



PROGRAMACIÓN

WEB

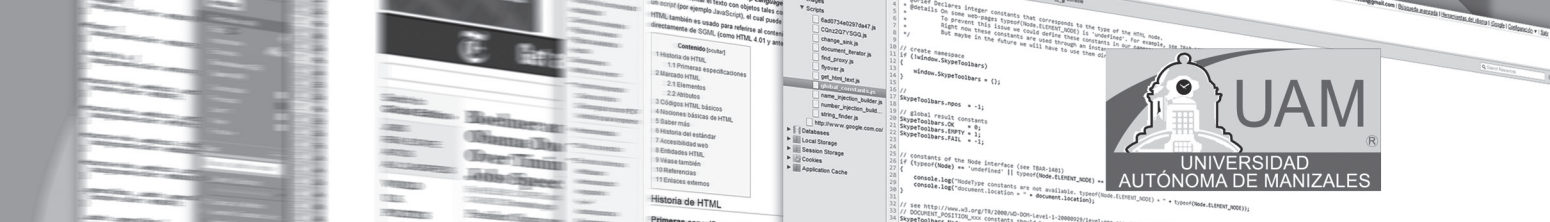


Contenido

CARTA DE PRESENTACIÓN	9
CARTA DESCRIPTIVA DE LA ASIGNATURA	10
CONTENIDO PROGRAMÁTICO	12
MAPA CONCEPTUAL DE CONTENIDOS	12
BIBLIOGRAFIA BÁSICA CONSULTADA	13
SESION NÚMERO UNO	
HTML BASICO	15
1.1. OBJETIVOS ESPECÍFICOS	15
1.2. DESARROLLO TEMÁTICO	15
1.2.1. HTML (HyperTextMarkupLanguage)	15
1.2.2. Inicio de HTML	15
1.2.3. Código básico de una página	16
1.2.4. Implementación de algunos programas haciendo uso de los tags básicos	16
1.2.5. Otras tags importante que se usan regularmente al hacer páginas web	20
1.3. CONCLUSIONES	25
SESION NÚMERO DOS	
HTML INTERMEDIO	27
2.1. OBJETIVOS ESPECÍFICOS	27
2.2. DESARROLLO TEMÁTICO	27
2.2.1. Texto preformateado	27
2.2.2. Acentos y otros caracteres especiales	28
2.2.3. Tabla de caracteres importantes	29
2.2.4 LISTAS	30
2.2.5. Como hacer comentarios en html	32
2.2.6. Tablas con HTML	32
2.2.7. Otros atributos importantes para tener en cuenta con tablas	34
2.3. CONCLUSIONES	35
SESIÓN NÚMERO TRES	
HTML AVANZADO	37
3.1. OBJETIVOS ESPECÍFICOS	37
3.2. DESARROLLO TEMÁTICO	37
3.2.1. Formularios en HTML	37
3.2.2. CGI o Common Gateway Interface	37
3.2.3. La elaboración de una plantilla consta de dos fases:	37
3.2.4. Marcos	43
3.2.5. La directiva <FRAME>	44
3.2.6. Capas	47
3.3. CONCLUSIONES	49

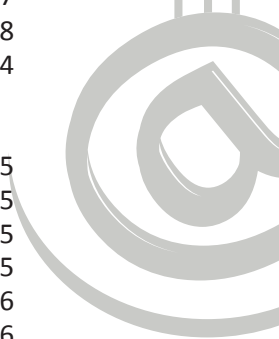


SESION NÚMERO CUATRO	
PÁGINAS CON ESTILO Y VALIDACIÓN DE FORMULARIOS CON JAVASCRIPT	51
4.1. OBJETIVOS ESPECÍFICOS	51
4.2. DESARROLLO TEMÁTICO	51
4.2.1. Hojas con estilo	51
4.2.2. Redefinición de etiquetas	52
4.2.3. Separar estilos	53
4.2.4. Validación de formularios con javascript	55
4.3. CONCLUSIONES	62
SESION NÚMERO CINCO	
INCIANDO EL LENGUAJE DE PROGRMACION PHP	63
5.1. OBJETIVOS ESPECÍFICOS	63
5.2. DESARROLLO TEMÁTICO	63
5.2.1. ¿Qué es PHP?	63
5.2.2. Trabajemos con PHP	64
5.2.3. Instalación del servidor de aplicaciones web para PHP	64
5.2.4. Variables	65
5.2.5. Operadores. En php se tienen los siguientes operadores:	67
5.3. CONCLUSIONES	71
SESION NÚMERO SEIS	
CONTINUANDO CON EL LENGUAJE PHP	73
6.1. OBJETIVOS ESPECÍFICOS	73
6.2. DESARROLLO TEMÁTICO	73
6.2.1. Condicionales	73
6.2.2. Bucles o LOOPS	77
6.2.3. Formato de salida para cadenas de caracteres o números	80
6.2.4. Manejo de cadenas	81
6.3. CONCLUSIONES	83
SESION NÚMERO SIETE	
PROGRAMACIÓN ORIENTADA A OBJETOS CON PHP	85
7.1. OBJETIVOS ESPECÍFICOS	85
7.2. DESARROLLO TEMÁTICO	85
7.2.1. El contexto.	85
7.2.2. Conceptos básicos de objetos	85
7.2.3. Características bases de la POO.	86
7.2.4. Atributos y métodos de la clase	89
7.2.5. Características de los constructores	91
7.2.6. Herencia	92
7.2.7. Polimorfismo	93
7.3. CONCLUSIONES	96
SESION NÚMERO OCHO	
RELACION ENTRE EL LENGUAJE PHP Y LAS BASES DE DATOS	97
8.1. OBJETIVOS ESPECÍFICOS	97
8.2. DESARROLLO TEMÁTICO	97
8.2.1. CREANDO LA BASE DE DATOS	97



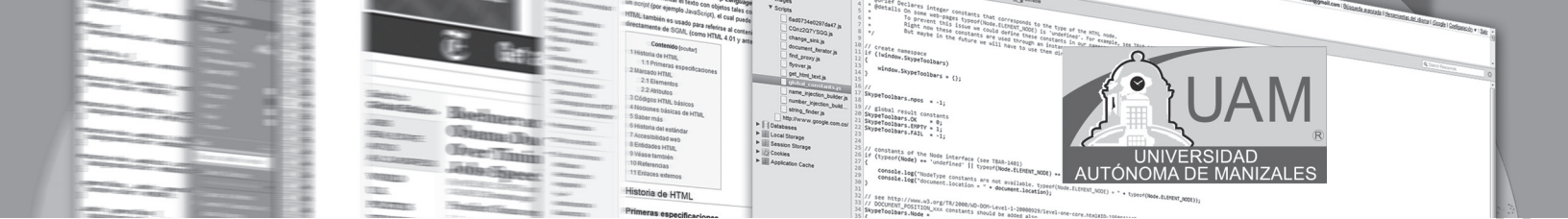
Universidad Autónoma de Manizales

8.2.2. Conexión a la base de datos desde PHP	100
8.2.3 CONSULTA A LA BASE DE DATOS	101
8.2.4. Insertando registros	102
8.2.5. Borrando registros	103
8.2.6 Autenticación	104
8.3 CONCLUSIONES	105
SESION NÚMERO NUEVE	
MANEJO DE SESIONES CON PHP	107
9.1. OBJETIVOS ESPECÍFICOS	107
9.2. DESARROLLO TEMÁTICO	107
9.2.1. SESIONES	107
9.2.2 APLICACIÓN DE LAS SESIONES EN FORMULARIOS DE LOGUEO.	108
9.3. CONCLUSIONES	114
SESION NÚMERO DIEZ	
PUBLICACIÓN EN LA WEB	115
10.1. OBJETIVOS ESPECÍFICOS	115
10.2. DESARROLLO TEMÁTICO	115
10.2.1. Página inicial de un desarrollo web	115
10.2.2. Hosting para residencia de un desarrollo web	116
10.2.3. Dominio	116
10.2.4. Subdominio	117
10.2.5. Encontrando un sitio de residencia web (hosting)	117
10.2.6. Contacto con la empresa prestadora del servicio	118
10.2.7. Subiendo archivos al sitio web	121
10.3. CONCLUSIONES	125
11. ANEXO DE INSTALACIONES	127
11.1. OBJETIVOS ESPECÍFICOS	127
11.2. DESARROLLO TEMÁTICO	127
11.2.1. Programas incluidos en el cd	127
11.2.2. Instalando notepad++	127
11.2.3. Instalando el Happdit	131
11.2.4. Instalando el APPSERV	132
11.2.5. Instalando FILEZILLA	135



```
51 DOCUMENT_POSITION_PRECEDING : 0x0;
52 DOCUMENT_POSITION_FOLLOWING : 0x0;
53 DOCUMENT_POSITION_CONTAINS : 0x0;
54 DOCUMENT_POSITION_CONTAINED_BY : 0x0;
55 DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC : 0x0;
56 }
57
58 // redefine setAttribute function to emulate DOM2 modified num
59 // http://bugzilla.mozilla.org/show_bug.cgi?id=4911
60 Element.prototype.setAttribute = Element.prototype.setAttribute
61 Element.prototype.setAttribute = function(name, val)
62 {
63     // save the previous value
64     var prev = this.getAttribute(name);
65 }
```



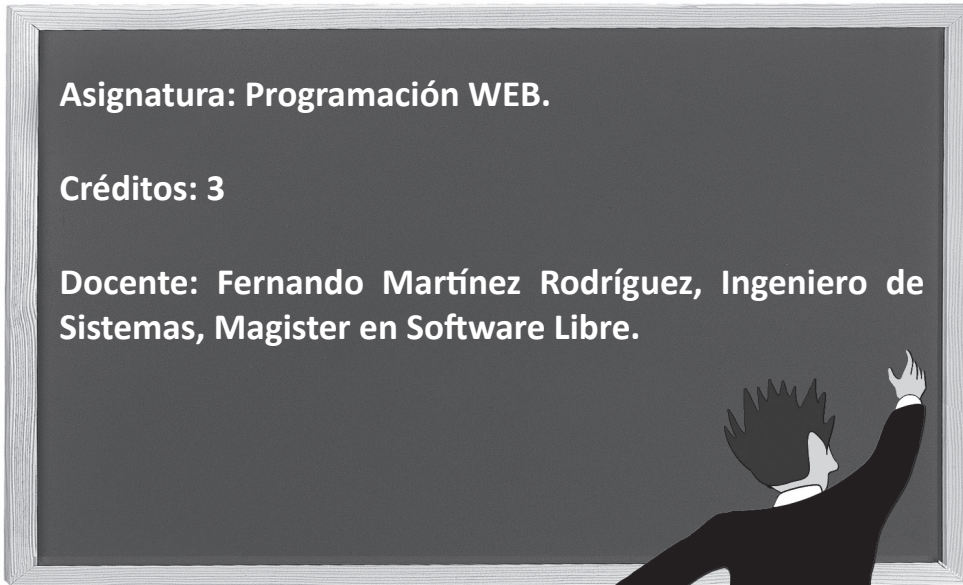


Universidad Autónoma de Manizales

Asignatura: Programación WEB.

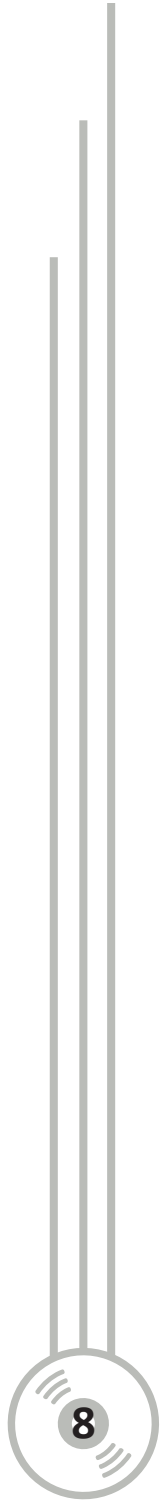
Créditos: 3

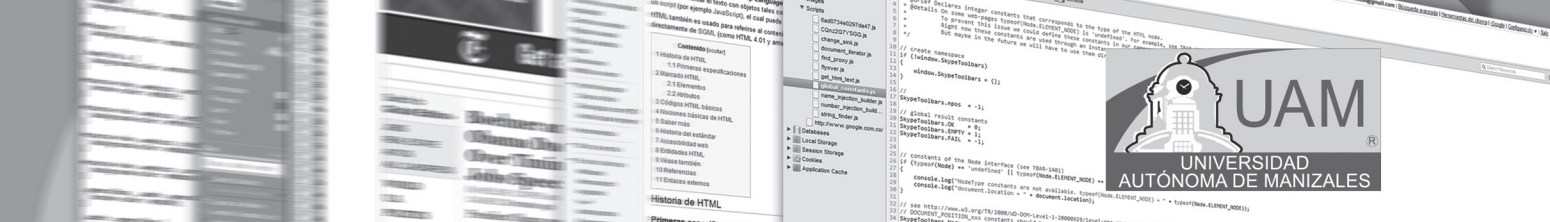
Docente: Fernando Martínez Rodríguez, Ingeniero de Sistemas, Magister en Software Libre.



Programación WEB

```
51 DOCUMENT_POSITION_PRECEDING : 0x0;
52 DOCUMENT_POSITION_FOLLOWING : 0x0;
53 DOCUMENT_POSITION_CONTAINS : 0x0;
54 DOCUMENT_POSITION_CONTAINED_BY : 0x0;
55 DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC : 0x0;
56 }
57
58 // redefine setAttribute function to emulate DOM2 modified num
59 // http://bugzilla.mozilla.org/show_bug.cgi?id=4911
60 Element.prototype.setAttribute = Element.prototype.setAttribute
61 Element.prototype.setAttribute = function(name, val)
62 {
63     // save the previous value
64     var prev = this.getAttribute(name);
65 }
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```





Universidad Autónoma de Manizales

CARTA DE PRESENTACIÓN

Estimado Estudiante.

El desarrollo web es un campo de la programación relativamente joven, comparado con los otros tipos de programación. Este es uno de los campos de acción más requeridos en la actualidad para tecnólogos e ingenieros, ya que mediante la misma se crea buena parte del entorno en la red de redes, es decir de Internet.

El diseño y desarrollo web ha venido revolucionado la forma como las empresas presentan su imagen comercial y su objeto de negocio. Son millones las páginas web publicadas, todas ellas con uno u otro estilo; lo anterior ha vendido generando enormes y variadas líneas de acción, sobre las que puede moverse en el campo laboral un tecnólogo o ingeniero. La dinámica con la que este tipo de desarrollo evoluciona día a día, ha cambiado el rol clásico del diseñador y programador web, a los cuales se les presenta a diario un desafío: el de estar a la vanguardia de los cambios, y acciones laborales por desarrolla; ya sea frente a la presentación a nivel mundial de una determinada empresa respecto a la imagen de la misma y de su objeto de negocio, o a la generación de comercio electrónico entre otros campos de acción que se pueden desarrollar. La programación web, ha permitido que muchas transacciones comerciales, bancarias, de salud, etc. Se puedan realizar usando medios electrónicos, principalmente potenciando y masificando el uso de la red: Internet.

Debido a la importancia y crecimiento de Internet como una herramienta para la transmisión y recepción de la información, hace que el uso y comprensión de herramientas para el desarrollo web, sea una necesidad sentida para el tecnólogo e ingeniero de hoy. Es imperativo que deban aprender los aspectos básicos, medios y avanzados del desarrollo web, dado que esto les hará más competitivos en el mundo laboral, pues contarán con capacidades apetecidas por múltiples empresas, que ofrecen sus servicios por medio de la web, para ampliar la cobertura de su objeto de negocio por medio del comercio electrónico; además de ayudar en la captura y procesamiento de datos de los clientes y demás actividades que se hacen vía web. Adicionalmente el Tecnólogo e Ingeniero que sigan esta línea de trabajo podrán desarrollarse como empresarios en varias especialidades del desarrollo web permitiendo de esta forma aumentar la autonomía laboral propia de un profesional de Tecnología e Ingeniería de la Universidad Autónoma de Manizales.

Esta asignatura pretende poner al Tecnólogo en análisis y programación de sistemas de información en la ruta de desarrollo web, buscando aumentar sus competencias, frente a posibilidades de negocio que muestra bastantes puertas abiertas, bajo n posibilidades que fomentarán la creatividad del estudiante, como proyección de su vida profesional.

¡BIENVENIDOS!



Programación WEB

CARTA DESCRIPTIVA DE LA ASIGNATURA

Introducción

El curso estará estructurado en diez (10) sesiones de aprendizaje; cada uno de sus elementos se describe como parte del contenido programático. En cada sesión encontrará los conceptos fundamentales trazados por los objetivos que se quieren alcanzar en ella. El material ha sido escrito como continuidad de las asignaturas de Programación, de tal forma que encontrará a lo largo del módulo la necesidad de recordar y aplicar los tópicos referentes a las mismas.

Estrategia de Aprendizaje

El módulo ha sido pensando bajo la **Metodología Sinfónica Virtual**, la cual persigue, tener una conversación dialógica entre todas las herramientas de aprendizaje ofrecidas por la Universidad Autónoma de Manizales en el modelo de educación a distancia. Para lograr alcanzar esto, es importante que siga el módulo rigurosamente y de forma sistemática, pues en él encontrará en todo momento que le será exigida la realización de actividades, laboratorios, consultas, etc. Para poder llamar su atención se han colocado íconos de conocimiento que tienen un significado especial al interior de lo escrito en el módulo; estos íconos son:



Pare

PARE. Este ícono pretende indicarle a Usted, la importancia de retomar conceptos vistos en algún módulo anterior y por ende es importante retomarlos para entender mucho mejor el nuevo conocimiento. Este ícono pretende también, hacer notar aspectos importantes de la temática tratada, es decir resaltar aspectos de bastante importancia.



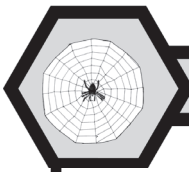
Consultemos

CONSULEMOS. Este ícono le indicará que debe buscar información sobre una temática específica. Una vez que Usted tenga la información compilada deberá ingresar al Aula Virtual y buscar el link marcado como CONSULTA para enviarla en un documento adjunto. En estos documentos se debe visualizar su nivel de análisis frente a la temática investigada.



Actividad

ACTIVIDAD. Este ícono le indicará que debe realizar un trabajo adicional, generalmente lectura y a partir de ésta realizar la ACTIVIDAD PROPUESTA en el Aula Virtual, dependiendo de la sesión donde se encuentre trabajando en ese momento.



Practicando el desarrollo WEB

PRACTICANDO EL DESARROLLO WEB. El ambiente de desarrollo que utilizaremos en esta asignatura tiene varios aspectos. Estos son: Notepad++, HapEdit, Appserv (Apache, php y Mysql); al interior del material escrito encontrará gran cantidad de ejemplos escritos en html, Javascript y php; estos ejemplos están explicados en el módulo, pero Usted deberá buscarlos en el apartado denominado: EJEMPLOS DE LA SESIÓN, ubicados en el Aula Virtual y digitarlos y compilarlos por su cuenta. El Objetivo de esto es que Usted observe detenidamente que es lo que está ocurriendo en la teoría frente a la práctica de cada temática tratada.



Aplicando lo aprendido

APLICANDO LO APRENDIDO. Esta asignatura se aprende analizando algoritmos, diseñando algoritmos y realizando programas. No nos podemos quedar únicamente con los ejemplos y la teoría entregada; por lo anterior creemos, por experiencia, que cualquier asignatura que sirva de base para la programación se aprende: REALIZANDO LABORATORIOS. Por tanto Usted deberá ir al Aula Virtual y buscar, en el apartado denominado LABORATORIO DE SESIÓN, el enunciado de los ejercicios propuestos por el docente para aplicar lo visto en la sesión. Una vez realizados deberán ser enviados como adjuntos en el lugar indicado de cada apartado.



En el Aula Virtual

EN EL AULA VIRTUAL. Este será el punto donde se centralizarán todos los materiales educativos; encontrará otra serie de materiales adicionales al que encuentra en este módulo escrito que buscan ahondar en temáticas que no se han escrito aquí, pero que son importante en su proceso de aprendizaje de la asignatura.

Adicional al material escrito, recibirá un CD con el *software* necesario para avanzar en esta asignatura y obtener los éxitos deseados en el desarrollo la misma.

De igual forma durante el desarrollo de este módulo, se podrá contar con las sesiones de teleclases al igual que sesiones de Video Conferencias para nuestras tutorías; estas sesiones son utilizadas para encontrarnos en tiempo real y construir juntos en conocimiento de la asignatura, en la búsqueda del aumento de espectro de conocimiento.

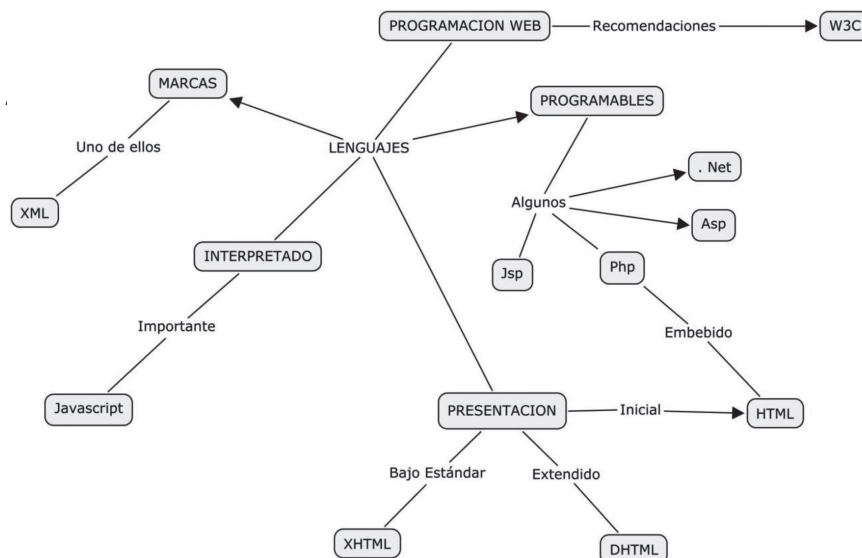
Es importante que se apropie de todos los contenidos, de las prácticas, de los laboratorios, consultas, actividades, investigaciones, foros, tutorías virtuales, etc. Para alcanzar los logros propuestos. No dude en preguntar, trataremos por todos los medios que Usted logre un nivel bastante competente en esta asignatura denominada: Programación Web.



CONTENIDO PROGRAMÁTICO

Sesión	Contenidos a Tratar.
1	Html básico: Esta sesión describe los aspectos iniciales del lenguaje de programación marcado de Hipertexto.
2	Html intermedio: En esta sesión se pretende ahondar sobre este lenguaje, avanzando en el manejo de tablas y listas.
3	Html avanzado: En esta sesión ahondamos en el manejo de formularios para el manejo de información de usuarios
4	Páginas con estilo y Validación de formularios con Javascript: Esta sesión ahonda el aspecto de los formularios validados con el lenguaje de programación interpretado Javascript.
5	Iniciando el lenguaje de programación php: Esta sesión estudia los aspectos básicos del lenguaje de programación php.
6	Continuando con el lenguaje php: en esta sesión se estudian los procesos de decisión y bucles en php.
7	Programación orientada a objetos con php: esta sesión enfoca todo lo del paradigma de POO, redireccionándolo a un lenguaje web como lo es php versión 5.0
8	Relación entre el lenguaje php y las bases de datos: Esta sesión retoma los aspectos básicos de las bases de datos y hace una relación directa con el lenguaje de programación php.
9	Manejo de sesiones con php: Esta sesión permite mostrar el manejo de sesiones para el logueo de usuarios en el ingreso de un desarrollo indicado.
10	Publicación en la web: Esta sesión se enfoca al aspecto de poder publicar desarrollos web en un Hosting gratuito o pago.
11	Anexo de instalaciones: Este muestra los aspectos pormenorizados para instalar los distintos paquetes de software que se usan al interior de todo el módulo.

MAPA CONCEPTUAL DE CONTENIDOS





Actividad

Ingrese al AULA VIRTUAL, busque la pestaña marcada con: "INICIO", ubique dentro de ella la ACTIVIDAD INTRODUCTORIA, la encontrará al final. Ingrese al FORO SOCIAL y haga su presentación personal, para tener el gusto de conocerl@.

BIBLIOGRAFIA BÁSICA CONSULTADA

- Becerra Santamaría Becerra. (2009). Lenguaje PHP y la Tecnología AJAX. Editorial por computador. Editorial Computec – Rama. Alfaomega.
- Pavón Puertas, Jacobo (2006). Creación de un portal con PHP y MYSQL. Editorial Alfaomega – Rama.
- Polanco, Juan. (1999). PERL páginas web interactivas. Editorial Computec – Rama. Alfaomega.
- Powell, Thomas A.– Mc Graw Hill. (1999) Manual de Referencia HTML. Buenos Aires. Osborne. Capítulo 2.
- Soria, Ramón. (1998.). Navegar en Internet HTML, diseño y creación de páginas web.
- Talens Oliag, Sergio y Hernández Orallo, José. (1996). HTML. Manual de Referencia. Madrid: Editorial Paraninfo.
- LA WEB DEL PROGRAMADOR. <http://www.lawebdelprogramador.com/>
- MEMORIAS DE UN APRENDIZ DE PHP. <http://www.rinconastur.net/php/php35.php>
- PHP YA. <http://www.phpya.com.ar/>
- WEB ESTILO. <http://www.webestilo.com/>
- BEYET. INTERNET SERVICES. <http://byethost.com/>

```
51 DOCUMENT_POSITION_PRECEDING : 0x04,
52 DOCUMENT_POSITION_FOLLOWING : 0x08,
53 DOCUMENT_POSITION_CONTAINS : 0x10,
54 DOCUMENT_POSITION_CONTAINED_BY : 0x20,
55 DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC : 0x40
56 }
57
58 // redefine setAttribute function to emulate DOM2 modified num
59 // http://bugzilla.mozilla.org/show_bug.cgi?id=4911
60 Element.prototype.setAttribute = Element.prototype.setAttribute
61 Element.prototype.setAttribute = function(name, val)
62 {
63     // save the previous value
64     var prev = this.getAttribute(name);
65 }
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```



1.1. OBJETIVOS ESPECÍFICOS

- Reconocer los aspectos básicos que hacen parte del lenguaje HTML.
- Aplicar los tags básicos de HTML en la realización de páginas web estáticas.
- Identificar las distintas posibilidades de aplicación de los tags que forman parte del lenguaje HTML.
- Caracterizar los tags de HTML que hacen cambios a textos y colores.
- Clasificar los tags de HTML que manejan imágenes y vínculos a otras páginas.

1.2. DESARROLLO TEMÁTICO

1.2.1. HTML (HyperTextMarkupLenguaje)

Es el lenguaje de programación utilizado para crear las páginas web de cualquier sitio. Básicamente se trata de una especie de editor de texto, como el Word, por ejemplo. La principal diferencia es que con el HTML hemos de utilizar lo que se denominan **etiquetas o tags**, para formatear el texto, imágenes, etc.

1.2.2. Inicio de HTML

Para escribir programas de HTML, se tienen múltiples posibilidades. Una es abrir Block de notas (o algún editor de texto) y escribir el código allí. La otra forma es utilizar algún editor de HTML; en nuestro caso usaremos Notepad++, dada la facilidad de uso y sus altas posibilidades para mantener el rastreo sintáctico del código y permitir la posibilidad de verificar rápidamente si el código está bien escrito. (**Ver anexo de instalaciones: Instalando Notepad++**)

Al ingresar al Notepad++ se abre un archivo nuevo y seleccionado en Lenguaje por la H, el marcado como HTML. Como se muestra en la figura 1.

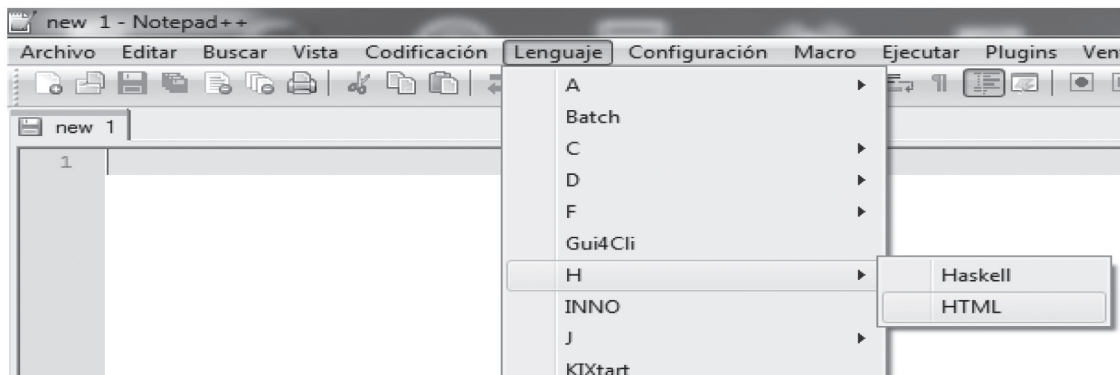


Figura 1. Creando un archivo HTML en Notepad++

SESION NÚMERO UNO HTML BÁSICO

1.2.3. Código básico de una página

HTML trabaja con unas etiquetas especiales, a las que denominamos TAGS. Todos los tags se escriben entre los símbolos: < y >. El cuerpo básico de un programa realizado en HTML es:

```

<html>
<head>
<title></title>
</head>

<body>

</body>
</html>

```

Tengamos presente algunos aspectos importantes al realizar un programa en HTML.

1.2.3.1. Punto A. Todas las páginas HTML comienzan con el código de inicio <html> y terminan con el código </html>

Así que ponemos:

```

<html>
</html>

```

1.2.3.2. Punto B. Las páginas HTML se dividen en un encabezado y un cuerpo. Para el encabezamiento tenemos que poner los "tags": <head> y </head>

Tenemos que ponerlo dentro de las instrucciones inicial y final de página (paso A), por lo tanto nos queda:

```

<html>
<head>
</head>
</html>

```

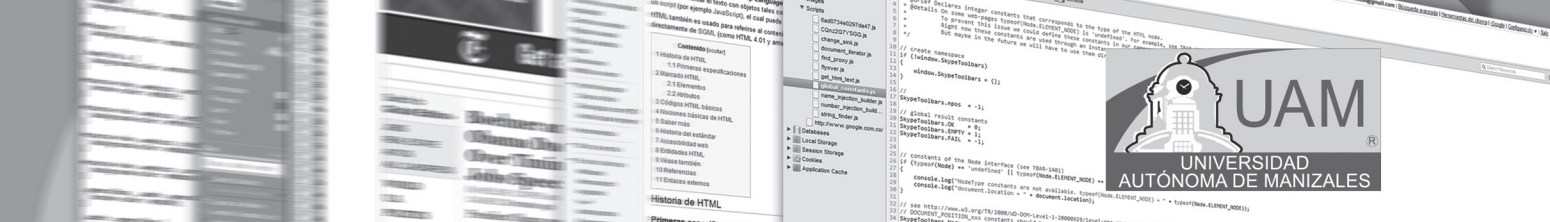
1.2.3.3. Punto C. Igualmente insertamos las instrucciones para el cuerpo de la página <body> y </body>. Con lo que nos queda:

```

<html>
<head>
</head>
<body>
</body>
</html>

```

1.2.4. Implementación de algunos programas haciendo uso de los tags básicos



Universidad Autónoma de Manizales

1.2.4.1. Ingresando título de la página

Ingrese a Notepad++ y digite el código que se muestra en la figura 2. (en el Browser)

```
S1_ejemplo1.html
1 <html>
2 <head>
3 <title> MI PRIMERA PAGINA </title>
4 </head>
5 <body>
6 </body>
7 </html>
```

Figura 2. Título en el Browser

Grabe el archivo teniendo presente que desde el inicio, se le indicó al sistema que el lenguaje del código es HTML. Observe la figura 3.

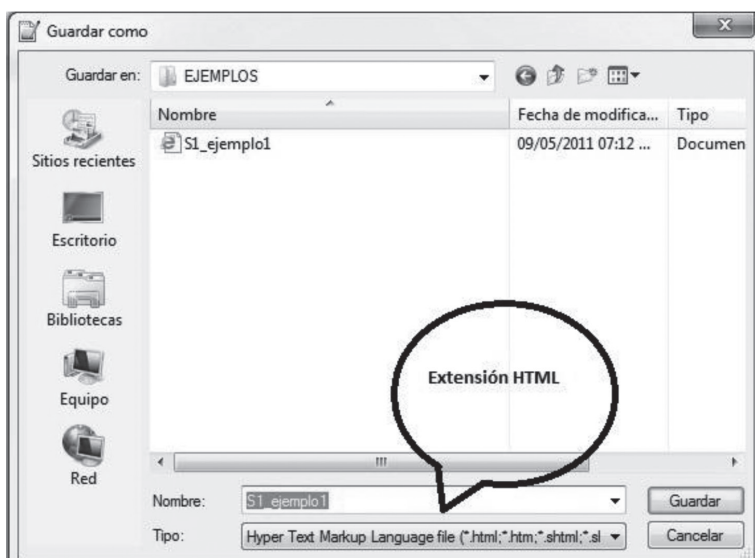


Figura 3. Guardando programas HTML

¡IMPORTANTE!

Para ver el resultado que se obtiene en cualquier programa, siga los siguientes pasos. Seleccione del menú de herramientas: Ejecutar y seleccione el navegador donde lo desea visualizar. No olvide que estamos programando para la WEB y todo se visualiza en los navegadores: IE=Internet Explorer; Firefox; Safari; Chrome, etc. En la figura 4, seleccionamos IE.

SESION NÚMERO UNO HTML BÁSICO

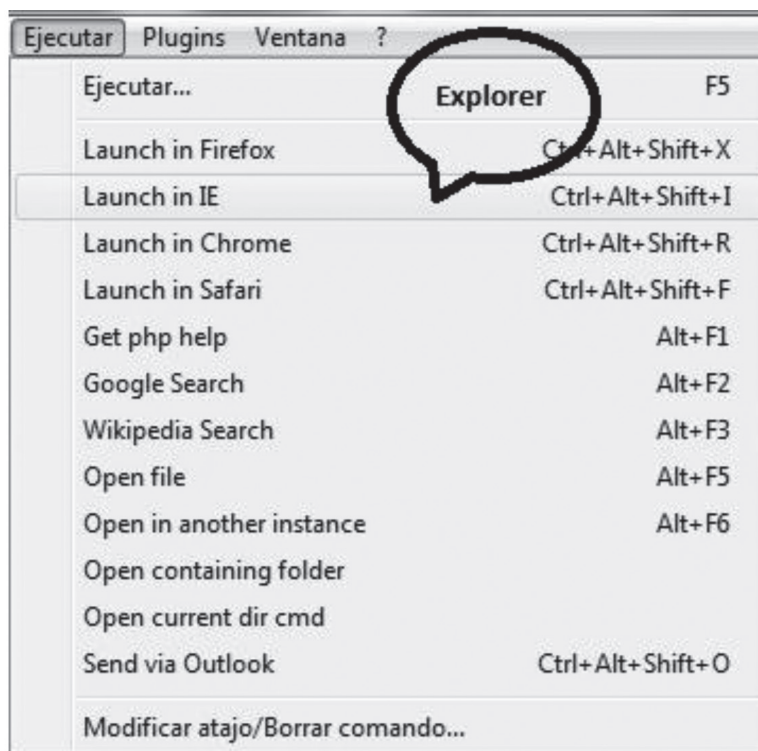


Figura 4. Mostrar resultado en Internet Explorer

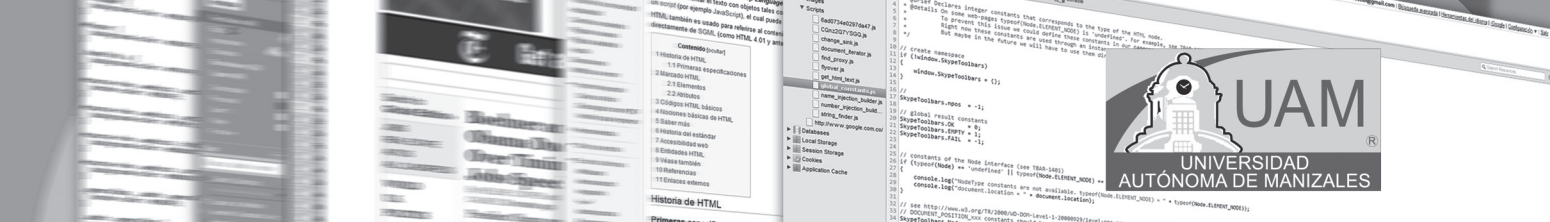
El resultado que se obtiene en este caso es el título de la página de navegación en el Browser. Ver figura 5.



Figura 5. Muestra el Título

1.2.4.2. Ingresando texto en la página

En el siguiente ejemplo, lo que se pretende es mostrar algo dentro de la página y no solamente el Título. Abrimos un nuevo proyecto en Notepad++ y escribimos el código HTML. Ver figura 6.



Universidad Autónoma de Manizales

```
S1_ejemplo1.html S1_ejemplo2.html  
1 <html>  
2 <head>  
3 <title> MI PRIMERA PAGINA </title>  
4 </head>  
5 <body>  
6     Aquí escribo lo que deseo... HELLO WORLD!!!!  
7 </body>  
8 </html>
```

Figura 6. Código del ejemplo 2. Texto en el cuerpo

La salida de este programa al correrlo se puede visualizar en la figura 7.



Figura 7. Salida del cuerpo del programa

1.2.4.3. Centrando texto en la página

En la figura 8, se muestra el código que se escribe para centrar un letrero al interior de una página; en este caso se usa el tag: <center>; si quisiéramos alinear a la izquierda usamos el tag: <left> y si deseamos alinear a la derecha se usa el tag: <right>. La figura 9 muestra el texto en salida al centro de la página.

```
S1_ejemplo1.html S1_ejemplo2.html S1_ejemplo3.html  
1 <html>  
2 <head>  
3 <title> MI PRIMERA PAGINA </title>  
4 </head>  
5 <body>  
6     <center> HELLO WORLD!!!! </center>  
7 </body>  
8 </html>
```

Figura 8. Código para centrar texto



SESION NÚMERO UNO HTML BÁSICO



Figura 9. Salida por pantalla del código centrado

1.2.4.4. Mostrar texto en negrita

En muchas ocasiones se hace necesario resaltar el texto; lo que comúnmente denominamos: texto en negrilla. Para realizar esto usamos el tag: o ; los dos cumplen el mismo objetivo; ver figura 10. Para ver la salida por pantalla del código digitado observe la figura 11.

```

1 <html>
2 <head>
3 <title> MI PRIMERA PAGINA </title>
4 </head>
5 <body>
6 <center> <strong> HOLA MUNDO!!!! </strong> </center>
7 <!--O PUEDE SER ASÍ -->
8 <center> <B> HOLA MUNDO!!!! </B> </center>
9 </body>
10 </html>
  
```

Figura 10. Colocando negrilla al texto



Figura 11. Texto en negrilla

1.2.5. Otras tags importante que se usan regularmente al hacer páginas web

1.2.5.1. Poner el texto en cursiva

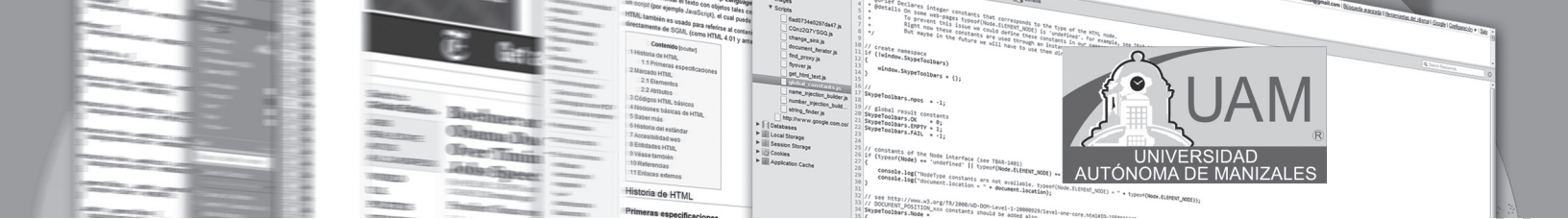
Para ponerlo en cursiva utilizamos “i” (de italian). Esto se logra colocando un tag: <i>texto en cursiva</i>. No olvide: por normas estándar todo tag que se abre, se debe cerrar.

1.2.5.2. Poner el texto subrayado

Para subrayado utilizamos “u” (underline). Para realizar esto usamos el tag: <u>subrayado</u>. Generalmente se subraya parte de la frase.

1.2.5.3. Cambiando color de background

En ocasiones es necesario cambiar el color del fondo de la página; para lograr este objetivo se usa el tag: en el cuerpo <bgcolor>. Es importante hacer claridad que los colores se pueden llamar



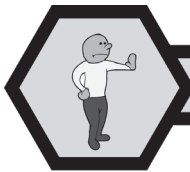
Universidad Autónoma de Manizales

con su nombre en inglés; por ejemplo, si deseamos color azul escribiríamos: blue. Esta práctica no es recomendable, dado que la norma estándar para que cualquier navegador publique lo que se desea fielmente indica que se debe escribir en el código internacional de programación WEB. La figura 12, muestra el código de algunos de los colores básicos.

aliceblue (#F0F8FF)	antiquewhite (#FAEBD7)	aqua (#00FFFF)	aquamarine (#7FFFD4)
azure (#F0FFFF)	beige (#F5F5DC)	bisque (#FFE4C4)	
blanchedalmond (#FFEBCD)	blue (#0000FF)	blueviolet (#8A2BE2)	brown (#A52A2A)
burlywood (#DEB887)	cadetblue (#5F9EA0)	chartreuse (#7FFF00)	chocolate (#D2691E)
coral (#FF7F50)	cornflowerblue (#6495ED)	cornsilk (#FFF8DC)	crimson (#DC143C)

Figura 12. Algunos códigos de colores para HTML

Observe que según esta tabla el color azul será: #0000FF; este código garantiza que cualquier navegador muestre el mismo color.



Pare

Si desea obtener un color específico, existen en la red de la internet gran cantidad de sitios que ayudan a encontrar el código respectivo; uno de ellos está en la dirección:

<http://html-color-codes.info/codigos-de-colores-hexadecimales/>

Veamos un ejemplo donde se cambia el color del fondo de la página. La figura 13 muestra el código digitado en Notepad++; Este ejemplo muestra también dos nuevos tags. Uno permite cambiar el tamaño de la letra; este es el tag: ; el tamaño va a criterio de autor de la página web; el tamaño va desde **1 hasta 7 (con modificaciones de + y -)**. El otro permite alinear un texto; este es el tag: <div align="center"> No olvide que la alineación puede ser: **right, left, center, justify**.

```

1  <html>
2  <head>
3  <title> MI PRIMERA PAGINA </title>
4  </head>
5  <body bgcolor="#CCCCCC">
6  <div align="center"> <font size="5">CAMBIANDO TAMAÑO</font></div>
7  </body>
8  </html>

```

Figura 13. Cambio de color



SESION NÚMERO UNO HTML BÁSICO

La salida de este código se muestra en la figura 14. Observe el cambio de tamaño de la letra y el color de fondo que ha pasado de blanco a gris.

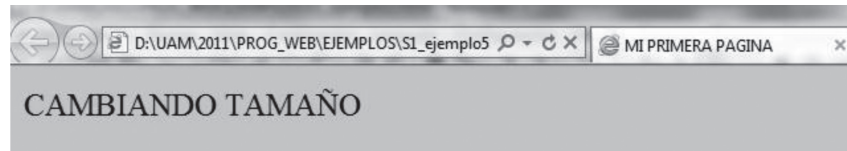
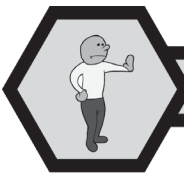


Figura 14. Cambiando tamaño de letra y color del fondo



Pare

La instrucción para poner color a las letras es

`aquí el texto al que queremos dar color`

Por EJEMPLO: Para poner el texto en rojo `este texto se pone rojo`. Para Color azul = Blue. No olvide utilizar mejor el código de colores estándar, visto anteriormente.

1.2.5.5. Poner titulares

Los titulares o encabezamiento de los textos son otra forma de modificar el tamaño del texto. La orden para encabezamiento es “H1” para un titular muy grande, “H2” para uno grande, H3, H4, H5 y para letra pequeña H6. En la figura 15, se muestra el uso de estos tags.

```
1 <html>
2 <head>
3 <title> MI PRIMERA PAGINA </title>
4 </head>
5 <body>
6 <H1>Encabezamiento muy grande</H1>
7 <H2>Encabezamiento grande</H2>
8 <H3>Encabezamiento menos grande</H3>
9 <H6>Encabezamiento muy pequeño</H6>
10 </body>
11 </html>
```

Figura 15. Código para cambio de tamaño de los textos

La salida por pantalla de este programa se muestra en la figura 16. Observe como los textos van cambiando de tamaño:

Encabezamiento muy grande

Encabezamiento grande

Encabezamiento menos grande

Encabezamiento muy pequeño

Figura 16. Salida por pantalla de cambio de tamaños

Universidad Autónoma de Manizales

1.2.5.6. Salto de línea

En muchas ocasiones es necesario dar un salto de línea, algo como un enter o salto de carro en la página. En este caso se usa el tag:
 La figura 17, muestra el código que se usa.

```
1 <html>
2 <head>
3 <title> MI PRIMERA PAGINA </title>
4 </head>
5 <body bgcolor="#CCCCCC">
6 Estoy muy entretenido <br>
7 Es realmente bueno el curso de PROGRAMACION WEB<br>
8 de la UAM
9 </body>
10 </html>
```

Figura 17. Código Salto de línea

La salida por pantalla de este código se muestra en la figura 18.

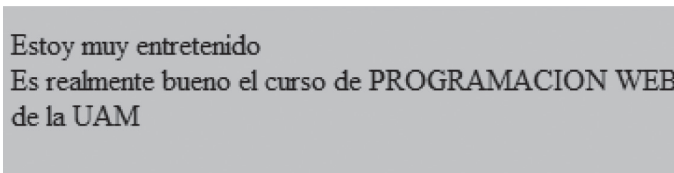


Figura 18. Salida de salto de línea

1.2.5.7. Creando links

Desde una página WEB, se puede ir a otra página web o a otro portal web; lo anterior se logra haciendo uso de los links. La figura 19, muestra el código que se usa.

```
1 <html>
2 <head>
3 <title>SESION No 1</title>
4 </head>
5 <body bgcolor="#CCCCCC">
6 <div align="center"> <font size="5">Veamos algunos links</font></div>
7 <!--Pagina web-->
8 <a href="http://www.fermath.org"> Sitio Web de Fermath </a>
9 <!--a otra página-->
10 <br>
11 <a href="http://www-app.etsit.upm.es/~alvaro/manual/manual.html"> Documento Basico en HTML </a>
12 <!--en otra ubicación-->
13 <br>
14 <a href="http://www.linux-es.org">Trabajando con Linux</a>
15 </body>
16 </html>
```

Figura 19. Trabajando con links



SESION NÚMERO UNO HTML BÁSICO

1.2.5.8. Trabajando con imágenes

Efectivamente una página web, sin imágenes no es una página atractiva. Es fundamental trabajar con la mayor cantidad de imágenes que impacten al visitante del sitio. En la figura 20 se visualiza su uso. En la figura 21, se ve la salida de ese código.

```
1 <html>
2 <head>
3   <title>SESION No 1</title>
4 </head>
5 <body bgcolor="#CCCCCC">
6   <div align="center"> <font size="5">Imágenes</font></div>
7   
8 </body>
9 </html>
```

Figura 20. Trabajando con imágenes

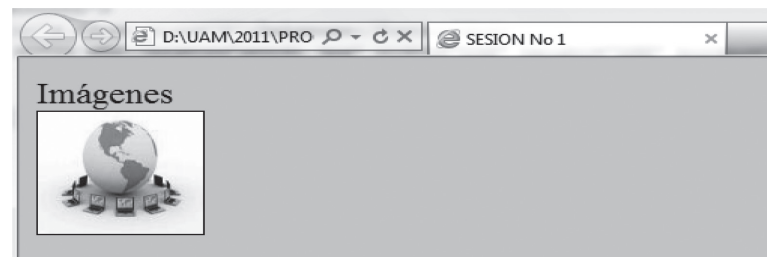
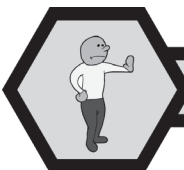


Figura 21. Salida de imágenes



Pare

Es muy importante en los sitios web, referenciar el correo electrónico para que se establezca el contacto con el dueño del sitio. Lo anterior se logra usando la instrucción que se visualiza en la figura 22.

```
1 <html>
2 <head>
3   <title>SESION No 1</title>
4 </head>
5 <body bgcolor="#CCCCCC">
6   <a href="mailto:hola@hola.com">AQUI EL AVISO A QUIEN SE ENVIARÁ EL MAIL</a>
7 </body>
8 </html>
```

Figura 22. Referenciando una dirección mail



Practicando el desarrollo WEB

Busque en el AULA VIRTUAL el apartado denominado: "EJEMPLOS DE LA SESIÓN 1"; o en el CD adjunto al material escrito, los ejemplos de la SESIÓN_1. Abra cada uno de los ejemplos, usando el ambiente adecuado. Digite o copie el programa y observe el resultado. Interiorice lo visto hasta el momento, mediante el análisis de cada uno de los ejemplos propuestos.

Universidad Autónoma de Manizales



Consultemos

Busque en la red o alguno de los textos propuestos en la bibliografía: ¿Qué es XHTML y DHTML? ¿Cómo se modifican los colores de un Link? Organice su respuesta y escríbala en un archivo de texto; una vez tenga el archivo grabado en su computador, ingrese al Aula Virtual y busque el link marcado como CONSULTA No 1. Ingrese y envíelo como un adjunto.



Actividad

Ingrese a AULA VIRTUAL y desarrolle la actividad marcada como: ACTIVIDAD No 1. Esta le ayudará a reforzar lo planteado en este módulo.



Aplicando lo aprendido

Ingrese al Aula Virtual y ubique en la Sesión el apartado denominado: "LABORATORIO DE SESIÓN", despliegue el archivo LABORATORIO No 1, realice el algoritmo pedido luego de estar seguro que cumple las pruebas de escritorio, envíelo al docente en el link denominado: Entrega LABORATORIO No 1.



En el Aula Virtual

EN EL AULA VIRTUAL Encontrará documentación adicional que le servirá para ampliar los conocimientos, frente a las temáticas tratadas en esta sesión. Revíselos y consulte a sus tutores si surge alguna inquietud.

1.3. CONCLUSIONES

- La principal característica del HTML es el uso de etiquetas o *tags*, para formatear el texto, imágenes, etc.
- Con HTML se puede dar formato a: título, texto, color de texto, fondos, salida por pantalla, vínculos, tratamiento de imágenes de las distintas páginas web que se desarrollen.
- Para escribir código HTML, se puede utilizar un block de notas o un ambiente de desarrollo como el Notepad++.
- Para establecer la salida por pantalla de un código HTML realizado, se escribe el nombre de la página en un navegador cualquiera, como si fuera una dirección web.
- Las extensiones básicas de código HTML es .htm, .html, xhtml, entre otros.

```
51 DOCUMENT_POSITION_PRECEDING : 0x00,  
52 DOCUMENT_POSITION_FOLLOWING : 0x01,  
53 DOCUMENT_POSITION_CONTAINS : 0x02,  
54 DOCUMENT_POSITION_CONTAINED_BY : 0x03,  
55 DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC : 0x04  
56 }  
57  
58 // redefine setAttribute function to emulate DOM2 modified num  
59 // http://bugzilla.mozilla.org/show_bug.cgi?id=4911  
60 Element.prototype.setAttribute = Element.prototype.setAttribute  
61 Element.prototype.setAttribute = function(name, val)  
62 {  
63     // save the previous value  
64     var prev = this.getAttribute(name);  
65 }  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000
```



2.1. OBJETIVOS ESPECÍFICOS

- Definir el diseño de textos preformateados con HTML.
- Distinguir la forma de escribir acentos y caracteres especiales en HTML
- Categorizar los distintos elementos que se tienen en cuenta para crear tablas en HTML.
- Explicar los distintos tipos de listas que se pueden crear en HTML.
- Identificar la forma de hacer comentarios en páginas HTML.
- Distinguir los distintos atributos que se pueden usar en tablas con HTML.

2.2. DESARROLLO TEMÁTICO

2.2.1. Texto preformateado

En ocasiones al diseñar una página *web*, se desea hacer una salida por pantalla que muestre algo novedoso como se muestra en la figura 23

```

      ( @ @ )
-----oOoO- ( _ ) -oOoO-----
|
|      Fernando Martínez R
|      LA VIDA ES BELLA
|
|-----Oooo.-----
      .oooo      ( )
      ( )         ( )
      \ (         \ /
      \ _         \ /
  
```

Figura 23. Salida con formato

Para lograr esto utilizamos una nueva etiqueta: `<PRE>...</PRE>` El nombre asignado a esta etiqueta proviene de la palabra "PREformateado" (o *PREformatted* en inglés) y su función es mantener todo formateado manual que sea hecho en el texto escrito entre la instrucción de inicio (`<PRE>`) y la instrucción de fin (`</PRE>`). La figura 24 muestra el código que implementa la salida anterior:

SESION NÚMERO DOS HTML INTERMEDIO

```

1 <HTML>
2 <HEAD>
3 <TITLE> USO DEL PREFORMATO </TITLE>
4
5 </HEAD>
6
7 <BODY>
8
9 <PRE>
10
11      ( @ @ )
12      -----o00o- ( _ )-o00o-----
13      |
14      |      Fernando Martínez R
15      |      LA VIDA ES BELLA
16      |
17      -----Oooo.-----
18      .oooO      ( )
19      ( )      ) /
20      \ (      (/
21      \ )
22
23 </PRE>
24 </BODY>
25 </HTML>

```

Figura 24. Código de Pre formato

2.2.2. Acentos y otros caracteres especiales

Al hacer páginas web, se debe guardar una relación directa con las normas de ortografía que se marcan para cada idioma. En el caso de Colombia, se usa ortografía para el idioma español. Este idioma tiene acentos o algunos caracteres especiales para que las letras suenen, como la ñ. Para su adecuada visualización en cualquier navegador, se usan códigos especiales, como las Entidades de caracteres con nombres y de caracteres numéricos **1 Entidades de caracteres con nombre.**

2.2.2.1. Son aquellas en las que los símbolos ‘&’ y ‘;’ se pone el nombre (o abreviatura) asignado a ese carácter. Sólo existen para los caracteres especiales más usados. En la figura 25 se puede ver un ejemplo de este tipo de entidades en el que vemos la estructura general que todas tienen.

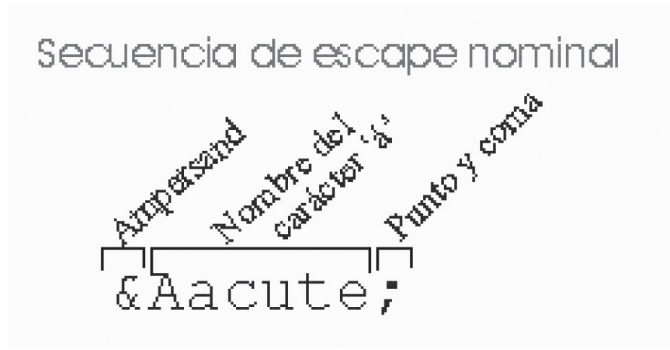
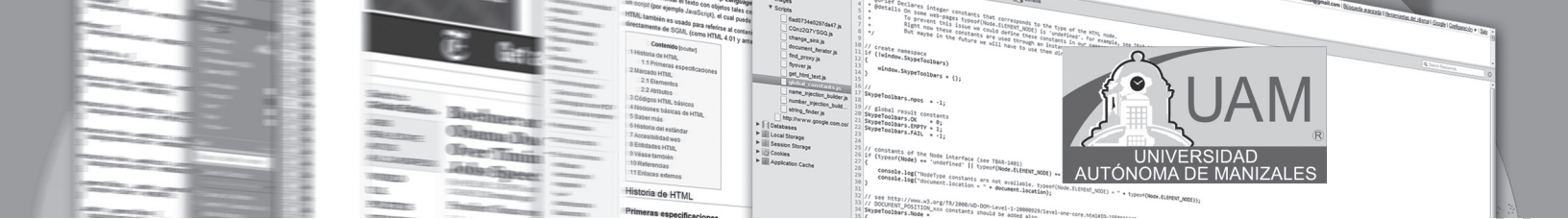


Figura 25. Una entidad de carácter con nombre tiene tres partes: ‘&’ + ‘nombre’ + ‘;’



Universidad Autónoma de Manizales

2.2.2.2. Entidades de caracteres numéricas: en este tipo de entidades entre ‘&’ y ‘;’ se escribe el número asignado a ese carácter en el estándar ISO-Latin-1 precedido de una almohadilla: ‘#’. Este tipo de entidades son menos usadas que las anteriores aunque tienen la ventaja de abarcar cualquier letra posible en cualquier idioma. En la figura 26 se muestra la estructura de este tipo de entidades usando de nuevo como ejemplo la letra A mayúscula y acentuada.

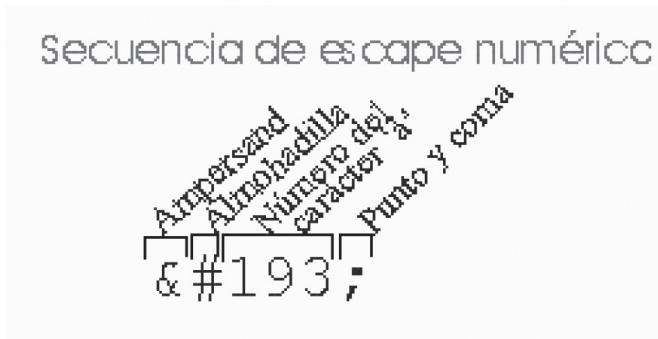


Figura 26. Una entidad de carácter numérica tiene cuatro partes: ‘&’ + ‘#’ + ‘número’ + ‘;’

2.2.3. Tabla de caracteres importantes

La figura 27, muestra la tabla con los caracteres que se deben tener en cuenta y la forma como se deben escribir, para que se lean en cualquier navegador.

Carácter	Secuencia escape	Sec. escape numérica	Carácter	Secuencia escape	Sec. escape numérica
á	á	á	Á	Á	Á
é	é	é	É	É	É
í	í	í	Í	Í	Í
ó	ó	ó	Ó	Ó	Ó
ú	ú	ú	Ú	Ú	Ú
ü	ü	ü	Ü	Ü	Ü
ñ	ñ	¤	Ñ	Ñ	¥
¡	-----	¡	¿	-----	¿

Carácter	Secuencia de escape	Sec. de esc. numérica
<	<	‹
>	>	›
&	&	&
“	"e;	"

Carácter	Secuencia de escape	Sec. de esc. numérica
©	©	©
®	®	®
[espacio]	 	-----

Figura 27. Tabla de caracteres especiales



SESION NÚMERO DOS HTML INTERMEDIO

La figura 28, muestra el código donde se puede visualizar el uso de caracteres para mostrar la a tildada...

```

1 <HTML>
2 <HEAD>
3 <TITLE> caracteres especiales </TITLE>
4 </HEAD>
5 <BODY>
6 &aacute;
7 <BR>
8 &#225;
9 </BODY>
10 </HTML>

```

Figura 28. Uso de caracteres especiales.

2.2.4 LISTAS

Una lista, o listado, es una enumeración de dos o más elementos y suelen disponerse de tal forma que se facilite la distinción entre ellos. El lenguaje HTML define los siguientes tipos de listas: **Numerada u ordenada. No ordenada. Listas de glosario o de definición. Menús. Usadas con párrafos cortos. Listas de directorio.**

2.2.4.1 LISTA DESORDENADA.

La etiqueta ... (UL proviene de *Unordered List*, que significa lista no ordenada). Dentro de estas etiquetas insertaremos cada elemento con la etiqueta

2.2.4.2 LISTA ORDENADA.

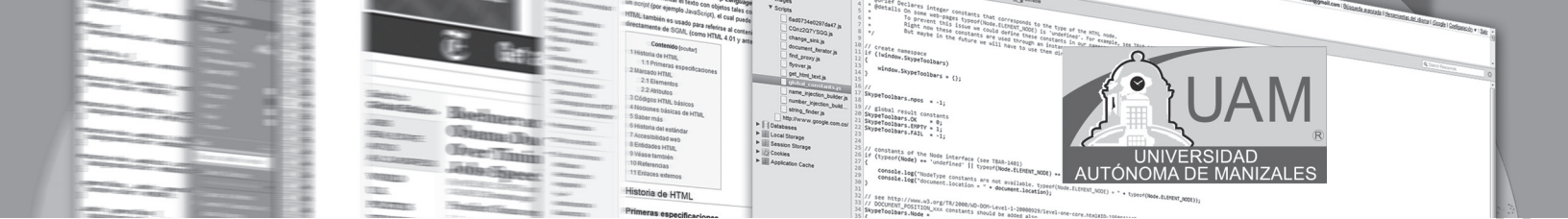
La etiqueta ... (OL: Ordered List, Lista ordenada), y cada elemento se inserta, al igual que en el caso de listas no ordenadas, con la etiqueta .

2.2.4.3 GLOSARIO.

- *La palabra que queremos definir:* para insertar la palabra usaremos la etiqueta <DT> que proviene de Definition Term (Término de definición). Esta nueva etiqueta funciona de forma muy similar a y la instrucción de fin (</DT>) es igualmente optativa.
- *La definición o descripción de esa palabra:* una vez indicada la palabra es hora de poner su definición. Para ello usamos otra nueva etiqueta: <DD> cuyas siglas significan *Definition Description* (parte de la definición que contiene la descripción). La definición suele ser un párrafo y será mostrado por los navegadores sangrado con respecto al texto anterior.

2.2.4.4 MENU Y DIRECTORIO

En Explorer las listas de menú (<MENU>) y de directorio (<DIR>) son mostradas de igual manera que las listas no numeradas ().



Universidad Autónoma de Manizales

2.2.4.5. ANIDAMIENTO DE LISTAS

Vistos todos los tipos de listas se nos puede plantear una pregunta: ¿Se puede incluir una lista dentro de otra? La respuesta es SI. El lenguaje HTML permite insertar unas listas dentro de otras ya sean del mismo o de distinto tipo. A esta posibilidad se le llama anidamiento de listas y la forma de realizarlo es incluir todo el código de una lista dentro de uno de los elementos de la otra. La figura 29, muestra el código para ejemplos variados de listas.

```

1 <HTML>
2 <HEAD>
3 <TITLE> listas </TITLE>
4 </HEAD>
5 <BODY>
6 Yo practico:
7 <UL>
8 <LI>Fútbol
9 <LI>Baloncesto
10 <LI>Tenis
11 <LI>etc.
12 </UL>
13 LISTAS DESORDENADAS
14 Mi ordenador tiene las siguientes características:
15 <UL>
16 <LI>Procesador AMD Athlon 64
17 <LI>2 MB en RAM
18 <LI>Tarjeta de sonido 32 bits
19 <LI>Tarjeta de video con 10 Mb
20 </UL>
21 LISTAS ORDENADAS
22 Un buen equipo adaptado a las nuevas tecnologías consta de:
23 <OL>
24 <LI>CPU rápida
25 <LI>Impresora a color de buena definición.
26 <LI>Altavoces y cámara de video.
27 <LI> video.
28 <LI>Módem de velocidad 28K o superior.
29 </OL>
30 </BODY>
31 </HTML>

```

Figura 29. Ejemplo de Listas

La salida a este código se muestra en la figura 30.



SESION NÚMERO DOS HTML INTERMEDIO

Yo practico:

- Fútbol
- Baloncesto
- Tenis
- etc.

LISTAS DESORDENADAS Mi ordenador tiene las siguientes características:

- Procesador AMD Athlon 64
- 2 MB en RAM
- Tarjeta de sonido 32 bits
- Tarjeta de video con 10 Mb

LISTAS ORDENADAS Un buen equipo adaptado a las nuevas tecnologías consta de:

1. CPU rápida
2. Impresora a color de buena definición.
3. Altavoces y cámara de video.
4. video.
5. Módem de velocidad 28K o superior.

Figura 30. Salida por pantalla del manejo de listas

2.2.5. Como hacer comentarios en html

Para hacer comentarios del código en HTML, se escribe entre `<!--` y `-->` Como se muestra en el ejemplo siguiente: `<!-- Aquí se escribe el comentario -->` Depende del programador que tantos comentarios y de qué amplitud, se puede desea hacer cada uno de ellos.

2.2.6. Tablas con HTML

Para crear las tablas será necesario usar una nueva etiqueta: *TABLE*, que significa, en inglés, tabla. Esta etiqueta consta de instrucción de inicio, `<TABLE>`, e instrucción de fin, `</TABLE>`. Entre ambas debemos introducir otras tres etiquetas, que definirán la estructura de la tabla. Estas etiquetas son:

- **TR:** la etiqueta *Table Row* nos permitirá insertar filas en la tabla. La tabla tendrá tantas filas como apariciones de esta etiqueta haya entre `<TABLE>` y `</TABLE>`. La instrucción de inicio de TR, `<TR>`, marca el comienzo de la línea, la instrucción de fin, `</TR>`, marca su final. La instrucción de fin es optativa, si no se usa se considera que una línea ha acabado cuando comienza otra o cuando acaba la tabla. Entre el comienzo y el fin de la línea debemos insertar las celdas de la tabla.
- **TH:** ete es el primero de los dos tipos de celdas existentes en el lenguaje HTML. El nombre de la etiqueta proviene del inglés *Table Header*, encabezado de tabla. Al igual que la etiqueta anterior el comienzo del encabezado viene dado por la instrucción de inicio, `<TH>` y finaliza con la instrucción de fin, `</TH>`, o con el comienzo de otra celda. En ge-

Universidad Autónoma de Manizales

neral entre el comienzo y el fin se suele insertar texto, que será mostrado en negrita o subrayado y centrado.

- TD: este es el segundo tipo de celda de las tablas HTML. El nombre de la etiqueta, *Table Data* (Datos de tabla), nos induce a pensar que su función será introducir todos los datos que queramos en las celdillas definidas de esta forma. Y así será, de hecho podemos

```
5 <BODY>
6 <TABLE BORDER="5">
7   <TR><!--ENCABEZADO -->
8   <TH>
9     Encabezado1
10  </TH>
11  <TH>
12    Encabezado2
13  </TH>
14  <TH>
15    Encabezado3
16  </TH>
17 </TR><!--FIN DEL ENCABEZADO -->
18 <TR><!--FILA 1 -->
19 <TD><!--DATOS FILA 1 COLUMNA 1 -->
20  Filal,Columna1
21 </TD>
22 <TD><!--DATOS FILA 1 COLUMNA 2 -->
23  Filal,Columna2
24 </TD>
25 </TR><!--FIN DE LA FILA 1 -->
26 <TR><!--FILA 2 -->
27 <TD>
28  Filal2,Columna1
29 </TD>
30 <TD>
31  Filal2,Columna2
32 </TD>
33 </TR><!--FIN DE LA FILA 2 -->
34 </TABLE>
35 </BODY>
```

insertar cualquier elemento de HTML: imágenes, listas, texto formateado e incluso otras tablas. La diferencia de esta celda con la anterior es pequeña, y su existencia separada sólo tiene sentido si consideramos el carácter descriptivo del HTML. En la práctica la diferencia principal es que el texto de los encabezados de tabla aparecerá resaltado y centrado y el de las celdas normales (TD) no.

El uso de las instrucciones de fin de TR, TD y TH es, como hemos dicho, optativo. Sin embargo, en las primeras versiones del lenguaje HTML era obligatorio usarlas y por esta razón algunos navegadores antiguos las necesitan para poder entender las tablas. Es por ello que nuestra recomendación es usar siempre dichas instrucciones de fin para seguir con nuestro objetivo de llegar a la mayor cantidad de público posible.

Figura 31. Código para genera tablas

La figura 31 Muestra algunos tags usados para generar una tabla. La salida de este código se visualiza en la figura 32.

Encabezado1	Encabezado2	Encabezado3
Fila1,Columna1	Fila1,Columna2	
Fila2,Columna1	Fila2,Columna2	

Figura 32. Salida del código mostrado en la figura 31



SESION NÚMERO DOS HTML INTERMEDIO

2.2.7. Otros atributos importantes para tener en cuenta con tablas

El atributo del que hablamos es BORDER, que significa borde en inglés. Este atributo puede tomar un valor en pixeles que representa el grosor del borde a mostrar. Si incluimos BORDER sin ningún valor es equivalente a BORDER="1"

Poniendo
 o las celdas que antes no existían ahora si son dibujadas, pero sin nada dentro.

Con la etiqueta CAPTION podemos poner un título a la tabla. Este título puede situarse encima (ALIGN="top") o debajo (ALIGN="bottom") de la tabla. La figura 33 muestra un ejemplo usando algunos de estos atributos. La salida por pantalla de este código se puede visualizar en la figura 34.

```

6 <BODY>
7 <TABLE>
8 <CAPTION ALIGN="top">
9 <BIG>
10 <B>
11 Tabla de ejemplo
12 </B>
13 </BIG>
14 </CAPTION>
15 <TR>
16 <TD>
17 Fila1,Columna1
18 </TD>
19 <TD>
20 Fila1,Columna2
21 </TD>
22 <TR>
23 <TD>
24 Fila2,Columna1
25 </TD>
26 <TD>
27 Fila2,Columna2
28 </TD>
29 </TABLE>
30 </BODY>

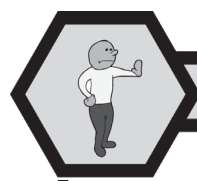
```

Figura 33. Otros atributos.

Tabla de ejemplo

Fila1,Columna1	Fila1,Columna2
Fila2,Columna1	Fila2,Columna2

Figura 34. Salida de otros atributos



Pare

El uso de las TABLAS en código HTML es bastante importante en la etapa de diseño de Formularios. Para la creación de un formulario muy bien formateado, donde los campos se centren totalmente, se toma como base una tabla con borde cero; lo anterior hace que el diseño sea bastante limpio en los formularios de usuario.



Practicando el desarrollo WEB

Busque en el AULA VIRTUAL el apartado denominado: "EJEMPLOS DE LA SESION 2"; o en el CD adjunto al material escrito, los ejemplos de la SESION_2. Abra cada uno de los ejemplos, usando el ambiente adecuado. Digite o copie el programa y observe el resultado. Interiorice lo visto hasta el momento, mediante el análisis de cada uno de los ejemplos propuestos.

Universidad Autónoma de Manizales



Consultemos

Busque en la red o alguno de los textos propuestos en la Bibliografía: ¿Cuál es la historia del lenguaje de programación HTML; dónde están sus orígenes? ¿Quiénes son la W3C?; ¿quiénes fueron sus autores desde el inicio y cuál es el futuro de este lenguaje de programación? Organice su respuesta y escríbala en un archivo de texto; una vez tenga el archivo grabado en su computador, ingrese al Aula Virtual y busque el link marcado como CONSULTA No 2. Ingrese y envíelo como un adjunto.



Actividad

Ingrese a AULA VIRTUAL y desarrolle la actividad marcada como: ACTIVIDAD No 2. Esta le ayudará a reforzar lo planteado en este módulo.



Aplicando lo aprendido

Ingrese al Aula Virtual y ubique en la Sesión el apartado denominado: "LABORATORIO DE SESIÓN", despliegue el archivo LABORATORIO No 2, realice el algoritmo pedido luego de estar seguro que cumple las pruebas de escritorio, envíelo al docente en el link denominado: Entrega LABORATORIO No 2.



En el Aula Virtual

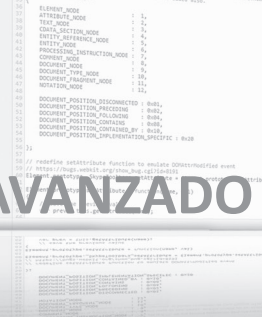
EN EL AULA VIRTUAL Encontrará documentación adicional que le servirá para ampliar los conocimientos, frente a las temáticas tratadas en esta sesión. Revíselos y consulte a sus tutores si surge alguna inquietud.

2.3. CONCLUSIONES

- En HTML se pueden hacer diseños preformateados que permiten la salida por pantalla de páginas con algún toque personalizado.
- Es importante tener en cuenta al momento de crear caracteres especiales o tildar palabras, la forma estándar para que estos sean leídos sin problemas en cualquier navegador.
- Al crear tablas se debe considerar cada uno de los elementos que conforman las filas y las columnas; ocasionalmente se tiene en cuenta el borde y la alineación.

SESION NÚMERO DOS HTML INTERMEDIO

- Los tipos de lista que se pueden considerar en HTML son: lista ordenada, desordenada, glosario, menú y directorio. Algunas pueden ser ordenadas.
- Al igual que en cualquier lenguaje de programación en HTML se pueden hacer comentarios entre: `<!--` y `-->`, ya sea de una o varias líneas.



3.1. OBJETIVOS ESPECÍFICOS

- Reconocer la importancia de la CGI, en la formación de formularios.
- Identificar las fases importantes en la elaboración de plantillas.
- Categorizar los distintos elementos de la marca INPUT.
- Enunciar los distintos elementos de los marcos.
- Reconocer la importancia de la directiva FRAME.
- Identificar los distintos elementos que conforman las capas.

3.2. DESARROLLO TEMÁTICO

3.2.1. Formularios en HTML

Otra de las características importantes de la web es la posibilidad que los usuarios nos proporcionen información, es decir, la tan nombrada interactividad. Mediante la utilización de formularios podemos demandar ciertos datos a nuestros usuarios, bien sea a un email o a una base de datos.

3.2.2. CGI o Common Gateway Interface

La mayor parte de los elementos HTML de que disponemos permite al visitante visualizar los contenidos de un sitio, pero no interactuar con él. Dicho de otro modo, la relación usuario/página es unidireccional y estática. Con las plantillas, por el contrario, el usuario puede interactuar con el sitio enviando sus comentarios, haciendo peticiones sin necesidad de escribir mensajes por correo electrónico, firmando en el libro de visitas, contestando a encuestas y sondeos, etc. Todo ello define una relación bidireccional, que es posible sólo gracias a la intervención de programas residentes en servidores WWW: se trata de los denominados **CGI o Common Gateway Interface (Interfaz Común de Pasarela)**.



A decir verdad, un formulario puede elaborarse también sin pasar por un CGI, pero las desventajas y límites que comporta este procedimiento no pueden obviarse por lo que no debería adoptarse sino en casos de extrema necesidad. En las páginas que siguen trataremos también detalladamente esta posibilidad.

3.2.3. La elaboración de una plantilla consta de dos fases:

- impostación de las marcas para la creación de la plantilla, de los campos y del botón de envío;



SESION NÚMERO TRES HTML AVANZADO

- elaboración de un script CGI en servidor (o llamamiento de un script ya impostado)

3.2.3.1. Las marcas de los formularios

Se puede incluir un formulario en una página web con las etiquetas **<FORM>** y **</FORM>**.

La sintaxis usual de las marcas citadas es la siguiente:

<FORM method="get|post" action="http://www.tuosito.com/cgi-in/nome_script.cgi">

Si METHOD está impostado como GET, los datos se envían al servidor y se separan en dos variables. Para este método, el número máximo de caracteres que puede contener el formulario es de 255. Utilizando "method=post", el script CGI recibe directamente los datos sin un proceso previo de decodificación. Esta característica hace que el script pueda leer una cantidad ilimitada de caracteres.



Pare

Hasta aquí hemos visto la forma de crear un formulario, pero nos queda lo más importante su contenido, que se define mediante los tags **<INPUT>**, **<TEXTAREA>** y **<SELECT>**. El primero introduce la típica caja en el formulario para que el usuario la rellene, y puede ser de varios tipos, el segundo es para introducir textos largos, y el tercero para presentar listas de opciones y que el usuario elija una.

3.2.3.2. La marca <INPUT>

La marca de base para la definición de los elementos de un formulario es **<INPUT>**, que se utiliza para agregar botones, menús de selección, contraseñas, etc. A **<INPUT>** se le pueden asignar los ocho (8) valores que analizamos a continuación.

3.2.3.2 .1. Type="TEXT"

Este valor crea los típicos campos de texto, en los que generalmente se solicitan datos tales como el nombre o la dirección de correo electrónico. Es un valor usado sobre todo para informaciones no predefinidas, que cambian cada vez. TEXT tiene tres atributos opcionales, presentes también en el ejemplo: **maxlength** (el número máximo de caracteres que puede ser introducido en el campo y que no es posible superar), **size** (la anchura de la cadena dentro de la página) y **value** (visualiza un texto por defecto dentro de la cadena).

<INPUT type="TEXT" name="nombre" maxlength="40" size="33" value="Tu nombre">

3.2.3.2.2. Type="PASSWORD"

Este campo de texto funciona como el anterior, pero con una pequeña diferencia que su propio nombre indica: cuando se escribe dentro del cajetín blanco, no se ven los caracteres sino los clásicos asteriscos de las contraseñas. En realidad, los datos no se codifican de ningún modo por lo que esta marca no consigue eliminar la inseguridad de fondo.

<INPUT type="PASSWORD" name="nombre" maxlength="40" size="33">

3.2.3.2.3. Type="CHECKBOX"

Este atributo se utiliza para representar datos de tipo "sí/no" y "verdadero/falso". Crea pequeñas casillas cuadradas que hay que puntear o dejar en blanco. Si la casilla está punteada, INPUT restituye un valor al CGI; en caso contrario, no restituye ningún valor. Value impostado en "sí" significa que por defecto la casilla está punteada. Checked controla el estado inicial de la casilla, en el momento de cargar la página.

3.2.3.2.4. Type="RADIO"

Este atributo tiene funciones similares al visto anteriormente, pero contiene más posibilidades de selección. Seleccionando una de las voces presentes, y siempre y cuando tengan todas un valor "name" idéntico, las restantes se deseleccionan automáticamente.

```
<INPUT type="RADIO" name="calificación" value="suficiente">
<INPUT type="RADIO" name="calificación" value="bueno">
<INPUT type="RADIO" name="calificación" value="excelente">
```

3.2.3.2.5. Type="SUBMIT"

El clásico botón de envío que remite el formulario con todo su contenido. La dimensión del botón depende de la longitud del texto.

```
<INPUT type="SUBMIT" value="Envía">
```

3.2.3.2.6. Type="RESET"

Botón que vuelve a impostar todo el formulario eliminando los datos introducidos.

```
<INPUT type="RESET" value="Reinicio">
```

3.2.3.2.7. Type="IMAGE"

Función similar a la de la tecla "ENVÍA", pero con la diferencia de que en lugar del botón de la opción por defecto, aparece una imagen.

```
<INPUT type="IMAGE" src="pulsante.gif">
```

3.2.3.3. TEXTAREA

TEXTAREA se utiliza para introducir comentarios o campos que prevén muchas líneas de texto. La anchura se imposta con "cols" y la altura con "rows". WRAP="physical" establece que, en el caso de que el texto introducido supere la anchura de la ventana, salte de línea automáticamente.

```
<TEXTAREA cols=40 rows=5 WRAP="physical" name="comentario"></textarea>
```

3.2.3.4. SELECT

SELECT es un elemento que permite la creación de listas con varias posibilidades de selección. En nuestro ejemplo, hemos hipotizado que se solicita una calificación sobre un sitio web.

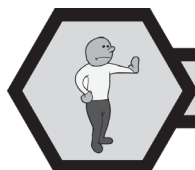


SESION NÚMERO TRES HTML AVANZADO

```

<SELECT size=1 cols=4 NAME="calificación">
<OPTION selected value=ninguna>
<OPTION value=bueno> Bueno
<OPTION value=suficiente> Suficiente
<OPTION value=excelente> Excelente
</select>

```



Pare

EJEMPLO DE CREACION DE FORMULARIO.

Este código trata de reunir todos los conceptos que se han enunciado anteriormente. Dígitelo en su computador y establezca que el resultado es el que se muestra en la figura 35.

```

<html>
<head>
<title>EJEMPLO DE FORMULARIO</title>
</head>
<body>

```

3.2.3.5. <!--ETIQUETA PRINCIPAL PARA CREAR FORMULARIOS-->

```
<form name="Formulario" method="POST" action="mailto:sigmapico@gmail.com">
```

3.2.3.6. <!--TITULO PRINCIPAL-->

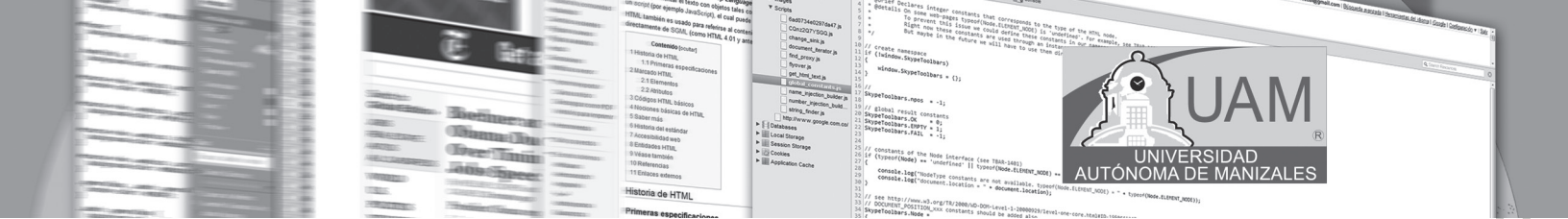
```
<p align="center"><b><font face="Bernard MT Condensed" size="5"
color="#FF0000">SUSCRIPCIÓN</font></b></p>
```

3.2.3.7. <!--SE CREA UNA TABLA-->

```
<table cellSpacing=1 width="96%" border=0>
<tr align=middle>
```

3.2.3.8. <!--USO DEL TYPE TEXT-->

```
<td width="15%" bgColor=#e1e1e1 height=21>Nombres:</td>
```

Universidad Autónoma de Manizales

```

<td width="32%" bgColor=#e1e1e1 height=21 colspan="2"><input type="text"
name="nombres" size="23"></td>

<td width="23%" bgColor=#e1e1e1 height=21 colspan="2">Apellidos:</td>

<td width="23%" bgColor=#e1e1e1 height=21><input type="text" name="apellidos"
size="24"></td>

</tr>

<tr align=middle>

<td width="47%" bgColor=#e1e1e1 height=21 colspan="3">Pais:

<input type="text" me="pais" size="26"></td>

<td width="46%" bgColor=#e1e1e1 height=21 colspan="3">Departamento

<input type="text" name="departamento" size="26"> </td>

</tr>

<tr align=middle>

<td width="43%" bgColor=#e1e1e1 height=21 colspan="2">Dirección</td>

<td width="86%" bgColor=#e1e1e1 height=21 colspan="4"><input type="text"
name="direccion" size="50"></td>

</tr>

<tr>

<td width="17%" bgColor=#e1e1e1 height=21 colspan="4">

<p align="left"><font size="1"><font face="Arial">Fecha de Nacimiento</font></font>

<b><font face="Arial" size="1">(dd/mm/aaaa)</font></b><input type="text"
name="fechanace" size="14"></p>

</td>

```

3.2.3.9. <!--USO DEL CHECKED-->

```

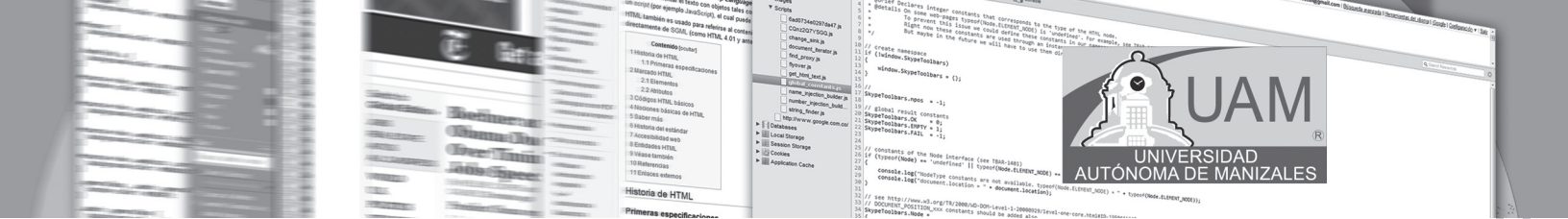
<td width="112%" bgColor=#e1e1e1 height=21 colspan="2">

Sexo&nbsp;F <input type="radio" value="F" checked
name="sexo">&nbsp;&nbsp;&nbsp;M<input type="radio" value="M" name="sexo"></td>

</tr>

```





Universidad Autónoma de Manizales

3.2.3.13. <!--USO DEL SUBMIT-->

```
<p align="center"><input type="submit" value="Enviar" name="accion"></p>
```

```
<p>&nbsp;</p>
```

```
</table>
```

```
</body>
```

```
</html>
```

SUSCRIPCIÓN

Nombres: <input type="text"/>		Apellidos: <input type="text"/>	
País: <input type="text"/>		Departamento: <input type="text"/>	
Dirección: <input type="text"/>			
Fecha de Nacimiento (dd/mm/aaaa) <input type="text"/>		Sexo <input type="radio"/> F <input type="radio"/> M <input type="radio"/> O	
Colores del producto <input checked="" type="checkbox"/> Azul <input type="checkbox"/> Verde <input type="checkbox"/> Rojo <input type="checkbox"/> Amarillo		Producto a pedir <input type="text" value="Computador HP"/>	
OBSERVACIONES			
<input type="text"/>			
<input type="button" value="Enviar"/>			

Figura 35. Salida por pantalla de un formulario de inscripción

3.2.4. Marcos

Los **marcos** (vulgarmente conocidos como **frames**) permiten dividir la pantalla en diversas zonas que pueden actuar independientemente unas de otras. Resultan útiles para mostrar permanentemente en una ventana el sumario de una página, y en otra, el contenido seleccionado.

Para definir las diferentes subventanas o frames se utilizan las directivas **<FRAMESET>**, **</FRAMESET>** y **<FRAME>**. La directiva **<FRAMESET>** indica cómo se va a dividir la ventana principal. Pueden incluirse varias directivas **<FRAMESET>** anidadas con la finalidad de subdividir una división.

Los atributos **ROWS** y **COLS** definen si la división de la pantalla se hace por filas o por columnas. Cuando se definen las filas, los tamaños de las subventanas (en puntos o en porcentajes) se indican de la superior a la inferior. En el caso de las columnas, los tamaños se indican de izquierda a derecha.



Por ejemplo:

<FRAMESET ROWS="25%,50%,25%">

creará tres subventanas horizontales; la primera ocupará el 20% de la ventana principal, la segunda un 50% y la tercera un 25%.

<FRAMESET COLS="120*,100">

creará tres subventanas verticales, la primera y la tercera tendrán una anchura fija de 120 y 100 puntos respectivamente. La segunda (*) ocupará el resto de la ventana principal.

<FRAMESET COLS = "20%,*"> <FRAMESET ROWS = "30%,*">

anidará las dos directivas. La primera dividirá la ventana principal en dos subventanas verticales, la primera ocupará un 20% de la ventana principal y la segunda el resto. La segunda directiva volverá a subdividir la primera subventana creada anteriormente, pero esta vez en dos subventanas horizontales, la superior ocupará un 30% de la subventana, y la inferior el resto.

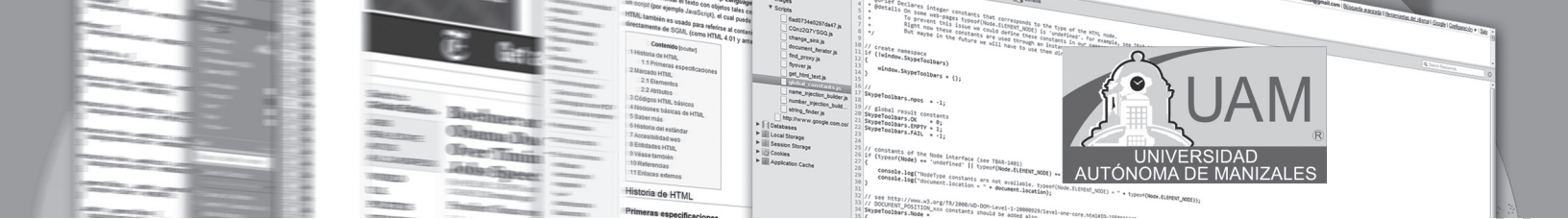
3.2.5. La directiva <FRAME> indica las propiedades de cada subventana. Es necesario indicar una directiva <FRAME> para cada subventana creada. Los parámetros posibles son: (ver figura 36)

<code>name="nombre"</code>	Nombre de la subventana
<code>src="URL"</code>	Contenido de la ventana.
<code>marginwidth=n</code>	Margen izquierdo y derecho de la subventana (en puntos).
<code>marginheight=n</code>	Margen superior e inferior de la subventana (en puntos).
<code>scrolling="yes no auto"</code>	Indica si se aplica una barra de desplazamiento a la subventana en caso de que la página que se cargue en ella no quepa en los límites de la subventana.
<code>noresize</code>	Evita que el usuario pueda redimensionar las subventanas. Omitiendo este parámetro, el usuario podrá modificar el tamaño de los marcos seleccionando el borde con el ratón y desplazándolo.

Figura 36. Parámetros para Frame

¡IMPORTANTE!

La definición de los marcos debe ir antes de la definición del cuerpo del documento (<BODY>).



Universidad Autónoma de Manizales

3.2.5.1. EN LA PAGINA QUE ESTA DENTRO DEL MARCO

Cuando se quiera mostrar una página hay que indicar al navegador en qué subventana debe aparecer. Por defecto, se mostrará en la ventana donde se encuentre el enlace, pero se puede escoger la subventana de destino del enlace incluyendo el parámetro TARGET en la etiqueta <A HREF>, que puede tener los siguientes valores: (ver figura 37).

<code>target="nombre_ventana"</code>	Muestra el contenido del enlace en la subventana indicada.
<code>target="_blank"</code>	Abre una nueva ventana y muestra el enlace en ella.
<code>target="_self"</code>	El contenido del hipertexto aparece en la subventana activa.
<code>target="_parent"</code>	Muestra el contenido del enlace en el anterior FRAMESET definido. Si no hay ningún FRAMESET definido se muestra a pantalla completa eliminando todos los marcos de la ventana.
<code>target="_top"</code>	Elimina todas las subventanas de la pantalla y muestra el enlace a pantalla completa.

Figura 37. Valores para la etiqueta

¡IMPORTANTE!

Para eliminar el borde gris de los marcos, se debe insertar el siguiente código:

```
<frameset cols="20%,60%,20%" border=0>
```

Para impedir que los marcos sean redimensionados por el visitante:

```
<frame name="alto" src="top.htm" noresize>
```

Para eliminar siempre las barras de desplazamiento (scrollbars):

```
<frame name="alto" src="top.htm" scrolling="no">
```

Para mostrarlas siempre:

```
<frame name="alto" src="top.htm" scrolling="yes">
```

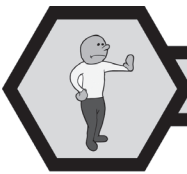
Para mostrarlas sólo cuando son necesarias:

```
<frame name="alto" src="top.htm" scrolling="auto">
```

Para regular la distancia del contenido del marco al margen superior (marginheight) y a los márgenes izquierdo y derecho (marginwidth):

```
<frame name="alto" src="top.htm" marginheight=2 marginwidth=5>
```

SESION NÚMERO TRES HTML AVANZADO



Pare

EJEMPLO DE CREACION DE SITIO WEB CON FRAME.

Este código trata de reunir todos los conceptos que se han enunciado anteriormente. Digítelo en su computador y establezca que el resultado es el que se muestra en la figura 42. Detecte que se tienen cuatro códigos (Figuras 38 a 41)...uno principal y tres que dependen del Principal que declara el FRAME.

```

1 <HTML>
2 <HEAD>
3 <TITLE> CREANDO FRAMES </TITLE>
4 </HEAD>
5 <frameset rows="20%, 80%">
6   <frame name="UNO" src="S3_ejemplo21.html">
7   <frameset cols="20%, 80%">
8     <frame name="DOS" src="S3_ejemplo22.html">
9     <frame name="TRES" src="S3_ejemplo23.html">
10  </frameset>
11 </frameset>
12
13 <BODY>
14
15 </BODY>
16 </HTML>

```

Figura 38. Código principal de FRAME

```

1 <HTML>
2 <HEAD>
3 <TITLE> PARTE SUPERIOR </TITLE>
4 </HEAD>
5 <BODY>
6   PAGINA UNO
7 </BODY>
8 </HTML>

```

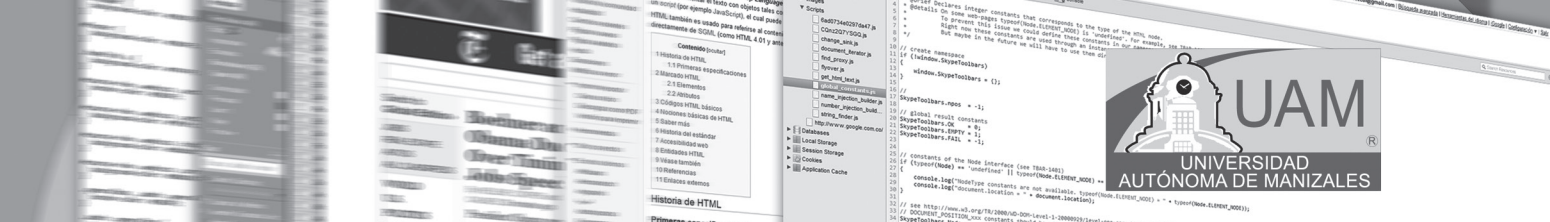
Figura 39. Código 1

```

1 <HTML>
2 <HEAD>
3 <TITLE> COSTADO IZQUIERDO INFERIOR </TITLE>
4 </HEAD>
5
6 <BODY>
7   SE PUEDE INCRUSTAR TABLAS, IMAGENES, BOTONES, TEXTO, ETC...
8 </BODY>
9 </HTML>

```

Figura 40. Código 2



Universidad Autónoma de Manizales

```

1  <HTML>
2  <HEAD>
3  <TITLE> CENTRO DE LA PAGINA</TITLE>
4  </HEAD>
5
6  <BODY>
7    <B><CENTER><B>UNIVERSIDAD AUTÓNOMA DE MANIZALES</B></CENTER></B>
8  </BODY>
9  </HTML>

```

Figura 41. Código 3

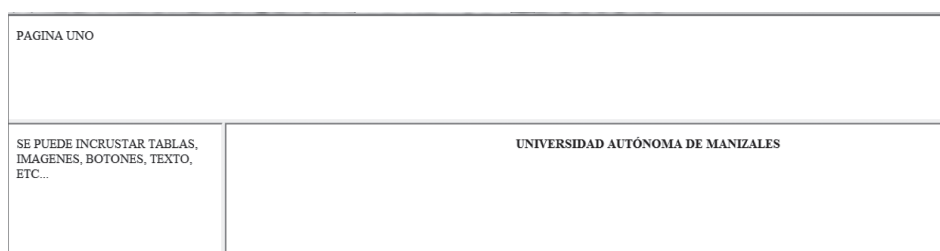


Figura 42. Salida por pantalla usando FRAME

3.2.6. Capas

Las capas no son más que unos recuadros, que pueden situarse en cualquier parte de la página, en los que podemos insertar contenido HTML. Dichas capas pueden ocultarse y solaparse entre sí, lo que proporciona grandes posibilidades de diseño.

Podemos insertar una capa a través de las etiquetas `<div>` y `</div>`, que como ya vimos, sirven para agrupar bloques de texto. A través del atributo `id` se le da un nombre a la capa, y a través del atributo `style` se establecen el resto de propiedades de la capa.

Algunas Ordenes: **left** (izquierda), **top** (superior), **position: absolute** o **relative**, **width** (anchura) y **height** (altura), **z-index** puede establecerse el índice de la capa dentro de la página, **visibility: visible** o **hidden**, con: **layer-background-image** y **background-image** se puede establecer una imagen, con: **layer-background-color** y **background-color** se puede establecer un color de fondo, Hay otras que encuentras en Internet. La figura 43, muestra el código que se usa para crear capas. En la figura 44 se visualiza la salida por pantalla de dicho código.

```

1  <HTML>
2  <HEAD>
3  <TITLE>USANDO CAPAS</TITLE>
4  </HEAD>
5  <BODY>
6    <div id="capa" style="position:relative;
7      width:200px; height:115px;
8      z-index:3; left: 200px; top: 100px;
9      visibility: visible; background-color: #0099CC;
10     layer-background-color: #0099CC;
11     background-image: url(imagenes/fondocapa.gif);
12     layer-background-image: url(imagenes/fondocapa.gif);
13     clip: rect(3 auto auto 3); ">
14    <BR>
15    <P ALIGN="CENTER">Este texto est&aacute; dentro de una capa. UNIVERSIDAD
16    AUTONOMA DE MANIZALES.</P>
17  </div>
18  </BODY>
19  </HTML>

```

Figura 43. Código para hacer capas en HTML



SESION NÚMERO TRES HTML AVANZADO

Este texto está dentro de una capa. UNIVERSIDAD AUTONOMA DE MANIZALES.

Figura 44. Ejemplo de Salida usando Capas



Practicando el desarrollo WEB

Busque en el AULA VIRTUAL el apartado denominado: “EJEMPLOS DE LA SESIÓN 3”; o en el CD adjunto al material escrito, los ejemplos de la SESIÓN_3. Abra cada uno de los ejemplos, usando el ambiente adecuado; Digite o copie el programa y observe el resultado. Interiorice lo visto hasta el momento, mediante el análisis de cada uno de los ejemplos propuestos.



Consultemos

Busque en la red o alguno de los textos propuestos en la bibliografía: ¿Qué es XML y cuáles son reglas? Organice su respuesta y escríbala en un archivo de texto; una vez tenga el archivo grabado en su computador, ingrese al Aula Virtual y busque el link marcado como CONSULTA No 3. Ingrese y envíelo como un adjunto.



Actividad

Ingrese a AULA VIRTUAL y desarrolle la actividad marcada como: ACTIVIDAD No 3. Esta le ayudará a reforzar lo planteado en este módulo.



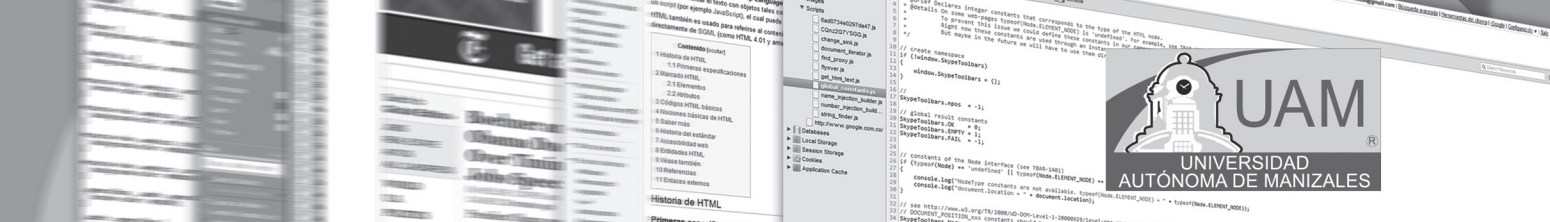
Aplicando lo aprendido

Ingrese al Aula Virtual y ubique en la Sesión el apartado denominado: “LABORATORIO DE SESIÓN”, despliegue el archivo LABORATORIO No 3, realice el algoritmo pedido luego de estar seguro que cumple las pruebas de escritorio, envíelo al docente en el link denominado: Entrega LABORATORIO No 3.



En el Aula Virtual

Encontrará documentación adicional que le servirá para ampliar los conocimientos, frente a las temáticas tratadas en esta sesión. Revíselos y consulte a sus tutores si surge alguna inquietud.



Universidad Autónoma de Manizales

3.3. CONCLUSIONES

- La CGI es el Common Gateway Interface es decir la Interfaz Común de Pasarela, que permite a los usuarios interactuar con algún desarrollo web.
- Las fases fundamentales en la elaboración de plantillas son: la impostación de las marcas para la creación de la plantilla, de los campos y del botón de envío; y la elaboración de un script CGI en servidor (o llamamiento de un script ya impostado).
- La marca INPUT es la que nos permite clasificar la entrada de datos en un formulario, esta puede ser: de texto, password, lista, etc.
- Los marcos o frames, permiten dividir la pantalla en diversas zonas que pueden actuar independientemente unas de otras.
- La directiva <FRAME> indica las propiedades de cada subventana.
- Las capas no son más que unos recuadros, que pueden situarse en cualquier parte de la página, en los que podemos insertar contenido HTML. Dichas capas pueden ocultarse y solaparse entre sí, lo que proporciona grandes posibilidades de diseño.



```
51 DOCUMENT_POSITION_PRECEDING : 0x04,  
52 DOCUMENT_POSITION_FOLLOWING : 0x08,  
53 DOCUMENT_POSITION_CONTAINS : 0x10,  
54 DOCUMENT_POSITION_CONTAINED_BY : 0x20,  
55 DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC : 0x40  
56 }  
57  
58 // redefine setAttribute function to emulate DOM2 modified num  
59 // http://bugzilla.mozilla.org/show_bug.cgi?id=4911  
60 Element.prototype.setAttribute = Element.prototype.setAttribute  
61 Element.prototype.setAttribute = function(name, val)  
62 {  
63     // save the previous value  
64     var prev = this.getAttribute(name);  
65 }  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000
```



PÁGINAS CON ESTILO Y VALIDACIÓN DE FORMULARIOS CON JAVASCRIPT

```

184 ELEMENT_NODE | 1 |
185 TEXT_NODE | 3 |
186 CDATA_SECTION_NODE | 4 |
187 ENTITY_REFERENCE_NODE | 5 |
188 ENTITY_NODE | 6 |
189 PROCESSING_INSTRUCTION_NODE | 7 |
190 COMMENT_NODE | 8 |
191 DOCUMENT_NODE | 9 |
192 DOCUMENT_TYPE_NODE | 10 |
193 DOCUMENT_FRAGMENT_NODE | 11 |
194 NOTATION_NODE | 12 |
195
196 DOCUMENT_POSITION_DISCONNECTED | 0 |
197 DOCUMENT_POSITION_PRECEDING | 0 |
198 DOCUMENT_POSITION_FOLLOWING | 0 |
199 DOCUMENT_POSITION_CONTAINS | 0 |
200 DOCUMENT_POSITION_CONTAINED_BY | 0 |
201 DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC | 0 |
202
203 Element.prototype.getAttribute = function( attr )
204 {
205     // save the previous value
206     var proto = this.getAttribute( attr );
207 }
    
```

4.1. OBJETIVOS ESPECÍFICOS

- Identificar la definición de páginas con estilo.
- Caracterizar la redefinición de etiquetas.
- Categorizar la forma de separar los estilos en las páginas en cascada.
- Identificar los distintos elementos que conforman la validación de los formularios con Javascript.

4.2. DESARROLLO TEMÁTICO

4.2.1. Hojas con estilo. ¿Qué son las hojas de estilo? Una hoja de estilo permite aumentar el control del diseñador sobre cómo se verán sus páginas Web, asociando un conjunto de propiedades de formato físico a los elementos estándar del lenguaje HTML. En la figura 45, se visualiza un diagrama que permite establecer los cambios que se logran al usar páginas u hojas con estilo... Algunos autores las denominan CSS.



Figura 45. El resultado de las páginas con estilo

De esta forma, el resultado final que observa la persona en su browser es una mezcla entre las características predefinidas para cada comando HTML y la hoja de estilo.

Pare

CSS son las siglas de «Cascade StyleSheet». Se trata de una especificación sobre los estilos físicos aplicables a un documento HTML, y trata de dar la separación definitiva de la lógica (estructura) y el físico (presentación) del documento.

4.2.1.1. Aplicación directa

Tenemos varias posibilidades para definir un estilo: especificarlo directamente en la etiqueta en la que queremos usarlo, definirlo aparte y aplicarlo en las etiquetas que queramos, o definir estilos globales para las etiquetas (que podrán ser cambiados en las que no se desee aplicarlos). La etiqueta que utilizamos es:

<etiqueta STYLE="propiedad1:valor;...;propiedadN:valor"> ... </etiqueta>

Un ejemplo de la declaración de una página con estilo podría ser:

La negrita que vemos <B STYLE="font-size:14pt;color:red">es más grande y está en rojo.

En la figura 46, se puede observar cómo se codificaría para obtener estilos mucho más atractivos al realizar sitios web. En la figura 47, se puede visualizar la salida por pantalla de este código.

```

1 <HTML>
2 <HEAD>
3 <TITLE> USO DE ESTILOS </TITLE>
4 </HEAD>
5 <BODY>
6 La negrita que vemos <B STYLE="font-size:14pt;color:red">es más grande y está en rojo</B>
7 </BODY>
8 </HTML>
    
```

Figura 46. Usando estilos

La negrita que vemos **es más grande y está en rojo**

Figura 47. Salida por pantalla usando estilos

4.2.2. Redefinición de etiquetas

Suele ocurrir que queremos definir estilos que se apliquen a todas las etiquetas del documento, es decir, queremos que todo el documento tenga un cierto tipo de letra, que las tablas tengan otro, que las cabeceras tengan un color determinado, para ello, definiremos estilos globales por medio de la etiqueta <STYLE> ... </STYLE> como se muestra en la figura 48.

```

<STYLE TYPE="text/css">
<!--
Etiqueta1, Etiqueta2 : {propiedad1:valor}
Etiqueta3 : {propiedad1:valor;...;propiedad5:valor}
Etiqueta4 : {propiedad1:valor;...;propiedadT:valor}
.Clase1 : {propiedad1:valor;...;propiedadT:valor}
-->
</STYLE>
    
```

Figura 48. Codificación de páginas con estilo CSS



Es recomendable que definamos estos estilos globales dentro de la cabecera del documento (entre <HEAD> ... </HEAD>) para asegurarnos de que se aplicarán a todas las etiquetas para las que se haya definido un estilo.

4.2.3. Separar estilos

Podemos definir los estilos en un fichero distinto al documento HTML, y después referenciarlo desde el propio documento HTML. Esto lo haremos con la siguiente etiqueta, dentro de la cabecera del documento (entre <HEAD> ... </HEAD>). La etiqueta que usamos para lograr este cometido es:

```
<LINK REL="stylesheet" TYPE="text/css" HREF="URL_Hoja.css">
```

4.2.3.1. Se puede resumir todo esto que se ha dicho en lo siguiente:

1. **HOJA DE ESTILO INTERNA:** se puede incluir el código de los estilos en la cabecera (<head>) del documento. En este caso, hay que utilizar la etiqueta de HTML <style>, indicando que es código de hojas de estilos (CSS). La escritura de dicho código, es muy similar al siguiente:

```
<style type="text/css">  
  
BODY {  
  
background-color: #FEEFCB;  
  
color: #000000;  
  
}</style>  
  
<BODY>....<BODY>
```

Para visualizar mediante un ejemplo esta forma de hacer páginas con estilo, observe con detenimiento la figura 49 y la salida de dicho código, en la figura 50.



```

1 <HTML>
2 <HEAD>
3   <TITLE>Ejemplo con hoja de estilo Interna</TITLE>
4   <STYLE TYPE="text/css">
5   <!--
6     BODY {font-family:Verdana,sans-serif;font-size:x-small;
7         margin-left:0.25in; margin-right:0.25in}
8     H2 {font-family:Verdana,sans-serif;font-size:14pt;color:red}
9     B, TD {font-family:Verdana,sans-serif;font-size:x-small;color:olive}
10    TH {font-family:Verdana,sans-serif;font-size:x-small;
11        color:white;background-color:#0080C0}
12    PRE, TT, CODE {font-family:Courier New,Courier;font-size:9pt;color:maroon}
13  -->
14 </STYLE>
15 </HEAD>
16 <BODY BGCOLOR=white>
17 <H2>Prueba de definición de estilos en un bloque aparte</H2>
18 Como puede verse, la apariencia de esta página queda
19 completamente definida por los estilos que hemos especificado en el bloque
20 STYLE en la cabecera del documento. Los márgenes son más amplios de lo habitual,
21 la <B>negrita</B> tiene un tamaño y un color fijos, los trozos de texto en teletipo
22 como <TT>este fragmento</TT> también tienen definida su fuente, tamaño y color,
23 y vamos a ver cómo quedan las tablas, para finalizar el ejemplo:<P>
24 <CENTER>
25 <TABLE BORDER=1 CELSPACING=2 CELLPADDING=2>
26   <TR> <TH>Cabecera 1</TH> <TH>Cabecera 2</TH> </TR>
27   <TR> <TD>Celda (1,1)</TD> <TD>Celda (1,2)</TD> </TR>
28 </TABLE>
29 </CENTER>
30 </BODY>
31 </HTML>

```

Figura 49. Codificación de CSS interno

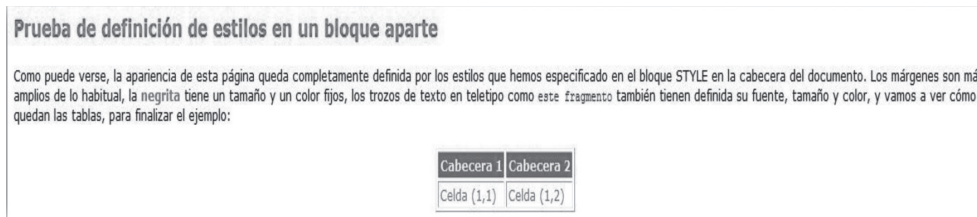
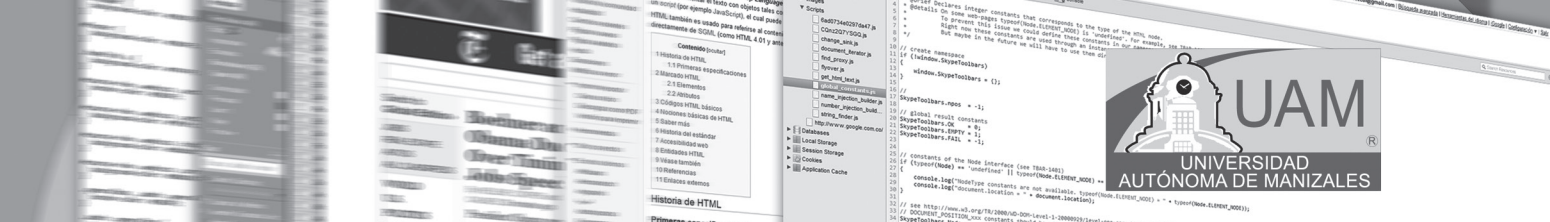


Figura 50. Salida por pantalla de CSS interno

- Para conectar un archivo de hojas de estilos con el resto de archivos HTML que conforman un sitio web hay que colocar una llamada en la cabecera (<head>) de cada documento, en cuya redacción se introduce el nombre del archivo CSS con el atributo href:

```
<link rel="stylesheet" type="text/css" href="estilos.css">
```

Para visualizar mediante un ejemplo esta forma de hacer páginas con estilo, observe con detenimiento la figura 51, donde se ve en forma detallada la página que carga un archivo que contiene los estilos a usar dentro de ella; en la figura 52 se visualiza la página que contiene los estilos que se pueden cargar en una o varias páginas que la llamen para usar los estilos allí definidos (es lo que llamamos CSS externo) y la salida de dicho código, se puede visualizar en la figura 50. Es decir es lo mismo, realizado desde dos enfoques diferentes.



Universidad Autónoma de Manizales

```

1 <HTML>
2 <HEAD>
3 <TITLE> Ejemplo con hoja de estilo externa </TITLE>
4 <LINK REL="stylesheet" TYPE="text/css" HREF="S4_ejemplo_CSS.css">
5 </HEAD>
6 <BODY BGCOLOR=white>
7 <H2>Prueba de definición de estilos en una hoja de
8 estilo externa</H2>
9 Como puede verse, la apariencia de esta página queda
10 completamente definida por los estilos que hemos
11 especificado en el bloque STYLE en la cabecera del
12 documento. Los márgenes son más amplios de lo habitual,
13 la <B>negrita</B> tiene un tamaño y un color fijos, los
14 trozos de texto en teletipo como <TT>este fragmento</TT>
15 también tienen definida su fuente, tamaño y color,
16 y vamos a ver cómo quedan las tablas, para finalizar
17 el ejemplo: <P>
18 <CENTER>
19 <TABLE BORDER=1 CELSPACING=2 CELLPADDING=2>
20 <TR> <TH>Cabecera 1</TH> <TH>Cabecera 2</TH> </TR>
21 <TR> <TD>Celda (1,1)</TD> <TD>Celda (1,2)</TD> </TR>
22 </TABLE>
23 </CENTER>
24 </BODY>
25 </HTML>

```

Figura 51. Página que usará estilos de un archivo externo

```

1 /* Definición de estilos en un archivo aparte */
2 /* Estilo para el documento */
3 BODY {font-family:Verdana,sans-serif;font-size:x-small;
4 margin-left:0.25in; margin-right:0.25in}
5
6 /* Estilo para la cabecera de nivel 2 */
7 H2 {font-family:Verdana,sans-serif; font-size:14pt;color:red}
8
9 /* Estilos para otras etiquetas */
10 B, TD {font-family:Verdana,sans-serif;font-size:x-small;
11 color:olive}
12 TH {font-family:Verdana,sans-serif;font-size:x-small;
13 color:white;background-color:#0080C0}
14 PRE, TT, CODE {font-family:Courier New,Courier;
15 font-size:9pt;color:maroon}

```

Figura 52. Página que define los estilos

4.2.4. Validación de formularios con javascript

4.2.4.1. ¿Dónde y cómo incluir javascript?

La forma más frecuente de hacerlo es utilizando la directiva <script> en un documento HTML (se pueden incluir tantas directivas <script> como se quiera en un documento). El formato es el siguiente:



SESION NÚMERO CUATRO PÁGINAS CON ESTILO Y VALIDACIÓN DE FORMULARIOS CON JAVASCRIPT

```
<script language="Javascript 1.x">
```

El atributo lenguaje hace referencia a la versión de Javascript que se va a utilizar en dicho script.

4.2.4.2. ¿Para qué usamos javascript?

Al realizar páginas WEB o sitios WEB, en muchas ocasiones nos veremos en la necesidad de crear formularios, como se aprendió en la Sesión No 3. La experiencia nos demuestra, que al diseñar y desarrollar dichos formularios, se debe tener presente la validación de algunos campos, que un usuario determinado del mismo, podría llegar confundir la información que debe ser diligenciada. Para lograr este cometido se usan funciones; estas permiten comparar caracteres permitidos e indicar que los campos sean diligenciados. Veamos algunos de ellos.

1. La primer línea que escribimos después del </HEAD> es:

```
<script language="javascript">
```

esto indicará que iniciaremos a escribir código de Javascript. Seguidamente se declara la función de validación.

2. La función que define el formulario a validar es: **funcion validacion(formulario)** { de esta forma se inicia la validación de los formularios.
3. Dentro de la validación se escriben los caracteres y formatos aceptados en los respectivos campos. Para validar campos, en la medida que ese o esos campos acepten determinados caracteres, se usa la siguiente instrucción:

```
var er_nombres = /^[a-z][A-Z]|á|é|í|ó|ú|ñ|ü|\s|\.|-)+$/
```

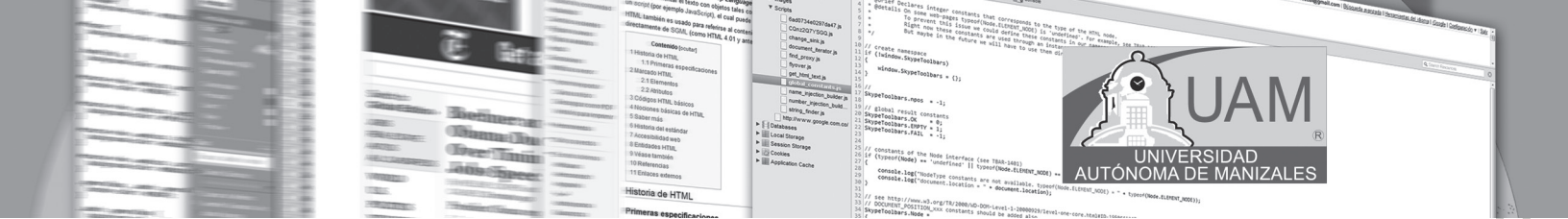
Es importante observar que en este caso se define inicialmente la variable con **var**; luego va el nombre de dicha variable a validar, iniciando con **er_** se escribe una igualdad y se inicia la serie de caracteres que se aceptan con un **/^**(; dentro encontramos [a-z], esto significa que se acepta el abecedario en minúscula; el siguiente grupo de caracteres aceptados es separado por un **|**; observe que luego se ha escrito [A-Z] significa que se acepta también el abecedario en mayúsculas. De igual forma se separan y escriben los otros caracteres aceptados en el campo llamado nombre. Finalmente, luego del carácter final aceptado, se cierra con **|\s|\.|-)+\$/**

4. La siguiente rutina que se muestra es muy usada para dar forma a las fechas que generalmente se toman en los formularios; ver figura 53.

```
//fechas, formato dd/mm/aaaa o d/m/aa
var er_mes31dias = /^[1-3]0|[0-2][1-9]|31|[0-9]\/(1|01|3|03|5|05|7|07|8|08|10|12)\/(1999|20[0-1][0-9]|2020)$/
var er_mes30dias = /^[1-3]0|[0-2][1-9]|[0-9]\/(4|04|6|06|9|09|11)\/(1999|20[0-1][0-9]|2020)$/
var er_mes28dias = /^[1-2]0|[0-2][1-8]|[0-1]9|[0-9]\/(02|2)\/(1999|200[1-3]|200[5-7]|2009|201[0-1]|201[3-5]|201[7-9])$/
var er_mes29dias = /^[1-2]0|[0-2][1-9]|[0-9]\/(02|2)\/(2000|2004|2008|2012|2016|2020)$/
```

Figura 53. Formato para validar fechas

5. Una vez se han indicado los respectivos caracteres y formatos aceptados, se procede a



Universidad Autónoma de Manizales

escribir la rutina de comprobación para cada uno de los campos. Por ejemplo, si queremos validar el campo nombre escrito con anterioridad se escribiría la siguiente rutina:

```
//comprueba campo de nombre  
  
if(!er_nombres.test(formulario.nombres.value)) {  
  
    alert('Contenido del campo NOMBRE no válido.')  
    return false  
  
}
```

Para comprobar la fecha, se escribiría la rutina que se muestra en la figura 54.

```
//comprueba la fecha segun calendario de nacimiento(hasta el 2020, ojo)  
if (!(er_mes31dias.test(formulario.fechanace.value) ||  
    er_mes30dias.test(formulario.fechanace.value) ||  
    er_mes29dias.test(formulario.fechanace.value) ||  
    er_mes28dias.test(formulario.fechanace.value))) {  
    alert('Contenido del campo FECHA DE NACIMIENTO no válido.')    return false  
}
```

Figura 54. Validación del formato fecha

6. Al final de la validación de todos los campos escribimos:

```
}  
  
</script>
```

Esto indica que se ha cerrado todo el código escrito en lenguaje Javascript. No se puede olvidar que todo esto se escribe antes del BODY.

7. La primera línea que escribimos dentro del body es: (ver figura 55)

```
<!-- Para validar el formulario -->  
<form onsubmit="return validacion(this)" name="Formvalidacion" METHOD="POST" action="a otro lado">  
<!-------FIN VALIDACION FORM----->
```

Figura 55. Línea de validación de formulario

Esta línea que se muestra en la figura 55, permite validar el formulario escrito anteriormente.



EJEMPLO DE VALIDACION DE FORMULARIO:

```

<html>
<head>
<title>SUSCRIPCIÓN</title>
</head>
<script language="javascript">
//VALIDACION
function validacion(formulario) {
    var er_nombres = /^[a-z][A-Z]|á|é|í|ó|ú|ñ|ü|s|\.|-)+$/
    var er_apellidos = /^[a-z][A-Z]|á|é|í|ó|ú|ñ|ü|s|\.|-)+$/
    var er_pais = /^[a-z][A-Z]|á|é|í|ó|ú|ñ|ü|s|\.|-)+$/
    var er_departamento = /^[a-z][A-Z]|á|é|í|ó|ú|ñ|ü|s|\.|-)+$/
    var er_direccion = /^[a-z][A-Z][0-9\s+\-]|á|é|í|ó|ú|ñ|ü|s|\.|-)+$/
        //fechas, formato dd/mm/aaaa o d/m/aa
    var er_mes31dias = /^[1-3]0|[0-2][1-9]|31|[0-9]\|(1|01|3|03|5|05|7|07|8|08|10|12)\|(1999|20[0-1][0-9]|2020)$/
    var er_mes30dias = /^[1-3]0|[0-2][1-9]|[0-9]\|(4|04|6|06|9|09|11)\|(1999|20[0-1][0-9]|2020)$/
    var er_mes28dias = /^[1-2]0|[0-2][1-8]|[0-1]9|[0-9]\|(02|2)\|(1999|200[1-3]|200[5-7]|2009|201[0-1]|201[3-5]|201[7-9])$/
    var er_mes29dias = /^[1-2]0|[0-2][1-9]|[0-9]\|(02|2)\|(2000|2004|2008|2012|2016|2020)$/

        //comprueba campo de nombre
    if(!er_nombres.test(formulario.nombres.value)) {
        alert(«Contenido del campo NOMBRE no válido.»)
        return false
    }

    //comprueba campo de apellido
    if(!er_apellidos.test(formulario.apellidos.value)) {
        alert(«Contenido del campo APELLIDOS no válido.»)
        return false
    }

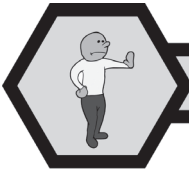
    //comprueba campo de pais
    if(!er_pais.test(formulario.pais.value)) {
        alert(«Contenido del campo PAIS no válido.»)
        return false
    }

    //comprueba campo de departamento
    if(!er_departamento.test(formulario.departamento.value)) {
        alert(«Contenido del campo DEPARTAMENTO no válido.»)
        return false
    }

    //comprueba campo direccion
    if(!er_direccion.test(formulario.direccion.value)) {
        alert(«Contenido del campo DIRECCION no válido.»)
    }
}

```


Universidad Autónoma de Manizales



Pare

De Javascript se puede escribir bastante, máxime cuando se puede hacer programación. Para efectos de este curso usaremos únicamente lo necesario para la gestión de formularios y su aplicabilidad en la gestión de información que se encuentra ligada directamente con bases de datos. Esto será un tema que se tocará más adelante.



Practicando el desarrollo WEB

Busque en el AULA VIRTUAL el apartado denominado: "EJEMPLOS DE LA SESIÓN 4"; o en el CD adjunto al material escrito, los ejemplos de la SESIÓN_4. Abra cada uno de los ejemplos, usando el ambiente adecuado. Digite o copie el programa y observe el resultado. Interiorice lo visto hasta el momento, mediante el análisis de cada uno de los ejemplos propuestos.



Consultemos

Busque en la red o alguno de los textos propuestos en la Bibliografía: ¿Cómo se puede hacer programación mediante el lenguaje de programación Javascript? Realice un ejemplo simple. Organice su respuesta y escríbala en un archivo de texto; una vez tenga el archivo grabado en su computador, ingrese al Aula Virtual y busque el link marcado como CONSULTA No 4. Ingrese y envíelo como un adjunto.



Actividad

Ingrese a AULA VIRTUAL y desarrolle la actividad marcada como: ACTIVIDAD No 4. Esta le ayudará a reforzar lo planteado en este módulo.



Aplicando lo aprendido

Ingrese al Aula Virtual y ubique en la Sesión el apartado denominado: "LABORATORIO DE SESIÓN", despliegue el archivo LABORATORIO No 4, realice el algoritmo pedido luego de estar seguro que cumple las pruebas de escritorio, envíelo al docente en el link denominado: Entrega LABORATORIO No 4.



En el Aula Virtual

encontrará documentación adicional que le servirá para ampliar los conocimientos, frente a las temáticas tratadas en esta sesión. Revíselos y consulte a sus tutores si surge alguna inquietud.



4.3. CONCLUSIONES

- Una hoja de estilo permite aumentar el control del diseñador sobre cómo se verán sus páginas Web, asociando un conjunto de propiedades de formato físico a los elementos estándar del lenguaje HTML.
- Las etiquetas en las páginas CSS, son las que permiten indicar las distintas propiedades que generalizan el estilo de una o varias páginas.
- Los estilos de las páginas CSS, pueden definirse de forma interna y de forma externa por invocación de la página que contiene los distintos estilos.
- Los programas Javascript van incrustados en los documentos HTML, y se encargan de realizar acciones en el cliente, como pueden ser pedir datos, confirmaciones, mostrar mensajes, crear animaciones, comprobar campos, etc.

INICIANDO EL LENGUAJE DE PROGRAMACIÓN PHP

5.1. OBJETIVOS ESPECÍFICOS

- Identificar las características básicas del lenguaje de programación php.
- Detectar la forma de trabajo del servidor de aplicaciones.
- Reconocer la sintaxis del lenguaje de programación php.
- Identificar los distintos elementos que forman y con los que se puede trabajar el lenguaje de programación php.
- Distinguir los distintos operadores en php.

5.2. DESARROLLO TEMÁTICO

5.2.1. ¿Qué es PHP?

PHP es un lenguaje de scripting que permite la navegación dinámica de contenidos en un servidor web, su nombre oficial es PHP: HyperText Preprocessor. Entre sus principales características se pueden destacar:

- Potente y robusto lenguaje de programación embebido en documentos HTML.
- Alto rendimiento y fácil uso.
- Dispone de librerías de conexión con la mayoría de los sistemas de gestión de bases de datos para el almacenamiento de información permanente en el servidor.
- A partir de la versión cinco (5) se puede trabajar con el paradigma de programación orientada a objetos, lo que permite mayor versatilidad en el desarrollo y mayor aceptación en el mercado.

5.2.1.1. Ventajas de PHP

- **Gratuito:** no es necesario realizar ningún desembolso económico para desarrollar sistemas de información empleando este versátil lenguaje.
- **Portable y multiplataforma:** existen versiones del intérprete para múltiples plataformas (Windows, Unix, Linux, etc). Esto permite que las aplicaciones puedan ser portadas de una plataforma a otra sin necesidad de modificar una sola línea de código.
- **Eficiente:** PHP consume muy pocos recursos en el servidor, por lo que con un equipo relativamente sencillo es posible desarrollar interesantes aplicaciones.

5.2.1.2. ¿Dónde se ejecuta el código de PHP?

PHP se ejecuta en el servidor antes que la página sea enviada al usuario que realizó la petición. Esto incluye los siguientes pasos:



SESION NÚMERO CINCO INICIANDO EL LENGUAJE DE PROGRAMACIÓN PHP

- El usuario pulsa sobre un enlace solicitando un documento.
- Llega la solicitud al servidor y localiza el documento. Por la extensión del archivo se determina que es un archivo que contiene código PHP y lanza el intérprete.
- El intérprete lanza el script solicitado y genera un resultado (habitualmente una página HTML) que se devuelve al servidor para que éste a su vez lo transfiera al cliente.
- Se visualiza el documento en el navegador del usuario.

5.2.2. Trabajemos con PHP

PHP es un lenguaje que se embebe dentro de código HTML, por tanto se hace necesario señalar el punto o puntos específicos donde se utiliza el lenguaje php. Lo anterior se logra haciendo uso de etiquetas, éstas son:

1. La forma corta o mejor LA MAS USADA:

```
<? Abrir modo php
```

```
?> Cerrar modo php
```

2. OTRA FORMA:

```
<?php y ?>
```

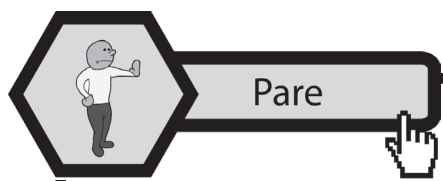
Cualquiera de estas formas tiene el mismo efecto.

5.2.3. Instalación del servidor de aplicaciones web para PHP

Para correr cualquier programa realizado en php, se hace necesario contar con un servidor que permita visualizar estas aplicaciones; El servidor que se usa es el Apache. Hace algún tiempo instalar este servidor era un poco complicado, pero los tiempos han cambiado; hoy día se cuenta con programas como el Appserv, el cual es una herramienta OpenSource para Windows que facilita la instalación de Apache, MySQL y PHP. Estas tres aplicaciones se configuran en forma automática y en forma muy rápida. Como valor agregado incorpora phpMyAdmin para el manejo de MySQL. (**Ver anexo de instalaciones: Instalando Appserv**)

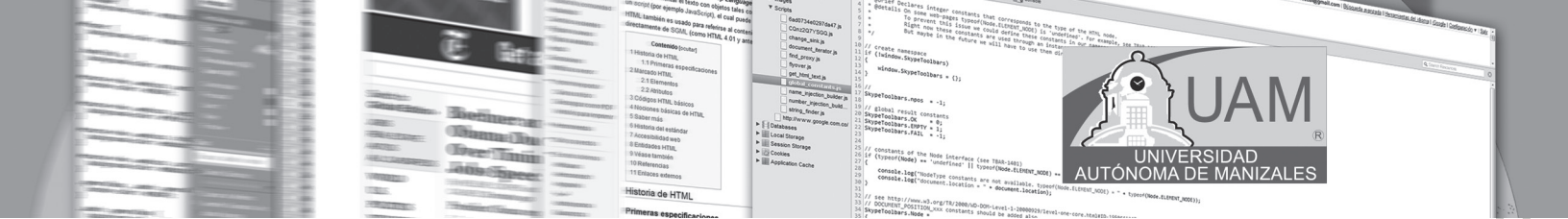
5.2.3.1. Instrucciones

Instrucción **Echo/Print**: se utiliza para imprimir algo en pantalla (cadena, variable). Tenga presente que todos los archivos que se trabajará de ahora en adelante deben ser colocados dentro del root del servidor (c:\appserv\www\) o en htdocs.



Pare

Para la ejecución de un programa realizado en php, se debe abrir una sesión en el navegador que se desee: internet explorer, mozilla, etc, y escribir la siguiente línea: http://localhost/nombre_archivo Tenga presente escribir el nombre del archivo en forma correcta, con su extensión .php; al grabar un archivo realizado en php no se puede olvidar esta extensión para ser reconocido por el navegador.



Universidad Autónoma de Manizales

5.2.4. Variables

En PHP, las variables se representan como un signo de dólar seguido por el nombre de la variable. El nombre de la variable es sensible a minúsculas y mayúsculas. **No se requiere declararlas.**

Ejemplo 1:

```

<html>
<head>
<title>Ejemplo 1. Sesion 5</title>
</head>
<body bgcolor="#FFFFFF">
<?
    $variable1="Curso diseño Web: ";
    $variable2="HTML, PHP y MYSQL";
    echo $variable1." ".$variable2;
?>
</body>
</html>

```

Notas:

- Observe el carácter punto para concatenar cadenas.
- El punto y coma (;) se utiliza al final de cada instrucción de PHP.

Figura 57. Aspectos importantes de PHP

Pare

En el párrafo anterior comentábamos que, para PHP, las variables eran definidas anteponiendo el símbolo dólar (\$) al nombre de la variable que estábamos definiendo.

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

Variables numéricas		
Almacenan cifras		
Enteros	\$entero=2002;	Numeros sin decimales
Real	\$real=3.14159;	Numeros con o sin decimal

Figura 58. Variables numéricas en PHP



SESION NÚMERO CINCO INICIANDO EL LENGUAJE DE PROGRAMACIÓN PHP

Variables alfanuméricas		
Almacenan textos compuestos de números y/o cifras		
Cadenas	Almacenan variables alfanuméricas	<code>\$cadena="Hola amigo";</code>

Figura 59. Otras variables en PHP

5.2.4.1. Objetos

Se trata de conjuntos de variables y funciones asociadas. Presentan una complejidad mayor que las variables vistas hasta ahora, pero su utilidad es más que interesante.

Tablas		
Almacenan series de informaciones numéricas y/o alfanuméricas		
Arrays	Son las variables que guardan las tablas	<code>\$sentido[1]="ver";</code> <code>\$sentido[2]="tocar";</code> <code>\$sentido[3]="oir";</code> <code>\$sentido[4]="gusto";</code> <code>\$sentido[5]="oler";</code>

Figura 60. Tratamiento de objetos

Aplicación directa para nuestro sitio web. Enumeramos algunas de las variables y la información que nos aportan: consideramos las siguientes variables:

\$HTTP_USER_AGENT

Nos informa principalmente sobre el sistema operativo y tipo y versión de navegador utilizado por el internauta. Su principal utilidad radica en que, a partir de esta información, podemos redireccionar nuestros usuarios hacia páginas optimizadas para su navegador o realizar cualquier otro tipo de acción en el contexto de un navegador determinado.

\$HTTP_ACCEPT_LANGUAGE

Nos devuelve la o las abreviaciones de la lengua considerada como principal por el navegador. Esta lengua o lenguas principales pueden ser elegidas en el menú de opciones del navegador. Esta variable resulta también extremadamente útil para enviar al internauta a las páginas escritas en su lengua, si es que existen.

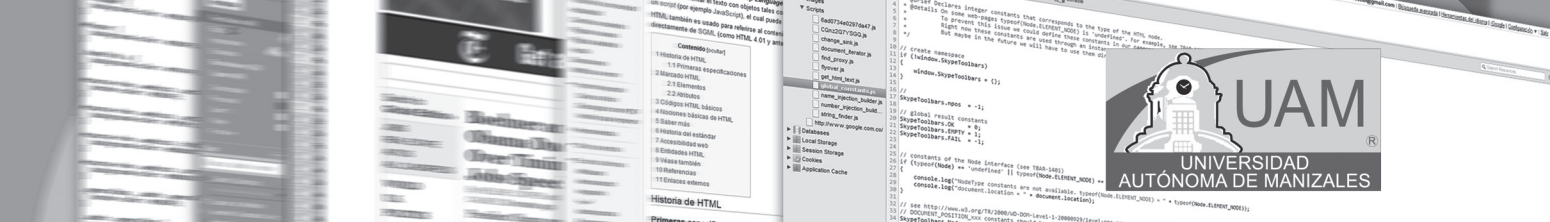
\$HTTP_REFERER

Nos indica la URL desde la cual el internauta ha tenido acceso a la página. Muy interesante para generar botones de "Atrás" dinámicos o para crear nuestros propios sistemas de estadísticas de visitas.

\$PHP_SELF

Nos devuelve una cadena con la URL del script que está siendo ejecutado. Muy interesante para crear botones para recargar la página.

\$HTTP_GET_VARS



Universidad Autónoma de Manizales

Se trata de un array que almacena los nombres y contenidos de las variables enviadas al script por URL o por formularios GET.

\$HTTP_POST_VARS

Se trata de un array que almacena los nombres y contenidos de las variables enviadas al script por medio de un formulario POST

\$HTTP_COOKIES_VARS

Se trata de un array que almacena los nombres y contenidos de las *cookies*. Veremos qué son más adelante.

\$PHP_AUTH_USER

Almacena la variable usuario cuando se efectúa la entrada a páginas de acceso restringido. Combinado con \$PHP_AUTH_PW resulta ideal para controlar el acceso a las páginas internas del sitio.

\$PHP_AUTH_PW

Almacena la variable password cuando se efectúa la entrada a páginas de acceso restringido. Combinado con \$PHP_AUTH_USER resulta ideal para controlar el acceso a las páginas internas del sitio.

\$REMOTE_ADDR

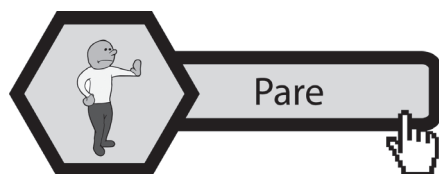
Muestra la dirección IP del visitante.

\$DOCUMENT_ROOT

Nos devuelve el *path físico* en el que se encuentra alojada la página en el servidor.

\$PHPSESSID

Guarda el identificador de sesión del usuario. Veremos más adelante en qué consisten las sesiones.



Las variables, como base de información de un lenguaje, pueden ser creadas, modificadas y comparadas con otras por medio de los llamados operadores.

5.2.5. Operadores. En php se tienen los siguientes operadores:

5.2.5.1 Los operadores aritméticos: nos permiten realizar operaciones numéricas con nuestras variables.



SESION NÚMERO CINCO

INICIANDO EL LENGUAJE DE PROGRAMACIÓN PHP

OPERADORES ARITMETICOS PHP

OPERADOR	NOMBRE	EJEMPLO
+	Adición	\$a + \$b
-	Sustracción	\$a - \$b
*	Multiplicación	\$a * \$b
/	División	\$a / \$b
%	Modulus	\$a % \$b

Figura 61. Operadores aritméticos en PHP

5.2.5.2. Operadores de incremento. Sirven para aumentar o disminuir de una unidad el valor de una variable.

Operadores de preincremento

El operador ++\$A		
Valor inicial de la variable	Sintaxis	Resultado
\$a=23	echo ++\$a	24
\$a=23	echo ++\$a*2	48

El operador --\$A		
Valor inicial de la variable	Sintaxis	Resultado
\$a=23	echo --\$a	22
\$a=23	echo --\$a*2	44

El operador \$A+=n		
Valor inicial de la variable	Sintaxis	Resultado
\$a=23	echo \$a+=5; echo \$a	28
\$a=23	echo 2*\$a+=5; echo 2*\$a	56

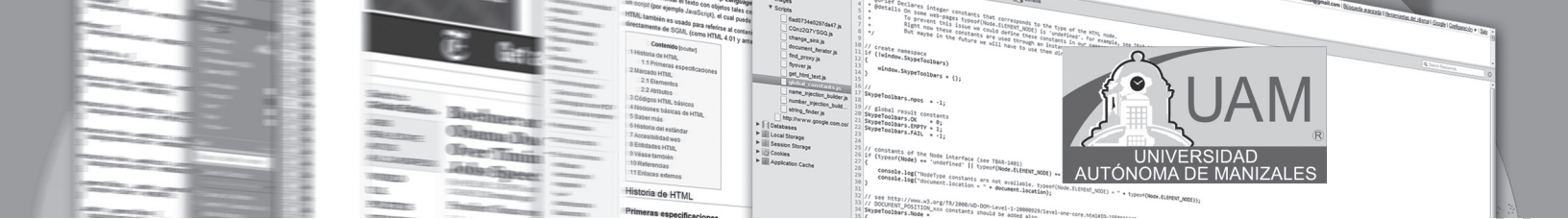
El operador \$A-=n		
Valor inicial de la variable	Sintaxis	Resultado
\$a=23	echo \$a-=5; echo \$a	18
\$a=23	echo 2*\$a-=5; echo 2*\$a	36

Figura 62. Operadores de preincremento

Operadores de post-incremento

El operador \$A++		
Valor inicial de la variable	Sintaxis	Resultado
\$a=23	echo \$a++; echo \$a	24
\$a=23	echo 2*\$a++; echo 2*\$a	48

Figura 63. Operadores de postincremento



Universidad Autónoma de Manizales

5.2.5.3. Operadores combinados. Una forma habitual de modificar el valor de las variables es mediante los operadores combinados:

OPERADORES COMBINADOS DE ASIGNACIÓN PHP

OPERADOR	JSO	EQUIVALENTE A
+=	$\$a += \b	$\$a = \$a + \$b$
=	$\$a = \b	$\$a = \$a \cdot \$b$
^=	$\$a ^= \b	$\$a = \$a \wedge \$b$
/=	$\$a /= \b	$\$a = \$a / \$b$
%=	$\$a \% = \b	$\$a = \$a \% \$b$
.	$\$a .= \b	$\$a = \$a . \$b$

Figura 64. Operadores Combinados en PHP

5.2.5.4. Operadores de comparación. Se utilizan principalmente en nuestras condiciones para comparar dos variables y verificar si cumple o no la propiedad del operador.

OPERADORES DE COMPARACION PHP

OPERADOR	NOMBRE	USO
==	igual	$\$a == \b
===	idéntico	$\$a === \b
!=	desigual	$\$a != \b
<>	desigual	$\$a <> \b
<	menor que	$\$a < \b
>	mayor que	$\$a > \b

Figura 65. Operadores de comparación

5.2.5.5. Operadores lógicos. Se usan en combinación con los operadores de comparación cuando la expresión de la condición lo requiere.

OPERADORES LÓGICOS PHP

OPERADOR	NOMBRE	USO	RESULTADO
!	NOT	$!\$b$	Devuelve verdadero si $\$b$ es falso y viceversa.
&&	AND	$\$a \&\& \b	Devuelve verdadero si $\$a$ y $\$b$ son verdaderos; en otro caso devuelve falso.
	OR	$\$a \b	Devuelve verdadero si alguno de los dos $\$a$ ó $\$b$ o ambos son verdaderos; en otro caso devuelve falso.
and	AND	$\$a \text{ and } \b	Lo mismo que $\&\&$ pero con baja prioridad.
or	OR	$\$a \text{ or } \b	Lo mismo que $ $ pero con baja prioridad.

Figura 66. Operadores Lógicos



SESION NÚMERO CINCO INICIANDO EL LENGUAJE DE PROGRAMACIÓN PHP

Ejemplo 2:

```

<html>
<head>
<title>Ejemplo 2</title>
</head>
<body bgcolor="#FFFFFF">
<?
    $a=5; //esto es un entero
    $b=3; //esto es un entero

    $resultado=$a*$b;
    $division=$a/$b;

    echo $resultado;
    echo "<BR>";
    echo $division;
    echo "<BR>";
    echo number_format($division,2);
?>
</body>
</html>

```



Practicando el desarrollo WEB

Busque en el AULA VIRTUAL el apartado denominado: "EJEMPLOS DE LA SESIÓN 5"; o en el CD adjunto al material escrito, los ejemplos de la SESIÓN_5. Abra cada uno de los ejemplos, usando el ambiente adecuado. Digite o copie el programa y observe el resultado. Interiorice lo visto hasta el momento, mediante el análisis de cada uno de los ejemplos propuestos.



Consultemos

Busque en la red o alguno de los textos propuestos en la bibliografía: ¿Qué otros lenguajes de programación para web, existen? Haga un cuadro comparativo. Organice su respuesta y escríbala en un archivo de texto; una vez tenga el archivo grabado en su computador, ingrese al Aula Virtual y busque el link marcado como CONSULTA No 5. Ingrese y envíelo como un adjunto.



Actividad

Ingrese a AULA VIRTUAL y desarrolle la actividad marcada como: ACTIVIDAD No 5. Esta le ayudará a reforzar lo planteado en este módulo.

Universidad Autónoma de Manizales



Aplicando lo aprendido

Ingrese al Aula Virtual y ubique en la Sesión el apartado denominado: "LABORATORIO DE SESIÓN", despliegue el archivo LABORATORIO No 5, realice el algoritmo pedido luego de estar seguro que cumple las pruebas de escritorio, envíelo al docente en el link denominado: Entrega LABORATORIO No 5.



En el Aula Virtual

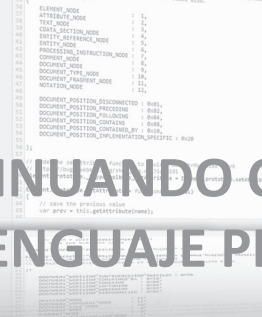
EN EL AULA VIRTUAL Encontrará documentación adicional que le servirá para ampliar los conocimientos, frente a las temáticas tratadas en esta sesión. Revíselos y consulte a sus tutores si surge alguna inquietud.

5.3. CONCLUSIONES

- PHP es un lenguaje de *scripting* que permite la navegación dinámica de contenidos en un servidor web, su nombre oficial es PHP: HyperText preprocessor.
- Entre sus principales características de php se pueden destacar: potente y robusto lenguaje de programación embebido en documentos HTML, alto rendimiento y fácil uso, dispone de librerías de conexión con la mayoría de los sistemas de gestión de bases de datos para el almacenamiento de información permanente en el servidor, a partir de la versión cinco (5) se puede trabajar con el paradigma de programación orientada a objetos, lo que permite mayor versatilidad en el desarrollo y mayor aceptación en el mercado.
- Para correr cualquier programa realizado en php, se hace necesario contar con un servidor que permita visualizar estas aplicaciones; El servidor que se usa es el apache.
- Php es un lenguaje embebido en HTML, su sintaxis es similar al del lenguaje C; todo se escribe entre `<? y ?>`
- Para la aplicación directa de un determinado sitio web se hace uso de variables.
- Los operadores de php son: Aritméticos, de incremento, combinados, de comparación y lógicos.

```
51 DOCUMENT_POSITION_PRECEDING : 0x00,  
52 DOCUMENT_POSITION_FOLLOWING : 0x01,  
53 DOCUMENT_POSITION_CONTAINS : 0x02,  
54 DOCUMENT_POSITION_CONTAINED_BY : 0x03,  
55 DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC : 0x04  
56 }  
57  
58 // redefine setAttribute function to emulate DOM2 modified num  
59 // http://bugzilla.mozilla.org/show_bug.cgi?id=4911  
60 Element.prototype.setAttribute = Element.prototype.setAttribute  
61 Element.prototype.setAttribute = function(name, val)  
62 {  
63     // save the previous value  
64     var prev = this.getAttribute(name);  
65 }  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000
```





6.1. OBJETIVOS ESPECÍFICOS

- Caracterizar el uso de los condicionales mediante el lenguaje php.
- Identificar la forma de uso del condicional IF y SWITCH.
- Caracterizar el uso de los Bucles mediante el lenguaje php.
- Identificar el uso de For, While y do While.
- Reconocer el formato de salida para cadenas de caracteres o números.
- Reconocer la sintaxis para el manejo de cadenas en php.

6.2. DESARROLLO TEMÁTICO

6.2.1. Condicionales

Las sentencias condicionales nos permiten ejecutar o no unas ciertas instrucciones dependiendo del resultado de evaluar una condición. Las más frecuentes son la **instrucción if** y la **instrucción switch**.

Sentencia if ... else Observe en la siguiente figura la sintaxis de la misma.

```

<?php
if (condición)
{
    Sentencias a ejecutar cuando la
    condición es cierta.
}
else
{
    Sentencias a ejecutar cuando la
    condición es falsa.
}
?>
    
```

Figura 67. Sintaxis del if

6.2.1.1. IF...ELSEIF...ELSE

La sentencia IF...ELSEIF...ELSE permite ejecutar varias condiciones en cascada. Para este caso veremos un ejemplo, en el que utilizaremos los operadores lógicos.



SESION NÚMERO SEIS CONTINUANDO CON EL LENGUAJE PHP

```

<?php
if (condición)
{
  Sentencias a ejecutar con condición es cierta.
}
Elseif
{
  Sentencias a ejecutar si condición es falsa.
}
else
{ .....}
?>

```

Ejemplo 1:

```

<html>
<head>
<title>EJEMPLO 1</title>
</head>
<body bgcolor="#FFFFFF">
<?
    $a = 8;
    $b = 3;
    if ($a < $b)
    {
        echo "a es menor que b";
    }
    else
    {
        echo "a no es menor que b";
    }
?>
</body>
</html>

```

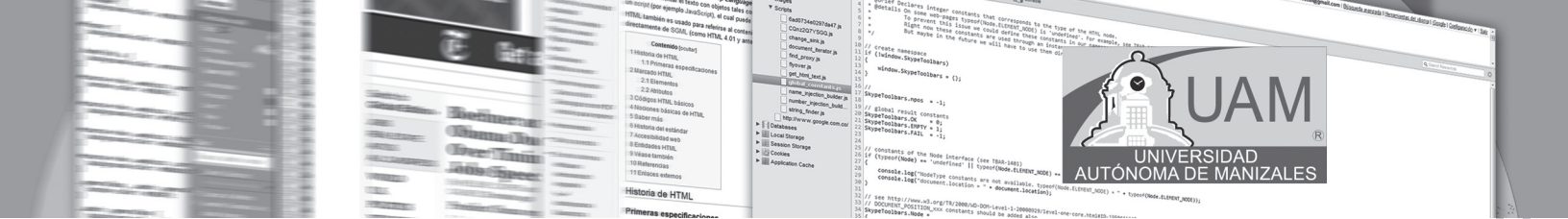
El siguiente ejemplo muestra el uso del *if ... Elseif y else*:

Ejemplo 2:

```

<html>
<head>
<title>EJEMPLO 2</title>
</head>
<body bgcolor="#FFFFFF">

```



Universidad Autónoma de Manizales

```

<?
    $cantidad=58;
    if ($cantidad < "10"){
        echo "Patricia";
    }
    elseif (($cantidad >="11" && ($cantidad <"20"))
        {
            echo "EVA";
        }
    elseif (($cantidad >="20" && ($cantidad <"30"))
        {
            echo "MARITZA";
        }
    else {
        echo "MAFE";
    }
};
?>
</body>
</html>

```

6.2.1.2. SWITCH...CASE...DEFAULT

Una alternativa a IF...ELSEIF...ELSE, es la sentencia SWITCH, la cual evalúa y compara cada expresión de la sentencia CASE con la expresión que evaluamos, si llegamos al final de la lista de CASE y encuentra una condición verdadera, ejecuta el código de bloque que haya en DEFAULT. Si encontramos una condición verdadera debemos ejecutar un BREAK para que la sentencia SWITCH no siga buscando en la lista de CASE.

```

<?php
    switch(condicion) {
        case 1
            si se cumple condicion 1
            break;
        case 2
            si se cumple condicion 2
            break;
            etc.....
        default: o ;
            SI NO SE CUMPLE....
    }
?>

```

Obsérvese el siguiente ejemplo donde se hace uso de estas sentencias:

Ejemplo 3



SESION NÚMERO SEIS CONTINUANDO CON EL LENGUAJE PHP

```

<html>
<head>
<title>EJEMPLO 3</title>
</head>
<body bgcolor="#FFFFFF">
<?
    $dia="Miercoles";
    switch ($dia)
    {

        case "Lunes":
            echo "Hoy es Lunes";
            break;

        case "Martes":
            echo "Hoy es Martes";
            break;

        case "Miercoles":
            echo "Hoy es Miercoles";
            break;

        case "Jueves":
            echo "Hoy es Jueves";
            break;

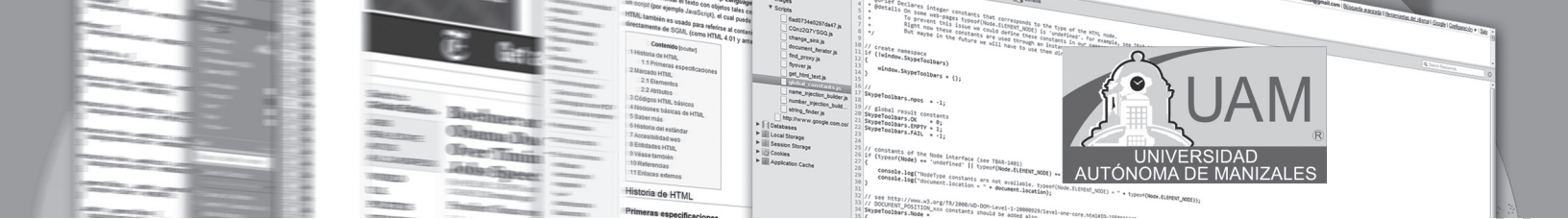
        case "Viernes":
            echo "Hoy es Viernes";
            break;

        case "Sábado":
            echo "Hoy es Sábado";
            break;

        case "Domingo":
            echo "Hoy es Domingo";
            break;

        default;
            echo "Esa cadena no corresponde a ningún día de la semana";
    }
?>
</body>
</html>

```



Universidad Autónoma de Manizales

6.2.2. Bucles o LOOPS

Los bucles nos permiten **iterar conjuntos de instrucciones**, es decir repetir la ejecución de un conjunto de instrucciones mientras se cumpla una condición. Se cuenta con los siguientes LOOPS: Sentencia **while** ; Sentencia **for** ; Sentencia **Do...While** ; veamos cada uno de estos.

6.2.2.1. While

La sentencia while ejecuta un bloque de código mientras se cumpla una determinada condición. Su sintaxis:

```
<?php
while (condición)
{
    intrucciones a ejecutar.
}
?>
```

Ejemplo 4:

```
<html>
<head>
<title>Ejemplo de PHP SWITCH</title>
</head>
<body>
<?php
    $num = 1;
    while ($num < 5) {
        echo $num."<br>";//el . concatena cadenas
        $num++;
    }
?>
<!-- otro ejemplo
<?php
    $i=0;
    while ($i<10)
    {
        echo "El valor de i es ", $i,"<br>";
        $i++;
    }
?>
-->
</body>
</html>
```



SESION NÚMERO SEIS CONTINUANDO CON EL LENGUAJE PHP

6.2.2.2. Sentencia: FOR

El bucle FOR no es estrictamente necesario, cualquier bucle FOR puede ser sustituido fácilmente por otro WHILE. Sin embargo, el bucle FOR resulta muy útil cuando debemos ejecutar un bloque de código a condición de que una variable se encuentre entre un valor mínimo y otro máximo. Su sintaxis es:

```

<?php
    for (inicial ; condición ; ejecutar en iteración)
    {
        instrucciones a ejecutar.
    }
?>

```

Observe el siguiente ejemplo para visualizar el uso de la sentencia FOR.

Ejemplo 5:

```

<html>
<head>
<title>Ejemplo de PHP FOR</title>
</head>
<body>
<?php
    for ($num = 1; $num <=5; $num++)
    {
        echo $num;
    }
?>
</body>
</html>

```

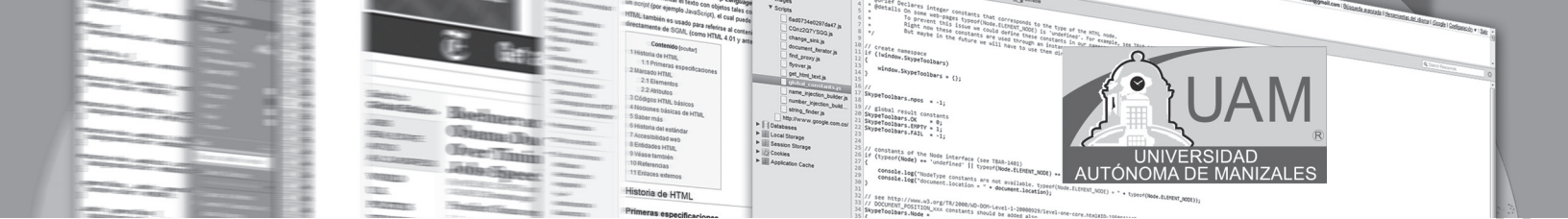
Ejemplo 6:

Obsérvese el siguiente ejemplo:

```

<html>
<head>
<title>Ejemplo de PHP break EN FOR</title>
</head>
<body>
<?php
    for ($num = 1; $num <=5; $num++)
    {
        echo $num;
        if ($num == 3)

```



Universidad Autónoma de Manizales

```

    {
    echo "Aquí nos salimos \n";
    break;
    }
?>
</body>
</html>

```

6.2.2.3. DO...WHILE

Esta sentencia es similar a WHILE, salvo que con esta sentencia primero ejecutamos el bloque de código y después se evalúa la condición, por lo que el bloque de código se ejecuta siempre al menos una vez.

```

<?php
do {
    SENTENCIAS QUE HACE
}
while (CONDICION);
?>

```

Ejemplo 7:

```

<html>
<head>
<title>Ejemplo de PHP BREAK DO WHILE</title>
</head>
<body>
<?php

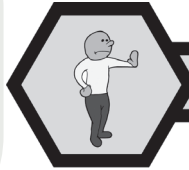
    $num = 1;
    do {
    echo $num;
    if ($num == 3)
    {
    echo "Aquí nos salimos \n";
    break;
    }
    $num++;
    } while ($num < 5);

?>
</body>
</html>

```



SESION NÚMERO SEIS CONTINUANDO CON EL LENGUAJE PHP



Pare

SALIDA POR PANTALLA DE UN PROGRAMA EN PHP. Hasta ahora hemos usado la instrucción echo para realizar salida a pantalla, esta instrucción es bastante limitada ya que no nos permite formatear la salida. La instrucción printf que nos da mucha más potencia. La sintaxis de esta Sentencia printf sería:

```
<?php
printf(cadena formato, variable1, variable2...);
?>
```

Ejemplo 8:

```
<html>
<head>
  <title>Ejemplo de PHP FOR</title>
</head>
<body>
<?php
    $var="texto";
    $num=3;
    printf("Puede fácilmente intercalar <b>%s</b> con números <b>%d</b>
<br>",$var,$num);
    printf("<TABLE BORDER=1 CELLPADDING=10>");
    for ($i=0;$i<5;$i++)
    {
        printf("<tr><td>%.2f</td></tr>",$i);
    }
    printf("</table>");
?>
</body>
</html>
```

6.2.3. Formato de salida para cadenas de caracteres o números

La cadena de formato puede incluir una serie de caracteres especiales que indican como formatear las variables que se incluyen en la instrucción. En la figura 68 se dan algunos elementos con su respectivo tipo de variable.

Elemento	Tipo de variable
%s	Cadena de caracteres.
%d	Número sin decimales.
%f	Número con decimales.
%c	Carácter ASCII.
Aunque existen otros tipos, estos son los más importantes.	

Figura 68. Formato de salida

6.2.4. Manejo de cadenas

Al igual que en otros lenguajes de programación, el manejo de cadenas es fundamental en php. Algunas instrucciones importantes para el manejo de estas cadenas son:

- **strlen(cadena)**. Nos devuelve el número de caracteres de una cadena.
- **split(separador,cadena)**. Divide una cadena en varias usando un carácter separador.
- **sprintf(cadena de formato, var1, var2...)**. Formatea una cadena de texto al igual que printf pero el resultado es devuelto como una cadena.
- **substr(cadena, inicio, longitud)**. Devuelve una subcadena de otra, empezando por inicio y de longitud longitud.
- **chop(cadena)**. Elimina los saltos de línea y los espacios finales de una cadena.
- **strpos(cadena1, cadena2)**. Busca la cadena2 dentro de cadena1 indicándonos la posición en la que se encuentra.
- **str_replace(cadena1, cadena2, texto)**. Reemplaza la cadena1 por la cadena2 en el texto.

Ejemplo 9:

```
<html>
<head>
<title>Ejemplo de CADENAS</title>
</head>
<body>
<?php

//LONGITUD DE LA CADENA
echo strlen("maritzazambrano"),<br>;
echo "<BR>";
//DIVIDE LA CADENA EN OTRAS SEPARADAS
$palabras=split(" ","LA VIDA ES AMOR");
for($i=0;$palabras[$i];$i++)
    echo $palabras[$i],<br>;
echo "<BR>";
//RESULTADO DEVUELTO COMO CADENA
$resultado=sprintf("8x5 = %d <br>",$8*5);
echo $resultado,<br>;
echo "<BR>";
//REEMPLAZA CADENA POR OTRA..COMIENZO Y LONGITUD
echo substr("Devuelve una subcadena de otra",9,3),<br><br>;
    echo "<BR>";
//ELIMINA SALTOS DE LINEA
if (chop("cadena \n\n ") == "cadena")
    echo "Iguales.<br>";
    echo "<BR>";
```



SESION NÚMERO SEIS CONTINUANDO CON EL LENGUAJE PHP

```
//BUSCA PALABRA DENTRO DE LA FRASE
echo strpos("ITC año de la refundacion", "año");<br><br>;
echo "<BR>";
//REEMPLAZA PALABRA
echo str_replace("verde","rojo","Un pez de color verde, como verde es
la hierba.");<br>;
```

```
?>
</body>
</html>
```



Practicando el desarrollo WEB

Busque en el AULA VIRTUAL el apartado denominado: "EJEMPLOS DE LA SESION 6"; o en el CD adjunto al material escrito, los ejemplos de la SESIÓN_6. Abra cada uno de los ejemplos, usando el ambiente adecuado. digite o copie el programa y observe el resultado. Interiorice lo visto hasta el momento, mediante el análisis de cada uno de los ejemplos propuestos.



Consultemos

Busque en la red o alguno de los textos propuestos en la bibliografía: ¿en qué consiste el patrón de diseño web: MVC, es decir el Modelo Vista Controlador? Haga una gráfica que lo represente. Organice su respuesta y escríbala en un archivo de texto; una vez tenga el archivo grabado en su computador, ingrese al Aula Virtual y busque el link marcado como CONSULTA No 6. Ingrese y envíelo como un adjunto.



Actividad

Ingrese a AULA VIRTUAL y desarrolle la actividad marcada como: ACTIVIDAD No 6. Esta le ayudará a reforzar lo planteado en este módulo.



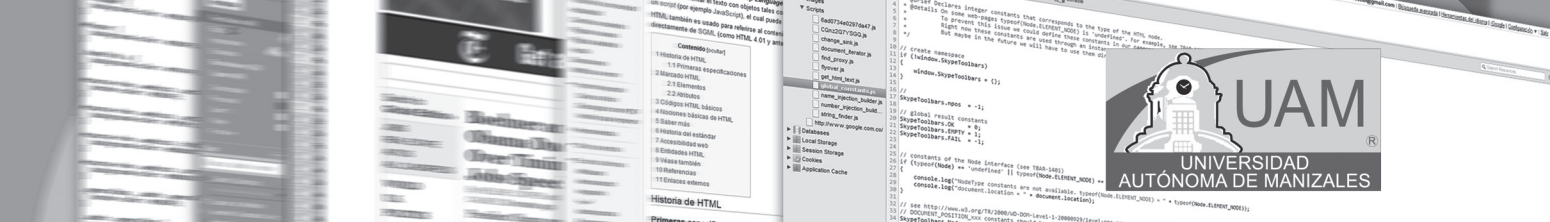
Aplicando lo aprendido

Ingrese al Aula Virtual y ubique en la Sesión el apartado denominado: "LABORATORIO DE SESIÓN", despliegue el archivo LABORATORIO No 6, realice el algoritmo pedido luego de estar seguro que cumple las pruebas de escritorio, envíelo al docente en el link denominado: Entrega LABORATORIO No 6.



En el Aula Virtual

encontrará documentación adicional que le servirá para ampliar los conocimientos, frente a las temáticas tratadas en esta sesión. Revíselos y consulte a sus tutores si surge alguna inquietud.



Universidad Autónoma de Manizales

6.3. CONCLUSIONES

- Las sentencias condicionales nos permiten ejecutar o no unas ciertas instrucciones dependiendo del resultado de evaluar una condición. Las más frecuentes son la instrucción if y la instrucción switch.
- La sentencia IF...ELSEIF...ELSE permite ejecutar varias condiciones en cascada; mientras que en la sentencia SWITCH, la cual evalúa y compara cada expresión de la sentencia CASE con la expresión que evaluamos, si llegamos al final de la lista de CASE y encuentra una condición verdadera, ejecuta el código de bloque que haya en DEFAULT. Si encontramos una condición verdadera debemos ejecutar un BREAK para que la sentencia SWITCH no siga buscando en la lista de CASE.
- Los bucles nos permiten iterar conjuntos de instrucciones, es decir repetir la ejecución de un conjunto de instrucciones mientras se cumpla una condición.
- La sentencia WHILE ejecuta un bloque de código mientras se cumpla una determinada condición; el bucle FOR resulta muy útil cuando debemos ejecutar un bloque de código a condición de que una variable se encuentre entre un valor mínimo y otro máximo. La sentencia do - While es similar a WHILE, salvo que con esta sentencia primero ejecutamos el bloque de código y después se evalúa la condición, por lo que el bloque de código se ejecuta siempre al menos una vez.
- La cadena de formato puede incluir una serie de caracteres. Al igual que otros lenguajes de programación estas permiten el manejo de palabras, frases u oraciones.



```
51 DOCUMENT_POSITION_PRECEDING : 0x04,  
52 DOCUMENT_POSITION_FOLLOWING : 0x08,  
53 DOCUMENT_POSITION_CONTAINS : 0x10,  
54 DOCUMENT_POSITION_CONTAINED_BY : 0x20,  
55 DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC : 0x40  
56 }  
57  
58 // redefine setAttribute function to emulate DOM2 modified num  
59 // http://ha-nakil.org/2006/04/14/01/  
60 Element.prototype.setAttribute = Element.prototype.setAttribute  
61 Element.prototype.setAttribute = function(name, val)  
62 {  
63     // save the previous value  
64     var prev = this.getAttribute(name);  
65 }  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000
```



PROGRAMACIÓN ORIENTADA A OBJETOS CON PHP

```

51 ELEMENT_NODE | 1;
52 ATTRIBUTE_NODE | 2;
53 TEXT_NODE | 3;
54 CDATA_SECTION_NODE | 4;
55 ENTITY_REFERENCE_NODE | 5;
56 ENTITY_NODE | 6;
57 PROCESSING_INSTRUCTION_NODE | 7;
58 COMMENT_NODE | 8;
59 DOCUMENT_NODE | 9;
60 DOCUMENT_FRAGMENT_NODE | 10;
61 NOTATION_NODE | 11;
62
63 DOCUMENT_POSITION_DISCONNECTED | 0;
64 DOCUMENT_POSITION_PRECEDING | 1;
65 DOCUMENT_POSITION_FOLLOWING | 2;
66 DOCUMENT_POSITION_CONTAINS | 3;
67 DOCUMENT_POSITION_CONTAINED_BY | 4;
68 DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC | 5;
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

7.1. OBJETIVOS ESPECÍFICOS

- Identificar el contexto de php bajo el paradigma de programación orientado a objetos.
- Exponer los conceptos básicos del paradigma de programación orientado a objetos para php.
- Reconocer las características fundamentales de programación orientada a objetos para php.
- Identificar los conceptos de atributos y métodos de una clase.
- Reconocer las particularidades de los métodos constructores en php.

7.2. DESARROLLO TEMÁTICO

7.2.1. El contexto. A partir de la versión cinco (5), mediante el lenguaje de programación php, se puede hacer uso del paradigma de programación orientada a objetos. En esencia las definiciones son las mismas que se tomaron al momento de cursar las asignaturas: *Fundamentos de programación orientada a objetos* y *Programación Orientada a Objetos*; en aquella ocasión se usó el lenguaje de programación Java; ahora se hace uso de estos conceptos para sean modelados, mediante el lenguaje de programación Web, denominado php.

El lenguaje PHP tiene la característica de permitir programar con las siguientes metodologías:

- **Programación Lineal.** Es cuando desarrollamos todo el código disponiendo instrucciones PHP alternando con el HTML de la página.
- **Programación Estructurada.** Es cuando planteamos funciones que agrupan actividades a desarrollar y luego dentro de la página llamamos a dichas funciones que pueden estar dentro del mismo archivo o en una librería separada.
- **Programación Orientada a Objetos.** Es cuando planteamos clases y definimos objetos de las mismas.

7.2.2. Conceptos básicos de objetos

Para mayor sencillez y teniendo presente lo indicado en el primer párrafo de este módulo, se tomarán en cuenta las definiciones sencillas de David Tavárez, socializadas a la comunidad académica en: <http://pixelar.me/programacion-orientada-a-objetos-en-php/>. [Visitado el 12 de junio de 2011] En su sitio indica las siguientes, definiciones que resumen lo visto en asignaturas anteriores:





SESION NÚMERO SIETE PROGRAMACIÓN ORIENTADA A OBJETOS CON PHP

7.2.2.1. Objeto. Se refiere comúnmente a atributos (elementos, variables) y comportamientos (métodos, operaciones) encapsulados en una entidad. Son un método natural para implementar estructuras de datos abstractos, trayendo “físicamente” componentes de datos con procedimientos para luego manipularlos. Todo objeto tiene estado, exhibe un comportamiento bien definido y posee identidad única.

7.2.2.2. Clase. Es un tipo de dato definido por el programador que determina las estructuras de datos que lo forman y las funciones asociadas con él, es decir es un modelo con el que se construyen los objetos.

Se define como el constructor utilizado como “plano” para crear un Objeto. Describe el estado y el comportamiento del objeto creado. Un objeto creado por una clase es una instancia de la clase y la clase que creó esa instancia es considerada como el tipo del objeto. Un ejemplo de un objeto creado por una clase “fruta” sería “piña”, otro “pera” y así sucesivamente. Es la abstracción de un concepto.

7.2.2.3. Instancia. Se crea cuando una construye un objeto. Las instancias de una clase comparten los mismos atributos, pero pueden diferir en qué los atributos contengan. Por ejemplo, una clase “Persona” describe los atributos comunes de una persona; cada persona es generalmente parecida, pero varía en atributos como “altura” y “peso”. La clase listaría tipos de atributos, así como también las acciones que una persona podría realizar: “saltar”, “caminar”, “andar”, “comer”. La instancia es el objeto creado en tiempo de corrida de una aplicación. Otro ejemplo sería: “Lassie” es una instancia de la clase “Perro” y así también “Pluto” sería otra instancia de la misma clase.

7.2.3. Características bases de la POO.

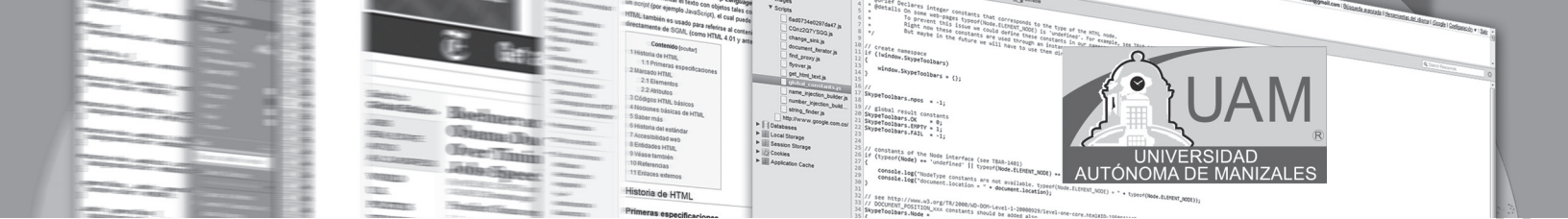
7.2.3.1. Herencia. Es la capacidad formar nuevas Clases usando clases previamente definidas. Las nuevas clases, conocida como Clases Derivadas, toman (o heredan) los atributos y los métodos preexistentes de la clase “madre”. Es un intento para ayudar a reusar código con poca o casi ninguna modificación.

7.2.3.2. Métodos. Se refiere como las funciones de un objeto, haciendo un paralelo con programación estructurada.

7.2.3.3. Abstracción. Es un concepto o idea no asociada a una instancia. El concepto se originó por la analogía con la abstracción en matemáticas. Teóricamente envuelve la facilidad de definir objetos que representan “actores” abstractos que pueden realizar un trabajo o acción. Un ejemplo sería una clase “casa” que defina la estructura de una casa.

7.2.3.4. Encapsulamiento. Se refiere a “esconder” el estado de una propiedad o método de un objeto.

7.2.3.5. Polimorfismo. Es la habilidad de un tipo “A” que aparece y es usado como otro tipo “B”. Significa que un tipo “A” de alguna manera deriva de un tipo “B” o que un tipo “A” implementa una interface que representa un tipo “B”. Por ejemplo, si a un “Perro” se le ordenara que hablara, éste ladraría, mas si a un “Cerdo” se le ordenara que hablara respondería con un “oink oink”. Este último símil es muy simpático.



Universidad Autónoma de Manizales

EJEMPLO 1:

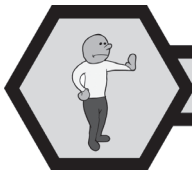
En este primer ejemplo se describe gran parte de estos conceptos dados anteriormente. La figura 69 lo hace notar en forma precisa. Ejemplo tomado junto a su explicación del sitio PHP YA. <http://www.phpya.com.ar/poo/temarios/descripcion.php?cod=36&punto=2&inicio=0>. [Visitado el 13 de junio de 2011]

```

01 <html>
02 <head>
03 <title>Pruebas</title>
04 </head>
05 <body>
06 <?php
07 class Persona {
08     private $nombre;
09     public function inicializar($nom)
10     {
11         $this->nombre=$nom;
12     }
13     public function imprimir()
14     {
15         echo $this->nombre;
16         echo '<br>';
17     }
18 }
19
20 $per1=new Persona();
21 $per1->inicializar('Juan');
22 $per1->imprimir();
23 $per2=new Persona();
24 $per2->inicializar('Ana');
25 $per2->imprimir();
26 ?>
27 </body>
28 </html>

```

Figura 69. Php con POO



Pare

La sintaxis básica para declarar una clase es:

```

class [Nombre de la Clase] {
    [atributos]
    [métodos] }

```



SESION NÚMERO SIETE PROGRAMACIÓN ORIENTADA A OBJETOS CON PHP

EXPLICACIÓN DEL EJEMPLO ANTERIOR:

Siempre conviene buscar un nombre de clase lo más próximo a lo que representa. La palabra clave para declarar la clase es *class*, seguidamente el nombre de la clase y luego encerramos entre llaves de apertura y cerrado todos sus atributos (variable) y métodos (funciones).

Nuestra clase Persona queda definida entonces:

```
class Persona {
    private $nombre;
    public function inicializar($nom)
    {
        $this->nombre=$nom;
    }
    public function imprimir()
    {
        echo $this->nombre;
        echo '<br>';
    }
}
```

Los atributos normalmente son privados (*private*), ya veremos que esto significa que no podemos acceder al mismo desde fuera de la clase; Lo anterior asegura el uso del concepto de **encapsulamiento**. Luego para definir los métodos se utiliza la misma sintaxis que las funciones del lenguaje PHP.

Decíamos que una clase es un molde que nos permite definir objetos. Ahora veamos cual es la sintaxis para la definición de objetos de la clase Persona:

```
$per1=new Persona();
$per1->inicializar('Juan');
$per1->imprimir();
```

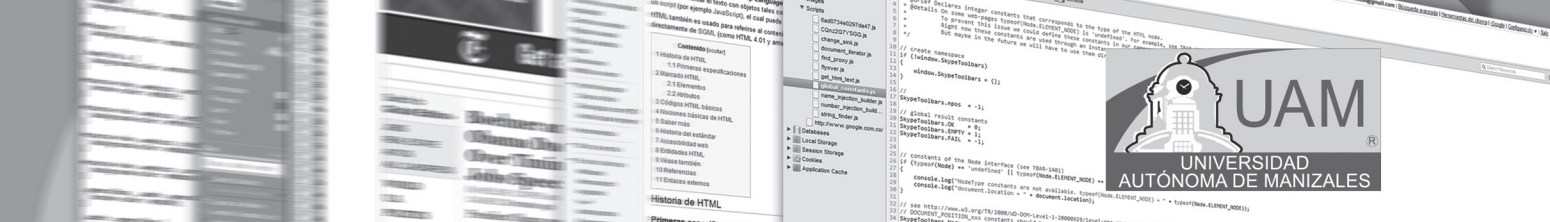
Definimos un objeto llamado *\$per1* y lo creamos asignándole lo que devuelve el operador *new*. Siempre que queremos crear un objeto de una clase utilizamos la sintaxis *new* [Nombre de la Clase]. **No olvide que esto, también se denomina instanciación.**

Luego para llamar a los métodos debemos anteceder el nombre del objeto el operador *->* y, por último, el nombre del método. Para poder llamar al método, éste debe ser definido público (con la palabra clave *public*). En el caso que tenga parámetros se los enviamos:

```
$per1->inicializar('Juan');
```

También podemos ver que podemos definir tantos objetos de la clase Persona como sean necesarios para nuestro algoritmo:

```
$per2=new Persona();
$per2->inicializar('Ana');
$per2->imprimir();
```

Universidad Autónoma de Manizales

Esto nos da una idea que si en una página WEB tenemos 2 menús, seguramente definiremos una clase Menú y luego crearemos dos objetos de dicha clase.

Esto es una de las ventajas fundamentales de la Programación Orientada a Objetos (POO), es decir reutilización de código (gracias a que está encapsulada en clases) es muy sencilla.

Lo último a tener en cuenta en cuanto a la sintaxis de este primer problema es que cuando accedemos a los atributos dentro de los métodos debemos utilizar los operadores **\$this->** (**this y ->**):

```
public function inicializar($nom)
{
    $this->nombre=$nom;
}
```

El atributo \$nombre solo puede ser accedido por los métodos de la clase Persona.

7.2.4. Atributos y métodos de la clase

Cuando creamos un objeto de una clase determinada, los atributos declarados por la clase son localizadas en memoria y pueden ser modificados mediante los métodos. En la figura 70, observamos detenidamente el manejo de los atributos; de igual forma se puede visualizar el uso de los métodos. Los métodos son como las funciones en los lenguajes estructurados, pero están definidos dentro de una clase y operan sobre los atributos de dicha clase. Los métodos también son llamados las responsabilidades de la clase. Para encontrar las responsabilidades de una clase hay que preguntarse qué puede hacer la clase. El objetivo de un método es ejecutar las actividades que tiene encomendada la clase a la cual pertenece. Ejemplo tomado junto a su explicación del sitio PHP YA. [Visitado el 13 de junio de 2011] <http://www.phpya.com.ar/poo/temarios/descripcion.php?cod=36&punto=2&inicio=0>.

```
01 <html>
02 <head>
03 <title>Pruebas</title>
04 </head>
05 <body>
06 <?php
07 class Menu {
08     private $enlaces=array();
09     private $titulos=array();
10     public function cargarOpcion($en,$tit)
11     {
12         $this->enlaces[]=$en;
13         $this->titulos[]=$tit;
14     }
15     public function mostrar()
16     {
17         for($f=0;$f<count($this->enlaces);$f++)
18         {
19             echo '<a href="'. $this->enlaces[$f]. '">'. $this->titulos[$f]. '</a>';
20             echo " ";
21         }
22     }
23 }
24 $menul=new Menu();
25 $menul->cargarOpcion('http://www.google.com','Google');
26 $menul->cargarOpcion('http://www.yahoo.com','Yahoo');
27 $menul->cargarOpcion('http://www.msn.com','MSN');
28 $menul->mostrar();
29 ?>
30 </body>
31 </html>
```

Figura 70. Atributos



SESION NÚMERO SIETE PROGRAMACIÓN ORIENTADA A OBJETOS CON PHP

Como podemos ver normalmente los atributos de la clase se definen inmediatamente después que declaramos la clase:

```
class Menu {
    private $enlaces=array();
    private $titulos=array();
```

En este ejemplo específico, muestra también el uso de vectores. El primer método que añade a los vectores los datos que llegan como parámetro:

```
public function cargarOpcion($en,$tit)
{
    $this->enlaces[ ]=$en;
    $this->titulos[ ]=$tit;
}
```

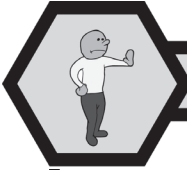
Conviene darle distinto nombre a los parámetros y los atributos (por lo menos inicialmente para no confundirlos). El segundo método, muestra los datos que van alimentando cada uno de los arreglos definidos anteriormente. En este ejemplo disponemos un for y hacemos que se repita tantas veces como elementos tenga el vector \$enlaces (**es lo mismo preguntar a uno u otro cuantos elementos tienen ya que siempre tendrán la misma cantidad**). Para obtener la cantidad de elementos del vector utilizamos la función **count**. Dentro del for imprimimos en la página el hipervínculo:

```
echo '<a href="'. $this->enlaces[$f]."'>'. $this->titulos[$f]. '</a>';
```

Hay que acostumbrarse que cuando accedemos a los atributos de la clase se le antecede el operador **\$this->** y seguidamente el nombre del atributo propiamente dicho. Si no hacemos esto estaremos creando una variable local y el algoritmo fallará. Por último, para hacer uso de esta clase Menu debemos crear un objeto de dicha clase (lo que en programación estructurada es definir una variable):

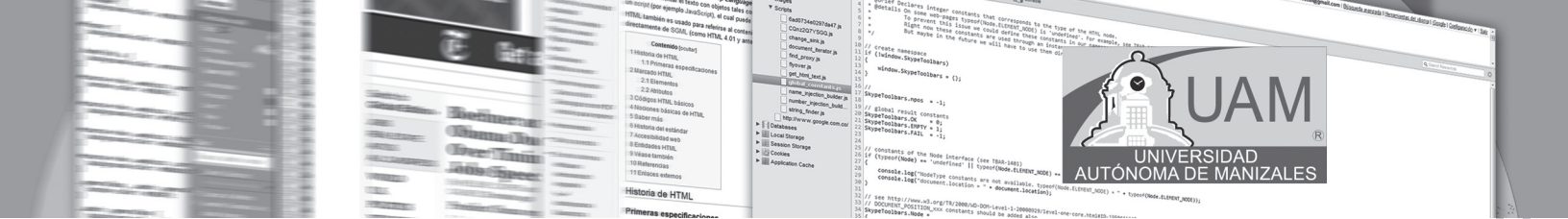
```
$menu1=new Menu();
$menu1->cargarOpcion('http://www.google.com','Google');
$menu1->cargarOpcion('http://www.yahoo.com','Yhahoo');
$menu1->cargarOpcion('http://www.msn.com','MSN');
$menu1->mostrar();
```

Creamos un objeto mediante el operador *new* y seguido del nombre de la clase. Luego llamamos al método *cargarOpcion* tantas veces como opciones necesitemos para nuestro menú (recordar que **SOLO** podemos llamar a los métodos de la clase si definimos un objeto de la misma). Finalmente, llamamos al método *mostrar* que imprime en la página nuestro menú.



Pare

METODO CONSTRUCTOR. El constructor es un método especial de una clase. El objetivo fundamental del constructor es inicializar los atributos del objeto que creamos.



Universidad Autónoma de Manizales

7.2.5. Características de los constructores

- El constructor se ejecuta inmediatamente luego de crear un objeto y no puede ser llamado nuevamente.
- Un constructor no puede retornar dato.
- Un constructor puede recibir parámetros que se utilizan normalmente para inicializar atributos.
- El constructor es un método opcional, de todos modos es muy común definirlo.

Veamos la sintaxis del constructor:

```
public function __construct([parámetros])  
{  
    [algoritmo]  
}
```

Debemos definir un método llamado `__construct` (es decir utilizamos dos caracteres de subrayado y la palabra `construct`). El constructor debe ser un método público (`public function`). Además hemos dicho que el constructor puede tener parámetros. Ver figura 71. Ejemplo tomado junto a su explicación del sitio PHP YA. [Visitado el 13 de junio de 2011]

<http://www.phpya.com.ar/poo/temarios/descripcion.php?cod=36&punto=2&inicio=0>.

```
01 <html>  
02 <head>  
03 <title>Pruebas</title>  
04 </head>  
05 <body>  
06 <?php  
07 class CabeceraPagina {  
08     private $titulo;  
09     private $ubicacion;  
10     public function __construct($tit,$ubi)  
11     {  
12         $this->titulo=$tit;  
13         $this->ubicacion=$ubi;  
14     }  
15     public function graficar()  
16     {  
17         echo '<div style="font-size:40px;text-align:'.$this->ubicacion.'">';  
18         echo $this->titulo;  
19         echo '</div>';  
20     }  
21 }  
22  
23 $cabecera=new CabeceraPagina('El blog del programador','center');  
24 $cabecera->graficar();  
25 >>  
26 </body>  
27 </html>
```

Figura 71. Definición del método constructor



SESION NÚMERO SIETE PROGRAMACIÓN ORIENTADA A OBJETOS CON PHP

Hay que tener mucho cuidado cuando definimos el constructor, ya que el más mínimo error (**nos olvidamos un carácter de subrayado, cambiamos una letra de la palabra *construct***) nuestro algoritmo no funcionará correctamente, ya que nunca se ejecutará este método (ya que no es el constructor).

7.2.6. Herencia

La herencia significa que se pueden crear nuevas clases partiendo de clases existentes, que tendrá todos los atributos y los métodos de su 'superclase' o 'clase padre' y además se le podrán añadir otros atributos y métodos propios. En PHP, a diferencia de otros lenguajes orientados a objetos (C++), una clase sólo puede derivar de una única clase, es decir, PHP no permite herencia múltiple. Utilizamos la palabra clave ***extends*** y seguidamente el nombre de la clase padre (con esto estamos indicando que todos los métodos y atributos de la clase.

7.2.6.1. Superclase o clase padre

Clase de la que desciende o deriva una clase. Las clases hijas (descendientes) heredan (incorporan) automáticamente los atributos y métodos de la clase padre.

7.2.6.2 Subclase

Clase descendiente de otra. Hereda automáticamente los atributos y métodos de su superclase. Es una especialización de otra clase. Admiten la definición de nuevos atributos y métodos para aumentar la especialización de la clase.

El siguiente ejemplo muestra el concepto de herencia en el lenguaje de programación php. Ver las figuras: 72a y 72b. Ejemplo tomado junto a su explicación del sitio PHP YA. [Visitado el 13 de junio de 2011]

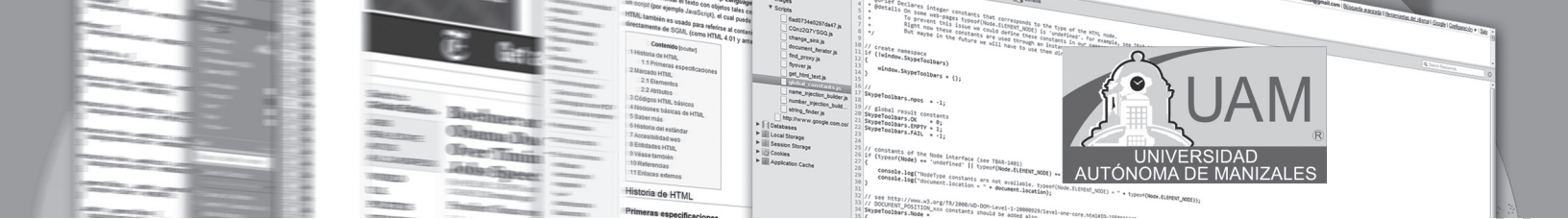
<http://www.phpya.com.ar/poo/temarios/descripcion.php?cod=36&punto=2&inicio=0>.

```

01 <html>
02 <head>
03 <title>Pruebas</title>
04 </head>
05 <body>
06 <?php
07
08 class Operacion {
09     protected $valor1;
10     protected $valor2;
11     protected $resultado;
12     public function cargar1($v)
13     {
14         $this->valor1=$v;
15     }
16     public function cargar2($v)
17     {
18         $this->valor2=$v;
19     }
20     public function imprimirResultado()
21     {
22         echo $this->resultado.'  
';
23     }
24 }
25

```

Figura 72a, Manejo de herencia



Universidad Autónoma de Manizales

```
01 <html>
02 <head>
03 <title>Pruebas</title>
04 </head>
05 <body>
06 <?php
07
08 class Operacion {
09     protected $valor1;
10     protected $valor2;
11     protected $resultado;
12     public function cargar1($v)
13     {
14         $this->valor1=$v;
15     }
16     public function cargar2($v)
17     {
18         $this->valor2=$v;
19     }
20     public function imprimirResultado ()
21     {
22         echo $this->resultado.'<br>';
23     }
24 }
25
```

Figura 72b, Manejo de herencia

7.2.7. Polimorfismo

Una subclase en PHP puede redefinir un método, es decir que podemos crear un método con el mismo nombre que el método de la clase padre. Ahora cuando creamos un objeto de la subclase, el método que se llamará es el de dicha subclase.

Lo más conveniente es sobrescribir métodos para completar el algoritmo del método de la clase padre. No es bueno sobrescribir un método y cambiar completamente su comportamiento.

Veamos nuestro problema de las tres clases: Operación, Suma y Resta. Sobrescribiremos en las subclases el método imprimirResultado (el objetivo es que muestre un título indicando si se trata del resultado de la suma de dos valores o el resultado de la diferencia de dos valores). Esto es lo que en teoría llamamos un tipo de polimorfismo. No olvide los conceptos vistos en POO. Ver figura 73a y 73b. Ejemplo tomado junto a su explicación del sitio PHP YA. [Visitado el 13 de junio de 2011]

<http://www.phpya.com.ar/poo/temarios/descripcion.php?cod=36&punto=2&inicio=0>.



SESION NÚMERO SIETE PROGRAMACIÓN ORIENTADA A OBJETOS CON PHP

```
01 <html>
02 <head>
03 <title>Pruebas</title>
04 </head>
05 <body>
06 <?php
07 class Operacion {
08     protected $valor1;
09     protected $valor2;
10     protected $resultado;
11     public function cargar1($v)
12     {
13         $this->valor1=$v;
14     }
15     public function cargar2($v)
16     {
17         $this->valor2=$v;
18     }
19     public function imprimirResultado ()
20     {
21         echo $this->resultado.'<br>';
22     }
23 }
24
25 class Suma extends Operacion{
26     public function operar()
27     {
28         $this->resultado=$this->valor1+$this->valor2;
29     }

```

Figura 73a. Manejando polimorfismo

```
30     public function imprimirResultado ()
31     {
32         echo "La suma de $this->valor1 y $this->valor2 es:";
33         parent::imprimirResultado ();
34     }
35 }
36 class Resta extends Operacion{
37     public function operar ()
38     {
39         $this->resultado=$this->valor1-$this->valor2;
40     }
41     public function imprimirResultado ()
42     {
43         echo "La diferencia de $this->valor1 y $this->valor2 es:";
44         parent::imprimirResultado ();
45     }
46 }
47 $suma=new Suma ();
48 $suma->cargar1 (10);
49 $suma->cargar2 (10);
50 $suma->operar ();
51 $suma->imprimirResultado ();
52 $resta=new Resta ();
53 $resta->cargar1 (10);
54 $resta->cargar2 (5);
55 $resta->operar ();
56 $resta->imprimirResultado ();
57 ?>
58 </body>
59 </html>
```

Figura 73b. Manejando polimorfismo

Universidad Autónoma de Manizales

De programación orientada a objetos con php, se puede escribir cantidades: por ejemplo de colaboración entre objetos, modificadores de acceso, clases abstractas y concretas, métodos destructores y estáticos, etc. El objetivo de esta sesión es fundamentar los conceptos básicos de POO usando el lenguaje php. Depende del estudiante el seguir ahondando en esta temática.



Practicando el desarrollo WEB

Busque en el AULA VIRTUAL el apartado denominado: “EJEMPLOS DE LA SESIÓN 7”; o en el CD adjunto al material escrito, los ejemplos de la SESIÓN_7. Abra cada uno de los ejemplos, usando el ambiente adecuado. Digite o copie el programa y observe el resultado. Interiorice lo visto hasta el momento, mediante el análisis de cada uno de los ejemplos propuestos.



Consultemos

Busque en la red o alguno de los textos propuestos en la bibliografía: investigue el contexto histórico que hace que el lenguaje de programación php, pueda hacerse usando el paradigma de programación orientado a objetos. Organice su respuesta y escríbala en un archivo de texto; una vez tenga el archivo grabado en su computador, ingrese al Aula Virtual y busque el link marcado como CONSULTA No 7. Ingrese y envíelo como un adjunto.



Actividad

Ingrese a AULA VIRTUAL y desarrolle la actividad marcada como: ACTIVIDAD No 7. Esta le ayudará a reforzar lo planteado en este módulo.



Aplicando lo aprendido

Ingrese al Aula Virtual y ubique en la Sesión el apartado denominado: “LABORATORIO DE SESIÓN”, despliegue el archivo LABORATORIO No 7, realice el algoritmo pedido luego de estar seguro que cumple las pruebas de escritorio, envíelo al docente en el link denominado: Entrega LABORATORIO No 7.



En el Aula Virtual

encontrará documentación adicional que le servirá para ampliar los conocimientos, frente a las temáticas tratadas en esta sesión. Revíselos y consulte a sus tutores si surge alguna inquietud.

SESION NÚMERO SIETE PROGRAMACIÓN ORIENTADA A OBJETOS CON PHP

7.3. CONCLUSIONES

- A partir de la versión cinco (5), mediante el lenguaje de programación php, se puede hacer uso del paradigma de programación orientada a objetos; en esencia son las mismas definiciones de los cursos de POO.
- **Objeto.** Se refiere comúnmente a atributos (elementos, variables) y comportamientos (métodos, operaciones) encapsulados en una entidad. Clase. Es un tipo de dato definido por el programador que determina las estructuras de datos que lo forman y las funciones asociadas con él, es decir es un modelo con el que se construyen los objetos.
- **Herencia.** Es la capacidad formar nuevas Clases usando clases previamente definidas. Encapsulamiento. Se refiere a “esconder” el estado de una propiedad o método de un objeto. Una subclase en PHP puede redefinir un método, es decir que podemos crear un método con el mismo nombre que el método de la clase padre. Ahora cuando creamos un objeto de la subclase, el método que se llamará es el de dicha subclase, a esto lo llamamos polimorfismo.
- Cuando creamos un objeto de una clase determinada, los atributos declarados por la clase son localizadas en memoria y pueden ser modificados mediante los métodos.
- Las características de los constructores son: El constructor se ejecuta inmediatamente luego de crear un objeto y no puede ser llamado nuevamente. Un constructor no puede retornar dato. Un constructor puede recibir parámetros que se utilizan normalmente para inicializar atributos. El constructor es un método opcional, de todos modos es muy común definirlo.

RELACIÓN ENTRE EL LENGUAJE PHP Y LAS BASES DE DATOS

8.1. OBJETIVOS ESPECÍFICOS

- Reconocer el manejo de la base de datos Mysql mediante el phpMyAdmin de Appserv.
- Describir la conexión que se puede establecer entre Mysql y el lenguaje de programación php.
- Caracterizar la inserción, borrado, búsqueda de registros de una base de datos en Mysql mediante el lenguaje de programación php.

8.2. DESARROLLO TEMÁTICO

8.2.1. CREANDO LA BASE DE DATOS

Para la realización de este curso sobre PHP con acceso a base de datos hemos elegido la base de datos MySQL por ser gratuita, y por ser también la más empleada en entornos UNIX, para lo cual, el servidor donde tenemos alojadas las páginas nos tiene que proporcionar herramientas para crearla o acceso al Telnet para que la creamos nosotros mismos.

Para crear la base de datos, se usará el administrador de Bases de datos, que se creó al momento de instalar el Appserv. (**Ver anexo de instalación**). Para su ingreso siga los siguientes pasos:

1. Ingresamos al **localhost**. Esto se logra, abriendo una sesión en cualquier navegador y digitando la dirección: localhost (ver figura 74)

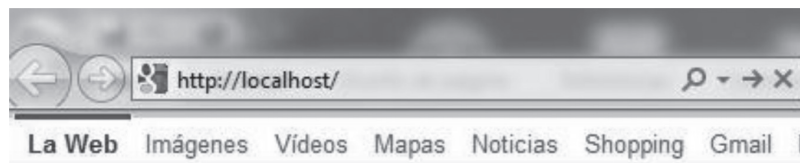


Figura 74. Ingresando al localhost

2. Ahora debe buscar el administrador de Bases de datos y dar click sobre el:

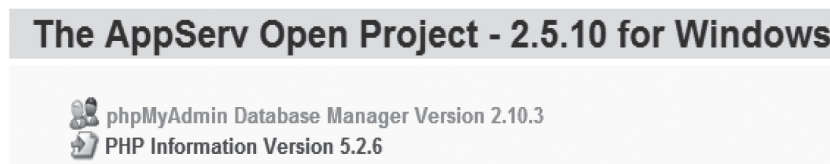


Figura 75. Buscando el phpMyAdmin Database



SESION NÚMERO OCHO RELACIÓN ENTRE EL LENGUAJE PHP Y LAS BASES DE DATOS

- 3. Se pedirá el nombre de usuario y contraseña: casi siempre y como se indicó en el anexo de instalación el usuario es: **root** y la contraseña es: **admin**; depende de los datos que se asignaron.

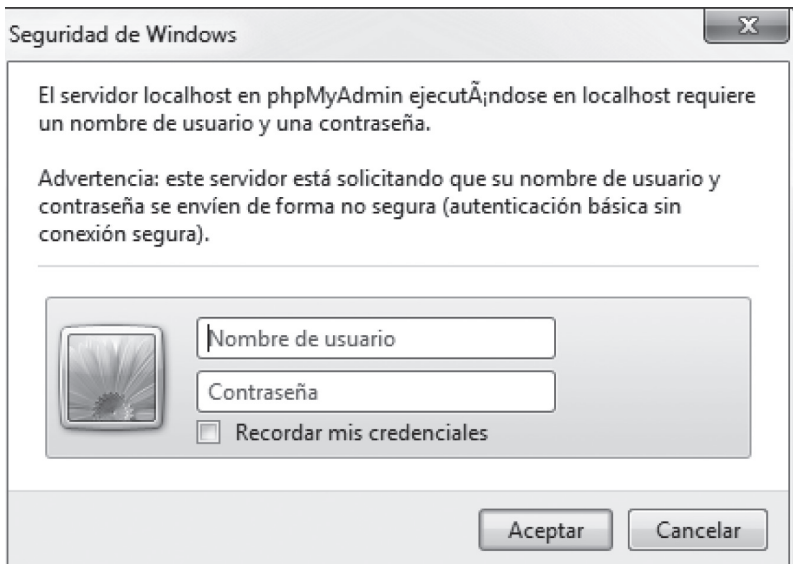


Figura 76. Datos de ingreso al localhost

- 4. Se creará una nueva base de datos. El nombre depende de Usted. En este ejemplo la llamaremos UAM.

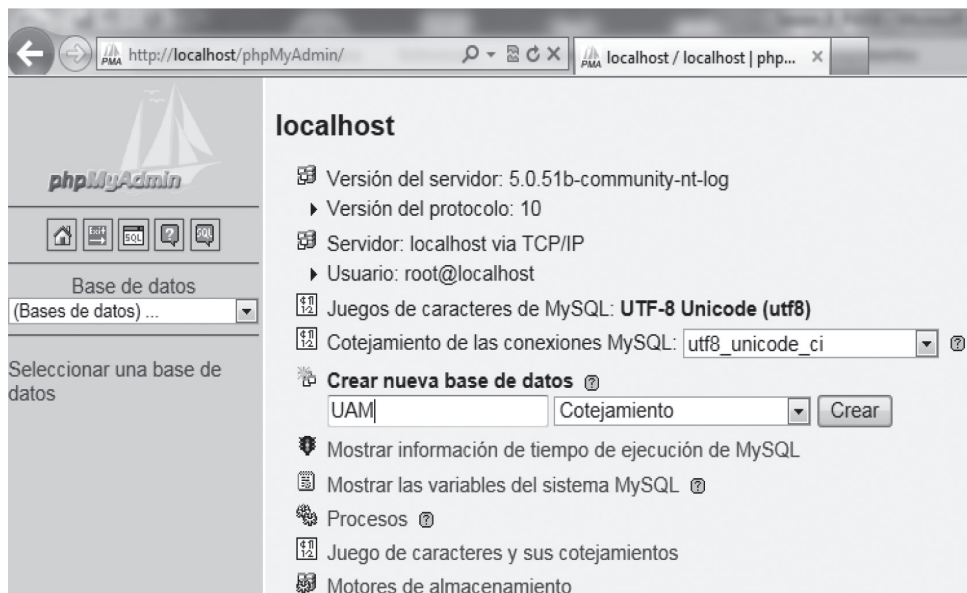


Figura 77. Creando la base de datos

Universidad Autónoma de Manizales

5. Se confirma que se creó.



Figura 78. Confirmación de la creación de la base de datos

6. Una vez conseguido esto debemos crear las tablas en la base de datos, la descripción de las tablas contienen la estructura de la información que almacenaremos en ellas. Para lo cual, usaremos en lenguaje de consultas SQL común para todas las bases de datos relacionales. En este ejemplo creamos una tabla llamada prueba con 3 campos: un campo identificador, que nos servirá para identificar unívocamente una fila con el valor de dicho campo, otro campo con el nombre de una persona y por último un campo con el apellido de la persona. En fundamentos de Bases de datos o en la asignatura Base de datos, se aprendió a crear la tabla usando la siguiente sintaxis o query SQL:

```
CREATE TABLE prueba (  
  Id int(11) DEFAULT '0' NOT NULL auto_increment,  
  Nombre varchar(100),  
  Apellidos varchar(100),  
  PRIMARY KEY (Id),  
  UNIQUE Id (Id) );
```

En nuestro caso y por facilidad usaremos el administrador como se muestra en las figuras siguientes: (Ver figura 79).

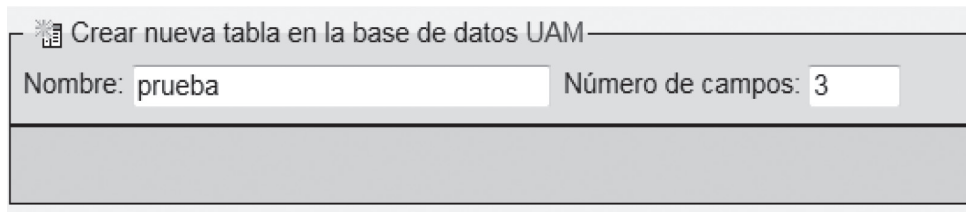


Figura 79. Creando tablas en la BD

SESION NÚMERO OCHO RELACIÓN ENTRE EL LENGUAJE PHP Y LAS BASES DE DATOS

Al darle continuar, diligenciamos los datos conforme lo indicamos en el párrafo anterior.

Campo	Tipo	Longitud/Valores ¹	Cotejamiento	Atributos	Nulo	Predeterminado ²	Extra
Id	INT	11			not null	0	auto_increment
Nombre	VARCHAR	100			not null		
Apellidos	VARCHAR	100			not null		

Figura 80. Diligenciando los campos

- 7. Al darle Grabar se obtiene el mensaje de confirmación de creación de la tabla.

```

consulta SQL:
CREATE TABLE `prueba` (
  `id` INT( 11 ) NOT NULL AUTO_INCREMENT PRIMARY KEY ,
  `Nombre` VARCHAR( 100 ) NOT NULL ,
  `Apellidos` VARCHAR( 100 ) NOT NULL
) ENGINE = MYISAM ;

```

Figura 81. Confirmación de creación de tabla

8.2.2. Conexión a la base de datos desde PHP

Una vez que tenemos creada la base de datos en nuestro servidor, el siguiente paso es conectarnos a la misma desde una página PHP. Para ello, PHP nos proporciona una serie de instrucciones para acceder a bases de datos MySQL. La figura 82, muestra el código que logra la conexión:

```

01 <!-- Manual de PHP de WebEstilo.com -->
02 <html>
03 <head>
04   <title>Ejemplo de PHP</title>
05 </head>
06 <body>
07 <?php
08 function Conectarse()
09 {
10   if (!($link=mysql_connect("localhost","root","admin")))
11   {
12     echo "Error conectando a la base de datos.";
13     exit();
14   }
15   if (!mysql_select_db("UAM",$link))
16   {
17     echo "Error seleccionando la base de datos.";
18     exit();
19   }
20   return $link;
21 }
22
23 $link=Conectarse();
24 echo "Conexión con la base de datos conseguida.<br>";
25
26 mysql_close($link); //cierra la conexion
27 ?>
28 </body>
29 </html>

```

Figura 82. Código php de conexión a la base de datos

8.2.3 CONSULTA A LA BASE DE DATOS

Una vez que nos hemos conectado con el servidor de bases de datos, ya podemos realizar consultas a las tablas de la base de datos.

Para facilitar la programación hemos separado la función de conexión en una librería aparte, de tal manera que la incluiremos en todas las páginas que accedan a la base de datos. En nuestro caso el programa será: S8_ejemplo1.php; dado que fue el que se codificó para la conexión a la base de datos. En la figura 83, se puede visualizar el código de consulta.

```

01 <html>
02 <head>
03   <title>Ejemplo de PHP</title>
04 </head>
05 <body>
06 <H1>Ejemplo de uso de bases de datos con PHP y MySQL</H1>
07 <?php
08   include("S8_ejemplo1.php");
09   $link=Conectarse();
10   $result=mysql_query("select * from prueba",$link);
11   ?>
12   <TABLE BORDER=1 CELSPACING=1 CELLPADDING=1>
13     <TR><TD>&nbsp;Nombre</TD><TD>&nbsp;Apellidos&nbsp;</TD></TR>
14   <?php
15
16   while($row = mysql_fetch_array($result)) {
17     printf("<tr><td>&nbsp;";
18     printf("%s</td><td>&nbsp;";
19     printf("%s&nbsp;</td></tr>", $row["Nombre"]
20     , $row["Apellidos"]);
21   }
22   mysql_free_result($result);
23   mysql_close($link);
24   ?>
25 </table>
26 </body>
27 </html>

```

Figura 83. Código de consulta

Al ejecutar el programa... No se puede olvidar que los códigos php se colocan en la carpeta www del Appserv; dado que es la carpeta pública del servidor local. Se obtiene algo similar a lo mostrado en la figura 84.

Ejemplo de uso de bases de datos con PHP y MySQL

Conexión con la base de datos conseguida.

Nombre	Apellidos
--------	-----------

Figura 84. Conexión establecida para consulta



SESION NÚMERO OCHO RELACIÓN ENTRE EL LENGUAJE PHP Y LAS BASES DE DATOS

En este ejemplo hemos utilizado tres (3) instrucciones nuevas: `mysql_query`, `mysql_fetch_array` y `mysql_free_result`. Con la instrucción `mysql_query` hemos hecho una consulta a la base de datos en el lenguaje de consultas SQL, con la instrucción `mysql_fetch_array` extraemos los datos de la consulta a un array y con `mysql_free_result` liberamos la memoria usada en la consulta.

8.2.4. Insertando registros

Hasta ahora nos hemos conectado a una base de datos y hemos hecho consultas a la misma, ahora presentaremos como introducir nuevo registros en la base de datos.

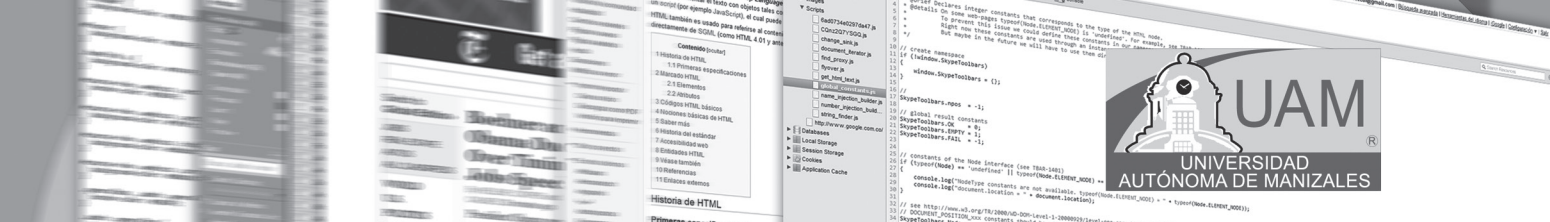
Para ello, usaremos un formulario y en el ACTION del FORM `<FORM ACTION="S8_ejemplo4.php">` indicaremos que debe ser procesado una página PHP, esta página lo que hará será introducir los datos del formulario en la base de datos. Las figuras 85a y 85b muestran el código que permite mostrar el formulario de ingreso de datos.

```
01 <!-- Manual de PHP de WebEstilo.com -->
02 <html>
03 <head>
04   <title>Ejemplo de PHP</title>
05 </head>
06 <body>
07 <H1>Ejemplo de uso de bases de datos con PHP y MySQL</H1>
08 <FORM ACTION="S8_ejemplo4.php">
09 <TABLE>
10 <TR>
11   <TD>Nombre:</TD>
12   <TD><INPUT TYPE="text" NAME="nombre" SIZE="20" MAXLENGTH="30"></TD>
13 </TR>
14 <TR>
15   <TD>Apellidos:</TD>
16   <TD><INPUT TYPE="text" NAME="apellidos" SIZE="20" MAXLENGTH="30"></TD>
17 </TR>
18 </TABLE>
19 <INPUT TYPE="submit" NAME="accion" VALUE="Grabar">
20 </FORM>
21 <hr>
```

Figura 85a Formulario para insertar

```
22 <?php
23   include("S8_ejemplo1.php");
24   $link=Conectarse();
25   $result=mysql_query("select * from prueba",$link);
26 >
27 <TABLE BORDER=1 CELLSPACING=1 CELLPADDING=1>
28 <TR><TD>&nbsp;</TD><TD><B>Nombre</B></TD> <TD>&nbsp;</TD><TD><B>Apellidos</B>&nbsp;</TD></TR>
29 <?php
30
31   while($row = mysql_fetch_array($result)) {
32     printf("<tr><td>&nbsp;</td><td>&nbsp;</td><td>&nbsp;</td><td>&nbsp;</td></tr>",
33       $row["Nombre"], $row["Apellidos"]);
34   }
35   mysql_free_result($result);
36   mysql_close($link);
37 >
38 </table>
39 </body>
40 </html>
```

Figura 85b Continuación del formulario para insertar



Universidad Autónoma de Manizales

```

01 <?php
02 include("S8_ejemplo1.php");
03 $link=Conectarse();
04 mysql_query("insert into prueba (Nombre,Apellidos) values ('$nombre',
05 |'$apellidos')",$link);
06 mysql_close($link);
07
08 ?>

```

Figura 86. Ingresando datos

La primera página PHP S8_ejemplo3.php es un formulario que nos permite introducir nombre y apellido para añadirlo a la base de datos, seguido de una consulta que nos muestra el contenido de la tabla prueba. El formulario llama a la página S8_ejemplo4.php, que añadirá los datos a la tabla.

La segunda página S8_ejemplo4.php se conecta a la base de datos y añade un nuevo registro con la instrucción insert del lenguaje de base de datos SQL.

8.2.5. Borrando registros

El borrado de registros es uno de los procesos más sencillos. Para indicar que elemento vamos a borrar hemos usado un enlace a la página S8_ejemplo6.php pasándole el Id de cada registro; es decir, la llave primaria; de esta manera, la página S8_ejemplo6.php, que borra los registros, sabe que elemento de la tabla ha de borrar. Las figuras 87 y 88 muestran el código que permite verificar este proceso.

```

01 <html>
02 <head>
03   <title>Ejemplo de PHP</title>
04 </head>
05 <body>
06 <h1>Ejemplo de uso de bases de datos con PHP y MySQL</h1>
07 <?php
08   include("S8_ejemplo1.php");
09   $link=Conectarse();
10   $result=mysql_query("select * from prueba",$link);
11   ??
12   <TABLE BORDER=1 CELLSPACING=1 CELLSPACING=1>
13     <TR><TD>&nbsp;Nombre</TD><TD>&nbsp;Apellidos&nbsp;</TD></TR>
14   <?php
15
16   while($row = mysql_fetch_array($result)) {
17     printf("<tr><td>&nbsp;%s</td><td>&nbsp;%s&nbsp;</td></tr>", $row["Nombre"]
18     , $row["Apellidos"]);
19   }
20   mysql_free_result($result);
21   mysql_close($link);
22   ??
23 </table>
24 </body>
25 </html>

```

Figura 87. Formulario de borrado de registros



SESION NÚMERO OCHO RELACIÓN ENTRE EL LENGUAJE PHP Y LAS BASES DE DATOS

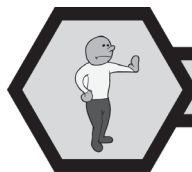
```

01 <?php
02 include("S8_ejemplo1.php");
03 $link=Conectarse();
04 mysql_query("delete from prueba where Id = $id",$link);
05 mysql_close($link);
06

```

Figura 88. Código de borrado de registros

La página S8_ejemplo6.php se conecta a la base de datos y borra el registro indicado en la variable \$id que ha sido pasado desde la página S8_ejemplo5.php. Una vez el registro se ha borrado, lanza un mensaje.



Pare

VISUALIZACION Y ADMINISTRACION

. Las páginas de administración tienen como finalidad administrar el contenido de las páginas de visualización y están protegidas por una autenticación, generalmente van opciones como listar, modificar, eliminar, soportadas por la autenticación.

8.2.6 Autenticación: generalmente encontramos dos tipos de autenticación. Con variables de ambiente como se realizaba con la versión de php3 y a partir de la versión 4 de php incursionaron el manejo con sesiones. Observemos un par de ejemplos:

1. **Variables de ambiente:** las variables que se usan son: \$PHP_AUTH_USER y \$PHP_AUTH_PW. Estas variables se cargan en el ambiente mientras el navegador esté abierto y permiten manejar páginas protegidas por contraseñas.
2. **Autenticación:** determina si la variable php_aut_user está cargada en el ambiente. Una vez se carga permanece activa hasta que el navegador se cierre



Practicando el desarrollo WEB

Busque en el AULA VIRTUAL el apartado denominado: "EJEMPLOS DE LA SESION 8"; o en el CD adjunto al material escrito, los ejemplos de la SESIÓN_8. Abra cada uno de los ejemplos, usando el ambiente adecuado. Digite o copie el programa y observe el resultado. Interiorice lo visto hasta el momento, mediante el análisis de cada uno de los ejemplos propuestos.



Consultemos

Busque en la red o alguno de los textos propuestos en la bibliografía: ¿Qué son métodos: GET y POST en el manejo de formularios usados en la web? Organice su respuesta y escríbala en un archivo de texto; una vez tenga el archivo grabado en su computador, ingrese al Aula Virtual y busque el link marcado como CONSULTA No 8. Ingrese y envíelo como un adjunto.

Universidad Autónoma de Manizales



Actividad

Ingrese a AULA VIRTUAL y desarrolle la actividad marcada como: **ACTIVIDAD No 8**. Esta le ayudará a reforzar lo planteado en este módulo.



Aplicando lo aprendido

Ingrese al Aula Virtual y ubique en la Sesión el apartado denominado: **“LABORATORIO DE SESIÓN”**, despliegue el archivo **LABORATORIO No 8**, realice el algoritmo pedido luego de estar seguro que cumple las pruebas de escritorio, envíelo al docente en el link denominado: **Entrega LABORATORIO No 8**.



En el Aula Virtual

EN EL AULA VIRTUAL Encontrará documentación adicional que le servirá para ampliar los conocimientos, frente a las temáticas tratadas en esta sesión. Revíselos y consulte a sus tutores si surge alguna inquietud.

8.3 CONCLUSIONES

- Mediante el administrador de la base de datos de Appserv se puede gestionar de una manera fácil y simple el manejo de las distintas bases de datos usando el motor de base de datos Mysql.
- Mediante código de programación en php, se puede establecer la conexión de la aplicación con la base de datos y a partir de allí se puede gestionar las tablas y registros de las mismas.
- Como en toda base de datos, se puede insertar, borrar y buscar registro usando para esto programación y/o el administrador de la base de datos MyAdmin de Mysql.

```
51 DOCUMENT_POSITION_PRECEDING : 0x00,  
52 DOCUMENT_POSITION_FOLLOWING : 0x01,  
53 DOCUMENT_POSITION_CONTAINS : 0x02,  
54 DOCUMENT_POSITION_CONTAINED_BY : 0x03,  
55 DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC : 0x04  
56 }  
57  
58 // redefine setAttribute function to emulate DOM2 modified num  
59 // http://bugzilla.mozilla.org/show_bug.cgi?id=4911  
60 Element.prototype.setAttribute = Element.prototype.setAttribute  
61 Element.prototype.setAttribute = function(name, val)  
62 {  
63     // save the previous value  
64     var prev = this.getAttribute(name);  
65 }  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000
```





9.1. OBJETIVOS ESPECÍFICOS

- Definir el manejo de sesiones en desarrollo web.
- Implementar el manejo de sesiones en formularios de logueo propios en la programación web.

9.2. DESARROLLO TEMÁTICO

9.2.1. SESIONES

Una sesión al tiempo en el que un usuario determinado se encuentra navegando en un sitio web; si un usuario no navega por el sitio web durante un determinado tiempo, se dice que ha terminado su sesión en el sitio y a partir de allí cuando vuelve a ingresar lo hace en una nueva sesión. El concepto de sesión es útil porque es posible asociar a cada sesión un identificador único de forma tal de registrar la actividad del usuario en el sitio web y mantener persistencia utilizando únicamente este identificador.

Las sesiones nos van a permitir mantener información relacionada con la sesión en todas las páginas que integran nuestro sitio. Algunas de las funciones que vamos a utilizar para el manejo de sesiones:

- **session_start** : crea una nueva sesión si es que no existe o retoma la sesión existente.
- **session_unset**: elimina todas las variables de la sesión.
- **session_destroy**: destruye todos los datos guardados en una sesión.

Tenemos que colocar la función `session_start` en todas nuestras páginas, pero debes colocarla antes de enviar alguna etiqueta HTML, dato o un espacio de lo contrario te devolverá un error. Ver figura 89.

```

1  <?php
2      session_start();
3  ?>
4  <HTML>
5  <BODY>
6  <?php
7      echo "La variable de sesión es:" . SID;
8  ?>
9  </BODY>
10 </HTML>
    
```

Figura 89. Usando `session_start`

Una vez creada la sesión podemos utilizar la variable súper global `$_SESSION` para almacenar las variables que necesitamos dentro de la sesión. La figura 90 muestra la sintaxis.



SESION NÚMERO NUEVE MANEJO DE SESIONES CON PHP

```

1  <?php
2      session_start ( ) ;
3  ?>
4  <HTML>
5  <BODY>
6  <?php
7      $_SESSION["usuario"] = "Andrea";
8      $_SESSION["idusuario"] = 543;
9  ?>
10 </BODY>
11 </HTML>

```

Figura 90. Usando \$_SESSION

Estas variables van a estar disponibles en todas las páginas en que tengamos la sesión. Ahora bien para eliminar todas las variables de sesión registradas utilizaremos la función **“session_unset”**, la sintaxis es la siguiente: **session_unset()**; Ahora bien, si deseamos solo eliminar una variable lo haremos utilizando la función **unset**. La sintaxis sería la siguiente: **unset(\$_SESSION[“nombrevariable”])**; para eliminar la sesión utilizaremos la función **session_destroy()**;

Ejemplos tomados del sitio web especializado en desarrollo web, denominado: Code Drinks: <http://www.codedrinks.com/index.php/2010/05/27/manejo-de-sesiones-en-php/> [Visitado el 20 de junio de 2011]

9.2.2 APLICACIÓN DE LAS SESIONES EN FORMULARIOS DE LOGUEO.

Creando la tabla, dentro de la base de datos uam, creada con anterioridad.

Campo	Tipo	Longitud/Valores ¹	Cotejamiento	Atributos	Nulo	Predeterminado ²	Extra
idusuario	INT	11		UNSIGNED ZEROFILL	not null		auto_increment
Nombre	VARCHAR	20			not null		
nick	VARCHAR	20			not null		
contrasenia	VARCHAR	20			not null		

Comentarios de la tabla:
 Motor de almacenamiento: MyISAM
 Cotejamiento: latin1_bin

Figura 91. Creando la tabla usuario en la base de datos: uam

```

consulta SQL:
CREATE TABLE `usuario` (
  `idusuario` INT( 11 ) UNSIGNED ZEROFILL NOT NULL AUTO_INCREMENT PRIMARY KEY ,
  `Nombre` VARCHAR( 20 ) NOT NULL ,
  `nick` VARCHAR( 20 ) NOT NULL ,
  `contrasenia` VARCHAR( 20 ) NOT NULL
) ENGINE = MYISAM CHARACTER SET latin1 COLLATE latin1_bin;

```

Figura 92. Confirmación de creación de la tabla

Universidad Autónoma de Manizales

Insertamos registros:

```
- Ejecutar la(s) consulta(s) SQL en la base de datos uam: ? -  
INSERT INTO usuario(idusuario, Nombre, nick,contrasenia)  
VALUES (1,'Fernando Martinez', 'fer','123')
```

Figura 93. Insertando registros a la base de datos uam en la tabla usuario

Confirmación del registro insertado en la base de datos.

Filas insertadas: 1 (La consulta tardó 0.0013 seg)

consulta SQL:

```
INSERT INTO usuario( idusuario, Nombre, nick, contrasenia )  
VALUES ( 1, 'Fernando Martinez', 'fer', '123' )
```

Figura 94. Confirmación de inserción

De otra forma se puede insertar también registros en la tabla. Se puede hacer directamente en insertar.

Base de datos: uam ▶ Tabla: usuario

SQL Buscar Insertar Exportar

Figura 95. Insertar directamente

Campo	Tipo	Función	Nulo	Valor
idusuario	int(11) unsigned zerofill			2
Nombre	varchar(20)			Maritza Zambrano
nick	varchar(20)			mary
contrasenia	varchar(20)			456

Figura 96. Insertando directamente

Verificación de inserción usando este método.

Filas insertadas: 1

consulta SQL:

```
INSERT INTO `uam`.`usuario` (  
`idusuario`,  
`Nombre`,  
`nick`,  
`contrasenia`  
)  
VALUES (  
`2`, `Maritza Zambrano`, `mary`, `456`  
);
```

Figura 97. Verificación de inserción de registro

Verificación de registros en la tabla. Esto se hace seleccionando Examinar que se encuentra en la parte superior.

SESION NÚMERO NUEVE MANEJO DE SESIONES CON PHP



Figura 98. Verificación de registros en la tabla usuario de la BD uam

Ahora escribimos los programas en php, que mostrarán el uso de sesiones. Para esto se hará un programa que permite el ingreso de un usuario a un sitio web. Inicialmente se pide el nombre del usuario y la contraseña de ingreso. Ver figuras 99 a 102

```

01 <?php
02 class sesion {
03     function __construct() {
04         session_start ();
05     }
06     public function set($nombre, $valor) {
07         $_SESSION [$nombre] = $valor;
08     }
09     public function get($nombre) {
10         if (isset ( $_SESSION [$nombre] )) {
11             return $_SESSION [$nombre];
12         } else {
13             return false;
14         }
15     }
16     public function elimina_variable($nombre) {
17         unset ( $_SESSION [$nombre] );
18     }
19     public function termina_sesion() {
20         $_SESSION = array();
21         session_destroy ();
22     }
23 }
24 ?>
    
```

Figura 99. Sesión.class.php

Las siguientes tres gráficas muestran el código que se escribe para permitir el ingreso de un usuario a un determinado sitio.

```

01 <?php
02     require_once("sesion.class.php");
03
04     $sesion = new sesion();
05
06     if( isset($_POST["iniciar"]) )
07     {
08
09         $usuario = $_POST["usuario"];
10         $password = $_POST["password"];
11
12         if(validarUsuario($usuario,$password) == true)
13         {
14             $sesion->set("usuario",$usuario);
15
16             header("location: principal.php");
17         }
18         else
19         {
20             echo "Verifica tu nombre de usuario y contraseña";
21         }
22     }
23 //Continúa en la siguiente figura
    
```

Figura 100a. Código inicial de logueo de usuarios. Login.php

```

23 //Continuación
24
25 function validarUsuario($usuario, $password)
26 {
27     $conexion = new mysqli("localhost","root","admin","uam");
28     $consulta = "select contrasenia from usuario where nick = '$usuario'";
29
30     $result = $conexion->query($consulta);
31
32     if($result->num_rows > 0)
33     {
34         $fila = $result->fetch_assoc();
35         if( strcmp($password,$fila["contrasenia"]) == 0 )
36             return true;
37         else
38             return false;
39     }
40     else
41         return false;
42 }
43
44 >>
    
```

Figura 100b. Continuación del código de logueo



SESION NÚMERO NUEVE MANEJO DE SESIONES CON PHP

```

45 <html>
46 <head>
47 <title></title>
48 </head>
49
50 <body>
51 <form name="frmLogin" action="<?php echo $_SERVER['PHP_SELF'];?>" method="POST">
52   <div>
53     <div> <label>Usuario: </label> <input type="text" name = "usuario"/></div>
54     <div><label>Contraseña: </label> <input type="password" name = "password"
55       /></div>
56     <div><input type="submit" name ="iniciar" value="Iniciar Sesion"/></div>
57   </div>
58 </form>
59 </body>
60 </html>

```

Figura 100c. Culminación del código de logueo

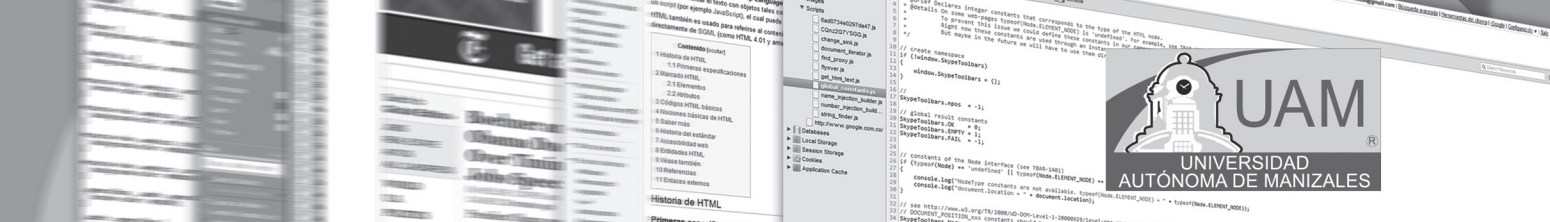
Código de la página de recepción una vez se valida el usuario en el sistema.

```

01 <?php
02   require_once("sesion.class.php");
03
04   $sesion = new sesion();
05   $usuario = $sesion->get("usuario");
06
07   if( $usuario == false )
08   {
09     header("Location: login.php");
10   }
11   else
12   {
13     ?>
14     <HTML><head>
15     <title></title>
16     </head>
17     <body>
18     <h1>Hola: <?php echo $sesion->get("usuario"); ?> </h1>
19     <a href="cerrarsesion.php"> Cerrar Sesion </a>
20     <p> Aqui va el contenido de la pagina </p>
21     </body>
22     </HTML>
23
24     <?php
25     }
26 ?>

```

Figura 101. Página de inicio, luego de validar el ingreso. principal.php



Universidad Autónoma de Manizales

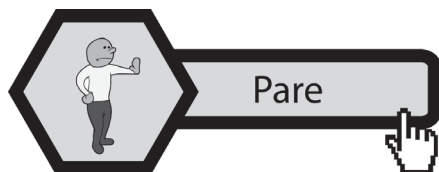
La siguiente figura muestra el cierre de sesión.

```

01 <?php
02     require_once("sesion.class.php");
03
04     $sesion = new sesion();
05     $usuario = $sesion->get("usuario");
06     if( $usuario == false )
07     {
08         header("Location: login.php");
09     }
10     else
11     {
12         $usuario = $sesion->get("usuario");
13         $sesion->termina_sesion();
14         header("location: login.php");
15     }
16 >

```

Figura 102. cerrarsesion.php



Pare

Ejemplos tomados del sitio web especializado en desarrollo web, denominado: Code Drinks: [Visitado el 20 de junio de 2011]

<http://www.codedrinks.com/index.php/2010/06/24/formulario-de-login-con-php-y-mysql/>



Practicando el desarrollo WEB

Busque en el AULA VIRTUAL el apartado denominado: "EJEMPLOS DE LA SESIÓN 9"; o en el CD adjunto al material escrito, los ejemplos de la SESIÓN_9. Abra cada uno de los ejemplos, usando el ambiente adecuado. Digite o copie el programa y observe el resultado. Interiorice lo visto hasta el momento, mediante el análisis de cada uno de los ejemplos propuestos.



Consultemos

Busque en la red o alguno de los textos propuestos en la bibliografía: ¿Cómo se realiza el encriptado de contraseñas con md5? Organice su respuesta y escríbala en un archivo de texto; una vez tenga el archivo grabado en su computador, ingrese al Aula Virtual y busque el link marcado como CONSULTA No 9. Ingrese y envíelo como un adjunto.



Actividad

Ingrese a AULA VIRTUAL y desarrolle la actividad marcada como: ACTIVIDAD No 9. Esta le ayudará a reforzar lo planteado en este módulo.



SESION NÚMERO NUEVE MANEJO DE SESIONES CON PHP



Aplicando lo
aprendido

Ingrese al Aula Virtual y ubique en la Sesión el apartado denominado: “LABORATORIO DE SESIÓN”, despliegue el archivo LABORATORIO No 9, realice el algoritmo pedido luego de estar seguro que cumple las pruebas de escritorio, envíelo al docente en el link denominado: Entrega LABORATORIO No 9.



En el Aula Virtual

encontrará documentación adicional que le servirá para ampliar los conocimientos, frente a las temáticas tratadas en esta sesión. Revíselos y consulte a sus tutores si surge alguna inquietud.

9.3. CONCLUSIONES

- Una sesión al tiempo en el que un usuario determinado se encuentra navegando en un sitio web; si un usuario no navega por el sitio web durante un determinado tiempo, se dice que ha terminado su sesión en el sitio y a partir de allí cuando vuelve a ingresar lo hace en una nueva sesión.
- Una de las aplicaciones típicas del manejo de sesiones es el logueo que generalmente se construye para el ingreso de usuarios a un desarrollo web determinado.

SESION NÚMERO DIEZ PUBLICACIÓN EN LA WEB



Nombre	Fecha de modifica...	Tipo	Tamaño
 imagenes	27/06/2011 02:27 ...	Carpeta de archivos	
 index	27/06/2011 02:37 ...	Documento HTML	2 KB

Figura 103. Organización en el desarrollo web

```

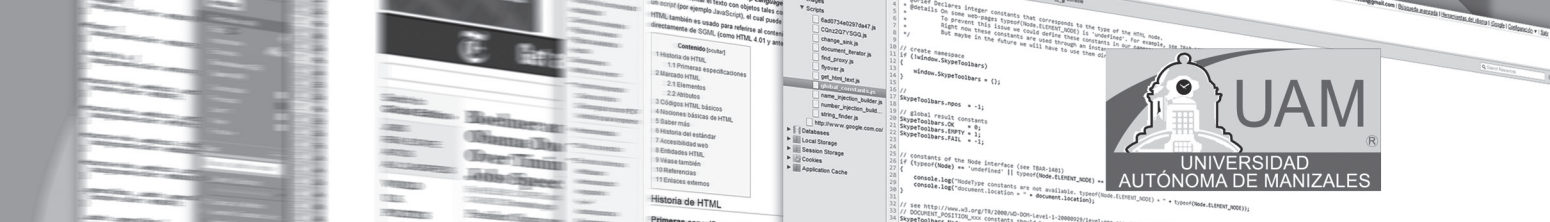
1  <HTML><HEAD><TITLE>Programación Web</TITLE>
2  <body >
3  <BR>
4  <BR>
5  <BR>
6  <P align=center><FONT face=impact color=#ff0000 size=7><STRONG>UNIVERSIDAD
7  AUTÓNOMA DE MANIZALES</STRONG></FONT></P>
8  <BR>
9  <p align="center" ><IMG style="WIDTH: 55px; HEIGHT: 75px" height=127
10 hspace=12 src="imagenes/construccion1.gif" width=97>
11 <P align=center><FONT color=#150567 size=3><STRONG><FONT face=Arial>TECNOLOGIA
12 EN ANALISIS Y PROGRAMACION DE SISTEMAS DE INFORMACION</FONT></STRONG></FONT></P>
13 <BR>
14 <p align="center" ><font color=#348017 face="Arial Black" size="4" >SITIO WEB EN
15 CONSTRUCCION</font>
16 <p align="center" ><FONT face="Arial Black" size=2>Contacte al Profesor
17 para cualquier inquietud:</FONT><BR> <A href="mailto:fmartinez@autonoma.edu.co" >
18 <font size="3" >fmartinez@autonoma.edu.co</font></A></p>
19 <BR>
20 <p align="center" ><IMG style="WIDTH: 70px; HEIGHT: 50px" height=97 hspace=12
21 src="imagenes/construccion3.gif" width=97></FONT>
22 <BR>
23 <p align="center" ><b><i>
24 <!--ASI SE HACEN COMENTARIOS -->
25 <p align="center" ><font face="Arial Black" size="1">@2011 WebMaster: Ferm@th.
26 Bogotá.Colombia</font></p>
27 </body>
28 </HTML>
    
```

Figura 104. Index.html

10.2.2. Hosting para residencia de un desarrollo web

Es importante tener en cuenta que para disponer de una página web visible se hace necesario un lugar donde alojarla. Ese “lugar” (espacio web) lo puede ofrecer un servidor de *hosting* (**que significa alojamiento de páginas web**) bien de forma gratuita o bien pagando cierta cantidad. Más adelante se tocará esta temática de planes gratuitos y de pago donde alojar algún sitio web, e incluso instrucciones detalladas de cómo registrarse en un servidor gratuito.

Por otro lado, una vez tenemos el espacio para nuestra página web, los visitantes necesitan escribir algo en sus navegadores para dirigirse a nuestra nueva web. Esto es la URL o dirección de la web que puede ser o bien un dominio escogido (**y pagado**) por tí, como por ejemplo www.fermath.org o www.google.es o www.mipagina.es o bien un subdominio (**los gratuitos suelen ser subdominios**) que tienen una forma algo más larga, por ejemplo el sitio donde reside el aula virtual de la UAM en EDUPOL: <http://aulavirtualuam.edupol.com.co>. Veremos entonces cómo registrar un dominio y también cómo relacionar un dominio o subdominio con un hosting, etc.



Universidad Autónoma de Manizales

10.2.3. Dominio

Un dominio o nombre de dominio es el nombre que identifica un sitio web. Cada dominio tiene que ser único en la internet. Por ejemplo, “www.masadelante.com” es el nombre de dominio de la página web de Masadelante. Un solo servidor web puede servir múltiples páginas web de múltiples dominios, pero un dominio sólo puede apuntar a un servidor.

Un dominio se compone normalmente de tres partes: en www.masadelante.com, las tres uves dobles (www), el nombre de la organización (masadelante) y el tipo de organización (com).

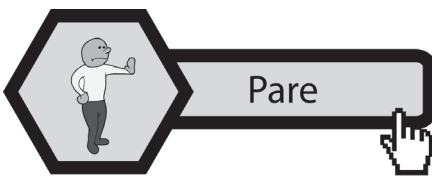
Los tipos de organización más comunes son .COM, .NET, .MIL, y .ORG, que se refieren a comercial, *network*, militar, y organización (originalmente sin ánimo de lucro, aunque ahora cualquier persona puede registrar un dominio .org).

Puesto que la internet se basa en direcciones IP, y no en nombres de dominio, cada servidor web requiere de un servidor de nombres de dominio (DNS) para traducir los nombres de los dominios a direcciones IP. Cada dominio tiene un servidor de nombre de dominio primario y otro secundario.

Tomado de: <http://www.masadelante.com/faqs/dominio> [Visitado el 27 de junio de 2011]

10.2.4. Subdominio

Un subdominio es un dominio dentro de un dominio. Generalmente los dominios inician con www.elsitio.com, mientras que los subdominios inician con: <http://subdominio.elsitio.com>; obsérvese que el subdominio está ligado de un dominio por medio de un punto y este no inicia con www como si lo hace el dominio. Haciendo un poco de claridad al respecto, se debe tener presente que el subdominio normalmente se escribe delante del dominio al que está asociado el subdominio, separado por un “.” por ejemplo, trabajos.profesionalhosting.com es el subdominio trabajo de profesionalhosting.com, su principal ventaja es que un subdominio tiene todas las propiedades de un dominio con la diferencia de que este es gratis y con la única desventaja de que aumenta un poco la dirección.



Pare

Al contratar un servicio de residencia para un sitio Web (hosting) en general los contratos traen consigo un dominio que es obsequiado por la compañía que presta el servicio de Hosting.

Para adquirir un subdominio existen bastantes sitios en la red que ofrecen este servicio, algunos son tan buenos que dan bastante espacio y no colocan publicidad en el espacio cedido; otros hacen totalmente lo opuesto.

10.2.5. Encontrando un sitio de residencia web (hosting)

Como se ha indicado anteriormente, si se desarrolla un sitio web es necesario hacerlo público. Lo ideal es contratar hosting para poder a residir un sitio web desarrollado; esto dará mayor autonomía en la administración y soporte del mismo; por *default* se obtendría un dominio registrado. Para efectos didácticos en el aprendizaje de esta asignatura se tomará como referencia una empresa que preste servicio de residencia gratuita. Se deja claro aquí que existen bastantes sitios que prestan este mismo servicio que aquí se describirá al detalle y donde pondremos a residir el ejemplo considerado anteriormente, denominado: `index.html`



SESION NÚMERO DIEZ PUBLICACIÓN EN LA WEB

10.2.6. Contacto con la empresa prestadora del servicio

Lo primero que se hace es buscar en la red una empresa que presente el servicio de residencia de sitios web. En este caso se debe digitar: <http://byethost.com> El sitio seleccionado ha sido evaluado por varios años, encontrando su servicio como uno de los más serios y completos del mercado; tanto el servicio gratuito como el comercial son bastante robustos y se garantiza la estabilidad del mismo. Por los motivo se ha seleccionado esta empresa. La figura 105 muestra la página de contacto.

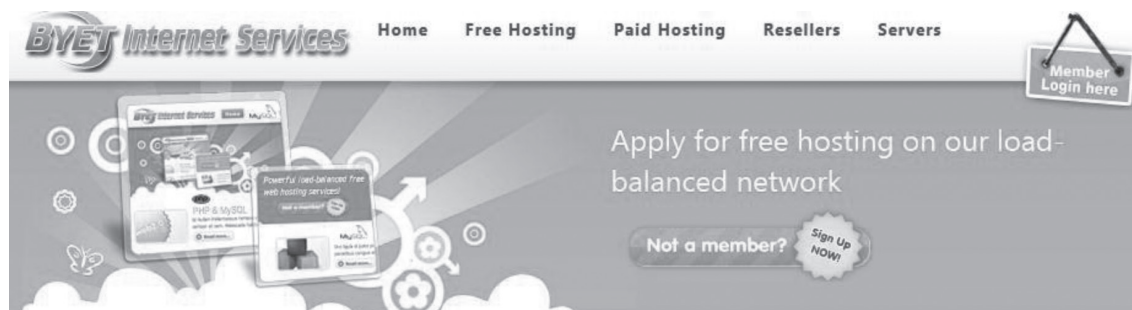


Figura 105. Contactado Byethost.com

De click donde dice **“Free Hosting”** y una vez allí, de nuevamente click en: **“Click here to sign up for free Hosting”** Es importante que se completen los campos del formulario que se presenta. La figura 106, muestra el formulario de solicitud paro obtener servicio de Hosting gratuito. Es fundamental recordar que el servicio le otorgará una dirección URL para su sitio web, a manera de subdominio. Tenga presente el USERNAME, este será el que inicialmente identificará su sitio web.

Signup for free hosting

Fill out the form below using Mozilla Firefox and your free hosting account will be activated instantly!
(Internet Explorer is not recommended)

Please do not translate this page, you will not be able to signup though a translator!

Username

Password

Email Address

Site Category Choose from Below

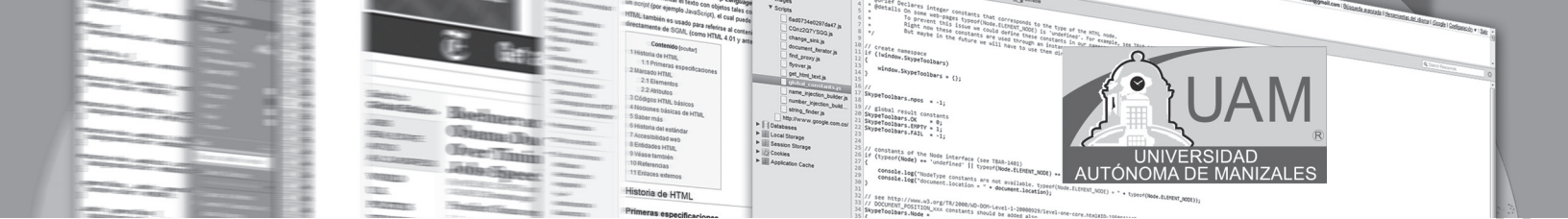
Site Language Choose from Below

Security Code D 4 D 0 7

Enter Security Code

Figura 106. Formulario de solicitud de servicio

Complete totalmente los campos que son solicitados; el sistema le enviará un correo donde confirmará que un ser humano y no una máquina está solicitado el servicio. Las figura 107 y 108 muestran los dos correos que deberá recibir. No olvide seguir los pasos que le indique el sistema.



Universidad Autónoma de Manizales

```
Dear byethost17.com's Member,  
  
Thank you for registering with byethost17.com.  
  
Please click on the link below to activate your account. If the page  
does not display, you may copy and paste the link to your browser.  
  
http://securesignup.net/activate.php?user=b17\_8387173&email=sigmapi@matematicas.net&hash:  
  
best regards  
byethost17.com
```

Figura 107. Correo inicial de solicitud

```
Dear Valued Customer,  
  
Thank you for registering at byethost17.com. We are delighted to inform  
you that your application b17_8387173 has been successful !  
  
Your account has been activated and you should be able to login to the  
Cpanel now.  
  
Take careful note of your login details below and consider printing them  
for your own records.  
  
If you like our free hosting service, please consider upgrading your  
account to one of our Premium Hosting Plans and get a free .COM, NET or  
INFO domain name:  
  
Our premium paid hosting service has very large hosting plans using the  
industry leading cPanel X control panel, without any of the restrictions  
or limitations that are placed on free hosting accounts such as PHP  
sockets, CPU quotas, SMTP & POP email accounts, no file size limits and  
much more, all hosted on ultra fast quad CPU servers.  
  
For more information please visit : https://securesignup.net/portal/  
  
For help on using your hosting and control panel please visit :  
http://tutorials.securesignup.net/
```

Figura 108. Correo final de solicitud

Este último correo es importantísimo en la medida que en le informarán los datos importantes para gestionar todo en la red, referente al sitio que desea publicar. La figura 109, muestra la lista de datos que le enviará el sistema. Se debe guardar estos datos con gran sigilo dado que son los que permitirán su ingreso al su sitio.

```
-----  
Cpanel Username:  
Cpanel Password:  
Your URL:  
  
FTP Server :  
FTP Login :  
FTP Password :  
MySQL Database Name:  
MySQL Username :  
MySQL Password :  
MySQL Server:  
  
Cpanel URL:  
-----
```

Figura 109. Datos para gestionar el sitio



SESION NÚMERO DIEZ PUBLICACIÓN EN LA WEB

10.2.6.1. Entendiendo cada uno de estos datos

1. Cpanel Username: Usuario del panel de control registrado.
2. Cpanel Password: Contraseña de ingreso del usuario al panel de control.
3. Your URL: La dirección del sitio Web. Al ser gratis en un subdominio.
4. FTP Server: Dirección para acceder el sitio usando protocolo FTP.
5. FTP Login: Usuario para acceder el sitio usando protocolo FTP.
6. FTP Password: Contraseña para acceder el sitio usando protocolo FTP.
7. MySQL Database Name: Nombre de la Base de datos. Debe crearse.
8. MySQL Username: Usuario de la base de datos.
9. MySQL Password: Contraseña de la base de datos.
10. MySQL Server: Servidor de la base de datos.
11. CPANEL URL: Dirección para acceder el panel de control del sitio.

10.2.6.2. Ingresando al panel de control

Para gestionar el panel de control, donde se tendrán todos los archivos del sitio al igual que la base de datos, se hace necesario digitar la dirección que dice en CPANEL URL, allí verá una ventana parecida a la mostrada en la figura 110.

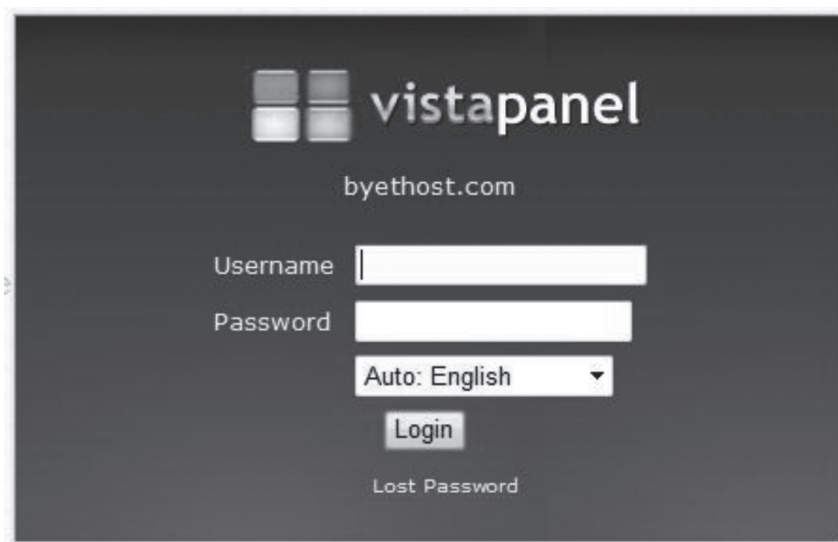


Figura 110. Ingreso al panel de control

Universidad Autónoma de Manizales

Aquí se digitan los datos que pide el sistema para ingresar al panel de control. Una vez dentro verá una página similar a la que se muestra en la figura 111.

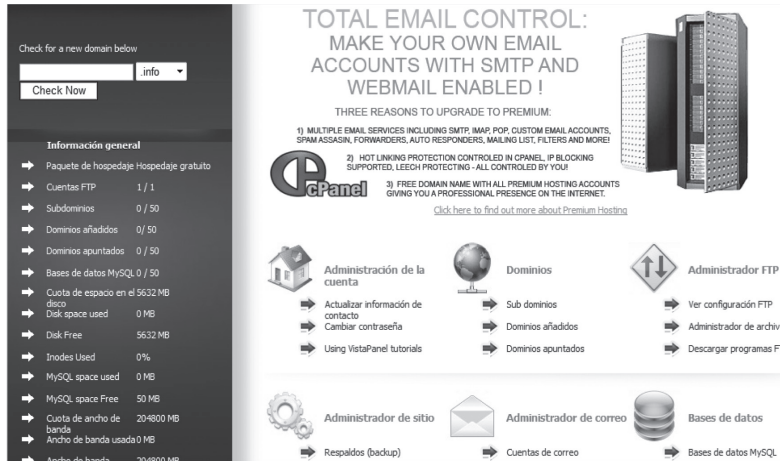
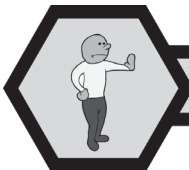


Figura 111. El Cpanel o panel de control



Pare

Haga un recorrido por el **PANEL DE CONTROL** o **CPANEL**, y reconozca las distintas opciones que puede tener, para la gestión de su sitio web.

10.2.7. Subiendo archivos al sitio web

Para subir archivos al sitio web, se pueden realizar de dos formas:

1. **Usando un servicio ftp:** este se usará si el sitio web tiene bastantes carpetas de gran tamaño.
2. **Usando el Administrador de archivos del Cpanel:** este es el indicado si se trata de pocos archivos que se desean subir al tiempo.

Para efectos de este módulo, se hará el ejemplo usando la forma 2. Se hace el ejemplo subiendo el index.html creado desde el inicio de esta sesión. Para lograrlo se siguen los siguientes pasos:

1. Ubique administrador de archivos en la sección Administrador del sitio. Ver figura 112.



Figura 112. Administrador de Archivos

SESION NÚMERO DIEZ PUBLICACIÓN EN LA WEB

- Al ingresar a esta parte se encontrará una pantalla muy parecida a la que se muestra en la figura 113.

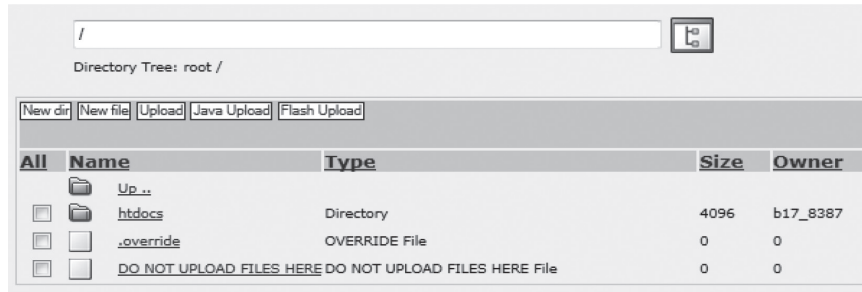


Figura 113. Ubicación de archivos

- Los documentos o archivos que se desean subir se deben colocar en la carpeta pública: www; aquí es llamada htdocs. Al desplegar esta parte, se encuentra la existencia de un index.html, el cual ha sido colocado por default en el sistema, este archivo se debe borrar (se señala y se borra seleccionando delete y confirmado en borrado). Se subirán los archivos creados para este ejemplo, los cuales son: el nuevo index.html y una carpeta adicional que contiene las gráficas que necesita dicho archivo. Por facilidad se crea un .zip; este se sube y se descomprime en la misma carpeta htdocs. En las figuras 114 y 115 se observa esto.



Figura 114. Subiendo un archivo .zip

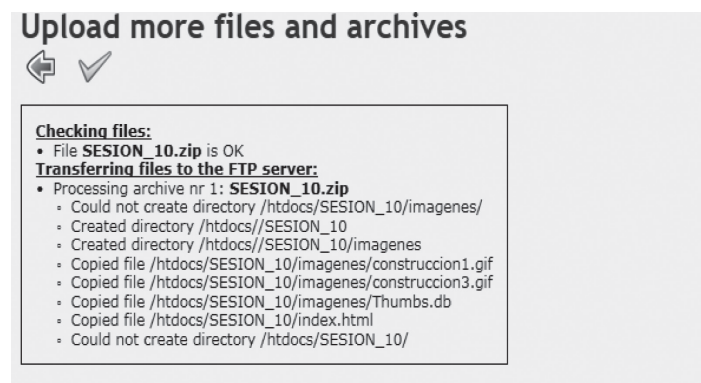
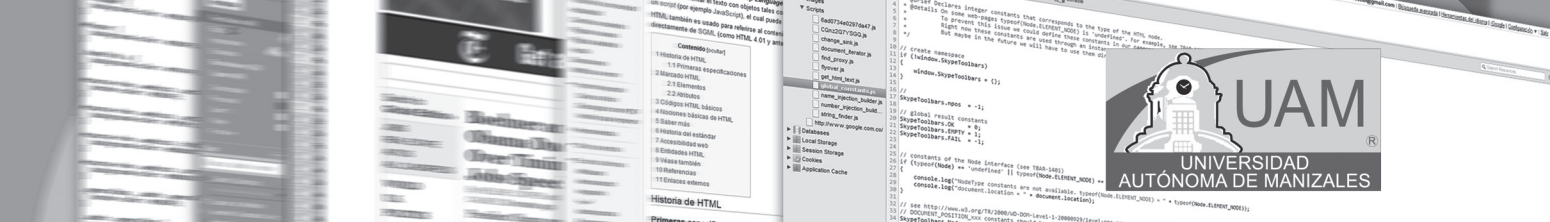


Figura 115. Confirmación de transferencia de archivos



Universidad Autónoma de Manizales

Es importante que se tenga presente que los archivos se descomprimen de inmediato, al subir por este medio al Hosting.

- Se organizan los archivos conforme se diseñó el sitio. Todo lo anterior se debe hacer en la carpeta htdocs. No olvide que el archivo visible por importancia debe ser: index.html (puede ser también un index.php, index.jsp, etc). Ver la figura 116.

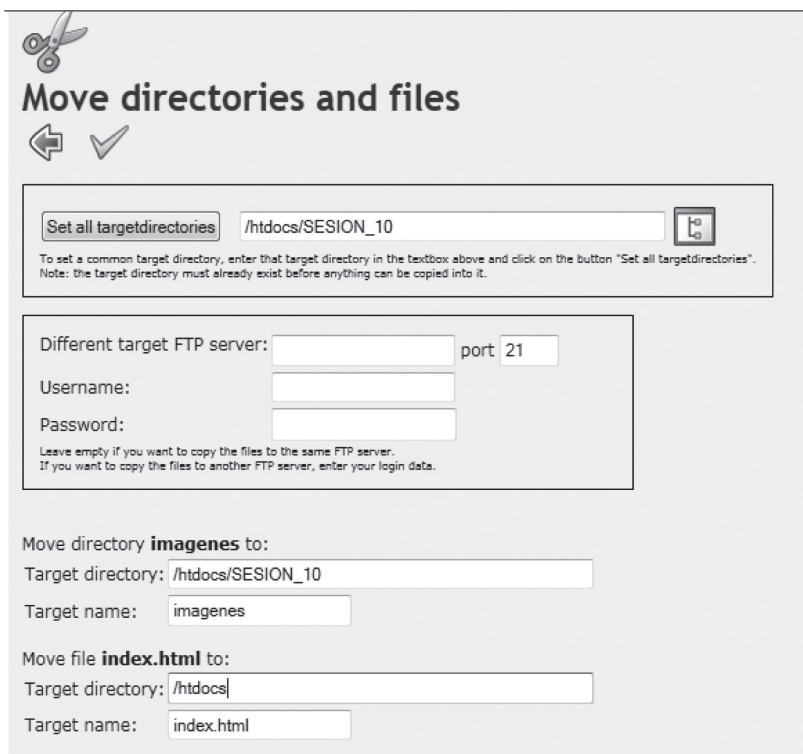


Figura 116. Moviendo archivos

- Los archivos deben quedar en el sitio indicado, como se muestra en la figura 117.

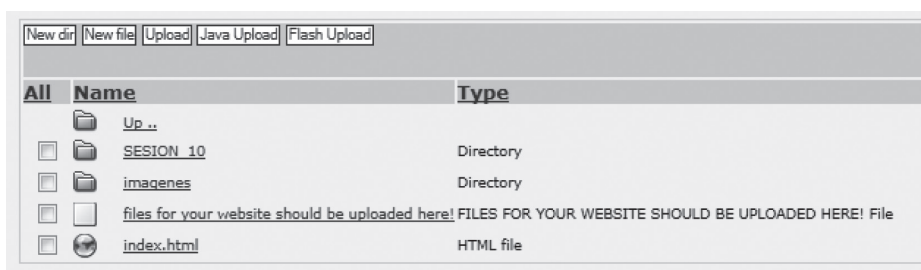


Figura 117. Archivos ubicados en Htdocs, carpeta pública

- Una vez realizado esto, ya puede digitar la dirección URL de su sitio web; deberá funcionar óptimamente. En este caso se debe mostrar una página similar a la siguiente, si tomo como base este ejemplo. Ver figura 118.



SESION NÚMERO DIEZ PUBLICACIÓN EN LA WEB UNIVERSIDAD AUTÓNOMA DE MANIZALES



TECNOLOGIA EN ANALISIS Y PROGRAMACION DE SISTEMAS DE INFORMACION

SITIO WEB EN CONSTRUCCION

Contacte al Profesor para cualquier inquietud:
fmartinez@autonoma.edu.co



@2011 WebMaster: Fern@th. Bogotá, Colombia

Figura 118. Salida por pantalla en la Web del index.html de ejemplo

7. No olvide cerrar su Cpanel, por seguridad. Esto se logra en cerrar sesión.



Practicando el desarrollo WEB

Busque en el AULA VIRTUAL el apartado denominado: "EJEMPLOS DE LA SESION 10"; o en el CD adjunto al material escrito, los ejemplos de la SESIÓN_10. Abra cada uno de los ejemplos, usando el ambiente adecuado. Digite o copie el programa y observe el resultado. Interiorice lo visto hasta el momento, mediante el análisis de cada uno de los ejemplos propuestos.



Consultemos

Busque en la red o alguno de los textos propuestos en la bibliografía: ¿cuáles son los protocolos de la internet? ¿cómo se establece si un dominio está asignado u ocupado? Organice su respuesta y escríbala en un archivo de texto; una vez tenga el archivo grabado en su computador, ingrese al Aula Virtual y busque el link marcado como CONSULTA No 10. Ingrese y envíelo como un adjunto.



Actividad

Ingrese a AULA VIRTUAL y desarrolle la actividad marcada como: ACTIVIDAD No 10. Esta le ayudará a reforzar lo planteado en este módulo.



Aplicando lo aprendido

Ingrese al Aula Virtual y ubique en la Sesión el apartado denominado: “LABORATORIO DE SESIÓN”, despliegue el archivo LABORATORIO No 10, realice el algoritmo pedido luego de estar seguro que cumple las pruebas de escritorío, envíelo al docente en el link denominado: Entrega LABORATORIO No 10.



En el Aula Virtual

EN EL AULA VIRTUAL Encontrará documentación adicional que le servirá para ampliar los conocimientos, frente a las temáticas tratadas en esta sesión. Revíselos y consulte a sus tutores si surge alguna inquietud.

10.3. CONCLUSIONES

- Cuando se hace algún desarrollo que será publicado en la web, tal como un sitio o portal web, se hace necesario tener como base una página que será la encargada de cargar las demás páginas involucradas en el desarrollo realizado
- Es importante tener en cuenta que para disponer de una página web visible se hace necesario un lugar donde alojarla. Ese “lugar” (espacio web) lo puede ofrecer un servidor de Hosting (que significa alojamiento de páginas web) bien de forma gratuita o bien pagando cierta cantidad.
- Un dominio o nombre de dominio es el nombre que identifica un sitio web. Cada dominio tiene que ser único en Internet. Un subdominio es un dominio dentro de un dominio.
- Cada panel de control de un Hosting determinado, tiene sus opciones para facilitar la gestión de archivos y bases de datos entre otras opciones.
- El archivo que gestiona los otros archivos siempre se denomina index, este puede tener extensión: html, php, asp, jsp, etc.

```
51 DOCUMENT_POSITION_PRECEDING : 0x00,  
52 DOCUMENT_POSITION_FOLLOWING : 0x01,  
53 DOCUMENT_POSITION_CONTAINS : 0x02,  
54 DOCUMENT_POSITION_CONTAINED_BY : 0x03,  
55 DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC : 0x04  
56 }  
57  
58 // redefine setAttribute function to emulate DOM2 modified num  
59 // http://bugzilla.mozilla.org/show_bug.cgi?id=4911  
60 Element.prototype.setAttribute = Element.prototype.setAttribute  
61 Element.prototype.setAttribute = function(name, val)  
62 {  
63     // save the previous value  
64     var prev = this.getAttribute(name);  
65 }  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000
```



11. ANEXO DE INSTALACIONES

11.1. OBJETIVOS ESPECÍFICOS

- Aprender a instalar Notepad++.
- Aprender a desplegar HapEdit.
- Aprender a instalar Appserv.
- Aprender a instalar Filezilla.

11.2. DESARROLLO TEMÁTICO

11.2.1. Programas incluidos en el cd

La figura 119, muestra el contenido que se encuentra en CD, que se envía como complemento al material escrito, enviado a cada uno de los centros asociados.

nombre	fecha de modifca...	tipo
APPSERV	27/06/2011 07:21 ...	Carpeta de archivos
FTP	27/06/2011 07:24 ...	Carpeta de archivos
HPPEDIT	27/06/2011 07:14 ...	Carpeta de archivos
NOTPAD++	27/06/2011 07:16 ...	Carpeta de archivos
PARA_COMPRIMIR	27/06/2011 07:37 ...	Carpeta de archivos

Figura 119. Contenido del CD, adicional a material escrito

Al interior del material escrito se hace uso de estos programas para comprender los distintos conceptos. El manejo de cada uno de estos paquetes de desarrollo se socializará a medida que se realicen las diversas tutorías sobre cada una de las temáticas.

11.2.2. Instalando notepad++

Notepad++ es un programa muy versátil, que permite trabajar fácilmente en la escritura de código para programación. Notepad++ es un editor de texto y de código fuente libre con soporte para varios lenguajes de programación. Una de sus limitaciones es que solo funciona en Microsoft Windows. Este excelente programa se parece al Bloc de notas en cuanto al hecho de que puede editar texto sin formato y de forma simple. No obstante, incluye opciones más avanzadas que pueden ser útiles para usuarios avanzados como desarrolladores y programadores, una de estas es la de poder ir visualizando el resultado del código que se está escribiendo. Este software es libre y se distribuye bajo los términos de la Licencia Pública General de GNU.

Para su instalación busque en el CD que se envió junto a este material escrito la carpeta marcada como: NOTPAD++ y de doble click en el archivo instalador, como se muestra en la figura 120.

Nombre	Fecha de modifca...	Tipo
npp.5.9.2.Installer	27/06/2011 07:16 ...	Aplicación

Figura 120. Instalador de Notepad++



11. ANEXO DE INSTALACIONESB

Permita la instalación de este programa si su sistema operativo requiere permisos de instalación. Una vez lo permita deberá seleccionar el lenguaje a configurar en su instalación; como lo muestra la figura 121.

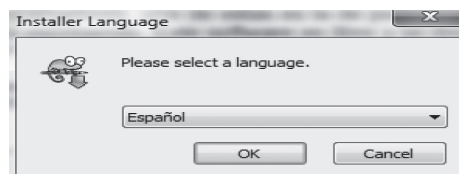


Figura 121. Seleccionando el idioma de instalación del Notepad++

Ahora indique al instalador que siga con el recorrido de la misma normalmente. Seleccione siguiente como lo muestra la figura 122.

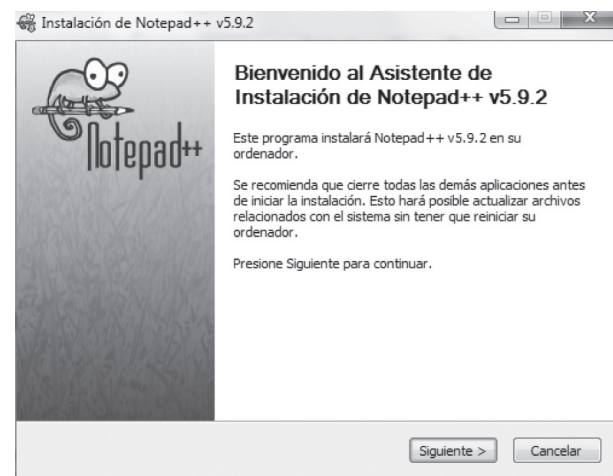


Figura 122. Instalando Notepad++

Ahora debe aceptar la licencia de instalación. Ver figura 123.

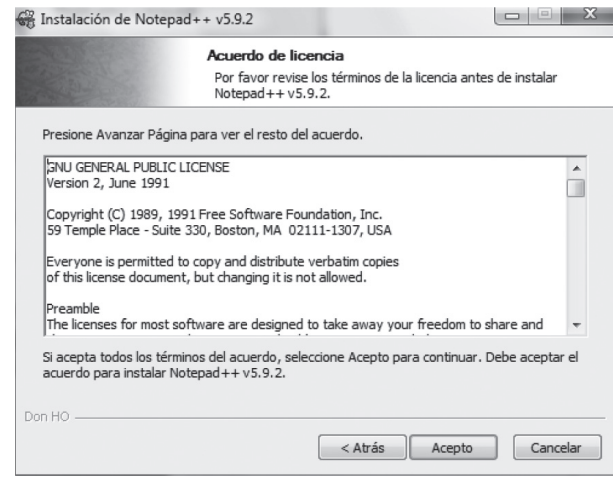
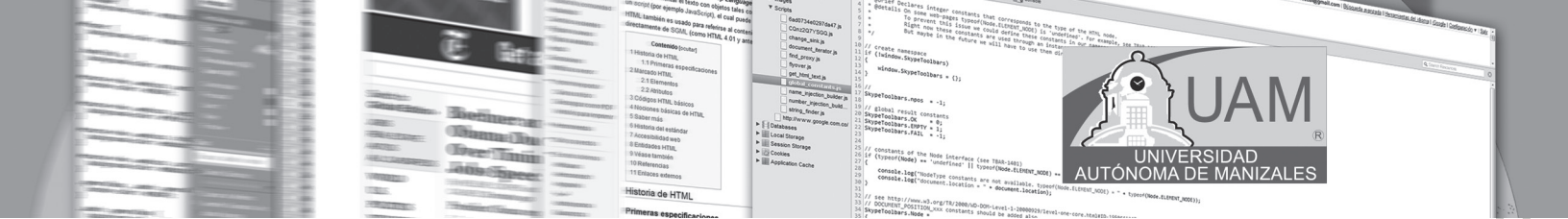


Figura 123. Licencia del Notepad++



Universidad Autónoma de Manizales

Ahora indique la ruta donde lo desea instalar; La ruta de instalación la selecciona cada uno de los usuarios, según su conveniencia. Ver figura 124.

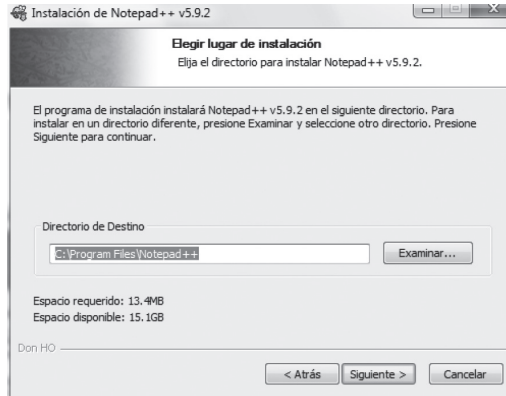


Figura 124. Seleccionando el lugar de instalación.

Seleccione los componentes. Esta parte se deja tal como aparece en la figura 125.

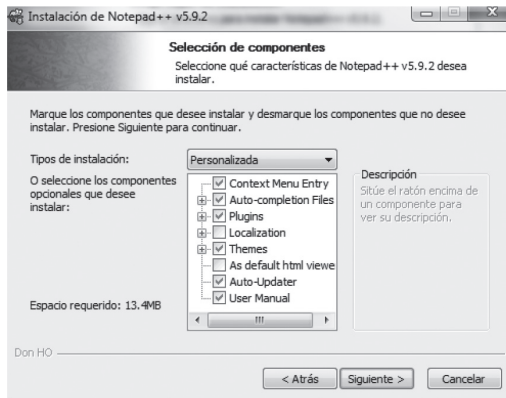


Figura 125. Componentes de instalación

Seleccione como se muestra en la figura 126, que se cree un ícono rápido para abrirlo directamente en el escritorio de su computador.

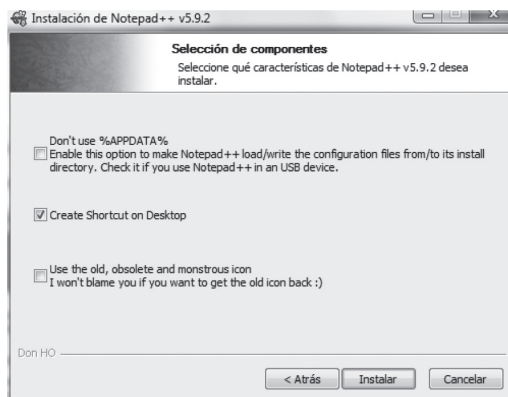


Figura 126. Creando ícono de apertura rápida



11. ANEXO DE INSTALACIONESB

Al seleccionar Instalar, el programa inicia su instalación en el disco duro, en la ruta que se le indicó en un paso anterior. Ver figura 127.

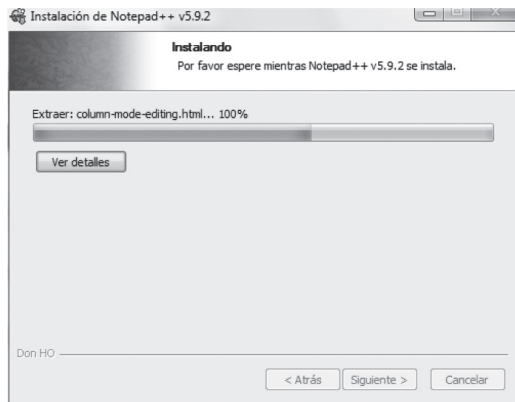


Figura 127. Recorrido de instalación

Al finalizar muestra una imagen parecida a la descrita en la figura 128. Ya está listo para trabajar con este programa.

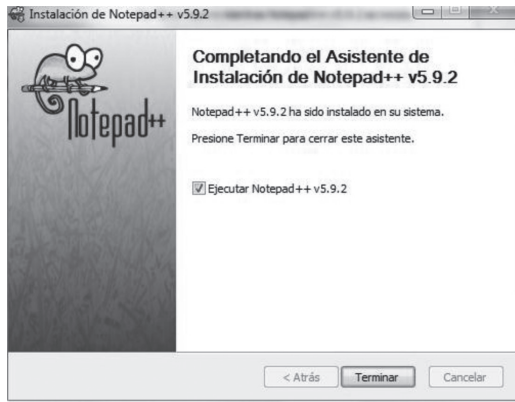


Figura 128. Finalización con éxito en la instalación del Notepad++

Al presionar Terminar se abre el programa como se muestra en la figura 129. El manejo de este programa se explica en las sesiones del módulo escrito.

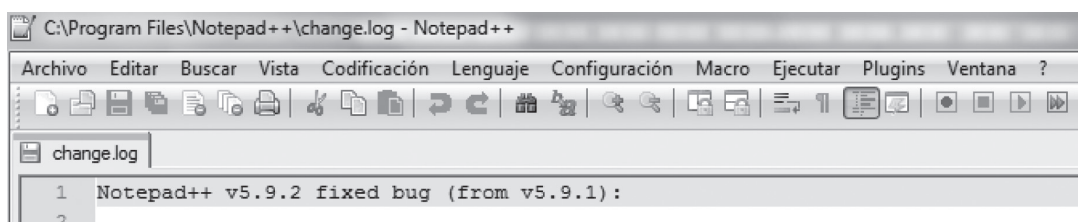
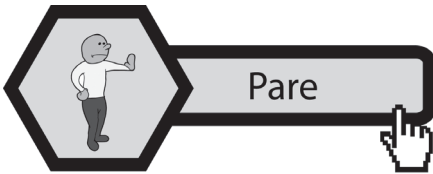


Figura 129. Apertura del Notepad++

11.2.3. Instalando el Happedit

HAPedit es el acrónimo para Html Asp Php editor; muy útil para todos los desarrolladores de webs dinámicas que necesiten un editor win32 en modo texto. Sus principales características son: coloreado de sintaxis para html/php, html/asp, html, JavaScript, css y sql; mostrar los resultados en el navegador; visor de proyectos; “compilación” de código php; edición de etiquetas HTML; completación de código; previsualización de imágenes, consola SQL, FTP Manager. Este software es **freeware**. Dentro de las recomendaciones que dan sus autores, una de ellas es: Este programa es freeware (libre), NO PUEDE SER VENDIDO O INTERCAMBIADO. Cada copia, duplicación, distribución, transferencia en red, línea telefónica u otro medio electrónico, para uso privado o colectivo es ALTAMENTE recomendado. La última versión se hizo el 28 Mayo 2004. La distribución que aquí se hace es netamente educativa y bajo ningún fin comercial. Tomado de: <http://hapedit.free.fr/hapedit.php?LangId=ES> [Junio 26 de 2011]



Pare

HAPPEDIT, no se instala. Este programa se despliega; esto significa que se toma la carpeta que encuentran en el CD enviado junto a su material escrito y copiarla en alguna ubicación de su disco duro. Luego se puede crear un acceso directo del ejecutable.

11.2.3.1. Creando acceso directo de Hapedit

La figura 130, se muestra como se crea el acceso directo y que archivo seleccionar para crear el acceso directo.

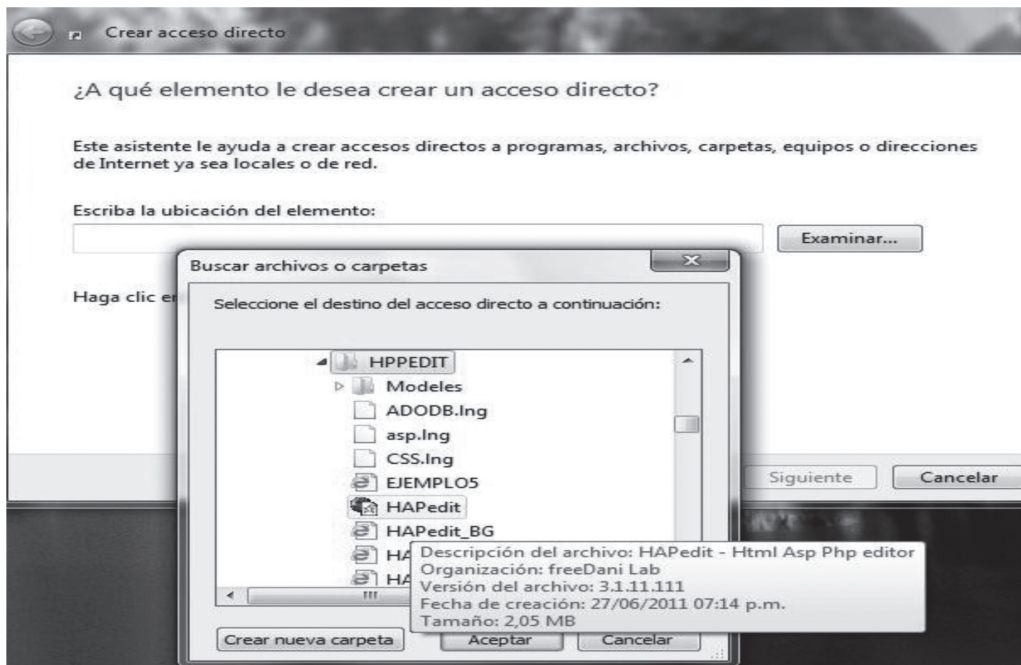


Figura 130. Creando acceso directo para HAPedit



11. ANEXO DE INSTALACIONESB

Al finalizar la creación del acceso directo en el escritorio se fija un ícono similar al mostrado en la figura 131.



Figura 131. Icono de acceso HAPEDIT

Al correr el programa se visualiza un entorno, similar al de la figura 132.

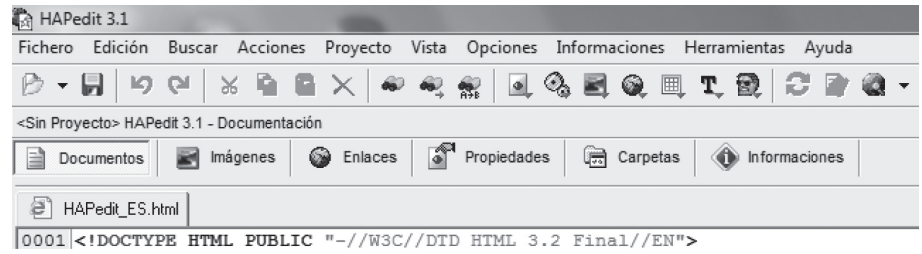


Figura 132. Entorno de desarrollo de HAPedit

11.2.4. Instalando el APPSERV

Appserv es una herramienta OpenSource para Windows que facilita la instalación de Apache, MySQL y PHP en la cual estas aplicaciones se configuran en forma automática. Como extra incorpora phpMyAdmin para el manejo de MySQL. La mayoría de de bases de datos y gestores de contenidos web requieren estos componentes. En ocasiones, configurarlos provoca quebraderos de cabeza. AppServ facilita mucho la labor y ahorra bastante tiempo.

Para su instalación busque en el CD que se envió junto a este material escrito la carpeta marcada como: APPSERV y de doble click en el archivo instalador, como se muestra en la figura 133.

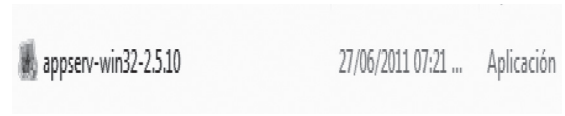
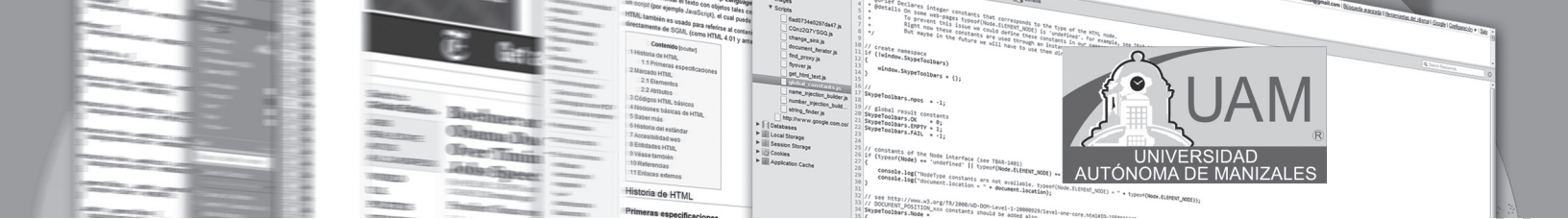


Figura 133. Instalador del appserv

Permita la instalación de este programa si su sistema operativo requiere permisos de instalación. Una vez lo permita deberá seleccionar el lenguaje a configurar en su instalación; como lo muestra la figura 134.



Figura 134. Inicio de instalación del Appserv



Universidad Autónoma de Manizales

Valide la licencia como se muestra en la figura 135.

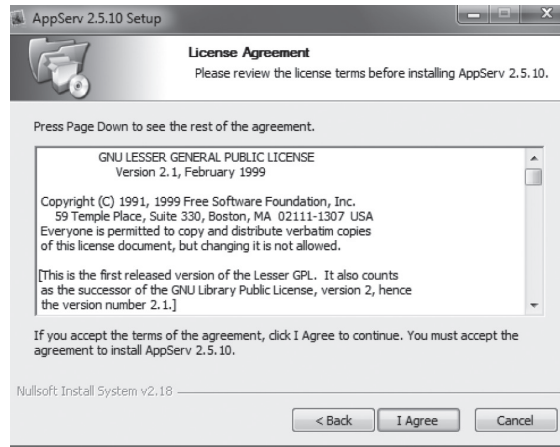


Figura 135. Aceptando licencia de appserv

Indique ahora el destino de instalación del programa. Ver figura 136

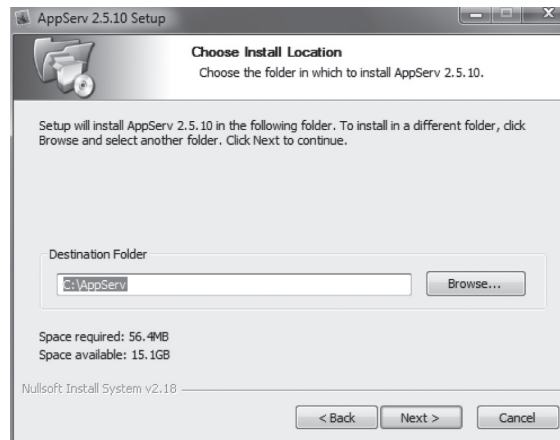


Figura 136. Seleccionando la ruta de instalación

Se seleccionan todos los componentes. Ver figura 137.

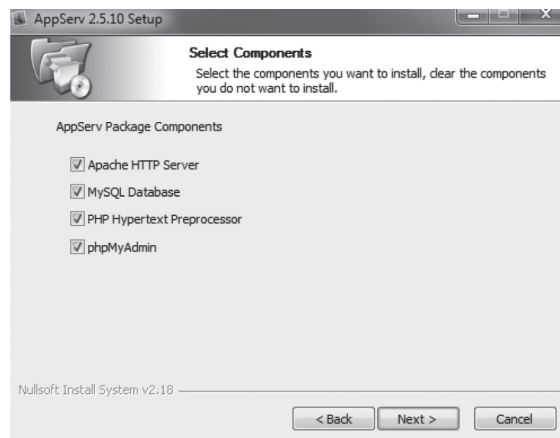


Figura 137. Seleccionando Componentes



11. ANEXO DE INSTALACIONESB

Esta parte es fundamental. Ahora se debe seleccionar el nombre del servidor y el correo del administrador. Escriba en el nombre del servidor localhost, en el mail escriba su mail y en el puerto deje el mismo por default el 80. Ver figura 138.

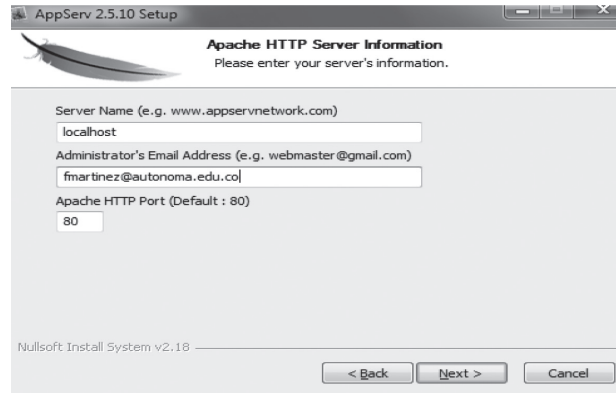


Figura 138. Información del servidor

Ahora escriba la contraseña del usuario que se llama root. La contraseña que usamos en forma estándar es: admin. Ver figura 139



Figura 139. Configuración del servidor

Al darle la instalación esta se inicializa. Permita la instalación de la misma, si el sistema operativo así lo indica. Ver figura 140.

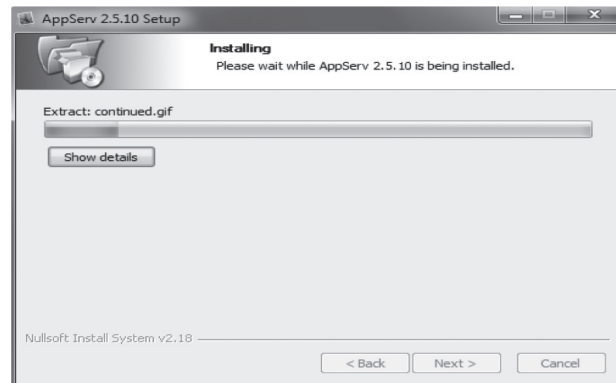
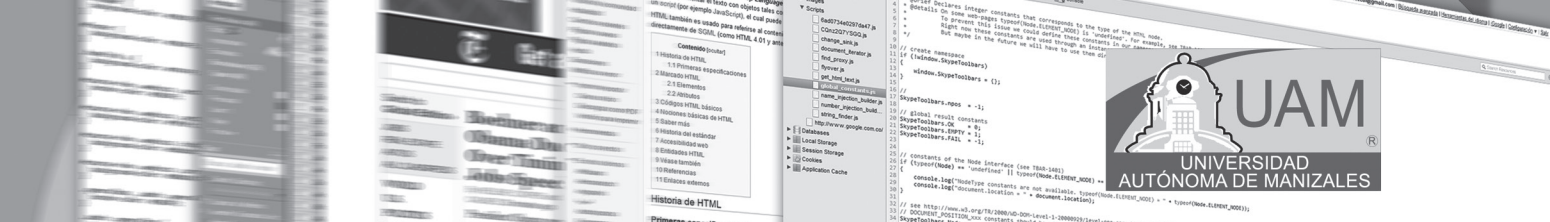


Figura 140. Recorrido de instalación



Universidad Autónoma de Manizales

Una vez termina la instalación el programa está listo para ser iniciado. Ver figura 141.

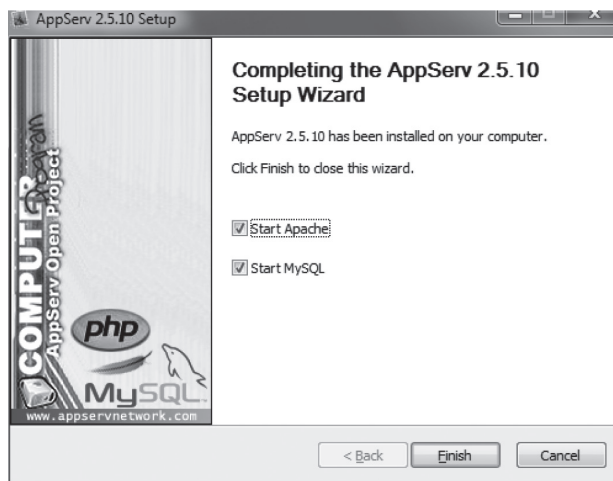
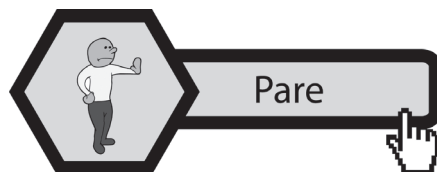


Figura 141. Completado la instalación del appserv



Pare

El appserv iniciará cada vez que se inicie Windows. Lo anterior garantiza que el servidor siempre esté arriba, para ejecutar cualquier aplicación en local. Para probar su instalación correcta, escriba en algún navegador: <http://localhost> La salida debe ser la página con todas las componentes (servidor Apache, php y Mysql) de Appserv.

11.2.5. Instalando FILEZILLA

FileZilla - es el mejor cliente FTP, gratuito. Posee una agradable e intuitiva interfaz. Sustenta FTP, SFTP y FTP sobre SSL. Es multiplataforma y está disponible para Linux, FreeBSD y MacOS X. Muy cómodo para actualizar archivos en su hosting. FileZilla es un cliente FTP multiplataforma de código abierto y software libre, licenciado bajo la Licencia Pública General de GNU. Soporta los protocolos FTP, SFTP y FTP sobre SSL/TLS (FTPS). Inicialmente fue diseñado para funcionar en Microsoft Windows, pero desde la versión 3.0.0, gracias al uso de wxWidgets, es multiplataforma, estando disponible además para otros sistemas operativos, entre ellos GNU/Linux, FreeBSD y Mac OS X.

Tomado de <http://es.wikipedia.org/wiki/FileZilla>. [Visitado el 26 de junio de 2011]

Para su instalación busque en el CD que se envió junto a este material escrito la carpeta marcada como: FTP, ingrese y de doble click en el archivo instalador, como se muestra en la figura 142.



11. ANEXO DE INSTALACIONESB

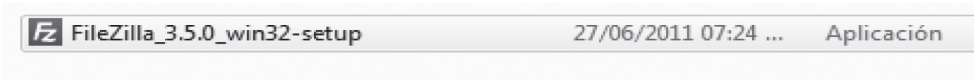


Figura 142. Instalador de Filezilla

Iniciando la instalación. La figura 143 muestra la ventana que sale al iniciar la instalación.

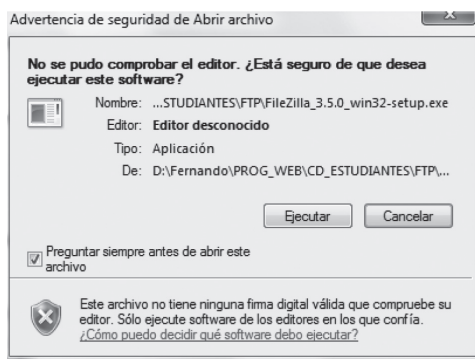


Figura 143. Instalando Filezilla

Una vez que se da permiso al sistema para que se instale, se indica la licencia de este programa. Ver figura 144.

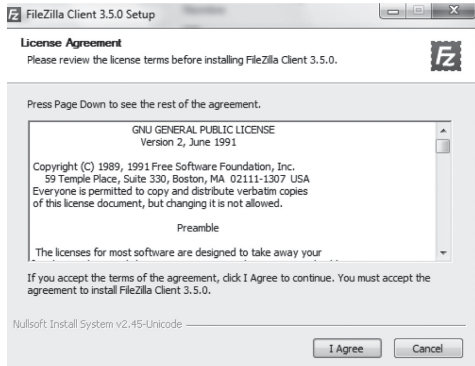


Figura 144. Licencia de Filezilla

La figura 145 indica la particularidad de instalación en un equipo para uso personal o para varios usuarios.

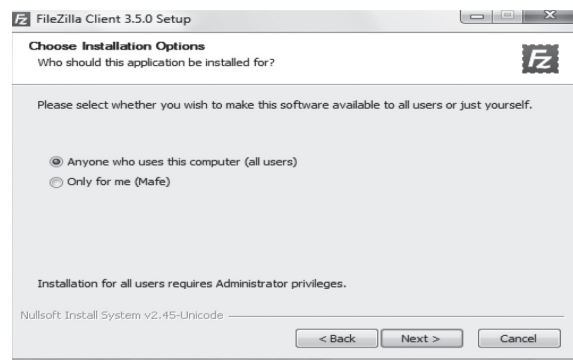
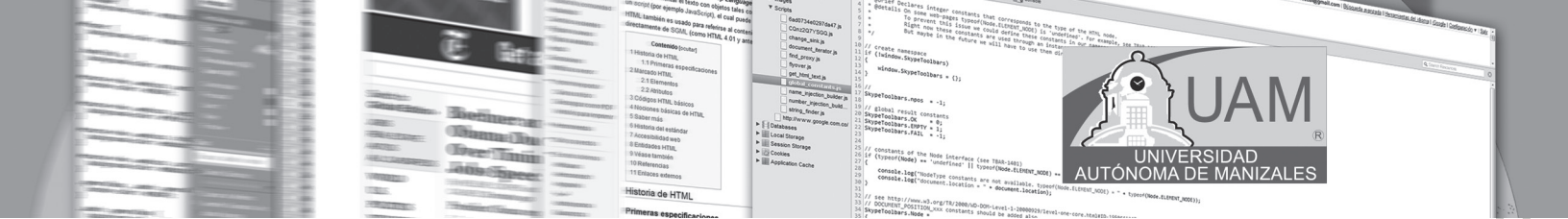


Figura 145. Selección de usos de instalación



Universidad Autónoma de Manizales

Componentes de instalación. Ver figura 146.

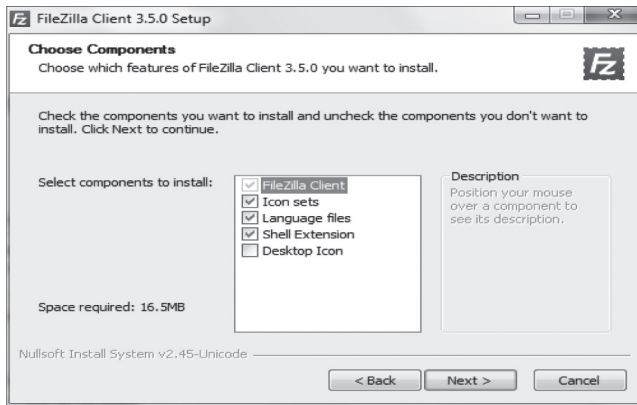


Figura 146. Componentes de instalación

Luego le debe indicar la ruta de instalación como en los programas anteriores. Al darle instalar el programa inicia su recorrido. Al final de la misma sale la venta de culminación de esta instalación como se muestra en la figura 147.

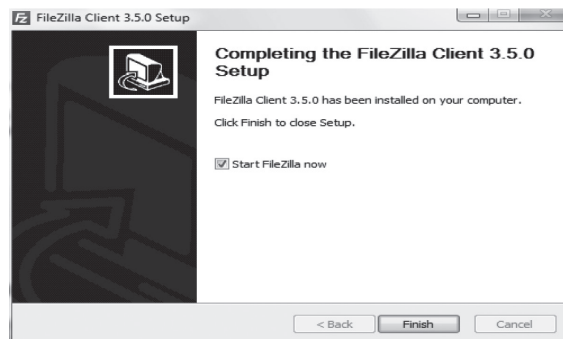


Figura 147. Finalización de instalación

Las figuras 148 muestran el ambiente de Filezilla y el icono creado en el escritorio para ingresar al mismo.

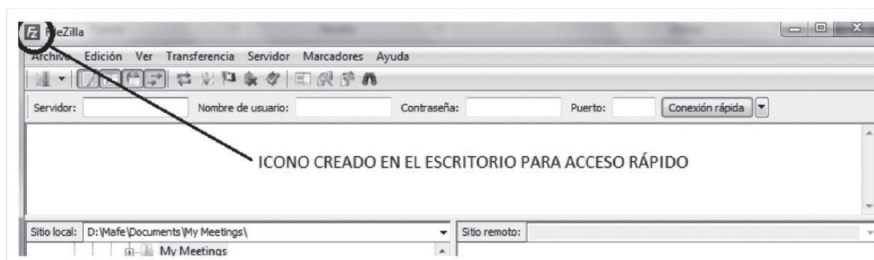


Figura 148. Ambiente de trabajo de Filezilla



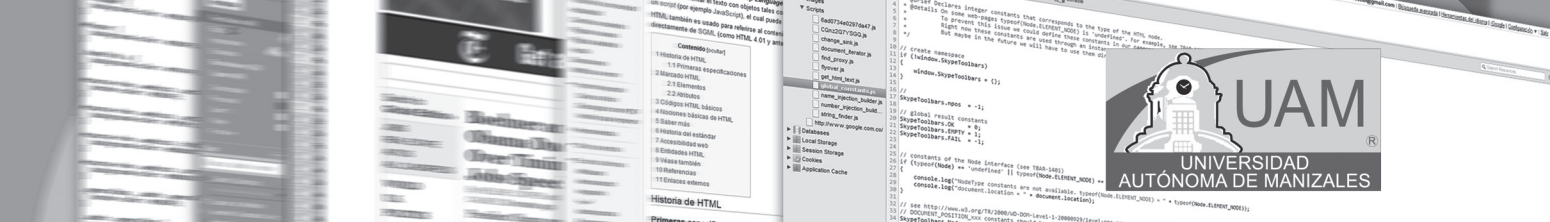
11. ANEXO DE INSTALACIONESB



Pare



El manejo de cada uno de estos paquetes de desarrollo se socializará a medida que se realicen las diversas tutorías sobre cada una de las temáticas.



Universidad Autónoma de Manizales



Programación WEB

```
51 DOCUMENT_POSITION_PRECEDING : 0x00,  
52 DOCUMENT_POSITION_FOLLOWING : 0x01,  
53 DOCUMENT_POSITION_CONTAINS : 0x02,  
54 DOCUMENT_POSITION_CONTAINED_BY : 0x03,  
55 DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC : 0x04  
56 }  
57  
58 // redefine setAttribute function to emulate DOM2 modified num  
59 // http://bugzilla.mozilla.org/show_bug.cgi?id=4911  
60 Element.prototype.setAttribute = Element.prototype.setAttribute  
61 Element.prototype.setAttribute = function(name, val)  
62 {  
63     // save the previous value  
64     var prev = this.getAttribute(name);  
65 }  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000
```

