, Versiones y Años de fabricación diferentes. Por este motivo la página principal solo carga las marcas del pais en el que opera el usuario
 - **API de Modelos:** Una vez que el usuario selecciona la marca del vehículo en cuestión se hace un llamado AJAX a la api de modelos para solicitar los modelos de esa marca. El formato del llamado es: “*GET /brands/46/models*”. Este llamado popula el combo de modelos.
 - **API de Versiones:** Con el campo de Modelos poblado, ahora el usuario puede seleccionar una versión del modelo. Cuando lo hace se dispara un llamado del formato: “*GET /brands/46/models/439/versions*” y esto popula el combo de versiones para que el usuario pueda elegir la versión correspondiente a su vehículo.
 - **API de Precios:** Con el campo de versiones seleccionado se dispara un tercer llamado AJAX para popular el combo de precios, que contiene el año de fabricación y precio. El formato es: “*GET /brands/46/models/439/versions/5658/prices*” y retorna todos los años en los que es posible que se haya fabricado esa versión de ese modelo de esa marca y su precio asociado. Identificar el vehículo con este nivel de precisión será muy útil a la hora de ofrecer servicios como seguros, auxilios mecanicos etc.
 - **Ejemplo de Formulario completo con AJAX:**

Marca	<input style="width: 100%;" type="text" value="VOLKSWAGEN"/>
Modelo	<input style="width: 100%;" type="text" value="Vento"/>
Version	<input style="width: 100%;" type="text" value="2.0 TSI SPORTLINE DSG (200CV) (L11)"/>
Año	<input style="width: 100%;" type="text" value="2011: \$269000"/>

	ACTA DE PROYECTO – CloudFleet	
Versión 1	1/4/2016	

Modelo de datos:

Para almacenar la información de respaldo al sistema expuesto, es necesaria la creación de las siguientes tablas en la base de datos.

Detalle de tablas:

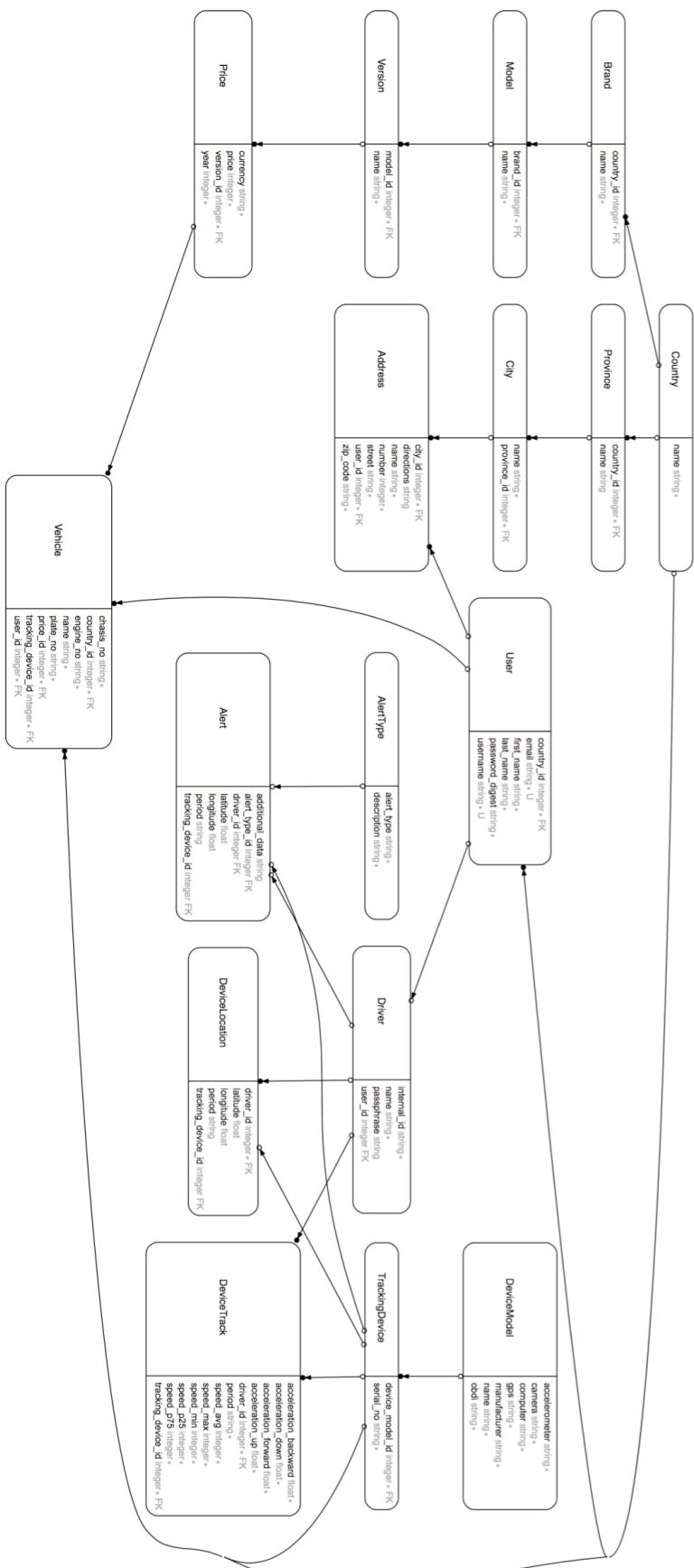
Nombre de tabla	Tipo	Propósito
alert_types	Parametrica	Listado de los posibles tipos de alertas: Velocidad, Denuncias etc.
alerts	Datos	Listado exhaustivo de las alertas recibidas para cada vehículo en cada lugar y momento. Es una tabla muy grande.
brands	Parametrica	La lista de marcas por país. Contiene una lista de algo mas de 30 marcas por cada país
cities	Parametrica	La lista de ciudades para cada provincia.
countries	Parametrica	La lista de países donde opera CloudFleet
device_locations	Datos	El detalle exaustivo de las ubicaciones donde estuvo un dispositivo de trackeo. Resolución de 10 segundos. Es la tabla mas grande del sistema.
device_models	Parametrica	Lista corta de modelos de dispositivo y las funcionalidades asociadas que tiene.
device_tracks	Datos	Trackeos de un dispositivo determinado, datos de velocidad, aceleración etc. Resolución de un minuto, es la segunda tabla mas grande del sistema.
drivers	Datos	Lista de conductores, es la lista de conductores habilitados para manejar vehículos de un usuario.
models	Parametrica	Es la lista de modelos de una marca. Ej, para Volkswagen: Gol, Bora, Vento. Etc.
prices	Parametrica	Es la lista de años y precios que una versión determinada de un vehículo puede tener.
provinces	Parametrica	Es la lista de provincias que tiene un país.
users	Datos	Los usuarios (empresas) que usan la plataforma cloudfleet.
vehicles	Datos	Vehículos de una empresa (usuario)
versions	Parametrica	Versiones de un modelo de una marca determinada en un país.

ACTA DE PROYECTO – CloudFleet

Versión 1

1/4/2016

DER



ACTA DE PROYECTO – CloudFleet

Versión 1

1/4/2016

Parametrización de la base de datos:

Las tablas paramétricas son aquellas que contienen información necesaria para que el sistema funcione. En muchos sistemas, la información paramétrica es sencilla, pero en este caso supone todo un desafío y es una parte importante del costo de este proyecto.

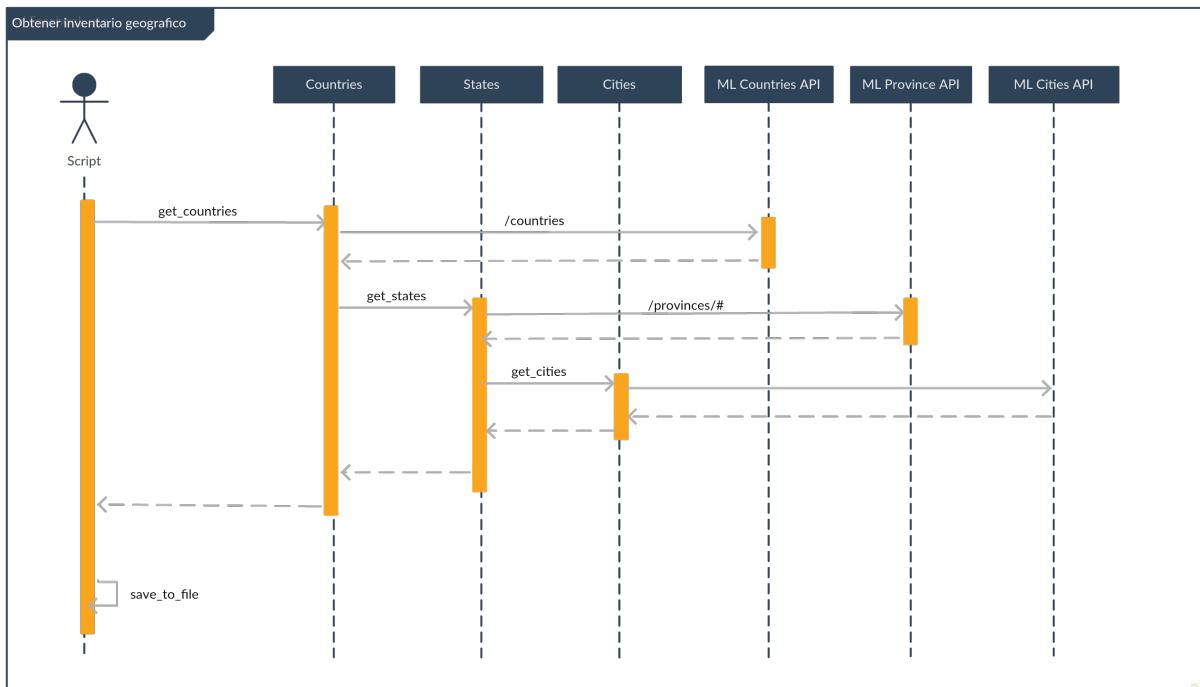
Las parametrizaciones se dividen en tres grandes grupos

- **Datos geográficos:** Tener un muy buen detalle del país de residencia, ciudad, provincia etc. Es un aspecto clave de la tabulación de la información. Que luego servirá a la hora de establecer correlaciones, establecer servicios nuevos etc. La información geográfica necesaria para operar son:
 - **Paises:** Argentina, Colombia, Uruguay, Brasil, Mexico y Venezuela.
 - **Provincias:** Todas las provincias de cada uno de esos países y su vínculo entre ellos.
 - **Ciudades:** Todas las ciudades de cada una de las provincias y el vínculo entre ellas.

Para poder satisfacer este requerimiento se utilizó la API de MercadoLibre.Com

(<http://developers.mercadolibre.com>). Se codificó el script ruby

download_cities_and_cars_data.rb con el cual también se descarga la información de fabricantes.



- **Datos de fabricantes de vehículos: Marcas, Modelos, Versiones y precios:** Tener un detallado inventario de los automóviles, modelos, versiones y precios por año para cada país es muy importante en términos de tabulación de la información. Y es clave para que empresas de seguro puedan operar en la plataforma. Para cumplir con este requerimiento la mejor opción consistió en navegar en grafo de APIs de mercadolibre que contiene los inventarios completos de los países en cuestión. En cada caso son obtenidos y organizados por mercadolibre para de los entes reguladores o las cámaras industriales de cada país. En el caso de Argentina por ejemplo se trata de ACARA. Mediante el uso de este mecanismo logramos recopilar una base de datos muy completa con 95.000 variaciones de vehículos. Las entidades en cuestión son:
 - **Marca:** Las marcas que se comercializan en un país determinado, ej: VOLKSWAGEN,

	ACTA DE PROYECTO – CloudFleet	
Versión 1	1/4/2016	

VOLVO, FIAT, FORD, etc.

- **Modelo:** Los modelos de cada marca y su vínculo con la misma. Ej: para Volkswagen: *Vento, Bora, Gol.*
- **Versión:** La versión especifica de un modelo y su vinculo con modelo y marca: Ej: *Sportline DSG 200CV (L11)*.
- **Año/Precio:** La tupla año y precio y el vínculo con una versión determinada indica el precio de cada versión de un modelo de una marca dependiendo del año de fabricación de la unidad.
- **Datos internos de CloudFleet:**
 - **Tipos de alertas:** Son insertados uno por uno en el script seeds.rb.

Simulación de casos de uso

Con el objetivo de poder probar la aplicación y hacer demostraciones de funcionamiento con volumen, tanto a clientes como a inversores y ante la ausencia de clientes iniciales con gran volumen se incluyen dentro del presupuesto inicial de desarrollo todas aquellas tareas que estan relacionadas con la generación de datos validos que permitan vender la herramienta y conceptualizar su uso. Las entidades a simular son:

- **Dispositivos:**
- **Población de usuarios:**
- **Población de vehículos:**
- **Población de conductores:**
- **Simulación de recorridos:**
 - **## TODO CLASS DIAGRAM ##**
 - **## TODO ACTIVITY DIAGRAM ##**
 - **## TODO FLOW CHART ##**
- **Simulación de trayectorias:**
- **Simulación de aceleraciones:**
- **Simulación de alertas:**

Planificación del proyecto

Como parte del proceso de subcontratación del desarrollo de la versión inicial del software y su presupuesto se ha realizado un Gant con tareas Jerárquicas y subtareas de mayor profundidad y detalle con el proposito de entender el costo y de fijar un plazo de entrega del código fuente.

Los hitos más importantes son:

- **Desarrollo y población de estructuras de datos básicas:** La creación de las estructuras de datos de parametrización y la búsqueda y recolección de fuentes para estos datos mediante scripts automatizados que permitan la actualización de los mismos in trabajo humano. (Ej: 95.000 precios de vehículos)
- **Construcción de entidades principales:** Implica la creación de todas aquellas entidades que son la base fundacional de la aplicación, como por ejemplo los usuarios y el login. Estas son las

ACTA DE PROYECTO – CloudFleet		
Versión 1	1/4/2016	

que permitirán avanzar en otros aspectos más avanzados y que darán forma a la aplicación.

- **Construcción del motor de tracking:** El motor de tracking son todas las funcionalidades tanto de API como de frontend y los detalles avanzados como refresco AJAX para cumplir con el trackeo de las unidades en tiempo real.
- **Construcción del motor de reporting:** Por último, cuando el sistema ya cuenta con los datos necesarios para operar la construcción de la funcionalidad de reporte de comportamiento de un conductor en particular, que permitirá evaluar el desempeño de los conductores de forma independiente.

ACTA DE PROYECTO – CloudFleet

Versión 1

1/4/2016

	Name	Duration	Start	Finish	Predecessors	Resource Names	15 May 16	22 May 16	29 May 16	5 Jun 16	12 Jun 16	19 Jun 16
1	Desarrollo y población de las estructuras de datos básicas	9 days	5/16/16 8:00 AM	5/26/16 5:00 PM			F S S M T W T F S M T W T F S M T W T F S S M T W T F S M T W					
2	Países, provincias y ciudades	4 days	5/16/16 8:00 AM	5/19/16 5:00 PM								
3	Utilización de la API de interfaz del motor para obtener la información y bajarla a JSON	2 days	5/16/16 8:00 AM	5/17/16 5:00 PM								
4	Seed inicial de la información a las estructuras de datos	2 days	5/16/16 8:00 AM	5/19/16 5:00 PM	3							
5	Marcas, modelos, versiones, prejos	7 days	5/16/16 8:00 AM	5/26/16 5:00 PM								
6	Utilización de la API de interfaz del motor en gratis a JSON	4 days	5/18/16 8:00 AM	5/23/16 5:00 PM	3							
7	Seed inicial de información	3 days	5/24/16 8:00 AM	5/26/16 5:00 PM	6							
8	Construcción de entidades principales	10 days	5/16/16 8:00 AM	5/27/16 5:00 PM								
9	Home Page. Menús y estilos	2 days	5/16/16 8:00 AM	5/17/16 5:00 PM								
10	Usuario y autenticación	3 days	5/16/16 8:00 AM	5/19/16 5:00 PM								
11	Alta de usuario	1 day	5/16/16 8:00 AM	5/16/16 5:00 PM								
12	Login, sesiones y logout	2 days	5/17/16 8:00 AM	5/18/16 5:00 PM	11							
13	Vehículos	7 days	5/19/16 8:00 AM	5/27/16 5:00 PM	10							
14	Alta de vehículo	2 days	5/19/16 8:00 AM	5/20/16 5:00 PM								
15	Ajax API gestión de modelos	1 day	5/27/16 8:00 AM	5/27/16 5:00 PM	14,7							
16	Modificación de vehículos	1 day	5/23/16 8:00 AM	5/23/16 5:00 PM	14							
17	Listado y gestión de vehículos	1,5 days	5/23/16 8:00 AM	5/24/16 1:30 PM	14							
18	Conductores	2,5 days	5/19/16 8:00 AM	5/23/16 5:00 PM	10							
19	Alta de conductores	1 day	5/19/16 8:00 AM	5/19/16 5:00 PM								
20	Modificación de conductores	1 day	5/20/16 8:00 AM	5/20/16 5:00 PM	19							
21	Listado y gestión de conductores	1,5 days	5/20/16 8:00 AM	5/23/16 1:00 PM	19							
22	Construcción de motor de tracking	18,25 days	5/16/16 8:00 AM	6/9/16 10:30 AM								
23	Dispositivos de trackeo tipos y unidades	2 days	5/30/16 8:00 AM	5/31/16 5:00 PM	8							
24	Trackeo de dispositivo y ubicaciones	4 days	5/30/16 8:00 AM	6/2/16 5:00 PM	8							
25	Construcción de entidades	2 days	5/30/16 8:00 AM	5/31/16 5:00 PM								
26	Construcción de API	2 days	6/1/16 8:00 AM	6/2/16 5:00 PM	25							
27	Simulación de conducción de vehículos	14,75 days	5/16/16 8:00 AM	6/3/16 3:00 PM								
28	Utilización de API de Google Maps, ubicaciones zonas	4 days	5/16/16 8:00 AM	5/19/16 5:00 PM								
29	Motor de lista (Aceleraciones, velocidades y Nombres)	4 days	5/20/16 8:00 AM	5/23/16 5:00 PM	28							
30	Modelo de compañía de Taxis, Unidades y Nombres	2,75 days	5/26/16 8:00 AM	5/30/16 3:30 PM	29							
31	Modelo de trayectos en el comunitario y ciudad de Buenos Aires	4 days	5/30/16 3:00 PM	6/3/16 3:00 PM	30							
32	Dashboard de monitoreo de vehículos en tiempo real	9 days	5/16/16 8:00 AM	5/26/16 5:00 PM								
33	Integración de tablejs open street map	4 days	5/16/16 8:00 AM	5/19/16 5:00 PM								
34	Agregado de componentes diáfanos tablejs	3 days	5/20/16 8:00 AM	5/24/16 5:00 PM	33							
35	Actualización en background algoritmo AJAX	2 days	5/25/16 8:00 AM	5/28/16 5:00 PM	34							
36	Construcción del motor de alertas	8,25 days	5/30/16 8:00 AM	6/9/16 10:30 AM	8							
37	Creadión de la entidad y creación de API de generación de alertas	3 days	5/30/16 8:00 AM	6/1/16 5:00 PM								
38	Modelo de alertas para la flota de taxis	2 days	6/2/16 8:00 AM	6/3/16 5:00 PM	37							
39	Integración de las alertas a dashboard de monitoreo de vehículos en tiempo real	1,75 days	6/6/16 8:00 AM	6/7/16 3:00 PM	38,32							
40	Integración del servicio AJAX	1 day	6/7/16 3:00 PM	6/8/16 3:00 PM	39							
41	Vínculo entre las alertas y los layers del mapa	0,5 days	6/8/16 3:00 PM	6/9/16 10:00 AM								
42	Construcción de motor de reporting	7 days	6/9/16 10:00 AM	6/20/16 10:00 AM	22,36,27							
43	Estructura del reporte general, links y gestión de las fechas	1 day	6/9/16 10:00 AM	6/10/16 10:00 AM								
44	Integración de estilos y librerías de Charts.js	4 days	6/9/16 10:00 AM	6/13/16 10:00 AM								
45	Reporte de velocidad, aceleración y distancia de conductor	3 days	6/15/16 10:00 AM	6/20/16 10:00 AM	43,44							

ACTA DE PROYECTO – CloudFleet

Versión 1

1/4/2016

Despliegue productivo:

Para la operación de la compañía, será preciso montar la infraestructura necesaria. Para operar hemos decidido hacerlo en el Cloud Público, los motivos son los siguientes:

- Foco en salir rápido al mercado. Los procesos de compra e instalación de infraestructura llevan mucho tiempo.
- Infraestructura propia necesaria es privativa para el tamaño inicial de cloud fleet por los elevados costos fijos.
- Se espera que la empresa crezca muy rápidamente o en su defecto nunca prospere, en cualquiera de los dos casos tener la posibilidad de contar con infraestructura variable acorde a la demanda, es un aspecto clave que solo el cloud público nos puede brindar.
- La versatilidad para la aplicación de políticas rigurosas de seguridad.
- La posibilidad de ejecutar Workloads On Demand para la minería de datos (BigData).
- La necesidad de contar con tecnologías de DataWarehouse económicas.

Hemos decidido hacer nuestro despliegue en Amazon Web Services para hacer el despliegue, los motivos son:

- Líder en el mercado. (Facturación 2015 3x que su competidor siguiente Microsoft).
- Líder indiscutido según Gartner 2015 (Gartner Magic Quadrant).
- Recursos entrenados en el mercado para la administración y desarrollo en AWS
- Cantidad de productos disponibles para iniciativas estratégicas de CloudFleet: BigData, Wharehousing y Internet of Things.



Dado que en AWS existen grandes descuentos por la reserva y pago anticipado de los recursos de computo utilizado, hemos trabajado con nuestro CFO, el Sr Pedro Ramirez (CFO) y acordamos hacer reservas por los recursos mínimos para que la operación funcione durante un año y luego pagar precios mayores “OnDemand” para aquellos consumos adicionales producto del crecimiento del negocio y hacer una revisión de los costos y las posibilidades de reservas cada 3 meses.

Para la versión inicial, trabajamos en un presupuesto para:

- Período de un año.
- Operatoria mínima viable.
- Stack completo de tecnologías.

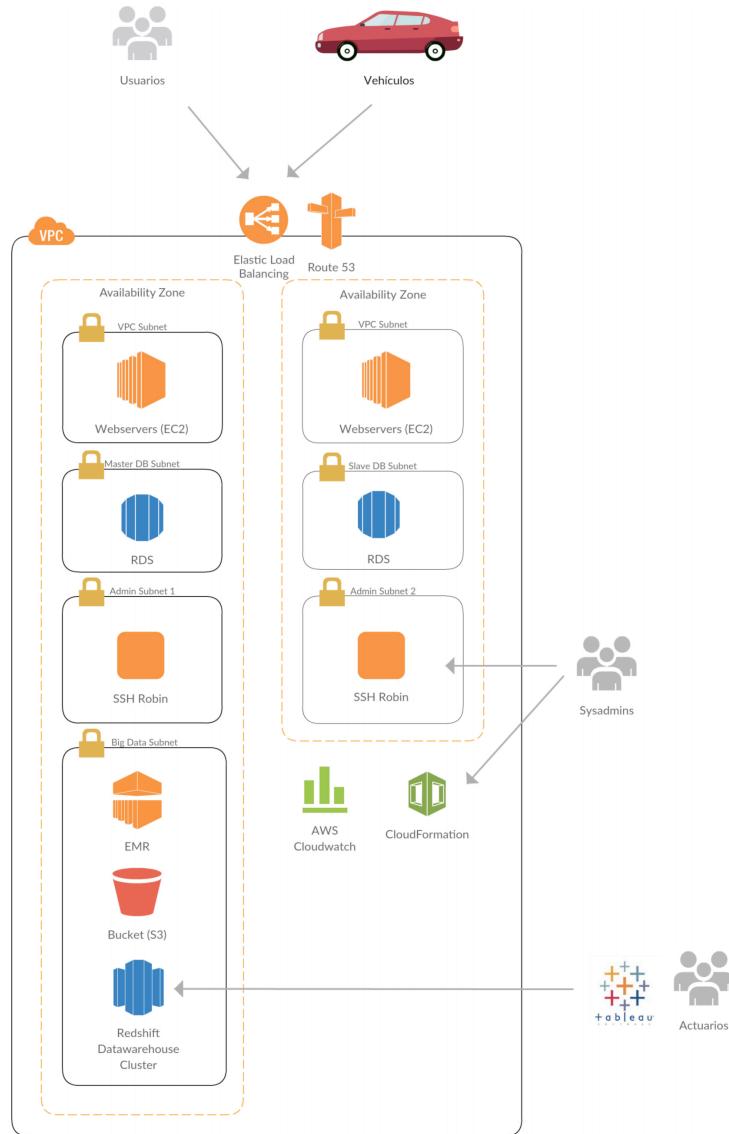
ACTA DE PROYECTO – CloudFleet

Versión 1

1/4/2016

El stack tecnológico:

En el stack tecnológico están considerados los costos de hosting tanto del sitio como de las APIs así como todos los servicios de Big Data, Datawarehousing y monitoreo de las aplicaciones desplegadas en el cloud.



Las capas son:

- **WebServers**: Servidores que corren la aplicación Ruby. Application Programming Interface (API) y Frontend HTML5 Reactivo. Están ubicados en hasta 4 datacenters distintos.
- **Base de datos**: Base de datos Aurora, propietaria de AWS compatible con MySQL, permite escalar las lecturas y escrituras mediante mecanismos de clustering. También conserva 6 copias de los datos para alta durabilidad en datacenters distintos.
- **Datawarehouse**: Se usará Amazon Redshift por su flexibilidad para escalar, bajo costo inicial comparado con otras herramientas como Teradata o Hexadata y la flexibilidad para importar información de S3.
- **BigData**: El almacenamiento de grandes volúmenes de información se realizará en S3 (Simple Storage Service) por su bajo costo, sencillez y durabilidad de 99.999999%. Se usará EMR (Elastic Map Reduce) para la ejecución de Jobs de Big Data con Hadoop y Spark.

La alta disponibilidad y la recuperación de desastres

El despliegue de la aplicación es un aspecto clave de la alta disponibilidad de la aplicación y la gestión ante desastres.

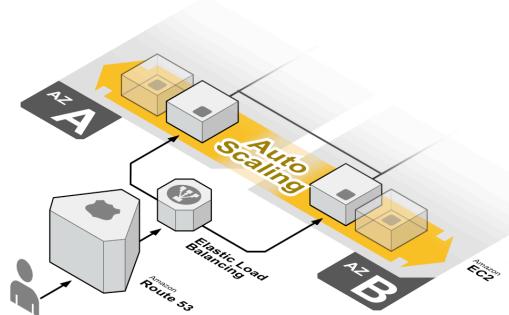
El objeto de este diseño es asegurar la continuidad del negocio ante una serie de sucesos externos que podrían suceder, como por ejemplo:

Alta disponibilidad:

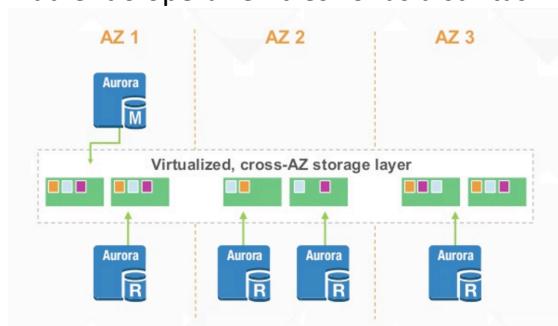
- Problemas de energía en un Datacenter
- Implementaciones fallidas.
- Problemas de disponibilidad de la conectividad.

La alta disponibilidad ataca los problemas más frecuentes y probables pero de menor impacto. Para mitigar estos problemas, CloudFleet estará desplegado en múltiples datacenters y con mecanismos automáticos de fail over.

- **WebServers:** Los webservers estarán asociados a Auto Scaling Groups, una entidad de Amazon WebServices que permite administrar un cluster de servidores automáticamente. Adaptando la cantidad de los mismos a la utilización demandada y la disponibilidad de las Availability Zones (Datacenters dentro de una region). En caso de perdida de un Datacenter, el servicio automáticamente redimensionará en los restantes para atender la demanda. Se cuenta con cuatro Availability Zones en la region de Virginia.



- **Database:** La base de datos operará en RDS Aurora (Amazon Web Services). Este sistema de bases de datos permite un failover automático e instantáneo a otra Availability Zone en caso de que una zona falle. Pudiendo operar en tres zonas distintas.



	ACTA DE PROYECTO – CloudFleet	
Versión 1	1/4/2016	

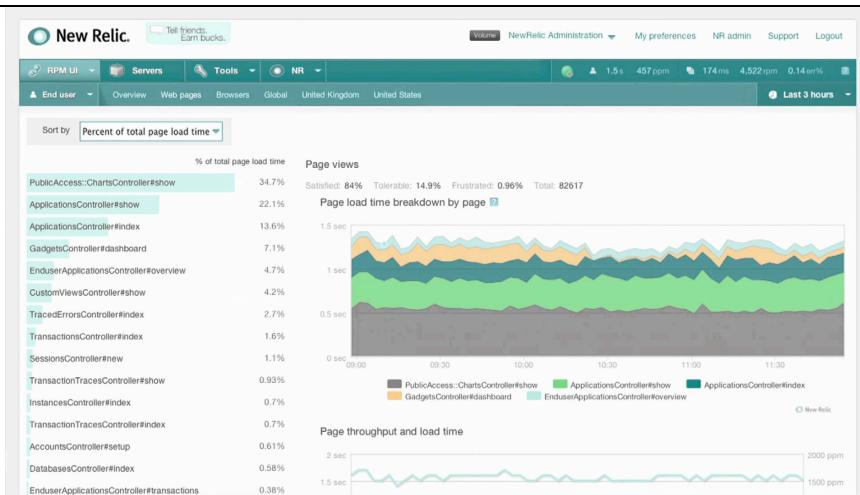
- **BigData y Datawarehouseing:** Los datos seran almacenados en S3 que tiene altisima durabilidad y copias en 6 ubicaciones. Respecto a las herramientas de trabajo, podrán tener un impacto operativo ante la perdida de un datacenter pero si la falla es prolongada (>2hs) podrá implementarse rapidamente un failover en otra zona sin costo alguno.

Además de una arquitectura resiliente en todas las capas y servicio es imperativo contar con herramientas y procesos adecuados para la gestión de incidentes.

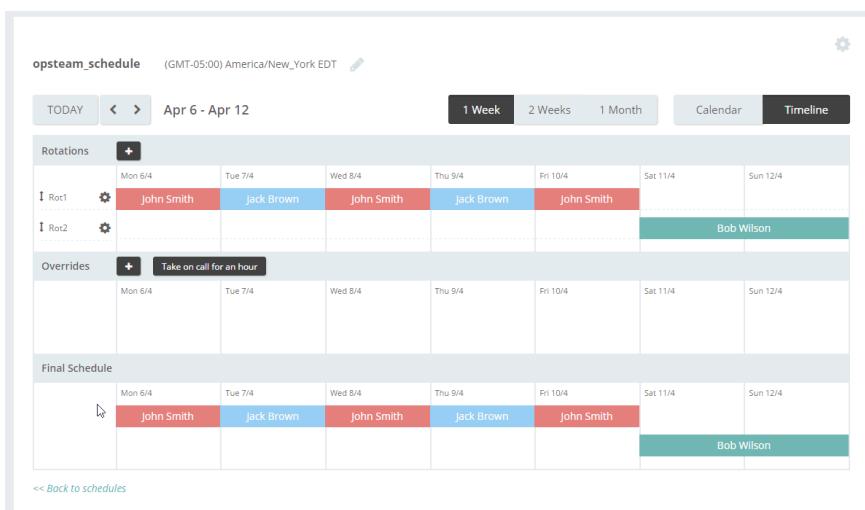
Un monitoreo y plan de reacción eficiente consta de dos partes fundamentales:

- **Detección del problema:** Es importante tener controles efectivos y abarcativos sobre el funcionamiento de la aplicación y el negocio. Hay dos grandes problemas a evitar en esta parte.
 - **Ausencia de control:** los controles abarcativos como metricas HTTP o aspectos del negocio, suelen ser una buena forma de evitar que algún aspecto de la aplicación o la infraestructura opera sin los controles necesarios.
 - **Falsos positivos:** La configuración de alertas con umbrales poco realistas o agresivos generará alertas por cuestiones que si bien pueden ser poco deseables son habituales. Estos problemas no se resuelven los sábados a las 3AM y es importante mantener las alertas en el ámbito de lo atípico para mantener una cultura comprometida que entienda las alertas como algo que se debe accionar.
 - **New Relic:** Es la herramienta elegida para ejecutar el monitoreo de la aplicación e infraestructura. Usando NewRelic se pueden configurar alertas a todos los niveles:
 - **Client Side RUM:** Real User Monitoring permite mediante un plugin JavaScript en la aplicación Web medir y detectar problemas con versiones de Browsers o conectividad lenta, descarga de contenidos etc.
 - **HTTP Monitoring:** Permite verificar tiempos de respuesta y cantidad de errores de la aplicación del lado del servidor.
 - **Server Monitoring:** Permite advertir problemas relacionados con la configuración o la capacidad de computo.
 - **Insights:** Permite hacer un control en tiempo real de las métricas de negocio cuya afectación puede implicar un problema técnico. Por ejemplo: ningún vehículo reporta la ubicación en la última hora.

	ACTA DE PROYECTO – CloudFleet	
Versión 1	1/4/2016	



- **Notificación efectiva:** La habilidad de notificar a la persona correcta para resolver el problema y dar seguimiento al mismo de forma correcta y en tiempos mínimos.
 - **Guardias:** La posibilidad de configurar grupos de guardias y guardias secundarias.
 - **Estalación:** La posibilidad de escalar un problema a un superior si el problema no se resuelve por un periodo prolongado de tiempo.
 - **Opsgenie:** es la herramienta que cuenta con todo lo necesario para brindar este servicio, con una aplicación móvil y la posibilidad de notificar a los interesados usando: Push Notifications, SMS, e-mail y llamados telefónicos.



- **Capacidad de diagnosticar:** Es la capacidad de establecer las causas del problema una vez identificada la existencia de alguno.
 - **Log de errores:** Visualizar la lista de errores cuantificados por tipo.
 - **Threads profiler:** Poder ver cuanta CPU o tiempo de ejecución/espera consume cada parte de la aplicación.
 - **Diferenciación de tiempos:** Entender cuanto tiempo insume cada capa de la aplicación, la base de datos, los caches, el render etc.
 - **Memoria:** Entender el consumo de memoria de la aplicación para detectar posibles memory leaks.

ACTA DE PROYECTO – CloudFleet	
Versión 1	1/4/2016

- **Capacidad de accionar:**

- **Rollback de versiones de software:** Es imperativo que el proceso de despliegue de versiones de la aplicación sea muy ágil y rápido así se puede volver atrás una versión en caso de subir una versión a producción que contenga errores.
- **Bash:** La posibilidad de que el equipo de SysAdmins pueda acceder al entorno productivo usando SSH para poder hacer troubleshooting avanzado.
- **Amazon Run Command:** Amazon Run Command permitirá accionar sobre los servidores de forma masiva en caso de caídas para poder resolver rápidamente los problemas.

ACTA DE PROYECTO – CloudFleet

Versión 1

1/4/2016

Recuperación ante desastres:

La capacidad para recuperarse ante desastres es ese conjunto de estrategias, arquitecturas y procesos para lograr la continuidad del negocio ante fenómenos externos de fuerza mayor pero de bajísima probabilidad. No se espera de estas acciones que sean inmediatas, pero si que recuperen el funcionamiento de la plataforma en un término razonable (<24Hs) en otra región del mundo que no este afectada por el conflicto.

Ejemplos de este tipo de incidentes son:

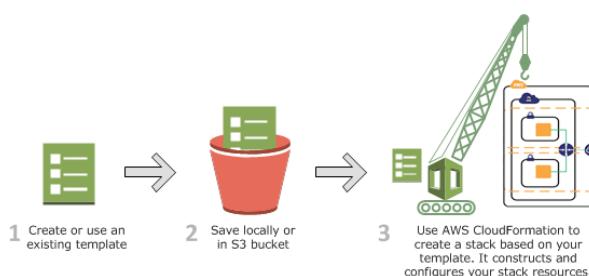
- Inundaciones
- Terremotos
- Guerras

Para esto, AWS cuenta con las siguientes regiones alrededor del mundo:



Estos serán los elementos de arquitectura y tecnología que nos permitirán migrar a otra región en un tiempo razonable:

- **Webservers:** Ante la perdida total de los 4 datacenters en una región, se podrá
 - **CloudFormation:** Cloud Formation es una tecnología de Amazon Web Services que permite documentar en configuración un stack completo de hardware con sus interacciones. Ejemplo: Subredes, Balanceadores, Servidores, Grupos de autoescalamiento etc. La infraestructura se creará con este mecanismo y por lo tanto será muy sencillo duplicar la infraestructura en otra región del mundo en caso de que por un evento de fuerza mayo sea necesario migrarla.

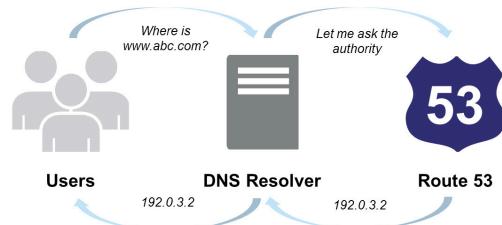


ACTA DE PROYECTO – CloudFleet

Versión 1

1/4/2016

- **Route 53 GTM (Global Traffic Manager):** Route 53 es el servicio de Amazon Web Services que, entre otras cosas, permite manejar el volumen de tráfico que llega a cada región mediante la utilización de DNS.



- **Bases de datos:** Habrá dos consideraciones arquitectónicas importantes enfocadas en lograr la continuidad de las bases de datos en caso de desastre.
 - **Snapshots a S3:** La base de datos hará snapshots a S3 con copias a otra región para poder recuperar fotos de la base de datos en el tiempo en caso de desastres.
 - **Slave Database:** Habrá una base de datos en modo slave en otra región que recibirá los cambios actualizados con una demora de unos pocos segundos. En caso de desastre, esa base de datos podrá convertirse en primaria.
- **BigData:** La capa de big data puede sufrir un período prolongado de indisponibilidad de cara a un desastre natural, sin afectar la continuidad del negocio. Tendremos suficiente tiempo para dedicarnos a reconstruir esta parte de la infraestructura. Aún así es preciso preservar los datos históricos para poder seguirlos usando una vez recuperados.
 - **S3:** Los datos son mantenidos en mas de una región en caso de que haya perdida de una región entera, podremos acceder rápidamente a los mismos desde la otra.

	ACTA DE PROYECTO – CloudFleet	
Versión 1		1/4/2016

Planificación del crecimiento y operatoria mínima viable:

Con el objetivo de poder trabajar en un presupuesto certero de costos consideramos el escenario de mínima que el negocio considera que enfrentaremos. Para el próximo año, tenemos la información mes a mes de:

- **Operaciones:** cuantos vehículos contarán con el dispositivo y cuantos usuarios estarán trackeando los mismos desde sus computadoras.
- **Estimación de Tráfico (GB):** Para cada una de las operaciones (trackear y monitorear) establecimos el volumen de tráfico que maneja, para poder establecer el costo de varios servicios, entre ellos el balanceo de carga, trafico de internet y almacenamiento de BigData etc.
- **Requests x Minuto (RPM):** El volumen de tráfico en requests es fácilmente calculable en función de las operaciones y nos servirá para calcular la cantidad de servidores mínima.
- **Servidores Web Reales:** Es el número de servidores que tendremos en cada instancia. Contempla la posibilidad de un crecimiento esporádico de tráfico y eventualmente errores.
- **Horas de instancia:** Es el número de horas de instancias virtuales que se usarán en cada mes. Es un número fácil de inferior a partir del número anterior y nos permitirá entender el volumen de utilización de servicios de monitoreo.

ACTA DE PROYECTO – CloudFleet

Versión 1

1/4/2016

Parametros de calculo	Valor
Tamaño del POST de vehículos	1024
Tamaño del refresco de mapa	5000
Vehículos x controlador	15

Mes	Vehículos	Controladores	Trafico Vehículos GB	Trafico Controladores GB	Requests x Min	#Servers (Max 2000RPM)	#Web Servers real	Instances Hr
1	100	6	4.12	7.24	136	0.68	2	1440
2	130	8	5.36	9.41	177	0.88	2	1440
3	169	10	6.96	12.24	230	1.15	5	3600
4	220	13	9.05	15.91	299	1.49	5	3600
5	286	17	11.77	20.68	388	1.94	5	3600
6	371	22	15.30	26.89	505	2.52	10	7200
7	483	29	19.89	34.96	656	3.28	10	7200
8	627	38	25.85	45.44	853	4.27	10	7200
9	816	49	33.61	59.07	1109	5.55	10	7200
10	1060	64	43.69	76.80	1442	7.21	15	10800
11	1379	83	56.80	99.84	1875	9.37	15	10800
12	1792	108	73.83	129.79	2437	12.19	15	10800
	Totales		306.22	538.27				
								74880

ACTA DE PROYECTO – CloudFleet

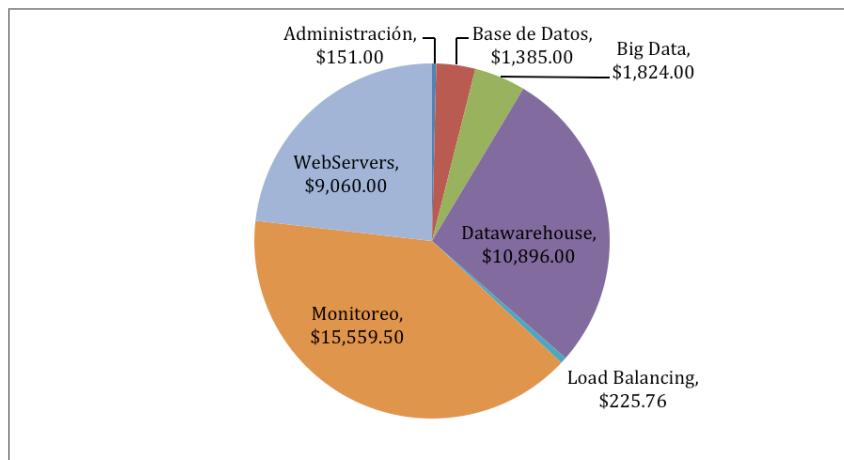
Versión 1

1/4/2016

Estimación de costos por un año:

Con los datos anteriores, estimamos el costo de los servicios de hosting relacionados para ese volumen de crecimiento estimado. El presupuesto se ha organizado por capas:

- **Load Balancing:** se utilizará el servicio Elastic Load Balancer de AWS para balancear el tráfico HTTPS. También se utilizará Route53 para el manejo de dominios de internet, pero su costo es despreciable. ELB tiene un costo por horas de uso y otro por tráfico, ambos están considerados.
- **WebServers:** Son los servidores que hostean la aplicación Web Reactiva y la API. Están ubicados en distintos datacenters para lograr alta disponibilidad. Es importante destacar que en este punto se consideraron las erogaciones por reservas a un año. Muchas de las cuales se harán casi sobre el final. En estos casos, queda un activo para el año entrante, donde no habrá que volver a pagar por esas reservas hasta que se venzan.
- **Administración:** Los servidores de administración son aquellos a los que nuestro equipo de Sysadmins se conectará para:
 - Desplegar nuevas versiones de software.
 - Efectuar operaciones de mantenimiento.
 - Hacer troubleshooting en caso de indisponibilidad del servicio.
- **Base de Datos:** La base de datos sera Aurora, una base de datos con protocol MySQL pero con posibilidades de cluster para escalabilidad y alta disponibilidad. Se contempla el uso de la instancia mas pequeña de la alternativa “Compute intensive” con discos SSD.
- **Monitoreo:** Se consideran dos servicios:
 - New Relic: Control de las variables aplicativas, depuración y troubleshooting.
 - OpsGenie: Gestión de incidentes, notificación a celulares, organización de guardias y escalación de tickets.
- **Big Data:** Considera, tanto el almacenamiento para la retención de datos a largo plazo en una plataforma durable y flexible para poder disponer: Simple Storage Service (AWS S3) así como la plataforma de explotación: Elastic Map Reduce (AWS EMR).
- **Datawarehouse:** AWS RedShift, un cluster de 5 nodos económicos y 4 licencias de Tableau Desktop.



(*)Gastos por rubro (USD)

ACTA DE PROYECTO – CloudFleet

Versión 1

1/4/2016

Capa	Servicio	Cantidad	Unidad	Precio unidad	USD	Link:
Load Balancing	Elastic Load Balancing - Hours	8760	Elastic Load Balancers Hs	\$ 0.03	\$ 219.00	https://aws.amazon.com/elasticloadbalancing/pricing/
Load Balancing	Elastic Load Balancing - GB of Data processed	845 GB		\$ 0.01	\$ 6.76	https://aws.amazon.com/elasticloadbalancing/pricing/
WebServers	t2.large instance 1 year up front	15 Instances		\$ 604.00	\$ 9,060.00	https://aws.amazon.com/ec2/pricing/
Administración	t2.small instance 1 year up front	1 Instances		\$ 151.00	\$ 151.00	https://aws.amazon.com/ec2/pricing/
Base de Datos	Relational Database Service - Aurora	1 Databases		\$ 1,385.00	\$ 1,385.00	https://aws.amazon.com/rds/aurora/pricing/
Monitoreo	New Relic	75000 Instances Hr		\$ 0.20	\$ 15,000.00	https://newrelic.com/application-monitoring/pricing
Monitoreo	OpsGenie	5 Users		\$ 24.00	\$ 120.00	https://www.opsgenie.com/pricing
Monitoreo	AWS CloudWatch Detailed 1 min	105 Instance Month		\$ 3.50	\$ 367.50	https://aws.amazon.com/cloudwatch/pricing/
Monitoreo	AWS CloudWatch Dashboards	24 Dashboards Month		\$ 3.00	\$ 72.00	https://aws.amazon.com/cloudwatch/pricing/
Big Data	Amazon Elastic Map Reduce Instances	24000 Instance Hr (m4.large)		\$ 0.07	\$ 1,680.00	https://aws.amazon.com/elasticmapreduce/pricing/
Big Data	S3 (Simple Storage System)	4800 GB/month		\$ 0.03	\$ 144.00	https://aws.amazon.com/s3/pricing/
Datawarehouse	Tableau Desktop (Licencia)	4 Licencias		\$ 999.00	\$ 3,996.00	
Datawarehouse	Amazon Redshift	5 Instances/yr		\$ 1,380.00	\$ 6,900.00	https://aws.amazon.com/redshift/pricing/
		Total (USD)	\$ 39,101.26			
		Total (ARS)	\$ 554,339.15			

	ACTA DE PROYECTO – CloudFleet	
Versión 1	1/4/2016	

Riesgos Identificados

Se realizó un análisis para identificar, catalogar y priorizar los riesgos más significativos que enfrenta el negocio.

En primer lugar se identificaron los riesgos en cuatro diferentes frentes:

- **Tecnológico:** Qué desafíos podría enfrentar la organización por futuras disruptiones tecnológicas. Ya sea por cambios en las tecnologías que afecten al negocio así como competidores con una gran ventaja en este sentido.
- **Legal:** Qué desafíos legales puede tener la organización en términos legales, de regulación y de gobierno.
- **Dependencias comerciales & tecnológicas:** Qué problemas puede tener CloudFleet por su lugar en la cadena de valor. Sus dependencias de proveedores y continuidad de contratos.
- **Importaciones & Pago a proveedores:** Todos aquellos desafíos relacionados con el comercio internacional, la regulación del mismo y las esperables variaciones de contexto.

Luego, una vez identificados los riesgos, se trabajó en identificar cual es la principal afectación de cada riesgo. Las afectaciones posibles son:

- **Mercado:** El riesgo identificado puede afectar significativamente la participación de CloudFleet en el mercado que compite.
- **Financiero:** El riesgo pone en riesgo la situación financiera de CloudFleet, haciendo necesaria una replanificación profunda, cambio de estrategia, búsqueda de préstamos o socios capitalistas.
- **Continuidad:** El riesgo en cuestión podría afectar la continuidad del negocio, una parte de él o su crecimiento por un período prolongado de tiempo.

Una vez identificadas las afectaciones se clasificará la Probabilidad de que tal suceso ocurra, luego la magnitud esperada del daño si el mismo se produce y por último el horizonte de tiempo donde se espera que el suceso pueda tener lugar.

ACTA DE PROYECTO – CloudFleet

Versión 1

1/4/2016

Tipo de Riesgo	Riesgo	Afectación	Descripción	Probabilidad	Impacto	Horizonte
Tecnológico	Nuevas Tecnologías de manejo autónomo	Mercado	La aparición de nuevas tecnologías de conducción asistida o automática restan valor al concepto de controlar al conductor.	Alto	Alto	Lejano
	Interés de las grandes empresas de internet	Mercado	Las grandes empresas de Internet; Facebook/Google, cuentan con productos relacionados, si estuvieran interesadas podrían competir con mucha ventaja.	Baja	Alto	Medio
	Desconocimiento de compromiso	Mercado	La posibilidad legal real de los proveedores de servicios de desconocer algún compromiso. Sea este un seguro o un servicio, debido al incumplimiento de alguna política (zona de cobertura, velocidad de manejo etc) en el caso de que la única evidencia de tal incumplimiento sea la información provista por el dispositivo y los servidores de CloudFleet. Defender la validez legal de la información de nuestros sensores y nuestros servidores, así como la integridad de nuestra organización, son factores claves del éxito de CloudFleet.	Moderada	Medio	Medio
Legal	Responsabilidad solidaria	Financiero	La posibilidad de que un tribunal entienda que el incumplimiento de un servicio contratado mediante el uso de la plataforma sea responsabilidad total o solidaria de CloudFleet, sentaría un antecedente que pondría en riesgo la continuidad del negocio y podría significar un gran costo para la compañía.	Baja	Alto	Medio
	Ausencia de marco regulatorio	Continuidad	La ausencia de un marco regulatorio para esta actividad (intermediador comercial para servicios del automóvil y para la gestión de servicios on demand y/o de tasação variable es un riesgo grande dado que si surgiera una regulación, la misma podría comprometer la continuidad del negocio.	Baja	Medio	Medio
Dependencias comerciales, tecnológicas	Proveedores de telecomunicaciones	Continuidad	La dificultad para acordar con los proveedores telecomunicaciones o un eventual aumento de los costos de las mismas ponen en riesgo la continuidad del negocio. Mucho mas teniendo en cuenta que los jugadores de telecomunicaciones móviles son pocos en todos los países de operación.	Moderada	Medio	Corto
	Proveedor de desarrollo de software	Financiero	Una vez contratado el proveedor de desarrollo de software podía: A) Incumplir, tanto en tiempos de entrega como en calidad o completitud del software B) Adoptar una posición difícil y aumentar los costos o retrasar entregas por considerarnos un cliente cautivo.	Moderada	Medio	Corto
	Fallas del proveedor de cloud	Continuidad	El proveedor de cloud podría fallar dejando fuera de servicio la plataforma.	Baja	Medio	Medio
Importaciones y Pagos al exterior	Importación de aparatos de trackeo	Continuidad	Por dificultades comerciales en los distintos países de operación, y siendo estos económicas emergentes con un marco jurídico y político muchas veces impredecible, es apropiado considerar estos inconvenientes podrían dificultar seriamente la cadena de abastecimiento.	Baja	Medio	Lejano
	Pago a proveedores de cloud en el exterior	Continuidad	No es infrecuente que en los países de operación haya problemas cambiarios y dificultades para hacer giros al exterior. Al menos dos de los países de operación (Argentina y Venezuela) tienen o tuvieron restricciones de este orden en los últimos 2 años.	Moderada	Alto	Medio

ACTA DE PROYECTO – CloudFleet

Versión 1

1/4/2016

Restricciones

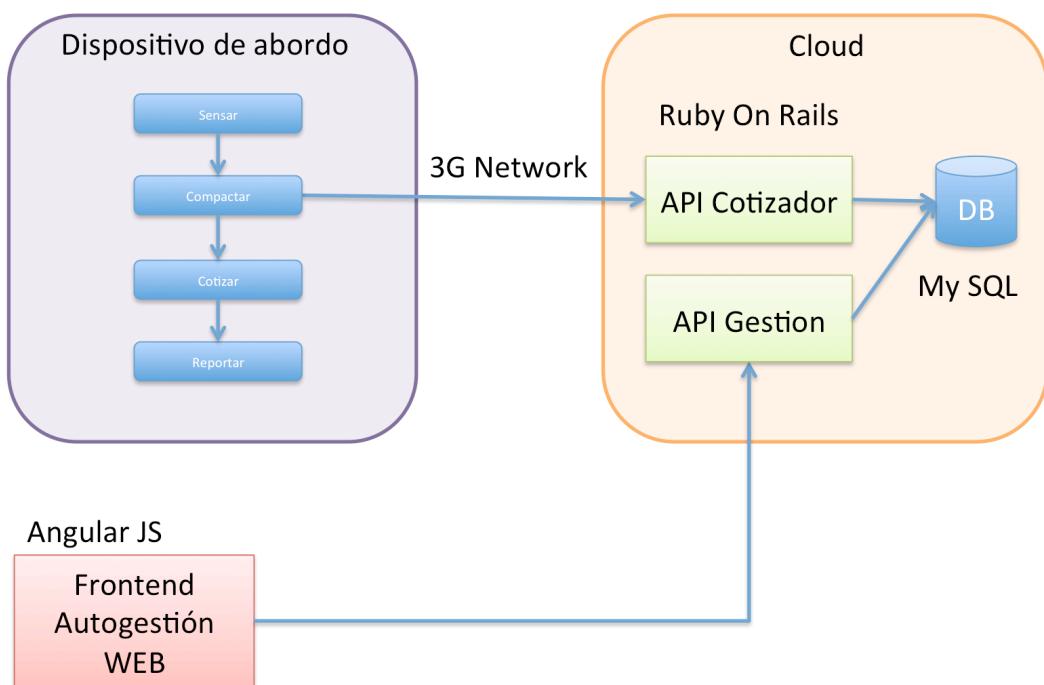
Este apartado detalla aquellas funcionalidades que no serán implementadas para la entrega de este trabajo. Sin embargo, muchas de ellas fueron analizadas en detalle, aquí se dejan también estas referencias para el trabajo futuro.

Queda excluido en la presente entrega:

- El desarrollo de la plataforma de servicios.
- El desarrollo de las integraciones con los prestadores de servicios.
- La unidad de abordo para el trackeo.
- El software de la unidad de abordo, solo se incluye el diseño y descripción pero no la implementación.

Funcionalidades excluidas: Dispositivo de abordo

El dispositivo de abordo queda fuera del alcance de implementación de este trabajo, pero se ha analizado su viabilidad y profundizado sobre su diseño técnico tanto en terminus de hardware como de software.



Componentes de Hardware:

Raspberry PI: B Plus

La microcomputadora Raspberry PI correrá el software Java encargado de:

- 1) Recibir la información de: OBDII del vehículo, el GPS, Acelerómetro y Cámara.
- 2) Enviar información de control a los sistemas en el cloud.
- 3) Informar los costos al usuario simil Taxi usando la pantalla LCD.
- 4) Informar de malas prácticas de manejo o zonas consideradas peligrosas.

	ACTA DE PROYECTO – CloudFleet	
Versión 1		1/4/2016



OBDII Bluetooth interface: ELM 327

Esta interfaz va conectada al vehículo. Todos los vehículos fabricados después de 1997 cuentan con este conector y permitirá al sistema monitorear los aspectos relacionados con la conducción (aceleración, velocidad, frenadas, giros) así como el estado general del vehículo: frenos, emisiones, alertas de la computadora, último service.



Bluetooth USB dongle: Standard

El dongle Bluetooth se encargará de hacer las lecturas del ELM 327 mencionado anteriormente.



Raspberry GPS Microstack:

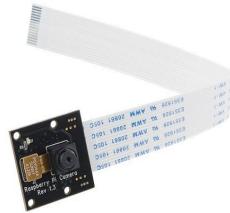
El GPS será leído por el software a bordo del vehículo y permitirá controlar las zonas por donde se mueve el vehículo, en qué horarios y tiempo de permanencia. Así como un control adicional de la velocidad y aceleraciones.

	ACTA DE PROYECTO – CloudFleet	
Versión 1	1/4/2016	



Raspberry PI Infrared Camera Module:

El sistema de abordo registrará a modo caja negra el video de caras al frente del vehículo, lo cual servirá para determinar, en muchos casos, las responsabilidades en caso de accidente.



Raspberry PI Touch Screen:

La pantalla informará al usuario si la cobertura esta en línea, del costo de la misma y sobre aquellos excesos de manejo o lugares de estadía que hacen que su cobertura suba de precio.

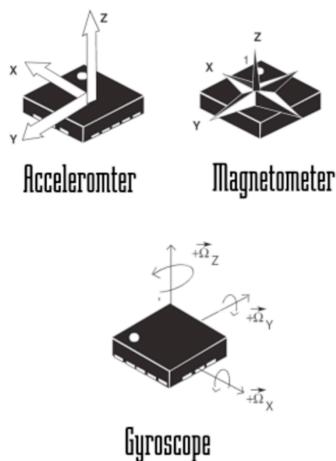


USB 3G Dongle.

Será el servicio de comunicaciones mediante el cual se enviará la información de costos y firmas de validación a los servidores.

Berry IMU

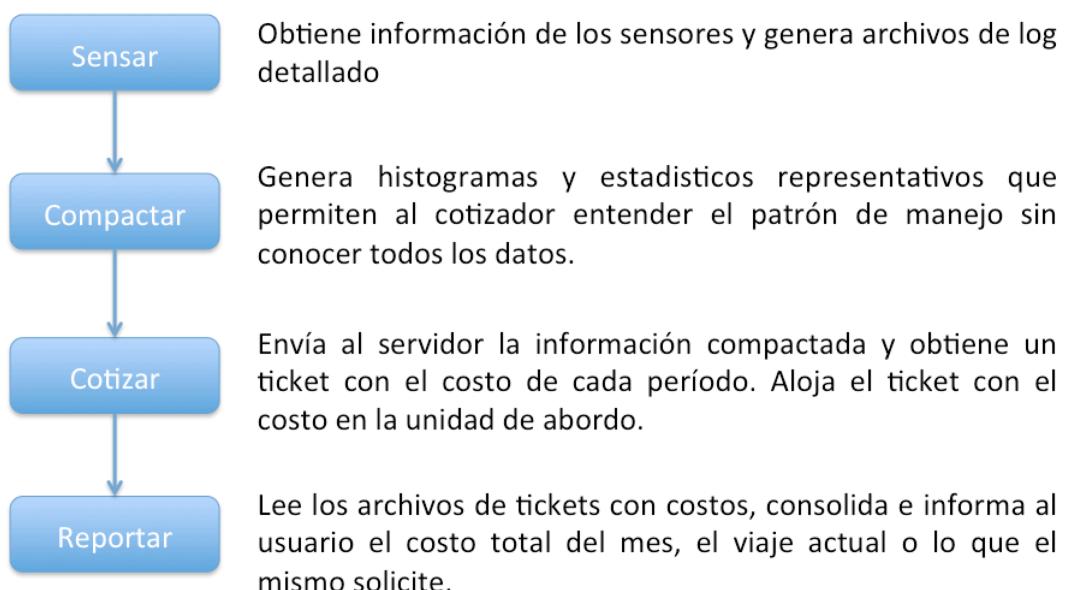
El acelerómetro servirá para detectar malas maniobras, excesos y además identificar siniestros.

**16Gb SD Card.**

Será donde se almacenarán los fragmentos de video anteriores a un siniestro. El video viejo se irá borrando pero aquellos momentos asociados con bruscas desaceleraciones serán guardados de forma permanente y un MD5 de la filmación será enviado a los servidores.

Software del modulo de abordo:

El sistema contará con dos entornos, uno con todos los sistemas que funcionan en el dispositivo de abordo que funciona en un Raspberry Pi dentro del automóvil y el otro que es un servidor en el Cloud que funciona recepcionando y cotizando tickets. En este mismo servidor también se hosteará una API mediante la cual se podrá consultar el consumo actualizado.

Módulos:

ACTA DE PROYECTO – CloudFleet

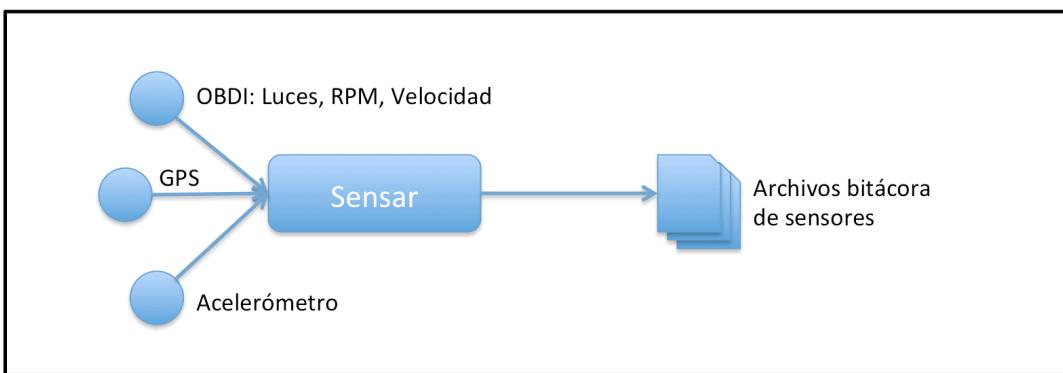
Versión 1

1/4/2016

Módulo sensar:

Propósito:

Leer la información del GPS, el acelerómetro y los parámetros de la interfaz OBDI y guardarlos en archivos de datos.



Salidas:

\$RTI_HOME/sensores/gps/AÑO/MES/DIA/HORA_MINUTO.data
\$RTI_HOME/sensores/acelerometro/AÑO/MES/DIA/HORA_MINUTO.data
\$RTI_HOME/sensores/obdi/AÑO/MES/DIA/HORA_MINUTO.data

Formato de las salidas:

El formato de los archivos será CSV y contendrá una fila por segundo con el siguiente formato.

```
01, 0.2g
02, 0.1g
03, 0.0g
...
59, 0.1g
```

Ejemplo para el acelerómetro:

\$RTI_HOME/sensores/acelerometro/2016/03/27/16_25.data

ACTA DE PROYECTO – CloudFleet

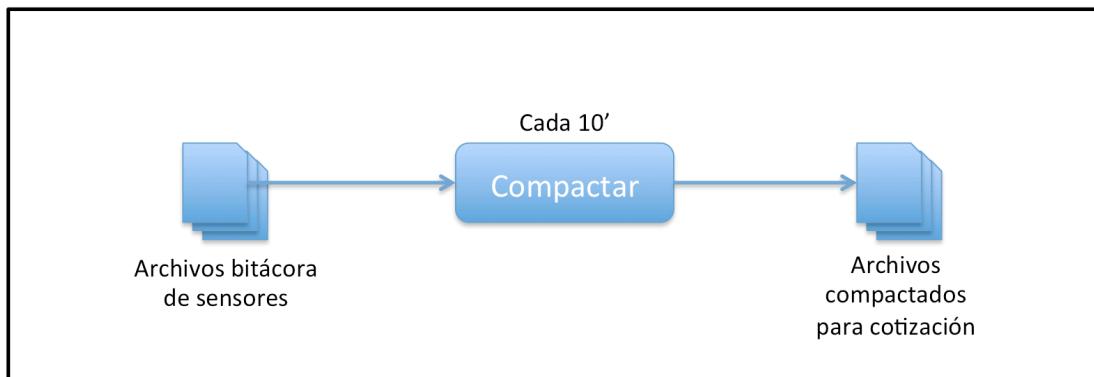
Versión 1

1/4/2016

Módulo compactar:

Propósito:

Generar un informe compacto de la situación de los sensores sobre un periodo de cotización (10 minutos) para ser enviada al cotizador y obtener el precio de la póliza en el período.



Funcionamiento:

El proceso correrá como un demonio, cada diez minutos se despertará y leerá los archivos en los subdirectorios de sensores correspondientes

Parámetros:

./compactar

Archivo de control:

\$RTI_HOME/status/compacted.date

Entradas:

Subdirectorios de \$RTI_HOME/sensores/

Salidas:

\$RTI_HOME/compactado/AÑO_MES_DÍA_HH24_MI.data

Formato

El formato del archivo de salida será JSON (Java Script Object Notation). El contenido será un histograma de los valores de cada entidad durante el período compactado (GPS, Acelerómetro y parámetros OBDI).

ACTA DE PROYECTO – CloudFleet**Versión 1****1/4/2016**

```
{  
    "fecha_desde": "2016-03-25 20:50:00",  
    "fecha_hasta": "2016-03-25 20:59:59",  
    "acelerometro": {  
        "unidad": "g",  
        "min": "0",  
        "max": "1.3",  
        "histo": {  
            "0.0-0.3": "40%",  
            "0.3-0.5": "30%",  
            "0.5-1.0": "10%",  
            "1.0-1.3": "10%",  
            "1.3-1.5": "10%"  
        }  
    },  
    "obdi": {  
        "velocidad": {  
            "unidad": "km/h",  
            "min": "0",  
            "max": "193",  
            "histo": {  
                "000-020": "10%",  
                "020-040": "10%",  
                "040-060": "10%",  
                "060-080": "5%",  
                "080-100": "5%",  
                "100-120": "10%",  
                "120-140": "20%",  
                "140-300": "50%"  
            }  
        },  
        "rpm": {  
            "unidad": "rpm",  
            "histo": {  
                "0000-1000": "5%",  
                "1000-2000": "10%",  
                "2000-3000": "15%",  
                "4000-5000": "30%",  
                "5000-6000": "40%"  
            }  
        },  
        "luces": {  
            "histo": {  
                "ninguna": "0%",  
                "posición": "100%",  
                "bajas": "100%",  
                "altas": "60%"  
            }  
        },  
        "gps": {  
            "BSAS-CAMPANA01": "100%"  
        }  
    }  
}
```

Ej: \$RTI_HOME/compactado/compactado-20160325_2050.data

ACTA DE PROYECTO – CloudFleet

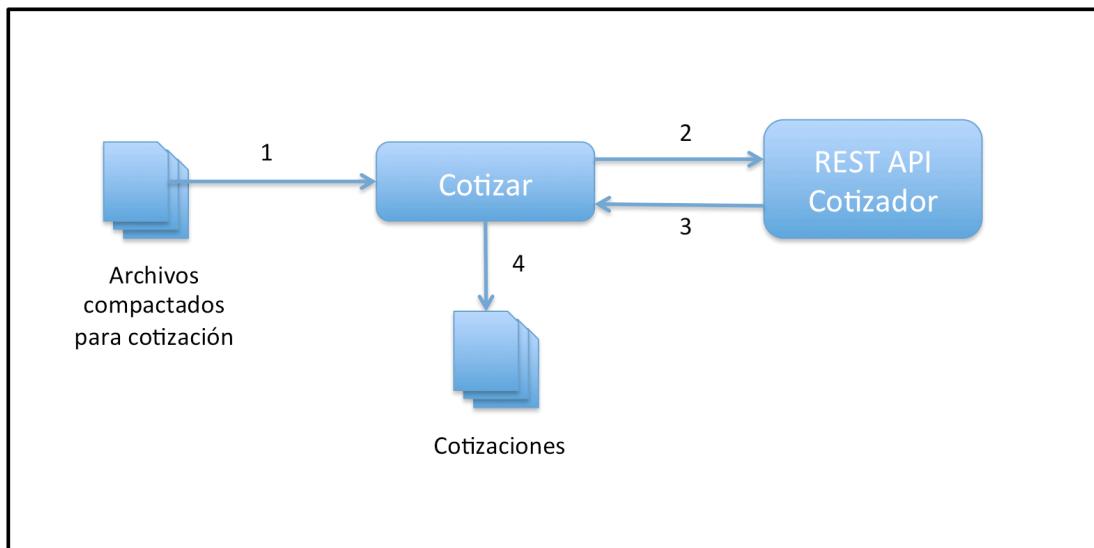
Versión 1

1/4/2016

Módulo cotizar:

Propósito:

El propósito del módulo de cotización es enviar los archivos compactados de los sensores por el proceso de compactación al servidor de cotización y almacenar el resultado.



Entradas:

`$RTI_HOME/compactado/AÑO_MES_DÍA_HH24_MI.data`

Salidas

`$RTI_HOME/cotizado/AÑO_MES_DÍA_HH24_MI.data`

ACTA DE PROYECTO – CloudFleet

Versión 1

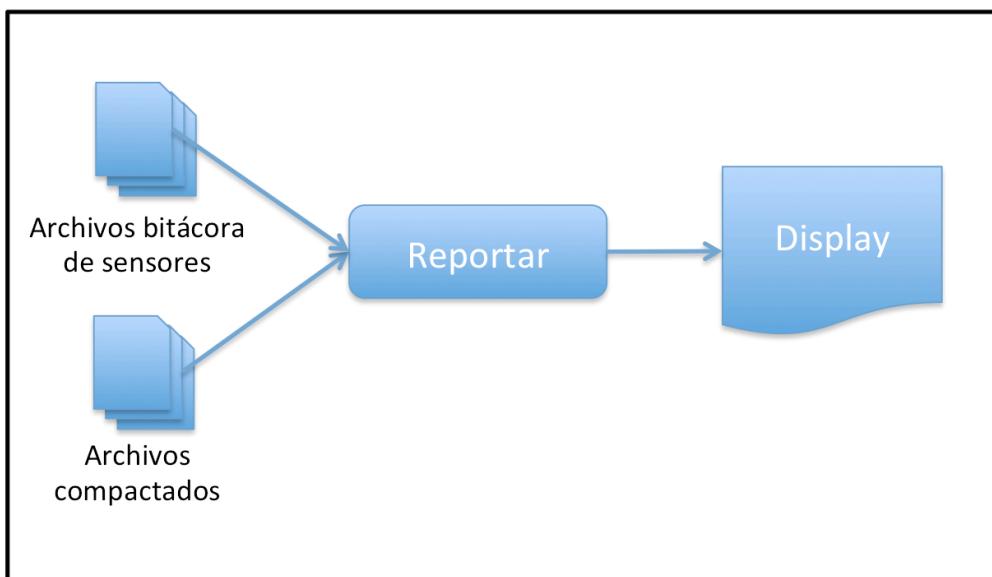
1/4/2016

Módulo reportar

Propósito

El módulo reportar es el encargado de mostrar por pantalla la información referente a los consumos y los parámetros relevantes de conducción.

Este módulo no tiene una salida por archivos, dado que reporta directamente a la pantalla del dispositivo.



Entradas:

\$RTI_HOME/cotizado/AÑO MES DÍA HH24 MI.data
\$RTI_HOME/sensores/gps/AÑO/MES/DIA/HORA_MINUTO.data
\$RTI_HOME/sensores/accelerometro/AÑO/MES/DIA/HORA_MINUTO.data
\$RTI_HOME/sensores/obdi/AÑO/MES/DIA/HORA_MINUTO.data

	ACTA DE PROYECTO – CloudFleet	
Versión 1		1/4/2016

Anexos

Código fuente:

- Código fuente del frontend de CloudFleet.
- Código fuente de la API de CloudFleet.
- Código fuente del generador de información geográfica (Ciudades, Provincias, Paises)
- Código fuente del generador de información de vehículos. (Marcas, Modelos, Versiones, Precios)
- Código fuente del simulador de trayectorias que utiliza google maps.
- Código fuente del simulador de empresa de logistica y alertas.

Gráficos:

- Diagrama de la arquitectura de despliegue de CloudFleet.
- Diagrama de entidades.
- Diagrama de clases.
- Diagrama de mocks de pantalla.

Planillas:

- Presupuestos.

Videos:

- Demostración de uso de CloudFleet con simulador de flotas.

ACTA DE PROYECTO – CloudFleet		
Versión 1	1/4/2016	

Aprobaciones:

Firma Sponsor 1
(ej: Presidente)

Fecha:_____

Firma Sponsor 2
(Ej. Vicepresidente)

Fecha:_____