



REPUBLICA BOLIVARIANA DE
VENEZUELA UNIVERSIDAD MONTEÁVILA
COMITÉ DE ESTUDIOS DE POSTGRADO



**ESPECIALIZACIÓN EN PLANIFICACIÓN,
DESARROLLO Y GESTIÓN DE
PROYECTOS**

Bases Funcionales Para el Diseño de un
Curso Propedéutico para la Enseñanza de Algoritmos en el Colegio Universitario
"Francisco de Miranda"
(CPROALGORIT)

**Trabajo Especial de Grado, para optar al Título de Especialista en
Planificación, Desarrollo y Gestión de Proyectos, presentado por:**
Yamarte Montes, Yrama Del Valle, CI V-6.284.459

Asesorado por:
Sarache Oliveros, Xarifa Margarita
Vizcaya Carrillo, Fernando

Caracas, marzo de 2017

**REPÚBLICA BOLIVARIANA DE VENEZUELA
UNIVERSIDAD MONTEÁVILA
COMITÉ DE ESTUDIOS DE POSTGRADO**

**ESPECIALIZACIÓN EN PLANIFICACIÓN,
DESARROLLO Y GESTIÓN DE PROYECTOS**

Bases Funcionales Para el Diseño de un
Curso Propedéutico para la Enseñanza de Algoritmos en el Colegio Universitario
"Francisco de Miranda"
(CPROALGORIT)

**Trabajo Especial de Grado, para optar al Título de Especialista en
Planificación, Desarrollo y Gestión de Proyectos, presentado por:**
Yamarte Montes, Yrama Del Valle, CI V-6.284.459

Asesorado por:
Sarache Oliveros, Xarifa Margarita
Vizcaya Carrillo, Fernando

Caracas, marzo de 2017

Señores:

Universidad Monteávila
Comité de Estudios de Postgrado
Especialización en Planificación, Desarrollo y Gestión de Proyectos

Atención: Profesora Geraldine Cardozo

Referencia: **Aprobación de Asesoría**

Por medio de la presente le informo que hemos revisado el borrador final del Trabajo Especial de Grado de (los) Ciudadano (s): **Yamarte Montes, Yrama Del Valle**, titular de la Cédula de Identidad N° **96.284.459**; cuyo título tentativo es: **“Bases Funcionales Para el Diseño de un Curso Propedéutico para la Enseñanza de Algoritmos en el Colegio Universitario ‘Francisco de Miranda’ (CPROALGORIT)”**, la cual cumple con los requisitos vigentes de esta casa de estudio para asignarles jurado y su respectiva presentación.

A los 15 días del mes de marzo del 2017

Sarache Oliveros, Xarifa Margarita
Asesor de Seminario de Trabajo Especial de Grado III

Vizcaya Carrillo, Fernando
Asesor Académico

CARTA DE AUTORIZACIÓN DE LA INSTITUCIÓN

CARTA DE AUTORIZACIÓN DE LA INSTITUCIÓN

Especialización en Planeación, Desarrollo y Gestión de Proyectos
Comité de Estudios de Postgrado
Universidad Monteávila
Presente.-

Nos dirigimos a ustedes para informarles que hemos autorizado a **Yamarte Montes, Yrama Del Valle**, a hacer uso de la información proveniente de nuestra institución Colegio Universitario "Francisco de Miranda", para documentar y soportar los elementos de los distintos análisis estrictamente académicos que conllevarán a la realización del Trabajo Especial de Grado "**Bases Funcionales Para el Diseño de un Curso Propedéutico para la Enseñanza de Algoritmos en el Colegio Universitario 'Francisco de Miranda' (CPROALGORIT)**"; como requisito para optar al grado de "**Especialista en Planeación, Desarrollo y Gestión de Proyectos**", exigido por Comité de Estudios de Postgrado de la Universidad Monteávila.

En la ciudad de Caracas, a los 15 días del mes de marzo de 2017.


Marisela Ramírez de Eustache
C.I. V-5.426.174

Sub-Directora Académica





REPÚBLICA BOLIVARIANA DE VENEZUELA
UNIVERSIDAD MONTEÁVILA
COMITÉ DE ESTUDIOS DE POSTGRADO
ESPECIALIZACIÓN EN PLANIFICACIÓN,
DESARROLLO Y GESTIÓN DE PROYECTOS



TRABAJO ESPECIAL DE GRADO

**Bases Funcionales Para el Diseño de un
Curso Propedéutico para la Enseñanza de Algoritmos en el Colegio
Universitario "Francisco de Miranda"
(CPROALGORIT)**

Autora: Yamarte Montes, Yrama Del Valle
Asesores: Sarache Oliveros, Xarifa Margarita,
Vizcaya Carrillo, Fernando
Año: 2017

RESUMEN

La creación de algoritmos exige la adquisición de un conjunto de vocabularios, terminología y conceptos propios de la Algorítmica, así como su adecuado manejo. Es un aprendizaje que comprende una serie de dificultades que el estudiante debe superar para lograr las habilidades y destrezas requeridas para formarse competentemente en la programación de computadoras. La disminución de esas dificultades se puede lograr mediante el fortalecimiento de las competencias básicas del estudiante y la ayuda necesaria para que el manejo de las complejidades asociadas a la creación de algoritmos; a través de la realización de un curso propedéutico para tal fin. Esta investigación trata sobre la definición de las bases funcionales que se considera deben fundamentar el diseño de un curso propedéutico para enseñar algoritmos. Establece las bases pedagógicas, temáticas y didácticas que lo deben sustentar en el marco de ejecución del Programa Nacional de Informática en el Colegio Universitario "Francisco de Miranda". El curso propedéutico para enseñar algoritmos es preparatorio para la unidad curricular "Algorítmica y Programación". Los docentes pertenecientes a la referida especialidad tienen la formación tecnológica y experiencia en la enseñanza de la programación de computadoras. Finalmente, se propone la formulación de un proyecto para ser presentado a las autoridades y obtener la aprobación para su desarrollo e implementación.

Línea de Trabajo: Generación de Proyectos

Palabras clave: Enseñanza de algoritmos, teorías pedagógicas, proyectos.

Nomenclatura UNESCO: (110599) Metodología-Otras (Proyectos), (120302) Lenguajes Algorítmicos, (330416) Diseño Lógico, (531204) Educación y (580203) Desarrollo de Asignaturas.

DEDICATORIA

A la memoria de DJYM, mi padre.

A la memoria de RDYY, mi sobrino.

Ha sido muy difícil lidiar con sus pérdidas.

AGRADECIMIENTOS

A mi madre, a mis hermanos, a mis sobrinos y a mi familia por ser mi origen.

A mis afectos más cercanos y
a mis amigos por comprender mi ausencia de estos últimos tiempos.

Al Profesor Fernando Vizcaya por su orientación, paciencia y comprensión.

A la Profesora Xarifa Sarache por animarme para el logro de este objetivo.

A la Profesora Marcella Prince por su oportuno consejo.

A la Profesora Nancy Meléndez por sus observaciones.

A los compañeros de la especialización por su invaluable apoyo.

Al Universo por su sabiduría y las infinitas gamas del conocimiento.

INDICE GENERAL

	PÁG.
INDICE DE FIGURAS.....	xi
INDICE DE TABLAS.....	xii
LISTA DE ACRÓNIMOS Y SIGLAS.....	xiii
INTRODUCCIÓN.....	1
CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA	
1.1 Área Temática.....	3
1.2 Planteamiento del Problema.....	4
1.3 Pregunta de la Investigación.....	5
1.4 Objetivos de la Investigación.....	6
1.4.1 Objetivo General.....	6
1.4.2 Objetivos Específicos.....	6
1.5 Justificación e Importancia de la Investigación.....	6
1.6 Limitaciones y Alcance de la Investigación.....	7
CAPÍTULO II: MARCO TEÓRICO	
2.1 Antecedentes de la Investigación.....	9
2.2 Referencias Teóricas.....	16
2.2.1 Sobre los Algoritmos.....	16
2.2.2 Sobre la Educación.....	47
2.2.3 Enseñanza de Algoritmos.....	49
2.3 Definición de Términos Básicos.....	50
2.3.1 Relacionados con Algoritmos.....	50
2.3.2 Relacionados con Educación.....	52
2.4 Marco Conceptual de Proyectos.....	55
2.4.1 Aspectos Generales.....	55
2.4.2 Metodologías de Proyectos.....	58
2.5 Marco Legal.....	60

CAPÍTULO III: MARCO METODOLÓGICO

3.1 Tipo de Investigación.....	61
3.2 Diseño de la Investigación.....	61
3.3 Unidad de Análisis.....	62
3.4 Técnicas y Herramientas de Recolección de Datos.....	62
3.5 Ámbito del Dominio de Conocimientos.....	63
3.6 Organización de la Investigación.....	64
3.7 Aspectos Administrativos.....	65
3.7.1 Recursos: humanos, materiales y financieros.....	65
3.7.2 Cronograma de Actividades.....	66

CAPÍTULO IV: MARCO ORGANIZACIONAL

4.1 Colegio Universitario “Francisco de Miranda”.....	67
4.1.1 Reseña Institucional.....	67
4.1.2 Visión.....	67
4.1.3 Misión.....	68
4.1.4 Valores.....	68
4.2 Programa Nacional de Formación Informática.....	68
4.2.1 Programas Nacionales de Formación.....	68
4.2.2 Programa Nacional de Formación en Informática.....	70

CAPÍTULO V: RESULTADOS

5.1 Análisis de la Situación Problemática.....	73
5.1.1 Origen del Proyecto.....	73
5.1.2 Identificación y Análisis de Involucrados.....	73
5.1.3 Análisis de Problemas: Árbol del Problema.....	76
5.1.4 Análisis de Objetivos: Árbol de Objetivos.....	78
5.1.5 Análisis de Alternativas: Selección de Alternativa.....	80
5.2 Definición de Bases Pedagógicas.....	81
5.2.1 Punto de Partida: El Constructivismo.....	81
5.2.2 Aprender a Aprender.....	82

5.2.3 Motivación.....	82
5.2.4 Competencias.....	83
5.2.5 Pensamiento Computacional.....	85
5.3 Definición de Bases Temáticas.....	85
5.4 Definición de Bases Didácticas.....	87
5.4.1 Principios Didácticos.....	87
5.4.2 Estrategias Metodológicas.....	88
5.5 Formulación del Programa del Curso Propedéutico.....	89
5.6 Plan Preliminar de Desarrollo e Implementación.....	91
5.6.1 Premisas.....	91
5.6.2 Estructura Desagregada de Trabajo.....	92
CAPÍTULO VI: LECCIONES APRENDIDAS.....	93
CAPÍTULO VII: CONCLUSIONES Y RECOMENDACIONES.....	94
CAPÍTULO VIII: REFERENCIAS BIBLIOGRÁFICAS.....	97
CAPÍTULO IX: ANEXOS	
A. Origen de la palabra “Algoritmo”.....	103
B. PNFI: Malla Curricular.....	105
C. Unidad Curricular “Algoritmo y Programación” del PNFI.....	107
D. Insumos Elaboración Pruebas Diagnóstico.....	111
E. Elementos Claves para el Diseño del Curso Propedéutico Para Enseñar Algoritmos.....	113

INDICE DE FIGURAS

FIGURA		Pág.
1	Ciclo Genérico del Desarrollo de un Programa.....	17
2	Ciclo Resolución de Problemas Por Computadora.....	18
3	Esquema General de un Algoritmo.....	21
4	Simbología de los Diagramas de Flujo.....	37
5	Estructura Genérica del Ciclo de Vida del Proyecto y, Costo y Nivel de Dotación de Personal.....	56
6	Ciclo de Vida Genérico y Fases de un Proyecto de Ingeniería.....	57
7	Interacción de los Grupos de Procesos de un Proyecto.....	58
8	Sistema de Marco Lógico.....	59
9	Herramientas Claves del Enfoque de Marco Lógico.....	60
10	Dominios de conocimientos del ámbito de la investigación.....	63
11	Cronograma Estimado de Actividades.....	66
12	“Árbol del Problema” Resultante.....	78
13	“Árbol de Objetivos” Resultante.....	79
14	Relación Desarrollo Etapas Elaboración de Algoritmo y Competencias.....	84
15	Clasificación de Algoritmos.....	86
16	EDT del Proyecto CPROALGORIT.....	92

INDICE DE TABLAS

TABLA	Pág.
1 Recursos Humanos.....	65
2 Recursos Materiales.....	65
3 Presupuesto Estimado.....	65
4 Pasos para Identificar y Analizar a los Involucrados.....	74
5 Resultado de la Identificación de Involucrados.....	75
6 Resultado del Análisis de Involucrados.....	75
7 Pasos para la Definición del Problema Central.....	77
8 Pasos para el Desarrollo del Árbol del Problema.....	77
9 Pasos para el Análisis de Objetivos.....	79
10 Pasos para el Análisis de Alternativas de Solución.....	80
11 Lista de Recursos para el Curso Propedéutico.....	89

LISTA DE ACRÓNIMOS Y SIGLAS

CPROALGORIT: Curso Propedéutico de Algoritmos.

CUFM: Colegio Universitario “Francisco de Miranda”.

EDT: Estructura Desagregada de Trabajo.

EML: Enfoque del Marco Lógico

MML: Metodología del Marco Lógico

PMBOK: Project Management Body Of Knowlegde.

PMI: Project Management Institute.

PNF: Programa Nacional de Formación Universitaria.

PNFI: Programa Nacional de Formación en Informática.

UNESCO: United Nations Educational, Scientific and Cultural Organization.

INTRODUCCIÓN

La creación de algoritmos exige la adquisición de un conjunto de vocabularios, terminología y conceptos propios de la Algorítmica, así como su adecuado manejo. Es un aprendizaje que comprende una serie de dificultades que el estudiante debe superar para lograr las habilidades y destrezas requeridas para formarse competentemente en la programación de computadoras.

La disminución de esas dificultades se puede lograr mediante el fortalecimiento de las competencias básicas del estudiante y la ayuda necesaria para que el manejo de las complejidades asociadas a la creación de algoritmos; a través de la realización de un curso propedéutico para tal fin.

Esta investigación trata sobre la definición de las bases funcionales que se considera deben fundamentar el diseño de un curso propedéutico para enseñar algoritmos. El documento resultante de la investigación, se estructuró en ocho capítulos descritos a continuación:

Capítulo I, se define el área temática, se realiza la descripción del **Planteamiento del Problema** y la pregunta de investigación. Se presentan los objetivos del trabajo a través de los cuales se llegará a una solución; además de la justificación e importancia de la investigación planteada, así como la definición de limitaciones y alcance.

El **Capítulo II** es el **Marco Teórico**, consiste en la exposición de los aspectos que constituye el marco teórico de la investigación.

El **Capítulo III** es el **Marco Metodológico** de la investigación, aquí se caracteriza el enfoque y tipo de investigación realizada; se definen el diseño, la unidad de análisis, así como las técnicas y herramientas usadas para la

recolección de información. También se muestra la organización dada a la investigación y aspectos de tipo administrativo.

Capítulo IV, se realiza el **Marco Contextual**, aquí se describen los principales aspectos organizacionales de la Institución Universitaria y del Programa Nacional de Formación que conforma el contexto del problema planteado en la investigación.

En el **Capítulo V** se muestran los **Resultados**, producto del análisis efectuado y las actividades realizadas para lograr cada uno de los objetivos definidos en este trabajo de investigación.

Capítulo VI, contiene las **Conclusiones, Recomendaciones y Lecciones Aprendidas** obtenidas con la realización del trabajo de investigación.

Finalmente se presentan el **Capítulo VII** y **Capítulo VIII** que contienen, respectivamente, las **Referencias Bibliográficas** consultadas para la realización de esta investigación y los **Anexos**.

CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA

1.1 Área Temática

En la vida se realizan rutinas diariamente: La rutina que se sigue al levantarse en la mañana antes de salir a la calle; las acciones que se siguen para efectuar una llamada telefónica, los pasos a seguir para comprar un ticket para viajar en Metro, por mencionar algunas. En la Escuela, y a lo largo de la carrera estudiantil, en las diferentes asignaturas o disciplinas se enseñan procedimientos –o conjunto de pasos o acciones- para realizar actividades o resolver problemas, por ejemplo: en Biología, Física o Química los pasos a seguir para llevar a cabo un experimento; en Castellano cómo hacer para analizar una oración, o en Matemática los pasos a seguir para resolver una ecuación de segundo grado. Esas rutinas o secuencias de pasos que describen cómo realizar una tarea o actividad pueden ser denominadas **Algoritmos**.

Durante el transcurso de la vida académica y profesional, básicamente se hace uso de los **Algoritmos** que se han aprendido para realizar las actividades propias del área de desempeño, con la utilización del computador, el cual se encuentra muy presente en el mundo actual. Y justamente para dicha herramienta, son los **Algoritmos** los que constituyen la base de su funcionamiento. Un computador por sí mismo no es capaz de hacer tarea alguna, requiere que se le indique todas y cada uno de los pasos que va a realizar. Un programador es la persona quien se encarga de la creación de los diferentes programas o aplicaciones -*software*- que se ejecutarán en un computador, pero antes debe saber elaborar los algoritmos que, traducidos a programas, operarán en dicha máquina.

Una cuestión es ser usuario de un algoritmo y otra cosa, muy distinta, es crearlos. La elaboración de algoritmos exige la utilización de la capacidad de resolución de problemas, análisis lógico-matemático y el desarrollo del pensamiento abstracto. Para el desarrollo de las competencias inherentes a la programación de computadoras es importante tener una preparación

previa que permita al estudiante el entendimiento de los conceptos propios de los **Algoritmos**, así podrá aplicarlos con éxito y llevar a cabo una correcta elaboración de los mismos.

Por las razones antes expuestas el área temática que aborda este trabajo es la **Enseñanza y Aprendizaje de Algoritmos a estudiantes de informática**.

1.2 Planteamiento del Problema

La programación de computadoras está basada en la composición de secuencias de comandos o instrucciones básicas que le ordenan al sistema informático las acciones a ejecutar. Los lenguajes de programación de computadoras tienen sus bases conceptuales en la **Algorítmica**¹. Esta disciplina tiene como objeto de estudio los *algoritmos* que, en términos generales, se definen como una secuencia (finita) de instrucciones o pasos bien definidos y ordenados lógicamente para realizar una tarea o resolver un problema.

El aprendizaje de *algoritmos* exige la adquisición y un adecuado manejo de un conjunto de vocabulario, terminología y conceptos propios de la *Algorítmica*: Acciones, objetos algorítmicos, variables, identificadores, constantes, composición y secuencia de instrucciones, entre otros, que van a permitir que el estudiante aprenda a escribir o elaborar los *algoritmos* que, posteriormente, traducidos en un lenguaje particular, serán convertidos en programas informáticos o de computación.

En general, una de las dificultades que enfrentan los docentes para la enseñanza de *algoritmos* es la carencia de habilidades, competencias y/o conocimientos básicos de análisis y razonamiento lógico-matemático que tienen los estudiantes que inician su carrera de informática o computación, lo cual -a la larga- se traduce en: (i) Dificultades para proseguir los estudios de las materias del eje de Programación, debido a que se abordan conceptos más complejos cuyas bases se encuentran en los conceptos básicos tratados en *algoritmos*; (ii) Una alta tasa de repitencia de dicha asignatura; (iii)

¹Hay autores que utilizan *Algoritmia* aunque en la DRAE se define así: "Ciencia del cálculo aritmético y algebraico; teoría de los números".

Rechazo por la Programación siendo esta una disciplina básica para su formación profesional; (iv) e incluso, también ocasiona casos de deserción estudiantil de la carrera. En particular, el Colegio Universitario “Francisco de Miranda” (**CUFM**) no escapa a la situación antes descrita, dado que en el Programa Nacional de Formación en Informática (**PNFI**) ocurren esas mismas dificultades con sus estudiantes.

1.3 Pregunta de la Investigación

Sobre la exposición anterior se formula la siguiente interrogante para la investigación:

¿Cuáles son las bases que deben fundamentar el diseño de un curso propedéutico para la enseñanza de algoritmos?

Del planteamiento a investigar surgen las siguientes interrogantes:

- ¿Cuáles son los actores y elementos que intervienen en el proceso de inicio de estudios de la Algorítmica y Programación? ¿Cuáles dificultades afronta los actores?
- ¿Cuáles son las bases que fundamentan el diseño de un curso propedéutico?
- ¿Cuáles son las habilidades y competencias básicas que el estudiante debe tener o desarrollar, o en su defecto propiciar el docente, para que afronte con éxito la creación de algoritmos? ¿El pensamiento lógico y analítico?
- ¿Cuáles son los conocimientos para desarrollar esas habilidades y competencias necesarias para la creación de algoritmos?
- ¿Cuáles estrategias de enseñanza favorecen la adquisición de los conocimientos para el desarrollo del pensamiento lógico y analítico necesarios para iniciarse en el estudio de los algoritmos?

1.4 Objetivos de la Investigación

1.4.1 General

Definir las bases funcionales para diseñar un curso propedéutico que permita la enseñanza de algoritmos, a los estudiantes del trayecto inicial del Programa Nacional de Formación Informática del Colegio Universitario “Francisco de Miranda”.

1.4.2 Específicos

- 1) Estudiar la situación problemática planteada mediante el análisis de los actores involucrados, sus causas y efectos, y alternativas de solución.
- 2) Determinar las bases pedagógicas que fundamentan el diseño de un curso propedéutico para la enseñanza de algoritmos.
- 3) Establecer la base temática que debe contemplar el diseño de un curso propedéutico para la enseñanza de algoritmos a través de la definición de los contenidos a cubrir (términos, conceptos y conocimientos fundamentales).
- 4) Definir las bases didácticas del diseño de un curso propedéutico para la enseñanza de algoritmos mediante la identificación de los elementos claves sobre algoritmos, estrategias metodológicas, recursos y actividades a emplear por el docente así como la especificación de las habilidades y competencias a desarrollar por el estudiante.
- 5) Elaborar la propuesta de un curso propedéutico para la enseñanza de algoritmos.
- 6) Proponer un plan para la implementación del curso propedéutico.

1.5 Justificación e Importancia de la Investigación

- El diseño un curso propedéutico para la enseñanza de algoritmos da la posibilidad de crear estrategias pedagógicas, desde un enfoque teórico-práctico, que proporcionen herramientas para el desarrollo de las competencias necesarias para el estudio de los algoritmos,

constituyéndose en un medio para la consolidación de la formación de los estudiantes de Ingeniería Informática.

- El curso propedéutico para la enseñanza de algoritmos permitirá estandarizar el contenido que se estudia en el Trayecto Inicial del Programa Nacional de Formación en Informática (PNFI) del Colegio Universitario “Francisco de Miranda” (CUFM).
- Esta investigación pretende ofrecer a los profesores del Trayecto Inicial del PNFI en el CUFM, una instancia que les permita desarrollar el pensamiento lógico y analítico necesario para estudiar los *algoritmos*, los cuales son la base fundamental de su formación como Ingenieros en Informática. Adicionalmente, tanto las estrategias de enseñanza y aprendizaje como los materiales del curso, podrán ser utilizados por los profesores, particularmente los encargados del Eje Curricular de Algorítmica y Programación.
- El curso propedéutico propuesto podrá ser utilizado o aplicado en otras instituciones de Venezuela, donde se imparte el Programa Nacional de Formación en Informática (PNFI).
- En esta investigación se aplica los principios de enfoque de proyectos, principalmente los correspondientes a la concepción y organización; dada su importancia que tiene para la planificación del proyecto y su posterior desarrollo.

1.6 Limitaciones y Alcance de la Investigación

- El acceso para la revisión de la información académica debe ser gestionado directamente con el departamento de Control de Estudio, previa autorización escrita de la Dirección y la División Académica del CUFM y la Coordinación del PNFI.
- La propuesta incluye los fundamentos básicos de la programación que comprenderá el curso propedéutico, y serán las correspondientes a las nociones básicas de algoritmos: Presentación, objetos algorítmicos, acciones algorítmicas básicas (lectura o entrada de datos, escritura o

salida de datos y asignación) y composición de acciones (secuencias, condicionales y los ciclos 'repetir' y 'mientras').

- El alcance de este trabajo de investigación incluye la formulación o diseño del curso propedéutico y la propuesta de un plan para su realización. Queda fuera del alcance la ejecución o implementación del curso propedéutico.

CAPÍTULO II: MARCO TEÓRICO

En este capítulo se presentan los antecedentes de la investigación más relevantes que fueron consultados durante la revisión documental. También se exponen conceptos y terminología propia de cada uno de los dominios de conocimiento que conforman el marco referencial del este trabajo de investigación. El capítulo finaliza las referencias de las leyes, decretos, reglamentos y disposiciones que constituyen el marco legal.

2.1 Antecedentes de la Investigación

De las fuentes consultadas sobre enseñanza y aprendizaje de algoritmos, se han elegido los siguientes escritos, por considerarlos importantes para el presente trabajo de investigación:

- Teague (2015) en “**Neo-Piagetian Theory and the Novice Programmer**”, trabajo para la obtención del grado en *Doctor of Philosophy (IT)*, cuyo propósito es entender por qué tantos estudiantes encuentran difícil aprender a programar computadoras. Efectúa un estudio que proporciona evidencia que indica que el desarrollo de habilidades de programación se puede describir utilizando el marco neo-piagetiano de desarrollo cognitivo. Muestra que el desarrollo de las habilidades de programación son *secuenciales y acumulativas*. Entender cómo los estudiantes (programadores novatos) desarrollan esas habilidades permite a los educadores ayudar de una mejor manera, con métodos más efectivos, a aquellos que tienen dificultades para aprender programación.

Este trabajo tiene como motivación el cuestionamiento sobre las dificultades en el aprendizaje de la programación que confrontan los estudiantes, lo cual puede ser extrapolado al aprendizaje de algoritmos ya que estos constituyen el paso previo a la elaboración de programas. Además, proporciona un

marco teórico para la elaboración del fundamento de los aspectos pedagógicos del trabajo a desarrollar durante la presente investigación.

Palabras clave: Teoría Neo-Piagetiana, desarrollo cognitivo, programadores novatos, aprendizaje de la programación, razonamiento abstracto, dificultades en el aprendizaje de la programación.

- Sáez y otros (2015), el trabajo presentado en el artículo “**El desarrollo de la habilidad: implementar algoritmos. Teoría para su operacionalización**”, tiene como objetivo definir una propuesta que fundamente teóricamente la habilidad “implementar algoritmos”. Determina, mediante un estudio del arte y el análisis de su tratamiento en la práctica, una propuesta para la concepción teórica y la estructura interna de esa habilidad; asimismo establece pautas generales para su sistematización. Lo cual debe trabajarse en tres niveles en correspondencia con los niveles de asimilación del conocimiento, y con las acciones y operaciones que conforman la estructura funcional de la antes referida habilidad. La propuesta de estructura permite guiar a los profesores en el diseño didáctico de tareas docentes asociadas al desarrollo de dicha habilidad en los estudiantes.

El aporte de esta investigación es el establecimiento de una concepción teórica sobre la estructura conceptual y operacional de la habilidad “implementar algoritmos”.

Palabras clave: Habilidad, teoría, operaciones, implementar, algoritmo.

- Selby (2014) en su Trabajo de Doctorado: “**How Can Teaching Of Programming Be Used To Enhance Computational Thinking Skills?**”, realiza un estudio que explora las formas en que la programación de computadoras puede emplearse como una herramienta para enseñar el pensamiento computacional y la resolución de problemas. Con base a los datos recolectados de profesores, académicos y profesionales,

seleccionados intencionalmente por su conocimiento de los temas de resolución de problemas, pensamiento computacional o la enseñanza de la programación, y su posterior análisis; se desarrolló una taxonomía de pensamiento computacional. Además, modelo las relaciones entre los procesos cognitivos, la pedagogía de la programación y los niveles percibidos de dificultad de las habilidades de pensamiento computacional.

Esta investigación aporta una visión que muestra la vinculación entre la programación de computadoras y la enseñanza, mostrando cómo aquella contribuye al desarrollo de habilidades para la resolución de problemas y el desarrollo del pensamiento computacional, que igualmente puede extrapolarse a la enseñanza de algoritmos.

Palabras clave: enseñanza de la programación, resolución de problemas, pensamiento computacional, habilidades del pensamiento computacional.

- Del Prado y Lamas (2014), su trabajo “**Alternativas para la enseñanza del pseudocódigo y diagrama de flujo**”, tiene como propósito la mejora del proceso de enseñanza y aprendizaje de la materia Fundamentos de Informática, encontrando nuevas formas para que los alumnos pueden desarrollar sus habilidades de solución de problemas a través del pseudocódigo y diagramas de flujo; ya que al inicio de ese aprendizaje, los algoritmos son más importantes que los lenguajes de programación para enseñar a resolver problemas de una forma sistematizada y enfocada al diseño.

El anterior contribuye a la investigación planteada porque presenta un análisis de las herramientas automatizadas para la enseñanza y aprendizaje de la lógica de programación, lo cual establece criterios para su selección como recurso didáctico un curso propedéutico para la enseñanza de algoritmos.

Palabras clave: enseñanza y aprendizaje, pseudocódigo, diagramas de flujo, usabilidad.

- Case (2012) “**An Animated Pedagogical Agent For Assisting Novice Programmers Within A Desktop Computer Environment**”, esta tesis doctoral demuestra cómo un agente pedagógico animado virtual puede extender las acciones del modelo creencias-deseos-intenciones para proveer el apoyo de docentes y mentores en el *coaching* a programadores principiantes que están aprendiendo su primer lenguaje de programación. La pedagogía del aprendizaje cognitivo proporciona la base teórica de la estrategia de tutoría de agentes. También se fundamenta en el razonamiento basado en casos, el *coaching* y la interacción con el alumno. Los resultados indican que en un pequeño estudio controlado cuando los principiantes son asistidos por el agente pedagógico virtual, son más productivos que los que trabajan sin su ayuda y son mejores en los ejercicios de resolución de problemas. Además, manifiestan un mayor grado de compromiso y aprendizaje del lenguaje.

Esta investigación doctoral proporciona elementos para la definición de las pedagógicas relacionadas con el aprendizaje de la programación. Además de sustentos sobre las dificultades que los estudiantes tienen para ese aprendizaje.

Palabras clave: enseñanza y aprendizaje, pseudocódigo, diagramas de flujo, usabilidad.

- Teague (2011) en su trabajo “**Pedagogy of Introductory computer programming: A people-first approach**”, para la obtención de su grado de Maestría en Tecnologías de la Información, realiza un estudio sobre las dificultades que confrontan los estudiantes en el aprendizaje de la programación, planteándose lo siguiente “*¿La programación es realmente tan*

difícil o hay otras barreras al aprendizaje que tienen un efecto serio y perjudicial en la progresión del estudiante?”. Tanto los resultados empíricos como los resultados anecdóticos de sus experimentos apoyaron su hipótesis.

Este estudio proporciona elementos sobre las dificultades que los estudiantes tienen para el aprendizaje de la programación y una serie de barreras adicionales que agravan esas dificultades.

Palabras clave: introducción a la programación, aprendizaje de programación, pedagogía de la programación, aprendizaje colaborativo, programación entre pares.

- López y otros (2011), “**Una plataforma de evaluación automática con una metodología efectiva para la enseñanza/aprendizaje en programación de computadores**”, es un trabajo que, partiendo de la premisa que aprender a programar computadores es un proceso difícil para los estudiantes novatos y un desafío a las metodologías empleadas por los docentes, presenta una plataforma de evaluación automática que apoya el proceso de enseñanza y aprendizaje en cursos introductorios de programación de computadores para estudiantes de ingeniería. Esta plataforma utiliza mecanismos que combinan análisis estático/dinámico y aplican evaluación de comprensión/análisis en línea, permitiendo una retroalimentación personalizada a los alumnos. También muestra los criterios usados para crear secuencias didácticas de problemas y una forma de aplicarlas efectivamente mediante la evaluación automática.

El trabajo aporta fundamentos teóricos y enfoques sobre la enseñanza de programación; y formula propuestas de estrategias metodológicas que debe seguir el docente para la enseñanza de la programación. Siendo esto totalmente extrapolable al ámbito de conocimiento de los algoritmos.

Palabras clave: evaluación automática de programas, aprendizaje activo, autoeficacia, modelos mentales, análisis estático y dinámico.

- Bennedsen (2007), investigador PhD, en **“Teaching and Learning Introductory Programming - A Model-Based Approach”**, trabajo de disertación sobre un enfoque basado en modelos para el aprendizaje y la enseñanza de la programación orientada a objetos. Sugiere pautas pedagógicas y comparte mejores prácticas en dicha enseñanza en cursos introductorios.

Este trabajo contribuye a la investigación por su enfoque sobre el aprendizaje y la enseñanza de la programación orientada a objetos, aporta fundamentación teórica sobre la enseñanza de la programación que son extrapolables a la enseñanza de los algoritmos.

Palabras clave: aprendizaje y enseñanza de programación, programación orientada a objetos, enfoque basado en modelos, pedagogía de la programación.

- Moroni y Señas (2005), **“Estrategias para la enseñanza de la programación”**, en su trabajo presentan un ambiente de aprendizaje que contiene un editor interactivo de algoritmos, un constructor automático de trazas y un traductor de algoritmos a programas en lenguaje Pascal. Esto con el propósito de contribuir a la mejora del proceso de enseñanza y aprendizaje en programación; basándose en que una estrategia valedera es comenzar a enseñar programación utilizando los algoritmos como recursos esquemáticos para plasmar el modelo de la resolución de un problema.

La anterior investigación aporta enfoques sobre la enseñanza de algoritmos computacionales y las estrategias metodológicas a aplicar.

Palabras clave: ambiente de aprendizaje, algoritmos, chequeo de algoritmos, enseñanza de la programación, resolución de problema.

- Guardián Soto (2003), trabajo de grado de maestría **“Estrategias para promover el aprendizaje significativo de la asignatura de análisis de algoritmos en el nivel de educación superior”**, presenta una propuesta metodológica para introducir a los estudiantes del curso de análisis de algoritmos en el uso y aplicación de estrategias de aprendizaje significativo. Esto los ayudó a mejorar su proceso de aprendizaje y, por ende, su rendimiento académico. La propuesta se fundamenta en la teoría denominada constructivismo.

Esta investigación también aporta elementos para la estructuración del trabajo que se quiere desarrollar y para la formulación de elementos del marco referencial desde la perspectiva de la teoría del aprendizaje.

Palabras clave: análisis de algoritmos, estrategia de aprendizaje significado, teoría constructivista.

- Blaked (2002), **“A Review and Assessment of Novice Learning Tools for Problem Solving and Program Development”**, este trabajo de maestría tiene por objetivo revelar las características que deben tener los componentes que integran una herramienta de aprendizaje para los programadores novatos, de modo que sea efectiva. Con base a los hallazgos obtenidos del estudio realizado, se hacen las recomendaciones para el desarrollo de herramientas de aprendizaje de los principiantes.

El aporte de este trabajo se encuentra en la descripción de las dificultades con las que se encuentran los aprendices de programación, clasificándolas del siguiente modo: dificultades de causas pedagógicas, de causas psicológicas, dificultades del lenguaje de programación, las originadas para

realizar *debugging*, por influencias externas, entre otras. Esto puede extrapolarse al aprendizaje de los algoritmos.

Palabras clave: aprendizaje de la programación, resolución de problemas, análisis de algoritmos, herramientas para el aprendizaje de la programación.

2.2 Referencias Teóricas

2.2.1 Sobre los Algoritmos

Con base a expuesto por Yamarte (2005), en esta sección muestra los conceptos básicos de algoritmos.

2.2.1.1 Algoritmos y Programación

Hasta ahora se ha comentado que un algoritmo describe cómo hacer una tarea o actividad o cómo resolver un problema específico. A continuación se dan dos definiciones del término algoritmo, que se encuentran dentro de la orientación del mundo de la computación:

Definición 1:

Un algoritmo es una secuencia ordenada de pasos precisos (sin ambigüedades o contradicciones) que permiten realizar una tarea o resolver un problema específico.

Definición 2:

Un algoritmo es una solución genérica de un problema particular.

Un algoritmo debe tener las siguientes características:

- a) **Efectivo**: Cumplir con el objetivo para el cual fue creado.
- b) **Preciso**: Indicar claramente el orden y la *acción* a realizar en cada paso.
- c) **Robusto**: Al ejecutar un algoritmo repetidas veces con el mismo conjunto de datos siempre debe proporcionar los mismos resultados, además en todo momento debe ser capaz de dar una respuesta adecuada a los datos dados.
- d) **Delimitado**: Los pasos deben estar delimitado por una marca (o etiqueta) de inicio y una de fin.

e) **Finito**: Un algoritmo tiene que culminar su ejecución en un momento determinado, no debe ejecutarse infinitamente.

Los algoritmos que se referencian en este trabajo son aquellos del mundo de la programación de computadoras, también llamados *algoritmos de programación* o *algoritmos computacionales*, de aquí en adelante serán referidos solamente como algoritmos.

Una vez desarrollado un algoritmo se traduce a un lenguaje de programación, para que pueda ser ejecutado en una computadora. En ese contexto, un algoritmo es producto de la fase de diseño en la que se construye un programa que ha de ejecutarse en una computadora.

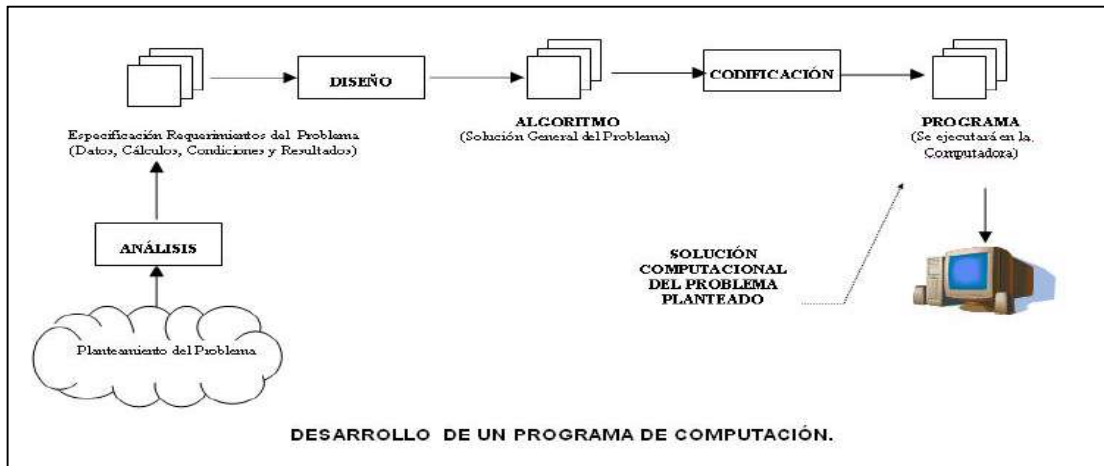


Figura 1: Ciclo Genérico del Desarrollo de un Programa.

2.2.1.2 Resolución de Problemas Por Computadoras

En la figura 3.2.1.1 se muestra, de forma general, los pasos a seguir desde que se plantea el problema hasta que se obtiene el programa que se ejecutará en la computadora. Cuando se tiene un problema que se quiere resolver mediante una computadora, deben seguirse una secuencia de pasos para la elaboración de una solución, por lo que se puede decir que esa secuencia de pasos ilustrada en la figura corresponde a un algoritmo (*¿por qué?*).

La resolución de problemas por computadoras comprende dos fases: la primera, tiene como objetivo elaborar el algoritmo o solución general del problema y la segunda, obtener a partir del algoritmo anterior el programa que se ejecutará o funcionará en la computadora.

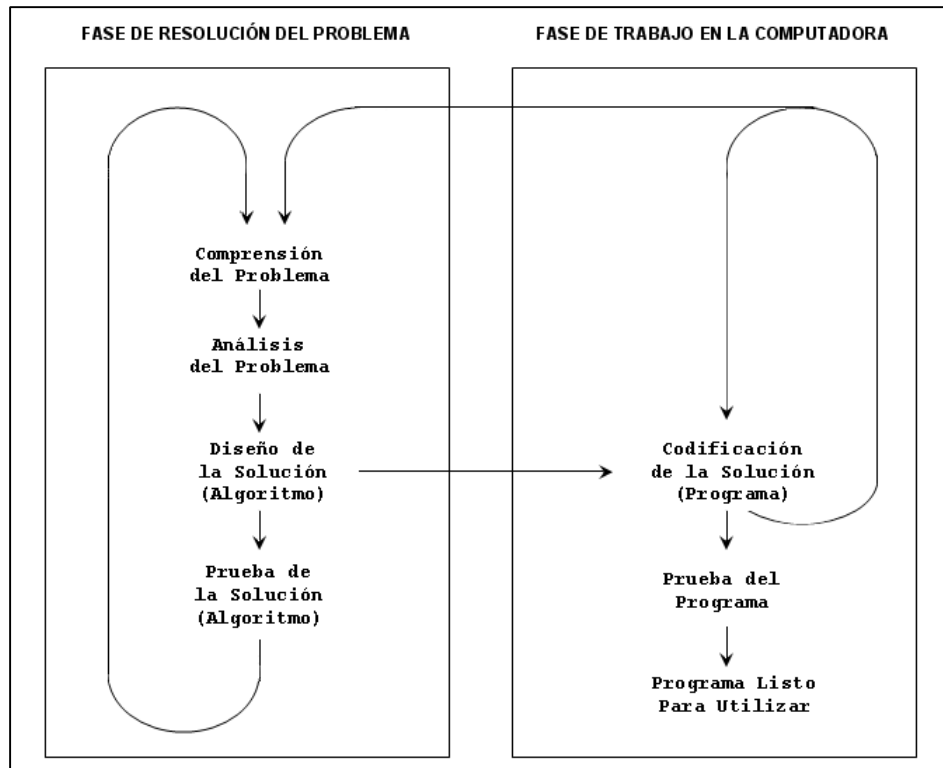


Figura 2: Ciclo Resolución de Problemas Por Computadora.

Para construir un algoritmo o solución del problema, hay que enfocarse primero en la fase de resolución del problema. En el mundo de la Computación, y disciplinas afines²; el término más apropiado para referirse a la creación, lógica de secuenciación, clasificación, definiciones, conceptos, técnicas, formas de uso o aplicaciones y todo lo relacionado con los algoritmos computacionales, sistemas digitales o sistemas informáticos, es *algorítmica*. No obstante, a pesar de su amplia utilización en diversas aplicaciones de diferentes disciplinas, no se tienen definiciones únicas, estructuradas y formalizadas ni de *algorítmica* ni de *algoritmo*.

² Informática, Electrónica, Tecnologías de la Información y Comunicación (TIC), entre otras.

La algorítmica es la disciplina que se encarga del estudio y elaboración de los algoritmos, mientras que la programación (de computadoras) cubre la fase siguiente a la creación del algoritmo, la construcción de los programas cuya implementación se hace en el computador.

2.2.1.3 Algoritmo: Fase de Resolución del Problema

Como se comentó anteriormente, para elaborar un algoritmo o solución del problema, hay que enfocarse primero en la resolución del problema, aplicando una serie de pasos que permitirán obtener el algoritmo que representa una solución del mismo.

Fase de Resolución del Problema

1. Comprensión del problema
2. Análisis del problema
3. Diseño de la solución
4. Prueba de la solución

1. Comprensión del problema

El problema a resolver puede ser planteado en forma oral o escrita, independientemente de la forma en que se haga, el mismo debe ser un planteamiento coherente, que no posea ambigüedades o contradicciones.

Comprender el problema significa que se tenga claro el problema, que se conozca precisamente en **qué** consiste el mismo. Para ello se deben realizar las siguientes actividades:

- Leer (o repasar mentalmente) detenidamente el planteamiento, tantas veces como sea necesario.
- Formular las preguntas necesarias para aclarar las dudas.
- Realizar un “inventario” de la información que poseemos para resolver el problema.
- Buscar la información que nos hace falta, mediante: revisión de bibliografía, apuntes, consultas a otras personas, etc.
- Pensar el problema, reflexionarlo, preguntarme a mí mismo las cuestiones referentes al problema.

- Verbalizar el problema: expresar con mis propias palabras, en forma clara y coherente, el problema planteado.

2. Análisis del problema

El análisis es un proceso de reflexión del problema. En este proceso se efectúa una separación de los elementos constituyentes del problema, se ve cómo está formado y cuáles de esos elementos son relevantes (importantes) o no para resolverlo.

En problemas sencillos la descomposición será sencilla. En problemas complejos la descomposición requerirá de una mayor atención y esfuerzo intelectual, para poder determinar los sub-problemas, menos complejos, que integran el problema original.

El análisis debe llevar a una visualización del problema que nos permita:

- ✓ Obtener o derivar la información realmente importante para su resolución.
- ✓ Transformar el problema inicial e problemas más pequeños, esto es hacer una descomposición que nos ayude a resolver el problema a través de la resolución de los problemas más pequeños.
- ✓ Expresar en términos de una notación adecuada (lenguaje escrito, lenguaje matemático, gráficos, etc.) los requerimientos del problema.

Para lograr un análisis apropiado se pueden contestar las siguientes preguntas:

- Respecto a los datos:
 - ⇒ ¿Qué datos proporciona el planteamiento del problema?
 - ⇒ ¿Qué datos se necesitan para resolver el problema?
 - ⇒ ¿Hay que generar algún dato en función de los ya conocidos? ¿A partir de cuáles datos se genera?
 - ⇒ ¿Cuáles son los datos de salida (resultados) que hay que proporcionar?
- Respecto a la conformación del problema:

- ⇒ ¿Se puede simplificar el problema? ¿Cuáles son las partes o sub-problemas que conforman el problema?
- ⇒ ¿Hay dependencia entre los sub-problemas? ¿Cuál es el orden para resolverlos?
- Respecto a problemas que haya resuelto antes:
 - ⇒ ¿El problema se parece a alguno que ya resolví?
 - ⇒ ¿En qué se relaciona?
 - ⇒ ¿Qué elementos sirven para solucionar el problema actual?

3. Diseño de la solución

En este momento se plantean los pasos a seguir para resolver el problema, aquí se indica **cómo** se resuelve el problema, se diseña y se elabora una solución. Aquí se presenta el **algoritmo** que resuelve el problema, teniendo en cuenta el esquema que indica las partes a diseñar en un algoritmo:

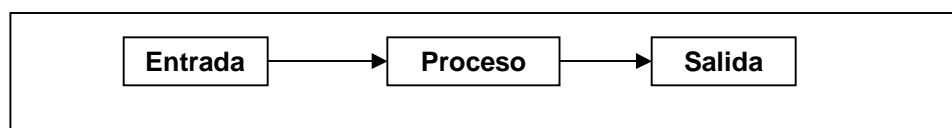


Figura 3: Esquema General de un Algoritmo.

- **Entrada:** Especifica los pasos para la obtención de los datos de entrada a procesar en el algoritmo.
- **Proceso:** Abarca los pasos que indican las acciones (operaciones y/o cálculos) requeridos para obtener la solución adecuada.
- **Salida:** Conformada por los pasos que se encargan de proporcionar o mostrar el(los) resultado(s) obtenido.

Es muy importante que los pasos o acciones a seguir en el algoritmo estén ordenados, ya que de ello depende que se obtenga o no la solución del problema a resolver, ya que ése orden indica la secuencia de ejecución de los pasos, los cuales deben ser señalados o especificados de forma clara y precisa.

4. Prueba de la solución

Consiste en probar la solución ejecutándola paso a paso, a partir de un conjunto de datos de prueba. La idea es evaluar los resultados obtenidos para determinar si el problema está bien resuelto o si es necesario realizar algún ajuste en la solución para lo cual se debe volver al inicio de la fase para hacer las modificaciones necesarias.

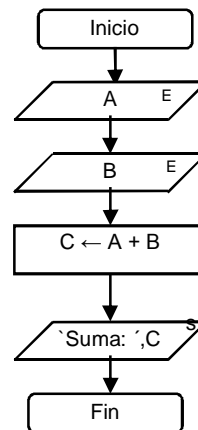
Ejemplos de Algoritmos:

Algoritmo #1:

Notación: Pseudo-código

- (0) Inicio
- (1) Leer A
- (2) Leer B
- (3) $C \leftarrow A + B$
- (4) Escribir 'Suma: ', C
- (5) Fin

Notación: Diagrama de Flujo



Comentarios:

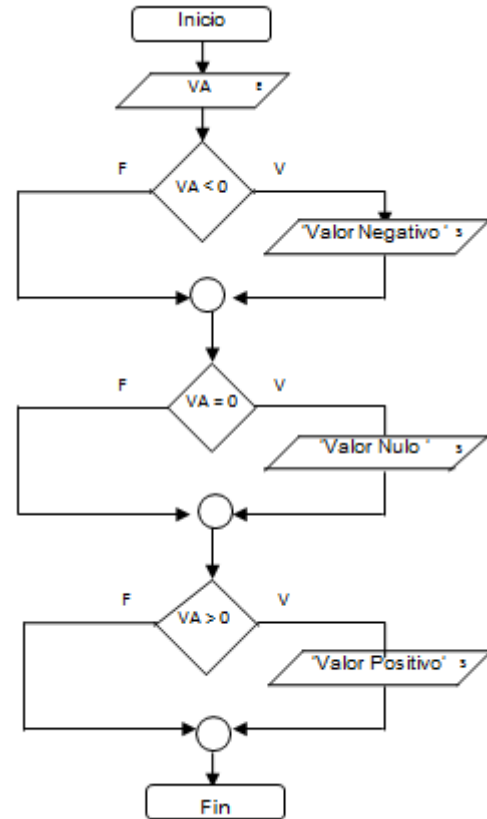
El algoritmo anterior es una solución del clásico ejercicio “Calcular la suma de un par de valores numéricos”. Se usan dos *variables* para representar los valores a sumar y una tercera que contendrá la suma en cuestión. Finalmente se muestra el resultado obtenido. Pregunta: ¿Cuáles acciones del algoritmo corresponden a la Entrada, Proceso y Salida de Datos?

Algoritmo #2:

Notación: Pseudo-Código

- (0) Inicio
- (1) Leer VA
- (2) Si $VA < 0$ entonces
 - (2.1) Escribir 'Valor Negativo'Fsi
- (3) Si $VA = 0$ entonces
 - (3.1) Escribir 'Valor Nulo'Fsi
- (4) Si $VA > 0$ entonces
 - (4.1) Escribir 'Valor Positivo'Fsi
- (5) Fin

Notación: Diagrama de Flujo



Comentarios:

El algoritmo anterior determina si un valor numérico (VA) dado es negativo, nulo o positivo. Para lo cual realiza una serie de preguntas que permiten comparar VA contra cero, mostrando el mensaje adecuado a la *condición* que se cumpla.

En éste caso el algoritmo no efectúa cálculos matemáticos, en su lugar realiza operaciones de comparación, a través de preguntas, que le permiten establecer la salida apropiada.

2.2.1.4 Elementos Conceptuales Para Elaborar Algoritmos

Aquí se abordan los elementos de base para llevar a cabo la construcción de un algoritmo. Se definen los conceptos básicos: tipos de datos, objetos algorítmicos, acciones algorítmicas elementales, formas de notación algorítmica, acciones de decisión y acciones cíclicas básicas.

2.2.1.4.1 Conceptos Básicos

- *Tipos de Datos*

La información que se utiliza para resolver un problema está formada por una selección de los datos que son relevantes para elaborar una solución del mismo. Los datos constituyen la base del problema, ya que son los elementos fundamentales a procesar para establecer una solución adecuada. Un dato es una abstracción o representación simbólica que proporciona información sobre un hecho, descripción de una propiedad o característica de una persona o de un objeto. Ejemplos de datos:

- Primera letra del abecedario: A
- Número Pi: 3.14
- Mediodía: 12 m
- Altura de una persona: 1.60 m
- Edad de una persona: 23 años
- Nombre de mujer: María
- Nombre de hombre: Luís
- ¿La tierra es redonda?: Verdad

De acuerdo a la naturaleza de los valores que pueda tomar un dato estableceremos el tipo al cual corresponde:

Tipos de Datos Numéricos

- Entero:
Refiere valores pertenecientes al conjunto de números enteros (conjunto Z), ejemplos: 4, 5, 100, -200, 0, etc.; abarca tanto valores negativos como positivos o nulos.
- Real:
Refiere valores correspondientes al conjunto de números reales (conjunto R), el cual posee parte decimal, ejemplos: 2.5, 125.99, 0.5, etc.; también abarca tanto valores negativos como positivos o nulos.

Tipos de Datos No Numéricos

- Carácter:
Refiere al conjunto conformados por los caracteres alfabéticos (letras), dígitos numéricos y caracteres especiales. Para hacer referencia a valores de éste tipo los mismos deben ser delimitados entre apóstrofe (comilla simple), ejemplos: 'a', 'V', 'U', 'A', '9', '8', '1', '*', ' ' (espacio o blanco), etc. La longitud máxima de este tipo es 1, ya que puede contener el carácter nulo (") .
- Cadena:
Son aquellos valores que contienen una secuencia finita de más de un carácter, los cuales deben ser indicados entre apostrofes (comilla simple), ejemplos: 'casa', 'luna', 'resultado #3', 'La computadora', etc. También se les conoce como literales.

Carácter Nulo

Es una cadena de caracteres cuya longitud es cero, lo que significa que no contiene carácter alguno. Se representa a través de dos apóstrofes seguidos ("). No se debe confundir con el carácter blanco, el cual se representa con un espacio encerrado entre apóstrofes (' ').

- Lógico:
Refiere el conjunto de valores lógicos o booleanos conformados por los valores Verdad y Falso.

El tipo de dato también indica las operaciones o cálculos que pueden ser aplicados al dato en cuestión, por ejemplo, a los datos numéricos se les puede aplicar operaciones tales como suma, resta, multiplicación o división.

- *Objetos Algorítmicos*

Un algoritmo realiza operaciones sobre los datos del problema que soluciona, dichos datos se representa a través de objetos que son operados en el algoritmo. Un objeto algorítmico es una representación de un dato o valor importante para la solución del problema. Los objetos básicos para desarrollar un algoritmo son las variables, constantes y expresiones.

Variables

Una variable es un objeto algorítmico referenciable mediante un *identificador* o *nombre*, tiene asociada un tipo de dato y un único valor en un momento determinado, el cual puede cambiar por efecto de una *acción* algorítmica. Lo que significa que en un lapso determinado, una variable -como su nombre lo indica-, puede tomar diferentes valores.

Identificador o Nombre

Es una secuencia de caracteres que identifican un objeto algorítmico. Dicha secuencia puede estar integrada por caracteres alfabéticos y/o numéricos, y de ser necesario, el carácter especial de subrayado (_), cuya formación está sujeta a las siguientes reglas:

- 1.- El primer carácter de la secuencia debe ser un carácter alfabético.
- 2.- No debe contener espacios.
- 3.- Debe tener un uso mnemotécnico, esto quiere decir que el nombre esté relacionado con el significado del dato que representa.

Ejemplos:

* Identificadores o nombres válidos: A, B, SUMA, FACTOR1, FACTOR2.

* Identificadores no válidos o incorrectos: 9R, LA SUMA, C#1.

Nota: Para una distinción más clara de los objetos variable y constante (nominada), sólo usaremos combinaciones con letras mayúsculas.

En un algoritmo las variables son utilizadas para representar los datos relevantes del problema, independientemente del valor que tenga para una situación determinada. En los ejemplos mostrados en la sección 3.2.1.3., se observa la utilización de variables que representan los datos relevantes para

el problema resuelto, a continuación se presenta una tabla que describe las variables utilizadas en cada uno de esos algoritmos:

Algoritmo	Variable	Tipo de Dato	Tipo de Uso
#1	A: representa el primer sumando	Numérico - Real	Entrada
	B: representa el segundo sumando	Numérico - Real	Entrada
	C: contiene la suma de A + B	Numérico - Real	Salida
#2	VA: representa el valor a clasificar	Numérico - Real	Entrada

Constantes

Una constante es un objeto algorítmico cuyo valor permanece fijo durante todo el algoritmo. Puede ser utilizada de forma explícita indicando directamente el valor de la constante o mediante la asignación de un nombre o identificador, a través del cual se hace referencia a la constante. Esta segunda forma es muy ventajosa cuando se requiere utilizar una constante a lo largo de un algoritmo, ya que en caso de tener que modificarla, el cambio se realiza sólo en un lugar.

Ejemplos:

* Constantes explícitas

Tipo de Dato	Valor
Numérico Entero	3, -3, 0, 100
Numérico Real	3.5 0.3 4.2
Carácter	'a' 'M' '*' '#' 'Z'
Cadena	'COSA' 'Resultado' 'Las 3 calles' 'Luis José'
Lógico	Verdad Falso

* Referenciadas mediante un identificador:

Tipo de Dato	Nombre	Valor
Numérico Entero	PRIMER_PAR	2
Numérico Real	PI	3.14
Carácter	DOBLE_RAYA	'--'
	PUNTO	'.'
Cadena	CAPITAL_VZLA	'Caracas'
Lógico	FINALIZO	Falso

Expresiones

Una expresión es una combinación válida de operandos y operadores, de acuerdo a las reglas que señale una notación particular³.

Un operando es un valor representado por objeto algorítmico (variable, constante o expresión) y un operador es un símbolo que denota una operación o cálculo a efectuar sobre los operandos. La validez de una expresión está sujeta a que se combinen operandos de tipo de datos compatibles según la operación que vaya a ser aplicada.

Ejemplos:

Expresiones Válidas: (i) $A + B$ (ii) $B * H / 2$ (iii)
 $(N1+N2+N3)/3$

Expresiones No Válidas: (iv) $+ C B$ (v) $7 9 * A -$ (vi) $X Y - A$

Al igual que las variables y las constantes, las expresiones tienen un tipo de dato asociado, el cual viene determinado por el tipo de dato del valor resultante al evaluar la expresión, es decir, si el resultado que se obtiene es del tipo numérico, entonces se dice que la expresión es del tipo numérico.

Evaluar una Expresión

Es determinar el valor resultante o resultado de la expresión y consiste en la aplicación de las operaciones o cálculos, indicados por los operadores, a los valores de los operandos que la conforman.

³Aquí se usa la típica notación matemática <operando><operador><operando>, también conocida como **Notación Infija**.

Expresiones Numéricas

Una expresión numérica es aquella cuyo resultado es del tipo de dato numérico (entero o real). Dependiendo de la operación, en una expresión numérica se pueden mezclar datos del tipo entero y real.

Operaciones Básicas

Las operaciones o cálculos numéricos básicos a usar en el desarrollo de un algoritmo son: suma, resta, multiplicación y división.

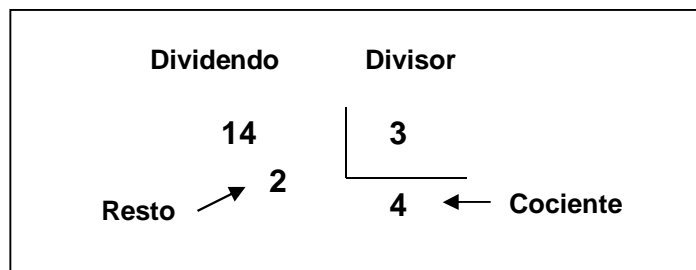
Operación	Operador	Operandos Requeridos
Suma	+	2
Resta	-	2
Multiplicación	*	2
División	/	2

Operaciones DIV y MOD

Definiremos un par de operaciones numéricas que nos serán de mucha en la realización de algoritmos: DIV y MOD.

La operación DIV proporciona el cociente de una división entera⁴ y la operación MOD nos da el resto de dicha división. Esto significa que los operandos involucrados tienen que ser del tipo de dato numérico entero. Veamos cómo se utilizan dichos operadores, para ello recordemos los factores que integran la operación de división:

$$\text{Dividendo} = \text{Divisor} * \text{Cociente} + \text{Resto}$$



⁴ En ésta división no se calcula los decimales, la operación culmina cuando se obtiene del lado del dividendo un número menor al divisor.

El primer operando, tanto en el DIV como en el MOD, que se indica al lado izquierdo del operador, especifica el dividendo y el segundo operando, indicado al lado derecho, representa el divisor.

<operando1>**DIV**<operando2>

<operando1>**MOD**<operando2>

La expresión **14 DIV 3** proporciona el cociente de la división entera 14 entre 3. ¿Cuál es el resultado que se obtiene con dicha expresión?, como ya se ilustró en el ejemplo, el valor obtenido es 4.

La expresión **14 MOD 3** calcula el resto de la división entera 14 entre 3, y su resultado es 2 (¿por qué?).

Dominio y Rango de las Operaciones

En matemática los conceptos de dominio y rango de una función definen, respectivamente, los conjuntos de partida y de llegada de la misma.

Dominio y Rango

El dominio de una operación nos indica la cantidad y tipo de dato de los operandos necesarios para efectuar la operación.

El rango nos define el tipo de dato del resultado que se obtiene.

Partiendo de lo anterior, para cada operación se muestra una tabla con las combinaciones válidas de los tipos de datos que puede admitir, así como el tipo del resultado que se obtiene:

Suma				
Entero	+	entero	=	entero
Entero	+	real	=	real
Real	+	entero	=	real
Real	+	real	=	real

Resta				
Entero	-	entero	=	entero
Entero	-	real	=	real
Real	-	entero	=	real
Real	-	real	=	real

Multiplicación				
Entero	*	entero	=	entero
Entero	*	real	=	real
Real	*	entero	=	real
Real	*	real	=	real

División				
Entero	/	entero	=	real
Entero	/	real	=	real
Real	/	entero	=	real
Real	/	real	=	real

Cociente de División Entera (DIV)				
Entero	DIV	entero	=	entero

Resto de División Entera (MOD)				
Entero	MOD	entero	=	entero

Ejemplos:

Expresión Numérica	Resultado
2 + 3.5	5.5
18 - 9	9
4 * 2	8
15 / 3	5

Expresiones Alfanuméricas

Una expresión alfanumérica es aquella cuyo resultado es del tipo de dato carácter o del tipo cadena. La concatenación será la única operación de caracteres que utilizaremos, su operador es el mismo de la suma numérica (+).

Concatenación				
carácter	+	Carácter	=	cadena
carácter	+	Cadena	=	cadena
cadena	+	Carácter	=	cadena
cadena	+	Cadena	=	cadena

Ejemplos:

Expresión Alfanumérica	Resultado
'La' + 'Casa'	'LaCasa'
'La' + ' ' + 'Casa'	'La Casa'
'Resultado' + '#1'	'Resultado #1'

Expresiones Lógicas

Una expresión lógica es aquella cuyo resultado sólo puede ser verdad o falso, es decir, su tipo de dato es lógico. Este tipo de expresión es de gran utilidad en el desarrollo de soluciones algorítmicas, ya que a través de la misma se pueden establecer la evaluación de condiciones o preguntas necesarias para la toma de decisiones en un algoritmo.

Para construir expresiones lógicas se usan las operaciones relacionales o de comparación.

Operaciones Relacionales		
Operación	Operador	Operandos Requeridos
Igual que	=	2
Diferente que	<>	2
Menor que	<	2
Menor o igual que	<=	2
Mayor que	>	2
Mayor o igual que	>=	2

Ejemplos:

Expresión	Significado	Resultado
$3 = 4$	¿3 es igual a 4?	falso
$5 < 8$	¿5 es menor que 8?	verdad
$160 \geq 100$	¿160 es mayor o igual que 100?	falso
$33 \neq 79$	¿33 es diferente que 79?	verdad
$2 = 2$	¿2 es igual a 2?	verdad

En algunos casos se necesita realizar la evaluación de dos o más condiciones, por lo que se debe utilizar más de una expresión lógica, lo cual se hace uniendo las expresiones mediante los llamados conectores u operadores lógicos.

Conectores Lógicos		
Operación	Operador	Operandos Requeridos
Conjunción (Y lógico)	Y	2
Disyunción (O lógico)	O	2
Negación	No	1

Ejemplos:

Expresión Lógica	Significado
$A > B$ y $B > C$	A mayor que B y B mayor que C
$Y = 2$ o $Z = 5$	E igual a 2 o Z igual 5
$D < 0$ y $A > 0$	D menor que 0 y A mayor que 0

Expresiones como las antes mostradas reciben el nombre de **expresiones compuestas**. ¿Cómo se evalúa una expresión compuesta? Se evalúa cada expresión simple o individual por separado y luego, esos resultados se usan para la evaluación del operador lógico que conecta a las expresiones. Veamos cómo se realiza:

Seguidamente se presenta la tabla de las combinaciones de valores a evaluar con los conectores lógicos:

Y lógico				
Verdad	Y	verdad	=	verdad
Verdad	Y	falso	=	falso
Falso	Y	verdad	=	falso
Falso	Y	falso	=	falso

O lógico				
verdad	o	verdad	=	verdad
verdad	o	falso	=	verdad
falso	o	verdad	=	verdad
falso	o	falso	=	falso

Negación		
no	(verdad)	= falso
no	(falso)	= verdad

Una expresión lógica puede ser negada, esto consiste en negar todos y cada uno de los operadores (relacionales y/o conectores) que conforman la expresión original.

Ejemplos:

Expresión Original	Negación Expresión Original
$A = B$	$A \neq B$
$X \leq Y$	$X > Y$
$C = 0$ o $C = 5$	$C \neq 0$ y $C \neq 5$
$A > B$ y $B > C$	$A \leq B$ o $B \leq C$
$X > 0$ y ($Y < 0$ o $Z < 0$)	$X \leq 0$ o ($Y \geq 0$ y $Z \geq 0$)

En la siguiente tabla se muestran la negación de los operadores relacionales y los conectores lógicos:

Operador	Negación
=	\neq
\neq	=
<	\geq
\leq	>
>	\leq
\geq	<
y	o
o	y

Precedencia de las Operaciones

Es el orden predeterminado que indica la prioridad de evaluación de cada operación. Lista de precedencia:

1. Negación (no).
2. Multiplicación (*), división (/), DIV, resto (MOD) e "Y lógico".
3. Suma (+), resta (-) u "O lógico"
4. Operaciones relacionales (=, \neq , <, >, \leq , \geq).

Cuando una expresión tiene operaciones de distintos tipos (numéricos, alfanuméricos y lógicos), primero se evalúan las operaciones numéricas, luego las relacionales y finalmente los conectores lógicos.

Si se tienen dos operaciones de igual prioridad se evalúan, según su orden de aparición, de izquierda a derecha.

El orden de precedencia de las operaciones puede alterado, encerrando entre paréntesis aquellas expresiones que se quiere sean evaluadas con mayor prioridad.

- *Acciones Algorítmicas Elementales*

Una acción algorítmica es una orden o comando con una duración finita y un efecto determinado. Cada acción algorítmica se define mediante la indicación de su sintaxis y semántica.

Sintaxis y Semántica

La sintaxis indica el formato con que debe escribirse una acción.

La semántica proporciona la descripción del significado y funcionamiento de la acción, definiendo claramente cuál es el efecto que causa sobre el objeto algorítmico sobre el que actúa.

Acción de Lectura de Datos

Los datos que se procesan en un algoritmo se obtienen mediante la ejecución de una acción de lectura (**Leer**). Para la computadora se trata de la obtención, desde una fuente externa, de los datos a procesar.

Sintaxis:

Leer <variable>

Semántica:

Permite dar una entrada o proporcionar un valor (cualquiera, no calculado) a la variable referenciada, a través de su nombre, en la acción **Leer**.

Ejemplo:

Leer V Indica que debe dársele un nuevo valor a la variable A

Acción de Salida de Datos

Así como se requiere tener una acción para la entrada de datos, también es necesario disponer de una que permita mostrar los resultados del procesamiento, éste tipo acción se denomina acción de escritura (**Escribir**). Desde el punto de vista de la computadora, el objetivo es proporcionar los resultados del procesamiento efectuado, es decir, mostrar los datos de salida.

Sintaxis:

Escribir <lista de objetos algorítmicos>

Semántica:

Permite dar una salida o mostrar los valores de cada uno de los objetos que conforman la lista de la acción **Escribir**. Si la lista tiene más de un objeto algorítmico, estos deben ser separados por coma.

Ejemplos:

Escribir V Muestra el valor que tenga almacenado la variable V

Escribir 'Cosa' Escribe la constante, tipo cadena, 'Cosa'

Escribir 'Resultado = ', C Muestra la cadena 'Resultado = ' sigueel
valor de C

Acción de Asignación

Esta acción permite efectuar el cambio de valor de una variable. La asignación se ejecuta en dos partes: primero, se hace la evaluación de la expresión que indica los cálculos a realizar; y segundo, se reemplaza el valor de la variable sujeta al cambio, por el valor resultante de la expresión.

Para que la acción de asignación sea válida el tipo de dato de la expresión debe ser compatible con el de la variable.

Sintaxis:

<variable>←<expresión>

Semántica:

Una vez evaluada la <expresión> (lado derecho) se almacena el valor obtenido en la <variable> (lado izquierdo).

Ejemplos:

(i) X ← 2

(ii) A ← B + C * (D / 4)

(iii) L ← falso

(iv) MSJ ← 'Este es el resultado'

Secuenciación de Acciones

Sintaxis:

<acción algorítmica₁>
<acción algorítmica₂>
:
<acción algorítmica_n>

Semántica:

Consiste en colocar en un orden determinado un conjunto de acciones algorítmicas, en función de la tarea que se debe ejecutar. Al secuenciar acciones se está construyendo un algoritmo completo o un segmento del mismo.

Ejemplo:

```
(0) Inicio
(1)  Leer A
(2)  Leer B
(3)   $C \leftarrow A + B$ 
(4)  Escribir C
(5) Fin
```

Nótese que la secuencia de acciones anterior conforma un algoritmo completo, cuyo objetivo es sumar los valores A y B. Las acciones (0) y (5) corresponden a los delimitadores del texto del algoritmo.

- *Notaciones Algorítmicas*

Para que un algoritmo sea preciso cada acción que lo integra debe estar claramente definida, tanto en su sintaxis (forma de escribirla) como en su semántica o significado. Esto implica que el algoritmo esté escrito en una notación bien definida. Los dos tipos de notaciones más relevantes para la escritura de algoritmos son las siguientes:

Diagramas:

Consiste en una representación gráfica de las acciones de un algoritmo. El tipo de diagrama más conocido y utilizado es el diagrama de flujo. Su funcionamiento se basa en la secuenciación de figuras, que representan las acciones, unidas por flechas que indican el flujo u orden de ejecución a seguir. Son prácticos para algoritmos con pocas acciones, pero muy engorrosos para algoritmos extensos, a pesar de tener un poder gráfico que permite visualizar la funcionalidad global del algoritmo. Símbolos básicos de un diagrama de flujo:


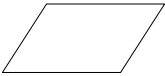

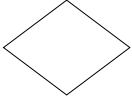


Símbolo	Función
	Delimita el inicio y el fin del algoritmo.
	Denota una acción de entrada o de salida de datos.
	Representa una la realización de un asignación o de un proceso más complejo.
	Representa la realización de una pregunta para la toma de una decisión.
	Indica el flujo de la ejecución.
	Conector de dos o más flujos de ejecución.

Figura 4: Simbología de los Diagramas de Flujo.

Pseudocódigo:

Es un lenguaje para escribir algoritmos, se basa en la utilización de un conjunto de palabras reservadas junto a las reglas que rigen su escritura. Surgió como una alternativa ante el empleo del diagrama de flujo, por ser una herramienta más práctica de usar al permitir una modificación más fácil de los algoritmos elaborados.

El pseudocódigo es una notación intermedia entre la descripción en lenguaje natural de las acciones de un algoritmo y su posterior traducción al lenguaje de programación seleccionado para su ejecución en la computadora. Aunque es una notación flexible con respecto a un lenguaje de programación, se caracteriza por estar sujeta a un conjunto de normas que regulan su utilización con el fin de facilitar la traducción del algoritmo a un lenguaje de programación:

- i) Para favorecer la comprensión del algoritmo se pide el uso de la *indentación* o *sangría* respecto al margen izquierdo, lo cual ayuda a delinear visualmente el alcance de las acciones.

- ii) Definición de objetos para la representación de datos (variables, constantes y expresiones).
- iii) Definición de acciones para el procesamiento de datos:
 - Uso de acciones básicas (entrada y salida de datos, y asignación).
 - Uso de acciones de decisión.
 - Uso de acciones cíclicas o repetitivas.
 - Secuenciación de acciones.
- iv) Definición de líneas o bloques de comentarios para la documentación del algoritmo.

El pseudocódigo tiene como ventaja facilitar el aprendizaje de los lenguajes de programación, debido a su similitud con los mismos en el manejo de los conceptos de base que conforman la notación.

- *Acciones Algorítmicas de Decisión*

¿Por qué usar una Acción de Decisión? En muchos casos la resolución de un problema requiere que se detecten situaciones que permitan tomar decisiones sobre las acciones a seguir. Esto se logra a través de la evaluación de condiciones representadas mediante expresiones lógicas. Se diseña el algoritmo de tal forma que dependiendo de la condición que se cumpla se pueda seguir las acciones adecuadas para lograr una respuesta apropiada.

Las acciones algorítmicas que permiten la toma de decisiones, también conocidas como acciones de selección, permiten la realización de una pregunta cuya respuesta es un valor lógico (verdad o falso) y dependiendo del valor obtenido ejecuta la secuencia de acciones indicadas. A continuación se describen las dos acciones algorítmicas, básicas, para la toma de decisión.

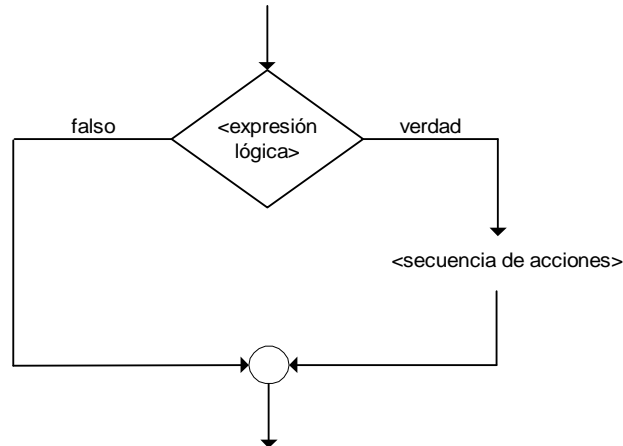
Acción Condicional Simple

Sintaxis:

Pseudocódigo:

```
Si <expresión lógica> entonces  
    <secuencia de acciones algorítmicas>  
Fin-Si
```

Diagrama de Flujo:



Semántica:

Ejecuta la <secuencia de acciones algorítmicas> siempre y cuando la evaluación de la <expresión lógica> haya dado como resultado el valor **verdad**, de lo contrario no se ejecuta dicha secuencia y se pasa a la siguiente acción que esté inmediatamente después de la etiqueta **Fin-Si**, cuyo objetivo es delimitar el alcance de la acción de decisión.

Ejemplos:

```
Si A= 0 entonces  
    Escribir "Valor igual a cero"  
Fin-Si
```

```
Si (X = 0) y (Y = 0) entonces  
    Escribir "Punto en el Origen"  
Fin-Si
```

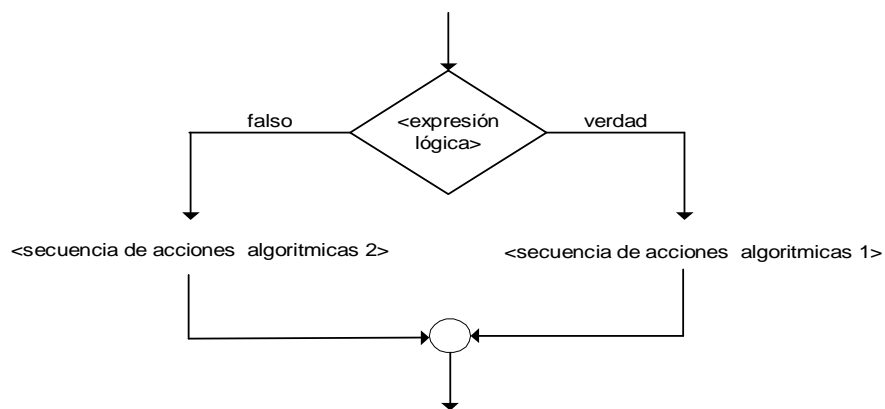
Acción Condicional Doble

Sintaxis:

Pseudocódigo:

```
Si <expresión lógica>entonces  
    <secuencia de acciones algorítmicas 1>  
si-no  
    <secuencia de acciones algorítmicas 2>  
Fin-Si
```

Diagrama de Flujo:



Semántica:

Ejecuta la <secuencia de acciones algorítmicas₁> siempre y cuando la evaluación de la <expresión lógica> haya dado como resultado el valor **verdad**, en caso contrario (**falso**) se ejecuta la <secuencia de acciones algorítmicas₂>. Una vez que se ejecute cualquiera de las dos alternativas se pasa a la siguiente acción que esté inmediatamente después de la etiqueta **Fin-Si**.

Ejemplos:

```
Si A= 0 entonces  
    Escribir 'Valor igual a cero'  
si-no  
    Escribir 'Valor diferente de  
cero'  
Fin-Si
```

```
Si (X = 0) y (Y = 0) entonces  
    Escribir 'Punto en el Origen'  
si-no  
    Escribir 'Punto fuera del  
Origen'  
Fin-Si
```

Equivalencia Funcional entre Condicionales

Una secuencia de condicionales simples puede ser reescrita utilizando condicionales dobles, conservando la misma funcionalidad de las acciones originales. Esto se llama equivalencia funcional, ya que ambos segmentos de acciones tendrán la misma función. Igual se puede hacer con una anidación de condicionales dobles, puede ser reescrita con una secuencia de condicionales simple que conserve su funcionalidad.

Secuencia de Condicionales

Es una secuencia de acciones algorítmicas de decisión, generalmente de un mismo tipo (simple o doble), utilizadas para evaluar las condiciones correspondientes a un problema.

Anidación de Condicionales Dobles

Se dice que se tiene una anidación de condicionales o nido de preguntas, cuando en la rama “si-no” de un condicional doble se coloca otro condicional doble en el cual se aplica el mismo principio.

La anidación de condicionales se utiliza cuando en un problema se requiere evaluar condiciones excluyentes (aquellas que no ocurren simultáneamente). Al estar bien estructurados los condicionales dobles, no es necesario realizar la última pregunta.

Secuencia de Condicionales Simples Reescrita con Condicionales Dobles

Secuencia Original:

:
Si $A = 0$ entonces
 Escribir 'Valor igual a cero'
Fin-Si
Si $A \neq 0$ entonces
 Escribir 'Valor diferente de
cero'
Fin-Si
:

Secuencia Reescrita:

:
Si $A = 0$ entonces
 Escribir 'Valor igual a cero'
si-no
 Escribir 'Valor diferente de
cero'
Fin-Si
:

Anidación de Condicionales Dobles Reescrita con Condicionales Simples:

Secuencia Original:

```
:  
Si V > 0 entonces  
    Escribir 'Valor Positivo'  
si-no  
    Si V = 0 entonces  
        Escribir 'Valor Nulo'  
    si-no  
        Escribir 'Valor Negativo'  
    Fin-Si  
Fin-Si  
:
```

Secuencia Reescrita:

```
:  
Si V > 0 entonces  
    Escribir 'Valor Positivo'  
Fin-Si  
Si V = 0 entonces  
    Escribir 'Valor Nulo'  
Fin-Si  
Si V < 0 entonces  
    Escribir 'Valor Negativo'  
Fin-Si  
:
```

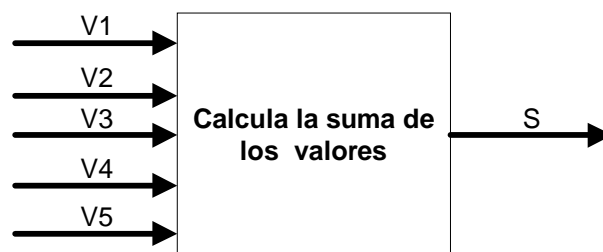
- *Acciones Algorítmicas Cíclicas*

¿Por qué usar una Acción Cíclica? Hay problemas cuya solución requiere repetir varias veces la ejecución de un grupo de acciones. Esto se logra a través de la utilización de las acciones algorítmicas cíclicas, ya que permiten agrupar el conjunto de acciones que se quieren repetir. A través de un ejemplo se presenta la utilización de una acción cíclica:

Ejercicio:

Calcular la suma de 5 valores numéricos.

Diagrama E/S



Entradas:

- V1: numérico real, representa el valor #1.
- V2: numérico real, representa el valor #2.
- V3: numérico real, representa el valor #3.
- V4: numérico real, representa el valor #4.
- V5: numérico real, representa el valor #5.

Salidas:

- S: numérico real, representa la pendiente de la suma de los valores.

Análisis

Como ya se ha comentado en otros ejemplos, la suma no tiene limitantes para ser efectuada, por lo tanto se sumarán los cinco valores y el resultado se almacenará en la variable S.

Algoritmo

A continuación se presentan cuatro versiones del algoritmo que efectúa la suma de los cinco valores:

Suma (versión 1)	Suma (versión 2)	Suma (versión 3)	Suma (versión 4)
Inicio Leer V1 Leer V2 Leer V3 Leer V4 Leer V5 S V1+V2+V3+V4+V5 Escribir S Fin	Inicio Leer V S ← V Leer V S ← S + V Leer V S ← S + V Leer V S ← S + V Leer V S ← S + V Leer V S ← S + V Escribir S Fin	Inicio S ← 0 Leer V S ← S + V Leer V S ← S + V Leer V S ← S + V Leer V S ← S + V Leer V S ← S + V Leer V S ← S + V Escribir S Fin	Inicio S ← 0 C ← 0 Repetir Leer V S ← S + V C ← C + 1 Hasta C = 5 Escribir S Fin

←

¿Y si en vez de 5 valores son 10 ó 100 ó 1000 ?

En la versión 1 se hace uso de las cinco variables de entrada. En función de la pregunta, la versión 2 es una mejora de la anterior, ya que hace uso de una sola variable para representar las entradas y reordena las acciones, permitiendo ver que en la solución se repite la secuencia leer y sumar. La versión 3 mejora la 2, ya que inicializa la variable S y corrobora que la base de la solución es la repetición de la lectura y la suma.

Si en vez de 5 valores fuesen 1000, entonces habría que agregar 995 repeticiones de las acciones que están encerradas en el recuadro color lila, lo cual haría que se tenga un algoritmo muy largo y repetitivo. Esto se puede evitar con la utilización de una acción algorítmica cíclica que permita la repetición del par de acciones del recuadro.

La versión 4 hace uso de la acción **Repetir**, para lo cual se tuvo que agregar una variable (C) que permite contar la cantidad de veces que se pasa por el

ciclo. De éste modo sólo con cambiar el valor presente en la línea **Hasta** (delimitador del ciclo) se ajusta el algoritmo para que suma la cantidad de valores que allí se especifique, sin tener un algoritmo excesivamente largo.

La utilización de acciones algorítmicas cíclicas (o ciclos) requiere que se tenga claro para qué se quiere utilizar, cuándo iniciarlo, cuándo detenerlo. Además, los ciclos dan paso a otros usos de las variables, se verá que pueden ser empleadas como contadores y acumuladores.

Aquí se abordan las acciones cíclicas básicas de la algorítmica: **Repetir** y **Mientras**. Se presenta su sintaxis, semántica y uso, así como la equivalencia entre ambas.

Acción Cíclica Repetir

Sintaxis:

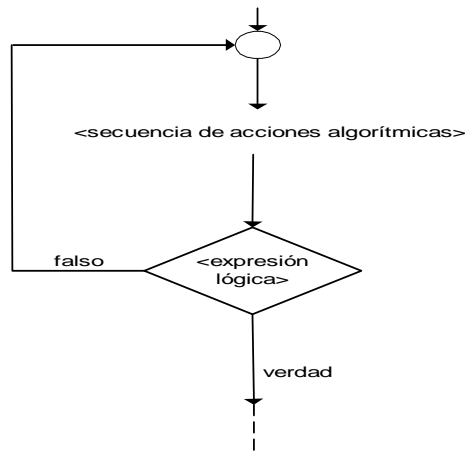
Pseudocódigo:

Repetir

<secuencia de acciones algorítmicas>

Hasta <expresión lógica>

Diagrama de Flujo:



Semántica:

Permite la ejecución de la <secuencia de acciones algorítmicas> repetidas veces hasta que se la evaluación de la <expresión lógica> sea verdad. El ciclo Repetir se ejecuta al menos una vez.

Ejemplos:

```
:  
C ← 0  
Repetir  
    Escribir "Bienvenido"  
    C ← C + 1  
Hasta C = 3  
:
```

```
:  
C ← 0  
A ← 0  
Repetir  
    C ← C + 1  
    A ← A + C  
Hasta C = 5  
Escribir "Suma = ", A  
:
```

Contadores y Acumuladores

Cuando una variable aparece a ambos lados de una asignación y se le suma un valor fijo, se dice que la variable está siendo utilizada como **contador**.

Cuando una variable aparece a ambos lados de una asignación y se le suma un valor no siempre fijo, se dice que la variable está siendo utilizada como **acumulador**.

En los ejemplos anteriores la variable **C es un contador** que se incrementa de uno en uno; y la variable **A es un acumulador**, que se utiliza para ir almacenando la suma de los valores que toma la variable C.

Antes de utilizar un contador y/o un acumulador, debe ser inicializado adecuadamente en un valor conveniente que no altere el resultado que se quiere obtener.

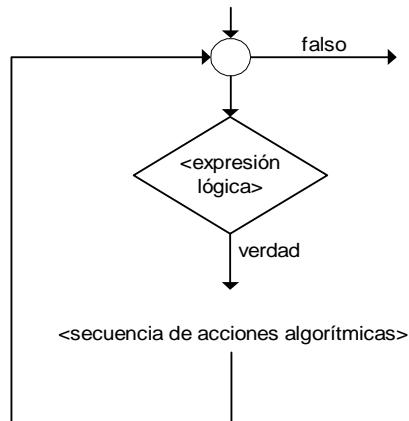
Acción Cíclica Mientras

Sintaxis:

Pseudocódigo:

```
Mientras <expresión lógica>hacer  
    <secuencia de acciones algorítmicas>  
Fin-Mientras
```

Diagrama de Flujo:



Semántica:

La acción cíclica **Mientras** permite la ejecución de la <secuencia de acciones algorítmicas> siempre y cuando la evaluación de la <expresión lógica> sea verdad, al dejarse de cumplir la condición no se vuelve a entrar al ciclo Mientras, este se ejecuta al cero o más veces.

Ejemplos:

: C ← 0 Mientras C < 3 hacer Escribir "Bienvenido" C ← C + 1 Fin-Mientras :	: C ← 0 A ← 0 Mientras C < 3 hacer C ← C + 1 A ← A + C Fin-Mientras Escribir "Suma = ", A :
--	--

Equivalencia Funcional entre Acciones Cíclicas

Al igual que en las acciones condicionales, entre las acciones cíclicas se puede establecer la equivalencia funcional, esto es que un ciclo repetir pueda ser simulado con un ciclo mientras y también viceversa.

El ciclo Repetir se ejecuta al menos una vez, es por eso que al reescribirlo con un ciclo Mientras, antes de éste se coloca la secuencia de acciones que están dentro del Repetir (para realizar un ciclo) y luego se arma el ciclo Mientras con la condición negada del ciclo Repetir (¿por qué?).

2.2.2 Sobre la Educación

2.2.2.1 Teorías Pedagógicas

La Pedagogía tiene como objeto de estudio a la Educación, se encarga de los aspectos relacionados con el hecho educativo, inherente a los humanos como seres individuales y sociales. Dependiendo de la concepción psicológica y de los enfoques que los orienten, los procesos relativos a la educación (enseñanza, aprendizaje, etc.) pueden describirse desde una corriente pedagógica particular, con lo cual se determinan y justifican las acciones a realizar en la práctica educativa.

2.2.2.1.1 La Teoría Conductista

Denominada también conductismo, esta teoría concibe el aprendizaje como un acto mecánico, reduciendo al ser humano al esquema estímulo-respuesta y estudiando su conducta, comportamiento observable o respuesta ante determinados estímulos. Se fundamenta teóricamente en la corriente psicológica homónima. Unos de los mayores exponentes de esta tendencia fue Skinner.

2.2.2.1.2 La Teoría Constructivista

Esta teoría propone una concepción interactiva, donde el proceso educativo se lleva a cabo de forma dinámica, participativa e interactiva. Siendo el conocimiento una auténtica construcción operada por la persona que aprende. En esta teoría el alumno, estudiante o discente tiene un rol protagónico y el profesor o docente un rol de orientador. Dos representantes emblemáticos del constructivismo son: Piaget, desde la perspectiva del individuo y Vygotski, desde una perspectiva social.

2.2.2.2 Enseñanza y Aprendizaje

2.2.2.2.1 La Enseñanza

Desde una óptica tradicional se considera que la esencia de la enseñanza es la transmisión de información mediante la comunicación directa o con la utilización de medios auxiliares, sencillos o complejos.

El proceso de enseñanza consiste fundamentalmente en lograr que en el individuo quede el conocimiento que lo faculte para enfrentar situaciones nuevas de una forma adaptativa, y que paralelamente pueda apropiarse de manera consciente de las situaciones particulares que acontezcan en su entorno.

La enseñanza persigue agrupar a los hechos, clasificarlos, compararlos y descubrir sus regularidades e interdependencias.

2.2.2.2.2 El Aprendizaje

El aprendizaje puede ser considerado como un proceso de naturaleza extremadamente compleja que se caracteriza por la adquisición de un nuevo conocimiento, habilidad o capacidad, que para que sea considerado como tal, debe ser permanente, susceptible de manifestarse en un tiempo futuro y contribuir a la solución de situaciones concretas, incluso diferentes en su esencia a las que motivaron inicialmente el desarrollo del conocimiento, habilidad o capacidad.

2.2.2.2.3 Aprendizaje Significativo

El aprendizaje significativo es el proceso que sucede en el individuo, en el cual su actividad perceptiva lo lleva a incorporar nuevas ideas, hechos, convertidos en nuevo conocimiento en su estructura cognoscitiva. Aprender un contenido implica atribuirle un significado, construir una representación o un modelo mental del mismo.

Es un aprendizaje relevante que se ha logrado interiorizar y retener luego de haber encontrado un sentido teórico o una aplicación real para su vida; este tipo de aprendizaje va más allá de la memorización, ingresando al campo de la comprensión, aplicación, síntesis y evaluación.

2.2.2.2.4 Aprender a Aprender

Para que el individuo alcance el objetivo irrenunciable de aprender a aprender es necesario que desarrolle y aprenda a utilizar estrategias de exploración y descubrimiento, así como de planificación y control del propio proceso de aprendizaje. Su aporte a dicho proceso no se limita a un conjunto de conocimientos, incluye también actitudes, motivaciones, expectativas, cuyo origen está en las experiencias que constituyen su propia historia.

2.2.2.2.5 Elementos del Proceso de Enseñanza y Aprendizaje

Elementos que integran el proceso de enseñanza y aprendizaje:

- Componentes Personales
 - Sujeto que aprende: Alumno, estudiante o discente
 - Sujeto que enseña: Profesor o docente
 - Interacciones: Profesor-Estudiante, Estudiante-Estudiante, Estudiante-Grupo de Estudiantes, Profesor-Grupo de Estudiante
- Componentes no personales
 - Objetivos: ¿Para qué?
 - Contenidos: ¿Que se enseña?
 - Método: ¿Cómo se da el proceso?
 - Forma: ¿Cuáles actividades se llevan a cabo?
 - Medios: ¿Qué recursos se emplean?
 - Evaluación: ¿Cómo se valida o verifica?

2.2.3 Enseñanza de Algoritmos

2.2.3.1 El Proceso de Enseñanza y Aprendizaje de Algoritmos

El proceso de enseñanza y aprendizaje de algoritmos debe concebirse como un ámbito en el cual el principal protagonista es el estudiante y el profesor o docente cumple una función de orientador o facilitador del proceso de aprendizaje. Es el estudiante quien construyen el conocimiento a partir de su participación activa (lee, interactúa, practica, nutre su experiencia personal y académica, y reflexiona sobre ella) e intercambia sus puntos de vista con sus compañeros y el profesor para aprender. La idea es que el estudiante se empodere y se comprometa con un aprendizaje de por vida.

2.2.3.2 Competencias a Desarrollar

- ✓ Razonamiento Lógico
- ✓ Razonamiento Abstracto
- ✓ Resolución de Problemas
- ✓ Capacidad de Observación
- ✓ Capacidad Analítica
- ✓ Reconocimiento de Patrones
- ✓ Capacidad de Sintetizar
- ✓ Capacidad de Inferir
- ✓ Capacidad de Inducir
- ✓ Capacidad Deductiva
- ✓ Capacidad de Toma de Decisiones
- ✓ Coordinación Viso-Manual

2.3 Definición de Términos Básicos

2.3.1 Relacionados con Algoritmos

Acción algorítmica: Es un comando u orden de duración finita y efecto determinado. Cada acción algorítmica se define mediante la indicación de su sintaxis y semántica.

Algoritmia:

Ciencia del cálculo aritmético y algebraico; teoría de los números.

Algorítmica: Área de la Computación que estudia origen, construcción, cualidades y características de los algoritmos, valida las técnicas para su diseño, análisis de su funcionalidad y operatividad en las diversas aplicaciones.

Algoritmo: (1) Es una secuencia ordenada de pasos precisos (sin ambigüedades o contradicciones) que permiten realizar una tarea o resolver un problema específico. (2) Es una solución genérica de un problema particular.

Algoritmo Computacional: Algoritmo que será implementado en un computador.

Asignación: Acción algorítmica elemental, mediante la cual se asigna a una variable el resultado de la evaluación de una expresión.

Constante: Es un objeto algorítmico cuyo valor permanece fijo durante todo el algoritmo. Puede ser utilizada de forma explícita indicando directamente el valor de la con

Dato: Es una abstracción o representación simbólica que proporciona información sobre un hecho, descripción de una propiedad o característica de una persona o de un objeto.

Entrada de Dato: Acción algorítmica elemental que permite la obtención, desde una fuente externa, de los datos a procesar.

Expresión: Es una combinación válida de operandos y operadores, de acuerdo a las reglas que señalen una notación particular (post-fija, infija, etc.).

Identificador: Es una secuencia de caracteres que identifican un objeto algorítmico. Dicha secuencia puede estar integrada por caracteres alfabéticos y/o numéricos y de ser necesario, por algún símbolo especial. La conformación de un identificador está sujeto a las reglas de sintaxis que determine el lenguaje (algorítmico o de programación) a utilizar.

Instrucción: Orden o comando de un lenguaje de programación de computadoras.

Lenguaje de Programación: Es un lenguaje formal diseñado para realizar procesos que han de ejecutarse en una computadora.

Notación Algorítmica: Conjunto de reglas (sintaxis y semántica) que determinan la forma en que debe ser escrito un algoritmo. Dicha notación puede ser gráfica (diagrama de flujo) o no gráfica (pseudo-código).

Objeto Algorítmico: Es una representación de un dato o valor importante para la solución del problema o realización de la tarea o actividad que realiza el algoritmo.

Programa: Secuencia de instrucciones ordenadas de forma lógica para realizar una tarea o actividad, se ejecutan en una computadora. *Software* del computador.

Salida de Dato: Acción algorítmica elemental que permite mostrar los datos procesados.

Secuenciación de acciones: Composición de acciones algorítmicas en un orden determinado para un fin específico. También denominado *secuenciamiento* de acciones.

Semántica: Descripción del significado, actividad y efecto de una acción algorítmica o instrucción de un lenguaje de programación.

Sintaxis: Reglas que define la secuencia correcta de los elementos de una acción algorítmica, en particular; y en general, de una notación algorítmica o de un lenguaje de programación.

Tipo de Dato: Determinación de la naturaleza, numérica o no, que puede tener un objeto algorítmico y define el conjunto de operaciones o cálculos que les pueden ser aplicados, así como su rango valores.

Valor: Expresión de la cantidad que puede tener, en un momento determinado, con relación a su tipo de dato un objeto algorítmico.

Variable: Es un objeto algorítmico referenciable mediante un *identificador* o *nombre*, tiene asociada un tipo de datos y un único valor en un momento determinado, el cual puede cambiar por efecto de una *acción algorítmica*. Esto significa que en un lapso determinado, una variable –como su nombre lo indica- puede tomar diferentes valores.

2.3.2 Relacionados con Educación

Actitud: (1) Disposición de ánimo manifestada de algún modo. (2) Tendencia o predisposición aprendida, más o menos generalizada y de tono afectivo, a responder de un modo bastante persistente y característico, por lo común positiva o negativamente (a favor o en contra), con referencia a una situación, idea, valor, objeto o clase de objetos materiales, o a una persona o grupo de personas.

Aptitud:(1) Capacidad para operar competentemente en una determinada actividad. (2) Carácter o conjunto de condiciones que hacen a una persona especialmente idónea para una función determinada.

Ambiente de Aprendizaje: Espacio educativo, físico o virtual, conformado por un conjunto de recursos, herramientas que posibilitan la interacción didáctica del proceso de enseñanza y aprendizaje.

Aprendizaje: Es el proceso a través del cual se adquieren o modifican habilidades, destrezas, conocimientos, conductas o valores como resultado del estudio, la experiencia, la instrucción, el razonamiento y la observación. Este proceso puede ser analizado desde distintas perspectivas, por lo que existen distintas teorías del aprendizaje.

Aprendizaje Basado en Problemas: También conocido como aprendizaje basado en proyectos, se basa en varias corrientes pedagógicas especialmente en el constructivismo. Los estudiantes de manera autónoma, pero guiados por el docente, deben encontrar la respuesta a una pregunta, solución a un problema o ejecución de un proyecto, de forma que para conseguir realizarlo correctamente, tengan que buscar, entender e integrar y aplicar los conceptos básicos del contenido del problema así como los

relacionados. De este modo, por sí mismos consiguen elaborar un diagnóstico de las necesidades de aprendizaje, construir el conocimiento de la materia y trabajar cooperativamente. También incluye el desarrollo del pensamiento crítico como parte constitutiva del proceso de enseñanza y aprendizaje.

Aprendizaje Colaborativo: Aprendizaje que se produce en el ámbito educativo, mediante la interacción con los demás integrantes de esa comunidad. Está determinada por la comunicación y el contacto interpersonal con los docentes y los compañeros de grupo dando la posibilidad de enriquecer los conocimientos, de ampliar perspectivas y del desarrollo personal cada integrante.

Aprendizaje Por Descubrimiento: Este proceso consiste en la adquisición de conceptos, principios o contenidos a través de un método de búsqueda activa, sin una información inicial sistematizada del contenido a aprender.

Aprendizaje Por Ensayo y Error: Proceso de aprendizaje en el cual el sujeto de aprendizaje se enfrenta a una nueva situación sin saber cuál es la respuesta correcta y comienza emitiendo una variada gama de ellas, hasta que casualmente da con la respuesta correcta, recibiendo posteriormente un reforzamiento positivo.

Aprendizaje Significativo: Aprendizaje relevante que el estudiante ha logrado interiorizar y retener luego de haber encontrado un sentido teórico o una aplicación real para su vida; este tipo de aprendizaje va más allá de la memorización, ingresando al campo de la comprensión, aplicación, síntesis y evaluación.

Aprendizaje social: Se entiende por aprendizaje social al conjunto de aprendizajes que hacen referencia a conductas específicas y directamente ligadas a la vida social, como hábitos sociales, actitudes, valores.

Aprender: Proceso dirigido a adquirir el conocimiento de algo por medio del estudio o de la experiencia, o a la construcción de nuevas competencias, habilidades o destrezas.

Capacidad: Aptitud que se tiene en una determinada disciplina o práctica.

Conductismo: Teoría que concibe el aprendizaje como algo mecánico, deshumano y reduccionista. Defiende el empleo de procedimientos estrictamente experimentales para estudiar el comportamiento observable (la conducta) y niega toda posibilidad de utilizar los métodos subjetivos. Su fundamento teórico está basado en el esquema estímulo-respuesta, siendo ésta el resultado de la interacción entre el organismo que recibe el estímulo y el medio ambiente.

Constructivismo: Teoría que propone un paradigma donde el proceso de enseñanza y aprendizaje se percibe y lleva a cabo como un proceso dinámico, participativo e interactivo, de modo que el conocimiento sea una auténtica construcción operada por la persona que aprende.

Competencia: Aptitud demostrada individualmente para utilizar el saber práctico, la capacidad profesional, las cualificaciones o los conocimientos teóricos para realizar una función determinada o afrontar situaciones y requisitos profesionales tanto habituales como cambiantes.

Destreza: Capacidad o habilidad para realizar algún trabajo, principalmente relacionado con trabajos físicos o manuales.

Didáctica: Parte de la Pedagogía que estudia los procesos y elementos existentes en el proceso de enseñanza y el aprendizaje, se ocupa de las técnicas y métodos de enseñanza.

Educación: Promover el desarrollo intelectual, emocional, social y cultural del educando, es decir, desarrollar sus potencialidades psíquicas y cognitivas propias del educando.

Enseñanza: Proceso de transmisión de conocimientos, técnicas, normas, y/o habilidades. Se basa en diversos métodos, realizado a través de una serie de instituciones, y con el apoyo de una serie de materiales. Comprende la interacción de los siguientes elementos: docente o facilitador, discente, alumno o estudiante, objeto o tema de conocimiento y entorno educativo.

Enseñar: Proceso dirigido a guiar, orientar, ayudar o asesorar los procesos individuales de aprendizaje.

Estrategia de Aprendizaje: Conjunto de actividades, técnicas, recursos y medios, planificados de acuerdo a las necesidades de los alumnos; teniendo como objetivo facilitar la adquisición del conocimiento; así como también, hacer más efectivo el proceso de aprendizaje.

Estrategia de Enseñanza: Procedimiento que el docente utiliza con el fin de orientar, guiar y ayudar a los alumnos a realizar su actividad adecuadamente, para poder lograr los objetivos de aprendizaje que se le propongan.

Habilidad: (1) Gracia y destreza en ejecutar algo. (2) Aptitud innata, talento, destreza o capacidad que ostenta una persona para llevar a cabo con éxito, determinada actividad, trabajo u oficio. Ciertas habilidades sólo se adquieren mediante el aprendizaje, requiriendo para su potenciación e incremento de la puesta en práctica y el entrenamiento.

Propedéutica: Enseñanza preparatoria para el estudio de una disciplina da nombre a la acumulación de conocimientos y disciplinas que son necesarios para abordar y entender cualquier materia. El término proviene del

griego pró (“antes”) y paideutikós (“referente a la enseñanza”). Permite sistematizar los procedimientos y técnicas que se requieren para concretar el desafío.

Razonamiento: Consiste en la facultad que permite resolver problemas, extraer conclusiones y aprender de manera consciente de los hechos, estableciendo conexiones causales y lógicas necesarias entre ellos.

Razonar: Exponer razones para explicar o demostrar algo.

Unidad de Enseñanza: Conjunto organizado de objetivos y estrategias que da sentido unitario y eficaz al acto didáctico mediante la consideración integrada, secuencial y estructurada de los contenidos, metodología, actividades y recursos didácticos.

2.4 Marco Conceptual de Proyectos

En esta sección se presentan de forma general los conceptos básicos de proyectos, sus características y aspectos de su organización.

2.4.1 Aspectos Generales

2.4.1.1 Qué es un Proyecto

Un proyecto es un esfuerzo temporal realizado para llevar a cabo la creación de producto, servicio o resultado único. La mayor parte de los proyectos se emprenden para crear un resultado duradero. Un proyecto tiene un principio y un final definidos. El final se alcanza cuando se logran los objetivos del proyecto, cuando se termina el proyecto porque sus objetivos no se cumplirán o no pueden ser cumplidos, o cuando ya no existe la necesidad que dio origen al proyecto.

2.4.1.2 Características

- Tiene un objetivo definido;
- Es único y de carácter temporal;
- Tiene un alcance definido a través de la definición de requisitos y requerimientos específicos.
- Cumple con el alcance para el que fue creado;
- Requiere de organización y manejo de recursos (personal, material, financiero y tiempo);

- Se realizan de forma progresiva mediante la definición de pasos, etapas o fases;
- Según su escenario de desarrollo pueden tener un mayor o menor grado de incertidumbre.

2.4.1.3 Tipos de Proyecto

Los proyectos pueden ser clasificados con base a diversos criterios, la más extendida clasificación es en cuanto a la generación de rentabilidad, dando paso a dos grandes tipos:

- **Proyectos de Inversión:** Son proyectos productivos que buscan generar rentabilidad económica para obtener ganancias en dinero.
- **Proyectos Sociales:** Su realización tiene como finalidad generar un impacto positivo en sus beneficiarios, lo cual no necesariamente se mide en dinero. Son llevados a cabo por el Estado, organismos multilaterales o empresas para cumplir con sus objetivos de responsabilidad social.

2.4.1.4 Ciclo de Vida de un Proyecto

Independientemente de la complejidad y el tamaño del proyecto, su ciclo de vida puede configurarse dentro una estructura genérica. Para el PMBOK (2013) ese ciclo está conformado por: Inicio del proyecto; organización y preparación; ejecución del trabajo y cierre del proyecto; como se ilustra en el siguiente gráfico:

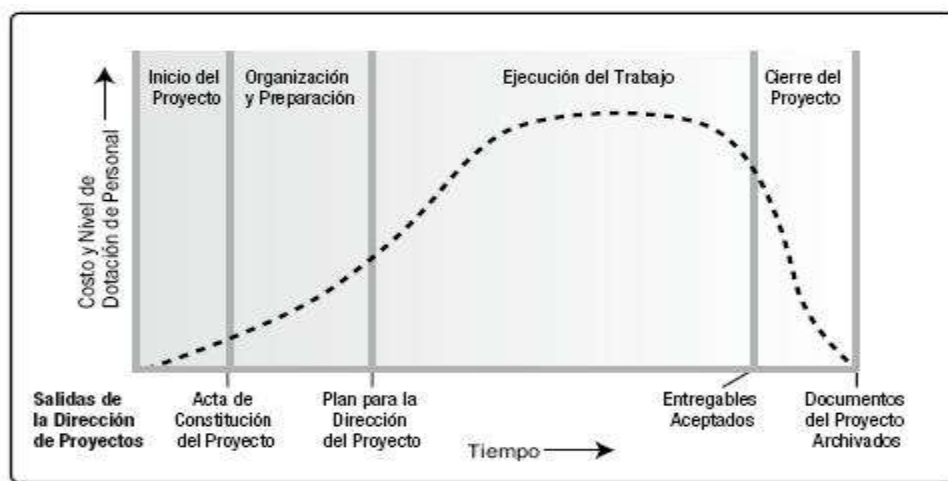


Figura 5: Estructura Genérica del Ciclo de Vida del Proyecto Y, Costo y Nivel de Dotación de Personal. Fuente: PMI (2013)

2.4.1.5 Fases de un Proyecto

Según el PMI (2013), una fase del proyecto está conformada por un conjunto de actividades relacionadas de manera lógica, que culmina con la finalización de una o más entregas del producto o productos elaborados en dicha fase. Un proyecto se puede ser dividido en una cantidad determinada de fases que normalmente tienen una duración o esfuerzo diferentes. Las fases del proyecto suelen completarse en forma secuencial, pero también pueden superponerse en determinadas circunstancias de los proyectos.

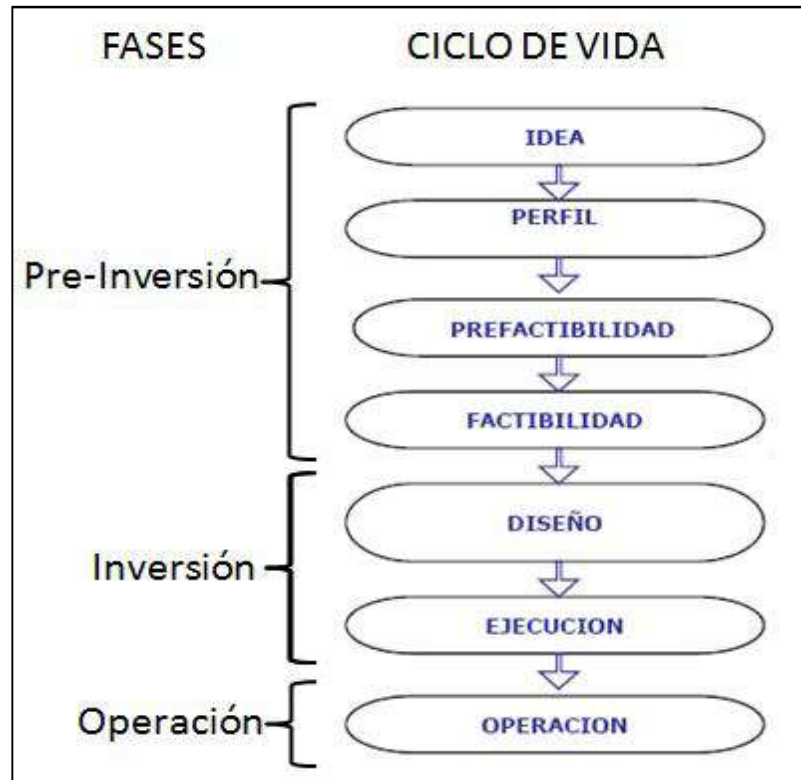


Figura 6: Ciclo de Vida Genérico y Fases de un Proyecto de Ingeniería.
Fuente: Adaptación basada en Albis (2013)

2.4.1.6 Manejo o Dirección de un Proyecto

Según el PMBOK (2013), la dirección de proyectos es la aplicación de los conocimientos, habilidades, herramientas y técnicas a las actividades que se

tiene que realizar en el proyecto para cumplir con los requisitos y requerimientos del mismo.

2.4.2 Metodologías de Proyectos

2.4.2.1 PMBOK del PMI

PMBOK es el acrónimo de *Project Management Body Of Knowledge*, creado por el PMI, siglas en inglés de *Project Management Institute*. Es una guía para el manejo de proyectos que contiene un compendio de estándares y mejores prácticas organizadas en trece áreas de conocimientos y cinco grupos de procesos.

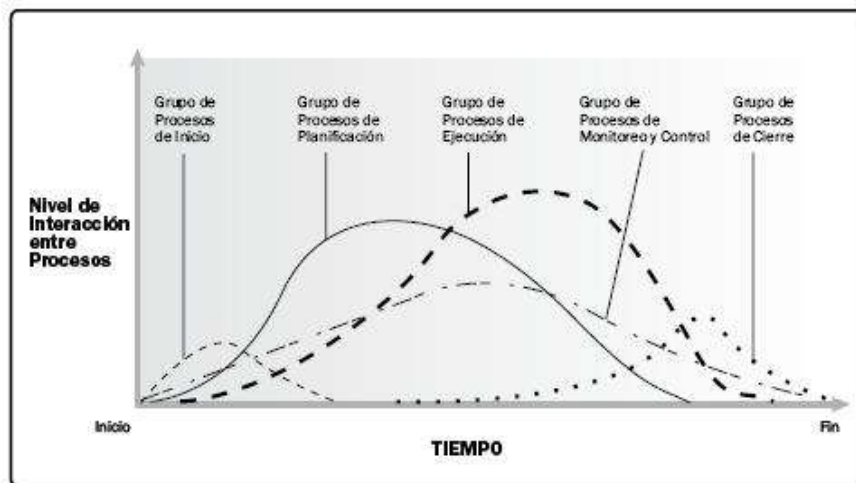


Figura 7: Interacción de los Grupos de Procesos de un Proyecto.
Fuente: PMI (2013).

El PMBOK es una guía ampliamente utilizada para la definición, planeación, ejecución y gestión de programas, portafolios de proyectos, así como de proyectos individuales; ya que proporciona un marco estandarizado para la realización de las actividades propias de fase o etapa del proyecto desde su inicio hasta su culminación o cierre.

2.4.2.2 Enfoque del Marco Lógico

También conocida como Metodología del Marco Lógico (MML), según lo cita Nardi (2006), el Banco Interamericano de Desarrollo lo define así: “*el marco lógico es una herramienta para facilitar el proceso de conceptualización, diseño, ejecución y evaluación de proyectos*”.

Además utilizarse para la formulación de proyectos, puede emplearse para su planeación, ejecución y monitoreo, de acuerdo a Ortegón (2005).

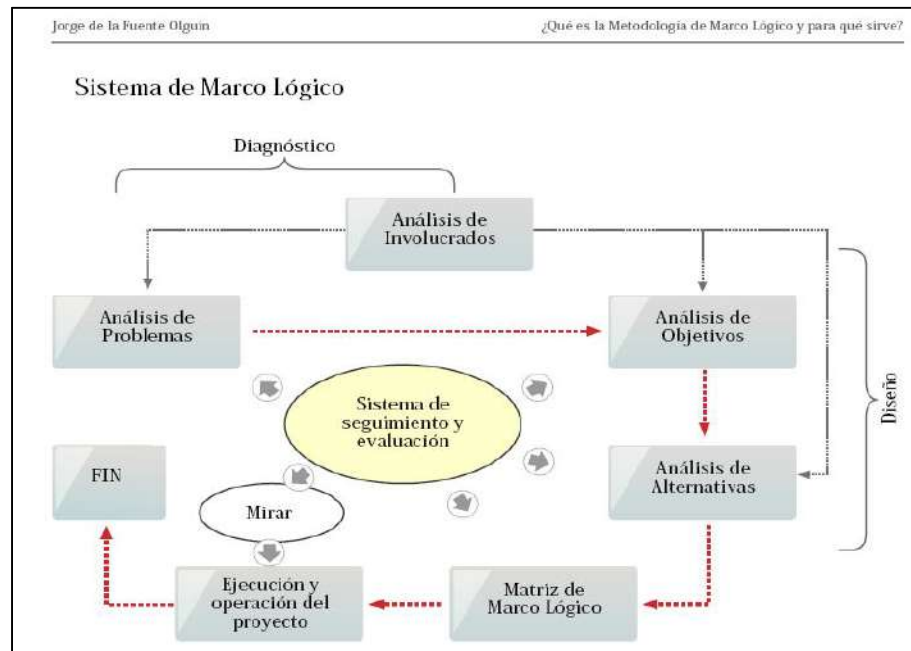


Figura 8: Sistema de Marco Lógico. Fuente: Ortegón (2005).

La metodología o enfoque marco lógico provee un conjunto de métodos, técnicas y herramientas que apoyan la generación y gestión de proyectos, Ortegón (2005), entre las que destacan: a) técnicas para la identificación de problemas; b) elaboración de mapas de los involucrados; c) métodos del árbol de problemas y árbol de objetivos; d) análisis y selección de las alternativas para definir las posibles soluciones. Las anteriores se utilizan para la definición y selección del problema a trabajar en el proyecto.

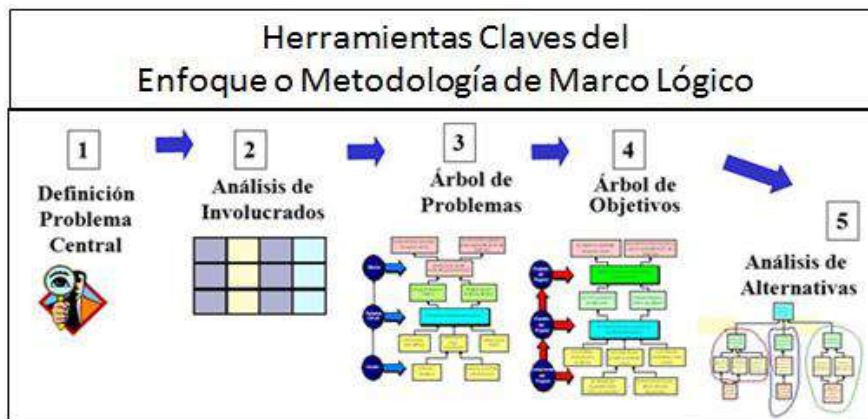


Figura 9: Herramientas Claves del Enfoque de Marco Lógico.
Fuente: Adaptación de http://images.slideplayer.es/17/5430546/slides/slide_4.jpg.

2.5 Marco Legal

- ***Derecho a la Educación: Artículo 102. Constitución de la República Bolivariana de Venezuela***, con la Enmienda No.1, de fecha 15 de febrero de 2009. Gaceta Oficial No 5.908 Extraordinaria. Caracas, 19 de febrero de 2009.
- ***Principios y valores rectores de la educación; Educación y Cultura y El Estado Docente***: Artículos 3, 4 y 5, respectivamente. Ley Orgánica de Educación. Gaceta Oficial No 5.929 Extraordinaria. Caracas, 15 de agosto de 2009.
- ***Promoción y estímulo de Cultores y Cultoras para la ciencia, la tecnología y la innovación***: Artículo 35. Ley Orgánica de Ciencia, Tecnología e Innovación (LOCTI). Caracas, 16 de diciembre de 2010.
- El Programa Nacional de Formación en Informática (PNFI) permite consolidar la formación crítica productiva del profesional en Informática propiciando la formación humanista, sociopolítica, comprometido con los cambios económicos, sociales, políticos, culturales, tecnológicos del país. Documento rector del PNFI (2008).
- Normativa General de los Estudios de Postgrado para las Universidades e Institutos Debidamente Autorizados por el Consejo Nacional de Universidades.

CAPÍTULO III: MARCO METODOLÓGICO

En este capítulo se definen el enfoque y tipo de investigación, se describen el diseño, unidad de análisis, las técnicas y herramientas de recolección de datos utilizados para su realización. También se presentan las fases definidas para su desarrollo y algunos aspectos administrativos.

3.1 Tipo de Investigación

En este trabajo la aproximación al conocimiento se realiza de un modo inductivo e interpretativo, por lo cual el enfoque empleado en la investigación es de carácter **cuantitativo**, que por el grado de profundidad con que se aborda, cuya finalidad es describir con suficiente detalle las características relevantes de la situación de estudio, esta es una **investigación descriptiva**. Se realiza un análisis sistemático para entender la problemática planteada mediante la descripción de sus elementos y determinación de sus posibles causas y efectos; se describen el marco contextual y teórico de la investigación.

Con base al objetivo general a lograr en este trabajo: definición de las bases funcionales para realizar el diseño de un curso propedéutico para la enseñanza de algoritmos, se trata de una **investigación aplicada** porque se quiere obtener un resultado concreto mediante el estudio y aplicación de teorías ya establecidas, dirigido fundamentalmente a efectuar un aporte o solución para el problema planteado.

3.2 Diseño de Investigación

Por las estrategias utilizadas para la recolección o recopilación de datos e información, es una investigación con diseño **documental** y **de campo no experimental**.

La información y los datos necesarios para la realización de este trabajo de investigación, se obtuvo a través de los siguientes tipos de diseño:

- Documental o Bibliográfico: Revisión de fuentes documentales, impresas y digitales, antecedentes de la investigación y para la conformación de la

sustentación conceptual o marco teórico de la investigación. Constituida por trabajos especiales de grado (de licenciatura, especializaciones, maestrías y doctorados); libros, artículos y disertaciones.

- De Campo: Recolección de datos de fuentes primarias, en este caso, de los profesores del CUFM que integran la unidad curricular “Algorítmica y Programación” del PNFI.

3.3 Unidad de Análisis

Según Hernández Sampieri, Fernández-Collado y Baptista (2006, p.237), la unidad de análisis comprende los sujetos, objetos, sucesos o comunidades de estudio; refiere el interés sobre el qué o quiénes se estudia, lo cual a su vez depende del planteamiento de la investigación y su alcance.

Con base al planteamiento anterior la unidad de análisis de este trabajo de investigación está conformada por la Coordinación del PNFI del CUFM, los profesores y contenidos de la unidad curricular “Algorítmica y Programación”.

3.4 Técnicas Y Herramientas de Recolección de Datos

A continuación se describe el conjunto de técnicas y herramientas utilizado para la recolección de datos y su procesamiento:

- Entrevistas no estructuradas: Conversación directa y abierta sobre la problemática de la enseñanza y aprendizaje de algoritmos, con profesores de la unidad curricular Algorítmica y Programación.
- Recolección de información durante la realización de un foro para la presentación de una propuesta de reorganización de contenidos de los módulos Programación I y Programación II, asignaturas de la ya referida unidad curricular.
- Juicio experto: Solicitud de su opinión sobre la investigación propuesta así como su valoración sobre la enseñanza de algoritmos, a dos de los profesores con más experiencia docente en la enseñanza de programación.

- Análisis documental: De los antecedentes para un mayor entendimiento del problema y la elaboración del marco contextual y teórico de la investigación.
- Análisis de contenido: Para la creación de la propuesta de un curso propedéutico para la enseñanza y aprendizaje de algoritmos, se debió revisar los diferentes documentos y programas (PNF, PNFI, entre otros) existentes para los estudios universitarios del tema seleccionado.
- Herramientas claves de la metodología del marco lógico: Definición del problema central, análisis de involucrados, árbol de problema, árbol de objetivos y análisis de alternativas de solución.

3.5 **Ámbito del Dominio de Conocimientos**

De acuerdo planteamiento del problema, el alcance definido y a las diferentes áreas de conocimientos involucradas en la elaboración de este trabajo, el ámbito del dominio de conocimientos de la investigación está determinado por la intersección de las siguientes áreas: Algorítmica y Programación, Educación y Proyectos; lo cual se representa con la siguiente figura:

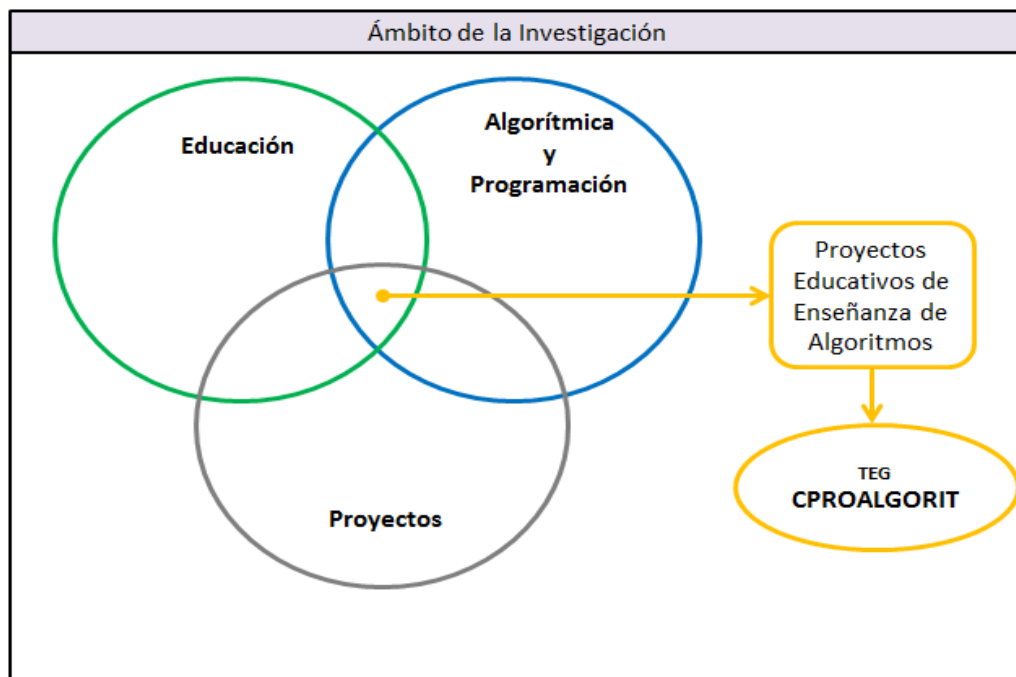


Figura 10: Dominios de conocimientos del ámbito de la investigación.
Fuente: Elaboración propia.

3.6 Organización de la Investigación

Este trabajo de investigación es un proyecto, por lo que requirió de una organización y definición de sus actividades, las cuales se agruparon de forma lógica en las siguientes fases secuenciales:

- *Fase Exploratoria*

Consistió en la selección del problema, formulación de la investigación y definición de los objetivos a desarrollar; también se elaboró el contexto institucional del trabajo. Se realizó la investigación documental y revisión de los trabajos en cuanto a los antecedentes, relación temática, teorías pedagógicas y proyectos educativos; definiendo los dominios de conocimiento o ámbito de esta investigación. Se establecieron los elementos constitutivos de la investigación y de la metodología a utilizar.

- *Fase Analítica*

Esta fase se realizó en dos partes: la primera, para finalizar la recolección de información y la definición de las técnicas y herramientas a aplicar; y en la segunda, se procedió a procesar la información recabada, clasificándola analizándola según el dominio de conocimiento al que pertenece. Se analizaron las teorías educativas, pedagógicas y didácticas relacionadas con este trabajo de investigación; la aplicación de los elementos metodológicos de proyectos; y el contenido de la unidad curricular “Algorítmica y Programación”, así como la temática relacionada con algoritmos.

- *Fase Resolutiva*

Es la fase más productiva de la investigación ya que se trabajó en función del logro de los objetivos, a fin de obtener los resultados visionados: Definición de las Bases Funcionales y formulación de la propuesta de solución: Diseño de un Curso Propedéutico para la Enseñanza de Algoritmos y Elaboración de un Plan Preliminar para su Desarrollo e Implementación.

- *Fase de Cierre*

Consistió en la culminación de la escritura del documento del trabajo de investigación para su entrega formal a las autoridades académicas y la elaboración de la presentación del trabajo.

3.7 Aspectos Administrativos

3.7.1 Recursos: humanos, materiales y financieros

Tabla 1: Recursos Humanos.

Recursos Humanos	Cantidad
Tutor Académico UMA	01
Estudiante Trayecto Inicial PNFI	45
Docente de Algoritmos y Programación del PNFI	05
Auxiliar Docente de Algoritmos y Programación del PNFI	03
Coordinador PNFI	01
Jefe División Académica del CUFM	01

Tabla 2: Recursos Materiales.

Recursos Materiales	Cantidad
Resma de Papel Carta	04
Computador Portátil	01
Internet	01
Impresora	01
Cartuchos de Impresora	03

Tabla 3: Presupuesto Estimado.

PRESUPUESTO ESTIMADO			
Descripción	Cantidad	Precio Unitario BsF	Precio Total BsF
Resma de Papel Carta	5	6.000,00	30.000,00
Computador Portátil	1	400.000,00	400.000,00
Internet (mensual)	18	1.500,00	27.000,00
Impresora	1	200.000,00	200.000,00
Cartuchos de Impresora	3	60.000,00	180.000,00
Gastos de Transporte (mensual)	18	16.000,00	288.000,00
Gastos de Alimentación (mensual)	18	80.000,00	1.440.000,00
Total			2.565.000,00

3.7.2 Cronograma de Actividades

CRONOGRAMA DE ACTIVIDADES																															
#	Actividad	Semana-Mes-Año																													
		oct-16					nov-16					dic-16					ene-17					feb-17					mar-17				
		1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	
1	Elaborar, validar y aplicar encuesta para determinar el grado de conocimientos básicos y habilidades en análisis y razonamiento lógico-matemático de los estudiantes del trayecto inicial del PNFI					*	*	*	*	*	*																				
2	Preparar y aplicar la entrevista para identificar las estrategias metodológicas, recursos didácticos y medios tecnológicos que utilizan los docentes en el proceso enseñanza y aprendizaje de algoritmos					*	*	*	*	*	*																				
3	Procesamiento de encuestas y entrevistas																*	*	*												
4	Selección de las estrategias pedagógicas y enfoques didácticos																*	*	*												
5	Diseño de la propuesta del curso propedéutico CPROALGORIT																*	*	*	*	*	*									
6	Elaborar plan propuesto para la implementación del curso propedéutico CPROALGORIT																				*	*	*	*							
7	Elaborar el documento del TEG																*	*	*	*	*	*	*	*	*	*	*	*	*		

Figura 11: Cronograma Estimado de Actividades.

CAPÍTULO II: MARCO ORGANIZACIONAL

4.1 Colegio Universitario “Francisco de Miranda”

4.1.1 Reseña Institucional

El Colegio Universitario "Francisco de Miranda" (CUFM) es una institución del Estado Venezolano, fue creado por Decreto N° 1620 el 20 de febrero de 1974, firmado por el Dr. Rafael Caldera de conformidad con lo dispuesto en el Parágrafo Único del Artículo 10º de la Ley de Universidades y los Artículos 2º y 4º del Reglamento de Institutos y Colegios Universitarios.

La sede central está ubicada en la Esquina de Mijares, en la parroquia Altavista de Caracas. Allí funciona la Dirección, el Consejo Directivo y las Divisiones Académicas, tiene como vecinos al Banco Central de Venezuela, la hermosa Plaza "Juan Pedro López", el Ministerio del Poder Popular para la Educación, y las iglesias Las Mercedes y Santa Capilla.

La institución a lo largo de su historia ha impartido carreras cortas de 3 años (6 semestres) en el área de Técnico Superior Universitario, entre las carreras que destaca son: TSU en Administración Mención Empresas Energéticas (Hidrocarburos), TSU en Administración Mención Banca y Finanzas, TSU en Administración Mención Recursos Humanos, TSU en Educación Integral, TSU en Informática, TSU en Contaduría Pública y por último TSU en Administración Mención Transporte y Distribución de Bienes.

A partir del año 2009, en el marco de la transformación universitaria y al igual que la mayoría de los Colegios e Institutos Universitarios de toda Venezuela, el CUFM comienza a ofertar carreras largas impartiendo los diferentes Programas Nacionales de Formación (PNF): Ingeniería Informática, Licenciatura en Administración, Licenciatura en Contaduría Pública.

4.1.2 Visión

Ser una institución universitaria de excelencia en el plano docente, de investigación, de extensión y de gestión. En estrecha correspondencia con la

sociedad del conocimiento que exige el siglo XXI y con los valores de equidad y justicia que plantea la Constitución de la República Bolivariana de Venezuela en aras de la construcción de una sociedad independiente y democrática.

4.1.3 Misión

El Colegio Universitario "Francisco de Miranda", es una institución de educación del Estado venezolano cuya misión es formar profesionales universitarios competentes con el área técnico profesional, con gran sensibilidad social e identidad nacional y latinoamericana que se incorporen al proceso de transformación del país para aportar soluciones. Así como generar y difundir conocimientos y realizar acciones hacia la comunidad local, regional, nacional e internacional para contribuir a mejorar la calidad de vida.

4.1.4 Valores

Cada uno de los actores, que conforman nuestra comunidad universitaria, desarrolla una actividad mancomunada que tiene como valores:

- La Equidad
- La Participación
- La Solidaridad
- El Respeto
- La Calidad
- La Cooperación
- La Identidad Nacional

4.2 Programa Nacional de Formación en Informática

4.2.1 Programas Nacionales de Formación

4.2.1.1 Definición

Conforme a la Resolución 2963 del 13 de mayo de 2008, los programas nacionales de formación (PNF) son el conjunto de actividades académicas, conducentes a títulos, grados o certificaciones de estudios de educación

universitaria, creados por iniciativa del Ejecutivo Nacional, a través del Ministerio del Poder Popular para la Educación Universitaria, diseñados con la cooperación de instituciones de educación universitaria nacionales, atendiendo a los lineamientos del Plan de Desarrollo Económico y Social de la Nación, para ser administrados en distintos espacios del territorio nacional. Conforme a esta Resolución, la creación de cada PNF se realiza mediante acto administrativo, dictado por el Ministerio del Poder Popular para la Educación Universitaria y este mismo organismo tiene la potestad de autorizar la gestión de los PNF a distintas instituciones de educación universitaria, señalando el ámbito geográfico de dicha autorización.

4.2.1.2 Características

Los Programas Nacionales de Formación tienen como características comunes:

1. La **formación humanista** como aspecto de vital importancia para la formación integral del futuro y la futura profesional, sustentada en la integración de contenidos y experiencias dirigidas a la formación en el ejercicio de la ciudadanía democrática, la solidaridad, la construcción colectiva y la acción profesional transformadora con responsabilidad ética y perspectiva sustentable.
2. La **vinculación con las comunidades y el ejercicio profesional a lo largo de todo el trayecto formativo**; el abordaje de la complejidad de los problemas en contextos reales con la participación de actores diversos; la consideración de la multidimensionalidad de los temas y problemas de estudio; así como el trabajo en equipos interdisciplinarios y el desarrollo de visiones de conjunto, actualizadas y orgánicas de los campos de estudio, en perspectiva histórica, y apoyadas en soportes epistemológicos coherentes y críticamente fundados.
3. La **conformación de los ambientes educativos** como espacios comunicacionales abiertos, caracterizados por la libre expresión y el debate de las ideas, el respeto y la valoración de la diversidad, la multiplicidad de fuentes de información, la integración de todos los

participantes como interlocutores y la reivindicación de la reflexión como elementos indispensables para la formación, asociados a ambientes de formación y prácticas educativas ligados a las necesidades y características de las distintas localidades que propicien el vínculo con la vida social y productiva.

4. La **participación activa** y comprometida de los estudiantes en los procesos de creación intelectual y vinculación social, relacionados con investigaciones e innovaciones educativas vinculadas con el perfil de desempeño profesional y conducentes a la solución de los problemas del entorno, en consideración de sus dimensiones éticas, políticas, culturales, sociales, económicas, técnicas y científicas, garantizando la independencia cognoscitiva y la creatividad de los estudiantes.
5. **Modalidades curriculares flexibles**, adaptadas a las distintas necesidades educativas, a las diferentes disponibilidades de tiempo para el estudio, a los recursos disponibles, a las características de geohistóricas y al empleo de métodos de enseñanza que activen los modos de actuación del futuro profesional.
6. La **definición de sistemas de evaluación** que promuevan el aprendizaje, la reflexión y el mejoramiento, considerando los distintos actores y aspectos del quehacer educativo y valorando su impacto social.
7. La promoción, el reconocimiento y la **acreditación de experiencias** formativas en distintos ámbitos.

4.2.2 Programa Nacional de Formación en Informática

4.2.2.1 Conformación

En el marco de la Resolución N° 2.963, se constituyó la Comisión Técnica Interinstitucional del Programa Nacional de Formación en Informática (CTPNFI), promulgada por el MPPES, el 22 de Mayo del año 2008 a través del Vice Ministerio de Políticas Académicas, quien fue el órgano encargado de su ejecución, según lo descrito en el Artículo 14.

Dicha Comisión tuvo como objetivo la construcción colaborativa y consensuada del Programa Nacional de Formación en Informática (PNFI),

con las características descritas para los PNF. Estuvo conformada por representantes de las siguientes instituciones para la época referida: Ministerio del Poder Popular para la Educación Superior (MPPES) como ente coordinador, Institutos Universitarios de Tecnología (IUT), Colegios Universitarios (CU), Universidad Nacional Abierta (UNA) y Universidad Bolivariana de Venezuela (UBV), además de representantes de los Ministerios del Poder Popular para las Telecomunicaciones y la Informática y para la Ciencia y Tecnología.

4.2.2.2 Componentes

1. Fundamentación del Programa: se realizó una investigación documental conjuntamente con un proceso de análisis reflexivo sobre las bases fundamentales para la construcción del programa, atendiendo a los requerimientos del Plan de Desarrollo Económico y Social de la Nación 2007-2013, el Plan Nacional de Ciencia, Tecnología e Innovación 2005-2030 y los lineamientos del Plan de Telecomunicaciones e Informática y Servicios Postales 2007-2013, articulado a los lineamientos de UNESCO y otros organismos internacionales y nacionales dedicados a la formación universitaria del talento humano especializado en la informática.
2. Perfiles de Egreso: la comisión realizó un análisis de las necesidades de los profesionales en el área de informática de las diferentes regiones del país y de los requerimientos planteados por las instituciones directamente vinculadas al área.
3. Estructura Curricular del Programa: se desarrolló considerando el proyecto socio-tecnológico como núcleo central del PNFI, la formación crítica transformadora como eje transversal y longitudinal y las áreas de saberes contentivas de las unidades de formación.
4. Matriz Curricular y Contenidos Sinópticos: se efectuaron mesas de trabajo conformadas por áreas de saberes.
5. Perfil de los Profesores-Asesores: definidos a partir de las áreas de saberes y los ejes longitudinales del PNFI. Valorada su disposición a

compartir la construcción de conocimientos desde una perspectiva de intercambio horizontal, con una visión humanista, ecológica e integral.

6. Criterios de Administración del Programa: basado en el principio de la no exclusión, se define: la modalidad de estudio, horarios, materiales educativos requeridos e infraestructura tecnológica-académica necesaria para su administración.
7. Criterios de Implantación del Programa: se fundamenta en los principios de inclusión, equidad, acción sistémica, flexibilidad, diversidad, municipalidad y territorialidad. Se definen las políticas de inclusión y accesibilidad, los sistemas de ingreso, permanencia y egreso, así como el sistema de apoyo y mejoramiento de la calidad de vida estudiantil y personas con discapacidad.

CAPÍTULO V: RESULTADOS

5.1 Análisis de la Situación Problemática

El estudio o análisis de la problemática planteada en el capítulo I, se realizó mediante la aplicación de pasos seleccionados de la Metodología Marco del Lógico, con base a una adaptación hecha de los pasos presentados por Nardi (2006) y Ortegón y otros (2005); los cuales se resumen en los siguientes:

- 1) Origen del Proyecto
- 2) Identificación y Análisis de los Involucrados
- 3) Identificación y Análisis del Problema
- 4) Análisis de los Objetivos
- 5) Análisis y Selección de Alternativas de Solución

5.1.1 Origen del Proyecto

Descripción del planteamiento del problema detectado, situación problemática o necesidad a resolver o iniciativa de mejora.

En el Capítulo I se expuso la situación problemática a estudiar en este trabajo, el cual tiene como objetivo general definir las bases funcionales para diseñar un curso propedéutico para la enseñanza de algoritmos.

5.1.2 Identificación y Análisis de Involucrados

Este proceso se realiza en dos partes: (1) se identifican a los involucrados o relacionados del proyecto; tiene en cuenta a las personas, a los grupos, instituciones y organizaciones, a quiénes interesa o afecta el problema; y (2) se analizan los involucrados, se describen y clasifican de acuerdo a su grado de participación con el proyecto, se priorizan y se elabora una tabla resumen de ellos.

A continuación se presentan los pasos del proceso y los resultados obtenidos por su aplicación:

Tabla 4: Pasos para Identificar y Analizar a los Involucrados.

IDENTIFICACIÓN Y ANÁLISIS DE INVOLUCRADOS	
PASO	DESCRIPCIÓN
a. Identificación de involucrados.	✓ Elaborar un listado de todos participantes o involucrados directos e indirectos o relacionados que inciden en aspectos del proyecto, o que pueden ser afectados positiva o negativamente.
b. Clasificar a los involucrados mediante el establecimiento de categorías.	✓ Identificar los participantes o involucrados, sea individuales o colectivos, grupos, gremios y similares, y con esta base establecer categorías para el análisis.
c. Describir a los involucrados.	✓ Especificar claramente la relación de los involucrados, su influencia, los intereses o dificultades, o cómo afecta el proyecto a cada individuo o grupo.
d. Analizar a los involucrados.	✓ Valorar e interpretar a todos los involucrados para definir qué tanto y cómo pueden ser incorporados en el proyecto.
e. Priorización de los involucrados	<p>✓ Con base a los resultados del análisis, fijar una prioridad a cada involucrado que indique cual es su grado de participación o incidencia -positiva o negativa- en el problema y su solución.</p> <p>✓ Para priorizar se pueden utilizar criterios como:</p> <ul style="list-style-type: none"> - Tipo de relación con lo planteado - Afectación, aportes, intereses y/o necesidades - Potencial en términos de contribuciones, fortalezas y debilidades - Relaciones de cooperación o dependencia, de conflictos e intereses.
f. Elaborar una tabla resumen de los involucrados.	✓ Presentar con base a la prioridad establecida un cuadro o tabla resumen de los involucrados.

Tabla 5: Resultado de la Identificación de Involucrados.

I.- Identificación de Participantes, Involucrados y Relacionados			
Personas	Grupos de interés	Instituciones	Otros
<ul style="list-style-type: none"> ✓ Estudiantes que ingresan al Trayecto Inicial del PNFI en el CUFM ✓ Estudiantes del PNFI en el CUFM ✓ Profesores del CUFM de la Unidad Curricular Algorítmica y Programación del PNFI ✓ Coordinador del PNFI en el CUFM ✓ Jefe División Académica del CUFM 	<ul style="list-style-type: none"> ✓ Equipo Directivo del CUFM ✓ Control de Estudios del CUFM ✓ Centro de Estudiantes del CUFM ✓ Grupo Organizador de Olimpiadas de Programación en el CUFM 	<ul style="list-style-type: none"> ✓ Comisión Técnica Inter-institucional del PNF ✓ MPP Educación Universitaria, Ciencia y Tecnología 	<ul style="list-style-type: none"> ✓ Comunidad Universitaria del CUFM en General

Tabla 6: Resultado del Análisis de Involucrados.

II.- Análisis de Participación de los Involucrados					
Involucrado	Tipo	Relación	Rol	Influencia	Prioridad
✓ Profesores del CUFM de la Unidad Curricular Algorítmica y Programación del PNFI	Persona Natural	Directa	Contribuyente Beneficiario Evaluador	Alta	1
✓ Estudiantes que ingresan al Trayecto Inicial del PNFI en el CUFM	Persona Natural	Directa	Beneficiario Evaluador	Alta	1
✓ Coordinador del PNFI en el CUFM	Persona Natural	Directa	Apoyo	Alta	1
✓ Jefe División Académica del CUFM	Persona Natural	Indirecta	Apoyo	Alta	1
✓ Estudiantes del PNFI en el CUFM	Persona Natural	Indirecta	Beneficiario	Alta	2
✓ Profesores del CUFM de las otras unidades curriculares del PNFI	Persona Natural	Indirecta	Apoyo Para Consultar	Alta	2

Tabla 6: Resultado del Análisis de Involucrados (continuación).

II.- Análisis de Participación de los Involucrados (continuación)						
Involucrado	Tipo	Relación	Rol	Influencia	Prioridad	
✓ Equipo Directivo del CUFM	Grupo de Interés	Indirecta	Apoyo	Alta	3	
✓ Comisión Técnica Inter-institucional del PNF	Institución	Indirecta	Evaluador	Alta	3	
✓ MPP Educación Universitaria, Ciencia y Tecnología	Institución	Indirecta	Apoyo	Alta	3	
✓ Centro de Estudiantes del CUFM	Grupo de Interés	Indirecta	Apoyo	Media	4	
✓ Control de Estudios del CUFM	Grupo de Interés	Indirecta	Apoyo	Baja	4	
✓ Grupo Organizador Olimpiadas de Programación en el CUFM	Grupo de Interés	Indirecta	Evaluador	Baja	4	
✓ Comunidad Universitaria del CUFM en General	Grupo de Interés	Indirecta	No Definido	Baja	5	

5.1.3 Análisis de Problemas: Árbol del Problema

En el análisis del problema se determinan las causas y efectos del problema planteado en este trabajo de investigación. El resultado del análisis se expresa mediante un Árbol del Problema. Lo anterior se efectúa mediante lo siguiente:

- *Definición del Problema Central*

El punto de partida para la formulación del proyecto es la identificación y el análisis de la situación actual, lo cual permite identificar los problemas sobre los cuales se propone intervenir, y seleccionar el problema central que será abordado por el proyecto.

Tabla 7: Pasos para la Definición del Problema Central.

DEFINICIÓN DEL PROBLEMA CENTRAL	
PASO	DESCRIPCIÓN
a. Enunciar y seleccionar los principales problemas sobre la situación planteada.	1) Listado e identificación de los principales problemas actuales (no futuros o anteriores).
b. Especificar la existencia del problema central.	2) Un problema es una situación negativa, no es la carencia o déficit de algún tipo de recurso.

- *Desarrollo del Árbol del Problema*

Uno de los instrumentos metodológicos comúnmente utilizado para representar el problema central es el “Árbol del Problema”. Permite analizar las causas y efectos del problema planteado, facilitando precisar la contribución del proyecto a la solución de la problemática planteada.

Tabla 8: Pasos para el Desarrollo del Árbol del Problema.

DESARROLLO DEL ÁRBOL DEL PROBLEMA	
PASO	DESCRIPCIÓN
Enunciar el problema central	Formular el problema central en forma resumida y concreta. Escribirlo en el centro de la página.
Seleccionar y registrar las causas del problema central	Registrar las causas en la parte baja -raíz- del árbol. Organizarlas en orden de causalidad, y especificar número posible de causas directas.
Seleccionar y registrar los efectos del problema central	Registrar los efectos en la parte alta -ramas- del árbol. Organizarlas en orden de causalidad, y especificar número posible de causas directas.
Definición del esquema: problema central, causas y efectos	Elaborado el árbol preliminar, revisar, verificar, ajustar, validar su lógica y consistencia.

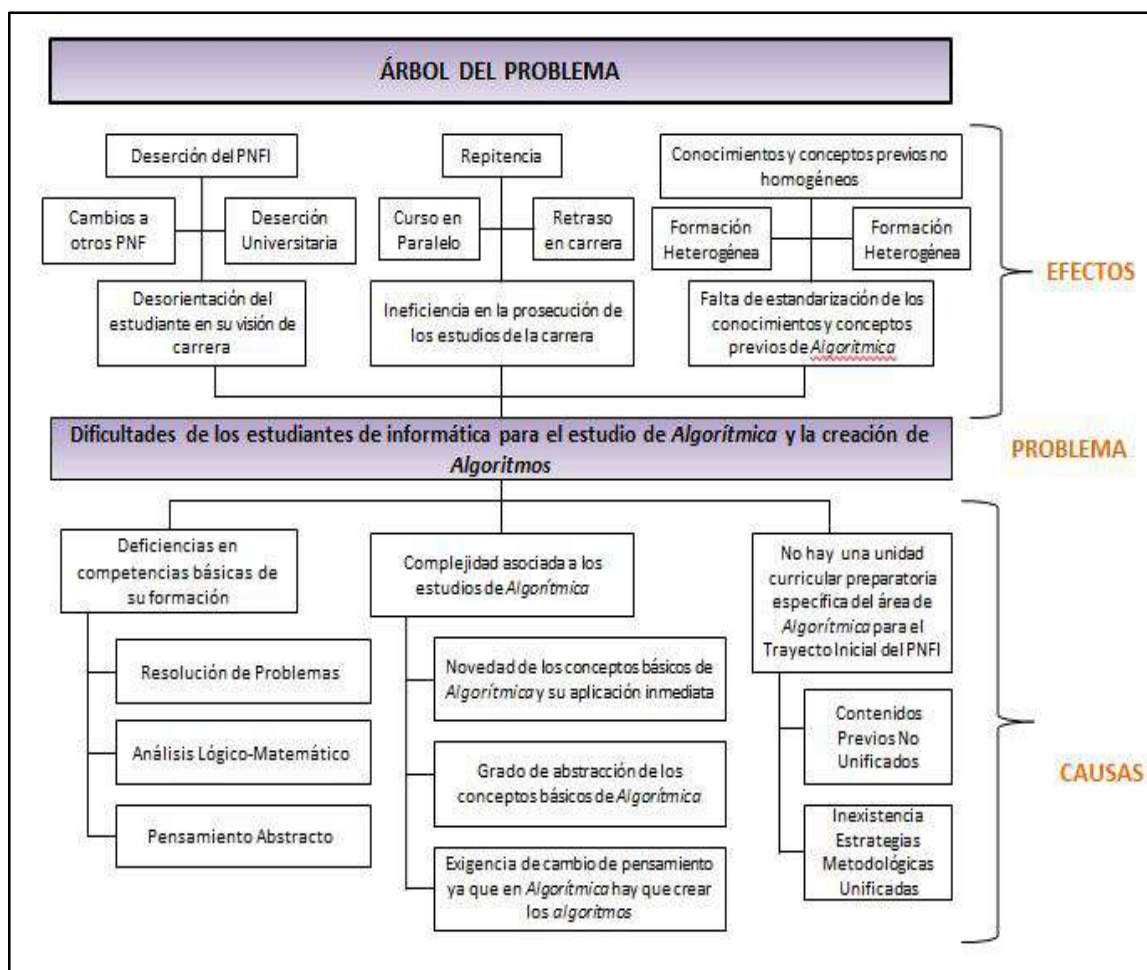


Figura 12: “Árbol del Problema” Resultante.

5.1.4 Análisis de Objetivos: Árbol de Objetivos

El análisis de objetivos permite identificar las posibles soluciones, expresándolas de manera contraria al problema y a los enunciados sobre causas y efectos del árbol de problemas.

En el análisis de objetivos se transformó el problema en una situación positiva futura, hacia la que se quiere llegar con el proyecto que estamos formulando, teniendo como punto de partida el “Árbol del Problema”.

A continuación se presenta una tabla descriptiva de los pasos aplicados para el análisis de objetivos y el árbol de objetivos resultante:

Tabla 9: Pasos para el Análisis de Objetivos.

ANÁLISIS DE OBJETIVOS	
PASO	DESCRIPCIÓN
a. Transformar los problemas en objetivos	✓ Enunciar los problemas en situaciones positivas o condiciones deseables y posibles.
b. El problema central es el objetivo o central Las causas son medios Los efectos son fines	✓ Diseñar nuevamente y bajo el modelo de un árbol lo que se transforma en objetivo central, las causas son las raíces, los fines son las ramas.
c. Revisar y validar el árbol de objetivos	✓ Analizar la consistencia del objetivo planteado con sus medios y fines. Ajustarlo las veces que se considere necesario. ✓ Se pueden ajustar, añadir o eliminar frases; o descartar objetivos que no se consideren necesarios y fundamentales.

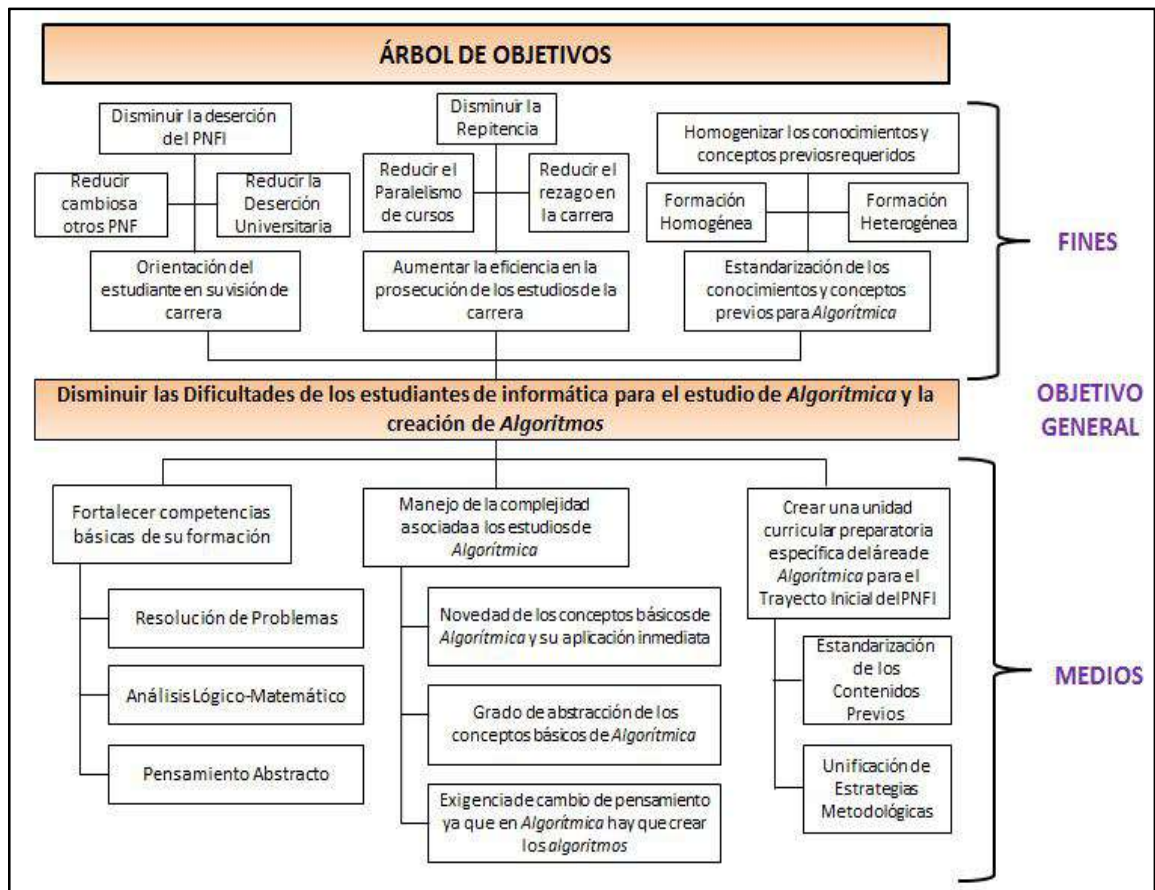


Figura 13: “Árbol de Objetivos” Resultante.

5.1.5 Análisis de Alternativas: Selección de Alternativa

El problema planteado puede solucionarse de diferentes maneras; con base a los diversos medios para lograr el objetivo (ver el árbol de objetivos) se definen las alternativas de solución a plantearse. Para lo cual se aplica lo siguiente:

Tabla 10: Pasos para el Análisis de Alternativas de Solución.

ANÁLISIS DE ALTERNATIVAS DE SOLUCIÓN	
PASO	DESCRIPCIÓN
a. Excluir objetivos no deseables o realizables	✓ Analizar todos los objetivos establecidos y desechar los que no son posibles de realizar o los que no son factibles o deseables de realizar por el momento.
b. Definir acciones que permitan lograr los medios	✓ Se especifican actividades que permitan lograr los objetivos definidos como medios para superar el problema central.
c. Establecer las alternativas posibles	✓ De acuerdo con los medios y las acciones definidas se organizan las alternativas posibles para realizar el análisis de la selección o combinación para definir la alternativa del proyecto.

- *Alternativas de Solución Resultantes*

Opción 1: Fortalecimiento de las competencias básicas del estudiante para la resolución de problemas, análisis lógico-matemático y pensamiento abstracto.

Opción 2: Ayudar al estudiante en manejo de la complejidad asociada a los estudios de Algorítmica: novedad y grado de abstracción de los conceptos básicos y su aplicación inmediata; y exigencia de cambio de pensamiento requerida para la creación de los algoritmos.

Opción 3: Creación de una *unidad curricular preparatoria específica* del área de Algorítmica para el trayecto inicial del PNFI que estandarice los contenidos previos de la unidad curricular “Algorítmica y Programación” para el Trayecto Inicial del PNFI y unifique las estrategias metodológicas a utilizar.

- *Selección de la Alternativa de Solución*

La selección de la alternativa de solución se efectúa en función de su valoración sobre varios aspectos: se analizan las diferentes alternativas planteadas, se determina si son excluyentes o complementarias; se decide las que se descartan por alguna razón de peso; y se comparan con base a criterios como costo, costo/beneficio, posibilidad de efectuarse, probabilidad de éxito, tiempo y condiciones de realización, entre otros que se definan.

Para el caso de esta investigación se selecciona la Opción 2 por considerar que tiene más posibilidades de realizarse, puede incluir la opción 1, y además, se alinea con los objetivos planteados en este trabajo.

5.2 Definición de Bases Pedagógicas

En esta sección se muestra el resultado del procesamiento aplicado a los datos e información obtenidos durante la fase exploratoria de la investigación. Mediante un análisis crítico e interpretativo de los elementos focales, se estableció la definición de las bases pedagógicas que fundamentan el diseño de un curso propedéutico para enseñar algoritmos.

5.2.1 Punto de Partida: El Constructivismo

Se ratifica la suscripción de la corriente pedagógica constructivista porque describe el proceso educativo desde una concepción de interactividad entre los actores involucrados, que propicia una dinámica de participación entre ellos. Asimismo concibe el conocimiento como una construcción del estudiante o persona que aprende, asumiendo un rol protagónico en el proceso, en el cual el docente cumple una función de orientación. Desde esa perspectiva pedagógica se determinan y justifican las acciones a realizar en la práctica educativa. En el documento rector del PNFII (2008) se expresa "...Este enfoque se basa en la investigación, la reflexión, la construcción del conocimiento y la promoción del trabajo cooperativo..." (p.77).

Es importante destacar que en la unidad curricular "Algorítmica y Programación" el enfoque del conocimiento en cuestión, los algoritmos, está

determinado por la creación y no por su uso, es decir, el estudiante debe desarrollar la capacidad de crear los algoritmos, lo que va más allá de ser un mero usuario de estos. Esto tiene correspondencia con la concepción de la corriente pedagógica seleccionada.

5.2.2 Aprender a Aprender

A los postulados de la UNESCO que a continuación se enuncian:

- Aprender a Ser
- Aprender a Convivir
- Aprender a Hacer
- Aprender a Conocer

Los cuales se referencian para señalar que forman parte de la fundamentación pedagógica del Programa Nacional de Formación, en PNFI (2008, p.46-47), se agrega el siguiente:

- Aprender a Aprender

Se considera de suma importancia para el proceso de enseñanza y aprendizaje de los algoritmos, por las implicaciones que tiene para la realización de estudios en esa área de conocimiento, ya que significa la iniciación de un proceso de alfabetización tecnológica fundamental en la formación profesional de la disciplina informática o computacional.

5.2.3 Motivación

Aunque el PNFI hace hincapié en la dimensión social del programa, destacando la importancia de los procesos de socialización para el desarrollo y formación del estudiante; de lo cual se puede inferir que se mantendrá en interacción permanente con su proceso educativo, se considera importante la incorporación de un componente motivacional liderizado por el docente que, extendido al grupo, incentive al estudiante a comprometerse con su formación técnica, con la adquisición y desarrollo de nuevos conocimientos, así como de su evolución personal.

5.2.4 Competencias

Una competencia consiste en una aptitud que, demostrada de forma individual, comprende la combinación una serie de conocimientos, capacidades, habilidades y destrezas para utilizar los conocimientos teóricos y saberes prácticos para realizar una función determinada o afrontar situaciones y requisitos profesionales tanto habituales como cambiantes.

A continuación se describe tres clases de competencias que se manejan en el ámbito educativo:

- Las competencias básicas requeridas para un desempeño efectivo en la vida cotidiana o para llevar a cabo cualquier tipo de estudios, en este caso, a nivel universitario. Se corresponden a las que deben haber sido adquiridas desde los estudios escolares básicos hasta el bachillerato.
- Las competencias específicas: el estudiante durante el lapso de su formación académica universitaria debe desarrollar y demostrar la adquisición de competencias específicas de su carrera, estas refieren las correspondientes a una disciplina determinada y se clasifican en tres tipos:
 - Del Saber: aquellas competencias académicas o relativas a conocimientos teóricos, a la adquisición del conocimiento;
 - Del Hacer: competencias disciplinares o conjunto de conocimientos prácticos requeridos para cada disciplina o sector profesional; y,
 - Del Saber Hacer: competencias del ámbito profesional, que incluyen habilidades de comunicación e indagación o investigación, así como el *know how*, que se aplican al ejercicio de una profesión concreta.
- Las competencias transversales o genéricas: su desarrollo potencia el de otras destrezas, habilidades y competencias en sí misma, con independencia de la formación profesional. También se les denomina competencias para la vida y se clasifican en:
 - Instrumentales: incluyen habilidades cognoscitivas, capacidades metodológicas, destrezas lingüísticas y tecnológicas.

- Interpersonales: de orden individual son relativas a la expresión de sentimientos, habilidades de crítica y autocrítica; así como destrezas sociales relacionadas (capacidad de trabajar en equipo, expresión de compromiso social o ético, entre otros).
- Sistémicas: combinan la comprensión, la sensibilidad y el conocimiento que permiten al individuo ver cómo las partes de un todo se relacionan y se agrupan.

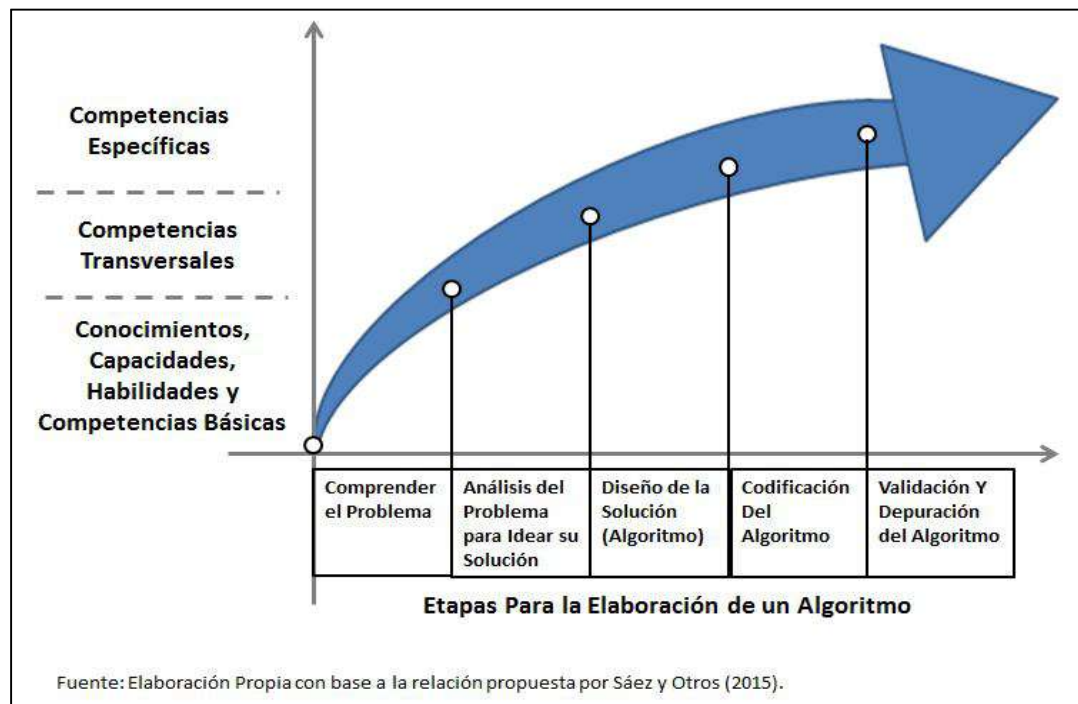


Figura 14: Relación Desarrollo Etapas Elaboración de Algoritmo y Competencias.
Fuente: Elaboración Propia Con Base a la Propuesta por Sáez y Otros (2015).

El diseño del curso propedéutico para enseñar algoritmos debe orientarse hacia la formación o desarrollo en competencias. En correspondencia a la concepción pedagógica que fundamenta al PNFI y en atención a las tendencias en cuanto a formación académica en general. En la actualidad es tal la generación y producción de información sobre los conocimientos a disponibilidad del estudiante que el énfasis tiene que hacerse en el logro de competencias para su adecuada aplicación o utilización.

5.2.5 Pensamiento Computacional

Se finaliza esta sección con una breve reseña sobre un término, de actualidad, que vincula el proceso de elaboración de programas y su aplicación en la educación: “Pensamiento Computacional”. En su trabajo Zapata-Ros (2015), plantea lo siguiente:

El aprendizaje de la programación debe hacerse de forma gradual y progresiva, desde las primeras etapas de formación individual, tal y como se hace con las habilidades básicas: lectura, escritura y matemática.

En la programación hay una forma específica de pensar que propicia el análisis y la relación de ideas, para la organización y la representación lógica. Dichas habilidades se favorecen con la realización de ciertas actividades en entornos específicos de aprendizaje desde los primeros años o etapas de la formación individual, se trata del desarrollo de un pensamiento específico: el pensamiento computacional. Zapata-Ros concluye que ese pensamiento trata de una “competencia compleja o más bien un complejo de habilidades” (p.40).

5.3 Definición de Bases Temáticas

En esta sección se presentan los contenidos que conforman la base temática a tratar en el curso propedéutico para la enseñanza de algoritmos. Se seleccionaron en correspondencia con el enfoque pedagógico descrito, el conocimiento a estudiar -algoritmos- y a las habilidades, destrezas y competencias; que el estudiante que se inicia en el aprendizaje de la algorítmica y programación debe desarrollar.

Se propone que el contenido programático se divida en dos partes, a saber:

La primera parte del contenido del curso propedéutico es la siguiente:

- Resolución de Problemas
- Introducción a la Lógica Proposicional
- Algebra de Boole
- Sistemas Numéricos:
 - o Conceptos de Sistemas Numéricos
 - o Sistema Numérico Decimal
 - o Sistema Numérico Binario

La segunda parte del curso, corresponde a los contenidos propios de algorítmica. Antes de especificarlo se presenta el fundamento matemático que justifica su selección: Clasificación de algoritmos según la teoría de la complejidad computacional.

Los estudios iniciales de algorítmica y programación se realizan con base a la elaboración de problemas que pueden ser resueltos computacionalmente, teniendo como propósito la elaboración de algoritmos de tipo determinístico; los cuales tienen las características descritas en la sección 3.2.1.1. Dicha clasificación se ilustra con el mapa de conceptos que se muestra en la siguiente figura:

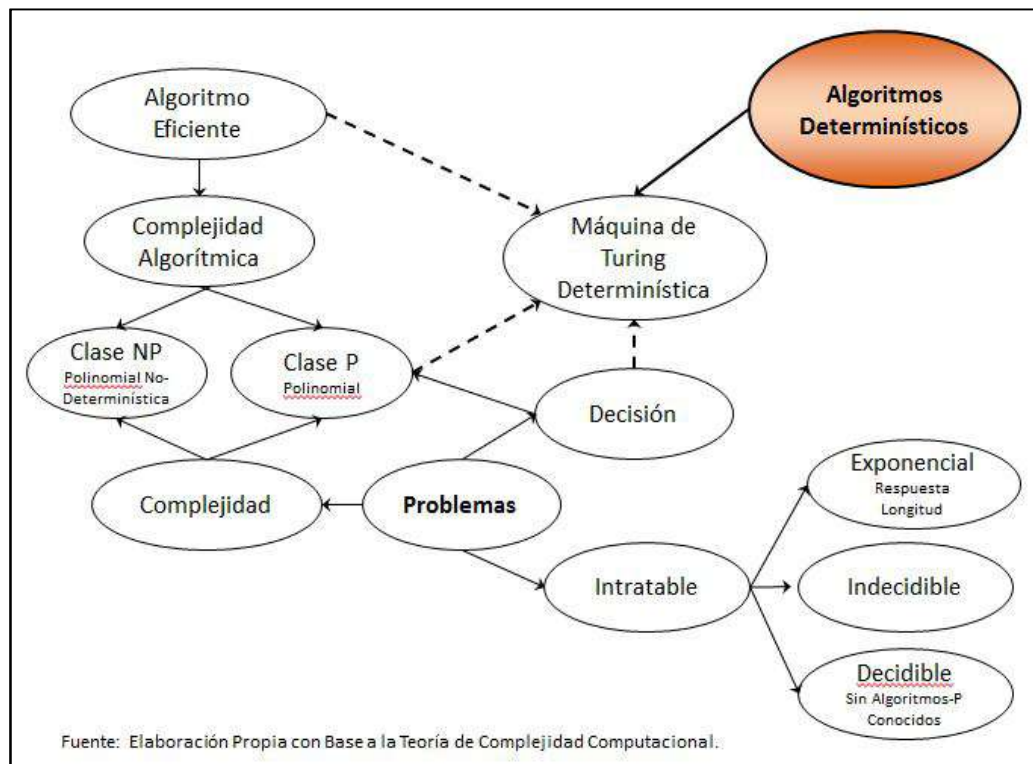


Figura 15: Clasificación de Algoritmos.

Fuente: Elaboración propia con base la Teoría de la Complejidad Computacional

Presentada la clasificación del tipo de algoritmo, se destaca que su creación se produce con la utilización de lenguajes imperativos, descartando para dicho aprendizaje el uso de lenguajes de programación lógicos y funcionales.

Por lo antes expuesto, el contenido algorítmico de la segunda parte del curso propedéutico es el siguiente:

- Origen, definición y aplicación del término algoritmo
- Resolución de Problemas Por Computadoras
- Elementos Para Elaborar Algoritmos:
 - o Conceptos Básicos
 - o Objetos Algorítmicos
 - o Acciones Algorítmicas Elementales
 - o Acciones Algorítmicas de Decisión
 - o Acciones Algorítmicas Cíclicas

En el capítulo II, Referencia Teóricas, sección 2.2.1 Sobre los Algoritmos, se detalla ampliamente en qué consiste cada uno de los tópicos indicados en la lista anterior.

5.4 Definición de Bases Didácticas

La Didáctica estudia los procesos y elementos existentes en el proceso de enseñanza y el aprendizaje, se ocupa de los métodos, las técnicas, los recursos y actividades adecuados para llevar a cabo la enseñanza. En esta sección se muestra la propuesta de los elementos que constituyen la base didáctica del curso propedéutico para enseñar algoritmos.

5.4.1 Principios Didácticos

En Herrán (2011) describe un conjunto de características e intenciones que conforman el sistema de “principios didácticos” bajo los cuales se desempeña un docente universitario para hacer de la enseñanza un proceso efectivo e innovador y con los cuales fundamenta la aplicación de sus estrategias metodológicas.

La siguiente lista de principios fue elaborada con base a la presentada por Herrán (2011) en su trabajo:

- Flexibilidad en la planificación para adaptarse a situaciones que se den.
- Capacidad de adaptación al contexto del proceso educativo.

- Promoción de la participación activa del estudiante.
- Asunción de un rol orientador en el proceso de enseñanza y aprendizaje.
- Incentivación de un clima adecuadamente positivo y agradable para el proceso de enseñanza y aprendizaje.
- Obtención de resultados como producto del trabajo efectuado.

5.4.2 Estrategias Metodológicas

En esta sección se presenta el conjunto de técnicas, actividades y recursos que se proponen para la definición de las estrategias metodológicas a aplicar en el curso propedéutico para enseñar algoritmos. Se proponen teniendo correspondencia con las bases pedagógicas (ver sección 5.2).

5.4.2.1 Técnicas Didácticas

- ✓ Exposición Didáctica
- ✓ Resolución de Problemas
- ✓ Talleres o Prácticas Individuales
- ✓ Trabajo en Pequeños Grupos
- ✓ Socialización de Resultados
- ✓ Discusión Grupal
- ✓ Revisión o Lectura Grupal del Material Documental
- ✓ Proyectos Didácticos (en Algoritmos)
- ✓ Tutoría entre pares
- ✓ Trabajo de investigación
- ✓ Actividades Lúdicas

5.4.2.2 Actividades

- *Del Docente:*
 - ✓ Preparación y elaboración del material didáctico
 - ✓ Revisión y selección del material bibliográfico
 - ✓ Organización de las sesiones de clases
 - ✓ Exposición didáctica de los contenidos
 - ✓ Orientación individual
 - ✓ Orientación grupal

- ✓ Gestión del proceso instruccional durante el curso
- *Del Estudiante*
 - ✓ Lectura del material bibliográfico
 - ✓ Procesamiento individual de la información
 - ✓ Procesamiento grupal de la información
 - ✓ Preparación y elaboración de los ejercicios requeridos
 - ✓ Preparación y elaboración de los ejercicios complementarios

5.4.2.3 Recursos

Lista de recursos requeridos para el curso propedéutico de enseñanza de algoritmos:

Tabla 11: Lista de Recursos para el Curso Propedéutico.

Recursos Curso Propedéutico	
Tipo	Descripción
Materiales Físicos	- Pizarra Blanca - Marcadores para pizarra blanca - Computadoras (operativas y habilitadas)
Dotación Física	- Salón para clases teóricas - Laboratorio de Computación
Material Didáctico	- Libros sobre introducción a la programación - Guías de ejercicios
Tecnológicos	- Aplicación(es) para la producción de algoritmos: PSEint - Aplicación(es) para la realización de actividades inductivas para la programación: The Turtle, Scratch, entre otros - Uso de Internet

5.5 Formulación del Programa del Curso Propedéutico

A continuación se muestra la formulación preliminar del programa propuesto para el curso propedéutico para enseñar algoritmos:

<p>CURSO PROPEDEUTICO PARA ENSEÑAR ALGORITMOS</p> <p>PROGRAMA</p> <p>(Versión Inicial)</p>

Descripción:

El curso propedéutico para la enseñanza de algoritmos va dirigido a los estudiantes del trayecto inicial del Programa Nacional de Formación en Informática (PNFI). Es preparatorio para la unidad curricular "Algorítmica y Programación".

Se trata de un curso teórico-práctico que exige el manejo de los conocimientos teóricos propuestos en el contenido y de su aplicación en la resolución de problemas. Requiere de un manejo operativo mínimo del computador dado que se utilizará herramientas de software para la creación y prueba de los algoritmos.

El Docente debe promover la participación activa del estudiante y motivar el compromiso con su propio desarrollo; ya que la formación en Algorítmica y Programación exige una actitud activa para lograr la creación de los algoritmos.

Objetivo General:

A partir del análisis realizado a los problemas planteados y de los conocimientos adquiridos sobre Algorítmica, el estudiante podrá diseñar algoritmos que solucionen esos problemas.

Contenido:

Para lograr el objetivo general, el programa consta de cuatro unidades, a desarrollarse de forma secuencial:

Unidad I: Razonamiento Numérico y Matemático

Objetivo: Aplicar el razonamiento numérico y matemático en la resolución de problemas en función de los aportes que tienen en la elaboración de algoritmos.

Temario:

1. Sistemas Numéricos en General
2. Sistema Numérico Decimal
3. Sistema Numérico Binario
4. Equivalencias entre el Sistema Decimal y Sistema Binario
5. Resolución de Problemas

Unidad II: Introducción a la Lógica Proposicional y Álgebra de Boole

Objetivo: Presentar las nociones de la lógica proposicional y del álgebra booleana en función de su relación directa con los algoritmos.

Temario:

1. Nociones de Lógica
2. Proposiciones
3. Conectivo Lógicos: negación, conjunción y disyunción

4. Tabla de Verdad
5. Introducción al Algebra de Boole

Unidad III: Algoritmos Básicos

Objetivo: Diseñar algoritmos que solucionen problemas de tipo secuencial.

Temario:

1. Algoritmo: Origen, definición, características y clasificación
2. Método Para Elaborar Algoritmos
3. Objeto Algorítmicos: tipos de datos, variables, constantes y expresiones
4. Acciones Algorítmicas Elementales: leer, asignar y escribir
5. Acciones Algorítmicas de Decisión

Unidad III: Algoritmos Cíclicos o Iterativos

Objetivo: Diseñar algoritmos que solucionen problemas de tipo cíclico o iterativo.

Temario:

1. Problemas que requieren soluciones cíclicas
2. Ciclo o iteración: estructura y condiciones
3. Variables en los Ciclos: Contadores y Acumuladores
4. Acciones Algorítmicas: Repetir y Mientras
5. Equivalencia Funcional de los Ciclos

Evaluación:

- Formativa mediante la realización de las actividades definidas con base a las estrategias didácticas seleccionadas.
- Sumativa a través de actividades individual que permitan demostrar el dominio los conocimientos y las habilidades, destrezas y competencias alcanzadas.

5.6 Plan Preliminar Para el Desarrollo e Implementación

Hasta aquí se han presentado las definiciones de las bases pedagógicas, temáticas y didácticas sobre las que se fundamenta la creación de un curso propedéutico para enseñar algoritmos; y una versión inicial de su programa. En esta sección se propone un plan preliminar para el desarrollo y la implementación o ejecución del curso propedéutico.

5.6.1 Premisas

- La ejecución del curso propedéutico puede realizarse dentro de la dinámica propia del PNFI, los recursos materiales a utilizar se encuentran en los ya establecidos. Por lo cual no se exige la adquisición de recursos

extraordinarios particulares dado que se tiene apalancamiento en la operatividad propia de la ejecución del PNFI en la institución.

- Los Docentes que pueden llevar a cabo el curso propedéutico para enseñar algoritmos, son los que pertenecen al equipo de la unidad curricular “Algoritmos y Programación”.
- La coordinación del PNFI avala la continuidad de la propuesta del curso propedéutico y la formulación del proyecto para ser presentado ante las autoridades del CUFM.

5.6.2 Estructura Desagregada de Trabajo

La continuidad del diseño del curso propedéutico está dada por la ejecución de las actividades estructuradas a seguir en el marco un proyecto: CPROALGORIT. A continuación se presenta la estructura desagregada de trabajo (EDT) para realizar su formulación, desarrollo e implementación:

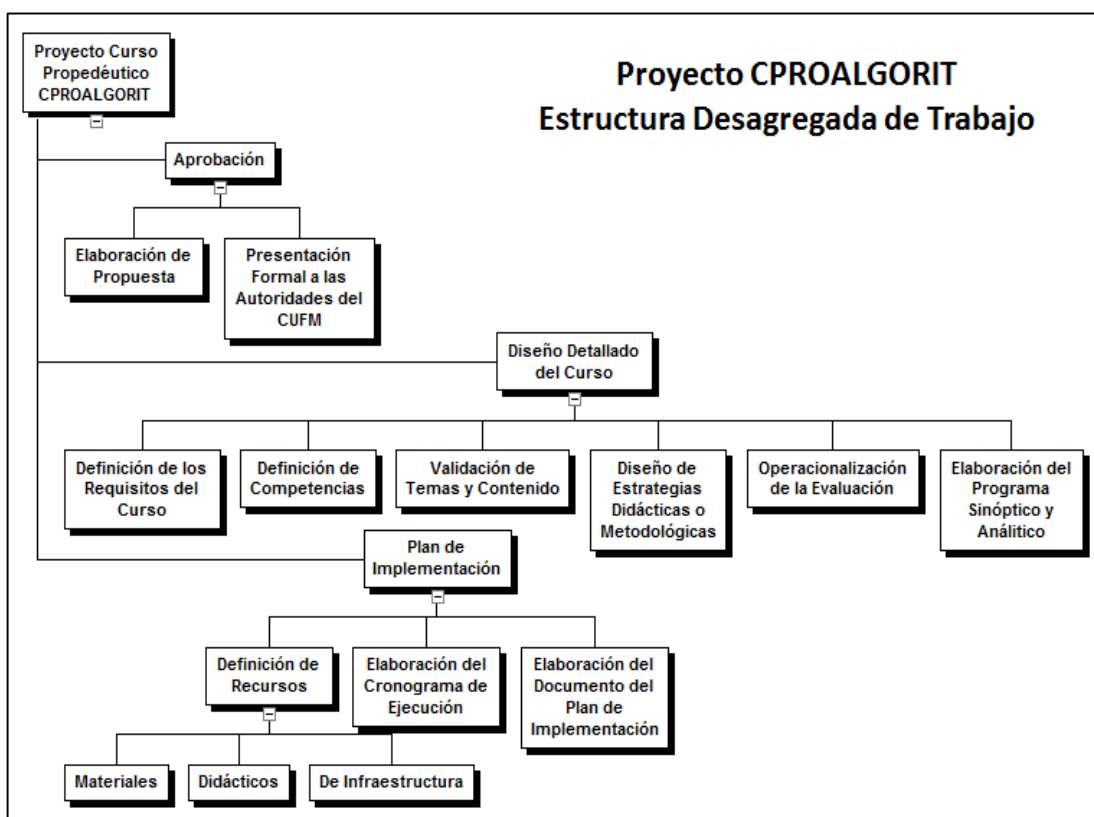


Figura 16: EDT del Proyecto CPROALGORIT. Fuente: Elaboración propia.

CAPÍTULO VI: LECCIONES APRENDIDAS

A raíz de las reflexiones hechas sobre la realización de este trabajo de investigación, se presenta las que se consideran que son las lecciones aprendidas más relevantes:

- ✓ Una vez formulado el problema y delimitado su alcance, es de suma importancia la definición de objetivos en la estructuración y organización del proyecto. En este caso del proyecto de investigación que, es el trabajo especial de grado de la especialización de planificación, desarrollo y gestión de proyectos.
- ✓ El desarrollo del proyecto de investigación puede ser reorientado con base a los hallazgos obtenidos de la revisión documental que se realiza para la obtención de antecedentes y marco referencial del trabajo; lo cual conduce a efectuar una actualización de la documentación ya trabajada. Esto puede repetirse una y otra vez, más aún con las facilidades de acceso a la información que se tiene en estos tiempos. La definición del alcance de la investigación y sus objetivos, constituyen el anclaje para no perderse en el mar de información que da la *Internet*.
- ✓ El marco metodológico es determinante para la culminación exitosa del trabajo de investigación, porque proporciona el marco de acciones a ejecutar para desarrollarlo apropiadamente.

CAPÍTULO VII: CONCLUSIONES Y RECOMENDACIONES

6.1 Conclusiones

El trabajo de investigación logró la realización de los objetivos específicos planteados debido a que:

- 1) Se estudió la situación problemática mediante la aplicación de pasos seleccionados de la Metodología del Marco Lógico. Se analizaron los actores involucrados, las sus causas y efectos del problema y las alternativas de solución; para la elaboración de la tabla resumen de los involucrados, el árbol de problema, el árbol de objetivos y la selección de la alternativa de solución.
- 2) Se determinó las bases pedagógicas que fundamentan el diseño de un curso propedéutico para la enseñanza de algoritmos; con la selección del constructivismo como la teoría pedagógica que enmarca, tanto la especificación de las competencias básicas que el estudiante debe fortalecer, como las competencias específicas para el área algorítmica que deben desarrollar.
- 3) Se estableció la base temática que debe contemplar el diseño de un curso propedéutico para la enseñanza de algoritmo. Proporcionando el fundamento formal que sustenta la selección de los contenidos a ser cubiertos en el curso.
- 4) Se definió las bases didácticas que sustentan la selección de las estrategias metodológicas de enseñanza, recursos y actividades a emplear por el docente propuestas en el diseño de un curso propedéutico para la enseñanza de algoritmos mediante la identificación de los elementos claves sobre algoritmos, estrategias metodológicas, recursos y actividades a emplear por el docente.
- 5) Se elaboró la propuesta de un curso propedéutico para la enseñanza de algoritmos a través de la formulación de un programa de estudios.
- 6) Se propone un plan para dar continuidad a la formulación del proyecto planteado que, una vez culminada y obtenida la aprobación de su

realización, se podrá proceder a la ejecución y posterior implementación del curso propedéutico diseñado.

Por lo antes expuesto se puede afirmar que se cumplió el objetivo general de la investigación: Definir las bases funcionales para diseñar un curso propedéutico que permita la enseñanza de algoritmos, a los estudiantes del trayecto inicial del Programa Nacional de Formación Informática del Colegio Universitario “Francisco de Miranda”.

En la enseñanza de la algorítmica y la programación el docente debe utilizar estrategias metodológicas orientadas a estimular el desarrollo de las habilidades y destrezas necesarias, para que el estudiante logre alcanzar un pensamiento con enfoque algorítmico, manejando las exigencias y complejidades requeridas para la creación de algoritmos. Por lo tanto, el docente en la referida especialidad debe tener una formación integral tanto en los aspectos tecnológicos de la informática como en el área pedagógica.

Con el desarrollo de este trabajo especial de grado, se logró el manejo instrumental y aplicación de los conocimientos obtenidos durante la especialización realizada: planificación, desarrollo y gestión de proyectos; que en este caso, abarcó dos proyectos: uno, el proyecto de investigación del trabajo especial de grado y dos, la formulación de un proyecto educativo, objeto de estudio de la investigación, para el establecimiento de las bases funcionales para diseñar un curso propedéutico para la enseñanza de algoritmos.

6.2 Recomendaciones

- Dar continuidad a la formulación del proyecto para el diseño del curso propedéutico, completarlo y presentarlo antes las autoridades del Colegio Universitario “Francisco de Miranda”, a fin de obtener la aprobación, aval y apoyo de la institución para su ejecución e implementación.
- Realizar una revisión del registro académico del PNFI para hacer mediciones y cálculos para obtener las estadísticas sobre el ingreso, deserción, prosecución de los estudiantes, así como del desempeño histórico en la unidad curricular “Algoritmo y Programación”; esto proporcionaría información certera para la toma de decisiones sobre los cambios o mejoras a introducir en el programa.
- La realización de los cuadros de las competencias básicas requeridas para el ingreso y la realización de estudios del PNFI; las competencias generales, las transversales y las específicas, que debe desarrollar el estudiante con la unidad curricular “Algorítmica y Programación”, para formarse como un profesional competente del área de la informática.
- Se recomienda a las instituciones universitarias venezolanas que realicen investigación sobre la enseñanza y aprendizaje de la algorítmica y la programación, porque consolidará el vínculo –ya existente- entre el mundo de la educación y el mundo informático; mediante la obtención de conocimientos aplicados que promuevan mejoras educativas y tecnológicas; así como de estudios básicos que fundamenten ese enlace, por ejemplo, investigación sobre “Pedagogía de la Algorítmica” o “Pedagogía de la Programación”. Así sucedió con el trabajo conjunto llevado a cabo por Jean Piaget y Seymour Papert, siendo este último quien sentó las bases y elementos que dan el origen al término “Pensamiento Computacional”, actualmente en boga y promovido activamente por Jeanette Marie Wing.

CAPÍTULO VIII: REFERENCIAS BIBLIOGRÁFICAS

- Ala-Mutka, K. (2005). **Problems In Learning And Teaching Programming**. Papers. Tampere University of Technology, Finland. Recuperado de: https://www.cs.tut.fi/~edge/literature_study.pdf
- Albis, María Alejandra (2011). **Ciclos y fases de la identificación de proyectos**. Artículo. Universidad del Norte. Colombia. Recuperado de: <http://manglar.uninorte.edu.co/bitstream/handle/10584/2202/Ciclo%20del%20proyecto.pdf?sequence=1>
- Álvarez, Walter; Lacués, Eduardo; Pagano, Magdalena (2006). **Diseño de un curso nivelación al ingreso a la universidad, a partir de la caracterización del perfil de los ingresantes**. En Martínez, Gustavo (Ed.), Acta Latinoamericana de Matemática Educativa (pp. 514-520). México DF, México: Comité Latinoamericano de Matemática Educativa A. C. Recuperado de <http://funes.uniandes.edu.co/5586/>
- Andino J., Elsa M. (2014). **Instrumentos Operativos de Gestión Como Herramientas para el Desarrollo de Indicadores de Medición de Proyectos**. Trabajo de Maestría. Universidad Técnica de Ámbato. Ecuador.
- Apiola, M. (2013). **Creativity-Supporting Learning Environments: Two Case Studies on Teaching Programming**. Dissertation. Faculty of Science. University of Helsinki. Finland. Recuperado de: https://helda.helsinki.fi/bitstream/handle/10138/40146/apiola_dissertation.pdf?sequence=1
- Bennedsen, Jens. (2007). **Teaching and Learning Introductory Programming - A Model-Based Approach**. Dissertation. Aarhus University, Denmark Recuperado de: <https://www.duo.uio.no/bitstream/handle/10852/9962/bennedsen.pdf?sequence=1>
- Bladek, C.J. (2002). **A review and assessment of novice learning tools for problem solving and program development**. Master Thesis. New

Jersey Institute Technology. United States. Recuperado de:
<http://archives.njit.edu/vol01/etd/2000s/2002/njit-etd2002-089/njit-etd2002-089.pdf>

Bruno, Oscar Ricardo (2013). **SEPA Programación: Sistema Experto para Aprendizaje de Programación**. Instituto de Investigación en Ciencias de la Administración, Facultad de Ciencias de la Administración. Universidad Del Salvador. El Salvador. Recuperado de
<http://p3.usal.edu.ar/index.php/anuarioinvestigacion/article/view/2393>

Burón, J. (1993). **Enseñar a Aprender: Introducción a la Metacognición**. Bilbao: Ediciones Mensajeros.

Case, Desmond Robert.(2012). **An Animated Pedagogical Agent For Assisting Novice Programmers Within A Desktop Computer Environment**. Doctoral thesis, Staffordshire University. United King. Recuperado de: <http://eprints.staffs.ac.uk/1979/>

Contreras H., W. (2008). **Sistema de Rúbricas para la Evaluación de Habilidades y Actitudes en la Enseñanza del Diseño**. Trabajo de Maestría. Universidad Bio-Bio. Chile.

De la Fuente Olgún, J. (). **¿Qué es la Metodología de Marco Lógico y para qué sirve?** Facultad de Ciencias Físicas y Matemáticas. Universidad de Chile. Recuperado de: http://www.mgpp.cl/wp-content/uploads/2012/06/3_Marco_Logico.pdf

De Raadt, M. (2007). **A Review of Australasian Investigations into Problem-Solving and the Novice Programmer**. University of Southern Queensland, Australia. Recuperado de:
<https://core.ac.uk/download/pdf/11036781.pdf>

Deek, F., Hiltz, S., Kimmel, H. y Rotter, N. (1999). **Cognitive Assessment of Students' Problem Solving and Program Development Skills**. New Jersey Institute Technology. United States. Recuperado de:
https://www.academia.edu/20575945/Cognitive_Assessment_of_Students_Problem_Solving_and_Program_Development_Skills?auto=download

- Deek, F. y McHugh, J, (2003). **Problem Solving And Cognitive Foundations For Program Development : An Integrated Model.** Papers.New Jersey Institute Techonology. United States. Recuperado de:
<http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=5B7630A529F156CF8883CE6C12FBB1CD?doi=10.1.1.574.2044&rep=rep1&type=pdf>
- Del Prado, Ana y Lamas, Nancy (2014). **Alternativas para la enseñanza del pseudocódigo y diagrama de flujo.** Facultad de Tecnología y Ciencias Aplicadas. Universidad Nacional de Catamarca. Argentina.
- Guardián Soto, Beatriz (2003). **Estrategias para promover el aprendizaje significativo de la asignatura de análisis de algoritmos en el nivel de educación superior.** Trabajo de Maestría. Instituto Politécnico Nacional. México. Recuperado de
<http://tesis.ipn.mx:8080/xmlui/handle/123456789/14139>
- Hernández Sampieri, R.; Fernández-Collado, C. y Baptista L., P. (2006). **Metodología de la Investigación.** Mc-Graw Hill. Cuarta Edición. México.
- Herrán, A. de la (2011). **Técnicas didácticas para una enseñanza más formativa.** En N. ÁlvarezAguilar y R. Cardoso Pérez (Coords.), Estrategias y metodologías para la formación delestudiante en la actualidad. Camagüey (Cuba): Universidad de Camagüey (ISBN: 978-959-16-1404-9).Recuperado dehttps://www.uam.es/personal_pdi/profesorado/agustind/textos/teuniv.pdf
- Jackson, D. y Miller, R. (2009).**A New Approach To Teaching Programming.** Artículo. Recuperado de:
<http://people.csail.mit.edu/dnj/articles/teaching-6005.pdf>
- López Reguera, Jorge, Hernández Rivas, Cecilia, &Farran Leiva, Yussef (2011). **Una plataforma de evaluación automática con una metodología efectiva para la enseñanza/aprendizaje en programación de computadores.** *Ingeniare. Revista chilena de*

ingeniería, 19(2), 265-277. Recuperado de:

<http://www.scielo.cl/pdf/ingeniare/v19n2/art11.pdf>

Ministerio del Poder Popular Para la Educación Superior (2008). **Programa Nacional de Formación en Informática (PNFI)**. Contenidos Sinópticos y Analíticos. Caracas.

Ministerio del Poder Popular Para la Educación Universitaria (2009). Dirección General de Currículo y Programas Nacionales de Formación. **Lineamientos para Programas Nacionales de Formación**. Versión 2.0. República Bolivariana de Venezuela. Recuperado de: http://www.cufm.tec.ve/Contenido/Descargas/LINEAMIENTOS_VERSION2.pdf

Morales González, Á., Muñoz Montaña, J., & Peláez Valencia, L. (2006). **Una experiencia pedagógica para programas de ingeniería: La enseñanza de algoritmos mediada por la lógica cognitiva y la elaboración de juegos de lógica**. *Revista Educación en Ingeniería*, 1(2), 26-32. Recuperado de <http://www.educacioneningeneria.org/index.php/edi/article/view/38>

Moroni, Norma y Señas, Perla (2005). **Estrategias para la enseñanza de la programación**. Universidad Nacional del Sur. Bahía Blanca. Argentina. Recuperado de <http://sedici.unlp.edu.ar/handle/10915/18901>

Nardi, Alejandra M. (2006). **Diseño de Proyectos Bajo el Enfoque de Marco Lógico**. Universidad Nacional de Córdoba. Córdoba. Argentina. Recuperado de http://docencia.unet.edu.ve/Coordinaciones/SComunitario/archivos/Trabajo_Marco_Logico.pdf

Ortegón, E., Pacheco, J.F. y Prieto, A. (2005). **Metodología del marco lógico para la planificación, el seguimiento y la evaluación de proyectos y programas**. CEPAL. Reimpresión 2015. Recuperado de <http://www.cepal.org/es/publicaciones/5607-metodologia-marco-logico-la-planificacion-seguimiento-la-evaluacion-proyectos>

- Picardo J., Oscar (2004). **Diccionario Pedagógico**. UPAEP. Recuperado de:
<http://online.upaep.mx/campusvirtual/ebooks/diccionario.pdf>
- Project Managment Institute (PMI) (5ta Ed.)(2013). **Guia de los Fundamentos para la Dirección de Proyectos (Guía del PMBOK)**.
- Ramírez, E. (2015). **Alpha: una notación algorítmica basada en pseudocódigo**. *Télématique*, vol. 14, 97-121. Recuperado de:
<http://www.redalyc.org/articulo.oa?id=78435427007>
- Saez Villavicencio, A d I C; Menéndez Pérez, J S; Puentes Puentes, U; Ciudad Ricardo, F Á; (2015). **El desarrollo de la habilidad: implementar algoritmos. Teoría para su operacionalización**. *Revista Cubana de Ciencias Informáticas*, 9() 99-112. Recuperado de
<http://www.redalyc.org/articulo.oa?id=378343683005>
- Sánchez Baca, EdderAlaín (2015). **Sistema Interactivo Basado en un Intérprete de Algoritmos Para Mejorar el Método de Aprendizaje de los Alumnos del Curso Fundamentos de Programación**. Trabajo Especial de Grado. Facultad de Ingeniería. Universidad Católica Santo Toribio de Mogrovejo. Chiclayo. Recuperado de:
<http://tesis.usat.edu.pe/handle/usat/508>
- Selby, Cynthia Collins.(2014). **How can the teaching of programming be used to enhance computational thinking skills?** Doctoral Thesis.University of Southampton, Southampton Education School, United King. Recuperado de: <http://eprints.soton.ac.uk/366256/>
- Serrano, J. M. y Pons, R. M. (2011). **El constructivismo hoy: enfoques constructivistas en educación**. *Revista Electrónica de Investigación Educativa*, 13(1). Recuperado de:
<http://redie.uabc.mx/vol13no1/contenido-serranopons.html>
- Teague, Donna. (2011). **Pedagogy of Introductory computer programming: A people-first approach**.Queensland University of Technology.Faculty of Science and Engineering, Master of Information Technology Thesis.Australia. Recuperado de:
http://eprints.qut.edu.au/46255/1/Donna_Teague_Thesis.pdf

- Teague, Donna. (2015). **Neo-Piagetian Theory and the Novice Programmer**. Queensland University of Technology. Faculty of Science and Engineering, Doctoral Thesis. Australia. Recuperado de: http://eprints.qut.edu.au/86690/1/Donna_Teague_Thesis.pdf
- UPEL (4ta Ed.) (2006). **Manual de Trabajos de Grado de Especialización y Maestría y Tesis Doctorales**. Caracas.
- Wirth, N. (1986). **Algoritmos + Estructuras de Datos = Programas**. Madrid: Del Castillo.
- Wirth, N. (1987). **Algoritmos y Estructuras de Datos**. México: Prentice-Hall Hispanoamérica.
- Yamarte, Y. (1998). **AIDA. Ambiente Interactivo de Aprendizaje. (Caso de Estudio: Algoritmica-Nociones Básicas)**. Trabajo especial de grado. Facultad de Ciencias. Caracas: UCV.
- Yamarte, Y. (2005). **Propuesta de un Manual Teórico-Práctico para la Enseñanza de Algoritmos, dirigido a Estudiantes del Primer Año de Informática del Nivel Medio Diversificado y Profesional**. Trabajo especial de grado. Facultad de Humanidades y Educación. Caracas: UCV.
- Zapata-Ros, Miguel (2015). **Pensamiento computacional: Una nueva alfabetización digital. Computational Thinking: A New Digital Literacy**. RED-Revista de Educación a Distancia, 46(4). 15-Sept.-2015. Recuperado de: <http://www.um.es/ead/red/46/zapata.pdf>

Páginas digitales:

<http://www.cufm.tec.ve/index.php?Intro=1>. Consultada el 29/06/2016, a la 1:05 pm.

CAPÍTULO IX: ANEXOS

A. Origen de la palabra “*Algoritmo*”

Se atribuye el origen de la palabra *algoritmo* a las derivaciones provenientes del nombre del **Abu Abdallah Muḥammad ibn Mūsā al-Jwārizmī (Abu Yāffar)**, también conocido como **al-Juarismi**. Fue un persa musulmán (780-850 DC, aproximadamente), matemático, astrónomo y geógrafo; es considerado el padre del álgebra y, sin duda, el primer pensador y creador *algorítmico*.

“¿Cómo se escribe al-Khwārizmī?”⁵

En los libros de historia podemos encontrarnos con el nombre de nuestro matemático escrito de muchas maneras. Las rayas sobre la a y la i pueden estar presentes o no, o ser substituidas por acentos circunflejos, por razones tipográficas, pero ésa no es la diferencia más importante. Con rayas o acentos circunflejos, podemos encontrarnos con al-Khwārizmī, al-Jwārizmī, o al-Hwārizmī, y también al-Khowārizmī, al-Jowārizmī, o al-Howārizmī, o al-Khuwārizmī, al-Juwārizmī, o al-Huwārizmī, pero también con una e en el lugar de la primera i, al-Khwārezmī, al-Khowārezmī, al-Khuwārezmī, etc.

El origen de tanta variación está en que en realidad el nombre de nuestro matemático se escribe con un alfabeto distinto del latino, el alfabeto árabe o alifato, y se escribe así:

الخوارزمي

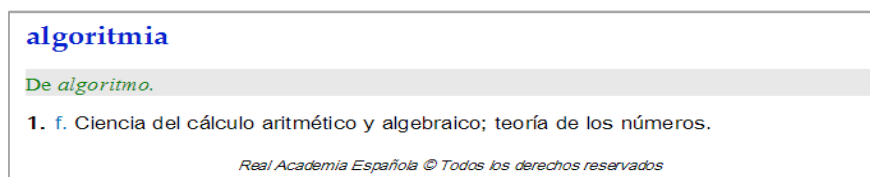
De la obra **al-Juarismi**, destacan los libros: “*Kitāb al-Mukhtasar fī hisāb al-jabr wa’l-muqābala*” (*Libro conciso de cálculo de restauración y oposición*), “*Kitāb al-hisāb al-‘adad al-hindī*” (*Libro del cálculo con los números hindúes*) y “*Istikhrāj ta’rīkh al-Yahūd*” (*Determinación del calendario judío*). **Al-Juarismi** creó *algoritmos* para el cálculo de fechas, el cálculo aritmético, realizó una introducción compacta al cálculo, usando reglas para completar y reducir ecuaciones; sistematizó la resolución de ecuaciones cuadráticas, entre otros aportes. **Al-Juarismi** no sólo creó conocimiento, además lo recopiló de los griegos e hindúes en diferentes áreas, aparte de las matemáticas, también en astronomía, astrología, geografía e historia.

⁵ Tomado de Puig, L. (2008). **Historias de al-Khwārizmī (1ª entrega)**. Revista SUMA+, Número 58, p. 1. Universitat de València.

“Los libros de al-Khwārizmī⁶”

Al-Khwārizmī es conocido sobre todo por su libro de álgebra y por el libro en que introduce el sistema de numeración posicional y cifrado de los hindúes y el cálculo aritmético en ese sistema, y es razonable que así sea por la importancia que ambos libros han tenido en la historia de las matemáticas. Sin embargo, se tiene noticia de que escribió un buen número de libros en varias disciplinas”.

Hay un segundo término vinculado etimológicamente a la palabra *algoritmo*, se trata de *algoritmia*. En el diccionario de la Real Academia Española lo definen así:



Indiscutiblemente la *algoritmia* está inscrita en el ámbito de la matemática, ya que sus objetos de estudio⁷ son el cálculo aritmético y algebraico, así como la teoría de los números. **Al-Juarismi** hizo grandes contribuciones al álgebra, ideó un *algoritmo* para la resolución de ecuaciones cuadráticas. De hecho, uno de sus libros dio origen al nombre del álgebra. *Algoritmia* y *algoritmica*, tienen un origen común en la palabra algoritmo pero sus ámbitos de acción son diferentes, por lo cual no deberían utilizarse como sinónimos. En el mundo de la Computación, la Informática, la Electrónica, las Tecnologías de la Información y Comunicación (TIC) y disciplinas afines; el término más apropiado para referirse a la creación, lógica de secuenciación, clasificación, definiciones, conceptos, técnicas, formas de uso o aplicaciones y todo lo relacionado con los algoritmos computacionales, sistemas digitales o sistemas informáticos, debe ser algorítmica. No obstante, a pesar de su amplia utilización en diversas aplicaciones de diferentes disciplinas, no se tienen definiciones únicas, estructuradas y formalizadas ni de algorítmica ni de algoritmo.

⁶ Tomado de Puig, L. (2009). **Historias de al-Khwārizmī (2ª entrega): los libros**. Revista SUMA+, Número 59, p. 106. Universitat de València.

⁷ Objetos estudio ideales, abstractos, no fácticos. Según Bunge, M. “La Ciencia. Su Método y Filosofía”, p. 6. Recuperado de https://users.dcc.uchile.cl/~cgutierr/cursos/INV/bunge_ciencia.pdf.

B. PNFI: Malla Curricular⁸

Software (Especialización en Software Libre)		Hardware (Especialización en Seguridad de Redes)																	
POSTGRADO																			
INGENIERO EN INFORMÁTICA Total UC= 184																			
IV				Redes Avanzadas	3	Formación Crítica IV	1			Proyecto Socio Tecnológico IV	4	Auditoría Informática	4			3	1	13	
							1				4	Gestión de Proyectos Informáticos	4				1	13	
								1				4	Seguridad Informática	4	Administración de Bases de Datos	3		1	13
DESARROLLADOR DE APLICACIONES																			
III	Investigación de Operaciones	3				Formación Crítica III	1			Proyecto Socio Tecnológico III	3	Ingeniería Software II	3	Modelado de +BD	3			13	
	Matemática Aplicada	3					1				3					3		13	
		3	Sistemas Operativos	3				1				3						13	
TÉCNICO SUPERIOR UNIVERSITARIO EN INFORMÁTICA Total UC = 106																			
II						Formación Crítica II	1	Paradigmas de Programación	4	Proyecto Socio Tecnológico II	3	Ingeniería del Software I	3			3	1	15	
	Matemática II	3					1				3						1	15	
		3	Redes de Computadora	3				1							Bases de Datos	3		1	18
SOPORTE TÉCNICO A USUARIOS Y EQUIPOS																			
I	Matemática I	3	Arquitectura del Computador	3	Formación Crítica I	1	Algorítmica y Programación	4	Proyecto Socio Tecnológico I	3							1	15	
		3					1									3	1	18	
		3						1										15	
Trayecto	Unidad Curricular	UC	Unidad Curricular	UC	Unidad Curricular	UC	Unidad Curricular	UC	Unidad Curricular	UC	Unidad Curricular	UC	Unidad Curricular	UC	Unidad Curricular	UC	Electivas (UC)	Idiomas (UC)	Total UC
TRAYECTO INICIAL		Unidad Curricular	UC	Unidad Curricular		UC	Unidad Curricular		UC	Unidad Curricular		UC	Unidad Curricular		UC	Total UC			
		Matemática	5	Proyecto Nacional y Nueva Ciudadanía		3	Taller de Introducción a la Universidad y al Programa		2							10			

Figura B.1- Malla Curricular del Programa Nacional de Formación en Informática (PNFI).

⁸Documento Rector del PNFI, MPPES, noviembre 2008. Recuperado de <http://nixon-docspnfi.blogspot.com/>

		Gobierno Bolivariano de Venezuela		Ministerio del Poder Popular para el Poder Judicial y la Educación Superior		Venezuela		ANEXO IV (P. 103-104)		
TRAYECTO 1 / TRIMESTRE 1										
Unidad Curricular	Módulo	Código	HTEA		HTEI		THTE		UC	PERFIL DEL PROFESOR ASESOR
			Semanal	Trimestral	Semanal	Trimestral	Semanal	Trimestral		
Matemática I	Lógica	PIMT113	5	60	3	30	8	90	3	Lic. en Matemática, Matemático(a), Ingeniero(a), Lic. en Educación mención Matemática, Profesores en Informática o profesionales afines.
Arquitectura del Computador	Estructura del Computador	PIAC113	5	60	3	30	8	90	3	Ingeniero(a) en Informática, Sistemas, Computación, Lic. en Informática, Computación, Profesores en Informática o profesionales afines.
Formación Crítica I	Cultura, Deporte y Recreación I	PIFC111	2	24	1	06	3	30	1	Profesionales de Ciencias Sociales, Humanísticas o Tecnológicas.
Algorítmica y Programación	Algorítmica y Programación	PIAP114	6	72	4	48	10	120	4	Ingeniero(a) en Informática, Sistemas, Computación, Lic. en Informática, Computación, Profesores en Informática o profesionales afines.
Proyecto Socio Tecnológico I	Soporte Técnico a usuarios y equipos I	PIPT113	5	60	3	30	8	90	3	Ingeniero(a) en Informática, Sistemas, Computación, Lic. en Informática, Computación, Profesores en Informática, Profesionales de Ciencias Sociales y Humanísticas con experiencia en desarrollo y seguimiento en proyectos informáticos.
Idiomas	Inglés – Comprensión Lectora I	PIID111	2	24	1	06	3	30	1	Lic. en Idiomas mención Inglés o profesionales afines con experiencia comprobable en el área.
TOTAL			25	300	16	150	40	450	16	

83

Figura B.2- Carga Horaria del Trayecto 1 del PNFI

		Gobierno Bolivariano de Venezuela		Ministerio del Poder Popular para el Poder Judicial y la Educación Superior		Venezuela		ANEXO IV (P. 103-104)		
TRAYECTO 1 / TRIMESTRE 2										
Unidad Curricular	Módulo	Código	HTEA		HTEI		THTE		UC	PERFIL DEL PROFESOR ASESOR
			Semanal	Trimestral	Semanal	Trimestral	Semanal	Trimestral		
Matemática I	Estadística y Probabilidades I	PIMT123	5	60	3	30	8	90	3	Lic. en Matemática, Matemático(a), Ingeniero(a), Lic. en Educación mención Matemática, Profesores en Informática o profesionales afines.
Arquitectura del Computador	Sistemas Operativos I	PIAC123	5	60	3	30	8	90	3	Ingeniero(a) en Informática, Sistemas, Computación, Lic. en Informática, Computación, Profesores en Informática o profesionales afines.
Formación Crítica I	Informática, Política de Estado y Soberanía I	PIFC121	2	24	1	06	8	30	1	Profesionales de Ciencias Sociales, Humanísticas o Tecnológicas.
Algorítmica y Programación	Programación I	PIAP124	6	72	4	48	10	120	4	Ingeniero(a) en Informática, Sistemas, Computación, Lic. en Informática, Computación, Profesores en Informática o profesionales afines.
Proyecto Socio Tecnológico I	Soporte Técnico a usuarios y equipos II	PIPT123	5	60	3	30	8	90	3	Ingeniero(a) en Informática, Sistemas, Computación, Lic. en Informática, Computación, Profesores en Informática, Profesionales de Ciencias Sociales y Humanísticas con experiencia en desarrollo y seguimiento en proyectos informáticos.
Electiva I		PIEL123	5	60	3	30	8	90	3	Ingeniero(a) en Informática, Sistemas, Computación, Lic. en Informática, Computación, Profesores en Informática o profesionales afines.
Idiomas	Inglés – Comprensión Lectora II	PIID121	2	24	1	06	3	30	1	Lic. en Idiomas mención Inglés o profesionales afines con experiencia comprobable en el área.
TOTAL			30	360	18	180	48	540	18	
ELECTIVAS PROPUESTAS										
Diseño Instruccional en las TIC			Capital Intelectual y Recursos Humanos				Gestión del Conocimiento			

84

Figura B.3- Carga Horaria del Trayecto 2 del PNFI

C. Unidad Curricular “Algorítmica y Programación” del PNFI⁹

SINOPSIS DE UNIDADES CURRICULARES	
ÁREA DE CONOCIMIENTO:	PRIMER TRAYECTO
PROGRAMA DE FORMACIÓN:	INFORMÁTICA
PERFIL DE SABERES	
<p>APRENDER A CONOCER: a través del Proyecto Socio Tecnológico I, los contenidos de las unidades curriculares estructuradas en el primer trayecto; Matemática I, Arquitectura del Computador, Formación Crítica I: Informática, Políticas de Estado y Soberanía, Algorítmica y Programación, Electiva I e Idiomas.</p> <p>APRENDER A HACER: en el Proyecto Socio Tecnológico I, insertar a los participantes en una dinámica de búsqueda y construcción de saberes a través de los contenidos estructurados en las unidades curriculares del trayecto I.</p> <p>APRENDER A CONVIVIR: mediante la incorporación del computador a las actividades diarias del ser humano preservando el ambiente y la comunidad.</p> <p>APRENDER A SER: responsable, corresponsable, autónomo, creativo, seguro, constante, tolerante, emprendedor, tenaz, abierto al cambio, curioso, respeto al otro, disposición al trabajo colaborativo.</p> <p>APRENDER A EMPRENDER: Proyectos orientados al desarrollo y ejecución de soluciones relacionadas con soporte técnico a usuarios y equipos.</p>	

81

Figura C.1- PNFI-Trayecto1: Sinopsis Unidades Curriculares

CONTENIDO SINÓPTICO					
PROGRAMA NACIONAL DE FORMACIÓN EN INFORMÁTICA					
Unidad Curricular:	Trayecto	Trimestre	Código	Unidades de Crédito	
ALGORÍTMICA Y PROGRAMACIÓN	1	1	PIAP114	4	
Módulo: ALGORÍTMICA Y PROGRAMACIÓN					
Horas de Trabajo del Estudiante Acompañado (HTEA)		Horas de Trabajo de Estudio Independiente (HTEI)		Total Horas de Trabajo del Estudiante (THTE)	
6 semanal		48 trimestral		120 trimestral	
SABERES		ESTRATEGIAS		EVALUACIÓN	
<p>Conocer 1. Algoritmo y programación. 2. Estándares de calidad en el diseño de algoritmos. 3. Datos y entidades primitivas. 4. Metodología para el análisis y planteamiento de problemas. 5. Programación Estructurada. 6. Programación Modular. 7. Técnicas de mantenimiento de programas.</p> <p>Hacer Construcción de algoritmos aplicados a problemas reales y de programas aplicando los principios fundamentales de la programación estructurada con el uso de estructuras de datos estáticas.</p> <p>Ser Responsable, proactivo, con pensamiento crítico, analítico, generador de soluciones.</p> <p>Convivir Interactuar con individuos o grupos utilizando los saberes y destrezas para el beneficio de la sociedad.</p> <p>Emprender Proyectos orientados al desarrollo y ejecución de soluciones relacionadas con soporte técnico a usuarios y equipos.</p>		<p>Ofrecer respuestas a problemas reales, desarrollando algoritmos y programas aplicando los principios fundamentales de la programación estructurada con el uso de estructuras de datos estáticas, mediante una serie de operaciones detalladas y no ambiguas, que conduzcan a la resolución de soluciones cumpliendo con estándares de calidad.</p>		<p>Prácticas formativas</p> <p>Prácticas sumativas</p>	
REFERENCIAS: Material Instruccional y documental relacionado con Algoritmos.					

106

Figura C.2- PNFI-Trayecto1-T1: Contenido Sinóptico de “Algorítmica y Programación”

⁹Documento Rector del PNFI, MPPES, noviembre 2008. Recuperado de <http://nixon-docspnfi.blogspot.com/>

CONTENIDO ANALÍTICO					
Unidad Curricular: ALGORÍTMICA Y PROGRAMACIÓN		Trayecto	Trimestre	Código	Unidades de Crédito
Módulo: ALGORÍTMICA Y PROGRAMACIÓN		1	1	PIAP114	4
Horas de Trabajo del Estudiante Acompañado (HTEA)		Horas de Trabajo de Estudio Independiente (HTEI)		Total Horas de Trabajo del Estudiante (THTE)	
6 semanal		72 trimestral		4 semanal	
				48 trimestral	
				10 semanal	
				120 trimestral	
SABERES		ESTRATEGIAS		RECURSOS	EVALUACIÓN
<p>Unidad 1: Algoritmo y Programas</p> <ul style="list-style-type: none"> o Concepto de Algoritmos y Programas. o Lenguaje algorítmico y de programación. o Partes de un algoritmo. o Características y elementos para construir un algoritmo. o Formas de representar un algoritmo: Lenguaje Natural, pseudocódigo y diagrama de flujo. o Traza de un algoritmo (corrida en frío). <p>Unidad 2: Estándares de Calidad en el Diseño de Algoritmos y Construcción de Programas</p> <ul style="list-style-type: none"> o Introducción a los estándares de calidad. o Forma de trazabilizar un algoritmo. o Formas y técnicas de documentar algoritmos y programas. o Introducción a la elaboración del manual del sistema, usuario y programas. o Técnicas de escritura y pruebas de algoritmos y programas. <p>Unidad 3: Datos y Entidades Primitivas</p> <ul style="list-style-type: none"> o Concepto y diferencia entre dato e información, tipos de datos. o Los Operadores: Concepto y tipos de operadores. o Las Expresión: concepto, tipos y evaluación de expresiones, Los Identificadores: Concepto, Reglas de escritura, las variables: clasificación de variables según su contenido (numéricas, lógicas y alfanuméricas) y su uso (de trabajo, contador y acumulador), constantes. 		<p>El programa de teoría se basará en clases semanales, en las que se incluyen las explicaciones de los temas y seminarios de problemas. El programa de prácticas se articula habitualmente en sesiones semanales de cuatro horas, y estará orientado a practicar en los laboratorios de computación, de la manera más directa posible, los temas explicados en las clases teóricas. También se dedican clases teóricas y prácticas a la introducción y discusión de ejercicios de mayor complejidad.</p> <p>El horario prevé dos horas semanales de teoría y dos grupos de laboratorio con cuatro horas de prácticas cada uno. La teoría se inicia con temas relacionados con la especificación y la verificación de algoritmos, paralelamente en las prácticas se desarrolla los conceptos de programación básicos.</p> <p>A partir de la unidad de programación estructurada teoría y práctica se trabajan simultáneamente los participantes resolverán en práctica de forma autónoma ejercicios con lenguajes de programación estructurada en los que tendrán que aplicar los conocimientos adquiridos.</p>		<p>Pizarra magnética</p> <p>Marcadores</p> <p>Material Educativo</p> <p>Computarizado: Material Instruccional, Software Instruccional</p> <p>Computador</p> <p>Proyector Multimedia</p> <p>Plataforma Tecnológica</p> <p>Laboratorio de computación</p> <p>Apoyo técnico</p>	<p>Evaluaciones formativas y sumativas donde se incluya parte teórica y parte práctica del módulo correspondiente a la unidad curricular</p> <p>La parte práctica en laboratorio se evaluará mediante dos mecanismos:</p> <p>1. Seguimiento de trabajo de prácticas mediante la entrega de algunos de los ejercicios propuestos en clase</p> <p>2. Un mini-proyecto de programación a entregar al final de trimestre</p>

107

Figura C.3- PNFI-Trayecto1-T1: Contenido Analítico de “Algorítmica y Programación”

CONTENIDO ANALÍTICO					
<p>Unidad 4: Metodología para el Análisis y Planteamiento de Problemas</p> <ul style="list-style-type: none"> o Identificación del Problema. o Identificación de los datos necesarios (entradas). o Identificación de los datos a obtener (salidas). o Descripción de las operaciones a utilizar (cálculos). o Descripción de los pasos para llegar a la solución (procesos). <p>Unidad 5: Programación Estructurada</p> <ul style="list-style-type: none"> o Teoremas de la programación estructurada. o Estructuras de control: <ul style="list-style-type: none"> o Estructuras de decisión: concepto, tipos y sintaxis: Condicional, simples, dobles, múltiples, anidadas y selectiva. o Estructuras de control iterativas: Concepto y tipos, Ciclo Mientras y Repetir; sintaxis y métodos para salir del ciclo. Ciclo Para; Sintaxis. Ciclos anidados. <p>Unidad 6: Programación Modular</p> <ul style="list-style-type: none"> o Funciones y procedimientos: Definición y Declaración. o Ambito de variables: Datos locales y globales. o Llamada de una función y pase d parametros. <p>Unidad 7: Técnicas de Mantenimiento de Programas</p> <ul style="list-style-type: none"> o Técnica de prueba caja negra. o Reingeniería de programas. 		<p>Para cada tema teórico-práctico se dispondrá de una guía que permita la ejecución del mismo, y el será evaluado sobre un subconjunto representativo de estos ejercicios, los cuales deben ser defendidos de forma presencial ante el profesor-asesor.</p> <p>En las clases teóricas se desarrollaran cada uno de los temas con apoyo de material didáctico.</p> <p>Estas sesiones culminan con la propuesta de un mini-proyecto en el que se ensamblan las diferentes unidades. En el cual el participante deberá entregar los ejercicios asignados y tendrán que ser discutidos de manera presencial con el profesor asesor en las horas prácticas.</p> <p>Durante todo el trimestre estarán activas las clases virtuales, los foros para la consulta y resolución de dudas, como herramienta complementaria a la tutoría presencial.</p>			
<p>REFERENCIAS:</p> <p>1. L. Joyanes, Fundamentos de programación, Ed. McGraw-Hill, 1996, 2002.</p> <p>2. G. Bassard, P. Bratley, Fundamentos de algoritmia, Ed. Prentice-Hall, 1998.</p>					

108

Figura C.4- PNFI-Trayecto1-T1: Continuación Contenido Analítico de “Algorítmica y Programación”

CONTENIDO SINÓPTICO					
PROGRAMA NACIONAL DE FORMACIÓN EN INFORMÁTICA					
Unidad Curricular: ALGORÍTMICA Y PROGRAMACIÓN	Trayecto	Trimestre	Código	Unidades de Crédito	
Módulo: PROGRAMACIÓN I	1	2	PIAP124	4	
Horas de Trabajo del Estudiante Acompañado (HTEA)	Horas de Trabajo de Estudio Independiente (HTEI)		Total Horas de Trabajo del Estudiante (THTE)		
6 semanal	72 trimestral		4 semanal	48 trimestral	10 semanal 120 trimestral
SABERES			ESTRATEGIAS	EVALUACION	
<p>Conocer 1. Arreglos. 2. Tratamiento de cadenas de caracteres. 3. Estructuras de Registros. 4. Punteros. 5. Archivos. 6 La recursividad.</p> <p>Hacer Construir programas con estructuras de datos estáticas y dinámicas básicas, que permitan aplicar métodos de búsqueda y ordenación de elementos sobre problemas específicos.</p> <p>Ser Responsable, Corresponsable, Autónomo, Participativo, Creativo, Comunicativo, proactivo, con pensamiento crítico, analítico, generador de soluciones, abierto al cambio.</p> <p>Convivir Con equipos de desarrolladores de software, preservando al ser humano, al ambiente y a la sociedad.</p> <p>Emprender Proyectos orientados al desarrollo de componentes de software.</p>			Encuentros de los participantes con el profesor asesor, lo que permitirá el desarrollo de componentes de software, utilizando material instruccional que facilite su ejecución de manera teórica práctica, en los laboratorios de computación.	Prácticas formativas Prácticas sumativas	
REFERENCIAS: Fuentes documentales y material instruccional relacionados con programación estructurada y estructura de datos.					




109

Figura C.5- PNFI-Trayecto1-T2: Contenido Sinóptico de “Algorítmica y Programación”

CONTENIDO ANALÍTICO					
Unidad Curricular: ALGORÍTMICA Y PROGRAMACIÓN	Trayecto	Trimestre	Código	Unidades de Crédito	
Módulo: PROGRAMACIÓN I	1	2	PIAP124	4	
Horas de Trabajo del Estudiante Acompañado (HTEA)	Horas de Trabajo de Estudio Independiente (HTEI)		Total Horas de Trabajo del Estudiante (THTE)		
6 semanal	72 trimestral		4 semanal	48 trimestral	10 semanal 120 trimestral
SABERES			ESTRATEGIAS	RECURSOS	EVALUACION
<p>Unidad 1: Arreglos</p> <ul style="list-style-type: none"> o Los arreglos: Definición, clasificación y operaciones básicas. o Métodos de Ordenamiento o Métodos de búsqueda <p>Unidad 2: Tratamiento de Cadenas de Caracteres</p> <ul style="list-style-type: none"> o Definición, funciones asociadas y operaciones con cadenas. <p>Unidad 3: Estructuras de Registros</p> <ul style="list-style-type: none"> o Definición, declaración y acceso o Estructura y arreglo <p>Unidad 4: Punteros</p> <ul style="list-style-type: none"> o Definición, declaración, operadores y operaciones o Punteros y funciones o Punteros y estructuras <p>Unidad 5: Archivos</p> <ul style="list-style-type: none"> o Concepto, estructura y acceso. o Tipos de archivo: datos y de textos. o Métodos para realizar la gestión de archivos. <p>Unidad 6: Recursividad</p> <ul style="list-style-type: none"> o Fundamentos teóricos: <ul style="list-style-type: none"> • Definición, ámbito de aplicación, utilidad o Ventajas y desventajas de la recursividad. o Diseño y Escritura de programas recursivos. 			<p>El programa de teoría se basará en clases semanales, en las que se incluyen las explicaciones de los temas y seminarios de problemas. El programa de prácticas se articula habitualmente en sesiones semanales de cuatro horas, y estará orientado a practicar en los laboratorios de computación, de la manera más directa posible, los temas explicados en las clases teóricas. También se dedican clases teóricas y prácticas a la introducción y discusión de ejercicios de mayor complejidad.</p> <p>El horario prevé dos horas semanales de teoría y dos grupos de laboratorio con cuatro horas de prácticas. La teoría se inicia con temas relacionados con la especificación y la verificación de algoritmos, paralelamente en las prácticas se desarrolla los conceptos básicos de programación.</p>	<p>Pizarra magnética Marcadores Material Educativo Computarizado: Material Instruccional, Software Instruccional Computador Proyector Multimedia Plataforma Tecnológica Laboratorio de computación Apoyo técnico</p>	<p>Evaluaciones formativas y sumativas donde se incluya parte teórica y parte práctica del módulo correspondiente a la unidad curricular</p> <p>La parte práctica en laboratorio se evaluará mediante dos mecanismos:</p> <ol style="list-style-type: none"> 1. Seguimiento de trabajo de prácticas mediante la entrega de algunos de los ejercicios propuestos en clase 2. Un mini-proyecto de programación a entregar al final de trimestre
REFERENCIAS: Fuentes documentales y material instruccional relacionados con programación estructurada y estructura de datos.					

110

Figura C.6- PNFI-Trayecto1-T2: Contenido Analítico Unidad “Algorítmica y Programación”

  					
CONTENIDO SINÓPTICO					
PROGRAMA NACIONAL DE FORMACIÓN EN INFORMÁTICA					
Unidad Curricular: ALGORÍTMICA Y PROGRAMACIÓN	Trayecto	Trimestre	Código	Unidades de Crédito	
Módulo: PROGRAMACIÓN II	1	3	PIAP134	4	
Horas de Trabajo del Estudiante Acompañado (HTEA)		Horas de Trabajo de Estudio Independiente (HTEI)		Total Horas de Trabajo del Estudiante (THTE)	
6 semanal		72 trimestral		4 semanal	
		48 trimestral		10 semanal	
				120 trimestral	
SABERES		ESTRATEGIAS		EVALUACIÓN	
<p><i>Conocer</i> 1. Implementación de programas con estructuras de datos dinámicas. 2. Listas enlazadas. 3. Pilas. 4. Colas. 5. Árboles.</p> <p><i>Hacer</i> Construir programas aplicando las estructuras de datos dinámicas.</p> <p><i>Ser</i> Responsable, proactivo, con pensamiento crítico, analítico, generador de soluciones.</p> <p><i>Convivir</i> Interactuar con individuos o grupos utilizando los saberes y destrezas para el beneficio de la sociedad.</p> <p><i>Emprender</i> Proyectos orientados al desarrollo de algoritmos y programas con estructuras dinámicas.</p>		Ofrecer respuestas a problemas reales, desarrollando programas con estructuras dinámicas, que conduzcan a la resolución de soluciones cumpliendo con estándares de calidad.		Prácticas formativas Prácticas sumativas	
REFERENCIAS: Material Instruccional y documental de estructura de datos dinámicas.					

111

Figura C.7- PNFI-Trayecto1-T3: Contenido Sinóptico “Algorítmica y Programación”

  							
CONTENIDO ANALÍTICO							
Unidad Curricular: ALGORÍTMICA Y PROGRAMACIÓN	Trayecto	Trimestre	Código	Unidades de Crédito			
Módulo: PROGRAMACIÓN II	1	3	PIAP134	4			
Horas de Trabajo del Estudiante Acompañado (HTEA)		Horas de Trabajo de Estudio Independiente (HTEI)		Total Horas de Trabajo del Estudiante (THTE)			
6 semanal		72 trimestral		4 semanal			
		48 trimestral		10 semanal			
				120 trimestral			
SABERES		ESTRATEGIAS		RECURSOS			
<p>Unidad 1: Implementación de Programas con Estructuras de Datos Dinámicas</p> <p>Unidad 2: Listas Enlazadas</p> <ul style="list-style-type: none"> o Concepto y clasificación o Listas simplemente enlazadas: fundamentos teóricos, clasificación, operaciones básicas: <ul style="list-style-type: none"> • Declaración, punteros de cabecera y cola, operador de selección, inserción, búsqueda y eliminación de elementos) o Listas doblemente enlazadas y Circulares: o Declaración, recorrido, inserción y eliminación de elementos. <p>Unidad 3: Pilas</p> <ul style="list-style-type: none"> o Definición, especificaciones y tipo de pilas implementadas con arreglos y punteros. <p>Unidad 4: Colas</p> <ul style="list-style-type: none"> o Definición, especificaciones y Tipo de colas implementadas con arreglos. <p>Unidad 5: Árboles</p> <ul style="list-style-type: none"> o Árboles Generales y binarios. o Estructura y representación de un árbol binario. o Árboles de expresión. o Recorrido de un árbol. o Árboles binario de búsqueda: operaciones 		<p>El programa de teoría se basará en clases semanales, en las que se incluyen las explicaciones de los temas y seminarios de problemas. El programa de prácticas se articula habitualmente en sesiones semanales de cuatro horas, y estará orientado a practicar en los laboratorios de computación, de la manera más directa posible, los temas explicados en las clases teóricas. También se dedican clases teóricas y prácticas a la introducción y discusión de ejercicios de mayor complejidad.</p> <p>El horario prevé dos horas semanales de teoría y dos grupos de laboratorio con cuatro horas de prácticas. La teoría se inicia con temas relacionados con la especificación y la verificación de algoritmos, paralelamente en las prácticas se desarrolla los conceptos básicos de programación.</p>		<p>Pizarra magnética Marcadores Material Educativo Computarizado: Material Instruccional, Software Instruccional Computador Proyector Multimedia Plataforma Tecnológica Laboratorio de computación. Apoyo técnico</p>		<p>Evaluaciones formativas y sumativas donde se incluya parte teórica y parte práctica del módulo correspondiente a la unidad curricular</p> <p>La parte práctica en laboratorio se evaluará mediante dos mecanismos:</p> <ol style="list-style-type: none"> 1. Seguimiento de trabajo de prácticas mediante la entrega de algunos de los ejercicios propuestos en clase 2. Un mini-proyecto de programación a entregar al final de trimestre 	
REFERENCIAS: Material Instruccional y documental de estructura de datos dinámicas.							

112

Figura C.8- PNFI-Trayecto1-T3: Contenido Analítico “Algorítmica y Programación”

D. Insumos Elaboración Pruebas Diagnóstico: Definición Operativa de las Variables, Dimensiones e Indicadores

Objetivo	Variable	Dimensiones	Indicadores
Determinar el grado de conocimientos básicos y habilidades en análisis y razonamiento lógico-matemático, que tienen los estudiantes del Programa Nacional de Formación en Informática requeridos para el curso propedéutico.	Conocimientos Básicos en Matemática que posee el estudiante	Conjuntos Numéricos	Naturales
			Enteros
			Reales
			Imaginarios
		Cálculo Aritmético (Conjunto Entero)	Suma
			Resta
			Multiplicación (suma resumida)
			División (resta resumida)
		Algoritmos Aritméticos	Determinación de Factores Primos
			Cálculo del Mínimo Común Múltiplo
			Cálculo del Máximo Común Divisor
		Algebra Elemental	Conceptos: Variable, Incógnita, Coeficiente, Índice y Raíz
			Operaciones algebraicas (suma, resta, multiplicación, división, potenciación, radicación o raíces)
			Operaciones de relación (igual, diferente, mayor, menor, mayor o igual, menor o igual)
			Identificación de Ecuaciones: Identificación y construcción
			Transformación de expresiones utilizando el lenguaje algebraico (expresiones algebraicas, ecuaciones e inecuaciones)
Algoritmos Algebraicos	Potencia cuadrada de una suma		
	Potencia cuadrada de una resta		
	Raíz de una ecuación de segundo grado		
	Resolución de Polinomios		

Objetivo	Variable	Dimensiones	Indicadores
Determinar el grado de conocimientos básicos y habilidades en análisis y razonamiento lógico-matemático, que tienen los estudiantes del Programa Nacional de Formación requeridos para el curso propedéutico.	Razonamiento lógico-matemático: Aspectos de la Lógica Proposicional que maneja informalmente el estudiante	Valores de Verdad (lógica bivalente)	Verdad
			Falso
			Valores Binarios
		Operadores Lógicos	Negación (no)
			Conjunción (y)
			Disyunción (o)
			Condicional (si)
			Bicondicional (si y sólo si)
			Disyunción opuesta (ni... ni...)
			Disyunción excluyente (o esto... o aquello...)
	Expresiones en lenguaje natural		
	Razonamiento lógico-matemático: Grado de Resolución de Problemas que posee el estudiante	Partes de un Problemas	Planteamiento
			Identificación de Datos
			Determinación del Procedimiento (Cálculos)
			Obtención de la Solución (Resultados)
		Problemas Matemáticos de Aritmética	Simple (una sola operación)
			Baja dificultad
			De dificultad intermedia
			De alta dificultad
	Complejos (razonamiento y cálculo)		

E. Elementos Claves para el Diseño del Curso Propedéutico Para Enseñar Algoritmos

Objetivo	Variable	Dimensiones	Indicadores
Identificar los elementos claves para diseñar un curso propedéutico para la enseñanza de algoritmos, considerandolo expuesto en el programa de la unidad curricular “Algorítmica y Programación” del Programa Nacional de Formación en Informática.	Organización de la Unidad Curricular: Módulo I: Algorítmica	Objetivos	General
			Terminal (de cada parte o unidad)
			Específicos
		Contenido Programático	Cantidad de partes o unidades
			Contenido de cada parte o unidad
			Referencias Bibliográficas
	Estrategias Metodológicas utilizadas por el docente	Técnicas	Enseñanza individual
			Enseñanza grupal
		Recursos	Material Impreso
			Material Digital
			Pizarra y Marcadores
			Computador
			Software para la enseñanza de algoritmos
Software para el Desarrollo de Algoritmos			
Lenguajes de programación			
Actividades	Del Docente		
	Del Estudiante: Individuales Grupales		