

Simulación de Eventos Discretos

Trabajo práctico Nro 1

Grupo 5:

Matías Brunstein Macri LU:348/99

Empalme AU1 - Dellepiane

Introducción

El objetivo de este trabajo es modelar y simular una versión simplificada del empalme entre la Autopista 1 (Perito Moreno) y la Avenida Dellepiane en el sentido Capital Federal. El objetivo de la simulación es estudiar cómo los cambios en la estructura de las autopistas (longitud de las vías y cantidad de carriles por vías) y la cantidad de cabinas de peaje habilitadas influyen al flujo de vehículos.

Parte 1: Descripción y análisis del sistema real

El sistema real es mucho más complejo que el modelo simulado, pero las simplificaciones realizadas tienen un mínimo impacto en el resultado final. El siguiente esquema representa el empalme de las autopistas.

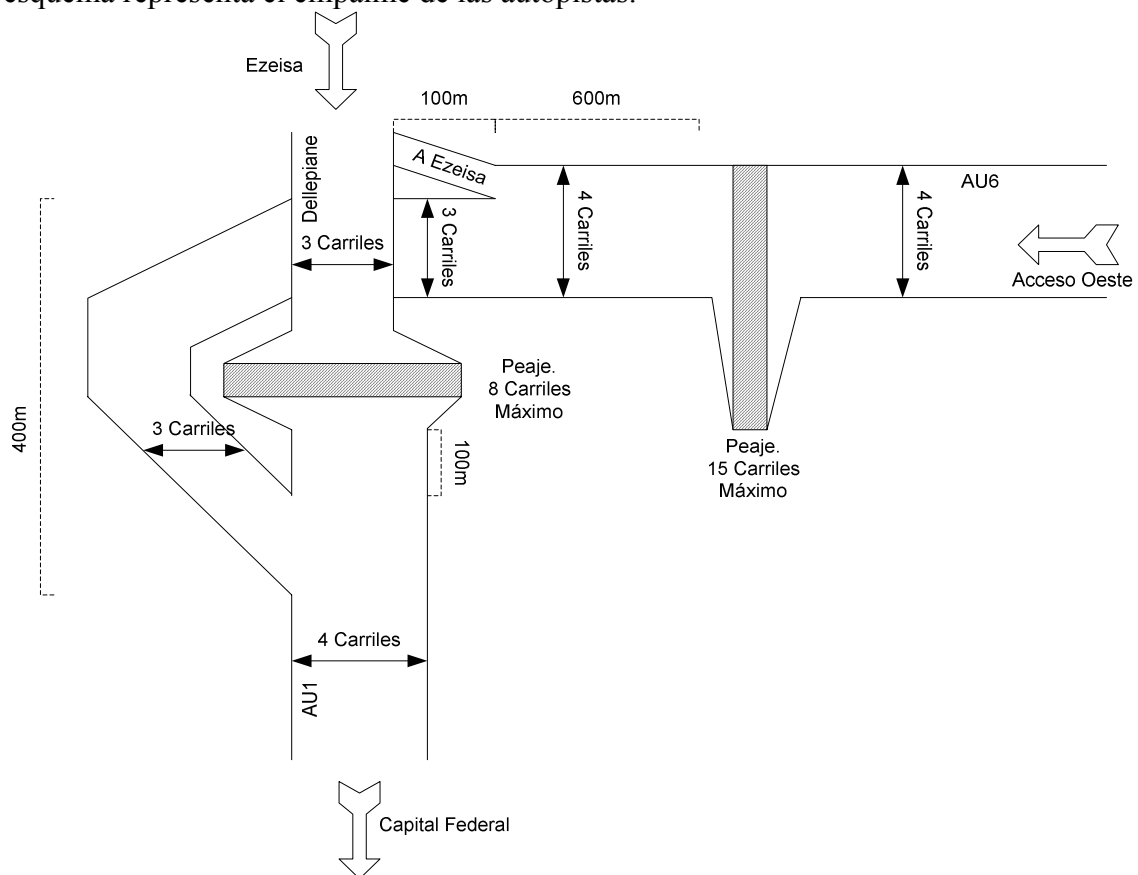


Figura 1: Empalme AU6 – Dellepiane

Los vehículos ingresan al sistema por las dos entradas: desde el Acceso Oeste, por la Autopista 6, y desde Ezeisa por la Avenida Dellepiane. Luego de los peajes, los vehículos convergen en la Autopista 1 (25 de Mayo).

Objetivos de la simulación

El objetivo de la simulación es estudiar cómo los cambios en la estructura de las autopistas (longitud de las vías y cantidad de carriles por vías) y la cantidad de cabinas de peaje habilitadas influyen al flujo de vehículos.

Para esto, el modelo debe representar las siguientes variables de estado:

- *Longitud de las vías*: Es un parámetro del modelo y permite calcular el tiempo que toma a un vehículo llegar de un extremo a otro de una vía y la capacidad de dicha vía.
- *Velocidad máxima*: La velocidad máxima que un vehículo puede desarrollar en una vía. También es un parámetro y permite calcular el tiempo que toma a un vehículo llegar de un extremo a otro de una vía.
- *Carriles por vía*: Es el parámetro más importante de cada vía. Más adelante se describe como interactúa este parámetro con el resto del modelo.
- *Cantidad de autos en cada vía*: Permite medir la congestión de las vías.
- *Cantidad de cabinas habilitadas por peaje*.

Simplificaciones sobre el sistema real

Estas son las principales simplificaciones que se realizaron sobre el sistema real.

- Se considera que todos los vehículos son del mismo tamaño.
- Se considera que todos los vehículos utilizan cabinas de peaje de pago manual.
- A lo largo de las *vías*, los vehículos se desplazan a la máxima velocidad permitida (si hay espacio suficiente).
- Los vehículos no cambian de carril a lo largo de las vías.
- No se consideró el desvío hacia Ezeisa después del peaje de la AU6.

Descripción general del modelo

El modelo consta de 4 niveles: un modelo acoplado TOP que reúne los tres modelos acoplados principales, un segundo nivel formado por dichos 3 modelos, un tercer nivel que combina modelos acoplados con modelos atómicos y un último nivel formado por los modelos atómicos más simples.

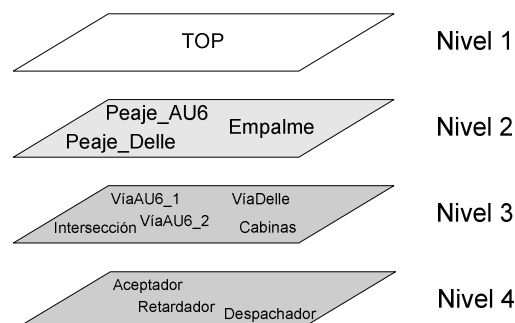


Figura 2: Niveles del modelo

La idea general es modelar por separado ambos peajes (Peaje_AU6 y Peaje_Delle) y el empalme. Al acoplar estos tres modelos se obtiene el TOP. A su vez,

cada uno de estos modelos está conformado por varios modelos atómicos. A continuación se especifica cada uno de los modelos involucrados.

Parte 2: Especificación del modelo

Especificación de los modelos acoplados

Modelo TOP

Este es el modelo principal y es el que integra a todos los submodelos. El comportamiento del modelo TOP precisamente el comportamiento del sistema que se desea estudiar.

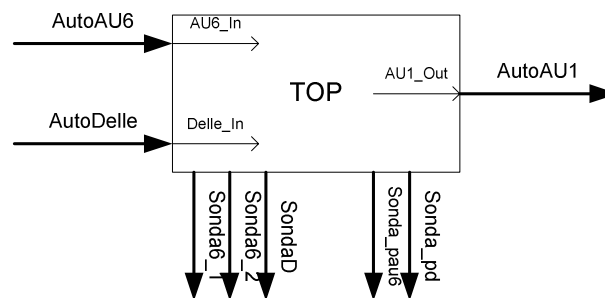


Figura 3: Modelo TOP

El modelo posee 2 puertos de entrada y uno de salida:

- El puerto **AutoAU6** representa el ingreso de vehículos al sistema desde la Autopista Perito Moreno. Cada vez que llegan autos se produce el evento externo *AU6_In*. El valor del mensaje que ingresan por este puerto representa la cantidad de vehículos que ingresan al sistema en ese instante de tiempo.
- El puerto **AutoDelle** representa el ingreso de vehículos al sistema desde la Avenida Dellepiane. Cada vez que llegan autos se produce el evento externo *Delle_In*. El valor del mensaje que ingresan por este puerto representa la cantidad de vehículos que ingresan al sistema en ese instante de tiempo.
- Además, por cada valor que se desee medir, se puede agregar un puerto de salida conectado a la salida **Sonda** de cualquiera de los componentes.

A su vez posee un puerto externo llamado **AutoAU1** que representa los autos que salen del sistema. El valor del mensaje que ingresan por este puerto representa la cantidad de vehículos que salen del sistema en ese instante de tiempo.

El modelo TOP es un modelo acoplado que está conformado por tres submodelos también acoplados. El siguiente diagrama muestra el esquema de acoplamiento de estos modelos:

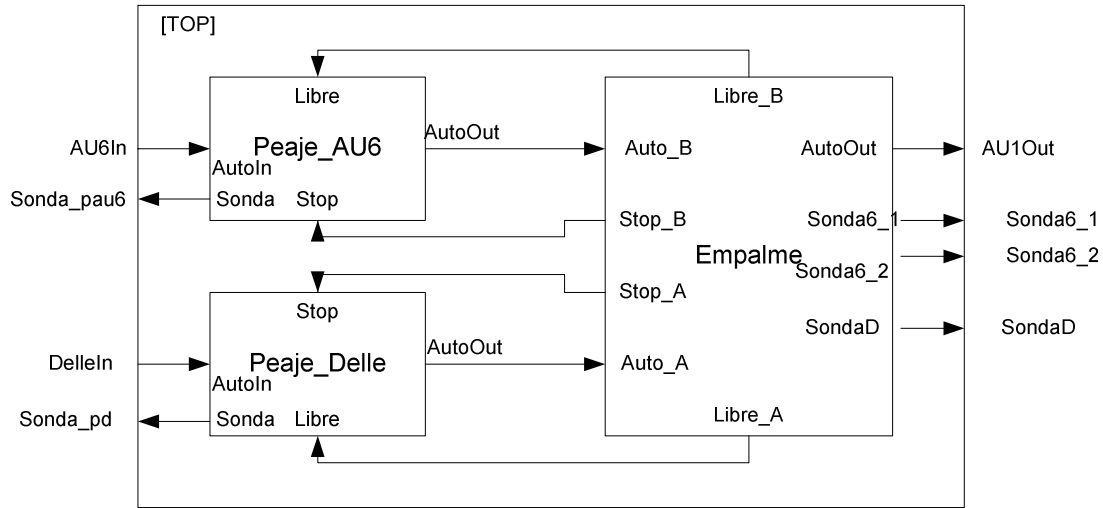


Figura 4: Esquema de acoplamiento del modelo TOP

Especificación formal del modelo

$$Top = \langle X, Y, \{Peaje_AU6, Peaje_Delle, Empalme\}, I_i, Z_{ij} \rangle$$

Donde:

$$X = \{AU6In, DelleIn\}$$

$$Y = \{Sonda_pau6, Sonda_pd, Sonda6_1, Sonda6_2, SondaD, AU1Out\}$$

$$I(Peaje_AU6) = \{self, Empalme\}$$

$$I(Peaje_Delle) = \{self, Empalme\}$$

$$I(Empalme) = \{self, Peaje_AU6, Peaje_Delle\}$$

$$I(self) = \{Peaje_AU6, Peaje_Delle\}$$

$$Z = \{(self.AU6In \rightarrow Peaje_AU6.AutoIn), (self.DelleIn \rightarrow Peaje_Delle.AutoIn), \\ (Peaje_AU6.Sonda \rightarrow self.Sonda_PAU6), (Peaje_Delle.Sonda \rightarrow self.Sonda_pd), \\ (Peaje_AU6.AutoOut \rightarrow Empalme.Auto_B), \\ (Peaje_Delle.AutoOut \rightarrow Empalme.Auto_A), \\ (Empalme.Stop_A \rightarrow Peaje_Delle.Stop), (Empalme.Stop_B \rightarrow Peaje_AU6.Stop), \\ (Empalme.Libres_A \rightarrow Peaje_Delle.Libres), (Empalme.Libres_B \rightarrow Peaje_AU6.Libres), \\ (Empalme.Sonda6_1 \rightarrow self.Sonda6_1), (Empalme.Sonda6_2 \rightarrow self.Sonda6_2), \\ (Empalme.SondaD \rightarrow self.SondaD)\}$$

$$Select: Peaje_Delle > Peaje_AU6 > Empalme$$

Modelo Peaje_x

Los modelos **Peaje_AU6** y **Peaje_Delle** son conceptualmente idénticos, sólo varían los parámetros de configuración de los modelos atómicos que los conforman (de no se por una limitación de la herramienta de simulación, el modelo podría especificarse una sola vez en el archivo de configuración). Por lo tanto podemos decir que ambos pertenecen a la clase **Peaje** y que, pese a ser modelos acoplados, poseen parámetros de configuración.

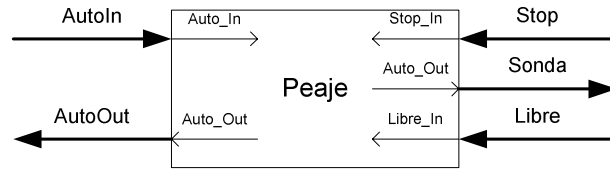


Figura 5: Modelo Peaje

El modelo posee 1 puertos de salida que representa los autos que salen del peaje y 3 puertos de entrada:

- El puerto **AutoIn** representa el ingreso de vehículos al modelo. Cada vez que llegan autos se produce el evento externo *Auto_In*. El valor del mensaje que ingresan por este puerto representa la cantidad de vehículos que ingresan al modelo en ese instante de tiempo.
- El puerto **Stop** (genera el evento externo *Stop_in*) indica que el modelo receptor (conectado a **AutoOut**) no tiene capacidad para recibir más vehículos. El valor del mensaje que ingresan por este puerto representa la cantidad de vehículos que no han podido ingresar al modelo.
- El puerto **Libre** (genera el evento externo *Libre_in*) indica que el modelo receptor (conectado a **AutoOut**) tiene nuevamente capacidad para recibir vehículos.

Los parámetros de configuración de este modelo son los siguientes:

- *Capacidad de la cola*: Indica cuál es la máxima cantidad de vehículos que pueden aguardar en la cola del peaje.
- *Tiempo de atención*: El tiempo que le toma a un vehículo ingresar a una cabina, pagar y salir del peaje.

El siguiente diagrama muestra el esquema de acoplamiento de los submodelos que conforman al modelo **Peaje**:

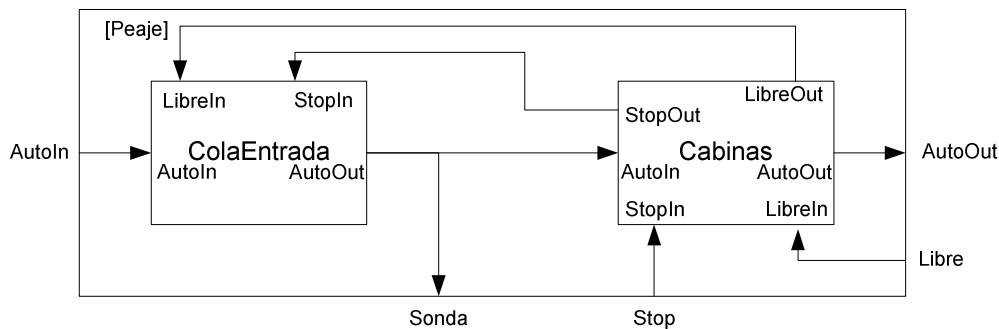


Figura 6: Esquema de acoplamiento del modelo Peaje

Especificación formal del modelo

$$Top = \langle X, Y, \{ColaEntrada, Cabinas\}, I_i, Z_{ij} \rangle$$

Donde:

$$X = \{AutoIn, Stop, Libre\}$$

$$Y = \{Sonda, AutoOut\}$$

$$I(ColaEntrada) = \{Cabinas, self\}$$

$$I(Cabinas) = \{self, ColaEntrada\}$$

$$I(self) = \{Cabinas, ColaEntrada\}$$

$$Z = \{(self.AutoIn \rightarrow ColaEntrada.AutoIn), (self.Libre \rightarrow Cabinas.LibreIn), (self.Stop \rightarrow Cabinas.StopIn), (ColaEntrada.AutoOut \rightarrow \{self.Sonda, Cabinas.AutoIn\}), (Cabinas.AutoOut \rightarrow self.AutoOut), (Cabinas.StopOut \rightarrow ColaEntrada.StopIn), (Cabinas.LibreOut \rightarrow ColaEntrada.LibreIn)\}$$

Select: ColaEntrada > Cabinas

Modelo Empalme

Este es quizá el modelo más complicado. Representa la zona de empalme comprendida entre la salida de ambos peajes y la Autopista 1.

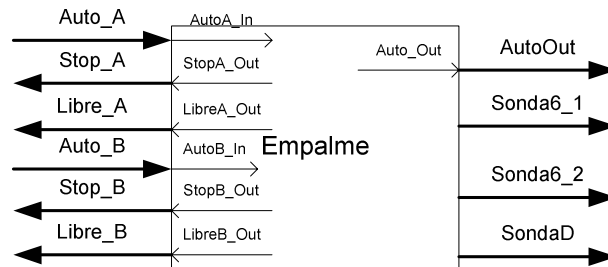


Figura 7: Modelo Empalme

Los puertos de este modelo son:

- **AutoA y AutoB:** Representan los vehículos que llegan al empalme desde ambos orígenes.
- **StopA y StopB:** Envía un mensaje indicando que no pueden ingresar más vehículos desde el origen indicado. El valor del mensaje representa la cantidad de vehículos que no han podido ingresar.
- **LibreA y LibreB:** Envía un mensaje indicando que ya pueden volver a entrar vehículos desde el origen indicado.

El siguiente diagrama muestra el esquema de acoplamiento de los submodelos que conforman al modelo **Empalme**:

Modelos Vía y Cabinas

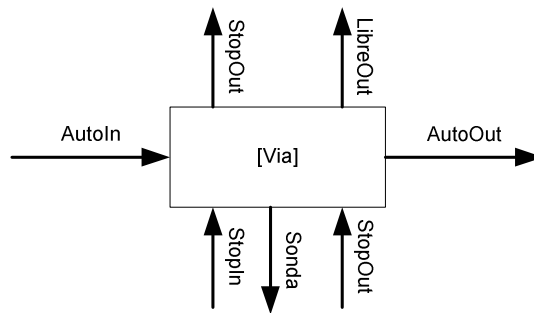


Figura 9: Modelo Vía

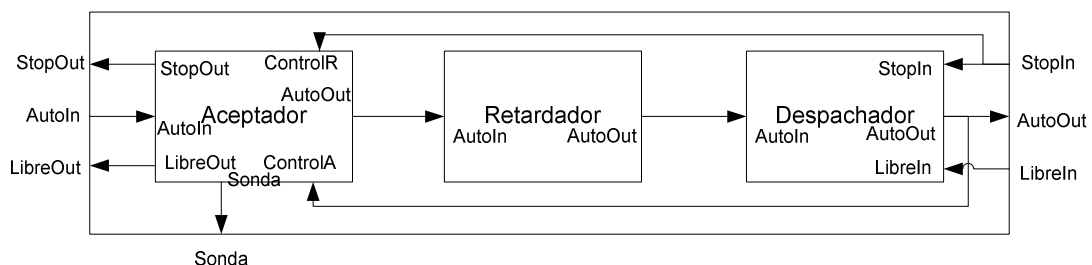
Este modelo representa una vía (o tramo) de la autopista. Mientras hay lugar disponible, los vehículos ingresan a la vía y salen de ella un tiempo después. Dicho tiempo representa la relación entre el largo de la vía y la velocidad máxima.

Las cabinas de las estaciones de peaje pueden modelarse de la misma manera, de modo que se utilizará una copia de este modelo.

Los puertos de este modelo son:

- **AutoIn:** Representan los vehículos que ingresan a la vía.
- **StopIn:** Por este puerto se recibe una señal que indica al modelo que no debe enviar más vehículos. También recibe la cantidad de autos que no pudieron ser aceptados por el receptor.
- **LibreIn:** Por este puerto recibe una señal que le indica al modelo que puede reanudar el envío de vehículos.
- **StopOut:** Por este puerto se envía una señal que indica al modelo emisor que no debe enviar más vehículos. También envía la cantidad de autos que no pudieron ser aceptados por falta de espacio.
- **LibreOut:** Por este puerto recibe una señal que le indica al modelo emisor que puede reanudar el envío de vehículos.
- **Sonda:** Este puerto genera periódicamente salidas informando la cantidad de vehículos en la vía.

El siguiente diagrama muestra el esquema de acoplamiento de los submodelos que conforman al modelo **Vía**:



Especificación formal del modelo

$$Top = \langle X, Y, \{Aceptador, Retardador, Despachador\}, I_i, Z_{ij} \rangle$$

Donde:

$$X = \{AutoIn, StopIn, LibreIn\}$$

$$Y = \{Sonda, AutoOut, StopOut, LibreOut\}$$

$$I(self) = \{Aceptador, Despachador\}$$

$$I(Aceptador) = \{self, Retardador\}$$

$$I(Despachador) = \{self, Aceptador\}$$

$$I(Retardador) = \{Despachador\}$$

$$Z = \{(self.AutoIn \rightarrow Aceptador.AutoIn), (self.StopIn \rightarrow \{Despachador.StopIn, Aceptador.ControlR\}), (self.LibreIn \rightarrow Despachador.LibreOut), (Aceptador.Sonda \rightarrow Self.Sonda), (Aceptador.StopOut \rightarrow Self.StopOut), (Aceptador.LibreOut \rightarrow Self.LibreOut), (Aceptador.AutoOut \rightarrow Retardador.AutoIn), (Retardador.AutoOut \rightarrow Despachador.AutoIn), (Despachador.AutoOut \rightarrow \{self.AutoOut, Aceptador.ControlA\})\}$$

Select: Aceptador > Retardador > Despachador

Especificación de los modelos atómicos

En esta sección, se utilizan FSMs para representar estados, transiciones internas y externas, scheduling de eventos internos y la función λ . La siguiente es la sintaxis utilizada:

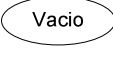
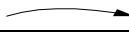
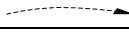
	Estado. Si un estado no tiene ninguna transición interna que salga de él, significa que dicho estado se encontrará activo hasta que se produzca una transición externa.
	Transición por evento externo
	Transición por evento interno
Event(n)	Evento externo, 'n' representa el valor del mensaje
$\lambda = \text{Event}(n)$	Salida generada ante un evento interno, 'n' representa el valor del mensaje.
$\{q = \alpha\}$	Asignación de un valor a una variable de estado
$[q > \alpha]$	Condición para que se produzca una transición. Ante un evento interno, siempre se produce una transición, pero el estado de destino depende de la condición.
$\langle t_c = \alpha \rangle$	Scheduling de un evento interno. La expresión t_c a la izquierda del igual representa la función de avance de tiempo y a la derecha el tiempo transcurrido desde que el evento se "schedulingó".

Tabla 1: Sintaxis de las FSMs

Modelo Cola

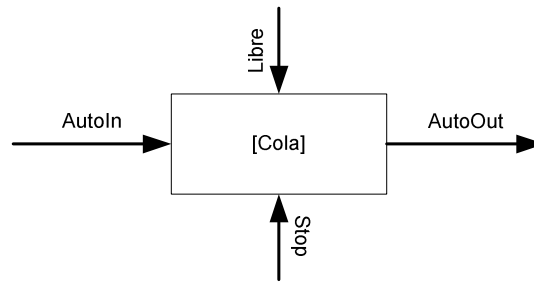


Figura 10: Modelo Cola

Este modelo atómico representa la cola de vehículos que se forma frente a un peaje y posee dos parámetros de configuración:

- *Capacidad*: La cantidad máxima de vehículos que pueden estar esperando para pagar el peaje. Este no es un parámetro que resulte relevante a los objetivos de la simulación, pero es importante para evitar problemas de memoria al momento de correr la simulación.
- *Carriles*: La cantidad de carriles de la cola. Debe coincidir con la cantidad de cabinas habilitadas en el peaje.
- *Preparacion*: Tiempo de preparación.

Los puertos **AutoIn** y **AutoOut** representan la entrada y salida de vehículos a la cola. Al llegar un mensaje a estos puertos, se generan eventos externos con el mismo nombre. El valor del mensaje recibido indica la cantidad de vehículos que entran o salen de la cola.

Los puertos **Stop** y **Libre** permiten manejar el flujo de salida de vehículos. Cuando el receptor ha agotado su capacidad para recibir vehículos, envía un mensaje con valor mayor que 0 hacia el puerto **Stop**; El valor de este mensaje indica la cantidad de vehículos que fueron “rechazados” por el receptor. El puerto **Libre** le indica a la cola que puede continuar enviando vehículos.

Variables de estado

Las variables de estado del modelo son las siguientes:

- q : Cantidad de vehículos en la cola.
- E : Que representa el estado representado en la FSM.:
 - *Vacio*: No hay autos en la cola
 - *Circulando*: Los autos de la cola pueden salir
 - *Parado*: No hay espacio en el modelo destino

Constantes

Las constantes de este modelo (inicializadas mediante los parámetros de configuración) son:

- M : Capacidad Máxima de la cola.
- C : Cantidad de carriles.
- ε : Tiempo que le toma a un vehículo salir de la cola

Transición entre estados

La siguiente FSM muestra las funciones de transición internas y externas.

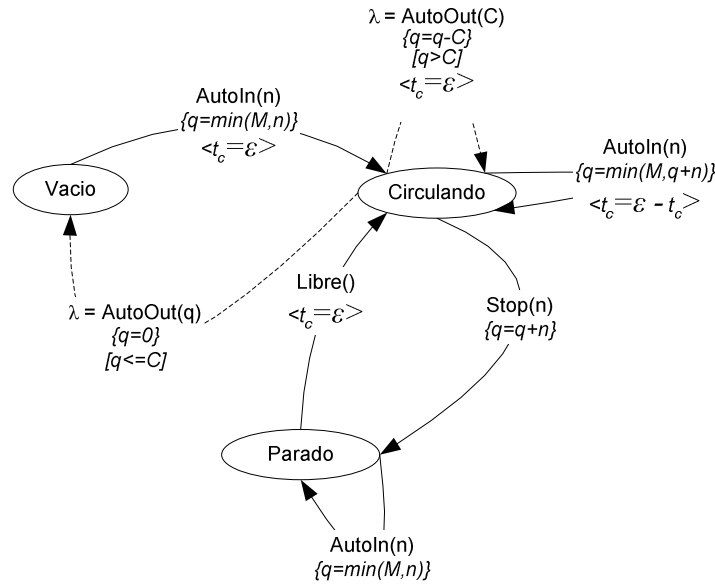


Figura 11: FSM del modelo Cola

Especificación formal del modelo

$$Cola = \langle X, S, Y, \delta_{int}, \delta_{ext}, \lambda, t_a \rangle$$

Donde

$$X = \{autoIn(n), stop(n), libre\}$$

$$Y = \{autoOut(n)\}$$

$$S = \langle E, q \rangle \text{ con } E \in \{\text{vacío}, \text{parado}, \text{circulando}\} \text{ y } q \in \mathbb{N}$$

$$\delta_{int}(S) = \begin{cases} \langle \text{circulando}, q - C \rangle & \text{si } q > C \\ \langle \text{vacío}, 0 \rangle & \text{si } q \leq C \end{cases}$$

$$\delta_{ext}(S, e, x) = \begin{cases} \langle \text{circulando}, \min(n, M) \rangle & \text{si } E = \text{vacío} \wedge x = \text{autoIn}(n) \\ \langle \text{parado}, n \rangle & \text{si } E = \text{vacío} \wedge x = \text{stop}(n) \\ \langle \text{parado}, \min(q + n, M) \rangle & \text{si } E = \text{parado} \wedge x = \text{autoIn}(n) \\ \langle \text{circulando}, q \rangle & \text{si } E = \text{parado} \wedge x = \text{libre} \\ \langle \text{parado}, q + n \rangle & \text{si } E = \text{circulando} \wedge x = \text{stop}(n) \\ \langle \text{parado}, \min(q + n, M) \rangle & \text{si } E = \text{circulando} \wedge x = \text{autoIn}(n) \end{cases}$$

$$\lambda(S) = \begin{cases} \text{autoOut}(C) & \text{si } q > C \wedge E = \text{circulando} \\ \text{autoOut}(q) & \text{si } q \leq C \wedge E = \text{circulando} \end{cases}$$

$$t_a(S) = \begin{cases} \infty & \text{si } E = \text{vacío} \\ \infty & \text{si } E = \text{parado} \\ \varepsilon - tt & \text{si } E = \text{circulando} \wedge \text{cond} \\ \varepsilon & \text{si en caso contrario} \end{cases}$$

Siendo 'cond' verdadero sii se llega S por el evento $autoIn(n)$ cuando E valía 'circulando'.

Estrategias de testing

A continuación se describen los casos de testing planeados para este modelo.

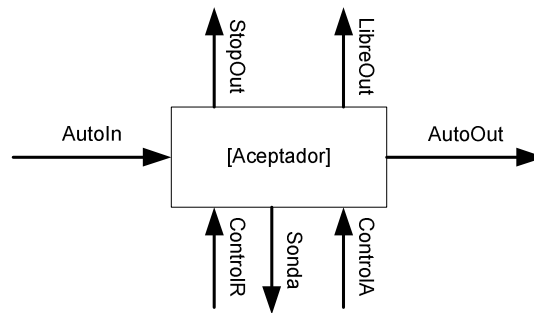
Objetivo: Testear la capacidad de soportar un flujo de entrada superior al de salida con una Entrada de n ($n < C$) vehículos en intervalos inferiores a ε	
00:00:00:00 autoIn 1 00:00:00:02 autoIn 1 00:00:00:04 autoIn 1 00:00:00:06 autoIn 1 00:00:00:08 autoIn 1 00:00:00:10 autoIn 1 00:00:00:12 autoIn 1 00:00:00:14 autoIn 1 00:00:00:16 autoIn 1 00:00:00:18 autoIn 1 00:00:00:20 autoIn 1	TEST 1.1 Configuración del modelo [cola1] carriles : 4 capacidad : 50 preparacion : 00:00:00:005 Resultado esperado : Salidas de 3 vehículos cada 5ms

Objetivo: Testear la capacidad de soportar un flujo de entrada superior al de salida con una Entrada de n ($n < C$) vehículos en intervalos inferiores a ε	
00:00:00:00 autoIn 1 00:00:00:01 autoIn 1 00:00:00:03 autoIn 1 00:00:00:04 autoIn 1 00:00:00:05 autoIn 1 00:00:00:06 autoIn 1 00:00:00:07 autoIn 1 00:00:00:08 autoIn 1 00:00:00:09 autoIn 1 00:00:00:10 autoIn 1 00:00:00:11 autoIn 1 00:00:00:13 autoIn 1 00:00:00:14 autoIn 1 00:00:00:15 autoIn 1 00:00:00:16 autoIn 1 00:00:00:17 autoIn 1 00:00:00:18 autoIn 1 00:00:00:19 autoIn 1	TEST 1.2 Configuración del modelo [cola1] carriles : 4 capacidad : 50 preparacion : 00:00:00:005 Resultado esperado : Salidas de 4 vehículos cada 5ms

Objetivo: Testear la capacidad de soportar un flujo de entrada superior al de salida con una entrada de n ($n > C$) vehículos en intervalos de $2.\varepsilon$.	
00:00:00:00 autoIn 5 00:00:00:10 autoIn 5 00:00:00:20 autoIn 5 00:00:00:30 autoIn 5	TEST 1.3 Configuración del modelo [cola1] carriles : 4 capacidad : 50 preparacion : 00:00:00:005

Objetivo: Testear El correcto funcionamiento del control de flujo de salida. Para esto se debe probar el interrumpir y reanudar el flujo de salida mientras ingresan vehículos y cuando ya han ingresado todos.	
00:00:00:000 autoIn 5 00:00:00:010 autoIn 5 00:00:00:013 stop 3 00:00:00:020 autoIn 5 00:00:00:030 autoIn 5 00:00:00:040 libre 0 00:00:00:100 stop 5 00:00:00:200 libre 0	TEST 1.4 Configuración del modelo [cola1] carriles : 4 capacidad : 50 preparacion : 00:00:00:005 Resultado esperado: 00:00:00:005 salida 4 00:00:00:010 salida 4 00:00:00:045 salida 4 00:00:00:050 salida 4 00:00:00:055 salida 4 00:00:00:060 salida 3 00:00:00:205 salida 4 00:00:00:210 salida 1

Modelo Aceptador



La función de este modelo es servir como válvula de entrada para una vía (o estación de peaje). Posee los siguientes parámetros de configuración:

- *Capacidad*: La cantidad máxima de vehículos que permitirá pasar inicialmente.
- *Preparacion*: Tiempo de preparación.

Un aceptador tiene definida una cantidad máxima de vehículos que permitirá pasar al modelo siguiente. Por el puerto **AutoIn** registra la entrada de vehículos que luego del tiempo especificado por el parámetro *Preparación* salen por el puerto **AutoOut**. Una vez que se alcanzó el máximo de vehículos, el modelo envía una señal por el puerto **StopOut** que le indica al emisor que debe dejar de enviar vehículos.

Los Puertos **ControlA** y **ControlR** le indica al modelo que n vehículos han abandonado el sistema y por ende puede dejar pasar n vehículos más o que n vehículos han sido rechazados por el receptor y por ende reingresaron al sistema. Cuando nuevamente hay espacio disponible, se envía una señal por el puerto **LibreOut**.

El puerto **Sonda** genera salidas periódicamente informando la cantidad de vehículos en el sistema.

Variables de estado

Las siguientes variables conforman el estado del sistema.

- q : Cantidad de vehículos que hay en el sistema.
- r : Cantidad de vehículos que serán rechazados.
- s : Cantidad de vehículos que pasarán.
- l : Debe indicarse al emisor que puede reanudar el envío.
- $lastCtrl$: indica el último mensaje de control de flujo que se envió al emisor.
 - *Stop*
 - *Libre*
- E : Que representa el estado representado en la FSM.:
 - *Pasivo*: No hay autos en la cola.
 - *Activo*: Los autos de la cola pueden salir.

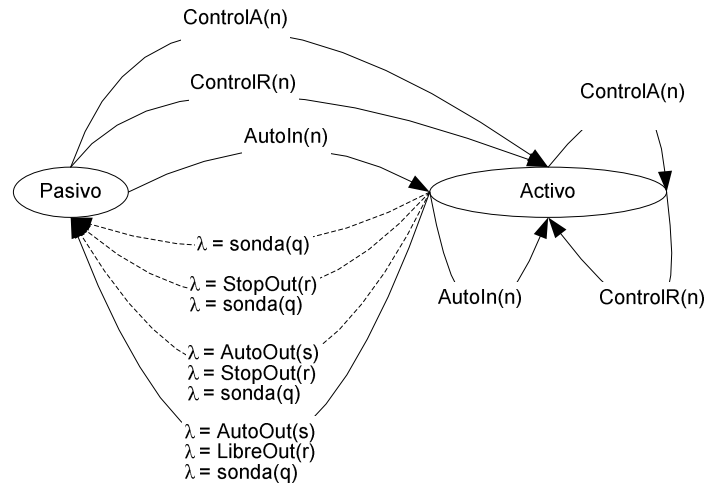
Constantes

Las constantes de este modelo (inicializadas mediante los parámetros de configuración) son:

- M : Capacidad de vehículos que el **aceptador** permitirá que acumule el sistema.
- ε : Tiempo de preparación.

Transición entre estados

La siguiente FSM muestra las funciones de transición internas y externas.



Especificación formal del modelo

$$\text{Aceptador} = \langle X, S, Y, \delta_{int}, \delta_{ext}, \lambda, t_a \rangle$$

Donde

$$X = \{autoIn(n), controlR(n), controlA(n)\}$$

$$Y = \{autoOut(n), libreOut(), stopOut(n)\}$$

$$S = \langle E, q, r, s, l, lastCtrl \rangle \text{ con } E \in \{pasivo, activo\} \text{ y } (q, r, s, l, lastCtrl) \in N$$

El siguiente código muestra la función $\delta_{int}(S)$

```

Function  $\delta_{int}(S)$  {
  if ( $q < M$  &&  $l == 1$ ) lastCtrl = libre;
  if ( $r > 0$ ) lastCtrl = stop;
   $s = 0$ ;  $r = 0$ ;  $l = 0$ ;
}

```

De la misma forma, resulta mucho más fácil comprender el funcionamiento de la función $\delta_{ext}(S, e, x)$ a partir del siguiente pseudocódigo. El código muestra como se modifican cada una de las variables de estado.

```

Function  $\delta_{int}(S, e, x)$  {
  switch(E) {
    case pasivo:
      if ( $x = autoIn(n)$ ) {
        if ( $n + q > M$ ) {
           $r = (n + q) - \max(M, q)$ ;
           $s = \max(0, M - q)$ ;
           $q = \max(M, q)$ ;
        } else {
           $s = n$ ;  $q = q + n$ ;
        }
      } else if ( $x = controlR(n)$ ) {
         $q = q + n$ ;
      } else if ( $x = controlA(n)$ ) {
         $q = q - n$ ;
         $l = 1$ ;
      }
    case activo:
      if ( $x = autoIn(n)$ ) {
        if ( $n + q > M$ ) {
           $r += (n + q) - \max(M, q)$ ;
           $s += \max(0, M - q)$ ;
           $q = \max(M, q)$ ;
        } else {

```

```

        s += n; q = q + n;
    }
} else if (x = controlR(n)) {
    q = q + n - min(s, n);
    r = r + min(s, n);
    s = max(0, s - n);

} else if (x = controlA(n)) {
    q = q - n + min(r, n);
    s = s + min(r, n);
    r = max(0, r - n);
    l = 1;
}
}
}

```

Por último, esta es la función $\lambda(S)$

```

Function  $\lambda(S)$  {
    if (s > 0)    sendOutput( msg.time(), autoOut, s );
    if (r > 0)    sendOutput( msg.time(), stopOut, r );
    if (q < M && l == 1 && lstCtrl == stop) sendOutput( msg.time(), libreOut, 0 );
    sendOutput( msg.time(), sonda, q );
}

```

Por último, la función $ta(S)$ se define así:

$$ta(S) = \begin{cases} \infty & \text{si } E = \text{pasivo} \\ \varepsilon & \text{si } E = \text{activo} \end{cases}$$

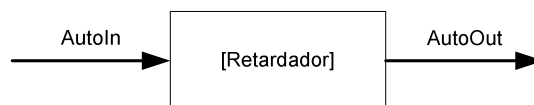
Estrategias de testing

A continuación se describen los casos de testing planeados para este modelo.

Objetivo: Testear que efectivamente, el modelo sólo permite que ingresen M vehículos y rechaza el resto.	
00:00:00:00 autoIn 10 00:00:00:01 autoIn 10 00:00:00:02 autoIn 10 00:00:00:03 autoIn 10 00:00:00:04 autoIn 10 00:00:00:05 autoIn 10 00:00:00:06 autoIn 10 00:00:00:07 autoIn 10 00:00:00:08 autoIn 10 00:00:00:09 autoIn 10	TEST 2.1 Configuración del modelo [model] preparacion : 00:00:00:005 capacidad : 15 Resultado esperado : Generar eventos autoOut por un valor total de 15 y stop por un total de 85.

Objetivo: Testear la respuesta del modelo ante los eventos controlR y controlA en los estados pasivo y activo.	
00:00:00:00 autoIn 10 00:00:00:01 autoIn 10 00:00:00:15 controlA 10 00:00:00:35 controlA 1 00:00:00:45 controlA 1 00:00:00:50 autoIn 10 00:00:00:60 autoIn 10 00:00:00:70 controlA 10 00:00:00:535 controlA 1 00:00:00:545 controlA 1	TEST 2.1 Configuración del modelo [model] preparacion : 00:00:00:005 capacidad : 15 Resultado esperado : No deben generarse 2 salidas consecutivas del evento stop ni del evento libre.

Modelo Retardador



El objetivo de este modelo es simular el tiempo que le lleva a un vehículo atravesar una vía de una x distancia. Los parámetros para este modelo son los siguientes:

- *Carriles*: La cantidad carriles de la vía.

- *Tiempo*: Tiempo mínimo que le toma a un vehículo atravesar la vía de un extremo al otro.
- *Distancia*: La distancia mínima, en tiempo, entre dos vehículos consecutivos.

Los puertos **AutoIn** y **AutoOut** representan la entrada y salida de vehículos a la vía. Con cada evento autoIn que se produce, ingresan n vehículos al sistema. Luego de el tiempo definido por *Tiempo*, los vehículos salen, en grupos de no más que *Carriles* vehículos juntos a una separación de por lo menos *Distancia* entre cada grupo.

Variables de estado

Las variables de estado del modelo son las siguientes:

- Q : La cola con las *horas* de salida de cada vehículo.
- Lh_i : La última hora de salida registrada para cada carril.
- E : Que representa el estado representado en la FSM.:
 - *Pasivo*
 - *Activo*

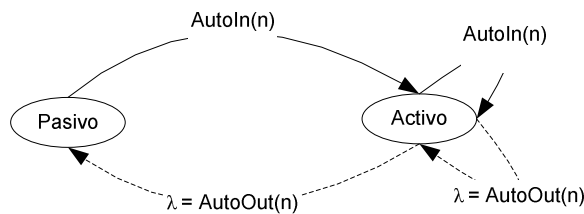
Constantes

Las constantes de este modelo (inicializadas mediante los parámetros de configuración) son:

- C : Cantidad de carriles.
- ε : Tiempo que le toma a un vehículo atravesar la vía.
- Δ : Separación mínima entre dos vehículos.

Transición entre estados

La siguiente FSM muestra las funciones de transición internas y externas.



Especificación formal del modelo

$$Retardador = \langle X, S, Y, \delta_{int}, \delta_{ext}, \lambda, t_a \rangle$$

Donde

$$X = \{autoIn(n)\}$$

$$Y = \{autoOut(n)\}$$

$$S = \langle E, Q, s, lh_i \rangle \text{ con } E \in \{pasivo, activo\}, (q, r, s, l, lastCtrl) \in N \text{ y } 0 \leq i < C$$

$$\delta_{int}(S) = \begin{cases} \langle \text{pasivo}, Q, 0, lh \rangle & \text{si } vacio(Q) \\ \langle \text{activo}, Q', s', lh \rangle & \text{si } \neg vacio(Q) \end{cases}$$

Donde

$$s' = \# \{x / x \in Q \wedge x = \min(Q)\}$$

$$Q' = Q - \{x / x \in Q \wedge x = \min(Q)\}$$

$$\delta_{ext}(S) = \begin{cases} \langle \text{activo}, Q^1, s, lh^1 \rangle & \text{si } E = \text{activo} \\ \langle \text{activo}, Q^2, s^2, lh^1 \rangle & \text{si } \neg vacio(Q) \end{cases}$$

Donde

$$s^2 = \# \{x / x \in Q^1 \wedge x = \min(Q^1)\}$$

$$Q^2 = Q^1 - \{x / x \in Q^1 \wedge x = \min(Q^1)\}$$

Q^1 y lh^1 se define con el siguiente algoritmo:

```
while (n > 0) {
    tSalida = max(lh[min(lh)] + Distancia, t + Tiempo);
    lh[min(lh, C)] = tSalida;
    for (int i = 0; i < C; i++) {
        if (lh[i] < tSalida - Distancia) {
            lh[i] = tSalida - Distancia;
        }
    }
    Q.push_back(tSalida);
    n--;
}
```

$$\lambda(S) = \text{autoOut}(s) \text{ si } E = \text{activo}$$

$$ta(S) = \begin{cases} \infty & \text{si } E = \text{pasivo} \\ \min(Q) & \text{si } E = \text{activo y se llegó a S por un evento int. o desde pasivo} \\ & \text{si } E = \text{activo y se llegó a S por un evento externo desde activo} \end{cases}$$

Estrategias de testing

A continuación se describen los casos de testing planeados para este modelo.

Objetivo: Testear si los eventos de salida respetan el espaciado definido por el parámetro <i>Distancia</i>	
00:00:01:00 autoIn 4 00:00:02:00 autoIn 4 00:00:03:00 autoIn 4 00:00:04:00 autoIn 4 00:00:05:00 autoIn 4 00:00:06:00 autoIn 4	TEST 3.1 Configuración del modelo [model] tiempo : 00:01:00:00 distancia : 00:00:05:00 carriles : 4 Resultado esperado : 6 salidas de 4 vehículos espaciadas por 5 segundos con una diferencia de de un minuto con las entradas.

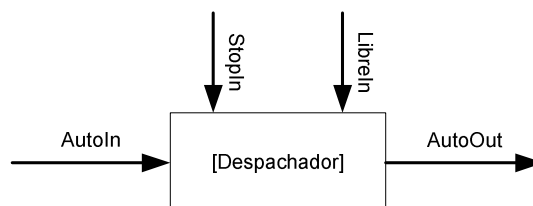
Objetivo: Testear si funciona correctamente el reordenamiento de los vehículos dentro de la cola para entradas inferiores a C.	
00:00:01:00 autoIn 1 00:00:02:00 autoIn 1 00:00:03:00 autoIn 1 00:00:04:00 autoIn 1 00:00:05:00 autoIn 1 00:00:06:00 autoIn 1	TEST 3.2 Configuración del modelo [model] tiempo : 00:01:00:00 distancia : 00:00:05:00 carriles : 4

00:00:07:00 autoIn 1 00:00:08:00 autoIn 1 00:00:09:00 autoIn 1	Resultado esperado : 00:01:01:000 autoout 1 00:01:02:000 autoout 1 00:01:03:000 autoout 1 00:01:04:000 autoout 1 00:01:06:000 autoout 1 00:01:07:000 autoout 1 00:01:08:000 autoout 1 00:01:09:000 autoout 1 00:01:11:000 autoout 1
--	--

Objetivo: Testear si funciona correctamente el reordenamiento de los vehículos dentro de la cola para entradas superiores a C.	
00:00:01:00 autoIn 5 00:00:02:00 autoIn 5 00:00:03:00 autoIn 5	TEST 3.3 Configuración del modelo [model] tiempo : 00:01:00:00 distancia : 00:00:05:00 carriles : 4 Resultado esperado : 00:01:01:000 autoout 4 00:01:06:000 autoout 4 00:01:11:000 autoout 4 00:01:16:000 autoout 3

Objetivo: Testear si funciona correctamente el reordenamiento de los vehículos dentro de la cola para entradas superiores a 2C.	
00:00:01:00 autoIn 9 00:00:02:00 autoIn 9 00:00:03:00 autoIn 9	TEST 3.3 Configuración del modelo [model] tiempo : 00:01:00:00 distancia : 00:00:05:00 carriles : 4 Resultado esperado : 00:01:01:000 autoout 4 00:01:06:000 autoout 4 00:01:11:000 autoout 4 00:01:16:000 autoout 4 00:01:21:000 autoout 4 00:01:26:000 autoout 4 00:01:31:000 autoout 3

Modelo Despachador



La función de este modelo es simular la salida de vehículos de una vía. Permite al receptor manejar el flujo de vehículos. Posee los siguientes parámetros de configuración:

- *Preparacion*: Tiempo de preparación.

Cuando se recibe un evento por el puerto **AutoIn**, se registra el ingreso de vehículos y los mismos son despachados a intervalos definidos por *Preparación* por el puerto **AutoOut**. Mediante los puertos **StopIn** y **LibreIn** el modelo receptor puede manejar el flujo de vehículos.

Un despachador siempre intentará despachar tantos vehículos como tenga registrados y es responsabilidad de receptor el despachar los vehículos excedentes.

Variables de estado

Las siguientes variables conforman el estado del sistema.

- q : Cantidad de vehículos a despachar.
- R : indica el último mensaje de control de flujo que se recibió del receptor.
 - *Stop*
 - *Libre*
- E : Que representa el estado representado en la FSM.:
 - *Activo*: Despachando vehículos.
 - *Pasivo*: Esperando que los vehículos ingresen.

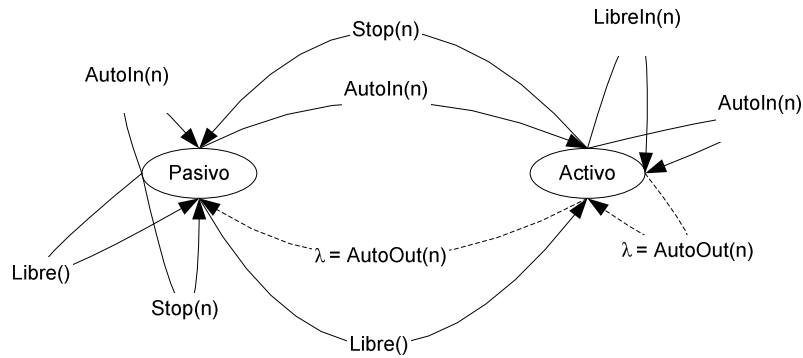
Constantes

Las constantes de este modelo (inicializadas mediante los parámetros de configuración) son:

- ε : Tiempo de preparación.

Transición entre estados

La siguiente FSM muestra las funciones de transición internas y externas.



Especificación formal del modelo

$$Cola = \langle X, S, Y, \delta_{int}, \delta_{ext}, \lambda, t_a \rangle$$

Donde

$$X = \{autoIn(n), stopIn(n), libreIn\}$$

$$Y = \{autoOut(n)\}$$

$$S = \langle E, R, q \rangle \text{ con } E \in \{activo, pasivo\}, R \in \{libre, stop\} \text{ y } q \in N$$

$$\delta_{int}(S) = \{ \langle pasivo, R, 0 \rangle \text{ si } E = activo$$

$$\delta_{ext}(S, e, x) = \begin{cases} \langle activo, libre, q+n \rangle & si & E = pasivo \wedge x = autoIn(n) \wedge R = libre \\ \langle pasivo, stop, q+n \rangle & si & E = pasivo \wedge x = autoIn(n) \wedge R = stop \\ \langle pasivo, libre, q \rangle & si & E = pasivo \wedge x = libre() \\ \langle pasivo, stop, q+n \rangle & si & E = pasivo \wedge x = stop(n) \\ \langle activo, libre, q+n \rangle & si & E = activo \wedge x = autoIn(n) \\ \langle activo, libre, q \rangle & si & E = activo \wedge x = libre() \\ \langle pasivo, stop, q+n \rangle & si & E = activo \wedge x = stop(n) \end{cases}$$

$$\lambda(S) = \{autoOut(q) \text{ si } E = activo\}$$

$$ta(S) = \begin{cases} \infty & si & E = pasivo \\ \varepsilon & si & E = activo \text{ y viene de una transicion interna o de pasivo} \\ \varepsilon - tt & si & E = activo \text{ y viene de una transicion externa desde activo} \end{cases}$$

Estrategias de testing

A continuación se describen los casos de testing planeados para este modelo.

Objetivo: Testear que el modelo conserva la cantidad de vehículos y los despacha correctamente aún cuando lleguen en intervalos menores a <i>Preparacion</i> .	
00:00:00:00 autoIn 10 00:00:00:01 autoIn 10 00:00:00:02 autoIn 10 00:00:00:03 autoIn 10 00:00:00:04 autoIn 10 00:00:00:05 autoIn 10 00:00:00:06 autoIn 10 00:00:00:07 autoIn 10 00:00:00:08 autoIn 10 00:00:00:09 autoIn 10	TEST 4.1 Configuración del modelo [model] preparacion : 00:00:00:005 Resultado esperado : Salida de vehículos en intervalos de 5ms.

Objetivo: Verificar el correcto funcionamiento de los puertos de control de flujo	
00:00:00:10 autoIn 10 00:00:00:20 autoIn 10 00:00:00:30 autoIn 10 00:00:00:40 stopIn 5 00:00:01:10 autoIn 10 00:00:01:20 autoIn 10 00:00:01:30 autoIn 10 00:00:01:40 libreIn 0 00:00:01:50 stopIn 20 00:01:00:00 libreIn 0	TEST 4.2 Configuración del modelo [model] preparacion : 00:00:00:005 Resultado esperado : Salida de 30 vehículos, luego la salida se interrumpe por un minuto y luego se reanuda. Por último salida de 20 vehículos una hora después.

Modelo Intersección

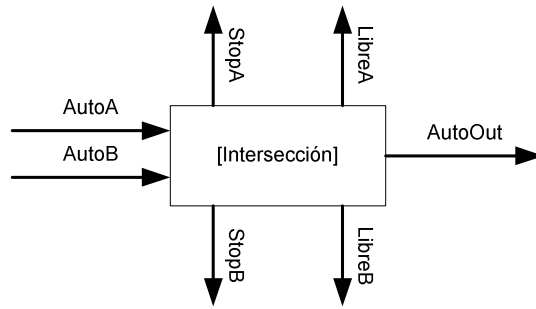


Figura 12: Modelo Intersección

Este es el último de los modelos atómico implementados y representa la intersección entre dos vías.

Los puertos **Libre** y **Stop** son utilizados para manejar el flujo de vehículos que ingresan al modelo por los puertos **Auto**. Por el puerto **AutoOut** se generan mensajes que representan la salida de vehículos del modelo. El puerto **AutoA** tiene prioridad sobre el puerto **AutoB**.

Al igual que una vía, una intersección tiene un número de carriles que es la máxima cantidad de vehículos que pueden salir del sistema en una unidad de tiempo.

Variables de estado

Las siguientes variables conforman el estado del sistema.

- q : Cantidad de vehículos a despachar.
- ra : Cantidad de vehículos de A que serán rechazados.
- rb : Cantidad de vehículos de B que serán rechazados.
- sla : Indica que un vehículo de A ha sido rechazado y debe enviarse el evento *libre* en cuanto sea posible.
- slb : Indica que un vehículo de B ha sido rechazado y debe enviarse el evento *libre* en cuanto sea posible.
- $nextTime$: Tiempo faltante para la próxima salida de vehículos.
- E : Que representa el estado propiamente dicho
 - *Activo*: Despachando vehículos.
 - *Pasivo*: Esperando la entrada de vehículos.
 - *Reiniciando*: Preparado para enviar los mensajes **LibreA/B**
 - *Rechazando*: Preparado para enviar los mensajes **STOPA/B**

Parámetros

Este modelo posee los siguientes parámetros de configuración:

- *Carriles*: Indica la cantidad de vehículos que pueden salir del modelo en una unidad de tiempo.
- *Preparación* (ϵ): Tiempo de preparación
- *Reinicio* (ζ): Tiempo de reinicio.

Especificación formal del modelo

$$Cola = \langle X, S, Y, \delta_{int}, \delta_{ext}, \lambda, t_a \rangle$$

Donde

$X = \{autoA(n), autoB(n)\}$

$Y = \{autoOut(n), stopA(n), stopB(n), libreA(), libreB()\}$

$S = \langle E, q, ra, rb, sla, slb, nextTime \rangle$ con $E \in \{\text{activo}, \text{pasivo}, \text{reiniciando}, \text{rechazando}\}$, $(q, ra, rb, nextTime) \in N$ y $(sla, slb) \in \{0, 1\}$

El siguiente código especifica como cambia S con la función $\delta_{int}(S)$.

```
function  $\delta_{int}(S)$  {
  if (E == activo) {
    q = 0;
    if (sla || slb) {
      E = reiniciando;
    } else {
      E = pasivo;
    }
  } else if (E == reiniciando) {
    ra = 0; rb = 0;
    sla = 0; slb = 0;
    E = pasivo;
  } else if (E == rechazando) {
    ra = 0; rb = 0;
    E = activo;
  }
}
```

El siguiente código especifica como cambia S con la función $\delta_{ext}(S, e, x)$.

```
function  $\delta_{ext}(S, e, x)$  {
  if (q > 0 || ra > 0 || rb > 0) {
    if (E == pasivo) {
      if (ra > 0 || rb > 0) {
        E = rechazando;
        nextTime =  $\varepsilon - \varsigma$ ;
      } else {
        E = activo;
      }
    } else if (E == activo) {
      if ((ra > 0 || rb > 0) && (nextChange() > reinitTime)) {
        nextTime = nextChange() -  $\varsigma$ ;
        E = rechazando;
      } else {
        E = pasivo;
      }
    }
  }
}
```

La función $\lambda(S)$ se define como lo muestra el siguiente segmento de código

```
if (E == activo) {
  if (q > 0) sendOutput( msg.time(), autoOut, q );
  if (ra > 0) sendOutput( msg.time(), stopA, ra );
  if (rb > 0) sendOutput( msg.time(), stopB, rb );
} else if (E == reiniciando) {
  if (ra > 0) sendOutput( msg.time(), libreA, 0 );
  if (rb > 0) sendOutput( msg.time(), libreB, 0 );
}
```

Por último

$$ta(S) = \begin{cases} \infty & \text{si } E = \text{pasivo} \\ \varepsilon & \text{si } E = \text{activo y viene de una transicion interna o de pasivo} \\ \varepsilon - tt & \text{si } E = \text{activo y viene de una transicion externa desde activo} \\ \varsigma & \text{si } E = \text{reiniciando} \vee E = \text{rechazando} \end{cases}$$

Estrategias de testing

Para testear el funcionamiento de este modelo, lo que se hizo fue acoplarlo con dos “despachadores”. De esta forma se puede probar el funcionamiento de los puertos de control de flujo con un sencillo caso de prueba.

Objetivo: Testear el funcionamiento del control de flujo y la coordinación de los dos modelos origen.	
00:00:00:00 autoA 10 00:00:00:01 autoB 10 00:00:00:02 autoA 10 00:00:00:03 autoB 10 00:00:00:04 autoA 10 00:00:00:05 autoB 10	<p>TEST 5.1</p> <p>[top]</p> <p>components : int@Interseccion da@Despachador db@Despachador</p> <p>Out : autoOut In: autoA autoB</p> <p>Link : autoA autoIn@da Link : autoB autoIn@db Link : autoOut@int autoOut</p> <p>Link : autoOut@da autoA@int Link : stopA@int stopIn@da Link : libreA@int libreIn@da</p> <p>Link : autoOut@db autoB@int Link : stopB@int stopIn@db Link : libreB@int libreIn@db</p> <p>[int]</p> <p>preparacion : 00:00:00:005 reinicio : 00:00:00:001 carriles : 4</p> <p>[da]</p> <p>preparacion : 00:00:00:005</p> <p>[db]</p> <p>preparacion : 00:00:00:005</p>

Parte 3: Implementación y ejecución

Testing de modelos atómicos

En la sección “Especificación formal del modelo

$$Top = \langle X, Y, \{Aceptador, Retardador, Despachador\}, I_i, Z_{ij} \rangle$$

Donde:

$$X = \{AutoIn, StopIn, LibreIn\}$$

$$Y = \{Sonda, AutoOut, StopOut, LibreOut\}$$

$$I(self) = \{Aceptador, Despachador\}$$

$$I(Aceptador) = \{self, Retardador\}$$

$$I(Despachador) = \{self, Aceptador\}$$

$$I(Retardador) = \{Despachador\}$$

$$Z = \{(self.AutoIn \rightarrow Aceptador.AutoIn), (self.StopIn \rightarrow \{Despachador.StopIn, Aceptador.ControlR\}), (self.LibreIn \rightarrow Despachador.LibreOut), (Aceptador.Sonda \rightarrow Self.Sonda), (Aceptador.StopOut \rightarrow Self.StopOut), (Aceptador.LibreOut \rightarrow Self.LibreOut), (Aceptador.AutoOut \rightarrow Retardador.AutoIn), (Retardador.AutoOut \rightarrow Despachador.AutoIn), (Despachador.AutoOut \rightarrow \{self.AoutOut, Aceptador.ControlA\})\}$$

$$Select: Aceptador > Retardador > Despachador$$

Especificación de los modelos atómicos” se vieron algunas estrategias para testear, en forma individual, el funcionamiento de los modelos atómicos. Los resultados de ejecutar los tests no difieren de los resultados esperados, de modo que no aporta mucho volver a mostrarlos.

Resultados de la simulación

Los eventos de entrada que se utilizaron para la simulación, intentan representar el ingreso de vehículos al sistema en hora pico, esto es aproximadamente de 7:45 a 8:30 Hs. El siguiente gráfico muestra la distribución que se utilizó para crear los eventos externos. Para ambas entradas se utilizó la misma distribución, pero datos distintos (Test 6.1).

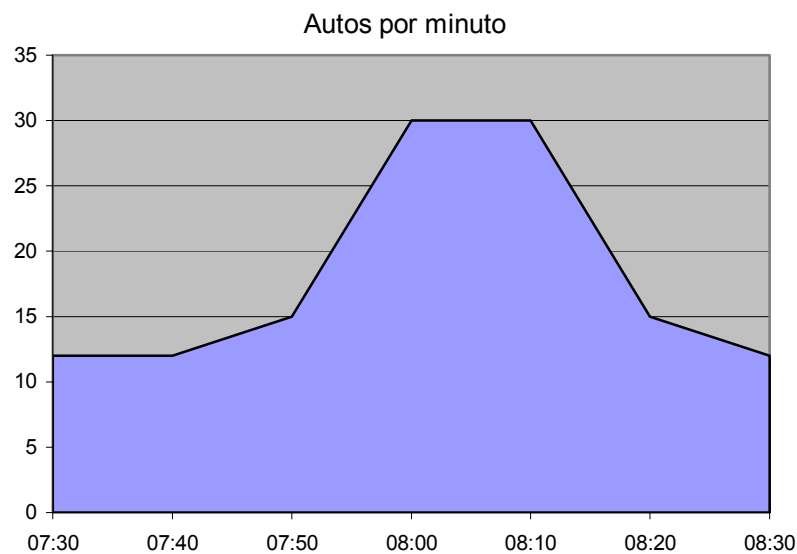


Figura 13: Distribución de los eventos de entrada

La variable que se tuvo en cuenta para analizar la simulación fue la cantidad de vehículos que se encontraban en cada tramo del sistema a lo largo del tiempo. Para ello se utilizaron los puertos “sonda” que reportan periódicamente la carga de vehículos de cada una de las secciones del sistema. El siguiente gráfico muestra cómo se dividió el sistemas en secciones.

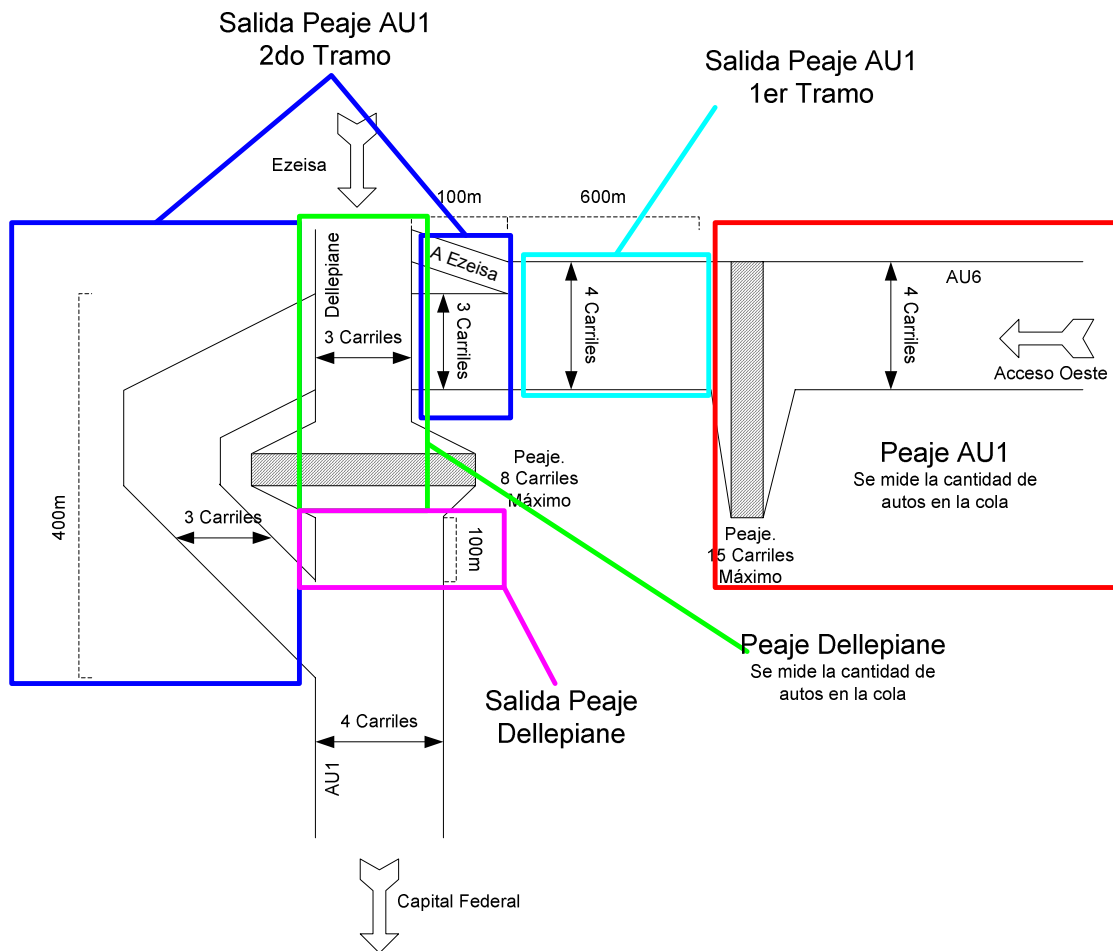
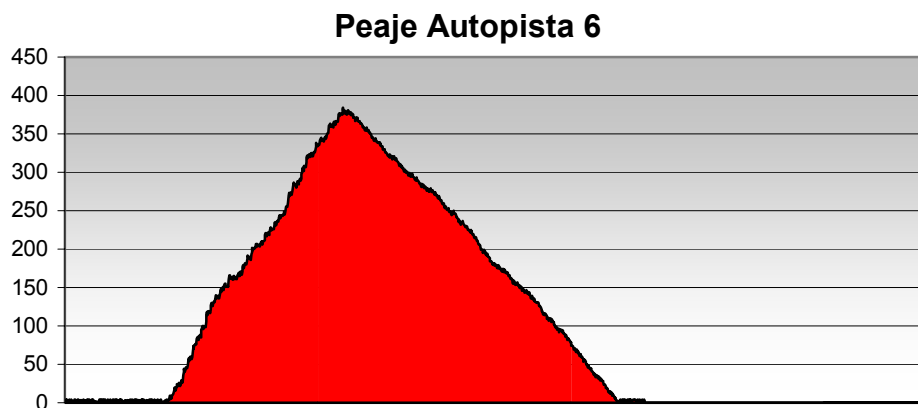


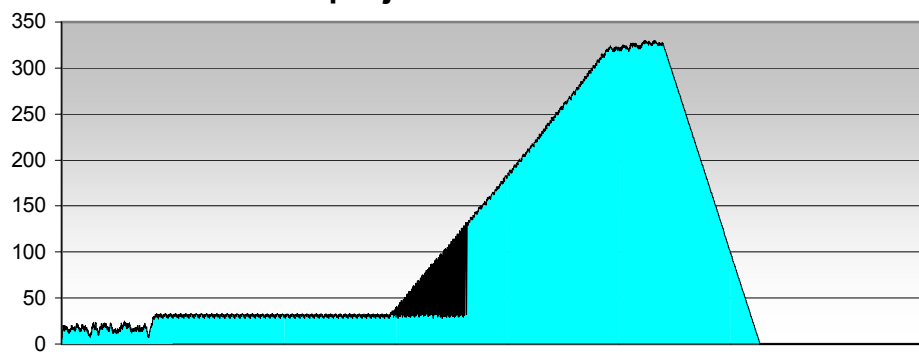
Figura 14: Secciones del sistema que fueron estudiadas

Los siguientes gráficos muestran el estado del tránsito en cada sector a lo largo del tiempo. La escala de tiempo en todos los gráficos es la misma, comenzando a las 7:45Hs y terminando a las 9:00 Hs (el último vehículo ingresó al sistema a las 8:32 Hs), de modo que resulta fácil ver cómo se desplaza el tráfico a lo largo del tiempo y qué zonas del sistema se ven más afectadas. Estos resultados corresponden al test 6.1.

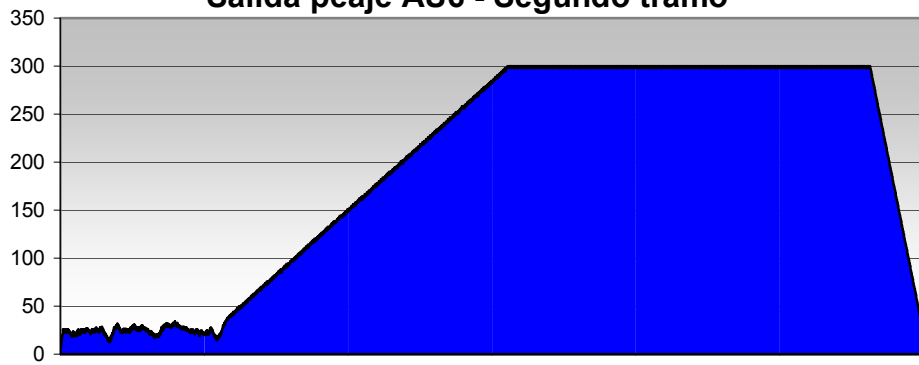
Como se puede ver en los gráficos, en la autopista Perito Moreno, la cola del peaje permanece con pocos autos la mayor parte del tiempo, pero se producen grandes congestionamientos en la salida del peaje. Con el peaje de la avenida Dellepiane pasa lo contrario, el peaje se congestiona rápidamente pero el tránsito hacia la 25 de Mayo es fluido.



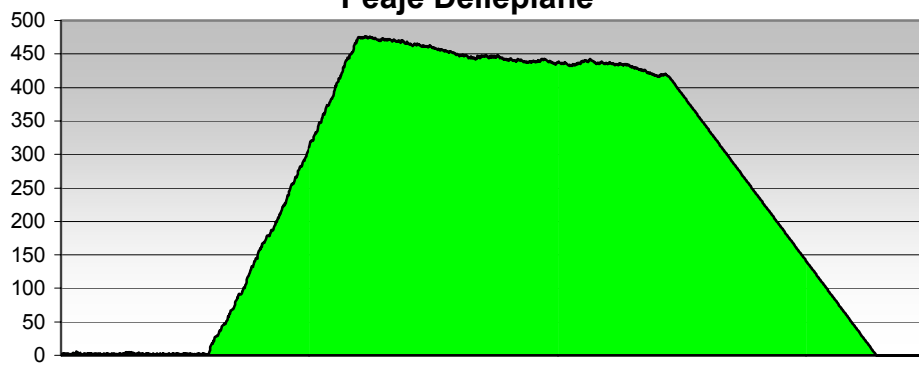
Salida del peaje de AU6 - Primer Tramo



Salida peaje AU6 - Segundo tramo



Peaje Dellepiane



Salida peaje Dellepiane



Figura 15: Gráficos del estado del tránsito en las distintas zonas

Los últimos dos tests que se realizaron sobre el modelo completo fueron los siguientes:

En el Test 6.2 se utilizó una entrada con una distribución similar a la de la *Figura 13*: Distribución de los eventos de entrada para alimentar la entrada del peaje de la autopista Perito Moreno, mientras la Avenida Dellepiane recibía un flujo promedio de 6 autos por minuto. Como resultado se obtuvo que, si bien el tránsito en la Perito Moreno se “ponía pesado”, seguía siendo fluido y ninguna de las vías llegaba a saturarse.

En el Test 6.3 se utilizó una entrada con una distribución similar a la de la *Figura 13*: Distribución de los eventos de entrada para alimentar la entrada del peaje de la Avenida Dellepiane, mientras la autopista Perito Moreno recibía un flujo promedio de 6 autos por minuto. Como resultado se obtuvo un gran embotellamiento a la entrada del peaje de la Avenida Dellepiane (debido a la poca capacidad del mismo) y un tránsito fluido y liviano en el resto del sistema.

Conclusiones

Los resultados obtenidos se asemejan mucho a una situación real, es decir, si bien el sistema (o la autopista) tiene capacidad suficiente para soportar un importante flujo de autos desde cualquiera de los dos ingresos, debido a la forma del empalme, el sistema no soporta el flujo de vehículos producido en la hora pico.

Lamentablemente no fue posible conseguir datos oficiales para compararlos con los resultados obtenidos y así calcular la exactitud de los mismos.

Algo notable es que pese a las simplificaciones que se realizaron sobre el sistema real para obtener el modelo a simular, el comportamiento obtenido concuerda razonablemente bien con el comportamiento del sistema real. De aumentarse el detalle de la simulación, modelando el flujo de los autos en forma más precisa (contemplando variaciones de velocidad, pasos de carriles, etc.), el modelo podría utilizarse tanto para planear mejoras en la autopista como para predecir el comportamiento del tránsito ante evento fortuitos (como accidentes o vehículos detenidos) y de esta forma planear el mejor curso de acción para minimizar las demoras.