

Modelación de las fallas y tiempo de reparación del fusil FAMAE SG 542-1 aplicando redes de Petri coloreadas jerárquicas

*Modeling failures and repair times of the rifle FAMAE SG 542-1
applying hierarchical coloured Petri nets*

Nicolás Mathias Aguayo Quiroz¹  <https://orcid.org/0000-0001-5707-7728>

Héctor Rodrigo Reyes Campaña^{1*}  <https://orcid.org/0000-0001-6941-8198>

Recibido 09 de septiembre de 2021, aceptado 22 de noviembre de 2022

Received: September 09, 2021 Accepted: November 22, 2022

RESUMEN

Esta publicación presenta la modelación y simulación de un sistema complejo aplicando el método de redes de Petri coloreadas jerárquicas para representar los procesos de falla y reparación, así como la obtención de datos estadísticos que indiquen su desempeño. Se realizó la simulación en el modelo de un sistema mecánico caracterizado por el fusil de guerra FAMAE SG 542-1, sistema de uso habitual en el Ejército de Chile, a través de los softwares CPN Tools y Python representando cómo se desempeñan los componentes que permiten el funcionamiento de esta arma. Este comportamiento es registrado a través de ambos softwares obteniendo los datos de las simulaciones con lo que será posible predecir la condición de un componente y del sistema en el futuro para generar planes de acción correspondientes.

Palabras clave: Modelación, simulación, proceso, redes de Petri, predicción, mantenimiento, falla, mantenibilidad.

ABSTRACT

This paper presents the modelling and simulation of a complex system applying the hierarchical coloured Petri nets method to represent the failure and repair processes, as well as obtaining statistical data indicating its performance. The simulation was performed on the model of a mechanical system characterized by the FAMAE SG 542-1 war rifle, commonly used in the Chilean Army, through the CPN Tools and Python software representing how components perform of operation of this weapon. This behaviour is recorded through both software, obtaining the simulation data so that it will be possible to predict the condition of a component and the system in the future to generate corresponding action plans.

Keywords: Modelling, simulation, process, Petri nets, prediction, maintenance, failure, maintainability.

¹ Universidad de Santiago de Chile. Departamento de Ingeniería Mecánica. Santiago, Chile.

E-mail: nicolas.aguayo.q@usach.cl; hector.reyes.c@usach.cl

* Autor de correspondencia: hector.reyes.c@usach.cl

INTRODUCCIÓN

Realizar experimentos con prototipos reales para obtener información a través del método de prueba y error genera un elevado costo y, además, se necesita una gran cantidad de recursos, como tiempo o personal, para lograr cumplir con el objetivo en estudio. A raíz de esto, los científicos John von Neumann y Stanisław Ulam, durante el desarrollo de la bomba atómica en la Segunda Guerra Mundial, necesitaban comprender la difusividad de los neutrones aleatorios en material de fusión. Debido a esto desarrollaron el método Monte Carlo para simular comportamiento que estas partículas presentaban [1]. De ahí en más, la simulación ha sido ocupada en diversos ámbitos. Se utilizó en la Guerra Fría con fines militares, en la predicción meteorológica e incluso en este último tiempo ha llegado a ámbitos de uso cotidiano y familiar a través de software computacionales [9].

Para entender el concepto de simulación, primero se debe comprender el concepto de modelación, que es la representación de un sistema real de la forma más completa posible mediante diagramas, ecuaciones y/o algoritmos [11], mientras que la simulación es la representación de ese modelo en evolución o cambio con respecto al tiempo, bajo condiciones establecidas, obteniendo una representación del cambio en el sistema a través del modelo [16]. Luego de hacer una significativa cantidad de simulaciones, es posible realizar un proceso de análisis para la mejora continua del modelo y obtener conclusiones relevantes [7].

El proceso de simulación se realiza a través de computadores, y es una herramienta importante en el ámbito de la ingeniería para la obtención de información que asemeje la realidad a partir de datos reales. Es un proceso que, en la mayoría de los casos, es más rápido y con un costo menor que realizar pruebas en sistemas reales obteniendo prácticamente los mismos o mejores resultados, lo que atribuye una gran ventaja en el ámbito económico y en el tiempo involucrado. La desventaja radica, en general, en el tiempo necesario para el entendimiento del sistema, el personal altamente calificado y la gran cantidad de simulaciones que se deben realizar para llegar a soluciones convergentes que permitan detener la simulación, considerando, además, de que los datos de salida sean representativos y aceptables.

Existen distintos tipos de simulaciones que son posibles de generar, como, por ejemplo, continuas, discretas, estáticas, dinámicas, estocásticas, entre otras, las que son utilizadas por distintos métodos, siendo el método Monte Carlo [10] uno de los más relevantes, el que será utilizado para desarrollar este estudio de las fallas y tiempo de reparación de los sistemas complejos.

En el área de mantenimiento se utilizan estas herramientas para realizar análisis de repuestos críticos, vida útil, tiempo de reparación, entre otros. Lo anterior, mediante el uso de los datos reales de un sistema los que han sido registrados en función del tiempo. Ejemplo de lo anterior es un estudio de mantenimiento para el ejército de Australia [20] realizado mediante la simulación en CPN Tools, donde desarrollan un modelo que permite a esta institución simular distintos escenarios para gestionar al personal, activos y localización de estos activos. Es interesante observar los distintos escenarios planteados de manera determinista, sin embargo, al no utilizar distribuciones estadísticas con números aleatorios, los resultados obtenidos solo varían dependiendo de la situación planteada, obviando la aleatoriedad que puede estar presente en la mantención de activos.

En general, la información utilizada para realizar estudios de ingeniería es la registrada en un periodo establecido como en el caso anteriormente descrito, por lo que encontrar los datos como repuestos críticos o tiempos de reparación, requiere un análisis que puede significar la utilización de una significativa cantidad de horas hombre. Esto puede ser modificado a través de simulaciones, que para el ámbito de la ingeniería de mantenimiento son generalmente de carácter estocástico, y estarán enfocadas en el tiempo de funcionamiento hasta la falla y los tiempos de actividades de mantenimiento utilizando distribuciones estadísticas paramétricas, permitiendo de esta manera comprender cómo se comportarán estos parámetros con respecto al tiempo. Existen estudios que han realizado este tipo de análisis, un ejemplo de ello es el caso de estudio de un subsistema de una planta de energía nuclear desarrollado por Electricité de France company [21], siendo el subsistema físico dos turbo bombas de agua. Este subsistema es modelado a través de redes de Petri con el software CPN Tools donde se realizan simulaciones deterministas y simulaciones

estocásticas utilizando el método Monte Carlo obteniendo como resultado la disponibilidad, tiempo medio hasta la primera falla, tiempo medio entre fallas y tiempo medio de reparación para el sistema completo por cada turbina.

En base a estos trabajos, con el objetivo de modelar y simular un proceso real, siendo el caso de este estudio un sistema representado por un fusil, un proyecto original en Sudamérica, el modelo del sistema es realizado a través de redes Petri coloreadas jerárquicas, donde su funcionamiento es simulado a través del software CPN Tools y, además, mediante el software Python, en el que se pretende obtener indicadores y determinar el personal necesario mediante la simulación de un modelo que utilice método determinista y estocástico que hacen posible estudios ingenieriles en el área de mantenimiento sobre el sistema analizado.

Para entender de mejor manera como es realizado el modelo del sistema y su posterior simulación, es necesario definir y comprender los conceptos utilizados para su desarrollo. Esta información es presentada en el siguiente segmento, donde se introduce los conceptos método Monte Carlo, redes de Petri, análisis FMECA entre otros, los cuales son de vital importancia para crear el modelo y a través de la simulación obtener resultados.

FUNDAMENTACIÓN TEÓRICA

Con el fin de realizar el modelo del sistema y realizar simulaciones en base a él, en este apartado son explicados los conceptos esenciales para su desarrollo. Lo primero es conocer el sistema y sus elementos a analizar, por lo que es realizado sobre el mismo un análisis FMECA, como sigue.

Análisis FMECA

El Análisis de los Modos de Falla, Efectos y Criticidad (FMECA) es una técnica mediante la que es posible identificar los modos de falla potenciales para un sistema, proceso, máquina y/o herramientas [15].

El FMECA permite identificar el riesgo que presenta una falla del sistema mediante el cálculo del número de prioridad de riesgo (*RPN*), el que está asociado a las características de las fallas, las cuales son severidad (*S*), posibilidad ocurrencia

(*O*) y probabilidad de detección (*D*), a través de la ecuación (1) presentada a continuación:

$$RPN=S \times O \times D \quad (1)$$

En la ecuación (1), las variables de *S*, *O* y *D* tienen valores de acuerdo con tablas. Los valores indicados en estas tablas son asignados utilizando el criterio de los responsables de realizar el análisis [10].

Luego de identificar los componentes del sistema que presenten mayor índice *RPN*, es realizado el modelo en base a ellos. Estos elementos presentan una cierta cantidad de fallas en un periodo de tiempo debido a su vida útil y al tiempo de reparación, los que poseen una distribución estadística que modela su desempeño. A través del método Monte Carlo, es posible analizar el comportamiento aleatorio de la cantidad de fallas, tiempo de vida útil (en número de tiros disparados) y tiempo de reparación (en minutos).

Método Monte Carlo

El método Monte Carlo es una técnica de simulación que, utilizando métodos matemáticos, permite resolver un problema a través de muestras aleatorias en un modelo del sistema donde el comportamiento de estas simula un proceso real que es determinado por una o más distribuciones estadísticas, por lo tanto, es un proceso caracterizado como estocástico [17].

El método de simulación Monte Carlo fue creado por los científicos John von Neumann y Stanislaw Ulam durante la Segunda Guerra Mundial [9]. La intención original de este método era resolver integrales utilizando números aleatorios a razón de que estas no podían ser resueltas mediante métodos analíticos, sin embargo, luego este método fue aplicado para simular otro tipo de sistemas que empleasen números aleatorios [1].

Este método permite simular el comportamiento de un sistema complejo, para posterior análisis de los datos registrados mediante las simulaciones [10]. Con la obtención de esta información son calculados valores estadísticos con los que es posible predecir el rendimiento del sistema modelado. A raíz de lo anterior, este método es el utilizado para simular la red de Petri coloreada jerárquica (HCPN) en que se modela el sistema.

Desde el punto de vista del mantenimiento, el método Monte Carlo utiliza entre otros aspectos, para los tiempos de falla y tiempos de reparación de los distintos componentes de un sistema o proceso para poder predecir su desempeño en el tiempo [8].

Los softwares utilizados para simular y resolver el modelo desarrollado mediante una HCPN para esta investigación son CPN Tools y Python. Estos programas se basan en el método Monte Carlo utilizando distribuciones estadísticas para calcular el tiempo de falla (cantidad de tiros disparados) y el tiempo de reparación (minutos). Estos datos son registrados después de cada simulación y evaluados cuando convergen a un valor específico para conocer cuando detener la simulación, lo cual se logra realizando una significativa cantidad de simulaciones.

Método de generación de tiempos

Para realizar el análisis del sistema en estudio es necesario generar aleatoriamente tiempos de vida útil hasta la falla y los tiempos de reparación. Para ello son utilizadas distribuciones estadísticas las que son capaces de generar muestras aleatorias de tiempo utilizando los números pseudoaleatorios con los parámetros correspondientes a cada distribución.

Para un mejor y fácil entendimiento de la simulación, las distribuciones utilizadas para generar muestras aleatorias de tiempo es este artículo son la distribución exponencial y la distribución normal. Sin embargo, puede ser utilizada cualquier distribución estadística.

La distribución exponencial es utilizada para generar una muestra aleatoria de tiempo de vida útil (cantidad de tiros disparados) mediante la ecuación (2) [10].

$$t = -\mu \ln(X) \quad (2)$$

Donde X es un número pseudoaleatorio obtenido mediante el uso de una distribución uniforme que pertenece al intervalo $[0,1]$.

La distribución normal es utilizada para generar una muestra aleatoria de tiempo de reparación (en minutos) utilizando la ecuación (3) [10]:

$$t = (X-6)\sigma + \mu \quad (3)$$

Donde X es la sumatoria del Límite Central, generado a partir de la suma de doce números aleatorios obtenidos por medio de una distribución uniforme en el intervalo $[0,1]$, lo cual se presenta en la ecuación (4):

$$X = \sum_{i=1}^{12} X_i \quad (4)$$

Cuando es posible generar aleatoriamente tiempos de vida hasta la falla y tiempos de reparación mediante las distribuciones estadísticas empleadas en la simulación, se debe realizar un análisis y estudio de estas muestras, pero antes, es necesario conocer la cantidad de datos a analizar. Para ello, se emplea una prueba de convergencia que indica cuando detener la simulación, criterio que es explicado en la siguiente sección.

Criterio convergencia

La convergencia es un método utilizado para determinar el momento de detener la simulación y existen diferentes formas de lograrlo. En este trabajo es utilizado para detener la simulación realizada en Python.

Para detener la simulación los criterios utilizados son los siguientes:

1. Simulaciones hasta que cada falla y reparación obtengan 5.000 resultados.
2. La diferencia porcentual entre promedios [19] de las primeras $n \times 100$ simulaciones con los de las $n-1 \times 100$ simulaciones anteriores sea menor al 0,5%, donde n es los periodos de medición con $n \in \{2, 3, 4, 5, \dots, \infty\}$.

El modelo desarrollado será diagramado mediante redes de Petri coloreadas jerárquicas, las que representarán el comportamiento del sistema a través de simulaciones realizadas mediante Python y CPN Tools. En este último, el criterio para detener la simulación es alcanzar una cantidad similar de sistemas simulados utilizando Python.

Redes de Petri clásicas (PN)

La definición que Tadao Murata utiliza para referirse a las redes de Petri (PN) es “una herramienta de modelación gráfica y matemática utilizada para describir y estudiar sistemas caracterizados como

concurrentes, asíncronos, distribuidos, paralelos, no deterministas y/o estocásticos” [5]. Estas redes fueron creadas por Carl Adam Petri en el año 1962 [4] y, desde entonces, muchas variaciones y extensiones han sido desarrolladas, como, por ejemplo, las redes de Petri estocásticas, redes de Petri temporizadas o las redes de Petri coloreadas.

Según lo señalado por Bause y Kritzinger [2] las redes de Petri clásicas u ordinarias son representaciones bipartitas que relacionan los lugares con las transiciones y viceversa. Estas redes están representadas por los componentes descritos a continuación:

- **Lugar/Place:** Un lugar es representado mediante círculos o elipses, los cuales pueden contener tokens. Sirven para indicar una condición o un objeto del sistema modelado.
- **Marca/Token:** Un token es una marca representada generalmente como un punto. Indica el valor específico de la condición o el objeto que se encuentra en un lugar.
- **Transición/Transition:** Es simbolizada generalmente mediante rectángulos lo que representan alguna actividad que cambia el valor del token.
- **Arco/Arc:** Un arco es representado como una flecha. Especifican una conexión entre los lugares y las transiciones.

La representación gráfica de estos componentes que conforman una red de Petri es posible observarlos en la Figura 1 presentada a continuación:

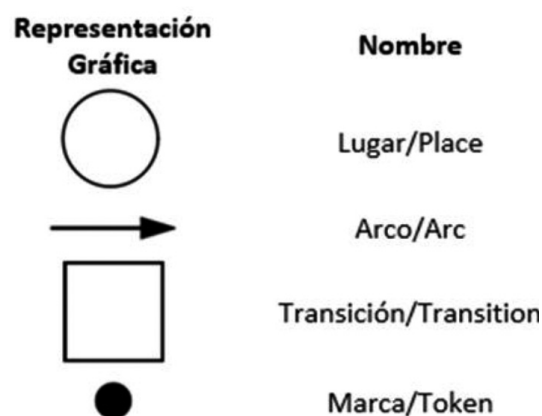


Figura 1. Componentes de una red de Petri.

Redes de Petri coloreadas (CPN)

Las redes de Petri coloreadas (CPN) son una extensión de las redes de Petri (PN), la que combina todas las capacidades de esta última, como lo son las redes de Petri jerárquicas, redes de Petri estocásticas, redes de Petri temporizadas, entre otras, con las capacidades de un lenguaje de programación de alto nivel [3]. La principal diferencia entre las CPN y las PN es que las primeras pueden alojar tokens con diferentes características entre sí en cada lugar, y por lo mismo, en la misma red, como, por ejemplo, los distintos tipos de repuesto de una máquina. Al agregar más de un tipo de token en la red, sumado a las nuevas características que agregadas a las clásica PN, permite realizar redes más robustas y eficientes sobre algún proceso o sistema. Las nuevas características permiten a las CPN, por nombrar algunas, hacer inscripciones en los arcos las cuales condicionan el paso de tokens, tener tiempos de retraso en transiciones, tokens con diferentes características, entre otras.

Como indica Murata [5], “una red (N) es un conjunto con cuatro elementos $N = (P, T, A, W)$ y una red de Petri es un conjunto de cinco elementos $PN = (P, T, A, W, M_0)$ ”. Utilizando estos conceptos, Jensen y Kristensen [3] estipulan que las CPN se pueden definir matemáticamente como el conjunto de nueve elementos principales. Esta definición se aprecia en la ecuación (5):

$$CPN = (P, T, A, W, M_0, \Sigma, C, V, G) \quad (5)$$

En la ecuación (6) es indicado el conjunto P , que es el conjunto de lugares, donde $\{p_1, p_2, \dots, p_m\}$ representan un conjunto finito de lugares y conjuntos no vacíos:

$$P = \{p_1, p_2, \dots, p_m\} \quad (6)$$

La ecuación (7) presenta el conjunto T , el cual es un conjunto de transiciones, donde $\{t_1, t_2, \dots, t_n\}$ representan el conjunto finito de transiciones y conjuntos no vacíos:

$$T = \{t_1, t_2, \dots, t_n\} \quad (7)$$

La ecuación (8) presenta el conjunto de arcos A . Estos representan la dirección del flujo de los tokens desde los lugares hacia transiciones y viceversa:

$$A \subseteq (P \times T) \cup (T \times P) \quad (8)$$

W es una función de arco. Esta función indica la cantidad de tokens que circularán a través cada arco de la red y es representada como indica la ecuación (9):

$$W \in [A \rightarrow \{1, 2, \dots, j\}] \quad (9)$$

M_0 indica los tokens iniciales de la red. Indica la cantidad de tokens en los lugares de entrada. M_0 es presentada en la ecuación (10):

$$M_0 \in [P \rightarrow \{0, 1, \dots, k\}] \quad (10)$$

La ecuación (11) indica que los lugares y transiciones son conjuntos disjuntos, esto significa que sus conjuntos no tienen elementos en común:

$$P \cap T = \emptyset \text{ y } T \cap P = \emptyset \quad (11)$$

Σ representa el conjunto finito de colores y conjuntos no vacíos.

C representa la función de color, la cual asigna un conjunto de colores a cada lugar, lo cual se indica en la ecuación (12):

$$C: P \rightarrow \Sigma \quad (12)$$

V representa un conjunto finito de variables tipo, tal que $Type(v) \in \Sigma$ para todo $v \in V$, es decir, son los diferentes tipos de variables que pueden conformar un token.

G representa una función de guardia, donde a cada conjunto $t \in T$ le es asignada una expresión booleana (Verdadero o Falso).

Redes de Petri coloreadas jerárquicas (HCPN)

Al desarrollar el modelo de un sistema mediante una CPN, esta red puede ser demasiado larga y compleja por lo que es conveniente realizar sub-redes que se conecten entre ellas y la red principal [18]. Estas sub-redes se encuentran dentro de transiciones sustitutas (T_{SUB}). Las T_{SUB} poseen un conjunto de lugares de conexión (P_{sock}) que conectan la sub-red con otras sub-redes o la red principal. Los P_{sock} pueden ser de entrada (P_{sock}^{in}), de salida (P_{sock}^{out}) y de entrada/salida ($P_{sock}^{I/O}$) [3].

De este modo, una HCPN es una metodología de modelación visual donde una T_{SUB} se utiliza para compactar redes complejas y de gran tamaño para una mejor representación del sistema modelado [8]. Según Jensen y Kristensen [3] una HCPN se define matemáticamente por la ecuación (13) presentada a continuación:

$$T_{SUB} \subseteq T \quad (13)$$

Para desarrollar redes de Petri coloreadas de manera gráfica con todas las características mencionadas, es utilizado el software CPN Tools. En la Figura 2 se observa cómo son representados para este trabajo los lugares, tokens, arcos y transiciones mediante este software.

En CPN Tools todos los lugares (a) deben tener un nombre y tener asignado un *color set*, el cual indica que tipo de token puede ser almacenado en dicho lugar. Además, algunos lugares en la red poseen tokens iniciales. De existir un token que se encuentre en un lugar, este es representado mediante un punto que indica la cantidad de elementos.

Las transiciones (b) pueden poseer distintas funciones. En este trabajo solo es utilizada la función de tiempo de retraso “@+” que indica la cantidad de tiempo que tardará la actividad que sea realizada en la transición. Las transiciones sustitutas (d) son transiciones que poseen una sub-red de la red principal.

Los arcos (c) son representado mediante una flecha. Estos deben poseer una inscripción que indica

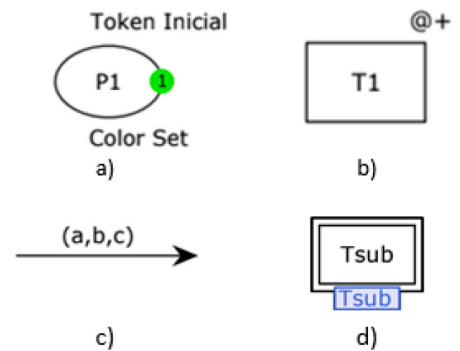


Figura 2. Componentes y funciones utilizadas en la realización de una red de Petri en CPN Tools.

que tipo de token será transmitido a la siguiente transición o lugar. La inscripción puede ser un token, una función o una condición.

METODOLOGÍA

Para realizar el modelo que permitirá analizar el tiempo de vida y el tiempo de reparación de los componentes del sistema, lo primero es conocer cuáles son estos componentes y analizar las fallas potenciales que pueda tener el sistema en estudio.

Mecanismo del sistema en estudio

De acuerdo con la información de uso público del Ejército de Chile y fuentes abiertas, el arma de guerra FAMAE SG 542-1 [6] es un fusil de asalto de calibre 7,62 x 51 mm que actualmente se desarrolla bajo la licencia de FAMAE. Este fusil es utilizado por las tropas regulares dentro del Ejército de Chile y los ciudadanos que, de acuerdo con el DL Nº 2.603, realizan el servicio militar obligatorio. Como cualquier otro fusil de asalto, es un sistema mecánico compuesto de diferentes partes y piezas ensamblables, que juntas hacen funcionar el mecanismo percutor para generar el

disparo del proyectil. Las partes principales que componen cualquier fusil son los presentados en la Tabla 1 [13, 14].

En la Figura 3 es posible apreciar el Fusil de Instrucción SG 542-1 el cual es desarrollado por FAMAE para facilitar el entrenamiento militar en tiempos de paz [12].

Previo a la generación del modelo y posterior simulación en CPN Tools y Python, se debe determinar las fallas potenciales entre las piezas y componentes del mecanismo. Lo anterior puede ser logrado a través de un análisis FMECA realizado en el sistema en estudio. Este análisis es explicado en el segmento realizado a continuación.

Fuente: modificado de www.famae.cl [12].



Figura 3. Fusil SG 542-1.

Tabla 1. Componentes del sistema.

Nº	Nombre del Componente	Descripción
1	Caja de mecanismo	Aloja el percutor que permite disparar la munición. También posee el sistema de seguridad. Al momento de presionar el disparador para percutar la munición, deja salir el percutor el cual golpea la sección posterior del cartucho generando la salida de gases que lo impulsan.
2	Cañón	Una vez percutada la munición, esta es eyectada a través del cañón impulsada por los gases generados. El cañón es un tubo estriado del calibre de la munición, el que produce que la munición, al momento de ser disparada, gire alrededor de su propio eje generando su estabilidad.
3	Vástago	Una vez disparada la munición, el vástago retrocede debido a la presión generada sobre el pistón de los gases, quitando el cartucho utilizado.
4	Culata	Parte posterior del fusil la cual se apoya en el hombro del tirador para disminuir la fuerza producida al percutar una munición.
5	Cargador	Porta la munición del arma y posee un mecanismo interior a base de resortes que permite dejar la siguiente munición a disposición del preparador.
6	Guardamanos	Parte del fusil diseñada y encargada de proteger la mano de apoyo del tirador.
7	Preparador	Tiene la función de cargar la siguiente munición.
8	Pistón de gases	Una vez que los gases ingresan a la cámara en la que se encuentra el pistón, estos generan una presión sobre él, como consecuencia hace retroceder al vástago para que cumpla su función.
9	Cierre	Golpea la parte posterior de la munición para que expulse los gases y el proyectil salga del arma a través del cañón.

FMECA sistema en estudio

Al realizar el análisis FMECA sobre el sistema de estudio, los parámetros S , O y D utilizados en la ecuación (1), reciben valores de acuerdo con los criterios utilizados en Tabla 2, Tabla 3 y Tabla 4 respectivamente. El criterio y la calificación para este análisis fue realizado mediante la metodología de *focus group*, con los integrantes del Departamento de Ingeniería, Investigación y Desarrollo (DI2D), dependiente de la división de Mantenimiento del Ejército de Chile (DIVMAN).

Las tablas para determinar los parámetros S , O y D , fueron formadas en base a 4 criterios de evaluación y una calificación respectiva, la cual depende de la falla.

Identificando la calificación para la falla que podría sufrir cada componente, se calcula el valor del RPN utilizando la ecuación (1), el cual se presenta en la Tabla 5.

Tabla 2. Tabla criterio de severidad.

Severidad	Calificación
Alta	4
Media	3
Baja	2
Nula	1

Tabla 3. Tabla de criterio de posibilidad de ocurrencia.

Ocurrencia	Calificación
Frecuentemente	4
Ocasionalmente	3
Remotamente	2
Nunca o Muy poco probable	1

Tabla 4. Tabla de criterio de probabilidad de detección.

Probabilidad Detección	Calificación
Nula	4
Baja	3
Media	2
Alta	1

De acuerdo con la Tabla 5, los tres elementos que presentan el mayor índice RPN son las fallas que pueden ser producidas en los siguientes componentes: caja de mecanismo, vástago y cañón. La caja de mecanismos tiene fallas en sus piezas debido al desgaste y manipulación del usuario. El vástago falla principalmente porque el resorte cumple su vida útil al realizar cierta cantidad de ciclos, dejando de cumplir su función. Por último, el cañón falla debido a los gases de empuje y al desgaste producido por el roce entre las paredes del tubo del cañón y el proyectil disparado, el cual presenta una velocidad promedio de 820 [m/s] [14].

Al ser estos tres componentes los que presentan un mayor potencial de falla en el sistema, con el propósito de estudiar la distribución de estos, tanto para el tiempo de vida hasta la falla como el tiempo de reparación, es producido el algoritmo que permite entender los procesos de falla y reparación por los que pasa este mecanismo para posteriormente desarrollar el modelo, que en CPN Tools y en Python, será utilizado para simular el desempeño del sistema. Este algoritmo es presentado en la sección “Modelación del sistema”, la cual es descrita a continuación.

Modelación del sistema

El modelamiento del sistema consiste en diseñar una red de Petri coloreada Jerárquica en base a al algoritmo que representa su funcionamiento en la Figura 4, donde el esquema representa el proceso de falla-reparación del sistema en estudio. Al inicio del proceso se presenta un sistema para realizar pruebas en de falla. Si falla, el sistema es remitido

Tabla 5. Cálculo de RPN .

Componente/Parte que puede fallar del sistema	S	O	D	RPN
Caja mecanismo	4	4	4	64
Cañón	4	2	4	32
Vástago	4	4	4	64
Culata	3	2	1	6
Cargador	3	2	3	18
Guardamano	3	1	1	6
Preparador	4	1	3	12
Pistón de gas	4	1	4	16
Cierre	4	1	2	8

a un proceso de reparación dependiendo de los mantenedores disponibles. Al terminar el proceso de reparación, es evaluado si el proceso debe seguir simulando, lo que se realiza de una manera con el software CPN Tools y de otra con Python, maneras que fueron indicadas en la sección criterio de convergencia que fue contextualizado anteriormente.

Modelo del sistema

En base a los procesos indicados en el punto anterior, debido a los diferentes tipos de tokens utilizados en la red, los que pueden ser contenidos en los diferentes lugares, se decide realizar el modelo del sistema a través de una red de Petri coloreada jerárquica temporizada que se indica en la Figura 5. En ella es posible apreciar los lugares y transiciones conectadas en la red.

En la Figura 5 es posible apreciar que el token inicial de la red se encuentra en el lugar “Sig. Fusil”, que

es el encargado de contener el fusil que ingresará al proceso de prueba. Este lugar está conectado mediante arcos a la sub-red “Proceso de Falla”, donde es evaluado si el sistema falla luego de una cantidad de tiros aleatoria. Si no falla, el fusil es enviado al lugar “Fusil Funcionando”, en caso contrario, este es enviado al lugar “Fusil Falla Identificada”. Desde este lugar, los fusiles son enviados a reparación en la sub-red “Proceso de Reparación”. Finalmente, al ser reparado, el fusil llega al lugar “Fusil Reparado”.

La sub-red “Proceso de Falla”, que se muestra en la Figura 5, es posible observarla en detalle en la Figura 6.

La sub-red observada en la Figura 6 indica el proceso para detectar si el fusil sigue funcionando o falla luego de realizar una cantidad aleatoria de disparos. En esta sub-red es contabilizada la cantidad de sistemas que siguen funcionando y que han fallado, estos últimos son registrados mediante la sub-red

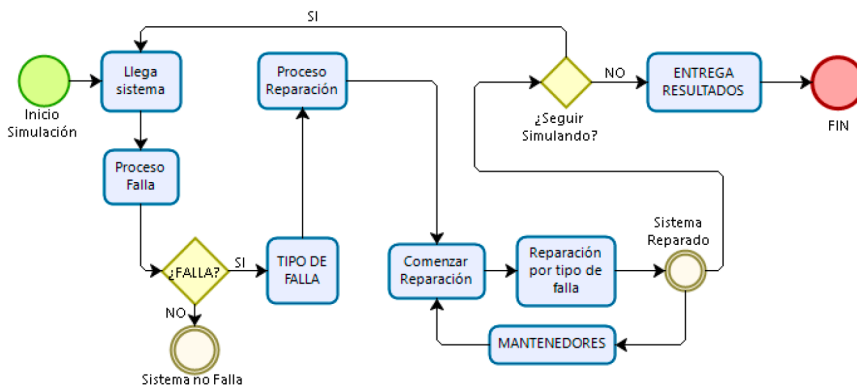


Figura 4. Algoritmo proceso falla-reparación del sistema. Realizado en Bizagi Modeler.

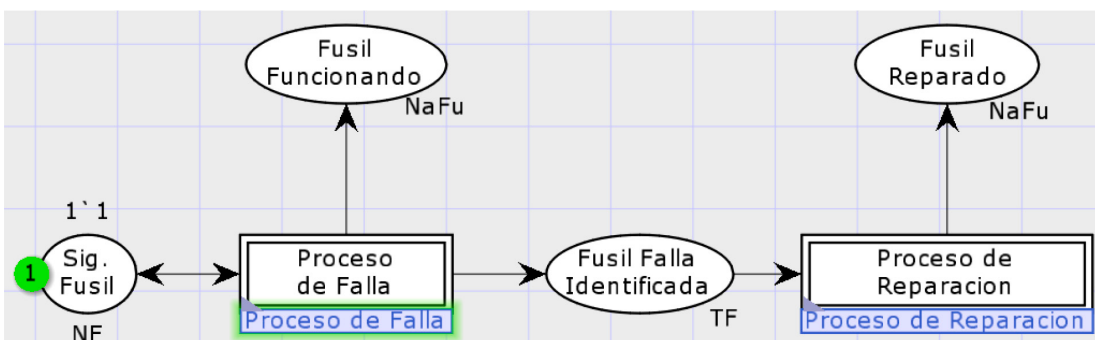


Figura 5. Red principal falla-reparación fusil. Realizado en CPN Tools.

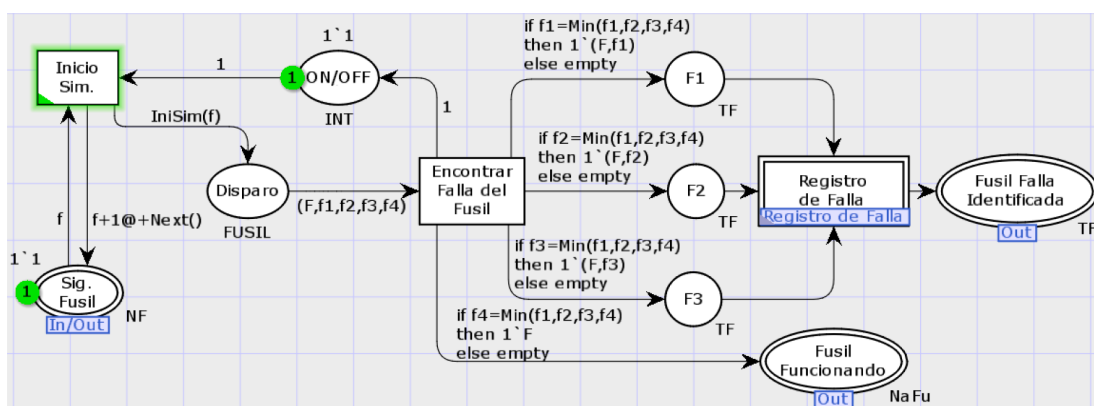


Figura 6. Sub-red “Proceso de Falla”. Realizado en CPN Tools.

interior “Registro de Falla” que se observa en la Figura 7. En ella se señala la cantidad de sistemas que han fallado por tipo de fallas y la cantidad total de proyectiles utilizados de todos los fusiles por tipo de falla. Luego de identificar y registrar la información, los fusiles son enviados al proceso de reparación.

Los sistemas con falla, al salir de la sub-red “Proceso de Falla”, entran a la siguiente sub-red, “Proceso de Reparación”, para ser reparados. Esta sub-red es presentada en detalle en la Figura 8. En ella se indica que cuando llega el fusil, un mantenedor disponible se dispone a realizar el trabajo de reparación. También se registra el momento en el tiempo en que el sistema es reparado, lo cual permite realizar un análisis de las colas de espera por cantidad de mantenedores mediante este software.

Dentro de la sub-red de “Proceso de Reparación”, se encuentra la sub-red “Registro de Reparacion” observada en la Figura 9. En ella se contabiliza la cantidad de sistemas reparados por tipo de falla y se registra la cantidad de tiempo total de reparación de todos los sistemas por tipo de falla.

Las transiciones, lugares, tokens y funciones utilizadas para desarrollar el modelo mediante una HCPN son indicadas en el apartado Anexo Tablas.

Este modelo, exceptuando la sección para realizar muestreo de mantenedores y lista de espera, ha sido desarrollado también a través del software Python. Mediante la simulación del modelo en este programa, es posible realizar el muestreo aleatorio de las fallas, tiempo de falla (en tiros disparados) y el tiempo de reparación (minutos).

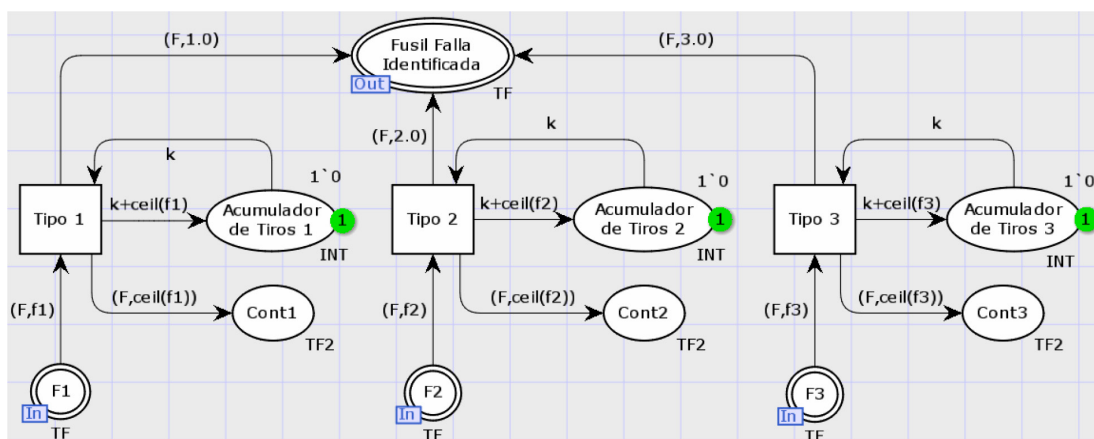


Figura 7. Sub-red “Registro de Falla”. Realizado en CPN Tools.

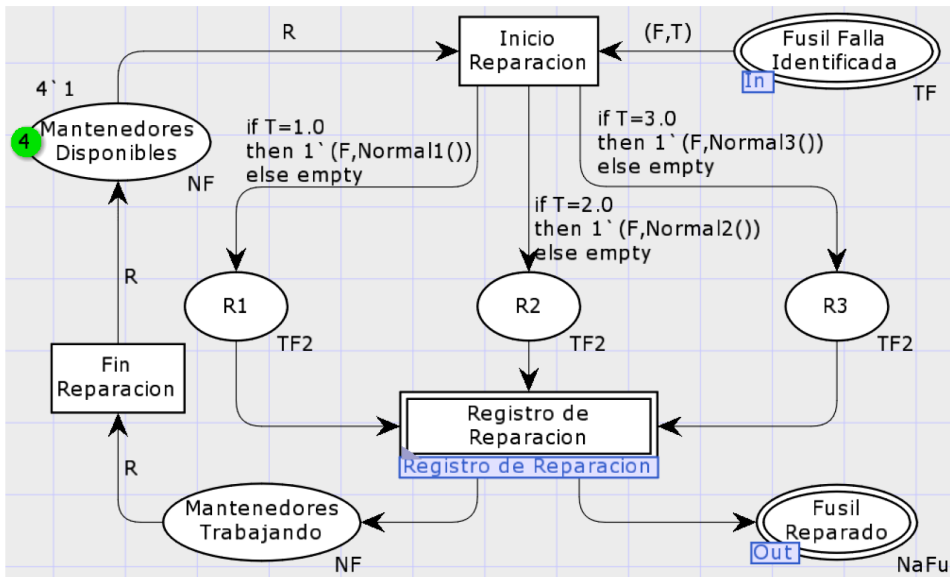


Figura 8. Sub-red “Proceso de Reparación”. Realizado en CPN Tools.

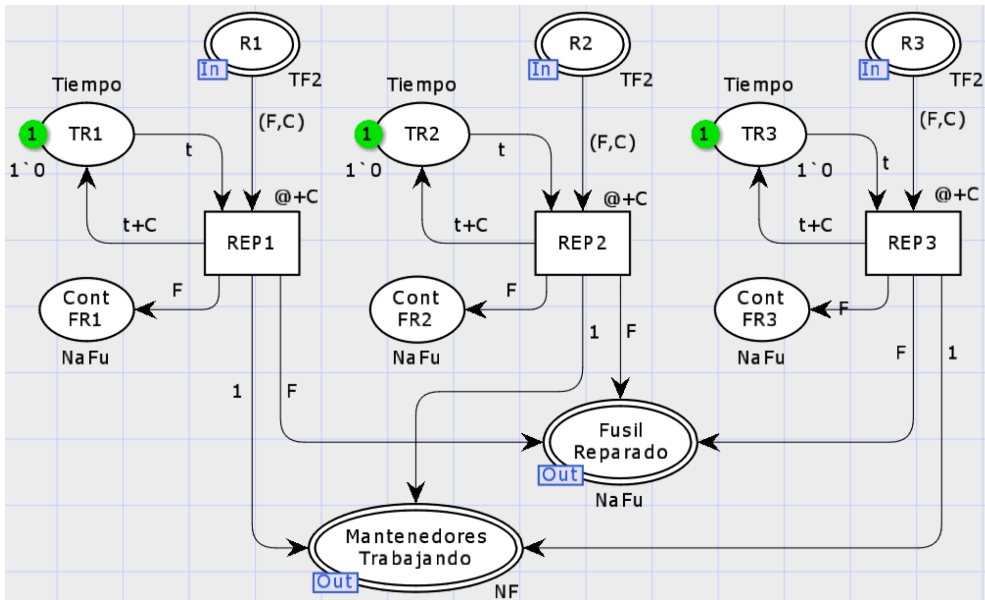


Figura 9. Sub-red “Registro de Reparación”. Realizado en CPN Tools.

Las distribuciones utilizadas para realizar muestreos aleatorios al simular el modelo desarrollado, donde se indican la cantidad de fallas y el tiempo de reparación, fueron la distribución exponencial y la distribución normal respectivamente.

La razón de utilizar la distribución exponencial para indicar la cantidad de fallas es porque esta

distribución tiene un excelente rendimiento para representar el tiempo de vida útil hasta la falla de un elemento o componente, en este caso particular, el tiempo de vida en cantidad de disparos. Utilizando esta distribución, el vástago tiene una media de vida de 500 proyectiles, la caja de mecanismos 200 proyectiles y el cañón 600 proyectiles.

Por otra parte, se utilizó la distribución normal para realizar el muestreo aleatorio de tiempo generado. Esto debido a la facilidad en su uso y porque, como su nombre lo dice, genera una distribución normal alrededor de la media de tiempo de reparación indicado como parámetro. El vástago tendrá un tiempo medio de reparación de 100 [min] ($1\frac{2}{3}$ [h]). La caja de mecanismos tendrá un tiempo medio de reparación de 165 [min] ($2\frac{3}{4}$ [h]) y el del cañón será 200 [min] ($3\frac{1}{3}$ [h]). Todos tendrán una desviación estándar de 45 [min] ($\frac{3}{4}$ [h]).

Cabe destacar que los sistemas utilizados para realizar este estudio entrarán al proceso de prueba en un intervalo entre 30-60 [min] de manera aleatoria. Lo anterior, es para poder establecer de la óptima cantidad de mantenedores necesarios en el proceso.

A objeto de generar un real sustento a la simulación realizada, todos los valores utilizados fueron estimados, ajustados y ponderados para esta publicación académica, a objeto de resguardar cualquier dato de carácter industrial o de defensa que pueda ser potencialmente clasificado.

Mediante los modelos desarrollados, entendiendo como funcionan los procesos de falla y reparación, es posible, a través la información generada luego de una significativa cantidad de simulaciones en CPN Tools y realizando simulaciones hasta que los datos puedan lograr la convergencia en Python, obtener resultados que son posibles de observar en el siguiente apartado.

RESULTADOS

En los siguientes segmentos son presentados los resultados obtenidos mediante la simulación en CPN Tools y Python.

Cantidad de fallas

Al realizar 105.000 simulaciones en la HCPN modelada a través del software CPN Tools y simulaciones hasta la convergencia en Python, se obtiene, con respecto a los sistemas que presentaron fallas, una cantidad de sistemas ensayados similar en ambos programas computacionales. Esta información indicada en la Tabla 6.

En la Tabla 6 es posible observar que, utilizando ambos softwares, la falla tipo 2, es decir falla la caja de mecanismos, es la que posee un mayor porcentaje de ocurrencia. Le sigue la falla tipo 1 (falla vástago), luego la falla tipo 3 (falla cañón) y, por último, los sistemas que no presentan falla.

En la misma tabla es posible verificar que los porcentajes de falla, para cada tipo de falla simulando en ambos softwares, tienen un valor bastante similar. Estos indican que el componente con mayor índice de falla la caja de mecanismo, concentrado más del 50% de las fallas. El vástago concentra aproximadamente el 20% de las fallas, el cañón alrededor de 17% del total de sistemas simulados. Al ser un sistema en serie, cualquier tipo de falla causará que el sistema no pueda funcionar. Concentrar los esfuerzos en las fallas producidas por la caja de mecanismo seria la prioridad pues es la causante de que más del 50% de este tipo de sistemas no pueda funcionar continuamente por mayor tiempo.

Cantidad de tiros disparados hasta la falla

En el punto anterior se observó la cantidad de fallas y el porcentaje de estas, pero estas no indican en qué momento ocurren.

Tabla 6. Cantidad de fallas 105.000 simulaciones en CPN Tools y hasta la convergencia en Python.

	CPN Tools		Python	
	Cantidad	Porcentaje	Cantidad	Porcentaje
Sistemas sin Falla	3.265	11,32%	3.309	11,52%
Sistemas con Falla Tipo 1	5.947	20,62%	5.720	19,91%
Sistemas con Falla Tipo 2	14.725	51,05%	14.702	51,17%
Sistemas con Falla Tipo 3	4.907	17,01%	5.000	17,40%
Sistemas simulados	28.844	100%	28.731	100%

A través de los lugares contadores y acumuladores presentes en la sub-red “Proceso de Falla” de la red principal modelada en CPN Tools, es posible observar la cantidad de sistemas con fallas por tipo de falla y la suma de la cantidad total de tiros disparados que cada sistema realizó hasta la fallar. Esta información se puede apreciar en la Tabla 7, donde se han aproximado los valores utilizando el criterio del primer decimal.

En Python, para encontrar la cantidad de disparos hasta la falla se han realizado simulaciones hasta que se cumpla el criterio de convergencia, obteniendo los resultados indicados en la Tabla 8, que es presentada a continuación.

Es posible estudiar, mediante la Figura 10, la evolución datos de la simulación en Python hasta llegar a los resultados convergentes presentados en la Tabla 8. En esta figura es posible apreciar que el sistema falla por la cantidad de tiros disparados, además, se observa que las fallas producidas por estos tres componentes convergen a una cantidad similar de tiros.

Al analizar la Tabla 7 y Tabla 8, se observa que, utilizando ambos softwares, el sistema falla en promedio, cercano a los 102 proyectiles disparados, situación que es coherente por las medias de falla de la distribución utilizada (distribución exponencial con $\mu = 500, 200$ y 600 tiros para Falla tipo 1, 2 y

Tabla 7. Promedio de disparos hasta fallar por tipo de falla.

Falla	Significado	Disparos Totales [u]	Cantidad de Fusiles Fallados [u]	Promedio Disparos hasta la Falla [u]
Tipo 1	Falla vástago	611.670	5.947	103
Tipo 2	Falla caja mecanismo	1.512.881	14.725	103
Tipo 3	Falla cañón	500.732	4.907	102

Tabla 8. Convergencia tiros hasta la falla.

Falla	Caja mecanismo	Vástago	Cañón
Convergencia [u]	103,77	99,88	105,06
Aproximado [u]	104	100	105

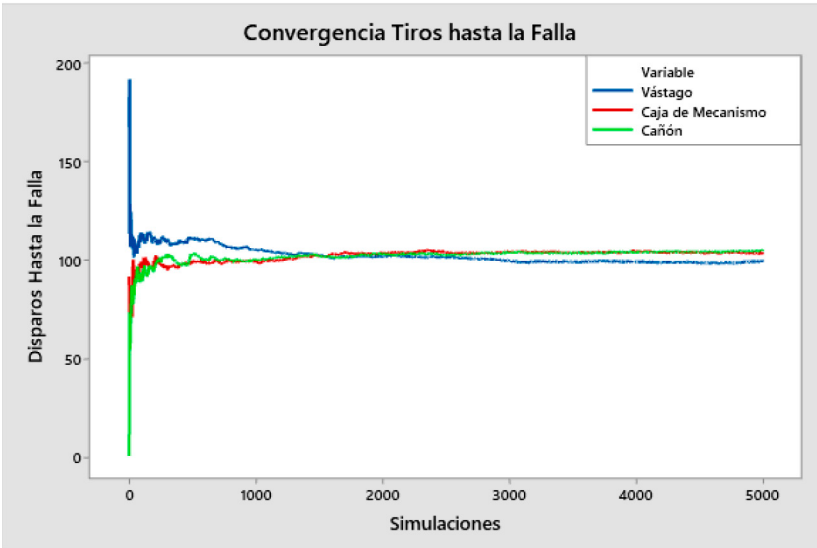


Figura 10. Gráfico convergencia tiros hasta la falla. Graficado en Minitab.

3 respectivamente) y, además, porque las fallas en el sistema presentan un carácter estocástico.

Tiempo de reparación por falla

Mediante el software CPN Tools, en la sub-red “Proceso de Reparación”, utilizando los acumuladores y contadores de fusiles reparados fue posible obtener el tiempo de reparación por falla en minutos, los que son transformados a horas. Lo anterior es presentado en la Tabla 9, donde los resultados fueron obtenidos en minutos y transformados a horas.

Utilizando la programación de un código en el lenguaje Python, es posible calcular el tiempo

medio por reparación por tipo de falla a través de la convergencia. Esta información es presentada en la Tabla 10, donde los tiempos convergentes de reparación, según el criterio de convergencia, fueron obtenidos en minutos y transformados a horas.

El gráfico de convergencia de tiempo de reparación para cada tipo de falla es presentado en la Figura 11.

Nuevamente, los resultados obtenidos mediante la simulación en ambos softwares son similares, lo que se puede evidenciar al comparar la Tabla 9 y Tabla 10.

Los resultados que entregan estas tablas indican que el cañón es el componente que más tarda en

Tabla 9. Tiempo promedio de reparación por tipo de falla mediante CPN Tools.

Falla	Tiempo total de Reparación [h]	Cantidad de Fusiles Reparados [u]	Promedio Tiempo de Reparación por Falla [h]
Tipo 1	9.924,78	5.947	1,67
Tipo 2	40.504,32	14.724	2,75
Tipo 3	16.353,92	4.907	3,33

Tabla 10. Convergencia tiempo reparación.

Falla	Vástago	Caja mecanismo	Cañón
Tiempo [h]	1,66	2,75	3,34

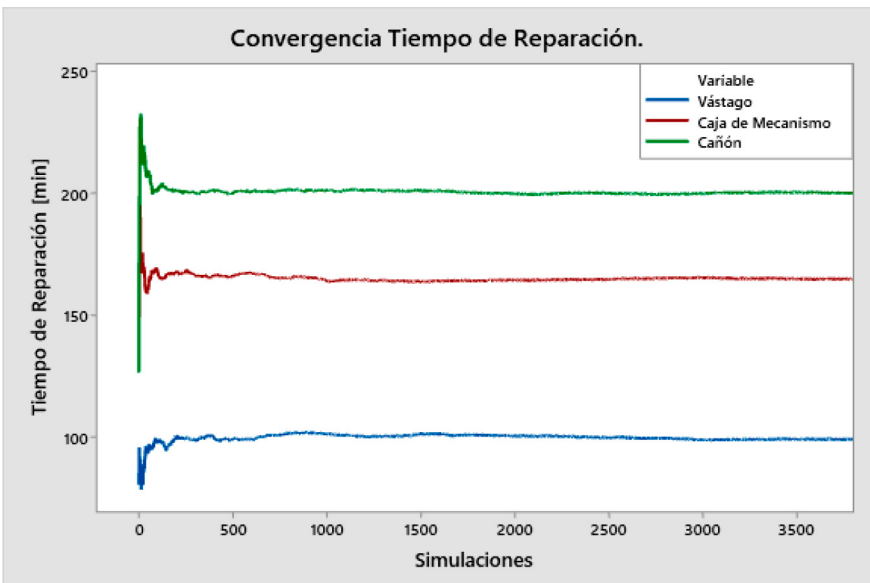


Figura 11. Gráfico convergencia tiempo de reparación. Graficado en Minitab.

ser reparado, seguido por la caja de mecanismos y el vástago.

Como se expresó anteriormente, la cantidad de fallas producidas por el mal funcionamiento del vástago es de alrededor del 20% del total de sistemas simulados, mientras que las fallas por cañón son un 17%. Ambas cantidades son similares, pero al momento de analizar los tiempos de reparación, el del cañón casi el doble que el del vástago. La caja de mecanismo, por su parte, tiene un tiempo medio de reparación de $2\frac{3}{4}$ [h], un tiempo que se acerca al de reparación del cañón, pero concentra más del 50% de las fallas totales, por lo cual el tiempo total es significativamente mayor.

Convergencia

Como ya se fue mencionado, en la Figura 10, se aprecia el gráfico Convergencia tiros hasta la falla. En él es posible observar que la media de tiros comienza a converger luego de las 3.000 simulaciones. Antes de eso, en especial antes de las 1.500 simulaciones, la media de tiros hasta la falla de los tres elementos analizados del sistema en estudio tiene un valor con una fluctuación importante.

Realizando un análisis similar en la Figura 11, es posible observar que los tiempos medios de reparación comienzan a converger alrededor de las 1.500 simulaciones para la caja de mecanismo, cerca de las 2.500 simulaciones para el cañón y después de las 3.000 simulaciones para el vástago. Antes de estas simulaciones y en especial antes de

las 1.500 simulaciones para todos los componentes, se observa que los tiempos medios de reparación de cada componente del sistema fluctúan de manera considerable con los tiempos medios adyacentes.

Equipo de mantenedores

Para evitar que sistemas que han fallado se encuentren mayor tiempo fuera de servicio por no existir mantenedores disponibles para repararlos, el modelo realizado en CPN Tools es simulado en diversos escenarios con un número distinto de mantenedores cada vez. Lo anterior, para encontrar la cantidad óptima que permitan disminuir el tiempo de espera para que los fusiles sean mantenidos y evitar un exceso de mantenedores.

Para realizar lo anterior se utiliza la función de monitoreo “Mark Size” y la de “Simulation performance report” de CPN Tools en los lugares “Fusil Falla Identificada” y “Mantenedores”. Esto permite obtener los datos estadísticos al simular en este software. Son realizadas 5000 simulaciones en diez oportunidades para 1, 2, 3, 4, 5, 6 y 7 mantenedores. Al graficar los datos entregados por los reportes del software se observa el promedio de mantenedores disponibles y el promedio de los sistemas en espera. El gráfico señalado es el presentado en la Figura 12.

La Figura 12a indica, en cantidades promedio obtenidas mediante la herramienta “Simulation performance report”, los mantenedores trabajando, disponibles y sistemas en cola de espera que fueron

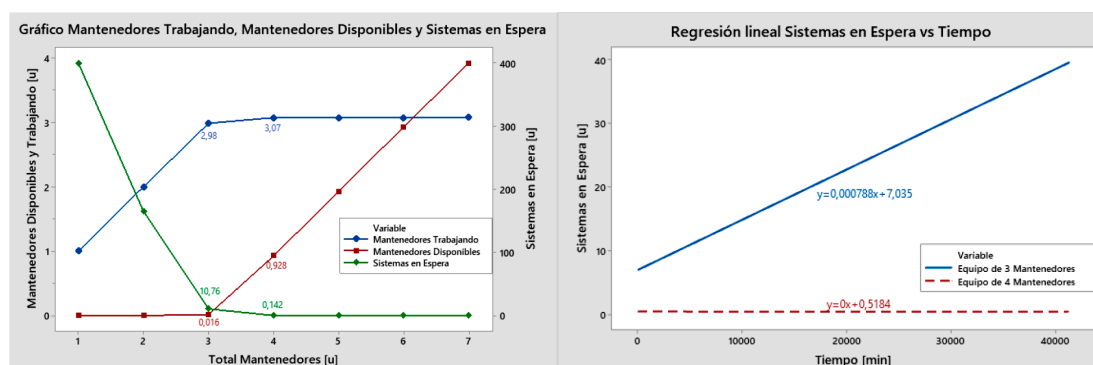


Figura 12. a) Gráfico mantenedores trabajando, mantenedores disponibles y sistemas en espera dependiendo del total de mantenedores. b) Gráfico regresión sistemas en espera vs tiempo para equipo de 3 y 4 mantenedores. Graficado en Minitab.

registrados al realizar las simulaciones en CPN Tools. Es posible observar que el óptimo, con el objetivo de mantener menos sistemas en espera y considerando las condiciones indicadas en el sistema, sería un equipo de tres o cuatro mantenedores. Con tres mantenedores, luego de 5.000 simulaciones en diez oportunidades, se obtuvo en promedio de 10,76 sistemas en espera y prácticamente los trabajadores siempre están trabajando. Con cuatro mantenedores ocurre algo diferente, los sistemas en cola de espera prácticamente desaparecen y en promedio hay menos de 1 mantenedor disponible. La cantidad de mantenedores trabajando es 3,07 y se mantiene así incluso con un mayor número de mantenedores.

La Figura 12b indica mediante una regresión lineal, utilizando los datos entregados por la herramienta "Marking Size", cómo evoluciona la cantidad de elementos en espera con un equipo de 3 y 4 mantenedores en función del tiempo.

Solo considerando la cantidad de sistemas en espera, un equipo de cuatro mantenedores sería lo ideal debido a que la cola de espera generada sería prácticamente nula. En el caso del equipo de tres mantenedores, los sistemas en espera irían al alza a medida que transcurra el tiempo llegando a 40 unidades en aproximadamente 1 mes (43.200 [min]). Esto indica que se acumula un sistema en la cola de espera cada 1270 minutos aproximadamente (21 [h]).

Analizando la cantidad de mantenedores de manera económica, considerando un costo estimado por concepto de remuneraciones de 1,5 UF diarias y trabajando 30 días al mes, utilizando un equipo de tres y cuatro mantenedores, los recursos monetarios utilizados mensualmente serían 135 UF y 180 UF respectivamente. Claramente, entre el equipo de tres y cuatro mantenedores, es más económica la primera opción. Esta opción, aun siendo la más económica, generaría colas de espera en el transcurso de un mes, lo que podría aumentar los costos más allá que pagar a cuatro mantenedores.

Dicho lo anterior, la opción de escoger un equipo de tres o cuatro mantenedores depende exclusivamente de las necesidades de la empresa. Si quiere disminuir los costos por concepto de remuneración, escoger un equipo de tres mantenedores es la opción. Por otro lado, si la necesidad es mantener los sistemas el

menor tiempo posible fuera de servicio, un equipo de cuatro mantenedores es la alternativa a elegir.

Una opción viable, sería tener un equipo de cuatro mantenedores, pero que el cuarto que está disponible según indica la Figura 12a) pueda realizar funciones administrativas y de esta forma alternar entre tres y cuatro mantenedores activos según la situación lo amerite.

CONCLUSIONES

En consecuencia, la finalidad de este estudio fue generar la modelación y simulación de un sistema complejo, caracterizado por el fusil de guerra FAMAE SG 542-1, a través de la aplicación de las redes de Petri, en particular, las HCPN. Dado que, con el experimento estadístico, fue posible establecer la cantidad de fallas, el tiempo de vida útil hasta la falla, el tiempo de reparación de estas y la cantidad óptima de mantenedores, estos fundamentales indicadores pueden ser utilizados por la industria militar, en el campo de la Ingeniería aplicada.

El desarrollo de este estudio a través de programas computacionales, que utilizan simulaciones estocásticas y discretas, permitió comprender y predecir el desempeño de componentes de este sistema bajo las condiciones reales de funcionamiento, en un tiempo relativamente menor que el que se hubiese empleado con prototipos reales. Las condiciones generadas en este modelo pueden ser modificadas y adecuadas a otro sistema de características equivalentes, permitiendo que este proceso para determinar fallas y tiempos de reparación sea replicable a otros sistemas similares, diferentes, de mayor o menor complejidad que pueda utilizar la industria de la defensa para implementar estudios de ingeniería.

A través de la simulación numérica del modelo realizado mediante una red de Petri jerárquica, se pudo demostrar que empleando el método de simulación Monte Carlo para obtener aleatoriedad en la ocurrencia de sucesos, que los resultados obtenidos, relacionados con el porcentaje de fallas, vida útil en tiros del sistema y tiempos de reparación, utilizando los softwares Python y CPN Tools son equivalente y no presentan una diferencia significativa que sugiera elegir uno en desmedro del otro. Los criterios que permiten escoger uno sobre el otro

son las características que estos puedan ofrecer, como la posibilidad de convergencia en Python, o la simulación dinámica en CPN Tools.

Estos estudios de Ingeniería, basados en procedimiento de simulación, permiten observar no solo el comportamiento del sistema, sino también el proceso de reparación y cómo actúan los mantenedores. A través de esto es posible encontrar cuales son las condiciones críticas del sistema, la cantidad de fallas, el tiempo de reparación, acumular una cola de espera, entre otros. Conocerlas hace posible enfrentarlas y desarrollar un plan de acción que permita manejarlas de la mejor manera, disminuyendo las pérdidas generadas por la disminución de la disponibilidad.

Por último, en la ingeniería de mantenimiento, el conservar los sistemas de trabajo para que puedan cumplir su función es una tarea fundamental. Hoy en día, contar con herramientas de simulación, y personal capacitado que pueda realizar modelos y simulaciones de sistemas en estudio, que permitan predecir el desempeño de sus componentes es algo necesario, puesto que, generando estas predicciones a través de la información de salida, es posible ahorrar una significativa cantidad de recursos y eventualmente, aumentar las utilidades.

Propuestas

Finalmente, después de realizar el respectivo análisis e interpretación de los datos de salida obtenidos, se propone generar las siguientes acciones de mejora continua:

- Contar con un stock de repuestos de los componentes, conforme a prioridades de acuerdo con mayor frecuencia de falla y que más tiempo mantienen al sistema fuera de servicio.
- Generar y mantener modelos mediante el uso de redes de Petri y sus extensiones de todos los sistemas, para que en evento de estos sufrir fallas, poder comprender rápidamente que ocurrió y tener planes de acción futuros con el objetivo que puedan volver a funcionar lo más pronto posible.
- Estudiar los sistemas a través de estas redes puede indicar si es rentable que el sistema sea reparado o, de otro modo, comprar un conjunto que presente una menor tasa de falla.
- Mantener un equipo de profesionales en la simulación en las organizaciones responsables

del mantenimiento para resolver y analizar este tipo de problemas es algo que todas las compañías debiesen considerar para disminuir costos y tiempos de ensayo en prototipos reales.

AGRADECIMIENTOS

Los autores de este artículo académico agradecen a la División de Mantenimiento del Ejército de Chile (DIVMAN) y al Departamento de Ingeniería, Investigación y Desarrollo (DI2D) por las facilidades y apoyo brindado en la realización de esta investigación relacionada con Ingeniería de mantenimiento.

REFERENCIAS

- [1] O. Morales Ponce. “Simulación de Monte Carlo en autómatas celulares”. Fecha de consulta: 10 de abril de 2021. URL: <http://delta.cs.cinvestav.mx/~mcintosh/oldweb/s1998/oscar/resumen.html>
- [2] F. Bause and P.S. Kritzinger. “Stochastic Petri Nets: An introduction to the theory”. Vieweg+Teubner Verlag, Segunda Edición, pp. 77-80. Lengerich, Alemania. ISBN: 3-528-15535-3. 2002.
- [3] K. Jensen and L.M. Kristensen “Coloured Petri Nets: Modelling and validation of concurrent systems”. Springer. Berlin, Germany. ISBN: 978-3-642-00283-0. 2009.
- [4] C.A. Petri. “Communication with Automata”. Rome Air Development Center, Research and Technology Division. Report RADCTR-65-377. Vol. 1. New York, USA. 1966.
- [5] T. Murata. “Petri Nets: Properties, analysis and applications”. Proceedings of the IEEE. Vol. 77 N° 4, pp. 541-580. 1989. ISSN: 0018-9219. DOI: 10.1109/5.24143.
- [6] FAMA. “Fusil FAMA SG 542-1M calibre 7,62x51mm”. Fecha de acceso: 15 de enero de 2020. URL: http://www.famae.cl/pdf/fichas/armamento-menor/Fusil_Famae_SG_542_1M_Calibre_7-62x51mm.pdf
- [7] J.A. Sokolowski and C.M. Banks. “Principles of modelling and simulation: A multidisciplinary approach”. John Wiley & Sons. New Jersey, USA. ISBN: 978-0-470-28943-3. 2009.
- [8] H. Reyes. “Generalised fleet maintenance modelling”. PhD Thesis. University of Nottingham. Nottingham, England. 2018.

- [9] Facultat d'Informàtica de Barcelona. "Simulación". Fecha de consulta: 01 de febrero de 2020. URL: <https://www.fib.upc.edu/retro-informatica/avui/simulacio.html>
- [10] J.D. Andrew and T.R. Moss. "Reliability and risk assessment". Professional Engineering Publishing. Chippenham, Wiltshire, UK. ISBN: 1-86058-290-7. 2002.
- [11] J. Wadsworth. "Análisis de sistema de producción animal - tomo 2: Las herramientas básicas". FAO, Roma. Vol 2. Roma, Italia. ISBN: 92-5-304089-0. 1997.
- [12] FAMAE. "Fusil de instrucción SG 542-1". Fecha de acceso: 15 de enero de 2020. URL: http://www.famae.cl/pdf/fichas/armamento-menor/Fusil_de_Instrucciones_SG_542_1.pdf
- [13] Ó. Pastor Escalona and A. Sepúlveda Osses. "Diseño preliminar de un cajón de los mecanismos monobloque de aleación ferrosa para el fusil SG 542-1 FAMAE". Boletín Científico Tecnológico. Academia Politécnica Militar. N° 23. Santiago, Chile, 2019. ISSN: 0718-1191.
- [14] FAMAE. "Manual de operación y mantenimiento fusiles de asalto FAMAE". Fecha de acceso: 20 de marzo de 2021. URL: https://www.erelogistics.com/eresystems/ereelitemanuals/FAMAE%20SG-540-542-543%20Series%20Rifles%20Espa%C3%B1ol_Ingl%C3%A9s%20Manual.pdf
- [15] J.F. Ríos Puerta and J. Garcés Gómez. "Desarrollo de un plan de mantenimiento preventivo basado en el análisis Fmea y monitoreo mediante el software AM para un sistema Gemi Cmd Bombas". Universidad EAFIT. Bachelor Thesis. Medellín, Colombia, 2006.
- [16] A.M. Law and W.D. Kelton "Simulation modeling & analysis". McGraw-Hill. Second edition. New York, USA. ISBN: 0-07-100803-9. 1991.
- [17] J.I. Illana. "Métodos Monte Carlo". Universidad de Granada, Departamento de Física Teórica y del Cosmos. 2013. URL: <https://www.ugr.es/~jillana/Docencia/FM/mc.pdf>
- [18] S. Christensen and L.O. Jepsen. "Modelling and simulation of a network management system using hierarchical coloured Petri Nets". Proceedings of the 1991 European simulation multiconference. 1991. ISSN: 0105-8517.
- [19] S. Ross. "Simulation". Academic Press. Fifth edition. Amsterdam, Netherlands. ISBN: 978-0-12-415825-2. 2013.
- [20] G.E. Gallasch, B. Francis and J. Billington. "Seeking improved CPN tools simulator performance: evaluation of modeling strategies for an army maintenance process". Tenth Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools. 2009. ISSN: 0105-8517.
- [21] B. Pinna, G. Babykina, N. Brinzei and J. Pétin. "Deterministic and stochastic dependability analysis of industrial systems using Coloured Petri Nets approach". Annual Conference of the European Safety and Reliability Association, pp. 2969-2977. 2013.

ANEXO TABLAS

Tabla 11. Transiciones, lugares y color set de la red principal.

Nombre	Tipo	Descripción
Sig. Fusil	Lugar	Mantiene el nuevo fusil que ingresará a prueba.
Proceso de Falla	Sub-red	Sub procesos que indica si el fusil sigue funcionando o falla.
Fusil Funcionando	Lugar	Acumulador de fusiles que no fallaron.
Fusil Falla Identificada	Lugar	Conecta dos sub-redes. Lugar por el que pasan lo fusiles que fallaron en el proceso anterior para entrar en el proceso de reparación.
Proceso de Reparacion	Sub-red	Proceso en el que se rapara el fusil por los mantenedores.
Fusil Reparado	Lugar	Lugar que acumula el total de sistemas reparados.
NF	Color Set	Enteros temporizados.
Nafu	Color Set	String.
TF	Color Set	String*Real temporizado.

Tabla 12. Transiciones, lugares, color set y funciones de la sub-red “Proceso de Falla”.

Nombre	Tipo	Descripción
Sig. Fusil	Lugar	Mantiene el nuevo fusil que ingresará a prueba.
ON/OFF	Lugar	Lugar que permite que solo un fusil este en proceso de prueba.
Inicio Sim.	Transición	Le da un nombre el fusil que llega desde “ <i>Next Rifle</i> ” e indica la cantidad de disparos a la que falla el arma a través de la función “ <i>IniSim(F)</i> ”.
Disparo	Lugar	Aloja al token que representa al fusil y la cantidad de disparos hasta la falla.
Encontrar Falla del Fusil	Transición	Determina si le fusil falla o no. Si falla, determina qué tipo de falla.
Fusil Funcionando	Lugar	Acumula los fusiles que han sobrevivido al proceso de prueba.
F1, F2 y F3	Lugares	Lugares por al que llegan los fusiles que tienen falla 1, 2 y 3 respectivamente.
Registro de Falla	Transición	Subproceso para registrar cantidad de fallas por tipo y tiempo de falla.
Fusil Falla Identificada	Lugar	Lugar que indica el fusil y el tipo de falla que tiene. Final del proceso. Conecta con la siguiente sub-red.
NF	Color Set	Enteros Temporizados.
FUSIL	Color Set	Permite los tokens del tipo String*Real*Real*Real*Real temporizado, es decir, los tokens $F, f1, f2, f3, f4$.
NaFu	Color Set	String.
TF	Color Set	String*Real temporizado.
IniSim(f)	Función	Crea un token de la forma $F, f1, f2, f3, f4$, donde: $F = "F"^{NF.mkstr(f)}$, $f1 = exponential(1.0/500.0)$ $f2 = exponential(1.0/200.0)$, $f3 = exponential(1.0/600.0)$ $f4 = exponential(1.0/900.0)$.
Min(f1, f2, f3, f4)	Función	Entrega el valor mínimo entre los parámetros f1, f2, f3 y f4 mediante: $Min(f1, f2, f3, f4) = Real.min(Real.min(f1, f2), Real.min(f3, f4))$.
Next()	Función	Indica que el siguiente sistema llegará entre en un intervalo de [30,60] minutos después.
f	Token	Indica el fusil y el tiempo en que llega el siguiente fusil.
F	Token	Indica, en formato de string, que fusil esta está realizando la simulación de la forma “FX”, donde “X” indica el número del fusil.
$(F, f1, f2, f3, f4)$	Token	f1, f2, f3, f4 indican la cantidad de disparos a los que fallaría el fusil según tipo de falla.
$(F, f1), (F, f2), (F, f3)$	Token	Indican el fusil y la cantidad de disparos dependiendo de la falla.

Tabla 13. Transiciones, lugares, color set y funciones de la sub-red “Registro de Falla”.

Nombre	Tipo	Descripción
Tipo 1, Tipo 2 y Tipo 3	Transiciones	Transiciones que conectan con tres lugares. Los contadores de sistemas fallados por tipo de falla, lo acumuladores de disparos totales por tipo de falla y el lugar “ <i>Rifle Failure Identified</i> ”.
Cont1, Cont2 y Cont3	Lugares	Registan la cantidad de fusil que han fallado por tipo de falla.
Acumulador de Tiros 1, 2 y 3	Lugares	Indican la cantidad de disparos total que han efectuado todas las armas que registradas en los lugares “ <i>Count1</i> ”, “ <i>Count2</i> ” y “ <i>Count3</i> ” respectivamente.
INT	Color Set	Enteros.
TF2	Color Set	String*Entero temporizado.
TF	Color Set	String*Real temporizado.
Ceil (x)	Función	Aproxima un número real hacia el entero superior.
k	Token	Indica la cantidad total de disparos a las que han fallado los fusiles que han sido simulados.
k + ceil (x)	Token	Suma la cantidad de disparos a las que ha fallado el fusil que está siendo simulado.
(F, ceil (x))	Token	Indica el fusil y la cantidad de disparos a las que este ha fallado.
(F, 1.0), (F, 2.0), (F, 3.0)	Token	Indica el fusil y el tipo de falla que posee.

Tabla 14. Transiciones, lugares, color set y funciones de la sub-red “Proceso de Reparacion”.

Nombre	Tipo	Descripción
Fusil Falla Identificada	Lugar	Lugar que indica el fusil y el tipo de falla que tiene. Final del proceso. Conecta con la siguiente sub-red.
Mantenedores Disponibles	Lugar	Indica la cantidad de Mantenedores disponibles
Inicio Reparacion	Transición	Transición donde llega el mantenedor junto al fusil a reparar. De aquí sale el fusil con el tiempo de reparación por el tipo de falla según una distribución normal utilizando la ecuación (3).
R1, R2 y R3	Lugares	Indican el fusil, el tiempo de reparación y el momento en el tiempo en que se encuentra.
Registro de Reparacion	Sub-red.	Sub proceso para registrar cantidad de sistemas reparados por tipo de falla y el tiempo de reparación total por tipo de falla.
Fusil Reparado	Lugar	Indica la cantidad total de Fusiles Reparados.
Mantenedores Trabajando	Lugar	Indica que y cuantos Mantenedores se encuentran trabajando en la reparación de algún sistema.
Fin Reparación	Transición	Cuando el mantenedor en “ <i>Maintainers Working</i> ” termina su trabajo con el sistema pasa por la transición para volver a “ <i>Maintainers Availables</i> ”.
Nafu	Color Set	String.
TF	Color Set	String*Real temporizado.
TF2	Color Set	String*Entero temporizado.
NF	Color Set	Enteros Temporizados.
Normal 1()	Función	Devuelve una muestra aleatoria de tiempo de distribución normal con parámetros $(\mu, \sigma) = (100.0, 45.0)$.
Normal 2()	Función	Devuelve una muestra aleatoria de tiempo de distribución normal con parámetros $(\mu, \sigma) = (165.0, 45.0)$.
Normal 3()	Función	Devuelve una muestra aleatoria de tiempo de distribución normal con parámetros $(\mu, \sigma) = (200.0, 45.0)$.
(F, T)	Token	Indica el fusil y el tipo de falla que posee.
(F, Normal 1()), (F, Normal 2 ()), (F, Normal 3 ())	Token	Indica el fusil y el tiempo de reparación por tipo de falla.
R	Token	Indica mantenedores.

Tabla 15. Transiciones, lugares, color set y funciones de la sub-red “Registro de Reparacion”.

Nombre	Tipo	Descripción
R1, R2 y R3	Lugares	Indican el fusil, el tiempo de reparación y el momento en el tiempo en que se encuentra.
REP1, REP2 y REP3	Transición	Transmiten información de R1, R2 y R3 a los lugares ContFR1, ContFR2, ContFR3, TR1, TR2, TR3 y Mantenedores Trabajando según corresponda.
ContFR1, ContFR2 y ContFR3	Lugares	Cuentan la cantidad de fusiles reparados por tipo de falla.
TR1, TR2 y TR3	Lugares	Acumulan el tiempo total de reparación de los por tipo de falla.
Mantenedores Trabajando	Lugar	Indica que y cuantos mantenedores se encuentran trabajando en la reparación de algún sistema.
Fusil Reparado	Lugar	Indica la cantidad total de fusiles reparados, independiente del tipo de falla.
Tiempo	Color Set	Entero temporizado.
(F, C)	Token	Indica el fusil y el tiempo C en que será reparado por el mantenedor.
@ + C	Tiempo de Retraso	Añade el tiempo que le tomará al mantenedor para reparar el fusil y estar disponible nuevamente.
t	Token	Indica el tiempo total que ha tomado reparar todos los fusiles de algún tipo de falla en particular.
t + C	Token	Suma el tiempo C que ha tomado repara el fusil que está siendo simulado.