

```

1 <template>
2   <div class="audio-page mb-5">
3     <div class="shadow-sm mb-5 p-5 bg-white rounded
4       ">
5       <header class="text-center border-bottom
6         border-warning mb-4">
7         <h1 class="text-dark">MediaStream Recording
8         API</h1>
9         <h3>aka Media Recorder API</h3>
10        </header>
11        <p class="lead">
12          La API de MediaStream Recording nos permite
13          capturar datos (de audio y/o video) que son
14          generados por un objeto de tipo MediaStream
15          o HTMLMediaElement para analizarlos,
16          procesarlos o guardarlos.
17        </p>
18      </div>
19
20      <div class="shadow-sm p-5 bg-white rounded">
21        <b-button variant="primary" class="mr-2" @
22          click="start" :disabled="startDisabled">Start</b-
23          button>
24        <b-button variant="secondary" @click="stop" :
25          disabled="stopDisabled">Stop</b-button>
26
27        <hr>
28
29        <div v-if="hasAudios">
30          <ul class="list-unstyled audio-list">
31            <li v-for="(item, idx) in audioList" :key
32              ="idx" class="audio-item mb-3">
33              <div class="d-flex justify-content-
34                between">
35                <audio :controls="true" :loop="false"
36                  :src="item.src" class="mr-2 w-100"></audio>
37                <div class="d-flex align-items-center
38                  ">
39                  <b-button variant="danger" size="sm

```

```
26 " @click="removeItem(idx)"> Eliminar </b-button>
27     </div>
28 </div>
29 </li>
30 </ul>
31 </div>
32
33 </div>
34 </div>
35 </template>
36
37 <script>
38 export default {
39   name: 'Audio',
40   data () {
41     return {
42       supported: undefined,
43       mediaRecorder: undefined,
44       chunks: [],
45       audioList: [],
46       isRecording: false
47     }
48   },
49   computed: {
50     startDisabled () {
51       return this.isRecording === true
52     },
53     stopDisabled () {
54       return this.isRecording === false
55     },
56     hasAudios () {
57       return this.audioList.length > 0
58     }
59   },
60   methods: {
61     start () {
62       this.mediaRecorder.start()
63       this.isRecording = true
64       console.log('State: ', this.mediaRecorder.
```

```

64 state)
65     console.log('recorder started')
66 },
67 stop () {
68     this.mediaRecorder.stop()
69     this.isRecording = false
70     console.log('State: ', this.mediaRecorder.
state)
71 },
72 removeItem (idx) {
73     this.audioList.splice(idx, 1)
74 }
75 },
76 created () {
77     // Comprobar si el navegador soporta esta '
feature'
78     if (navigator.mediaDevices && navigator.
mediaDevices.getUserMedia) {
79         this.supported = true
80
81         // Request permission for audio only
82         navigator.mediaDevices.getUserMedia({ audio
: true })
83         .then((stream) => {
84             // La interfaz MediaStream representa
una secuencia de contenido multimedia.
85             // Una transmisión consta de varias
pistas, como pistas de video o audio.
86             console.log(stream)
87
88             // Crea un objeto MediaRecorder dado un
objeto MediaStream
89             this.mediaRecorder = new MediaRecorder(
stream)
90
91             // Manejar los datos disponibles. Los
guardamos en un array para usarlos luego
92             this.mediaRecorder.ondataavailable = (e
) => {

```

```

93         this.chunks.push(e.data)
94     }
95
96     // Evento onStop
97     this.mediaRecorder.onstop = (e) => {
98         console.log(stream)
99         console.log('recorder stopped')
100
101         // Un objeto Blob representa un objeto
102         tipo fichero de datos planos inmutables
103         // Los Blobs representan datos que no
104         necesariamente se encuentran en un formato nativo
105         de JavaScript
106         var blob = new Blob(this.chunks, { '
107         type': 'audio/ogg; codecs=opus' })
108
109         // Vaciamos los 'trozos'
110         this.chunks = []
111         const audioURL = window.URL.
112         createObjectURL(blob)
113
114         this.audioList.push({
115             src: audioURL
116         })
117
118         window.URL.revokeObjectURL(blob)
119     }
120
121     .catch((err) => {
122         console.log('Err: ' + err)
123     })
124 } else {
125     this.supported = false
126 }
127 }
128 }
129 }
130 }
131 }
132 }
133 }
134 }
135 }
136 }
137 }
138 }
139 }
140 }
141 }
142 }
143 }
144 }
145 }
146 }
147 }
148 }
149 }
150 }
151 }
152 }
153 }
154 }
155 }
156 }
157 }
158 }
159 }
160 }
161 }
162 }
163 }
164 }
165 }
166 }
167 }
168 }
169 }
170 }
171 }
172 }
173 }
174 }
175 }
176 }
177 }
178 }
179 }
180 }
181 }
182 }
183 }
184 }
185 }
186 }
187 }
188 }
189 }
190 }
191 }
192 }
193 }
194 }
195 }
196 }
197 }
198 }
199 }
200 }
201 }
202 }
203 }
204 }
205 }
206 }
207 }
208 }
209 }
210 }
211 }
212 }
213 }
214 }
215 }
216 }
217 }
218 }
219 }
220 }
221 }
222 }
223 }
224 }
225 }
226 }
227 }
228 }
229 }
230 }
231 }
232 }
233 }
234 }
235 }
236 }
237 }
238 }
239 }
240 }
241 }
242 }
243 }
244 }
245 }
246 }
247 }
248 }
249 }
250 }
251 }
252 }
253 }
254 }
255 }
256 }
257 }
258 }
259 }
260 }
261 }
262 }
263 }
264 }
265 }
266 }
267 }
268 }
269 }
270 }
271 }
272 }
273 }
274 }
275 }
276 }
277 }
278 }
279 }
280 }
281 }
282 }
283 }
284 }
285 }
286 }
287 }
288 }
289 }
290 }
291 }
292 }
293 }
294 }
295 }
296 }
297 }
298 }
299 }
300 }
301 }
302 }
303 }
304 }
305 }
306 }
307 }
308 }
309 }
310 }
311 }
312 }
313 }
314 }
315 }
316 }
317 }
318 }
319 }
320 }
321 }
322 }
323 }
324 }
325 }
326 }
327 }
328 }
329 }
330 }
331 }
332 }
333 }
334 }
335 }
336 }
337 }
338 }
339 }
340 }
341 }
342 }
343 }
344 }
345 }
346 }
347 }
348 }
349 }
350 }
351 }
352 }
353 }
354 }
355 }
356 }
357 }
358 }
359 }
360 }
361 }
362 }
363 }
364 }
365 }
366 }
367 }
368 }
369 }
370 }
371 }
372 }
373 }
374 }
375 }
376 }
377 }
378 }
379 }
380 }
381 }
382 }
383 }
384 }
385 }
386 }
387 }
388 }
389 }
390 }
391 }
392 }
393 }
394 }
395 }
396 }
397 }
398 }
399 }
400 }
401 }
402 }
403 }
404 }
405 }
406 }
407 }
408 }
409 }
410 }
411 }
412 }
413 }
414 }
415 }
416 }
417 }
418 }
419 }
420 }
421 }
422 }
423 }
424 }
425 }
426 }
427 }
428 }
429 }
430 }
431 }
432 }
433 }
434 }
435 }
436 }
437 }
438 }
439 }
440 }
441 }
442 }
443 }
444 }
445 }
446 }
447 }
448 }
449 }
450 }
451 }
452 }
453 }
454 }
455 }
456 }
457 }
458 }
459 }
460 }
461 }
462 }
463 }
464 }
465 }
466 }
467 }
468 }
469 }
470 }
471 }
472 }
473 }
474 }
475 }
476 }
477 }
478 }
479 }
480 }
481 }
482 }
483 }
484 }
485 }
486 }
487 }
488 }
489 }
490 }
491 }
492 }
493 }
494 }
495 }
496 }
497 }
498 }
499 }
500 }
501 }
502 }
503 }
504 }
505 }
506 }
507 }
508 }
509 }
510 }
511 }
512 }
513 }
514 }
515 }
516 }
517 }
518 }
519 }
520 }
521 }
522 }
523 }
524 }
525 }
526 }
527 }
528 }
529 }
530 }
531 }
532 }
533 }
534 }
535 }
536 }
537 }
538 }
539 }
540 }
541 }
542 }
543 }
544 }
545 }
546 }
547 }
548 }
549 }
550 }
551 }
552 }
553 }
554 }
555 }
556 }
557 }
558 }
559 }
560 }
561 }
562 }
563 }
564 }
565 }
566 }
567 }
568 }
569 }
570 }
571 }
572 }
573 }
574 }
575 }
576 }
577 }
578 }
579 }
580 }
581 }
582 }
583 }
584 }
585 }
586 }
587 }
588 }
589 }
590 }
591 }
592 }
593 }
594 }
595 }
596 }
597 }
598 }
599 }
600 }
601 }
602 }
603 }
604 }
605 }
606 }
607 }
608 }
609 }
610 }
611 }
612 }
613 }
614 }
615 }
616 }
617 }
618 }
619 }
620 }
621 }
622 }
623 }
624 }
625 }
626 }
627 }
628 }
629 }
630 }
631 }
632 }
633 }
634 }
635 }
636 }
637 }
638 }
639 }
640 }
641 }
642 }
643 }
644 }
645 }
646 }
647 }
648 }
649 }
650 }
651 }
652 }
653 }
654 }
655 }
656 }
657 }
658 }
659 }
660 }
661 }
662 }
663 }
664 }
665 }
666 }
667 }
668 }
669 }
670 }
671 }
672 }
673 }
674 }
675 }
676 }
677 }
678 }
679 }
680 }
681 }
682 }
683 }
684 }
685 }
686 }
687 }
688 }
689 }
690 }
691 }
692 }
693 }
694 }
695 }
696 }
697 }
698 }
699 }
700 }
701 }
702 }
703 }
704 }
705 }
706 }
707 }
708 }
709 }
710 }
711 }
712 }
713 }
714 }
715 }
716 }
717 }
718 }
719 }
720 }
721 }
722 }
723 }
724 }
725 }
726 }
727 }
728 }
729 }
730 }
731 }
732 }
733 }
734 }
735 }
736 }
737 }
738 }
739 }
740 }
741 }
742 }
743 }
744 }
745 }
746 }
747 }
748 }
749 }
750 }
751 }
752 }
753 }
754 }
755 }
756 }
757 }
758 }
759 }
760 }
761 }
762 }
763 }
764 }
765 }
766 }
767 }
768 }
769 }
770 }
771 }
772 }
773 }
774 }
775 }
776 }
777 }
778 }
779 }
780 }
781 }
782 }
783 }
784 }
785 }
786 }
787 }
788 }
789 }
790 }
791 }
792 }
793 }
794 }
795 }
796 }
797 }
798 }
799 }
800 }
801 }
802 }
803 }
804 }
805 }
806 }
807 }
808 }
809 }
810 }
811 }
812 }
813 }
814 }
815 }
816 }
817 }
818 }
819 }
820 }
821 }
822 }
823 }
824 }
825 }
826 }
827 }
828 }
829 }
830 }
831 }
832 }
833 }
834 }
835 }
836 }
837 }
838 }
839 }
840 }
841 }
842 }
843 }
844 }
845 }
846 }
847 }
848 }
849 }
850 }
851 }
852 }
853 }
854 }
855 }
856 }
857 }
858 }
859 }
860 }
861 }
862 }
863 }
864 }
865 }
866 }
867 }
868 }
869 }
870 }
871 }
872 }
873 }
874 }
875 }
876 }
877 }
878 }
879 }
880 }
881 }
882 }
883 }
884 }
885 }
886 }
887 }
888 }
889 }
890 }
891 }
892 }
893 }
894 }
895 }
896 }
897 }
898 }
899 }
900 }
901 }
902 }
903 }
904 }
905 }
906 }
907 }
908 }
909 }
910 }
911 }
912 }
913 }
914 }
915 }
916 }
917 }
918 }
919 }
920 }
921 }
922 }
923 }
924 }
925 }
926 }
927 }
928 }
929 }
930 }
931 }
932 }
933 }
934 }
935 }
936 }
937 }
938 }
939 }
940 }
941 }
942 }
943 }
944 }
945 }
946 }
947 }
948 }
949 }
950 }
951 }
952 }
953 }
954 }
955 }
956 }
957 }
958 }
959 }
960 }
961 }
962 }
963 }
964 }
965 }
966 }
967 }
968 }
969 }
970 }
971 }
972 }
973 }
974 }
975 }
976 }
977 }
978 }
979 }
980 }
981 }
982 }
983 }
984 }
985 }
986 }
987 }
988 }
989 }
990 }
991 }
992 }
993 }
994 }
995 }
996 }
997 }
998 }
999 }
1000 }

```

```
127 /*
128   audio::-webkit-media-controls-panel,
129   audio::-webkit-media-controls-mute-button,
130   audio::-webkit-media-controls-play-button
131   audio::-webkit-media-controls-timeline,
132   audio::-webkit-media-controls-timeline-container
133   ,
134   audio::-webkit-media-controls-current-time-
display,
135   audio::-webkit-media-controls-time-remaining-
display,
136   audio::-webkit-media-controls-volume-slider,
137   audio::-webkit-media-controls-volume-slider-
container,
138   audio::-webkit-media-controls-seek-back-button,
139   audio::-webkit-media-controls-seek-forward-
button,
140   audio::-webkit-media-controls-fullscreen-button,
141   audio::-webkit-media-controls-rewind-button,
142   audio::-webkit-media-controls-return-to-realtime
-button,
143   audio::-webkit-media-controls-toggle-closed-
captions-button
144   color red
145   background-color aqua
146 */
147 audio {
148   width: 600px;
149   box-shadow: 5px 5px 20px rgba(0,0, 0, 0.4);
150   border-radius: 90px;
151   transform: scale(1.05);
152 }
153 </style>
154
```