

**HERRAMIENTA SOFTWARE PARA LA SIMULACIÓN DE
LOS ALGORITMOS DE PROGRAMACION NO LINEAL**

**DIEGO FERNANDO RIVERA CUADROS
LUÍS RAUL GOMEZ FIGUEREDO**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA**

2008

**HERRAMIENTA SOFTWARE PARA LA SIMULACIÓN DE
LOS ALGORITMOS DE PROGRAMACIÓN NO LINEAL**

**DIEGO FERNANDO RIVERA CUADROS
LUÍS RAUL GOMEZ FIGUERO**

**DIRECTOR
JUAN CARLOS REYES**

**CODIRECTOR
HECTOR NIÑO QUIÑONEZ**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA**

2008

DEDICATORIA

*A DIOS, QUIEN ME FORTALECIO, Y ME BRINDO LA SABIDURIA
PARA PODER CUMPLIR ESTA META TAN IMPORTANTE EN
MIVIDA.*

*A MIS PADRES, QUIENES CON SU APOYO Y CONFIANZA
SUPIERON GUIARME POR EL CAMINO DE LA HONESTIDAD, PARA
ASI PODER LOGRAR CON INTEGRIDAD MIS METAS.*

*A MIS HERMANOS, POR ESTAR CONMIGO EN TODOS LOS
MOMENTOS EN LOS QUE NECESITE SU AYUDA Y COMPAÑÍA.*

*A MI NOVIA, POR ESTAR CONMIGO EN LOS MOMENTOS MÁS
IMPORTANTES EN ESTA FASE DE MI VIDA, POR BRINDARME SU
AMOR Y SU APOYO CUANDO MAS LO NECESITE.*

DIEGO R.

INTRODUCCIÓN.....	11
1. DESCRIPCIÓN DEL PROYECTO.....	12
2 MARCO TEORICO.....	14
2.1 Tecnologías de la Información y la Comunicación.....	14
2.2 Conceptos Fundamentales de los Métodos de Optimización.....	21
2.2.1 Caracterización de Máximos y Mínimos.....	22
2.2.2 Convexidad.....	23
2.2.3 Óptimos Locales y Globales.....	25
2.2.4 Condiciones de Kuhn–Tucker.....	27
2.2.5 Derivada.....	28
2.2.6 Discriminante o Matriz Hessiana.....	31
2.3 TÉCNICAS DE OPTIMIZACIÓN.....	32
2.3.1 Búsqueda de Fibonacci:.....	33
2.3.2 Búsqueda de la Sección Dorada.....	34
2.3.3 Método Gradiente Descendente:.....	35
2.3.4 Métodos de gradiente conjugado.....	35
2.3.5 Ajuste Cuadrático (Método DSC, Davies, Swann y Campey).....	36
2.3.6 Ajuste Cúbico.....	37
2.3.7 Método de Newton.....	37
2.3.8 Método Quasi-Newton.....	38
2.4 HEURÍSTICA.....	39
2.4.1 Metaheurística.....	40
2.4.2 Colonia de Hormigas (<i>Ant Colony System</i>).....	41
2.4.3 RECOCIDO SIMULADO.....	45
3. HERRAMIENTAS UTILIZADAS EN LA APLICACIÓN.....	47
3.1 PHP (Hypertext Preprocessor).....	47
3.2 BASE DE DATOS MySQL.....	47
3.3 JOOMLA.....	48
4 METODOLOGÍA DE ANÁLISIS Y DISEÑO.....	48
4.1 Ciclo de vida del desarrollo software.....	49
4.1.1 Proceso Unificado.....	50
4.1.2 Método del desarrollo por prototipos:.....	51
4.1.3 Modelo Cascada.....	51
4.1.3.1 Ingeniería y Análisis del Sistema.....	52
4.1.3.2 Análisis de los requisitos del software.....	53
4.1.3.3 Diseño.....	55
4.1.3.4 Codificación.....	56
4.1.3.5 Prueba.....	56
5 UML (Lenguaje de Modelado Unificado).....	57
5.1 Introducción.....	57
5.1.1 Objetivos del documento.....	57
5.1.2 Descripción general del proyecto.....	57
5.2 MODELO ORGANIZACIONAL.....	59
5.2.1 Definición de procesos (Automáticos y Manuales.....	59
5.3 MODELO FUNCIONAL.....	60
5.3.1 Diseño de la interfase de usuario.....	60
5.3.2 Diseño del Registro.....	60

5.3.3 Casos de uso (General).....	61
5.3.3.1 Caso de Uso (Registro)	62
5.3.3.2 Caso de Uso (Opinar en Foro)	62
5.3.3.3 Caso de Uso (Usuario)	63
5.3.3.4 Caso de Uso (Administrador).....	63
5.4 MODELO LÓGICO	64
5.4.1 Diagrama de Clases.....	64
5.4.2 Diagrama de Estados.....	64
5.4.2.1 Diagrama de Estados del Registro	64
5.4.3 Diagramas de Secuencias	65
5.4.4 Modelo de Datos	66
5.5 MODELO FÍSICO.....	66
5.5.1 Diagrama de Despliegue	66
5.5.2 Herramientas de desarrollo, ambiente de producción.....	67
5.6 SEGURIDAD DEL SISTEMA	67
5.6.1 Control de acceso al sistema.....	67
5.6.2 Control de acceso a los datos.....	67
5.6.3 Copias de seguridad	67
6 DISEÑO DE PRUEBAS	68
6.1 Descripción de escenarios.....	68
7 DESARROLLO E IMPLEMENTACIÓN.....	72
7.1 Descripción del sistema	72
7.2 Ejemplos Aplicativos.....	75
7.3 Manual del Usuario.....	81
8. CONCLUSIONES.....	90
9. RECOMENDACIONES.....	91
10. BIBLIOGRAFIA.....	92

TABLA DE FIGURAS

FIG 1: NOTACIÓN DE LEIBNIZ	29
FIG.2 AJUSTE CUADRÁTICO	36
FIG.3 MÉTODO DE NEWTON	37
FIG.4 MODELO COLONIA DE HORMIGAS.....	42
FIG.5 GRAFICA MODELO CASCADA	52
FIG.6 MODELADO DEL SISTEMA	54
FIG.7 PROCESO DE GENERACIÓN DE PRUEBAS.....	56
FIG.9 INTERFAZ DE USUARIO.....	60
FIG.10 DISEÑO DE REGISTRO.....	60
FIG 11 CASO DE USO (GENERAL).....	61
FIG 12 CASO DE USO (REGISTRO).....	62
FIG 13 CASO DE USO (OPINAR FORO).....	62
FIG 14 CASO DE USO (USUARIO).....	63
FIG 15 CASO DE USO (ADMINISTRADOR).....	63
FIG 16 DIAGRAMAS DE CLASE.....	64
FIG 17 DIAGRAMA DE ESTADO DE REGISTROS	64
FIG 18 DIAGRAMA DE SECUENCIAS	65
FIG 19 MODELO DE DATOS	66
FIG 20 DIAGRAMA DE DESPLIEGUE.....	66
FIG 21 DESCRIPCION DEL SISTEMA	72
FIG 22 BASE DE DATOS	72
FIG 23 ENCRIPCIÓN PASSWORD	73
FIG 24 CONTENIDOS EN BD.....	73
FIG 26 ADMINISTRACIÓN BD.....	74
FIG 27 MENÚ PRINCIPAL.....	81
FIG 28 RECURSOS	82
FIG 29 MÉTODOS.....	83
FIG 30 PRESENTACIÓN CONTENIDOS.....	84
FIG 31 VISTA DEL MODULO ALGORITMO	84
FIG 32 VISTA MÓDULO DIAGRAMA	84
FIG 33 VISTA MODULO ANIMACIONES	85
FIG 34 VISTA MODULO PROBLEMAS.....	85
FIG 33 VISTA SECCIÓN METAHEURÍSTICA	86
FIG 34 MODULO ADMINISTRADOR	86
FIG 35 VISTA MODULO FORMULARIO DE ACCESO.....	87
FIG 36 VISTA MODULO ENCUESTA	89

RESUMEN

TITULO

HERRAMIENTA SOFTWARE PARA LA SIMULACIÓN DE LOS ALGORITMOS DE PROGRAMACIÓN NO LINEAL. *

AUTORES

DIEGO FERNANDO RIVERA CUADROS

LUÍS RAUL GOMEZ FIGUEREDO **

PALABRAS CLAVES

Sistema de información, joomla, lenguaje php, bases de datos, Internet.

DESCRIPCIÓN

Este documento describe el análisis, diseño, desarrollo e implementación para la construcción del software que simula los algoritmos utilizados en la asignatura de programación no lineal del plan de estudio de ingeniería de sistemas. Dicho software se diseñó como complemento a las clases presenciales de la materia, y como soporte de autoaprendizaje para personas interesadas en el tema.

El software consta básicamente de actividades de aprendizaje, aplicaciones, y el manual del usuario.

Las actividades de aprendizaje son herramientas que permiten al estudiante:

- Interactuar con el software de forma que pueda intercambiar dudas, ideas e información; evaluarse, desarrollar tareas y crear sus propias definiciones de conceptos básicos de programación no lineal.

Las aplicaciones son ejemplos desarrollados como componentes del paquete para mostrar paso a paso el comportamiento de los diferentes algoritmos y se encuentran definidos en tutoriales. En estos tutoriales se encuentran definidos los contenidos principales de los temas estudiados, los cuales vienen ilustrados como animaciones y gráficos.

En los manuales lo que se pretende es, guiar a los usuarios sobre el manejo y comprensión de los módulos del software, y del funcionamiento total del sistema.

Este proyecto fue desarrollado bajo un software de gestión de contenidos (joomla), el cual maneja el lenguaje de programación php y bases de datos MySQL. El software se basó en la metodología **cascada** (Ciclo de vida del Software), la cual permitió desarrollar fase a fase todo el diseño, desarrollo e implementación del sistema.

* Trabajo de Investigación.

** Facultad de Ingeniería Física – Mecánicas, Escuela de Ingeniería de Sistemas e Informática. Director: Ing. Juan Carlos Reyes.

SUMMARY

TITLE

SOFTWARE TOOLS FOR SIMULATING ALGORITHMS NONLINEAR PROGRAMMING *

AUTHORS

DIEGO FERNANDO RIVERA CUADROS
LUÍS RAUL GOMEZ FIGUEREDO **

KEY WORDS

Sistema de información, joomla, lenguaje php, bases de datos, Internet.

DESCRIPTION

This document describes the analysis, design, development and implementation for building software that simulates the algorithms used in the course of non-linear programming from the plan study of systems engineering. This software was designed as complement to the classroom courses, and as support for individual learning to people interested in the topic.

The software basically consists of learning activities, applications and the user manual.

The learning activities are tools that allow the student to:

- Interacting with the software in a way that you can share concerns, ideas and information; itself evaluation, perform tasks and create its own definitions of basic concepts of non-linear programming.

Applications are examples developed as components of the package to show step by step the behaviour of the different algorithms and are defined in tutorials. In these tutorials are defined the main contents of the subjects studied, which are illustrated as animations and graphics.

In the manuals the aim is, to guide users on the management and understanding of the software modules, and the total system operation.

This project was developed under content management software (joomla), which handles the programming language PHP and data bases MySQL. The software was based on the waterfall methodology (Software Lifecycle), which helped to develop stage to stage the whole design; development and implementation of the system.

* Research Working

** Faculty of Physical – Mechanics Engineering's, Systems and Informatic Engineering
Director: Ing. Juan Carlos Reyes

INTRODUCCIÓN

El aprendizaje en línea o e-learning en los últimos años ha logrado ubicarse en una posición de significativa importancia dentro del marco académico y empresarial mundial, mostrando cada vez con mayor claridad una serie de interesantes posibilidades en el mejoramiento continuo de los procesos heterogéneos de generación y captación de información y conocimiento.

Es el objetivo de la labor de investigación que aquí se expone, apoyar estos procesos de enseñanza y aprendizaje, con el desarrollo de una herramienta computacional que sea a su vez generadora de nuevas herramientas que hagan cada vez más enriquecedora y efectiva esta labor de educar y aprender.

Las TIC juegan un papel muy importante, ya que son un conjunto de servicios que mejoran la calidad de vida dentro de un entorno, e integran un sistema a otros rompiendo los límites que existen entre estos.

Las tecnologías como tal no se encargan del estudio, desarrollo, implementación, almacenamiento y distribución de la información mediante la utilización de hardware y software como recursos de los sistemas informáticos.

Como concepto no informático se refieren a saberes que hacen referencia a la utilización de múltiples medios informáticos para almacenar, procesar y difundir todo tipo de información, telemática, etc. con diferentes finalidades (formación educativa, organización y gestión empresarial, toma de decisiones en general, etc.).

El tema significativo de esta tesis son los algoritmos utilizados en la asignatura programación no lineal y su estructura como son: tutoriales, manuales y aplicaciones las cuales nos garantizan un aprovechamiento del conocimiento de manera cooperativa y colaborativa haciendo uso de estas TIC's (Tecnologías de la Información y la Comunicación).

1. DESCRIPCIÓN DEL PROYECTO

TÍTULO DEL PROYECTO

Herramienta software para la simulación de algoritmos de programación no lineal

OBJETIVOS

OBJETIVO GENERAL

Desarrollar e implementar una metodología para apoyar el aprendizaje significativo de los estudiantes de la materia programación no lineal en la escuela de ingeniería de sistemas e informática (EISI).

OBJETIVOS ESPECÍFICOS

Entender la programación no lineal como marco de referencia de problemas de optimización, reconocer las aplicaciones típicas, y comprender los fundamentos teóricos de los diferentes algoritmos para aplicar el algoritmo adecuado al problema bajo estudio.

Desarrollar una herramienta software que permita:

- Crear un soporte para el trabajo en el área de programación no lineal, que incluye sistema de entrada de datos, salida de los mismos y análisis gráficos
- Contener los métodos cuantitativos más usados en las ciencias administrativas y la administración de operaciones.
- Por medio de un interfaz que los profesores y estudiantes tengan un fácil acceso a los diferentes módulos de decisión para resolver problemas.
- Ofrecer la suficiente documentación de los módulos para el entendimiento y utilización óptima de la herramienta.

JUSTIFICACIÓN

La falta de herramientas software que apoyen las prácticas de los algoritmos sobre programación no lineal, impide el eficaz desarrollo de los contenidos teóricos y prácticos que se ofrecen en la materia que lleva el mismo nombre; bien sea por la falta de plataformas que permitan la implementación de los algoritmos o por programas que no visualizan los resultados deseados y ordenados para su posterior análisis.

Considerando que la tecnología esta en pro del mejoramiento continuo y no debe ser una barrera para el ejercicio del razonamiento, se propone la elaboración del material necesario con la ventajas descritas a continuación: Una herramienta integrada que dará la posibilidad a los usuarios de unir la teoría con la práctica, eficientemente; plantear soluciones a problemas a partir de los datos obtenidos y vislumbrar extensiones de los conocimientos. Un paquete ajustado y de libre distribución contribuirá al desarrollo de las temáticas de la programación no lineal, además de ajustarse a los nuevos modelos de pedagogía y entornos propuestos para la educación.

IMPACTO

Se espera que con la ayuda de esta herramienta en el momento de que todos los usuarios ingresen y hagan uso de esta, puedan tener una idea más clara de lo que significan y en que se pueden aplicar los algoritmos usados en la programación no lineal, también se espera que esta herramienta sea de gran utilidad y ayuda en el desarrollo de los temas vistos en el área de programación no lineal.

Los usuarios tendrán la oportunidad de encontrar toda la información necesaria para todas las consultas y dudas que se tengan de los temas propuestos, desde cualquier punto donde se tenga acceso a Internet, lo cual ayudara a difundir de una manera diferente a la tradicional estos temas, debido a que tiene muchos ejemplos y animaciones lo suficientemente claros y con la mayor documentación posible, ayudando a que esta metodología sea mas constructiva y mas significativa para los estudiantes.

VIABILIDAD

El software cubrirá una falencia que hay en el área de PNL puesto que no existe una herramienta al alcance del estudiantado para complementar el aprendizaje en esta materia, la herramienta tendrá una interfaz grafica sencilla y eficiente para el fácil manejo de los diferentes módulos presentados en esta, se tendrá beneficios como el de tener un espacio para los estudiantes además de las clases teóricas para complementar su aprendizaje y afianzar sus conocimientos.

La herramienta a desarrollar será compatible con cualquier sistema operativo, lo cual hará que haya gran facilidad en su uso en cualquier ambiente, ya que se desarrollara sobre la plataforma Java SE 6 que hace que el desarrollo sea más fácil, rápido y a menor costo y ofrece mayor funcionalidad para servicios Web, soporta lenguajes dinámicos, diagnósticos y aplicaciones de escritorio.

Al trabajar con el software Java se tendrá la oportunidad de incluir un conjunto de servicios Web client si los beneficiarios lo requieren. Las herramientas a utilizar en el proyecto son software libre, y desde luego influirá en que los costos sean bajos en el desarrollo del proyecto.

La población estudiantil será beneficiada con nuestro proyecto puesto que tendrán la oportunidad de manejar un software que no necesita licencias, y se evitara la utilización de software con elevados costos, además este será accesible para cualquier persona.

2 MARCO TEORICO

2.1 Tecnologías de la Información y la Comunicación

Las Tecnologías de la Información y la Comunicación (TIC), se encargan del estudio, desarrollo, implementación, almacenamiento y distribución de la información mediante la utilización de hardware y software como medio de sistema informático.

Las tecnologías de la información y la comunicación son una parte de las tecnologías emergentes que habitualmente suelen identificarse con las siglas TIC y que hacen referencia a la utilización de medios informáticos para almacenar, procesar y difundir todo tipo de información o procesos de formación educativa.

Definición. Según la Asociación americana de las tecnologías de la información (Information Technology Association of America, ITAA): sería “el estudio, el diseño, el desarrollo, el fomento, el mantenimiento y la administración de la información por medio de sistemas informáticos, esto incluye todos los sistemas informáticos no solamente la computadora, este es solo un medio más, el más versátil, pero no el único; también los teléfonos celulares, la televisión, la radio, los periódicos digitales, etc.”

En pocas palabras, las Tecnologías de la información tratan sobre el empleo de computadoras y aplicaciones informáticas para transformar, almacenar, gestionar, proteger, difundir y localizar los datos necesarios para cualquier actividad humana.

La instrumentación tecnológica es una prioridad en la comunicación de hoy en día, debido a que las tecnologías de la comunicación son la diferencia entre una civilización desarrollada y otra en vías de desarrollo. Éstas poseen la característica de ayudar a comunicarnos porque se desaparecen las distancias geográficas y el tiempo.

Las TIC optimizan el manejo de la información y el desarrollo de la comunicación. Permiten actuar sobre la información y generar mayor conocimiento e inteligencia. Abarcan todos los ámbitos de la experiencia humana. Están en todas partes y modifican los ámbitos de la experiencia cotidiana: el trabajo, las formas de estudiar, las modalidades para comprar y vender, los trámites, el aprendizaje y el acceso a la salud, entre otros.

En definitiva, las actividades que implican el desarrollo humano dependen de cómo la gente domine las TIC. En muchos países, las TIC ya están inmersas en diferentes actividades rutinarias:

Procesos. El sufragio virtual y la identificación digital a través de la pupila o de la huella digital son un claro ejemplo de participación y seguridad por las TIC.

Métodos. Las colas que sistematizan la atención del público en algunos bancos, el uso de métodos estadísticos para la toma de decisiones y las líneas de montaje en algunas fábricas ahorran esfuerzo, tiempo y dinero a empresas y personas.

Organizaciones. Las TIC han fomentado la creación de la reingeniería, de las normas ISO y de centros especializados en la estandarización de operaciones.

Para medir el desarrollo de las TIC en un determinado país o región existen los indicadores TIC, un conjunto de medidas, variables y parámetros que buscan dar con el ambiente para el desarrollo y construcción de la Sociedad de la Información, labor que está a cargo de los organismos públicos y privados en el mundo.

Para el desarrollo particular de las personas, las herramientas TIC juegan en las sociedades un papel clave que permite logros antes ni soñados.

Uno de estos instrumentos es el aprendizaje-e, es decir el uso de las tecnologías multimedia y la internet para mejorar la calidad del aprendizaje, hacerlo accesible a la gente que no está cerca a instituciones educativas y poner a disposición de todos innovadoras formas de educación en cualquier ambiente en que uno esté.

Otro es el gobierno-e, que se refiere al empleo del Internet y las TIC para conseguir una mejor administración del gobierno mediante la transparencia y el acceso público a la información, reforzando la asociación fundamental entre el sector público y los ciudadanos. El gobierno-e también fomenta una participación más amplia de los ciudadanos en el proceso y gestión de los gobiernos. Además, puesto que se apoya en la transparencia, es un arma más eficaz contra la corrupción.

También está el comercio-e o comercio electrónico, que tiene que ver con el intercambio de bienes y servicios realizado gracias a un soporte de protocolos y plataformas digitales estandarizadas, lo que permite llegar a acuerdos sin tener que estar presente, un gran ahorro en dinero y tiempo.

También se constituye en una herramienta que permite el acceso a información sobre precios, oferta y demanda, para que compradores y productores obtengan los mejores precios.

Pero ninguno de estos beneficios será posible en su plenitud en América Latina y el Caribe si no se da una asociación entre el sector público, el privado y la sociedad civil para promover el desarrollo de las TIC. El sector público tiene que estudiar la manera de corregir el fracaso de los mercados y alentar la competencia para que la Sociedad de la Información sea una realidad. El sector privado juega un papel importante por sus inversiones en las TIC. Finalmente, la sociedad civil tiene que trabajar en estrecha relación con las comunidades para reforzar toda iniciativa.

Esta alianza es fundamental para realizar las transformaciones que necesitan los países de la región. Invertir en tecnología no significa descuidar las demás necesidades prioritarias. Por el contrario, esta inversión permite alcanzar e incluso sobrepasar eficazmente las metas en educación, salud y otros sectores estratégicos.

Beneficios y ventajas de las TIC

El beneficio de las TIC dependerá, en gran medida, de cómo las use una determinada comunidad y cuánta importancia les otorgue en su desarrollo. De todos modos, parece claro que vivimos en tiempos en los que la máxima creatividad del hombre puede marcar la diferencia, porque la nueva economía ya no está tan centrada en los recursos naturales ni en las materias primas, sino en los flujos electrónicos de información. En esos términos, los beneficios podrían ser los siguientes:

- Facilitan las comunicaciones.
- Eliminan las barreras de tiempo y espacio.
- Favorecen la cooperación y colaboración entre distintas entidades.
- Aumentan la producción de bienes y servicios de valor agregado.
- Potencialmente, elevan la calidad de vida de los individuos.
- Provocan el surgimiento de nuevas profesiones y mercados.
- Reducen los impactos nocivos al medio ambiente al disminuir el consumo de papel y la tala de árboles y al reducir la necesidad de transporte físico y la contaminación que éste pueda producir.
- Aumentan las respuestas innovadoras a los retos del futuro.
- La Internet, como herramienta estándar de comunicación, permite un acceso igualitario a la información y al conocimiento.

Como conclusión se puede insistir en que son la calidad y la forma en las que los contenidos son producidos, transmitidos y percibidos por las personas, lo que garantiza el real aprovechamiento de las TIC en todas las áreas de la vida cotidiana. Las TIC bien utilizadas pueden generar nuevas oportunidades de acceso a la información, crear capacidades, mejorar la productividad, impulsar el

desarrollo y, en definitiva, permitir avanzar en la creación de igualdad de opciones. Y es que las TIC son una herramienta cada vez más poderosa, pues participan en los mercados mundiales, promueven una gestión política más transparente y responsable, mejoran la prestación de servicios básicos y aumentan las oportunidades.

Sin embargo, se debe insistir una y otra vez en el hecho de que las TIC deben ser vistas sólo como herramientas que deben estar al servicio del ser humano y, por tanto, son un medio y no un fin en sí mismas. Únicamente viéndolas de ese modo se podrá aprovecharlas al máximo en función de una mayor calidad de vida de las personas, que implique una mejor educación, un mayor acceso a los servicios de salud y más oportunidades económicas en cuanto a empleo.

CONTEXTO EN COLOMBIA

Las Tecnologías de Información y Comunicación (TIC) han cambiado la manera de relacionarnos con otros, revolucionado muchos campos de la actividad humana. La presencia física ya no es una condición necesaria para estar en contacto con personas y proveedores de información relacionada con nuestra actividad personal o profesional, dado que haciendo uso de TIC podemos alcanzar, a precios razonables y con creciente eficacia, la interacción requerida con quienes deseamos comunicarnos.

La toma de decisiones apoyada en información apropiada no es privilegio de pocos, toda vez que hay redes globales o locales que permiten consultar distintas fuentes, que es posible la interacción sincrónica o asincrónica con otras personas o grupos que pueden saber sobre lo que nos interesa, que es posible almacenar, procesar y analizar datos e información pertinente y oportuna, así como crear, alimentar y depurar modelos mentales, que se plasman en modelos computarizados.

Sin embargo, pareciera que es poco el valor que se agrega con TIC a los procesos educativos, a pesar de que hay una creciente oferta de recursos informáticos. Contar con equipos y de redes en educación, al igual que en cualquier dominio, no es lo que hace la diferencia, sino lo que se hace con ellos.

Los educadores y la informática

Hay educadores que siguen aferrados a esquemas que se centran en transmitir y asimilar conocimiento comprobado, con lo que para ellos las TIC son una buena ocasión para poner a disposición de más gente y con recursos multimedia, aquello que cree que deben enseñar. Y su gran reto es hallar buenas herramientas para difundir mejor lo que saben o lo que desean que aprendan sus alumnos.

También tienen el reto de hallar los mejores medios para hacer llegar a los destinatarios lo que han mediatizado, dependiendo de su distribución geográfica y del acceso que tengan a recursos informáticos.

Hay educadores que piensan que la informática es un recurso a cargo de los que saben de eso, para complementar lo que ellos hacen en sus clases. De vez en cuando llevan a sus estudiantes a la sala de informática, para que el encargado los “ponga a hacer” alguna actividad apoyada en los recursos informáticos disponibles (juegos, lenguajes de programación, sistema operativo, herramientas de productividad, navegadores y buscadores en la Internet).

Los recursos informáticos se convierten así en una “niñera digital” que entretiene a los alumnos o pasan a ser un “objeto de estudio” y no una herramienta de trabajo.

Los docentes que han decidido apoyar aprendizaje por descubrimiento mediante indagación y colaboración pueden hallar en las TIC oportunidades de ayudar a sus alumnos a vivir experiencias directas interactuando con el objeto del conocimiento, con modelos del mismo, con personas que tienen distintas perspectivas sobre el mismo, así como a construir y expresar sus propios modelos mentales acerca de lo que estudian.

El gran reto no es que el docente halle y apropie las TIC que permitan hacer esto, aunque esto ayuda, sino proponer los ambientes de aprendizaje que propicien lo que se desea, integrando recursos de aprendizaje que puedan jugar distintos roles complementario.

CLASIFICACIÓN DE TIC TOMANDO EN CUENTA TIPOS DE MEDIOS Y ENFOQUES EDUCATIVOS

El contexto anterior nos permite entender por qué no podemos hablar de TIC sin tener en cuenta el enfoque educativo que queremos usar, así como las características fundamentales de los distintos tipos de medios y el nivel de cultura informática que pueden tener nativos e inmigrantes digitales. Los enfoques educativos algorítmicos y heurísticos, que son opuestos pero

complementarios, se pueden instrumentar con tres tipos de medios transmisivos, activos e interactivos, que también se complementan.

De este modo, al clasificar las TIC tomando en cuenta estas dimensiones, estamos reconociendo sus propiedades fundamentales como medio e indicamos la posibilidad que tienen de apoyar el enfoque educativo al que son más cercanas, pero enfatizamos el papel vital que tiene quien facilita el proceso, y el enfoque que usa para hacerlo.

TIC's	EJEMPLO DE TIC
TIC eminentemente transmisiva	<ul style="list-style-type: none"> • Tutoriales para apropiación y afianzamiento de contenidos. • Ejercitadores de reglas o principios, con retroalimentación directa o indirecta. • Bibliotecas digitales, videotecas digitales, audiotecas digitales, enciclopedias digitales. • Sitios en la red para recopilación y distribución de información. • Sistemas de automatización de procesos, que ejecutan lo esperado dependiendo del estado de variables indicadoras del estado del sistema.

Enfoque algorítmico

Centrado en el que enseña, sigue reglas para una efectiva transmisión del que sabe hacia los que desean aprender, hay facilitación desde el centro.

Enfoque heurístico

Centrado en el que aprende, sigue principios que orientan la construcción individual o colaborativa de conocimiento, hay facilitación desde el lado.

Medios transmisivos

Buscan apoyar la entrega efectiva de mensajes del emisor a los destinatarios.

Medios activos

Buscan permitir que quien aprende actúe sobre el objeto de estudio y, a partir de la experiencia y reflexión, genere y afine sus ideas sobre el conocimiento que subyace a dicho objeto.

Medios interactivos

Buscan permitir que el aprendizaje se dé a partir de diálogo constructivo, sincrónico o asincrónico, entre co-aprendices que usan medios digitales para comunicarse.

El proceso de invención y desarrollo de las TIC no está concluido. Las TIC son un mundo en evolución que combina los adelantos de muchas disciplinas. Nuestro granito de arena, como educadores, está ligado a la creación y prueba reflexiva de ambientes que hagan el mejor uso de estas oportunidades. Que sea ésta ocasión de unirnos al esfuerzo global para enriquecer la educación con el apoyo de la informática.

En cuanto al uso educativo de las TIC, parece claro que éstas abren nuevas posibilidades, pero **a condición de que:**

- superemos las orientaciones tecnocrática (los medios por los medios) y míticas (la salvación por la vía de las nuevas tecnologías)
- insertemos el uso de las TIC en una **pedagogía diferente a la habitual**, preocupada sólo por la transmisión pasiva y/o la destreza técnica, evitando usar las TIC sólo como medio novedoso o con finalidad en sí mismas
- tengamos en cuenta la **desigualdad social y territorial**, que las TIC pueden aumentar (por ejemplo, si el acceso a Internet favorece el aprendizaje en un área, pero unas personas lo tienen en el hogar y otras no); el objetivo igualitario ha de ser fundamental y permanente, tanto en la forma de usar las TIC en la educación como en la pelea por su democratización social
- no olvidemos las **diferentes sensibilidades y formas de aprendizaje** del alumnado (por ejemplo, diferentes formas de acercamiento y trabajo entre chicos y chicas, en algunos casos al menos)
- utilicemos las TIC **relacionando críticamente lo `real y lo `virtual**, de tal forma que se enriquezcan ambos contextos
- no olvidemos que el uso de las TIC significa, en algunos sentidos al menos, un aumento de la **`dependencia tecnológica**, lo que implica **limitaciones** prácticas, socioeconómicas y personales.

2.2 Conceptos Fundamentales de los Métodos de Optimización

Motivación para estudiar formas de optimización:

Existe una enorme variedad de actividades en el mundo cotidiano que pueden ser útilmente descritas como sistemas, desde sistemas físicos tales como una planta industrial hasta entidades teóricas tales como los modelos económicos. La operación eficiente de esos sistemas usualmente requiere un intento por optimizar varios índices que miden el desempeño del sistema. Algunas veces, esos índices son cuantificados y representados como variables algebraicas. Entonces se deben encontrar valores para esas variables, que maximicen la ganancia o beneficio del sistema, o bien minimicen los gastos o pérdidas. Se asume que las variables dependen de ciertos factores.

Algunos de esos factores a veces están bajo el control (al menos parcialmente) del analista responsable del desempeño del sistema

En la formulación de muchos problemas de optimización no se puede hacer la hipótesis de linealidad que caracteriza a la programación lineal. No existen procedimientos generales para problemas no lineales. Se han desarrollado un gran número de algoritmos especiales para tratar casos particulares. Muchos de esos procedimientos se basan en la teoría matemática relacionada con el análisis de la estructura de tales problemas. Esta teoría se llama generalmente optimización clásica.

Una de las principales contribuciones modernas a esta teoría fue hecha por Kuhn y Tucker quienes desarrollaron lo que se conoce como las condiciones de Kuhn – Tucker.

La colección de técnicas desarrolladas por esta teoría se llama programación no lineal. Pese a que muchos problemas de programación no lineal son muy difíciles de resolver, hay cierto número de problemas prácticos que pueden ser formulados de manera no lineal y resueltos por los métodos existentes. Esto incluye el diseño de entidades tales como transformadores eléctricos, procesos químicos, condensadores de vapor y filtros digitales.

Optimización como una rama de la matemática

Se puede ver, que la teoría de la optimización es matemática por naturaleza. Típicamente involucra la maximización o minimización de una función (a veces desconocida) que representa el desempeño de algún sistema.

Esto se resuelve encontrando los valores de las variables (cuantificables y controlables) que hacen que la función alcance su mejor valor. A fin de entender como operan los algoritmos se requieren conocimientos de álgebra lineal y cálculo diferencial con varias variables.

Algunos de los problemas de la teoría de optimización se pueden resolver por las técnicas clásicas del cálculo avanzado (tales como métodos Jacobianos y el uso de multiplicadores de Lagrange). Sin embargo, la mayoría de los problemas de optimización no satisfacen las condiciones necesarias para ser resueltos de esta manera. Muchos de los otros problemas, pese a poder ser tratados con las técnicas clásicas, se resuelven más eficazmente si se utilizan métodos diseñados para cada caso particular. A través de la historia de las matemáticas se ha construido una colección de tales técnicas. Algunas han sido olvidadas y reinventadas, otras recibieron poca atención hasta que las computadoras las hicieron utilizables.

El grueso de material al respecto es de origen reciente debido a que muchos de los problemas, tales como el flujo de tráfico, recién ahora cobran interés y también debido al gran número de investigadores disponibles actualmente para analizar tales problemas. Cuando ese material es catalogado dentro de un cuerpo autocontenido de conocimientos el resultado es una nueva rama de las matemáticas aplicadas.

2.2.1 Caracterización de Máximos y Mínimos

Máximos y Mínimos

Recordemos que f derivable, es estrictamente creciente (decreciente) en a si, y sólo si $f'(a) > 0$ ($f'(a) < 0$); lo que geoméricamente significa que la pendiente de la recta tangente en dicho punto es positiva (negativa).

Recordemos también que si f derivable posee un máximo o un mínimo relativo en entonces $f'(x) = 0$; es decir, ese es un punto de tangente horizontal.

Veamos los criterios básicos para decidir si un punto es máximo o mínimo relativo:

Por la definición en un entorno del punto.

Por la variación del signo de la derivada primera en un entorno del punto, aunque la función no sea derivable en dicho punto:

f decreciente en (a, c) y creciente en (c, b) posee un mínimo en $(c, f(c))$.

f creciente en (a, c) y decreciente en (c, b) posee un máximo en $(c, f(c))$.

Por el signo de la derivada segunda en dicho punto (la función ha de ser dos veces derivable).

Si $f'(a) = 0$ y $f''(x) > 0$, f posee en a un mínimo local.

Si $f'(a) = 0$ y $f''(x) < 0$, f posee en a un máximo local.

2.2.2 Convexidad

Se puede ver a continuación las propiedades más importantes de los conjuntos convexos y de las funciones convexas, tales conjuntos y funciones juegan un rol muy importante en la optimización por dos motivos:

Cuando se tienen funciones y conjuntos convexos se puede identificar con seguridad los llamados “óptimos globales”.

Muchas operaciones y transformaciones conservan la convexidad, por lo que se puede construir funciones y conjuntos convexos más complicados a partir de los más sencillos.

Definición: Combinación Convexa.

Dados $(x, y) \in \mathfrak{R}^n$, una *combinación convexa* de ellos es cualquier punto de la forma:

$$z = \alpha x + (1 - \alpha) y$$

Con $\alpha \in \mathfrak{R} \mid 0 \leq \alpha \leq 1$

- Si $\alpha \neq 0$ y $\alpha \neq 1$, se dice que z es una combinación convexa estricta (o propia).
- La interpretación geométrica es que z es un punto del segmento de recta determinado por (x, y) .

Definición: Conjunto Convexo.

$S \subset \mathfrak{R}^n$ es un *conjunto convexo* si

$$[\alpha x + (1 - \alpha) y] \in S$$

$$\forall (x, y) \in S, \forall \alpha \in \mathfrak{R}: 0 \leq \alpha \leq 1$$

Definición: Función Convexa (en un conjunto convexo).

Sea $S \subseteq \mathfrak{R}^n$, un conjunto convexo.

La función $f: S \rightarrow \mathfrak{R}$ es convexa en S si:

$$f[\alpha x + (1 - \alpha) y] \leq \alpha f(x) + (1 - \alpha) f(y)$$

$$\forall (x, y) \in S, \forall \alpha \in \mathfrak{R}: 0 \leq \alpha \leq 1$$

Si $S = \mathfrak{R}^n$, se dice que f es *convexa*.

Composición de conjuntos y funciones convexas

La importancia de las funciones y conjuntos convexas radica en que las condiciones necesarias para que una solución sea un óptimo local se convierten en condiciones suficientes para que sea óptimo global, cuando las funciones y conjuntos en cuestión son convexas.

Sean S y T conjuntos convexas, $\alpha \in \mathfrak{R}$ y f una función convexa. Entonces los siguientes conjuntos también son convexas:

- i. $\alpha S = \{\alpha x \mid x \in S\}$ iii. $S + T = \{x+y \mid x \in S, y \in T\}$ v. $\{(x, \mu) \mid f(x) \leq \mu\}$
 ii. $S \cap T$ iv. $\{x \mid f(x) \leq \alpha\}$

Sean f y g funciones convexas en un conjunto convexo S , y $\alpha \geq 0$.

Entonces las siguientes funciones también son convexas en S .

1. $h_1(x) = f(x) + g(x)$
2. $h_2(x) = \alpha f(x)$
3. $h_3(x) = \max\{f(x), g(x)\}$

Sean $f_i: I_i \rightarrow \mathfrak{R}$, ($i = 1, \dots, n$) funciones convexas en el intervalo $I_i \subseteq \mathfrak{R}$.

$f(x) = \sum_{i=1}^n f_i(x)$ es una función convexa en $I = \{x \in \mathfrak{R}^n \mid x_j \in I_j\}$

Funciones convexas reales, de una sola variable

Para poder utilizar los resultados anteriormente expuestos es necesario poder reconocer las funciones convexas. Por tal motivo, en primer lugar se analizarán las funciones de una sola variable.

Una función $f: S \rightarrow \mathfrak{R}$ es convexa en el conjunto convexo S , si:

$$f((1-\lambda)x + \lambda y) \leq (1-\lambda)f(x) + \lambda f(y)$$

$$\forall x, y \in S, \text{ con } \lambda \geq 0 \text{ o bien } \lambda \leq 1.$$

Una función diferenciable $f: I \rightarrow \mathfrak{R}$ con $I \subseteq \mathfrak{R}$ es convexa si:

$$f(y) \geq f(x) + f'(x)(y-x) \\ \forall x, y \in I$$

Una función diferenciable f en un intervalo $I \subseteq \mathfrak{R}$ es convexa si f' es no decreciente.

Una función f diferenciable hasta el segundo orden en un intervalo $I \subseteq \mathfrak{R}$ es convexa sii $f'' \geq 0$ en I

Funciones convexas en \mathfrak{R}^n

Una función diferenciable f en un conjunto convexo S es convexa sii

$$f(y) \geq f(x) + \nabla f(x) \cdot (y - x), \forall x, y \in S$$

Una función f diferenciable hasta el segundo orden en $S \subseteq \mathfrak{R}^n$ convexo y abierto

es convexa sii la matriz de derivadas segundas: $\left(\frac{\partial^2 f}{\partial x_i \partial x_j} \right)$ es semidefinida positiva en \mathfrak{R}^n .

2.2.3 Óptimos Locales y Globales

En este capítulo se estudian las propiedades de las soluciones óptimas. Se brinda además un conjunto de condiciones necesarias y suficientes para que una solución sea local o globalmente óptima.

Óptimos locales y globales, direcciones factibles y direcciones de descenso

La siguiente es la forma general de un problema de optimización:

$$G(x) \begin{cases} \text{Minim } f(x) \\ \text{Sujeto } a : \\ x \in F \end{cases}$$

Y los problemas de programación matemática, tienen la forma siguiente:

$$M(x) \begin{cases} \text{Minim } f(x) \\ \text{Sujeto } a : \\ g(x) \leq 0 \\ x \in X \end{cases}$$

Todos los problemas de optimización pueden ser escritos en la forma general (G).

Se supondrá que $F \subseteq \mathfrak{R}^n$ y $f: F \rightarrow \mathfrak{R}$, donde F es la región factible y f la función objetivo.

En un problema de programación matemática se supone que:

$$X \subseteq \mathfrak{R}^n, \quad g: X \rightarrow \mathfrak{R}^m \quad \text{y} \quad f: X \rightarrow \mathfrak{R}.$$

En el problema (M) la región factible es el conjunto: $\{x \in X \mid g(x) \leq 0\}$

Definición: $x^* \in F$ es un óptimo global de (G) sii $f(x) \geq f(x^*) \quad \forall x \in F$

Definición: $x^* \in F$ es un óptimo local de (G) sii $\exists \varepsilon > 0$, tal que se cumple que $f(x) \geq f(x^*) \quad \forall x \in F$, con $\|x - x^*\| < \varepsilon$.

La mayoría de los métodos para hallar un óptimo son por búsqueda direccional: partiendo de un punto de F , se define una dirección de búsqueda y se determina un nuevo punto a lo largo de ésta. Por ese motivo son de interés los conceptos de dirección factible y dirección de descenso.

Condiciones necesarias y suficientes para óptimos

Con las definiciones que preceden, más las de convexidad, se puede definir un conjunto de condiciones de optimalidad. Previamente se deberá distinguir entre condición necesaria y condición suficiente.

Una condición necesaria para un óptimo (local) es una condición que se cumple en cada solución óptima, aunque también podría cumplirse en otras soluciones.

Puede ser utilizada para descartar, en la búsqueda del óptimo, aquellas soluciones que no cumplen la condición necesaria de optimalidad, o limitar a priori la búsqueda a aquellas soluciones que cumplen dicha condición.

Una condición suficiente para un óptimo (local) es una condición que si se cumple garantiza que la solución es óptima (local).

Se puede utilizar en un procedimiento en el que se trata de lograr que la condición se cumpla, para de ese modo obtener una solución óptima.

Teorema: Si $x^* \in F$ es un óptimo local de (G) no existe en ese punto ninguna dirección de descenso que además sea factible.

En consecuencia: una condición necesaria para que $x^* \in F$ sea un óptimo local, es que en ese punto no exista ninguna dirección factible de descenso.

Teorema: Si en un problema del tipo (G), la función objetivo f es diferenciable, y $x^* \in F$ es un óptimo local de (G), entonces $\nabla f(x^*) \cdot d \geq 0$ para todas las direcciones factibles d en x^* .

Corolario: Si la función objetivo f de (G) es diferenciable, y x^* es un punto interior de F que cumple ser óptimo local, entonces: $\nabla f(x^*) = 0$.

Teorema: Si la función objetivo f , y la región factible F de (G) son convexas y además no existe una dirección factible de descenso en $x^* \in F$, entonces x^* es un óptimo global del problema (G).

Nótese que el teorema precedente establece una condición suficiente de optimalidad para un problema en que la función objetivo y la región factible son convexas.

Sean f y F convexas en el problema (G), con f diferenciable. Si existe $x^* \in F$ tal que $\nabla f(x^*) \cdot d \geq 0$ para todas las direcciones factibles d en x^* , entonces x^* es un óptimo global de (G).

2.2.4 Condiciones de Kuhn–Tucker

Para problemas de programación matemática también se pueden aplicar los teoremas anteriores. Para este tipo de problemas existen además condiciones de optimalidad especiales, las llamadas condiciones de Kuhn–Tucker. Estas serán introducidas a través de una analogía con la mecánica.

Considérese el siguiente problema:

$$M(x) \begin{cases} \text{Minim } f(x) \\ \text{Sujeto a:} \\ g_i(x) \leq 0 \quad i = 1, \dots, m \\ x \in X \end{cases}$$

Los óptimos locales de (M) pueden ser vistos como las posiciones de equilibrio estable de una partícula que se mueve en un campo de fuerza cuyo potencial es f , en una región X , y condicionada a moverse en el interior de una región delimitada por ciertas "paredes": $V_i = \{x \mid g_i(x) = 0\}$.

Supóngase que la partícula está en reposo en \bar{x} , punto interior de X .

La partícula está expuesta a fuerzas de reacción r_i perpendiculares a las paredes y orientadas hacia el interior, como los gradientes ∇g_i son también perpendiculares a las paredes, pero orientados hacia el exterior, las fuerzas reactivas de las paredes pueden expresarse como: $r_i = -\lambda_i \nabla g_i(\bar{x})$ para parámetros $\lambda_i \geq 0$, suponiendo que $\nabla g_i(\bar{x}) \neq 0$.

Las fuerzas reactivas sólo actúan en las paredes que están en contacto con la partícula, es decir que $\lambda_i \neq 0$ solamente cuando $g_i(\bar{x}) = 0$. Esto se expresa en la forma siguiente:

$$\lambda_i g_i(\bar{x}) = 0, \quad i = 1, \dots, m.$$

Finalmente, para que la partícula esté en reposo se exige balance de las fuerzas:

$$-\nabla f(x) + \sum_{i=1}^m (-\lambda_i \nabla g_i(x)) = 0$$

El resumen de estas exigencias son las Condiciones de Kuhn – Tucker:

$$\begin{aligned} g(\bar{x}) &\leq 0, && \text{Es decir: que el punto esté en la región factible.} \\ \lambda_i g_i(\bar{x}) &= 0, && i = 1, \dots, m, \text{ fuerza sólo en las paredes "activas".} \\ \lambda_i &\geq 0 && \text{Las fuerzas apuntan hacia el interior de la región factible.} \\ \nabla f(\bar{x}) + \sum \lambda_i \nabla g_i(\bar{x}) &= 0, && \text{Equilibrio de fuerzas.} \end{aligned}$$

Teorema de Kuhn – Tucker:

Sea $\bar{x} \in X$ un punto interior que es óptimo local del problema (M), y sean f y g_i funciones diferenciables. Si se cumplen las “constraint qualifications” apropiadas existe $\bar{\lambda} \in \mathfrak{R}^m$ que satisface las siguientes condiciones de Kuhn – Tucker:

$$\begin{aligned} g(\bar{x}) &\leq 0 \\ \lambda_i g_i(\bar{x}) &= 0, \quad i = 1, \dots, m \\ \lambda &\geq 0 \\ \nabla f(\bar{x}) + \sum \bar{\lambda}_i \nabla g_i(\bar{x}) &= 0 \end{aligned}$$

2.2.5 Derivada

Geoméricamente, la derivada de una función en un punto es el valor de la pendiente de la recta tangente en dicho punto. La pendiente está dada por la tangente del ángulo que forma la recta tangente a la curva (función) con el eje de las abcisas, en ese punto.

La derivada de una función mide el coeficiente de variación de dicha función. Es decir, provee una formulación matemática de la noción del coeficiente de cambio. El coeficiente de cambio indica lo rápido que crece (o decrece) una función en un punto (razón de cambio promedio) respecto del eje x de un plano

cartesiano de dos dimensiones. Por ejemplo si tomamos la velocidad de algo, su coeficiente es la aceleración, la cual mide cuánto cambia la velocidad en un tiempo dado.

La derivada es un concepto que tiene muchas aplicaciones. Se aplica en aquellos casos donde es necesario medir la rapidez con que se produce el cambio de una magnitud o situación.

Definición de derivada como límite

En terminología algo anticuada, diferenciación manifiesta el coeficiente en que una cantidad "y" cambia a consecuencia de un cambio en otra cantidad "x" con la que tiene una relación funcional. Usando el símbolo " Δ " para referirse a tal cambio, se define tal coeficiente como el límite del cociente

$$\frac{\Delta y}{\Delta x}$$

Cuando Δx tiende (o se aproxima) a cero.

En la notación de Leibniz, se escribe la derivada de y con respecto a x como sigue:

$$\frac{dy}{dx}$$

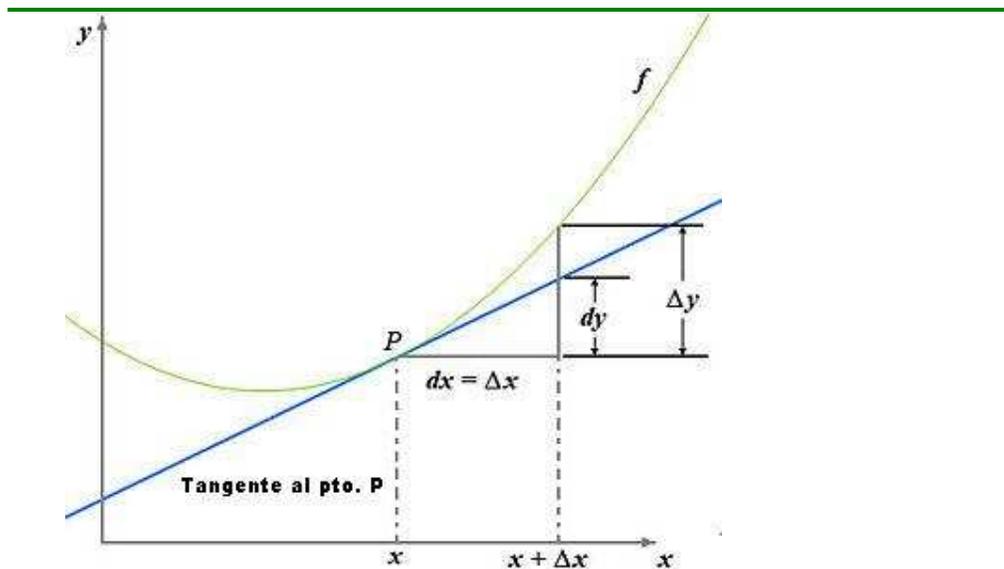


FIG 1: NOTACIÓN DE LEIBNIZ

Tomado de <http://es.wikipedia.org/wiki/Derivada>

Notación Derivadas Parciales

En matemáticas una ecuación en derivadas parciales (a veces abreviado como EDP) es una relación entre una función u de varias variables independientes x , y , z , t ,... y las derivadas parciales de u respecto de esas variables. Las ecuaciones en derivadas parciales se emplean en la formulación matemática de procesos de la física y otras ciencias que suelen estar distribuidos en el espacio y el tiempo. Problemas típicos son la propagación del sonido o del calor, la electrostática, la electrodinámica, la dinámica de fluidos, la elasticidad, la mecánica cuántica y muchos otros.

Notación de las derivadas parciales

Para el siguiente ejemplo, f será una función de x e y .

Derivadas parciales de primer orden:

$$\frac{\partial f}{\partial x} = f'_x = \partial_x f$$

Derivadas parciales (dobles) de segundo orden:

$$\frac{\partial^2 f}{\partial x^2} = f''_{xx} = \partial_{xx} f, \quad \frac{\partial^2 f}{\partial y^2} = f''_{yy} = \partial_{yy} f$$

Derivadas cruzadas de segundo orden:

$$\frac{\partial^2 f}{\partial x \partial y} = f''_{xy} = \partial_{xy} f, \quad \frac{\partial^2 f}{\partial y \partial x} = f''_{yx} = \partial_{yx} f$$

2.2.6 Discriminante o Matriz Hessiana

El hessiano, conocido también como discriminante o matriz hessiana, fue introducido en el año de 1844 por Hesse, matemático alemán quien nació en 1811 y murió en 1874. Esto sucedió luego de que Carl Gustav Jacob Jacobi (1804-1851) introdujera “los jacobianos”. Lo que hizo Jacobi con esto fue expresar los cambios de variable de las integrales múltiples en términos de estos.

MATRIZ HESSIANA DE DOS VARIABLES

Si tenemos un ejercicio con dos variables, obtendremos una matriz hessiana 2 x 2. Si el ejercicio fuera de tres variables, la matriz hessiana será 3 x 3, y así sucesivamente. Para el caso de dos variables, la matriz hessiana 2 x 2 se genera de la siguiente manera:

$$\begin{vmatrix} f_{xx} & f_{xy} \\ f_{xy} & f_{yy} \end{vmatrix} = \begin{vmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{vmatrix}$$

En este trabajo se estará usando la notación que aparece en el miembro izquierdo de las ecuaciones por considerarlo más sencillo de comprender a primera vista.

MATRIZ HESSIANA DE TRES VARIABLES

Antes de presentar ejemplos, se muestra la matriz resultante cuando se trabaja con ejercicios o problemas de tres variables. La matriz hessiana será de 3 x 3 y queda de esta forma:

$$\begin{vmatrix} f_{xx} & f_{xy} & f_{xz} \\ f_{yx} & f_{yy} & f_{yz} \\ f_{zx} & f_{zy} & f_{zz} \end{vmatrix}$$

2.3 TÉCNICAS DE OPTIMIZACIÓN

En matemáticas se puede definir Programación no lineal (PNL) como el proceso de resolución de un sistema de igualdades y desigualdades sujetas a un conjunto de restricciones sobre un conjunto de variables reales desconocidas, con una función objetivo a maximizar, cuando alguna de las restricciones o la función objetivo no son lineales.

De una manera sencilla podemos enunciar el problema de programación no lineal de una forma muy sencilla:

$$\max_{x \in X} f(x) \quad \text{Maximizar una función objetivo}$$

O

$$\min_{x \in X} f(x) \quad \text{Minimizar una función objetivo (de costo)}$$

Donde

$$\begin{aligned} f &: R^n \rightarrow R \\ X &\subseteq R^n. \end{aligned}$$

Algunos problemas de optimización matemática se componen de variables enteras y/o binarias lo que los convierte en problemas de tipo enteros y/o combinatoriales respectivamente, por esta razón, su solución, a través de computadores, emplea tiempos grandes cuando se usan técnicas exactas como Branch and Bound, Benders, Balas, entre otras.

Procurando mejorar la búsqueda de soluciones a este tipo de problemas se han empleado, en los últimos tiempos, técnicas heurísticas y técnicas inteligentes como Colonia de Hormigas, Recocido Simulado, Búsqueda Tabú, Algoritmos Genéticos, entre otras. En general, están basados en la aleatoriedad y en algunos criterios obtenidos de la experiencia del diseñador para encontrar una buena solución. Estas técnicas usan la función objetivo solo para evaluar la calidad de las soluciones, en muchas ocasiones ignorando la información matemática (gradiente) del problema que se encuentra en esta función.

La Programación No Lineal (PNL) provee una serie de herramientas que manipulan en forma estricta los espacios de búsqueda de solución de los problemas, aprovechan información matemática del problema para dirigirse en cada paso hacia un punto de buena calidad, mejorando de esta manera la llegada a la solución.

Además, PNL permite el modelamiento de restricciones no lineales, una característica muy útil para la formulación dada en el presente trabajo a los problemas que involucran variables enteras.

Estas características mencionadas se deben a que en problemas de PNL, el cumplimiento de las condiciones de Karush-Kuhn-Tucker (condiciones de primer orden) y algunas condiciones de segundo orden son requeridas para evaluar la factibilidad y la optimalidad de los puntos que se van encontrando.

Técnicas Clásicas de Optimización

2.3.1 Búsqueda de Fibonacci:

Este método determina el mínimo valor de una función f sobre un intervalo cerrado $[c_1, c_2]$. Esta función puede estar definida en un dominio más amplio, pero el método requiere que dicho intervalo de búsqueda sea definido. Se asume que f es unimodal.

El mínimo es determinado (al menos aproximadamente) mediante la evaluación en un cierto número de puntos. Se pretende definir una estrategia de búsqueda que seleccione la observación siguiente basada en los valores funcionales de las observaciones anteriores

Esto se define según el siguiente problema:

Encontrar como seleccionar sucesivamente N observaciones, sin contar con un conocimiento explícito de la función, de forma tal que podamos encontrar la más pequeña región de incertidumbre posible en donde se encuentre el mínimo.

Esta región de incertidumbre es determinada en cualquier caso por: las observaciones (sus valores funcionales) y la suposición de que f es unimodal.

Procedimiento para la reducción de la sección de incertidumbre:

1. Especificar N

2. Calcular $\left(\frac{F_{N-1}}{F_N} \right)$

3. Colocar simétricamente desde los extremos del intervalo inicial a distancia

$\left(\frac{F_{N-k+1}}{F_N}\right)d_1$, dos observaciones.

4. De acuerdo a donde se encuentre la muestra con menor valor funcional se

determina la región de incertidumbre, $d_2 = \left(\frac{F_{N-1}}{F_N}\right)d_1$

5. La tercera muestra es colocada simétricamente dentro de este nuevo intervalo con respecto a la observación ya incluida en el intervalo, de forma tal que la amplitud de la región de incertidumbre sea:

$$d_k = \left(\frac{F_{N-k+1}}{F_N}\right)d_1$$

2.3.2 Búsqueda de la Sección Dorada

Esta es una técnica simple que ubica al óptimo por reducciones repetitivas de un rango de búsqueda. Parte de la evaluación de la función en 2 puntos diferentes a una distancia razonable entre ellos, para luego evaluar un punto situado dentro del rango limitado por aquellos números y reemplazar el extremo más lejano al mínimo (en un problema de maximización) por este punto; tal punto es seleccionado de manera que esté en una proporción de $\frac{\sqrt{5}-1}{2}$ respecto al intervalo de mayor longitud, medido desde el punto medio del intervalo.

El procedimiento es iterativo hasta hallar una aproximación a la solución que sea adecuada. Las ventajas de este método de búsqueda son su eficiencia y el hecho de que no emplea información de la derivada de la función, aunque su uso está confinado a funciones convexas.

Pertenece a los métodos de búsqueda lineal basados en intervalos, además es una versión mejorada de la búsqueda de Fibonacci.

En la búsqueda de la Sección Dorada se usan tres valores de la función para detectar el valor extremo, se toma un cuarto número, y se determina donde ocurre el mínimo, en los primeros tres o los últimos tres valores.

Se minimiza la evaluación de la función objetivo al reemplazar los valores anteriores con los nuevos.

2.3.3 Método Gradiente Descendente:

Es un método de descenso en el que se elige $dk = -\nabla f(xk)$

Parece interesante por su simplicidad y por el hecho de que si se elige como producto escalar en \mathbb{R}^n el producto euclídeo, la dirección de mayor descenso en x es $-\nabla f(xk)$.

- Si f de clase C^2 y el método de gradiente con paso óptimo (conocido como método de mayor descenso) converge hacia un punto \bar{x} que verifica las condiciones suficientes de optimalidad de segundo orden, entonces el método tiene una tasa de convergencia lineal.
- Si el condicionamiento de la matriz hessiana en el óptimo es muy grande la convergencia es muy lenta.
- La utilidad del algoritmo de gradiente es realmente servir de base a otros métodos más eficaces.
- Señalemos también que el método de gradiente con paso constante converge, para un paso suficientemente pequeño.

2.3.4 Métodos de gradiente conjugado

Gradiente conjugado para un criterio cuadrático

Se considera el problema

$$(\mathcal{PC}) \left\{ \min_{x \in \mathbb{R}^n} f(x) = \frac{1}{2} x^t A x - b^t x \right.$$

Donde A es una matriz $n \times n$, simétrica, definida positiva y b es un vector de \mathbb{R}^n .

Suponemos conocidos n vectores d_1, \dots, d_n , linealmente independientes y

$$d_k^t A d_j = 0, \quad \text{si } k \neq j$$

Expresado en la base de los $\{d_j\}$ el problema se transforma en

$$\left\{ \min_{\tilde{x} \in \mathbb{R}^n} \sum_{k=1}^n \frac{1}{2} d_k^t A d_k \tilde{x}_k^2 - \tilde{b}^t \tilde{x} \right.$$

Donde \tilde{x}_k es la k-ésima componente de x sobre la base de los $\{d_j\}$ y donde \tilde{b} es un vector de R^n tal que $\tilde{b}_k = b_i d_k$.

Entonces el problema (PC) se descompone en n problemas independientes triviales.

Señalemos que el cálculo de las direcciones conjugadas no necesita el conocimiento explícito de A . Basta poder calcular el gradiente en un punto.

El caso particular del algoritmo anterior cuando $v_k = -\nabla f(x_k)$ se llama método de gradiente conjugado. Si $\nabla f(x_k) = 0$ el algoritmo se para. Si $\nabla f(x_k) = 0$ se puede señalar que se ha minimizado f en el hiperplano generado por las $k-1$ direcciones d_i , que es idéntico al hiperplano generado por las $k-1$ direcciones $\nabla f(x_i)$.

La convergencia del método de gradiente conjugado es mucho mejor que la del método de gradiente con paso óptimo.

2.3.5 Ajuste Cuadrático (Método DSC, Davies, Swann y Campey)

El método DSC es un método de búsqueda lineal por ajuste de curvas (interpolación polinómica), es recomendado para determinar la región donde se encuentra el mínimo en funciones de una sola variable.

En la búsqueda unidimensional DSC, se toman pasos cuya dimensión se va incrementando sucesivamente hasta que el mínimo es sobrepasado y luego se realiza una interpolación cuadrática. En la figura se muestra el procedimiento del método DSC.

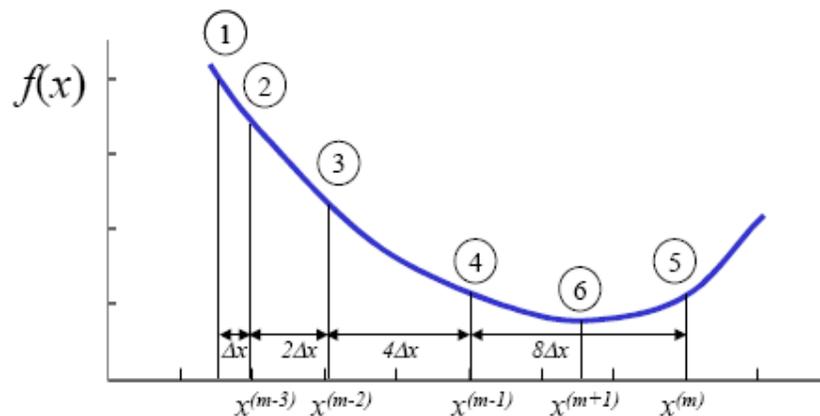


FIG.2 AJUSTE CUADRÁTICO

2.3.6 Ajuste Cúbico

Dados x_{k-1} y x_k junto a $f(x_{k-1})$, $f'(x_{k-1})$, $f(x_k)$, y $f'(x_k)$ es posible ajustar una ecuación cúbica en los puntos.

El punto x_{k+1} (mínimo) puede ser determinado como el punto mínimo relativo de esta ecuación cúbica.

$$x_{k+1} = x_k - (x_k - x_{k-1}) \left[\frac{f'(x_k) + u_2 - u_1}{f'(x_k) - f'(x_{k-1}) + 2u_2} \right]$$

Donde,

$$u_1 = f'(x_{k-1}) - f'(x_k) - 3 \frac{f'(x_{k-1}) + f'(x_k)}{x_{k-1} - x_k}$$

$$u_2 = \left[u_1^2 - f'(x_{k-1})f'(x_k) \right]^{\frac{1}{2}}$$

2.3.7 Método de Newton

Supongamos una función f de una variable a ser minimizada y supongamos que en x_k es posible evaluar $f(x_k)$, $f'(x_k)$ y $f''(x_k)$. Entonces es posible construir una función cuadrática a partir del desarrollo de Taylor:

$$q(x) = f(x_k) + f'(x_k) \cdot (x - x_k) + \frac{1}{2} f''(x_k)(x - x_k)^2$$

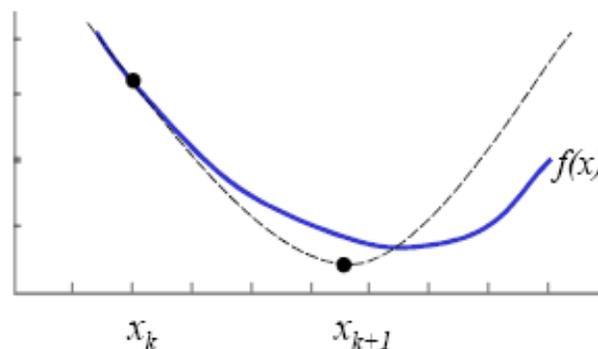


FIG.3 MÉTODO DE NEWTON

Se puede estimar x_{k+1} determinando el punto donde la derivada de q se hace cero.

$$q'(x_{k+1}) = f'(x_k) + f''(x_k)(x_{k+1} - x_k) = 0$$

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)} \quad \text{Nótese que no depende de } f(x) \text{ y } \alpha = \frac{1}{f''(x_k)}$$

El método puede ser visto como la resolución iterativa de ecuaciones de la forma $g(x) = 0$, donde, cuando es aplicada a minimización, hacemos $g(x) \equiv f'(x_k)$

$$x_{k+1} = x_k - \frac{g(x_k)}{g'(x_k)}$$

Implementación

Para la implementación de este método es necesario calcular la primera y segunda derivada de la función como derivadas direccionales, obteniendo un valor escalar, de la siguiente manera,

$$f'(x_k) = \nabla f(x_k) \cdot \vec{d}$$

$$f''(x_k) = \vec{d}^T \cdot H(x_k) \cdot \vec{d}$$

Donde d es el vector unitario de la dirección de descenso.

2.3.8 Método Quasi-Newton

Cuando no es posible evaluar analíticamente las primeras y segundas derivadas, se pueden emplear métodos de diferencias finitas para calcularlas:

$$f'(x) = \frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x}$$

$$f''(x) = \frac{f(x + \Delta x) - 2f(x) + f(x - \Delta x)}{\Delta x^2}$$

2.4 HEURÍSTICA

Se denomina heurística a la capacidad de un sistema para realizar de forma inmediata innovaciones positivas para sus fines. La capacidad heurística es un rasgo característico de los humanos, desde cuyo punto de vista puede describirse como el arte y la ciencia del descubrimiento y de la invención o de resolver problemas mediante la creatividad y el pensamiento lateral o pensamiento divergente.

Como disciplina científica, la heurística es aplicable a cualquier ciencia e incluye la elaboración de medios auxiliares, principios, reglas, estrategias y programas que faciliten la búsqueda de vías de solución a problemas; o sea, para resolver tareas de cualquier tipo para las que no se cuente con un procedimiento algorítmico de solución. Según Horst Müller: Los Procedimientos Heurísticos son formas de trabajo y de pensamiento que apoyan la realización consciente de actividades mentales exigentes. Los Procedimientos Heurísticos como Método científico pueden dividirse en principios, reglas y estrategias.

- Principios Heurísticos: constituyen sugerencias para encontrar (directamente) la idea de solución; posibilita determinar, por tanto, a la vez, los medios y la vía de solución. Dentro de estos principios se destacan la analogía y la reducción.
- Reglas Heurísticas: actúan como impulsos generales dentro del proceso de búsqueda y ayudan a encontrar, especialmente, los medios para resolver los problemas. Las Reglas Heurísticas que más se emplean son:
 - Separar lo dado de lo buscado.
 - Confeccionar figuras de análisis: esquemas, tablas, mapas, etc.
 - Representar magnitudes dadas y buscadas con variables.
 - Determinar si se tienen fórmulas adecuadas.
 - Utilizar números (estructuras más simples) en lugar de datos.
 - Reformular el problema.

2.4.1 Metaheurística

Una Metaheurística es un método heurístico para resolver un tipo de problema computacional general, usando los parámetros dados por el usuario sobre unos procedimientos genéricos y abstractos de una manera que se espera eficiente. Normalmente, estos procedimientos son heurísticos. El nombre combina el prefijo griego "meta" ("más allá", aquí con el sentido de "nivel superior") y "heurístico" (de εὕρισκειν, *heuriskein*, "encontrar").

Las metaheurísticas generalmente se aplican a problemas que no tienen un algoritmo o heurística específica que dé una solución satisfactoria; o bien cuando no es posible implementar ese método óptimo. La mayoría de las metaheurísticas tienen como objetivo los problemas de optimización combinatoria, pero por supuesto, se pueden aplicar a cualquier problema que se pueda reformular en términos heurísticos, por ejemplo en resolución de ecuaciones booleanas.

El objetivo de la optimización combinatoria es encontrar un objeto matemático finito (por ejemplo, un vector de bits o permutación) que maximice (o minimice, dependiendo del problema) una función especificada por el usuario de la Metaheurística. A estos objetos se les suele llamar *estados*, y al conjunto de todos los estados candidatos se le llama *espacio de búsqueda*. La naturaleza de los estados y del espacio de búsqueda son usualmente específicos del problema.

La función a optimizar se le llama *función objetivo*, y se da al usuario como un procedimiento caja-negra que evalúa el estado actual o la función. Dependiendo de la metaheurística, el usuario puede tener que dar otras funciones caja-negra que produzcan un nuevo estado, generen variantes del estado actual, elijan un estado entre varios, aporten valores máximos o mínimos para la función objetivo en un conjunto de estados, y en ese estilo.

Algunas metaheurísticas mantienen en cada instante de ejecución un único *estado actual*, y lo cambian en cada iteración por uno nuevo. Este paso básico se conoce como *transición de estado*, *movimiento* o *actualización del estado*. El movimiento es *colina arriba* o *colina abajo* dependiendo de si los valores que da la función objetivo se incrementa o se decrementa. El nuevo estado puede estar construido desde la nada por un *generador de estados* dado por el usuario. Alternativamente, el nuevo estado puede derivar del estado actual por un *mutador* proporcionado por el usuario; en este caso, el nuevo estado se conoce como *vecino* del estado actual. Generadores y mutadores son habitualmente procedimientos probabilísticos. El conjunto de todos los nuevos estados dados por el mutador es el *vecindario* del estado actual.

Metaheurísticas más sofisticadas mantienen, en vez de un único estado actual, un *conjunto de varios estados candidato*. Así, el paso básico añade o elimina estados de este conjunto. En este caso, los procedimientos dados por el usuario

seleccionan estados para ser descartados, y generan nuevos estados a añadir. El último estado puede ser generado como *combinación* o *cruce* de dos o más estados del conjunto.

Una metaheurística puede guardar información del *óptimo actual*, escogiendo el estado óptimo entre todos los óptimos actuales obtenidos en varias etapas del algoritmo.

Dado que el número de candidatos puede ser muy grande, normalmente, las metaheurísticas están diseñadas de manera que puedan ser interrumpidas por un *tiempo máximo* especificado por el usuario. Si no se interrumpen, algunas *metaheurísticas exactas* examinarán todos los candidatos, y usarán métodos heurísticos sólo para escoger el orden de la enumeración; de hecho, siempre devolverán un óptimo real, si el *tiempo máximo* es lo suficientemente grande. En cambio, otras metaheurísticas dan sólo una garantía probabilística pobre de poder alcanzar el óptimo, de manera que cuando el *tiempo máximo* se aproxima a infinito, la probabilidad de examinar cada candidato tiende a 1.

Algunas aplicaciones:

2.4.2 Colonia de Hormigas (*Ant Colony System*)

La observación de la naturaleza ha sido una de las principales fuentes de inspiración para la propuesta de nuevos paradigmas computacionales. Así nacieron diversas técnicas de Inteligencia Artificial como: los Algoritmos Genéticos (*Genetic Algorithms*), Recocido Simulado (*Simulated Annealing*), Redes Neuronales (*Neural Networks*), y entre estas técnicas, el sistema basado en Colonias de Hormigas (*Ant Colony System*).

Resulta realmente interesante analizar como las hormigas buscan su alimento y logran establecer el camino más corto para luego regresar a su nido. Para esto, al moverse una hormiga, deposita una sustancia química denominada *feromona* como una señal odorífera para que las demás puedan seguirla. Las feromonas son un sistema indirecto de comunicación química entre animales de una misma especie, que transmiten información acerca del estado fisiológico, reproductivo y social, así como la edad, el sexo y el parentesco del animal emisor, las cuales son recibidas en el sistema olfativo del animal receptor, quien interpreta esas señales, jugando un papel importante en la organización y la supervivencia de muchas especies.

Al iniciar la búsqueda de alimento, una hormiga aislada se mueve a ciegas, es decir, sin ninguna señal que pueda guiarla, pero las que le siguen deciden con buena probabilidad seguir el camino con mayor cantidad de feromonas. Considere la Figura en donde se observa como las hormigas establecen el camino más corto. En la figura (a) las hormigas llegan a un punto donde tienen que decidir por uno de los caminos que se les presenta, lo que resuelven de manera aleatoria. En consecuencia, la mitad de las hormigas se dirigen hacia un extremo y la otra mitad hacia el otro extremo, como ilustra la figura (b). Como las hormigas se mueven aproximadamente a una velocidad constante, las que eligieron el camino más corto alcanzarán el otro extremo más rápido que las que tomaron el camino más largo, quedando depositado mayor cantidad de feromona por unidad de longitud, como ilustra la figura (c).

La mayor densidad de feromonas depositadas en el trayecto más corto hace que éste sea más deseable para las siguientes hormigas y por lo tanto la mayoría elige transitar por él. Considerando que la evaporación de la sustancia química hace que los caminos menos transitados sean cada vez menos deseables y la realimentación positiva en el camino con más feromonas, resulta claro que al cabo de un tiempo casi todas las hormigas transitan por el camino más corto.

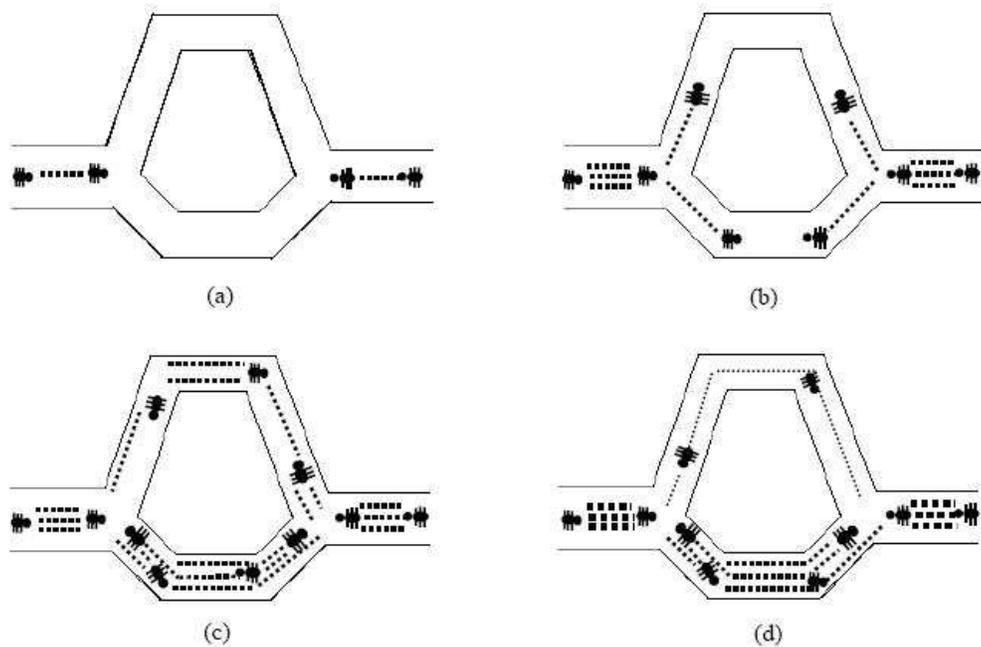


FIG.4 MODELO COLONIA DE HORMIGAS

De las Hormigas Naturales a la Metaheurística de Optimización

Los algoritmos de OCH se inspiran directamente en el comportamiento de las colonias reales de hormigas para solucionar problemas de optimización combinatoria.

Se basan en una colonia de hormigas artificiales, esto es, unos agentes computacionales simples que trabajan de manera cooperativa y se comunican mediante rastros de feromona artificiales.

Los algoritmos de OCH son esencialmente algoritmos constructivos: en cada iteración del algoritmo, cada hormiga construye una solución al problema recorriendo un grafo de construcción. Cada arista del grafo, que representa los posibles pasos que la hormiga puede dar, tiene asociada dos tipos de información que guían el movimiento de la hormiga:

- Información heurística, que mide la preferencia heurística de moverse desde el nodo r hasta el nodo s , o sea, de recorrer la arista a_{rs} . Se denota por h_{rs} . Las hormigas no modifican esta información durante la ejecución del algoritmo.
- Información de los rastros de feromona artificiales, que mide la “deseabilidad aprendida” del movimiento de r a s . Imita a la feromona real que depositan las hormigas naturales. Esta información se modifica durante la ejecución del algoritmo dependiendo de las soluciones encontradas por las hormigas. Se denota por t_{rs} .

En esta sección se presentan los pasos que llevan desde las hormigas reales a la OCH. A partir de ahora debe tenerse en cuenta que los algoritmos OCH presentan una doble perspectiva:

- Por un lado, son una abstracción de algunos patrones de comportamiento naturales relacionados con el comportamiento que permite encontrar el camino más corto.
- Por otro lado, incluyen algunas características que no tienen una contrapartida natural, pero que permiten que se desarrollen algoritmos para obtener buenas soluciones al problema que se pretende resolver (por ejemplo, el uso de información heurística que guíe el movimiento de las hormigas).

La Hormiga Artificial

La hormiga artificial es un agente computacional simple que intenta construir soluciones posibles al problema explotando los rastros de feromona disponibles y la información heurística. Sin embargo, en algunos problemas, puede también construir soluciones no válidas que podrán ser penalizadas dependiendo de lo inadecuado de la solución.

- Busca soluciones válidas de costo mínimo para el problema a solucionar.

- Tiene una memoria L que almacena información sobre el camino seguido hasta el momento, esto es, L almacena la secuencia generada. Esta memoria puede usarse para:
 - (i) Construir soluciones válidas
 - (ii) Evaluar la solución generada
 - (iii) Reconstruir el camino que ha seguido la hormiga.

Aplicaciones de la Optimización Basada en Colonias de Hormigas

Los algoritmos de OCH se han aplicado a un gran número de problemas de optimización combinatoria diferentes. Las aplicaciones actuales de la OCH se distribuyen dentro de dos clases fundamentales. La primera clase de problemas está compuesta por los problemas de *optimización combinatoria NP-duros*, para los que las técnicas clásicas ofrecen a menudo un comportamiento pobre. Una característica común a casi todas las aplicaciones exitosas de la OCH es la combinación de las hormigas con algoritmos de búsqueda local que refinan las soluciones ofrecidas por las hormigas. La segunda clase de aplicaciones se compone de *problemas dinámicos de caminos mínimos*, donde la instancia del problema que hay que solucionar cambia durante la ejecución del algoritmo. Estos cambios pueden afectar a la topología del problema, como por ejemplo la disponibilidad de los enlaces, etc. o, si la topología del problema es fija, características como los costes de los arcos pueden variar con el tiempo. En este caso, el algoritmo tiene que adaptarse a la dinámica del problema. Esta última clase incluye aplicaciones de la OCH al enrutamiento en redes de comunicaciones.

[Tomado de ci2s.ugr.es/publications/ficheros/OCHModelosyNuevosEnfoque\(Chapter\).pdf](http://ci2s.ugr.es/publications/ficheros/OCHModelosyNuevosEnfoque(Chapter).pdf)

2.4.3 RECOCIDO SIMULADO

El algoritmo de recocido simulado (Simulated Annealing Algorithm - SAA) pertenece una clase de Algoritmos de búsqueda local (Local Search Algorithms – LSA) comúnmente llamada Algoritmos de Umbral (Threshold Algorithm - TA). Hay dos razones por las cuales los TA resultan interesantes dentro de los LSA:

- Parecen andar bien en una amplia gama de problemas reales (prácticos).
- Algunos TA, como el SAA, tienen características que permiten hacer un análisis de la convergencia.

Analogía Física

El método del recocido se utiliza en la industria para obtener materiales más resistentes, o más cristalinos, en general, para mejorar las cualidades de un material.

El proceso consiste en “derretir” el material (calentarlo a muy alta temperatura). En esa situación, los átomos adquieren una distribución “azarosa” dentro de la estructura del material y la energía del sistema es máxima. Luego se hace descender la temperatura muy lentamente por etapas, dejando que en cada una de esas etapas los átomos queden en equilibrio (es decir, que los átomos alcancen una configuración óptima para esa temperatura). Al final del proceso, los átomos forman una estructura cristalina altamente regular, el material alcanza así una máxima resistencia y la energía del sistema es mínima.

Experimentalmente se comprueba que si la temperatura se hace descender bruscamente o no se espera suficiente tiempo en cada etapa, al final la estructura del material no es la óptima.

El método del Recocido Simulado

Sea S el conjunto de soluciones posibles del sistema (a las que identificamos con los diferentes “estados del sistema”) y tenemos dada una función costo sobre los elementos de S (a la que identificamos con la “energía del sistema”). Se quiere encontrar un elemento en S que minimice la función costo (análogamente, se trata de encontrar un estado en el cual la energía del sistema sea mínima), asumimos que los estados del sistema tienen la función de distribución de probabilidad de Boltzman. La probabilidad de que el sistema se encuentre en el estado j es:

$$P(j) = (1/Zt) \exp [- c(j) / t]$$

Donde $Zt = \sum \exp [- c(i) / t]$ (suma sobre todos los elementos i de S)
 t es la temperatura del sistema y $c(i)$ es el costo de la solución i .

Sea S^* el subconjunto de S de las soluciones que minimizan c (globalmente, es decir soluciones óptimas del problema). Para t suficientemente chico:

$$\exp[-c(j^*)/t] \gg \exp[-c(j)/t]$$

para todo $j \neq j^*$

Entonces:

$$P(j) = \frac{\exp[-c(j)/t]}{\sum_{j^* \in S^*} \exp[-c(j^*)/t]} = \begin{cases} 0 & \text{si } j \neq j^* \\ 1/S^* & \text{si } j = j^* \end{cases}$$

(Esto se obtiene de tomar t tendiendo a 0)

Por lo tanto: $\sum P(j^*) = 1$ (suma sobre todos los j^* en S^*).

Algoritmo de Recocido Simulado

El algoritmo se divide en etapas. A cada etapa le corresponde una temperatura menor que la que tenía la etapa anterior (a esto hace referencia la monotonía: después de cada etapa la temperatura baja, se enfría el sistema). Por lo tanto hace falta un criterio de cambio de la temperatura ("cuanto tiempo" se espera en cada etapa para dar lugar a que el sistema alcance su "equilibrio térmico").

En general este método hace uso de algunos conceptos de mecánica estadística, los ejemplos mas claros en los cuales se usa este tipo de métodos es en la metalurgia donde se lleva un sólido para llevarlo de estado sólido a un estado de equilibrio térmico.

3. HERRAMIENTAS UTILIZADAS EN LA APLICACIÓN

3.1 PHP (*Hypertext Preprocessor*)

Es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor.

El código PHP se incluye entre etiquetas especiales de comienzo y final que permiten entrar y salir del modo PHP. Lo que distingue a PHP de la tecnología Javascript, la cual se ejecuta en la máquina cliente, es que el código PHP es ejecutado en el servidor. PHP puede hacer cualquier cosa que se pueda hacer con un script CGI, como procesar la información de formularios, generar páginas con contenidos dinámicos, o mandar y recibir cookies.

PHP es un lenguaje de programación de estilo clásico, con variables, sentencias condicionales, bucles, funciones. No es un lenguaje de marcas como podría ser HTML, XML o WML. El resultado es normalmente una página HTML pero igualmente podría ser una página WML.

PHP es un lenguaje con asequible estructura de programación, la facilidad de llevar a cabo sentencias SQL embebidas, además de permitir la posibilidad de correr en diferentes tipos de servidores, entre ellos Apache.

Quizás la característica más potente y destacable de PHP es su soporte para una gran cantidad de bases de datos. Escribir una interfaz vía web para una base de datos es una tarea simple con PHP.

PHP también soporta el uso de otros servicios que usen protocolos como IMAP, SNMP, NNTP, POP3, HTTP y derivados. También se pueden abrir sockets de red directos (raw sockets) e interactuar con otros protocolos.

3.2 BASE DE DATOS MySQL

El software MySQL proporciona un servidor de base de datos SQL (Structured Query Language) veloz, multi-hilo, multiusuario y robusto. El servidor está proyectado tanto para sistemas críticos en producción soportando intensas cargas de trabajo como para empotrarse en sistemas de desarrollo masivo de software. El software MySQL tiene licencia dual, pudiéndose usar de forma gratuita bajo licencia GNU o bien adquiriendo licencias comerciales de MySQL AB en el caso de no desear estar sujeto a los términos de la licencia GPL. MySQL es una marca registrada de MySQL AB.

Tomado de <http://www.mastermagazine.info/termino/6051.php>

<http://www.virtual.unal.edu.co/cursos/sedes/manizales/4060029/lecciones/cap11-2.html>

3.3 JOOMLA

Joomla! es un sistema de administración de contenidos de código abierto construido con PHP bajo una licencia GPL. Este administrador de contenidos se usa para publicar en Internet e intranets utilizando una base de datos MySQL. En Joomla! se incluyen características como: hacer caché de páginas para mejorar el rendimiento, indexamiento Web, feed RSS, versiones imprimibles de páginas, flash con noticias, blogs, foros, polls (encuestas), calendarios, búsqueda en el sitio Web, e internacionalización del lenguaje. Su nombre es una pronunciación fonética para anglófonos de la palabra swahili jumla que significa "todos juntos" o "como un todo". Se escogió como una reflexión del compromiso del grupo de desarrolladores y la comunidad del proyecto.

La primera versión de Joomla! (Joomla! 1.0.0) fue publicada el 16 de septiembre de 2005. Se trataba de una versión mejorada de Mambo 4.5.2.3 combinada con otras modificaciones de seguridad y anti-bugs. Actualmente los programadores han publicado Joomla! 1.5 Stable bajo un código completamente reescrito y construido bajo PHP 5.

4 METODOLOGÍA DE ANÁLISIS Y DISEÑO

Tomado de la guía para la elaboración y presentación de proyectos de la pontificia universidad Javeriana de Cali.

En el presente documento buscamos responder a la necesidad que tenemos los estudiantes y profesionales de presentar metódicamente los resultados de las investigaciones. Tratando de acoger en nuestro proyecto la revisión de nuevas fuentes de información, tales como normas internacionales sobre documentación de la Organización Internacional para la Normalización (ISO), IEEE Professional Communication Society y la Real Academia Española en cuanto al aspecto lingüístico y ortográfico.

En este proyecto se pretende describir y documentar toda la información relacionada con la programación no lineal teniendo en cuenta que para cada disciplina del conocimiento debe existir un propósito que la rija, lo cual condiciona a un estilo propio para obtener los resultados propuestos para un excelente aprendizaje significativo en el área, como objetivo principal.

Todo esto llevado de la mano del Ingeniero Juan Carlos Reyes el cual con su apoyo, su actitud cooperativa y su valiosa opinión en el proceso de Recolección y consolidación de la información, nos ha hecho comprender mas a fondo el significado de todo lo que conlleva el área de PnL, esperamos que al haber aplicado sus recomendaciones obtengamos resultados coherentes con las necesidades de todas las personas que podrían usar nuestra herramienta.

Teniendo en cuenta las necesidades de nuestro proyecto y haciendo un análisis entre los distintos tipos de metodologías para el ciclo de vida del software decidimos conjuntamente utilizar el modelo de cascada ya que es el más básico de todos los modelos, y su secuencia de fases hace que vayamos teniendo una clasificación de metas bien definidas y una serie de actividades dentro de cada una de ellas, las cuales podemos resolver de una manera eficiente y eficaz, además este modelo nos permite que halla una retroalimentación en cada fase antes de continuar a la siguiente.

4.1 Ciclo de vida del desarrollo software

Todo proyecto de ingeniería tiene unos fines ligados a la obtención de un producto, proceso o servicio que es necesario generar a través de diversas actividades. Algunas de estas actividades pueden agruparse en fases porque globalmente contribuyen a obtener un producto intermedio, necesario para continuar hacia el producto final y facilitar la gestión del proyecto. Al conjunto de las fases empleadas se le denomina “ciclo de vida”.

La definición de un ciclo de vida facilita el control sobre los tiempos en que es necesario aplicar recursos de todo tipo (personal, equipos, suministros, etc.) al proyecto. Si el proyecto incluye subcontratación de partes a otras organizaciones, el control del trabajo subcontratado se facilita en la medida en que esas partes encajen bien en la estructura de las fases. El control de calidad también se ve facilitado si la separación entre fases se hace corresponder con puntos en los que ésta deba verificarse (mediante comprobaciones sobre los productos parciales obtenidos).

De la misma forma, la práctica acumulada en el diseño de modelos de ciclo de vida para situaciones muy diversas permite que nos beneficiemos de la experiencia adquirida utilizando e El ciclo de vida básico de un software consta de los siguientes procedimientos:

- Definición de objetivos: definir el resultado del proyecto y su papel en la estrategia global.
- Análisis de los requisitos y su viabilidad: recopilar, examinar y formular los requisitos del cliente y examinar cualquier restricción que se pueda aplicar.
- Diseño general: requisitos generales de la arquitectura de la aplicación.
- Programación (programación e implementación): es la implementación de un lenguaje de programación para crear las funciones definidas durante la etapa de diseño.
- Prueba de unidad: prueba individual de cada subconjunto de la aplicación para garantizar que se implementaron de acuerdo con las especificaciones.

- Integración: para garantizar que los diferentes módulos se integren con la aplicación. Éste es el propósito de la prueba de integración que está cuidadosamente documentada.
- Prueba beta (o validación), para garantizar que el software cumple con las especificaciones originales.
- Documentación: sirve para documentar información necesaria para los usuarios del software y para desarrollos futuros.
- Implementación
- Mantenimiento: para todos los procedimientos correctivos (mantenimiento correctivo) y las actualizaciones secundarias del software (mantenimiento continuo).

4.1.1 Proceso Unificado

El Proceso Unificado es un proceso de software genérico que puede ser utilizado para una gran cantidad de tipos de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de competencia y diferentes tamaños de proyectos.

Provee un enfoque disciplinado en la asignación de tareas y responsabilidades dentro de una organización de desarrollo. Su meta es asegurar la producción de software de muy alta calidad que satisfaga las necesidades de los usuarios finales, dentro de un calendario y presupuesto predecible.

El Proceso Unificado se basa en componentes (component-based), lo que significa que el sistema en construcción está hecho de componentes de software interconectados por medio de interfaces bien definidas (well-defined interfaces).

El Proceso Unificado usa el Lenguaje de Modelado Unificado (UML) en la preparación de todos los planos del sistema. De hecho, UML es una parte integral del Proceso Unificado, fueron desarrollados a la par.

Los aspectos distintivos del Proceso Unificado están capturados en tres conceptos clave: dirigido por casos de uso (use-case driven), centrado en la arquitectura (architecture-centric), iterativo e incremental. Esto es lo que hace único al Proceso Unificado.

Tomado de <http://yaqui.mx/uabc.mx/~molguin/as/RUP.htm>

<http://es.kioskea.net/genie-logiciel/cycle-de-vie.php3>

4.1.2 Método del desarrollo por prototipos:

Los sistemas pueden desarrollarse con métodos y lenguajes de programación convencionales, aunque no tengan todas las características y toques finales de un sistema terminado. Quizás los informes no tengan encabezados, logos, etc., falten controles de entradas y procesamiento. Lo importante es el ensayo, y hallar los requerimientos.

Las razones para desarrollar prototipos de sistemas es que los requerimientos de información no siempre están bien definidos, pueden ser demasiados vagos aún al formular el diseño. En otros casos, es probable que una investigación de sistemas bien llevada, de como resultado un conjunto muy amplio de requerimientos de sistemas, pero construir un sistema que satisfaga a todos ellos quizás necesite del desarrollo de nueva tecnología.

Los prototipos permiten evaluar situaciones extraordinarias donde los encargados de diseñar e implantar sistemas no tienen información ni experiencia, o también donde existen situaciones de riesgo y costos elevados, y aquellas donde el diseño propuesto es novedoso y aún no ha sido probada.

La información obtenida con su uso se aplica en un nuevo diseño que se emplea, otra vez, como prototipo y que revela más información valiosa sobre diseño. El proceso se repite las veces que sea necesario para revelar los requerimientos esenciales del diseño.

4.1.3 Modelo Cascada

El modelo cascada se define como una secuencia de actividades a ser seguidas en orden, donde la principal estrategia es analizar y definir los requerimientos y seguir el proceso de desarrollo del software hacia puntos de revisión bien definidos.

Problemas y Riesgos Con el Modelo Cascada

- Alto riesgo en sistemas nuevos debido a problemas en las especificaciones y en el diseño.
- Bajo riesgo para desarrollos bien comprendidos utilizando tecnología conocida.

Grafica del Modelo Cascada

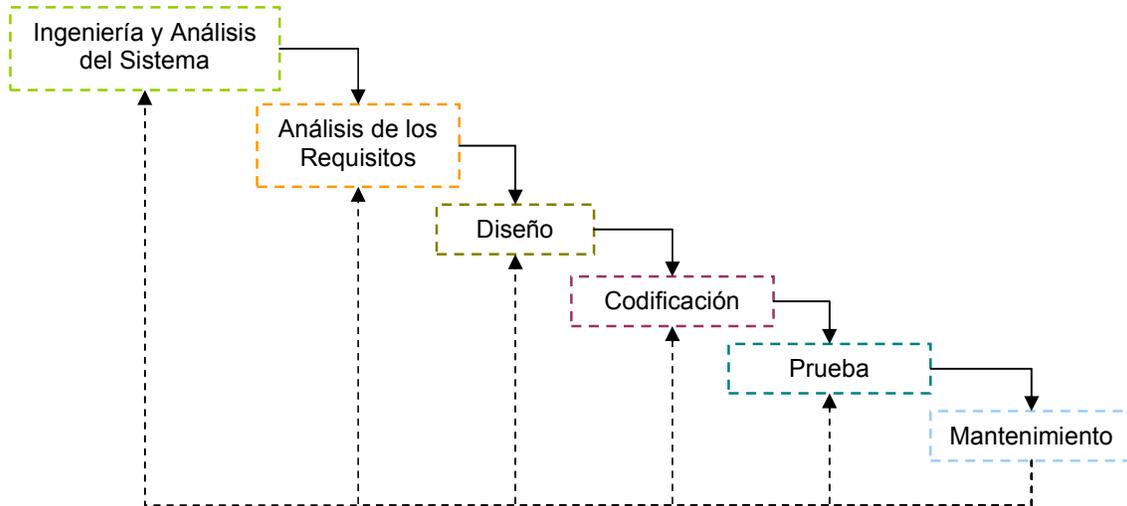


FIG.5 GRAFICA MODELO CASCADA

4.1.3.1 Ingeniería y Análisis del Sistema

Este es un completo y amigable software para la simulación de algoritmos de programación no lineal. Incluye una completa documentación de todos los algoritmos usados en PnL y de algunos métodos metaheurísticos (colonia de hormigas, Búsqueda tabú, recocido simulado), Simulaciones de los algoritmos mas usados y acordados con anterioridad con el director del proyecto, animaciones para la comprensión total de todos los temas (limite, derivada, integral, hessiano, jacobiano) manejados en los algoritmos simulados, etc.

Puede ser ejecutado bajo cualquier sistema operativo ya que este se realizo sobre plataforma Web. La interfase de usuario, los manuales y la ayuda en línea están disponibles en la dirección www.artenacional.com/tesis.

Este software fue implementado principalmente para servir como complemento educacional en el desarrollo de la asignatura programación no lineal, en el cual cualquier persona podrá tener acceso a toda la información incluida en el paquete sin ninguna restricción, esta basado en una interfaz amigable para el usuario, con funciones anteriormente nombradas, útiles para trabajar eficiente y eficazmente.

4.1.3.2 Análisis de los requisitos del software

Es el primer paso del análisis del sistema. En este proceso analizamos e identificamos las necesidades que requiere la asignatura discutiéndolas con el director de la misma, planteando metas globales, necesidades y requerimientos para el desarrollo óptimo de la asignatura.

Se definió la planificación temporal (cronograma de actividades) y presupuestal (Costos que conlleva la realización del proyecto), y otros puntos que puedan ayudar a la identificación y desarrollo del proyecto.

El análisis de requerimientos lo subdividimos en 5 ítems:

- Reconocimiento del problema

La programación no lineal (PNL), además de su parte teórica requiere de un fuerte trabajo práctico. La enseñanza se realiza actualmente, basada en la explicación que realiza el profesor sobre la solución del problema (exposición de los algoritmos para la optimización de las funciones), y ante el eventual hecho de desear la participación del estudiante, las operaciones necesarias para la solución de los problemas son, algunas veces, demasiado extensas, de manera que se pierde sentido práctico en el desarrollo de la clase, además el sin número de algoritmos para encontrar la solución del problema no permite el análisis del mejor algoritmo.

- Evaluación y Síntesis

La solución inicial para este problema, se encuentra en que los estudiantes programen estos algoritmos utilizando cualquier tipo de tecnologías en cuanto a lenguajes de programación, muchas de las cuales no pueden funcionar por diversas razones, en los equipos de la universidad, por ejemplo la utilización de MATLAB es compleja, debido a las escasas licencias que están a disposición de los estudiantes de ingeniería de sistemas. Así mismo, el objetivo del área tampoco se encuentra en el desarrollo de herramientas para resolver los problemas, realmente este no es el sentido práctico hacia el cual se enfoca.

- Especificación

Con este proyecto se desea disponer de un sistema de solución de problemas de programación no lineal usando los diferentes modelos matemáticos conocidos, además hacer que el usuario interactúe con el sistema para así desarrollar sus conocimientos en el área.

- Revisión

En esta parte del análisis se tomarán en cuenta las opiniones del director, para crear o rediseñar cualquier parte del sistema según los requerimientos que sean necesarios.

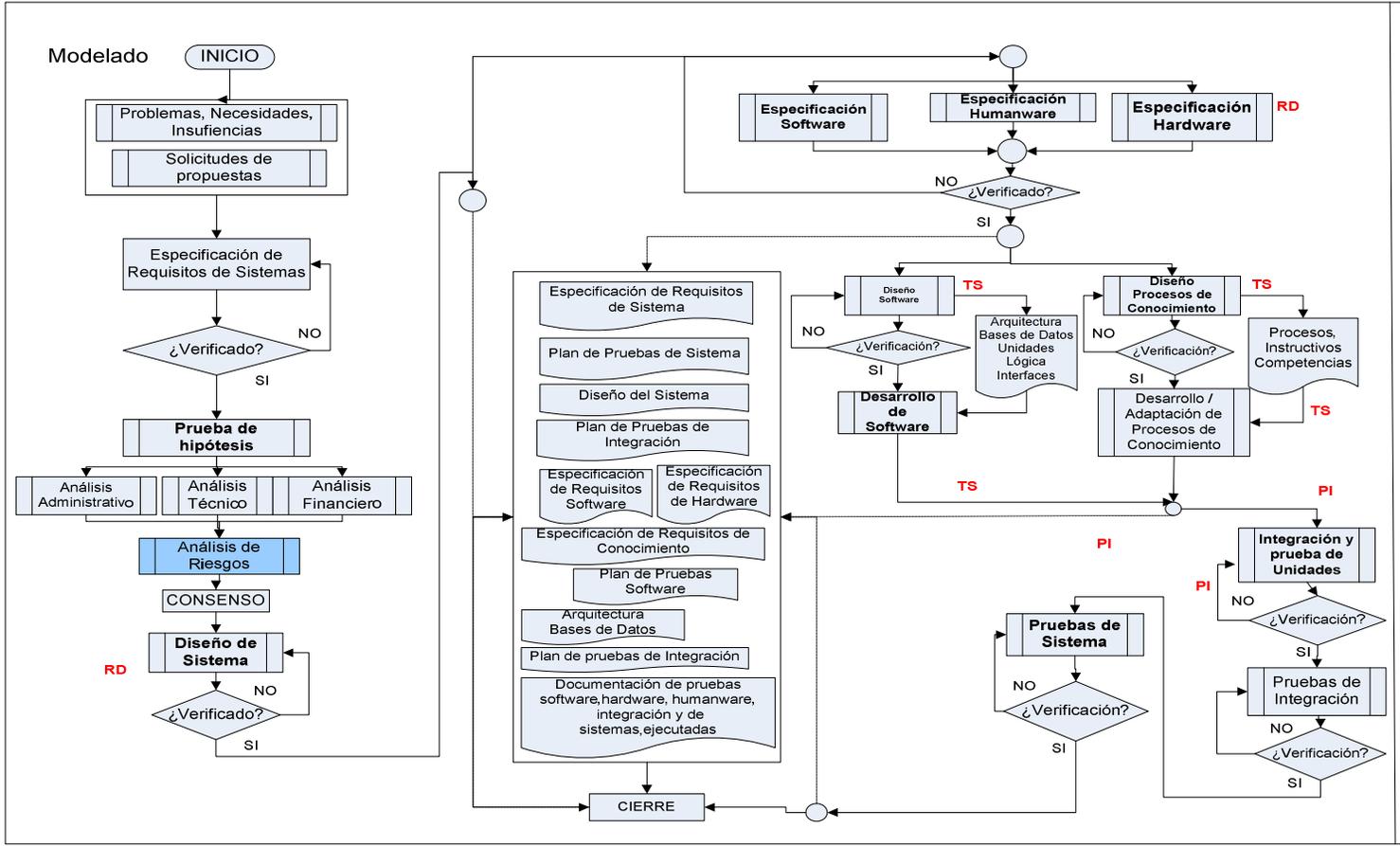


FIG.6 MODELADO DEL SISTEMA

4.1.3.3 Diseño

“El diseño del software se enfoca en cuatro atributos distintos del programa: la estructura de los datos, la arquitectura del software, el detalle procedimental y la caracterización de la interfaz. El proceso de diseño traduce los requisitos en una representación del software con la calidad requerida antes de que comience la codificación”.

El Diseño de Sistemas se define como el proceso de aplicar ciertas técnicas y principios con el propósito de definir un dispositivo, un proceso o un Sistema, con suficientes detalles como para permitir su interpretación y realización física, siguiendo esta definición nuestra etapa de diseño la encerramos en cuatro etapas:

- El diseño de los datos

En este paso después de recopilar toda la información de varias fuentes (Internet, libros) acerca de los algoritmos y las herramientas a utilizar, desarrollamos la estructura de una forma coherente de tal forma que fuese coherente y organizada, para que el sistema cumpliera con las necesidades identificadas en la etapa anterior.

- El Diseño Arquitectónico

En este proceso del diseño identificamos los subsistemas que conforman el sistema (administrador de usuarios, el gestor de artículos, etc.) y sus componentes principales (gestor de contenidos y el gestor de simulaciones, etc.).

- El Diseño de la Interfaz

“Describe como se comunica el Software consigo mismo, con los sistemas que operan junto con el y con los operadores y usuarios que lo emplean”.

Se diseño una interfaz para el usuario de manera que el sistema se presente lo más amigable posible para el usuario, tratando de definir un ambiente sencillo de manejar, donde todos los parámetros definidos estén al alcance según el uso que se le este dando a la herramienta.

Además este es un sistema de administración de contenidos construido con PHP bajo una licencia GPL, utilizando una base de datos MySQL, también se incluyen características como versiones imprimibles de las paginas, calendarios, foros, etc.

- El Diseño de procedimientos

En esta fase transformamos los elementos estructurales de la arquitectura del programa, esta parte del Diseño del Software se baso en darle Calidad al proyecto, puesto que en este punto fue donde se materializo los requerimientos del sistema.

4.1.3.4 Codificación

El diseño debe traducirse en una forma legible para la maquina. El paso de codificación realiza esta tarea. Si el diseño se realiza de una manera detallada la codificación puede realizarse mecánicamente.

4.1.3.5 Prueba

“Una vez que se ha generado el código comienza la prueba del programa. La prueba se centra en la lógica interna del software, y en las funciones externas, realizando pruebas que aseguren que la entrada definida produce los resultados que realmente se requieren”.

Los objetivos de la fase de pruebas del sistema es verificar que el comportamiento externo del sistema software satisface los requisitos establecidos en las primeras fases para los usuarios del mismo.

Este trabajo describe los modelos necesarios para generar de manera sistemática un conjunto de pruebas que permitan verificar la implementación de los requisitos funcionales de un sistema software.

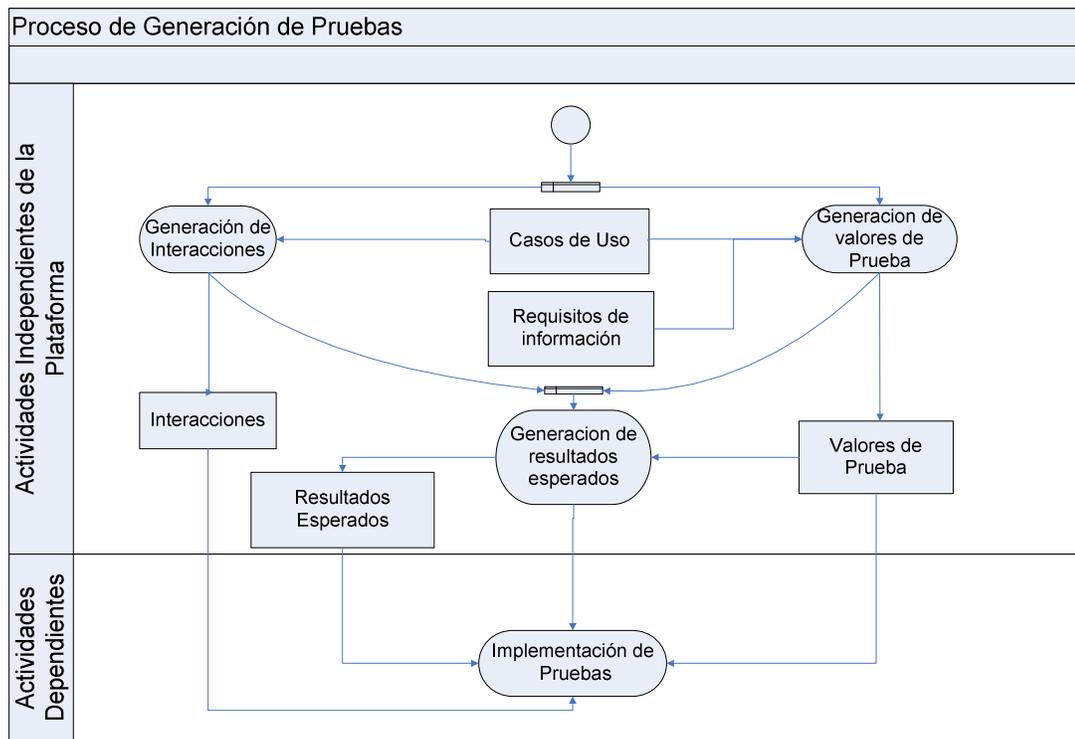


FIG.7 PROCESO DE GENERACIÓN DE PRUEBAS

La ventaja de este método radica en su sencillez debido a que se siguen los pasos intuitivos necesarios a la hora de desarrollar el software.

5 UML (Lenguaje de Modelado Unificado)

5.1 Introducción

5.1.1 Objetivos del documento

- Brindar al equipo de desarrollo las herramientas necesarias para hacer del análisis y el diseño las bases sobre las que se construya el sistema software.
- Realizar énfasis en el correcto diseño de las principales funcionalidades del sistema como parte fundamental para cumplir las expectativas de los usuarios en torno a la aplicación.
- Establecer las directrices que regirán el proceso de desarrollo de la aplicación mediante la implementación de estándares que optimicen las labores del equipo de trabajo

5.1.2 Descripción general del proyecto

Los Módulos para la simulación y explicación de los algoritmos de programación no lineal de la herramienta permitirán llevar a cabo una interacción del usuario con la herramienta, teniendo en cuenta que todos tendrán la posibilidad de simular los diferentes algoritmos con ayuda de los ejemplos incluidos en la herramienta o con nuevos según lo que requiera el usuario.

A través de dicha aplicación los usuarios podrán realizar sus prácticas en la asignatura de programación no lineal de una manera eficiente, sencilla y ágil. Además permitirá la consulta de la información de cada algoritmo y las definiciones que se deben tener en cuenta para cada uno de estos, de forma pertinente de manera que apoye al aprendizaje significativo sobre la asignatura.

A continuación se hace una breve descripción de las principales necesidades:

ID	Nombre	Descripción	Prioridad
1	Describir de manera eficiente la teoría del área de programación no lineal.	Esta descripción se hará para tener una mayor comprensión sobre los temas, además de ser una ayuda al momento de resolver los problemas teóricos.	Alta
2	Diseñar un ambiente agradable para el usuario.	Se diseñó un ambiente en el cual los usuarios tengan la oportunidad de observar los temas tratados en el área PnL, e interactuar con las diferentes ayudas para un óptimo aprendizaje del área.	Alta
3	Realizar la simulación de los algoritmos más utilizados en PnL.	Estas simulaciones se diseñaron de tal forma que el usuario pudiera manejarlas de una manera dinámica, con lo que se desea que todos los conceptos queden claros y entendidos.	Alta
4	Complemento teórico con ejercicios teóricos, pseudo códigos y diagramas de flujo.	Esto con el fin de manejar los resultados de los algoritmos de manera que el usuario tenga la capacidad de sacar sus propias conclusiones y pueda aportar nuevas ideas al tema.	Media
5	Implementación del Sistema.	Se desea que el usuario pueda utilizar el software desde cualquier lugar, para que este no esté ligado a un solo punto de trabajo, y este pueda ingresar a cualquier momento del día, para esto el sistema está implementado sobre plataforma Web.	Alta
6	Modificación de los Datos del Sistema	El administrador del sistema manejará una plantilla que le permite hacer cambios tanto en la interfaz como en los datos de artículos, ejercicios, etc.	Baja
7	Estadísticas	Es una sección en la que simplemente se quiere obtener información sobre la acogida que obtuvo nuestro sistema.	Baja
8	Registros	Existe una sección en la plantilla de trabajo en la que se manejará todo el registro de los usuarios	Media

5.2 MODELO ORGANIZACIONAL

5.2.1 Definición de procesos (Automáticos y Manuales).

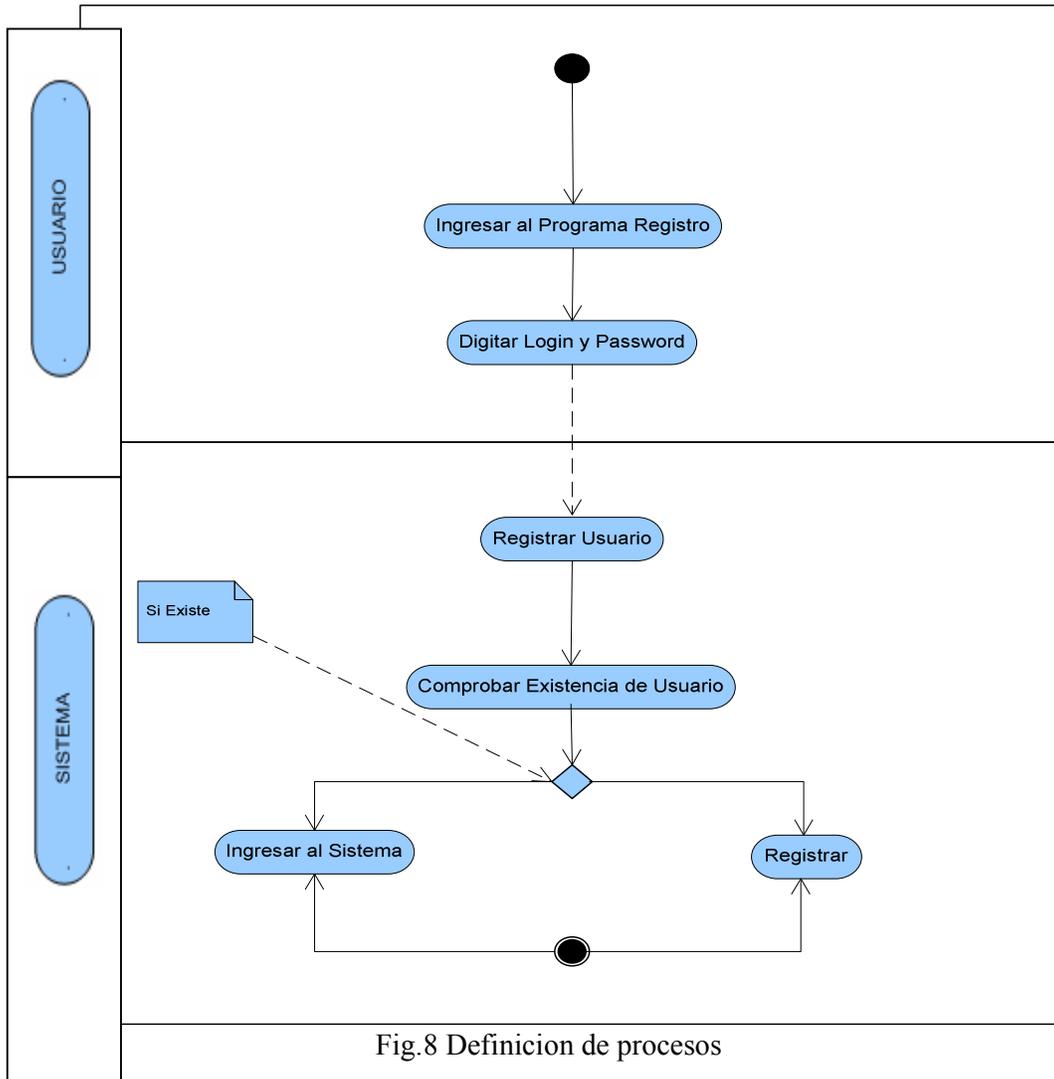


Fig.8 Definición de procesos

Características de los usuarios.

- Estudiantes que cuentan con un grado medio / alto de conocimientos sobre PNL e informática.
- Profesor, quien cuenta con un grado alto de manejo de la asignatura y de herramientas software.

5.3 MODELO FUNCIONAL

5.3.1 Diseño de la interfase de usuario



FIG.9 INTERFAZ DE USUARIO

5.3.2 Diseño del Registro

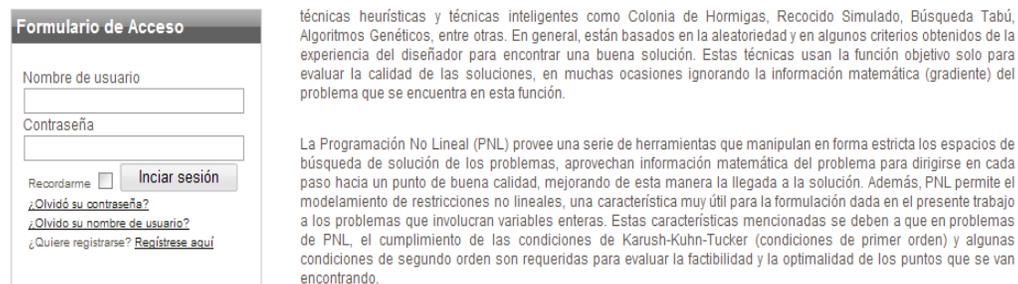


FIG.10 DISEÑO DE REGISTRO

5.3.3 Casos de uso (General)

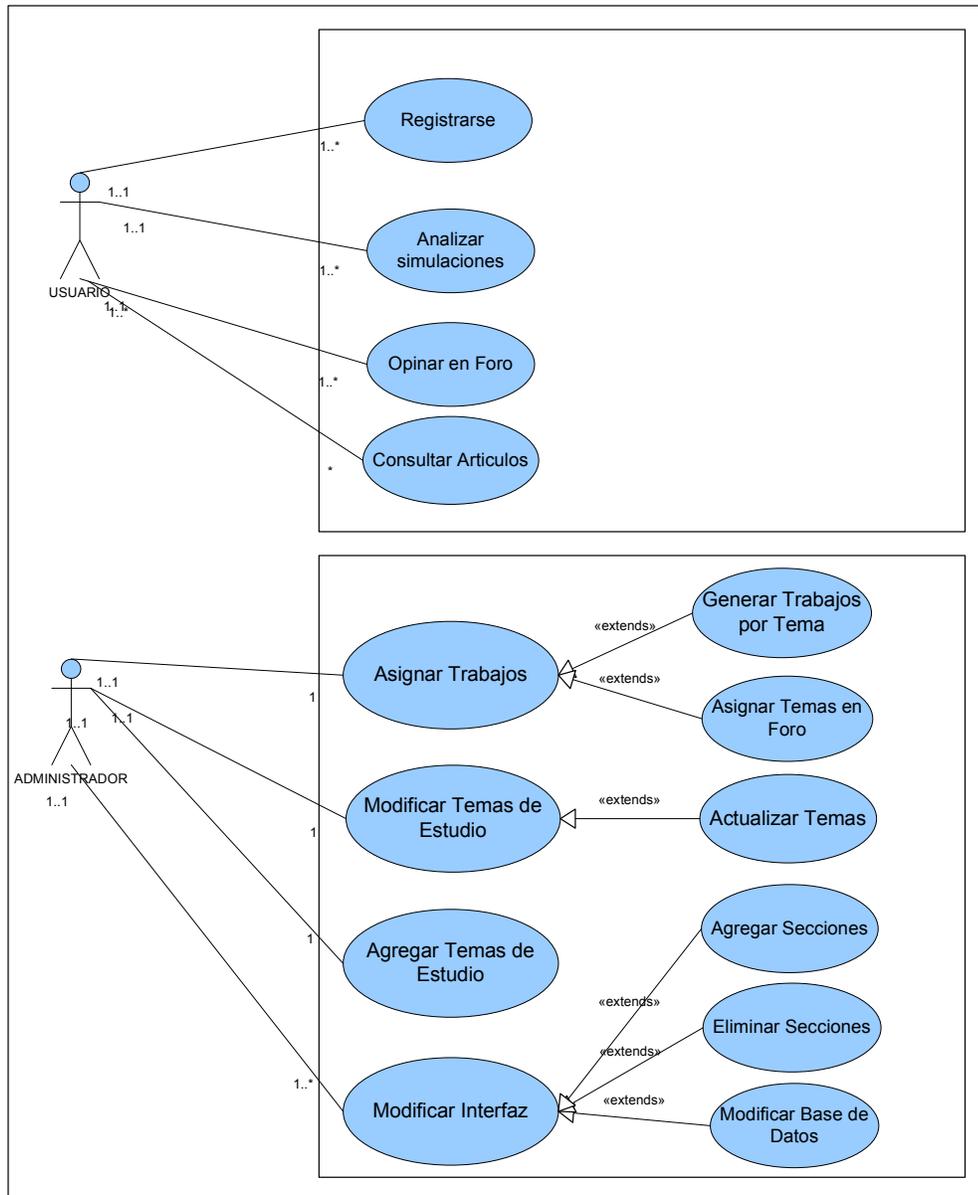


FIG 11 CASO DE USO (GENERAL)

5.3.3.1 Caso de Uso (Registro)

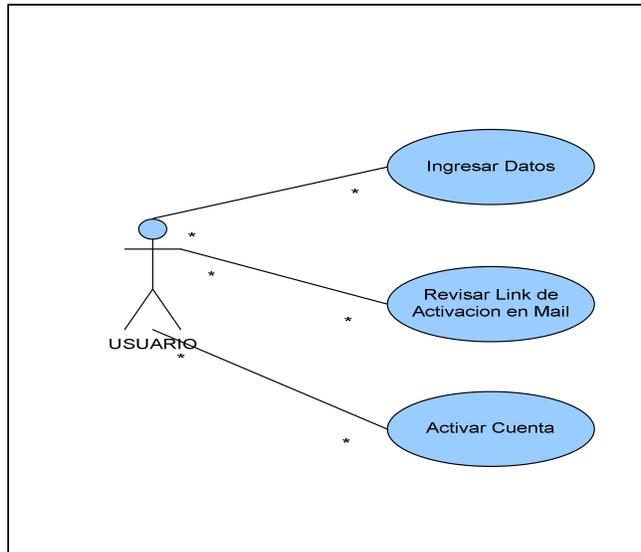


FIG 12 CASO DE USO (REGISTRO)

5.3.3.2 Caso de Uso (Opinar en Foro)

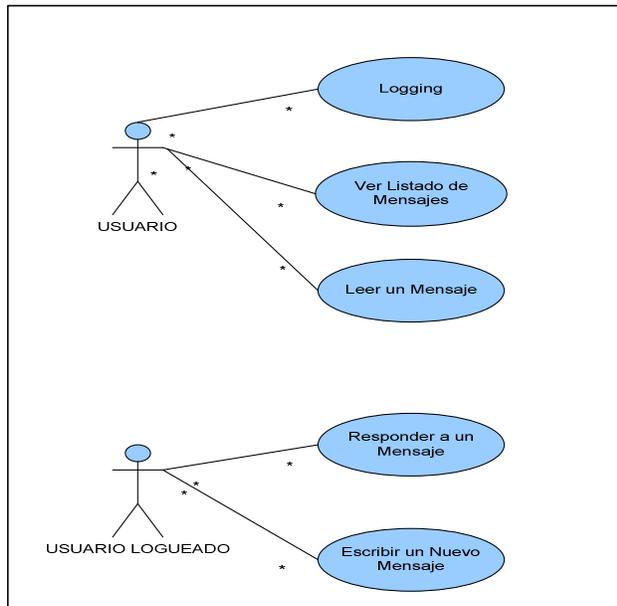


FIG 13 CASO DE USO (OPINAR FORO)

5.3.3.3 Caso de Uso (Usuario)

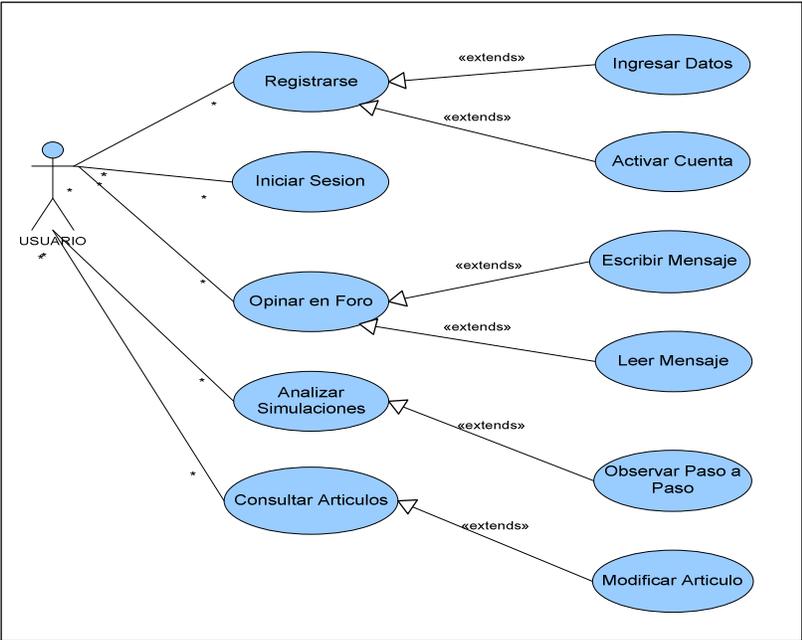


FIG 14 CASO DE USO (USUARIO)

5.3.3.4 Caso de Uso (Administrador)

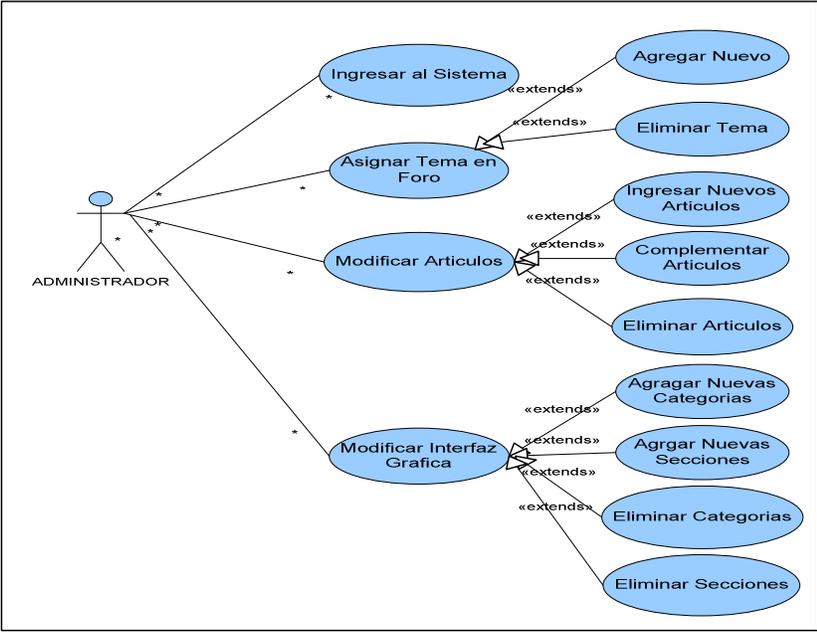


FIG 15 CASO DE USO (ADMINISTRADOR)

5.4 MODELO LÓGICO

5.4.1 Diagrama de Clases

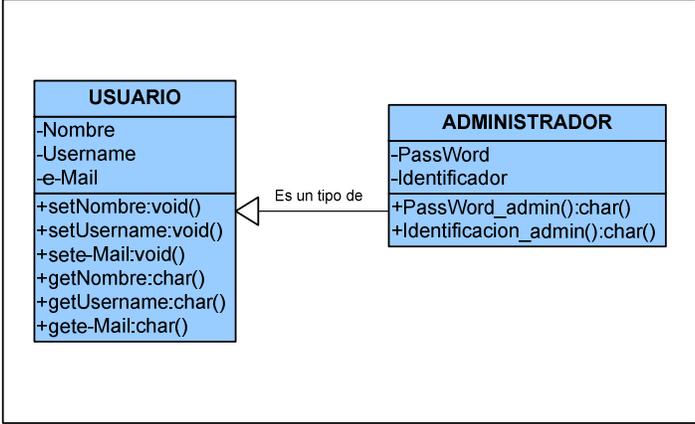


FIG 16 DIAGRAMAS DE CLASE

5.4.2 Diagrama de Estados

5.4.2.1 Diagrama de Estados del Registro

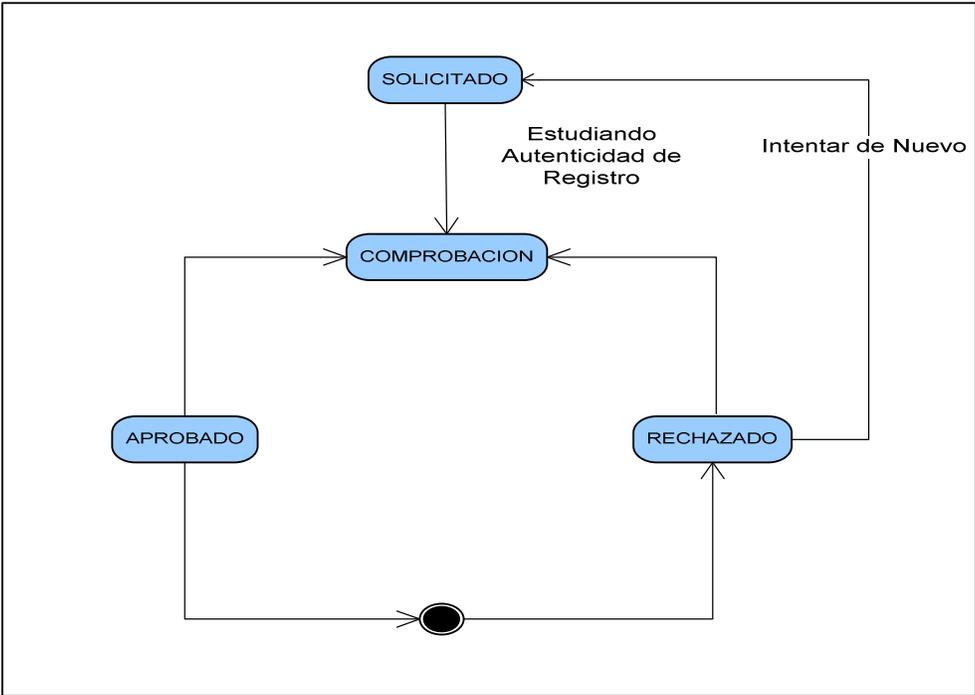


FIG 17 DIAGRAMA DE ESTADO DE REGISTROS

5.4.3 Diagramas de Secuencias

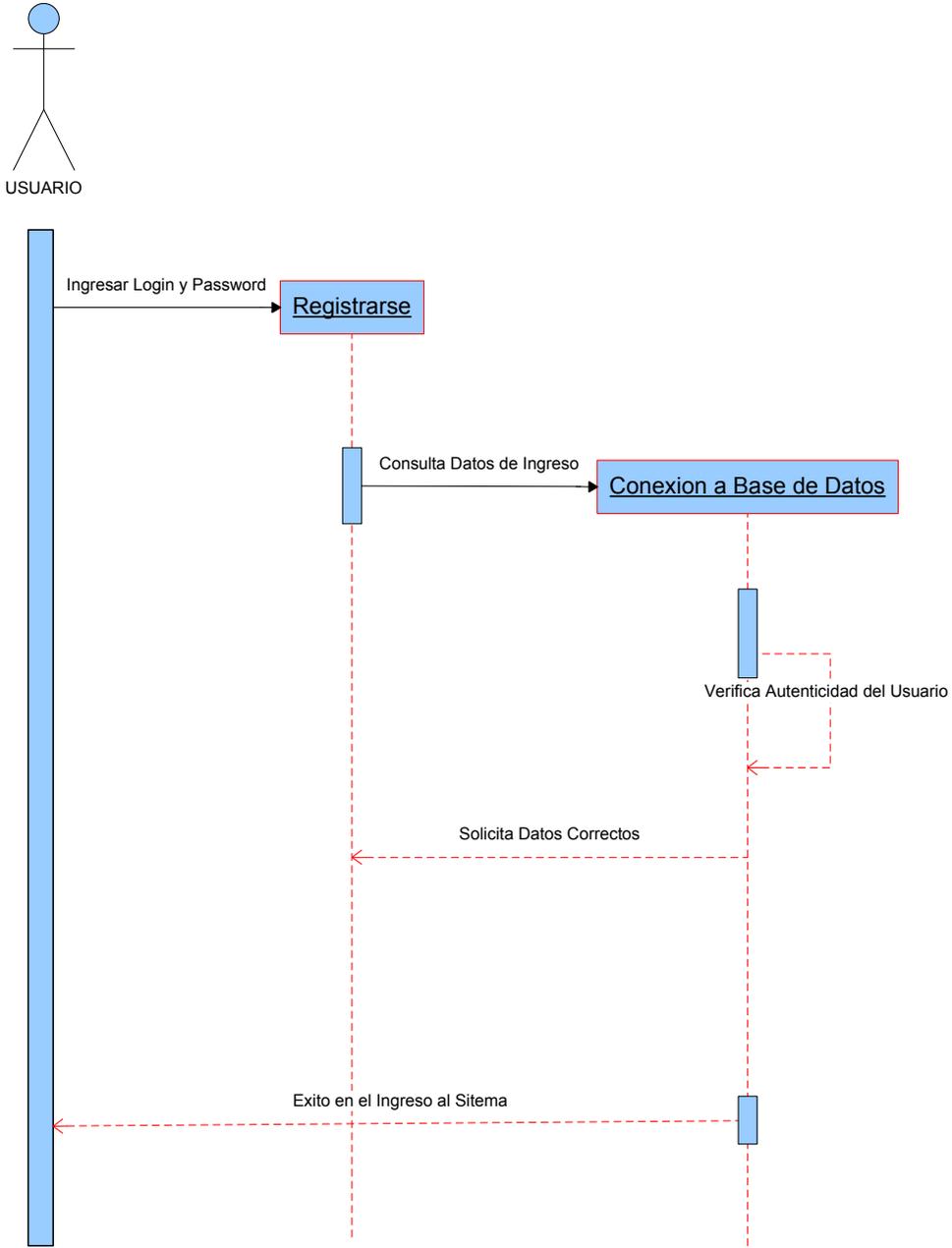


FIG 18 DIAGRAMA DE SECUENCIAS

5.4.4 Modelo de Datos

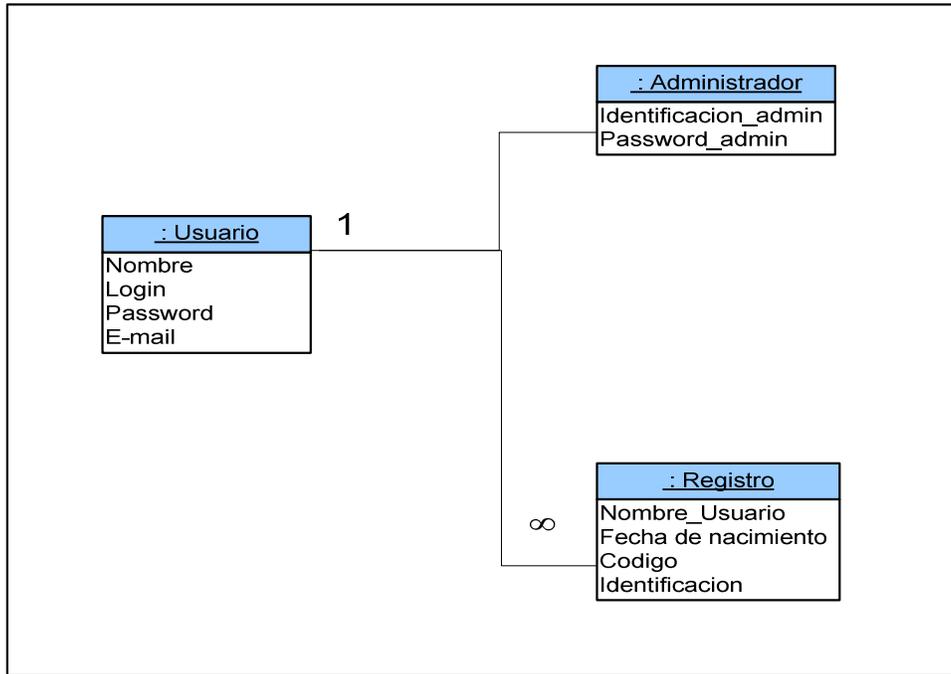


FIG 19 MODELO DE DATOS

5.5 MODELO FÍSICO

5.5.1 Diagrama de Despliegue

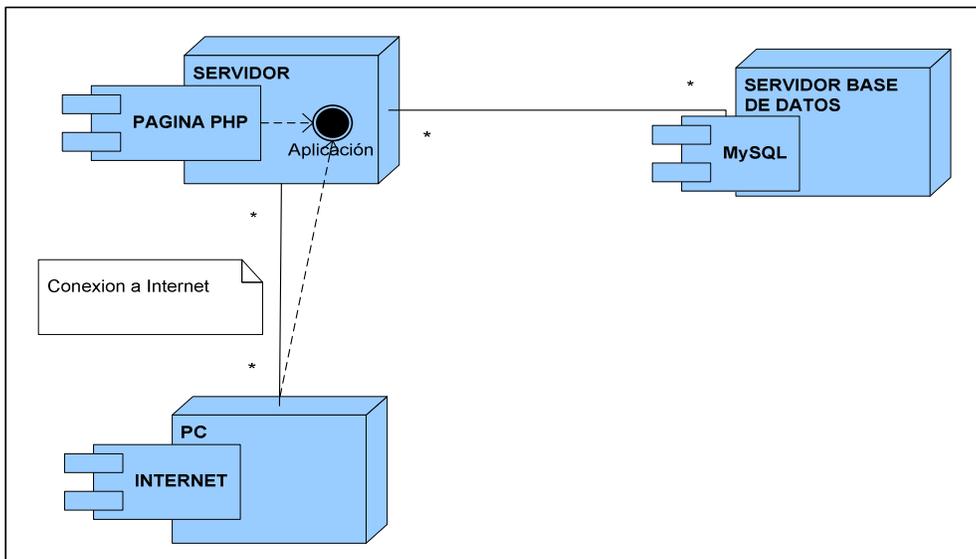


FIG 20 DIAGRAMA DE DESPLIEGUE

5.5.2 Herramientas de desarrollo, ambiente de producción

La aplicación fue desarrollada en una plataforma Web orientada a Internet, para ajustarse a los requerimientos la empresa. El software se va a desarrollar en un gestor de contenidos llamado Joomla! 1.5.3, construido con php bajo una licencia GPL. En cuanto a la base de Datos, la aplicación usara base de datos MySQL.

5.6 SEGURIDAD DEL SISTEMA

5.6.1 Control de acceso al sistema

El usuario al ingresar a la aplicación de registro debe ingresar login y password, el sistema inicialmente verifica que el usuario este registrado en la base de datos, si este cumple con las condiciones de accesibilidad podrá ingresar al sistema, de lo contrario el sistema mostrara una advertencia de que no se encuentra registrado en el sistema o si es una falla en el password muestra una advertencia (“login y password no coinciden”) para que el usuario lo corrija y así posteriormente pueda hacer su ingreso.

5.6.2 Control de acceso a los datos

El acceso a los datos ser manejará, creando los respectivos privilegios del usuario que administrara el sistema en las correspondientes tablas de la base de datos, y en los diferentes módulos a los que este tendrá permiso de acceder, de este modo garantizamos que los usuarios solo tengan permiso para acceder al frontend.

5.6.3 Copias de seguridad

Se realizará backups de la información diariamente, que se almacenará en el Servidor.

6 DISEÑO DE PRUEBAS

6.1 Descripción de escenarios

ID	1
Nombre	Registrar Usuario
Descripción	Es el proceso por medio del cuál un usuario se registra para el posterior ingreso al sistema
Actores	Principal: Usuario (Estudiante)
Prioridad	Muy alta.
Pre -Condiciones	El Usuario puede ingresar al sistema desde cualquier punto no necesariamente tiene que ser en el aula de clase, en cualquier sitio podrá hacer uso de todas las utilidades, teniendo en cuenta las condiciones necesarias de software para ejecutar todos los complementos de la herramienta.
Flujo de eventos principal	El usuario ingresa su login y password y selecciona la opción ingresar al sistema, la aplicación consulta la existencia de dicho usuario y de resultar exitosa, tendrá total acceso al sistema.
Flujo de eventos alternativos	El usuario ingresa su login y password y selecciona la opción ingresar al sistema, la aplicación consulta la existencia de dicho usuario y de resultar no exitosa, el sistema lo redirige a que realice su registro, de ocurrir un error al momento de ingresar login o password el usuario tendrá una advertencia para que lo corrija.
Post-condiciones	Después de llevar a cabo el proceso de registro tendrá total acceso a la herramienta.

ID	2
Nombre	Opinar en Foro
Descripción	Permite crear un mensaje en el foro de discusión
Actores	Usuario logueado en el sistema
Prioridad	Media
Pre-condiciones	El usuario debe haberse logueado en el sistema
Flujo de eventos principal	<ol style="list-style-type: none"> 1. El actor pulsa sobre el botón para crear un nuevo mensaje. 2. El sistema lo envía a una nueva sección donde podrá introducir el título del mensaje y una zona de mayor tamaño para introducir el cuerpo del mensaje.
Flujo de eventos alternativos	<ol style="list-style-type: none"> 1. El actor introduce el título del mensaje y el cuerpo del mismo. 2. El sistema almacena los datos.
Post-condiciones	Ninguno
	El mensaje será almacenado en el sistema

ID	3
Nombre	Crear Usuario
Descripción	Permite registrarse e ingresar al sistema con los permisos para utilizar todas las herramientas del sistema
Actores	Usuario interesados en PnL.
Prioridad	Alta
Pre-condiciones	El usuario debe haberse logueado en el sistema, o en su defecto registrarse para obtener su ingreso.
Flujo de eventos principal	<ol style="list-style-type: none"> 1.El actor pulsa sobre el link <u>regístrate aquí</u> 2.El usuario se loguea para el ingreso 3. El Usuario Crea mensajes en foro, analiza simulaciones, consulta y modifica artículos. 4. El sistema reenvía a la diferentes secciones al usuario según su necesidad
Flujo de eventos alternativos	Ninguno
Post-condiciones	Los mensajes y datos han sido almacenados en la base de datos del sistema

ID	4
Nombre	Administrador del Sistema
Descripción	Permite registrarse e ingresar al sistema con los permisos para utilizar todas las herramientas del sistema
Actores	Administrador
Prioridad	Alta
Pre-condiciones	El administrador debe tener una cuenta de usuario propia con su username y password predefinidos.
Flujo de eventos principal	<ol style="list-style-type: none"> 1. El administrador tendrá un usuario y un password único. 2. El administrador se loguea para el ingreso 3. El administrador crea temas en foro, agrega simulaciones, modifica artículos e interfaz. 4. El sistema ubica al administrador en la categoría, sección o artículo el cual se quiere modificar o eliminar.
Flujo de eventos alternativos	Ninguno
Post-condiciones	Los cambios a la interfaz o en los datos serán almacenados en el sistema

ID	5
Nombre	Modificar Artículos
Descripción	Tanto los usuarios como los administradores tendrán la capacidad de agregar, modificar y eliminar los artículos de cada sección.
Actores	Administrador y Usuario
Prioridad	Media
Pre-condiciones	El administrador y el usuario deben estar logueados
Flujo de eventos principal	<ol style="list-style-type: none"> 1. El administrador y el usuario se loguean 2. El administrador agrega, modifica o elimina artículos 3. El usuario modifica artículos 4. El sistema maneja diferentes interfaces según la persona que este haciendo los cambios.
Flujo de eventos alternativos	El administrador tendrá la posibilidad de dar los permisos a los usuarios para que hagan dichos cambios
Post-condiciones	Todos lo cambios se guardaran en el sistema

Guiados del documento, Objetos, componentes y Estructuras con UML, The Catalysis Approach, por Desmond F. D'Souza y Alan C. Wills, Addison Wesley Longman, 1998.

ID	6
Nombre	Acceso a los contenidos
Descripción	Se le permitirá el acceso a las personas con cualquier perfil, tanto a los artículos (parte teórica), como a los demás componentes como son simulaciones, y animaciones.
Actores	Usuarios y Administrador
Prioridad	Alta
Pre-condiciones	Los usuarios deben estar registrados, el administrador tiene acceso directo.
Flujo de eventos principal	<ol style="list-style-type: none"> 1. Los usuarios luego de estar registrados y haber ingresado al sistema tendrá la posibilidad de acceder de manera directa a los contenidos. 2. El administrador al tener acceso directo al sistema, puede ingresar a cualquier componente del sistema. 3. El sistema al verificar la autenticidad de los datos tanto de usuarios como de administrador, permite el acceso a los diferentes componentes (teoría, simulaciones, animaciones).
Flujo de eventos alternativos	Si el usuario no se ha registrado se le denegaran los permisos para ingresar a los componentes, y de lo contrario se mostrara un mensaje de error, por lo cual deberá regresar a la pagina principal y crear una cuenta para su posterior ingreso.
Post-condiciones	Las personas registradas tendrán acceso libre a los contenidos.

ID	7
Nombre	Publicación de Eventos
Descripción	En esta sección se podrá tener conocimiento de todos los eventos que el administrador (profesor) asignara durante el semestre.
Actores	Administrador
Prioridad	Media
Pre-condiciones	El administrador tendrá la responsabilidad de tener en cuenta las fechas adecuadas para la asignación de lo eventos a realizar.
Flujo de eventos principal	<ol style="list-style-type: none"> 1. El administrador tendrá una sección en la cual podrá asignar todos los eventos de una manera ordenada. 2. Los eventos postulados aparecerán en forma slide, que es una forma agradable de mostrarlos ante el usuario. 3. Para estos eventos el sistema tendrá una sección en la pagina principal, donde los usuarios verán todos los eventos asignados por el administrador.
Flujo de eventos alternativos	Ninguno
Post-condiciones	Asignación de eventos para el periodo presente.

7 DESARROLLO E IMPLEMENTACIÓN

7.1 Descripción del sistema

En esta sección describiremos la forma en que se distribuyeron todos los componentes de la herramienta.

A continuación se muestra la carpeta que contiene el sitio web, y sus respectivas subcarpetas que contienen todos los archivos del sistema, como archivos de texto, imágenes, plugins, etc.

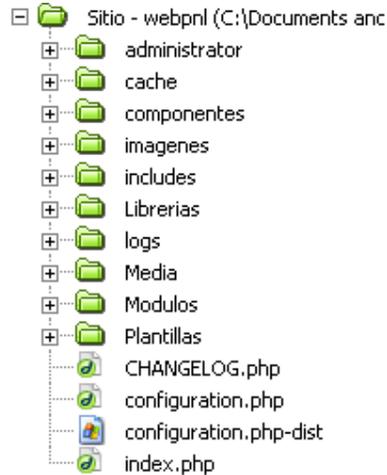


FIG 21 DESCRIPCION DEL SISTEMA

En la figura se muestra parte del código php de las categorías, secciones y artículos que pertenecen al sistema

Filas	introtext
1	<h1>Conceptos Basicos</h1><h1> </h1><p align="just...
1	<h4>Joomla! features:...
1	<h4>General </h4><p>In Joomla! 1.5 all User i...
1	<p align="justify">Definición: x⁻€ F e...
1	<p align="justify">Recordemos que f derivable, es ...
1	<p class="MsoNormal" align="justify"> <img s...
1	<p> </p><p> </p><p><u>Links Comp...
1	<p> </p><p>Antecedentes:</p>...
1	<p> </p><p>Contacto: </p>
1	<p> </p><p>Mapa del Sitio:</...</p>
1	<p> </p><p>Mision</p><p align...
1	<p> </p><p>Preguntas Frecuentes:</st...

FIG 22 BASE DE DATOS

En la figura se muestra la manera en que se aplica la encriptación de los password, dicha aplicación se basa en el algoritmo MD5 (Algoritmo de Resumen del Mensaje 5), el cual es un algoritmo de reducción criptográfico de 128 bits ampliamente usado.

←T→	
Filas	password
1	558912a4273ff0b4ede54e32e8ae5d40:EDoUwrYkjHhTrypdf...
1	59ae45e5b38055dae77f41c833b9d35b:L5ABwO0ERUS4TuwP9...
1	66f9f2c81caa27909568d1aade487a83:14i7W3p1RG8DAu8Ed...
1	e10adc3949ba59abbe56e057f20f883e

FIG 23 ENCRIPCIÓN PASSWORD

Todos los artículos, categorías y secciones del sitio quedan registrados en el sistema de base de datos de la siguiente manera:

←T→	
Filas	name
1	Acerca de PnL
1	Administrador
1	Antecedentes
2	Ayuda
1	Busqueda de Fibonacci
1	Busqueda por Seccion Dorada
1	Caracteristicas
1	Category Blog
1	Category Table
1	Condiciones Kuhn - Tucker
2	Contacto
3	Convexidad
1	Derivada
1	Funcion Unimodal
1	Hessiano
1	Inicio
1	Interpolacion Cuadratica
1	Jacobiano
1	Logout
1	Manual
2	Mapa del Sitio
2	Maximos y Minimos
1	MetaHeuristica
1	Metodo de Ascenso Suficiente
1	Metodo de Cauchy
1	Metodo de Hooke y Jeeves

FIG 24 CONTENIDOS EN BD

El modulo del registro utiliza varios campos, los cuales tiene sus propios parámetros y características, esto queda evidenciado en la siguiente figura la cual muestra nombre del campo, parámetros, etc.,

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra			
<input type="checkbox"/>	id	int(11)			No		auto_increment			
<input type="checkbox"/>	name	varchar(255)	latin1_swedish_ci		No					
<input type="checkbox"/>	username	varchar(150)	latin1_swedish_ci		No					
<input type="checkbox"/>	email	varchar(100)	latin1_swedish_ci		No					
<input type="checkbox"/>	password	varchar(100)	latin1_swedish_ci		No					
<input type="checkbox"/>	usertype	varchar(25)	latin1_swedish_ci		No					
<input type="checkbox"/>	block	tinyint(4)			No	0				
<input type="checkbox"/>	sendEmail	tinyint(4)			Sí	0				
<input type="checkbox"/>	gid	tinyint(3)		UNSIGNED	No	1				
<input type="checkbox"/>	registerDate	datetime			No	0000-00-00 00:00:00				
<input type="checkbox"/>	lastvisitDate	datetime			No	0000-00-00 00:00:00				
<input type="checkbox"/>	activation	varchar(100)	latin1_swedish_ci		No					
<input type="checkbox"/>	params	text	latin1_swedish_ci		No					

FIG 26 ADMINISTRACIÓN BD

7.2 Ejemplos Aplicativos

- **Búsqueda de Fibonacci**

Considere el siguiente problema

$$\text{Min } f(x) = 10 + x^3 - 2x - 5 \exp(x)$$

Deberá realizar lo siguiente:

Resuelva el problema de manera que se encuentre el óptimo con una precisión de 0.0001. Utilice $\alpha = -5.0$ y $\beta = 5.0$. Realizar un reporte donde muestre cómo derivó la cantidad de iteraciones necesarias para alcanzar la precisión requerida. En su reporte debe incluirse el valor de x^* (con la precisión correspondiente) obtenido y el de $f(x^*)$ correspondiente. Estos valores deberán incluirse también en el reporte.

- **Método de la Sección Dorada**

Considere el siguiente problema

$$\text{Min } f(x) = x^2 - 10 \exp(0.1x)$$

Deberá realizar lo siguiente:

Resuelva el problema de manera que se encuentre el óptimo con una precisión de 0.0001. Utilice $\alpha = -10$ y $\beta = 5$. Los resultados deben incluir el intervalo final de incertidumbre (estos valores deberán presentarse sin la normalización), así como el valor de x^* (con la precisión correspondiente) obtenido y el de $f(x^*)$ correspondiente. Estos valores deberán incluirse también en el reporte.

- **Método de Estimaciones Cuadráticas Sucesivas**

Considere el siguiente problema

$$\text{Min } f(x) = (10x^3 + 3x^2 + 5)^2$$

Deberá realizar lo siguiente

1. Realice tres iteraciones del método de estimaciones sucesivas de Powell. Utilice $x_1 = 2.0$ y $\Delta = 0.5$. Incluya todos los pasos del algoritmo así como todos los cálculos intermedios y el resultado final (después de tres iteraciones) en su reporte.

2. Realice tres iteraciones ($N = 3$) del método de estimaciones sucesivas de Powell. Utilice $x_1 = 2.0$ y $\Delta = 0.5$. La respuesta debe incluir los valores de F_{min} y X_{min} , así como los valores de x_1 , x_2 , x_3 y x_m usados por el algoritmo a cada iteración. También deberá obtenerse el valor final de x^* y el de su $f(x^*)$ correspondiente. Estos valores deberán incluirse también en el reporte.

- **Método de Newton-Raphson**

Considere el siguiente problema

$$\text{Max } f(x) = \frac{0.5}{\sqrt{1+x^2}} - \sqrt{1+x^2} \left[1 - \frac{0.5}{1+x^2} \right] + x$$

Deberá realizar lo siguiente

Resuelva el problema de manera que se encuentre el óptimo con una precisión de 1×10^{-4} . Incluya en su reporte los pasos realizados para obtener la primera y segunda derivada de esta función. Utilice $x_1 = 0.1$. En los resultados se deberá incluir el valor de x^* (con la precisión correspondiente) obtenido y el de $f(x^*)$ correspondiente. Estos valores deberán incluirse también en el reporte.

- Resuelva el mismo problema usado con el método de Newton-Raphson, pero adoptando el método de cuasi-Newton. Deberá incluir en un reporte la salida de los resultados producida por el mismo. Utilice $x_1 = 0.1$, $\Delta x = 0.01$ y $\epsilon = 1 \times 10^{-5}$.

Estos 3 valores deberán ser las entradas para resolver el problema, en los resultados se deberá incluir el valor de x^* (con la precisión correspondiente) obtenido y el de $f(x^*)$ correspondiente. Estos valores deberán incluirse también en el reporte

Ejemplos Geométricos

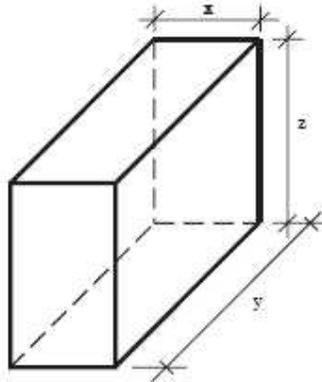
- **Paquete Postal**

Un paquete postal es una caja de dimensiones x , y , z (ver la Figura 3.1), que debe verificar los siguientes requisitos para ser aceptado en la oficina de correos La altura m as el perímetro de la base no puede exceder 108 cm

$$z + 2x + 2y = 108; \quad x, y, z = 0 \quad \text{pues no son posibles las longitudes negativas}$$

Se buscan las tres dimensiones que maximizan el volumen

$$V(x, y, z) = xyz$$



Esquema de las dimensiones del paquete postal

- **El Transporte de Arena**

El coste de transportar arena de un lugar a otro en un contenedor de dimensiones x, y, z es 2 dólares por cada viaje completo. Suponiendo que el precio del material de las paredes superior e inferior, y de los laterales del contenedor son el triple y el doble, respectivamente, de las paredes anterior y posterior, encontrar el precio mínimo de transportar $c = 50\text{m}^3$ de arena. Si z es la altura del contenedor, el precio del mismo será

$$k (3xy + 2(2xz + 2yz) + xy)$$

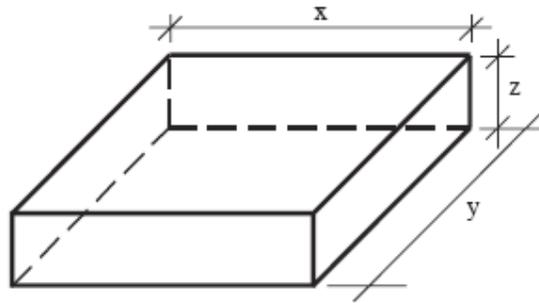
Donde $k = 4$ es una constante de proporcionalidad. Se debe añadir el precio del transporte en sí, que viene dado por

$$2 \frac{50}{xyz}$$

El precio total, que debe minimizarse, es entonces

$$C(x, y, z) = k (3xy + 2(2xz + 2yz) + xy) + 2 \frac{50}{xyz}$$

Bajo las restricciones $x, y, z \geq 0$.

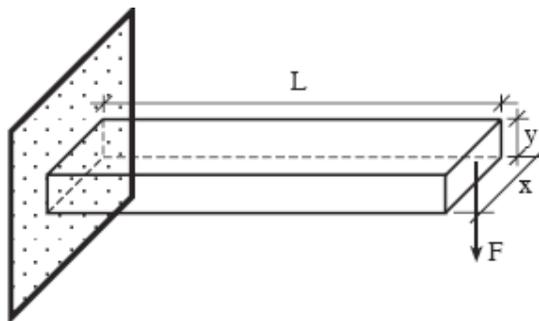


Boceto del problema de transporte de arena

Ejemplos Mecánicos:

Otra categoría importante de ejemplos se refieren a situaciones que surgen en Ingeniería mecánica. Se incluyen varias situaciones típicas que pueden darse en condiciones diversas.

- **El voladizo**



Voladizo

Desearse diseñarse un voladizo de sección rectangular y longitud dada para conseguir peso mínimo, y a la vez asegurar una deformación máxima transversal bajo la acción de una carga vertical actuando en el extremo libre. El material para construir el voladizo tiene un peso específico conocido.

Sean x e y la anchura y altura que se buscan. Sean L , F , S , y ϕ , la longitud, la carga en el extremo libre, la deformación máxima permitida, y el peso específico, respectivamente, esto es, el conjunto de datos de nuestro problema.

El objetivo es minimizar el peso

$$W(x, y) = \phi Lxy.$$

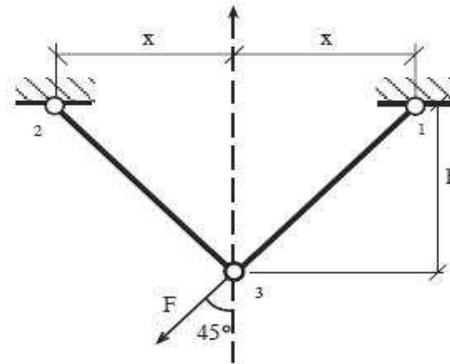
Como la anchura x debe ser mayor que 0.5 y, de acuerdo con la teoría de resistencia de materiales, la deformación en el extremo libre viene dada por $FL^3/3EI$, donde E es el modulo de Young del material del voladizo, e $I = xy^3/12$ es el correspondiente momento de inercia de la sección rectangular, se tiene que minimizar $W(x, y)$ bajo las restricciones:

$$\frac{4FL^3}{Exy^3} \leq S$$

$$x \geq 0.5$$

$$x, y \geq 0$$

- **La estructura de dos barras**



Se desea diseñar la estructura con dos barras según tres objetivos distintos: peso mínimo, la tensión máxima no debe exceder un cierto umbral, y el desplazamiento en el pivote 3 no debe superar un cierto valor. El conjunto de datos del problema es:

- ϕ : el peso específico del material de las barras
- E : el modulo de Young del material de la estructura

- F: la carga que actúa en el pivote fijo con un ángulo de 45 ° a la izquierda del eje Y
- S0: la tensión máxima admisible
- D0: el desplazamiento máximo admisible del pivote 3
- h: la altura de la estructura

Las variables que se necesitan para determinar el diseño óptimo son:

- x: la distancia desde los pivotes fijos al eje Y
- z: el área de la sección de los brazos de la estructura
- D: el desplazamiento del pivote 3
- S¹: la tensión en el pivote 1
- S²: la tensión en el pivote 2
- W: el peso total de la de la estructura

Por tanto debe minimizarse

$$W(x, z) = 2\rho z \sqrt{x^2 + h^2}$$

Bajo

$$D(x, z) = \frac{F}{Eh^2 2\sqrt{2}} \frac{(h^2 + x^2)^{3/2} (h^4 + x^4)^{1/2}}{x^2 z} \leq D_0$$

$$S^1(x, z) = \frac{F}{2\sqrt{2}h} \frac{(h+x)\sqrt{x^2 + h^2}}{xz} \leq S_0$$

$$S^2(x, z) = \frac{F}{2\sqrt{2}h} \frac{(h-x)\sqrt{x^2 + h^2}}{xz} \leq S_0$$

$$x, y \geq 0$$

7.3 Manual del Usuario

Introducción

En esta sección se explicara todas y cada una de las funcionalidades del sistema, con respecto a cada perfil de los usuarios.

La herramienta maneja una interfaz sencilla, la cual se subdividió en categorías, secciones, artículos y los componentes complementarios. Todo se implemento teniendo en cuenta las reglas de usabilidad, en el cual la prioridad es que cada usuario obtenga de la herramienta sus objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso especificado.

USUARIO

Cuando una persona se encuentra con perfil de usuario podrá acceder a la herramienta siguiendo cada uno de los pasos enunciados en el capítulo de UML como lo son registro, activación de cuenta y logueo para el ingreso.

En primera instancia, en el momento del ingreso el sistema exhibirá las diferentes categorías con sus secciones, como se muestra a continuación:

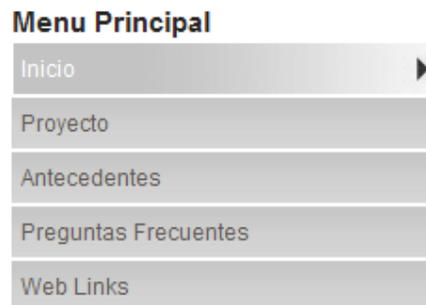


FIG 27 MENÚ PRINCIPAL

1. La categoría Menú Principal esta subdividida en las 5 secciones mostradas en la figura anterior, cada sección en esta categoría esta encargada de mostrar una información general sobre el sistema, a continuación se definirá cada una de estas secciones:

Inicio: Es la sección principal de la pagina puesto que en ella se exhibe la introducción a lo que es programación no lineal, en su concepto matemático.

Proyecto

Esta sección muestra la visión y misión de la herramienta.

Antecedentes (Estado del Arte)

Es un artículo donde se define el inicio del desarrollo del sistema, el cual se basa inicialmente en una investigación, donde se trata de ir tras las huellas del tema, para permitir ubicarnos y determinar como han sido tratados los temas de la investigación (Temas de Programación no Lineal) y cual es su usabilidad en este momento.

Preguntas Frecuentes:

Sección donde se pueden resolver las dudas mas frecuentes.

Web Links

Sección de ayuda donde estarán varios enlaces para poder obtener mucha mas información en Internet sobre los temas tratados en la asignatura.

- 2. La categoría recursos** esta compuesta de 4 secciones en las cuales esta la información y componentes de los algoritmos utilizados en PnL.

Recursos

Metodos de Optimizacion
MetaHeuristica
Administrador
Ayuda

FIG 28 RECURSOS

Métodos de Optimización:

En esta sección se muestra como esta compuesta la programación no Lineal, sus diferentes métodos, y los algoritmos que maneja cada uno de estos.

Especificando lo enunciado observemos las siguientes figuras:



FIG 29 MÉTODOS

En la figura se muestran los métodos a estudiar, tomando el método de búsquedas unidimensionales como ejemplo, haciendo click en este enlace obtendremos una lista con los algoritmos usados en este método.

Ya en la lista de los algoritmos haciendo click en cualquiera de los enlaces se mostrara en el body de la página una interfaz sencilla, especificando en este los ítems relacionados con el algoritmo que se quiere estudiar, como los son su teoría, algoritmo, diagrama de flujo, simulación, y problemas o ejercicios propuestos.

En las siguientes figuras se exhibe la forma en que se implemento el sistema de los algoritmos:

Busqueda por Sección Dorada

[Teoria](#) [Algoritmo](#) [Diagrama](#) [Simulacion](#) [Problemas](#)

Esta es una técnica simple que ubica al óptimo por reducciones repetitivas de un rango de búsqueda. Parte de la evaluación de la función en 2 puntos diferentes a una distancia razonable entre ellos, para luego evaluar un punto situado dentro del rango limitado por aquellos números y reemplazar el extremo más lejano al mínimo (en un problema de maximización) por este punto; tal punto es seleccionado de manera que esté en una proporción de $5/2$ respecto al intervalo de mayor longitud, medido desde el punto medio del intervalo.

El procedimiento es iterativo hasta hallar una aproximación a la solución que sea adecuada. Las ventajas de este método de búsqueda son su eficiencia y el hecho de que no emplea información de la derivada de la función, aunque su uso está confinado a funciones convexas.

Pertenece a los métodos de búsqueda lineal basados en intervalos, además es una versión mejorada de la búsqueda de Fibonacci.

En la búsqueda de la Sección Dorada se usan tres valores de la función para detectar el valor extremo, se toma un cuarto número, y se determina donde ocurre el mínimo, en los primeros tres o los últimos tres valores.

Se minimiza la evaluación de la función objetivo al reemplazar los valores anteriores con los nuevos, haciendo que se cumplan las siguientes condiciones: La primera condición especifica que la suma de las dos sublongitudes l_1 y l_2 debe ser igual a la longitud original del intervalo. La segunda indica que el cociente o razón de las longitudes debe ser igual.

$$l_0 = l_1 + l_2 \quad (1)$$

FIG 30 PRESENTACIÓN CONTENIDOS

Busqueda por Seccion Dorada

Teoria Algoritmo Diagrama Simulacion Problemas

```
function [xo,fo] = opt_gs(f,a,b,r,TolX,TolFun,k)
h = b - a;
rh = r*h; c = b - rh; d = a + rh;
fc = feval(f,c);
fd = feval(f,d);
if k == 0 | (abs(h) < TolX & abs(fc - fd) < TolFun)
if fc <= fd, xo = c; fo = fc;
else xo = d; fo = fd;
end
if k == 0, fprintf('Solo los mejores resultados de las iteraciones'), end
else
if fc < fd, [xo,fo] = opt_gs(f,a,d,r,TolX,TolFun,k - 1);
else [xo,fo] = opt_gs(f,c,b,r,TolX,TolFun,k - 1);
end
end
```

FIG 31 VISTA DEL MODULO ALGORITMO

Busqueda por Seccion Dorada

Teoria Algoritmo Diagrama Simulacion Problemas

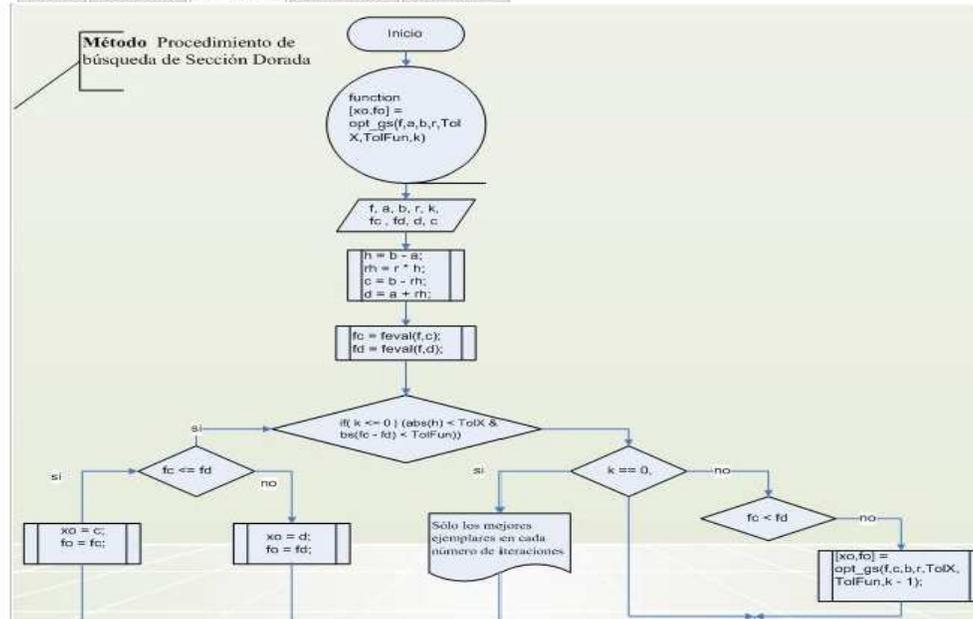


FIG 32 VISTA MÓDULO DIAGRAMA

Busqueda por Seccion Dorada



FIG 33 VISTA MODULO ANIMACIONES

Busqueda por Seccion Dorada

Teoria Algoritmo Diagrama Simulacion Problemas

Considere el siguiente problema $\text{Min } f(x) = x^2 - 10 \exp. (0.1x)$ Deberá realizar lo siguiente: Resuelva el problema de manera que se encuentre el óptimo con una precisión de 0.0001. Utilice $\epsilon = 10^{-5}$ y $\delta = 5$. Los resultados deben incluir el intervalo final de incertidumbre (estos valores deberán presentarse sin la normalización), así como el valor de x^* (con la precisión correspondiente) obtenido y el de $f(x^*)$ correspondiente. Estos valores deberán incluirse también en el reporte.

FIG 34 VISTA MODULO PROBLEMAS

La herramienta esta diseñada con el fin de ofrecerle al usuario comodidad al momento de estudiar los algoritmos.

Aplicando el modo de tabs en la interfaz se propuso que el usuario obtuviera mucha mas eficiencia en el manejo y la comprensión de estos temas, ya que tendrá la posibilidad de ver de manera casi paralela todas las características de estos, cubriendo las falencias y dudas del usuario.

Metaheurística

En esta sección se definen conceptos básicos de la heurística y los algoritmos más importantes de este tipo de métodos, a su vez estos algoritmos se implementaron de la misma manera que los algoritmos de los métodos de la sección anterior.

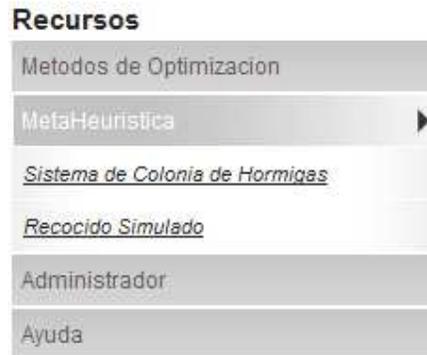


FIG 33 VISTA SECCIÓN METAHEURÍSTICA

Administrador

El administrador tendrá la oportunidad ingresar a este modulo y realizar los cambios en el sistema desde la siguiente interfaz.



FIG 34 MODULO ADMINISTRADOR

Categoría Conceptos Clave

En esta categoría se tienen nombrados de manera específica cada uno de los conceptos claves, los cuales son muy importantes tener claros sus conceptos ya que estos son la base de todos los algoritmos usados, y directamente ayudaran al aprendizaje óptimo de los métodos.

Conceptos Clave

Maximos y Minimos
Convexidad
Optimos Locales y Globales
Condiciones Kuhn - Tucker
Derivada
Hessiano
Jacobiano

Modulo Formulario de Acceso para el Usuario

En este formulario solo se registraran y loguean usuarios como estudiantes, el esquema de este formulario es el siguiente:

Formulario de Acceso

Nombre de usuario

Contraseña

Recordarme

[¿Olvidó su contraseña?](#)

[¿Olvidó su nombre de usuario?](#)

[¿Quiere registrarse? Regístrese aquí](#)

técnicas heurísticas y técnicas inteligentes como Colonia de Hormigas, Recocido Simulado, Búsqueda Tabú, Algoritmos Genéticos, entre otras. En general, están basados en la aleatoriedad y en algunos criterios obtenidos de la experiencia del diseñador para encontrar una buena solución. Estas técnicas usan la función objetivo solo para evaluar la calidad de las soluciones, en muchas ocasiones ignorando la información matemática (gradiente) del problema que se encuentra en esta función.

La Programación No Lineal (PNL) provee una serie de herramientas que manipulan en forma estricta los espacios de búsqueda de solución de los problemas, aprovechan información matemática del problema para dirigirse en cada paso hacia un punto de buena calidad, mejorando de esta manera la llegada a la solución. Además, PNL permite el modelamiento de restricciones no lineales, una característica muy útil para la formulación dada en el presente trabajo a los problemas que involucran variables enteras. Estas características mencionadas se deben a que en problemas de PNL, el cumplimiento de las condiciones de Karush-Kuhn-Tucker (condiciones de primer orden) y algunas condiciones de segundo orden son requeridas para evaluar la factibilidad y la optimalidad de los puntos que se van encontrando.

FIG 35 VISTA MODULO FORMULARIO DE ACCESO

El formulario de acceso de datos es el que se muestra en la siguiente figura, al momento de ingresar los datos y dar click en registrarse este envía un correo al E-mail asignado por el usuario para que active su cuenta en el sistema y posteriormente de este proceso pueda hacer su ingreso a los componentes del sistema.

Registro

Los campos marcado con un asterísco (*) son obligatorios

Nombre: *

Nombre de usuario: *

E-mail: *

Contraseña: *

Verifique su clave: *

Formulario de Acceso para el Administrador

Esta sección es única y restringida para que solamente el administrador tenga la oportunidad de acceder a los componentes del sistema, y pueda realizar los cambios que el considere necesarios.

Acceso a la administración

Usa un nombre de usuario y contraseña válido para poder tener acceso a la administración.

[Regresar a la página de inicio](#)

Nombre de usuario

Contraseña

Idioma



Existen varios componentes que son un complemento del sistema como lo es el encuesta, foro, usuarios en línea, estos están diseñados de la siguiente forma:

En la encuesta lo que se pretende es saber que acogida a tenido el sistema, y según estos resultados en un futuro renovar el sistema según la opinión de los usuarios.

Esta encuesta muestra los resultados por cantidad de votos, porcentaje y gráficamente, esto se puede observar en la Figura.

Encuesta

Encuesta sobre el sitio

Te parece util el sitio

Ayuda con tu actividad academica

No lo ves funcional

Te parece una perdedera de tiempo

[Resultados](#)

Seleccione una encuesta

Número de votantes

2

Primer voto

Miércoles, 16 de Julio de 2008 11:26

Último voto

Miércoles, 16 de Julio de 2008 16:37

Encuesta sobre el sitio

Hits	Percent	Graph
Te parece util el sitio		
1	50%	
Ayuda con tu actividad academica		
1	50%	
No lo ves funcional		
0	0%	
Te parece una perdedera de tiempo		
0	0%	

FIG 36 VISTA MODULO ENCUESTA

Cada usuario tendrá la oportunidad de votar una sola vez en el día, si después de haber votado se intentara de nuevo, el sistema hará que aparezca el siguiente mensaje:

- **Usted ya votó en esta encuesta hoy!**

8. CONCLUSIONES

- El sitio desarrollado para la asignatura de Programación no Lineal permite que los usuarios desplieguen sus conocimientos básicos de cálculo, y los lleven al nivel de desarrollar matemáticamente procesos por medio de algoritmos de optimización
- La herramienta esta compuesta de toda la información necesaria para desarrollar todo el plan de la asignatura PnL, esto se logra disponiendo en un mismo sistema de conceptos básicos, desarrollo de los algoritmos en código y en diagrama de flujo, como también de animaciones para el completo entendimiento de los métodos de optimización.
- La herramienta maneja una interfaz sencilla y agradable, en la cual los usuarios podrán acceder a todos los módulos, y a la información descrita en ellos de una manera eficiente debido a los parámetros de usabilidad utilizados,
- El uso de la herramienta se podrá realizar desde cualquier sitio donde se cuente con acceso a Internet, con esto se logra que toda la comunidad tenga un acceso sin restricciones, aparte de las que tiene el sistema.
- El administrador tendrá la capacidad de realizar todos los cambios necesarios en el sistema, desde una interfaz sencilla, guiándose de un manual de usuario que se le entregara al momento de implementar el sistema.
- Este software servirá como complemento educativo para el mejoramiento en el aprendizaje significativo de la asignatura, debido a que este se compone de muchas áreas, las cuales fueron vistas en el transcurso de nuestra carrera, y ayudara al refuerzo de todas ellas.
- Las herramientas basadas en las TIC's para proponer metodologías de aprendizaje se encuentran en desarrollo, es bajo el uso de estándares y no existen métodos confiables para validar el éxito de las mismas. El trabajo realizado arrojó resultados satisfactorios a nuestras métricas personales y las expectativas de nuestros mentores, pero aun mas importante que las herramientas en si, es la gestión que se haga en torno de las aplicaciones y la cultura de apoyo tecnológico, para conocer expectativas, mejoras, impacto y resultados.
- El aprendizaje significativo se hace difícil de valorar en cuanto a lo cuantitativo, el diseño de la herramienta se hizo en base a alta usabilidad y vinculación de multimedia explicativa, esta se ve que una forma de alta aceptación por los usuarios

9. RECOMENDACIONES

- Se recomienda que los usuarios tengan una actitud responsable en el manejo de la herramienta ya que ellos tendrán un permiso, que les permite realizar cambio es varios de los artículos, todo esto para que el desarrollo del plan se lleve a cabo exitosamente.
- El mantenimiento y administración deberá realizarse únicamente por el administrador, o por personal autorizado el cual tenga los conocimientos suficientes de la herramienta, para así evitar problemas de funcionalidad en el sistema.
- El sistema fue desarrollado de cierta forma que este se puede actualizar y renovar según las necesidades de los usuarios. Sería necesario el aporte tanto del administrador como de los usuarios para que la herramienta sea cada vez más robusta, y con alcances superiores a los logrados en el actual sistema.

10. BIBLIOGRAFÍA

- Bazaraa, Shetty and Sherali, Nonlinear Programming: Theory y Applications. Wiley, 1994.
- Fletcher, R., Practical Methods of Optimization. Wiley, 1987.
- Goldberg, D.E., Genetic Algorithms in Search, Optimization y Machine Learning, Addison-Wesley (1989).
- Himmelblau, Applied Nonlinear Programming. McGraw-Hill, 1972.
- Luenberger, Introduction to Linear and Nonlinear Programming. Addison Wesley, 1984.
- Nash, S. and Sofer, A., Linear and Nonlinear Programming. McGraw-Hill, 1996.

INTERNET

- www.ica.luz.ve/~lzerpa/OptimizacionParalIngenieros.html
- <http://es.wikipedia.org/wiki/Heur%C3%ADstica>
- <http://es.wikipedia.org/wiki/Metaheur%C3%ADstica>
- [ci2s.ugr.es/publications/ficheros/OCH%20Modelos%20y%20Nuevos%20Enfoques%20\(Chapter\).pdf](http://ci2s.ugr.es/publications/ficheros/OCH%20Modelos%20y%20Nuevos%20Enfoques%20(Chapter).pdf)
- <http://es.kioskea.net/genie-logiciel/cycle-de-vie.php3>
- <http://yaqui.mxl.uabc.mx/~molguin/as/RUP.htm>
- www.dm.uba.ar/materias/optimización/2005/1/SimAnneal.doc