



Universidad Autónoma
del Estado de México



Computación aplicada

Libro de texto basado en competencias

Carlos Alberto Corro Rogel · Fabiola López Vargas
Gabriela Patricia González Navarro
Rebeca Pérez Villegas · Isaura Rivera Jiménez

6°
Semestre

Computación aplicada

Libro de texto basado en competencias

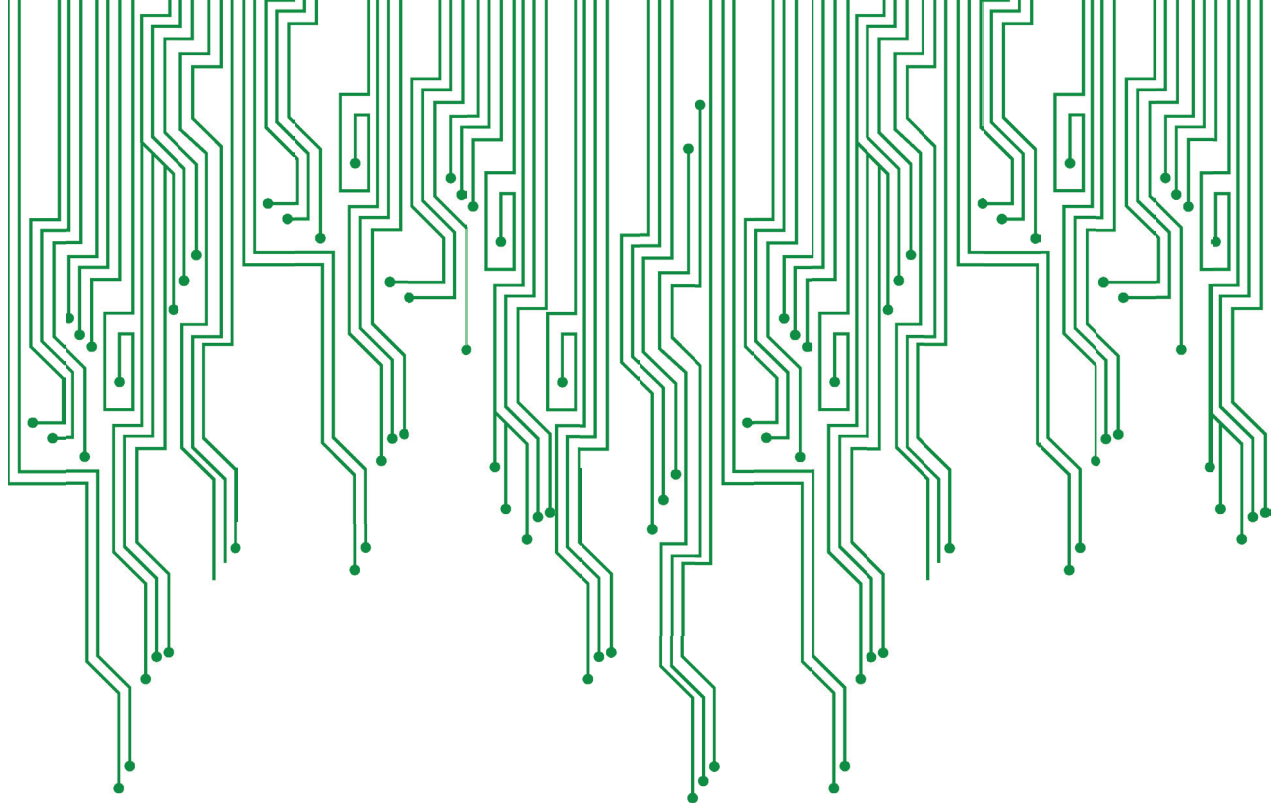
DIRECCIÓN DE PUBLICACIONES UNIVERSITARIAS
Editorial de la Universidad Autónoma del Estado de México

Doctor en Ciencias e Ingeniería Ambientales
Carlos Eduardo Barrera Díaz
Rector

Doctora en Humanidades
María de las Mercedes Portilla Luja
Secretaria de Difusión Cultural

Doctor en Administración
Jorge Eduardo Robles Alvarez
Director de Publicaciones Universitarias

Doctora en Investigación Educativa
Eva Lilia García Escobar
Directora de Estudios de Nivel Medio Superior



Computación aplicada

Libro de texto basado en competencias

Carlos Alberto Corro Rogel
Fabiola López Vargas
Gabriela Patricia González Navarro
Rebeca Pérez Villegas
Isaura Rivera Jiménez

Primera edición, febrero 2022

COMPUTACIÓN APLICADA

Libro de texto basado en competencias

Autores

Carlos Alberto Corro Rogel

Plantel Lic. Adolfo López Mateos de la Escuela Preparatoria de la UAEM

Fabiola López Vargas

Plantel Lic. Adolfo López Mateos de la Escuela Preparatoria de la UAEM

Gabriela Patricia González Navarro

Plantel Ignacio Ramírez Calzada de la Escuela Preparatoria de la UAEM

Rebeca Pérez Villegas

Plantel Cuauhtémoc de la Escuela Preparatoria de la UAEM

Isaura Rivera Jiménez

Plantel Nezahualcóyotl de la Escuela Preparatoria de la UAEM

Universidad Autónoma del Estado de México

Av. Instituto Literario 100 Ote.

Toluca, Estado de México

C. P. 50000

Tel: (52) 722 277 38 35 y 36

<http://www.uaemex.mx>

El contenido de este material es responsabilidad de los autores, así como el tratamiento basado en competencias realizado a los libros de texto. Queda prohibida la reproducción o transmisión total o parcial del contenido de la presente obra, sin el consentimiento previo y por escrito del editor. La Universidad Autónoma del Estado de México se deslinda de cualquier acción legal derivada de este material.

® Derechos reservados, 2018

ISBN: 978-607-633-393-8

Hecho en México

Responsable del Programa de Diseño y Producción Editorial: L. D. G. Miguel Angel Conzuelo Endeje

Corrección de estilo: Rocío Franco López

Portada e interiores: L. D. G. Gustavo Osvaldo Godínez Ferreyra



MENSAJE

A través de su historia, la Universidad Autónoma del Estado de México ha tenido como uno de sus objetivos fundamentales lograr que el estudiantado reciba una educación basada en el libre pensamiento, apelando a un sentido científico. La Escuela Preparatoria de la UAEM continuará mostrándose como un punto de inflexión para la comunidad estudiantil por la calidad de la educación que se ofrece al tiempo que se enseña a adquirir conocimientos, valores, habilidades y destrezas que favorezcan el entendimiento de nuestro entorno social y ambiental.

La responsabilidad de la Universidad Autónoma del Estado de México es asegurar que la educación que se imparte en la Escuela Preparatoria contribuya a ampliar el horizonte personal y profesional de los estudiantes. El enfoque para la apropiación del conocimiento parte de aprovechar la experiencia de vida y desde la autonomía de pensamiento y la reflexión lograr una comprensión más amplia de la realidad.

El material educativo que tienes en tus manos ha sido elaborado colegiadamente entre especialistas de cada una de las áreas del conocimiento. La creatividad, el pensamiento crítico y reflexivo, así como el espíritu científico del claustro de docencia e investigación se reúne en esta antología con el fin de que te acerques al conocimiento con una mirada crítica. Este ejemplar ha sido elaborado con el fin de establecer un aprendizaje dialógico que estimule tu curiosidad, que te sirva como guía, y que invariablemente te provoque pasión por el conocimiento.

El objetivo es lograr que recibas una educación integral e incluyente que incida de manera positiva en tu trayectoria escolar. Por ello es importante que asumas un compromiso personal, empeño y dedicación que te permita apropiarte del conocimiento.

Patria, Ciencia y Trabajo

Doctor en Ciencias e Ingeniería Ambientales
Carlos Eduardo Barrera Díaz
Rector

Contenido

Introducción	9
Módulo 1. Metodología de programación	11
1.1. Resolución de problemas	14
1.1.1. Pasos para la solución de problemas	15
1.1.2.1. Análisis del problema	15
1.1.2.2. Selección de la mejor alternativa	15
1.1.2.3. Diagrama de flujo	15
1.1.2.4. Prueba de escritorio	17
1.2. Diseño de algoritmos	18
1.2.1. Concepto de algoritmo	18
1.2.2. Características de los algoritmos	18
1.2.3. Tipos datos	18
1.2.3.1. Numérico	18
1.2.3.2. Literal	19
1.2.3.2. Constante	19
1.2.4. Operadores aritméticos y su prioridad; relacionales y lógicos	19
1.2.5. Instrucciones y sentencias	22
1.3. Herramientas para el diseño de algoritmos	29
1.3.1. Pseudocódigo	29
1.3.2. Transcripción	29
1.3.3. Pruebas de computador	29
1.3.4. Documentación externa	30
Módulo 2. Programación y aplicaciones	33
2.1. Lenguaje de programación y proyecto	36
2.2. Editor del lenguaje de programación	37
2.2.1. Entorno	38
2.3. Codificación de instrucciones básicas.	
Sintaxis de sentencias: lectura y escritura de datos	42
2.3.1. Estructuras de control	44
2.3.1.1. Estructuras secuenciales, selectivas y repetitivas	45

2.4. Compilación	50
2.5. Depuración	50
2.6. Ejecución	50
Módulo 3. Proyecto web	53
3.1. Proyecto web	56
3.1.1. Concepto de un proyecto web	56
3.2. Componentes básicos	56
3.2.1. Introducción	57
3.2.2. Objetivo del proyecto web	58
3.2.3. Recursos disponibles	58
3.2.4. Beneficio esperado	60
3.2.5. Diseño estructurado	60
3.2.6. Cronograma de actividades	63
Módulo 4. Implementación del sitio web	65
4.1. Software desarrollador de sitios web	68
4.1.1. Entorno de trabajo para desarrollar un sitio web	74
4.1.2. Elementos de una página web (tablas, texto, botones, imágenes, vínculos, hipervínculos, video, sonido, animaciones, marcos, etc.)	83
4.1.3. Publicación del sitio	92
4.1.4. Mantenimiento	93
Fuentes	98

Introducción

El presente texto está totalmente apegado al programa de estudios más actualizado de la Universidad Autónoma del Estado de México, en la materia de Computación Aplicada, destacando que dicha materia se encuentra dentro del área de informática cuyo campo ha tenido un auge importante en la actualidad.

Con este texto se pretende guiar a los estudiantes del sexto semestre de preparatoria introduciéndolos al ámbito de los algoritmos y la programación para que apliquen estas herramientas en las actividades de su vida cotidiana y para que les sirva como elemento didáctico para lograr en sus estudiantes un aprendizaje significativo.

En el módulo 1 se analizan los conocimientos básicos que pueden apoyar al estudiante para resolver situaciones cotidianas por medio del planteamiento de algoritmos.

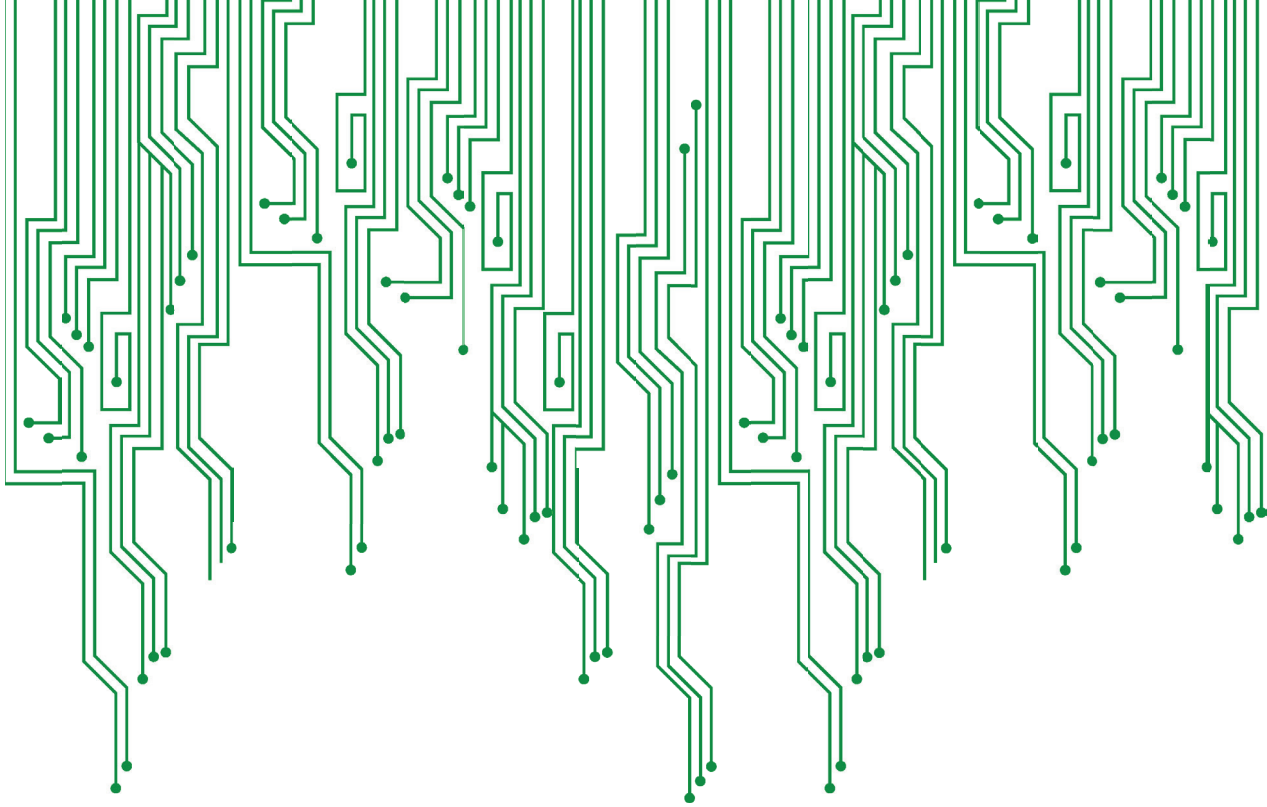
El alumno será capaz de tomar decisiones para poder plantear la solución de un problema mediante una metodología; por otra parte, se les enseña a interpretar dichos problemas de distintas formas utilizando pseudocódigos y diagramas de flujo.

En el módulo 2 se propone una metodología para la solución de problemas por medio de una computadora, el objetivo principal radica en que los estudiantes puedan ser analíticos al momento de enfrentarse a cualquier adversidad e identifiquen los elementos que componen un lenguaje de programación.

Para que los estudiantes puedan comprobar que sus planteamientos son formulados correctamente se utilizará un programa de uso libre denominado DevC++ cuya función es la interpretación de los códigos.

Para el módulo 3 se enseña a establecer un proyecto web con una metodología que les ayude a proponer la elaboración de un buen proyecto, bajo los requerimientos mínimos para desarrollar un sitio web.

En el módulo 4 veremos el diseño y la creación de páginas web, y cómo pueden poner en marcha su propia página.



Módulo 1

Metodología de programación



Imagen: pxhere.com

Utiliza una metodología de programación para desarrollar alternativas de solución de problemas cuantitativos de su entorno utilizando las tecnologías de la información y la comunicación.





Competencias disciplinares

Comunicación Básicas

5. Analiza las relaciones entre dos o más variables de un proceso social o natural para determinar o estimar su comportamiento.
10. Identifica e interpreta la idea general y el posible desarrollo de un mensaje oral y escrito en una segunda lengua, recurriendo a conocimientos previos, elementos no verbales y contexto cultural.
11. Se comunica en una lengua extranjera mediante el discurso lógico, oral o escrito, congruente con la situación comunicativa.

Extendidas

8. Valora la influencia de los sistemas y medios de comunicación en su cultura, su familia y su contexto socioculturales diferentes.
10. Analiza los beneficios e inconvenientes del uso de las tecnologías de la información y la comunicación para optimización de las actividades cotidianas.

Competencias genéricas

4. Escucha, interpreta y emite mensajes pertinentes en distintos contextos mediante la utilización de medios, códigos y herramientas apropiadas.
5. Desarrolla innovaciones y propone soluciones a partir de métodos establecidos.
 - 5.1. Sigue instrucciones y procedimientos de manera reflexiva, comprendiendo como cada uno de sus pasos contribuye al alcance de un objetivo.
 - 5.2. Ordena información de acuerdo a categorías, jerarquías y relaciones.
 - 5.6. Utiliza las tecnologías de la información y la comunicación para procesar e interpretar información.

Actividad diagnóstica

Instrucciones: Subraya la respuesta correcta.

1. Situación en la que se requiere de un plan de acción para su solución
 a) Programa b) Problema c) Sistema operativo
2. Es la representación gráfica de un algoritmo
 a) Diagrama de flujo b) Pseudocódigo c) Problema
3. Tipo de problema que involucra valores numéricos
 a) Cualitativos b) Cuantitativos c) Personales
4. Tipo de problema en el cual se evalúan gustos y preferencias
 a) Cuantitativos b) Cualitativos c) Personales
5. Consiste en dar diferentes datos de entrada al programa, para seguir la secuencia indicada en el diagrama, hasta obtener los resultados
 a) Algoritmo b) Problema c) Prueba de escritorio
6. Es un plan preciso para ejecutar una secuencia de acciones ordenadas para alcanzar un propósito propuesto en un tiempo finito
 a) Algoritmo b) Problema c) Prueba de escritorio
7. Son ejemplos de operadores aritméticos
 a) < , > , <= , >= , <> , = b) y , o , no c) + , - , * , / , mod
8. Son ejemplos de operadores relacionales
 a) < , > , <= , >= , <> , = b) y , o , no c) + , - , * , / , mod
9. Son ejemplos de operadores lógicos
 a) < , > , <= , >= , <> , = b) y , o , no c) + , - , * , / , mod
10. Es una estructura de control que se ejecuta de acuerdo con una condición
 a) Secuencial b) Alternativa c) Repetitiva



Actividad



Introducción

Todos los días el ser humano trata de darle solución a cada problema que se le presenta o de mejorar las soluciones disponibles. Para algunos problemas fundamentales ha encontrado soluciones brillantes que consisten en una serie de acciones, que siempre que se hagan de manera ordenada y precisa conducen a la respuesta correcta. Algunas de esas soluciones han requerido el trabajo, la inteligencia y la persistencia de muchas generaciones. Hoy la sociedad cuenta con ese legado de soluciones, además de las máquinas capaces de ejecutarlas precisa y velozmente.

El desarrollo de algoritmos es un tema fundamental en el diseño de programas, por lo que el alumno debe tener buenas bases para poder desarrollar de manera fácil y rápida sus programas.

La computadora no solo es una máquina que puede realizar procesos para darnos resultados sin que tengamos la noción exacta de las operaciones que hace para llegar a esos resultados. Además de lo anterior, con la computadora también podemos diseñar soluciones a la medida de problemas específicos que se nos presenten. Más aún, si estos involucran operaciones matemáticas complejas y/o repetitivas, o requieren del manejo de un volumen muy grande de datos.

1.1. Resolución de problemas

Metodología para la solución de problemas por medio de la computadora

Es el conjunto de métodos y técnicas disciplinadas que ayudan al desarrollo de programas.

Problema: situación en la que los elementos con que se cuenta requieren de un plan de acciones para su solución (Cairó, 2005).

Tipos de problemas

A) Problemas cuantitativos

Son problemas que involucran para su solución cantidades numéricas.

Ejemplo: matemáticas financieras, económicas y algunas científicas.

La solución de todo problema cuantitativo requiere de utilizar variables y constantes.

B) Problemas cualitativos

Son aquellos que solo requieren apoyarse en la toma de decisiones y que no tienen una fórmula establecida para alcanzar una solución. Ejemplo: seleccionar una carrera, un proceso legal, entre otros.

1.1.1. Pasos para la solución de problemas

Los pasos que exige el desarrollo de un sistema son los siguientes:

1.1.1.1. Análisis del problema

Una vez entendido el problema para resolverlo es preciso analizar:

- Los datos o resultados que se esperan.
- Los datos de entrada que nos suministran.
- El proceso al que se requiere someter esos datos con el fin de obtener los resultados esperados.
- Áreas de trabajo, fórmulas y otros recursos necesarios.

Una recomendación muy práctica es ponerse en el lugar de la computadora y analizar que es necesario, qué ordenarle y en qué secuencia, para poder producir los resultados esperados. También da buenos resultados hacer similitudes con la labor de un empleado que hace el mismo trabajo que deseamos programarle a la computadora.

1.1.1.2. Selección de la mejor alternativa

Si analizamos el problema, es posible que haya varias soluciones; lo importante es determinar cuál es la mejor alternativa: la que produce los resultados esperados en el menor tiempo y con el menor costo. Aquí también es muy válido el principio de que las cosas siempre se pueden hacer mejor.

1.1.1.3. Diagrama de flujo

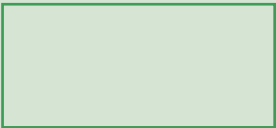
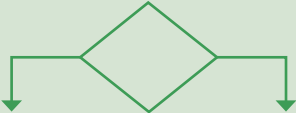
Una vez que sabemos cómo resolver el problema, pasamos a dibujar gráficamente la estructura de la alternativa seleccionada. Eso es precisamente un diagrama de flujo: la representación gráfica de una secuencia lógica de pasos a cumplir por la computadora para producir un resultado esperado.

La experiencia nos ha demostrado que resulta muy útil trasladar esos pasos lógicos, planteados en el diagrama, a frases que indiquen lo mismo; es decir, hacer una codificación del programa utilizando instrucciones en español. Como si le estuviéramos hablando a la computadora. Esto es lo que denominaremos **algoritmo** o **pseudocódigo**.

Cuando logremos la habilidad para desarrollar programas es posible que no tengamos que hacer un diagrama de flujo; en su lugar, podremos hacer directamente el pseudocódigo del programa (Joyanes, 2008).

A continuación relaciona las columnas para describir la simbología utilizada para el diseño de un diagrama de flujo.



Símbolo	Significado
	Inicio / Fin. Determina el inicio y el fin del algoritmo.
	Entrada de datos. Representa el ingreso de los datos al programa.
	Proceso. Representa las operaciones que se hacen para obtener el resultado.
	Decisión. Representa las operaciones de tipo lógico que contiene el algoritmo.
	Imprimir resultados. (Mostrar datos en pantalla).
	Flujo de datos. Indica la secuencia del flujo de operaciones.
	Conectores. Se utilizan para conectar bloques del diagrama cuando es grande y se hace necesario dividirlo.

Reglas para la construcción de diagramas de flujo

1. Todo diagrama debe tener un inicio y un fin.
2. Las líneas utilizadas para indicar la dirección del flujo del diagrama deben ser rectas, verticales y horizontales.
3. Todas las líneas utilizadas para indicar la dirección del flujo del diagrama deben estar conectadas. La conexión puede ser un símbolo que exprese lectura, escritura, proceso, decisión, impresión, conexión o fin de diagrama.
4. El diagrama de flujo debe ser construido de arriba hacia abajo y de izquierda a derecha.
5. La notación utilizada en el diagrama de flujo debe ser independiente del lenguaje de programación.
6. Cuando hacemos una tarea compleja es conveniente poner comentarios que expresen o ayuden a entender lo que hicimos.
7. Si el diagrama de flujo requiriera más de una hoja para su construcción debemos utilizar los conectores adecuados y enumerar las páginas convenientemente.
8. No puede llegar más de una línea a un símbolo.

1.1.1.4. Prueba de escritorio

Para cerciorarnos de que el diagrama (y/o el pseudocódigo) está bien, y para garantizar que luego el programa que codifiquemos también funcione correctamente, es conveniente someterlo a una prueba de escritorio. Esta prueba consiste en dar diferentes datos de entrada al programa y seguir la secuencia indicada en el diagrama hasta obtener los resultados esperados.

El análisis de estos nos indicará si el diagrama está correcto o si hay necesidad de hacer ajustes. Se recomienda dar diferentes datos de entrada y considerar todos los posibles casos, aun los de excepción o no esperados, para asegurarnos de que el programa no producirá errores en la ejecución cuando se presenten estos casos.

Actividad de aprendizaje 1

Establece un planteamiento para calcular la nómina de un trabajador, son necesarios su nombre, el número de horas trabajadas, el pago por hora laborada y que al final muestre el nombre del trabajador y el total por pagar. Identifica los datos de entrada, el proceso y la salida.



Actividad



1.2. Diseño de algoritmos

1.2.1. Concepto de algoritmo

Para Cairó (2006), un algoritmo es un plan preciso para ejecutar una secuencia de acciones ordenadas para alcanzar un propósito propuesto en un tiempo finito; es decir, es el conjunto de operaciones y procedimientos que debe seguirse para resolver un problema.

1.2.2. Características de los algoritmos

Un algoritmo debe poseer las siguientes cinco propiedades:

- 1) Debe ser preciso.
- 2) Debe estar bien definido.
- 3) Debe ser finito.
- 4) Debe ser flexible.
- 5) Debe ser diseñado utilizando un estilo entendible para cualquiera.

Los algoritmos pueden ser representados de muchas formas. Para cualquier algoritmo puede haber muchas representaciones de este, algunas mucho mejores que otras. No existe alguna representación que sea la mejor para todos los algoritmos. A continuación se describen algunas posibles representaciones:

Verbal: el algoritmo es expresado en palabras.

Algebraica: el algoritmo es expresado matemáticamente con símbolos y fórmulas.

Tabular: el algoritmo es expresado por una o más tablas.

Diagrama de flujo: el algoritmo es representado como diagrama con cajas de acción, ligadas por líneas que muestran el orden en que se ejecutan o la secuencia de acciones. Este se denomina el flujo de control.

Pseudocódigo: el algoritmo se representa como un conjunto de instrucciones escritas, usando una mezcla de lenguaje natural y notaciones matemáticas. La forma de las instrucciones es similar a las que tienen los lenguajes de programación.

1.2.3. Tipos de datos

Existen diferentes tipos de datos, como simples y estructurados. En este libro se mencionan únicamente los simples.

1.2.3.1. Numéricos

Estos pueden ser de dos tipos:

Entero. El tipo entero es una representación del conjunto de los números enteros. La razón de esta limitación está en la necesidad de utilizar un espacio finito, y fijo, para cada dato.

Real. Son datos numéricos que tienen parte decimal.

1.2.3.2. Literal

Carácter. Son los tipos de datos que almacenan una letra, número, símbolo o signo.

Cadena. Son un conjunto de caracteres que permiten representar valores identificables de forma descriptiva, esto incluye nombres, direcciones, etc. No se puede hacer operaciones con ellos (Joyanes, 1987).

1.2.3.3. Constante

Es un valor de tipo permanente, que no puede modificarse, al menos no dentro del contexto o situación para el cual está previsto. Ejemplo: π , gravedad, días de la semana, meses, etcétera.

1.2.4. Operadores aritméticos y su prioridad; relacionales y lógicos

Permiten la realización de operaciones matemáticas con los valores (variables y constantes).

Los operadores aritméticos pueden ser utilizados con tipos de datos enteros o reales. Si ambos son enteros, el resultado es entero; si alguno de ellos es real, el resultado es real. Operando (operador) Operando.

Operadores aritméticos

- + suma,
- resta,
- * multiplicación,
- mod módulo (residuo de la división entera)
- / división,
- ^ exponenciación

Prioridad de los operadores aritméticos

Todas las expresiones entre paréntesis se evalúan primero. Las expresiones con paréntesis anidados se evalúan de adentro hacia afuera, el paréntesis más interno se evalúa primero.

Dentro de una misma expresión los operadores se evalúan en el siguiente orden:

1. ^ Exponenciación
2. *, /
3. +, -



Ejemplos:

$$7 / 2 = 3.5$$

$$12 \bmod 7 = 5$$

$$4 + 2 * 5 = 14$$



Actividad de aprendizaje 2

Resuelve las siguientes operaciones aplicando la prioridad de los operadores aritméticos.

$5 + 8 * 3 + 120 =$	$(8 * 3) + 14 - 2 =$	$(2 * 6 / 2) + 15 / 3 =$
$(12 / 3 * 2) - 5 + 20 =$	$5 + 6 + (20 * 3 / 2) - 11 =$	$15 \bmod 2 =$

Operadores relacionales

Se utilizan para establecer una relación entre dos valores. Luego compara estos valores entre sí y esta comparación produce un resultado de certeza o falsedad (verdadero o falso).

Los operadores relacionales comparan valores del mismo tipo (numéricos o cadenas).

Éstos tienen el mismo nivel de prioridad en su evaluación.

Los operadores relacionales tienen menor prioridad que los aritméticos.

Operadores relacionales

- > Mayor que
- < Menor que
- >= Mayor o igual que
- <= Menor o igual que
- < > Diferente
- = Igual

Ejemplos:

Si $a = 10$, $b = 20$, $c = 30$

$a + b > c$ Falso

$a - b < c$ Verdadero

$a - b = c$ Falso

$a * b < > c$ Verdadero

Actividad de aprendizaje 3



Actividad

Resuelve el siguiente ejercicio utilizando los operadores relacionales.

Si: $a = 7$, $b = 4$, $c = 9$ entonces:

Sentencia	Resultado
$a * b > c$	
$c - a < b$	
$a - b = c$	
$a + b < > c$	

Operadores lógicos

Estos operadores se utilizan para establecer relaciones entre valores lógicos, pueden ser resultado de una operación relacional. Estos son:

AND	Y
OR	O
NOT	NO

Ejemplo:

Si $a = 10$, $b = 20$, $c = 30$

$(a < b) \text{ and } (b < c)$

V V
V

Operador **NOT** Operador **NOT**

Operando Resultado

V	F
F	V

Operador **AND** Operador **AND**

Operando1 Operador Operando2 Resultado

V	AND	V	V
V		F	F
F		V	F
F		F	F



Operador	OR	Operador	OR
Operando1	OR	Operando2	Resultado
V	OR	V	V
V		F	V
F		V	V
F		F	F



Actividad de aprendizaje 4

Resuelve el siguiente ejercicio utilizando los operadores lógicos.

Sea: Si $a = 5$, $b = 10$, $c = 20$

$(a > b) \text{ or } (b < c)$

1.2.5. Instrucciones y sentencias

Un programa es una secuencia de instrucciones mediante las cuales se ejecutan diferentes acciones de acuerdo con los datos que se estén procesando. El programa debe incluir instrucciones para las acciones que deban ejecutarse sobre cada uno de los tipos de datos admitidos. Un programa se compone de estructuras de datos, operaciones primitivas elementales y estructuras de control, como se muestra a continuación:

Programa = estructuras de datos
 + operaciones primitivas elementales
 + estructuras de control

Estructuras algorítmicas de control

Para Pólya (1965) las estructuras de operación de programas son un grupo de formas de trabajo, que permiten, mediante la manipulación de variables, realizar ciertos procesos específicos que nos llevan a la solución de problemas. Estas estructuras se clasifican de acuerdo con su complejidad:

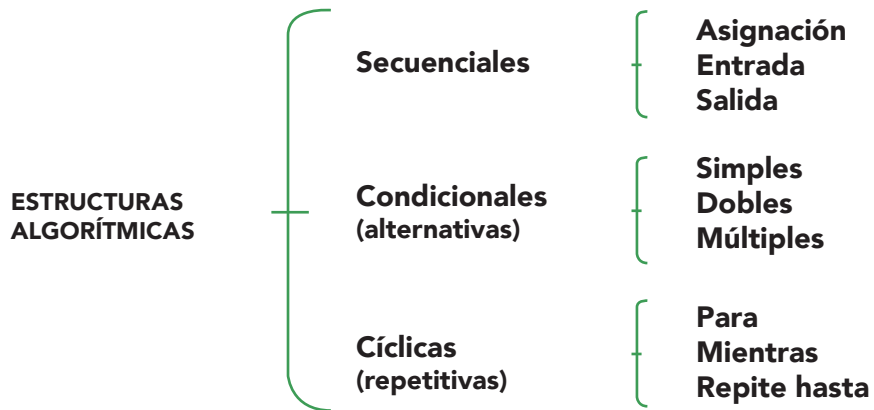


Figura 1.1. Estructuras algorítmicas

Estructuras secuenciales

Son estructuras con una entrada y una salida, en la que figuran una serie de acciones cuya ejecución es lineal (sigue una instrucción tras otra), y en el orden en que aparecen.

Ejemplo: programa que solicite un nombre, un número de cuenta y tres calificaciones, que calcule el promedio y que al final muestre el nombre, número de cuenta y el promedio.

Inicio

```

Cadena nombre(20)
Entero vcta
Real cal1,cal2,cal3,prom
Escribir ("Escribe tu nombre")
Leer nombre
Escribir ("Escribe tu numero de cuenta")
Leer vcta
Escribir ("Teclea tus tres calificaciones")
Leer cal1,cal2,cal3
Prom=(cal1+cal2+cal3)/3
Escribir ("tu nombres es_____ tu no. De cuenta_____y tu promedio es:_____",
nombre,vcta,prom)
  
```

Fin

Actividad de aprendizaje 5

**Actividad**

Estructura el ejercicio anterior para que pueda funcionar en el programa PseInt (intérprete de pseudocódigo).

Estructuras condicionales (alternativas)

Es una estructura con una entrada y una salida, en la cual se ejecuta una acción de entre varias según una condición, o se realiza una acción según el cumplimiento, o no, de una determinada condición.

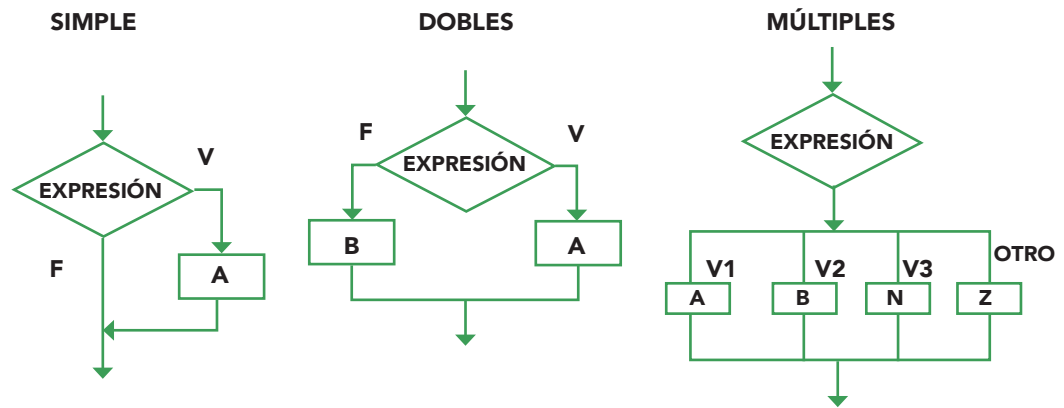


Figura 1.2. Estructuras condicionales

Pseudocódigo

Si (condición) entonces
A
Fin si

Pseudocódigo

Si (condición) entonces
A
Otro caso
B
Fin si

Pseudocódigo

Caso (expresión)
Expresión = V1
A
Expresión = V2
B
Expresión = VN
N
Otro caso
Z
Fin caso

Figura 1.3. Condicionales

Simple

Pseudocódigo

Inicio

Cadena nombre(20)
Entero vcta
Real cal2,cal2,cal3,prom
Escribir ("Escribe tu nombre")
Leer nombre
Escribir ("Escribe tu numero de cuenta")
Leer vcta
Escribir ("Teclea tus tres calificaciones")
Leer cal1,cal2,cal3
Prom=(cal1+cal2+cal3)/3

Si (prom >= 6) entonces

Escribir ("Aprobado")

Fin si

Fin

Doble

Pseudocódigo

Inicio

Cadena nombre(20)

Entero vcta

Real cal2,cal2,cal3,prom

Escribir ("Escribe tu nombre")

Leer nombre

Escribir ("Escribe tu numero de cuenta")

Leer vcta

Escribir ("Teclea tus tres calificaciones")

Leer cal1,cal2,cal3

Prom=(cal1+cal2+cal3)/3

Si (prom >= 6) entonces**Escribir ("Aprobado")****Otro caso****Escribir ("Reprobado")****Fin si****Fin**

Actividad de aprendizaje 6

**Actividad**

Estructura los ejercicios anteriores para que puedan funcionar en el programa PseInt (intérprete de pseudocódigo).

Múltiple

Ejemplo: programa que muestre un menú con el que el usuario pueda elegir una operación (suma, resta, multiplicación o división) que desea ejecutar con base en dos números pedidos anteriormente, desplegar el resultado de esa operación elegida.

Ejemplo:

Pseudocódigo

Inicio

Entero opc,

Real num1,num2, res

Escribir ("Escribe tu primer número")

Leer num1

Escribir ("Escribe tu segundo número")

Leer num2

Escribir ("MENU DE OPCIONES")

Escribir ("1.- SUMA")

Escribir ("2.- RESTA")

Escribir ("3.-MULTIPLICACION")

Escribir ("DIVISIÓN")

Escribir ("Teclea tu opción")

Leer opc

Caso (opc)

Opc=1



```

res=num1 + num2
Opc=2
res=num1 - num2
Opc=3
res=num1 * num2
Opc=4
Si (num2 < > 0) entonces
res=num 1 / num2
otro caso
Escribir ("Error")
Fin si
Fin Caso
Escribir("El resultado de tu operación es:", res)
Fin
    
```



Actividad de aprendizaje 7

Estructura el ejercicio anterior para que pueda funcionar en el programa PseInt (intérprete de pseudocódigo).

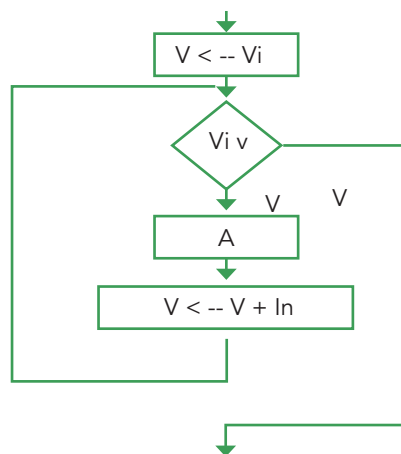
Estructuras repetitivas

Son estructuras con una entrada y una salida en la cual se repite una acción, un número determinado o indeterminado de veces, dependiendo del cumplimiento de una condición. Estas pueden ser:

Para (For)
Mientras (while)
Repita (Do while)

Estructura para (for)

Se repite una acción un número fijo de veces (N). El ciclo se inicia con un valor inicial V_i hasta llegar a un valor final, el incremento se hace en el mismo ciclo dentro del diagrama de flujo y para el pseudocódigo este va implícito.



Pseudocódigo

```

Para V <-- Vi hasta Vf hacer
    A
Fin para
    
```

Figura 1.4. Estructura for

Ejemplo: hacer un diagrama de flujo y pseudocódigo de un programa que pida 10 números enteros y realice e imprima la suma de éstos.

Entero i,suma=0,num

Inicio

Para (i=1 hasta i<=10)

Escribir ("Dame un número")

Leer num

suma=num+suma

i=i+1

Fin para

Escribir ("La suma es: ",suma)

Fin

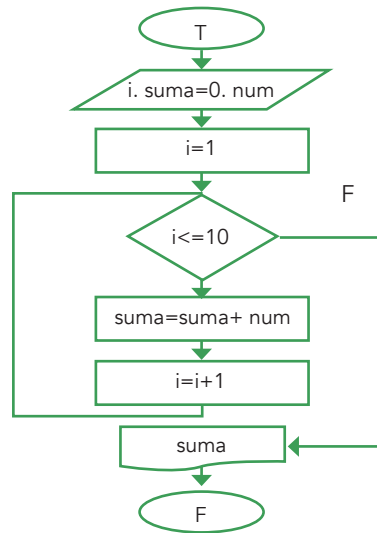


Figura 1.5. Ejemplo de for

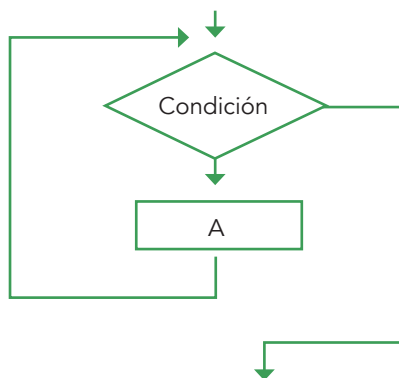
Actividad de aprendizaje 8



Estructura el ejercicio anterior para que pueda funcionar en el programa Pselnt (intérprete de pseudocódigo).

Estructura mientras (while)

Repita una acción mientras se cumpla la condición. El número de repeticiones oscila entre 0 e infinito, dependiendo de la evaluación de la condición.



Pseudocódigo

Mientras (condición)

A

Fin Mientras

Figura 1.6. Mientras



Ejemplo. Hacer un diagrama de flujo y pseudocódigo de un programa que pida 10 números enteros y realice e imprima la suma de estos.

Entero $i=1$, $\text{suma}=0$, num

Inicio

Mientras ($i < 10$) hacer

Escribir ("Dame un número")

Leer num

$\text{suma} = \text{num} + \text{suma}$

$i = i + 1$

Fin Mientras

Escribir (" La suma es: ", suma)

Fin



Actividad de aprendizaje 9

Estructura el ejercicio anterior para que pueda funcionar en el programa PseInt (intérprete de pseudocódigo).

Estructura repite (do while)

Repite una acción hasta que se cumpla la condición, la cual se evalúa después de cada ejecución. El número de repeticiones oscila entre 1 e infinito, dependiendo de la evaluación de la condición.

Entero $i = 1$, $\text{suma}=0$, num

Inicio

Mientras ($i < 10$) hacer

Escribir ("Dame un número")

Leer num

$\text{suma} = \text{num} + \text{suma}$

$i = i + 1$

Fin Mientras

Escribir ("La suma es: ", suma)

Fin

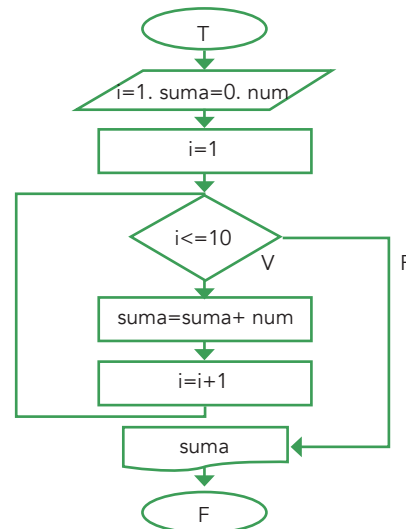


Figura 1.7. Ejemplo mientras



Actividad de aprendizaje 10

Estructura el ejercicio anterior para que pueda funcionar en el programa PseInt (intérprete de pseudocódigo).

1.3. Herramientas para el diseño de algoritmos

1.3.1. Pseudocódigo

Mezcla de lenguaje de programación y español que se emplea, dentro de la programación estructurada, para hacer el diseño de un programa.

Ventajas de utilizar un pseudocódigo en un diagrama de flujo

- Ocupa menos espacio en una hoja de papel.
- Permite representar de manera fácil operaciones repetitivas complejas.
- Es muy fácil pasar de pseudocódigo a un programa en algún lenguaje de programación.
- Si se siguen las reglas se pueden observar claramente los niveles que tiene cada operación.

1.3.2. Transcripción

Es necesario que llevemos el programa codificado a un medio que sea aceptado como entrada por la computadora: lo grabamos en un disco flexible o lo grabamos en un disco duro. Este programa es el que se conoce como **programa fuente**.

En programación, cuando se desarrolla un programa, en la fase de codificación se llevará a cabo la transcripción, que consiste en que el compilador traducirá el código fuente a código máquina, también llamado **código objeto**, siempre y cuando, el propio compilador no detecte ningún error en dicho código fuente.

1.3.3. Pruebas de computador

Cuando tenemos el programa ejecutable ordenamos a la computadora que lo ejecute, para lo cual suministramos datos de prueba, como lo hicimos en la prueba de escritorio. Analizamos los resultados obtenidos, luego puede ocurrir cualquiera de estas situaciones:

- a) La lógica del programa está bien, pero hay errores sencillos, los cuales se corrigen modificando algunas instrucciones o incluyendo unas nuevas.
- b) Hay errores ocasionados por fallas en la lógica.
- c) Hay errores muy graves y lo más aconsejable es que regresemos para analizar de nuevo el problema.
- d) No hay errores y los resultados son los esperados. En este caso, podemos guardar el programa permanentemente en una librería o biblioteca del computador, para sacarlo de allí cuando necesitemos ejecutarlo de nuevo.



1.3.4. Documentación externa

Cuando el programa ya está listo para ejecutarse es conveniente que hagamos su documentación externa siguiendo las normas de la instalación o las recomendaciones indicadas. Una buena documentación incluye siempre:

- Enunciado del problema
- Diagrama de pasada
- Narrativo con la descripción de la solución
- Relación de las variables o campos utilizados en el programa, cada uno con su respectiva función
- Diagrama del programa
- Listado de la última compilación
- Resultados de la ejecución del programa

Resumen

En el primer módulo conocimos los conceptos de problemas, así como los pasos para solucionarlos con el objetivo de que los alumnos analicen las problemáticas que se encuentran en su entorno y puedan hacer un análisis más detallado. Por otra parte, se conoce una metodología que ayuda a elegir la opción más apropiada.

También se trató el tema de algoritmos y sus características, que ayuda a representar los problemas utilizando gráficos para tener un panorama más amplio del conflicto. Conocimos los tipos de datos y la utilización de la prioridad de los operadores aritméticos, así como las instrucciones y sentencias.

Al final del módulo se analizan las estructuras de los pseudocódigos y se hacen pruebas de escritorio para saber si nuestro planteamiento es correcto, también se hace la documentación que sustenta su trabajo.

Autoevaluación

¿Considero que lo hice bien?

Este es un proceso de reflexión sobre lo que obtuviste después de diseñar y hacer la demostración práctica. Responde la tabla siguiente colocando una X en el cuadro correspondiente a cada ítem (5 representa "estoy totalmente de acuerdo", y 1 "estoy totalmente en desacuerdo"):

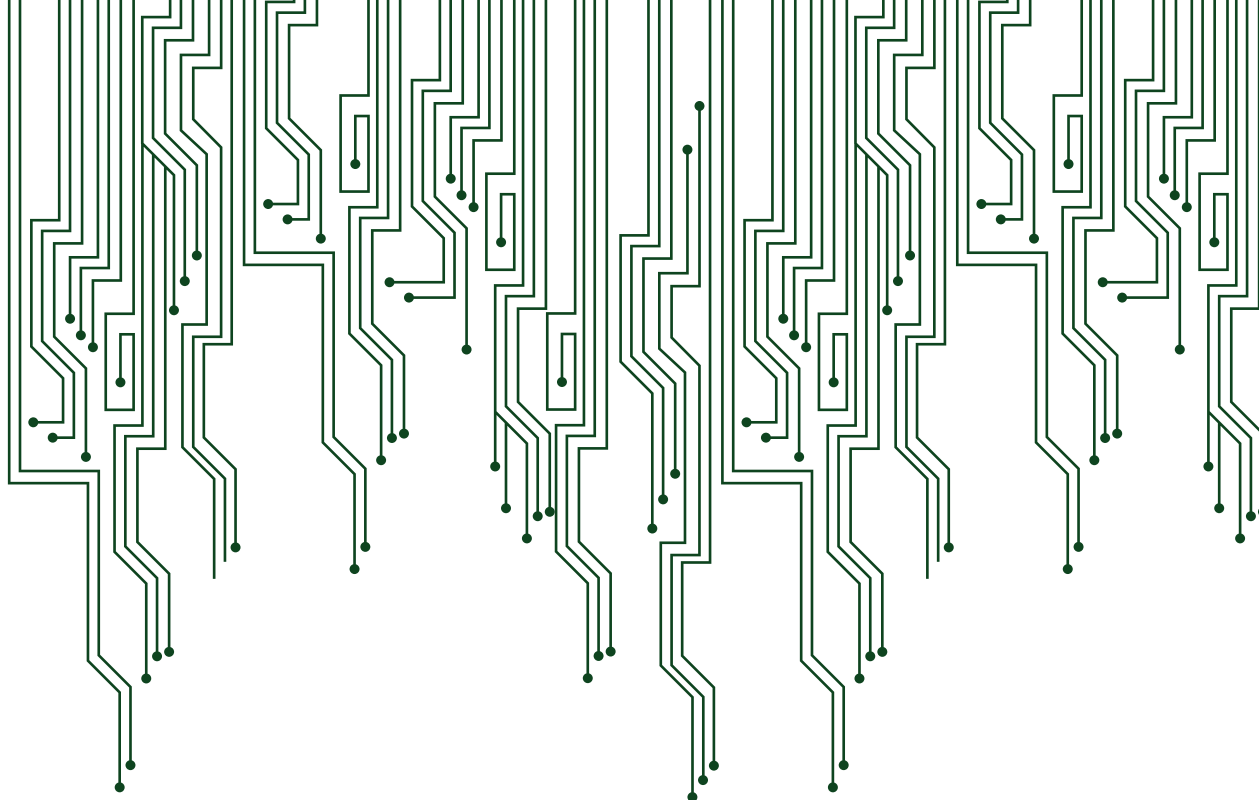
	5	4	3	2	1
1. Asumí una actitud propositiva, entusiasta, responsable.					
2. El aprendizaje y las experiencias que me dejó la participación en la actitud me son útiles.					
3. Pude explicar con claridad los conceptos y definiciones que revisamos y aplicamos a lo largo de cada actividad.					
4. Puedo aplicar lo aprendido en mi vida diaria.					
Mi calificación general sería de: (anota un valor numérico en escala del 1 a 10).					

Coevaluación

Puedes compartir la autoevaluación que hiciste con tus compañeros con la intención de intercambiar ideas o experiencias de aprendizaje. En caso de que hayas tenido alguna dificultad sería una buena aportación compartirla.

La calificación de los compañeros con quienes intercambié mi autoevaluación fue (anotar los valores numéricos que tenga asignados).

Ítem	Compañero 1	Compañero 2	Compañero 3
1			
2			
3			
4			



Módulo 2

Programas y aplicaciones



Propósito del módulo

Implementa aplicaciones informáticas, para solucionar problemas mediante el uso de la computadora en diversos contextos utilizando un lenguaje de programación.

Imagen: pxhere.com





Competencias disciplinarias

Comunicación

Básicas

9. Analiza y compara el origen, desarrollo y diversidad de los sistemas y medios de comunicación.
10. Identifica e interpreta la idea general y el posible desarrollo de un mensaje oral y escrito en una segunda lengua, recurriendo a conocimientos previos, elementos no verbales y contexto cultural.
11. Se comunica en una lengua extranjera mediante el discurso lógico, oral o escrito, congruente con la situación comunicativa.

Extendidas

8. Valora la influencia de los sistemas y medios de comunicación en su cultura, su familia y su contexto socioculturales diferentes.
10. Analiza los beneficios e inconvenientes del uso de las tecnologías de la información y la comunicación para optimización de las actividades cotidianas.

Competencias genéricas

4. Escucha, interpreta y emite mensajes pertinentes en distintos contextos mediante la utilización de medios, códigos y herramientas apropiados.
 - 4.2. Aplica distintas estrategias comunicativas según quienes sean sus interlocutores, el contexto en el que se encuentra y los objetivos que persigue.
5. Desarrolla innovaciones y propone soluciones a partir de métodos establecidos.
 - 5.1. Sigue instrucciones y procedimientos de manera reflexiva, comprendiendo cómo cada uno de sus pasos contribuye al alcance de un objetivo.
 - 5.2. Ordena información de acuerdo con categorías, jerarquías y relaciones.
 - 5.6. Utiliza las tecnologías de la información y la comunicación para procesar e interpretar información.

**Actividad**

Actividad diagnóstica

Instrucciones: subraya la respuesta correcta.

1. Se le conoce como la forma de expresar la solución a un problema de manera que sea comprensible para una computadora.

- a) Programa b) Pseudocódigo c) Problema

2. Es un ejemplo de lenguaje de programación

- a) C++ b) Word c) Access

3. Es la única función que no puede faltar en un programa de C.

- a) Suma b) Main() c) Escribir

4. Es un ejemplo de librería de C.

- a) `#include<stdio.h>`
b) `import java.util.Scanner`
c) `#### main`

5. Son nombres usados para referenciar las variables, las funciones, las etiquetas y otros objetos definidos por el programador.

- a) Algoritmo b) Problema c) Identificadores

6. Estas son referenciadas solo por sentencias que están dentro del bloque en que han sido declaradas.

- a) Algoritmo b) Variables globales c) Variables locales

7. Estas se reconocen a lo largo de todo el programa y se usan en cualquier parte del código

- a) Algoritmo b) Variables globales c) Variables locales

8. Sirve para mostrar un texto en pantalla en C.

- a) `Printf` b) `Scanf` c) `System.out`

9. Sirve para leer datos en C.

- a) `Printf` b) `Scanf` c) `System.out`

10. Es una estructura de control repetitiva en C.

- a) `For` b) `if` c) `Switch`



Introducción

C es un lenguaje estructurado, cuyo componente principal es la función. En C las funciones son los bloques consecutivos en los que se desarrolla toda la actividad de los programas, permiten definir las tareas de un programa y codificarlas por separado, permitiendo así que los programas sean modulares. El lenguaje de programación C se caracteriza por ser de uso general, con una sintaxis sumamente compacta y de alta portabilidad, de fácil aprendizaje.

Es un lenguaje que mejorará mucho la forma en que comprendemos los demás lenguajes, ya que para hacer cosas que vemos casi automáticas en otros lenguajes de alto nivel, aquí se deben realizar con la lógica con que el lenguaje de máquina funciona, permitiendo de esta manera saber exactamente qué pasa en cada parte de una aplicación y mejorar nuestra lógica de programación.

Conocer el uso de punteros es esencial en C, así apreciamos su lógica de funcionamiento, entendiendo cómo se reserva la memoria, cómo se libera, su utilización, cómo podemos almacenar la información de distintas maneras, convertir tipos de datos, pasar variables por valor y referencia a funciones, etcétera.

2.1. Lenguaje de programación y proyecto

Antes de empezar a programar en un lenguaje determinado hay que conocer una serie de conceptos básicos comunes a todos estos lenguajes.

Para empezar, debemos definir qué es un programa. Un programa puede entenderse como la forma de expresar la solución a un problema de manera que sea comprensible para el ordenador. En otras palabras, un programa es un conjunto ordenado de instrucciones que se dan a la computadora indicando el conjunto de operaciones o tareas que se desean llevar a cabo.

Una instrucción es un conjunto de símbolos que representa una orden de operación o tratamiento para el ordenador.

A la hora de crear un programa, hay que tener en cuenta los siguientes pasos:

1. Hacer un análisis del problema definiendo los requisitos a los que se deben dar solución.
2. Diseñar y verificar el algoritmo que se seguirá para solucionar el problema.
3. Se codifica el algoritmo diseñado con anterioridad en un lenguaje de programación.
4. Se traduce el código fuente escrito a código máquina empleando compiladores, de esta manera se genera el programa ejecutable por el ordenador.
5. Se comprueba que el programa generado funciona correctamente. En caso de encontrar algún error, se depura el código.
6. Se hace la documentación del programa realizado.

¿Qué es el lenguaje de programación C?

El lenguaje de programación C es un lenguaje de alto nivel que se caracteriza por los siguientes puntos:

- En C las mayúsculas y minúsculas son diferentes Vnom _ vnom _ VNOM
- Al final de cada sentencia debe ir punto y coma (;)
- Toda sentencia de caracteres encerrados entre /* */ es un comentario
- El uso de un bloque de código (grupo de sentencias), en C se crea colocando una secuencia de sentencias entre { }; es decir, { indica el inicio del bloque y } el fin del bloque.
- La única función que debe estar absolutamente presente es la denominada main(), función que es llamada al comienzo de la ejecución del programa.

Elementos básicos de un lenguaje

Estructura general de un programa codificado en C.

```
Archivos de Cabecera <*.h>
Definición de constantes
Declaraciones globales
Declaración de funciones;
Main(argumentos)
{
Variables locales
Flujo de sentencias
}
Ejemplos de archivos de cabecera
#include<stdio.h>
#include<conio.h>
#include<math.h>
```

2.2. Editor del lenguaje de programación

Lo primero que debes hacer es descargar e instalar el compilador Dev C++, de la siguiente liga: <http://www.bloodshed.net/devcpp.html> o también puedes descargarlo de la dirección de Bloodshed software, que son los autores del entorno.

El compilador Dev C++ es totalmente gratuito, aunque los autores agradecerían que te registraras y pagaras una pequeña cantidad que les ayudará a seguir desarrollando nuevos productos (licencia de software de dominio público GPL).

Alguna de las principales características del entorno Dev C++ son:

- Depurador integrado (GDB o Insight)
- Editor de programas fuente sensible a la sintaxis C-C++ configurable.
- Editor multiventana con múltiples opciones de edición.
- Se puede trabajar con ficheros independientes o con proyectos multi-ficheros.
- Generador de paquetes de instalación de programas para entorno Windows.
- Puede generar programas DOS (modo consola), aplicaciones Windows y DLLs.
- Edición de ficheros de recursos Windows con generador de menús integrado.



- Ventanas independientes para el gestor de proyectos, editor y resultados de compilación.
- Resultados de compilación, enlace (linker) y generación de recursos.
- Genera automáticamente esqueletos de programas en C y C++.
- Generación automática de "Makefile"
- Dos juegos de íconos para los menús y las barras de herramientas.
- Permite integrar herramientas externas mediante el "tool manager"
- Plantillas para la creación de nuevos tipos de proyectos.
- Etcétera...

Dev C++ ha sido realizado por Colin Laplace (desarrollo principal del IDE) y Hongli Lai (actualizaciones del IDE, pantalla de arranque e íconos). El compilador Mingw ha sido desarrollado por Mumit Khan, Jan Jaap van der Heidjen, Colin Hendrix y programadores del grupo GNU por todo el mundo.

2.2.1. Entorno

Mi primer programa en C "Bienvenido a la programación en C" usando DevC++

Nuestro primer programa en el lenguaje de programación C consistirá en escribir en pantalla la frase **"Bienvenido a la programación en C"**.

Para ello, utilizaremos el entorno de programación Dev-C (de distribución gratuita).

Lo primero que hay que hacer es crear un nuevo Proyecto, para ello vamos al menú Archivo/Nuevo/Archivo fuente como se indica en la siguiente figura:

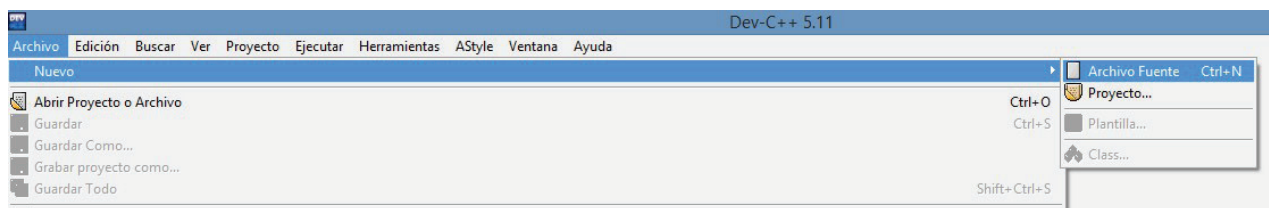


Figura 2.1. Nuevo proyecto

Por defecto, el entorno de programación Dev-C++ crea el archivo main con los parámetros necesarios. Por así decirlo, crea una ayuda para el programador, pero lo que interesa es aprender desde cero. Por eso borramos todo lo que aparece en archivo main.c y nos quedará igual que la pantalla siguiente:

Ahora escribiremos el código de nuestro programa:

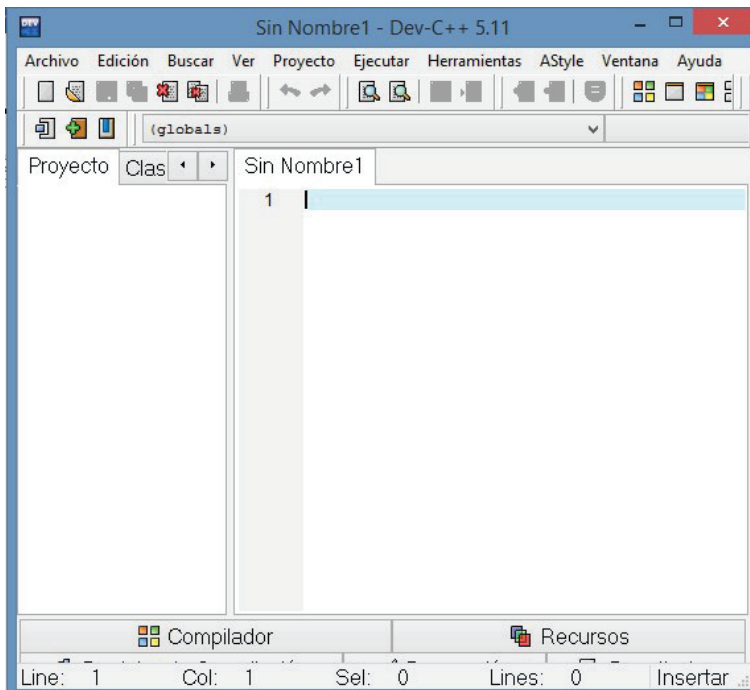


Figura 2.2. Inicio.

Como ya hemos visto, un programa escrito en C debe seguir una estructura. Lo primero que debe aparecer en nuestro código es la definición de las librerías de funciones. En nuestro caso, solo necesitamos la declaración de la librería `stdio.h`. Esta librería es la biblioteca estándar del lenguaje C. En ella se definen las macros, las constantes, las declaraciones de funciones y la definición de tipos usados por varias operaciones estándar de entrada y salida.

Para mostrar por pantalla el texto que deseamos, necesitamos la función `printf()` definida en la librería `stdio.h`. Entre los paréntesis escribimos el mensaje que queremos que se muestre entre comillas dobles (" "). Esto debe ir dentro del programa principal, `main`, descrito anteriormente.

El código de nuestro programa debe tener el siguiente aspecto:

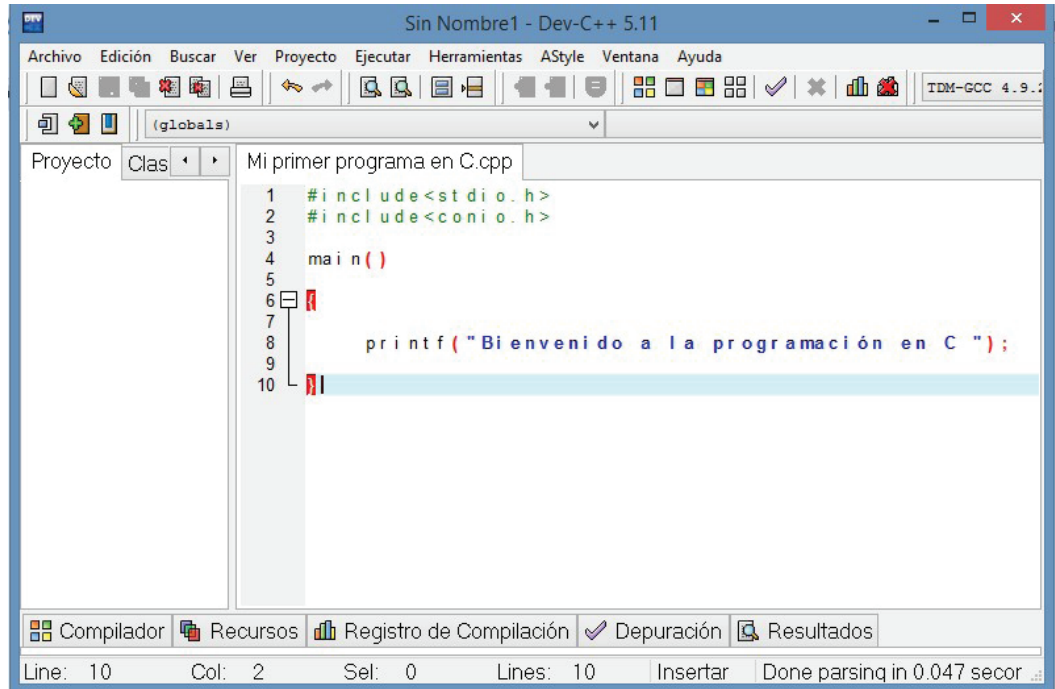


Figura 2.3. Código 1.

Una vez que está escrito nuestro código, hay que compilarlo. Compilar lo que hacemos es traducir nuestro código a un lenguaje entendible por la computadora.

Para ello hacemos clic en el siguiente botón:

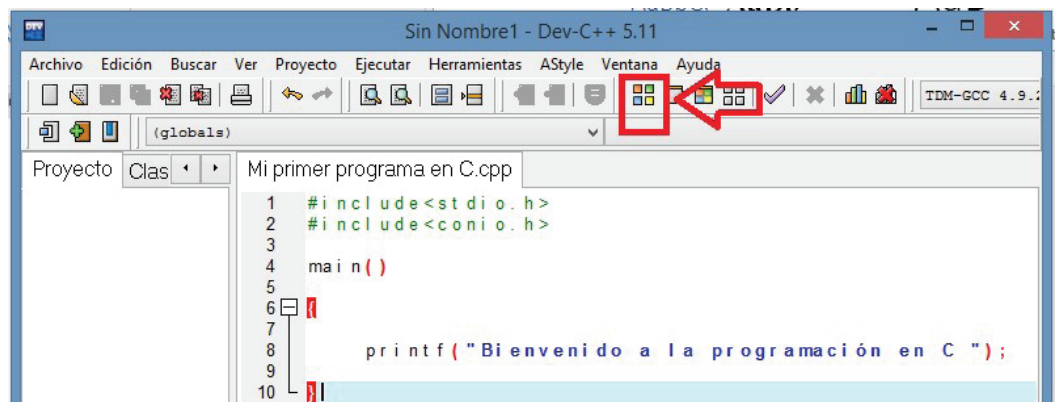


imagen 4 compilar

Si en algún momento aparece un error es porque no hemos escrito nuestro código de la manera correcta. En este caso, Dev-C nos indicará la línea donde está nuestro error. Un ejemplo es el siguiente:

Borraremos en la línea donde escribimos por pantalla el ";". La pantalla de error será la siguiente:

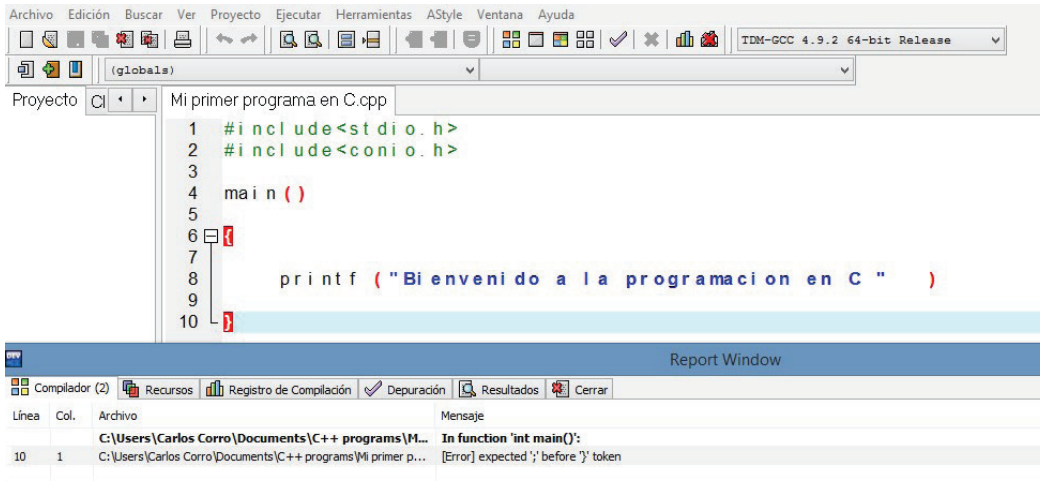


Figura 2.5. Errores

Nos indica que hay un error de sintaxis antes de ")" y está en la línea 10. Solucionamos el error y volvemos a compilar. Una vez compilado presionamos en el siguiente botón para ejecutar el programa.

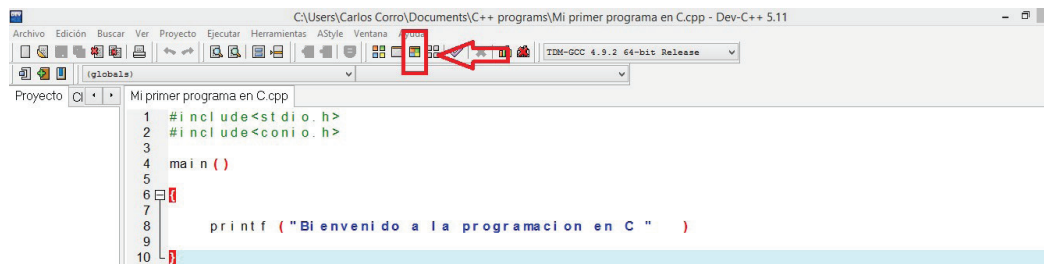


Figura 2.6. Ejecutar

Y tendremos como resultado lo siguiente:

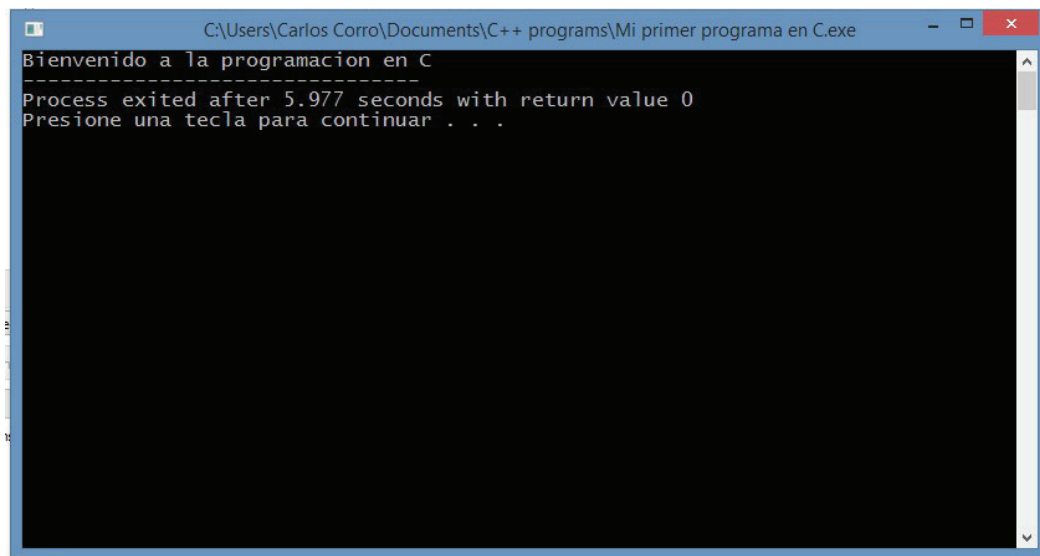


Figura 2.7. Programa ejecutado 1



Actividad de aprendizaje 1

Captura el código en Devc++, utilizando el ejemplo anterior; compílalo y ejecútalo hasta llegar a la ventana mostrada en la parte superior.

2.3. Codificación de instrucciones básicas. Sintaxis de sentencias: lectura y escritura de datos

Variables y tipos

Identificadores

Son nombres usados para referenciar las variables, las funciones, las etiquetas y otros objetos definidos por el programador, su longitud varía de uno a varios caracteres. Cabe mencionar que un identificador no debe ser igual que una palabra clave del lenguaje.

Tipos de datos

Tipo de codificación en "c" rango

TIPO	CODIFICACIÓN EN "C"	RANGO
Carácter	Char	0 a 255
Entero	Int	-32768 a 32768
Coma flotante	Float	Aprox. 6 dígitos de precisión
Coma flotante de doble posición	Double	Aprox. 12 dígitos de precisión
Sin valor	void	Sin valor

Figura 2.8. Tipos de datos.

Variables. Identificador, palabra o elemento que puede ser sustituido por diversos términos o valores; es decir, su valor puede cambiar durante la ejecución del programa. Como en todos los lenguajes deben ser declaradas antes de ser usadas, su sintaxis es la siguiente:

Tipo nombre de la variable; int edad;

Tipo lista de variables; separadas por comas int num1, num2, a,b;

Tipos de variables

a) Variables locales. Son declaradas dentro de las funciones. Estas son referenciadas solo por sentencias que están dentro del bloque en que han sido declaradas.

b) Variables globales. Estas se reconocen a lo largo de todo el programa y se usan en cualquier parte del código, se declaran preferentemente antes de la función principal main ().

Inicialización de variables

Sintaxis:

Tipo Nombre_variable=constante

Las variables globales son inicializadas al comienzo del programa, se supone 0 a menos de que se especifique lo contrario. Las locales cada vez que se entra al bloque en que están definidas. Ejemplo:

```
Int x = 100;
```

Tipo nombre de la variable

Sentencias de asignación

Nombre_variable= expresión;

Donde expresión puede ser tan simple como una constante o tan compleja como una combinación de variables, operadores y constantes.

Expresiones y sintaxis

Operadores

Finalmente, ya conociendo los conceptos anteriores podemos decir que las variables son manipuladas por los operadores para formar expresiones.

Entradas y salidas

Lógicos, relacionales y aritméticos

LÓGICOS		RELACIONALES		ARITMÉTICOS	
&&	Y	>	Mayor	+	Suma
	O	<	Menor	-	Resta
!	NO	>=	Mayor que	*	multiplicación
		<=	Menor que	/	División
		==	Igual	%	Módulo
		!=	Diferente	--	Decremento
				++	incremento

Figura 2.9. Operadores.

Códigos de barra invertida

Código	Acción
\b	Espacio atrás
\f	Salto de página
\n	Nueva línea
\r	Salto de carro
\t	Tabulación horizontal
\“	Dobles comillas
\‘	Simple comillas
\0	Nulo
\\	Barra invertida
\a	alerta

Figura 2.10. Barra invertida.



printf Escribe en pantalla

`printf("%código", argumento)`

scanf Lee desde el teclado

`scanf ("%código",&argumento)`

Código. Se refiere al tipo de variable que será desplegada en pantalla para printf y leída para scanf.

Argumento. Se refiere a las variables que serán leídas o desplegadas en pantalla.

Gets. Lee una cadena de caracteres, espera un enter que indica el término de la cadena. Su formato es `gets(vcadena)`;

Getch(). Lee un carácter sin escribirlo en pantalla. Formato `getc( )`;

Puts. Escribe en pantalla una cadena de caracteres. Formato `puts(vcadena)`;



Actividad de aprendizaje 2

Escribe el siguiente código para calcular el área del círculo.

```
[*] area del círculo.cpp
1  #include<stdio.h>
2  #include<conio.h>
3  main( )
4  {
5      float pi=3.1416, radio, area;
6      printf("Teclea el radio \n");
7      scanf("%f",&radio);
8      area=pi*(radio*radio);
9      printf("El área del círculo es %f \n", area);
10 }
```

Figura 2.11. Área del círculo.

2.3.1. Estructuras de control

Las estructuras de control determinan la secuencia en la que se ejecutarán las instrucciones de un programa.

Las estructuras de control se dividen en tres categorías en función del flujo de ejecución:

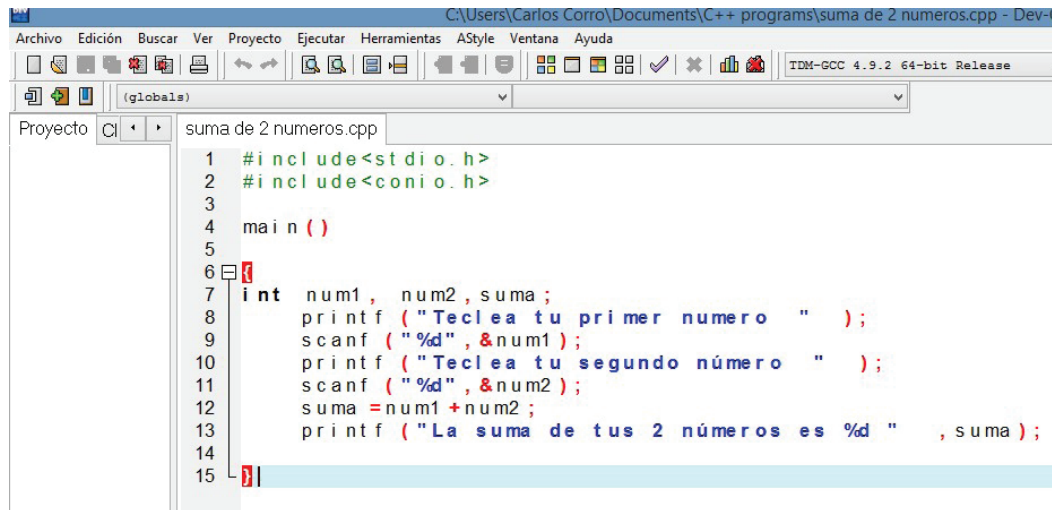
- Estructura secuencial.
- Estructura selectiva.
- Estructura iterativa o repetitiva.

2.3.1.1. Estructuras secuenciales, selectivas y repetitivas

Estructuras secuenciales

La estructura secuencial está formada por una secuencia de instrucciones que se ejecutan en orden, una a continuación de la otra.

Ejemplo: programa para calcular la suma de 2 números.



```

1 #include<stdio.h>
2 #include<conio.h>
3
4 main ()
5
6 {
7     int num1, num2, suma;
8     printf ("Teclea tu primer numero ");
9     scanf ("%d",&num1);
10    printf ("Teclea tu segundo número ");
11    scanf ("%d",&num2);
12    suma = num1 + num2;
13    printf ("La suma de tus 2 números es %d ", suma);
14
15 }
  
```

Figura 2.12. Suma.

Actividad de aprendizaje 3

Captura el código del programa que calcula la suma de 2 números enteros y muestra el resultado.



Actividad

Actividad de aprendizaje 4

Realiza los siguientes programas utilizando Devc++.



Actividad

Programas secuenciales

1. Realiza un programa con el cual calcules la nómina de un trabajador; pide el número de horas trabajadas y pide el costo por hora laborada. Imprime el total a pagar.
2. Realiza un programa que solicite el costo de dos artículos y sus precios, posteriormente, que calcule el total a pagar incluyendo el cálculo del IVA = 16%.
3. Realiza un programa que solicite el nombre de un estudiante y tres calificaciones, que al final imprima el nombre del estudiante y su promedio.
4. Elabora un programa tal que, dado el costo de un artículo vendido y la cantidad de dinero entregado por el cliente, calcule e imprima el cambio que debe entregársele al mismo.



Estructuras selectivas

Las estructuras condicionales controlan si una sentencia o bloque de sentencias se ejecutan, en función del cumplimiento, o no, de una condición o expresión lógica.

C++ tiene dos estructuras de control para la selección, **if** y **switch**.

Ejemplo: programa que solicite el nombre de una persona, su estatura, y que te indique si puedes o no subir a un juego.

if

```

1 #include<stdio.h>
2 #include<conio.h>
3 main()
4 {
5     float est;
6     char nombre[20];
7     printf("Teclea tu nombre ");
8     scanf("%s",&nombre);
9     printf("Tecea tu estatura \a");
10    scanf("%f",&est);
11    if(est >= 1.50)
12    {
13        printf("%s Puedes subir al juego\a", nombre);
14    }
15    else
16        printf("%s NO Puedes subir al juego\a", nombre);
17 }

```

Figura 2.13. Estatura.

Switch

Ejemplo: programa que solicita dos números y de acuerdo con un menú los suma, resta, multiplica o divide y muestra el resultado.

```

1 #include<stdlib.h>
2 #include<stdio.h>
3 #include<conio.h>
4 float val1, val2, res;
5 char opc;
6 main()
7 {
8     do{
9         printf("PROGRAMA PARA REALIZAR OPERACIONES\n\n");
10        printf("\nEscribe tu primer numero ");
11        scanf("%f",&val1);
12        printf("Escribe tu segundo numero ");
13        scanf("%f",&val2);
14        printf("Menu de Opciones\n\n");
15        printf("a) suma de dos numeros\n");
16        printf("b) resta de dos numeros\n");
17        printf("c) multiplicacion de dos numeros\n");
18        printf("d) division de dos numeros\n");
19        printf("Salir presiona s\n");
20        printf("\nElige una opcion ");
21        scanf("%s",&opc);
22    }
23    switch (opc)
24    {
25        case 'a':
26        {
27            printf("Elegiste la suma\n");
28            res=val1+val2;
29            printf("\nEl resultado de tu operacion es %.1f: ", res);
30            getch();
31            break;
32        }
33    }
34 }

```

Figura 2.14. Switch.

```

33     case 'b':
34     {
35         printf("Elegiste la resta\n");
36         res=val 1-val 2;
37         printf("\nEl resultado de tu operacion es %.1f: ",res);
38         getch();
39         break;
40     }
41     case 'c':
42     {
43         printf("Elegiste la multiplicacion\n");
44         res=val 1*val 2;
45         printf("\nEl resultado de tu operacion es %.1f: ",res);
46     }
47     getch();
48     break;
49     }
50     case 'd':
51     {
52         printf("Elegiste la division\n");
53         if(val 2 !=0)
54         {
55             res=val 1/val 2;
56             printf("\nEl resultado de tu operacion es %.1f: ",res);
57             getch();
58         }
59         else
60             printf("No existe division entre 0");
61         getch();
62         break;
63     }
64     default :
65         printf("\nOpcion sin operacion ");
66         getch();
67     } while(opc != 's');
68 }

```

Figura 2.15. Switch 2 1.

Actividad de aprendizaje 5



Actividad

Captura el código utilizando la estructura de control switch de un programa que solicita dos números y que de acuerdo con un menú los suma, resta, multiplica o divide, y muestra el resultado.

Actividad de aprendizaje 6



Actividad

Programas selectivos

1. Hacer un programa con el cual calcules el pago a un taxista, que pida el número de kilómetros por recorrer y el costo por kilómetro, si la cantidad de kilómetros por recorrer es mayor que 150 debe disminuir en 5% el costo. Imprime el total a pagar en ambos casos.
2. Elabora un programa para calcular el precio del boleto de ida y vuelta en ferrocarril, se debe conocer la distancia del viaje de ida y el tiempo de estancia. Se sabe además que si el número de días de estancia es superior a siete y la distancia total de ida y vuelta por recorrer es superior a 800 km, el billete tendrá una disminución de 30%. El precio por km es de \$0.17.
3. Hacer un programa con el cual calcules el precio de un terreno, del cual solicites las medidas de largo y ancho, así como el costo por metro cuadrado; si el número total de metros es mayor o igual 400 m² se obtendrá un descuento de 25%. Imprimir el costo en ambos casos.



Estructuras repetitivas

Son estructuras con una entrada y una salida en la cual se repite una acción, un número determinado o indeterminado de veces, dependiendo del cumplimiento de una condición.

Estas pueden ser:

- Para (For)
- Mientras (while)
- Repite (Do while)

Ejemplo: hacer un programa que solicite 10 números y que luego los sume.

For

for suma de 10 num.cpp

```
1 #include <stdio.h>
2 #include <conio.h>
3 int i, suma, num;
4 main()
5 {
6     for (i = 0; i <= 9; i++)
7     {
8         printf("Escribe el valor %d : ", i + 1);
9         scanf("%d", &num);
10        suma = suma + num;
11    }
12    printf("\nla suma de los 10 numeros es: %d ", suma);
13    getch();
14 }
```

Figura 2.16. Suma de 10 números.



Actividad de aprendizaje 7

Captura el código utilizando la estructura de control for de un programa que pida 10 números y los sume.

While

while suma de 10 num.cpp

```
1 #include <stdio.h>
2 #include <conio.h>
3 int i = 0, suma, num;
4 main()
5 {
6     while (i <= 9)
7     {
8         printf("Escribe el valor %d : ", i + 1);
9         scanf("%d", &num);
10        suma = suma + num;
11        i++;
12    }
13    printf("\nla suma de los 10 numeros es: %d ", suma);
14    getch();
15 }
```

Figura 2.17. While.

Actividad de aprendizaje 8



Actividad

Captura el código utilizando la estructura de control while, en un programa que pida 10 números y los sume.

Do while

do while suma de 10 num.cpp

```

1  #include <stdio.h>
2  #include <conio.h>
3  int i=0, suma, num;
4  main()
5  {
6      do
7      {
8          printf ("Escribe el valor %d : " , i+1);
9          scanf ("%d", &num);
10         suma = suma + num;
11         i++;
12     } while (i <= 9);
13     printf ("\n la suma de los 10 numeros es: %d " , suma);
14     getch();
15 }
```

Figura 2.18. Do while.

Actividad de aprendizaje 9



Actividad

Captura el código utilizando la estructura de control do while de un programa que pida 10 números y los sume.

Actividad de aprendizaje 10



Actividad

Programas repetitivos

Haz los siguientes programas con los tres tipos de ciclos: for, while, do while.

1. Haz un programa con el cual utilices un ciclo para pedir la estatura de 10 jugadores de básquetbol, con esos 10 datos saca el promedio de sus estaturas, y fuera del ciclo imprime el promedio con el siguiente texto "La media de las estaturas es: ".
2. Haz un programa utilizando un ciclo que permita imprimir la serie numérica de 7 en 7, debe comenzar en 7 y terminar en 56.
3. Haz un programa en el que uses un ciclo para pedir las ventas de 7 días; con esos 7 datos, saca el promedio de las ventas y fuera del ciclo imprime el promedio con el siguiente texto "El promedio de ventas semanal es: ".



2.4. Compilación

En programación, cuando se desarrolla un programa, en la fase de codificación se lleva a cabo la compilación, que consiste en que el compilador traduce el código fuente a código máquina, también llamado código objeto, siempre y cuando el propio compilador no detecte ningún error en dicho código fuente.

2.5. Depuración

El concepto depuración en informática se refiere al proceso de limpieza que se hace en un programa para identificar y corregir errores o problemas de programación. Del inglés *debugging*, que traducido de forma literal significa “eliminación de bichos (bugs)”, es el nombre con el que se conoce coloquialmente a los errores de programación.

Dado que con el paso del tiempo el software y los sistemas electrónicos son más complejos, se han ido desarrollando varias técnicas de depuración para detectar anomalías, corregir funcionalidades y optimizar el código fuente. Se dice que un programa se encuentra depurado cuando está libre de errores. En el proceso de depuración se hace un seguimiento en el funcionamiento de dicho programa, se observan los valores de las distintas variables, y se analizan también los resultados obtenidos en las operaciones.

Una vez depurado el programa, localizados los errores y subsanados, se procederá a depurar de nuevo hasta que el programa no arroje error alguno, tanto en tiempo de programación como en tiempo de ejecución. Al escribir programas es habitual cometer errores. Se ha establecido un promedio de aproximadamente un error por cada 10 líneas de programa, esto minimiza la posibilidad de que un programa funcione correctamente a la primera.

Para simplificar la fase de depuración es conveniente utilizar herramientas destinadas para este fin, la más habitual es el depurador o *debugger*. A través de esta herramienta se puede intervenir durante la ejecución de un programa para saber cómo se lleva a cabo la ejecución hasta ese momento.

2.6. Ejecución

En informática, ejecutar es la acción de comenzar la carga de un programa o de cualquier archivo ejecutable.

En otras palabras, la ejecución es el proceso mediante el cual una computadora lleva a cabo las instrucciones de un programa informático.

Se pueden ejecutar programas compilados (por ejemplo, en Windows, los .EXE) o programas interpretados (por ejemplo, los *scripts*).

Ejecutar un programa implica que estará en estado de ejecución y, por ende, en memoria, hasta que se finalice.

Resumen

Este módulo describe el concepto y uso de un lenguaje de programación identificando los componentes del entorno de trabajo de C y se hacen algunas pruebas respecto de la aplicación de las herramientas creando, abriendo y editando el código. También se estructura un proyecto mediante la elaboración de un programa.

Se analiza el procedimiento de compilación y ejecución de un programa.

Autoevaluación

¿Considero que lo hice bien?

Este es un proceso de reflexión sobre lo que obtuviste después de diseñar y hacer la demostración práctica. Responde la tabla siguiente colocando una X en el cuadro correspondiente a cada ítem (5 representa "estoy totalmente de acuerdo" y 1 "estoy totalmente en desacuerdo"):

	5	4	3	2	1
1. Asumí una actitud propositiva, entusiasta, responsable.					
2. El aprendizaje y las experiencias que me dejó la participación en la actitud me son útiles.					
3. Pude explicar con claridad los conceptos y definiciones que revisamos y aplicamos a lo largo de cada actividad.					
4. Puedo aplicar lo aprendido en mi vida diaria.					
Mi calificación general sería de: (anota un valor numérico en escala del 1 a 10).					

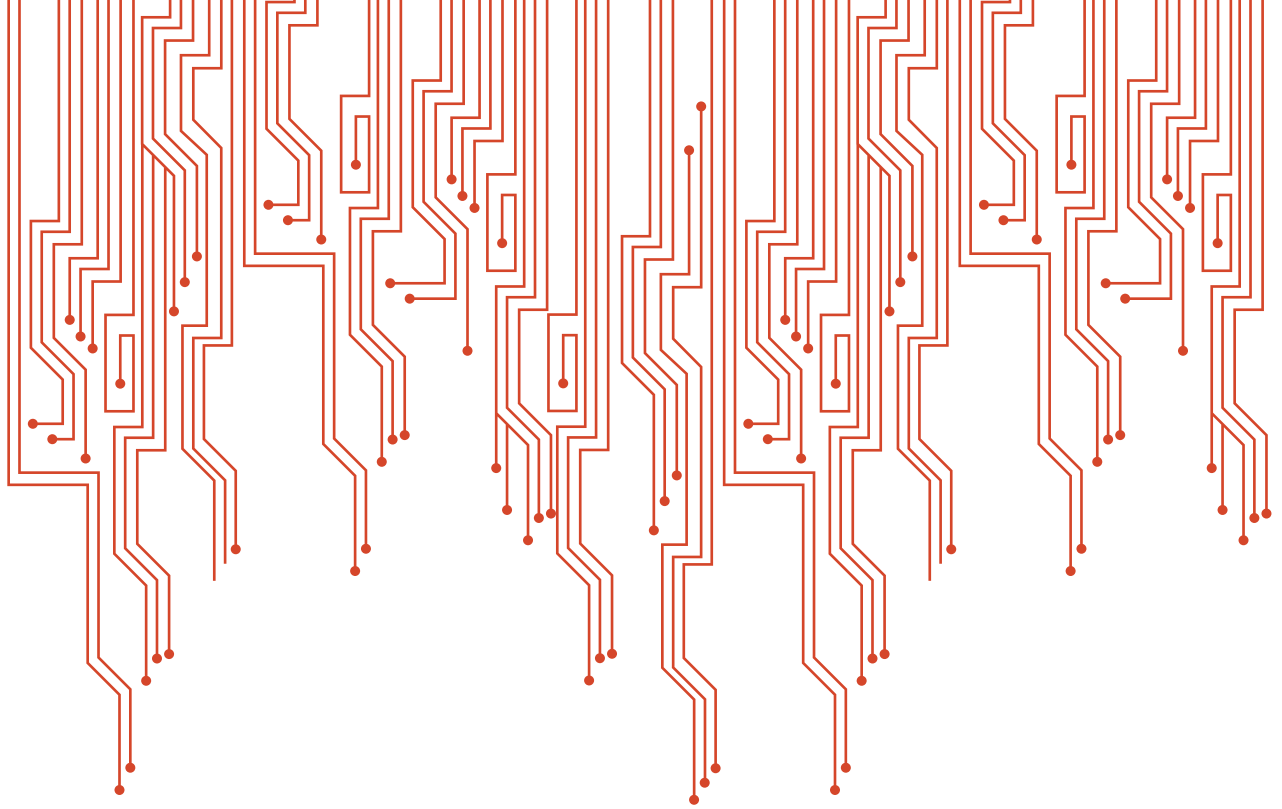


Coevaluación

Puedes compartir la autoevaluación que hiciste con tus compañeros con la intención de intercambiar ideas o experiencias de aprendizaje. En caso de que hayas tenido alguna dificultad sería una buena aportación compartirla.

La calificación de los compañeros con quienes intercambié mi autoevaluación fue (anotar los valores numéricos que tenga asignados):

Ítem	Compañero 1	Compañero 2	Compañero 3
1			
2			
3			
4			



Módulo 3

Proyecto web



Propósito del módulo

Planea y diseña un proyecto de sitio web con la utilización de las tecnologías de la información y la comunicación, para difundir las actividades culturales y productos académicos a la comunidad en general.

Imagen: pxhere.com



Competencia disciplinares

Comunicación

Básicas

4. Argumenta la solución obtenida de un problema, con métodos numéricos, gráficos, analíticos o variacionales, mediante el lenguaje verbal, matemático y el uso de las tecnologías de la información y la comunicación.
9. Analiza y compara el origen, desarrollo y diversidad de los sistemas y medios de comunicación.
10. Identifica e interpreta la idea general y posible desarrollo de un mensaje oral y escrito en una segunda lengua, recurriendo a conocimientos previos, elementos no verbales y contexto cultural.
11. Se comunica en una lengua extranjera mediante el discurso lógico, oral o escrito, congruente con la situación comunicativa.

Extendida

8. Valora la influencia de los sistemas y medios de comunicación en la cultura, la familia y contextos socioculturales diferentes.
10. Analiza los beneficios e inconvenientes del uso de las tecnologías de la información y la comunicación para la optimación de las actividades cotidianas.

Competencia genéricas y atributos

4. Escucha, interpreta y emite mensajes pertinentes en distintos contextos mediante la utilización de medios, códigos y herramientas apropiados.
 - 4.2. Aplica distintas estrategias comunicativas según quienes sean sus interlocutores, el contexto en el que se encuentra y los objetivos que persigue.
5. Desarrolla innovaciones y propone soluciones a partir de métodos establecidos.
 - 5.2. Ordena información de acuerdo con categorías, jerarquías y relaciones.
 - 5.6. Utiliza las tecnologías de la información y la comunicación para procesar e interpretar información.
8. Participa y colabora de manera efectiva en equipos diversos.
 - 8.1. Propone maneras de solucionar un problema o desarrollar un proyecto en equipo, definiendo un curso de acción con pasos específicos.
 - 8.2. Aporta puntos de vista con apertura y considera los de otras personas de manera reflexiva.
 - 8.3. Asume una actitud constructiva, congruente con los conocimientos y habilidades con los que cuenta dentro de distintos equipos de trabajo.

Actividad diagnóstica

Instrucciones: subraya la respuesta correcta.

1. ¿Cuál es el objetivo de un proyecto web?
 - a) Incrementar ventas
 - b) Sólo por diversión
 - c) Servicio adicional al cliente

2. No cambian en el transcurso del proyecto web, ya que si lo hacen podría cambiar todo el proyecto.
 - a) Objetivos del proyecto
 - b) Beneficio del proyecto
 - c) Diseño del proyecto

3. ¿Por qué es importante tomar en cuenta en la creación de un proyecto web, el público al que va dirigido?
 - a) Lograr el éxito del proyecto
 - b) No es necesario tomarlos en cuenta
 - c) Para mostrarles la información

4. ¿Qué es un *web hosting*?

a) Nombre del sitio	b) Servicio de almacenamiento	c) Sitio web
---------------------	-------------------------------	--------------

5. ¿Qué es un dominio?

a) Nombre del sitio	b) Servicio de almacenamiento	c) Sitio web
---------------------	-------------------------------	--------------

6. Es un ejemplo de editor web

a) Dreamweaver	b) HTML	c) FrontPage
----------------	---------	--------------

7. Es una plantilla que muestra la distribución de los elementos dentro de una página web, se utilizan espacios en blanco mostrando las dimensiones del espacio.

a) Layout	b) Wireframe	c) Mapa de navegación
-----------	--------------	-----------------------

8. Es un esquema, una guía visual que representa el esqueleto de un sitio web, ordena el contenido usualmente en este esquema

a) Layout	b) Wireframe	c) Mapa de navegación
-----------	--------------	-----------------------

9. Debe comunicar la relación entre los enlaces que contiene, de forma que los usuarios entiendan las opciones que tienen para navegar en el sitio.

a) Layout	b) Wireframe	c) Mapa de navegación
-----------	--------------	-----------------------

10. Es uno de los cronogramas más utilizados para la planeación de proyectos. En él se describen los tiempos de un proyecto, y se muestran las actividades y sus plazos de ejecución si mencionar los costos.

a) Cronograma	b) Lista de actividades	c) Diagrama de Gantt
---------------	-------------------------	----------------------



Introducción

En la actualidad la velocidad de comunicación que ofrece internet, la hace una poderosa herramienta para las empresas que ofrecen bienes y servicios o solo muestran información, es por ello que el diseño de un sitio web se vuelve mucho más complejo de lo que imaginamos, lograr la atención del público es cada vez más difícil. Es por eso que la creación de un sitio web deberá establecer una metodología para garantizar el éxito del sitio.

En este apartado se proponen cuatro fases para el desarrollo de un sitio web, como primer punto se hace la revisión de la planeación en la que se establecen los objetivos, los recursos de hardware y software necesarios mediante un cronograma de actividades; en segundo término, se hace el diseño (layout, wireframe y mapa de navegación) y la construcción del sitio para posteriormente, en tercer plano, hacer la implementación y, como última fase, la reconstrucción; ya que un sitio web debe ir modificándose con el paso del tiempo para mantenerse actualizado y ser de interés para los usuarios.

3.1. Proyecto web

3.1.1. Concepto de un proyecto web

Proyecto

Un proyecto se define como un “plan y disposición que se forma para la realización de un tratado, o para la ejecución de algo de importancia”.

O como “primer esquema o plan de cualquier trabajo que se hace a veces como prueba antes de darle la forma definitiva” (DRAE, 2017).

Proyecto web

Se le llama proyecto web, ya que tiene como resultado concreto un elemento en la red, que responde a una necesidad funcional precisa, movilizandolos recursos necesarios (humanos, técnicos, tecnológicos, materiales, etc.) para su realización, para ajustarse a los márgenes presupuestarios permitidos, respetando un calendario preestablecido (Hiard, 2017).

3.2. Componentes básicos

Al desarrollar un proyecto web es importante contar con una serie de elementos, que den validez y fiabilidad al proyecto garantizando el éxito de este. Dentro del desarrollo de un proyecto web se incluyen las siguientes fases:



3.2.1. Planeación

La fase principal en la creación de un sitio web consiste en recopilar la mayor cantidad de información posible para conocer lo que sí se quiere y lo que no se quiere para el sitio. Esta información se recopila mediante la observación, indagando en sitios web de la competencia, escuchando comentarios, mirando sitios web que nos agraden.

Para hacer un sitio web sólido, funcional y claro, en la fase de planeación es esencial determinar los siguientes puntos:

- Propósito: qué se está ofreciendo en el sitio web, información o datos, un producto o un servicio.
- Objetivo: qué hará el usuario dentro del sitio.
- Características del público objetivo: ya que son ellos los que utilizarán el sitio, los buenos sitios web se centran en el público al que están dirigidos. Para ello deben tomarse en cuenta características como: edad, antecedentes culturales, nivel profesional, así como nivel de capacidad (para descargar alguna aplicación o complemento, etcétera).
- Contenido: el éxito del desarrollo web se basa en la eficacia con la que el contenido y los gráficos comunican un mensaje al usuario. Si un gráfico tarda demasiado tiempo en cargar la página, el usuario la abandonará y el objetivo no se cumplirá. Deberá tenerse especial cuidado en la elección del contenido.

Actividad de aprendizaje 1

Elige un sitio web que te agrade y esté relacionado con el producto o servicio que tu ofreces, y analiza la información que ahí se presenta con los elementos siguientes:





URL del sitio:	
Propósito:	
Objetivo:	
Características del público: (A quién va dirigido)	
Contenido (texto, gráficos):	

3.2.2. Objetivo del proyecto web

Este punto es fundamental para iniciar un proyecto. Tener claro qué es lo que se quiere hacer permite unificar esfuerzos, minimizando costos y maximizando los recursos. Los objetivos principales por los cuales se hace un sitio web son interactuar con el usuario, comprar un producto, obtener información sobre algún tema, llenar un formulario para un presupuesto, realizar una descarga, etcétera.

Al establecer el objetivo se debe responder a la pregunta "¿qué quiero que hagan mis visitantes en el sitio web?". Al ser un pilar fundamental, no podemos dejar que los objetivos cambien durante el desarrollo del mismo, pues podría variar completamente de lo que ya se tiene.



Actividad de aprendizaje 2

De acuerdo con tu proyecto de desarrollo de un sitio web determina el objetivo que tendrá, redáctalo con un verbo en infinitivo que conteste a la pregunta "¿qué quiero que hagan mis visitantes al ver el sitio?". Después de que lo revise tu profesor (a) intégralo a tu proyecto web.

3.2.3. Recursos disponibles

En esta etapa de planeación del sitio web se deben tener en cuenta los elementos de hardware y software necesarios para su desarrollo. Entre los requerimientos disponibles para desarrollar un sitio web es necesario reconocer lo que es un servicio de hospedaje web, también conocido como **hosting**, y lo que es un nombre de dominio.

Un hospedaje web se define como el servicio que provee a los usuarios de internet, almacenamiento de información, imágenes, video, o cualquier contenido accesible mediante la red, a través de servidores que son computadoras, como las que tenemos en casa pero más poderosas y con mayor capacidad de almacenamiento, que brindan a los usuarios una dirección propia de internet. En estos servidores está guardada toda la información sobre cuentas de correo electrónico, medios sociales, blogs, aplicaciones y todo lo que se encuentra en la red de redes (Ecolohosting, 2017). Un nombre de dominio debe ser un nombre fácil de recordar asociado con una dirección IP física de internet. Se trata de un nombre único que se muestra después del signo @ en las direcciones de correo electrónico y después de www en las direcciones web.

Por ejemplo, el nombre de dominio example.com se podría traducir a la dirección física 198.102.434.8. Al utilizar un nombre de dominio para identificar una ubicación en internet, en lugar de la dirección IP numérica, resulta mucho más fácil recordar y escribir direcciones web. Cualquiera puede comprar un nombre de dominio. Solo debes ir a un **registrador** o un **host** de dominios, encontrar un nombre que nadie más esté usando y abonar una pequeña cuota anual para ser el propietario (Google, 2017).

El nombre de dominio debe tener las siguientes características:

- Relacionarse con la empresa, servicio o producto que se ofrece.
- Fácil de recordar.
- Debe ser corto.
- Evitar letras que fonéticamente se confundan.

Para ser publicado el sitio en internet, se debe reservar el dominio y ubicar el servicio de hospedaje en empresas que se especializan en brindar este servicio.

Otros recursos que deben considerarse son los editores de diseño web (aplicaciones que son diseñadas especialmente para la creación de documentos electrónicos desarrollados en lenguaje HTML o XHTML), y las animaciones. También debes tener en cuenta las licencias, si es que se trata de software comercial o software libre. Este tema será tocado en el módulo 4, "Implementación del sitio web".

Actividad de aprendizaje 3

Selecciona un nombre de dominio con las características antes expuestas. Después debe ser revisado por tu profesor (a) intégralo a tu proyecto web.





Actividad de aprendizaje 4

Haz una investigación en la que señales al menos tres servicios de alojamiento de sitios web. Cuando la información que obtuviste esté aprobada por tu profesor (a) intégralo a tu proyecto web.

Nombre del servicio	Características que ofrece	Costo del servicio

3.2.4. Beneficio esperado

Los beneficios esperados que se obtienen de la implementación de un sitio web bien estructurado para una empresa que ofrece bienes y servicios son:

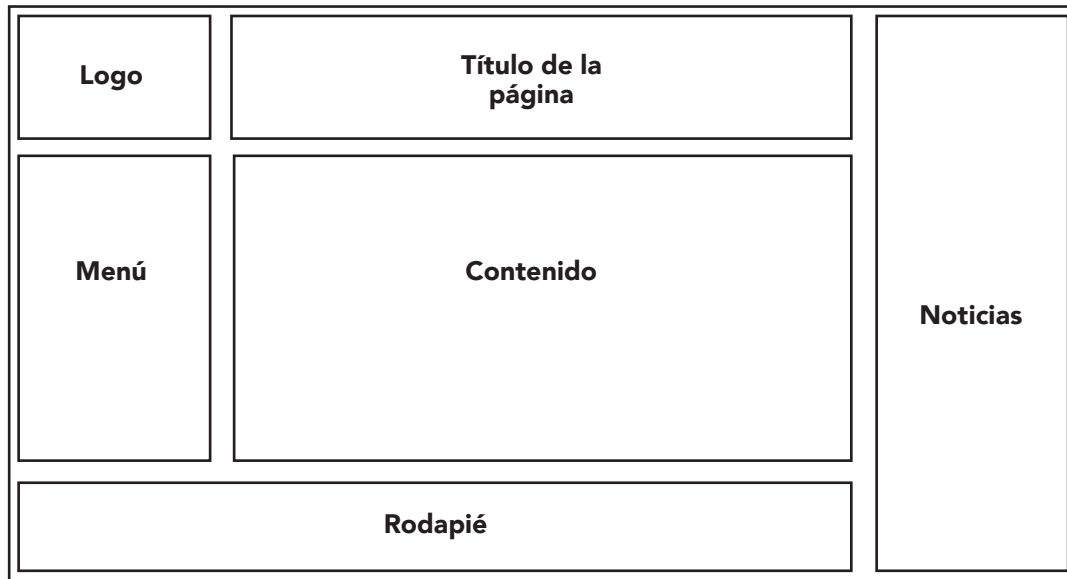
- Mejorar el servicio al cliente.
- Disponibilidad las 24 horas del día.
- Facilita a los clientes información detallada acerca de sus productos y servicios.
- Promueve la compra en línea.
- Reduce costos en publicidad.
- Ampliación de un mercado local a un mercado nacional o internacional.

3.2.5. Diseño estructurado

El diseño de una página está directamente relacionado con la distribución de la información y la funcionalidad de la misma. Para garantizar la funcionalidad y la distribución es preciso hacer el *layout* y el *wireframe*, generando un concepto del sitio antes de la construcción de este.

Un **layout** o **plantilla** muestra la distribución de los elementos dentro de una página web; se utilizan espacios en blanco que muestran las dimensiones del mismos. Esto garantiza la distribución del sitio, como se muestra a continuación:

Un **wireframe** o **esquema** es una guía visual que representa el esqueleto de un sitio web, ordena el contenido, usualmente carece de color y de estilo tipográfico, su objetivo radica en la funcionalidad, el comportamiento y la jerarquía del contenido.



- Diseño de información: es la ubicación en la presentación y priorización de información de manera que pueda facilitar el entendimiento. Para sitios web, los elementos informativos deben estar ordenados de manera que reflejen los objetivos y tareas del usuario.
- Diseño de navegación: debe comunicar la relación entre los enlaces que contiene, de forma que los usuarios entiendan las opciones que tienen para navegar dentro del sitio.
- Diseño de interfaz: incluye seleccionar y ordenar los elementos de la interfaz para permitir a los usuarios interactuar con la funcionalidad del sistema. El objetivo es facilitar la usabilidad y eficiencia tanto como sea posible. Elementos comunes que se encuentran en el diseño de interfaces son los botones de acción, campos de texto, cajas de verificación, botones y menús desplegables.

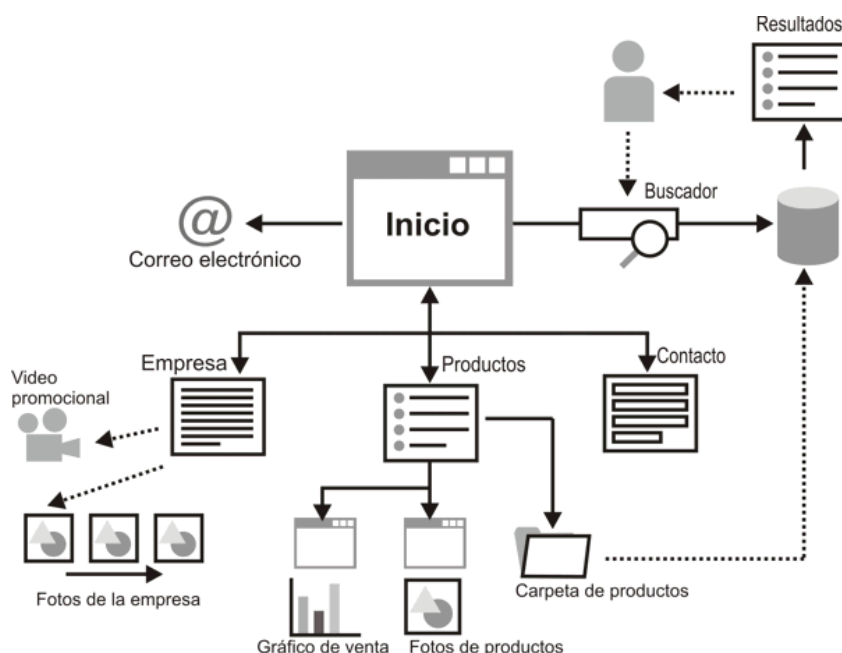


Figura 3.1.
Ejemplo de
un mapa de
navegación.

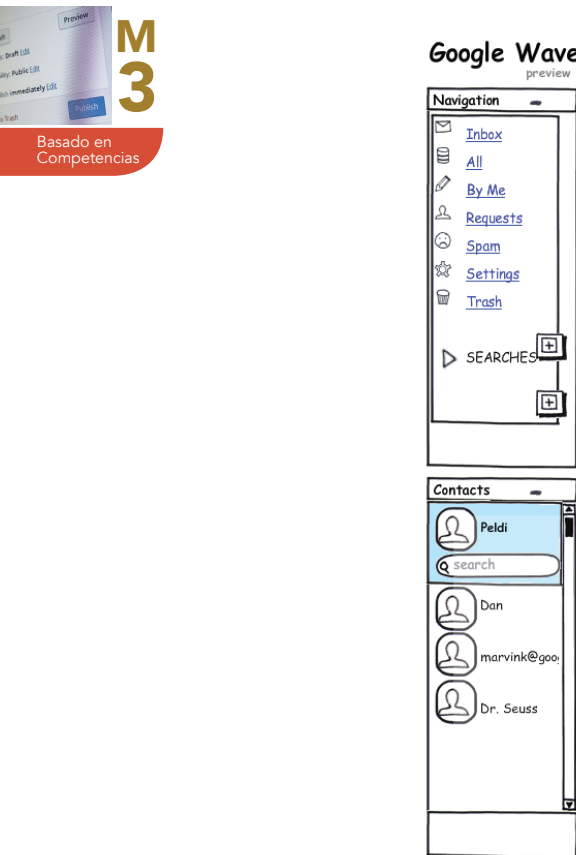


Figura 3.2. Ejemplo de *wireframe*.

Una vez terminado el *layout* y el *wireframe* se muestra al cliente, para obtener su aprobación. Y si es necesario se hacen las modificaciones pertinentes.

Fuentes y colores

Un aspecto primordial acerca de los sitios web es el formato de la información. Debe ser legible y visualmente agradable, esto por lo general es el factor de éxito en los sitios. Los tipos y estilos de letra desempeñan un papel clave en el aspecto de las páginas: el grosor, el tamaño de la letra y tipografía con o sin patines, así como la armonía de los colores y los gráficos, son características que se deben considerar al momento de diseñar un sitio web. Para diseñar todo esto deben tenerse en cuenta las características del público al que está dirigido el sitio.



Actividad de aprendizaje 5

En un procesador de texto integra tu proyecto web, el *layout* y el *wireframe*, con su mapa de navegación con las características antes mencionadas. Intégralo para que tu profesor (a) haga la revisión correspondiente.

3.2.6. Cronograma de actividades

Una de las herramientas más efectivas para la planeación de un proyecto son los cronogramas, pues nos permiten estructurar y evaluar cada una de las actividades realizadas en el desarrollo de un sitio web. Los cronogramas permiten temporizar las actividades, para dar una fecha límite de entrega del proyecto, ayudar a establecer las expectativas de los clientes, y poder anticipar costos no contemplados o extensiones de lapsos de entrega.

El diagrama de Gantt es uno de los cronogramas más usados para la planeación de proyectos. En él se describen los tiempos de un proyecto, se muestran las actividades y sus plazos de ejecución sin mencionar los costos. Esta herramienta permite contestar las preguntas: ¿cuándo terminará el proyecto?, ¿cuándo se hará tal actividad?, ¿qué avance de proyecto tendré en tal fecha?

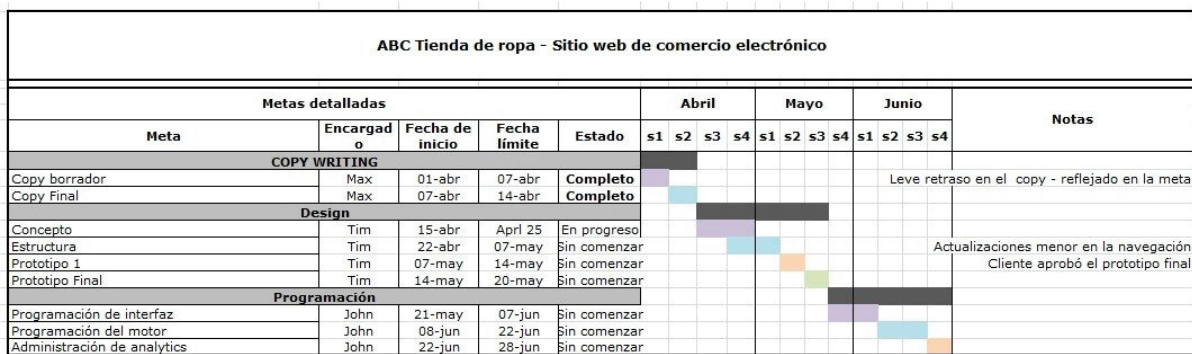


Figura 3.3. Ejemplo de gráfica de Gantt.

Actividad de aprendizaje 6



Actividad

En un procesador de texto integra tu proyecto web, una gráfica de Gantt en la que se vean reflejadas las fases de un proyecto web con las características que aquí se proponen.

Resumen

En este módulo 3, "Proyecto Web", repasamos las fases de realización de este, vimos que la planeación del sitio es lo más importante en el proceso, pues si no se tiene una buena planeación nuestro proyecto no responderá a las necesidades por las cuales se creó. Es primordial establecer el objetivo, ya que de él parte toda la planeación, el diseño y la construcción del proyecto. Se debe investigar la mejor opción de almacenamiento del sitio web, así como el dominio más indicado para generar impacto en los usuarios.

Una vez terminada la fase de planeación o de saber qué es lo que se necesita para el proyecto, viene la fase de diseño del sitio, para hacerlo existen algunas herramientas como el layout, el wireframe y el mapa de navegación. Posteriormente construimos nuestro sitio web.



Autoevaluación

¿Considero que lo hice bien?

Este es un proceso de reflexión acerca de lo que obtuviste después de hacer la planeación y el diseño de tu proyecto web. Responde la tabla siguiente, coloca una X en el cuadro correspondiente a cada ítem (5 representa "estoy totalmente de acuerdo" y 1 "estoy totalmente en desacuerdo"):

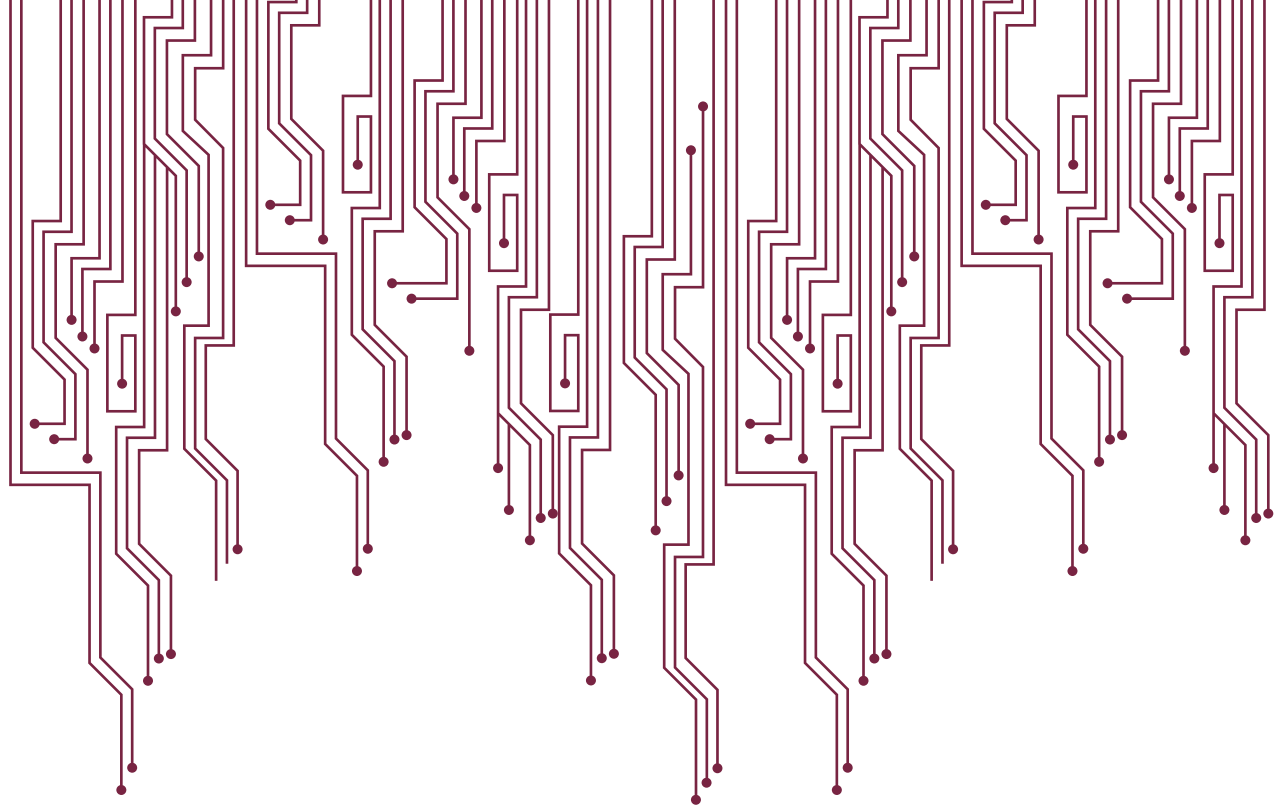
	5	4	3	2	1
1. Asumí una actitud propositiva, entusiasta, responsable.					
2. El aprendizaje y las experiencias que me dejó la participación en la actitud me son útiles.					
3. Pude explicar con claridad los conceptos y definiciones que revisamos y aplicamos a lo largo de cada actividad.					
4. Puedo aplicar lo aprendido en mi vida diaria.					
Mi calificación general sería de: (anota un valor numérico en escala del 1 a 10).					

Coevaluación

Puedes compartir la autoevaluación que hiciste con tus compañeros con la intención de intercambiar ideas o experiencias de aprendizaje. En caso de que hayas tenido alguna dificultad sería una buena aportación compartirla.

La calificación de los compañeros con quienes intercambié mi autoevaluación fue (anotar los valores numéricos que tenga asignados).

Ítem	Compañero 1	Compañero 2	Compañero 3
1			
2			
3			
4			



Módulo 4

Implementación del sitio web



Imagen: pxhere.com

Elabora e implementa un sitio web con un proyecto útil, veraz e innovador para difundir información.



Competencias disciplinarias

Comunicación

Básicas

9. Analiza y compara el origen, desarrollo y diversidad de los sistemas y medios de comunicación.
10. Identifica e interpreta la idea general y el posible desarrollo de un mensaje oral y escrito en una segunda lengua, recurriendo a conocimientos previos, elementos no verbales y contexto cultural.
11. Se comunica en una lengua extranjera mediante el discurso lógico, oral o escrito, congruente con la situación comunicativa.

Extendidas

8. Valora la influencia de los sistemas y medios de comunicación en la cultura, la familia y los contextos socioculturales diferentes.
10. Analiza los beneficios e inconvenientes del uso de las tecnologías de la información y la comunicación para la optimación de las actividades cotidianas.
11. Aplica las tecnologías de la información y la comunicación en el diseño de estrategias para la difusión de productos y servicios, en beneficio del desarrollo personal y profesional.

Competencias genéricas y atributos

4. Escucha, interpreta y emite mensajes pertinentes en distintos contextos mediante la utilización de medios, códigos y herramientas apropiados.
 - 4.2. Aplica distintas estrategias comunicativas de acuerdo con sus interlocutores, el contexto en el que se encuentra y los objetivos que persigue.
5. Desarrolla innovaciones y propone soluciones a partir de métodos establecidos.
 - 5.2. Ordena información de acuerdo con categorías, jerarquías y relaciones.
 - 5.6. Utiliza las tecnologías de la información y la comunicación para procesar e interpretar información.
8. Participa y colabora de manera efectiva en equipos diversos.
 - 8.1. Propone maneras de solucionar un problema o desarrollar un proyecto en equipo, definiendo un curso de acción con pasos específicos.
 - 8.2. Aporta puntos de vista con apertura y considera los de otras personas de manera reflexiva.
 - 8.3. Asume una actitud constructiva, congruente con los conocimientos y las habilidades con los que cuenta dentro de distintos equipos de trabajo.

Actividad diagnóstica

Instrucciones: subraya la respuesta o respuestas correctas.

1. Es una aplicación que está destinada a la construcción, diseño y edición de sitios, videos y aplicaciones web basados en estándares:

- a) Adobe Photo Shop
- b) Adobe Dreamweaver
- c) Microsoft FrontPage

2. Estas nos permiten la comunicación de manera remota entre dos estaciones, copiar archivos desde un equipo a un servidor y transferir archivos y carpetas:

- a) Aplicaciones de diseño b) Aplicaciones HTML c) Aplicaciones FTP

3. De los siguientes elementos, ¿cuáles son gráficos de una página web?

- a) Botones b) Íconos c) Fondos

4. Es un grupo de páginas web relacionadas:

- a) Archivo HTML b) Página web c) Sitio web

5. Selecciona cuál de estas opciones son usos del texto en la web:

- a) Menús b) Títulos c) Contenido

6. Cuando estamos haciendo una página web, nuestros archivos de esta página deben estar guardados:

- a) En el escritorio de la computadora
- b) En cualquier lugar de la computadora c) En una misma carpeta

7. ¿Cómo definimos un color en sistema hexadecimal?

- a) Rojo b) 245.2554.220 c) FFF5DC

8. Son los tipos de etiquetas:

- a) Etiqueta de inicio, etiqueta de fin y etiqueta de inicio y cierre
- b) Etiqueta de inicio, etiqueta de apertura y etiqueta de inicio y cierre
- c) Etiqueta de cierre, etiqueta de fin y etiqueta de inicio y cierre

9. Es el lenguaje de marcado para la elaboración de páginas web:

- a) FrontPage b) HTML c) Dreamweaver

10. Son las aplicaciones que permiten ver las páginas web:

- a) Hipervínculos b) Exploradores c) Mapa de Bits



Introducción

¡Bienvenidos al mundo de las páginas web! Un mundo que puede ser muy beneficioso para ustedes.

En este módulo les explicaremos los aspectos básicos que conforman una página web, cómo funcionan y cómo pueden resultarles útiles.

Recuerden que en computación aplicada veremos el diseño y la creación de páginas web, y que podemos poner en marcha su sitio.

Por lo general, un sitio web se construye en torno de una página central, la “página de bienvenida”, que ofrece vínculos a un conjunto de páginas alojadas en el mismo servidor y, a veces, a vínculos “externos” que llevan a páginas alojadas en otro servidor.

4.1. Software desarrollador de sitios web

Al momento de desarrollar un sitio web se debe comenzar por responder a un grupo muy concreto de preguntas, que permitirán aclarar y definir de manera básica las expectativas existentes en torno a la aplicación web y la interacción que se generará.

Dichas preguntas podrían ser las siguientes:

1. ¿En realidad es necesario el sitio web?
2. ¿Para qué?
3. ¿Qué es lo que buscarán los usuarios en ese sitio?
4. ¿Qué imagen se quiere proyectar a los usuarios con el sitio web?
5. ¿Qué servicios interactivos se integrarán a través del sitio web?
6. ¿Con qué recursos se cuenta para el desarrollo del sitio web?
7. ¿Cuánto tiempo debe invertirse en el desarrollo del sitio web?
8. ¿Qué impacto causará el sitio web en los usuarios?

Todo proyecto debe contar con objetivos definidos, contra los cuales puede contrastarse la efectividad de las acciones que se desarrollen.

Un sitio web, por lo tanto, es un espacio virtual en internet. Se trata de un conjunto de páginas web, accesibles desde un mismo dominio o subdominio de la World Wide Web (WWW).

Es importante establecer que en internet encontramos una gran variedad de tipos de sitios web que suelen diferenciarse fundamentalmente por la clase de contenido que ofrecen o por el servicio que brindan a cualquiera de las personas que se encuentran navegando por la red.

Por ejemplo, podemos destacar los conocidos blogs. Son una especie de bitácora en la que una persona en particular escribe post relacionados con sus gustos, sus aficiones o sus conocimientos de una manera determinada. De esta forma

podemos encontrar los blogs que son utilizados simplemente para contar, en forma de diario, lo que les sucede día con día, o especializados en libros, en cine, o en cualquier tema que se te ocurra. Para crearlos existen diversas plataformas gratuitas, como Blogger, Blogspot o WordPress.

Los sitios web incluyen documentos HTML, fotografías, sonidos, videos, animaciones Flash y otro tipo de contenidos que pueden compartirse en línea. La URL raíz del sitio web se conoce como **portada** o **homepage**. Lo más habitual es que esta portada facilite el acceso a todas las páginas del sitio mediante hipervínculos (también conocidos como **enlaces** o **links**).

Por ejemplo: el sitio web más visitado en el mundo Google se encuentra en <http://www.google.com/>. Es decir, Google.com es la homepage del sitio. Desde allí es posible tener acceso al resto de las páginas y aplicaciones que componen el sitio, como la aplicación de Drive (<https://drive.google.com/drive>) o la tienda (<https://play.google.com/store>).

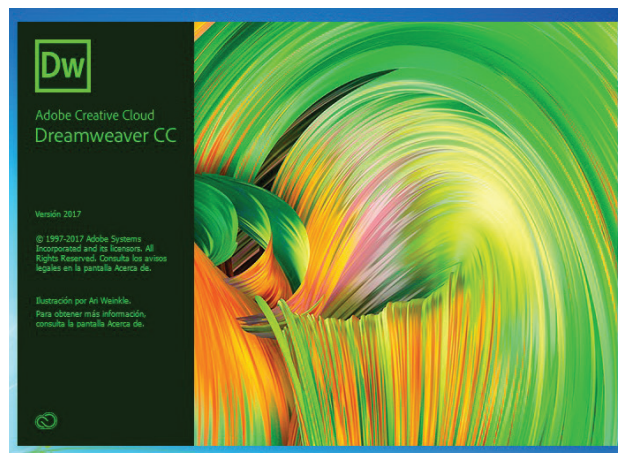
Existe una gran cantidad de programas y herramientas que nos permiten crear páginas web. Algunas de ellas mucho más complejas que otras, esto lo podemos ver en la clasificación que daremos a continuación, también haremos un repaso acerca de sus costos, algunas tienen uno muy elevado.

Aunque para crear páginas web existen infinidad de programas y herramientas, la realidad es que son muy pocos aquellos que se acoplan a las necesidades de las personas con pocos y escasos conocimientos en desarrollo web.

Su objetivo es claro: que tú y otras personas que no se dedican precisamente al desarrollo web puedan tener como alternativa una herramienta intuitiva, versátil y novedosa, pero con el poderío de las herramientas de nivel profesional.

Adobe Dreamweaver

Con seguridad esta es una de las herramientas que conoces o que al menos has escuchado mencionar desde hace tiempo. Se trata de un editor HTML para diseñar, codificar y desarrollar sitios, páginas y aplicaciones web. Si necesitas hacer un sitio con calidad, sin duda Dreamweaver es una buena opción. Sin embargo, para poder sacarle el máximo provecho a esta herramienta deberás contar con muchos conocimientos en el desarrollo de páginas web o lenguajes de programación en web, como HTML o el mismísimo Javascript.



Aunque es cierto que Dreamweaver no ofrece una gran cantidad de ventajas, cuenta con un ambiente gráfico, es compatible con Windows y Mac. Es un software de paga; es decir, que trabaja mediante una licencia que debe estar instalada en

Figura 4.1.
Dreamweaver.



tu computadora. Quizás esto suene bien, sin embargo, no es precisamente lo que la tendencia marca en el software actual.

WebSite X5 Evolution



WebSite X5 Evolution se trata de una herramienta más económica, cuenta con muchas ventajas.

Características de WebSite X5 Evolution:

- Cuenta con versión económica y una costosa.
- Cuenta con diseño *responsive*.
- Se adapta a cualquier dispositivo.
- Tiene un diseño de plantillas incluido.
- Permite subir tu sitio a internet.

Figura 4.2. WebSite X5 Evolution.

Avanquest WebEasy Professional



Para continuar con las herramientas WYSIWYG para crear páginas web tenemos Avanquest WebEasy Professional. La carta principal de este programa es la posibilidad que ofrece de tener a la mano más de mil combinaciones posibles de plantillas. Si consideramos que en total cuenta con unas 600 plantillas para personalizar tus diseños, sin duda se vuelve una herramienta indispensable para quienes buscan algo más visual y sencillo, aunque de nueva cuenta, te avisamos que son necesarios conocimientos para trabajar con una herramienta WYSIWYG.

Es mucho más barato que las herramientas anteriores, por lo que las ventajas se suman considerablemente. Así que veremos las características de Avanquest, para ver que más nos ofrece.

Características de Avanquest WebEasy Professional:

- Editor WYSIWYG.
- Precio más económico, aunque no demasiado.
- No es recomendable para usuarios avanzados.
- No necesitas saber programar.
- Cuenta con módulos para redes sociales.

Figura 4.3. Avanquest WebEasy Professional.



MAGIX Web Designer 11

Continuamos con los editores HTML que nos ayudan a crear páginas web, así que debemos mencionar a MAGIX. Este editor cuenta con esa fórmula que cualquiera necesita para poder crear un sitio web exitoso con facilidad. Solo por ser parte de la familia Magix, esto ya garantiza su renombre, pues es una de las empresas más reconocidas en la actualidad en cuestiones multimedia. Esta herramienta ofrece:

Figura 4.4. MAGIX Web Designer 1.

Características de MAGIX Web Designer 11

- Adaptación para cualquier dispositivo.
- Edición de imágenes básica.
- Integración con redes sociales.
- Cuenta con versión básica y Premium.
- La versión Premium cuenta con hospedaje de regalo y un dominio.

¿Para quiénes son ideales este tipo de herramientas? Principalmente para personas que ya conocen un poco del medio, del lenguaje de programación y del ambiente de trabajo óptimo para el desarrollo de un sitio web desde cero. Por fortuna, dentro de los programas para crear páginas web que se lanzaron en 2017, ya podemos encontrar los editores visuales.

Editores visuales para crear páginas web

Jimdo

¿Quieres hacer una tienda *online* y no sabes cómo? Con un editor de los ya mencionados no podrás lograrlo tan fácilmente. Sin embargo, en el ámbito de los editores visuales existen herramientas especializadas en lo que son las tiendas *online*. Un ejemplo de este tipo de herramientas es Jimdo.

Como ya se dijo, Jimdo es una herramienta muy fácil de usar; es una de las características a su favor.



Figura 4.5. Jimdo.

Otras características favorables son que Jimdo cuenta con opciones de optimización, que te brindan la posibilidad de disfrutar de una cuenta de e-mail de forma gratuita, lo que será muy útil para hacer negocios, sin duda.

Wix

Esta es quizás una de las herramientas más conocidas, pues se le ha hecho mucha publicidad. Wix es uno de los editores *online* más utilizados. ¿Qué nos ofrece?

Una de sus principales características son sus estupendos diseños, que son muy visuales. Esto es lo más sobresaliente, y la razón por la que muchas personas que crean sitios la prefieren.



Figura 4.6. Wix.

Wix también destaca por contar con una buena cantidad de animaciones y, por supuesto, la posibilidad de adquirir aplicaciones en la *app market*, algo que otras plataformas de este tipo no tienen. Wix también cuenta con una versión gratuita demo.



Figura 4.7.
Weebly.

Y en segundo; Weebly no cuenta con soporte en español, por lo que si llegas a tener algún problema, seguramente se te hará muy complicado entender la plataforma. Aunque es muy fácil de utilizar, su soporte en inglés lo complica.

Weebly

Si nuestro enfoque está dirigido a la facilidad de uso es posible que esta herramienta nos ayude. Weebly quizás es superior a las dos herramientas anteriores. Su facilidad de uso es impresionante y sus diseños son muy bonitos, incluso utilizar su tienda *online* es muy sencillo.

Sin embargo, hay un detalle con Weebly. En primer lugar, comprar un dominio en Weebly es muy costoso.

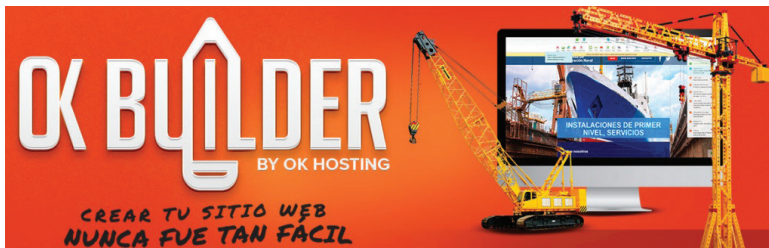


Figura 4.8. Ok
Builder.

Entre las ventajas que OK Builder nos ofrece se destacan las más de 172 plantillas para crear un sitio web, diseños elegantes y acordes con lo que se usa hoy en día en tendencias de diseño web.

OK Builder

Ok Builder tiene una característica fundamental: es tan sencillo de usar que prácticamente no es necesario tener ningún conocimiento previo en cuestiones de programación o lenguajes informáticos, y además puedes hacer una prueba gratis para que conozcas sus bondades con anticipación.

Esto sumado al hecho de que se puede crear un sitio web directamente desde un dispositivo móvil. De esta forma puedes subir fotos, generar información o incluso logotipos en minutos.

Algo que OK Builder te ofrece, y otras herramientas no, es la posibilidad de optimizar tu sitio web frente a buscadores como Google. Te facilita mantener bien optimizado el contenido y cada una de las secciones para que puedas analizar tu posicionamiento constantemente y no dependas solo de las visitas por redes sociales, de tal forma que las personas que necesiten algo de tu sitio web puedan encontrarlo desde el mismo Google.

Actividad de aprendizaje 1



Actividad

Haz un cuadro comparativo con los ocho programas mencionados anteriormente, en el cual incluyas ventajas, desventajas y costos.

Veremos cuáles son los elementos básicos de Dreamweaver CC: la pantalla, las barras, los paneles, etc., para saber diferenciar entre cada uno de ellos. Aprenderemos cómo se llaman, dónde están y para qué sirven. También veremos cómo obtener ayuda, por si en algún momento no sabemos cómo seguir trabajando.

Los requerimientos del sistema para un mejor desempeño del programa y los sistemas operativos recomendados para Dreamweaver MX son los siguientes:

Requisitos del sistema para Dreamweaver CC 2017.5

Windows

1. Procesador Intel® Core 2 o AMD Athlon® 64 (2 GHz o más rápido).
2. Microsoft Windows 7 con Service Pack 1, Windows 8.1 o Windows 10.
3. 2 GB de RAM (se recomienda 4 GB).
4. 2 GB de espacio disponible en disco duro para la instalación; se requiere espacio libre adicional (aproximadamente 2 GB) durante la instalación. Dreamweaver no se puede instalar en dispositivos de almacenamiento flash extraíbles.
5. Resolución de 1280 x 1024 con tarjeta de video de 16 bits.
6. Es preciso disponer de conexión a internet y estar registrado para poder activar el software, validar las suscripciones y acceder a los servicios en línea.

MacOS

- Procesador Intel multinúcleo con compatibilidad de 64 bits.
- MacOS v10.12, OS X v10.11 o v10.10.
- 2 GB de RAM (se recomienda 4 GB).
- 2 GB de espacio disponible en disco duro para la instalación; se requiere espacio libre adicional (aproximadamente 2 GB) durante la instalación. Dreamweaver no se puede instalar en dispositivos de almacenamiento flash extraíbles.
- Resolución de 1280 x 1024 con tarjeta de video de 16 bits.
- Es preciso disponer de conexión a internet y estar registrado para poder activar el software, validar las suscripciones y acceder a los servicios en línea.

Actividad de aprendizaje 2



Actividad

Revisa los recursos con los que cuenta tu computadora en casa y valida si es factible la instalación.



4.1.1. Entorno de trabajo para desarrollar un sitio web

El espacio de trabajo de Dreamweaver CC permite ver las propiedades de los documentos y los objetos, además de la barra de herramientas para que puedas hacer operaciones en los documentos rápidamente.

Nota: las imágenes mostradas en este módulo se obtuvieron de Adobe Dreamweaver CC 2017. Algunos términos y procedimientos se tomaron como referencia de las páginas de internet descritas en las referencias electrónicas.

Además, coloca muchas de las operaciones más frecuentes en barras de herramientas para que puedas hacer cambios en los documentos rápidamente.

Una vez que tenemos el software en nuestro equipo y lo utilizamos por primera vez, se desplegará la figura siguiente, que podemos cambiar una vez que el modo de trabajo ha comenzado.

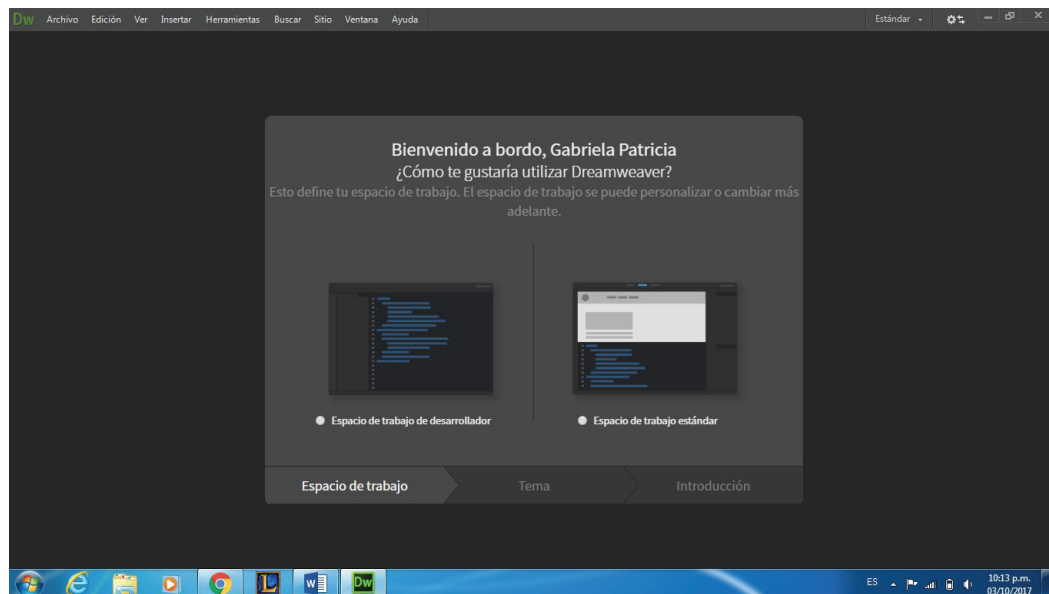


Figura 4.9. Espacio de trabajo.

Una vez que elegimos el espacio de trabajo nos aparecerá la pantalla de la figura 3.13, en la pantalla del espacio de trabajo Diseño podremos ver sus componentes fundamentales. Es importante estudiar los nombres de los diferentes elementos para que sea más fácil entender el resto del curso.

La pantalla que se muestra a continuación (y en general todas las de este curso) quizá no coincidan exactamente con la que ves en tu ordenador, ya que cada usuario puede decidir qué elementos quiere ver en cada momento y dónde verlos.

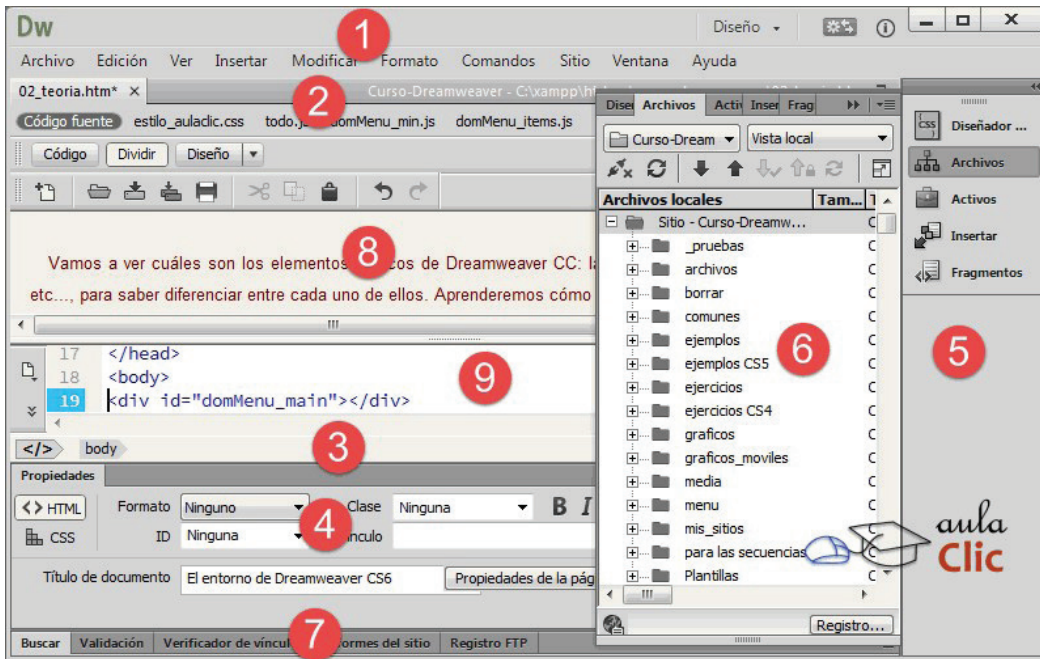


Figura 4.10. Entorno de Dreamweaver CC.

1. Las barras.
2. Las pestañas de documentos.
3. La barra de estado.
4. El panel de propiedades.
5. Área de paneles.
6. Paneles y ventanas.
7. Línea inferior de pestañas.
8. Vista diseño del documento.
9. Vista código del documento.

1. Las barras

La barra de aplicación

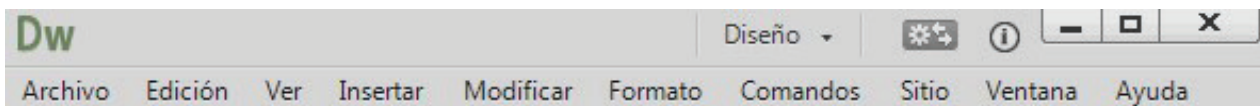
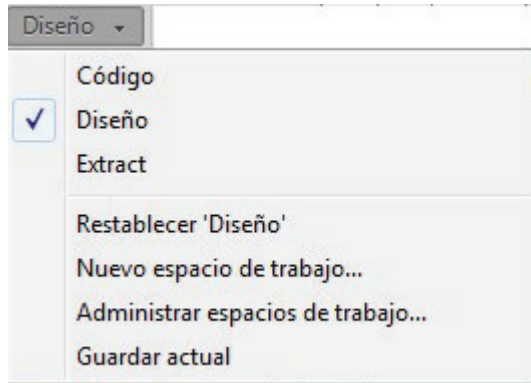


Figura 4.11. Barra de aplicación.

La barra de aplicación se incorporó en la versión anterior de Dreamweaver. Si tenemos la ventana maximizada veremos todos sus elementos.

Esta barra contiene los siguientes elementos: los menús (en la imagen, en la parte inferior, Archivo, Edición, Ver, etc.), el botón de ajuste de sincronización, el conmutador de espacio de trabajo y el botón del sistema de ayuda *online*.



Más hacia la derecha observamos el conmutador del espacio de trabajo, como un desplegable. Un espacio de trabajo es la configuración del entorno (paneles visibles y su disposición) que podemos guardar y cargar, esta es su imagen desplegada.

Figura 4.12. Menú desplegado.

Los menús están agrupados en categorías.

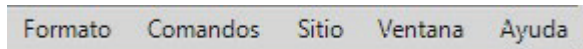


Figura 4.13. Menú categorías.

2. Las pestañas de documento

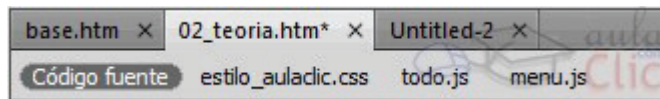


Figura 4.14. Pestañas de documento.

Cada archivo que tengamos abierto mostrará una pestaña con su nombre, lo que nos permitirá cambiar de uno a otro fácilmente. Si junto al nombre aparece un *, indica que ese archivo tiene cambios sin guardar. Podemos cerrar cada documento haciendo clic en el tache, o acceder a otras acciones haciendo clic con el botón derecho.

Debajo de las pestañas encontramos los archivos que utiliza nuestra página, se puede acceder a ellos con un clic. Esto nos ahorrará bastante tiempo.

3. La barra de estado

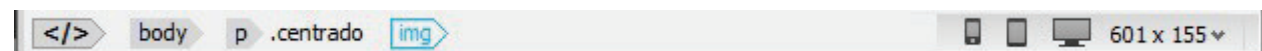


Figura 4.15. Barra de estado.

Encontramos esta barra debajo de la ventana de documento, y nos da información sobre el mismo.

A la izquierda, encontramos el selector de etiquetas. Nos sirve para seleccionar etiquetas completas. Más a la derecha encontramos tres iconos para ver la página en tres tamaños de pantalla: móvil, tableta y PC escritorio. Al final hay otros datos como el tamaño de la ventana, el tamaño de la página o su codificación.

La barra de herramientas estándar



Figura 4.16. Barra de herramientas estándar.

La barra de herramientas estándar contiene íconos para realizar las acciones más habituales del menú Archivo y Edición. De izquierda a derecha: Nuevo archivo, Abrir, Guardar, Guardar todo, Imprimir el código fuente, Cortar, Copiar, Pegar, Deshacer y Rehacer. Esta barra puede ser muy útil, aunque en realidad casi todo el mundo usa esos comandos directamente con el teclado (Ctrl + G para guardar, Ctrl + C para copiar, etcétera).

La barra de herramientas de documento



Figura 4.17. Barra de herramienta de documento.

La barra de herramientas de documento contiene íconos que nos permiten cambiar entre las distintas vistas de edición y la vista en vivo, vista previa en multipantalla, tener acceso cómodamente al administrador de archivos y, si está activa la vista en vivo, botones de navegación de nuestro sitio. Los elementos que presenta dependen de la vista que estemos utilizando.

Dreamweaver utiliza ventanas flotantes similares a las barras de herramientas, que se conocen como paneles (ventanas) o inspectores. La diferencia entre panel e inspector es que, en general, la apariencia y opciones de un inspector cambian dependiendo del objeto seleccionado, mientras que el panel nos da acceso a opciones generales fijas.

A través de la opción Ventana, de la barra de menús, es posible mostrar u ocultar cada uno de los paneles o inspectores. A continuación, veremos el panel de Propiedades y más adelante el panel Insertar.

4. El panel de Propiedades

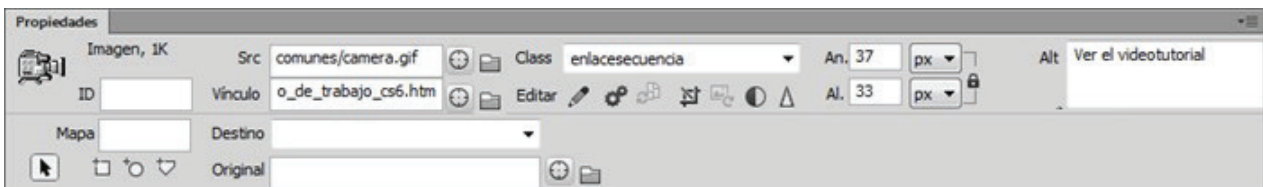


Figura 4.18. El panel de Propiedades.

El inspector de Propiedades muestra las opciones propias del objeto o texto seleccionado, permitiéndonos editarlas, por lo que se convierte en uno de los elementos más utilizados en Dreamweaver. Por ejemplo, cuando el elemento seleccionado sea una imagen, como sucede en la imagen que puedes ver, mostrará su ubicación, dimensiones, peso, clase, etcétera.

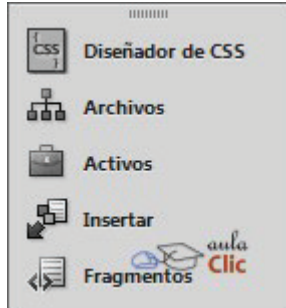


Figura 4.19. Área de paneles.

5. Área de paneles

En la parte derecha de la pantalla tenemos el área o pila de paneles.

Inicialmente el área de paneles contiene unos determinados elementos, pero podemos quitar y poner los elementos que queramos. Para quitar un elemento del área de paneles basta hacer clic con el botón derecho sobre su nombre para que aparezca un menú con la opción Cerrar. Para añadir un elemento al área de paneles hay que ir al menú Ventana y hacer clic en el elemento que queramos añadir, si el elemento se abre en una ventana flotante bastará arrastrarlo al área de paneles.

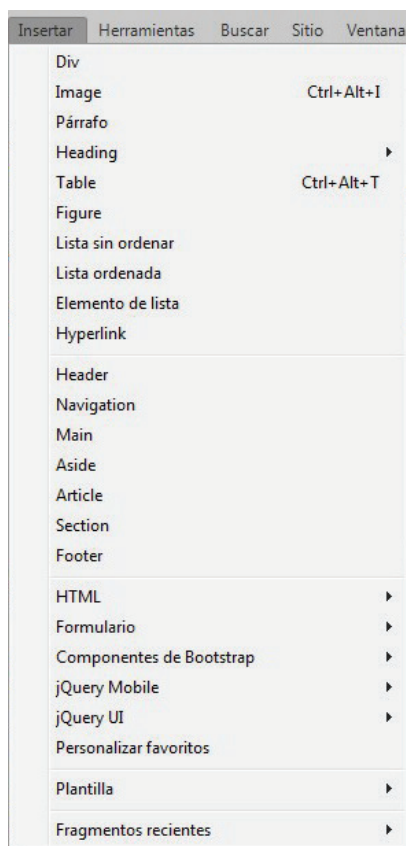


Figura 4.20.
Panel insertar.

6. Paneles

Una vez que tenemos el área de paneles con los elementos que más nos interesen, podemos abrirlos o desplegarlos de tres formas.

1. Abrir el panel dentro del área de paneles.
2. Abrir el panel al lado del área de paneles.
3. Abrir como panel flotante.

El panel Insertar

En el panel Insertar encontramos todos los elementos que podemos encontrar en ese menú, clasificados en categorías. Podemos emplearlo para insertar imágenes, enlaces, multimedia, tablas, formularios... cualquier elemento que nuestra web necesite.

7. Línea inferior de pestañas

En la parte inferior de la pantalla tenemos una línea que contiene varias pestañas (Buscar, Validación, etc...)

Para abrir una pestaña basta hacer clic sobre ella, para cerrarla hay que hacer doble clic en su título.



Figura 4.21.
Línea inferior de
pestañas.

Vistas de un documento

Podemos editar el documento con distintas vistas, que puedes cambiar a través de la barra de documento:



Figura 4.22. Vista de documento.

8. La vista Diseño

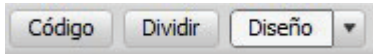


Figura 4.23. Menú de vistas.

La vista Diseño permite trabajar con el editor visual.

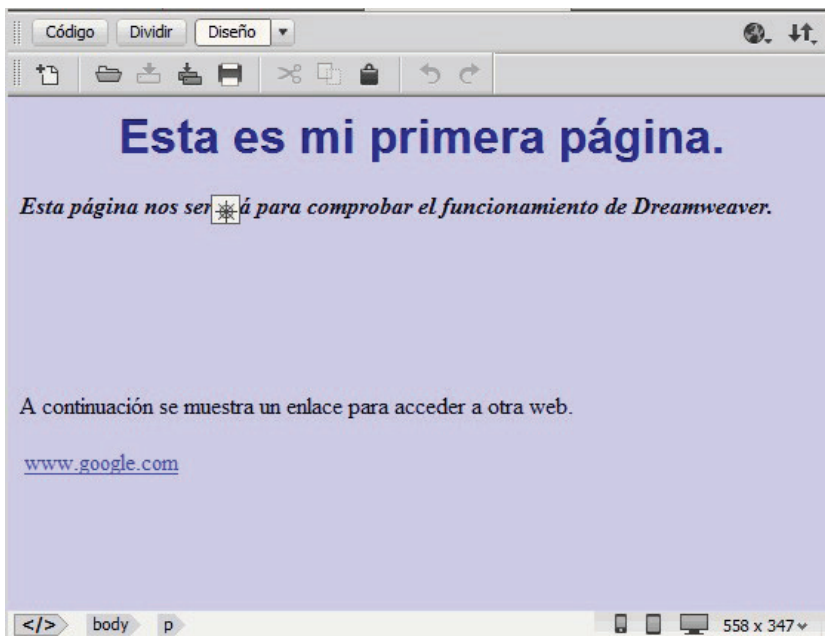


Figura 4.24. Vista Diseño.

Nos ofrece un aspecto muy similar al resultado final, pero totalmente editable.

Actividad de aprendizaje 3

En vista Diseño crea un sitio web de prueba con el nombre de la asignatura, similar a la figura 4.24.





9. La vista Código

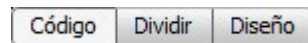
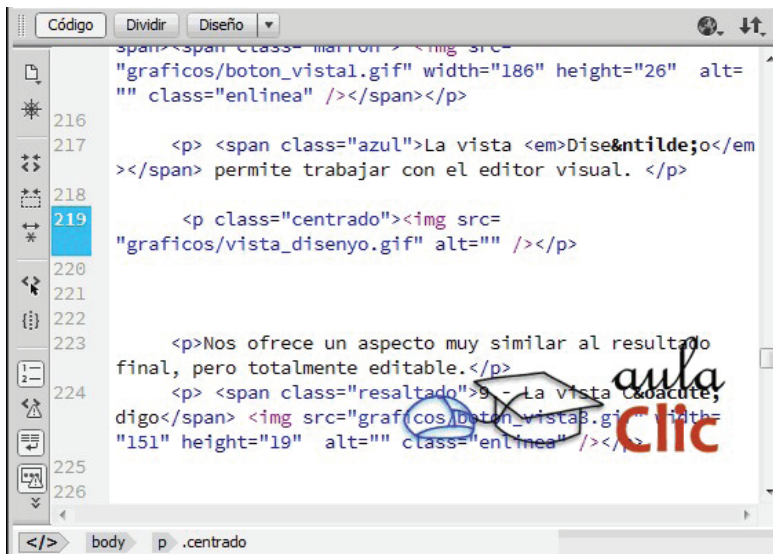


Figura 4.25. Menú vistas.

La vista Código se utiliza para poder trabajar en un entorno totalmente de programación, de código fuente. No permite tener una referencia visual directa de cómo va quedando el documento según se modifica el código.



El código que vemos es el que genera Dreamweaver al editar el contenido en la vista Diseño.

La mayor parte del tiempo trabajaremos en vista Diseño, pero es importante que sepamos manejar el código HTML, por lo cual se propone la actividad 4.

Figura 4.26. Vista Código.



Actividad de aprendizaje 4

En vista Código elabora una plantilla de documento HTML.

- 1) Para comenzar vamos a vista Código.
- 2) Como cualquier página HTML, tendrá la cabecera y el cuerpo del documento, ambas partes situadas entre la etiqueta de apertura < HTML> y su análoga de cierre < /HTML>
- 3) La cabecera del documento sería:


```
<HTML>
<head>
<title>COMPUTACIÓN APLICADA </title>
</head>
```
- 4) El cuerpo del documento quedaría como el siguiente:


```
<body>
<!--etiquetas que forman el cuerpo-->
MI PRIMERA PÁGINA EN COMPUTACIÓN APLICADA.
</body>
```

- 5) Por último, terminaremos con la etiqueta de finalización de un documento html, `</html>`.
- 6) Guardaremos el documento con el nombre "COMPUTACIONAPLICADA1" y la extensión .html . "COMPUTACIONAPLICADA1.html"
- 7) Por lo tanto el documento completo será:


```
<html>
<head>
<title>COMPUTACIÓN APLICADA </title>
</head>
<body>
<!--etiquetas que forman el cuerpo-->
MI PRIMER PÁGINA EN COMPUTACIÓN APLICADA.
</body>
</html>
```

La vista Dividir

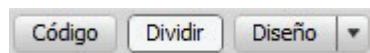


Figura 4.27. Menú Vistas.

La vista Dividir permite dividir la ventana en dos zonas: Código y Diseño, o bien Código y En vivo. La zona derecha muestra el código fuente, y la izquierda el editor visual. Cuando se hace un cambio en alguna de las zonas, este se aplica directamente sobre la otra. Sin embargo, la edición tanto del código como del editor visual es distinta si se utiliza Código y Diseño o Código y En vivo. Dependiendo de la combinación de botones, trabajaremos de modo distinto.

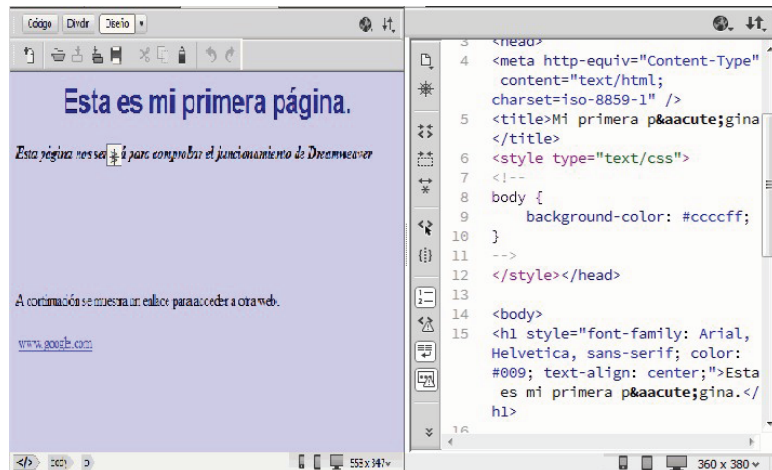


Figura 4.28. Men  dividir.



La Vista en vivo

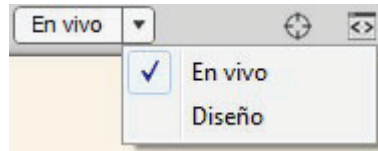


Figura 4.29. Vista en vivo.

La Vista en vivo ofrece una vista prácticamente idéntica al resultado final (al menos para el navegador Chrome, ya que usa el mismo motor de navegación que este).

La Ayuda

A través del menú Ayuda puedes ir a varias opciones, veamos algunas de ellas:

- Si seleccionas la opción Ayuda *online* de Dreamweaver o simplemente pulsando F1 se accede a la ayuda en línea de Adobe Community Help, donde puedes hacer búsquedas por temas o por palabras clave.
- Novedades de Dreamweaver te mostrará una animación con las mejoras más recientes del programa.
- Ver algunos videos (en inglés) de sus nuevas funciones.
- Comprobar si hay actualizaciones en línea disponibles del programa.
- Iniciar sesión con nuestra ID de Adobe.
- Acceder al Centro de soporte de Dreamweaver en la web.
- Acceder al Foros en línea de Adobe.

Además, en la barra de aplicación encontramos un botón que abre nuestro navegador directamente en la página de ayuda en línea.

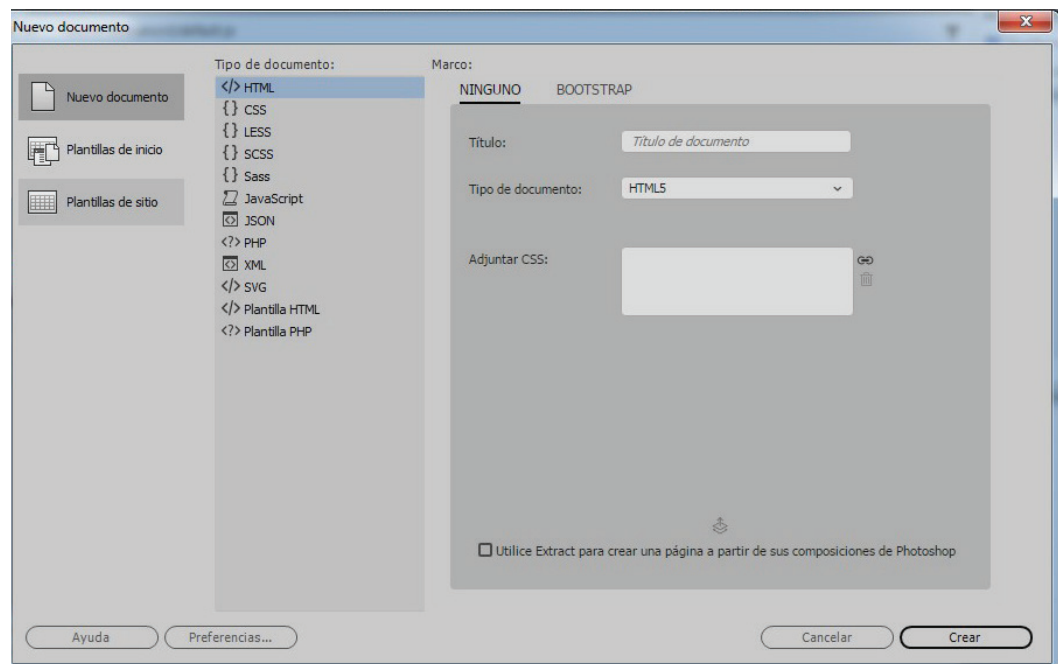


Figura 4.30.
Nuevo documento.

Para crear un documento nuevo en blanco:

1. Seleccione Archivo > Nuevo.
2. Aparecerá la ventana anterior.
3. Haga clic en el botón Crear. El documento nuevo se abrirá en la ventana del documento.
4. Guarde el documento.

Guardar archivos con Dreamweaver

Es necesario guardar un documento después de crearlo. Para guardar un documento nuevo:

1. Seleccione Archivo > Guardar.
2. En el cuadro de diálogo que aparece a continuación ve hasta la carpeta en la que deseas guardar el archivo, donde guardarás todo lo relacionado con el sitio web.
3. En el cuadro de texto Nombre de archivo, introduce un nombre para el archivo. Evita utilizar espacios y caracteres especiales en los nombres de archivos y carpetas. Asimismo, no comiences los nombres de los archivos con números. En concreto, no utilices caracteres especiales (como é, ç o ¥) ni signos de puntuación (como dos puntos, barras inclinadas o puntos) en los nombres de archivos que desees colocar en un servidor remoto; muchos servidores cambian estos caracteres durante la carga, lo que provoca que se rompan los vínculos existentes con los archivos.
4. Haz clic en Guardar.

Apertura de archivos en Dreamweaver

En Dreamweaver se pueden abrir y editar documentos con gran facilidad.

Para abrir un archivo:

1. Selecciona Archivo > Abrir.
2. En el cuadro de diálogo Abrir, selecciona el archivo y haz clic en Abrir.

4.1.2. Elementos de una página web (tablas, texto, botones, imágenes, vínculos, hipervínculos, video, sonido, animaciones, marcos, etcétera)

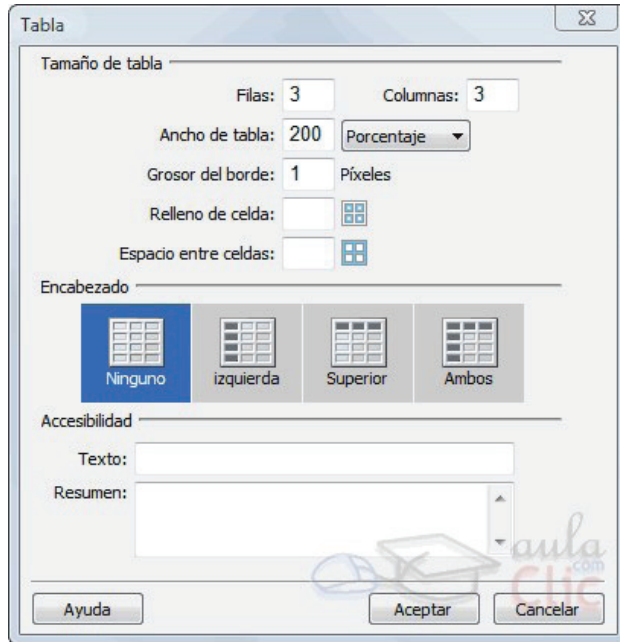
Veremos cómo trabajar con tablas. Podremos, entre otras cosas, insertar tablas, combinar celdas, insertar filas o columnas y cambiar el tamaño del borde.

La finalidad de las tablas es presentar una serie de datos de forma clara y organizada dividiéndolos en filas y columnas.

Tabla

Insertar una tabla

Para insertar una tabla hay que dirigirse al menú Insertar, a la opción Tabla.



En la nueva ventana habrá que especificar el número de Filas y Columnas que tendrá la tabla, así como el Ancho de la misma.

El Ancho puede ser definido como Píxeles o como Porcentaje. La diferencia entre uno y otro es que el ancho en Píxeles es siempre el mismo, independientemente del tamaño de la ventana del navegador en la que se visualice la página. En cambio, el ancho en Porcentaje indica la proporción de la tabla respecto a la página (o del elemento contenedor) y se ajustará al tamaño de la ventana del navegador, esto permite que los usuarios que tengan pantallas grandes, aprovechen todo el ancho de pantalla.

Figura 4.31. Tabla.

El texto

Las características del texto seleccionado pueden ser definidas a través del menú Formato, y a través del inspector de propiedades.



Actividad de aprendizaje 5

En este ejercicio crearemos un documento HTML para iniciarnos con el tamaño de la fuente.

1. La cabecera del documento junto con la etiqueta de apertura sería:


```
<html>
<head>
<title>TAMAÑOS, FORMATOS Y COLORES</title>
</head>
```
2. El cuerpo del documento quedaría como el siguiente:


```
<body style="color: rgb(0,0,0); background-color: rgb(120,120,120);">
<p style=" color: black; font-weight: bold; font-size: 40px;"> tamaño de40 px</p>
<p style=" color: yellow; font-weight: bold;"font-size: small;">#FFFF00
tamaño small</p>
<p style=" color: white; font-weight: bold; "font-size: medium;">#FFFFFF
tamaño medium</p>
<p style=" color: red; font-weight: bold; "font-size: large;">#FF0000
tamaño de large</p>
<p style=" color: lime; font-weight: bold; "font-size: xx-large;">#00FF00
tamaño xx-large</p>
<p style=" color: blue; font-weight: bold;"font-size: xx-small;">#0000FF
tamaño xx-small</p>
```

```

<p style=" color: fuchsia; font-weight: bold; "font-size: 2%;"> tamaño 2%</p>
<p style=" color: orange; font-weight: bold; "font-size: 50%;"> tamaño50%</p>
<p style=" color: skyblue; Font-weight: bold; "font-size: 80%;">tamaño80%</p>
<p style=" color: coral; font-weight: bold; "font-size: 200%;">tamaño200%</p>
<p style=" color: purple; font-weight: bold "font-size: 300%;">tamaño300%</p>
<h2>Pruebas de formatos</h2>
Esto es normal
<sub>y esto es subíndice</sub><br>
Esto es normal
<sup>y esto es superíndice</sup><br>
<br>
</body>
</html>

```

3. Guardar como TAMAÑOSFORMATOSYCOLORES .html

Imágenes

Las imágenes son un aspecto muy importante de la web. Ya sea como complementos de la información o como parte del diseño. Las imágenes hacen una página mucho más atractiva a ojos del visitante. No obstante no conviene abusar, pues aumentarán mucho el tamaño final de la web.

Para insertar una imagen hay que dirigirse al menú Insertar, después a la opción Imagen y dentro de esta, Imagen otra vez.

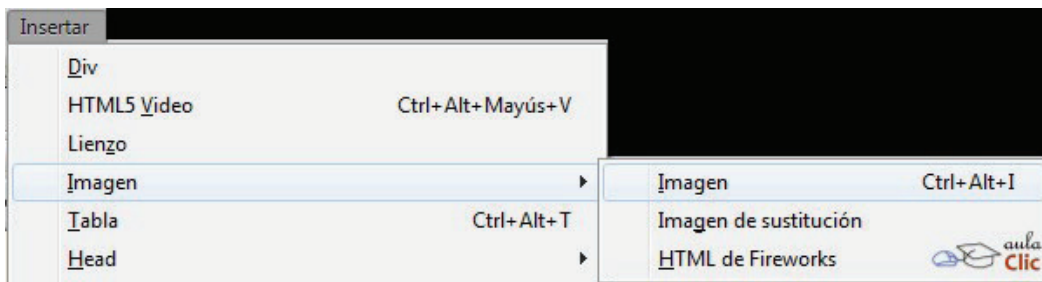


Figura 4.32. Insertar imagen.

Después de esto, ya es posible seleccionar una imagen a través de la nueva ventana. Cuando te acostumbres, te será más cómodo acceder con la combinación de teclas Ctrl + Alt + I.

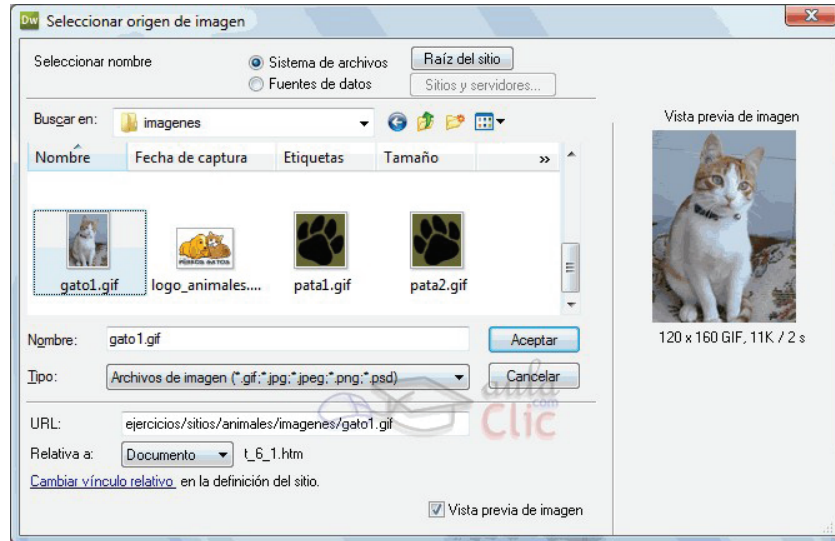


Figura 4.33. Seleccionar origen de la imagen.



Actividad de aprendizaje 6

Insertar una imagen con código html

```
<html>
<head>
<meta charset="utf-8">
<title>PINGUINO</title>
<link rel="shortcut icon" href="Penguins.gif" >
</head>
<body>
<imgsrc="file:///C:/Users/Public/Pictures/SamplePictures/Penguins.jpg"
width="1024" height="768" alt="" />
</body>
</html>
```

Hiperenlaces

Un hiperenlace, hipervínculo o vínculo, no es más que un enlace, que al ser pulsado lleva de una página o archivo a otra página o archivo.

Es posible asignar un vínculo a un texto, a una imagen, o a parte de una imagen.

Cuando creamos un enlace, lo que realmente hacemos es crear una etiqueta `<a></a>` que es la que en HTML se encarga de definir los enlaces. Puedes observar esto en la barra de estado:

```
<body> <p> <a>
```

Figura 4.34. Hiperenlaces.

Actividad de aprendizaje 7



En esta actividad crearemos una tabla de 9 filas x 3 columnas. En las filas pares insertaremos en cada celda una imagen pequeña de un tipo de flor y en las filas impares pondremos el nombre de cada una. Una vez hecho esto crearemos un enlace en cada imagen para que al pulsar, por ejemplo, en la ROSA se abra una página que hable de ROSAS.

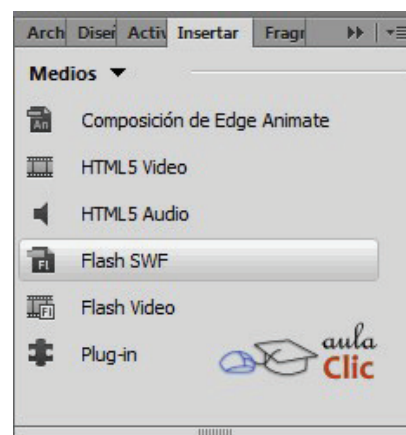
Multimedia

Veremos cómo insertar una animación Flash, un sonido y un video que son los más empleados.

Las películas Flash son animaciones, cuyo archivo tiene la extensión SWF. Es frecuente verlas en las páginas iniciales de los sitios web, a modo de presentación para los usuarios, como *banners* publicitarios, como botones... Hasta hace pocos años, Flash era uno de los elementos multimedia más empleados en las páginas web, aunque su uso fue decreciendo con la llegada de HTML 5 y el video en mp4. Es importante añadir que el formato de las películas Flash no es soportado por un buen número de dispositivos móviles, como tabletas y *smartphones*, por lo que no debe ser utilizado en páginas web dirigidas a estos dispositivos.

Las películas Flash pueden insertarse en una página a través del menú Insertar → Medios, opción Flash SWF, o pulsando Ctrl + Alt + F.

También pueden insertarse empleando el panel Insertar en la categoría Medios, pulsando sobre la opción  Flash SWF.



Veamos las opciones más importantes que nos ofrece el inspector de propiedades cuando tenemos seleccionado el archivo Flash:

Figura 4.35.
Insertar Flash.

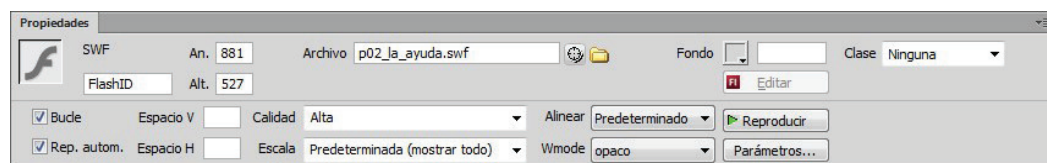


Figura 4.36. Propiedades de Flash.

Audio

No es muy habitual incluir sonido en una página web, pues algunos usuarios suelen escuchar su propia música cuando navegan en internet, por lo que tener también sonido en cada página que se visita puede resultar molesto.

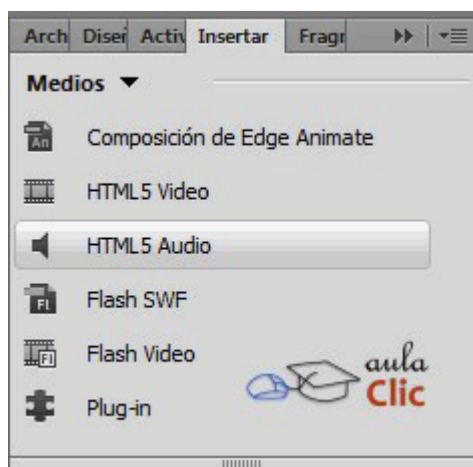


A pesar de ello, incluir un sonido agradable, apropiado al contenido de la página, puede hacerla más atractiva. Las páginas que contienen sonido suelen ofrecer la posibilidad de activarlo o desactivarlo, para aquellos usuarios que no deseen escuchar el sonido de la página. Toma en cuenta que siempre es mejor tener esa opción a que el usuario abandone la página.

Los formatos de sonido más habituales en internet son, fundamentalmente, el MP3 y el WAV, aunque existen otros formatos diferentes que también pueden utilizarse (MIDI).

Lo ideal es incluir algún archivo de audio que no ocupe mucho espacio, y no por ello significa que sea de mala calidad.

Se puede insertar un archivo de audio en un documento de dos formas, con la nueva etiqueta `<audio>` o mediante un *plug-in*.



1. Nueva etiqueta `<audio>`

La etiqueta `<audio>` solo funcionará en los navegadores compatibles con HTML 5, IE a partir de la versión 9 y las versiones de Firefox 3.5, Chrome 3.0 y Safari 3.0, además de algunos otros. Afortunadamente, ya son la mayoría.

Puedes insertar un audio con el menú Insertar → Medios opción HTML 5 Audio. En el panel Insertar, categoría Medios, encontrarás la misma opción.

Figura 4.37. Insertar audio.

La forma de la etiqueta es la siguiente:

```
<audio> <source src="archivo sonido" type="MIME-type"/> </audio>
```

Plug-in

Esta segunda forma de reproducir sonido se utiliza cada vez menos. Para insertar un archivo de audio mediante *plug-in* tienes que dirigirte al menú Insertar, Medios, opción Plug-in. También es posible usar el panel Insertar, categoría Medios.

Pueden ocurrir dos cosas:

Si tu navegador tiene instalado el *plug-in* necesario para reproducir el archivo, se mostrará un control similar a este (imagen de IE8):



Figura 4.38. Plug-in.

En cambio, si el *plug-in* necesario no está instalado, el navegador ofrecerá la posibilidad de hacerlo, como en esta imagen de Firefox.

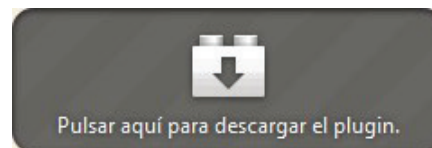


Figura 4.39.
Descargar
Plug-in.

En Dreamweaver no se mostrarán los controles de audio. Todos los archivos que son insertados como *plug-in* aparecen representados dentro de Dreamweaver por una imagen como la de la izquierda.

Los sonidos se reproducen automáticamente al cargarse la página, y se reproducen solo una vez. Estos valores no pueden definirse como propiedades a través del inspector de propiedades, pero si los conocemos podemos emplear el botón Parámetros...

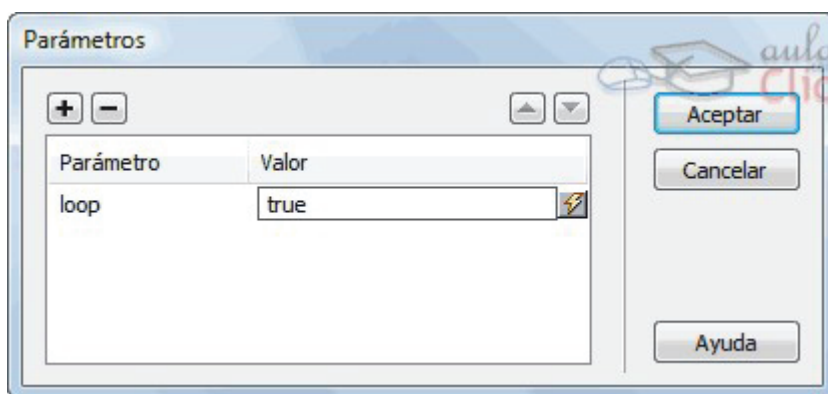


Figura 4.40. Parámetros.

Por ejemplo, para que el archivo se reproduzca continuamente le hemos añadido `loop="true"`.

Si no queremos que se reproduzca automáticamente, podemos añadir también `autostart="false"`

La línea de código del archivo de audio nos quedaría del siguiente modo:

```
<embed src="media/audio.mid" autostart="false" loop="true"></embed>
```

Si quieres poner música de fondo en una página web sin que aparezcan los controles de audio, puedes escribir este código directamente en la vista Código.

```
<bgsound src="cancion1.wav" loop="-1">
```

Insértalo después de la etiqueta `</title>`. Con el parámetro `loop` puedes decidir cuántas veces quieres que se reproduzca, 1, 2, 3... (con -1 se reproduce de modo continuo).



No obstante, insistimos en que no es recomendable poner música sin controles de reproducción.

Videos

En ocasiones puede ser interesante incluir algún video en una página web, pero hay que tener en cuenta que los videos suelen ocupar mucho espacio en disco, por lo tanto, precisan de mucho tiempo para descargarse.

En este apartado nos referimos a insertar video de un archivo alojado en nuestro sitio. No a mostrar videos de sitios especializados, como YouTube.

Para insertar un archivo de video en un documento puedes hacerlo de dos formas, con la nueva etiqueta `<video>` o mediante un *plug-in*.

1. Video con HTML 5

Para insertar un archivo de video en la página podemos usar el menú Insertar → Medios opción HTML5 Video. También es posible utilizar el panel Insertar, categoría Medios.

El formato básico de la etiqueta `<video>` es el siguiente.

```
<video>
<source src="archivo vídeo" type="MIME-type"/>
</video>
```

La instrucción típica para reproducir video sería, por ejemplo:

```
<video width="160" height="120" controls>
<source src="media/cotorra.mp4" type="video/mp4"/>
<source src="media/cotorra.ogv" type="video/ogv"/>
Este navegador no puede reproducir este archivo de vídeo. Actualice el navegador.
</video>
```

Es muy similar a la que hemos visto para sonido, salvo que aparecen los atributos de tamaño del video (`width`, `height`), que también pueden modificarse en el Inspector de propiedades cuando el elemento está seleccionado.

La etiqueta `<video>` tiene varios atributos opcionales, entre los que podemos citar los siguientes:

- `autoplay`. Hace que el video comience a reproducirse al cargarse la página.
- `loop`. Para que se reproduzca una y otra vez, sin parar.
- `preload`. Para que se cargue al hacerlo la página especificaremos `preload = "auto"`, para que se cargue al pulsar *play*, `preload = "none"`.

Creación y almacenamiento de una página nueva

Después de crear un sitio nuevo ya puedes empezar a crear páginas web. Comienza creando una página nueva y guardándola en la carpeta raíz local.

1. En Dreamweaver, Selecciona Archivo > Nuevo
2. En la ficha Sitio > Nuevo Sitio
3. En la Pestaña Básicas nos guiará en la configuración de la nueva carpeta.
4. En la Pestaña Avanzadas no tenemos asistente y aparte de la carpeta local se pueden configurar las carpetas: remota y la de prueba.
5. Posteriormente a la definición de nuestro sitio, aparecen en el Panel de Archivos los archivos de los que consta nuestro sitio (ahora vacío). Desde aquí podemos administrar, borrar, copiar, pegar, mover y abrir los archivos que componen el sitio como si estuviéramos en el explorador de Windows.

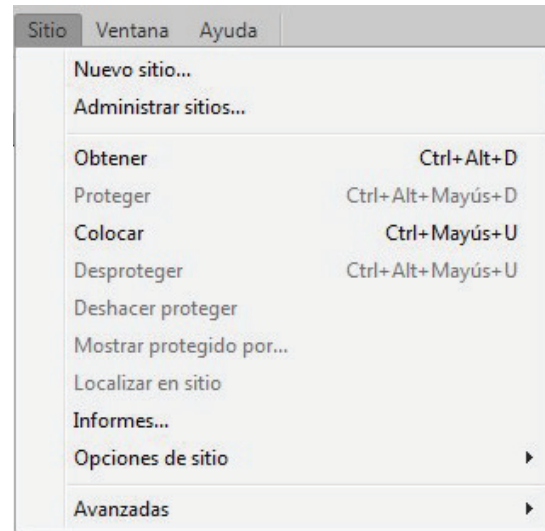


Figura 4.41. Nuevo sitio.

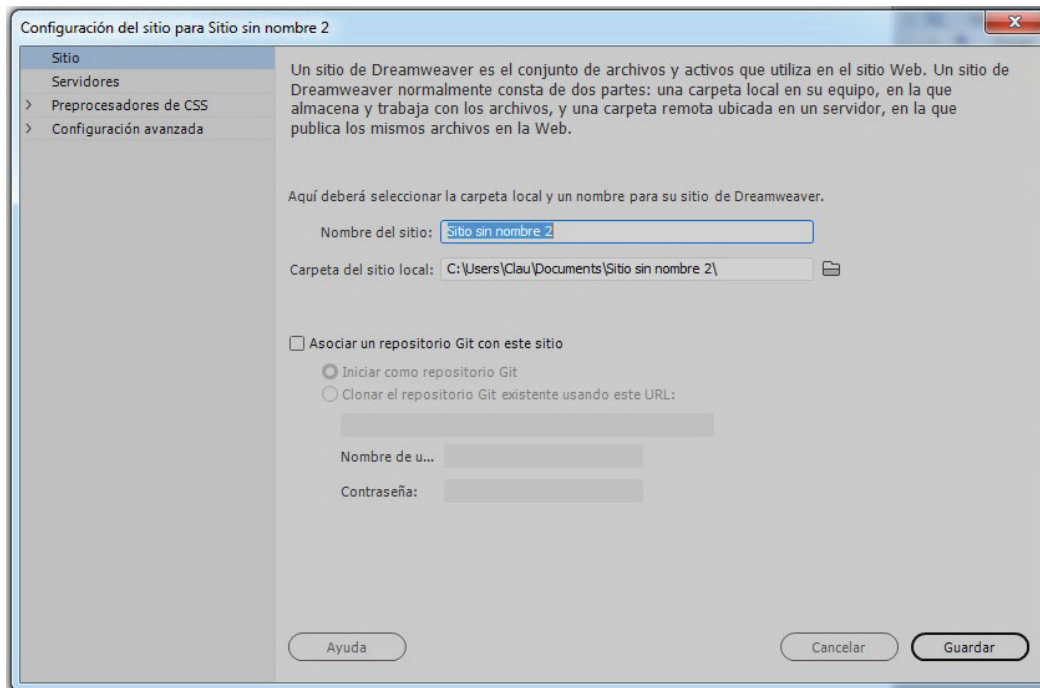


Figura 4.42. Configuración del sitio.

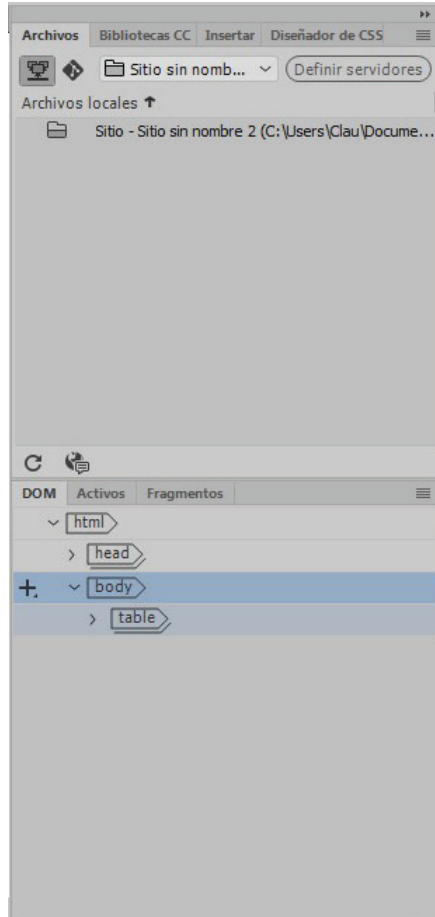


Figura 4.43. Menú Sitio.

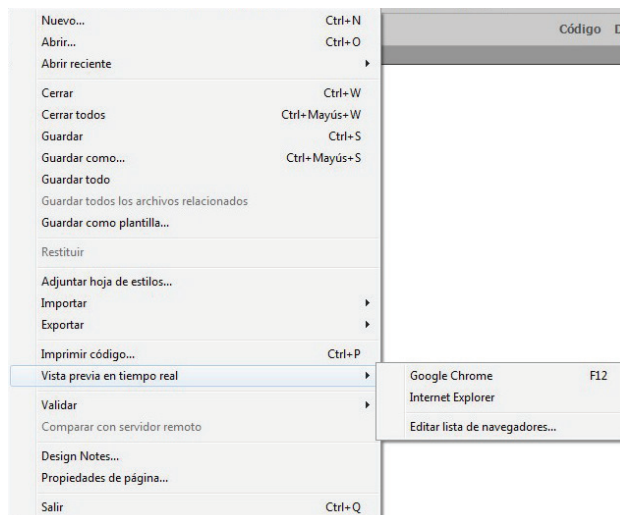
Dreamweaver permite crear aplicaciones dinámicas basadas en bases de datos como MySQL, SQL, Oracle, Informix.

4.1.3. Publicación del sitio

Es necesario asegurarse que una vez creado un sitio WEB se compruebe localmente. Como ya se mencionó, es necesario asegurarse de que las páginas tengan la apariencia y el funcionamiento esperado en los navegadores de destino, que no haya vínculos rotos y que las páginas no tarden demasiado en cargarse. También puedes comprobar todo el sitio y solucionar los problemas antes de publicarlo.

Haz una vista previa a las páginas en todos los navegadores y plataformas posibles. De este modo podrás observar las diferencias que no se puedan ver comprobando el navegador de destino. Comprueba si hay vínculos rotos y repáralos. Otros sitios experimentan cambios de diseño y organización, por lo que es posible que las páginas con las que tiene establecidos vínculos se hayan movido o borrado. Para obtener una lista de vínculos a sitios externos haz una comprobación de vínculos.

Vista previa en navegadores



Se usa para acceder a la vista previa de los documentos en los navegadores de destino en cualquier momento. Para ello no es necesario guardar previamente el documento.

El contenido vinculado no aparece cuando se hace una vista previa de documentos en el navegador local. Esto se debe a que los navegadores, a diferencia de los servidores, no reconocen la raíz de los sitios.

Para obtener la vista previa del documento en un navegador elije:

Archivo > vista previa en tiempo real

Y selecciona uno de los navegadores que aparece en la lista.

Figura 4.44. Vista previa en tiempo real.

Si aún no has seleccionado un navegador, elije: ??????

4.1.4. Mantenimiento

Hoy en día casi todas las empresas promocionan sus productos o servicios mediante la web.

Esto les brinda una relación más rápida y directa con clientes potenciales. Además, los sitios web brindan al usuario la facilidad de acceso a la información de la empresa sin la necesidad de tener que contactar directamente con la misma.

Por eso el desarrollo de sitios web implica dedicación, conocimiento y esfuerzo, incluyendo la tecnología adecuada para el buen manejo de los datos y, por supuesto, la seguridad en la transmisión de los mismos. Sin embargo, cuando el desarrollador cree que ha terminado, comienza la etapa que mayor esfuerzo y tiempo consumirá: la de mantenimiento.

Haber desarrollado un sitio web y haberlo subido a internet no basta, puesto que también se debe considerar como se dará a conocer, cada cuanto tiempo se actualizará el contenido, verificar su buen funcionamiento, o si se actualizará el diseño del sitio a uno más llamativo y/o moderno. Recordemos que “mantener un sitio fresco alienta a la gente a regresar”.

El mantenimiento por lo general implica el continuo control del buen funcionamiento sobre los archivos y estructuras de directorios de nuestro sitio web.

El mantenimiento de un sitio pequeño puede necesitar de dos o tres horas por semana. En un sitio grande, el mantenimiento puede ser un trabajo de tiempo completo.

Asegúrate de incluir los costos de mantenimiento en tu presupuesto durante la fase de planeación, para que no los tome por sorpresa este costo. Si estás planeando un sitio grande y ambicioso o gradualmente quieres agregar más contenido y complejidad, trabajar con un programador o diseñador experto desde el principio puede ahorrar mucho tiempo y problemas en lo posterior.

Empezar con un sitio bien diseñado es la manera más efectiva de evitar actualizaciones y mantenimientos costosos. Los desarrolladores expertos de la red cobran un promedio de 75 dólares por hora. Puede parecer mucho, pero el costo lo vale si se quiere incluir funciones avanzadas, como animaciones en línea, formatos o herramientas de búsqueda.

Hay una regla no escrita en la red que establece armar un presupuesto similar al usado para la creación del sitio, y dedicarlo al mantenimiento anual del mismo. Por ejemplo, si necesitamos 100 dólares para el diseño inicial del sitio web, dejaremos otros 100 para usarlos durante el resto del año en mantenimiento y alojamiento del sitio.

Lugares prioritarios de mantenimiento

El mantenimiento de un sitio web es un proceso constante que comienza después de su lanzamiento y termina cuando este es dado de baja. Si ponemos en marcha



el sitio, la siguiente pregunta que debemos formularnos es: ¿qué partes de mi sitio requieren mantenimiento continuo?

a. Mantenimiento de Enlaces

Como los documentos HTML y sus componentes gráficos relacionados están enlazados de una manera específica, cualquier cambio o adición que se haga a los documentos o directorios existentes puede afectar la relación entre ellos. El resultado más común es que los enlaces se rompan, las imágenes se mezclen, o las páginas no se carguen adecuadamente. La retroalimentación de los usuarios, por lo general, se da a través de correo electrónico. Esto puede ser una parte importante para identificar este tipo de problemas, con el fin de ser resueltos de forma oportuna.

b. Mantenimiento del Contenido

Actualizar el sitio involucra cambiar el contenido. Esto puede ser tan fácil como controlar los enlaces a otros sitios para asegurarse de que sean válidos, o tan complejo como agregar nuevas funciones a los formatos. Los recursos y los costos para mantener un sitio moderno y en buen estado dependen del tamaño y la complejidad del sitio y de qué tan seguido tiene que ser actualizado.

Recuerda que agregar nuevo contenido no significa necesariamente eliminar el viejo. Algún material viejo como comunicados de prensa, actualizaciones de programas, artículos y transcripciones de discursos pueden ser útiles a los usuarios y deberían ser archivados. Asegúrate de que la información archivada sea de fácil acceso.

Otra manera simple de informar a los usuarios de que se ha actualizado el sitio, o algunas páginas, es agregar una anotación.

c. Mantenimiento del Software

Es importante revisar los sistemas y aplicaciones basados en la web periódicamente con respecto al manejo de información, riesgos potenciales a la seguridad, la actuación del sistema, debilidades, y modelos de uso, y tomar las medidas convenientes para arreglarlo.

El protocolo HTTP, utilizado en la web para la transferencia de datos a través de la red no es lo suficientemente seguro como para transmitir datos delicados o de índole personal, como pueden ser los números de documentos o de tarjeta de crédito de un particular o determinados archivos comerciales de una empresa.

Por este motivo se han desarrollado un par de protocolos nuevos que vienen a suplir esta falencia.

d. Mantenimiento del Diseño

Una empresa debe mantener una imagen ante el público. Diseños frescos y modernos para mostrar que estamos a la vanguardia y en constante búsqueda

de lo actual. Un diseño que no muestre lo que es la empresa o esté compuesto de elementos poco atractivos o antiguos podría mostrar a la empresa como anticuada o intrascendente. Generalmente una imagen empresarial debe hacer un cambio cada dos años, dependiendo del rubro al que se dedique. Es lógico imaginar que el sitio web debe contener elementos de la imagen corporativa de la empresa, como el logotipo, colores corporativos, etcétera. Sin embargo, el diseño en sí puede cambiarse por otro más innovador o llamativo.

“Primero, todo entra por los ojos” y por mucho que nuestra página contenga información bien estructurada, enlaces perfectos y seguridad en los datos de alta calidad, si no tiene un diseño estético, equilibrado y de fácil entendimiento los usuarios simplemente pasarán de largo por no entender o no encontrar el toque que llame su atención.

Es necesario lograr que el sitio web obtenga consistencia y una imagen distintiva, mediante el uso de fondos, gráficos y logos, colores uniformes en el texto y los enlaces, barras de navegación en el mismo lugar, etc.; de la misma manera que una tienda comercial mantiene una estética única, tanto para su casa central como para sus sucursales.

¿Cómo podemos idear una estrategia efectiva de mantenimiento?

Comienza por decidir con qué frecuencia necesitas o deseas actualizar tu sitio y qué tan extensas serán esas actualizaciones. Por ejemplo, un editor de noticias muy probablemente querrá actualizar la información a diario, en algunos casos incluso hasta por hora. Un comerciante minorista querrá actualizar su sitio en cualquier momento que haya mercancía nueva o descontinuada, u ofertas especiales.

En seguida, pon atención a lo que tus usuarios están haciendo y diciendo. ¿Cuánta gente está visitando tu sitio y a dónde van? Existen maneras para rastrear qué partes de tu sitio están siendo frecuentadas. Si tu sitio está hospedado por un proveedor de servicios de internet, este debe proporcionarte un reporte detallado por periodos de tiempo. Hay un número de programas de software disponible de forma comercial o compartida para que la gente los ejecute en sus propios servidores. Muchos le dan información en tiempo real como quiénes están en su sitio web, de dónde están viniendo, las páginas a las que están accediendo y mucho más.

Ofrece una manera de que los usuarios proporcionen retroalimentación. El método más común es vía correo electrónico. Utilice esa información para identificar y resolver los problemas técnicos de manera oportuna. Usa comentarios cualitativos acerca de tu sitio junto con la información de rastreo de uso para orientar las decisiones sobre qué contenido mantener, quitar o mejorar.

Si estás presionado en tiempo o en recursos, mantener una base de datos simple de todas tus páginas, incluyendo una breve descripción del contenido de cada página, y los enlaces y gráficos relacionados puede ser de mucha ayuda. Al momento en que tu sitio crezca, o si delegas el mantenimiento a alguien más, la base de datos será útil. Existe software disponible para usar este tipo de bases de datos para automatizar muchas de las actualizaciones.



Actividad de aprendizaje 8

Crea el sitio web de tu proyecto integrador, con todos los conocimientos adquiridos, siguiendo las instrucciones de tu maestro.

Resumen

En este módulo vimos cómo iniciar una nueva página web partiendo desde una página en blanco. Desde la configuración hasta los primeros pasos en el diseño.

Dreamweaver CC se ha rediseñado con una interfaz moderna y un motor de codificación flexible y rápido para ofrecer a los diseñadores web y a los desarrolladores de aplicaciones o interfaz de usuarios formas, más sencillas de crear, codificar y gestionar sitios web con una apariencia espectacular en cualquier pantalla que está destinada a la construcción y edición de sitios y aplicaciones web basada en estándares.

Adobe Dreamweaver es una aplicación de estudio que está destinada a la construcción y edición de sitios web basados en estándares.

La gran ventaja de este editor sobre otros es su gran poder de ampliación y personalización del mismo, puesto que en este programa, sus rutinas están hechas en JavaScript-C. Lo que ofrece una gran flexibilidad en estas materias, esto hace que los archivos del programa no sean instrucciones de C++ sino, rutinas de JavaScript que hacen un programa muy fluido, lo cual genera, que programadores y editores web hagan extensiones para su programa y lo pongan a su gusto.

Autoevaluación

¿Considero que lo hice bien?

Éste es un proceso de reflexión sobre lo que obtuviste después de diseñar y realizar la demostración práctica. Responde la tabla siguiente, colocando una X en el cuadro correspondiente a cada ítem (5 representa "estoy totalmente de acuerdo" y 1 "estoy totalmente en desacuerdo").

	5	4	3	2	1
1. Asumí una actitud propositiva, entusiasta, responsable.					
2. El aprendizaje y las experiencias que me dejó la participación en la actitud me son útiles.					
3. Pude explicar con claridad los conceptos y definiciones que revisamos y aplicamos a lo largo de cada actividad.					
4. Puedo aplicar lo aprendido en mi vida diaria.					
Mi calificación general sería de: (anota un valor numérico en escala del 1 a 10)					

Coevaluación

Puedes compartir la autoevaluación que hiciste con tus compañeros con la intención de intercambiar ideas o experiencias de aprendizaje. En caso de que hayas tenido alguna dificultad sería una buena aportación compartirla.

La calificación de los compañeros con quienes intercambié mi autoevaluación fue (anotar los valores numéricos que tenga asignados).

Ítem	Compañero 1	Compañero 2	Compañero 3
1			
2			
3			
4			

Fuentes

Bibliografía

- Cairó, O. (2005). *Metodología de la programación*. México: Alfaomega.
- Cairó, O. (2006). *Fundamentos de programación (Piensa en C)*. México: Pearson Prentice Hall.
- Joyanes Aguilar, L. (2008). *Fundamentos de programación*. España: Mc Graw Hill.
- Pólya, G. (1965). *Cómo plantear y resolver problemas*. (J. Zagazagoitia, Trad.) Trillas.
- Real Academia Española. (2017). *Proyecto*. En Diccionario de la lengua española. Madrid.

Referencias electrónicas

- Ecolohosting. (6 de Octubre de 2017). ecolohosting.com. Obtenido de <https://ecolohosting.com/>
- Google. (8 de Octubre de 2017). Google.com. Obtenido de <https://support.google.com/a/answer/2573637?hl=es#B>
- Hiard, V. (25 de 09 de 2017). Google Books. Barcelona: eni. Obtenido de https://books.google.com.mx/books?id=_lAzbehV16cC&pg=PA218&dq=dise%C3%B1o+de+proyectos+web&hl=es-419&sa=X&redir_esc=y#v=onepage&q&f=false
- <http://www.aulacli.es/dreamweaver-cc.htm>
- <http://www.todo-dreamweaver.com>
- <http://www.programatium.com/dreamweaver.html>
- http://www.svetlian.com/Webmaster/dream_tutor.html

Directorio

Doctor en Ciencias e Ingeniería Ambientales

Carlos Eduardo Barrera Díaz

Rector

Doctor en Ciencias Computacionales

José Raymundo Marcial Romero

Secretario de Docencia

Doctora en Ciencias Sociales

Patricia Zarza Delgado

Secretaria de Investigación y Estudios Avanzados

Doctor en Ciencias de la Educación

Marco Aurelio Cienfuegos Terrón

Secretario de Rectoría

Doctora en Humanidades

María de las Mercedes Portilla Luja

Secretaria de Difusión Cultural

Doctor en Ciencias del Agua

Francisco Zepeda Mondragón

Secretario de Extensión y Vinculación

Doctor en Educación

Octavio Crisóforo Bernal Ramos

Secretario de Finanzas

Doctora en Ciencias Económico Administrativas

Eréndira Fierro Moreno

Secretaria de Administración

Doctora en Ciencias Administrativas

María Esther Aurora Contreras Lara Vega

Secretaria de Planeación y Desarrollo Institucional

Doctora en Derecho

Luz María Consuelo Jaimes Legorreta

Abogada General

Doctor en Ciencias Sociales

Luis Raúl Ortiz Ramírez

Secretario Técnico de la Rectoría

Licenciada en Comunicación

Ginarely Valencia Alcántara

Directora General de Comunicación Universitaria

Doctor en Ciencias Sociales

Luis Raúl Ortiz Ramírez

**Director de Centros Universitarios y Unidades Académicas Profesionales
Región A y Encargado del Despacho Región B**

Doctora en Investigación Educativa

Eva Lilia García Escobar

Directora de Estudios de Nivel Medio Superior

Planteles dependientes de la Escuela Preparatoria

Doctor en Tecnología e Innovación de la Educación
Arturo Mejía Zamora

Plantel "Lic. Adolfo López Mateos"

Maestro en Ciencias de la Educación
Erick Vieyra Méndez

Plantel "Nezahualcóyotl"

Maestro en Administración Pública y Gobierno
Raúl Juárez Toledo

Plantel "Cuauhtémoc"

Doctora en Ciencias de la Educación
María del Rocío García de León Pastrana

Plantel "Ignacio Ramírez Calzada"

Doctor en Ciencias de la Educación
Alejandro Morales Velázquez

Plantel "Dr. Ángel Ma. Garibay Kintana"

Maestro en Educación con Orientación en Innovación y Tecnología Educativa
Sergio Luis Roldán Vásquez

Plantel "Texcoco"

Maestro en Docencia
Erick Rodolfo Sánchez Cortés

Plantel "Sor Juana Inés de la Cruz"

Química Farmacéutica Bióloga
Janette Jaimes García

**Encargada del Despacho de la Dirección del Plantel
"Dr. Pablo González Casanova"**

Doctora en Educación
María del Rosario Sotelo Hurtado

Encargada del Despacho de la Dirección del Plantel "Isidro Fabela Alfaro"

Licenciada en Comunicación
Ana Vianey Suárez Castro

**Encargada del Despacho de la Dirección del Plantel
"Mtro. José Ignacio Pichardo Pagaza"**