

```

01.  /* Objeto de Datos */
02.  var datos = {
03.      fun: [],
04.      xi: [],
05.      fi: [],
06.      h: 0.001,
07.      tam: {n:3,m:3}
08.  };
09.  function initialEvents(){
10.      $('#btnCalcular').click(function(){
11.          init();
12.          $('#tablasDer').html("");
13.          getNewtonRaphson();
14.      });
15.      $('#btnCalcularPaso').click(function(){
16.          init();
17.          $('#tablasDer').html("");
18.          __getNewtonRaphson();
19.      });
20.      $('#btnReset').click(function(){
21.          $('#tablasDer').html("");
22.          resetDatos();
23.      });
24.      $('#btnDefault').click(function(){
25.          $('#tablasDer').html("");
26.          resetDatos();
27.          defaultDatos();
28.      });
29.  }
30.
31.  /* Funciones */
32.  function defaultDatos(){
33.      $('#initialX').val('1');
34.      $('#initialY').val('3');
35.      $('#initialZ').val('0.5');
36.      $('#initialF1').val('function(x,y,z){ return (x*x)-(2*y)+(x*z)+2 }');
37.      $('#initialF2').val('function(x,y,z){ return (x*y*z)+(2*z)-3 }');
38.      $('#initialF3').val('function(x,y,z){ return (y*y)+(z*z)-6 }');
39.  }
40.  function resetDatos(){
41.      /* Obteniendo matriz Xo */
42.      var initialVar = document.forms.initialVar.children;
43.      [].forEach.call(initialVar,function(i,index){
44.          i.children[1].value = "";
45.      });
46.      /* Obteniendo matriz de funciones */
47.      var initialFun = document.forms.initialFun.children;
48.      [].forEach.call(initialFun,function(i,index){
49.          i.children[1].value = "";
50.      });
51.      datos = {
52.          fun: [],
53.          xi: [],
54.          fi: [],
55.          h: 0.001,
56.          tam: {n:3,m:3}
57.      };
58.  }
59.  function init(){
60.      var cont = 0;
61.      /* Obteniendo matriz Xo */
62.      var initialVar = document.forms.initialVar.children;
63.      [].forEach.call(initialVar,function(i,index){
64.          if(i.children[1].value == ""){
65.              cont++;
66.          }else{
67.              datos.xi[index] = parseFloat(i.children[1].value);
68.          }

```

```

69.     });
70.     /* Obteniendo matriz de funciones */
71.     var initialFun = document.forms.initialFun.children;
72.     [].forEach.call(initialFun,function(i,index){
73.         if(i.children[1].value == ""){
74.             cont++;
75.         }else{
76.             datos.fun[index] = eval('(' + i.children[1].value + ')');
77.         }
78.     });
79.     /* Resolver las funciones con los valores iniciales Matriz Fo*/
80.
81.     datos.xi.forEach(function(i,index){
82.         datos.fi[index] = datos.fun[index].apply(this,datos.xi);
83.     });
84.     if(cont>0){
85.         $('#msg').text('Le falta llenar datos');
86.     }
87.     else{
88.         $('#msg').text('');
89.     }
90.     getJacobi();
91. }
92. function derivada(index_fun,index_xi){
93.     var d = 0,
94.         n = 0,
95.         x = datos.xi[index_xi],
96.         fun = datos.fun[index_fun],
97.         h = datos.h,
98.         fi = datos.fi[index_fun],
99.         fi1 = 0;
100.    n++;
101.    x = x+h;
102.    switch(index_xi){
103.        case 0: fi1 = fun(x,datos.xi[1],datos.xi[2]);break;
104.        case 1: fi1 = fun(datos.xi[0],x,datos.xi[2]);break;
105.        case 2: fi1 = fun(datos.xi[0],datos.xi[1],x);break;
106.        default:break;
107.    }
108.    d = (fi1 - fi)/h;
109.    return d;
110.
111. }
112.
113. function getJacobi(){
114.     var jacobí = new Array();
115.
116.     for(var i =0;i<datos.tam.n;i++){
117.         jacobí[i] = new Array();
118.         for(var j =0;j<datos.tam.m;j++){
119.             jacobí[i].push(derivada(i,j));
120.         }
121.     }
122.     return jacobí;
123. }
124.
125. function getNewtonRaphson(){
126.     var jacobí = new Matrix(getJacobi()),
127.         invJacobí = jacobí.clone().inverse(),
128.         x0 = new Matrix(new Array(datos.xi)),
129.         f0 = new Matrix(new Array(datos.fi)),
130.         cont = 0,
131.         xr = new Matrix([[0,0,0]]);
132.
133.     while(cont<10){
134.         cont++;
135.         xr = x0.subtract(f0.multiply(invJacobí));
136.         if(compare(xr,x0)){
137.             break;
138.         }

```

```

139.         x0 = xr.clone();
140.     }
141.     var html = "<h3>Resultado</h3><table class='table table-striped table-hover table-bordered'>
<tbody>";
142.     for(var i=0;i<xr.cols;i++){
143.         html+= "<tr><td>"+xr[0][i].toPrecision(4)+"</td></tr>";
144.     }
145.     html+="</tbody></table>";
146.     $('#tablasDer').append(html);
147.     return xr;
148. }
149.
150. function compare(x,y){
151.     var cont = 0;
152.     for(var i=0;i<x.rows;i++){
153.         for(var j=0;j<x.cols;j++){
154.             var a = parseFloat(x[i][j].toPrecision(4));
155.             var b = parseFloat(y[i][j].toPrecision(4));
156.             if(a === b){
157.                 cont++;
158.             }
159.         }
160.     }
161.     if(cont == x.cols){
162.         return true;
163.     }
164.     else{
165.         return false;
166.     }
167. }
168.
169.
170. /* Paso por paso functions */
171.
172. function __derivada(index_fun,index_xi){
173.     var d = 0,
174.         n = 0,
175.         x = datos.xi[index_xi],
176.         fun = datos.fun[index_fun],
177.         h = datos.h,
178.         fi = datos.fi[index_fun],
179.         fi1 = 0;
180.     var nombreVar = "";
181.     switch(index_xi){
182.         case 0: nombreVar = "X";break;
183.         case 1: nombreVar = "Y";break;
184.         case 2: nombreVar = "Z";break;
185.         default:break;
186.     }
187.     var nombreFun = "";
188.     switch(index_fun){
189.         case 0: nombreFun = "f1";break;
190.         case 1: nombreFun = "f2";break;
191.         case 2: nombreFun = "f3";break;
192.         default:break;
193.     }
194.     var html = "<h6>d"+nombreFun+"/d"+nombreVar+"</h6><table class='table table-striped table-
hover table-bordered'><thead class='thead-dark'><tr><th>n</th><th>"+ nombreVar +"</th>
<th>"+nombreFun+"</th><th>d</th></tr></thead><tbody>";
195.     html+= "<tr><td>"+n+"</td><td>"+x+"</td><td>"+fi+"</td><td></td></tr>";
196.
197.     n++;
198.     x = x+h;
199.     switch(index_xi){
200.         case 0: fi1 = fun(x,datos.xi[1],datos.xi[2]);break;
201.         case 1: fi1 = fun(datos.xi[0],x,datos.xi[2]);break;
202.         case 2: fi1 = fun(datos.xi[0],datos.xi[1],x);break;
203.         default:break;
204.     }
205.     d = (fi1 - fi)/h;

```

```

206.         html+= "<tr><td></td><td></td><td></td><td>" + d + "</td></tr>";
207.         html+= "<tr><td>" + n + "</td><td>" + x + "</td><td>" + f1 + "</td><td></td></tr>";
208.
209.         html+= "</tbody></table>";
210.         $('#tablasDer').append(html);
211.         return d;
212.
213.     }
214.
215.     function __getJacobi(){
216.         var jacobi = new Array();
217.         $('#tablasDer').append('<h4>Calcular Derivadas</h4><br>');
218.         var html = "<h4>Matriz Jacobi</h4><table class='table table-striped table-hover table-  
bordered'><tbody>";
219.         for(var i = 0; i < datos.tam.n; i++){
220.             jacobi[i] = new Array();
221.             html+= "<tr>";
222.             for(var j = 0; j < datos.tam.m; j++){
223.                 jacobi[i].push(__derivada(i,j));
224.                 html+= "<td>" + jacobi[i][j].toPrecision(4) + "</td>";
225.             }
226.             html+= "</tr>";
227.         }
228.         html+= "</tbody></table>";
229.         $('#tablasDer').append(html);
230.         return jacobi;
231.     }
232.
233.     function __getNewtonRaphson(){
234.         var jacobi = new Matrix(__getJacobi()),
235.             invJacobi = jacobi.clone().inverse(),
236.             x0 = new Matrix(new Array(datos.xi)),
237.             f0 = new Matrix(new Array(datos.fi)),
238.             cont = 0,
239.             xr = new Matrix([[0,0,0]]);
240.
241.         var html = "<h4>Matriz Inversa Jacobi</h4><table class='table table-striped table-hover table-  
bordered'><tbody>";
242.         for(var i = 0; i < invJacobi.rows; i++){
243.             html+= "<tr>";
244.             for(var j = 0; j < invJacobi.cols; j++){
245.                 html+= "<td>" + invJacobi[i][j].toPrecision(4) + "</td>";
246.             }
247.             html+= "</tr>";
248.         }
249.         html+= "</tbody></table>";
250.         $('#tablasDer').append(html);
251.
252.         while(cont < 10){
253.             cont++;
254.             xr = x0.subtract(f0.multiply(invJacobi));
255.             if(compare(xr, x0)){
256.                 break;
257.             }
258.             x0 = xr.clone();
259.         }
260.         var html = "<h4>Resultado</h4><table class='table table-striped table-hover table-bordered'>  
<tbody>";
261.         for(var i = 0; i < xr.cols; i++){
262.             html+= "<tr><td>" + xr[0][i].toPrecision(4) + "</td></tr>";
263.         }
264.         html+= "</tbody></table>";
265.         $('#tablasDer').append(html);
266.         return xr;
267.     }
268.
269.     /* Inicializar Eventos */
270.     initialEvents();

```