

Paquete didáctico: Programación

Especialización en Docencia

José Luis Contreras Carrillo
TRABAJO TERMINAL

Contenido

Introducción	1
Descripción sintética de la asignatura	4
Competencia de la asignatura	4
Competencias genéricas:	4
Competencias complementarias:	5
Competencias profesionales:	5
Contenido del curso	6
Contenido del paquete.....	6
Cuadro Sintético del Curso	8
Planeación didáctica.....	11
Guía de uso	21
Relación de actividades de aprendizaje.....	23
Relación de presentaciones electrónicas	24
Relación de lecturas	25
Observaciones para el instructor	25
Planes de sesión	26

Introducción

Programación es la primera asignatura del área ocupacional “Desarrollo tecnológico”, promueve el razonamiento como una actividad mental que permite lograr una estructura y organización en las ideas, lo que permite llegar a conclusiones fundamentales al momento de emitir opiniones o resolver problemas. En esta asignatura, el estudiante desarrollará el razonamiento lógico mediante la construcción de algoritmos y resolución de problemas, apoyados de diagramas de flujo de información, así como también de ejercicios que pongan en práctica la elaboración de pseudocódigos y estructuras de control. Si el alumno ejercita y aprende el lenguaje de programación está mejor preparado para la toma de decisiones, pues al comprender el ciclo para la programación hace suyo los elementos para organizar y estructurar sus ideas.

La asignatura de programación se centra en el aprendizaje de algoritmos. Un algoritmo es una secuencia lógica de pasos o instrucciones, bien definidas, ordenadas y finitas, que permiten llevar a cabo una tarea específica. Si bien es cierto que los algoritmos son la base de la programación y son indispensable para poder desarrollar software, también lo es que su uso en la vida cotidiana fundamental y que en la mayoría de las ocasiones se utilizan de manera inconsciente. Los algoritmos están presentes en cada momento, desde el instante en el que un individuo se despierta por las mañanas, hasta el último segundo antes de quedarse dormido por las noches.

El colegio en el cual imparto clases, está incorporado a la Universidad Autónoma de Yucatán, tomando en cuenta el nuevo plan de estudios establecido por esta, el colegio ha ofertado a sus estudiantes el área ocupacional de “Desarrollo tecnológico”. Como ya se ha mencionado, Programación es la primera asignatura de esta área ocupacional y al ser reciente en la oferta educativa, aún no se ha desarrollado material didáctico que sirva de apoyo tanto para el docente como para los estudiantes en el proceso de desarrollo de competencias.

La elaboración de este trabajo terminal persigue el objetivo de proporcionar un manual estructurado y organizado de tal manera que sirva como apoyo didáctico; tanto para el catedrático como para el estudiante de la asignatura de programación. Este manual podrá ser

utilizado como guía durante el proceso de enseñanza-aprendizaje, facilitará a los estudiantes el proceso para obtener un aprendizaje significativo y desarrollar las competencias establecidas.

La importancia de este proyecto radica en que como se ha mencionado, el punto de partida de la programación es el diseño de algoritmos, el área ocupacional de “Desarrollo tecnológico” incluye en semestres posteriores las asignaturas de bases de datos, robótica y redes. En estas asignaturas que completan el programa de estudios del área ocupacional, son importantes las habilidades que se tengan en el desarrollo de algoritmos, ya que en las tres se aplican de una u otra manera. Si estas habilidades no son desarrolladas satisfactoriamente en la asignatura de programación, el estudiante podría presentar posteriormente algún tipo de necesidad educativa especial como, por ejemplo, habilidades deficientes acumuladas.

Manual de operaciones

Descripción sintética de la asignatura

El propósito principal de la asignatura es que el estudiante desarrolle el razonamiento lógico mediante la construcción de algoritmos y resolución de problemas, apoyados de diagramas de flujo de información, así como también de ejercicios que pongan en práctica la elaboración de pseudocódigos y estructuras de control.

Está diseñada para los alumnos del tercer semestre de bachillerato. La duración del curso es de 12 semanas, que son equivalentes a 24 sesiones de 2 módulos de 45 minutos cada una, con un total de 48 módulos de 45 minutos.

Competencia de la asignatura

Soluciona problemas de su contexto mediante la construcción de algoritmos, de diagramas de flujo, de pseudocódigos y estructuras de control, utilizando el razonamiento lógico, la simbología estándar y las reglas de sintaxis.

Competencias genéricas:

- Se expresa en español en forma oral y escrita en diversas situaciones comunicativas, utilizando correctamente el idioma.
- Usa las TIC en diversos contextos, de manera pertinente y responsable.
- Aplica los conocimientos de acuerdo al contexto y requerimientos de la situación, con pertinencia.
- Trabaja de manera cooperativa con otros en ambientes multi, inter y transdisciplinarios.
- Toma decisiones de índole académica y personal de manera eficaz y eficiente.
- Manifiesto compromiso con la calidad y la mejora continua en su desempeño académico y en su vida personal de manera responsable.
- Trabaja diversas tareas en el ámbito académico y personal de manera eficaz y eficiente.

Competencias complementarias:

Básicas

- CDB 1. Utiliza las habilidades de comunicación oral y escrita en español, en lengua materna y en lengua extranjera ante los diversos y cambiantes contextos (local, nacional e internacional) con pertinencia, coherencia y creatividad apoyado con herramientas tecnológicas.
- CDB 3. Utiliza las tecnologías de la información y comunicación con pertinencia, para su beneficio en la optimización de las actividades en los contextos personal, escolar, familiar, respetando la integridad de sus interlocutores.

Propedéuticas

- CDPC 1. Organiza la información a través de diversos procesos estructurales que le permiten obtener datos comprobados y útiles para su desempeño escolar, personal y familiar.
- CDPC 3. Evalúa los procesos de recolección, tratamiento y transmisión de la información de acuerdo con situaciones de la vida real.
- CDP 4. Desarrolla métodos para la organización de datos para compartir información con criterio científico y formal.
- CDPC 7. Utiliza los medios y recursos tecnológicos de forma activa y efectiva como herramientas que le permitan mejorar en todos los ámbitos en que se desempeña.

Competencias profesionales:

- CPDT Soluciona problemas de su contexto, mediante el desarrollo de productos y procesos innovadores sobre dispositivos automatizados, modelos de datos, redes y comunicaciones.

Contenido del curso

Tema	Sesiones	Horas
Algoritmos y Planteamiento de problemas	10	20
Diagramas de flujo y Pseudocódigo	8	16
Estructuras de control	6	12

Contenido del paquete

El presente paquete didáctico proporcionará los recursos y los materiales adecuados que permitan a los estudiantes adquirir nuevas habilidades y desarrollar las competencias que la asignatura plantea. Este estará dividido en los siguientes apartados.

- **Programa Sintético**

Presenta de forma breve, concisa y esquemática, el programa del curso propuesto en este paquete didáctico. Concentra las unidades, las competencias, las sesiones, las actividades de aprendizaje, las presentaciones electrónicas y las lecturas que se incluyen; con la finalidad de estructurar tanto la aplicación de las actividades como el uso de los materiales.

- **Planeación didáctica**

Despliega el desglose de cada período con su respectiva competencia, la secuencia de contenidos y su desagregado, las estrategias de enseñanza y aprendizaje, la descripción general de las actividades de aprendizaje, las horas de dichas actividades, así como los niveles de dominio que adquirirá el estudiante en el curso.

- **Planes de sesión**

Definen las actividades y tareas a realizar en cada sesión de clase. En cada plan de sesión se especifica la unidad a la que pertenece la sesión, el tema a abordar, el objetivo y la

duración de la sesión, los materiales y recursos a utilizar, la estrategia a seguir detallando el tiempo estimado para cada actividad y la bibliografía.

- **Actividades de Aprendizaje**

Cada actividad de aprendizaje describe como se llevarán a cabo las tareas asignadas a los estudiantes durante la sesión. En ellas se especifica el nombre de la actividad de aprendizaje, las instrucciones para llevarla a cabo, el formato y la forma de entrega, así como la nomenclatura deseada.

- **Instrumento de evaluación**

Las actividades integradoras incluyen un instrumento de evaluación que permitirá conocer el nivel de dominio que desarrolló el estudiante con respecto a los temas abordados.

- **Presentaciones**

Cuando la sesión así lo requiere, se incluye una presentación electrónica que puede utilizar el profesor para abordar un determinado tema durante la sesión, en ella se sintetizan los contenidos de forma que el estudiante pueda comprenderlos.

- **Lecturas**

Cuando la sesión así lo requiere, se incluyen una o más lecturas que permiten llevar a cabo las actividades de aprendizaje.

Cuadro Sintético del Curso

A continuación, se presenta, de forma breve, concisa y esquemática, el programa del curso propuesto en este paquete didáctico, con la finalidad de estructurar tanto la aplicación de las actividades como el uso de materiales.

	Sesión	Tema	Actividad de aprendizaje	Presentación	Lectura
Período I	1	Introducción a la programación	ADA 01 – Examen de diagnóstico ADA 02 - Tipos de software	Presentación 01 – Introducción a la asignatura	Lectura 01 – Tipos de Software
	2	Metodología en la solución de problemas	ADA 03 – Diario de aprendizaje ADA 04 – Resolvamos un problema		Lectura 02 – Solución de problemas Lectura 03 – Solución de problemas y programación
	3	Metodología en la solución de problemas	ADA 05 – Identificando problemas		
	4	Integración de habilidades	ADA 06 – Metodología de solución de problemas		
	5	Introducción a los algoritmos	ADA 07 – Kahoot ADA 08 – Cuadro sinóptico		Lectura 04 – Algoritmos en programación
	6	Algoritmos en la vida cotidiana	ADA 09 – El avión más veloz		

			ADA 10 – Creando algoritmos		
	7	Diseño de algoritmos	ADA 11 – Pensamientos algorítmico y procedimental		
	8	Integración de habilidades	ADA 12 – Construyendo algoritmos ADA 13 – Diario reflexivo		
Período II	9	Diseño modular	ADA 14 – Diseño modular y refinamiento sucesivo ADA 15 – Diario reflexivo		Lectura 05 – Diseño modular y refinamiento sucesivo
	10	Diseño modular y refinamiento sucesivo	ADA 16 – Aprestamiento ADA 17 – Siguiendo algoritmos		
	11	Integración de habilidades	ADA 18 – Solución de problemas, con algoritmos		
	12	Conceptos básicos	ADA 19 – Conceptos básico		
	13	Conceptos básicos	ADA 20 – Exposición de conceptos básicos	Presentación 02 – Conceptos básicos	
	14	Operadores y operandos	ADA 21 – Operadores y operandos		
	15	Integración de habilidades	ADA 22 – Exposición de operadores y operandos	Presentación 03 – Operadores y operandos	

	16	Asignaciones y expresiones	ADA 23 – Asignaciones y expresiones ADA 24 – Diagramas de Flujo y Pseudocódigo	Presentación 04 – Asignaciones y expresiones	
	17	Integración de habilidades	ADA 25 – Pseudocódigo ADA 26 – Solución de problemas utilizando pseudocódigo		
Período III	18	Diseño de diagramas de flujo	ADA 27 – Diagramas de flujo		
	19	Integración de habilidades	ADA 28 – Solución de problemas utilizando diagramas de flujo		
	20	Estructuras secuenciales y selectivas	ADA 29 – Estructuras secuenciales y selectivas		
	21	Estructuras repetitivas	ADA 30 – Estructuras repetitivas o cíclicas		
	22	Integración de habilidades	ADA 31 – Estructuras de control repetitivas		
	23	Proyecto integrador	ADA 32 – Proyecto Semestral		
	24	Proyecto integrador	ADA 32 – Proyecto Semestral		

Planeación didáctica

DATOS GENERALES DE IDENTIFICACIÓN	
Nombre de la asignatura	Programación
Tipo	Optativa
Ubicación	Tercer semestre
Duración total en horas	48
Requisitos académicos previos	Ninguno

COMPETENCIA DE LA ASIGNATURA

Soluciona problemas de su contexto mediante la construcción de algoritmos, de diagramas de flujo, de pseudocódigos y estructuras de control, utilizando el razonamiento lógico, la simbología estándar y las reglas de sintaxis.

CONTEXTUALIZACIÓN

Programación es la primera asignatura del área ocupacional Desarrollo Tecnológico, promueve el razonamiento como una actividad mental que permite lograr una estructura y organización en las ideas, lo que permite llegar a conclusiones fundamentales al momento de emitir opiniones o resolver problemas. En esta asignatura, el estudiante desarrollará el razonamiento lógico mediante la construcción de algoritmos y resolución de problemas, apoyados de diagramas de flujo de información, así como también de ejercicios que pongan en práctica la elaboración de pseudocódigos y estructuras de control. Si el alumno ejercita y aprende el lenguaje de programación está mejor preparado para la toma de decisiones, pues al comprender el ciclo para la programación hace suyo los elementos para organizar y estructurar sus ideas. Se relaciona con las asignaturas: Las Matemáticas en mi entorno, matemáticas, pensamiento lógico y crítico.

Unidad I: Algoritmos

Competencia de la Unidad I:		Desarrolla algoritmos, de manera creativa y reflexiva, para utilizarlo de apoyo para resolver problemas del ámbito escolar y de la vida diaria.				
Resultados de aprendizaje	Contenidos	Estrategias de enseñanza y aprendizaje	Actividades de aprendizaje		Evidencias de aprendizaje	Puntaje
			Descripción	Duración		
Identifica las alternativas de solución de problemas del ámbito escolar y cotidiano, teniendo en cuenta las partes que lo conforman.	<p>Solución de problemas:</p> <ul style="list-style-type: none"> • Analizar el problema. • Identificar datos disponibles (de entrada) • Precisar los resultados esperados (datos de salida) • Establecer estrategias o metodologías de solución. 	Aprendizaje cooperativo	<p>ADA: En equipo de 3 personas resuelven la situación planteada, utilizando la metodología de solución de problemas.</p> <p>Recursos y materiales</p> <ul style="list-style-type: none"> • PC – Word • Internet • Plataforma institucional 	8	<p>Evidencia</p> <p>Documento de redacción y solución de problemas.</p> <p>Criterio de evaluación</p> <p>Redacta un problema detallando los datos de entrada y lo que se espera obtener con la solución del mismo.</p> <p>Plantea la solución del problema apoyándose en las estrategias y metodologías abordadas.</p>	8

<p>Utiliza algoritmos para la solución de problemas del ámbito escolar y cotidiano, considerando los pasos para su elaboración.</p>	<p>¿Qué es un algoritmo? Pensamiento algorítmico.</p>	<p>Mapa cognitivo y Diario reflexivo</p>	<p>ADA: De manera individual crea algoritmos para resolver los escenarios planteados, utilizando las secciones establecidas para el desarrollo del mismo.</p> <p>Recursos y materiales</p> <ul style="list-style-type: none"> • PC – Word • Internet • Plataforma institucional 	<p>8</p>	<p>Evidencia</p> <p>Algoritmos creados para la solución de los problemas.</p> <p>Criterio de evaluación</p> <p>Crea soluciones coherentes cumpliendo con las características que debe poseer un algoritmo.</p> <p>Se apoya en los datos de entrada para realizar el o los procesos que permitan obtener los datos de salida esperados.</p>	<p>12</p>
Total				16	Total	20

Unidad II: Diagramas de flujo y pseudocódigo

Competencia de la Unidad II:		Diseña diagramas de flujo utilizando algoritmos que sean efectivos en la resolución de problemas de forma clara y lógica (ordenada).				
Resultados de aprendizaje	Contenidos	Estrategias de enseñanza y aprendizaje	Actividades de aprendizaje		Evidencias de aprendizaje	Puntaje
			Descripción	Duración		
Utiliza algoritmos para la solución de problemas de programación, considerando las características de un buen algoritmo para su elaboración.	<p>Algoritmos:</p> <ul style="list-style-type: none"> • Características • Partes • Clasificación <p>Establecer estrategias o metodologías de solución.</p>	Aprendizaje cooperativo	<p>ADA: En equipo de 3 personas resuelven la situación planteada, integrando la metodología de solución de problemas con el diseño de algoritmos.</p> <p>Recursos y materiales</p> <ul style="list-style-type: none"> • PC – Word • Internet <p>Plataforma institucional</p>	8	<p>Evidencia</p> <p>Documento de redacción y solución de problemas.</p> <p>Criterio de evaluación</p> <p>Redacta un problema detallando los datos de entrada y lo que se espera obtener con la solución del mismo.</p> <p>Diseña un algoritmo para la solución del problema tomando en cuenta las características de los mismos.</p>	8

Describe los diferentes tipos de datos y operadores utilizados en programación, para resolver problemas cotidianos.	<p>Tipos de datos:</p> <ul style="list-style-type: none"> • Numéricos • Alfanuméricos • Booleanos <p>Identificadores</p> <ul style="list-style-type: none"> • Constantes • Variables <p>Operadores y operandos</p>	Cuadro sinóptico	<p>ADA: En equipos de 3 integrantes elabora una presentación, describiendo los tipos de datos, identificadores y operadores abordados</p> <p>Recursos y materiales</p> <ul style="list-style-type: none"> • PC – Word • Internet • Plataforma institucional 	5	<p>Evidencia</p> <p>Presentación y Exposición.</p> <p>Criterio de evaluación</p> <p>Incluye los diferentes tipos de datos, identificadores y operadores.</p>	8
Utiliza pseudocódigo para representar algoritmos y diagramas de flujo de problemas cotidianos, considerando las reglas básicas para su elaboración.	<p>El pseudocódigo</p> <p>Reglas para la elaboración de pseudocódigo.</p> <p>Conceptos básicos de programación</p>	Mapa cognitivo de algoritmo	<p>ADA: De manera individual desarrolla pseudocódigo, considerando las reglas básicas para su elaboración</p> <p>Recursos y materiales</p> <ul style="list-style-type: none"> • PC – Word • Internet • Plataforma institucional 	5	<p>Evidencia</p> <p>Documento con pseudocódigo.</p> <p>Criterio de evaluación</p> <p>Representa con pseudocódigo los algoritmos y diagramas de flujo considerando las reglas básicas para su elaboración.</p>	6
Total				16	Total	20

Unidad III: Estructuras de control

Competencia de la Unidad III:		Soluciona problemas de su contexto mediante la construcción de estructuras de control, utilizando el razonamiento lógico, la simbología estándar y las reglas de sintaxis.				
Resultados de aprendizaje	Contenidos	Estrategias de enseñanza y aprendizaje	Actividades de aprendizaje		Evidencias de aprendizaje	Puntaje
			Descripción	Duración		
Utiliza diagramas de flujo para representar algoritmos de problemas cotidianos, considerando los símbolos estandarizados para su elaboración.	Representación gráfica de algoritmos. Simbología de los diagramas de flujo. Reglas para la elaboración de diagramas de flujo. Conceptos básicos de programación	Diagrama de flujo	ADA: De manera individual desarrolla diagramas de flujo, utilizando los símbolos establecidos Recursos y materiales <ul style="list-style-type: none"> • PC – Word • Internet Plataforma institucional	5	Evidencia Documento con diagramas de flujo. Criterio de evaluación Representa los algoritmos utilizando los símbolos establecidos.	6

<p>Emplea estructuras de selección para desarrollar soluciones a diversos problemas, de forma lógica, ordenada y eficiente</p>	<p>Estructuras de selección</p>	<p>Simulación</p>	<p>ADA: De manera individual emplea estructuras de selección para resolver los planteamientos, de forma lógica, ordenada y eficiente.</p> <p>Recursos y materiales</p> <ul style="list-style-type: none"> • PC – Word • Internet <p>Plataforma institucional</p>	<p>6</p>	<p>Evidencia</p> <p>Solución a problemas mediante estructuras de selección.</p> <p>Criterio de evaluación</p> <p>Emplea diferentes estructuras de selección de acuerdo al problema presentado.</p>	<p>6</p>
<p>Emplea estructuras de control para desarrollar soluciones a diversos problemas, de forma lógica, ordenada y eficiente</p>	<p>Estructuras de control</p>	<p>Simulación</p>	<p>ADA: De manera individual emplea estructuras de control para resolver los planteamientos, de forma lógica, ordenada y eficiente.</p> <p>Recursos y materiales</p> <ul style="list-style-type: none"> • PC – Word • Internet <p>Plataforma institucional</p>	<p>6</p>	<p>Evidencia</p> <p>Solución a problemas mediante estructuras de control.</p> <p>Criterio de evaluación</p> <p>Emplea diferentes estructuras de control de acuerdo al problema presentado.</p>	<p>8</p>
Total				<p>16</p>	Total	<p>20</p>

DESCRIPCIÓN DE LOS NIVELES DE DOMINIO

Puntaje	Categoría	Descripción
90 – 100	Sobresaliente (SS)	Soluciona los problemas que se le presentan mediante la construcción de algoritmos, diagramas de flujo, pseudocódigos y estructuras de control, utilizando el razonamiento lógico y la toma de decisiones, respetando la simbología estándar y cumpliendo con las reglas de sintaxis, empleando todos los recursos que se proporcionan y compartiendo estos en la plataforma.
80 – 89	Satisfactorio (SA)	Soluciona la mayoría de los problemas que se le presentan mediante la construcción de algoritmos, diagramas de flujo, pseudocódigos y estructuras de control, utilizando el razonamiento lógico y la toma de decisiones, respetando la simbología estándar y cumpliendo con las reglas de sintaxis, empleando la mayoría de los recursos que se proporcionan.
70 – 79	Suficiente (S)	Soluciona algunos de los problemas que se le presentan mediante la construcción de algoritmos, diagramas de flujo, pseudocódigos y estructuras de control, utilizando el razonamiento lógico y la toma de decisiones, respetando la simbología estándar y cumpliendo con las reglas de sintaxis, empleando algunos de los recursos que se proporcionan y compartiendo estos en la plataforma.
0 – 69	No acreditado (NA)	No logra solucionar los problemas que se le presentan mediante la construcción de algoritmos, diagramas de flujo, pseudocódigos y estructuras de control, utilizando el razonamiento lógico y la toma de decisiones, respetando la simbología estándar y cumpliendo con las reglas de sintaxis, empleando algunos de los recursos que se proporcionan y compartiendo estos en la plataforma.

Acreditación de la asignatura	
Evaluación de proceso	60 %
Evaluación de producto	40 %
Total	100 %

Evaluación de proceso		
Evidencias de aprendizaje	Criterios de evaluación	Porcentaje
Documento de redacción y solución de problemas	Redacta y soluciona un problema detallando los datos de entrada, salida y apoyándose en las estrategias y metodologías abordadas.	8
Algoritmos creados para la solución de los problemas	Crea soluciones lógicas, ordenadas y eficientes, cumpliendo con las características que debe poseer un algoritmo.	12
Cuadro sinóptico	Incluye los diferentes tipos de datos, identificadores y operadores.	6
Documento con diagramas de flujo	Representa correctamente los algoritmos utilizando los símbolos establecidos.	8
Documento con pseudocódigo	Representa con pseudocódigo los algoritmos y diagramas de flujo considerando las reglas básicas para su elaboración.	6
Solución a problemas mediante estructuras de selección.	Emplea las diferentes estructuras selección de acuerdo al problema presentado	8
Solución a problemas mediante estructuras de control.	Emplea las diferentes estructuras de control de acuerdo al problema presentado	12
	Total	60

Evaluación de producto		
Evidencias de aprendizaje	Criterios de evaluación	Porcentaje
Desarrollo de proyecto	Desarrolla algoritmos, de manera creativa y reflexiva, para utilizarlo de apoyo para resolver el problema que se le presenta.	40
	Diseña diagramas de flujo utilizando algoritmos que sean efectivos en la resolución de problemas de forma clara y lógica (ordenada).	
	Soluciona problemas de su contexto mediante la construcción de estructuras de control, utilizando el razonamiento lógico, la simbología estándar y las reglas de sintaxis.	
	Total	40

Guía de uso

Antes de utilizar este paquete didáctico, el docente debe asegurarse de revisar los contenidos del programa de estudios y si lo amerita actualizar las actividades y recursos de este paquete. Al estar relacionada la programación con la tecnología, es probable que los planes de estudio se actualicen con mayor frecuencia que otras asignaturas, para adaptarlos a situaciones actuales.

El apartado de descripción sintética de la asignatura, detalla las competencias establecidas en el programa de estudios, mismas que se pretenden lograr al concluir el semestre y después de utilizar este paquete didáctico. Así mismo, se detallan los temas que se abordarán en el semestre y el número de sesiones y horas dedicados a cada tema.

El presente paquete didáctico se compone de 24 sesiones de 90 minutos cada una. En el programa sintético, se presenta un compendio de cada una de estas 24 sesiones, detallando el tema a abordar en cada una, las actividades a resolver y los recursos y materiales didácticos a utilizar durante la misma. Se recomienda utilizar este apartado para tener un panorama general de cómo se abordará el contenido de la asignatura y para evaluar si requiero o no adaptarlo a su contexto y a sus estudiantes.

Al inicio del semestre, se recomienda presentar al estudiante la descripción de la asignatura y el plan sintético, así como los criterios de evaluación y el proyecto que entregarán al final, que deben desarrollar gradualmente conforme transcurren las sesiones de clase. Así como también, se recomienda proporcionarles los planes de sesión, actividades de aprendizaje y lecturas, en formato digital y/o impreso.

Los planes de sesión describen las actividades a realizar en cada una de las 24 sesiones que tendrán lugar durante el transcurso del semestre, para la asignatura. Cada plan especifica la unidad a la que pertenece la sesión, el tema a abordar, el objetivo y un resumen de la sesión, la duración, las actividades, materiales y recursos a utilizar, la estrategia a seguir y la bibliografía.

El documento está organizado por sesiones de clase, iniciando en la sesión uno y concluyendo en la 24, después de cada plan de sesión se incluyen las actividades de aprendizaje de dicha sesión, las presentaciones electrónicas, las lecturas y los instrumentos de evaluación, en caso de existir. Cabe aclarar que el instructor tiene la libertad de realizar modificaciones de acuerdo con las necesidades de sus estudiantes y de su contexto.

Relación de actividades de aprendizaje

En el siguiente cuadro se enlista las actividades de aprendizaje programadas para el curso:

	Actividad de aprendizaje	Sesión
Unidad I	ADA 01 – Examen de diagnóstico	1
	ADA 02 - Tipos de software	1
	ADA 03 – Diario de aprendizaje	2
	ADA 04 – Resolvamos un problema	2
	ADA 05 – Identificando problemas	3
	ADA 06 – Integración de habilidades	4
	ADA 07 – Kahoot	5
	ADA 08 – Cuadro sinóptico	5
	ADA 09 – El avión más veloz	6
	ADA 10 – Creando algoritmos	6
	ADA 11 – Pensamientos algorítmico y procedimental	7
	ADA 12 – Construyendo algoritmos	8
	ADA 13 – Diario reflexivo	8
Unidad II	ADA 14 – Diseño modular y refinamiento sucesivo	9
	ADA 15 – Diario reflexivo	9
	ADA 16 – Aprestamiento	10
	ADA 17 – Siguiendo algoritmos	10
	ADA 18 – Integración de habilidades	11
	ADA 19 – Conceptos básico	12
	ADA 20 – Exposición de conceptos básicos	13
	ADA 21 – Operadores y operandos	14
	ADA 22 – Exposición de operadores y operandos	15
	ADA 23 – Asignaciones y expresiones	16
	ADA 24 – Diagramas de Flujo y Pseudocódigo	16

Unidad III	ADA 25 – Pseudocódigo	17
	ADA 26 – Solución de problemas utilizando pseudocódigo	17
	ADA 27 – Diagramas de flujo	18
	ADA 28 – Solución de problemas utilizando diagramas de flujo	19
	ADA 29 – Estructuras secuenciales y selectivas	20
	ADA 30 – Estructuras repetitivas o cíclicas	21
	ADA 31 – Estructuras de control repetitivas	22
	ADA 32 – Proyecto Semestral	23
	ADA 32 – Proyecto Semestral	24

Relación de presentaciones electrónicas

Las presentaciones incluidas en este paquete servirán de apoyo al instructor para la exposición de los temas respectivos.

Número	Título	Sesión
1	Introducción a la asignatura	1
2	Conceptos básicos	13
3	Operadores y operandos	15
4	Asignaciones y expresiones	16

Relación de lecturas

El estudiante puede apoyarse en las lecturas incluidas en el presente paquete didáctico para realizar las actividades. Se recomienda que complemente con la bibliografía sugerida en cada sesión y con material de Internet. También se recomienda que estos materiales sean revisados por el instructor antes de iniciar el curso con el fin de actualizar los que sean necesarios. A continuación, se detalla los materiales de lectura y las sesiones donde se utilizarán.

Número	Título	Sesión
1	Tipos de Software	1
2	Solución de problemas	2
3	Solución de problemas y programación	2
4	Algoritmos en programación	5
5	Diseño modular y refinamiento sucesivo	9

Observaciones para el instructor

Aunque las actividades están diseñadas para que los estudiantes trabajen de manera autónoma, ya sea en equipo o individualmente, es importante que el docente supervise constantemente la realización de las mismas y las guíe, con el fin de lograr un aprendizaje significativo con cada una de ellas.

Al inicio del semestre se recomienda explicar el proyecto que se llevará a cabo para la evaluación de producto, establecer la metodología a seguir, los criterios de evaluación y subirlo a la plataforma educativa.

Aunque se utilizan algunas herramientas digitales para realizar las actividades presentadas, el docente puede incorporar otras más actuales o que considere que se adapten mejor a las mismas. Puede apoyarse en las herramientas que ofrece la plataforma educativa para llevar a cabo alguna actividad.

Es importante dar al estudiante la realimentación correspondiente a cada actividad en tiempo y forma, ya que las actividades permiten un aprendizaje gradual en el transcurso del semestre, si con alguna de ellas el estudiante no logra el objetivo planteado, podría dar lugar a la necesidad educativa especial conocida como deficiencia acumulada.

En cada actividad, proporcione a los estudiantes el o los materiales necesarios para llevarla a cabo y dé a conocer lo que se espera de ellos en la realización, con el fin de que estén consientes hacia donde deben dirigirse y como.

Al inicio del semestre establezca claramente el plan de trabajo a realizar, calendarice las actividades considerando los días que por uno u otro motivo se suspenderán clases, especifique la fecha de entrega de las actividades, así como de las sanciones cuando no se cumplan los plazos y tiempos acordados.

Planes de sesión

El presente paquete didáctico consta de veinticuatro planes de sesiones de clase, en los cuales se describe de manera efectiva, la forma en que puede ser utilizado cada componente del paquete didáctico en cada sesión. Cada plan contiene el número de sesión al que se refiere, la unidad temática, los temas a desarrollar, los objetivos de la clase, las estrategias de enseñanza-aprendizaje, cada actividad a realizar cuenta con el tiempo programado para su buen desarrollo, los recursos didácticos requeridos, la metodología y las referencias bibliográficas básicas. En este apartado se incluyen también las actividades de aprendizaje, las presentaciones electrónicas y las lecturas a utilizar en cada sesión.

Plan de Sesión

3º Semestre de Programación

Unidad 1: Algoritmos
Tema: Introducción a la programación
Duración: 90 minutos
Mtro. José Contreras

Primera sesión

Objetivo(s):

- Que el estudiante conozca las competencias, los contenidos, las estrategias de aprendizaje y las estrategias de evaluación, de la asignatura de programación.
- Que el estudiante identifique los diferentes tipos de software que existen en el mercado, como están constituidos y el proceso de codificación.

Resumen:

- Conocer las competencias, los contenidos, las estrategias de aprendizaje y las estrategias de evaluación, de la asignatura, es indispensable para saber hacia donde nos vamos a dirigir durante el desarrollo de la asignatura. Existen diferentes tipos de software y aunque cada uno de ellos tienen alguna función en particular, estos están creados de la misma manera, con instrucciones dadas en forma de líneas de código.

Estrategia:

- 25' Introducción a la asignatura, proyecto semestral y criterios de evaluación.
- 20' Evaluación diagnóstica – Actividad 01.
- 20' Cómo se crea el software y porque es importante la programación
- 20' Lectura 01 y Actividad 02.
- 5' Síntesis de la sesión.

Actividades de Aprendizaje:

- Realizar las actividades 01 y 02.

Recursos Didácticos:

- Proyector, pizarrón, plumones, internet, equipos de cómputo y/o Smartphone.

Referencias Bibliográficas:

- Elizondo Callejas, R. A. (2010). Informática 2. Serie integral por competencias. México: Patria.
- Joyanes Aguilar, L. (2001). Fundamentos de programación. Algoritmos, Estructuras de datos Objetos. (3ra. ed.). España: Mc Graw Hill.
- Aho, A. V., Hopcroft, J. E., Ullman, J. D. (1998). Estructuras de datos y algoritmos. México: Addison – Wesley Longman.

ADA 01 – Examen de diagnóstico

De manera individual responde a las siguientes preguntas, describe cada respuesta lo mejor que puedas.

1. ¿Qué es Hardware?

R=

2. ¿Qué es software?

R=

3. ¿Cómo se crea un software?

R=

4. ¿Qué es programación?

R=

5. ¿Qué entiendes por instrucciones?

R=

6. ¿Qué es un problema?

R=

7. ¿Cuáles son los pasos que sigues para resolver un problema?

R=

8. ¿Qué es un algoritmo?

R=

9. ¿Para qué sirve un algoritmo?

R=

10. ¿Qué es un diagrama y para qué sirve?

R=

11. ¿Qué entiendes por sintaxis?

R=

12. ¿Qué entiendes por variables?

R=

13. ¿Qué son las restricciones?

R=

14. ¿Cuáles son los operadores matemáticos y cuales los relacionales?

R=

15. ¿Qué tipos de datos conoces?

R=

Guarda tu documento en PDF y súbelo a la plataforma con la siguiente nomenclatura:

Apellidos_Nombre_ADA01 ej. RodriguezMedina_Alejandro_ADA01

ADA 02 - Tipos de software

Instrucciones: Escribe la definición de cada tipo de software en la segunda fila y arrastra a la tercera las imágenes de abajo según el recuadro que le corresponde. Guarda tu documento en PDF y súbelo a la plataforma con la siguiente nomenclatura: Apellidos_Nombre_ADA02

Software de sistema	Software de desarrollo	Software de aplicación



Presentación 01 – Introducción a la asignatura

PROGRAMACIÓN

TERCER SEMESTRE
ASIGNATURA OPTATIVA
48 HORAS PRESENCIALES

OBJETIVO DE LA SESIÓN

- Que el estudiante conozca los contenidos, los contextos, las estrategias de aprendizaje y las estrategias de evaluación de la asignatura de programación.

INTENCIONALIDAD FORMATIVA DE LA ASIGNATURA

- Es la primera asignatura del área ocupacional Desarrollo Tecnológico.
- En esta asignatura, desarrollará el razonamiento lógico mediante la construcción de algoritmos y resolución de problemas, creación de diagramas de flujo de información, así como creación de esquemas que pongan en práctica la elaboración de pseudocódigos y estructuras de control.
- Aprenderá y aplicará en lenguaje de programación estando mejor preparado para la toma de decisiones, para el conocimiento aplicado para la programación basándose en los elementos para organizar y controlar sus ideas.

COMPETENCIAS GENÉRICAS A DESARROLLAR

- Se expresa en español en forma oral y escrita en diversas situaciones comunicativas, utilizando correctamente el idioma.
- Utiliza las TIC en diversos contextos, de manera pertinente y responsable.
- Aplica los conocimientos de acuerdo al contexto y requerimientos de la situación, con pertinencia.
- Trabaja de manera cooperativa con otros en ambientes multi, inter y transdisciplinarios.
- Toma decisiones de índole académica y personal de manera eficaz y eficiente.
- Muestra compromiso con la calidad y la mejora continua en su desempeño académico y en su vida personal de manera responsable.
- Trabaja diversas tareas en el ámbito académico y personal de manera eficaz y eficiente.

COMPETENCIAS COMPLEMENTARIAS A DESARROLLAR

Básicas

- CDR 1. Utiliza las habilidades de comunicación oral y escrita en español, en lengua materna y en lengua extranjera ante los diversos contextos comunicativos locales, nacionales e internacionales con pertinencia, coherencia y capacidad de apoyo con herramientas tecnológicas.
- CDR 3. Utiliza las tecnologías de la información y comunicación con pertinencia, para su beneficio en la adquisición de las actividades en los contextos personal, escolar, familiar, respectando la integridad de sus interlocutores.

COMPETENCIAS COMPLEMENTARIAS A DESARROLLAR

Preprofesionales

- CDPC 3. Organiza la información a través de diversos procesos estructurales que le permitan diseñar listas comparativas y útiles para su desarrollo escolar, personal y familiar.
- CDPC 5. Evalúa las fuentes de información, tratamiento y transmisión de la información de acuerdo con situaciones de la vida real.
- CDPC 6. Desarrolla métodos para la organización de datos para compartir información con entorno científico y técnico.
- CDPC 7. Utiliza los medios y recursos tecnológicos de forma activa y efectiva como herramientas que le permitan mejorar en todos los ámbitos en que se desempeña.

COMPETENCIAS PROFESIONALES A DESARROLLAR

- CPDT Soluciona problemas de su contexto, mediante el desarrollo de productos y procesos innovadores sobre dispositivos automatizados, móviles de datos, redes y comunicaciones.

COMPETENCIA A DESARROLLAR EN LA ASIGNATURA

- Soluciona problemas de su contexto mediante la construcción de algoritmos, de diagramas de flujo, de pseudocódigos y estructuras de control, utilizando el razonamiento lógico, la sintaxis estándar y las reglas de sintaxis.

CONTENIDO

- Algoritmos
- Diagramas de flujo
- Pseudocódigo
- Estructuras de control

ESTRATEGIAS DE ENSEÑANZA Y APRENDIZAJE

- Prácticas operativas.
- Resolución de problemas y ejercicios
- Organizador gráfico
- Aprendizaje temático
- Aprendizaje orientado a proyectos

ESTRATEGIAS GENERALES DE EVALUACIÓN

Evaluación de procesos - 40%	Evaluación de productos - 60%
Prácticas operativas Evaluación de problemas Pruebas de desempeño	Desarrollo de proyectos

Lectura 01 – Tipos de Software

La computadora no puede realizar ninguna función por sí sola, necesita de instrucciones que le digan que debe de hacer y que operaciones necesita realizar. Estas instrucciones, agrupadas en forma de "**programas**" y que se cargan en la memoria de la computadora forman lo que se denomina **software**. Es la parte intangible del equipo de cómputo.

Haciendo una analogía con un televisor, el software serían todos los programas que podemos ver a través de la pantalla, como los noticieros, las películas o las novelas. Estos programas son intangibles, los vemos cuando el televisor está encendido, pero no podemos tocarlos.

Por lo tanto, el **software es un componente** que permite que la computadora pueda desempeñar **tareas inteligentes**; dirige en forma adecuada a los elementos físicos o hardware, es el software lo que indica al hardware **en qué secuencia y bajo qué lógica** hay que hacer los cálculos y las manipulaciones de datos que se introducen. **Sin el software, la computadora sería un conjunto de piezas inservibles.**

El software es cada uno de los programas que se utilizan para dirigir la operación de la computadora. Una computadora puede ser utilizada para hacer diferentes tareas. Entre muchas otras cosas, para:



De acuerdo a la tarea que vaya a realizar, entonces se utilizan diferentes tipos de software o programas. Muchas veces decimos programas o datos refiriéndonos a la misma cosa, pero no es así.

Programas



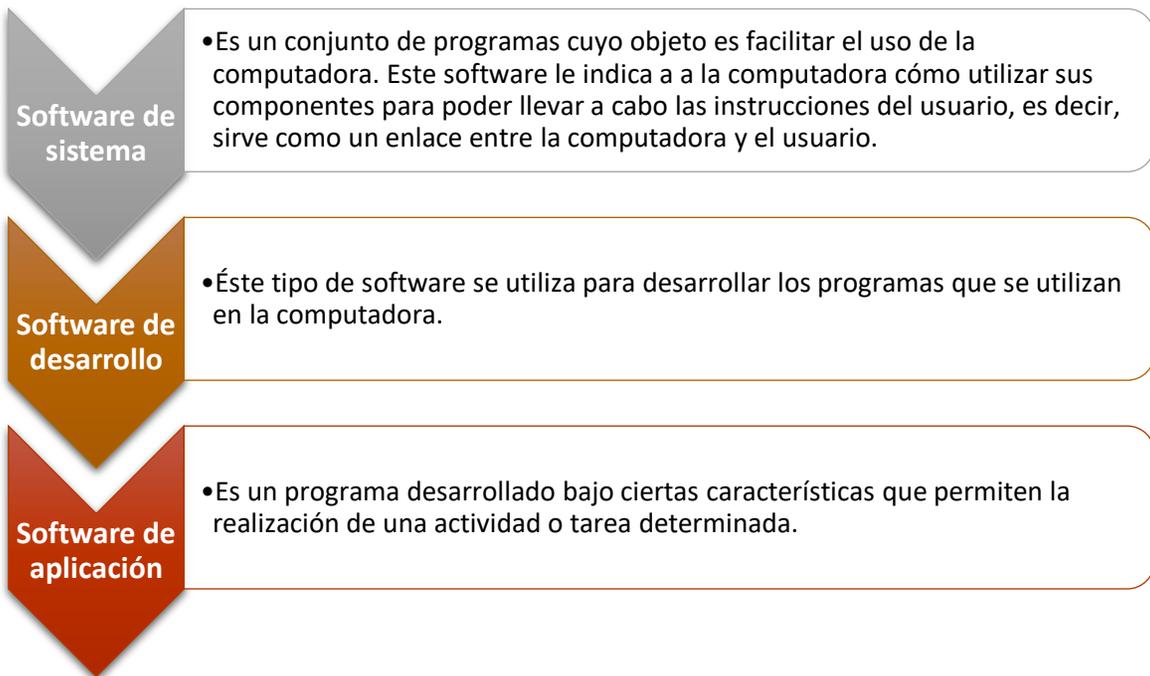
Los programas dan instrucciones para realizar tareas al hardware o sirven de conexión con otro software.

Datos



Los datos solamente existen para su uso eventual por un programa

El software se puede agrupar en tres grandes categorías:



➤ ¿Para qué es un software de sistema?

El software de sistema es indispensable para controlar las funciones básicas de la computadora, cuando ésta se enciende el sistema operativo se carga en la memoria y permanece ahí hasta que la computadora se apaga. El sistema operativo que tiene la computadora que estas utilizando se llama Windows.

Algunos ejemplos de sistemas operativos son **Microsoft Windows Vista, XP, 7, 8**, el sistema operativo de **Macintosh**, Unix, Linux, entre otros.



➤ **¿Para qué es un software de desarrollo?**

En un supermercado cada vez que un cajero nos cobra un producto, lo pasa por un lector de código que alimenta un programa que cobra productos, ese programa está realizado en un software de desarrollo bajo un cierto lenguaje de programación.

Algunos ejemplos de software de desarrollo son los lenguajes de programación C#, PHP, HTML, Visual Basic Net, entre otros.

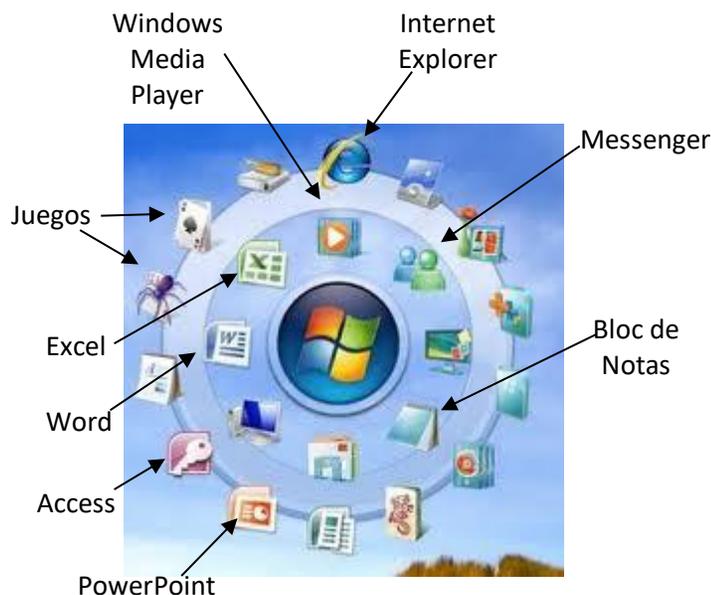


➤ **¿Para qué es un software de aplicación?**

Estos programas son empleados por el usuario para simplificar su vida, ya sea si decide escribir un libro, una tesis o un memorándum, o para poder manejar grandes volúmenes de datos; o aquellos programas que le sirvan para emitir una nómina de una empresa. Estos programas son los más solicitados por la mayoría de usuarios de computadoras debido a los beneficios o servicios que reciben de ellos.

El software de aplicación más difundido en el mercado se puede dividir en varias categorías, las principales son:

- **Procesadores de palabras:** Sirve para crear documentos
- **Hojas electrónicas de cálculo:** Sirve para realizar cálculos
- **Bases de datos:** Sirve para almacenar y organizar datos
- **Presentadores:** Sirve para realizar presentaciones gráficas
- **Comunicación:** Sirve para tener comunicación electrónica y navegar en Internet



Plan de Sesión

3º Semestre de Programación

Unidad 1: Planteamiento de problemas
Tema: Metodología en la solución de problemas
Duración: 90 minutos
Mtro. José Contreras

Segunda sesión

Objetivo(s):

- Que el estudiante conozca la importancia de solucionar problemas personales, de su comunidad, estado o país, utilizando la metodología de solución de problemas.
- Que el estudiante realice el análisis de problemas, siguiendo la metodología de solución de problemas.

Resumen:

- Conocer y seguir la metodología de solución de problemas, es fundamental si se desea obtener resultados óptimos ante un situación problema que se presente. Lograr formular el problema de cualquier situación presentada, precisando los resultados que se esperan, así como identificar los datos de los que se disponen y determinar las restricciones, es esencial para establecer los procesos que llevaran a la solución

Estrategia:

- 05' Síntesis de la sesión anterior
- 10' Introducción al tema de problemas
- 25' Lectura 02 y Actividad 03.
- 10' Introducción al tema de solución de problemas y programación
- 35' Lectura 03 y Actividad 04.
- 05' Síntesis de la sesión.

Actividades de Aprendizaje:

- Realizar las actividades 03 y 04.

Recursos Didácticos:

- Proyector, pizarrón, plumones, internet, equipos de cómputo y/o Smartphone.

Referencias Bibliográficas:

- Elizondo Callejas, R. A. (2010). Informática 2. Serie integral por competencias. México: Patria.
- Joyanes Aguilar, L. (2001). Fundamentos de programación. Algoritmos, Estructuras de datos Objetos. (3ra. ed.). España: Mc Graw Hill.
- Aho, A. V., Hopcroft, J. E., Ullman, J. D. (1998). Estructuras de datos y algoritmos. México: Addison – Wesley Longman.

ADA 04 – Resolvamos un problema

Por equipos de trabajo investigar un problema relacionado a su entorno, escolar, cotidiano, cultural, etc., así como las posibles alternativas para darles solución. Utiliza el siguiente formato.

Análisis de problema
Formular el problema:
Precisar los resultados esperados (meta y submetas):
Identificar datos disponibles (estado inicial):
Determinar las restricciones:
Establecer procesos (operaciones) para la solución:

Guarda tu documento en PDF y súbelo a la plataforma con la siguiente nomenclatura:

Apellidos_Nombre_ADA04 ej. RodriguezMedina_Alejandro_ADA04

Lectura 02 – Solución de problemas

Un problema se puede definir como una situación en la cual se trata de alcanzar una meta y para lograrlo se deben hallar y utilizar unos medios y unas estrategias. La mayoría de problemas tienen algunos elementos en común:

- Un estado inicial;
- Una meta, lo que se pretende lograr;
- Un conjunto de recursos, lo que está permitido hacer y/o utilizar; y
- Un dominio, el estado actual de conocimientos, habilidades y energía de quien va a resolverlo

Casi todos los problemas requieren, que quien los resuelve, los divida en submetas que, cuando son dominadas, llevan a alcanzar el objetivo.

A través del tiempo, la humanidad ha utilizado diversas estrategias generales para resolver problemas. Destacan los siguientes métodos o estrategias de tipo general:

Ensayo y error: Consiste en actuar hasta que algo funcione. Puede tomar mucho tiempo y no es seguro que se llegue a una solución. Es una estrategia apropiada cuando las soluciones posibles son pocas y se pueden probar todas, empezando por la que ofrece mayor probabilidad de resolver el problema.

Lluvia de ideas: Consiste en formular soluciones viables a un problema. El modelo propuesto por Mayer (1992) plantea: definir el problema; generar muchas soluciones (sin evaluarlas); decidir los criterios para estimar las soluciones generadas; y emplear esos criterios para seleccionar la mejor solución.

Algoritmos: Consiste en aplicar adecuadamente una serie de pasos detallados que aseguran una solución correcta. La programación de computadores se apoya en este método.

Otros:

- **Razonamiento analógico.**
- **Pensamiento lateral.**
- **Modelo de procesamiento de información.**
- **Análisis de medios y fines.**
- **Sistemas de producción.**
- **Iluminación.**
- **Heurística.**

Como se puede apreciar, hay muchas estrategias para solucionar problemas; sin embargo, vamos a enfocarnos principalmente en la estrategia algorítmica.

Según Polya (1957), cuando se resuelven problemas, intervienen cuatro operaciones mentales:

1. Entender el problema
2. Trazar un plan
3. Ejecutar el plan (resolver)
4. Revisar

Numerosos autores de textos escolares de matemáticas hacen referencia a estas cuatro etapas planteadas por Polya.

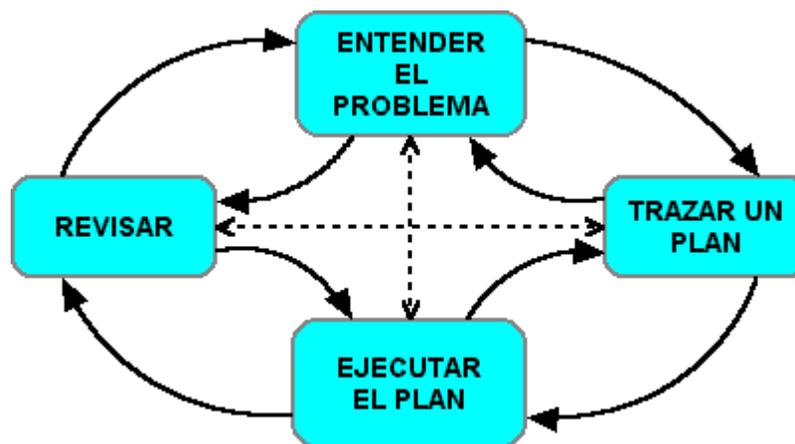


Ilustración 1-1: Interpretación dinámica y cíclica de las etapas planteadas por Polya para resolver problemas.

Es importante hacer énfasis en la naturaleza dinámica y cíclica de la solución de problemas. En el intento de trazar un plan, las personas pueden concluir que necesitan entender mejor el problema y deben regresar a la etapa anterior; o cuando han trazado un plan y tratan de ejecutarlo, no encuentran cómo hacerlo; entonces, la actividad siguiente puede ser intentar con un nuevo plan o regresar y desarrollar una nueva comprensión del problema.

Para llegar a la solución de un problema matemático, se pueden identificar las siguientes sugerencias propuestas:

1. Comprender el problema.

- Leer el problema varias veces
- Establecer los datos del problema
- Aclarar lo que se va a resolver (¿Cuál es la pregunta?)
- Precisar el resultado que se desea lograr
- Determinar la incógnita del problema
- Organizar la información
- Agrupar los datos en categorías
- Trazar una figura o diagrama.

2. Hacer el plan.

- Escoger y decidir las operaciones a efectuar.
- Eliminar los datos inútiles.
- Descomponer el problema en otros más pequeños.

3. Ejecutar el plan (resolver).

- Ejecutar en detalle cada operación.
- Simplificar antes de calcular.
- Realizar un dibujo o diagrama

4. Analizar la solución (revisar).

- Dar una respuesta completa
- Hallar el mismo resultado de otra manera.
- Verificar por apreciación que la respuesta es adecuada.

EJEMPLO

En un juego, el ganador obtiene una ficha roja; el segundo, una ficha azul; y el tercero, una amarilla. Al final de varias rondas, el puntaje se calcula de la siguiente manera: Al cubo de la cantidad de fichas rojas se adiciona el doble de fichas azules y se descuenta el cuadrado de las fichas amarillas. Si Andrés llegó 3 veces en primer lugar, 4 veces de último y 6 veces de intermedio, ¿Qué puntaje obtuvo? (Adaptado de Melo (2001), página 30).

NOTA: Utilizando las etapas propuestas, resuelve el problema.

Lectura 03 – Solución de problemas y programación

Numerosos autores de libros sobre programación, plantean cuatro fases para elaborar un procedimiento que realice una tarea específica. Estas fases concuerdan con las operaciones mentales descritas por Polya para resolver problemas:

1. Analizar el problema (Entender el problema)
2. Diseñar un algoritmo (Trazar un plan)
3. Traducir el algoritmo a un lenguaje de programación (Ejecutar el plan)
4. Depurar el programa (Revisar)

Como se puede apreciar, hay una similitud entre las metodologías propuestas para solucionar problemas matemáticos y las cuatro fases para solucionar problemas específicos de áreas diversas, mediante la programación de computadores.

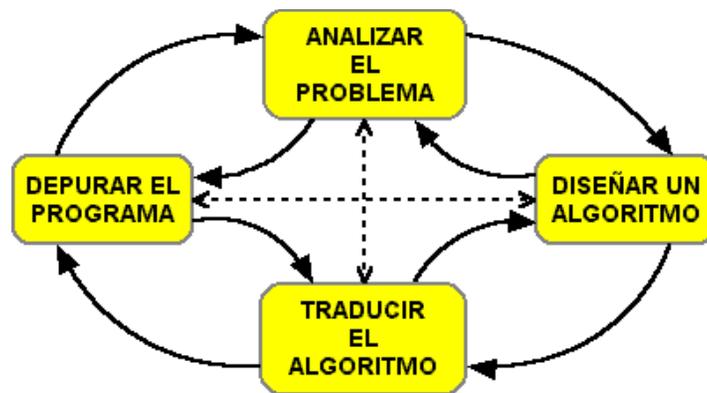


Ilustración 1-3: fases para elaborar un programa de computador.

Fase 1: Analizar el problema (entenderlo)

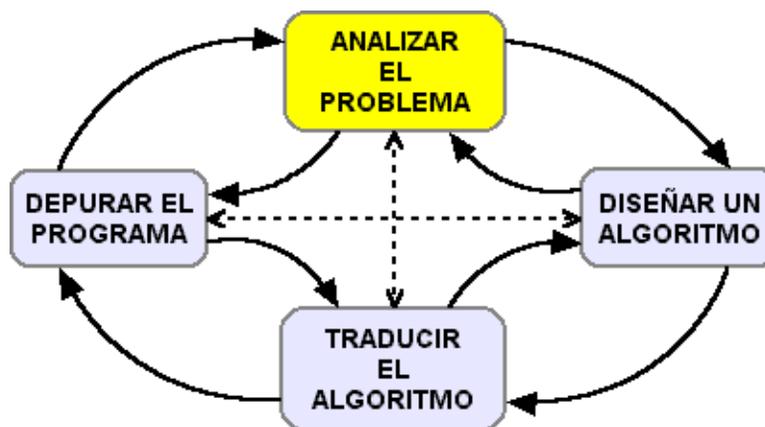


Ilustración 1-4: Primera fase del ciclo de programación.

Los programas de computador tienen como finalidad resolver problemas específicos y el primer paso consiste en definir con precisión el problema hasta lograr la mejor comprensión posible. Una forma de realizar esta actividad se basa en formular claramente el problema, especificar los resultados que se desean obtener, identificar la información disponible (datos), determinar las restricciones y definir los procesos necesarios para convertir los datos disponibles (materia prima) en la información requerida (resultados).

Estas etapas coinciden parcialmente con los elementos generales que están presentes en todos los problemas:

1. Especificar claramente los resultados que se desean obtener (meta y submetas)
2. Identificar la información disponible (estado inicial)
3. Definir los procesos que llevan desde los datos disponibles hasta el resultado deseado (operaciones)

Etapas a desarrollar en la fase de análisis de un problema



Formular el problema

La solución de un problema debe iniciar por determinar y comprender exactamente en qué consiste ese problema. La mayoría de los problemas que se resuelven en el aula de clase llegan a tus manos perfectamente formulados.

EJEMPLO

OPCIÓN 1: Juan Felipe es jefe de bodega en una fábrica de pañales desechables y sabe que la producción diaria es de 744 pañales y que en cada caja donde se empaican para la venta caben 12 pañales. ¿Cuántas cajas debe conseguir Juan Felipe para empaicar los pañales fabricados en una semana?

Sin embargo, esto no sucede en informática, al momento de diseñar un sistema se te planteará una situación en la cual deberás deducir por cuenta propia que es lo que debes hacer o, dicho de otra manera, deberás “Formular el problema” a partir de los resultados esperados.

EJEMPLO

OPCIÓN 2: Juan Felipe es jefe de bodega en una fábrica de pañales desechables y una de las tareas del día consiste en llamar al proveedor de los empaques y ordenarle la cantidad suficiente de cajas para empacar los pañales fabricados en la semana próxima. El jefe de producción le informó ayer a Juan Felipe que la producción diaria será de 744 pañales y en cada caja cabe una docena de ellos. ¿Qué debe hacer Felipe?

La comprensión lingüística del problema (entender el significado de cada enunciado) es muy importante. El estudiante debe realizar una lectura previa del problema con el fin de obtener una visión general de lo que se le pide y una segunda lectura para poder responder preguntas como:

- ¿Puedo definir mejor el problema?
- ¿Qué palabras del problema me son desconocidas?
- ¿Cuáles son las palabras clave del problema?
- ¿He resuelto antes algún problema similar?
- ¿Qué información es importante?
- ¿Qué información puedo omitir?

En esta etapa se debe hacer una representación precisa del problema (Rumbaugh, 1996); especificar lo más exactamente posible lo que hay que hacer (no cómo hay que hacerlo).

Precisar los resultados esperados (meta y submetas)

Para establecer los resultados que se esperan (meta) es necesario identificar la información relevante, ignorar los detalles sin importancia, entender los elementos del problema y activar el esquema correcto que permita comprenderlo en su totalidad.

Determinar con claridad cuál es el resultado final (producto) que debe devolver el programa es algo que ayuda a establecer la meta. Es necesario analizar qué resultados se solicitan y qué formato deben tener esos resultados.

El estudiante debe preguntarse:

- ¿Qué información me solicitan?
- ¿Qué formato debe tener esta información?

Identificar datos disponibles (estado inicial)

Otro aspecto muy importante en la etapa de análisis del problema consiste en determinar cuál es la información disponible. Deberás preguntarte:

- ¿Qué información es importante?
- ¿Qué información no es relevante?
- ¿Cuáles son los datos de entrada? (conocidos)
- ¿Cuál es la incógnita?
- ¿Qué información me falta para resolver el problema? (datos desconocidos)
- ¿Puedo agrupar los datos en categorías?

EJEMPLO

Esteban está ahorrando para comprar una patineta que vale 55.000 pesos. Su papá le ha dado una mesada de 5.000 pesos cada semana durante 7 semanas. Por lavar el auto de su tío tres veces recibió 8.000 pesos. Su hermano ganó 10.000 pesos por hacer los mandados de su mamá y 4.000 por sacar a pasear el perro. ¿Esteban tiene ahorrado el dinero suficiente para comprar la patineta o aún le falta?

NOTA: Antes de continuar resuelve el problema. Es importante para tu aprendizaje.

R/.

Formular el problema: Ya se encuentra claramente planteado.

Resultados esperados: Si o no tiene Esteban ahorrado el dinero suficiente para comprar una patineta que vale 55.000 pesos. Datos disponibles: Los ingresos de Esteban: 5.000 pesos por 7 semanas + 8.000 pesos. Los 10.000 y 4.000 pesos que ganó el hermano de Esteban son irrelevantes para la solución de este problema y se pueden omitir.

Determinar las restricciones

Resulta fundamental que determines aquello que está permitido o prohibido hacer y/o utilizar para llegar a una solución. En este punto se deben exponer las necesidades y restricciones (no una propuesta de solución). Deberás preguntarte:

- ¿Qué condiciones me plantea el problema?
- ¿Qué está prohibido hacer y/o utilizar?
- ¿Qué está permitido hacer y/o utilizar?
- ¿Cuáles datos puedo considerar fijos (constantes) para simplificar el problema?
- ¿Cuáles datos son variables?
- ¿Cuáles datos debo calcular?

Establecer procesos (operaciones)

Consiste en determinar los procesos que permiten llegar a los resultados esperados a partir de los datos disponibles. Deberás preguntarte:

- ¿Qué procesos necesito?
- ¿Qué fórmulas debo emplear?
- ¿Cómo afectan las condiciones a los procesos?
- ¿Qué debo hacer?
- ¿Cuál es el orden de lo que debo hacer?

En la medida de lo posible, es aconsejable dividir el problema original en otros más pequeños y fáciles de solucionar (submetas), hasta que los pasos para alcanzarlas se puedan determinar con bastante precisión (módulos). Esto es lo que en programación se denomina diseño descendente o top-down.

Al realizar divisiones sucesivas del problema en otros más pequeños y manejables (módulos), hay que tener cuidado para no perder de vista la comprensión de este como un todo. Luego de dividir el problema original en submetas (módulos), debes integrar cada parte de tal forma que le permita comprender el problema como un todo.

No olvides las otras fases del ciclo de programación

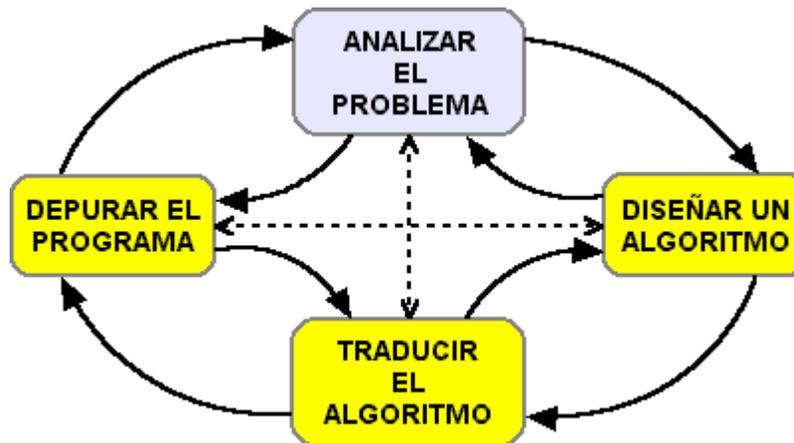


Ilustración 1-7: Fases segunda, tercera y cuarta, del ciclo de programación.

Hasta este punto terminamos. Trabajaremos con la fase dos más adelante.
¡¡Buen trabajo!!

Plan de Sesión

3º Semestre de Programación

Unidad 1: Planteamiento de problemas
Tema: Metodología en la solución de problemas
Duración: 90 minutos
Mtro. José Contreras

Tercera sesión

Objetivo(s):

- Que el estudiante identifique problemas de su comunidad, estado o país, y plantee la manera de abordarlos utilizando la metodología de solución de problemas.

Resumen:

- Observar en el entorno las diferentes situaciones que se presentan e identificar un problema existente, es fundamental para poder proponer una solución. Si se desea obtener resultados óptimos ante la situación problema identificada, es importante seguir paso a paso la metodología para la solución de problemas. La práctica de dicha metodología, permitirá formular el problema de cualquier situación presentada, precisando los resultados que se esperan, así como identificar los datos de los que se disponen, las restricciones y los procesos que llevarán a la solución

Estrategia:

- 05' Síntesis de la sesión anterior
- 45' Actividad 05
- 35' Presentación de la actividad al grupo.
- 05' Síntesis de la sesión.

Actividades de Aprendizaje:

- Realizar la actividad 05.

Recursos Didácticos:

- Proyector, pizarrón, plumones, internet, equipos de cómputo y/o Smartphone.

Referencias Bibliográficas:

Elizondo Callejas, R. A. (2010). Informática 2. Serie integral por competencias. México: Patria.
Joyanes Aguilar, L. (2001). Fundamentos de programación. Algoritmos, Estructuras de datos y Objetos. (3ra. ed.). España: Mc Graw Hill.
Aho, A. V., Hopcroft, J. E., Ullman, J. D. (1998). Estructuras de datos y algoritmos. México: Addison – Wesley Longman.

ADA 05 – Identificando problemas

En equipos de 3 integrantes, realicen una presentación, en donde siguiendo la metodología de solución de problemas, para dos situaciones reales, formulen el problema de cada situación, precisen los resultados esperados (meta y submetas), identifiquen los datos disponibles (estado inicial), determinen las restricciones y establezcan los procesos (operaciones) para la solución de cada una de ellas. Apóyense en el formato utilizado en la actividad 04

Guarda tu documento en formato de presentación electrónica y súbelo a la plataforma con la siguiente nomenclatura:

Apellidos_Nombre_ADA05 ej. RodriguezMedina_Alejandro_ADA05

Plan de Sesión

3º Semestre de Programación

Unidad 1: Planteamiento de problemas

Tema: Integración de habilidades

Duración: 90 minutos

Mtro. José Contreras

Cuarta sesión

Objetivo(s):

- Que el estudiante aplique la metodología de solución de problemas para resolver problemas de la vida real.

Resumen:

- Después de haber trabajado en la solución de problemas durante las semanas anteriores, es tiempo de integrar las habilidades obtenidas hasta el momento.

Estrategia:

- 05' Síntesis de la sesión anterior
- 70' Actividad 06
- 10' Presentación de la actividad al grupo.
- 05' Síntesis de la sesión.

Actividades de Aprendizaje:

- Realizar la actividad 06.

Recursos Didácticos:

- Proyector, pizarrón, plumones, internet, equipos de cómputo y/o Smartphone.

Referencias Bibliográficas:

- Elizondo Callejas, R. A. (2010). Informática 2. Serie integral por competencias. México: Patria.
- Joyanes Aguilar, L. (2001). Fundamentos de programación. Algoritmos, Estructuras de datos Objetos. (3ra. ed.). España: Mc Graw Hill.
- Aho, A. V., Hopcroft, J. E., Ullman, J. D. (1998). Estructuras de datos y algoritmos. México: Addison – Wesley Longman.

ADA 06 – Integración de habilidades

Recientemente 12 niños y su entrenador quedaron atrapados en una cueva en Tailandia. El país a pedido ayuda internacional y México ha decidido enviar un equipo de apoyo. Tú y dos compañeros serán enviados para colaborar en el rescate.

La misión de tu equipo es participar en el plan de rescate, para lo cual deberán realizar el análisis de la situación.

Entrega: El formato de actividad 06 con la siguiente información:

- Formular el problema
- Precisar los resultados esperados (meta y submetas)
- Identificar datos disponibles (estado inicial)
- Determinar las restricciones
- Establecer procesos (operaciones)

Guarda tu documento en PDF y súbelo a la plataforma con la siguiente nomenclatura:

Equipo_X_ADA01 ej. Equipo_4_ADA01

Apóyate en las lecturas de la unidad, en internet y en las siguientes ligas:

- <https://ww.elmundo.es/internacional/2018/07/09/5b433160ca4741fd308b4607.html>
- <https://www.elmundo.es/internacional/2018/07/08/5b423b4be5fdea23318b4600.html>
- http://www.elmundo.es/internacional/2018/07/02/5b3a4a1522601de06d8b4595.html?cid=MNOT23801&s_kw=1
- <http://www.elmundo.es/internacional/2018/07/03/5b3b95b8ca474176128b460d.html>
- <http://www.elmundo.es/internacional/2018/07/02/5b3a4a1522601de06d8b4595.html>
- http://www.elmundo.es/internacional/2018/07/09/5b431acd22601d81478b459f.html?cid=MNOT23801&s_kw=3
- https://elpais.com/internacional/2018/07/09/actualidad/1531128804_560904.html
- https://elpais.com/elpais/2018/07/10/videos/1531214313_294471.html
- https://elpais.com/internacional/2018/07/10/actualidad/1531203121_258727.html

Guía de evaluación – ADA 06

Criterios de evaluación	5	4	3	2	1
1. Formula el problema.					
2. Describe los datos de entrada con que se cuentan					
3. Describe los datos de salida que se desean					
4. Describe el proceso para solucionar el problema					
5. Describe las restricciones que se tienen					
6. El proceso tiene congruencia con los resultados esperados.					
7. Sigue correctamente y en orden los pasos para analizar un problema					

Plan de Sesión

3º Semestre de Programación

Unidad 1: Algoritmos
Tema: Introducción a los algoritmos
Duración: 90 minutos
Mtro. José Contreras

Quinta sesión

Objetivo(s):

- Que el estudiante sepa que es un algoritmo, como se clasifican, las partes que lo integran, las formas de representarlo y las características que debe tener para que sea funcional.

Resumen:

- Los algoritmos son la base de la programación y de la automatización de las soluciones para los problemas presentados. Poder diseñarlos teniendo en cuenta las características que lo harán funcional y las diferentes maneras de representarlos, será fundamental en la solución de problemas, tanto de software como de la vida real.

Estrategia:

- 05' Síntesis de la sesión anterior
- 10' Introducción al tema de algoritmos
- 45' Lectura 04 y Actividad 07
- 10' Compartir con el grupo
- 15' Actividad 08 – Cuadro sinóptico
- 05' Síntesis de la sesión.

Actividades de Aprendizaje:

- Realizar la actividad 07.

Recursos Didácticos:

- Proyector, pizarrón, plumones, internet, equipos de cómputo y/o Smartphone.

Referencias Bibliográficas:

- Elizondo Callejas, R. A. (2010). Informática 2. Serie integral por competencias. México: Patria.
- Joyanes Aguilar, L. (2001). Fundamentos de programación. Algoritmos, Estructuras de datos Objetos. (3ra. ed.). España: Mc Graw Hill.
- Aho, A. V., Hopcroft, J. E., Ullman, J. D. (1998). Estructuras de datos y algoritmos. México: Addison – Wesley Longman.

ADA 07 – Kahoot

1. ¿Cuál es el orden de las etapas o fases del ciclo de programación?
 - Analizar el problema
 - Diseñar un algoritmo
 - Traducir el algoritmo
 - Depurar el programa
2. Conjunto de instrucciones que le indican al hardware que tipo de acciones debe efectuar con los datos
 - Programa para computadora o software
3. Es la acción de codificar o escribir esas instrucciones sobre un entorno llamado lenguaje de programación, para generar programas
 - La programación
4. Herramienta que permite describir claramente un conjunto finito de instrucciones, ordenadas secuencialmente y libres de ambigüedad, que debe llevar a cabo un computador para lograr un resultado previsible.
 - Los Algoritmos
5. Cuáles son los tres elementos que conforman los algoritmos
 - Entrada, Proceso y Salida.
6. Son los datos que se requiere para la ejecución del algoritmo
 - Datos de entrada
7. Es un ejemplo de datos de entrada.
 - Clave de acceso
8. Son las diferentes operaciones que se realizan para trabajar los datos y generar la información requerida por el usuario
 - Proceso
9. Antes de iniciar el diseño de un algoritmo debemos tener:
 - Una definición clara y precisa del problema
10. Es una característica de los algoritmos.
 - Debe ser finito

11. Característica de los algoritmos que nos dice que: No se deben permitir dobles interpretaciones.

- Debe estar definido

12. Tres partes que debe describir la definición de un algoritmo

- Entrada
- Proceso
- Salida

13. Tipos de algoritmos

- Cualitativos
- Cuantitativos

14. Tipos de lenguajes Algorítmicos

- Gráficos
- No gráficos

15. Métodos para representar un algoritmo

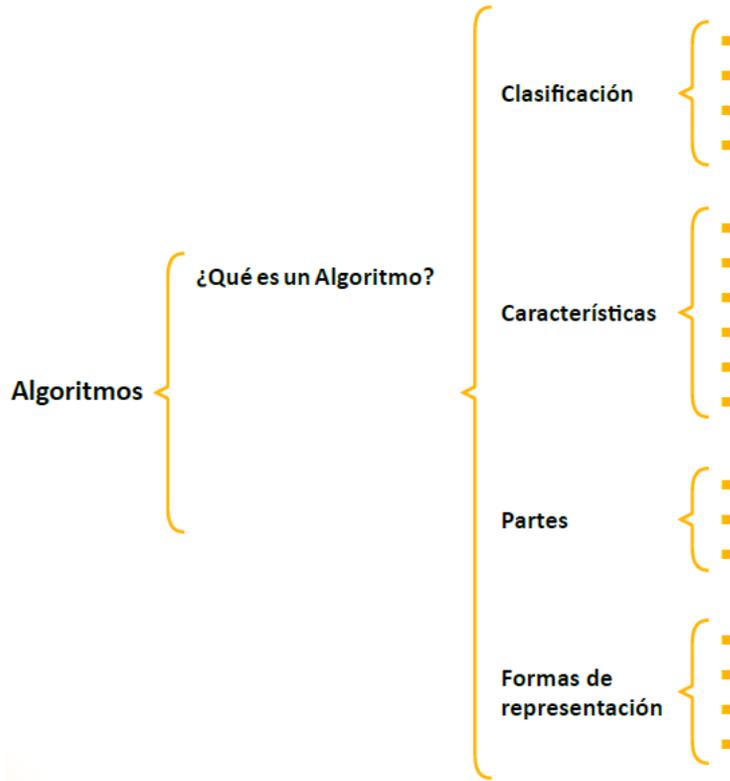
- Lenguaje natural
- Diagrama de flujo
- Pseudocódigo

Toma un Screenshot de tu resultado en Kahoot y súbelo a la plataforma con la siguiente nomenclatura:

Apellidos_Nombre_ADA07 ej. RodriguezMedina_Alejandro_07

ADA 08 – Cuadro sinóptico

Completa la información del siguiente cuadro sinóptico, incluyendo las definiciones de los conceptos completados.



Guarda el cuadro completado en formato PDF y súbelo a la plataforma con la siguiente nomenclatura:

Apellidos_Nombre_ADA08 ej. RodriguezMedina_Alejandro_ADA08

Lectura 04 – Algoritmos en programación

Por el momento, podemos resumir que únicamente luego de analizar detalladamente el problema hasta entenderlo completamente se pasa a la siguiente etapa o fase del ciclo de programación, la cual es diseñar un Algoritmo, pero.... ¿Qué es un algoritmo?

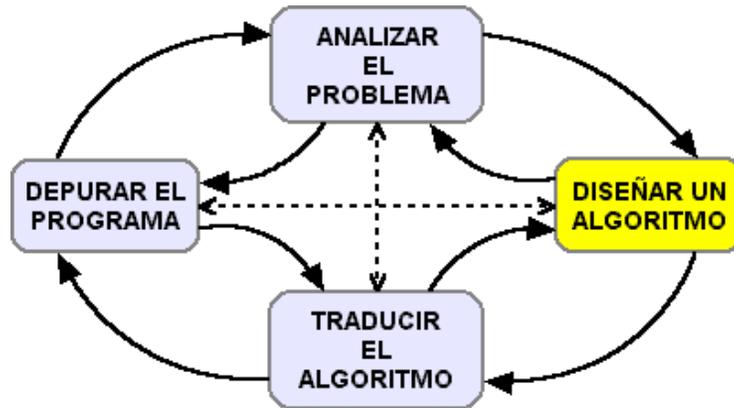


Ilustración 2-1: Segunda fase del ciclo de programación.

¿Qué es un algoritmo?

Antes de entrar de lleno al tema de algoritmos, hablemos un poco acerca de programación. Con la computadora podemos diseñar soluciones a la medida, de problemas específicos que se nos presenten. Como vimos anteriormente, un programa para computadora es un conjunto de instrucciones que le indican al hardware que tipo de acciones debe efectuar con los datos. La programación es la acción de codificar o escribir esas instrucciones sobre un entorno llamado lenguaje de programación, para generar programas

Existen lenguajes de programación para casi todas las necesidades, desde aplicaciones para manejar directamente un circuito electrónico, sistemas de nómina, redes sociales, hasta software para registrarse y realizar compras por Internet. Incluso los equipos móviles emplean programas para buscar un nombre en la lista telefónica, para jugar, navegar en internet y muchas más opciones según el tipo de equipo con el que se cuenta.

Con la programación se puede hacer lo que se te ocurra y se necesite, es un amplio mundo con muchísimos campos de aplicación. Sin embargo, el diseño de soluciones a la medida de nuestros problemas, requiere como en otras disciplinas una metodología que nos enseñe de manera gradual, la forma de llegar a estas soluciones.

La metodología necesaria para resolver problemas mediante programas, se denomina metodología de la programación y el eje central de esta metodología es el **algoritmo**.

El desarrollo de algoritmos es un tema fundamental en el diseño de programas por lo cual el aprendiz debe tener buenas bases que le sirvan para poder desarrollar de manera fácil y rápida sus programas. Necesita aprender a resolver problemas de un modo riguroso y sistemático. Y básicamente un algoritmo es un método para resolver un problema.

Definiciones de algoritmo.

1. Un algoritmo es una serie de pasos organizados que describe el proceso que se debe seguir, para dar solución a un problema específico.
2. Un algoritmo es una secuencia de pasos coherentes y finitos para resolver un problema.
3. En un algoritmo se fijan una serie de acciones precisas y lógicas que conllevan a resolver cualquier problema.

En el ámbito de la computación, los Algoritmos son una herramienta que permite describir claramente un conjunto finito de instrucciones, ordenadas secuencialmente y libres de ambigüedad, que debe llevar a cabo un computador para lograr un resultado previsible. Vale la pena recordar que un programa de computador consiste de una serie de instrucciones muy precisas y escritas en un lenguaje de programación que el computador entienda como lo es Scratch (Que veremos más adelante).

Los algoritmos están conformados por tres elementos: Entrada, Proceso y Salida. En algunos casos un algoritmo puede requerirlos en cualquier orden, o emplear uno o varios, según la temática que se plantee para resolver el problema. Analiza el siguiente ejemplo:

Suponga que se desea sumar dos números, y saber el valor de esa suma.

Entrada: es conocer los dos números, por ejemplo 5 y 9.

Proceso: ejecutar la operación.

$$\begin{array}{r} 5 \\ + 9 \\ \hline 14 \end{array}$$

Salida: es obtener el valor de la suma que es igual a 14. Ya se ha realizado el primer algoritmo, se creó una secuencia de pasos coherentes y dependientes.

Coherentes en la medida que se obtuvo un valor de la forma correcta como siempre se efectúa una suma.

Dependiente en que no se puede sumar dos números que no se conocen, y obtener el resultado de la suma sin sumar los números.

DATOS DE ENTRADA

Se entiende por datos de entrada, todos los datos de los cuáles se requiere para la ejecución del algoritmo. Para ilustrar esta definición se planteará el siguiente ejemplo: Cuando se desea retirar dinero del cajero electrónico, se ingresa la tarjeta de crédito a la ranura del cajero, seguidamente la pantalla del cajero solicita el número de la clave para comprobar que el número de clave corresponde a esa tarjeta de crédito y realizar las operaciones que se desean.

- El dato de entrada para esta operación fue el número de clave. Sin ella, la tarjeta de crédito no tendría validez alguna.

Para ser más generales se entiende como datos de entrada todas aquellas acciones que requieren del ingreso de datos, bien sea desde el teclado (como números de clave, valores, nombre, entre otros), como selecciones de menús de opciones (tipo de operación, cantidad de efectivo, entre otros.)

PROCESO

Corresponde a las diferentes operaciones que se realizan para procesar datos y generar información requerida por el usuario. Continuando con el ejemplo del cajero electrónico, tenemos:

Al ingresarse el número de la clave, el cajero electrónico realiza la operación de validar la tarjeta, como se mencionó anteriormente. Este proceso de validación lo hace revisando el conjunto de datos o registros de usuarios y tratando de ubicar al cliente. Si encuentra al cliente en la lista de clientes, le permite realizar operaciones. Pero si no lo encuentra le indica que la clave es incorrecta y debe ingresar otro número de clave.

De forma homóloga cuando se presionan teclas en un computador, este genera una cantidad de procesos para capturar la tecla presionada y determinar que tipo de operación se desea realizar.

DATOS DE SALIDA.

Los datos de salida son el objetivo por el cual se genera un proceso. En otras palabras, es la función por la cual se implementan sistemas de información (software). Los datos de salida que pueden visualizarse en la pantalla del computador en una página mediante la impresión, es realmente lo que le interesa el usuario del sistema. Continuando con el ejemplo:

Terminada la comprobación del número de clave, el usuario solicita un recibo para conocer su saldo. Selecciona la operación de impresión del recibo y espera la impresión del mismo.

La resolución de un problema mediante programación exige el diseño de un algoritmo que resuelva el problema propuesto. Recuerda que se requiere una definición clara y precisa del problema. Es importante que se conozca lo que se desea que realice la computadora; mientras esto no se conozca del todo no tiene caso pasar a la etapa de diseño del algoritmo.

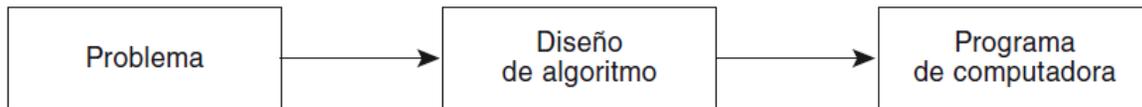


Figura 2.14. Resolución de un problema.

Los pasos para la resolución de un problema mediante programación son:

1. Diseño del algoritmo, que describe la secuencia ordenada de pasos que conducen a la solución de un problema dado. (Análisis del problema y desarrollo del algoritmo.)
2. Expresar el algoritmo como un programa en un lenguaje de programación adecuado. (Fase de codificación o Programación)
3. Ejecución y validación del programa por la computadora.

Para llegar a la realización de un programa es necesario el diseño previo de un algoritmo, de modo que sin algoritmo no puede existir un programa.

Los algoritmos son independientes tanto del lenguaje de programación en que se expresan como de la computadora que los ejecuta. En cada problema el algoritmo se puede expresar en un lenguaje diferente de programación y ejecutarse en una computadora distinta; sin embargo, el algoritmo será siempre el mismo. Así, por ejemplo, en una analogía con la vida diaria, una receta de un plato de cocina se puede expresar en español, inglés o francés, pero cualquiera que sea el lenguaje, los pasos para la elaboración del plato se realizarán sin importar el idioma del cocinero.

En la ciencia de la computación y en la programación, los algoritmos son más importantes que los lenguajes de programación o las computadoras. Un lenguaje de programación es tan sólo un medio para expresar un algoritmo y una computadora es sólo un procesador para ejecutarlo. Tanto el lenguaje de programación como la computadora son los medios para obtener un fin: conseguir que el algoritmo se ejecute y se efectúe el proceso correspondiente.

Dada la importancia del algoritmo en la ciencia de la computación, un aspecto muy importante será el diseño de algoritmos. El diseño de la mayoría de los algoritmos requiere creatividad y conocimientos profundos de la técnica de la programación.

Características de los algoritmos

Las características fundamentales que debe cumplir todo algoritmo son:

- Debe tener un punto de inicio.
- Debe ser preciso e indicar el orden de realización de cada paso, es decir, el orden de ejecución de las instrucciones debe estar perfectamente indicado. Cuando se ejecuta varias veces, con los mismos datos iniciales, el resultado debe ser el mismo siempre. La precisión implica determinismo.
- Debe estar definido. Si se sigue un algoritmo dos veces, se debe obtener el mismo resultado cada vez. No debe permitir dobles interpretaciones. Debe ser claro lo que hace, de forma que quien ejecute los pasos sepa qué, cómo y cuándo hacerlo.
- Debe ser finito. Si se sigue un algoritmo, se debe terminar en algún momento. Debe terminar después de una cantidad finita de pasos.
- Sólo puede ejecutarse una operación a la vez.
- Debe ser general, es decir, soportar la mayoría de las variantes que se puedan presentar en la definición del problema.

Como ya mencionamos, la definición de un algoritmo debe describir tres partes: Entrada, Proceso y Salida. En el algoritmo de receta de cocina citado anteriormente se tendrá:

- Entrada: Ingredientes y utensilios empleados.
- Proceso: Elaboración de la receta en la cocina.
- Salida: Terminación del plato (por ejemplo, cordero).

La ejecución de un algoritmo no debe implicar ninguna decisión subjetiva, ni tampoco debe de hacer preciso el uso de la intuición ni de la creatividad. Por tanto, se puede considerar que una receta de cocina es un algoritmo si describe precisamente la forma de preparar un cierto plato, proporcionándonos las cantidades exactas que deben de utilizarse y también instrucciones detalladas acerca del tiempo que debe de guisarse. Por otra parte, si se incluyen nociones vagas tales como “Salpimentar a su gusto” o “Guísese hasta que este medio hecho” entonces no se podría llamar algoritmo.

PASOS PARA REALIZAR TAREAS

En la naturaleza hay muchos procesos que puedes considerar como Algoritmos ya que tienen procedimientos y reglas. Incluso, muchas veces no somos conscientes de ellos.

Por ejemplo, el proceso digestivo es un concepto de algoritmo con el que convivimos a diario sin que nos haga falta una definición precisa de este proceso. El hecho de que conozcamos cómo funciona el sistema digestivo, no implica que los alimentos que consumimos nos alimenten más o menos.

La familiaridad de lo que sucede día a día nos impide ver muchos algoritmos que pasan a nuestro alrededor. Procesos naturales como la gestación, las estaciones, la circulación sanguínea, los ciclos planetarios, etc., son algoritmos naturales que generalmente pasan desapercibidos.

En resumen, un Algoritmo es una secuencia ordenada de instrucciones, pasos o procesos que llevan a la solución de un determinado problema. Los hay tan sencillos y cotidianos como seguir la receta del médico, abrir una puerta, lavarse las manos, entre otros; hasta los que conducen a la solución de problemas muy complejos.

Ejemplo 01:

Lavarnos los dientes es un procedimiento que realizamos varias veces al día. Veamos la forma de expresar este procedimiento como un Algoritmo:

Inicio

1. Tomar la crema dental
2. Destapar la crema dental
3. Tomar el cepillo de dientes
4. Aplicar crema dental al cepillo
5. Tapar la crema dental
6. Abrir la llave del lavamanos
7. Remojar el cepillo con la crema dental
8. Cerrar la llave del lavamanos
9. Frotar los dientes con el cepillo
10. Abrir la llave del lavamanos
11. Enjuagarse la boca
12. Enjuagar el cepillo
13. Cerrar la llave del lavamanos
14. Secarse la cara y las manos con una toalla

Fin

Ejemplo 02:

Veamos que algo tan común como los pasos para cambiar una bombilla (foco) se pueden expresar en forma de Algoritmo:

Inicio

1. Ubicar una escalera o un banco debajo de la bombilla fundida
2. Tomar una bombilla nueva
3. Subir por la escalera o al banco
4. Girar la bombilla fundida hacia la izquierda hasta soltarla
5. Enroscar la bombilla nueva hacia la derecha en el plafón hasta apretarla
6. Bajar de la escalera o del banco

Fin

Un ejemplo elemental es el Algoritmo 5.1.

Algoritmo 5.1 Preparar una taza de té.

Entrada: tetera, taza, bolsa de té

Salida: taza de té

Inicio

- Tomar la tetera
- Llenarla de agua
- Encender el fuego
- Poner la tetera en el fuego
- Esperar a que hierva el agua
- Tomar la bolsa de té
- Introducirla en la tetera
- Esperar 1 minuto
- Echar el té en la taza

Fin

A diferencia de los seres humanos que realizan actividades sin detenerse a pensar en los pasos que deben seguir, los computadores son muy ordenados y necesitan que quien los programan les diga cada uno de los pasos que deben realizar y el orden lógico de ejecución.

Numera en orden lógico los pasos siguientes (para pescar):

- ___ El pez se traga el anzuelo.
- ___ Enrollar el sedal.
- ___ Tirar el sedal al agua.
- ___ Llevar el pescado a casa.
- ___ Quitar el Anzuelo de la boca del pescado.
- ___ Poner carnada al anzuelo.
- ___ Sacar el pescado del agua.

Es importante tener en cuenta que un problema siempre podrá ser resuelto por una cantidad N de algoritmos. Durante el diseño es posible y aconsejable, realizar comparaciones entre algoritmos que resuelven el mismo problema y elegir el mejor según el objetivo.

La bondad de un algoritmo puede medirse por dos factores:

- El tiempo que se necesita para ejecutarlo. Para tener una idea aproximada de ello, basta con saber el número de instrucciones de cada tipo necesarias para resolver el problema.
- Los recursos que se necesitan para implantarlo.

Así, una vez diseñado un primer algoritmo, conviene realizar una evaluación del mismo. Si se decide que éste no es eficiente será necesario o bien diseñar uno nuevo o bien optimizar el original. Optimizar un algoritmo consiste en introducir modificaciones en él, tendentes a disminuir el tiempo que necesita para resolver el problema o a reducir los recursos que utiliza.

TIPOS DE ALGORITMOS

- **Cualitativos:** Son aquellos en los que se describen los pasos utilizando palabras.
- **Cuantitativos:** Son aquellos en los que se utilizan cálculos numéricos para definir los pasos del proceso.

Lenguajes Algorítmicos: Es una serie de símbolos y reglas que se utilizan para describir de manera explícita un proceso.

Tipos de Lenguajes Algorítmicos

- **Gráficos:** Es la representación gráfica de las operaciones que realiza un algoritmo. El más utilizado es el **diagrama de flujo**, que es una representación gráfica de secuencias de pasos a realizar.
- **No Gráficos:** Representa en forma descriptiva las operaciones que debe realizar un algoritmo. El más común es el **Pseudocódigo** que describe un algoritmo de forma similar a un lenguaje de programación, pero sin su rigidez. Su forma es más parecida al lenguaje natural.

REPRESENTACIÓN DE LOS ALGORITMOS

Para representar un algoritmo se debe utilizar algún método que permita independizar dicho algoritmo del lenguaje de programación elegido. Ello permitirá que un algoritmo pueda ser codificado indistintamente en cualquier lenguaje. Para conseguir este objetivo se precisa que el algoritmo sea representado gráfica o numéricamente, de modo que las sucesivas acciones no dependan de la sintaxis de ningún lenguaje de programación, sino que la descripción pueda servir fácilmente para su transformación en un programa, es decir, su codificación.

Los métodos usuales para representar un algoritmo son:

- Lenguaje natural (español, inglés...)
- Diagrama de flujo,
- Lenguaje de especificación de algoritmos: pseudocódigo,

Plan de Sesión

3º Semestre de Programación

Unidad 1: Algoritmos
Tema: Algoritmos en la vida cotidiana
Duración: 90 minutos
Mtro. José Contreras

Sexta sesión

Objetivo(s):

- Que el estudiante reflexione sobre el uso de algoritmos para resolver problemas de la vida cotidiana.
- Que el estudiante diseñe algoritmos teniendo en cuenta las características de los algoritmos funcionales..

Resumen:

- Entender la utilidad de los algoritmos para resolver problemas de la vida cotidiana, es fundamental para relacionar su uso con situaciones diarias. Diseñarlos teniendo en cuenta sus características, asegurará que el resultado sea un algoritmo funcional .

Estrategia:

- 05' Síntesis de la sesión anterior
- 10' Introducción a algoritmos en la vida cotidiana
- 25' Actividad 09 y compartida
- 25' Actividad 10
- 35' Compartir con tres compañeros y verificar que el algoritmo es funcional
- 05' Síntesis de la sesión.

Actividades de Aprendizaje:

- Realizar las actividades 09 y 10.

Recursos Didácticos:

- Proyector, pizarrón, plumones, internet, equipos de cómputo y/o Smartphone.

Referencias Bibliográficas:

- Elizondo Callejas, R. A. (2010). Informática 2. Serie integral por competencias. México: Patria.
- Joyanes Aguilar, L. (2001). Fundamentos de programación. Algoritmos, Estructuras de datos Objetos. (3ra. ed.). España: Mc Graw Hill.
- Aho, A. V., Hopcroft, J. E., Ullman, J. D. (1998). Estructuras de datos y algoritmos. México: Addison – Wesley Longman.

ADA 09 – El avión más veloz

Trabajas en una compañía de diseño de aviones y te han elegido para participar en un concurso y demostrar que sus aviones son los mejores.

Elabora un avión de papel que te permita competir y obtener ese primer lugar que todos desean.

Al terminar la actividad, cuando lo indique el docente, comparte tu experiencia con el grupo respondiendo a las siguientes preguntas:

- ¿Para qué me sirvieron los algoritmos?
- ¿Qué hubiese sucedido sino los hubiese empleado?
- ¿De qué me di cuenta?
- Cómo me pueden servir los algoritmos en mi vida?

Tomate un selfie con tus aviones construidos y súbelo a la plataforma con la siguiente nomenclatura:

Apellidos_Nombre_ADA09 ej. RodriguezMedina_Alejandro_ADA09

ADA 10 – Creando algoritmos

Describe detalladamente el algoritmo para lograr cada una de las siguientes tareas

1. Ir a la escuela (desde tu cama, hasta el aula)
2. Prepárate un sándwich
3. Sacar la basura para que se la lleven
4. Tomar un boomerang y colocarlo de estado en WhatsApp
5. Realizar una vídeo llamada con Facebook desde la computadora de un amigo
6. Publicar una historia en Instagram
7. Cambiar la llanta de un automóvil
8. Un algoritmo que creas que es útil en tu vida

Guarda tu documento en PDF y súbelo a la plataforma con la siguiente nomenclatura:

Apellidos_Nombre_ADA10 ej. RodriguezMedina_Alejandro_ADA10

Plan de Sesión

3º Semestre de Programación

Unidad 1: Algoritmos
Tema: Diseño de algoritmos
Duración: 90 minutos
Mtro. José Contreras

Séptima sesión

Objetivo(s):

- Que el estudiante diseñe algoritmos para resolver los problemas que se le planteen, teniendo en cuenta sus características,

Resumen:

- Diseñar algoritmos para resolver las diferentes situaciones que se planteen, teniendo en cuenta sus características, asegurará que el resultado sea un algoritmo funcional .

Estrategia:

- 05' Síntesis de la sesión anterior
- 10' Introducción al diseño de algoritmos
- 45' Actividad 11
- 25' Compartir con el grupo y verificar que el algoritmo es funcional
- 05' Síntesis de la sesión.

Actividades de Aprendizaje:

- Realizar la actividad 11.

Recursos Didácticos:

- Proyector, pizarrón, plumones, internet, equipos de cómputo y/o Smartphone.

Referencias Bibliográficas:

- Elizondo Callejas, R. A. (2010). Informática 2. Serie integral por competencias. México: Patria.
- Joyanes Aguilar, L. (2001). Fundamentos de programación. Algoritmos, Estructuras de datos Objetos. (3ra. ed.). España: Mc Graw Hill.
- Aho, A. V., Hopcroft, J. E., Ullman, J. D. (1998). Estructuras de datos y algoritmos. México: Addison – Wesley Longman.

ADA 11 – Pensamientos algorítmico y procedimental

Escribe el algoritmo para resolver el nivel 5 de cada uno de los siguientes juegos. No olvides respetar las características de los algoritmos

- <http://cu.game-game.com/178531/>
- <http://cu.game-game.com/169716/>
- <http://www.minijuegos.com/juego/sokoban-mega-mine>
- <http://cu.game-game.com/93136/>

Guarda tu documento en PDF con la siguiente nomenclatura y súbelo a la plataforma:

Apellidos_Nombre_ADA11 ej. RodriguezMedina_Alejandro_ADA11

Plan de Sesión

3º Semestre de Programación

Unidad 1: Algoritmos
Tema: Diseño de algoritmos
Duración: 90 minutos
Mtro. José Contreras

Octava sesión

Objetivo(s):

- Que el estudiante practique el diseño de algoritmos para resolver los problemas que se le planteen, teniendo en cuenta sus características,

Resumen:

- Practicar el diseño de algoritmos para resolver las diferentes situaciones que se planteen, teniendo en cuenta sus características, es fundamental para desarrollar la habilidad y asegurar que el resultado sea funcional .

Estrategia:

- 05' Síntesis de la sesión anterior
- 45' Actividad 12
- 15' Compartir con el grupo y resolver los problemas
- 20' Actividad 13
- 05' Síntesis de la sesión.

Actividades de Aprendizaje:

- Realizar las actividades 12 y 13.

Recursos Didácticos:

- Proyector, pizarrón, plumones, internet, equipos de cómputo y/o Smartphone.

Referencias Bibliográficas:

- Elizondo Callejas, R. A. (2010). Informática 2. Serie integral por competencias. México: Patria.
- Joyanes Aguilar, L. (2001). Fundamentos de programación. Algoritmos, Estructuras de datos Objetos. (3ra. ed.). España: Mc Graw Hill.
- Aho, A. V., Hopcroft, J. E., Ullman, J. D. (1998). Estructuras de datos y algoritmos. México: Addison – Wesley Longman.

ADA 12 – Construyendo algoritmos

Resuelve los 8 escenarios del nivel basico de Lightbot

- <http://lightbot.com/flash.html>

Resuelve almenos 5 escenarios de la sección “Comandos, automatats, procedimientos”

- <http://pilasbloques.program.ar/online/#/libros/2>

Toma un ScreenShot de los 8 escenarios de Lightbot desbloqueados y súbelo a la plataforma con la siguiente nomenclatura:

Apellidos_Nombre_ADA12_01 ej. RodriguezMedina_Alejandro_ADA12_01

Toma un ScreenShot de cada escenario resuelto de pilasbloques, integralo en un documento pdf y súbelo a la plataforma con la siguiente nomenclatura:

Apellidos_Nombre_ADA12_02 ej. RodriguezMedina_Alejandro_ADA12_02

ADA 13 – Diario reflexivo

Realiza un reporte mencionando los aprendizajes más relevantes que has tenido hasta el momento, para ello responde a las siguientes preguntas:

- ¿Cuáles son las ventajas del uso de los algoritmos en tu vida cotidiana y en la solución de problemas?
- ¿Cuál es el concepto y cuales las características de los algoritmos?
- ¿Cuál es la aplicación de los algoritmos en la solución de problemas.

Guarda tu documento en PDF y súbelo a la plataforma con la siguiente nomenclatura:

Apellidos_Nombre_ADA13 ej. RodriguezMedina_Alejandro_ADA13

Plan de Sesión

3º Semestre de Programación

Unidad 1: Algoritmos
Tema: Diseño modular
Duración: 90 minutos
Mtro. José Contreras

Novena sesión

Objetivo(s):

- Que el estudiante entienda el concepto de diseño modular y refinamiento sucesivo aplicado a algoritmos.

Resumen:

- El diseño de algoritmos en muchas ocasiones requiere que se repita el mismo procedimiento una y otra vez hasta cumplir determinada situación. El diseño modular permitirá crear módulos que puedan ser llamados una y otra vez en el diseño del algoritmo. Para crear módulos óptimos es importante entender y aplicar el refinamiento sucesivo..

Estrategia:

- 05' Síntesis de la sesión anterior
- 45' Lectura 05 y actividad 14
- 15' Compartir con el grupo y resolver los problemas
- 20' Actividad 15
- 05' Síntesis de la sesión.

Actividades de Aprendizaje:

- Realizar las actividades 14 y 15.

Recursos Didácticos:

- Proyector, pizarrón, plumones, internet, equipos de cómputo y/o Smartphone.

Referencias Bibliográficas:

- Elizondo Callejas, R. A. (2010). Informática 2. Serie integral por competencias. México: Patria.
- Joyanes Aguilar, L. (2001). Fundamentos de programación. Algoritmos, Estructuras de datos Objetos. (3ra. ed.). España: Mc Graw Hill.
- Aho, A. V., Hopcroft, J. E., Ullman, J. D. (1998). Estructuras de datos y algoritmos. México: Addison – Wesley Longnan.

ADA 14 – Diseño modular y refinamiento sucesivo

Después de realizar la lectura 05, completa los niveles de procedimientos de:

- <http://lightbot.com/flash.html>

Crea un documento de Word en el cual incluyas un screenshot por cada escenarios de procedimiento resuelto. Guarda tu documento en PDF y súbelo a la plataforma con la siguiente nomenclatura:

Apellidos_Nombre_ADA14 ej. RodriguezMedina_Alejandro_ADA14

ADA 15 – Diario reflexivo

Redacta una reflexión sobre lo que aprendiste sobre diseño modular y refinamiento sucesivo, basándote en las siguientes preguntas:

- ¿Porque crees que sea útil?
- ¿Como podrías utilizarlo en alguna situación de tu vida cotidiana?
- Escribe 5 ejemplos donde creas que se utilicen el diseño modular.

Guarda tu documento en PDF y súbelo a la plataforma con la siguiente nomenclatura:

Apellidos_Nombre_ADA15 ej. RodriguezMedina_Alejandro_ADA15

Lectura 05 – Diseño modular y refinamiento sucesivo

Una computadora no tiene la capacidad para solucionar problemas a menos que se le indiquen los pasos sucesivos a realizar. Estos pasos sucesivos son las instrucciones a ejecutar que constituyen lo que ya conocemos, como el algoritmo.

En la etapa de análisis del proceso se determina **qué** hace el programa y en la etapa de diseño se determina **cómo** hace el programa la tarea solicitada. Existen dos métodos o estrategias en las cuales debemos apoyarnos al momento de resolver problemas complejos; el primero es el diseño descendente o modular y el segundo el refinamiento sucesivo. Explicaremos ambos a continuación.

Diseño Descendente o Modular

Los problemas complejos se pueden resolver más eficazmente cuando se descomponen en subproblemas que sean más fáciles de resolver que el original. Este método se denomina divide y vencerás y consiste en convertir un problema complejo en otros más simples que, una vez resueltos, en su conjunto nos solucionen el original. Al procedimiento de descomposición de un problema en subproblemas más simples, (llamados módulos) para, a continuación, seguir dividiendo estos subproblemas en otros más simples, se le denomina diseño descendente. Las ventajas más importantes de este tipo de diseño son:

1. El problema se comprende más fácilmente al dividirse en partes más simples denominadas módulos o. Adelantemos que cuando demos el salto a la programación, utilizaremos esta idea constantemente, de forma que hablaremos también de procedimientos, o subprogramas.
2. Las modificaciones en los módulos son más fáciles, pues estamos ante algoritmos más sencillos.
3. La comprobación del problema se puede realizar más fácilmente, al poder localizar los posibles fallos con mayor precisión.

Refinamiento por pasos

Normalmente, en el primer diseño del algoritmo nos limitamos a señalar o describir un número reducido de pasos (un máximo de doce aproximadamente) y además incompletos. Tras esta primera descripción, éstos se amplían en una descripción más detallada con más pasos específicos. Este proceso se denomina refinamiento sucesivo del algoritmo. Para problemas complejos se necesitan con frecuencia diferentes niveles de refinamiento antes de que se pueda obtener un algoritmo claro, preciso y completo.

Refinamiento sucesivo, es la metodología por la que, después de un primer esbozo del algoritmo, se expresa con mayor detalle en cada uno de sus pasos. Estos se especifican con mayor minuciosidad, de forma más extensa y con más pasos específicos. En cada nivel de refinamiento hay que considerar dos fases: ¿Qué hace el módulo? para a continuación responder a ¿Cómo lo hace?

Como es natural, dependiendo de la complejidad del problema se necesitarán diferentes y sucesivos niveles de refinamiento antes de que pueda obtenerse un algoritmo con suficiente nivel de detalle.

Así pues, el diseño del algoritmo se descompone en las fases recogidas en la Figura 2.16.

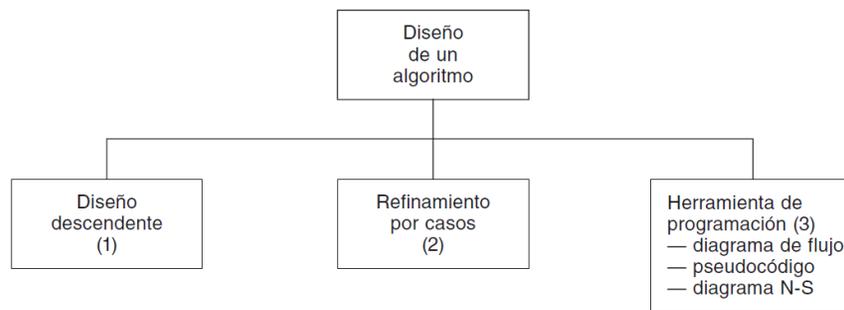


Figura 2.16. Fases del diseño de un algoritmo.

El diseño del algoritmo es un proceso que convierte los resultados del análisis del problema en un diseño modular con refinamientos sucesivos que permitan una posterior traducción a un lenguaje. El diseño del algoritmo es independiente del lenguaje de programación en el que se vaya a codificar posteriormente.

En el ejemplo que vimos, del cálculo de la longitud y superficie de un círculo, a pesar de presentar un bajo nivel de complejidad, en su diseño, se puede descomponer en subproblemas más simples y refinarlos:

1. leer datos de entrada,
2. calcular superficie y longitud,
3. escribir resultados.

Subproblema	Refinamiento
leer radio	leer radio
calcular superficie	superficie = 3.141592 * radio ^ 2
calcular circunferencia	circunferencia = 2 * 3.141592 * radio
escribir resultados	escribir radio, circunferencia, superficie

Consideremos el algoritmo que responde a la pregunta ¿Qué hacer para ver la película de Harry Potter? La respuesta es muy sencilla y puede ser descrita en forma de algoritmo general de modo similar a:

1. ir al cine
2. comprar una entrada (billete o ticket)
3. ver la película
4. regresar a casa

El algoritmo descrito es muy sencillo; sin embargo, como ya se ha indicado en párrafos anteriores, el algoritmo general se descompondrá en pasos más simples en un procedimiento denominado refinamiento sucesivo, ya que cada acción puede descomponerse a su vez en otras acciones simples. Así, por ejemplo, un primer refinamiento del algoritmo ir al cine se puede describir de la forma siguiente:

1. **inicio**
2. ver la cartelera de cines en internet
3. **si** proyectan "Harry Potter" **entonces**
 - 3.1. ir al cine**si_no**
 - 3.2. decidir otra actividad
 - 3.3. ir al paso 7**fin_si**
4. **si** hay fila **entonces**
 - 4.1. ponerse en ella
 - 4.2. **mientras** haya personas delante **hacer**
 - 4.2.1. avanzar en la cola**fin_mientras****fin_si**
5. **si** hay localidades **entonces**
 - 5.1. comprar una entrada
 - 5.2. pasar a la sala
 - 5.3. localizar la(s) butaca(s)
 - 5.4. **mientras** proyectan la película **hacer**
 - 5.4.1. ver la película**fin_mientras**
 - 5.5. abandonar el cine**si_no**
 - 5.6. refunfuñar**fin_si**
6. volver a casa
7. **fin**

Plan de Sesión

3º Semestre de Programación

Unidad 1: Algoritmos
Tema: Diseño modular y refinamiento sucesivo
Duración: 90 minutos
Mtro. José Contreras

Decima sesión

Objetivo(s):

- Que el estudiante aplique el diseño modular y refinamiento sucesivo al diseñar algoritmos que permitan solucionar un problema.

Resumen:

- El diseño de algoritmos en muchas ocasiones requiere que se repita el mismo procedimiento una y otra vez hasta cumplir determinada situación. El diseño modular permitirá crear módulos que puedan ser llamados una y otra vez en el diseño del algoritmo. Para crear módulos óptimos es importante entender y aplicar el refinamiento sucesivo. Durante esta sesión se desarrollará la habilidad para aplicar tanto el diseño modular como el refinamiento sucesivo.

Estrategia:

- 05' Síntesis de la sesión anterior
- 45' Actividad 16
- 15' Compartir con el grupo y resolver los problemas
- 20' Actividad 17
- 05' Síntesis de la sesión.

Actividades de Aprendizaje:

- Realizar las actividades 16 y 17.

Recursos Didácticos:

- Proyector, pizarrón, plumones, internet, equipos de cómputo y/o Smartphone.

Referencias Bibliográficas:

- Elizondo Callejas, R. A. (2010). Informática 2. Serie integral por competencias. México: Patria.
- Joyanes Aguilar, L. (2001). Fundamentos de programación. Algoritmos, Estructuras de datos Objetos. (3ra. ed.). España: Mc Graw Hill.
- Aho, A. V., Hopcroft, J. E., Ullman, J. D. (1998). Estructuras de datos y algoritmos. México: Addison – Wesley Longman.

ADA 16 – Aprestamiento

Desarrollemos los pensamientos algorítmico y procedimental. Entra a la liga <http://plastelina.net/> y crea el algoritmo para resolver los problemas presentados en los siguientes juegos:

- Wolf Sheep & Cabbage
- Cannibals & Missioneries
- Family Crisis
- Elevators Logic
- Frogs Logic

Crea un documento de Word en el cual incluyas un screenshot de los problemas resueltos. Y una reflexión acerca de cómo crees que te ayudaron los temas vistos hasta el momento para poder resolverlos; ¿que crees que hubiese sido diferente sino hubieses tenido conocimiento de los temas?.

Guarda tu documento en PDF y súbelo a la plataforma con la siguiente nomenclatura:

Apellidos_Nombre_ADA16 ej. RodriguezMedina_Alejandro_ADA16

ADA 17 – Siguiendo algoritmos

Resuelve un cubo de rubik sigue el algoritmo que se encuentra en <https://how-to-solve-a-rubix-cube.com/como-resolver-un-cubo-rubik-es/>

Crea un documento de Word en el cual incluyas una selfie con el cubo de rubik resuelto. Incluye una reflexión acerca de cómo fue el proceso de seguir el algoritmo, incluye que fue lo que te resultó más sencillo, que fue lo que te resultó más complicado y que le cambiarías al algoritmo.

Guarda tu documento en PDF y súbelo a la plataforma con la siguiente nomenclatura:

Apellidos_Nombre_ADA17 ej. RodriguezMedina_Alejandro_ADA17

Plan de Sesión

3º Semestre de Programación

Unidad 1: Algoritmos
Tema: Integración de habilidades
Duración: 90 minutos
Mtro. José Contreras

Onceava sesión

Objetivo(s):

- Que el estudiante integre las habilidades que ha desarrollado hasta el momento, en el diseño de algoritmos que permitan solucionar un problema.

Resumen:

- Integrar las habilidades en el diseño de un algoritmo que resuelva una determinada situación, ayudará a los estudiantes a interiorizar sus habilidades y aterrizar sus ideas..

Estrategia:

- 05' Síntesis de la sesión anterior
- 75' Actividad 18
- 10' Compartir con el grupo y cierre de la sesión

Actividades de Aprendizaje:

- Realizar la actividad 18.

Recursos Didácticos:

- Proyector, pizarrón, plumones, internet, equipos de cómputo y/o Smartphone.

Referencias Bibliográficas:

- Elizondo Callejas, R. A. (2010). Informática 2. Serie integral por competencias. México: Patria.
- Joyanes Aguilar, L. (2001). Fundamentos de programación. Algoritmos, Estructuras de datos Objetos. (3ra. ed.). España: Mc Graw Hill.
- Aho, A. V., Hopcroft, J. E., Ullman, J. D. (1998). Estructuras de datos y algoritmos. México: Addison – Wesley Longnan.

ADA 18 – Solución de problemas utilizando algoritmos

Eres el líder de proyecto de la empresa de desarrollo los mejores videojuegos de Internet, a ti y a tu equipo de trabajo, integrado por dos colaboradores, les han pedido desarrollar el análisis del problema y el algoritmo para crear un nuevo videojuego. Por tal motivo, apoyándote en el formato de metodología para la solución de problemas y en lo visto hasta el momento, realizarás el análisis y el diseño del algoritmo para un videojuego.

Incluye una reflexión por cada miembro del equipo, acerca del proceso de desarrollo de la actividad, incluye que fue lo que te resultó más sencillo, que fue lo que te resultó más complicado y que has aprendido hasta el momento.

Guarda tu documento en PDF y súbelo a la plataforma con la siguiente nomenclatura:

Apellidos_Nombre_ADA18 ej. RodriguezMedina_Alejandro_ADA18

Plan de Sesión

3º Semestre de Programación

Unidad 2: Diagramas de flujo y pseudocódigo

Tema: Conceptos básicos

Duración: 90 minutos

Mtro. José Contreras

Doceava sesión

Objetivo(s):

- Que el estudiante defina los conceptos básicos que se utilizan en la representación de algoritmos a través de diagramas de flujo y pseudocódigo.

Resumen:

- Para poder diseñar algoritmos empleando diagramas de flujo y/o pseudocódigo, es fundamental entender los conceptos y símbolos que se utilizan en el proceso.

Estrategia:

- 05' Síntesis de la sesión anterior
- 75' Actividad 19
- 10' Síntesis de la sesión.

Actividades de Aprendizaje:

- Realizar la actividad 19.

Recursos Didácticos:

- Proyector, pizarrón, plumones, internet, equipos de cómputo y/o Smartphone.

Referencias Bibliográficas:

- Joyanes Aguilar, L. (2001). Fundamentos de programación. Algoritmos, Estructuras de datos y Objetos. (3ra. ed.). España: Mc Graw Hill.
- García Carrillo, R. (2003). Técnicas de programación. México: Mc Graw Hill.
- Aho, A. V., Hopcroft, J. E., Ullman, J. D. (1998). Estructuras de datos y algoritmos. México: Addison – Wesley Longman.

ADA 19 – Conceptos básico

Eres parte de un equipo de programadores de una reconocida empresa. A ti y a dos de tus colaboradores de trabajo se les ha encargado explicar a los nuevos reclutas los conceptos básicos de programación. Por tal motivo, apoyados en las lecturas proporcionadas, han decidido preparar una presentación electrónica, para exponer los siguientes conceptos básicos:

1. Tipos de datos, como se representan y ejemplos de su uso en una situación real.
 - Números
 - Enteros
 - Reales
 - Cadena de caracteres
 - Lógicos o Booleanos
 - Falso
 - Verdadero
2. Qué son, como se representan, cuando y para que se utilizan los siguientes elementos, y ejemplos de su uso en una situación real:
 - Variables
 - Identificadores
 - Constantes
 - Contadores
 - Acumuladores
 - Palabras reservadas

Metodología de elaboración:

1. Lee de manera individual.
2. Comparte con tu equipo lo que has entendido.
3. Construyan la presentación con las ideas de todos los integrantes.

Guarda tu documento en formato de presentación electrónica y súbelo a la plataforma con la siguiente nomenclatura:

Apellidos_Nombre_ADA19 ej. RodriguezMedina_Alejandro_ADA19

Plan de Sesión

3º Semestre de Programación

Unidad 2: Diagramas de flujo y pseudocódigo

Tema: Conceptos básicos

Duración: 90 minutos

Mtro. José Contreras

Treceava sesión

Objetivo(s):

- Que el estudiante aterrice las ideas que tiene acerca de los conceptos básicos que se utilizan en la representación de algoritmos, a través de diagramas de flujo y pseudocódigo.

Resumen:

- Para poder diseñar algoritmos empleando diagramas de flujo y/o pseudocódigo, es fundamental entender y aplicar los conceptos y símbolos que se utilizan en el proceso.

Estrategia:

- 05' Síntesis de la sesión anterior
- 65' Actividad 20
- 15' Exposición docente
- 05' Síntesis de la sesión.

Actividades de Aprendizaje:

- Realizar la actividad 20.

Recursos Didácticos:

- Proyector, pizarrón, plumones, internet, equipos de cómputo y/o Smartphone.

Referencias Bibliográficas:

- Joyanes Aguilar, L. (2001). Fundamentos de programación. Algoritmos, Estructuras de datos y Objetos. (3ra. ed.). España: Mc Graw Hill.
- García Carrillo, R. (2003). Técnicas de programación. México: Mc Graw Hill.
- Aho, A. V., Hopcroft, J. E., Ullman, J. D. (1998). Estructuras de datos y algoritmos. México: Addison – Wesley Longman.

ADA 20 – Exposición de conceptos básicos

Llegó el momento de capacitar a los nuevos reclutas. Tu y tus colaboradores deberán pasar a exponer el tema preparado durante la sesión anterior, explicando:

1. Tipos de datos, como se representan y ejemplos de su uso en una situación real.
 - Números
 - Enteros
 - Reales
 - Cadena de caracteres
 - Lógicos o Booleanos
 - Falso
 - Verdadero

2. Qué son, como se representan, cuando y para que se utilizan los siguientes elementos, y ejemplos de su uso en una situación real:
 - Variables
 - Identificadores
 - Constantes
 - Contadores
 - Acumuladores
 - Palabras reservadas

No olvides incluir los ejemplos, ya que son importantes para aterrizar las ideas

Presentación 02 – Conceptos básicos

<h2>UNIDAD 02</h2> <p>Diagramas de Flujo y Pseudocódigo</p>	<h2>CONCEPTOS BÁSICOS DE PROGRAMACIÓN</h2> <ul style="list-style-type: none">• Antes de abordar los temas de Diagrama de Flujo y Pseudocódigo es importante saber algunos conceptos básicos de programación y el encargado de explicarnos estos conceptos está FBI.• Sigue las instrucciones que encontrarás en la plataforma y prepara una presentación para explicar a tus compañeros los conceptos básicos de programación. ADA 01. 	<h2>TIPOS DE DATOS</h2> <ul style="list-style-type: none">• Números<ul style="list-style-type: none">• Enteros• Reales• Cadenas de caracteres• Lógicos o Booleanos<ul style="list-style-type: none">• Falso• Verdadero
<h2>CONCEPTOS BÁSICOS DE PROGRAMACIÓN</h2> <ul style="list-style-type: none">• Cuando alguien se te acerca y te pregunta como te llamas, tu número de teléfono, donde vives, ¿Cómo sabes tu esa información?• Los datos a procesar por una computadora, deben almacenarse en casillas o celdas de memoria para su posterior utilización.• Estas casillas de memoria tienen un nombre que permite su identificación: Identificador, nombre o etiqueta• El valor de las celdas de memoria puede ser variable o constante	<h2>VARIABLES</h2> <ul style="list-style-type: none">• ¿Que son Variables?<ul style="list-style-type: none">• Las variables son objetos que pueden cambiar su valor durante la ejecución.• ¿Que almacenan las variables?<ul style="list-style-type: none">• Diferentes tipos de datos (Existen tantos tipos de variables como tipos de datos)• ¿Como se nombra una variable?<ul style="list-style-type: none">• Utilizando identificadores	<h2>IDENTIFICADORES</h2> <ul style="list-style-type: none">• ¿Qué son los identificadores?<ul style="list-style-type: none">• Son nombres que se dan a los elementos (variables, constantes y procedimientos) utilizados para resolver un problema y poder diferenciar unos de otros.• Al asignar nombres de identificadores se deben tener en cuenta algunas reglas. ¿Cuáles son?
<h2>CONSTANTES</h2> <ul style="list-style-type: none">• ¿Qué son?<ul style="list-style-type: none">• Son datos que no cambian durante la ejecución• ¿Cuándo se utilizan?<ul style="list-style-type: none">• Cuando se tiene un valor que se sabe no cambiará durante la ejecución de un programa. Ej. Pi=3.1416	<h2>CONTADORES</h2> <ul style="list-style-type: none">• ¿Qué son?<ul style="list-style-type: none">• Es una variable que va aumentando para llevar el control de ejecución de un proceso o procedimiento.• ¿Cuándo se utilizan?<ul style="list-style-type: none">• Cuando se desea saber cuantas veces se ha realizado un proceso o procedimiento.	<h2>PALABRAS RESERVADAS</h2> <ul style="list-style-type: none">• ¿Que son?<ul style="list-style-type: none">• Son palabras que no se pueden utilizar para nombrar variables, constante o procedimientos, porque son de uso exclusivo del lenguaje de programación.

Plan de Sesión

3º Semestre de Programación

Unidad 2: Diagramas de flujo y pseudocódigo

Tema: Operadores y operandos

Duración: 90 minutos

Mtro. José Contreras

Catorceava sesión

Objetivo(s):

- Que el estudiante identifique los diferentes tipos de operadores y operandos que se utilizan para crear una expresión.
- Que el estudiante resuelva asignaciones y expresiones utilizadas en la solución de problemas.

Resumen:

- Las asignaciones y expresiones creadas utilizando operadores y operandos, permitirán resolver los problemas y/o automatizar las soluciones.

Estrategia:

- 05' Síntesis de la sesión anterior
- 75' Actividad 21
- 10' Síntesis de la sesión.

Actividades de Aprendizaje:

- Realizar la actividad 21.

Recursos Didácticos:

- Proyector, pizarrón, plumones, internet, equipos de cómputo y/o Smartphone.

Referencias Bibliográficas:

- Joyanes Aguilar, L. (2001). Fundamentos de programación. Algoritmos, Estructuras de datos y Objetos. (3ra. ed.). España: Mc Graw Hill.
- García Carrillo, R. (2003). Técnicas de programación. México: Mc Graw Hill.
- Aho, A. V., Hopcroft, J. E., Ullman, J. D. (1998). Estructuras de datos y algoritmos. México: Addison – Wesley Longman.

ADA 21 – Operadores y operandos

Continuas con la capacitación de los nuevos reclutas y en esta ocasión deberás explicarles acerca de operadores, operandos y expresiones. Por tal motivo, apoyados en las lecturas proporcionadas, tu y tus dos colaboradores han decidido preparar una presentación electrónica, para exponer los siguientes conceptos:

1. Que y cuales son los operandos y los operadores, para que sirven, como se representan, su prioridad. Incluir ejemplos de su uso en la vida cotidiana.

- Aritméticos:
- Relacionales:
- Alfanuméricos:
- Lógicos:

2. Que es una asignación y que son las expresiones. Incluir ejemplos de su uso en la vida cotidiana.

Metodología de elaboración:

1. Lee de manera individual.
2. Comparte con tu equipo lo que has entendido.
3. Construyan la presentación con las ideas de todos los integrantes.

Guarda tu documento en formato de presentación electrónica y súbelo a la plataforma con la siguiente nomenclatura:

Apellidos_Nombre_ADA21 ej. RodriguezMedina_Alejandro_ADA21

Nota: Aclara tus dudas con el docente.

Plan de Sesión

3º Semestre de Programación

Unidad 2: Diagramas de flujo y pseudocódigo

Tema: Operadores y operandos

Duración: 90 minutos

Mtro. José Contreras

Quinceava sesión

Objetivo(s):

- Que el estudiante aterrice las ideas que tiene acerca de los operadores, operandos, asignaciones y expresiones, que se utilizan en la representación de algoritmos, a través de diagramas de flujo y pseudocódigo.

Resumen:

- Para poder diseñar algoritmos empleando diagramas de flujo y/o pseudocódigo, es fundamental entender y aplicar los operadores, operandos, asignaciones y expresiones, que se utilizan en el proceso.

Estrategia:

- 05' Síntesis de la sesión anterior
- 65' Actividad 22
- 15' Exposición docente
- 05' Síntesis de la sesión.

Actividades de Aprendizaje:

- Realizar la actividad 22

Recursos Didácticos:

- Proyector, pizarrón, plumones, internet, equipos de cómputo y/o Smartphone.

Referencias Bibliográficas:

- Joyanes Aguilar, L. (2001). Fundamentos de programación. Algoritmos, Estructuras de datos y Objetos. (3ra. ed.). España: Mc Graw Hill.
- García Carrillo, R. (2003). Técnicas de programación. México: Mc Graw Hill.
- Aho, A. V., Hopcroft, J. E., Ullman, J. D. (1998). Estructuras de datos y algoritmos. México: Addison – Wesley Longman.

ADA 22 – Exposición de operadores y operandos

Llegó el momento de capacitar a los nuevos reclutas. Tú y tus colaboradores deberán pasar a exponer el tema preparado durante la sesión anterior, explicando:

3. Que y cuales son los operandos y los operadores, para que sirven, como se representan, su prioridad. Incluir ejemplos de su uso en la vida cotidiana.

- Aritméticos:
- Relacionales:
- Alfanuméricos:
- Lógicos:

4. Que es una asignación y que son las expresiones. Incluir ejemplos de su uso en la vida cotidiana.

No olvides incluir los ejemplos, ya que son importantes para aterrizar las ideas

Presentación 03 – Operadores y operandos

UNIDAD 02

Diagramas de Flujo y Pseudocódigo

REPASO

- Conceptos básicos

OPERADORES Y OPERANDOS

- Las siguientes operaciones y expresiones son las únicas permitidas en la realización de un algoritmo:

OPERADORES ARITMÉTICOS

Operación	Operador	Ejemplo
Suma	+	a+b
Resta	-	a-b
Unario	- (cambio de signo)	-a
Multiplicación	*	a*b
División	/	a/b
Módulo (resto de división entera)	%	a%b

NÚMEROS ENTERO Y REALES

La división entera es diferente a la división de reales

Si los dos operandos son enteros el resultado es un entero

Si los dos operandos son reales el resultado es un real

Si uno de los dos operandos es real el resultado es real

Ej: Para a y b enteros con valores: a=7 y b=2
Para c y d reales con valores: c=9 y d=2

Operación	Resultado	Tipo
a/b	4	entero
c/d	4.5	real
a/d	4.5	real
a%b	1	entero

PRIORIDAD DE OPERADORES

Operador
=
*, /, %
+, -

↓ MAYOR PRIORIDAD

↓ MENOR PRIORIDAD

OPERADORES RELACIONALES

Permiten comparar dos datos del mismo tipo para obtener un resultado lógico: V o F

Los datos pueden ser variables o constantes de tipo numérico o carácter

Operación	Operador	Ejemplo	Resultado
Igual que	==	9==9	V
Diferente de	!=	9!=10	V
Menor que	<	7<15	V
Mayor que	>	22.5>11.8	V
Menor o igual que	<=	15<=15	V
Mayor o igual que	>=	35<=40	F

OPERADORES LÓGICOS

Permiten operar con valores lógicos para obtener un resultado también lógico

Operación	Operador	Ejemplo
Negación	NO	NO a
Conjunción	Y	A Y b
Disyunción	O	A O b

↓ MAYOR P

↓ MENOR P

OPERADORES LÓGICOS

Valor a	Resultado
NO a	
V	F
F	V

OPERADORES LÓGICOS

a Y b		
Valor a	Valor b	Resultado
V	V	V
V	F	F
F	V	F
F	F	F

OPERADORES LÓGICOS

a O b		
Valor a	Valor b	Resultado
V	V	V
V	F	V
F	V	V
F	F	F

EXPRESIONES

- Una expresión es cualquier combinación de operadores variables y constantes. Devuelve un valor de un tipo determinado
- Ejemplos:
 - (b * h) / 2
 - max
 - A > B and A < C
 - 23 + 35 * 2

EXPRESIONES

9 + 7 * 8 - 36 / 5	2 * 5 + 7.8 / 4 <= 2.5 / 6
9 + 36 - 30 / 2	10 + 7.8 / 4 <= 2.5 / 6
9 + 5 * 7 (añadido entero)	10 + 1.95 <= 0.41
65 - 7	11.95 <= 0.41
58	F
8 % 2 * 3 + 1	
8 % 6 + 1	
2 + 1	
3	

ASIGNACIÓN

- Sintaxis: 'variable' = 'expresión'

 - Primero se evalúa la expresión
 - Después se asigna el valor a la variable
 - El valor anterior de la variable se pierde/olvida

- Ejemplos:
 - intArea = (b * h) / 2
 - intTotal = 23 + 35 * 2

Plan de Sesión

3º Semestre de Programación

Unidad 2: Diagramas de flujo y pseudocódigo

Tema: Asignaciones y expresiones

Duración: 90 minutos

Mtro. José Contreras

Dieciseisava sesión

Objetivo(s):

- Que el estudiante resuelva asignaciones y expresiones, que se utilizan en la representación de algoritmos, a través de diagramas de flujo y pseudocódigo.
- Que el estudiante aprenda a utilizar diagramas de flujo y pseudocódigo para poder representar algoritmos.

Resumen:

- Los dos lenguajes más utilizados para representar algoritmos son los diagramas de flujo y el pseudocódigo, saber escribirlos será indispensable para diseñar soluciones en programación. Para poder escribir algoritmos empleando diagramas de flujo y/o pseudocódigo, es fundamental saber representar y resolver las asignaciones y expresiones, que se utilizan en el proceso.

Estrategia:

- 05' Síntesis de la sesión anterior
- 15' Actividad 23
- 10' Resolución grupal
- 35' Actividad 24
- 20' Exposición docente
- 05' Síntesis de la sesión.

Actividades de Aprendizaje:

- Realizar la actividad 23

Recursos Didácticos:

- Proyector, pizarrón, plumones, internet, equipos de cómputo y/o Smartphone.

Referencias Bibliográficas:

- Joyanes Aguilar, L. (2001). Fundamentos de programación. Algoritmos, Estructuras de datos y Objetos. (3ra. ed.). España: Mc Graw Hill.
- García Carrillo, R. (2003). Técnicas de programación. México: Mc Graw Hill.
- Aho, A. V., Hopcroft, J. E., Ullman, J. D. (1998). Estructuras de datos y algoritmos. México: Addison – Wesley Longman.

ADA 23 – Asignaciones y expresiones

Resuelve los siguientes ejercicios:

$7/2$		$2.1*(1.5+3.0* 4.1)$	
$12 \bmod 7$		$12-3*4+25/(2+3)$	
$4+2*5$		$4^2*2/(12/4)$	
$23*2/5$		$6*4/8+5$	
$3+5*(10-(2+4))$		$45/((28-((6*8-12)/3))-9)$	
$3.5+5.09-140/40$		$(22-24/2)*6/12$	
Si a = 10 b = 20 c = 30			
Expresión	T / F	Expresión	T / F
$a+b>c$		$6*4/8+5=8$	
$a-b<c$		"Hola" = "Hello"	
$a-b=c$		$c>30$	
$a*b<>c$		$b>20$	
Si a = 10 b = 20 c = 30			
Expresión	T / F	Expresión	T / F
$(a<b) \text{ and } (b<c)$		$\text{Not}(4+(18/6)=7) \text{ and } c=30$	
$(a>b) \text{ or } (c>b)$		$\text{Not}(4+(28/7)=9) \text{ or } c=a$	
$\text{Not}(a>b) \text{ and } (c>b)$		$25/5*(4-2)=a \text{ or } b=c$	
$\text{Not}(c>30)$		$\text{Not}(12*2-6=(32/8+2)*3)$	

Incluye una reflexión acerca del proceso de desarrollo de la actividad, que fue lo que te resultó más sencillo, que fue lo que te resultó más complicado y que has aprendido hasta el momento.

Guarda tu documento en PDF y súbelo a la plataforma con la siguiente nomenclatura:

Apellidos_Nombre_ADA23 ej. RodriguezMedina_Alejandro_ADA23

ADA 24 – Diagramas de Flujo y Pseudocódigo

En equipos de 3 personas, después de realizar las lecturas referentes a “Diagramas de Flujo y Pseudocódigo”, redacten en una presentación electrónica tres ejemplos de problemas y su respectiva algoritmo de solución, representado con diagramas de flujo y con pseudocódigo, así como sus conclusiones resaltando lo más importante, que fue lo que te resultó más sencillo, que fue lo que te resultó más complicado y que has aprendido hasta el momento.

Guarda tu documento en formato de presentación electrónica y súbelo a la plataforma con la siguiente nomenclatura:

Apellidos_Nombre_ADA24 ej. RodriguezMedina_Alejandro_ADA24

Presentación 04 – Asignaciones y expresiones

UNIDAD 02
Diagramas de flujo y Pseudocódigo

1

REPASO

• Conceptos básicos

2

TIEMPO DE DEMOSTRAR TUS HABILIDADES

• Ejercicios con operadores y variables. Realiza la actividad 26, guárdala en pdf

3

$7/2$	$2.1*(1.5+3.0*4.1)$
$12 \bmod 7$	$12-3*4+25/(2+3)$
$4+2*5$	$4*2/2/(12/4)$
$23*2/5$	$6*4/8+5$
$3+5*(10-(2+4))$	$45/((28-((6*9-12)/3)-9)$
$3.5+5.99-140/40$	$(22-24/2)*6/12$

4

Si a = 10 b = 20 c = 30

Expresión	T / F	Expresión	T / F
$a+b>c$		$6+4/8+5=8$	
$a-b<c$		"Hola" = "Hello"	
$a-b=c$		$c>30$	
$a*b<>c$		$b>20$	

5

Si a = 10 b = 20 c = 30

Expresión	T / F	Expresión	T / F
$(a<b) \text{ and } (b<c)$		$\text{Not}(4+(18/6)=7) \text{ and } c=30$	
$(a>b) \text{ or } (c>b)$		$\text{Not}(4+(28/7)=9) \text{ or } c=a$	
$\text{Not}((a>b) \text{ and } (c>b))$		$25/5*(4-2)=a \text{ or } b=c$	
$\text{Not}(c>30)$		$\text{Not}(12*2.6=(32/8+2)*3)$	

6

Plan de Sesión

3º Semestre de Programación

Unidad 2: Diagramas de flujo y pseudocódigo

Tema: Diseño de pseudocódigo

Duración: 90 minutos

Mtro. José Contreras

Diecisieteava sesión

Objetivo(s):

- Que el estudiante desarrolle la habilidad de resolver problemas utilizando pseudocódigo, para representar algoritmos.

Resumen:

- Desarrollar la habilidad de entender un problema y plantear la solución a través de un algoritmo representado por medio de pseudocódigo, será indispensable para diseñar soluciones en programación.

Estrategia:

- 05' Síntesis de la sesión anterior
- 20' Actividad 25
- 10' Compartir la actividad con el grupo
- 40' Actividad 26
- 10' Compartir la actividad con el grupo
- 05' Síntesis de la sesión.

Actividades de Aprendizaje:

- Realizar la actividad 25

Recursos Didácticos:

- Proyector, pizarrón, plumones, internet, equipos de cómputo y/o Smartphone.

Referencias Bibliográficas:

- Joyanes Aguilar, L. (2001). Fundamentos de programación. Algoritmos, Estructuras de datos y Objetos. (3ra. ed.). España: Mc Graw Hill.
- García Carrillo, R. (2003). Técnicas de programación. México: Mc Graw Hill.
- Aho, A. V., Hopcroft, J. E., Ullman, J. D. (1998). Estructuras de datos y algoritmos. México: Addison – Wesley Longman.

ADA 25 – Pseudocódigo

Es momento de resolver los primeros problemas utilizando pseudocódigo. De manera individual resuelve los siguientes problemas utilizando pseudocódigo.

- **Problema:** Leer dos números reales y escribir la suma.
- **Problema:** Leer dos números reales y escribir la media aritmética.
- **Problema:** Leer un número entero x y calcular $y=x^3$. Escribir el resultado.
- **Problema:** Hallar el valor absoluto de un número x .
- **Problema:** Leer un número entero x y calcular $y=|x|^3$. Escribir el resultado.

Guarda tu documento en formato PDF y súbelo a la plataforma con la siguiente nomenclatura:

Apellidos_Nombre_ADA25 ej. RodriguezMedina_Alejandro_ADA25

ADA 26 – Solución de problemas utilizando pseudocódigo

En parejas, escriban el pseudocódigo para los siguientes problemas planteados:

1. Suponga que un individuo desea invertir su capital en un banco y desea saber cuánto dinero ganara después de un mes si el banco paga a razón de 2% mensual.
2. Un vendedor recibe un sueldo base más un 10% extra por comisión de sus ventas, el vendedor desea saber cuánto dinero obtendrá por concepto de comisiones por las tres ventas que realiza en el mes y el total que recibirá en el mes tomando en cuenta su sueldo base y comisiones.
3. Una tienda ofrece un descuento del 15% sobre el total de la compra y un cliente desea saber cuánto deberá pagar finalmente por su compra.
4. Un aprendiz desea saber cuál será su calificación final en la materia de Algoritmos. Dicha calificación se compone de los siguientes porcentajes:
 - 55% del promedio de sus tres calificaciones parciales.
 - 30% de la calificación del examen final.
 - 15% de la calificación de un trabajo final.
5. Un maestro desea saber qué porcentaje de hombres y que porcentaje de mujeres hay en un grupo de estudiantes.
6. Dada una cantidad en pesos, obtener la equivalencia en dólares, asumiendo que el tipo de cambio es variable.

7. Calcular el nuevo salario de un obrero si obtuvo un incremento del 25% sobre su salario anterior.
8. En un hospital existen tres áreas: Ginecología, Pediatría, Traumatología. El presupuesto anual del hospital se reparte conforme a la sig. tabla:

Área Porcentaje del presupuesto

- Ginecología 40%
- Traumatología 30%
- Pediatría 30%

Obtener la cantidad de dinero que recibirá cada área, para cualquier monto presupuestal.

9. El dueño de una tienda compra un artículo a un precio determinado. Obtener el precio en que lo debe vender para obtener una ganancia del 30%.
10. Tres personas deciden invertir su dinero para fundar una empresa. Cada una de ellas invierte una cantidad distinta. Obtener el porcentaje que cada quien invierte con respecto a la cantidad total invertida.

Plan de Sesión

3º Semestre de Programación

Unidad 2: Diagramas de flujo y pseudocódigo

Tema: Diseño de diagramas de flujo

Duración: 90 minutos

Mtro. José Contreras

Dieciochoava sesión

Objetivo(s):

- Que el estudiante desarrolle la habilidad para escribir diagramas de flujo que representen un algoritmo que soluciona un problema.

Resumen:

- Desarrollar la habilidad de representar algoritmos utilizando diagramas de flujo, será indispensable para diseñar soluciones en programación.

Estrategia:

- 05' Síntesis de la sesión anterior
- 55' Actividad 27
- 25' Compartir la actividad con el grupo
- 05' Síntesis de la sesión.

Actividades de Aprendizaje:

- Realizar la actividad 27

Recursos Didácticos:

- Proyector, pizarrón, plumones, internet, equipos de cómputo y/o Smartphone.

Referencias Bibliográficas:

- Joyanes Aguilar, L. (2001). Fundamentos de programación. Algoritmos, Estructuras de datos y Objetos. (3ra. ed.). España: Mc Graw Hill.
- García Carrillo, R. (2003). Técnicas de programación. México: Mc Graw Hill.
- Aho, A. V., Hopcroft, J. E., Ullman, J. D. (1998). Estructuras de datos y algoritmos. México: Addison – Wesley Longnan.

ADA 27 – Diagramas de flujo

Es momento de resolver los primeros problemas utilizando diagramas de flujo.

- **Problema:** Leer dos números reales y escribir la suma.
- **Problema:** Leer dos números reales y escribir la media aritmética.
- **Problema:** Leer un número entero x y calcular $y=x^3$. Escribir el resultado.
- **Problema:** Hallar el valor absoluto de un número x .
- **Problema:** Leer un número entero x y calcular $y=|x|^3$. Escribir el resultado.
- **Problema:** Leer un número real del teclado. Calcular el valor de "q". Teniendo en cuenta que si el valor leído se encuentra en el intervalo $(0,10]$, el resultado que se le asigna a "q" es uno. En caso contrario "q" toma el valor de cero. Escribir el resultado.
- **Problema:** Diseñar un algoritmo para calcular y escribir el cuadrante al que pertenece un punto (x, y) en el plano, cuyas coordenadas se proporcionan. El resultado ha de ser un número entero r de 1 a 4 indicando el cuadrante.

Guarda tu documento en formato PDF y súbelo a la plataforma con la siguiente nomenclatura:

Apellidos_Nombre_ADA27 ej. RodriguezMedina_Alejandro_ADA27

Plan de Sesión

3º Semestre de Programación

Unidad 2: Diagramas de flujo y pseudocódigo
Tema: Solución de problemas utilizando diagramas de flujo
Duración: 90 minutos
Mtro. José Contreras

Diecinueveava sesión

Objetivo(s):

- Que el estudiante desarrolle la habilidad de resolver problemas utilizando pseudocódigo, para representar algoritmos.

Resumen:

- Desarrollar la habilidad de entender un problema y plantear la solución a través de un algoritmos representado por medio de pseudocódigo, será indispensable para diseñar soluciones en programación.

Estrategia:

- 05' Síntesis de la sesión anterior
- 55' Actividad 28
- 25' Compartir la actividad con el grupo
- 05' Síntesis de la sesión.

Actividades de Aprendizaje:

- Realizar la actividad 28

Recursos Didácticos:

- Proyector, pizarrón, plumones, internet, equipos de cómputo y/o Smartphone.

Referencias Bibliográficas:

- Joyanes Aguilar, L. (2001). Fundamentos de programación. Algoritmos, Estructuras de datos y Objetos. (3ra. ed.). España: Mc Graw Hill.
- García Carrillo, R. (2003). Técnicas de programación. México: Mc Graw Hill.
- Aho, A. V., Hopcroft, J. E., Ullman, J. D. (1998). Estructuras de datos y algoritmos. México: Addison – Wesley Longman.

ADA 28 – Solución de problemas utilizando diagramas de flujo

Realiza los diagramas de flujo para resolver los siguientes problemas planteados

1. En un almacén se hace un 20% de descuento a los clientes cuya compra supere los \$1000
¿Cuál será la cantidad que pagara una persona por su compra?
2. Un obrero necesita calcular su salario semanal, el cual se obtiene de la sig. manera:
 - Si trabaja 40 horas o menos se le paga \$16 por hora
 - Si trabaja más de 40 horas se le paga \$16 por cada una de las primeras 40 horas y \$20 por cada hora extra.
3. Hacer un algoritmo que imprima el nombre de un artículo, clave, precio original y su precio con descuento. El descuento lo hace con base en la clave. Si la clave es 01 el descuento es del 10% y si la clave es 02 el descuento en del 20% (solo existen dos claves).
4. Hacer un algoritmo que calcule el total a pagar por la compra de camisas. Si se compran tres camisas o más se aplica un descuento del 20% sobre el total de la compra y si son menos de tres camisas un descuento del 10%.
5. Una empresa quiere hacer una compra de varias piezas de la misma clase a una fábrica de refacciones. La empresa, dependiendo del monto total de la compra, decidirá que hacer para pagar al fabricante.
 - Si el monto total de la compra excede de \$500 000 la empresa tendrá la capacidad de invertir de su propio dinero un 55% del monto de la compra, pedir prestado al banco un 30% y el resto lo pagará solicitando un crédito al fabricante.
 - Si el monto total de la compra no excede de \$500 000 la empresa tendrá capacidad de invertir de su propio dinero un 70% y el restante 30% lo pagará solicitando crédito al fabricante.
 - El fabricante cobra por concepto de intereses un 20% sobre la cantidad que se le pague a crédito.

Guarda tu documento en formato PDF y súbelo a la plataforma con la siguiente nomenclatura:

Apellidos_Nombre_ADA31 ej. RodriguezMedina_Alejandro_ADA31

Plan de Sesión

3º Semestre de Programación

Unidad 3: Estructuras de control
Tema: Estructuras secuenciales y selectivas
Duración: 90 minutos
Mtro. José Contreras

Veinteava sesión

Objetivo(s):

- Que el estudiante identifique las estructuras de control secuenciales y selectivas, así como el momento de emplear cada una en el diseño de algoritmos, para la solución de problemas.

Resumen:

- En el diseño de algoritmos para la solución de problemas, se emplean tanto estructuras secuenciales como selectivas, identificar cada una de ellas y saber elegir el momento y situación adecuados para emplearlas, será fundamental en un contexto de programación.

Estrategia:

- 05' Síntesis de la sesión anterior
- 45' Actividad 29
- 25' Compartir la actividad con el grupo
- 10' Ejercicios
- 05' Síntesis de la sesión.

Actividades de Aprendizaje:

- Realizar la actividad 29

Recursos Didácticos:

- Proyector, pizarrón, plumones, internet, equipos de cómputo y/o Smartphone.

Referencias Bibliográficas:

ADA 29 – Estructuras secuenciales y selectivas

Tú y tu equipo de trabajo son los expertos en estructuras secuenciales y selectivas, los otros programadores de otras áreas les han solicitado que les expliquen en que consisten dichas estructuras. Por tal motivo, tu y los otros dos integrantes de tu equipo, han decidido apoyarse en una presentación electrónica en la cual incluirán lo siguiente:

1. ¿Qué es el flujo de un programa?
2. ¿Cuál es el orden común de las instrucciones y en qué consiste?
3. ¿Qué es un bloque de instrucciones?
4. ¿Cuáles son las dos estructuras que cambian el flujo de control de instrucciones?
5. ¿Como funcionan las estructuras condicionales? y ejemplos.
 - SI
 - Si-Sino
 - Si-Sino-Si
 - Switch

Utilizando estructuras secuenciales y de selección, diseña el pseudocódigo y el diagrama de flujo para resolver el siguiente problema:

- Saber si un número leído, diferente de cero, es positivo o negativo. Escribirlo en pantalla.

Plan de Sesión

3º Semestre de Programación

Unidad 3: Estructuras de control

Tema: Estructuras repetitivas

Duración: 90 minutos

Mtro. José Contreras

Veintiunava sesión

Objetivo(s):

- Que el estudiante identifique las estructuras de control repetitivas, así como el momento de emplear cada una de ellas en el diseño de algoritmos, para la solución de problemas.

Resumen:

- En el diseño de algoritmos para la solución de problemas, se emplean estructuras repetitivas, identificar cada una de ellas y saber elegir el momento y situación adecuados para emplearlas, será fundamental para desarrollar algoritmos eficaces.

Estrategia:

- 05' Síntesis de la sesión anterior
- 50' Actividad 30
- 30' Compartir la actividad con el grupo
- 05' Síntesis de la sesión.

Actividades de Aprendizaje:

- Realizar la actividad 30

Recursos Didácticos:

- Proyector, pizarrón, plumones, internet, equipos de cómputo y/o Smartphone.

Referencias Bibliográficas:

ADA 30 – Estructuras repetitivas o cíclicas

Tú y tu equipo de trabajo también son los expertos en estructuras repetitivas, los otros programadores de otras áreas les han solicitado que les expliquen en que consisten dichas estructuras. Por tal motivo, tu y los otros dos integrantes de tu equipo, han decidido apoyarse en una presentación electrónica en la cual incluirán lo siguiente:

1. ¿Qué es una estructura?
2. ¿Qué significa cíclicas?
3. Qué son las estructuras cíclicas en los algoritmos
4. Como se utilizan las variables contador dentro de la estructura cíclica
5. ¿Como funcionan las estructuras condicionales? y ejemplos.
 - Hacer - Mientras
 - Hacer - Para
 - Repetir Hasta

Utilizando estructuras secuenciales y de selección, diseña el pseudocódigo y el diagrama de flujo para resolver el siguiente problema:

- Leer 10 números y decir cuántos son positivos y cuantos negativos.

Plan de Sesión

3º Semestre de Programación

Unidad 3: Estructuras de control
Tema: Estructuras repetitivas
Duración: 90 minutos
Mtro. José Contreras

Veintidosava sesión

Objetivo(s):

- Que el estudiante desarrolle su habilidad en el uso de estructuras de control repetitivas, identificando el momento de emplear cada una de ellas en el diseño de algoritmos.

Resumen:

- En el diseño de algoritmos para la solución de problemas, se emplean estructuras repetitivas, desarrollar la habilidad para crearlas, identificar cada una de ellas y saber elegir el momento y situación adecuados para emplearlas, será fundamental para desarrollar algoritmos eficaces.

Estrategia:

- 05' Síntesis de la sesión anterior
- 50' Actividad 31
- 30' Compartir la actividad con el grupo
- 05' Síntesis de la sesión.

Actividades de Aprendizaje:

- Realizar la actividad 31

Recursos Didácticos:

- Proyector, pizarrón, plumones, internet, equipos de cómputo y/o Smartphone.

Referencias Bibliográficas:

ADA 31 – Estructuras de control repetitivas

En parejas resuelve los siguientes problemas utilizando la estructura de control repetitiva que mejor se adapte a la situación.

1. Calcular el promedio de un alumno que tiene 7 calificaciones en la materia de Diseño Estructurado de Algoritmos.
2. Leer 10 números e imprimir solamente los números positivos
3. Leer 20 números e imprimir cuantos son positivos, cuantos negativos y cuantos neutros
4. Determinar cuántos hombres y cuantas mujeres se encuentran en un grupo de n personas, suponiendo que los datos son extraídos alumno por alumno.
5. Suponga que se tiene las calificaciones de un grupo de 40 alumnos. Realizar un algoritmo para calcular la calificación media y la calificación más baja de todo el grupo
6. Calcular e imprimir la tabla de multiplicar de un número cualquiera. Imprimir el multiplicando, el multiplicador y el producto.
7. Una persona debe realizar un muestreo con 50 personas para determinar el promedio de peso de los niños, jóvenes, adultos y viejos que existen en su zona habitacional. Se determinan las categorías con base en la siguiente tabla:

Categoría edad

- Niños 0 - 12
 - Jóvenes 13 - 29
 - Adultos 30 - 59
 - Viejos 60 en adelante
8. Al cerrar un expendio de naranjas, 15 clientes que aún no han pagado recibirán un 15% de descuento si compran más de 10 kilos. Determinar cuánto pagará cada cliente y cuanto percibirá la tienda por esas compras
 9. En una empresa se requiere calcular el salario semanal de cada uno de los n obreros que laboran en ella. El salario se obtiene de la siguiente forma: Si el obrero trabaja 40 horas o menos se le paga \$20 por hora Si trabaja más de 40 horas se le paga \$20 por cada una de las primeras 40 horas y \$25 por cada hora extra.
 10. Un grupo de 100 estudiantes presentan un examen de Física. Diseñe un programa que lea por cada estudiante la calificación obtenida y calcule e imprima:
 - La cantidad de estudiantes que obtuvieron una calificación menor a 50.
 - La cantidad de estudiantes que obtuvieron una calificación de 50 o más pero menor que 70.
 - La cantidad de estudiantes que obtuvieron una calificación de 70 o más pero menor que 80.
 - La cantidad de estudiantes que obtuvieron una calificación de 80 o más.

Plan de Sesión

3º Semestre de Programación

Unidad 3: Estructuras de control
Tema: Proyecto integrador
Duración: 90 minutos
Mtro. José Contreras

Veintitresava sesión

Objetivo(s):

- Que el estudiante integre las habilidades obtenidas durante el semestre en un proyecto en el cual aplique cada uno de los temas vistos.

Resumen:

- Aplicar los conocimientos obtenidos, las habilidades desarrolladas y la actitud adquirida en un proyecto integrador para resolver una situación que se podría presentar en la vida real, es fundamental aterrizar los temas vistos, reflexionar y generar un aprendizaje significativo.

Estrategia:

- 05' Síntesis de la sesión anterior
- 80' Actividad 32 – Proyecto integrador
- 05' Síntesis de la sesión.

Actividades de Aprendizaje:

- Realizar la actividad 32

Recursos Didácticos:

- Proyector, pizarrón, plumones, internet, equipos de cómputo y/o Smartphone.

Referencias Bibliográficas:

ADA 32 – Proyecto Semestral

Tienes una empresa de desarrollo de software en crecimiento y un empresario muy importante te ha pedido que desarrolles un juego para su hijo en Scratch. Por tal motivo has decidido utilizar todas tus habilidades, para desarrollar el mejor juego posible siguiendo ciertas especificaciones, mismas que se encuentran señaladas en la guía de evaluación.

El documento a entregar deberá contener los siguiente:

- El planteamiento del juego como una problemática
- Seguir la metodología de solución de problemas planteando cada paso para resolverlo.
- Los algoritmos representados con pseudocódigo, resaltando las estructuras selectivas y de control.
- Los algoritmos representados con diagramas de flujo, resaltando las estructuras selectivas y de control.
- El juego desarrollado en el programa scratch.

Al concluir el juego, deberás compartir la liga de ingreso con tus colegas en la sección de novedades y adjuntarla en este espacio para entregarla.

No olvides revisar la guía de evaluación para saber los criterios de evaluación.

Guía de observación – Proyecto semestral

Criterios del contenido del documento	5	4	3	2	1
1. Plantea el escenario del juego como una problemática a resolver.					
2. Establece las restricciones.					
3. Indica la mecánica del juego					
4. El juego funciona					
Criterios del código					
5. Utiliza al menos dos variables					
6. Utiliza al menos una estructura iterativa de control					
7. Utiliza al menos dos estructuras de condición					
8. Agrega sonidos					
9. Cuando el juego termina, enseña un mensaje de “Game over”					
10. El juego siempre empieza con el personaje en la misma posición					
11. Enseña la puntuación del juego					
12. Despliega mensajes de texto, de acuerdo al evento					
13. Pregunta el nombre del jugador					
14. Al fina enseña la puntuación obtenida					

Plan de Sesión

3º Semestre de Programación

Unidad 3: Estructuras de control
Tema: Proyecto integrador
Duración: 90 minutos
Mtro. José Contreras

Veinticuatroava sesión

Objetivo(s):

- Que el estudiante integre las habilidades obtenidas durante el semestre en un proyecto en el cual aplique cada uno de los temas vistos.

Resumen:

- Aplicar los conocimientos obtenidos, las habilidades desarrolladas y la actitud adquirida en un proyecto integrador para resolver una situación que se podría presentar en la vida real, es fundamental aterrizar los temas vistos, reflexionar y generar un aprendizaje significativo.

Estrategia:

- 05' Síntesis de la sesión anterior
- 80' Actividad 32 – Proyecto integrador
- 05' Síntesis de la sesión.

Actividades de Aprendizaje:

- Realizar la actividad 32

Recursos Didácticos:

- Proyector, pizarrón, plumones, internet, equipos de cómputo y/o Smartphone.

Referencias Bibliográficas: