

**Universidad de Cienfuegos “Carlos Rafael Rodríguez”
Facultad de Informática**



**Tesis presentada en opción al grado de master en Nuevas
Tecnologías para la Educación**

TÍTULO

**Herramienta Informática de apoyo al PDE de la Disciplina
Matemática General en la Carrera Ingeniería Informática:
Matemática IV**

AUTOR: Ing. Orlando Tejeda Jiménez.

**TUTORES: Dr. Narciso Rubén de León Rodríguez.
Dr. Roberto Fuentes Garí.**

2010

Resumen

La inminente emigración a filosofía de software libre o código abierto que ocurre en nuestro país y en el Ministerio de Educación Superior en particular (MES) y la utilización en la actualidad del asistente matemático Derive 6, versión publicada en el 2003 y con licencia privativa, en el desarrollo del Proceso Docente Educativo (PDE) de la asignatura Matemática IV de la Disciplina Matemática General en la Carrera Ingeniería Informática de la Universidad de Cienfuegos, son los antecedentes fundamentales de la presente investigación.

La utilización del software privativo en la docencia tiene grandes desventajas en comparación con su contrapartida el software libre, entre las que se destacan la imposibilidad de reutilización del código de programación, las restricciones de copia y distribución, así como la obligatoriedad de licencia para su uso. Muchos son los asistentes matemáticos basados en filosofía de software libre que existen y pueden ser utilizados con fines docentes, después de estudiar algunos de ellos se decide usar Maxima con su interfaz gráfica para desarrollar la Herramienta Informática que, además del asistente matemático mencionado, está compuesta por: los métodos numéricos estudiados en Matemática IV programados en la herramienta, con lenguaje de programación Lisp, que brinda Maxima y una Página Web que estudiantes y profesores pueden utilizar como documento de consulta para trabajar con la asignatura.

Se demuestra la viabilidad del desarrollo de la Herramienta Informática, por brindar principalmente beneficios intangibles a la Universidad y por su ventaja de ser basada en filosofía de software libre en comparación con el Derive 6 (filosofía de software privativo).

Índice

Resumen	1
Índice	2
Introducción	1
1. Marco Teórico	5
1.1. Importancia de la Carrera Ingeniería Informática en la actualidad. Proyecciones.....	5
1.2. Tendencias actuales de la Educación Superior Cubana.	6
1.3. ¿Por qué estudiar Matemáticas en la Carrera Ingeniería Informática?	7
1.4. Matemática IV en la Carrera Ingeniería Informática.	8
1.5. Necesidad de que los estudiantes de informática se vinculen con sistemas informáticos que contribuyan al desarrollo de sus habilidades.....	10
1.6. Descripción de los asistentes matemáticos, basados en filosofía de software libre, disponibles.....	16
1.7. Asistente matemático, basado en filosofía de software libre seleccionado para desarrollar la Herramienta Informática.	23
2. Descripción y construcción de la solución propuesta.	25
2.1. Descripción de la solución propuesta.....	25
2.2. Principios básicos del análisis y diseño estructurado.....	26
2.3. Diagramas de flujo de datos (DFD).	27
2.4. Representación lógica en los procesos computacionales. Lenguaje natural estructurado (LNE).....	30
2.5. Diccionario de datos (DD)	40
2.6. Relaciones entre las herramientas del análisis y diseño estructurado.	73
3. Propuesta y Validación.....	74
3.1. Presentación de la solución propuesta.....	74
3.2. Validación de la Herramienta Informática	77
Conclusiones	82
Recomendaciones.....	83
Bibliografía.....	84

Introducción

Tecnologías de la Información y las Comunicaciones (TIC) es un término que ha sido definido de múltiples formas por diferentes autores, a continuación se presentan dos de esas definiciones.

Las TIC son “...un conjunto de aparatos, redes y servicios que se integran o se integrarán a la larga, en un sistema de información interconectado y complementario. La innovación tecnológica consiste en que se pierden las fronteras entre un medio de información y otro.” (Bartolomé A, 1995, pág. 15).

“Las TIC han sido conceptualizadas como la integración y convergencia de la computación microelectrónica, las telecomunicaciones y la técnica para el procesamiento de datos, sus principales componentes son: el factor humano, los contenidos de la información, el equipamiento, la infraestructura material, el software y los mecanismos de intercambio electrónico de información, los elementos de política y regulaciones y los recursos financieros” (Reinier. Pérez, 2005, pág. 51). Podemos clasificar el componente software, tomando en cuenta la licencia bajo la cual se distribuyen, en: libres o privativos.

Software libre: permite ejecutar el programa con cualquier propósito; acceder al código fuente para estudiar su funcionamiento y adaptarlo a las necesidades de cada uno; redistribuir copias; mejorar el programa y luego publicarlo para el bien de toda la comunidad (Richard M. Stallman, 2004, pág. 45).

Investigaciones realizadas [PDF “Matemáticas y Software libre para la docencia en la Universidad de Cádiz”. 8 de junio de 2005], [“Fundamentación del uso de software libre en la universidad pública. Enseñando Matemática con herramientas alternativas”. Octubre de 2007] apoyan la idea de utilizar software libre en la docencia, debido a sus ventajas técnicas, morales y pedagógicas por sobre sus contrapartes privativas.

Software privativo: trae consigo varias desventajas. Una es su licenciamiento, que impide distribuirlo, lo cual conlleva a la piratería. Otra, la más importante desde la perspectiva del educador, es que no es posible estudiar cómo funciona y por tanto no es posible modificarlo para hacer herramientas propias. Aquí es donde entra el software libre. No sólo simboliza un ahorro económico, también representa el retorno del control hacia los usuarios por permitir hacer lo que se quiera con el software: estudiarlo, modificarlo, mejorarlo y fundamentalmente compartirlo. Esto

último permite formar valores éticos que son importantes fomentar en los estudiantes universitarios.

En 2005, Cuba inició su "emigración a software libre y plataformas de código abierto en general". Desde entonces quedó orientado que esta "emigración" fuera un proceso continuo y organizado, y se crea el Grupo Ejecutivo Nacional encabezado por el Ministro de Informática y las Comunicaciones (Christian Peralta Guzmán, 2009, pág. 3). En la actualidad varias son las organizaciones involucradas en este proceso de migración y entre ellos el Ministerio de Educación Superior (MES).

En la carrera Ingeniería Informática en particular, es muy importante que los estudiantes utilicen software libre que les permitan acceder al código fuente, para estudiarlo, poder hacer modificaciones, crear sus propios algoritmos y hasta tener la posibilidad de publicar sus mejoras. Todo esto unido al proceso de emigración a software libre en que se encuentra el Ministerio de Educación Superior (MES), hacen necesario buscar alternativas *libres* para los sistemas *privativos* que los estudiantes utilizan en las diferentes asignaturas.

En la planificación de la asignatura Matemática IV de la Disciplina Matemática General de la Carrera Ingeniería Informática, puede ser de gran apoyo la utilización de una Herramienta Informática basada en la filosofía de software libre, de esta forma los estudiantes pueden desarrollar nuevos algoritmos a partir de los realizados con anterioridad como queda explícitamente planteado en los objetivos educativos del Plan de estudios D de la asignatura: Contribuir a la formación computacional de los estudiantes mediante la utilización de asistentes matemáticos, la creación de algunos programas y el desarrollo de la capacidad de algoritmizar.(MES, 2007, pág. 117)

En la actualidad no están dadas las condiciones para utilizar la filosofía de software libre en el desarrollo del PDE de la asignatura Matemática IV, por ello se considera oportuno definir el siguiente problema científico.

Problema Científico

Necesidad de tener una Herramienta Informática basada en filosofía de software libre para apoyar el PDE de la asignatura Matemática IV de la Disciplina Matemática General en la Carrera Ingeniería Informática.

Objeto de Investigación

El proceso docente-educativo (PDE) de la Disciplina Matemática General en la Carrera Ingeniería Informática, así como los asistentes matemáticos basados en filosofía de software libre que ya existan y permitan desarrollar la Herramienta Informática para esta disciplina.

Campo de Acción

La asignatura Matemática IV de la Disciplina Matemática General en la Carrera Ingeniería Informática.

Objetivo General de la Investigación

Desarrollar una Herramienta Informática basada en filosofía de software libre para apoyar el PDE de la asignatura Matemática IV de la Disciplina Matemática General en la Carrera Ingeniería Informática.

Objetivos Específicos

- ✓ Analizar el modelo del profesional de Ingeniería Informática (Plan de estudios D)
- ✓ Analizar el programa de la Disciplina Matemática General en la Carrera Ingeniería Informática.
- ✓ Analizar el programa de la asignatura Matemática IV.
- ✓ Analizar los asistentes matemáticos basados en filosofía de software libre que ya existan y permitan desarrollar la Herramienta Informática.
- ✓ Diseñar e implementar la Herramienta Informática.
- ✓ Validar la Herramienta Informática.

Hipótesis

Si se utiliza una Herramienta Informática, basada en filosofía de software libre, de apoyo a la asignatura Matemática IV de la Disciplina Matemática General en la Carrera Ingeniería Informática, entonces se favorece el cumplimiento de los objetivos de esta asignatura.

Tareas Científicas

- ✓ Sintetizar aspectos relacionados con:

- El programa de la asignatura Matemática IV de la Disciplina Matemática General en la Carrera Ingeniería Informática.
 - Asistentes matemáticos, basados en filosofía de software libre, existentes que permitan desarrollar la Herramienta Informática.
- ✓ Identificar los requerimientos.
 - ✓ Preparar la Herramienta Informática.
 - ✓ Documentar la investigación.

Métodos y técnicas utilizadas en el proceso de investigación.

Métodos teóricos.

El método de inducción-deducción para garantizar un resultado de calidad en la herramienta informática a proponer.

El método de análisis histórico-lógico para resumir y precisar la información recopilada sobre los asistentes matemáticos basados en filosofía de software libre que ya existen.

Los métodos de análisis y síntesis para conocer los objetivos del PDE de la asignatura Matemática IV en la Carrera Ingeniería Informática planteados en Plan D y la influencia de la utilización de asistentes matemáticos en la misma.

Métodos empíricos de investigación.

El método de revisión de documentos se utiliza para analizar los documentos normativos relacionados con la carrera Ingeniería Informática, principalmente el Plan de estudios D.

Aporte Práctico

El aporte práctico de esta investigación se concreta en el desarrollo de una Herramienta Informática, basada en filosofía de software libre, para apoyar el PDE de la asignatura Matemática IV de la Disciplina Matemática General en la Carrera Ingeniería Informática.

1. Marco Teórico

1.1. Importancia de la Carrera Ingeniería Informática en la actualidad. Proyecciones.

El avance de la ciencia y la tecnología de la computación y las comunicaciones es irrefutable, no solo a nivel mundial, sino también en la sociedad y la economía cubana en las que la informática ha continuado introduciéndose y desarrollándose vertiginosamente.

En la Resolución Económica del V Congreso del Partido Comunista de Cuba, de octubre de 1997, queda declarada explícitamente la importancia actual de la informática y la industria del software al expresarse:

- En particular las industrias ligera, de materiales de construcción, otras ramas de la industria básica, las actividades industriales de diferentes organismos, y sideromecánica y electrónica, incluida la del software, deberán trabajar con calidad, por incrementar la sustitución de importaciones, y desarrollar la exportación de líneas de elevada eficiencia.
- El país debe encaminarse resueltamente a la modernización informática mediante un programa integral que involucre a las organizaciones que deben proveer los recursos materiales, financieros e intelectuales y a las entidades económicas, políticas y sociales que deben traducirlos en más y mejores productos y servicios. (MES, 2007, pág. 14)

De ahí la importancia de la Carrera Ingeniería Informática, en la cual a partir del curso 2002-2003 se produjo un importante cambio cuantitativo y cualitativo en el escenario de formación de sus egresados. En este nuevo escenario hay tres formas diferentes de cursar el plan de estudio:

- El Curso Regular Diurno (CRD) que se estudia en el ISPJAE y otras provincias del país.
- El plan de Continuidad de Estudios para maestros de Computación Básica de primaria que se cursa en las Sedes Universitarias Municipales.
- El plan de la Universidad de las Ciencias Informáticas (UCI), con régimen de internado para estudiantes procedentes de todo el país, que posee condiciones

productivas que posibilitan muy alta dedicación al estudio y al trabajo. (MES, 2007, pág. 16)

Debido a lo anterior y estimulado, por otra parte, en las proyecciones del país basadas en las exportaciones de alto valor agregado, resultado de “producciones intelectuales”, en especial la biotecnología, la industria médico-farmacéutica, la informática y otros, que igualmente tienen un importante impacto social (Carlos Cañedo, 2008, pág. 57); constituyen razones importantes que justifican la necesidad de una nueva concepción curricular que ya se está evidenciando con la aprobación por el MES del plan de estudios D en julio de 2007.

1.2. Tendencias actuales de la Educación Superior Cubana.

La Informatización de la Sociedad Cubana ha provocado profundas transformaciones en los métodos de enseñanza en los diferentes niveles, implicando cambios importantes en los roles tradicionales del profesor y el estudiante. En la educación superior es esta una de las razones que justifica la necesidad de una nueva concepción curricular.

Entre las tendencias de la educación superior cubana está la disminución de la actividad presencial de los estudiantes, a partir de la introducción de nuevos métodos en el proceso de formación que centran su atención principal en el autoaprendizaje de los estudiantes, y entre los cuales desempeña un importante papel las tecnologías de la información y las comunicaciones (TIC). (Carlos Cañedo, 2008, pág. 24)

“Como consecuencia de un amplio y generalizado empleo de las TIC se producen importantes transformaciones en el proceso de formación, que han de expresarse fundamentalmente en el cambio en los roles de profesores y estudiantes y en el empleo de métodos más colaborativos, que permitan mayor asincronismo en el proceso de formación. En relación con ello se debe atender prioritariamente aspectos tales como la utilización de software profesionales, el empleo generalizado de plataformas interactivas de trabajo que fortalezcan el autoaprendizaje de los estudiantes y la introducción de prácticas de laboratorios virtuales”. (Carlos Cañedo, 2008, pág. 35)

En la Carrera Ingeniería Informática también son evidentes estas tendencias de la educación superior cubana y se recogen en el Plan de estudios D como por ejemplo

la de incrementar el estudio independiente, asistido por las Tecnologías de la Información y las Comunicaciones (TIC). (MES, 2007, pág. 16)

En el caso de la disciplina Matemática Básica para la Carrera Ingeniería Informática también están presentes estas tendencias y así está planteado en sus Indicaciones Metodológicas y de Organización del Plan D.

“Es necesario diseñar adecuadamente el modo de trabajo de la asignatura, el sistema de evaluación y el sistema de trabajo independiente. Se debe tender a:

- Incrementar el uso de los medios de cómputo, la utilización de bibliotecas y la creación de programas sencillos.
- Plantear problemas al estudiante en los que, de modo independiente, debe realizarse la creación, adaptación e interpretación de un modelo, su solución por computadora, el análisis de los errores y la validación del resultado.”(MES, 2007, pág. 115)

1.3. ¿Por qué estudiar Matemáticas en la Carrera Ingeniería Informática?

Desde el primer año de la carrera se evidencia la importancia de las matemáticas en la formación del ingeniero informático ya que en los objetivos instructivos de ese año se plantean, entre otros:

- Desarrollar el pensamiento abstracto, el análisis lógico y el razonamiento inductivo y deductivo.
- Adquirir habilidades elementales en la interpretación de gráficos y otras formas de representación tridimensional.(MES, 2007, pág. 19)

La disciplina Matemática General, es aquella en que se desarrollan los fundamentos de la formación de un especialista en Ingeniería Informática, dado que todo ingeniero considera representaciones técnicas y científicas en términos matemáticos con los cuales reflejan los rasgos cuantitativos de los fenómenos que estudia. De tal modo, el objetivo de esta disciplina es lograr que el ingeniero informático domine el aparato matemático que lo haga capaz de modelar y analizar los procesos técnicos, económicos, productivos y científicos, utilizando en ello, tanto métodos analíticos como aproximados y haciendo uso eficiente de las técnicas de cómputo.

El propósito de la enseñanza de la disciplina Matemática consiste en adiestrar a los estudiantes en la utilización de los distintos métodos analíticos y aproximados, en el uso de asistentes matemáticos y en la implementación de esquemas de cálculo en máquinas computadoras, desarrollando así su pensamiento lógico, heurístico y algorítmico. (MES, 2007, pág. 105)

Contribución general que hace el estudio de las matemáticas a la formación del ingeniero informático:

- Amplía la madurez matemática y la capacidad de trabajo con la abstracción.
- Desarrolla habilidades para la comunicación de propiedades y características de magnitudes en forma gráfica, numérica, simbólica y verbal.
- Contribuye a conformar una cultura científica general e integral actualizada, que toma en cuenta:
 - El uso de la computación en la resolución de problemas,
 - El procesamiento de literatura técnica,
 - El manejo de lenguaje interdisciplinario. (MES, 2007, pág. 107)

La disciplina Matemática General de la Carrera Ingeniería Informática está conformada por cinco asignaturas semestre, tres en primer año (Álgebra, Matemática I, Matemática II) y dos en segundo (Matemática III, Matemática IV). La computación debe estar presente en cada una de las asignaturas que componen la disciplina (MES, 2007, pág. 114), vinculada esencialmente a los tópicos siguientes:

- Derivación
- Integración
- Ecuaciones diferenciales ordinarias
- Series
- Métodos numéricos en general.

1.4. Matemática IV en la Carrera Ingeniería Informática.

La asignatura Matemática IV forma parte de la Disciplina Matemática General de la Carrera Ingeniería Informática. Se imparte en el segundo semestre del segundo año. Se puede resumir como la asignatura que trata el cálculo numérico.

Organización por temas.

- Tema 1: Solución numérica de ecuaciones.
- Tema 2: Métodos numéricos del Algebra Lineal.
- Tema 3: Aproximación de funciones.
- Tema 4: Derivación e integración numérica.
- Tema 5: Métodos numéricos en ecuaciones diferenciales.
- Tema 6: Optimización de funciones.

Métodos Numéricos.

- Tema 1: Bisección, Regula_Falsi, Newton_Rapson, Whittaker y Secante.
- Tema 2: Jacobi y Gauss_Seidel.
- Tema 3: Ajuste con el método de los mínimos cuadrados e Interpolación con los métodos de Lagrange y Newton.
- Tema 4: Derivación con los métodos Adelante, Atrás y Central e Integración con los métodos Simpson y Trapecios.
- Tema 5: Euler, Runge_Kutta primero, segundo, tercero y cuarto orden.
- Tema 6: Coordenada a Coordenada y Gradiente.

Módulo Bibliográfico.

- Matemática Numérica, Álvarez, M., Cuba, Editorial Pueblo y Educación.
- Cálculo con Geometría Analítica, Segunda edición, Swokowski, Earlw, México, Grupo Editorial Iberoamérica, 1989.
- Ecuaciones Diferenciales, Segunda edición, Zill, Dennis G, México, Grupo Editorial Iberoamérica, 1988.
- Matemática numérica, Suárez Alonso, Margarita, Cuba, Editorial Pueblo y Educación.
- Métodos Numéricos en ecuaciones diferenciales y ajuste de curvas, Tamayo Fajardo, Manuel H. y Borges Trenard, Miguel A, Cuba, Editorial Pueblo y Educación.

El Análisis Numérico es un campo muy amplio de la Matemática. Se le conoce indistintamente como Matemática Numérica, Cálculo Numérico, Matemática Computacional ó Métodos de Cómputo. Se puede definir como: “la teoría y la práctica del cálculo eficiente y la estimación del error de la solución aproximada de

muchos problemas de aplicación de la Matemática. Una de las diferencias primarias entre la Matemática y la Matemática Numérica es que la primera carece del concepto de eficiencia. Luego la Matemática Numérica tiene entre sus objetivos la elección del procedimiento más adecuado y su conveniente aplicación para la solución de un problema particular” (S Benedeto, 200?, pág. 49).

1.5. Necesidad de que los estudiantes de informática se vinculen con sistemas informáticos que contribuyan al desarrollo de sus habilidades.

Los estudiantes de informática deben vincularse con diferentes sistemas informáticos que contribuyan tanto al desarrollo de sus habilidades, como a la formación de una cultura informática que les permita valorar el funcionamiento de los sistemas y adquirir experiencia.

En particular la utilización de software que permitan reutilizar el código y modificarlo es mucho mejor para su formación profesional.

En las diferentes asignaturas de la carrera deben utilizarse software de aplicación específicos del perfil del graduado, ya sean paquetes profesionales o desarrollados por el claustro de profesores u otros especialistas del país. Estos sistemas informáticos deben ser usados por los estudiantes como apoyo en su labor docente e investigativa. (MES, 2007, pág. 51)

Todas las disciplinas de la carrera tienen que colaborar en la formación general del egresado, en las diferentes dimensiones (MES, 2007, pág. 50). Es por ello que en las asignaturas básicas como matemática hay que buscar alternativas para lograr un mayor vínculo con la especialidad y así una mejor contribución a la formación integral de los estudiantes.

Asistentes Matemáticos

Los mediadores, intérpretes o intermediarios entre el ordenador y las ciencias de las matemáticas son aquellos programas desarrollados con este fin, es lo que se conoce comúnmente como asistentes matemáticos o herramientas matemáticas. Estos se han desarrollado para apoyar el proceso docente educativo en el ámbito educacional y el procesamiento matemático de modelos, en el ámbito de proyectos reales (investigativo).

Un programa debe cumplir determinados requerimientos o requisitos para poder ser usado con éxito como herramienta matemática, sobre todo si se enmarca en el ámbito de educación superior. Entre los requerimientos fundamentales deben señalarse los de cálculos científicos (requerimiento mínimo), cálculos simbólicos y cálculos numéricos y matricial (estos dos últimos requerimientos en función de las especificidades de cada herramienta) (J. Rafael Rodríguez Galván, 2005, pág. 5)

Existen múltiples asistentes matemáticos (desarrollados unos con fines académicos y otros para apoyar problemas matemáticos) que cumplen con estos requisitos y cuentan con muchas otras funcionalidades. Tomando en cuenta la licencia bajo la cual se distribuyen, estos pueden clasificarse en dos grandes grupos: asistentes matemáticos propietarios y los basados en filosofía de software libre.

Importancia de utilizar asistentes matemáticos, basados en filosofía de software libre, en la docencia.

Software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. De modo más preciso, se refiere a cuatro libertades de los usuarios del software:

- La libertad de usar el programa, con cualquier propósito (libertad 0).
- La libertad de estudiar cómo funciona el programa, y adaptarlo a tus necesidades (libertad 1). El acceso al código fuente es una condición previa para esto.
- La libertad de distribuir copias (libertad 2).
- La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. (libertad 3). El acceso al código fuente es un requisito previo para esto.

Un programa es software libre si los usuarios tienen la libertad de distribuir copias, sea con o sin modificaciones, sea gratis o cobrando una cantidad por la distribución, a cualquiera y hacia cualquier lugar. El ser libre de hacer esto significa entre otras cosas que no se tiene que pedir o pagar permisos.

También se debe tener la libertad de hacer modificaciones y utilizarlas de manera privada en tu trabajo u ocio, sin ni siquiera tener que anunciar que dichas

modificaciones existen. Si se publican los cambios, no se tiene por qué avisar a nadie en particular.

La libertad para usar un programa significa la libertad para cualquier persona u organización de usarlo en cualquier tipo de sistema informático, para cualquier clase de trabajo, y sin tener obligación de comunicárselo al desarrollador o a alguna otra entidad específica.

La libertad de distribuir copias debe incluir tanto las formas binarias o ejecutables del programa como su código fuente, sean versiones modificadas o sin modificar (distribuir programas de modo ejecutable es necesario para que los sistemas operativos libres sean fáciles de instalar). Está bien si no hay manera de producir un binario o ejecutable de un programa concreto (ya que algunos lenguajes no tienen esta capacidad), pero debe tenerse la libertad de distribuir estos formatos si se encontrara la manera de crearlos.

Para que las libertades de hacer modificaciones y de publicar versiones mejoradas tengan sentido, se debe tener acceso al código fuente del programa.

Para que estas libertades sean reales, deben ser irrevocables mientras no se haga nada incorrecto; si el desarrollador del software tiene el poder de revocar la licencia aunque no le hayas dado motivos, el software no es libre.

Son aceptables, sin embargo, ciertos tipos de reglas sobre la manera de distribuir software libre, mientras no entren en conflicto con las libertades centrales. Por ejemplo, copyleft, regla que implica que cuando se redistribuya el programa, no se pueden agregar restricciones para denegar a otras personas las libertades centrales. Esta regla no entra en conflicto con las libertades centrales, sino que más bien las protege.

"Software libre" no significa "no comercial". Un programa libre debe estar disponible para uso comercial, desarrollo comercial y distribución comercial. El desarrollo comercial del software libre ha dejado de ser inusual; el software comercial libre es muy importante.

Es aceptable que haya reglas acerca de cómo empaquetar una versión modificada, siempre que no bloqueen a consecuencia de ello la libertad de publicar versiones modificadas. Reglas como "Si se hace disponible el programa de esta manera, se debe hacer disponible también de esta otra" pueden ser igualmente aceptables, bajo la misma condición. También es aceptable que la licencia requiera que, si se ha distribuido una versión modificada y el desarrollador anterior pide una copia de ella, debe enviársele.

A veces las normas de control de exportación del gobierno y las sanciones mercantiles pueden restringir tu libertad de distribuir copias de los programas a nivel internacional. Los desarrolladores de software no tienen el poder de eliminar o sobrepasar estas restricciones, pero lo que pueden y deben hacer es rehusar el imponerlas como condiciones de uso del programa. De esta manera, las restricciones no afectarán actividades y gente fuera de las jurisdicciones de estos gobiernos.

Cuando se habla de software libre, es mejor evitar términos como: regalar o gratis, porque esos términos implican que lo importante es el precio, y no la libertad.

¿Por qué el software libre en las aulas?

La elección de los programas usados en el aula se guía por consideraciones que deberían ser objetivas, algunas de las cuales como la calidad del producto y su adecuación a los descriptores de la asignatura son, por supuesto, requisitos previos. Pero en la mayoría de los casos, existe más de un programa informático que cumple razonablemente bien los requisitos anteriores, por lo que, para la definitiva elección, es preciso recurrir a consideraciones adicionales.

Con frecuencia, estas consideraciones han estado relacionadas con la tradición y la inercia a usar las mismas herramientas de años anteriores dificultando la posibilidad de que los alumnos puedan utilizar y comparar diferentes herramientas en el aula. Pero la velocidad con que evoluciona la sociedad de la información puede hacer que tecnologías que hoy son hitos incuestionables sufran mañana severos cambios e incluso lleguen a ser superadas y olvidadas en cuestión de años. Por tanto una formación basada en la excesiva dependencia de una única herramienta comercial, puede llegar, con el tiempo, a ser obsoleta. Los estudiantes deberían estar formados en habilidades generales, en conocimiento neutral, y no en los productos

concretos de una marca comercial. Sólo de esta manera se garantizará el carácter universal de los conocimientos adquiridos y se evitará que la no disponibilidad de un producto o sus carencias evidencien las lagunas del proceso formativo.

El software con licencia libre, permite y fomenta el disponer de varias herramientas a la vez, complementarias o capaces de interactuar entre sí, cada una de las cuales contará con sus puntos fuertes y sus debilidades. Aunque el profesor se decante por una de ellas, siempre podrá ofrecer a sus alumnos la enriquecedora posibilidad de experimentar con otras, de resolver un mismo problema desde distintas perspectivas y de saciar su curiosidad a aquellos que cuenten con mayores inquietudes.

Conviene destacar, además, que la dependencia de una herramienta privativa en el ámbito educativo conlleva problemas éticos añadidos, puesto que de forma irremisible provoca en el alumnado la seducción por una marca cuyo precio hará que, en la mayoría de los casos, no pueda ser adquirida legalmente, incitando su copia ilegal. En el otro extremo se sitúa el caso del software libre permitiendo que el profesor comparta con sus alumnos, con toda legalidad, las herramientas utilizadas (quizás, acompañadas de material docente propio).

Por otra parte, en un nivel superior, el software libre ofrece la interesante posibilidad de que los alumnos puedan acceder al código fuente, a la forma en que está programada la herramienta que están utilizando, brindándoles la posibilidad de agregar nuevos módulos a la misma. Esto tiene, en matemáticas, el gran valor de poder estudiar la manera en la que un programa de primer nivel implementa en la práctica los algoritmos que han sido estudiados en las clases teórico-prácticas. El poder observar y modificar el código de un programa profesional, utilizado por miles de personas de todo el mundo, constituye una experiencia tremendamente gratificante, de gran valor docente, como refuerzo y motivación.

Y en último lugar, uno de los argumentos más importantes pero, con frecuencia, no suficientemente valorado, debido quizás al desconocimiento del software libre y a la asimilación social de los valores que conlleva el software privativo: impulsando el software en el aula y con él los valores éticos asociados, estaremos basando la educación en pilares como la libertad, el conocimiento, la solidaridad y la colaboración.

Investigaciones realizadas apoyan la idea de utilizar software libre en la docencia, debido a sus ventajas técnicas, morales y pedagógicas por sobre sus contrapartes privativas.

En junio de 2005 J. Rafael Rodríguez Galván profesor del Departamento de Matemáticas de la Universidad de Cádiz publicó: “Matemáticas y Software libre para la docencia en la Universidad de Cádiz “. En este artículo se concluye que:

- Una formación basada en la excesiva dependencia de una única herramienta comercial, puede llegar, con el tiempo, a ser obsoleta.
- La dependencia de una herramienta privativa en el ámbito educativo conlleva problemas éticos añadidos, puesto que de forma irremisible provoca en el alumnado la seducción por una marca cuyo precio hará que, en la mayoría de los casos, no pueda ser adquirida legalmente, incitando su copia ilegal.
- Al usar en el aula una herramienta con licencia libre, el profesor cuenta con ventajas adicionales a la hora de la planificación y el desarrollo de la asignatura, derivadas de tener la garantía de que un programa con software libre podrá ser instalado y usado por sus alumnos en cualquier lugar fuera de la universidad.
- El Software Libre posibilita que los alumnos puedan acceder al código fuente, a la forma en que está programada la herramienta que están utilizando en clase. Esto aporta el gran valor de poder estudiar la manera en la que un programa de primer nivel implementa en la práctica los algoritmos que han sido estudiados en las clases teórico-prácticas. El poder observar y modificar el código de un programa profesional, utilizado por miles de personas de todo el mundo, constituye una experiencia tremendamente gratificante, de gran valor docente, como refuerzo y motivación.
- Utilizar software libre en el aula impulsa valores éticos asociados en los cuales se basa la educación como: la libertad, el conocimiento, la solidaridad y la colaboración (J. Rafael Rodríguez Galván, 2005, págs. 4-5).

En octubre de 2007 A Pardini de la Universidad nacional de La Plata publicó: “Fundamentación del uso de software libre en la universidad pública. Enseñando Matemática con herramienta alternativas”. En este artículo se defiende la siguiente idea:

“El avance en la comunidad científica siempre fue acompañado por la libre circulación de las ideas y el conocimiento; entonces, ¿por qué educamos usando herramienta informáticas cuyo modelo de distribución no solo fomenta todo lo

contrario sino que tampoco nos permite estudiar su funcionamiento? El ámbito académico supuestamente tiene que favorecer la creatividad, innovación, aprendizaje, cooperación y libre flujo de ideas, sin embargo si enseñamos utilizando herramienta privativas estamos atentando contra estos ideales” (A. Pardini, 2007, pág. 1).

1.6. Descripción de los asistentes matemáticos, basados en filosofía de software libre, disponibles.

A continuación se describen algunas de las opciones más extendidas en el mundo del software matemático con licencia libre, que pudieran servir para desarrollar la Herramienta Informática.

No aparecen todos los asistentes matemáticos disponibles con licencia libre (sería prácticamente imposible, dado su cantidad), solo se describen las relacionadas con el objetivo de esta investigación.

MACAULAY2:

Macaulay2 es un sistema de algebra computacional orientado a la Geometría Algebraica y al Álgebra Conmutativa. Este software está disponible en código fuente para portarlo, y en forma compilada para Linux, Sun OS, Solaris, Windows. (“Macaulay - Wikipedia, la enciclopedia libre,” s.d.), (Hal Schenck, 2003, pág. 105)

SINGULAR:

Es un sistema de álgebra computacional (CAS) para cálculos polinómicos con énfasis especial en las necesidades del álgebra conmutativa, de la geometría algebraica, y de la teoría de singularidades. (“Singular-Wikipedia, la enciclopedia libre,” s.d.), (H. Schönemann Lossen, 2006, pág. 92)

Características:

- Principales objetos de cómputo: polinomios, ideales y módulos sobre una gran variedad de anillos.
- Gran variedad de algoritmos implementados, escrito en C/C++, o como bibliotecas.
- Uso de lenguaje de programación similar a C.
- Documentación extensa.

- Disponible para la mayoría de las plataformas: Unix (HP-UX, SunOS, Solaris, Linux, AIX), Windows, Macintosh.

MAXIMA:

El sistema de álgebra computacional **Maxima** es un motor de cálculo simbólico escrito en lenguaje Lisp publicado bajo licencia GNU GPL. Cuenta con un amplio conjunto de funciones para hacer manipulación simbólica de polinomios, matrices, funciones racionales, integración, derivación, series de Taylor, transformadas de Laplace, resolver ecuaciones diferenciales y sistemas, calcular límites, trabajar con vectores, listas y tensores manejo de gráficos en 2D y 3D, manejo de números de coma flotante muy grandes, expansión en series de potencias y de Fourier. Permite exportar los resultados a Tex y su funcionamiento está muy bien documentado.

Maxima es la continuación libre del proyecto Macsyma (MAC's SYmbolic MANipulator), iniciado en los 60 en el prestigioso MIT (Massachusetts Institute of Technology), este constituyó el primer proyecto con éxito en la automatización de operaciones matemáticas de tipo simbólico. En este sentido, puede considerarse el padre de todos los programas de cálculo simbólico actuales (en especial Mathcad y Maple). ("Maxima-Wikipedia, la enciclopedia libre," s.d.), (A. Pardini, 2007, pág. 3)

Maxima posee varias interfaces: **xMaxima**, **wxMaxima**.

Hay además, interfaces Web, que permiten trabajar desde cualquier sitio con acceso a internet. Como por ejemplo: WebMathematics Interactive. (J. Rafael Rodríguez Galván, 2005, pág. 8)

Características:

- Cuenta con el apoyo de una comunidad de desarrollo muy dinámica, que trabaja constantemente para su actualización, mejora y soporte.
- Disponible para distintas plataformas, tales como GNU/Linux, MS Windows, etc.
- Tiene comandos propios para la representación de gráficos 2D y 3D que, por defecto, utiliza el programa gnuplot.
- Cuenta con manuales y documentación que pueden descargarse libremente de internet.
- Incorpora un completo lenguaje de programación propio, derivado del Máxima original.

- Interfaz con usuario: por defecto, utiliza una interfaz poco amigable (de tipo consola de texto). Pero existen programas específicamente diseñados para facilitar su uso como xMaxima o wxMaxima.

wxMaxima:

wxMaxima es una interfaz que pretende facilitar el uso de Maxima, a través de una serie de menús, barras de botones y ventanas asociadas. (J. Rafael Rodríguez Galván, 2007, págs. 21-22)

Características:

- Existen versiones tanto para sistemas operativos GNU/Linux como para Windows.
- Integra distintos documentos de ayuda de Maxima de forma que éstos son fácilmente accesibles a través del ratón.
- Introducción de matrices, creación de gráficas, cálculo de límites, de derivadas o integrales, etc. Son numerosos los conceptos que cuentan con ventanas específicas gracias a las cuales se simplifica el uso de Maxima.
- Para las personas que prefieren usar los comandos de Maxima, existe una entrada de diálogo que permite introducirlos, acceder y editar los comandos anteriores, etc.

xMaxima:

xMaxima es una interfaz gráfica que se incluye por defecto con el programa Maxima y que tiene una filosofía similar a wxMaxima. Históricamente, ha sido la interfaz gráfica "oficial" de Maxima y en algunos sistemas como MSWindows es la que se utiliza por defecto.

Sin embargo, presenta algunas desventajas con respecto a wxMaxima:

- Su interfaz gráfica, basada en las librerías TCL/TK, es antigua y tiene menos prestaciones.
- No cuenta con ventanas y menús que puedan servir como apoyo para la introducción de comandos en Maxima.

OCTAVE:

Es un programa de código libre (realizado en C++) que constituye una avanzada herramienta matemática para realizar complejos algoritmos y cálculos numéricos.

Proporciona una conveniente interfaz de línea de comandos para resolver numéricamente problemas lineales y no lineales, y para realizar otros experimentos numéricos utilizando un lenguaje que es compatible con el de Matlab.

MATLAB es considerado su equivalente comercial. Entre varias características que comparten se puede destacar que ambos ofrecen un intérprete permitiendo ejecutar órdenes en modo interactivo. Octave no es un sistema de álgebra computacional como podría ser Maxima, sino que usa un lenguaje que está orientado al análisis numérico. (John W. Eaton, 2008, pág. 10)

GNU Octave es un software de distribución libre.

Características:

- Octave está escrito en C++ usando la librería STL.
- Tiene un intérprete de su propio lenguaje y permite una ejecución interactiva o por lotes.
- Puede extenderse el lenguaje con funciones y procedimientos por medios de módulos dinámicos.
- Utiliza otros programas GNU para ofrecer al usuario crear gráficos para luego imprimirlos o guardarlos (Grace).
- Dentro del lenguaje también se comporta como una consola de órdenes (shell). Esto permite listar contenidos de directorios.

- Además de correr en plataformas Unix también lo hace en Windows.
- Puede cargar archivos con funciones de Matlab de extensión *.m*.

El lenguaje Octave

- Es un lenguaje interpretado.
- No permite pasar argumentos por referencia. Siempre se pasan por valor.
- No permite punteros.
- Se pueden generar scripts.
- Soporta gran parte de las funciones de la librería estándar de C.
- Puede extenderse para ofrecer compatibilidad a las llamadas al sistema UNIX.
- El lenguaje está pensado para trabajar con matrices y provee mucha funcionalidad para trabajar con éstas.
- No es un lenguaje de programación orientado a objetos. Por lo tanto, no tiene clases ni objetos.

EULER: Es un programa de cálculo numérico y matricial al estilo de Matlab y Octave aunque no es compatible con ellos. Es un programa sencillo y flexible con una interfaz bastante cómoda tanto en sus versiones para GNU/Linux como MS Windows("Euler," s.d.). Permite:

- Realizar buenos gráficos en 2D y 3D.
- Programar en un moderno lenguaje de programación, soporta valores por defecto para parámetros y cantidad de parámetros variables.
- Incluye las funciones básicas requeridas para este tipo de programas: Integración y diferenciación numérica, estadística, interpolación, transformada de Fourier, ecuaciones diferenciales, etc.

GiNaC: Es una biblioteca en C++ para realizar cálculos simbólicos. El programa ginsh posee una interfaz iterativa que permite trabajar con la librería GiNaC a través de un interprete de comandos. Ginsh tiene aplicaciones directas a la docencia y puede usar la librería GiNaC como motor de cálculo en un programa en C++ permitiendo manipular algebraicamente datos matemáticos por ejemplo funciones indicando valores iniciales o de contorno, leídos de un fichero o introducidos iterativamente por el usuario.

Scilab: Es un excelente programa con cierta compatibilidad con Matlab, con una interfaz integrada que puede resultar agradable para los amantes de los entornos de ventanas aunque la versión de GNU/Linux resulta menos vistosas por la galerías gráficas que usa. Cuenta con módulos para la programación distribuida y gráfica (con el ratón) y posibilidad de exportar a FOLTRAN.

Nota: presenta un inconveniente, no cumple con Debian's Free Software Guidelines por lo que es considerado non free, aún siendo gratuito y estando su código disponible para su uso y estudio.

R: es un entorno y un potente lenguaje de cálculo estadístico, similar al programa S (con licencia privativa) y en gran parte compatible con él.

Entre sus características se encuentra además el contar con una abundante cantidad de operadores y funciones relacionadas con el cálculo matricial y que lo hacen apto para ser usado con tal propósito.

R destaca, además de por la potencia de sus facetas estadísticas, por su efectividad, extensibilidad y por incluir una excelente serie de herramientas gráficas propias, relacionadas con la representación y análisis de datos.

R se distribuye gratuitamente bajo los términos de la GNU General Public Licence; su desarrollo y distribución son llevados a cabo por varios estadísticos conocidos como el Grupo Nuclear de Desarrollo de R (Emmanuel Paradis, 2002, pág. 27).

Se encuentra disponible en plataformas como: Windows, Linux (Debian, Mandrake, RedHat, SuSe), Macintosh y Alpha Unix.

TELA: es un lenguaje de cálculo numérico diseñado, especialmente, para tareas relacionadas con el pre-proceso y post-proceso numérico. Incluye un intérprete de comandos y soporte de gráficos 2D y 3D basados en plotmtv, un buen programa para la visualización de datos científicos.

Utiliza de forma nativa el formato HDF, uno de los estándares específicamente diseñados para la lectura y escritura e intercambio de datos científicos.

AXIOM: es un potente sistema de computación científica, creado en 1971 por IBM y comercializado con licencia privativa. No fue hasta septiembre de 2002 cuando Axiom fue liberado por el NAG (Numerical Algorithms Group), que lo había comprado a IBM en los años 1990. A partir de su liberación, su comunidad de desarrolladores se está centrando en su extensión, incluyendo aspectos como su interfaz de usuario o soporte de gráficos.

Según el Axiom Book, la potencia de Axiom se basa en sus excelentes características estructurales y a su expansibilidad sin límites: es un sistema abierto, modular y diseñado para soportar un gran número de nuevas características con un mínimo incremento en su complejidad estructural.

Características:

- Existe una excelente documentación.
- Es un programa sólido, potente y expansible, con una amplia comunidad que lo soporta.
- Soporte de datos estructurados, moderno lenguaje de programación que impulsa la programación estructurada y el chequeo fuerte de tipos, como en Pascal o en Ada.

- Su interfaz de usuario, por defecto, es en modo consola de texto, aunque admite otras posibilidades, desde su ejecución en un buffer de emacs hasta el uso de TEXmacs, un editor de textos basado en LATEX que permite empotrar sesiones interactivas con otros programas, entre ellos Axiom.
- Disponible para distintos sistemas operativos: GNU/Linux, Mac OSX, MSWindows,...
- Gráficos: El soporte de gráficos 2D todavía no estaba disponible en Debian GNU/Linux, aunque estaba a punto de ser incluido.

YACAS: (Yet Another Computer Algebra System) es un entorno de cálculo simbólico de propósito general. Incluye todas las capacidades básicas que se suponen en este tipo de programas: cálculo en precisión arbitraria, aritmética racional, números complejos, cálculo de derivadas, resolución de ecuaciones (simbólica y numéricamente), etc. ("Yacas," s.d.)

Características:

- Es un programa moderno, escrito en C++, que está siendo desarrollado por una comunidad muy dinámica. Incluye una amplia documentación.
- Existen versiones para numerosos sistemas (GNU/Linux, Mac OSX, distintos sabores de UNIX, MS Windows, etc.)
- Interfaz con usuario: Se utiliza la clásica consola de texto para introducir los comandos de YACAS, aunque existe una interfaz propia de usuario, todavía experimental, llamada proteus y basada en la librería gráfica fltk. Además, existe la posibilidad de utilizar como interfaz el programa TEXmacs.

Gráficos

- YACAS incluye soporte de gráficos 2D a través de gnuplot, conocida herramienta de gráficos 2D y 3D interactiva y orientada a línea de comandos.
- Además, existe soporte de gráficos 3D mejorado siempre que se cuente con el programa Superficie, que proporciona gráficos de gran calidad mediante OpenGL.

Se trata de un programa en desarrollo que ya ofrece interesantes perspectivas. Eso sí, existen versiones para GNU/Linux pero no para MS Windows.

PARI: es una biblioteca de álgebra computacional escrita, en este caso, en C e inicialmente diseñada para teoría de números por Henri Cohen (Université Bordeaux I, France) y otros colaboradores.

Actualmente, permite numerosas operaciones adicionales (matrices, polinomios,...).

Existe una interfaz, GP, que facilita el uso de Pari mediante un intérprete de comandos interactivo.

El entorno 5.2 también utiliza Pari como motor matemático.

1.7. Asistente matemático, basado en filosofía de software libre seleccionado para desarrollar la Herramienta Informática.

Después de hacer un estudio detallado sobre los asistentes matemáticos con licencia libre disponibles para desarrollar la Herramienta Informática, tomando en cuenta las características principales y funcionalidades de cada uno de ellos, se decidió seleccionar a Maxima con su interfaz xMaxima.

Justificación de la selección:

- Maxima brinda la interfaz xMaxima fácil, cómoda y agradable que cubre las necesidades básicas en el cálculo simbólico para un alumno de primer curso de una titulación universitaria.
- Maxima tiene entre sus funcionalidades, las requeridas para cumplir con el plan de estudios de las asignaturas Matemática I, II, III y posibilita el desarrollo de los métodos numéricos necesarios para su utilización en la Matemática IV, lo que puede influir positivamente en el PDE de la Disciplina Matemática General de la Carrera Ingeniería Informática.
- Maxima cuenta con un lenguaje de programación que puede permitir a los estudiantes de informática desarrollar sus propios algoritmos a partir de los realizados y de esta forma contribuir a la formación del profesional.
- Maxima es un sistema con licencia libre, por tanto responde a la política del país y del MES de utilizar software libre en lugar de privativo.
- Investigaciones realizadas apoyan la idea de utilizar Maxima porque demuestran que cubre las necesidades básicas de cálculo simbólico para un alumno de primer curso de una titulación universitaria, tal es el caso de las investigaciones realizadas por J. Rafael Rodríguez Galván de la universidad de Cádiz, España: (J. Rafael Rodríguez Galván, 2005, pág. 12), (J. Rafael

Rodríguez Galván, 2007, pág. 28) y por A. Pardini de la universidad nacional de la Plata, Argentina (A. Pardini, 2007, pág 37).

Conclusión.

En el presente capítulo, se ha realizado un estudio teórico sobre el contenido que es utilizado durante la investigación. Además se realiza un análisis detallado de algunos asistentes matemáticos basados en filosofía de software libre y se argumenta la decisión de escoger Maxima, con su interfaz xMaxima, como el adecuado para desarrollar la Herramienta Informática.

2. Descripción y construcción de la solución propuesta.

Introducción.

En el presente capítulo, utilizando los elementos que componen los métodos de análisis y diseño de sistemas estructurados, se realiza la descripción y construcción de la solución propuesta. Como parte de esta metodología se detallan: los diagramas de flujo de datos, la representación lógica en los procesos computacionales (Lenguaje natural estructurado) y el diccionario de datos. Además se describe la utilización integral de todos estos elementos, se describe la concepción general del sistema propuesto y se plantean los requisitos mínimos de: rendimiento, software y hardware.

2.1. Descripción de la solución propuesta.

Concepción general del sistema.

El sistema debe mostrar a estudiantes y profesores las funcionalidades necesarias para su utilización en la Matemática I, II, III así como opciones para calcular los diferentes métodos numéricos que se estudian en la asignatura Matemática IV y permitir crear nuevos algoritmos a partir de los desarrollados mediante un lenguaje de programación propio del sistema.

Requisitos mínimos de:

Rendimiento.

- La capacidad de procesamiento para ejecutar cada uno de los algoritmos y la envergadura de los cálculos que requieran un nivel de procesamiento elevado pueden variar en dependencia de la cantidad de iteraciones que deseen realizar los usuarios.
- Los tiempos de respuesta del sistema pueden variar en dependencia de la cantidad de iteraciones que se deseen realizar y las respuestas cuentan con un alto nivel de confiabilidad.

Software.

- Para la instalación del sistema propuesto se necesita la utilización del sistema operativo Windows 95 o superior, o Linux.

Hardware.

Para la instalación del sistema propuesto se requiere disponer, como mínimo, de computadoras con las siguientes características:

- 128 Mb. de RAM
- 512 Mb. de HDD Libre
- Mouse
- Procesador de velocidad 1GHz

2.2. Principios básicos del análisis y diseño estructurado.

El análisis y diseño estructurado de un sistema informático depende de utilizar un enfoque dialéctico y definir el sistema, moviéndose de lo más general a lo más particular, de ahí, generalizar de nuevo y así sucesivamente. En este capítulo se utilizarán los métodos de análisis y diseño estructurado difundidos por Yourdon (Yourdon, 1982), Weinberg (V. Weinberg, 1978) (V. Weinberg, 1978) y De Marco (T. De Marco, 1979).

La concepción de los autores mencionados en el párrafo anterior comienza por definir un sistema al mayor nivel de agregación posible y representarlo en un diagrama, de ahí descomponerlo en sistemas más detallados y representar cada nivel de descomposición en varios diagramas más detallados, hasta llegar a un nivel que no admite más descomposición.

Este lenguaje de modelación utiliza los siguientes símbolos para representar el flujo informativo que se produce en el sistema: saetas para indicar el flujo de datos entre procesos, círculos para indicar los procesos y cuadrados para indicar fuentes o destinos de información.

Las herramientas que forman parte de esta metodología de análisis y diseño estructurado son las siguientes:

- Diagramas de flujo de datos (DFD).
- Representación lógica en los procesos computacionales. Lenguaje natural estructurado (LNE).
- Diccionario de datos (DD).
- Diagrama de estructura de datos (DED).

2.3. Diagramas de flujo de datos (DFD).

Los DFD o diagramas de burbujas en el análisis y diseño estructurado, son representaciones reticulares de un sistema, definiendo a este en términos de sus elementos componentes y de las relaciones informativas existentes entre ellos.

Los DFD se elaboran partiendo de lo más general (en este caso se llama “Diagrama de contexto”), bajando a un nivel más detallado (“Diagrama cero”) y llegando hasta los niveles más elementales, donde el proceso ya no admite más descomposición, con lo que se llega a las “Tareas funcionales” (Lázaro J. Blanco, Ida R. Gutztat, 1991, pág. 208).

DIAGRAMA DE CONTEXTO

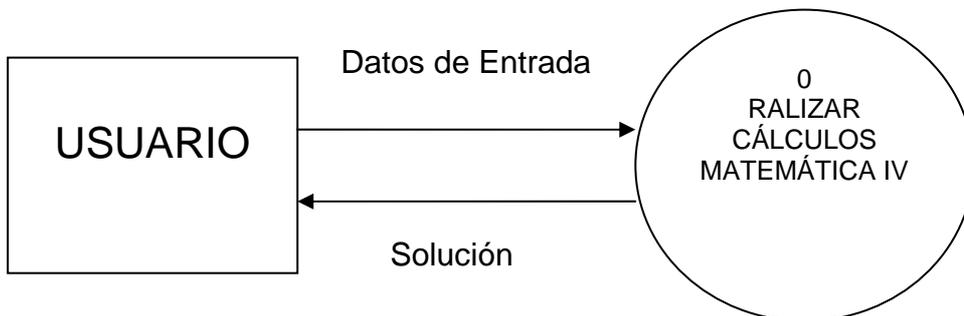
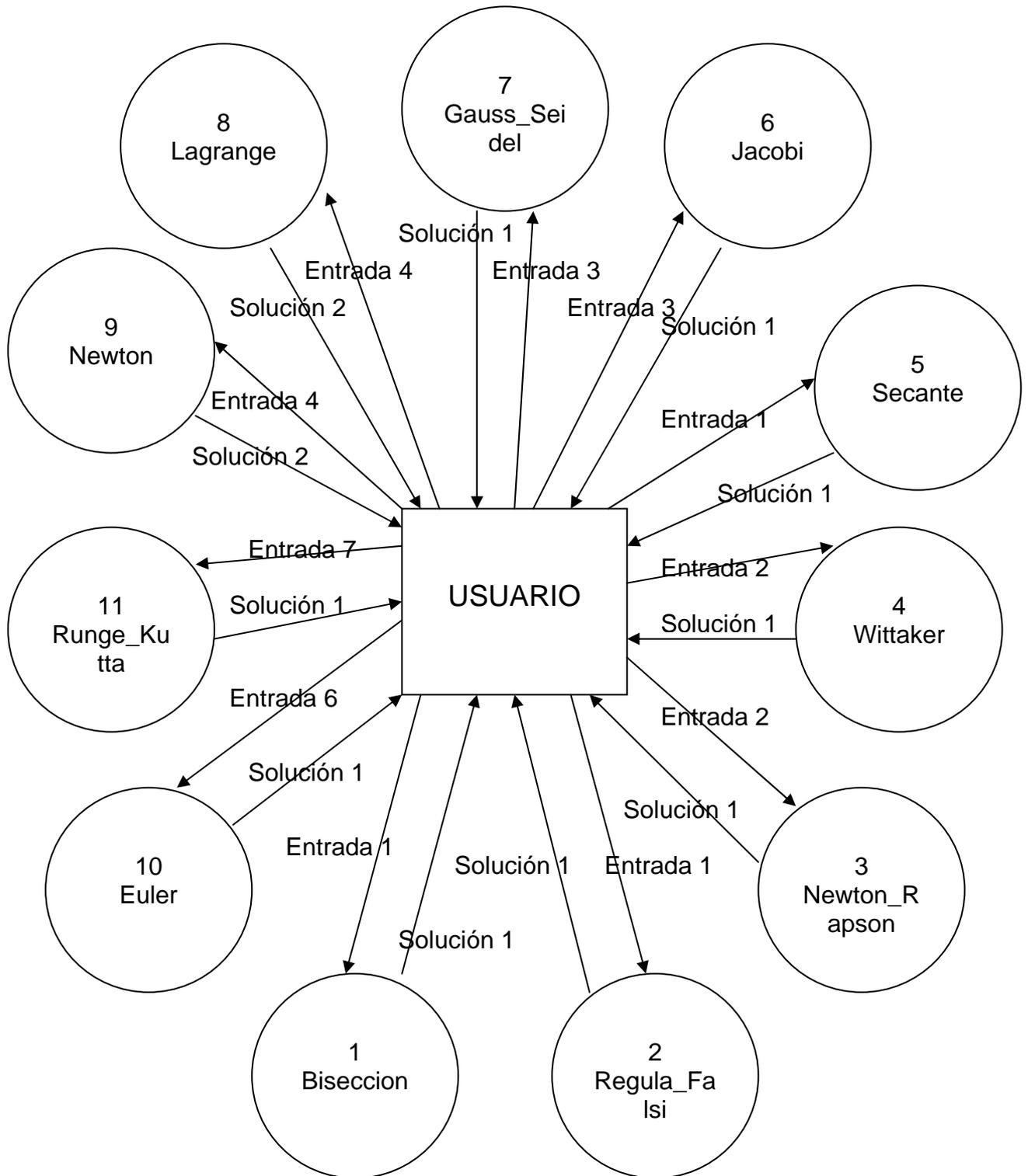
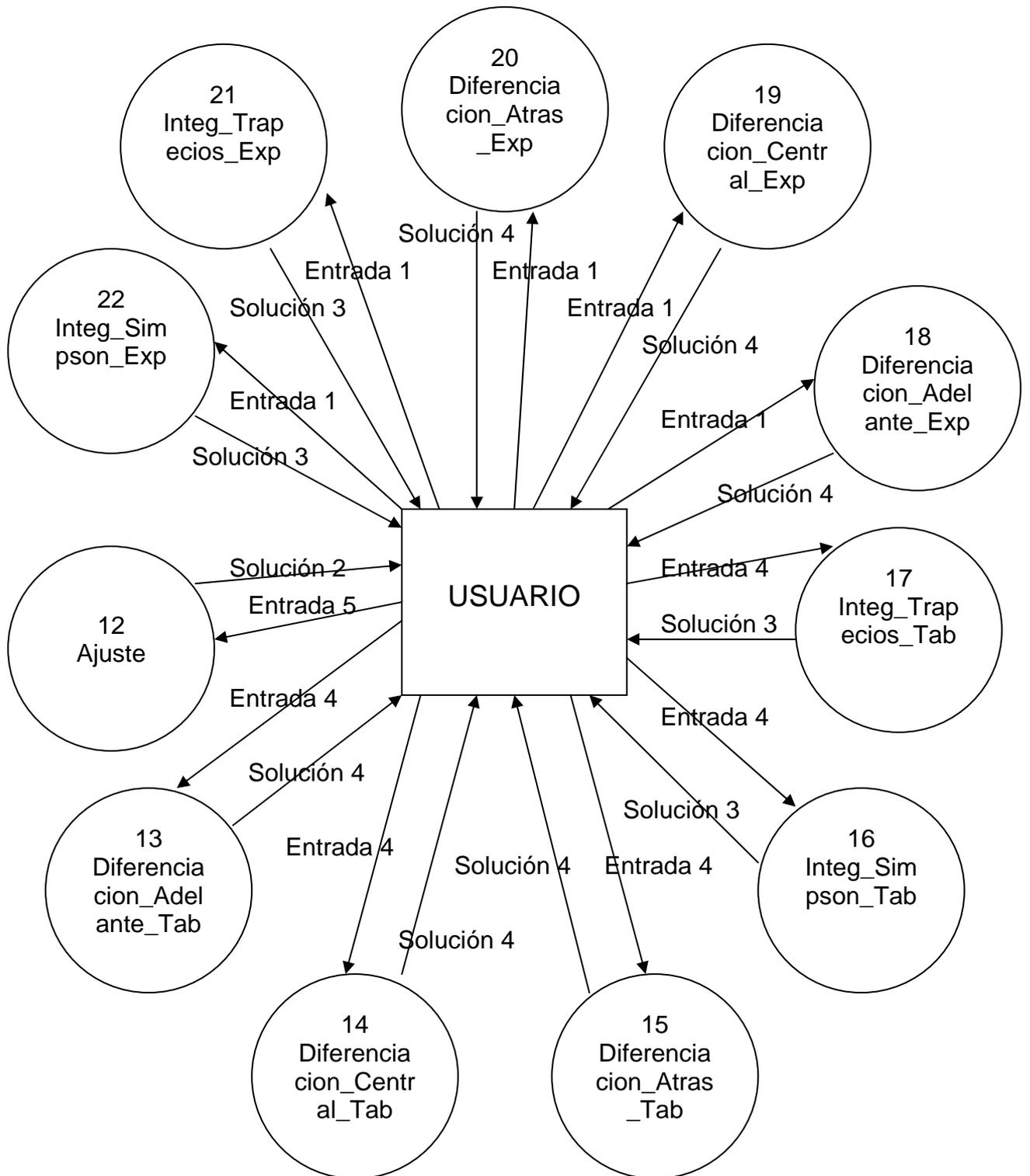
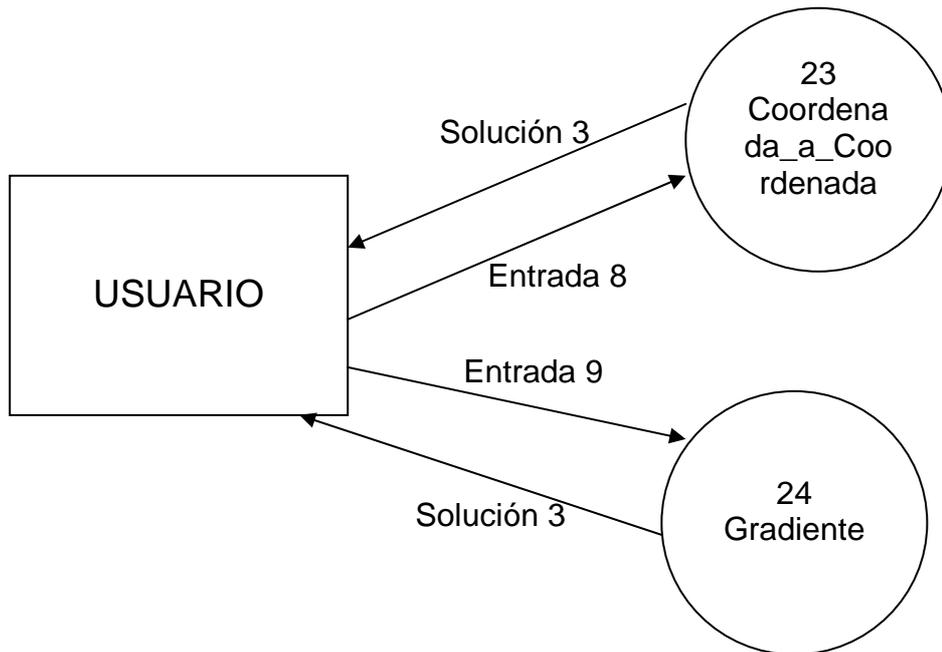


DIAGRAMA (0) REALIZAR CÁLCULOS MATEMÁTICA IV







2.4. Representación lógica en los procesos computacionales. Lenguaje natural estructurado (LNE).

Después de representar cada nivel de descomposición en varios diagramas detallados, hasta llegar al nivel que no admite más descomposición, se debe realizar la representación lógica de los procesos computacionales mediante (LNE).

El lenguaje estructurado es una adaptación del lenguaje natural, utilizada para describir los procesos lógicos que transcurren en una tarea. No es propiamente un lenguaje de programación, sino un método de plantear un problema, de forma tal que resulte sencilla y simple su comprensión y su traducción a un lenguaje de programación convencional (Lázaro J. Blanco, Ida R. Gutztat, 1991, pág. 245).

Biseccion (Entrada 1)

Inicializar variables

Repetir hasta cantidad de iteraciones

Si se cumple el teorema de Bolzano

Realizar cálculos para una iteración en el método de Biseccion

Sino

Error "No se cumple el teorema de Bolzano."

Devolver solución del método Biseccion (Solución 1)

Fin del método Biseccion

Regula_Falsi (Entrada 1)

Inicializar variables

Repetir hasta cantidad de iteraciones

Si se cumple el teorema de Bolzano

Realizar cálculos para una iteración en el método de Regula_Falsi

Sino

Error "No se cumple el teorema de Bolzano."

Devolver solución del método Regula_Falsi (Solución 1)

Fin del método Regula_Falsi

Newton_Rapson (Entrada 2)

Inicializar variables

Repetir hasta cantidad de iteraciones

Si se cumple la condición suficiente

Realizar cálculos para una iteración en el método de Newton_Rapson

Sino

Error "No se cumple la condición suficiente."

Devolver solución del método Newton_Rapson (Solución 1)

Fin del método Newton_Rapson

Whittaker (Entrada 2)

Inicializar variables

Repetir hasta cantidad de iteraciones

Si se cumple la condición suficiente

Realizar cálculos para una iteración en el método de Whittaker

Sino

Error "No se cumple la condición suficiente."

Devolver solución del método Whittaker (Solución 1)

Fin del método Whittaker

Secante (Entrada 1)

Inicializar variables

Repetir hasta cantidad de iteraciones

Si se cumple la condición suficiente

Realizar cálculos para una iteración en el método Secante

Sino

Error "No se cumple la condición suficiente."

Devolver solución del método Secante (Solución 1)

Fin del método Secante

Jacobi (Entrada 3)

Inicializar variables

Si la cantidad de ecuaciones, variables y valores iniciales son iguales y diferentes de cero

Si pueden eliminarse los ceros de la diagonal principal de la matriz de los coeficientes de las ecuaciones intercambiando filas

Repetir hasta cantidad de iteraciones

Realizar cálculos para una iteración en el método Jacobi

Sino

Error "Hay valores 0 en la diagonal principal de la matriz de los coeficientes de las ecuaciones y no se pueden eliminar intercambiando filas."

Sino

Error "Las cantidades de ecuaciones, variables y valores iniciales deben ser iguales, y diferentes de 0."

Devolver solución del método Jacobi (Solución 1)

Fin del método Jacobi

Gauss_Seidel (Entrada 3)

Inicializar variables

Si la cantidad de ecuaciones, variables y valores iniciales son iguales y diferentes de cero

Si pueden eliminarse los ceros de la diagonal principal de la matriz de los coeficientes de las ecuaciones intercambiando filas

Repetir hasta cantidad de iteraciones

Realizar cálculos para una iteración en el método Gauss_Seidel

Sino

Error "Hay valores 0 en la diagonal principal de la matriz de los coeficientes de las ecuaciones y no se pueden eliminar intercambiando filas."

Sino

Error "Las cantidades de ecuaciones, variables y valores iniciales deben ser iguales, y diferentes de 0."

Devolver solución del método Gauss_Seidel (Solución 1)

Fin del método Gauss_Seidel

Lagrange (Entrada 4)

Inicializar variables

Si la longitud de las listas de los valores de las variables dependientes e independientes es mayor o igual que 2.

Si las longitudes de las listas de los valores de las variables dependientes e independientes son iguales

Si los valores de la lista de las variables independientes son ascendentes

Calcular el polinomio de interpolación por el método de Lagrange

Sino

Error "Los valores de la lista de las variables independientes deben ser ascendentes."

Sino

Error "Las longitudes de las listas de los valores de las variables dependientes e independientes deben ser iguales."

Sino

Error "La longitud de las listas de los valores de las variables dependientes e independientes debe ser mayor o igual a 2."

Devolver solución del método Lagrange (Solución 2)

Fin del método Lagrange

Newton (Entrada 4)

Inicializar variables

Si la longitud de las listas de los valores de las variables dependientes e independientes es mayor o igual que 2.

Si las longitudes de las listas de los valores de las variables dependientes e independientes no son iguales

Si los valores de la lista de las variables independientes son ascendentes

Calcular el polinomio de interpolación por el método de Newton

Sino

Error "Los valores de la lista de las variables independientes deben ser ascendentes."

Sino

Error "Las longitudes de las listas de los valores de las variables dependientes e independientes deben ser iguales."

Sino

Error "La longitud de las listas de los valores de las variables dependientes e independientes debe ser mayor o igual a 2."

Devolver solución del método Newton (Solución 2)

Fin del método Newton

Ajuste (Entrada 5)

Inicializar variables

Si la longitud de las listas de los valores de las variables dependientes e independientes es mayor o igual que 2.

Si las longitudes de las listas de los valores de las variables dependientes e independientes son iguales

Si el grado del polinomio o modelo escogido para el ajuste es mayor que 0

Calcular el polinomio de Ajuste

Sino

Error " El grado del polinomio o modelo escogido para el ajuste debe ser mayor que 0."

Sino

Error "Las longitudes de las listas de los valores de las variables dependientes e independientes deben ser iguales."

Sino

Error "La longitud de las listas de los valores de las variables dependientes e independientes debe ser mayor o igual a 2."

Devolver solución del método Ajuste (Solución 2)

Fin del método Ajuste

Diferenciacion_Adelante_Tab (Entrada 4)

Inicializar variables

Si las longitudes de las listas de los valores de las variables dependientes e independientes son iguales

Repetir desde uno hasta la longitud de la lista de variables dependientes o independientes menos uno

Calcular el valor aproximado de la derivada para el intervalo actual mediante el método Diferenciacion_Adelante_Tab

Sino

Error "Las longitudes de las listas de los valores de las variables dependientes e independientes deben ser iguales."

Devolver solución del método Diferenciacion_Adelante_Tab (Solución 4)

Fin del método Diferenciacion_Adelante_Tab

Diferenciacion_Atras_Tab (Entrada 4)

Inicializar variables

Si las longitudes de las listas de los valores de las variables dependientes e independientes son iguales

Repetir desde dos hasta la longitud de la lista de variables dependientes o independientes menos uno

Calcular el valor aproximado de la derivada para el intervalo actual mediante el método Diferenciacion_Atras_Tab

Sino

Error "Las longitudes de las listas de los valores de las variables dependientes e independientes deben ser iguales."

Devolver solución del método Diferenciacion_Atras_Tab (Solución 4)

Fin del método Diferenciacion_Atras_Tab

Diferenciacion_Central_Tab (Entrada 4)

Inicializar variables

Si las longitudes de las listas de los valores de las variables dependientes e independientes son iguales

Repetir desde dos hasta la longitud de la lista de variables dependientes o independientes menos uno

Calcular el valor aproximado de la derivada para el intervalo actual mediante el método Diferenciacion_Central_Tab

Sino

Error "Las longitudes de las listas de los valores de las variables dependientes e independientes deben ser iguales."

Devolver solución del método Diferenciacion_Central_Tab (Solución 4)

Fin del método Diferenciacion_Central_Tab

Diferenciacion_Adelante_Exp (Entrada 1)

Inicializar variables

Si el paso es mayor que el valor positivo de la resta entre el extremo final e inicial del intervalo

Repetir desde uno hasta cantidad de iteraciones

Si se puede evaluar la función en el punto

Calcular el valor aproximado de la derivada para el punto siguiente mediante el método Diferenciacion_Adelante_Exp

Sino

Error "La función se indefine en el intervalo. "

Sino

Error "EL paso debe ser mayor que el valor positivo de la resta entre el extremo final e inicial del intervalo."

Devolver solución del método Diferenciacion_Adelante_Exp (Solución 4)

Fin del método Diferenciacion_Adelante_Exp

Diferenciacion_Atras_Exp (Entrada 1)

Inicializar variables

Si el paso es mayor que el valor positivo de la resta entre el extremo final e inicial del intervalo

Repetir desde dos hasta cantidad de iteraciones

Si se puede evaluar la función en el punto

Calcular el valor aproximado de la derivada para el punto anterior mediante el método Diferenciacion_Atras_Exp

Sino

Error "La función se indefine en el intervalo. "

Sino

Error "EL paso debe ser mayor que el valor positivo de la resta entre el extremo final e inicial del intervalo."

Devolver solución del método Diferenciacion_Atras_Exp (Solución 4)

Fin del método Diferenciacion_Atras_Exp

Diferenciacion_Central_Exp (Entrada 1)

Inicializar variables

Si el paso es mayor que el valor positivo de la resta entre el extremo final e inicial del intervalo

Repetir desde dos hasta cantidad de iteraciones

Si se puede evaluar la función en el punto

Calcular el valor aproximado de la derivada para el punto actual mediante el método Diferenciacion_Central_Exp

Sino

Error "La función se indefine en el intervalo. "

Sino

Error "EL paso debe ser mayor que el valor positivo de la resta entre el extremo final e inicial del intervalo."

Devolver solución del método Diferenciacion_Central_Exp (Solución 4)

Fin del método Diferenciacion_Central_Exp

Integ_Simpson_Exp (Entrada 1)

Inicializar variables

Si la cantidad de subintervalos es mayor que dos e impar

Repetir desde dos hasta cantidad de iteraciones

Si se puede evaluar la función en el punto

Calcular el valor aproximado de la integral para la iteración mediante el método Integ_Simpson_Exp

Sino

Error "La función se indefine en el intervalo. "

Sino

Error "La cantidad de subintervalos debe ser mayor que dos y par."

Devolver solución del método Integ_Simpson_Exp (Solución 3)

Fin del método Integ_Simpson_Exp

Integ_Trapecios_Exp (Entrada 1)

Inicializar variables

Si la cantidad de subintervalos es mayor que dos

Repetir desde dos hasta cantidad de iteraciones

Si se puede evaluar la función en el punto

Calcular el valor aproximado de la integral para la iteración mediante el método Integ_Trapecios_Exp

Sino

Error "La función se indefine en el intervalo. "

Sino

Error "La cantidad de subintervalos debe ser mayor que dos."

Devolver solución del método Integ_Trapecios_Exp (Solución 3)

Fin del método Integ_Trapecios_Exp

Integ_Simpson_Tab (Entrada 4)

Inicializar variables

Si la longitud de las listas de los valores, de las variables dependientes e independientes es mayor que dos y cantidad impar

Repetir desde dos hasta cantidad de iteraciones

Si se puede evaluar la función en el punto

Calcular el valor aproximado de la integral para la iteración mediante el método Integ_Simpson_Tab

Sino

Error "La función se indefine en el intervalo."

Sino

Error "La longitud de las listas de los valores, de las variables dependientes e independientes debe ser mayor que dos y cantidad impar."

Devolver solución del método Integ_Simpson_Tab (Solución 3)

Fin del método Integ_Simpson_Tab

Integ_Trapecios_Tab (Entrada 4)

Inicializar variables

Si la longitud de las listas de los valores, de las variables dependientes e independientes es mayor que dos

Repetir desde dos hasta cantidad de iteraciones

Si se puede evaluar la función en el punto

Calcular el valor aproximado de la integral para la iteración mediante el método Integ_Trapecios_Tab

Sino

Error "La función se indefine en el intervalo. "

Sino

Error "La longitud de las listas de los valores, de las variables dependientes e independientes debe ser mayor que dos."

Devolver solución del método Integ_Trapecios_Tab (Solución 3)

Fin del método Integ_Trapecios_Tab

Euler (Entrada 6)

Inicializar variables

Si el paso y la cantidad de iteraciones son valores mayores que 0

Si la longitud de las listas de variables, valores iniciales y la longitud + 1 de la lista de ecuaciones es mayor que 1

Repetir desde uno hasta cantidad de iteraciones

Calcular el valor aproximado del sistema de ecuaciones diferenciales mediante el método de Euler

Sino

Error " Si entre las variables dependientes hay alguna que no forme parte de al menos una ecuación, no es tomada en cuenta. La longitud de las listas de variables, valores iniciales y la longitud + 1 de la lista de ecuaciones debe ser mayor que 1."

Sino

Error "El paso y la cantidad de iteraciones deben ser valores mayores que 0."

Devolver solución del método Euler (Solución 1)

Fin del método Euler

Runge_Kutta (Entrada 7)

Inicializar variables

Si el orden del sistema de ecuaciones diferenciales es menor o igual que cuatro y mayor o igual que uno

Si el paso y la cantidad de iteraciones son valores mayores que 0

Si la longitud de las listas de variables, valores iniciales y la longitud + 1 de la lista de ecuaciones es mayor que 1

Repetir desde uno hasta cantidad de iteraciones

Calcular el valor aproximado del sistema de ecuaciones diferenciales mediante el método de Runge_Kutta

Sino

Error " Si entre las variables dependientes hay alguna que no forme parte de al menos una ecuación, no es tomada en cuenta. La longitud de las listas de variables, valores iniciales y la longitud + 1 de la lista de ecuaciones debe ser mayor que 1."

Sino

Error "El paso y la cantidad de iteraciones deben ser valores mayores que 0."

Sino

Error "El orden del sistema de ecuaciones diferenciales debe ser menor o igual que cuatro y mayor o igual que uno."

Devolver solución del método Runge_Kutta (Solución 1)

Fin del método Runge_Kutta

Coordenada_a_Coordenada (Entrada 8)

Inicializar variables

Si el paso y la cantidad de iteraciones son valores mayores que 0

Si la cantidad de variables, valores iniciales y valores finales son iguales

Si cada valor de la lista de valores iniciales de las variables es menor que su correspondiente valor final

Repetir desde uno hasta cantidad de subintervalos

Encontrar el valor aproximado de máximo o mínimo según los datos de entrada mediante el método de Coordenada_a_Coordenada

Sino

Error " Cada valor de la lista de valores iniciales de las variables debe ser menor que su correspondiente valor final."

Sino

Error "La cantidad de variables, valores iniciales y valores finales deben ser iguales."

Sino

Error "El paso y la cantidad de iteraciones deben ser valores mayores que 0."

Devolver solución del método Coordenada_a_Coordenada (Solución 3)

Fin del método Coordenada_a_Coordenada

Gradiente (Entrada 9)

Inicializar variables

Si el paso y la cantidad de iteraciones son valores mayores que 0

Si la cantidad de variables, valores iniciales y valores finales son iguales

Si cada valor de la lista de valores iniciales de las variables es menor que su correspondiente valor final

Repetir desde uno hasta cantidad de subintervalos

Encontrar el valor aproximado de máximo o mínimo según los datos de entrada mediante el método de Gradiente

Sino

Error "Cada valor de la lista de valores iniciales de las variables debe ser menor que su correspondiente valor final."

Sino

Error " La cantidad de variables, valores iniciales y valores finales deben ser iguales."

Sino

Error " El paso y la cantidad de iteraciones deben ser valores mayores que 0."

Devolver solución del método Gradiente (Solución 3)

Fin del método Gradiente

2.5. Diccionario de datos (DD)

El diccionario de datos es una herramienta que permite definir los términos asociados con el desarrollo de un sistema, sin permitir ambigüedades, indefiniciones, definiciones vagas o no precisas. O sea, permite definir (sintáctica, semántica y pragmáticamente) los datos y los procesos lógicos y físicos que transcurren en el sistema, tanto desde el punto de vista del usuario (proceso lógico) como del programador (proceso físico) (Lázaro J. Blanco, Ida R. Gutztat, 1991, pág. 258).

Definición de flujo de datos	
Nombre del flujo de datos	Entrada 1.
Esquema	Diagrama (0) Realizar Cálculos Matemática IV.
Descripción	Grupo de datos de entrada que deben ser tecleados por el usuario, separados por coma, necesarios para ejecutar los procesos especificados en <i>Destino</i> .
Composición	fx : miembro izquierdo de la ecuación después de igualada a cero. x : variable de la ecuación. a : extremo inferior del intervalo. b : extremo superior del intervalo. n : cantidad de iteraciones.
Fuente	Usuario.
Destino	Procesos: 1, 2, 5, 18, 19, 20, 21 y 22

Tabla 1: Entrada 1.

Definición de flujo de datos	
Nombre del flujo de datos	Entrada 2.
Esquema	Diagrama (0) Realizar Cálculos Matemática IV.
Descripción	Grupo de datos de entrada que deben ser tecleados por el usuario, separados por coma, necesarios para ejecutar los procesos especificados en <i>Destino</i> .
Composición	fx : miembro izquierdo de la ecuación después de igualada a cero. x : variable de la ecuación. a : primera aproximación a la solución de la ecuación. n : cantidad de iteraciones.
Fuente	Usuario.
Destino	Procesos: 3 y 4

Tabla 2: Entrada 2.

Definición de flujo de datos	
Nombre del flujo de datos	Entrada 3.
Esquema	Diagrama (0) Realizar Cálculos Matemática IV.
Descripción	Grupo de datos de entrada que deben ser tecleados por el usuario, separados por coma, necesarios para ejecutar los procesos especificados en <i>Destino</i> .
Composición	ec: lista de ecuaciones del sistema. vbles: lista de variables del sistema. val: lista de valores iniciales para cada variable, en su mismo orden. n: cantidad de iteraciones.
Fuente	Usuario.
Destino	Procesos: 1 y 2

Tabla 3: Entrada 3.

Definición de flujo de datos	
Nombre del flujo de datos	Entrada 4.
Esquema	Diagrama (0) Realizar Cálculos Matemática IV.
Descripción	Grupo de datos de entrada que deben ser tecleados por el usuario, separados por coma, necesarios para ejecutar los procesos especificados en <i>Destino</i> .
Composición	ec: lista de ecuaciones del sistema. vbles: lista de variables del sistema. val: lista de valores iniciales para cada variable, en su mismo orden. n: cantidad de iteraciones.
Fuente	Usuario.
Destino	Procesos: 8, 9, 13, 14, 15, 16 y 17

Tabla 4: Entrada 4.

Definición de flujo de datos	
Nombre del flujo de datos	Entrada 5.
Esquema	Diagrama (0) Realizar Cálculos Matemática IV.
Descripción	Grupo de datos de entrada que deben ser tecleados por el usuario, separados por coma, necesarios para ejecutar el proceso especificado en <i>Destino</i> .
Composición	valdep : lista de los valores de la variable dependiente. valind : lista de los valores de la variable independiente. n : grado del polinomio a modelar.
Fuente	Usuario.
Destino	Proceso: 12

Tabla 5: Entrada 5.

Definición de flujo de datos	
Nombre del flujo de datos	Entrada 6.
Esquema	Diagrama (0) Realizar Cálculos Matemática IV.
Descripción	Grupo de datos de entrada que deben ser tecleados por el usuario, separados por coma, necesarios para ejecutar el proceso especificado en <i>Destino</i> .
Composición	ecnes : lista de los miembros derechos de las ecuaciones, después de tener en cada miembro izquierdo a la derivada. vbles : lista de las variables del sistema, la primera se toma como la variable independiente. vinles : lista de los valores iniciales para cada variable. h : paso para resolver el sistema de ecuaciones diferenciales. n : cantidad de iteraciones.
Fuente	Usuario.
Destino	Proceso: 10

Tabla 6: Entrada 6.

Definición de flujo de datos	
Nombre del flujo de datos	Entrada 7.
Esquema	Diagrama (0) Realizar Cálculos Matemática IV.
Descripción	Grupo de datos de entrada que deben ser tecleados por el usuario, separados por coma, necesarios para ejecutar el proceso especificado en <i>Destino</i> .
Composición	<p>ecnes: lista de los miembros derechos de las ecuaciones, después de tener en cada miembro izquierdo a la derivada.</p> <p>vbles: lista de las variables del sistema, la primera se toma como la variable independiente.</p> <p>vinles: lista de los valores iniciales para cada variable.</p> <p>h: paso para resolver el sistema de ecuaciones diferenciales.</p> <p>n: cantidad de iteraciones.</p> <p>orden: orden con que se desea resolver el sistema de ecuaciones diferenciales.</p>
Fuente	Usuario.
Destino	Proceso: 11

Tabla 7: Entrada 7.

Definición de flujo de datos	
Nombre del flujo de datos	Entrada 8.
Esquema	Diagrama (0) Realizar Cálculos Matemática IV.
Descripción	Grupo de datos de entrada que deben ser tecleados por el usuario, separados por coma, necesarios para ejecutar el proceso especificado en <i>Destino</i> .
Composición	<p>f: expresión analítica de la función objetivo (función a analizar).</p> <p>vbles: lista de las variables independientes de la función objetivo.</p> <p>vinles: lista de los valores iniciales de cada una de las variables independientes de la función objetivo.</p> <p>vfinles: lista de los valores finales de cada una de las variables independientes de la función objetivo.</p> <p>max_min: si lo que se desea encontrar es un máximo se debe entrar max y si lo que se desea encontrar es un mínimo se debe entrar min.</p> <p>h: paso de iteración.</p> <p>n: cantidad de iteraciones.</p>
Fuente	Usuario.
Destino	Proceso: 23

Tabla 8: Entrada 8.

Definición de flujo de datos	
Nombre del flujo de datos	Entrada 9.
Esquema	Diagrama (0) Realizar Cálculos Matemática IV.
Descripción	Grupo de datos de entrada que deben ser tecleados por el usuario, separados por coma, necesarios para ejecutar el proceso especificado en <i>Destino</i> .
Composición	<p>f: expresión analítica de la función objetivo (función a analizar).</p> <p>vbles: lista de las variables independientes de la función objetivo.</p> <p>vinles: lista de los valores iniciales de cada una de las variables independientes de la función objetivo.</p> <p>vfinles: lista de los valores finales de cada una de las variables independientes de la función objetivo.</p> <p>max_min: si lo que se desea encontrar es un máximo se debe entrar max y si lo que se desea encontrar es un mínimo se debe entrar min.</p> <p>n: cantidad de iteraciones.</p>
Fuente	Usuario.
Destino	Proceso: 24

Tabla 9: Entrada 9.

Definición de flujo de datos	
Nombre del flujo de datos	Solución 1.
Esquema	Diagrama (0) Realizar Cálculos Matemática IV.
Descripción	Resultado emitido al usuario después de ejecutar los procesos especificados en <i>Fuente</i> .
Composición	Matriz de datos que sus filas representan la cantidad de iteraciones y las columnas elementos como: solución para cada variable, máximo error de cálculo, evaluación de las soluciones de cada variable en la función y extremos del nuevo intervalo de iteración.
Fuente	Procesos: 1, 2, 3, 4, 5, 6, 7, 10 y 11
Destino	Usuario.

Tabla 10: Solución 1.

Definición de flujo de datos	
Nombre del flujo de datos	Solución 2.
Esquema	Diagrama (0) Realizar Cálculos Matemática IV.
Descripción	Resultado en forma de polinomio emitido al usuario después de ejecutar los procesos, especificados en <i>Fuente</i> , relacionados con Interpolación y Ajuste.
Composición	Polinomio de Interpolación o Ajuste en función de la variable 'x'.
Fuente	Procesos: 8, 9 y 12
Destino	Usuario.

Tabla 11: Solución 2.

Definición de flujo de datos	
Nombre del flujo de datos	Solución 3.
Esquema	Diagrama (0) Realizar Cálculos Matemática IV.
Descripción	Resultado en forma de valor real emitido al usuario después de ejecutar los procesos, especificados en <i>Fuente</i> , relacionados con el cálculo aproximado de Integrales y Optimización de funciones.
Composición	Número real.
Fuente	Procesos: 16, 17, 21, 22, 23 y 24
Destino	Usuario.

Tabla 12: Solución 3.

Definición de flujo de datos	
Nombre del flujo de datos	Solución 4.
Esquema	Diagrama (0) Realizar Cálculos Matemática IV.
Descripción	Resultado en forma de lista de valores reales emitido al usuario después de ejecutar los procesos, especificados en <i>Fuente</i> , relacionados con el cálculo aproximado de la Derivada en cada punto de iteración.
Composición	Lista de números reales, cada valor corresponde al cálculo aproximado de la Derivada en un punto, de los entrados de forma tabular o calculados a partir del valor inicial y el paso de iteración.
Fuente	Procesos: 13, 14, 15, 18, 19 y 20
Destino	Usuario.

Tabla 13: Solución 4.

Definición de elemento de datos	
Nombre del elemento de datos	Ecuación.
Descripción	Miembro izquierdo de la ecuación después de igualada a cero.
Composición	Tolos los elementos que se pueden involucrar en la expresión analítica de una función de cualquier tipo: algebraica, exponencial, logarítmica, trigonométrica, hiperbólica.
Procesos asociados	Procesos: 1, 2, 3, 4, 5, 18, 19, 20, 21, 22, 23 y 24
Característica del dato	Cadena.

Tabla 14: Ecuación.

Definición de elemento de datos	
Nombre del elemento de datos	Variable.
Descripción	Una letra cualquiera que significa la variable independiente o dependiente de las funciones en cuestión.
Composición	Una letra o palabra.
Procesos asociados	Procesos: 1, 2, 3, 4, 5, 18, 19, 20, 21 y 22
Característica del dato	Cadena.

Tabla 15: Variable.

Definición de elemento de datos	
Nombre del elemento de datos	Valor inicial.
Descripción	Número que representa el extremo inferior de un intervalo de una variable.
Composición	Número real.
Procesos asociados	Procesos: 1, 2, 3, 4, 5, 18, 19, 20, 21 y 22
Característica del dato	Numérico.

Tabla 16: Valor inicial.

Definición de elemento de datos	
Nombre del elemento de datos	Valor final.
Descripción	Número que representa el extremo superior de un intervalo de una variable.
Composición	Número real.
Procesos asociados	Procesos: 1, 2, 3, 4, 5, 18, 19, 20, 21 y 22
Característica del dato	Numérico.

Tabla 17: Valor final.

Definición de elemento de datos	
Nombre del elemento de datos	Paso de iteración.
Descripción	Número que representa la diferencia del valor de la variable entre una iteración y otra.
Composición	Número real.
Procesos asociados	Procesos: 10, 11 y 23
Característica del dato	Numérico.

Tabla 18: Paso de iteración.

Definición de elemento de datos	
Nombre del elemento de datos	Cantidad de iteraciones.
Descripción	Número que representa la cantidad de iteraciones a realizar en el método numérico seleccionado.
Composición	Número entero.
Procesos asociados	Procesos: 1, 2, 3, 4, 5, 6, 7, 10, 11, 12, 18, 19, 20, 21, 22, 23 y 24
Característica del dato	Numérico.

Tabla 19: Cantidad de iteraciones.

Definición de elemento de datos	
Nombre del elemento de datos	Lista de ecuaciones.
Descripción	Lista de elementos de datos <i>Ecuación</i> definido con anterioridad. Estos elementos deben estar separados por coma y encerrados entre corchetes.
Composición	Una o más ecuaciones separadas por coma y encerradas entre corchetes.
Procesos asociados	Procesos: 6, 7, 10 y 11
Característica del dato	Lista.

Tabla 20: Lista de ecuaciones.

Definición de elemento de datos	
Nombre del elemento de datos	Lista de variables.
Descripción	Lista de elementos de datos <i>Variable</i> definido con anterioridad. Estos elementos deben estar separados por coma y encerrados entre corchetes.
Composición	Una o más variables separadas por coma y encerradas entre corchetes.
Procesos asociados	Procesos: 6, 7, 10, 11, 23 y 24
Característica del dato	Lista.

Tabla 21: Lista de variables.

Definición de elemento de datos	
Nombre del elemento de datos	Lista de valores iniciales.
Descripción	Lista de elementos de datos <i>Valor inicial</i> definido con anterioridad. Estos elementos deben estar separados por coma y encerrados entre corchetes.
Composición	Una o más valores iniciales separados por coma y encerrados entre corchetes.
Procesos asociados	Procesos: 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 23 y 24
Característica del dato	Lista.

Tabla 22: Lista de valores iniciales.

Definición de elemento de datos	
Nombre del elemento de datos	max_min.
Descripción	Elemento utilizado para la optimización de funciones, en caso que se desee calcular un mínimo de la función se debe entrar min como valor y en caso de que se desee calcular un máximo de la función se debe entrar max.
Composición	Cadena 'max' o 'min'.
Procesos asociados	Procesos: 23 y 24
Característica del dato	Cadena.

Tabla 23: max_min.

Definición de tarea funcional	
Nombre de la tarea funcional	Biseccion.
Número de identificación del DFD	0
Esquema	<pre> graph LR U[USUARIO] -- "Entrada 1" --> P((1 Biseccion)) P -- "Solución 1" --> U </pre>
Descripción	Permite encontrar aproximadamente los ceros o raíces de una función utilizando el método numérico Bisección, si los datos entrados por el usuario cumplen con el teorema de Bolzano.
Flujo de datos de entrada	Entrada 1.
Flujo de datos de salida	Solución 1.
Proceso lógico	Ver epígrafe 2.6: Biseccion (Entrada 1).

Tabla 24: Biseccion (Anexo 2).

Definición de tarea funcional	
Nombre de la tarea funcional	Regula_Falsi.
Número de identificación del DFD	0
Esquema	
Descripción	Permite encontrar aproximadamente los ceros o raíces de una función utilizando el método numérico Regula - Falsi, si los datos entrados por el usuario cumplen con el teorema de Bolzano.
Flujo de datos de entrada	Entrada 1.
Flujo de datos de salida	Solución 1.
Proceso lógico	Ver epígrafe 2.6: Regula_Falsi (Entrada 1).

Tabla 25: Regula_Falsi (Anexo 2).

Definición de tarea funcional	
Nombre de la tarea funcional	Newton_Rapson.
Número de identificación del DFD	0
Esquema	
Descripción	Permite encontrar aproximadamente los ceros o raíces de una función utilizando el método numérico Newton - Rapson, si la aproximación inicial entrada por el usuario posibilita la convergencia del método.
Flujo de datos de entrada	Entrada 2.
Flujo de datos de salida	Solución 1.
Proceso lógico	Ver epígrafe 2.6: Newton_Rapson (Entrada 2).

Tabla 26: Newton_Rapson (Anexo 2).

Definición de tarea funcional	
Nombre de la tarea funcional	Wittaker.
Número de identificación del DFD	0
Esquema	<pre> graph LR USUARIO[USUARIO] -- "Entrada 2" --> Wittaker((4 Wittaker)) Wittaker -- "Solución 1" --> USUARIO </pre>
Descripción	Permite encontrar aproximadamente los ceros o raíces de una función utilizando el método numérico Wittaker, si la aproximación inicial entrada por el usuario posibilita la convergencia del método.
Flujo de datos de entrada	Entrada 2.
Flujo de datos de salida	Solución 1.
Proceso lógico	Ver epígrafe 2.6: Wittaker (Entrada 2).

Tabla 27: Wittaker (Anexo 2).

Definición de tarea funcional	
Nombre de la tarea funcional	Secante.
Número de identificación del DFD	0
Esquema	
Descripción	Permite encontrar aproximadamente los ceros o raíces de una función utilizando el método numérico Secante, si no se indefine ni se hace cero la aproximación de la primera derivada de la función.
Flujo de datos de entrada	Entrada 1.
Flujo de datos de salida	Solución 1.
Proceso lógico	Ver epígrafe 2.6: Secante (Entrada 1).

Tabla 28: Secante (Anexo 2).

Definición de tarea funcional	
Nombre de la tarea funcional	Jacobi.
Número de identificación del DFD	0
Esquema	
Descripción	Permite encontrar de forma aproximada la solución de un sistema de ecuaciones lineales en varias variables, utilizando el método numérico Jacobi, si no existen ceros en la diagonal principal o existen y se pueden eliminar intercambiando las filas de la matriz de los coeficientes de las ecuaciones, y la cantidad de elementos de las listas de ecuaciones, variables y valores iniciales que se le pasan al método (Entrada 3) son iguales.
Flujo de datos de entrada	Entrada 3.
Flujo de datos de salida	Solución 1.
Proceso lógico	Ver epígrafe 2.6: Jacobi (Entrada 3).

Tabla 29: Jacobi (Anexo 3).

Definición de tarea funcional	
Nombre de la tarea funcional	Gauss_Seidel.
Número de identificación del DFD	0
Esquema	
Descripción	Permite encontrar de forma aproximada la solución de un sistema de ecuaciones lineales en varias variables, utilizando el método numérico Gauss - Seidel, si no existen ceros en la diagonal principal o existen y se pueden eliminar intercambiando las filas de la matriz de los coeficientes de las ecuaciones, y la cantidad de elementos de las listas de ecuaciones, variables y valores iniciales que se le pasan al método (Entrada 3) son iguales.
Flujo de datos de entrada	Entrada 3.
Flujo de datos de salida	Solución 1.
Proceso lógico	Ver epígrafe 2.6: Gauss_Seidel (Entrada 3).

Tabla 30: Gauss_Seidel (Anexo 3).

Definición de tarea funcional	
Nombre de la tarea funcional	Lagrange.
Número de identificación del DFD	0
Esquema	
Descripción	Permite encontrar de forma aproximada el polinomio de interpolación de la función tabular entrada por el usuario al método (Entrada 4), utilizando el método numérico Lagrange, si la lista de valores de la variable independiente es ascendente, y la cantidad de elementos de las listas de valores de las variables independiente y dependiente que se le pasan al método (Entrada 4) son iguales y mayor o igual que dos.
Flujo de datos de entrada	Entrada 4.
Flujo de datos de salida	Solución 2.
Proceso lógico	Ver epígrafe 2.6: Lagrange (Entrada 4).

Tabla 31: Lagrange (Anexo 4).

Definición de tarea funcional	
Nombre de la tarea funcional	Newton.
Número de identificación del DFD	0
Esquema	
Descripción	Permite encontrar de forma aproximada el polinomio de interpolación de la función tabular entrada por el usuario al método (Entrada 4), utilizando el método numérico Newton, si la lista de valores de la variable independiente es ascendente, y la cantidad de elementos de las listas de valores de las variables independiente y dependiente que se le pasan al método (Entrada 4) son iguales y mayor o igual que dos.
Flujo de datos de entrada	Entrada 4.
Flujo de datos de salida	Solución 2.
Proceso lógico	Ver epígrafe 2.6: Newton (Entrada 4).

Tabla 32: Newton (Anexo 4).

Definición de tarea funcional	
Nombre de la tarea funcional	Euler.
Número de identificación del DFD	0
Esquema	
Descripción	Permite encontrar de forma aproximada la solución de un sistema de ecuaciones diferenciales, utilizando el método numérico Euler, si la longitud de las listas de variables, valores iniciales y la longitud + 1 de la lista de ecuaciones es mayor que 1, y el paso de iteración y la cantidad de iteraciones son valores mayores que cero.
Flujo de datos de entrada	Entrada 6.
Flujo de datos de salida	Solución 1.
Proceso lógico	Ver epígrafe 2.6: Euler (Entrada 6).

Tabla 33: Euler (Anexo 8).

Definición de tarea funcional	
Nombre de la tarea funcional	Runge_Kutta.
Número de identificación del DFD	0
Esquema	
Descripción	Permite encontrar de forma aproximada la solución de un sistema de ecuaciones diferenciales, utilizando el método numérico Runge - Kutta, si la longitud de las listas de variables, valores iniciales y la longitud + 1 de la lista de ecuaciones es mayor que 1, el paso de iteración y la cantidad de iteraciones son valores mayores que cero, y el valor del orden con el que se desea realizar el método se encuentra entre 1 y 4 incluyendo ambos extremos.
Flujo de datos de entrada	Entrada 7.
Flujo de datos de salida	Solución 1.
Proceso lógico	Ver epígrafe 2.6: Runge_Kutta (Entrada 7).

Tabla 34: Runge_Kutta (Anexo 8).

Definición de tarea funcional	
Nombre de la tarea funcional	Ajuste.
Número de identificación del DFD	0
Esquema	
Descripción	Permite encontrar de forma aproximada el polinomio de Ajuste de la función tabular entrada por el usuario al método (Entrada 5), utilizando el método de mínimos cuadrados, si el valor que define el grado del polinomio escogido como modelo es mayor que cero, y la cantidad de elementos de las listas de valores de las variables independiente y dependiente que se le pasan al método (Entrada 5) son iguales y mayor o igual que dos.
Flujo de datos de entrada	Entrada 5.
Flujo de datos de salida	Solución 2.
Proceso lógico	Ver epígrafe 2.6: Ajuste (Entrada 5).

Tabla 35: Ajuste (Anexo 5).

Definición de tarea funcional	
Nombre de la tarea funcional	Diferenciacion_Adelante_Tab.
Número de identificación del DFD	0
Esquema	<pre> graph LR U[USUARIO] -- "Entrada 4" --> P((13 Diferenciacion_Adelante_Tab)) P -- "Solución 4" --> U </pre>
Descripción	Permite encontrar de forma aproximada las derivadas para cada punto, con excepción del último, de la función tabular entrada por el usuario, utilizando el método numérico Diferenciación hacia adelante, si la cantidad de elementos de las listas de valores de las variables independiente y dependiente que se le pasan al método (Entrada 4) son iguales.
Flujo de datos de entrada	Entrada 4.
Flujo de datos de salida	Solución 4.
Proceso lógico	Ver epígrafe 2.6: Diferenciacion_Adelante_Tab (Entrada 4).

Tabla 36: Diferenciacion_Adelante_Tab (Anexo 6).

Definición de tarea funcional	
Nombre de la tarea funcional	Diferenciacion_Central_Tab.
Número de identificación del DFD	0
Esquema	<pre> graph LR U[USUARIO] -- "Entrada 4" --> P((14 Diferenciación Central Tab)) P -- "Solución 4" --> U </pre>
Descripción	Permite encontrar de forma aproximada las derivadas para cada punto, con excepción del último y el primero, de la función tabular entrada por el usuario, utilizando el método numérico Diferenciación central, si la cantidad de elementos de las listas de valores de las variables independiente y dependiente que se le pasan al método (Entrada 4) son iguales.
Flujo de datos de entrada	Entrada 4.
Flujo de datos de salida	Solución 4.
Proceso lógico	Ver epígrafe 2.6: Diferenciacion_Central_Tab (Entrada 4).

Tabla 37: Diferenciacion_Central_Tab (Anexo 6).

Definición de tarea funcional	
Nombre de la tarea funcional	Diferenciaci _n _Atras_Tab.
Número de identificación del DFD	0
Esquema	<pre> graph LR U[USUARIO] -- "Entrada 4" --> P((15 Diferenciaci_n_Atras_Tab)) P -- "Solución 4" --> U </pre>
Descripción	Permite encontrar de forma aproximada las derivadas para cada punto, con excepción del primero, de la función tabular entrada por el usuario, utilizando el método numérico Diferenciación hacia atrás, si la cantidad de elementos de las listas de valores de las variables independiente y dependiente que se le pasan al método (Entrada 4) son iguales.
Flujo de datos de entrada	Entrada 4.
Flujo de datos de salida	Solución 4.
Proceso lógico	Ver epígrafe 2.6: Diferenciaci _n _Atras_Tab (Entrada 4).

Tabla 38: Diferenciaci_n_Atras_Tab (Anexo 6).

Definición de tarea funcional	
Nombre de la tarea funcional	Integ_Simpson_Tab.
Número de identificación del DFD	0
Esquema	<pre> graph LR U[USUARIO] -- "Entrada 4" --> P((16 Integ_Simpson_Tab)) P -- "Solución 3" --> U </pre>
Descripción	Permite encontrar de forma aproximada la integral de la función tabular entrada por el usuario, utilizando el método numérico Simpson, si la cantidad de elementos de las listas de valores de las variables independiente y dependiente que se le pasan al método (Entrada 4) son iguales.
Flujo de datos de entrada	Entrada 4.
Flujo de datos de salida	Solución 3.
Proceso lógico	Ver epígrafe 2.6: Integ_Simpson_Tab (Entrada 4).

Tabla 39: Integ_Simpson_Tab (Anexo 7).

Definición de tarea funcional	
Nombre de la tarea funcional	Integ_Trapecios_Tab.
Número de identificación del DFD	0
Esquema	<pre> graph LR U[USUARIO] -- "Entrada 4" --> P((17 Integ_Trapecios_Tab)) P -- "Solución 3" --> U </pre>
Descripción	Permite encontrar de forma aproximada la integral de la función tabular entrada por el usuario, utilizando el método numérico Trapecios, si la cantidad de elementos de las listas de valores de las variables independiente y dependiente que se le pasan al método (Entrada 4) son iguales.
Flujo de datos de entrada	Entrada 4.
Flujo de datos de salida	Solución 3.
Proceso lógico	Ver epígrafe 2.6: Integ_Trapecios_Tab (Entrada 4).

Tabla 40: Integ_Trapecios_Tab (Anexo 7).

Definición de tarea funcional	
Nombre de la tarea funcional	Diferenciacion_Adelante_Exp.
Número de identificación del DFD	0
Esquema	<pre> graph LR U[USUARIO] -- "Entrada 1" --> P((18 Diferenciación Adelante_Exp)) P -- "Solución 4" --> U </pre>
Descripción	Permite encontrar de forma aproximada las derivadas para cada punto que se obtiene de la evaluación de la función analítica entrada por el usuario, con excepción del último, utilizando el método numérico Diferenciación Adelante, si la función no se indefine en el intervalo y el paso es mayor que el valor positivo de la resta entre el extremo superior e inferior del intervalo.
Flujo de datos de entrada	Entrada 1.
Flujo de datos de salida	Solución 4.
Proceso lógico	Ver epígrafe 2.6: Diferenciacion_Adelante_Exp (Entrada 1).

Tabla 41: Diferenciacion_Adelante_Exp (Anexo 6).

Definición de tarea funcional	
Nombre de la tarea funcional	Diferenciacion_Central_Exp.
Número de identificación del DFD	0
Esquema	<pre> graph LR U[USUARIO] -- "Entrada 1" --> P((19 Diferenciación Central_Exp)) P -- "Solución 4" --> U </pre>
Descripción	Permite encontrar de forma aproximada las derivadas para cada punto que se obtiene de la evaluación de la función analítica entrada por el usuario, con excepción del último y el primero, utilizando el método numérico Diferenciación Central, si la función no se indefine en el intervalo y el paso es mayor que el valor positivo de la resta entre el extremo superior e inferior del intervalo.
Flujo de datos de entrada	Entrada 1.
Flujo de datos de salida	Solución 4.
Proceso lógico	Ver epígrafe 2.6: Diferenciacion_Central_Exp (Entrada 1).

Tabla 42: Diferenciacion_Central_Exp (Anexo 6).

Definición de tarea funcional	
Nombre de la tarea funcional	Diferenciacion_Atras_Exp.
Número de identificación del DFD	0
Esquema	<pre> graph LR U[USUARIO] -- "Entrada 1" --> P((20 Diferenciacion_Atras_Exp)) P -- "Solución 4" --> U </pre>
Descripción	Permite encontrar de forma aproximada las derivadas para cada punto que se obtiene de la evaluación de la función analítica entrada por el usuario, con excepción del primero, utilizando el método numérico Diferenciación Atrás, si la función no se indefine en el intervalo y el paso es mayor que el valor positivo de la resta entre el extremo superior e inferior del intervalo.
Flujo de datos de entrada	Entrada 1.
Flujo de datos de salida	Solución 4.
Proceso lógico	Ver epígrafe 2.6: Diferenciacion_Atras_Exp (Entrada 1).

Tabla 43: Diferenciacion_Atras_Exp (Anexo 6).

Definición de tarea funcional	
Nombre de la tarea funcional	Integ_Trapecios_Exp.
Número de identificación del DFD	0
Esquema	<pre> graph LR U[USUARIO] -- "Entrada 1" --> P((21 Integ_Trapecios_Exp)) P -- "Solución 3" --> U </pre>
Descripción	Permite encontrar de forma aproximada la integral de la función analítica entrada por el usuario, utilizando el método numérico Trapecios, si la función no se indefine en el intervalo.
Flujo de datos de entrada	Entrada 1.
Flujo de datos de salida	Solución 3.
Proceso lógico	Ver epígrafe 2.6: Integ_Trapecios_Tab (Entrada 1).

Tabla 44: Integ_Trapecios_Exp (Anexo 7).

Definición de tarea funcional	
Nombre de la tarea funcional	Integ_Simpson_Exp.
Número de identificación del DFD	0
Esquema	<pre> graph LR U[USUARIO] -- "Entrada 1" --> P((22 Integ_Simpson_Exp)) P -- "Solución 3" --> U </pre>
Descripción	Permite encontrar de forma aproximada la integral de la función analítica entrada por el usuario, utilizando el método numérico Simpson, si la función no se indefine en el intervalo y la cantidad de subintervalos es un valor par.
Flujo de datos de entrada	Entrada 1.
Flujo de datos de salida	Solución 3.
Proceso lógico	Ver epígrafe 2.6: Integ_Simpson_Tab (Entrada 1).

Tabla 45: Integ_Simpson_Exp (Anexo 7).

Definición de tarea funcional	
Nombre de la tarea funcional	Coordenada_a_Coordenada.
Número de identificación del DFD	0
Esquema	<pre> graph LR U[USUARIO] -- "Entrada 8" --> P((23 Coordenada _a_Coordenada)) P -- "Solución 3" --> U </pre>
Descripción	Permite encontrar de forma aproximada el máximo o mínimo de la función analítica entrada por el usuario, si no se indefine en el intervalo, utilizando el método numérico Coordenada a Coordenada.
Flujo de datos de entrada	Entrada 8.
Flujo de datos de salida	Solución 3.
Proceso lógico	Ver epígrafe 2.6: Coordenada_a_Coordenada (Entrada 8).

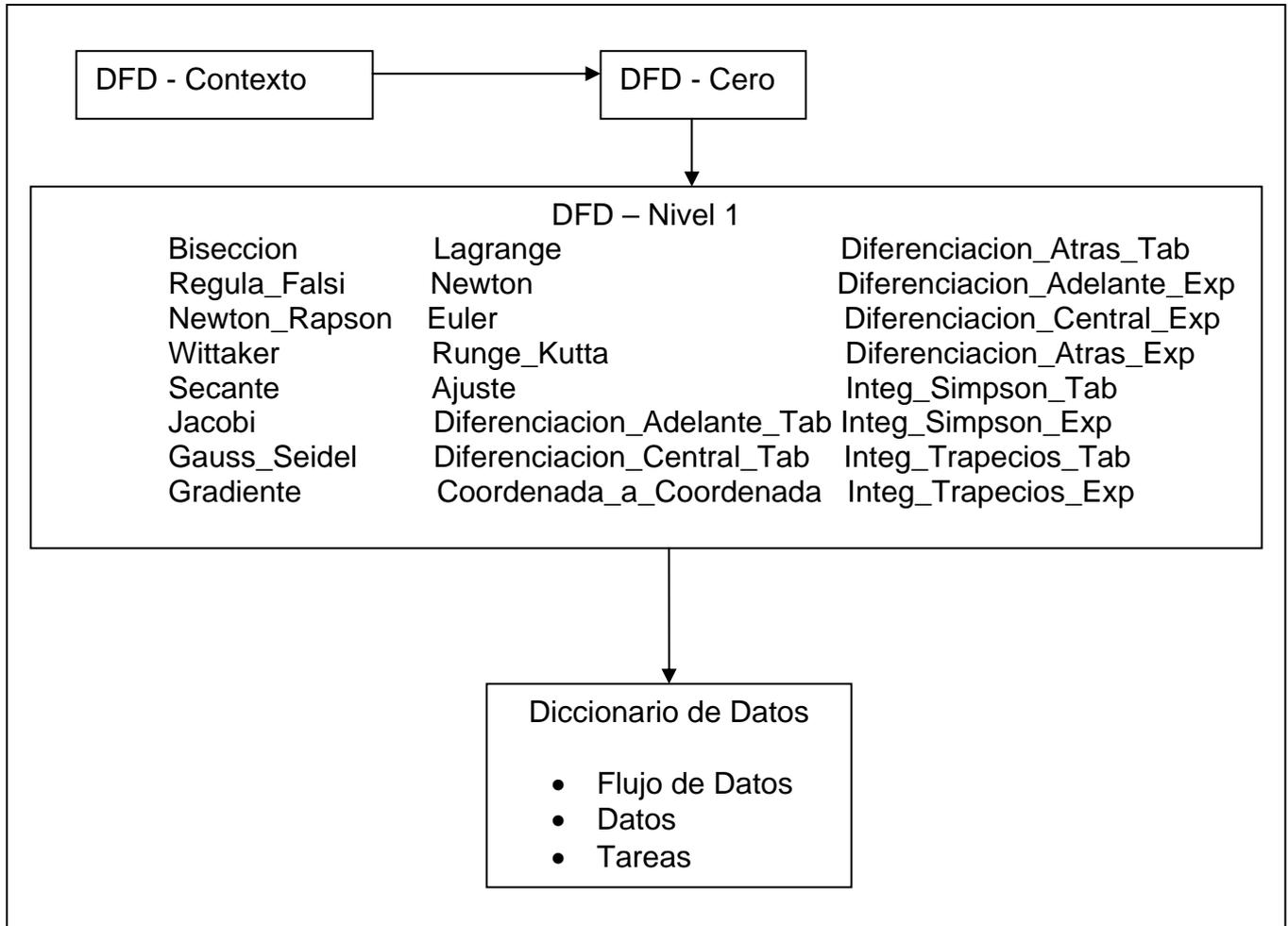
Tabla 46: Coordenada_a_Coordenada (Anexo 9).

Definición de tarea funcional	
Nombre de la tarea funcional	Gradiente.
Número de identificación del DFD	0
Esquema	
Descripción	Permite encontrar de forma aproximada el máximo o mínimo de la función analítica entrada por el usuario, utilizando el método numérico Gradiente, si no se indefine el cálculo del paso para cada iteración.
Flujo de datos de entrada	Entrada 8.
Flujo de datos de salida	Solución 3.
Proceso lógico	Ver epígrafe 2.6: Gradiente (Entrada 8).

Tabla 47: Gradiente (Anexo 9).

2.6. Relaciones entre las herramientas del análisis y diseño estructurado.

La perfecta integración que forman las herramientas que componen el análisis y diseño estructurado de sistemas se muestra en el gráfico de la figura siguiente:



Conclusión.

En el presente capítulo, utilizando los métodos de análisis y diseño de sistemas estructurados, se realiza la descripción de la solución propuesta, quedando definidos: los diagramas de flujo de datos, la representación lógica en los procesos computacionales (Lenguaje natural estructurado), el diccionario de datos y el diagrama de integración de todos estos elementos. Además se describe la concepción general del sistema propuesto y los requerimientos mínimos del sistema (rendimiento, software y de hardware).

3. Propuesta y Validación

Introducción

En el presente capítulo se presenta la propuesta para solucionar el problema de investigación, se demuestra la necesidad y factibilidad de su aplicación en el desarrollo del PDE de la asignatura Matemática IV de la Disciplina Matemática General en la Carrera Ingeniería Informática, se definen los elementos que la componen así como todas las consideraciones que se deben tener en cuenta para su utilización y se valida mediante una comparación con el asistente matemático Derive utilizado en la actualidad, tomando como guía el Plan D de la asignatura.

3.1. Presentación de la solución propuesta

Título de la propuesta

Herramienta Informática basada en filosofía de software libre.

Justificación de la Herramienta Informática

El proceso continuo y organizado de emigración a software libre y plataformas de código abierto en general que se inició en Cuba en el año 2005 es el suceso fundamental que evidencia el desarrollo de la Herramienta Informática. Además en la carrera Ingeniería Informática en particular, es muy importante que los estudiantes utilicen software basado en filosofía libre que les permita acceder al código fuente, para estudiarlo, poder hacer modificaciones, crear sus propios algoritmos y publicarlos. En la asignatura Matemática IV de la Disciplina Matemática General de la Carrera Ingeniería Informática, es significativo el beneficio que aporta la utilización de una Herramienta Informática basada en la filosofía de software libre y aún más si los estudiantes la han utilizado con anterioridad en las asignaturas Matemática I, II y III, ya que deben haber adquirido experiencia con su lenguaje de programación y con las funcionalidades que brinda en general, facilitando de esta forma el cumplimiento de los objetivos educativos del Plan de estudios D de la asignatura.

Objetivo de la Herramienta Informática

Favorecer y apoyar el PDE de la asignatura Matemática IV de la Disciplina Matemática General de la Carrera Ingeniería Informática.

Factibilidad de la aplicación de la Herramienta Informática

La Herramienta Informática se comprobó de forma práctica al instalarse en diferentes computadoras con diferentes configuraciones de hardware, mostrando resultados positivos en su funcionamiento. Su legalidad queda resguardada por la condición de ser basada en filosofía de software libre con la excepción de un acuerdo que se puede leer en el proceso de instalación, pero que no representa ninguna restricción legal para su uso.

Beneficios tangibles e intangibles

Los beneficios obtenidos con el desarrollo del sistema son fundamentalmente intangibles, ya que permite: a los estudiantes de las carreras de ingenierías estudiar la matemática numérica y cada uno de sus métodos. Por otra parte, la Herramienta Informática permite aprovechar las tecnologías existentes, vinculando la computación con la matemática, ya que para su realización se tuvo en cuenta la forma en que metodológicamente se trabaja la matemática numérica en los centros de educación superior.

Análisis de costos y beneficios

El sistema, no implica costo alguno para la Universidad de Cienfuegos Carlos Rafael Rodríguez, sin embargo, el desarrollo de todo producto informático va asociado a un costo relacionado principalmente con el salario de la persona que lo realiza y el justificarlo depende de los beneficios tangibles e intangibles que produce.

Para la realización de este sistema no fue necesaria una inversión en los medios técnicos ni en requerimientos de lenguajes, como el propuesto por Maxima, para su ejecución.

El sistema propuesto trae consigo beneficios sobre todo intangibles para la Universidad de Cienfuegos Carlos Rafael Rodríguez, para sus estudiantes y profesores de matemática, lo que indica que es factible implementar el sistema propuesto.

Descripción de la Herramienta Informática

Herramienta Informática

La Herramienta Informática está compuesta por los siguientes elementos:

- Asistente matemático Maxima con su interfaz gráfica xMaxima (Anexo 1), basado en filosofía de software libre.
- Métodos numéricos programados en Maxima y guardados en “Matemática Numérica.mac” fichero que debe copiarse en C:\Archivos de programa\Maxima-5.17.1\share\maxima\5.17.1\share después de realizada la instalación del asistente matemático; debe ser cargado en Maxima para su utilización ejecutando el comando “load(“Matemática Numérica”);” y está compuesto por:
 - Código de programación de los métodos numéricos estudiados en la asignatura Matemática IV.
- Página Web (Anexo 10) “MatNum.html” que debe ser copiada en C:\Archivos de programa\Maxima-5.17.1\share\maxima\5.17.1\xmaxima después de realizada la instalación del asistente matemático, puede mostrarse en el navegador web que brinda xMaxima, cargándola desde esa dirección y está compuesta por:
 - Introducción.
 - Lista de los temas de la asignatura Matemática IV.
 - Métodos numéricos programados para cada tema.
 - Ejemplos de los métodos numéricos programados, con la posibilidad de mostrar la solución en el asistente matemático.
 - Código de los métodos numéricos programados.
 - Métodos numéricos propuestos en el asistente matemático para cada tema.
 - Ejemplos de los métodos numéricos propuestos en asistente matemático, con la posibilidad de mostrar la solución en el asistente matemático.
 - Código de los métodos numéricos propuestos en el asistente matemático.

3.2. Validación de la Herramienta Informática

Plan D para Matemática IV

Objetivos Instructivos.

1. Seleccionar el método numérico más adecuado para la solución de un problema asociado con los temas de la asignatura.
2. Resolver un problema previamente modelado utilizando un método numérico y técnicas de computación.
3. Caracterizar el tratamiento de los errores, procurando disminuir la influencia de su propagación y realizando el análisis y validación del resultado obtenido.

Objetivos Educativos.

1. Contribuir a la formación de la concepción científica del mundo y del pensamiento científico mediante la comprensión de cómo se realiza un modelo matemático y cómo esto es un reflejo de la realidad.
2. Contribuir a que se desarrollen las capacidades cognoscitivas de los estudiantes, los hábitos de utilizar la literatura científica, la capacidad de razonamiento y del pensar lógicamente mediante el estudio de los temas de la asignatura.
3. Contribuir a la formación computacional de los estudiantes mediante la utilización de asistentes matemáticos, la creación de algunos programas y el desarrollo de la capacidad de algoritmizar.
4. Contribuir a la formación de la personalidad del alumno consolidando los hábitos de proceder reflexivamente y evaluar críticamente los resultados de su trabajo y la adecuación del método utilizado.
5. Contribuir al desarrollo de la capacidad de interpretar modelos creados, así como de crear nuevos modelos matemáticos para su posterior solución numérica.

Conocimientos básicos a adquirir.

Fundamentos de los métodos numéricos. Métodos numéricos del Álgebra Lineal: Operaciones con matrices. Sistemas de ecuaciones lineales. Raíces de ecuaciones. Aproximación de funciones. Derivación e integración numérica. Solución de Ecuaciones Diferenciales. Optimización.

Habilidades básicas a dominar.

1. Formular las expresiones de trabajo en los métodos numéricos de manera tal que se disminuya la propagación de errores.
2. Describir los rasgos esenciales de los métodos numéricos de solución de ecuaciones y realizar su interpretación geométrica en el caso de raíces reales.
3. Resolver ecuaciones utilizando el método adecuado a las características del problema y evaluar la calidad de la solución obtenida.
4. Interpretar las técnicas de ajuste e interpolación como formas de la aproximación de funciones y describir sus rangos esenciales.
5. Modelar y resolver problemas en que se requiera la aproximación de funciones utilizando técnicas de interpolación y ajuste, seleccionando la adecuada al problema.
6. Describir los métodos numéricos para la solución de sistemas de ecuaciones lineales. Analizar el condicionamiento de la matriz asociada. Modelar y resolver computacionalmente problemas evaluando la calidad de la solución obtenida.
7. Analizar el comportamiento de la derivación numérica y de su inestabilidad.
8. Describir los rasgos esenciales de las fórmulas de integración numérica.
9. Calcular integrales seleccionando la fórmula adecuada de acuerdo a las características del problema y evaluar la calidad de la solución.
10. Describir e interpretar geoméricamente los métodos numéricos de solución de EDO y sistemas de EDO.
11. Modelar y resolver problemas que conduzcan a EDO y SEDO, evaluando la calidad de las soluciones.
12. Describir e interpretar los métodos numéricos de búsqueda unidimensional y de búsqueda gradiente para obtener extremos de funciones y sus limitaciones.
13. Modelar y resolver problemas vinculados al perfil en que sea necesario encontrar extremos de funciones.
14. Interpretar el proceso de solución numérica iterativa como la generación de una sucesión de soluciones intermedias que converja a la solución del problema.

15. Utilizar asistentes matemáticos y otras técnicas de computación en la solución de problemas por métodos numéricos.

Derive

Es un asistente matemático con las siguientes posibilidades: Aritmética, Álgebra, Gráficos 2D y 3D, Cálculo, Vectores y matrices, Funciones y Programación (se pueden generar programas en C, Fortran, Pascal y Basic). Derive es muy sencillo de usar, y cuenta con una interfaz atractiva e intuitiva. Basta introducir la función que se quiera utilizar, y luego automáticamente se pueden dibujar, simplificar, aproximar, factorizar, diferenciar, o integrar. Permite sumar, multiplicar, transponer e invertir matrices. Las ecuaciones se pueden resolver analítica o aproximadamente.

El asistente matemático DERIVE es el utilizado actualmente para apoyar el PDE de esta asignatura Matemática IV. Este es un asistente matemático profesional con amplias posibilidades, pero no es software libre porque no cumple con las libertades descritas anteriormente. Se tomarán específicamente los aspectos fundamentales del Plan D para Matemática IV, donde se evidencian los principales problemas de este asistente matemático.

Dificultades en el cumplimiento de algunos Objetivos Educativos con Derive

#2 Contribuir a que se desarrollen las capacidades cognitivas de los estudiantes, los hábitos de utilizar la literatura científica, la capacidad de razonamiento y del pensar lógicamente mediante el estudio de los temas de la asignatura.

Derive no aporta mucho en este objetivo, la ayuda propuesta tiene el contenido disperso y no siempre se conoce el método aplicado en la solución de un problema, dificultando el desarrollo de las capacidades cognitivas, este aspecto fue mejorado con el módulo agregado en la investigación realizada en la Cátedra de matemáticas de la Facultad de Ingeniería Informática de la Universidad de Cienfuegos "Carlos Rafael Rodríguez", aunque no completamente ya que no se muestran ejemplos resueltos de los métodos programados para facilitar el estudio de los estudiantes. A pesar de que cuenta con una herramienta de programación no se facilita el aprendizaje de los estudiantes por no poder interactuar con el código de programación de los métodos propuestos en el paquete.

#3 Contribuir a la formación computacional de los estudiantes mediante la utilización de asistentes matemáticos, la creación de algunos programas y el desarrollo de la capacidad de algoritmizar.

La utilización del Derive en las Universidades Cubanas como asistente matemático puede verse limitada por su condición de software privativo. El programa está protegido por la legislación internacional “copyright”, por lo que queda prohibido su copia o distribución o de cualquiera de sus componentes sin una licencia válida. La última distribución del Derive fue en el año 2003 con su versión 6.

La creación de algunos programas y el desarrollo de la capacidad de algoritmizar se dificultan al no poderse interactuar con el código de programación de los métodos propuestos en el paquete.

Cumplimiento de Objetivos Educativos seleccionados con la Herramienta Informática

#2 Contribuir a que se desarrollen las capacidades cognoscitivas de los estudiantes, los hábitos de utilizar la literatura científica, la capacidad de razonamiento y del pensar lógicamente mediante el estudio de los temas de la asignatura.

La Página Web por la organización por temas del contenido de la asignatura Matemática IV y por las funcionalidades que brinda, como son: la declaración de los métodos numéricos programados y las variables que se deben pasar como parámetros, los ejemplos propuestos y el código de programación; facilitan el desarrollo de las capacidades cognoscitivas de los estudiantes y el estudio de los temas de la asignatura.

#3 Contribuir a la formación computacional de los estudiantes mediante la utilización de asistentes matemáticos, la creación de algunos programas y el desarrollo de la capacidad de algoritmizar.

El asistente matemático Maxima se distribuye bajo la licencia GPL y brinda al usuario ciertas libertades: libertad para utilizarlo, libertad para modificarlo y adaptarlo a sus propias necesidades, libertad para distribuirlo, libertad para estudiarlo y aprender su funcionamiento.

La gratuidad del programa, junto con las libertades recién mencionadas, hacen de Maxima una formidable herramienta pedagógica de investigación y de cálculo técnico, accesible a todos los presupuestos, tanto institucionales como

individuales. Además de poder ejecutarse en otros sistemas operativos con plataforma Linux.

La creación de algunos programas y el desarrollo de la capacidad de algoritmizar se facilitan al poderse interactuar con el código de programación de los métodos propuestos en el paquete.

Conclusión

En el presente capítulo quedó definida la Herramienta Informática y todos los elementos que la componen así como la necesidad y factibilidad de su aplicación y desarrollo, la validación mediante la comparación con el asistente matemático privativo Derive que se utiliza actualmente, tomando como guía el Plan D de la asignatura Matemática IV de la Disciplina Matemática General en la Carrera Ingeniería Informática, demuestra la pertinencia de la Herramienta Informática para ser utilizada en la docencia, por su principal característica de estar basada en filosofía de software libre.

Conclusiones

- Los asistentes matemáticos basados en filosofía de software privativo no son apropiados para apoyar el PDE de la asignatura Matemática IV de la carrera Ingeniería Informática.
- El asistente matemático, basado en filosofía de software libre, seleccionado (Maxima) es adecuado para desarrollar la Herramienta Informática necesaria para apoyar el PDE de la asignatura Matemática IV de la carrera Ingeniería Informática.
- La Herramienta Informática desarrollada, basada en filosofía de software libre, facilita el perfeccionamiento de su propio código de programación y constituye una poderosa base de estudio para desarrollar nuevos algoritmos.
- La Herramienta Informática desarrollada favorece el cumplimiento de los objetivos de la asignatura Matemática IV de la carrera Ingeniería Informática, por estar basada en filosofía de software libre.

Recomendaciones

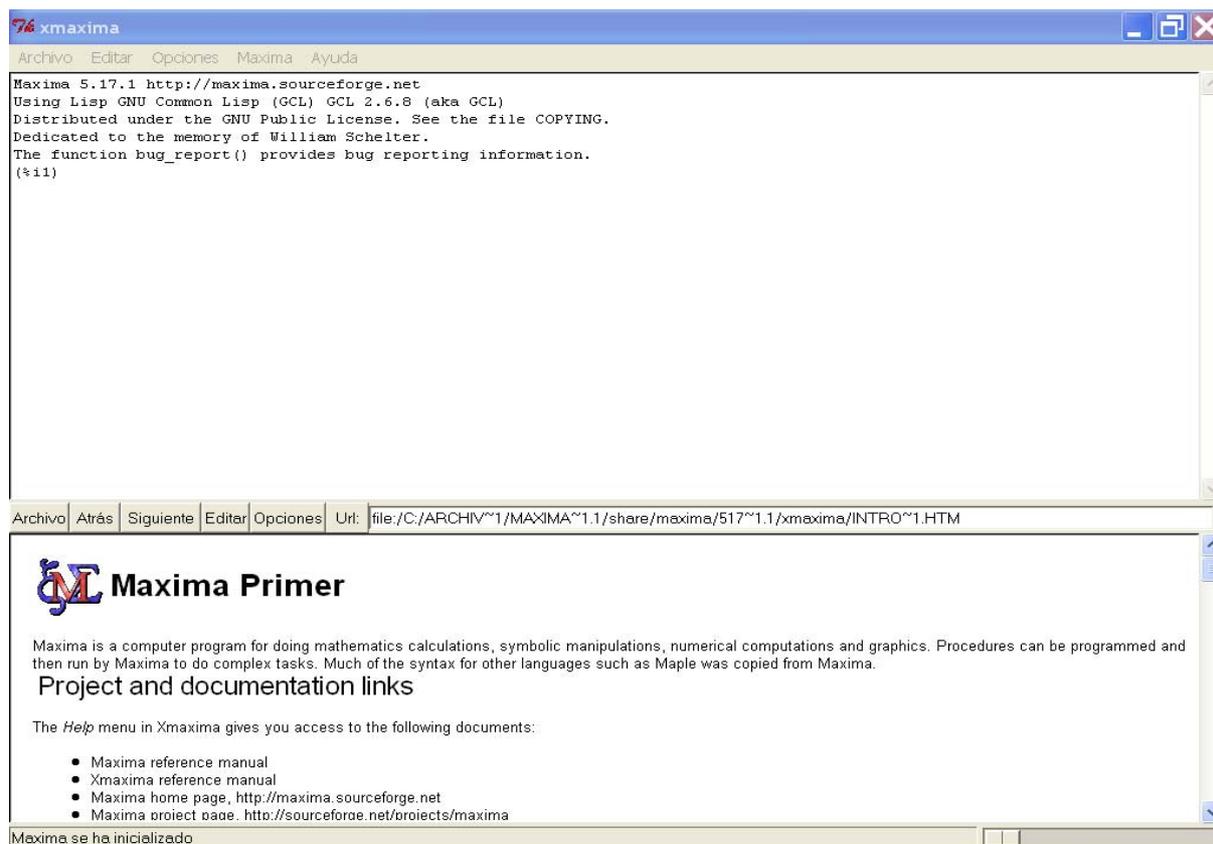
- No continuar utilizando el asistente matemático Derive en el PDE de la asignatura Matemática IV de la Disciplina Matemática General en la Carrera Ingeniería Informática por su condición de software privativo.
- Utilizar la Herramienta Informática desarrollada, basada en filosofía de software libre, en el PDE de la asignatura Matemática IV de la Disciplina Matemática General de la Carrera de Ingeniería Informática.
- Continuar perfeccionando la Herramienta Informática desarrollada a partir del estudio de su propio código de programación, aprovechando las facilidades que brinda la filosofía de software libre.
- Utilizar la Herramienta Informática en el PDE de la asignatura Matemática IV de la Disciplina Matemática General en la Carrera Ingeniería Informática.
- Analizar la posibilidad de exportar esta investigación a otras universidades.

Bibliografía

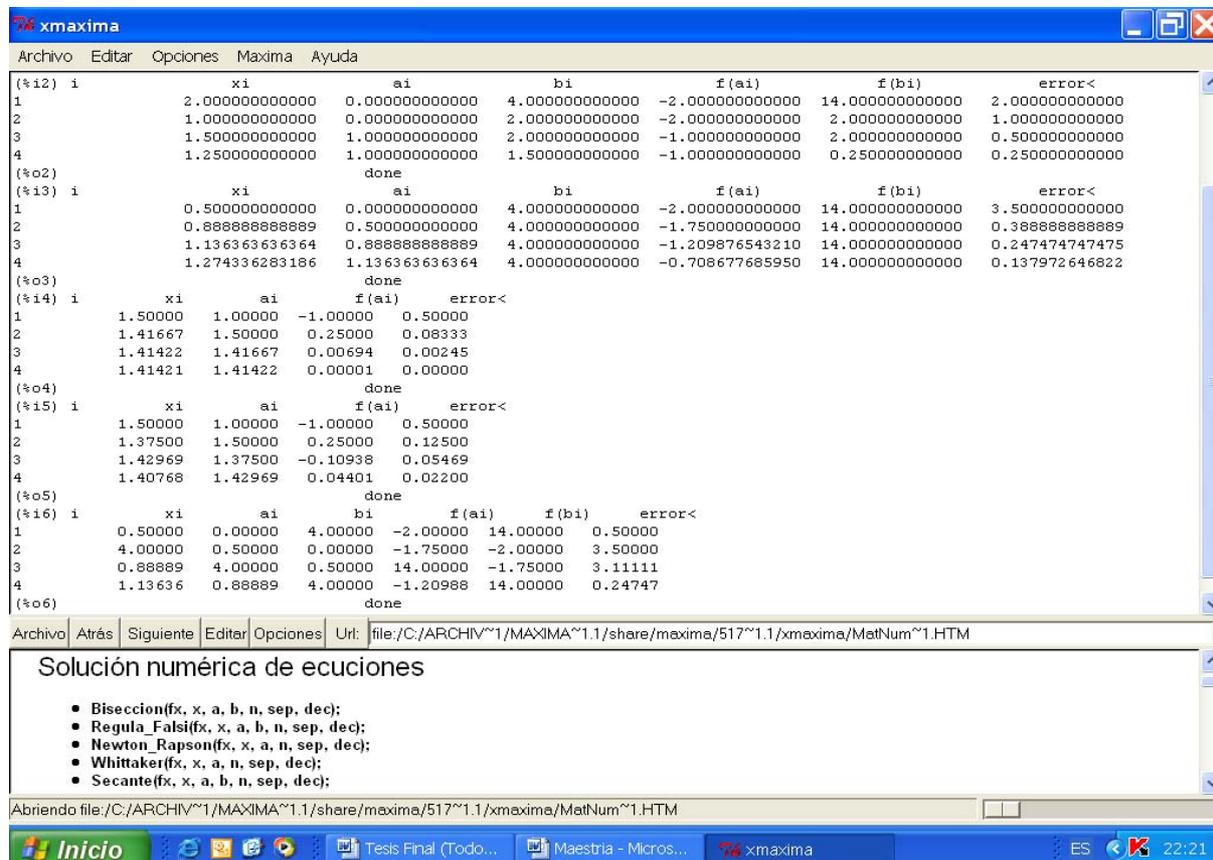
- A Bartolomé. (1995). *Nuevas tecnologías para la educación*. La Habana: Ed. Pueblo y Educación.
- A Pardini. (2007). *Fundamentación del uso de software libre en la universidad pública. Enseñando matemática con herramienta alternativas*. La Plata, Argentina: Universidad Nacional de La Plata.
- Carlos Cañedo. (2008). *El Diseño Curricular En La Educación Superior*. Cienfuegos: Universidad de Cienfuegos "Carlos Rafael Rodríguez".
- Colectivo de Autores. (2008a). Axiom book. Retrieved from <http://page.axiom-developer.org/zope/Plone/refs/books/axiom-book2.pdf>.
- Colectivo de Autores. (2008b). Derive. *Wikipedia, la enciclopedia libre*. Retrieved from <http://es.wikipedia.org/wiki/Derive>.
- Emmanuel Paradis. (2002). *R para principiantes*. Francia: Universit Montpellier.
- Euler. (2008). . Retrieved from <http://euler.sourceforge.net/index.html>.
- H. Schönemann Lossen. (2006). *Years of Singular Experiments in Mathematics*.
- Hal Schenck. (2003). *Computational Algebraic Geometry*.
- I. Gómez Plana. (2000). *La conectividad es la clave*.
- J L. Adams. (1991). *A comparison of computer -assisted instructional methods*. *Educ. Sci. Technol.*
- J. Rafael Rodríguez Galván. (2005). *Matemáticas y Software libre para la docencia en la Universidad de Cádiz*. España: Universidad de Cádiz.
- J. Rafael Rodríguez Galván. (2007). *Maxima con wxMaxima: software libre en el aula de matemáticas*. España: Universidad de Cádiz.
- John W. Eaton, D. B. (2008). *GNU Octave. A high-level interactive language for numerical computations*.

- Lázaro J. Blanco Encinosa, Ida R. Gutsztat. (n.d.). *Sistemas Informáticos. Teoría, métodos de elaboración, técnicas y herramientas* (Vol. 1). Cuba: Universidad de La Habana.
- Macaulay - Wikipedia, la enciclopedia libre. (2008). . Retrieved from <http://en.wikipedia.org/wiki/Macaulay>.
- Manuel. Álvarez Blanco, Alfredo Guerra Hernández, Rogelio Lau Fernández. (2004). *Matemática numérica*. La Habana: Felix Varela.
- Maxima-Wikipedia, la enciclopedia libre. (2008). . Retrieved from <http://es.wikipedia.org/wiki/Maxima>.
- P. Freire. (1993). *Pedagogy of the oppressed*. New York: Continuum.
- Plan de estudio D. Ingeniería Informática Presencial. (2007). . MES.
- Reinier. Pérez García. (2005). *Impacto de la Informatización en la Sociedad Cubana*. Ciudad de La Habana: Universidad de Ciencias Informáticas.
- Richard M. Stallman. (2004). Software libre para una sociedad libre. Retrieved from http://www.traficantes.net/index.php/trafis/editorial/catalogo/coleccion_mapas/software_libre_para_una_sociedad_libre.
- S. Benedeto. (200). *The influence of computer on Mathematics*. Italia: CISM.
- Singular-Wikipedia, la enciclopedia libre. (2008). . Retrieved from <http://es.wikipedia.org/wiki/SINGULAR>.
- T. De Marco. (1979). *Structured Analysis and Systems Specification*. EE.UU.
- V. Weinberg. (1978). *Structured Analysis*. New Jersey, EE.UU.
- Yacas. (2008). . Retrieved from <http://yacas.sourceforge.net>.

Anexos



Anexo 1: Asistente matemático Maxima (Interfaz gráfica xMaxima).



Anexo 2: Solución numérica de ecuaciones.

Archivo Editar Opciones Maxima Ayuda

```
(%i7) i
      x          y          z          error<
0      0.000000000000  0.000000000000  0.000000000000  0.000000000000
1      1.200000000000  1.200000000000  1.200000000000  1.200000000000
2      0.960000000000  0.960000000000  0.960000000000  0.240000000000
3      1.008000000000  1.008000000000  1.008000000000  0.048000000000
4      0.998400000000  0.998400000000  0.998400000000  0.009600000000
5      1.000320000000  1.000320000000  1.000320000000  0.001920000000
6      0.999936000000  0.999936000000  0.999936000000  0.000384000000
done
```

```
(%i8) i
      x          y          z          error<
0      0.00000  0.00000  0.00000  0.00000
1      1.20000  1.08000  0.97200  1.20000
2      0.99480  1.00332  1.00019  0.20520
3      0.99965  1.00002  1.00003  0.00485
4      1.00000  1.00000  1.00000  0.00035
5      1.00000  1.00000  1.00000  0.00001
6      1.00000  1.00000  1.00000  0.00000
done
```

Archivo Atrás Siguiente Editar Opciones Url: file:/C:/ARCHIV~1/MAXIMA~1.1/share/maxima/517~1.1/xmaxima/MatNum~1.HTM

Solución numérica de sistema de ecuaciones lineales

- **Jacobi(ec, vbles, val, n, sep, dec);**
- **Gauss_Seidel(ec, vbles, val, n, sep, dec);**

Definición de variables

- **ec:** lista de ecuaciones del sistema. ej: [x+y,x-2*y].
- **vbles:** lista de variables del sistema. ej: [x,y].
- **val:** lista de valores iniciales para cada variable, en su mismo orden. ej: [1,2].
- **n:** cantidad de iteraciones.

Opcionales

- **sep:** valor entero entre 1 y 16, que especifica el ancho de cada columna de los resultados.
- **dec:** valor entero entre 1 y 12, que especifica la cantidad de decimales de los resultados.

Ejemplos

Abriendo file:/C:/ARCHIV~1/MAXIMA~1.1/share/maxima/517~1.1/xmaxima/MatNum~1.HTM

Anexo 3: Solución numérica de sistema de ecuaciones lineales.

Archivo Editar Opciones Maxima Ayuda

```
(%i9)
lagrange_pol(x) = 17.37 (x - 0.3) (x - 0.2) x - 50.75333333333335 (x - 0.5) (x - 0.2) x
                + 33.55666666666667 (x - 0.5) (x - 0.3) x
done
```

```
(%i10)
newton_pol(x) = 0.173333333333333 (x - 0.3) (x - 0.2) x + 0.083666666666669 (x - 0.2) x + 1.0067 x
done
```

Archivo Atrás Siguiente Editar Opciones Url: file:/C:/ARCHIV~1/MAXIMA~1.1/share/maxima/517~1.1/xmaxima/MatNum~1.HTM

Aproximación de Funciones

Interpolación

- **Lagrange(valdep, valind);**
- **Newton(valdep, valind);**

Definición de variables

- **valdep:** lista de los valores de la variable dependiente. ej: [1,2,-1,8].
- **valind:** lista de los valores de la variable independiente. ej: [1,2,5,6]. Note que los valores de esta lista deben ir en ascenso.

Ejemplos

- **Lagrange([0,0.20134,0.30452,0.52110], [0,0.2,0.3,0.5]);**

$$\text{lagrange_pol}(x) = 17.37 (x - 0.3) (x - 0.2) x - 50.75333333333335 (x - 0.5) (x - 0.2) x + 33.55666666666667 (x - 0.5) (x - 0.3) x$$
- **Newton([0,0.20134,0.30452,0.52110], [0,0.2,0.3,0.5]);**

$$\text{newton_pol}(x) = 0.173333333333333 (x - 0.3) (x - 0.2) x + 0.083666666666669 (x - 0.2) x + 1.0067 x$$

Programación

Abriendo file:/C:/ARCHIV~1/MAXIMA~1.1/share/maxima/517~1.1/xmaxima/MatNum~1.HTM

Anexo 4: Aproximación de funciones (Interpolación).

The screenshot shows the xmaxima software interface. The main window displays the results of polynomial fitting for two cases:

Case 1 (Linear fit):

$$\text{ajuste_pol}(x) = \frac{45x + 50}{82x + 41}$$

Case 2 (Quadratic fit):

$$\text{ajuste_pol}(x) = \frac{73x^2 + 585x + 6}{781x^2 + 1562x + 71}$$

The help window titled "Ajuste" provides the following information:

- Ajuste(valdep, valind, n);**
- Definición de variables:**
 - valdep:** lista de los valores de la variable dependiente. ej: [1,2,-1,8].
 - valind:** lista de los valores de la variable independiente. ej: [1,2,5,6].
 - n:** grado del polinomio a modelar. $n > 0$.
- Ejemplos:**
 - Ajuste([1,-2,2,4], [1,2,6,9], 1);** Modelo Lineal.

$$\text{ajuste_pol}(x) = \frac{45x + 50}{82x + 41}$$
 - Ajuste([1,-2,2,4], [1,2,6,9], 2);** Modelo Cuadrático.

$$\text{ajuste_pol}(x) = \frac{73x^2 + 585x + 6}{781x^2 + 1562x + 71}$$

Anexo 5: Aproximación de funciones (Ajuste).

The screenshot shows the xmaxima software interface displaying numerical differentiation results:

Case 1:

$$f'(1) \text{ es aproximadamente igual a: } -3$$

$$f'(2) \text{ es aproximadamente igual a: } 4$$

$$f'(6) \text{ es aproximadamente igual a: } 2$$

Case 2:

$$f'(2) \text{ es aproximadamente igual a: } 1$$

$$f'(6) \text{ es aproximadamente igual a: } 6$$

$$f'(9) \text{ es aproximadamente igual a: } -4$$

Case 3:

$$f'(2) \text{ es aproximadamente igual a: } -3$$

$$f'(6) \text{ es aproximadamente igual a: } 4$$

$$f'(9) \text{ es aproximadamente igual a: } 2$$

Case 4:

$$f'(1) \text{ es aproximadamente igual a: } -\frac{16}{5}$$

$$f'(3) \text{ es aproximadamente igual a: } -\frac{25}{3}$$

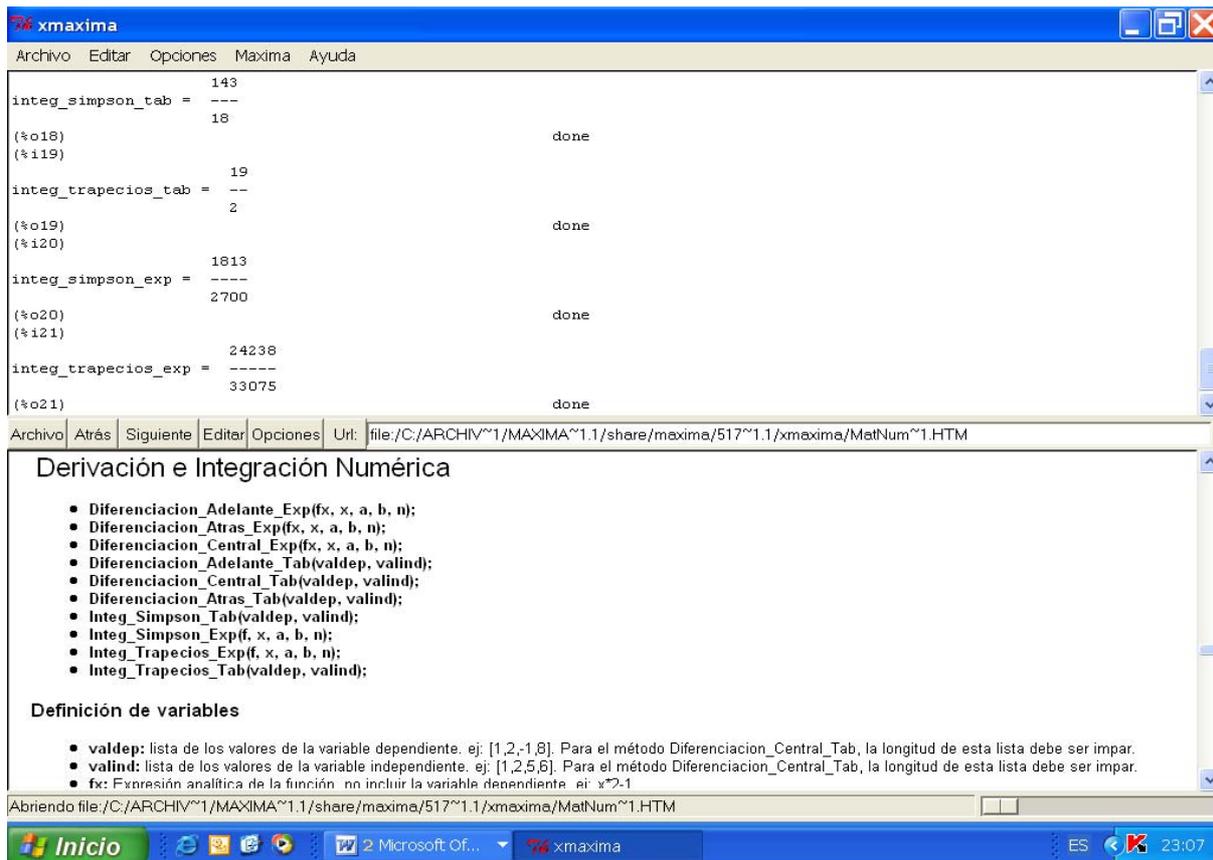
$$f'(7) \text{ es aproximadamente igual a: } -\frac{216}{1225}$$

$$f'(3) \text{ es aproximadamente igual a: } -\frac{32}{441}$$

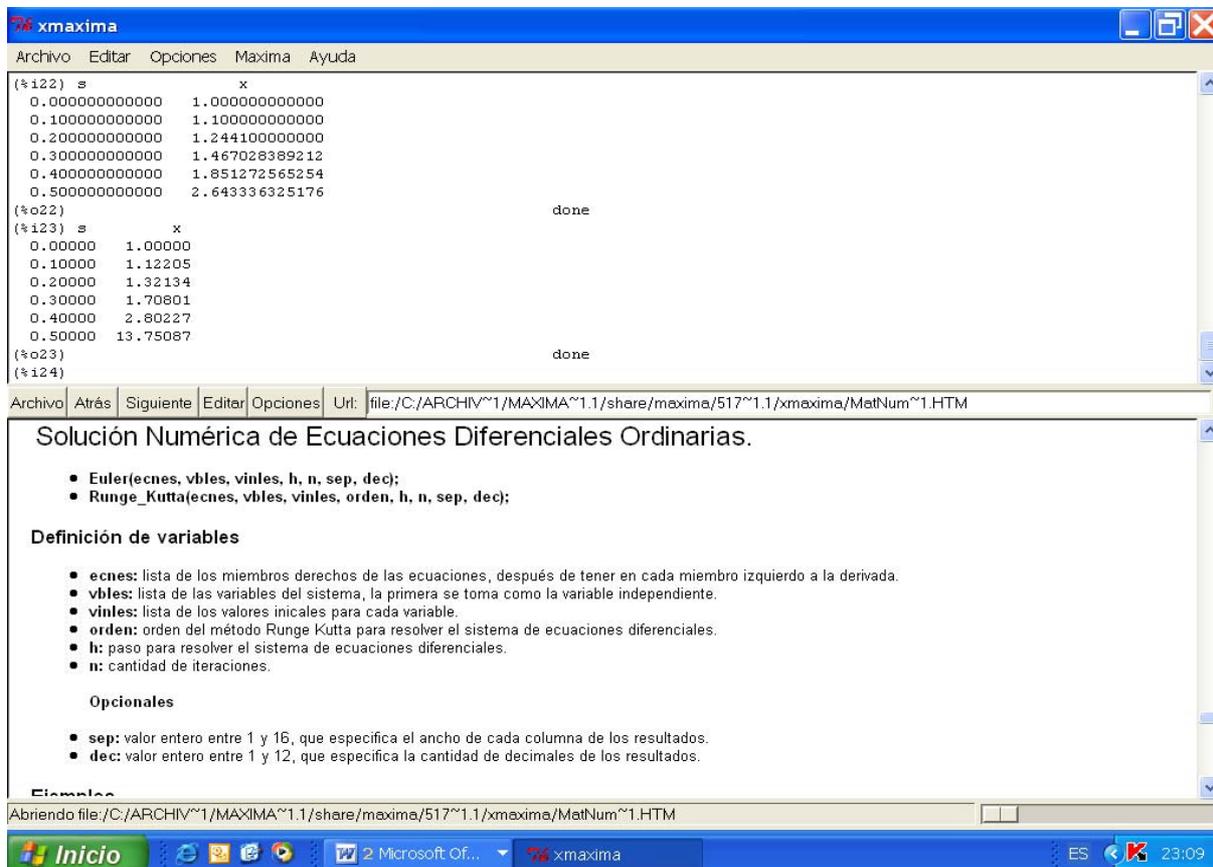
The help window titled "Derivación e Integración Numérica" lists the following functions:

- Diferenciacion_Adelante_Exp(f(x, a, b, n));
- Diferenciacion_Atras_Exp(f(x, a, b, n));
- Diferenciacion_Central_Exp(f(x, a, b, n));
- Diferenciacion_Adelante_Tab(valdep, valind);
- Diferenciacion_Central_Tab(valdep, valind);
- Diferenciacion_Atras_Tab(valdep, valind);
- Integ_Simpson_Tab(valdep, valind);
- Integ_Simpson_Exp(f(x, a, b, n));
- Integ_Trapecios_Exp(f(x, a, b, n));

Anexo 6: Derivación e integración numérica (Derivación).



Anexo 7: Derivación e integración numérica (Integración).



Anexo 8: Solución numérica de ecuaciones diferenciales ordinarias.

The screenshot shows the xMaxima application window. The menu bar includes 'Archivo', 'Editar', 'Opciones', 'Maxima', and 'Ayuda'. The main text area displays the results of two optimization processes:

```
(%i24) x          y
-2.000000000000 -3.000000000000
 0.000000000000 -0.000000000000
 0.000000000000 -0.000000000000
 0.000000000000 -0.000000000000
 0.000000000000 -0.000000000000
 0.000000000000 -0.000000000000
min f aproximadamente = 3.8980785963211196E-28
(%o24) done
(%i25) x          y
-2.000000 -3.000000
-0.890600 -1.335900
 0.218800  0.328200
-0.058550 -0.087820
 0.010790  0.016180
-0.006550 -0.009820
min f aproximadamente = 1.3927010257695183E-4
(%o25) done
(%i26)
```

The status bar shows the URL: file:/C:/ARCHIV~1/MAXIMA~1.1/share/maxima/517~1.1/xmaxima/MatNum~1.HTM.

Below the main window, a document titled 'Optimización Numérica' is displayed. It contains the following sections:

- Coordenada_a_Coordenada(f, vbles, vinles, vfinles, max_min, h, n, sep, dec);**
- Gradiente(f, vbles, vinles, max_min, n, sep ,dec);**

Definición de variables

- f:** expresión analítica de la función objetivo (función a analizar). ej: x^2+x*y
- vbles:** lista de las variables independientes de la función objetivo. ej: [x,y].
- vinles:** lista de los valores iniciales de cada una de las variables independientes de la función objetivo. ej: [0,1].
- vfinles:** lista de los valores finales de cada una de las variables independientes de la función objetivo. ej: [1,2].
- max_min:** si lo que se desea encontrar es un máximo se debe entrar max y si lo que se desea encontrar es un mínimo se debe entrar min.
- h:** paso de iteración.
- n:** cantidad de iteraciones.

Opcionales

- sep:** valor entre 1 y 16 que especifica el ancho de cada columna de los resultados.

The taskbar at the bottom shows the Windows Start button, several application icons, and the system tray with the date '23:27' and language 'ES'.

Anexo 9: Optimización numérica.

The screenshot shows the xMaxima application window. The menu bar includes 'Archivo', 'Editar', 'Opciones', 'Maxima', and 'Ayuda'. The main text area displays the following information:

```
Maxima 5.17.1 http://maxima.sourceforge.net
Using Lisp GNU Common Lisp (GCL) GCL 2.6.8 (aka GCL)
Distributed under the GNU Public License. See the file COPYING.
Dedicated to the memory of William Schelter.
The function bug_report() provides bug reporting information.
(%i1)
```

The status bar shows the URL: file:/C:/ARCHIV~1/MAXIMA~1.1/share/maxima/517~1.1/xmaxima/MatNum~1.HTM.

Below the main window, a document titled 'Matemática Numérica' is displayed. It features the logo of the University of Murcia and the following text:

El Análisis Numérico es un campo muy amplio de la Matemática. Se le conoce indistintamente como Matemática Numérica, Cálculo Numérico, Matemática Computacional ó Métodos de Cómputo. Se puede definir como: "la teoría y la práctica del cálculo eficiente y la estimación del error de la solución aproximada de muchos problemas de aplicación de la Matemática".

Nota: para utilizar los ejemplos propuestos en esta página se debe ejecutar el siguiente comando de xMaxima.

- `load("Matemática Numérica")`

Organización por temas

- Tema 1: Solución numérica de ecuaciones.
- Tema 2: Solución numérica de sistema de ecuaciones lineales.
- Tema 3: Aproximación de funciones.
- Tema 4: Derivación e integración numérica.
- Tema 5: Métodos numéricos en ecuaciones diferenciales.
- Tema 6: Optimización de funciones.

Solución numérica de ecuaciones

- `Biseccion(fx, x, a, b, n, sep, dec);`
- `Regula_Falsifx, x, a, b, n, sep, dec);`
- `Newton_Rapson(fx, x, a, n, sep, dec);`
- `Whittaker(fx, x, a, n, sep, dec);`
- `Secante(fx, x, a, b, n, sep, dec);`

Definición de variables

- fx:** miembro izquierdo de la ecuación después de igualarla a cero. ej: x^2-1

The taskbar at the bottom shows the Windows Start button, several application icons, and the system tray with the date '23:27' and language 'ES'.

Anexo 10: Página Web.