```
/* Objeto de Datos */
var datos = {
    fun: [],
    xi:  [],
    fi:  [],
    h:   0.001,
    tam: {n:2,m:2}
};
function initialEvents(){
    $('#btnCalcular').click(function(){
        init();
        $('#tablasDer').html("");
        getNewtonRaphson();
    });
    $('#btnCalcularPaso').click(function(){
        init();
        $('#tablasDer').html("");
        __getNewtonRaphson();
    });
    $('#btnReset').click(function(){
        $('#tablasDer').html("");
        resetDatos();
    });
    $('#btnDefault').click(function(){
        $('#tablasDer').html("");
        resetDatos();
        defaultDatos();
    });
}

/* Funciones */
function defaultDatos(){
    $('#initialX').val('1');
    $('#initialY').val('2');
    $('#initialF1').val('function(x1,x2){ return 2*(x1*x1)-x1*x2+x2-10 }');
    $('#initialF2').val('function(x1,x2){ return x1+2*x2+(x2*x2)-6 }');
}
function resetDatos(){
    /* reseteando matriz Xo */
    var initialVar = document.forms.initialVar.children;
    [].forEach.call(initialVar,function(i,index){
        i.children[1].value = "";
    });
    /* reseteando matriz de funciones */
    var initialFun = document.forms.initialFun.children;
    [].forEach.call(initialFun,function(i,index){
        i.children[1].value = "";
    });
    datos = {
        fun: [],
        xi:  [],
        fi:  [],
        h:   0.001,
        tam: {n:2,m:2}
    };
}
function init(){
    var cont = 0;
    /* Obteniendo matriz Xo */
    var initialVar = document.forms.initialVar.children;
    [].forEach.call(initialVar,function(i,index){
        if(i.children[1].value == ""){
            cont++;
        }else{
            datos.xi[index] = parseFloat(i.children[1].value);
        }
```

```
    });
    /* Obteniendo matriz de funciones */
    var initialFun = document.forms.initialFun.children;
    [].forEach.call(initialFun,function(i,index){
        if(i.children[1].value == ""){
            cont++;
        }else{
            datos.fun[index] = eval('('+i.children[1].value+')');
        }
    });
    /* Resolver las funciones con los valores iniciales Matriz Fo*/

    datos.xi.forEach(function(i,index){
        datos.fi[index] = datos.fun[index].apply(this,datos.xi);
    });
    if(cont>0){
        $('#msg').text('Le falta llenar datos');
    }
    else{
        $('#msg').text('');
    }
    getJacobi();
}
function derivada(index_fun,index_xi){
    var d   = 0,
        n   = 0,
        x   = datos.xi[index_xi],
        fun = datos.fun[index_fun],
        h   = datos.h,
        fi  = datos.fi[index_fun],
        fi1 = 0;
        n++;
        x = x+h;
            switch(index_xi){
                case 0: fi1 = fun(x,datos.xi[0],datos.xi[1]);break;
                case 1: fi1 = fun(datos.xi[0],x,datos.xi[1]);break;
                default:break;
            }
            d = (fi1 - fi)/h;
            return d;

    }

    function getJacobi(){
        var jacobi = new Array();

        for(var i =0;i<datos.tam.n;i++){
            jacobi[i] = new Array();
            for(var j =0;j<datos.tam.m;j++){
                jacobi[i].push(derivada(i,j));
            }
        }
        return jacobi;
    }

    function getNewtonRaphson(){
        var jacobi    = new Matrix(getJacobi()),
            invJacobi = jacobi.clone().inverse(),
            x0        = new Matrix(new Array(datos.xi)),
            f0        = new Matrix(new Array(datos.fi)),
            cont      = 0,
            xr        = new Matrix([[0,0,0]]);

        while(cont<10){
            cont++;
            xr = x0.subtract(f0.multiply(invJacobi));
```

```
133.                if(compare(xr,x0)){
134.                    break;
135.                }
136.                x0 = xr.clone();
137.            }
138.            var html = "<h3>Resultado</h3><table class='table table-striped table-
    hover table-bordered'><tbody>";
139.            for(var i=0;i<xr.cols;i++){
140.                html+= "<tr><td>"+xr[0][i].toPrecision(4)+"</td></tr>";
141.            }
142.            html+="</tbody></table>";
143.            $('#tablasDer').append(html);
144.            return xr;
145.        }
146.
147.        function compare(x,y){
148.            var cont = 0;
149.            for(var i=0;i<x.rows;i++){
150.                for(var j=0;j<x.cols;j++){
151.                    var a = parseFloat(x[i][j].toPrecision(4));
152.                    var b = parseFloat(y[i][j].toPrecision(4));
153.                    if(a === b){
154.                        cont++;
155.                    }
156.                }
157.            }
158.            if(cont == x.cols){
159.                return true;
160.            }
161.            else{
162.                return false;
163.            }
164.        }
165.
166.
167.        /* Paso por paso functions */
168.
169.        function __derivada(index_fun,index_xi){
170.            var d   = 0,
171.                n   = 0,
172.                x   = datos.xi[index_xi],
173.                fun = datos.fun[index_fun],
174.                h   = datos.h,
175.                fi  = datos.fi[index_fun],
176.                fi1 = 0;
177.                var nombreVar = "";
178.                switch(index_xi){
179.                    case 0: nombreVar = "X";break;
180.                    case 1: nombreVar = "Y";break;
181.                    case 2: nombreVar = "Z";break;
182.                    default:break;
183.                }
184.                var nombreFun = "";
185.                switch(index_fun){
186.                    case 0: nombreFun = "f1";break;
187.                    case 1: nombreFun = "f2";break;
188.                    case 2: nombreFun = "f3";break;
189.                    default:break;
190.                }
191.                var html = "<h6>d"+nombreFun+"/d"+nombreVar+"</h6><table class='tabl
    e 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>";
192.                html+= "<tr><td>"+n+"</td><td>"+x+"</td><td>"+fi+"</td><td></td></tr
    >";
193.
```

```
194.                 n++;
195.                 x = x+h;
196.                 switch(index_xi){
197.                     case 0: fi1 = fun(x,datos.xi[1],datos.xi[2]);break;
198.                     case 1: fi1 = fun(datos.xi[0],x,datos.xi[2]);break;
199.                     case 2: fi1 = fun(datos.xi[0],datos.xi[1],x);break;
200.                     default:break;
201.                 }
202.                 d = (fi1 - fi)/h;
203.                 html+= "<tr><td></td><td></td><td></td><td>"+d+"</td></tr>";
204.                 html+= "<tr><td>"+n+"</td><td>"+x+"</td><td>"+fi1+"</td><td></td></t
    r>";
205. 
206.                 html+="</tbody></table>";
207.                 $('#tablasDer').append(html);
208.                 return d;
209. 
210.         }
211. 
212.         function __getJacobi(){
213.             var jacobi = new Array();
214.             $('#tablasDer').append('<h4>Calcular Derivadas</h4><br>');
215.             var html = "<h4>Matriz Jacobi</h4><table class='table table-
    striped table-hover table-bordered'><tbody>";
216.             for(var i =0;i<datos.tam.n;i++){
217.                 jacobi[i] = new Array();
218.                 html+= "<tr>";
219.                 for(var j =0;j<datos.tam.m;j++){
220.                     jacobi[i].push(__derivada(i,j));
221.                     html+= "<td>"+jacobi[i][j].toPrecision(4)+"</td>";
222.                 }
223.                 html+="</tr>";
224.             }
225.             html+="</tbody></table>";
226.             $('#tablasDer').append(html);
227.             return jacobi;
228.         }
229. 
230.         function __getNewtonRaphson(){
231.             var jacobi    = new Matrix(__getJacobi()),
232.                 invJacobi = jacobi.clone().inverse(),
233.                 x0        = new Matrix(new Array(datos.xi)),
234.                 f0        = new Matrix(new Array(datos.fi)),
235.                 cont      = 0,
236.                 xr        = new Matrix([[0,0,0]]);
237. 
238.             var html = "<h4>Matriz Inversa Jacobi</h4><table class='table table-
    striped table-hover table-bordered'><tbody>";
239.             for(var i=0;i<invJacobi.rows;i++){
240.                 html+= "<tr>";
241.                 for(var j=0;j<invJacobi.cols;j++){
242.                     html+= "<td>"+invJacobi[i][j].toPrecision(4)+"</td>";
243.                 }
244.                 html+="</tr>";
245.             }
246.             html+="</tbody></table>";
247.             $('#tablasDer').append(html);
248. 
249.             while(cont<10){
250.                 cont++;
251.                 xr = x0.subtract(f0.multiply(invJacobi));
252.                 if(compare(xr,x0)){
253.                     break;
254.                 }
255.                 x0 = xr.clone();
256.             }
```

```
257.           var html = "<h4>Resultado</h4><table class='table table-striped table-
    hover table-bordered'><tbody>";
258.           for(var i=0;i<xr.cols;i++){
259.               html+= "<tr><td>"+xr[0][i].toPrecision(4)+"</td></tr>";
260.           }
261.           html+="</tbody></table>";
262.           $('#tablasDer').append(html);
263.           return xr;
264.       }
265.
266.       /* Inicializar Eventos */
267.       initialEvents();
```