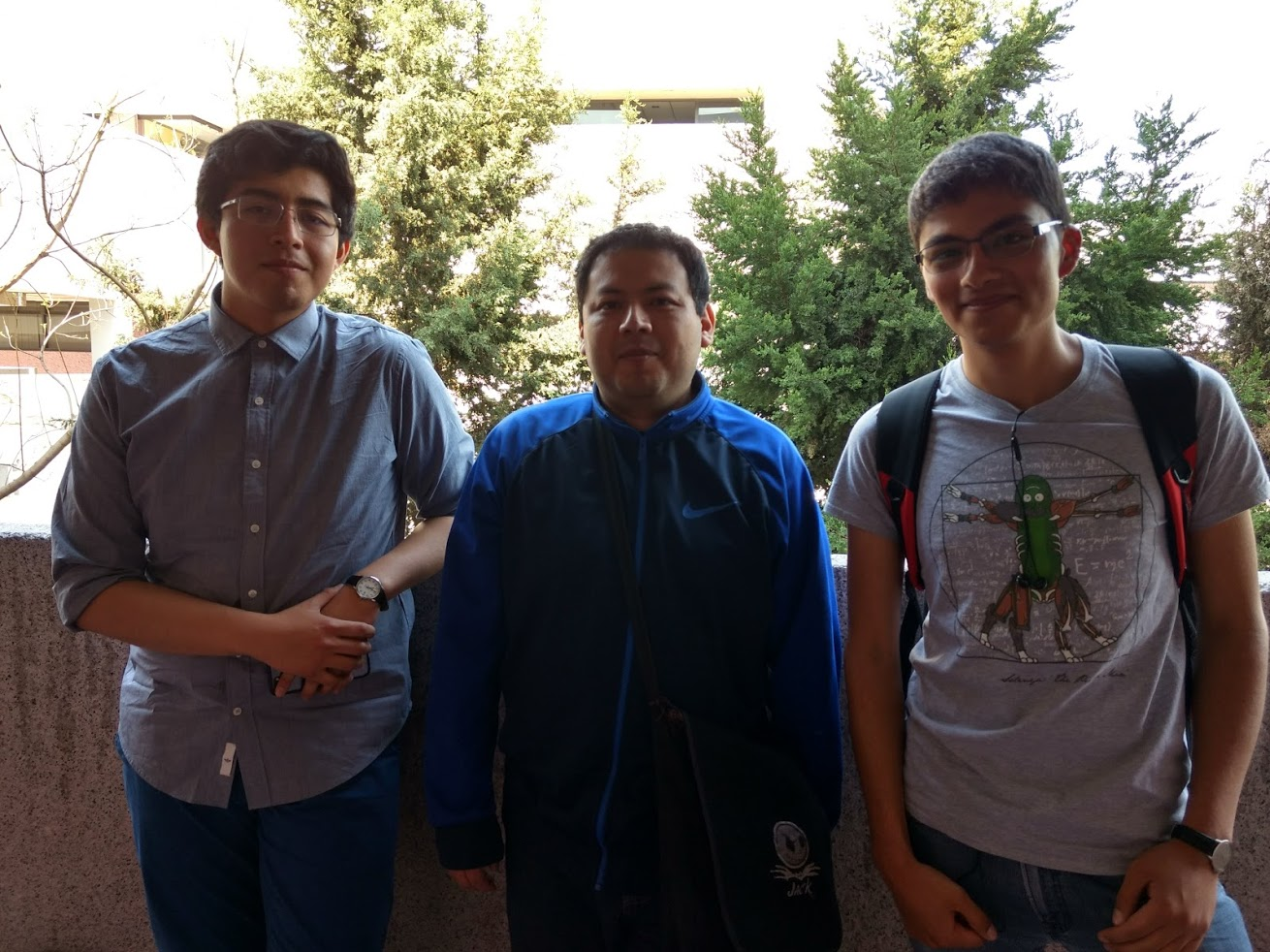
Equipo: Git Gud (Equipo Arbol)

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



Análisis de Algoritmos

M. en C. Edgardo Adrián Franco Martínez  
Grupo: 3CM3  
Fecha: 18/Junio/2018

Convex hull

Proyecto Final

Instituto Politécnico Nacional

Escuela Superior de Cómputo

Índice

[Ejecución 2](#_Toc517116877)

[Convex (main) 4](#_Toc517116878)

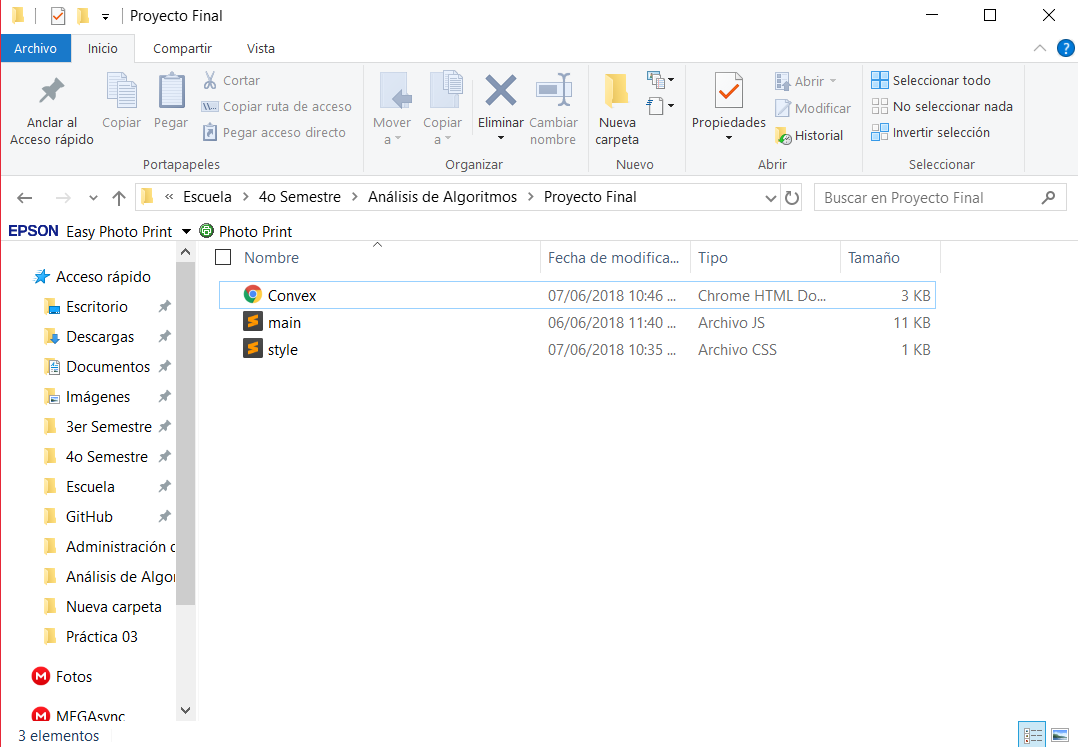
[Main.js 5](#_Toc517116879)

[Style.css 10](#_Toc517116880)

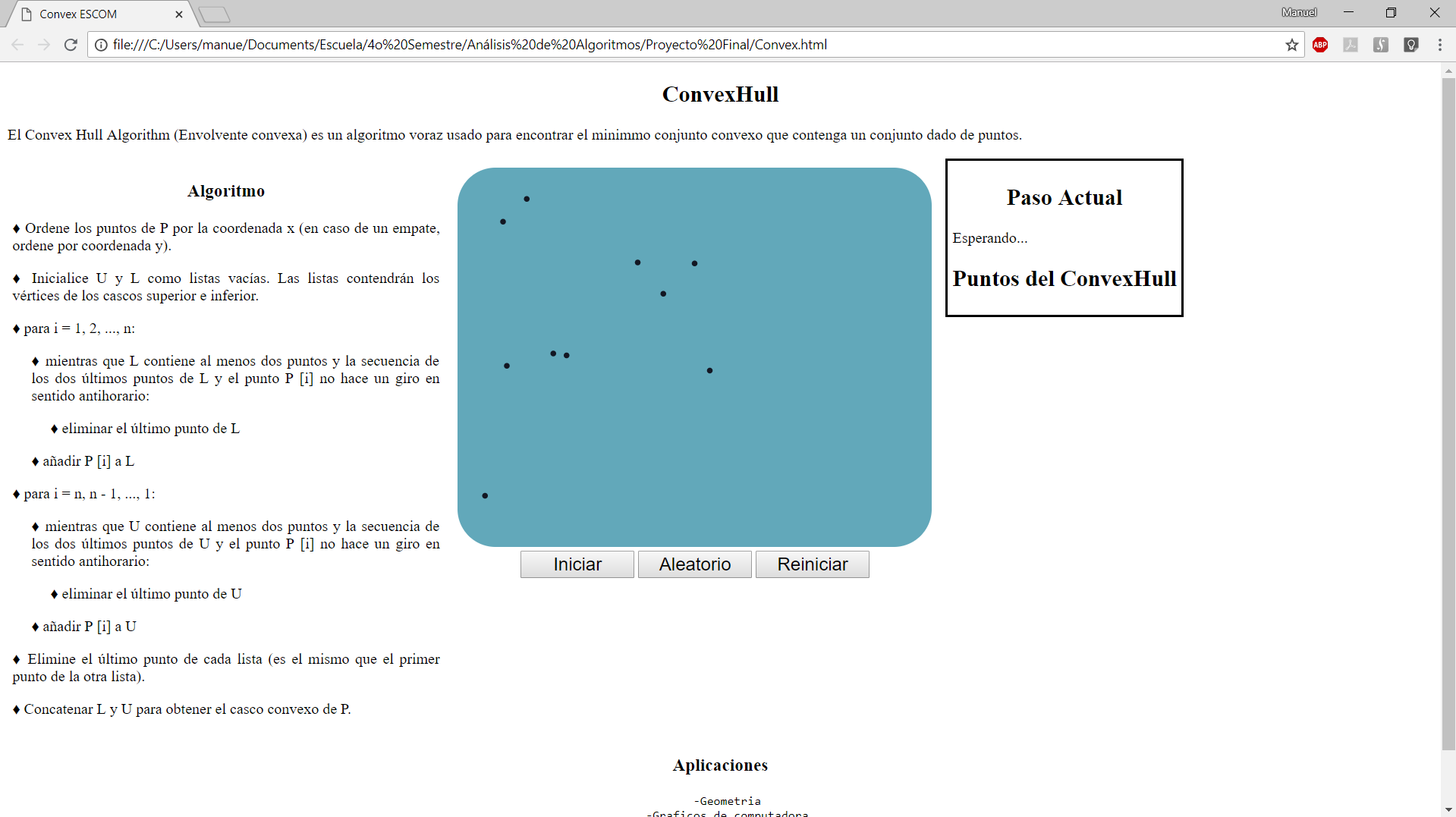
# Ejecución

En Windows, ejecutar las siguientes instrucciones:

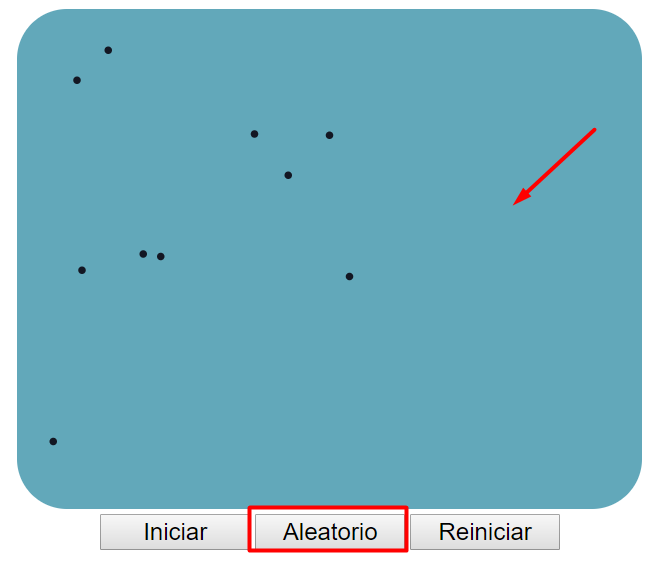
1. Colocar los archivos en una misma carpeta:



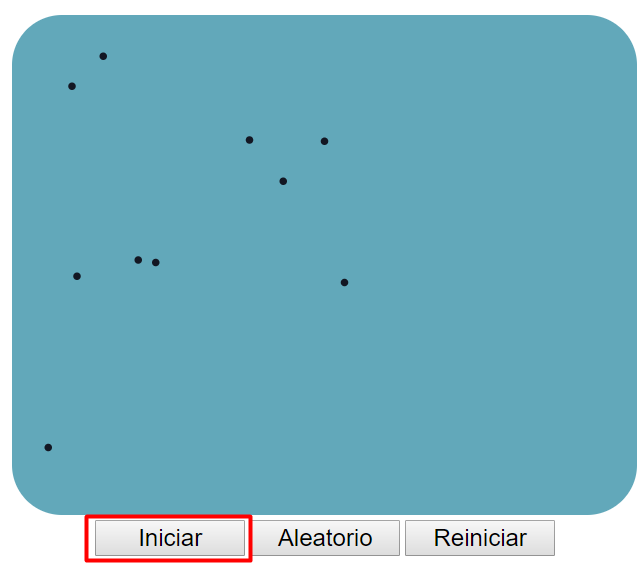
1. Hacer doble click en el archivo Convex.html, esto nos abrirá la simulación en nuestro navegador predeterminado:



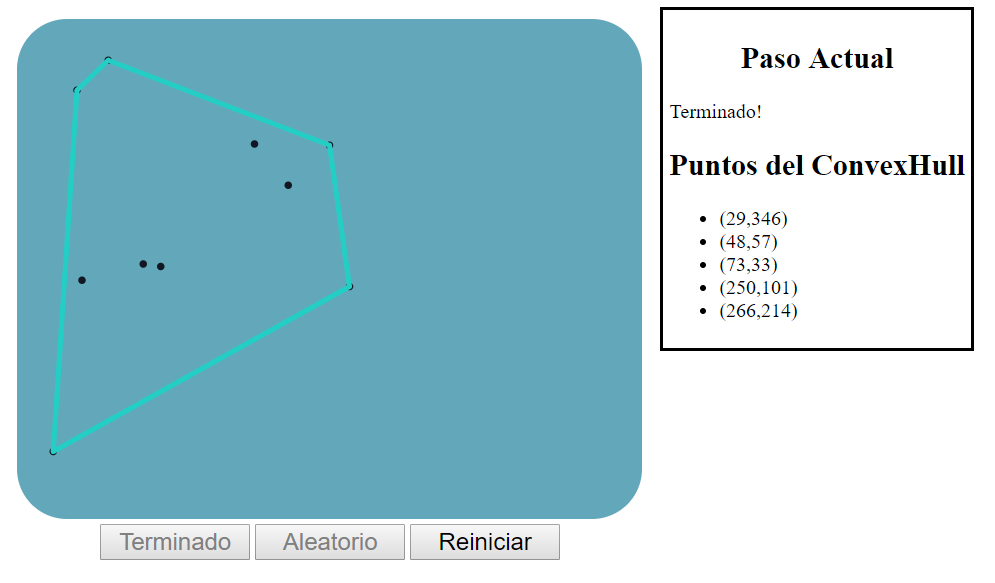
1. Basta con añadir puntos manualmente o haciendo click en “Aleatorio”:



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



1. Verificar los resultados mediante la simulación del algoritmo:



# Convex (main)





# Main.js

1. function Point(x, y, g) {
2. this.x = x || 0;
3. this.y = y || 0;
4. this.graphic = g;
5. this.main\_line = null;
6. };
7. Point.prototype.x = null;
8. Point.prototype.y = null;
9. Point.prototype.graphic = null;
10. Point.prototype.main\_line = null;
11. window.onload = function() {
12. Raphael.fn.line = function(startX, startY, endX, endY) {
13. return this.path('M' + startX + ' ' + startY + ' L' + endX + ' ' + endY);
14. };
15. var paper = Raphael("canvas", 500, 400);
16. var clearbtn = $('#clearbtn');
17. var randombtn = $('#randombtn');
18. var runbtn = $('#runbtn');
19. var points = [];
20. var canvas = null;
21. var locked = false;
22. var convexH = null;
23. var auxtimer = null;
25. function lineAnim(p, q) {
26. line = paper.line(p.x, p.y, p.x, p.y).attr({
27. 'stroke-linecap': 'round',
28. 'stroke-linejoin': 'round',
29. 'stroke': '#23cec5'
30. });
31. p.main\_line = line.animate({
32. 'stroke-width': '4',
33. 'path': 'M' + p.x + ' ' + p.y + ' L' + q.x + ' ' + q.y
34. }, 100);
35. }
37. function updateColorAnim(p) {
38. c = paper.circle(p.x, p.y, 1).animate({
39. r: 10,
40. fill: '#131723',
41. "stroke-width": 0
42. }, 200);
43. }
45. function deleteLineAnim(p) {
46. p.main\_line.animate({
47. 'stroke': 'rgba(255, 0, 0, 0.69)',
48. }, (4000) / (points.length) \* 2);
49. p.main\_line.animate({ //'stroke': 'rgba(255, 0, 0, 0.69)',
50. 'stroke-width': '0'
51. }, (1000));
52. }
54. function addPointToList(p) {
55. var node = document.createElement("li");
56. var textnode = document.createTextNode("(" + p.x + "," + p.y + ")");
57. node.appendChild(textnode);
58. document.getElementById("PointsInCH").appendChild(node);
59. }
61. function removePointFromList() {
62. $('li', ul).last().remove()
63. }
64. var auxF = {
65. lineAnim: lineAnim,
66. deleteLineAnim: deleteLineAnim,
67. addPointToList: addPointToList,
68. };
70. function getMousePos(e) {
71. var totalOffsetX = 0;
72. var totalOffsetY = 0;
73. var canvasX = 0;
74. var canvasY = 0;
75. var currentElement = document.getElementById('canvas');
76. do {
77. totalOffsetX += currentElement.offsetLeft - currentElement.scrollLeft;
78. totalOffsetY += currentElement.offsetTop - currentElement.scrollTop;
79. } while (currentElement = currentElement.offsetParent);
80. canvasX = e.pageX - totalOffsetX - document.body.scrollLeft;
81. canvasY = e.pageY - totalOffsetY - document.body.scrollTop;
82. return new Point(canvasX, canvasY, null, null);
83. }
85. function addPointAnim(p) {
86. if (locked) {
87. return;
88. }
89. c = paper.circle(p.x, p.y, 1).animate({
90. r: 3,
91. fill: '#131723',
92. "stroke-width": 0
93. }, 200);
94. p.graphic = c;
95. points.push(p);
96. }
98. function clear() {
99. paper.clear();
100. canvas = paper.rect(0, 0, 500, 400, 40).attr({
101. fill: '#62a8ba',
102. stroke: "none"
103. });
104. points = [];
105. unlock();
106. running = false;
107. convexH = null;
108. runbtn.text('Iniciar');
109. runbtn.attr('disabled', false);
110. $("#PointsInCH").empty();
111. document.getElementById("currentState").innerHTML = "Esperando...";
112. canvas.mouseup(function(e) {
113. p = getMousePos(e);
114. addPointAnim(p);
115. });
116. }
118. function lock() {
119. locked = true;
120. randombtn.attr('disabled', true);
121. if (convexH == null) {
122. convexH = new CHAlgorith(points, auxF);
123. }
124. }
126. function unlock() {
127. locked = false;
128. randombtn.attr('disabled', false);
129. }
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. };
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.addPointToList(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 : (" + points[1].x + "," + points[1].y + ")";
246. this.addToHull(1);
247. }
248. }
249. this.i = 1;
250. if (this.points.length <= 2) {
251. document.getElementById("currentState").innerHTML = "No hay suficientes puntos";
252. this.state = this.States.DONE;
253. }
254. return this.state != this.States.DONE;
255. }
256. if (this.toTheRight) {
257. this.i++;
258. if (this.i >= this.points.length) {
259. this.i = this.points.length - 2;
260. this.toTheRight = false;
261. this.addToHull(this.i);
262. document.getElementById("currentState").innerHTML = "Agregando primer punto: (" + points[0].x + "," + points[0].y + ")";
263. }
264. }
265. if (!this.toTheRight) {
266. this.i--;
267. if (this.i == -1) {
268. this.ConvexHull.splice(this.ConvexHull.length - 1);
269. document.getElementById("currentState").innerHTML = "Terminado!";
270. this.state = this.States.DONE;
271. $('li', PointsInCH).last().remove() return false;
272. }
273. }
274. this.addToHull(this.i);
275. document.getElementById("currentState").innerHTML = "Agregando  punto: (" + points[this.i].x + "," + points[this.i].y + ")";
276. this.state = this.States.VERIFYING;
277. this.r = this.ConvexHull[this.ConvexHull.length - 1];
278. return true;
279. case this.States.VERIFYING:
280. document.getElementById("currentState").innerHTML = "Verificando...";
281. n = this.ConvexHull.length;
282. if (n <= 2) {
283. this.state = this.States.MOVING;
284. return this.iterate();
285. }
286. this.p = this.ConvexHull[n - 3];
287. this.q = this.ConvexHull[n - 2];
288. if (!this.rightSide(this.p, this.q, this.r)) {
289. this.removeFromHull();
290. return true;
291. }
292. this.state = this.States.MOVING;
293. return this.iterate();
294. }
295. return true;
296. }
297. }

# Style.css

