**Estructura del código PÉEK**

La aplicación PÉEK se desarrolló con la arquitectura Cliente-Servidor, donde el usuario interactúa con una página web (Cliente) y se conecta con Python (Servidor) que realiza todo el procesamiento de la señal de electrocardiograma y el genera un diagnóstico por cada ciclo cardíaco que se presente en ella.

En lo que resta del documento, a la señal de Electrocardiograma (ECG) se le denominará “trazo”.

La interfaz del cliente de la aplicación PÉEK fue desarrollada utilizando Anvil, un servicio web para diseño de aplicaciones Full Stack, el cual permite su edición en HTML y CSS, o utilizar el modo gráfico para facilitar la ubicación y edición de objetos mediante el arrastre y personalización gráfica. Para esta aplicación, fue utilizado el modo gráfico.

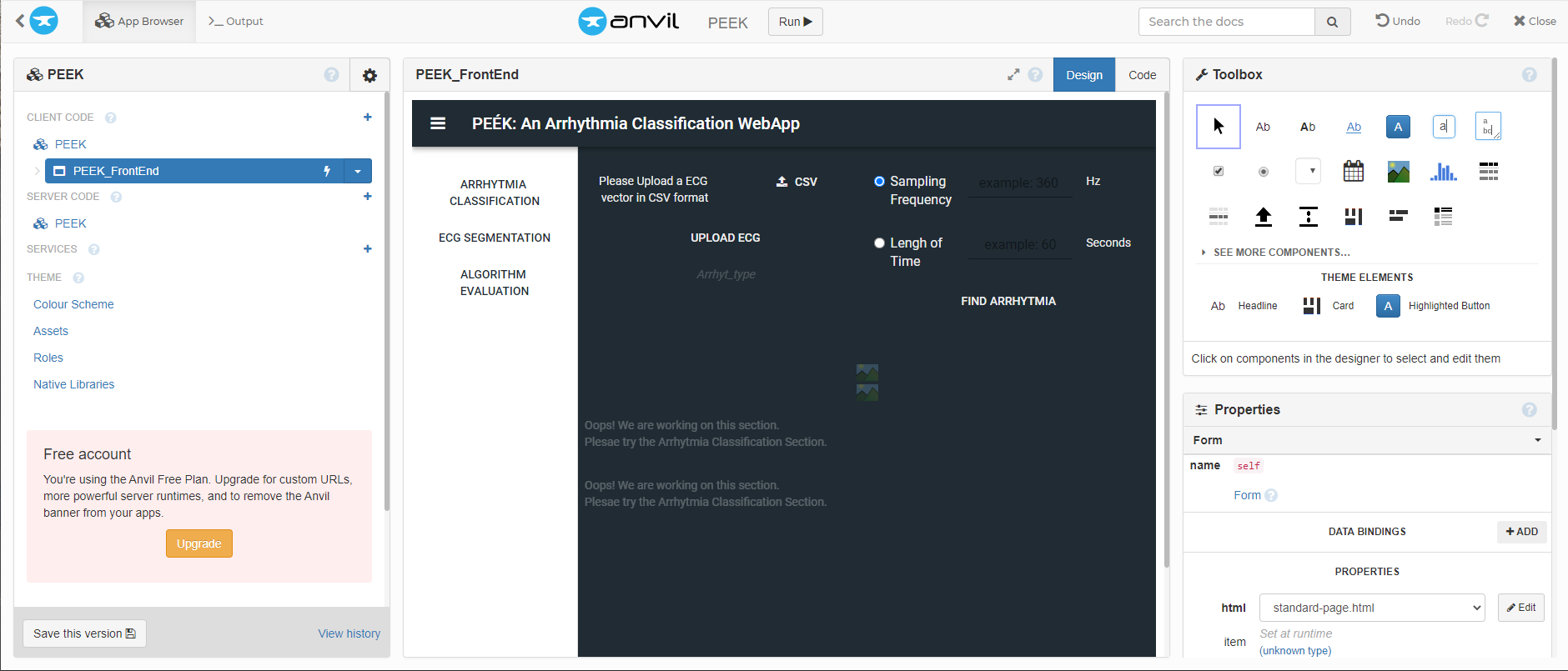


Figura. 1 Ambiente de programación por bloques Anvil

La interfaz de la aplicación PÉEK presenta la siguiente información:

1. Nombre de la aplicación web.
2. Menú hamburguesa con funciones que se desarrollarán como trabajo futuro.
3. Botón para subir el trazo de ECG como vector en archivo CSV.
4. Botón para cargar el archivo seleccionado y presentarlo en pantalla.
5. Selectores con sus respectivos cuadros de texto para declarar la frecuencia de muestreo del trazo o la duración del mismo en segundos.
6. Botón para comenzar la identificación de arritmias y plasmarlo en pantalla.
7. Visualizador de trazo electrográfico tras cargarlo desde el archivo CSV.
8. Visualizador de trazo electrográfico con marcas de arritmias identificadas.

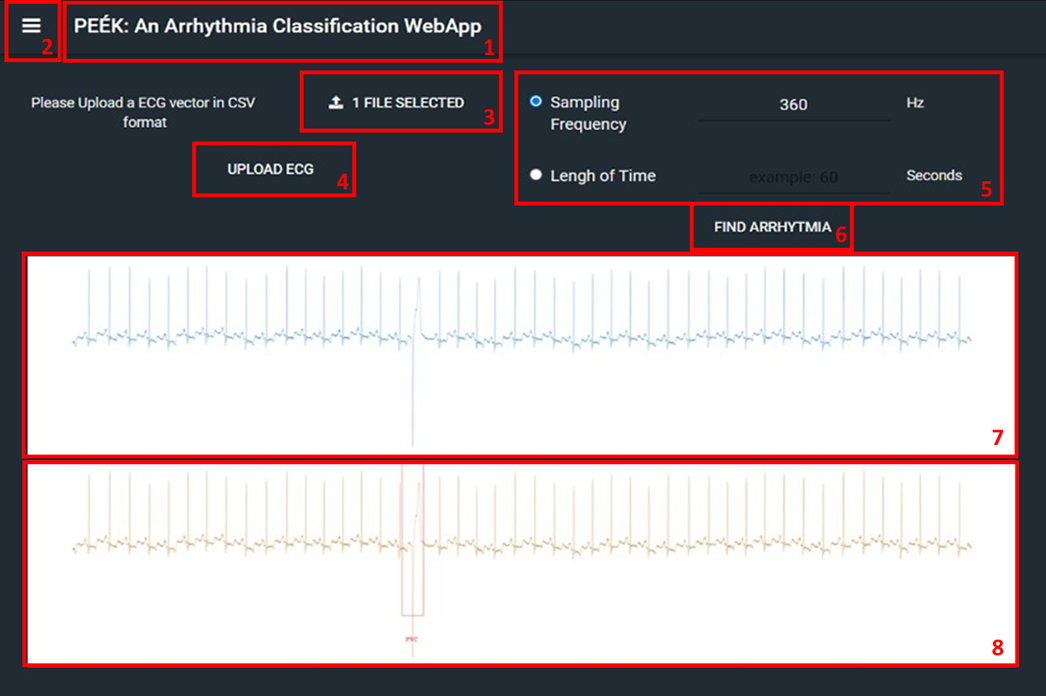


Figura. 2 Elementos de la interfaz PÉEK

La funcionalidad de la aplicación, tanto la interacción Cliente-Servidor como el procesamiento dentro del servidor fueron desarrolladas en Python, esto se muestra la página 27 en adelante

.

A continuación, se muestran los códigos CSS y HTML generados a partir del diseño por bloques.

---------------------------------

standard-page.html

---------------------------------

1. <[**link**](http://december.com/html/4/element/link.html) href="https://fonts.googleapis.com/css?family=Roboto:300,400,500" rel="stylesheet" rel="preload" as="font" crossorigin="anonymous">
3. <**[div](http://december.com/html/4/element/div.html)** class="structure">
4. <[**div**](http://december.com/html/4/element/div.html) class="app-bar" anvil-drop-container=".anvil-container" anvil-drop-redirect=".placeholder">
5. <[**a**](http://december.com/html/4/element/a.html) class="sidebar-toggle" anvil-if-slot-empty="top-left-btn" anvil-hide-if-slot-empty="left-nav" anvil-drop-slot="top-left-btn" href="javascript:void(0)"><[**i**](http://december.com/html/4/element/i.html) class="fa fa-bars"></[**i**](http://december.com/html/4/element/i.html)></[**a**](http://december.com/html/4/element/a.html)>
6. <[**a**](http://december.com/html/4/element/a.html) class="sidebar-toggle anvil-designer-only" anvil-if-slot-empty="top-left-btn" anvil-if-slot-empty="left-nav" anvil-drop-slot="top-left-btn"><[**i**](http://december.com/html/4/element/i.html) class="fa fa-blank"></[**i**](http://december.com/html/4/element/i.html)></[**a**](http://december.com/html/4/element/a.html)>
7. <**[div](http://december.com/html/4/element/div.html)** class="top-left-btn" anvil-slot="top-left-btn"></**[div](http://december.com/html/4/element/div.html)**>
8. <**[div](http://december.com/html/4/element/div.html)** class="title" anvil-slot="title">
9. <**[div](http://december.com/html/4/element/div.html)** class="placeholder anvil-designer-only" anvil-if-slot-empty="title" anvil-drop-here>Drop title here</**[div](http://december.com/html/4/element/div.html)**>
10. </**[div](http://december.com/html/4/element/div.html)**>
11. <**[div](http://december.com/html/4/element/div.html)** class="app-bar-nav" anvil-slot="nav-right">
12. <**[div](http://december.com/html/4/element/div.html)** class="placeholder anvil-designer-only" anvil-if-slot-empty="nav-right" anvil-drop-here>Drop a FlowPanel here</**[div](http://december.com/html/4/element/div.html)**>
13. </**[div](http://december.com/html/4/element/div.html)**>
14. <**[div](http://december.com/html/4/element/div.html)** style="clear:both"></**[div](http://december.com/html/4/element/div.html)**>
15. </**[div](http://december.com/html/4/element/div.html)**>
17. <**[div](http://december.com/html/4/element/div.html)** class="nav-holder">
18. <**[div](http://december.com/html/4/element/div.html)** class="left-nav anvil-measure-this" anvil-slot-repeat="left-nav" anvil-drop-container=">.anvil-container">
19. </**[div](http://december.com/html/4/element/div.html)**>
20. <**[div](http://december.com/html/4/element/div.html)** class="left-nav-placeholder anvil-designer-only" anvil-if-slot-empty="left-nav" anvil-drop-slot="left-nav">
21. <**[div](http://december.com/html/4/element/div.html)** class="prompt">To add a sidebar, drop a ColumnPanel here.</**[div](http://december.com/html/4/element/div.html)**>
22. </**[div](http://december.com/html/4/element/div.html)**>
24. <**[div](http://december.com/html/4/element/div.html)** class="content">
25. <[**div**](http://december.com/html/4/element/div.html) anvil-slot-repeat="default" class="anvil-measure-this"></[**div**](http://december.com/html/4/element/div.html)>
26. <**[div](http://december.com/html/4/element/div.html)** class="placeholder drop-here" anvil-if-slot-empty="default" anvil-drop-slot="default">Drop a ColumnPanel here.</**[div](http://december.com/html/4/element/div.html)**>
27. </**[div](http://december.com/html/4/element/div.html)**>
28. </**[div](http://december.com/html/4/element/div.html)**>
29. <**[div](http://december.com/html/4/element/div.html)** class="nav-shield"></**[div](http://december.com/html/4/element/div.html)**>
30. </**[div](http://december.com/html/4/element/div.html)**>
31. <**[div](http://december.com/html/4/element/div.html)** anvil-drop-default anvil-drop-redirect=".placeholder" anvil-drop-container=".anvil-container"></[**div**](http://december.com/html/4/element/div.html)>
33. <[**script**](http://december.com/html/4/element/script.html)>
34. function hideSidebar() {
35. var ln = $('.structure > .nav-holder > .left-nav');
36. ln.animate({left: -ln.outerWidth()}, function() {
37. ln.removeClass("in-transition shown").addClass("hidden");
38. $('.nav-shield').removeClass("shown");
39. $(window).trigger('resize');
40. });
41. }
42. function showSidebar() {
43. var ln = $('.structure > .nav-holder > .left-nav');
44. $('.nav-shield').addClass("shown");
45. ln.addClass("shown").removeClass("hidden").css({left: "-100%"}).css({left: -ln.outerWidth()}).animate({left: 0}, function() {
46. ln.removeClass("in-transition");
47. });
48. $(window).trigger('resize');
49. }
50. $('.sidebar-toggle, .nav-shield').off('click').on('click', function() {
51. var ln = $('.structure > .nav-holder > .left-nav');
52. if (ln.is(":visible") || $('.nav-shield').is(".shown")) {
53. hideSidebar();
54. } else if(!ln.is(":empty")) {
55. showSidebar();
56. }
57. });
58. $('.left-nav').off('click').on('click', 'a, button', function() {
59. if ($('.nav-shield').is(":visible")) {
60. $('.nav-shield').trigger('click');
61. }
62. });
63. </[**script**](http://december.com/html/4/element/script.html)>

---------------------------------

theme.css

---------------------------------

1. */\*\**
2. *This CSS implements the Material Design look and feel for Anvil apps.*
3. *\*\*/*
5. */\* Stop panel-cols cutting off drop shadows \*/*
6. .anvil-panel-col {
7. **padding-bottom**: 10px;
8. **margin-bottom**: -10px;
9. }
11. .content > .placeholder {
12. **margin**: 16px;
13. **color**: #888;
14. **font-size**: 18px;
15. **outline**: 1px dotted;
16. **padding**: 16px;
17. **text-align**: center;
18. }
20. */\* Put things on a 4px grid (none of this 7px nonsense) \*/*
21. .has-text .anvil-component-icon.left-icon { **margin-right**: 8px; }
22. .has-text .anvil-component-icon.right-icon { **margin-left**: 8px; }
24. a>.anvil-component-icon.left\_edge-icon,
25. a>.anvil-component-icon.right\_edge-icon,
26. .anvil-label>.anvil-component-icon.left\_edge-icon,
27. .anvil-label>.anvil-component-icon.right\_edge-icon {
28. **padding-top**: 8px;
29. **padding-bottom**: 8px;
30. }
31. .anvil-label>.label-text, a>.link-text {
32. **padding-top**: 8px;
33. **padding-bottom**: 8px;
34. }
35. .file-loader>label {
36. **padding**: 8px;
37. }
38. .checkbox label, .radio label {
39. **padding-top**: 8px; **padding-right**: 8px; **padding-bottom**: 8px;
40. }
41. .anvil-spacing-above-none { **margin-top**: 0px; }
42. .anvil-spacing-above-small { **margin-top**: 4px; }
43. .anvil-spacing-above-medium { **margin-top**: 8px; }
44. .anvil-spacing-above-large { **margin-top**: 16px; }
45. .anvil-spacing-below-none { **margin-bottom**: 0px; }
46. .anvil-spacing-below-small { **margin-bottom**: 4px; }
47. .anvil-spacing-below-medium { **margin-bottom**: 8px; }
48. .anvil-spacing-below-large { **margin-bottom**: 16px; }

51. .col-padding.col-padding-tiny { **padding**: 0 2px; }
52. .column-panel.col-padding-tiny > .anvil-panel-section > .anvil-panel-section-container > .anvil-panel-section-gutter { **margin**: 0 -2px; }
54. .col-padding.col-padding-small { **padding**: 0 4px; }
55. .column-panel.col-padding-small > .anvil-panel-section > .anvil-panel-section-container > .anvil-panel-section-gutter { **margin**: 0 -4px; }
57. .col-padding.col-padding-medium { **padding**: 0 8px; }
58. .column-panel.col-padding-medium > .anvil-panel-section > .anvil-panel-section-container > .anvil-panel-section-gutter { **margin**: 0 -8px; }
60. .col-padding.col-padding-large { **padding**: 0 12px; }
61. .column-panel.col-padding-large > .anvil-panel-section > .anvil-panel-section-container > .anvil-panel-section-gutter { **margin**: 0 -12px; }
63. .col-padding.col-padding-huge { **padding**: 0 20px; }
64. .column-panel.col-padding-huge > .anvil-panel-section > .anvil-panel-section-container > .anvil-panel-section-gutter { **margin**: 0 -20px; }
66. .flow-panel.flow-spacing-tiny > .flow-panel-gutter { **margin**: 0 -2px; }
67. .flow-panel.flow-spacing-tiny > .flow-panel-gutter > .flow-panel-item {
68. **margin-left**: 2px;
69. **margin-right**: 2px;
70. }
72. .flow-panel.flow-spacing-small > .flow-panel-gutter { **margin**: 0 -4px; }
73. .flow-panel.flow-spacing-small > .flow-panel-gutter > .flow-panel-item {
74. **margin-left**: 4px;
75. **margin-right**: 4px;
76. }
78. .flow-panel.flow-spacing-medium > .flow-panel-gutter { **margin**: 0 -8px; }
79. .flow-panel.flow-spacing-medium > .flow-panel-gutter > .flow-panel-item {
80. **margin-left**: 8px;
81. **margin-right**: 8px;
82. }
84. .flow-panel.flow-spacing-large > .flow-panel-gutter { **margin**: 0 -12px; }
85. .flow-panel.flow-spacing-large > .flow-panel-gutter > .flow-panel-item {
86. **margin-left**: 12px;
87. **margin-right**: 12px;
88. }
90. .flow-panel.flow-spacing-huge > .flow-panel-gutter { **margin**: 0 -20px; }
91. .flow-panel.flow-spacing-huge > .flow-panel-gutter > .flow-panel-item{
92. **margin-left**: 20px;
93. **margin-right**: 20px;
94. }

97. */\* Typography \*/*
98. body {
99. **font-family**: Roboto, Noto, Arial, sans-serif;
100. **font-size**: 14px;
101. **line-height**: 1.4286;
102. **background-color**: #202B33;
103. }
105. **@media** print {
106. body {
107. **background-color**: black;
108. }
109. }
111. a, a:focus {
112. **text-decoration**: none;
113. **color**: %color:Primary 700%;
114. }
116. a:hover, a:active {
117. **color**: %color:Primary 500%;
118. }
120. .anvil-label {
121. */\*  font-size: 16px;\*/*
122. }
124. .anvil-role-subheading {
125. **font-size**: 16px;
126. **line-height**: 24dp;
127. }
129. .anvil-role-headline {
130. **font-size**: 24px;
131. **line-height**: 32px;
132. }
134. .anvil-role-input-prompt {
135. **font-size**: 16px;
136. **line-height**: 1.5;
137. }
139. .anvil-role-display-4 {
140. **font-size**: 112px;
141. **font-weight**: light;
142. }
144. .anvil-role-text {
145. **font-size**: 14px;
146. }
147. .anvil-role-text > .label-text, .anvil-role-text .link-text {
148. **padding-top**: 0;
149. **padding-bottom**: 0;
150. }
152. */\* Page structure: App bar with optional left nav \*/*
153. .structure {
154. **display**: flex;
155. **flex-direction**: column;
156. **min-height**: 100vh;
157. **min-height**: calc(100vh - %anvil-banner-height%);
158. **height**: 0; */\* To make flex-grow work in IE \*/*
159. }
161. */\* However, this breaks the designer's height measurement. IE isn't supported for the designer, so set it back. \*/*
162. .designer .structure {
163. **height**: initial;
164. }
166. .app-bar {
167. **flex**: 0 0;
168. **flex-basis**: auto;
169. **flex-basis**: content;
170. }
172. .nav-holder {
173. **flex-grow**: 1;
174. **overflow-y**: auto;
175. }
177. **@media** print {
178. .nav-holder {
179. **overflow-y**: initial;
180. }
181. }

184. */\* Mobile and desktop margins for content \*/*
185. .content > \* > .anvil-container {
186. **padding**: 8px;
187. }
189. @media(**min-width**:991px) {
190. .content > \* > .anvil-container {
191. **padding**: 16px 24px;
192. }
193. }
195. .content .anvil-measure-this {
196. **padding-bottom**: 1px; */\* Prevent margin collapse messing up embedding \*/*
197. }
199. */\* Allow overflows to show drop shadows in ColumnPanels*
200. *This can create unwanted scrollbars; we compensate for this at the top level with .nav-holder .content {overflow-x: hidden;}*
201. *\*/*
202. .anvil-container-overflow {
203. **overflow-x**: visible;
204. **overflow-y**: visible;
205. }

208. */\* Desktop: Nav bar pinned \*/*
210. .nav-holder {
211. **display**: flex;
212. **flex-direction**: row;
213. }
215. .nav-holder .left-nav {
216. **position**: relative;
217. **flex-shrink**: 0;
218. **min-width**: 160px;
219. **max-width**: 400px;
220. }
222. .nav-shield {
223. **display**: none;
224. }
226. */\* Mobile: Nav bar is a modal overlay \*/*
228. @media(**max-width**:998px) {
229. html:not(.designer) .nav-holder {
230. **display**: block;
231. }
232. html:not(.designer) .nav-holder .left-nav {
233. **position**: fixed;
234. **top**: 0;
235. **bottom**: 0;
236. **width**: calc(100% - 56px);
237. **max-width**: 360px;
238. **z-index**: 3;
239. **border-right**: none;
240. */\* 16dp \*/* **box-shadow**: 0 16px 24px 2px rgba(0, 0, 0, 0.14), 0 6px 30px 5px rgba(0, 0, 0, 0.12), 0 8px 10px -5px rgba(0, 0, 0, 0.2);
242. **display**: none;
243. **transition**: right 0.5s;
244. }
246. html:not(.designer) .nav-holder .left-nav.shown {
247. **display**: block;
248. }
249. .nav-shield.shown {
250. **display**: block;
251. **position**: fixed;
252. **top**: 0;
253. **bottom**: 0;
254. **left**: 0;
255. **right**: 0;
256. **z-index**: 2;
257. **background-color**: rgba(0,0,0,0.2);
258. }
259. }
261. .nav-holder .left-nav, .left-nav-placeholder {
262. **display**: flex;
263. **flex-direction**: column;
264. **background**: white;
265. **border-right**: 1px solid #e0e0e0;
266. **font-size**: 14px;
267. **font-weight**: 500;
268. **color**: rgba(0,0,0,0.87);
269. **overflow-x**: hidden;
270. **overflow-y**: auto;
271. }
273. .left-nav-placeholder {
274. **display**: block;
275. **padding**: 8px;
276. **line-height**: 1;
277. **width**: 58px;
278. }
279. .left-nav-placeholder .prompt {
280. **display**: inline-block;
281. **white-space**: nowrap;
282. **transform**: translate(-50%,0) rotate(-90deg) translate(-50%,0) translate(15px,16px);
283. **padding**: 16px 16px;
284. **color**: #888;
285. **outline**: 1px dotted #888;
286. **visibility**: hidden;
287. }
288. .anvil-highlight .left-nav-placeholder .prompt {
289. **visibility**: visible;
290. }
292. .left-nav > .column-panel {
293. **padding**: 24px 0;
294. }
295. .left-nav > .column-panel > .anvil-panel-section > .anvil-panel-section-container:not(.full-width-row) {
296. **margin**: 0 16px;
297. **width**: initial;
298. **max-width**: initial;
299. **overflow-x**: visible;
300. }
301. .left-nav > .column-panel > .anvil-panel-section:first-child > .anvil-panel-section-container.full-width-row {
302. **margin-top**: -24px;
303. }
304. .left-nav > .column-panel > .anvil-panel-section > .anvil-panel-section-container > .anvil-panel-section-gutter > .anvil-panel-row > .anvil-panel-col {
305. **overflow-x**: visible;
306. }
308. */\* Make all the sidebar icons look nice \*/*
310. .left-nav .anvil-component-icon {
311. **font-size**: 160%;
312. }
314. .left-nav .anvil-component-icon.left-icon {
315. **width**: 0;
316. **margin-right**: 56px;
317. **position**: relative;
318. **top**: 0.15em;
319. }
321. .left-nav .anvil-component-icon.left\_edge-icon, .left-nav .anvil-component-icon.right\_edge-icon {
322. **left**: 16px;
323. **padding-top**: 2px;
324. **width**: 0;
325. }
327. .left-nav .anvil-component-icon.right\_edge-icon {
328. **left**: initial;
329. **right**: 16px;
330. **top**: 0;
331. **padding-top**: 2px;
332. }
334. */\* Sidebar links (and labels with edge icons) go +16px wider*
335. *(Top-level columns in ColumnPanels get overflow-x visible [see above]to enable this)*
336. *\*/*
338. .left-nav a, .left-nav .anvil-label.left\_edge-icon, .left-nav .anvil-label.right\_edge-icon {
339. **color**: rgba(0,0,0,0.87);
340. **margin**: 0 -16px;
341. **padding**: 4px 16px;
342. }
344. .left-nav .anvil-component.left\_edge-icon {
345. **padding-left**: 72px;
346. }
348. .left-nav a:hover, .left-nav .anvil-role-selected {
349. **color**: %color:Primary 700%;
350. **background-color**: rgba(0,0,0,.08);
351. }
353. .designer .nav-holder .left-nav {
354. **min-width**: 56px;
355. }
357. .designer .nav-holder .left-nav > .anvil-component {
358. **min-width**: 160px;
359. }
361. .nav-holder .left-nav > .anvil-component {
362. **margin-top**: 0;
363. **margin-bottom**: 0;
364. }
366. .nav-holder .content {
367. **flex**: 1;
368. **overflow-x**: hidden;
369. }
371. */\* Components: App Bar \*/*
372. .app-bar {
373. **min-height**: 56px;
374. **line-height**: 40px;
375. **padding**: 0 16px 0 72px;
377. **font-size**: 20px;
378. **font-weight**: 500;
379. **color**: white;
380. **background-color**: %color:Primary 500%;
382. **z-index**: 1;
383. */\* 4dp \*/* **box-shadow**: 0 4px 5px 0 rgba(0, 0, 0, 0.14), 0 1px 10px 0 rgba(0, 0, 0, 0.12), 0 2px 4px -1px rgba(0, 0, 0, 0.2);
384. }
386. .app-bar .anvil-component {
387. **margin-top**: 0;
388. **margin-bottom**: 0;
389. }
391. .app-bar > .sidebar-toggle { **display**:block; **float**:left;  **margin-left**: -61px; }
392. .app-bar > .top-left-btn { **float**: left; **margin-left**: -60px; **margin-right**: -40px; }
393. .app-bar > .title { **float**: left; }
394. .app-bar > .title > .placeholder { **outline**: 1px dotted; **padding-left**: 16px; **padding-right**: 16px; **margin**: 8px 8px 0; **display**:none; }
395. .app-bar > .app-bar-nav { **float**: right; **font-weight**: normal; }
396. .app-bar > .app-bar-nav .placeholder { **margin-top**: 8px; **padding**: 0 8px; }
397. .anvil-highlight .app-bar > .app-bar-nav .placeholder { **outline**: 1px dotted #ccc; }
398. .anvil-highlight .app-bar > .title > .placeholder { **display**:block; }
400. .app-bar a, .app-bar .anvil-component {
401. **display**: block;
402. **color**: white;
403. **line-height**: 30px;
404. **min-width**: 30px;
405. **min-height**: 30px;
406. **margin**: 5px 4px 0;
407. }
409. .app-bar .title .anvil-component {
410. **margin-left**: 0;
411. }
413. .app-bar input.anvil-component, .app-bar .anvil-component input {
414. **line-height**: 30px;
415. **color**: white;
416. }
418. .app-bar select.form-control {
419. **height**: 43px; */\* 30px line height + 12px padding + 1px border \*/*
420. }
422. .app-bar .anvil-component-icon-present .anvil-component-icon, .app-bar .sidebar-toggle .fa {
423. **width**: 30px;
424. **text-align**: center;
425. }
427. */\* Don't put spacing on the top-level container (eg FlowPanel) \*/*
428. .app-bar-nav > .anvil-container {
429. **margin**: 0;
430. **padding**: 0;
431. **min-height**: 56px;
432. }
434. .anvil-highlight .app-bar-nav > .anvil-container:not(.has-components):not(a) {
435. **outline**: 1px dotted #fff;
436. **margin-top**: 8px;
437. **min-height**: 40px;
438. **padding**: 5px 8px;
439. }
441. .anvil-highlight .app-bar-nav > .anvil-container:not(.has-components):not(a)::after {
442. **content**: "Drop Links here"
443. }
445. */\* Rejig margin vs padding to give links the halo \*/*
446. .app-bar a, .app-bar a.anvil-component {
447. **padding**: 5px;
448. **margin**: 8px 0 0;
449. **border-radius**: 20px;
450. }
452. .app-bar a .link-text {
453. **padding**: 0 4px;
454. }
455. .app-bar a.has-text .anvil-component-icon.left {
456. **margin-right**: 4px;
457. }
458. .app-bar a.has-text .anvil-component-icon.right {
459. **margin-left**: 4px;
460. }
462. .app-bar a:hover, .app-bar a:active {
463. **color**: #eee; **text-decoration**: none;
464. **background-color**: %color:Primary 700%;
465. }

468. */\* Component: Dialog (alert), Notification \*/*
469. .modal-content, body>div[data-notify="container"] {
470. **border-radius**: 2px;
471. */\* 24dp \*/* **box-shadow**: 0 9px 46px 8px rgba(0, 0, 0, 0.14), 0 11px 15px -7px rgba(0, 0, 0, 0.12), 0 24px 38px 3px rgba(0, 0, 0, 0.2)
472. !important;
473. }
475. .modal-header {
476. **padding**: 24px 24px 0px;
477. **font-size**: 24px;
478. **line-height**: 32px;
479. **border**: 0;
480. }
482. .modal-body {
483. **padding**: 20px 24px 24px;
484. **font-size**: 16px;
485. **color**: #747474;
486. }
488. .modal-footer {
489. **padding**: 8px 8px 8px 24px;
490. **border**: 0;
491. }
493. .modal-footer .btn {
494. **background-color**: transparent;
495. **color**: %color:Primary 500%;
496. }
497. .modal-footer .btn, .modal-footer .btn:focus, .modal-footer .btn:active {
498. **box-shadow**: none;
499. **top**: 0;
500. }
502. */\* Notifications \*/*
503. body>div[data-notify="container"] {
504. **background**: white;
505. **border**: none;
506. **padding**: 24px;
507. }
509. body>div[data-notify="container"]>span[data-notify="title"] {
510. **display**: block;
511. **font-size**: 18px;
512. **font-weight**: 500;
513. **color**: #333333;
514. }
516. body>div[data-notify="container"]>span[data-notify="message"] {
517. **display**: block;
518. **padding**: 20px 0 0;
519. **font-size**: 16px;
520. **color**: #747474;
521. }
523. body>div[data-notify="container"].alert-success{
524. **background-color**: #d4edda;
525. }
527. body>div[data-notify="container"].alert-success>span[data-notify="title"]{
528. **color**: #155724;
529. }
531. body>div[data-notify="container"].alert-danger{
532. **background-color**: #f8d7da;
533. }
535. body>div[data-notify="container"].alert-danger>span[data-notify="title"]{
536. **color**: #721c24;
537. }
539. body>div[data-notify="container"].alert-warning{
540. **background-color**: #fff3cd;
541. }
543. body>div[data-notify="container"].alert-warning>span[data-notify="title"]{
544. **color**: #856404;
545. }

548. */\* Component: Button \*/*
550. .btn, .btn-default, .file-loader>label {
551. **border-radius**: 2px;
552. **font-size**: 14px;
553. **font-weight**: 500;
554. **padding**: 8px 16px;
555. **min-height**: calc(16px + 1.42857143em);
556. **text-transform**: uppercase;
558. **border**: 0;
559. **background-image**: none;
560. **background-color**: transparent;
561. **color**: %color:Primary 500%;
562. **text-shadow**: none;
563. **box-shadow**: none;
564. -webkit-box-shadow: none;
566. **position**: relative;
567. }
569. .btn:hover, .btn:focus, .file-loader>label:hover {
570. **background-color**: rgba(153,153,153,0.2);
571. **background-image**: none;
572. **outline**: none;
573. **color**: %color:Primary 500%;
574. }
576. .btn:active, .btn:active:focus {
577. **background-color**: rgba(153,153,153,0.4);
578. **background-image**: none;
579. **outline**: none;
580. **box-shadow**: none;
581. **color**: %color:Primary 500%;
582. }
584. .btn[disabled] { **color**: rgba(0, 0, 0, .25); }
585. .anvil-role-raised > .btn[disabled], .anvil-role-primary-color > .btn[disabled], .anvil-role-secondary-color > .btn[disabled] {
586. **background-color**: rgba(0,0,0,0.12) !important;
587. **color**: rgba(0,0,0,0.25) !important;
588. **box-shadow**: none !important;
589. }
591. .anvil-role-raised > .btn, .anvil-role-primary-color > .btn, .anvil-role-secondary-color > .btn, .anvil-role-raised.file-loader>label, .anvil-role-primary-color.file-loader>label, .anvil-role-secondary-color.file-loader>label,
592. .btn-primary, .btn-success, .btn-info, .btn-warning, .btn-danger
593. {
594. */\* 2dp \*/*  **box-shadow**: 0 2px 2px 0 rgba(0, 0, 0, 0.14), 0 3px 1px -2px rgba(0, 0, 0, 0.2), 0 1px 5px 0 rgba(0, 0, 0, 0.12);
595. }
597. .anvil-role-raised > .btn:active, .anvil-role-primary-color > .btn:active, .anvil-role-secondary-color > .btn:active, .btn-primary:active,
598. .btn-primary:active, .btn-success:active, .btn-info:active, .btn-warning:active, .btn-danger:active {
599. */\* 8dp \*/* **box-shadow**: 0 8px 10px 1px rgba(0, 0, 0, 0.14), 0 3px 14px 2px rgba(0, 0, 0, 0.12), 0 5px 5px -3px rgba(0, 0, 0, 0.2);
600. **top**: -1px;
601. }
603. .anvil-role-primary-color > .btn, .btn-primary, .anvil-role-primary-color.file-loader>label {
604. **color**: white;
605. **background-color**: %color:Primary 500%;
606. }
608. .anvil-role-secondary-color > .btn, .anvil-role-secondary-color.file-loader>label {
609. **color**: white;
610. **background-color**: %color:Secondary 500%;
611. }
613. .btn-success, .btn-info, .btn-warning, .btn-danger {
614. **color**: white;
615. }
617. .anvil-role-primary-color > .btn:hover, .anvil-role-primary-color > .btn:active, .anvil-role-primary-color > .btn:focus,
618. .btn-primary:hover, .btn-primary:active, .btn-primary:focus {
619. **color**: white;
620. **background-color**: %color:Primary 700%;
621. }
623. .anvil-role-secondary-color > .btn:hover, .anvil-role-secondary-color > .btn:active, .anvil-role-secondary-color > .btn:focus {
624. **color**: white;
625. **background-color**: %color:Secondary 700%;
626. }
628. */\* Component: Card \*/*
630. .anvil-role-card {
631. **overflow**: hidden;
632. **border-radius**: 2px;
633. **background-color**: white;
634. **padding**: 8px 12px;
635. */\* 2dp \*/*  **box-shadow**: 0 2px 2px 0 rgba(0, 0, 0, 0.14), 0 3px 1px -2px rgba(0, 0, 0, 0.2), 0 1px 5px 0 rgba(0, 0, 0, 0.12);
636. }
638. */\* full\_width\_row sections go full bleed \*/*
639. .column-panel.anvil-role-card > .anvil-panel-section > .anvil-panel-section-container.full-width-row {
640. **margin-left**: -12px;
641. **margin-right**: -12px;
642. **width**: initial;
643. **max-width**: initial;
644. }
646. */\* If the first row of a card is a full\_width\_row, it goes to the top \*/*
647. .column-panel.anvil-role-card > .anvil-panel-section:first-child > .anvil-panel-section-container.full-width-row {
648. **margin-top**: -8px;
649. }
650. .column-panel.anvil-role-card > .anvil-panel-section:last-child > .anvil-panel-section-container.full-width-row {
651. **margin-bottom**: -8px;
652. }


656. */\* Component: TextBox + TextArea \*/*
658. input.anvil-component, textarea.anvil-component, .anvil-component select, .anvil-datepicker input {
659. **font-size**: 16px;
660. **line-height**: 1.5;
661. **border-radius**: 0;
662. **background-color**: transparent;
663. **color**: rgba(0,0,0,0.87);
665. **box-shadow**: none;
666. }
668. .anvil-dropdown {
669. **font-size**: 16px;
670. }
671. .anvil-component select {
672. **font-size**: inherit;
673. }
675. input.anvil-component, .anvil-component select, .anvil-datepicker input {
676. **padding**: 8px 0 4px;
677. **border**: 0;
678. **border-bottom**: 1px solid rgba(0,0,0,0.42);
680. **transition**: border-bottom 0.2s, padding-bottom 0.2s;
681. -moz-transition: none; */\* Firefox is jumpy, so disable transitions \*/*
682. }
684. textarea.anvil-component {
685. **padding**: 4px 8px;
686. **border**: 1px solid rgba(0,0,0,0.42);
687. **border-radius**: 2px;
688. **transition**: border 0.2s, padding 0.2s;
689. -moz-transition: none; */\* Firefox is jumpy, so disable transitions \*/*
690. }
692. .anvil-dropdown select.form-control {
694. -webkit-appearance: none;
695. -moz-appearance: none;
696. **padding**: 8px 2em 4px 0;
698. */\* ew ew ew - select elements are weird and don't obey line-height, so we just*
699. *set the height explicitly. Ugh. \*/*
700. **height**: calc(1.5em + 13px); */\* 8px + 4px padding + 1px border = 13px \*/*
701. }
703. .anvil-dropdown {
704. **position**: relative;
705. }
706. */\* Select arrow styling \*/*
707. .anvil-dropdown form:before {
708. **content**: "**\25BC**";
709. **position**: absolute;
710. **right**: 0;
711. **font-size**: 80%;
712. **line-height**: calc(1.8em + 8px);
713. **color**: #555;
714. **pointer-events**:none;
715. **padding**: 4px 4px;
716. }
718. .anvil-component select, .anvil-datepicker input { **margin-bottom**: 4px; }
719. input.anvil-component.anvil-spacing-below-none { **margin-bottom**: 4px; }
720. input.anvil-component.anvil-spacing-below-small { **margin-bottom**: 8px; }
721. input.anvil-component.anvil-spacing-below-medium { **margin-bottom**: 12px; }
722. input.anvil-component.anvil-spacing-below-large { **margin-bottom**: 20px; }
724. input.anvil-component::-webkit-input-placeholder, textarea.anvil-component::-webkit-input-placeholder {
725. **color**: rgba(0,0,0,0.54);
726. }
727. input.anvil-component::-moz-placeholder, textarea.anvil-component::-moz-placeholder {
728. **color**: rgba(0,0,0,0.54);
729. }
730. input.anvil-component::placeholder, textarea.anvil-component::placeholder  {
731. **color**: rgba(0,0,0,0.54);
732. }
734. .app-bar input.anvil-component::-webkit-input-placeholder, .app-bar textarea.anvil-component::-webkit-input-placeholder {
735. **color**: rgba(255,255,255,0.8);
736. }
737. .app-bar input.anvil-component::-moz-placeholder, .app-bar textarea.anvil-component::-moz-placeholder {
738. **color**: rgba(255,255,255,0.8);
739. }
740. .app-bar input.anvil-component::placeholder, .app-bar textarea.anvil-component::placeholder  {
741. **color**: rgba(255,255,255,0.8);
742. }
744. input.anvil-component:hover, .anvil-component select:hover, .anvil-datepicker input:hover {
745. **border-bottom**: 2px solid rgba(0,0,0,.87);
746. **padding-bottom**: 3px;
747. }
749. input.anvil-component:focus, .anvil-component select:focus, .anvil-datepicker input:focus {
750. **border-bottom**: 2px solid %color:Primary 700%;
751. **padding-bottom**: 3px;
752. **box-shadow**: none;
753. }
755. input.anvil-component[disabled], .anvil-component select[disabled], .anvil-datepicker input[disabled] {
756. **border-bottom**: 1px dashed #888;
757. **padding-bottom**: 4px;
758. **background-color**: transparent;
759. }
761. textarea.anvil-component:hover {
762. **border**: 2px solid rgba(0,0,0,0.87);
763. **padding**: 3px 7px;
764. }
766. textarea.anvil-component:focus {
767. **border**: 2px solid %color:Primary 700%;
768. **padding**: 3px 7px;
769. **box-shadow**: none;
770. }
772. textarea.anvil-component[disabled] {
773. **border**: 1px dashed #888;
774. **padding**: 4px 8px;
775. **background-color**: transparent;
776. }
778. .daterangepicker td.active {
779. **background-color**: %color:Primary 500%;
780. }
782. .daterangepicker td.active:hover {
783. **background-color**: %color:Primary 700%;
784. }
786. .daterangepicker .btn-success {
787. **color**: %color:Primary 500%;
788. }
790. */\* Component: CheckBox, RadioButton \*/*
791. */\* Ugh. These are definitely TODO right now. \*/*
793. .checkbox, .radio {
794. **font-size**: 16px;
795. }
797. */\* Shadow values taken from*
798. *https://stackoverflow.com/questions/30533055/calculating-shadow-values-for-all-material-design-elevations*
799. *\*/*

A continuación se explica el código de Python programado en el cliente, llamado peek\_frontend, para la retención del trazo y la interacción con el servidor.

---------------------------------

peek\_frontend.py

---------------------------------

1. **from** .\_anvil\_designer **import** PEEK\_FrontEndTemplate
2. **from** anvil **import** \*
3. **import** anvil.server
5. **class** PEEK\_FrontEnd(PEEK\_FrontEndTemplate):
6. **def** \_\_init\_\_(self, \*\*properties):
7. *# Se definen las propiedades del Form de PÉEK*
8. self.init\_components(\*\*properties)

11. *############################## FUNCTION SECTION #####################*
12. **def** arrhyt\_class\_click(self, \*\*event\_args):
13. """Se activa cuando el botón es presionado"""
14. self.Class\_RowPanel.visible=True
15. self.Segm\_Panel.visible=False
16. self.AlgEval\_Panel.visible=False

19. **def** segment\_loc\_click(self, \*\*event\_args):
20. """Se activa cuando el botón es presionado"""
21. self.Class\_RowPanel.visible=False
22. self.Segm\_Panel.visible=True
23. self.AlgEval\_Panel.visible=False
25. **def** algorithm\_eval\_click(self, \*\*event\_args):
26. """Se activa cuando el botón es presionado"""
27. self.Class\_RowPanel.visible=False
28. self.Segm\_Panel.visible=False
29. self.AlgEval\_Panel.visible=True
31. *######################################################################*

34. *## Si el botón UPLOAD ECG  es presionado, se manda la información para Frecuencia de muestreo*
35. *## y el trazo hacia el servidor mediante el llamado de Anvil hacia la función Upload\_ECG.*
36. *## regresando como respuesta el trazo ECG completo en una imagen con renglones del trazo*
37. *## cada 20 segundos*
38. **def** button\_classifier\_copy\_click(self, \*\*event\_args):
39. *# Si el selector se encuentra en Frecuencia de muestreo, se manda la información del cuadro*
40. *# de texto en Frecuencia de muestreo*
41. **if** (self.FreqM.selected==True):
42. FS=self.text\_box\_FS.text
43. FST=1
44. *# Si el selector se encuentra en Duración del trazo, se manda la información del cuadro*
45. *# de texto en tiempo en segundos*
46. **if** (self.TimeL.selected==True):
47. FS=self.text\_box\_FS.text
48. FST=2
50. *## El archivo seleccionado se almacena en ECG\_ORIG y se manda a llamar la función Upload\_ECG*
51. *## del servidor, enviando el archivo multimedia y las variables FS y FST*
52. ECG\_ORIG=self.file\_loader\_1.file
53. self.original\_ecg.source = anvil.server.call('Upload\_ECG',
54. ECG\_ORIG,FS,FST)
55. self.original\_ecg.visible=True

58. *## Si el botón Find arrhytmia es presionado, se envía la misma información que en la función*
59. *## Upload ECG, pero regresa el trazo con marcas en cada ciclo cardíaco según como fue clasificado.*
60. *## Este sería el diagóstico generado por PÉEK*
61. **def** button\_classifier\_click(self, \*\*event\_args):
62. *# Si el selector se encuentra en Frecuencia de muestreo, se manda la información del cuadro*
63. *# de texto en Frecuencia de muestreo*
64. **if** (self.FreqM.selected==True):
65. FS=self.text\_box\_FS.text
66. FST=1
67. *# Si el selector se encuentra en Duración del trazo, se manda la información del cuadro*
68. *# de texto en tiempo en segundos*
69. **if** (self.TimeL.selected==True):
70. FS=self.text\_box\_FS.text
71. FST=2
73. *## El archivo seleccionado se almacena en ECG\_ORIG y se manda a llamar la función arrhyt\_classifier*
74. *## del servidor, enviando el archivo multimedia y las variables FS y FST*
75. ECG\_ORIG=self.file\_loader\_1.file
76. self.arrhytmia\_ecg.source = anvil.server.call('arrhyt\_classifier',
77. ECG\_ORIG,FS,FST)
78. self.arrhytmia\_ecg.visible=True

Aquí se muestra el procesamiento del trazo desde el servidor Python con una explicación de sus componentes.

---------------------------------

peek\_backend.py

---------------------------------

1. *# -\*- coding: utf-8 -\*-*
2. """PEEK\_BackEnd.ipynb
4. Automatically generated by Colaboratory.
6. Original file is located at
7. https://colab.research.google.com/drive/1jsRBJDOPqXYc9n1qlBiNmftHj2szUWVH
9. # Librerías
10. """
12. *# Commented out IPython magic to ensure Python compatibility.*
13. *###### Importación de librerías necesarias para el uso de PÉEK   #######*
14. !ln -sf /opt/bin/nvidia-smi /usr/bin/nvidia-smi
15. !pip install gputil *# GPUtil es un módulo de Python para obtener el estado de la GPU de NVIDA utilizando nvidia-smi*
16. !pip install psutil *# psutil es una módulo para recuperar información sobre los procesos en ejecución y la utilización*
17. *# del sistema (CPU, memoria, discos, red, sensores) en Python.*
18. !pip install py-ecg-detectors *# ECG Detectors es un módulo con funciones para el reconocimiento de picos R en un ECG*
19. *# basados en los métodos de detección den mayor porcentaje y probados en la base de datos*
20. *# MIT-BIH*
22. *# Utilidades*
23. **import** time
24. **import** psutil
25. *# %matplotlib inline*
27. *# Funciones para el procesamiento de la imagen y arreglos numéricos*
28. **import** io
29. **from** PIL **import** Image
30. **import** numpy **as** np
31. **import** pandas **as** pd
32. **import** matplotlib.pyplot **as** plt
33. **import** os
34. **import** tensorflow **as** tf
35. **import** cv2
36. **import** cv2 **as** cv
38. *# Funciones requeridas para el manejo del modelo y la imagen*
39. **import** keras
40. **from** keras.preprocessing **import** image
41. **from** keras.applications.imagenet\_utils **import** preprocess\_input
42. **from** keras.models **import** Sequential
43. **from** keras.layers **import** Dense, Dropout, Flatten, Activation
44. **from** keras.layers **import** Conv2D, MaxPooling2D
45. **from** keras.models **import** Model
46. **from** keras **import** utils **as** np\_utils
48. *# Función para*
49. **from** ecgdetectors **import** Detectors
51. *# Librería y funciones requeridas para el manejo de medios y conexión con Anvil*
52. !pip install anvil-uplink
53. **import** anvil.server
54. **import** anvil.media
55. **import** anvil.tables **as** tables
56. **from** anvil.tables **import** app\_tables
57. **import** anvil.media
59. """# Obtener categorías"""
61. !ls /content/drive/MyDrive/AlexanderTesis/2021-1/Datasets/Image/Derivations/MLII
63. root = '/content/drive/MyDrive/AlexanderTesis/2021-1/Datasets/Image/Derivations/MLII'
64. exclude = ['Prueba']
65. train\_split, val\_split = 0.7, 0.15
67. categories = [x[0] **for** x **in** os.walk(root) **if** x[0]][1:]
68. categories = [c **for** c **in** categories **if** c **not** **in** [os.path.join(root, e) **for** e **in** exclude]]
70. **print**(categories)
72. """# Conexión con Interfaz Gráfica"""
74. *# Conexión cliente-servidor mediante el Anvil-Uplink ID*
75. anvil.server.connect("G4SKKTLX4IJI2P45FWMBJK3J-7I6IYR6U2OJCTQUL")
77. """# Obtener modelo"""
79. *# Se clonan los parámetros y pesos del modelo previamente entrenado y almacenado en una dirección*
80. model = keras.models.load\_model('/content/drive/MyDrive/AlexanderTesis/2021-1/Datasets/model210821.h5')
82. """# Procesamiento"""
84. *####### Cargar imagen de ECG ############*
86. @anvil.server.callable    *# Esta línea define a la función como "llamable por Anvil"*
87. **def** Upload\_ECG(orECG, FqM, FST):
89. *#Transforma el archivo tipo multimedia de Anvil en formato CSV para leerlo con Pandas*
90. **with** anvil.media.TempFile(orECG) **as** f:
91. ECGv = pd.read\_csv(f) *# Vector a partir del archivo CSV*
93. *# Si FST = 1, significa que la variable FqM se trata de ua Frecuencia de muestreo; de lo contrario,*
94. *# se calcula la Frecuencia de muestreo en base a la duración del trazo ECG y el número de muestras.*
95. **if** FST == 1:
96. FqM=int(FqM)
97. **if** FST == 2:
98. FqM=int(FqM)/len(ECGv)
100. *# Se multiplica la Frecuencia de muestreo por 20 para obtener el total de muestras en 20 segundos.*
101. TwSec=FqM\*20
102. t=np.arange(TwSec)
104. *# Se calcula el número de renglones que tendrá la imagen del reporte ECG que se mostrará en pantalla.*
105. Rows=int(len(ECGv)/TwSec)
106. Wrow=np.arange(Rows)
108. *### Separar el trazo por segmentos de 20 segundos y mostrarlo en un plot ###*
109. **for** j **in** Wrow:
110. ECGf = plt.figure()
111. ECGf, axes = plt.subplots(figsize=(120,6))
112. plt.axis('off')
113. axes.plot(t,ECGv[(j\*TwSec):((j+1)\*TwSec)])
115. *# Eliminar los espacios blancos del plot*
116. plt.gca().set\_axis\_off()
117. plt.subplots\_adjust(top = 1, bottom = 0, right = 1, left = 0, hspace = 0, wspace = 0)
118. plt.margins(0,0)
119. plt.gca().xaxis.set\_major\_locator(plt.NullLocator())
120. plt.gca().yaxis.set\_major\_locator(plt.NullLocator())
122. *# Guardar plot en imagen y eliminar plot para la liberar RAM utilizada*
123. ECGf.savefig('/content/Ventana\_'+str(j)+'.png',bbox\_inches='tight', pad\_inches=0)
124. ECGf.clf()
125. plt.clf()
126. plt.cla()
127. plt.close()
128. ECGf=plt.close()

131. *# Generar imagen del último renglón/segmento del ECG con una linea recta en el espacio vacío*
132. ECGtail=(ECGv[(Rows\*TwSec):(len(ECGv))]).to\_numpy()
133. ECGtail=np.transpose(ECGtail)[0]
134. DTz=np.zeros((TwSec-((len(ECGv))-(Rows\*TwSec))))
135. DTz=DTz+np.mean(ECGtail)
136. ECGwhite=np.concatenate((ECGtail,DTz))
137. ECGf = plt.figure()
138. ECGf, axes = plt.subplots(figsize=(120,6))
139. plt.axis('off')
140. axes.plot(t,ECGwhite)
142. *# Eliminar los espacios blancos del plot*
143. plt.gca().set\_axis\_off()
144. plt.subplots\_adjust(top = 1, bottom = 0, right = 1, left = 0, hspace = 0, wspace = 0)
145. plt.margins(0,0)
146. plt.gca().xaxis.set\_major\_locator(plt.NullLocator())
147. plt.gca().yaxis.set\_major\_locator(plt.NullLocator())
149. *# Guardar plot en imagen y liberar RAM*
150. ECGf.savefig('/content/Ventana\_'+str(Rows)+'.png',bbox\_inches='tight', pad\_inches=0)
151. ECGf.clf()
152. plt.clf()
153. plt.cla()
154. plt.close()
155. ECGf=plt.close()
157. *## Comienza la concatenación de renglones del trazo ECG ##*
158. ECGor = cv2.imread('/content/Ventana\_0.png')
159. **for** k **in** (Wrow):
160. ECGadd = cv2.imread('/content/Ventana\_'+str(k+1)+'.png')
161. ECGor = cv2.vconcat([ECGor,ECGadd])
163. *# Tranformación del formaro array a imagen*
164. ECGor = Image.fromarray(ECGor, 'RGB')
166. *# Se almacena y se transforma a archivo multimedia Anvil como respuesta a la función Upload\_ECG*
167. ECGor.save('/content/ECG\_orig.png')
168. img10 = anvil.media.from\_file('/content/ECG\_orig.png', "image/png")
170. **return**  img10
172. *####### ARRHYTMIA CLASSIFIER ############*
174. @anvil.server.callable
175. **def** arrhyt\_classifier(orECG, FqM, FST):
177. *# Obtiene el tiempo inicial antes del procesamiento de la imagen*
178. tmpo = time.time()
180. *#Transforma el archivo tipo multimedia de Anvil en formato CSV para leerlo con Pandas*
181. **with** anvil.media.TempFile(orECG) **as** f:
182. ECGv = pd.read\_csv(f)
184. *# Si FST = 1, significa que la variable FqM se trata de ua Frecuencia de muestreo; de lo contrario,*
185. *# se calcula la Frecuencia de muestreo en base a la duración del trazo ECG y el número de muestras.*
186. **if** FST == 1:
187. FqM=int(FqM)
188. **if** FST == 2:
189. FqM=int(FqM)/len(ECGv)
191. *# Se multiplica la Frecuencia de muestreo por 20 para obtener el total de muestras en 20 segundos.*
192. TwSec=FqM\*20
193. t=np.arange(TwSec)
195. *# Se calcula el número de renglones que tendrá la imagen del reporte ECG que se mostrará en pantalla.*
196. Rows=int(len(ECGv)/TwSec)
197. Wrow=np.arange(Rows)
199. *# Vector de tiempo para 298 píxeles de imagen*
200. t2=np.arange(298)
202. *# Se transforma el arreglo de ECG a vector de 1 dimensión*
203. A = np.squeeze(np.asarray(ECGv))
205. *# Se define la frecuencia de muestreo a utilizar con el detector de picos R*
206. detectors = Detectors(FqM)
208. *# Se extraen los picos R identificados en el trazo de ECG*
209. r\_peaks = detectors.swt\_detector(A)
211. *# Se crea arreglo de cantidad de categorías (3) por cantidad de picos identificados*
212. p1=np.zeros((int(len(r\_peaks)-1),len(categories)))
214. *# Vector de caracteres con el tamaño de la cantidad de picos identificados*
215. p2 = ["" **for** i **in** range(len(r\_peaks)-2)]
217. *# Vector numérico desde el 0 hasta el número de picos identificados menos 2*
218. p3=np.array(range(int((len(r\_peaks)-2))))
220. *#Recortar la imagen y recorrerla de inicio a fin*
221. **for** i **in** (p3):
222. ECGs = plt.figure()
223. ECGs, axes = plt.subplots(figsize=(2.99,2.99))
224. plt.axis('off')
225. axes.plot(t2,A[(r\_peaks[i]-120):(r\_peaks[i]+178)])
226. ECGs.savefig('/content/Prf.png',bbox\_inches='tight', pad\_inches=0)
227. img = image.load\_img('/content/Prf.png', target\_size=(299, 299))
228. ECGs.clf()
229. plt.clf()
230. plt.cla()
231. plt.close()
232. ECGs=plt.close()
233. x = np.expand\_dims(img, axis=0)
234. x = preprocess\_input(x)
236. *# Un ciclo cardiaco pasa por el modelo para clasificarlo entre alguna de las categorías*
237. probabilities\_old = model.predict([x])
239. *# Se almacena en una posición del vector p1*
240. p1[i]=probabilities\_old
242. *# Se identifica la categoría con mayor probabilidad*
243. mx=max(probabilities\_old[0])
245. *# Se almacena el resultado de la categoría a la que pertenece el ciclo cardiaco*
246. **if** probabilities\_old[0,0] == mx:
247. **print**("Normal Sinus Rythm")
248. Arr="Normal Sinus Rythm"
249. p2[i]="NSR"
250. **elif** probabilities\_old[0,1] == mx:
251. **print**("Premature Ventricular Contraction")
252. Arr="Premature Ventricular Contraction"
253. p2[i]="PVC"
254. **elif** probabilities\_old[0,2] == mx:
255. **print**("Other")
256. Arr="Other"
257. p2[i]="Other"
258. **else**:
259. **print**("We dont find it")
260. Arr="We dont find it"
262. *# Parámetros iniciales para el recorrido de los ciclos*
263. p2v=np.array(range(len(p2)))
264. j=0
265. i=0
267. *# Se inicia con el primer renglon del trazo ECG almacenado previamente*
268. img\_1=cv2.imread('/content/Ventana\_'+str(j)+'.png')
269. (h,w,c)=img\_1.shape[:3]
271. *# Comienza el recorrido de ciclos para marcarlos en las imágenes de 20 segundos según su categoría*
272. **for** i **in** p2v:     *# Se recorre cada posición de pico R identificada*
273. **if** r\_peaks[(i+1)] < ((j+1)\*TwSec):  *# Mientras no se rebasen los 20 segundos, se hará el recorrido en el renglón actual*
275. **if** p2[i]=='NSR':  *# Si el ciclo actual fue clasificado como normal, marcar el texto "NSR" en la imagen*
276. img\_2 = cv2.putText(img\_1, 'NSR', (int((r\_peaks[(i+1)]-(j\*TwSec))/(TwSec/w)),int(1\*h/8)),cv2.FONT\_HERSHEY\_COMPLEX , 1.5, (0, 0, 255), 2)
278. **if** p2[i]=='PVC':  *# Si el ciclo fue clasificado como PVC, marcar un recuadro en la imagen con el texto PVC*
279. img\_2 = cv2.rectangle(img\_1,(int((r\_peaks[(i+1)]-(j\*TwSec)-120)/(TwSec/w)),1),(int((r\_peaks[(i+1)]-(j\*TwSec)+178)/(TwSec/w)),int(4\*h/4)),(0,0,255),2)
280. img\_2 = cv2.putText(img\_1, 'PVC', (int((r\_peaks[(i+1)]-(j\*TwSec))/(TwSec/w)),int(1\*h/8)),cv2.FONT\_HERSHEY\_COMPLEX , 1.5, (0, 0, 255), 2)
282. **if** p2[i]=='Other':  *# Si el ciclo fue clasificado como "Other" (otro), marcar un recuadro en la imagen con el texto Other*
283. img\_2 = cv2.rectangle(img\_1,(int((r\_peaks[(i+1)]-(j\*TwSec)-120)/(TwSec/w)),1),(int((r\_peaks[(i+1)]-(j\*TwSec)+178)/(TwSec/w)),int(4\*h/4)),(0,0,255),2)
284. img\_2 = cv2.putText(img\_1, 'Other', (int((r\_peaks[(i+1)]-(j\*TwSec))/(TwSec/w)),int(1\*h/8)),cv2.FONT\_HERSHEY\_COMPLEX , 1.5, (0, 0, 255), 2)
286. **else**:     *# Si se superaron los 20 segundos, pasar al siguiente renglón y comenzar nuevamente el recorrido*
287. img\_3 = Image.fromarray(img\_2, 'RGB')
288. img\_3.save('/content/ECG\_Arr\_'+str(j)+'.png')
289. j=j+1
290. img\_1=cv2.imread('/content/Ventana\_'+str(j)+'.png')
291. (h,w,c)=img\_1.shape[:3]
293. **if** p2[i]=='NSR':  *# Si el ciclo actual fue clasificado como normal, marcar el texto "NSR" en la imagen*
294. img\_2 = cv2.putText(img\_1, 'NSR', (int((r\_peaks[(i+1)]-(j\*TwSec))/(TwSec/w)),int(1\*h/8)),cv2.FONT\_HERSHEY\_COMPLEX , 1.5, (0, 0, 255), 2)
296. **if** p2[i]=='PVC':  *# Si el ciclo fue clasificado como PVC, marcar un recuadro en la imagen con el texto PVC*
297. img\_2 = cv2.rectangle(img\_1,(int((r\_peaks[(i+1)]-(j\*TwSec)-120)/(TwSec/w)),1),(int((r\_peaks[(i+1)]-(j\*TwSec)+178)/(TwSec/w)),int(4\*h/4)),(0,0,255),2)
298. img\_2 = cv2.putText(img\_1, 'PVC', (int((r\_peaks[(i+1)]-(j\*TwSec))/(TwSec/w)),int(1\*h/8)),cv2.FONT\_HERSHEY\_COMPLEX , 1.5, (0, 0, 255), 2)
300. **if** p2[i]=='Other':  *# Si el ciclo fue clasificado como "Other" (otro), marcar un recuadro en la imagen con el texto Other*
301. img\_2 = cv2.rectangle(img\_1,(int((r\_peaks[(i+1)]-(j\*TwSec)-120)/(TwSec/w)),1),(int((r\_peaks[(i+1)]-(j\*TwSec)+178)/(TwSec/w)),int(4\*h/4)),(0,0,255),2)
302. img\_2 = cv2.putText(img\_1, 'Other', (int((r\_peaks[(i+1)]-(j\*TwSec))/(TwSec/w)),int(1\*h/8)),cv2.FONT\_HERSHEY\_COMPLEX , 1.5, (0, 0, 255), 2)
304. *# Pasar la imagen de su formato tipo arreglo a formato de imagen y almacenarla en PNG*
305. img\_3 = Image.fromarray(img\_2, 'RGB')
306. img\_3.save('/content/ECG\_Arr\_'+str(Rows)+'.png')
308. *# Comienza proceso de eliminación de imágenes creadas para vaciar el almacenamiento consumido*
309. ECGdiag = cv2.imread('/content/ECG\_Arr\_0.png')
310. os.remove('/content/ECG\_Arr\_0.png')
311. os.remove('/content/Ventana\_0.png')
313. *# Antes de eliminar als imágenes de 20 segundos, se unen de forma vertical para crear una imagen*
314. *# completa con el trazo completo y las marcas según la clasificación de cada ciclo*
315. **for** k **in** (Wrow):
316. ECGadd = cv2.imread('/content/ECG\_Arr\_'+str(k+1)+'.png')
317. ECGdiag = cv2.vconcat([ECGdiag,ECGadd])
318. os.remove('/content/ECG\_Arr\_'+str(k+1)+'.png')
319. os.remove('/content/Ventana\_'+str(k+1)+'.png')
321. *# Pasar la imagen completa del diagnóstico de su formato tipo arreglo a formato de imagen y almacenarla en PNG*
322. ECGdiag = Image.fromarray(ECGdiag, 'RGB')
323. ECGdiag.save('/content/ECG\_diag.png')
325. *# Se almacena y se transforma a archivo multimedia Anvil como respuesta a la función Arrhyt\_classifier*
326. imgarr = anvil.media.from\_file('/content/ECG\_diag.png', "image/png")
327. **print**(time.time() - tmpo)
329. **return**  imgarr
331. *# Función de Anvil mantiene al servidor en un ciclo infinito, esperando la respuesta contínua de*
332. *# la interfaz web*
333. anvil.server.wait\_forever()