Capítulo IV

Implementación del detector de malware

4.1 Introducción.

El malware, como se describió en el capítulo uno, es cualquier software dañino capaz de afectar a un equipo de diferentes formas. De acuerdo con la clasificación del malware descrita en el capítulo uno, los distintos tipos de malware pueden trabajar de forma individual o en conjunto para conseguir su objetivo.

A pesar de que existen herramientas de seguridad informática muy robustas basadas en la detección de malware por firmas digitales, algoritmos de comportamiento anormal, etc., los perpetradores explotan vulnerabilidades presentes en S.O. y crean ataques de día cero capaces de burlar la detección empleada por estas herramientas de seguridad. Este problema motivó a implementar una técnica de detección de malware diferente a las técnicas descritas. La técnica propuesta se basa en la detección de malware por patrones típicos de comportamiento referentes a capas del modelo OSI.

En este capítulo se explican las diversas técnicas utilizadas para la detección del malware. Además de explicar el desarrollo y funcionamiento del plugin, “escaneo”, creado para la detección de malware.

4.2 Estrategia de protección

Los conceptos de seguridad informática descritos en la sección 1.5 del capítulo 1 tienen el objetivo de proporcionar una visión general de lo que es la seguridad informática, los criterios usados para la clasificación de ataques y las buenas prácticas utilizadas, con el fin de poder desarrollar el detector de malware basado en el monitoreo de red eficazmente.

4.2.1 Medidas de protección.

Todo sistema de seguridad debe de contar con diferentes mecanismos de protección, como pueden ser antivirus, antispam, firewalls, IDS, IPS, entre otros. Entre mayores herramientas de seguridad se tengan implementadas se garantiza tener una seguridad más robusta. Sin embargo, ningún sistema es 100% infalible, por este motivo radican las buenas prácticas de los usuarios, el tener diferentes niveles de jerarquía en el acceso a la información, privilegios de acuerdo con el cargo realizado, herramientas de seguridad activadas las 24 horas del día, entre otras acciones.

Para proteger los activos o bienes es de gran utilidad plantearse las siguientes preguntas:

- ¿Qué se quiere proteger?
- ¿De qué se quiere proteger?
- ¿Cómo se quiere proteger?

El realizar un análisis con ayuda de estas preguntas garantiza una mejor aplicación de los mecanismos de seguridad en los bienes y activos a proteger. Estas tres preguntas fueron de gran utilidad en la creación del detector de malware:

- ¿Qué se quiere proteger? Se desea proteger la red central de la institución en donde esta implementado este proyecto de tesis, en especial se quiere proteger los activos y bienes de la institución que puedan generar pérdidas potenciales o una grave alteración en el funcionamiento de la red.

- ¿De qué se quiere proteger? De cualquier evento que presente un comportamiento anormal en la red de la institución. Especialmente de malware que deja evidencia en la red y sea observado mediante el monitoreo de red. Ya sea de malware ejecutado por usuarios internos o usuarios externos
- ¿Cómo se quiere proteger? Adicionalmente a las herramientas de seguridad (antivirus, antispam, firewalls, IDS, IPS, políticas de seguridad, entre otros) que ya se encuentran instaladas en la institución, se pretende integrar un nuevo mecanismo de seguridad basado en la detección de malware que genera comportamientos anormales en la red de la institución.

  Por medio del software Nfsen se creó un plugin que se ejecuta cada cinco minutos en búsqueda de comportamientos anormales descritos en la sección 4.3. Al detectar un comportamiento anormal se notifica inmediatamente vía email al administrador de red.

4.3 Comportamiento del malware.

Como se describió en el capítulo uno, el malware tiene una extensa clasificación y no es posible definir un comportamiento general sobre él. Sin embargo, por medio de investigaciones realizadas del comportamiento del malware se observa que generalmente presentan técnicas de infección y propagación comunes independientemente del tipo de malware. Todas estas técnicas fueron resultado de una investigación realizada acerca del comportamiento del malware en páginas orientadas a seguridad, como son:

- ✓ CERT
- ✓ SANS
- ✓ Insecure.org, entre otras.

Además de lectura de libros orientados a seguridad y búsqueda de información en internet.

A continuación se describirán las técnicas más comunes utilizadas por el malware para tratar de infectar a sus víctimas.

1) Tratar de pasar desapercibidos.

Uno de los principales objetivos de cualquier malware es pasar desapercibido. Debido a que si el usuario o la herramienta de seguridad no se dan cuenta de su presencia, el malware cumplirá con su objetivo sin ningún problema.

Un ejemplo típico de este comportamiento son los keylogger. Generalmente este malware es instalado por el usuario sin darse cuenta o por algún otro malware que ha abierto una *backdoor* en el sistema; el *keylogger* se encargará de recolectar la información tecleada por el usuario y enviarla a su creador. El usuario no se da cuenta de esta acción debido a que el ataque muestra un comportamiento pasivo y no causa daños al sistema. Sin embargo, este malware estará cumpliendo con su objetivo de recolectar y enviar información.

2) Tratar de infectar a otros equipos.

El malware que ha logrado infectar a un equipo, tratará de explotar la vulnerabilidad detectada en el equipo "víctima" sobre otros equipos pertenecientes a la misma red en el menor tiempo posible. Típicamente el malware que actúa bajo este comportamiento genera miles de conexiones en un tiempo pequeño, tratando de infectar a la mayor cantidad de equipos en el menor tiempo posible. Los métodos de infección más comunes son los siguientes.

Escaneo de puertos.

El malware que trate de infectar a diferentes equipos mediante un escaneo de puertos, presenta un comportamiento descrito de la siguiente forma:

- El malware que ha logrado infectar a una computadora con dirección IP *"xxx.xxx.xxx.xxx"*, buscara infectar a otra computadora perteneciente a la misma red con dirección IP *"yyy.yyy.yyy.yyy",* explotando la misma vulnerabilidad que encontró en el equipo víctima.
- Generalmente el malware que emplea esta técnica utilizará un puerto origen alto *"zzzz" y aleatorio*, tratando de infectar a una computadora con dirección IP *"yyy.yyy.yyy.yyy"* realizando un escaneo de puertos de forma secuencial hacia un puerto destino de la maquina víctima *"wwww"* de la siguiente forma:

xxx.xxx.xxx.xxx **:** zzzz **->** yyy.yyyy.yyy.yyy **:** wwww
xxx.xxx.xxx.xxx **:** zzzz **->** yyy.yyyy.yyy.yyy **:** wwww+1
xxx.xxx.xxx.xxx **:** zzzz **->** yyy.yyyy.yyy.yyy **:** wwww+2
.
xxx.xxx.xxx.xxx **:** zzzz **->** yyy.yyyy.yyy.yyy **:** wwww+n

Dónde: xxx.xxx.xxx.xxx = Ip origen zzzz: Pto Origen yyy.yyy.yyy.yyy = Ip destino wwww: Pto Destino

- Cuando el malware ha logrado infectar a una nueva víctima, repetirá el proceso descrito en otro equipo.

El escaneo de puertos que se realiza puede variar dependiendo del tipo de malware que utiliza esta técnica. Por medio de investigaciones en páginas enfocadas en seguridad como CERT, SANS, Insecure.org, entre otras, se observa que el malware frecuentemente realiza escaneo de puertos de forma secuencial y hacia puertos altos. Debido a que generalmente, los puertos altos no son muy utilizados y muchos usuarios tienen abiertos estos puertos.

Generalmente el malware que ejecuta la técnica de escaneo de puertos utiliza el protocolo orientado a conexión (TCP) realizando un "handshake". Sin embargo, algunos malware utilizan el protocolo no orientado a conexión (UDP) enviando paquetes y esperando la respuesta del equipo víctima, o utilizan el protocolo ICMP mediante un "echo request" (ping), buscando puertos libres.

Generalmente los gusanos es el malware que utiliza esta técnica descrita, sin embargo se puede presentar que otro tipo de malware haga uso de esta técnica por la capacidad de polimorfismo que tiene el malware. Algunos ejemplos de malware conocido que infectan a sus víctimas por medio de un escaneo de puertos son: Stumbler, Blaster, entre otros.

Escaneo de IP's.
El malware que trate de infectar a diferentes equipos mediante un escaneo de IP's presenta un comportamiento descrito de la siguiente forma:

- El malware que ha logrado infectar a una computadora con dirección IP *"xxx.xxx.xxx.xxx"*, buscará infectar a otra computadora perteneciente a la misma red o a otra red distinta con dirección IP *"yyy.yyy.yyy.yyy",* explotando la misma vulnerabilidad que encontró en el equipo víctima.
- Generalmente el malware que emplea esta técnica utiliza un puerto origen alto. Puede que en cada intento de infectar a una víctima varié el puerto origen utilizado. Por medio del socket establecido en la computadora infectada *"xxx.xxx.xxx.xxx:zzzz", el malware* tratará de infectar a una computadora con dirección IP *"yyy.yyy.yyy.yyy"* realizando un escaneo de IP's de forma secuencial en el tercer o cuarto octeto de la dirección IP de la víctima. El escaneo de IP's tiene como característica que el puerto destino *"wwww"* siempre será el mismo:

xxx.xxx.xxx.xxx **:**zzzzz**->** yyy.yyyy.yyy.yyy **:** wwww
xxx.xxx.xxx.xxx **:** zzzzz **->** yyy.yyyy.yyy.yyy+1 **:** wwww
.
xxx.xxx.xxx.xxx **:** zzzzz **->** yyy.yyyy.yyy.yyy+n **:** wwww

Ó

xxx.xxx.xxx.xxx **:** zzzzz **->** yyy.yyyy.yyy.yyy **:** wwww
xxx.xxx.xxx.xxx **:** zzzzz **->** yyy.yyyy.yyy+1.yyy **:** wwww
.
xxx.xxx.xxx.xxx **:** zzzzz **->** yyy.yyyy.yyy+n.yyy **:** wwww

Dónde: xxx.xxx.xxx.xxx = Ip origen zzzz: Pto Origen yyy.yyy.yyy.yyy = Ip destino wwww: Pto Destino

- Cuando el malware ha logrado infectar a una nueva víctima, repetirá el proceso descrito en otro equipo.

Al igual que en un escaneo de puertos, el protocolo utilizado por el malware al realizar un escaneo de IP's puede variar. Generalmente el malware que emplea esta técnica utiliza el protocolo TCP o el protocolo UDP.

Los gusanos es el malware que comúnmente utiliza la técnica de escaneo de IP's. Pero al igual que en un escaneo de puertos, puede que otro tipo de malware ocupe esta técnica. Algunos malware conocidos que conocidos que realizan un escaneo de IP's son: Conficker, Sasser, SQL Inyection, etc. Todos ellos tienen como característica el realizar escaneos de IP's de forma secuencial hacia un puerto destino igual en la víctima.

3) Consumir recursos.
El malware que presenta este comportamiento se caracteriza por tratar de consumir todos los recursos asignados en la computadora que ha infectado. Generalmente el malware genera cantidades de tráfico excesivas e impide que la víctima utilice eficazmente los recursos que se le han asignado. El comportamiento presentado por el malware que emplea esta técnica se describe de la siguiente forma:

- Generalmente el malware tratará de infectar a otras víctimas mediante un escaneo de puertos o escaneo de IP's.

- Cuando el malware ha logrado infectar a varios equipos, utilizará a todas las victimas para realizar un ataque de negación de servicios utilizando todo, o la mayor parte, del ancho de banda asignado a las computadoras zombies para enviar múltiples conexiones hacia la computadora víctima, generalmente servidor, con el objetivo de saturar y tirar la conexión de la computadora víctima.
- En el ataque DDoS realizado tendrá como objetivo enviar múltiples conexiones de máquinas infectadas hacia una dirección IP *"yyy.yyy.yyy.yyy"* y un puerto destino *"wwww"*, generando un tráfico excesivo. En las maquinas zombies puede haber variaciones en el uso del puerto origen de cada una de ellas y pueden pertenecer a una misma red o a redes distintas. El malware que emplea esta técnica actúa de la siguiente forma:

xxx.xxx.xxx.xxx : zzzz -> yyy.yyy.yyy.yyy : wwww
xxx.xxx.xxx.xxx : zzzz -> yyy.yyy.yyy.yyy : wwww
xxx.xxx.xxx.xxx : zzzz -> yyy.yyy.yyy.yyy : wwww
.
xxx.xxx.xxx.xxx : zzzz -> yyy.yyy.yyy.yyy : wwww
Dónde: xxx.xxx.xxx.xxx = Cualquier Ip origen zzzzz: Cualquier Puerto Origen
yyy.yyy.yyy.yyy = Ip destino wwww: Puerto Destino

Generalmente el malware que utiliza esta técnica son las bootnets. Este malware se encarga de infectar equipos y realizar ataques de negación de servicios (DoS) sobre la víctima. Las *botnets* se suelen instalar mediante *backdoors* o un gusano programado para que realice esta esa función específica. Sin embargo también hay virus y troyanos que emplean esta técnica.

Los ataques DoS o DDoS frecuentemente usan el protocolo TCP en su ejecución, aunque cierto malware también utiliza el protocolo UDP o aplican la técnica de "Ping of death" del puerto ICMP. La cantidad de tráfico generado por un ataque DoS es excesiva. El ataque DoS tiene como principal objetivo el saturar la máquina víctima y tirar, o interrumpir el servicio.

4) Actualización del malware y envió de información hacia direcciones Ip'S desconocidas por el usuario

La mayoría del malware una vez instalado en la victima, tratará de actualizarse constantemente, descargar nuevo malware o enviar información perteneciente de la máquina víctima a su creador. Hay que poner especial atención en este comportamiento si el malware ha infectado un servidor, debido a que puede haber fuga de información confidencial. El malware que utiliza esta técnica generalmente realiza las siguientes acciones:

- Actualizarse.
- Recolectar información de la maquina infectada con dirección IP *"xxx.xxx.xxx.xxx"* y enviarla hacia una dirección IP *"yyy.yyy.yyy.yyy"* utilizando puertos origen generalmente altos y de forma secuencial *"zzzz"*. De la siguiente forma:

xxx.xxx.xxx.xxx : zzzz -> yyy.yyy.yyy.yyy : wwww
xxx.xxx.xxx.xxx : zzzz+1 -> yyy.yyy.yyy.yyy : wwww
xxx.xxx.xxx.xxx : zzzz+2 -> yyy.yyy.yyy.yyy : wwww
.
xxx.xxx.xxx.xxx : zzzz+3 -> yyy.yyy.yyy.yyy : wwww
Dónde: xxx.xxx.xxx.xxx = Ip origen zzzz: Puerto Origen
yyy.yyy.yyy.yyy = Cualquier Ip destino wwww: Cualquier Puerto Destino

El puerto destino de las maquinas externas no es de gran importancia, debido a que el malware puede usar puertos destinos aleatorios. También hay que poner especial atención en el tráfico generado por cada conexión. A mayor cantidad de tráfico generado significa mayor fuga de información.

Normalmente el malware que utiliza esta técnica establece conexiones TCP hacia direcciones IP externas a la empresa y de rango sospechoso. Sin embargo, el malware también puede enviar información por medio del protocolo UDP.

Con base en las cuatro técnicas descritas, se pretende crear un plugin en el software Nfsen capaz de detectar malware que presente alguno de los comportamientos descritos en la red central de la institución. Hay que señalar que el objetivo del tema de tesis es detectar malware por medio del monitoreo de red, utilizando el protocolo Netflow, de tal modo que cualquier malware que no presente un comportamiento en la red o no cumpla con alguna de los cuatro métodos analizados no será detectado.

Sin embargo, el proyecto de tesis puede ser ampliado hacia la detección de cualquier tipo de malware o la creación de un IDS o IPS.

Para la creación de este plugin, fue necesario conocer el esquema de seguridad que se tiene implementado en la institución, este esquema sirvió como referencia para conocer las áreas más vulnerables en la institución.

4.4 Esquema de seguridad implementado en la institución

En la institución se tiene implementado un esquema de seguridad robusto; En cada red descrita se tienen implementadas diversas herramientas de seguridad, como lo son: antivirus, antispam, firewall, IDS, IPS, listas de control de acceso (ACL), iptables, entre otras. Además todo software nuevo a instalar en cualquier computadora deberá de ser verificado minuciosamente antes de su instalación. La figura 4.1 muestra a grandes rasgos el esquema de seguridad implementado.

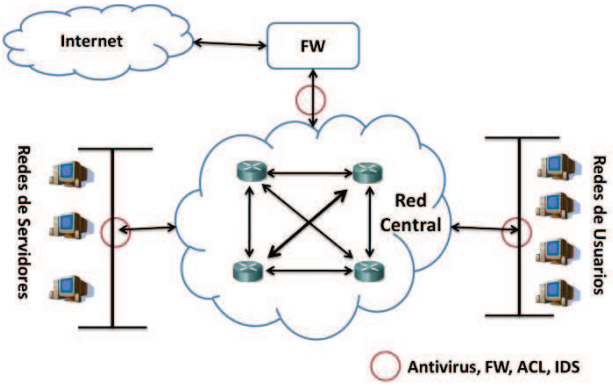


Figura 4.1 Esquema de seguridad implementado sobre la institución

Como se observa en esta imagen se hace énfasis en la protección de redes de usuarios y redes de servidores. Sin embargo, se puede presentar el caso de que un usuario accidentalmente o intencionalmente instale algún malware en la red central de la institución y este malware no sea detectado por las herramientas de seguridad instaladas, principalmente por ser un malware de día cero del cual no se tiene información alguna.

Por este motivo el objetivo del plugin a crear en el software Nfsen es trabajar en conjunto con las distintas herramientas de seguridad instaladas y la posible detección de malware de día cero, además con el desarrollo de este plugin se pretende una solución innovadora en la detección de malware.

4.5 Creación del plugin escaneo en Nfsen.

Como se describió en el capítulo tres, Nfsen permite crear módulos de programación adicionales al software y adaptados de acuerdo a las necesidades requeridas. En este proyecto de tesis se decidió programar un módulo enfocado en la detección de malware basado en los cuatro comportamientos descritos por las siguientes razones:

- La opción de alertas proporcionada por el software Nfsen permite acotar la información a rangos muy específicos, sin embargo no es posible al ejecutar la alerta, hacer referencia de las direcciones IP que se han detectado anormales, esto presenta un problema al tratar de analizar qué dirección IP es la causante de la anomalía detectada en la alerta.
- En los profiles soportados por Nfsen es posible crear un profile que observe un aumento en el tráfico anormal al promedio; por medio de análisis basados en filtros es posible obtener las direcciones IP que están causando este aumento inusual, sin embargo este procedimiento tiene que ser ejecutado manualmente por el usuario.

El objetivo de crear el plugin enfocado en la detección del malware es tener un módulo completamente automático y con una mínima intervención humana. El plugin creado se ejecutará cada que el software Nfsen reciba un nuevo archivo nfcapd. Este plugin se encargará de analizar el último archivo obtenido en búsqueda de los cuatro comportamientos del malware descritos. En caso de encontrar alguna anomalía, el plugin se encargará de notificar inmediatamente vía email de la anomalía detectada.

4.5.1 Estrategia de desarrollo del plugin escaneo.

El plugin enfocado en la detección de malware llamado *"escaneo"* será un módulo que formará parte del software Nfsen, encargándose de realizar las siguientes acciones en cada ejecución:

- Obtener e interpretar el último archivo nfcapd generado.
- Separar el contenido del archivo por medio de expresiones regulares.
- Verificar el resultado del punto anterior en búsqueda de patrones de comportamiento anormales, específicamente se buscará que en la red central se realice:
  - Escaneo de puertos,
  - Escaneo de IP's
  - Ataque de negación de servicios
  - Envió de información hacia el exterior.
- Generar el archivo de salida "anomalías" en caso de encontrar algún comportamiento anormal.
- Notificar inmediatamente vía email de la anomalía encontrada y guardar esta anomalía en una base de datos MySQL.
- Visualizar los resultados de la ejecución del plugin backend "escaneo.pm" Utilizando el módulo frontend "escaneo.php".

Los lenguajes de programación utilizados para la creación de este plugin fueron Perl y PHP, debido a que el software Nfsen solo permite la creación de módulos backend en perl y módulos frontend en PHP y HTML

4.5.2 Componentes del plugin Escaneo.

La tabla 4.1 tiene el objetivo de mostrar las funciones requeridas para la ejecución del plugin escaneo.

Tabla 4.1 Componentes del plugin escaneo

Plugin:	Escaneo.pm	Escaneo.php
Objetivo:	Ejecutar el plugin en cada actualización del software Nfsen en búsqueda de comportamientos anormales descritos en la sección 4.3	Mostrar los resultados de la ejecución del módulo "escaneo.pm" en la interfaz web del software Nfsen.
Lenguaje utilizado	Perl	Php
Funciones creadas	✓ Init ✓ Run ✓ Cleanup ✓ Separa ✓ Agrupa ✓ Agrupa_ip ✓ Guarda ✓ Envia_correo ✓ Epuertos ✓ Compara ✓ Eips ✓ Compara_Ip ✓ Exterior ✓ Compara_exterior ✓ DoS ✓ ComparaDos	✓ Escaneo_ParseInput ✓ Escaneo_run

Como se observa en esta tabla, el módulo escaneo.pm tiene como objetivo el analizar en cada actualización del software Nfsen el último archivo nfcapd generado por el colector Nfdump en búsqueda del comportamiento definido en la sección 4.3

Si en algún momento se llegase a detectar un comportamiento anormal que cumple con alguno de los puntos descritos se creará el archivo "anomalías" con el objetivo de guardar la información detectada como anormal y mandar llamar a la función envía_correo, notificando sobre el evento encontrado.

El módulo "escaneo.php" tiene como objetivo el mostrar los resultados de la ejecución del módulo "escaneo.pm" en la interfaz web del software Nfsen.

La descripción de cada función utilizada se mostrará mediante su diagrama de flujo específico y una breve explicación de las tareas ejecutadas por cada función.

El diagrama de flujo general de la ejecución del plugin escaneo se muestra en la figura 4.2.

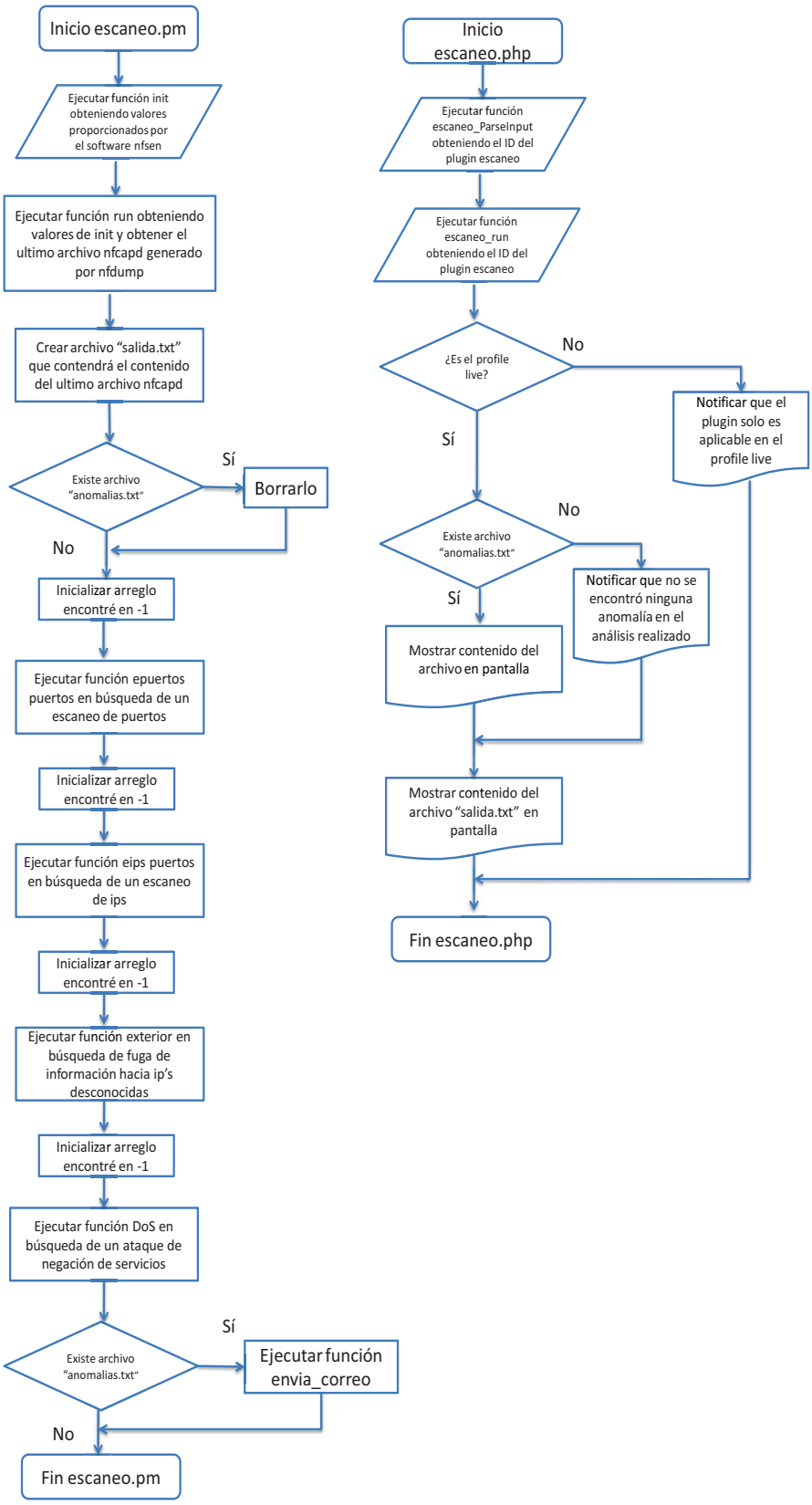


Figura 4.2 Diagrama de flujo general del plugin escaneo

4.5.3 Funcionamiento del plugin escaneo.

4.5.3.1 Funcionamiento del modulo *"escaneo.pm"*.

El objetivo del plugin backend 'escaneo.pm' es analizar el último archivo obtenido por el software Nfdump en búsqueda de patrones de comportamiento anormales presentes en la red central de la institución. Específicamente se buscará aquel malware que haya realizado un escaneo de puertos o un escaneo de IP's, o que esté enviando información mayor a 5 Mb de direcciones IP pertenecientes a redes de servidores hacia direcciones IP externas, o que realice ataques DoS o DDoS.

Las siguientes tres funciones son especiales en el uso de Nfsen. La figura 4.3 muestra el diagrama de flujo de cada una de ellas:

Init: Función especial creada en cualquier plugin a desarrollar en Nfsen. Esta función se encarga de contener toda la información necesaria para la ejecución de la función run.

Run: Función encargada de ejecutar el plugin en cada actualización del software Nfsen. Esta función recibe parámetros obtenidos de función init. Además se encarga de obtener el último archivo nfcapd generado, guardar este archivo en el arreglo "registros" y ejecutar a las funciones "epuertos, eips, DoS y exterior" en búsqueda de comportamientos anormales. Si la ejecución de las funciones mencionadas arroja un comportamiento anormal, se ejecuta a la función "envía_correo".

Cleanup: Función especial encargada de terminar la ejecución del plugin al momento de finalizar el software Nfsen.

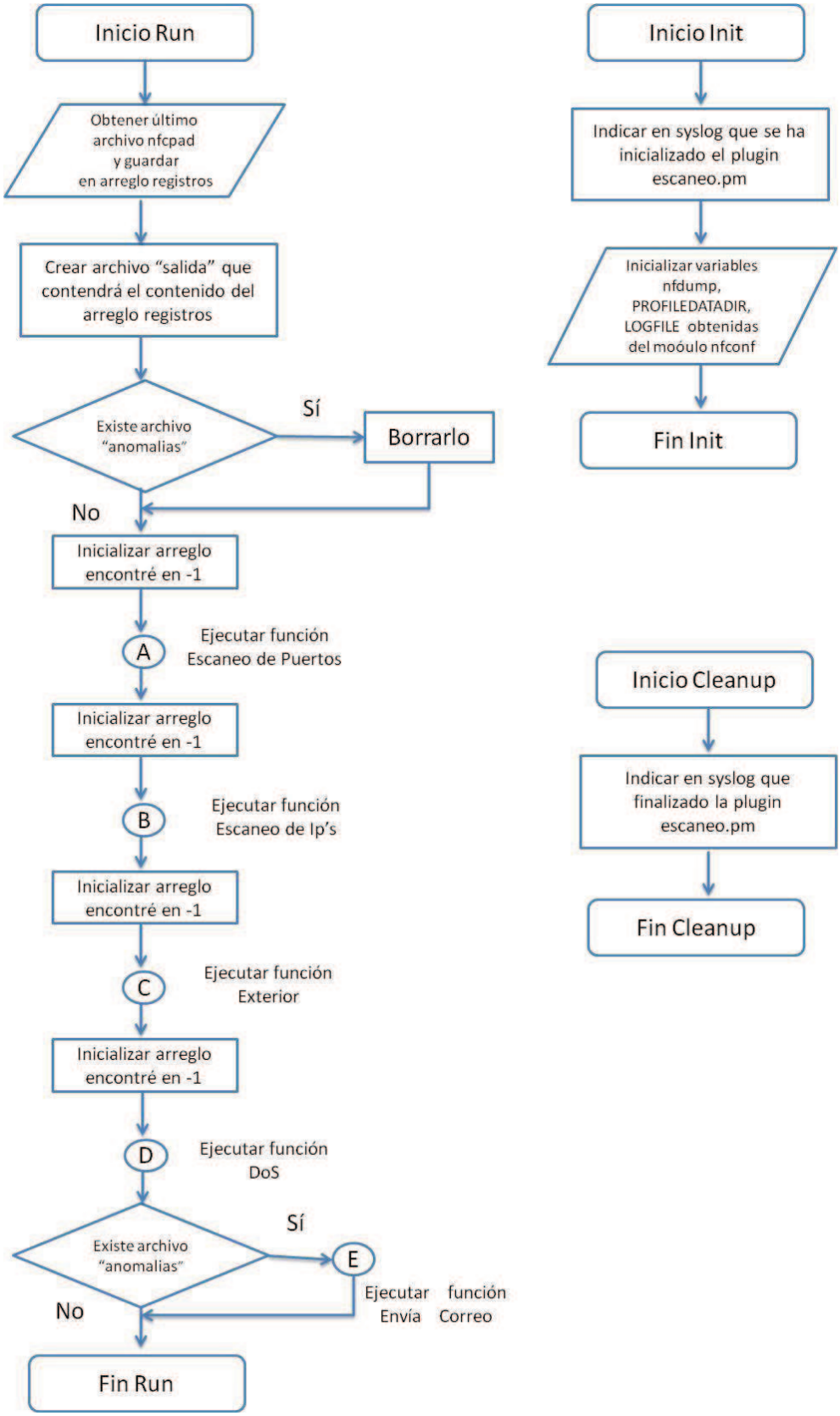


Figura 4.3 Diagrama de flujo de las funciones Init, Run y Cleanup

Las siguientes funciones fueron creadas con el objetivo de separar la información por medio de expresiones regulares:

Separa: Función creada para separar el contenido del arreglo "registros" por medio de expresiones regulares y guardar el resultado en el arreglo "ip" que contendrá la información con el formato "protocolo ip_origen**:**puerto_origen **->** ip_destino**:**puerto_destino paquetes tráfico".

El diagrama de flujo de esta función se muestra en la figura 4.4.

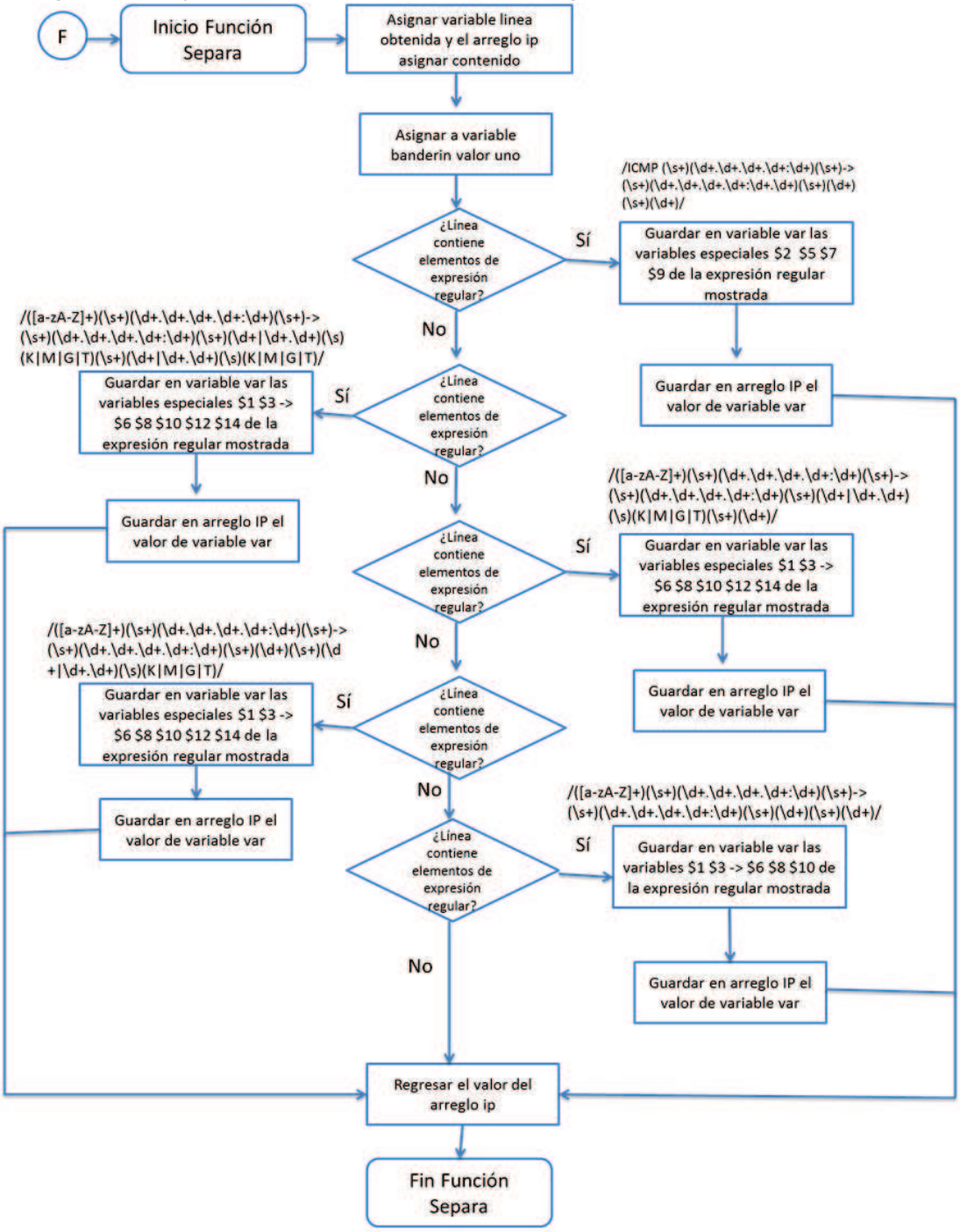


Figura 4.4 Diagrama de flujo de la función Separa

Agrupa: Función encargada de separar la información contenida en el arreglo "ip" por medio de expresiones regulares en el siguiente formato:

```
      aaa       xxx.xxx.xxx.xxx :  zzzz   ->  yyy.yyy.yyy.yyy :  wwww       bbb
(protocolo)   (ip ori)          (pto ori)        (ip dst)          (pto dst)  (trafico)
```

Cada valor separado se guarda en variables especiales con el objetivo de realizar análisis en estas variables en búsqueda de un escaneo de puertos y un ataque Dos o DDoS. Esta función es utilizada por las funciones "epuertos y DoS".

El diagrama de flujo de esta función se muestra en la figura 4.5.

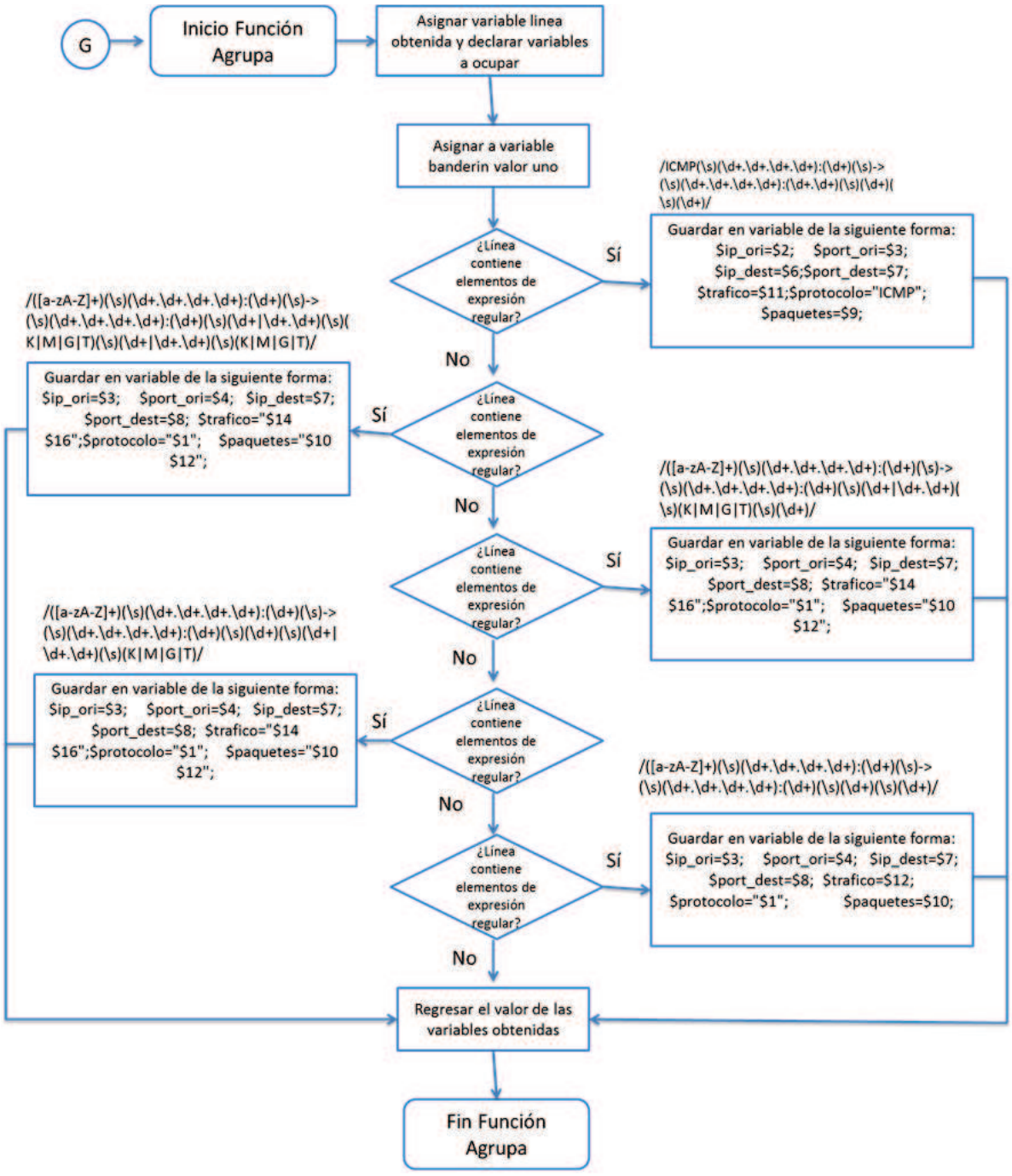


Figura 4.5 Diagrama de flujo de la función Agrupa

Agrupa_ip: Función encargada de separar la información contenida en el arreglo "ip" por medio de expresiones regulares en el siguiente formato:

```
        aaa    xxx.xxx.xxx.xxx  : zzzz  ->    yyy.yyy .    yyy  .   yyy    : wwww    bbb
(Protocolo)   (ip ori)          (pto ori)     (ip dst1)  (ip dst2) (ip dst3)  (pto dst)  (trafico)
```

Cada valor separado se guarda en variables especiales con el objetivo de realizar análisis sobre ellas en búsqueda de un escaneo de IP's y envío de información hacia el exterior. Esta función es utilizada por las funciones "eips y exterior".

El diagrama de flujo de esta función se muestra en la figura 4.6.

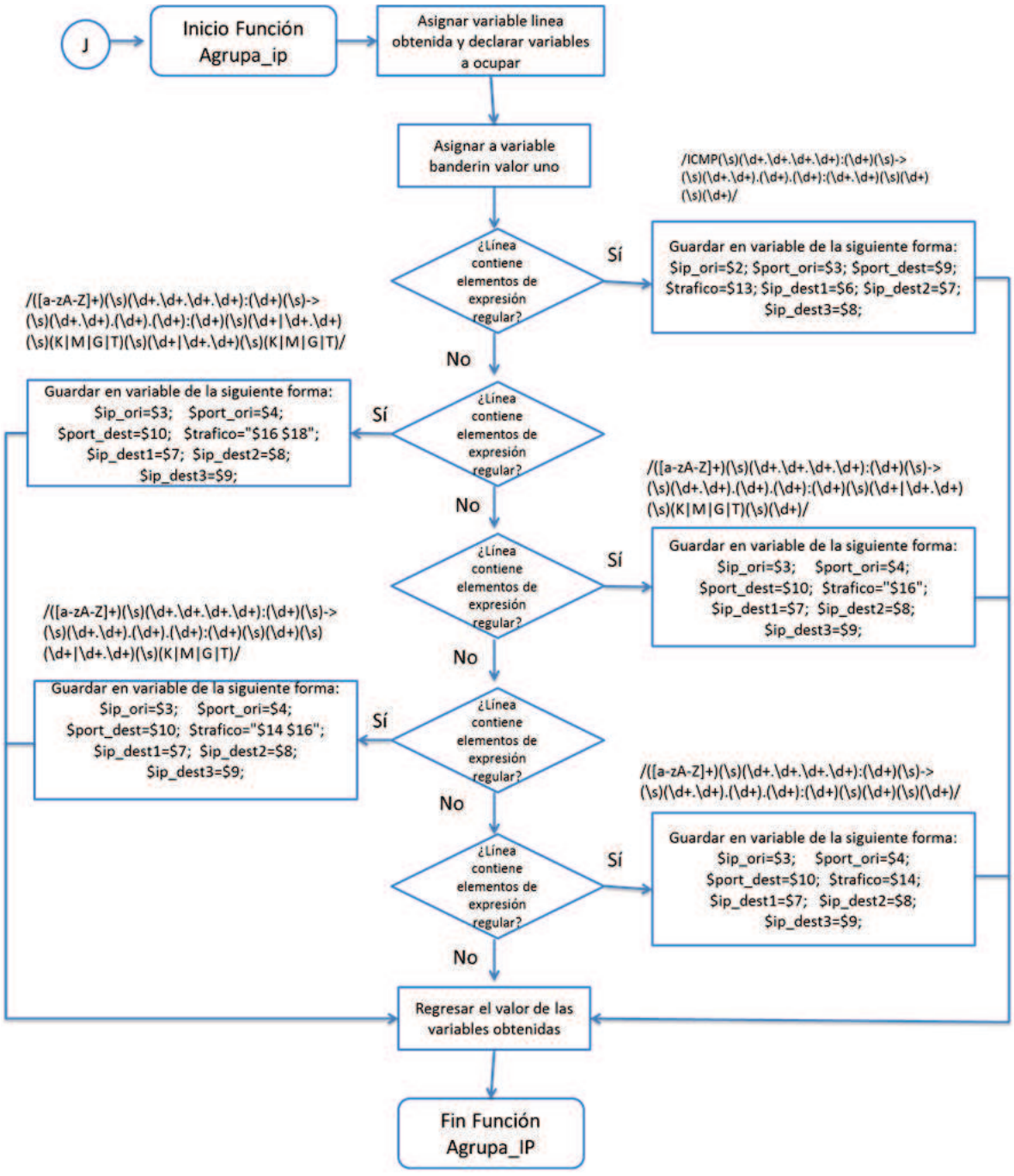


Figura 4.6 Diagrama de flujo de la función Agrupa_Ip

Las siguientes funciones fueron creadas con el objetivo de guardar y notificar en caso de encontrarse alguna anomalía en el análisis realizado.

Guarda: Función encargada de guardar las anomalías encontradas por las funciones epuertos, eips, exterior y DoS en el archivo de texto “anomalías”.

En la figura 4.7 muestra el diagrama de flujo creado para esta función.

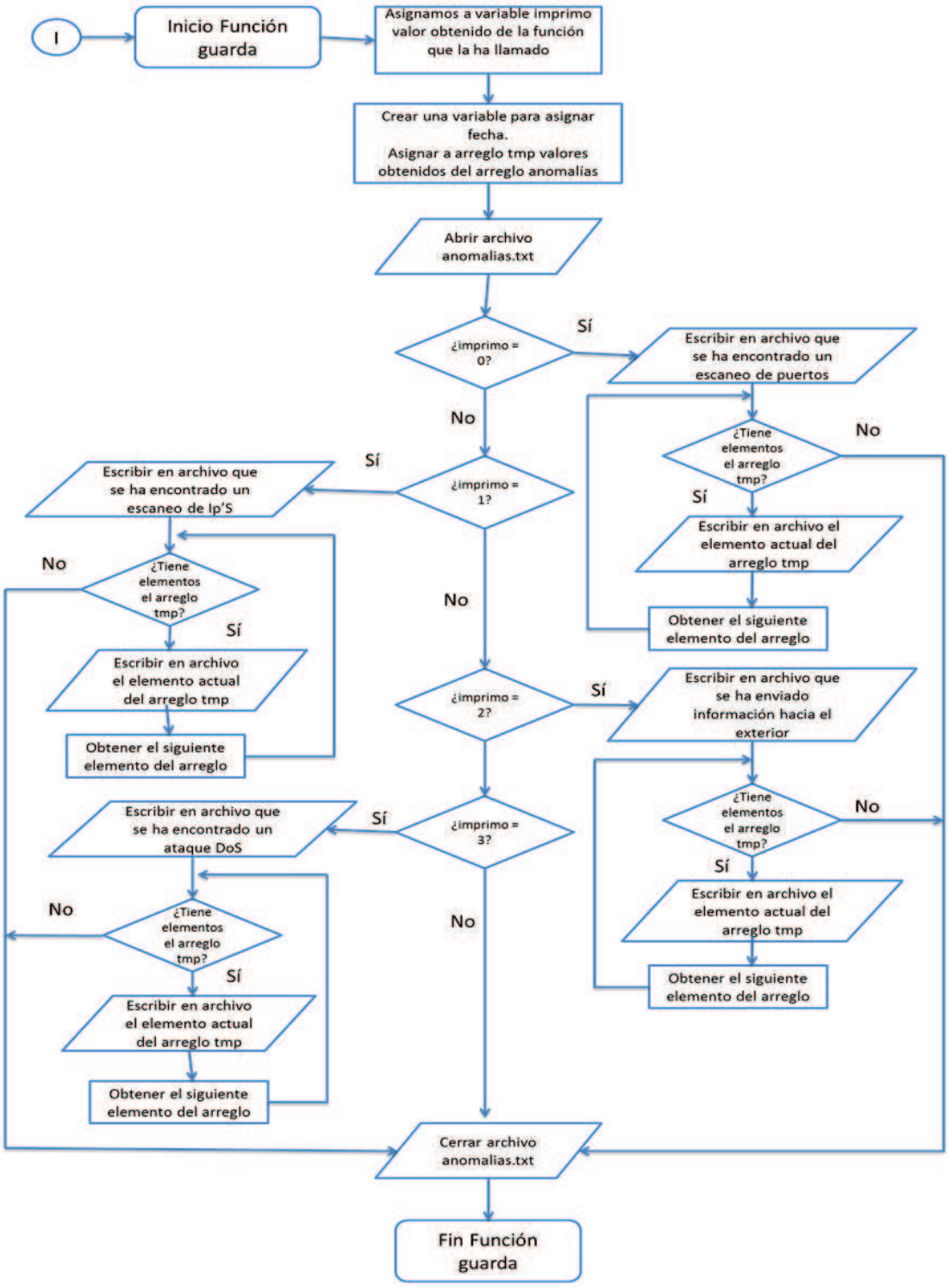


Figura 4.7 Diagrama de flujo de la función Guarda

Envía_correo: Función encargada de enviar un email con el contenido del archivo “anomalías” notificando sobre la(s) anomalía(s) encontrada(s).

Para él envío del email se utiliza al servicio sendmail. En la figura 4.8 se muestra el diagrama de flujo de esta función.

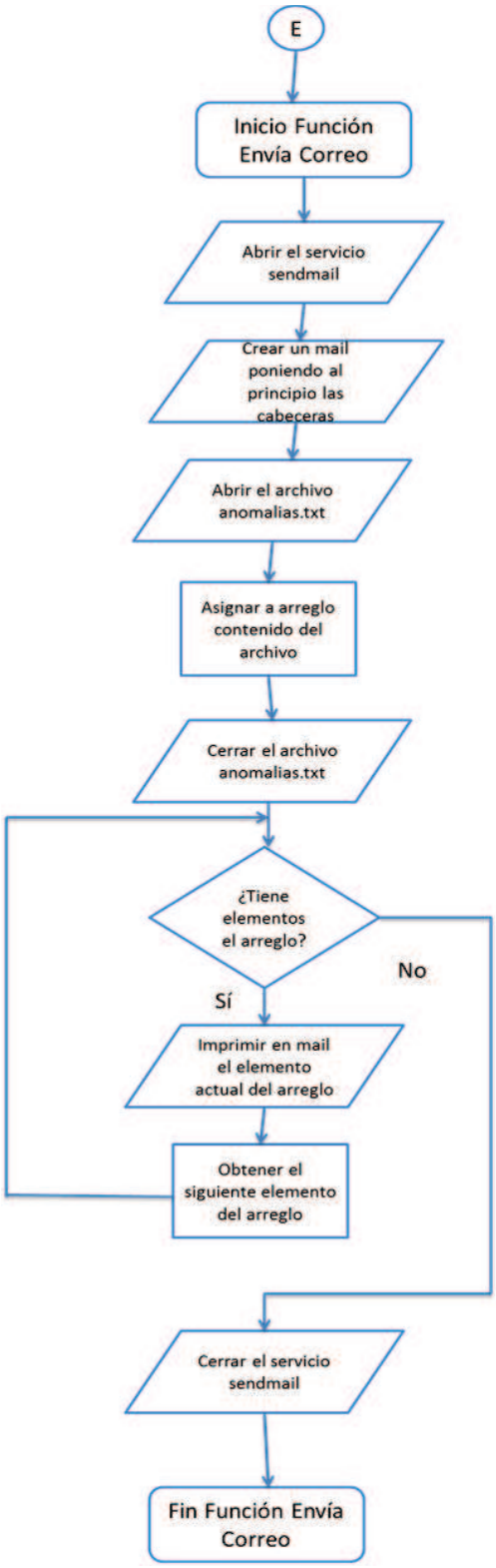


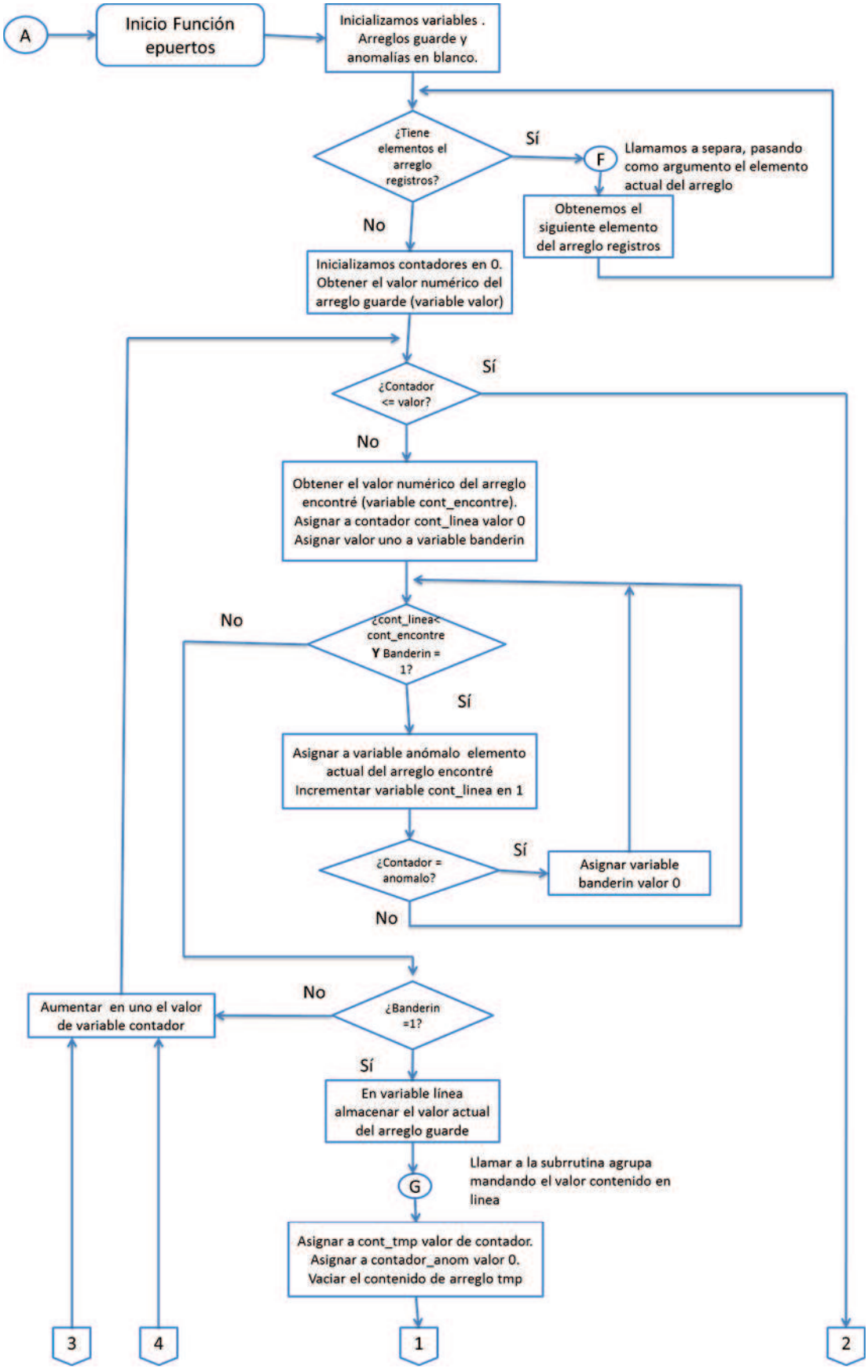
Figura 4.8 Diagrama de flujo de la función Envia_Correo

Las siguientes funciones fueron creadas con el objetivo analizar la información en búsqueda de comportamientos anormales presentes sobre la red interna de la institución:

Epuertos: Función creada con el objetivo de verificar si se ha realizado un escaneo de puertos. Su funcionamiento se describe a detalle a continuación:

1. Recibe el último archivo nfcapd obtenido en la función run y lo guarda en el arreglo *"registros"*.
2. Recorre todos los elementos contenidos en el arreglo *"registros",* separando cada elemento contenido en este arreglo mediante la función separa. El resultado obtenido se guarda en el arreglo *"guarde"*.
3. En la variable *"valor"* se hace referencia a cuantos elementos se han agrupado en el arreglo *"guarde"* (referencia numérica). Se inicializan contador y contador anormal en cero.
4. Se inicializa el primer ciclo while que recorrerá todos los elementos del arreglo *"guarde"* hasta que la variable *"contador"* supere el número de la variable *"valor"*. Cuando se cumpla la acción indicada ir al paso diecisiete.
5. Se verifica si en previas iteraciones se ha detectado algún elemento como anormal. En caso afirmativo se asocia a variable *"banderin"* valor cero.
6. Si el valor de variable *"banderin"* es igual a uno se procede a obtener el elemento específico a analizar del arreglo *"guarde"* con ayuda del valor actual de variable *"contador"*, el resultado se guarda en variable línea. En caso contrario ir al paso quince.
7. Se manda llamar a función agrupa con el objetivo de separar la información contenida en la variable *"linea"* en las variables: *"ip_ori, pto_ori, ip_dest, pto_dest, trafico"*.
8. El valor contenido en variable *"contador"* se asocia a variable *"cont_tmp"*. El arreglo *"tmp"* que contiene elementos detectados como anormales se vacía.
9. Se ejecuta el segundo ciclo while que recorrerá todos los elementos del arreglo *"guarde"* hasta que la variable *"cont_tmp"* supere el número de la variable valor. En caso de que la variable *"cont_tmp"* sea mayor a la variable *"valor"* ir al paso catorce.
   El objetivo del 2º ciclo while es recorrer los elementos que se tengan por debajo del valor actual de la variable *"contador"*.
10. Se incrementa en uno el valor de *"cont_tmp".* Se ejecuta la misma acción indicada en el paso cinco. La diferencia radica en que si la variable *"banderin"* es igual a cero, se regresa al paso nueve.
11. Se ejecuta la misma acción indicada en el paso seis. La diferencia radica en obtener el elemento específico de la arreglo *"guarde"* con ayuda del valor actual de la variable *"cont_tmp"*.
12. Se ejecuta la misma acción indicada en el paso siete. La diferencia radica en que se guardan los resultados en las variables temporales *"ip_ori_tmp, pto_ori_tmp, ip_dest_tmp, pto_dest_tmp, trafico_tmp".*
13. Se ejecuta la función compara en búsqueda de escaneo de puertos, pasando como argumentos las variables obtenidas del paso siete y once.
14. Regresar al paso nueve.
15. Si el valor de *"contador_anom"* obtenido como resultado de la función compara es mayor o igual a cinco, se guardan los elementos contenidos en arreglo *"tmp"* en el arreglo *"anomalias"*.
16. Incrementar el valor en uno de variable *"contador"* y regresar al paso cuatro.
17. Si existe el arreglo *"anomalías"*, asociar a variable imprimo valor cero y ejecutar función guarda, pasando como argumento el valor contenido en variable *"imprimo"* y el arreglo *"anomalias"*.
18. Fin función escaneo de puertos.

En la figura 4.9 se muestra el diagrama de flujo creado para esta función.



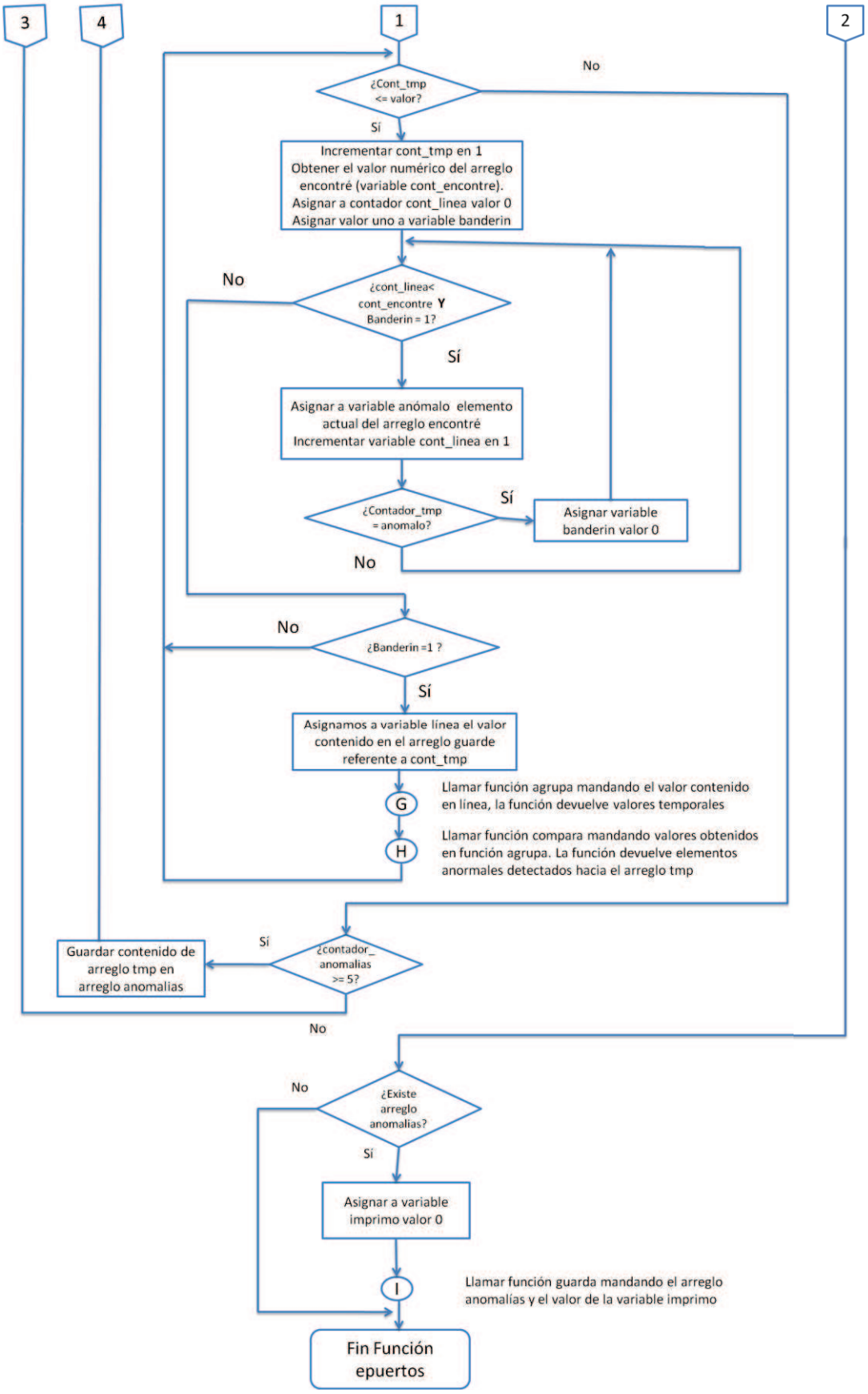


Figura 4.9 Diagrama de flujo de la función epuertos

Compara: Función utilizada por epuertos con el objetivo de verificar el elemento actual y el elemento temporal en búsqueda de un aumento en uno del puerto destino de la siguiente forma:

Ip_ori = Ip_ori_tmp **Y** Ip_dest=Ip_dest_tmp **Y** Port_dest = Port_dest_tmp.

En caso de detectar un elemento anormal, este elemento se guarda en el arreglo *"tmp"* y se incrementa el valor de *"contador_anomal"* en uno.

El diagrama de flujo de esta función se muestra en la figura 4.10.

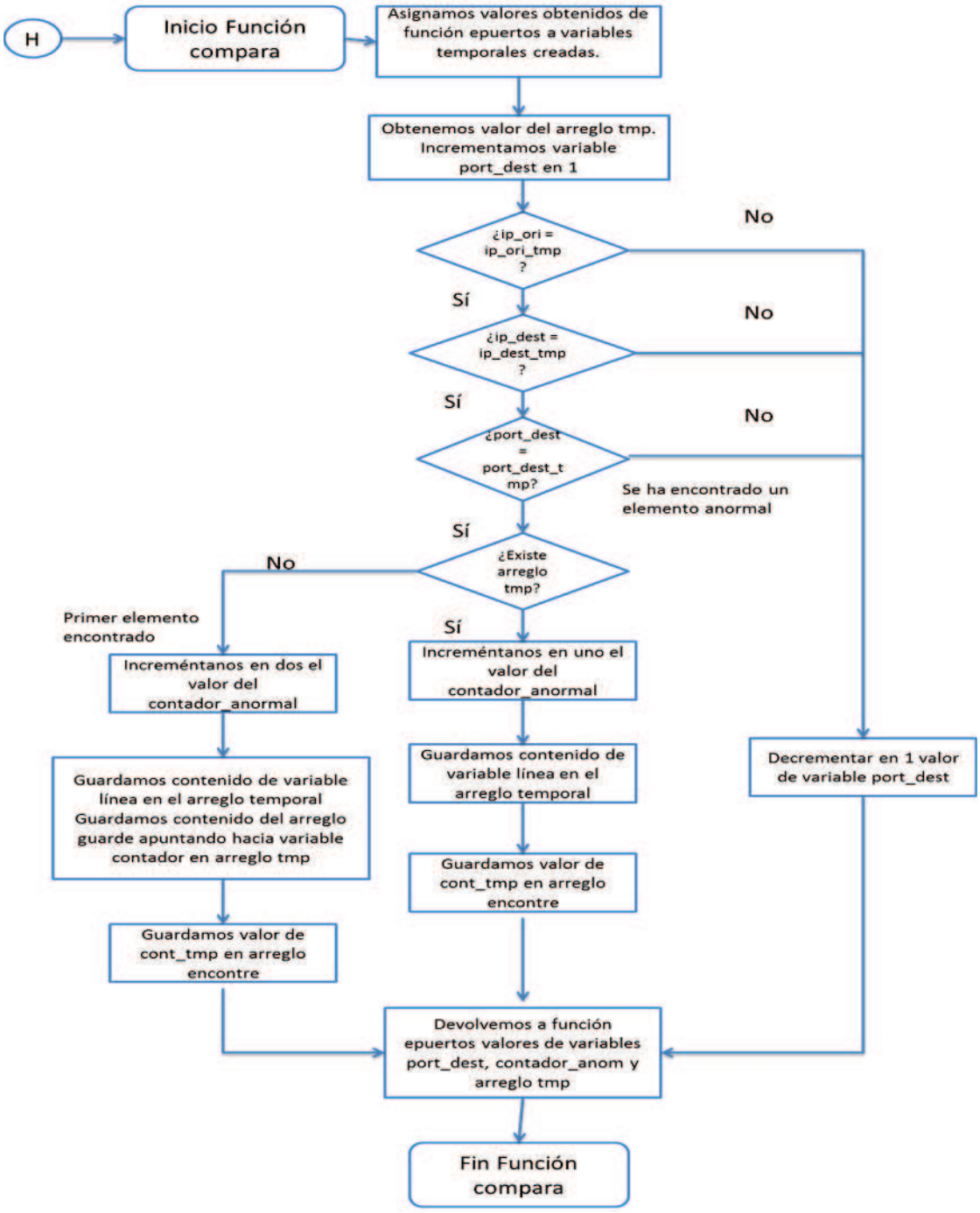
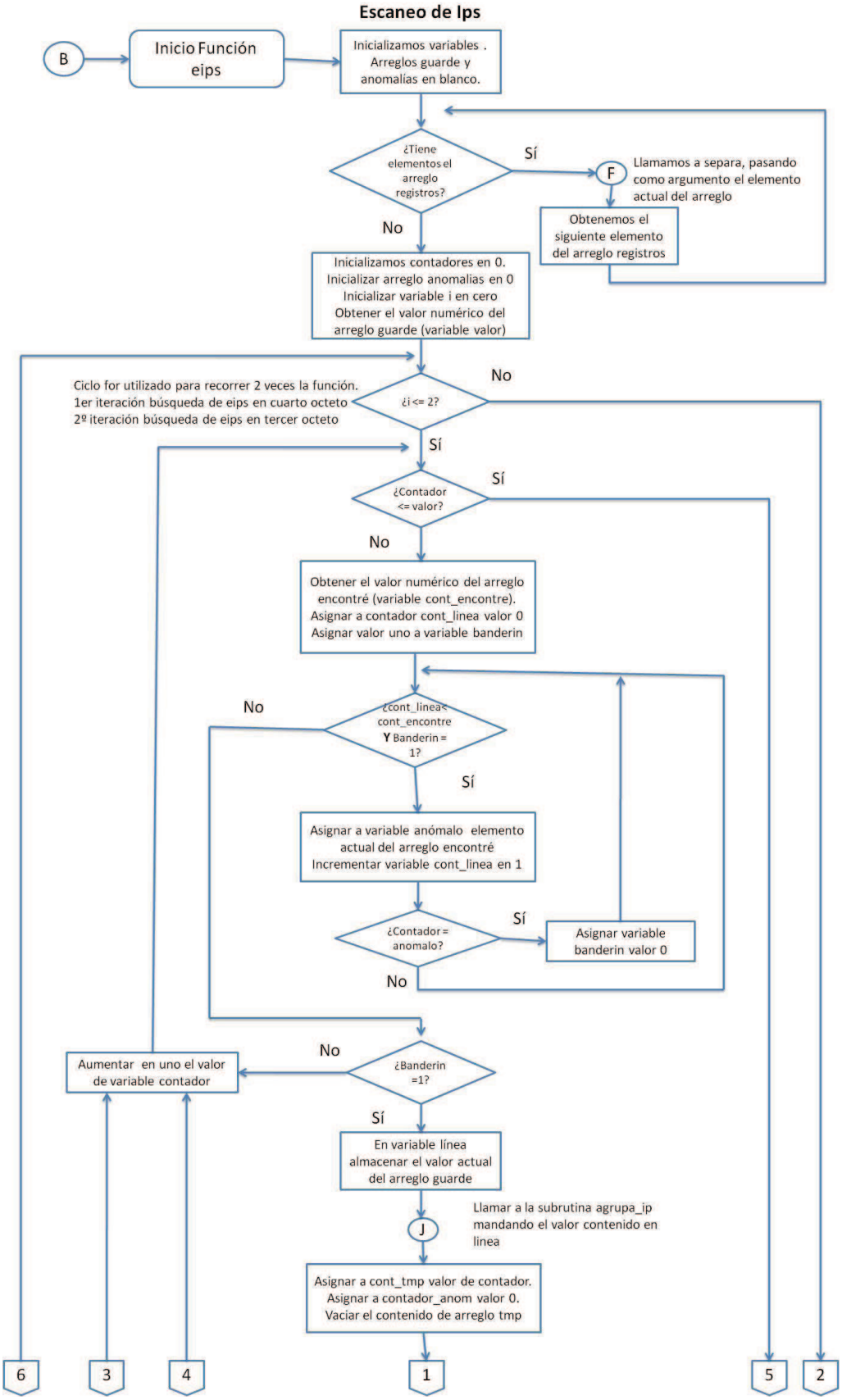


Figura 4.10 Diagrama de flujo de la función compara

Eips: Función creada con el objetivo de verificar si se ha realizado un escaneo de IP's en el tercer o cuarto octeto. Su funcionamiento se describe a detalle a continuación:

1. Recibe el ultimo archivo nfcapd obtenido en la función run y guardado en el arreglo *"registros"*.
2. Recorre todos los elementos contenidos en el arreglo *"registros",* separando cada elemento contenido en este arreglo ejecutando a la función separa. El resultado obtenido se guarda en el arreglo *"guarde"*.
3. Se inicializa un ciclo for que se ejecutará dos veces: En la primera iteración se buscará un escaneo de IP's en el cuarto octeto; en la segunda iteración se buscará un escaneo de IP's en el tercer octeto. En la tercera iteración ir al paso diecinueve.
4. En la variable *"valor"* se hace referencia a cuantos elementos se han agrupado en el arreglo *"guarde"* (referencia numérica). Se inicializan variables *"contador"* y *"contador anormal"* en cero.
5. Se inicializa el primer ciclo while que recorrerá todos los elementos del arreglo *"guarde"* hasta que la variable *"contador"* supere el número de la variable *"valor"*. Cuando se cumpla esta acción ir al paso dieciocho.
6. Se verifica si en previas interacciones se ha encontrado algún elemento como anormal. En caso afirmativo se asocia a variable *"banderin"* valor cero.
7. Si el valor de variable *"banderin"* es igual a uno se procede a obtener el elemento específico a analizar del arreglo *"guarde"* con ayuda del valor actual de variable *"contador"*, el resultado se guarda en variable "línea". En caso contrario ir al paso diecisiete.
8. Se ejecuta la función *"agrupa_ip"* con el objetivo de separar la información contenida en la variable *"linea"* en variables: *"ip_ori, pto_ori, ip_dest1, ip_dest2, ip_dest3, pto_dest, trafico"*.
9. El valor contenido en variable "contador" se asocia a variable *"cont_tmp"*. El arreglo *"tmp"* que contiene elementos detectados como anormales se vacía.
10. Se ejecuta el 2º ciclo while que recorrerá todos los elementos del arreglo *"guarde"* hasta que la variable *"cont_tmp"* supere el número de la variable valor. En caso de que la variable *"cont_tmp"* sea mayor a la variable *"valor"* ir al paso quince.
    El objetivo del 2º ciclo while es recorrer los elementos que se tengan por debajo del valor actual de la variable *"contador"*.
11. Se incrementa en uno el valor de *"cont_tmp".* Se ejecuta la misma acción indicada en el paso seis.
12. Se ejecuta la misma acción indicada en el paso siete. La diferencia radica en obtener el elemento específico del arreglo *"guarde"* con ayuda del valor actual de la variable *"cont_tmp"*, y en que si la variable *"banderin"* es igual a cero se regresa al paso diez.
13. Se ejecuta la misma acción indicada en el paso ocho. La diferencia radica en que se guardan los resultados en las variables temporales *"ip_ori_tmp, pto_ori_tmp, ip_dest1_tmp, ip_dest2_tmp, ip_dest3_tmp, pto_dest_tmp, trafico_tmp"*.
14. Se ejecuta la función compara_ip en búsqueda de escaneo de Ip's en el tercer o cuarto octeto, depende en que iteración del ciclo for nos encontremos, pasando como argumentos las variables obtenidas del paso ocho y trece.
15. Regresar al paso diez.
16. Si el valor de *"contador_anom"* obtenido como resultado de la función compara es mayor o igual a cinco, se guardan los elementos contenidos en el arreglo *"tmp"* en el arreglo *"anomalias"*.
17. Incrementar el valor en uno de variable *"contador"* y regresar al paso cinco.
18. Incrementar en uno valor de variable i y regresar al paso tres.
19. Si existe el arreglo *"anomalías"*, asociar a variable imprimo valor uno y ejecutar función guarda, pasando como argumento el valor contenido en variable *"imprimo"* y el arreglo *"anomalias"*.
20. Fin Función escaneo de Ip's

En la figura 4.11 se muestra el diagrama de flujo creado para esta función.



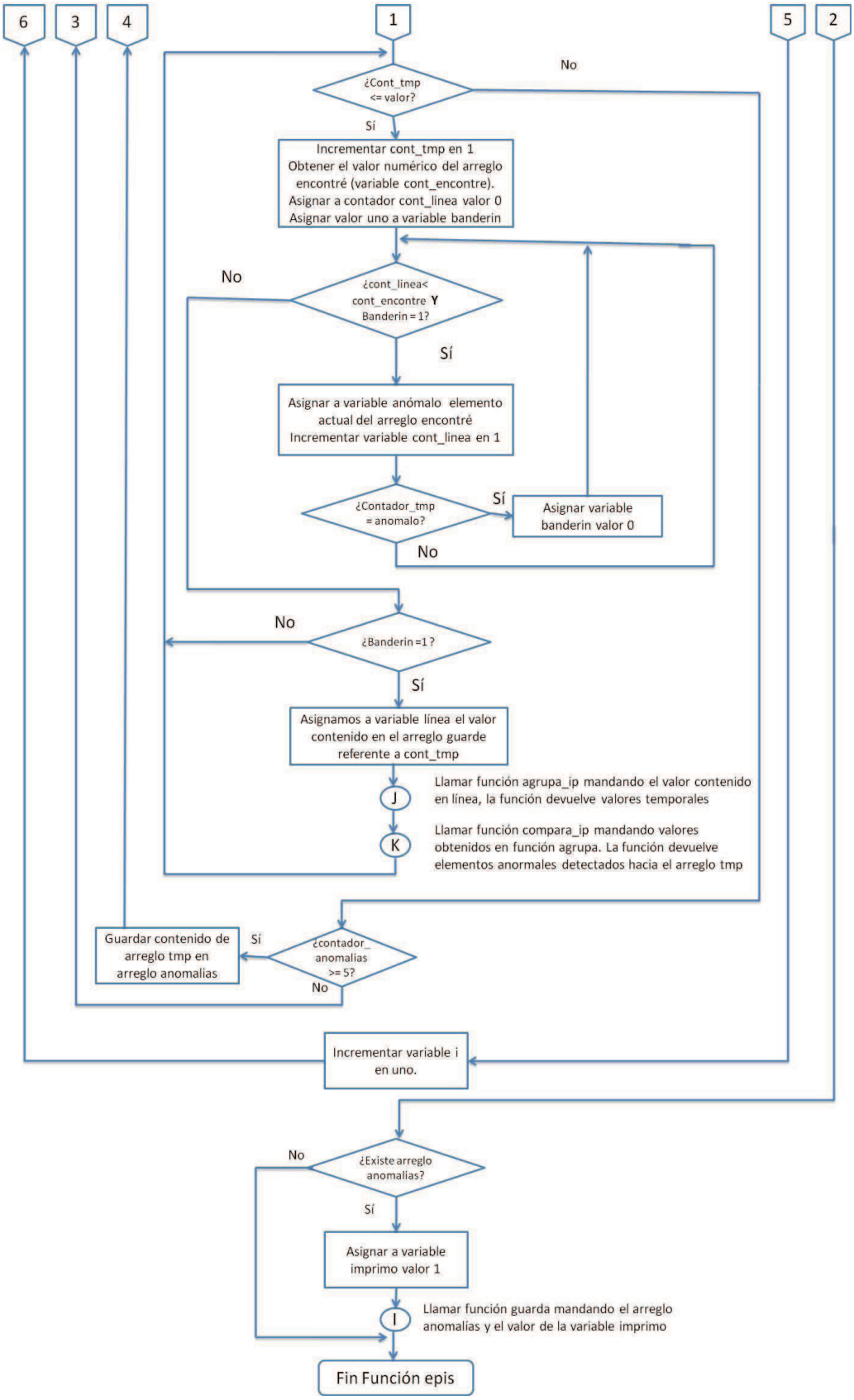


Figura 4.11 Diagrama de flujo de la función eips

compara_ip: Función utilizada por la función "eips" con el objetivo de verificar el elemento actual y el elemento temporal en búsqueda de un aumento en uno del tercer o cuarto octeto de la dirección IP destino, dependiendo de la iteración del ciclo for, de la siguiente forma:

var_ipori = var_ipori_tmp **Y** var_ipdst = var_ipdst_tmp

Donde *"var_ipori"* tiene el socket *"ip_ori:pto_ori"*, *"var_ipori_tmp"* tiene el socket *"ip_ori_tmpi:pto_ori_tmp"*, *"var_ipdst"* tiene el socket *"ip_dst:pto_dst"* y *"var_ipdst_tmp"* tiene el socket *"ip_dst_tmpi:pto_dst_tmp"*

En caso de detectar un elemento anormal, se guarda este elemento en el arreglo *"tmp"* y se incrementa el valor de *"contador_anomal"* en uno, el diagrama de flujo de esta función se muestra en la figura 4.12.

Exterior: Función creada con el objetivo de verificar si se ha enviado información de alguna red de servidores hacia direcciones IP no permitidas (generalmente externas al rango de la institución), con un tráfico mayor a 5 Mb y un aumento en el puerto origen de forma secuencial. Su funcionamiento es similar a la función "epuertos", solo cambia en los siguientes puntos:

- ✓ En el paso siete se manda a llamar a la función agrupa_ip, guardando el resultado en variables *"ip_ori, pto_ori, ip_dest1, ip_dest2, ip_dest3, pto_dest, trafico"*
- ✓ En el paso doce se manda a llamar nuevamente a función agrupa_ip, pero ahora se guarda el resultado en variables temporales: *"ip_ori_tmp, pto_ori_tmp, ip_dest_tmp1, ip_dest_tmp2, ip_dest_tmp3, pto_dest_tmp, trafico_tmp"*
- ✓ En el paso trece se ejecuta la función compara_exterior, pasando como argumento las variables obtenidas de los dos pasos anteriores.
- ✓ En el paso diecisiete se asigna a variable imprimo el valor de dos.

En la figura 4.13 se muestra el diagrama de flujo de esta función.

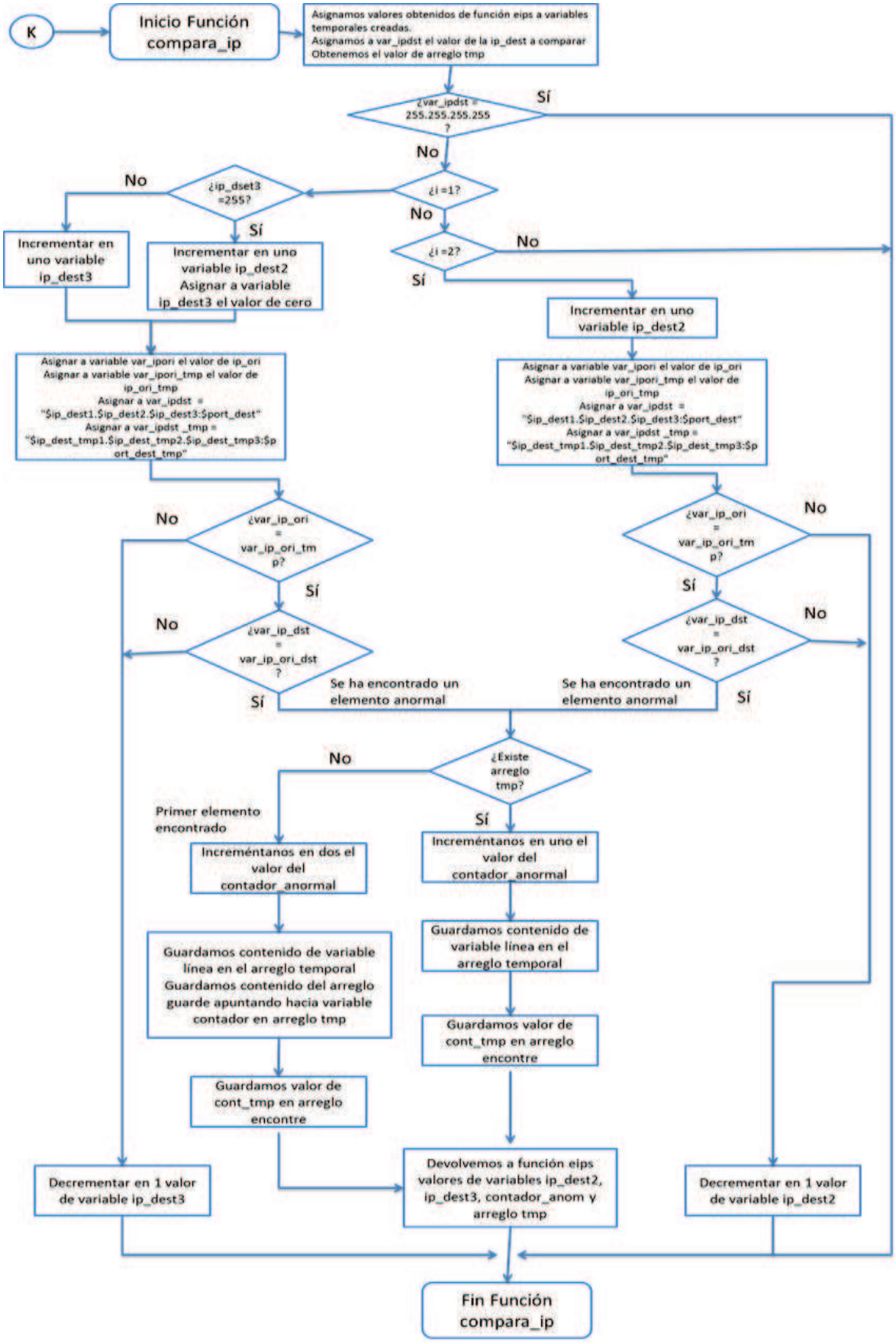
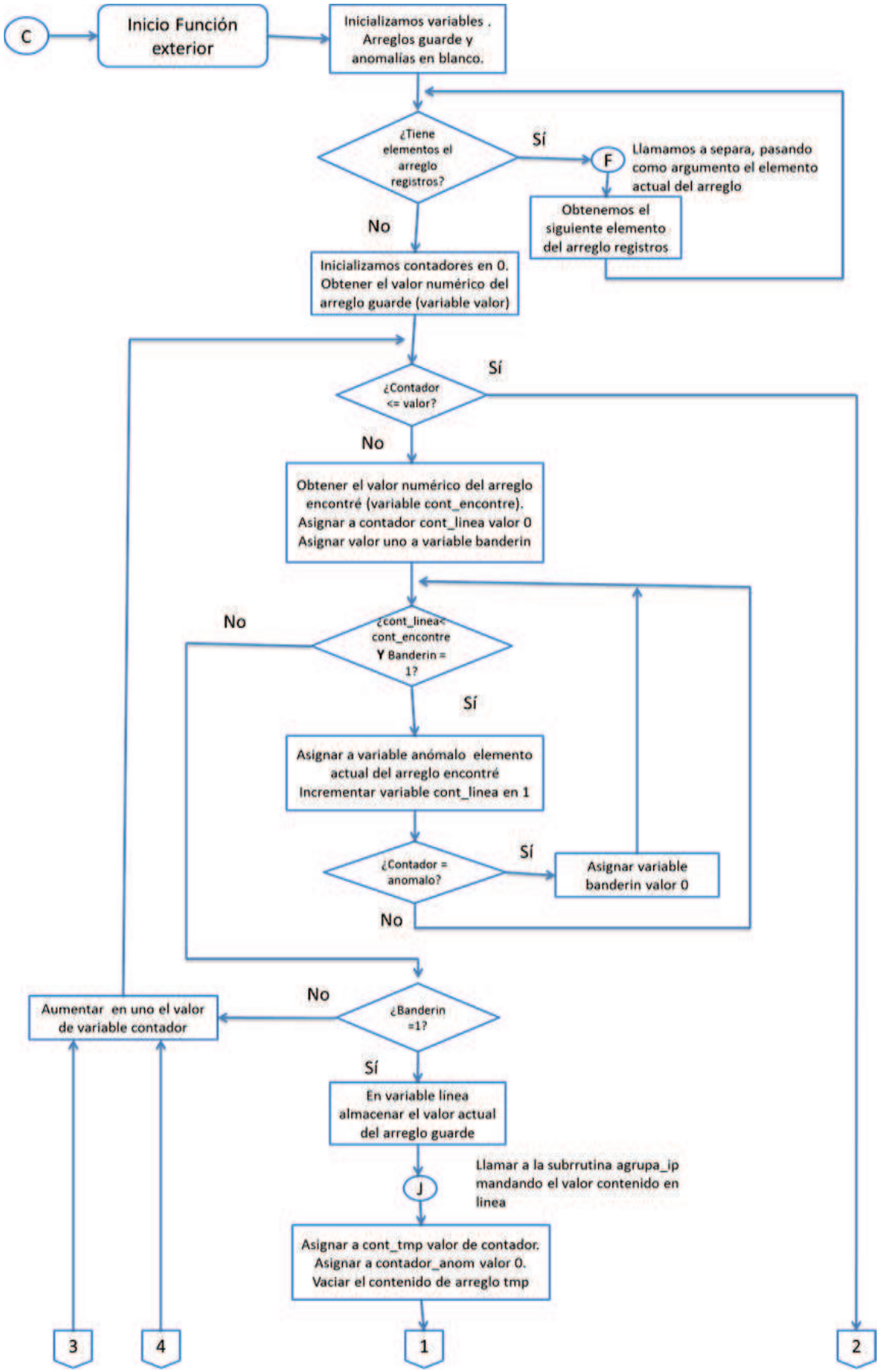


Figura 4.12 Diagrama de flujo de la función compara_ip


C
Inicio Función exterior
Inicializamos variables . Arreglos guarde y anomalías en blanco.
¿Tiene elementos el arreglo registros?
Sí
F
Llamamos a separa, pasando como argumento el elemento actual del arreglo
Obtenemos el siguiente elemento del arreglo registros
No
Inicializamos contadores en 0. Obtener el valor numérico del arreglo guarde (variable valor)
¿Contador <= valor?
Sí
No
Obtener el valor numérico del arreglo encontré (variable cont_encontre). Asignar a contador cont_linea valor 0 Asignar valor uno a variable banderin
¿cont_linea< cont_encontre Y Banderin = 1?
No
Sí
Asignar a variable anómalo elemento actual del arreglo encontré Incrementar variable cont_linea en 1
¿Contador = anomalo?
Sí
Asignar variable banderin valor 0
No
¿Banderin =1?
No
Aumentar en uno el valor de variable contador
Sí
En variable línea almacenar el valor actual del arreglo guarde
J
Llamar a la subrrutina agrupa_ip mandando el valor contenido en linea
Asignar a cont_tmp valor de contador. Asignar a contador_anom valor 0. Vaciar el contenido de arreglo tmp
3
4
1
2

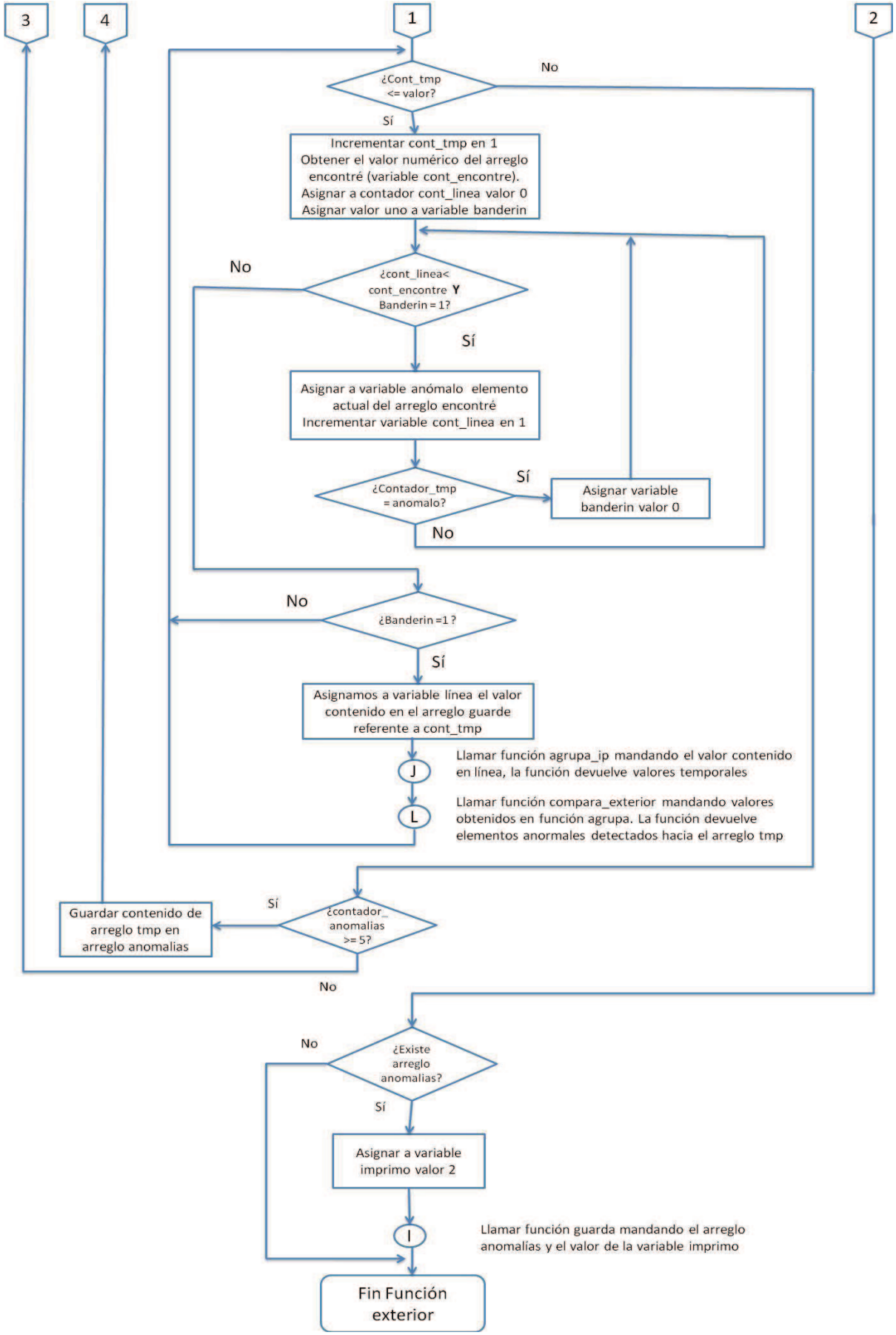


Figura 4.13 Diagrama de flujo de la función exterior

Compara_exterior: Función utilizada por la función "exterior" con el objetivo de verificar el elemento actual y el elemento temporal, pertenecientes a una red de servidores, en búsqueda de un aumento en uno del puerto origen y un envió de información mayor a 5 Mb hacia direcciones ip exteriores al rango permitido en la institución, de la siguiente forma:

"Ip_origen" pertenezca a una red de servidores **Y** *"ip_origen igual a ip_origen_tmp"* **Y** *"puerto_ori_tmp"* se haya incrementado en con respecto al valor de *"puerto_ori"* **Y** *"trafico_tmp sea mayor a 5242880"*.

En caso de detectar un elemento anormal, se guardara este elemento en el arreglo *"tmp"* y se incrementa el valor de *"contador_anomal"* en uno.

El diagrama de flujo de esta función se muestra en la figura 4.14.

Dos: Función creada con el objetivo de verificar si se ha realizado un ataque DoS o un ataque DDoS en una dirección IP perteneciente a la institución con un tráfico mayor a 50 Mb. Su funcionamiento es similar a la función epuertos, solo cambia en los siguientes puntos:

- ✓ En el paso trece se ejecuta la función compara_DoS, pasando como argumento las variables obtenidas en el paso siete y doce.
- ✓ En el paso quince se compara con un valor de variable *"contador_anom"* mayor a veinticinco.
- ✓ En el paso diecisiete se asigna a variable imprimo el valor de tres.

En la figura 4.15 se muestra el diagrama de flujo de esta función.

Compara_dos: Función utilizada por la función "DoS" con el objetivo de verificar si en el elemento actual y el elemento temporal, se ha presentado una conexión mayor a 50 Mb, de la siguiente forma:

"ip_destino igual a ip_destino_tmp" **Y** *"puerto_destino igual a puerto_destino_tmp"* **Y** *"trafico_tmp sea mayor a 52428800"*.

En caso de detectar un elemento anormal, se guarda este elemento en el arreglo *"tmp"* y se incrementa el valor de *"contador_anomal"* en uno.

El diagrama de flujo se muestra en la figura 4.16.

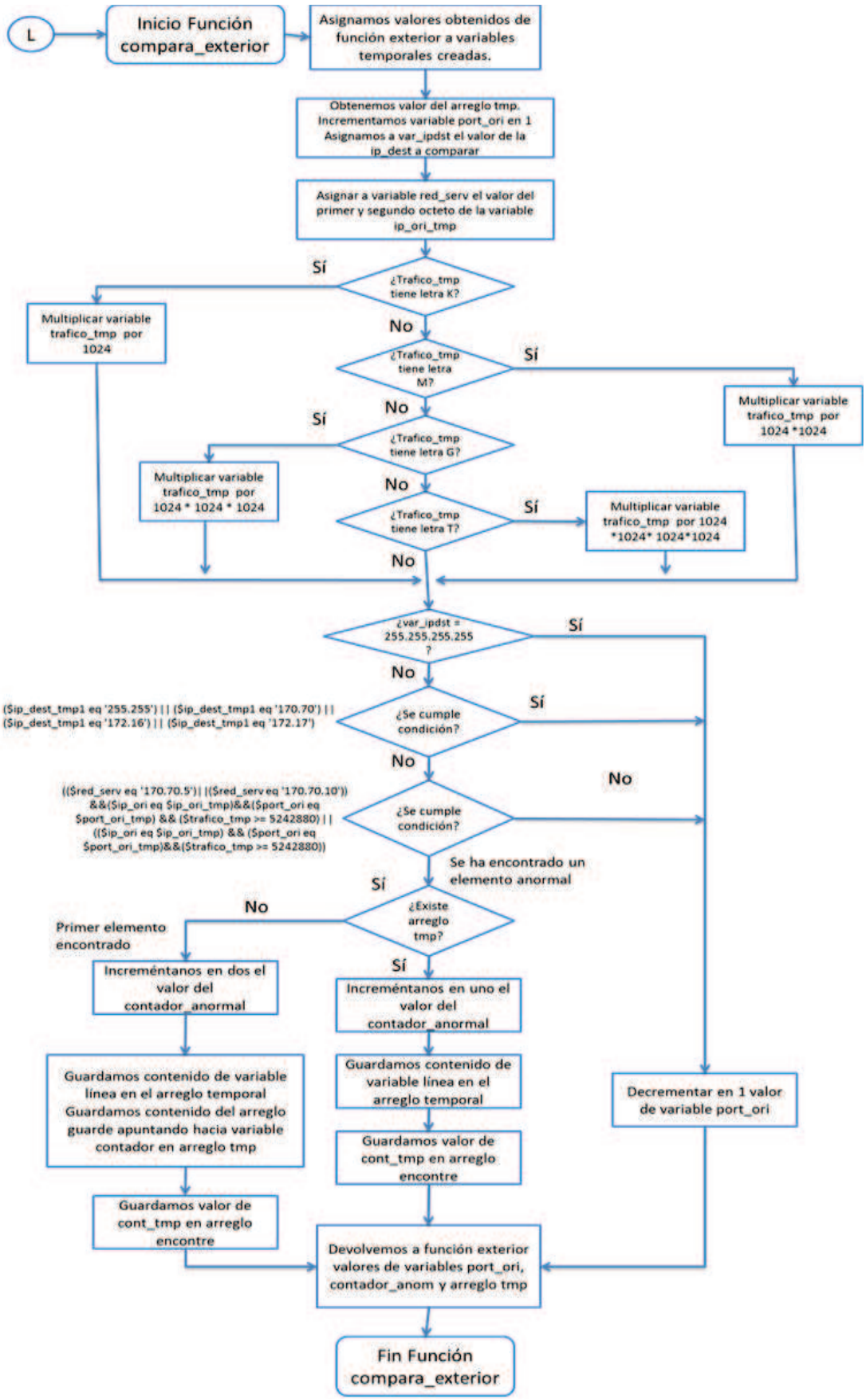
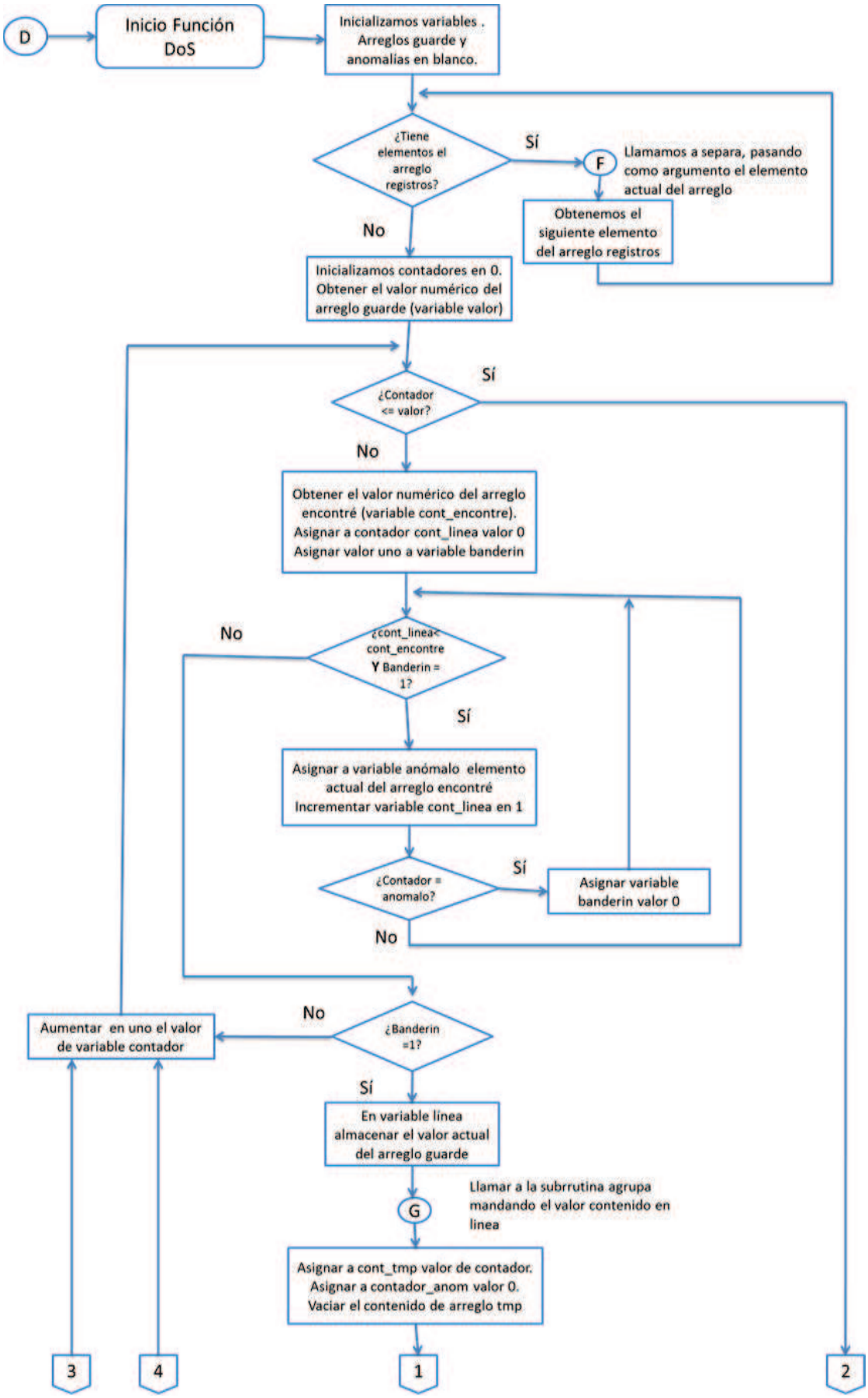


Figura 4.14 Diagrama de flujo de la función compara_exterior.


D
Inicio Función DoS
Inicializamos variables . Arreglos guarde y anomalías en blanco.
¿Tiene elementos el arreglo registros?
Sí
F
Llamamos a separa, pasando como argumento el elemento actual del arreglo
Obtenemos el siguiente elemento del arreglo registros
No
Inicializamos contadores en 0. Obtener el valor numérico del arreglo guarde (variable valor)
¿Contador <= valor?
Sí
No
Obtener el valor numérico del arreglo encontré (variable cont_encontre). Asignar a contador cont_linea valor 0 Asignar valor uno a variable banderin
¿cont_linea< cont_encontre Y Banderin = 1?
No
Sí
Asignar a variable anómalo elemento actual del arreglo encontré Incrementar variable cont_linea en 1
¿Contador = anomalo?
Sí
Asignar variable banderin valor 0
No
¿Banderin =1?
No
Aumentar en uno el valor de variable contador
Sí
En variable línea almacenar el valor actual del arreglo guarde
G
Llamar a la subrrutina agrupa mandando el valor contenido en linea
Asignar a cont_tmp valor de contador. Asignar a contador_anom valor 0. Vaciar el contenido de arreglo tmp
3
4
1
2

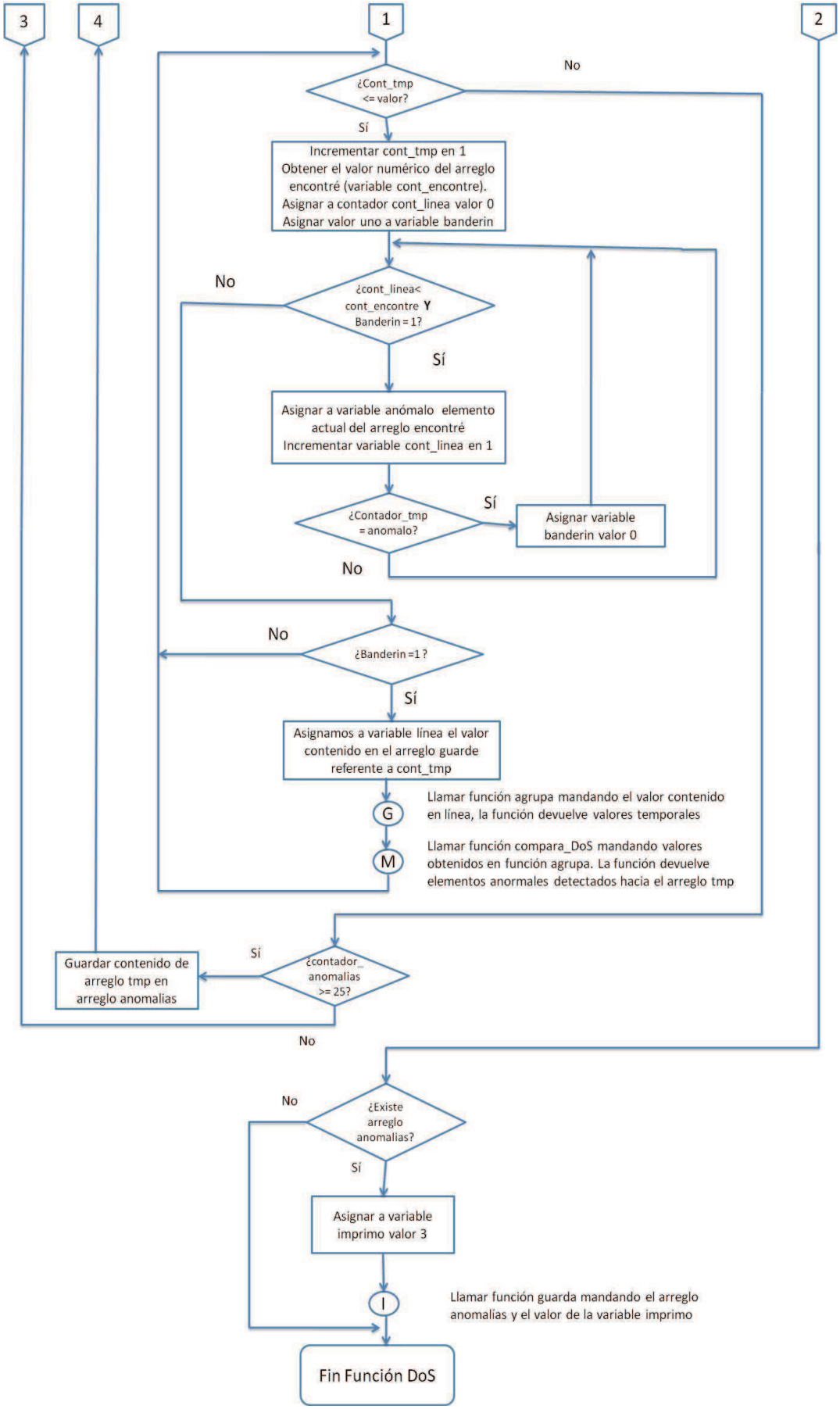


Figura 4.15 Diagrama de flujo de la función Dos.

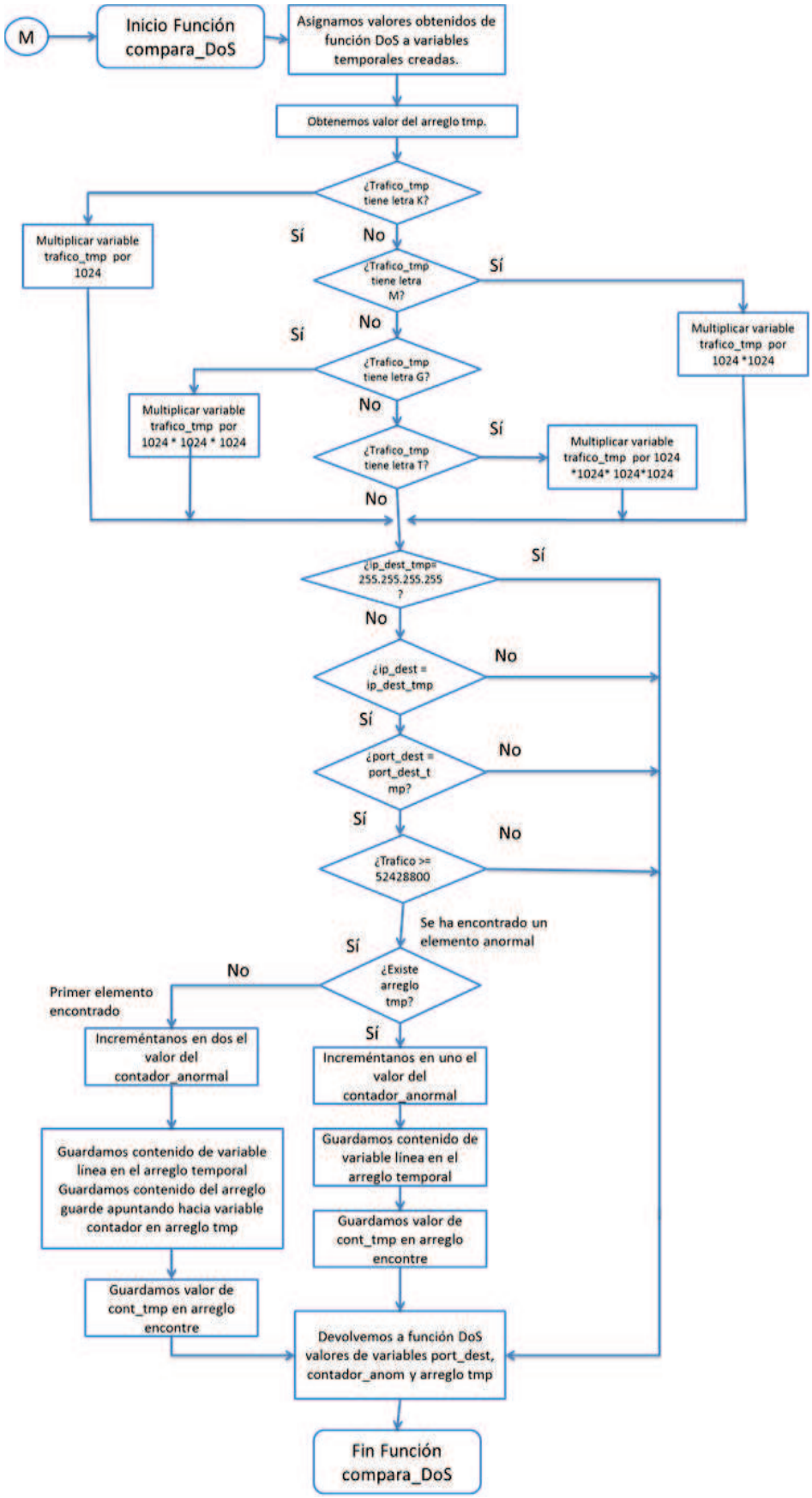


Figura 4.16 Diagrama de flujo de la función compara_dos

4.5.3.2 Funcionamiento modulo *"Escaneo.php"*.

El módulo frontend "escaneo.php" fue creado con el objetivo de visualizar los resultados obtenidos por el módulo "escaneo.pm" en la interfaz gráfica del software Nfsen. Este módulo se encarga de mostrar en pantalla el resultado del archivo "anomalías", en caso de haberse detectado una anomalía, y de mostrar el archivo nfcapd que se analizó.

El diagrama de flujo mostrado en la figura 4.2 es el utilizado por el modulo "escaneo.php". La función *"escaneo_ParseInput"* no es ocupada en este módulo frontend creado, sin embargo esta función debe de existir.

En la función "escaneo_Run" se realizan todas las acciones programadas y mostradas en el diagrama de flujo de la figura 4.2.

En el anexo D se muestra el código desarrollado en la creación del módulo frontend "escaneo.php" y el módulo backend "escaneo.pm".

En el capítulo V se mostrará el funcionamiento del software "Listry-AIGC", tanto para la función del monitoreo de red como en la detección de malware mediante el plugin creado. Este software está conformado por lo siguiente:

- Instalación y configuración del módulo HTTPS.
- Instalación y configuración del software Nfsen.
- La creación del plugin escaneo en el software Nfsen.
- Instalación y configuración del software MySQL y OpenWebmail.
- Instalación y configuración del software Navicat.