

```

1  {% load static %}
2  <!DOCTYPE html>
3  <html>
4
5  <head>
6      <%- include('../partials/head'); %>
7
8      <script type="module" defer>
9          import { websocket } from "/js/WebSocket.js";
10
11          let updateQueue = new Queue();
12          let waitTime = 100;
13          let counter = {};
14
15          websocket.events = [
16              {
17                  name: 'connect',
18                  handler: function () {
19                      customAlert("Conectado", 'success');
20                  }
21              }, {
22                  name: 'disconnect',
23                  handler: function (reason) {
24                      customAlert("Desconectado!", 'error');
25                  }
26              }, {
27                  name: 'handshake',
28                  handler: function (data) { }
29              }, {
30                  name: 'update',
31                  handler: function (data) {
32                      // TODO recieve many node progresses
33                      updateQueue.enqueue(JSON.parse(data));
34                  }
35              }
36          ];
37
38
39          // Graphic functions
40          function updateCounter() {
41              let head, body;
42
43              $("#result-head").html("");
44              $("#result-body").html("");
45
46              for (let [key, val] of Object.entries(counter)) {
47                  head = document.createElement("th");
48                  $(head).prop("scope", "col");
49                  $(head).addClass("text-center");
50                  $(head).text(key);
51
52                  body = document.createElement("td");
53                  $(body).addClass("text-center");
54                  $(body).text(val);

```

```

55
56     $("#result-head").append(head);
57     $("#result-body").append(body);
58 }
59 }
60 function customAlert(text, mode = '', callback = () => {
61     $("#div-alert").fadeOut('fast');
62 }) {
63     $("#div-alert").fadeOut('fast', function () {
64
65         $("#div-alert").removeClass("alert-warning");
66         $("#div-alert").removeClass("alert-success");
67         $("#div-alert").removeClass("alert-danger");
68
69         switch (mode) {
70             case "success":
71                 $("#div-alert").addClass("alert-success");
72                 break;
73             case "error":
74                 $("#div-alert").addClass("alert-danger");
75                 break;
76             default:
77                 $("#div-alert").addClass("alert-info");
78                 break;
79         }
80
81         $("#div-alert").text(text);
82         $("#div-alert").fadeIn('slow', callback);
83     });
84 }
85 function drawRipple(x, y) {
86     let node = $(".ripple").clone();
87
88     node.addClass("animate");
89     node.css({
90         left: x,
91         top: y
92     });
93
94     $(".ripple").replaceWith(node);
95 };
96
97
98 // Node functions
99 function moveToNode(marker, node, time) {
100     return marker.animate({
101         width: node.outerWidth(),
102         height: node.outerHeight(),
103         left: node.offset().left,
104         top: node.offset().top,
105         opacity: '0.7'
106     }, time).promise();
107 }
108 async function animateNode(start, end, isEnd, time = waitTime, callback = () => {

```

```

109     let origin, dest;
110     let progress;
111     let marker = $("#progress-marker").clone().removeAttr("id").addClass(["step",
112
113     // Get nodes
114     origin = findNode(start);
115     dest = findNode(end);
116
117     // Add appropriate styling
118     marker.addClass(isEnd ? "bg-success" : "bg-danger");
119
120     // Move progress marker between nodes
121     await moveToNode(marker, origin, 0);
122     await moveToNode(marker, dest, time);
123     callback();
124 }
125 function findNode(text) {
126     return $("div").filter(function () {
127         return $(this).text() === text;
128     });
129 }
130
131
132 // Document ready
133 $(document).ready(function () {
134     $.get("/get/tree/", (data, status) => {
135         new Treant(JSON.parse(data), () => {
136             $(".node-end").addClass("bg-secondary");
137         });
138     });
139     websocket.connect();
140
141     $("#form-control").submit(function (ev) {
142         ev.preventDefault();
143
144         counter = {};
145         updateCounter();
146
147         $.get("start/", function () { });
148     });
149
150     // Node animation dequeuing
151     setInterval(function () {
152         if (updateQueue.getLength() == 0) {
153             return;
154         }
155
156         // Remove all previous markers
157         $(".step").remove();
158
159         // Get next animation steps
160         let steps = updateQueue.dequeue();
161         let currentChar;

```

```

163         let endNode;
164
165         // Process steps simultaneously
166         steps.forEach(step => {
167             animateNode(step.nodeStart, step.nodeEnd, step.nodeIsEnd);
168
169             // Count words
170             if (step.nodeIsEnd) {
171
172                 // If the counter is null, create counter
173                 if (counter[step.nodeStart] == null) {
174                     counter[step.nodeStart] = 0;
175                 }
176                 ++counter[step.nodeStart];
177                 updateCounter();
178
179                 // Add ripple effect to node
180                 endNode = findNode(step.nodeStart);
181                 drawRipple(endNode.offset().left, endNode.offset().top);
182             }
183
184             currentChar = step.currentChar;
185         });
186
187         // Add character to progress 'bar'
188         $("#span-progress").text($("#span-progress").text().substring(1) + currentCh
189     }, waitTime);
190     });
191     </script>
192 </head>
193
194 <body class="d-block">
195     <div id="div-alert-container" class="d-flex justify-content-center fixed-top">
196         <div id="div-alert" class="alert text-justify" style="display: none;"></div>
197     </div>
198     <div class="container">
199         <div class="row">
200             <div class="d-flex w-100 flex-column">
201                 <%- include('../partials/header'); %>
202
203                 <main role="main" class="inner cover text-center w-100">
204                     <h1 class="cover-heading">Grafo de Aut mata Webay</h1>
205                     <p>
206                         Grafo representando a un aut mata que detecta varias palabras.
207                     </p>
208                     <form id="form-control">
209                         <button type="submit" class="btn-primary p-2 rounded">Iniciar</button>
210                     </form>
211                     <h3>Progreso:<br><span id="span-progress">
212 </span></h3>
213                     <br>
214                     </main>
215                 </div>
216             </div>
217         </div>
218     </div>

```

```

216 </div>
217 <div class="d-block">
218   <div class="d-flex justify-content-center">
219     <div class="col-auto">
220       <table class="table table-responsive">
221         <thead>
222           <tr id="result-head"></tr>
223         </thead>
224         <tbody>
225           <tr id="result-body"></tr>
226         </tbody>
227       </table>
228     </div>
229   </div>
230   <div id="div-grafo-container" class="d-flex justify-content-center">
231     <div id="div-grafo"></div>
232   </div>
233 </div>
234
235 <!-- Ripple -->
236 <div id="ripple-container" class="position-absolute w-100 h-100 overflow-hidden fixe
237   <div class="ripple position-absolute overflow-hidden"></div>
238 </div>
239
240 <!-- Progress marker -->
241 <div id="progress-marker" class="position-absolute p-3"></div>
242 </body>
243
244 </html>

```
