```
1 <!DOCTYPE html>
2 <html>
3
4
   <head>
5
     <%- include('../partials/head'); %>
6
7
       <script>
8
         var animationTime = 1000;
9
         var animationQueue = [];
10
          var currentAnimation;
11
          var result;
12
13
          function movePiece(piece, elementBoard, time) {
14
           return piece.animate({
              left: elementBoard.offset().left + ((elementBoard.outerWidth
15
                 () - piece.outerWidth()) / 2),
              top: elementBoard.offset().top + ((elementBoard.outerHeight
16
                  () - piece.outerHeight()) / 2)
17
           }, time).promise();
          }
18
          async function transportPiece(idOrigin, idDestination, time =
19
             animationTime) {
20
            let piece = $("#divFicha").clone().removeAttr("id").addClass("
               step").removeClass("d-none").appendTo("body");
21
22
            let elementOrigin = findBoardElement("Q" + idOrigin);
23
            let elementDestination = findBoardElement("Q" + idDestination)
               ;
24
25
            await movePiece(piece, elementOrigin, 0);
26
           await movePiece(piece, elementDestination, time);
27
          }
28
29
30
          function findBoardElement(idBoard) {
31
            return $("td").filter(function () {
32
              return $(this).text() == idBoard;
33
           });
34
35
36
          $(document).ready(function () {
37
            $("#btnStart").click(function () {
38
              $.post("/procesar/cadena/", {
               input: $("#txtBinario").val()
39
40
              }, function (data, status) {
41
                data = JSON.parse(data);
42
43
                // Setting input text
44
                $("#txtBinario").val(data.binario);
45
46
                // Setting future result text
47
                result = data.result;
48
49
                console.log(data);
```

```
50
51
                // Queueing animations
52
                animationQueue = data.animations;
53
54
                if (data.valid) {
                  alert("Cadena v lida");
55
56
                }
57
              });
58
            });
59
60
            // Node animation dequeuing
61
            setInterval(function () {
62
              if (animationQueue.length == 0) {
                $("#titleResult").text(result);
63
64
                return;
65
66
67
              $(".step").remove();
68
69
              // Animating current
70
              currentAnimation = animationQueue.shift();
71
              $("#divProgress").html(`(${currentAnimation.nodeName})<br>
72
                Entrada: ${currentAnimation.input}<br>
73
                Pila: ${currentAnimation.stack}`
74
              );
75
76
            }, animationTime);
77
          });
78
        </script>
79
   </head>
80
81
   <body class="d-block">
82
      <div id="div-alert-container" class="d-flex justify-content-center</pre>
         fixed-top">
        <div id="div-alert" class="alert text-justify" style="display:</pre>
83
           none;"></div>
     </div>
84
     <div class="container">
85
86
        <div class="row">
87
          <div class="d-flex w-100 flex-column">
            <%- include('../partials/header'); %>
89
90
              <main role="main" class="inner cover text-center w-100">
91
                <h1 class="cover-heading">Automata de Pila</h1>
92
93
                  Implementaci n de un automata de pila con el lenguaje
                      libre de contexto \{0^n 1^n \mid n >= 1\}.
94
                95
                <br>
96
                <div class="row d-flex justify-content-center">
                  <div id="divControles" class="col-sm-6 text-center">
97
98
                    <div id="divInput" class="form-group">
99
                       <input type="text" class="form-control" id="</pre>
                          txtBinario" placeholder="Inserte la entrada">
```

```
100
                        <small class="form-text text-muted">Dejar esto
                           vac o para que se genere aleatoriamente.</small
                     </div>
101
102
                   </div>
103
                 </div>
104
                 <br>
105
                 <div id="divStart">
106
                   <button id="btnStart" type="button" class="btn</pre>
                       btn-primary">Iniciar</button>
107
                 </div>
108
               </main>
           </div>
109
         </div>
110
      </div>
111
112
      <br>
      <div class="d-block">
113
        <div id="divContent" class="d-block text-center">
114
115
116
           <h1>Proceso:</h1>
117
           <h2 id="titleResult"></h2>
118
           <br><br><br><br><br>></pr>
119
           <h3>
120
             <div id="divProgress"></span>
121
           </h3>
122
        </div>
123
      </div>
124
    </body>
125
126
    </html>
    @media only screen and (min-width: 768px) and (max-width: 991px) {
 2
 3
 4
      #main {
        width: 712px;
 5
 6
         padding: 100px 28px 120px;
 7
     /* .mono {
 10
        font-size: 90%;
 11
      } */
 12
 13
     .cssbtn a {
 14
       margin-top: 10px;
 15
        margin-bottom: 10px;
16
        width: 60px;
 17
        height: 60px;
 18
        font-size: 28px;
 19
        line-height: 62px;
```

1 class TelgramRequestHandler(object):

```
def handle(self):
           addr = self.client_address[0]  # Client IP-adress
telgram = self.request.recv(1024)  # Recieve telgram
3
4
           print "From: %s, Received: %s" % (addr, telgram)
5
6
           return
1
2 @media only screen and (min-width: 768px) and (max-width: 991px) {
3
4
     #main {
       width: 712px;
5
6
       padding: 100px 28px 120px;
7
    /* .mono {
9
10
      font-size: 90%;
11
    } */
12
13
    .cssbtn a {
14
      margin-top: 10px;
15
      margin-bottom: 10px;
16
       width: 60px;
17
       height: 60px;
18
       font-size: 28px;
19
       line-height: 62px;
20
```