

Capítulo 4

Implementación del software

4.1 Hardware utilizado

El hardware utilizado para desarrollar el sistema y para las pruebas es una PC laptop HP Pavilion con un procesador Pentium IV a 2.4 Ghz y con una memoria RAM de 512 Mb.

4.2 Software utilizado

El sistema esta programado en el lenguaje de programación JAVA que como principal ventaja permite una buena movilidad entre plataformas, así como una gran facilidad de controlar elementos tales como threads (hilos), lectura de XML's, lectura y escritura de archivos de texto, entre otras cosas.

Siguiendo en la misma línea, se utilizó HTML y JSP's para dar formato a las interfaces de salida en web, tanto para el administrador como para los usuarios en general. Los JSP's permitieron compartir los objetos de salida del sistema programado en JAVA y por lo tanto no hubo problemas de compatibilidad ni nada similar.

Por otro lado, se utilizó el contenedor Jakarta Tomcat, en su versión 5.5.9, lo que permitió utilizar la misma computadora que contiene al sistema como un servidor al que pudieran acceder tanto el administrador como los usuarios. El contenedor Tomcat permite visualizar la salida web escrita en HTML y JSP's.

Así mismo, también se utilizó la librería de DOM, la cual permite abrir, leer, manipular y extraer el contenido de los archivos XML. En este caso estas librerías se utilizaron para decodificar los archivos de configuración XML con los que se puede controlar el comportamiento del sistema en sus respuestas. Se seleccionó el uso del formato XML para archivos de configuración por la facilidad de comprensión para usuarios humanos y el nivel de organización que presentan los datos sin llegar a tener una Base de Datos completa. Los archivos XML son perfectos para este sistema por la cantidad de datos que necesita como entradas (mínimos para una base de datos que requeriría una infraestructura más grande), la facilidad de entendimiento para los humanos, el enfoque semi-estructurado de los datos y la organización que facilita el procesamiento dentro del sistema de los mismos.

4.3 Arquitectura general

El sistema se monta sobre una arquitectura básica bajo el patrón de diseño llamado *Observer*, y se compone principalmente de las siguientes partes generales:

- Recolección de datos de entrada: Esta parte del sistema es la entrada de datos detectados por los sensores simulados hacia el sistema. Los principales objetos encargados de esto son: Simulador de vehículos, simulador de clima, Objeto `mx.udlap.observer.Coche`, Objeto `mx.udlap.observer.Weather`.
- Interpretación de datos de entrada y control central del sistema: El objeto `Core` es el corazón del sistema ya que contiene un `Thread` que vivirá para monitorear las entradas y es lo que le da vida al sistema, también se encarga de encausar los datos obtenidos (organizarlos en clima o información vial), así como transformar los datos viales (de tráfico) de entrada en información de utilidad para los usuarios y administradores; en este punto es donde se le da forma al objeto `HighwayStatus`. El punto anterior junto con este forman el principal patrón de diseño “Observer” al monitorear constantemente los sensores en espera de posibles entradas. Las principales partes involucradas son: Objeto `mx.udlap.system.Core`, Objeto `mx.udlap.observer.Weather`, Objeto `mx.udlap.observer.HighwayStatus`.

- Escritura de archivo log: La información ya procesada por el objeto Core, así como los diferentes climas detectados dentro de las diferentes zonas y la topografía de la autopista, son enviados a un archivo log histórico. Este objeto también funciona mediante el patrón de diseño *Observer*. Los objetos encargados de esto son: `mx.udlap.io.WriteOutputFile`, `mx.udlap.io.WriteGeneralLog`, `mx.udlap.observer.Weather`, `mx.udlap.observer.HighwayStatus`.
- Escritura de salida web para usuarios y administradores (información vial y de clima): Al igual que el escritor del archivo log, la información calculada por el objeto Core de clima y tráfico, es enviada a la interfaz web junto con la topografía de la autopista por medio de los objetos `Weather` y `HighwayStatus`, para que los usuarios puedan ver el estado interno de la autopista antes de ingresar a ella. Para los administradores se ofrece una interfaz menos depurada que la de los usuarios a fin de que se pueda tener un espectro más amplio sobre la información que fluye dentro del sistema. Este objeto también funciona mediante el patrón de diseño *Observer*. Los objetos involucrados son: `mx.udlap.observer.Weather`, `mx.udlap.observer.HighwayStatus`, `mx.udlap.io.WriteHighwayWeatherObject`.
- Calificación de la información obtenida: A partir de la información calculada por el objeto Core se hace una “calificación” de la situación actual dentro de la autopista para poder decidir si estamos ante alguna situación de emergencia alta, media o una situación normal que no amerita mayor atención. Los valores “umbral” para poder tomar decisiones de si la situación es normal o peligrosa para cada zona son dados por el administrador por medio de un archivo de configuración XML llamado `Constraints.xml`. Este objeto funciona también mediante el patrón de diseño *Observer*. Los objetos que hacen posible esto son: `mx.udlap.observer.Weather`, `mx.udlap.observer.HighwayStatus`, `mx.udlap.qualification.Qualification`, `mx.udlap.qualification.TrafficQualification`, `mx.udlap.qualification.WeatherQualification`, `mx.udlap.qualification.AccidentDetection`, `mx.udlap.qualification.TrafficConstraints`, `mx.udlap.qualification.WeatherConstraints`, `mx.udlap.xml.XDTreeFactory2`, `mx.udlap.xml.ReducedTreeLeaf2`.

- Elección de la información que aparecerá dentro de la autopista y administración de los tableros dinámicos (VMS o *Variable Message Sign*): Habiendo calificado la situación para cada una de las zonas de la autopista, se elige de un pool de mensajes, provistos por el administrador mediante un archivo XML llamado `vms.xml`, los mensajes de texto que mejor convengan según la situación que se viva dentro de cada zona, así mismo se elige la información que aparecerá dentro de cada tipo de tablero (I, II o III) existente dentro de los límites de la autopista. Los objetos que lo hacen son: `mx.udlap.qualification.ManageVMS`, `mx.udlap.xml.ReducedTreeLeaf_vms`, `mx.udlap.xml.XDTreeFactory_vms`, `mx.udlap.vms.MessageSign`, `mx.udlap.vms.VMS`.
- Escritura de salida para anuncios VMS: Finalmente una vez decidido qué mensajes se pondrán para cada tipo de tablero existente dentro de la autopista, se procede a escribir dichos resultados en el web, para que pueda ser visitado por los administradores y verificar el funcionamiento de la autopista. Los objetos que hacen esto son: `mx.udlap.vms.MessageSign`, `mx.udlap.vms.VMS`, `mx.udlap.io.WriteOutputVMSInfo`.

4.4 Objetos básicos

4.4.1 Objeto `HighwayStatus`

Este objeto fue creado con la finalidad de poder describir el estado, en términos de vialidad y distribución de zonas dentro de la autopista. Los datos obtenidos por el objeto `Core`, y que provienen de los sensores simulados, son procesados por el mismo y transformados en información útil, la que a su vez se transporta a través del sistema dentro de este objeto. Este objeto sirve como medio de comunicación entre las partes internas del sistema, así como para informar la situación monitoreada a las interfaces web, las cuales despliegan versiones depuradas de este objeto. Los sensores de velocidad dentro de la autopista (simulados en este sistema), proveen la información necesaria para actualizar constantemente los campos dentro de este objeto. Los datos más importantes que contiene son:

- Flujo detectado por zona
- Flujo promedio dentro de la autopista
- Unidad utilizada para describir el flujo
- Densidad detectada por zona
- Densidad promedio dentro de la autopista
- Unidad utilizada para describir la densidad
- Velocidad detectada por zona
- Velocidad promedio dentro de la autopista
- Unidad detectada para describir a la velocidad
- Conteo total de vehículos que han transitado por la autopista.
- Número de zonas
- Longitud en kilómetros de cada zona

Básicamente dentro de este objeto se encuentra toda la información necesaria de las zonas de la autopista, así como sus longitudes, el conteo de vehículos que han transitado, las velocidades, los flujos y las densidades detectadas y calculadas por zona, así como estos mismos valores pero en promedio a lo largo de toda la autopista.

Es importante hacer notar que en este objeto se definen a manera de valores constantes las unidades con las que se manejarán la velocidad, el flujo y la densidad. Esto es primordialmente para efectos de salida de datos, así que si en algún caso se quisiera cambiar de unidad, habría que asegurarse que los datos de entrada que provienen de los sensores estén en las nuevas unidades y posteriormente cambiar dichos valores dentro de este objeto. Los valores *default* con los que se maneja este objeto son:

- Velocidad: km/h (Kilómetros por hora)
- Flujo: vph (Vehículos por hora)

- Densidad: `vehicles-per-km` (Vehículos por kilómetro)

Dentro del sistema existe sólo un único objeto de este tipo, ya que este objeto simboliza las situaciones detectadas dentro de la autopista y no podría haber doble información.

4.4.2 Objeto `Weather`

Este objeto es el descriptor de la información obtenida por los sensores de clima dentro de la autopista. Dichos sensores (en este caso simulados) se encargan de monitorear continuamente el estado climatológico dentro de las zonas en la autopista, para después llenar los campos dentro de este objeto y poder manejar la información climatológica dentro de la carretera.

Este objeto contiene los siguientes campos:

- Humedad
- Unidades de la humedad
- Temperatura
- Unidades de la temperatura
- Velocidad del viento
- Unidad de la velocidad del viento
- Visibilidad
- Unidad para la visibilidad
- Presencia de hielo
- Zona en que se hizo la medición
- Hora en que se hizo la medición

A diferencia del objeto `HighwayStatus`, habrá un objeto de este tipo por cada zona que

exista en la autopista, ya que cada clima puede variar dentro de cada zona, especialmente si las zonas son muy grandes y pudiera haber grandes cambios de clima dentro de ellas.

Este objeto tiene la finalidad de representar el clima presente dentro de cada una de las zonas existentes en la carretera para que después pueda ser utilizado como información de estado interno de la autopista. Al igual que el objeto `HighwayStatus` se manejan unidades predefinidas:

- Unidad de temperatura: °C (Grados Centígrados)
- Unidad de humedad: % (Porcentaje de humedad en el ambiente)
- Unidad para la velocidad del viento: kph (kilómetros por hora)
- Unidad de medida para la visibilidad: kms (kilómetros)
- Para la presencia del hielo simplemente se utiliza un valor binario, existe o no existe hielo.

Todos estos datos son recolectados por los sensores climatológicos distribuidos a lo largo de la autopista (en este caso simulados) y pasados al sistema quien se encarga de llenar los objetos `Weather` respectivos para cada zona existente dentro de la autopista.

4.4.3 Objeto Coche

La función de este objeto es básica, y es la de representar a un vehículo cualquiera que circule dentro de la autopista. Es creado por el simulador de vehículos (`mx.udlap.sim.CarSimulator`) y contiene información básica como la velocidad del vehículo, un id, y la hora, los minutos y los segundos en que fue detectado por el sensor dicho vehículo. Para que sea posible la detección se aplica el patrón de diseño *Observer*, por lo tanto este objeto va acompañado con su respectivo objeto `observer` y objeto `subject`, `mx.udlap.observer.CarObserver` y `mx.udlap.observer.CarSubject` respectivamente. Estos últimos, como se ha mencionado, forman parte del patrón de diseño *Observer* y sirven para la detección de nuevos datos generados por los simuladores.

4.4.4 Objeto Core

El objeto central del sistema es el `mx.udlap.system.Core` y es el encargado de recibir la información generada por los simuladores (`mx.udlap.sim.CarSimulator` emulando a los sensores de velocidad y `mx.udlap.sim.WeatherSimulator` a los sensores climatológicos) para después convertir estos datos sin aparente sentido en información útil para el sistema, los administradores y los usuarios.

En el ámbito de detección de tráfico funciona básicamente como un contador de automóviles para cada zona, cuando un auto ingresa en una zona se resta un auto en la zona anterior. De esta manera es posible obtener la velocidad por zona, la velocidad promedio, el flujo por zona, el flujo promedio, la densidad por zona y la densidad promedio, información con la que se generarán los objetos `mx.udlap.observer.HighwayStatus` que servirán posteriormente para escribir el archivo log, mandar la información al sitio de visualización web, calificar la situación y finalmente tomar decisiones para los escenarios detectados dentro de la autopista.

4.5 Implementación de los módulos

En esta parte se explicará con más detalle el funcionamiento de las partes en que se ha separado el sistema para comprender mejor su funcionamiento y a la vez para comprender la finalidad de cada objeto importante involucrado en el funcionamiento del mismo. También se explicará la organización y formatos de los archivos de configuración disponibles para los administradores encargados del sistema.

4.5.1 Simulador de tráfico

Esta es una aplicación que no se considera estrictamente como parte del sistema ya que no interviene

en el procesamiento del mismo, sin embargo, es una aplicación muy importante porque es la que nos permite modificar las entradas de información de tráfico hacia nuestro sistema.

Este simulador permite emular vehículos transitando dentro de nuestra autopista también simulada, lo hace mediante la generación de objetos `mx.udlap.observer.Coche`, a los cuales les asigna aleatoriamente una velocidad de cruce que será con la que transiten durante todo su trayecto dentro de nuestra autopista virtual, esta velocidad es escogida aleatoriamente dado un rango de velocidades especificado por el usuario, dando la velocidad mínima de los vehículos y la máxima que tendrán los mismos, así mismo, también el sistema les asigna un *id* único con el cual se facilita la depuración del sistema, es importante saber que éste *id* no es un dato que comparta con el sistema general.

Los vehículos generados por este simulador no tendrán cambios de velocidad dentro de la autopista y serán generados de acuerdo a un *timer* definido por el usuario administrador. Dicho *timer* consiste en seleccionar una cantidad de vehículos a aparecer por evento y seleccionar también cada cuantos segundos ocurrirá el evento. Por ejemplo, se puede seleccionar que aparezcan 2 vehículos por evento cada 3 segundos. El simulador cuenta con una interfaz visual que se iniciará una vez que corramos el sistema completo.

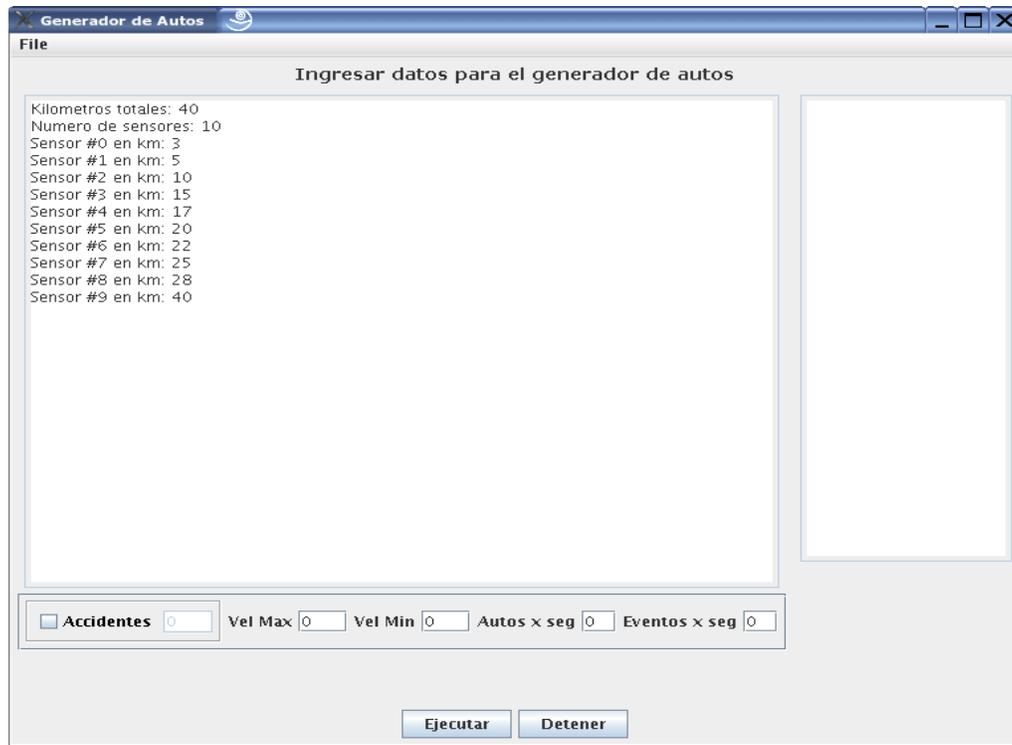


Imagen 4.1. La interfaz de control del simulador de tráfico del sistema

Esta aplicación requiere de entradas en un archivo de texto, esta entrada lleva por nombre `sensores.txt`, en él, se le indica al simulador la topografía de nuestra carretera con los sensores que tenemos disponibles. A continuación se muestra un ejemplo del archivo de entrada con comentarios (para 3 zonas):

Archivo sensores.txt

```

40      <-- Longitud total de la autopista
3       <--Contamos con 3 sensores
3       <--El primero se encuentra en el km 3
20      <--El segundo en el km 20
40      <--El tercero en el km 40

```

Dada la configuración mostrada en el archivo `sensores.txt`, por ejemplo, en el siguiente diagrama se muestra la representación gráfica de la distribución de los sensores, en este caso simulados, y su relación con el sistema y con el módulo que recolecta la información.

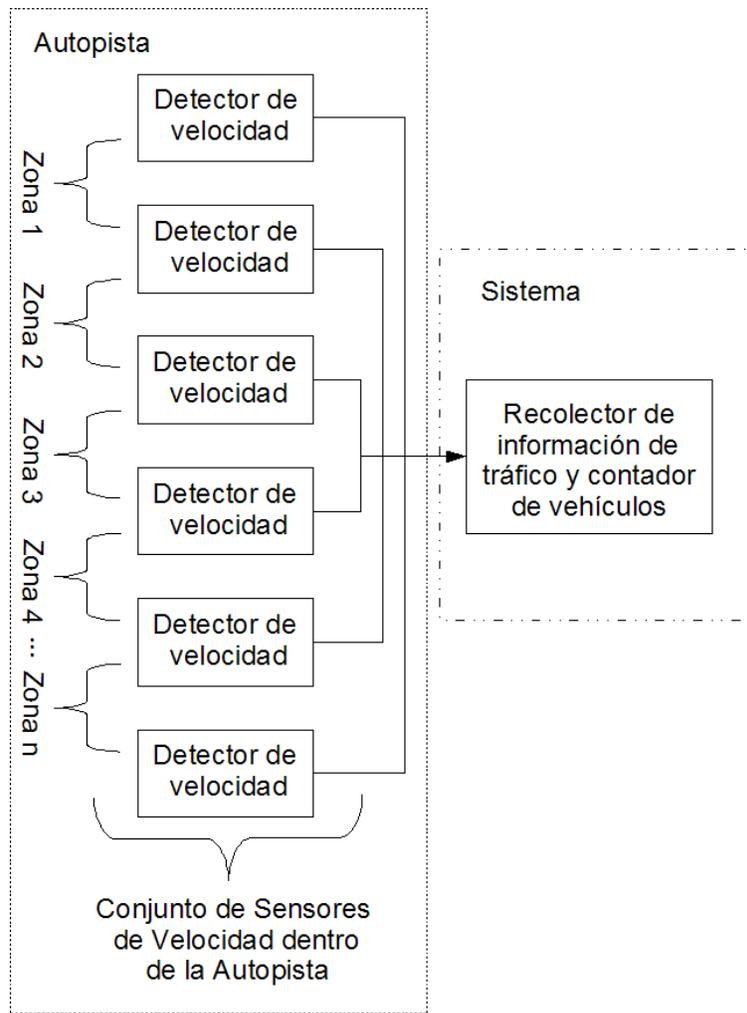


Imagen 4.2. Diagrama de bloques de los sensores de entrada para datos vehiculares

4.5.2 Simulador de clima

El simulador de clima es una aplicación de simulación que nos ayuda a generar datos de climas aleatorios para cada una de las zonas. Esta aplicación se encarga de generar diferentes climas para poder probar las capacidades del sistema sin tener físicamente sensores climatológicos para pruebas.

La aplicación genera objetos `mx.udlap.observer.Weather` con climas aleatorios cada minuto o cada hora (estos valores son modificables si se desea) según se elija. Los rangos (valores máximos y valores mínimos) para temperatura, humedad, velocidad del viento, hielo y visibilidad están predefinidos, pero también son modificables.

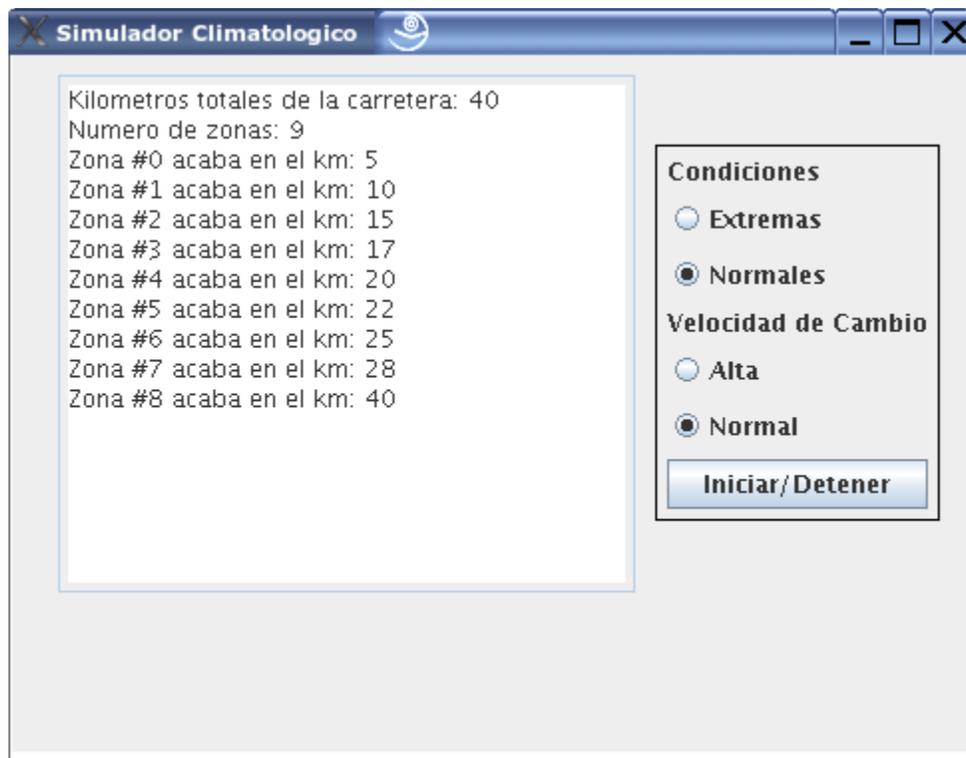


Imagen 4.3. Interfaz del simulador de clima

Cuenta también con opción de generar clima “normal” o clima extremo, esto con la finalidad de probar la respuesta del sistema ante tales escenarios. Así mismo, tiene la opción de activar cambios rápidos de clima (cada minuto) o cambios normales (alrededor de cada hora). Utiliza también el mismo archivo de texto de entrada que el simulador de vehículos además de otro archivo de entrada llamado zonas.txt del cual se muestra un ejemplo (con 4 zonas). Así también cuenta con una interfaz gráfica disponible para realizar los cambios más fácilmente.

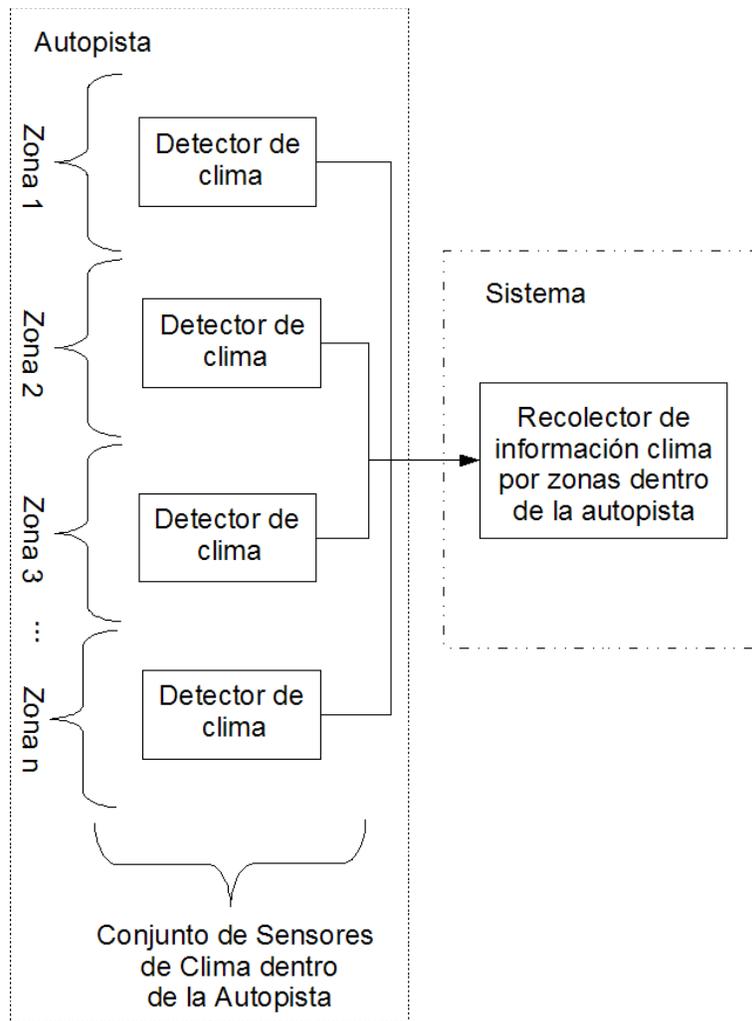


Imagen 4.4. Diagrama de bloques para el hardware de entrada que provee información de clima dentro de la autopista

Archivo zonas.txt

```

4      <--Número de zonas
3      <--Primera zona inicia en el km 3
13     <--Segunda zona inicia en el km 13
20     <--Tercera zona inicia en el km 20
31     <--Cuarta zona inicia en el km 31
40     <--Longitud total de la autopista

```

4.5.3 Procesamiento de datos obtenidos de los sensores

El procesamiento de los datos generados por los simuladores es hecho por el objeto `mx.udlap.system.Core`. La principal finalidad de este objeto es la de recolectar todos los datos que generen los sensores tanto de tráfico como de clima (en este caso generados por los simuladores de tráfico y clima), para después generar información utilizable para los usuarios.

Este objeto utiliza como entrada de configuración un archivo de texto llamado `zonas.txt` que deberá de estar dentro de la carpeta `mx.udlap.io`. Dicho archivo es el mismo utilizado por el simulador de clima y servirá para dar a conocer la topografía de zonas dentro de la autopista, esto con la única finalidad de poder organizar los datos obtenidos en zonas.

En el caso de los datos generados por el simulador de tráfico, dicho simulador genera objetos tipo `mx.udlap.observer.Coche` dándoles valores aleatorios de velocidad. Por medio del patrón de diseño *Observer*, el simulador de tráfico avisa que algún sensor ha sido estimulado por un vehículo y reporta qué sensor ha sido activado y la velocidad a la que pasó el vehículo. Con esta información el objeto `Core` realiza un conteo de vehículos, esto con la finalidad de llevar un registro del número de vehículos que han transitado, además de registrar la velocidad detectada para poder obtener una nueva velocidad promedio dentro de la zona en que se realizó el evento; además la nueva velocidad servirá para obtener también la velocidad promedio dentro de la totalidad de la autopista.

El conteo de vehículos se utilizará principalmente para obtener la densidad dentro de la zona en que ha circulado el vehículo detectado y para obtener la densidad promedio dentro de toda la autopista. La velocidad por zonas y promedio así como la densidad por zonas y promedio servirán para obtener también el flujo por zonas y promedio dentro de la autopista. Todos estos datos son generados por el objeto `Core` cada 20 segundos con los datos recolectados en este lapso de tiempo. Si existiera la necesidad, es posible también cambiar este lapso de tiempo a uno que mejor parezca, siendo siempre especificado en segundos.

Con la información calculada a partir de los datos obtenidos de los simuladores, el objeto `Core` se encarga de crear un objeto `mx.udlap.observer.HighwayStatus` con los datos actualizados y se encargará de mantenerlo actualizado con toda la información que se vaya generando durante el tiempo que el sistema se encuentre activo.

Con respecto a la detección de clima, el sistema se acopla al simulador generador de climas de la misma manera que con el sistema generador de tráfico, por medio de un patrón de diseño `Observer`. Sabiendo esto, se puede explicar que cada vez que el simulador registre un cambio de clima dentro de las zonas, alertará al objeto `Core` quién obtendrá por medio de un arreglo de objetos `mx.udlap.observer.Weather` los nuevos climas detectados dentro de cada una de las zonas.

Finalmente, una vez obtenidos los objetos `Weather` (uno por cada zona de la autopista) y el objeto `HighwayStatus`, el objeto `Core` se encarga de difundir esta información a múltiples partes. Todas las partes a las que avisa este objeto están acopladas a él por medio del patrón de diseño `Observer` que permite una actualización de datos inmediata sin tener que acoplar fuertemente los objetos.

La primera de ellas es el escritor de archivo log llamado `mx.udlap.io.WriteGeneralLog` en donde se escribirá constantemente (cada que se registre un cambio de estado en la autopista tanto de clima como de tráfico) una representación del nuevo estado de la autopista en formato texto simple. El segundo es el escritor de objetos Web denominado `mx.udlap.io.WriteHighwayWeatherObject`, que servirá para que los usuarios puedan acceder a la información de estado de clima/tráfico dentro de la autopista, así como también los administradores.

Por último esta misma información (el arreglo de objetos `Weather` y el objeto `HighwayStatus`) se enviará también a el objeto llamado `mx.udlap.qualification.Qualification`, que será el que se encargue de calificar las situaciones tráfico/clima para una posterior toma de decisiones.

Como se ha podido ver este objeto es una de las principales partes del sistema y de él depende el buen funcionamiento y flujo de la información interna, además que él se encarga de generar toda la información a partir de datos obtenidos directamente desde los sensores de clima o tráfico. Básicamente está conformado por un thread que vivirá para monitorear constantemente a los sensores por cualquier cambio de estado.

4.5.4 Sistema central de calificación de situaciones

Este módulo es junto con el objeto `Core` uno de los más importantes ya que de él depende la interpretación que la autopista le da a la información de tráfico y clima. La principal función de éste módulo es la de “calificar” las situaciones registradas dentro de la autopista. Este tipo de calificación será en base a lineamientos establecidos por los administradores por medio de archivos de configuración en formato XML.

En primer lugar el objeto `mx.udlap.qualification.Qualification` recibe desde el objeto `Core` el arreglo de objetos `Weather` y el objeto `HighwayStatus` para poderles dar una calificación. Las calificaciones de los eventos ocurridos dentro de la autopista se generan en base a los objetos `mx.udlap.qualification.TrafficConstraints`, `mx.udlap.qualification.WeatherConstraints`, `mx.udlap.qualification.TrafficQualification` y `mx.udlap.qualification.WeatherQualification`. El objeto `Qualification` leerá el archivo XML `constraints.xml` utilizando los objetos `mx.udlap.xml.XDTreeFactory2` y `mx.udlap.xml.ReducedTreeLeaf2`, este último hará uso de las librerías para el parsing de archivos XML llamadas DOM y así poder acceder al contenido del archivo XML.

El archivo XML leído contiene los *constraints* (restricciones) dados por el administrador dependiendo de la topografía de la autopista, es decir, si la zona 1, por ejemplo, contiene muchos segmentos de carretera rectos y se conoce un clima normalmente seco, entonces en el archivo XML se

pueden especificar datos (para tráfico y clima) que servirán como umbral para detectar situaciones anormales o fuera de lo común para dicha zona. Para este caso habiendo muchos tramos rectos se puede dar por “normal” velocidades cercanas al límite permitido, y conociendo que la zona cuenta con un clima seco, entonces se podría dar como restricciones un clima sin lluvia, temperatura media, velocidad del viento baja, etc. Si el clima llegara a cambiar fuera de estas especificaciones o simplemente hubiera situaciones de riesgo, como por ejemplo hielo, lluvia, viento fuerte, el sistema registraría que la situación detectada por los sensores está sobrepasando los valores de seguridad especificados y calificará como peligrosa la situación. Más adelante se explicará cómo se califican las diferentes situaciones.

A manera de estandar el sistema maneja diferentes claves para el tipo de información que se maneja. Por ejemplo, la información obtenida de los sensores de tráfico será conocida como tipo 1, así mismo, la información obtenida desde los sensores de clima será conocida como información tipo 2 y por último la información obtenida desde la cámara que detecta problemas con lluvia será conocida como tipo 3.

Clasificación de la información obtenida desde los sensores	
Información de tráfico (vial)	Tipo 1
Información climatológica	Tipo 2
Información visual (cámara detectora de lluvia)	Tipo 3

Tabla 4.1. Clasificación de la información de sensores

Por otro lado, el archivo `constraints.xml` tiene un formato (en XML) que permite definir las restricciones o *constraints* de una manera más accesible y esta pensado para ser entendido claramente por un usuario y a la vez facilita al sistema el ordenamiento de datos para su procesamiento. El siguiente ejemplo consta de dos zonas:

Ejemplo de constraints

```
<constraint>
  <name>11</name>
  <description>FLUJO</description>
  <qualification-value>180</qualification-value>
```

```

    <increments>850</increments>
    <zone>
      <number>1</number>
      <minimum-value>2300</minimum-value>
    </zone>
    <zone>
      <number>2</number>
      <minimum-value>2300</minimum-value>
    </zone>
  </constraint>

```

El archivo XML de configuración para las restricciones de tráfico o clima se compone de lo siguiente:

- *Traffic/Weather Constraint*
 - *Constraints*
 - *Name*
 - *Description*
 - *Qualification Value*
 - *Zone*
 - *Number*
 - *Minimum Value*
 - *Increments*

Las *tags* anteriormente nombradas son descritas con detalle a continuación:

- `<traffic-constraints></traffic-constraints>` `o` `<weather-constraints></weather-constraints>` Estas *tags* indican que queremos especificar una restricción de tráfico o una restricción de clima.
- El tag `<constraint>` nos indica que estamos hablando de una restricción. Cada *tag* de este tipo contiene diversos *tags* establecidos.
- El primer tag es el llamado `<name>` sirve para identificar al *constraint*, es decir, es un id que debe de ser único. Para construir los nombres de los problemas a calificar se han unido los tipos de problema con un id único para el problema. Se han dado valores predeterminados para que

todos los *constraints* de tráfico inicien con el número 1, los de clima iniciarán con el número 2 y los problemas detectados por la cámara (visuales) comenzarán con el número 3 (ver tabla 4.1). Por ejemplo, si quisiéramos nombrar los *constraints* de tráfico iríamos desde el 11 hasta el 1n, siendo n el último número de restricción de tráfico integrado al sistema.

- El siguiente *tag* es el `<description>` sirve principalmente para que el usuario sepa de qué restricción se está hablando.
- El *tag* `<qualification-value>` indica la “gravedad” que provocaría la restricción si se saliera de los niveles establecidos por el administrador de seguridad. Es decir, a un problema de exceso de velocidad se le asigna una calificación mucho mayor (es más peligroso) que un problema de humedad en el clima o un problema de temperatura, los cuales son calificados con valores mucho más bajos.
- El *tag* denominado `<zone>` contiene *tags* internos y principalmente sirve para representar a una zona de la autopista y los valores de seguridad para cada una de ellas.
- El *tag* llamado `<number>` indica el número de zona a la que se está refiriendo, esto es para identificar completamente a qué zona de la autopista se está refiriendo.
- `<minimum-value>` Este *tag* indica el valor de seguridad para la zona especificada y para la restricción especificada en el *tag* `<description>`. Por ejemplo si nos refiriéramos a la velocidad, el valor de seguridad para una zona imaginaria con curvas pronunciadas sería de 70 kilómetros por hora. Esto es muy útil ya que si los datos recolectados dentro de esta zona sobrepasan los valores de seguridad la zona se califica con el valor indicado en el *tag* `<qualification-value>` y esto nos ayudará a tomar decisiones para saber qué tipo de advertencias e información daremos a los usuarios de la autopista más adelante.
- El *tag* que se llama `<increments>` indica que habiendo rebasado los límites de seguridad especificados por el *tag* `<minimum-value>` entonces se comenzará a calificar en incrementos especificados por este *tag*. Es decir, si nuestra velocidad de seguridad está especificada en 90 kilómetros por hora pero la velocidad detectada promedio dentro de la zona analizada es de 120 kilómetros por hora, y tenemos un incremento de 10 kilómetros por hora, entonces tenemos un exceso de 30 kilómetros por hora, es decir, 3 veces el incremento, por lo que la calificación será multiplicada por 3. Este resultado, finalmente nos dará la calificación para la restricción de velocidad para una zona.

Ejemplo de *tag* <constraint>

```
<constraint>
  <name>23</name>
  <description>VIENTO</description>
  <qualification-value>150</qualification-value>
  <zone>
    <number>1</number>
    <minimum-value>10</minimum-value>
  </zone>
  <increments>100</increments>
</constraint>
```

Para poder calificar una situación dada dentro de una zona, el objeto *Qualification* utiliza las restricciones o *constrains* antes mencionadas de la siguiente manera. Por ejemplo, si en la zona 0 (la primera zona) de la autopista consideramos una velocidad segura (*minimum-value*) de 90 kilómetros por hora y nuestros sensores detectan una velocidad promedio de 120 km/h entonces tendríamos un exceso de velocidad de 30 kilómetros por hora; si a su vez, en nuestras restricciones definimos a la velocidad con una gravedad de (*qualification-value*) de 200 y con incrementos (*increments*) de 10, entonces tenemos que los 30 kilómetros por hora sobrantes representan a 3 incrementos (3 de 10 kilómetros por hora) por lo que la gravedad, especificada de 200) se multiplicaría por 3, lo que como resultado nos daría 600. Por lo tanto, para la zona 0 tendríamos una calificación de 600 en el rubro de velocidad. Por cada rubro o problema que controlemos (velocidad, visibilidad, hielo, densidad, etc) se generará una calificación por zona.

Velocidad Detectada Promedio dentro de la zona X: 120 km/h

Velocidad limite designada por el administrador para la zona X: 90 km/h

$$120 \text{ km/h} - 90 \text{ km/h} = 30 \text{ km / h}$$

Exceso de 30 km/h

Gravedad de la situación "Exceso de velocidad": 200

Incrementos cada 10 km/h

Número de incrementos para el exceso detectado: 3

Calificación final para la zona X en la situación "Exceso de Velocidad": $3 \times 200 = 600$

4.5.5 Objetos para representar a las restricciones (*constraints*)

Las restricciones dadas por el administrador del sistema se representan en un archivo XML llamado `constraints.xml`. El sistema se encarga de leer este archivo y de mapearlo, dependiendo si es climatológico o de tráfico, dentro de las clases llamadas `mx.udlap.Qualification.TrafficQualification` y `mx.udlap.Qualification.WeatherQualification`.

Estos archivos contienen datos precargados (modificables por el administrador) para que, en caso de que ocurriera algún problema en la lectura del archivo XML que impidiera cargar la información correctamente dentro del objeto o esta estuviera en formato indebido, el sistema sea capaz de seguir funcionando con los datos *default* precargados dentro de cada objeto. Es importante recalcar que la información predeterminada contenida dentro de estos objetos existe únicamente como información de emergencia en caso de que ocurriera un error en la lectura del archivo XML y ya que esta información tiene tendencia a ser modificada constantemente, es mucho más cómodo para el administrador y menos problemático modificar estos valores directamente sobre el XML.

Una vez que toda la información ha sido cargada sin problemas y de manera correcta dentro de los objetos *constraints* para cada tipo de calificación (tráfico, visual o clima) el objeto `TrafficConstraint`, `WeatherConstraint` y `VisualConstraint` están listos para ser manejados por el objeto `Qualification` y así poder realizar las calificaciones por zona correspondientes.

4.5.6 Calificación de situaciones

El objeto `Qualification` pasa los objetos *constraints* (de tráfico, clima y visual) a los objetos `mx.udlap.qualification.TrafficQualification`,

`mx.udlap.qualification.WeatherQualification` y `mx.udlap.qualification.VisualQualification` que serán los encargados de contener el algoritmo que calificará las situaciones detectadas previamente por los sensores apostados en la autopista.

Estos objetos reciben también desde `Qualification` los valores detectados por los sensores dentro de toda la autopista y para cada problema a calificar. El algoritmo encargado de calificar las situaciones revisa cada problema (velocidad, densidad o flujo, para el caso de tráfico), organizado por tipos (clima, tráfico, visual), si los problemas superan el límite establecido por el administrador como seguro (`minimum-value`) dentro del archivo XML llamado `constraints.xml`, entonces se comienza a multiplicar la gravedad de la situación por cada incremento excedente con respecto al valor de seguridad (ver 4.5.4).

Si sólo existiera una sola situación que excede su valor de seguridad, entonces la calificación de dicha zona calculada sería igual a la calificación de la situación que excede los límites de seguridad. En caso de que existieran más situaciones que exceden los límites de seguridad dentro de un mismo tipo de problema, entonces el sistema suma las calificaciones finales provocando que la zona calificada obtenga aún una calificación más alta por su posible peligrosidad.

En el caso de que no existieran situaciones que excedieran los límites de seguridad establecidos dentro del archivo XML por el administrador, entonces la zona sería calificada con el valor **0**, lo que representa que todas las situaciones dentro de la zona sensada se encuentran por debajo de los valores de seguridad previamente establecidos para dicha zona. Lo anterior es calificado por el sistema como una zona con condiciones normales y con las lecturas bajas ante los límites impuestos por los administradores. En la siguiente parte a este tipo de zonas se les llamará “Generales” ya que se prestan para presentar información de tipo general o informativas dentro de ellas.

Las calificaciones por zona van acompañadas de una pequeña clave (de un dígito) para identificar el principal problema dentro de la zona, no importando si es una o varias las situaciones de

peligro, siempre es identificada la que se aleja más de su valor de seguridad y se le da una clave para identificar el mayor problema a enfrentar dentro de la zona actual que se esta calificando.

Finalmente los objeto calificadores de problemas específicos (*TrafficQualification*, *WeatherQualification* y *VisualQualification*) entregarán una matriz que comprende las calificaciones obtenidas por la zona. El formato de la matriz entregada por cada tipo (tráfico, clima y visual) se muestra a en la siguiente tabla.

Especificación de la matriz de salida de los objetos <i>Qualification</i>			
	Número de zona	Calificación total dentro de la zona (peligrosidad)	Principal problema de la zona
Número de zonas dentro de la autopista (1-n)	Zona 0	100	11
	Zona 1	0	0
	Zona n	975	23

Tabla 4.2. Descripción de la matriz resultante de los objetos *TrafficQualification*, *WeatherQualification* y *VisualQualification*.

Dentro del sistema se generarán 3 de estas matrices con las calificaciones por cada tipo de problema (tráfico, clima y visual) conteniendo todas las calificaciones con el principal problema por zona dentro de ellas. Esta información será la base para detectar posibles accidentes.

Por último es importante mencionar que los objetos *TrafficQualification* y *WeatherQualification* están separados del objeto *Qualification* por razones de modularidad, ya que se podría cambiar el algoritmo (de ser necesario) para calificar en cualquiera de estos objetos sin afectar a ningún otro ni afectar el funcionamiento del objeto *Qualification* ni del sistema.

Por otro lado, el sistema cuenta también con un detector de lluvia por medio del procesamiento

de imágenes. Este sensor está hecho para que afecte a el nivel de humedad detectado dentro de la autopista, por medio de imágenes. Para esto se utilizará una cámara para que tome dichas imágenes, ésta es simulada y el sistema tomará una imagen para procesar desde el directorio, especificado previamente, llamado `mx.udlap.image_processing.input_files`.

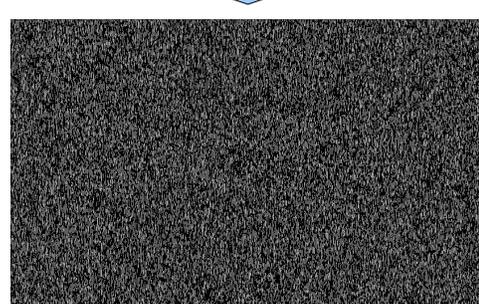
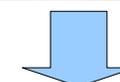
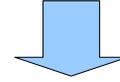
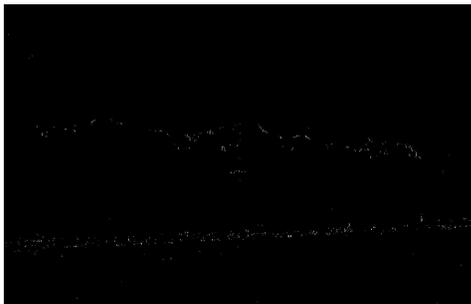
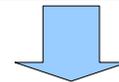
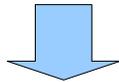
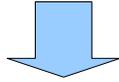
En primer lugar, el sistema toma sólo una imagen (aleatoriamente) que se encuentre dentro de este directorio. Después el sistema lee la configuración de *constraints* para la parte visual desde el archivo XML `constraints.xml`. Posteriormente, el sistema procede a detectar la lluvia (si existiera) dentro de la imagen, por medio del siguiente proceso. La imagen de entrada es convertida a tonos grises, es decir, cada valor de las 3 matrices que componen una imagen (Rojo, Verde y Azul) son igualados, esto da tonos grises en la imagen; lo anterior se hace para hacer más fácil la detección de las gotas de lluvia. El siguiente paso, es hacer una Convolución de la imagen con una matriz como la mostrada en la tabla 4.3. Esto se hace para detectar las líneas dentro de la imagen que sean verticales o lo más cercano a la verticalidad. Una vez que se ha detectado las líneas verticales o casi verticales (buscando por gotas de agua o trazas de las gotas de agua cayendo), se obtiene una imagen en tonos de grises con sólo las líneas verticales marcadas, por lo que lo próximo a hacer es aplicarle un operador Threshold. La finalidad de este operador es la de hacer binaria los tonos de la imagen a partir de un nivel. En este caso el nivel seleccionado es de 60. Por lo que todos los valores encontrados en la imagen menores o iguales a 60 se convierten en 0 y los valores mayores a 60 son convertidos en 255 (el mayor valor posible para un color en una imagen).

Con el paso anterior se logra obtener con claridad todas las líneas verticales en un color blanco, mientras que todas las demás partes de la imagen se convierten en negro. Por último, el sistema cuenta la cantidad de valores blancos en la imagen. En una imagen normal, sin lluvia, se obtienen valores de blancos alrededor de 1,099; mientras que para imágenes con lluvia ligera se obtienen valores desde 5,742 hasta 37,134; para imágenes opacas o no opacas con lluvia muy fuerte o muy cerrada, los valores encontrados son desde 76,832 hasta 251,346. Como se puede observar se obtienen valores muy bien marcados de los niveles de lluvia detectados.

Tabla 4.3 Matriz utilizada para la Convolución con la imagen detectada		
-1	2	-1
-1	2	-1
-1	2	-1

Por último, ya obtenido el nivel de puntos blancos que tiene la imagen, se procede a calificar la situación de igual manera que las situaciones de tráfico y de clima. Solo que esta vez el objeto `mx.udlap.qualification.VisualQualification` realiza su calificación dado el nivel de blanco encontrado. La matriz resultante de esta calificación es dada al objeto `AccidentDetection` para que a su vez sea integrada a las demás calificaciones obtenidas.

A continuación se muestra el proceso de detección de lluvia. En la serie de la izquierda se puede ver una imagen sin lluvia, así como también se puede apreciar que los puntos detectados por el operador *threshold* son muy pocos, insuficientes para declarar lluvia. Por otro lado, la serie de la derecha muestra una imagen con bastante lluvia, por lo que se puede apreciar que durante el proceso se hacen notablemente visibles las gotas de lluvia.



Escala de grises

Convolución

Threshold

Imagen 4.5 y 4.6. Procesamiento de imágenes para la detección de lluvia, izquierda sin lluvia, derecha con lluvia.

4.5.7 Detección de posibles accidentes

El objeto llamado `mx.udlap.qualification.AccidentDetection` es el encargado de detectar posibles situaciones en donde exista un accidente de tipo vial, para posteriormente poder etiquetar a la zona conflictiva como zona accidentada, así como también es el encargado de organizar y complementar las demás entradas (clima y visual) con la entrada de los sensores de tráfico, para poder tener una imagen más completa de la situación existente dentro de la autopista y poder tener una mejor respuesta a los accidentes o diferentes combinaciones de situaciones que existan dentro.

En primer lugar el objeto `AccidentDetection` recibe de parte del objeto `Qualification` la matriz de salida que proviene desde los objetos `TrafficQualification`, `VisualQualification` y `WeatherQualification` (ver tabla 4.2), por lo que estará recibiendo 3 matrices con las calificaciones organizadas por tipo de problema (tráfico, clima o visual).

Posteriormente, el objeto `AccidentDetection` obtiene los valores promedio de la autopista para velocidad y densidad de tráfico para poder comparar los valores promedios contra los valores individuales dentro de cada una de las zonas de la autopista. Si el valor individual es mayor (para el caso de la densidad) o menor (para el caso de la velocidad) en comparación con el valor promedio multiplicado este por un valor *default*, definido tanto para velocidad como para densidad, entonces el sistema puede asumir que existe una posible situación de accidente dentro de una zona. Los valores *default* pueden ser modificados, si fuera necesario, por el administrador. Existe un valor *default* para la densidad y otro valor aparte para la velocidad, que han sido definidos, el primero con el valor de 2 y el segundo también con el valor de 2.

De manera que el sistema vigila las condiciones generales (en promedio) para asumir una “normalidad” dentro de la autopista, si los valores individuales exceden este valor “normal” se puede pensar que existe peligro de encontrarnos frente a un accidente que provoque la disminución notable en la velocidad promedio dentro de una zona y a la vez provoque un incremento grande de congregación de vehículos dentro de una misma zona.

Al mismo tiempo el objeto cuenta con una bandera llamada “*Soft-Detection*” cuya finalidad es la de cambiar la sensibilidad de respuesta para detección de accidentes. Si dicha bandera esta activada, el sistema, además del proceso anteriormente mencionado, verificará si la zona de la que se tiene sospecha esta ocupada y si la zona siguiente a la que se tiene sospecha tiene una ocupación (densidad) de 0. Es decir, esta bandera ayuda a detectar bloqueos totales dentro de la autopista no solamente basándose en valores de velocidad y ocupación promedio. Sea cual sea la condición de la bandera *Soft-Detection*, el sistema, una vez identificado un accidente, procede a etiquetar dicha zona como zona accidentada.

Después de que el sistema ha identificado todas las zonas como accidentes o bloqueos, procede entonces a etiquetar las zonas sobrantes como zonas con problemas de tráfico, con problemas de clima o problemas detectados por la cámara (visuales), basándose para esto en la matrices provenientes de los objetos *TrafficQualification*, *WeatherQualification* y *VisualQualification*. El sistema verifica si dentro de la zona analizada se detectaron problemas de tráfico, problemas climatológicos o problemas detectados por la cámara y si así fue, entonces simplemente procede a etiquetar dicha zona con el problema detectado. Para el caso de las calificaciones de la cámara o visuales, es importante hacer notar que la calificación obtenida por este sensor afecta únicamente a los problemas de visibilidad y humedad dentro de la autopista, por lo que la calificación obtenida por el sistema visual se unirá a la calificación obtenida por el sistema de clima.

Ahora, pudieran existir casos en que los problemas detectados no sean únicos, es decir, que no solo se haya detectado un tipo problema, sino dos problemas juntos en una misma zona. En este caso el sistema etiquetará de manera especial a esta zona para poder incluir a las dos calificaciones, tomará en cuenta el problema detectado y su calificación, por lo que se puede decir que la zona quedaría doblemente calificada. A continuación se muestra un ejemplo de matriz de salida para el objeto *AccidentDetection* en donde se etiquetan a las diferentes zonas de diversas maneras. Después se presenta un resumen de las claves utilizadas por este mismo objeto para denotar las diferentes situaciones posibles. Nota, si la zona presenta a los tres problemas detectados, el sistema sólo tomará en cuenta al tráfico y al clima por tener mayor prioridad.

Matriz de salida ejemplo para el objeto AccidentDetection					
Zonas	Situación	Calificaciones			
Zona 0 (Doble calificación)	-1	550	11	120	21
Zona 1	0	0		0	
Zona 2	2	342		23	

Tabla 4.4 Ejemplo de matriz de salida para el objeto AccidentDetection en donde se muestra una posible salida de este objeto. La matriz muestra un ejemplo de salida doble, un ejemplo de salida general y un ejemplo de salida con problemas de clima.

Valores utilizados por el objeto AccidentDetection en su matriz de salida	
Valor utilizado para describir una situación	Descripción
0	Se describe una situación general o sin problemas detectados.
1	Se describe una situación con dificultades de tráfico.
2	Describe una situación de dificultades climatológicas
4	Sirve para describir zonas con accidentes detectados
-1	Sirve para describir zonas con doble calificación.

Tabla 4.5. Muestra los valores utilizados por el objeto AccidentDetection dentro de la matriz de salida y que sirven para identificar los posibles problemas presentados dentro de la autopista en cada zona.

Por último, el objeto AccidentDetection regresa esta matriz calificada y etiquetada al objeto Qualification, para que este a su vez la pase para poder definir la administración de mensajes para los tableros electrónicos VMS y así poder dar a conocer a lo usuarios de la pista la información obtenida y procesada por el sistema para poder advertirlos, prevenirlos e informarlos sobre distintas situaciones.

4.5.8 Administración de VMS (*Variable Message Sign*)

Una de las principales funciones dentro del sistema es la de poder dar a conocer a los usuarios dentro de la autopista, la información recolectada y procesada así como las condiciones detectadas por los

diferentes sensores, tanto de velocidad como de clima. Como se comentó en capítulos anteriores el sistema cuenta con 3 tipos de tableros electrónicos VMS, el tipo 1 se encontrará principalmente en áreas de descanso o casetas, el tipo II se encontrará a lo largo de toda la autopista, pero por su tamaño y su forma únicamente podrá controlar los límites de velocidad y por último el tipo III se encontrará también a lo largo de toda la autopista pero podrá mostrar mensajes de texto informativos o preventivos. Para poder llevar a cabo esta tarea es necesario contar con los tableros electrónicos (tipo I, II y III) que serán los encargados de comunicar toda la información ya procesada por el sistema, los datos recabados por los sensores que pudieran ser de utilidad o de importancia para los usuarios así como también mostrar mensajes de alerta para posibles escenarios peligrosos o simplemente mostrar mensajes de interés general, todo esto se busca que se logre de la mejor manera posible.

La función de administrar y asignar la información para ser mostrada por cada uno de los tableros electrónicos es realizada principalmente por el objeto llamado `mx.udlap.qualification.ManageVMS`. Dicho objeto se encarga primordialmente de recibir las calificaciones de peligrosidad o de no peligrosidad, por zonas, dentro de la autopista, desde el objeto llamado `mx.udlap.qualification.Qualificaton` para posteriormente elegir qué tipo de mensaje será mostrado en qué tablero, así como el tipo del tablero también será tomado en cuenta para decidir la clase de mensaje que deberá aparecer para informar o prevenir a los usuarios de la autopista.

El proceso comienza al recibir la calificación final por zonas desde el objeto `Qualification`, además previamente el objeto `ManageVMS` lee la configuración y distribución de los tableros electrónicos de mensajes variables (VMS) desde el archivo XML llamado `vms.xml` así como también lee la configuración y distribución de los tableros con sus respectivos tipos desde el mismo archivo XML, ambos procesos se explican con detalle más adelante en el siguiente punto.

Una vez que el objeto `ManageVMS` tiene conocimiento de la distribución física de los tableros así como los tipos de tableros y su posición (dada en kilómetros dentro de la autopista) y también tiene registrado el *pool* de mensajes predefinidos por el administrador del sistema (mensajes generales, de clima, de tráfico y de accidentes) es posible definir qué información aparecerá en qué tipo de tableros y en qué tableros así como toda la información y mensajes que saldrán hacia los usuarios.

En primera instancia este objeto identifica dentro de todos los tableros electrónicos VMS a qué zona pertenece cada tablero y los agrupa por zonas. Al tener organizados todos los VMS, de cualquier tipo, en zonas dentro de la autopista, el siguiente paso consiste en identificar qué mensajes tendrán que aparecer en los tableros tipo III, ya que la información derivada de este proceso servirá de base para los demás tipos de tableros VMS (tipos I y II). Para lograr esto el objeto recurre a la calificación final dada por el objeto *Qualification* de manera que comienza buscando antes que nada las situaciones “normales” o que no merecen una mayor atención, en este proyecto se denominaron estas situaciones como “Generales” y su finalidad es meramente informativa.

Es importante definir las características que contienen los mensajes de cualquier tipo. Los mensajes utilizados dentro del sistema son identificados e individualizados por medio de una clave única que indica de qué tipo de mensaje se trata, se utiliza básicamente la misma nomenclatura explicada en el punto 4.5.7 y en la tabla 4.4 y están dadas de la siguiente manera:

- Los mensajes con la clave 0 son llamados mensajes Generales y pueden ser simplemente información general sobre la autopista, información restrictiva o informativa para los usuarios de la autopista, avisos importantes por parte de los administradores de la misma o cualquier tipo de información que se desea que se lea por los usuarios, siempre y cuando no exista una situación de mayor gravedad detectada automáticamente por el sistema (accidentes).
- Los mensajes con la clave 1 son los que representan a los mensajes que se utilizarán para combatir problemas de vialidad o tráfico. Dichos problemas pueden ser exceso de velocidad, una gran densidad de vehículos o gran flujo de tránsito dentro de algún tramo de la autopista. Estos mensajes principalmente van dirigidos a controlar un problema vial en específico, por ejemplo, el exceso de velocidad.
- Los mensajes clasificados con la clave 2 son los que serán mostrados para atacar a las situaciones de tipo climatológicas. Como los humanos no pueden controlar el clima, la intención de este tipo de mensajes es básicamente informativa y preventiva. Estos mensajes van enfocados a motivar la prevención de los usuarios ante alguna situación climatológica de peligro. Los problemas con el clima que se pueden detectar son: presencia de hielo en la pista,

problemas de visibilidad (neblina o lluvia fuerte), velocidad del viento, temperatura ambiental y la humedad en el ambiente.

- Los mensajes que se identifican con la clave número 4 son los mensajes dedicados a combatir problemas aún más graves como accidentes o alguna situación que impida el flujo de tráfico. Estos problemas son de la mayor gravedad existente y deberán de ser tratados con mayor prioridad que ningún otro. Los mensajes de este tipo contendrán textos principalmente informativos y de advertencia. Los administradores deben de asumir las peores situaciones y deben de ingresar mensajes previniendo a los usuarios del problema al que se enfrentarán más adelante y se les podrá avisar que de ser posible abandonen la autopista para no tener problemas de embotellamientos, así como que den la prioridad y el paso a los vehículos de emergencia.

Todos los mensajes sin importar de qué tipo es contienen una serie de características que ayudan al sistema a tratarlos, organizarlos y asignarlos a cada uno de los tableros electrónicos VMS. Las características comunes a todos los tipos de mensaje son:

- **Text.** Es la característica más importante ya que es la información que aparecerá directamente en los tableros electrónicos VMS. Deberá ser escrito de acuerdo al tipo de problema que se atacará, ya sea vial, de clima, accidentes o simplemente informativo o preventivo.
- **Id.** Es un identificador único e irrepetible para cada mensaje. En sistemas con una gran cantidad de mensajes esta característica es de mucha ayuda al administrador del sistema para encontrar algún mensaje en especial.
- **Type.** Es la clave con la que se identifica el tipo del mensaje. Puede ser 0 para general, 1 para problemas de tráfico, 2 para problemas climatológicos y 4 para accidentes.
- **Base Qualification.** Este dato es en el que el sistema se basará para determinar si el mensaje debe de ser proyectado o no. Es decir, es el mínimo de calificación a partir que el mensaje en cierta zona aparecerá o no. Por ejemplo, si se detecta un problema de visibilidad en la zona 2 que se califica con 500 y nuestro sistema tiene 2 mensajes para el aviso de problemas de visibilidad el primero con Base Qualification de 800 y el segundo con Base Qualification de 400 entonces el sistema desplegará dentro de la zona 2 el mensaje de Base Qualification 400 porque es el que no rebasa a la calificación detectada en dicha

zona.

- **Attacking Problem.** Este dato nos dice además del type, qué problema específico está atacando el mensaje actual. Esta compuesto en su primera posición por el tipo de problema a atacar y en su segunda posición por el problema específico a atacar. Por ejemplo, si nuestro mensaje es de tráfico y ataca a la velocidad entonces su **Attacking Problem** sería 11. En la tabla 4.5 se explican todos los problemas específicos.
- **Description.** Este dato es creado únicamente con la intención de facilitar el uso por parte de los administradores ya que contiene una breve descripción de qué hace el mensaje actual y para qué propósito fue creado, esto con lenguaje común y corriente y sin la necesidad de descifrar el formato XML.

Tipo	Nombre	Problema específico	Nombre	Descripción
0	General	-	-	-
1	Tráfico	1	Flujo	Número de vehiculos que transitan un kilómetro en una hora.
		2	Velocidad	Distancia que viaja un vehículo en un tiempo determinado.
		3	Densidad	Número de vehículos presentes en un kilómetro en un tiempo dado.
2	Clima	1	Temperatura	Temperatura dada en °C en una zona dada.
		2	Humedad	Presencia de partículas de agua en el ambiente, dada en porcentaje (%)
		3	Viento	Velocidad del viento, en km/h
		4	Visibilidad	Capacidad de visión a cierta distancia, dada en km
		5	Hielo	Presencia de hielo o no presencia de hielo
4	Accidentes	-	-	-

Tabla 4.6. Descripción de los problemas específicos manejados por el sistema.

Es importante mencionar que el sistema requiere de por lo menos un mensaje para cada tipo y para cada problema manejado por él. Es decir, por lo menos 1 mensaje general o de tipo 0, por lo menos 3 de tipo de tráfico (velocidad, densidad y flujo), por lo menos 5 para los problemas

climatológicos (visibilidad, humedad, velocidad del viento, temperatura y presencia de hielo) y por último por lo menos un mensaje dedicado al aviso oportuno de accidentes dentro de la autopista.

Además de todos los datos o características que contienen todos los mensajes en general sin importar el tipo, los mensajes de tipo 0 o Generales específicamente presentan dos características extras. Se muestran a continuación:

1. *Admin*. Este dato indica al sistema si este mensaje es un mensaje con características administrativas, es decir, que tendrá una prioridad superior a los tipos 0, 1 y 2. Los administradores deben de tener cuidado con este dato ya que podrían hacer que el sistema no mostrara información de tipo 1 o 2 que pudiera ser importante para los usuarios.
2. *Initial km*. Esta característica nos indica hasta qué kilómetro podrá ser válido el mensaje, es decir, si se requiere dar información, por ejemplo, de reparaciones en el kilómetro 15, entonces sería deseable que los mensajes advirtiendo de dichas reparaciones aparecieran hasta el kilómetro 15 y no más allá ya que serían inútiles después de este punto. Esta característica se encarga de este problema.

Para resumir las características de los mensajes y las características especiales de los mensajes tipo General se muestra la siguiente tabla:

Tipo de mensaje	Características
Tráfico / Clima	<ul style="list-style-type: none"> • <i>Text</i> • <i>Id</i> • <i>Type</i> • <i>Base Qualification</i> • <i>Attacking Problem</i> • <i>Description</i>
General	<ul style="list-style-type: none"> • Todas las características anteriores de Tráfico y Clima más... • <i>Admin</i> • <i>Initial Km</i>

Tabla 4.7. Condensado de las características específicas de los mensajes tipo Generales y las características de los mensajes tipo Tráfico y Clima

Los mensajes tipo General son mucho más maleables dado que están diseñados para mostrar información relevante por parte de los administradores de la autopista. Para esto se le han dado a este tipo de mensajes dos opciones extras además de las que ya tienen los mensajes tipo Tráfico y Clima. La primera de ellas es la llamada `initial_km` y cuya finalidad combinada con la característica `admin` dan un abanico de opciones para controlar la exposición de dichos mensajes ante los usuarios. El sistema siempre buscará primero proyectar todos los mensajes que sean tipo de tipo “administrador” y que tengan límite de kilometraje, esto para recalcar las situaciones en las que existan, por ejemplo, reparaciones dentro de la autopista. En dado caso que no existiera esta combinación de características en el *pool* de mensajes Generales, el sistema proyectará los mensajes con límite de kilometraje.

Pudiera existir el caso en que los administradores no desearan mostrar mensajes con límite de kilómetros, en este caso los administradores del sistema deberán de colocar el valor 0 en el *tag* `initial_km`, al hacer esto, el sistema permitirá aparecer a estos mensajes a lo largo de toda la autopista. Este tipo de mensajes se denominan “mensajes comodines”.

Es de gran importancia mencionar también que, dependiendo del tipo de mensajes existe una prioridad implícita específica que depende del tipo del mismo. Existen 3 tipos de prioridad, los mensajes con menor prioridad pueden ser sobrescritos (si así se requiriera) por el sistema ante la necesidad de mostrar un mensaje con mayor prioridad. Los mensajes con mayor prioridad pueden sobrescribir a cualquier tipo de mensaje excepto a uno de su misma prioridad.

Por orden de prioridad primero aparecerán los mensajes Generales, después de éstos con el nivel 2 de prioridad se encuentran los mensajes de tráfico y clima y por último y con mayor prioridad que cualquier otro se encuentran los mensajes dedicados al aviso de accidentes carreteros. Es importante mencionar que esta prioridad de los mensajes es únicamente implícita y no hay opción a modificarla. También es relevante recalcar que con la característica llamada `admin`, perteneciente únicamente a los anuncios tipo 0 o Generales, estos pueden subir de prioridad para evitar ser sobrescritos por los mensajes con nivel 2 (clima y tráfico), esto con la finalidad de poner mensajes importantes del administrador al alcance de los usuarios bajo casi cualquier circunstancia. Esta opción

debe de ser manejada con mucho cuidado porque el sistema combinará mensajes de problemas tráfico/clima junto con mensajes General que tengan opción “admin” activada. Sin embargo, si el caso se diera, seguirían siendo reemplazados por mensajes de tipo 4, es decir, de accidentes. Teniendo todas estas opciones disponibles los administradores del sistema tendrán a la mano una herramienta que podrá reaccionar automáticamente ante las situaciones inesperadas dadas dentro de la autopista, así como reaccionar ante las situaciones de tráfico cambiantes en cada instante, pero también tendrán el control para poder mostrar mensajes tanto preventivos como informativos a los usuarios en cualquier momento. El objeto `ManageVMS`, al localizar una zona con una situación 0. inmediatamente le asigna a todos los tableros existentes dentro de dicha zona mensajes de tipo 0 o Generales. Los mensajes tipo 0 o generales aparecerán sólo en las zonas en dónde no hay problemas extras (tráfico, clima o accidentes) detectados.

Después el sistema buscará dentro de la autopista problemas de accidentes buscando por problemas tipo 4. Al encontrar problemas de accidentes asignará mensajes preventivos tipo 4 a la zona conflictiva. Después de hacer esto comenzará a asignar mensajes de este mismo tipo hacia atrás cubriendo una distancia de hasta 10 kilómetros antes del accidente para avisar a los usuarios oportunamente. La distancia máxima de difusión de información de accidentes puede ser modificada a la que mejor convenga al administrador. Todos los mensajes con menor prioridad serán reemplazados por los mensajes tipo 4.

Posteriormente el sistema buscará por problemas detectados de tipo 1 y 2, es decir, de tipo de tráfico y clima, respectivamente. Al detectar un problema de cualquiera de estos tipos el sistema automáticamente asignará mensajes atacando a los problemas detectado (para tráfico, velocidad, densidad o flujo y para clima, temperatura, visibilidad, etc.). Es decir, el sistema es más específico y detecta además de qué tipo de problema es, qué problema específico se presenta en cada zona y asigna un mensaje para contrarrestar dicha situación o, en su defecto, para prevenir a los usuarios de dicho problema. La distancia máxima de difusión está definida como 5 kilómetros antes del problema detectado (este valor también puede ser modificado si fuera necesario) para poder avisar oportunamente a los usuarios.

Es importante mencionar que si se dieran problemas de tipo 4 o tipo 1 o 2 dentro de la primera

zona de la autopista, las distancias máximas para cada caso no se pueden cumplir (ya que la primera zona, o zona 0, de la autopista no tiene zonas anteriores) por lo que la difusión se dará únicamente dentro de la misma primera zona o zona 0.

Por último es necesario señalar que se tiene pensado que los tableros VMS tipo III puedan mostrar hasta 2 mensajes, intercalándolos, para así poder mostrar aún más información. La manera en que el sistema aborda esta situación es la siguiente. Se da por hecho que lo ideal es mostrar solamente un sólo mensaje por tablero VMS, pero existen situaciones que se prestan para poder aprovechar los dos lugares disponibles pensados en un tablero VMS. Los únicos mensajes que serán mostrados compartiendo el mismo VMS son los tipo tráfico y tipo clima.

Un caso en que ocurre esta situación es que la matriz entrante, proveniente del objeto *Qualification*, contenga una zona en que se este contemplando una doble calificación (ver tabla 4.4 y subtema 4.5.7). En este caso el sistema asignará los dos lugares disponibles a cada una de las calificaciones que se puedan dar, por ejemplo, velocidad/visibilidad; en el caso del ejemplo el sistema mostrara un mensaje para reducir la velocidad y otro con precaución de la poca visibilidad. La segunda situación en donde se puede dar este caso es que dos zonas de difusión de problemas diferentes se empalmen. Es decir, si la zona 0 (del kilómetro 0 al 10) tiene un problema de visibilidad y la zona 1 (del kilómetro 10 al 15) tiene un problema de velocidad, y tenemos una zona de difusión de 5 kilómetros, entonces el tramo que va del kilómetro 5 al 10 se verá empalmado con dos tipos diferentes de problemas. En este caso el sistema utiliza doble mensaje durante este tramo avisando de las dos situaciones conflictivas que se viven en ambas zonas.

Otra opción en la que se pueda presentar esta situación de doble mensaje dentro de un sólo VMS es en el caso de que en dicha zona existieran mensajes asignados tipo general pero que tengan la opción `admin` activada, en esta situación el sistema utiliza las dos opciones de mensaje dentro de un tablero VMS para mostrar el mensaje informativo o preventivo del administrador y a la vez un problema detectado ya sea de tipo climatológico o de tipo tráfico. Para poder aclarar un poco todas las posibles situaciones, a continuación se muestra una tabla con posibles mensajes mostrados por tableros tipo III en diferentes situaciones.

Tabla 4.8.			
Ejemplos de mensajes mostrados en los tablero tipo III con diferentes opciones posibles.			
Tipo de Mensaje	Situación a atacar (clave)	2 mensajes disponibles en un mismo VMX	
		Mensaje 1	Mensaje 2
General (tipo 0)	General (0)	Respete los límites de velocidad	-
Accidente (tipo 4)	Accidente (4)	PRECAUCIÓN ACCIDENTE GRAVE ADELANTE CARRILES BLOQUEADOS	PRECAUCIÓN AUTOPISTA BLOQUEADA UTILICE PRÓXIMAS SALIDAS
Tráfico (tipo 1)	Exceso de velocidad (12)	DISMINUYA SU VELOCIDAD	-
General con admin (tipo 0) y Clima (tipo 2)	General (0) y Presencia de Hielo (25)	Precio de la próxima caseta \$25	CUIDADO HAY HIELO EN EL ASFALTO
Doble información (tipo -1)	Demasiada Densidad (13) y Poca visibilidad (24)	PRECAUCIÓN CONGESTIÓN UTILICE VÍAS ALTERNAS	PRECAUCIÓN POCA O NULA VISIBILIDAD
General con admin (tipo 0) y Tráfico (tipo 1)	General (0) y Exceso de velocidad (12)	Precaución reparaciones hasta el km 23	DISMINUYA SU VELOCIDAD

En el siguiente diagrama de bloques se muestra cómo el sistema distribuye la información de los mensajes a mostrar zona por zona y VMS por VMS, dependiendo del mensaje electrónico que se desee mostrar.

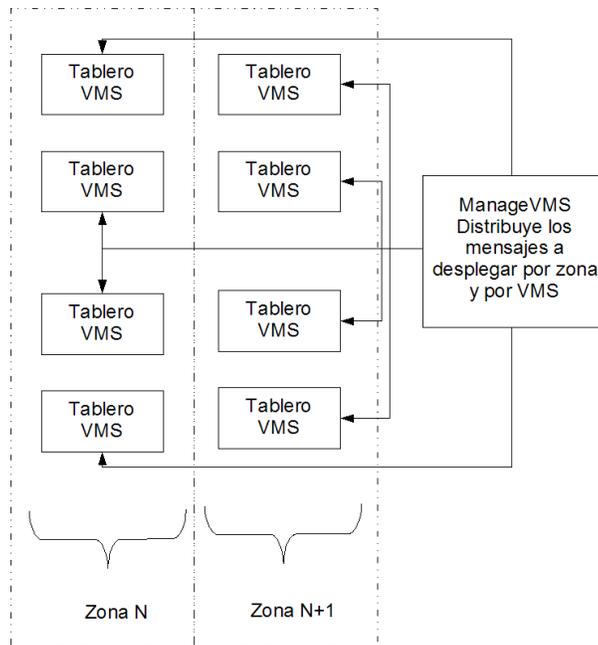


Imagen 4.7. Diagrama de bloques de hardware mostrando cómo la información es diseminada a través de las zonas dentro de la autopista y de los VMS.

Después de que el sistema ha calificado las diferentes situaciones dentro de la autopista, ya habiendo mezclando las diferentes opciones y tomando en cuenta la configuración dada por el administrador, prosigue entonces a definir los valores que se mostrarán dentro de los tableros tipo I y tipo II. Para este tipo de tableros no existen mensajes predefinidos por los administradores ya que la información mostrada en ellos se deriva de los datos ya procesados con anterioridad y para el caso de los tableros tipo I, los mensajes más importantes ya mostrados en los tableros tipo III.

Para el primer tipo de tableros, tableros tipo I, al ser colocados estos en áreas de descanso o casetas de cuota, la información mostrada será un resumen de lo que está adelante del usuario al momento de encontrarse con este tipo de tableros. A diferencia de los tableros tipo III en donde la información que se muestra es la correspondiente únicamente a la zona a la que pertenece x tablero, la información mostrada en los tableros tipo I será el resumen de 1 o varias zonas que estén dentro de los 10 kilómetros próximos (cantidad modificable) a dicho tablero. Este tablero muestra información resumida de temperatura, visibilidad, humedad y si hay o no presencia de hielo dentro de los próximos 10 kilómetros. Este resumen se obtiene calculando el promedio de la temperatura, visibilidad y humedad dentro de las próximas zonas que abarquen los 10 kilómetros.

Además del resumen climatológico, se muestra también, en caso de que existiera alguno, un mensaje de advertencia de accidente; si no hubiera accidentes, se mostrara un mensaje general calificado con la bandera de administrador, si no existiera tampoco, el sistema proyectará un mensaje con kilometraje definido (opción `initial_km`). En la última de las circunstancias que no existiera ninguna de las opciones anteriormente explicadas, entonces el sistema recurre a mostrar simplemente un mensaje cualquiera de tipo general. En la siguiente tabla se muestra un par de ejemplos de salida para los tableros tipo I.

Tabla 4.9. Ejemplos de salida para tableros tipo I	
Ejemplo 1	<p>PRECAUCIÓN AUTOPISTA BLOQUEADA</p> <p>Resumen de clima: Temperatura: 24°C Visibilidad: 2 kilómetros No se detecta presencia de hielo</p>
Ejemplo 2	<p>Reparaciones desde el km 20 hasta el km 23</p> <p>Resumen de clima: Temperatura: 0°C Visibilidad 10 kilómetros ¡Se detecta presencia de hielo!</p>

Finalmente, los tableros tipo II están dedicado únicamente a controlar la velocidad dentro de la autopista. Para esto el sistema detectará la velocidad promedio dentro de cada zona. Después debe de tomar en cuenta el límite de seguridad especificado por el administrador del sistema dentro de el archivo de configuración llamado `constraints.xml`. Una vez hecho lo anterior, el sistema compara la calificación dada por el objeto `mx.udlap.qualification.Qualification` para cada zona, esto con la finalidad de obtener un límite de velocidad para dicha zona.

Si el sistema detecta una situación combinada o doble, de problemas detectados de tráfico con problemas detectados de clima, entonces el sistema calculará la velocidad límite permitida en un porcentaje restado a la velocidad de seguridad definida por el administrador. Es decir, si la velocidad de seguridad dada por el administrador es de 110 kilómetros por hora, el sistema definirá la velocidad máxima permitida en un 36% menos (18% + 18%), para este caso, la velocidad máxima permitida sería de 70 kilómetros por hora. Lo mismo sucederá si el sistema detecta dentro de la zona en cuestión

problemas de accidentes.

En caso de que el sistema detecte únicamente problemas de velocidad, definirá la velocidad máxima permitida en un 18% menos de la velocidad de seguridad definida por los administradores del sistema. Por lo que la velocidad permitida sería de 90 kilómetros por hora. Por último si el sistema no detecta ningún tipo de problema dentro de la zona, la velocidad límite será exactamente igual a la definida como velocidad de seguridad por los administradores. De esta manera y manejando únicamente los límites de velocidad dinámicamente se podría lograr un mejor flujo de tráfico sin dejar a un lado las características de la zona y los posibles problemas que pudieran suceder, protegiendo de esta manera a los usuarios y sin comprometer en gran manera la fluidez de la autopista.

Es importante recordar que cada zona tiene su velocidad de seguridad definida por el administrador, por lo que el sistema provee una flexibilidad de acuerdo a las diferencias dadas entre zonas. Nota: El porcentaje definido de 18% de variación de la velocidad de seguridad se podrá modificar si el administrador considera otro porcentaje más adecuado a las necesidades del sistema. En la siguiente tabla se muestran las tres posibilidades para este tipo de tablero.

Tabla 4.10. Ejemplos posibles para un tablero tipo II dada una velocidad de seguridad de 110 km/h y un porcentaje definido del 18%		
Problema detectado	Operación	Límite de velocidad calculado
Exceso de velocidad (problema 12)	110-18%	90 km/h
Congestionamiento (problema 13) y Poca visibilidad (problema 24)	110-(18% x 2)	70 km/h
General (problema 0)	110-0%	110 km/h

Es importante recalcar que los tableros tipo II, a diferencia de los otros dos, no tienen una distancia definida de difusión de su información ya que únicamente actúan dentro de sus correspondientes zonas, así la información mostrada será de uso inmediato a la zona en la que se muestra y no en otras zonas.

De esta manera el administrador tiene muchas opciones diferentes para mostrar a los usuarios

toda la información que necesiten y a la vez el sistema tiene varias maneras de hacer llegar la información a los usuarios. El sistema muestra una gran flexibilidad en cuanto a opciones de respuesta y opciones frente a diversas situaciones, lo anterior lo hace un sistema complejo en cuanto a tipo de respuestas y tipo de reacciones a diferentes situaciones.

Para poder haber calificado todas las situaciones dentro de la autopista, el objeto `ManageVMS`, dentro de todas las zonas y después de haber elegido cada salida, para cada tablero VMS, para cada zona y para cada tipo de tablero VMS, el sistema requiere de los objetos llamados `mx.udlap.vms.VMS` y `mx.udlap.vms.MessageSign`. El primero se encarga de describir un mensaje cualquiera para cualquier tipo de tablero VMS mientras que el segundo describe un tablero VMS físicamente, ya que contiene el kilómetro en donde se encuentra dicho tablero, la zona a la que pertenece, el tipo de tablero, entre otras cosas, y a su vez, contiene un arreglo de objetos de tipo VMS, dependiendo de cuantos mensajes a la vez se puedan desplegar a la vez, en este caso 2, los cuales describen el mensaje o la información que debe de mostrar el tablero. Se ha dejado modificable la opción de más de 2 o menos de 2 mensajes a la vez para los tablero electrónicos VMS, con la finalidad de facilitar el trabajo en futuras aplicaciones del sistema, sin embargo, el presente proyecto se ha definido solamente con la opción de 2 mensajes compartiendo un tablero electrónico VMS.

4.5.9 Entrada/salida de archivos

El sistema contiene varios tipos de entrada y salida de archivos. Desde archivos de texto, archivos XML y objetos escritos en disco para uso en web, el sistema contiene una variedad de interacción con diferentes tipos de archivos. Los archivos de entrada principalmente ayudan a configurar el funcionamiento del sistema así como controlar algunas variables que afectan al desempeño del mismo. En la siguiente tabla se muestra un resumen de los archivos de texto utilizados como entrada al sistema y una breve descripción de su finalidad. Nota: Todos los archivos de entrada de texto del sistema se encuentran dentro de la carpeta `mx.udlap.io`.

Tabla 4.11. Descripción de los archivos de texto de entrada al sistema	
Nombre del archivo	Descripción
<code>zonas.txt</code>	Este archivo es utilizado únicamente por el simulador de climas <code>WeatherSimulator</code> y por <code>Core</code> . Su finalidad es la de describir las zonas dentro de la autopista
<code>sensores.txt</code>	Este archivo es utilizado por el simulador de tráfico <code>CarSimulator</code> y por el administrador de mensajes <code>ManageVMS</code> . Su finalidad es la de describir la fisonomía de la autopista en kilómetros totales y la posición de los sensores de tráfico dada en los kilómetros en donde se encuentran los mismos.
<code>input_files</code>	Son utilizados por los lectores de archivos para el módulo de visión en web <code>ReadHighwayWeatherObject</code> . Se utiliza como un archivo en donde se colocan todas las rutas de objetos a leer/escribir para uso en el módulo web.

El objeto encargado de leer archivos de texto es llamado `mx.udlap.io.ReadInputFile` y es principalmente de uso general, ya que cualquier módulo del sistema que desee leer cualquier archivo de texto simplemente llama a este objeto dándole la ruta del archivo a leer y este objeto le regresará en una sola cadena todo el contenido del archivo leído. Ver punto 4.5.1 y 4.5.2.

En el caso de los archivos de salida con formato de texto, el objeto encargado de escribir cualquier tipo de archivo de texto es llamado `mx.udlap.io.WriteOutputFile` el cual es también de uso general ya que cualquier módulo del sistema puede llamarlo cuando sea necesario, mandándole solamente una cadena de texto y la ruta a dónde escribir. El archivo se encargará de crear el archivo y en caso de que el archivo ya este creado, existe una opción para elegir si se desea sobrescribir por completo el archivo con la nueva información o solamente se desea agregar la nueva información al final del archivo ya existente.

El archivo de texto de salida del sistema es llamado `log.txt` y es escrito directamente por el objeto `mx.udlap.io.WriteGeneralLog` que utiliza la información obtenida desde el objeto `mx.udlap.system.Core`. El archivo `log` contiene todos los cambios detectados por el sistema. Muestra los cambios de clima detectados con la hora y fecha en que se detectó, así como las condiciones detectadas dentro de cada una de las zonas. También muestra los cambios detectados

dentro de la autopista con la hora y fecha así como una descripción de los valores promedio dentro de la autopista, el conteo de vehículos dentro de esta y los valores específicos detectados dentro de cada una de las zonas. Ver punto 4.5.3.

4.5.10 Archivos XML

El sistema utiliza archivos XML como una forma de tener datos de configuración de una manera semi-ordenada y a la vez que tengan un formato entendible a los usuarios administradores para que sea más fácil modificar estos archivos de una manera rápida, segura y entendible y sin tener conocimiento alguno sobre algún sistema aparte como una Base de Datos.

Existen únicamente dos archivos XML que sirven como archivos de configuración para el sistema. Estos archivos son:

- `constraints.xml`. Este archivo sirve para definir situación por situación los límites permitidos o límites de seguridad definidos tanto para tráfico como para clima. Este archivo debe de ser manejado por el administrador y se deberá definir una calificación con la que se le dará cierta importancia a alguna situación que pueda ocurrir dentro de la carretera. Ver punto 4.5.4
- `vms.xml`. Este archivo es en el que el administrador del sistema plasma la situación de distribución de los tableros electrónicos VMS y además muestra los posibles mensajes de texto a aparecer en dichos tableros. Estos mensajes están organizados por tipo y situación lo que facilita su administración y ayuda al usuario ingresar cuantos mensajes requiera para el sistema. Ver punto 4.5.8.

4.5.11 Visualización en web

La última parte del sistema consiste en poder simular una salida de tableros electrónicos VMS así como mostrar interfaces especiales para los administradores y para los usuarios normales. En la salida

simulada de tableros electrónicos VMS se muestran las salidas de todos los tableros VMS existentes dentro de la autopista organizados por tipo (I, II y III). La página de información para el usuario administrador muestra toda la información necesaria que existe dentro de la autopista y por último la página para usuarios comunes muestra la información que se podría necesitar antes de ingresar a una carretera para poder monitorear la situación prevaleciente dentro de la autopista.

Para poder lograr una visualización desde un navegador web, el sistema requiere de un contenedor/servidor en donde colocar los archivos y las páginas que se mostrarán por la red. El contenedor utilizado en este caso es el Jakarta Tomcat en su versión 5.5.9. Como primer paso es necesario escribir en disco los objetos que describen la situación vivida dentro de la autopista. Estos objetos son `mx.udlap.observer.HighwayStatus`, `mx.udlap.observer.Weather` y `java.util.Vector`, conteniendo este último objetos de tipo `mx.udlap.vms.MessageSign`, así los dos primeros objetos son tomados desde el objeto `Core` y se escriben a disco por medio del objeto `mx.udlap.io.WriteHighwayWeatherObject`; el `Vector` es formado a partir de la salida final del objeto `ManageVMS` y contiene una colección de objetos `MessageSign`, cada uno de estos representando a un tablero VMS existente dentro de la autopista, este objeto es escrito a disco por medio del objeto llamado `mx.udlap.io.WriteOutputVMSInfo`. La localización del sitio a dónde escribir estos objetos se especifica en el archivo de texto llamado `input_files` y esta ubicación deberá estar siempre dentro de el contenedor para facilitar la salida a web.

Una vez escritos los objetos con toda la información que describe la situación actual de la autopista, entonces los lectores que se encuentran dentro del contenedor, pero que forman parte también del sistema, se encargan de leer todos los objetos para poder mostrarlos en una forma completa, en caso de la interfaz para administradores, o condensada, para el caso de los usuarios normales. La interfaz de salida web para la información de tableros VMS y que simula la salida real a tableros de la autopista forma parte del mismo sistema pero no tiene preferencia de usuarios, es decir, no tiene restricciones de ningún tipo. El objeto encargado de leer a los objetos descriptores de tráfico desde el disco es `mx.udlap.web.io.ReadHighwayConfiguration`, para leer la situación de clima es el objeto llamado `mx.udlap.web.ioReadHighwayWeatherObject` y finalmente para poder leer toda la salida de los diferentes tipos de tableros VMS dentro de la autopista se encarga el objeto `mx.udlap.web.io.ReadVMSOutput`.