

Recomendaciones adicionales a los ejercicios propuestos

Capítulo 1

En este anexo se citan textualmente algunos de los ejercicios propuestos y se brindan diversas sugerencias que complementan las expresadas a lo largo del texto. En algunos casos estará la resolución; en otros, se mencionarán aspectos más concretos con la finalidad de que el estudiante tenga mayores indicaciones que faciliten su adecuada resolución.

Tema 1.1. Resolución de ejercicios pares.

Programa #2. Realizar un programa, con su correspondiente tabla de pruebas y diagrama de flujo, que a partir del valor de los tres lados de un triángulo nos indique su área y el tipo de triángulo (escaleno, equilátero o isósceles).

Recomendaciones y respuesta:

Además del código, observe que se están colocando instrucciones para el despliegue adecuado de los acentos y la validación de que el triángulo exista. En el caso de las condicionales anidadas, se empleó una llave redundante con la finalidad de percibir mejor el alcance de cada bloque. Este tipo de situaciones se suelen dejar de lado en los cursos básicos de programación.

```
/* Determinación del tipo de triángulo */
```

```

#include <stdio.h>
#include <math.h>
int main() {
    float a, b, c, s, area;
    printf("Bienvenido.\n\n");
    printf("Este programa te ayudará a obtener el tipo de ");
    printf("triángulo (escaleno, isósceles o equilátero) ");
    printf("a partir de sus tres lados.\n");
    printf("Teclea los tres lados separados por un espacio: ");
    scanf("%f %f %f", &a, &b, &c);

    /* previamente se validará que sea posible construir el triángulo */
    if ( (a >= b + c) || (b >= a + c) || (c >= a + b))
        printf("No puede formarse un triángulo de esas dimensiones.\n");
    else {

        /* determinar el tipo de triángulo */
        printf ("Se trata de un triángulo ");
        if (a == b && b == c)
            printf("equilátero.");
        else {
            if (a == b || b == c || a == c)
                printf("isósceles.");
            else
                printf("escaleno.");
        }
    }
    return 0;
}

```

Programa #4. Realizar un programa que entregue la siguiente serie hasta el número 100: 1, 2, 3, 4, 5, 6, aplauso, 8, 9, 10, 11, 12, 13, aplauso, 15 ... En este caso, acompañarlo con su respectivo diagrama de flujo. (Observe que se están sustituyendo los múltiplos de siete por la palabra aplauso).

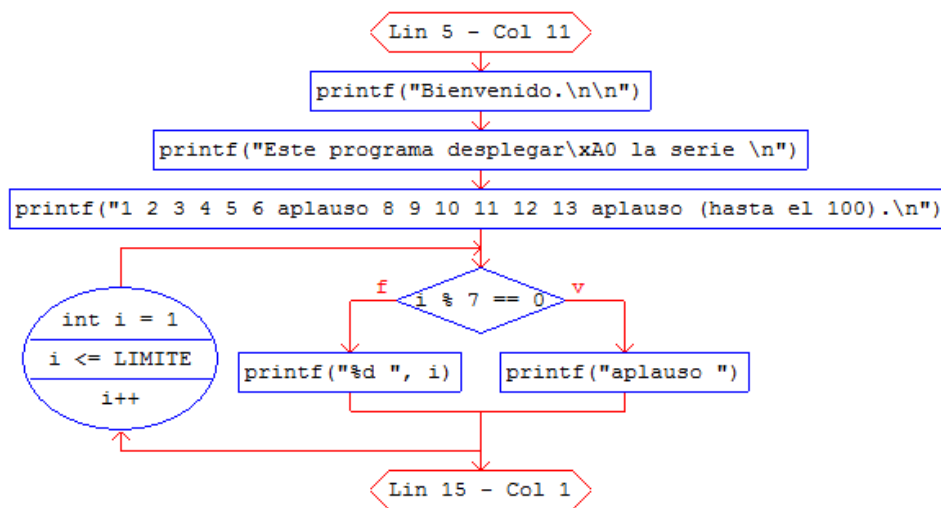
Recomendaciones y respuesta:

En este caso destacaremos la definición de la constante. En términos generales, si un valor aparece en más de un lugar en el programa, conviene colocarlo en una constante a fin de facilitar el mantenimiento. Además, se reproduce el diagrama de flujo que genera automáticamente el compilador Zinjai. Aproveche este tipo de opciones que brinda cada uno de los IDE, que día con día se hacen potentes y reducen en mayor porcentaje el tiempo de programación.

```

/* Programa para producir serie aplauso */
#include <stdio.h>
#include <math.h>
#define LIMITE 100
int main() {
    printf("Bienvenido.\n\n");
    printf("Este programa desplegar\xA0 la serie \n");
    printf("1 2 3 4 5 6 aplauso 8 9 10 11 12 13 aplauso (hasta el 100).\n");
    for (int i = 1; i <= LIMITE; i++) {
        if (i % 7 == 0)
            printf("aplauso ");
        else
            printf("%d ", i);
    }
}

```



Programa #6. Intente determinar qué desplegará el programa 1-1 sin ejecutarlo y posteriormente corrobore su respuesta con la corrida. A continuación cambie el parámetro que recibe la subrutina de 5 a 100,000. ¿Cuál es el resultado?

Recomendaciones y respuesta:

Hacer una prueba de escritorio parece trivial. No lo es. La gran mayoría de estudiantes reprueba un examen basado en pruebas de escritorio la primera vez que lo hacen, incluso aquéllos que saben codificar. Saber hacer pruebas de escritorio en programación estructurada ayuda para la interpretación de diagramas de sistemas WEB.

```

/* Ejercicio para prueba de escritorio */
#include <stdio.h>
double calculo (int x);

double calculo(int x) {
    double suma;
    int i;
    suma=0.0;
    for (i=0; i<=x; i++) {
        if ((i==2)||((i%2)==0)) {
            suma=suma+(1.0/(2*i+1));
        }
        else suma=suma-(1.0/(2*i+1));
    }
    return 4*suma;
}

int main(void) {
    int x;
    double aux;
    aux=calculo(5);
    printf("El resultado es %6.4f", aux);
    return 0;
}

```

Respuesta: El resultado es 2.9760 tal y como está. Al cambiar el parámetro a 100000 la salida resulta 3.1416.

Las secciones 1.2, 1.3 y 1.4 sugirieron los siguientes ejercicios:

Tema 1.2. Tomar al menos dos sitios WEB que se usen mucho, ya sea para PC o celular, y tratar de contestar las siguientes preguntas: ¿cuál es la función principal que realizan?; ¿qué tareas pueden realizar?; ¿algunas están de sobra?; ¿hay funcionalidades que dejan fuera y deberían tener? También es muy ilustrativo aplicar los mismos cuestionamientos a un par de aplicaciones que no usamos porque finalmente no nos convencieron.

Tema 1.3. En esta sección se describirán de manera general las características de calidad del software, ejemplificándolas con sitios WEB recomendables para estudiantes de informática. Es necesario ir navegando por cada uno de ellos para formar una opinión propia, además de dar seguimiento durante un par de semanas a alguno de los dos más interactivos para percibir su

verdadera potencialidad: studio.code.org para el aprendizaje de la lógica de programación o duolingo, para el caso de idiomas.

Tema 1.4. En esta sección se irán mencionando varios sitios WEB. Hay que explorarlos, en este caso haciendo hincapié en lo innovador de sus ideas. En el caso de megasinapsis, sugerimos darse de alta en su Facebook y dar seguimiento por un par de semanas; para Google Art Project localizar el calendario azteca, que también fue digitalizado a la más alta resolución.

Recomendaciones:

Las características a evaluar podrían ser: corrección y funcionalidad, robustez, facilidad de uso e imagen atractiva, portabilidad y compatibilidad, seguridad, facilidad de mantenimiento desde la visión del usuario, y, finalmente, oportunidad y economía.

La escala de calificación podría ser de 0 a 3 (o algún límite similar), donde 3 es la calificación máxima si se considera que la característica de calidad está cubierta de manera óptima. Si no es así, se asignará una menor puntuación, hasta llegar a cero si se juzga que la cobertura en ese punto es nula.

Característica a evaluar	Sitio recomendable (incluyendo Duolingo, Studio Code y/o Megasinapsis)	Sitio no recomendable

Capítulo 2

Citamos una de las posibles respuestas correctas a las preguntas pares del final del capítulo, obtenidas textualmente (o casi textualmente) del texto a lo largo del capítulo.

Respuestas a preguntas pares:

2.- ¿Qué es una dirección IP y qué relación tiene con el nombre de un dominio?

Los Servidores en Internet son identificados por una dirección IP única conformada por una agrupación de cuatro números de tres dígitos en un rango de 0 a 255. La dirección IP identifica de manera lógica y jerárquica una computadora o algún dispositivo conectado a la red.

Se le conoce como dominio al nombre con el que se identifica un sitio Web. Es un nombre que puede estar compuesto por caracteres tanto alfabéticos y/o numéricos (alfanumérico) y se asocia a una dirección física de una computadora o algún otro dispositivo que se conecta a Internet. A esta dirección se le conoce como dirección IP (Internet Protocol o Protocolo de Internet).

En internet podemos encontrar una infinidad de dominios, que se basan en direcciones IP, por lo que se requiere de un Servidor de Nombres de Dominio (DNS) que facilite la traducción de los nombres de dominios a direcciones IP, es decir, el DNS traduce un nombre de dominio en una dirección IP o traduce una dirección IP en un nombre de dominio.

Existen IP de tipo públicas sobre las cuales no tenemos control, ya que por lo general son asignadas por los proveedores de servicios de internet (ISP). Estas IP pueden ser estáticas (se nos asigna una IP fija) o dinámicas (IP disponible al momento de la conexión: al desconectarnos y volvernos a conectar esta cambia). También existen las IP privadas. Podemos tener control sobre ellas y podemos asignarlas a algún equipo de cómputo u otros

dispositivos; también se puede lograr la asignación de IP privadas de forma automática o dinámica mediante un protocolo de configuración de host dinámico (DHCP, Dynamic Host Configuration Protocol), lo que facilita la administración de la red.

4.- ¿Cuál es la diferencia entre un Navegador (browser) y un buscador de internet?

Para poder visualizar las páginas Web requerimos de un navegador [como Chrome o Internet Explorer], aplicación que normalmente está instalada o reside en la computadora del usuario o cliente, o en cualquiera de sus dispositivos móviles.

Los buscadores [como el de Google] son programas desarrollados con potentes algoritmos de rastreo e indexación para la búsqueda de información que integran elementos de Inteligencia Artificial y tecnología semántica.

6.- ¿Qué tecnologías intervienen para tener un acceso fijo y un acceso móvil a internet?

Mediante el acceso fijo se emplea el mismo cable Ethernet que se conecta físicamente a una computadora (telefónico, coaxial, o fibra óptica) y así se obtienen ventajas considerables respecto al ancho de banda que pueda ofrecer el proveedor.

El Acceso Móvil a internet normalmente está asociado con lo que se conoce como WiFi (Wireless Fidelity), el tipo de señal que utiliza ondas electromagnéticas para lograr la transmisión de datos. Sin embargo, su alcance está limitado a un determinado radio de cobertura que depende directamente de la infraestructura que puedan ofrecer los proveedores de este servicio.

8.- Menciona al menos 3 lenguajes de programación que se utilicen del lado del cliente (front-end) y del lado del servidor (back-end).

Del lado del front-end destacan HTML, XML y JavaScript. Para el back-end existe PHP, PERL, Ruby, ASP y JSP.

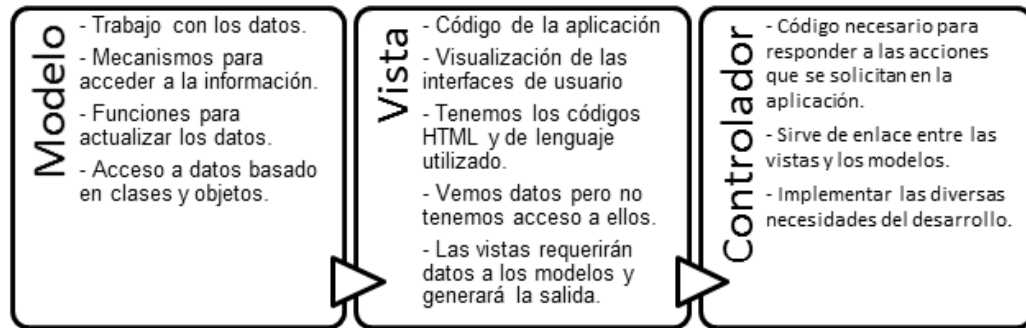
10.- ¿Qué son los Frameworks y cuando se utilizan? Mencione algunos de ellos.

Un Framework: es un entorno o ambiente de trabajo para desarrollo; es una herramienta auxiliar, hoy muy utilizada junto con los lenguajes de programación. En ella se tienen a la mano una serie de ayudas, compiladores, intérpretes, bibliotecas, plantillas, rutinas, librerías, máquinas virtuales y otros elementos que facilitan y ayudan durante el proceso de programación y desarrollo de sistemas.

Entre los frameworks podemos encontrar:

Herramienta o lenguaje	Framework
JavaScript	JQuery
Java	Spring, Struts
Python	Django
PHP5	Zend, Symfony, CodeIgniter
Ruby	on Rails
Microsoft	.NET WFC
C o Perl.	Alguna interfaz CGI
Plataformas de dispositivos móviles	JQuery
HTML5, CSS3 y Javascript	PhoneGap
Joomla	Zengrid
WordPress	Genesis
Sistemas operativos: iPhone, Android y Blackberry	Titanium

12.- ¿Explicar brevemente en que consiste el modelo MVC?



Plantea la separación de código en diferentes áreas de responsabilidad: modelo (interacción con los datos), vista (parte visual de la aplicación) y controlador (sirve como enlace entre las vistas y los modelos, como un “receptionista” que controla las acciones, pero no las lleva a cabo).

14.- Desarrolla una lista de actividades para crear una página Web desde su inicio hasta su publicación.

Algunas actividades que tendremos que desarrollar al momento de crear una página o sitio Web:

- Tener un objetivo claro y bien definido de lo que se quiere.
- Crear un proyecto de diseño, en donde definiremos los componentes (otras páginas) que integran nuestra página principal.
- Definir el contenido que tendremos en las páginas.
- Estructurar de tal forma que sea fácil su acceso, es decir, definir bien su estructura y arquitectura, tratando de establecer lo que se conoce como mapa de sitio (enlaces a todos los contenidos).
- Una vez que esté bien terminada y sin problemas, se deberá probar o “testear” en todos sus puntos y niveles con el fin de detectar errores.

- Ahora ya tendrás lista tu página para ser alojada o subida en algún servidor. Para ello se puede declarar tu equipo de cómputo como servidor local o en su defecto contratar un servicio de alojamiento para tu página (hosting).
- Contar con un nombre de dominio con el cual tendrás acceso a tu página desde cualquier lugar.
- Transferir todos tus archivos que integran tu página Web al servidor
- probar tu página, para verificar que tenga una funcionalidad completa.

Lo anterior es de manera general y sin mayor detalle. Conforme se tenga más experiencia se enriquecerá la descripción y su nivel de detalle aumentará.

Capítulo 3

Respuestas a pregunta pares:

Al igual que en el capítulo 2, citamos una de las posibles respuestas correctas a las preguntas pares del final del capítulo, obtenidas textualmente (o casi textualmente) del texto a lo largo del capítulo.

2.- ¿Qué relación tiene la Ingeniería de software con el desarrollo de aplicaciones, sistemas, portales o páginas Web?

La Ingeniería de Software es una de las ramas de la Informática que en términos generales trata aspectos relacionados con el desarrollo, operación y mantenimiento de productos de software, utilizando para ello, herramientas, procesos (técnicas) y métodos que faciliten su producción (programas, aplicaciones y/o sistemas), y que cumplan con los estándares que aseguren su calidad en todo momento.

La Ingeniería WEB se fundamenta en muchos conceptos y principios de la Ingeniería de Software pero con especificaciones propias del desarrollo de Sistemas Web. Se puede entender como el conjunto de métodos, técnicas y herramientas que deben ser utilizadas para abordar el buen desarrollo de sistemas y/o aplicaciones Web.

4.- Describa brevemente en qué consisten los siguientes estándares: ISO/IEC 12207 e ISO/IEC 9126:2001.

En general la Norma ISO 12207 establece todo lo relacionado al ciclo de vida del software, desde la conceptualización de ideas hasta la retirada. Consta de procesos para la adquisición y suministro de proyectos y servicios del software, definiendo puntos a seguir para su control y mantenimiento.

El estándar ISO9126 (2001) presenta un marco conceptual para el modelo de calidad y define un conjunto de características, refinadas en subcaracterísticas, las cuales debe

cumplir todo producto de software para ser considerado de calidad. Permite evaluar la calidad del producto de software y establece las características de la calidad de software.

6.- ¿Cómo aplicarías La Matriz PIT (Proceso-Información-Tecnología), al momento de iniciar el desarrollo de cualquier tipo de software?

Aún sin hacer uso de una metodología (o en combinación con alguna de ellas) antes de iniciar el desarrollo de cualquier tipo de aplicación se propone el uso de la matriz PIT como una herramienta más para tratar de obtener un diagnóstico rápido para guiar al desarrollador a plantearse una serie de cuestionamientos sencillos que lo ubiquen en la necesidad de desarrollar o no un nuevo producto y que consideraciones básicas deberá hacer.

8.- Explica en que consiste la Metodología de Diseño de Hipermedia Orientada a Objetos (MDHOO).

Esta metodología es el resultado de una extensión de Hypertext Design Model (HDM), utilizada para crear modelos de utilidad para el diseño de una aplicación de hipertexto. Se caracteriza por el diseño de las aplicaciones hipermedia, enfocándose en procesos de Ingeniería del Software. Está orientada a objetos y es muy aceptada para desarrollos de aplicaciones hipermedia, sobre todo en aplicaciones basadas en Web. Una de sus principales características es la de facilitar herramientas para controlar el desarrollo de aplicaciones cuyo comportamiento es complejo. Además de mantener bien identificados y por separado los procesos de diseño, con respecto a los del desarrollo.

10.- ¿Qué fases o etapas consideras que se deben de cubrir al momento de desarrollar un proyecto de desarrollo para aplicaciones Web?

Fase de estudio ejecutivo del proyecto

Fase de Planeación

Fase de análisis de requerimientos
Fase de Diseño
Fase de elaboración y creación de contenidos
Fase de Desarrollo
Fase de Pruebas
Fase de Control de calidad
Fase de implementación

12.- A lo largo de un desarrollo de sistemas o aplicaciones Web, ¿Qué importancias tienen los requerimientos y el o los usuarios?

Los requerimientos de los usuarios son el punto de partida de todo sistema WEB, ya sea que se trate de un usuario específico (como la adaptación de un sistema contable) o como un grupo masivo de usuarios (como las APP). En una primera fase del sistema se deben establecer los requerimientos en general, para posteriormente detallarlos conforme avance el proyecto.

Capítulo 4

Recomendaciones para el aprendizaje:

En primera instancia se mencionó la necesidad de reproducir los ejemplos citados bajo consideraciones que se juzgaron convenientes. Es recomendable que el estudiante vuelva a llevarlos a cabo sin ninguna ayuda en el editor de HTML que elija y, como último paso, realizar ejercicios similares con ligeras variantes.

El orden recomendado es el siguiente:

1.- Guardar en texto sencillo en Word como HTML para observar la gran cantidad de código que deja y confirmar que este no es un camino viable.

2.- Realizar la pantalla siguiente en HTML auxiliándose del editor que sea de su agrado (en el libro se sugiere PsPad). En esta versión no importan los estándares.

La Máscara

"Una y otra vez desgarró su piel ante el espejo, tratando de quitar la máscara que cubría su rostro. De pronto se detuvo; no tenía caso proseguir. La máscara era su verdadero rostro."

José Luis López Goytia

Si te gustan los minicuentos, puedes ver [El Cuento. Revista de Imaginación.](#)

3.- Se trata de obtener una pantalla como la mostrada a continuación, bajo los estándares HTML5-CSS3 validados en el sitio <http://validator.w3.org>.

Tres bellos pensamientos

Fragmentos de grandes obras literarias		
Autor	Pensamiento	Fuente
Antoine de Saint-Exupéry	"He aquí mi secreto, que no puede ser más simple: solo con el corazón se puede ver bien, lo esencial es invisible para los ojos."	El Principito
Ermilo Abreu	"...Hemos cambiado el conocimiento por la emoción, que es también una manera de penetrar en la verdad de las cosas."	Canek
Isabel Allende	"Entonces el hombre sintió el olor de animal montuno que se desprendía de esa mujer, el calor de incendio que irradiaban sus caderas, el roce terrible de sus cabellos, el aliento de yerbabuena susurrando en su oreja las dos palabras secretas a las cuales tenía derecho. - Son tujas, Coronel-dijo ella al retirarse-. Puedes emplearlas cuanto quieras."	Líder indígena que encabezó la rebelión maya de 1761. Cuentos de Eva Luna

Recomendación adicional: repetir el ejercicio por lo menos en dos ocasiones con variantes sin recurrir a ninguna ayuda, más allá del editor. Esto ayuda a crear un aprendizaje a mediano plazo. ¡No basta hacerlo una sola vez!

4.- Hacer pantallas que utilicen los tipos de campos y propiedades incorporados en HTML5. El primer conjunto pueden ser: color, fecha, hora, correo electrónico, url, cuadros de lista con función autocompletado, audio y vídeo

5.- Realizar programas con JavaScript que involucren recepción de datos, condicionales, ciclos, arreglos y subrutinas.

Recomendación adicional: como ya se mencionó, hacer al menos dos ejercicios similares sin ver ningún material impreso ni en línea.

6.- Explorar sitios que exploren las posibilidades de canvas 2D y 3D, y reproducir los ejemplos más sencillos. Entre otras sugerencias, puede utilizarse <http://www.kevs3d.co.uk/dev/>.

Capítulo 5

Estos fueron los ejercicios de aplicación recomendados al final del capítulo.

- **Ejercicio 1:** Instale en su computadora los entornos de desarrollo básicos para la programación en PHP, JSP y ASP.
- **Ejercicio 2:** Desarrolle el código correspondiente en PHP, JSP y ASP para elaborar un formulario que pida las horas trabajadas al día, el número de días trabajados y el pago por hora. Como resultado se deberá entregar en una página pago de las horas y días trabajados en función del cálculo que corresponda.
- **Ejercicio 3:** Desarrollar el código correspondiente en PHP, JSP y ASP para desarrollar un formulario que pida el nombre, dirección de correo y número de teléfono celular y lo guarde en una base de datos de MySQL o en su caso en Access.
- **Ejercicio 4:** Implemente los ejercicios vistos a lo largo de este capítulo en el respectivo servidor para PHP, JSP o ASP.
- **Ejercicio 5:** Desarrolle una aplicación Web en PHP, JSP y ASP que actualice una base de datos MySQL o en su caso en Access (altas, bajas, modificaciones y consultas). Utilice las sentencias de SQL.

Recomendaciones:

Como se observará, no se trata de preguntas cerradas, sino de temas a desarrollar. La página www.megasinapsis.com puede brindarle hipervínculos a sitios de material sobre ambientes WEB que aplican a este caso. Las ligas concretas:

<https://webbrain.com/brainpage/brain/7682D029-E72B-3ABF-2619-A832ECB39BBF#-3714>,

para fuentes de información (cursos en línea, revistas, etc.), tanto de aspectos generales como del área de informática.

Para referencias específicas en torno a la programación WEB puede revisarse

<https://webbrain.com/brainpage/brain/7682D029-E72B-3ABF-2619-A832ECB39BBF#-3726>.

En cuanto a las preguntas puntuales, nuevamente damos una de las posibles respuestas correctas a los cuestionamientos de número par.

2.- ¿Qué es un servidor web y cuál es su función?

El servidor WEB se encarga de interpretar y ejecutar los programas que son empleados para la programación WEB dinámica, es decir, aquella que hace toma información de la computadora que funge como servidor y/o accesos a una base de datos, proporcionando el dinamismo y la interactividad que el usuario requiere de una aplicación. Dentro de los lenguajes más utilizados en este contexto encontramos a PHP, ASP, JSP, PERL y Python.

4.- ¿Qué es un servidor Apache y qué tiene que ver con PHP, MySQL y HTML?

Dentro de los servidores WEB, el más popular es Apache Tomcat, bajo el cual se trabaja con PHP. También puede servir para ejecutar Javasever Pages, siempre y cuando no se trate de Java Enterprise Edition, la versión empresarial.

6.- ¿Qué son los scripts y cuál es su función principal?

Son programas que se ejecutan del lado del servidor (escritas con ASP, JPS o PHP) y que suelen combinarse con HTML y JavaScript.

8.- ¿Qué papel juega el HTML dentro de cualquiera de las plataformas de desarrollo Web?

El HTML indica al navegador (browser) la forma en que se presentará la información. Esta puede ser mediante texto, imágenes o video, así como su ubicación dentro de una página Web.

CSS (Cascading Style Sheets) hace posible la creación de páginas web, en conjunto con HTML. Considera y brinda facilidades para el desarrollo de páginas estéticas, además de propiciar la estandarización en la presentación de texto respecto al tipo de letra, colores de fondo, tamaños e imágenes, entre otras, con lo que se puede obtener un desarrollo de mejor calidad y presentación de una aplicación Web.

Cuando se llama a una página WEB, el servidor busca en sus archivos internos la página requerida. Una vez localizada, la devuelve al navegador, que la interpreta y la visualiza sin ningún problema, considerando que la página solo contiene código HTML. Si la página tiene insertado código PHP, ASP o JSP, este es interpretado y lo traduce a código HTML para que el navegador los pueda visualizar o mostrar.

Capítulo 6

A lo largo del capítulo se fueron externando diversas sugerencias didácticas, que a continuación concentramos como una guía de aprendizaje, con algunos agregados que creímos pertinente incluir.

1.- La programación WEB al inicio del siglo. Es necesario analizar el código, aunque no llevarlo a cabo. De esta forma se notarán todas las situaciones de riesgo que existían en desarrollos WEB anteriores al Modelo-Vista-Controlador. No basta con entenderlas en lo general: hay que ubicarlas con relación al código específico. Dicho análisis podría reflejarse en uno de los organizadores gráficos que existen (el más conocido: los mapas mentales).

2.- Ubique los diferentes contextos tecnológicos que existen, tanto aquéllos tradicionales (PHP, JPS, ASP) como aquellos relativamente nuevos (Python, Node .js, Unity, etc) y trate de encontrar ventajas, potencialidades y limitaciones de cada uno de ellos. No desdeñe posibilidades ni “combinaciones extrañas” (por ejemplo: con tecnología Microsoft se puede desarrollar para Android). Trate de sintetizarlo todo en una presentación gráfica (en este caso, pareciera que una herramienta como Prezi puede ser de gran ayuda). Trate de hallar las fuentes más confiables y actualizadas de información -artículos de revistas especializadas, tesis, cursos en línea, etc.-, tanto pagadas como de acceso libre (excepto libros piratas) pero renuncie a validar todos y cada uno de los aspectos, pues es prácticamente imposible lograrlo.

3.- Como ya se comentó a lo largo del capítulo, reproduzca el vídeo <https://www.youtube.com/watch?v=q10XMtDjTtg&index=3&list=PL5F907F818938859E> de Jesús Conde para darse cuenta de la forma en que se trabaja con un IDE para WEB –Netbeans, en este caso-. En este momento, con HTML y JSP, aún sin Modelo-Vista Controlador. Al decir reproducir, queremos decir visualizarlo, entenderle y recrearlo paso a paso, ya con el software de desarrollo instalado en su computadora.

4.- Haga lo mismo, pero ahora con el vídeo de Helmont Smith localizado en <https://www.youtube.com/watch?v=S-FGoEBjTSA>. Con él tendrá un panorama general del Modelo-Vista-Controlador (MVC).

5.- Como complemento final, ahora es el turno de otros vídeos del mismo autor para el modelo aplicado con acceso a base de datos.

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

<https://www.youtube.com/watch?v=ODAbnZLL4T8>

6.- La recomendación más extraña: si terminó el curso y ya no tocó el tema durante un mes, vuelva a leer el capítulo y a realizar los puntos 3, 4 y 5 para que no desaparezca rápidamente la información en su cerebro. Si no se refuerza el tema, la probabilidad de pronto olvido es altísima. Ya hemos observado dicho fenómeno en diversos temas de programación que se ven al final de un curso, como polimorfismo y apuntadores.

Capítulo 7

Son tres los aspectos que se pueden desglosar como ejercicios complementarios en este capítulo:

1.- La identificación de las tecnologías como usuario final. No es obvio como parece. Al menos deben considerarse tipos de dispositivos, sistemas operativos y sus versiones, resoluciones de pantalla, sitios para compras de app y comunicación inalámbrica. Sugerimos crear un organizador gráfico que pueda entrelazar temas.

2.- La creación de menús vía HTML con herramientas de apoyo como Dreamweaver o algún IDE tipo Netbeans y Eclipse. En este caso, aplicado completamente hacia dispositivos móviles. La recreación del ejemplo puede ser un muy buen primer acercamiento.

3.- En un libro introductorio como este no es posible avanzar a profundidad sobre cada uno de las áreas. Hemos arañado la superficie del tema únicamente. Para quien desee adentrarse en la programación de dispositivos móviles le recomendamos las plataformas de cursos en línea. A fines de 2016 uno de los cursos que pueden servir como base para la programación en Android es el de “Desarrollo de aplicaciones móviles con Android”, de Anahí Salgado, disponible en la plataforma Coursera. Dos recomendaciones para tomar este tipo de cursos: si ya está familiarizado en el tema, reserve el 80% del tiempo que le sugieren para el curso; si no es así, al menos duplique dicho tiempo. Por otra parte, el acceso al material es gratuito, mientras que la evaluación por parte de los mismos compañeros es pagada, aunque a un costo simbólico. Usted valore si valdría la pena o no pagar ese costo.