

UNIVERSIDAD DE MANIZALES

Módulo de Aprendizaje Desarrollo de Algoritmos Semestre I

Manizales agosto de 2011

Contenido

Presentación	6
Contextualización	7
Situación Problemática	7
Planteamiento.	7
Ideograma.....	8
Explicación del Ideograma.....	9
Conocimientos previos.	9
Competencias del módulo	12
Evaluación de los aprendizajes	14
Material Pedagógico de apoyo.....	21
1 TECNICA PARA RESOLVER PROBLEMAS CON AYUDA DEL COMPUTADOR.	27
Competencias y tabla de saberes.....	27
Tabla de saberes.....	27
1.1 ANÁLISIS Y CLASIFICACIÓN DEL ENUNCIADO DEL PROBLEMA EN SUS ELEMENTOS: ...	29
1.2 DIAGRAMAS DE ENTRADA DE DATOS-PROCESO-SALIDA DE INFORMACIÓN:	30
1.3 ANÁLISIS DE PROCESOS ARITMÉTICOS:.....	31
Actividades de aprendizaje	52
Evaluación de los aprendizajes.....	56
Resultados del aprendizaje	56
2.1 ESTRUCTURAS DE SELECCIÓN:	59
Evaluación de los aprendizajes Unidad 2	74
Resultados del aprendizaje:	74
CODIFICACIÓN Y LENGUAJES DE PROGRAMACIÓN	76
Competencias y tabla de saberes unidad 3.....	76
3 CODIFICACIÓN Y LENGUAJES DE PROGRAMACIÓN.....	77
3.1 Pasos para llevar un llevar la solución a un problema del pseudocódigo al código fuente:	77
Actividades de aprendizaje	91
Evaluación de los aprendizajes Unidad 3	94

Resultados del aprendizaje	94
Grafica 1 Entrada-Proceso-Salida	31
Grafica 2 Entrada-Proceso-Salida para el Ejemplo	31
Grafica 3 Diseño Interfaz Hombre Máquina para el Ejemplo	33
Grafica 4 Expresión algebraica.....	40
Grafica 5 Diagrama de Flujo de Datos para el Ejemplo	42
Grafica 6 Diagrama de flujo decisión simple.....	61
Grafica 7 Diagrama de flujo decisión doble	61
Grafica 8 Diagrama de flujo decisión compuesta	62
Grafica 9 Diagrama de flujo decisión múltiple	63
Grafica 10 Diagrama de flujo decisión secuencial	65
Grafica 11 Diagrama de flujo decisión anidado.....	66
Grafica 12 Diagrama de flujo, Estructuras de Repetición: Para	68
Grafica 13 Diagrama de flujo, Estructuras de Repetición: Ejecute Mientras.....	69

Índice de tablas

Tabla 1 Elementos de una Interfaz	32
Tabla 2 Tabla de Datos.....	38
Tabla 3 Tabla de Datos para el Ejemplo	39
Tabla 4 Operadores Aritméticos.....	40
Tabla 5 Jerarquía de los Operadores Aritméticos.....	40
Tabla 6 Conversión de Expresiones Aritméticas a Computacionales.....	41
Tabla 7 Símbolos Diagramas de Flujo.....	41
Tabla 8 Plantilla Prueba de Escritorio	43
Tabla 9 Plantilla Prueba de Escritorio para el Ejemplo.....	43
Tabla 10: tabla de relación de operadores.....	59
Tabla 11 lista de operadores relacionales	59
Tabla 12 operadores lógicos.....	60

FICHA TÉCNICA

MODULO DE APRENDIZAJE	VERSIÓN	TIPO DE MÓDULO	FECHA
Desarrollo de algoritmos	1	Disciplinar Profesional	2011
PROGRAMA		SEMESTRE	
TECNICO PROFESIONAL EN CONFIGURACION DE CONTENIDOS COMERCIALES WEB		I	
DISEÑADO POR			
Ing. Carlos Betancourt Correa Ingeniero de Sistemas Especialista en Telecomunicaciones Magister en Educación Docencia			
CREDITOS		3	Nº SEMANAS
HORAS APRENDIZAJE AUTONOMO		114	
HORAS ACOMPAÑAMIENTO TUTORIAL		30	9
NORMAS Y REFERENTES			
Basado en la norma de competencia			
220501007 Desarrollar el sistema que cumpla con los requerimientos de la solución informática			

Nombre del módulo DESARROLLO DE ALGORITMOS	
Competencia general del módulo	
Diseñar estructuras de datos eficientes para un determinado problema, teniendo en cuenta principios de abstracción, condiciones técnicas y lenguajes especificados	
Presentación del módulo.	
<p>Presentación</p> <p>En la actualidad todas las empresas en cualquiera de los sectores productivos manejan grandes volúmenes de información de manera digital, es por esto que se requieren tanto personas que operen estos sistemas como personas que participen en los procesos de desarrollo que administran dicha información.</p> <p>El Desarrollo de Algoritmos es el inicio de la comprensión y representación de un problema de información para su posterior propuesta de solución, por tanto la fundamentación básica les va a permitir desarrollar habilidades superiores de pensamiento para abstraer, analizar, comprender y proponer soluciones pertinentes a problemas de manejo de información, requisitos indispensables para que las empresas mantengan los niveles de competitividad requeridos en una sociedad en la cual la toma de decisiones depende de la calidad de los datos y la información.</p> <p>Señores estudiantes, este módulo de Desarrollo de Algoritmos tiene como objetivo iniciarlos en los fundamentos básico de la programación de computadores por medio del desarrollo del pensamiento lógico. Este módulo consta de 3 créditos académicos lo que significa que en aproximadamente 144 horas distribuidas entre aprendizaje individual, colaborativo y con acompañamiento directo del docente ustedes apropiaran las competencias propuestas.</p> <p>El eje temático del Desarrollo de Algoritmos gira alrededor de 3 unidades:</p> <p>Solución de problemas de manejo de información como punto de partida para el análisis, clasificación y diseño de soluciones de baja complejidad; seguidamente se incorporan elementos tales como las estructuras de selección y repetición para soluciones de mediana complejidad y finalizamos con la codificación en un lenguaje de programación.</p>	

Contextualización

Situación Problemática

El manejo de volúmenes de información hace parte de la cotidianidad de las pequeñas, medianas y grandes empresas, ¿Cómo solucionar los problemas de manejo de información en las empresas? ; ¿Cómo se representan estos problemas?; ¿Cómo se llevan las soluciones a la etapa computacional?. **Alrededor de estos interrogantes se desarrollará el contenido del módulo.**

Planteamiento.

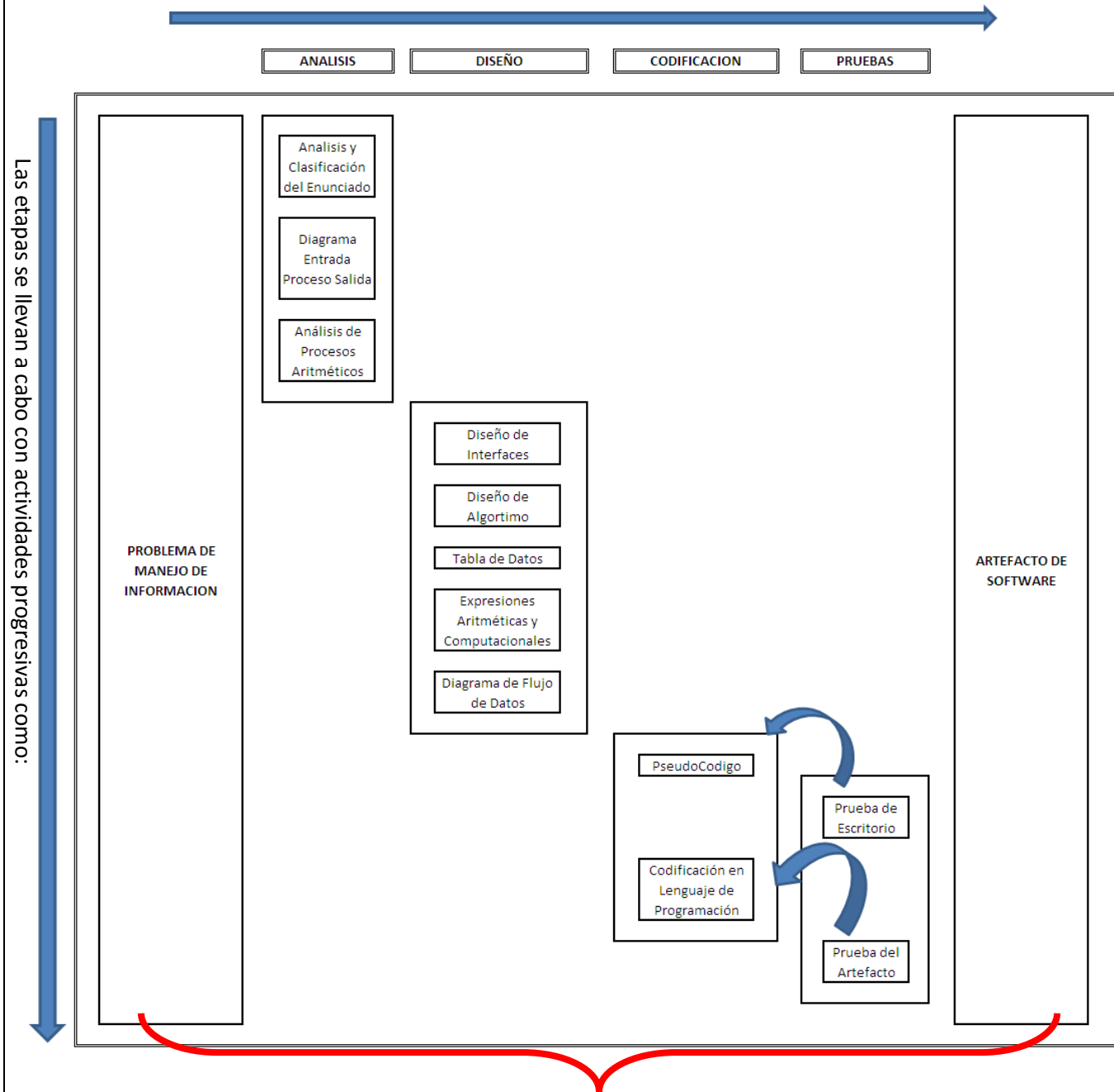
La programación de computadores está asociada con la evolución histórica del computador. En sus inicios era compleja ya que se desarrollaba mediante el lenguaje de máquina o bajo nivel(Assembler), a medida que el computador fue evolucionando paralelamente los lenguajes fueron subiendo de nivel, esto es más cercanos al lenguaje humano, facilitando y por tanto masificando su uso. En la actualidad la programación se ha hecho más sencilla y han surgido lenguajes más intuitivos y direccionados a facilitar y agilizar el desarrollo de software en beneficio de soluciones optimas y de calidad.

El desarrollo de algoritmos se ha constituido en una forma eficiente para definir como resolver un problema utilizando una secuencia lógica que parte de un enunciado empírico tomado del contexto para luego convertirse en un lenguaje más elaborado y entendible por los sistemas de computo.

Para quien pretenda desempeñarse en las áreas de la tecnología es de suma importancia comprender la lógica y los fundamentos de la programación de computadoras como la base de funcionamiento.

Ideograma

El ciclo de desarrollo de un programa tiene las etapas de:



Los problemas de manejo de la información de un contexto, se pueden sistematizar aplicando las etapas y necesarias para la producción de artefactos de software.

Explicación del Ideograma

Para proponer soluciones a problemas de manejo de información mediante la utilización de los algoritmos y las computadoras se requiere transitar por las siguientes fases: Análisis, Diseño, Codificación y Pruebas.

Desglosando cada una de las fases que se deben realizar de forma secuencial encontramos:

La fase de Análisis está compuestas por tres etapas: Análisis y Clasificación del enunciado, Elaboración de Diagramas de Entrada-Proceso y Salida y Análisis de los procesos Aritméticos.

La fase de Diseño está compuestas por cinco etapas: Diseño de Interfaces, Diseño de Algoritmos, Tabla de Datos, Expresiones Aritméticas y Computacionales y Diagrama de Flujo de Datos.

La fase de Codificación está compuestas por dos etapas: Pseudocódigo y Codificación en Lenguaje de Programación.

Se culmina el proceso con la fase de Pruebas, compuestas por Pruebas de Escritorio y Pruebas de Artefacto.

Señor estudiante, si usted es riguroso con la comprensión y aplicación de cada una de estas fases, el éxito estará garantizado.

Conocimientos previos.

Para que el estudiante asimile en un mayor porcentaje los contenidos del módulo, es deseable que aplique las siguientes actividades para que tenga una referencia de su estado de conocimiento actual referente al módulo.

Actividad nro 1:

- Nombre de la actividad: Taller individual de autoevaluación de conocimientos previos:
 - Operaciones aritméticas básicas
 - Razonamiento lógico
 - Razonamiento Abstracto
 - Interpretar preposiciones
- Nombre del recurso: Cuestionario Moodle
- Indicaciones paso a paso: en la plataforma Educativa encontrará un recurso denominado encuesta de conocimientos previos, ingrese y responder la siguiente encuesta basados en sus conocimientos.
- Producto a entregar: cuestionario en línea
- Preguntas
 1. Indique la respuesta de cada una de las siguientes operaciones aritméticas (Tenga en cuenta la prioridad de los operadores):

- $30 + 15 * 3 - 2$
- $3 * 18 + (15 / 5) 2 + 17$
- $((5 + 8 - 6) * (12 - 24) / 2)$
- $8 - (-15 + 23) * 2$
- $(17 + 3)^3 - (2 * 4 / 6)^4$

2. Realice las siguientes Operaciones con fraccionarios

- $\left(3 + \frac{1}{4}\right) - \left(2 + \frac{1}{6}\right) =$
- $\frac{15}{3} + \frac{36}{3} - \frac{24}{2}$
- $3^4 - 15 + 16 * \sqrt{8}$
- $\sum_{i=1}^{30} i * 2$
- $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

3. Realice los siguientes ejercicios de Razonamiento Abstracto

- | | | | |
|---|---|---|-----|
| ↙ | ↘ | ↖ | ??? |
|---|---|---|-----|

 a:

↙

 b:

↘

 c:

↗

- | | | |
|---|---|---|
| ⬆ | ⬇ | ⬅ |
| ⬆ | ⬇ | ⬅ |
| ⬆ | ⬇ | ⬅ |

 a:

⬆

 b:

⬇

 c:

⬅

- | | | | | | |
|---|---|---|---|---|-----|
| 9 | 7 | 8 | 6 | 7 | ??? |
|---|---|---|---|---|-----|

 a:

7

 b:

5

 c:

8

- | | | | | | |
|---|---|---|---|---|-----|
| 2 | 4 | 2 | 6 | 4 | ??? |
|---|---|---|---|---|-----|

 a:

6

 b:

4

 c:

10

- Fecha de inicio: semana 1
- Fecha límite: semana 1

Actividad nro 2:

- Nombre de la actividad: Taller individual de autoevaluación de conocimientos previos:
- Nombre del recurso: Cuestionario Moodle
- Indicaciones paso a paso: en la plataforma Educativa encontrará un recurso denominado Taller de Conocimientos Previos, ingrese y responder la siguiente encuesta basados en sus conocimientos.
 1. Describa de manera detallada cada uno de los pasos que usted realiza para realizar una llamada telefónica.
 2. Resuelva el siguiente Sudoku

		1			7	5	3	6
6	8		2	1	5		4	
9	5				6	8		
8		5	7	2			6	4
7	4			6		3	5	9
	1			5	9			
	7		6		2			3
1		4		3	8			2
2	3		1			6	8	5

3. Ordene de mayor a menor por estatura los integrantes de su familia de acuerdo al genero.

4. Calcule los ingresos y los gastos mensuales de su familia y emita un juicio sobre su estado económico mensual.

- Producto a entregar: cuestionario en línea
 - Fecha de inicio: semana 1
 - Fecha límite: semana 1

Competencias del módulo			
Cognoscitivas	Procedimentales	Actitudinales	Integrativas
<p>Conceptualizar sobre la terminología básica relacionada con la solución de problemas en forma algorítmica y sus respectivas representaciones. Definir los conceptos básicos en las etapas del ciclo de desarrollo de un programa</p> <p>Identificar los problemas que pueden ser resueltos con el apoyo del computador .</p>	<p>Representar en un diagrama la entrada y salida de datos del problema que desea resolver. Diseñar algoritmos funcionales empleando estructuras condicionales o de repetición según el tipo de problema a resolver.</p> <p>Realizar diagramas de flujo para representar la solución de problemas utilizando la notación establecida. Escribir el pseudocódigo que permita representar algoritmos propuestos para la solución del problema</p> <p>Diseña algoritmos funcionales empleando estructuras condicionales o de repetición según el tipo de problema a resolver.</p> <p>Realizar diagramas de flujo para representar la solución de problemas utilizando la notación establecida. Escribir el pseudocódigo que permita representar algoritmos propuestos para la solución del problema</p> <p>Definir los conceptos básicos en las etapas del ciclo de desarrollo de un programa.</p> <p>Identificar las características principales de un lenguaje de</p>	<p>Proponer alternativas en la gestionar la información para movilizarse dentro de la dinámica de actualización permanente en el área de las tecnologías de la información y la comunicación</p> <p>Respetar la propiedad intelectual en la realización de las actividades de acuerdo con las normativas establecidas</p> <p>Establecer el ritmo de trabajo para cumplir en forma pertinente con las actividades propuestas en los tiempos establecidos.</p> <p>Exponer el punto de vista argumentado sobre las situaciones Problemáticas apoyado en la fundamentación teórico práctica de su formación profesional.</p> <p>Tomar decisiones frente a la resolución de problemas y ejecución de actividades apoyado en sus conocimientos y habilidades.</p> <p>Comunicar sus ideas e inquietudes relacionadas con los aspectos académicos utilizando los protocolos y recursos disponibles.</p>	<p>Mantener comunicación efectiva a nivel interno y externo de la organización mediante los recursos tecnológicos empleando lenguaje específico del área y protocolos de comunicación.</p> <p>Aplicar pensamiento lógico para proponer soluciones en el uso de los recursos tecnológicos y su aplicación en la gestión de la información basado en las necesidades de la organización y los recursos disponibles.</p>

	<p>programación y su función en el desarrollo de programas.</p> <p>Codificar artefactos de software empleando el paradigma orientado a objetos y el lenguaje de programación definido</p> <p>Manipular el código fuente teniendo en cuenta las especificidades de cada uno de ellos.</p> <p>Documentar artefactos de software y elaborar sus respectivos manuales de acuerdo con el paradigma empleado</p>	<p>Participar en las discusiones de grupo y en los procesos de socialización de los trabajos con capacidad argumentativa</p> <p>Interactuar (con los participantes, tutor, docente y contenidos) en el aula virtual a través de los recursos y actividades para alcanzar las competencias propuestas.</p> <p>Respetar y valora las posiciones divergentes dentro del equipo, buscando a la vez conciliarlas para integrar los aportes positivos y mejorar los logros del grupo</p>	
--	--	--	--

Valoración de las competencias

Primer nivel (Dimensión interpretativa)	Identifica y conceptualiza cada uno de los componentes que implica la solución de problemas mediante el desarrollo de algoritmos.
Segundo nivel (Dimensión argumentativa)	Detalla y explica aquellos procesos que se involucran en el tránsito de un problema desde su etapa precomputacional hasta la computacional.
Tercer nivel (Dimensión propositiva)	Genera y plantea ideas, que permitan dar solución a problemas de manejo de información de una manera coherente y lógica.
Cuarto nivel (Dimensión apropiación)	Apropia técnicas de solución a situaciones planteadas relacionadas con la obtención, análisis y seguimiento de requerimientos

Evaluación de los aprendizajes

COGNITIVAS							
COMPETENCIA	CRITERIOS DE EVALUACIÓN	0	1	2	3	4	5
Conceptualizar sobre la terminología básica relacionada con la solución de problemas en forma algorítmica y sus respectivas representaciones.	Identifica los conceptos propios de los algoritmos						
	Establece los pasos lógicos para desarrollar algoritmos						
Definir los conceptos básicos en las etapas del ciclo de desarrollo de un programa.	Describe las etapas del ciclo de desarrollo						
	Caracteriza los hitos de cada etapa						
Identificar los problemas que pueden ser resueltos con el apoyo del computador.	Reconoce situaciones problemáticas que pueden ser sistematizadas						
	Describe alternativas de soluciones a situaciones problemáticas						
Identificar las características principales de un lenguaje de programación y su función en el desarrollo de programas	Define los elementos básico de un lenguaje de programación						
	Reconoce las estructuras de los lenguajes de programación.						

Procedimentales							
COMPETENCIA	CRITERIOS DE EVALUACIÓN	0	1	2	3	4	5
Representar en un diagrama la entrada y salida de datos del problema que desea resolver.	Utiliza la notación para diagramas de entrada, proceso y salida de datos.						
	Representa estructuras condicionales						
	Representa estructuras repetitivas						
Diseña algoritmos funcionales empleando estructuras condicionales o de repetición según el tipo de problema a resolver.	Aplica las estructuras básicas de programación a la solución de problemas.						
	Emplea la notación especificada en el diseño de algoritmos						
Realizar diagramas de flujo para representar la solución de problemas utilizando la notación establecida.	Utiliza la notación para diagramas de flujo						
	Representa un mediante el diagrama de flujo						
	Escribe el diagrama de flujo en el pseudocódigo correspondiente						

Escribir el pseudocódigo que permita representar algoritmos propuestos para la solución del problema	Utiliza la estructura lógica del pseudocódigo						
Codificar artefactos de software empleando el paradigma orientado a objetos y el lenguaje de programación definido	Escribe el código fuente correspondiente al pseudocódigo.						
	Verifica la funcionalidad producto de la codificación del artefacto de software						
Documentar artefactos de software y elaborar sus respectivos manuales de acuerdo con el paradigma empleado	Elabora la documentación técnica del artefacto codificado						
	Escribe los comentarios necesarios para documentar el código fuente						

Actitudinales y transversales							
COMPETENCIA	CRITERIOS DE EVALUACIÓN	0	1	2	3	4	5
Proponer alternativas en la gestión la información para movilizarse dentro de la dinámica de la actualización permanente en el área de las tecnologías de la información y la comunicación.	Referencia fuentes de información relacionadas con los temas tratados						
	Amplia la información de las temáticas mediante consultas en otras fuentes						
Respetar la propiedad intelectual en la realización de las actividades de acuerdo con las normativas establecidas	Referencia las fuentes de donde obtuvo la información de acuerdo con las normas de derechos de autor.						
	Realiza las actividades propuestas con sentido ético, sin evidenciar copia de otros compañeros.						
	Presentar los trabajos en forma ordenada y de acuerdo a las normas y requerimientos exigidos.						
Establecer el ritmo de trabajo para cumplir en forma pertinente con las actividades propuestas en los tiempos establecidos	Realiza las actividades según el cronograma establecido.						
	Participa en los encuentros síncronos siguiendo los protocolos definidos.						
Aplicar el pensamiento lógico en la elaboración conceptual y argumentativa de soluciones	Abstrae problemas del mundo real y proponer soluciones mediante el diseño de software.						

relacionadas con el manejo de la información a través de los recursos tecnológicos	Argumenta sobre las alternativas de solución propuestas con base en los fundamentos teóricos.						
Tomar decisiones frente a la resolución de problemas y ejecución de actividades apoyado en sus conocimientos y habilidades.	Identifica varias alternativas de solución frente a la aplicación de algoritmos.						
	Decide la mejor alternativa de solución apoyado en sus conocimientos.						
Participar en las discusiones de grupo y en los procesos de socialización de los trabajos con capacidad argumentativa	Genera discusión con sus aportes en espacios asíncronos y encuentros síncronos. Respetar el punto de vista divergente de los demás estudiantes. Participa activamente en los encuentros síncronos programados en el módulo						
Comunicar sus ideas e inquietudes relacionadas con los aspectos académicos utilizando los protocolos	Utiliza los canales de comunicación disponibles en el aula para interactuar con el tutor y los estudiantes Comunica sus ideas e inquietudes relacionadas con los aspectos académicos utilizando los protocolos						
Respetar y valorar las posiciones divergentes dentro del equipo, buscando a la vez conciliarlas para integrar los aportes positivos y mejorar los logros del grupo	Plantea su punto de vista en forma argumentada.						
	Respetar los planteamientos divergentes de los compañeros						
Interactuar (con los participantes: tutor, docentes y contenidos) en el aula virtual a través de recursos y actividades para alcanzar las competencias establecidas.	Utiliza los recursos tecnológicos con habilidad Participa activamente en la actividad en el tiempo establecido. Respetar las opiniones de los demás, tratando con respeto las diferencias Interactúa a través de los medios tecnológicos dispuestos en el aula						

(Autoevaluación)

HABILIDADES	CRITERIOS DE EVALUACIÓN	0	1	2	3	4	5
Autoaprendizaje	Me apoyé en los contenidos del módulo de aprendizaje para alcanzar las competencias establecidas						
	Alcancé las competencias propuestas en el módulo de aprendizaje						
	Establecí técnicas que me permitieron aprender más fácilmente.						
Responsabilidad	Realice las actividades de aprendizaje cumpliendo con los criterios establecidos en cada una de ellas.						
	Cumplí con el programa de trabajo definido en el módulo						
	Realicé aportes pertinentes y oportunos a los espacios de discusión síncronos y asíncronos.						
	Participé en actividades de aprendizaje de colaborativas						
Interacciones	Mantuve una comunicación continua con el docente- tutor a través de los canales dispuestos en el aula.						
	Mantuve una comunicación continua con mis compañeros a través de medios síncronos y asincrónicos.						
	Solicité ayuda al docente-tutor para aclarar dudas y plantear inquietudes.						
Capacidad de búsqueda y análisis de la información	Consulté otras fuentes de información complementarias para profundizar las temáticas del componente.						
	Seleccioné la información teniendo en cuenta su procedencia y pertinencia						
	Respeté la propiedad intelectual						
Alcance de las competencias	Represente en un diagrama la entrada y salida de datos del problema que desea resolver.						
	Documente artefactos de software y elaborar sus respectivos manuales de acuerdo con el paradigma empleado						
	Codifique artefactos de software empleando el paradigma orientado a objetos y el lenguaje de programación definido						
	Realicé diagramas de flujo para representar la solución de problemas utilizando la notación establecida.						

	Diseñe algoritmos funcionales empleando estructuras condicionales o de repetición según el tipo de problema a resolver.						
	Escribí el pseudocódigo que permita representar algoritmos propuestos para la solución del problema						
	Escribí el pseudocódigo que permita representar algoritmos propuestos para la solución del problema						

GUIA DE ACTIVIDADES DE APRENDIZAJE

UNIDAD DE APRENDIZAJE	ESTRATEGIA DE APRENDIZAJE	ACTIVIDAD DE APRENDIZAJE		RECURSO	VALORACIÓN	FECHA LIMITE
UNIDAD 1 Desarrollo de algoritmos	ACTIVIDAD AUTONOMA	Act 2	Mapa mental de algoritmos	Freemiend Tarea moodle Préstamo en línea	5%	Primera Semana
		Act 4	Aplicación de la plantilla guía	Tarea moodle	5%	Segunda semana
	ACTIVIDAD COLABORATIVA	Act 5	Ejercicios de aplicación	Tarea de moodle	10%	Tercera semana
	ACOMPañAMIENTO TUTORIAL	Act 1	Presentación y contextualización del módulo	Video Conferencia Adobeconnect (3 horas)	5%	Primera Semana
		Act 3	Video Clase Fundamentos de algoritmos	Video Conferencia Adobeconnect (2 horas)	5%	Segunda semana
		Act 6	Retroalimentación de la unidad I	Video Conferencia Adobeconnect (3 horas)	5%	Tercera semana
UNIDAD DE APRENDIZAJE	ESTRATEGIA DE APRENDIZAJE	ACTIVIDAD DE APRENDIZAJE		RECURSO	VALORACIÓN	FECHA LIMITE
	ACTIVIDAD AUTONOMA	Act 2	Mapa mental de ciclos y estructuras	Freemiend Tarea moodle Préstamo en línea	5%	Cuarta Semana

UNIDAD 2 Ciclos y estructuras		Act 3	Aplicación de la plantilla guía	Tarea moodle	10%	Quinta semana
	ACTIVIDAD COLABORATIVA	Act 4	Ejercicios de aplicación	foro, tarea (4 horas para moderar el foro)	10%	Sexta semana
	ACOMPANIAMIENTO TUTORIAL	Act 1	Presentación de la unidad II	Video Conferencia Adobeconnect (3 horas)	5%	Cuarta Semana
		Act 5	Retroalimentación de la unidad I	Video Conferencia Adobeconnect (3 horas)	5%	Sexta semana
UNIDAD DE APRENDIZAJE	ESTRATEGIA DE APRENDIZAJE	ACTIVIDAD DE APRENDIZAJE		RECURSO	VALORACIÓN	FECHA LIMITE
UNIDAD 3 Codificación y lenguajes de programación	ACTIVIDAD AUTONOMA	Act 2	Mapa mental de lenguajes de programación	Freemind Tarea moodle Préstamo en línea	5%	séptima Semana
	ACTIVIDAD COLABORATIVA	Act 3	Ejercicios de aplicación	foro, tarea (4 horas para moderar el foro)	10%	Novena semana
	ACOMPANIAMIENTO TUTORIAL	Act 1	Presentación de la unidad II	Video Conferencia Adobeconnect (3 horas)	5%	Séptima Semana
		Act 4	Retroalimentación de la unidad I	Video Conferencia Adobeconnect (3 horas)	10%	Octava semana

Observaciones: **LA NUMERACION DE LAS ACTIVIDADES TIENE EL ORDEN LOGICO EN EL QUE EL ESTUDIANTE LAS DEBE REALIZAR**

Material Pedagógico de apoyo

Unidad 1

Lecturas requeridas

Lectura N° 1 Betancourt, C. (2011). Módulo Desarrollo de Algoritmos Unidad 1. Universidad de Manizales. Manizales: CEDUM.

Lectura N° 2 “RESOLUCIÓN DE PROBLEMAS CON COMPUTADORAS Y HERRAMIENTAS DE PROGRAMACIÓN”. En este capítulo del libro del Dr. Luis Joyanes Aguilar, encontrará varios métodos propuestos por el autor para realizar soluciones a problemas de manejo de información.

Lecturas Complementarias

Lectura N° 1 “METODOLOGÍA PARA LA RESOLUCIÓN DE PROBLEMAS ”. En este capítulo del libro del Dr. Omar Iván Trejos, encontrará la propuesta del autor para abordar la solución de problemas de forma algorítmica.

Lectura N° 2 “APRENDER A PROGRAMAR: ALGORITMOS Y FUNDAMENTOS DE PROGRAMACIÓN ORIENTADOS A LA INGENIERÍA Y CIENCIAS”. En este libro del autor Mario Rodríguez, encontrará un recorrido de cómo se puede resolver problemas con el apoyo de los lenguajes de programación y las computadoras.

Unidad 2

Lecturas requeridas

Lectura N° 1 Betancourt, C. (2011). Módulo Desarrollo de Algoritmos Unidad 2. Universidad de Manizales. Manizales: CEDUM.

Lectura N° 2 Capítulo 4. Estructuras Selectivas. En este capítulo del libro de Fundamentos de Programación del Dr. Luis Joyanes Aguilar, se presenta un recorrido sobre las estructuras básicas de programación referentes a las selecciones (decisiones simples – decisiones múltiples – decisiones anidadas).

Lectura N° 3 Capítulo 5. Estructuras Repetitivas. En este capítulo del libro de Fundamentos de Programación del Dr. Luis Joyanes Aguilar, se presenta un recorrido sobre las estructuras repetitivas(while – Do-While-For).

Lecturas Complementarias

Lectura N° 1 Capítulo 7 Decisiones. En este capítulo del libro del Dr. Omar Iván Trejos, encontrará la propuesta del autor para abordar las estructuras de Decisión necesarias para la resolución de problemas.

Lectura N° 2 Capítulo 8 Ciclos. En este capítulo del libro del Dr. Omar Iván Trejos, encontrará la propuesta del autor para abordar las estructuras Repetitivas necesarias para la resolución de problemas.

Unidad 3

Lecturas requeridas

Lectura N° 1 Betancourt, C. (2011). Módulo Desarrollo de Algoritmos Unidad 3. Universidad de Manizales. Manizales: CEDUM.

Lectura N° 2 Capítulo 2 Tipos de Datos y Operadores. En este capítulo del libro del Dr. Cesar Becerra, encontrará un recorrido por los datos y tipos de datos que maneja el lenguaje de programación Java.

Lectura N° 3 Capítulo 3 Tipos de Programas que se escriben con Java. En este capítulo del libro del Dr. Cesar Becerra, se realiza una explicación de cómo codificar programas básicos en el lenguaje de programación java.

Lecturas Complementarias

Lectura N° 1 Capítulo 1 Historia y Evolución de Java. En este capítulo del libro Java 2 V 5.0, se hace un recuento de los orígenes y la evolución de Java como herramienta de programación Orientada a Objetos.

Lectura N° 2 Capítulo 4 Control de Ejecuciones. En el capítulo 4 del libro Piensa en Java, encontrará una explicación de la manera como se ejecutan los programas desarrollados en lenguaje Java.

Bibliografía General

- BECERRA, C. (2010). Enseñanza de la programación con Java, 2ª Edición. Bogotá: Por Computador.
- SIERRA, M. PROGRAMADOR CERTIFICADO JAVA 2, 3ª Edición. España: Rama.
- JOYANES, L. (2008). *Fundamentos de la programación. Algoritmos y Estructura de Datos*, 4ª Edición. Madrid: McGraw-Hill.
- BRUCE, E. (2007). *Piensa en Java* . 1ª Edición. Madrid: Pretice-Hall.
- JOYANES, L.; RODRIGUEZ, L; FERNANDEZ, M. (2003). *Fundamentos de programación Libro de problemas*. 2ª Edición. Madrid: McGraw-Hill.
- CEBALLOS, F. (2003). *Curso de Programación Java 2*, 2ª Edición. Madrid: Rama.
- SCHILT, H. (2005). Java 2 V 5.0. 1ª Edición. Madrid: Anaya.

Bibliografía en Internet

Recurso	Descripción
http://www.oracle.com/us/technologies/java/index.html	Documentación oficial de java
http://www.oracle.com/technetwork/java/javase/downloads/index.html	Página oficial para descargar el software de javaSE
http://netbeans.org/	Página oficial para descargar el entorno de desarrollo NetBeans
http://www.jcreator.org/download.htm	Página oficial para descargar el entorno de desarrollo JCreator
http://dfd.softonic.com/	Enlace para descargar D.F.D. Herramienta para hacer diagramas de flujo de datos
http://www.carlospes.com/curso_de_algoritmos/	Curso de algoritmos
http://es.wikibooks.org/wiki/Fundamentos_de_programaci%C3%B3n/Algoritmos_y_programas	Enlace a una página con información referente a los Fundamentos de Programación
http://es.wikibooks.org/wiki/Fundamentos_de_programaci%C3%B3n/T%C3%A9nicas_b%C3%A1sicas_de_programaci%C3%B3n	Enlace a una página con información referente a técnicas de programación

Recurso	Descripción
http://es.wikibooks.org/wiki/Programaci%C3%B3n en Java	Enlace a una página con información referente al lenguaje de programación Java a nivel básico
http://www.programacion.com/articulo/tutorjava_nivel_basico_97	Enlace a una página con información referente al lenguaje de programación Java a nivel básico
http://www.proactiva-calidad.com/java/	Enlace a una página con información referente al lenguaje de programación Java a nivel intermedio
https://docs.google.com/document/pub?id=1NojO_QHVke5lgeOpdxF5B6KIFdnNOp8vHo5fR3J3H0I#h.pcsk0yrsd51n	Videotutoriales del lenguaje de programación Java
http://www.youtube.com/watch?v=JbcB3AUwVBY	Videotutoriales (1-42) del lenguaje de programación Java

Término	Significado	Fuente
Abstracción	Acción y efecto de abstraer o abstraerse	Real Academia Española. (2011) Diccionario de la Lengua Española (22ª Edición),[versión electrónica] disponible en <u>www.rae.es</u>.
Algoritmo	Conjunto ordenado y finito de operaciones que permite hallar la solución de un problema.	Real Academia Española. (2011) Diccionario de la Lengua Española (22ª Edición),[versión electrónica] disponible en <u>www.rae.es</u>.
Constante	Es una incognita que se utiliza en los lenguajes de programación que no puede cambiar su contenido en el transcurso del programa.	http://glosarium.com/term/1507,14,xhtml
Dato	Antecedente necesario para llegar al conocimiento exacto de algo o para deducir las consecuencias legítimas de un hecho.	Real Academia Española. (2011) Diccionario de la Lengua Española (22ª Edición),[versión electrónica] disponible en <u>www.rae.es</u>.
Decisión	Determinación, resolución que se toma o se da en una cosa dudosa	Real Academia Española. (2011) Diccionario de la Lengua Española (22ª Edición),[versión electrónica] disponible en <u>www.rae.es</u>.
Diagrama de Flujo de Datos	Un diagrama de flujo de datos (DFD por sus siglas en español e inglés) es una representación gráfica para la maceta del "flujo" de datos a través de un sistema de información. Se puede utilizar para la visualización de procesamiento de datos (diseño estructurado)	http://es.wikipedia.org/wiki/
Información	Aplicación sistemática de uno o varios programas sobre un conjunto de datos para utilizar la información que contienen.	Real Academia Española. (2011) Diccionario de la Lengua Española (22ª Edición),[versión electrónica] disponible en <u>www.rae.es</u>.

Término	Significado	Fuente
Instrucción	Conjunto de reglas o advertencias para algún fin	Real Academia Española. (2011) Diccionario de la Lengua Española (22ª Edición),[versión electrónica] disponible en www.rae.es.
Lenguaje de Programación	Un lenguaje de programación es un idioma artificial diseñado para expresar computaciones que pueden ser llevadas a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana	http://es.wikipedia.org/wiki/
Operador	es un símbolo matemático que indica que debe ser llevada a cabo una operación especificada	Domingo Agustín Vázquez. «Diccionario de ciencias
Proceso	Conjunto de las fases sucesivas de un fenómeno natural o de una operación	Real Academia Española. (2011) Diccionario de la Lengua Española (22ª Edición),[versión electrónica] disponible en www.rae.es.
Prueba	Operación que se ejecuta para comprobar que otra ya hecha es correcta	Real Academia Española. (2011) Diccionario de la Lengua Española (22ª Edición),[versión electrónica] disponible en www.rae.es.
Repetición	Dicho de un aparato o de un mecanismo: Que, una vez puesto en marcha, repite su acción automáticamente.	Real Academia Española. (2011) Diccionario de la Lengua Española (22ª Edición),[versión electrónica] disponible en www.rae.es.
Variable	Estructura de programación que contiene datos. Puede contener números o caracteres alfanuméricos y el programador le asigna un nombre único. Mantiene los datos hasta que un nuevo valor se le asigna o hasta que el programa termine.	http://glosarium.com/term/1507,14,xhtml

1 TECNICA PARA RESOLVER PROBLEMAS CON AYUDA DEL COMPUTADOR.

Competencias y tabla de saberes

Unidad de aprendizaje 1		
Competencias		
<p>Identificar los problemas que pueden ser resueltos con el apoyo del computador</p> <p>Representar en un diagrama la entrada y salida de datos del problema que desea resolver.</p> <p>Diseñar algoritmos funcionales empleando estructuras condicionales o de repetición según el tipo de problema a resolver.</p> <p>Realizar diagramas de flujo para representar la solución de problemas utilizando la notación establecida.</p> <p>Escribir el pseudocódigo que permita representar algoritmos propuestos para la solución del problema</p> <p>Conceptualizar sobre la terminología básica relacionada con la solución de problemas en forma algorítmica y sus respectivas representaciones</p>		
Tabla de saberes		
Saber hacer	Saber saber	Saber ser
<p>Representa entradas, procesos y salidas como resultado del análisis y descomposición del enunciado de un problema para proponer alternativas de solución.</p> <p>Diseñar una interfaz hombre máquina acorde con la solución planteada</p> <p>Convertir expresiones aritméticas en expresiones computacionales</p> <p>Clasificar los elementos de la solución del problema en variables y/o constantes</p> <p>Representar soluciones algorítmicas en instrucciones computacionales</p> <p>Diagramación y testing de solución algorítmica mediante diagrama de flujo</p> <p>Convertir diagramas de flujo a escritura de pseudocodigos</p>	<p>Diagramas de Entrada de datos-Proceso-Salida de Información</p> <p>Algoritmo</p> <p>Técnicas para la solución de problemas</p> <p>Conceptos básicos de interfaces hombre-máquina</p> <p>Lógica matemática</p> <p>Operadores aritméticos-lógicos y relacionales</p> <p>Expresiones aritméticas</p> <p>Expresiones computacionales</p> <p>Conceptos de variables y constantes</p> <p>Datos y tipos de datos</p> <p>Instrucciones y tipos de instrucciones computacionales y su respectiva sintaxis</p> <p>Diagramas de Flujo de Datos</p> <p>Prueba de Escritorio (testing)</p> <p>Pseudocodigo</p>	<p>Activo en la participar en trabajos colaborativos</p> <p>Interesado en la actualización permanente a través del acceso a diferentes fuentes de información.</p> <p>Seguro en la toma de decisiones frente a la resolución de problemas y ejecución de actividades apoyado en sus conocimientos y habilidades.</p> <p>Responsable en el manejo de la información</p> <p>Propositivo en alternativas de solución a los problemas a partir del análisis de causas, generando valor a las actividades.</p> <p>Respetuoso de las opiniones de los demás.</p>

Abierto al cambio tecnológico.

Interesado en la actualización permanente

Creativo en la presentación de estrategias de solución a los problemas a partir del análisis de causas, generando valor a las actividades.

Interesado en las actividades relacionadas con la investigación

1. DESARROLLO DE ALGORITMOS

RESUMEN

La algoritmia entendida como la técnica base sobre la cual se fundamentan la solución de problemas de manejo de información se apoya en pensamiento lógico necesario para la identificación de la secuencia que llevará a la resolución de un problema de manejo de información. En esta unidad usted como estudiante hará un recorrido sobre una técnica que le facilite la abstracción de los requerimientos de un problema hasta llegar a los pasos necesarios para la resolución de problemas de manejo de información; el uso y aplicación secuencial de cada uno de dichos pasos, le permitirá trascender en las etapas PRECOMPUTACIONAL, COMPUTACIONAL Y POSTCOMPUTACIONAL, necesarias para la resolución de problemas.

1.1 ANÁLISIS Y CLASIFICACIÓN DEL ENUNCIADO DEL PROBLEMA EN SUS ELEMENTOS:

Para la solución problemas de manejo de información aplicando algoritmos, debemos desarrollar la habilidad de leer, analizar el enunciado y clasificar sus partes en los elementos claves que nos irán mostrando la ruta hacia la solución. Estos elementos son:

- **Capturas de datos:**
Identifica cuales son los elementos de entrada o datos que el usuario debe digitar o ingresar al sistema y que servirán como insumos para realizar los procesos aritméticos que permite obtener los resultados de la información solicitada.
- **Operaciones Aritméticas:**
Identifica cuales son las operaciones aritméticas que le aplicaremos a los datos capturados y que darán como resultado la información que están solicitando
- **Preguntas:**
Identifica explícitamente cuales son las respuestas de información que están solicitando.
- **Observaciones:**
Identifica características adicionales en el enunciado y que en un momento determinado apoyaran labúsqueda de la solución.

A continuación se desarrolla un ejemplo en el que se aplica los elementos anteriormente vistos:

Ejemplo:

Enunciado del problema:

Dados dos números enteros positivos calcular su suma

Análisis del enunciado del problema

Capturas de datos:

Para calcular la suma de **dos números**, inicialmente se deben de capturar esos dos números, generalmente esta operación se hace por teclado

Operaciones Aritméticas:

Para **calcular** la suma de dos números es necesario realizar la operación aritmética de suma

Preguntas:

En el enunciado se está preguntado:

Cuál es la **suma** de los dos números?

Observaciones:

Nos informan que **nos suministrarán dos números**, que serán los capturados por teclado

Nos informan que **los números deben de ser enteros**

Nos informan que **los números deben de ser positivos**

Enunciado final:

Dados dos números enteros positivos calcular su suma

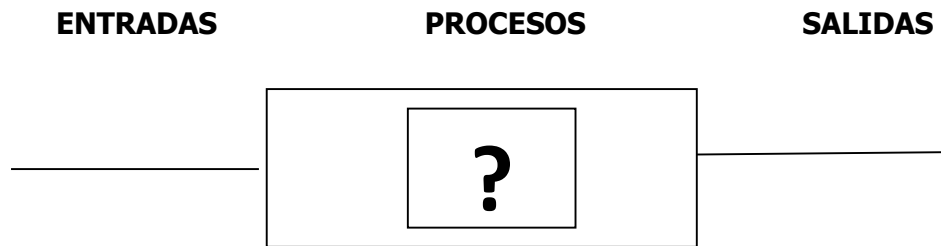
Luego de analizar e identificar los elementos del enunciado, se procede a la elaboración del diagrama de datos-proceso-salida, el cual se describe a continuación.

1.2 DIAGRAMAS DE ENTRADA DE DATOS-PROCESO-SALIDA DE INFORMACIÓN:

Haciendo uso de los elementos anteriormente identificados en el enunciado, se clasifican y grafican en los tres componentes ENTRADA – PROCESOS – SALIDAS, que son generales para cualquier problema de manejo de información, los cuales se identifican en la Gráfica 1, Entrada-Proceso-Salida y se explican a continuación.

- **Entradas:** Son los datos de entrada del problema. Qué me están dando para resolver el problema, se presentan con una línea al lado izquierdo para cada información y se bautizan con un nombre representativo de lo que se refiere.
- **Salidas:** Son las preguntas a resolver. Qué resultados me están pidiendo. se presentan con una línea al lado derecho y se bautizan con la información que nos están solicitando.

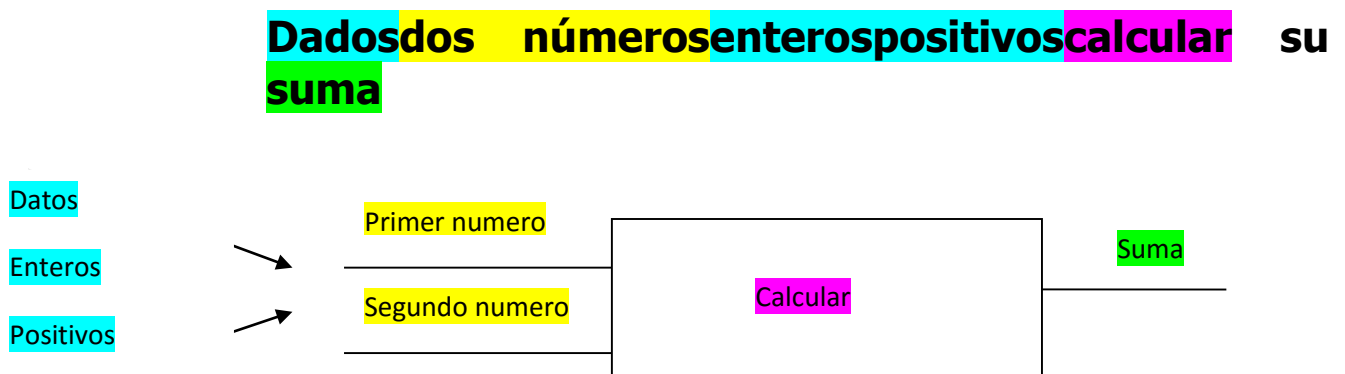
- **Procesos:** Son las operaciones generalmente aritméticas que manipulan las entradas para convertirlas en salidas. Se representan con un rectángulo y en el interior una signo de interrogación, en este paso de la técnica no es relevante en este momento analizarlos, ya que en pasos posteriores lo haremos.



Grafica 1 Entrada-Proceso-Salida

A continuación se desarrolla un ejemplo en el que se aplica los elementos anteriormente vistos:

Ejemplo:



Gráfica 2,

Grafica 2 Entrada-Proceso-Salida para el Ejemplo

Recomendaciones:

No tendría sentido resolver problemas en los cuales no existieran como mínimo entradas, procesos y salidas.

Luego de elaborado la gráfica Entrada-Proceso-Salida, se procede a la elaboración del Análisis de Procesos Aritméticos, el cual se describe a continuación.

1.3 ANÁLISIS DE PROCESOS ARITMÉTICOS:

Los procesos se hacen mediante cálculos u operaciones generalmente aritméticos, se refieren a la forma de manipular las entradas para convertirlas en salidas.

Es abstraer las operaciones necesarias para especificar la parte de los PROCESOS del diagrama E – P – S

A continuación se desarrolla un ejemplo en el que se aplica los elementos anteriormente vistos:

Ejemplo:

Para calcular la suma de dos números es necesario la operación aritmética de suma.

Suma = Primer numero + Segundo numero

Para el desarrollo de este ejemplo existen dos instrucciones:

Instrucciones de asignación Suma = "Indica que almacenamos el resultado de la operación en Suma"

Instrucción aritmética Primer número + Segundo numero "Indica que se realiza la suma de los dos numeros"

Cada vez que realicemos una operación aritmética le debemos asignar a una variable. En un ejercicio puede existir una o muchas operaciones aritméticas.

Luego de analizados los Procesos Aritméticos, se elabora el Diseño de Interfaz Hombre Máquina, el cual se describe a continuación.

• DISEÑO INTERFAZ HOMBRE MÁQUINA:

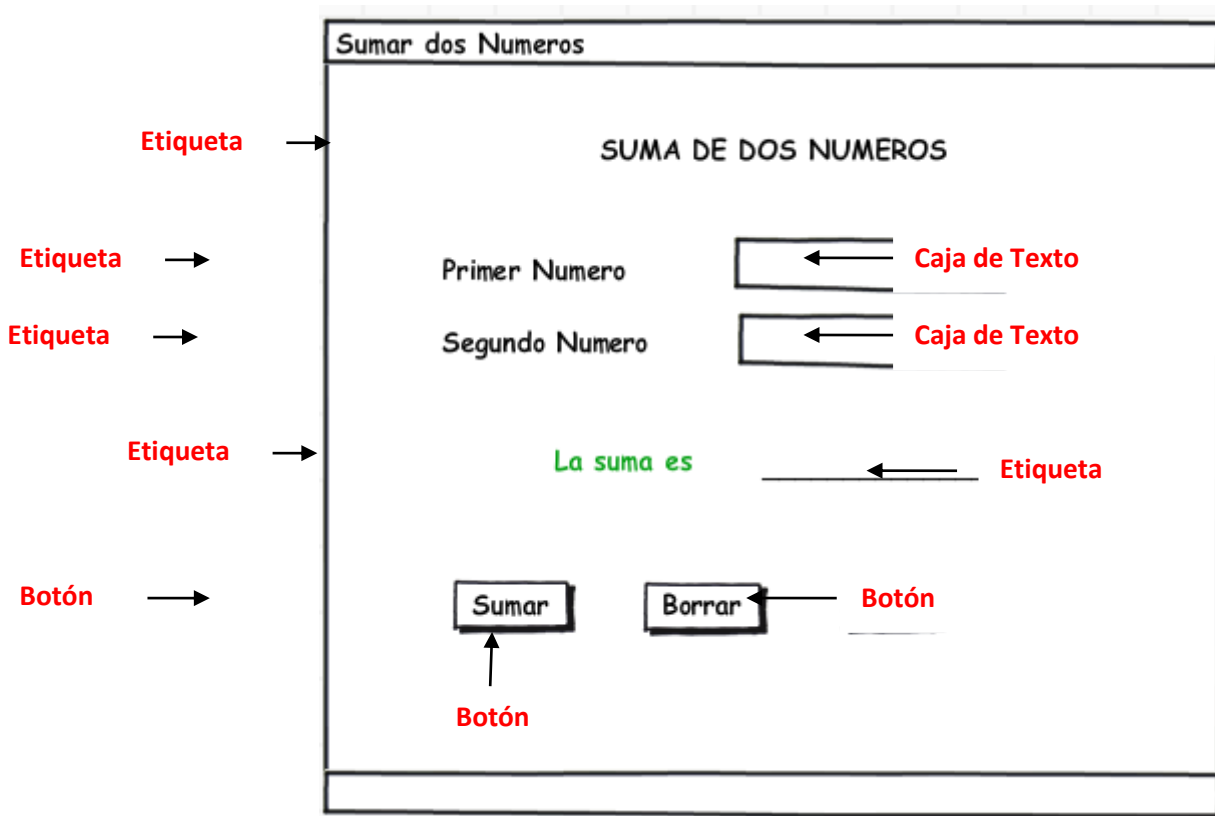
Si bien estamos buscando resolver problemas de manejo de información, también es cierto que nuestra pretensión es que nos apoyemos en el computador como herramienta tecnológica en la cual se va a capturar, procesar e imprimir los resultados, mediante la utilización de un lenguaje de programación. Con el diseño de la interfaz hombre máquina buscamos representar la forma en que va a interactuar el usuario con el computador en el cual se va a ejecutar la solución.

Los elementos básicos que encontramos en las interfaces hombre maquina son:

Elemento
Etiquetas
Cajas de Texto
Botones
ComboBox
Check Box
Radio Button
Areas
TextArea

Tabla 1 Elementos de una Interfaz

A continuación se desarrolla un ejemplo en el que se aplica los elementos anteriormente vistos:



Grafica 3 Diseño Interfaz Hombre Máquina para el Ejemplo

Luego de elaborada la Interface Hombre Máquina, se procede a la elaboración del Algoritmo, el cual se describe a continuación.

• **ALGORITMOS:**

Con el algoritmo se busca formalizar la solución del problema mediante una secuencia lógica y ordenada de pasos que van desde la captura de los datos pasando por los procesos hasta llegar a la solución y terminando en la impresión de la información como resultado solicitado.

Características de los Algoritmos

- Un único Inicio y un único fin
- Diseño Top-Down (Arriba - Abajo)
- Preciso
- Definido
- Finito

A continuación se desarrolla un ejemplo en el que se aplica los elementos anteriormente vistos:

Ejemplo:

0. Inicio
1. Leer numero Uno
2. Leer numero Dos
3. Sumar numero Uno más numero Dos
4. Escribir la respuesta
5. Fin

Luego de elaborado el Algoritmo, se procede a la elaboración de la Tabla de Datos, la cual se describe a continuación.

• **TABLA DE DATOS:**

La tabla de datos busca especificar y clasificar los elementos utilizados en la solución del problema desde la perspectiva de cómo los computadores almacenan los datos en la memoria.

A cada dato, cada información que utilicemos en la solución del problema se le debe de brindar la oportunidad de ser manejada por la memoria del computador, y cada una se debe de definir.

Identificador:

Es el nombre con que se va a bautizar el espacio de memoria donde vamos a almacenar un dato, podemos almacenar variables o constantes.

Con los identificadores también llamamos funciones, métodos, procedimientos, clases entre otros elementos.

Reglas de los Identificadores:

Cada lenguaje de programación define sus propias reglas para definir identificadores, entre los más comunes tenemos:

- El primer carácter debe ser una letra
- Puede contener letras, números y algunos símbolos especiales (solo después del primer carácter)
- En algunos lenguajes hace diferencia entre mayúsculas y minúsculas
- No permite espacios el blanco
- No permite iniciar con números o símbolos especiales
- No se permite utilizar tildes

Recomendación:

- Cuando el identificador es una palabra debe ser escrita completamente en minúscula.
- Cuando el identificador es de más de una palabra debe ser escrita unida y la primer palabra en minúscula y la segunda con Mayúscula inicial.
- No se recomienda utilizar artículos (el, la, de,...)

Tipos de datos:

Variables:

Es aquel espacio en memoria que definimos y que a lo largo de la ejecución de la solución puede cambiar su valor almacenado de acuerdo a los procesos de asignación que sufra.

Constantes:

Es aquel espacio en memoria que definimos y que a lo largo de la ejecución de la solución no cambia, es decir luego de inicializarse permanecerá con ese valor hasta el final de la ejecución del programa.

Clasificación de los datos:

Para definir un espacio en memoria debemos declarar de qué tipo de dato vamos a almacenar en dicho espacio, esto entre otras características es definir el tamaño en memoria que ocupará.

Los tipos de dato se clasifican en:

Numéricos:

Los tipos de numéricos son aquellos con los cuales se van a almacenar o van a intervenir en operaciones aritméticas o de comparación.

Entero: Es el tipo de dato en el que se almacena en una variable o constante un valor entero, que no tiene parte decimal.

Ejemplo:

Entero numeroDiasTrabajados

Entero numeroTablaMultiplicar

Entero contadorGiros

Real: Es el tipo de dato en el cual se almacena en una variable o constante un valor que tiene parte decimal.

Ejemplo:

Real resultadoDivision

Real PesosPorDolar

Real notaDefinitiva

Texto

Los tipos de datos texto son aquellos en los cuales vamos a almacenar letras o palabras o representación de número pero que no intervienen en operaciones aritméticas.

Carácter

Es el tipo de dato en el cual se almacena en una variable o constante un valor correspondiente a una letra.

Ejemplo:

Carácter genero

Carácter estadoCivil

Carácter jornadaTrabajo

Es el tipo de dato en el cual se almacena en una variable o constante un valor correspondiente a una secuencia de caracteres o una palabra.

Ejemplo:

Cadena nombreEmpleado

Cadena ciudadNacimiento

Cadena nombreEmpresa

Logicos

Boolean

Es el tipo de dato en el cual vamos a almacenar en una variable o constante un valor correspondiente a un dato que solo acepta uno de dos valores posibles generalmente falso o verdadero.

Ejemplo:

Boolean respuestaExamen

Boolean borrado

Boolean aplicaDescuento

Instrucciones sobre las Variables y/o Constantes:

Las instrucciones que se aplican sobre los datos son:

- **Declaración:** Consiste en separar un espacio de memoria de un tipo específico de dato y de tamaño de acuerdo a la necesidad.

Ejemplos:

Cada uno de los ejemplos del aparte **CLASIFICACION DE LOS TIPOS DE DATOS** aplican como declaración.

- **Inicialización:** Consiste en dar el valor inicial que va a tener la variable o la constante en la memoria. En este proceso se utiliza una instrucción de asignación y puede realizarse en el mismo momento de la declaración o posterior a ella.

Generalmente las variables numéricas se inicializan con el valor de cero y las de texto con espacios en blanco y los booleanos en falso.

Ejemplo:

```
numeroEstudiantesGrupo = 0
```

```
notaPrimerParcial = 0.0
```

```
genero = ""
```

```
nombreMadre = ""
```

```
padreVive = falso
```

- **Asignación:** Es modificar el valor de inicialización de la variable o constante de acuerdo a la necesidad del usuario o a un cálculo matemático.

```
numeroEstudiantesGrupo = 40
```

```
notaPrimerParcial = 3.0
```

```
genero = "M"
```

```
nombreMadre = "Ana María"
```

```
padreVive = Verdadero
```

Esquema General para la declaración de variables:

<tipoVariable> <nombreVariable> = <valorInicial>;

- **Especificación de la Tabla de datos:**

Mediante la utilización de esta tabla hacemos una relación y una clasificación de los datos utilizados durante el desarrollo de toda la propuesta de solución, e incluye los siguientes elementos:

Identificador:

Palabra con que vamos a identificar el dato en la memoria del computador

Tipo:

Se clasifica el dato de acuerdo a la **TABLA DE DATOS** vista anteriormente.

Valor Inicial:

Se informa del valor con que se va a inicializar el identificador al momento de declararlo.

Ámbito:

Se especifica en que parte de la solución aparece o es utilizado el identificador, esas partes son Entrada (E), Proceso (P) y Salida (S).

Observación:

Se especifican elementos que pueden ser de ayuda para el entendimiento o solución del problema, generalmente se colocan las formulas que calculan el valor de los datos.

Documentación:

Mediante un texto se describe la utilización del identificador en la solución propuesta, debe ser muy concisa.

La tabla de datos se representa de la siguiente manera:

Identificador	Tipo	Tipo Dato	Valor Inicial	Ámbito	Observación	Documentación

Tabla 2 Tabla de Datos

A continuación se desarrolla un ejemplo en el que se aplica los elementos anteriormente vistos:

Ejemplo:

Identificador	Tipo	Tipo Dato	Valor Inicial	Ámbito			Observación	Documentación
				E	P	S		
numeroUno	Variable	Entero	0	E				Variable donde se va a almacenar un dato ingresado por

								teclado por el usuario
numeroDos	Variable	Entero	0	E				Variable donde se va a almacenar un dato ingresado por teclado por el usuario
Suma	Variable	Entero	0		P	S	Suma= numeroUno + numeroDos	Variable donde se va a almacenar la respuesta producto de una operación aritmética

Tabla 3 Tabla de Datos para el Ejemplo

Luego de elaborada la Tabla de Datos, se procede a la elaboración de la Tabla De Conversión De Expresiones Aritméticas a Computacionales, como se describe a continuación.

• TABLA DE CONVERSIÓN DE EXPRESIONES ARITMÉTICAS A COMPUTACIONALES:

La tabla de expresiones tiene dos objetivos fundamentales, como son:

- Estructurar las expresiones aritméticas
- Traducir las expresiones aritméticas a computacionales

EXPRESION ALGEBRAICA:

Como se observa en la Grafica 4, las expresiones aritméticas son una forma de representar las formulas que realizan los cálculos buscando con ello obtener un resultado en ellas intervienen elementos como variables, constantes, operadores aritméticos, datos de tipo numérico.

La expresión aritmética está compuesta por tres elementos que son: la variable dependiente, el operador de igual y la combinación de números, variables, operadores aritméticos entre otros.

Lo importante en esta etapa de la solución es conocer como se realizar los cálculos para obtener los resultados y no como los realizará el computador.

A continuación se desarrolla un ejemplo en el que se aplica los elementos anteriormente vistos:

Ejemplo:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$y = \pm\sqrt{1+a}$$

Grafica 4 Expresión algebraica

OPERADORES ARITMETICOS Y SU JERARQUIA:


Operador	Operación
+	Suma
-	Resta
*	Multipliación
/	División
^	Exponenciación
	Radicación

Tabla 4 Operadores Aritméticos

JERARQUIA DE LOS OPERADORES ARITMÉTICOS:

La jerarquía de los operadores aritméticos consiste en priorizar cuales operadores relacionados en la tabla 4 se ejecutan primero y en qué orden, así:


Operador	Categoría	Observación
()	1	Primero se resuelven los más internos
 , ^	2	
*, /	3	
+, -	4	

Tabla 5 Jerarquía de los Operadores Aritméticos

Observación:

Para los operadores tiene el mismo nivel, se resuelven de izquierda a derecha en el caso de + y del -, se resuelve primero el signo más luego el menos.

EXPRESION COMPUTACIONAL:

Es una forma de escribir las expresiones algebraicas, mediante una serie de instrucciones que sean entendibles por los lenguajes de programación.

A continuación se desarrolla un ejemplo en el que se aplica los elementos anteriormente vistos:

Ejemplo:

La tabla 6 tiene la siguiente presentación:

Expresiones Aritméticas	Expresiones Computacionales
-------------------------	-----------------------------

$\frac{\text{Hipotenusa}}{\sqrt{\text{catetoUno}^2 + \text{catetoDos}^2}} =$	$\text{hipotenusa} = \text{SQR}(\text{catetoUno}^2 + \text{catetoDos}^2)$
$\text{Perimetro} = \text{catetoUno} + \text{catetoDos} + \text{hipotenusa}$	$\text{perímetro} = \text{catetoUno} + \text{catetoDos} + \text{hipotenusa}$
$\text{Area} = \frac{\text{catetoUno} * \text{catetoDos}}{2}$	$\text{area} = \text{catetoUno} * \text{catetoDos} / 2$

Tabla 6 Conversión de Expresiones Aritméticas a Computacionales

Luego de elaborar la tabla de Conversión de Expresiones Aritméticas a Expresiones Computacionales, se procede a la elaboración del Diagrama de Flujo de Datos, el cual se describe a continuación.

• DIAGRAMA DE FLUJO DE DATOS:

El Diagrama de Flujo de Datos es una representación gráfica del algoritmo utilizando instrucciones que se acercan a la etapa computacional.

Cada instrucción del algoritmo tiene su representación gráfica en el Diagrama de Flujo de Datos, los cuales se describen en la Tabla 7, Símbolos Diagramas de Flujo.


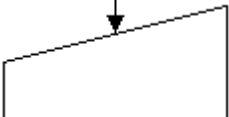

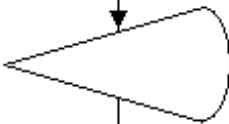
NOMBRE DEL SIMBOLO	SIMBOLO
Inicio – Fin	
Captura por Teclado	
Proceso	
Impresión o Salida de Datos	

Tabla 7 Símbolos Diagramas de Flujo

Recomendaciones:

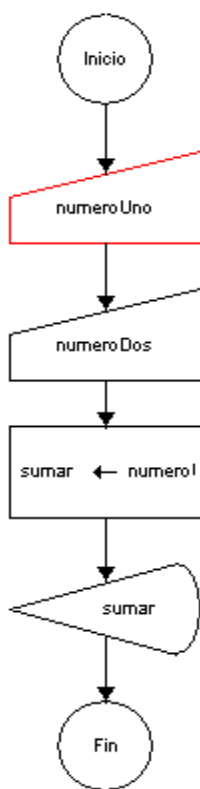
El Diagrama de Flujo de Datos de cada ejercicio, debe de guardar total coherencia con el algoritmo, así como con la tabla de datos y de expresiones. Adicionalmente debe presentarse en el mismo orden que el algoritmo.

Software a utilizar:

Existen aplicaciones de software para representar Diagramas de Flujo de Datos, entre ellos tenemos:

A continuación se desarrolla un ejemplo en el que se aplica los elementos anteriormente vistos:

Ejemplo del D.F.D.:



Grafica 5 Diagrama de Flujo de Datos para el Ejemplo

Una vez elaborado el Diagrama de Flujo de Datos, se procede a la elaboración de la Prueba de Escritorio, utilizando la grafica 8 plantilla prueba de escritorio, la cual se describe a continuación.

• PRUEBA DE ESCRITORIO:

Con la prueba de escritorio se hace un seguimiento al comportamiento de las variables y los valores que toman a lo largo de la ejecución de la solución y a medida que suceden cada una de las instrucciones.

Luego de tener en la plantilla de prueba de escritorio todas las variables identificadas se procederá a realizar una verificación en papel del funcionamiento del diagrama de flujo, resultados que se escriben en la misma plantilla, adicionalmente se realiza una solución manual al problema y se comparan los resultados con los de la plantilla, los cuales deben ser iguales.

Es recomendable realizar la prueba de escritorio en varias ocasiones con los diferentes valores posibles que pueda llegar a digitar el usuario, es decir tomar una muestra de valores y hacer la validación de la solución.

En caso de que las respuestas arrojadas por el D.F.D. en las pruebas de escritorio no sean iguales, nos tendremos que remitir a los pasos anteriores de algoritmos, tabla de datos, tabla de operaciones para buscar y corregir el error.

	Variables y/o Constantes				Salidas		Estado
	Variable <i>a</i>	Variable <i>b</i>	Variable <i>n</i>	Calculo Manual	Salida Algoritmo	
Inicialización	0	0	0			
Paso 1							
Paso 2							
...							
Paso <i>n</i>							

Tabla 8 Plantilla Prueba de Escritorio

A continuación se desarrolla un ejemplo en el que se aplica los elementos anteriormente vistos:

Ejemplo:

	Variables y/o Constantes			Salidas		Estado
	numeroUno	numeroDos	Suma	Calculo Manual	Salida Algoritmo	
Inicialización	0	0	0			
Paso 1	5					
Paso 2	10					
Paso 3			15			
Paso 4				15	15	OK

Tabla 9 Plantilla Prueba de Escritorio para el Ejemplo

Luego de elaborada la Prueba de Escritorio, se procede a la elaboración del Pseudocódigo, el cual se describe a continuación.

• PSEUDOCODIGO:

En el pseudocódigo es la etapa posterior a la verificación del algoritmo y es a su vez la etapa anterior a la codificación en un lenguaje de programación, es una etapa intermedia en la cual se transcribe el algoritmo a una escritura con la estructura y reglas propia de un Lenguaje de Programación.

Recomendaciones Adicionales:

- Generalmente en los pseudocódigos las líneas de texto que no hacen parte de la codificación de documentan de la siguiente manera:

// Documentación de una línea

/* Apertura de documentación de más de una línea

*/ Cierre de documentación de más de una línea

Partes del Pseudocódigo:

Información del Programa:

Nombre del Archivo:

Se escribe el nombre con que se va a guardar el archivo y su extensión correspondiente, si hay varios archivos involucrados en la solución entonces se relaciona todos.

Descripción:

Mediante un texto se describe puntualmente la funcionalidad del programa que estamos desarrollando.

Autor:

Nombre de la persona que desarrollo el software

Versión:

Númericamente se describe el número de la versión del programa. Es muy posible que un programa luego de desarrollado sufra mejoras o adaptaciones, para cada una de ella se debe de realizar en un nueva versión de forma secuencial.

Fecha:

Fecha de la terminación del programa

Modulo Principal

Área de Declaración de Variables:

Entradas

Procesos

Salidas

Fin_Modulo_Principal

A continuación se desarrolla un ejemplo en el que se aplica los elementos anteriormente vistos:

EJEMPLO:

/*

Información del Programa:

Nombre de Archivos:

Diagrama de Flujo de Datos: sumaNumeros.dfd

Intefaz: formularioPrincipal.png

Pseudocódigo: sumaNumeros.txt

Proyecto Java: proyectoSumaNumeros

Ubicación: D:\Proyectos\java\sumaNumeros

Descripción:

En este proyecto se capturan por teclado dos números enteros y se realizan tres funcionalidades (Sumar, Borrar y Salir)

Autor:

Carlos Betancourt Correa

Version:

1.0

Fecha:

Junio 15 de 2011

*/

Modulo Principal

// Área de Declaración e inicialización de Variables:

Entero numeroUno = 0 ;

Entero numeroDos = 0 ;

Entero suma = 0 ;

// Entradas

Leer (numeroUno) ;

Leer (numeroDos) ;

// Procesos

Suma = numeroUno + numeroDos ;

// Salidas

Escribir (suma);

Fin_Modulo_Principal

Con el pseudocódigo anteriormente visto culminamos con los elementos necesarios para solucionar un problema de manejo de información hasta la etapa precomputacional, tema correspondiente a la primera unidad.

A continuación se desarrolla un ejemplo en el que se aplica los elementos tratados en esta unidad:

EJEMPLO APLICADO:

Mediante un ejemplo al que le aplicaremos paso a paso la técnica anteriormente expuesta en la búsqueda del apropiación de la misma:

ENUNCIADO:

Capturar por teclado los 2 lados correspondientes a los catetos de un triángulo rectángulo, se pide calcular e imprimir los valores de la hipotenusa, el perímetro y el área.

ANÁLISIS Y CLASIFICACIÓN DEL ENUNCIADO DEL PROBLEMA EN SUS ELEMENTOS:

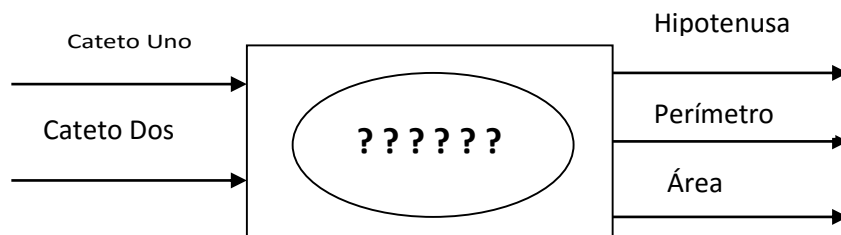
Tabla de Elementos del Enunciado:

Elemento	Valor en el Enunciado
Captura de Datos	2 lados
Preguntas	Hipotenusa Perímetro Área
Procesos	Calcular
Observaciones	Los dos lados corresponden a los catetos La figura Geométrica es un Triangulo Rectángulo.

Enunciado como resultado luego del análisis:

Capturar por teclado los 2 lados correspondientes a los catetos de un triángulo rectángulo, se pide calcular e imprimir los valores de la hipotenusa, el perímetro y el área.

DIAGRAMA ENTRADA – PROCESO – SALIDA:



ANÁLISIS DE PROCESOS ARITMÉTICOS:

Calcular:

$$\text{Hipotenusa} = \sqrt{\text{catetoUno}^2 + \text{catetoDos}^2}$$

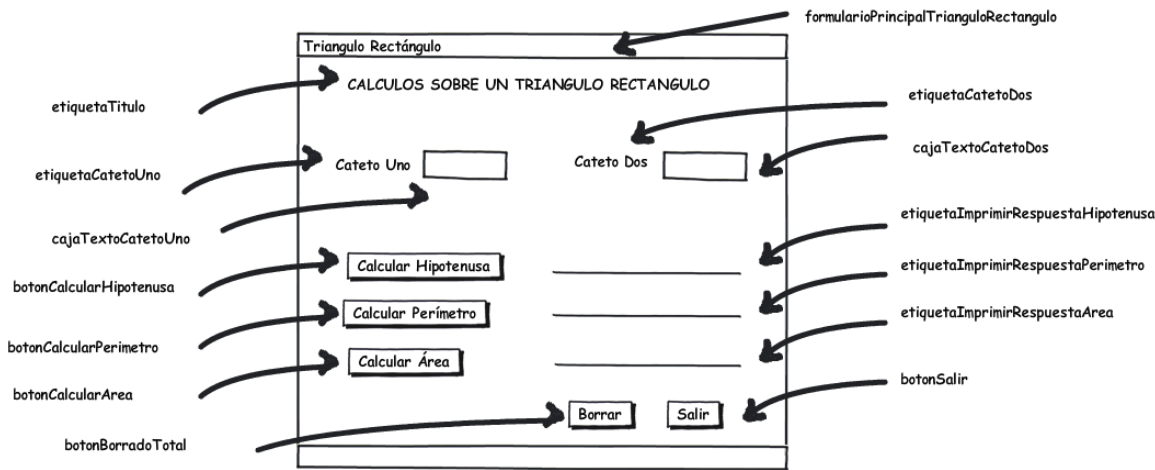
Calcular:

$$\text{Perimetro} = \text{catetoUno} + \text{catetoDos} + \text{hipotenusa}$$

Calcular:

$$\text{Area} = \frac{\text{catetoUno} * \text{catetoDos}}{2}$$

DISEÑO DE INTERFAZ HOMBRE MÁQUINA:



ALGORITMO:

Paso	Descripción
0	Inicio
1	Leer Cateto Uno
2	Leer Cateto Dos
3	Calcular Hipotenusa
4	Calcular Perimetro
5	Calcular Area
6	Imprimir Hipotenusa
7	Imprimir Perimetro
8	Imprimir Area
9	Salir

TABLA DE DATOS:

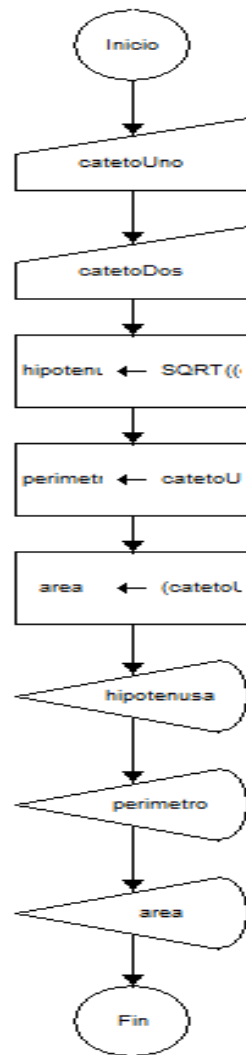
Identificador	Tipo	TipoDato	Valor Inicial	Ambito			Observaciones	Documentación
				E	P	S		
catetoUno	Variable	Entero	0	E				Variable donde se va a almacenar un

								dato ingresado por el usuario y que corresponde al valor de un cateto
catetoDos	Variable	Entero	0	E				Variable donde se va a almacenar un dato ingresado por el usuario y que corresponde al valor de un cateto
hipotenusa	Variable	Real	0.0		P	S	Es producto de un proceso (ver tabla de procesos)	Variable donde se va a almacenar la respuesta de una operación aritmética
perimetro	Variable	Real	0.0		P	S	Es producto de un proceso (ver tabla de procesos)	Variable donde se va a almacenar la respuesta de una operación aritmética
Area	Variable	Real	0.0		P	S	Es producto de un proceso (ver tabla de procesos)	Variable donde se va a almacenar la respuesta de una operación aritmética

TABLA DE CONVERSIÓN DE EXPRESIONES ARITMÉTICAS A COMPUTACIONALES:

Expresiones Aritméticas	Expresiones Aritméticas
$\text{Hipotenusa} = \sqrt{\text{catetoUno}^2 + \text{catetoDos}^2}$	$\text{Hipotenusa} = (\text{pow}(\text{catetoUno}, 2) + \text{pow}(\text{catetoDos}, 2))$
$\text{Perimetro} = \text{catetoUno} + \text{catetoDos} + \text{hipotenusa}$	$\text{Perimetro} = \text{catetoUno} + \text{catetoDos} + \text{hipotenusa}$
$\text{Area} = \frac{\text{catetoUno} * \text{catetoDos}}{2}$	$\text{Area} = \text{catetoUno} * \text{catetoDos} / 2$

DIAGRAMA DE FLUJO DE DATOS:



PRUEBA DE ESCRITORIO:

Variables y/o Constantes					
catetoUno	catetoDos	hipotenusa	Perímetro	area	Salida
0	0	0.0	0.0	0.0	
2	3	3.6	8.6	6	Hipotenusa : 3.6 Perímetro : 8.6 Área : 6

PSEUDOCODIGO:

/* SECCIÓN DE INFORMACIÓN DEL PROGRAMA

Programa: trianguloRectangulo
Nombre del Archivo: trianguloRectangulo.java
Descripción: Programa que pide el valor de dos lados de un triangulo rectángulo por teclado y calcula e imprime la hipotenusa, el perímetro y el área.
Autor: Carlos Betancourt Correa
Fecha: Junio 20 de 2011
Version: 1.0

*/

// SECCIÓN MODULO PRINCIPAL

/* SECCIÓN DE DECLARACIÓN E INICIALIZACION DE VARIABLES

Entero catetoUno=0;
 catetoDos=0;

Real hipotenusa=0.0;
 perímetro=0.0;
 area=0.0;

/* SECCIÓN DE CAPTURA DE DATOS

ESCRIBIR ("CALCULOS SOBRE UN TRIANGULO RECTANGULO : ");
ESCRIBIR ("Cateto Uno : ");
LEER (catetoUno);
ESCRIBIR ("Cateto Dos : ");
LEER (catetoDos);

/* SECCIÓN DE PROCESOS

CALCULAR hipotenusa = RAIZ2((catetoUno^2)+(catetoDos^2));

CALCULAR perímetro = hipotenusa + catetoUno + catetoDos;
CALCULAR area = (catetoUno * catetoDos / 2);

/* SECCIÓN IMPRESIÓN DE RESULTADOS

ESCRIBIR ("La hipotenusa es : "+hipotenusa);
ESCRIBIR ("El perímetro es : "+perímetro);
ESCRIBIR ("La area es : "+area);

/* FIN DEL MUDULO PRINCIPAL

Actividades de aprendizaje

Actividades Autónomas

Actividad N° 2
Nombre de la actividad: Mapa mental sobre los conceptos de ciclos y estructuras
Nombre del recurso a utilizar. FreeMind, préstamo en línea, tarea moodle
Competencias <ul style="list-style-type: none"> • Representa en un diagrama la entrada y salida de datos del problema que desea resolver. • Diseña algoritmos funcionales empleando estructuras condicionales o de repetición según el tipo de problema a resolver. • Realizar diagramas de flujo para representar la solución de problemas utilizando la notación establecida. • Escribir el pseudocódigo que permita representar algoritmos propuestos para la solución del problema
Indicaciones pasó a paso: <ol style="list-style-type: none"> 1. Después de realizar la lectura a unidad 02 del módulo y de los textos sugeridos, 2. Lectura de los Capítulos 4 y 5 del libro de Fundamentos de Programación de Luis Joyanes Aguilar Tercera Edición 3. Realice un mapa mental que represente los elementos fundantes necesarios para resolver problemas de manejo de información mediante algoritmos y el uso de estructuras de selección y repetición. 4. Para ello se sugiere emplear el software FreeMind, en el aula virtual encuentra el tutorial y el enlace para descargar. 5. Suba el archivos en el recurso tarea
Producto a entregar Plantilla diligenciada por cada uno de los ejercicios, subirlas en el recurso tarea.
Fecha límite cuarta semana
Porcentaje de valoración 5%

Actividad N° 3
Nombre de la actividad: Aplicación de la plantilla Guía para la Resolución de Problemas con Apoyo del Computador.
Nombre del recurso a utilizar.

<p>Tarea-Moodle</p> <p>Competencias</p> <ul style="list-style-type: none"> • Representa en un diagrama la entrada y salida de datos del problema que desea resolver. • Diseña algoritmos funcionales empleando estructuras condicionales o de repetición según el tipo de problema a resolver. • Realizar diagramas de flujo para representar la solución de problemas utilizando la notación establecida. • Escribir el pseudocódigo que permita representar algoritmos propuestos para la solución del problema
<p>Indicaciones pasó a paso:</p> <p>Después de realizar la lectura de la unidad 02 del módulo, diligencie la plantilla guía para la resolución de problemas para cada uno de los siguientes ejercicios propuestos:</p> <ol style="list-style-type: none"> 1. Capture por teclado un número entero e imprima por pantalla si el número es par o impar. 2. Lea por teclado el nombre de un estudiante, el nombre de una materia y la nota obtenida para cada uno de las 3 evaluaciones que se le practicaron durante el periodo académico. Se pide : <ul style="list-style-type: none"> • Calcular la nota definitiva sabiendo que se calcula por promedio • Imprimir un mensaje si aprobó o reprobó la asignatura sabiendo que para ganarla la nota definitiva debe ser igual o mayor a 3.0 3. Capture un numero entero por teclado y realice el calculo el factorial de ese número.
<p>Producto a entregar</p> <p>Plantilla diligenciada por cada uno de los ejercicios, subirlas en el recurso tarea.</p>
<p>Fecha límite quinta semana</p>
<p>Porcentaje de valoración 5%</p>

Actividades colaborativas

<p>Actividad N° 4</p>
<p>Nombre de la actividad</p> <p>Realización de ejercicios propuestos.</p>
<p>Nombre del recurso a utilizar.</p> <p>Tarea de moodle, préstamo en línea</p> <p>Competencias o indicador de competencias</p> <ul style="list-style-type: none"> • Diseña algoritmos funcionales empleando estructuras condicionales o de repetición según el tipo de problema a resolver. • Realizar diagramas de flujo para representar la solución de problemas utilizando la notación establecida. • Escribir el pseudocódigo que permita representar algoritmos propuestos para la solución del problema. • Definir los conceptos básicos en las etapas del ciclo de desarrollo de un programa.
<p>Indicaciones pasó a paso:</p> <p>Inicie con las lecturas:</p> <ul style="list-style-type: none"> • Lectura del Unidad 02 del módulo.

- Lectura: Del Libro Fundamentos de Programación del Luis Joyanes Aguilar, el capítulo 4 (Flujo s de Control Estructuras Selectivas), y 5 (Flujo s de Control. Estructuras Repetitivas)
- Realice los problemas propuestos al final de cada uno de los capítulos.
- Plantee en el foro inquietudes las preguntas que surjan en el proceso

Momento Individual:

Realice las lecturas del Libro Fundamentos de Programación de Luis Joyanes Aguilar de los capítulo 4 (Flujo s de Control Estructuras Selectivas), y 5 (Flujo s de Control. Estructuras Repetitivas). Plantee en el foro las inquietudes y las preguntas que surjan respecto a las lecturas.

Momento de Socialización:

Coserte con su grupo horarios de encuentro para que realicen los comparativos sobre las soluciones que cada uno propuso a los ejercicios y selección la que consideren con mayor eficiencia.

Construcción Colectiva:

El equipo enviará el trabajo con los ejercicios resueltos anexando una conclusión referente a las estrategias utilizadas en el desarrollo del mismo y las ventajas que éstas representan.

Producto a entregar

Documento con los ejercicios resueltos.

Fecha límite sexta semana

4 horas estimadas para moderar el foro inquietudes.

Porcentaje de valoración 10%

Actividades de Acompañamiento tutorial

Actividad N°1 Presentación de la Unidad II

Nombre del recurso a utilizar. Video conferencia

Competencias o indicador de competencias.

Interactuar (con los participantes, tutor, docente y contenidos) en el aula virtual a través de los recursos y actividades para alcanzar las competencias propuestas.

Indicaciones pasó a paso: En el chat se realiza la presentación de los contenidos y actividades de la Unidad de Aprendizaje II.

En este encuentro se presentan los aspecto tener a tener en cuenta para el uso de ciclos y estructuras.

Temas a tratar:

- Condición
- Operadores Relacionales
- Decisiones, su clasificación, funcionamiento y aplicación Estructuras repetitivas, su clasificación, funcionamiento y aplicación
- Representación gráfica y en pseudocódigo de estructuras condicionales y repetitivas
- Informe sobre las competencias alcanzadas en las diferentes actividades realizadas.

<ul style="list-style-type: none"> Los estudiantes harán participación activa con base en la preguntas que surjan durante la exposición a en el abordaje de las lecturas recomendadas para la unidad
Fecha límite cuarta semana
Porcentaje de valoración 5%

Actividad N° 5
Nombre de la actividad Retroalimentación de la Unidad II
Competencias <ul style="list-style-type: none"> Demostrar interés por la actualización permanente a través del acceso a diferentes fuentes de información. Cumplir con el cronograma de trabajo estableciendo su propia dinámica de aprendizaje
Nombre del recurso a utilizar. Video conferencia
Indicaciones pasó a paso: La video conferencia tiene como fin realizar la retroalimentación de los temas desarrollados en la Unidad II y socializar aspectos de los trabajos individuales y en equipo. En este encuentro se discute sobre los aspectos a tener en cuenta para el uso de ciclos y estructuras Las pregunta orientadoras se planean a partir de la identificación de las necesidades que surjan en el desarrollo de la unidad I
Fecha Límite sexta semana
Porcentaje de valoración 5%

Evaluación de los aprendizajes

Resultados del aprendizaje

- Utiliza diagramas para la representación de problemas y su solución Representar en un diagrama la entrada y salida de datos del problema que desea resolver.
- Elabora diagramas de flujo funcionales que evidencien la solución lógica de un problema.

Actividad	Competencias	Criterios	%
Actividades autónomas			
Actividad Nº 2 Mapa mental algoritmos	<ul style="list-style-type: none"> • Identificar las características generales de los problemas propuestos que pueden ser resueltos de forma algorítmica y con el apoyo del computador. • Conceptualizar sobre la terminología básica relacionada con la solución de problemas en forma algorítmica y sus respectivas representaciones. 	<ul style="list-style-type: none"> • Reconoce situaciones problemáticas que pueden ser sistematizados • Describe alternativas de soluciones a situaciones problemáticas • Identifica los conceptos propios de los algoritmos • Establece los pasos lógicos para desarrollar algoritmos 	5%
Actividad Nº 4 aplicación de plantilla guía	<ul style="list-style-type: none"> • Representa en un diagrama la entrada y salida de datos del problema que desea resolver. • Diseña algoritmos funcionales empleando estructuras condicionales o de repetición según el tipo de problema a resolver. • Realizar diagramas de flujo para representar la solución de problemas utilizando la notación establecida. • Escribir el pseudocódigo que permita representar algoritmos propuestos para la solución del problema 	<ul style="list-style-type: none"> • Utiliza la notación para diagramas de entrada, proceso y salida de datos. • Representa estructuras condicionales • Representa estructuras repetitivas • Aplica las estructuras básicas de programación a la solución de problemas. • Emplea la notación especificada en el diseño de algoritmos • Utiliza la notación para diagramas de flujo • Representa un algoritmos mediante el diagrama de flujo • Escribe el diagrama de flujo en el pseudocódigo correspondiente • Utiliza la estructura lógica del pseudocódigo 	5%
Actividades colaborativas			

Actividad Nº 5 Ejercicios de aplicación	<ul style="list-style-type: none"> Diseña algoritmos funcionales empleando estructuras condicionales o de repetición según el tipo de problema a resolver. Realizar diagramas de flujo para representar la solución de problemas utilizando la notación establecida. 	<ul style="list-style-type: none"> Representa estructuras repetitivas Aplica las estructuras básicas de programación a la solución de problemas. Emplea la notación especificada en el diseño de algoritmos Utiliza la notación para diagramas de flujo Representa un algoritmos mediante el diagrama de flujo 	10%
Actividades de acompañamiento tutorial			
Actividad Nº1 Presentación y contextualización del Módulo	<ul style="list-style-type: none"> Interactuar (con los participantes, tutor, docente y contenidos) en el aula virtual a través de los recursos y actividades para alcanzar las competencias propuestas. 	<ul style="list-style-type: none"> Participa en los encuentros síncronos definidos en el módulo Interactúa con los compañeros a través de medios síncronos y asíncronos Accede en forma permanente al aula virtual para consultar los contenidos Utiliza de los recursos tecnológicos necesarios para acceder a los contenidos y realizar las actividades 	5%
Actividad Nº3 Video Clase Fundamentos los algoritmos	<ul style="list-style-type: none"> Representa en un diagrama la entrada y salida de datos del problema que desea resolver. Diseña algoritmos funcionales empleando estructuras condicionales o de repetición según el tipo de problema a resolver. Realizar diagramas de flujo para representar la solución de problemas utilizando la notación establecida. Escribir el pseudocódigo que permita representar algoritmos propuestos para la solución del problema 	<ul style="list-style-type: none"> Utiliza la notación para diagramas de entrada, proceso y salida de datos. Representa estructuras condicionales Representa estructuras repetitivas Aplica las estructuras básicas de programación a la solución de problemas. Emplea la notación especificada en el diseño de algoritmos Utiliza la notación para diagramas de flujo Representa un algoritmos mediante el diagrama de flujo Escribe el diagrama de flujo en el pseudocódigo correspondiente Utiliza la estructura lógica del pseudocódigo 	5%
Actividad Nº 6 Retroalimentación Unidad I	<ul style="list-style-type: none"> Demostrar interés por la actualización permanente a través del acceso a diferentes fuentes de información. Cumplir con el cronograma de trabajo estableciendo su propia dinámica de aprendizaje 	<ul style="list-style-type: none"> Participa en los encuentros síncronos definidos en el módulo Interactúa con los compañeros a través de medios síncronos y asíncronos Accede en forma permanente al aula virtual para consultar los contenidos Utiliza de los recursos tecnológicos necesarios para acceder a los contenidos y realizar las actividades. 	5%

Unidad de aprendizaje 2 Ciclos y estructuras básicas de programación Competencias		
Diseña algoritmos funcionales empleando estructuras condicionales o de repetición según el tipo de problema a resolver. Realizar diagramas de flujo para representar la solución de problemas utilizando la notación establecida. Escribir el pseudocódigo que permita representar algoritmos propuestos para la solución del problema Definir los conceptos básicos en las etapas del ciclo de desarrollo de un programa.		
Tabla de saberes		
Saber hacer	Saber saber	Saber ser
Incorporación de las estructuras de decisión y de repetición en las soluciones planteadas. Representación de las estructuras de decisión y de repetición en instrucciones computacionales Diagramación y testing de las estructuras de decisión y de repetición en diagramas de flujo. Convertir las estructuras de decisión y de repetición de diagramas de flujo a pseudocódigo	Condición Operadores Relacionales Decisiones, su clasificación, funcionamiento y aplicación Estructuras repetitivas, su clasificación, funcionamiento y aplicación Representación gráfica y en pseudocódigo de estructuras condicionales y repetitivas	Respetuoso cuando se expresa hacia los compañeros de trabajo y hacia el tutor. Consciente y ético para realizar las actividades. Organizado para presentar las actividades. Responsable en entregar completas las actividades Cumplido en entregar las actividades Interesado en realizar actividades colaborativas. Interesado a la hora de consultar temas referentes a los temas propuestos. Creativo para realizar las actividades propuestas durante la ejecución del módulo Respetuoso con los derechos de autor al presentar trabajos escritos.

2 CICLOS Y ESTRUCTURAS

RESUMEN:

En la resolución de problemas utilizando la programación de computadores es muy frecuente que los pasos propuestos para resolver un problema contengan situaciones en las cuales se daba tomar una decisión para elegir una entre varias posibilidades al igual que ejecutar la repetición de uno o más pasos buscando con ello la eficiencia, es por ello que este capítulo se centra en guiar al estudiante en un recorrido por las estructuras tanto de decisión como de repetición, vistas como herramientas de apoyo a la solución lógica de problemas.

2.1 ESTRUCTURAS DE SELECCIÓN:

Las estructuras de selección es una poderosa instrucción que nos permite ejecutar un solo un bloque de instrucciones en uno de dos caminos posibles previa la realización de una evaluación.

2.1.1 Partes de una estructura de selección:

Comparación:

Una comparación es la instrucción que nos permite analizar mediante un operador de relación, cual es la relación existente entre dos elementos, así:

Elemento	Operador Relacional	Elemento
Numero		Número
Variable		Variable
Expresión		Expresión
Constante		Constante

Tabla 10: tabla de relación de operadores

Operadores Relacionales:

Los operadores relacionales son:

Operador	Símbolo	Observación
Mayor	>	Limite Cerrado
Menor	<	Limite Cerrado
Igual	=	
Mayor o Igual	>=	Limite Abierto
Menor o Igual	<=	Limite Abierto
Diferente	!=	Negación de la igualdad

Tabla 11 lista de operadores relacionales

Operadores Lógicos:

Los operadores lógicos son aquellos que nos permite agrupar comparaciones y nos retornan valores booleanos (falso o verdadero) o numéricos (0 o diferente de cero), se utilizan según el caso que estemos comparando.

Clasificación:

Operador	Simbolos que lo representan
Y	&& - AND
O	- OR
Negación	! – NOT

Tabla 12 operadores lógicos

Observación:

Cuando unimos dos o más comparaciones mediante un operador **Y** , ambas se deben de retornar verdadero para que la parte verdadera se ejecute, de lo contrario se ejecutará la parte falsa.

Cuando unimos dos o más comparaciones mediante un operador **OR** , con una sola que retorne verdadero se ejecutará la parte verdadera, si todas las comparaciones retornan falso entonces se ejecutará la parte falsa.

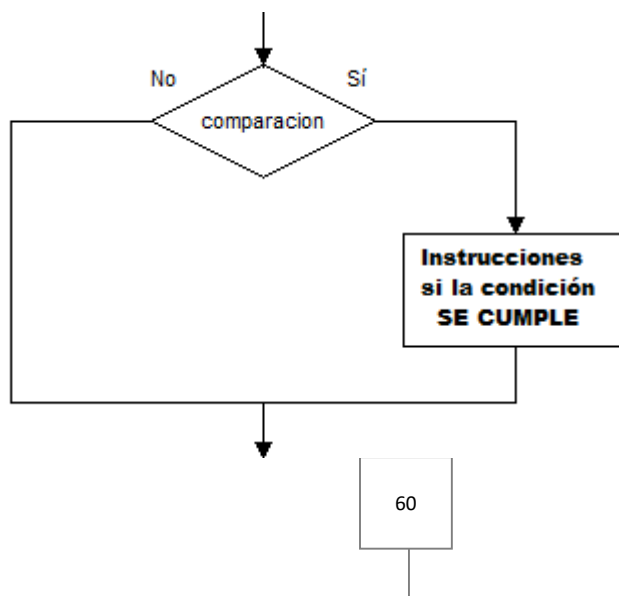
2.1.2 Clasificación de las Estructuras de Selección:

Simple:

Definición:

Las decisión simple es aquella instrucción que dependiendo del resultado que arroje la comparación ejecuta solo el bloque de instrucciones existentes por el verdadero.

Diagrama de Flujo:



Grafica 6 Diagrama de flujo decisión simple

PseudoCódigo:

·
·
·

Si (comparacion)

Verdadero

Instrucciones por el Verdadero

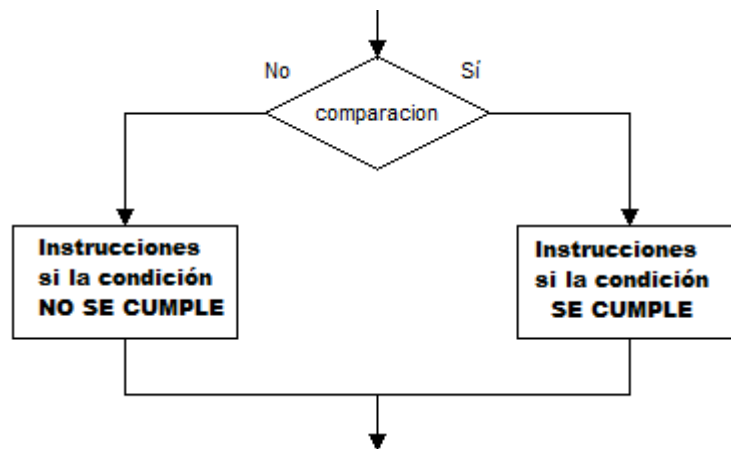
·
·

Doble:

Definición:

Las decisión doble es aquella instrucción que dependiendo del resultado que arroje la comparación ejecuta el bloque de instrucciones existentes por el verdadero o por el favor

Diagrama de Flujo:



Grafica 7 Diagrama de flujo decisión doble

PseudoCódigo:

·
·
·

Si (comparacion)

Verdadero

Instrucciones por el Verdadero

Falso

Instrucciones por el Verdadero

FIN_SI

•
•
•

Compuesta

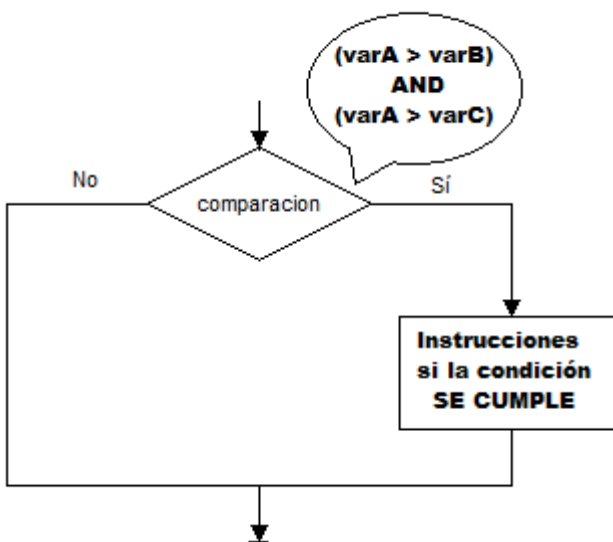
Definición:

La decisión compuesta es aquella instrucción que en la comparación incluye varias relaciones unidas por los operadores lógicos AND u OR.

AND: Cuando las relaciones están unidas por el operador **AND** todas las relaciones de comparación se deben de cumplir para que ejecute el bloque de instrucciones del camino VERDADERO.

OR: Cuando las relaciones están unidas por el operador **OR** solo se debe de cumplir una de todas las relaciones de comparación para que ejecute el bloque de instrucciones del camino VERDADERO.

Diagrama de Flujo:



Grafica 8 Diagrama de flujo decisión compuesta

PseudoCódigo:

•
•
•

Si (varA > varB) AND (vaA > varC)
Verdadero
Instrucciones por el Verdadero

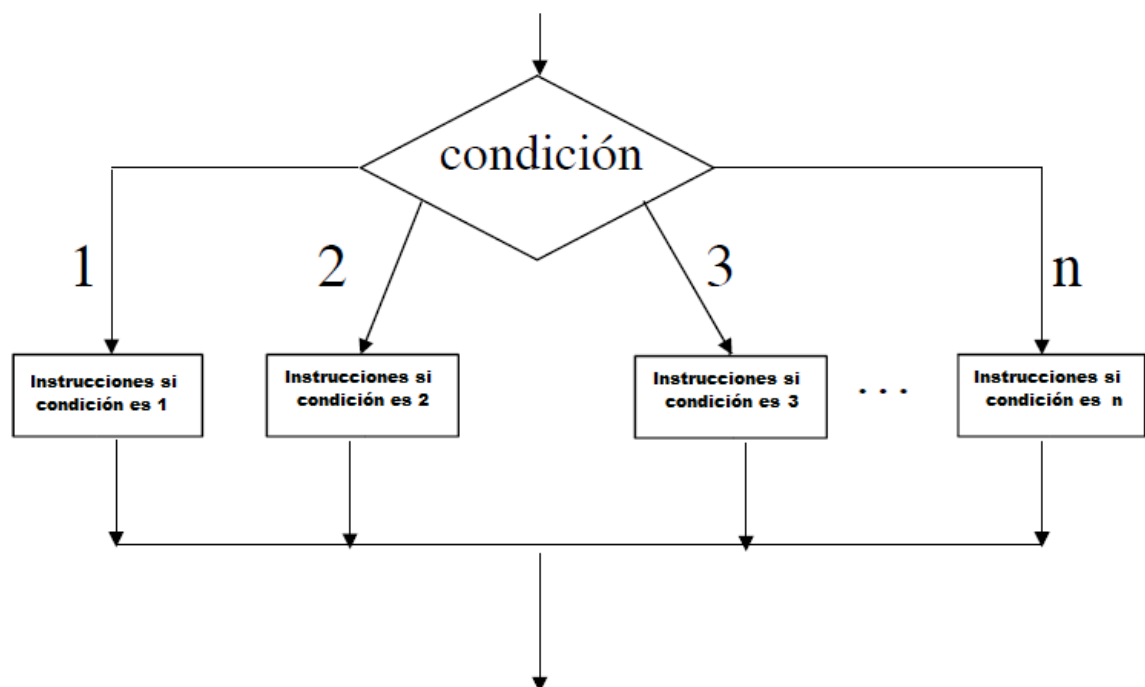
·
·
·

Múltiple

Definición:

Las decisión Multiple es aquella instrucción que dependiendo si la condición se cumple, ejecuta el bloque de instrucciones existentes de uno de N caminos posibles.

Diagrama de Flujo:



Grafica 9 Diagrama de flujo decisión múltiple

PseudoCódigo:

·
·

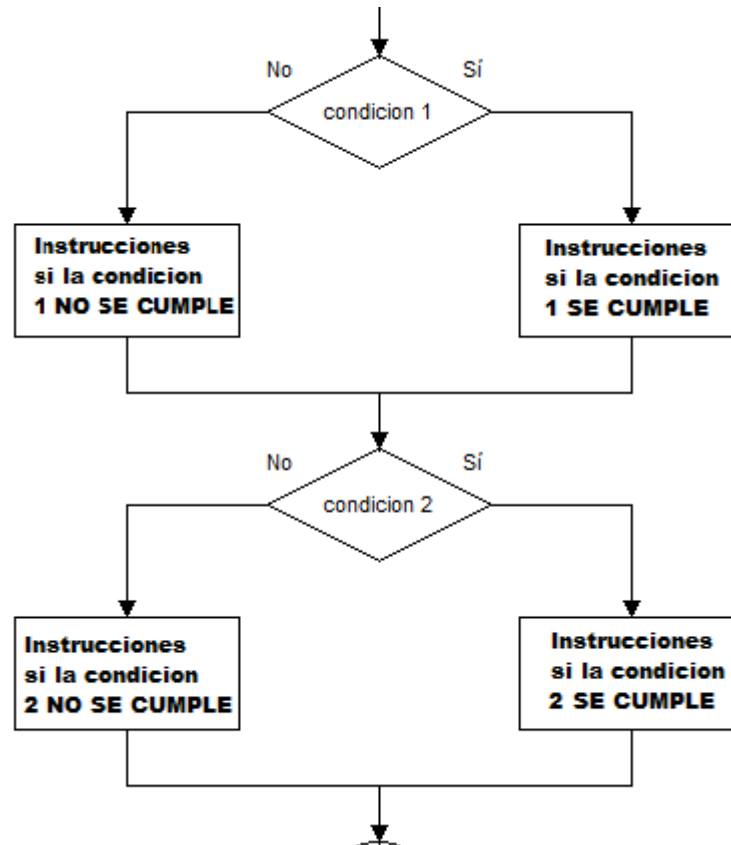
```
·  
DECISIONMULTIPLE(Condicion)  
  CASO 1 :  
    Instrucciones si Condicion = 1  
  CASO 2 :  
    Instrucciones si Condicion = 2  
  CASO 3 :  
    Instrucciones si Condicion = 3  
  ·  
  ·  
  ·  
  CASO n :  
    Instrucciones si Condicion = n  
  OTROSCASOS  
    Instrucciones si Condicion no se cumple  
  ·  
  ·  
  ·
```

Secuencial

Definición:

Las decisiones Secuenciales realmente no son una instrucción sino una forma de organización en caso de necesitar aplicar varias decisiones. Se ejecutan una a continuación de otras sin tener ninguna relación entre ellas.

Diagrama de Flujo:



Grafica 10 Diagrama de flujo decisión secuencial

PseudoCódigo:

```

.
.
.
Si_1 (comparación 1)
    Verdadero
        Instrucciones por el Verdadero
    Falso
        Instrucciones por el Verdadero
FIN_SI_1

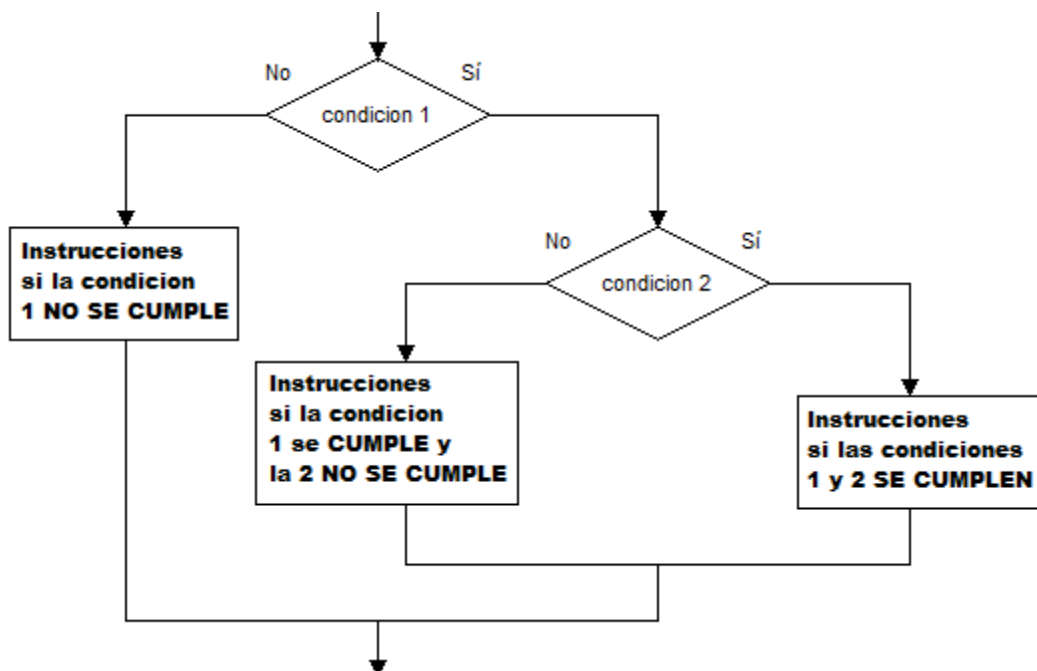
Si_2 (comparación 2)
    Verdadero
        Instrucciones por el Verdadero
    Falso
        Instrucciones por el Verdadero
FIN_SI_2
  
```

Anidada

Definición:

Las decisiones Anidadas no son una instrucción sino una forma de organización en caso de necesitar aplicar varias decisiones. Se ejecutan una a continuación de otras teniendo relación directa con la anterior. Para que la decisión más interna se ejecute, se debe de haber cumplido la decisión externa anterior.

Diagrama de Flujo:



Grafica 11 Diagrama de flujo decisión anidado

Pseudocodigo:

```
.  
. .  
Si_1 (comparación 1)  
    Verdadero  
        Si_2 (comparación 2)
```

Verdadero
Instrucciones por el Verdadero
Falso
Instrucciones por el Falso
FIN_SI_2
Falso
Instrucciones por el Verdadero
FIN_SI_1
.
.
.
.

ESTRUCTURAS DE REPETICIÓN:

Las estructuras de repetición es una poderosa instrucción que nos permite repetir una o más instrucciones previa la realización de una evaluación.

Partes de una estructura de repetición:

Clasificación de las Estructuras de Repetición:

Para:

Definición:

Las Estructuras de repetición PARA es una de las instrucción que permite repetir la ejecución de un bloque de instrucciones previa la verificación de algunas características como son:

Inicio:

Evalúa si la condición de inicio del ciclo se cumple para empezar a repetir las instrucciones.

Fin:

Evalúa si la condición para que finalice del ciclo se cumple y dejar de repetir las instrucciones.

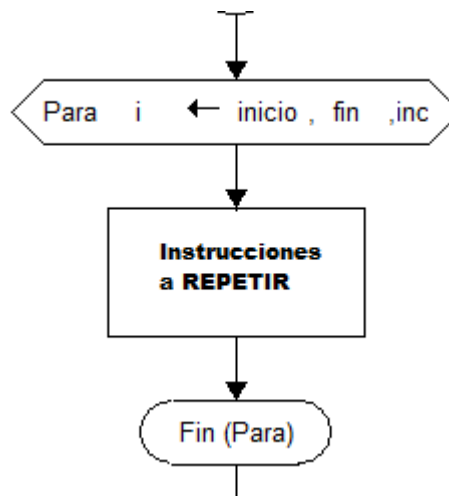
Incremento:

Es la instrucción que incrementa en una cantidad determinada el número de vueltas que da el ciclo.

Características de Ciclo Para:

- Conocer el inicio
- Conocer el número de vueltas que va a realizar
- Conocer el incremento

Diagrama de Flujo:



Grafica 12 Diagrama de flujo, Estructuras de Repetición: Para

Pseudocódigo:

•
•
•

PARA (contador=INICIO, contador<=FIN, INCREMENTO)
INSTRUCCIONES A REPETIR.
FIN_PARA

•

Ejecute Mientras:

Definición:

Las Estructuras de repetición EJECUTE_MIENTRAS es una de las instrucción que permite repetir la ejecución de un bloque de instrucciones mientras una condición se cumpla, sus características son:

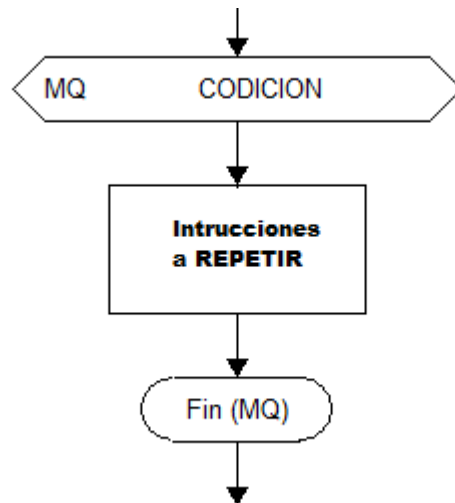
Condición del Ciclo:

Evalúa si la condición de inicio del ciclo se cumple para empezar a repetir las instrucciones.

Características de Ejecute Mientras:

- Conocer la condición de inicio
- La condición se evalúa al comienzo de la instrucción
- Es controlado por contador
- No se necesita conocer ni el número de repeticiones ni el incremento.

Diagrama de Flujo:



Grafica 13 Diagrama de flujo, Estructuras de Repetición: Ejecute Mientras

Pseudocodigo:

•
•
•

EJECUTE_MIENTRAS(CONDICION)
INSTRUCCIONES A REPETIR.
FIN_EJECUTE_MIENTRAS

•
•
•

Ejemplo:

Repita - Hasta:

Definición:

Actividades de aprendizaje

Actividades Autónomas

Actividad N° 2
Nombre de la actividad: Mapa mental sobre los conceptos fundamentales referentes a la resolución de problemas de forma algorítmica.
Nombre del recurso a utilizar. FreeMind, préstamo en línea, tarea moodle
Competencias Definir los conceptos básicos en las etapas del ciclo de desarrollo de un programa.
Indicaciones pasó a paso: <ol style="list-style-type: none">Después de realizar la lectura a unidad 02 del módulo y de los textos sugeridos,Lectura de los Capítulos 4 y 5 del libro de Fundamentos de Programación de Luis Joyanes Aguilar Tercera EdiciónRealice un mapa mental que represente los elementos fundantes necesarios para resolver problemas de manejo de información mediante algoritmos y el uso de estructuras de selección y repetición.Para ello se sugiere emplear el software FreeMind, en el aula virtual encuentra el tutorial y el enlace para descargar.Suba el archivos en el recurso tarea
Producto a entregar Plantilla diligenciada por cada uno de los ejercicios, subirlas en el recurso tarea.
Fecha límite cuarta semana
Porcentaje de valoración 5%

Actividad N° 3
Nombre de la actividad: Aplicación de la plantilla Guía para la Resolución de Problemas con Apoyo del Computador.
Nombre del recurso a utilizar. Tarea-Moodle
Competencias <ul style="list-style-type: none">Representa en un diagrama la entrada y salida de datos del problema que desea resolver.Diseña algoritmos funcionales empleando estructuras condicionales o de repetición según el tipo de problema a resolver.Realizar diagramas de flujo para representar la solución de problemas utilizando la notación establecida.Escribir el pseudocódigo que permita representar algoritmos propuestos para la solución del problema
Indicaciones pasó a paso:

Después de realizar la lectura de la unidad 02 del módulo, diligencie la plantilla guía para la resolución de problemas para cada uno de los siguientes ejercicios propuestos:

4. Capture por teclado un número entero e imprima por pantalla si el número es par o impar.
5. Lea por teclado el nombre de un estudiante, el nombre de una materia y la nota obtenida para cada uno de las 3 evaluaciones que se le practicaron durante el periodo académico. Se pide :
 - Calcular la nota definitiva sabiendo que se calcula por promedio
 - Imprimir un mensaje si aprobó o reprobó la asignatura sabiendo que para ganarla la nota definitiva debe ser igual o mayor a 3.0
6. Capture un numero entero por teclado y realice el calculo el factorial de ese número.

Producto a entregar

Plantilla diligenciada por cada uno de los ejercicios, subirlas en el recurso tarea.

Fecha límite: Quinta semana

Porcentaje de valoración 10%

Actividades colaborativas

Actividad N° 4

Nombre de la actividad

Realización de ejercicios propuestos.

Nombre del recurso a utilizar.

Tarea de moodle, préstamo en línea

Competencias o indicador de competencias

- Diseña algoritmos funcionales empleando estructuras condicionales o de repetición según el tipo de problema a resolver.
- Realizar diagramas de flujo para representar la solución de problemas utilizando la notación establecida.
- Escribir el pseudocódigo que permita representar algoritmos propuestos para la solución del problema.
- Definir los conceptos básicos en las etapas del ciclo de desarrollo de un programa.

Indicaciones pasó a paso:

Inicie con las lecturas:

- Lectura del Unidad 02 del módulo.
- Lectura: Del Libro Fundamentos de Programación del Luis Joyanes Aguilar, el capítulo 4 (Flujo s de Control Estructuras Selectivas), y 5 (Flujo s de Control. Estructuras Repetitivas)
- Realice los problemas propuestos al final de cada uno de los capítulos.

Momento Individual:

Realice las lecturas del Libro Fundamentos de Programación de Luis Joyanes Aguilar de los capítulo 4 (Flujo s de Control Estructuras Selectivas), y 5 (Flujo s de Control. Estructuras Repetitivas). Plantee en el foro las inquietudes y las preguntas que surjan respecto a las lecturas.

Momento de Socialización:

Coserte con su grupo horarios de encuentro para que realicen los comparativos sobre las soluciones que cada uno propuso a los ejercicios y selección la que consideren con mayor eficiencia.

Construcción Colectiva:

El equipo enviará el trabajo con los ejercicios resueltos anexando una conclusión referente a las estrategias utilizadas en el desarrollo del mismo y las ventajas que éstas representan.

Producto a entregar

Documento con los ejercicios resueltos.

Fecha límite Sexta semana

Porcentaje de valoración 10%

Actividades de Acompañamiento tutorial

Actividad N°1 Presentación de la Unidad II

Nombre del recurso a utilizar. Video conferencia

Competencias o indicador de competencias.

Indicaciones pasó a paso: En el chat se realiza la presentación de los contenidos y actividades de la Unidad de Aprendizaje II.

En este encuentro se presentan los aspectos a tener en cuenta para el uso de ciclos y estructuras.

Temas a tratar:

- Condición
- Operadores Relacionales
- Decisiones, su clasificación, funcionamiento y aplicación Estructuras repetitivas, su clasificación, funcionamiento y aplicación
- Representación gráfica y en pseudocódigo de estructuras condicionales y repetitivas
- Informe sobre las competencias alcanzadas en las diferentes actividades realizadas.
- Los estudiantes harán participación activa con base en las preguntas que surjan durante la exposición a en el abordaje de las lecturas recomendadas para la unidad

Fecha límite cuarta semana

Porcentaje de valoración 5%

Actividad N° 5

Nombre de la actividad Retroalimentación de la Unidad II

Competencias

1. Demostrar interés por la actualización permanente a través del acceso a diferentes fuentes de información.
2. Cumplir con el cronograma de trabajo estableciendo su propia dinámica de aprendizaje

Nombre del recurso a utilizar. Video conferencia

Indicaciones pasó a paso:

La video conferencia tiene como fin realizar la retroalimentación de los temas desarrollados en la Unidad II y socializar aspectos de los trabajos individuales y en equipo.
En este encuentro se discute sobre los aspectos a tener en cuenta para el uso de ciclos y estructuras Las pregunta orientadoras se planean a partir de la identificación de las necesidades que surjan en el desarrollo de la unidad II

Fecha Límite sexta semana

Porcentaje de valoración 5%

Evaluación de los aprendizajes Unidad 2

Resultados del aprendizaje:

- Aplica estructuras de decisión y de repetición en las soluciones en los casos necesarios.
- Convertir las estructuras de decisión y de repetición de diagramas de flujo a pseudocódigo
- Diagrama y verifica soluciones algorítmicas en las que intervienen estructuras de decisión y de repetición mediante diagrama de flujo
- Convierte diagramas de flujo y sus elementos a escritura de pseudocódigos

Actividad	Competencias	Criterios	%
Actividades autónomas			
Actividad Nº 2 <i>Mapa mental ciclos y estructuras</i>	<ul style="list-style-type: none"> • Definir los conceptos básicos en las etapas del ciclo de desarrollo de un programa. 	<ul style="list-style-type: none"> • Describe las etapas del ciclo de desarrollo • Caracteriza los hitos de cada etapa 	5%
Actividad Nº 3 <i>aplicación de plantilla guía</i>	<ul style="list-style-type: none"> • Representa en un diagrama la entrada y salida de datos del problema que desea resolver. • Diseña algoritmos funcionales empleando estructuras condicionales o de repetición según el tipo de problema a resolver. • Realizar diagramas de flujo para representar la solución de problemas utilizando la notación establecida. • Escribir el pseudocódigo que permita representar algoritmos propuestos para la solución del problema 	<ul style="list-style-type: none"> • Utiliza la notación para diagramas de entrada, proceso y salida de datos. • Representa estructuras condicionales • Representa estructuras repetitivas • Aplica las estructuras básicas de programación a la solución de problemas. • Emplea la notación especificada en el diseño de algoritmos • Utiliza la notación para diagramas de flujo • Representa un algoritmo mediante el diagrama de flujo • Escribe el diagrama de flujo en el pseudocódigo correspondiente • Utiliza la estructura lógica del pseudocódigo 	10%
Actividades colaborativas			
Actividad Nº 4 <i>Ejercicios de aplicación</i>	<ul style="list-style-type: none"> • Diseña algoritmos funcionales empleando estructuras condicionales o de repetición según el tipo de problema a resolver. • Realizar diagramas de flujo para representar la solución de problemas utilizando la notación establecida. 	<ul style="list-style-type: none"> • Representa estructuras repetitivas • Aplica las estructuras básicas de programación a la solución de problemas. • Emplea la notación especificada en el diseño de algoritmos • Utiliza la notación para diagramas de flujo • Representa un algoritmo mediante el diagrama de flujo 	10%
Actividades de acompañamiento tutorial			

Actividad N°1 Presentación y contextualización del Módulo	<ul style="list-style-type: none"> • Interactuar (con los participantes, tutor, docente y contenidos) en el aula virtual a través de los recursos y actividades para alcanzar las competencias propuestas. 	<ul style="list-style-type: none"> • Participa en los encuentros síncronos definidos en el módulo • Interactúa con los compañeros a través de medios síncronos y asíncronos • Accede en forma permanente al aula virtual para consultar los contenidos • Utiliza de los recursos tecnológicos necesarios para acceder a los contenidos y realizar las actividades 	5%
Actividad N° 5 Retroalimentación Unidad I	<ul style="list-style-type: none"> • Demostrar interés por la actualización permanente a través del acceso a diferentes fuentes de información. • Cumplir con el cronograma de trabajo estableciendo su propia dinámica de aprendizaje 	<ul style="list-style-type: none"> • Participa en los encuentros síncronos definidos en el módulo • Interactúa con los compañeros a través de medios síncronos y asíncronos • Accede en forma permanente al aula virtual para consultar los contenidos • Utiliza de los recursos tecnológicos necesarios para acceder a los contenidos y realizar las actividades. 	5%

CODIFICACIÓN Y LENGUAJES DE PROGRAMACIÓN

Competencias y tabla de saberes unidad 3

Unidad de aprendizaje 3		
Competencias		
<p>Definir los conceptos básicos en las etapas del ciclo de desarrollo de un programa.</p> <p>Identificar las características principales de un lenguaje de programación y su función en el desarrollo de programas.</p> <p>Codificar artefactos de software empleando el paradigma orientado a objetos y el lenguaje de programación definido</p> <p>Manipular el código fuente teniendo en cuenta las especificidades de cada uno de ellos.</p> <p>Documentar artefactos de software y elaborar sus respectivos manuales de acuerdo con el paradigma empleado</p> <p>Validar la coherencia de la solución y su funcionamiento aplicando los correctivos necesarios</p>		
Tabla de saberes		
Saber hacer	Saber saber	Saber ser
<p>Representar pseudocodigos a lenguaje de programación</p> <p>Documentar artefactos de software</p> <p>Ejecutar pruebas y realizar correcciones a código fuente.</p>	<p>Características fundamentales de la programación orientada a objetos</p> <p>Sintaxis e Instrucciones del lenguaje de programación JAVA</p> <p>Procesos de Edición, Compilación y ejecución de programas</p> <p>Manuales del programador y del usuario</p>	<p>Respetuoso cuando se expresa hacia los compañeros de trabajo y hacia el tutor.</p> <p>Consciente y ético para realizar las actividades.</p> <p>Organizado para presentar las actividades.</p> <p>Responsable en entregar completas las actividades</p> <p>Cumplido en entregar las actividades</p> <p>Interesado en realizar actividades colaborativas.</p> <p>Interesado a la hora de consultar temas referentes a los temas propuestos.</p> <p>Creativo para realizar las actividades propuestas durante la ejecución del módulo</p> <p>Respetuoso con los derechos de autor al presentar trabajos escritos.</p>

3 CODIFICACIÓN Y LENGUAJES DE PROGRAMACIÓN

RESUMEN

La solución a los problemas de manejo de información propuestas mediante la utilización de las tecnologías de la información, hacen un tránsito desde la etapa pre computacional a la computación, debido a esto pertinente que el profesional en desarrollo de software tenga un pleno conocimiento de cómo llevar una solución diseñada en algoritmo o pseudocódigo a un lenguaje de programación y aplicar los pasos para el desarrollo de software como son la edición la compilación y la ejecución. En la actualidad los lenguajes de programación que se están imponiendo son aquellos que permiten la orientación a objetos, esto debido a las facilidades que presentan para la reutilización y mantenimiento del código fuente.

3.1 Pasos para llevar un llevar la solución a un problema del pseudocódigo al código fuente:

3.1.1 Etapas en el desarrollo de Software:

Edición:

La edición es el proceso mediante el cual traducimos el Pseudocódigo a un Lenguaje de programación. Para realizar estas traducciones necesitamos conocer las reglas léxicas y sintácticas de lenguaje que vamos a utilizar. Esta etapa es desarrollada por el programador.

Compilación

La compilación es el proceso mediante el cual el lenguaje de programación verifica que el código fuente digitado cumpla con las reglas léxicas y sintácticas del lenguaje, de no ser así mostrará un aviso de error indicando la línea y el concepto por el cual se presentó. Hasta no corregir todos los errores detectados en la etapa de compilación no se podrá avanzar a la siguiente etapa de ejecución. Esta etapa es desarrollada por el programador.

Ejecución

La Ejecución es el proceso mediante el cual podremos ver en pleno funcionamiento en un computador la solución propuesta, a ella se llega luego de cumplir a satisfacción y coherencia todas las etapas analizadas en el presente modulo. En esta etapa interviene tanto el programador como el usuario final del programa o software.

Verificación y Pruebas:

La Verificación y Pruebas es el proceso mediante el cual realizamos un testing del software en modo de ejecución para detectar y corregir los últimos detalles que se puedan presentar. Este es el paso anterior a la entrega del proyecto.

Documentación:

La Documentación es el proceso mediante el cual realizamos guías para una fácil comprensión de las líneas de código por parte de las personas que a futuro le hagan mantenimiento o actualización al código fuente así como a los usuarios para una fácil consulta sobre la forma de usar el software.

3.1.2 Lenguajes de Programación:

Java:

Entornos Integrados de Desarrollo:

NetBeans:

Que es?

Es una herramienta de software que sirve como entorno integrado de desarrollo donde se puede realizar de forma fácil los procesos de Edición, Compilación y Ejecución.

Donde Bajarlo?

Se puede obtener de la dirección:

<http://netbeans.org/community/releases/69/>

PASOS PARA REALIZAR UN PROYECTO EN JAVA UTILIZANDO CON IDE NETBEANS:

1. Tenga a la mano el enunciado, léalo y analícelo con detenimiento
2. Aplique los pasos de la TÉCNICA, explicados anteriormente.
3. Cree un Proyecto
4. Cree una Interfaz
5. Agregue a la interfaz los elementos del DISEÑO DE INTERFACES modificando los nombres de las variables y los valores de las propiedades que considere.
6. Declare e inicialice las variables de acuerdo a la TABLA DE DATOS
7. Ingrese a cada Boton y realice la programación de acuerdo al DIAGRAMA DE FLUJO y al PSEUDOCODIGO.
8. Ejecute el programa y verifique su correcto funcionamiento

Ejemplo:

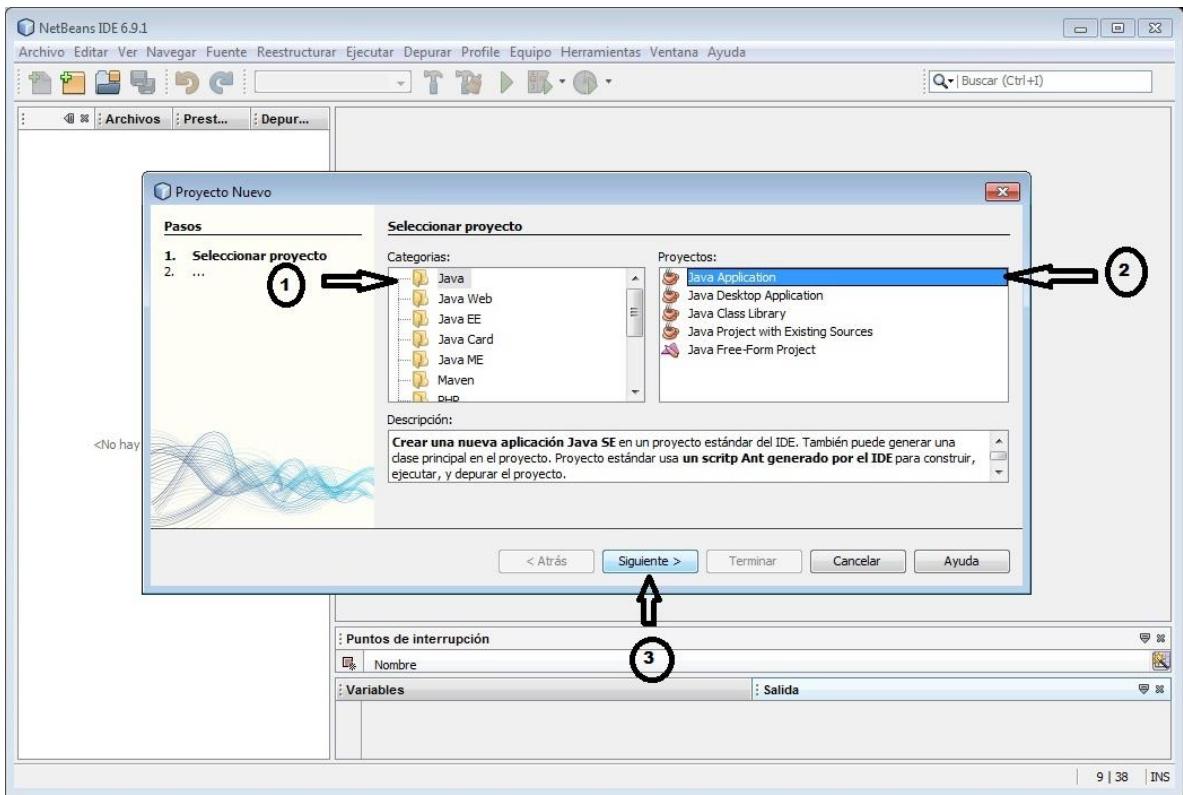
Retomando el problema de ejemplo de la primera unidad, lo llevaremos a la etapa de código fuente donde aplicaremos cada uno de los siguientes pasos descritos:

PASOS 1 Y 2: TENGA A LA MANO EL ENUNCIADO, LÉALO Y ANALÍCELO CON DETENIMIENTO:

PASO 3: CREE EL PROYECTO:

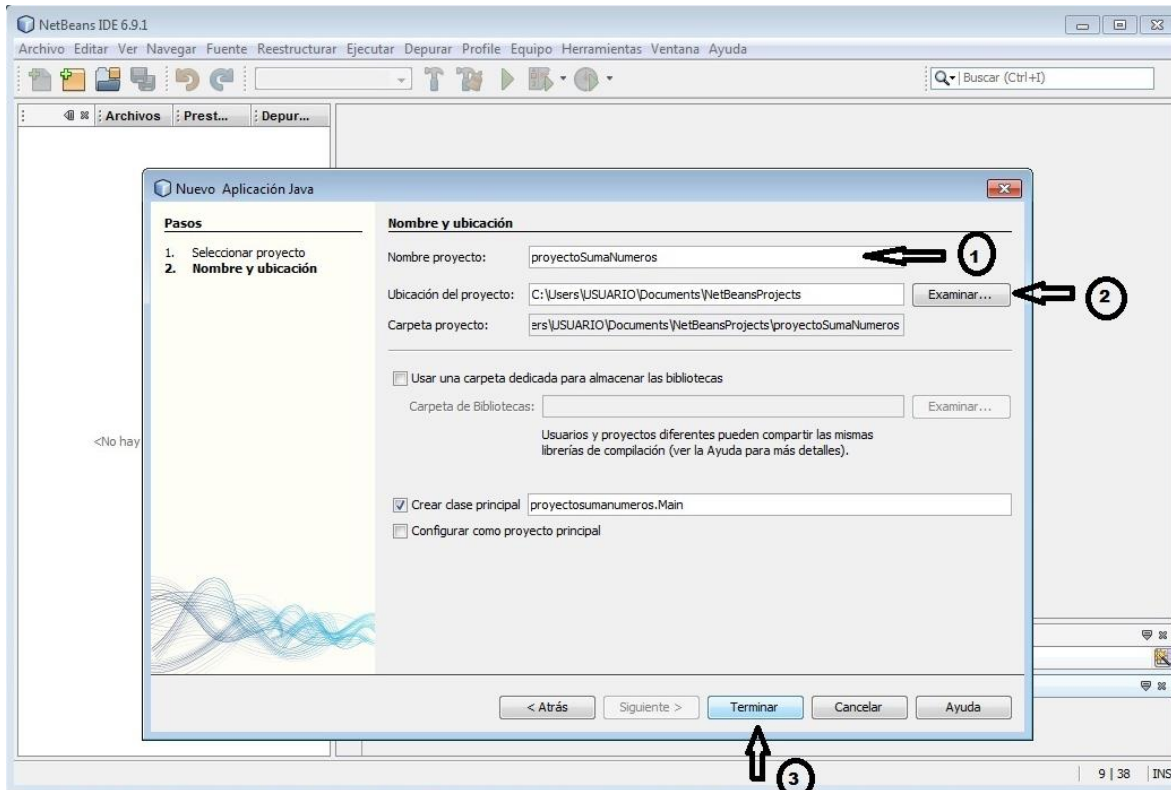
Cargue el programa NetBeans y cree un proyecto siguiendo los pasos como se describen en la imagen a continuación:

Seleccione del menú Desplegable la opción Proyecto Nuevo y aparecerá una ventana similar a la siguiente:



Paso	Descripción
0	Archivo – Proyecto Nuevo
1	En la ventana Categorías escoja Java
2	En la ventana de Proyectos escoja Java Aplication
3	Haga CLICK en el botón Siguiente

A continuación podrá dar nombre y ubicación al proyecto, así:

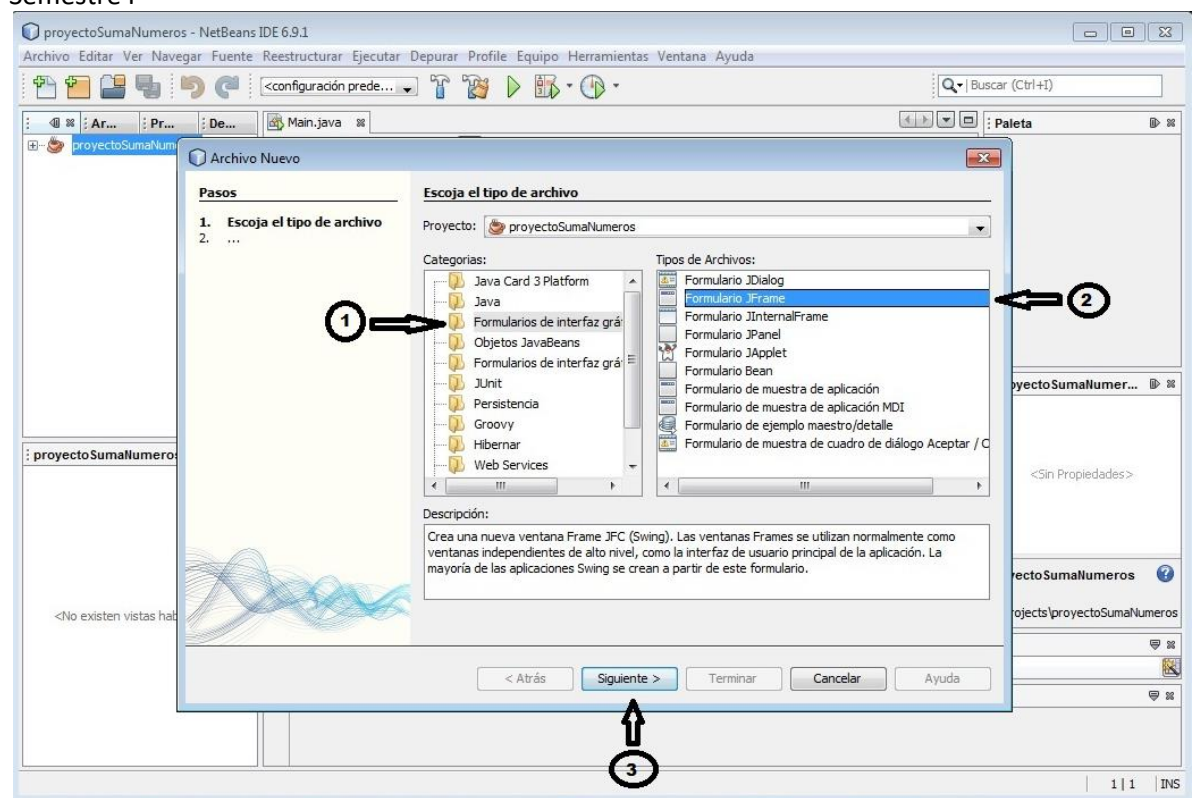


Paso	Descripción
1	Digite el nombre del Proyecto sin espacios en blanco
2	Seleccione la ruta donde desee que quede almacenado
3	Haga CLICK en el botón Terminar

PASO 4: Cree una Interfaz:

Para crear la interfaz de usuario siga los siguientes pasos como se describe en la imagen a continuación:

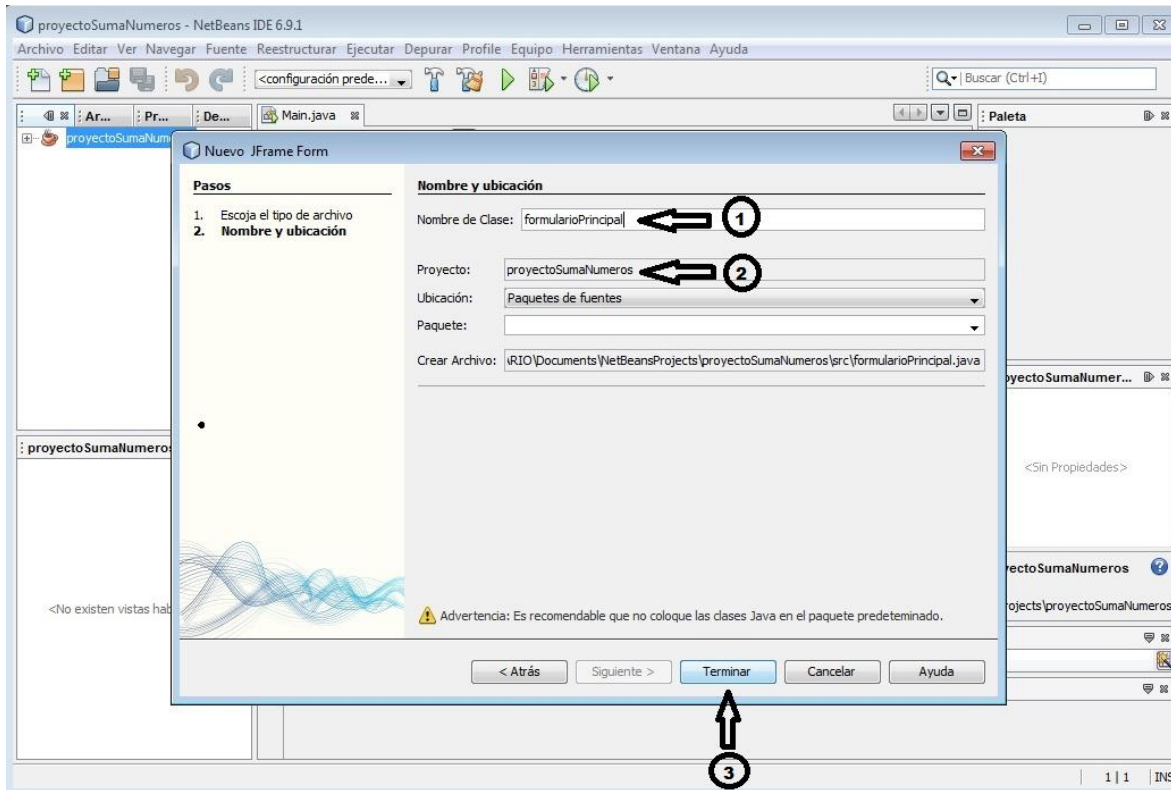
Módulo de Aprendizaje
Desarrollo de Algoritmos
Semestre I



Paso	Descripción
0	Archivo – Archivo Nuevo
1	En la ventana Categorías escoja Formulario de Interfaz Gráfica
2	En la ventana Tipos de Archivo escoja Formulario JFrame
3	Haga CLICK en el botón Siguiente

A continuación podrá dar nombre al formulario, así:

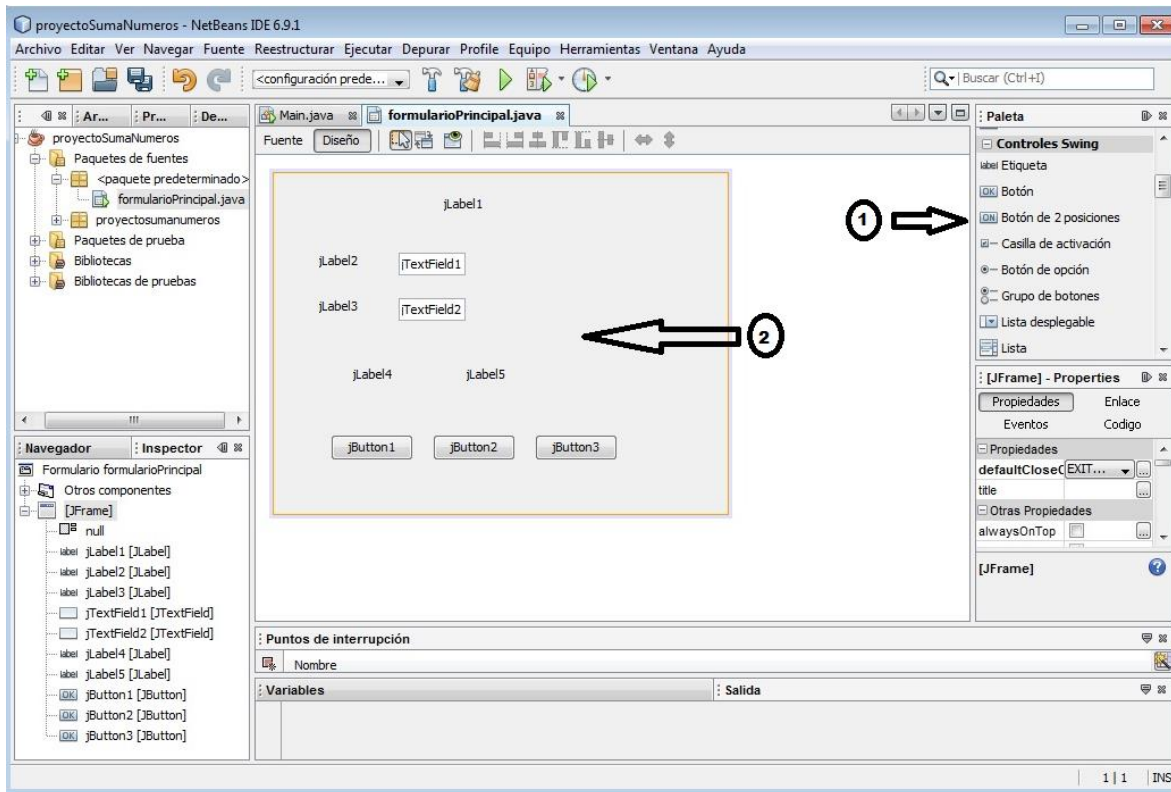
Módulo de Aprendizaje
Desarrollo de Algoritmos
Semestre I



Paso	Descripción
1	Digite el nombre del Formulario sin espacios en blanco
2	Seleccione el proyecto al cual pertenece el formulario
3	Haga CLICK en el botón Terminar

PASO 5: Agregue a la interfaz los elementos del DISEÑO DE INTERFACES

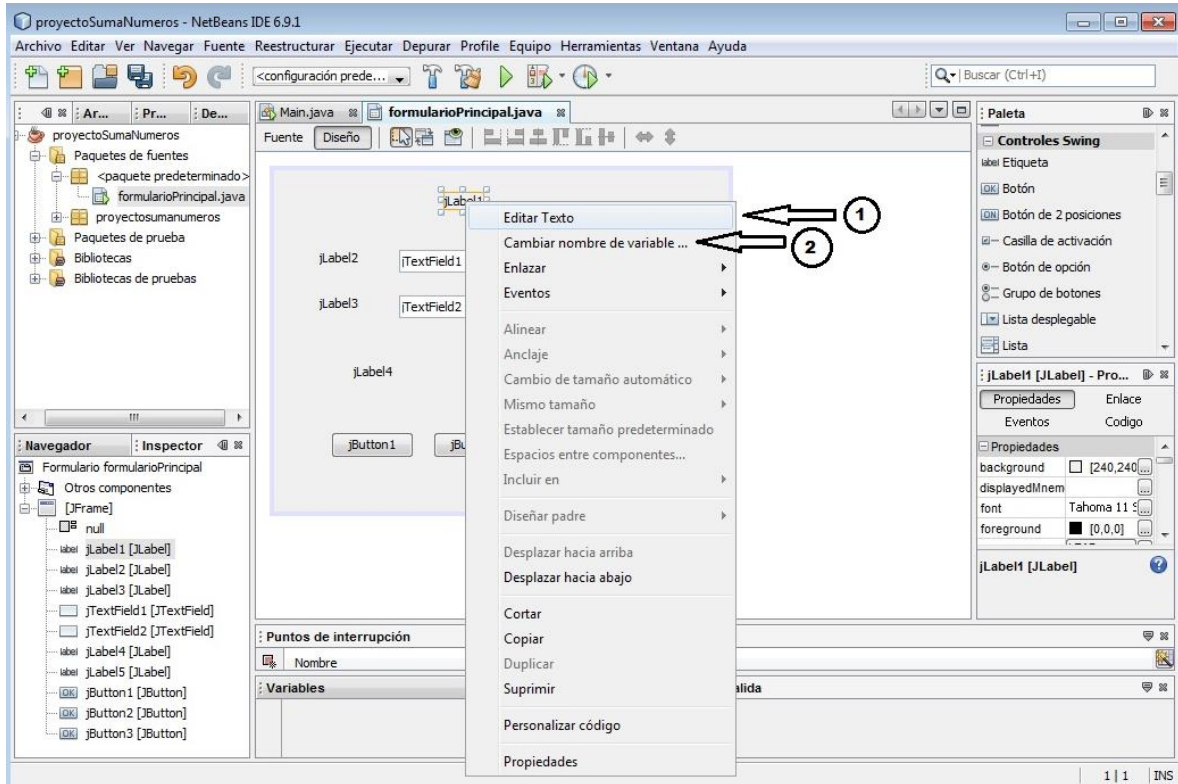
Módulo de Aprendizaje
Desarrollo de Algoritmos
Semestre I



Paso	Descripción
1	Seleccione los objetos de la Paleta
2	Arrastre y ubique los objetos en el formulario de acuerdo al diseño de la interfaz

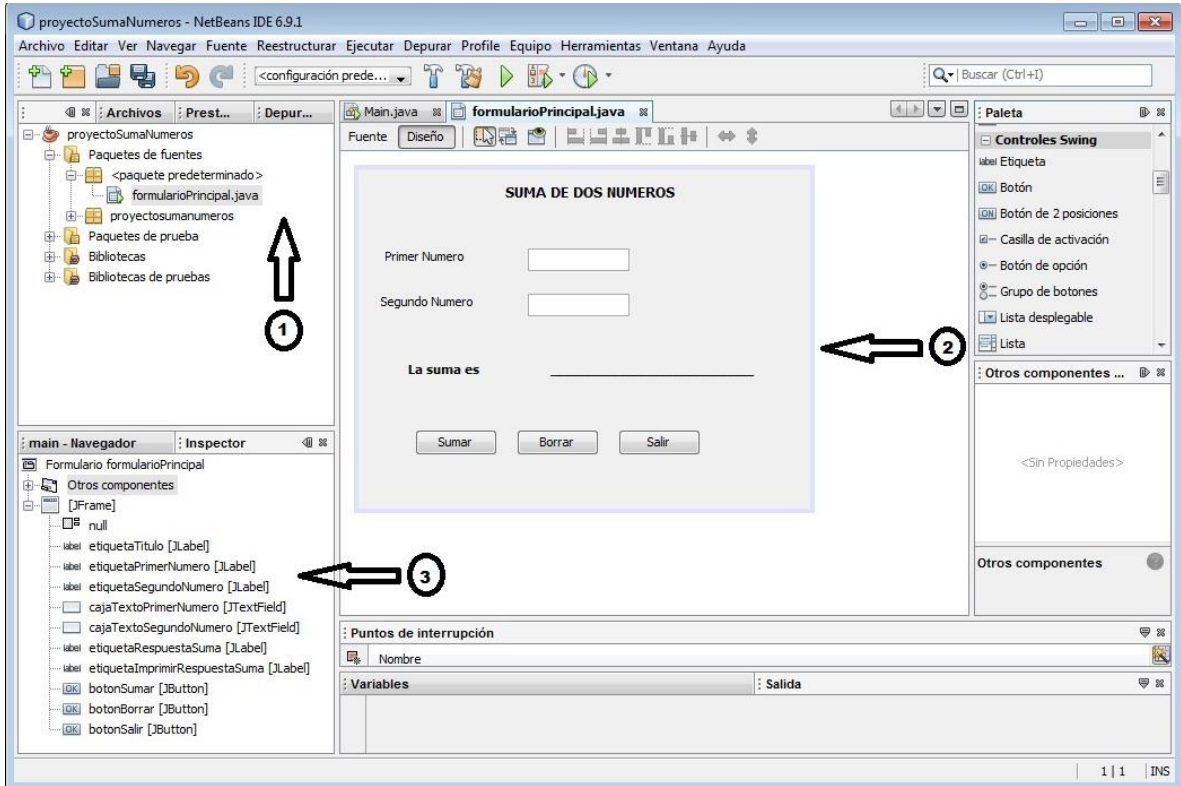
Ahora Modifico tanto el nombre de los objetos como sus propiedades de acuerdo a la necesidad:

Módulo de Aprendizaje
Desarrollo de Algoritmos
Semestre I



Paso	Descripción
1	Modifico el texto que va a aparecer en el formulario cuando se ejecute
2	Modifico el nombre del objeto que es con el que se va a conocer en el código fuente

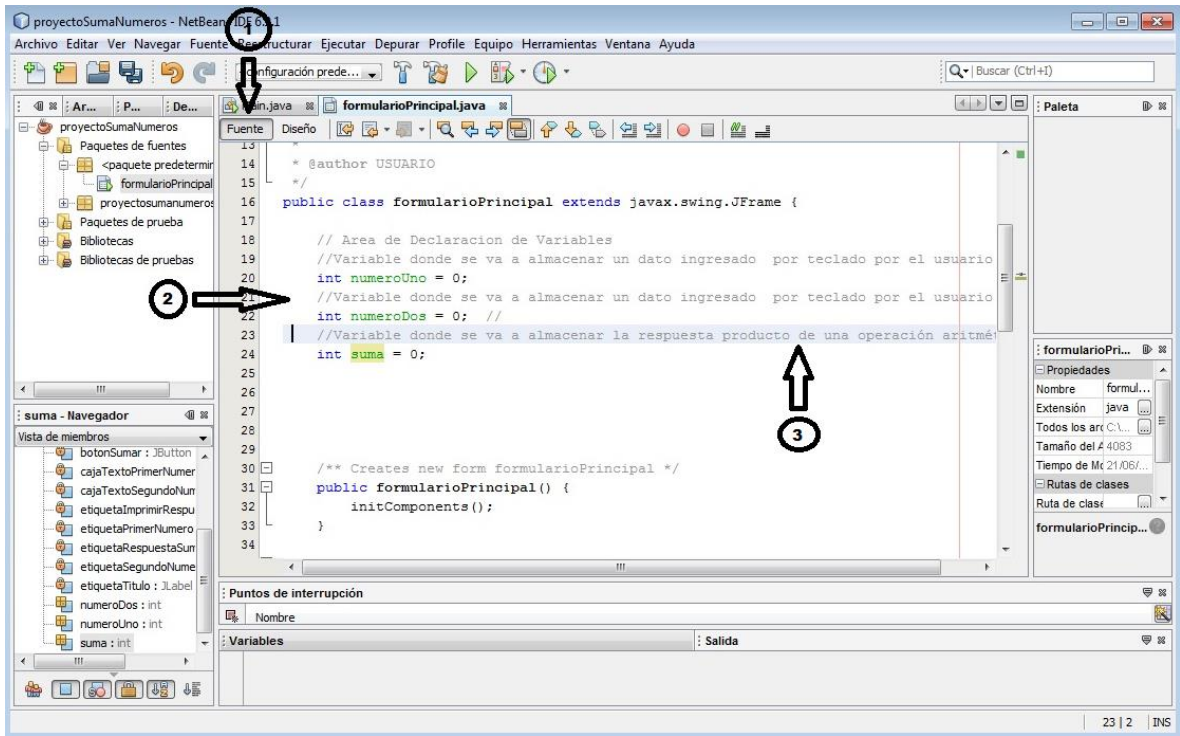
Y obtenemos como resultado final el formulario con las siguientes características:



Paso	Descripción
1	En la ventana de proyectos queda incluido el formulario
2	El formulario queda con una presentación similar a la del diseño de interfaces
3	Todos los componentes del formulario quedan en la pestaña inspector

PASO 6: DECLARE E INICIALICE LAS VARIABLES DE ACUERDO A LA TABLA DE DATOS:

En la pestaña de código fuente agrego la codificación de acuerdo a la tabla de datos y de procesos, de la siguiente manera:

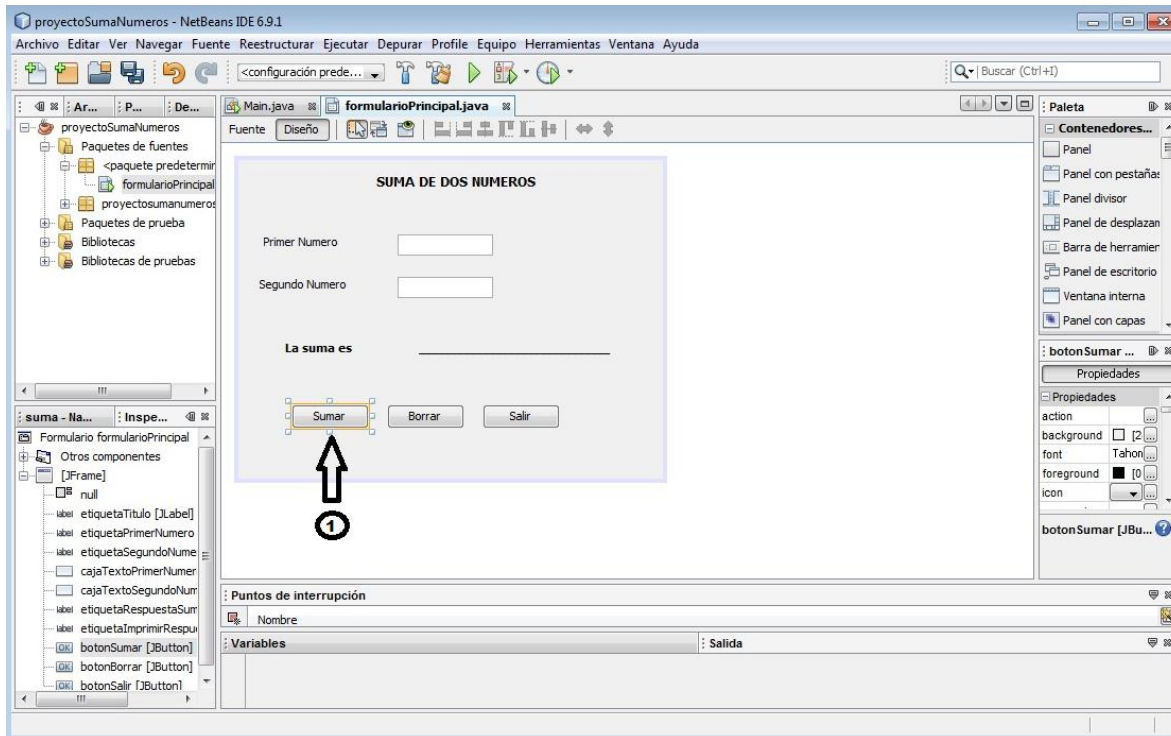


Paso	Descripción
1	Hago click en la pestaña fuente para poder adicionar codificación
2	Declaro las variables de acuerdo a la tabla de datos aplicando las reglas del lenguaje de programación escogido.
3	Documente el código fuente de manera concisa y clara

PASO 7: INGRESE A CADA BOTON Y REALICE LA PROGRAMACIÓN:

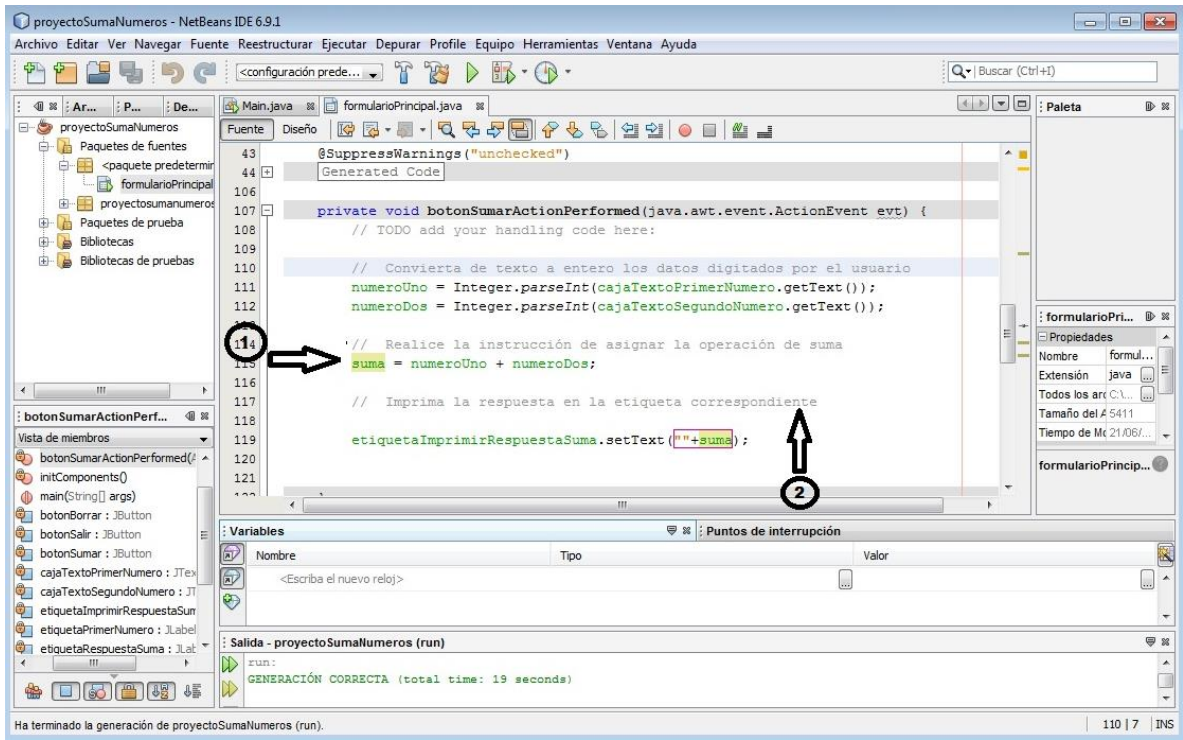
Digite la programación en cada botón de acuerdo al pseudocódigo y aplicando las instrucciones computacionales

Módulo de Aprendizaje
Desarrollo de Algoritmos
Semestre I



Paso	Descripción
1	Hago doble click en el botón sumar para ingresar a modo de codificación

Codifique de acuerdo a las necesidades y los procesos descritos tanto en el diagrama de flujo como en el pseudocódigo, así:



Paso	Descripción
1	Codifico las instrucciones de acuerdo al pseudocódigo y aplicando las reglas del lenguaje de programación escogido.
2	Documente el código fuente de manera concisa y clara

PASO 8: EJECUTE EL PROGRAMA Y VERIFIQUE SU CORRECTO FUNCIONAMIENTO:

Ejecuto el programa para ver su correcto funcionamiento y posición de los elementos en el formulario.

SUMA DE DOS NUMEROS

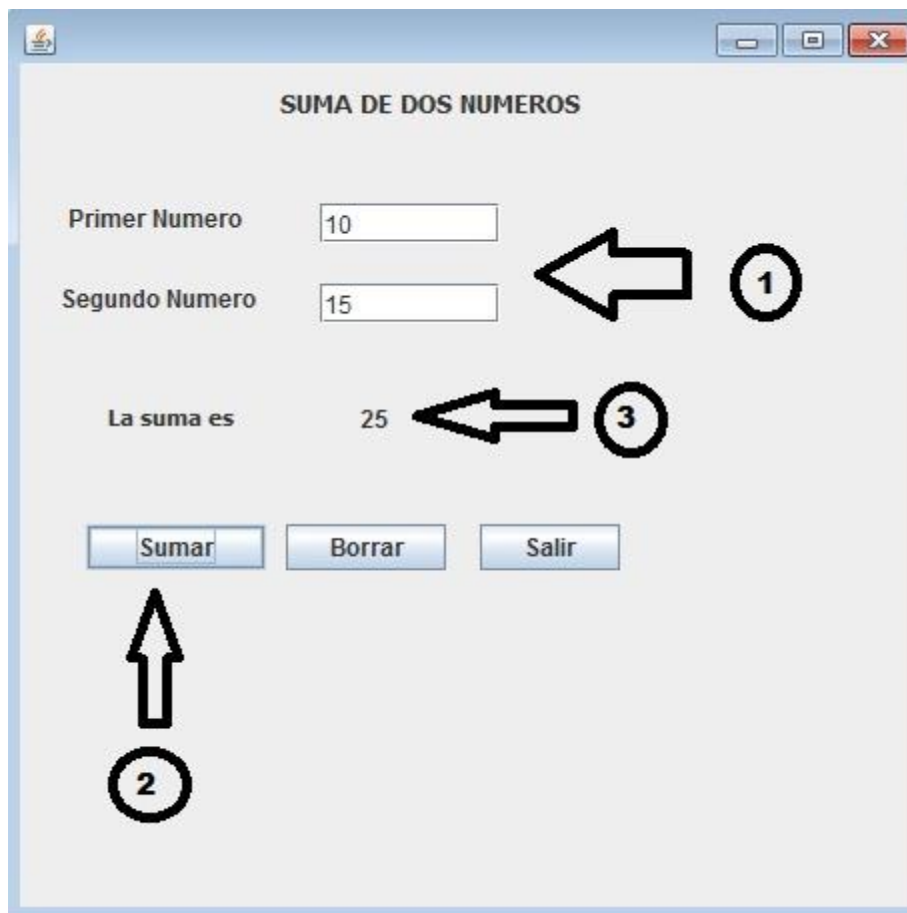
Primer Numero

Segundo Numero

La suma es _____

Paso	Descripción
1	En el menú ejecutar selecciono la opción ejecutar archivo

Ingreso datos aproximado a la realidad para verificar el correcto funcionamiento de la lógica de la aplicación y que apoye en la solución al problema, puedo utilizar los mismos datos utilizados en la prueba de escritorio y hacer una comparación de resultados los cuales deben ser exactamente iguales, así:



Paso	Descripción
1	Digito Datos de prueba en las cajas de texto
2	Hago Click en el botón Sumar para que se ejecute las instrucciones correspondientes
3	Verifico que la respuesta sea igual a la de las pruebas de escritorio.

UNIDAD 3

Actividades de aprendizaje

Actividades Autónomas

Actividad N° 2
Nombre de la actividad: Mapa mental sobre los conceptos fundamentales referentes a la resolución de problemas de forma algorítmica.
Nombre del recurso a utilizar. FreeMind, préstamo en línea, tarea moodle
Competencias Identificar las características principales de un lenguaje de programación y su función en el desarrollo de programas
Indicaciones pasó a paso: Realice las siguientes lecturas <ul style="list-style-type: none">• Lectura de la unidad 03 del módulo.• Lectura de Manuales Referentes al lenguaje java.• Lectura de Manuales Referentes al IDE NetBeans.• Después de realizar la lectura a unidad 03 del módulo y de los textos sugeridos, realice un mapa mental que represente los elementos fundantes necesarios para codificar artefactos de software en un lenguaje de programación.• Para ello se sugiere emplear el software FreeMind, en el aula virtual encuentra el tutorial y el enlace para descargar.
Producto a entregar Mapa mental de forma digital
Fecha límite: séptima semana
Porcentaje de valoración 5%

Actividades colaborativas

Actividad N° 3
Nombre de la actividad Realización de ejercicios propuestos.
Nombre del recurso a utilizar. Tarea-moodle
Competencias o indicador de competencias <ul style="list-style-type: none">• Escribir el pseudocódigo que permita representar algoritmos propuestos para la solución del problema• Documentar artefactos de software y elaborar sus respectivos manuales de acuerdo con el paradigma empleado Validar la coherencia de la solución y su funcionamiento aplicando los correctivos necesarios
Indicaciones pasó a paso: Después de realizar la lectura de la unidad 03 del módulo y de tener a mano y diligenciada todas las plantillas guía para la resolución de problemas, codifique cada uno de esos ejercicios. Realice los problemas propuestos al final de cada uno de los capítulos.

Plantee las dudas que surjan en el proceso en el foro Inquietudes.

Momento Individual:

Ordene todas las plantillas guía para resolución de problemas de cada uno de los ejercicios de las unidades anteriores.

- Bajar, instalar y configure el software javaSE
- Bajar, instalar y configure el IDE NewtBeans.

Momento de Socialización:

Coserte con su grupo horarios de encuentro para que realicen los comparativos sobre las soluciones que cada uno propuso a los ejercicios y selección la que consideren con mayor eficiencia y codifíquela el artefacto de software para cada uno de ellos.

Construcción Colectiva:

El equipo enviará el trabajo con los ejercicios resueltos anexando una conclusión referente a las estrategias utilizadas en el desarrollo del mismo y las ventajas que estas representan.

Actividades previas

Realización de todas las actividades de las unidades anteriores (01 y 02)

Producto a entregar

Carpeta para cada uno de los artefactos de software desarrollados.
Debe contener el código fuente del artefacto

Fecha límite novena semana

Porcentaje de valoración 10%

Actividades de Acompañamiento tutorial

Actividad N°1 Presentación de la Unidad II

Nombre del recurso a utilizar. Video conferencia

Competencias o indicador de competencias.

Interactuar (con los participantes, tutor, docente y contenidos) en el aula virtual a través de los recursos y actividades para alcanzar las competencias propuestas.

Indicaciones pasó a paso: En el chat se realiza la presentación de los contenidos y actividades de la Unidad de Aprendizaje III.

En este encuentro se presentan los aspectos a tener en cuenta para el uso de lenguajes de programación.

Temas a tratar:

- Características fundamentales de la programación orientada a objetos
- Sintaxis e Instrucciones del lenguaje de programación JAVA
- Procesos de Edición, Compilación y ejecución de programas
- Manuales del programador y del usuario

Fecha séptima semana

Porcentaje de valoración 10%

Actividad N° 4

Nombre de la actividad Retroalimentación de la Unidad II

Competencias

3. Demostrar interés por la actualización permanente a través del acceso a diferentes fuentes de información.

4. Cumplir con el cronograma de trabajo estableciendo su propia dinámica de aprendizaje

Nombre del recurso a utilizar. Video conferencia

Indicaciones pasó a paso:

La video conferencia tiene como fin realizar la retroalimentación de los temas desarrollados en la Unidad III y socializar aspectos de los trabajos individuales y en equipo.

En este encuentro se discute sobre los aspectos a tener en cuenta en el uso de los lenguajes de programación.

Los estudiantes socializan los trabajos realizados

Las pregunta orientadoras se planean a partir de la identificación de las necesidades que surjan en el desarrollo de la unidad I

Fecha Límite octava semana

Porcentaje de valoración 5%

Evaluación de los aprendizajes Unidad 3

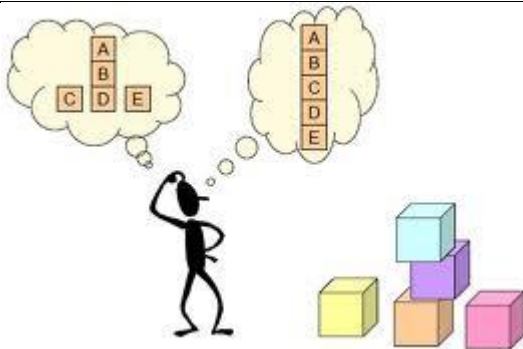
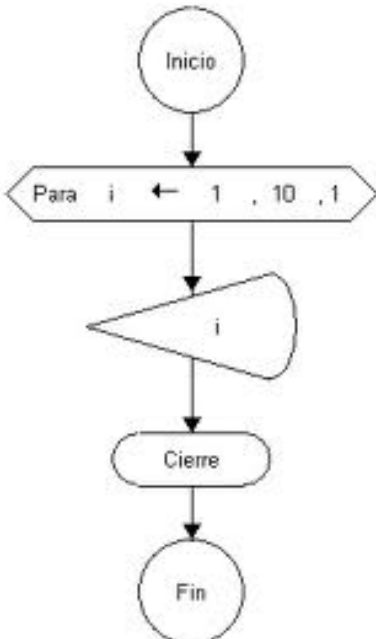
Resultados del aprendizaje

- Representa pseudocodigos a lenguajes de programación
- Documenta artefactos de software
- Ejecuta pruebas y realiza correcciones a código fuente.

Actividad	Competencias	Criterios	%
Actividades autónomas			
Actividad Nº 2 <i>Mapa mental lenguajes de programación</i>	<ul style="list-style-type: none"> • Identificar las características principales de un lenguaje de programación y su función en el desarrollo de programas 	<ul style="list-style-type: none"> • Define los elementos básico de un lenguaje de programación • Reconoce las estructuras de los lenguajes de programación. 	5%
Actividades colaborativas			
Actividad Nº 3 <i>Ejercicios de aplicación</i>	<ul style="list-style-type: none"> • Escribir el pseudocódigo que permita representar algoritmos propuestos para la solución del problema • Codificar artefactos de software empleando el paradigma orientado a objetos y el lenguaje de programación definido • Documentar artefactos de software y elaborar sus respectivos manuales de acuerdo con el paradigma empleado 	<ul style="list-style-type: none"> • Escribe el diagrama de flujo en el pseudocódigo correspondiente • Utiliza la estructura lógica del pseudocódigo • Escribe el código fuente correspondiente al pseudocódigo. • Verifica la funcionalidad producto de la codificación del artefacto de software. • Elabora la documentación técnica del artefacto codificado • Escribe los comentarios necesarios para documentar el código fuente 	10%
Actividades de acompañamiento tutorial			

Actividad N°1 Presentación de la unidad III	<ul style="list-style-type: none"> • Interactuar (con los participantes, tutor, docente y contenidos) en el aula virtual a través de los recursos y actividades para alcanzar las competencias propuestas. 	<ul style="list-style-type: none"> • Participa en los encuentros síncronos definidos en el módulo • Interactúa con los compañeros a través de medios síncronos y asíncronos • Accede en forma permanente al aula virtual para consultar los contenidos • Utiliza de los recursos tecnológicos necesarios para acceder a los contenidos y realizar las actividades 	5%
Actividad N° 4 Retroalimentación Unidad III	<ul style="list-style-type: none"> • Demostrar interés por la actualización permanente a través del acceso a diferentes fuentes de información. • Cumplir con el cronograma de trabajo estableciendo su propia dinámica de aprendizaje 	<ul style="list-style-type: none"> • Participa en los encuentros síncronos definidos en el módulo • Interactúa con los compañeros a través de medios síncronos y asíncronos • Accede en forma permanente al aula virtual para consultar los contenidos • Utiliza de los recursos tecnológicos necesarios para acceder a los contenidos y realizar las actividades. 	10 %

IMÁGENES:

<p>Capítulo 1</p>	
<p>Capítulo 2</p>	 <pre> graph TD Inicio((Inicio)) --> Para[/Para i ← 1, 10, 1/] Para --> i((i)) i --> Cierre([Cierre]) Cierre --> Fin((Fin)) </pre>
<p>Capítulo 3</p>	