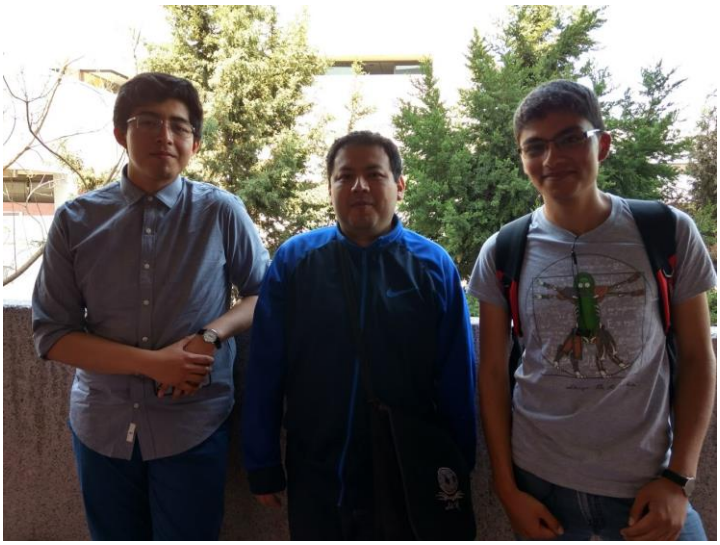
Instituto Politécnico Nacional

Escuela Superior de Cómputo



CONVEX HULL

Proyecto Final

Análisis de Algoritmos

M. en C. Edgardo Adrián Franco Martínez

Grupo: 3CM3

Fecha: 18/Junio/2018

Equipo: Git Gud (Equipo Arbol)

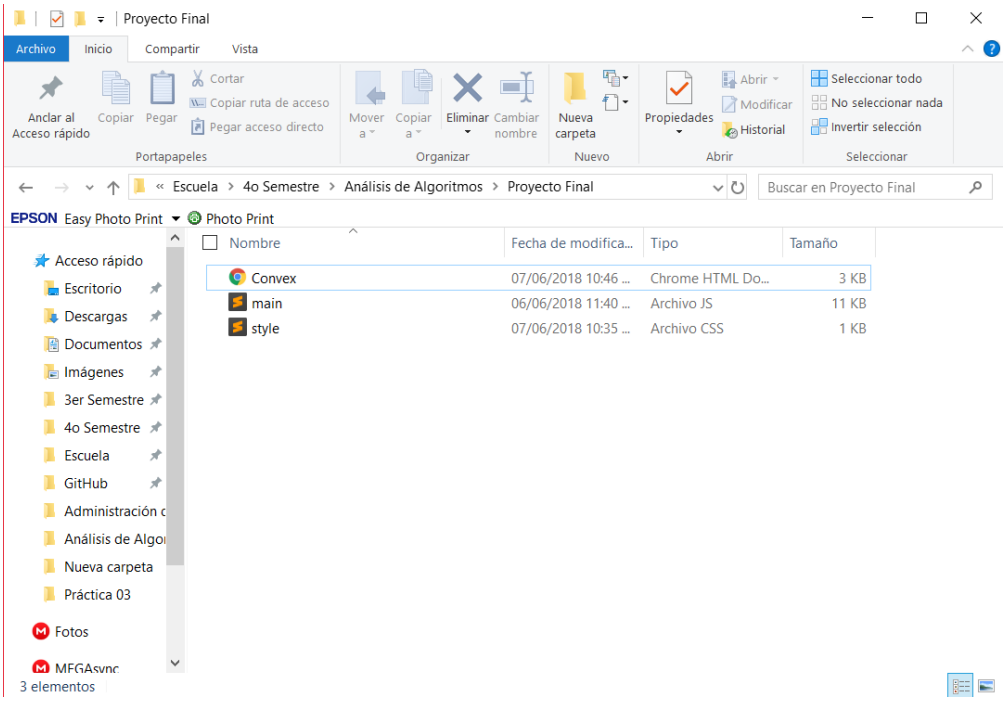
- Calva Hernández José Manuel 2017630201
- Meza Madrid Raúl Damián 2017631051
- Montaño Ayala Alan Israel 2016630260

Índice

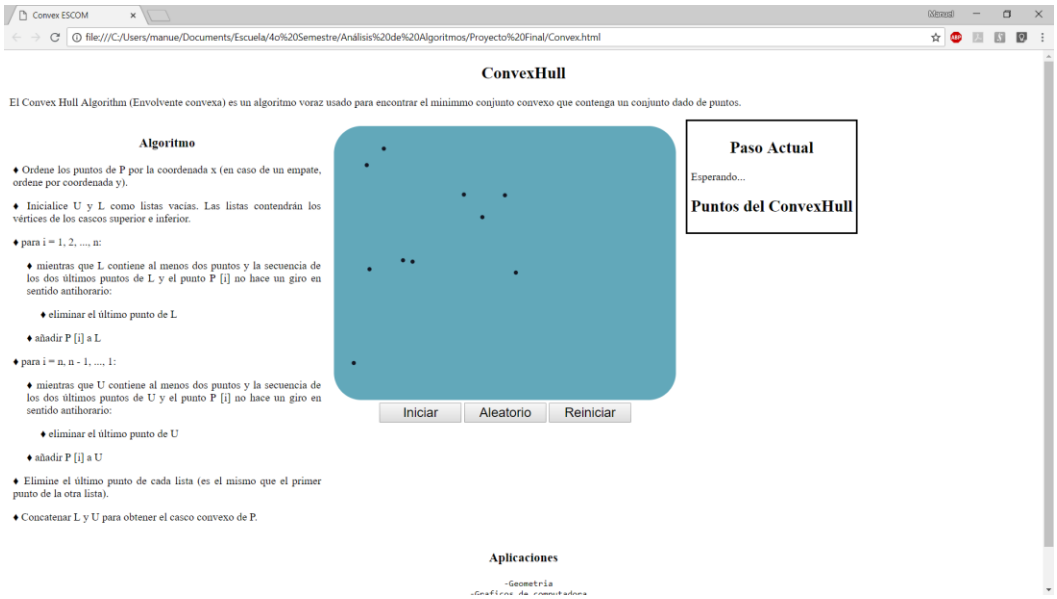
Ejecución

En Windows, ejecutar las siguientes instrucciones:

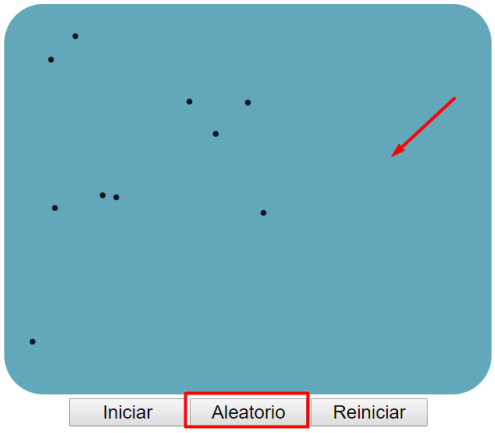
1. Colocar los archivos en una misma carpeta:



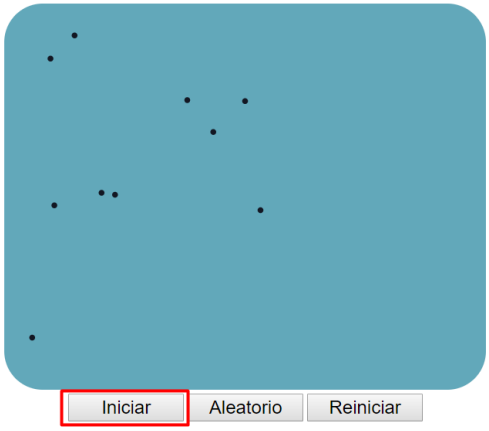
2. Hacer doble click en el archivo Convex.html, esto nos abrirá la simulación en nuestro navegador predeterminado:



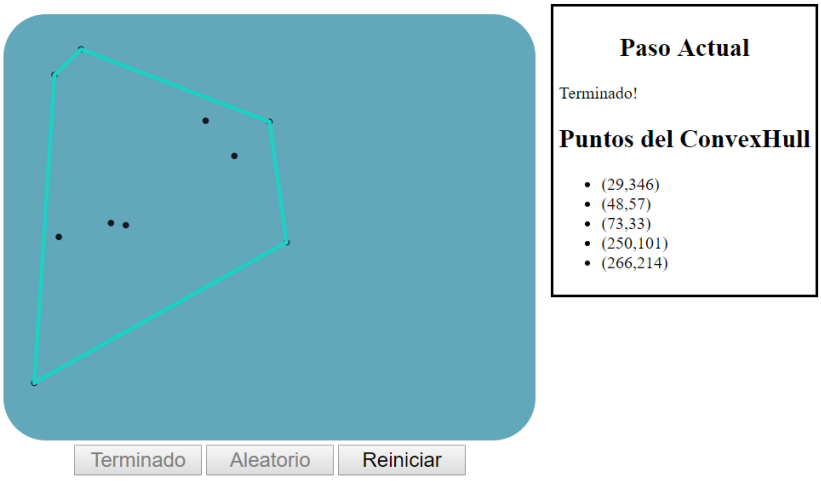
3. Basta con añadir puntos manualmente o haciendo click en “Aleatorio”:



4. Ejecutaremos la simulación dando click en “Iniciar”:



5. Verificar los resultados mediante la simulación del algoritmo:



Convex (main)

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <link rel="stylesheet" href="style.css">

    <title>Convex ESCOM</title>
</head>

<body>
<div id="body">
    <h2>ConvexHull</h2>
    <div id="grid">
      <p>
El Convex Hull Algorithm (Envolvente convexa) es un
algoritmo voraz usado para encontrar el minimmo conjunto
convexo que contenga un conjunto dado de puntos.
      </p>
    </div>
    <div id="grid">
      <div id="leftPanel">


        <h3>Algoritmo</h3>
          <p id="p1">&#9830 Ordene los puntos de P por la coordenada x (en caso de un empate, ordene por coordenada y).</p>
          <p id="p2">&#9830 Inicialice U y L como listas vacías. Las listas contendrán los vértices de los cascos superior e inferior.</p>
          <p id="p3">&#9830 para i = 1, 2, ..., n:</p>
          <p id="p4">&#9830 mientras que L contiene al menos dos puntos y la secuencia de los dos últimos puntos de L y el punto P [i] no
          hace un giro en sentido antihorario:</p>
          <p id="p5">&#9830 eliminar el último punto de L</p>
          <p id="p6">&#9830 añadir P [i] a L</p>
          <p id="p7">&#9830 para i = n, n - 1, ..., 1:</p>
          <p id="p8">&#9830 mientras que U contiene al menos dos puntos y la secuencia de los dos últimos puntos de U y el punto P [i] no
          hace un giro en sentido antihorario:</p>
          <p id="p9">&#9830 eliminar el último punto de U</p>
          <p id="p10">&#9830 añadir P [i] a U</p>
          <p id="p11">&#9830 Elimine el último punto de cada lista (es el mismo que el primer punto de la otra lista).</p>
      <p id="p12">&#9830 Concatenar L y U para obtener el casco convexo de P.</p>
    </div>
    <div id="canvas">
      <div id="botones">
          <button id="runbtn" type="button">Iniciar</button>
          <button id="randombtn" type="button">Aleatorio</button>
          <button id="clearbtn" type="button">Reiniciar</button>
      </div>
  </div>
    <div id="rightPanel">
      <p>
        <h2>Paso Actual</h2>
        <p id="currentState"></p>
        <h2>Puntos del ConvexHull</h2>
        <ul id="PointsInCH">

        </ul>
      </p>
    </div>
    <div id="grid">
      <h3>Aplicaciones</h3>
      <pre style="text-align: center;">
  -Geometria
  -Graficos de computadora
  -Planeo de rutas para robot
  -etc.
        </pre>
    </div>
    <div id="grid">
    </div>

</div>
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/raphael/2.2.7/raphael.js"></script>
    <script src="main.js"></script>
</body>
```

Main.js

```
function Point(x, y, g) {
    this.x = x || 0;
    this.y = y || 0;
    this.graphic = g;
    this.main_line = null;
};
Point.prototype.x = null;
Point.prototype.y = null;
Point.prototype.graphic = null;
Point.prototype.main_line = null;
window.onload = function() {
    Raphael.fn.line = function(startX, startY, endX, endY) {
        return this.path('M' + startX + ' ' + startY + ' L' + endX + ' ' + endY);
    };
    var paper = Raphael("canvas", 500, 400);
    var clearbtn = $('#clearbtn');
    var randombtn = $('#randombtn');
    var runbtn = $('#runbtn');
    var points = [];
    var canvas = null;
    var locked = false;
    var convexH = null;
    var auxtimer = null;

    function lineAnim(p, q) {
        line = paper.line(p.x, p.y, p.x, p.y).attr({
            'stroke-linecap': 'round',
            'stroke-linejoin': 'round',
            'stroke': '#23cec5'
        });
        p.main_line = line.animate({
            'stroke-width': '4',
            'path': 'M' + p.x + ' ' + p.y + ' L' + q.x + ' ' + q.y
        }, 100);
    }

    function updateColorAnim(p) {
        c = paper.circle(p.x, p.y, 1).animate({
            r: 10,
            fill: '#131723',
            "stroke-width": 0
        }, 200);
    }

    function deleteLineAnim(p) {
        p.main_line.animate({
            'stroke': 'rgba(255, 0, 0, 0.69)',
        }, (4000) / (points.length) * 2);
        p.main_line.animate({ //'stroke': 'rgba(255, 0, 0, 0.69)',
            'stroke-width': '0'
        }, (1000));
    }

    function addPointToList(p) {
        var node = document.createElement("li");
        var textnode = document.createTextNode("(" + p.x + "," + p.y + ")");
        node.appendChild(textnode);
        document.getElementById("PointsInCH").appendChild(node);
    }

    function removePointFromList() {
```

```
        $('li', ul).last().remove()
    }
    var auxF = {
        lineAnim: lineAnim,
        deleteLineAnim: deleteLineAnim,
        addPointToList: addPointToList,
    };

    function getMousePos(e) {
        var totalOffsetX = 0;
        var totalOffsetY = 0;
        var canvasX = 0;
        var canvasY = 0;
        var currentElement = document.getElementById('canvas');
        do {
            totalOffsetX += currentElement.offsetLeft - currentElement.scrollLeft;
            totalOffsetY += currentElement.offsetTop - currentElement.scrollTop;
        } while (currentElement = currentElement.offsetParent);
        canvasX = e.pageX - totalOffsetX - document.body.scrollLeft;
        canvasY = e.pageY - totalOffsetY - document.body.scrollTop;
        return new Point(canvasX, canvasY, null, null);
    }

    function addPointAnim(p) {
        if (locked) {
            return;
        }
        c = paper.circle(p.x, p.y, 1).animate({
            r: 3,
            fill: '#131723',
            "stroke-width": 0
        }, 200);
        p.graphic = c;
        points.push(p);
    }

    function clear() {
        paper.clear();
                canvas = paper.rect(0, 0, 500, 400, 40).attr({
                    fill: '#62a8ba',
                    stroke: "none"
                });
                points = [];
                unlock();
                running = false;
                convexH = null;
                runbtn.text('Iniciar');
                runbtn.attr('disabled', false);
                $("#PointsInCH").empty();
                document.getElementById("currentState").innerHTML = "Esperando...";
                canvas.mouseup(function(e) {
                    p = getMousePos(e);
                    addPointAnim(p);
                });
            }

            function lock() {
                locked = true;
                randombtn.attr('disabled', true);
                if (convexH == null) {
                    convexH = new CHAlgorith(points, auxF);
                }
            }

```

```
126.            function unlock() {
127.                locked = false;
128.                randombtn.attr('disabled', false);
129.            }
130.
131.            function StartPause() {
132.                if (running) {
133.                    running = false;
134.                    window.clearInterval(auxtimer);
135.                    runbtn.text('Continuar');
136.                    runbtn.attr('disabled', false);
137.                    clearbtn.attr('disabled', false);
138.                } else {
139.                    running = true;
140.                    lock();
141.                    if (convexH == null) {
142.                        convexH = new CHAlgorith(points, auxF);
143.                    }
144.                    runbtn.text('Pausa');
145.                    clearbtn.attr('disabled', true);
146.                    auxtimer = window.setInterval(function() {
147.                        r = convexH.iterate();
148.                        if (!r) {
149.                            window.clearInterval(auxtimer);
150.                            runbtn.text('Terminado');
151.                            runbtn.attr('disabled', true);
152.                            clearbtn.attr('disabled', false);
153.                        }
154.                    }, 800);
155.                }
156.            }
157.            clearbtn.click(function() {
158.                clear();
159.            });
160.            runbtn.click(function() {
161.                StartPause();
162.            });
163.            randombtn.click(function() {
164.                if (!locked) {
165.                    for (var i = 0; i < 10; i++) {
166.                        var per = 0.9;
167.                        var x = Math.floor(Math.random() * paper.width);
168.                        var y = Math.floor(Math.random() * paper.height);
169.                        x = Math.floor(x * per + ((1.0 - per) / 2.0) * paper.width);
170.                        y = Math.floor(y * per + ((1.0 - per) / 2.0) * paper.height);
171.                        addPointAnim(new Point(x, y, null));
172.                    }
173.                }
174.            });
175.            clearbtn.click();
176.            randombtn.click();
177.        };
178.
179.        function CHAlgorith(points, auxF) {
180.            this.States = {
181.                SORTING: 's',
182.                MOVING: 'm',
183.                VERIFYING: 'v',
184.                DONE: 'd'
185.            };
186.            this.points = points;
187.            this.auxF = auxF;
188.            this.toTheRight = true;
189.            this.state = this.States.SORTING;
```

```
190.            this.i = 0;
191.            this.ConvexHull = [];
192.            this.p = this.q = this.r = null;
193.            this.convexSort = function() {
194.                this.points = this.points.sort(function(a, b) {
195.                    if (a.x - b.x == 0) {
196.                        return b.y - a.y;
197.                    }
198.                    return a.x - b.x;
199.                });
200.            }
201.            this.rightSide = function(p, q, r) {
202.                var determinante = p.x * q.y - p.x * r.y - p.y * q.x + p.y * r.x + q.x * r.y - q.y * r.x;
203.                return determinante >= 0;
204.            }
205.            this.addToHull = function(index) {
206.                var p = this.points[index];
207.                this.ConvexHull.push(p);
208.                this.auxF.addPointToList(p);
209.                var n = this.ConvexHull.length;
210.                if (this.auxF) {
211.                    if (n > 1) {
212.                        this.auxF.lineAnim(this.ConvexHull[n - 2], this.ConvexHull[n - 1]);
213.                    }
214.                }
215.            }
216.            this.removeFromHull = function() {
217.                var n = this.ConvexHull.length;
218.                var p = this.ConvexHull[n - 2];
219.                var paux = this.ConvexHull[n - 1];
220.                $('li', PointsInCH).last().remove() $('li', PointsInCH).last().remove() this.auxF.addPointToL
    ist(paux);
221.                this.ConvexHull.splice(n - 2, 1);
222.                n--;
223.                if (this.auxF) {
224.                    this.auxF.deleteLineAnim(p);
225.                    this.auxF.deleteLineAnim(this.ConvexHull[n - 2]);
226.                    this.auxF.lineAnim(this.ConvexHull[n - 2], this.ConvexHull[n - 1]);
227.                }
228.            }
229.            this.iterate = function() {
230.                switch (this.state) {
231.                    case this.States.DONE:
232.                        return false;
233.                    case this.States.SORTING:
234.                        document.getElementById("currentState").innerHTML = "Ordenando! puntos...";
235.                        this.convexSort();
236.                        this.state = this.States.MOVING;
237.                        return this.iterate();
238.                    case this.States.MOVING:
239.                        document.getElementById("currentState").innerHTML = "Buscando siguiente punto";
240.                        if (this.i == 0 && this.toTheRight) {
241.                            if (this.points.length > 0) {
242.                                document.getElementById("currentState").innerHTML = "Agregando primer punto:
    (" + points[0].x + "," + points[0].y + ")";
243.                                this.addToHull(0);
244.                                if (this.points.length > 1) {
245.                                    document.getElementById("currentState").innerHTML = "Agregando : (" + poi
    nts[1].x + "," + points[1].y + ")";
246.                                    this.addToHull(1);
247.                                }
248.                            }
249.                            this.i = 1;
250.                            if (this.points.length <= 2) {
```

```
                                document.getElementById("currentState").innerHTML = "No hay suficientes puntos";
                                this.state = this.States.DONE;
                            }
                            return this.state != this.States.DONE;
                        }
                        if (this.toTheRight) {
                            this.i++;
                            if (this.i >= this.points.length) {
                                this.i = this.points.length - 2;
                                this.toTheRight = false;
                                this.addToHull(this.i);
                                document.getElementById("currentState").innerHTML = "Agregando primer punto: (" + points[0].x + "," + points[0].y + ")";
                            }
                        }
                        if (!this.toTheRight) {
                            this.i--;
                            if (this.i == -1) {
                                this.ConvexHull.splice(this.ConvexHull.length - 1);
                                document.getElementById("currentState").innerHTML = "Terminado!";
                                this.state = this.States.DONE;
                                $('li', PointsInCH).last().remove() return false;
                            }
                        }
                        this.addToHull(this.i);
                        document.getElementById("currentState").innerHTML = "Agregando  punto: (" + points[this.i].x + "," + points[this.i].y + ")";
                        this.state = this.States.VERIFYING;
                        this.r = this.ConvexHull[this.ConvexHull.length - 1];
                        return true;
                    case this.States.VERIFYING:
                        document.getElementById("currentState").innerHTML = "Verificando...";
                        n = this.ConvexHull.length;
                        if (n <= 2) {
                            this.state = this.States.MOVING;
                            return this.iterate();
                        }
                        this.p = this.ConvexHull[n - 3];
                        this.q = this.ConvexHull[n - 2];
                        if (!this.rightSide(this.p, this.q, this.r)) {
                            this.removeFromHull();
                            return true;
                        }
                        this.state = this.States.MOVING;
                        return this.iterate();
                }
                return true;
            }
        }
```

Style.css

```
body{
  background-color: #fff;
  }
  #body {
    }
    h3,h2{
      text-align: center;
    }
    #leftPanel{
      max-width: 30%;
      display: inline-block;
      vertical-align: top;
      padding: 5px;
      margin-left: 0px;
    }
    p{
      text-align: justify;
    }

    #canvas {
        height: 400px;
        display: inline-block;
        width: 500px;
        padding: 10px;
        padding-bottom: 5px;
        cursor: crosshair;
        margin-left: auto;
        margin-right: auto;
    }

    #rightPanel{
      background-color: #fff;
      border-color: #000;
      max-width: 30%;
      display: inline-block;
      vertical-align: top;
      padding: 5px;
      border: solid;
    }
```

```
    #body button {
        font-size: 1.2em;
        width: 120px;
    }
    #botones{
      text-align: center;
    }
    #p4,#p8,#p6,#p10{
      margin-left: 20px;
    }
    #p5,#p9{
      margin-left: 40px;
    }
```