DEPARTAMENTO DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico 2

12 de Noviembre de 2015 — Ingeniería de Software II

Integrante	LU	Correo electrónico
Lambrisca, Santiago	274/10	santiagolambrisca@gmail.com
Mancuso, Emiliano	597/07	emiliano.mancuso@gmail.com
Mataloni, Alejandro	706/07	amataloni@gmail.com
Reartes, Marisol	422/10	mreartes5@gmail.com

Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Ciudad Universitaria - (Pabellón I/Planta Baja)
Intendente Güiraldes 2160 - C1428EGA
Ciudad Autónoma de Buenos Aires - Rep. Argentina
Tel/Fax: (54 11) 4576-3359
http://www.fcen.uba.ar

Índice

1. Especificación de Atributos de Calidad

Escenario 1: Disponibilidad

Descripción: Ante eventualidades como pérdidas de información por catástrofes naturales, se desea que esta información se mantenga protegida y disponible en todo momento

- Fuente: Externa
- Estímulo: Se corta la luz en Mendoza
- Artefacto: Sistema
- Entorno: Normal
- Respuesta: Se redirigen los datos a procesar a otros nodos distribuidos que pueden completar el trabajo.
- Medición de respuesta: El 99,99 % de los casos la información se encuentra disponible. Además, los datos siguen estando disponible por los nodos replica.

Escenario 2: Disponibilidad

Descripción: Se busca que haya conectividad en todo momento para que los datos de GPS se puedan enviar al sistema

- Fuente: Externa
- Estímulo: Se detecta baja conectividad
- Artefacto: Sistema
- Entorno: Normal
- Respuesta: Se notifica a los administradores del sistema y a un sistema alternativo para que provea conectividad.
- Medición de respuesta: En menos de 12 horas llega el dron al lugar de baja conectividad.

Escenario 3: Disponibilidad

Descripción: Cuando no hay buena conectividad se desea que los datos medidos por el GPS no se pierdan

- Fuente: Externa
- Estímulo: No hay conectividad
- Artefacto: Sistema de Controlador vehicular
- Entorno: Normal
- Respuesta: Se guardan los datos localmente, por 24 horas, hasta que vuelva a haber conectividad.
- Medición de respuesta: El 99,999 % de los datos no se pierden

Escenario 4: Performance

Descripción: Se desea que el sistema ande muy rápido soportando el gran volumen de datos de todos los autos registrados del país.

- Fuente: Externa
- Estímulo: Llegan 10000 mediciones hechas por GPS
- Artefacto: Sistema
- Entorno: Normal
- Respuesta: Los datos son procesados y se generan las infracciones correspondientes
- Medición de respuesta: En a lo sumo 5 segundos las infracciones son generadas satisfactoriamente.

Escenario 5: Performance

Descripción: Los datos deben poder ser visualizados en un mapa en tiempo real, con el menor delay posible.

- Fuente: Externa
- Estímulo: Se genera una medición en un vehículo que implica una infracción de exceso de velocidad
- Artefacto: Sistema
- Entorno: Normal
- Respuesta: Se procesa la medición y se persiste la infracción.
- Medición de respuesta: La información de la infracción se puede ver en el mapa en menos de 2 hs

Escenario 6: Seguridad

Descripción: Es importante que el sistema esté protegido frente ataques externos

- Fuente: Individuo no identificado
- Estímulo: Envío de mediciones fraudulentas
- Artefacto: Sistema
- Entorno: Normal
- Respuesta: Se detecta que no proviene de una fuente confiable, se descarta y se loguea el ataque.
- Medición de respuesta: El 99,9 % de los casos se detecta el ataque satisfactoriamente.

Escenario 7: Seguridad

Descripción: El acceso a datos está restringido a los roles de cada usuario.

- Fuente: Individuo identificado (empleado de Drones SA)
- Estímulo: Consulta las mediciones recolectadas
- Artefacto: Sistema
- Entorno: Normal
- Respuesta: Se brindan los datos a través de una interfaz web.
- Medición de respuesta: Los datos presentados corresponden al nivel de acceso que tiene el usuario autenticado.

Escenario 8: Seguridad

Descripción: La sensibilidad de la información acumulada por nuestro sistema debe asegurarse de tal forma que permita auditar los movimientos en caso de ser pedido por la Defensoría del Pueblo.

- Fuente: Agente de la Defensoría del Pueblo
- Estímulo: Desea auditar los movimientos de un conductor
- Artefacto: Sistema
- Entorno: Normal
- Respuesta: Se muestra el registro detallado del último mes de actividad del conductor
- Medición de respuesta: En el 99,99 % de los casos la información está disponible.

Escenario 9: Modificabilidad

Descripción: Se pretende la incorporación de nuevos tipos de infracciones.

- Fuente: Equipo de desarrollo
- Estímulo: Se desea agregar un nuevo tipo de infracción
- Artefacto: Sistema
- Entorno: En ejecución
- Respuesta: Se agrega un el nuevo tipo de infracción sin alterar otras funcionalidades
- Medición de respuesta: Se invierten menos de 8 horas hombre

Escenario 10: Usabilidad

Descripción: Los usuarios deben tener una herramienta para poder acceder al historial de infracciones

- Fuente: Externa
- Estímulo: Un conductor pide revisar su actividad
- Artefacto: Interfaz web para conductores
- Entorno: En diseño
- Respuesta: Se accede al repositorio de auditoría local y se muestran los datos solicitados al interesado
- Medición de respuesta: De un total de 1000 usuarios encuestados (con nieveles de satisfacción entre 1 y 10), el 95 % de los usuarios estuvo satisfecho con nivel mayor o igual a 8.

2. Arquitectura del TP 2

Para empezar, se presenta la arquitectura correspondiente al Tp2.

2.1. General

Tenemos dos formas de observar nuestra arquitectura, la primera con un enfoque general donde se ven los componentes externos a nuestro sistema que interactúan con nosotros. Con esta visión, nuestra arquitectura se ve como un único componente monolítico pues es más sencillo a este nivel, sin embargo, contamos con una arquitectura distribuida a lo largo y ancho del país que podremos ver con más detalle en las siguientes vistas.

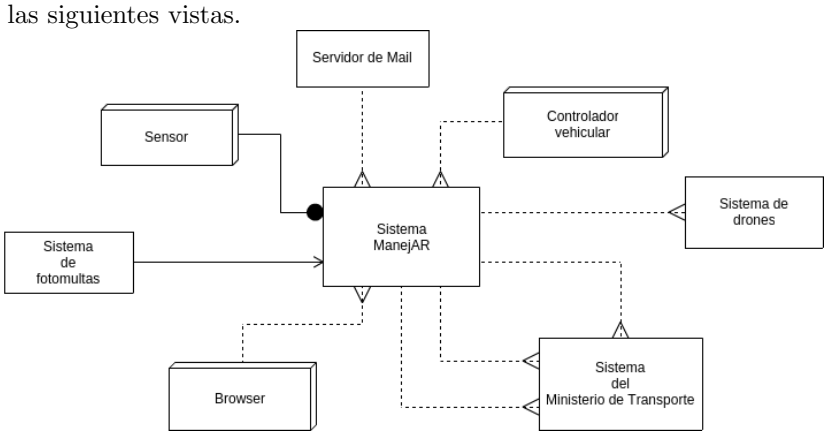

Figura 1: Arquitectura general

Como podemos ver, son varios los componentes externos que interactúan con nuestro sistema, cada uno con sus particularidades que pasaremos a detallar.

2.2. Sensor

Los **sensores** son nuestros dispositivos ubicados a lo largo del país, sobre todo en La Pampa, que utilizamos para medir la conectividad de la zona. En estos dispositivos, distinguimos dos componentes claves para describir la arquitectura.

- Receptor de mediciones de conectividad: Recolecta las mediciones del hardware y las encola en el próximo componente.
- Comunicador con administrador de sensores: Toma las mediciones y las envía al Administrador de sensores de nuestro sistema **ManejAR**.

La particularidad de la conexión entre este último componente y el sistema manejar es que el medio tiene pérdida de paquetes pero decidimos no crear una estrategia de reenvío pues enviamos esas mediciones cada 5 segundos.

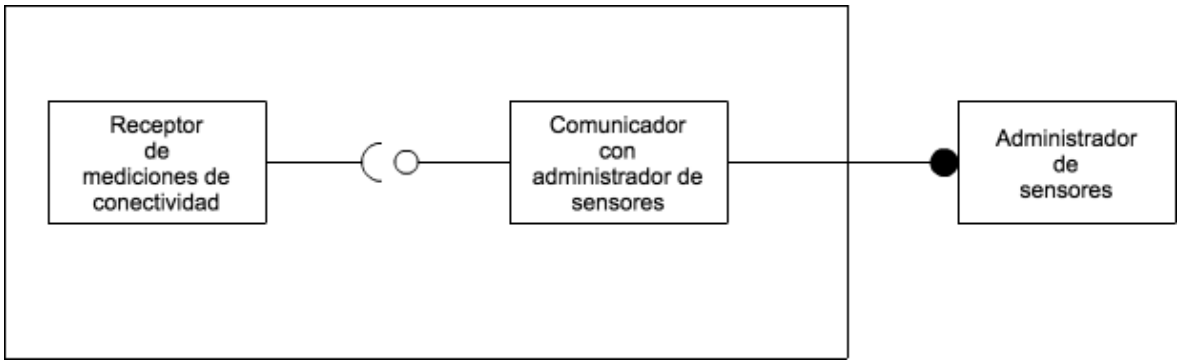

Figura 2: Arquitectura interna de los Sensores

2.3. Control Vehicular

El **Controlador Vehicular**, es el componente que se encuentra en cada uno de los vehículos que deben ser monitoreados. En este componente encontramos los siguientes subcomponentes.

- **Receptor de mediciones**: Este se encarga de recibir las mediciones realizadas por el GPS de su vehículo, y las guarda en un repositorio de **Mediciones**.
- **Mediciones**: Es el repositorio donde se almacenan las mediciones del GPS. Es importante la presencia de este componente de almacenamiento, porque nos permite preservar las mediciones hasta que hayan sido envíadas, fundamental para no perder mediciones ante un problema de conectividad. Este componente impacta en el *Escenario 3*.
- **Transmitidor**: Encargado de asegurarse del envío de las mediciones al **Balanceador** del Sistema Manejar. Antes de enviar cualquier dato, se comunica con el **Encriptador** y el **Checksum** para proteger la integridad y confidencialidad de los datos (*Escenario 6*). Cada medición enviada exitosamente sera borrada del repositorio **Mediciones**. En caso de no disponer de conectividad, se reintentara el envío de la medición cuando logre conectarse. De esta manera nos aseguramos no perder mediciones, y contar con todas las mediciones generadas.

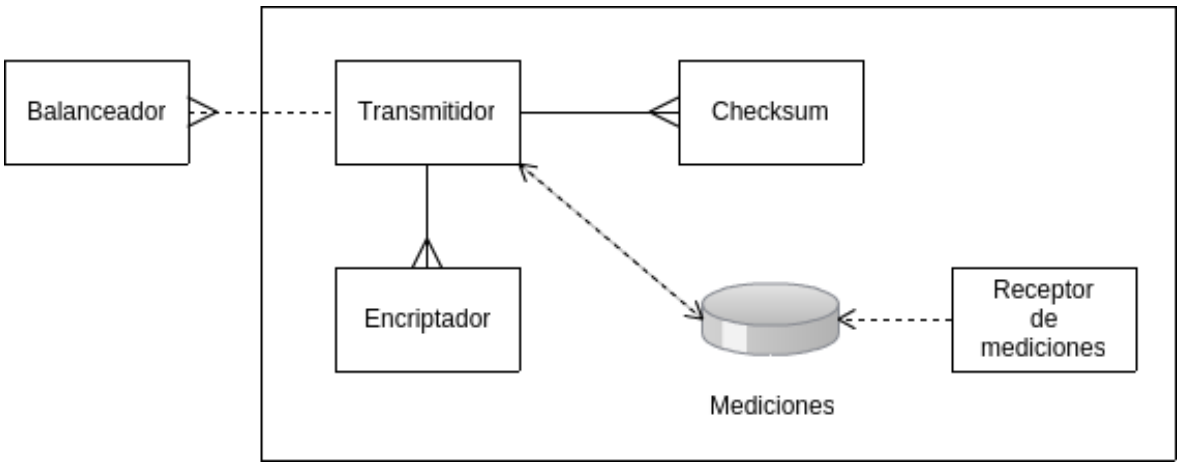

Figura 3: Arquitectura interna de los Controladores vehiculares

2.4. Sistema Manejar

Si nos concentramos en los componentes dentro del recuadro principal, podemos organizarlos en dos categorías:

- Esta en servidor central
- Distribuido en diferentes nodos a lo largo del pais.

El *Controlador de datos de GPS* y el *Administrador de Base de Datos* son los únicos que estan distribuidos en todos los nodos a lo largo del país, con el objetivo de optimizar el procesamiento de datos y a su vez, asegurar la disponibilidad del servicio ya sea del procesamiento como la redundancia de las bases de datos. Esto aporta para cumplir el *Escenario 4*

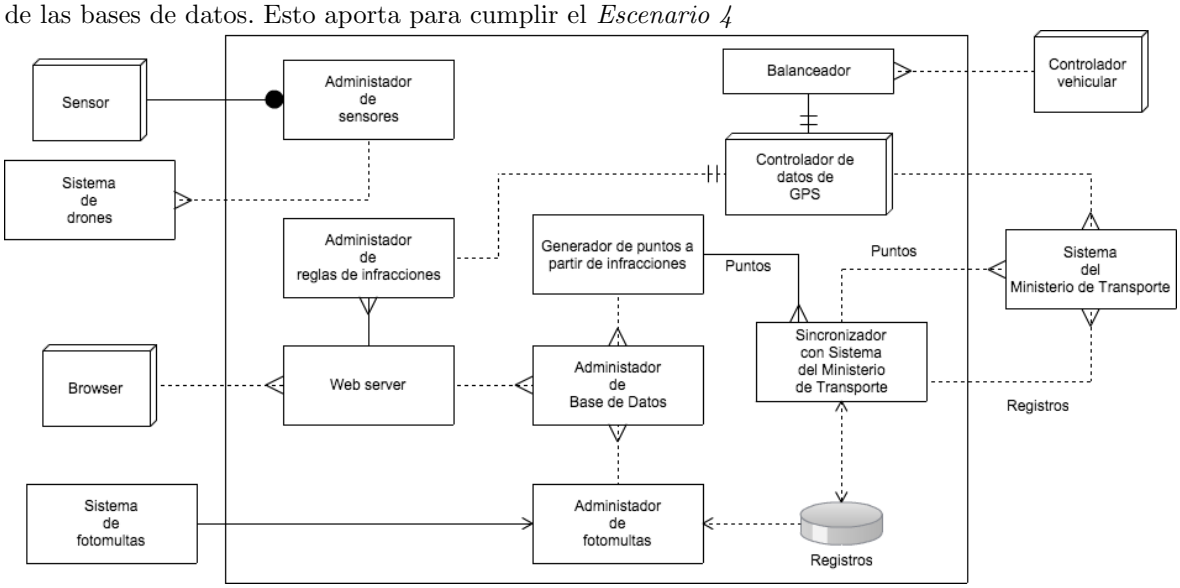

Figura 4: Arquitectura interna del Sistema ManejAR

La mayoría de estos componentes los explicaremos en detalle más adelante con su respectiva vista. Para los que no, alcanza la siguiente descripción de los mismos.

2.4.1. Administrador de reglas de infracciones

Es el encargado de recibir las modificaciones de las reglas de infracciones y notificarlas a todos los **Controladores de datos de GPS** usando un *broadcast*. Mediante este componente, se pueden agregar facilmente nuevos tipos de infracciones cumpliendo con el *Escenario 9*.

2.4.2. Generador de Puntos a partir de Infracciones

Todas las noches este componente toma las infracciones del **Administrador de Bases de Datos** y calcula el puntaje a restar a cada conductor. Luego se lo informa al **Sincronizador con Sistema del Ministerio de Transporte**.

2.4.3. Sincronizador con Sistema del Ministerio de Transporte

Las responsabilidades de este componente son dos. La primera es actualizar nuestra base de registros directamente del **Sistema del Ministerio de Transporte** para poder identificar los conductores profesionales de los particulares. La segunda envía los puntos a descontar a cada conductor profesional.

2.4.4. Balanceador

El balanceo de carga es un concepto usado que se refiere a la técnica usada para compartir el trabajo a realizar entre varios procesos, ordenadores, discos u otros recursos. El balanceo de carga se

mantiene gracias a un algoritmo que divide de la manera más equitativa posible el trabajo, para evitar los así denominados cuellos de botella y mantener una alta performance y disponibilidad del sistema. Con esto cumplimos el *Escenario 1, 4 y 5.*

2.5. Controlador de datos de GPS y Administrador de Bases de Datos

2.5.1. Controlador de datos de GPS

Recibe las mediciones directas del GPS, las desencripta y las publica en un *BUS* al cual estan subscriptos los siguientes componentes:

- Persistidor temporal para auditoría
- Persistidor de datos para empresa de drones
- Generador de estadísticas
- Generador de infracciones (Múltiples instancias)

Los primeros tres son triviales, pero el último merece dar una descripción más detallada.

Las múltiples instancias de este proceso, comparten un mismo *pipe* pero el primero que lo recibe lo remueve de la cola. De esta manera nos garantizamos que no se repita el procesamiento de los datos y poder paralelizar este proceso, pues la cantidad de datos en este *pipe* es inmensa. Cada una de estas instancias, deberá comunicarse con el Sistema del Ministerio de Transporte, para obtener los límites de velocidades. Además debe leer del repositorio de Reglas de infracciones, las reglas para poder generar las infracciones según corresponda.

Las infracciones generadas serán guardadas en un repositorio de Infracciones comunicándose con el *Administrador de Bases de Datos*. Este proceso nos garantiza cumplir con el *Escenario 4 y 5.*

Las *Reglas de infracciones* son actualizadas por medio de un *ABM*, que recibe las actualizaciones del *Administrador de reglas de infracciones.*

2.5.2. Administrador de Bases de Datos

Es el encargado de mantener los datos replicados en los distintos nodos, asegurándose de brindar una consistencia eventual entre ellos, con el objetivo de brindarnos performance y disponibilidad. Es su responsabilidad, hacer de intermediario entre quienes quieran leer o escribir de los siguientes repositorios:

- Auditoría: Guarda información para que los conductores puedan auditar sus multas dentro del período de 1 mes. (*Escenario 8 y 10*).
- Estadísticas: Almacena las estadísticas generadas por el **Generador de estadísticas**
- Drones: Guarda los datos que le serán brindados a la empresa de drones.
- Infracciones: Se guardan las infracciones generadas, que serán utilizadas para calcular los puntajes.

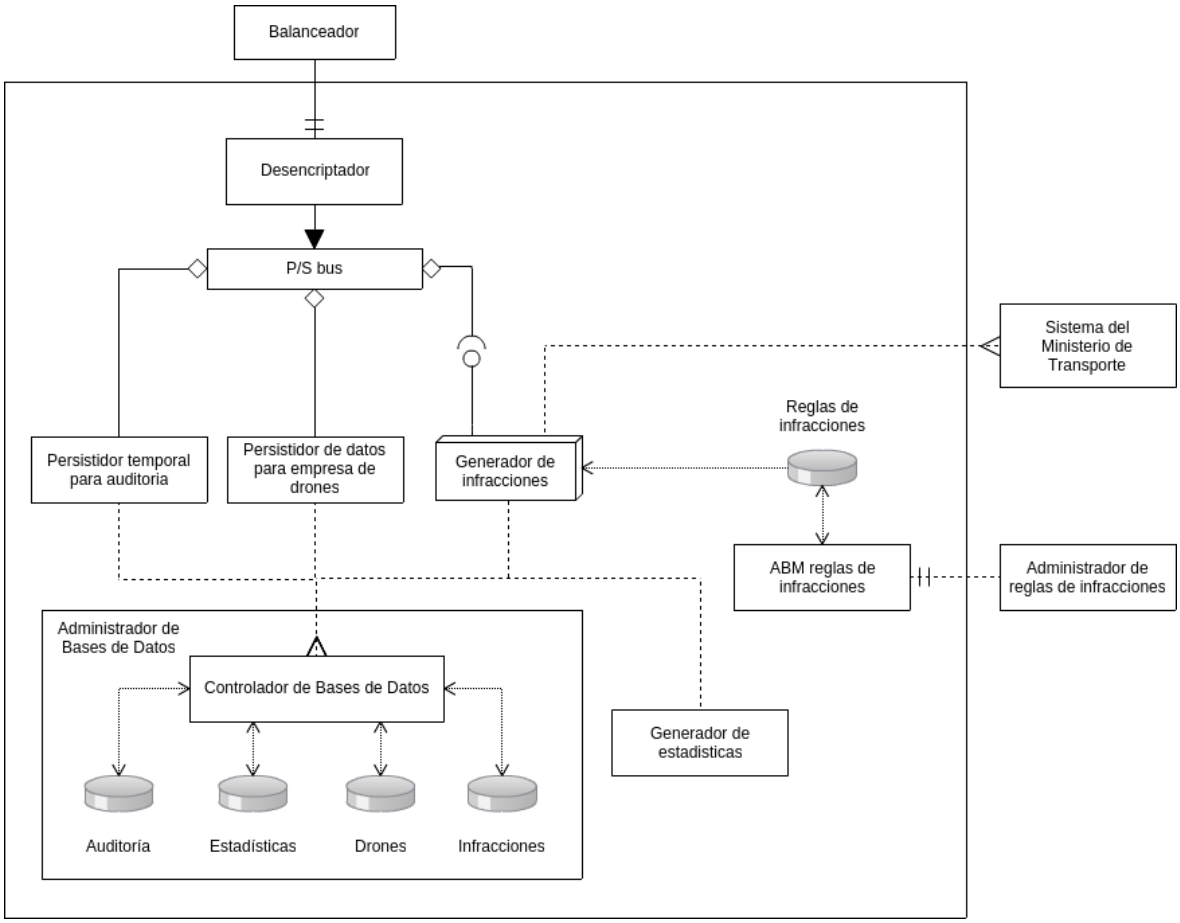

Figura 5: Arquitectura interna del Controlador de datos de GPS del Sistema ManejAR

2.6. Administrador de Fotomultas

Filtra las multas recibidas del Sistema de fotomultas, para almacenar solo las infracciones correspondientes a conductores profesionales. Este componente ya manda las infracciones generadas, no es necesario procesarlas, ślo se encarga de hacer el filtro explicado anteriormente.

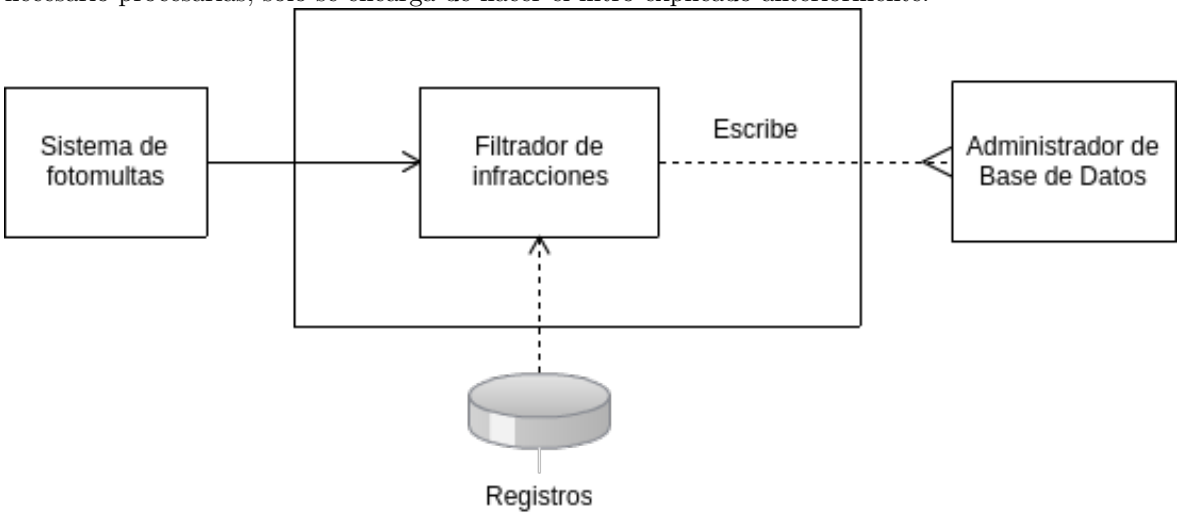

Figura 6: Arquitectura interna del Administrador de Fotomultas del Sistema ManejAR

2.7. Administrador de sensores

Este proceso, es el responsable de recibir las mediciones de los sensores, que primero seran monitoreadas por el **Monitor de sensores**. Éste controla que todos los sensores se esten comunicando periódicamente, utilizando la táctica de *heartbeat*. En caso de no recibir mensajes de alguno de los sensores, envía una alerta al **Comunicador con Sistema de Drones**, que accionará según corresponda. Al mismo tiempo, el **Monitor de Sensores**, encola las mediciones al **Procesador**, el cual las almacena en un repositorio y realiza un análisis de las mediciones en el tiempo para detectar baja conectividad y alertar al **Comunicador**. El **Comunicador** debera enviar un mail a los técnicos correspondientes, al **Sistema de drones** y además, guardar un log del uso de los drones para fines de auditoria. Estas medidas impactan en el *Escenario 2*.

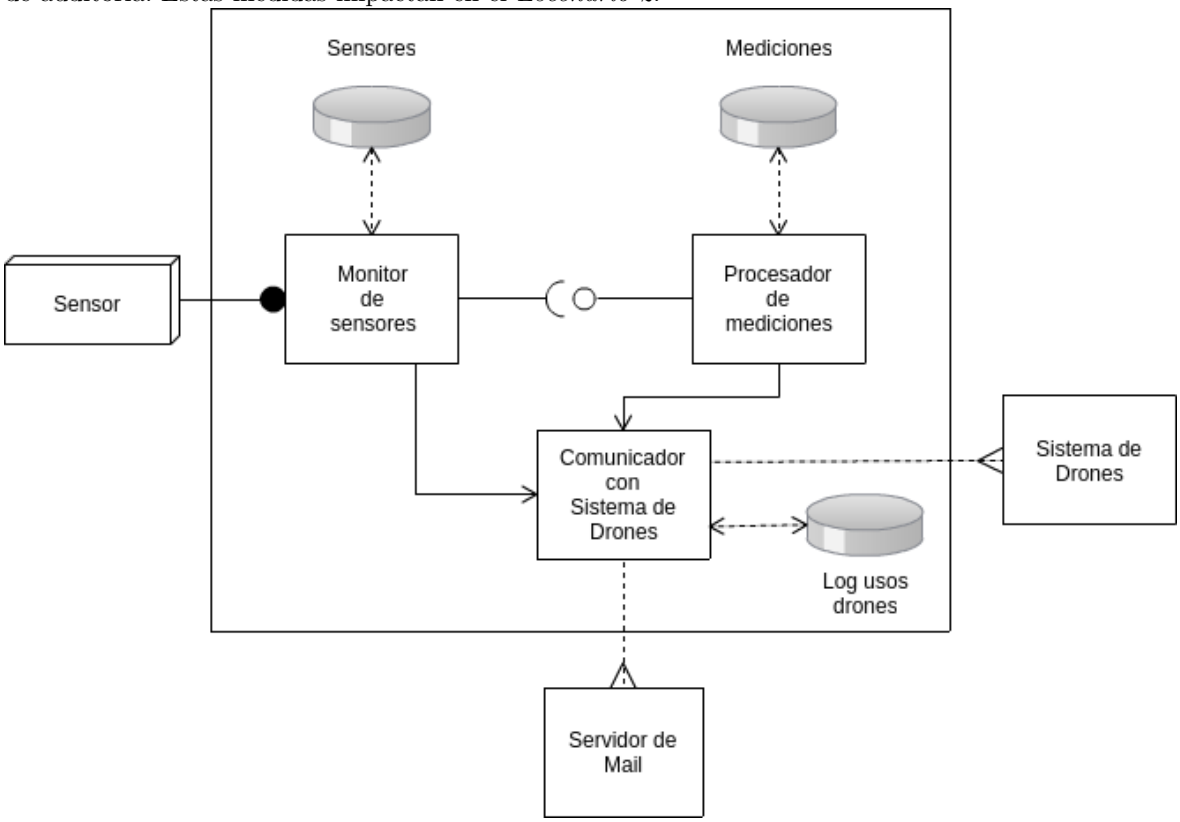

Figura 7: Arquitectura interna del Administrador de Sensores del Sistema ManejAR

2.8. Web Server

Funciona como una interfaz, para que los distintos usuarios puedan autenticarse y acceder a la informacion que les corresponda segun sus permisos, cumpliendo con el *Escenario 7, 8 y 10*.

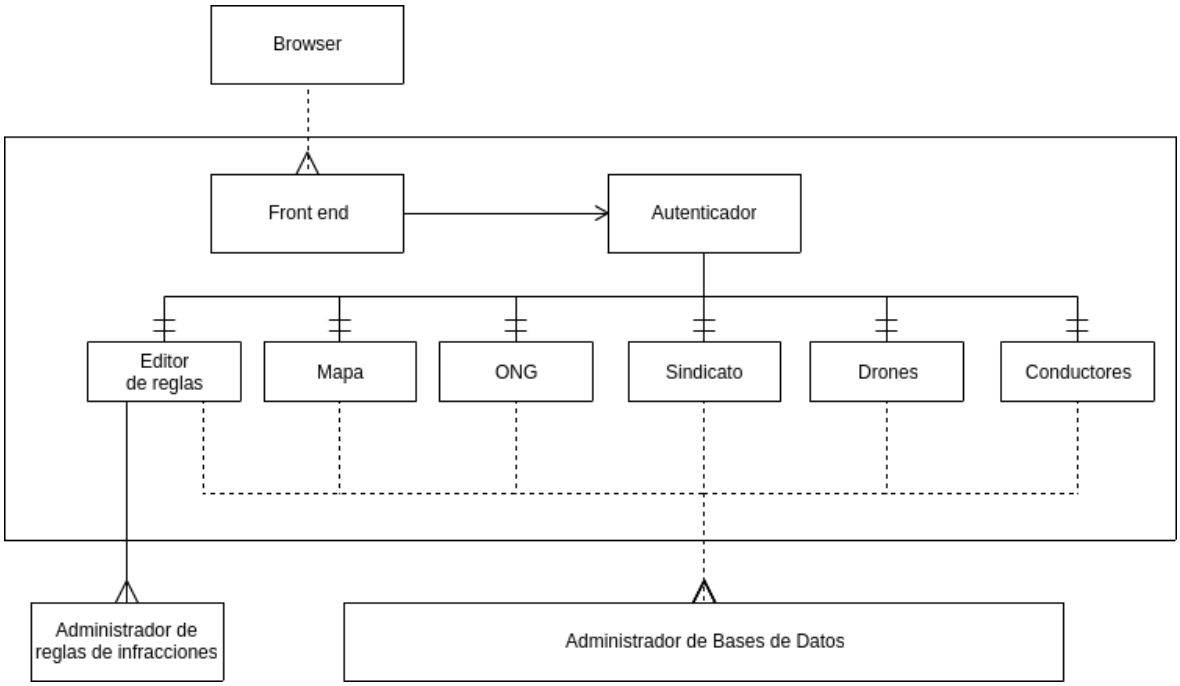

Figura 8: Arquitectura interna del Web Server del Sistema ManejAR

2.9. Referencias de los conectores

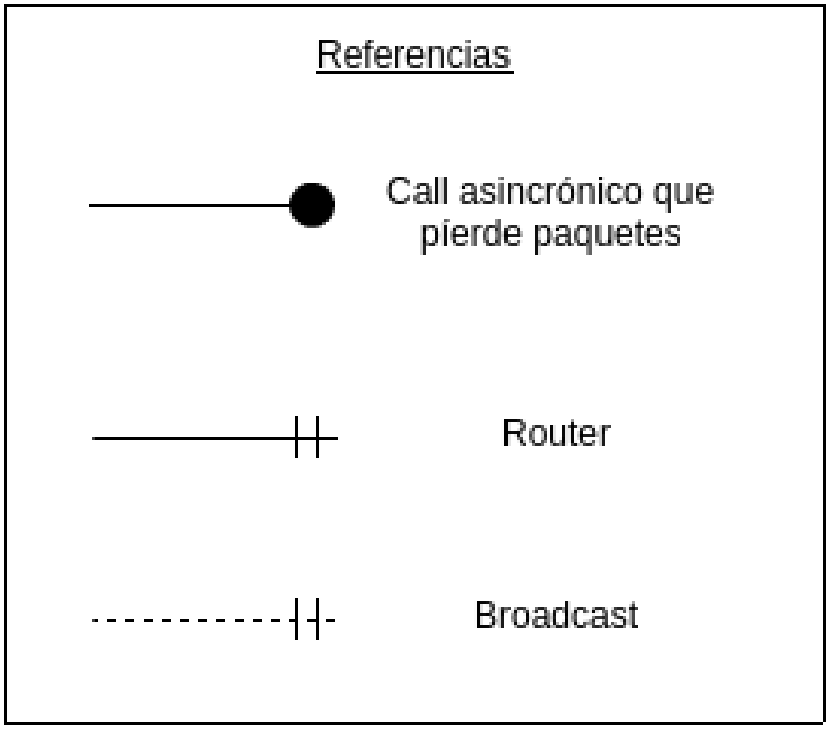

Figura 9: Referencias

3. Arquitectura Tp1

A continuación, se presenta la arquitectura relacionada con el Tp1 desarrollado por el equipo.

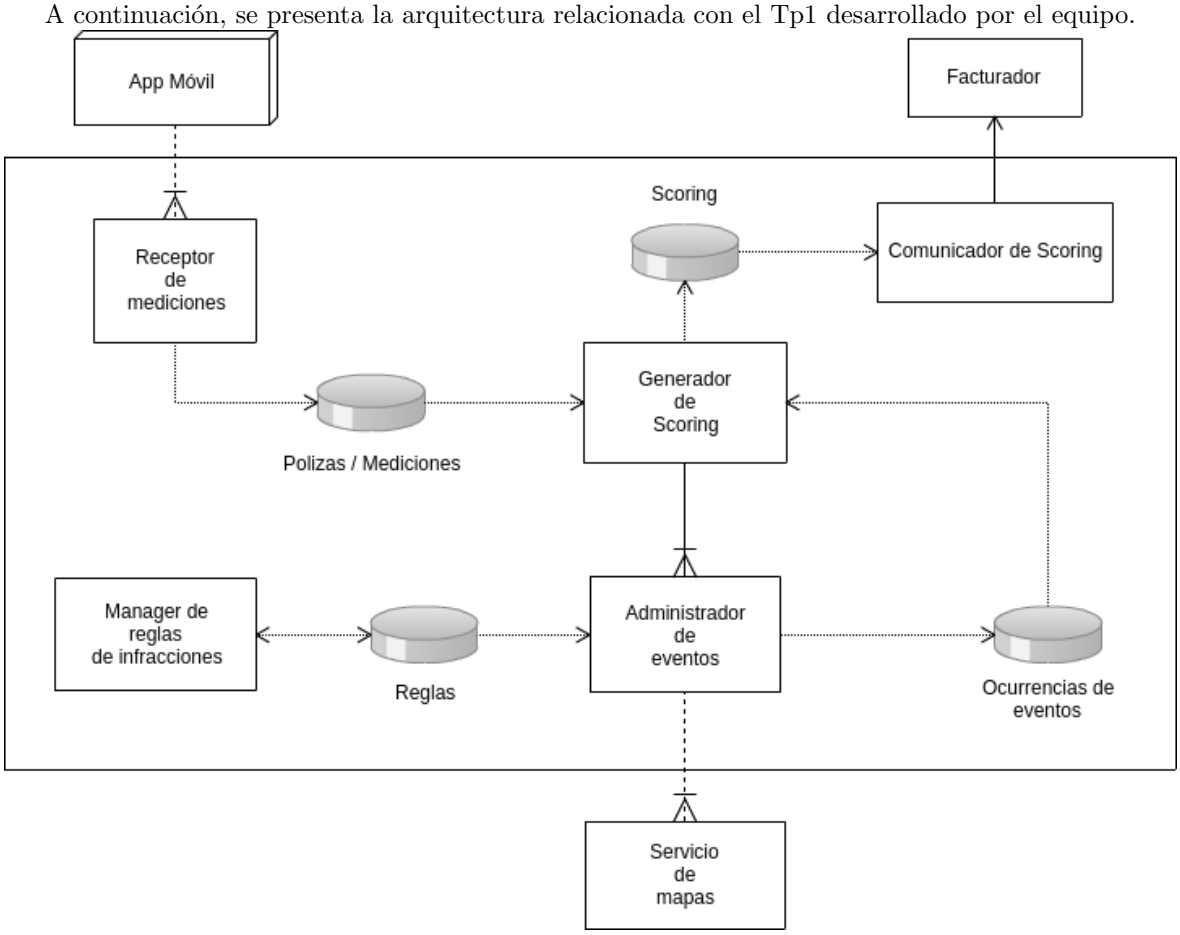

Figura 10: Arquitectura correspondiente al TP1

Como se puede observar, la arquitectura del primer trabajo práctico es bastante más sencilla que la del segundo. Esto se puede notar debido a la cantidad de componentes que tiene cada una y las responsabilidades de cada componente.

La arquitectura general del tp1 representa sólo una parte de la del tp2, principalmente la del *Controlador de Datos de GPS* del *Sistema ManejAR*. En particular, el componente *Generador de infracciones* es similar al *Generador de Scoring* en el sentido que ambos generan infracciones a partir de ciertas reglas preestablecidas.

Por lo tanto, el componente del tp1 está replicado muchas veces en el tp2, ya que en este último caso hay redundancia en el *Controlador de Datos de GPS* y, dentro de éste, en el *Generador de infracciones*.

4. Comparaciones

4.1. Comparación de los métodos utilizados

A partir de la experiencia del grupo luego de haber desarrollado ambos trabajos prácticos se pudieron encontrar las siguientes similitudes y diferencias.

Una de las principales diferencias entre Unified Process (UP) y los métodos ágiles es que en la primera se sabe desde el principio en qué tarea estará trabajando cada recurso en todo momento del proyecto. En cambio, en la segunda en cada iteración se seleccionan qué tareas hacer y luego cada integrante del grupo de trabajo las va tomando para desarrollarlas.

Otra diferencia se puede encontrar en las iteraciones de cada uno. En UP hay distintas tipos de iteraciones (fases): Incio, Elaboración, Construcción y Transición. Cada una de las fases hace énfasis en distintas disciplinas. Por el contrario, en los métodos ágiles no hay distinción entre tipos de iteraciones.

Generalmente, en ambas técnicas, se utiliza una duración fija para las iteraciones de un mismo proyecto. Sin embargo, ésto no es una regla y la duración puede ser distinta en ciertas iteraciones.

En los métodos ágiles las tareas de cada iteración se escriben como user stories, mientras que en UP se distinguen casos de uso que luego serán asignados a las iteraciones.

Otra disimilitud entre los métodos es que en UP se puede observar claramente las dependencias entre tareas de una misma iteración, mientras que en los métodos ágiles, como Scrum, las tareas son independientes.

4.2. Programming in the small vs Programming in the Large

Cuando hablamos de *Programming in the small* (DOO) es justamente como modelamos la realidad en una computadora. Utilizando objetos para representar entes de la realidad que colaboran entre ellos. Muchas veces podemos utilizar ciertos patrones, que son soluciones a problemas comunes de otros programadores al intentar modelar la realidad. Sin embargo, existen algunos aspectos de la realidad que escapan a esta tecnica como son la concurrencia y disponibilidad que necesitan otras desiciones de diseño a un mas alto nivel para poder contemplar los atributos de calidad. Esto es lo que se conoce como *Programming in the Large* y es el foco principal del TP2, donde tenemos que diseñar una solución.

5. Conclusiones

Por primera vez en la carrera tenemos que diseñar un sistema a gran escala considerando diversos factores como la escalabilidad, disponibilidad, seguridad, etc. Como grupo nos entusiasmo diseñar una aplicación con este enfoque más cercano a la industria, donde los escenarios son reales y las desiciones son justificadas empíricamente. Por eso decimos que este TP se acerca más a la realidad, donde los atributos de calidad como son la disponibilidad o performance se ven comprometidos por la realidad misma:

- sismos
- baja conectividad
- alto volumen de datos
- ataques
- etc.

El hecho de especificar los atributos de calidad y tenerlos presentes en todo momento nos ayudo a planificar y prever las falencias del sistema. Fuimos más conscientes de tomar desiciones de arquitectura y discutir diversas soluciones de antemano para evitar una mala planificación del proyecto.

Con la parte de planificación, la metodología ágil nos resulto bastante sencilla de realizar pues los cuatro estamos acostumbrados a utilizarla en el día a día laboral. Eso hizo que con las otras metodologías estemos un poco perdidos ya que ninguno tenia experiencia previa.