UNIVERSIDAD NACIONAL DE CÓRDOBA
Facultad de Ciencias Exactas, Físicas y Naturales

TRABAJO PRÁCTICO FINAL

MODELADO DE UN SISTEMA AUTOMÁTICO DE EXTRACCIÓN DE AGUA

PROGRAMACIÓN CONCURRENTE

Alumnos:

Asson, Leandro 38003886
Rodriguez Felici, Joaquín 38029521

Docente: Orlando Micolini

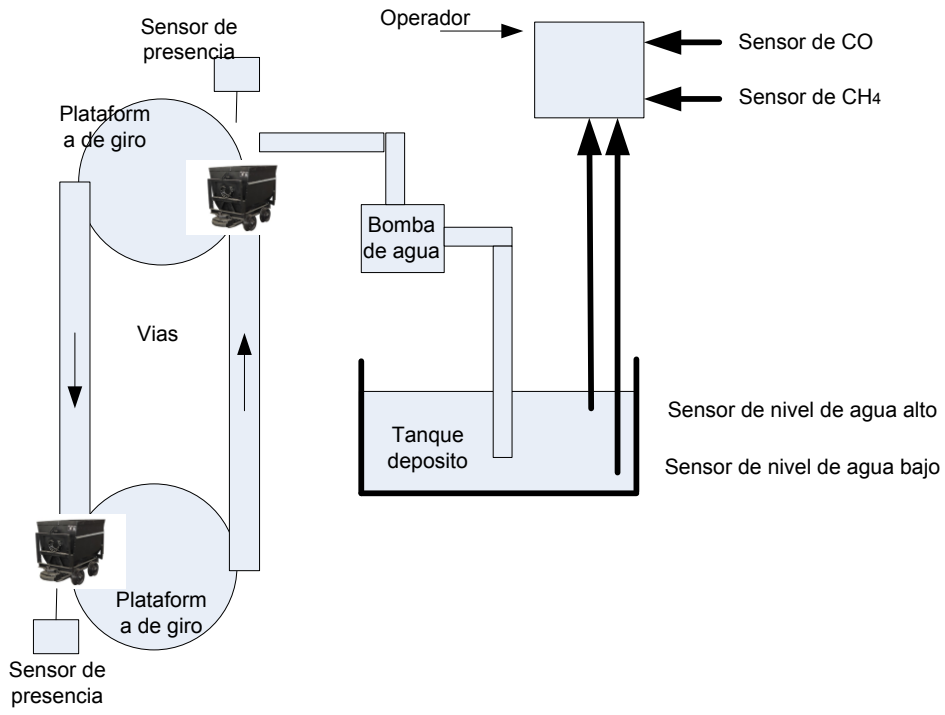
Ing. En Computación

2016

Índice

Introducción

El caso de estudio es un sistema de extracción de agua del sumidero del pozo de una mina. El mismo incluye una serie de sensores para controlarlo y asegurar tanto su funcionamiento como la seguridad del entorno. Además de verificar que haya agua en el pozo lista para ser extraída, el ambiente cuenta con un cierto porcentaje de metano en el aire. Debido a esto, debe controlarse con precisión los momentos en los que la bomba se encuentra encendida, ya que este caso nunca puede darse cuando, por ejemplo, el nivel de metano en el aire es mayor a un límite, que se considera como el máximo permitido. A su vez, el agua extraída es depositada en un carrito, perteneciente a un subsistema automático con dos carros. Este subsistema debe informar las posiciones de los carros para prender la bomba en el momento en que el carrito vacío se encuentre presente y listo para su llenado, así como un sensor dentro del carro para saber cuándo el mismo está lleno. Un diagrama representativo del sistema es el siguiente:



Debido a la cantidad de diferentes actores que interactuarán con el sistema simultáneamente, es necesario hacer un análisis y abarcar el problema desde un punto de vista concurrente.

Se procede a continuación a comenzar el análisis del caso de estudio, comenzando por identificar los actores y establecer los requerimientos de nuestro programa.

Requerimientos

Funcionales

1. La bomba debe activarse automáticamente cuando todas las condiciones necesarias se dan al mismo tiempo (metano bajo, nivel de agua suficiente y carrito presente en la plataforma de llenado).
2. El sistema debe alertar a través de alarmas ante situaciones que puedan considerarse anormales o peligrosas. Por ejemplo, cuando se tiene un nivel de metano más alto del mínimo o cuando, aún luego de haber apagado la bomba, se sensa un flujo de agua que no debería existir.

No funcionales

1. La bomba no debe funcionar cuando hay un nivel de metano mayor al mínimo tolerado.
2. La bomba no debe funcionar cuando el nivel de agua en el sumidero es menor al mínimo predeterminado.
3. El sistema debe responder en menos de 200 ms para garantizar las condiciones de seguridad necesarias.

Análisis del problema

Para comenzar a implementar el problema, debieron:

1. Identificarse los actores.
2. Definirse las acciones que cada actor puede realizar.
3. Determinarse el flujo del sistema, dependiendo del resultado de cada acción realizada por un actor.

Para esto, se realizaron dos diagramas que ayudarán a comprender tanto el funcionamiento total del sistema como las funciones de cada uno de los componentes.

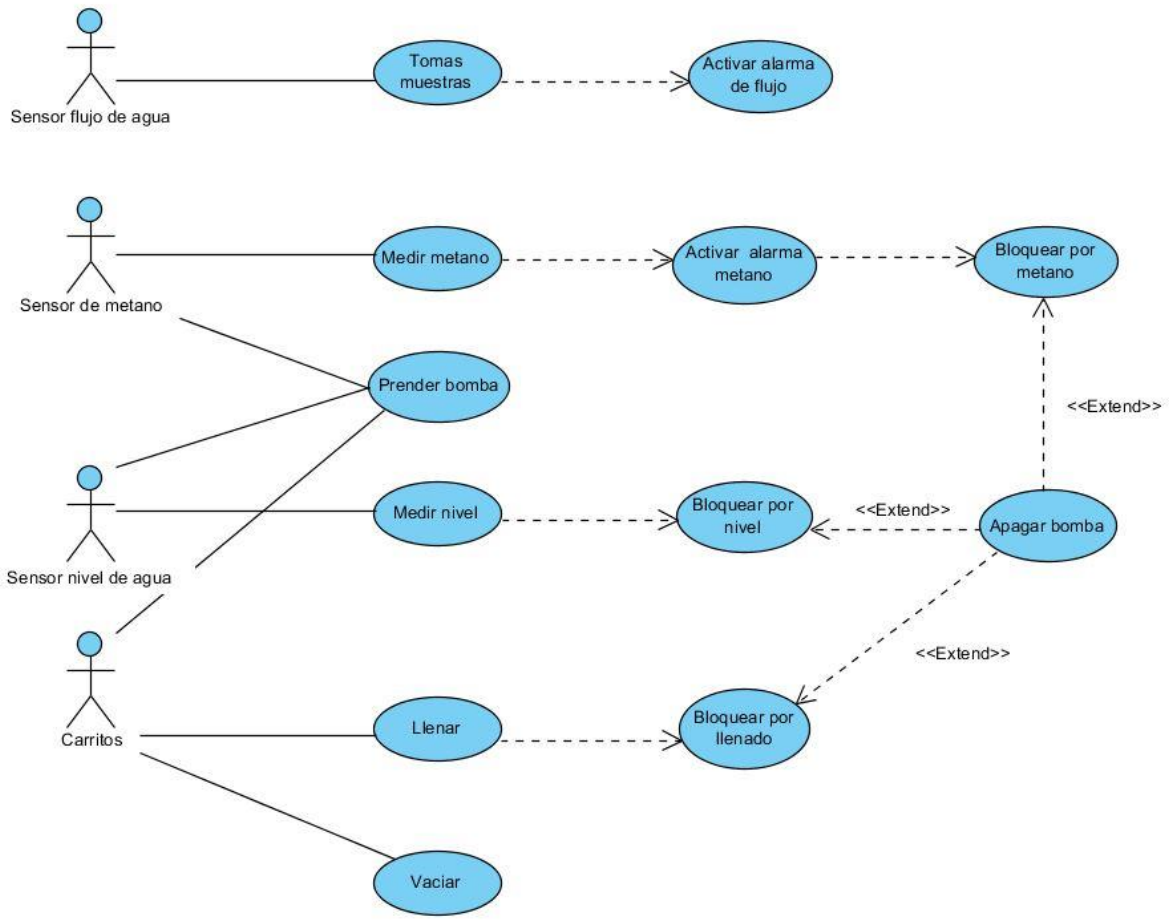
Diagrama de casos de uso

Los actores involucrados en el sistema serán:

1. Sensor de metano: Encargado de realizar las muestras, para luego ser analizadas y operar acorde al resultado. Su resultado provocará que la bomba se prenda o apague.

2. Sensor de nivel de agua: Encargado de informar si el nivel de agua en el sumidero es menor o mayor al mínimo predeterminado. Su resultado también provocará que la bomba se prenda o se apague.
3. Sensor de flujo de agua: En este caso, su resultado no afectará el estado de la bomba, ya que este sensor solo opera cuando la bomba acaba de apagarse, y enciende una alarma si continúa habiendo flujo.
4. Carrito: El sensor de presencia de carrito informará si la bomba puede o no ser prendida. A su vez, cuando el mismo se llene, ocasionará que la bomba se apague por llenado de carrito, volviendo al estado inicial.

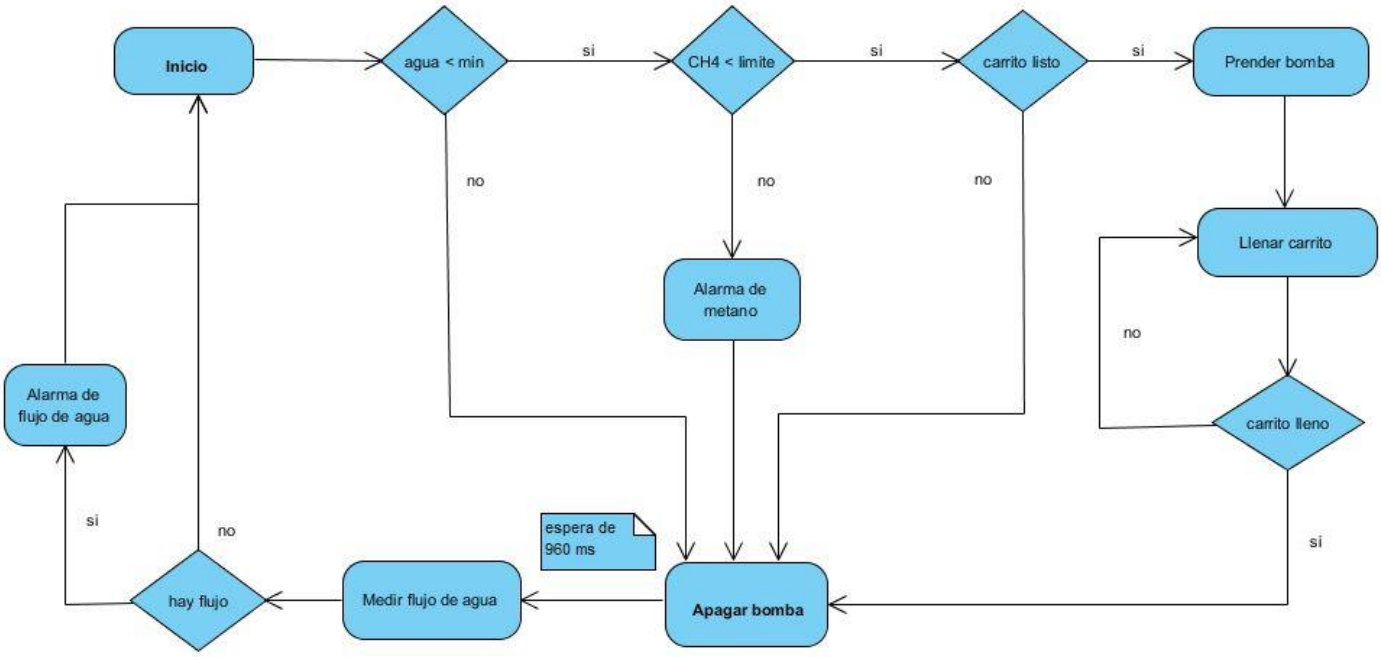
Con esta información, se puede comprender el siguiente diagrama con facilidad:



Por ejemplo, para el caso del sensor de metano, podemos observar que el mismo comenzará por medirlo, y en el caso de dar mayor al límite, activará una alarma y bloqueará la bomba, apagándola.

Diagramas de actividades

Otra forma de comprender el funcionamiento del programa (y de facilitar el proceso de codificación), es pre definir el diagrama de actividades, el cual es solo una representación gráfica del algoritmo, mostrando el flujo de ejecución del programa y sus cambios ante bifurcaciones.



En este diagrama podemos observar:

1. Que la bomba solo se enciende cuando:

    a. El nivel de agua es mayor al mínimo.
    b. El nivel de metano es menor al límite.
    c. El carrito está en la plataforma de llenado.

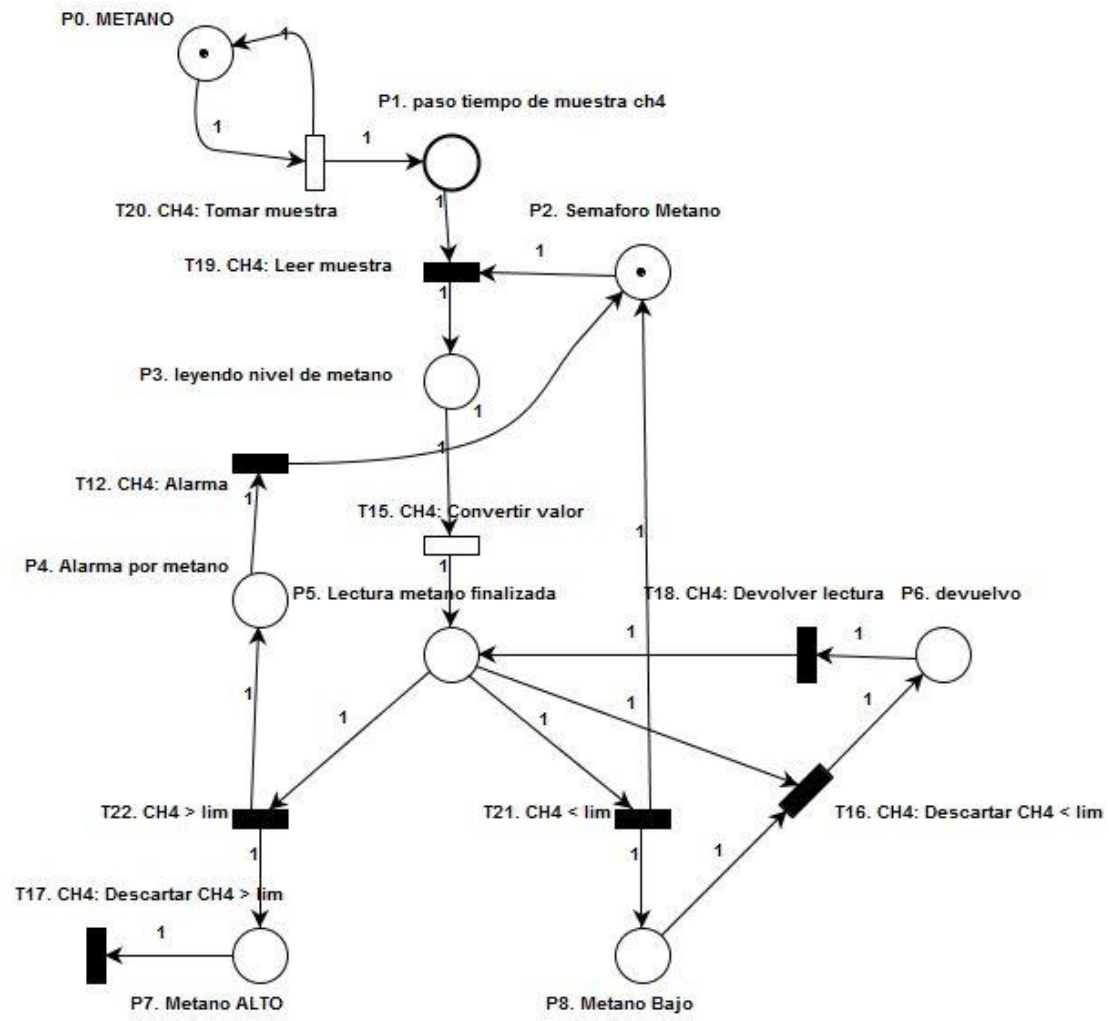
2. Que la misma se apaga cuando una de las tres condiciones anteriores no se cumple.
3. Que luego de apagarse, se comprueba que el agua no siga fluyendo, y que en el caso de hacerlo, simplemente se activa una alarma.

Implementación

Debido al número de actores que interactúan con el sistema simultáneamente, se procederá a abarcar el problema desde un punto de vista concurrente. En este caso, antes de comenzar a programar, se creará una ***red de Petri*** asociada al sistema para poder simularlo y comprobar su correcto funcionamiento antes de pasar a la siguiente etapa. Se mostrará a continuación cada parte de la red por separado:

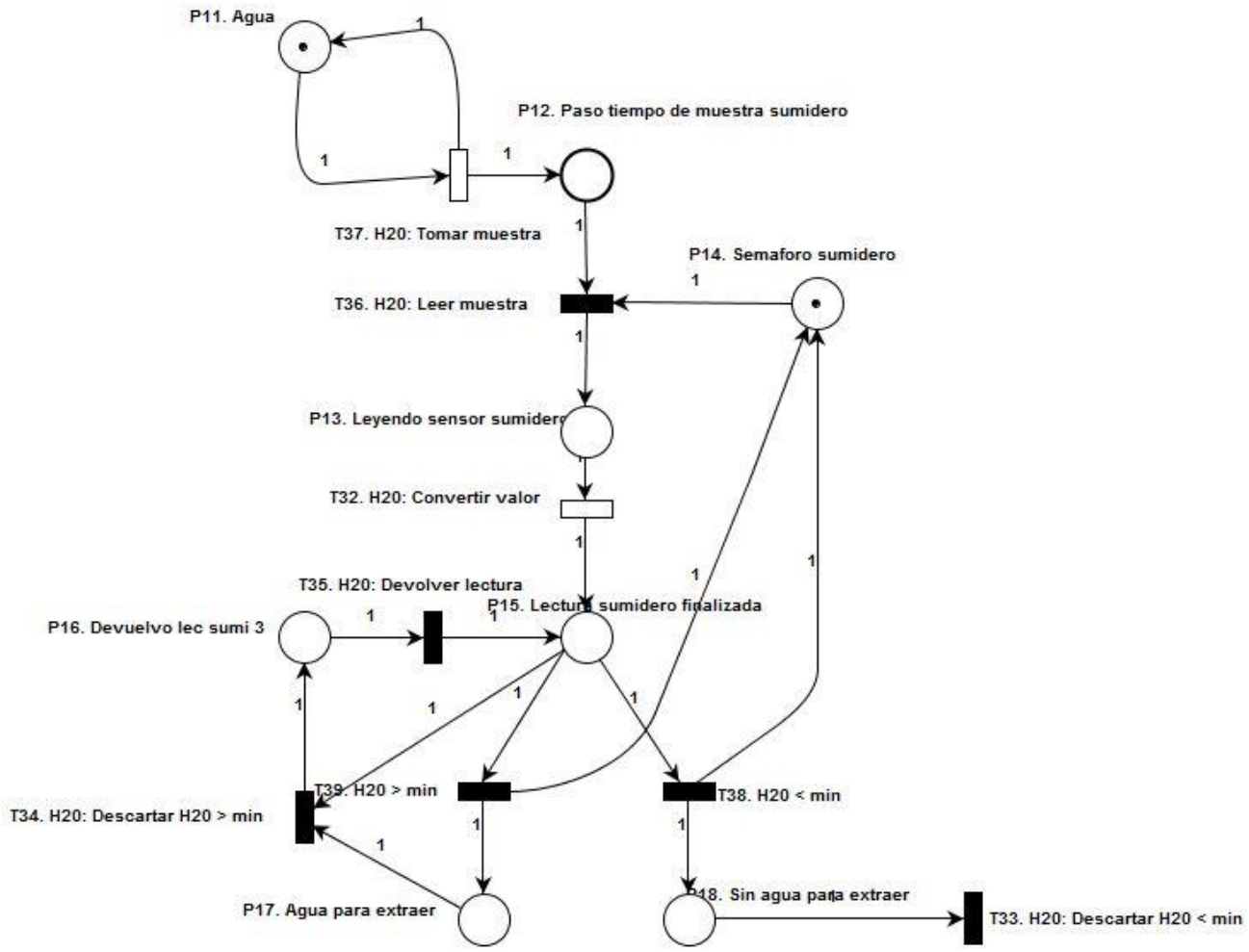
Sensor de metano

Simplemente consta de un generador de muestras (el tiempo de disparo de la transición será el tiempo cada el cual el sensor tome muestras). Habrá dos posibles resultados: metano ALTO o metano BAJO, y la bomba operará de acuerdo a él.



Sensor de nivel de agua

Mismo funcionamiento que el sensor de nivel de metano. También consta de un generador de muestras y otorga uno de dos posibles resultados.

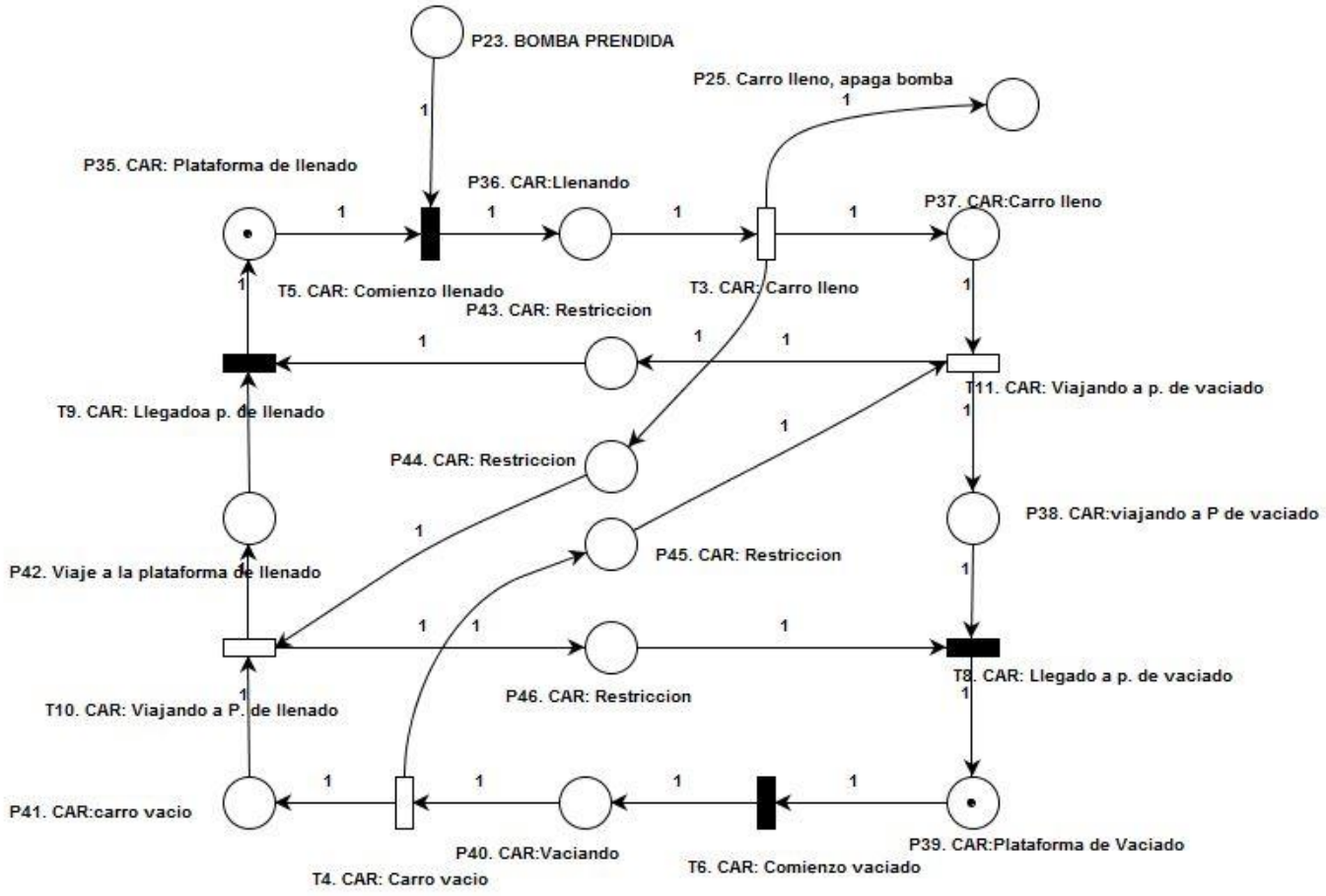


La transición T33 solo se encuentra presente para eliminar muestras antiguas. De la misma manera, las transiciones T34 y T35 están para eliminar una muestra antigua antes de generar una nueva.

Sistema de carritos

Se tendrán dos plataformas: una de llenado y otra de vaciado. A su vez, habrá por supuesto dos carritos, viajando entre una y otra constantemente. Las plazas de

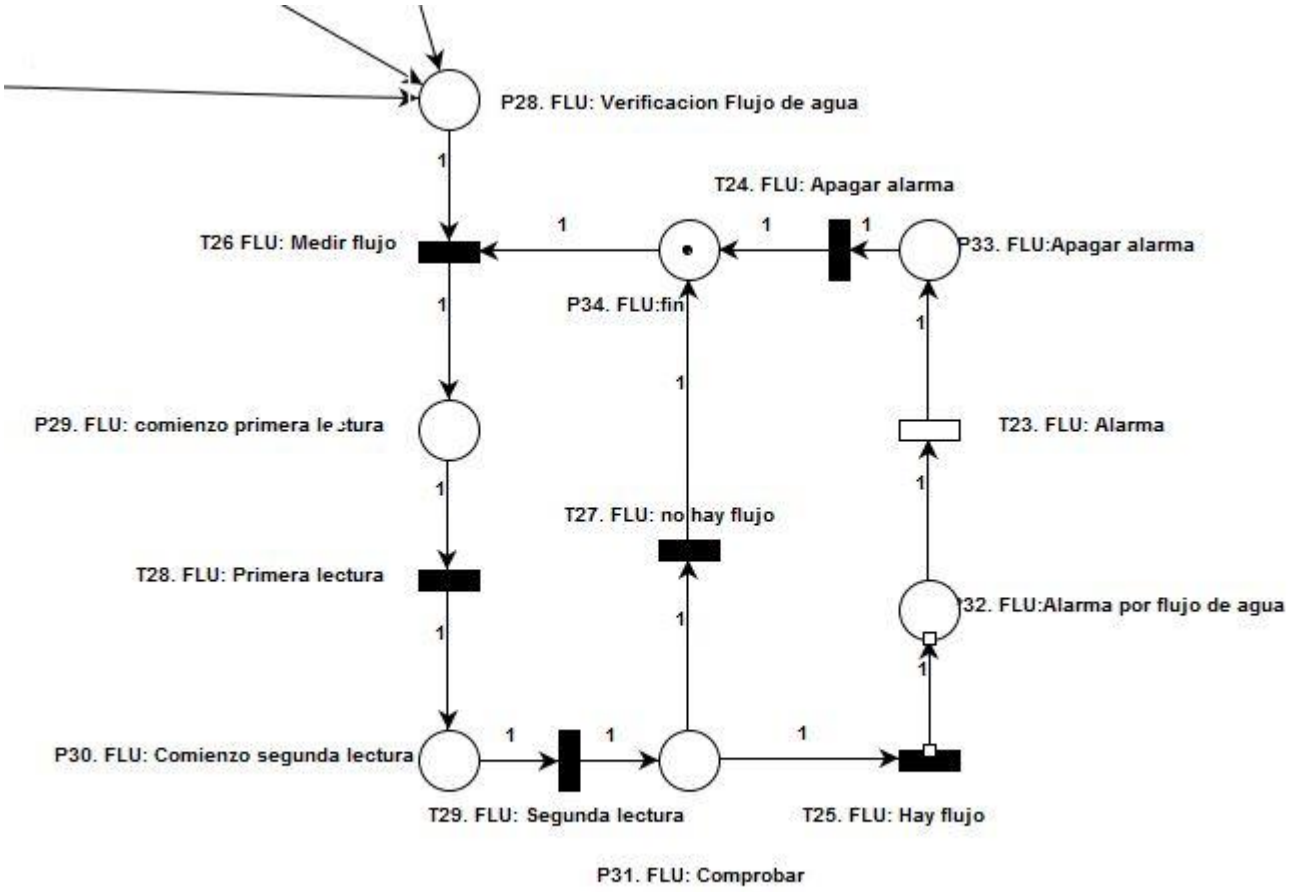
restricción aseguran que un carrito no saldrá de la plataforma de llenado antes de que el otro esté listo para partir de la de vaciado, y viceversa.



Debe tenerse en cuenta que las dos plazas que interactúan directamente con la bomba (bomba prendida y carro lleno), están solo parcialmente representadas, no mostrándose el resto de interacciones para simplificar el diagrama.

Sensor de flujo de agua

El mismo solo se activa cuando hay un *token* en la plaza P28. Y son los transiciones *bloquear por metano*, *bloquear por nivel y bloquear por carrito* las encargadas de poner uno en ese lugar, para sensar el flujo cada vez que se apague la bomba. Por otra parte, cuando haya flujo (y no deba haberlo), se encenderá una alarma, representada por la plaza P32.



Bomba

La misma depende del resto de los actores, pero nuevamente, se han omitido las relaciones, solamente dejando aquellas plazas y transiciones esenciales para la comprensión de su funcionamiento.

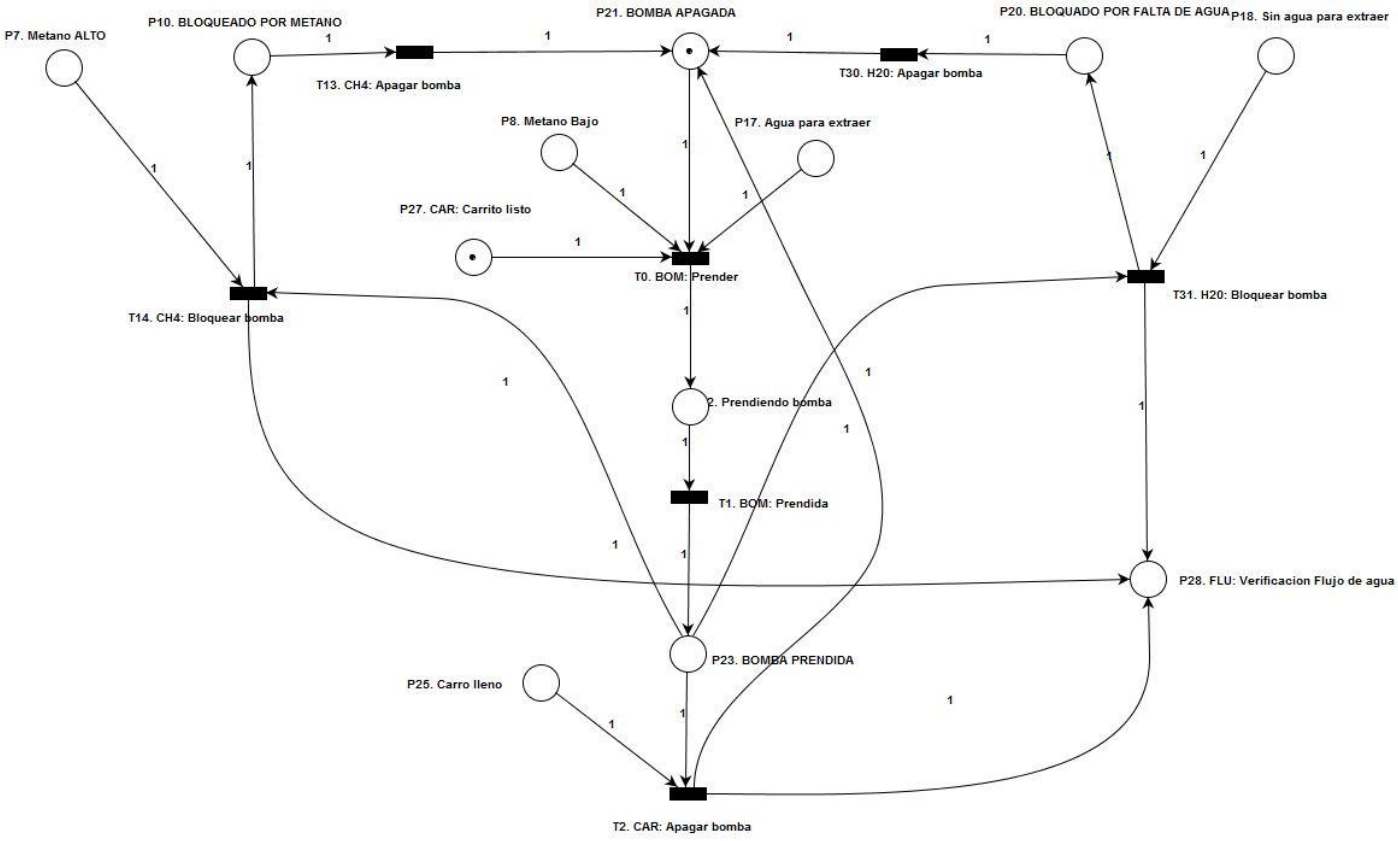
Se observa nuevamente que las condiciones para prender la bomba son:

1. Carrito listo (en plataforma de llenado).
2. Metano bajo (menor al límite).
3. Agua para extraer (mayor al mínimo).

A su vez, la misma puede pasar al estado *bomba apagada* por tres motivos:

1. Disparo de T2 (cuando el carrito está lleno).
2. Disparo de T14 (hay un nivel de metano mayor al límite tolerado).
3. Disparo de T31 (hay un nivel de agua menor al mínimo).

El extracto de red para esta parte del sistema es el siguiente:



Diseño

Diagramas de clases

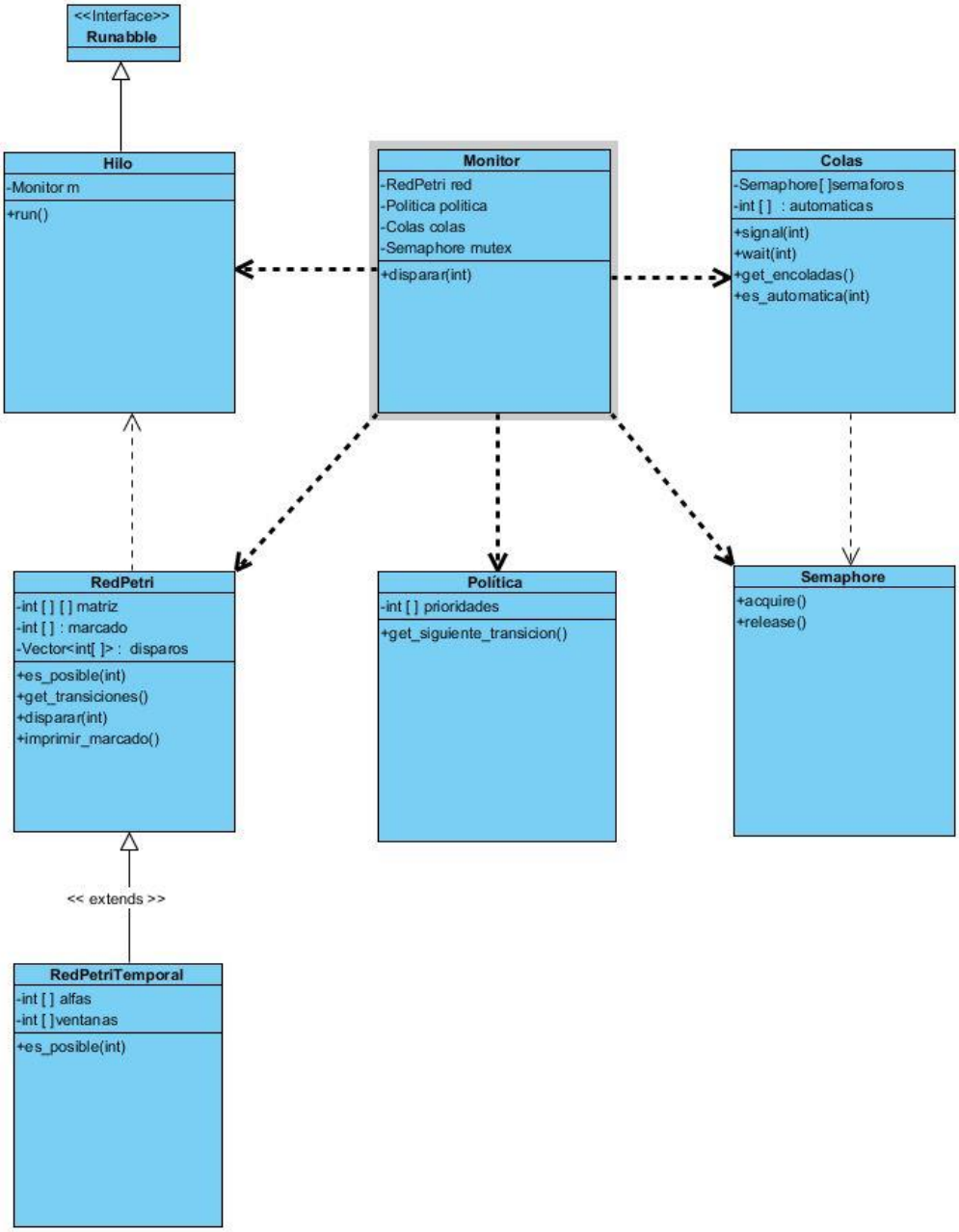
Se presenta a continuación el diagrama de clases del sistema. El mismo consta de:

1. *Hilo*: Se implementó una sola clase hilo que disparará una transición diferente dependiendo del "tipo de hilo que sea". Habrá cuatro hilos en nuestro programa:

    a. Sensor de metano.
    b. Sensor de flujo de agua.
    c. Sensor de nivel de agua.
    d. Carritos.

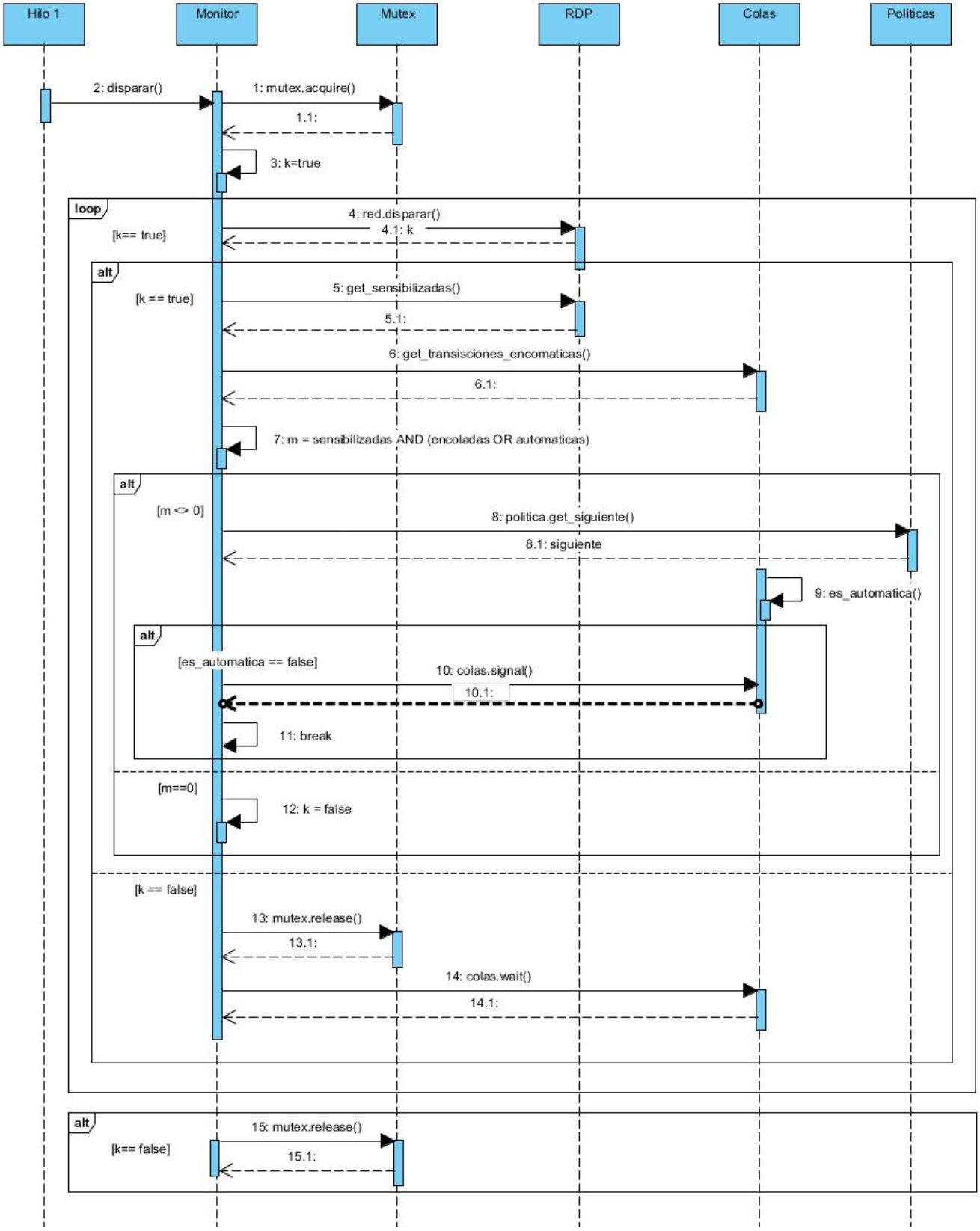
2. *RedPetri*: Contendrá la matriz de incidencia de la red, así como el marcado inicial y toda la lógica necesaria para disparar las transiciones, e informar del nuevo estado de la red una vez disparadas.
3. *Monitor*: Controla el acceso a la red por parte de los hilos. Garantiza por un lado que no haya acceso simultáneo a la red, y por otro maneja los hilos y decide que transiciones disparar, preguntando a la red que transiciones están disponibles, a las colas quienes están esperando, y a la política que transición de las disponibles tiene la prioridad.

4. *Colas*: Se mantiene una cola (semáforo) por cada transición que no sea automática. Informa al monitor que transiciones tienen hilos esperando para ser disparadas y cuales son transiciones automáticas.
5. *Política*: Decide cuál de las transiciones disponibles será la siguiente en dispararse, teniendo en cuenta las prioridades determinadas con anterioridad.
6. *Semaphore*: Clase de java que implementa un semáforo.
7. *RedPetriTemporal*: Clase que hereda de *RedPetri*, para cuando se tenga una red temporal, en la cual haya transiciones que demoran cierto tiempo para dispararse luego de ser sensibilizadas. Solo tiene el método *es_posible*(), el cual verifica no solamente que una transición temporal esté sensibilizada sino también que el tiempo de la misma haya pasado antes de ser disparada.

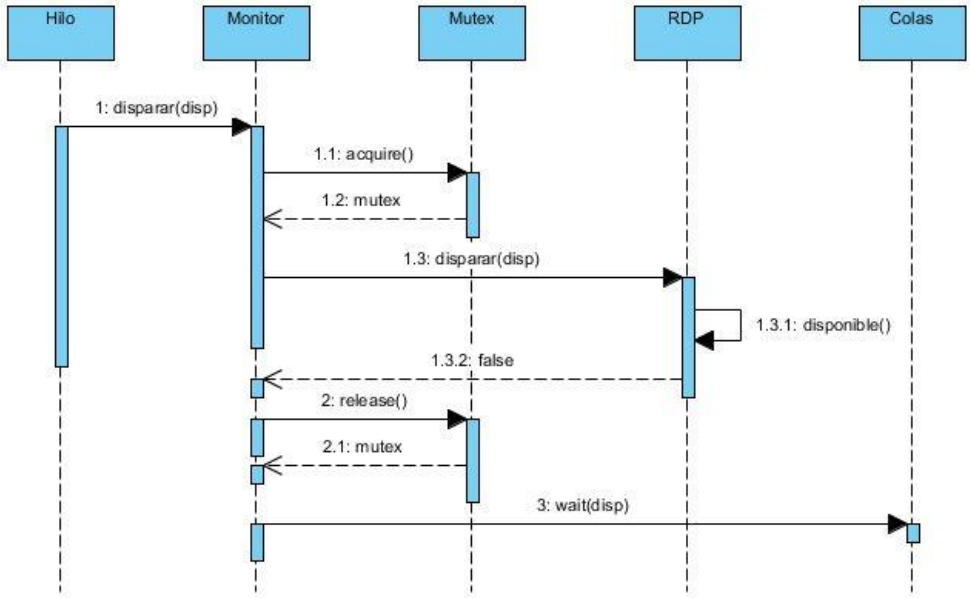


Diagramas de secuencia

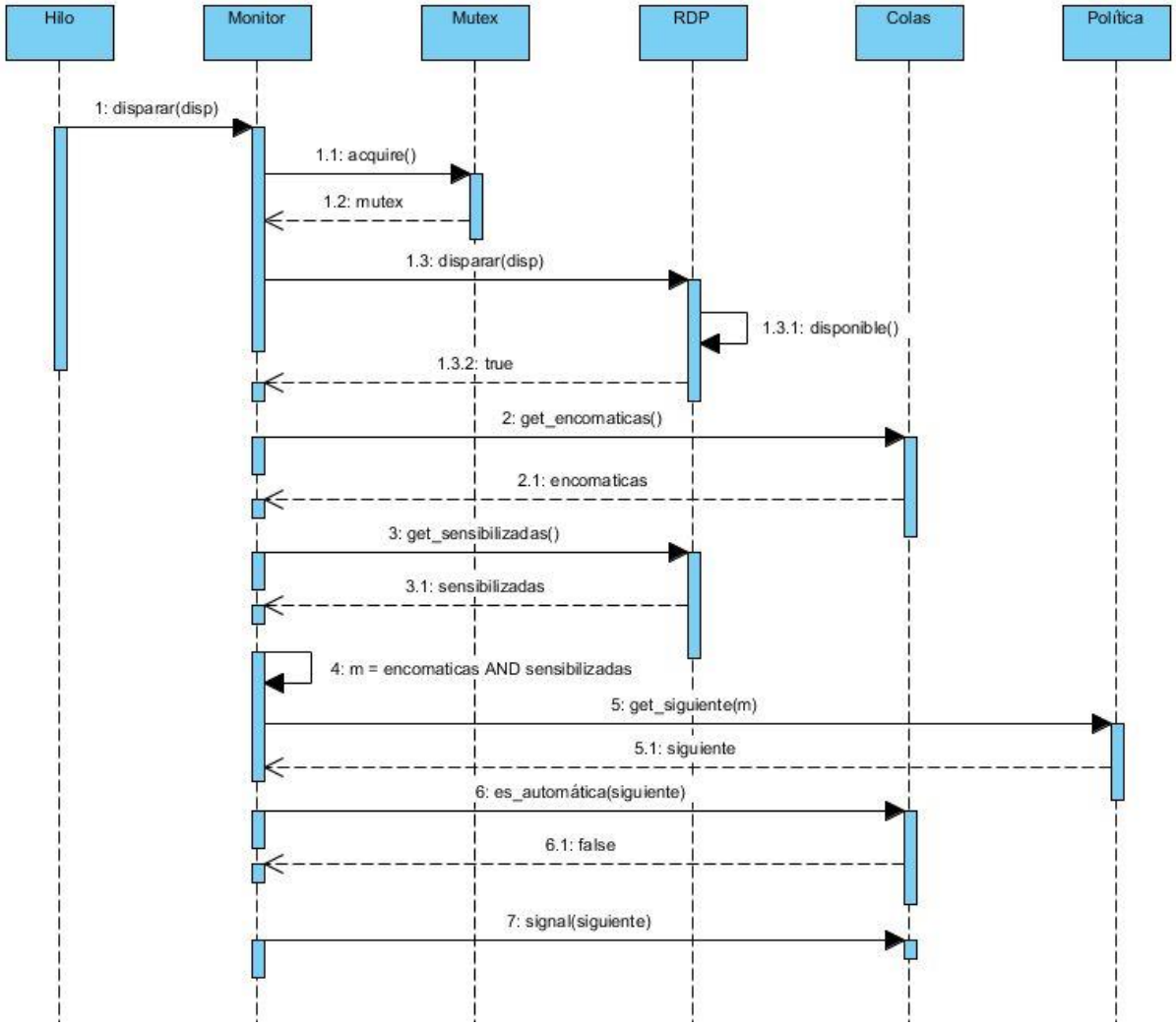
Diagrama del monitor y su relación con el resto de las clases:



Disparo de una transición no sensibilizada



sd Disparo de una transición no automática



Tests

Tests de la red

Antes de pasar a la etapa de codificación, se hicieron Tests de la red para poder asegurar en la mayor medida posible su correcto funcionamiento. Para realizar esto, simplemente se plantearon todos los estados posibles de la bomba (encendida y apagada) y se simularon los eventos que pueden alterar estos estados.

Se muestra a continuación todos los casos posibles y el comportamiento **deseado** del sistema ante cada evento:

	Condiciones	Estado de la bomba	
		Bomba apagada	Bomba prendida
1	C1: Metano alto.	Descarta muestra y alarma por metano	Bloqueo, apago bomba y alarma.
2	C2: Metano bajo.	Con posibilidad de ser prendida.	No pasa nada.
3	C3: Sin agua por extraer.	Descarta muestra.	Bloqueo y apago bomba.
4	C4: Agua por extraer.	Con posibilad de ser prendida.	No pasa nada.
5	C5: Carro vacío en plataforma de llenado.	Con posibilidad de ser prendida.	Comienza a llenarse el carro.
6	C6: Carro lleno en plataforma de llenado	No pasa nada.	Se apaga la bomba.
7	C2: Metano bajo. C4: Nivel agua alto. C5: Carros vacíos en plataforma de llenado.	Prender bomba	No pasa nada.

Entonces, se hicieron varios tests para estos casos, y se muestran algunos a continuación:

Metano alto:

Bomba apagada	Bomba prendida
CH4: Tomar muestra	BOM: Prender
CH4: Leer muestra	BOM: Prendida
CH4: Convertir valor	CAR: Comienzo llenado
CH4 > lim	CAR: Devolver bomba
CH4: Descartar CH4 > lim	CH4: Demora
CH4: Alarma	CH4: Tomar muestra
	CH4: Leer muestra
	CH4: Convertir valor
	CH4 > lim
	CH4: Bloquear bomba
	CH4: Apagar bomba
	FLU: Medir flujo

Observamos cómo, cuando la bomba está apagada y viene una muestra de metano alto, simplemente se descarta la muestra y se enciende la alarma, sin realizar ninguna otra acción y sin ser afectada la bomba.

Por el contrario, cuando se dispara la transición *CH4 < lim*, se observa que se disparan directamente a continuación las transiciones *bloquear bomba* y *apagar bomba*.

1. Sin agua por extraer:

Bomba apagada	Bomba prendida
H20: Tomar muestra	CH4 < lim
H20: Leer muestra	H20: Leer muestra
H20: Convertir valor	H20: Convertir valor
H20 < min	H20 > min
H20: Descartar H20 < min	BOM: Prender
	BOM: Prendida
	H20: Demora
	H20: Tomar muestra
	CAR: Comienzo llenado
	CAR: Devolver bomba
	H20: Leer muestra
	H20: Convertir valor
	H20 < min
	H20: Bloquear bomba
	H20: Apagar bomba
	FLU: Medir flujo

En este caso, cuando se dispara la transición *H2O < min* y la bomba está apagada, simplemente se descarta la muestra.

Por el contrario, cuando se da esto y la bomba está prendida, vemos que se apaga instantáneamente, disparándose la transición *apagar bomba*.

2. Carrito en plataforma de llenado:

Bomba apagada	Bomba prendida
No sucede nada.	CAR: Llegado a plata forma de vaciado
	CAR: Llegado hacia plataforma de llenado
	H20: Demora
	H20: Tomar muestra
	H20: Leer muestra
	H20: Convertir valor
	H20 > min
	CH4: Tomar muestra
	CH4: Leer muestra
	CH4: Convertir valor
	CH4 < lim
	BOM: Prender
	BOM: Prendida
	CAR: Comienzo llenado

Cuando el carrito llega a la plataforma de llenado pero la bomba está apagada, el carrito simplemente se queda ahí hasta que la misma se encienda, por lo cual nada sucede.

Por otro lado, cuando la bomba está prendida (y se dieron el resto de las condiciones necesarias, como *H2O > lim* y *CH4 < lim*, la misma se enciende y se comienza el llenado.

3. Carrito lleno:

Bomba apagada	Bomba prendida
No sucede nada.	BOM: Prender
	BOM: Prendida
	CAR: Comienzo llenado
	CAR: Devolver bomba
	CAR: Carro lleno
	CAR: Apagar bomba
	FLU: Medir flujo

De la misma manera que para el caso anterior, no sucede nada cuando la bomba está apagada.

Cuando la bomba está prendida, y se dispara la transición *carro lleno*, se disparar automáticamente *apagar bomba*, cumpliendo con los requerimientos.

4. Cumplimiento de las tres condiciones de encendido:

Bomba apagada	Bomba prendida
CAR: Llegado a plata forma de vaciado	No pasa nada.
CAR: Llegado hacia plataforma de llenado	
CH4: Tomar muestra	
H20: Tomar muestra	
H20: Leer muestra	
CH4: Leer muestra	
H20: Convertir valor	
H20 > min	
CH4: Convertir valor	
CH4 < lim	
BOM: Prender	
BOM: Prendida	

Vemos como, cuando la bomba está apagada y se dan disparan las transiciones necesarias (*llegado a plataforma de llenado, H20 > min* y *CH4 < lim*), la bomba se enciende.

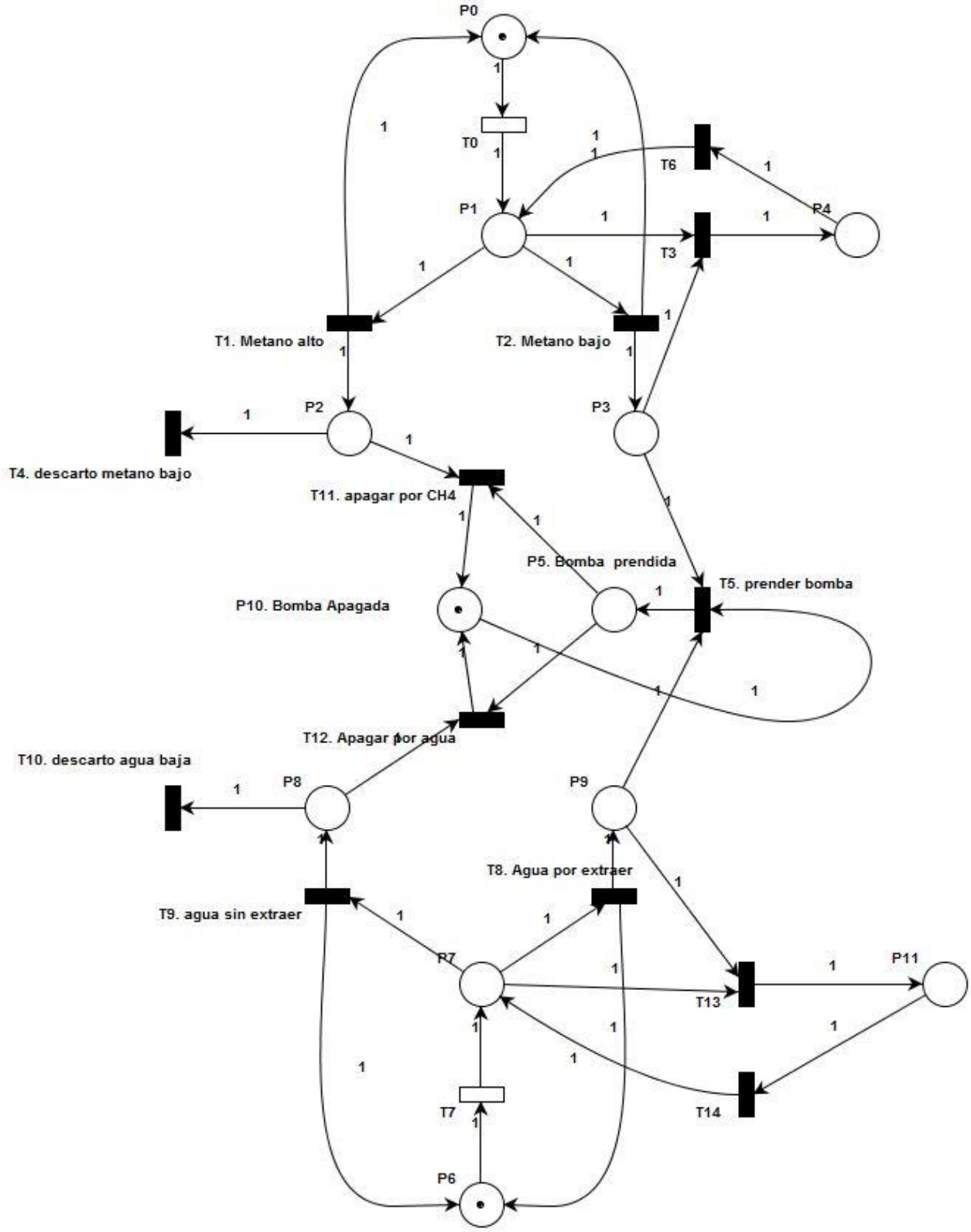
Tests del programa

Para verificar que el programa realiza lo que tiene que realizar y cumple con los requerimientos, se creó una red de Petri alternativa, la cual representa una versión simplificada de la bomba.

Para comprender los resultados de los tests, es necesario saber:

1. T0, T5 y T7 son disparadas por hilos, con lo cual no son automáticas, y tendrán una cola de hilos asociada.
2. El resto de las transiciones son automáticas.
3. T4 y T10 son transiciones con tiempo (500 ms), lo que quiere decir que, una vez sensibilizadas, solo podrán ser disparadas pasados 500 ms.

La red de Petri de este sistema sensibilizado es el siguiente:



Se muestran a continuación los resultados de los tests:

1. Test 1: Disparar una transición no sensibilizada

Para realizar esto, se guarda el marcado antes de intentar disparar la transición, luego se intenta dispararla, y luego se comprueba los marcados, debiendo ser estos iguales ya que la transición no debería haberse podido disparar. El resultado del test fue:

```
-----------------------------------------------------------------------------------------------------------------------

Test 1: Disparar transicion no sensibilizada.

Marcado: [1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0] | Plazas con token: [ P0 P6 P10 ] | Transiciones sensibilizadas: [ T0 T7 ]
main intneta disparar T3
Marcado: [1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0] | Plazas con token: [ P0 P6 P10 ] | Transiciones sensibilizadas: [ T0 T7 ]

-----------------------------------------------------------------------------------------------------------------------
```

2. Test 2: Disparar transiciones sensibilizadas

Se intentan disparar una serie de transiciones, para comprobar que el marcado de la red cambia de acorde a lo deseado. Antes de hacer nada, se guardó el marcado que se quiere obtener al terminar de disparar todas las transiciones, y al final se obtiene el marcado final, y se lo compara con el deseado. El test dio lo siguiente:

```
-----------------------------------------------------------------------------------------------------------------------

Test 2: Disparar transiciones sensibilizadas.

Marcado: [1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0] | Plazas con token: [ P0 P6 P10 ] | Transiciones sensibilizadas: [ T0 T7 ]
main dispara T7
Marcado: [1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0] | Plazas con token: [ P0 P7 P10 ] | Transiciones sensibilizadas: [ T0 T8 T9 ]
main dispara T8
Marcado: [1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0] | Plazas con token: [ P0 P6 P9 P10 ] | Transiciones sensibilizadas: [ T0 T7 ]
main dispara T7
Marcado: [1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0] | Plazas con token: [ P0 P7 P9 P10 ] | Transiciones sensibilizadas: [ T0 T8 T9 T13 ]
main dispara T13
Marcado: [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1] | Plazas con token: [ P0 P10 P11 ] | Transiciones sensibilizadas: [ T0 T14 ]
main dispara T14
Marcado: [1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0] | Plazas con token: [ P0 P7 P10 ] | Transiciones sensibilizadas: [ T0 T8 T9 ]
main dispara T9
Marcado: [1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0] | Plazas con token: [ P0 P6 P8 P10 ] | Transiciones sensibilizadas: [ T0 T7 T10 ]

-----------------------------------------------------------------------------------------------------------------------
```

3. Test 3: Obtener transiciones sensibilizadas

Se dispara una transición para comprobar que el vector de transiciones sensibilizadas luego del disparo es el deseado:

```
-----------------------------------------------------------------------------------------------------------------------

Test 3: Obtener transiciones sensibilizadas.

Transiciones sencibilizdas inicialmente:

Marcado: [1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0] | Plazas con token: [ P0 P6 P10 ] | Transiciones sensibilizadas: [ T0 T7 ]
main dispara T0
Marcado: [0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0] | Plazas con token: [ P1 P6 P10 ] | Transiciones sensibilizadas: [ T1 T2 T7 ]

-----------------------------------------------------------------------------------------------------------------------
```

4. Test 4: Verificar que se dispara la transición de mayor prioridad:

Teniendo en cuenta que el vector de prioridades es el siguiente:

{1,1,1,2,2,4,1,1,1,1,2,5,5,2,1}

Se observa que, al momento de calcular la mayor prioridad, al final (cuando T0, T5 y T7) están sensibilizadas, la política nos dice que la siguiente transición a dispararse será T5, dado que tiene una prioridad de 4, mientras que T0 y T7 tienen una prioridad de 1.

El test compara el valor esperado "5" con el valor retornado por la política, corroborándose que funciona correctamente.

```
-----------------------------------------------------------------------------------------------------------------------

Test 4: Verificar que se dispare la transición de mayor prioridad a continuación

main dispara T0
Marcado: [0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0] | Plazas con token: [ P1 P6 P10 ] | Transiciones sensibilizadas: [ T1 T2 T7 ]
main dispara T2
Marcado: [1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0] | Plazas con token: [ P0 P3 P6 P10 ] | Transiciones sensibilizadas: [ T0 T7 ]
main dispara T7
Marcado: [1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0] | Plazas con token: [ P0 P3 P7 P10 ] | Transiciones sensibilizadas: [ T0 T8 T9 ]
main dispara T8
Marcado: [1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0] | Plazas con token: [ P0 P3 P6 P9 P10 ] | Transiciones sensibilizadas: [ T0 T5 T7 ]
Siguiente transición: 5

-----------------------------------------------------------------------------------------------------------------------
```

5. Test 5: Verificar que los hilos van a las colas de cada transición

En este caso, se crea un hilo que intenta disparar una transición no sensibilizada, y se observa al final como la cola asociada a esa transición se ha incrementado en uno, por lo cual el hilo permanece dormido hasta que esa transición sensibilice y alguien lo despierte.

```
-----------------------------------------------------------------------------------------------------------------------

Test 5: Verificar que los hilos van a las colas asociadas a las transiciones


Estado de colas ANTES de que el Hilo-5 intente disparar T5:
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

Hilo-5 TIENE el mutex
Hilo-5 se DUERME
Hilo-5 SUELTA el mutex

Estado de colas DESPUES de que el Hilo-5 intente disparar T5:
[0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0]

-----------------------------------------------------------------------------------------------------------------------
```

6. Test 6: Disparo de una transición NO sensibilizada por tiempo

Se disparan transiciones hasta que se sensibilice T4 (recordar que T4 es una transición con tiempo), e inmediatamente se obtienen las transiciones sensibilizadas, preguntando por T4. Veremos que nos informa que T4 está sensibilizada pero faltan 500 segundos para poder dispararse.

Se comprueban los marcados para ver que el intento de disparo de T4 no cambió el marcado, puesto que la misma no está lista para dispararse.

```
Test 6: Verificar si se puede disparar una transición sensibilizada pero cuyo tiempo aún no se ha cumplido.

main dispara T0
Marcado: [0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0] | Plazas con token: [ P1 P6 P10 ] | Transiciones sensibilizadas: [ T1 T2 T7 ]
main dispara T1
Marcado: [1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0] | Plazas con token: [ P0 P2 P6 P10 ] | Transiciones sensibilizadas: [ T0 T7 ]
Se obtienen las transiciones sensibilizadas
La transición T4 está sensibilizada pero faltan 500ms

------------------------------------------------------------------------------------------------------------------------
```

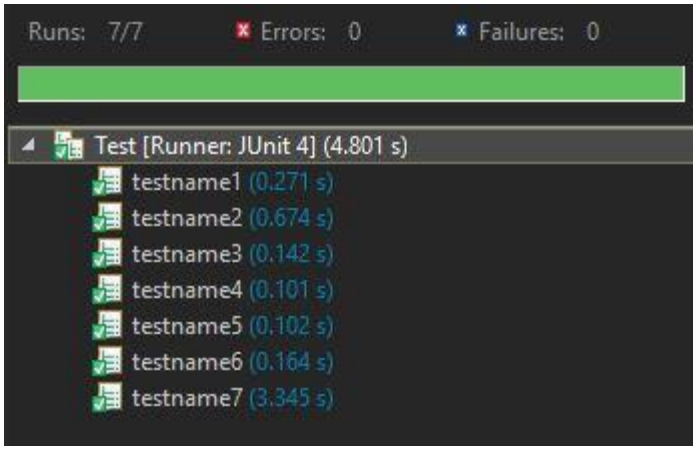
7. Test 7: Disparo de una transición sensibilizada por tiempo

En este caso, se realiza el mismo procedimiento que para el test 6, solo que esta vez, luego de sensibilizar T4, se espera el tiempo necesario (500 ms), hasta que la misma esté lista para dispararse.

Luego se intenta dispararla, y se comprueba los marcados para ver que el mismo cambió acorde a lo esperado y que T4 fue efectivamente disparada.

```
------------------------------------------------------------------------------------------------------------------------

Test 7: Verificar si se puede disparar una transición sensibilizada cuyo tiempo ya se ha cumplido.

main dispara T0
Marcado: [0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0] | Plazas con token: [ P1 P6 P10 ] | Transiciones sensibilizadas: [ T1 T2 T7 ]
main dispara T1
Marcado: [1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0] | Plazas con token: [ P0 P2 P6 P10 ] | Transiciones sensibilizadas: [ T0 T7 ]
La transición T4 está sensibilizada pero faltan 499ms

pasan los 500 ms

La transición T4 ha terminado su tiempo y está sensibilizada
Marcado: [1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0] | Plazas con token: [ P0 P2 P6 P10 ] | Transiciones sensibilizadas: [ T0 T4 T7 ]
main dispara T4
Marcado: [1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0] | Plazas con token: [ P0 P6 P10 ] | Transiciones sensibilizadas: [ T0 T7 ]

------------------------------------------------------------------------------------------------------------------------
```

Observamos entonces que todos los tests fueron exitosos:



Si bien el hecho de que todos los tests hayan tenido éxito no garantiza un perfecto funcionamiento de la red, podemos afirmar que la falla de uno de ellos garantizaría el mal funcionamiento de la misma.

Tests del sistema

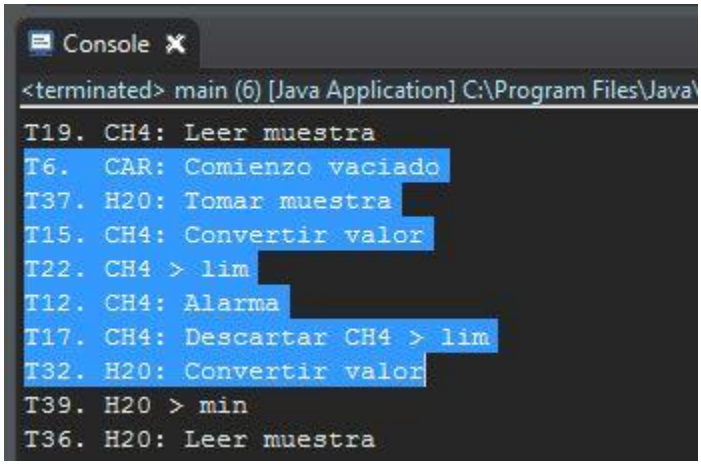
Una vez comprobado que el programa disparar las redes como se espera, procedemos a realizar los mismos tests que se realizaron con la red de Petri con anterioridad, pero esta vez en el programa. Se modificaron los parámetros de impresión del programa para que solo muestre las transiciones disparadas, y no el manejo de hilos y de tiempo.

Nuevamente, los resultados esperados ante los diversos eventos pueden verse en el cuadro de la sección *tests de la red*

1. Metano alto:

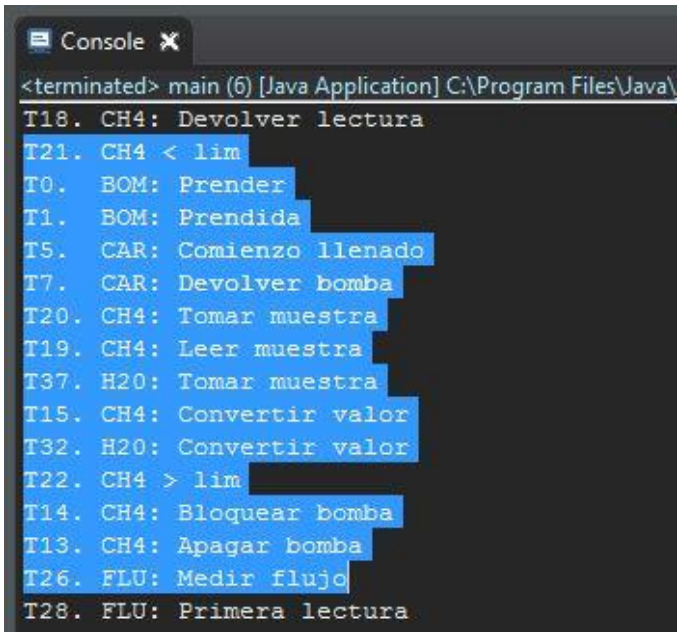
Con la bomba apagada, solo se enciende una alarma y se descarta la muestra:

Bomba apagada
T6. CAR: Comienzo vaciado
T37. H20: Tomar muestra
T15. CH4: Convertir valor
T22. CH4 > lim
T12. CH4: Alarma
T17. CH4: Descartar CH4 > lim
T32. H20: Convertir valor



Con la bomba prendida, la misma debe bloquearse y apagarse:

Bomba prendida
T21. CH4 < lim
T0. BOM: Prender
T1. BOM: Prendida
T5. CAR: Comienzo llenado
T7. CAR: Devolver bomba
T20. CH4: Tomar muestra
T19. CH4: Leer muestra
T37. H20: Tomar muestra
T15. CH4: Convertir valor
T32. H20: Convertir valor
T22. CH4 > lim
T14. CH4: Bloquear bomba
T13. CH4: Apagar bomba
T26. FLU: Medir flujo



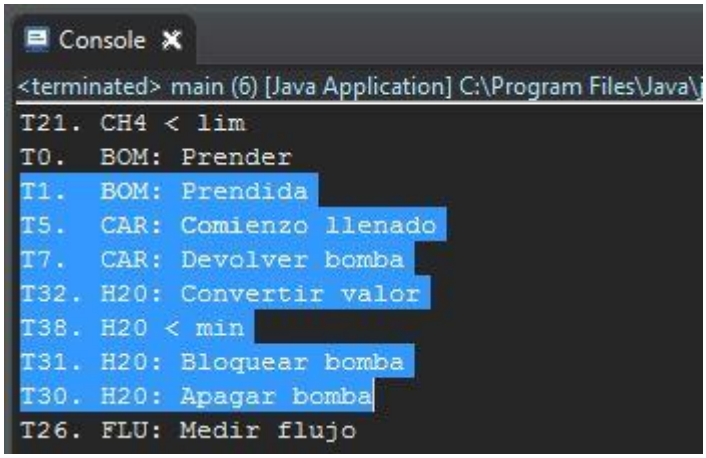
2. Sin agua por extraer:

Con la bomba apagada, simplemente se descarta la muestra:

Bomba apagada
T36. H20: Leer muestra
T20. CH4: Tomar muestra
T19. CH4: Leer muestra
T37. H20: Tomar muestra
T24. FLU: Apagar alarma
T32. H20: Convertir valor
T38. H20 < min
T33. H20: Descartar H20 < min

Con la bomba prendida, la bomba debe bloquearse y apagarse:

Bomba prendida
T0. BOM: Prender
T1. BOM: Prendida
T5. CAR: Comienzo llenado
T7. CAR: Devolver bomba
T32. H20: Convertir valor
T38. H20 < min
T31. H20: Bloquear bomba
T30. H20: Apagar bomba

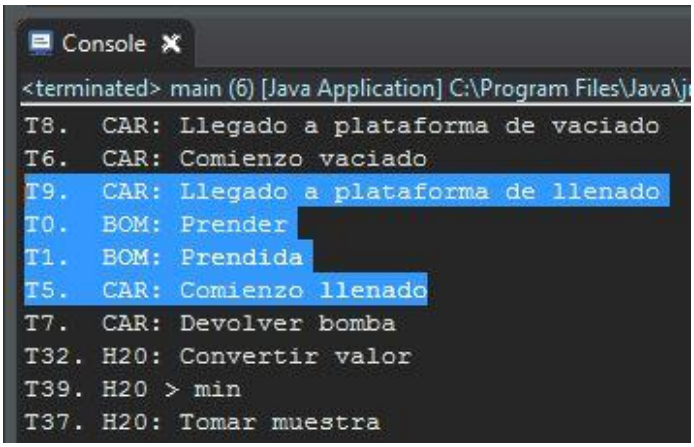


3. Carrito en plataforma de llenado:

Con la bomba apagada, no sucede nada, ya que quiere decir que no se han dado las otras dos condiciones para que se encienda la bomba y se comience el llenado.

Por el contrario, cuando la bomba está prendida, se comienza inmediatamente a llenar el carrito, como vemos a continuación:

Bomba prendida
T8. CAR: Llegado a plataforma de vaciado
T6. CAR: Comienzo vaciado
T9. CAR: Llegado a plataforma de llenado
T0. BOM: Prender
T1. BOM: Prendida
T5. CAR: Comienzo llenado
T7. CAR: Devolver bomba
T32. H20: Convertir valor

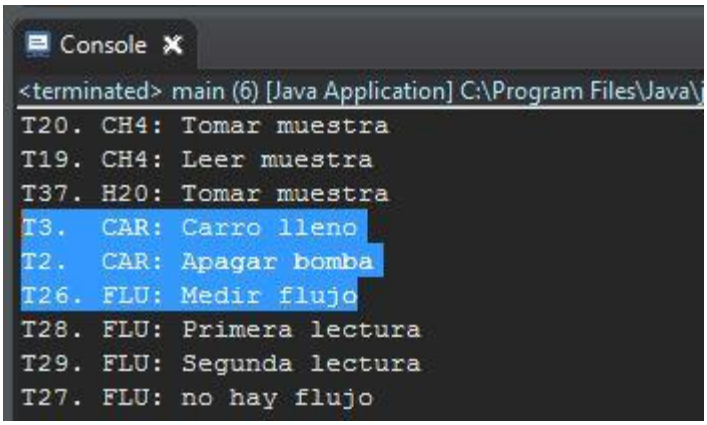


4. Carrito lleno:

Cuando la bomba está apagada, nunca puede darse el caso de que el carro se llene.

Por otro lado, cuando ésta está prendida, un carrito lleno debería implicar un inmediato apagado de la bomba:

Bomba prendida
T20. CH4: Tomar muestra
T19. CH4: Leer muestra
T37. H20: Tomar muestra
T3. CAR: Carro lleno
T2. CAR: Apagar bomba
T26. FLU: Medir flujo
T28. FLU: Primera lectura
T29. FLU: Segunda lectura



5. Cumplimiento de las tres condiciones de encendido:

Por último, se comprobó que la bomba se encienda si y solo si se han dado las tres condiciones que habíamos especificado, y podemos observar que la misma lo hace exactamente cuándo se han dado las mismas:

Bomba apagada
T15. CH4: Convertir valor
T32. H20: Convertir valor
T21. CH4 < lim
T39. H20 > min
T10. CAR: Viajando a plataforma de llenado
T36. H20: Leer muestra
T8. CAR: Llegado a plataforma de vaciado
T6. CAR: Comienzo vaciado
T9. CAR: Llegado a plataforma de llenado
T0. BOM: Prender
T1. BOM: Prendida
T5. CAR: Comienzo llenado