

LIBRO FUNDAMENTOS DE PROGRAMACIÓN,
GUÍA DE AUTOENSEÑANZA

CARLOS ANDRÉS ZAPATA OSPINA

UNIVERSIDAD DE MANIZALES
FACULTAD DE INGENIERIA
INGENIERIA EN SISTEMAS Y TELECOMUNICACIONES
MANIZALES
2006

LIBRO FUNDAMENTOS DE PROGRAMACIÓN,
GUIA DE AUTOENSEÑANZA

CARLOS ANDRÉS ZAPATA OSPINA

Trabajo de grado para optar el título de
Ingeniero en Sistemas y Telecomunicaciones

Presidente de trabajo de grado
DIEGO FERNANDO QUINTERO
Ingeniero Electrónico

UNIVERSIDAD DE MANIZALES
FACULTAD DE INGENIERIA
INGENIERIA EN SISTEMAS Y TELECOMUNICACIONES
MANIZALES
2006

Nota de aceptación

Firma del presidente del jurado

Firma del jurado

Firma del jurado

Manizales 27 de noviembre de 2006

*A mí dedicada, comprensiva y amorosa familia,
A mis maestros, Víctor Hugo Angel y Marlon Manrique.
A todos mis antiguos, presentes y futuros alumnos.*

AGRADECIMIENTOS

El desarrollo de este trabajo de grado fue posible al apoyo incondicional de personas que han creído en mis capacidades y que de muchas maneras han aportado su granito de arena.

De todo corazón mil gracias a: Diego Fernando Quintero presidente de trabajo de grado, al “flaco” Luis Carlos Correa asesor metodológico, Samuel Piedrahita asesor, Andrés Mauricio Martínez quien me guío en el cuento de los compiladores, Hector José Sarmiento quien me retó en el cuento de escribir, a Carlos Cortés Carrillo quien siempre ha creído en mi, a Jaime Uldarico Cardona quien experimento con todo este proyecto, Carlos Betancourth quien tiene en cuenta mis aportes.

Agradecimiento especial a mi ángel de la guarda, Sra. Dora González

CONTENIDO

pág.

INTRODUCCIÓN

1. DESCRIPCION DEL AREA PROBLEMÁTICA

2. OBJETIVOS

2.1. OBJETIVO GENERAL

2.2. OBJETIVOS ESPECÍFICOS

3. JUSTIFICACIÓN

4. MARCO TEÓRICO

5. METODOLOGÍA

6. RESULTADOS

7. CONCLUSIONES

8. RECOMENDACIONES

BIBLIOGRAFIA

RESUMEN

Con la creación de la tipografía en el año de 1440 por el impresor alemán Johannes Gensfleisch Gutenberg y perfeccionada años después (1448), la masificación de la literatura adquirió la importancia que antes se encontraba limitada por la producción de manos de escritores. La más famosa de las impresiones fue la Biblia latina. La historia de los medios de comunicación para difundir conocimientos comenzaba a cambiar de una manera global, los libros no serian mas entonces recursos exclusivos de algunos pocos, sino con un gran impulso cultural un derecho para todos.

Los libros son el medio de comunicación mas usado por todo el mundo, desde sus humildes inicios hasta el día de hoy recorren todas las lenguas, dialectos y tipos de escritura como el código Braille. Existen sobre todos los géneros y temáticas, destacando entre los más importantes: los cuadernillos de apuntes de Leonardo da Vinci (considerados tesoros de la humanidad), el quijote de la mancha, el poema de Cid campeador, la Biblia, el Coran y los derechos del hombre.

La tecnología moderna se ha apoyado innumerablemente en la producción escrita de obras para difundir conceptos. Por cada equipo de computo, herramienta software o hardware se produce un tutorial de uso, un manual de especificaciones, en resumen un libro. Además la tecnología ha querido reemplazar este medio de comunicación por uno digital con la creación de los ebooks (libros electrónicos) con la limitante que la actividad de la lectura en una pantalla de computador o cualquier dispositivo móvil desgasta la visión y hace de la lectura una dinámica pausada y molesta.

Una obra escrita por si sola no tiene la capacidad de reemplazar el método que realiza un facilitador humano en el proceso enseñanza aprendizaje en sus etapas iniciales. Cuando el lector tiene un cierto nivel en el tema o en técnicas de aprendizaje puede acogerse al llamado modelo de autoaprendizaje o auto enseñanza (autodidacta), donde el libro contiene una serie de pasos previamente definidos bajo un modelo pedagógico de aprendizaje que le permitirán abordar una temática.

ABSTRACT

With the creation of the typography in the year of 1440 by the German printer Johannes Gensfleisch Gutenberg and perfected years later (1448), the masificación of written Literature acquired the importance that before was limited by the production of hands of writers. Most famous of the impressions it was the Latin Bible. The history of mass media to spread knowledge began to change of a global way, the books do not serian but then exclusive resources of some few, but with a great cultural impulse a right for all.

The books are the mass media but used everywhere, from their humble beginnings to today cross all the languages, dialectos and types of writing like the Braille code. They exist on all the thematic sorts and, emphasizing between most important: the booklets of notes of Leonardo give Vinci (considered treasures of the humanity), quijote of the spot, the poem of valiant Cid, the Bible, the Coran and the rights of the man.

The modern technology has leaned innumerably in the written work production to spread concepts. By each equipment of I compute, software tool or hardware takes place a use tutorial, a manual of specifications, in summary a book. In addition the technology has wanted to replace this mass media by one digitalis with the creation of ebooks (electronic books) with the limitante that the activity of the reading in a screen of computer or any movable device wears away the vision and makes of the reading a slow dynamics and bothers.

A written work by itself does not have the capacity to replace the method that a human facilitador makes in the process education learning in its initial stages. When the reader has a certain level in the subject or in learning techniques model of autoaprendizaje or car can take refuge in the call education (self-taught), where

the book contains a series of steps previously defined under a pedagogical model of learning that will allow him to approach a thematic one.

INTRODUCCIÓN

La producción de material escrito se ha convertido en uno de los objetivos principales de las entidades de educación, ya que se considera que una organización educativa es productora de conocimiento cuando tiene la capacidad de generar nuevas ideas y puede expresarlas.

En la actualidad, Colombia posee bajos índices de producción escrita, siendo más crítica la situación, que no se tiene una cultura en pos de la lectura. Es difícil generar materiales escritos en donde ni siquiera se usan de la manera más adecuada. A partir de lo anterior se ha ido estableciendo una nueva visión e importancia, a que cada institución fomente el desarrollo de producción escrita que tanto fortalece el manejo del conocimiento y miden de manera directa el dominio que se tiene.

Las ciencias de la computación se han apoyado casi en su totalidad en los medios impresos a través de libros, manuales, tutoriales, entre otros para difundir conceptos que son interpretados y usados en diferentes ambientes. Toda la literatura técnica para el habla latina proviene de otros países, y son traducciones realizadas en México o España.

A partir del anterior estado del arte, el fin de este documento es la presentación del libro "Fundamentos de Programación - Guía de AutoEnseñanza", es una obra escrita dedicada a la tecnología de computación en el área de programación de computadoras, hecho tangible gracias a la definición de un área problemática precisa para un problema anteriormente establecido y formulado en el anteproyecto que precede este trabajo final.

Entre los resultados más relevantes se citan: la obtención de una obra escrita con las características establecidas desde el inicio de la propuesta, con el acompañamiento de una herramienta software donde el alumno podrá poner a prueba sus avances y conocimientos. La estructura del libro se plantea de una manera amena, con una gran calidad en la diagramación para la presentación de contenidos. Además contiene muchos ejemplos explicados paso a paso por cada capítulo, ejercicios para desarrollar y un lenguaje sencillo.

1. DESCRIPCIÓN DEL ÁREA PROBLEMÁTICA

Los fundamentos, metodología o introducción a la programación, corresponden a uno de los principales componentes en el currículo de las ciencias de la computación. En el ámbito nacional y latino americano, incluyendo España, la asignatura está constituida de forma general por los siguientes conceptos: algoritmia, diseño y construcción de algoritmos y estructuras de datos.

La diferencia se acentúa en las estrategias, perfiles e intereses de cada institución; las cuales habitualmente proporcionan el conocimiento de alguna de las siguientes formas:

Determinados centros adoptan el proceso educativo, asumiendo los aspectos algorítmicos y enseñarlos en un lenguaje natural al estudiante, haciendo uso de textos y ejercicios escritos.

Por otro lado, otros institutos toman un lenguaje de programación de tercera generación, como C, Pascal o Basic, Java, Delphi, .Net por ejemplo, adecuando la serie de elementos básicos a las reglas del lenguaje seleccionado. Esta opción permite un primer acercamiento en la interacción del hombre sobre la máquina.

La desventaja de la primera, radica en el desconocimiento de cómo el usuario se relaciona con el computador, controlando y ejecutando procesos; requiriendo este tipo de enseñanza de un alto nivel de abstracción. La siguiente, aunque cuenta con el apoyo de los recursos computacionales de la que carece la anterior, falla en la forma como aprovecha el lenguaje, ya que se hace mayor hincapié en el aprendizaje de la sintaxis del lenguaje específico y luego la conexión a los principios de programación. Esta razón destaca la problemática de cómo se

transforma una herramienta computacional, diseñada con el propósito de dar soporte al resolver un problema, en donde prima el conjunto de componentes del lenguaje, antes que en la conceptualización.

La fundamentación tiene como objetivo, la aproximación al entendimiento en la sistematización de computadoras de una manera fácil, transparente y amena, desde el conjunto de conceptos básicos del estudiante. Un acercamiento hacia la solución ideal con respecto a herramientas de aprendizaje o simuladores, la han planteado el proyecto DFD de la Universidad del Magdalena, en Santa Marta, con una herramienta de implementación de algoritmos en diagrama de flujo, y su homólogo el diseño del lenguaje LORO de la Universidad Autónoma de Manizales en Pseudocódigo basado en Java y C++.

Los componentes que facilitan el aprendizaje de algoritmos son: el diagrama de flujo como elemento gráfico representativo, similar a organigramas de las compañías, donde se visualiza la jerarquía de mando y procesos y; el Pseudocódigo como parte narrativa de un proceso. El estudio de manera minuciosa de cualquiera de los dos, garantiza el correcto aprendizaje de fundamentos de programación, permitiendo el fluido proceso de la enseñanza, desde dos perspectivas diferentes con un fin común.

Actualmente existe poca variedad de libros guías sobre fundamentos de programación, destacando la poca producción nacional sobre la materia y temas técnicos, ya que la documentación que llega al país es generalmente de la península ibérica, o traducciones de escritores de habla inglesa. En un país que vive cambios a nivel educativo, vistos inicialmente en las pruebas de calidad superior "Ecaes" y el sistema de créditos, que garantizan el mejoramiento de la educación superior y pretenden competir con el mundo, es ligera la posición de los profesionales para mostrar su dominio y punto de vista referente a un tema, al no plasmar su genialidad y difundirla al mundo. Es también baja la cantidad de

publicaciones y escritos que hacen las universidades para con sus estudiantes y para difusión mundial: en un país donde el hábito de lectura es pobre y el de producción escrita es mínimo.

Es importante destacar que las carreras profesionales con énfasis computacional son nuevas comparadas con la ingeniería civil, eléctrica, entre otras, pero es en la computacional donde se han realizado grandes avances tecnológicos. Es un amplio terreno para llenar vacíos y potenciar conocimientos mediante difusión escrita de trabajos

Mientras la producción escrita aumente, el nivel académico y cultural crece proporcionalmente; según sean la cantidad y calidad de escritos de una comunidad se sabe qué habla esa comunidad.

2. OBJETIVOS

2.1. OBJETIVO GENERAL

Elaborar un libro sobre la temática de fundamentos de programación con un manejo práctico y didáctico.

2.2. OBJETIVOS ESPECÍFICOS

1. Plasmar mediante un libro los conocimientos alcanzados, en pos de facilitar y proporcionar herramientas en el proceso educativo de fundamentos de programación.
2. Brindar un punto de vista más práctico en lo referente a documentación técnica de algoritmia, teniéndose en cuenta el proceso que realizan los estudiantes para adquirir el conocimiento y, los docentes en proporcionarlo.
3. Innovar con respecto a los demás libros técnicos de fundamentos de programación, con la inclusión de un capítulo que comprende: las técnicas para la resolución de problemas y la exposición de una cierta cantidad de ejercicios que motiven el desarrollo de la lógica.
4. Generar una nueva expectativa sobre la producción escrita y la importancia que toma sobre investigaciones y desarrollos, orientando focos de mejoramiento que apunten al fortalecimiento y garantía de la calidad.
5. Explorar sobre la combinación de textos técnicos de algoritmia con herramientas de desarrollos de los mismos, como si se estuviese aprendiendo un lenguaje de alto nivel.

6. Facilitar la vinculación temprana de los recursos computacionales, en el proceso educativo como medio de aprendizaje interactivo.

3. JUSTIFICACIÓN

Esta propuesta es importante para toda la comunidad académica relacionada con las ciencias de la computación, independientemente del nivel, la zona geográfica o institución; por que contribuye en diferentes ambientes, y está apoyada en los estudios teóricos y prácticos realizados de manera separada por varios centros de investigación. Primero, plasma la efectividad del pseudolenguaje UPSAM de la Universidad Pontificia de Salamanca en Madrid, en España, en una dimensión práctica real, como medio de aprendizaje estandarizado para el desarrollo de algoritmos. Segundo, se plantean los conocimientos dados sobre un proceso educativo y confrontados con la experiencia del autor, en un libro que acompañe y guíe al interesado en el camino de la programación de computadoras, proporcionando unas bases prácticas y didácticas.

Los aportes de manera concreta al campo de la computación en los marcos teóricos y tecnológicos, se visualizan en el material educativo que podrá emplearse de las siguientes formas:

- Con el acompañamiento de un docente como facilitador del conocimiento, para la enseñanza dirigida.
- Para la creciente tendencia de estudio de autoaprendizaje.

Dadas las características y la calidad de la obra, contribuirá a este caso de estudio y por tanto, no corresponde a la simple generación de un manual de algoritmos, sino a la producción de un libro con las estructuras temáticas enfocadas al aprendizaje de la programación en general, permitiendo profundizar en un lenguaje de programación con mayor facilidad.

El resultado de todo este proceso en una primera instancia, es una obra escrita que facilite el aprendizaje desde el punto de vista del principiante en algoritmia, que presente muchos ejercicios explicativos desarrollados paso a paso y permita un primer acercamiento e interacción con el computador empleando herramientas de uso gratuito. La visión proyectada en la presente propuesta, pretende servir como modelo de aprendizaje a toda la comunidad de habla hispana, como un estándar para el mejoramiento en los pilares de programación, a partir de la importancia en el desarrollo de algoritmos con una obra exclusiva y dedicada en esta inicial e importante etapa de programación de computadoras.

Y como segunda y más importante instancia, la producción literaria de un material bibliográfico que permita desde diferentes usos brindar una referencia temática, sobre fundamentos de programación vista desde el entorno nacional y con proyección internacional, esperando al mejoramiento crítico por personas o entes que fomenten la literatura técnica en computación y, de manera sencilla, incursionar en este nuevo campo desde una perspectiva nacional.

Los beneficios derivados de esta propuesta de grado, se pueden percibir en la potencialidad que ofrece la obra técnica acompañada de las herramientas de ejecución de algoritmos, apoyando el proceso educativo de fundamentación en programación, sacando provecho a la tecnología actual, en el fortalecimiento de escenarios que faciliten la adquisición de conocimiento con una mayor interactividad y lúdica. Además de la creación de nuevos espacios de trabajo de investigación sobre herramientas computacionales para el máximo aprovechamiento del computador, también, la motivación y la expectativa de plasmar, medir y dirigir los conocimientos al público mundial mediante la comunicación escrita permitiendo una nueva visión del país desde una mirada tecnológica proponente.

4. MARCO TEÓRICO

4.1. ALGORITMIA

Algoritmo, según la Real Academia, es un conjunto ordenado y finito de operaciones que permite encontrar la solución a un problema cualquiera. Ejemplos sencillos de algoritmos son una receta de cocina o las instrucciones para armar una bicicleta. Los primeros algoritmos registrados datan de Babilonia, originados en las matemáticas como un método para resolver un problema usando una secuencia de cálculos más simples. Esta palabra tiene su origen en el nombre de un famoso matemático y erudito árabe del siglo IX, Al-Khorezmi, a quien también le debemos las palabras guarismo y álgebra. Actualmente algoritmo se usa para denominar a la secuencia de pasos a seguir para resolver un problema usando un computador (ordenador). Por esta razón, la algoritmia o ciencia de los algoritmos, es uno de los pilares de la informática (ciencia de la computación en inglés).

Un algoritmo es la especificación detallada de los pasos necesarios para llevar a cabo una tarea específica. En la escuela, por ejemplo, se aprende una serie de pasos para realizar sumas, restas, multiplicaciones y divisiones, estos son algoritmos. Una receta de cocina es uno de los ejemplos más claros de algoritmo. El conjunto de trámites que debe realizar en cualquier institución pública para la expedición de un documento, como por ejemplo, la licencia de conducir, también es un algoritmo. Un programa de computo es un algoritmo hecho para ser ejecutado por un ordenador. La algoritmia también es un arte, pues la creación de un algoritmo requiere de grandes dosis de heurística y creatividad. La creación de un programa va a estar muy influida por factores externos al modelo matemático, como por ejemplo, el lenguaje de programación que se va a usar, las limitaciones físicas que imponen los ordenadores o incluso los plazos para la finalización de un proyecto.

4.2. DISEÑO Y ANÁLISIS DE ALGORITMOS

El desarrollo de un algoritmo tiene varias etapas. Primero se modela el problema que se necesita resolver, a continuación se diseña la solución, luego ésta se analiza para determinar su grado de corrección y eficiencia, y finalmente se traduce a instrucciones de un lenguaje de programación que un computador entenderá. El *modelo* especifica todos los supuestos acerca de los datos de entrada y de la capacidad computacional del algoritmo. El *diseño* se basa en distintos métodos de resolución de problemas, muchos de los cuales serán presentados más adelante. Para el *análisis* de un algoritmo debemos estudiar cuántas operaciones se realizan para resolver un problema. Si tenemos un problema x diremos que el algoritmo realiza $A(x)$ operaciones (costo del algoritmo). Al valor máximo de $A(x)$ se le denomina el peor caso y al mínimo el mejor caso. En la práctica, interesa el peor caso, pues representa una cota superior al costo del algoritmo. Sin embargo, en muchos problemas esto ocurre con poca frecuencia o sólo existe en teoría. Entonces se estudia el promedio de $A(x)$, para lo cual es necesario definir la probabilidad de que ocurra cada x , $p(x)$, y calcular la suma ponderada de $p(x)$ por $A(x)$. Aunque esta medida es mucho más realista, muchas veces es difícil de calcular y otras ni siquiera podemos definir $p(x)$ porque no conocemos bien la realidad o es muy difícil de modelar. Si podemos demostrar que no existe un algoritmo que realice menos operaciones para resolver un problema, se dice que el algoritmo es *óptimo*, ya sea en el peor caso o en el caso promedio, dependiendo del modelo. Por esta razón, el análisis realimenta al diseño, para mejorar el algoritmo.

4.3. EL PSEUDOCÓDIGO

El Pseudocódigo es un lenguaje de pseudoprogramación utilizado para escribir algoritmos computacionales. Como lenguaje de pseudoprogramación, el pseudocódigo es una imitación de uno o más lenguajes de programación. De esta

manera podemos encontrar pseudocódigos orientados a lenguajes de programación como Pascal, Java, C, C++, etc. En el caso de este curso orientaremos los pseudocódigos a los lenguajes Java, C y C++. El objetivo del pseudocódigo es permitir que el programador se centre en los aspectos lógicos de la solución, evitando las reglas de sintáxis de los lenguajes de programación. No siendo el pseudocódigo un lenguaje formal, los pseudocódigos varían de un programador a otro, es decir, no hay un pseudocódigo estándar.

4.4. VARIABLES

Una variable es una localización o casillero en la memoria principal que almacena un valor que puede cambiar en el transcurso de la ejecución del programa. Cuando un programa necesita almacenar un dato, necesita una variable. Toda variable tiene un nombre, un tipo de dato y un valor. Antes de poder utilizar una variable es necesario declararla especificando su nombre y su tipo de dato. Para declarar variables usaremos los siguientes formatos:

Declaración de una variable:

tipo nombre

Declaración de varias variables con el mismo tipo de dato:

tipo nombre1, nombre2, nombre3, ..., nombren

Donde:

tipo

Es el tipo de dato de la variable que puede ser: ENTERO (si la variable almacenará un número entero) , REAL (si la variable almacenará un número decimal) , CHARACTER (si la variable almacenará un carácter), CADENA (si la variable almacenará un conjunto de caracteres) o LOGICO (si la variable almacenará el valor verdadero o el valor falso).

nombre1, nombre2, ..., nombren

Nombres de las variables . El nombre de una variable debe comenzar con una letra, un símbolo de subrayado o un símbolo de dólar. Los demás caracteres del nombre puede ser letras, símbolos de subrayado o símbolo de dólar. Debe considerarse también que una letra mayúscula se considera diferente de una letra minúscula.

4.5. LENGUAJE DE MÁQUINA

Cada tipo de microprocesador contiene un conjunto de instrucciones que realizan ciertas operaciones sobre una o más palabras de bits; las instrucciones van también codificadas en bits. No queremos hacer aquí una discusión sobre arquitectura de ordenadores, por lo que con esto debe valer por ahora.

Se entiende que escribir sólo con dos teclas, el 0 y el 1, es incómodo. Históricamente, a la hora de diseñar un algoritmo para que el ordenador ejecutara, se escribía mediante unas etiquetas mnemotécnicas; éste fue el origen del lenguaje ensamblador. Por ejemplo quizás en una cierta arquitectura la instrucción de borrado de memoria (Memory Clear, en inglés) corresponda al código 010. Pronto surgieron programas que leían, siguiendo el ejemplo, MC, y lo sustituían por 010.

4.6. LENGUAJE ENSAMBLADOR

El código máquina tenía dos grandes inconvenientes para los programadores:

- El primero es que se trata de unas instrucciones difíciles de recordar ya que no guardan relación con la operación que se está realizando.

- El segundo inconveniente es que puede y de hecho hay diferencias entre las instrucciones de un procesador a otro.

Todo esto ha llevado a "poner nombre" a las instrucciones de código máquina de manera que a una secuencia concreta de bits que realiza una operación se le pone un nombre sencillo que identifique la operación. Esta traducción a un lenguaje más sencillo para las personas resulta en una mayor comodidad para el programador, además el proceso de traducción inverso de lenguaje ensamblador a código máquina puede ser realizado por un sencillo programa.

4.7. LENGUAJES DE ALTO NIVEL

Sobre este lenguaje ensamblador inicial se fueron construyendo otros lenguajes de programación de más alto nivel; esto significa que ocultan ciertos aspectos de manera que el programador no se ha de preocupar sobre si en la máquina que quiere que se ejecute el algoritmo el MC corresponde a la instrucción 101 o 010. Se produce, por tanto, una abstracción de datos, muy deseable para poder utilizar el trabajo de otros para avanzar un paso más en vez de tener que "reinventar la rueda", como se suele decir. Estos textos en los que se codifican los algoritmos son los códigos fuente; siguen las reglas sintácticas de un determinado lenguaje de programación. Existen numerosos lenguajes de programación, y se utiliza uno u otros según sus características se adecuen más o menos a la resolución de nuestro problema.

4.8. TRADUCTORES E INTÉRPRETES

Tras la escritura del algoritmo, un compilador o un intérprete (otros programas) transformarán el texto en código máquina que el procesador es capaz de ejecutar.

Toda esta abstracción permite resolver problemas alejados de sumar números binarios, como pueden ser la consulta de esta misma enciclopedia o jugar a un videojuego en 3D.

5. METODOLOGÍA

El desarrollo de este proyecto se enmarcó por las siguientes fases, compuestas por una serie de actividades. Las fases se muestran a continuación:

5.1. Fase 1. Definición del área problemática

El inicio de este trabajo de grado parte de la labor docente, el autor de este proyecto ha desempeñado este rol desde sus inicios como estudiante y de manera oficial en el colegio franciscano Agustín Gemelli. Una de las labores de mayor dificultad es enseñar algoritmia teniendo en cuenta el punto de vista del estudiante, es decir, enseñar fundamentos de programación teniendo en cuenta la naturaleza del alumno (estudiantes de tecnología o ingeniería, estudiantes de secundaria o simplemente personas interesadas en las ciencias de la computación). Teniendo en cuenta lo anterior surge la idea de generar una obra escrita que sirva de guía para todos los interesados en esta área sin hacer diferencia en sus intereses.

Seguidamente se analizó una serie de problemáticas buscando la más afín con los intereses de la facultad de ingeniería y, por ende, por el autor de la propuesta. Una vez definido el tema a tratar, se verificó el estado del arte a través de consulta bibliográfica para conocer qué se ha trabajado con respecto a la temática y determinar su viabilidad. Seguidamente se inició la búsqueda de los antecedentes que acompañan el tema definido, para conocer el material bibliográfico existente. Para este caso de estudio se analizaron todos los textos escritos que contenían como objetivo principal la enseñanza de los fundamentos de programación. Adicionalmente se estudió cada material encontrado de manera minuciosa y crítica para saber la forma de abordar y tratar el tema.

Es de anotar que la propuesta inicial apunta sólo al desarrollo de un libro sobre fundamentos de programación; pero por el estudio del tema surge la adhesión de herramientas de desarrollo de algoritmos que permitirán el desarrollo práctico de los temas del libro en un entorno de interacción con un sistema de cómputo, por lo anterior surgen las herramientas llamadas DFD de la universidad del Magdalena y el ANALIZADOR DE ALGORITMOS del autor de esta propuesta desarrollado como reto intelectual.

5.2. Fase 2. Anteproyecto

Esta fase plasmó los alcances e intereses descubiertos previamente, definiendo la metodología a seguir, delimitándolo con los objetivos a alcanzar y sustentando la importancia del proyecto mediante la justificación. A partir de aquí el comité de trabajos de grado emitió un concepto favorable y viable de la propuesta para la continuación y desarrollo.

5.3. Fase 3. Análisis y diseño de la estructura del libro

De la mano del presidente, los asesores y autor de la propuesta, se definió cuáles van a ser los contenidos y objetivos que se propone el libro alcanzar, con el fin de lograr un mayor público objetivo que pueda aprovechar los conocimientos plasmados. Con la participación de un diseñador visual se estructuró el cuerpo del libro, para mantener la atención del lector, empleando una serie de convenciones e iconografía para tener en consideración. El asesor temático evaluó cuál es la mejor estructura para el manejo de los algoritmos conservando y aprovechando los estudios de otras entidades.

5.4. Fase 4. Producción del libro

El modelo para la producción y validación que se utilizó para este material escrito se denomina espiral¹ (es una técnica utilizada en el desarrollo de prototipos de software, que realiza validación y verificaciones constantes que se amoldo fácilmente al desarrollo de una obra escrita) por que cada vez que se genera material se revisa teniendo en cuenta siempre las entregas anteriores y correcciones indicadas, realizando siempre una rectificación de todo el material desde el principio cada vez.

En esta etapa apareció la figura del asesor pedagógico que evaluó la forma cómo se está proporcionando el conocimiento a los estudiantes, combinando la fundamentación técnica y la pedagogía utilizada para hacer más fácil la comprensión del texto. Una vez terminado el libro se procedió a una última revisión

¹ El modelo espiral para la ingeniería de software ha sido desarrollado para cubrir las mejores características tanto del ciclo de vida clásico, como de la creación de prototipos, añadiendo al mismo tiempo un nuevo elemento: el análisis de riesgo. El modelo representado mediante la espiral de la figura 2.4, define cuatro actividades principales:

1. Planificación:
2. determinación de objetivos, alternativas y restricciones. Análisis de riesgo
3. : análisis de alternativas e identificación/resolución de riesgos. Ingeniería
4. : desarrollo del producto del "siguiente nivel", Evaluación del cliente

: Valorización de los resultados de la ingeniería Durante la primera vuelta alrededor de la espiral se definen los objetivos, las alternativas y las restricciones, y se analizan e identifican los riesgos. Si el análisis de riesgo indica que hay una incertidumbre en los requisitos, se puede usar la creación de prototipos en el cuadrante de ingeniería para dar asistencia tanto al encargado de desarrollo como al cliente. El cliente evalúa el trabajo de ingeniería (cuadrante de evaluación de cliente) y sugiere modificaciones. Sobre la base de los comentarios del cliente se produce la siguiente fase de planificación y de análisis de riesgo. En cada bucle alrededor de la espiral, la culminación del análisis de riesgo resulta en una decisión de "seguir o no seguir".

Con cada iteración alrededor de la espiral (comenzando en el centro y siguiendo hacia el exterior), se construyen sucesivas versiones del software, cada vez más completa y, al final, al propio sistema operacional. El paradigma del modelo en espiral para la ingeniería de software es actualmente el enfoque más realista para el desarrollo de software y de sistemas a gran escala. Utiliza un enfoque evolutivo para la ingeniería de software, permitiendo al desarrollador y al cliente entender y reaccionar a los riesgos en cada nivel evolutivo. Utiliza la creación de prototipos como un mecanismo de reducción de riesgo, pero, lo que es más importante permite a quien lo desarrolla aplicar el enfoque de creación de prototipos en cualquier etapa de la evolución de prototipos.

completa del material por parte de todos los integrantes del proyecto de manera individual y conjunta.

5.5. Fase 5. Adhesión de herramientas para el desarrollo de algoritmos

En esta fase se realizó una búsqueda de herramientas que permitieran analizar, validar y ejecutar algoritmos. El grupo SMART de la universidad del Magdalena apoyó el desarrollo de este proyecto, permitiendo usar su herramienta el DFD para el desarrollo de algoritmos en diagrama de flujo. También se cuenta con la herramienta ANALIZADOR DE ALGORITMOS que facilita el desarrollo de algoritmos en pseudocódigo. Ambas herramientas se complementan y permiten una primera etapa de interacción con el computador, permitiendo al lector un nuevo ambiente de aprendizaje.

5.6. Fase 6. Informe final

Esta última fase recopila todas experiencias que harán parte de todo el proceso en la generación, desarrollo, validación, corrección y publicación (impresión de 3 ejemplares) del libro fundamentos de programación y cd de herramientas que lo acompaña.

Este informe será evaluado por el comité de trabajos de grado, partiendo de lo propuesto sobre el presente documento, con el fin de verificar cada una de las fases.

6. RESULTADOS

La generación de este proyecto de grado proporcionó una amplia gama de elementos, que apoyan el proceso de aprendizaje en lo que corresponde a los fundamentos de programación.

Los resultados tangibles se presentan en una obra escrita para el acompañamiento en la asignatura fundamentos de programación, o como mecanismo autodidacta para que con una introducción previa, el lector de la misma, inicie a su propio ritmo e interés el recorrido hacia la programación de computadoras personales.

A continuación se listan los contenidos temáticos de cada uno de los capítulos y la introducción de cada uno:

Capitulo 1. Introducción a las Computadoras, y lenguajes de programación

- 1.1 Introducción.
- 1.2 Historia.
- 1.3 Hardware.
- 1.4 Software.
- 1.5. Programación.

INTRODUCCIÓN

Una computadora u ordenador es un dispositivo electrónico desarrollado para ejecutar un conjunto de instrucciones, facilitar el manejo de la información y procesar datos a grandes velocidades.

El avance de la electrónica hizo posible desarrollar este tipo de maquina, que es utilizada por toda la sociedad en innumerables usos. Son herramientas esenciales prácticamente en todos los campos de investigación y en tecnología aplicada. Pero es para destacar que aun faltan muchos usos por dársele y sacar su máximo provecho.

El objetivo de este capitulo pretende ambientar al lector sobre conceptos concisos acerca de la computación, abordando ítems de su evolución desde dos puntos de vista, a nivel físico o también llamado Hardware (componentes de un ordenador, tipos de ordenadores, etc.) y en la capa lógica o software (sistemas operativos, herramientas aplicadas, lenguajes de programación) encargado de administrar la primera. Ya que es imposible hablar de una sin tenerse en cuenta la otra, y los avances en ambas ramas repercuten directamente en la otra. Además de proporcionar unos conceptos claros para que el lector pueda afianzar sus conocimientos y generar unas sólidas bases.

Capitulo 2. Ejercicios de Lógica

2.1 Introducción.

2.2. Técnicas para solucionar problemas.

2.3. Ejercicios de lógica.

INTRODUCCIÓN

El interés de este capitulo es proporcionar un conjunto de elementos que permitan afrontar cualquier tipo de problema, de una manera más eficaz facilitando el ahorro de energía y tiempo. También se enfoca de una forma didáctica el uso de la lógica, en la manera de pasar un tiempo resolviendo unos ejercicios y comprender la importancia de la constante ejercitación mental.

El primer numeral presenta una serie de trucos que deben tenerse en cuenta en el momento de enfrentar y dar solución a diferentes puntos problemáticos.

Capitulo 3. Fundamentos

3.1 Introducción.

3.2. Algoritmo.

3.3. Tipos de datos.

3.4. Variables.

3.5. Constantes.

3.6. Representación de Expresiones aritméticas.

INTRODUCCIÓN

El propósito de este capítulo es proporcionar al lector los conceptos básicos de fundamentos de programación a partir de una primera técnica denominada algoritmo. De una manera progresiva se irán conociendo los elementos de la algoritmia hasta desarrollar programas básicos computacionales.

Capitulo 4. Algorimia

4.1 Introducción.

4.2. Representación de los algoritmos.

4.3. Operadores.

INTRODUCCIÓN

Los algoritmos son los primeros pasos hacia el mundo de la programación de computadores. La importancia de su estudio, permite mejorar la capacidad de abstracción que desarrolla también las matemáticas y además facilita la aplicación de los conocimientos básicos de programación, a cualquier campo de las ciencias

de la computación de una manera transparente, es decir con las bases que se proporcionan aquí el lector podrá estudiar mas lenguajes de programación de ultima generación, aprehender sistemas de desarrollo multimedia, trabajar en herramientas de simulación y aplicaciones matemáticas.

Es por lo anterior, que los algoritmos son el campo más importante que todo estudiante de computación o áreas afines debe conocer para garantizar desde un referente lógico.

Capitulo 5. Funciones establecidas, Funciones del sistema

5.1. Introducción.

5.2. Funciones.

5.2.1. funciones matemáticas.

5.2.2. funciones de cadena.

INTRODUCCIÓN

Los elementos establecidos son componentes de uso interno por un lenguaje de programación que realiza una operación especial. En otras palabras; si se observa con detenimiento los operadores aritméticos, cada uno de ellos realiza tareas específicas, por ejemplo: el operador + (mas) realiza la suma entre dos variables o constantes numéricas de cualquier tipo. El operador – (menos) efectúa la resta entre dos elementos. Partiendo de ese concepto, existen una serie de medios para hacer tareas específicas, como: la raíz cuadrada, el seno, el coseno, la secante, entre otros. Este tipo de medios llamados funciones simplifican las tareas que va a realizar un algoritmo. Una función es un algoritmo que realiza una operación específica, y puede ser invocado cualquier número de veces.

Capitulo 6. Control de Flujo I. Estructuras condicionales

- 6.1. Introducción.
- 6.2. Control de flujo toma de decisiones.
- 6.3. Estructuras condicionales.

INTRODUCCIÓN

Todos los algoritmos desarrollados hasta el momento se ejecutan de manera secuencial, es decir se ejecutan desde el inicio hasta el fin sin modificaciones y siempre van a desarrollar el mismo proceso con diferentes valores.

Las estructuras condicionales permiten la múltiple ejecución de instrucciones según los contenidos de las variables o datos que ingrese el usuario, es decir, según los datos suministre el usuario pueden ejecutarse una serie u otra de instrucciones.

Capitulo 7. Control de Flujo II. Estructuras Repetitivas

- 7.1. Introducción.
- 7.2. Control de flujo toma de decisiones.
- 7.3. Estructuras condicionales.

INTRODUCCIÓN

Con el capítulo anterior se dio un poco de dinamismo a los algoritmos, pero existen problemas que requieren de la ejecución de instrucciones ya desarrolladas. Con las estructuras repetitivas o ciclos, se da la primera posibilidad de reutilizar código ya generado y usarlo el número de veces que se necesite.

Capitulo 8. La prueba de Escritorio

8.1. Introducción.

8.2. Reglas para usar una prueba de escritorio.

INTRODUCCIÓN

La prueba de escritorio permite conocer el funcionamiento de un algoritmo de manera completa. La prueba de escritorio hace un seguimiento instrucción por instrucción sobre cada uno de los elementos dentro de un algoritmo. Es una herramienta de análisis y validación de algoritmos muy fácil de usar y de aplicar a cada uno de los algoritmos, e incluso a codificación sobre un lenguaje de programación.

Capitulo 9. Arreglos

9.1. Introducción.

9.2. Arreglos unidimensionales o vectores.

9.3. Arreglos bidimensionales o matrices.

INTRODUCCIÓN

Dada la complejidad y características de algunos problemas, se proporcionan nuevos tipos de datos a partir de los existentes vistos en capítulos iniciales. Los problemas computacionales requieren de espacios en memoria más amplios para almacenar datos de gran similitud.

Los arreglos unidimensionales y bidimensionales, también llamados vectores y matrices respectivamente, proporcionan mayor capacidad en el almacenamiento de datos comparado con las variables usadas a lo largo del contenido.

Capitulo 10. Funciones

10.1. Introducción.

10.2. Definición.

10.3. Tipos de funciones.

INTRODUCCIÓN

En el capítulo cinco se conoció una serie de elementos fijos por un lenguaje de programación para desarrollar tareas específicas como: el cálculo de la raíz cuadrada, el seno, el coseno, la fecha del sistema, entre otras, denominadas funciones establecidas. Estas permiten acelerar el proceso de desarrollo de programas de cómputo, siendo usadas tantas veces como sea necesario. En este capítulo se va a tratar el desarrollo de otras funciones definidas por el programador, adicionales y complementarias a las existentes.

Otro resultado de forma indirecta, son las herramientas computacionales Analizador de algoritmos y DFD que acompaña la obra y permiten que el lector verifique sus avances y resultados de los algoritmos realizados en Pseudocódigo y diagrama de flujo respectivamente.

Un impacto positivo que apoya el éxito del presente proyecto, fue el uso que se le dio en la materia fundamentos de programación, orientada por el Ing Jaime Uldarico Cardona, en el programa de tecnología de sistemas, el docente Ing. Germán William Londoño en el programa ingeniería de sistemas y telecomunicaciones y el docente Tec. Carlos Andrés Zapata Ospina, en los laboratorios de taller de programación de ambos programas, pertenecientes a la Universidad de Manizales. Además se suma el proceso de validación que se realizó en el programa de ingeniería telemática de la universidad Católica de

Manizales con el docente Diego Fernando Nuñez y con el apoyo de Jaime Alberto Sepulveda.

El uso de estas herramientas inició la discusión académica, acerca del uso de algoritmos estructurados o el paso completo a programación orientada a objetos para las etapas iniciales de fundamentación, ya que la programación orientada a objetos se basa en clases para modelar los elementos de la naturaleza pero las acciones o métodos son netamente estructurados.

7. CONCLUSIONES

1. El desarrollo de la obra técnica escrita denominada fundamentos de programación, guía de autoenseñanza permitió reflejar una serie de elementos que deben ser considerados en el momento de preparar futuros profesionales. De forma reiterativa, no se tienen hábitos de lectura y mucho menos de escritura, es difícil plasmar ideas y hacerlas comprensibles ante un público.
2. Basado en la experiencia tenida, el desarrollo de un libro debe planificarse muy bien y determinar metas fijas y a cortos plazos, ya que el mundo de la tecnología se encuentra en constante evolución por tanto, una obra debe pensarse, elaborarse y publicarse en poco tiempo con el fin de que mantenga en un buen periodo de vigencia.
3. La obra generada facilita de manera completa el proceso de aprendizaje para todos los esquemas como son:
 - Clases magistrales como libro guía.
 - Esquema de créditos como trabajo independiente.
 - Modelo de aprendizaje a distancia como libro guía de consultoría y de ejercicios.
4. La generación de una obra técnica sobre fundamentos de programación y su conjunto de herramientas, genera muchas expectativas por la importancia y posibles usos. En la visita de pares académicos se mencionó este proyecto siendo de relevancia para facultad, por el esquema creativo que implica y la posición dentro del medio por la producción escrita.

8. RECOMENDACIONES

1. Debería considerarse el desarrollo de un sistema de información para el manejo y atención de las propuestas y proyectos de grado. Un sistema donde se pueda registrar todo el proceso desde que inicia el anteproyecto, con sus respectivas revisiones y correcciones hasta el documento final. Además del conocer todo los pasos que han llevado a cabo durante el ciclo de vida de la propuesta, permite aplicar correctivos y visualizar de forma más global la magnitud y alcances del proyecto. También permite más interactividad en el manejo de los procesos y cambia de manera radical el esquema meramente receptivo que se tiene.
2. El proceso de asimilación, manejo y dominio a las temáticas que se orientan en la facultad es un buen punto de partida para el análisis de próximas obras técnicas, que apoyen y resalten el proceso académico que se lleva acabo dentro de las aulas. Literatura sobre el lenguaje de programación Java existe mucha, pero ¿qué tal combinarlo con los contenidos de las asignaturas y presentar una obra completa y dirigida a estudiantes de ingeniería donde se enmarque el modelo de núcleos problémicos?
3. Debería darse una discusión académica por el uso de este proyecto, aun más con la creación de la tecnología semi - presencial a la cual apunta la facultad. Además en el esquema de flexibilidad curricular se diseño como materia opcional fundamentos de programación, siendo un tema de gran importancia para los siguientes flujos académicos.

FUENTES BIBLIOGRÁFICAS

BOBADILLA, Jesus. Active Server Pages 3, creación de aplicaciones Web a través de ejemplos. México. Alfaomega grupo editor.

BRASSARD y BRATLEY, G, P. Fundamentos de algoritmia. Madrid, España. Prentice Hall, 1999.

CEBALLOS, Francisco Javier. El lenguaje de programación Visual Basic .Net. México, México. Alfaomega, 2003.

CEBALLOS, Francisco Javier. Enciclopedia de Microsoft Visual Basic 6.0 México, México. Alfaomega, 1999.

DEILTEL Y DEITEL, H.M y P.J.. Como programar en Java. Naucalpan de Juárez. México. Prentice Hall, 1998.

DOHERTY, MANNING, Donal, Michelle. Aprendiendo Borland Jbuilder 3 en 21 días. 53370 Naucalpan de Juárez. México. 2000. Prentice Hall Hispoamericana.

GOTTFIEND, Byron, Programación en C. Madrid, España. McGraw Hill, 1997.

GUITIÉRREZ, Abraham. Php 5 a través de ejemplos. Naucalpan de Juárez. México. Alfaomega grupo editor.

JOYANES AGUILAR, Luis. Turbo/Borland Pascal 7, iniciación y referencia. Madrid, España. McGraw Hill, 1997.

----. Fundamentos de Programación. Madrid, España. McGraw Hill, 1997.

LIPSCHUTZ, Seymour. Matemáticas para computación. México, México. McGraw Hill, 1992.

MCKELVY, Mike. Visual Basic 5. España. España. 1997. Prentice Hall Hispoamericana.

SAND, Paul A.. Pascal para macintosh. Naucalpan de Juárez. México. McGraw Hill, 1987.

TAMAYO ALZATE, Alonso. Algoritmos estructurados. Manizales, Colombia. Centro de publicaciones Universidad Nacional de Colombia, seccional Manizales, 1991.