



BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE  
PUEBLA

FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

“Construcción de una página web con PHP y  $\LaTeX$   
para el aprendizaje de las matemáticas”

TESIS PROFESIONAL

QUÉ PARA OBTENER EL TÍTULO DE

LICENCIADO EN MATEMÁTICAS

PRESENTAN

Maura Cruz Hueyo Chino  
Rosa Velázquez Medina

DIRECTOR DE TESIS

MC Genaro Luna Carreto

Puebla, Pue.

Febrero 2015

# Agradecimientos

Los de Maura:

Mi tesis esta dedicada a la memoria de mis padres Avelina Chino Coyotl y Eustaquio Hueyo Romero. Además, para mi hermana Yolanda Hueyo Chino, a quien quiero mucho por su enorme apoyo durante mi carrera.

Los de Rosa:

Dedico este trabajo a Dios por apoyarme en todos mis esfuerzos y que nunca me abandona.

A mis padres: María Ofelia Medina González y Félix Velázquez Pérez aunque él ya no esté

A mi esposo con compartir mis triunfos: Leobardo Lira Reyes

A mis hijos por robarles un poco de su tiempo: David Lira Velázquez Daniel Lira Velázquez

A mis hermanos por apoyarme cuando los necesitaba: Lázara Velázquez Medina. Lázaro Velázquez Medina. J. Concepción Velázquez Medina. Modesta Velázquez Medina.

Gracias a todos ellos por compartir una vez más mis triunfos; Dios los bendiga

Finalmente, le damos las gracias a nuestro amigo GENARO LUNA CARRETO por su paciencia y apoyo en la dirección de esta tesis

# Introducción

Una página de internet esta constituida por dos partes: estructura y contenido.

La estructura, se refiere a la distribución de elementos, como títulos, menús, botones, colores etc. En su elaboración, confluyen una multitud de componentes y lenguajes con el fin de lograr una buena apariencia e interactividad con el usuario. Dentro de ellos, HTML, es el primario. Es factible dar buena apariencia a documentos web con sus instrucciones, pero se obtienen mejores resultados con lenguajes especiales como CSS.

Por otro lado, dentro de la estructura y con respecto al ingreso, descarga e intercambio de información, ha surgido otro tipo de programación que lo logra en forma muy eficiente. En este rubro se encuentran los lenguajes Java, JavaScript, Asp, PHP, AJAX, etc.

PHP es un lenguaje de programación que surgió en 1994, gracias al groenlandés, Rasmus Lerdorf. Es conocido por quienes se dedican a la generación de páginas web dinámicas, webmail, portales sociales etc. Se encuentra, por ejemplo, al llenar un formulario en internet, pues tiene una relación inseparable con bases de datos. Sin embargo, su presencia es poco notada, pues el término “página web” es muy amplio.

Dentro de sus virtudes están:

1. Los elementos necesarios para instalarlo son gratuitos. Tiene una estructura sintáctica sencilla. Además, se obtienen resultados inmediatos con poco código.
2. La universalidad de ejecución que poseen sus programas. Una vez que

se encuentran en algún servidor y sin instalar algún programa adicional, funcionan en celulares, tabletas y computadoras diversas.

3. Su omnipresencia en la internet.
4. La seguridad que representa su código, pues no es visible en las páginas web. De manera que es posible diseñar generadoras de exámenes, portales con cuentas de usuarios (con MySQL) con contraseña, etc.
5. Actualmente, es posible combinarlo con LaTeX vía web. Lo que produce un resultado espectacular en las fórmulas matemáticas, químicas, etc.

Precisamente, su buena relación con  $\text{\LaTeX}$  nos permitió vislumbrar las posibilidades de desarrollo de herramientas matemáticas que puedan ser usadas en diversos dispositivos electrónicos.

Así pues, el **objetivo general de esta tesis** es proporcionar a los docentes elementos de programación PHP, HTML y  $\text{\LaTeX}$  orientados hacia la construcción de programas computacionales, con un contenido matemático, incrustados en páginas web.

Como una consecuencia natural de ello, es posible lograr contenidos educativos de alta calidad.

Ahora hablemos de la estructura del trabajo. Se divide en 4 capítulos:

1. El primero explica la construcción y estilización básica de una página web con HTML y CSS.
2. En el segundo capítulo, se informa de las capacidades tipográficas de HTML, en cuanto a simbología matemática. Además, se explica la construcción de fórmulas matemáticas con  $\text{\LaTeX}$  en páginas web.
3. Se aborda la instalación y ejecución de programas PHP, así como su interacción con  $\text{\LaTeX}$ .
4. La cuarta parte, contiene el código de algunos programas básicos. La finalidad es mostrar las posibilidades de PHP operacionalmente junto con las virtudes tipográficas de  $\text{\LaTeX}$  en la web.

# Justificación

Es importante la **inclusión activa** de maestros y alumnos a las nuevas formas en que la información fluye. Particularmente, las referentes a la educación. Mal o bien, la transferencia digital de datos ha llegado para quedarse. Esa es la nueva dinámica mundial. En el Programa VIII (2) del Plan de Desarrollo Institucional Buap 2013-2017, se menciona:

“Es prioritario formar a los docentes en un segundo idioma y para el uso de las nuevas tecnologías de información y comunicación (TIC) como apoyo al proceso de enseñanza y aprendizaje”

Sin duda, han cambiado las formas de aprender, los hábitos de estudio. Es verdad que en internet, existe información sobre casi todo. Sin embargo, mucha de la información es poco seria y algunas veces errónea. Al momento de realizar tareas y trabajos, al estudiante, no le importa si tiene libros a disposición o no, busca la información que le parece pertinente en la web. No es posible revertir esta práctica. Entonces, es necesario poner a su disposición material virtual adecuado. Ya no sólo, páginas web estáticas, sino verdaderos programas, software que genere y muestre la solución detallada a ciertos problemas.

Es innegable que existe una cantidad inmensa de software sobre matemáticas, pero es generado, en su mayoría, con propósitos comerciales, no existe alguna intención pedagógica. Además, los programas computacionales de este tipo, en general, no son hechos por personas especializadas en matemáticas, lo cual engendra la misma problemática de aprendizaje. El software debe ir acompañado de la solución numérica, así como una explicación de la misma.

En cuanto al diseño de software didáctico distinguimos dos partes fundamentales:

1. Es necesario el conocimiento de algún lenguaje de programación
2. Tener conocimientos amplios en aspectos didáticos y de aprendizaje de las matemáticas.

En este trabajo abordaremos el primer punto, usando los lenguajes PHP, HTML y  $\text{\LaTeX}$ .

# Contenido

<b>Introducción</b>	<b>III</b>
<b>Justificación</b>	<b>v</b>
<b>1. Preliminares</b>	<b>1</b>
1.1. Historia de HTML y CSS . . . . .	1
1.2. Página web básica con HTML . . . . .	2
1.2.1. Elementos clásicos HTML en página web . . . . .	4
1.2.2. Estilo CSS (cascading style sheets) en página web . . . . .	7
<b>2. Simbología matemática en HTML</b>	<b>11</b>
2.1. Símbolos por defecto en HTML . . . . .	11
2.2. HTML y Mathjax . . . . .	14
<b>3. PHP, breve introducción</b>	<b>17</b>
3.1. Historia y modo de acción . . . . .	17
3.2. Instalación y comandos básicos . . . . .	18
3.2.1. Estructuras de control . . . . .	22
3.2.2. Datos de entrada, cadenas y arrays en PHP . . . . .	23
3.3. LaTeX y PHP . . . . .	27
<b>4. Ejemplos básicos</b>	<b>31</b>
4.1. Ecuación cuadrática . . . . .	31
4.2. Suma de potencias de números naturales . . . . .	37
4.3. Producto punto en $\mathbb{R}^n$ . . . . .	41
4.4. El coeficiente binomial como función de PHP . . . . .	45
4.5. Solución a triángulos rectángulos . . . . .	50
4.6. La integral definida . . . . .	59

VIII

*CONTENIDO*

**Conclusiones**

**67**

**Bibliografía**

**68**

# Parte 1

## Preliminares

La comprensión del trabajo requiere del conocimiento básico del lenguaje HTML. En este capítulo, mencionamos sus comandos básicos y los efectos en una página web. Además, introducimos los comandos de un lenguaje complementario a HTML: CSS. Ambos, logran efectos llamativos, muy importantes durante el aprendizaje.

### 1.1. Historia de HTML y CSS

HTML, son la siglas de HyperText Markup Language, que en castellano corresponde a *lenguaje de marcas de hipertexto*. Es un lenguaje de sintáxis muy sencilla que permite la construcción de páginas web.

Creado por el británico Tim Berners-Lee en 1990, HTML nació con el objetivo de que científicos pudiesen compartir, intercambiar y acceder a información de sus investigaciones. En realidad, su surgimiento se dio a la par del nacimiento de la **World Wide Web** (www), que es un medio de transporte importantísimo para conocer internet.

El conocimiento de HTML, es de importancia capital, si se desea distribuir información en internet. Además, otro punto a favor, es que los programas necesarios para su funcionamiento vienen por defecto en cualquier computadora.

Actualmente, es de uso frecuente en las escuelas de educación media y

media superior, sobre todo si se busca un primer contacto con la programación en general.

Por otro lado, HTML vino a ser complementado por las hojas de estilo en cascada o CSS (siglas en inglés de *cascading style sheets*). En realidad, se trata de un lenguaje creado en 1995, por el grupo **World Wide Web Consortium**. Su utilidad radica en dotar de una mejor apariencia o presentación a una página web, en cuanto a colores, fuentes, espaciamiento, márgenes, etc. Debido a su popularidad, la mayoría de navegadores tienen soporte para CSS.

## 1.2. Página web básica con HTML

Naturalmente que existe una variedad importante de programas que realizan por nosotros la tarea engorrosa de escribir en código HTML, lo cual, para los propósitos de este trabajo, no es deseable. De alguna manera, el uso de programas sofisticados que generan páginas espectaculares como dreamweaver, nos aleja de la posibilidad de creación de nuestras propias herramientas. Así que, construiremos una página web muy rudimentaria, pero suficiente, para nuestros objetivos.

En primer lugar, es necesario un editor. Es posible empezar con el clásico bloc de notas de windows, sin embargo, es incómodo. Se aconseja un editor que contenga numeración por línea y colores que nos indiquen donde se localizan los comandos o funciones.

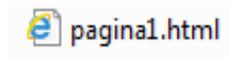
Un pequeño programa que satisface las características descritas anteriormente es notepad++<sup>1</sup>, que es de código abierto y distribución gratuita. Existen versiones de tal software que no requieren instalación, es suficiente con descomprimir. Es el caso de la versión *minimalist*, que sugerimos usar. Al momento de descargar y descomprimir, debemos buscar y ejecutar el archivo notepad++.exe.

Es muy importante que empecemos estableciendo cierto orden. Para ello, crearemos una carpeta que podamos ubicar rápidamente, por ejemplo, la colocaremos en el escritorio. A continuación, abrimos el programa. Iniciamos

---

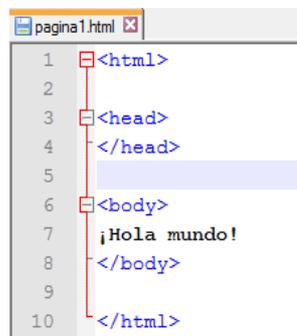
<sup>1</sup>La dirección de su página principal: <http://notepad-plus-plus.org/>

y guardamos un documento nuevo con el nombre `pagina1.html`. El contenido de la carpeta debe verse de la siguiente forma:



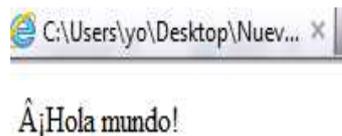
En realidad, la imagen del archivo depende del navegador que tengas como predefinido. Si le das clic, se abrirá una página del navegador en blanco, pues aún no hemos escrito nada.

Agreguemos algunas etiquetas html al archivo `pagina1.html`, a través de notepad++ y después guardemos los cambios:



```
1 <html>
2
3 <head>
4 </head>
5
6 <body>
7 ¡Hola mundo!
8 </body>
9
10 </html>
```

Se trata de una captura de pantalla con el código del programa clásico “hola mundo”. La ejecución del programa se realiza a través de la recarga de la página en blanco que ya creamos o pulsando la tecla F5. Se debe ver de la siguiente manera:



Se nota la aparición de un símbolo extraño al costado izquierdo del signo de admiración. Lo que ocurre es que html no reconoce el símbolo de admiración izquierdo, pues no se usa en la escritura inglesa. Hace algunos años era un martirio colocar tales símbolos. Se tenía que recurrir a instrucciones especiales de código HTML. La exclamación izquierda, por ejemplo, es resultado del comando:

`&iexcl;`

Si se deseaba escribir: ¡Cayó de rodillas! Se tenía que codificar:

`&iexcl; Cay&oacute; de rodillas!`

En la actualidad, ese problema se terminó. Todos los símbolos como acentos, exclamaciones, interrogación, diéresis, etc., se agregan automáticamente colocando la instrucción siguiente:

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>
```

en la parte del encabezado, esto es, el `<head>` de la página.

El objetivo de esta sección, construir una página web básica, sería agotador, si recurrimos a explicar cada comando, cada paso y su correspondiente visualización a través capturas de pantalla. Es mucho más fructífero hacernos de un diseño, una plantilla y explicar su codificación.

Ahora, colocaremos la captura de pantalla (figura 1.1) del código de una página, así como su salida en el navegador (figura 1.2). De esta manera, explicaré cada uno de sus elementos, distinguiendo entre instrucciones HTML y CSS. Es bueno experimentar la visualización de nuestra página en los diferentes navegadores, como chrome, firefox, explorer, opera, etc., con el fin de notar alguna diferencia.

### 1.2.1. Elementos clásicos HTML en página web

Nótese que las etiquetas html, se escriben por pares. Una indica la apertura `<>` y otra el cierre `</>`. Las que corresponden a `<html>`, se encuentran en las líneas 1 y 30. Las que corresponden a `<body>` en las líneas 10 y 29. Por otro lado, las que corresponden al párrafo `<p>` se localizan en 16 y 27.

Ya mencionamos la instrucción de `meta` de la línea 4. Se refiere al procesamiento y codificación de caracteres (unicode UTF-8).

```
pagina1.html x
1 <html>
2 <head>
3 <meta http-equiv='X-UA-Compatible' content='IE=edge,chrome=1' />
4 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
5 <style>
6 p{padding: 10px 10px 10px 10px}
7 #rodearimagen{float:left; margin: 25px 5px 0px 5px;}
8 </style>
9 </head>
10 <body bgcolor="">
11 <center><div style="background-color:#D0C2C2; width:700px; text-align:justify;
12 border: 2px solid black; font-size:13pt;">
13 <center><h1 style="background:#958888; margin:0px"> MATEMÁTICASFCE </h1></center>
14 
15 <center><h3>¿Quiénes somos?</h3></center>
16 <p>Matemáticasfce es un grupo de profesores que imparten asignaturas de
17 matemáticas en la Facultad de Ciencias de la Electrónica de la BUAP.
18 Se ha tenido la iniciativa de construir ésta página con la finalidad
19 de mantenerte informado sobre los diversos aspectos académicos
20 referentes a tu formación. Intentamos aglutinar la experiencia
21 de muchos años de enseñanza.
22 Por el momento se han incluido, la totalidad de los programas de
23 las materias impartidas en la facultad, el calendario de las
24 evaluaciones colegiadas por asignatura, más de una cuarentena de
25 páginas con diversidad de temas útiles e infinidad de ejercicios
26 resueltos, software funcional en diversas plataformas, una generadora
27 de exámenes aleatorios nivel medio y medio superior.</p>
28 </div></center>
29 </body>
30 </html>
```

Figura 1.1 Código básico de página web

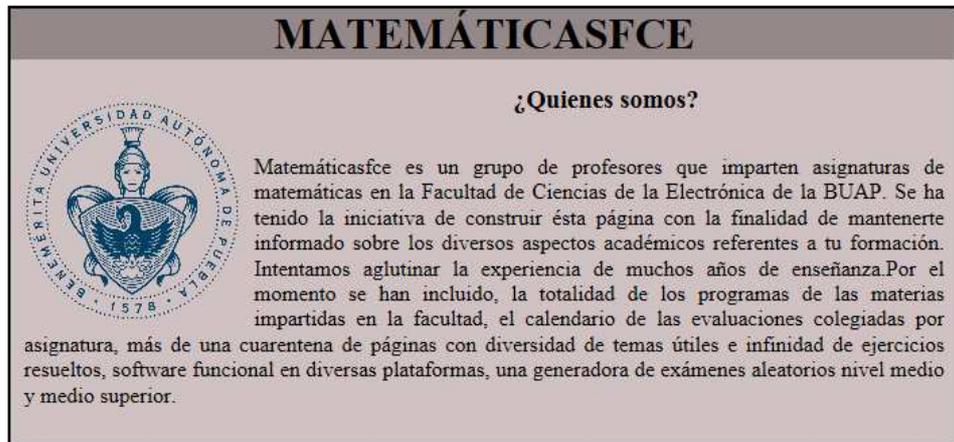


Figura 1.2 Salida de página web

La línea 3, es otro comando *meta*, cuyo propósito es que internet explorer, acepte funcionalidades más amplias logrando cierta compatibilidad, es decir, una visualización de páginas muy semejantes en diferentes navegadores.

En el caso del `<body>` de nuestro código, se nota que en la línea 10 aparece `<body bgcolor='''>`. Colocando un color específico dentro de las comillas de `bgcolor` cambia el color del fondo de la página. Hay dos formas de escribirlo. Una de ellas es escribir en inglés, por ejemplo:

```
<body bgcolor='red'>
```

También, utilizando notación hexagesimal: `#FF0000`. En internet, existen portales que muestran los códigos hexagesimales de una infinidad de colores.

Una etiqueta muy importante para organizar información es `<div>`. Es posible visualizar su acción, pues logra generar un contenedor más pequeño como el gris donde se encuentran nuestra información. Puede imaginar que la página principal es una hoja blanca tamaño carta y el `<div>` sería una hoja gris en el centro de la hoja. Sus características, se explicarán en la próxima sección.

Ahora bien, en la línea 14, se localiza

```
<img src='buap.png' width='170px' id='rodearimagen'>
```

que es la instrucción para incluir imágenes en la web. El elemento `id`, se explicará en la próxima sección. Es evidente que dentro de las comillas que corresponden a `<src>`, se pone el nombre y ubicación de la imagen, junto con su extensión. Es posible incluir imágenes de diferente formato: gif, jpg, bmp, svg, png, etc. El atributo `width` corresponde al tamaño de la imagen en px (píxeles).

Uno de los elementos más importantes en una página web, son los enlaces. Explicaremos la creación de uno de ellos. Note que en la línea 16 se localiza la palabra: “Mateamticascfe” Convirtámosla en enlace. En lugar de la palabra, escribamos todo el texto acotado por las etiquetas

```
<a>...</a>
```

como lo indica la siguiente imagen:

```
<a href="http://matematicascfe.netii.net/">Matemáticasfce</a> es un grupo
```

Obviamente, en `href` se coloca entre colillas la dirección electrónica donde nos enviará el enlace. La salida será, claro, el clásico enlace web



### 1.2.2. Estilo CSS (cascading style sheets) en página web

Se han incluido varios comandos de css en nuestra plantilla . En la etiqueta `<div>`, es posible visualizar varias comandos dentro de `style`:

```
background-color:#D0C2C2; width:700px; text-align:justify; border:
2px solid black
```

Cada propiedad asignada se encuentra separada por dos puntos (:) y cada comando por punto y coma (;).

El primero, es el color del contenido del `<div>`. El segundo se refiere al ancho del `<div>`. Es obvio, que los siguientes se refieren al texto justificado

y la orilla negra del mismo. Contenedores `<div>` pueden agregarse en documentos web tantos como se deseen.

Hemos agregado, en la instrucción de la imagen, el atributo `id`. Es sólo un indicador, que nos dice que la imagen está regida por las instrucciones que se encuentran en `#rodearimagen`, dentro del `<head>`, específicamente dentro de `<style>`, esto es, en la línea 7. Estos comandos, lo único que generan es un margen alrededor de la figura, así como logran que el texto rodee a la imagen. El `padding`, tiene cuatro medidas en pixeles. Corresponden a: margen arriba, margen derecho, margen abajo, margen izquierdo.

Se ha colocado una serie de órdenes CSS, incrustadas dentro del documento HTML. En la etiqueta `<style>` hay algunas y dentro de los comandos otras. Escribimos de esta forma por simplicidad. En realidad, por lo general y para para mayor orden, se construyen dos documentos. Uno HTML, que como ya notamos, se refiere más a la estructura, y otro, cuya extensión sería `.css` y que contiene la presentación de la página.

En la figura 1.3, se muestra el código de la página1, pero con menos elementos. No aparece en el `<head>` la etiqueta `<style>`. En su lugar (línea 5), se localiza

```
<link rel='stylesheet' type='text/css'
      href='pagina1.css' />
```

que informa que la hoja que da estilo a la página1 se encuentra en el archivo `pagina1.css`. Además, se presenta un `<div>` con menos elementos, pero contiene la etiqueta

```
class='contenedor'
```

Debemos entender que el estilo del `div` esta en el archivo `pagina1.css`, con la etiqueta `.contenedor`. En el archivo CSS (vea figura 1.4), tal etiqueta se visualiza en la línea 6. Naturalmente que el aspecto de la página web no cambia en absoluto.

```

pagina1.html x
1 <html>
2 <head>
3   <meta http-equiv='X-UA-Compatible' content='IE=edge,chrome=1' />
4   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
5   <link rel="stylesheet" type="text/css" href="pagina1.css"/>
6 </head>
7 <body bgcolor="">
8 <center><div class="contenedor">
9   <center><h1 style="background:#958888; margin:0px"> MATEMÁTICASFCE </h1></center>
10  
11  <center><h3>¿Quiénes somos?</h3></center>
12  <p>Matemáticasfce es un grupo de profesores que imparten asignaturas de
13  matemáticas en la Facultad de Ciencias de la Electrónica de la BUAP.
14  Se ha tenido la iniciativa de construir ésta página con la finalidad
15  de mantenerte informado sobre los diversos aspectos académicos
16  referentes a tu formación. Intentamos aglutinar la experiencia
17  de muchos años de enseñanza.
18  Por el momento se han incluido, la totalidad de los programas de
19  las materias impartidas en la facultad, el calendario de las
20  evaluaciones colegiadas por asignatura, más de una cuarentena de
21  páginas con diversidad de temas útiles e infinidad de ejercicios
22  resueltos, software funcional en diversas plataformas, una generadora
23  de exámenes aleatorios nivel medio y medio superior.</p>
24 </div></center>
25 </body>
26 </html>

```

Figura 1.3 Código página con enlace CSS

```

pagina1.html x | pagina1.css x
1
2
3 p{padding: 10px 10px 10px 10px}
4 #rodearimagen{ float:left; margin: 25px 5px 0px 5px;}
5
6 .contenedor{
7   background-color:#D0C2C2;
8   width:700px;
9   text-align:justify;
10  border: 2px solid black;
11 }

```

Figura 1.4 Archivo CSS



## Parte 2

# Simbología matemática en HTML

La página construida en el capítulo anterior, contiene solamente textos e imágenes. En este capítulo, insertaremos algunos símbolos matemáticos que posee HTML. Para ello, borraremos el contenido del párrafo y la imagen, con el propósito que haya espacio para empezar a colocar símbolos. Se entiende que de aquí en adelante estaremos escribiendo solamente entre las etiquetas `<div>`.

### 2.1. Símbolos por defecto en HTML

Claro que existe una lista completa de símbolos matemáticos HTML y algunos tienen apariencia muy aceptable. Es posible escribir cierto tipo de problema matemático con claridad.

En la figura 2.1 se observa la codificación de un problema, cuya salida se encuentra en la figura 2.2.

Conviene resaltar la aparición de los comandos que generan listas numeradas y no numeradas HTML, esto es,

```
<ol><li></li>... </ol> y <ul><li></li>... </ul>
```

respectivamente. Por cada `<li></li>` aparecerá un número o punto según sea el caso.

```

15 <ol>
16 <li> Se tienen tres conjuntos: A = { 1, 2, 3, 4, -1, -2, -3, 8 },
17   B = { 1, -1, 5, 8 } y C = { 7, 9, 0 }.<br>
18 <br>
19   En las opciones existe un x tal que:
20   <center> x &isin; A - (A &cap; B)</center> ¿Cuál es?
21 <ul>
22   <li>4</li>
23   <li>-1</li>
24   <li>5</li>
25   <li>7</li>
26 </ul>
27
28 </li>
29
30 <li></li>
31 </ol>

```

Figura 2.1 Código símbolos matemáticos HTML

**MATEMÁTICASFCE**

1. Se tienen tres conjuntos:  $A = \{ 1, 2, 3, 4, -1, -2, -3, 8 \}$ ,  $B = \{ 1, -1, 5, 8 \}$  y  $C = \{ 7, 9, 0 \}$ .

En las opciones existe un x tal que:

$$x \in A - (A \cap B)$$

¿Cuál es?

- 4
- -1
- 5
- 7

2.

Figura 2.2 Salida símbolos matemáticos HTML

<code>&amp;divide;</code>	÷	<code>&amp;plusmn;</code>	±	<code>&amp;not;</code>	¬	<code>&amp;Omega;</code>	Ω
<code>&amp;rarr;</code>	→	<code>&amp;forall;</code>	∀	<code>&amp;exist;</code>	∃	<code>&amp;empty;</code>	∅
<code>&amp;sum;</code>	∑	<code>&amp;Delta;</code>	Δ	<code>&amp;#8596;</code>	↔	<code>&amp;equiv;</code>	≡
<code>&amp;isin;</code>	∈	<code>&amp;sup2;</code>	²	<code>&amp;frac14;</code>	¼	<code>&amp;#8836;</code>	∉
<code>&amp;notin;</code>	∉	<code>&amp;radic;</code>	√	<code>&amp;infin;</code>	∞	<code>&amp;and;</code>	∧
<code>&amp;or;</code>	∨	<code>&amp;ang;</code>	∠	<code>&amp;#8735;</code>	⊥	<code>&amp;cup;</code>	∪
<code>&amp;cap;</code>	∩	<code>&amp;int;</code>	∫	<code>&amp;sube;</code>	⊆	<code>&amp;le;</code>	≤
<code>&amp;ge;</code>	≥	<code>&amp;perp;</code>	⊥	<code>&amp;real;</code>	ℝ	<code>&amp;#8484;</code>	ℤ

**Figura 2.3** Algunos símbolos matemáticos HTML

En la línea 20, aparecen los comandos

`&isin;` y `&cap;`

que generan la pertenencia conjuntual  $\in$  y la intersección  $\cap$ .

En la tabla 2.3 colocamos algunos símbolos, junto con su codificación HTML.

Hay posibilidades de generalizar un poco el que se refiere a la potencia cuadrada, esto es, `&sup2;`. El comando

`x<sup>m</sup>`

tendría como salida  $x^m$ .

En general, la simbología HTML tiene buenas características. Sin embargo, no es posible escribir cocientes de números de buena apariencia. Definitivamente, no es lo mismo  $\frac{5}{4}$ , que  $5/4$ . Note que, en el caso de la raíz,  $\sqrt{\quad}$ , aparece sin tapa. De manera que  $\sqrt{345}$ , se ve muy diferente que  $\sqrt{345}$

```

<script src='http://cdn.mathjax.org/mathjax/latest/MathJax.js'
type='text/javascript'>
  MathJax.Hub.Config({ HTML: ["input/TeX","output/HTML-CSS"],
  TeX: { extensions: ["AMSmath.js","AMSsymbols.js"],
    equationNumbers: { autoNumber: "AMS" } },
    extensions: ["tex2jax.js"],
    jax: ["input/TeX","output/HTML-CSS"],
    tex2jax: { inlineMath: [ ['$','$'], ["\\(", "\\)"] ],
    displayMath: [ ['$$','$$'], ["\\[", "\\]"] ],
    processEscapes: true },
    "HTML-CSS": { availableFonts: ["TeX"],
    linebreaks: { automatic: true } } });
</script>

```

Figura 2.4 Javascript de MaThJaX

## 2.2. HTML y Mathjax

Desde el año 2009, el problema de la visualización adecuada de fórmulas científicas en la red, parece haber encontrado su fin, y de la mejor forma: tipografía  $\text{\LaTeX}$ .

Mathjax es un motor de visualización JavaScript (vea [8]) para  $\text{\LaTeX}$ , cuya principal característica, es su funcionalidad en la mayoría de navegadores web, incluyendo los contenidos en dispositivos móviles.

Hay posibilidades de usar MathJaX si se dispone de conexión a internet o de manera offline. Para el primer caso, se debe incluir el script de la figura 2.4 en cualquier parte del `<head>` en nuestra página HTML y eso es todo. Si ahora escribimos una fórmula con comandos  $\text{\LaTeX}$ , en cualquier parte de nuestro documento, ésta se cargará automáticamente.

**Ejemplo 2.2.1.** Escribamos la fórmula general con sintáxis  $\text{\LaTeX}$  en la página que habíamos comenzado. Agregue el script que ya se mencionó en el `<head>` y en alguna parte del `<body>` de nuestra página web escriba lo siguiente:

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

La salida después de guardar y recargar la página es:

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

**Ejemplo 2.2.2.** Ahora un sistema de ecuaciones sencillo. En el `<body>` escribamos lo siguiente:

```
\[
\left\{
\begin{array}{rcr}
2x+y &=& 8 \\
x+y &=& -4
\end{array}
\right.
\]
```

Al guardar y recargar la página, se observará:

$$\begin{cases} 2x + y = 8 \\ x + y = -4 \end{cases}$$

Una buena parte de los comandos  $\text{\LaTeX}$  es aplicable a páginas web. El libro [7] contiene muchos ejemplos sencillos sobre codificación  $\text{\LaTeX}$ .

Ahora bien, sin internet, es factible utilizar MathJaX. Es necesario descargarlo de

<http://docs.mathjax.org/en/latest/installation.html>

Actualmente se encuentra en la versión 2.4 y pesa 32.6 mb. Una vez descargado, lo colocamos en la misma carpeta donde se encuentre nuestra página.

La descompresión generará una carpeta llamada `MathJax-2.4-latest`. Sólo resta agregar el script de la figura 2.5 en el `<head>` de la página. No es necesario borrar el otro script.

Si al momento de realizar tu página no se genera la numeración de ecuaciones, considera agregar

```
11 <script type="text/javascript" src="MathJax-2.4-latest/MathJax.js">
12   MathJax.Hub.Config({
13     extensions: ["tex2jax.js"],
14     jax: ["input/TeX", "output/HTML-CSS"],
15     tex2jax: {inlineMath: [{"\$", "\$"}, [{"\"(", "\")"}]}
16   });
17 </script>
```

Figura 2.5 Javascript de MaThJaX

```
<script type="text/x-mathjax-config">
  MathJax.Hub.Config({ TeX: { equationNumbers:
    {autoNumber: "all"} } });
</script>
```

en el <head> de la página.

## Parte 3

# PHP, breve introducción

La construcción de nuestros programas necesita 4 ingredientes: HTML, CSS, PHP y  $\text{\LaTeX}$ . Ya revisamos todos, excepto PHP. Éste, será el encargado de realizar las operaciones indicadas en la página.

### 3.1. Historia y modo de acción

El término PHP, corresponde originalmente a las siglas de *Personal Home Page*, cuya traducción en castellano es *página personal*. Es un lenguaje de código abierto creado por Rasmus Lerdorf en 1994 ([3]). El nombre original se debe a que Lerdorf, deseaba conocer cuántas veces era visto su curriculum en su página personal, para tal efecto, escribió varios programas en PERL. La utilidad potencial que representaban tales programas, le permitió visualizar su evolución como un lenguaje independiente.

Sus siglas también se asocian con el término *Hypertext Preprocessor*, pues su motor, analiza las páginas buscando si hay alguna instrucción o comando que ejecutar antes de mostrarlas al cliente.

Su código puede ser encajado en documentos HTML<sup>1</sup>, incluso generarlos. Por otro lado, su sintáxis tiene algunas características de C, Perl y Java, lo cual la hace sencilla de asimilar.

---

<sup>1</sup>Los especialistas de PHP aseguran, en términos formales, que éste genera la página HTML.

Actualmente, es de importancia capital en el desarrollo de páginas dinámicas. Por su relación estrecha con Mysql, va relacionado con bases de datos. Desde contraseñas, portales con cuentas de usuario, etc. Su utilidad es de diversa índole. Desde su página oficial

<http://php.net/>

es posible descargar actualizaciones, manuales, etc.

¿Cuál es la característica distintiva de PHP? Bueno, existen muchos tipos de lenguajes que se ejecutan del lado del cliente, como Javascript, esto es, se ejecuta en el navegador o browser del equipo. Por su parte PHP, al momento de darle clic a algun botón de envío, la información viaja hacia otra computadora (servidor), la cual procesa los códigos y regresa la información procesada mediante una página HTML. Tal forma de ejecución, permite cierta universalidad, es decir, no importa si la página web es abierta en cualquier sistema operativo, las operaciones y procesos se realizan en otro lado y solo se mostrarán los resultados en pantalla. Así que, la información procesada puede ser vista en diversos dispositivos móviles o de escritorio, incluso celulares, tablet, etc., sin instalaciones adicionales.

Dentro de otras bondades se encuentra su gratuidad, soporte para múltiples sistemas operativos, servidores web y bases de datos (MySQL). Como dato adicional considere el hecho que el código PHP escrito en una página web no es posible visualizarlo.

## 3.2. Instalación y comandos básicos

Ya mencionamos en la sección anterior, que PHP se ejecuta en un servidor<sup>2</sup>. Normalmente, una vez creados los archivos `php`, estos tendrían que colocarse o subirse a algún servidor externo. Sin embargo, la creación de páginas web requiere la realización de pruebas antes de su visualización en internet. Para ello, es de vital importancia poseer un servidor local. Lo más fácil es que nuestro mismo ordenador sea tal servidor. Existen varios programas que logran tal propósito. En este caso usaremos WAMP, que es gratuito y

---

<sup>2</sup>Existen versiones de PHP que se ejecutan en línea de comandos, sin necesidad de servidor.

de código abierto (open source).

Según la descripción de su página oficial [11]:

WampServer es un entorno de desarrollo web de Windows. Se le permite crear aplicaciones web con Apache 2, PHP y una base de datos MySQL. Junto, PhpMyAdmin permite administrar fácilmente tus bases de datos.

La instalación es sencilla y para evitar dificultades posteriores, se debe establecer la configuración por defecto. Al momento de finalizar, en la barra de tareas deberá aparecer un icono verde como el siguiente:



Escribamos nuestro primer programa. Las instrucciones de código `php` deben estar delimitadas por las etiquetas:

```
<?php    ...instrucciones    ...    ?>
```

Ahora bien, al momento de instalar `wamp`, se genera una carpeta `www`, cuya ruta es la siguiente:

```
c → wamp → www
```

Muy importante es no olvidar que, cualquier tipo de archivo `php`, para que pueda ser ejecutado localmente, deberá guardarse en dicha carpeta.

Con notepad++ creamos el archivo

```
file1.php
```

y lo guardamos en la carpeta `www` que se mencionó anteriormente. Coloquemos el contenido indicado en la figura 3.1.

Es el momento de ejecutar nuestro archivo. Abramos el navegador y escribamos en la barra de direcciones:

```
127.0.0.1/file1.php
```

```
1 <html>
2 <body >
3 <center>
4 <div style='width:250px; background:#CCCCCC;
5 text-align:justify;'>
6 <?php
7 echo " HOLA MUNDO";
8 ?>
9 </div>
10 </center>
11 </body>
12 </html>
```

Figura 3.1 Archivo PHP

```
1 <?php
2 echo "
3 <html>
4 <body>
5 <center>
6 <div style='width:250px; background:#CCCCCC;
7 text-align:justify;'>
8 HOLA MUNDO
9 </div>
10 </center>
11 </body>
12 </html>";
13 ?>
```

Figura 3.2 Archivo PHP

HOLA MUNDO

Figura 3.3

La salida del programa se ubica en la figura 3.3. Es el clásico: “hola mundo”. En realidad, es factible producir un resultado menos ostentoso, pero equivalente sólo con escribir:

```
<html>
<body>
<?php echo ‘‘HOLA MUNDO’’; ?>
</body>
</html>
```

La idea de escribirlo un poco más adornado, es mostrar dos formas de escribir código `php`. Si corremos el programa de la figura 3.2, se notará que se obtiene el mismo resultado que con el código de la imagen 3.3. Es visible en el primer caso, que el código `php`, se encuentra entre las líneas 6 y 8, y que se encuentra, aparentemente, incrustado entre comandos de HTML. En el segundo caso, `<?php ?>` abarca todo el programa, y los comandos HTML, son producidos por él.

Estos ejemplos que hemos colocado, logran evidenciar el comando del lenguaje PHP que imprime en pantalla. Su sintáxis:

```
<?php echo ‘‘...’’; ?>
```

El ejemplo sencillo

```
<?php echo‘‘Hello Georgie!!!’’; ?>
```

producirá la salida en pantalla: Hello Georgie!!!

## Variables PHP

Como en cualquier lenguaje de programación, es importante la definición de variables. En PHP, los nombres de las variables, van precedidos del signo de pesos: `$`.

**Ejemplo 3.2.1.** Escribamos en una página el código de la figura 3.4

Al momento de su ejecución saldrá el nombre Francisco tres veces seguidas. Note que en la línea 4, se encuentra dicho nombre, entrecomillado, asociado a la variable `$nom`

```

1 <html>
2 <body >
3 <?php
4 $nom='Francisco';
5 echo $nom;
6 echo "<br>$nom";
7 echo '<br>'.$nom.'';
8 ?>
9 </body>
10 </html>

```

Figura 3.4 Tres formas de imprimir

### 3.2.1. Estructuras de control

Las estructuras de control permiten modificar la ejecución de un programa. Generalmente se agrupan en condicionales, como el `if()` y bucles, como el `while`. Existen multitud de ellas, pero me limitaré a exponer la sintaxis de las que ocuparemos.

El `if` es una de las de mayor uso, su sintaxis es:

```

if(condición) {
    ... sentencias ...
}

```

El intérprete de PHP evalúa la condición y en caso de ser verdadera se ejecutarán las sentencias.

De la misma naturaleza, se encuentra el `if...else`

```

if(condición) {...sentencias...} else {...sentencias...}

```

Ahora bien, dentro de los bucles, se enlistarán dos estructuras: `while` y `for`.

```

while(condición) {
    ... sentencias ...
}

```

Esta instrucción nos permite realizar un conjunto de instrucciones, un número determinado de veces. Por su parte, el ciclo `for` es muy semejante a `while`, pero ligeramente más complejo.

```
for(exp_ini; exp_condición; bucle) {  
    ... sentencias ...  
}
```

En la expresión de inicialización (`exp_ini`), se declara e inicia la variable o variables. La condición, por su parte, debe contener los requerimientos para que las sentencias en `{ }` se ejecuten. Finalmente, el bucle modifica el valor de las variables utilizadas como contadores del mismo.

### 3.2.2. Datos de entrada, cadenas y arrays en PHP

Se reconocen en los formularios de HTML, la forma estándar de entrada de datos que tomará PHP. Veamos como procede la toma y procesamiento de datos.

La sintáxis de un formulario HTML se aprecia en la siguientes líneas:

```
<form action='archivo.php' method='post' >  
<input type='text' name='a' size='10' >  
<input type='submit' name='enviar' value='CALCULAR'>  
</form>
```

Dichas instrucciones generan las clásicas cajas de texto que abundan en internet. Expliquemos brevemente sus partes.

El atributo `action` indica la dirección del archivo o url donde se procesará la información. Por su parte, `method` indica la forma de transferencia de datos, en este caso, `post`. El segundo renglón, genera propiamente dicho, la caja de texto, donde entren 10 caracteres y cuyo dato de entrada, al ser enviado será identificado por PHP como

`$_POST['a']`

El penúltimo renglón, produce un botón de envío con la palabra: `CALCULAR`. El `name` de esta línea, es el nombre con el cual sabremos si el botón fue oprimido.

Se debe decir que el formulario, puede ser colocado en una página HTML, pero la página de procesamiento, indicada en `action`, en nuestro caso, debe

```
1 <html>
2 <body >
3 <center>
4 <div style='width:250px; background:#CCCCCC;
5 text-align:justify;'>
6 <form action='file1.php' method='post' >
7 <input type='text' name='tunombre' size='10' >
8 <input type='submit' name='enviar' value='ENVIAR'>
9 </form>
10 <?php
11 if(isset($_POST['enviar'])) {
12     $tunombre=$_POST['tunombre'];
13     echo "HOLA $tunombre ";
14 }
15 ?>
16 </div>
17 </center>
18 </body>
19 </html>
```

Figura 3.5

ser PHP.

Explicado así suena complicadísimo. Con el fin de clarificar ideas hagamos un ejemplo.

**Ejemplo 3.2.2.** Escribiremos una página que pida nombre por teclado y al momento de enviar aparezca un letrero de bienvenida con el nombre introducido.

Tomemos la página PHP que ya creamos: `file1.php`. Recuerde que corresponde al código de la figura 3.1. Lo modificaremos, agregaremos un formulario y un `if` dentro del código PHP. Vea la figura 3.5.

El formulario corresponde a las líneas 6 a 9. Se ha colocado como el archivo que procesará la información, el mismo de salida. El nombre introducido en la caja de texto (línea 7), se identificará al momento del envío como



Figura 3.6

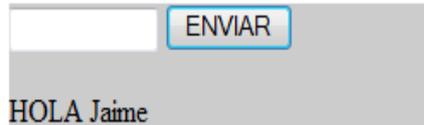


Figura 3.7

```
$_POST['tunombre'].
```

Por otro lado, el botón ENVIAR tiene nombre `enviar` (con minúsculas). La apariencia del formulario se encuentra en la figura 3.6.

Al momento de cargar la página, el comando `if`, verifica antes que nada, si el botón ENVIAR ha sido pulsado. Ésta tarea la realiza con la función

```
isset($_POST['enviar'])
```

¿Qué ocurre al momento de colocar tu nombre y pulsar ENVIAR? La función `isset` indica tal hecho y entonces, se ejecutan las sentencias escritas dentro del `if`. Por simplicidad, asignamos (vea línea 12) un nombre más sencillo a la variable enviada

```
$tunombre = $_POST['tunombre']
```

En realidad, para no excederse en nombres de variables, recuperamos el nombre con el cual fue identificada en el formulario. Posteriormente, viene la impresión en pantalla (vea figura 3.7)

### String y arrays

Ahora bien, es momento de hablar de dos conceptos muy importantes de PHP: `string` y `arrays`.

Por un lado, un **string** representa un cadena de caracteres. En las secciones anteriores, ya se introdujeron ejemplos:

```
$nom='Francisco';
```

De hecho, en los formularios, los datos introducidos son tomados como cadenas.

Es posible identificar a cada caracter de un **string**. Si consideramos a la variable `$nom='Francisco'`, resulta sencillo, entender que

```
$nom[0]=F,$nom[1]=r, $nom[2]=a ...
```

El desarrollo del trabajo nos presentará otros comandos referentes a **string**.

Por su parte, un **array**, hablando en términos computacionales, es un vector o matriz cuya estructura permite el almacenamiento de un conjunto de datos, bajo un mismo nombre.

La estructura de un **array** tiene, evidentemente, una infinidad de aplicaciones. Agregue el hecho de que los datos que formen parte de **array** pueden ser de diferente naturaleza: números, **string**, **array**, etc.

**Ejemplo 3.2.3.** Observe la sintaxis de un **array**:

```
$T=array(2/3,'pera','2/3');
```

De igual manera que un **string**, cada entrada de un **array** puede identificarse:

```
$T[0]=0.666666,$T[1]=pera, $T[2]=2/3
```

Note que en el caso de `$T[0]`, dio el resultado de la división. En `$T[2]`, el resultado es `2/3`, pues es considerado como **string** y no como número. Durante el desarrollo de aplicaciones se verá una forma para traducir fracciones **string** a números, separando su numerador y denominador, usando un **array**.

### 3.3. LaTeX y PHP

En las secciones anteriores, se estudió la sintáxis tanto de  $\text{\LaTeX}$  como de PHP. En el caso de éste último, se emplea la expresión

$$\langle ?\text{php} \dots ? \rangle$$

con el propósito de señalar la apertura y cierre de instrucciones PHP. Además, sus variables empiezan siempre con el signo \$.

En el caso de  $\text{\LaTeX}$ , es de uso frecuente, aunque poco recomendado, el uso de

$$\text{\$} \dots \text{\$} \text{ y } \text{\$\$} \dots \text{\$\$}$$

para indicar modo matemático en línea y centrado.

En este punto nos damos cuenta que hay incompatibilidad por el uso reservado del símbolo \$ en ambos lenguajes. Afortunadamente, existe otra manera de indicar modo matemático en  $\text{\LaTeX}$  que no interfiere con PHP, el cual ya se ha mencionado en la sección 2.2. Nos referimos a

$$\text{\textbackslash}( \dots \text{\textbackslash}) \text{ y } \text{\textbackslash}[ \dots \text{\textbackslash}]$$

**Ejemplo 3.3.1.** Nos servirá la misma página `file1.php` que ya tenemos, sólo la guardaremos como `file2.php` y borraremos el contenido del `div`.

En la figura 3.8 se muestra el código con diversas formas probables de imprimir instrucciones  $\text{\LaTeX}$ . La vista en pantalla sería

$$\text{\textit{rac}23}, \frac{2}{3}, \frac{2}{3}, \frac{2}{3}, \sqrt{4}, \sqrt{\$r}, \sqrt{\$r}, \sqrt{4}$$

respectivamente.

Algunas dan buenos resultados, otras no tanto. La instrucción con mucha versatilidad, y creemos más adecuada, está en la línea 52.

En ocasiones hemos tenido problemas con el reconocimiento de comandos  $\text{\LaTeX}$  en impresiones PHP. La mayoría de las veces, los hemos solucionado usando la estructura de la instrucción de la línea 52, y duplicando el `backslash` (`\`) en los comandos  $\text{\LaTeX}$ .

```

43 <?php
44 $r=4;
45 echo "<br> \(\ \frac{2}{3} \),";
46 echo "\(\ \\\frac{2}{3} \), ";
47 echo ' \(\ \frac{2}{3} \), ';
48 echo ' \(\ \displaystyle \\\frac{2}{3} \), ';
49 echo "\(\ \sqrt{ $r} \), ";
50 echo '\(\ \sqrt{ $r} \), ';
51 echo '\(\ \\\sqrt{ $r} \), ';
52 echo '\(\ \sqrt{'. $r. ' } \)';
53 ?>

```

Figura 3.8

```

80 <?php
81 echo ' \(\
82     \left( \begin{array}{rr}
83         2 & 3 \\
84         4 & 5
85     \end{array} \right) \)';
86 ?>
87 \(\ \left( \begin{array}{rr}
88     2 & 3 \\
89     4 & 5
90 \end{array} \right) \)
91
92 <?php //sirve con doble comilla o comilla simple
93 echo '\(\
94     \\\left( \\\begin{array}{rr}
95         2 & 3 \\
96         4 & 5
97     \\\end{array} \\\right) \)';
98 ?>

```

Figura 3.9

$$(2 \quad 3 \quad 4 \quad 5) \begin{pmatrix} 2 & 3 \\ 4 & 5 \end{pmatrix} \begin{pmatrix} 2 & 3 \\ 4 & 5 \end{pmatrix}$$

Figura 3.10

**Ejemplo 3.3.2.** Vea la codificación de la figura 3.9, es para matrices. Su salida se encuentra en la imagen 3.10

La primera está impresa por un código `php` (líneas 80-86). Se nota que su estructura es estándar para  $\text{\LaTeX}$ . Sin embargo, su salida no es óptima.

Ahora, en las líneas 87-90, se encuentra el mismo código pero fuera de la instrucción PHP y se imprime adecuadamente.

El último ámbito PHP, mostrado en la figura, contiene duplicados los símbolos `\`, excepto los de entrada y salida del modo matemático, y la salida es muy buena. Este último, funciona de la misma manera si colocamos comilla doble.

En el siguiente ejemplo, se muestra la interacción de un comando `while` que genera impresiones en pantalla  $\text{\LaTeX}$ .

**Ejemplo 3.3.3.** Considere la suma de fracciones siguiente:

$$\frac{1}{2} + \frac{2}{3} + \frac{3}{4} + \frac{4}{5} + \frac{5}{6}$$

Con el código de la figura 3.11, obtendremos la misma suma y con un bucle `while`, generaremos automáticamente una suma más grande.

Vea la página generada por el código en la figura 3.12. Es indeseable el símbolo “+” que quedó al final de la suma de la segunda expresión. Es posible eliminarlo si el `while` lo corremos hasta menores que trece (línea 108) y agregamos en la línea 113, lo siguiente:

```
echo '\frac{13}{14}';
```

```

104  $$\frac{1}{2}+\frac{2}{3}+\frac{3}{4}+\frac{4}{5}+\frac{5}{6} $$
105  <?php
106  $i=1;
107  echo '[';
108  while($i<= 13){
109  $j=$i+1;
110  echo ' \frac{ '.$i.' }{'.$j.' }+';
111  $i++;
112  }
113
114  echo '[';
115  ?>

```

Figura 3.11

$$\frac{1}{2} + \frac{2}{3} + \frac{3}{4} + \frac{4}{5} + \frac{5}{6}$$

$$\frac{1}{2} + \frac{2}{3} + \frac{3}{4} + \frac{4}{5} + \frac{5}{6} + \frac{6}{7} + \frac{7}{8} + \frac{8}{9} + \frac{9}{10} + \frac{10}{11} + \frac{11}{12} + \frac{12}{13} + \frac{13}{14} +$$

Figura 3.12

## Parte 4

# Ejemplos básicos

El propósito de este capítulo es presentar algunos procesos matemáticos mecanizados por un programa. Ejemplos sencillos que permitan ilustrar técnicas de programación, así como las bondades visuales y, por lo tanto didácticas de PHP y su salida  $\text{\LaTeX}$ . Aunque hay posibilidades de generar programas muy interesantes, como el proceso de Gram-Schmidt ([5]), el proceso de Gauss-Jordan para calcular la inversa de una matriz ([4]), Regla de Cramer, etc. y todo lo que se pueda lograr con un lenguaje de programación.

### 4.1. Ecuación cuadrática

La solución de la ecuación cuadrática con el uso de la fórmula general, siempre se presta como excelente ejemplo ilustrativo. A decir verdad, el código de solución es sencillo. Sin embargo, la finalidad es explicar, presentar los pasos de solución, para que el estudiante de nivel medio y medio superior, genere y obtenga solución a sus propias ecuaciones de manera correcta.

La forma estándar de escribir una ecuación cuadrática es:

$$ax^2 + bx + c = 0$$

Recuerde que dicha ecuación tiene soluciones reales, si la expresión, llamada discriminante,  $D = b^2 - 4ac$ , es mayor o igual a cero.

El código de la figura 4.1 colocado en el `body` de una página web, genera tres cajas de texto para introducir los coeficientes  $a, b, c$  de la ecuación cuadrática. Expliquemos la forma de procesar los datos:

```

38 <form action='cuadraticatess.php' method='post' >
39 <input type='text' name='a' size='7' style="font-size:15pt">$x^2+$
40 <input type='text' name='b' size='7' style="font-size:15pt">$x+$
41 <input type='text' name='c' size='7' style="font-size:15pt">$=0$
42 <br><input type='submit' name='enviar' value='ENVIAR'>
43 </form>
44 <?php
45 if(isset($_POST['enviar'])) {
46     $a=$_POST['a']; $b=$_POST['b']; $c=$_POST['c'];
47     $D=pow($b,2)-4*$a*$c;
48     if($D>=0) {
49         $x1=(-$b+sqrt($D))/(2*$a);
50         $x2=(-$b-sqrt($D))/(2*$a);
51         echo "<br>\(x1='.$x1.')\<br>";
52         echo "<br>\(x2='.$x2.')\<br>";
53     } else {echo "La ecuación no tiene soluciones reales";}
54     }
55 <?>

```

**Figura 4.1** Programa que resuelve ecuaciones cuadráticas

1. Introducción de los coeficientes:  $a, b, c$
2. Oprimir enviar
3. La función `isset`, indica que se ha realizado el envío
4. Entra en acción la estructura de control `if`, pues su condición ahora es verdadera.
5. Cálculo del discriminante  $D$
6. Otra estructura de control `if` entra en acción. Si  $D \geq 0$ , calcula e imprime soluciones. De lo contrario (`else`), imprime la carencia de soluciones

El discriminante  $D$  es calculado<sup>1</sup> por la expresión de la línea 47.

---

<sup>1</sup>Aquí aparecen los operadores aritméticos clásicos:  $*$ ,  $/$ ,  $-$ . Así como el operador de comparación:  $>=$  (mayor o igual). En algún momento se usará  $<>$  para indicar diferente.

Si el discriminante  $D$  es mayor o igual a cero (línea 48), muestra en pantalla las soluciones, calculadas a partir de las expresiones de las líneas 49 y 50. En caso contrario ( $D < 0$ ), tomado por el `else`, imprime un mensaje que expresa la carencia de soluciones reales.

Es notable la facilidad de la aplicación. Sin embargo, muestra la forma de acción de varios comandos. Agregaremos una serie de elementos  $\text{\LaTeX}$  que muestren la solución un poco más detallada y vistosa.

Explicemos un poco del código (vea figura 4.2). En la línea 51, aparece la función

```
substr_count()
```

que guarda el número de veces que aparece el símbolo `/` en la variable `$a`. Lo mismo ocurre para los renglones 53 y 55, en el caso de `$b` y `$c`. Esta instrucción nos indicará si el usuario introdujo una fracción. Si fue así,

```
preg_split()
```

de la línea 52, generará un `array` a partir del `string $a`, partiéndolo en elementos precisamente donde aparece `/`. Si se introdujo `2/3`, se produce

```
$aa=array(2,3),
```

de donde `$a=$aa[0]/$aa[1]` (línea 52), es ahora un número.

Por otro lado, es claro que `pow()` (línea 57) es la función potencia, `sqrt()` (línea 59) la raíz y `round()` (línea 60) es el redondeo de un número, en este caso a 4 dígitos.

El resto es código  $\text{\LaTeX}$ , estructurado, como lo indica la sección 3.3.

**Ejemplo 4.1.1.** La salida para la ecuación

$$x^2 - 5x + 2 = 0$$

se vería como lo indica la figura 4.3.

La apariencia del programa ya es buena, pero no se aborda el caso de soluciones complejas. Razón por la cual, proporcionamos un código (figura 4.4) el cual se debe agregar en la parte del `else`.

```

47 <?php
48 if(isset($_POST['enviar'])){
49     $a=$_POST['a']; $b=$_POST['b']; $c=$_POST['c'];
50     $a1=$a;$b1=$b;$c1=$c;
51     if( substr_count( $a, "/" ) <>0 ){
52         $aa=preg_split("/\\/", $a);$a=$aa[0]/$aa[1];}
53     if( substr_count( $b, "/" ) <>0 ){
54         $bb=preg_split("/\\/", $b);$b=$bb[0]/$bb[1];}
55     if( substr_count( $c, "/" ) <>0 ){
56         $cc=preg_split("/\\/", $c);$c= $cc[0]/$cc[1];}
57     $D=pow($b,2)-4*$a*$c;
58     if($D>=0){
59         $x1=(-$b+sqrt($D) )/(2*$a); $x2=(-$b-sqrt($D) )/(2*$a);
60         $x1=round($x1,4);$x2=round($x2,4);
61         echo '<br>las soluciones de la ecuación:
62         \ ( ('. $a1 .' )x^2+ ('. $b1 .' )x + ('. $c1 .' ) =0, \ )
63         son: \ (x1='. $x1 .' \), \ (x2='. $x2 .' \)<br>';
64         echo "<br>Método de solución:
65         \[ \\begin{align*}
66         x &= \\frac{-b \\pm \\sqrt{b^2 -4 ac}}{2a} \\ \\ \\
67         x &= \\frac{-($b1) \\pm \\sqrt{($b1)^2 -4 ($a1) ($c1)}}{2($a1)} \\ \\ \\
68         x &= \\frac{-($b) \\pm \\sqrt{ $D}}{2($a)} \\ \\ \\
69         \\end{align*} \\]
70         \ ( \\displaystyle x1= \\frac{-($b) + \\sqrt{ $D}}{2($a)} = $x1 \\),
71         \ ( \\displaystyle x2= \\frac{-($b) - \\sqrt{ $D}}{2($a)} = $x2 \\)
72         ";
73         } else { echo 'la ecuación no tiene soluciones reales';}
74     }
75 ?>
76

```

Figura 4.2 Programa que resuelve ecuaciones cuadráticas

**ECUACIÓN CUADRÁTICA**

$x^2 +$    $x +$    $= 0$

Las soluciones de la ecuación:  $(1)x^2 + (-5)x + (2) = 0$ , son:  $x_1 = 4.5616$ ,  $x_2 = 0.4384$

Método de solución:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$
$$x = \frac{-(-5) \pm \sqrt{(-5)^2 - 4(1)(2)}}{2(1)}$$
$$x = \frac{-(-5) \pm \sqrt{17}}{2(1)}$$
$$x_1 = \frac{-(-5) + \sqrt{17}}{2(1)} = 4.5616, x_2 = \frac{-(-5) - \sqrt{17}}{2(1)} = 0.4384$$

**Figura 4.3** Solución de una ecuación cuadrática

```

76 $D1=-$D; echo "<br> La ecuación \(( $a1)x^2+ ($b1)x + ($c1) =0\) no tiene
77 soluciones (reales), pues su discriminante es negativo, es decir
78 \\begin{align*}
79 b^2-4ac &= ($b1)^2-4($a1)($c1) \\|\\|\\|
80 &= ($b)^2-4($a)($c) \\|\\|\\|
81 &= $D <0
82 \\end{align*}
83 <u>Nota </u>: si te encuentras estudiando números complejos, lee esto:
84 <br><br>Método de solución para el caso complejo:<br><br>
85 \\begin{align*}
86 x &= { -b\pm \sqrt{b^2-4ac} \over 2a } \\|\\|\\|
87 x &= { -($b1)\pm \sqrt{($b1)^2-4($a1)($c1)} \over 2($a1) } \\|\\|\\|
88 x &= { -($b)\pm \sqrt{($b)^2-4($a)($c)} \over 2($a) } \\|\\|\\|
89 x &= { -($b)\pm \sqrt{ $D } \over 2($a) } \\|\\|\\|
90 \\end{align*}
91 Siempre hay dos soluciones. La primera solución se obtiene cambiando
92 el símbolo \(( \pm) por \(( +):
93 \\begin{align*}
94 x1 &= { -($b)+ \sqrt{-1} \sqrt{ $D1 } \over 2($a) } \\|\\|\\|
95 x1 &= { -($b)+ i \sqrt{ $D1 } \over 2($a) } \\|\\|\\|
96 x1 &= { -($b) \over 2($a)}+ { \sqrt{ $D1 } \over 2($a) } i
97 \\end{align*}
98 Realizando operaciones análogas y cambiando el símbolo \(( \pm) por \(( -):
99 \\begin{align*}
00 x2 &= { -($b)- \sqrt{ $D } \over 2($a) } \\|\\|\\|
01 x2 &= { -($b) \over 2($a)}- { \sqrt{ $D1 } \over 2($a) } i
02 \\end{align*}";

```

Figura 4.4 Código para soluciones complejas

```

40 <br>
41 <h3>Suma de $n$ cuadrados</h3>
42 <form action='cur.php' method='post'>
43   Introduce $n$: <input type='text' name='n' size='10'>
44   <input type='submit' name='enviar' value='CALCULAR' >
45 </form>
46 <?php
47 if(isset($_POST['enviar'])) {
48   $n=$_POST['n'];
49   $suma=$n*( $n+1)*(2*$n+1) / 6;
50   echo '<br> \(\ \displaystyle \sum_{i=1}^{'. $n. '}i=';
51   $i=1; $s='';
52   while($i<=$n) {
53     $sum=''. $s. '+' . $i. '^2';
54     $s=$sum;
55     $i++;}
56   $sum=substr($sum,1); /*substr quita el primer caracter*/
57   echo '' . $sum. '=' . $suma. '' ;
58   echo '\) ' ;
59 }
60 ?>

```

Figura 4.5 Código que suma cuadrados de números naturales

## 4.2. Suma de potencias de números naturales

Este ejemplo es muy importante porque muestra nuevamente la interacción del comando `while` de PHP, con la impresión en pantalla de símbolos  $\LaTeX$ .

Se trata de calcular la suma de los primeros  $n$ -cuadrados. Escriba en el `<body>` de una página web, debidamente codificada para aceptar  $\LaTeX$ , el código indicado en la figura 4.5.

Una vez guardado y ejecutado en el explorador debe verse un campo de texto como el de la figura 4.6.

Al momento de introducir, por ejemplo el valor 7, debe de obtenerse una salida como lo indica la figura 4.7.

Una buena parte del código sirve para generar automáticamente, a través

**Suma de  $n$  cuadrados**Introduce  $n$ :  

Figura 4.6

**Suma de  $n$  cuadrados**Introduce  $n$ :  

$$\sum_{i=1}^7 i = 1^2 + 2^2 + 3^2 + 4^2 + 5^2 + 6^2 + 7^2 = 140$$

Figura 4.7

de un ciclo `while`, la visualización de la suma de cuadrados, no para calcular la suma.

En la línea 49 se encuentra la fórmula

$$\sum_{i=1}^n = \frac{n(n+1)(2n+1)}{6}$$

que es la que realiza los cálculos. En las líneas 50 y 58 se localiza la apertura y cierre del campo matemático  $\text{\LaTeX}$ , junto con el código para imprimir sigma. Entre dichas líneas se localizan las instrucciones del ciclo `while`.

Claro que se puede obtener el cálculo de la suma sin recurrir a la fórmula, sólo hace falta un `while` muy semejante al que ya aparece en la codificación:

```

    $i=1;$t=0;
while($i<=$n) {
    $sum2=$t +pow($i,2);
    $t=$sum2;
    $i++;
}
echo '<br>'.$sum2.'<br>';

```

Hemos probado que la suma pedida coincide con `sum2`.

Un aspecto a destacar en esta parte, es que, si deseáramos, la suma de los primeros  $n$  números naturales elevados, por ejemplo a la 8, sólo sería cuestión de cambiar en el código anterior `pow($i,2)` por `pow($i,8)`. Es más, agreguemos un cuadro de texto al formulario y modifiquemos el código, de tal manera que se pida el número de naturales a sumar y la potencia a la cual tendrán que ser elevados (vea 4.8).

En la figura 4.9, se encuentra la suma de los cuatro primeros números naturales elevados a la potencia 11

salida del resultado, para los cuatro primeros números naturales elevados a la potencia 11.

```

41 <h3>Suma de potencias de números naturales</h3>
42 <form action='cur.php' method='post'>
43   Introduce el número de naturales a sumar:<br><br>
44   <input type='text' name='n' size='10'> <br><br>
45   Introduce la potencia:<br><br>
46   <input type='text' name='m' size='10'>
47   <input type='submit' name='enviar' value='CALCULAR' >
48 </form>
49 <?php
50 if(isset($_POST['enviar'])){
51     $n=$_POST['n'];
52     $m=$_POST['m'];
53     $i=1;$t=0;
54     while($i<=$n){
55         $suma=$t +pow($i, $m);
56         $t=$suma;
57         $i++;
58     }
59     echo '<br> \(\ \displaystyle \sum_{i=1}^{' . $n . '} i^{' . $m . '} =';
60     $i=1; $s='';
61     while($i<=$n){
62         $sum=' ' . $s . '+' . $i . '^{' . $m . '}';
63         $s=$sum;
64     }
65     $sum=substr($sum,1); /*substr quita el primer caracter*/
66     echo ' ' . $sum . '=' . $suma . ' ';
67     echo '\)';
68 }
69 ?>

```

Figura 4.8 Código para suma de potencias de naturales

**Suma de potencias de números naturales**

Introduce el número de naturales a sumar:

Introduce la potencia:

CALCULAR

$$\sum_{i=1}^4 i^{11} = 1^{11} + 2^{11} + 3^{11} + 4^{11} = 4373500$$

**Figura 4.9** Salida suma de potencias de naturales

En matemáticas, para obtener una fórmula que calcule tal suma, es necesario usar la propiedad telescópica de la sumatoria, y conocer fórmulas anteriores: cuadrados, cubos, cuartos y así hasta la décima potencia.

**4.3. Producto punto en  $\mathbb{R}^n$** 

En cualquier curso básico de álgebra lineal, se proporciona la definición de producto punto para un par de vectores en  $\mathbb{R}^n$ .

Si  $A = (a_1, \dots, a_n)$ ,  $B = (b_1, \dots, b_n)$ , entonces:

$$A \cdot B = a_1 b_1 + a_2 b_2 + \dots + a_n b_n$$

Veamos primero un código simple (figura 4.10) que obtiene el resultado. El programa generará dos cajas de texto, donde tendrás que escribir las componentes de los vectores separadas por comas.

En la imagen 4.11 se muestra la forma de introducir los datos. Una vez ejecutado, el resultado que genera, por supuesto, es 12.

Note que en el código aparecen, tres funciones muy importantes de PHP:

```

37 <form action='pptess.php' method='post' >
38 $A=$<input type='text' name='a' size='20' style="font-size:15pt" >
39 $,B=$<input type='text' name='b' size='20' style="font-size:15pt">
40 <br><input type='submit' name='enviar' value='ENVIAR'>
41 </form>
42 <?php
43 if(isset($_POST['enviar'])) {
44     $vec1=trim($_POST['a']); $vec2=trim($_POST['b']);
45     $T1=preg_split("/\\,/",$vec1);
46     $T2=preg_split("/\\,/",$vec2);
47     $lonT1=count($T1);
48     $lonT2=count($T2);
49     $i=0;$s=0;
50 while($i<= ($lonT1-1) ){
51     $pp=$T1[$i]*$T2[$i] + $s;
52     $s=$pp;
53     $i++;
54 }
55 echo $pp;
56 }
57 ?>

```

Figura 4.10 Código producto punto

,

Figura 4.11

```

55 //echo $pp;
56 echo '\(';
57 $i=0;$w='';
58 while ($i<= ($lonT1-1)){
59   $producto= ('.$T1[$i]. ')'. ''.$T2[$i]. ');
60   $producto= $w. '+'. $producto ;
61   $w= $producto;
62   $i++;
63 }
64 $producto=substr($producto,1);
65 echo ''.$producto.'='.$pp.'';
66 echo '\)';
67 }
68 ?>

```

Figura 4.12

`trim()`, `preg_split()`, `count()`

La función `trim()` simplemente elimina espacios en blanco. En el caso de

`preg_split(/\./, $vect2)`

construye un **array**, cuyas entradas, son exactamente los elementos separados por la coma. Por lo tanto, `$T1` y `$T2` son **array**.

Por último, `count()`, calcula el número de elementos de un **array**.

Una observación con respecto al ciclo `while` (línea 50), es que la condición para `i`, se da hasta la longitud de `T1` menos 1, lo que ocurre es que el ciclo empieza en cero (línea 49) y no en uno.

La salida del programa es simplemente un número. Agregaremos un código que escriba los productos y las sumas (4.12). La salida tiene una mejor apariencia (4.13)

Ahora lograremos que nuestro programa escriba los vectores, para ello es necesario un código más, vea la figura 4.14. En la figura 4.15 se encuentra la salida.

Quizá, para una buena cantidad de personas el trabajo anterior es suficiente. Sin embargo, las personas dedicadas al quehacer matemático, saben

ENVIAR

$$(-1)(2) + (0)(8) + (2)(1) + (-3)(-4) = 12$$

Figura 4.13

```

67 echo" <br>\( \overrightarrow{A} \cdot \overrightarrow{B}=\) ";
68   $i = 0;
69 echo " \( (";
70 while ($i<= ($lonT1-2)){
71   echo "$T1[$i], &nbsp; ";
72   $i++;
73 }
74 while ($i<= ($lonT1-2)){
75   echo "$T[$i], &nbsp; ";
76   $i++;
77 }
78 echo ''.$T1[$lonT1-1].'';
79 echo" ) \) ";
80 $i = 0;
81 echo " \(\cdot \: (";
82 while ($i<= ($lonT1-2)){
83   echo "$T2[$i], &nbsp; ";
84   $i++;
85 }
86 while ($i<= ($lonT1-2)){
87   echo "$T2[$i], &nbsp; ";
88   $i++;
89 }
90 echo ''.$T2[$lonT1-1].'';
91 echo" ) = $producto= $pp \) ";

```

Figura 4.14

$$\vec{A} \cdot \vec{B} = (-1, 0, 2, -3) \cdot (2, 8, 1, -4) = (-1)(2) + (0)(8) + (2)(1) + (-3)(-4) = 12$$

**Figura 4.15**

que es necesario investigar si el programa acepta números del tipo 2.3 y 2/6. Sin duda alguna, el programa acepta el primer tipo. El problema viene con los cocientes de enteros. Hay posibilidades de obtener un programa (visite [6]) que acepte racionales de un forma u otra y que la solución la presente en fracciones, pero esos casos no los veremos aquí.

## 4.4. El coeficiente binomial como función de PHP

El coeficiente binomial se basa en la definición de factorial.

Si  $n$  es un número natural:

$$n! = 1 \cdot 2 \cdot \dots \cdot n$$

Además, se define  $0! = 1$ .

Por la naturaleza del concepto, es fácil diseñar un programa que calcule el factorial. En realidad, ya existe dentro de PHP, la función `gmp_fact()`, que corresponde al factorial. Nosotros escribiremos un programa que lo calcule, como pretexto para explicar la construcción de funciones PHP.

En la figura 4.16, hemos agregado un código que calcula el factorial.

A su vez, en la figura 4.17, se localiza el resultado para  $n = 7$ . Se visualiza el producto de los números reales hasta el número dado.

Aunque la codificación es muy aparatosa, si observa con cuidado, el programa tiene dos partes:

1. Una de ellas es el cálculo del factorial, hablando numéricamente, con un bucle `while`

```

39 <form action='factorial.php' method='post' >
40 $n$=<input type='text' name='n' size='2' style="font-size:15pt">
41 <BR><br>
42 <input type='submit' name='enviar' value='CALCULAR SOLUCIÓN'>
43 </form>
44 <?php
45 if(isset($_POST['enviar'])){
46     $n=$_POST['n'];
47     $i=1;$m=1;$m2='';
48     while($i<=$n){
49         $f=$i*$m; $F=''.$m2.''.$i.'\cdot';
50         $m2=$F; $m=$f;
51         $i++;     }
52         $F=trim($F, '\cdot'); //quita el último punto \cdot
53         echo '<br>\(' . $n . ' != ' . $F . ' = ' . $f . '\)<br>';
54     }
55 ?>

```

Figura 4.16

$n=$

$7! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 \cdot 7 = 5040$

Figura 4.17

```
5 <?php
6 function factorial($n) {
7     $i=1;$m=1;
8     while($i<=$n){
9         $f=$i*$m;
10        $m=$f;
11        $i++;    }
12    return $f;    }
13 ?>
```

Figura 4.18 Función factorial

2. Por otro lado, la visualización del producto con la variable `$F`, pero usando el mismo bucle.

Por el momento no nos interesa que aparezca el producto de números sino sólo el resultado del factorial.

Escribiremos el código (vea figura 4.18) como una **función** de PHP y le llamaremos `factorial.php`. Es notable la eliminación de muchos elementos de visualización del código anterior y la etiquetas HTML. Sólo debe estar el código `php` que aparece en la figura 4.18.

La imagen 4.18, resulta un buen ejemplo ilustrativo, donde se muestra la estructura básica de una función PHP:

1. El nombre de la función por el cual será llamada (línea 6) y el dato a evaluar. En este caso, el nombre de la función es `factorial` y el dato a evaluar `$n`
2. Las operaciones que se harán con los datos introducidos
3. El valor de la función una vez realizadas las operaciones. En este caso `$f` (línea 12).

Debemos diseñar una página HTML, donde colocaremos el cálculo del coeficiente binomial. Llamémosla `coeficiente.php`.

```

45 <form action='coeficiente.php' method='post' >
46   \[ \displaystyle \binom{n}{k} \nonumber \]
47   Escribe los valores de
48   $n=$ <input type='text' name='n' size='2' /> &nbsp;   y
49   $k=$ <input type='text' name='k' size='2' /> <br><br>
50   <input type='submit' name='enviar' value='CALCULAR' />
51 </form>
52 <?php
53 include('factorial.php');
54 if( isset($_POST['enviar']) ) {
55     $n=$_POST['n'];
56     $k=$_POST['k'];
57     $c= factorial($n) / ( factorial($n-$k)*factorial($k));
58     echo '<br> \[ \displaystyle \binom {'. $n. '}{'. $k. '} ='. $c. '\)';
59 }
60 ?>

```

**Figura 4.19** Código del coeficiente binomial

Recordar la definición de coeficiente binomial

$$\binom{n}{k} = \frac{n!}{(n-k)!k!}$$

En la figura 4.19 se muestra el código para tal efecto. Sin duda, es un programa legible. La definición de coeficiente binomial se encuentra operacionalmente en la línea 57.

Una instrucción nueva es la localizada en la línea 53, a saber

```
include('factorial.php')
```

Es una estructura de control, que indica que durante la evaluación del programa, se usarán las instrucciones contenidas en el archivo `factorial.php`. Recuerda que, una estructura de control permite modificar la ejecución de un programa.

En la figura 4.20 se encuentra la salida para  $n = 120$ ,  $k = 8$ . La longitud del número, es de 12 cifras.

Es importante mencionar que, dentro de un mismo archivo se pueden definir varias funciones al mismo tiempo y la llamada no sería individual, sino

Escribe los valores de  $n =$   y  $k =$

CALCULAR

$$\binom{120}{8} = 840261910995$$

**Figura 4.20** Salida del cálculo del coeficiente binomial

con el nombre del archivo que las contiene y por medio de `include()`.

Un aspecto poco considerado, es la dimensión que adquieren las cifras. El cálculo de coeficientes binomiales puede dificultar las operaciones si se tratan números grandes. Piense en el factorial de 30

$$30! = 265252859812191058636308480000000$$

Es posible diseñar un programa más potente utilizando funciones que hagan operaciones con números muy grandes. Se sabe que `bcmul()` y `bcdiv` pueden lograr operar números sin importar su longitud.

En el programa `factorial.php` sustituya `$f` por

```
$f=bcmul(''.$i.'',''.$m.'');
```

Además, en el programa `coeficiente.php` cambie `$c` por

```
$c=bcdiv(factorial($n), bcmul( factorial($n-$k), factorial($k) ));
```

Con dichas modificaciones puede obtener los coeficientes con mayor facilidad. Por ejemplo:

$$\binom{1000}{20} = 339482811302457603895512614793686020778700$$

## 4.5. Solución a triángulos rectángulos

En esta sección construiremos un programa que resuelve triángulos rectángulos.

El programa estará formado por dos archivos `php`:

`tritess.php` y `tritesssolucion.php`

El primer archivo recibirá el caso y los datos. El segundo procesará la información y mostrará los resultados. Requerimos entonces una manera de mostrar el menú con los posibles casos. Para ello, dentro del campo de los formularios HTML, existe una etiqueta especial: `select`.

En la figura 4.21 se muestra un código y la imagen 4.22 parte de su salida en pantalla.

Se puede notar que la etiqueta `select` se localiza entre las líneas 71-82. El resultado de salida es la típica cortina que se despliega al instante de oprimir la flecha. En la línea 71, se encuentra su nombre: `caso`. Al momento de realizar alguna selección, la etiqueta toma el valor de la opción, en este caso del 1 al 10.

Otro dato a destacar es que, el envío, según lo indica la línea 70, será procesado por la misma página.

No se debe olvidar que las páginas que hemos estado trabajando contienen, principalmente, dos tipos de códigos: HTML y PHP. El archivo inicial `tritess.php` contendrá código PHP, que entrará en acción al oprimir `EMPEZAR SOLUCIÓN`.

En esta situación, es conveniente describir la serie de eventos que ocurrirán a través de un algoritmo:

1. Seleccionar caso  $i$ . Donde  $i \in \{1, \dots, 10\}$ .
2. La etiqueta `select`, toma el valor  $i$ , es decir, `caso==i`.
3. Oprimir `EMPEZAR SOLUCIÓN`

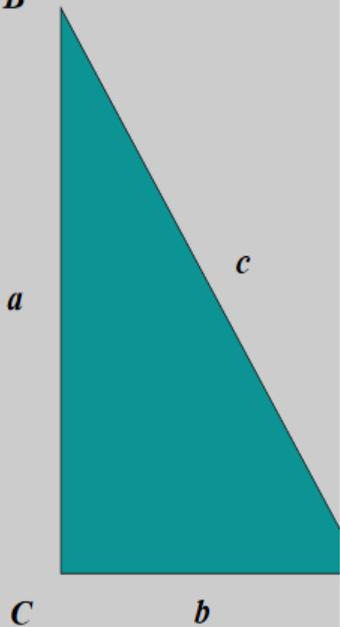
```
63 <center><h2>Programa que da solución a triángulos rectángulos</h2></center>
64 <br> <br>
65 <b>En la parte baja de la página encontrarás un repaso.</b> <br><br>
66 <center>
67 
68 </center> <br> <br> <br> <br>
69 <center>
70 <form action='tritess.php' method='post'>
71 <SELECT name='caso' style="" >
72 <option VALUE='1'>1.-¿Qué datos tienes?</option>
73 <OPTION VALUE='2'>2.-a y b (catetos)</OPTION>
74 <OPTION VALUE='3'>3.-c y a (hipotenusa y cateto)</OPTION>
75 <OPTION VALUE='4'>4.-c y b (hipotenusa y cateto)</OPTION>
76 <OPTION VALUE='5'>5.-Ángulo A y c (ángulo A e hipotenusa)</OPTION>
77 <OPTION VALUE='6'>6.-Ángulo A y a (ángulo A y cateto)</OPTION>
78 <OPTION VALUE='7'>7.-Ángulo A y b (ángulo A y cateto)</OPTION>
79 <OPTION VALUE='8'>8.-Ángulo B y c (ángulo B e hipotenusa)</OPTION>
80 <OPTION VALUE='9'>9.-Ángulo B y a (ángulo B y cateto)</OPTION>
81 <OPTION VALUE='10'>10.-Ángulo B y b (ángulo B y cateto)</OPTION>
82 </SELECT><BR><BR>
83 <center>
84 <input type='submit' name='enviado' value='EMPEZAR SOLUCIÓN' />
85 </center>
86 </form>
87 </center>
88
```

Figura 4.21

En la parte baja de la página encontrarás un repaso.

1.-¿Qué datos tienes?

EMPEZAR SOLUCIÓN



The diagram shows a right-angled triangle with vertices labeled  $B$  at the top,  $C$  at the bottom-left, and a right angle at the bottom-right. The vertical leg is labeled  $a$ , the horizontal leg is labeled  $b$ , and the hypotenuse is labeled  $c$ .

Figura 4.22

4. La función `isset`, indica que la variable `$_POST['enviado']` está definida, es decir, `isset($_POST['enviado'])` es verdadera
5. Entra en acción `if ( isset($_POST['enviado']) )`
6. La estructura de control anterior, tiene como sentencias, diez enunciados `if`, uno para cada caso. Obviamente, se ejecutará solo la que corresponde al caso *i*.
7. Las sentencias del `if` que corresponde al caso *i*, generan un formulario con cajas de texto, el cual será procesado por

`tritesssolucion.php`

Además, aparecerá un botón de envío llamado `enviar i`. En otras palabras,

`enviar1, enviar2, enviar3,enviar4,...,enviar10`

8. Al momento de oprimir `enviar i`, comienza la acción de

`tritesssolucion.php`

9. Se cumple la condición `isset($_POST['enviar i'])`
10. Entra en acción `if ( isset($_POST['envir i']) )`
11. La estructura de control anterior, tiene como sentencias, diez enunciados `if`, uno para caso. Obviamente, se ejecutará solo la que corresponde a `enviar i`.
12. Se procesan los datos y se imprimen resultados.

En la imagen 4.21, son visibles los diez casos. En realidad, son nueve, pues el número uno es la pregunta: ¿Qué datos tienes? (vea línea 72).

Ahora expliquemos el caso que corresponde a la opción dos, donde se conocen los dos catetos. No olvidemos considerar y repasar los pasos indicados en el algoritmo.

En la imagen 4.23, se muestra la codificación de la opción 1 y 2, dentro de las sentencias de

```

100 if ($_POST['caso']== 1) {echo "";}
101 if ($_POST['caso']== 2) {
102     .....
103     echo "<br><br><b>Coloca valores de catetos \ (a \) y \ (b\).</b>"
104     <form action='tritesssolucion.php' method='post' >
105     <br>
106     \ (a = \) <input type='text' name='a' size='8' />
107     \ (b = \) <input type='text' name='b' size='8' />
108     <BR><BR>
109     <input type='submit' name='enviar2' value='CALCULAR' style='
110     background-color:#1275ae;' />
111     </form>;
112 }

```

Figura 4.23

```
if(isset($_POST['enviado']))
```

El caso 2 (línea 101), producirá dos cajas de texto donde se deben colocar los valores de los catetos  $a$  y  $b$ . Además, un botón de envío con nombre: “enviar2” (línea 109). Se nota en la línea 104 que el envío del formulario será recibido y procesado por

```
tritesssolucion.php
```

En forma similar, cada `if` de la opción respectiva generará cajas de texto con el propósito de captar los datos respectivos, ya sea que se trate de lados o ángulos. Sin duda, dichos datos serán procesados por el archivo ya mencionado.

¿Qué ocurre al momento de enviar la información? Entonces entra en acción el archivo `tritesssolucion.php`. Se ejecutará solamente la estructura `if(isset($_POST['enviar2']))` (vea figura 4.24).

La línea 131 renombra a los catetos

```
a1=$_POST['a'] y b1=$_POST['b']
```

por cuestiones de sencillez. En las líneas 131 y 134, se realiza una conversión a número, por si acaso se introdujeron expresiones del tipo  $2/3$ .

```

129 if (isset($_POST['enviar2'])) {
130     .....
131     $a1=trim($_POST['a']); $b1=trim($_POST['b']);$a=$a1;$b=$b1;
132 if( substr_count( $a, "/" ) <>0 ){
133     $aa=preg_split("/\\/"/, $a); $a=$aa[0]/ $aa[1]; }
134 if( substr_count( $b, "/" ) <>0 ){
135     $bb=preg_split("/\\/"/, $b); $b=$bb[0]/ $bb[1]; }
136     .....
137     $c=sqrt(pow($a,2)+pow($b,2));
138     $A=Asin($a/$c);
139     $A=rad2deg($A);
140     $B=90-$A;

```

Figura 4.24

En la línea 131 se escribió  $a=a1$ , pero ¿existe diferencia entre  $a1$  y  $a$ ? Sí la hay. Si como dato de entrada se escribió  $1/2$ , entonces  $a1=1/2$  y  $a=0.5$ .

Entre las líneas 137-140, se obtienen los elementos faltantes del triángulo. Es clara la presencia del teorema de pitágoras en 137.

Por su parte, en el renglón 138, se aplica la definición de seno:

$$\text{sen } A = \frac{a}{c}$$

que claro, depende de las letras marcadas en el triángulo rectángulo de la página.

El despeje de  $A$  se logra con la función inversa de seno, que en el caso de PHP es: `Asin`. Otro aspecto de llamar la atención es el hecho de `Asin` se alimenta de radianes y produce radianes. Por ello, es necesaria la introducción de las funciones:

`rad2deg()` y `deg2rad()`

La primera traduce de radianes a grados, la otra de grados a radianes. En la línea 139 se realiza la conversión de radianes a grados para obtener  $B$ .

```

146 echo "
147 <br>
148 <h3>Método de solución, dados los catetos \((a, b)\)</h3><br><br>
149 En el caso de la hipotenusa \((c)\):
150 \\begin{align*}
151   c &= \sqrt{ a^2 + b^2 } \\\\
152   c &= \sqrt{ (\$a1)^2 + (\$b1)^2 } \\\\
153   c &= \sqrt{ (\$a) ^2 + (\$b)^2 } \\\\
154   c &= \$c \quad \\end{align*}
155
156 Ahora el ángulo \((A)\):
157 \\begin{align*}
158   \sin A &= \frac{a}{c} \\\\
159   A &= \arcsin \bigg( \frac{a}{c} \bigg) \\\\
160   A &= \arcsin \bigg( \frac{ \$a }{ \$c } \bigg) \\\\
161   A &= \$A \quad \\end{align*}
162 El ángulo \((B)\) es sencillo calcular a partir de los datos
163 disponibles.<br>
164 Sabemos que \((B+A=90)\), entonces
165
166 \\begin{align*}
167   B &= 90-A \\\\
168   B &= 90 -\$A \\\\
169   B &= \$B
170 \\end{align*} ";

```

Figura 4.25

Los resultados serán impresos con el código de la figura 4.25.

**Ejemplo 4.5.1.** Si introducimos el clásico caso de  $a = 3$  y  $b = 4$ , la salida se muestra en la figura 4.26.

**Método de solución, dados los catetos  $a, b$**

**$c$**

En el caso de la hipotenusa  $c$ :

$$c = \sqrt{a^2 + b^2}$$

$$c = \sqrt{(3)^2 + (4)^2}$$

$$c = \sqrt{(3)^2 + (4)^2}$$

$$c = 5$$

Ahora el ángulo  $A$ :

$$\sin A = \frac{a}{c}$$

$$A = \arcsin\left(\frac{a}{c}\right)$$

$$A = \arcsin\left(\frac{3}{5}\right)$$

$$A = 36.869897645844$$

El ángulo  $B$  es sencillo calcular a partir de los datos disponibles.  
Sabemos que  $B + A = 90$ , entonces

$$B = 90 - A$$

$$B = 90 - 36.869897645844$$

$$B = 53.130102354156$$

Figura 4.26

## 4.6. La integral definida

La integración es uno de los procesos matemáticos más estudiados. El problema del área, del cual surgió, nos remonta hasta la cultura helénica, esto es, a más de dos milenios. De manera que, a lo largo de los años, se han desarrollado varias formas de abordar su estudio.

Diversos autores recurren al supremo e ínfimo para construir el concepto de función integrable (vea por ejemplo [9, p. 355]). Otros, empiezan diferente: consideran las sumas de riemann ([10]). Esa será la versión que ocuparemos para nuestro programa.

Suponga que  $f$  es una función continua en  $[a, b]$ . Una forma de definir la integral (vea [10, p. 372]) es la siguiente:

$$\int_a^b f(x) = \lim_{n \rightarrow \infty} \sum_{i=1}^n \Delta x f(x_i)$$

donde  $\Delta x = \frac{b-a}{n}$  y  $x_i = a + i\Delta x$ .

La expresión  $\sum_{i=1}^n \Delta x f(x_i)$  es un tipo especial de suma de riemann, donde  $n$  es el número de subintervalos de igual longitud en los cuales se divide el intervalo  $[a, b]$ .

La clave en el desarrollo de nuestro programa, la podemos visualizar en la forma de la suma:

$$\sum_{i=1}^n \Delta x f(x_i) = \Delta x f(x_1) + \Delta x f(x_2) + \Delta x f(x_3) + \cdots + \Delta x f(x_n)$$

Si consideramos un valor particular de  $n$ , sólo se trata de ir evaluando la función  $f$  en el punto  $x_i$  y después multiplicarla por  $\Delta x$ , hasta llegar a  $n$ . Lo cual se puede lograr en forma recursiva utilizando un ciclo **while**.

La aproximación de la integral, propósito de esta sección, se realiza considerando la suma de riemann anterior y los siguientes datos proporcionados por el usuario: la función  $f$ , el intervalo  $[a, b]$  y el número  $n$ . Es bien sabido

```

8  <?php
9  function calc($equation)
10 {
11
12     $equation = preg_replace('/\s+/', '', $equation);
13     $number = '(?:0|[1-9]\d*)(?:\.\d*)?(?:[eE][+-]?\d+)?|pi|π';
14
15     $functions =
16     '(?:sinh?|cosh?|tanh?|acosh?|asinh?|atanh?|exp|log(10)?|deg2rad|rad2deg
17     |sqrt|pow|abs|intval|ceil|floor|round|(mt_)?rand|gmp_fact)';
18     $operators = '[\/*^\+,-,]'; // Allowed math operators
19     $regexp =
20     '/^([+-]?('.$number.'|'.$functions.
21     '\s*\((?1)+\)|\((?1)+\))?(?:'.$operators.'(?1))?)+';
22
23     if (preg_match($regexp, $equation))
24     {
25         $equation = preg_replace('!pi|π!', 'pi()', $equation);
26
27         eval('$result = '.$equation.';');
28     }
29     else
30     {
31         $result = false;
32     }
33     return $result;
34 }
35 ?>

```

Figura 4.27

que, entre más grande sea el valor de  $n$ , el resultado de la suma es una mejor aproximación a la integral.

Una parte muy importante del programa, como ya se mencionó, es la evaluación de la función en diversos valores. Para tal efecto, existe una función<sup>2</sup> de PHP, que llamaremos `evaluacion.php` cuyo código se aprecia en la figura 4.27, la cual logra realizar las operaciones de una expresión cualquiera, escrita en términos de las funciones matemáticas de PHP. Por ejemplo, si introduce la expresión

<sup>2</sup>Función construida en base a la función `eval()` de PHP. Se encuentra publicada en forma libre en <http://php.net/manual/es/function.eval.php>

$$\$a=1/\text{sqrt}(3)$$

llamaremos a la función por medio de la instrucción `calc`, la cual producirá el valor `calc(\$a)= 0.5773`.

La instrucción `calc` nos resuelve el problema de la evaluación de la función a integrar.

Ahora escribamos el algoritmo del programa `integral.php`:

1. Se solicita en cajas de texto, la función  $f$  en la variable  $x$ , escrita en términos de las funciones PHP:

$$*, /, \text{pow}(), \text{log}(), \text{exp}(), \text{asin}(), \dots$$

además, los valores  $a, b$  del intervalo y el valor de  $n$ .

2. Se calcula  $\Delta x = \frac{b-a}{n}$
3. Empieza un ciclo `while` de 1 hasta  $n$
4. Se obtiene de manera recursiva los valores de  $x_i$ . Después, cambiará la letra  $x$  de la función por los valores particulares  $x_i$
5. Evalúa la función  $f$  en los  $x_i$  usando la función `evaluacion.php`, a través de la instrucción `calc`
6. Luego multiplica  $\Delta x$  por las respectivas evaluaciones de  $f$  y suma elemento a elemento hasta llegar hasta  $n$ .
7. Se presentan resultados.

Es importante decir que los valores tomados por el formulario, serán procesados por la misma página, `integral.php`.

En la imagen 4.28 se encuentra el código PHP que hará los cálculos. Inicialmente, la codificación no incluía las líneas, 91, 94 y 100, que se puede notar se refieren a la función exponencial `exp()`. El problema que tuvimos es, como ya se mencionó en el algoritmo, que al momento de cambiar la letra  $x$  de la función  $f$  por los valores particulares, también se cambiaba la  $x$  de la palabra `exp()` y se generaban errores. ¿Cómo lo resolvimos? El renglón 91 cambia la palabra `exp` por `EXP` y así el comando

```

84 <?php
85 if ( isset($_POST['enviado']) ){
86     include('evaluacion.php');
87     $f=$_POST['f']; $a=calc($_POST['a']); $b=calc($_POST['b']);
88     $n=$_POST['n']; $d=($b-$a)/$n;
89     $i=1;$s=0;
90     while($i<= $n){
91         if(substr_count($f, 'exp')<>0){$f= str_replace('exp','EXP',$f);}
92         $x= $a + ($d*$i);
93         $h=str_replace('x','.$x.', $f);
94         if(substr_count($h, 'EXP')<>0){$h= str_replace('EXP','exp',$h);}
95         $h= calc($h);
96         $int= $s+ ($h*$d);
97         $s=$int;
98     $i++;
99     }
100 $f= str_replace('EXP','exp',$f);
101 echo '<br> \(\ \displaystyle \int_{'. $a. '}'^{'. $b. '}} \::: \cdot dx
102 \approx \cdot $int.\)\<br>';
103 }
104 ?>

```

Figura 4.28

$f(x) = \text{pow}(x,2) / (\text{pow}(x,4) + \text{pow}(x,2) + 1)$   
 $(a,b) = ( 5 , 10 )$   
 $n = 200$   
 CALCULAR INTEGRAL

Figura 4.29

```
str_replace('x', '$x.', $f)
```

de la línea 93, que cambia “x” por \$x, no encuentra ninguna x minúscula en EXP y la deja igual. Queda clara la sensibilidad de PHP entre mayúsculas y minúsculas.

El renglón 94, regresa a minúsculas EXP, pues el comando `calc` de la línea 95, buscará las funciones PHP y evaluará las que encuentre. Así pues, para la comprensión del código, no tome mucho en cuenta las líneas 91, 94 y 100.

**Ejemplo 4.6.1.** Calculemos la integral definida:

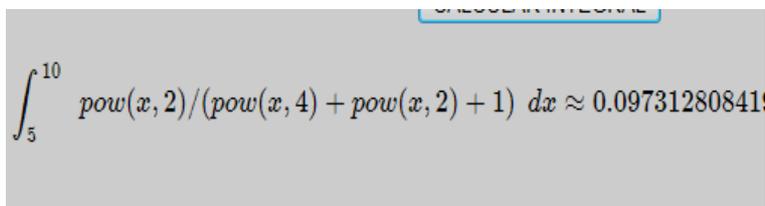
$$\int_5^{10} \frac{x^2}{x^4 + x^2 + 1} dx$$

El programa nos pedirá tres cosas: función, el intervalo y  $n$ . Elegiremos  $n = 200$ . En la figura 4.29, se muestra la forma de introducir los datos. Por su parte, la figura 4.30, muestra los resultados.

El problema original (vea [10, p. 396]), pide que se muestre la desigualdad

$$0 \leq \int_5^{10} \frac{x^2}{x^4 + x^2 + 1} dx \leq 0.1 \quad (4.1)$$

lo cual corresponde a lo obtenido por el software.



$$\int_5^{10} \text{pow}(x, 2) / (\text{pow}(x, 4) + \text{pow}(x, 2) + 1) dx \approx 0.09731280841$$

**Figura 4.30**

De pasada, demos la justificación matemática a la desigualdad 4.1, con el fin de hacer una diferenciación con el proceso computacional.

Tanto el polinomio del numerador como del denominador son positivos, por ser sumas de potencias pares, por ende

$$0 \leq \frac{x^2}{x^4 + x^2 + 1}$$

Por otro lado,  $x^4 + x^2 + 1 \geq x^4$ , de donde

$$\frac{1}{x^4 + x^2 + 1} \leq \frac{1}{x^4}$$

Si multiplicamos la desigualdad anterior por el número no negativo  $x^2$

$$\frac{x^2}{x^4 + x^2 + 1} \leq \frac{x^2}{x^4}$$

pero  $\frac{x^2}{x^4} = \frac{1}{x^2}$ . En resumen

$$0 \leq \frac{x^2}{x^4 + x^2 + 1} \leq x^{-2}$$

Ahora bien, integrando los miembros:

$$0 \leq \int_5^{10} \frac{x^2}{x^4 + x^2 + 1} dx \leq \int_5^{10} x^{-2} dx$$

donde  $\int_5^{10} x^{-2} dx = -\frac{1}{10} + \frac{1}{5} = 0.1$ . Con lo cual obtenemos la desigualdad 4.1

$f(x) = \sin(5*x)$   
 $(a, b) = ( 0 , \pi )$   
 $n = 200$   
 CALCULAR INTEGRAL

Figura 4.31

$\int_0^{3.1415926535898} \sin(5 * x) dx \approx 0.39979436209941$

Figura 4.32

**Ejemplo 4.6.2.** Nuestro programa, también acepta funciones trigonométricas. Calculemos:

$$\int_0^{\pi} \text{sen } 5x \, dx$$

Escribamos los datos como lo indica la figura 4.31. La solución de la integral se muestra en la figura 4.32.

**Ejemplo 4.6.3.** Calculemos la integral definida:  $\int_0^{\frac{3\sqrt{3}}{2}} \frac{x^3}{(4x^2 + 9)^{\frac{3}{2}}} dx$ .

Aquí la salida del programa:

$\int_0^{2.5980762113533} \text{pow}(x, 3) / \text{pow}(4 * \text{pow}(x, 2) + 9, 3/2) dx \approx 0.093760$

Lo cual coincide con el resultado de la integral ubicado en [10, p. 482].

Finalmente, es bueno decir que probamos con diferentes combinaciones de funciones, ya sean trigonométricas, exponenciales, logarítmicas, polinómicas y se lograron resultados satisfactorios.

# Conclusiones

En esta tesis se han proporcionado elementos de programación para la construcción de páginas web con PHP y  $\text{\LaTeX}$ . A través varios ejemplos ilustrativos se mostraron las potencialidades de la combinación de lenguajes:

- (a) PHP operacionalmente y
- (b)  $\text{\LaTeX}$  como presentador de resultados tipográficamente excelentes.

Es interesante mencionar que PHP tiene las mismas capacidades que cualquier lenguaje de programación. Tres ventajas que evidentemente posee:

- (a) La diversidad de objetos que puede manejar, desde imágenes hasta bases de datos.
- (b) Su capacidad de ejecución en internet a través de una página web.
- (c) No es necesario instalar, para su ejecución, complementos adicionales, no importando el sistema operativo.

Es de esperarse que muchos procesos matemáticos puedan automatizarse de esta manera: operaciones de fracciones, leyes exponentes, binomio de newton, inversa y diagonalización matricial, determinantes, solución de ecuaciones, integración y derivación numérica etc. Desde lo más simple hasta lo más complejo. Además, no es exclusivo de matemáticas. Es factible crear software que genere exámenes en línea de cualquier materia, enviar correos, que guarde datos, etc.



# Bibliografía

- [1] Bravo G. y Gutiérrez A. (2004). *PHP4 a través de ejemplos*. México: Alfaomega
- [2] Cibelli C. (2012). *PHP programación avanzada para profesionales*. Buenos aires, Argentina: Alfaomega.
- [3] *Historia de PHP*. Recuperado el 31 octubre de 2014, de <http://php.net/manual/es/history.php.php>
- [4] Luna Carreto G. (sf). *La inversa de una matriz por el método de gauss-jordan*. Recuperado el 31 octubre de 2014, de <http://matematicasfce.netii.net/inversa0.php>
- [5] Luna Carreto G. (sf). *Programa que ortogonaliza bases de  $R^n$* . Recuperado el 8 noviembre de 2014, de <http://matematicasfce.netii.net/gram.php>
- [6] Luna Carreto G. (sf). *Producto punto en  $R^n$* . Recuperado el 8 noviembre de 2014, de <http://matematicasfce.netii.net/productopunto.php>
- [7] Oetiker T., Partl H., et al. (2010). *La introducción no-tan-corta a  $\LaTeX$  2<sub>ε</sub> o  $\LaTeX$  en 127 minutos*. Versión electrónica.
- [8] *¿Qué es MathJax?* (2014). Recuperado el 25 de octubre de 2014, de <http://docs.mathjax.org/en/latest/mathjax.html>
- [9] Spivak M. (1996). *Cálculo infinitesimal*.(2a ed.). México: Reverté.
- [10] Stewart J. (2013). *Cálculo trascendentes tempranas*.(7a ed.). México: Cengage.
- [11] *Start with wampserver*. (sf). Recuperado el 25 de octubre de 2014, de <http://www.wampserver.com/en/>