

Music Picker

Para empezar, nuestro proyecto es un reproductor de música según el pulso, ritmo cardiaco, más específicamente en BPM.

Nuestro objetivo principal era que reprodujera música según las emociones, asociando el pulso a las emociones, algo que logramos.

¿Qué utilizamos en nuestro proyecto?

Los elementos principales que utilizamos son:

Como base utilizamos una Raspberry Pi Pico 2 W, proporcionada por la Universidad. Para medir el pulso utilizamos un sensor de pulso MAX30100 comprado por nosotros. Para conectar la raspberry utilizamos un cable USB – microUSB, y para la conexión entre la Raspberry y el sensor utilizamos cables dupont macho-macho, y una breadboard.

Los programas que utilizamos son: Github para subir el proyecto, avances, pruebas, la documentación y subir la presentación final del proyecto. Visual estudio para editar en Github. Thonny, para programar la raspberry pi y el sensor. API de Spotify para reproducir música. Render para hacer el servidor que conecta la API de Spotify y la raspberry.

¿Cómo funciona nuestro reproductor de música?

Para empezar, la Raspberry Pi Pico 2 W se conecta al sensor de pulso. A través de Render; una página que crea servidores de manera gratuita; conectamos la API de Spotify, la cual fue configurada anteriormente. En Render, subimos un repositorio con la configuración y con todas las variables necesarias que necesitaremos para hacer sonar música en Spotify, ya sean estas; canciones que queremos que se reproduzcan y los intervalos de pulso para cada emoción. Conectamos el repositorio de Github con Render, y una vez funcionando el servidor, ya puede recibir pulso y mandar la orden a la API de Spotify para que reproduzca música. Por otro lado, una vez el servidor está abierto, el sensor debe empezar a tomar datos (función configurada anteriormente), y enviar estos datos en BPM al servidor. Una vez enviado el pulso al servidor, este asocia el pulso a una emoción, y envía la orden de reproducir una canción asociada a la emoción correspondiente. Por el momento, para que Spotify reproduzca música debe estar abierto en algún dispositivo, recibe la orden desde la API y reproduce la canción.

Paso a paso de nuestro trabajo.

Lo primero que hicimos al comenzar nuestro proyecto fue configurar la raspberry. Al principio hicimos algunas pruebas, como conectarla a internet. Dentro de las pruebas que

hicimos, estuvo el crear un html para comprobar que la raspberry estuviera conectada. Dentro de esta web podíamos encender y apagar el led de la raspi.

Luego de esto comenzamos con el proceso más difícil dentro de nuestra investigación, encontrar una manera de tomar el pulso, y que este lograra reproducir música en Spotify.

Consideramos diferentes opciones, dentro de estas:

Primera: tomar el pulso con un reloj inteligente y a través de Bluetooth enviar el pulso a la raspberry y que esta reprodujera música.

Segunda: Tomar el pulso con una aplicación de celular que toma el pulso con la cámara, y a partir de esto enviar los datos a la raspberry.

Tercera: Utilizar un sensor de pulso, que este tome el pulso y se envíe a la raspberry.

De estas tres opciones, la que utilizamos fue la tercera ya que las dos anteriores nos presentaron distintos problemas. Por una parte, tomar el pulso con un reloj inteligente no fue factible ya que, dependiendo del tipo de reloj, estos están diseñados para comunicarse solo con sus propias apps, y los datos como el pulso están codificados. Para acceder a estos datos desde la raspi se habría requerido crear una app que imitara la oficial, lo cual para nuestro nivel de conocimiento resultaba difícil y poco práctico.

Para la segunda opción, tomar el sensor desde una aplicación de celular nos ocurrió algo muy similar, no nos era posible enviar los datos a la raspberry, y la aplicación no era lo suficientemente precisa.

Finalmente, nuestra opción final fue comprar un sensor de pulso.

Al investigar un poco sobre sensores, encontramos que el MAX30100 era compatible con la raspberry y dentro de los sensores que lo son, es uno de los más fáciles de programar, ya que este también se puede programar en Python.

Ya teniendo claro que utilizaríamos un sensor para tomar el pulso, empezamos la siguiente fase de nuestra investigación, como reproducir música.

Dentro de las primeras opciones siempre tuvimos el conectarnos a la API de Spotify, lo que al comienzo nos trajo dificultades, ya que para comunicarse con la API es necesario conectarse a través de https por su seguridad, y según esto, la memoria y la capacidad de procesamiento de la raspberry pi pico 2 w son muy limitadas para manejar conexiones web seguras. Así que empezamos a ver otras opciones, entre estas, conectar la API a un servidor utilizando como puente un computador que se conectara a la raspberry, una opción segura al comienzo, pero luego de tener algunos problemas en un computador por su seguridad, y no poder conectar al servidor, descartamos nuevamente esta idea. Si bien, era una buena opción, pero si utilizábamos este servidor dentro de un solo dispositivo, no podríamos conectarnos desde otro dispositivo, solo podríamos hacerlo funcionar en el dispositivo que tuviera el servidor, lo cual limitaba demasiado el objetivo de nuestro proyecto.

Descartando estas dos ideas, rescatamos algo de la segunda, el servidor, y empezamos a investigar como hacer un servidor, encontrando así, la opción de hacer un servidor en la nube. Aunque al principio esta también nos trajo problemas, pues para hacer un servidor en la nube estos se compran, encontramos paginas seguras que crean servidores gratis, con ciertas limitaciones, pero para lo que lo utilizaríamos nos funcionaba bien. Así fue como encontramos Render.

Antes de entrar a Render, lo primero que tuvimos que hacer fue comunicarnos con la API de Spotify, a través de una pagina web especial de la aplicación. Una vez teniendo la API, la cual se obtiene de una cuenta premium, podíamos empezar a trabajar con el servidor.

Cabe recalcar que, ya que no teníamos aun el sensor, creamos una pagina web dentro de la raspberry que podía simular el pulso, así fue como tuvimos un primer avance.

Al comienzo utilizamos el servidor con un computador, aun sin tener el servidor en la nube, conectamos la API a este servidor, y con el simulador de pulso logramos que Spotify reprodujera música, pero como mencionamos anteriormente, este se limitaba a solo un dispositivo.

Empezamos a trabajar con el servidor en la nube, específicamente en Render. Para esto investigamos como utilizar render. Para crear el servidor, una de las opciones fue crear un repositorio en Github y en este subir los códigos y librerías necesarias para hacerlo funcionar. Dentro de las dificultades que tuvimos estuvo el hecho de que para conectar con la API de Spotify Render nos pedía el Client ID y Secret ID, datos entregados por la API de Spotify que nos daban el acceso a la API, por lo cual estos no se podían subir al repositorio, por lo cual tuvimos que utilizar gitignore, herramienta que hasta el momento era desconocida para nosotros.

Luego de crear el repositorio y vincularlo a Render, finalmente pudimos hacer funcionar el servidor. Dentro de lo que tenia el repositorio del servidor, estaba toda la configuración, tanto las canciones, como los intervalos de pulso designado a cada emoción, y las canciones designadas a cada canción. Por el momento, solo consideramos tres emociones, más bien, estados: Relajado, normal y agitado, a cada uno de estos le asignamos tres canciones.

Una vez recibimos el sensor que compramos, entramos en el proceso de configurarlo y conectarlo a la raspberry, lo conseguimos una vez, sin embargo no logramos configurarlo bien, y la segunda vez que lo intentamos, por un error de conexión nuestro sensor se quemó, lo que retrasó un poco el avance de nuestro proyecto.

Debido al incidente que tuvimos, la primera prueba que hicimos del servidor fue con la web que ya teníamos en la raspberry. En la web escribíamos bpm aleatorios, estos eran recibidos por la raspberry, esta enviaba los datos al servidor, y una vez recibido por el servidor, este le enviaba la orden a la API de Spotify para que reprodujera música. Objetivo logrado, por el momento.

Algo que no consideramos al momento de programar de esta manera, es que en la web nosotros enviábamos datos de a uno, pero al tener el sensor, este estaría constantemente

enviando datos, y por consecuencia, el servidor estaría recibiendo toda esta cantidad de datos muy seguido y las canciones no durarían ni siquiera un segundo en reproducción y se cambiaría.

Finalmente recibimos el nuevo sensor, y esta vez tuvimos el mayor de los cuidados al conectarlo y programarlo. Al principio fue muy difícil ya que el sensor nos imprimía muchos datos en poco tiempo, y luego de unas pruebas, logramos que imprimiera bpm. Para esto también le asignamos la función de hacer que cada 20 bpm nos diera un promedio de bpm, y este dato sería enviado al servidor. A todo esto, tuvimos que también modificar el código del servidor, para que funcionara de tal manera que al recibir el promedio reprodujera una canción, y no cambiara de canción hasta que la que se estaba reproduciendo terminara, y luego se reprodujera otra canción a partir del último promedio registrado.

Luego de unas cuantas pruebas para verificar que el pulso que nos daba el sensor fuera lo más preciso posible, finalmente hicimos funcionar el servidor y el sensor en sintonía, y finalmente nuestro reproductor de música según el pulso está funcional.

Objetivos futuros.

Ya que podríamos considerar que nuestro objetivo principal se cumplió, teniendo el reproductor de música según el pulso, aún tenemos algunos objetivos a futuro para nuestro proyecto. Comenzando por el hecho de que nos gustaría asociar el pulso a más emociones, a parte de las tres básicas que tenemos.

Otro de nuestros objetivos a futuro, es reproducir música de manera aleatoria, ya sea de alguna playlist en Spotify, o dejar que según el algoritmo de la cuenta de Spotify se reproduzca música, un objetivo grande y quizás algo para más a largo plazo, ya que estaríamos fusionando las emociones con los gustos personales, para así crear una experiencia más personalizada.

Y por último, que el proyecto funcione sin la necesidad de estar conectado a un computador como fuente de alimentación. Por el momento y con los conocimientos que contamos no nos es posible, pero consideramos que para esto se necesitarían muchas más mejoras. El objetivo principal sería armonizar su apariencia para que sea algo más llamativo. Para ello tendríamos en investigar como hacer funcionar la raspberry con otra fuente de alimentación (con el objetivo de que no esté conectada a una fuente estática como un computador o un enchufe), como hacer que los códigos dentro funcionen. Aunque esto último es totalmente para un futuro.

Prototipo.

Ya que nuestro proyecto en realidad es solo un prototipo de lo que vendría a ser un proyecto mucho más elaborado y mucho más preciso, no consideramos que sea relevante lo mencionado anteriormente. Nuestro objetivo ya está cumplido, logramos (con los

componentes propuestos de un inicio) reproducir música según el pulso cardíaco. Datos como: la precisión del sensor, las limitaciones del servidor o incluso lo estético son aspectos que podrían ser considerados en un futuro (como ya mencionamos).

Al ser un prototipo de lo que vendría a ser un gran proyecto, los materiales y elementos con los que trabajamos fueron lo suficientemente útiles para mostrar la funcionalidad del proyecto y su objetivo, y estamos bastante satisfechos con su resultado, si bien, los datos del sensor no son muy precisos, el servidor tiene limitaciones de tiempo, y la raspberry la tenemos que tener conectada a una fuente; el proyecto cumple con los objetivos.

Si en un futuro se nos da la oportunidad de mejorar significativamente el proyecto sin duda lo tomaríamos.