

# Método para el Modelado y Prueba de Diagramas de Actividades Mediante Redes de Petri

Lionel R. Baquero Hernández, Luis E. Argota Vega, Osviel Rodríguez Valdés, Febe A. Ciudad Ricardo  
Facultad 6. Universidad de las Ciencias Informáticas  
La Habana, Cuba  
lrbaquero@estudiantes.uci.cu, {leargota, osvriel, fciudad}@uci.cu

*Resumen*— Las redes de Petri son ideales para describir y estudiar sistemas que procesan información y con características concurrentes, asíncronas, distribuidas, paralelas, no determinísticas y/o estocásticas. Sin embargo, la teoría clásica de estas redes se enfrenta al desafío de adecuarse a las nuevas necesidades de los sistemas informáticos que los ingenieros diseñan en la actualidad. Un diagrama de actividades representa una secuencia de actividades. Podría aplicarse a cualquier propósito, pero se considera especialmente útil para visualizar los flujos de trabajo y los procesos del negocio, o casos de uso. El presente trabajo se ha desarrollado con el objetivo de diseñar un método para el modelado y prueba de diagramas de actividades mediante redes de Petri. El método propuesto brinda una serie de pasos necesarios para modelar un diagrama de actividades con redes de Petri y a partir de dicha red obtener una representación matemática de la misma. Esta representación matemática se utiliza para con ella realizar una serie de operaciones basadas en el marcaje de la red que permiten realizar pruebas al diagrama. Este método es capaz de identificar errores comunes que se cometen en el modelado de los diagramas de actividades, lo cual se demostró a través del método de experimentación.

*Palabras clave*— método, diagramas de actividades, redes de Petri, modelado, prueba.

## I. INTRODUCCIÓN

Las redes de Petri (PN, por sus siglas en inglés) son un formalismo que surge en 1962 con el trabajo doctoral de Carl Adam Petri "Kommunikation mit Automaten" (Comunicación con Automatas), en Alemania [1]. Una PN puede ser definida como una herramienta gráfica y matemática de modelación que se puede aplicar en muchos sistemas. Particularmente son ideales para describir y estudiar sistemas que procesan información y con características concurrentes, asíncronas, distribuidas, paralelas, no determinísticas y/o estocásticas [2]. Desde su aparición hace ya más de cinco décadas se han realizado numerosos trabajos que han desarrollado una estable teoría para la especificación y verificación de sistemas concurrentes. Sin embargo, la teoría clásica de redes se enfrenta al desafío de adecuarse a las nuevas necesidades de los sistemas informáticos que los ingenieros diseñan en la actualidad [3].

Como parte del enfrentamiento al desafío de adecuar la teoría clásica de las PN a las nuevas necesidades recientemente se desarrolló una investigación que las vincula con los diagramas de casos de uso (DCU). En la mencionada investigación se diseñó un Método para el modelado y pruebas de diagramas de casos de uso mediante redes de Petri que se basa en una adaptación realizada a la definición formal de dichas redes específica para este tipo de diagramas [4]. Básicamente el método consiste en la transformación de un DCU en una PN, para luego mediante su definición formal

poder comprobar la correctitud del modelado. Con esto se pueden evitar numerosos errores que son cometidos frecuentemente en etapas tempranas del diseño de software.

Un diagrama de actividades representa una secuencia de actividades. Podría aplicarse a cualquier propósito (como para mostrar los pasos de un algoritmo), pero se considera especialmente útil para visualizar los flujos de trabajo y los procesos del negocio, o casos de uso. Uno de los flujos de trabajo (disciplinas) del Proceso Unificado (UP, por sus siglas en inglés) es el Modelado del Negocio; su propósito es entender y comunicar la estructura y la dinámica de la organización en el que se va a desplegar un sistema. Un artefacto clave de la disciplina del Modelado del Negocio es el Modelo de Objetos del Negocio (un superconjunto del Modelo del Dominio del UP), que visualiza esencialmente cómo funciona un negocio utilizando diagramas de clases, secuencia y actividades de UML. De esta manera, los diagramas de actividades se aplican especialmente dentro de la disciplina del Modelado del Negocio del UP [5].

El presente trabajo se ha desarrollado con el objetivo de diseñar un método para el modelado y prueba de diagramas de actividades (DA) mediante PN. Se pretende que con este método se puedan identificar errores comunes que se cometen en el modelado de DA.

## II. MATERIALES Y MÉTODOS

### A. Definición formal de redes de Petri

Para la presente investigación se asumirá como definición formal de redes de Petri la de [6], como se expone a continuación:

Una red de Petri es una cuádrupla  $R=(P, T, \alpha, \beta)$  donde  $P=\{p_1, p_2, \dots, p_n\}$ , es un conjunto finito y no vacío de lugares,  $T=\{t_1, t_2, \dots, t_m\}$  es un conjunto finito y no vacío de transiciones,  $\alpha:P \times T \rightarrow N$ , es la función de incidencia previa,  $\beta:T \times P \rightarrow N$ , es la función de incidencia posterior.

Una red de Petri se representa gráficamente por un grafo dirigido bipartido. Los lugares se representan por círculos, las transiciones por barras. Los lugares y las transiciones se conectan por arcos dirigidos. Un arco dirigido de un lugar  $p_i$  a una transición  $t_j$  existe sí y sólo sí  $\alpha(p_i, t_j) \neq 0$ . Análogamente, un arco dirigido de una transición  $t_x$  a un lugar  $p_i$  existe sí y sólo sí  $\beta(t_x, p_i) \neq 0$ . Cada arco se etiqueta con un entero natural,  $\alpha(p, t)$  o  $\beta(t, p)$ , que se denomina peso del arco. En la figura 1 se muestra un ejemplo de representación gráfica. En esta figura se tiene una red de Petri con dos lugares y dos transiciones, el lugar  $p_1$  está unido de la transición  $t_1$

mediante un arco  $\alpha(p_1, t_1) = 1$ , mientras que la transición  $t_1$  está unida al lugar  $p_2$  por dos arcos, entonces  $\beta(p_2, t_1) = 2$ .

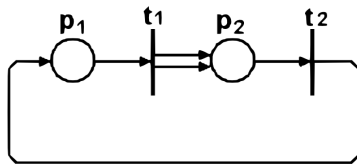


Fig. 1. Ejemplo de una red de Petri

Una red se representa matricialmente por medio de dos matrices. Sea  $|P| = n$  (número de lugares de la red), y sea  $|T| = m$  (número de transiciones de la red). Se denomina matriz de incidencia previa a la matriz:

$$C = [c_{ij}]_{nm} \text{ en la que } [c_{ij}] = \alpha(p_i, t_j). \quad (1)$$

Se denomina matriz de incidencia posterior a la matriz:

$$C^+ = [c'_{ij}]_{nm} \text{ en la que } [c'_{ij}] = \beta(p_i, t_j). \quad (2)$$

Es decir, en las matrices de incidencia los lugares numeran las filas (i) y las transiciones numeran las columnas (j), y cada elemento (i, j) expresa la incidencia que el lugar i tiene sobre la transición j. La representación matricial para la red de la figura 1 es de la siguiente manera:

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, C^+ = \begin{bmatrix} 0 & 1 \\ 2 & 0 \end{bmatrix}$$

Una red de Petri es pura si ninguna transición contiene un lugar que sea simultáneamente de entrada y de salida. La representación matricial de una red pura se simplifica definiendo una única matriz, C, denominada matriz de incidencia:

$$C = C^+ - C \quad (3)$$

Las redes de Petri están dotadas de unos conceptos dinámico-estructurales como el marcado, la sensibilidad y la evolución del marcado. El marcado es la asignación de un número entero no negativo a cada lugar (número de marcas). Si  $|P| = n$ , entonces un marcado se representa, en forma matricial, por un vector de n elementos denominado vector de marcado  $M(p_i)$ . La evolución del marcado le otorga a la PN un comportamiento dinámico que permite modelar evoluciones simultáneas de sistemas discretos. Una transición  $t \in T$  está sensibilizada si:  $M(p) \geq \alpha(p, t), \forall p \in P$ . El disparo de una transición sensibilizada t, es representado por la expresión siguiente:

$$M_j(p) = M_i(p) + \beta(p, t), \forall p \in P \quad (4)$$

En la figura 2 se puede ver la evolución del marcado de una red, una vez la transición  $t_1$  sensibilizada ha sido disparada.

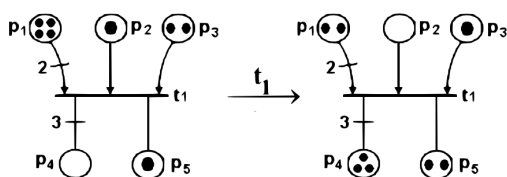


Fig. 2. Ejemplo de evolución del marcado

En las redes de Petri se tiene por lo general una secuencia de disparos a partir de un marcaje inicial  $M_0$ , y se representa por una secuencia de disparo de transiciones tal que:

$$M_0 \xrightarrow{t_1} M_1 \xrightarrow{t_2} \dots \xrightarrow{t_n} M_n = M_0 \xrightarrow{t} M_2$$

Se dirá que la secuencia  $\sigma = t_1 t_2 \dots t_r$  es aplicable a partir de  $M_0$ . A una secuencia de disparos  $\sigma$  se le asocia un vector característico  $\bar{\sigma}$ , cuya i-ésima componente es el número de ocurrencias del disparo de  $t_i$  en la secuencia  $\sigma$ . A partir de la definición de C y de la regla de evolución del marcado se puede obtener la ecuación de estado de una red de Petri (pura y marcada):

$$M_\sigma = M_0 + C \cdot \bar{\sigma} \quad (5)$$

Una red de Petri temporizada, es un par  $(R, Z)$  tal que  $R = (P, T, \alpha, \beta)$  y Z es una función que asigna un número real no negativo,  $z_i$  a cada transición de la red. A este  $z_i$  se le denomina tiempo de disparo de la transición  $t_i$ . La evolución del marcado es idéntica a la de una red de Petri normal, la única diferencia es que el disparo de la transición  $t_i$  tarda  $z_i$  unidades de tiempo en suceder una vez quedó sensibilizada.

### B. Aspectos fundamentales sobre los diagramas de actividades

Un diagrama de actividades ha sido diseñado para mostrar una visión simplificada de lo que ocurre durante una operación o proceso. Es una extensión de un diagrama de estados. El diagrama de estados muestra los estados de un objeto y representa las actividades como flechas que conectan a los estados. El diagrama de actividades resalta, precisamente, a las actividades. A cada actividad se le representa por un rectángulo con las esquinas redondeadas. El procesamiento dentro de una actividad se lleva a cabo, y al realizarse, se continúa con la siguiente actividad. Una flecha representa la transición de una a otra actividad. El diagrama actividad cuenta con un punto inicial (representado por un círculo relleno) y uno final (representado por una diana) [7].

Las ramas se muestran como condiciones de guarda en transiciones o como diamantes con múltiples flechas de salida etiquetadas. Una división o unión de control se representa de la misma manera que en un diagrama de estados, con múltiples flechas que entran o salen de una barra gruesa de sincronización. Un diagrama de actividades puede contener bifurcaciones, así como divisiones de control en hilos concurrentes. Los hilos concurrentes representan actividades que se pueden realizar concurrentemente por los diversos objetos o personas en una organización. La concurrencia se presenta con frecuencia a partir de la agregación, en la cual cada objeto tiene su propio hilo concurrente. [8]. Se podría aclarar, además la existencia de los hilos alternativos que no son más que secuencias de actividades que tienen un fin común.

Los elementos fundamentales de un DA permiten simular mediante su utilización coherente y conjunta procesos reales paso a paso. Con su utilización se adquiere una perspectiva más aterrizada para un mejor entendimiento de lo que se modela y puede servir como guía, entre otros artefactos, para la extracción y recopilación de requisitos. Luego de las consultas bibliográficas pertinentes se puede decir que están compuestos por: punto de inicio, actividades, bifurcaciones, barras de sincronización (que pueden ser de unión o de división), transiciones y punto de fin. Además en estos diagramas se representan tanto hilos concurrentes como alternativos.

### C. Modelado de diagramas de actividades mediante redes de Petri

Para el modelado de diagramas de actividades con PN será necesario transformar cada elemento del diagrama en su

equivalente. Para entender la equivalencia a la que se hace referencia entre los elementos de un DA y una PN, partiendo de su definición formal, será necesario entender que una serie de elementos de vital importancia.

El punto inicial, el punto final y las actividades pertenecerán al conjunto P, lo cual es lo mismo que decir que cada una de las instancias de estos elementos se convertirá en lugares en la PN. Cada lugar correspondiente a un punto final deberá relacionarse con una transición que a su vez se relacionará con el punto inicial. Es de gran importancia no olvidar esto dado que el método que se propone es iterativo.

Las bifurcaciones que suceden a una actividad y de la cual salen n posibles vías alternativas, se representarán como n transiciones. Estas transiciones parten del lugar (perteneciente al conjunto P) que se corresponde con su actividad precedente. El valor de n es tal que  $1 < n < 4$ . Es importante especificar que las bifurcaciones también pueden ser utilizadas para la reunificación de hilos alternativos. Por lo que es válido decir que puede existir el caso en que a una bifurcación entren n hilos alternativos y de esta salgan m posibles vías alternativas. Donde  $1 < n < 4$  y  $0 < m < 3$ . En este caso la solución será generar un lugar  $p \in P$ , seguida de m transiciones  $t \in T$ . Las transiciones de los DA se corresponderán con los arcos (pertenecientes a los conjuntos  $\alpha$  y/o  $\beta$ ) en las PN.

Las barras de sincronización serán representadas en la PN como transiciones (pertenecientes al conjunto T). De estas saldrán o entrarán (en dependencia de si son de división o unión respectivamente) n arcos, uno para cada hilo concurrente que exista. Donde  $n > 1$ . Seguido del arco los lugares pertenecientes a las primeras actividades de los hilos concurrentes, para evitar la existencia de dos transiciones consecutivas. El mismo procedimiento se debe aplicar para los lugares pertenecientes a la última actividad de un hilo concurrente, fusionando la transición que debe seguirle con la que le corresponde a la barra de sincronización de tipo unión que estará luego de esta. Puede existir el caso en que una barra de sincronización aparezca luego de una bifurcación, si esto ocurre se fusionan la transición perteneciente al hilo alternativo con la perteneciente a la barra de sincronización.

Para relacionar los elementos se deberá seguir la misma secuencia del DA, de forma tal que luego de un lugar aparece un arco, seguido de una transición y posteriormente otro arco y otro lugar; y así sucesivamente. En el caso de que al transformar los elementos del DA a sus equivalentes de una PN existan dos lugares consecutivos, se deberá colocar entre ellos una transición; relacionándolos mediante arcos. Esto se hace para respetar la definición formal de las PN que se asume en la investigación. Se deberán colocar estos elementos solo si entre ambos lugares existe un arco que los relacione en el DA.

Una vez entendido lo anteriormente planteado se puede definir una serie de pasos a seguir para el modelado de DA con PN:

1. Comprobar la existencia de un solo punto inicial.
2. Comprobar si existen bifurcaciones luego de una actividad, que estas ofrecen más de una posible vía alternativa.
3. Comprobar si existen bifurcaciones de reunificación que a estas entran más de un hilo alternativo.
4. Comprobar si existen barras de sincronización de tipo división que de esta salgan más de un hilo concurrente.

5. Comprobar si existen barras de sincronización de tipo unión que a esta entren más de un hilo concurrente.
6. Transformar cada elemento del DA en su equivalente de una PN y relacionar los elementos entre sí.

#### D. Pruebas de redes de Petri basadas en diagramas de actividades

Una vez modelado el DA con la utilización de PN será necesario realizar las pruebas pertinentes para evaluar la correctitud del mismo. Un diagrama de actividades es una PN pura y marcada, por lo que para las pruebas será necesario partir de la adaptación de su definición formal propuesta por [4]. Para ello se hace necesario hacer las siguientes anotaciones:

- Para el conjunto P debe cumplirse que la cantidad de lugares es como mínimo igual a 3 ( $|P| \geq 3$ ), dado que en los DA existirán al menos un inicio, un fin y una actividad.
- El vector  $M_0$  tendrá m filas donde  $m = |P|$  (cantidad de lugares), los valores serán 1 en las fila que se correspondan con el lugar asociado al inicio y 0 en otro caso.
- Los vectores  $M_1, M_2, \dots, M_k$  se calcularán a partir de la expresión  $M_k = M_{k-1} + C \cdot \bar{c}_{k-1}$  (6), la cual es una adaptación a la definición formal de las PN. Los valores serán 1 en las filas que se correspondan con los lugares marcados y 0 en otro caso. Donde  $k \leq |P|$ , dado que a partir de  $k = |P|$  el marcaje de la red es cíclico y se repetirá si son sensibilizadas las mismas transiciones.
- El vector característico  $\bar{c}_k$  será actualizado para cada marcaje y los valores serán 1 en las filas donde se corresponda con las transiciones sensibilizadas y 0 en otros casos. Las transiciones se deben sensibilizar de forma tal que si para un lugar existen varias transiciones que puedan ser sensibilizadas, se sensibilice primero la menor o la primera a la izquierda. Para los hilos concurrentes las transiciones serán sensibilizadas concurrentemente. Una vez sensibilizadas todas las transiciones de un hilo de menor cantidad de actividades se debe esperar la sensibilización de todas las existentes en los hilos concurrentes restantes. Una vez sensibilizadas todas las transiciones de todos los hilos concurrentes se pasa a la sensibilización de la transición correspondiente a la barra de sincronización de tipo unión a la cual entran.
- Para el marcaje de la red deberá cumplirse que dos iteraciones consecutivas siempre serán distintas  $M_k \neq M_{k-1}$ . Dado que si dos iteraciones consecutivas coinciden esto podrá ser interpretado como sinónimo de que el marcaje de la PN no ha cambiado de estado. Al identificarse que el marcaje no cambia de estado se puede afirmar que hay algún error en el diseño de la misma o esta no se corresponde con una PN basada en un DA.
- En el marcaje de la red un lugar  $p \in P$  solo puede permanecer marcado por dos iteraciones consecutivas si este se corresponde con una actividad de un hilo concurrente con otro que posee mayor cantidad de actividades. De lo contrario si esto ocurre se puede que el elemento del DA correspondiente a este lugar no está correctamente relacionado con los demás.
- Durante el marcaje de la red todos los lugares  $p \in P$  deben haber sido marcados al menos en una ocasión si todas las transiciones han sido sensibilizadas.

Luego de realizar las acotaciones pertinentes, se mostrarán los pasos necesarios para realizar las pruebas a una PN basada en un DA. Estos pasos deben seguirse teniendo en cuenta las acotaciones anteriormente explicadas, las cuales jugarán un papel fundamental en la ejecución de los siguientes pasos:

1. Verificar que  $|P| \geq 3$ .
2. Hallar  $M_0$ .
3. Hallar C.
4. Hallar  $\bar{\sigma}_0$ .
5. Calcular el marcaje para  $M_1 = M_0 + C \cdot \bar{\sigma}_0$ .
6. Comprobar que  $M_0 \neq M_1$ .
7. Comprobar si algún lugar ha permanecido marcado para ambos vectores de marcado ( $M_0, M_1$ ).
8. Si el paso 7 ha arrojado resultados positivos se debe comprobar si el lugar(es) que ha(n) permanecido marcado(s) se corresponden con un hilo concurrente con otro que posee mayor cantidad de actividades. Si no pertenece a un hilo concurrente con estas características se asume que hay elementos sueltos y/o no existe un punto final.
9. Si todas las transiciones no han sido sensibilizadas repetir desde el paso 2 para  $M_2, M_3, \dots, M_k$ . Si todas las transiciones han sido sensibilizadas terminar de iterar hasta que el marcaje vuelva a coincidir con el inicial ( $M_0$ ).

Estos pasos deberán realizarse iterativamente hasta que todas las transiciones de la PN estén sensibilizadas. Si al concluir las iteraciones de los anteriores pasos todos han transcurrido satisfactoriamente se puede afirmar que el diagrama ha sido correctamente modelado. Esta afirmación será independiente de si el DA cumple o no con las especificaciones del proceso que modela, será meramente referente al sentido conceptual del mismo.

### III. RESULTADOS Y DISCUSIÓN

El marco teórico [9] o revisión de la literatura [10; 11] evidenció con claridad la necesidad de adaptar la teoría clásica de las PN a las nuevas necesidades del software. Partiendo de esto se desarrolló un método basado en una adaptación de la definición formal de las mismas. Con la aplicación de este método se podrá evitar errores que comúnmente se cometen en el modelado de los DA. En este sentido se puede argumentar que hipotéticamente el método será capaz de identificar errores tales como:

- No existencia de punto inicial.
- Existencia de más de un punto inicial.
- Utilización innecesaria de bifurcaciones.
- Utilización innecesaria de barras de sincronización.
- Elementos sueltos o sin relacionar con otros.
- No existencia de al menos un punto final.

Para la demostración de la efectividad de la investigación desarrollada y de los resultados arrojados se realizará un experimento como método empírico. Se asumirá en este caso la definición ofrecida por [12], quien plantea que el experimento es la actividad que realiza el investigador donde:

- Aísla el objeto y las propiedades que estudia, de la influencia de otros factores no esenciales que puedan enmascarar la esencia del fenómeno.
- Reproduce el fenómeno objeto de estudio en condiciones controladas.

- Modifica las condiciones bajo las cuales tiene lugar el fenómeno de forma planificada.

El experimento que se desarrollará basará su objetivo en demostrar la hipótesis de que el método para el modelado y prueba de diagramas de actividades mediante redes de Petri es capaz de identificar los errores que se identificaron como comunes en el modelado de estos tipos de diagramas. Para cada uno de ellos se pondrá un ejemplo y se verá como el método es capaz de identificar el problema. Esto se hará paso a paso siguiendo los que se han definido en la presente investigación.

#### A. Identificación de no existencia de punto inicial o de más de un punto inicial.

Si tenemos un diagrama de actividades como se muestra en la siguiente figura:



Figura 3. Diagrama de actividades sin punto de inicio.

Es de esperar que el método es capaz de detectar este error en el paso 1 de los 6 necesarios para modelar un DCU con PN, de ahí que se puede afirmar su efectividad en este aspecto.

#### B. Identificación de utilización innecesaria de bifurcaciones.

Tomando como punto de partida el diagrama de la figura 4, en el cual se hace una utilización innecesaria de una bifurcación, se analizará para verificar si el método es capaz de detectarlo.



Figura 4. Diagrama de actividades con utilización innecesaria de bifurcación.

Si se siguen los pasos para el modelado de DA con PN en el paso 2 y 3 se podrá verificar que se detectará correctamente la utilización innecesaria de bifurcaciones.

#### C. Identificación de utilización innecesaria de barras de sincronización.

A partir del diagrama que se muestra en la figura 5 se puede evidencia claramente cómo responde el método a la utilización innecesaria de barras de sincronización.



Figura 5. Diagrama de actividades con utilización innecesaria de barra de sincronización.

En los pasos 4 y 5 para el modelado de DA mediante PN el método evita la utilización innecesaria de barras de sincronización, ya sean de división o de unión. Se puede asumir con toda certeza que en el diagrama de la figura 5 se podrá identificar rápidamente el mencionado error.

#### D. Identificación de elementos sueltos o sin relacionar con otros.

Suponiendo que tenemos el siguiente diagrama de actividades:

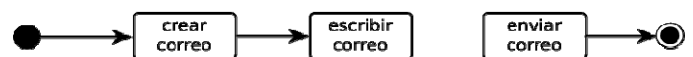


Figura 6. Diagrama de actividades con actividad sin relacionar.

Si seguimos los pasos para el modelado tenemos:

1. Existe un solo punto inicial.
2. No existen bifurcaciones luego de una actividad.
3. No existen bifurcaciones de reunificación.
4. No existen barras de sincronización de tipo división.
5. No existen barras de sincronización de tipo unión.

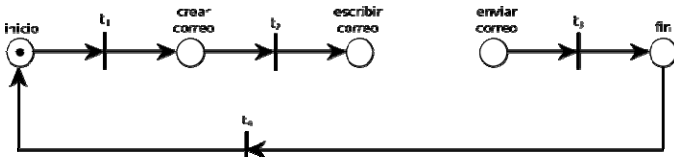


Figura 7. Red de Petri correspondiente al diagrama de la Figura 6.

Luego se deben seguir los pasos para realizar las pruebas, como se muestra a continuación:

1.  $|P| = 5$  dado que existe un punto inicial, tres actividades y un punto final.

$$2. M_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{matrix} \text{inicio} \\ \text{crear correo} \\ \text{escribir correo} \\ \text{enviar correo} \\ \text{fin} \end{matrix}$$

$$3. \text{ Se tiene que } C^+ = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{matrix} \text{inicio} \\ \text{crear correo} \\ \text{escribir correo} \\ \text{enviar correo} \\ \text{fin} \end{matrix} \text{ y}$$

$$C^- = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{matrix} \text{inicio} \\ \text{crear correo} \\ \text{escribir correo} \\ \text{enviar correo} \\ \text{fin} \end{matrix} \text{ por lo que}$$

$$C = \begin{bmatrix} -1 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{matrix} \text{inicio} \\ \text{crear correo} \\ \text{escribir correo} \\ \text{enviar correo} \\ \text{fin} \end{matrix}$$

$$4. \bar{\sigma}_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{matrix} t_1 \\ t_2 \\ t_3 \\ t_4 \end{matrix}$$

$$5. M_1 = M_0 + C \cdot \bar{\sigma}_0, \text{ por lo que } M_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$6. M_0 \neq M_1$$

7. No ha permanecido ningún lugar marcado para ambos vectores.
8. El paso 7 ha arrojado resultados negativos.
9. Solo ha sido sensibilizada  $t_1$  en  $\bar{\sigma}_0$  por lo que se debe volver a iterar desde el paso 2 para hallar  $M_2$ .

Segunda iteración de los pasos para realizar las pruebas:

$$2. M_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$3. C = \begin{bmatrix} -1 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{matrix} \text{inicio} \\ \text{crear correo} \\ \text{escribir correo} \\ \text{enviar correo} \\ \text{fin} \end{matrix}$$

$$4. \bar{\sigma}_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \begin{matrix} t_1 \\ t_2 \\ t_3 \\ t_4 \end{matrix}$$

$$5. M_2 = M_1 + C \cdot \bar{\sigma}_1, \text{ por lo que } M_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$6. M_1 \neq M_2$$

7. No ha permanecido ningún lugar marcado para ambos vectores.
8. El paso 7 ha arrojado resultados negativos.
9. Han sido sensibilizadas  $t_1$  y  $t_2$ , para  $\bar{\sigma}_0$  y  $\bar{\sigma}_1$  respectivamente, quedando  $t_3$  y  $t_4$  sin sensibilizar, por lo que se debe volver a iterar desde el paso 2 para hallar  $M_3$ .

Tercera iteración:

$$2. M_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$3. C = \begin{bmatrix} -1 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{matrix} \text{inicio} \\ \text{crear correo} \\ \text{escribir correo} \\ \text{enviar correo} \\ \text{fin} \end{matrix}$$

$$4. \bar{\sigma}_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{matrix} t_1 \\ t_2 \\ t_3 \\ t_4 \end{matrix}$$

dado que luego del lugar correspondiente a escribir correo no existe una transición, por lo que para este no se sensibilizará ninguna de las 4 existentes.

$$5. M_3 = M_2 + C \cdot \bar{\sigma}_2, \text{ por lo que } M_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$6. M_2 = M_3$$

7. El lugar asociado con escribir correo ha permanecido marcado para ambos vectores.
8. El paso 7 ha arrojado resultados positivos, por lo que se puede asumir que la actividad asociada con escribir correo está suelta y/o no existe al menos un punto final.

De esta forma queda evidenciado que el método es capaz de identificar elementos sueltos o sin relacionar con otros.

**E. Identificación de no existencia de al menos un punto final.**

El método propuesto identifica la ausencia de punto final de forma similar a la existencia de elementos sueltos o sin relacionar con otros.

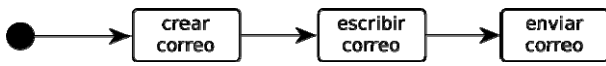


Figura 8. Diagrama de actividades sin punto final.

Si se siguen los pasos para el modelado de diagramas de actividades mediante redes de Petri se obtendrá:

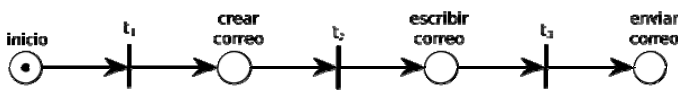


Figura 9. Red de Petri correspondiente al diagrama de la figura 8.

En los pasos para realizar las pruebas e método se comportará de forma similar al ejemplo anterior, si se realizan las iteraciones correspondientes se podrá comprobar que  $M_2 = M_4$ , de lo que se deriva que la actividad correspondiente a enviar correo está suelta y/o que no existe punto final. De esta forma queda identificada la ausencia del punto final en el DA de la figura 8.

**F. Ejemplo sin existencia de posibles errores a identificar.**

En la figura 10 se muestra un DA donde se pueden evidenciar todos los posibles elementos que puedan existir en un diagrama de este tipo. Para este caso en el que todos los elementos están correctamente relacionados y no se ha cometido ningún error conceptual, el método debe arrojar resultados satisfactorios. Anteriormente se ha experimentado con cada uno de los errores que el método es capaz de identificar. Se hace necesario evaluar la respuesta de esta propuesta ante un ejemplo en el cual no se muestran posibles errores a identificar.

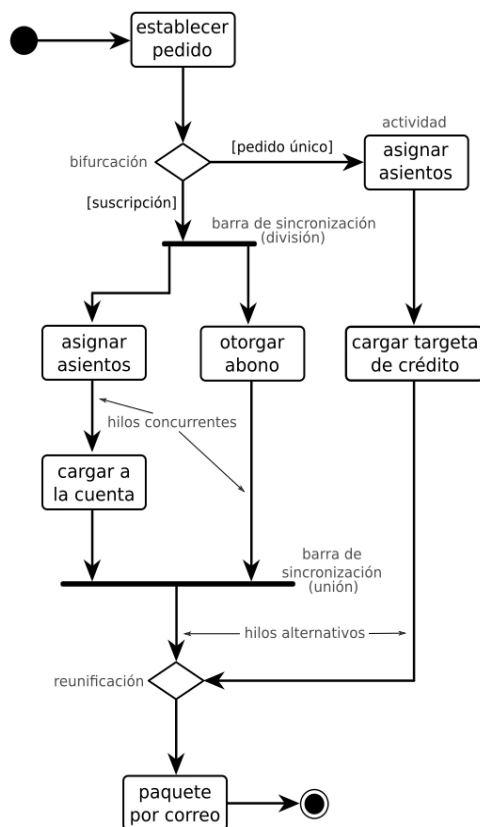


Figura 10. Diagrama de actividades sin existencia de posibles errores a identificar.

Siguiendo los pasos necesarios para modelar un DA mediante una red de Petri se podrá llegar rápidamente a la que se muestra en la figura 11. Verificando que:

1. Existe un punto final.
2. La bifurcación que aparece luego de la actividad establecer pedido ofrece dos vías alternativas.
3. A la bifurcación de reunificación que aparece antes de la actividad paquete por correo entran dos hilos alternativos.
4. De la barra de sincronización de división que aparece salen dos hilos concurrentes.
5. A la barra de sincronización de unión que aparece entran dos hilos concurrentes.
6. Obteniendo la siguiente red de Petri:

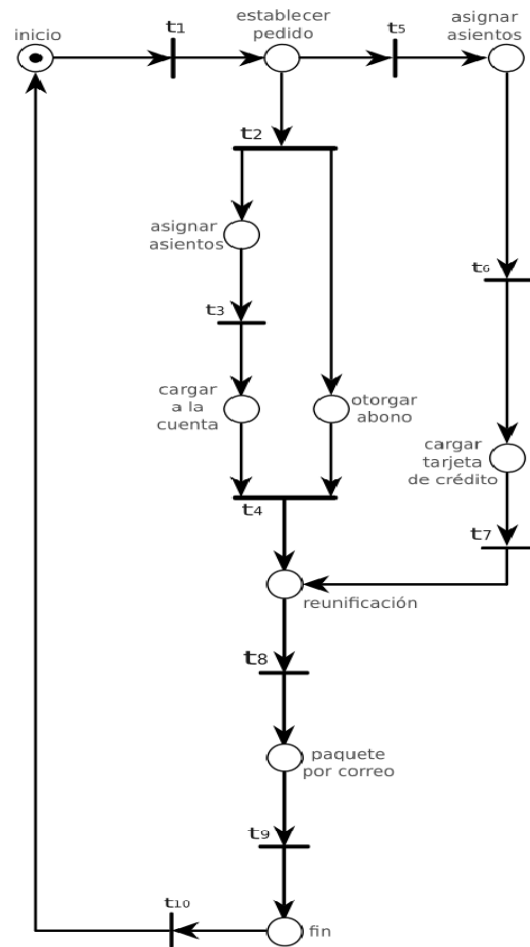


Figura 11. Red de Petri correspondiente al diagrama de la figura 10.

Para este caso si se realizan los pasos para realizar las pruebas pertinentes, se obtendrá:

$$M_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad C^+ = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Por lo que:

$$C^- = \begin{bmatrix} 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 \end{bmatrix}$$

$$C = \begin{bmatrix} -1 & 0 \\ 1 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & -1 \end{bmatrix}$$

Si se sensibilizan las transiciones en el orden  $t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}, t_{11}, t_{12}, t_{13}, t_{14}, t_{15}$  se podrán realizar 14 iteraciones a los pasos necesarios para realizar las pruebas, de forma satisfactoria. Por lo anterior se puede afirmar que este método responde de forma confiable ante diagramas sin posibles errores a identificar.

#### IV. CONCLUSIONES Y TRABAJO FUTURO

Con el desarrollo de la investigación se logró obtener una representación matemática a partir de un diagrama de actividades y con esta realizar pruebas que garantizan la evaluación de su correctitud. Se logró diseñar un método capaz de identificar problemas que se cometen comúnmente en el modelado de estos diagramas. Este método es capaz de identificar la ausencia de punto inicial, así como la existencia de más de uno de estos, la utilización innecesaria de bifurcaciones y de barras de sincronización, la existencia de elementos sueltos o sin relacionar con otros elementos, además de la ausencia de al menos un punto final. El resultado de la presente investigación producirá un impacto profundo en el modelado de software, permitiendo evaluar los diagramas de actividades.

Se plantea como trabajo futuro la automatización del método, así como la posible aplicación de las redes de Petri en el modelado y prueba de otros diagramas en el campo de la Ingeniería de Software.

#### REFERENCIAS

- [1] Petri, C. A. *Kommunikation mit Automaten*. Tesis Doctoral, Institut für Instrumentelle Mathematik, Bonn, 1962.
- [2] Meneses, A. *Redes de Petri*. [En línea] CINESTAV, 2002. [Consultado el: 11 de marzo de 2015]. Disponible en: [http://computacion.cs.cinvestav.mx/~ameneses/pub/tesis/mtesis/node5.html].
- [3] Rosa, F. *Redes de Petri móviles para la especificación y verificación de propiedades de seguridad en sistemas ubicuos*. Tesis Doctoral, Universidad Complutense, Madrid, 2007.
- [4] Baquero, L. R., Argota, L. E., Rodríguez O., Ciudad F. A. *Método para el modelado y prueba de diagramas de casos de uso mediante redes de Petri*. [En línea] Conferencia Científica, 2015. [Consultado el: 02 de octubre de 2015]. Disponible en: [http://conferenciainformatica.uci.cu/?q=node/4505].
- [5] Larman, C. *UML y Patrones*. Madrid, Pearson Educación, 2003. 624p.
- [6] Zapata, G. *Sistema de diagnóstico de fallas para sistemas de transmisión de energía utilizando redes de petri*. *Energética*, 2005, 33: p. 5-14.
- [7] Schuller, J. *Aprendiendo UML en 24 horas*. Prentice Hall, 2001. 448 p.
- [8] Rumbaugh, J., Jacobson I., Booch G. *El Lenguaje Unificado de Modelado, Manual de Referencia*. Addison Wesley, 2000. 526 p.

- [9] Sampieri, R., Collado, C., Lucio, P. *Metodología de la investigación*. Mc Graw Hill, 2006. 850p.
- [10] Creswell, J. *Educational research: Planning, conducting, and evaluating quantitative and qualitative research*. Pearson Education, 2012. 672p.
- [11] Mertens, D. M. *Research and evaluation in education and psychology: integrating diversity with quantitative, qualitative, and mixed methods*. Sage Publications, 2014. 536p.



**Lionel Rodolfo Baquero Hernández** estudiante de 5to año de Ingeniería en Ciencias Informáticas, Facultad 6, Universidad de las Ciencias Informáticas, La Habana, Cuba.



**Ing. Luis Enrique Argota Vega** graduado de Ingeniería en Ciencias Informáticas, profesor del Departamento de Ciencias Básicas de la Facultad 6, Universidad de las Ciencias Informáticas, La Habana, Cuba.



**Ing. Osviel Rodríguez Valdés** graduado de Ingeniería en Ciencias Informáticas, profesor del Departamento de Programación y Sistemas Digitales de la Facultad 6, Universidad de las Ciencias Informáticas, La Habana, Cuba.



**Dr. Febe Ángel Ciudad Ricardo** graduado como Ingeniero Informático en el año 2004 por la Universidad de Holguín. Titulado como Máster en Informática Aplicada en el año 2007 por la Universidad de las Ciencias Informáticas (UCI) y obtuvo el grado científico de Doctor en Ciencias de la Educación en el año 2012 por la Universidad de La Habana (UH). Imparte docencia de pregrado como profesor asistente en las disciplinas de Ingeniería y Gestión de Software, Metodología de la Investigación Científica y Formación Pedagógica. Es miembro de los claustros de las maestrías de Informática Aplicada, Informática Avanzada, Gestión de Proyectos y Educación a Distancia de la UCI. Desarrolla sus investigaciones en las temáticas de Ingeniería y Gestión de Software, con énfasis en el área del Software Educativo; así como en la Tecnología e Informática Educativas. Ha publicado diversos artículos científicos y ha participado en diferentes eventos nacionales e internacionales en estas áreas del conocimiento. Ha sido arquitecto, analista y líder de proyectos de desarrollo de software, jefe de departamento docente y asesor técnico – docente. Se desempeñó como Director del Centro de Innovación y Calidad de la Educación (CICE) de la UCI por varios años. Actualmente se desempeña como Decano de la Facultad Introductoria de Ciencias Informáticas (FICI) de la UCI.