


```

50
51         // Queueing animations
52         animationQueue = data.animations;
53
54         if (data.valid) {
55             alert("Cadena v lida");
56         }
57     });
58 });
59
60     // Node animation dequeuing
61     setInterval(function () {
62         if (animationQueue.length == 0) {
63             $("#titleResult").text(result);
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
83         fixed-top">
84         <div id="div-alert" class="alert text-justify" style="display:
85             none;"></div>
86     </div>
87     <div class="container">
88         <div class="row">
89         <div class="d-flex w-100 flex-column">
90             <%- include('../partials/header'); %>
91
92             <main role="main" class="inner cover text-center w-100">
93                 <h1 class="cover-heading">Automata de Pila</h1>
94                 <p>
95                     Implementaci n de un automata de pila con el lenguaje
96                     libre de contexto  $\{0^n 1^n \mid n \geq 1\}$ .
97                 </p>
98                 <br>
99                 <div class="row d-flex justify-content-center">
100                     <div id="divControles" class="col-sm-6 text-center">
101                         <div id="divInput" class="form-group">
102                             <input type="text" class="form-control" id="
103                                 txtBinario" placeholder="Inserte la entrada">

```

```

100         <small class="form-text text-muted">Dejar esto
           vac o para que se genere aleatoriamente.</small
           >
101     </div>
102 </div>
103 </div>
104 <br>
105 <div id="divStart">
106     <button id="btnStart" type="button" class="btn
           btn-primary">Iniciar</button>
107 </div>
108 </main>
109 </div>
110 </div>
111 </div>
112 <br>
113 <div class="d-block">
114     <div id="divContent" class="d-block text-center">
115
116         <h1>Proceso:</h1>
117         <h2 id="titleResult"></h2>
118         <br><br><br>
119         <h3>
120             <div id="divProgress"></span>
121         </h3>
122     </div>
123 </div>
124 </body>
125
126 </html>

```

```

1
2 @media only screen and (min-width: 768px) and (max-width: 991px) {
3
4     #main {
5         width: 712px;
6         padding: 100px 28px 120px;
7     }
8
9     /* .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;
20     }

```

```

1 class TelegramRequestHandler(object):

```

```

2     def handle(self):
3         addr = self.client_address[0]           # Client IP-address
4         telgram = self.request.recv(1024)       # Recieve telgram
5         print "From: %s, Received: %s" % (addr, telgram)
6         return

```

```

1
2 @media only screen and (min-width: 768px) and (max-width: 991px) {
3
4     #main {
5         width: 712px;
6         padding: 100px 28px 120px;
7     }
8
9     /* .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;
20    }

```
