



FACULTAD DE INGENIERIA Y CIENCIAS APLICADAS

ANÁLISIS DE PROCESOS USANDO REDES COMPLEJAS.

AUTOR

PABLO ALFREDO PICO TRUJILLO

AÑO

2020



FACULTAD DE INGENIERIA Y CIENCIAS APLICADAS

ANÁLISIS DE PROCESOS USANDO REDES COMPLEJAS.

Trabajo de Titulación presentado en conformidad con los requisitos establecidos para optar por el título de Ingeniería en Sistemas de Computación e Informática

PROFESOR GUÍA:
MARIO SALVADOR GONZALES RODRÍGUEZ

AUTOR:
PABLO ALFREDO PICO TRUJILLO

AÑO

2020

DECLARACIÓN DEL PROFESOR GUÍA

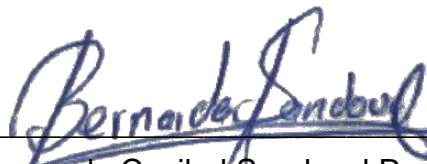
Declaro haber dirigido el trabajo, Análisis De Procesos Usando Redes Complejas, a través de reuniones periódicas con el estudiante Pablo Alfredo Pico Trujillo, en el segundo semestre del 2020, orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación.



Mario Salvador Gonzales Rodríguez
Doctor en Ingeniería informática y de telecomunicación
CI:0958376345

DECLARACIÓN DEL PROFESOR CORRECTOR

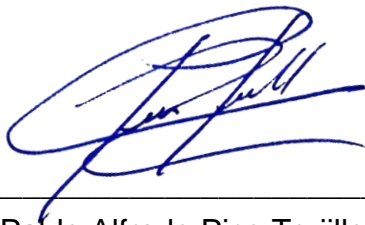
Declaro haber revisado este trabajo, Análisis De Procesos Usando Redes Complejas, del estudiante Pablo Alfredo Pico Trujillo, en el segundo semestre del 2020, dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación.



Bernarda Cecibel Sandoval Romo
Mestre en Ciencias da Computacao
CI: 1709974453

DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes.



Pablo Alfredo Pico Trujillo

CI: 1803727823

AGRADECIMIENTOS

A los docentes de la carrera de Ingeniería en Sistemas de la Universidad de las Américas, por haber compartido sus conocimientos a lo largo del periodo estudiantil, especialmente al Director Académico Marco Galarza, por su compromiso con el bienestar y la formación de los estudiantes.

Así mismo, quiero agradecer a mi tutor Mario Gonzales quien con su dirección, conocimiento y colaboración me guio a través de cada una de las etapas de este proyecto.

DEDICATORIA

Esta tesis está dedicada:

A mi querida esposa Andrea quien, con su amor y respaldo, me ayuda a alcanzar mis objetivos.

A mi hijo Pablo quien llena de alegría mi vida.

A mi abuelita Carmen, por sus valiosos consejos.

A mi madre Ligia por su amor, trabajo y sacrificio en todos estos años, gracias a ella he logrado llegar hasta aquí y convertirme en lo que soy. Ha sido un orgullo y privilegio el ser su hijo, eres la mejor madre.

RESUMEN

El análisis de procesos ha tomado una gran importancia en el ámbito empresarial. Las herramientas que facilitan la comprensión de los procesos de negocio de una organización son cada vez más importantes ya que permiten optimizar las actividades individuales, asegurando que los recursos se utilicen de manera óptima. Este trabajo busca establecer una metodología para el análisis de procesos de negocio usando teoría de redes. Para esto es necesario transformar un modelo de proceso de negocio hacia una red compleja con el fin de poder realizar análisis sobre la red. Este análisis nos permitiría obtener información más detallada sobre un proceso. Esta información nos da una idea clara de la importancia estructural y funcional de los elementos de la red. Se inicia planteando una convención que facilita la transición desde Notación de Modelado de Procesos de Negocio hacia redes complejas. Posteriormente se usó un script de Python para extraer la información de los objetos de flujo y objetos conexión, este script también se encarga de almacenar la información en diferentes formatos para una fácil reutilización de los datos en distintas herramientas. A continuación, con scripts tanto en Python como en R se procedió con la generación de las redes y su visualización. Con la información de nodos y enlaces de cada red se desarrolló una aplicación Shiny que se encarga de graficar y mostrar las medidas microscópicas (grado, coeficiente de clusterización, e intermediación) para cada uno de los nodos. Para concluir se profundiza el análisis de la red desde una perspectiva macroscópica para determinar las comunidades y su robustez. Se emplearon diferentes algoritmos de detección de comunidades y se contrastan los resultados con el fin de encontrar el algoritmo más apropiado para esta tarea. De igual manera se contrasta la vulnerabilidad de la red frente a distintos factores de error y ataque para poder determinar falencias o mejoras dentro de los procesos.

ABSTRACT

Process analysis has taken a great importance in the business field. The tools that facilitate the understanding of the business processes of an organization are increasingly important. This work aims to establish a methodology for the analysis of business processes using network theory. It begins by posing a convention that eases the transition from Business Process Modeling Notation to complex networks. Next, a Python script extracts the flow and connection objects information, this script also stores this information in different formats for easy reuse of the data in different tools. Then, with scripts in both Python and R, the networks and its visualizations are created. With the information of nodes and links from each network, a Shiny application is developed, this app oversees the analysis. This application allows obtaining microscopic measurements for each of the nodes, this information gives us a clear idea of the structural and functional importance of the elements of the network. To conclude, the analysis of the network is deepened from a macroscopic perspective to determine the communities and their robustness. Different community detection algorithms were used, and the results were compared to find the algorithm that yields better results. Likewise, the network vulnerability is contrasted against the different error and attack factors to determine deficiencies or improvements within the processes.

Índice

1. Capítulo 1. Introducción	16
1.1. Antecedentes.....	16
1.2. Alcance.	17
1.3. Justificación.	18
1.4. Objetivo General.	19
1.5. Objetivos específicos.	19
2. Capítulo 2. Marco Teórico.	20
2.1. Proceso de negocio.	20
2.2. Modelado de proceso de negocio.	20
2.3. Notación de modelado de procesos de negocio.....	20
2.4. Lenguaje de definición de procesos.	22
2.5. Grafos.....	22
2.6. Redes complejas.....	22
2.7. Análisis de la red.....	23
2.7.1. Robustez.	23
2.7.2. Detección de comunidad.....	23
2.8. Herramientas.	24
2.8.1. Software.....	24
2.8.2. Lenguajes de programación.....	25
3. Capítulo 3. Descripción Del Estudio Realizado.....	26
3.1. Definición de procesos.....	26
3.2. Modelos de los procesos negocio.....	28

3.2.1.	Proceso subasta de autos nuevos y usados.	29
3.2.2.	Proceso subasta de autos siniestrados.	30
3.3.	Equivalencia desde BPMN hacia redes complejas.	31
3.4.	Extracción de información.	32
3.5.	Creación de la red.	34
3.5.1.	Creación de la red y visualización usando Python.	34
3.5.2.	Creación de la red y visualización usando R.	39
3.6.	Visualización de las redes.	41
3.6.1.	Visualización de las redes generadas con Python.	41
3.6.2.	Visualización de redes de procesos por carriles generadas con python.	46
3.6.3.	Visualización de redes de procesos generadas con R.	49
3.7.	Análisis de la red.	53
3.7.1.	Medidas de los nodos de los procesos interconectados.	54
3.7.2.	Medidas para determinar la robustez.	62
3.7.3.	Detección de comunidades.	70
4.	Capítulo 4. Resultados.	77
4.1.	Resultados de la detección de comunidades.	77
4.2.	Resultados del análisis de robustez.	78
4.2.1.	Proceso subasta autos nuevos y usados.	78
4.2.2.	Proceso subasta de autos siniestrados.	79
5.	Capítulo 5. Conclusiones y Recomendaciones	80
5.1.	Conclusiones.	80
5.2.	Recomendaciones	81
6.	Referencias	82

7. Anexos	84
------------------------	-----------

Índice de Figuras.

Figura 1 Simbología del estándar BPMN	21
Figura 2 Formula de cercanía.	23
Figura 3 Procesos de la empresa Leilocar para subasta para autos nuevos o usados.	27
Figura 4 Procesos de la empresa Leilocar para subasta de autos siniestrados....	27
Figura 5 Proceso subasta de autos nuevos y usados (procesos A, B, I, F, C, D) .	29
Figura 6 Proceso subasta autos siniestrados (procesos E, F, G, H).	30
Figura 7 Exportar proceso en formato XPDL desde Bizagi	32
Figura 8 Ejemplo de la actividad envió de vehículo del proceso A en formato XPDL.	33
Figura 9 Pseudocódigo para la generación de redes complejas desde procesos de negocio en formato XPDL.	33
Figura 10 Ejemplo de archivos consolidados de nodos y enlaces de la red del proceso A.	37
Figura 11 Ejemplo de archivos de imagen de la red del proceso A.....	39
Figura 12 Red del proceso subasta de autos nuevos y usados.	42
Figura 13 Red del proceso subasta de autos nuevos y usados (Circular).	43
Figura 14 Red del proceso subasta de autos siniestrados.	44
Figura 15 Red del proceso subasta de autos siniestrados (Circular).	45
Figura 16 Red del proceso subasta de autos nuevos y usados (carril).	47
Figura 17 Red del proceso subasta autos siniestrados (carril).....	48
Figura 18 Red del proceso subasta autos nuevos y usados.	49
Figura 19 Red del proceso subasta de autos nuevos y usados (Circular)	50
Figura 20 Red del proceso subasta autos siniestrados.	51
Figura 21 Red del proceso subasta autos siniestrados (Circular).	52
Figura 22 Fórmula coeficiente de clusterización.	53
Figura 23 Fórmula coeficiente de intermediación.....	53
Figura 24 Detección de comunidad del proceso subasta de autos nuevos y usados con algoritmo Fast Greedy	71
Figura 25 Detección de comunidad del proceso subasta de autos siniestrados con algoritmo Fast Greedy.....	72
Figura 26 Detección de comunidad del proceso subasta de autos nuevos y usados con algoritmo Infomap (red dirigida).....	73

Figura 27 Detección de comunidad del proceso subasta de autos nuevos y usados con algoritmo Infomap (red no dirigida).....	74
Figura 28 Detección de comunidad del proceso subasta de autos siniestrados con algoritmo Infomap (red dirigida).....	75
Figura 29 Detección de comunidad del proceso subasta de autos siniestrados con algoritmo Infomap (red no dirigida).....	76
Figura 30 Pérdida de conectividad del proceso subasta de autos nuevos y usados frente a diferentes patrones de ataque.....	78
Figura 31 Pérdida de conectividad del proceso subasta de autos siniestrados frente a diferentes patrones de ataque.....	79

Índice de Tablas.

Tabla 1 Equivalencia entre elementos de BPMN y redes complejas.	31
Tabla 2 Medidas de los nodos del proceso subasta autos nuevos y usados.	54
Tabla 3 Distribución de enlaces en base al grado de los nodos del proceso subasta autos nuevos y usados.	59
Tabla 4 Medidas de los nodos del proceso subasta de autos siniestrados.	59
Tabla 5 Distribución de enlaces en base al grado de los nodos del proceso subasta autos siniestrados.	62
Tabla 6 Medidas de robustez del proceso subasta de autos nuevos y usados.	63
Tabla 7 Medidas de robustez del proceso subasta de autos siniestrados.	67
Tabla 8 Comunidades para los procesos interconectados.	76

Capítulo 1. Introducción

1.1. Antecedentes.

El Modelado de Procesos de Negocio (BPM por sus siglas en inglés) es un enfoque sistemático para identificar, levantar, documentar, diseñar, ejecutar, medir y controlar tanto los procesos manuales como automatizados, con la finalidad de lograr a través de sus resultados en forma consistente los objetivos de negocio que se encuentran alineados con la estrategia de la organización. BPM abarca el apoyo creciente de la Tecnología de la Información (TI por sus siglas en inglés) con el objetivo de mejorar, innovar y gestionar los procesos de principio a fin, que determinan los resultados de negocio, crean valor para el cliente y posibilitan el logro de los objetivos de negocio con mayor agilidad. (ABPMP -- Association of Business Process Management Professionals, 2013)

El modelado y Notación de Procesos de Negocio (BPMN por sus siglas en inglés), contiene una serie de objetos representados por símbolos. A la definición gráfica de la simbología y su uso por medio de reglas bien definidas se le llama también reglas sintácticas. Al significado de los símbolos y de los patrones que con ellos se pueden modelar se le llama semántica. (Freund, Rücker, & Hitpass, 2014)

La modelación de redes permite la resolución de múltiples problemas de programación matemática mediante la implementación de algoritmos especiales creados para tal fin, conocidos como Algoritmos de optimización de redes. Dentro de los problemas más comúnmente resueltos mediante la modelación de redes se encuentran problemas de modelos de transporte, transbordo además de los muy conocidos modelos de determinación de cronograma de actividades para proyectos como lo son el PERT y el CPM. (Salazar López, 2019).

La teoría de redes tiene aplicaciones en muchas disciplinas, incluyendo la física estadística, la física de partículas, la informática, la ingeniería eléctrica, la biología, la economía, las finanzas, la investigación de operaciones, la climatología y la sociología. Aplicaciones de la teoría de redes incluyen la Word Wide Web, Internet, redes de regulación genética, las redes metabólicas, redes sociales, epistemológicas redes, etc. (Network theory, 2019)

En la publicación *“Analyzing Process Models Using Graph Reduction Techniques”* se muestra cómo un proceso de negocio puede contener puntos muertos y problemas de sincronización que podrían comprometer la correcta ejecución de los flujos de trabajo. Determinar estos conflictos es una tarea compleja y requiere el desarrollo de algoritmos específicos que proporcionen información sobre la complejidad del proceso. En este caso, el algoritmo de reducción puede usarse para

contar posibles subredes de instancias de un modelo de proceso correcto. La contribución específica del artículo es una nueva técnica para satisfacer criterios de corrección en los modelos de proceso usando técnicas de reducción de redes. (Sadiq & Orłowska)

Basado en la explicación anterior, la presente línea de investigación pretenderá determinar los beneficios de representar un proceso de negocio como una red compleja. Mostrar como las métricas de la red y los posibles análisis que se pudiesen ejecutar sobre la misma dan una perspectiva diferente sobre el proceso, Por ejemplo, en análisis de la robustez de la red nos permite determinar la habilidad que tiene el proceso de resistir fallas o errores.

1.2. Alcance.

El proyecto desarrollado tiene como alcance realizar un estudio de los procesos de negocio como redes complejas. Con la finalidad de demostrar que al modelar un proceso de negocio como una red compleja se obtienen métricas adicionales, que permiten análisis más especializados como por ejemplo la detección de comunidad y la robustez. Para ello se tomó como base los datos provenientes de la tesis doctoral: Identificación y Modelado de Procesos en Empresa Leilocar, (Pantoja, 2012).

De los datos enmarcados en la tesis doctoral antes mencionada se ha obtenido los siguientes procesos de negocios que se modelarán como redes complejas, utilizando diferentes herramientas, para poder profundizar en su análisis:

- Recepción de automóviles, entrada de sistema y creación de subastas
- Ejecución de la subasta física.
- Negociación de ventas provisionales.
- Recogida de vehículos (subastas físicas y online).
- Creación y preparación de subasta de reclamos online.
- Participación en la subasta online.
- Análisis de subasta de reclamaciones en línea.
- Colección de vehículos subastados online.
- Creación de subastas en línea.
- Creación de la subasta de distribuidores en línea.
- Promociones del Centro de Clientes.

Los aspectos puntuales del estudio consisten en analizar los modelos generados para obtener criterios de complejidad y determinar posibles generalizaciones para cada flujo de proceso. Estas generalizaciones permiten determinar la conectividad

de las actividades, el peso de los carriles, el peso de los enlaces, su robustez del proceso, sus comunidades, etc.

1.3. Justificación.

Las organizaciones con el paso del tiempo y su expansión en el mercado se han visto en la necesidad de implementar nuevos procesos que permitan satisfacer no solo a sus clientes sino también a sus proveedores, empleados, accionistas entre otros que conforman la institución. Éstos nuevos procesos y actividades aplicadas deben permitir la optimización de recursos como son: recursos financieros, monetarios, tecnológicos de personal y demás que mantenga a flote a las empresas, es decir, los procesos deben estar correctamente entrelazados permitiendo así un flujo eficaz entre ellos. En este contexto, muchas de las organizaciones han implementado nuevos procesos en sus diferentes departamentos, pero sin un estudio previo el cual le permita conocer si dicha implementación causara pérdidas innecesarias o ganancias a futuro por optimización de recursos, es decir, de una forma empírica.

El modelado y notación de procesos de negocio son herramientas que tiene la capacidad de describir procedimientos comerciales bajo una representación gráfica fácil de entender. Este estándar de modelado permite simplificar las relaciones, colaboraciones, transiciones y transacciones entre diferentes áreas y procedimientos. BPMN adopta varias notaciones y elementos de otras herramientas de modelado, por ejemplo, el lenguaje de definición de proceso XML, componentes de los diagramas de actividad del lenguaje de modelado unificado (UML), patrones de lenguaje de ejecución de procesos de negocio (BPEL) y de flujo de trabajo (Workflow). (White, XPDL and BPMN, 2020) Debido estas particularidades el mal uso de los elementos puede producir errores semánticos fácilmente, por ejemplo, el uso incorrecto de compuertas lógicas, flujos inválidos, ejecución de subprocessos, flujos de mensajes, etc. La mayoría de estos errores pueden detectarse fácilmente ya que las herramientas de diagramación tienen una función de verificación incluida. Pero a pesar de esto el análisis de estos modelos se obstaculiza por la complejidad del lenguaje, especificación del estándar por lo que el uso de redes complejas para el análisis de estos diagramas podría simplificar su análisis.

El análisis de redes complejas aplicadas a los diferentes procesos de una organización permite observar desde un punto vista general todas las conexiones que se realizan entre los diferentes departamentos y las actividades que se desarrollan en cada uno de los de ellos. Mediante la aplicación de grafos y la visualización de los nodos se permitirá identificar e inferir matemáticamente si cada

uno de los procesos es indispensable o puede ser retirado. Adicional a ellos mediante el análisis se podrán establecer niveles, es decir, si se quisiera retirar un nodo se podrá observar si este es o no importante, pudiendo fortalecerlo o tomar medidas de contingencia y verificando que este no comprometa a otros nodos.

El análisis de redes complejas también proporciona información sobre las causas del cambio en el comportamiento de un sistema, organización o proceso, es decir, al realizar un análisis de diferentes procesos de negocio buscamos:

- Profundizar en el funcionamiento interno de la red (proceso), exponiendo las razones subyacentes de los comportamientos específicos de la red que de otra manera no se podrían aislar.
- Identificar patrones y comprender cómo influir en ellos.

Además de algunas posibles aplicaciones como:

- Comprender cómo se comporta un proceso específico.
- Comprender cómo responderá al comportamiento cambiante de los miembros y actividades en un proceso.
- Identificar las mejores condiciones en un sistema de procesos para modificarlos sin comprometer el flujo o la operación diaria.

1.4. Objetivo General.

Realizar un estudio para determinar la relevancia del uso de redes complejas para analizar procesos de negocio modelados con BPMN, señalando los posibles puntos críticos para cada proceso y recomendaciones. De esta manera obtener un conjunto de medidas de complejidad en un proceso de modelado, que permite un análisis rápido y fácil de la complejidad de diferentes procesos.

1.5. Objetivos específicos.

- Realizar el modelado de procesos usando Bizagi o cualquier otra herramienta de diagramado.
- Traducir los modelos estableciendo una convención para convertir el diagrama de flujo en BPMN a una red compleja.
- Representar las redes utilizando herramientas de visualización de redes (R, Python, Gephi).
- Analizar estructural y funcionalmente las redes (métricas).
- Interpretar las métricas de los análisis de los flujos de proceso.

Capítulo 2. Marco Teórico.

2.1. Proceso de negocio.

Un proceso describe una secuencia o flujo de actividades en una organización con el objetivo de llevar a cabo el trabajo. Los procesos se pueden definir en cualquier nivel desde Procesos de toda la empresa a Procesos realizados por una sola persona. Los procesos de bajo nivel se pueden agrupar juntos para lograr un objetivo comercial común. (ABPMP -- Association of Business Process Management Professionals, 2013)

2.2. Modelado de proceso de negocio.

El modelado de procesos comerciales es la representación gráfica de los procesos comerciales o flujos de trabajo de una empresa. Esto generalmente se realiza a través de diferentes métodos de gráficos, como el diagrama de flujo, el diagrama de flujo de datos, etc. El modelado de procesos de negocios generalmente se usa indistintamente con el mapeo de procesos de negocios, y pueden ser más o menos lo mismo, ambos se utilizan para representar gráficamente procesos como un medio para identificar posibles debilidades, mejoras o fortalezas. (ABPMP -- Association of Business Process Management Professionals, 2013)

2.3. Notación de modelado de procesos de negocio.

Un proceso describe una secuencia de actividades en una organización con el objetivo de llevar a cabo el trabajo en particular. BPMN define un proceso como una serie de actividades conectadas por flujo de secuencia. En BPMN un proceso se representa como un gráfico de elementos de flujo, que son un conjunto de actividades, eventos, puertas de enlace y Flujos de secuencia que definen la semántica de ejecución finita. (ABPMP -- Association of Business Process Management Professionals, 2013) La dirección de las puntas de flecha del flujo de secuencia determina el orden de la secuencia. BPMN proporciona a las empresas la capacidad de comprender sus procedimientos comerciales internos en una notación gráfica y les dará a las organizaciones la capacidad de comunicar estos procedimientos de manera estándar. En BPMN un proceso se representa como un gráfico de las actividades que pueden ser descritas en cualquier nivel, desde los procesos de toda la empresa hasta los procesos pequeños realizados por una sola persona. (White, PROCESS MODELING NOTATIONS AND WORKFLOW

PATTERNS, 2020) En la notación de modelado de procesos de negocio podemos encontrar objetos de flujo y objetos de conexión. Los objetos de flujo son: eventos, actividades, artefactos, etc. Los objetos de conexión son flujos de secuencia y asociaciones. En la Figura 1 podemos observar al descripción y representación gráfica de los objetos más comunes en BPMN.













Elemento	Variante	Descripción	Estándar BPMN
Evento	Inicio	Indica donde empieza un proceso.	
	Intermedio	Ocurren entre el inicio y fin de un proceso.	
	Fin	Indica donde culmina un proceso.	
Actividades	Tarea	Representa un trabajo específico dentro de una organización.	
	Subproceso	Resume un conjunto de tareas que se representa en otro diagrama.	
Decisión	Exclusivo	La decisión puede tomar una sola vía.	
	Inclusivo	La decisión puede tomar más de una vía.	
	Paralelo	Varias actividades pueden realizarse paralelamente.	
Artefacto	Data Store	Provee a las actividades un mecanismo para obtener o actualizar información que persiste en el proceso.	
	Data Object	Provee información sobre como los documentos, datos y otros objetos son usados.	
Conexión	Flujo de secuencia	Permite conectar el resto de los elementos en un diagrama de procesos de negocio.	
	Asociación	Permite conectar artefactos con objetos de conexión.	

Figura 1 Simbología del estándar BPMN

Adaptado de (ABPMP -- Association of Business Process Management Professionals, 2013)

2.4. Lenguaje de definición de procesos.

El lenguaje de definición de procesos XML (XPDL) es un formato estandarizado por la Workflow Management Coalition para que las definiciones de procesos de negocios se cambien entre diferentes herramientas de flujo de trabajo, suites de administración y herramientas de modelado. XPDL define esquemas XML en los que especifica partes del proceso empresarial. Con XPDL se puede intercambiar la semántica y los gráficos del proceso comercial. Actualmente, XPDL es el mejor formato de archivo de intercambio de diagramas BPMN y se ha creado específicamente para el almacenamiento de todo lo relacionado con el diagrama BPMN (Workflow Management Coalition, 2020). El objetivo principal de XPDL es el almacenamiento e intercambio del diagrama del proceso.

2.5. Grafos.

Un grafo es una estructura que representa un conjunto de objetos relacionados. Los objetos corresponden a abstracciones matemáticas llamadas nodos (vértices o puntos) y cada uno de los nodos está relacionados por enlaces (bordes o línea). En su forma más básica un grafo se representa en forma de diagrama como un conjunto de puntos o círculos para los vértices, unidos por líneas o curvas para los bordes. (Network theory, 2019)

2.6. Redes complejas.

Una red compleja es un Red que describe las interacciones presentes en un sistema complejo. Por ejemplo, el internet o los procesos de una empresa, estos sistemas tienen una estructura en forma de red subyacente que describe un complejo conjunto de interacciones. (Freeman, 1978)

2.7. Análisis de la red.

Aunque existen varios análisis que se pueden realizar a las redes complejas. En el estudio únicamente no enfocaremos en la robustez y la detección de comunidades.

2.7.1. Robustez.

La robustez está determinada por la pérdida de conectividad de una red. Esta conectividad es también llamada cercanía. La cercanía podría definirse como el inverso de la lejanía, que a su vez vendría a ser la suma de distancias a todos los demás nodos. Sin embargo, como la distancia entre los nodos en los componentes desconectados de una red es infinita, esta medida no puede aplicarse a las redes con componentes desconectados. Por lo tanto, se define la ecuación de cercanía como la suma de distancias inversas a todos los demás nodos (en lugar de la inversa de la suma de distancias a todos los demás nodos) tal como se puede observar en la Figura 2.

$$cercania(i) = \sum_j \frac{1}{d_{ij}}$$

Figura 2 Formula de cercanía.

Tomado de (Freeman, 1978)

2.7.2. Detección de comunidad.

También conocido como clusterización, la detección de comunidades en redes juega un papel importante en el área de investigación de reconocimiento de patrones ya que permite la identificación de grupos de nodos altamente relacionados en una red. (Pereira, Maldonado, & Nascimento, 2016)

2.7.2.1. Algoritmo Fast Greedy

Este algoritmo utiliza una optimización codiciosa en la cual, comenzando con cada vértice como el único miembro de una comunidad de uno, repetidamente une a las dos comunidades cuya fusión produce el mayor incremento de modularidad. Para una red de n nodos, después de $n - 1$ uniones, nos queda una sola comunidad y el

algoritmo se detiene. (Newman & Girvan, Finding and evaluating community structure in networks., 2004)

2.7.2.2. Algoritmo Infomap

El algoritmo Infomap caracteriza el problema de encontrar la agrupación óptima de una red como el problema de encontrar una descripción de información mínima de una caminata aleatoria. El algoritmo maximiza una función objetivo llamada Longitud Mínima de Descripción. Estudios anteriores han encontrado que el rendimiento de Infomap se mantiene estable para redes con hasta 100,000 nodos. (Emmons, Kobourov, Gallant, & Börner, 2020) Este algoritmo puede aplicarse a redes dirigidas y no dirigidas.

2.8. Herramientas.

Para la realización de la investigación usaremos como material de entrada procesos de negocios, los mismos que usando herramientas de software serán transformados desde BPMN a grafos.

2.8.1. Software.

Para el modelado de procesos en BPMN usaremos la herramienta Bizagi Modeler. Para la extracción de la información de los procesos crearemos un script en Python. Finalmente, para la creación y análisis de la red usaremos ambos lenguajes Python y R. A continuación, una breve descripción de las herramientas que usaremos:

2.8.1.1. Bizagi Modeler.

Bizagi Modeler es una aplicación para diagramar, documentar y simular procesos en un formato estándar conocido como Business Process Model and Notation (BPMN). Los procesos pueden publicarse en diferentes formatos (Word, PDF, SharePoint, o exportarse a Visio, png, bpm, svg o jpg y XPD, para ser compartidos y comunicados en toda la organización. (Bizagi, 2020)

2.8.2. Lenguajes de programación.

Para el estudio se usará los lenguajes Python y R debido a la popularidad que tiene para el análisis de datos y la gran cantidad de librerías para la creación y análisis de redes que disponen.

2.8.2.1. Python.

Python es un lenguaje de programación interpretado, orientado a objetos y de alto nivel con semántica dinámica. Sus estructuras de datos integradas de alto nivel, combinadas con tipeo y enlace dinámicos, lo hacen muy atractivo para los desarrolladores ya que permite una rápida creación de aplicaciones. La sintaxis simple y fácil de aprender de Python enfatiza la legibilidad y, por lo tanto, reduce el costo del mantenimiento del programa. Python admite módulos y paquetes, lo que impulsa la modularidad y la reutilización de código. El intérprete de Python y la extensa biblioteca estándar están disponibles en formato fuente o binario sin cargo para todas las plataformas principales, y se pueden distribuir libremente. (PSF, 2020)

Librerías usadas:

- Beautiful Soup 4
- Numpy
- Pandas
- Networkx
- Matplotlib

2.8.2.2. R.

R es un lenguaje y entorno para computación estadística y gráficos que proporciona una amplia variedad de técnicas estadísticas (modelos lineales y no lineales, pruebas estadísticas clásicas, análisis de series de tiempo, clasificación, agrupamiento, etc.). (R Foundation, 2020)

Librerías usadas:

- Igraph
- visNetwork
- Shiny
- NetSwan

Capítulo 3. Descripción Del Estudio Realizado.

Para la investigación es necesario definir los procesos que se estudiarán. Posteriormente se establecerá una equivalencia entre los elementos de BPMN y su representación en una red compleja. Más tarde, se procederá con la extracción de la información de los procesos. Con esta información se procederá a la creación y visualización de redes en dos de los lenguajes más populares Python y R. En Python usaremos Networkx en conjunto con Matplotlib. En R usaremos igraph en conjunto con VisNetwork y Shiny. Una vez generada la red, se procederá a obtener las métricas para los nodos individuales y la red como conjunto. Con estas métricas realizaremos un análisis de las comunidades, tolerancia a errores o ataques y el impacto cuando un nodo es eliminado.

3.1. Definición de procesos.

La investigación hace referencia al estudio de procesos de negocio, por lo que es crucial contar con modelos que hayan sido ampliamente estudiados y validados. Los modelos de proceso de negocio que usaremos se basaron en los procesos de la empresa brasileña Leilocar y han servido para una tesis doctoral denominada "*Identificación y modelado de procesos en la empresa Leilocar*" (Pantoja, 2012).

La empresa separa los flujos en base a la procedencia de los autos ya sean nuevos, usados o siniestrados.

En la Figura 3 se muestra el flujo de los procesos que están relacionados en la venta de autos nuevos provenientes de distribuidores y autos usados provenientes de personas naturales, los procesos que intervienen son:

- Proceso A. Recepción de vehículos, la entrada del sistema y la creación de la subasta.
- Proceso B. Implementación de la subasta física.
- Proceso C. Negociación de ventas provisionales.
- Proceso D. Colección de vehículos (físicas y subastas en línea).
- Proceso F. La participación en la subasta en línea.
- Proceso I. Creación de subastas en línea concesionarios de la marca.
- Proceso J. Creación de subastas en línea distribuidores

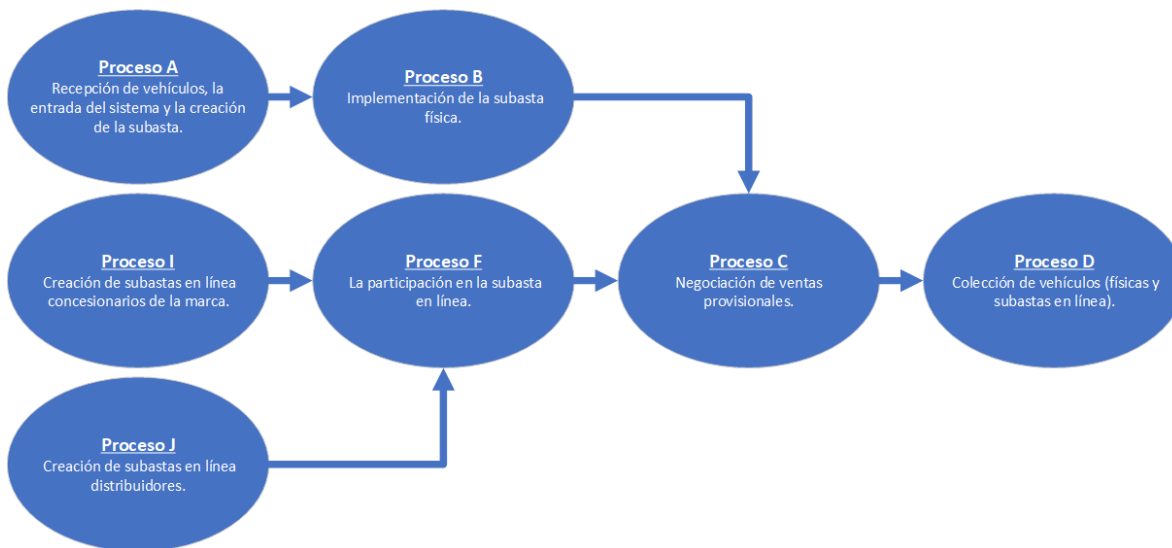


Figura 3 Procesos de la empresa Leilocar para subasta para autos nuevos o usados.

Adaptado de (Pantoja, 2012)

En la Figura 4 se muestra el flujo de los procesos que están relacionados en la venta de autos siniestrados provistos por aseguradoras, los procesos que intervienen son:

- Proceso E. Creación y preparación de subastas en línea.
- Proceso F. La participación en la subasta en línea.
- Proceso G. Análisis de subastas en línea.
- Proceso H. Recolección de vehículos siniestrados para subastas online.

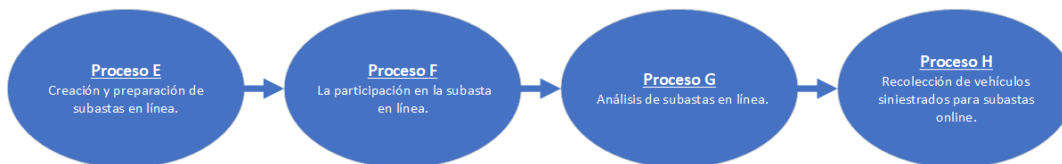


Figura 4 Procesos de la empresa Leilocar para subasta de autos siniestrados.

Adaptado de (Pantoja, 2012)

3.2. Modelos de los procesos de negocio.

El modelamiento de los procesos se realizó en la herramienta Bizagi, se complementaron los procesos base añadiendo artefactos y sus respectivas asociaciones. Además, se detalló el tipo para cada una de las actividades y eventos. Los modelos de los procesos individuales se encuentran en los anexos.

- Proceso A. Recepción de vehículos, la entrada del sistema y la creación de la subasta. **Anexo 15**
- Proceso B. Implementación de la subasta física. **Anexo 16**
- Proceso C. Negociación de ventas provisionales. **Anexo 17**
- Proceso D. Colección de vehículos (físicas y subastas en línea). **Anexo 18**
- Proceso F. La participación en la subasta en línea. **Anexo 19**
- Proceso I. Creación de subastas en línea concesionarios de la marca. **Anexo 20**
- Proceso J. Creación de subastas en línea distribuidores. **Anexo 21**
- Proceso E. Creación y preparación de subastas en línea. **Anexo 22**
- Proceso F. La participación en la subasta en línea. **Anexo 23**
- Proceso G. Análisis de subastas en línea. **Anexo 24**
- Proceso H. Recolección de vehículos siniestrados para subastas online.

Por último, se generaron dos modelos que combinan los procesos en base a su relación dentro de la empresa (autos nuevos, usados o siniestrados). Los procesos integrados son:

- El proceso subasta de autos nuevos y usados. (Figura 5)
- El proceso subasta de autos siniestrados. (Figura 6)

3.2.1. Proceso subasta de autos nuevos y usados.

El proceso subasta de autos nuevos y usados combina los siete procesos: proceso A, proceso I, proceso J, proceso B, proceso F, proceso C y proceso D que están relacionados en base a las actividades de subasta de autos nuevos provenientes de distribuidores y de autos usados provenientes de personas naturales. Este proceso interconectado se puede observar en la Figura 5.

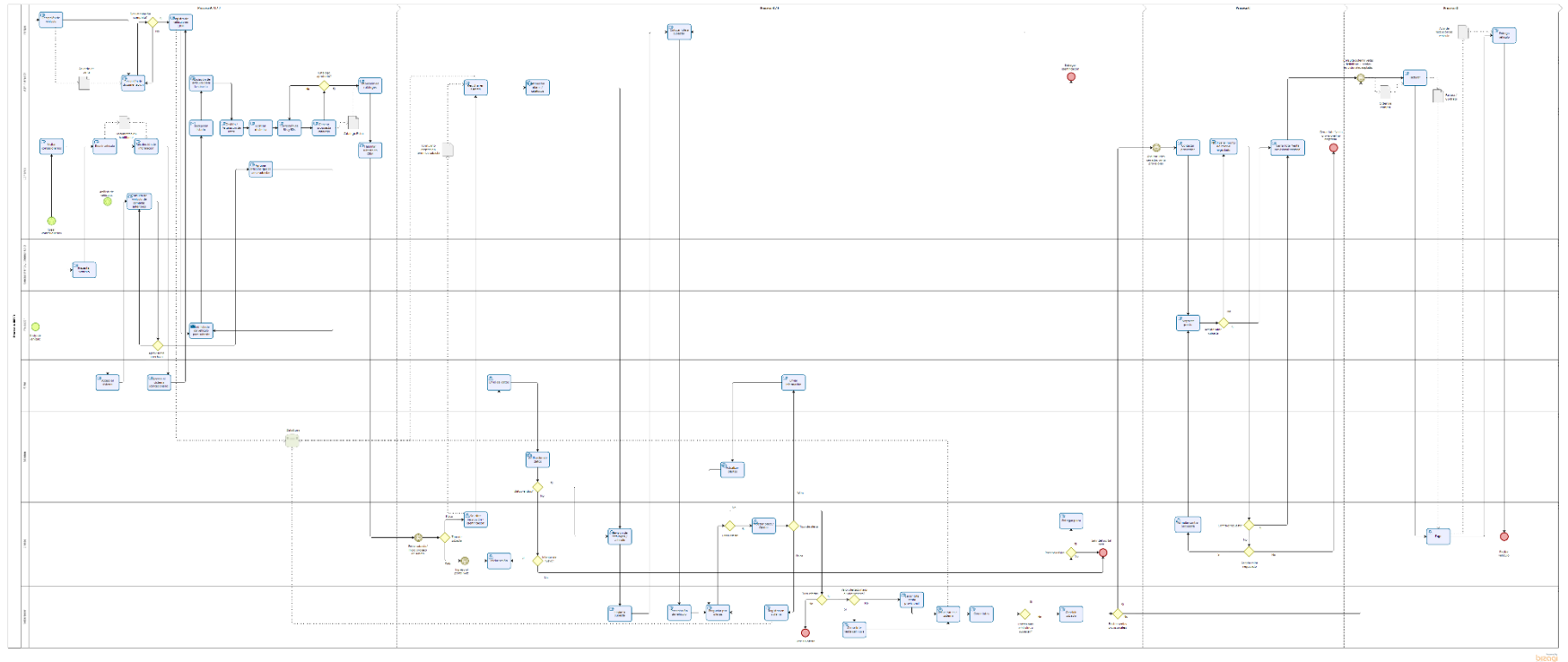


Figura 5 Proceso subasta de autos nuevos y usados (procesos A, B, I, F, C, D)

3.2.2. Proceso subasta de autos siniestrados.

El proceso subasta de autos siniestrados combina los cuatro procesos: proceso E, proceso F, proceso G y proceso H. que están relacionados en base a las actividades de subasta de autos siniestrados que provienen de aseguradoras.

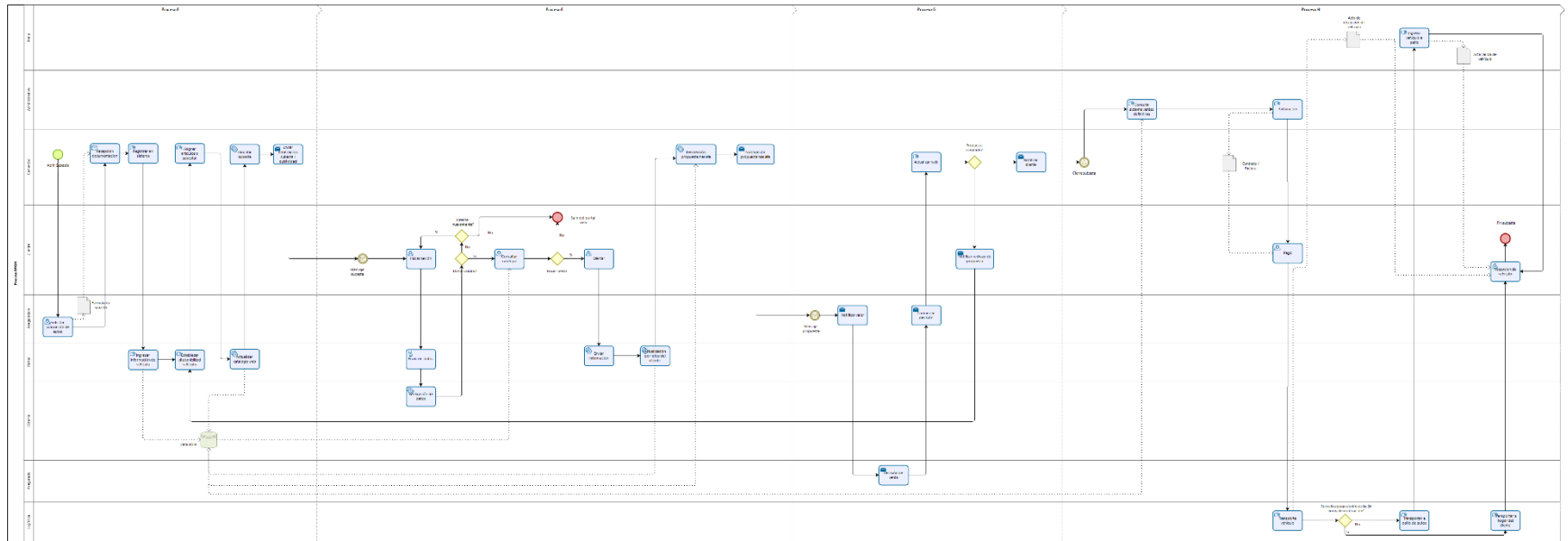







Figura 6 Proceso subasta autos siniestrados (procesos E, F, G, H).

3.3. Equivalencia desde BPMN hacia redes complejas.

De una manera básica todos los objetos de flujo y artefactos se representarán como nodos, mientras que los objetos de asociación se representarán como enlaces. Para facilitar la identificación de los tipos nodos se asoció una figura a los diferentes objetos de flujo y artefactos como se puede ver en la Tabla 1. La figura para cada nodo podrá variar dependiendo de la herramienta que se utilice para la representación visual de la red.

Tabla 1 Equivalencia entre elementos de BPMN y redes complejas.

BPMN		Redes Complejas	
Objetos de flujo y artefactos	Nodos Matplotlib (Python)	Nodos visNetwork (R)	
Eventos	○ Event		Event
Actividades	□ Activity		Activity
Puertas de enlace	△ Gateway		Gateway
Objeto de datos	◇ DataObject		DataObject
Almacén de datos	○ DataStore		DataStore
Objetos de conexión	Nodos Matplotlib (Python)	Enlaces visNetwork (R)	
Flujo de secuencia, flujo de mensajes, asociaciones	_____	_____	

3.4. Extracción de información.

Para que los scripts desarrollados en esta investigación puedan ser usados por un mayor número de interesados sin importar la herramienta de diagramado que usen, se decidió usar el formato XPDL. Este formato es usado para el intercambio de procesos de negocio entre diferentes herramientas de diagramado. XPDL admite todos los aspectos de la notación de definición de proceso BPMN, incluidas descripciones gráficas y propiedades del diagrama. (Workflow Management Coalition, 2020) (Bizagi, 2020) Al usar archivos XPDL como fuente de nuestros scripts logramos que los resultados sean replicables con procesos de negocios generados en cualquier herramienta que soporte este formato.

En el caso de la herramienta Bizagi el proceso para exportar el diagrama de proceso en formato XPDL es sencillo, basta con ir al menú de Export/Import y seleccionar XPDL tal como se muestra en la Figura 7

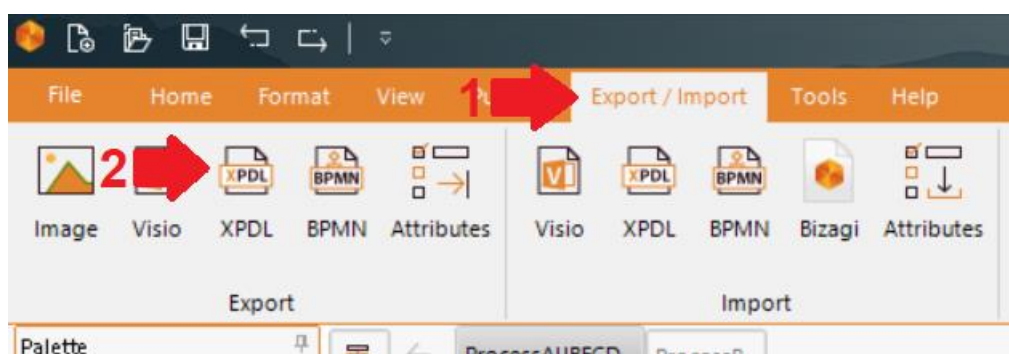


Figura 7 Exportar proceso en formato XPDL desde Bizagi

La estructura de un archivo XPDL es similar a la de un archivo XML, solo que con etiquetas específicas para la notación de BPMN. En la Figura 8 se muestra un ejemplo del archivo XPDL del proceso A, específicamente de la actividad Envió de vehículo.


```

188 <Activities>
189 <Activity Id="48486c8a-677e-4e32-a2f8-288d8010c007" Name="Envío de vehículo">
190 <Description />
191 <Event>
192 <StartEvent Trigger="None" />
193 </Event>
194 <Documentation />
195 <NodeGraphicsInfos>
196 <NodeGraphicsInfo ToolId="BizAgi_Process_Modeler" Height="30" Width="30" BorderColor="-10311914"
197 FillColor="-1638505">
198 <Coordinates XCoordinate="146" YCoordinate="764" />
199 <TextDirection xsi:nil="true" />
200 </NodeGraphicsInfo>
201 </NodeGraphicsInfos>
202 <ExtendedAttributes>
203 <ExtendedAttribute Name="RuntimeProperties" Value="{}" />
204 </ExtendedAttributes>
205 </Activity>

```

Figura 8 Ejemplo de la actividad envió de vehículo del proceso A en formato XPDL.

Una vez establecida la fuente de información de los procesos y el tipo de información que deseamos, se puede proceder con la generación de redes. Para un fácil entendimiento en la Figura 9 se presenta el pseudocódigo de lo que sería este proceso.

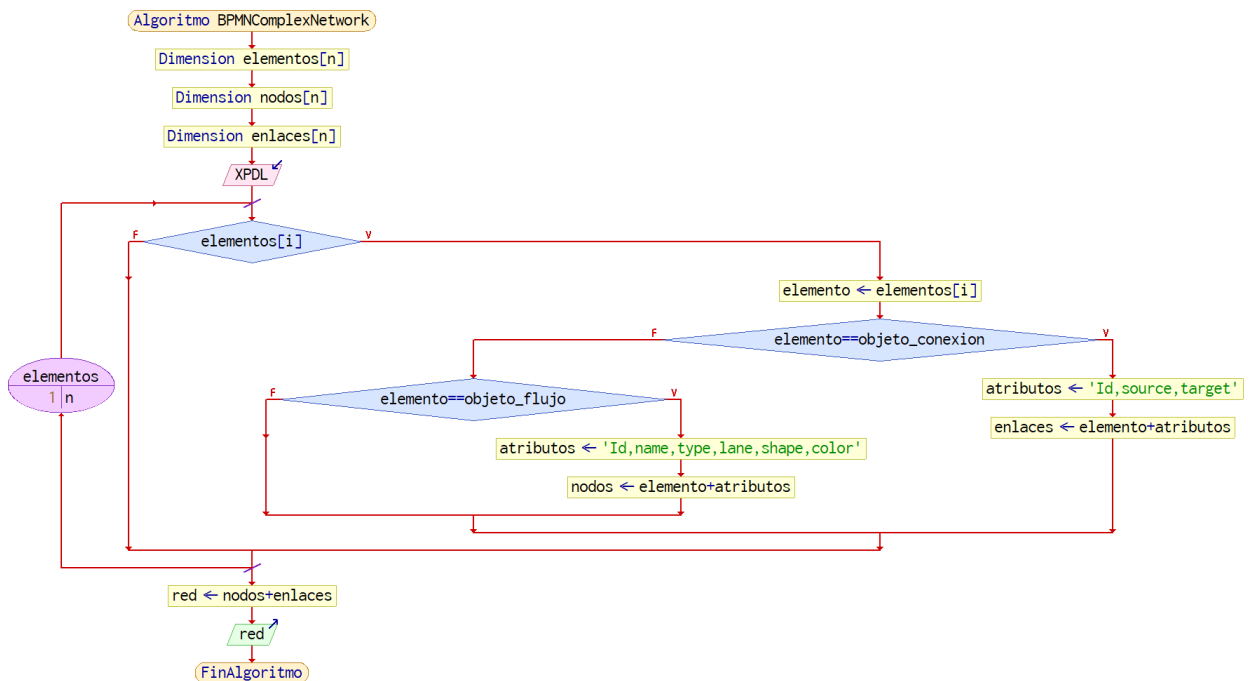


Figura 9 Pseudocódigo para la generación de redes complejas desde procesos de negocio en formato XPDL.

3.5. Creación de la red.

A continuación, se detalla el proceso para la creación y visualización de la red. Explicamos que la creación de la red se realiza con un script de python el mismo que se encarga de extraer la información del proceso leyendo un archivo XPDL. Esta información se manipula para consolidarla de manera que pueda usarse para la creación y visualización de la red. En este proceso se generan archivos que contiene la información de las redes creadas. Esta información puede usarse en diferentes herramientas. En el presente trabajo se usa esta información para crear y visualizar la red en un entorno diferente, escogemos R que es otro lenguaje de programación bastante popular para el análisis de redes.

3.5.1. Creación de la red y visualización usando Python.

Se ha decidió usar el lenguaje de programación Python ya que cuenta con un gran número de librerías para el análisis de redes complejas y su visualización. El script se encarga de lo siguiente:

- 1) Inicialización de variables.
- 2) Lectura del archivo XPDL y extracción de información.
- 3) Consolidación de información de nodos y enlaces.
- 4) Creación de la red.
- 5) Visualización de la red.

En la siguiente sección se detalla estos pasos.

3.5.1.1. Inicializar las variables que almacenaran.

Al principio se inicializan variables que almacenaran la siguiente información:

- Lista de colores para carriles, estos colores se generarán de forma aleatoria.
- Diccionario de figuras para nodos, se usan las figuras establecidas en la tabla de equivalencia (Tabla 1)
- Diccionario con las funciones de los diseños de red.

3.5.1.2. Leer el archivo XPDL y extracción de información.

Al leer el archivo se extrae la información de:

- Carriles.
- Objetos de flujo y artefactos
- Objetos de conexión.

Esta información se almacena en las variables iniciadas en la sección anterior. Anexo 1 Código Python para obtener carriles.

3.5.1.3. Carriles.

Para los carriles se extrae los campos: *Nombre, Id, Parentpool, posición x, posición y*. Esta información se complementa con una generación y asignación dinámica de colores formato cmap (tab20) para cada carril. El código que se encarga de esta funcionalidad se encuentra en el Anexo 1.

3.5.1.4. Objetos de flujo y artefactos.

Para objetos de flujo se extrae los campos: *nombre, id, posición x, posición y, tipo de objeto y subtipo de objeto*. No existe información del carril en los objetos, el carril es asignado analizando la posición del objeto con referencia a los carriles encontrados y su posición. Se complementa la información con una asignación de figura dependiendo del tipo de objeto. El código encargado de la extracción de actividades se puede visualizar en el Anexo 2. El código encargado de la extracción de artefactos se puede visualizar en el Anexo 3. Finalmente, el código encargado de la extracción de objetos se puede visualizar en el Anexo 4.

3.5.1.5. Objetos de conexión.

Para objetos de conexión se extraen los campos: *id, from y to*. El primer campo es el identificador del objeto mientras que los dos últimos son los identificadores de los objetos que están conectados. El código encargado de extraer los flujos de secuencia se encuentra en el Anexo 5. El código encargado de extraer las asociaciones se encuentra en el Anexo 6.

3.5.1.6. Consolidación de información de nodos y enlaces.

Una vez extraída la información, se procede a consolidarla. Para esto se unen las listas de los diferentes objetos en un dataframe. En el caso de los nodos, las columnas del dataframe son:

- Id. - identificador del objeto
- Name. - nombre del objeto
- Type. - tipo del objeto
- Subtype. - subtipo del objeto
- Height. – altura del objeto
- Width. – ancho del objeto
- X. – posición en el plano x del objeto
- Y. – posición en el plano y del objeto
- center_x.- posición en el plano x del centro del objeto
- center_y.-posición en el plano y del centro del objeto
- sum_source. – valor calculado de la suma de objetos cuyo flujo de actividad es dirigido hacia el nodo
- sum_target. – valor calculado de la suma de objetos cuyo flujo de actividad se origina en el nodo
- lane_id. - identificador del carril al que pertenece el nodo
- lane_name. - nombre del carril al que pertenece el nodo
- lane_color. - color que se asignó al carril

En el caso de los enlaces, las columnas del dataframe son:

- Id. – identificador del enlace
- source. -identificador del objeto de partida del enlace
- target. – identificador del objeto hacia el que va dirigido el enlace
- Type. - tipo de enlace
- Subtype. – subtipo del enlace
- source_lane_id. -identificador del carril en el que se origina el enlace
- source_lane_name. – nombre del carril en el que se origina el enlace
- source_lane_color. – color del carril en el que se origina el enlace
- target_lane_id. – identificador del carril hacia donde se dirige el enlace
- target_lane_name. – nombre del carril hacia donde se dirige el enlace
- target_lane_color. – color del carril hacia donde se dirige el enlace

El código que se encarga de consolidar la información antes mencionada se encuentra en el Anexo 7.

3.5.1.7. Creación de la red.

Partiendo de la información consolidada, crearemos la red usando la librería networkx. Para esto creamos la función graph_creator que recibe dos parámetros un dataframe de nodos y otro de enlaces. El código para la creación de la red de la está en el Anexo 8. Debido a que la red en un principio se crea únicamente con la información del dataframe de enlaces, es necesario que por medio de una iteración para cada uno de los nodos se añada sus propiedades (color en base a carril, figura en base a tipo de objeto, etc.). Al momento de crear la red el script se encarga de almacenar la información de los datos consolidados de nodos y enlaces en formato csv. Un ejemplo de los archivos consolidados de la red del proceso A se puede encontrar en la Figura 10.



Figura 10 Ejemplo de archivos consolidados de nodos y enlaces de la red del proceso A.

Adicionalmente se creó una función (lanes_grapher) que se encarga de crear una red ponderada del proceso basada en los carriles como nodos. Esta red facilita la visión general del proceso. Ya que nos muestra la relevancia de cada carril, así como la conectividad entre diferentes carriles. Para la creación de esta red es necesario generar información nueva de nodos y enlaces en base a los carriles.

Iniciamos estableciendo el carril al que cada actividad y su flujo pertenece. Con esta información generamos nuevos datos de nodos y enlaces. El código que se encarga de esto se puede observar en el Anexo 10. Posteriormente, creamos dos nuevos dataframes uno para enlaces y otro para nodos. El dataframe de enlaces contiene las siguientes columnas:

- Source. – identificador del carril desde el que se origina el enlace
- Target . – identificador del carril hacia el que se dirige el enlace
- count. – aquí se almacena la cantidad de enlaces en el carril, al ser una red dirigida los enlaces se cuentan en base a su dirección.

El dataframe de los nodos contiene las siguientes columnas:

- Id. – identificador del carril
- lane_name. – nombre del carril

- ParentPool. – identificador del proceso al que pertenece el carril
- height. – altura del carril
- width. – anchura del carril
- x. – posición del carril en el plano x
- y. – posición del carril en el plano y
- center_x. – posición central del carril en el plano x
- center_y. – posición central del carril en el plano y
- color. – color del carril
- count. – cantidad de objetos de flujo existentes en el carril. Esta columna representa el peso del nodo.

Con este paso se termina la transformación de un modelo de procesos de negocio a una red. Los siguientes pasos se encargarán de la visualización de la red.

3.5.1.8. Visualización de la red.

Antes de poder visualizar la red es necesario establecer un diseño a usar, en el script se implementó diferentes algoritmos para la representación de las redes. Actualmente se añadieron los 7 de los 8 diseños disponibles en networkx.

- a) fruchterman_reingold_layout
- b) circular_layout
- c) kamada_kawai_layout
- d) planar_layout
- e) shell_layout
- f) spring_layout
- g) spectral_layout

Para generar los gráficos de la red se creó la función graph_grapher. Los parámetros que necesita esta función son dataframe de nodos, dataframe de enlaces, dataframe de carriles y el diseño a ser usado. Esta función se encarga de generar una gráfica de la red. El código de la función se encuentra en el Anexo 9

Al momento de graficar cada una de las redes el script se encarga de almacenar las imágenes en archivos externos. La imagen de la red se almacena en tres formatos diferentes (pdf, png y svg) para cada uno de los diseños disponibles en el script. Un ejemplo de los archivos de imágenes de la red del proceso A se puede encontrar en la Figura 11






















	Proceso A circular_layout.pdf	19.2 KB	Adobe Acrobat Document
	Proceso A circular_layout.png	262 KB	IrfanView PNG File
	Proceso A circular_layout.svg	75.4 KB	SVG Document
	Proceso A fruchterman_reingold_layout.pdf	19.2 KB	Adobe Acrobat Document
	Proceso A fruchterman_reingold_layout.png	197 KB	IrfanView PNG File
	Proceso A fruchterman_reingold_layout.svg	75.5 KB	SVG Document
	Proceso A kamada_kawai_layout.pdf	19.2 KB	Adobe Acrobat Document
	Proceso A kamada_kawai_layout.png	158 KB	IrfanView PNG File
	Proceso A kamada_kawai_layout.svg	75.6 KB	SVG Document
	Proceso A planar_layout.pdf	19 KB	Adobe Acrobat Document
	Proceso A planar_layout.png	197 KB	IrfanView PNG File
	Proceso A planar_layout.svg	75.5 KB	SVG Document
	Proceso A shell_layout.pdf	19.2 KB	Adobe Acrobat Document
	Proceso A shell_layout.png	262 KB	IrfanView PNG File
	Proceso A shell_layout.svg	75.4 KB	SVG Document
	Proceso A spectral_layout.pdf	19.3 KB	Adobe Acrobat Document
	Proceso A spectral_layout.png	156 KB	IrfanView PNG File
	Proceso A spectral_layout.svg	75.6 KB	SVG Document
	Proceso A spring_layout.pdf	19.3 KB	Adobe Acrobat Document
	Proceso A spring_layout.png	158 KB	IrfanView PNG File
	Proceso A spring_layout.svg	75.6 KB	SVG Document

Figura 11 Ejemplo de archivos de imagen de la red del proceso A

La finalidad de almacenar la información de la red en archivos externos es poder usar diferentes herramientas para su edición, modificación o visualización.

Adicionalmente, se genera la visualización de la red por carriles. Para esto se llama individualmente a los métodos `draw_networkx_edges` para cada uno de los enlaces y `draw_networkx_nodes` para cada uno de los nodos. Es decir, se grafica individualmente cada uno de los nodos y enlaces, esto se debe a que por limitaciones de la librería solo se puede pasar las propiedades de peso para nodo y enlace de forma individual. En esta red los nodos son los diferentes carriles presentes en cada proceso. Su tamaño representa la cantidad de actividades que se encuentran en el carril y la anchura del enlace representa la cantidad de enlaces entre las diferentes actividades. El código de generación de la red se encuentra en el Anexo 11.

3.5.2. Creación de la red y visualización usando R.

Es posible realizar el proceso de lectura, agregación y consolidación de información de proceso de los archivos XPDL en R, pero por motivos de celeridad usaremos los archivos csv generados por el script en python como fuente de información para la

creación y visualización de redes usando R. Los archivos que usaremos son los csv de nodos y enlaces (*_edges.csv y *_nodes.csv, donde * puede ser reemplazado por el nombre del proceso).

La aplicación en R se encargará de:

- 1) Creación de la red.
- 2) Visualización de la red.

En la siguiente sección se detalla estos pasos.

3.5.2.1. Creación de la red.

Partimos leyendo la información de los nodos y enlace. Es necesario modificar la información para que tenga conformidad con los requisitos del nuevo paquete de visualización (asignar figuras, asignar colores por carril, modificar columnas). El código encargado de esta funcionalidad se encuentra en el Anexo 12

Una vez que la información de nodos y enlaces tiene el formato y las propiedades adecuadas, se procede a generar las redes. Debido a que ciertos análisis no se pueden realizar en redes dirigidas es necesario crear dos tipos de redes a partir de la misma información de cada proceso. La primera será una red dirigida (dnetwork) y la segunda una red no dirigida (unetwork). La única diferencia entre una red dirigida y una no dirigida es que en una red dirigida los enlaces tienen una dirección. El código que se encarga de esta funcionalidad se encuentra en el Anexo 13. Para la creación de la red es necesario pasar al paquete igrph tres argumentos: la información de enlaces, el tipo de red y la información de nodos.

3.5.2.2. Visualización de la red.

Para la visualización de la red usamos el paquete visNetwork. Se llama al método visNetwork pasando como argumentos la información de nodos, enlaces y propiedades de la red como se puede observar en el **Anexo 14**

3.6. Visualización de las redes.

A pesar de que se implementaron varios algoritmos para la visualización de la red, se estableció que los diseños circular y kamada ofrecen un mejor resultado para la presentación de información y son los formatos que se usaran para las ilustraciones de las redes.

Los otros algoritmos presentaron desventajas que se detalla a continuación:

En el caso del algoritmo planar, este ubica a los nodos en una forma similar a una w inversa independiente de la cantidad de nodos o enlaces que tenga la red.

En el caso del algoritmo fruchterman, shell y spring hay una gran cantidad de nodos que se sobreponen unos a otros lo cual le resta visibilidad a la red.

Por último, el algoritmo spectral ubica los nodos en una línea recta lo que evita la correcta visualización de los nodos o enlaces.

3.6.1. Visualización de las redes generadas con Python.

A continuación, se muestra las redes de los procesos combinados que se generaron con Python. Las redes de procesos individuales se encuentran en los anexos:

- Red del Proceso A. **Anexo 25**
- Red del Proceso B. **Anexo 26**
- Red del Proceso C. **Anexo 27**
- Red del Proceso D. **Anexo 28**
- Red del Proceso E. **Anexo 29**
- Red del Proceso F. **Anexo 30**
- Red del Proceso G. **Anexo 31**
- Red del Proceso H. **Anexo 32**
- Red del Proceso I. **Anexo 33**
- Red del Proceso J. **Anexo 34**

3.6.1.1. Red proceso subasta de autos nuevos y usados.

En la Figura 12 podemos observar el proceso subasta de autos nuevos y usados como una red compleja. Esta red está compuesta por 85 nodos, los colores asignados hacen referencia a los carriles al que las actividades pertenecen.

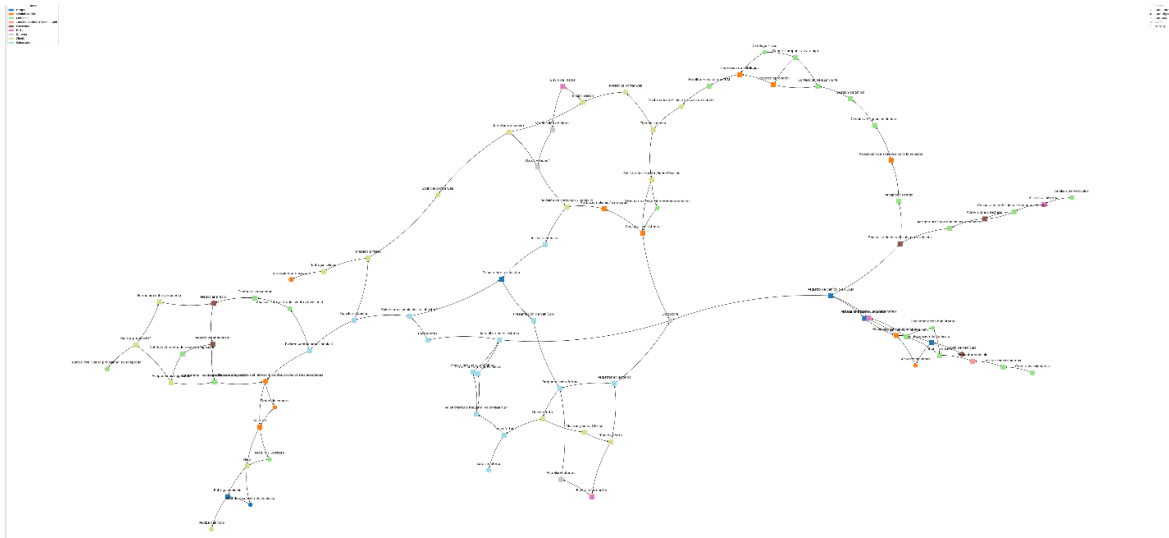


Figura 12 Red del proceso subasta de autos nuevos y usados.

3.6.1.2. Red proceso subasta de autos nuevos y usados.

En la Figura 13 podemos observar el proceso subasta de autos nuevos y usados como una red compleja con un diseño circular. Esta red está compuesta por 85 nodos, los colores asignados hacen referencia a los carriles al que las actividades pertenecen.

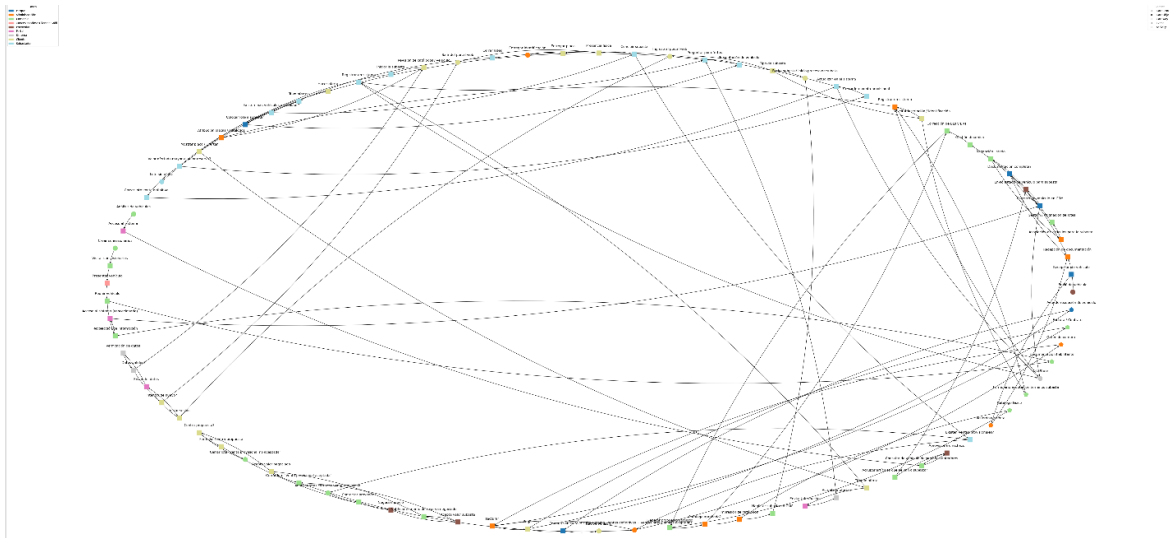


Figura 13 Red del proceso subasta de autos nuevos y usados (Circular).

3.6.1.3. Red proceso subasta de autos siniestrados.

En la Figura 14 podemos observar el proceso subasta de autos siniestrados como una red compleja. Esta red está compuesta por 48 nodos, los colores asignados hacen referencia a los carriles al que las actividades pertenecen.

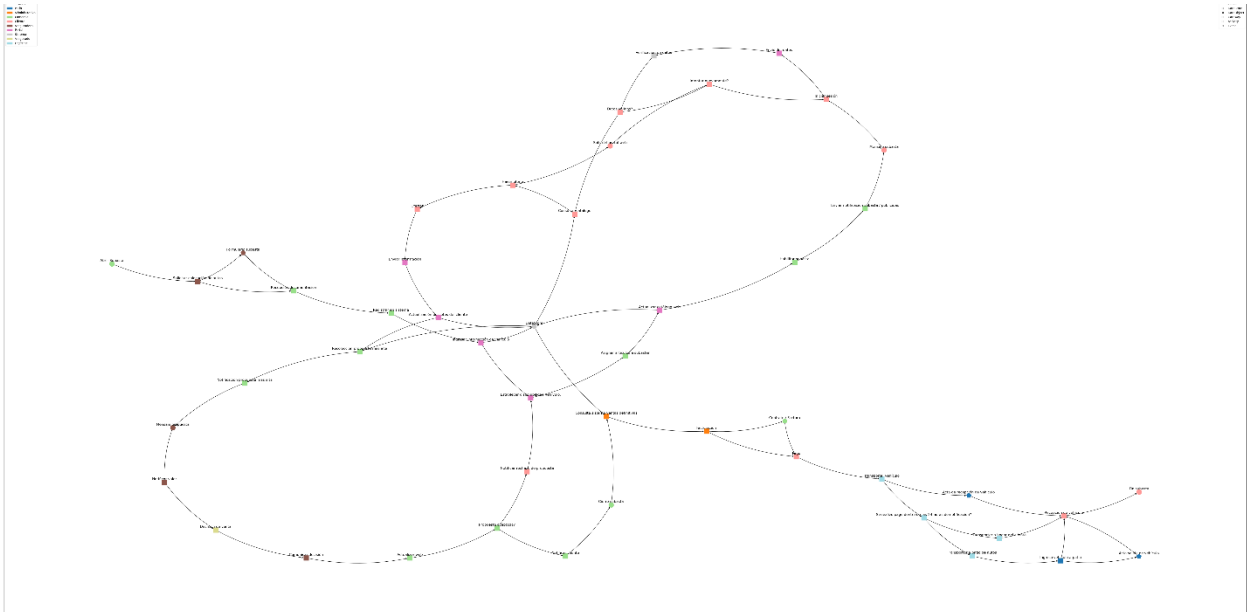


Figura 14 Red del proceso subasta de autos siniestrados.

3.6.1.4. Red proceso subasta de autos siniestrados.

En la Figura 15 podemos observar el proceso subasta de autos siniestrados como una red compleja con un diseño circular. Esta red esta compuesta por 48 nodos, los colores asignados hacen referencia a los carriles al que las actividades pertenecen.

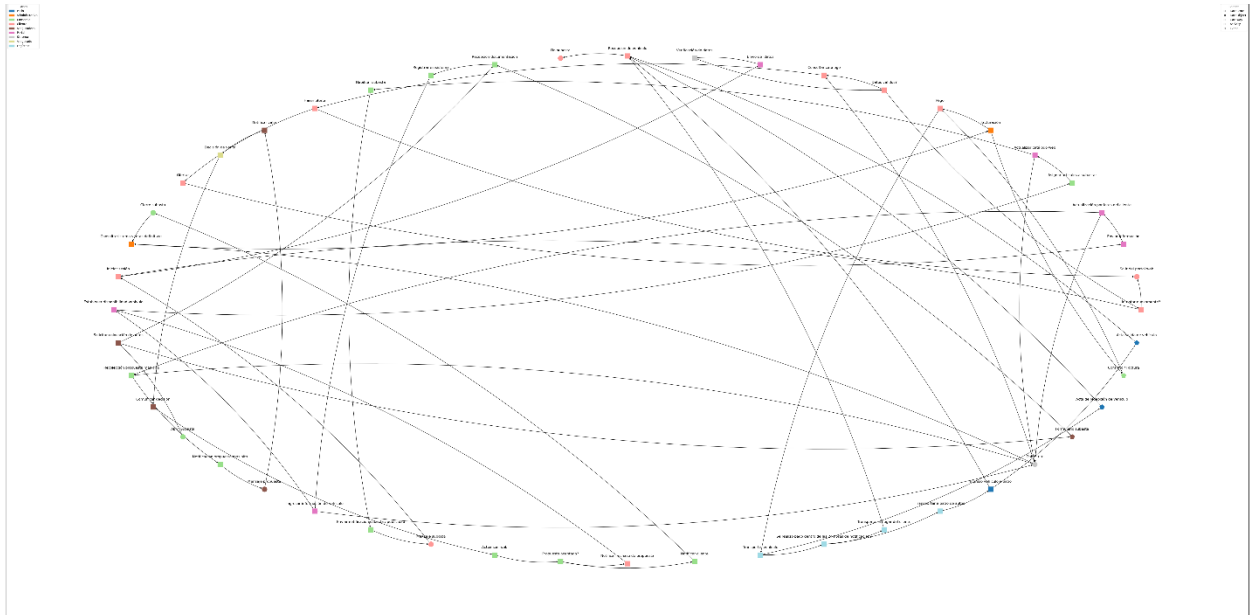


Figura 15 Red del proceso subasta de autos siniestrados (Circular).

3.6.2. Visualización de redes de procesos por carriles generadas con python.

A continuación, se muestra las redes de los procesos por carriles de los procesos combinados. En estas redes cada uno de los nodos representan un carril, es decir un área o departamento de la empresa. El tamaño del nodo refleja la cantidad de actividades que existen por carril. Un nodo más grande representa un mayor número de actividades mientras que un nodo pequeño representa un reducido número de actividades. En el caso de los enlaces, la anchura de la línea representa la cantidad de objetos de conexión que existen entre los carriles, un todo más oscuro denota una mayor cantidad de enlaces mientras que una línea más delgada muestra una menor cantidad de enlaces. Las redes de procesos por carriles para procesos individuales se encuentran en los anexos:

- Red del carril para el Proceso A. **Anexo 35**
- Red del carril para el Proceso B. **Anexo 36**
- Red del carril para el Proceso C. **Anexo 37**
- Red del carril para el Proceso D. **Anexo 38**
- Red del carril para el Proceso E. **Anexo 39**
- Red del carril para el Proceso F. **Anexo 40**
- Red del carril para el Proceso G. **Anexo 41**
- Red del carril para el Proceso H. **Anexo 42**
- Red del carril para el Proceso I. **Anexo 43**
- Red del carril para el Proceso J. **Anexo 44**

3.6.2.1. Red de carriles del proceso subasta de autos nuevos y usados.

En la Figura 16 podemos observar nueve nodos representando los carriles que intervienen en el proceso subasta de autos nuevos y usados (parque, administración, comercial, concesionarios, proveedor, portal, sistema, cliente y subastador)

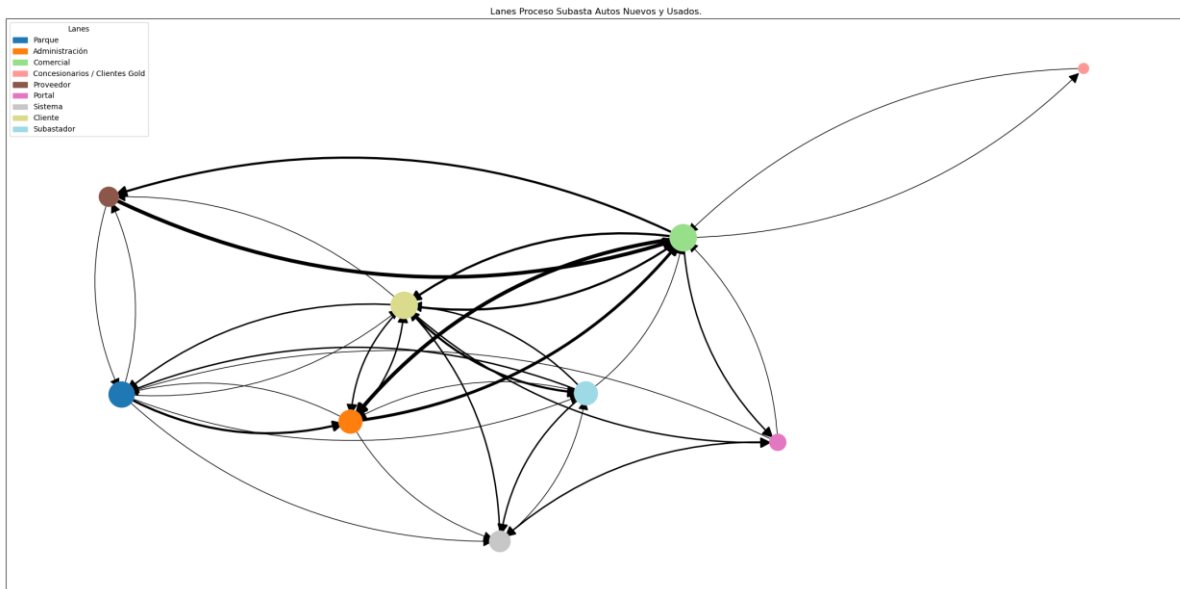


Figura 16 Red del proceso subasta de autos nuevos y usados (carril).

3.6.2.2. Red de carriles del proceso subasta de autos siniestrados

En la Figura 17 podemos observar nueve nodos representando los carriles que intervienen en el proceso subasta autos siniestrados (patio, administrativo, comercial, cliente, aseguradora, portal, sistema, asegurado, logística)

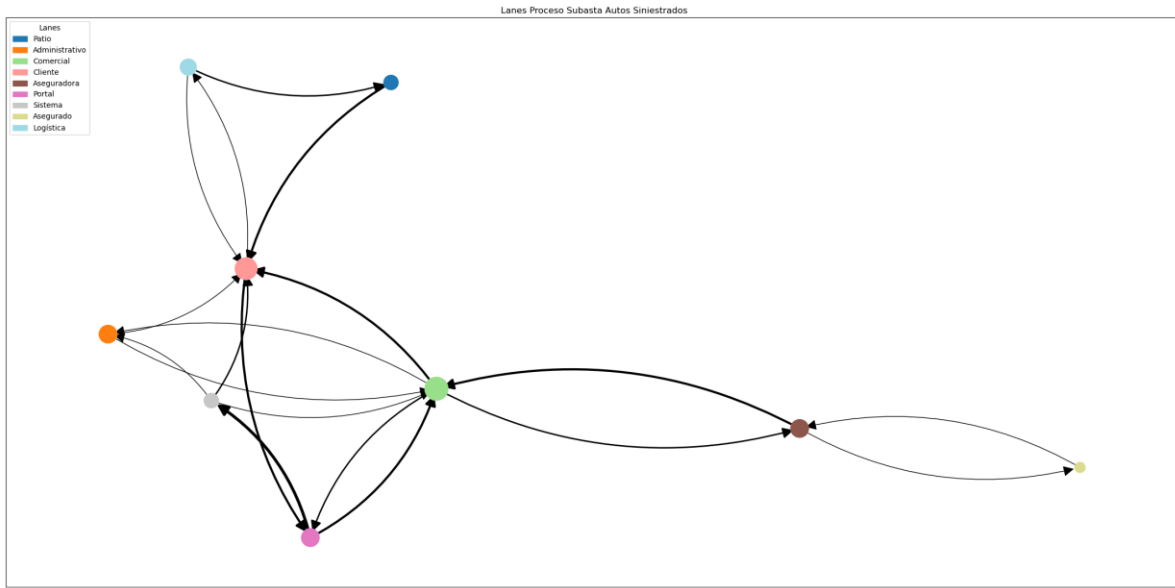


Figura 17 Red del proceso subasta autos siniestrados (carril).

3.6.3. Visualización de redes de procesos generadas con R.

Para no extender innecesariamente la longitud del documento, en la siguiente sección solo se adjunta las imágenes de los procesos combinados: proceso subasta de autos nuevos y usados y proceso subasta de autos siniestrados. Estas visualizaciones son las mismas que las de la sección 3.6.1 con la diferencia de haber sido generados con el lenguaje R. En caso de querer observar la red generada para cada uno de los procesos, se puede acceder a la aplicación BPMNComplexNetwork (<https://ppico.shinyapps.io/BPMNComplexNetwork/>). La aplicación permite la selección de que proceso se va a visualizar, además de una manera dinámica permite la selección de nodos individuales o por carriles.

3.6.3.1. Red proceso subasta autos nuevos y usados.

En la Figura 18 podemos observar la red del proceso unificado AIJBFCD. Los colores hacen referencia a los carriles a los que cada actividad pertenece mientras que la figura hace referencia a el tipo de actividad al que pertenece cada nodo. Dicha figura está usando el algoritmo Kamada Kawai.

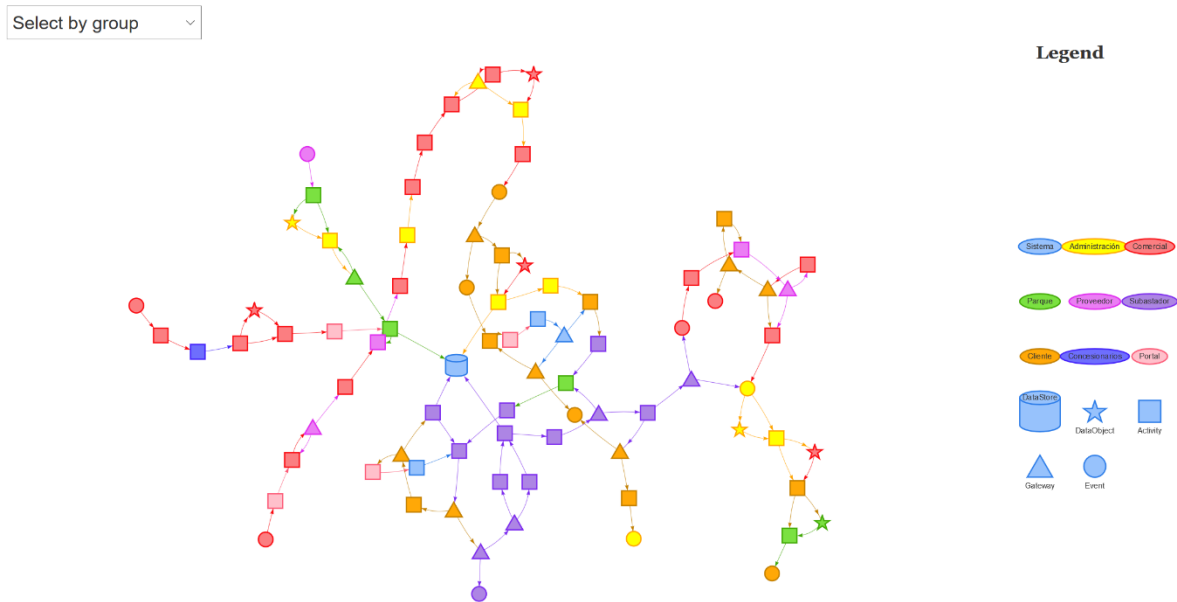


Figura 18 Red del proceso subasta autos nuevos y usados.

3.6.3.2. Red proceso subasta autos nuevos y usados.

En la Figura 19 podemos observar la red del proceso subasta autos nuevos y usados. Los colores hacen referencia a los carriles a los que cada actividad pertenece mientras que la figura hace referencia a el tipo de actividad al que pertenece cada nodo. Dicha figura está usando el algoritmo circular.

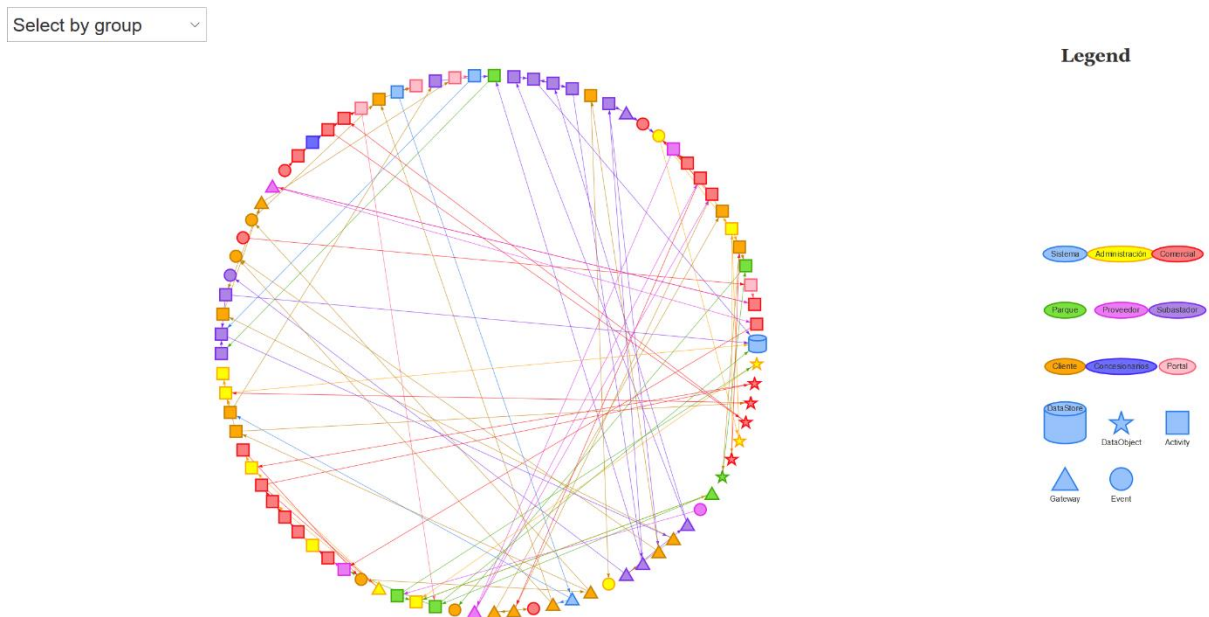


Figura 19 Red del proceso subasta de autos nuevos y usados (Circular)

3.6.3.3. Red proceso subasta de autos siniestrados

En la Figura 20 podemos observar la red del proceso subasta autos siniestrados. Los colores hacen referencia a los carriles a los que cada actividad pertenece mientras que la figura hace referencia a el tipo de actividad al que pertenece cada nodo. Dicha figura está usando el algoritmo Kamada Kawai.

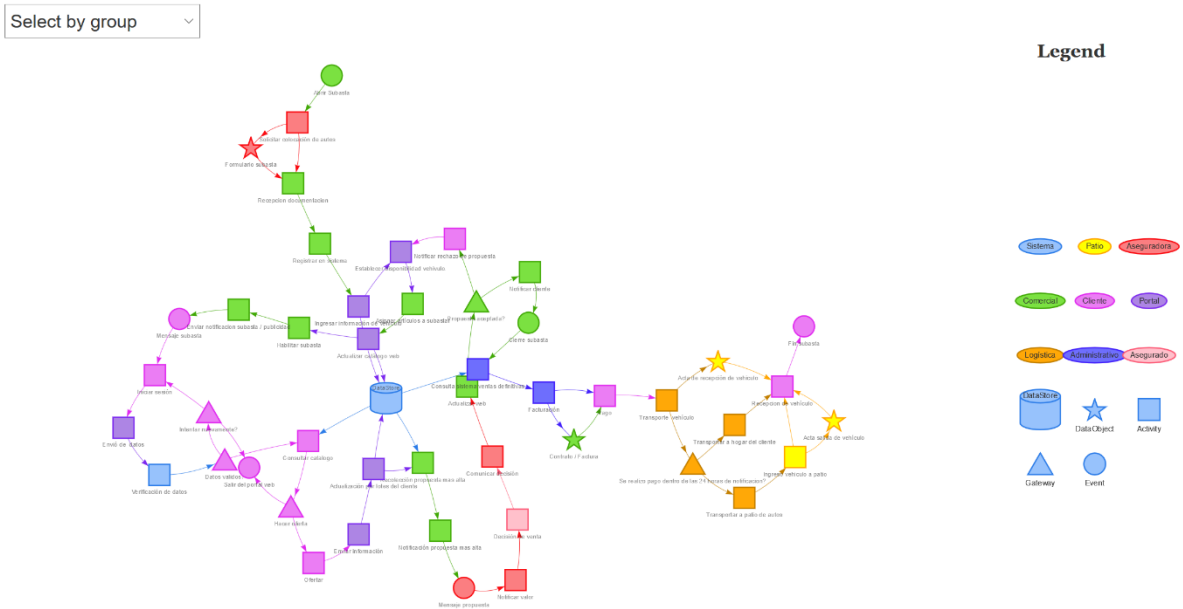


Figura 20 Red del proceso subasta autos siniestrados.

3.6.3.4. Red proceso subasta de autos siniestrados

En la Figura 21 podemos observar la red del proceso subasta autos siniestrados. Los colores hacen referencia a los carriles a los que cada actividad pertenece mientras que la figura hace referencia a el tipo de actividad al que pertenece cada nodo. Dicha figura está usando el algoritmo Circular.

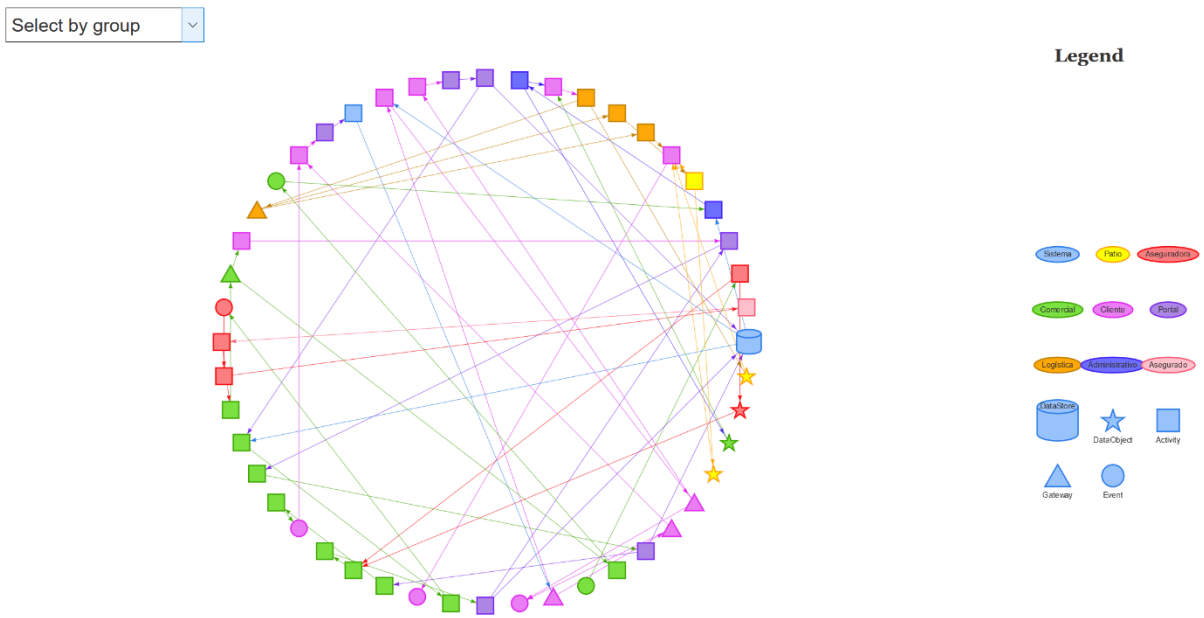


Figura 21 Red del proceso subasta autos siniestrados (Circular).

3.7. Análisis de la red.

El análisis de la red se realizó en R usando los paquetes igraph y netstan. Lo primero que debemos hacer es obtener métricas de los nodos. Se usaron tres medidas de centralidad comunes para los nodos individuales: grado, coeficiente de clusterización, e intermediación

- El grado, k_i , de un nodo es el número de enlaces conectados a un nodo i . En el caso de redes dirigidas se calcula la cantidad de enlaces entrantes conectados, la cantidad de enlaces salientes conectados y la cantidad total de enlaces conectados.
- El coeficiente de clusterización relaciona la conectividad de un nodo con sus vecinos. La fórmula que se usa para calcular este valor se encuentra en Figura 22. Esta medida se define como C_i , donde e_i es el número de enlaces que conectan los vecinos k_i del nodo i entre sí. (Newman M. E., Networks an Introduction, 2010)

$$C_i = 2e_i[k_i(k_i - 1)]^{-1}$$

Figura 22 Fórmula coeficiente de clusterización.

Tomado de (Newman M. E., Networks an Introduction, 2010)

- La intermediación, B_k , se define como el número de rutas más cortas entre dos nodos que pasan por k . donde $\sigma_{st}(k)$ es el número de caminos más cortos entre los nodos s y t que pasan a través del nodo k ; σ_{st} es el número total de caminos más cortos entre s y t . (Newman M. E., Networks an Introduction, 2010)

$$B_k = \sum_s \sum_t \frac{\sigma_{st}(k)}{\sigma_{st}}$$

Figura 23 Fórmula coeficiente de intermediación.

Tomado de (Newman M. E., Networks an Introduction, 2010)

El grado y coeficiente de agrupamiento representan la importancia de los nodos dentro de la estructura de la red. La intermediación representa la importancia funcional de un nodo al relacionar su posición con las rutas de flujo eficientes.

3.7.1. Medidas de los nodos de los procesos interconectados.

Las medidas obtenidas nos indican que el grado de conectividad de los procesos es bajo. Como se puede observar en las distribuciones de los nodos en base a su grado de la Tabla 3 y Tabla 5, la mayoría de los nodos tienen un enlace entrante, un enlace saliente, y dos enlaces totales. Esto se debe a que los procesos tienen un flujo de proceso bastante lineal, el flujo saliente o entrante de las actividades es reducido y únicamente existe una salida o una entrada para las actividades del proceso de negocio.

Los valores obtenidos de cada uno de los nodos se pueden ver a detalle en la Tabla 2 para el proceso de subasta de autos nuevos y usados; y en la Tabla 4 para el proceso de subasta de autos siniestrados. En la primera columna de la tabla encontramos el nombre del nodo. En la segunda, tercera y cuarta columna encontramos información del grado del nodo, cantidad de enlaces entrantes, cantidad de enlaces salientes y cantidad de enlaces totales respectivamente. En la quinta columna encontramos el coeficiente de clusterización del nodo, este valor nos indica que tan conectado está el nodo con sus vecinos, partiendo desde 0 no muy conectado hasta un valor de 1 que indica que el nodo está bastante conectado. En el caso de valores NA se debe a que posee una baja cantidad de enlaces y además carece de enlaces entrantes o salientes. En la quinta columna encontramos el valor de intermediación del nodo, este valor es alto debido a que representa el número de rutas, las cuales pasan por el nodo.

Tabla 2 Medidas de los nodos del proceso subasta autos nuevos y usados.

Nombre	Enlaces entrantes	Enlaces salientes	Enlaces totales	Coeficiente de clusterización	intermediación
DataStore	4	0	4	0	0
Acuerdo de venta	1	1	2	1	0
Catalogo Físico	1	1	2	0	718.5
Formulario aceptación términos subasta	1	1	2	1	0
documentación habilitante	1	1	2	1	0
Orden de compra	1	1	2	1	0

Nombre	Enlaces entrantes	Enlaces salientes	Enlaces totales	Coefficiente de clusterización	de intermediación
Factura / Contrato	1	1	2	1	0
Acta de recepción de vehículo	1	1	2	1	0
¿Documentación completa?	1	2	3	0	272
Envío de vehículo	0	1	1	NA	0
¿Valor ofertado mayor al valor reserva?	1	2	3	0	1536
Hacer oferta	1	2	3	0	1929
Presencia física	1	2	3	0	143
¿Existen más vehículos a subastar?	1	2	3	0	1446
Tuvo ofertas	1	2	3	0	1612
Entregar identificación	1	0	1	NA	0
Tipo de subasta	1	2	3	0	1607
¿Datos validos?	1	2	3	0	288
¿Intentar de nuevo?	1	2	3	0	42
Cerrar lote "venta provisional no aceptada"	1	0	1	NA	0
Acepta valor negociado	1	2	3	0	212
¿Contrapropuesta?	1	2	3	0	141
Acepta valor subasta	1	2	3	0	433
Recibe vehículo	1	0	1	NA	0
Registro de vehículo en CRM	2	2	4	0	804
Recepción de documentación	3	1	4	0.1666667	207
Recepción de vehículo	1	2	3	0.3333333	71

Nombre	Enlaces entrantes	Enlaces salientes	Enlaces totales	Coefficiente de clusterización	de intermediación
¿Catalogo aprobado?	1	2	3	0.33333333	719.5
Fecha subasta/ Inicio proceso de subasta	1	1	2	0	1582
Envió listado de vehículo para subasta	2	1	3	0	1175
Recepción listado	1	1	2	0	1222
Asociación de artículos para la subasta	1	1	2	0	1267
Gestión / asignación de lotes	1	1	2	0	1310
Gestión dinámica	1	1	2	0	1351
Corrección de BLs y BRs	2	1	3	0.33333333	1392
Generar propuesta catalogo	1	2	3	0.33333333	1428
Impresión de catálogos	2	1	3	0	1526
Habilitar subasta en CRM	1	1	2	0	1555
Solicitar inscripción / identificación	1	2	3	0.33333333	1413
Revisión de catálogos / vehículo	2	1	3	0	1692
Registrar en sistema	2	2	4	0.16666667	1441
Atribución placas / catálogos	1	1	2	0	1431
Presentación de vehículo	1	1	2	0	1768
Preguntar por ofertas	3	1	4	0	1946
Mostrar placa / Ofertar	1	1	2	0	242

Nombre	Enlaces entrantes	Enlaces salientes	Enlaces totales	Coefficiente de clusterización	de intermediación
Registrar oferta en sistema	1	2	3	0	93
Lote sin oferta	1	0	1	NA	0
Salir del portal web	2	0	2	0	0
Análisis de vehículos	0	1	1	NA	0
Ingreso al portal web	1	1	2	0	186
Tipo de oferta	1	2	3	0	225
Aprovisiona o rechazo	1	2	3	0	204
Crear concesionarios	0	1	1	NA	0
Visitar concesionarios	1	1	2	0	73
Presentar vehículo	1	1	2	0	144
Reunir vehículo	1	2	3	0.3333333	213
Recolección de información	2	1	3	0.3333333	345
Acceso al sistema (concesionario)	1	1	2	0	408
Iniciar sesión	2	1	3	0	252
Verificación de datos	1	1	2	0	276
Envío de datos	1	1	2	0	264
Iniciar la subasta	1	1	2	0	1693
Enviar información	1	1	2	0	57
Actualizar ofertas	1	1	2	0	40
Colocar lote a subastar	2	1	3	0	1785
Cerrar lote venta definitiva	1	1	2	0	730.5
Actualizar en el sistema	2	2	4	0	1485
Cerrar lote venta provisional	1	1	2	0	730.5
Cerrar lotes	1	1	2	0	1463
Entregar placa	1	1	2	0	61

Nombre	Enlaces entrantes	Enlaces salientes	Enlaces totales	Coefficiente de clusterización	de intermediación
Concluir subasta	1	2	3	0	1320
¿Existen ventas provisionales?	1	2	3	0	1080
Analizar lotes cerrados venta provisional	1	1	2	0	549
Consulta sistema ventas definitivas + ventas provisionales aceptadas	2	2	4	0.1666667	490
Negociar precio	2	1	3	0	486
Contactar proveedor	1	1	2	0	504
Notificar el monto de reserva negociado	1	1	2	0	265
Cerrar lote "Venta provisional aceptada"	2	1	3	0	64
Formular contrapropuesta	1	1	2	0	19
Facturar	2	2	4	0.3333333	360
Pago	2	2	4	0.3333333	222
Entrega vehículo	2	1	3	0.3333333	76
Acceso al sistema	1	1	2	0	70
Consulta de vehículo de semanas anteriores	2	1	3	0	138
Agrupar artículos que se van a subastar	1	1	2	0	268

Tabla 3 Distribución de enlaces en base al grado de los nodos del proceso subasta autos nuevos y usados.

Grado	# enlaces entrantes	% enlaces entrantes	# enlaces salientes	% enlaces salientes	# enlaces totales	% enlaces totales
0	3	0.04	6	0.07	0	0.00
1	61	<u>0.72</u>	51	<u>0.60</u>	7	0.08
2	18	0.21	28	0.33	36	<u>0.42</u>
3	2	0.02	0	0.00	33	0.39
4	1	0.01	0	0.00	9	0.11

Tabla 4 Medidas de los nodos del proceso subasta de autos siniestrados

Nombre.	Enlaces entrantes.	Enlaces salientes.	Enlaces totales.	Coficiente de clusterización.	Intermediación.
DataStore	3	3	6	0.0666667	560
Acta de recepción de vehículo	1	1	2	0	78
Formulario subasta	1	1	2	1	0
Contrato / Factura	1	1	2	1	0
Acta salida de vehículo	1	1	2	1	0
Hacer oferta	1	2	3	0	388
¿Intentar nuevamente?	1	2	3	0	13
Actualizar catálogo web	1	2	3	0	379
Notificar cliente	1	1	2	0	136
Abrir Subasta	0	1	1	NA	0
¿Datos validos?	1	2	3	0	285
Salir del portal web	2	0	2	0	0
Ingresar información de vehículo	1	2	3	0	210
Notificación propuesta más alta	1	1	2	0	381

Nombre.	Enlaces entrantes.	Enlaces salientes.	Enlaces totales.	Coefficiente de clusterización.	Intermediación.
Fin subasta	1	0	1	NA	0
Habilitar subasta	1	1	2	0	179
recepción documentación	2	1	3	0.3333333	132
Registrar en sistema	1	1	2	0	172
Mensaje subasta	1	1	2	0	199
Enviar notificación subasta / publicidad	1	1	2	0	189
Asignar artículos a subastar	1	1	2	0	369
Recolección propuesta más alta	2	1	3	0.3333333	371
Actualizar web	1	1	2	0	431
Notificar valor	1	1	2	0	401
Comunicar decisión	1	1	2	0	421
Mensaje propuesta	1	1	2	0	391
¿Propuesta aceptada?	1	2	3	0	441
Notificar rechazo de propuesta	1	1	2	0	283
¿Se realizó pago dentro de las 24 horas de notificación?	1	2	3	0	156
Cierre subasta	1	1	2	0	117
Iniciar sesión	2	1	3	0	255
Envío de datos	1	1	2	0	265
Verificación de datos	1	1	2	0	275
Consultar catálogo	2	1	3	0	378
Ofertar	1	1	2	0	374
Enviar información	1	1	2	0	384
Actualización por lotes del cliente	1	2	3	0.3333333	394
Facturación	1	2	3	0.3333333	385

Nombre.	Enlaces entrantes.	Enlaces salientes.	Enlaces totales.	Coefficiente de clusterización.	Intermediación.
Pago	2	1	3	0.33333333	333
Transporte vehículo	1	2	3	0	304
Transportar a patio de autos	1	1	2	0	80
Transportar a hogar del cliente	1	1	2	0	2
recepción de vehículo	4	1	5	0.1	45
Ingreso vehículo a patio	1	2	3	0.33333333	43
Consulta sistema ventas definitivas	2	1	3	0	408
Establecer disponibilidad vehículo.	2	1	3	0	359
Solicitar colocación de autos	1	2	3	0.33333333	46
Decisión de venta	1	1	2	0	411

Tabla 5 Distribución de enlaces en base al grado de los nodos del proceso subasta autos siniestrados.

Grado	# enlaces entrantes	% enlaces entrantes	# enlaces salientes	% enlaces salientes	# enlaces totales	% enlaces totales
0	1	0.02	2	0.04	0	0.00
1	37	<u>0.77</u>	33	<u>0.69</u>	2	0.04
2	8	0.17	12	0.25	25	<u>0.52</u>
3	1	0.02	1	0.02	19	0.40
4	1	0.02	0	0.00	0	0.00
5	0	0.00	0	0.00	1	0.02
6	0	0.00	0	0.00	1	0.02

3.7.2. Medidas para determinar la robustez.

Cada vez que un nodo es eliminado de la red, los componentes se podrían desconectar. Esta pérdida de conectividad nos permite determinar la robustez de una red.

3.7.2.1. Medidas de robustez de los procesos interconectados.

A continuación, la Tabla 6 y Tabla 7 muestran los valores que corresponden a la pérdida de conectividad para diferentes estrategias de ataque del proceso de subasta de autos nuevos y usados y del proceso de subasta de autos siniestrados respectivamente. La primera columna corresponde a los nombres de los nodos. La segunda columna corresponde a la fracción de nodos eliminados, este valor será incremental y corresponde a la cantidad de nodos atacados en la red. La tercera columna corresponde a la pérdida de conectividad inducida por un ataque basado en la intermediación. La cuarta columna corresponde a la pérdida de conectividad inducida por un ataque basado en grados. La quinta columna corresponde a la pérdida de conectividad inducida por el ataque en cascada. La sexta columna corresponde a la pérdida de conectividad inducida por un error aleatorio. Estas columnas muestran la pérdida de conectividad de la red y su valor va incrementando en base a la cantidad de nodos eliminados.

Tabla 6 Medidas de robustez del proceso subasta de autos nuevos y usados

Nombre.	Nodos eliminados.	Ataque basado en intermediación.	Ataque basado en grado.	Ataque en cascada.	Error aleatorio.
DataStore	0.011765	0.023529	0.091597	0.023529	0.047487
Acuerdo de venta	0.023529	0.046779	0.136134	0.046779	0.114078
Catalogo Físico	0.035294	0.092437	0.179552	0.405042	0.160972
Formulario aceptación términos subasta	0.047059	0.092437	0.20084	0.516527	0.213798
Documentación habilitante	0.058824	0.114846	0.221849	0.521569	0.26835
Orden de compra	0.070588	0.136975	0.242577	0.658543	0.314852
Factura / Contrato	0.082353	0.180392	0.321849	0.761905	0.355824
Acta de recepción de vehículo	0.094118	0.201681	0.644538	0.783473	0.402748
¿Documentación completa?	0.105882	0.222689	0.647619	0.793277	0.45856
Envío de vehículo	0.117647	0.243417	0.686835	0.843417	0.497639
¿Valor ofertado mayor al valor reserva?	0.129412	0.263866	0.687675	0.888235	0.539378
Hacer oferta	0.141176	0.284034	0.70028	0.909244	0.576457
Presencia física	0.152941	0.67451	0.776471	0.938095	0.616311
¿Existen más vehículos a subastar?	0.164706	0.683193	0.806443	0.943978	0.650773
Tuvo ofertas	0.176471	0.691597	0.815966	0.947059	0.680339
Entregar identificación	0.188235	0.729972	0.85098	0.9493	0.715697
Tipo de subasta	0.2	0.729972	0.914566	0.952941	0.742815
¿Datos validos?	0.211765	0.747619	0.917647	0.959944	0.764275
¿Intentar de nuevo?	0.223529	0.747899	0.919608	0.962185	0.784266

Nombre.	Nodos eliminados.	Ataque basado en intermediación.	Ataque basado en grado.	Ataque en cascada.	Error aleatorio.
Cerrar lote "venta provisional no aceptada"	0.235294	0.764426	0.92437	0.966947	0.797314
Acepta valor negociado	0.247059	0.772269	0.933613	0.968908	0.820048
¿Contrapropuesta?	0.258824	0.833613	0.933613	0.977031	0.837045
Acepta valor subasta	0.270588	0.841176	0.941176	0.979552	0.854821
Recibe vehículo	0.282353	0.848459	0.94986	0.984314	0.873513
Registro de vehículo en CRM	0.294118	0.855462	0.964986	0.985714	0.884429
Recepción de documentación	0.305882	0.888796	0.969748	0.988235	0.899003
Recepción de vehículo	0.317647	0.893838	0.973389	0.989076	0.909039
¿Catalogo aprobado?	0.329412	0.928852	0.97591	0.990476	0.917154
Fecha subasta/ Inicio proceso de subasta	0.341176	0.928852	0.97591	0.992437	0.923448
Envío listado de vehículo para subasta	0.352941	0.928852	0.976751	0.993277	0.930045
Recepción listado	0.364706	0.928852	0.978151	0.993557	0.93672
Asociación de artículos para la subasta	0.376471	0.928852	0.980672	0.993838	0.941602
Gestión / Asignación de lotes	0.388235	0.933613	0.980952	0.993838	0.946801
Gestión dinámica	0.4	0.938095	0.986275	0.993838	0.952527
Corrección de BLs y BRs	0.411765	0.957983	0.986835	0.993838	0.956319
Generar propuesta catalogo	0.423529	0.959944	0.988235	0.994118	0.961174

Nombre.	Nodos eliminados.	Ataque basado en intermediación.	Ataque basado en grado.	Ataque en cascada.	Error aleatorio.
Impresión de catálogos	0.435294	0.961625	0.992997	0.994118	0.963916
Habilitar subasta en CRM	0.447059	0.963305	0.993277	0.994398	0.966258
Solicitar inscripción / identificación	0.458824	0.966387	0.994678	0.994958	0.968415
Revisión de catálogos / vehículo	0.470588	0.967787	0.994958	0.995238	0.970969
Registrar en sistema	0.482353	0.970308	0.995798	0.995518	0.972672
Atribución placas / catálogos	0.494118	0.972549	0.995798	0.995518	0.975445
Presentación de vehículo	0.505882	0.973669	0.995798	0.995798	0.977401
Preguntar por ofertas	0.517647	0.97479	0.996078	0.996078	0.979118
Mostrar placa / Ofertar	0.529412	0.97619	0.996078	0.996078	0.981499
Registrar oferta en sistema	0.541176	0.978151	0.996078	0.996078	0.983241
Lote sin oferta	0.552941	0.981232	0.996359	0.996359	0.984454
Salir del portal web	0.564706	0.982633	0.996359	0.996639	0.985969
Análisis de vehículos	0.576471	0.984874	0.996639	0.996639	0.987356
Ingreso al portal web	0.588235	0.986835	0.996639	0.996639	0.988448
Tipo de oferta	0.6	0.988796	0.996639	0.996639	0.989798
Aprovisiona o rechazo	0.611765	0.990756	0.996639	0.996639	0.990549
Crear concesionarios	0.623529	0.991597	0.996919	0.997479	0.991443
Visitar concesionarios	0.635294	0.992437	0.996919	0.997479	0.992518
Presentar vehículo	0.647059	0.992437	0.996919	0.997759	0.993132
Reunir vehículo	0.658824	0.992997	0.997199	0.998039	0.994062
Recolección de información	0.670588	0.995238	0.997199	0.998039	0.994563

Nombre.	Nodos eliminados.	Ataque basado en intermediación.	Ataque basado en grado.	Ataque en cascada.	Error aleatorio.
Acceso al sistema (concesionario)	0.682353	0.996078	0.997199	0.998039	0.995162
Iniciar sesión	0.694118	0.996639	0.997759	0.998319	0.995597
Verificación de datos	0.705882	0.996919	0.998039	0.998599	0.996028
Envío de datos	0.717647	0.997759	0.998039	0.99944	0.99637
Iniciar la subasta	0.729412	0.998039	0.998039	0.99944	0.996658
Enviar información	0.741176	0.99888	0.998039	0.99972	0.996997
Actualizar ofertas	0.752941	0.99916	0.998039	0.99972	0.997423
Colocar lote a subastar	0.764706	0.99916	0.998039	0.99972	0.997683
Cerrar lote venta definitiva	0.776471	0.99944	0.998319	0.99972	0.997916
Actualizar en el sistema	0.788235	0.99972	0.99916	0.99972	0.998137
Cerrar lote venta provisional	0.8	1	0.99972	1	0.998401
Cerrar lotes	0.811765	1	1	1	0.998714
Entregar placa	0.823529	1	1	1	0.998894
Concluir subasta	0.835294	1	1	1	0.999056
¿Existen ventas provisionales?	0.847059	1	1	1	0.999202
Analizar lotes cerrados venta provisional	0.858824	1	1	1	0.999359
Consulta sistema ventas definitivas + ventas provisionales aceptadas	0.870588	1	1	1	0.999479
Negociar precio	0.882353	1	1	1	0.999602
Contactar proveedor	0.894118	1	1	1	0.999686
Notificar el monto de reserva negociado	0.905882	1	1	1	0.999762

Nombre.	Nodos eliminados.	Ataque basado en intermediación.	Ataque basado en grado.	Ataque en cascada.	Error aleatorio.
Cerrar lote "Venta provisional aceptada"	0.917647	1	1	1	0.999821
Formular contrapropuesta	0.929412	1	1	1	0.999866
Facturar	0.941176	1	1	1	0.999919
Pago	0.952941	1	1	1	0.999947
Entrega vehículo	0.964706	1	1	1	0.999978
Acceso al sistema	0.976471	1	1	1	0.999997
Consulta de vehículo de semanas anteriores	0.988235	1	1	1	1
Agrupar artículos que se van a subastar	1	1	1	1	1

Tabla 7 Medidas de robustez del proceso subasta de autos siniestrados

Nombre.	Nodos eliminados.	Ataque basado en intermediación.	Ataque basado en grado.	Ataque en cascada.	Error aleatorio.
DataStore	0.020833	0.041667	0.041667	0.041667	0.079273
Acta de recepción de vehículo	0.041667	0.494681	0.12234	0.494681	0.154619
Formulario subasta	0.0625	0.521277	0.199468	0.521277	0.224929
Contrato / Factura	0.083333	0.546986	0.371454	0.68883	0.294255
Acta salida de vehículo	0.104167	0.571809	0.643617	0.743794	0.36867
Hacer oferta	0.125	0.605496	0.658688	0.842199	0.427154
¿Intentar nuevamente?	0.145833	0.629433	0.678191	0.883865	0.507296
Actualizar catálogo web	0.166667	0.713652	0.679965	0.911348	0.566108
Notificar cliente	0.1875	0.715426	0.680851	0.921099	0.621959
Abrir Subasta	0.208333	0.77039	0.850177	0.930851	0.673174

Nombre.	Nodos eliminados.	Ataque basado en intermediación.	Ataque basado en grado.	Ataque en cascada.	Error aleatorio.
¿Datos validos?	0.229167	0.780142	0.862589	0.946809	0.720532
Salir del portal web	0.25	0.781028	0.909574	0.95656	0.759557
Ingresar información de vehículo	0.270833	0.781915	0.912234	0.969858	0.791853
Notificación propuesta más alta	0.291667	0.878546	0.941489	0.976064	0.816498
Fin subasta	0.3125	0.886525	0.946809	0.978723	0.84398
Habilitar subasta	0.333333	0.886525	0.951241	0.981383	0.86852
recepción documentación	0.354167	0.886525	0.952128	0.984043	0.886392
Registrar en sistema	0.375	0.893617	0.967199	0.986702	0.905204
Mensaje subasta	0.395833	0.905142	0.973404	0.991135	0.918387
Enviar notificación subasta / publicidad	0.416667	0.921986	0.97695	0.992908	0.931356
Asignar artículos a subastar	0.4375	0.929078	0.981383	0.993794	0.941844
Recolección propuesta más alta	0.458333	0.93617	0.991135	0.995567	0.94758
Actualizar web	0.479167	0.93617	0.991135	0.996454	0.955204
Notificar valor	0.5	0.942376	0.991135	0.996454	0.962784
Comunicar decisión	0.520833	0.947695	0.992021	0.996454	0.968342
Mensaje propuesta	0.541667	0.957447	0.992021	0.99734	0.973342
¿Propuesta aceptada?	0.5625	0.961879	0.992908	0.99734	0.976693
Notificar rechazo de propuesta	0.583333	0.963652	0.992908	0.998227	0.979672
¿Se realizo pago dentro de las 24 horas de notificación?	0.604167	0.964539	0.993794	0.998227	0.983094
Cierre subasta	0.625	0.964539	0.993794	0.998227	0.985638
Iniciar sesión	0.645833	0.968085	0.996454	0.999113	0.988103
Envío de datos	0.666667	0.974291	0.99734	0.999113	0.989663

Nombre.	Nodos eliminados.	Ataque basado en intermediación.	Ataque basado en grado.	Ataque en cascada.	Error aleatorio.
Verificación de datos	0.6875	0.975177	0.99734	0.999113	0.991418
Consultar catalogo	0.708333	0.977837	0.99734	0.999113	0.992934
Ofertar	0.729167	0.977837	0.99734	0.999113	0.994105
Enviar información	0.75	0.983156	1	1	0.995248
Actualización por lotes del cliente	0.770833	0.987589	1	1	0.996188
Facturación	0.791667	0.990248	1	1	0.997154
Pago	0.8125	0.998227	1	1	0.997784
Transporte vehículo	0.833333	0.999113	1	1	0.998298
Transportar a patio de autos	0.854167	1	1	1	0.998679
Transportar a hogar del cliente	0.875	1	1	1	0.999025
recepción de vehículo	0.895833	1	1	1	0.999486
Ingreso vehículo a patio	0.916667	1	1	1	0.999672
Consulta sistema ventas definitivas	0.9375	1	1	1	0.99984
Establecer disponibilidad vehículo.	0.958333	1	1	1	0.999947
Solicitar colocación de autos	0.979167	1	1	1	1
Decisión de venta	1	1	1	1	1

3.7.3. Detección de comunidades.

Los algoritmos de agrupación buscan capturar la noción de que los nodos deben estar conectados a muchos nodos en la misma comunidad, pero conectados a pocos nodos en otras comunidades. Usaremos dos algoritmos de agrupamiento en este estudio: Fast-Greedy e Infomap. Debido a que el algoritmo Fast-Greedy solo puede ser usado en redes no dirigidas primero debemos simplificar la red para poder usarlo. Se usan ambos algoritmos para poder contrastar la cantidad de comunidades que cada uno detecta.

3.7.3.1. Comunidades para el proceso subasta de autos nuevos y usados usando algoritmo Fast-Greedy.

Al aplicar el algoritmo de detección Fast Greedy para el proceso subasta de autos nuevos y usados se encontraron 11 comunidades, tal como se puede observar en la Figura 24.

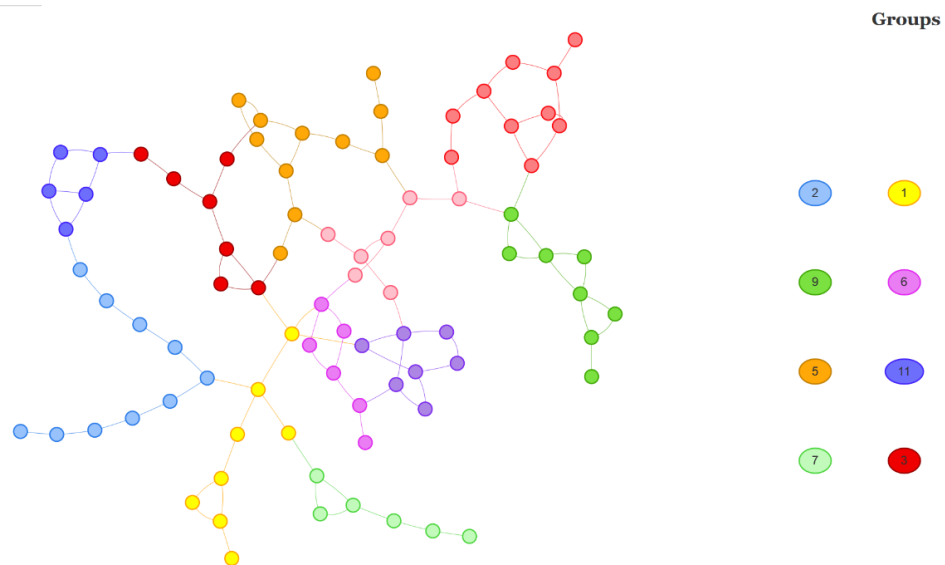


Figura 24 Detección de comunidad del proceso subasta de autos nuevos y usados con algoritmo Fast Greedy

3.7.3.2. Comunidades para el proceso subasta de autos siniestrados usando el algoritmo Fast-Greedy.

Al aplicar el algoritmo de detección Fast Greedy para el proceso subasta de autos siniestrados se encontraron 7 comunidades, tal como se puede observar en la Figura 25.

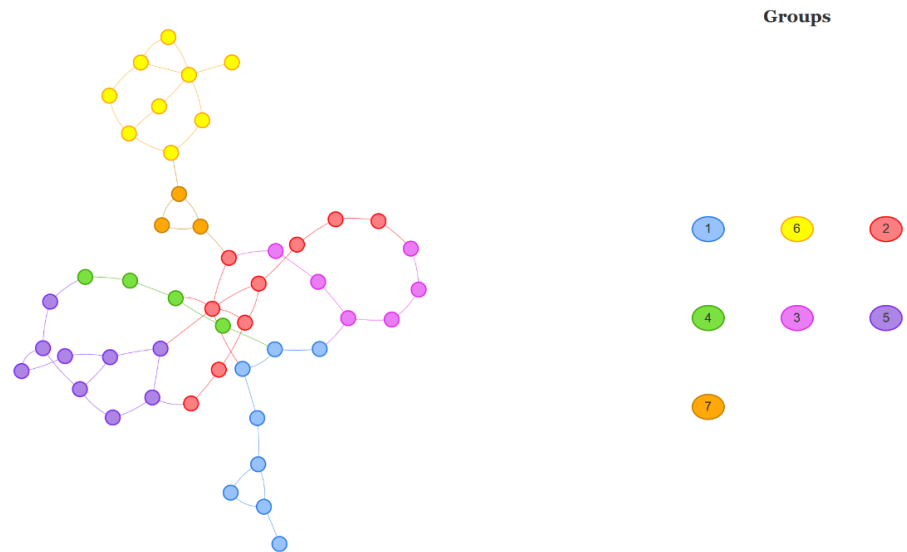


Figura 25 Detección de comunidad del proceso subasta de autos siniestrados con algoritmo Fast Greedy

3.7.3.3. Comunidades para el proceso subasta de autos nuevos y usados usando el algoritmo Infomap para una red dirigida.

Al aplicar el algoritmo de detección Infomap para el proceso subasta de autos nuevos y usados como una red dirigida se encontraron 14 comunidades, tal como se puede observar en la Figura 26.

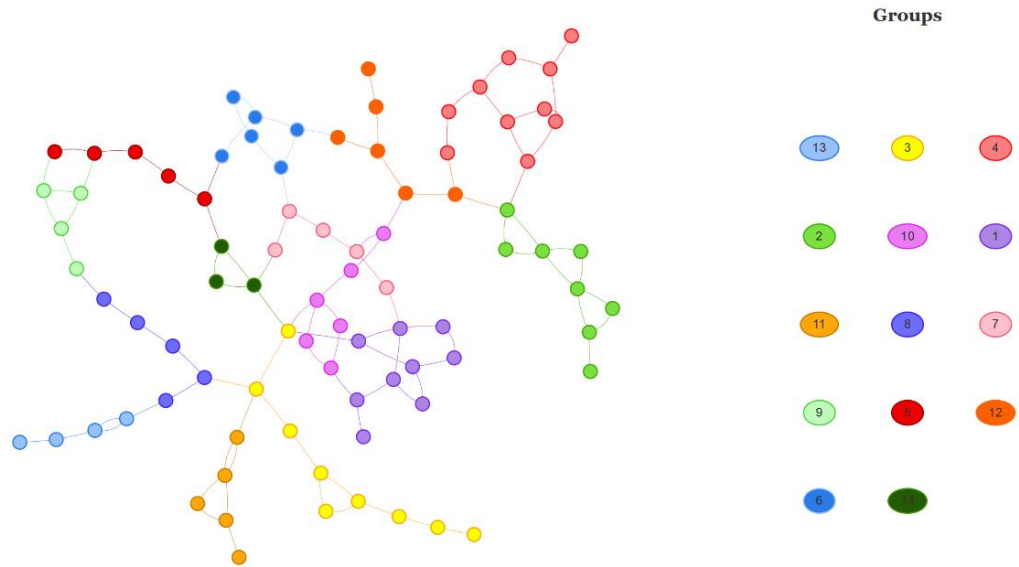


Figura 26 Detección de comunidad del proceso subasta de autos nuevos y usados con algoritmo Infomap (red dirigida)

3.7.3.4. Comunidades para el proceso subasta de autos nuevos y usados usando el algoritmo Infomap para un red no dirigida.

Al aplicar el algoritmo de detección Infomap para el proceso subasta de autos nuevos y usados como una red no dirigida se encontraron 15 comunidades, tal como se puede ver en la Figura 27.

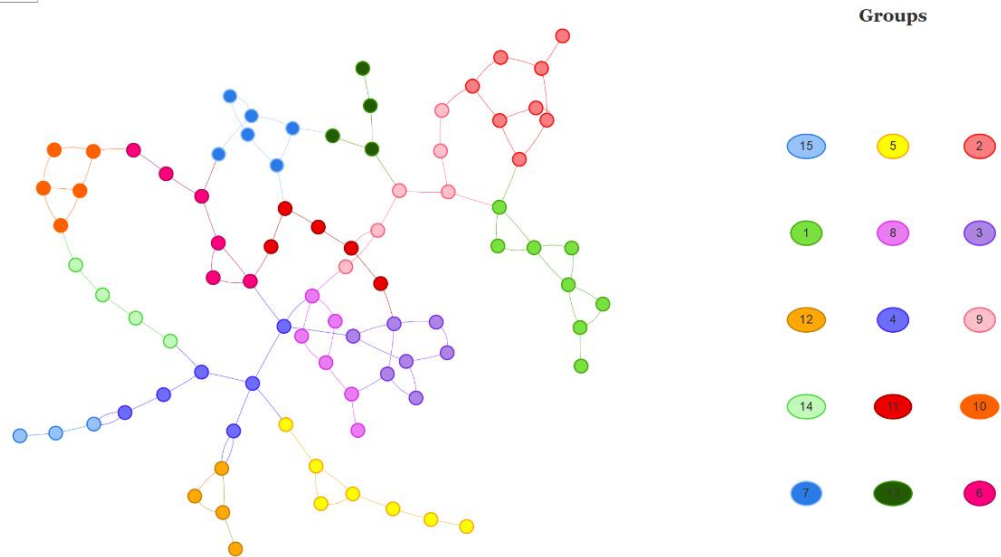


Figura 27 Detección de comunidad del proceso subasta de autos nuevos y usados con algoritmo Infomap (red no dirigida)

3.7.3.5. Comunidades para el proceso subasta de autos siniestrados usando algoritmo Infomap para una red dirigida.

Al aplicar el algoritmo de detección Infomap para el proceso subasta de autos siniestrados como una red dirigida se encontraron 14 comunidades, tal como se puede ver en la Figura 28.

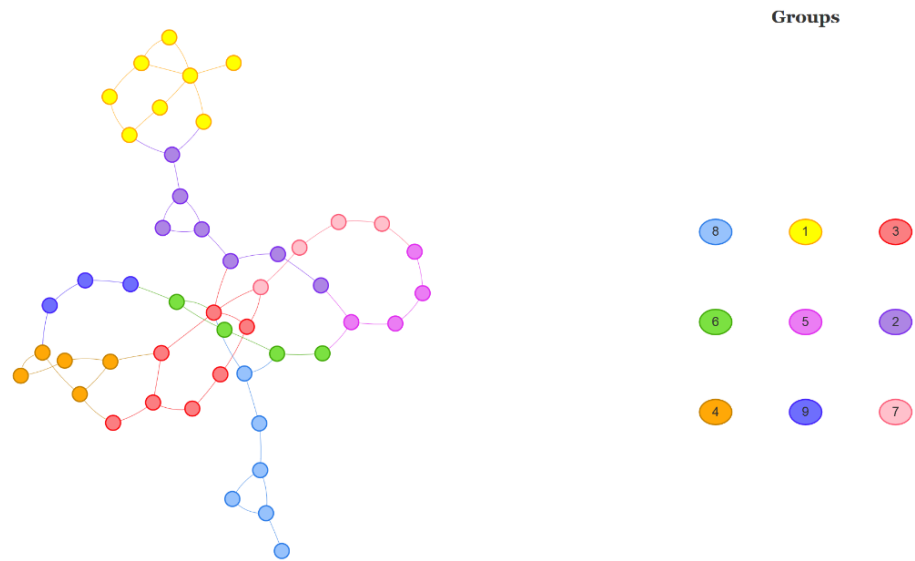


Figura 28 Detección de comunidad del proceso subasta de autos siniestrados con algoritmo Infomap (red dirigida)

3.7.3.6. Comunidades para el proceso subasta de autos siniestrados usando el algoritmo Infomap para un red no dirigida.

Al aplicar el algoritmo de detección Infomap para el proceso subasta de autos siniestrados como una red no dirigida se encontraron 9 comunidades, tal como se puede ver en la Figura 29.

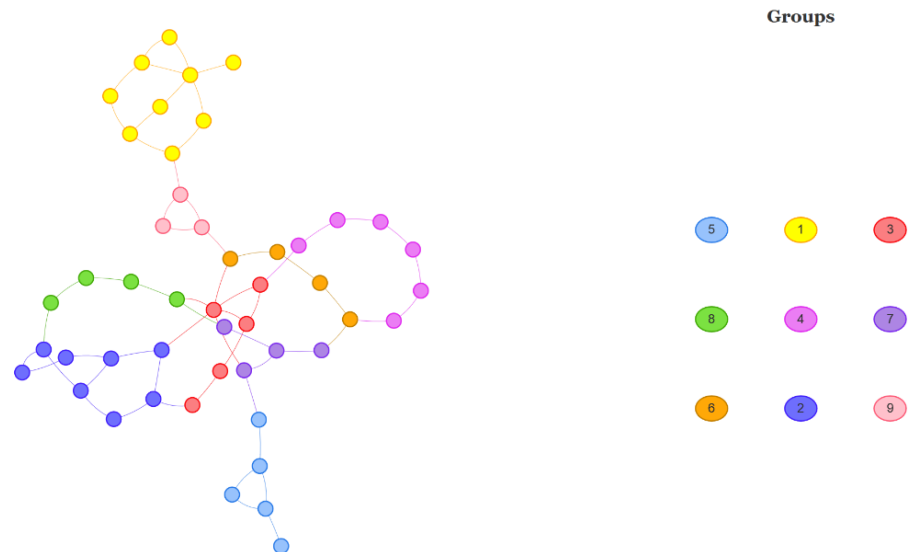


Figura 29 Detección de comunidad del proceso subasta de autos siniestrados con algoritmo Infomap (red no dirigida)

Como se puede observar en la Tabla 8 la detección de comunidades por el algoritmo Infomap nos da un valor similar a la cantidad de carriles. Debido a que únicamente se están analizando dos procesos unificados no podemos concluir que es el mejor algoritmo para la detección de comunidades.

Tabla 8 Comunidades para los procesos interconectados.

Nombre del Proceso.	Carriles	Fast-Greedy	Infomap (red no dirigida)	Infomap (red dirigida)
Proceso subasta de autos nuevos y usados.	9	11	14	15
Proceso subasta de autos siniestrados.	9	7	9	9

Capítulo 4. Resultados.

Las métricas obtenidas en la sección anterior nos permiten realizar análisis sobre la red.

4.1. Resultados de la detección de comunidades.

Aunque se aplicaron diferentes algoritmos para la detección de comunidades, no se logró llegar a un consenso sobre cual algoritmo ofrece una detección más fiable y realista. A pesar de esto y tomando los carriles como una “comunidad” para los diferentes nodos, el algoritmo que presento mejores resultados fue Infomap sobre redes no dirigidas. Debido a que no se realizó el levantamiento del proceso y no contamos con la retroalimentación de un especialista en los procesos de la empresa es difícil poder aseverar si las comunidades generadas puedan presentar una mejora para la ejecución del flujo de proceso. Recordemos que las comunidades de los nodos permitirían a los encargados de proceso delegar o establecer procesos de control a actores o áreas adyacentes al nodo y basándose en la comunidad a la que fue asignado. Adicionalmente, la detección de comunidades sería útil dichas comunidades pueden servir como base a las empresas u organizaciones que se encuentren levantando los procesos.

4.2. Resultados del análisis de robustez.

Debido a la naturaleza lineal de los procesos, su robustez es baja. Principalmente porque los nodos mantienen un bajo grado de conectividad. En las siguientes secciones podemos encontrar información sobre la robustez de cada uno de los procesos.

4.2.1. Proceso subasta autos nuevos y usados.

Los resultados para el proceso subasta de autos nuevos y usados nos muestra que es altamente vulnerable frente ataques o errores. El proceso muestra una leve resistencia para una pérdida reducida de nodos en ataques basados en intermediación, esta resistencia se reduce drásticamente cuando la pérdida de nodos se aproxima al 20% del total de nodos en la red. También podemos observar una pérdida de conectividad constante frente a un ataque aleatorio. El proceso es más vulnerable frente a ataques en cascada y basados en conectividad respectivamente. Esto se puede observar más a detalle en la Figura 30.

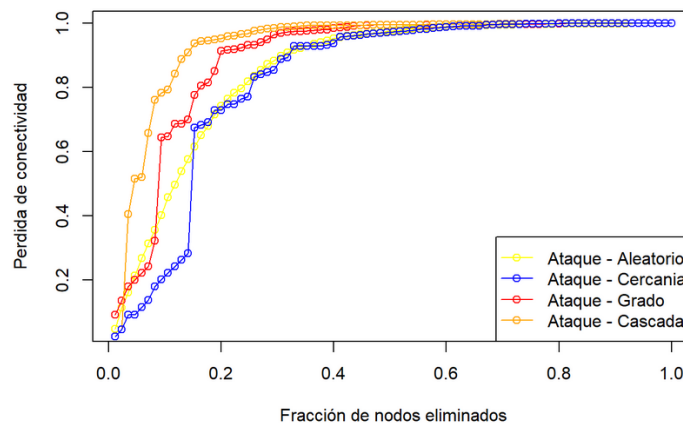


Figura 30 Pérdida de conectividad del proceso subasta de autos nuevos y usados frente a diferentes patrones de ataque.

4.2.2. Proceso subasta de autos siniestrados.

Los resultados para el proceso subasta de autos siniestrados nos muestra que es altamente vulnerable frente ataques o errores. El proceso muestra una pérdida de conectividad constante frente a ataques aleatorios y una debilidad alta frente a cualquier otro tipo de ataque perdiendo alrededor de la mitad de su conectividad al eliminar una quinta parte de sus nodos. Esto se puede observar más a detalle en la Figura 31.

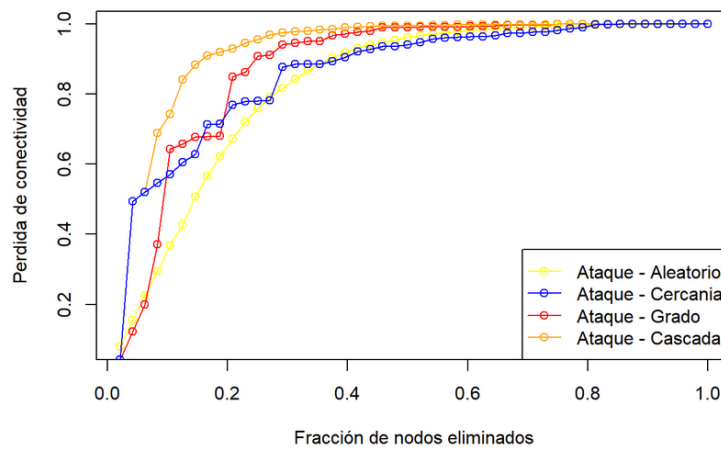


Figura 31 Pérdida de conectividad del proceso subasta de autos siniestrados frente a diferentes patrones de ataque.

Capítulo 5. Conclusiones y Recomendaciones

5.1. Conclusiones

Una vez realizado todo el proceso de investigación se concluye que el análisis de procesos de negocio usando redes complejas es viable y proporciona información detallada de características como comunidades y robustez, mismas que no son tomadas en cuenta cuando se usa únicamente BPMN.

En el caso de la detección de comunidades, nos hemos enfocado en encontrar comunidades de nodos que estén agrupados considerando las métricas de sus enlaces. Existen varias estrategias para la detección de comunidades, pero su gran mayoría usan redes no dirigidas. Debido a que los procesos de negocio generan redes dirigidas se seleccionó el algoritmo Infomap como la mejor opción para la detección de comunidades. Este análisis se puede transformar en una herramienta que permita a los encargados del proceso optimizar la asignación de funciones y responsabilidades a cada una de las áreas basándose en la pertenencia de los nodos a las diferentes comunidades. La detección de comunidades se puede enfocar dependiendo de la información que se requiera obtener sobre la topología de la red, esta información será diferente para cada empresa, proyecto u organización.

La robustez de cada nodo facilita la asignación y eliminación de recursos determinando la importancia de cada nodo para que el flujo del proceso no sea interrumpido. Así mismo permite determinar la vulnerabilidad que tiene cada nodo frente a errores o ataques y de esta manera poder definir flujos de proceso alternativos o carril de contingencia para nodos de vital importancia.

5.2. Recomendaciones

Es recomendable que las empresas u organizaciones no realicen una optimización de procesos o asignación de recursos de una manera empírica sobre todo cuando llevan una gran cantidad de procesos como las multinacionales. Al no poder visualizar de forma óptima las distintas actividades y flujos de una empresa es susceptible a repetir procesos que incurra en gastos innecesarios para la empresa o no se pueda distinguir un canal directo que se más eficaz.

Se debería formalizar el análisis de los procesos con herramientas como las presentadas en esta investigación ya que se podría gestionar de forma oportuna y eficaz los procesos de una empresa, es decir, la inversión de capital en proyectos puede generar un ahorro de tiempo, dinero y esfuerzo para las empresas.

Referencias

- Chavarriaga, J. E. (2017). Modelo programable para la serialización y evaluación de modelos heterogéneos en clientes web (Tesis Doctoral). Madrid, España: UAM. Departamento de Ingeniería Informática. Obtenido de <https://repositorio.uam.es/handle/10486/680505>
- ABPMP -- Association of Business Process Management Professionals. (2013). *BPM CBOK*.
- Barabási, A.-L. (2002). Statistical mechanics of complex networks. *REVIEWS OF MODERN PHYSICS, VOLUME 74*, 48-85.
- BATAGELJ, V., MRVAR, A., & DE NOOY, W. (2005). *Exploratory Network Analysis*. Cambridge University Press.
- Bizagi. (10 de 05 de 2020). *Basic publishing and export*. Obtenido de Bizagi Modeler and Modeler Services documentation: http://help.bizagi.com/process-modeler/en/index.html?where_to_share2.htm
- DataStorm. (20 de 06 de 2020). *visNetwork, an R package for interactive network visualization*. Obtenido de visNetwork: <https://datastorm-open.github.io/visNetwork/>
- Emmons, S., Kobourov, S., Gallant, M., & Börner, K. (15 de 06 de 2020). *Analysis of Network Clustering Algorithms and Cluster Quality Metrics at Scale*. Obtenido de US National Library of Medicine: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4938516/>
- Freeman, L. (1978). Centrality in social networks: Conceptual clarification. *Social Networks 1*, 215-239.
- Freund, J., Rücker, B., & Hitpass, B. (2014). *BPMN 2.0 Manual de Referencia y Guía Práctica*. Santiago de Chile: Dimacofi.
- Lhomme, S. (15 de 06 de 2020). *Network Strengths and Weaknesses Analysis*. Obtenido de The Comprehensive R Archive Network: <https://cran.r-project.org/web/packages/NetSwan/NetSwan.pdf>
- Network theory. (2019). *La teoría de redes - Network theory*. Obtenido de https://es.qwe.wiki/wiki/Network_theory
- Newman, M. (2003). The Structure and Function of Complex Networks. *Vol. 45, No. 2*, 167–256.

- Newman, M. E. (2010). *Networks an Introduction*. New York: Oxford University Press.
- Newman, M. E., & Girvan, M. (2004). Finding and evaluating community structure in networks. *Phys Rev E Stat Nonlin Soft Matter Phys.*, 64 - 70.
- Object Management Group. (15 de 5 de 2020). *Business Process Model and Notation Standard*. Obtenido de Object Management Group Business Process Model and Notation : <http://www.omg.org/spec/BPMN/2.0>
- Pantoja, O. (2012). *Identificación y Modelado de Procesos en Empresa Leilocar*. Porto: Faculdade de Engenharia da Universidade do Porto.
- Pereira, C., Maldonado, D., & Nascimento, M. (2016). A consensus graph clustering algorithm for directed networks. *Expert Systems with Applications*, 121-135.
- PSF. (10 de 03 de 2020). *What is Python? Executive Summary*. Obtenido de Python: <https://www.python.org/doc/essays/blurb/>
- R Foundation. (3 de 5 de 2020). *Introduction to R*. Obtenido de r-project: <https://www.r-project.org/about.html>
- Sadiq, W., & Orlowska, M. (s.f.). *Analizing Process Models Using Graph Reduction Techniques*. Department of Computer Science & Electrical Engineering.
- Salazar López, B. (12 de junio de 2019). *Ingeniería Industrial*. Obtenido de <https://www.ingenieriaindustrialonline.com/investigacion-de-operaciones/teoria-de-redes/>
- White, S. A. (10 de 04 de 2020). *PROCESS MODELING NOTATIONS AND WORKFLOW PATTERNS*. Obtenido de workflowpatterns: http://www.workflowpatterns.com/vendors/documentation/BPMN_wfh.pdf
- White, S. A. (15 de 2 de 2020). *XPDL and BPMN*. Obtenido de Business Process Model & Notation™ (BPMN™) | Object Maangement Group: https://www.omg.org/bpmn/Documents/XPDL_BPMN.pdf
- Workflow Management Coalition. (10 de 05 de 2020). *XPDL Support and Resources*. Obtenido de XPDL Workflow Management Coalition: <https://www.wfmc.org/standards/xpdl>

Anexos

Anexo 1 Código Python para obtener carriles

```
def carril(argsoup):
    global cmap
    tmppool = pool(argsoup)
    tmpcarril = tmppool.Carril.findAll("Lane")
    cmap = get_cmap(len(tmpcarril))
    carrillst = []
    tmpi = 0
    for i in tmpcarril:
        ilst = [i["Name"], i["Id"], i["ParentPool"], *get_position(i), cmap(tmpi)]
        carrillst.append(ilst)
        tmpi += 1
    return carrillst
```

Anexo 2 Código Python para obtener actividades

```
def activity_type(argactivity):
    tp = []
    if argactivity:
        if argactivity.find("Implementation"):
            tp = ["Activity", argactivity.Implementation.Task.findChild().name]
        elif argactivity.find("Route"):
            tp = ["Gateway", "Gateway"]
        elif argactivity.find("Event"):
            tp = ["Event", argactivity.Event.findChild().name]
    return tp

# Returns a multidimensional list with all the activities.
def activities(argsoup):
    tmpwp = workflow_process(argsoup)
    tmpactivities = tmpwp.Activities.findAll("Activity")
    activitylst = []
    for i in tmpactivities:
        ilst = [i["Name"], i["Id"], *activity_type(i), *get_position(i)]
        activitylst.append(ilst)
    return activitylst
```

Anexo 3 Código Python para obtener artefactos.

```
def object_type(argobject):
    ot = []
    if argobject.find("Object"):
        ot = ["DataObject", argobject.Object.findChild().name]
    return ot

def data_objects(argsoup):
    tmpwp = workflow_process(argsoup)
    objectsls = []
    if tmpwp.find("DataObjects"):
        tmpdo = tmpwp.DataObjects.findAll("DataObject")
        for i in tmpdo:
            ilst = [i["Name"], i["Id"], *object_type(i), *get_position(i)]
            objectsls.append(ilst)
    return objectsls
```

Anexo 4 Código Python para obtener objetos.

```
def object_type(argobject):
    ot = []
    if argobject.find("Object"):
        ot = ["DataObject", argobject.Object.findChild().name]
    return ot

# Returns a multidimensional list with all the dataobjects (artifacts).
def data_objects(argsoup):
    tmpwp = workflow_process(argsoup)
    objectsls = []
    if tmpwp.find("DataObjects"):
        tmpdo = tmpwp.DataObjects.findAll("DataObject")
        for i in tmpdo:
            ilst = [i["Name"], i["Id"], *object_type(i), *get_position(i)]
            objectsls.append(ilst)
    return objectsls
```


Anexo 5 Código Python para obtener transiciones.

```
def transition_type(argtransition):
    if argtransition.find("Condition").isSelfClosing:
        tr = ["SequenceFlow", "SequenceFlow"]
    elif argtransition.Condition.findChild():
        tr = ["SequenceFlow", argtransition.Condition.findChild().name]
    else:
        tr = ["SequenceFlow", ""]
    return tr

def transitions(argsoup):
    tmpwp = workflow_process(argsoup)
    tmptr = tmpwp.Transitions.findAll("Transition")
    transitionsls = []
    for i in tmptr:
        ilst = [i["Id"], i["From"], i["To"], *transition_type(i)]
        transitionsls.append(ilst)
    return transitionsls
```

Anexo 6 Código Python para obtener asociaciones.

```
def data_associations(argsoup):
    tmpwp = workflow_process(argsoup)
    tmpda = tmpwp.DataAssociations.findAll("DataAssociation")
    dals = []
    for i in tmpda:
        ilst = [i["Id"], i["From"], i["To"], "DataAssociation", "DataAssociation"]
        dals.append(ilst)
    return dals
```

Anexo 7 Código Python para consolidar información.

```
def consolidate_edges(argdata):
    tmpedges = [*transitions(argdata), *associations(argdata)]
    return tmpedges

def consolidate_nodes(argdata):
    tmpnodes = [*data_store(argdata), *data_objects(argdata), *activities(argdata)]
    tmpedges = consolidate_edges(argdata)
    for n in tmpnodes:
        n.append(sum(x[1].count(str(n[1])) for x in tmpedges))
        n.append(sum(x[2].count(str(n[1])) for x in tmpedges))
    return tmpnodes
```

Anexo 8 Código Python para creación de la red

```
def graph_creator(argdfn, argdfe):  
    g = nx.from_pandas_edgelist(argdfe, source="source", target="target",  
create_using=nx.DiGraph)  
    node_attr = argdfn.to_dict('Id')  
    nx.set_node_attributes(g, node_attr)  
    return g
```

Anexo 9 Código Python para graficar red.

```
def graph_grapher(argdfn, argdfe, arglane, argname="Process", layout=0, labels=False):
    # create graph
    g = graph_creator(argdfn, argdfe)
    pos, lbl = lyt_switcher(layout, g)
    fig, ax = plt.subplots(figsize=fig_size(g))
    # nx.set_edge_attributes(g, 'betweenness', bb)
    nx.draw_networkx_edges(g, pos, connectionstyle='arc3, rad=0.1')
    draw_nodes_detail(g, pos)
    # Change position for labels
    pos_higher = change_pos(pos, 0.03)
    # Draw labels
    if labels:
        nx.draw_networkx_labels(g, pos_higher, labels=nx.get_node_attributes(g, 'Name'))
    # Custom legends based on carril
    proxy_legend = []
    for lane in arglane:
        proxy_legend.append(mpatches.Patch(color=lane[9], label=lane[0]))
    carril_legend = plt.legend(handles=proxy_legend, loc="upper left", title="Carril",
fancybox=True)
    plt.gca().add_artist(carril_legend)
    # Custom legend for shapes
    shapes_legend = []
    for t in argdfn["Type"].unique():
        shapes_legend.append(Line2D([0], [0], linewidth=0, marker=shapes.get(t, "o"),
markedgecolor="black",
                                markerfacecolor="white", label=t))
    plt.legend(handles=shapes_legend, loc="upper right", title="Shapes", fancybox=True)
    # Set title
    t = '{}'.format(argname)
    plt.title(t)
    plt.show()
    adjacency_matrix(argdfe, t)
    save_image(fig, t + " " + lbl)
```

Anexo 10 Código Python para generar datos carriles.

```
# Create lanes dataframe
dfl = pd.DataFrame(arglane, columns=["lane_name", "Id", "ParentPool", "height", "width",
    "x", "y",
                                "center_x", "center_y", "color"])
dfe = argdfe.groupby(['source_lane_id',
    'target_lane_id']).size().reset_index().rename(columns={0: 'count'})
dfn = argdfn.groupby(['lane_name']).size().reset_index().rename(columns={0: 'count'})
dfn = pd.merge(dfl, dfn, on=['lane_name'])
dfn.set_index("Id", inplace=True)
dfe.rename(columns={'source_lane_id': 'source', "target_lane_id": "target"}, inplace=True)
```

Anexo 11 Código Python para graficar red por carriles.

```
def lanes_grapher(argdfn, argdfe, arglane, argname="Process", layout=0, labels=False):
    # create lanes graph
    g = nx.from_pandas_edgelist(dfe, source="source", target="target", edge_attr=True,
create_using=nx.DiGraph)
    node_attr = dfn.to_dict('Id')
    nx.set_node_attributes(g, node_attr)
    dfn.to_csv('{}_nodes_lanes.csv'.format(argname), index=True)
    dfe.to_csv('{}_edges_lanes.csv'.format(argname), index=True)
    # print(nx.get_node_attributes(g, "color")["3e263fb9-ee2f-4e07-90da-deb117ec9951"])
    pos, lbl = lyt_switcher(layout, g)
    fig, ax = plt.subplots(figsize=fig_size(g))
    node_colors = nx.get_node_attributes(g, 'color')
    # node_sizes = nx.get_node_attributes(g, 'count')
    edge_colors = nx.get_edge_attributes(g, 'color')
    edge_widths = nx.get_edge_attributes(g, 'count')
    d = dict(g.degree)
    nx.draw_networkx_edges(g, pos=pos,
                           width=list(edge_widths.values()),
                           connectionstyle='arc3, rad=0.2',
                           # edge_color=list(edge_colors.values()),
                           arrows=True, arrowsize=30,
                           with_labels=False,
                           ax=ax)
    nx.draw_networkx_nodes(g, pos=pos,
                           node_size=[v * 100 for v in d.values()], ax=ax,
                           alpha=1, arrows=True,
                           arrowsize=20, width=2, node_color='w'
                           )
    nx.draw_networkx_nodes(g, pos=pos,
                           node_color=list(node_colors.values()),
                           node_size=[v * 100 for v in d.values()], ax=ax,
                           alpha=1, arrows=True,
                           arrowsize=20, width=0
                           )
    pos_higher = change_pos(pos, 0.01)
    if labels:
        nx.draw_networkx_labels(g, pos_higher, labels=nx.get_node_attributes(g, 'Name'))
    proxy_legend = []
    for lane in arglane:
        proxy_legend.append(mpatches.Patch(color=lane[9], label=lane[0]))
    plt.legend(handles=proxy_legend, loc="upper left", title="Lanes", fancybox=True)
    # Set title
    t = 'Lanes {}'.format(argname)
    plt.title(t)
    plt.show()
    adjacency_matrix(dfe, t)
    save_image(fig, t + " " + lbl)
```

Anexo 12 Código R para la lectura información de nodos y enlaces.

```
process <- reactive(  
  {input$dataset}  
)  
dfnodes <- reactive(  
  read.csv(file = paste0(process(), "_nodes.csv"), head = T,  
    fileEncoding = "UTF-8")  
)  
dfedges <- reactive(  
  read.csv(file = paste0(process(), "_edges.csv"), head = T,  
    fileEncoding = "UTF-8")  
)
```


Anexo 13 Código R para la creación de redes con paquete igraph.

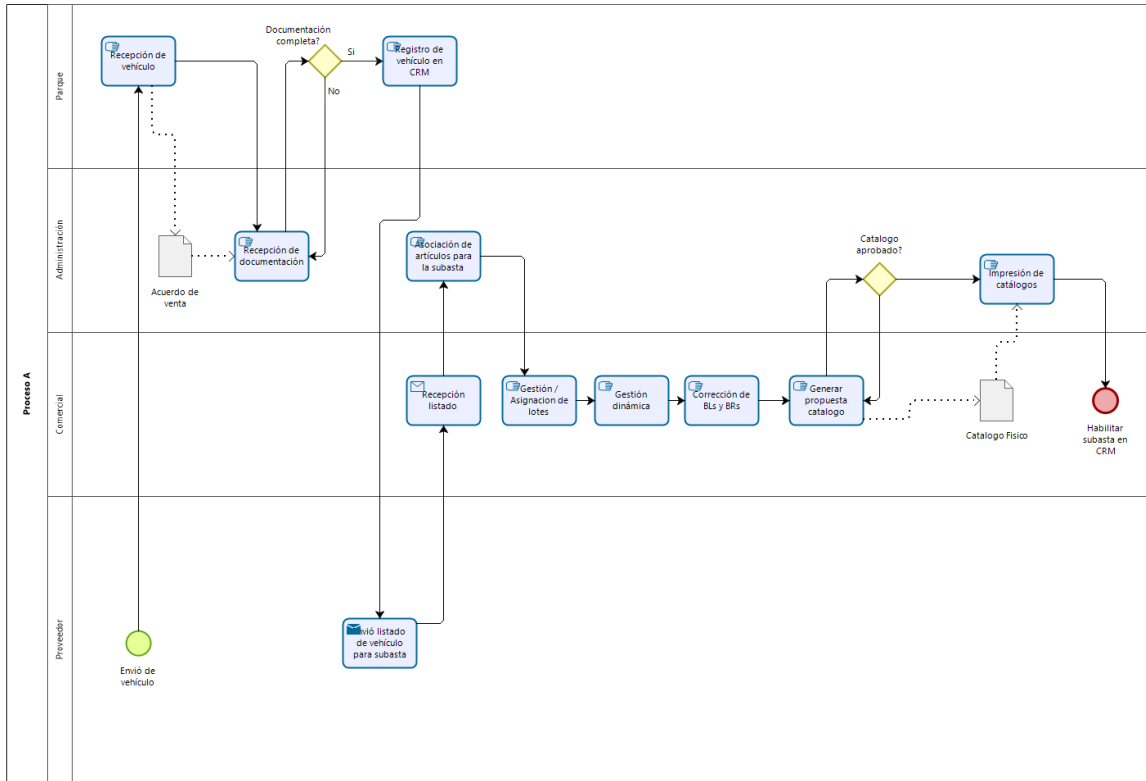
```
dnetwork <- reactive(  
  igraph::graph.data.frame(nedges(), directed = T, vertices = nnodes())  
)  
unetwork <- reactive(  
  igraph::graph.data.frame(nedges(), directed = F, vertices = nnodes())  
)
```

Anexo 14 Código R para visualización de red

```
output$network <- renderVisNetwork({
  nnodes <- nnodes()
  nedges <- nedges()
  visNetwork(nnodes, nedges, width = "100%") %>%
    visEdges(arrows = "to") %>%
    visOptions(nodesIdSelection = FALSE, selectedBy = list(variable = "group",
multiple = TRUE)) %>%
    visLegend(addNodes = list(list(label = "DataStore", shape = "database"),
                                list(label = "DataObject", shape = "star"),
                                list(label = "Activity", shape = "square"),
                                list(label = "Gateway", shape = "triangle"),
                                list(label = "Event", shape = "dot")),
              position = "right", main = "Legend", stepY = 100, width = 0.2, useGroups
= TRUE, ncol = 3) %>%
    visLayout(randomSeed = 12)
})
```

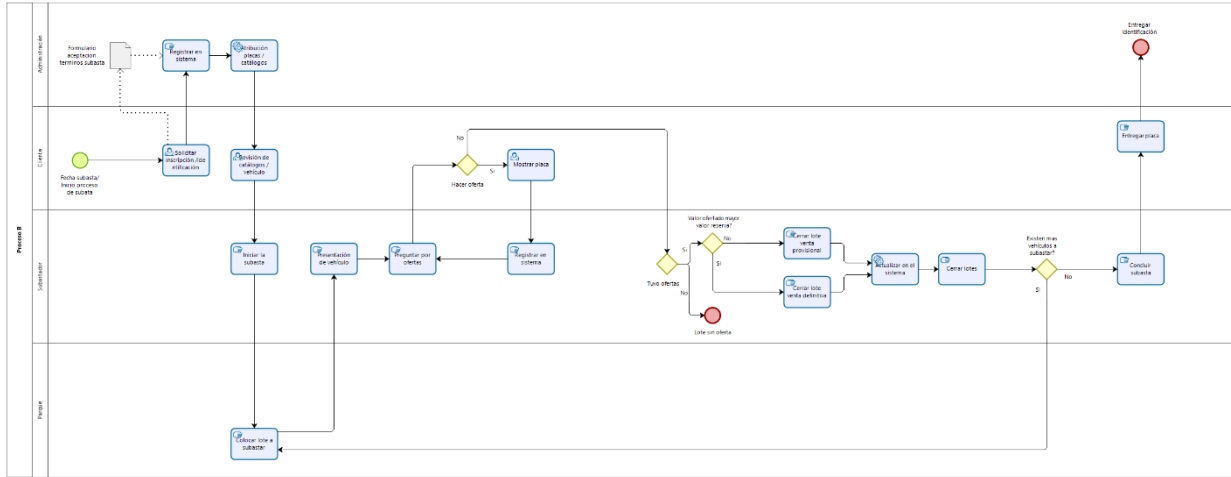
Anexo 15 Proceso A. Recepción de vehículos, la entrada del sistema y la creación de la subasta.

Adaptado de (Pantoja, 2012)



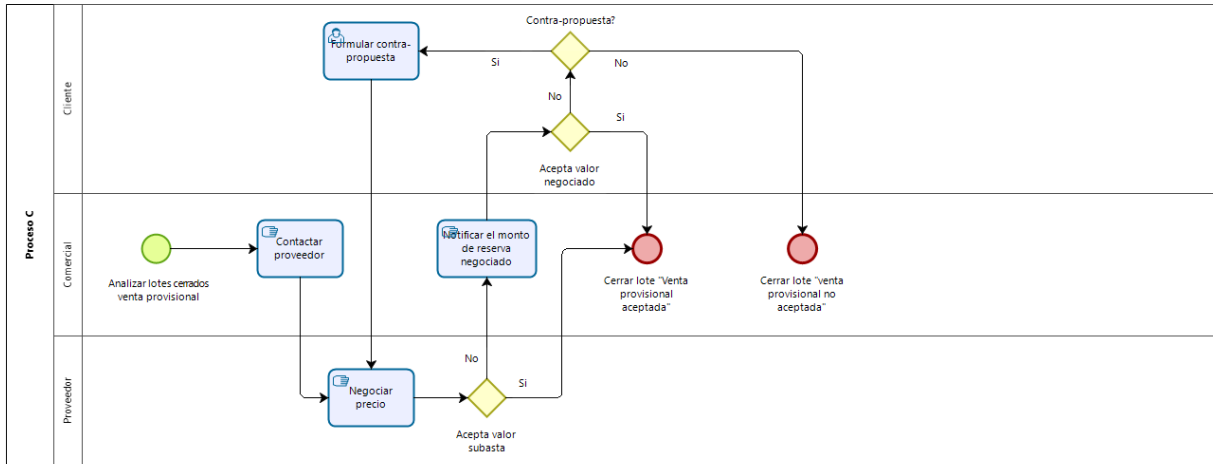
Anexo 16 Proceso B. Implementación de la subasta física.

Adaptado de (Pantoja, 2012)



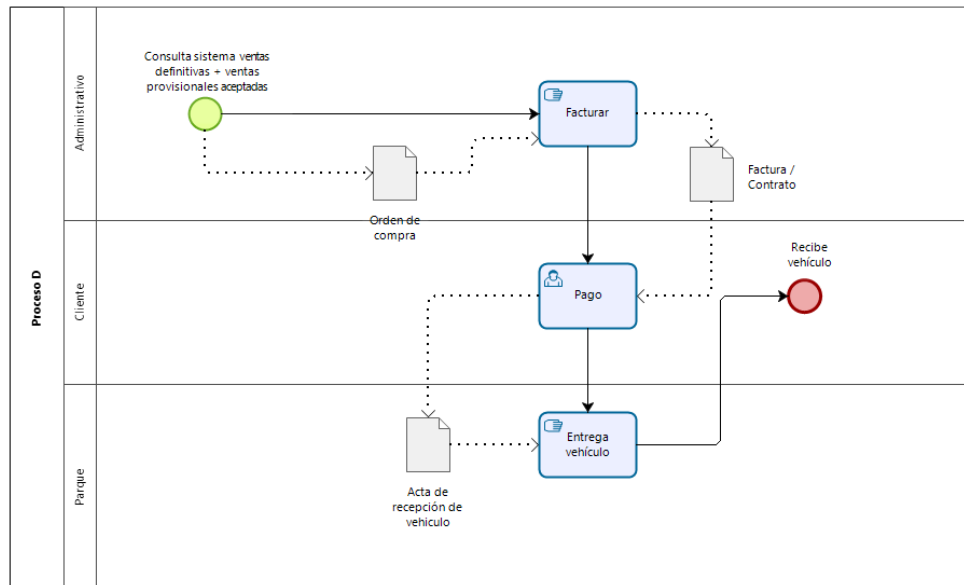
Anexo 17 Proceso C. Negociación de ventas provisionales.

Adaptado de (Pantoja, 2012)



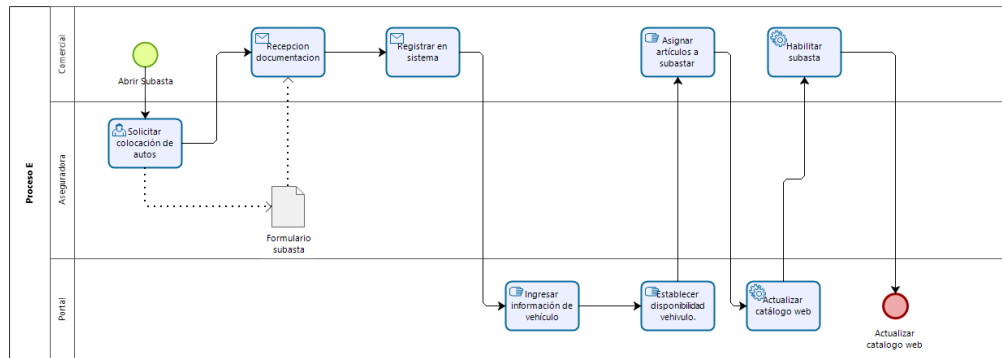
Anexo 18 Proceso D. Colección de vehículos (físicas y subastas en línea)

Adaptado de (Pantoja, 2012)



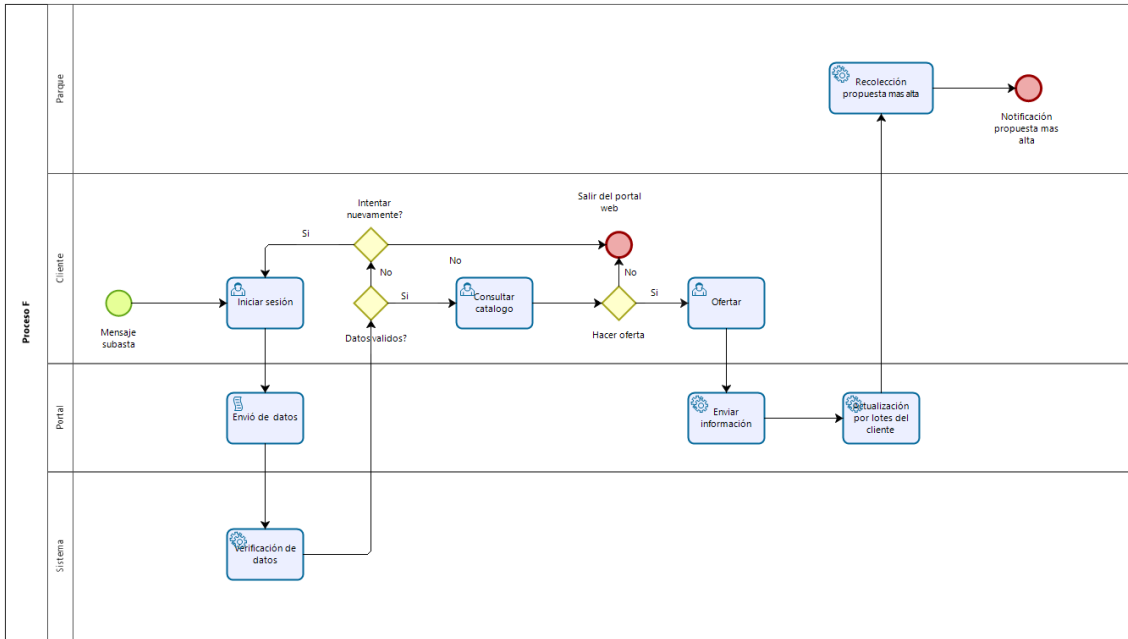
Anexo 19 Proceso E. Creación y preparación de subastas en línea.

Adaptado de (Pantoja, 2012)



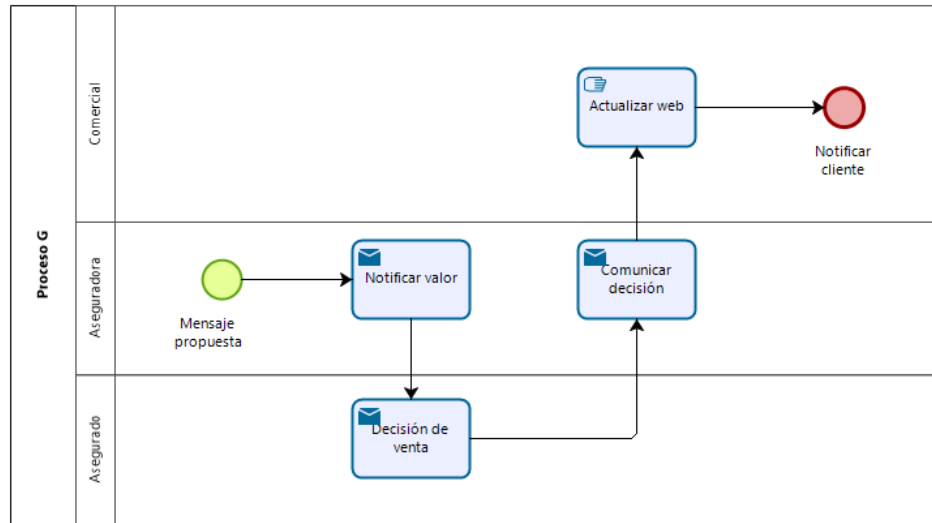
Anexo 20 Proceso F. La participación en la subasta en línea.

Adaptado de (Pantoja, 2012)



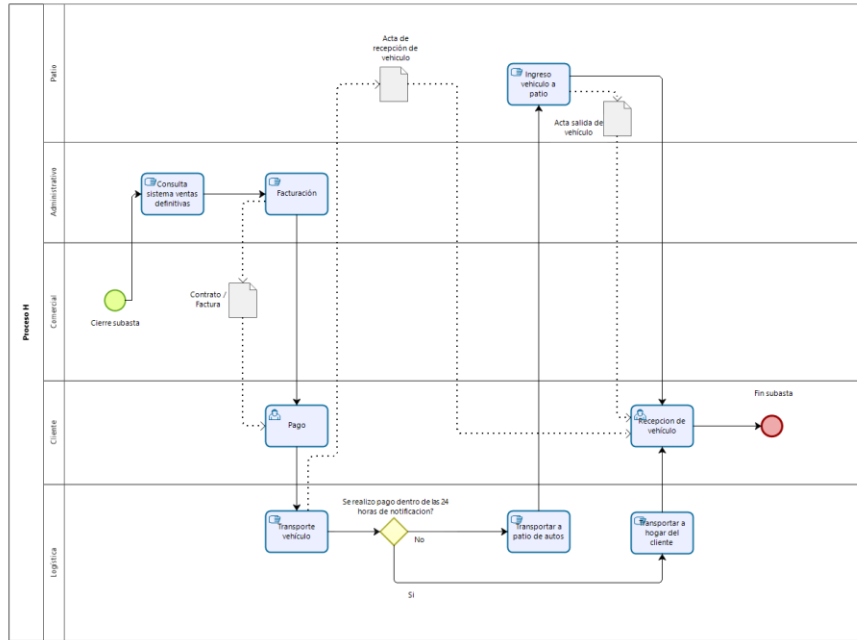
Anexo 21 Proceso G. Análisis de subastas en línea.

Adaptado de (Pantoja, 2012)



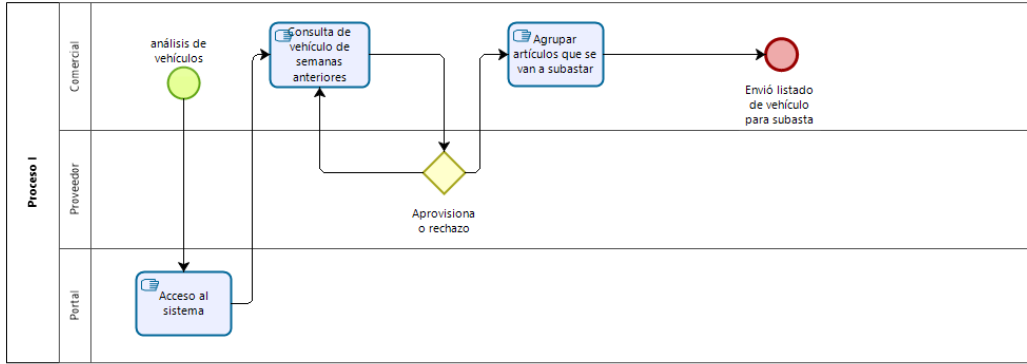
Anexo 22 Proceso H. Recolección de vehículos siniestrados para subastas online.

Adaptado de (Pantoja, 2012)



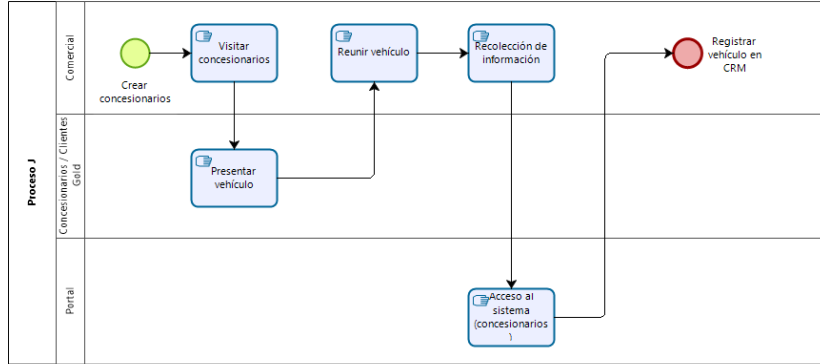
Anexo 23 Proceso I. Creación de subastas en línea concesionarios de la marca.

Adaptado de (Pantoja, 2012)

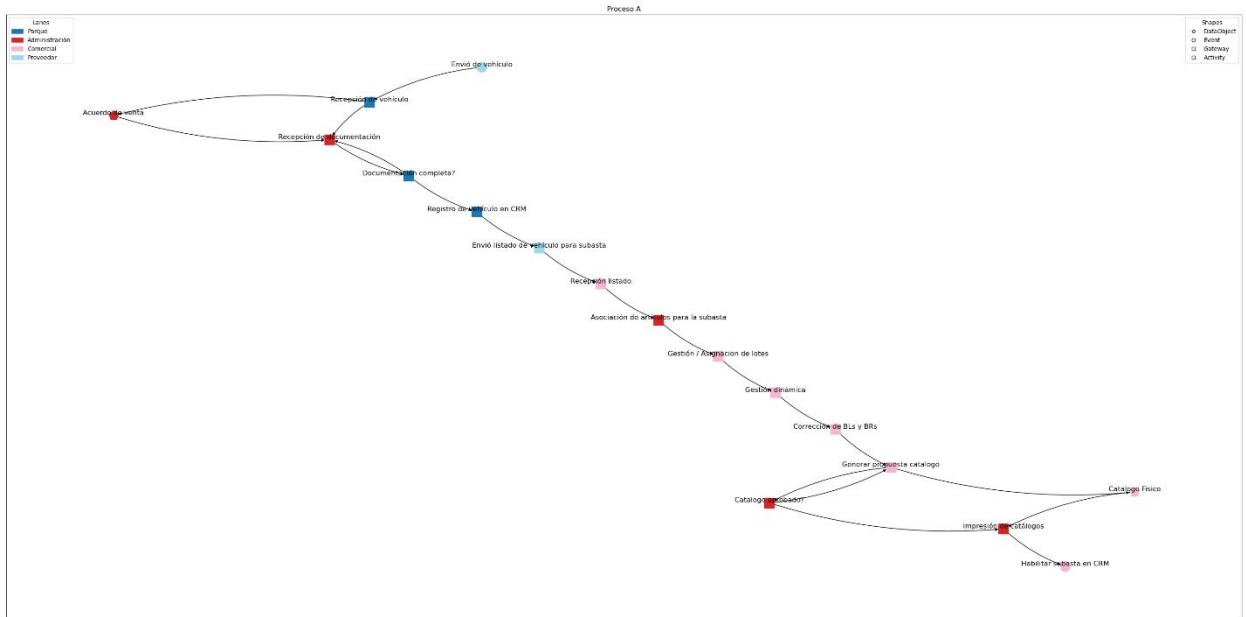


Anexo 24 Proceso J. Creación de subastas en línea distribuidores.

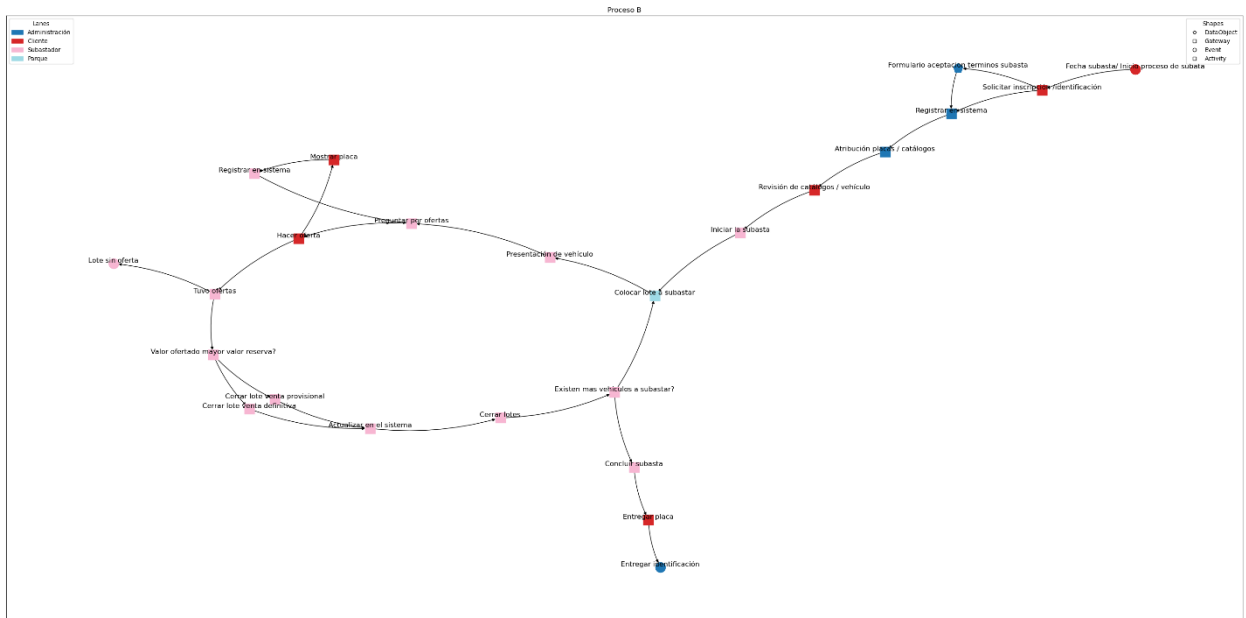
Adaptado de (Pantoja, 2012)



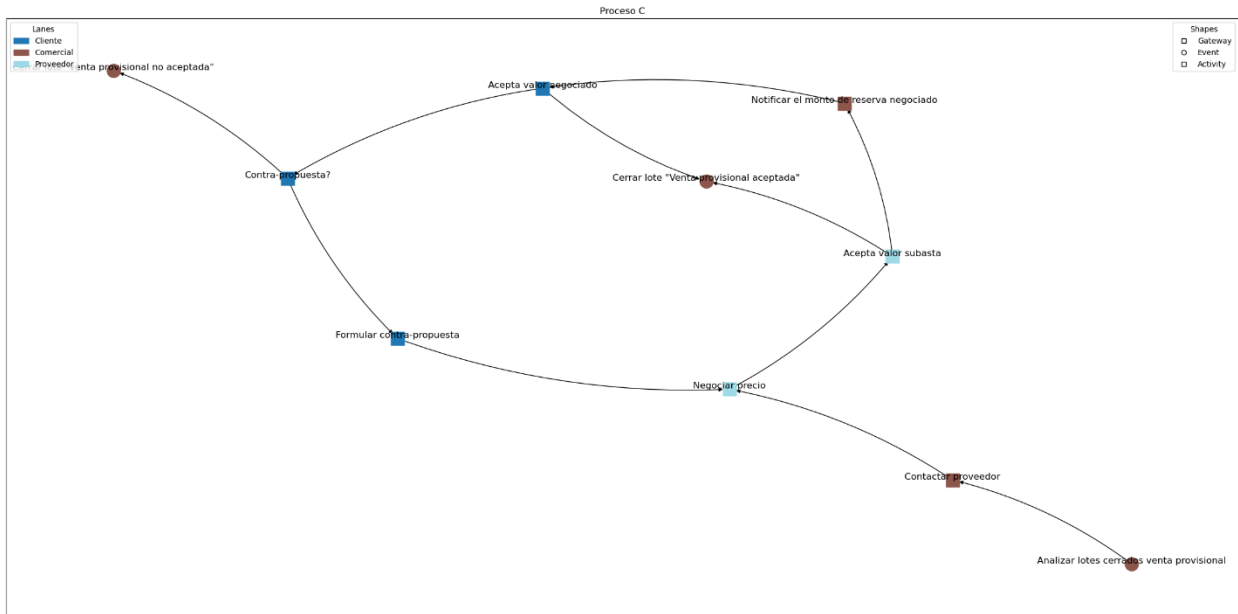
Anexo 25 Red Proceso A.



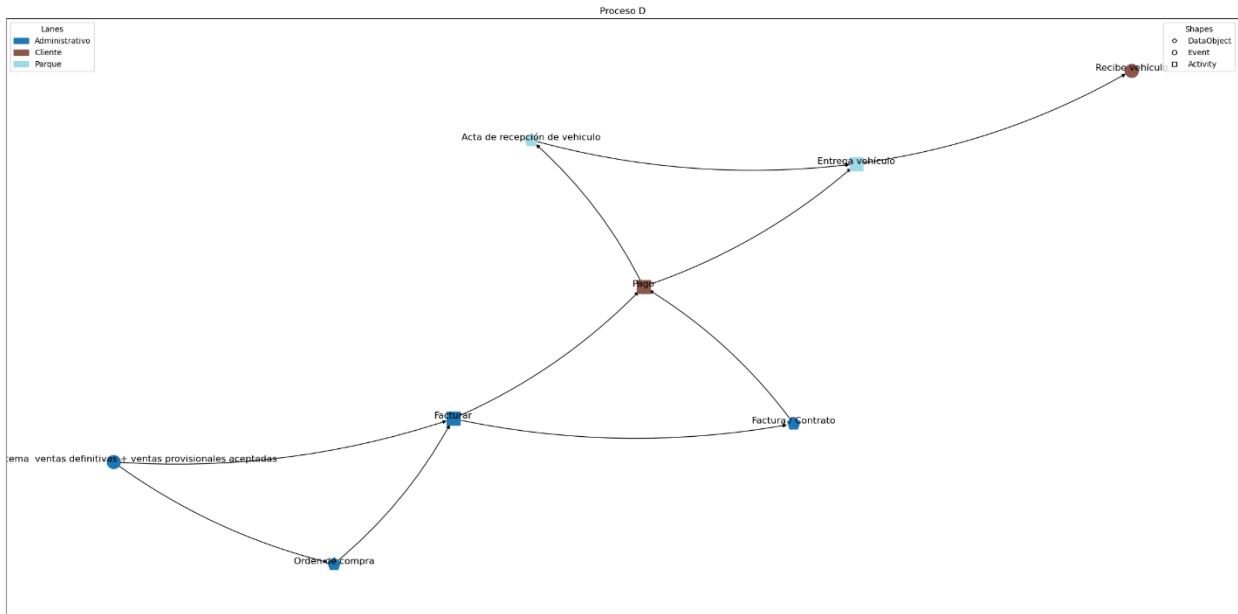
Anexo 26 Red Proceso B.



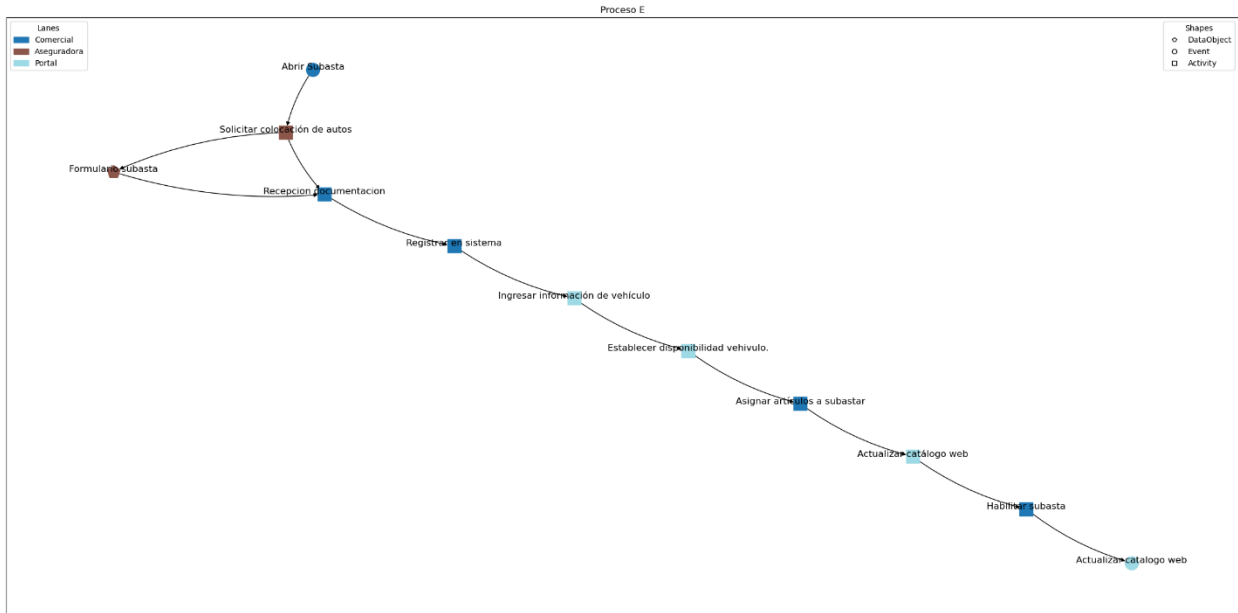
Anexo 27 Red Proceso C.



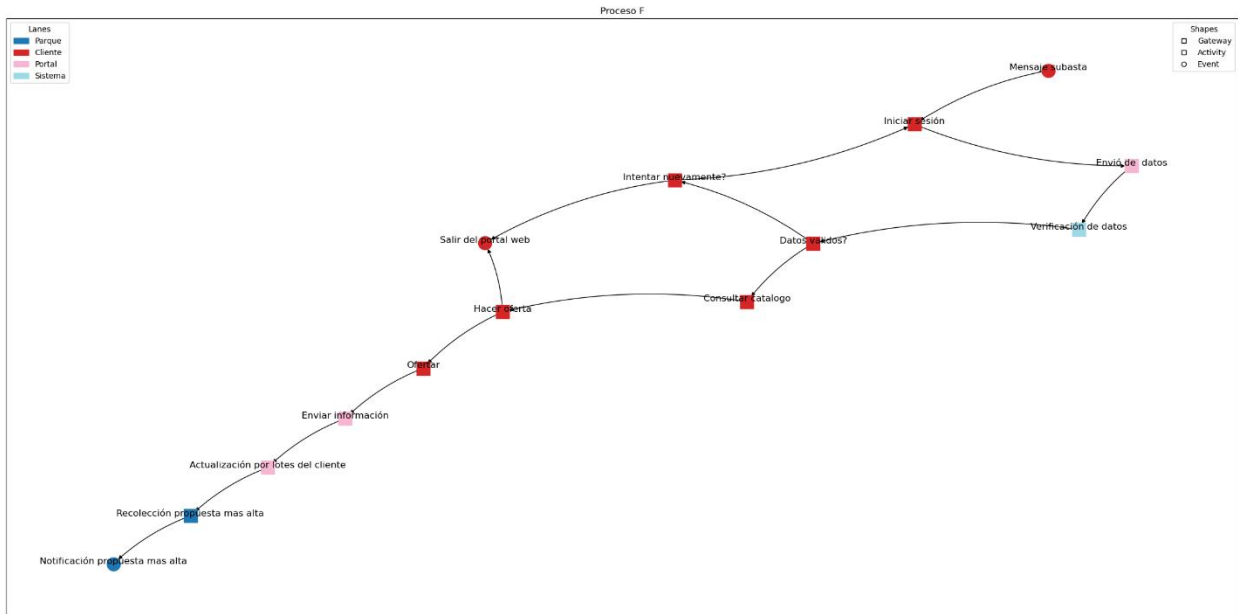
Anexo 28 Red Proceso D.



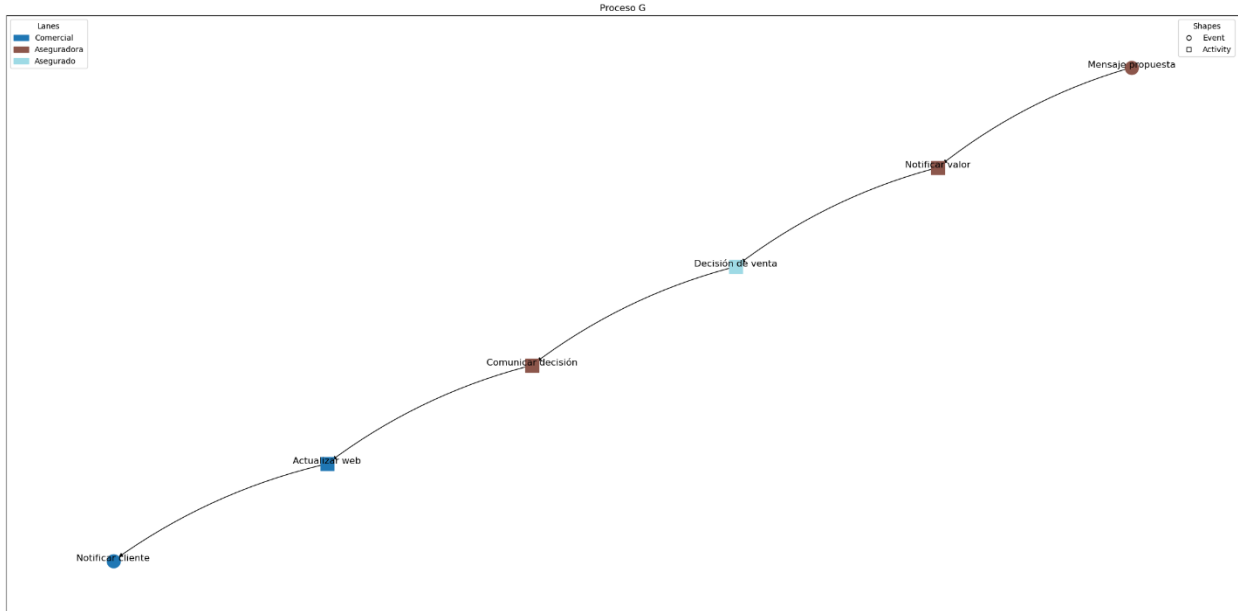
Anexo 29 Red Proceso E.



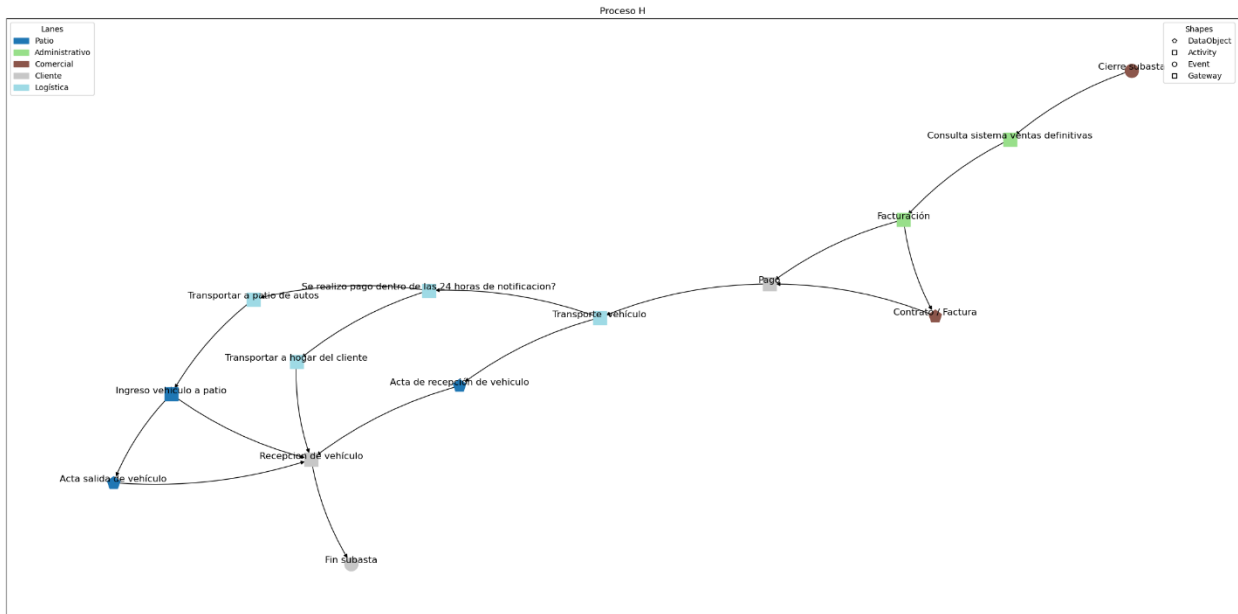
Anexo 30 Red Proceso F.



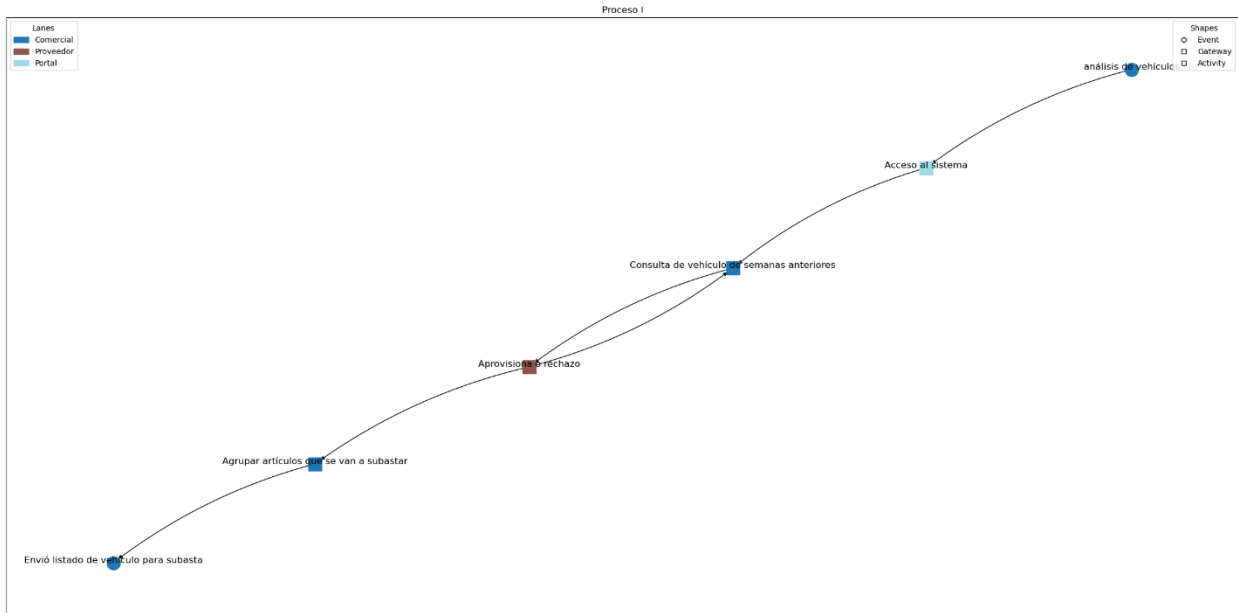
Anexo 31 Red Proceso G.



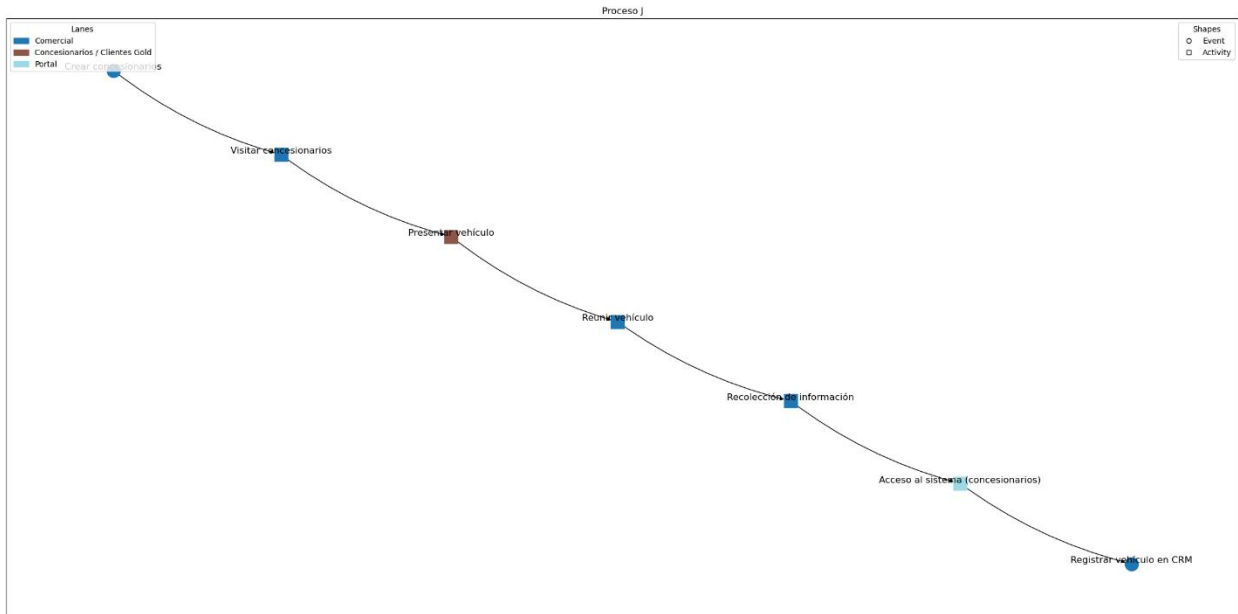
Anexo 32 Red Proceso H.



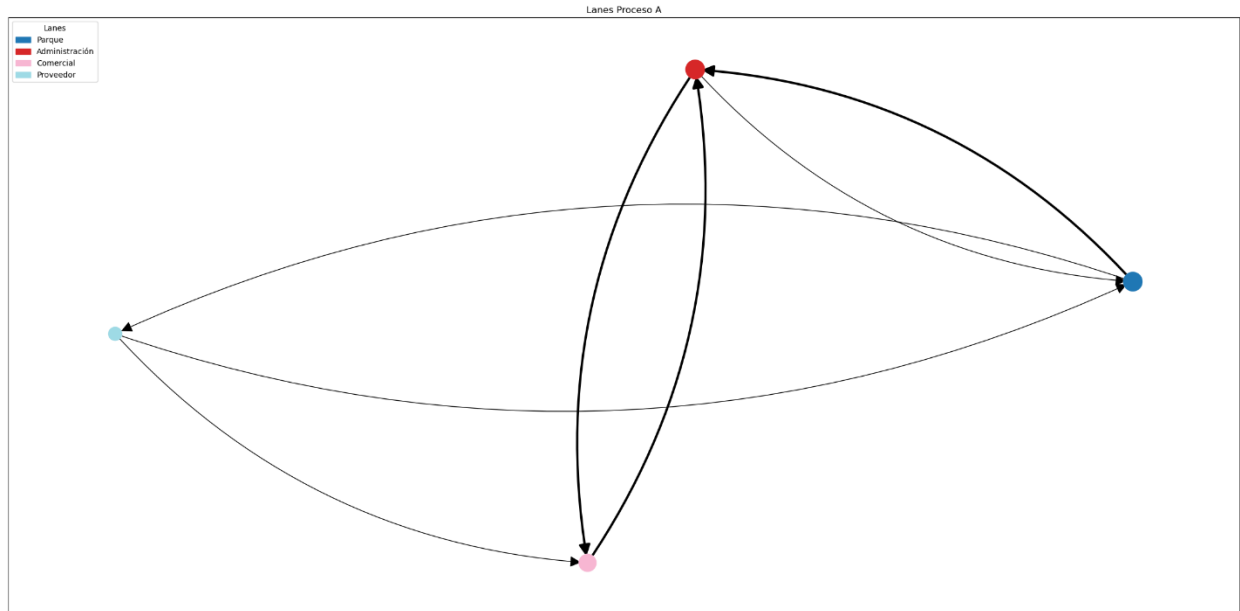
Anexo 33 Red Proceso I.



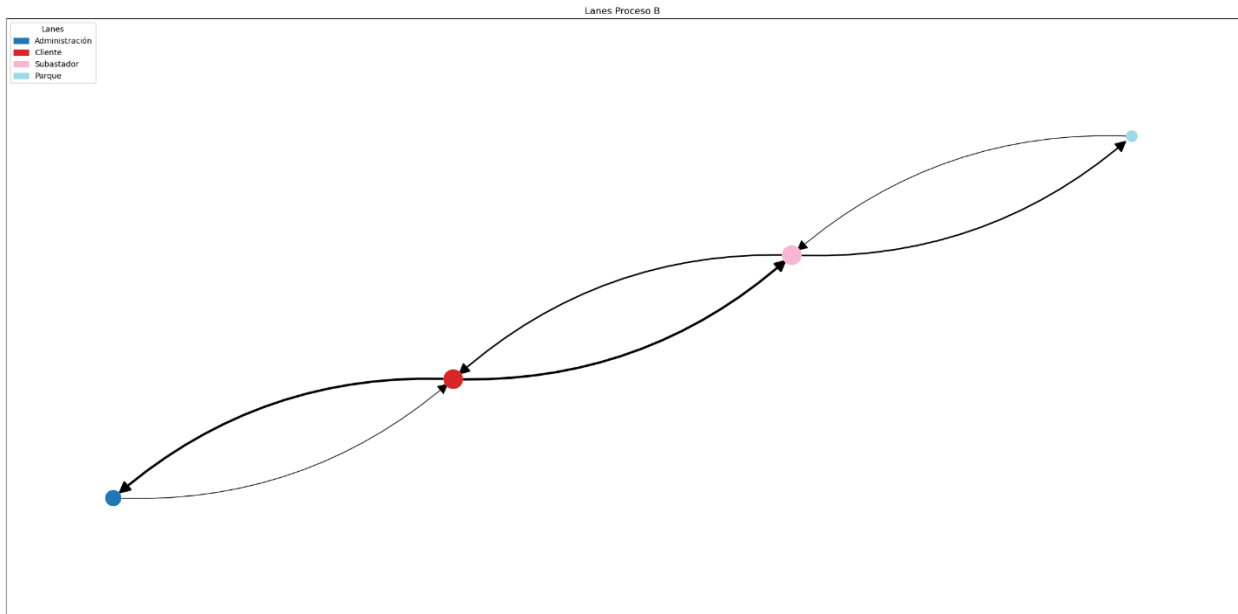
Anexo 34 Red Proceso J.



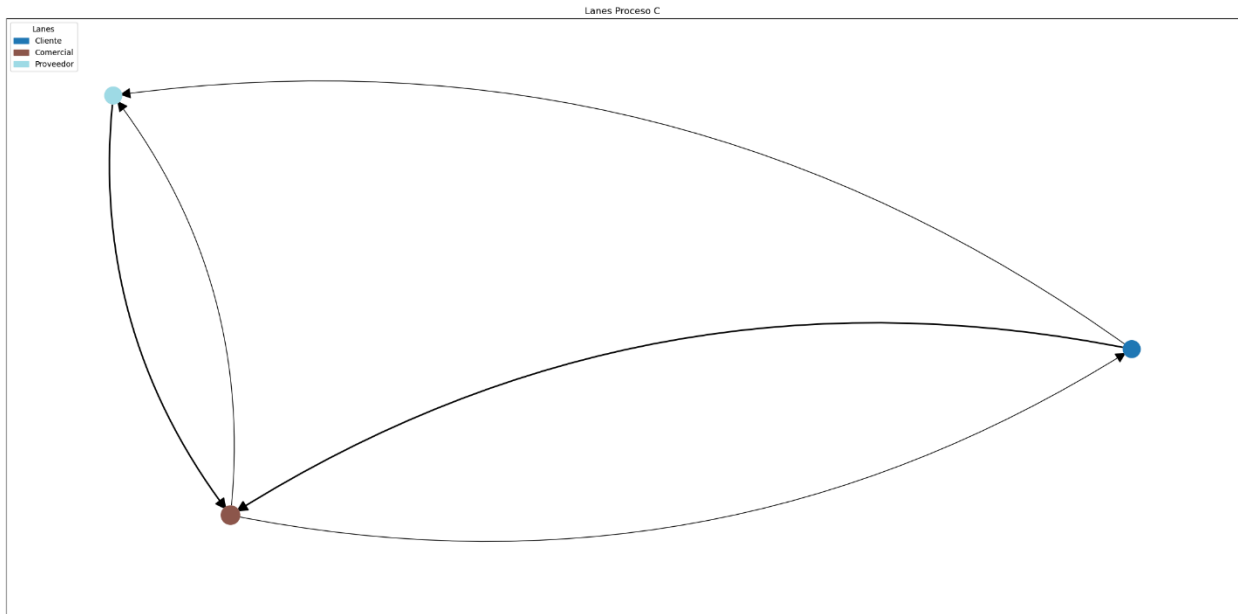
Anexo 35 Red Proceso A (carriles).



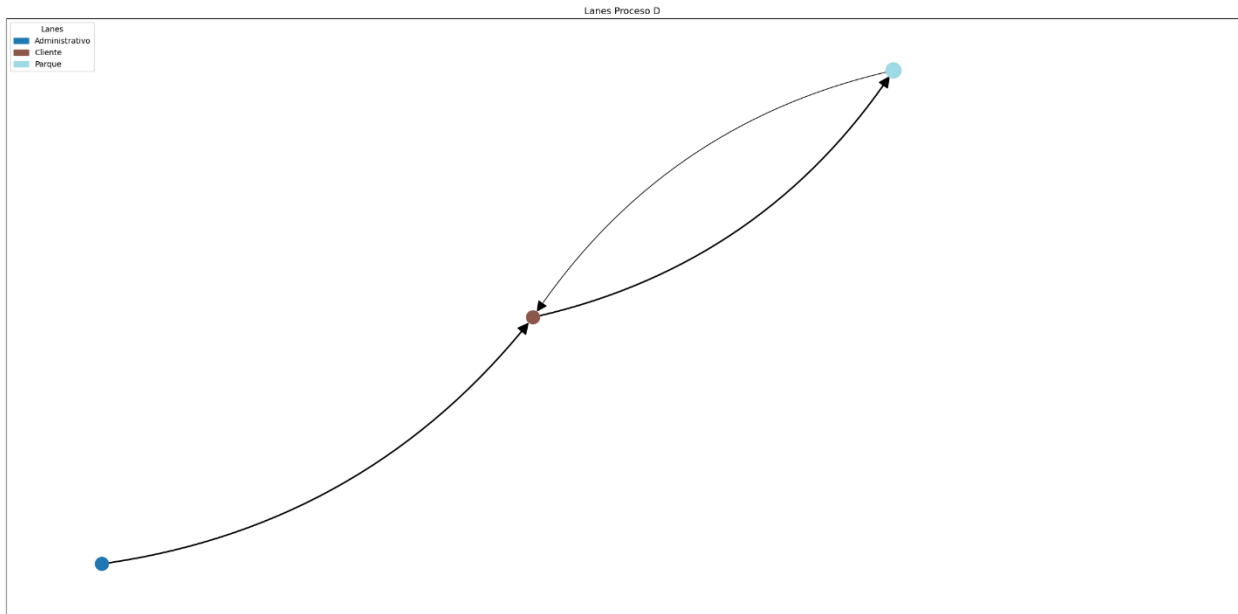
Anexo 36 Red Proceso B (carriles).



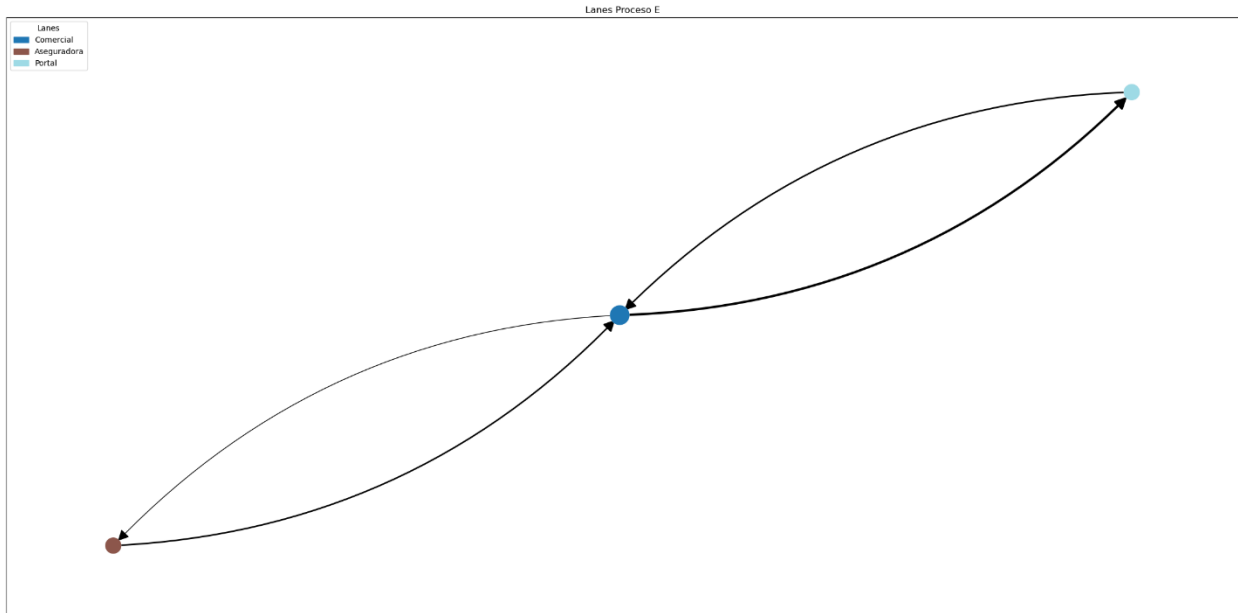
Anexo 37 Red Proceso C (carriles).



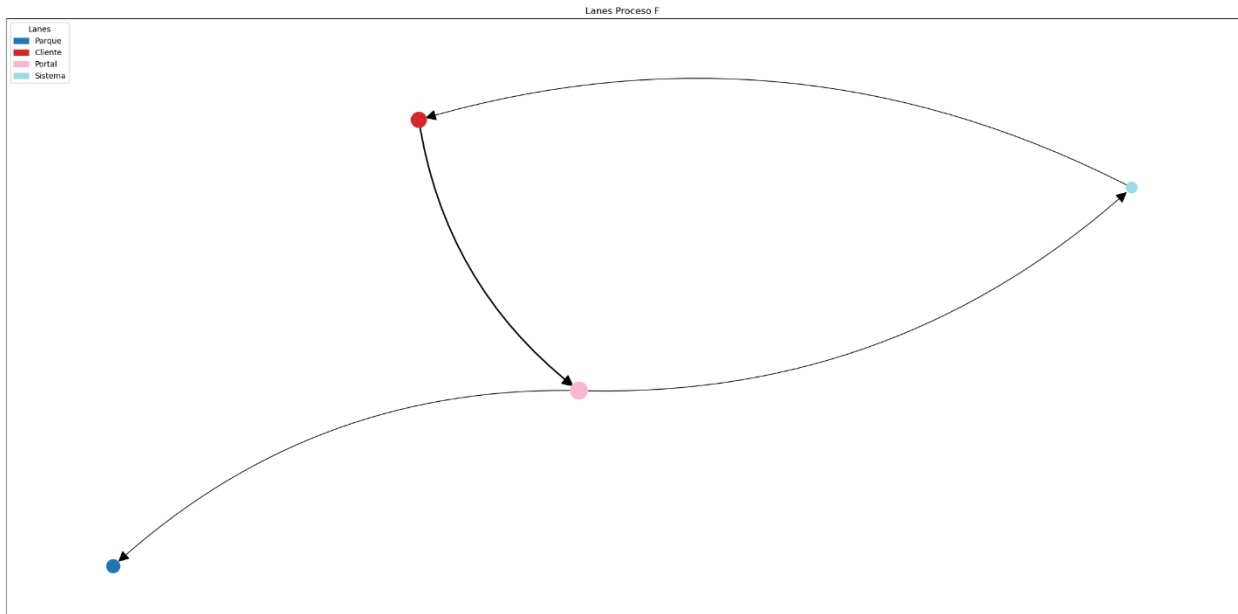
Anexo 38 Red Proceso D (carriles).



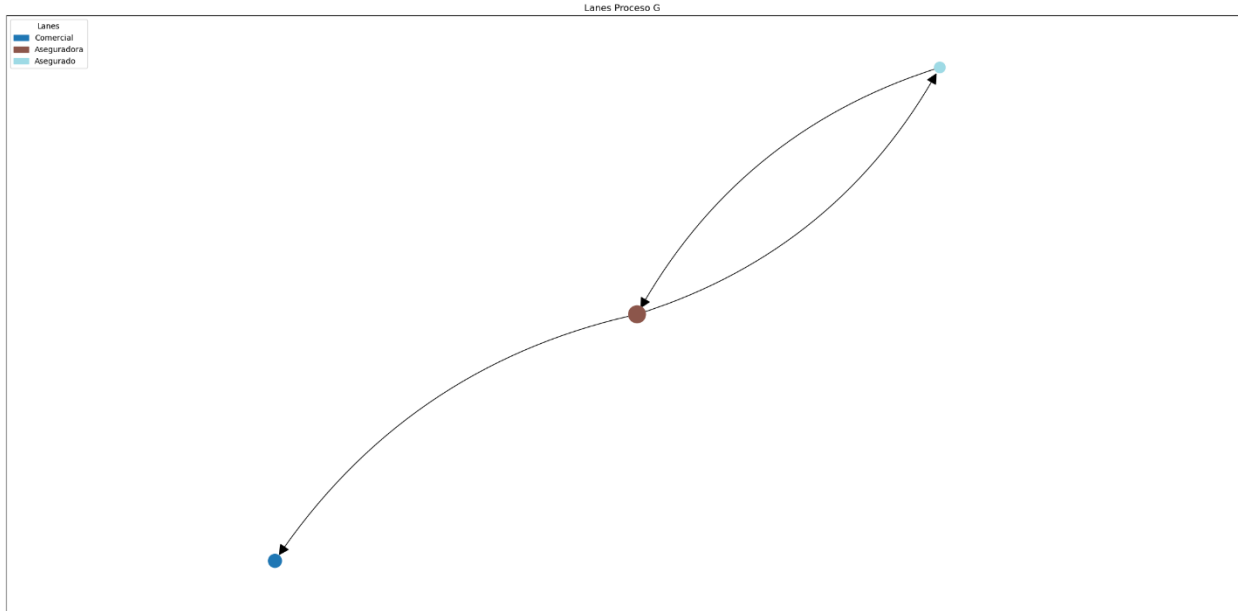
Anexo 39 Red Proceso E (carriles).



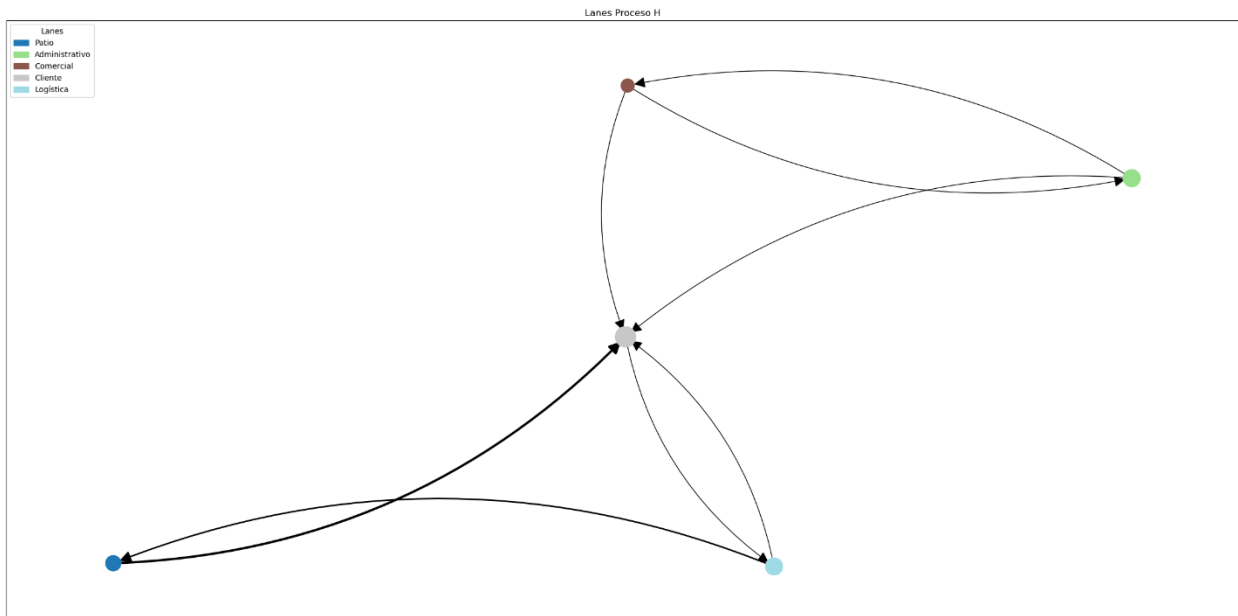
Anexo 40 Red Proceso F (carriles).



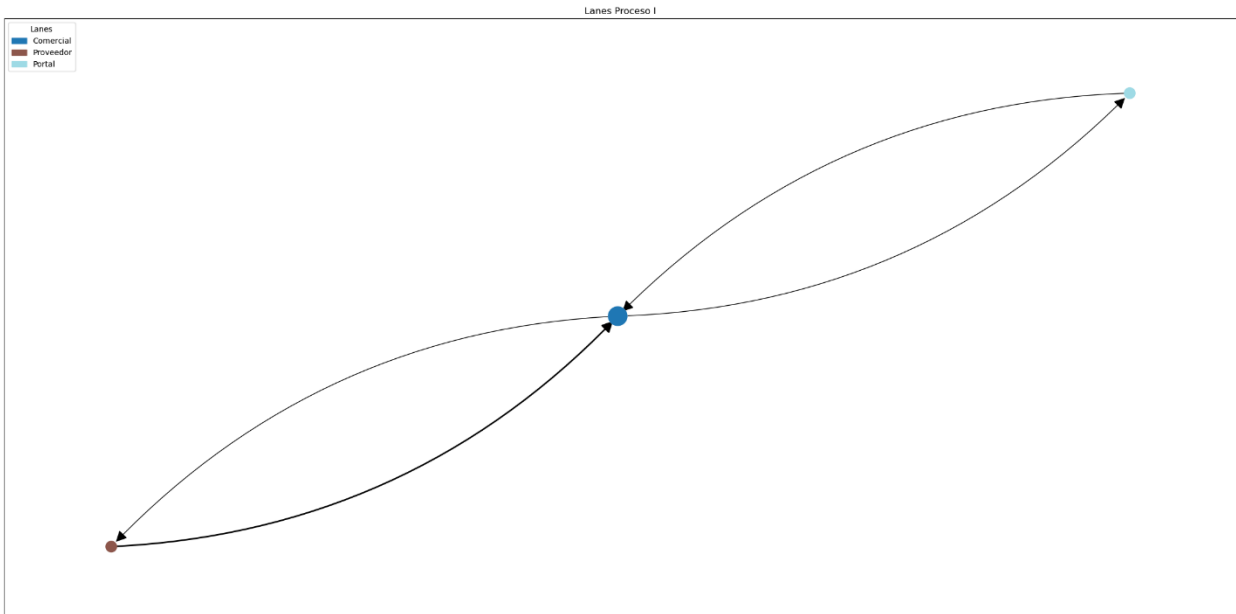
Anexo 41 Red Proceso G (carriles).



Anexo 42 Red Proceso H (carriles).



Anexo 43 Red Proceso I (carriles).



Anexo 44 Red Proceso J (carriles).

