Miércoles 2 de Noviembre, 2015

Documento de Requisitos

Redes de Petri

Integrantes:

- Diego Vergara
- Ramiro Urbina
- Manuel Hernández
- Patricio Sepúlveda

Métodos Formales

Histórico de Revisión:

Fecha	Versión	Descripción	Autor
27/11/2015	1.0	Primer acercamiento a Documento de requerimientos	Diego Vergara Manuel Hernández Ramiro Urbina Patricio Sepúlveda
28/11/2015	1.1	Agregar requerimientos no funcionales	Diego Vergara Manuel Hernández Ramiro Urbina Patricio Sepúlveda
01/11/2015	1.2	Perfeccionamiento de Requisitos	Diego Vergara Manuel Hernández Ramiro Urbina Patricio Sepúlveda

1. Descripción General:

Se requiere construir un programa que permita al Usuario representar el funcionamiento de sistemas mediante Redes de Petri. Es necesario que el Usuario pueda interactuar con el sistema, eso quiere decir que este tendrá la opción de manipular Plazas, Transiciones, Arcos y los Disparos disponibles, con una interfaz gráfica. El usuario podrá dibujar la red de Petri, además de realizar los disparos a elección, permitiendo visualizar la evolución propia de la Red.

1.1. Usuarios: El programa será pensado para ser ejecutado por solo un tipo de usuario. A este usuario, el programa le permitirá dibujar la red de petri, borrar elementos, visualizar las distintas matrices que la componen, y adicionalmente generar los disparos que se encuentren disponibles.

2. Requerimientos Funcionales:

En esta sección se describirán claramente los requisitos funcionales solicitados por el cliente, esperando próximamente satisfacer sus necesidades mediante un producto software que le permita modelar sistemas mediante redes de Petri. Los requerimientos funcionales son aquellos requerimientos que describen la funcionalidad misma del programa o sistema. Los requerimientos son:

Crear elementos:

2.1 – El software debe permitir al usuario crear Plazas, donde el mismo programa asigne el N° de Plaza. La creación de la plaza se realizará mediante una interfaz gráfica, donde el usuario seleccionará el elemento y lo posicionará en un espacio de dibujo determinado.

2.2 – El Programa debe permitir al usuario crear Transiciones, donde el mismo programa asigne el N° de la transición. La creación de la transición se realizará mediante una interfaz gráfica, donde el usuario seleccionará el elemento y lo posicionará en un espacio de dibujo determinado.

2.3 – El programa debe permitir al usuario crear Arcos que permitan la conexión entre Plazas y Transiciones. Este Arco posee un peso que es ingresado por el usuario cuando este lo crea. La construcción de arcos es mediante una interfaz gráfica donde el usuario selecciona el elemento y establece la relación entre la plaza y la transición.

2.4 – El programa debe permitir al usuario ingresar el número de marcas sobre una plaza ya creada.

2.5 – El programa debe permitir al usuario dibujar las redes de Petri con los elementos Plaza, Transición, arcos y número de marcas. Todo este trabajo de dibujo debe ser mediante los elementos de los requerimientos: 2.1, 2.2, 2.3 y 2.4

Eliminar Elementos:

2.6 – El software debe permitir al Usuario eliminar Plazas ya creadas. Una vez que una plaza es eliminada, no es posible obtener sus datos nuevamente. Esta eliminación debe ser utilizando la interfaz gráfica misma.

· Si una Plaza está conectada a una transición, y esta plaza es eliminada, los arcos que están conectadas a ella también son eliminados.

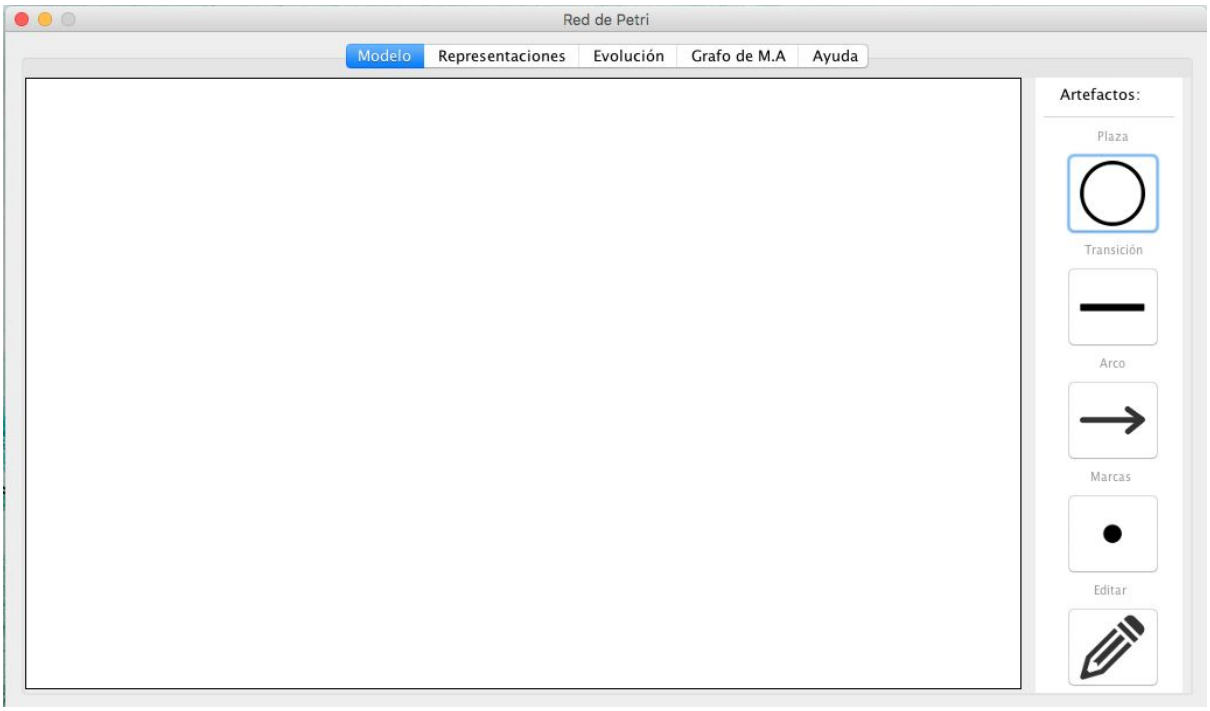
2.7 – El software debe permitir al Usuario eliminar Transiciones ya creadas. Una vez que una transición es eliminada, no es posible obtener sus datos nuevamente. Esta eliminación debe ser utilizando la interfaz gráfica misma.

· Si una Transición está conectada a una Plaza, y esta Transición es eliminada, los arcos que están conectadas a ella también son eliminados.

2.8 – El software debe permitir al Usuario eliminar los Arcos entre Plazas y Transiciones. Una vez que un arco es eliminado no es posible recuperar su información. Esta eliminación debe ser utilizando la interfaz gráfica misma.

2.9 - El software debe permitir al Usuario editar las posiciones de las transiciones, los arcos y las plazas a cualquier lugar del Panel, sin sobreponerse con otro artefacto

Interfaz de Crear, Eliminar y Editar elementos:



Visualización:

2.10 – EL producto software dispondrá para visualización la matriz de incidencia previa (Pre) propia del modelo de red de Petri creado por el usuario. Este especificará las Plazas, transiciones, y el peso de los arcos, propios del modelo. Esta estará disponible para su visualización en todo momento.

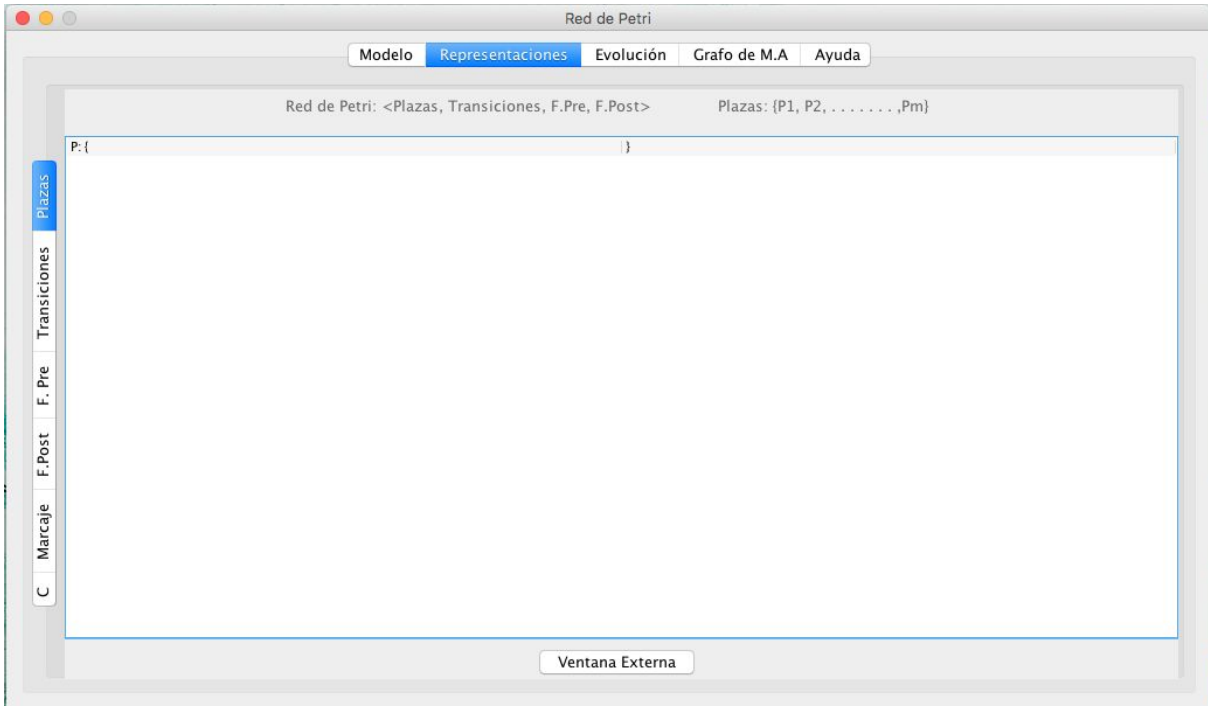
2.11 – El programa tendrá disponible la matriz de incidencia posterior (Post) propia del modelo de red de Petri creado por el usuario. Este especificará las Plazas, Transiciones, y peso de los arcos, propios del modelo. Esta estará disponible para su visualización en todo momento.

2.12 – El programa tendrá disponible la matriz de Marcaje () en cada ejecución del programa, señalando plazas y sus marcajes, permitiendo visualizar la evolución.

2.13 – El programa dispondrá la matriz de incidencia (C) para su visualización en todo momento. Esta matriz de incidencia C se compondrá de la resta de matrices Post y Pre.

2.14 – El programa dispondrá del modelo inicial creado por el usuario en todo momento para ser visualizado, aún después de la ejecución de los disparos.

Interfaz de Visualización:



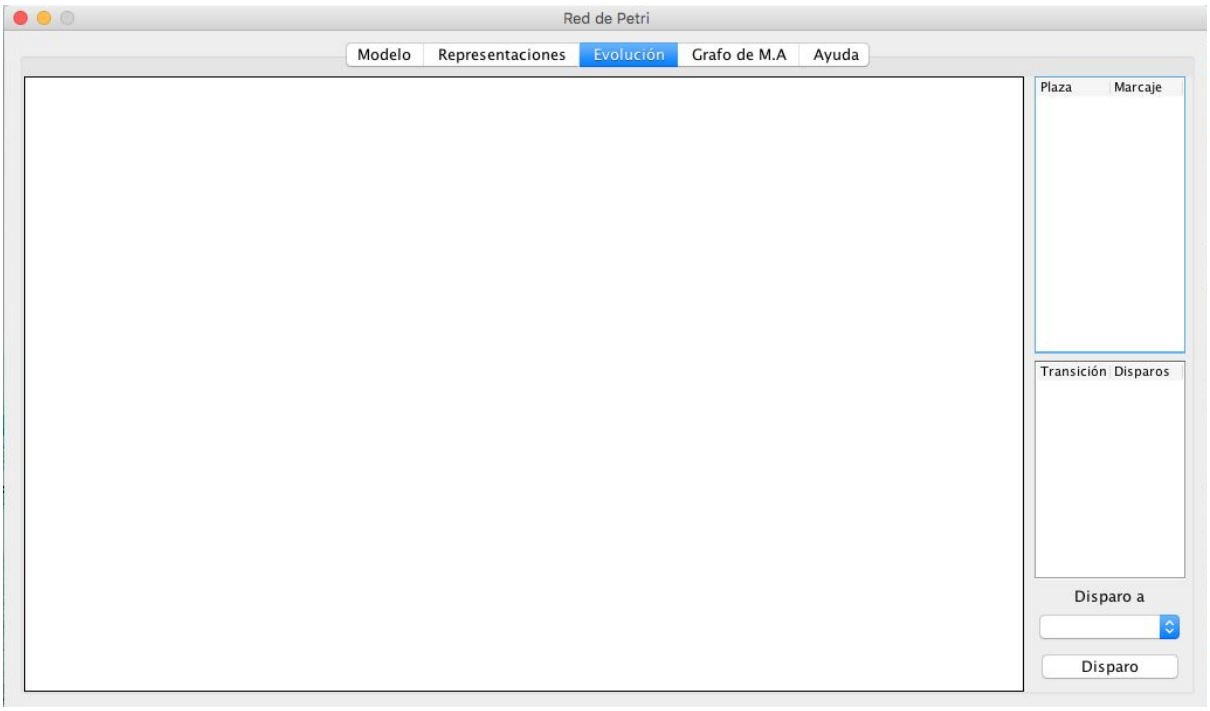
Disparos:

2.15 – El programa debe mostrar las transiciones sensibilizadas.

2.16 – El programa permitirá al Usuario seleccionar el disparo de la transición sensibilizada que el desee, y el software la ejecutará, modificando la red y sus marcas.

2.17 – El programa debe mostrar los cambios de marcaje en el diagrama ya dibujado, permitiendo visualizar la evolución de la red.

Interfaz de Disparos



2.18 – El programa debe mostrar el grafo de marcas accesibles de la red de Petri confeccionada por el usuario. Esta debe especificar las transiciones que son disparadas y los marcajes que se alcanzan con estas transiciones.

Visualizaciones en Ventanas Externas:

2.19 - El programa debe permitir al usuario visualizar la matriz de incidencia previa Pre en una ventana externa.

2.20 - El programa debe permitir al usuario visualizar la matriz de incidencia posterior Post en una ventana externa.

2.21 - El programa debe permitir al usuario visualizar la matriz de incidencia C en una ventana externa.

2.22 - El programa debe permitir al usuario visualizar las transiciones creadas de forma matricial a través de una ventana externa.

2.23 - El programa debe permitir al usuario visualizar las plazas creadas de forma matricial a través de una ventana externa

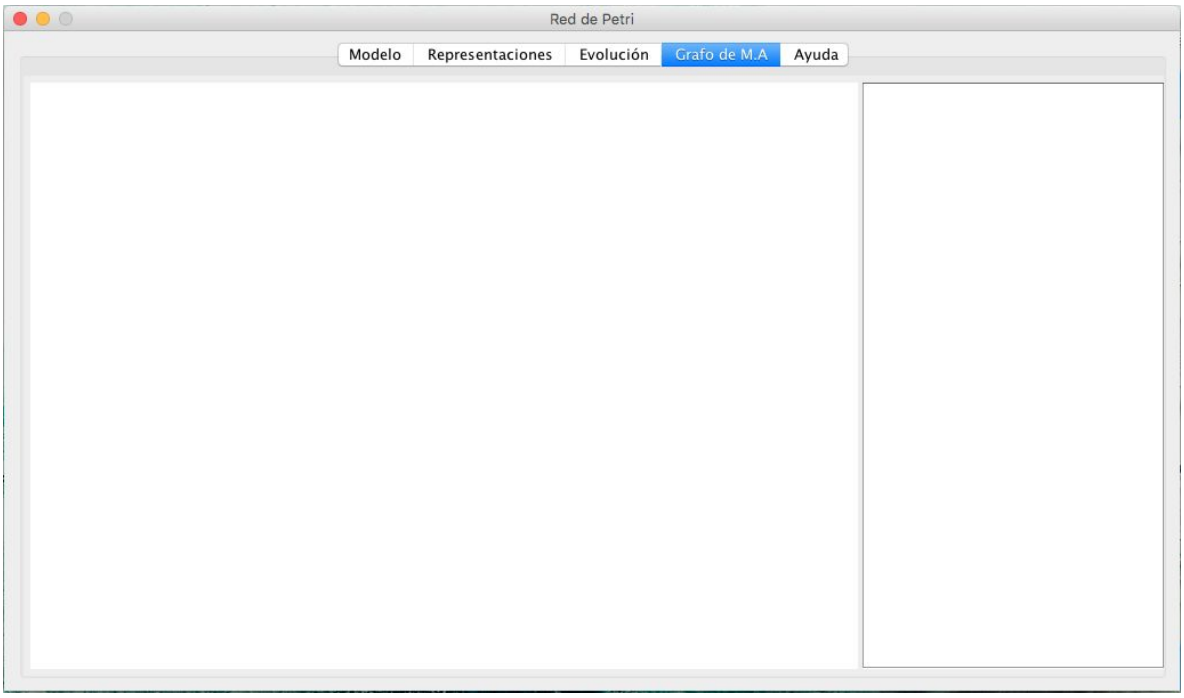
2.24 - El programa debe permitir al usuario visualizar la matriz de marcaje inicial a través de una ventana externa.

2.25 - El programa debe permitir al usuario visualizar la matriz de marcaje de evolución en una ventana externa.

2.26 - El programa debe permitir al usuario visualizar el vector característico de disparos mediante una ventana externa. Este vector contiene las transiciones y la cantidad de veces que se ha disparado.

2.27 - El programa debe permitir al usuario visualizar el vector característico. Este vector contiene las transiciones y la cantidad de veces que se han disparado.

2.28 - El programa debe permitir al usuario visualizar la el grafo de marcas accesibles.

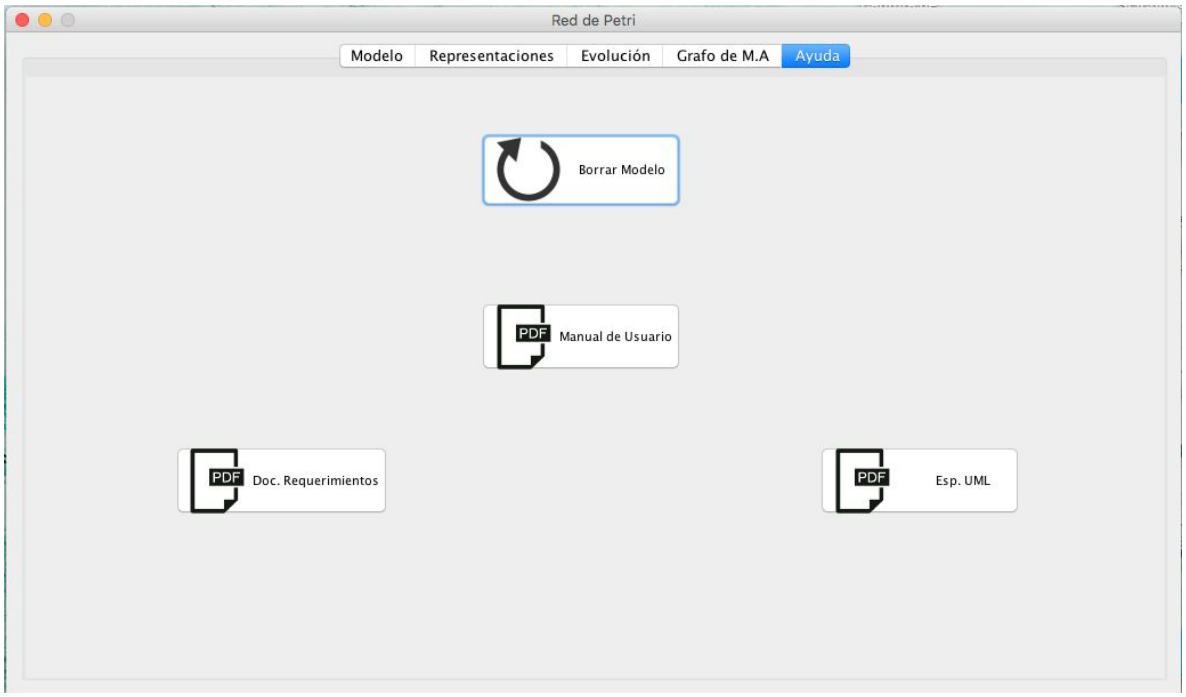


Ayuda

2.29 - El programa debe permitir al usuario acceder al manual de usuario.

2.30 - El programa debe permitir al usuario acceder a la documentación del software; Requisitos, UML.

2.31 - El programa debe permitir al usuario reiniciar el programa.



3. Requerimientos no funcionales:

3.1 – El programa debe ser desarrollado en el lenguaje de Programación JAVA.

3.2 – El programa debe poseer una interfaz gráfica que permita al usuario navegar fácilmente entre los distintos elementos que el software posee.

3.3 - El programa debe funcionar correctamente en los sistemas operativos MAC OSX y Windows (Versiones superiores a windows 7), ya sea en arquitecturas de 32 o de 64 bits.

3.4 - El programa debe seguir las restricciones propias de las redes de Petri, validando los errores que el usuario cometa construyendo su modelo.

3.5 - El programa debe ser fácil de instalar y ejecutar en las plataformas ya mencionadas.