



Guía de estudio

TERCER
SEMESTRE

Modelado de Sistemas y Principios de Programación



PLAN 2014
ACTUALIZADO



PLAN 2014

A C T U A L I Z A D O

CRÉDITO

Autores:

Ingrid Myriam Aburto Castro

Ajuste:

Ingrid Myriam Aburto Castro

Coordinador:

Luis Gabriel Mondragón Torres

Coordinador de Proyectos de Formación Laboral en Informática





PRESENTACIÓN

Con la finalidad de acompañar el trabajo con el plan y programas de estudio vigentes, además de brindar un recurso didáctico que apoye al cuerpo docente y al estudiantado en el desarrollo de los aprendizajes esperados; el Colegio de Bachilleres desarrolló, a través de la Dirección de Planeación Académica y en colaboración con el personal docente de los veinte planteles, las guías de estudio correspondientes a las tres áreas de formación: básica, específica y laboral.

Las guías pretenden ser un apoyo para que las y los estudiantes trabajen de manera autónoma con los contenidos esenciales de las asignaturas y con las actividades que les ayudarán al logro de los aprendizajes; el rol del cuerpo docente como mediador y agente activo en el aprendizaje del estudiantado no pierde fuerza, por el contrario, se vuelve fundamental para el logro de las intenciones educativas de este material.

Las guías de estudio también son un insumo para que las y los docentes lo aprovechen como material de referencia, de apoyo para el desarrollo de sus sesiones; o bien como un recurso para la evaluación; de manera que, serán ellos, quienes a partir de su experiencia definirán el mejor uso posible y lo adaptarán a las necesidades de sus grupos.

El Colegio de Bachilleres reconoce el trabajo realizado por el personal participante en la elaboración y revisión de la presente guía y agradece su compromiso, entrega y dedicación, los cuales se reflejan en el servicio educativo pertinente y de calidad que se brinda a más de 90,000 estudiantes.

La programación es una herramienta fundamental en el mundo en el que vivimos, convirtiéndose en una salida ocupacional importante. Además, su aprendizaje constituye una oportunidad al mejorar el razonamiento lógico. Programar se trata de usar nuestra creatividad e ingenio para resolver problemas y automatizar tareas con la ayuda de una computadora. Dicho de otra forma, la programación trata de entender, construir y modificar software, hoy presente en miles de dispositivos.

La salida ocupacional de Auxiliar Programador pretende que los estudiantes adquieran las competencias profesionales que les permitan insertarse exitosamente en el enorme mercado laboral mencionado y en el cual, el manejo de información a través de bases de datos es fundamental.

La asignatura de Modelado de Sistemas y Principios de Programación plantea abordar los contenidos necesarios para que comprendas el funcionamiento de la programación de tal forma que puedas resolver problemas y automatizar tareas con la ayuda de una computadora; la finalidad de la asignatura consiste en desarrollar una aplicación web básica, al proponer una solución informática a un requerimiento específico.

La era digital evoluciona de una forma exponencial, convirtiéndose en parte vital de la humanidad, por lo tanto, es importante conocer los principios básicos de programación, mismos que encontraras en esta guía. Aprenderás los conceptos fundamentales de programación, diseñaras soluciones informáticas elaborando diagramas de flujo, conocerás la aplicación de las estructuras lógicas de control y utilizarás un software que te permitirá aplicar estos conocimientos. Con esto se desarrolla el razonamiento y el pensamiento lógico para poder dar solución a las necesidades del usuario.

En este material se organizan los contenidos de lo general (requerimientos del cliente) a lo particular (aplicación de las estructuras lógicas de control) y de lo teórico (conceptos y características) a lo práctico (elaboración de una página web). La intención de este material es proporcionarte los elementos necesarios para que integres los aprendizajes de la asignatura, los ejercites y finalmente te evalúes.

Es importante recalcar que, dada la naturaleza de la asignatura, las actividades requieren del uso de una computadora e internet.

¡Bienvenido!

PRESENTACIÓN INTRODUCCIÓN

CORTE DE APRENDIZAJE 1 Análisis de problemas informáticos

Propósito	8
Conocimientos previos	9
Evaluación diagnóstica	10
Requerimientos del cliente	11
Actividad de aprendizaje 1	13
Plan de trabajo	14
Actividad de aprendizaje 2	18
Actividad de aprendizaje 3	19
Algoritmos	19
Actividad de aprendizaje 4	21
Actividad de aprendizaje 5	24
Autoevaluación	25
Fuentes Consultadas	26

CORTE DE APRENDIZAJE 2 Diseños de soluciones a problemas informáticos

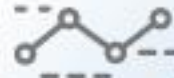
Propósito	28
Conocimientos previos	29
Evaluación diagnóstica	30
Diagrama de flujo	31
Actividad de aprendizaje 1	33
Actividad de aprendizaje 2	35
Actividad de aprendizaje 3	36
Estructuras lógicas de control	37
Actividad de aprendizaje 4	39
Estructuras de datos	41
Actividad de aprendizaje 5	42
Autoevaluación	44
Fuentes Consultadas	45

CORTE DE APRENDIZAJE 3 Desarrollo de soluciones informáticas para web

Propósito	48
Conocimientos previos	49
Evaluación diagnóstica	50
Utilidad del IDE WAMPServer	51
Actividad de aprendizaje 1	53
2. Lenguaje de programación PHP	54
Actividad de aprendizaje 2	58
Actividad de aprendizaje 3	60
Autoevaluación	62
Fuentes Consultadas	63
 EVALUACIÓN FINAL	 64

CORTE

1



Análisis de problemas informáticos mediante algoritmos

Aprendizajes esperados:

Contenidos específicos

1. Requerimientos del cliente

1.1 Técnicas para el levantamiento de información: entrevista y cuestionario

2. Plan de trabajo

2.1 Gráfica de Gantt

2.2 Asignación de Roles y tareas

3. Algoritmos

3.1 Variables

3.2 Constantes

3.3 Expresiones aritméticas, lógicas y de relación

Aprendizajes esperados

1. Elabora un plan de trabajo a partir de un problema requerido por un cliente.

2. Desarrolla el análisis del problema mediante un algoritmo.



Al finalizar el corte el estudiante será capaz de generar una propuesta de solución de una problemática del cliente, para ofrecer alternativas viables al momento de desarrollar aplicaciones informáticas.

Para que logres desarrollar los aprendizajes esperados correspondientes al corte 1 es importante considerar los siguientes conocimientos:

- ✓ Entrevistas
- ✓ Gráficas en Excel
- ✓ Trabajo en equipo
- ✓ Algoritmo
- ✓ Operadores aritméticos

Identifica lo que debes
saber para que la
comprensión de los
contenidos sea más fácil,
si descubres que has
olvidado algo ¡repásalo!



Instrucciones: Lee con atención cada una de las preguntas y escribe en el paréntesis la letra de la respuesta correcta.

1. Son símbolos que indican como se deben manipular los números: ()
a) Operadores b) Expresiones c) Variables d) Literales
2. Es una conversación entre dos o más personas, donde se intercambian ideas y opiniones: ()
a) Encuesta b) Reunión c) Entrevista d) Cuestionario
3. Algunos operadores aritméticos son: ()
a) +, -, *, / b) +, -, *, /, % c) +, -, *, /, & d) +, -, *, /, |
4. Característica del trabajo en equipo que genera respuestas inteligentes e ingeniosas a los problemas: ()
a) Liderazgo b) Metas c) Cooperación d) Creatividad
5. Es una representación visual de ciertos datos numéricos que nos permiten analizar los resultados: ()
a) Herramienta b) Gráfico c) Mapa d) Diagrama
6. Secuencia de pasos que permiten solucionar un problema: ()
a) Gráfico b) Algoritmo c) Diagrama d) Código
7. Característica del trabajo en equipo donde cada integrante conoce su rol y colabora con los otros integrantes: ()
a) Liderazgo b) Metas c) Cooperación d) Creatividad

1. REQUERIMIENTOS DEL CLIENTE

1.1 Técnicas para el levantamiento de información

La recolección de datos se refiere al uso de una gran diversidad de técnicas y herramientas que pueden ser utilizadas por el analista para desarrollar los sistemas de información, los cuales pueden ser la entrevistas, la encuesta, el cuestionario, la observación, el diagrama de flujo y el diccionario de datos.

Todos estos instrumentos se aplicarán en un momento en particular, con la finalidad de buscar información que será útil a una investigación en común. Las 5 principales técnicas de recolección de datos son:

1. Entrevista
2. La encuesta y cuestionario
3. La observación
4. Diccionario de datos
5. Diagrama de flujo

En este material nos enfocaremos solo en dos técnicas: Entrevista y cuestionario.

Entrevista

La entrevista es una conversación dirigida, con un propósito específico y que usa un formato de preguntas y respuestas. Se establece así un diálogo, pero un diálogo peculiar, asimétrico, donde una de las partes busca recoger informaciones y la otra se nos presenta como fuente de estas informaciones.

Una entrevista es un dialogo en el que la persona (entrevistador), generalmente un periodista hace una serie de preguntas a otra persona (entrevistado), con el fin de conocer mejor sus ideas, sus sentimientos su forma de actuar.

Preparación de la Entrevista

1. Determinar la posición que ocupa de la organización el futuro entrevistado, sus responsabilidades básicas, actividades, etc. (Investigación).
2. Preparar las preguntas que van a plantearse, y los documentos necesarios (Organización).
3. Fijar un límite de tiempo y preparar la agenda para la entrevista. (Sicología).
4. Elegir un lugar donde se puede conducir la entrevista con la mayor comodidad (Sicología).
5. Hacer la cita con la debida anticipación (Planeación).

Tipo de preguntas

- ✓ Preguntas abiertas: Son aquellas preguntas que describen hechos o situaciones por parte del entrevistado con una gran cantidad de detalles que a juicio del entrevistado son importantes.
- ✓ Preguntas cerradas: En las preguntas cerradas las respuestas posibles están cerradas al entrevistado, debido a que solamente puede responder con un número finito, tal como “ninguno”, “uno”, o “quince”. Una pregunta cerrada limita las respuestas disponibles al entrevistado.

Cuestionario

Los cuestionarios proporcionan una alternativa muy útil para la entrevista; sin embargo, existen ciertas características que pueden ser apropiada en algunas situaciones e inapropiadas en otra. Al igual que la entrevistas, deben diseñarse cuidadosamente para una máxima efectividad.

El desarrollo y distribución de los cuestionarios; por lo tanto, el tiempo invertido en esto debe utilizarse en una forma inteligente. También es importante el formato y contenido de las preguntas en la recopilación de hechos significativos.

Existen dos formas de cuestionarios para recabar datos: cuestionarios abiertos y cerrados, y se aplican dependiendo de si los analistas conocen de antemano todas las posibles respuestas de las preguntas y pueden incluirlas. Con frecuencia se utilizan ambas formas en los estudios de sistemas.

Cuestionario Abierto

Al igual que las entrevistas, los cuestionarios pueden ser abiertos y se aplican cuando se quieren conocer los sentimientos, opiniones y experiencias generales; también son útiles al explorar el problema básico, por ejemplo, un analista que utiliza cuestionarios para estudiar los métodos de verificación de crédito es un medio.

El formato abierto proporciona una amplia oportunidad para quienes respondan escriba las razones de sus ideas. Algunas personas, sin embargo, encuentran más fácil escoger una de un conjunto de respuestas preparadas que pensar por sí mismas.

Cuestionario Cerrado

El cuestionario cerrado limita las respuestas posibles del interrogado. Por medio de un cuidadoso estilo en la pregunta, el analista puede controlar el marco de referencia. Este formato es el método para obtener información sobre los hechos. También fuerza a los individuos para que tomen una posición y forma su opinión sobre los aspectos importantes.

Actividad de aprendizaje 1

La entrevista con un cliente es el primer paso de todo proyecto, para poder elaborarlo necesitamos datos que nos proporcione el cliente, como pueden ser las necesidades especiales y el diseño del sistema.

Instrucciones: Apóyate de un profesor, conocido o familiar que tenga conocimientos sobre informática y el desarrollo de sistemas, este tendrá el papel del dueño de la empresa y pídele que te ayude a realizar la siguiente entrevista, en la cual tu rol será del líder del proyecto que desarrollará un sistema informático para su empresa. No olvides anotar las respuestas.

Información general del proyecto

1. ¿Qué tipo de giro es el de su empresa?
2. ¿Cuál es el objetivo de crear el sistema?
3. ¿Cada cuánto tiempo quiere ver los avances que voy haciendo?
4. ¿Suele contratar profesionales o pasantes?
5. Si fuese necesario, ¿le importaría si cuento con otros profesionales para realizar ciertas partes del trabajo?
6. Si es necesario, ¿tendré disponibilidad para acceder a la información de su empresa?
7. ¿Cuenta con las licencias de software o algún hardware especial?

Sobre el acuerdo contractual

8. ¿Cuál es el presupuesto para este proyecto?
9. ¿Quién se quedará con los derechos al acabar el proyecto?
10. ¿Cuál es el plazo de entrega?
11. ¿Cuál va a ser la fecha de pago?

Seguimiento del proyecto

12. ¿Este proyecto tendrá que ser actualizado o revisado en algún momento?
13. ¿Hay algún otro proyecto que pueda realizar para tu empresa?

Plan de trabajo

Comenzar un proyecto sin un plan es como viajar sin mapa. Puede que al final llegues adónde querías ir, pero perdiendo mucho tiempo y dinero por el camino. Embarcarse en un proyecto sin un plan bien formulado puede conllevar arrastramiento del alcance, presupuestos malgastados y plazos incumplidos.

Sentarte con tu equipo y elaborar un plan de proyecto antes de empezar a escribir el código de programación o de perfeccionar los diseños, te permitirá fijar las expectativas que te guiarán al éxito.

Pasos para crear un proyecto.

Paso 1: Define tu proyecto

Tanto si estás diseñando una nueva aplicación o modificándola, comienza siempre por definir lo que estás tratando de conseguir de verdad. Hay seis elementos clave para definir el proyecto:

1. **Objetivos.** Es decir, lo que se pretende conseguir con este proyecto, tanto a nivel externo (beneficios, cuota de mercado, satisfacción del cliente) como interno (mejora de la infraestructura, optimización de procesos, retención de empleados).
2. **Alcance.** No importa lo detallado que sea tu plan, tu proyecto sufrirá cambios con los que tendrás que lidiar. En lugar de tratar de identificar todo lo que está incluido en el alcance de tu proyecto, resultará más sencillo identificar qué está sin duda fuera del alcance. Este también es un buen momento para definir quién se encargará de aprobar o rechazar los posibles cambios del alcance.

3. Criterios de éxito. ¿Qué determinará si este proyecto es un éxito o fracasa? Vuelve a revisar tus objetivos durante este paso. Los criterios habituales son que los proyectos se entreguen a tiempo y dentro del presupuesto, que el producto final cumpla con un cierto nivel de calidad y que resuelva un problema de negocio específico.

4. Entregables. Enumera los entregables esenciales de tu proyecto. Algunos ejemplos podrían ser una aplicación móvil que funcione para tu negocio, una tienda de “crepes” lista para el cliente o un nuevo sistema de nóminas para tus empleados. Sé tan minucioso como puedas.

5. Requisitos. Decide qué necesitas (recursos, personal, presupuesto, tiempo) para cumplir con los objetivos del proyecto y lograr los resultados, y haz un registro de todo. Consulta a todas las partes interesadas clave para asegurarte de que no te olvides de nada.

6. Calendario. Utiliza una estructura de descomposición del trabajo para determinar qué se debe entregar y cuándo, y luego utilízalo para definir el calendario base, los hitos del proyecto y los plazos (contenido en inglés).

Paso 2: Identifica riesgos, suposiciones y limitaciones

En esta etapa, nombra a un miembro del equipo responsable de la gestión de riesgos. Dependiendo del volumen del proyecto, esta persona podrías ser tú mismo, el gestor de proyectos u otro líder del proyecto. Lo importante es que alguien se encargue de supervisar cualquier riesgo que hayas identificado (por ejemplo, no cumplir con las fechas límite, sobrepasar el presupuesto). El software de gestión de proyectos puede hacer que la gestión de riesgos sea cosa de niños con una definición precisa del proyecto.

Si el proyecto debe completarse durante las horas de trabajo habituales (sin recurrir a horas extras) o con recursos ya disponibles, es importante que tu equipo lo sepa.

Paso 3: Organiza a los miembros de tu proyecto

Este paso es la esencia de cómo hacer un plan de trabajo para tu proyecto, después de todo, la gestión de proyectos consiste en gran medida en la gestión de personas y cuanto más detallado sea el trabajo que efectúes en los pasos uno y dos, más fácil será este paso.

Organizar tu proyecto significa identificar y documentar los aspectos siguientes:

- **Cliente.** ¿Quién recibirá el producto final? ¿Se trata de un cliente interno o externo?
- **Partes interesadas.** Las partes interesadas son las personas o los grupos interesados en la correcta consecución del proyecto. La identificación de las partes interesadas contribuirá al plan de comunicación de tu proyecto.
- **Funciones y responsabilidades.** Define la jerarquía organizativa del proyecto. ¿Quién tiene la última palabra en la toma de decisiones? ¿Quién se encarga de la comunicación con el cliente? ¿Quién supervisa el presupuesto y el calendario en la

herramienta de gestión de proyectos? ¿Quién se ocupa de hacer el trabajo de verdad?

Paso 4: Enumera los recursos de tu proyecto

En el primer paso identificaste algunos requisitos muy exigentes que se necesitan para finalizar con éxito el proyecto, y en el tercer paso identificaste los recursos humanos necesarios para llevar a cabo y gestionar el trabajo. Ahora ha llegado el momento de centrarse en los recursos específicos que se precisan para comenzar a trabajar.

Algunos ejemplos de recursos del proyecto podrían ser:

- Tecnología (ordenadores, software, dispositivos móviles)
- Salarios
- Vehículos
- Materiales físicos

Paso 5: Fija un plan de comunicación del proyecto

Bueno, ahora sabes cómo hacer un plan de trabajo y seguramente ya tienes un proyecto claramente definido con su calendario, presupuesto, recursos y equipo. Es hora de ponerse a trabajar, ¿verdad? Pues no. Todavía no tienes un plan de comunicación.

La mala comunicación es una de las causas principales del fracaso del proyecto y si se da por sentada una buena comunicación en equipo, podría ser el error de suposición que tire por tierra todo el proyecto.

2.1 Gráfica de Gantt

Un diagrama de Gantt es una herramienta útil para planificar proyectos. Al proporcionarte una vista general de las tareas programadas, todas las partes implicadas sabrán qué tareas tienen que completarse y en qué fecha.

Un diagrama de Gantt te muestra:

- La fecha de inicio y finalización de un proyecto
- Qué tareas hay dentro del proyecto
- Quién está trabajando en cada tarea
- La fecha programada de inicio y finalización de las tareas
- Una estimación de cuánto llevará cada tarea
- Cómo se superponen las tareas y/o si hay una relación entre ellas

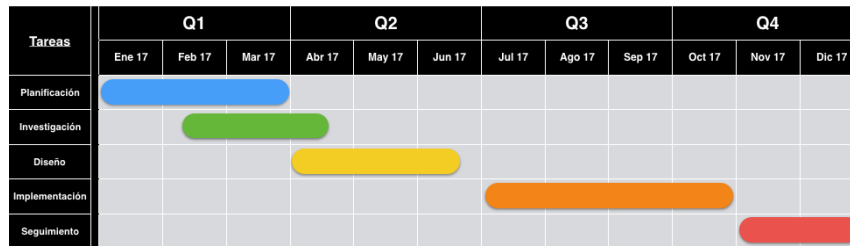


Fig. 1 Ejemplo de una gráfica de Gantt

Componentes de un diagrama de Gantt

Normalmente, un diagrama de Gantt contiene los siguientes elementos:

1. **Fechas:** las fechas de inicio y finalización permiten que los gestores de proyecto tengan una visión de cuándo empezará y terminará la totalidad del proyecto.
2. **Tareas:** los proyectos consisten en una serie de subtareas. Con un diagrama de Gantt, podrás hacer un seguimiento de estas subtareas para que ninguna sufra retrasos o se quede olvidada.
3. **Plazos de tiempo previstos:** el diagrama muestra cuándo debe llevarse a cabo cada tarea. Te ayudará a garantizar que cada subtask se completará según el programa y que todo el proyecto se terminará a tiempo.
4. **Tareas interdependientes:** algunas tareas se pueden llevar a cabo en cualquier momento, mientras que otras se deben completar antes o después de que empiece o termine otra tarea. Estas tareas que dependen de otras se pueden indicar en un diagrama de Gantt.
5. **Progreso:** el diagrama te muestra exactamente cómo se está desarrollando tu proyecto ya que te ofrece una representación de las tareas que ya se han completado. Al indicar la fecha actual, obtendrás una vista general de cuánto queda por hacer y verás si todo procede como estaba planeado para completar el proyecto a tiempo.

Para crear tu primer diagrama de Gantt, sigue el proceso que te presentamos a continuación:

1. Haz una lista de todas las tareas
2. Haz una lista de todas las tareas interdependientes
3. Crea una línea temporal del proyecto
4. Asigna tareas a los miembros del equipo

Con el objetivo de reforzar el contenido de Diagrama de Gantt, consulta el siguiente enlace que contiene un video que te mostrará la creación del diagrama desde Excel.

Diagrama de Gantt: <https://www.youtube.com/watch?v=i33tFqmkrX4>

Actividad de aprendizaje 2

Instrucciones: Elabora y completa la siguiente tabla de Excel que ejemplifica una gráfica de Gantt, misma que contiene las actividades que se deben realizar para desarrollar un sistema informático, deberás completar la gráfica rellenando las celdas de las actividades de acuerdo con las fechas de inicio y termino tal como se muestra en las 4 primeras actividades, guarda el archivo en una memoria USB.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
1	Gráfica de Gantt para elaborar un Sistema Informático																							
2																								
3	No.	Actividad	FECHAS		2022																			
4			INICIO	TERMINO	JUNIO				JULIO				AGOSTO				SEPTIEMBRE				OCTUBRE			
5					S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4
6	1	Selección de la empresa	10/06/2022	14/06/2022																				
7	2	Obtención de los requerimientos	17/06/2022	28/06/2022																				
8	3	Análisis de datos	24/06/2022	28/06/2022																				
9	4	Elaboración del diagnóstico	24/06/2022	12/07/2022																				
10	5	Elaboración del presupuesto	15/07/2022	26/07/2022																				
11	6	Propuesta del Sistema	29/07/2022	09/08/2022																				
12	7	Elaboración y desarrollo del sistema	05/08/2022	30/08/2022																				
13	8	Crear base de datos	05/08/2022	23/08/2022																				
14	9	Desarrollar procedimientos	05/08/2022	30/08/2022																				
15	10	Crear las entradas de datos	19/08/2022	30/08/2022																				
16	11	Desarrollo de la interfaz gráfica	19/08/2022	02/09/2022																				
17	12	Conectividad entre interfaz y BD	30/08/2022	13/09/2022																				
18	13	Desarrollo de aplicaciones de salida	09/09/2022	20/09/2022																				
19	14	Pruebas finales	20/09/2022	04/10/2022																				
20	15	Depuración final	04/10/2022	11/10/2022																				
21	16	Primera entrega de proyecto	08/07/2022	12/07/2022																				
22	17	Segunda entrega de proyecto	26/08/2022	30/08/2022																				
23	18	Entrega final	07/10/2022	11/10/2022																				
24																								
25																								

Fig. 2 Gráfica de Gantt para elaborar un sistema

2.2 Asignación de roles y tareas

El desarrollo de software es una actividad que, dada su complejidad, debe desarrollarse en grupo. Además, esta actividad requiere de distintas capacidades, las que no se encuentran todas en una sola persona. Por ello, se hace necesario formar el grupo de desarrollo con las personas que cubran todas las capacidades requeridas. Cada una de esas personas aportará al grupo parte del total de las capacidades necesarias para llevar a cabo con éxito el desarrollo, algunos de estos roles son:

- Administrador de proyecto
- Analista
- Diseñador
- Programador
- Téster
- Asegurador de calidad
- Documentador
- Ingeniero de manutención

- Ingeniero de validación
- Verificación
- Administrador de la configuración

Para cada uno de estos roles, se definen sus objetivos, actividades, interacción con otros roles, herramientas a utilizar, perfil de las personas en ese rol y un plan de trabajo.

Hay que señalar que es posible que no se requieran todos los roles en un desarrollo. Eso dependerá del tamaño y del tipo del desarrollo. Por ejemplo, el desarrollo de un sistema de información de gran tamaño requerirá más roles que uno de menor tamaño. Por otro lado, si el tipo del proyecto está enfocado más hacia la parametrización e integración de sistemas, requerirá algunos roles en menor medida y otros en mayor.

Es posible también que una persona realice las labores de más de un rol al mismo tiempo. Esto, sobre todo en proyectos de desarrollo de software más pequeños. No obstante, es imprescindible que dichas personas conozcan completamente todas sus tareas.

Actividad de aprendizaje 3

Instrucciones: Elabora un mapa conceptual que incluya la definición y actividades que deben realizar los siguientes roles dentro de un equipo: analista, administrador de proyecto, diseñador, programador y documentador, consulta el siguiente enlace que contiene la información para realizar el mapa conceptual:

Enlace: http://profayadira.yolasite.com/resources/Roles_desarrollo_software.pdf

Algoritmos

En informática, un algoritmo es una secuencia de instrucciones secuenciales, gracias al cual pueden llevarse a cabo ciertos procesos y darse respuesta a determinadas necesidades o decisiones. Se trata de conjuntos ordenados y finitos de pasos, que nos permiten resolver un problema o tomar una decisión.

Partes de un algoritmo

Todo algoritmo debe constar de las siguientes partes:

- Input o entrada. El ingreso de los datos que el algoritmo necesita para operar.
- Proceso. Se trata de la operación lógica formal que el algoritmo emprenderá con lo recibido del input.

- **Output o salida.** Los resultados obtenidos del proceso sobre el input, una vez terminada la ejecución del algoritmo.

Los algoritmos presentan las siguientes características:

- **Secuenciales.** Los algoritmos operan en secuencia, debe procesarse uno a la vez.
- **Precisos.** Los algoritmos han de ser precisos en su abordaje del tema, es decir, no pueden ser ambiguos o subjetivos.
- **Ordenados.** Los algoritmos se deben establecer en la secuencia precisa y exacta para que su lectura tenga sentido y se resuelva el problema, los pasos deben estar enumerados.
- **Finitos.** Toda secuencia de algoritmos ha de tener un fin determinado, no puede prolongarse hasta el infinito.
- **Concretos.** Todo algoritmo debe ofrecer un resultado en base a las funciones que cumple.
- **Definidos.** Un mismo algoritmo ante los mismos elementos de entrada (input) debe dar siempre los mismos resultados.

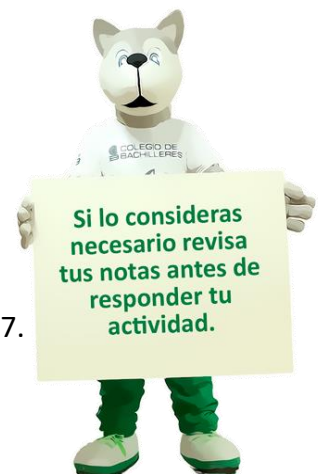
Ejemplos de algoritmos

Algoritmo verbal. Para hacer un batido de melón.

1. Inicio
2. Pelar el melón y picarlo en cubos.
3. Insertar los cubos en una licuadora.
4. Enchufar la licuadora si no está enchufada
5. Encender la licuadora y licuar durante 2 minutos
6. Apagar la licuadora y desenchufarla
7. Colar el jugo y servirlo en una jarra
8. Fin

Algoritmo matemático. Para determinar la suma de dos números: 15 y 7.

1. Inicio
2. Sumatoria de los números $15 + 7 = 22$.
3. Resultado 22
4. Fin



Con el objetivo de reforzar el contenido de Algoritmos, consulta el siguiente enlace que te explica con mayor amplitud y de una forma ingeniosa el contenido de Algoritmo.

¿Qué es un algoritmo? <https://www.youtube.com/watch?v=U3CGMyjzlvM>



Fig. 3 Video referente a algoritmos

Actividad de aprendizaje 4

Instrucciones: Elabora un algoritmo que realice las operaciones básicas que son: suma, resta, división y multiplicación de los números 10 y 5, puedes apoyarte del algoritmo matemático que se encuentra en el subtema 3.2 ejemplos de algoritmos.

3.1 Variables

Es un espacio en la memoria de la computadora que permite almacenar temporalmente un dato durante la ejecución de un proceso, su contenido puede cambiar durante la ejecución del programa.

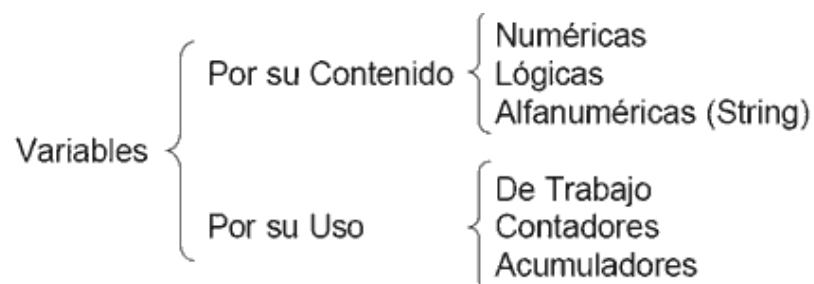
Para poder reconocer una variable en la memoria de la computadora, es necesario darle un nombre con el cual podamos identificarla dentro de un algoritmo.

Ejemplo:

$$\text{área} = \pi * \text{radio}^2$$

Las variables son: el radio, el área y la constante es pi

Clasificación de las Variables



Por su contenido

- Variables Numéricas: Son aquellas en las cuales se almacenan valores numéricos, positivos o negativos, es decir almacenan números del 0 al 9, signos (+ y -) y el punto decimal.

Ejemplo:

$\text{IVA} = 0.15$ $\pi = 3.1416$ $\text{costo} = 2500$

- Variables Lógicas: Son aquellas que solo pueden tener dos valores (cierto o falso) estos representan el resultado de una comparación entre otros datos.
- Variables Alfanuméricas: Esta formada por caracteres alfanuméricos (letras, números y caracteres especiales).

Ejemplo:

letra = a apellido = López dirección = Av. Libertad #190

Por su uso

- Variables de Trabajo: Variables que reciben el resultado de una operación matemática completa y que se usan normalmente dentro de un programa.
Ejemplo:
 $\text{Suma} = a + b / c$
- Contadores: Se utilizan para llevar el control del número de ocasiones en que se realiza una operación o se cumple una condición. Con los incrementos generalmente de uno en uno.
- Acumuladores: Forma que toma una variable y que sirve para llevar la suma acumulativa de una serie de valores que se van leyendo o calculando progresivamente.

3.2 Constantes

Una constante es un dato numérico o alfanumérico que no cambia durante la ejecución del programa.

Ejemplo:

$\pi = 3.1416$

3.3 Expresiones aritméticas, lógicas y de relación

Las expresiones son combinaciones de constantes, variables, símbolos de operación, paréntesis y nombres de funciones especiales.

Por ejemplo:

$a + (b + 3) / c$

Cada expresión toma un valor que se determina tomando los valores de las variables y constantes implicadas y la ejecución de las operaciones indicadas.

Una expresión consta de operadores y operandos. Según sea el tipo de datos que manipulan, se clasifican las expresiones en:

- Aritméticas
- Relacionales
- Lógicas

Ejemplo de un algoritmo usando variables, constantes y expresiones.

El siguiente algoritmo determina el área de un círculo, es similar a los algoritmos vistos anteriormente sin embargo integraremos los elementos de variables, constantes y expresiones, los cuales son parte de la programación, a continuación, se describen estos elementos para una mejor comprensión.

- A= Variable que almacenara el área.
- P= Variable que almacena el valor de 3.1416.
- R= Variable que almacena el valor del radio.

Algoritmo

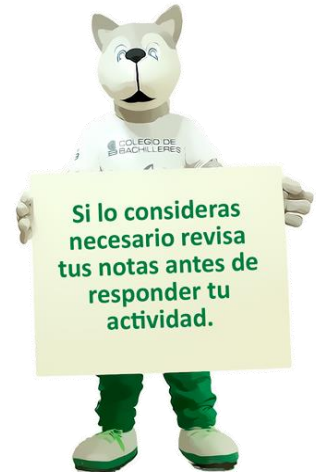
1. Inicio
2. Ingresa el valor del radio
3. Ingresa el valor de P
4. $A = P * r^2$
5. Resultado del área
6. Fin

Actividad de aprendizaje 5

Instrucciones: El siguiente algoritmo determina el promedio de una asignatura, aplicando el concepto de variables y considerando tres parciales, completa el algoritmo escribiendo en los espacios vacíos las palabras del recuadro:

P1 - Fin - tercer - 3 - Inicio - P3

1. _____
2. Ingresa la calificación del primer parcial
3. _____
4. Ingresa la calificación del segundo parcial
5. P2
6. Ingresa la calificación del _____ parcial
7. _____
8. $Prom = (P1 + P2 + P3) / ___$
9. El promedio es
10. Prom
11. _____



Técnicas para el levantamiento de información: entrevista y cuestionario

<https://femrecerca.cat/menases/publication/cuestionario-entrevista/cuestionario-entrevista.pdf>

Gráfica de Gantt

<https://www.youtube.com/watch?v=chR6kx4btDQ&t=70s>

Asignación de Roles y tareas

<https://www.youtube.com/watch?v=0M7TcfJ86eo>

Algoritmos

https://www.youtube.com/watch?v=sNV8BCYi4h0&list=PLg9145ptuAigaT_6U8vHwQqD9sCKT8DKN&index=2

Variables y constantes

<https://codigosdeprogramacion.com/cursos/?lesson=3-tipos-de-datos-variables-y-constantes>

Expresiones aritméticas, lógicas y de relación

https://www.youtube.com/watch?v=hnH_IrQ2bjE

https://es.wikiversity.org/wiki/Fundamentos_de_programaci%C3%B3n/Expresiones

Instrucciones: En relación con los contenidos vistos en este primer corte, completa la siguiente tabla de acuerdo con tus esfuerzos, la valoración de tu trabajo y los logros obtenidos.

¿Qué sabía de programación?	¿Cómo he ido aprendiendo?	¿Qué se ahora?

Debo mejorar en:

Sitios web

Capterra. (2000). Cómo hacer un plan de trabajo infalible. Recuperado el 04 de mayo de 2022 de <https://www.capterra.es/blog/1008/como-hacer-un-plan-de-trabajo-infalible>

Teamleader. (2005). ¿Qué es y para qué sirve un diagrama de Gantt? Recuperado el 04 de mayo de 2022 de <https://blog.teamleader.es/diagrama-de-gantt>

Documento digital

Padilla, D. (2003). Roles en el desarrollo del software. Recuperado el 04 de mayo de 2022 de http://profayadira.yolasite.com/resources/Roles_desarrollo_software.pdf

Video

Magic Markes. (21 de julio 2015) YouTube. ¿Qué es un algoritmo? Recuperado el 04 de mayo de 2022 de <https://www.youtube.com/watch?v=U3CGMyjzlvM>

Tabla de imágenes corte 1

Figura	Referencia
1ª	Google imágenes
2ª	Aburto, I. (2022) Grafica de Gantt
3ª	Magic Markes. (21 de julio 2015) YouTube. ¿Qué es un algoritmo?

CORTE

2



Diseño de soluciones a problemas informáticos mediante la programación estructurada

Aprendizajes esperados:

Contenidos específicos

1. Diagrama de flujo

- 1.1 Procesos
- 1.2 Simbología
- 1.3 Reglas de construcción para su elaboración

2. Estructuras lógicas de control

- 2.1 Secuencial
- 2.2 Selección: simple
- 2.3 Repetición: mientras-haz y repite-hasta

3. Estructuras de datos

- 3.1 Simples

Aprendizajes esperados

- 1. Diseña soluciones informáticas utilizando diagramas de flujo y pseudocódigo, que incluyan las estructuras lógicas de control y de datos

Al finalizar el corte el estudiante será capaz de diseñar aplicaciones informáticas; lo anterior le permitirá resolver problemas informáticos basados en el paradigma de programación estructurada.

RECOMENDACIÓN

Te sugerimos, revises los aprendizajes esperados antes de iniciar con el estudio del corte, realiza las anotaciones que sean necesarias.

Para que logres desarrollar los aprendizajes esperados correspondientes al corte 2, es importante considerar los siguientes conocimientos adquiridos en el primer corte:

- ✓ Algoritmo
- ✓ Variables
- ✓ Constantes
- ✓ Expresiones aritméticas, lógicas y de relación.



Identifica lo que debes
saber para que la
comprensión de los
contenidos sea más fácil,
si descubres que has
olvidado algo ¡repásalo!

Instrucciones: Relaciona el elemento de programación con su definición, escribiendo el número dentro del paréntesis correspondiente.

Elementos	Definición
1. Algoritmo	() Dato numérico o alfanumérico que no cambia durante la ejecución del programa.
2. Expresiones	() Herramienta útil para planificar proyectos, proporciona una vista general de las tareas programadas.
3. Constantes	() Su contenido puede cambiar durante la ejecución del programa.
4. Grafica de Gantt	() Secuencia de instrucciones secuenciales, gracias al cual pueden llevarse a cabo ciertos procesos y dar soluciones.
5. Variables	() Combinaciones de constantes, variables, símbolos de operación, paréntesis y nombres de funciones especiales.
6. Entrevista	() Conversación dirigida, con un propósito específico y que usa un formato de preguntas y respuestas.

1. Diagrama de flujo

Un diagrama de flujo es una representación gráfica de un algoritmo. Se trata de representar los pasos que sigue un proceso o algoritmo desde que se inicia hasta que se termina y para ello se utiliza una serie de elementos visuales que te ayuden a dibujar cada paso que sigue un proceso.

Características

- Es una herramienta sencilla de usar con un mínimo de formación/capacitación para dibujarlos e interpretarlos.
- Representa visualmente una forma esquemática de todos los pasos por los que atraviesa un proceso.
- Se utiliza una simbología en cada uno de los pasos que sigue un proceso.
- Se puede utilizar para dibujar un proceso complejo o dividir éste en subprocesos y dibujar un diagrama de flujo por cada uno de ellos.

1.1 Procesos

Es una secuencia específica de actividades, es decir, a los pasos a dar dentro del diagrama de flujo. Por ejemplo, en informática, los procesos son secuencias iniciadas o bien por disparadores programados dentro del sistema, o por intervenciones del usuario del sistema. Cada uno posee una dirección, un propósito y una serie de pasos a realizar.

1.2 Simbología

Los principales símbolos convencionales que se emplean en los diagramas de flujo son los siguientes:

SÍMBOLO	SIGNIFICADO
	Operación: Se usa para describir cualquier actividad. En el interior del rectángulo se escribe una breve descripción de la actividad.
	Límites del Proceso: Indica el inicio y el final de un proceso. En el interior del óvalo aparece la palabra inicio o fin.
	Punto de Decisión: Denota que en ese punto se toma una decisión. Los outputs salidas del diamante, son siempre dos y del tipo SI / No.
	Movimiento: Muestra el movimiento de un output entre distintos puntos de la organización.
	Conector: Señala que el output de ese proceso puede ser el input de otro (la letra indica el proceso de entrada)
	Dirección del flujo: Denota la dirección y el orden de los pasos del proceso
	Documento: Documento/registro.
	Listados: Listados / notas de trabajo acumulado, información referente a la actividad.
	Base de datos: Punto de archivo donde se retiene temporalmente la información, en espera que se cumplan otras condiciones para continuar el proceso. Puede llevar asociada una tarea de administración de almacenamiento.

1.3 Reglas de construcción para su elaboración

- 1.- Se debe de escribir de arriba hacia abajo y de izquierda a derecha
- 2.- Los símbolos se unen con líneas
- 3.- Se usan flechas para indicar el flujo de información
- 4.- Los símbolos de decisión deben tener más de una línea de flujo
- 5.- Todos los símbolos pueden tener más de una línea de entrada
- 6.- Todo símbolo (excepto las líneas de flujo) llevará en su interior información que indique su función.
- 7.- Un elemento del diagrama no puede tener más de una salida si no es un elemento de decisión.
- 8.- Las líneas de flujo no pueden cruzarse.

Ejemplos:

Vamos a retomar el ejercicio de algoritmo que hicimos en el corte anterior donde se buscaba determinar la suma de dos números: $15+7$.

Algoritmo

1. Inicio
2. Sumatoria de los números $15 + 7 = 22$.
3. Resultado 22
4. Fin

Diagrama de flujo

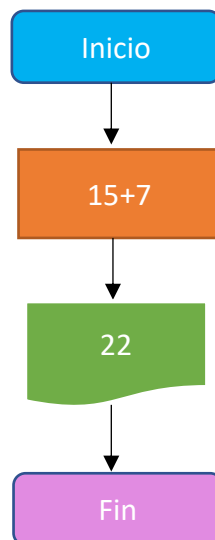


Fig. 4 Ejemplo de diagrama de flujo

Actividad de aprendizaje 1

Instrucciones: Retomaremos el algoritmo que determinaba el área de un círculo, completa el diagrama de flujo con el símbolo faltante o con el texto que debe tener el símbolo, de acuerdo con el algoritmo.

Algoritmo

1. Inicio
2. Ingresar el valor del radio

3. Ingresar el valor de P
4. $A = P * r^2$
5. Resultado del área
6. Fin

Diagrama de flujo

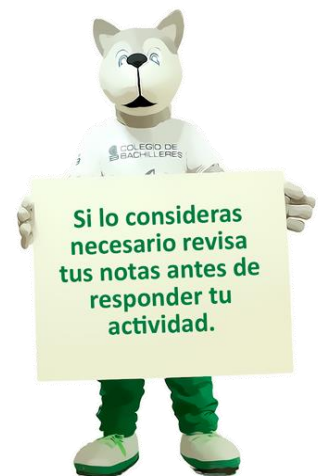
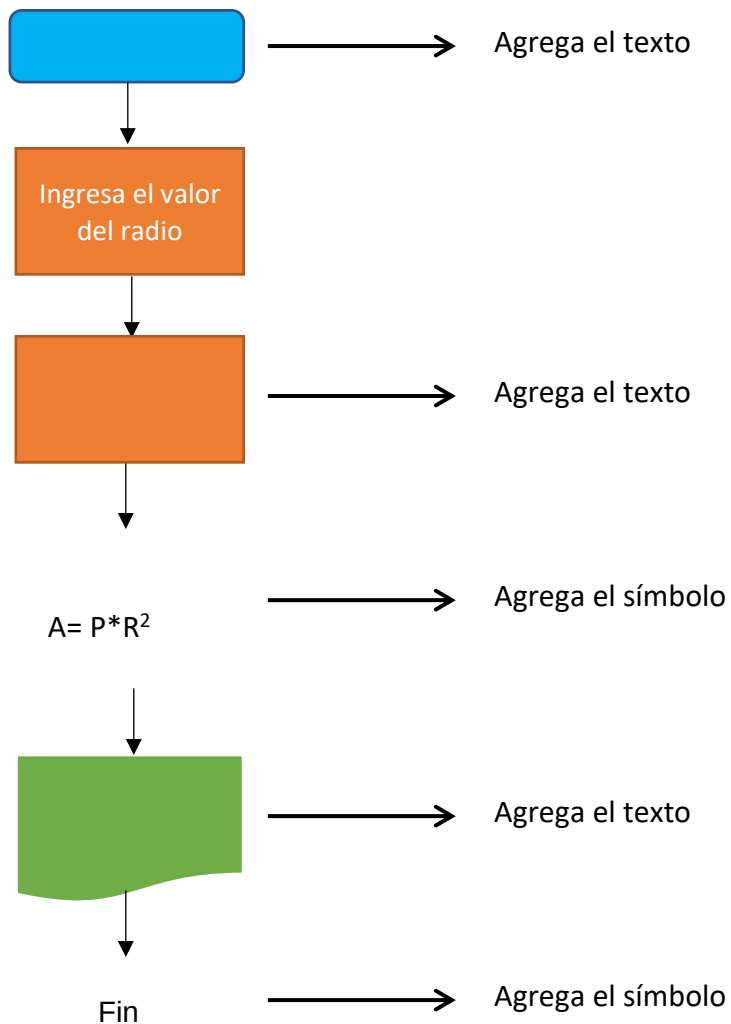


Fig. 5 Diagrama de flujo área del círculo

Para reforzar los aprendizajes referentes a diagramas de flujo, observa el siguiente video y descarga el programa PseInt en tu computadora, con este programa comenzarás a aplicar los conocimientos vistos hasta este momento de programación, PseInt está diseñado para aprender los principios básicos de programación y es muy fácil de utilizar.

Video: Descarga PseInt portable, enlace <https://youtu.be/hqT0Tbsl5q8>

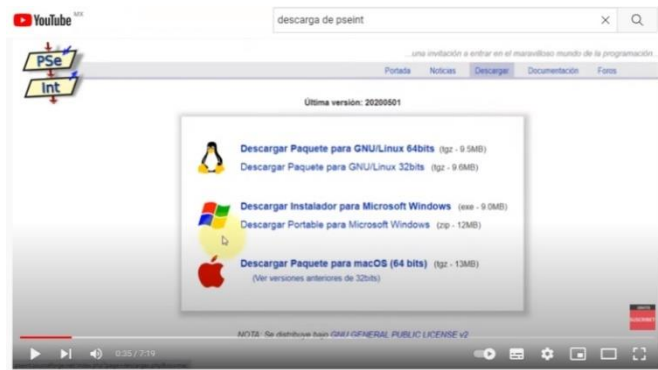


Fig. 6 Video descarga PseInt portable

Actividad de aprendizaje 2

Instrucciones: Observa el siguiente video y realiza el ejercicio de diagrama de flujo, que calcula el área de un triángulo, guarda el programa en una memoria USB con el nombre actividad de aprendizaje 2, ya que los entregarás como parte de la guía.

Video: Diagrama de flujo en PseInt

<https://www.youtube.com/watch?v=RTWzhECFNyc&t=201s>

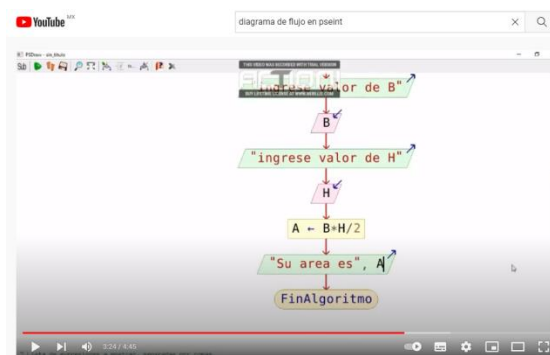


Fig. 7 video que calcula el área de un triángulo

Actividad de aprendizaje 3

Instrucciones: En el primer corte elaboraste el algoritmo que sumaba, restaba, dividía y multiplicaba los números 10 y 5, con base a ese algoritmo elabora el diagrama de flujo pero que ingrese dos números y realice las 4 operaciones recuerda hacerlo en el programa de PseInt y guardarlo con el nombre de operaciones en una memoria USB. En la siguiente imagen se muestra una forma de realizar el diagrama de flujo, sin embargo, puedes utilizar otras variables.

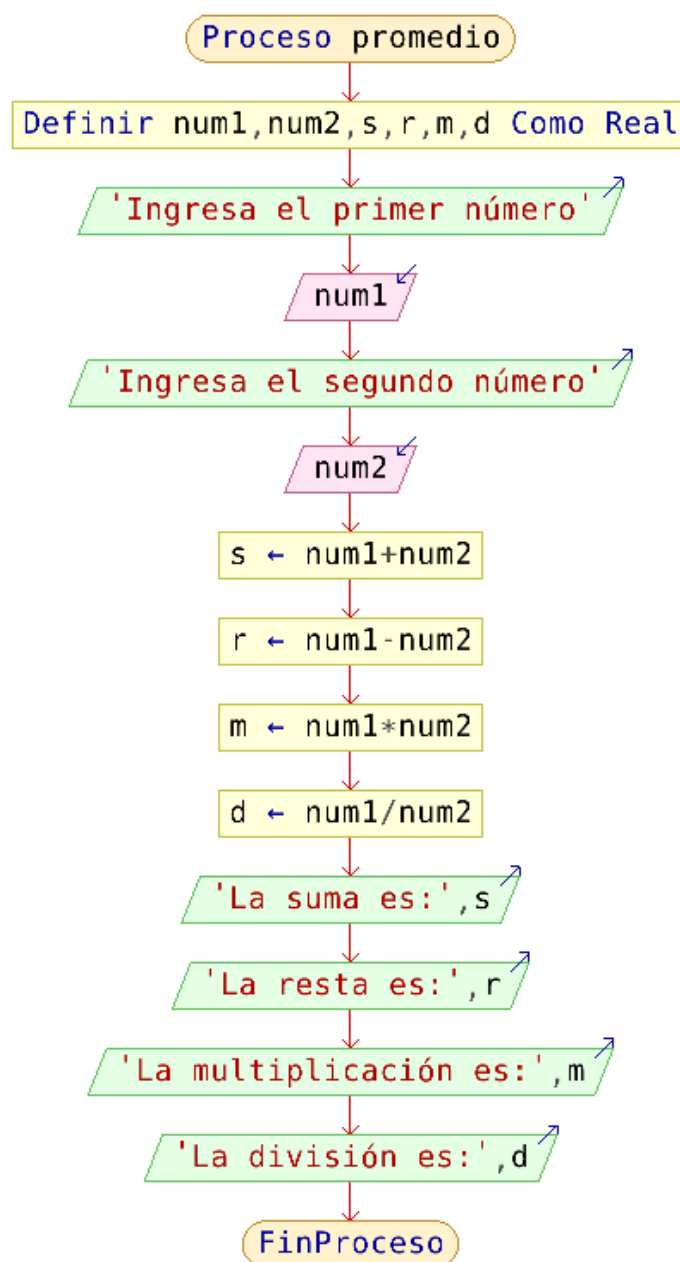


Fig. 8 Diagrama que calcula operaciones

2. Estructuras lógicas de control

Los algoritmos por naturaleza tienen una estructura secuencial, es decir, se presenta una acción tras otra. Una serie de acciones secuenciales es la estructura más básica de cualquier algoritmo.

No obstante que la secuencia es la base de los algoritmos, no siempre se da de esta forma. Por ejemplo, supongamos que tenemos que hacer un algoritmo para hacer agua de limón con limones reales, una de las operaciones puede ser probar lo dulce del agua, si le falta azúcar hay que ponerle más, si no le hace falta azúcar entonces hay que continuar; a este tipo de operación se le denomina una estructura de condición. En el mismo ejemplo del algoritmo para hacer agua de limón nos encontramos una operación que se repite varias veces que es la de exprimir los limones según la cantidad que tengamos, a este tipo de operación se le denomina una estructura iterativa.

2.1 Secuencial

Se da cuando una acción sigue a la otra. Es la más simple y la más común de todas y constituye la esencia de toda tarea programada.

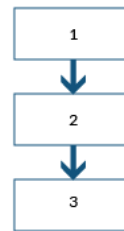


Fig. 9 Estructura secuencial

2.2 Selección simple

Este tipo de estructura se utiliza para tomar decisiones lógicas (verdadero o falso), también se conocen como estructuras de decisión o condición. Esta estructura evalúa una condición y en función del resultado se realiza una opción u otra (s).

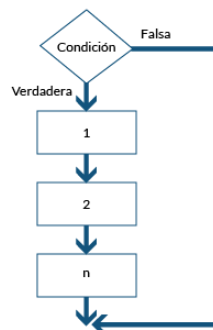


Fig. 10 Estructura de selección

2.3 Repetición: mientras-haz y repite-hasta

En los algoritmos encontramos acciones que se tienen que repetir un número determinado de veces según se requiera, para este tipo de iteraciones o ciclos se utilizan estructuras de control denominadas de iteración o repetición.

Hacer -mientras

En esta estructura se ejecutan primero las acciones asociadas al ciclo, se evalúa luego la condición y si resulta verdadera se repite el ciclo. Este tipo de estructuras se ocupa cuando se necesita que las acciones asociadas con el ciclo se realicen por lo menos una vez.

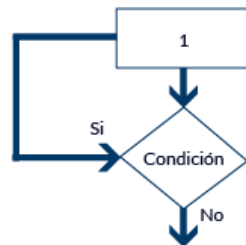


Fig. 11 Estructura hacer-mientras

Ejemplo

- Si tu equipo fue seleccionado para participar en un torneo puede jugar con otros equipos y mientras se mantenga en los primeros lugares, puede seguir participando en el torneo.

Desde – hasta

Esta estructura de control se utiliza cuando se conoce de antemano el número de ciclos. En un ciclo de este tipo se indica el valor inicial, el valor final y el incremento o decremento. A diferencia de los otros tipos de iteración como el de “Hacer – Mientras” no se requiere la evaluación explícita de una condición, sino que la condición está dada por el número de iteraciones indicadas previamente.

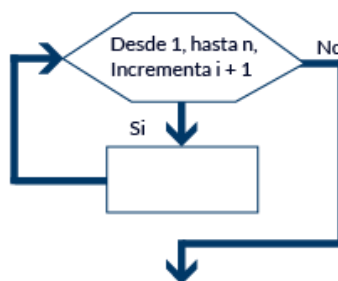


Fig. 12 Estructura desde – hasta

Para reforzar los aprendizajes referentes a Estructuras lógicas de control, observa el siguiente video el cual te ayudará a realizar la siguiente actividad de aprendizaje.

Video: Diagrama de flujo en PseInt, con condicional Si.

<https://www.youtube.com/watch?v=DIOBCZPFcaY&t=7s>

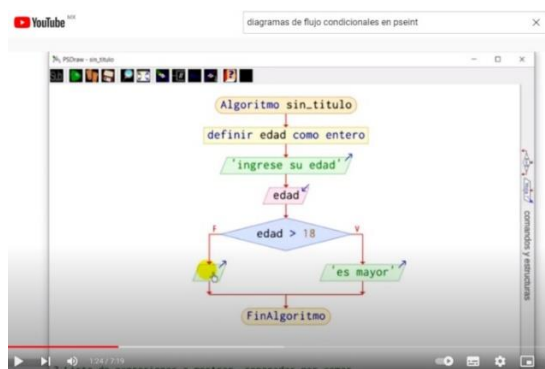


Fig. 13 Video de Estructuras lógicas en PSeInt

Actividad de aprendizaje 4

Instrucciones: Aplica la estructura selectiva doble y realiza el algoritmo y diagrama de flujo que determine si un alumno aprueba o reprueba una asignatura de acuerdo con su promedio, se sugieren estas variables, pero tú puedes utilizar otras variables si lo deseas, recuerda que debes realizar el diagrama en el programa de PSeInt y guardarlo con el nombre de promedio en la memoria USB.

- PP = Variable que almacena la calificación del primer parcial.
- SP = Variable que almacena la calificación del segundo parcial.
- TP = Variable que almacena la calificación del tercer parcial.
- Prom = Variable que almacena el promedio.

Algoritmo

1. Inicio
2. Ingresa la calificación del primer parcial
3. Leer PP
4. Ingresa la calificación del segundo parcial
5. Leer SP
6. Ingresa la calificación del tercer parcial
7. Leer TP
8. $Prom = (PP + SP + TP) / 3$
9. Si $Prom \geq 6$ entonces



10. Aprueba la asignatura
11. Si no
12. Reprueba la asignatura.
13. Fin

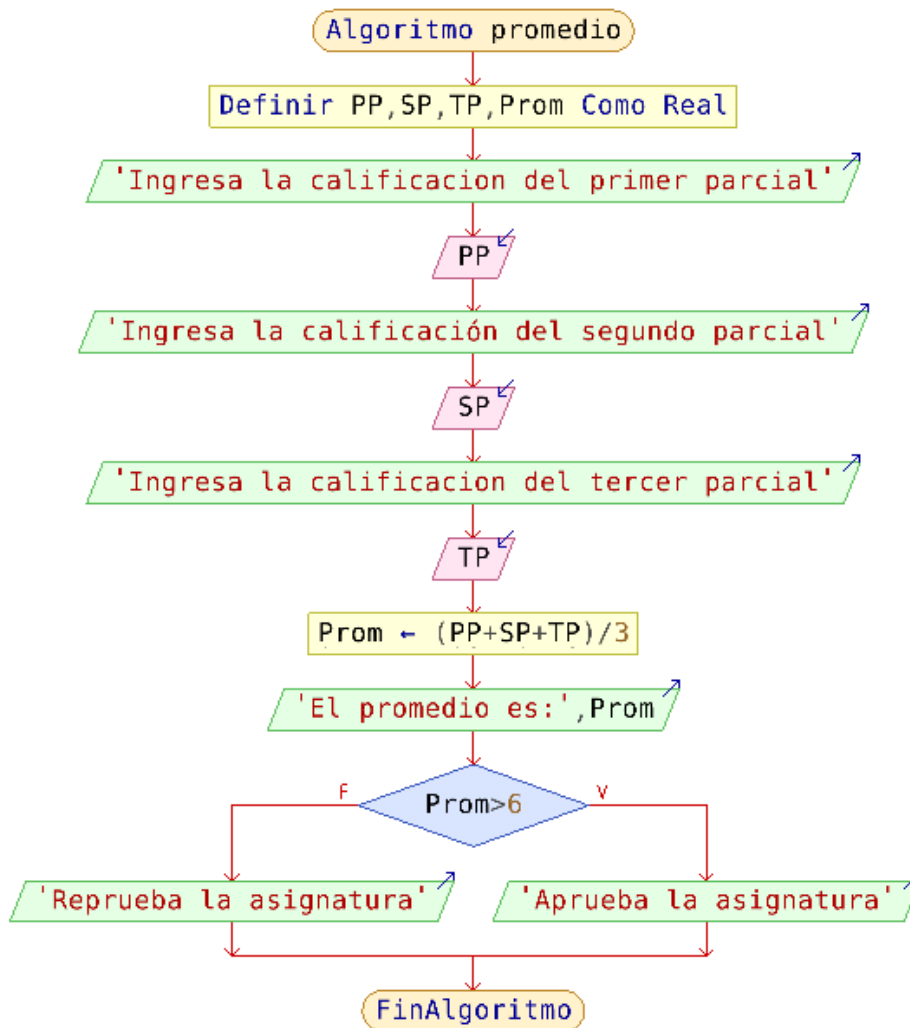


Fig. 14 Diagrama de flujo en PSeInt sobre la aprobación de asignatura

La ejecución del diagrama nos mostrará las siguientes ventanas:

```
PSelnt - Ejecutando proceso PROMEDIO
*** Ejecución Iniciada. ***
Ingresa la calificacion del primer parcial
> 9.2
Ingresa la calificación del segundo parcial
> 8.5
Ingresa la calificacion del tercer parcial
> 9.2
El promedio es:8.9666666667
Aprueba la asignatura
*** Ejecución Finalizada. ***
```

Fig. 15 Ventana que muestra si aprueba la asignatura

```
PSelnt - Ejecutando proceso PROMEDIO
*** Ejecución Iniciada. ***
Ingresa la calificacion del primer parcial
> 5.2
Ingresa la calificación del segundo parcial
> 4.3
Ingresa la calificacion del tercer parcial
> 3.6
El promedio es:4.3666666667
Reprueba la asignatura
*** Ejecución Finalizada. ***
```

Fig. 16 Ventana que muestra si reprueba la asignatura

3. Estructura de datos

El estudio de la estructura de datos es importante para la definición de la organización lógica de los mismos y para estar en condiciones de administrar la memoria real y la de la computadora. La elección de la estructura de datos dependerá de la organización del compilador y de la aplicación a ejecutar.

3.1 Simples

Los tipos de datos simples son los más utilizados para desarrollar programas básicos, los cuales se describen a continuación:

Numéricos

Este tipo de dato puede ser real o entero, dependiendo del tipo de dato que se vaya a utilizar.

- a) Enteros: Son números completos, no tienen componentes fraccionarios o decimales y pueden ser negativos o positivos.

Ejemplos de números enteros: 5, 6, -15, 4, 20, 17, 1340, 26

- b) Reales: Siempre tienen un punto decimal y pueden ser positivos o negativos. Un número real consta de un entero y una parte decimal.

Ejemplos de números reales: 0.08, 3739.41, 3.7452, -52.321, -8.12 3.0

Booleanos

También denominados booleanos, son datos que sólo pueden tomar uno de dos valores:

Cierto o verdadero (true) y falso (false).

Este tipo de datos se emplea para representar las alternativas (sí/no) a determinadas condiciones. Por ejemplo, cuando se quiere saber si un valor entero es par, la respuesta será verdadera o falsa.

Texto

Una cadena de caracteres es una sucesión de caracteres (letras, números u otros signos o símbolos). Si no se ponen restricciones al alfabeto, una cadena podrá estar formada por cualquier combinación finita de los caracteres disponibles (las letras de la 'a' a la 'z' y de la 'A' a la 'Z', los números del '0' al '9', el espacio en blanco ' ', símbolos diversos '!', '@', '%', etcétera).

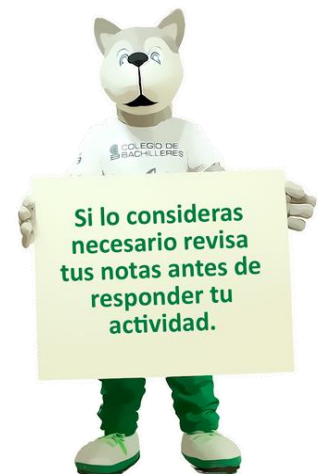
Actividad de aprendizaje 5

Instrucciones: Observa el siguiente video referente a estructuras de datos y realiza un resumen analítico en tu libreta de al menos 1 cuartilla completa, búscalo en este enlace.



<https://www.youtube.com/watch?v=TZtbYjJCviU>

Fig. 17 Video de estructura de datos





Simbología de los diagramas de flujo

<https://blogdelacalidad.com/diagrama-de-flujo-flujograma-de-proceso/>

http://uapas1.bunam.unam.mx/matematicas/diagramas_flujo/

Estructura secuencial

https://virtual.itca.edu.sv/Mediadores/dlp/23_estructuras_secuenciales.html

Estructura selección simple

<https://www.youtube.com/watch?v=4TKR47s1BHM>

Estructura de repetición

<https://www.youtube.com/watch?v=ZNkHWaiBEvA>

Estructura de datos

<https://www.youtube.com/watch?v=5k2DWMRTXMM>

Instrucciones: En relación con los contenidos vistos en el segundo corte, realiza esta autoevaluación siendo honesto, ya que lo importante es que obtengas los aprendizajes necesarios para este corte.

AUTOEVALUACIÓN		
Corte II		
Datos generales del proceso de evaluación		
Alumno:		
Instrucciones: Revisar las características que se solicitan y califique en la columna “Valor Obtenido” el valor asignado con respecto al “Valor del Reactivo”. En “OBSERVACIONES” haga las indicaciones que puedan ayudar a saber cuáles son las condiciones no cumplidas.		
Valor del reactivo	Característica para cumplir (Reactivo)	Valor Obtenido
20 %	Comprendí los contenidos propuestos para este corte.	
20 %	Realice las actividades de aprendizaje.	
15 %	Evaluó mi proceso de aprendizaje a partir de resolver las actividades planteadas.	
15 %	Desarrollo actividades extras para practicar lo aprendido.	
20 %	Los apuntes y actividades están ordenadas y resultas correctamente.	
10 %	Visito los enlaces sugeridos en el apartado conocer más para reforzar los conocimientos.	
100%	Calificación.	
Observaciones:		

Sitios web

IVE Consultores (2005). Diagramas de flujo de datos. Recuperado el 03 de mayo de 2022 de <https://iveconsultores.com/diagrama-de-flujo/>

UNAM (2017). Estructuras de control. Recuperado el 04 de mayo de 2022 de http://uapas1.bunam.unam.mx/matematicas/estructuras_de_control/

Universidad Carlos III Madrid (2016). Estructuras de datos simples. Recuperado el 05 de mayo de 2022 de http://www.cartagena99.com/recursos/alumnos/apuntes/TEMA_05.pdf

Videos

LearnWtutorials (21 dic. 2018). YouTube Estructuras de datos. Recuperado el 03 de mayo de 2022 de <https://www.youtube.com/watch?v=TZtbYjCviU>

La Geekipedia De Ernesto (15 de dic. 2016) YouTube Curso de programación desde cero. Recuperado el 04 de mayo de 2022 de <https://www.youtube.com/watch?v=4dKuDEWEAmo&list=PLyvsggKtwbLVva0XXvoqx91iaLmhgeNy7&index=3>

Tabla de imágenes corte 2

Figura	Referencia
4ª	IVE Consultores (2005). Diagramas de flujo de datos.
5ª	Elaboración propia
6ª	Elaboración propia
7ª	Franklin García. (2020) YouTube. Descargar e instalar PSeInt.
8ª	GabyB. (2019) YouTube. Cómo hacer un diagrama de flujo en PSeint.
9ª	Elaboración propia
10ª	UNAM (2017). Estructuras de control.
11ª	UNAM (2017). Estructuras de control.
12ª	UNAM (2017). Estructuras de control.
13ª	UNAM (2017). Estructuras de control.
14ª	Marcos Rios. (2021) YouTube. Sentencia o condicional If diagrama de flujo en PSeint.
15ª	Elaboración propia.
16ª	Elaboración propia.
17ª	Elaboración propia.
18ª	LearnWtutorials (21 dic. 2018). YouTube Estructuras de datos.

CORTE

3

Desarrollo de soluciones informáticas para web

Aprendizajes esperados:

Contenidos específicos

1. Utilidad del IDE WAMPServer

- 1.1 Instalación
- 1.2 Configuración
- 1.3 Administración del entorno de desarrollo web

2. Lenguaje de programación PHP

- 2.1 Estructura de un programa
- 2.2 Sintaxis
- 2.3 Estructuras de control
- 2.4 Variables
- 2.5 Constantes

Aprendizajes esperados

1. Prepara el equipo de cómputo para desarrollar una aplicación web con el IDE WAMP Server.
2. Elabora una página web mediante el uso de un lenguaje de programación estructurado del lado del servidor.



Al finalizar el corte el estudiante será capaz de desarrollar soluciones informáticas para Web en un lenguaje de programación estructurado; con la finalidad de resolver la problemática planteada por el cliente.

Para que logres desarrollar los aprendizajes esperados correspondientes al corte 3 es importante considerar los siguientes conocimientos referentes a páginas web:

- ✓ Estructura de una página
- ✓ Características de una página web
- ✓ Variables
- ✓ Constantes
- ✓ Operadores

Identifica lo que debes
saber para que la
comprensión de los
contenidos sea más fácil,
si descubres que has
olvidado algo ¡repásalo!



Instrucciones: Escribe dentro del recuadro el nombre del elemento que integra una página web.

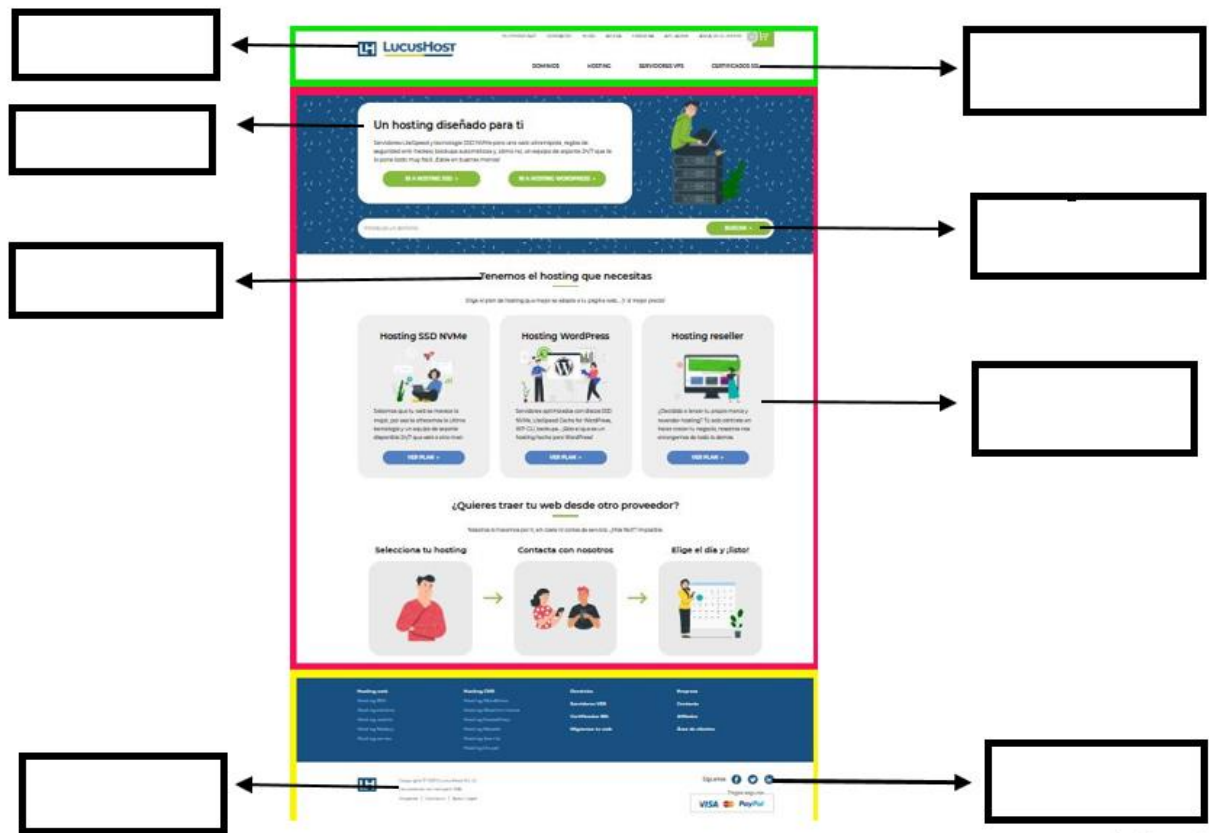


Fig. 18 Elementos de una página web

1. Utilidad del ide wampserver

WAMPServer es un entorno de desarrollo web para Windows en el cual se podrán crear aplicaciones web con Apache, PHP y base de datos en MySQL (motor de base de datos), se caracteriza porque puede ser usado de forma libre, es decir, no debemos de contar con alguna licencia de propietario el cual nos permita el uso de esta, ya que pertenece a la corriente de "open source".

Permite a usuarios sin conocimientos de sistemas instalar un servidor web de manera simple. Todo el proceso de instalación se realiza a través de un asistente por lo que las tareas de configuración del Web Server pasan desapercibidas.

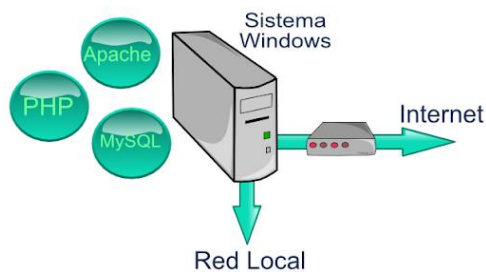


Fig. 19 Funcionamiento de WAMPServer

1.1 Instalación

Los pasos que puedes ver a continuación son válidos para cualquier versión de Windows. Para instalar WAMP en Windows basta con que sigas estos pasos:

1. Lo primero que vamos a hacer es descargar WAMP, así que accede a la página oficial de WAMP y haz clic en **Download** para ver las versiones de WAMP disponibles.



Fig. 20 Descarga de WAMPServer

- Ahora tendrás que seleccionar la versión de WAMP adecuada. Verás una versión de **32 bits (x32)** y otra de **64 bits (x64)**. Deberás descargarte una versión u otra dependiendo si tu sistema operativo es de 32 o de 64 bits.
- Ahora tendrás que descargar el instalador. Cuando hagas clic en una de las versiones, verás una serie de avisos indicando ciertas incompatibilidades con versiones anteriores de WAMP.

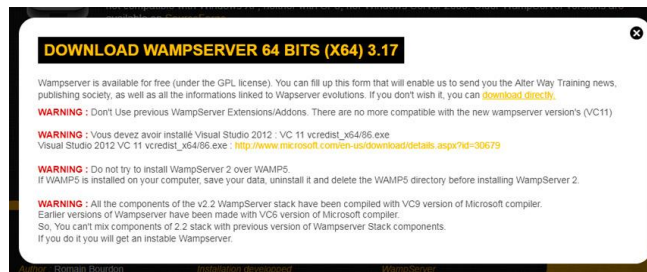


Fig. 21 Descargar el instalador

- Una vez que hayas descargado el instalador de WAMP debes ejecutarlo. Localiza el archivo descargado en la carpeta de descargas de tu navegador, que seguramente sea la carpeta de descargas de tu carpeta de usuario o el escritorio y ejecútalo.

1.2 Configuración

- Primero tendrás que seleccionar el idioma de WAMP. Si tu idioma no está en la lista (el español no lo está por ahora), selecciona inglés o aquel con el que mejor te defiendas. Haz clic en OK para continuar.

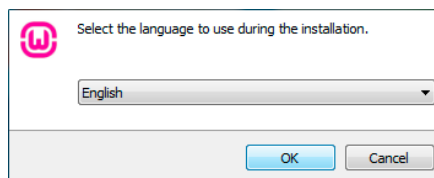


Fig. 22 Selección del idioma

- Marca la casilla correspondiente «I accept the agreement» para aceptar el acuerdo de licencia de WAMP. Continúa haciendo clic en Siguiente (Next).
- Ahora verás un mensaje en el que se indican los requisitos de WAMP y aquellos componentes que necesitas tener instalados en tu sistema para que WAMP funcione correctamente.

8. Introduce el nombre del directorio en el que quieres instalar WAMP. Es más que recomendable instalar WAMP en una carpeta del directorio raíz de tu disco duro.

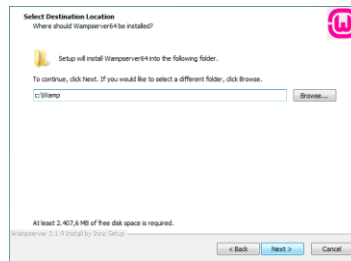


Fig. 23 Directorio raíz

9. Seguidamente, introduce la carpeta del menú de inicio en la que quieres que se creen los accesos directos a WAMP. Haz clic en Siguiente (Next) para continuar.
10. Finalmente, confirma los detalles introducidos y haz clic en Siguiente (Next) para continuar. La instalación dará comienzo, pero durante el proceso tendrás que seleccionar el navegador que quieres que WAMP use por default.

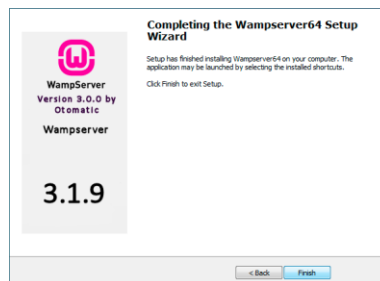


Fig. 24 Instalación completa

1.3 Administración del entorno de desarrollo web

Se refiere a la utilización de las herramientas del IDE (Integrated Development Environment o Entorno de Desarrollo Integrado) como son Apache, MySQL, PHPMyAdmin y PHP, para desarrollar soluciones web del lado del servidor.

Actividad de aprendizaje 1

Instrucciones: Observa el siguiente video referente a la instalación y configuración de WAMPServer (Del minuto 0 al minuto 9:52) y descarga el programa en tu computadora, adjunto el enlace del video:

<https://www.youtube.com/watch?v=mx67Re19XIQ>.



Fig. 25 Instalación y configuración de WAMPServer

2. Lenguaje de programación PHP

PHP es un lenguaje de código abierto muy popular, adecuado para desarrollo web y que puede ser incrustado en HTML. Es popular porque un gran número de páginas y portales web están creadas con PHP. Código abierto significa que es de uso libre y gratuito para todos los programadores que quieran usarlo. Incrustado en HTML significa que en un mismo archivo vamos a poder combinar código PHP con código HTML, siguiendo ciertas reglas.

PHP se utiliza para generar páginas web dinámicas. Hay que recordar que llamamos página estática a aquella cuyos contenidos permanecen siempre igual, mientras que llamamos páginas dinámicas a aquellas cuyo contenido no es el mismo siempre. Por ejemplo, los contenidos pueden cambiar con base a los cambios que haya en una base de datos, de búsquedas o aportaciones de los usuarios, etc.

2.1 Estructura de un programa

Un archivo PHP normalmente suele contener HTML y trozos de código PHP. Por ello antes de estudiar PHP es necesario haber estudiado HTML. A veces, se dice que PHP es un lenguaje “embebido” en HTML, es decir, incrustado en código HTML. Todos los fragmentos de código PHP deben comenzar con la siguiente cadena de caracteres: `<?php` y finalizar con `?>`.


```

01 <html>
02
03     <head>
04
05         <title>Mi primera página web</title>
06
07     </head>
08
09     <body>
10
11         <?php echo "Hola a todos" ?>
12
13     </body>
14
15 </html>

```

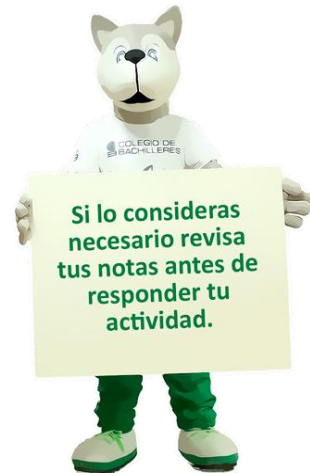


Fig. 26 Estructura de una página en PHP

Apertura y cierre

PHP tiene tag de apertura y cierre para poder identificar en qué momento deja de ser HTML y ahora se utiliza PHP.

<?php es el tag de apertura.

?> es el tag de cierre.

2.2 Sintaxis

Conjunto de reglas que indican la aplicación de las palabras claves de un lenguaje de programación, considerando una estructura definida y el uso de la lógica computacional para escribir el código fuente de una solución informática.

2.3 Estructuras de control

Un script PHP está construido a partir de sentencias. Una sentencia puede ser una asignación, una llamada a una función, un loop, una sentencia condicional o una sentencia vacía. Las sentencias normalmente finalizan con un punto y coma.

Las estructuras de control son sentencias que permiten controlar cómo el código fluye en nuestro script basándose en ciertos factores, algunas de ellas son:

If

La estructura de control **if** permite ejecutar un fragmento de código si la expresión provista junto con ella se evalúa como verdadera.

```

<?php
if ($x > $y) {
    echo "$x es mayor que $y";
}

```

Las sentencias if se pueden incluir unas dentro de otras indefinidamente.

else

Sirve para ejecutar una sentencia cuando otra no se cumple. **else** extiende una sentencia if, y se ejecuta cuando ésta es **false**. Siguiendo el ejemplo anterior:

```
<?php
if ($x > $y) {
    echo "$x es mayor que $y";
} else {
    echo "$y es mayor que $x";
}
```

while

Se ejecutan las sentencias dentro del while siempre y cuando se evalúen como true. El valor de la expresión se comprueba cada vez al inicio del loop o ciclo repetitivo, y la ejecución no se detendrá hasta que finalice la iteración (cada vez que PHP ejecuta las sentencias en un loop es una iteración). Si la expresión while se evalúa como false, las sentencias no se ejecutarán ni siquiera una vez.

También es posible agrupar varias instrucciones while dentro de una.

```
<?php
$i = 1;
while($i <= 10){
    echo $i;
    $i++;
}
```

do-while

Muy similares a los loops **while**, simplemente aquí la expresión para el loop se verifica al final en lugar de al principio, esto garantiza que el código se ejecute por lo menos la primera vez.

```
<?php
$i = 0;
do {
    echo $i;
} while ($i > 0);
```

Este loop se ejecutaría sólo una vez, ya que después no cumple la condición.

for

Los loops **for** son los más complejos en PHP.

```
<?php
for ($i = 1; $i <= 10; $i++) {
    echo $i;
} // Devuelve 123456789
```

- Las expresiones o conjunto de expresiones van separadas por punto y coma; y sólo hay 3.
- La primera expresión, `$i = 1`, se ejecuta una vez incondicionalmente al comienzo del bucle.
- La segunda expresión, `$i <= 10`, es una condición, si es true, se ejecutará la tercera expresión.
- La tercera expresión, `$i++`, es la acción para realizar si se cumple la segunda expresión.

switch

switch es como una serie de sentencias if. Es útil para comparar una misma **variable** o **expresión** con valores diferentes y ejecutar un código diferente a otro dependiendo de esos valores.

```
<?php
switch ($i) {
    case "perro":
        echo "\$i es un perro";
        break;
    case "gato":
        echo "\$i es un gato";
        break;
    case "avestruz":
        echo "\$i es un avestruz";
        break;
}
```

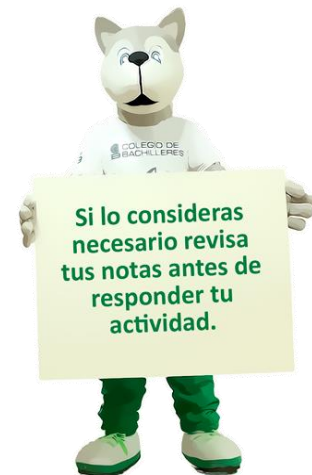
Actividad de aprendizaje 2

Instrucciones: Observa el siguiente video a partir del minuto 9:53, este explica cómo crear una página web sencilla utilizando WAMPServer, una vez observado el video responde el cuestionario y realiza la página sugerida, guárdala en una memoria USB ya que lo entregaras junto con la guía al profesor, adjunto el enlace del video:
<https://www.youtube.com/watch?v=Ja9UVEgAzEw>

Nota: Utiliza el bloc de notas para realizar esta actividad.



Fig. 27 Video para crear una página en PHP



2.4 Variables

Las variables tendrán el signo de dólares al principio, y deberá seguir ciertas reglas, por ejemplo:

- Debe empezar con una letra o guion bajo.
- Puede tener números, siempre y cuando no sea en la primera posición del nombre de la variable
- Al ser un lenguaje débil (No usa tipos de variables) no se coloca el tipo de variable como en otros lenguajes, y puede cambiar el tipo de variable en tiempo de ejecución.

```
1 | $contador = 0; //Correcto
2 |
3 | $_nombre = 'Angelos'; //Correcto
4 |
5 | $formato25 = TRUE; //Correcto
6 |
7 | $2var = 1.5; //Incorrecto
```

Fig. 28 Ejemplo de variables

Dependiendo de la información que contenga, una variable puede ser considerada de uno u otro tipo:

Variables numéricas

Este tipo de variables almacena cifras, números, que pueden tener dos clasificaciones distintas:

- Enteros \$entero=2002; Números sin decimales
- Reales \$real=3.14159; Números con o sin decimal

Variables alfanuméricas

Este tipo de datos almacena textos compuestos, cadenas de caracteres, que pueden contener letras, símbolos y números o cifras.

- Cadenas Almacenan variables alfanuméricas \$cadena="Hola amigo";

Booleanas

Este tipo de variables almacena un valor lógico, que puede valer verdadero o falso. Es muy común en la programación este tipo de variables booleanas.

- Booleano verdadero \$verdadero = true;
- Booleano falso \$falso = false;

2.5 Constantes

Es un identificador que contiene un valor que no es modificable durante la ejecución. Tiene las mismas reglas de identificación que las variables.

```
define ("cons", "dato");
```

Actividad de aprendizaje 3

Instrucciones: Elabora la siguiente página PHP con el bloc de notas como editor web, este ejercicio consiste en comparar las variables y determinar cuál es mayor, menor o igual utilizando la estructura if – else, recuerda guardar la actividad en una memoria USB para entregarla junto con la guía.

```
<html>
<head>
<title> Trabajando con la sentencia if..elseif </title>
</head>
<body>

<?php
//Declaramos e inicializamos variables
$a = 2;
$b = 3;

if ($a > $b) {
    echo "a es mayor que b";
} elseif ($a == $b) {
    echo "a es igual que b";
} else {
    echo "a es menor que b";
}
?>
</body>
</html>
```

Fig. 29 Ejemplo de página con if-else en PHP



Cuestionario

1. ¿Qué significan cada uno de los estados de WAMPServer?
2. Menciona 6 editores web.
3. ¿Dónde se almacenan las páginas de PHP?
4. ¿Por qué se asigna el nombre de índice?
5. ¿Qué servidor se está utilizando en el video?
6. ¿Cuáles son las etiquetas de apertura y cierre en PHP?
7. ¿Qué hace el servidor al escribir el código?
8. ¿Cuál es el procedimiento para visualizar la página en el navegador?



Instalación y configuración de WAMPServer

<https://www.ecodeup.com/instalar-wampserver-windows/>

Administración del entorno de desarrollo web

<https://www.youtube.com/watch?v=6ThsAz2QN3A>

Estructura y sintaxis de PHP

<https://blog.angelinux-slack.net/2016/11/11/estructura-basica-de-una-pagina-en-php/>

<https://www.youtube.com/watch?v=r3cCDMxYo3M>

Estructuras de control

<https://code.tutsplus.com/es/tutorials/php-control-structures-and-loops--cms-31999>

Variables y constantes en PHP

<https://www.youtube.com/watch?v=qM75SIRIs6g>

Instrucciones: En relación con los contenidos vistos en el tercer corte, selecciona alguno de los logros que alcanzaste en el desarrollo de este corte de acuerdo con cada contenido y escribe un breve comentario en el logro, se honesto ya que esta autoevaluación representa el aprendizaje obtenido de ese tema de acuerdo con tus esfuerzos y la valoración de tu trabajo.

Contenido	Logros		
	Lo logre	Lo estoy intentando	Necesito ayuda
Instalación y Configuración de WAMPServer			
Administración del entorno de desarrollo web			
Estructura de un programa en PHP			
Sintaxis de PHP			
Variables de PHP			
Constantes de PHP			
Estructuras de control			

Sitios web

Neoguías (2020). Instalar WAMPServer en windows. Recuperado el 04 de mayo de 2022 de <https://www.neoguias.com/instalar-wamp/>

Codegeando (2016). WAMPServer. Recuperado el 04 de mayo de 2022 de <https://codegeando.blogspot.com/2013/03/php-wampserver-definicion-instalacion-y.html>

Angelinix (2015). Estructura básica de una página en PHP. Recuperado el 04 de mayo de 2022 de <https://blog.angelinux-slack.net/2016/11/11/estructura-basica-de-una-pagina-en-php/>

Diego Lázaro (2018). Estructuras de control en PHP. Recuperado el 04 de mayo de 2022 de <https://diego.com.es/estructuras-de-control-en-php>

Videos

El canal Dany (23 marzo 2020). YouTube Instalar y configurar WAMPServer. Recuperado el 04 de mayo de 2022 de <https://www.youtube.com/watch?v=mx67Re19XIQ>

Tabla de imágenes corte 3

Figura	Referencia
19a	Tecno analista (2017). Elementos de una página web.
20a	Observatorio tecnológico (2018). Funcionamiento de WAMPServer.
21a	Dani Sánchez (2016). Descarga de WAMPServer.
22ª	CualHost (2018). Páginas web
23ª	Codegeando (2016). WAMPServer.
24ª	Neoguías (2020). Instalar WAMPServer en Windows
25ª	Neoguías (2020). Instalar WAMPServer en Windows
26ª	El canal Dany (23 marzo 2020). YouTube Instalar y configurar WAMPServer.
27ª	Angelinix (2015). Estructura básica de una página en PHP
28ª	Angelinix (2015). Estructura básica de una página en PHP
29ª	Pildorasinformaticas (2015). YouTube Crear una página web en PHP
30ª	Institutoonline (2020). Página web con if-else en PHP.

Aprendizaje Basado en Problemas

Ejercicio 1

La tienda Liverpool por motivo del buen fin, tiene la promoción de que si el monto en la compra de tres productos del departamento de línea blanca (refrigeradores, estufas, lavadoras, etc.) es mayor a \$ 15,000 tendrá un descuento del 15% de lo contrario solo será el 5%, deberá mostrarse el total a pagar de los productos comprados con el respectivo descuento.

Instrucciones: Realiza el algoritmo y diagrama de flujo en el programa PSeint que solucione la problemática planteada, se deben mostrar 3 mensajes para ingresar el costo de los productos, 1 mensaje con el total a pagar y el pago con descuento incluido, recuerda guardar el diagrama en una memoria USB.

La pantalla que se debe mostrar es la siguiente:

```
PSelnt - Ejecutando proceso BUENFIN
*** Ejecución Iniciada. ***
TIENDA LIVERPOOL
Ingresa el costo del primer producto
> 4600
Ingresa el costo del segundo producto
> 6500
Ingresa el costo del tercer producto
> 8200
El total a pagar es de:19300
Tiene un descuento del 30%
El pago con descuento es de:13510
*** Ejecución Finalizada. ***
```

Ejercicio 2

Elabora una página web en php que determine si un alumno aprueba o reprueba la asignatura de Modelado de sistemas, con base al promedio obtenido en los tres cortes, recuerda guardar la página en una memoria USB para entregarla con la guía.

PLAN 2014

ACTUALIZADO



Somos Lobos Grises,
somos Bachilleres



D.R. Colegio de Bachilleres. 2021-2022