



## HERRAMIENTA SOFTWARE PARA EL ANÁLISIS DE PROBLEMAS CUYA ABSTRACCIÓN SE FUNDAMENTA EN GRAFOS

### Resumen

Muchos problemas pueden resolverse de manera eficiente planteándose mediante la creación de grafos. Este trabajo plantea el desarrollo de una herramienta de software que permite representar la información y sus relaciones de manera gráfica, facilitando el análisis de ésta, apoyando de este modo la toma de decisiones durante la resolución de problemas. Este proyecto fue desarrollado con una metodología de cuatro fases: 1) investigación y análisis de información; 2) determinación del modelo conceptual del software por medio del enlace entre los fundamentos teóricos y las necesidades prácticas; 3) definición de los requerimientos de la herramienta software y 4) implantación de la herramienta. Como resultados fundamentales del desarrollo del proyecto se obtuvo: a) una clasificación de algoritmos utilizados en la aplicación de Teoría de Grafos que considera la perspectiva de diversos autores y expertos en la materia; b) una síntesis de los algoritmos más utilizados en seis áreas distintas; c) un modelo conceptual de una herramienta software escalable y d) una herramienta software que implementa el modelo conceptual considerando dos áreas específicas de aplicación. Entre las conclusiones más relevantes del proyecto se incluyen: a) algunos problemas fácilmente planteados como grafos, no tienen una posible resolución determinística, por lo que deben ser resueltos de manera heurística, lo que a su vez requiere de un alto tiempo de cómputo acorde con la complejidad del problema; b) la herramienta que se define y desarrolla debe

■ Ing. Alasia Delgado Saab.

[alasia.delgado@gmail.com](mailto:alasia.delgado@gmail.com)

■ Ing. Oriélisa J. Carapaica.

[orielisa@gmail.com](mailto:orielisa@gmail.com)

■ Dra. Lourdes M. Ortiz Sosa.

[lortiz@ucab.edu.ve](mailto:lortiz@ucab.edu.ve)

ser escalable para considerar la incorporación de funcionalidades y áreas de aplicación; c) la investigación es una actividad previa obligatoria para el desarrollo de cualquier herramienta innovadora; d) el desarrollo de software debe ser un proceso sistémico y determinado por la calidad; e) la herramienta desarrollada puede ser ampliada a corto plazo para un análisis cuantitativo extendido de gran valor en el desarrollo científico y la incorporación de reportes y algoritmos de análisis.

**Palabras clave:** grafos, modelaje de software.

### Abstract

*There are many problems that can be solved efficiently if they were approached through the use of graphs. This article deals with the development of a software application that allows to portray information and its' relationships graphically, thus simplifying data analysis and supporting the decision making process during problems resolution. This project has been developed under a four phase methodology: 1) Research and information analysis, 2) Determination of the conceptual model for the software, through the relationship between the established base theories and practical necessities, 3) Definition of the software requirements in place, and 4) Development of the application. Among the main results accomplished from the development of this project, there are: a) A classification of the algorithms used in Graph Theory considering the perspective of several authors and experts in the subject, b) An outline of some of the most used algorithms in six different areas of application, c) A conceptual model for a scalable software application and d) An application that implements that conceptual model considering two specific areas of application. Among the most relevant conclusions obtained from this project, there are: a) Some problems easily handled by graphs do not have a deterministic solution, because of this, they must be solved through the use of heuristics, which requires longer processing time according to the problem complexity, b) The application to be defined and developed should be scalable to allow for the incorporation of functionalities and different application areas, c) Previous research is a mandatory activity in the development of any innovative application, d) Software development should be an incremental process determined by quality, e) The application developed may be extended in*

*the short term for a greater scientometric analysis adding much value to scientific research, as well as extended to incorporate new reports and analysis algorithms.*

**Keywords:** *graphs, software modeling.*

### Planteamiento del problema

Muchas situaciones complejas de la vida cotidiana pueden resolverse eficientemente a través de la implantación y análisis de Grafos que representan la información vinculada al problema estudiado y sus relaciones de interés, como suele hacerse, en organigramas, diagramas de bases de datos, redes cuantitativas y pert-cpm para la gestión de proyectos. Sin embargo, considerando la cantidad de información requerida, puede resultar un trabajo por demás tedioso la estructuración de la información de esta manera y más aún la aplicación de métodos para el análisis de ésta, especialmente si esta tarea debe hacerse de forma manual. Como una alternativa para la creación y análisis de estas representaciones gráficas, existe la teoría de grafos, cuyas aplicaciones pueden hallarse, entre otras cosas, en herramientas informáticas para la resolución de problemas o conformando aplicaciones para el planteamiento y resolución de problemas en entornos muy específicos.

Muchas de las aplicaciones disponibles para la representación gráfica de la información requerida por la teoría de grafos no facilitan la extracción de datos para su análisis. Las que sí lo hacen, se encuentran orientadas al análisis de problemas específicos, son limitadas en funcionalidad o, muchas veces, presentan una interfaz que requiere cierto dominio de estas teorías por parte del usuario, lo cual dificulta la manipulación de la información.

Considerando lo antes descrito, se planteó el desarrollo de una herramienta de software que provea al usuario la facilidad de representar información relacional de manera gráfica, facilitando su análisis, soportando así la toma de decisiones. Para ello, presenta una interfaz orientada al área de interés del usuario, evitándole así el posible costo que conlleva la instalación de varias aplicaciones necesarias para cubrir los diversos temas de interés y aislándolo de los procesos y estructuras de datos necesarias para el análisis de grafos.

## 1. Objetivos

Como objetivo general de este proyecto, se planteó desarrollar una herramienta software que permita la construcción y manipulación de grafos para el análisis de problemas diversos.

Para el logro del objetivo general planteado, se especificaron compromisos de menor alcance entre los que se incluyen:

- Investigar la teoría de grafos a fin de definir los métodos más apropiados para implementar una herramienta que permita modelar situaciones para su análisis basado en grafos.
- Implementar una herramienta gráfica que facilite el diseño y manipulación de grafos.
- Implantar métodos que permitan la asociación y extracción de información a partir de un modelo basado en grafos.
- Aplicar la herramienta para el análisis y resolución de casos de prueba.

## Marco referencial

La base teórica del proyecto se encuentra dividida en dos temas principales:

- La teoría de grafos, en la que se explica un enfoque global de la misma, haciendo énfasis en la definición y caracterización de grafos y algoritmos frecuentemente utilizados así como áreas de aplicación.
- La ingeniería del software, donde se tratan temas que plantean la base tecnológica para la toma de decisiones requerida durante la implantación de la herramienta de software; haciendo énfasis en temas como estilos arquitectónicos disponibles y plataformas para la implementación de software.

### 2.1. Teoría de Grafos

Según la enciclopedia abierta Wikipedia (2005), se entiende como teoría de grafos la que se encarga de establecer los fundamentos y bases necesarias para resolver problemas de una determinada complejidad a través de estructuras matemáticas denominadas "Grafos". De acuerdo con esto, un *grafo*  $G$  puede ser definido como un par  $(V, E)$  donde  $V$  es un conjunto de puntos que conforman los vértices del grafo, y  $E$  el conjunto de relaciones binarias entre

los elementos de  $E$ , lo que permite establecer una relación con notación matricial. Estos componentes pueden poseer atributos que proveen mayor información al grafo. Como atributos comunes se pueden nombrar, la orientación o dirección de las aristas (definida entonces por un par ordenado) y la cantidad de aristas asociadas a un nodo. Existen otros atributos asociados a los nodos o aristas que denotan diversos tipos de relaciones y soportan además, en el caso de las aristas, lo que se entiende como el *peso de la arista* y determina un valor para la relación establecida. Dado que los grafos pueden poseer cualquier combinación de atributos en sus componentes, existen ciertas especificaciones de estas combinaciones que permiten denotar tipos de grafos y su clasificación.

Dado que el grafo no es más que una estructura de representación, el interés recae en las diversas formas de manipular esta representación. La información presentada en el grafo puede ser fácilmente analizada e interpretada gracias a los métodos y algoritmos existentes, que facilitan más que la extracción de la información, la automatización de este proceso.

A pesar de la versatilidad que poseen los grafos gracias a las variaciones en las características de sus componentes, la información que se extrae del grafo tiende a seguir ciertos lineamientos, delimitados en muchas ocasiones, de acuerdo al tipo de grafo que se presente.

Además de los algoritmos basados en la teoría de grafos, existen otros métodos utilizados para la resolución de planteamientos representados en grafos que han sido desarrollados y refinados para lidiar con temas y objetivos muy específicos. Estos algoritmos pueden ser obtenidos mediante el uso de heurísticas o algoritmos más específicos y son, por lo general, utilizados para los casos de optimización de resultados en donde los problemas suelen ser no determinísticos. Esta forma de resolución se puede encontrar en áreas como dinámica de sistemas, investigación operativa, ciencias de la computación y ejemplos puntuales en otros campos como son redes de computación entre tantos.

Según Algarra & Argilaga (2003), el análisis de grafos ha experimentado un rápido desarrollo en los últimos años, y ha llegado a convertirse en un importante instrumento estadístico-matemático para ámbitos tan diversos como la Investigación Operativa, la Psicología, la Genética, la Lingüística

ca, la Sociología, las Redes, la Cibermetría, etc. En psicología existe ya una cierta tradición en la utilización de la teoría de grafos para el análisis de textos. Pero una de las disciplinas que en mayor medida ha contribuido al desarrollo de la teoría de grafos ha sido, sin duda, la sociología, en la que el modelaje matemático de redes sociales, en las últimas décadas, ha impulsado aún más el desarrollo básico y las posibilidades de aplicación de la teoría de grafos.

## 2.2. Planteamiento para las condiciones de la implantación del software

Para lograr mejores resultados en la implementación de software, es importante que desde tempranas etapas del desarrollo se evalúen las condiciones de arquitectura y plataforma tecnológica a la que se verá enfrentada la aplicación software. Es un error común, en los desarrollos de software, que una vez culminado el proyecto se tenga que incurrir en gastos de nuevas tecnologías, porque la herramienta implantada no opera de manera correcta, requiriendo su adaptación a nuevas condiciones tecnológicas.

### 2.2.1. Estilos arquitectónicos

Grimán (2005) señala que no es un secreto que en los grandes sistemas, el logro de algunas cualidades como el desempeño, la disponibilidad, la seguridad, la reusabilidad, la portabilidad e interoperabilidad y la modificabilidad no sólo son dependientes de las prácticas a nivel de código (selección del lenguaje, diseño detallado, algoritmos, estructuras de datos y pruebas), sino que también son dependientes, especialmente, de la arquitectura del software. La arquitectura de software es el correcto punto de partida para analizar la calidad del mismo. La definición de la arquitectura del software se obtiene en las etapas iniciales del ciclo de vida de un sistema, y es un elemento que puede ser utilizado para determinar cuán bien el sistema cumplirá con sus requerimientos, tanto implícitos como explícitos.

### 2.2.2. Definición de la plataforma tecnológica

Una vez definidas la arquitectura, metodología, herramientas y usuarios del software, se pueden determinar los equipos, redes, sistemas operativos y demás software a utilizar.

Según Miranda (2005), la plataforma es un vehículo para la tecnología y no debe ser tomada como dada ni definir la tecnología o herramientas a usar.

Su definición debe depender de necesidades y herramientas así como de la noción, no de adquirir simples equipos para el funcionamiento de los sistemas, sino de implementar una "infoestructura" que brinde acceso a la tecnología y permita acceder, recopilar, divulgar y compartir información.

## Marco metodológico

Se planteó una metodología que consta de 4 fases fundamentales, descritas a continuación:

### 3.1. Primera fase: investigación preliminar

Esta fase comprende principalmente una labor de investigación documental cuyo objetivo es determinar las bases para una definición conceptual de la herramienta a desarrollar, mediante el enunciado de conceptos, la descripción de alternativas y la definición de estructuras y funcionalidades básicas necesarias para el planteamiento y posible resolución de situaciones basadas en grafos.

Debido a que en esta primera fase se realizan planteamientos fundamentales para el desarrollo final del proyecto software, la metodología a seguir tiene una forma iterativa, dirigida a revisar las observaciones encontradas durante el proceso y ampliar correctamente en los temas necesarios.

El resultado de esta fase comprende, fundamentalmente, la definición de diversas áreas apoyadas en la teoría de grafos junto con aplicaciones reales, características y algoritmos habitualmente utilizados en ella.

### 3.2. Segunda fase: modelo conceptual

Esta es una fase de enlace entre los fundamentos teóricos planteados y los requerimientos funcionales técnicos propios del desarrollo de una aplicación. En ella se limitan las áreas o temas que serán integrados en la herramienta, así como la identificación de los algoritmos a utilizar y sus requerimientos, necesarios para la resolución de los planteamientos en cada área.

### 3.3. Tercera fase: definición de la herramienta

Esta fase representa el punto de congruencia de las metodologías habituales para el desarrollo de software donde se definen los requerimientos funcionales de la herramienta, así como características y formatos de la misma que servirán de

guía para el desarrollo de la aplicación, tomando en consideración los lineamientos establecidos en la etapa anterior.

### 3.4. Cuarta fase: desarrollo de la herramienta

Para la implantación de la herramienta, la metodología a seguir se apoya en un método RUP (*Rational Unified Process*) simplificado que permite su adaptación tanto desde el punto de vista de la arquitectura como del tiempo con el que se dispone para el desarrollo del proyecto.

Esta metodología consta de una espiral con tres fases fundamentales recurrentes a lo largo del desarrollo de la herramienta software; estas fases son:

- Fase de inicio: comprende una etapa de preparación y evaluación para el desarrollo a seguir. En ella se plantean los objetivos inmediatos y se analizan los posibles riesgos a encontrar para alcanzar esos objetivos.
- Fase de desarrollo: una vez planteados los objetivos y métodos a seguir durante la fase de inicio, se procederá con el desarrollo de los componentes de la herramienta.
- Fase de pruebas: consta de la evaluación de los componentes realizados para verificar su integridad, funcionalidad y robustez.

La culminación de esta etapa deja como resultados la herramienta funcional, así como la documentación asociada a ésta, tales como manual de usuario y manual técnico o de sistema.

Esta separación del proceso de desarrollo permite identificar con mayor facilidad los objetivos y condiciones de éxito de cada fase, aplicando además en cada etapa una metodología especialmente orientada a lograr la meta allí definida.

## Desarrollo y resultados

Esta sección presenta las actividades realizadas, de acuerdo a las fases planteadas en la metodología referida en la sección anterior, para lograr la implantación de la herramienta software objetivo principal de este proyecto.

### 4.1. Primera fase: documentación preliminar

De manera de poder clasificar y organizar las ideas a tratar, la investigación documental fue di-

vidida en dos temas fundamentales, (a) Teoría de grafos y (b) Fundamentos técnicos de implantación. Considerando además los objetivos del proyecto, estos temas fueron subdivididos en áreas de mayor relevancia y de influencia directa para la implantación de la herramienta. Esta subdivisión se encuentra representada en la tabla 1 (Temas Relevantes de Investigación).

Temas	Sub-Temas
Teoría de grafos	<ul style="list-style-type: none"> <li>• Algoritmos basados en grafos.</li> <li>• Áreas relacionadas con la teoría de grafos.</li> <li>• Aplicaciones informáticas apoyadas en la teoría de grafos.</li> </ul>
Fundamentos técnicos de implantación	<ul style="list-style-type: none"> <li>• Estilos arquitectónicos.</li> <li>• Metodologías de implantación.</li> </ul>

Tabla 1: Temas Relevantes de Investigación.  
Fuente: Elaboración Propia.

Sobre la base de los temas seleccionados, se procedió a la compilación de la información a partir de diversas fuentes como artículos, libros de texto y documentos digitales. La información recopilada se complementa con la realización de entrevistas a personas expertas en los temas involucrados, cuyo aporte resultó muy satisfactorio.

Luego de sucesivas verificaciones de información, tanto con las personas involucradas, como con las diversas fuentes, la información fue relacionada y organizada para conformar el Marco Referencial del proyecto y se obtuvo como síntesis la clasificación de algoritmos aplicables para el análisis de grafos que se presenta en el Apéndice A, Algoritmos basados en la teoría de grafos.

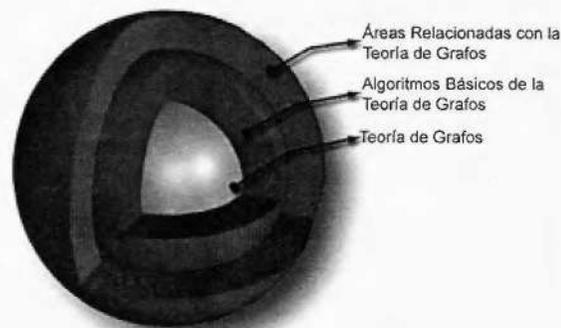


Figura 1: Modelo Conceptual de la herramienta.  
Fuente: Elaboración Propia.

## 4.2. Segunda fase: modelo conceptual

El modelo conceptual presentado en la figura 1 (Modelo Conceptual de la Herramienta), es una representación por capas donde el núcleo está conformado por los conceptos fundamentales de la teoría de grafos.

Apoyándose en los componentes básicos de grafos, se implementan diversos algoritmos que permiten la manipulación de los elementos de la primera capa y pasan a conformar el siguiente nivel de abstracción. Esta segunda capa sirve de interfaz entre los requerimientos del usuario y los grafos subyacentes; para el manejo de esos requerimientos se plantea una tercera capa (Áreas relacionadas con la teoría de grafos), que expone un análisis del área en particular para su interacción con el usuario.

### 4.2.1. Áreas y funcionalidades para la aplicación de Teoría de Grafos

Debido a la cantidad de temas involucrados con la teoría de grafos y a las diversas formas de abordarlos, se seleccionaron sólo algunos temas, evaluando diversos factores, como son, disponibilidad de información, grado de relación con algoritmos de la teoría de grafos y relevancia académica del tema. Estos temas fueron: Dinámica de Sistemas, Cienciometría, Genética, Manejo de Proyecto, Redes Computacionales y Redes Sociales

Una vez seleccionados los temas a considerar, se profundizó en la investigación sobre cada tema, siendo de particular interés los principales requerimientos y funcionalidades de éste, así como las posibles formas de resolverlos. Los resultados obtenidos se encuentran en el apéndice B (Áreas y funcionalidades para la aplicación de teoría de grafos).

La información obtenida fue analizada, en términos de utilidad y funcionalidad, resultando en información general de la funcionalidad de cada área, luego utilizada para identificar aquellos algoritmos que satisfagan los requerimientos planteados. Finalmente, la información fue filtrada para seleccionar los temas con la mayor funcionalidad considerando los algoritmos a utilizar. Este proceso puede apreciarse en la tabla 2 (Algoritmos de teoría de grafos utilizados en cada área), que presenta, del lado derecho, la cantidad de veces que el algoritmo es utilizado, y en la parte inferior la cantidad de algoritmos utilizados por área.

Como consecuencia al sistema de selección de algoritmos y áreas planteadas, se seleccionaron las áreas de *Redes Sociales* y *Cienciometría*, siendo la primera una de las áreas más representativas de la aplicación de la teoría de grafos y la segunda de especial relevancia académica para la implantación del proyecto. Para establecer la funcionalidad de éstas áreas, los algoritmos seleccionados para su implantación fueron los que resultan de más utilidad a una mayor cantidad de temas, los cuales se encuentran en la Tabla 2 con un "Total Uso" mayor que 1.

## 4.3. Tercera fase: definición de la herramienta

La definición de la herramienta consta fundamentalmente de la obtención y especificación de los requerimientos funcionales que deberá satisfacer; complementario a esto, es necesario identificar el entorno de implantación así como el lenguaje de desarrollo a utilizar, de manera que, en conjunto, satisfagan los requerimientos definidos.

Para la obtención de los requerimientos funcionales y no funcionales del sistema se realizaron diversas entrevistas y se recurrió a diversas fuentes, tanto bibliográficas como al estudio de aplicaciones resaltantes en los temas considerados de teoría de grafos, donde se estudió la funcionalidad básica ofrecida, el manejo de resultados, y el estilo de la interfaz básica utilizada; información que sirvió de complemento a la definición de la herramienta a implantar. La tabla 3 presenta los resultados obtenidos de esta investigación, estableciendo los requerimientos funcionales, no funcionales y de plataforma con los que debe contar la herramienta a desarrollar.

Una vez planteados los requerimientos funcionales de la herramienta, se procedió a desarrollarlos en funcionalidades específicas, obteniendo así los posibles casos de uso del sistema.

Para la definición del entorno de implantación se procedió a hacer un estudio de la plataforma de hardware y software disponible en la Universidad Católica Andrés Bello (UCAB), de manera de analizar los requerimientos básicos establecidos por la plataforma tecnológica disponible. A partir de este estudio, se estableció que en la UCAB se utilizan principalmente tres plataformas de sistemas operativos; éstas son, en orden de cantidad de computadores en los que está instalado, Windows 2000, Linux (Red Hat) y Solaris.

	Áreas						Total Uso
	Redes Sociales	Genéticos	Redes Computacionales	Diseños de Sistemas	Manejo de Proyectos	Combinatoria	
Recorrido de la matriz de adyacencia	✓	✓	✓	✓	✓	✓	6
Recorrido de la matriz de distancias	✓	✓	✓	✓	✓	✓	6
Caminos Hamiltonianos	✓	✓	✓			✓	4
Algoritmo de Roy-Warshall	✓		✓		✓	✓	4
DFS	✓		✓			✓	3
Grado de un vértice	✓		✓			✓	3
Algoritmo de Dijkstra	✓				✓	✓	3
Puntos de articulación	✓		✓			✓	3
BFS	✓					✓	2
Número Cromático	✓					✓	2
Algoritmo de Prim y Kruskal	✓		✓			✓	2
Árbol cobertor	✓		✓			✓	3
Ecuaciones diferenciales				✓			1
Ordenamiento topológico					✓		1
Cálculo del árbol Steiner			✓				1
Algoritmo de Euler		✓					1
Algoritmos de flujo			✓				1
Algoritmo Pert - CPM					✓		1
<b>Total Algoritmos:</b>	<b>12</b>	<b>3</b>	<b>9</b>	<b>2</b>	<b>4</b>	<b>11</b>	

Tabla 2: Algoritmos de teoría de grafos utilizados en cada área. Fuente: Elaboración propia.

Finalmente, se procedió a hacer un estudio de los principales lenguajes de programación orientados a objetos (Java y .Net) y las ventajas y desventajas que ofrecen en cuanto a las métricas de portabilidad, facilidad de uso y compatibilidad de lenguajes, haciendo notar que esta última está entre los requerimientos importantes del sistema de software.

Con base en el estudio de plataformas de desarrollo consideradas, se escogió la plataforma de Microsoft .Net Framework, ya que es compatible con gran variedad de lenguajes de programación y ofrece posibilidades de portabilidad a través del proyecto Mono, además de funcionar sobre el sistema operativo predominante en el entorno de implantación, ofreciendo una interfaz de usuario más amigable a los esperados usuarios del sistema.

#### 4.4. Cuarta fase: desarrollo de la herramienta

Llegada a esta fase del desarrollo del proyecto, se tiene definida la estructura general de la herra-

Requerimientos Funcionales	Requerimientos No Funcionales	Requerimientos de Plataforma
<ul style="list-style-type: none"> <li>• Debe permitir la selección del tema con el que trabajar.</li> <li>• Debe facilitar la creación de diagramas de gráfica mente.</li> <li>• Debe facilitar la creación de nuevas estructuras, configurando sus atributos según convenga.</li> <li>• Permitirá la exportación de los grafos en formato imagen.</li> <li>• Debe permitir el almacenamiento y carga en de la información en archivos planos.</li> <li>• Debe generar reportes que permitan extraer información del grafo.</li> <li>• Los reportes a generar deberán ser configurables,</li> <li>• Los reportes deberán ajustarse de acuerdo al tema que el usuario este trabajando.</li> </ul>	<ul style="list-style-type: none"> <li>• Será desarrollado de forma tal que facilite su extensibilidad</li> <li>• El sistema a desarrollar es una aplicación autónoma, y debe ser compatible con la plataforma de software disponible en la Universidad.</li> <li>• La selección de herramientas y componentes a utilizar como complemento a la herramienta principal de desarrollo deberán ser de licencia abierta.</li> </ul>	<p>Se recomienda poseer, como hardware mínimo en la computadora de instalación:</p> <ul style="list-style-type: none"> <li>• Memoria RAM de 128 MB.</li> <li>• Capacidad disponible de disco duro de 30 MB.</li> </ul>

Tabla 3: Requerimientos funcionales, no funcionales y de plataforma del sistema. Fuente: Elaboración propia.

mienta, así como los requerimientos de ésta y las principales funcionalidades a implantar.

##### 4.4.1. Primer ciclo

De acuerdo a la metodología establecida, el desarrollo de la aplicación fue realizado en diversos ciclos. El primer ciclo contempló la inicialización de la herramienta, que implica la carga de los elementos de interfaz del tema a manejar, así como la creación, lectura y escritura del grafo.

Se procedió a identificar los procesos de carga tanto de cada tema desarrollado (apoyando la extensibilidad), como la carga del grafo y sus componentes, lo que condujo a la definición de las estructuras de datos "Grafo", "Nodo" y "Arista" y, al uso del lenguaje XML como soporte a la herramienta.

Una vez definidas las estructuras fundamentales a utilizar, fue necesario establecer las relaciones entre ellas y sus principales interfaces de manipulación e integración con cada componente de la herramienta. Ya detalladas las relaciones e in-

terfaces, se procedió al desarrollo de este primer ciclo y a las posteriores verificaciones de robustez y funcionalidad de la aplicación.

Durante ese proceso de verificación se identificó un componente de código abierto (NetronGraphLib), disponible públicamente, que facilita el manejo de grafos y sus componentes. Contando con elementos más robustos, se pasó a definir la integración de los diversos módulos que conforman la herramienta.

La integración de esos elementos se encuentra representada en la figura 2 (Interacción de Componentes), que establece como base la librería NetronGraphLib, en la que se apoya una segunda capa para la creación y modificación de nodos y aristas, así como toda la funcionalidad básica requerida. La figura muestra además una capa final para la interacción de los componentes, que funciona además como interfaz para facilitar el entendimiento del usuario. Es de resaltar la modificación de la librería base, que fue necesaria para su adaptación a los requerimientos del sistema.

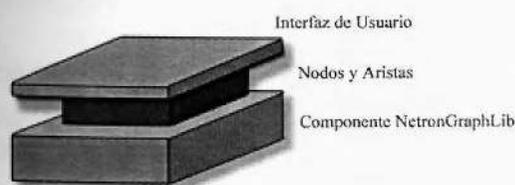


Figura 2: Interacción de Componentes.  
Fuente: Elaboración propia

#### 4.4.2. Segundo ciclo

Es en este ciclo donde se desarrollaron muchos de los requerimientos tanto funcionales como no funcionales de la herramienta y se definió además la estructura de navegación de la aplicación.

Uno de los procesos fundamentales llevados a cabo en esta etapa del desarrollo consistió en la adaptación de la herramienta al requerimiento de configuración de atributos de nodos y aristas, apoyándose en un proceso de creación y compilación dinámica de nodos y aristas. La figura 3 (Proceso de creación de estructuras) presenta una vista del proceso desarrollado para la adición de nuevas estructuras. Este proceso es además de gran importancia por ser utilizado para permitir la extensibilidad de la herramienta agregando funcionalidad extra, y conforma la base para el proceso de creación de reportes.



Figura 3: Proceso de creación de estructuras.  
Fuente: Elaboración propia.

En esta etapa además se implantaron los algoritmos de análisis de grafos seleccionados como producto de la investigación realizada.

Como resultado de este ciclo, se deja la aplicación en un estado casi terminado, estableciendo las bases para la creación de reportes sobre el grafo creado.

#### 4.4.3. Tercer ciclo

Para el desarrollo de este ciclo se realizó un estudio acerca de los requerimientos, así como la funcionalidad y estructuras de datos utilizadas en el sistema. Este estudio incluyó pruebas sobre los algoritmos implantados y análisis de sus resultados, así como la consideración de diversos componentes que facilitan la generación de reportes.

El estudio de los componentes y herramientas de generación de reportes además requirió de un análisis de interfaz, funcionalidad y flexibilidad. Esto permitió la identificación del formato de los reportes de la herramienta, los métodos de generación de éstos así como las posibles entradas que requieran de forma de estandarizar el proceso de generación de reportes y permitir la fácil extensibilidad de la herramienta en esta área.

En vista de que entre los requerimientos de la aplicación se encuentra la parametrización o personalización de los reportes generados y que la información no proviene de un manejador de base de datos, se decidió desarrollar un módulo de creación de reportes, con funcionalidad parecida a la de las herramientas estudiadas y cuyo resultado sea un reporte en formato web (html), de manera de aprovechar las prestaciones a nivel de interfaz y la portabilidad brindadas por el lenguaje HTML.

Para complementar el módulo desarrollado para la generación de reportes, se procedió a la búsqueda de componentes que faciliten la creación de gráficas, y que además, sean de código abierto y compatibles con el ambiente de desarrollo. El componente seleccionado que satisface estas características y además es configurable y de fácil uso es ZedGraph.

#### 4.4.4. Cuarto ciclo

Este cuarto ciclo está comprendido principalmente por la realización de documentación de apoyo al usuario y de soporte técnico para futuras versiones de la herramienta, así como de un período de pruebas a la aplicación, donde se intenta comprobar la integración de cada uno de los componentes desarrollados que la conforman.

Las pruebas realizadas constaron de la evaluación y, en algunos casos, la posterior modificación de los módulos utilizados en la herramienta. Para ello, además de las pruebas resultantes del constante uso de la aplicación, se realizaron estudios para medir el tiempo de carga de distintos grafos, con la intención de determinar la eficiencia relativa de la aplicación y se procedió a crear dos casos de uso, cada uno con la intención de simular la ejecución de la aplicación bajo las áreas de Cienciometría y Redes Sociales.

Como resultado de la medición de tiempos de carga en los distintos archivos probados, se obtuvieron los tiempos promedio de carga por nodo de 0,15 seg. y arista de 0,12 seg.

Estos resultados permiten estimar el tiempo de carga para grafos con gran cantidad de nodos, considerando que el tiempo de lectura para cada uno puede variar de acuerdo a la cantidad de atributos con información almacenada que posea, sin embargo, en líneas generales, esa variación es desestimable.

El apéndice C presenta un esquema de la navegación por la herramienta, mostrando las principales pantallas de interacción con el cliente y siguiendo las pautas establecidas en los distintos requerimientos funcionales planteados. En las distintas pantallas allí presentadas, se incluye, primeramente, la pantalla de inicio de la aplicación, que permite al usuario seleccionar el área de interés con la que desee trabajar; de allí, se muestra la pantalla principal de la herramienta, con el área de creación de grafos, el visor de nodos y la sección de propiedades de objetos entre otras cosas; finalmente se muestra la pantalla de creación de reportes, que presenta una interfaz que facilita el despliegue de la información deseada en el formato web; de manera adicional se muestran algunas pantallas intermedias, para la creación de estructuras de nodos y aristas, creación tabular de grafos y selección del tipo de componentes a utilizar, todas ellas accesibles en cualquier momento desde la pantalla principal.

El apéndice D presenta diversas imágenes de los reportes generados por la aplicación, basados en el estudio cienciométrico de los autores publicados en la revista Tekhne entre los años 1997 y 1999, así como un ejemplo del grafo allí trabajado.

## Conclusiones y recomendaciones

A continuación se presentan las principales conclusiones y recomendaciones vinculadas a los resultados y desarrollo del proyecto:

### 5.1. Conclusiones

#### 5.1.1. En cuanto a la investigación

- Algunos problemas planteados fácilmente como grafos no tienen una posible resolución determinística, por lo que deben ser resueltos de manera heurística, lo que a su vez requiere de un alto tiempo de cómputo acorde con la complejidad del problema.
- La investigación es una actividad previa obligatoria para el desarrollo de cualquier herramienta innovadora.

### 5.2. En cuanto al desarrollo de la aplicación

- El desarrollo de software debe ser un proceso sistémico y determinado por la calidad.
- La herramienta desarrollada puede ser extendida a corto plazo para un análisis cienciométrico extendido de gran valor en el desarrollo científico y la incorporación de reportes y algoritmos de análisis.
- La herramienta que se define y desarrolla debe ser escalable para considerar la incorporación de funcionalidades y áreas de aplicación.

### 5.3. Recomendaciones para trabajos futuros

Muchos temas científicos recurren a los planteamientos de la teoría de grafos como base para la resolución de situaciones que allí se puedan presentar, por ello, la realización de una investigación completa del tema de teoría de grafos y sus aplicaciones implica un arduo trabajo que requiere de la inclusión de muchas otras áreas del saber científico, por lo que es recomendable delimitar la investigación a áreas específicas de interés, con fines y procesos bien identificados, evitando así extenderse en otros temas relacionados pero sin relevancia para la investigación.

- Para futuras extensiones de la herramienta, es recomendable que la funcionalidad a agregar para cada nuevo tema sea analizada por expertos en el área, garantizando que el análisis del grafo brindado por la herramienta satisfaga las necesidades de los posibles usuarios en esa área de interés.
- Se recomienda la ampliación de la herramienta software agregando nuevos formatos de reporte, así como la creación de más estructuras de nodos, variando tanto su forma como su funcionalidad, de manera de hacer la aplicación más completa para el usuario.
- Se recomienda, además, la implantación de nuevos algoritmos que permitan el análisis de los grafos creados.

## BIBLIOGRAFÍA

- Algarra, P y Argilaga, M. (2003). *Aproximación al PERT en evaluación de programas desde las técnicas matemáticas de análisis de grafos*. Universidad de Barcelona. Consultado el 20 de octubre. Disponible en línea: [http://www.um.es/analesps/v09/v09\\_2/08-09\\_2.pdf](http://www.um.es/analesps/v09/v09_2/08-09_2.pdf)
- Anglés, R, y Matos, G. (2004). *Modelo de apoyo en la selección de la metodología para un desarrollo de software, basado en la gerencia del conocimiento y soportado en tecnología de información*. Venezuela: Universidad Católica Andrés Bello.
- Barrios, N., Alvarez, M., Marín, J., Valero, G., Baiz, R., González, R., Caldera, G. (2002). *Evaluación de Proyectos Informáticos*. Evaluación de Portales para la Empresa PETROVE, C.A. Venezuela: Universidad Católica Andrés Bello, Dirección de Formación Continua.
- Bertoa, M., Troya, J. y Vallecillo, A (2003). *Aspectos de calidad en el desarrollo de software basado en componentes*. Depto. Lenguajes y Ciencias de la Computación. Universidad de Málaga.
- Buschmann, Frank. Meunier, Regine. Rohnert, Hans. Sommerlad, Peter. Stal, Michael. (2000). *Pattern-Oriented Software Architecture: A System Of Patterns*. West Sussex, England: John Wiley & Sons Ltd. Consultado el 28 de octubre de 2005. Disponible en línea: <http://www.vico.org/pages/Patrons-Disseny/Pattern%20Layers/>
- Diestel, Reinhard. *Graph Theory*. Edición electrónica, 2005. Consultado el 22 de octubre de 2005. Disponible en línea: <http://www.math.uni-hamburg.de/home/diestel/books/graph.theory/GraphTheoryIII.pdf>
- Enciclopedia, *Dictionary.LaborLawTalk.com*. Consultada el día 24 de octubre de 2005. Disponible en línea: <http://encyclopedia.laborlawtalk.com>
- Enciclopedia, *Wikipedia, The Free Encyclopedia*. Consultada el día 24 de octubre de 2005. Disponible en línea: [http://en.wikipedia.org/wiki/Graph\\_\(graph\\_theory\)#Important\\_graphs](http://en.wikipedia.org/wiki/Graph_(graph_theory)#Important_graphs)
- Fussell, Mark L. (1996). *A good Architecture for Object-Oriented Information Systems. Structures, Designs and Patterns*. Publicaciones Chimu. Consultado el 28 de octubre de 2005. Disponible en línea: <http://www.chimu.com/publications/oopsla96tutorial23/oopsla96tutorial23.pdf>
- García Mireles, G. y Rodríguez Jacobo, J. (2001). *Aplicación del modelado de procesos en un curso de ingeniería de software*. Revista Electrónica de Investigación Educativa. Consultado el 20 de octubre de 2005. Disponible en Línea: <http://redie.ens.uabc.mx/vol3no2/contenido-mireles.html>
- Garlan, David. Shaw, Mary. (1994). *An Introduction to Software Architecture*. Escuela de Ciencias de la Computación, Universidad de Carnegie Mellon, Pittsburgh, Estados Unidos. Consultado el 28 de octubre de 2005. Disponible en línea: [http://rep1.iei.pi.cnr.it/~gnesi/matdid/intro\\_softarch.pdf](http://rep1.iei.pi.cnr.it/~gnesi/matdid/intro_softarch.pdf)
- Grimán, A. (2005). *Evaluación Arquitectónica de la Calidad de Software*. Venezuela: Universidad Simón Bolívar, División de Ciencias Físicas y Matemáticas.
- Jiménez, Luz E. (2003). *Comparación entre Java y .Net*. Universidad ICESI. Consultado el 15 de diciembre de 2005.
- Meza, Oscar & Ortega, Maruja(2004). *Grafos y Algoritmos*. Primera edición. Editorial Equinoccio. Universidad Simón Bolívar.
- Miranda, C. (2005). *Formulación de Estrategias de Tecnología de la Información en la Educación para el Desarrollo*. Consultado el 9 de noviembre de 2005. Disponible en línea: <http://www.civila.com/comun/presentaciones/261>
- Monzón, J. (2005). *Proyecto del Algoritmo de Fleury*. Universidad Politécnica de Madrid.

Consultado el 14 de octubre de 2005 en línea: <http://www.dma.fi.upm.es/fleury/definicionesGrafos.htm>

Ortiz, Lourdes y Sanchis, Francisco. (2004). La cienciometría como herramienta para la gestión del conocimiento. Su aplicación al caso de la investigación en Ingeniería del software. (MIFISIS'2004). Valladolid. España.

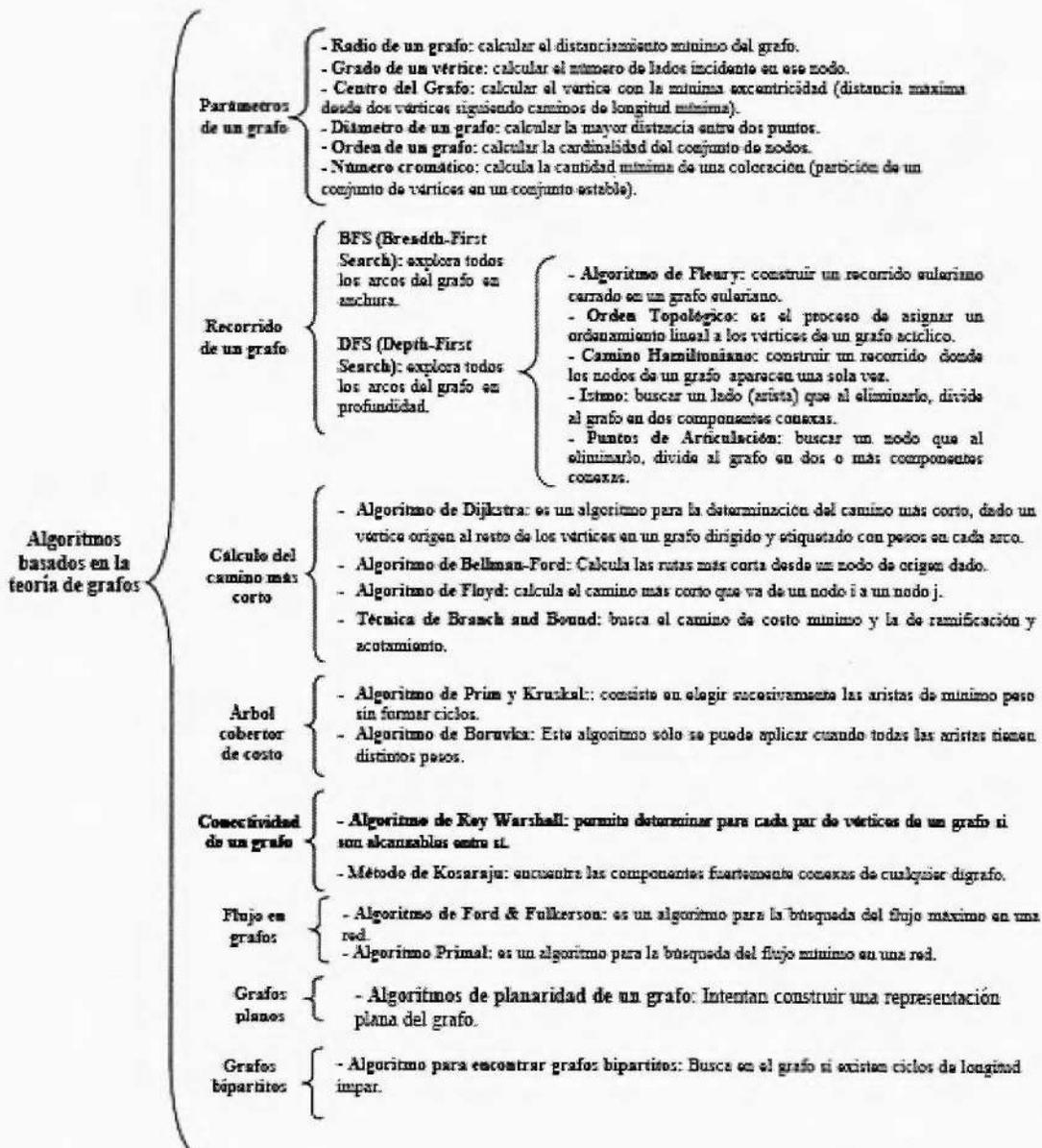
Publicaciones Honeywell. *Object-Oriented Software Architecture Styles*. Disponible en línea: [http://www.htc.honeywell.com/projects/dssa/dssa\\_what/dssa\\_oo\\_style.html](http://www.htc.honeywell.com/projects/dssa/dssa_what/dssa_oo_style.html)

Villaroel, R (2005). *Diseños de Sistemas*. Consultado el 28 de octubre de 2005. Disponible en línea: [http://www.eici.ucm.cl/Academicos/R\\_Villaroel/descargas/ing\\_sw\\_1/Arquitectura\\_Software.pdf](http://www.eici.ucm.cl/Academicos/R_Villaroel/descargas/ing_sw_1/Arquitectura_Software.pdf)

Weisstein, Eric. *Moore Graph*. MathWorld--A Wolfram. Consultado el 19 de octubre de 2005. Disponible en línea: <http://mathworld.wolfram.com/MooreGraph.html>

## Apéndices

APÉNDICE A: Algoritmos basados en la teoría de Grafos. Fuente: Elaboración propia a partir de Meza & Ortega, (2004) y de Monzón J. (2005)



	Factores de Análisis	Información básica del grafo	Tipos de Algoritmos (Funcionalidad)
Redes Sociales	<p>1) <i>Grado Nodal</i>: puede ser interpretado por la oportunidad de influir o ser influido directamente. Predice liderazgo, conocimiento, poder.</p> <p>2) <i>Intermediación</i>: Frecuencia en que un nodo aparece en el camino más corto que conecta a otros dos nodos.</p> <p>3) <i>Indicador de Popularidad (Eigenvector)</i>: el nodo que tiene la puntuación más alta es el que está conectado a muchos nodos bien conectados (grado del nodo).</p> <p>4) <i>Agujeros estructurales</i>: Nodos con lados de articulación, mientras más lados tenga más útil el nodo.</p> <p>5) <i>Solapamiento</i>: agrupación de nodos, tomando aquellos nodos que (a) no pertenezcan a un grupo o (b) pertenezcan a ambos, como nodos de solapamiento.</p> <p>6) <i>Red Egocéntrica</i>: una red "ego" es el sub grafo asociado a determinado nodo (el ego) con distancia 1.</p>	<p>- <i>Nodos</i>: Representan una persona (actor) o una organización.</p> <p>- <i>Aristas</i>: Representan una instancia de una relación social, puede ser de tipo: Parentesco, Basada en Roles, Cognitiva / Perceptual, Afectiva, De interacción, De afiliación</p>	<p>- Recorrer Matriz de adyacencia.</p> <p>- Recorrer Matriz de distancia.</p> <p>- Calculo del grado de un nodo.</p> <p>- Calculo de la densidad de un nodo.</p> <p>- Hallar los puntos de articulación.</p> <p>- Recorrer el grafo (DFS).</p>
Genética	<p>1) <i>Ensamblaje de secuencia</i>: El proceso de secuenciación del código genético de un organismo comienza "rompiendo" físicamente el ADN en millones de fragmentos aleatorios. La información contenida en cada uno de estos fragmentos se analiza experimentalmente. El problema surge a la hora de ensamblar de nuevo los fragmentos para formar una única secuencia completa.</p> <p>2) <i>Comparación de secuencias</i>. Cuando se descubre una secuencia nueva de ADN y se dispone de una base de datos de secuencias ya estudiadas, se plantea el problema de hallar la secuencia almacenada en la base de datos que más se parezca a la nueva secuencia.</p>	<p>- <i>Nodos</i>: Cada nodo representa una secuencia experimentalmente obtenida de un fragmento.</p> <p>- <i>Aristas</i>: Enlazan nodos si las secuencias correspondientes pueden ser agrupadas con otras.</p>	<p>- Algoritmos de "superposición-trazado-consenso"</p> <p>- Algoritmo EULER.</p> <p>- Hallar caminos hamiltonianos.</p>
Redes Computacionales	<p>1) <i>Aumentación de Grafos</i>: hallar la forma óptima de extender la red aumentando así su redundancia.</p> <p>2) <i>Congestión de la red</i>: puede darse en dos instancias, congestión de nodos o congestión de aristas, y en general, se refiere a la cantidad de caminos que circulan por un segmento de red, sea una arista o un nodo específico. La definición del camino viene establecida de acuerdo al protocolo utilizado.</p> <p>3) <i>Árbol Steiner</i>: se refiere al árbol de costo óptimo de la red.</p> <p>4) <i>Flujo de Red</i>: Viene definido por los tipos de conexiones utilizadas y los concentradores que unen las redes.</p> <p>5) <i>Ruta Mínima</i>: Se refiere a la ruta mínima entre dos puntos definidos.</p> <p>6) <i>Tiempo mínimo de BroadCast</i>: Se busca definir el tiempo mínimo que tarda un mensaje en llegar a todos.</p>	<p>- <i>Nodos</i>: Pueden ser, Terminales (Computadores) o Concentradores.</p> <p>- <i>Aristas</i>: representan las conexiones físicas de la red, varios tipos de cables o conexiones en caso de ser tecnología inalámbrica, con sus características de longitud y capacidad, entre otras.</p>	<p>- Recorrer el grafo (DFS).</p> <p>- Puntos de articulación.</p> <p>- Definir el flujo máximo de una red.</p> <p>- Hallar el árbol Steiner.</p> <p>- Hallar la ruta mínima (Dijkstra).</p> <p>- Determinar el grado de un nodo.</p> <p>- Determinar el árbol cobertor mínimo.</p>
Dinámica de Sistemas	<p>1) Construcción de modelos dinámicos que sirvan para el análisis de los impactos ambientales como en la gestión de los recursos naturales.</p> <p>2) Construcción de modelos dinámicos que sirvan de apoyo sobre las patologías del comportamiento humano tanto de forma individual como colectiva.</p> <p>3) Construcción de modelos dinámicos para mejorar la gestión de inventarios y cadenas de abastecimiento en general, no solo en procesos de fabricación.</p> <p>4) Construcción de modelos dinámicos que ayuden a comprender y diseñar políticas para manejar las complejas relaciones entre las diferentes funciones empresariales, y los resultados del negocio.</p>	<p>- <i>Nodos</i>: Representa una acumulación de flujos. Se pueden expresar en unidades.</p> <p>- <i>Aristas</i>: Representa un cambio de estado del sistema.</p> <p>Ejemplos de este tipo de variables son, producción y despachos, nacimientos y muertes, nuevos clientes y clientes que se pierden.</p>	<p>- Ecuaciones diferenciales.</p>
Manejo Proyectos	<p>1) PERT (Project Evaluation and Review Technique): es una técnica basada en redes que contemplan las relaciones de precedencia e interdependencia de actividades.</p>	<p>- <i>Nodos</i>: Representan los sucesos.</p> <p>- <i>Aristas</i>: Representan las actividades. Las flechas irán de izquierda a derecha indicando que se avanza en el tiempo.</p>	<p>- Algoritmo de Roy Warshal</p> <p>- Orden Topológico.</p> <p>- Calcular rutas críticas.</p>
Cienciometría	<p>1) <i>Autores por Área</i>: es ubicar a los autores según la especialidad de sus publicaciones.</p> <p>2) <i>País de publicación</i>: viene dado por el país en donde se registra la publicación, siendo indiferente a la nacionalidad del autor.</p> <p>3) <i>Frecuencia de citaciones por autor</i>: son las referencias que se hacen sobre un autor.</p>	<p>- <i>Nodos</i>: Representan publicaciones.</p> <p>- <i>Aristas</i>: Pueden existir 3 tipos de aristas dirigidas: una representa la referencia entre los artículos, otra la autocitación y la tercera a los artículos referidos en más de un artículo analizado.</p>	<p>- Matriz de adyacencia.</p> <p>- Recorrer Matriz de distancia.</p> <p>- Calcular el grado de un nodo.</p> <p>- Densidad de un nodo.</p> <p>- Puntos de articulación.</p> <p>- Recorrer el grafo (DFS).</p>



APÉNDICE D: Grafos y reportes generados por la aplicación.

The image displays three overlapping outputs from a software application. On the left is a network graph with nodes and connecting lines. In the center is a list of articles titled 'Artículos por año de Tekhne' with columns for year, title, and authors. On the right is a bar chart titled 'Gráfico' showing three bars of different heights.

**Artículos por año de Tekhne**

Año	Título	Autores
1998	Estudio Técnico-Económico de Operación de Control de Energía (Cercos a una mediana empresa)	Jessica Alvarez, Zulay Chung
	Verbetes de la sección de estado Papp- Robinson	Eng. J. Lacruce, N. Marquet
	Curso conceptual de distribución de una planta de fabricación de bobinas eléctricas	Francisco López, Walter Marquet
	Del mantenimiento preventivo al TPM	Ing. Armando Galo
	Controlador para la serie Multis	Lidia Romero, Manuel Barrios, Cecilia
	Módulo subprograma de la planificación de inventarios	Diego Casanovi
	Sistema de diagnóstico de vibraciones para equipos sobre situaciones críticas en instalaciones petroleras	Jose Pardo Escobar
	Sistema de control de distribución de humidificadores a nivel nacional de una empresa de Telecomunicaciones	Abu A. Odeh
	Estimación cuantitativa de efectos de los parámetros de proceso y la capacidad de la superficie y la impresión de resultados de manejo mecánico	
	Estudio Comparativo de Distribuidores Comerciales. Prácticas de	
Resumen en el área de estructuras para la Norma Antisísmica		
Recursos híbridos y desarrollo sustentable		
1997	Diagrama informático	Rogelio Ordoñez, Sila
	Actividad tecnológica e innovativa en Venezuela en periodo 1970-1990	Adriana Robles, Sila
	Uso de SDA (una oficina sobre los años)	Marcos Berti, Sila
1996	Operación conjunta de las bombas de los acederos Turb-Tur II en favor de la nueva operación de la línea de Soplamiento y el Embarco Guatirafa Edici	Walter Marquet, Sila
	Pagamento Justo	Enrique Anelli, Sila
	Análisis de los programas de Ingeniería Civil	Michael Brumack, Rodolfo Ordoñez
	Enfoque de la capacidad de embalse de un proceso industrial por métodos estadísticos	Ing. Roger Francisco, Nancy Lujan, Sila
1995	Módulo matemático para el diseño de equipos industriales por programación	
	El aprendizaje cooperativo de las fábricas de aprendizaje e su importancia	
	El terremoto de Caracas del 3 de Julio de 1997	
	Distribución espacial de nucleos microclimáticos	
1994	Propiedades para la recuperación espacial y el mejoramiento de la de carretera Carr. Boyerón I	
	Rapid Prototyping simultáneamente para el diseño de productos	
	La ingeniería concurrente: Facilita el trabajo en equipo	

**Gráfico**

El gráfico muestra tres barras representando datos para diferentes años. El eje vertical (Y) está etiquetado como 'Año' y tiene una escala de 0 a 12. El eje horizontal (X) está etiquetado como 'Año'. Las barras corresponden a los años 1995, 1996 y 1997, con valores aproximados de 12, 8 y 8 respectivamente.

Fuente: Elaboración propia.

	Factores de Análisis	Información básica del grafo	Tipos de Algoritmos (Funcionalidad)
Redes Sociales	<p>1) <i>Grado Nodal</i>: puede ser interpretado por la oportunidad de influir o ser influido directamente. Predice liderazgo, conocimiento, poder.</p> <p>2) <i>Intermediación</i>: Frecuencia en que un nodo aparece en el camino más corto que conecta a otros dos nodos.</p> <p>3) <i>Indicador de Popularidad (Eigenvector)</i>: el nodo que tiene la puntuación más alta es el que está conectado a muchos nodos bien conectados (grado del nodo).</p> <p>4) <i>Agujeros estructurales</i>: Nodos con lados de articulación, mientras más lados tenga más útil el nodo.</p> <p>5) <i>Solapamiento: agrupación de nodos, tomando aquellos nodos que (a) no pertenezcan a un grupo o (b) pertenezcan a ambos, como nodos de solapamiento.</i></p> <p>6) <i>Red Egocéntrica: una red "ego" es el sub grafo asociado a determinado nodo (el ego), con distancia 1.</i></p>	<p>- <i>Nodos</i>: Representan una persona (actor) o una organización.</p> <p>- <i>Aristas</i>: Representan una instancia de una relación social, puede ser de tipo: Parentesco, Basada en Roles, Cognitiva / Perceptual, Afectiva, De interacción, De afiliación</p>	<p>- Recorrer Matriz de adyacencia.</p> <p>- Recorrer Matriz de distancia.</p> <p>- Calculo del grado de un nodo.</p> <p>- Calculo de la densidad de un nodo.</p> <p>- Hallar los puntos de articulación.</p> <p>- Recorrer el grafo (DFS).</p>
Genética	<p>1) <i>Ensamblaje de secuencia</i>: El proceso de secuenciación del código genético de un organismo comienza "rompiendo" físicamente el ADN en millones de fragmentos aleatorios. La información contenida en cada uno de estos fragmentos se analiza experimentalmente. El problema surge a la hora de ensamblar de nuevo los fragmentos para formar una única secuencia completa.</p> <p>2) <i>Comparación de secuencias</i>. Cuando se descubre una secuencia nueva de ADN y se dispone de una base de datos de secuencias ya estudiadas, se plantea el problema de hallar la secuencia almacenada en la base de datos que más se parezca a la nueva secuencia.</p>	<p>- <i>Nodos</i>: Cada nodo representa una secuencia experimentalmente obtenida de un fragmento.</p> <p>- <i>Aristas</i>: Enlazan nodos si las secuencias correspondientes pueden ser agrupadas con otras.</p>	<p>- Algoritmos de "superposición-trazado-consenso"</p> <p>- Algoritmo EULER.</p> <p>- Hallar caminos hamiltonianos.</p>
Redes Computacionales	<p>1) <i>Aumentación de Grafos</i>: hallar la forma óptima de extender la red aumentando así su redundancia.</p> <p>2) <i>Congestión de la red</i>: puede darse en dos instancias, congestión de nodos o congestión de aristas, y en general, se refiere a la cantidad de caminos que circulan por un segmento de red, sea una arista o un nodo específico. La definición del camino viene establecida de acuerdo al protocolo utilizado.</p> <p>3) <i>Árbol Steiner</i>: se refiere al árbol de costo óptimo de la red.</p> <p>4) <i>Flujo de Red</i>: Viene definido por los tipos de conexiones utilizadas y los concentradores que unen las redes.</p> <p>5) <i>Ruta Mínima</i>: Se refiere a la ruta mínima entre dos puntos definidos.</p> <p>6) <i>Tiempo mínimo de BroadCast</i>: Se busca definir el tiempo mínimo que tarda un mensaje en llegar a todos.</p>	<p>- <i>Nodos</i>: Pueden ser, Terminales (Computadores) o Concentradores.</p> <p>- <i>Aristas</i>: representan las conexiones físicas de la red, varios tipos de cables o conexiones en caso de ser tecnología inalámbrica, con sus características de longitud y capacidad, entre otras.</p>	<p>- Recorrer el grafo (DFS).</p> <p>- Puntos de articulación.</p> <p>- Definir el flujo máximo de una red.</p> <p>- Hallar el árbol Steiner.</p> <p>- Hallar la ruta mínima (Dijkstra).</p> <p>- Determinar el grado de un nodo.</p> <p>- Determinar el árbol cobertor mínimo.</p>
Dinámica de Sistemas	<p>1) Construcción de modelos dinámicos que sirvan para el análisis de los impactos ambientales como en la gestión de los recursos naturales.</p> <p>2) Construcción de modelos dinámicos que sirvan de apoyo sobre las patologías del comportamiento humano tanto de forma individual como colectiva.</p> <p>3) Construcción de modelos dinámicos para mejorar la gestión de inventarios y cadenas de abastecimiento en general, no solo en procesos de fabricación.</p> <p>4) Construcción de modelos dinámicos que ayuden a comprender y diseñar políticas para manejar las complejas relaciones entre las diferentes funciones empresariales, y los resultados del negocio.</p>	<p>- <i>Nodos</i>: Representa una acumulación de flujos. Se pueden expresar en unidades.</p> <p>- <i>Aristas</i>: Representa un cambio de estado del sistema.</p> <p>Ejemplos de este tipo de variables son, producción y despachos, nacimientos y muertes, nuevos clientes y clientes que se pierden.</p>	<p>- Ecuaciones diferenciales.</p>
Manejo Proyectos	<p>1) PERT (Project Evaluation and Review Technique): es una técnica basada en redes que contemplan las relaciones de precedencia e interdependencia de actividades.</p>	<p>- <i>Nodos</i>: Representan los sucesos.</p> <p>- <i>Aristas</i>: Representan las actividades. Las flechas irán de izquierda a derecha indicando que se avanza en el tiempo.</p>	<p>- Algoritmo de Roy Warshal</p> <p>- Orden Topológico.</p> <p>- Calcular rutas críticas.</p>
Cienciometría	<p>1) <i>Autores por Área</i>: es ubicar a los autores según la especialidad de sus publicaciones.</p> <p>2) <i>País de publicación</i>: viene dado por el país en donde se registra la publicación, siendo indiferente a la nacionalidad del autor.</p> <p>3) <i>Frecuencia de citaciones por autor</i>: son las referencias que se hacen sobre un autor.</p>	<p>- <i>Nodos</i>: Representan publicaciones.</p> <p>- <i>Aristas</i>: Pueden existir 3 tipos de aristas dirigidas: una representa la referencia entre los artículos, otra la autocitación y la tercera a los artículos referidos en más de un artículo analizado.</p>	<p>- Matriz de adyacencia.</p> <p>- Recorrer Matriz de distancia.</p> <p>- Calcular el grado de un nodo.</p> <p>- Densidad de un nodo.</p> <p>- Puntos de articulación.</p> <p>- Recorrer el grafo (DFS).</p>