

**SOFTWARE EDUCATIVO PARA EL MANEJO GENERAL DE GRAFOS DE
FORMA DINÁMICA**

**ALEX DAVID BURGOS VELASCO
HECTOR FABIO GUZMÁN ESTUPIÑÁN**

**UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
PROGRAMA DE SISTEMAS
SAN JUAN DE PASTO**

2002

**SOFTWARE EDUCATIVO PARA EL MANEJO GENERAL DE GRAFOS DE
FORMA DINÁMICA**

**ALEX DAVID BURGOS VELASCO
HECTOR FABIO GUZMÁN ESTUPIÑÁN**

Trabajo de grado para obtener el título de Ingeniero de Sistemas

**UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
PROGRAMA DE SISTEMAS
SAN JUAN DE PASTO**

2002

**A Dios, quien me dio la fuerza
para salir adelante y a mi familia
por todos los sacrificios hechos
para que este sueño se haga
realidad.**

Alex David

A Dios, quien es mi guía en el camino de la vida y a mi familia por todo el apoyo brindado para salir adelante.

Héctor Fabio

NOTA DE ACEPTACION

Presidente del Jurado

Jurado

Jurado

AGRADECIMIENTOS A

Ingeniero Vicente Chamorro por su valiosa colaboración en cuanto a la pedagogía se refiere.

Ingeniero Juan Carlos Checa por los aportes en cuanto a la Teoría de Grafos.

Ingeniero Manuel Bolaños, asesor del proyecto quien nos dio las pautas necesarias en cuanto a la temática manejada.

A la Médica Mónica Lorena González y la Geógrafa Alexandra Muñoz por su valiosa amistad, colaboración y apoyo incondicional.

TABLA DE CONTENIDO

	Pág.
Introducción	1
1. ELEMENTOS DE IDENTIFICACIÓN	6
TITULO	6
DEFINICIÓN DEL PROBLEMA	6
OBJETIVOS	7
Objetivos generales	7
Objetivos específicos	7
DELIMITACIÓN Y ALCANCES	8
ANTECEDENTES	8
JUSTIFICACIÓN	10
2. MARCO TEÓRICO	12
MARCO HISTÓRICO	12
Historia de los computadores	12
Nacimiento de los primeros lenguajes	16
MARCO CONCEPTUAL	18
Formas sistemáticas para crear ambientes de aprendizaje	20
Enfoque Educativo algorítmico	21
Enfoque Educativo Heurístico	22

Modelos pedagógicos y teorías que sustentan el diseño de ambientes de enseñanza – aprendizaje.	24
Pedagogía tradicional	27
Pedagogía conductista	28
Pedagogía Romántica	28
Pedagogía Desarrollista	29
Pedagogía constructivista	29
Pedagogía de Gestalt	30
Psicología evolutiva de Jean Piaget	31
Pedagogía algo – Heurística	32
Recursos educativos en el proceso de aprendizaje	34
El computador como estrategia de aprendizaje didáctico	35
Informática educativa	37
Software Educativo	38
Clasificación General de software educativo	40
Tutoriales	40
Simuladores y Juegos educativos	41
Constructores	41
Sistemas de ejercitación y práctica	42
Lenguajes sintónicos y micro mundos explorativos	43
Sistemas expertos con fines educativos	43
Ambientes de aprendizaje asistidos por computador	44

EI MEC	44	
2.2.6.1.1. Herramientas y métodos para generar MECs		45
Estructuras de Información	48	
Grafos	50	
Definición de Grafos		50
Definición de Digrafos		51
Áreas de aplicación		52
Conceptos Básicos		53
Tipos de grafos		56
Grafos Débil y Fuertemente Conectados	56	
Grafos Eulerianos y Hamiltonianos	57	
Otros tipos de Grafos	58	
Almacenamiento de un grafo en la memoria de un computador		60
Recorridos de grafos		61
Recorrido de un grafo a lo Ancho	61	
Recorrido de un Grafo en profundidad		62
Aplicaciones de Grafos		62
Algoritmo de Prim		62
Algoritmo de Dijkstra	64	
Algoritmo de Warshall	66	
Algoritmo de Kruskal	67	
Algoritmo de Floyd		69

3. METODOLOGÍA	71
CICLO DE VIDA DEL DESARROLLO DE UN PROYECTO	72
Análisis	73
Diseño	75
Desarrollo	79
Prueba piloto del MEC	82
Prueba de campo del MEC	84
ANÁLISIS DE RIESGOS	85
Identificación de riesgos	86
Riesgos de Proyecto	86
Riesgos Técnicos	87
Riesgos del Negocio	89
Proyección o estimación de los Riesgos	90
Naturaleza De Los Riesgos	92
Prioridad del Riesgo	93
Relación riesgo – probabilidad e impacto	93
Niveles de referencia	94
3.2.5.1.1. Relación de las ternas con los niveles de referencia	95
Gestión y supervisión de riesgos	97
MÉTRICAS DE CALIDAD	100
Métricas de calidad del software	100
Métricas de calidad del área temática	104

ESTRATEGIAS DE SOLUCIÓN	107
Análisis de la solución	107
Desarrollo de la solución	109
Alternativas de solución para el desarrollo del software	115
Cronograma de actividades	116
4. GESTIÓN DEL PROYECTO	117
ENFOQUE Y TIPO DE INVESTIGACIÓN	117
NECESIDAD EDUCATIVA	118
Fuentes para la determinación de la necesidad educativa	120
Tipo de muestra	121
Población Objetivo	122
Área de contenido	123
Instrumentos para la recolección de información	124
ANÁLISIS DE PROBLEMAS EXISTENTES	125
DISEÑO DEL MEC	133
Selección de temática del MEC	133
Análisis de tareas de aprendizaje	134
Diseño comunicacional	136
Definición de las zonas de comunicación	138
Elementos constitutivos de las zonas de Comunicación	139
Acerca de los menús	139
Acerca del texto	142

Apoyo grafico	147
Diseño computacional	149
Estructura lógica para la interacción	149
Diagramas de Procesos (Análisis de flujos de información)	151
Diccionario de datos 1	151
Estructuras de datos	151
Tablas	152
Diccionario de datos 2	154
Diagramas entidad – relación	162
CONCLUSIONES	163
RECOMENDACIONES	165
BIBLIOGRAFÍA	167
ANEXOS	

LISTA DE FIGURAS

		Pág.
Figura 1.	Interfaz de herramienta de manejo de grafos AlgraF	10
Figura 2.	Esquema de la teoría del reflejo de Pavlov	28
Figura 3.	Representación gráfica de un grafo	50
Figura 4.	Rrepresentación gráfica de un dígrafo	51
Figura 5.	Representación Adyacencia	53
Figura 6.	Representación Incidencia	54
Figura 7.	Grado de un Vértice	55
Figura 8.	Grado de un Vértice (Digrafos)	55
Figura 9.	Grafos Fuerte y Débilmente Conectados	56
Figura 10.	Grafos Eulerianos y hamiltonianos	58
Figura 11.	Almacenamiento De Un Grafo En Memoria	60
Figura 12.	Recorrido De Un Grafo A Lo Ancho	51
Figura 13.	Algoritmo De Prim	63
Figura 14.	Algoritmo De Dijkstra	65
Figura 15.	Algoritmo De Floyd	70
Figura 16.	Ciclo de Vida del Desarrollo de un Proyecto	72
Figura 17.	Interfaz Inicio De Sesión	144
Figura 18.	Entorno De Trabajo Principal	145

Figura 19.	Entorno De Práctica Y Ejercitación De SEGraD	146
Figura 20.	Apoyo Grafico (Simulación Algoritmo De Prim)	148

LISTA DE CUADROS

		Pág.
Cuadro 1.	Historia de los lenguajes de programación	16
Cuadro 2.	Clasificación de los materiales educativos	
	Computarizados	24
Cuadro 3.	Modelo Analítico	33
Cuadro 4.	Riesgos del Proyecto	91
Cuadro 5.	Riesgos Técnicos	91
Cuadro 6.	Riesgos del Negocio	92
Cuadro 7.	Prioridad de los Riesgos	93
Cuadro 8.	Análisis de Tareas de Aprendizaje	135
Cuadro 9.	Textos Manejados en SEGraD	143

GLOSARIO

Adyacencia: Se dice que existe adyacencia entre dos vértices si están unidos por una arco. Para el caso de los Dígrafos o grafos dirigidos existen dos tipos de adyacencia: Adyacencia desde y Adyacencia hasta.

Estructuras de Información: Conjunto de datos cuya organización se caracteriza por las funciones de acceso que se usan para almacenar y localizar a elementos individuales de datos

Grado de un Vértice: El grado de un vértice es el número de arcos que inciden en ese vértice.

Grafo: Conjunto de vértices (puntos) conectados entre sí por medio de arcos (líneas). Representación gráfica de una estructura.

Hardware: Parte física de un computador, lo que se puede tocar. Hacen parte del hardware: Pantalla, teclado, mouse, etc.

Incidencia: Los arcos inciden en los vértices. Un arco incide en un vértice si una de sus puntas llega a ese vértice.

Listas de Adyacencia: Son listas que se desprenden de los nodos de una lista principal, a manera de ramas.

MEC: Material Educativo Computarizado.

Software: Son los programas del computador, enormes listados de instrucciones que le indican al computador cuales son las tareas que debe desempeñar; sin Software un computador es una máquina inservible, sin vida.

Software Educativo: Programa que permite cumplir y/o apoyar funciones educativas, brindando al usuario la oportunidad de un aprendizaje fácil, rápido y divertido.

RESUMEN

Una buena capacitación de los estudiantes en cada una de las ramas ofrecidas, es el objetivo principal de toda entidad de Educación Superior, con el fin de formar profesionales competentes.

SEGraD es una herramienta innovadora que aprovecha las facilidades que brinda el avance tecnológico actual, para apoyar el aprendizaje en el área de Teoría de Grafos de manera interactiva y amigable, desarrollando en el estudiante aptitudes creativas y agilizando el proceso de asimilación de la información. Su estructura se basa en cuatro módulos a saber: Administrativo, que permite al docente el mantenimiento general del software; Apoyo al Aprendizaje, que es donde se transmite la información sobre los temas de estudio; Ejercitación, que permite al estudiante aplicar los conocimientos adquiridos y experimentar sobre cada tema; y por último el módulo de Evaluación, donde el estudiante podrá demostrar cuanto ha aprendido. Además, SEGraD no sólo es una herramienta para aprendices, también es de gran utilidad a personas idóneas en el tema, ya que posee una herramienta para solución de problemas que requieran el uso de los temas ofrecidos.

SUMMARY

The main target of the Superior Education is to enable the students, in each one of the offered branches, aiming to form competent professionals.

An innovate tool called SEGraD which takes advantage of the facilities that the present technology offers in order to support the learning in the area of the graphos (vertexes joined by arcs), in an interactive an easy way, developing creative aptitudes on the students, as well as getting the process of assimilation of the information. Its structure is based on four modules described bellow: administrative, which lets the teacher to try on the general upkeeping of the software; SEGraD support the learning, where the information of different subjects are transmitted; training, which lets the student to put into practice the obtain knowledge, and to try it on every theme too; finally the education module, where the student will demonstrate what he has learned. SEGraD it's not only a tool for apprentices, but also it is very useful to people who are capable on this subject because SEGraD has a tool to solve problems that require the use of any of the offered topics.

INTRODUCCIÓN

Actualmente la Informática no puede ser una asignatura más, sino una herramienta que pueda ser útil a todas las materias, a todos los docentes y a las instituciones mismas que necesiten una organización y deban poder comunicarse con la comunidad en la que se encuentran.

Entre las aplicaciones más destacadas que ofrecen las nuevas tecnologías se encuentran el MEC (Material Educativo Computarizado) que se inserta rápidamente en el proceso de la educación. Ello es así porque refleja a cabalidad la manera en que el alumno piensa, aprende y recuerda, lo cual permite explorar fácilmente palabras, imágenes, sonidos, animaciones y videos e intercala pausas para analizar, reflexionar e interpretar en profundidad la información. Esto personaliza la educación al permitir a cada aprendiz avanzar según su propia capacidad. No obstante la mera aplicación de los MECs no asegura la formación de mejores alumnos y futuros ciudadanos.

Ahora, entre otros requisitos, los procesos deberán ir guiados y acompañados por el docente, quién seleccionará con criterio el material a estudiar, establecerá una

metodología de estudio, de aprendizaje y evaluación, que no convierta la información brindada a través de un CD-ROM en un simple libro animado, donde el alumno consuma grandes cantidades de información que no aporten demasiado a su formación personal.

Por encima de todo, el docente tendrá la precaución no solo de examinar cuidadosamente los contenidos de cada material a utilizar para detectar posibles errores, omisiones, ideas o conceptos equívocos, sino que también deberá fomentar entre los alumnos una actitud de atento juicio crítico frente a ello. Sin embargo la informática educativa carece aún de estima en influyentes núcleos de la población, porque se crean serios baches educativos que resultan difíciles de resolver y que finalmente condicionan el desarrollo global de la sociedad.

La mejoría en el aprendizaje resulta ser uno de los anhelos más importantes de todos los docentes; de allí que la enseñanza individualizada y el aumento de productividad de los mismos son los problemas críticos que se plantean en la educación.

El aprendizaje se logra mejor cuando es activo, es decir, cuando cada estudiante crea sus conocimientos en un ambiente dinámico de descubrimiento. La duración de las clases y la metodología empleada en la actualidad son factores que conducen a un aprendizaje pasivo. Dado que la adquisición de los conocimientos

no es activa para la mayoría de los estudiantes, la personalización hace que los docentes dediquen más tiempo a los estudiantes en forma individual o en grupos pequeños.

“Sólo cuando cada estudiante se esfuerza por realizar tareas, podemos prestarle atención como individuo”.

Por otra parte la incorporación de nuevos avances tecnológicos al proceso educativo necesita estar subordinada a una concepción pedagógica global que valore las libertades individuales. Sería loable la igualdad de oportunidades, hitos trascendentes en la formación de las personas, con vistas a preservar en la comunidad los valores de la verdad y la justicia. La computadora es entonces una herramienta, un medio didáctico eficaz que sirve como instrumento para formar personas libres y solidarias.

Así, es muy importante entender el verdadero significado de lo que representa y lo que acarrea el desarrollo de un Software Educativo.

En el Libro “Ingeniería del Software Educativo” los autores Alvaro H. y Galvis Panqueva, denominan como Software Educativo (MEC) a aquellos programas computarizados que de alguna manera contribuyen al proceso de aprendizaje; un material educativo computarizado, es un ambiente informático que le permite al

estudiante tener experiencias y resultados favorables en su proceso de aprendizaje, que correspondan a una necesidad educativa específica. El MEC se convierte entonces en una herramienta complementaria destinada a apoyar el proceso de aprendizaje para el cual ha sido diseñado.

Con base en lo anterior presentamos al comité curricular de la facultad de ingeniería un proyecto enfocado al diseño de un MEC, que intentará contribuir al desarrollo de esta área en nuestro medio universitario.

A pesar que en el pensum que nos correspondió desarrollar, en ninguna de las asignaturas se contempló el tratamiento teórico de las corrientes y enfoques pedagógicos, porque comprendemos que ése no es el perfil profesional de la carrera, nos hemos visto ante la necesidad de realizar investigaciones en tal sentido, ya que el aspecto pedagógico es la médula espinal de todo trabajo que pretenda ser o tener un destino didáctico. Debemos manifestar, que no siendo nuestra especialidad ni lo pedagógico ni lo didáctico, sí se tienen claros los soportes conceptuales necesarios para el desarrollo de nuestro proyecto, con las lógicas limitaciones, ya que consideramos que un trabajo de esta índole requiere de la acción de un grupo interdisciplinario, en el cual intervengan: un pedagogo, un experto en los conocimientos específicos de la asignatura a desarrollar y un programador o diseñador del software, como mínimo.

En la fase de análisis y diseño de este proyecto, se tuvo en cuenta enfoques pedagógicos, particularmente el algorítmico y heurístico, que nos dieron pautas esenciales para el diseño del material educativo computarizado, todo esto enmarcado dentro de los parámetros de la Ingeniería del Software Educativo.

El objetivo fundamental del Software Educativo SEGraD, es el de constituirse en un instrumento complementario que apoye el proceso de aprendizaje de la Teoría de Grafos.

No podemos afirmar que sea un instrumento autónomo que por sí mismo pueda conducir a un aprendizaje eficaz, ya que requiere de un posibilitador, facilitador o instructor o como quiera llamarse, que le permita al estudiante explorar las posibilidades que SEGraD le brinda, además de una verdadera actitud de querer aprender por parte del aprendiz y de apoyar por parte del posibilitador.

1. ELEMENTOS DE IDENTIFICACIÓN

TITULO

SOFTWARE EDUCATIVO PARA EL MANEJO GENERAL DE GRAFOS DE
FORMA DINÁMICA

1.2. DEFINICIÓN DEL PROBLEMA

Actualmente la Universidad de Nariño no cuenta con una herramienta que facilite el aprendizaje interactivo de la Teoría de Grafos. Los recursos empleados para la enseñanza de la Teoría de Grafos en la asignatura Estructuras de Información, no son lo suficientemente profundos y eficaces, si tenemos en cuenta la complejidad del tema y el soporte que nos puede brindar el avance tecnológico.

1.3. OBJETIVOS

1.3.1. Objetivo General

Desarrollar e implementar una aplicación computarizada que permita el aprendizaje de la Teoría de Grafos de una manera interactiva, práctica y eficaz.

1.3.2. Objetivos Específicos

- Presentar el perfil general de las Estructuras de Información.
- Dar a conocer los conceptos generales de la Teoría de Grafos.
- Enumerar y explicar los diversos algoritmos que se incluyen en la teoría de grafos.
- Describir el funcionamiento de cada uno de los algoritmos.
- Desarrollar un módulo de ejercitación y práctica para cada uno de los algoritmos propuestos.
- Desarrollar un módulo de evaluación que mida el grado de aprendizaje de los usuarios.
- Desarrollar un software educativo de fácil acceso, que sirva de apoyo y profundización para cualquier tipo de usuario interesado.

1.4. DELIMITACIÓN Y ALCANCE DEL PROYECTO

El software educativo, se enfocará en el desarrollo e implementación de los diversos apartados que componen lo que se conoce como Teoría de Grafos, que a su vez, hace parte de la temática de la asignatura Estructuras de Información del programa de Ingeniería de Sistemas de la Universidad de Nariño.

El software manejará de forma dinámica las definiciones, fundamentos, componentes, creación, recorridos, de dichos apartados, los cuales especificamos a continuación:

Conceptos básicos de grafos, tipos de grafos, creación de una lista de adyacencia, recorridos de un grafo, y los algoritmos de Warshall, Prim, Dijkstra, Floyd, y de Kruskal. La aplicación cumplirá con todos los requisitos que conlleva un Software Educativo.

1.5. ANTECEDENTES

De acuerdo a las revisiones realizadas sobre el desarrollo del Software Educativo, en nuestra región y sociedad Nariñense no ha sido posible encontrar documentación relacionada a la temática Que SEGraD maneja, por otro lado si hemos descubierto que en relación con las estructuras de datos existen

programas que, aunque mal llamados Software Educativos, utilizan una parte de la metodología que nosotros utilizamos, lo cual puede ser útil para el desarrollo del presente proyecto.

A nivel mundial y gracias al Internet descubrimos un software desarrollado en una universidad de España, lejos de ser un Software Educativo, más bien hace las veces de Herramienta Case y que todavía se encuentra en la etapa de desarrollo por tratarse también de un proyecto de grado, el software encontrado se denomina "Algraf - Algoritmos sobre Grafos", y que como sus autores explican, es un software que permite dar resolución a grafos por medio de los algoritmos que la teoría de Grafos plantea, pero que no tiene en cuenta ninguna metodología de enseñanza.

Al igual que SEGraD, Algraf maneja una interfaz muy amigable donde el usuario lo primero que visualiza es un menú estandarizado (Windows), fácil de manejar y desde donde se tiene acceso a todas las opciones existentes. El desarrollo del software es en lenguaje de programación Visual Basic 5.0 y por lo que nos pudimos dar cuenta lleva más o menos un 70% de su desarrollo total.

Interfaz De Algraf

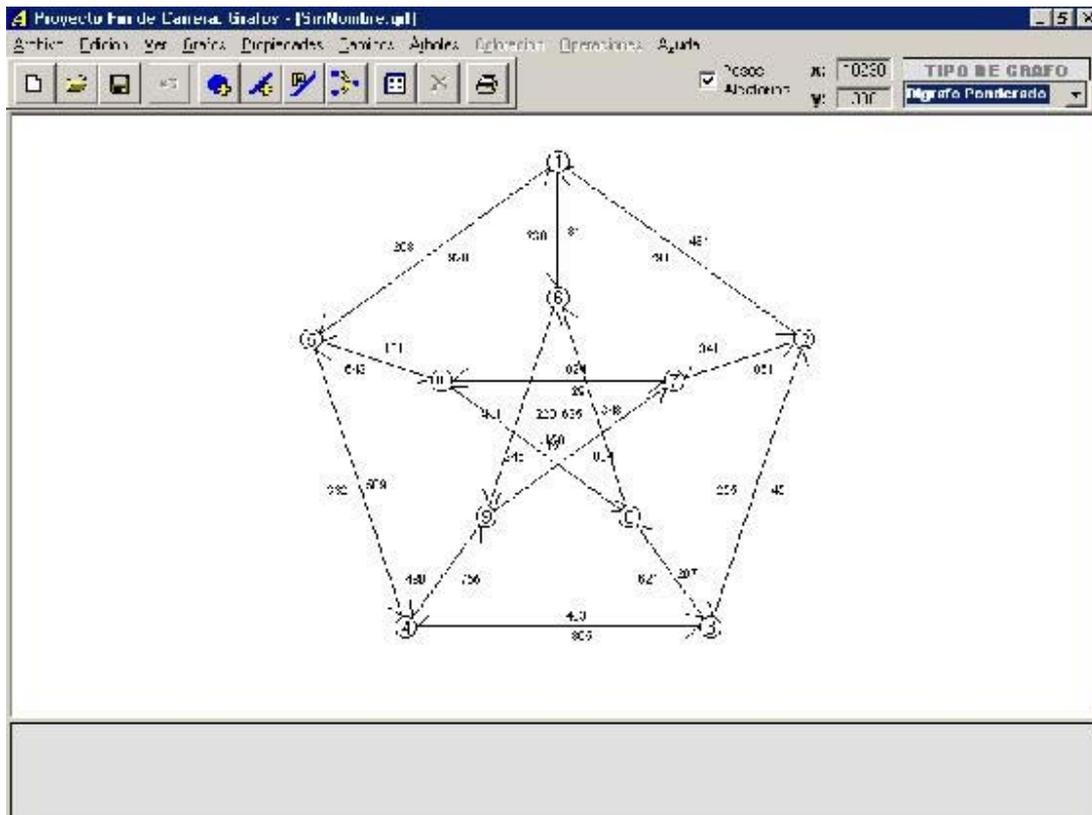


Figura 1.

INTERFAZ DE HERRAMIENTA DE MANEJO DE GRAFOS ALGRAF

1.6. JUSTIFICACIÓN

Toda entidad de Educación Superior tiene como objetivo principal, el capacitar a sus estudiantes en cada una de las ramas que brinda para llevar a cabo la

formación de profesionales competentes. Para ello deben mantener una constante evolución en todos los aspectos, principalmente en las metodologías, las cuales, a través de los años, han demostrado ser dependientes del desarrollo de la ciencia y la tecnología.

Normalmente, si se desea realizar cualquier tipo de consulta, la mayor parte de la información es posible encontrarla en medios tales como libros, apuntes, Internet, etc., cuyo acceso puede ser restringido y tedioso, sin la capacidad innovadora y llamativa que estimule a los interesados a conocer ó profundizar acerca del tema. Por ello y teniendo en cuenta que la Teoría de Grafos es parte esencial de la asignatura Estructuras de Información y que además, juega un papel importante en la toma de decisiones mediante Métodos de la Ruta Crítica y métodos de Cola o Espera, se propone el desarrollo de una herramienta por medio de un lenguaje visual que sirva de apoyo para el aprendizaje interactivo del tema en cuestión, de manera práctica, fácil y eficaz.

En nuestro medio no existe un Software Educativo que permita el aprendizaje de la teoría de grafos, por ello, consideramos que desarrollar esta herramienta hace parte de una propuesta innovadora e interesante.

2. MARCO TEÓRICO

2.1. MARCO HISTÓRICO

La evolución de los sistemas de computación han originado cambios importantes en la vida actual, en las diferentes áreas tales como la industria, la educación , las comunicaciones, las culturas de los pueblos en general. Por lo tanto conocer como fue su origen, sus cambios y su progreso permitirá tener una noción de los hechos más sobresalientes.

2.1.1. Historia de los Computadores

La evolución de los computadores puede relatarse en miles y miles de páginas, se inicia desde el mismo momento de la aparición del hombre, por lo tanto se hará énfasis en los aspectos más relevantes de la investigación.

Un artículo de Internet con respecto a la evolución histórica de los computadores menciona lo siguiente: “El primero y más perfecto computador que ha existido y

existirá es el cerebro humano, éste es un computador natural no construido por el hombre, normalmente siempre los historiadores se han preocupado de la evolución de los computadores, olvidándose que el cerebro fue el primero”.¹

La humanidad siempre ha pretendido construir un computador a imagen y semejanza del cerebro, debido a esto, hoy en día se habla de cerebros electrónicos, pero hasta ahora no se han logrado. En ese afán de construir un dispositivo que simule al cerebro, los científicos de los primeros tiempos utilizaron cuanto objeto se les ocurrió que pudiera servirles para procesar datos, hasta que en el 5000 a.C apareció presumiblemente en china el “ábaco”, considerado el primer computador refinado construido por el hombre.

La revista Byte² hace énfasis en los más grandes descubrimientos hasta antes de la aparición de los microcomputadores y clasifica la evolución por generaciones: 1963, William Oughtree inventa la regla de cálculo, 1642, Blaise Pascal, matemático y filósofo francés construyó la primera máquina sumadora usando ruedas y engranajes, en su honor a uno de los primeros lenguajes de programación se le dio el nombre de Pascal.

1833, Charles Babbage, trató de inventar la “máquina analítica”, pero la tecnología de la época no le permitió construir lo que se denominó la “locura de Babbage”,

¹ www.Civila.com/Sleiker/hilos.html

esta locura se quedo en papel pero definió todos los elementos básicos de un computador, así: Unidad aritmética y lógica, almacenamiento y unidad de control, por esto a Babbage se le considero como el padre de los computadores.

Después de Babbage vinieron muchos otros inventos que contribuyeron para que en 1944, se terminara de construir el primer computador electromecánico, el "MARK-1".

En 1946 J. Presper Eckert y John W. Mauchly de la Universidad de Pensilvania construyeron el primer computador electrónico, el "ENIAC (Electronic Numerical Integrator and Calculator)", basado en los estudios de John Vincent Atana Soft. Este no tenia nada parecido a los computadores de hoy, fue construido con 20.000 tubos de vacío, pesaba 30 toneladas, requería 150.000 Vatios de energía, realizaba 5.000 operaciones por segundo y ocupaba completamente uno de los sótanos de la Universidad.

Así comenzó la primera generación de computadores. Mas tarde en 1951 se libero la primera computadora comercial, el "ENIVAC-1", luego en 1959 con el uso del transistor en reemplazo de los tubos al vacío, aparecieron los computadores transistorizados, considerada la segunda generación, además nacen con ellos los lenguajes de programación de alto nivel como FORTRAN, LISP y COBOL. En

² Revista Byte Vol. 10 No 9

1964, la IBM anuncio los computadores de la serie 360, con lo que se da inicio a la tercera generación, estos ya usaban circuitos integrados o chips.

La cuarta generación se caracteriza por el avance en los equipos periféricos, los cuales aumentaron la velocidad de entrada y salida de datos e información, en esta época aparecen los primeros microcomputadores y fue el comienzo de lo que se puede denominar la quinta generación. De aquí en adelante la evolución de los computadores solo podría ser descrita utilizando las fechas de anuncios y apariciones en el mercado de las diferentes marcas y modelos, debido a la rápida expansión del mercado en estos equipos. En el fondo, en ésta evolución no se han producido cambios substanciales, más bien han sido cambios de forma, tamaño, capacidad de almacenamiento, rapidez de procesamiento, utilidades y precios.

Todas las liberaciones a nivel de hardware y software en forma que podría decirse desordenadas han reventado en una realidad latente para los usuarios, la necesidad de estandarización de la tecnología informática, iniciando por los componentes básicos de ella: Hardware y Software, envueltos día a día en cambios de velocidad de procesamiento y disponiendo cada uno de mayores capacidades de almacenamiento para los datos.

Sigue siendo característica de la evolución de los computadores, desde 1948 el que se siguen haciendo cada vez más pequeños, más rápidos y menos costosos.

Realmente en la actualidad la concepción de los computadores expresado en términos de microcomputadores, desaparecerá y en el futuro se hablará de estaciones de trabajo, donde una estación de trabajo es un microcomputador con todas las facilidades de procesamiento de datos conectados a una red.³

2.1.2. Nacimiento de los Primeros Lenguajes

Los lenguajes de programación han ido evolucionando a la par con los avances en los computadores, pasando desde los primeros de alto nivel, tipo carácter, hasta los lenguajes visuales orientados a objetos y eventos que crearon un nuevo estilo de desarrollar software. Veamos una tabla de relación de los lenguajes que marcaron el inicio del software de computadores:⁴

HISTORIA DE LOS LENGUAJES DE PROGRAMACIÓN				
LENGUAJE	AÑO	CREADOR	PREDECESOR	PROPOSITO
FORTRAN	1954 – 1955	J. Backus	--	Calculo numérico
ALGOL 60	1958 – 1960	Comité	FORTRAN	Calculo numérico
COBOL	1959 – 1960	Comité	--	Gestión

³ Correa Uribe, Guillermo. Desarrollo de algoritmos y sus aplicaciones P.4

⁴ www.ei.cs.ut.edu/~history/vonnewman.html

LIST	1956 – 1962	J. McCarthy	--	Calculo simbólico
PL / I	1963 – 1964	Comité IBM	FORTTRAN ALGOL 60 COBOL	Propósito general
ALGOL 68	1963 – 1968	Comité	ALGOL 60	Propósito general
PASCAL	1971	M. Wirth	ALGOL 60	Propósito general, educación: soporte de programación estructurada
PROLOG	1972	Colmenaver Roussel	ALGOL 68	Programación lógica
C	1974	D. Richie	ALGOL 68 BCPC	Programación de sistemas
PASCAL CONCURRENTE	1975	Brinch Hansen y	PASCAL	Programación concurrente
MODULA	1977	M. Wirth	PASCAL	Programación de sistemas tiempo real
ADA	1979	J. Tchbiah y otros	PASCAL	Programación general
MODULA – 2	1980	M. Wirth	MODULA	Programación de sistemas en tiempo real
C++	1982	Stroustrup	SIMULA 67 C	Programación orientada a objetos

Cuadro 1.

HISTORIA DE LOS LENGUAJES DE PROGRAMACIÓN

Algunos de los motivos que influyeron en los lenguajes de alto nivel son: La arquitectura de Von Newman (1946) basada de alguna forma en el diseño de la

máquina de Turing (1936), ha sido el modelo básico de la arquitectura durante mucho tiempo y a influido enormemente en el diseño de los primeros lenguajes, así como la programación de sistemas. El Fortran y el Algol fueron influidos por la notación algebraica ordinaria; el idioma inglés influyó en el diseño del Cobol; los estudios sobre el cálculo lambda realizados por Church (1941) pusieron las bases para la programación funcional del Lisp.

Para el desarrollo de software, actualmente las necesidades del usuario deben estar determinadas y establecidas explícitamente. Los primeros programadores consideraban que la base de la programación era la experiencia, cuando los programas se complican y resulta muy complejo depurarlos y modificarlos, los programadores empiezan a pensar en una metodología sistemática de programación, esto hace que la evolución de los lenguajes se haya visto muy marcada por un uso cada vez mayor de la “abstracción”, tanto de datos como de estructuras de control.

2.2. MARCO CONCEPTUAL

El presente trabajo de Grado, fundamentará sus diversos conceptos básicos teniendo en cuenta las siguientes bases teórico – técnicas: Mediante la teoría general de grafos, originalmente estudiada por Euler en el conocido acertijo de los

puentes de Königsberg, y posteriormente estudiada por el matemático alemán D. König, quién bautizo a los "grafos" con ese nombre, analizamos las potenciales aplicaciones, clarificaremos las ventajas, puntos críticos y desventajas que se puedan presentar.

Este tipo de estructuras de datos y las metodologías que manejan pueden ser aplicadas a todo tipo de ciencias e ingenierías donde el diseño forma parte esencial, como la arquitectura, la ingeniería civil y de sistemas entre otras. Utilizando este sistema de análisis, en las primeras instancias de cualquier proyecto, podrían evitarse situaciones indeseables generadas por una falta de sistematización clarificadora en los procesos, que se torna más necesaria, cuanto más elevada es la complejidad del diseño en desarrollo. Por ejemplo, el proceso de diseño arquitectónico plantea diversos problemas a resolver: relaciones espaciales, circulaciones, direccionalidad de recorrido, funcionalidad, estética, interconexiones y demás variables que lo transforman en un proceso altamente complejo. Los Grafos me dan las pautas para un diseño coherente y óptimo de los espacios y los recorridos.

El Software y El Hardware, hacen posible que los procesos en las aplicaciones anteriormente mencionadas tengan sentido. Software, en él radica la mayor parte de la magia que ha convertido al computador en la herramienta más poderosa de nuestros tiempos, también conocido como soporte lógico, que junto al soporte

físico, Hardware, parte física (lo que se puede tocar, ej. Pantalla, teclado, mouse etc.,) forman un todo, permitiendo procesar y ejecutar un sin número de instrucciones y actividades.

“El software, son los programas del computador, enormes listados de instrucciones que le indican al computador cuales son las tareas que debe desempeñar; sin Software un computador es una máquina inservible, sin vida”.

A diferencia del Hardware, el software no es algo físico, no se puede tocar, lo único tangible son los medios donde se almacena, como discos duros, CDs, micro discos, entre otros. Existen diversas clases de programas de computador; los dos principales grupos son: el Sistema Operativo y los tipo Aplicativo. Del primero de ellos depende la ejecución óptima de la mayoría de tipos de programas y aplicativos.

2.2.1 Formas Sistemáticas para Crear Ambientes de Aprendizaje

La actividad educativa relacionada con la incursión de la informática y los computadores para asistir el aprendizaje se enmarca bajo dos enfoques propuestos por Thomas Dwyer: El enfoque Algorítmico y el enfoque Heurístico. El

computador usado racionalmente y de manera planificada aumentará el rendimiento de la educación en términos de su calidad.

2.2.1.1. El Enfoque Educativo Algorítmico

Guía la forma de actuar de los docentes y estudiantes hacia la definición y realización de secuencias predeterminadas; mediante la transmisión de conocimientos desde quién sabe, hacia quién desea aprender.

Este enfoque determina una educación controlada por el docente, es decir, el decide para qué y que enseñar, al estudiante le queda el rol de asimilar al máximo lo que se le transmite, pero presentando limitaciones en su aprendizaje productivo (análisis, síntesis y evaluación).

El estudiante moderno no debe dejarse llevar por todo lo que los docentes dicen, ya que en ocasiones pueden equivocarse, además para poder realizarse como profesionales excelentes no es necesario que se memoricen grandes cantidades de libros sino, que ellos mismos se den cuenta de que es lo que están aprendiendo y que es lo que realmente les interesa conocer. Es por eso que la realización de un Software Educativo podría ser una alternativa para mejorar la condición del estudiante como futuro profesional.

Cuando el profesor da un sustento teórico que permita al estudiante analizar los problemas y explorar soluciones alternas, quien queda en entre dicho no suele ser el sistema de enseñanza – aprendizaje, ni el profesor, sino el estudiante que no sobrevivió a la experiencia. Quienes asumen la función profesoral están en la obligación de enriquecer continuamente los ambientes de aprendizaje que tienen a su cargo, de esta forma se está en la posibilidad de superar la práctica antigua y entrar a otra de tipo tecnológico para favorecer a alguien que aprenda.

2.2.1.2. Enfoque Educativo Heurístico

Predomina el aprendizaje basado en la experiencia y por descubrimiento, es necesario que el docente favorezca el desarrollo de las capacidades de autogestión del estudiante, es decir, llegar al conocimiento a partir de la experiencia creando sus propios modelos de conocimiento, logrando así, una educación controlada por el estudiante para lo cual el profesor utiliza estrategias que pueden llevar al estudiante a:

- Aprender a enfrentar fracasos y encarnar una conciencia que combine claramente lo que la persona es capaz de hacer.
- Construir sus propios modelos basados en la interpretación de lo que el profesor le propone.

- Recorrer por si mismo el camino para lograr metas bajo su propio control.

El docente debe hacer uso de ambientes educativos ricos, mantener motivados y activos a los estudiantes, promoviendo el desarrollo de las capacidades de autogestión de los estudiantes, enriqueciéndolo aún más con el uso adecuado de un computador. Se desea que el estudiante adquiera la capacidad resolver problemas con apoyo informático, siendo capaz de especificar, diseñar, desarrollar, probar, documentar y soportar la solución asistida por computador.

En conclusión, en la práctica interesa que el estudiante adquiera los conocimientos básicos, bajo el primer enfoque y resuelva los problemas complejos mediante el segundo.

Lo primordial para este fin, es la búsqueda de recursos imaginativos para el aprendizaje y análisis de circunstancias que faciliten la educación. “Se debe encontrar un modo de desarrollar un clima que no se centre simplemente en la enseñanza, sino en facilitar un ambiente de aprendizaje autodirigido”. Solo así se desarrollarán individuos creativos, abiertos a la experiencia, concientes de ella y que vivan un proceso de continuo crecimiento y cambio.

CLASIFICACIÓN DE LOS MATERIALES EDUCATIVOS COMPUTARIZADOS	
<i>Enfoque educativo</i>	<i>Tipo de material educativo según la función que asume</i>
Algorítmico	Sistema tutorial Sistema de ejercitación y práctica.
Heurístico	Simulador, Juego educativo, micromundo exploratorio Lenguaje sintónico, Sistema experto.
Algo – Heurístico	Sistema inteligente de enseñanza aprendizaje.

Cuadro 2.

CLASIFICACIÓN DE LOS MATERIALES EDUCATIVOS COMPUTARIZADOS

2.2.2. Modelos Pedagógicos y Teorías que Sustentan El Diseño de Ambientes de Enseñanza – Aprendizaje

Las teorías citadas pertenecen al campo de la psicología y la pedagogía y por ende su influencia más notable en el campo de la didáctica se localiza en el nivel metodológico, en el concepto de enfoque comunicativo.

“El único medio de impedir que la educación caiga bajo el yugo de la costumbre y degenera en automatismo maquinal e inmutable, es el tenerla siempre despierta mediante la reflexión”.⁵

La pedagogía tiene siempre un carácter prospectivo, “su objetivo no consiste en describir y explicar lo que es ó lo que ha sido, sino en determinar lo que debería ser. No están orientadas ni hacia el presente, ni hacia el pasado sino hacia el porvenir”.

Según Julio Pérez Erazo la pedagogía es la disciplina que conceptualiza, aplica y experimenta los conocimientos referentes a la enseñanza de los saberes científicos, y que por lo tanto la enseñanza dejó de ser una labor artesanal para convertirse en objeto de la pedagogía y de la investigación pedagógica para intervenir en los cambios del conocimiento, en las soluciones a las crisis de los paradigmas del mundo contemporáneo y en la verificación de los medios de comunicación y relación personal entre el profesor, alumno y los saberes, para que la educación mirada como proceso continuo de socialización, asimilación y creación del conocimiento, adquiera la calidad académica y científica necesarias.

En campo pedagógico clásico y moderno se han preocupado por responder estos cinco interrogantes fundamentales:

- 1) Qué tipo de hombre interesa formar?
- 2) Cómo y con qué estrategias Tecnometodológicas?
- 3) A través de qué contenidos, entrenamientos o experiencias?

⁵ Avaro H. Galvis Panqueva. Ingeniería del Software Educativo P. 19

- 4) A qué ritmo debe adelantarse el proceso de formación?
- 5) Quién predomina o dirige el proceso, el profesor o el alumno?

Rafael Flores Ochoa en su libro “Hacia una pedagogía del conocimiento”, afirma que las respuestas a los interrogantes anteriores pueden dar origen y de hecho deben hacerlo, a innumerables modelos pedagógicos.

Generalizando se puede decir que las teorías o modelos que más han marcado el enfoque educativo pertenecen a la corriente conocida como Escuela Nueva.

Desde principios de siglo algunos autores como Freinet, Dwyer, Gestalt, Cousinet criticaron el humanismo tradicional que reinaba entonces en el campo de la educación y trataron de implantar un nuevo estilo de transmisión del saber que podría ser obtenido gracias a un cambio de las técnicas aplicadas en el salón de clase. Para la Escuela Nueva la meta no es formar al estudiante para un objetivo exterior, sino capacitarlo para alcanzar la autonomía.

“Se trata de desarrollar la personalidad de cada uno y no solamente fomentar el almacenamiento pasivo de conocimientos”.⁶

⁶ www.uady.mx/~educacio/aprenda/construct.html

En esta concepción pedagógica, las relaciones dentro del salón de clase se han transformado radicalmente; Los educandos deben participar en la definición de la enseñanza, en la selección de los métodos y en el control de la formación.

El curso cede el paso a actividades realizadas de preferencia en grupo y con la participación creativa del alumno: juegos reales, simulaciones, conceptualización gramatical, resolución de problemas, etc,. El salón de clase se convierte en un espacio de intercambios de carácter social en los cuales los momentos de interacción de tipo simétrico son más numerosos que los de tipo complementario.

2.2.2.1. Pedagogía Tradicional

Entre los modelos pedagógicos citados por Julio Pérez Erazo se encuentra la pedagogía tradicional, la cual no ha perdido vigencia pese a que su origen y contemporaneidad data de la edad media, pues aún se lo aplica en la escuela colombiana, con frecuencia basta analizar el tipo de disciplina, de horario, de administración y desarrollo de currículo en el aula de clase, para demostrar que la afirmación no es falsa. Este modelo negado o discutido por muchos pedagogos, no ha dejado de ejercer una eficaz influencia.

2.2.2.2. Pedagogía Conductista

Los fundamentos de este modelo están dados en la psicología Bahaviorista, que considera a las operaciones psíquicas somáticas en las que intervienen los estímulos exógenos, el temperamento de cada uno y los hábitos preadquiridos, y la teoría del reflejo condicionado de Pavlov, que se sintetiza en la fórmula:



Figura 2.

ESQUEMA DE LA TEORÍA DEL REFLEJO DE PAVLOV

2.2.2.3. Pedagogía Romántica

Parte de la premisa que sostiene que el proceso educativo esta dado por la interioridad de la persona. Se fundamenta en el desarrollo integral del ser humano, logrando a la imagen de “Emilio” (obra de Rousseau) y en su concepción naturista.

2.2.2.4. Pedagogía Desarrollista

Es en cierta forma, una prolongación del modelo Romántico, por cuanto es partícipe de los planteamientos del Naturalismo Contemporáneo. Se fundamenta en el estudio psicológico del alumno en sus distintas fases su desarrollo: Infancia, adolescencia y juventud; a las que debe atenderse para que el proceso educativo sea normal y espontáneo. Este concepto de naturalismo es el que adoptan los representantes del modelo desarrollista como Piaget y Ferrier.

2.2.2.5. Pedagogía Constructivista (Jerome Bruner)

Estrategia pedagógica que parte de los saberes que posee la persona para la construcción del conocimiento, y complementa otras características que se dan durante el proceso de enseñanza. Luego hace un intercambio activo que lleva a la transformación de ambos para tomar conciencia de que realmente han sido transformador y transformado. El Constructivismo considera que :

- Es fundamental tener presente las acciones e intereses de los estudiantes, al igual que sus errores en la consolidación de los aprendizajes.
- Dentro del proceso del aprendizaje hay que aprovechar las situaciones problemáticas que dinamicen los esquemas que posee el alumno frente a

los diversos objetos de estudio permitiendo la formulación de hipótesis y confrontación de los mismos con las teorías eficientes.

- El conocimiento del aprendiz y su grupo como constructores del conocimiento a través de pasos o de etapas que sigue este proceso y los medios que se emplean en dicha construcción.

En el constructivismo el alumno no copia la realidad, ni la simula, ni la reproduce sino que la construye a través de la interacción con ella misma y con sus compañeros.

2.2.2.6. Cognoscitivismo y Pedagogía de Gestalt

Los individuos no responden tanto a estímulos, sino que actúan sobre la base de creencias, actitudes y un deseo de alcanzar ciertas metas. La comprensión que tenga una persona de su ambiente formado por su pasado, presente y futuro, además de una realidad concreta y otra imaginaria, la comprensión que tenga de su “campo“ será la estructura cognoscitiva del campo vital. Así, el aprendizaje puede entenderse como un cambio en las estructuras del campo vital del aprendiz.

El proceso de enseñanza – aprendizaje según Gestalt debe tomar en cuenta los siguientes elementos: Motivación Intrínseca (el aprendizaje es motivado cuando

esta relacionado con algo de interés o significancia para la persona), La Adquisición (discernimiento repentino), La Retención (lo recordado es algo que tiene significancia para la persona), Transferencia del Aprendizaje (solo si existe interés).

2.2.2.7. Cognoscitivismo y Psicología Evolutiva de Jean Piaget

Al igual que en la pedagogía definida por Gestalt, la filosofía de Piaget es fundamentalmente Kantiana. “La realidad se construye en el proceso de sintetizar sensaciones percibidas con las estructuras del conocimiento”, es decir que la realidad es simplemente una reconstrucción a través de procesos mentales que operan sobre los fenómenos del mundo de los sentidos.

Según Piaget, hay cuatro factores que intervienen en el proceso del aprendizaje: la maduración (fruto del desarrollo biológico), la experiencia (posibilidades concretadas mediante la ejercitación y la práctica), el equilibrio (reacción de contrarrestar la perturbación) y la transmisión social (transmisión lingüística, educación).

- El proceso de aprendizaje consiste en una asimilación sistemática y progresiva del “objeto” o de las “experiencias”, dicha incorporación implica

regularmente un proceso de ajuste de las estructuras asimilatorias de acuerdo con las características del objeto o de las experiencias en cuestión.

- El aprendizaje es siempre un proceso de actividad diferente de la recepción pasiva de conocimientos donde estos son construidos por el alumno que aprende.

2.2.2.8. Pedagogía Algo – Heurística

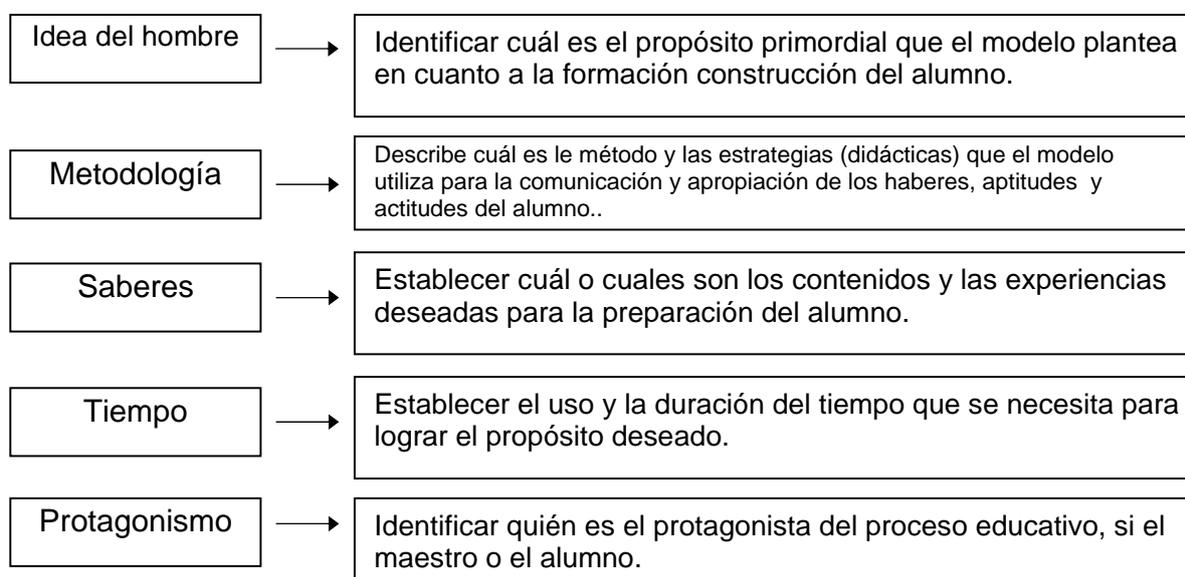
Landa L. N. Es una teoría general del aprendizaje orientada a la identificación de los procesos metales conscientes e inconscientes. Concibe a los procesos metales como secuencias estructuradas de unidades elementales de conocimiento y de operaciones relacionadas a una tarea. Se conciben problemas, procesos de tipo heurístico y semi – heurístico, así como algorítmicos o semi – algorítmicos.

Esta teoría se rige por los siguientes Principios:

- Es más importante enseñar procesos Algo – Heurísticos, que prescripciones (Conocimiento dado); el profesor debe conocer ambos.
- Es valioso enseñar a descubrir procesos, que darlos ya formulados.
- Descomponer los procesos en operaciones elementales del tamaño y talla apropiados a cada estudiante.

- Orienta su enseñanza para el desarrollo de métodos generales de pensamiento.

El marco teórico de cada pedagogía descrita anteriormente se construye a partir del modelo de análisis de la figura:



Cuadro 3.

MODELO ANALÍTICO

2.2.3. Recursos Educativos en el Proceso de Aprendizaje

Los recursos educativos dirigidos a la ayuda o soporte técnico en el proceso de aprendizaje permiten una nueva visión y aprovechamiento como estrategias que contribuyen a minimizar debilidades y amenazas en la construcción de conocimiento. El docente tendrá la obligación de gestionar la autorización de recursos educativos en el proceso de enseñanza aprendizaje de teoría de Grafos permitiendo una mejor interacción de sus estudiantes con el medio.

El proceso de aprendizaje se basa en experiencias previas, es decir, que los recursos educativos cumplen también con la función de reforzar las experiencias previas de los estudiantes, ó de traer a ellos las experiencias de otras personas que les sean útiles; su utilización se determina a partir de lo que se quiere enseñar y de los niveles de aprovechamiento que se esperan de los alumnos, para ello es necesario la implementación de recursos diferentes al libro y al discurso expositivo verbal, ya que no todos los conocimientos pueden estar contenidos en ellos.

Un recurso como el computador permitirá una mayor interacción de los estudiantes con el medio de trabajo, logrando una educación activa y eficiente basada en la experiencias de los alumnos que llevan el conocimiento de hechos prácticos y que eviten el tradicional verbalismo.

Es necesario también, que el educador este conciente y abierto a todos los beneficios que nos ofrece la era de la tecnología. Conceptos como realidad virtual, multimedia, telecomunicaciones, redes, informática, Internet y demás herramientas del mundo moderno nos permiten entrar en contacto con las nuevas tecnologías y por ello es responsabilidad y casi una obligación del docente involucrarse dentro de este ambiente y no estancarse en el convencionalismo.

“El descubrir y entender todas las ventajas que la era de la computación nos brinda nos hace día a día mucho mas competitivos y eficientes”.

2.2.4. El Computador como Estrategia de Aprendizaje Didáctico

Propiciar al educando ambientes de aprendizaje donde este encuentre computadores, textos, equipos audiovisuales y otro tipo de material didáctico no basta para que el alumno aprenda. La actitud del estudiante durante este proceso es la que permite aprender. Aunque se dispone de una serie de recursos como marcador, tablero, acetatos, diapositivas, películas, videos entre otras herramientas modernas para la enseñanza, sin lugar a dudas es el computador el que nos ofrece más ventajas. No hay herramienta tan útil, versátil, eficiente y moderna que permita una relación tan íntima e interactiva.

“Estamos entre una tecnología sin precedentes, sobre la cual se pueden construir sistemas educacionales que distinguen entre la transmisión de herencia cultural y la promoción del nuevo entendimiento.”

El aprendizaje apoyado por un Software Educativo bien empleado puede llegar a producir los siguientes resultados:

- 1) Reduce las equivocaciones en la medida en que el alumno se ejercita.
- 2) El tiempo de aprendizaje es relativo y de manera individual, ya que varía de acuerdo al ritmo de trabajo del alumno y es él quien decide cuando aprender.
- 3) La motivación del aprendizaje puede aumentar, debido al hecho de que los alumnos saben inmediatamente si han tenido éxito.
- 4) Un seguimiento personalizado, donde se pueda evaluar en forma individual y transparente a los alumnos para brindar refuerzos individuales y no grupales.

Alvaro Galvis nos dice que los estudiantes procesan más eficientemente la información si participan activamente en su tratamiento. Por otra parte la capacidad de cada individuo para aprender es única y puede variar dependiendo de el grado de motivación y participación que se le aplique, además de la ansiedad que le produzca el aprendizaje de las temáticas que se le propongan.

2.2.5. Informática Educativa

La transferencia racional de la tecnología depende en buena parte de la preparación de personal especializado, particularmente, la transferencia de tecnología en el área de la computación educativa tiene como base el aporte de especialistas en computación y educación; cabe señalar que la informática y la computación son campos tan exuberantes en generación y transmisión de conocimientos, que los avances de la tecnología llevan cada vez más a especializaciones dentro de esta rama del saber, se trabaja por ejemplo, en robótica, inteligencia artificial, telemática, ofimática, informática educativa, bases de datos, sistemas de información, computación grafica, etc,.

En cuanto a lo que nos concierne, se ha vuelto ineludible analizar las relaciones entre informática y educación, con el fin de aprovechar el potencial educativo que puede tener el uso de computadores en esta área, a sus diferentes niveles y modalidades.

Conviene que quienes ven elementos “mágicos” en la adquisición de computadores para el sistema educativo, pongan los pies sobre la tierra para asegurar así todas las condiciones que pudieran hacer efectivo ese potencial.

Esta ciencia ofrece un conjunto de orientaciones pedagógicas, en las cuales se utilizó el computador como un recurso educativo con ventajas para el alumno en su proceso cognitivo y para el educador en su proceso de seguimiento del estudiante, como creadora de espacios de aprendizaje proporcionando al proceso educativo atributos y características propias que permitan realizar actividades tanto en grupo como individualmente, permitiendo que el estudiante incremente su capacidad de pensar en forma lógica y formule procedimientos para la resolución de problemas.

La informática educativa ha sido enriquecida con nuevas herramientas y teorías cognitivas que promueven no solo eficiencia y efectividad de la enseñanza sino la calidad de la misma. Con esto no se pretende que el docente utilice el computador como base para la educación, sino que a través de éste, emplee diversidad de metodologías pedagógicas en un ambiente alegre, ameno y diferente a lo tradicional. Lo que se busca en sí, es contribuir substancialmente por medio del computador a la educación individual.

2.2.5.1. Software Educativo

(1) Programa que permite cumplir y/o apoyar funciones educativas, brindando al usuario la oportunidad de un aprendizaje fácil, rápido y divertido. (2) Programa

creado con la finalidad específica de ser utilizado como medio didáctico, es decir, para facilitar los procesos de aprendizaje.

Un software educativo puede ser realizado como apoyo para cualquier tipo de temática (de ingeniería, matemáticas, idiomas, geografía, dibujo, etc.), de formas muy diversas (a partir de cuestionarios, facilitando una información estructurada a los estudiantes, mediante la simulación de fenómenos, etc.) ofreciendo un entorno de trabajo flexible a las circunstancias y con oportunidades de interacción;

características esenciales:

- Son materiales elaborados con una finalidad didáctica.
- Utilizan el computador como soporte en el que los estudiantes realizan las actividades que ellos proponen.
- Son interactivos, permiten intercambio de información entre el computador y los estudiantes.
- Individualizan el trabajo de los estudiantes, ya que se adaptan al ritmo de trabajo de cada uno.
- Son fáciles de usar. Los conocimientos informáticos necesarios para utilizar la mayoría de estos programas son mínimos, aunque cada programa tiene sus normas de funcionamiento que es necesario conocer.

2.2.5.1.1. Clasificación General de Software Educativo

Los programas educativos a pesar de tener unos rasgos esenciales básicos y una estructura general común, se presentan con unas características muy diversas, por lo cual se han elaborado múltiples tipologías que clasifican un Software Educativo, veamos algunos de ellos:

2.2.5.1.1.1. Tutoriales

Son programas que dirigen el trabajo de los alumnos. Pretenden que, mediante la realización de ciertas actividades, los estudiantes aprendan o refuercen unos conocimientos y/o habilidades.

Contienen cuatro fases específicas: la fase Introdutoria, donde se genera la motivación, se centra la atención y se favorece la percepción selectiva de lo que se desea que el alumno aprenda; la fase de orientación inicial, donde se da la codificación, almacenaje y retención de lo aprendido; la fase de aplicación, en la que hay evocación y transferencia de lo aprendido y la fase de retroalimentación, en la que se demuestra lo aprendido y se ofrece retroinformación y refuerzo.

2.2.5.1.1.2. Simuladores y Juegos Educativos

apoyan el aprendizaje de tipo experiencial y conjetural, como base para lograr el aprendizaje por descubrimiento. Presentan un modelo o entorno dinámico (gráficos o animaciones interactivos) que facilitan su exploración y modificación, pueden realizar aprendizajes inductivos o deductivos mediante observación y manipulación. Lo esencial en ese tipo de software es que el alumno es un agente necesariamente activo que, además de participar en la situación debe continuamente procesar la información que el micromundo le proporciona en forma de situación problemática, condiciones de ejecución y resultados. En estos ambientes vivenciales de aprendizaje, una vez que el aprendiz hace suyo el reto propuesto por el profesor ó por el sistema, él es actor y fuente principal del aprendizaje a partir de su propia experiencia.

2.2.5.1.1.3. Constructores

Son programas que tienen un entorno programable. Ofrecen a los usuarios unos elementos simples con los cuales pueden construir elementos más complejos. De esta manera potencian el aprendizaje heurístico (inventar, descubrir, etc.) y, de acuerdo con las teorías cognoscitivas, facilitan a los alumnos la construcción de

sus propios aprendizajes y aumentan las capacidades y destrezas creativas e intelectuales.

2.2.5.1.1.4. Sistemas de Ejercitación y Práctica

Se parte de la base de que el aprendiz ya adquirió los conceptos y destrezas que va a practicar. Con ellos se trata de reforzar dos fases muy importantes durante el proceso de instrucción de los aprendices, la aplicación y retroinformación.

En un sistema de este tipo deben tenerse en cuenta tres condiciones: cantidad de ejercicios, variedad en los formatos con que se presentan y retroinformación que oriente con luz indirecta a las acciones del aprendiz. Importante en los sistemas de motivación y práctica dar prioridad a los módulos de motivación y refuerzo. Como de lo que se trata es que el aprendiz logre destreza en lo que se está practicando, lo que se logra con una amplia y variada ejercitación, por lo que es imprescindible crear un gancho dentro del programa que mueva al usuario a realizar una cantidad significativa de ejercicios que estén bien resueltos y sin ayuda.

2.2.5.1.1.5. Lenguajes Sintónicos y Micro Mundos Exploratorios

Por medio de lenguajes con los que no hay que aprender, donde el usuario ya está sintonizado con ellos y que se pueden usar naturalmente para interactuar con un micromundo en el que los comandos sean aplicables. En este tipo de aplicaciones el trabajo del docente es promover que el aprendiz resuelva los problemas descomponiéndolos en sus partes y a su vez, cada una de ellas en nuevas partes, hasta cuando se llegue a enunciados que tienen solución directa por medio del uso de una instrucción que entiende el computador.

2.2.5.1.1.6. Sistemas Expertos con Fines Educativos (Se)

Son sistemas de computación capaces de representar y razonar acerca de algún dominio, rico en conocimientos, con el ánimo de resolver problemas y dar consejo a quienes no son expertos en la materia.

Otra forma de llamar a los SE es *Sistemas Basados en conocimiento*. Esto se debe a que son sistemas que usan procedimientos y conocimientos de inferencia para resolver problemas que son suficientemente difíciles como para requerir

experiencia y conocimiento humano para su correcta solución. Un SE trabaja sobre la base de la motivación intrínseca y auto refuerzo.⁷

2.2.6. Ambientes de Aprendizaje Asistidos por Computador

2.2.6.1. El MEC

Una tecnología adquiere valor pedagógico cuando se la utiliza sobre la base del aprovechamiento de los recursos de comunicación; pero ello no es suficiente. El valor pedagógico le viene de su mediación para promover y acompañar el aprendizaje.

Un MEC es, ante todo, un ambiente informático que permite que la clase de aprendiz para el que se prepara viva el tipo de experiencias educativas que se consideran deseables para él frente a una necesidad educativa dada. Los MECs son aplicaciones que apoyan directamente el proceso de enseñanza – aprendizaje. Los MECs complementan lo que con otros medios o materiales de enseñanza – aprendizaje se ha intentado realizar, sin embargo no es su aspiración

⁷ Alvaro H. Galvis Panqueva. Ingeniería del Software Educativo – p. 20

reemplazar la labor de otros medios educativos que han demostrado su eficacia sino complementarla.

El docente por su parte debe darse a la tarea de conocer y analizar todo material educativo que plantee estrategias de enriquecimiento y mejoramiento del proceso de enseñanza – aprendizaje en aras de la investigación educativa; para lo cual es imprescindible que aborde los MECs como un ofrecimiento de las nuevas tecnologías, convirtiéndose en creador y administrador de ambientes de aprendizaje.

2.2.6.1.1. Herramientas y Métodos para Generar MECs

Un paquete de materiales de enseñanza – aprendizaje no necesariamente esta conformado por el MEC y sus manuales para cada tipo de usuario. Dependiendo del tipo de MEC se puede considerar la necesidad de utilizar opciones de apoyo (impresos o audiovisuales), que puedan o no estar disponibles. Las herramientas que hoy se pueden utilizar par la educación y el entretenimiento son mucho más ricas y sofisticadas que las tradicionales, y lo mejor de todo es que son muy fáciles de usar.

En cuanto al diseño, el Software Educativo debería cumplir con los siguientes requisitos teniendo en cuenta el tema de aprendizaje seleccionado:

1. Concebir un vehículo educativo formado por varios escenarios que se complementen.
2. Definir el rol que habría de tener el estudiante en cada uno de los escenarios y de forma global.
3. Diseñar gráficamente los escenarios incluyendo los aspectos de animación que aparecen en ellos.
4. Diseñar la interfaz hombre – máquina a través de la cual estudiante puede interactuar plenamente con los objetos de cada escenario.
5. Una vez diseñadas las interfaces y los escenarios se procede a su construcción mediante la generación y ensamble e las diferentes rutinas del sistema, para ello se debe tener en cuenta:

- *Modularidad*: es conveniente definir módulos que permitan separar y/o reutilizar apartes de programación y diseño dentro del programa. Los módulos pueden ser producto de inspeccionar las funciones por usuario y proceso por estructura lógica, es importante clasificar los procedimientos empleados en más de un modulo.
- *Manejo de memoria principal y secundaria*: ya que no todas las

instalaciones de computación del sector educativo cuentan con un hardware configurado para este tipo de aplicaciones que por lo general por su tamaño y manejo de graficas, sonidos etc, suelen consumir mucha memoria.

- *Legibilidad y documentación del código:* la programación además de ser estructurada, debe emplear estándares para variables y procedimientos.
- *Documentación para mantenimiento:* es imprescindible crear a mediada que se hace el desarrollo un manual para mantenimiento que permita realizar esta función cada vez que se requiera.

El manual debe contener como mínimo:

- Identificación de la aplicación.
- Compilador y librerías requeridas.
- Estructura global del programa y función de cada componente.
- Archivos fuente.
- Archivos de datos.
- Listado comentados del programa.
- Imágenes instructivas de la aplicación.
- Diccionario de datos.

2.2.6.2. Estructuras de Información

Las *Estructuras de Información* pueden entenderse como un conjunto de datos cuya organización se caracteriza por las funciones de acceso que se usan para almacenar y localizar a elementos individuales de datos. Las funciones de acceso pueden ser:

- Insertar un nuevo elemento a la colección.
- Retirar un elemento de la colección.
- Localizar un elemento dado.

El primer nivel, se enfoca la representación real de los datos en la computadora, que se construye de "abajo hacia arriba", empezando con los tipos de *datos básicos* como caracteres, enteros, reales, booleanos. La principal característica de éstos, es que ocupan una sola casilla de memoria y por tanto una variable hace referencia a un solo valor.

Los espacios en memoria pueden ser asignados de forma aleatoria o también podemos tener acceso a uno en particular, dependiendo de la aplicación que se este manejando o creando. Es necesario siempre liberar dichos espacios para evitar desbordamientos y desestabilizar el sistema.

El siguiente nivel, lo conforman *arreglos* y *registros*, que representan conjuntos de datos básicos. Estos se caracterizan por el hecho de hacer referencia a un grupo de casillas de memoria, con el nombre de una sola variable. El nivel final, se concentra en la forma en que los datos serán almacenados y accedidos. En este nivel, es posible ver los datos dentro de la estructura de información, la cual controla la manera en que el programa accede a los datos.

Existen diferentes tipos de estructuras que permiten cada una acceder o almacenar los datos:

- ✓ Pilas.
- ✓ Colas.
- ✓ Listas.
- ✓ Árboles.
- ✓ Grafos.

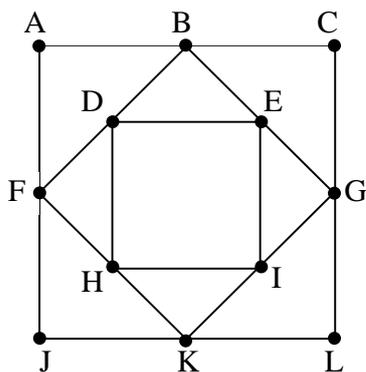
Cada método proporciona una solución a un determinado tipo de problema; cada uno es esencialmente un dispositivo que realiza una operación específica de almacenamiento y recuperación de la información dada, de acuerdo a la petición que reciba durante la ejecución de procesamiento de información.

2.2.6.3. Grafos

2.2.6.3.1. Definición de Grafos

Un Grafo es una estructura de datos compuesta por vértices y arcos, utilizados comúnmente en el manejo de redes, en la construcción de circuitos eléctricos, en estrategias de ventas en el área de la economía, cartografía y otra muchas áreas del diseño, la ingeniería y el conocimiento práctico.

Gráficamente un grafo se puede ver así:



Donde el conjunto de los **vértices** es:

$$V=\{A,B,C,D,E,F,G,H,I,J,K,L\}.$$

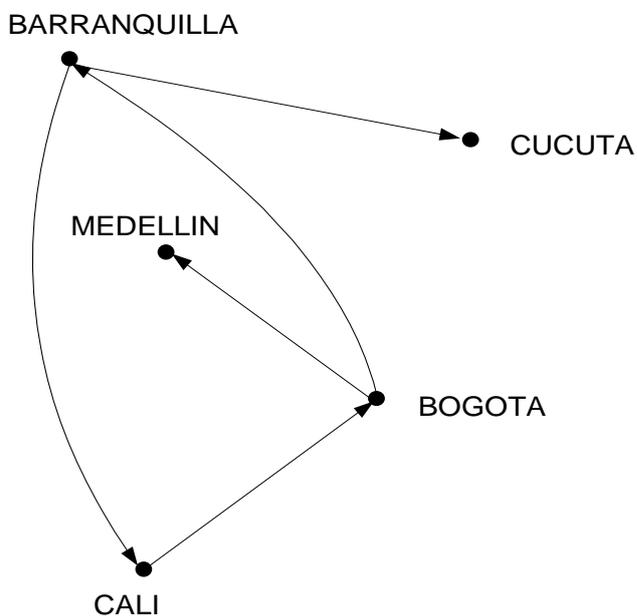
Un **arco** une dos vértices adyacentes. Por ejemplo el arco JK o el arco EG o GE. No se puede hablar del arco AC. Se puede hablar del arco AB o BC.

Figura 3.

REPRESENTACIÓN GRAFICA DE UN GRAFO

2.2.6.3.2. Grafo Dirigido o Dígrafo

Simplemente un grafo dirigido o Dígrafo es aquel en que sus arcos tienen una orientación. Por ejemplo:



El arco BOGOTA – MEDELLÍN es válido, en este dígrafo no existe el arco MEDELLÍN – BOGOTA.

El conjunto de **vértices** de este Dígrafo o grafo Dirigido sería:

$V \{BOGOTA, MEDELLÍN, CALI, CUCUTA, BARRANQUILLA\}$.

Figura 4.

REPRESENTACIÓN GRÁFICA DE UN DIGRAFO

El nombre de grafo proviene precisamente de la facilidad con que puede representarse gráficamente, y es su representación la que nos ayuda a comprender todas sus propiedades y aplicaciones. La mayoría de enfoques de

diseño y de flujo de datos pueden representarse por medio de este tipo de estructuras de información.

2.2.6.3.3. Áreas de Aplicación

La teoría de grafos se aplica en campos tan diversos como las ciencias sociales, ramas de la administración y la economía, ciencias físicas, ingenierías, cartografía, diseño estructural en áreas como la arquitectura y la electrónica puesto que permiten visualizar explícitamente las conexiones espaciales y optimizar su funcionamiento.

También desempeña un papel muy importante en varios campos dentro de las áreas en la comunicación, tales como, la teoría de la conmutación y el diseño lógico, en el diseño de redes son de vital importancia ya que por medio de esta metodología se puede detectar a un nivel mas específico las zonas (nodos) conflictivos y buscar las soluciones óptimas; en la robótica y la informática en áreas como la inteligencia artificial, gráficos por computadoras, sistemas operativos, estructuras de computadores, organización y recuperación de información.

Como consecuencia de su amplia gama de aplicaciones, es útil desarrollar y estudiar este tema en términos abstractos, interpretar sus resultados en términos de los objetos de aquel sistema concreto en el que estemos interesados para así obtener los resultados deseados.

2.2.6.3.4. Conceptos Básicos

Adyacencia: se dice que existe adyacencia entre dos vértices si están unidos por una arco. Para el caso de los Dígrafos o grafos dirigidos existen dos tipos de adyacencia: Adyacencia desde y Adyacencia hasta.

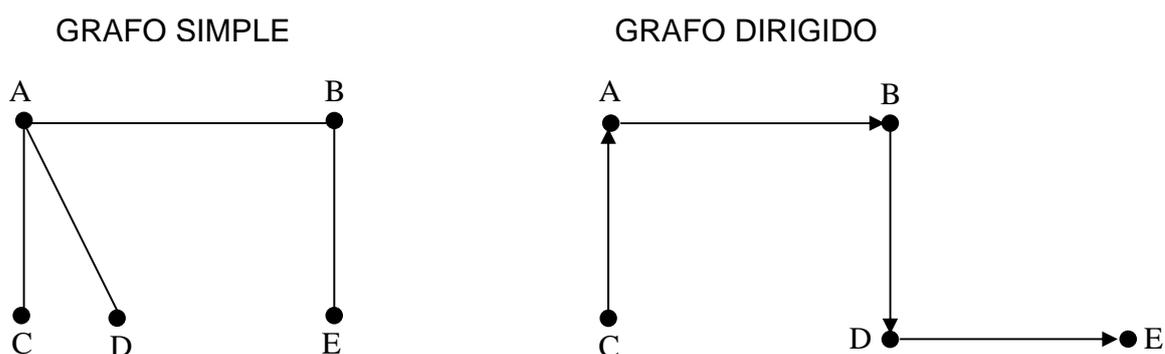


Figura 5.

REPRESENTACIÓN ADYACENCIA

En el grafo simple los vértices A y B son adyacentes. A es adyacente a B y a su vez B es adyacente a A. En el caso de los Dígrafos la adyacencia se expresa diciendo que A es adyacente hacia B o que D es adyacente hacia E. También que A es adyacente desde C y a su vez D es adyacente desde B.

Incidencia: los arcos inciden en los vértices. Un arco incide en un vértice si una de sus puntas llega a ese vértice. La incidencia también depende de si el grafo es Simple o Dirigido. Por ejemplo:

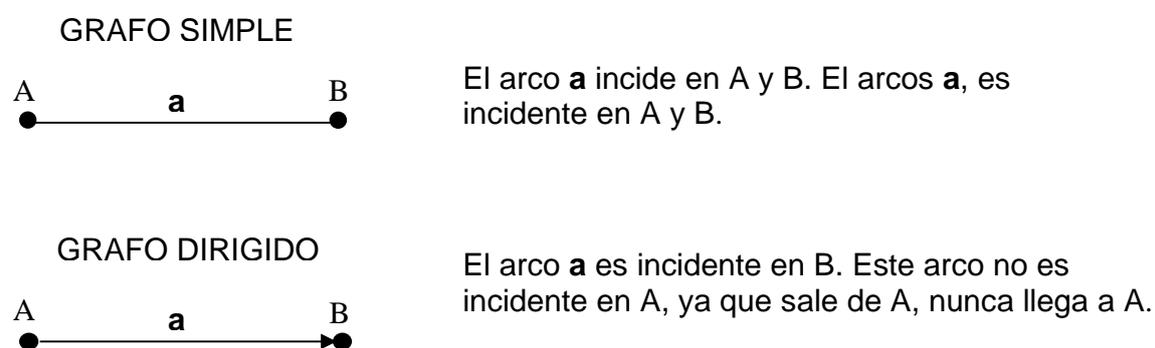
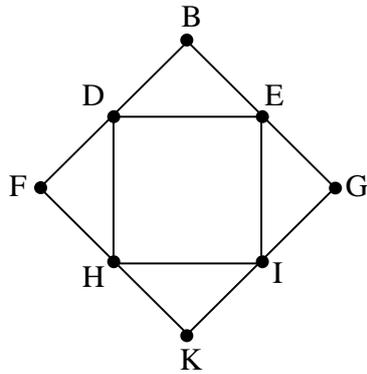


Figura 6.

REPRESENTACIÓN INCIDENCIA

Grado de un Vértice: el grado de un vértice es el número de arcos que inciden en ese vértice.

GRAFO SIMPLE



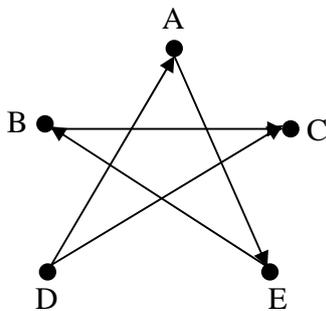
El grado del vértice F es 2.

El grado del vértice H es 4.

Figura 7.

GRADO DE UN VÉRTICE

GRAFO DIRIGIDO



Para este caso existen dos tipos de grado:

- *Grado de Entrada:* número de arcos que inciden en un vértice.
- *Grado de Salida:* número de arcos que parten del vértice.

A tiene 1 grado de entrada y 1 de salida.

C tiene 2 grados de entrada y 0 de salida.

Figura 8.

GRADO DE UN VÉRTICE (Digrafos)

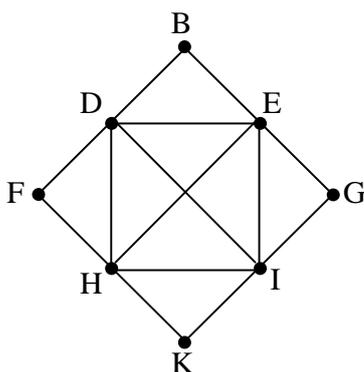
2.2.6.3.5. Tipos de Grafos

Existen diferentes tipos de grafos como son: Dirigido, Hamiltoniano, Euleriano, Regulares, Simples, Completos, entre los mas importantes, y diversos algoritmos de aplicación como: Warshall, Prim, Kruskal, Dijkstra, que nos permiten resolver problemas de toma de decisiones.

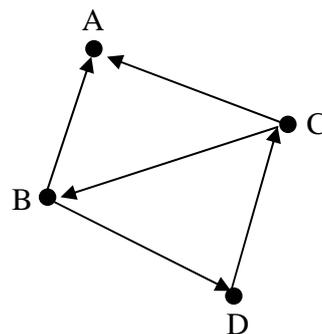
2.2.6.3.5.1. Grafos Fuerte y Débilmente Conectados

Fuertemente conectados: Un Grafo, ya sea Simple ó Dirigido, está Fuertemente conectado si desde cualquier vértice se puede llegar a todos los demás.

FUERTEMENTE CONECTADO



DEBILMENTE CONECTADO



Desde el vértice A no puedo llegar a todos los demás.

Figura 9.

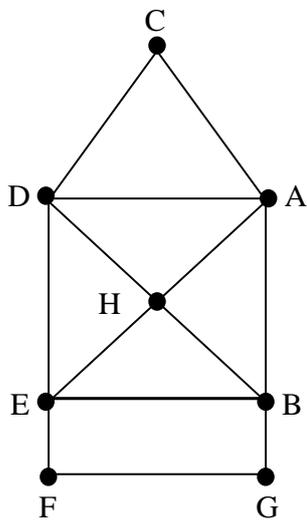
GRAFOS FUERTE Y DÉBILMENTE CONECTADOS

Débilmente conectados: Un Grafo Dirigido o Dígrafo es Débilmente Conectado, si por lo menos desde un vértice no se puede llegar a todos los demás. En un Grafo Simple ésta propiedad no se cumple ya que sus arcos son bidireccionales.

2.2.6.3.5.2. Grafos Eulerianos y Hamiltonianos

Grafo Euleriano: Un Grafo es Euleriano si partiendo desde cualquier vértice podemos recorrer todos los arcos llegando de nuevo al vértice origen. Los vértices pueden ser visitados cuantas veces sean necesarias, pero los arcos deben ser recorridos solamente una vez, es decir, si se tratara de un dibujo la línea no podría ser repintada. Para que un grafo sea Euleriano, todos los vértices deben tener un número par de arcos.

Grafo Hamiltoniano: Un grafo es Hamiltoniano si partiendo desde cualquiera de sus vértices, podemos recorrer todos los demás sin repetir ninguno, y finalmente llegar al vértice inicial. Al contrario de un Grafo Euleriano donde los vértices pueden ser visitados a conveniencia, para el Grafo Hamiltoniano los vértices deben ser visitados sólo una vez, y aunque no es necesario visitar todos los arcos, el grado de dificultad aumenta considerablemente.



Un recorrido partiendo desde el vértice A que demostraría que el grafo de la figura de la izquierda es *Euleriano* puede ser:

A C D E F G B E H D A H B A.

También puede ser *Hamiltoniano* y lo demuestra el siguiente recorrido iniciado desde el vértice A:

A C D E F G B H A.

Figura 10.

GRAFOS EULERIANOS Y HAMILTONIANOS

2.2.6.3.5.3. Otros Tipos de Grafos

Grafo Regular : Cuando hablamos de Grafos Regulares nos referimos a aquellos donde sus vértices tienen el mismo Grado de Incidencia.

Grafo Cíclico : Un Grafo Cíclico se define por una secuencia de vértices adyacentes (mínimo 3), donde el vértice inicial es el mismo vértice final; este tipo

de Grafos se recorren en un solo sentido, visitando todos los arcos y todos los vértices que lo componen. Un ciclo no es igual a un bucle.

Multígrafo : Se llama Multígrafo, a un Grafo donde como mínimo 2 de sus vértices están unidos por más de un arco.

Grafos Completos : Decimos que un Grafo es Completo, si cada uno de sus vértices tiene un grado igual a $n - 1$, donde n es el número de vértices que componen el grafo.

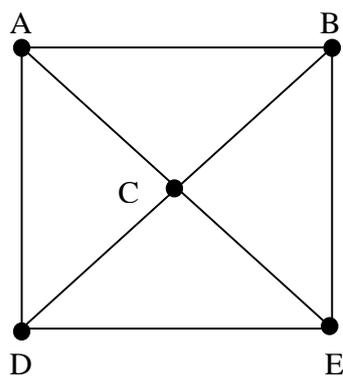
Grafos Conexos y Desconexos: La conectividad de los grafos depende de que existan o no caminos para unir los vértices que lo componen.

a) Grafos Conexos: en los cuales existe por lo menos un camino que una todos los vértices.

b) Grafos Desconexos: en los cuales existe por lo menos un par de vértices a los que no se tenga acceso.

2.2.6.3.6. Almacenamiento de un Grafo en la Memoria de un Computador

Un grafo puede ser almacenado de dos formas distintas: en un matriz de adyacencia, donde cada elemento $M[i][j]$, contiene un número cero (0) si el vértice i no es adjunto al vértice j , ó en una lista de adyacencia, donde se trabaja con arreglo de listas y, cada una de éstas almacena los adjuntos a un vértice dado. El manejo de la memoria del computador está determinado por el tipo de lenguaje de desarrollo de software que se desee emplear. Por ejemplo, en un lenguaje orientado a carácter, se debe realizar una rutina para la liberación de memoria. Por otro lado, en los lenguajes orientados a eventos o a objetos, dicha rutina no es necesaria, ya que el lenguaje mismo se encarga de liberar la memoria utilizada por medio de rutinas internas.



LISTA DE ADYACENCIA

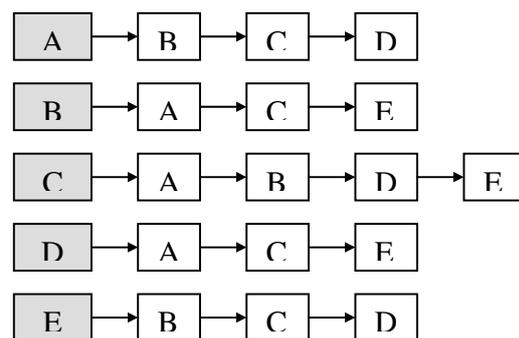


Figura 11.

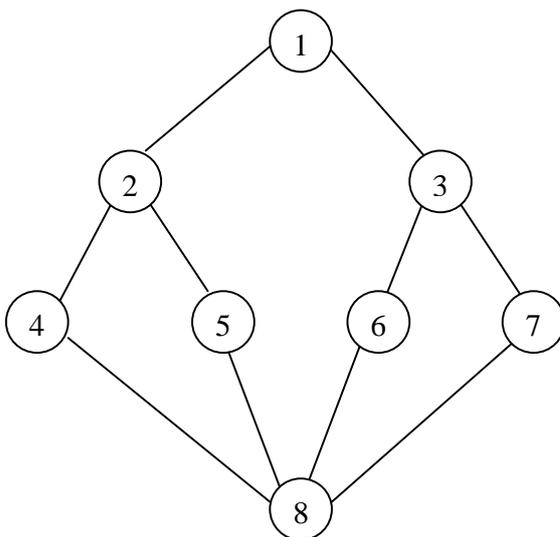
ALMACENAMIENTO DE UN GRAFO EN LA MEMORIA DE UN COMPUTADOR

2.2.6.3.7. Recorridos de un Grafo

Existen fundamentalmente 2 métodos clásicos de recorrer un grafo. Estos son a lo Ancho y en Profundidad. Un grafo a lo Ancho se recorre por niveles. En profundidad su recorrido se asemeja al recorrido inorden de un árbol.

2.2.6.3.7.1. Recorrido de un Grafo a lo Ancho

El Recorrido de un Grafo a lo ancho se realiza por niveles a partir de un vértice seleccionado. El resultado final, es una lista que contiene todos los vértices del grafo, el nombre del padre y el orden de visita para cada vértice.



Un recorrido a lo ancho del grafo de la figura de la izquierda, partiendo desde el vértice No 1 es:

1 3 2 8 5 7 6 4

se debe tener en cuenta que el padre del vértice con el cual se empieza el recorrido siempre es cero.

Figura 12.

RECORRIDO DE UN GRAFO A LO ANCHO

2.2.6.3.7.2. Recorrido de un Grafo en Profundidad

Al igual que en el recorrido a lo ancho, el recorrido en profundidad se inicia a partir de un vértice escogido, y su objetivo final es obtener el padre y el orden de visita de cada vértice, que dependerán del vértice inicial seleccionado y del orden de ingreso de los vértices adyacentes. El método para recorrer un grafo en profundidad, consiste en visitar el vértice original (supongamos que fuera A), luego el primer adyacente al vértice A (supongamos que sea B), luego el primer adjunto al vértice B (supongamos que sea C)... y así sucesivamente. Siguiendo este método el recorrido en profundidad iniciando desde el vértice No 1 del grafo de la figura anterior sería: 7 3 2 1 4 5 6 8 9.

2.2.6.3.8. Aplicaciones de Grafos (Algoritmos)

2.2.6.3.8.1. Algoritmo de Prim

El Algoritmo de Prim resuelve el problema de escoger la mejor alternativa entre las muchas que se puedan presentar y como resultado se obtendrá un conjunto de arcos de manera que conecten todos los vértices y además que la suma de sus valores (peso) sea mínima, es decir, obtener el mínimo árbol expandido. Para lograr el objetivo del algoritmo de Prim seguimos el siguiente proceso:

1. Determinamos en que vértice iniciamos el proceso y extendemos sus vértices hacia sus adyacentes no visitados.
2. El siguiente paso es ver en cuál de los vértices no visitados inciden 2 arcos. Si existe dicho vértice y los pesos de los arcos incidentes difieren, se procede a Eliminar el arco de mayor peso, si no, se deberá eliminar un arco al azar.
3. Seleccionamos entre los vértices no visitados, el vértice cuyo arco sea el de menor peso y extendemos todos los arcos. El proceso se repite N-1 veces, siendo N el número de vértices que componen el grafo. Por último sumamos los pesos del árbol expandido resultante.

Veamos un ejemplo:

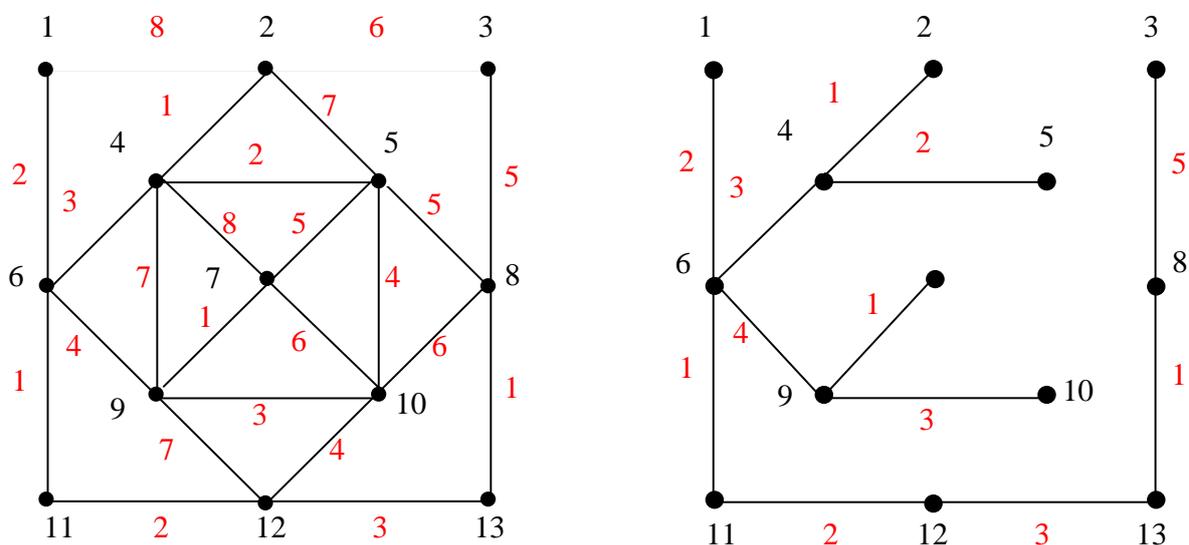


Figura 13.

ALGORITMO DE PRIM

En el ejemplo anterior la mínima sumatoria de los pesos es 28 y el orden del recorrido para llegar a este resultado es el siguiente: 12, 11, 6, 1, 4, 2, 5, 13, 8, 9, 10, 3.

2.2.6.3.8.2. Algoritmo De Dijkstra

El procedimiento de Dijkstra es parecido al Algoritmo de Prim. La diferencia consiste en que se escoge el vértice donde la suma de todos los arcos para llegar del vértice origen hasta dicho vértice, sea la mínima. El objetivo del Algoritmo de Dijkstra, es construir el árbol de caminos y calcular la distancia mínima entre un vértice seleccionado y los vértices restantes de un grafo ponderado.

Para lograr el objetivo del algoritmo de Dijkstra seguimos el siguiente proceso:

1. Determinamos en que vértice iniciamos el proceso y extendemos sus vértices hacia sus adyacentes no visitados.
2. Ahora, debemos escoger entre los vértices no visitados aquel donde la suma de los pesos de los arcos sea menor, desde el vértice origen hasta dicho vértice y sus arcos deberán ser extendidos hacia sus vértices adyacentes no visitados.

3. Si existen 2 rutas para llegar desde el vértice inicial hasta un mismo vértice , se selecciona la ruta cuya longitud sea mayor y se elimina, de lo contrario, es decir, si son iguales, se eliminará el último arco de la ruta más nueva.
4. Seleccionamos nuevamente el vértice no visitado donde la suma de los pesos de sus arcos sea la menor. El proceso se repite hasta que todos los vértices hayan sido visitados.
5. Por último, el resultado es el mínimo árbol expandido con las distancias mínimas desde un vértice origen hacia todos los demás.

Veamos un ejemplo:

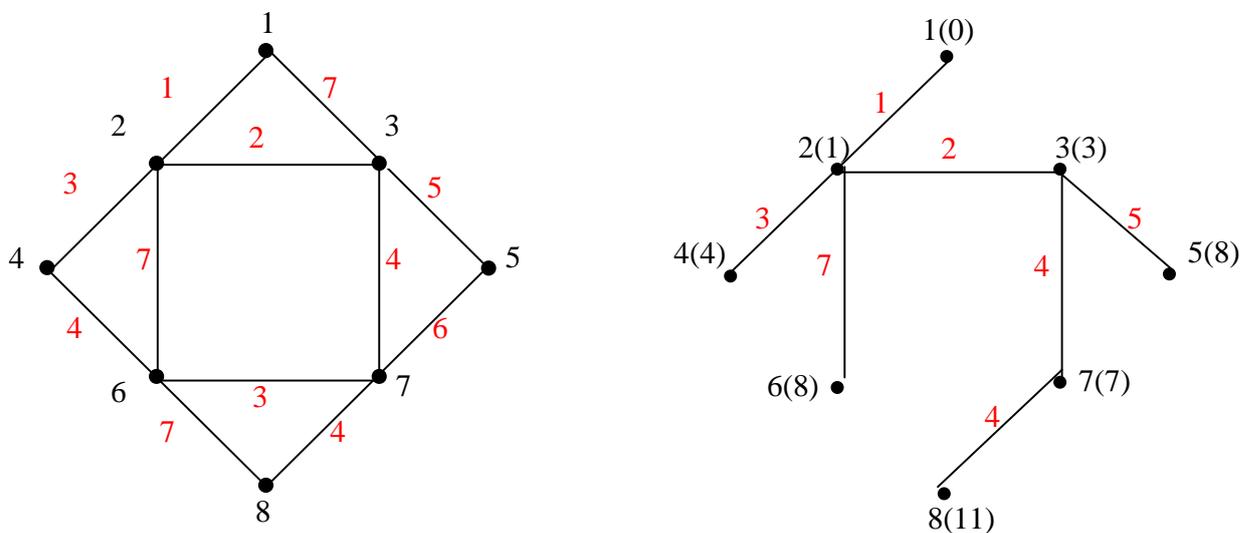


Figura 14.

ALGORITMO DE DIJKSTRA

2.2.6.3.8.3. Algoritmo De Warshall

El Algoritmo de Warshall resuelve el problema de saber si un dígrafo está fuertemente conectado o no. El algoritmo como tal está diseñado para hallar la solución basándose en una Matriz de Adyacencia y no tiene efecto alguno en Listas de Adyacencia. A fin de mantener el dinamismo de las Listas de Adyacencia, hemos desarrollado un algoritmo similar al de Warshall y que obtenga la misma solución, pero basándose en Listas de Adyacencia.

El algoritmo del que hablamos es recursivo ya que emplea de forma consecutiva recorridos del grafo en Profundidad, su uso se optimiza empleando grafos Dirigidos o Dígrafos. Cada vértice debe tener tres variables adicionales, inicializadas en cero. Una variable "*estado*" que determinará si el vértice ya ha sido visitado o no. La variable "*orden recorrido*" que definirá el orden de los vértices iniciales para los siguientes recorridos y la variable "*vértices visitados*" que almacenará el número de vértices inspeccionados durante el recorrido. Si el valor almacenado en esta variable, para cada vértice, es igual a N (N: número de vértices del grafo), decimos que el grafo está Fuertemente Conectado.

El proceso se resume en tantos recorridos en profundidad como vértices tenga el grafo dentro de un recorrido a lo ancho iniciando desde un vértice seleccionado.

2.2.6.3.8.4. Algoritmo de Kruskal

El Algoritmo de Kruskal se utiliza para buscar el mínimo árbol expandido de un grafo. Este Algoritmo cumple la misma función que el Algoritmo de Prim. La diferencia en cuanto a su aplicación, radica en que este último es más eficiente cuando el grafo a trabajar es denso, mientras que el método desarrollado por Kruskal, es más rápido cuando el grafo a trabajar es esparcido.

Para desarrollar el Algoritmo de Kruskal, es necesario tener conocimiento en temas como son: colas de prioridad, inserción y eliminación de nodos en un montículo, árboles orientados y listas de adyacencia. Razón por la cual este es considerado un poco más complejo.

1. Orden de Lectura del Grafo: El algoritmo de Kruskal inicia con la lectura de los datos del grafo, la cual no tiene un orden definido y por ende éste no influye en el resultado final del algoritmo. Supongamos que la lectura se hace en orden ascendente (o alfabético, lo cual es recomendable) donde primero se leyeron los datos del vértice 1, luego los del vértice 2, después los del 3 y así sucesivamente hasta llegar al vértice N. Cada una de los nodos leídos nos muestra la siguiente información (Peso, Vértice1, Vértice2).

Entendido el orden de lectura del grafo, el siguiente paso es la creación de *LA COLA DE PRIORIDAD* por medio de la Inserción y Eliminación de Nodos en un Montículo.

2. Creación del Montículo: Un montículo es un árbol binario donde cada nivel posee 2^N nodos con excepción del último, que puede no estar completo. Los niveles se llenan de izquierda a derecha siguiendo una filosofía de tipo ascendente. Para el caso del Algoritmo de Kruskal, cada nodo del montículo contiene los datos de una terna y el parámetro que rige la filosofía es el peso, es decir, en el montículo, el peso del nodo padre siempre será menor o igual que el de sus hijos.

Luego de obtenida la Cola de Prioridad, el siguiente paso es la creación del *ARBOL ORIENTADO* el cual es la clave para establecer la Lista de Adyacencia que define la estructura para la ejecución del último paso, la es la creación del *MINIMO ARBOL EXPANDIDO* (objetivo final de este algoritmo).

Mínimo árbol Expandido : En esta fase del algoritmo de Kruskal, debemos ir sacando una a una las ternas de la cola de prioridad para generar un árbol orientado, el cual nos dirá si los datos de la terna serán o no ingresados en la lista de adyacencia que a su vez define la estructura del mínimo árbol expandido. La

fase comienza con N árboles orientados donde N es el número de vértices del grafo. El proceso a seguir se resume en 3 pasos:

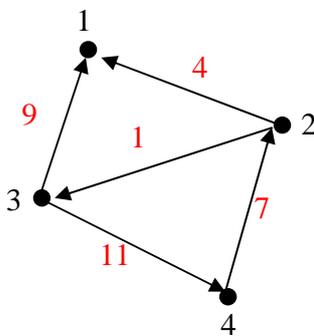
1. Sacar la primera terna de la cola de prioridad.
2. Unir, si es posible, los árboles en que se encuentran los vértices de la terna.
Si los árboles fueron procesados, los datos de la terna deben ingresar en la lista de adyacencia.
3. La lista de adyacencia nos da como resultado la estructura del Mínimo árbol Expandido con la sumatoria mínima total de los pesos resultantes del grafo propuesto.

2.2.6.3.8.5. Algoritmo de Floyd

Al igual que el Algoritmo de Warshall, el Algoritmo de Floyd fue creado para trabajar en base a una matriz de adyacencia. Por esta razón implementamos un algoritmo basado en el Recorrido en Profundidad y cuyo resultado es igual al obtenido con el algoritmo original, pero desarrollado utilizando listas de adyacencia, su uso se optimiza empleando grafos Dirigidos o Dígrafos. El objetivo primordial es calcular las distancias mínimas desde un vértice inicial hacia todos los demás vértices que componen el grafo; la operación se realiza $N-1$ veces donde N es el número de vértices que componen el grafo.

El método empleado se resume en hacer tantos recorridos en profundidad como vértices componen el grafo, internamente en cada uno de los recorridos se van almacenando en la estructura las distancias mínimas correspondientes dependiendo del vértice desde el cual se haya iniciado el algoritmo.

Veamos:



Listas de adyacencia (vértice: D. mínima)

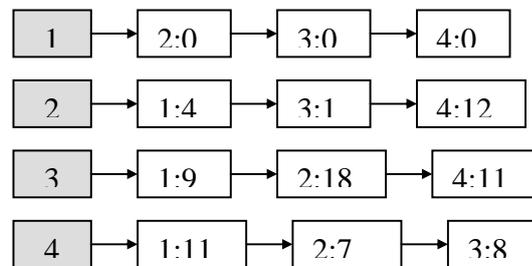


Figura 15.

ALGORITMO DE FLOYD

El resultado final, es una lista de adyacencia por cada vértice donde se almacenen todas las distancias mínimas desde cada uno de los vértices hacia todos los demás.

3. METODOLOGÍA

Existen metodologías que apoyan la tarea del educador facilitando los procesos de enseñanza – aprendizaje, encontrando un medio de soporte para desempeñar su labor y procurar vencer las limitaciones que tienen en su actual forma de trabajar.

Los MECs cuentan con un ambiente informático que permite que la clase de aprendiz para el que se preparo el MEC viva el tipo de experiencias educativas que se consideren deseables para hacer frente a una necesidad educativa dada.

Las nuevas estrategias educacionales se apoyan en sistemas de computación que destacan tecnologías sin precedentes. Estos sistemas educativos permiten transmitir una herencia cultural y la promoción de un nuevo entendimiento.

SEGraD mantendrá una metodología convencional de un proceso sistemático y progresivo del desarrollo de un proyecto, el paradigma del ciclo en el que se desarrolla el MEC contempla las siguientes actividades: Análisis, Diseño, Desarrollo y las correspondientes Pruebas.

3.1. CICLO DE VIDA DEL DESARROLLO DE UN PROYECTO

Galvis propone en su libro “Ingeniería de software Educativo” un doble ciclo para la selección o desarrollo de MECs como una posible metodología básica de trabajo.

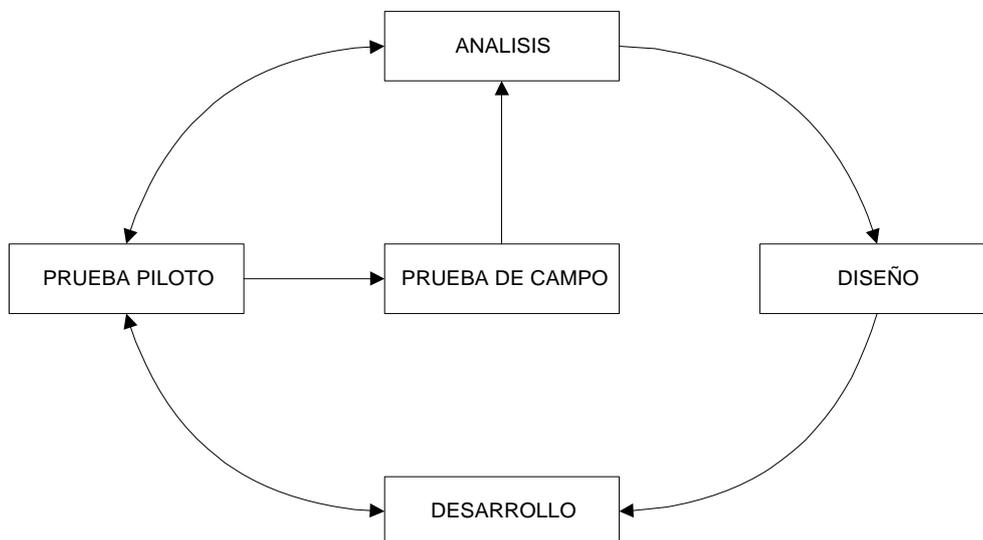


Figura 16.

CICLO DE VIDA DEL DESARROLLO DE UN PROYECTO

Según Galvis, el punto de partida es la identificación de necesidades educativas reales que conviene atender con el Material Educativo Computarizado (MEC), es decir siempre se debe comenzar por la etapa de análisis y agrega que “... dependiendo del resultado final de esta etapa se procede en el sentido contrario al

avance de las manecillas del reloj, cuando se trata de seleccionar un MEC; pero en el mismo sentido del avance de las manecillas si conviene su desarrollo”. En cada una de estas etapas se debe seguir ciertas pautas que permitirán que el desarrollo del MEC tenga el éxito esperado, alcance los objetivos y satisfaga las necesidades propuestas.⁸

3.1.1. Análisis

En esta etapa, se establecen las necesidades educativas que va a satisfacer el MEC, entendiéndose como necesidad educativa, según Galvis “... establecimiento de lo que hay que aprender con apoyo de un ambiente y actividades educativas”, estas necesidades se determinan a partir de los diferentes datos que proporcionan los usuarios (aprendices), los profesionales o docentes que tiene a cargo el grupo y el área temática, o se pueden extraer de investigaciones afines a la que se esta desarrollando.

Definidas las necesidades educativas, se deben tener claras las metas que se desea lograr en el proceso de apoyo durante el aprendizaje, para determinar los problemas existentes en los distintos temas y objetivos del plan de estudios; estos problemas se pueden establecer por medio de estrategias, encuestas observaciones, entre otros.

⁸ Galvis Panqueva. Ingeniería del Software Educativo - P. 70

También es necesario considerar las posibles causas de los problemas detectados como pueden ser: los alumnos, los materiales que se están utilizando para guiar la enseñanza, el profesor, el tiempo dedicado al tema de estudio, la cantidad o tipos de ejercicios, la metodología empleada, etc. Estas causas se pueden determinar a partir de encuestas y entrevistas, practicadas a los profesores y estudiantes y un análisis interno de los materiales de instrucción, Con lo que se determinan los objetivos que se pueden lograr con el material y además, todo aquello que no permita que se alcance los objetivos propuestos.

Dentro del análisis se deben plantear alternativas de solución para cada uno de los problemas detectados, entre los que se encuentran los apoyos informáticos, soluciones administrativas o académicas. Si opta por una solución computarizada, se debe especificar el tipo de MEC que conviene usar: sistema tutorial, simulador, de ejercitación y práctica etc; dependiendo de cómo va a ser la enseñanza y al tipo de usuarios que aprenderán con el MEC.

En el caso de que haya varios materiales que puedan satisfacer las necesidades detectadas, se las debe someter a ciclos de revisión y pruebas de MECs en lo posible teniendo su correspondiente documentación para determinar que por lo menos uno, satisface dichas necesidades.

Cuando no existe un MEC que cubra las necesidades, entonces el análisis termina con la formulación de un plan que se pueda llevar a cabo. Además se debe planear o prever lo referente a personal y tiempo que se va a dedicar en cada una de las etapas siguientes, así como también los recursos computacionales que serán necesarios.

3.1.2. Diseño

Galvis afirma que "...el diseño de un MEC está en función directa de los resultados de la etapa de análisis", esta etapa de diseño se subdivide en tres partes para llevar a cabalidad el desarrollo del MEC y cada una de estas partes debe estar documentada en forma completa y clara. En primer lugar está el Diseño Educativo en el que se determinará lo que se va a aprender, el ambiente en el que se va a aprender, como se mantendrá motivados a los aprendices y como se sabrá que estos están aprendiendo. En segundo lugar se desarrollará el Diseño de Comunicación en el que se establecen los tipos y características de los dispositivos de entrada y salida para la intercomunicación usuario – MEC, zonas de comunicación, características de estas zonas y verificación de que la interfaz satisface los requisitos mínimos deseables. Por último, se elabora cada Diseño Computacional en el que se determinan las funciones del MEC para cada uno de los usuarios, la estructura de los módulos, estructuras lógicas de los componentes

de la estructura principal y las estructuras de datos que se necesitan en memoria principal y secundaria para que funcione el MEC.

En detalle estos diseños deben contener:

- *Diseño Educativo*: este debe resolver interrogantes referentes al alcance, al contenido, a lo que va a resolver el MEC, así como también las características, físicas y lógicas del equipo en el que desarrollaría el MEC.

El objetivo terminal que es lo que se desea alcanzar con el MEC teniendo en cuenta lo que debe aprender el estudiante, se determina a partir de las necesidades detectadas en la etapa anterior (análisis).

El contenido esta basado en aquellos temas en que los aprendices necesitan mayores refuerzos, teniendo en cuenta que cumplan con lo que se tiene previsto desde la parte de los requisitos hasta el objetivo terminal. Dependiendo del tipo de MEC y de las características de los usuarios hacia quienes va dirigido éste, se especifica el micromundo o ambiente en el que se va a desarrollar el material educativo; éste debe ser agradable a los usuarios con el fin de que se sientan a gusto con lo que están trabajando y así el material logre cumplir sus expectativas acerca de lo que deben ó desean aprender.

Además del ambiente, en este diseño se debe trabajar muy bien la parte de motivación y refuerzo para que el aprendiz cumpla con los objetivos propuestos; esta motivación debe darse de acuerdo a los usuarios y al micromundo que se este manejando.

Por último, para que el aprendiz sepa cuanto ha aprendido y en que temas ó partes de contenido esta fallando, el MEC debe incluir evaluaciones relacionadas con cada objetivo especifico necesarios para alcanzar el objetivo terminal propuesto.

- *Diseño de Comunicación:* para manejar la interacción entre el usuario y el programa se necesita de comunicación, denominada interfaz. Se debe especificar tanto la interfaz de entrada como la de salida; es decir, determinar como se comunicará el usuario con el programa y viceversa. En esta parte se establecen los tipos de dispositivos de entrada y salida necesarios para que el usuario se intercomunique con el MEC, así como también las zonas de comunicación entre usuario y programa las cuales van asociadas a los dispositivos seleccionados y son: *zonas de trabajo*, donde el usuario tiene a disposición lo que le sirve de base para aprender, lleva a cabo las operaciones que desea realizar y aprecia el efecto de las decisiones que toma; *zonas de control del programa*, donde se localizan los menús y en general todos los controles que puede tener el usuario sobre el programa; y las *zonas de*

contexto para la acción a través de la cuales el usuario puede enterarse en que sección está trabajando y las ayudas ó accesorios a que puede tener acceso.

También en este diseño se deben establecer los sonidos, los colores, las animaciones y gráficos que se van a emplear en el micromundo, todo esto dependiendo del tipo de MEC y los usuarios a quienes va dirigido éste.

- *Diseño Computacional:* las funciones que va a cumplir el MEC se establecen a partir de las necesidades planteadas. Un MEC puede brindar a sus usuarios (profesores y alumnos) diversas posibilidades; por ejemplo, los alumnos podrían controlar la secuencia o cantidad de ejercicios, iniciar o terminar el programa; y el profesor podrá llevar un registro de los alumnos que utilizan el material y así saber el rendimiento que este tiene. En base a esto también podrá realizar análisis estadísticos.

En cuanto a la estructura lógica, afirma Galvis "... debería ser la base para formular el programa principal y cada uno de los procedimientos que requiere el MEC"; o sea, que debe ser de tal forma que permita el cumplimiento de todas las funciones definidas para cada usuario. También es indispensable determinar las estructuras de datos necesarios para el cumplimiento de las funciones definidas.

Para finalizar la etapa de diseño y verificar que lo que se está desarrollando va a cumplir con los objetivos propuestos y va a cubrir las necesidades detectadas, se debe elaborar un prototipo del MEC. Este se puede realizar de varias formas: por ejemplo, haciendo bocetos en papel de los ambientes, de las actividades y evaluaciones que se podrían hacer con el MEC, asociando con lo que puede ocurrir de acuerdo con las decisiones que tome el usuario; otra forma de hacerlo es utilizando el computador para definir la red de pantallazos. Esta verificación la deben llevar a cabo expertos en y una muestra representativa de los usuarios que van a utilizar el MEC. A partir de esto ellos podrán aportar sugerencias para mejorar el diseño del MEC.

3.1.3. Desarrollo

Al finalizar la etapa de análisis se recomienda determinar los recursos tanto humanos como técnicos necesarios para las etapas posteriores. Habiendo definido esto y teniendo lista la etapa de diseño se procede al desarrollo e implantación del MEC. Esta etapa se puede llevar a cabo utilizando una ó la combinación de las siguientes estrategias:

Cuando se cuenta con un grupo de especialistas en diferentes campos, en contenido, metodología e informática. En este caso el desarrollo estará a cargo del

especialista en informática, quien podrá consultar acerca de lo que se va haciendo a los demás miembros del grupo; estos también revisaran los módulos que vaya realizando el especialista en informática; es recomendable elaborar en primer lugar el módulo del profesor para que el especialista en contenido, junto con el profesor, lleven a cabo la introducción de archivos. Cuando no hay un especialista en informática se consideran dos alternativas. (1) contratar a un especialista en informática externo al grupo; (2) que el programa sea elaborado por los miembros del equipo de diseño que quieran aprender a usar un lenguaje. Si se opta por la primera alternativa, se deberá exigir al nuevo especialista en informática, realizar informes y revisiones parciales sobre lo que esta desarrollando. En el caso de que haya cambios, tendrá que consultarse con el grupo de diseño. Si se ha elegido la segunda alternativa se puede optar por dos posibilidades: utilizar un sistema autor o un lenguaje autor.

Para hacer uso adecuado del MEC, adaptarlo a nuevas necesidades y darle mantenimiento, es indispensable que se encuentre documentado el desarrollo del material y que se programe de forma estructurada y legible. La documentación puede ser de diversos caracteres.

Dentro del programa: dar nombres significativos a los procedimientos, variables y constantes; definir para cada procedimiento su función de variables de entrada y salida.

La documentación en el Manual de Usuario debe ser clara de tal forma que éste pueda resolver las situaciones que se le puedan presentar, así como también la forma de instalación y uso del MEC.

En el manual de mantenimiento se debe incluir: descripción del programa, del sistema computarizado y de las librerías imprescindibles para su ejecución; la estructura global del programa; definición de variables constantes, estructura de datos; archivos fuente y su organización. Cuando los programas están listos deben ser revisados con la ayuda de un experto, que es quien determinara si el MEC satisface las necesidades previstas, si los cambios que surgieron durante el desarrollo lo han mejorado, si se están cumpliendo las funciones definidas para cada usuario y si los contenidos cumplen con los requisitos del área temática.

Con esta revisión, el experto dará sugerencias para el mejoramiento del MEC. Sin embargo, aún no se puede afirmar que el MEC funcione perfectamente, para esto se deben realizar pruebas con los usuarios reales; ellos son los que pueden decir si en realidad el MEC funciona como se esperaba. Para llevar a cabo esta revisión, es necesario escoger una muestra representativa de usuarios, quienes probarán el MEC acompañados de uno de los desarrolladores. De esta forma se tratará de encontrar soluciones a los problemas que se presenten durante toda la ejecución del MEC y se determinara si la interfaz es apropiada para el tipo de usuarios.

3.1.4. Prueba Piloto del MEC

Con ésta se pretende ayudar a la depuración del MEC mediante la utilización de éste por un grupo representativo de usuarios para quienes fue desarrollado.

Con base a la preparación, administración y análisis de los resultados, se determina si el MEC cumple o no con los objetivos planteados.

Para realizar esta prueba se debe hacer una preparación que incluya la selección de la muestra, que consiste en escoger al azar un grupo representativo de usuarios, quienes deben cumplir con los prerequisites preestablecidos en el diseño educativo. El diseño y prueba de los instrumentos para recolectar información, que consiste en establecer que tan eficaz y eficiente es el MEC: para establecer la eficacia se requiere diseñar pruebas de rendimiento, las cuales permitirán saber cuanto aprendieron los usuarios, estas pruebas deben realizarse por lo menos al inicio y al final del uso del material, para saber por cada usuario cuanto sabía de los objetivos propuestos y cuantos alcanzó. Para establecer la eficiencia es necesario registrar por cada usuario el tiempo que interactuó con el MEC y los recursos externos al material que utilizó para el aprendizaje. Es importante además conocer las opiniones y sugerencias de los usuarios.

Cuando ya está listo el grupo escogido, se procede a utilizar el MEC en el momento en que corresponda el estudio de la temática. A cada uno de los

participantes se le realizará una prueba previa para saber que tanto saben de los objetivos propuestos. Para ejecutar el MEC se les proporciona a los alumnos todo el material necesario (software, manuales, equipo, etc.) y se deja que estos interactúen todo el tiempo que se considere conveniente para que se cumplan todos los objetivos y así poder someterlo a una última prueba. No debe ser problema si para esto los aprendices requieren más de una sesión de trabajo.

Al finalizar la prueba es conveniente un análisis de resultados, para conocer el rendimiento y la dedicación necesaria en cada usuario para lograr los objetivos. Además junto con esto se pueden establecer los aspectos negativos y positivos del MEC y las mejoras que se le pueden ó deben hacer.

Una vez obtenidos los resultados en la prueba piloto se toman decisiones condicionadas a los siguientes parámetros:

- Desechar el MEC en caso de que éste no cumpliera con los objetivos finales propuestos.
- Ajustar, corregir y adoptar para el MEC cambios que representen mejorías en el mismo.
- Hacer ajustes mayores al MEC, haciendo una retroalimentación hacia las fases que sean necesarias, en el análisis, el diseño o el desarrollo del mismo.

3.1.5. Prueba de Campo del MEC

En esta etapa se procede a la utilización del MEC por parte de toda la población objeto, lo que nos va a permitir comprobar en la vida real, que aquello que a nivel experimental tenía sentido, lo seguirá teniendo. Para que el uso del MEC tenga éxito es necesario que los docentes de la asignatura para la cual fue desarrollado, tengan una inducción acerca de la forma de manipulación total del software, así mismo en la manera de motivar y apoyar a los estudiantes durante su uso. En cuanto a los aprendices, se les debe dejar que interactúen libremente con el MEC, como se indicó en la etapa anterior, disponiendo cada uno de ellos con un tiempo mínimo de interacción. Es conveniente que la primera vez que hagan uso del material sea dentro de la hora de clase de la asignatura a que corresponde el MEC, aunque no es una camisa de fuerza.

Al igual que en la etapa anterior se puede hacer un análisis de la información sobre el rendimiento y la opinión de los estudiantes. Del resultado que se obtenga depende que el MEC se considere beneficioso para solucionar los problemas detectados, si requiere ajustes ó si definitivamente debe ser desechado. Dependiendo de la decisión que se tome se puede optar por seguir un paradigma diferente al empleado para el ciclo de vida del MEC, en esta etapa también se deben prever las necesidades futuras o cambiantes del currículo, de manera que cuando surjan nuevos requerimientos se inicie un nuevo ciclo de desarrollo.

3.2. ANÁLISIS DE RIESGOS

Actualmente los proyectos tienden a ser planificados y programados ineficazmente; los riesgos se tienen en cuenta una vez que se presentan y convierten en problemas reales. A veces parece que con el afán de comenzar, olvidamos ponernos de acuerdo y meditar un poco más en saber lo que queremos hacer. La planificación conlleva a que los organizadores y directivos del proyecto determinen y aprovechen el tiempo.

Como dice C. Torres Pastorino “dependemos del ayer pero somos dueños del mañana”, el pasado no nos pertenece y por lo tanto las acciones que realizamos hoy pueden cambiar y mejorar la situaciones futuras para así tener solidez y firmeza. El análisis de riesgos es primordial para una buena gestión de cualquier proyecto de desarrollo de software, ya que si no se combaten los riesgos ellos pueden hacer que el proyecto fracase en cualquier momento.

En cada una de las etapas de un proyecto se presentan riesgos y de la forma como se enfrente cada una de estas situaciones dependerá en gran medida el éxito o fracaso de todo el proyecto.

El planificador del proyecto debe visualizar y tener en cuenta que aspectos podrían a futuro afectar el desempeño del software, factores como el mercado, la

tecnología, los equipos, las normas, la calidad etc.,. A través de este análisis se vislumbra a un nivel más amplio los riesgos reales que podrían presentarse en el proyecto y la manera correcta de enfrentarlos para obtener soluciones oportunas y satisfactorias. El análisis consta de cuatro actividades esenciales a saber: Identificación, proyección, cálculo y gestión del proyecto.

3.2.1. Identificación de Riesgos

Consiste en clasificar los riesgos a nivel de: Proyecto, Técnicos y de Negocio.

3.2.1.1. Riesgos de Proyecto

En este aparte se identifican problemas presupuestarios, de agenda, de personal (organización y asignación), de recursos y requisitos del cliente, así como su impacto sobre el proyecto de software. Se incluyen también factores como tamaño, complejidad y estructura del proyecto.

En el proyecto de Software Educativo SEGraD se identificaron los siguientes riesgos de Proyecto:

- Dedicación parcial al proyecto debido a la realización de otras actividades académicas.(1)
- Presencia de problemas durante el diseño que retrasen el tiempo presupuestado para esta etapa. (2)
- Que el software no satisfaga las necesidades de los usuarios. (3)
- Falta de asesorías de docentes en el tema y en el lenguaje. (4)
- Información o fuentes obsoletas o erróneas. (5)
- Metodología pedagógica inapropiada. (6)
- Lenguaje de programación no apropiado. (7)
- Temas inapropiados en el currículo del material educativo. (8)

3.2.1.2. Riesgos Técnicos

En éste nivel de riesgos se identifican problemas de diseño, implementación, interfaz, verificación y mantenimiento, además se tiene en cuenta factores como la ambigüedad de la especificación, incertidumbre técnica, obsolescencia técnica y tecnología de punta.

En el proyecto de software educativo SEGraD se identificaron los siguientes riesgos técnicos:

- Retraso de la implementación. (1)
- Versatilidad: en la implementación puede ocurrir, que el paquete usado para esta aplicación, no sea suficientemente versátil para conseguir los resultados esperados. (2)
- Nuevas Tecnologías:(versiones superiores de software, uso de ambientes de realidad virtual) El software podría quedar a un bajo nivel con relación a las versiones de software que se este usando para la implementación de MECs. (3)
- Ejecución del software: se puede presentar que el software no corra satisfactoriamente en el equipo preestablecido, ya que al usar una gran cantidad de animaciones el equipo presente interrupciones continuas y la ejecución del software sea lenta. (4)
- Uso de dispositivos: es posible que se presente mal uso de dispositivos como de audio (parlantes, audífonos), y de entrada de datos (teclado y mouse). (5)
- Escaso soporte técnico y deficiencia de recursos (Hardware y Software). (6)
- Retraso en el diseño de la interfaz amigable. (7)
- Diseño mal enfocado para el logro de los objetivos. (8)

3.2.1.3. Riesgos del Negocio

Tienen que ver con el mercado incierto de venta del proyecto, la forma de vender el producto, el soporte que se ofrece y el presupuesto con que se cuenta.

En el proyecto de software educativo SEGraD se tuvieron en cuenta dos tipos de riesgos de Negocio, de mercado y de costo:

Riesgos de mercado:

- Puede presentarse que después del desarrollo del software educativo ninguna institución quiera implementarlo entre sus planes educativos.
- Negativa aceptación comercial del software a pesar de ser de buena calidad.
- No existe la certeza de que la inversión que se haga en el software se recupere con la venta del mismo.

Riesgos de costo:

- Es posible que para la ejecución del software se necesiten equipos con hardware especial lo que aumentaría los costos, esto lo haría poco viable.

3.2.2. Proyección o Estimación de los Riesgos

La escala de estimación del riesgo esta dada en variables cualitativas y variables probabilísticas matemáticamente dependiendo del riesgo.

Los porcentajes van de 1 a 100% dependiendo del grado de probabilidad del riesgo, los intervalos son asignados de forma personalizada dependiendo de los desarrolladores, nosotros consideramos una escala de intervalos que definen de forma clara e imparcial los tipos de riesgo detectados, así:

Bastante Improbable	(BI)	→	0 – 20%
Improbable	(I)	→	21 – 40%
Moderado	(M)	→	41 – 60%
Probable	(P)	→	61 – 80%
Bastante Probable	(BP)	→	81 – 100%

Riesgos del proyecto:

Riesgos	BI	I	M	P	BP
(1)			*		
(2)				*	
(3)			*		
(4)		*			
(5)		*			
(6)			*		
(7)			*		
(8)			*		

Cuadro 4.

RIESGOS DEL PROYECTO

En general los riesgos del proyecto son moderados con un promedio de 47%.

Riesgos del técnicos:

Riesgos	BI	I	M	P	BP
(1)		*			
(2)		*			
(3)		*			
(4)					*
(5)				*	
(6)		*			
(7)			*		
(8)			*		

Cuadro 5.

RIESGOS TÉCNICOS

En general los riesgos Técnicos son moderados con un promedio de 38%.

Riesgos del Negocio:

Riesgos	BI	I	M	P	BP
Mercado			*		
Costo			*		

Cuadro 6.

RIESGOS DEL NEGOCIO

En general los riesgos del Negocio son moderados con un promedio de 41%.

3.2.3. Naturaleza de los Riesgos

Indica los problemas potenciales que se pueden producir de acuerdo al grupo de riesgos a los cuales pertenecen.

Riesgos del proyecto: el principal problema sería que el proyecto no alcance los objetivos propuestos debido a deficiencias del aprendizaje de los usuarios ó al mal uso por parte del docente ó los mismos usuarios (estudiantes).

Riesgos técnicos: el principal problema que se puede presentar es que el software no sea implementado debido al mal uso de los dispositivos ó la falta de un equipo adecuado para la ejecución.

Riesgos del Negocio: el problema que se puede presentar es que el software no sea implementado debido a los altos costos.

3.2.4. Prioridad del Riesgo

De acuerdo con la identificación y análisis de los riesgos en cada uno de los grupos se puede determinar la prioridad ó impacto que tienen estos mediante la siguiente tabla:

1. Más probable.	Prioridad	Grupo Riesgo
	1	Riesgos Proyecto
2. Medianamente probable.	2	Riesgos Técnicos
3. Menos probable.	3	Riesgos Negocio

Cuadro 7.

PRIORIDAD DE LOS RIESGOS

3.2.5. Evaluación de Riesgos – Relación (Riesgo, Probabilidad, Impacto)

Para evaluar los riesgos se debe desarrollar una relación de acuerdo con los datos proporcionados por el análisis de riesgos, así:

Riesgo, probabilidad del riesgo, impacto de cada grupo de riesgos. Se debe examinar también la exactitud de las estimaciones que se han realizado durante la proyección del riesgo; se intentara dar prioridad a los riesgos que no se han cubierto y se comenzará a pensar en las formas de controlar y prevenir los riesgos que tengan mayor probabilidad de ocurrir. Para que la evaluación tenga un mejor resultado se define un nivel de referencia para cada riesgo.

3.2.5.1. Niveles de Referencia

Este aparte se encuentra enmarcado entre tres niveles típicos de referencia como son:

Coste, agenda y rendimiento; es decir si hay un nivel de exceso de coste, de excesiva duración ó de degradación de rendimiento (ó cualquier combinación de las tres) originara que no pueda haber un desarrollo normal del proyecto y en los peores casos que el proyecto fracase. De acuerdo con lo anterior:

- *El coste:* puede presentarse que el costo del proyecto sea demasiado alto y por falta de recursos y financiamiento no se logre cumplir con todos los objetivos propuestos.

- *Agenda*: de acuerdo con los riesgos enumerados anteriormente, es posible que motivos (tales como falta de información oportuna, falta de personal) atrasen el proyecto y no sea terminado en el plazo esperado.
- *Rendimiento*: si no se cumple a tiempo un actividad de acuerdo al cronograma de actividades, se presentara recargo de trabajo en las posteriores semanas.

3.2.5.1.1. Relación de las Ternas con los Niveles de Referencia

Las ternas están conformadas, en primer lugar por el grupo de riesgo, en segundo lugar por la probabilidad del grupo de riesgos y por último por la prioridad ó impacto del grupo de riesgos; lo anterior teniendo en cuenta los niveles de referencia. (Riesgo Proyecto, probabilidad %, impacto o prioridad)

1. (Riesgo del Proyecto, 47%, 1)

- *Coste*: es Moderadamente probable que ocurra esta clase de riesgos, ya que en el momento no se cuenta con un cliente definido, pero el mercado es bastante amplio.
- *Agenda*: es Moderadamente probable que ocurra esta clase de riesgos, y se considera con una probabilidad de primer lugar, ya que la falta de datos

oportunos por parte de la fuente de información ocasiona atrasos dentro del cronograma de actividades.

- *Rendimiento:* es Moderadamente probable que ocurra este tipo de riesgos debido a que puede presentarse acumulamiento de actividades.

2. (Riesgos Técnicos, 45%, 2)

- *Coste:* es Moderadamente probable que ocurra esta clase de riesgos, ya que estos hacen referencia a la parte técnica y tecnológica, donde se puede tener un control relativo; de todos modos se tratara de utilizar una tecnología adecuada para este tipo de proyectos
- *Agenda:* es Moderadamente probable que ocurra esta clase de riesgos, ya que se van a utilizar técnicas apropiadas o básicas que permitan conseguir la facilidad de adecuación a tecnologías de innovación, por tanto durante el periodo de desarrollo no se cambiaran estos.
- *Rendimiento:* es Moderadamente probable que ocurra este tipo de riesgos debido a que puede presentarse que la aplicación se torne muy pesada.

3. (Riesgos del Negocio, 50%, 3)

- *Coste:* es Moderadamente de que ocurra este tipo de riesgos porque el mercado educativo para este tipo de proyectos es muy amplio.
- *Agenda:* es Moderadamente que ocurran este tipo de riesgos, ya que como ingenieros de sistemas somos los encargados de promover y agilizar el

desarrollo y por lo tanto no traería consecuencias considerables dentro del cronograma de actividades.

- *Rendimiento:* es Moderadamente probable que ocurran estos riesgos porque cada día se ve la necesidad de nuevas tecnologías.

Combinaciones de Riesgos: al combinarse todos los riesgos afectaran principalmente al recurso tiempo, de una manera indirecta al recurso monetario, al no tener la financiación de una empresa o ente económico.

3.2.6. Gestión y Supervisión de Riesgos

Se determina los pasos de gestión del riesgo a partir de los temas asociados con cada Riesgo, con el fin de lanzar posibles soluciones que remedien el impacto de cada riesgo. Estos pasos están organizados en un plan de gestión y supervisión de riesgos; las soluciones se plantean de acuerdo al grupo de riesgo al cual pertenecen.

- Riesgos del Proyecto:
 - Asesorarse de personal idóneo para el desarrollo del software y las herramientas empleadas.

- Poner en práctica simulacros de algunas temáticas para observar el comportamiento y los resultados tanto del software como de los usuarios potenciales y la respuestas frente a la metodología de enseñanza utilizada.
 - Instruir al docente y a cada uno de los usuarios de SEGraD en el manejo de las interfaces y dispositivos.
 - Especificar al centro educativo donde se implementará que tipo de usuarios pueden hacer uso del software.
 - Seguir al máximo y cumplir el cronograma con todas sus especificaciones.
- Riesgos Técnicos:
 - Asesorarse de personas idóneas en el campo de diseño de Software Educativos.
 - Hacerse de los recursos necesarios de hardware y software actualizados que permitan un continuo desarrollo sin contratiempos.
 - Unificar criterios por parte del personal que brinda la asesoría tanto en la parte metodológica, como en la técnica.
 - Utilizar de la mejor manera las tecnologías a nuestro alcance, mediante diseños orientaos a futuro que permitan soportar innovaciones posteriores.

- Definir desde un principio unas características mínimas para los equipos utilizados para la implementación del software que permitan el óptimo desarrollo del mismo.
 - Realizar las pruebas de software y hardware necesarias que aseguren un correcto desempeño durante la implementación.
 - Brindar un mantenimiento periódico y garantizar acceso a las actualizaciones posteriores.
- Riesgos de negocio:
 - El software dado a conocer mediante conferencias, charlas y socializaciones a personas e instituciones interesadas.
 - Capacitar al personal que da a conocer el software, para que motive a quienes se interesen por el mismo.
 - Lograr un estándar que permita al software correr en equipos de no muy altas especificaciones técnicas, que es el caso de la mayoría de instituciones educativas y así obviar los altos costos de implementación.
 - Recurrir a una plataforma de lanzamiento reconocida para el software como lo es la Universidad de Nariño.

3.3. MÉTRICAS DE CALIDAD

Todo proyecto sin importar el área a la cual pertenece debe cumplir una serie de estándares que determinan la calidad de éste. El software a su vez cumple con una serie de características propias que determinaran su aceptación.

3.3.1. Métricas de Calidad de Software

La calidad del software es una compleja mezcla de ciertos factores que varían para diferentes aplicaciones y los clientes que las solicitan. Estas se centran en tres aspectos importantes, como son:

Las características operativas, su facilidad de soportar los cambios y su adaptación a nuevos entornos.

Características operativas, dentro de estas se plantean:

- *Corrección:* el MEC SEGraD sigue las características y temáticas proporcionadas por el centro educativo Udenar, por lo cual este es un aplicativo diseñado con una propósito específico, en un área específica (Teoría de Grafos).

- *Fiabilidad:* la confiabilidad en los datos proporcionados por el MEC SEGraD es de un 95 % o más, ya que al ser un software educativo utilizado en el aprendizaje de teoría de grafos, la precisión conceptual dependerá inicialmente del instructor (a).
- *Eficiencia:* el equipo en el cual correrá el programa será adecuado para este (características técnicas del equipo).
- *Integridad:* el software tendrá un 95 % o más de seguridad ya que solo el docente tendrá acceso a los registros y reportes de los logros alcanzados por cada alumno.
- *Fiabilidad de uso:* esta es del 90%, el software está diseñado para ser trabajado y aprobado en forma fácil, siendo de muy sencilla comprensión los resultados proporcionados por éste, además, los usuarios tendrán la opción de recurrir al docente, tutor o facilitador (ó como quiera llamarse) para resolver cualquier inquietud en las temáticas ó en el uso del computador.

Resolución del producto (cambios), dentro de estos están:

- *Facilidad de mantenimiento:* esta temática se encuentra ubicada entre un 95% y 100%, ya que el diseño y desarrollo del software será realizado en su

totalidad por el mismo equipo de trabajo, el cual conocerá a profundidad el código del programa.

- *Flexibilidad:* entendiendo este término como aumentar y modificar las temáticas este software tendrá un 100% de flexibilidad.
- *Facilidad de prueba:* dependerá en gran medida del estado anímico de la población objeto, por eso se considera que tendrá un 80% de facilidad de prueba.

Transición del producto (adaptabilidad), aquí cabe mencionar:

- *portabilidad:* el software correrá en un equipo de iguales características o mayores a las mínimas requeridas.
- *Reusabilidad:* debido a que el paquete está diseñado para un propósito específico su reusabilidad está en un 80%, siendo reusable en aplicativos con semejante características; es decir aplicativos que se utilizan para reforzar el aprendizaje de Teoría de Grafos.
- *Facilidad de interoperación:* al igual que en el anterior por ser un programa de propósito general la interoperación será del 80%.

Debido a la dificultad y en algunos casos la imposibilidad de desarrollar medidas directas de los anteriores factores de calidad, es necesario definir un conjunto adicional de métricas para desarrollar expresiones por cada uno de los factores,

tomando como base listas de comprobación propias para el software que se esta realizando. Entre estas tenemos:

- *Compleitud:* el software tendrá implementadas las funciones necesarias para la facilidad en su manejo y buen funcionamiento.
- *Concisión:* se utiliza para el desarrollo del aplicativo, programación orientada a objetos, que permitirá comprimir en algunos casos el código del programa. (manejo de herencia, relación entre objetos, etc.)
- *Consistencia:* la documentación del proyecto se esta llevando a cabo durante cada una de las etapas del ciclo de vida del mismo, pasando por el análisis de necesidades educativas, diseño educativo, diseño del sistema de comunicación, diseño computacional, desarrollo, prueba piloto, prueba de campo.
- *Tolerancia en errores:* el software diseñado será capaz de enfrentarse a errores generados por el usuario, considerando las posibles respuestas a estos.
- *Facilidad de expresión:* se podría ampliar el diseño, y los proceso que realiza el programa, siempre y cuando los nuevos módulos estén directamente relacionados con la temática general del software
- *Modularidad:* el MEC hará uso de este tipo de métrica con varios propósitos:
 - Para facilitar el mantenimiento del software, ampliar funciones y reutilizar algunos módulos para la implementación de nuevos sistemas.

- El programa se implementara de tal forma que no presente dificultad en su uso, tomando en cuenta características especiales de la población destino.
- El software debe poseer mecanismos especiales que aseguren la integridad de la información que se maneje, en especial de los estadísticos de cada estudiante, el cual será revisado y analizado por el docente o tutores.
- *Auto documentación:* la totalidad del código y el funcionamiento del software queda documentada facilitando la revisión y el mantenimiento del programa.
- *Simplicidad:* el programa podrá ser entendido sin dificultad por la sencillez y facilidad del manejo de los temas.
- *Independencia del sistema de software:* debido a que el aplicativo es diseñado para funcionar bajo un ambiente Windows, este estará dependiente del Sistema Operativo en el cual corre.
- *Formación:* gracias a la facilidad de uso el software podrá ser usado por un amplio grupo de usuarios.

3.3.2. Métricas de Calidad del Área Temática

La calidad de un MEC debe enmarcarse de la perspectiva educativa como informática. La primera se logra realizando un diseño educativo comprometido con el objetivo que se quiere alcanzar, aplicando los principios psicológicos y comunicacionales mas adecuados al tipo de objetivo y a la población a la cual va

dirigido el material educativo. La clave para lograr calidad desde la perspectiva informática, esta en usar técnica de especificación y una metodología de diseño y programación, que permita desarrollar en forma natural los factores de calidad del software.

De acuerdo con los criterios expuesto con anterioridad se puede considerar dos perspectivas igualmente importantes para determinar la calidad Temática de un MEC:

Se busca que el software sea:

- *Interactivo*: puede ser esta la característica fundamental para el logro del fin que se desea conseguir, acaparando toda atención del usuario, para lograr esta interactividad se refleja en el MEC la estructura de un mundo al que se hace referencia utilizando un micromundo (interfaz) atractivo y amigable para los usuarios, rodeado de color, sonidos, formas, donde estos sean capaces de comunicarse con la maquina de una manera fácil y divertida.
- *Transportable*: debido a la gran variedad de equipos computacionales que existen en las diferentes instituciones educativas con características graficas heterogéneas, el software educativo basado puramente en ambientes gráficos no puede funcionar satisfactoriamente siendo esta métrica subordinada por unas características mínimas que tienen que cumplir los equipos (debe poseer

por lo menos parlantes, audífonos, tarjeta de video VGA o superiores entre otras), además para facilitar el manejo del aplicativo se provee “tal cosa”.

- *Adaptable:* al igual que la anterior métrica, ésta hace referencia a características propias de los equipos que poseen los diferentes usuarios, y tiene que ver con las limitaciones de memoria principal. Se da como características mínimas 8 MB de Ram y un espacio en disco de mínimo 50 Mb.
- *Confiable:* el software tendrá implementado las funciones necesarias para su correcto funcionamiento como son iniciar, salir etc, y una interfaz amigable para obtener resultados satisfactorios en la relación hombre – máquina.
- *Eficiente:* debido a que se trata de un Software Educativo interactivo se Aprovechará al máximo características multimediales (sonidos; animaciones) haciendo un buen uso de los recurso con los que se dispone.

Desde el punto de vista del administrador ó el encargado del mantenimiento las siguientes condiciones son deseables:

- *Modularidad:* gracias a las facilidades de trabajo con los programas orientados a objetos se facilitara la agrupación de objetos con atributos y características similares obteniendo módulos fácilmente ensamblables dentro del programa ó en nuevas aplicaciones.
- *Responsabilidad:* se podrá reutilizar aquellos módulos independientes con funciones bien definidas, pero con relación al módulo de interfaz del

micromundo, ya que esta realizado pensando en la población objeto, únicamente se reutilizará en caso de realizar ampliaciones ó aplicaciones similares a la original.

- *Mantenimiento:* debe contener su documentación completa, los manuales de programador y de usuario que permitan la manipulación del software cuando se requiera, debe permitir la actualización de registros y ejemplos dentro de cada módulo temático.

3.4. ESTRATEGIA DE SOLUCIÓN

3.4.1. Análisis de Solución

Para poder atender las necesidades o resolver los problemas detectados dentro de la institución, es imprescindible saber a que se debieron y que puede contribuir a una posible solución. Se puede afirmar que el problema esencial acaecido es el la dificultad que tienen los estudiantes para aprender y asimilar de manera eficiente los temas relacionados con las Estructuras de Información, específicamente la relacionada con la Teoría de Grafos, buscando despertar el desarrollo de una actitud investigativa que conlleve a un acrecentamiento intelectual. Por lo tanto conviene buscar una forma de solución que permita

resolver dichas deficiencias y el como eventualmente la aplicación de un MEC podría ser de gran utilidad.

Una institución esta conformada por ciertos elementos que conforman en sí el sistema educativo, a saber: los alumnos, los docentes, administración, las metodologías empleadas, entre otras. En cuanto a los alumnos, la falta de motivación para estudiar los temas sobre la Teoría de Grafos, es sin lugar a dudas el punto crucial para la aplicación de MEC, existiendo alumnos con dificultades de asimilación que al no ser tenidos en cuenta perturban el buen desarrollo metodológico de los docentes.

Por otra parte, los materiales que posee el aula de estudio suelen ser en algunos casos defectuosos, trayendo como consecuencia el desinterés que el alumno no puede expresar hacia ellos (libros, figuras, revistas, tablero). El docente también puede ser causa de un posible fracaso de los estudiantes a no impartirles la suficiente motivación. Lo que se logra es crear un ambiente de desanimo, desinterés y la no comprensión de los temas expuestos perjudicando el proceso de aprendizaje.

El tiempo que se le dedica a algunos temas no suele ser el más apropiado, se da el caso de tiempo insuficiente o por el contrario escaso para la transmisión de los conocimientos, aburriendo a los estudiantes o creando aprietos a los estudiantes

que no logren asimilar la temática expuesta lo que traerá como consecuencia que los temas siguientes se tornen desagradables.

La metodología que se puede impartir no esta acorde con el proceso de aprendizaje, es decir, la forma como se le dicta clase al alumno evitando explotar al máximo la creatividad, el ingenio de cada uno de ellos; enfocándose en una enseñanza tradicional, la cual ya es obsoleta.

En vista de las causas anteriormente anotadas es imprescindible que la solución a implementar sea dirigida por el camino de la tecnología y en particular a través de la informática, con el desarrollo de un Software Educativo.

El MEC se regirá bajo los lineamientos de un enfoque Algo – Heurístico donde se combinan la transmisión y asimilación con el descubrimiento y la apropiación de un objeto de conocimiento, generando en los estudiantes habilidades y destrezas que mejoren su creatividad y capacidad intelectual.

3.4.2. Desarrollo de Solución

Para el desarrollo de MEC se han considerado los siguientes parámetros como la clave del éxito del mismo:

1. Utilización de Software de desarrollo de alto nivel

Que facilite el aumento de la productividad, los aspectos de auto documentación, que brinde confiabilidad y características adicionales para mejorar el desarrollo del MEC, entre las características necesarias están:

- Verificación fuerte de tipos de datos.
- Abstracción de datos.
- Compilación separada.
- Manejo de excepciones e interrupciones.
- Mecanismos de concurrencia.

Las herramientas de programación utilizadas en lenguajes de alto nivel van desde las mas elementales como depuradores hasta los compiladores, por ello se tomo la determinación de desarrollar el software por medio de un lenguaje visual, específicamente “Visual Basic versión 6.0 Empresarial”, el cual por su manejo muy sencillo de objetos y eventos nos facilitara el desarrollo de las interfaces, las simulaciones y el apoyo grafico requerido para la implementación de nuestro MEC.

Visual Basic funciona bajo ambientes Windows y permite la programación del MEC de una manera estructurada y legible lo que facilita al mismo tiempo la

documentación propia del trabajo, características muy importantes a la hora de realizar una actualización o modificación que la institución donde se implemente crea conveniente. En este tipo de lenguajes de desarrollo, los errores son muy fáciles de detectar y resolver, lo que ayuda a una programación mas estable y eficiente. En lo particular Visual Basic permite la incorporación de menús, cuadros de texto, botones de orden y de opción, casillas de verificación, cuadros de listas, barras de desplazamiento, fácil manejo de archivos, se pueden utilizar cuadrículas para utilizar datos tabulados, permite compatibilidad y comunicación con otras aplicaciones, el manejo de las bases de datos es muy seguro y sencillo. En cuanto a la interfaz, permite mantener activas varias ventanas en una pantalla de la misma aplicación o de otras tipo Windows ejecutadas simultáneamente.

Visual Basic 6.0 ed. Empresarial es una herramienta de programación que enfoca el desarrollo de aplicaciones a objetos y eventos, posee características relevantes que vale la pena mencionar como son: maneja de forma muy eficaz la encapsulación y la herencia, posee una gran cantidad de controles que permiten su programación por medio de eventos, estos controles pueden personalizarse por medio de una tabla de propiedades (posición, color tamaño, habilitarlo, verlo, etc); de ellos no se puede diseccionar su interior ni alterar la forma en que funcionan, pero si se permite al programador crear sus propios controles. Permite polimorfismo, es decir, múltiples formas de implementar la misma definición de

interfaz, logra también compilar programas en ficheros EXE en código totalmente nativo y tener como consecuencia programas más veloces.

Los editores del código tienen características especiales, como permitir seleccionar bloques de código y arrastrarlos hasta un nuevo lugar en vez del antiguo procedimiento de cortar y pegar, las ventanas de código poseen también márgenes que permiten colocar puntos de ruptura o interrupción de compilación.

Las ventanas de propiedades permiten agrupar las mismas por categorías o para ahorrar tiempo visualizar las de mayor interés de forma rápida.

Por último en Visual Basic 6.0 ED. Empresarial el viejo estilo de interfaz único (SDI) ha sido sustituido por el estilo de interfaz de documento múltiple (MDI) más convencional y mas conveniente, con el que en un formulario de VB, las ventanas de código, diseño y propiedades están contenidas en una sola ventana principal.

Dentro de las ventajas que se suceden al implementar el MEC con este tipo de software están:

- Poseer herramientas para el manejo de ventanas, menús y objetos de fácil acceso.
- Soportar aplicaciones y objetos de Microsoft Windows 9x.

- ❑ Aprovechar las ventajas de la arquitectura cliente – servidor.
- ❑ Soportes para exportar o importar archivos.
- ❑ Fácil manejo de las bases de datos.
- ❑ Las interfaces de usuario manejadas son muy amigables.
- ❑ Permite crear aplicaciones basadas en imágenes, videos, sonidos, lo que facilita la creación de animaciones y ambientes interactivos.
- ❑ La utilización del estilo de MDI facilita el diseño y la programación del software.

Además de Visual Basic se hizo uso complementario de otras herramientas de programación y manejo de imágenes que fueron indispensables para el desarrollo e implementación del MEC, entre las que destacamos:

- 🖥️ Plataforma o Sistema Operativo: Microsoft Windows Me
- 🖥️ Herramienta de propósitos generales: Microsoft office 2000
- 🖥️ Herramienta de manejo de imágenes: Adobe Photoshop 6.0
- 🖥️ Herramienta de manejo y creación de iconos: Microangelo 98
- 🖥️ Herramienta de manejo y creación de Gifs animados: GIF Movie Gear

2. Utilización de Hardware para desarrollo y montaje de la aplicación

El desarrollo del MEC necesita de unos requerimientos específicos de hardware para su desarrollo e implementación. SEGraD fue diseñado e implementado en un equipo genérico con las siguientes características:

- Procesador Pentium III de 500 Mhz.
- 64 MB de Memoria Ram.
- Disco duro de 20 GB
- Unidad de CD Room 32x Creative.
- Tarjeta de sonido Sound Blaster
- Tarjeta de video AGP de 8 Mb
- Impresora Hewlett Packard 692

3. Un grupo de trabajo Interdisciplinario:

Un grupo de personal selecto idóneo, que ayude en el desarrollo del software educativo y que sean especialistas cada uno en su área de trabajo. Entre el personal que se considero necesario para la creación del MEC están:

- Analistas y Programadores: Encargados de elaborar y traducir todo el análisis y el diseño en un lenguaje de alto nivel, en este caso Visual Basic V. 6.0 ed. Empresarial.
 - Héctor F. Guzmán E. y Alex D. Burgos V.
- Especialista en software y pedagogía educativa: quien orientará a los desarrolladores en cuanto a la pedagogía y las características que debe llevar un software de este tipo.

- Vicente chamorro (Docente Udenar)

- Especialista en Teoría de Grafos: quien definirá la temática que debe llevar el software para que cumpla con los objetivos propuestos de la asignatura en la cual será aplicado.
 - Manuel Bolaños (Docente Udenar)

- Tecnólogo en sistemas: quien será el encargado de verificar el correcto funcionamiento del software y hardware involucrado durante el desarrollo.
 - Byron Villarreal

3.4.2.1. Alternativas de Solución para el Desarrollo del Software

Se plantea una Herramienta de apoyo al aprendizaje que permita:

Al estudiante:

- Integración con el MEC mediante una interfaz didáctica que no involucre mucho elementos.
- Transmisión de conocimientos acerca de la Teoría de Grafos.
- Afianzamiento y practica.

- Autoevaluación.
- Evaluación.
- Registro del avance en forma personalizada.
- Fomentar el espíritu competitivo generado durante las evaluaciones sumativas del software.

Al docente:

- Complementar el desarrollo de sus clases con el material educativo computarizado, que se convierte en un elemento de motivación para el estudiante en la materia de Estructuras de Datos.
- Motivar a los estudiantes teniendo en cuenta los resultados obtenidos durante el proceso de autoevaluación de los estudiantes.
- Evaluar y registrar las acciones del estudiante mediante una base de datos protegida que almacene la historia del estudiante, y/o usuario durante el proceso de aprendizaje.

3.4.2. Cronograma de Actividades (Ver Anexo 2)

4. GESTIÓN DEL PROYECTO

4.1. ENFOQUE Y TIPO DE INVESTIGACIÓN:

El enfoque de la investigación es empírico – analítico y el tipo de investigación es descriptivo – explicativo.

Enfoque empírico – analítico:

Permite descomponer las realidades que se investigan a fin de identificar los elementos y relaciones causadas que pueden ser controladas.

Para identificar los problemas existentes en el proceso de aprendizaje de la Teoría de Grafos se centra la atención de los protagonistas del medio educacional que son los profesores, los estudiantes y la institución educativa.

Tipo de investigación descriptivo – explicativa:

El objetivo de este método es determinar las características esenciales de la población educativa y la realidad problemática que los afecta.

4.2. NECESIDAD EDUCATIVA

La creación de ambientes Enseñanza – aprendizaje apoyados con computador tiene sentido si responde a necesidades educativas prioritarias y relevantes y no existe otra solución que la satisfaga.

Con el fin de participar en la implantación de una infraestructura que permita la introducción del computador en el sistema educativo regional con un criterio pedagógico, la investigación de “SEGrAD” apoyará en principio el proceso del aprendizaje de la teoría de grafos aplicado a la solución de problemas con solución computacional.

Se plantea la necesidad de la implementación de un nuevo tipo de recurso educativo diferente a los libros y al discurso expositivo verbal, porque no todos los conocimientos pueden estar contenidos en ellos; recursos que permitan una mayor interacción de los estudiantes con el medio, recursos que logren una educación más creativa y afectiva basada en las experiencias de los alumnos y que eviten el tradicional verbalismo.

Los materiales educativos y su utilización se determinan a partir de lo que se quiere enseñar y de los niveles de aprovechamiento que se esperan de los estudiantes o aprendices. Estos aspectos se conocerán a partir de una

investigación de la población objeto para concretar el nivel de enseñanza que se desea impartir. En esta etapa, inicialmente, se pretende conocer las necesidades educativas de los estudiantes y en general del medio educativo, y de las cuales se parte para crear soluciones con el fin de subsanar estas deficiencias.

Las necesidades educativas se refieren a la desigualdad entre un estado educativo ideal (lo que debe ser) y otro existente (realidad). Por consiguiente su determinación debe llevar a resolver tres interrogantes: ¿Qué es lo ideal? (meta o aprendizaje esperado), ¿Qué de esto se puede satisfacer con lo que existe? (lo que se puede lograr o aprender con los medios y actividades aplicables), ¿Qué falta por alcanzar? (la necesidad). De este modo la determinación de necesidades educativas en el entorno de enseñanza – aprendizaje es equivalente al establecimiento de lo que hay que aprender con un apoyo de un ambiente y de recursos con que se cuenta.

La necesidad sentida⁹ es la que se presenta en la población objeto, la cual hace referencia al deseo de la comunidad educativa por saber algo. Por lo cual, la necesidad presente en la necesidad educativa debe identificarse simplemente preguntándole a los estudiantes que cursan la asignatura Estructuras de Información que les interesaría aprender de la Teoría de Grafos.

⁹ Burton, J.K.. Educational technology publications. Cit. Galvis. Ingeniería del Software Educativo P. 122

4.2.1. Fuentes para la Determinación de Necesidades Educativas

Los datos sobre las necesidades surgen al interactuar con los usuarios potenciales o con quienes conocen de verdad las necesidades de estos. Las fuentes de información definidas dentro del proyecto denominado “SEGraD” son los docentes y estudiantes de la Universidad de Nariño.

La recolección de fuentes de información se fundamenta en los siguientes puntos:

- ❑ Consultar los planes de estudios vigentes para establecer las metas educativas establecidas para los componentes y niveles de interés.
- ❑ Indagar con una muestra representativa de la población objeto sobre lo que les interesa aprender, según sea el caso, para el trabajo, para la vida ó simplemente porque les llama la atención.
- ❑ Indagar acerca de lo que en programas equivalentes se ofrece a destinatarios ó poblaciones educativas similares.

Como punto de partida para identificar las necesidades del medio educativo del software “SEGraD” se aplicaran encuestas a una muestra representativa de estudiantes. Los estudiantes son fuentes de información primaria para detectar y priorizar aspectos problemáticos.

Para conocer a fondo las necesidades educativas se hizo necesario definir unos objetivos:

- Conocer las características del proceso de información y las estrategias metodológicas empleadas en las instituciones para el conocimiento de las diferentes técnicas y herramientas utilizadas en la enseñanza de la Teoría de Grafos.
- Encontrar la necesidad de crear e implementar un software para el aprendizaje en el desarrollo de problemas por medio de Grafos.
- Detectar las deficiencias en la solución de problemas al crear, programar, utilizar y aplicar Grafos.
- Conocer que herramientas computacionales se utilizan como apoyo en el proceso de enseñanza – aprendizaje.

4.2.2. Tipo de Muestra

El tipo de muestra seleccionado para nuestra investigación es no probabilístico, ya que la selección de los elementos no provienen de la probabilidad, sino de las causas relacionadas con las características de la investigación (lo que causa el problema ó deficiencia), el procedimiento no es mecánico ni en base a fórmulas de probabilidad, sino que depende del proceso de toma de decisiones en forma

grupales, desde luego las muestras tomadas por decisiones subjetivas tienden a estar sesgadas (desviadas de lo que se pretende encontrar). La muestra es tomada de la población estudiantil perteneciente a la facultad de ingeniería, Programa de Ingeniería de Sistemas de la Universidad de Nariño, donde se imparte el aprendizaje de la teoría de grafos a nivel profesional.

4.2.2.1. Población Objetivo

Es importante conocer los destinatarios del MEC, en buena parte los sistemas de motivación y comunicación que se decidan elaborar depende de quienes son los usuarios potenciales del software. Para establecer sus principales características conviene resolver incógnitas como las siguientes:

- ¿Entre qué rango de edades están definidos?
- ¿Qué nivel de educación poseen?
- ¿Qué intereses y expectativas pueden tener los aprendices respecto al tema y objetivos que se pretendan lograr?
- ¿Qué experiencias tienen relevantes para el estudio del tema?
- ¿Tienen alguna actitud o característica que deba tomarse en cuenta?

El Software Educativo “SEGraD” esta dirigido tanto a estudiantes que cursen carreras universitarias en cuyo pensum se incluya la rama de la Teoría de Grafos, así como también para los docentes que tengan a su cargo esta materia con el fin de apoyar el proceso de aprendizaje de la temática, respondiendo a las expectativas y objetivos que se pretenden lograr y que estén expuestos dentro del pensum establecido.

4.2.2.2. Área de Contenido

Es de vital importancia reconocer las áreas de trabajo que van a ser beneficiadas por medio del desarrollo del material, ya que de estas depende el contexto sobre el cual la ejemplificación y la ejercitación serán diseñadas. Para tener claridad sobre el área de contenido se debe tener en cuenta por lo menos dos incógnitas:

- ¿Qué área de formación, de contenido y de instrucción se benefician con el desarrollo de este MEC?
- ¿Qué unidades de instrucción presentan problemas con el tema y los objetivos que se van a apoyar con el estudio del MEC?

SEGraD es desarrollado con el objetivo de apoyar el proceso de aprendizaje de la Teoría de Grafos; el área de contenido que se va a beneficiar con el material es:

Resolución de problemas mediante un programa de computador sobre la temática especificada, “Teoría de Grafos” , el cual cumple como mínimo con las siguientes fases:

- Aprendizaje de la teoría de forma expositiva y simulada.
- Afianzamiento y práctica de la teoría.
- Interpretación y análisis de un problema.
- Diseño y representación del problema en un Grafo (Forma dinámica).
- Resolución del Grafo por el algoritmo correspondiente.
- Evaluación.

4.2.2.3. Instrumentos para la Recolección de Información

La recolección de información se hizo a través de dos técnicas muy reconocidas y fáciles de interpretar: las encuestas y las entrevistas. Considerando que los profesores y los estudiantes son las principales fuentes potenciales de información se realizó la consulta de la siguiente manera:

❑ Consulta a estudiantes:

Los estudiantes son fuente de información primaria para detectar y priorizar aspectos problemáticos. el cuestionario fue manejado teniendo en cuenta las

dificultades y deficiencias en el proceso de aprendizaje, además los recursos y/o herramientas empleadas para la labor educativa. (ver Anexo 1 - encuestas a estudiantes).

□ Consulta a profesores:

El profesor es el ente guía, orientador y facilitador, quien colabora a desarrollar el potencial intelectual, social y cultural del educando, ellos saben con mayor confiabilidad en que puntos del contenido hay deficiencias y como o cuando los métodos de enseñanza empleados están fallando y se quedan cortos frente a las características del estudiante y los requerimientos del plan de estudio que guía la acción.

El objetivo de las entrevistas a los profesores, es conocer sus opiniones con respecto al proceso de enseñanza y aprendizaje de la Teoría de Grafos, como también los recursos empleados, el tipo de motivación, la metodología y el ambiente al cual son sometidos los educandos.

4.3. ANÁLISIS DE PROBLEMAS EXISTENTES

El punto de partida del ciclo de vida de una proyecto es la etapa de análisis, en la cual se identifican los problemas o situaciones problemáticas existentes dentro del

ambiente educativo impuesto para el aprendizaje de la Teoría de Grafos, así mismo las posibles soluciones ó alternativas aplicables, reales que puedan generar mejores resultados en la labor educativa.

☰ Conclusiones de las consultas a los Estudiantes:

1. Existe verdaderamente una dificultad de aprendizaje y asimilación de la temática impartida dentro de la asignatura de Sistemas de Información.

La mayoría de los estudiantes están de acuerdo en que si tienen dificultad para asimilar los temas y sustentaron sus respuestas con los siguientes argumentos:

- Complejidad de los temas.
- Falta de práctica y ejercitación en clase.
- Avance rápido de los temas.
- Intensidad horaria escasa.
- Ejercicios sin concretar.
- Metodología inapropiada empleada por el docente.
- Falta dinámica en la clase.
- Existe un poco de temor hacia la temática.
- Simplemente porque no le interesa la materia.

Se concluye que los estudiantes tienen dificultades en el aprendizaje de los temas que se desarrollan en clase, las causas de las falencias no dependen exclusivamente del docente, sino también del alumno, teniendo en cuenta que cada uno de ellos es un ser independiente con su método ó estilo particular de abordar el aprendizaje y de ello depende mucho también el interés y la motivación que le presente la temática, es latente una inclinación hacia métodos distintos al tradicional.

2. Dentro de los aspectos que para los estudiantes son más relevantes dentro de las deficiencias de aprendizaje expuestas anteriormente se encuentran:

- Complejidad de los temas.
- Falta de práctica y ejercitación en clase.
- Intensidad horaria escasa.
- Metodología inapropiada empleada por el docente.
- Existe un poco de temor hacia la temática.

Como el conocimiento de las Estructuras de Datos implica un aprendizaje productivo es de suma importancia que el estudiante se ejercite y resuelva problemas que permitan aplicar soluciones por medio de aplicaciones de grafos.

Partiendo en que “la realidad solo existe en la medida en que se la construye” podemos afirmar que el estudiante debe ser el protagonista en el proceso de aprendizaje, poseedor de una carácter crítico y constructivo que lo conduzca a formar sus propias experiencias y por ende su propio conocimiento.

3. A pesar que los docentes están de acuerdo en que la clásica forma de enseñar debe evolucionar a la par con la tecnología y las facilidades que esta nos brinda, no se hace nada por incurrir en ese cambio y gran parte de la culpa es de los estudiantes, que aunque son los directamente afectados no exigen que sus docentes busquen nuevas alternativas y son conformes con los típicos métodos Profesor – tablero – estudiante.

El uso de computadores como herramientas de apoyo al aprendizaje no son una camisa de fuerza para los docentes, existen otras herramientas que serían de gran ayuda y que permitirían dinamizar las clases y que los estudiantes asimilaban de mejor forma la transmisión del conocimiento (videos, acetatos, diapositivas, etc), ayudas que de ser utilizadas traerían al educando experiencias vivenciales y productivas para la adquisición de nuevos conocimientos.

4. Los estudiantes plantearon también soluciones que a su criterio podrían mejorar la calidad de la Enseñanza y el aprendizaje, entre sus sugerencias destacamos:

- Aumentar las prácticas en el aula de informática.
- Aumentar la intensidad horaria semanal.
- Mejorar la metodología de enseñanza.
- Explicaciones detalladas y profundas por parte de los docentes.
- Realizar talleres en grupos.
- Incrementar el uso de recursos diferentes al tablero para las explicaciones.
- Promover la participación unilateral en el aula de clase.
- Realizar mayor número de ejercicios prácticos.

Es importante entender y atender las necesidades de los estudiantes porque ellos son el centro del proceso educativo, en ellos se refleja el resultado del proceso de formación y la calidad de la educación impartida, es importante también que el docente tome conciencia en que su labor debe estar encaminada en la búsqueda de nuevas alternativas. Una estrategia que el docente puede emplear es la búsqueda de recursos a partir de los niveles de aprovechamiento que el espera de los alumnos.

5. El computador como apoyo a la labor educativa: en su totalidad los estudiantes creen conveniente la creación y uso de una Software Educativo que les apoye el aprendizaje de las Estructuras de Información específicamente el área de la teoría de grafos (objeto de nuestro estudio).

Como conclusión podemos decir que el computador es una alternativa, que apoya el proceso de aprendizaje. El software tiene como ventaja lo siguiente:

- Es un medio interactivo de comunicación.
- Desarrolla comprensión, imaginación, creatividad y lógica.
- Va al ritmo con el nivel de aprendizaje de los alumnos.
- Desarrolla todos los sentidos.
- Despierta el interés del estudiante por aprender los temas.

Cabe aclarar que el computador no reemplaza al docente en su labor educativa. La formación moral, ética, social, humanitaria y de sensibilidad solo el profesor puede ofrecerla.

Una aspecto destacable, es que el aprendizaje programado exige mejor atención por parte del alumno, le permite un seguimiento personal de los temas a tratar en su aprendizaje, también evalúa en forma individual y correcta a los alumnos para brindar refuerzos personalizados

☰ Conclusiones de las consultas a los Profesores:

1. Motivación del aprendizaje: la materia de estructuras de información implica el desarrollo productivo de ideas y destrezas por parte de los estudiantes, por lo tanto el docente debe proporcionar un ambiente participativo y dinámico, de esta manera el alumno se sentirá motivado y auto dirigido a crear soluciones a los problemas de las actividades den clase.

Para lograr una dinámica en grupo el docente debe buscar estrategias como: trabajos en grupo, actividades en el aula de informática, consultas, guías, uso de videos, acetatos, ejemplos de la vida real, etc.

Para la motivación de las clases los docentes utilizan como estrategia lo siguiente: ejemplos de la vida real donde se apliquen los conceptos expuestos, variedad de ejemplos, trabajos en grupo, recursos audiovisuales, recompensas ó estímulos por logros adquiridos.

No se podría globalizar el desempeño del profesorado, ya que cada uno utiliza un estilo de enseñanza diferente. La tendencia de los docentes en el proceso de aprendizaje es de hacer prácticas sus clases, permitiéndole al alumno desarrollar y construir nuevos modelos de pensamiento agilizando sus

procesos mentales de abstracción y asimilación del conocimiento en el estudio de las Estructuras de información.

2. En cuanto al software utilizado: los docentes desconocen ó no utilizan herramientas computacionales que les apoyen el proceso de enseñanza – aprendizaje, por lo tanto estiman la incursión de un Software Educativo para este fin.
3. Uso de recursos educativos: la mayoría de los docentes aún tiende a utilizar como único recurso educativo en la exposición de sus clases el tablero, evitando el uso de otros medios. Teniendo en cuenta que cada alumno tiene diferentes estilos de aprendizaje, algunos aprenden rápido y más fácilmente cuando utilizan sus sentidos (viendo, oyendo); otras cuando tienen la posibilidad razonar (elaborando una abstracción de una experiencia concreta o construyendo un concepto); memorizar algo o razonar lógica y creativamente. El empleo de diferentes herramientas ó recursos educativos en el aprendizaje de la teoría de grafos repercutirá en el grado de asimilación de los temas ó conceptos que se estén tratando.

Se puede concluir que a través de un aprendizaje programado se puede llegar al conocimiento, que dependiendo del buen uso que se le otorgue es posible alcanzar los objetivos educativos.

Como una ventaja de este proceso de aprendizaje es que el estudiante trabaja a su propio ritmo de estudio lo que permite un aprendizaje personalizado.

4.4. DISEÑO DEL MEC

El diseño esta en función directa de los resultados del análisis de necesidades educativas detectadas en la población educativa. La orientación y el contenido del MEC se derivan de la necesidad sentida de la población objeto que justifica la creación del sistema SEGraD.

4.4.1. Selección de los Temas que Contiene SEGraD

Para la selección de temas que debe contener el software se tubo en cuenta los datos anotados por las fuentes de información, realizadas a docentes y estudiantes y de acuerdo al contenido del plan curricular de la Universidad de Nariño, facultad de ingeniería, programa Ingeniería de Sistemas en el área de las Estructuras de Información, más específicamente en el área de la Teoría de Grafos.

Se organizó un programa con ayuda del docente encargado de la asignatura, el cual integra conceptos específicos y aplicaciones que definen la temática de una forma clara y completa.

La temática esta especificada en la delimitación del proyecto

Análisis de Tareas de Aprendizaje

Se requiere descomponer el objetivo terminal del sistema SEGrAD en tareas de aprendizaje subyacentes, se detallara hasta el punto en que se hallen todas las habilidades, conocimientos y destrezas que es necesario adquirir, como complemento a las que ya se traen para lograr el objetivo propuesto.

Se recurrió al planteamiento de “Abajo hacia Arriba”, partiendo de los sub objetivos que ya dominan los alumnos.

Se debe preguntar en cada nivel ¿Qué debo aprender conducente al objetivo terminal con lo que sé hasta el momento?; la respuesta indicara al menos un objetivo del amas alto nivel. Al repetir la pregunta de los sub objetivos se llega finalmente al objetivo terminal.

ANÁLISIS DE TAREAS DE APRENDIZAJE
OBJETIVO TERMINAL
6. PRACTICA Y EVALUACION Aprendizaje en la temática sobre la Teoría de Grafos, por medio de las prácticas y las evaluaciones.
OBJETIVOS SUBYACENTES
4. APLICACIONES DE GRAFOS Uso de los grafos, áreas donde pueden ser aplicados
4. RECORRIDOS DE GRAFOS Comprender el acceso a datos de interés almacenados en un grafo por medio de los recorridos.
3. ALMACENAMIENTO DE UN GRAFO EN MEMORIA Entender el como y porque del manejo de la memoria del computador durante el almacenamiento de una estructura de datos.
2. CONCEPTOS BÁSICOS Saber que es un Grafo, los tipos, para que sirven y los elementos que los conforman.
CONDUCTA DE ENTRADA
1. PRERREQUISITOS Conocimientos básicos sobre estructuras de información (pilas, listas, colas, árboles, montículos).

Cuadro 8.

ANÁLISIS DE TAREAS DE APRENDIZAJE

4.4.3. Diseño Comunicacional

La zona directamente involucrada en la comunicación e interacción entre hombre y el programa se denomina interfaz. En ella intervienen todos los mensajes que el usuario y el programa reciben durante la ejecución del software (sonoros, pictóricos, iónicos, etc.), también los dispositivos de entrada y salida de datos que están disponibles para el intercambio de mensajes (teclado, ratón, pantalla, impresora, parlantes, etc.), así como las zonas de comunicación habilitadas en cada una de las áreas de trabajo y los mismos dispositivos periféricos ya mencionados.

“El sistema de intercomunicación depende en gran medida de lo que se desea que el usuario aprenda y de las características socio – culturales y generacionales de este.”

Las interfaces deben ser diseñadas teniendo en cuenta la edad y las condiciones de desarrollo socio – cultural de los aprendices, la terminología y simbología actuales para adecuarlos a los mensajes que van a lograr un mayor entendimiento y que el usuario se sienta a gusto.

El manejo de los dispositivos de entrada y salida juega un papel importante dentro del diseño de un MEC, cada día hay más y mejores periféricos con los cuales

podemos facilitar la comunicación entre el usuario y el programa, es importante tener criterio para seleccionarlos, sacando el máximo provecho de cada uno de ellos, eso si, evitando que el programa dependa lo menos posible de su funcionamiento.

SEGraD maneja los periféricos de entrada y salida exigidos en cualquier equipo convencional:

Periféricos de entrada:

- ❑ *El teclado*: este dispositivo será de gran ayuda en el momento del registro del usuario. El tutor ó docente encargado tiene prioridad sobre el uso de este periférico por sus actividades de manejo de las bases de datos en donde la mayoría de los registros se hacen vía teclado. El aprendiz hará caso omiso del teclado mientras cuente con el mouse para hacer uso de los controles de trabajo del software, sino el teclado puede dar acceso a dichos controles por medio de combinaciones de teclas preestablecidas.
- ❑ *El Mouse (ratón)*: requerido en todo el transcurso de la aplicación, tanto para el aprendiz como para el tutor, su importancia radica en que permite el control directo de los controles de las zonas de trabajo de SEGraD. Su funcionamiento puede ser sustituido por el teclado.

🖨️ Periféricos de salida:

- ❑ *Monitor*: dispositivo indispensable para visualizar el contenido de la aplicación.
- ❑ *Impresora*: necesaria para imprimir los reportes y soportes para estudiantes y docentes.
- ❑ *Parlantes*: (opcional), da al usuario una mayor motivación y permite estar atento a los mensajes y procesos del software.

4.4.3.1. Definición de las Zonas de Comunicación

La definición de las zonas de comunicación, va de la mano con la selección de los dispositivos de entrada y salida seleccionados, lo que hará posible que el programa y el computador se entiendan. Normalmente, en la interfaz Hombre – máquina hay los siguientes tipos de zonas de comunicación, en cuya puesta en marcha se combinan dispositivos de salida (pantalla) y entrada (teclado, ratón, etc.):

- *Zonas de trabajo*:
 1. Región donde se ubican los apartes que le sirven para aprender (teoría, ejercicios, modelos).

2. Donde se lleva a cabo operaciones sobre el objeto de estudio (responder preguntas, alterar el estado de una variable).
3. Donde aprecia el efecto de las decisiones que toma (retroinformación, pistas, refuerzos)

La zona de trabajo puede tener subzonas, si el diseño lo requiere.

- *Zonas de control del programa:* donde será posible alterar el flujo y el ritmo del programa, suele estar asociado con los menús del programa, también está asociado con las posibilidades de reinicio o abandono del programa.

4.4.3.2. Elementos Constitutivos de las Zonas de Comunicación

4.4.3.2.1. Los Menús

SEGraD, manejan dos sistemas de menús (Textual y gráfico) que le permite a los usuarios tener acceso a todas las opciones del software. Al comienzo de la aplicación (Inicio de Sesión) se presenta una combinación de menú textual y gráfico lo que hace más eficiente el ingreso y permite el uso de ratón o el teclado según se crea conveniente.

Menú Inicio de Sesión: (textual y grafico)

Usuarios:

- Usuario Nuevo
- Usuario Antiguo
- Tour por SEGraD

Ejercitarme:

- Almacenamiento de grafos en memoria
- Grafos Eulerianos y Hamiltoniano
- Recorridos de grafos ▶
 - Recorrido de un Grafo a lo Ancho
 - Recorrido de un Grafo en Profundidad
- Aplicaciones ▶
 - Algoritmo de Warshall
 - Algoritmo de Prim
 - Algoritmo de Kruskal
 - Algoritmo de Dijkstra
 - Algoritmo de Floyd

GraDMaker:

- Herramienta de soluciones

Ayuda:

- Temas de ayuda F1
- Índice
- Acerca de

Al segundo menú, se tiene acceso siempre y cuando se este autorizado, se debe registrar el usuario antes de poder ingresar al entorno de trabajo de SEGraD.

Menú entorno de trabajo de SEGraD: (textual)

Archivo: (se activa solo para las formas de ejercitación)

- Nuevo
- Abrir
- Guardar
- Guardar como...
- Salir

Administrador: (uso exclusivo del “docente”)

- Ingreso del administrador

- Agregar grafos a la base de datos
- Actualizar registros
- Borrar registros de la base de datos
- Reportes generales ▶
 - Usuarios registrados
- Salir
- Grafos:
 - Principios Básicos
 - Tipos de grafos ▶
 - Grafos Fuerte y Débilmente Conectados
 - Grafos Eulerianos y Hamiltonianos
 - Otros tipos de grafos
 - Almacenamiento de un grafo en memoria
 - Recorridos de grafos ▶
 - Recorrido de un grafo a lo Ancho
 - Recorrido de un grafo en Profundidad
- Aplicaciones:
 - Algoritmo de Warshall
 - Algoritmo de Prim
 - Algoritmo de Kruskal
 - Algoritmo de Dijkstra
 - Algoritmo de Floyd
- Evaluaciones:
 - Elaborar evaluación ▶ (uso exclusivo del “docente”)
 - Conceptos Básicos
 - Tipos de Grafos
 - Almacenamiento de un grafo en memoria
 - Recorridos de un grafo
 - Algoritmo de Warshall
 - Algoritmo de Prim
 - Algoritmo de Kruskal
 - Algoritmo de Dijkstra
 - Algoritmo de Floyd
 - Realizar evaluación ▶
 - Habilitar evaluación dirigida
 - Dirigida(docente) ▶
 - Conceptos Básicos
 - Tipos de Grafos
 - Recorrido de un grafo a lo Ancho
 - Recorrido de un grafo en Profundidad
 - Algoritmo de Warshall
 - Algoritmo de Prim
 - Algoritmo de Dijkstra

- Algoritmo de Floyd
- Auto evaluación ▶
 - Conceptos Básicos
 - Tipos de Grafos
 - Recorrido de un grafo a lo Ancho
 - Recorrido de un grafo en Profundidad
 - Algoritmo de Warshall
 - Algoritmo de Prim
 - Algoritmo de Dijkstra
 - Algoritmo de Floyd
- Mostrar resultados ▶ (uso exclusivo del “docente”)
 - Consultas de avance
 - Reportes de evaluaciones ▶
 - Formativos
 - Sumativos
 - Autoevaluaciones
- Sistema: (uso exclusivo del “docente”)
 - Seguridad ▶
 - Cambio de claves
- Ventana:
 - Actualizar
 - Cascada
 - Mosaico horizontal
 - Mosaico vertical
- Ayuda:
 - Temas de ayuda F1
 - Índice
 - Acerca de

Acerca del Texto

SEGraD maneja un tipo de texto estandarizado sobre todas las áreas de trabajo, aquí presentamos una relación de los textos utilizados y las características que los distinguen:

TEXTOS MANEJADOS EN SEGRAD			
TEXTO	TIPO LETRA	TAMAÑO	UBICACIÓN
Títulos	Arial Alternative	18, 16	Superior central
Conceptos	MS Sans Serif	10	Debajo de los títulos
Explicaciones Resaltados Hipervínculos Controles	MS Sans Serif	8	Zona inferior izquierda, todos los botones del área de trabajo.
Pesos arcos	Franklin Gothic Demi Cond	7	Formas de ejercitación.
Subtítulos	Monotype Corsiva	16	Forma Otros tipos de grafos
Título	Forte	36	Acerca de
Números	Swis721 Ex BT	11	Tablas Formas de Kruskal
Símbolos matemáticos	Symath	16	Símbolos igual (=) y diferente (\neq)
Símbolos	Wingdings 3	18	Flechas

Cuadro 9.

TEXTOS MANEJADOS EN SEGRAD

Los textos fueron seleccionados teniendo en cuenta parámetros como:

- Densidad
- Legibilidad
- Distancia del usuario con la máquina
- Zona de trabajo a la que pertenecen

Interfaces de SEGraD:

□ Inicio de Sesión:



Zonas de control: Menú textual superior y gráfico interno

Figura 17.

INTERFAZ INICIO DE SESIÓN

SEGraD posee una forma de Inicio de Sesión, que es la primera en aparecer, donde se presenta a los usuario una combinación de menús textual y gráfico que permiten tener acceso a los diferentes entornos de trabajo de SEGraD. Las opciones a las que el usuario tiene acceso son: Registro, Ingreso, ayuda, GradMaker (herramienta de solución de grafos), Free Tour (vistazo al software) y salida.

□ Entorno de trabajo principal:



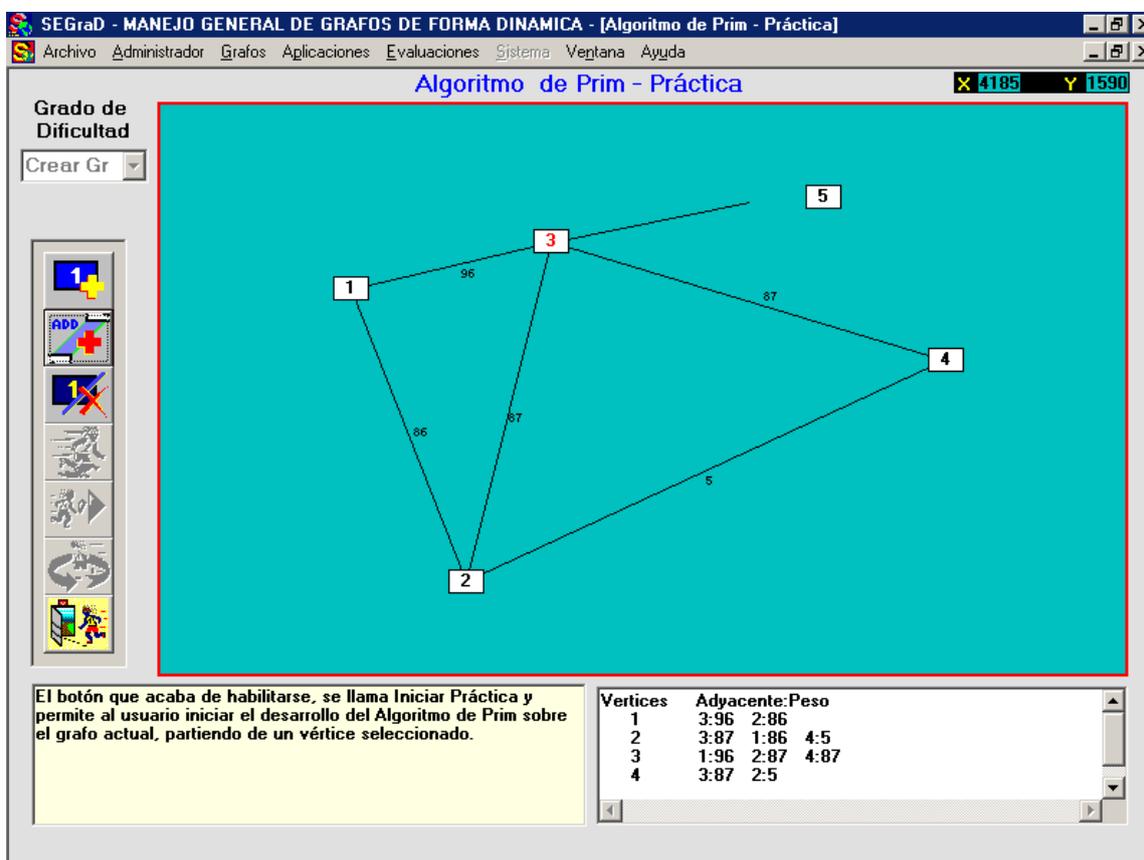
Zonas de control: Menú Principal Superior

Figura 18.

ENTORNO DE TRABAJO PRINCIPAL

El Entorno de Trabajo Principal presenta al usuario acceso a los elementos constitutivos de las zonas de comunicación (menú) y las áreas de trabajo cuando se ha seleccionado algunos de los apartes correspondientes dentro de las cuales a su vez están definidas las áreas de control.

□ Entorno de Práctica y ejercitación de SEGraD:



Zonas de control: Menú superior - Archivo, Barra de botones izquierda
Zonas de Trabajo: Area central - recuadro verde, cuadros inferiores - explicativos

figura 19.

ENTORNO DE PRÁCTICA Y EJERCITACIÓN DE SEGRAD

El entorno de Práctica y Ejercitación de SEGraD es en un gran porcentaje el responsable del aprendizaje de las temáticas expuestas en el módulo de apoyo al aprendizaje, se observan bien definidas sus áreas de trabajo y control y permite

que el usuario ponga en práctica los conceptos teóricos de una forma interactiva, clara y creativa.

4.4.3.3. Apoyo Grafico

Los gráficos que el MEC involucre dentro de el proceso de comunicación hombre maquina pueden ser de diferente índole, ya sean diagramas, dibujos y esquemas, animaciones, simulaciones y gráficos de tratamiento numérico.

SEGraD maneja como recurso principal animaciones y simulaciones de apoyo gráfico en la mayoría de sus entornos, las cuales sirven para mostrar, ensayar, destacar objetos y motivar a los usuarios dentro del proceso de apoyo al aprendizaje propuesto.

Las simulaciones y animaciones están controladas por medio de hipervínculos y botones a los que el usuario tiene libre acceso en el momento que crea conveniente y durante el avance del aprendizaje.

Simulación Algoritmo de Prim

ALGORITMO DE PRIM

El Algoritmo de Prim resuelve el problema de escoger la mejor alternativa entre las muchas que se puedan presentar y como resultado se obtendrá un conjunto de arcos de manera que conecten todos los vértices y además que la suma de sus valores (peso) sea mínima, es decir, obtener el mínimo árbol expandido.

Grafo ejemplo

Mínimo Arbol Expandido

Reiniciar Continuar Simulación Vértice No Visitado Vértice Visitado

Nuevamente definimos en cuál de los vértices no visitados inciden 2 arcos. En este caso es el **vértice 7** sobre el cual inciden los arcos que llegan de los **vértices 4 y 5**, de los cuales eliminaremos el arco procedente del **vértice 4 por ser el de mayor peso**. Dejando así todos los vértices no visitados con un sólo arco incidente.

Ejercitar Salir

Zonas de control: Menu superior Principal, botones: Reiniciar, continuar simulación Ejercitar, Control de velocidad de simulación, salir e hipervinculos.
Zonas de trabajo: Area central de representación simulada del algoritmo.

Figura 20.

APOYO GRAFICO (Simulación Algoritmo de Prim)

Diseño Computacional

Dentro de este aparte se establece cual será la estructura lógica que permita que el MEC cumpla con las funciones requeridas. La estructura lógica es la base para formular el programa principal y cada uno de los procedimientos que requiere el MEC.

Íntimamente ligadas a la estructura lógica están las estructuras de datos que se requieren para que el MEC cumpla eficientemente con su cometido.

4.4.4.1. Estructura Lógica del MEC

La estructura lógica expresa los procedimientos que el programa debe tener y sus interrelaciones, de modo que cumpla con las funciones definidas para cada uno de sus usuarios y que permita al aprendiz recorrer la estructura de aprendizaje que subyace a los objetivos buscados.

SEGraD basa su funcionamiento en 4 módulos de trabajo, donde los usuarios, tanto el facilitador como el estudiante, tienen sus funciones predeterminadas.

La estructura lógica del MEC representa la solución al problema y al ambiente de aprendizaje deseado.

- ☞ *Módulo Administrativo*: para uso exclusivo del facilitador, docente, profesor, administrador ó como quiera llamarse, donde se muestran todas las opciones de mantenimiento la base de datos, edición de los ejemplos para las prácticas y las evaluaciones.

- ☞ *Módulo de apoyo al aprendizaje*: (facilitador y estudiante), área general de trabajo donde se presentan las temáticas expositivas de forma teórica y apoyadas por una simulación, donde están definidas las zonas de control y de trabajo del software.

- ☞ *Módulo de práctica y afianzamiento*: (facilitador y estudiante), en el cual el usuario tiene la oportunidad de aplicar los conocimientos adquiridos en el módulo de apoyo al aprendizaje para así darse cuenta si verdaderamente la transmisión de conocimientos tubo éxito.

- ☞ *Módulo de evaluación*: (facilitador y estudiante), con funciones explícitas dependiendo del usuario, si es el estudiante, le permite realizar la evaluación correspondiente al tema actual de carácter no sumativo ó definitivo para usuarios que creen haber asimilado bien los conceptos, para el facilitador se habilitan por medio del módulo administrativo opciones de edición de los exámenes, ejercicios, tipos de evaluación y resultados

obtenidos. Las evaluaciones pueden ser generadas por SEGraD de forma aleatoria ó creadas por el facilitador personalizándolas; pueden ser de todo el contenido del software ó simplemente de una temática específica.

4.4.4.1.1. Diagramas de Procesos (Análisis de Flujos de Datos – Ver Anexo 3)

4.4.4.1.2. Diccionario de Datos 1 (Ver Anexo 4)

4.4.4.2. Estructura de Datos

Corresponde a la forma como se distribuyen en el computador toda la información que se maneja en el MEC para que pueda ser utilizada a medida que se vaya necesitando. No se puede disponer simultáneamente de toda la información del MEC en memoria principal, la cual casi siempre es escasa, por lo que se deben crear archivos en memoria secundaria desde los que se pueda recuperar ó almacenar la información.

SEGraD, para todo el manejo y almacenamiento de todos los datos generados trabaja una base de datos creada en Microsoft Acces 2000, donde la información completa esta registrada en 75 tablas de la siguiente manera:

4.4.4.2.1. Tablas – Base de Datos : Db_Grafos

- arcos
- vértices
- datos_generales
- e_arcos
- e_vértices
- f_Arcos
- f_vértices
- h_Arcos
- h_vértices
- k_Arcos
- k_vértices
- w_arcos
- w_vértices
- pb_catcol
- pb_catdt

- pb_catfmt
- pb_cattbl
- pb_catvld
- preguntas_almacenamiento
- preguntas_Dijkstra
- preguntas_Floyd
- preguntas_kruskal
- preguntas_prim
- preguntas_rancho
- preguntas_rprofundo
- preguntas_Warshall
- respuestas_almacenamiento
- respuestas_Dijkstra
- respuestas_Floyd
- respuestas_kruskal
- respuestas_prim
- respuestas_rancho
- respuestas_rprofundo
- respuestas_Warshall
- evaluación_pribas1
- evaluar_prinbasa
- evaluar_prinbasp

- evaluar_prinbasr
- evaluar_prinbasv

4.4.4.2.2. Diccionario de Datos 2

Tablas generales:

☐ TABLA vertices :

CAMPO	DESCRIPCIÓN	TIPO	TAMAÑO(c/res)
Nivel	Nivel del grafo al que corresponde	Numérico	Entero
Grafo	Nombre numérico del grafo al que corresponde	Numérico	Entero
vertice	Número del vértice	Numérico	Entero
left	Posición sobre el eje X	Numérico	Entero
top	Posición sobre el eje Y	Numérico	Entero

☐ TABLA arcos :

CAMPO	DESCRIPCIÓN	TIPO	TAMAÑO(c/res)
Nivel	Nivel del grafo al que corresponde	Numérico	Entero
Grafo	Nombre numérico del grafo al que corresponde.	Numérico	Entero
Arco	Número del arco.	Numérico	Entero
X1	Posición de la punta1 del arco con respecto al eje X.	Numérico	Entero
X2	Posición de la punta2 del arco con respecto al eje X.	Numérico	Entero
Y1	Posición de la punta1 del arco con respecto al eje Y.	Numérico	Entero
Y2	Posición de la punta2 del arco con respecto al eje Y.	Numérico	Entero
Vert_pta1	Vértice1 sobre el cual se ubica.	Numérico	Entero
Vert_pta2	Vértice2 sobre el cual se ubica.	Numérico	Entero
Pta1	Punta 1 lleva o no sentido	Numérico	Entero
Pta2	Punta 2 lleva o no sentido	Numérico	Entero
Peso	Valor del arco.	Numérico	Entero

☐ TABLA datos_generales:

CAMPO	DESCRIPCIÓN	TIPO	TAMAÑO(c/res)
Codigo	Id. Del usuario registrado en la base de datos.	Texto	13
Nombres	Nombres completos del usuario	Texto	50
apellidos	Apellidos completos del usuario	texto	50
Teléfono	Teléfono de usuario (si lo tiene)	Texto	10
Dirección	Lugar donde reside el usuario.	Texto	50
e-mail	Correo electrónico del usuario (si la tiene).	Texto	50
Programa	Programa en el que el usuario esta matriculado.	Texto	50
Semestre	Semestre actual que cursa.	Texto	2
Clave	Clave de acceso a SEGraD	Texto	10
Recordatorio	Palabra o frase que haga referencia a la clave.	texto	50

Dentro del diseño y las estructuras definidas en las tablas nombradas a continuación los campos son similares, por lo tanto simplemente las agruparemos y describiremos una de ellas.

Las tablas:

- e_vértices
- f_vértices
- h_vértices
- k_vértices
- w_vértices

☐ TABLA X_vertices :

CAMPO	DESCRIPCIÓN	TIPO	TAMAÑO(c/res)
Nivel	Nivel del grafo al que corresponde	Numérico	Entero
Grafo	Nombre numérico del grafo al que corresponde	Numérico	Entero
vertice	Número del vértice	Numérico	Entero
left	Posición sobre el eje X	Numérico	Entero
top	Posición sobre el eje Y	Numérico	Entero

Las tablas:

- e_arcos
- f_Arcos
- h_Arcos
- k_Arcos
- w_arcos

☐ TABLA X_arcos :

CAMPO	DESCRIPCIÓN	TIPO	TAMAÑO(c/res)
Nivel	Nivel del grafo al que corresponde	Numérico	Entero
Grafo	Nombre numérico del grafo al que corresponde.	Numérico	Entero
Arco	Número del arco.	Numérico	Entero
X1	Posición de la punta1 del arco con respecto al eje X.	Numérico	Entero
X2	Posición de la punta2 del arco con respecto al eje X.	Numérico	Entero
Y1	Posición de la punta1 del arco con respecto al eje Y.	Numérico	Entero
Y2	Posición de la punta2 del arco con respecto al eje Y.	Numérico	Entero
Vert_pta1	Vértice1 sobre el cual se ubica.	Numérico	Entero
Vert_pta2	Vértice2 sobre el cual se ubica.	Numérico	Entero
Pta1	Punta 1 lleva o no sentido	Numérico	Entero
Pta2	Punta 2 lleva o no sentido	Numérico	Entero
Peso	Valor del arco.	Numérico	Entero

Las tablas:

- preguntas_almacenamiento
- preguntas_Dijkstra
- preguntas_Floyd
- preguntas_kruskal
- preguntas_prim

- preguntas_rancho
- preguntas_rprofundo
- preguntas_Warshall

☐ TABLA preguntas_XXX:

CAMPO	DESCRIPCIÓN	TIPO	TAMAÑO(c/res)
Cod_PRE	Código de la pregunta	Numérico	Entero
Pregunta	Pregunta del tema específico	Memo	
Respuesta	Respuesta correcta	memo	

Las tablas:

- respuestas_almacenamiento
- respuestas_Dijkstra
- respuestas_Floyd
- respuestas_kruskal
- respuestas_prim
- respuestas_rancho
- respuestas_rprofundo
- respuestas_Warshall

☐ TABLA respuestas_XXX:

CAMPO	DESCRIPCIÓN	TIPO	TAMAÑO(c/res)
Cod_pre	Código de la pregunta	Numérico	Entero
Cod_res	Pregunta del tema específico	Numérico	Entero
Respuesta	Respuesta correcta	memo	

☐ TABLA evaluación_pribas1

CAMPO	DESCRIPCIÓN	TIPO	TAMAÑO(c/res)
Cod_eva	Código de evaluación	Numérico	Entero
Cod_gra	Código de grafo	Numérico	Entero
Cod_pre	Código de pregunta	Numérico	
Cod_opc1	Código respuesta incorrecta 1	Numérico	Entero
Cod_opc2	Código respuesta incorrecta 2	Numérico	Entero
Cod_opc3	Código respuesta incorrecta 3	Numérico	

CAMPO	DESCRIPCIÓN	TIPO	TAMAÑO(c/res)
Grafo	Nombre numérico del grafo al que corresponde.	Numérico	Entero
Arco	Número del arco.	Numérico	Entero
X1	Posición de la punta1 del arco con respecto al eje X.	Numérico	Entero
X2	Posición de la punta2 del arco con respecto al eje X.	Numérico	Entero
Y1	Posición de la punta1 del arco con respecto al eje Y.	Numérico	Entero
Y2	Posición de la punta2 del arco con respecto al eje Y.	Numérico	Entero
Vert_pta1	Vértice1 sobre el cual se ubica.	Numérico	Entero
Vert_pta2	Vértice2 sobre el cual se ubica.	Numérico	Entero
Pta1	Punta 1 lleva o no sentido	Numérico	Entero
Pta2	Punta 2 lleva o no sentido	Numérico	Entero
Peso	Valor del arco.	Numérico	Entero

▣ TABLA evaluación_prinbasp

CAMPO	DESCRIPCIÓN	TIPO	TAMAÑO(c/res)
Cod_gra	Código de grafo	Numérico	Entero
Cod_pre	Código de pregunta	Numérico	Entero
Pregunta	Pregunta	Memo	
Respuesta	Respuesta correcta	Memo	
Cod_eva	Evaluación a la que pertenece	Numérico	Entero

☐ TABLA evaluación_prinbasr

CAMPO	DESCRIPCIÓN	TIPO	TAMAÑO(c/res)
Cod_gra	Código de grafo	Numérico	Entero
Cod_pre	Código de pregunta	Numérico	Entero
Cod_res	Código de respuesta	Numérico	Entero
Respuesta	Respuesta correcta	Memo	
Cod_eva	Evaluación a la que pertenece	Numérico	Entero

☐ TABLA evaluación_prinbasv

CAMPO	DESCRIPCIÓN	TIPO	TAMAÑO(c/res)
Grafo	Código del grafo	Numérico	Entero
Vértice	Número de vértice	Numérico	Entero
Left	Posición eje X	Numérico	Entero
top	Posición eje Y	Numérico	Entero

4.4.4.2.3. Diagramas Relacionales en Db_Grafos (ver Anexo 5)

CONCLUSIONES

- No existe ninguna metodología ni herramienta que sean adecuadas en todas las circunstancias. A veces la metodología o la misma herramienta contribuyen a crear nuevos problemas o a revelar otros que no hayan sido previstos.
- La aplicación de cualquier MEC debe obedecer a un plan establecido por el docente, donde se trata de conjugar los diferentes elementos que participan el proceso de aprendizaje.
- El desarrollo de un MEC debe responder a las necesidades de un problema previamente identificado, que amerite verdaderamente el desarrollo de una herramienta específica de apoyo.
- El Software “SEGraD” es una herramienta facilitadora en el proceso de aprendizaje de la Teoría de Grafos tanto para el estudiante, que desea aprender, como para el docente.

- ▣ SEGraD no busca reemplazar la acción de otros medios educativos convencionales, más bien el su objetivo es el de complementarlos y buscar que el estudiante se identifique y asimile a satisfacción los temas de estudio, además de llevar un control o seguimiento en el avance del usuario sobre la temática manejada de forma progresiva.

- ▣ El software tiene características de un sistema experto, “sin llegar a serlo”, que le permiten comprobar la situación actual del usuario durante la ejecución y resolución de un problema o práctica de grafos mediante la validación de todas y cada uno de las posibles eventos que puedan llegarse a presentar, en general permite y ayuda al estudiante a detectar y corregir errores de diseño y resolución mientras diseña o ejecuta un algoritmo definido.

- ▣ “Unas buenas herramientas no hacen un buen profesor, pero un excelente profesor si emplea bien las herramientas” – (Eleanor L.) es necesario entender que aunque el estudiante puede por iniciativa propia crear un plan de estudio que le va permitir avanzar sobre la temática, es imprescindible un facilitador, tutor, docente o como quiera llamarse para que clarifique conceptos y resuelva dudas de funcionamiento o sobre la temática trabajada.

RECOMENDACIONES

- Esperamos que SEGraD sea tomado como piedra angular o modelo sobre el cual se cimentara una corriente de Software Educativo orientados a satisfacer las necesidades que presente la educación superior, no solo en nuestra universidad, sino en todas las instituciones educativas que tengan acceso a este material.

- Conformar un grupo de investigación que continúe con el estudio de SEGraD, con el propósito de expandir y complementar la aplicación del software en el proceso de aprendizaje de la materia de Estructuras de datos.

- Garantizar que el software “SEGraD ” será puesto a disposición del docente y estudiantes interesados en el proceso de aprendizaje de Estructuras de información, más específicamente el área de la Teoría de Grafos, y no será archivado como sucede en la mayoría de los casos.

- ☐ Motivar a la población estudiantil para que aumente su capacidad investigativa y genere proyectos que abran paso al progreso, no solo tecnológico sino también educativo de la universidad.

BIBLIOGRAFÍA

PRESSMAN, Roger S. Ingeniería de software un enfoque práctico.
Editorial McGraw Hill, 1998.

BECERRA, C. Santamaría. Estructuras de datos en C++.
Editorial Kimpres, 1997.

ARRAUT, Luis H. Ciencia, Tecnología y Educación.
Universidad de Cartagena. Tomo 7.

SENN, A. James. Análisis y diseño de sistemas de información, segunda edición,
México.
Editorial McGraw Hill, 1992.

TENEMBAUM, N. Aarón, LANGSAM, Yedidyan y AUGENSTEIN, J. Moshe J.
Estructuras de datos con C y C++.
Editorial Prentice – Hall Hispanoamericana, 1991.

KENDALL, Kenneth E. Y KENDALL, Julie E. Análisis y Diseño de Sistemas,
México.

Editorial Prentice – Hall Hispanoamericana, 1991.

HEILEMAN, L. Gregorio L.

Estructuras de datos, algoritmos y programación orientada a objetos.

GALVIS, Panqueva Alvaro. Ingeniería de software Educativo, Universidad de Los
Andes, 1992.

BURKE, Robert. Enseñanza Asistida por Ordenador.

Madrid. Ed. Paraninfo, 1986.

GALVIS Panqueva Alvaro, Metodología Para el desarrollo de Material Educativo
Computarizado (MEC), Bogotá, SENA, Bloque Modular de Informática Educativa,
Doc. # 2,1986.

MARK Allen Weiss, Estructuras de datos y algoritmos.

Editorial Addison Wesley Iberoamericana.

RUMBAUGH Blaha, PREMIERLANY Eddy Lorensen, Modelado y diseño orientado
a objetos.

Editorial Prentice Hall, 1991.

Aho, Hopcroft, Ullman : Algoritmos y estructuras de datos.

Editorial Addison-Wesley, 1988

McKINNEY, Bruce. Programación avanzada en Visual Basic.

Madrid: MC Graw Hill, 1996.

http://www.doe.ds.vb.es/te/ani96/marques_software.

ANEXOS

Anexo 1

ENCUESTA ESTRUCTURAS DINAMICAS DE DATOS LINEALES

Marque con una **X** donde corresponda

1. De las siguientes herramientas educativas cual cree que es la mas importante?

- | | |
|---|--------------------------------------|
| <input type="checkbox"/> a) Radio | <input type="checkbox"/> e) Internet |
| <input type="checkbox"/> b) Textos | <input type="checkbox"/> f) Todos |
| <input type="checkbox"/> c) Herramientas informáticas | <input type="checkbox"/> g) Otros? |
| <input type="checkbox"/> d) Televisión | Cuales? _____ |
| | _____. |

2. Actualmente, ¿Encuentra accesibles y de forma oportuna las anteriores herramientas para su consulta?

Si _____ No _____

3. Encuentra falencias de aprendizaje con la clásica metodología de enseñanza profesor – tablero – alumno?

No _____
Si _____
¿Cuales? _____

_____.

4. ¿Considera usted que la tecnología informática juega un papel primordial en el cambio de la metodología educativa?

No _____
Si _____
¿Como? _____

_____.

5. ¿Considera que los temas de Estructuras de Datos Lineales es un área fácil de manejar?

Si _____ No _____

6. ¿Cree usted que el apoyo por medio de Material Educativo Computarizado, marcaría alguna diferencia en el tratamiento de los temas de Teoría de Grafos?

Si _____ No_____

7. ¿Conoce alguna Herramienta Informática (Software Educativo), que permita adquirir un conocimiento claro y preciso de los temas de Teoría de Grafos?

No_____

Si_____¿Cual?_____

8. ¿Está de acuerdo en que el tiempo disponible para el tratamiento de estos temas es corto y a veces quedan temas sin profundizar?

Si _____ No_____

9. Si se implementara un Software Educativo tipo expositivo, de ejercitación - práctica y que además lo evalúe verificando su grado de aprendizaje de manera llamativa e interactiva sobre la Teoría de Grafos, ¿Ud. lo adoptaría?

Si _____ No_____

10. Que características primordiales cree usted que debe tener un Software Educativo?

a) Exposición Grafica _____

b) Exposición Lineal _____

c) Simulaciones _____

d) Ejercitación _____

e) Evaluación _____

f) Acceso a Internet _____

g)Otras? ¿Cuales?_____

ENCUESTA TEORIA DE GRAFOS

Marque con una X donde corresponda - - Luego de conocido el software SEGraD

METODOLOGÍA:

1. ¿Cree usted que el enfoque de tipo “Algorítmico - Heurístico” empleado en SEGraD, da buenos resultados en el proceso del aprendizaje?

Si ____ No____

2. ¿Logró usted comprender el manejo de grafos de forma dinámica empleado por el software SEGraD?

Si ____ No____

3. ¿Considera usted que las simulaciones están bien enfocadas y que cumplen con el objetivo de crear una idea del algoritmo que representa cada tema propuesto?

Si ____ No____

4. ¿Después de utilizar SEGraD, se siente capaz de realizar un algoritmo para alguno de los temas propuestos?

Si ____ No____

5. ¿Emplearía algún otro método además del propuesto por SEGraD, para realizar el correcto aprendizaje de los temas propuestos?

No____

Si ____ Cual? _____

_____.

DISEÑO: Emita su concepto.

1. En cuanto al menú

___ Bien, ___ Regular, ___ Mal

Recomendaciones _____

_____.

2. En cuanto a los colores manejados

___ Bien, ___ Regular, ___ Mal

Recomendaciones _____

_____.

3. En cuanto al módulo de practica

___ Bien, ___ Regular, ___ Mal

Recomendaciones _____

_____.

4. ¿Qué aspectos del software, según su criterio, necesitan mejorar?

_____.

5. ¿Qué le falta al software, que usted considere conveniente agregarle?

_____.

Anexo 3

Ver archivo “ANÁLISIS DE FLUJOS DE INFORMACION.vsd”, mediante el software “VISIO”.

Anexo 4

4.4.4.1.2. DICCIONARIO DE DATOS 1

Descripción de procesos:

Número: 1
Nombre: administrador
Descripción: uso exclusivo del administrador, docente o tutor, Registro y mantenimiento completo de la base de datos.
Flujo de datos de Entrada: datos Generales, grafos, preguntas, respuestas, evaluaciones.
Flujo de datos de Salida: datos Generales, grafos, preguntas, respuestas, evaluaciones.
Lógica del proceso: selección desde menú principal del entorno de Trabajo de SEGraD.

Número: 2
Nombre: manejo de grafos
Descripción: Modulo de apoyo al aprendizaje de teoría básica y general de grafos
Flujo de datos de Entrada:
Flujo de datos de Salida:
Lógica del proceso: selección desde menú principal del entorno de Trabajo de SEGraD.

Número: 3
Nombre: aplicaciones de grafos
Descripción: modulo de apoyo al aprendizaje de las aplicaciones y/o algoritmos más representativos.
Flujo de datos de Entrada:
Flujo de datos de Salida:
Lógica del proceso: selección desde menú principal del entorno de Trabajo de SEGraD.

Número: 4
Nombre: evaluaciones
Descripción: creación y realización de las evaluaciones correspondientes a los temas propuestos.
Flujo de datos de Entrada: preguntas, grafos
Flujo de datos de Salida: evaluaciones Estructuradas
Lógica del proceso: selección desde menú principal del entorno de Trabajo de SEGraD.

Número: 5
Nombre: sistema
Descripción: cambio de claves para usuario y administrador.
Flujo de datos de Entrada: claves
Flujo de datos de Salida:
Lógica del proceso: selección desde menú principal del entorno de Trabajo de SEGraD.

Número: 6
Nombre: archivo
Descripción: manejo de información desde medios magnéticos.
Flujo de datos de Entrada: grafos
Flujo de datos de Salida: archivos .SEG
Lógica del proceso: abrir y guardar grafos por medio de selección de opciones desde el menú principal de SEGraD.

Número: 1.2
Nombre: ingreso de grafos
Descripción: forma de ingreso de grafos para la ejercitación y las evaluaciones
Flujo de datos de Entrada: grafos
Flujo de datos de Salida:
Lógica del proceso: ingreso directo de grafos a la base de datos desde el aparte correspondiente en el menú principal. (Administrador...Ingreso grafos)

Número: 1.3
Nombre: mantenimiento de registros de usuario y administrador
Descripción: actualización de datos de usuario y administrador
Flujo de datos de Entrada: datos generales (personales, académicos, seguridad)
Flujo de datos de Salida: datos generales (personales, académicos, seguridad)
Lógica del proceso: actualización de datos generales por medio de selección desde menú principal. (Administrador...Actualizar registros)

Número: 1.4
Nombre: borrado total de la base de datos
Descripción: eliminación total de los registros generales (Personales, académicos y seguridad) de la base de datos.
Flujo de datos de Entrada: confirmación de borrado
Flujo de datos de Salida: vaciado de la base de datos
Lógica del proceso: eliminación de registros por medio de selección desde menú principal. (Administrador...Borrar registros DB)

Número: 1.5
Nombre: reportes generales
Descripción: reportes detallados de datos generales de los usuarios registrados
Flujo de datos de Entrada:
Flujo de datos de Salida: datos específicos generales (personales, académicos)
Lógica del proceso: resultados en pantalla o impresos por medio de selección desde menú principal. (Administrador...Reportes generales)

Número: 1.6
Nombre: mantenimiento evaluaciones formativas
Descripción: actualización de las tablas de datos de evaluaciones formativas.
Flujo de datos de Entrada: preguntas, respuestas
Flujo de datos de Salida: preguntas, respuestas
Lógica del proceso: ingreso o eliminación de preguntas y respuestas de la evaluación formativa (Administrador...Mantenimiento formativo)

Número: 2.1
Nombre: principios básicos
Descripción: presentación de los principios básicos sobre la teoría de grafos propuesta. (apoyo animado)
Flujo de datos de Entrada: botones (reiniciar, continuar, salir , ejercitar)
Flujo de datos de Salida: explicaciones estáticas
Lógica del proceso: selección de las temáticas propuestas en este aparte desde menú principal. (grafos...Principios básicos)

Número: 2.2
Nombre: tipos de grafos
Descripción: presentación de los tipos de grafos más comunes. (apoyo simulaciones)
Flujo de datos de Entrada: botones (reiniciar, continuar, salir , ejercitar)
Flujo de datos de Salida: explicaciones estáticas
Lógica del proceso: selección de las temáticas propuestas en este aparte desde menú principal. (grafos...tipos de grafos)

Número: 2.3
Nombre: almacenamiento de una grafo en memoria
Descripción: como almacenar un grafo en la memoria de un computador.
Flujo de datos de Entrada: botones (reiniciar, continuar, salir , ejercitar)
Flujo de datos de Salida: explicaciones estáticas
Lógica del proceso: selección de las temáticas propuestas en este aparte desde menú principal. (grafos...Almacenamiento..)

Número: 2.4
Nombre: recorridos de grafos
Descripción: clases de recorridos de un grafo (apoyo simulado)
Flujo de datos de Entrada: botones (reiniciar, continuar, salir , ejercitar)
Flujo de datos de Salida: explicaciones estáticas
Lógica del proceso: menú principal. (grafos...Recorridos de un grafo)

Número: 3.1
Nombre: algoritmo de Warshall
Descripción: presentación del algoritmo, aplicaciones (apoyo simulado)
Flujo de datos de Entrada: botones (reiniciar, continuar, salir , ejercitar)
Flujo de datos de Salida: explicaciones estáticas
Lógica del proceso: selección del algoritmo propuesto en este aparte desde menú principal. (Aplicaciones... algoritmo de Warshall)

Número: 3.2
Nombre: algoritmo de Prim
Descripción: presentación del algoritmo, aplicaciones (apoyo simulado).
Flujo de datos de Entrada: botones (reiniciar, continuar, salir , ejercitar)
Flujo de datos de Salida: explicaciones estáticas
Lógica del proceso: selección del algoritmo propuesto en este aparte desde menú principal. (Aplicaciones... algoritmo de Prim)

Número: 3.3
Nombre: algoritmo de kruskal
Descripción: presentación del algoritmo, aplicaciones (apoyo simulado)
Flujo de datos de Entrada: botones (reiniciar, continuar, salir , ejercitar)
Flujo de datos de Salida: explicaciones estáticas
Lógica del proceso: selección del algoritmo propuesto en este aparte desde menú principal. (Aplicaciones... algoritmo de kruskal)

Número: 3.4
Nombre: algoritmo de Dijkstra
Descripción: presentación del algoritmo, aplicaciones (apoyo simulado)
Flujo de datos de Entrada: botones (reiniciar, continuar, salir , ejercitar)
Flujo de datos de Salida: explicaciones estáticas
Lógica del proceso: selección del algoritmo propuesto en este aparte desde menú principal. (Aplicaciones... algoritmo de Dijkstra)

Número: 3.5
Nombre: algoritmo de Floyd
Descripción: presentación del algoritmo, aplicaciones (apoyo simulado)
Flujo de datos de Entrada: botones (reiniciar, continuar, salir , ejercitar)
Flujo de datos de Salida: explicaciones estáticas
Lógica del proceso: selección del algoritmo propuesto en este aparte desde menú principal. (Aplicaciones... algoritmo de Floyd)

Número: 4.1
Nombre: elaborar evaluación
Descripción: módulo evaluativo donde el tutor puede elaborar la evaluación sumativa correspondiente al tema que desee.
Flujo de datos de Entrada: preguntas, grafos, respuestas
Flujo de datos de Salida: evaluación estructurada.
Lógica del proceso: creación de las evaluaciones por medio de selección desde el menú principal. (evaluaciones...Elaborar evaluación)

Número: 4.2
Nombre: realizar evaluación
Descripción: aparte que permite al usuario evaluarse con las previas desarrolladas con anterioridad por el docente o tutor.
Flujo de datos de Entrada: Respuestas, procesos
Flujo de datos de Salida: resultados cuantitativos
Lógica del proceso: el usuario podrá evaluar los temas avanzados por medio de selecciones desde menú principal. (evaluaciones...Realizar evaluación)

Número: 4.3
Nombre: resultados evaluaciones
Descripción: datos resultantes sobre las evaluaciones propuestas.
Flujo de datos de Entrada:
Flujo de datos de Salida: calificación sumativa
Lógica del proceso: almacenamiento de resultados evaluativos en la base de datos. (evaluaciones...Mirar resultados)

Número: 2.2.1
Nombre: grafos fuerte y débilmente conectados
Descripción: presentación de la teoría, (apoyo simulado)
Flujo de datos de Entrada: botones (reiniciar, continuar, salir , ejercitar)
Flujo de datos de Salida: explicaciones estáticas
Lógica del proceso: selección del aparte por medio del menú principal. (grafos...Tipos de grafos...grafos F. y D. conectados)

Número: 2.2.2
Nombre: grafos eulerianos y hamiltonianos
Descripción: presentación de la teoría, (apoyo simulado)
Flujo de datos de Entrada: botones (reiniciar, continuar, salir , ejercitar)
Flujo de datos de Salida: explicaciones estáticas
Lógica del proceso: selección del aparte por medio del menú principal. (grafos...Tipos de grafos...grafos Eulerianos y hamiltonianos)

Número: 2.2.3
Nombre: otros tipos de grafos
Descripción: formas teóricas a modo informativo.
Flujo de datos de Entrada: botones (tipo de grafo, salir)
Flujo de datos de Salida: explicaciones textuales y gráficas estáticas
Lógica del proceso: selección del aparte por medio del menú principal. (grafos...Tipos de grafos...Otros tipos de grafos)

Número: 2.4.1
Nombre: recorrido a lo ancho
Descripción: definiciones y explicaciones específicas simuladas del recorrido.
Flujo de datos de Entrada: botones (reiniciar, continuar, salir , ejercitar)
Flujo de datos de Salida: explicaciones estáticas
Lógica del proceso: selección del aparte por medio del menú principal. (grafos...Recorridos de grafos...Recorrido a lo ancho)

Número: 2.4.2
Nombre: recorrido en profundidad
Descripción: definiciones y explicaciones específicas simuladas del recorrido.
Flujo de datos de Entrada: botones (reiniciar, continuar, salir , ejercitar)
Flujo de datos de Salida: explicaciones estáticas
Lógica del proceso: selección del aparte por medio del menú principal. (grafos...Recorridos de grafos...Recorrido en profundidad)

Número: 4.1.1 a 4.1.12
Nombre: evaluaciones temáticas trabajada por SEGraD
Descripción: creación de la evaluación deseada con datos proporcionados por SEGraD o ingresados por el tutor.
Flujo de datos de Entrada: preguntas, grafos, respuestas
Flujo de datos de Salida: evaluación estructurada de acuerdo al tema
Lógica del proceso: selección del aparte por medio del menú principal. (evaluaciones ... Elaborar evaluación...temática).

Número: 4.2.3
Nombre: auto evaluación
Descripción: evaluación aleatoria propuesta por SEGraD cuando el usuario lo considere conveniente.
Flujo de datos de Entrada: respuestas
Flujo de datos de Salida: resultados
Lógica del proceso: el usuario puede autoevaluarse seleccionando el aparte correspondiente en el menú (evaluaciones...Realizar evaluación...Temática).

Número: 4.2.1
Nombre: evaluación dirigida
Descripción: evaluación sumativa elaborada y programada por el docente
Flujo de datos de Entrada: respuestas
Flujo de datos de Salida: resultados cuantitativos
Lógica del proceso: elaboración de la evaluación programada. selección de la evaluación en el menú (evaluaciones ...Realizar evaluación...evaluación dirigida).

Número: 4.2.2
Nombre: evaluación aleatoria
Descripción: evaluación aleatoria sumativa propuesta por SEGraD
Flujo de datos de Entrada: respuestas
Flujo de datos de Salida: resultados cuantitativos
Lógica del proceso: elaboración de la evaluación programada seleccionada desde el aparte correspondiente en el menú principal. (evaluaciones ...Realizar evaluación...evaluación aleatoria)

Número: 4.2.4
Nombre: evaluación final
Descripción: propuesta por SEGraD, involucra toda la temática propuesta
Flujo de datos de Entrada: respuestas
Flujo de datos de Salida: resultados cuantitativos
Lógica del proceso: elaboración de la evaluación final desde el aparte correspondiente en el menú principal. (evaluaciones ...Realizar evaluación...final)

Número: 3.x.1
Nombre: cargar grafo niveles 1,2,3
Descripción: propone un grafo aleatorio par la practica dependiendo del nivel seleccionado
Flujo de datos de Entrada: nivel (1,2,3)
Flujo de datos de Salida: grafo en pantalla
Lógica del proceso: selección desde el combo grado de dificultad el nivel deseado (solo en formas de ejercitación)

Número: 3.x.2
Nombre: crear grafo
Descripción: selección desde el combo grado de dificultad la opción crear grafo (solo en formas de ejercitación)
Flujo de datos de Entrada: crear grafo
Flujo de datos de Salida: grafo en pantalla
Lógica del proceso: habilita botones de diseño de grafos.

Número: 3.x.3
Nombre: abrir grafo
Descripción: selección desde el menú principal la opción correspondiente. (Archivo...Abrir)
Flujo de datos de Entrada: selección de archivo .SEG
Flujo de datos de Salida: grafo en pantalla
Lógica del proceso: permite abrir un archivo .SEG donde se almacenan los datos que permiten visualizar un grafo en pantalla

Número: 3.x.4
Nombre: guardar grafo
Descripción: permite almacenar en un medio magnético el grafo que se encuentre en esos momentos en pantalla
Flujo de datos de Entrada: grafo en pantalla
Flujo de datos de Salida: grafos a disco
Lógica del proceso: selección desde el menú principal la opción correspondiente. (Archivo...guardar)

Número: 3.x.5
Nombre: ejecutar Práctica
Descripción: verifica el grafo en pantalla e inicializa las validaciones de acuerdo al algoritmo de la práctica que se este ejecutando.
Flujo de datos de Entrada: grafo en pantalla
Flujo de datos de Salida: instrucciones de procedimientos y Mensajes de error
Lógica del proceso: selección del botón correspondiente en las formas de ejercitación. (solo en las formas de ejercitación)

Número: 3.x.2.1
Nombre: adicionar vértice
Descripción: opción que permite para el diseño de un grafo crear vértices en tiempo de ejecución
Flujo de datos de Entrada: objeto vértice (Label)
Flujo de datos de Salida: visualización vértices
Lógica del proceso: selección del botón Adicionar vértice de la forma correspondiente. (solo en las formas de ejercitación)

Número: 3.x.2.2
Nombre: adicionar arco
Descripción: opción que permite para el diseño de un grafo crear arcos en tiempo de ejecución
Flujo de datos de Entrada: objeto Arco (Line)
Flujo de datos de Salida: visualización arcos
Lógica del proceso: selección del botón Adicionar arco de la forma correspondiente. (solo en las formas de ejercitación)

Número: 3.x.2.3
Nombre: eliminar vértice
Descripción: permite eliminar un vértice durante el diseño de un grafo en tiempo de ejecución.
Flujo de datos de Entrada: orden eliminación vértice
Flujo de datos de Salida: descargar vértice
Lógica del proceso: selección del botón Eliminar de la forma correspondiente. (solo en las formas de ejercitación)

Número: 3.x.2.4
Nombre: eliminar arco
Descripción: permite eliminar un arco durante el diseño de un grafo en tiempo de ejecución.
Flujo de datos de Entrada: orden eliminación arco
Flujo de datos de Salida: descargar arco
Lógica del proceso: selección del botón Eliminar de la forma correspondiente. (solo en las formas de ejercitación)

Número: 3.x.5.2
Nombre: validar grafo
Descripción: durante la ejecución de la practica valida cada una de las acciones realizadas por el usuario e indica los procedimientos correctos a seguir.
Flujo de datos de Entrada: Acciones realizadas en tiempo de ejecución.
Flujo de datos de Salida: Mensajes de error y explicación de procedimientos correctos.
Lógica del proceso: selección del botón cada vez que se realice un procedimiento (solo en formas de ejercitación)

Número: 3.x.3.1
Nombre: abrir archivo *. SEG
Descripción: opción del menú Archivo que sólo permite abrir archivos .SEG
Flujo de datos de Entrada: archivo .SEG
Flujo de datos de Salida: visualización de grafo en pantalla
Lógica del proceso: selección del aparte Abrir en el menú archivo desde el menú principal de SEGraD (solo para formas de ejercitación)

Número: 3.x.4.1
Nombre: guardar grafo
Descripción: opción del menú Archivo del menú principal que permite almacenar el grafo en pantalla en un medio magnético local o externo.
Flujo de datos de Entrada: grafo en pantalla (coordenadas, descripción)
Flujo de datos de Salida: almacenamiento
Lógica del proceso: selección del aparte guardar en el menú archivo desde el menú principal de SEGraD (solo para formas de ejercitación)

Número: 5.1.1
Nombre: cambio de clave
Descripción: forma de uso exclusivo del administrador, permite cambiar claves de acceso a los usuarios.
Flujo de datos de Entrada: código, clave anterior, nueva clave y recordatorio
Flujo de datos de Salida:
Lógica del proceso: selección de la opción desde el menú principal correspondiente. (Sistema...Cambio de clave)

Número: 4.1.3.1
Nombre: crear grafo
Descripción: habilita los botones que permiten el diseño de los grafos en tiempo de ejecución.
Flujo de datos de Entrada: sentencias dadas por usuario
Flujo de datos de Salida: carga de objetos creados (label, line)
Lógica del proceso: click sobre el botón correspondiente

Número: 4.1.3.2
Nombre: nuevo
Descripción: limpia la zona de creación del grafo (descarga objetos actuales dentro del área mencionada)
Flujo de datos de Entrada: sentencias dadas por usuario
Flujo de datos de Salida: descarga
Lógica del proceso: click sobre el botón correspondiente

Número: 4.1.3.3
Nombre: guardar grafo
Descripción: almacena el grafo creado en la tabla de datos correspondiente a la evaluación que se esta trabajando en ese momento.
Flujo de datos de Entrada: sentencias dadas por usuario
Flujo de datos de Salida:
Lógica del proceso: click sobre el botón correspondiente

Número: 4.1.3.4
Nombre: tipo de recorrido
Descripción: permite seleccionar el tipo de recorrido que vamos a aplicar en la evaluación sobre el grafo seleccionado
Flujo de datos de Entrada: selección por usuario
Flujo de datos de Salida: tipo de recorrido
Lógica del proceso: selección de recorrido correspondiente en combo Recorrido

Número: 4.1.3.5
Nombre: verificar recorrido según opción
Descripción: activa funciones de verificación que realizan el algoritmo de recorrido internamente.
Flujo de datos de Entrada: orden de verificación
Flujo de datos de Salida: mensajes de error o instrucciones
Lógica del proceso: click sobre el botón correspondiente

Número: 4.1.3.6
Nombre: abrir grafo
Descripción: permite seleccionar un grafo de una tabla de datos.
Flujo de datos de Entrada: código y nombre de grafo
Flujo de datos de Salida: carga grafo en pantalla
Lógica del proceso: selección desde el combo "grafo"

Número: 4.1.4.1
Nombre: crear grafo
Descripción: habilita los botones que permiten el diseño de los grafos en tiempo de ejecución.
Flujo de datos de Entrada: sentencias dadas por usuario
Flujo de datos de Salida: carga de objetos creados (label, line)
Lógica del proceso: click sobre el botón correspondiente

Número: 4.1.4.2
Nombre: nuevo
Descripción: limpia la zona de creación del grafo (descarga objetos actuales dentro del área mencionada)
Flujo de datos de Entrada: sentencias dadas por usuario
Flujo de datos de Salida: descarga
Lógica del proceso: click sobre el botón correspondiente

Número: 4.1.4.3
Nombre: guardar grafo
Descripción: almacena el grafo creado en la tabla de datos correspondiente a la evaluación que se esta trabajando en ese momento.
Flujo de datos de Entrada: sentencias dadas por usuario
Flujo de datos de Salida:
Lógica del proceso: click sobre el botón correspondiente

Número: 4.1.4.4
Nombre: abrir grafo
Descripción: permite seleccionar un grafo deseado desde un combo
Flujo de datos de Entrada: selección por usuario
Flujo de datos de Salida: carga de grafo
Lógica del proceso: selección de grafo en combo grafo

Número: 4.1.4.5
Nombre: verificar aplicación
Descripción: activa funciones de verificación que realizan el algoritmo correspondiente internamente.
Flujo de datos de Entrada: orden de verificación
Flujo de datos de Salida: mensajes de error o instrucciones
Lógica del proceso: click sobre el botón correspondiente

Descripción de Almacenamientos:

Nombre: datos generales
Descripción: tabla base de datos db_grafos
Área: global
Formato: tabla de datos
Datos: código, Nombres y apellidos, dirección, teléfono, e_mail, programa, semestre, clave y recordatorio
Comentarios:

Nombre: grafos
Descripción: tabla base de datos db_grafos
Área: local
Formato: tabla de datos
Datos: coordenadas, índices, Nombres de grafos
Comentarios: los datos son generalizados

Nombre: algoritmos
Descripción: tabla base de datos db_grafos
Área: local
Formato: tabla de datos
Datos: tipo de algoritmo
Comentarios: datos generalizados

Nombre: grafos_Disco
Descripción: almacenamiento en medios magnéticos locales o externos (disco duro o micro discos de 3.5")
Área: global
Formato: archivo de datos .SEG
Datos: coordenadas, índices, vértices y arcos
Comentarios:

Nombre: evaluaciones
Descripción: tabla base de datos db_grafos
Área: global
Formato: tabla de datos
Datos: preguntas, respuestas, opciones, grafos, nombres evaluaciones, tipos de evaluaciones
Comentarios: Datos generalizados

Nombre: nov_pantalla
Descripción: Almacenamiento temporal en memoria
Área: global
Formato: almacenamiento memoria, buffer
Datos: Coordenadas, índices
Comentarios:

Nombre: eval_privas
Descripción: almacenamiento de registros involucrados con la evaluación del tema Principios Básicos
Área: local
Formato: tabla de datos
Datos: preguntas, respuestas, opciones, grafos
Comentarios: eval_privas esta definido por un conjunto de tablas de datos que manejan solo la temática de conceptos básicos en las evaluaciones.

Nombre: eval_tipos
Descripción: almacenamiento de registros involucrados con la evaluación de los temas incluidos en el aparte de tipos de grafos
Área: global
Formato: tablas de datos
Datos: preguntas, respuestas, opciones, grafos
Comentarios: eval_tipos esta definido por un conjunto de tablas de datos que manejan solo la temática de tipos de grafos en las evaluaciones.

Nombre: eval_regra
Descripción: almacenamiento de registros involucrados con la evaluación de los temas incluidos en el aparte de recorridos de un grafo
Área: global
Formato: tablas de datos
Datos: preguntas, respuestas, opciones, grafos
Comentarios: eval_regra esta definido por un conjunto de tablas de datos que manejan solo la temática de recorridos de un grafo.

Nombre: eval_aplic
Descripción: almacenamiento de registros involucrados con la evaluación de los temas incluidos en el aparte de aplicaciones de grafos (algoritmos)
Área: global
Formato: tablas de datos
Datos: preguntas, respuestas, opciones, grafos
Comentarios: eval_aplic esta definido por un conjunto de tablas de datos que manejan solo la temática de aplicaciones (Alg. Warshall, Prim, kruskal, Dijkstra y Floyd) en las evaluaciones.

Descripción del Flujo de Datos:

Nombre: opción
Descripción: activa la forma o el proceso seleccionado para su manipulación
Tipo de Flujo de Datos: Interno
Datos: variables, sentencias
Comentarios: las opciones dependen del menú y/o de la forma activa

Nombre: Ingreso de datos
Descripción: apertura de la base de datos en cualquiera de sus tablas para ingresar registros por cualquiera de los procesos activos
Tipo de Flujo de Datos: Interno, Base de datos
Datos: registros
Comentarios:

Nombre: Acceso de datos
Descripción: apertura de la base de datos en cualquiera de sus tablas para acceder a datos específicos por cualquiera de los procesos activos.
Tipo de Flujo de Datos: Interno, Base de datos
Datos: registros
Comentarios:

Nombre: grafo aleatorio
Descripción: carga un grafo aleatorio desde la base de datos para ser visualizado en pantalla.
Tipo de Flujo de Datos: interno, Base de datos
Datos: registros (coordenadas e índices)
Comentarios:

Nombre: a pantalla
Descripción: proceso de conversión de los registros y visualización en pantalla de las acciones o decisiones tomadas en tiempo de ejecución.
Tipo de Flujo de Datos: pantalla
Datos: coordenadas, variables, sentencias
Comentarios: manejo de buffer y memoria de computador

Nombre: abrir grafo
Descripción: función definida de apertura de archivos . SEG
Tipo de Flujo de Datos: archivos .SEG
Datos: encabezados, Coordenadas, índices, No Vértices, No Arcos
Comentarios:

Nombre: guardar grafo
Descripción: Función definida de almacenamiento de archivos . SEG
Tipo de Flujo de Datos: archivos . SEG
Datos: coordenadas, índices, # de vértices y arcos
Comentarios:

Nombre: visualización
Descripción: conversión de datos numéricos a objetos interpretables por usuario.
Tipo de Flujo de Datos: pantalla
Datos: variables, sentencias
Comentarios:

Nombre: opción ejercitar
Descripción: selección que carga la forma de ejercitación correspondiente
Tipo de Flujo de Datos: Interno
Datos: variables
Comentarios:

Nombre: visualización de la evaluación
Descripción: permite al usuario acceder a una evaluación predeterminada
Tipo de Flujo de Datos: interno
Datos: preguntas, respuestas, grafos, opciones
Comentarios:

Nombre: acceso a reportes de datos de usuario
Descripción: acción que permite cargar reportes específicos del usuario o la aplicación.
Tipo de Flujo de Datos: interno, Base de datos
Datos: forma tipo reportes, registros
Comentarios:

Nombre: eliminación total de registros
Descripción: acción de vaciado de la base de datos
Tipo de Flujo de Datos: sentencia
Datos: registros
Comentarios: eliminación total de los registros relacionados de la base de datos, se usa esta opción para iniciar nuevos cursos.

Nombre: acceso a reportes sobre los grafos
Descripción: carga reportes específicos sobre los grafos disponibles para las evaluaciones o ejercitaciones de los usuarios.
Tipo de Flujo de Datos: interno, Base de datos
Datos: registros, forma tipo reporte
Comentarios:

Nombre: acceso a reportes evaluativos
Descripción: carga reportes específicos sobre resultados y avances evaluativos de los usuarios registrados.
Tipo de Flujo de Datos: interno, Base de datos
Datos: registros, forma tipo reporte
Comentarios:

Nombre: acceso a datos para evaluaciones
Descripción: acción empleada por el tutor cuando necesita crear evaluaciones o modificarlas.
Tipo de Flujo de Datos: interno, base de datos
Datos: registros
Comentarios: uso exclusivo del administrador

Nombre: acceso a grafos para evaluaciones
Descripción: acción aplicada para acceder a un grafo específico y anexarlo en una evaluación de usuario.
Tipo de Flujo de Datos: interno, base de datos
Datos: registros
Comentarios:

Nombre: resultados evaluaciones
Descripción: opción que carga reportes específicos individuales sobre los resultados de las evaluaciones realizadas.
Tipo de Flujo de Datos: interno; base de datos
Datos: forma de reporte, registros
Comentarios:

Nombre: realizar evaluación
Descripción: acción que carga una evaluación específica o aleatoria según se desee.
Tipo de Flujo de Datos: interno, base de datos
Datos: registros
Comentarios:

Nombre: opciones desde combos
Descripción: acciones específicas en un objeto (combo) que cargan registros o visualizan objetos en pantalla.
Tipo de Flujo de Datos: interno
Datos: variables, sentencias
Comentarios:

Nombre: opciones desde menú
Descripción: determinada por la opción seleccionada desde el menú principal.
Tipo de Flujo de Datos: interno
Datos: variables, sentencias
Comentarios:

Nombre: Aplicar algoritmo
Descripción: función que ejecuta el algoritmo seleccionado.
Tipo de Flujo de Datos: interno
Datos: variables y sentencias
Comentarios:

Nombre: opción desde barra de botones
Descripción: barra de botones que determina el diseño y manipulación de grafos.
Tipo de Flujo de Datos: interno
Datos: variables y sentencias
Comentarios:

Nombre: verificación
Descripción: función de validación de eventos realizados por el usuario
Tipo de Flujo de Datos: interno
Datos: variables, mensajes
Comentarios:

Nombre: mensajes de error
Descripción: mensajes emergentes por las validaciones de los eventos incorrectos realizados por los usuarios.
Tipo de Flujo de Datos: interno
Datos: variable, código de mensaje
Comentarios:

Nombre: almacenamiento de grafos
Descripción: ingreso de grafos en archivos o tablas de datos
Tipo de Flujo de Datos: interno
Datos: coordenadas, índices, nombre y tipo de grafo
Comentarios:

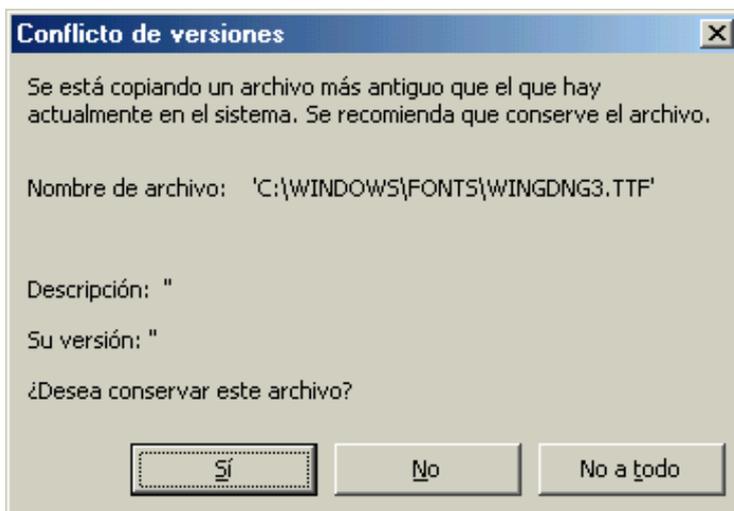
Manual de usuario

Al igual que otros programas, para hacer uso de SEGraD es necesario realizar antes un proceso de instalación el cual debe llevarse a cabo de la siguiente manera:

1. Insertar el CD- ROM que contiene el paquete de instalación de SEGraD.
2. Hacemos doble Click en el archivo setup.exe.
3. Seguimos las instrucciones que se presentan durante el proceso de instalación.

Durante la instalación debemos tener en cuenta algunos mensajes que se generan al tratar de copiar archivos ya existentes. Este caso se presenta con los archivos de tipo fuentes (.ttf) y algunos ocx que son necesarios para el correcto funcionamiento de SEGraD.

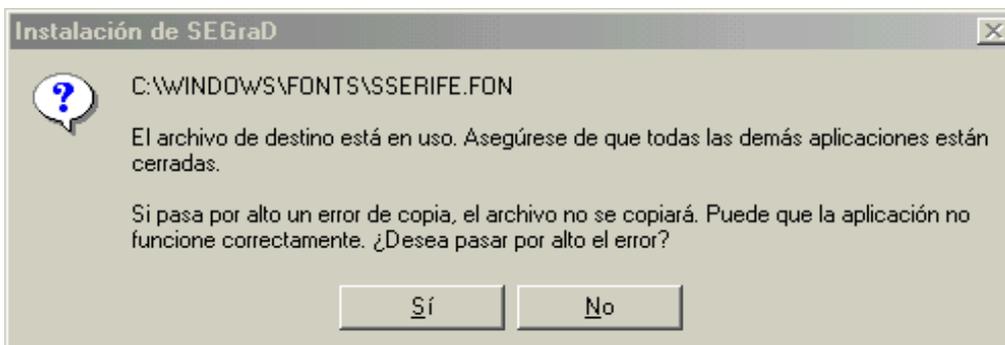
Veamos algunos de los posibles mensajes:



En este caso recomendamos hacer clic en la opción “Sí”.



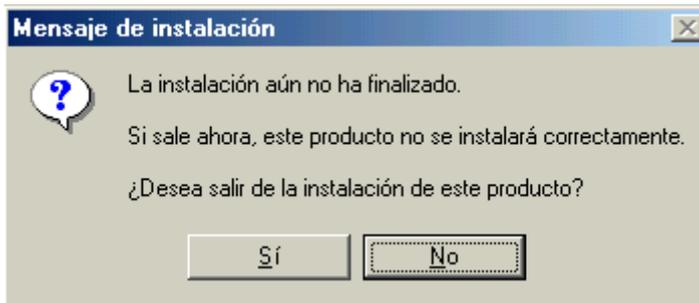
En este caso recomendamos hacer clic en la opción “Omitir”.



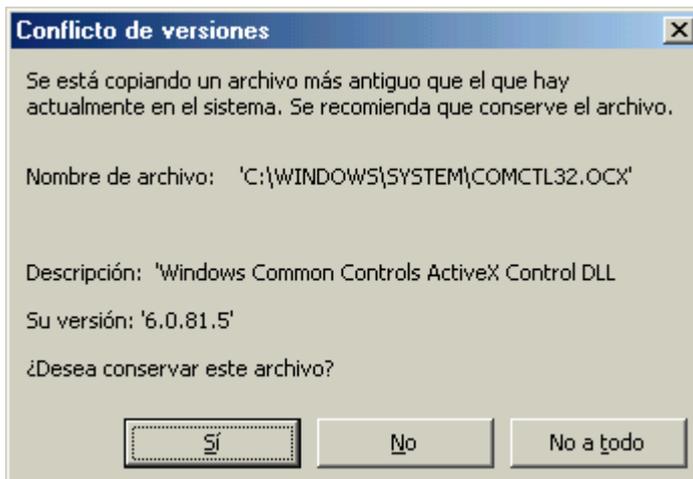
En este caso recomendamos hacer clic en la opción “Sí”.



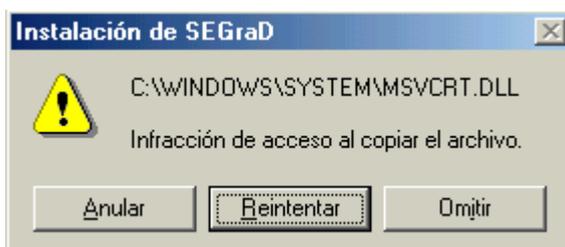
Este error se genera por la acción anterior en la cual se pasó por alto un error de copia de un archivo que estaba en uso.



Los errores anteriores no afectarán el correcto funcionamiento de SEGrAD, por ello si el usuario desea puede continuar con la instalación normalmente o en caso contrario abortarla.



En este caso recomendamos hacer clic en la opción “Sí”.



En este caso recomendamos hacer clic en la opción “Omitir”.



En este caso recomendamos hacer clic en la opción “Omitir”.

Cabe mencionar que todos los errores e infracciones anteriores se producen porque se debe garantizar el buen funcionamiento de SEGrAD y para ello debemos suponer que el equipo donde será instalado no cuenta con los componentes o archivos necesarios.

Una vez finalizada la instalación, procedemos a ejecutar el software SEGrAD ubicando el acceso directo al archivo SEGrAD.exe, haciendo click en Inicio->Programas->SEGrAD, con lo cual se abrirá la forma de “Inicio de Sesión”.

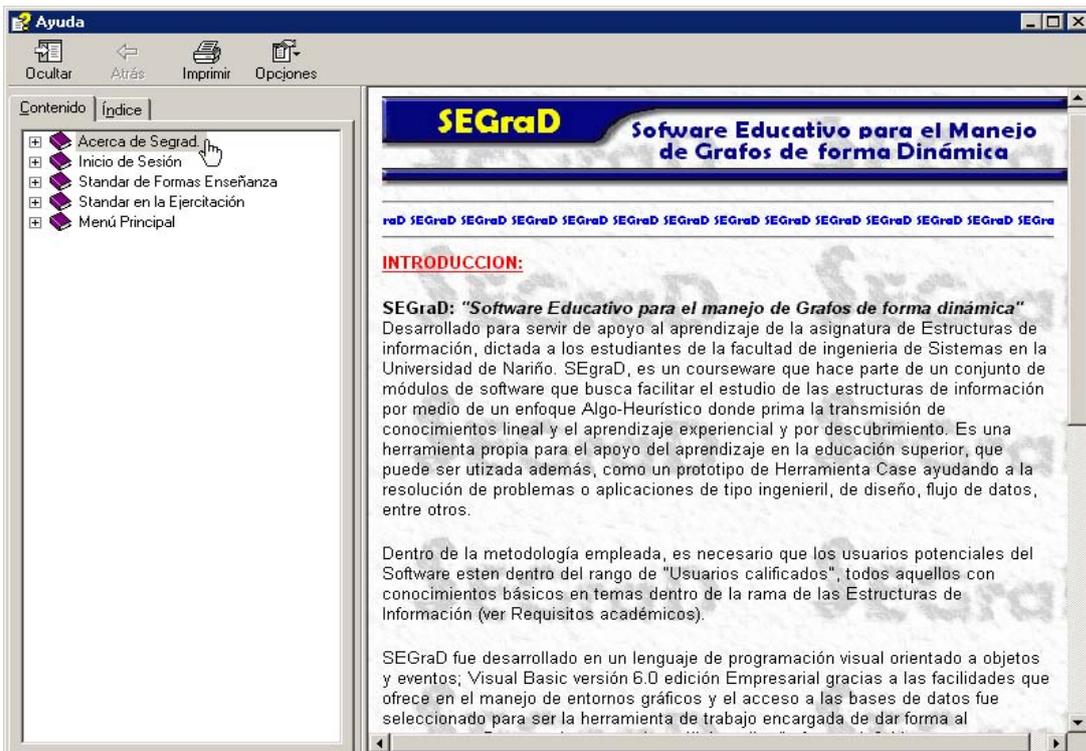


El funcionamiento de aquí en adelante, puede ser perfectamente asistido por la ayuda en línea con que cuenta SEGrAD. La ayuda puede ser invocada de 3

maneras: presionando la tecla F1, haciendo click en el menú textual Ayuda->Temas de Ayuda F1 ó en el menú gráfico Ayuda, como se muestra en la siguiente figura:

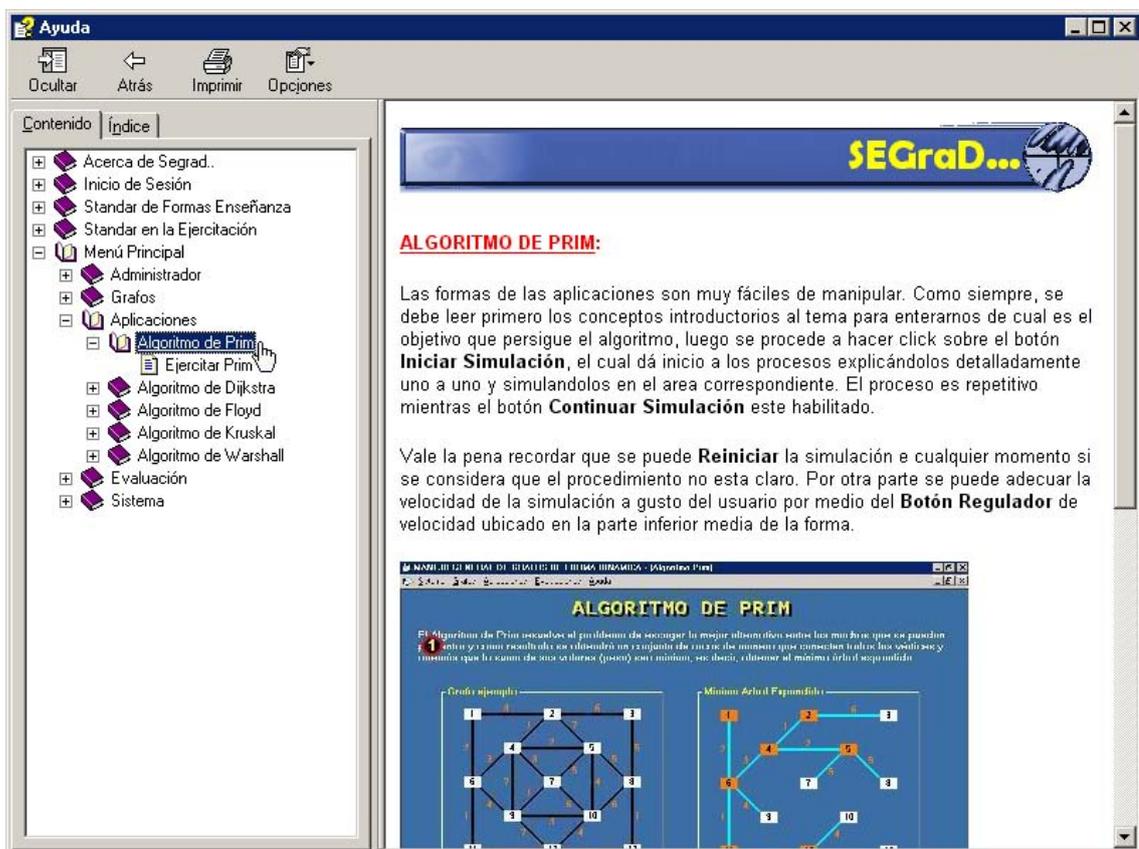


El resultado de cualquiera de las acciones anteriores es la visualización de la presentación inicial de la ayuda de SEGraD.



Veamos un ejemplo de cómo utilizar la ayuda. Suponiendo que un usuario desea conocer el tema “Algoritmo de Prim”, el procedimiento que deberá seguir es el siguiente:

- 1) Hacer click en el ítem Menú Principal->Aplicaciones->Algoritmo de Prim del árbol de contenido como muestra la figura.



- 2) En algunos casos es posible hallar hipervínculos que permiten conocer más a fondo la información que contiene SEGrAD del tema en estudio. Siguiendo con el ejemplo, en la explicación del Algoritmo de Prim se presenta un hipervínculo que permite visualizar la ayuda acerca del entorno de la ejercitación de éste algoritmo. Como se muestra en las figuras:

Ayuda

Ocultar Atrás Imprimir Opciones

Contenido Índice

- Acercas de Segrad..
- Inicio de Sesión
- Standar de Formas Enseñanza
- Standar en la Ejercitación
- Menú Principal
 - Administrador
 - Grafos
 - Aplicaciones
 - Algoritmo de Prim
 - Ejercitar Prim
 - Algoritmo de Dijkstra
 - Algoritmo de Floyd
 - Algoritmo de Kruskal
 - Algoritmo de Warshall
 - Evaluación
 - Sistema



1. **Concepto:** El Algoritmo de Prim resuelve el problema de escoger la mejor alternativa entre las muchas que se puedan presentar y como resultado se obtendrá un conjunto de arcos de manera que conecten todos los vértices y además que la suma de sus valores (peso) sea mínima, es decir, obtener el mínimo árbol expandido.

2. **Visualización de Progreso:** Util para la comprensión del progreso del algoritmo. Muestra la diferencia entre los vértices manipulados en determinado momento de la simulación.

3. **Ejercitar:** Comprendidos los conceptos el usuario puede ingresar al módulo de ejercitación, el cual nos ayudará a prepararnos para el siguiente paso que es la evaluación, que superada con éxito nos permitira el acceso al siguiente aparte del menú.

Ayuda

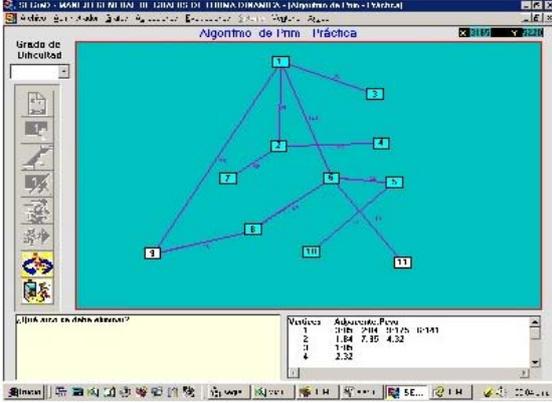
Ocultar Atrás Imprimir Opciones

Contenido Índice

- Acercas de Segrad..
- Inicio de Sesión
- Standar de Formas Enseñanza
- Standar en la Ejercitación
- Menú Principal
 - Administrador
 - Grafos
 - Aplicaciones
 - Algoritmo de Prim
 - Ejercitar Prim
 - Algoritmo de Dijkstra
 - Algoritmo de Floyd
 - Algoritmo de Kruskal
 - Algoritmo de Warshall
 - Evaluación
 - Sistema

SEGrAD...

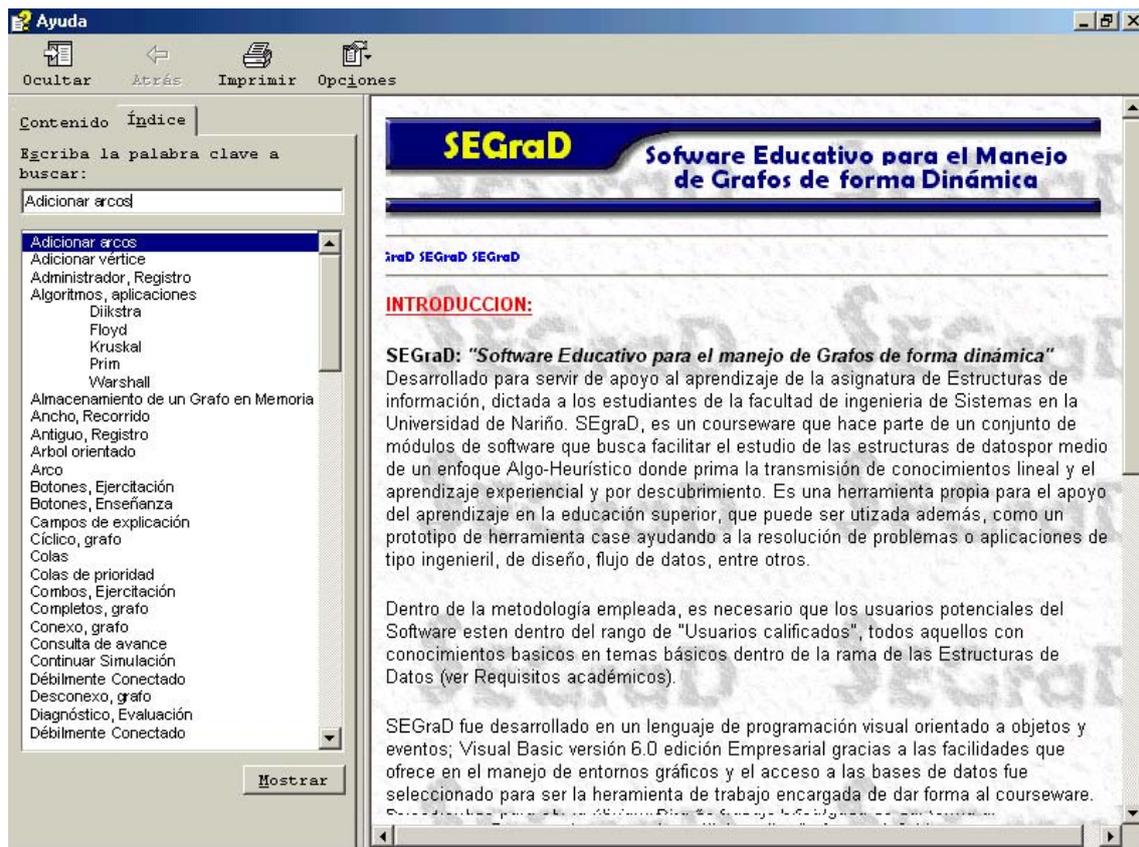
EJERCITACION PARA EL ALGORITMO DE PRIM:



Es muy sencillo manipular la forma de ejercitación del Algoritmo de **Prim** teniendo en cuenta los siguientes puntos:

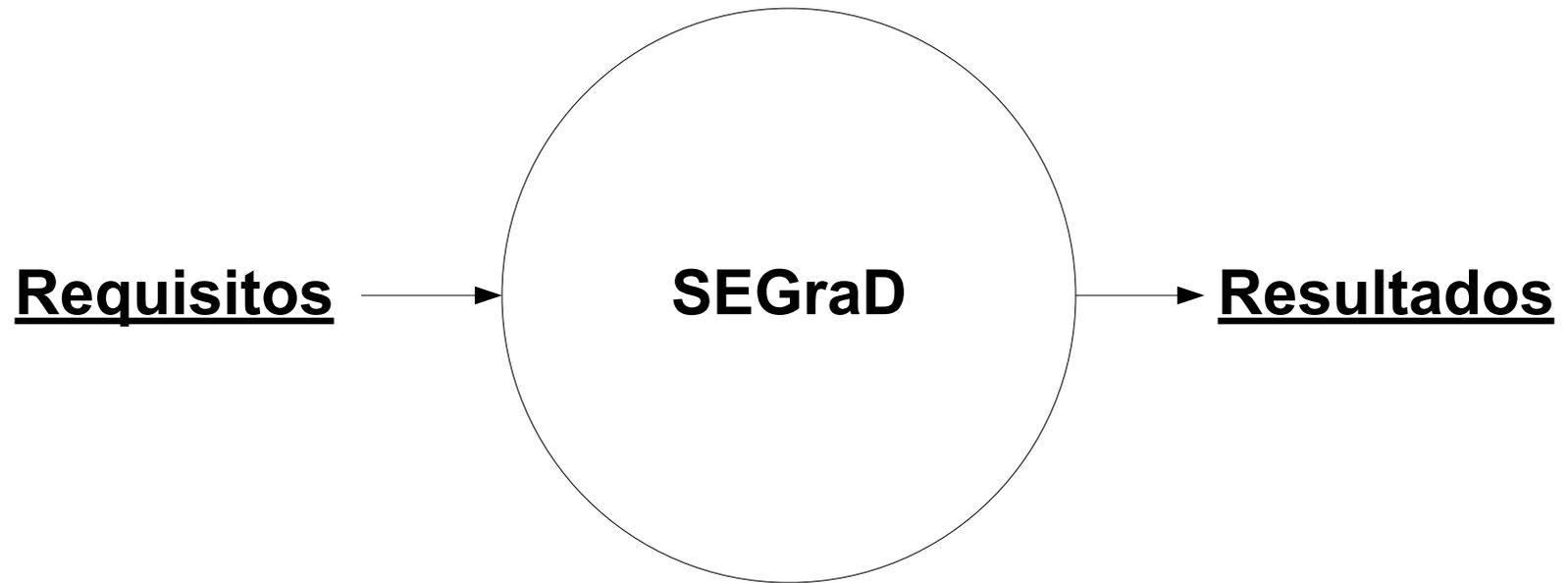
1. Si deseo ejercitarme dejando que SEGrAD me proponga un grafo al azar.
2. Si deseo ejercitarme creando mi propio Grafo.

La ayuda también permite consultar a manera de índice, lo que agiliza la búsqueda del tema requerido. Para ello se debe hacer click en la pestaña "Índice", se escribe la palabra clave en su respectivo campo y por último se hace click en el botón "Mostrar".

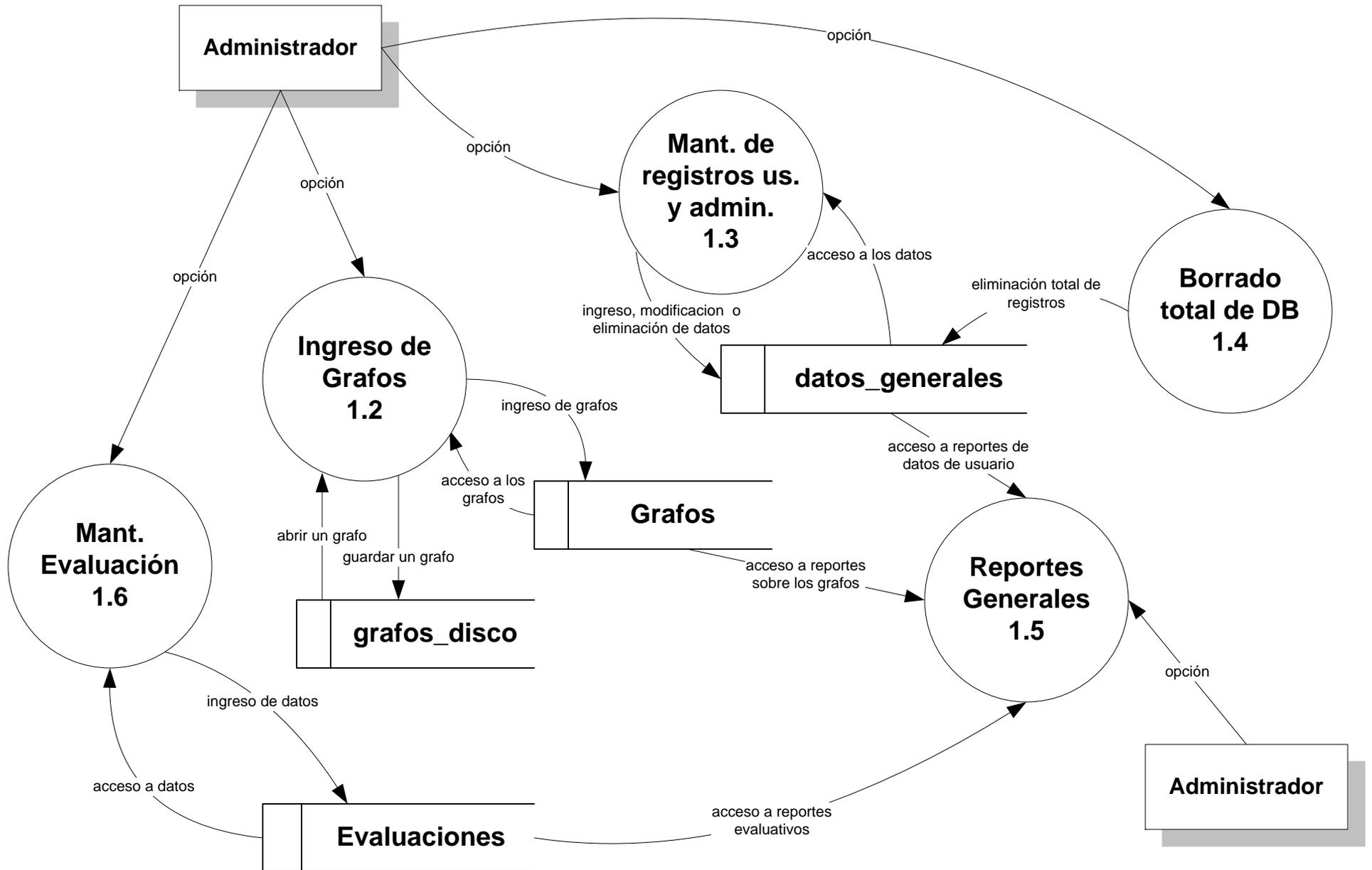


Siguiendo correctamente las instrucciones descritas en cada uno de los items de la ayuda, se facilitará el manejo de los entornos que ofrece SEGrAD, permitiendo al usuario aprovechar al máximo la información que éste brinda.

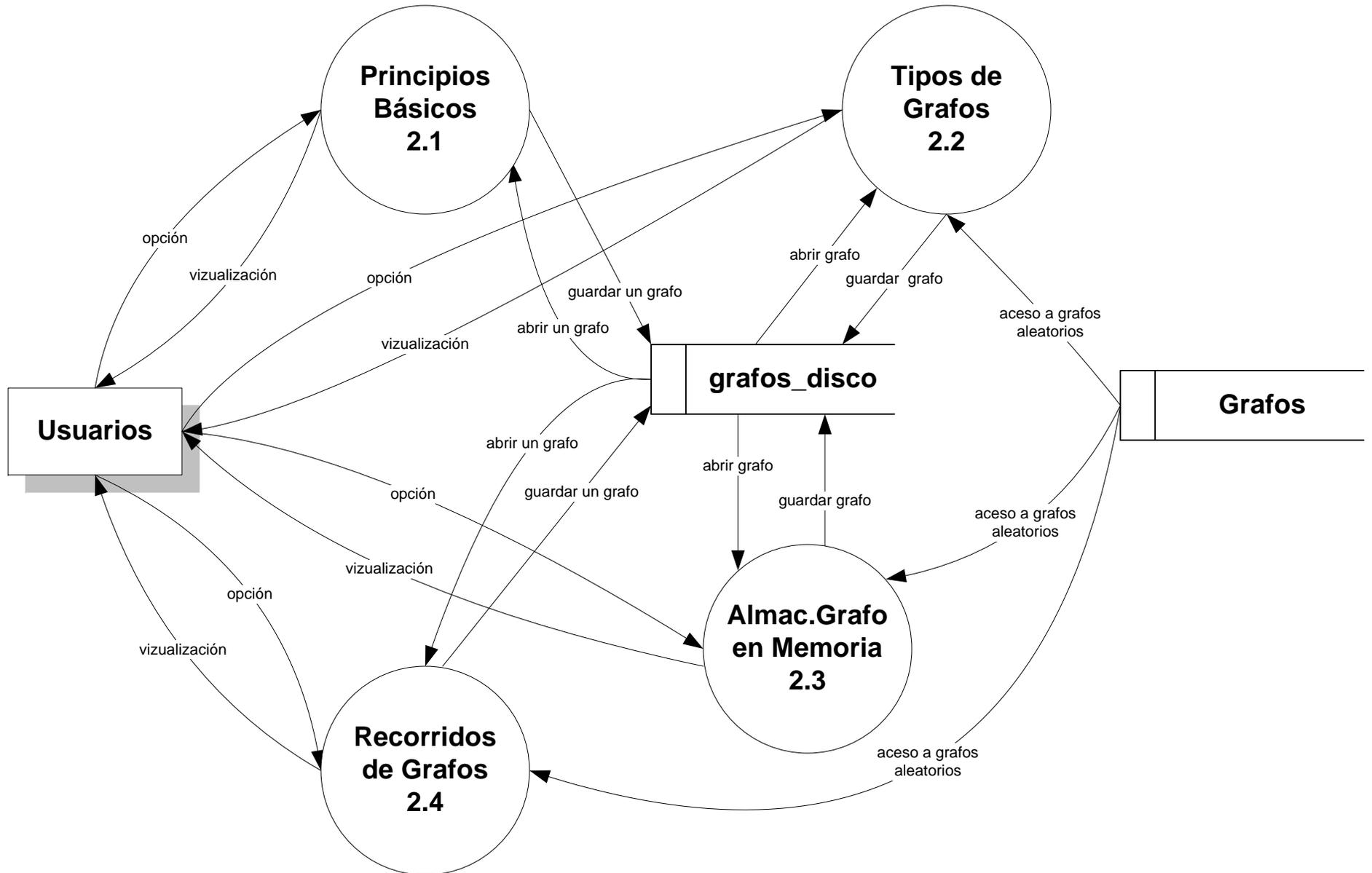
Nivel 0



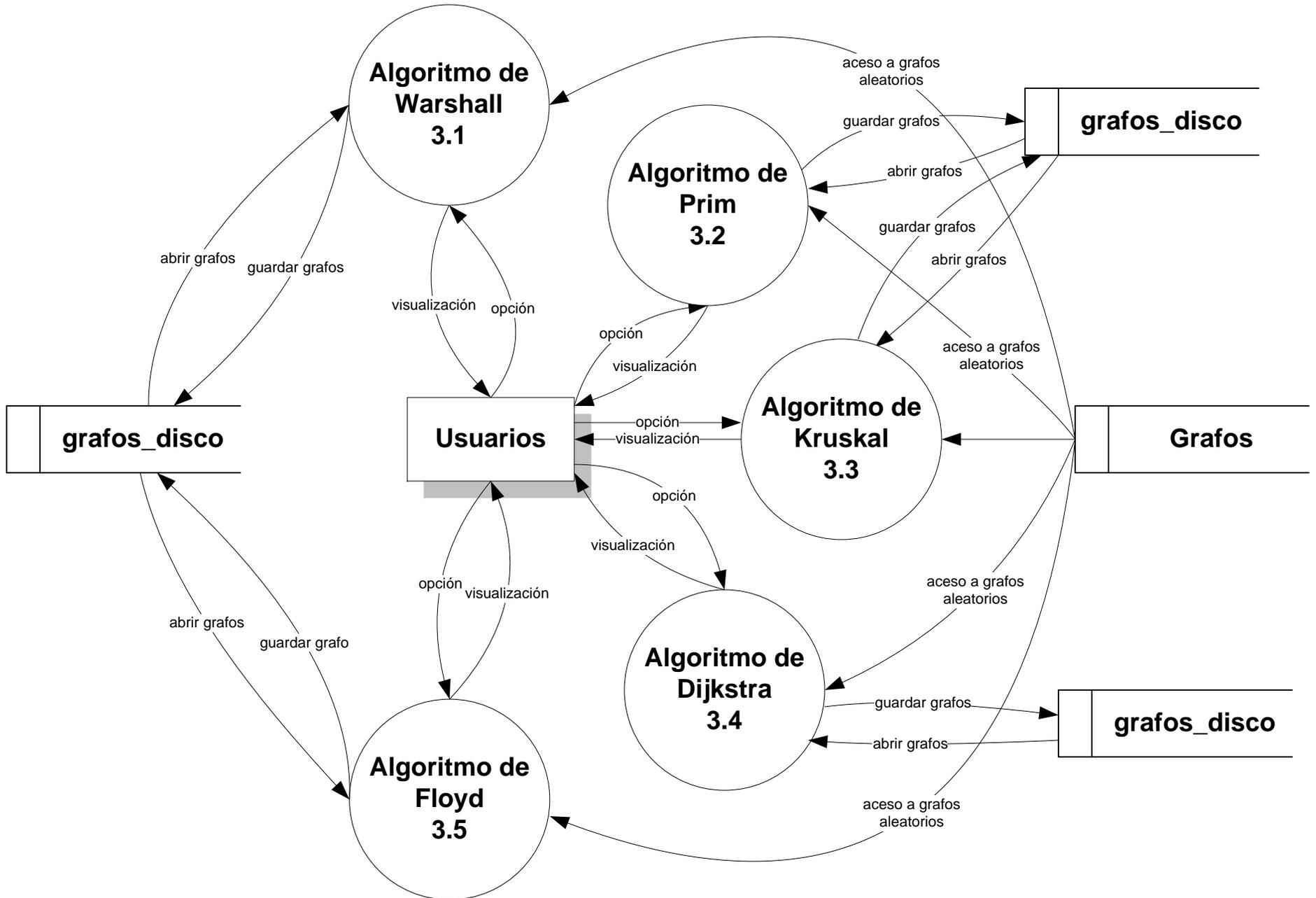
Nivel 2 (Administrador)



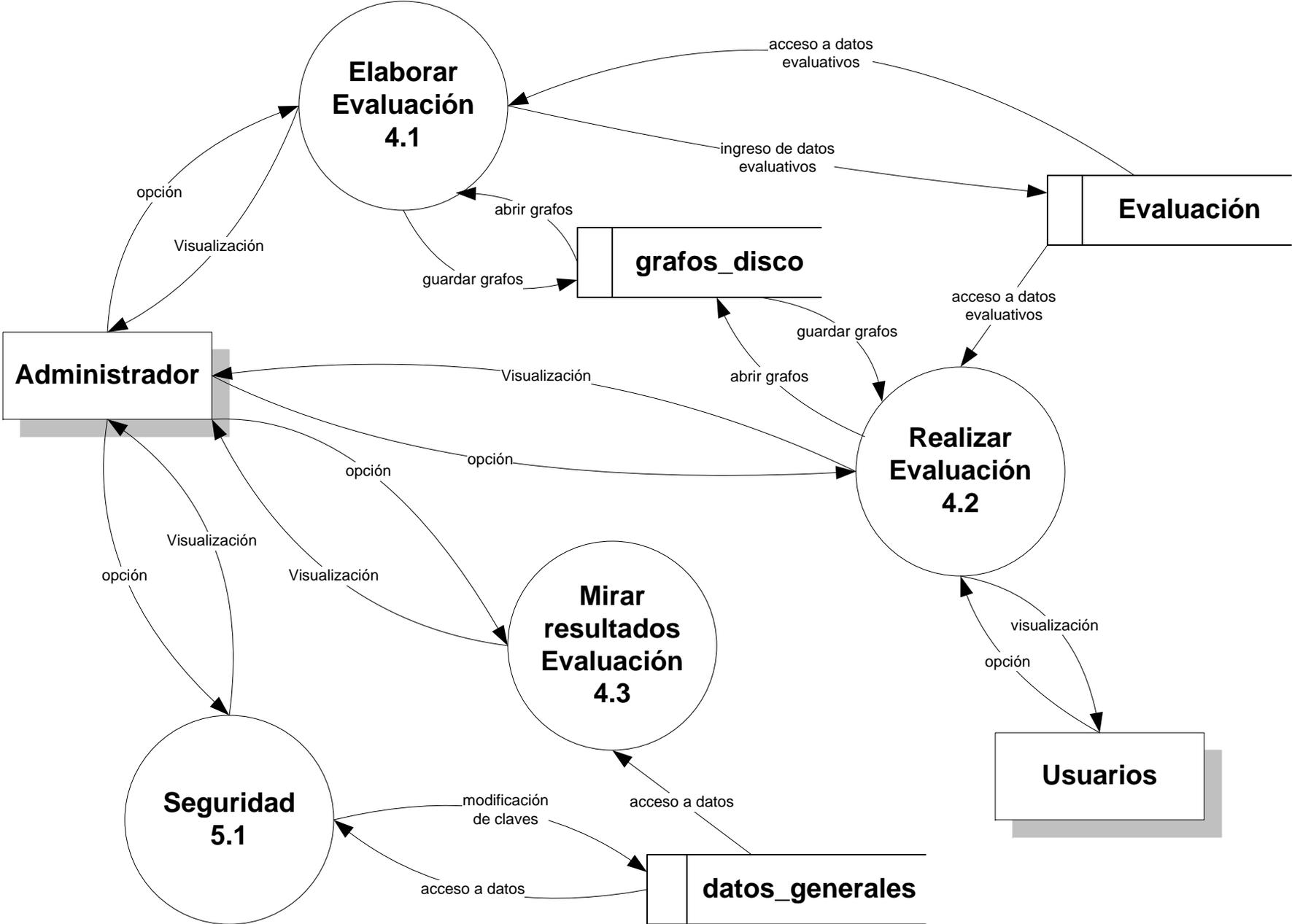
Nivel 2 (Grafos)



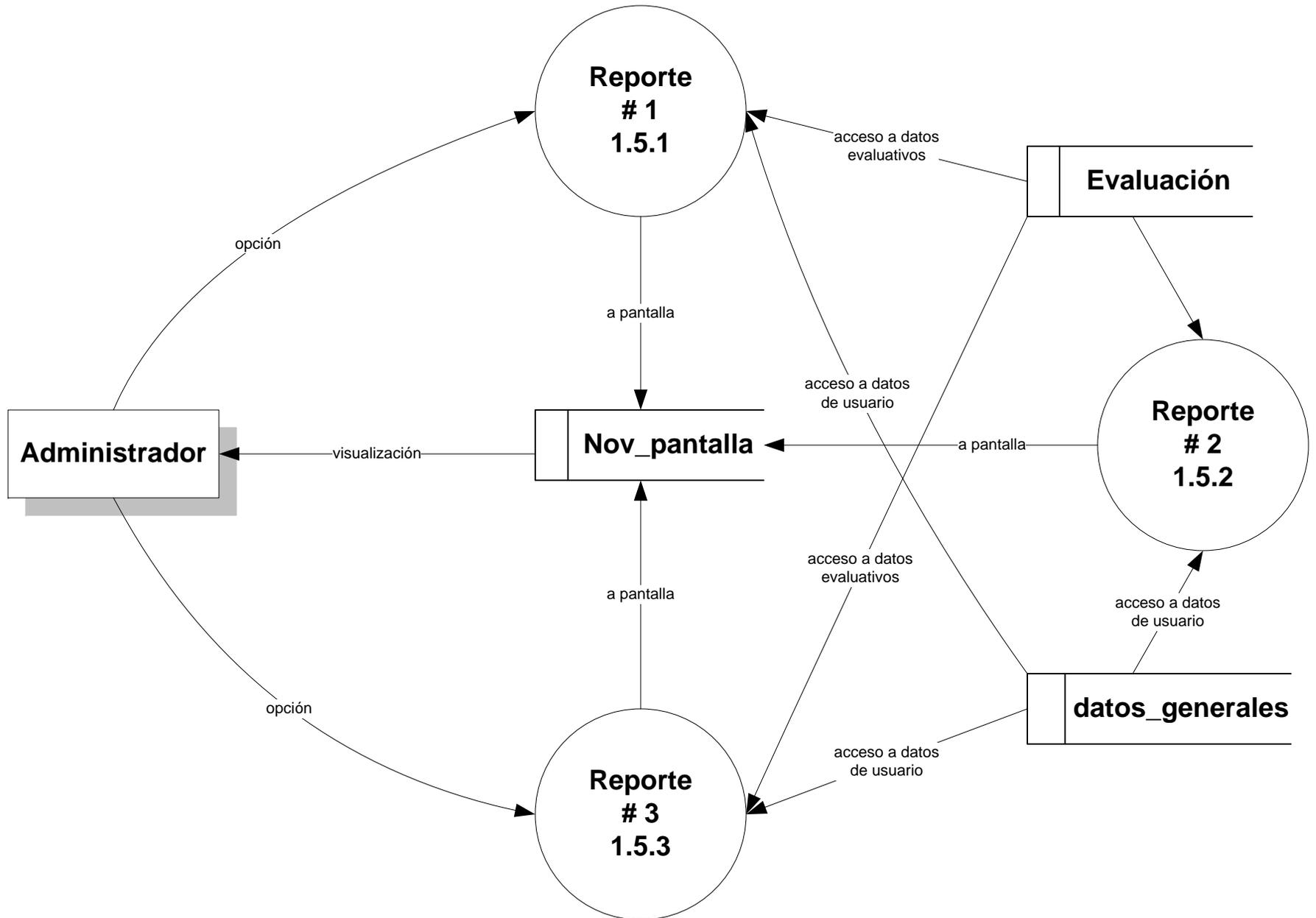
Nivel 2 (Aplicaciones)



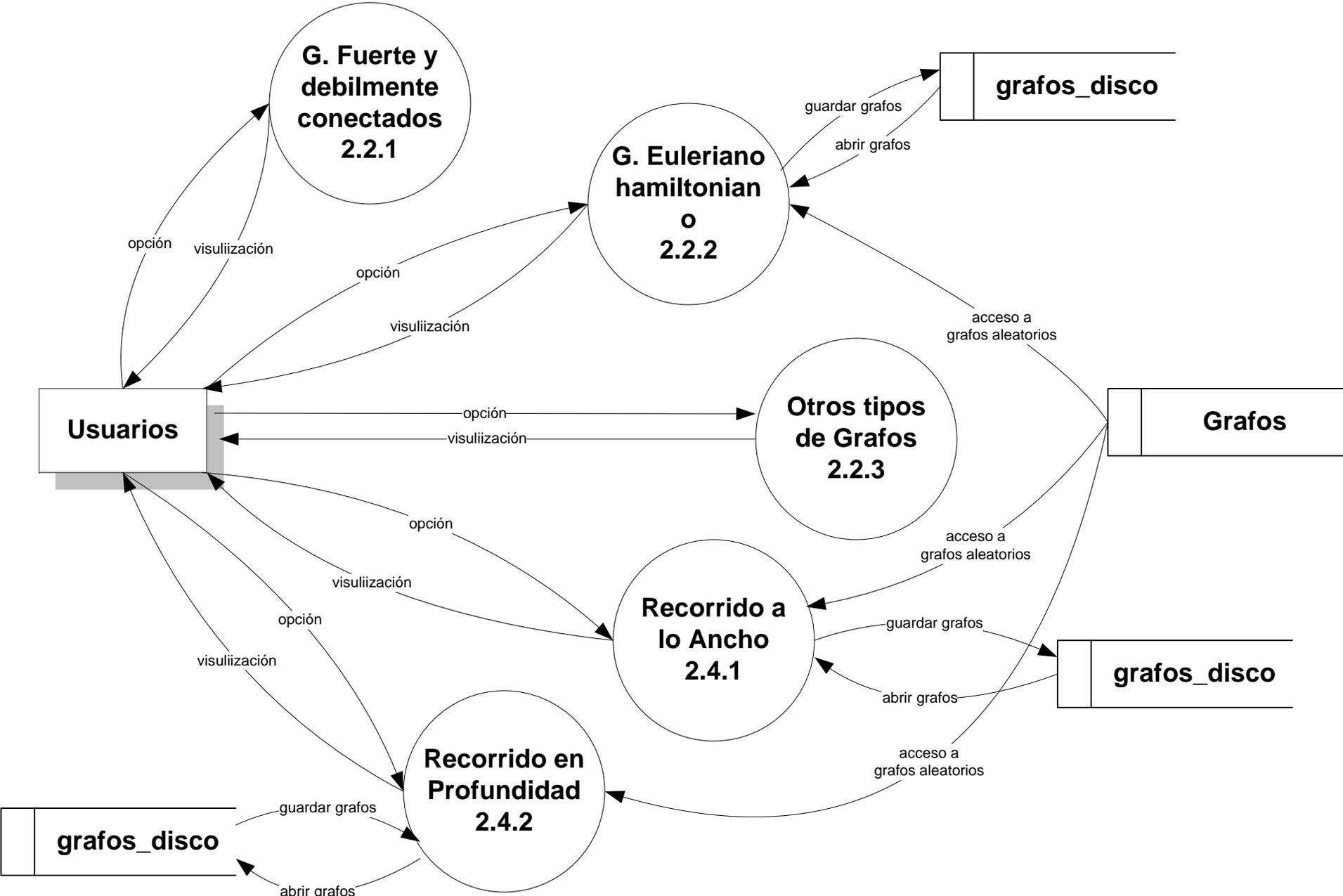
Nivel 2 (Evaluación y Sistema)



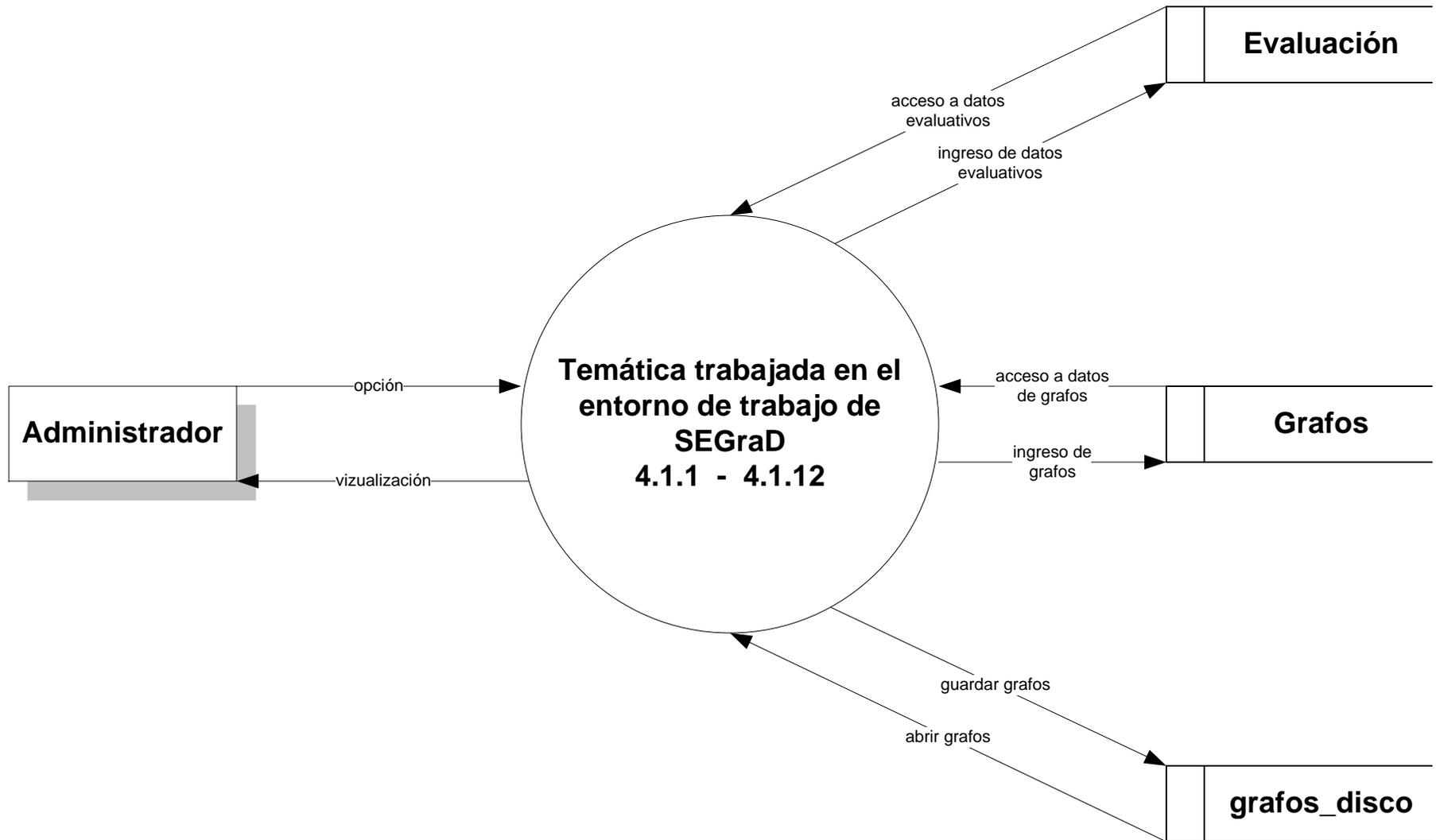
Nivel 3 (Reportes Generales)



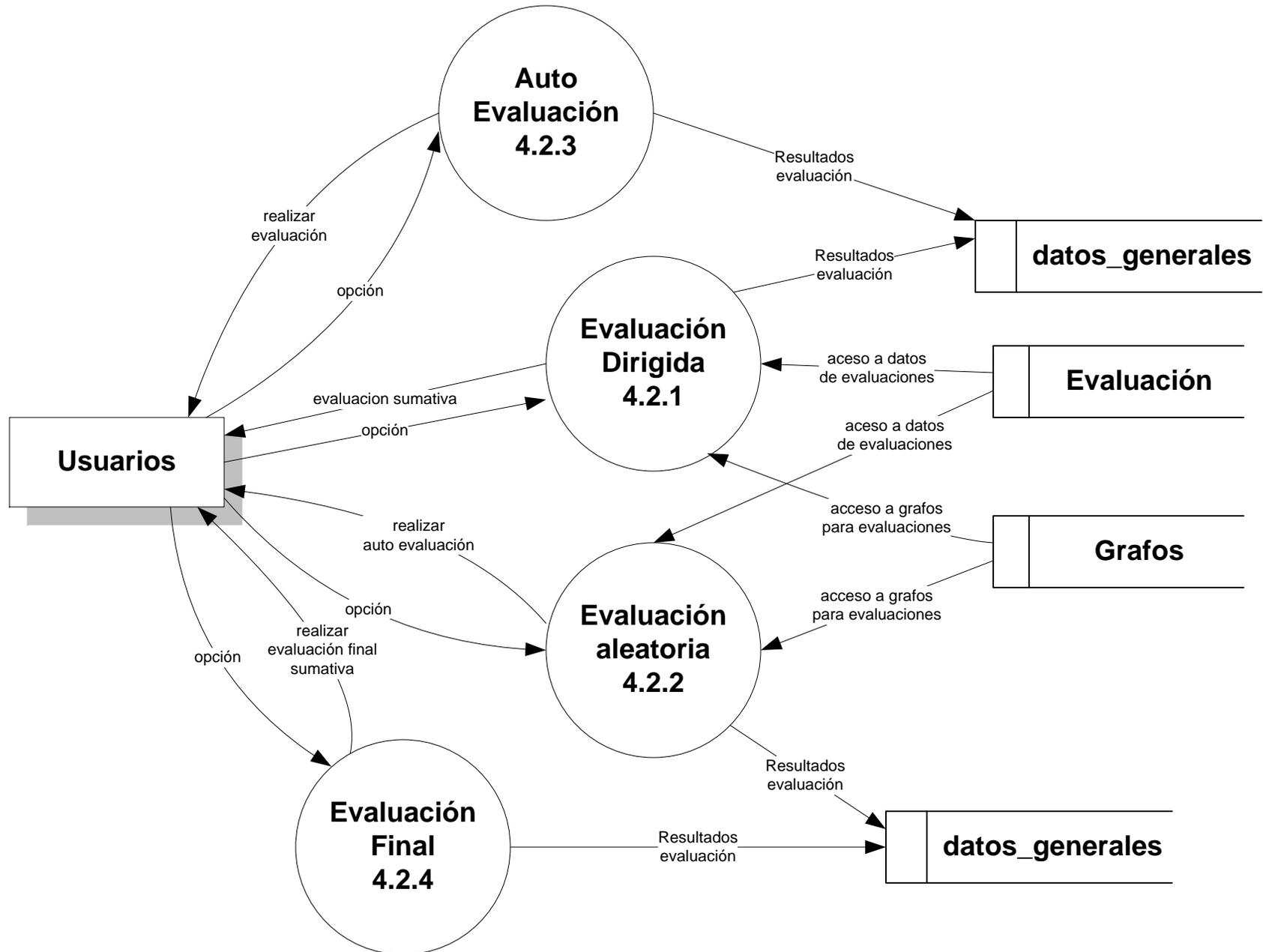
Nivel 3 (Tipos y Recorridos de Grafos)



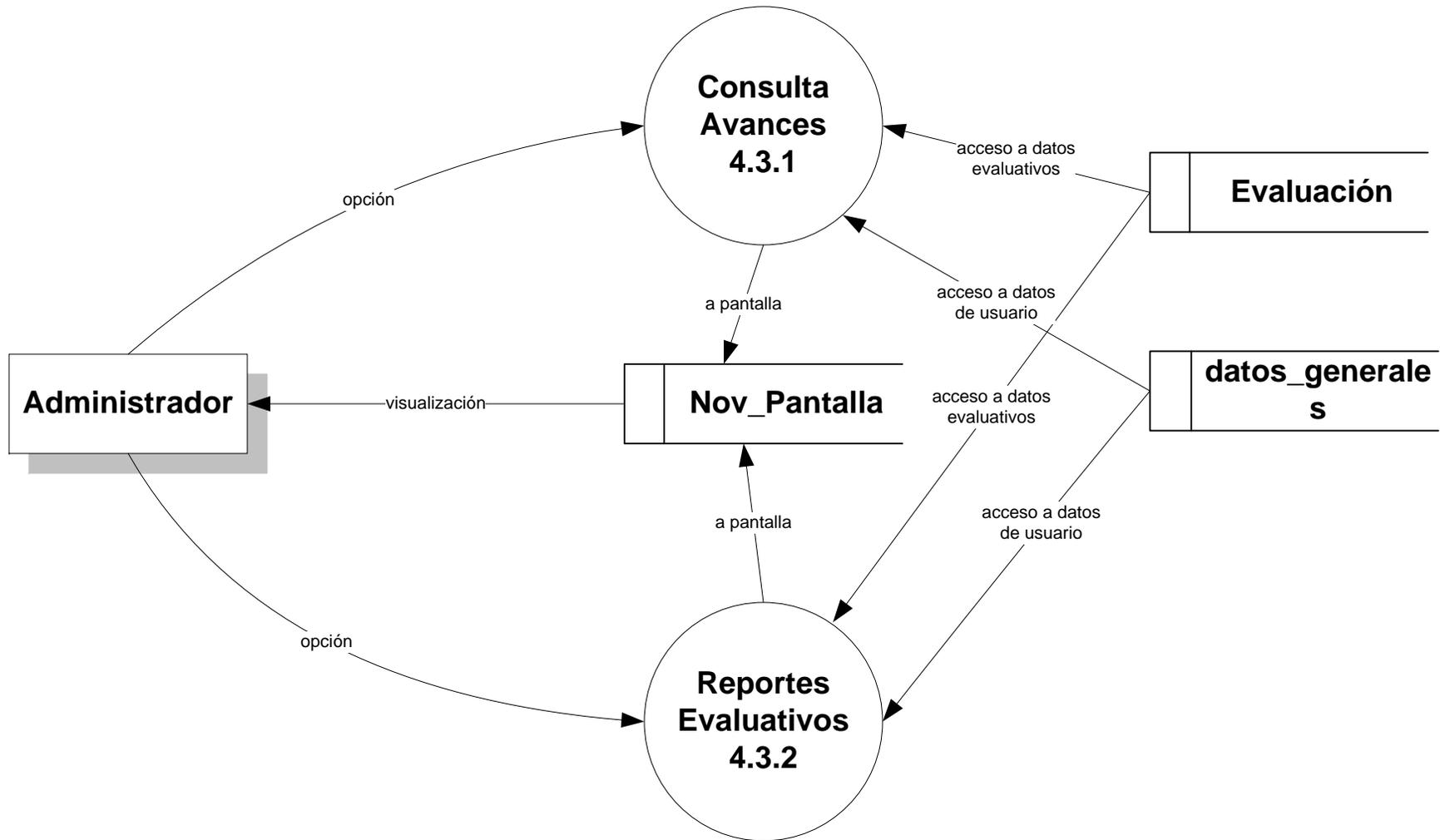
Nivel 3 (Elaborar Evaluación)



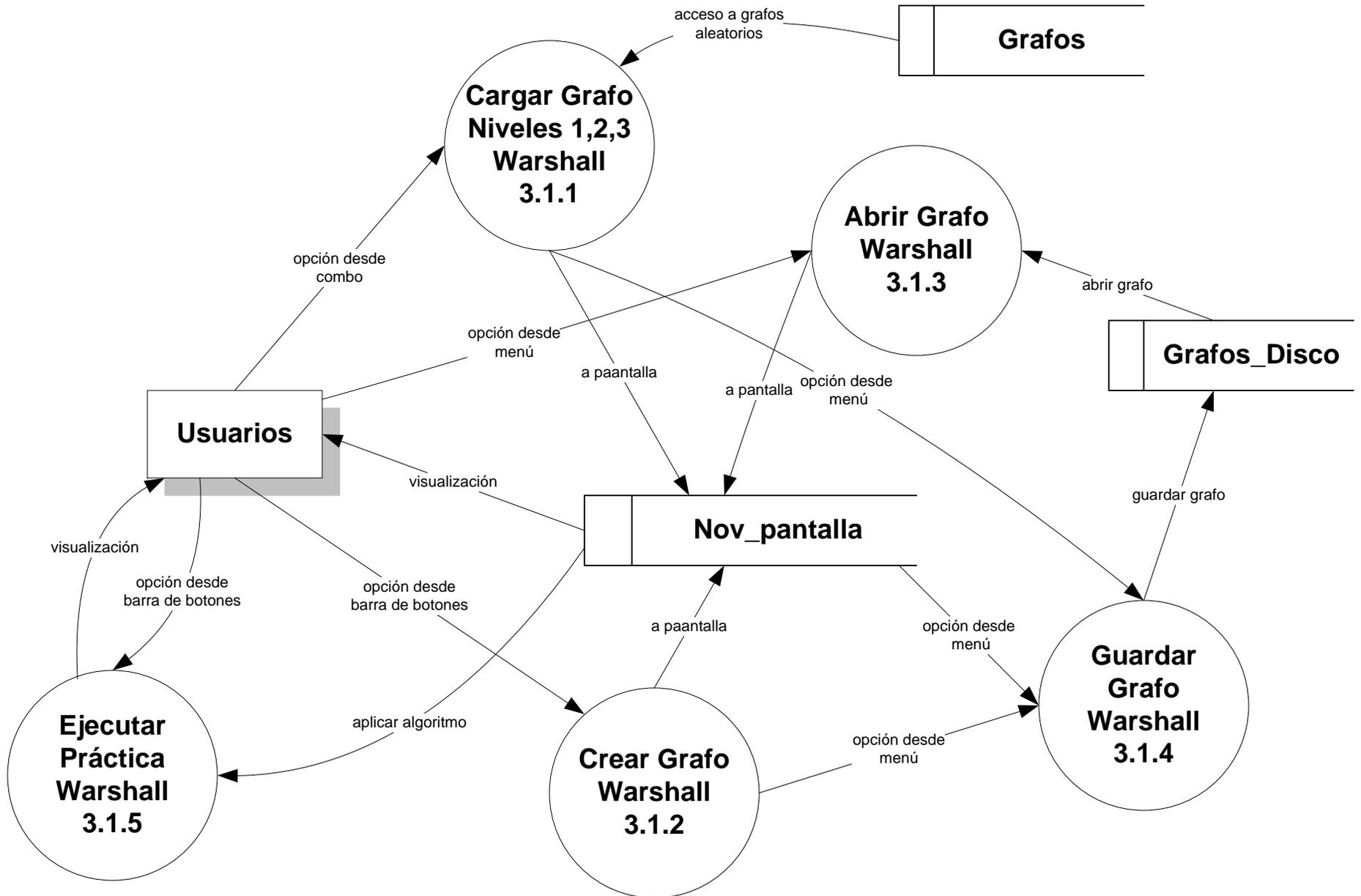
Nivel 3 (Realizar Evaluación)



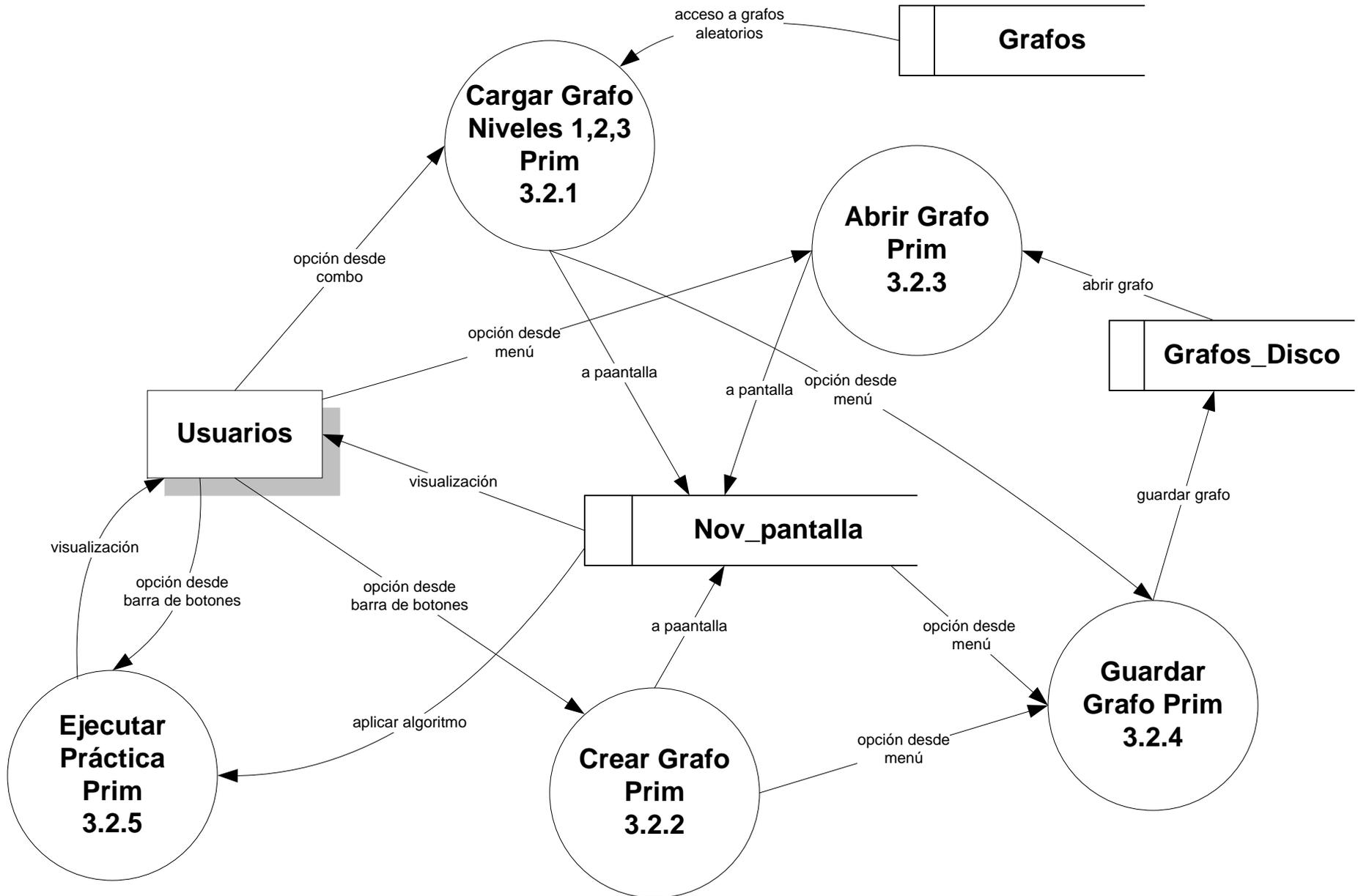
Nivel 3 (Mirar Resultados Evaluación)



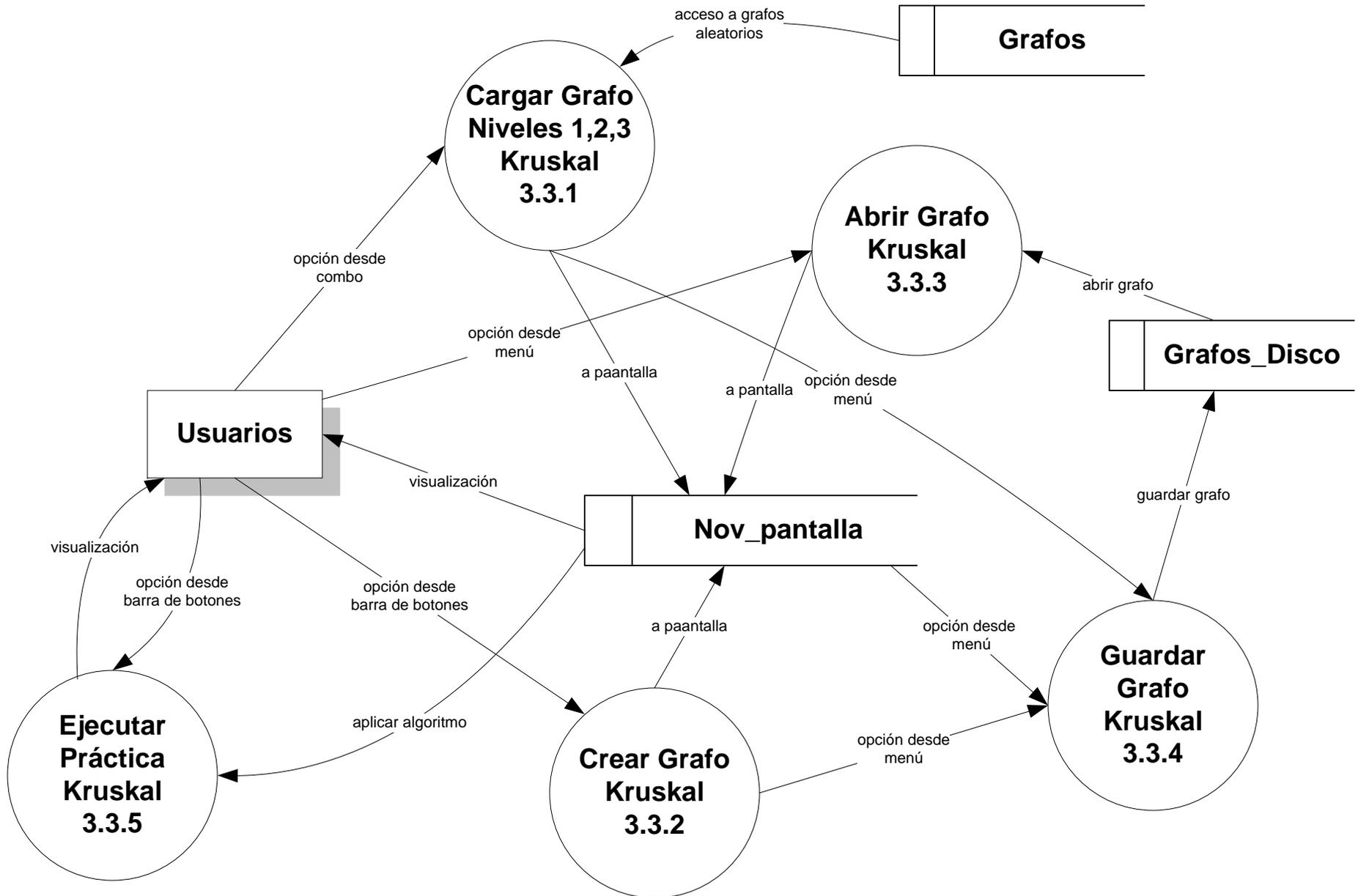
Nivel 4 (Ejercitación Alg. Warshall)



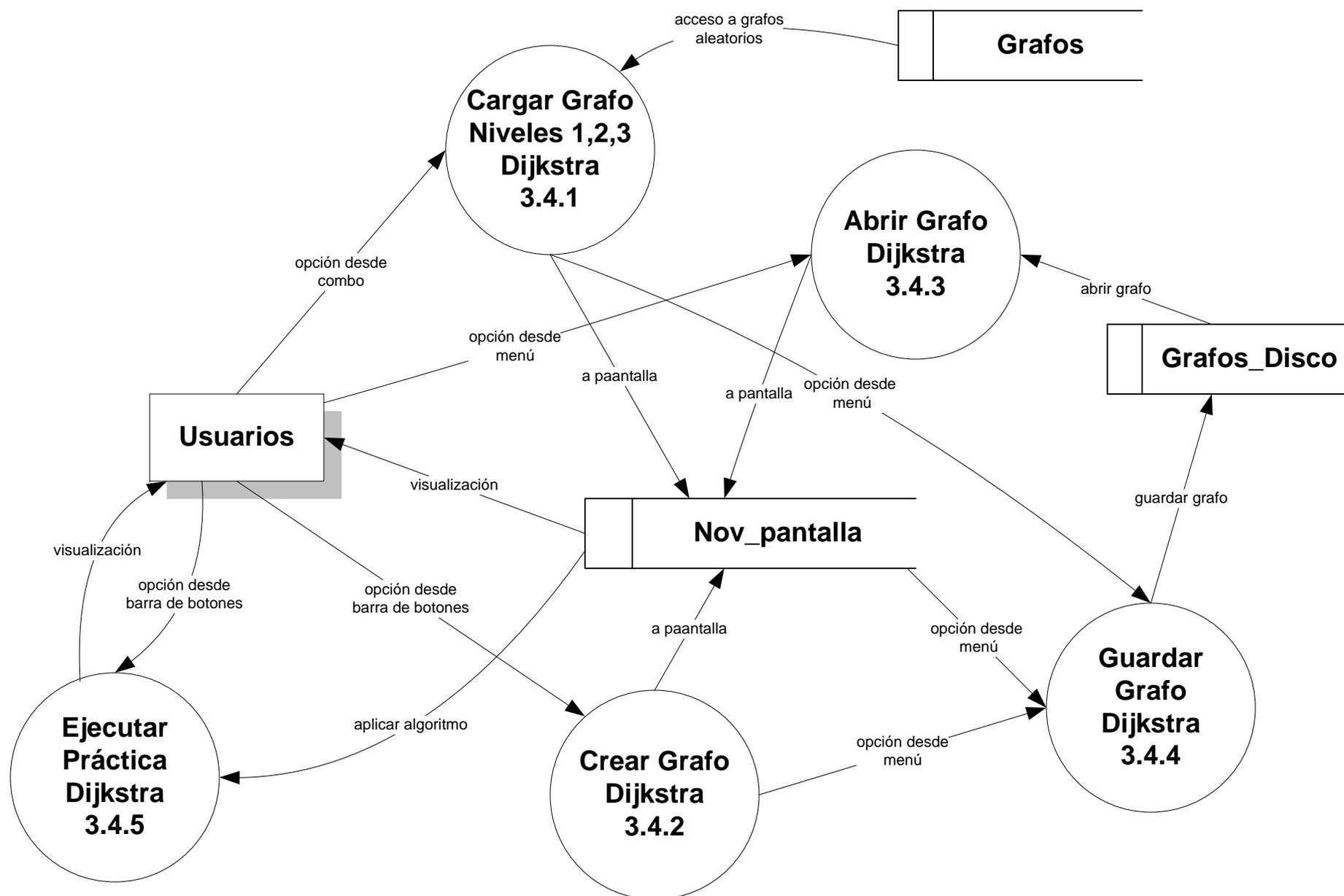
Nivel 4 (Ejercitación Alg. Prim)



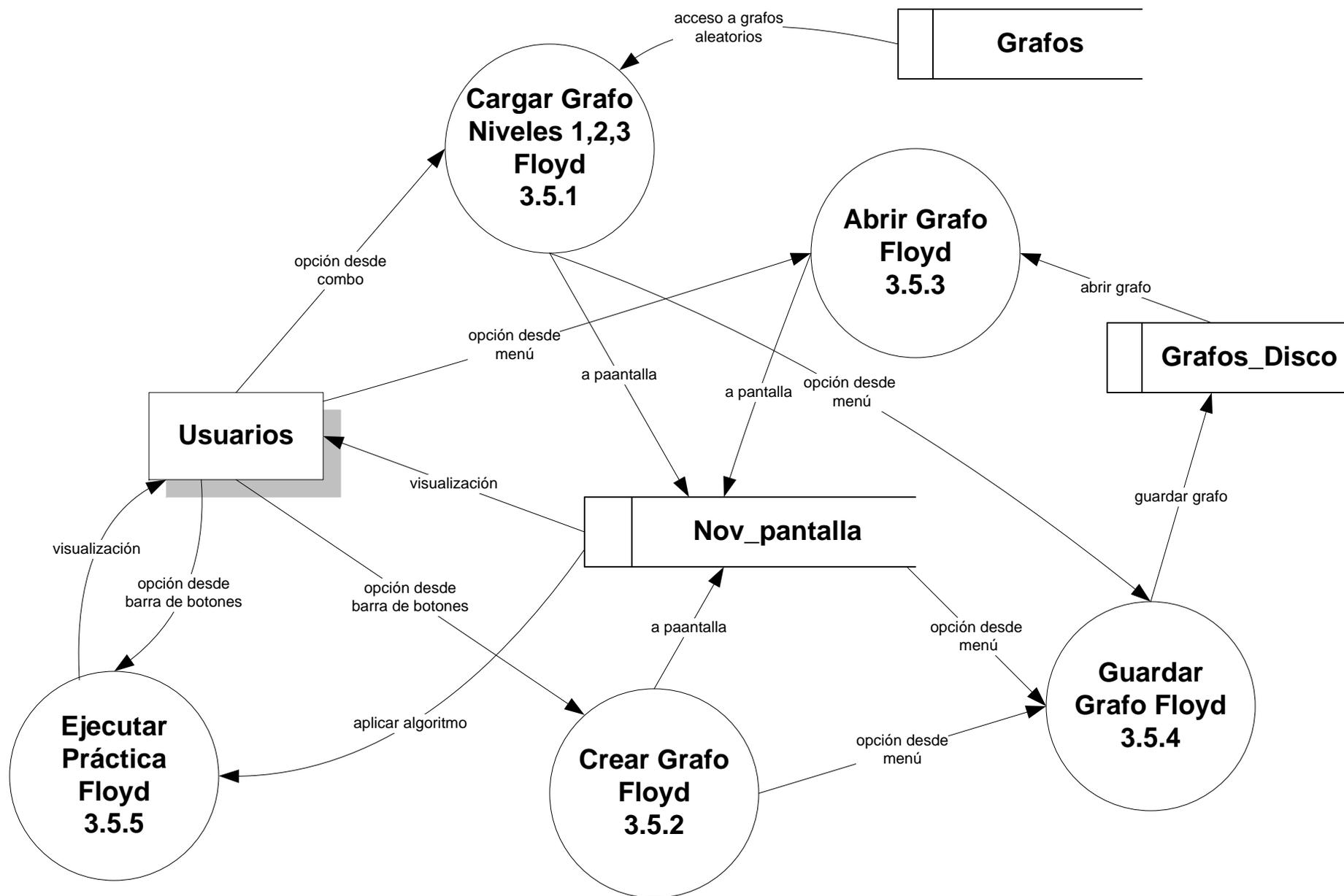
Nivel 4 (Ejercitación Alg. kruskal)



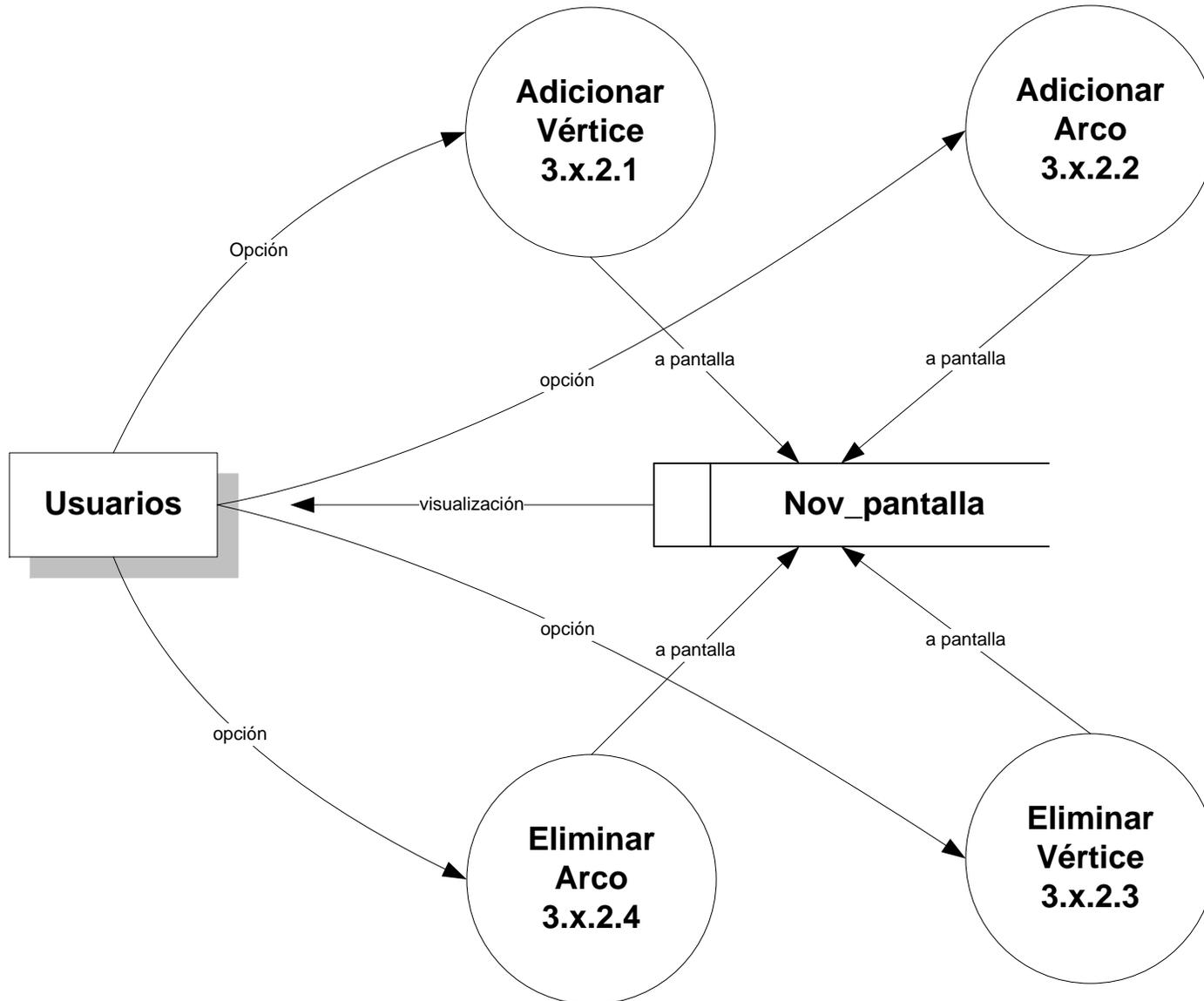
Nivel 4 (Ejercitación Alg. Dijkstra)



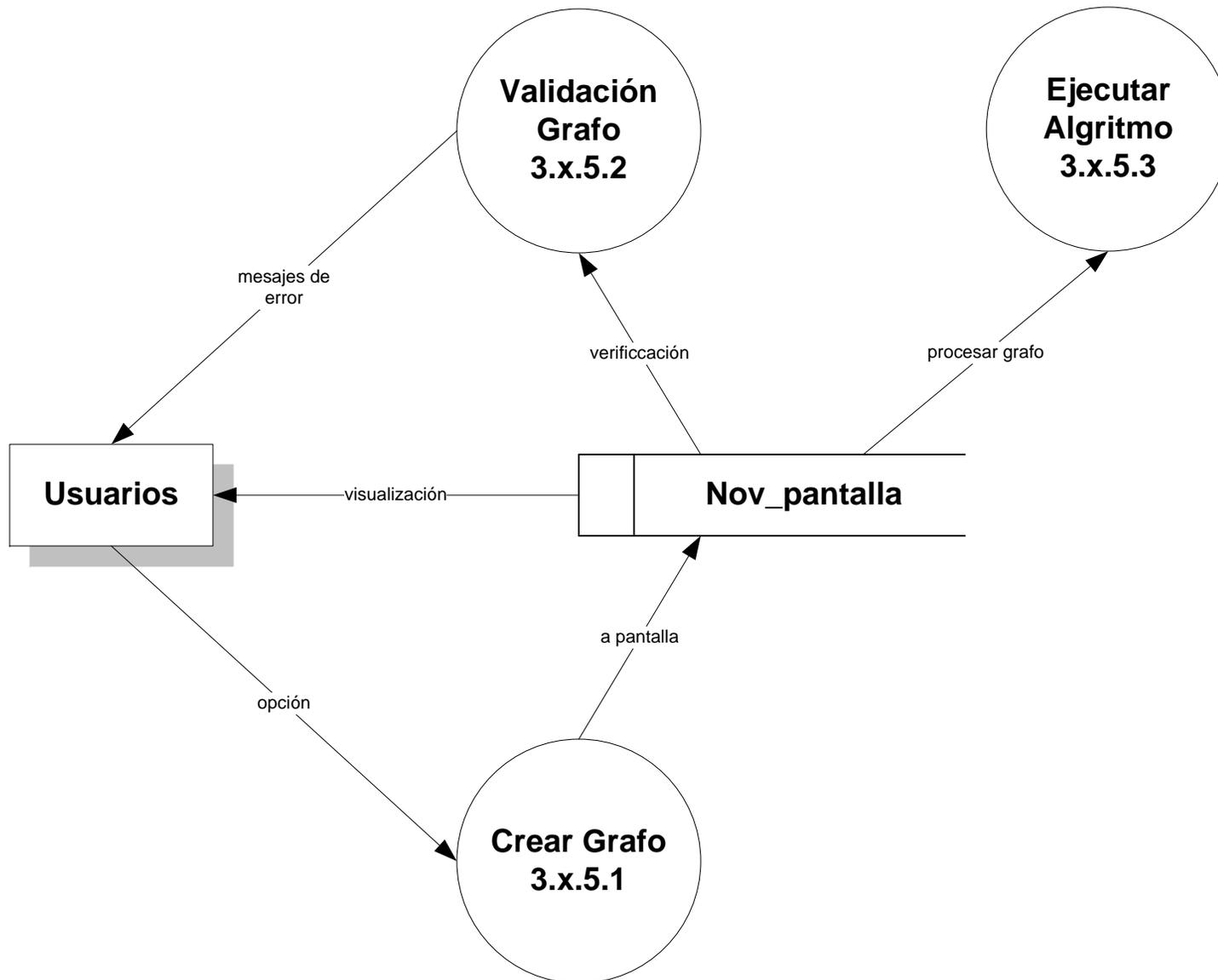
Nivel 4 (Ejercitación Alg. Floyd)



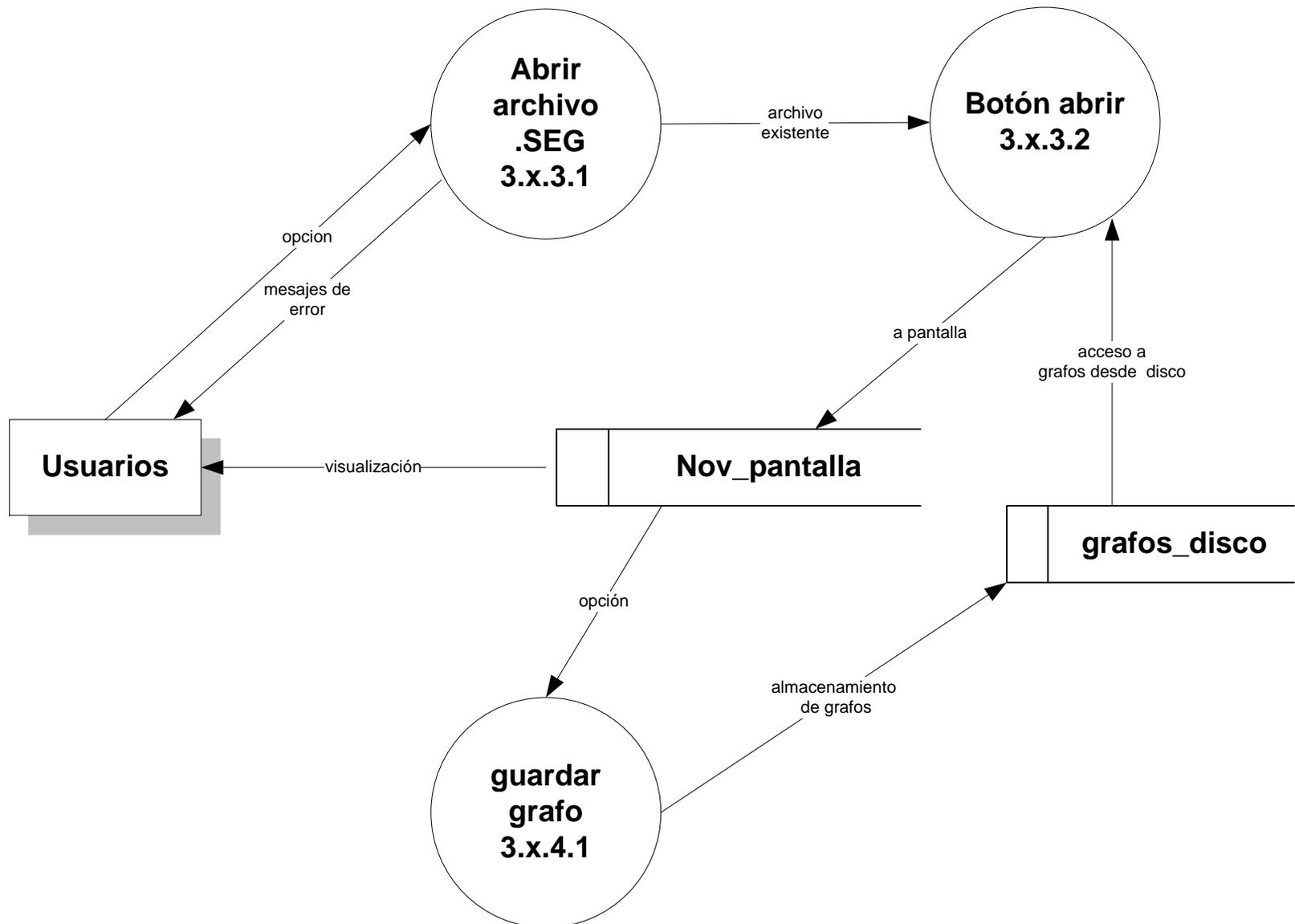
Nivel 4 (Crear Grafo)



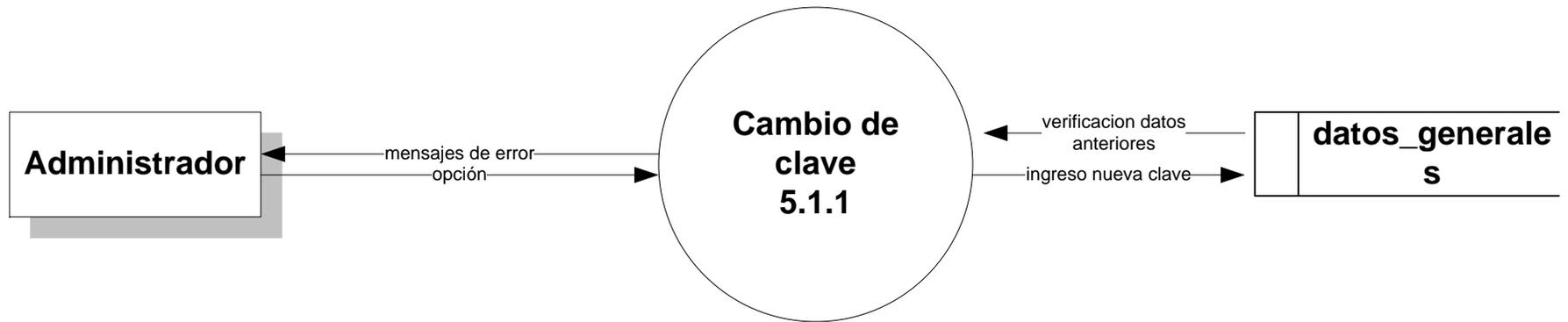
Nivel 4 (Ejecutar Práctica)



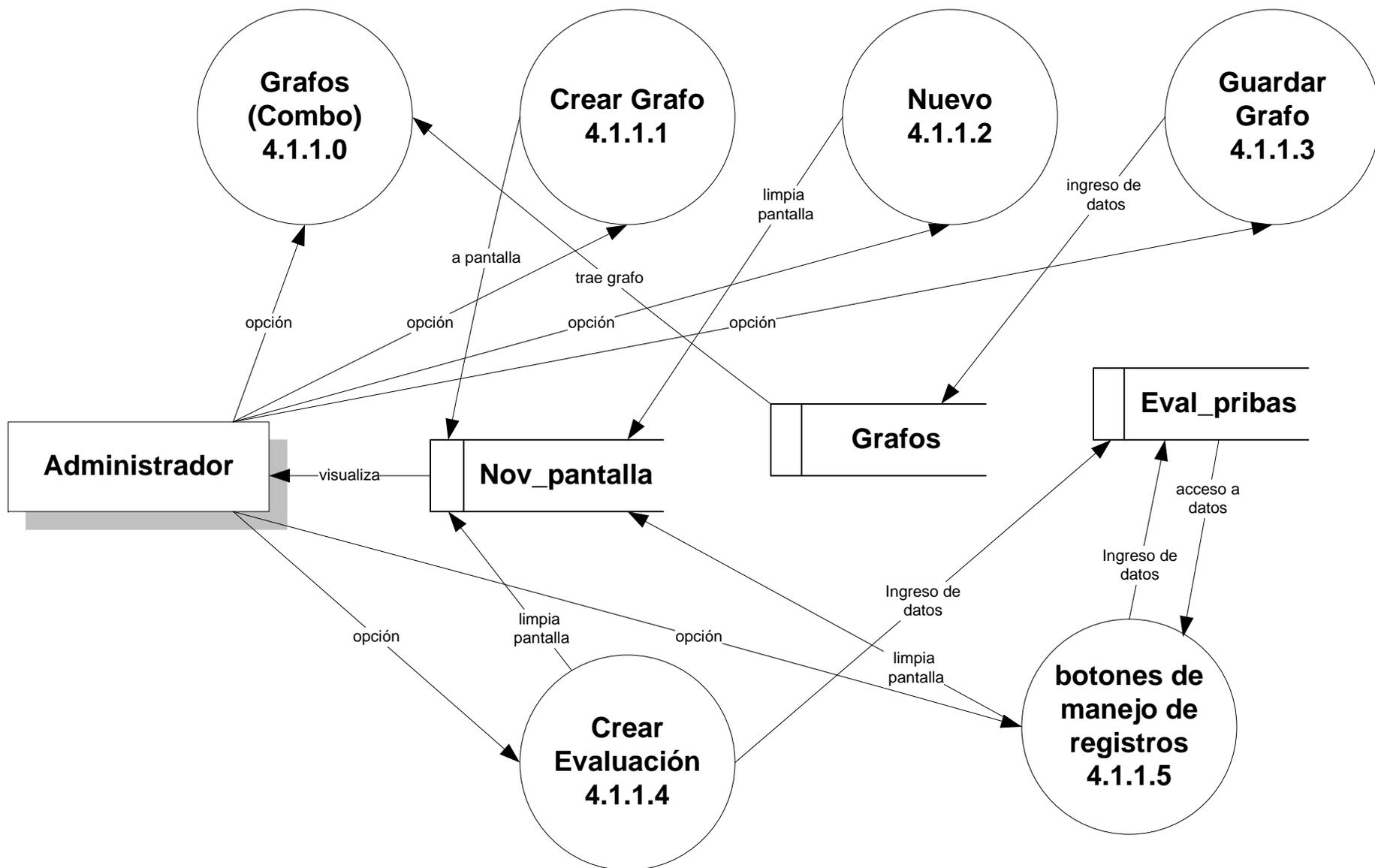
Nivel 4 (Abrir y Guardar Grafos)



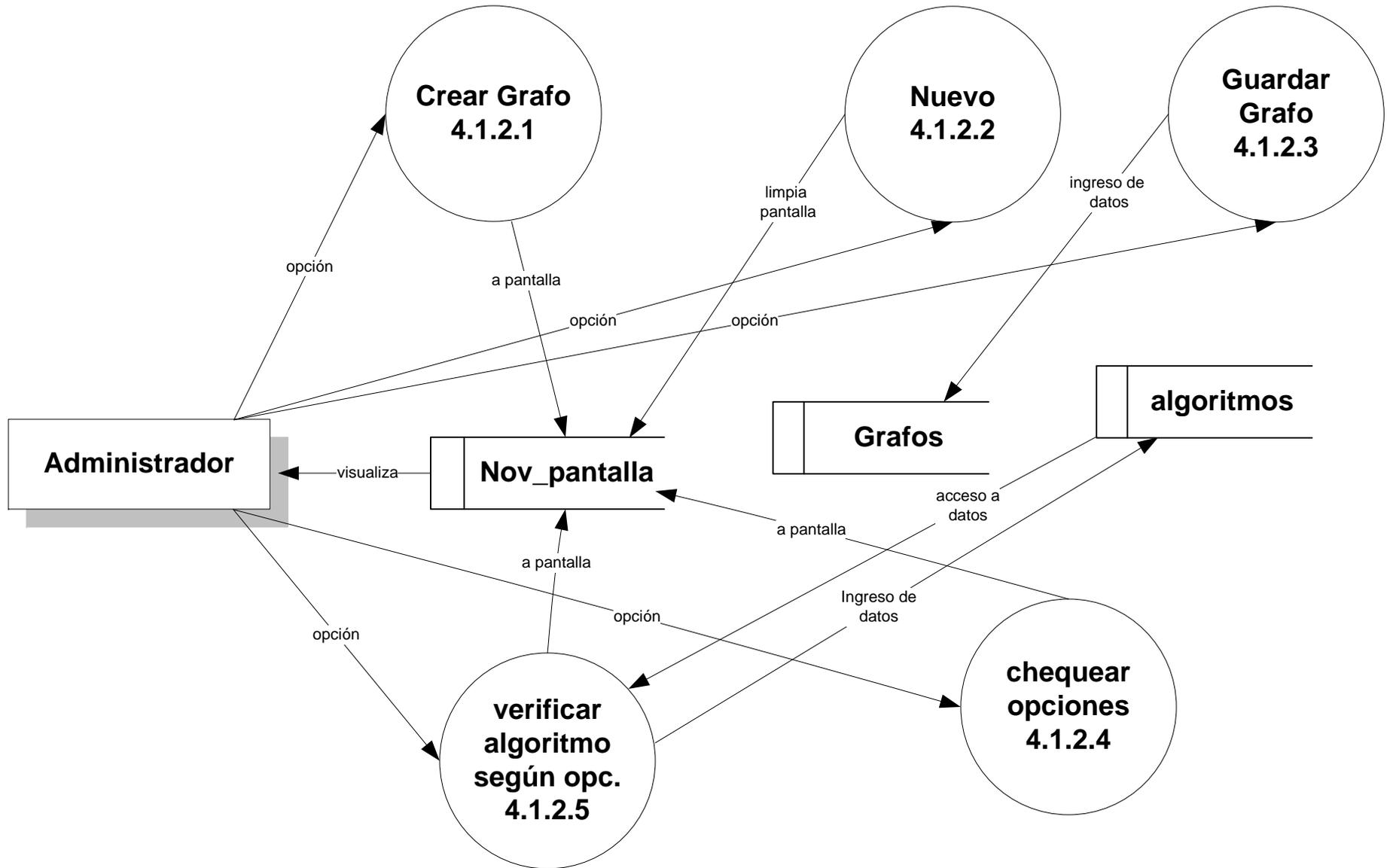
Nivel 3 (Cambio Clave)



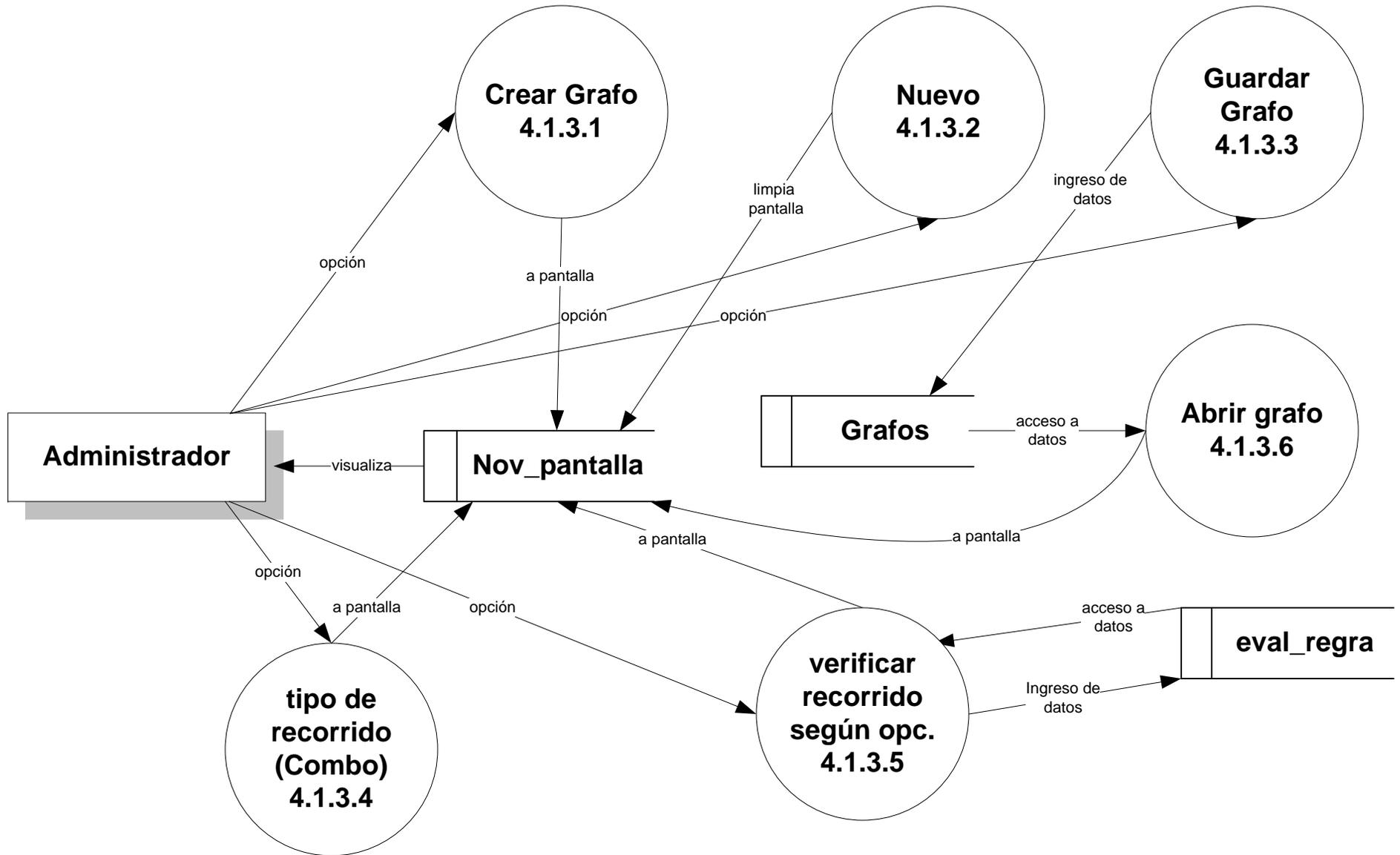
Nivel 4 (Evaluación Conceptos Básicos)



Nivel 4 (Evaluación Tipos de Grafos)



Nivel 4 (Evaluación Recorridos)



Nivel 4 (Evaluación para Aplicaciones)

