



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha

### Funciones JS

Html

Versión: 1.0

Descripción:

Estructura Html

Código:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7   <link rel="stylesheet" href="bootstrap/css/bootstrap.css">
8 </head>
9 <body>
10  <div class="container">
11    <h2>Lista Uno</h2>
12    <select class="form-select" id="lista-numeros" aria-label="Default select example"></select>
13
14    <h2>Lista Dos</h2>
15    <select class="form-select" id="lista-dos-numeros" aria-label="Default select example"></select>
16  </div>
17
18  <script src="js/array.js"></script>
19  <script src="bootstrap/js/bootstrap.js"></script>
20 </body>
21 </html>
```

Arrays

Versión: 1.0

Descripción:

Genera lista en un menú de opciones

Código:

```
1 let arreglo = [];
2 let iteracion;
3 let resultadoLista;
4 let numero;
5
6 arreglo = [1,2,3,4,5,6,7,8,9,10];
7
8 for(iteracion=0; iteracion<arreglo.length; iteracion++){
9   resultadoLista += `<option>` + arreglo[iteracion] + `</option>`;
10 }
11
12 document.getElementById('lista-numeros').innerHTML = resultadoLista;
13
14 arreglo = [];
15
16 for(iteracion=0; iteracion<10; iteracion++){
17   numero = iteracion + 1;
18   arreglo.push(numero);
19 }
20 resultadoLista= '';
21
22 for(iteracion=0; iteracion<arreglo.length; iteracion++){
23   resultadoLista += `<option>` + arreglo[iteracion] + `</option>`;
24 }
25 document.getElementById('lista-dos-numeros').innerHTML = resultadoLista;
```



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha

### Funciones JS

Previa:

Lista Uno

1

Lista Dos

1

1

2

3

4

5

6

7

8

9

10

2.

Html

Versión: 1.0

Descripción:

Estructura Html

Código:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Tablas de Multiplicar</title>
7   <link rel="stylesheet" href="bootstrap/css/bootstrap.css">
8   <link rel="stylesheet" href="css/estilos.css">
9 </head>
10 <body>
11   <h1>Tabla de Multiplicar</h1>
12
13   <div class="container">
14     <div class="col-6">
15       <div class="accordion accordion-flush" id="accordionFlushExample">
16
17       </div>
18     </div>
19   </div>
20
21   <script src="bootstrap/js/bootstrap.bundle.js"></script>
22   <script src="js/tablaMulti.js"></script>
23 </body>
24 </html>
```



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha

### Funciones JS

#### Arrays

Versión: 1.0

#### Descripción:

Genera las tablas de multiplicar dentro de acordeonos

#### Código:

```
1 document.addEventListener("DOMContentLoaded", function(){
2   let tabla = [];
3   let multiplo = [];
4   let iteracionTabla;
5   let iteracionMultiplo;
6   let numeroTabla;
7   let numeroMultiplo;
8   let resultado;
9   let printResultado = '';
10
11   for(iteracionTabla = 0; iteracionTabla < 6; iteracionTabla++){
12     multiplo = [];
13     numeroTabla = iteracionTabla + 1;
14
15     for(iteracionMultiplo = 0; iteracionMultiplo < 10; iteracionMultiplo++) {
16       numeroMultiplo = iteracionMultiplo + 1;
17       resultado = numeroTabla * numeroMultiplo;
18       multiplo.push(resultado);
19     }
20     tabla.push(multiplo);
21   }
22
23   console.log(tabla);
24   for(iteracionTabla = 0; iteracionTabla < tabla.length; iteracionTabla++){
25     numeroTabla = iteracionTabla + 1;
26     printResultado += `<div class="accordion-item">`;
27     printResultado += `<button class="accordion-button collapsed" type="button" data-bs-toggle="collapse" data-bs-target="#flush-collapse` + numeroTabla + `" aria-expanded="false" aria-controls="flush-collapse` + numeroTabla + `">`;
28     printResultado += `Tabla de multiplicar #` + numeroTabla;
29     printResultado += `</button>`;
30     printResultado += `<div id="flush-collapse` + numeroTabla + `" class="accordion-collapse collapse" data-bs-parent="#accordionFlushExample">`;
31
32     for(iteracionMultiplo = 0; iteracionMultiplo < 10; iteracionMultiplo++){
33       numeroMultiplo = iteracionMultiplo + 1;
34
35       printResultado += `<div class="accordion-body" style="padding: 0 10px;>`;
36       printResultado += `<table class="table">`;
37       printResultado += `<tr style="padding:0;margin:0;"><td style="padding:0;margin:0;">` + numeroTabla + ` X ` + numeroMultiplo + ` = ` + tabla[iteracionTabla][iteracionMultiplo] + `</td></tr>`;
38       printResultado += `</table>`;
39       printResultado += `</div>`;
40     }
41
42     printResultado += `</div>`;
43     printResultado += `</div>`;
44   }
45   document.getElementById('accordionFlushExample').innerHTML = printResultado;
46 });
```

#### Previa:

### Tabla de Multiplicar

Tabla de multiplicar #1	^
1 X 1 = 1	
1 X 2 = 2	
1 X 3 = 3	
1 X 4 = 4	
1 X 5 = 5	
1 X 6 = 6	
1 X 7 = 7	
1 X 8 = 8	
1 X 9 = 9	
1 X 10 = 10	
Tabla de multiplicar #2	v
Tabla de multiplicar #3	v
Tabla de multiplicar #4	v
Tabla de multiplicar #5	v



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha

### Funciones JS

3.

Html	Versión: 1.0
<b>Descripción:</b> Estructura Html	
<b>Código:</b>	
<pre>1  &lt;!DOCTYPE html&gt; 2  &lt;html lang="en"&gt; 3  &lt;head&gt; 4      &lt;meta charset="UTF-8"&gt; 5      &lt;meta name="viewport" content="width=device-width, initial-scale=1.0"&gt; 6      &lt;title&gt;Document&lt;/title&gt; 7      &lt;link rel="stylesheet" href="bootstrap/css/bootstrap.css"&gt; 8  &lt;/head&gt; 9  &lt;body&gt; 10 11      &lt;div class="container"&gt; 12          &lt;h1&gt;Factorial&lt;/h1&gt; 13          &lt;table class="table"&gt; 14              &lt;thead&gt; 15                  &lt;tr&gt; 16                      &lt;th scope="col"&gt;Numero&lt;/th&gt; 17                      &lt;th scope="col"&gt;Factorial&lt;/th&gt; 18                  &lt;/tr&gt; 19              &lt;/thead&gt; 20              &lt;tbody id="lista-factorial"&gt; 21 22              &lt;/tbody&gt; 23          &lt;/table&gt; 24      &lt;/div&gt; 25 26      &lt;script src="js/factorial.js"&gt;&lt;/script&gt; 27  &lt;/body&gt; 28  &lt;/html&gt;</pre>	



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha

### Funciones JS

Nombre de la función:

Versión: 1.0

Descripción:

Genera la factorial de 5 dentro de un tabla

Código:

```
1
2 let arreglo = [];
3 let iteracion;
4 let factorial;
5 let numero;
6 let resultaLista = "";
7
8 numero = 5;
9 factorial = 1;
10
11 for(iteracion = 1; iteracion <= numero; iteracion++){
12     factorial = factorial * iteracion;
13     arreglo.push({ numero: iteracion, factorial: factorial });
14 }
15
16 for (iteracion = 0; iteracion < arreglo.length; iteracion++) {
17     resultaLista += `<tr><td>${arreglo[iteracion].numero}</td><td>${arreglo[iteracion].factorial}</td></tr>`;
18 }
19
20 document.getElementById('lista-factorial').innerHTML = resultaLista;
```

Previa:

## Factorial

Numero	Factorial
1	1
2	2
3	6
4	24
5	120



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha

### Funciones JS

4.

Html

Versión: 1.0

Descripción:

Estructura Html

Código:

```
1 <!DOCTYPE html>
2 <html Lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Bingo</title>
7   <link rel="stylesheet" href="bootstrap/css/bootstrap.css">
8   <link rel="stylesheet" href="css/estilos.css">
9 </head>
10 <body style="background-color: #def0d8;">
11   <div class="container" id="bingo-1"></div>
12
13
14   <script src="bootstrap/js/bootstrap.bundle.js"></script>
15   <script src="js/bingo.js"></script>
16 </body>
17 </html>
```



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha

### Funciones JS

**Nombre de la función:**

**Versión: 1.0**

**Descripción:**

Genera un bingo con x resaltadas

**Código:**



# Tecnólogo en Análisis y Desarrollo de Software

## Ficha

### Funciones JS

```
1 document.addEventListener('DOMContentLoaded', function(){
2     let bingo = [];
3     let interno = [];
4     let iteracion1;
5     let iteracion2;
6     let contador = 0;
7     let tabla;
8     let printResultado = '';
9     let letras = ['B', 'I', 'N', 'O', 'G', 'O'];
10
11
12     //Bingo
13     console.log("\nBingo");
14     for(iteracion1=0; iteracion1<5; iteracion1++){
15         interno=[];
16         for(iteracion2=0; iteracion2<5; iteracion2++){
17             contador=contador+1;
18             tabla=contador*3;
19             interno.push(tabla);
20         }
21         bingo.push(interno);
22     }
23     console.log(bingo);
24
25     //Bingo simple
26
27
28     printResultado+=`<table class="table"><tr>`;
29     for (let contadorB = 0; contadorB < letras.length; contadorB++) {
30         printResultado+= `<th class = "borde color-letra">` + letras[contadorB] + `</th>`;
31     };
32     printResultado+=`</tr>`;
33     for(iteracion1 = 0; iteracion1 < 5; iteracion1++) {
34         printResultado+=`<tr>`
35         for(iteracion2 = 0; iteracion2 < 5; iteracion2++) {
36             printResultado+=`<td class="borde tabla-color">`+bingo[iteracion1][iteracion2]+`</td>`;
37         }
38         printResultado+=`</tr>`;
39     }
40     printResultado+=`</table>`;
41
42
43
44     //Bingo letras
45     for (iteracion1 = 0; iteracion1 < 5; iteracion1++) {
46         printResultado += `<table class="table">`;
47         printResultado += `<tr>`;
48
49         for (let contadorB = 0; contadorB < letras.length; contadorB++) {
50             printResultado += `<th class = "borde color-letra">` + letras[contadorB] + `</th>`;
51         };
52         printResultado += `</tr>`;
53         for (let contador = 0; contador < 5; contador++) {
54             printResultado += `<tr>`;
55             for (iteracion2 = 0; iteracion2 < 5; iteracion2++) {
56                 if (iteracion2 == iteracion1) {
57                     printResultado += `<td class = "borde tabla-color color-1">` + bingo[contador][iteracion2] + `</td>`;
58                 } else {
59                     printResultado += `<td class = "borde tabla-color">` + bingo[contador][iteracion2] + `</td>`;
60                 };
61             };
62             printResultado += `</tr>`;
63         };
64         printResultado += `</table>`;
65     };
66
67
68
69
70     //X grande
71     printResultado += `<table class="table"><tr>`;
72     for(contadorB=0; contadorB<5; contadorB++){
73         printResultado += `<th scope="col" class = "color-letra borde">`+letras[contadorB]+`</th>`;
74     }
75     printResultado += `</tr>`;
76     for(iteracion1=0; iteracion1<5; iteracion1++){
77         printResultado += `<tr>`;
78         for(iteracion2=0; iteracion2<5; iteracion2++){
79             if(iteracion1==iteracion2 || iteracion1-iteracion2==4){
80                 printResultado += `<td class="borde tabla-color color-x grande">`+bingo[iteracion1][iteracion2]+`</td>`;
81             }else{
82                 printResultado += `<td class="borde tabla-color">`+bingo[iteracion1][iteracion2]+`</td>`;
83             }
84         }
85         printResultado += `</tr>`;
86     }
87     printResultado += `</table>`;
88
89
90
91     //X varias
92     printResultado += `<table class="table"><tr>`;
93     for (let contadorB = 0; contadorB < letras.length; contadorB++) {
94         printResultado += `<th class = "borde color-letra">` + letras[contadorB] + `</th>`;
95     }
96     printResultado += `</tr>`;
97     for (iteracion1 = 0; iteracion1 < 5; iteracion1++) {
98         printResultado += `<tr>`;
99         for (iteracion2 = 0; iteracion2 < 5; iteracion2++) {
100             //X1
101
102             if (iteracion1 <= 3 && iteracion2 >= 1 && iteracion2 <= 4 && iteracion1 % 2 == 0 && iteracion2 % 2 == 1) {
103                 printResultado += `<td class="color-1 borde tabla-color">` + bingo[iteracion1][iteracion2] + `</td>`;
104             } else if (iteracion1 <= 2 && iteracion2 >= 1 && iteracion2 <= 2 && iteracion1 % 2 == 1 && iteracion2 % 2 == 0) {
105                 printResultado += `<td class="color-1 borde tabla-color">` + bingo[iteracion1][iteracion2] + `</td>`;
106             }
107             //X2
108
109             } else if (iteracion1>2&&iteracion2<=3&&iteracion1%2==0&&iteracion2%2==0){
110                 printResultado += `<td class="color-2 borde tabla-color">` + bingo[iteracion1][iteracion2] + `</td>`;
111             }else if (iteracion1>2&&iteracion2<=3&&iteracion1%2==1&&iteracion2%2==1){
112                 printResultado += `<td class="color-2 borde tabla-color">` + bingo[iteracion1][iteracion2] + `</td>`;
113             }
114             //X3
115
116             }else if (iteracion1>2&&iteracion2>=3&&iteracion1%2==0&&iteracion2%2==0){
117                 printResultado += `<td class="color-3 borde tabla-color">` + bingo[iteracion1][iteracion2] + `</td>`;
118             }else if (iteracion1>2&&iteracion2>=3&&iteracion1%2==1&&iteracion2%2==1){
119                 printResultado += `<td class="color-3 borde tabla-color">` + bingo[iteracion1][iteracion2] + `</td>`;
120             }else{
121                 printResultado += `<td class="borde tabla-color">` + bingo[iteracion1][iteracion2] + `</td>`;
122             }
123         }
124         printResultado += `</tr>`;
125     }
126     printResultado += `</table>`;
127
128     document.getElementById('bingo-1').innerHTML = printResultado;
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 }
```





Tecnólogo en Análisis y Desarrollo de  
Software  
Ficha

**Funciones JS**