

APRENDIZAJE MUSICAL RÍTMICO CON RHYTHME, TUS PRIMEROS PASOS  
MUSICALES

DIANA PAOLA RINCÓN MONTAÑA

WILMER RODRÍGUEZ CAMARGO

UNIVERSIDAD ANTONIO NARIÑO

FACULTAD DE INGENIERÍA DE SISTEMAS

PROGRAMA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

BOGOTÁ

2020-2

APRENDIZAJE MUSICAL RÍTMICO CON RHYTHME, TUS PRIMEROS PASOS  
MUSICALES

DIANA PAOLA RINCÓN MONTAÑA

WILMER RODRÍGUEZ CAMARGO

TRABAJO DE GRADO PARA OPTAR POR EL TÍTULO DE  
INGENIERO DE SISTEMAS Y COMPUTACIÓN

DIRECTOR

Msc. WILSON JAVIER FORERO

UNIVERSIDAD ANTONIO NARIÑO

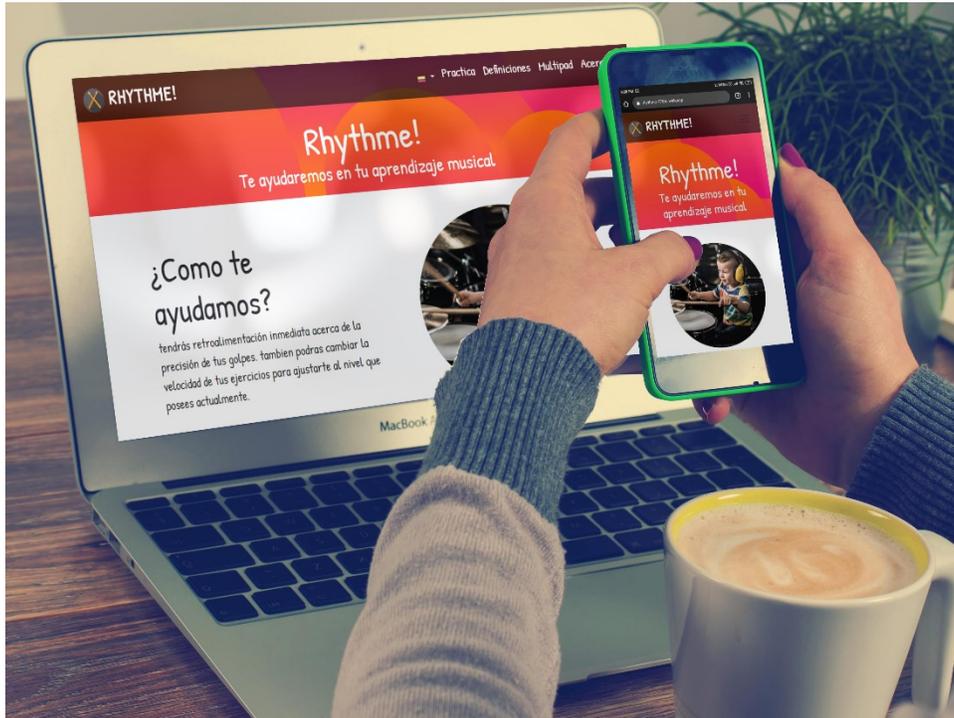
FACULTAD DE INGENIERÍA DE SISTEMAS

PROGRAMA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

BOGOTÁ, D.C.

2020-2

# MANUAL TÉCNICO



## APRENDIZAJE MUSICAL RÍTMICO CON RHYTHME, TUS PRIMEROS PASOS

### MUSICALES

Sitio web: <https://rhythme-776fc.web.app/>

## Tabla de contenido

	Pág.
1. Presentación.....	7
2. Objetivo.....	8
3. Herramientas utilizadas en el desarrollo .....	9
4. Composición del sitio web .....	11
4.1. Diagrama de paquetes .....	11
5. Jerarquía de componentes .....	12
5.1. Vista de despliegue .....	14
5.2. Mapa de navegación .....	15
6. Código Fuente.....	16
6.1. Obtener Código Fuente .....	16
6.2. Ramas.....	16
6.3. Compilación .....	17
6.4. Despliegue.....	17
7. Multipad .....	18

7.1.	Multipad basado en teclado .....	18
7.2.	Multipad Arduino .....	18
7.3.	Placas base compatibles .....	18
7.4.	Conexión del Arduino .....	19
7.5.	Diagrama de conexión de componentes electrónicos .....	19
8.	Resolución de problemas.....	20
8.1.	Sitio web.....	20
8.2.	Arduino .....	20
	Bibliografía .....	21

## Listado de figuras

	Pág.
Figura 1. Diagrama de paquetes.....	11
Figura 2. Jerarquía de componentes .....	12
Figura 3. Diagrama de componentes .....	13
Figura 4. Diagrama vista de despliegue.....	14
Figura 5. Diagrama mapa de navegación.....	15
Figura 6. Diagrama conexión de componentes.....	19

## **1. Presentación**

El siguiente manual guiará a los usuarios que harán soporte al sistema, el cual les dará a conocer los requerimientos y la estructura para la construcción del sistema, mostrando las herramientas necesarias para el soporte y desarrollo del sitio web.

## **2. Objetivo**

El presente manual tiene como objetivo informar, explicar y orientar al desarrollador la estructura, herramientas y arquitectura, para que se pueda realizar el soporte, futuras actualizaciones y desarrollo sobre el sitio web Rhythme.

### 3. Herramientas utilizadas en el desarrollo

- **JavaScript:** lenguaje de programación basado en prototipos orientado a objetos débilmente tipado y dinámico (EcuRed, 2020).
- **React:** biblioteca de desarrollo JavaScript diseñada para facilitar el desarrollo de aplicaciones de una sola página manejando solamente la interfaz de usuario basada en componentes.
- **Node.js:** entorno de ejecución basado en Javascript para la capa de servidor desarrollado a partir del motor V8 de Google.
- **Redux:** biblioteca de desarrollo JavaScript para el manejo del estado de la aplicación como una única fuente de verdad con un estado global e inmutable. Con esta librería se controla el flujo de la información de una manera sencilla y controlada.
- **Reactstrap:** librería de desarrollo que permite usar Bootstrap dentro de componentes React como componentes reutilizables.
- **React-fontawesome:** librería de componentes utilizables dentro de componentes react para la presentación de iconos SVG fontawesome.
- **React-router:** librería de desarrollo que permite una navegación fluida entre componentes creados en react y ser asociados a una URL.
- **Phaser.io:** framework 2D usado para crear juegos en HTML5, usando Canvas y WebGL para el renderizado y JavaScript/TypeScript como lenguaje de programación.

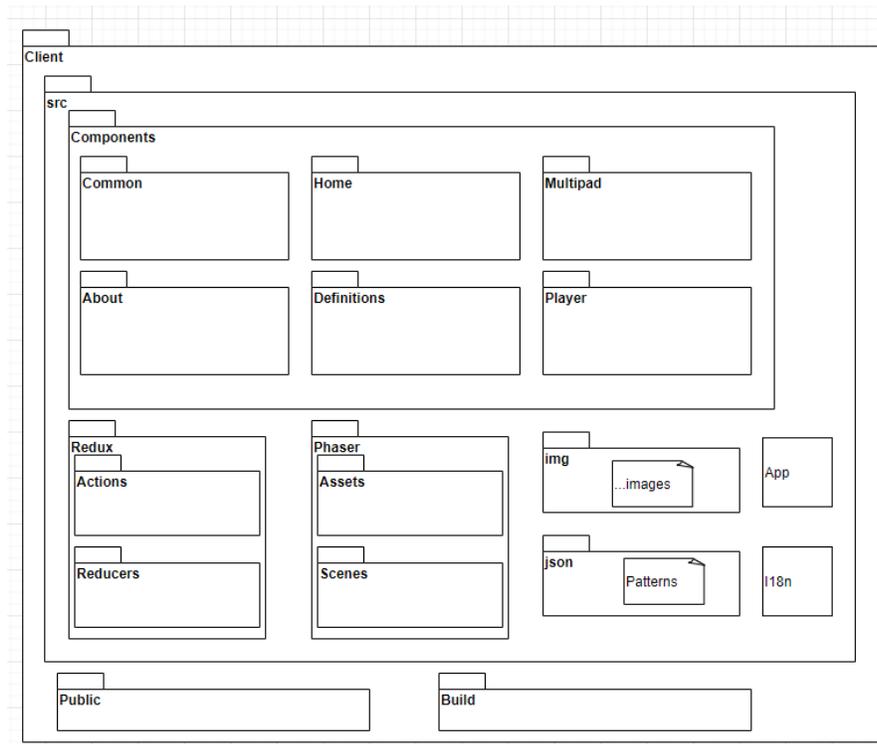
- **Firestore:** plataforma para el desarrollo de aplicaciones web y móviles desarrollada por Google. Esta plataforma permite almacenar las aplicaciones web, las bases de datos y las funciones en la nube, por lo que no se requiere un servidor local para el desarrollo y soporte de la aplicación.
- **Selenium-Katalon:** es una biblioteca de pruebas de software que provee las características de grabar y reproducir sin la necesidad de utilizar lenguajes de scripting para pruebas. Esta biblioteca usando el IDE Katalon, facilita el manejo la creación y ejecución de pruebas funcionales automatizadas puesto que cuenta con una interfaz amigable y de fácil uso, permitiendo así obtener validaciones del correcto funcionamiento del sistema. De igual manera, se logra identificar las oportunidades de mejora y fallas que presente el software.
- **Visual Studio Code:** IDE utilizado para este desarrollo debido a su gran versatilidad y facilidad de uso.
- **GitHub:** al utilizar el control de versiones git se escogió este sitio para llevar el control de versiones remoto debido a su amplio uso.
- **Arduino IDE:** este entorno de desarrollo se utiliza para realizar la programación de nuestra placa base Arduino, compilar el código y subirlo a la placa base.

## 4. Composición del sitio web

### 4.1. Diagrama de paquetes

El sitio web fue construido utilizando React como el framework principal, las librerías utilizadas y descritas anteriormente se muestran en la figura 1 que representa el diagrama de paquetes. En este diagrama se ve cómo fueron distribuidos los componentes de presentación, el paquete redux donde está todo lo relacionado a la librería Redux, el paquete phaser que contiene lo relacionado a la librería Phaser la cual se utilizó en el componente interactivo para la práctica de patrones.

Figura 1. Diagrama de paquetes

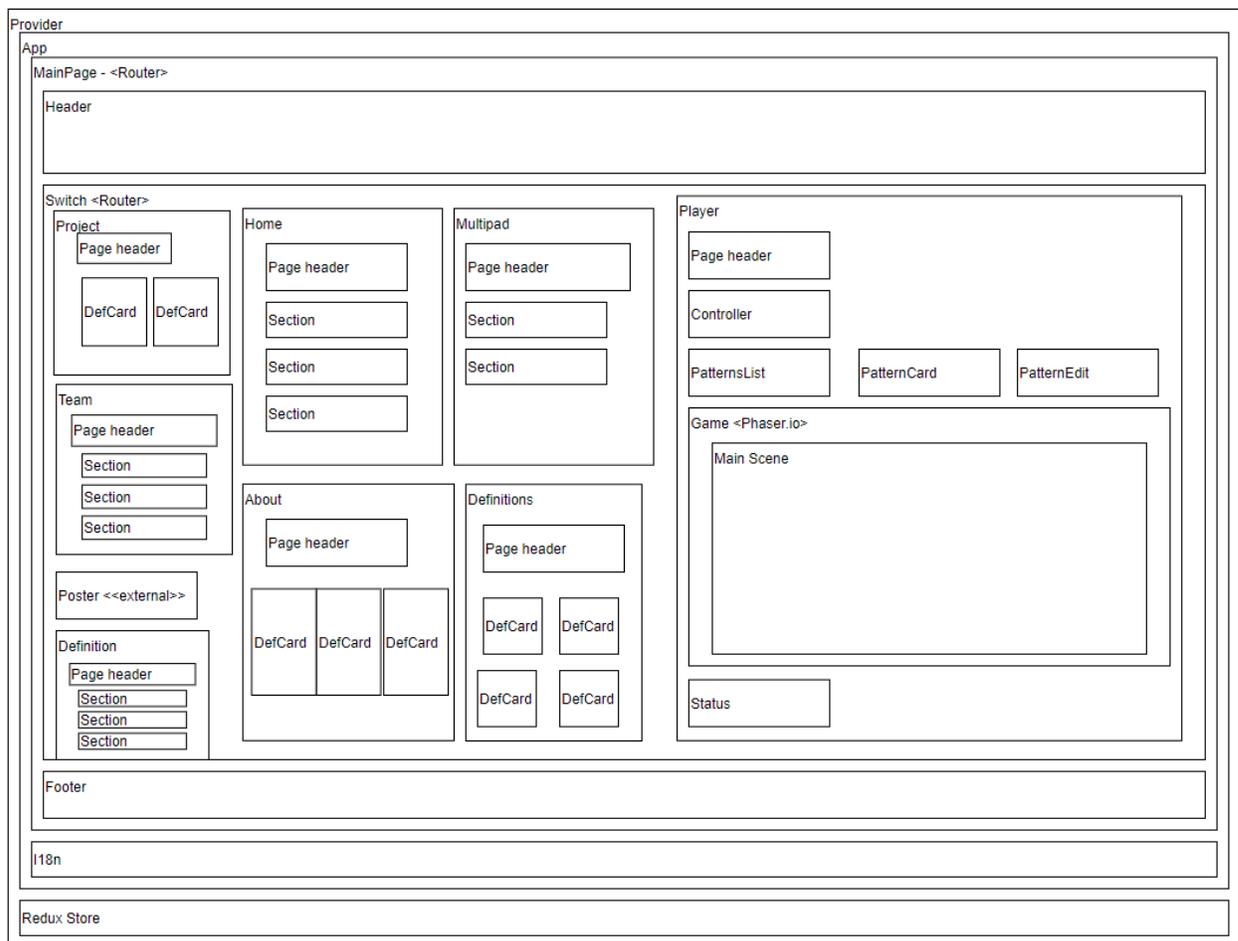


Fuente: desarrollo propio.

## 5. Jerarquía de componentes

El sitio fue construido usando el patrón Composite, en la figura 2 se observa la jerarquía con la que fue construido el sitio web, se muestra en detalle los componentes necesarios para que el sitio sea multi idioma, tenga el estado controlado por redux y como se crea cada página utilizando React Router.

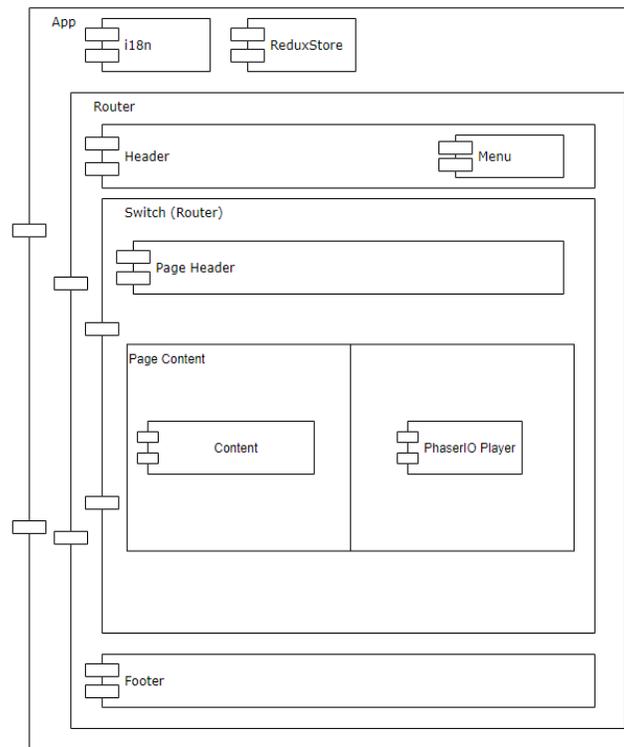
Figura 2. Jerarquía de componentes



Fuente: desarrollo propio.

A continuación, en la Figura 3, se presenta el diagrama de componentes que representa la división en componentes y sus dependencias.

*Figura 3. Diagrama de componentes*



*Fuente. desarrollo propio*

En este diagrama se observa la jerarquía de los componentes utilizados en el sitio web de manera general, el componente principal App es el contenedor de todo el desarrollo de React. Así mismo, a nivel de toda la aplicación se tienen dos componentes los cuales sirven para tareas globales en la aplicación, uno encargado de toda la presentación de los textos en los dos idiomas disponibles (i18n) y el otro es el que se encarga del estado de la aplicación (Redux Store).

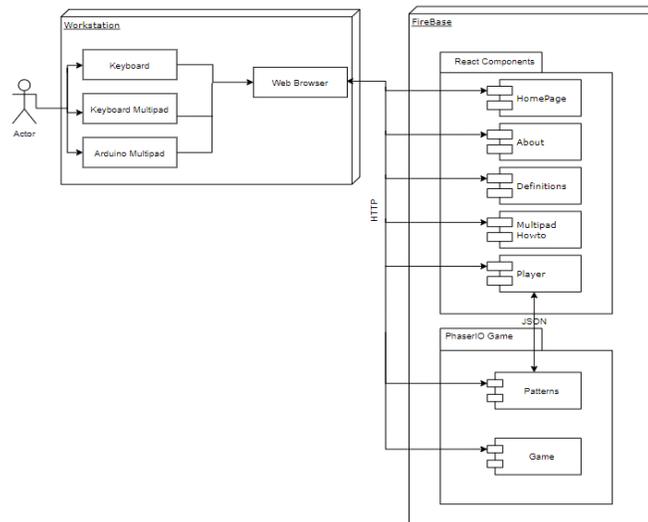
Siguiendo en la jerarquía el componente Router se encarga de la navegación del sitio, desde este componente se selecciona el componente con el contenido a mostrar además del encabezado y el pie de página del sitio.

## 5.1. Vista de despliegue

A continuación, en la Figura 4, se presenta el diagrama de vista de despliegue, el cual describe la relación entre los componentes del sitio web y las partes de software o hardware que se requieren.

En este diagrama se observa que la app está desplegada en Firebase, tanto los componentes React como los componentes de Phaser.io interactúan con el navegador y los 3 posibles tipos de entrada para la práctica.

Figura 4. Diagrama vista de despliegue.

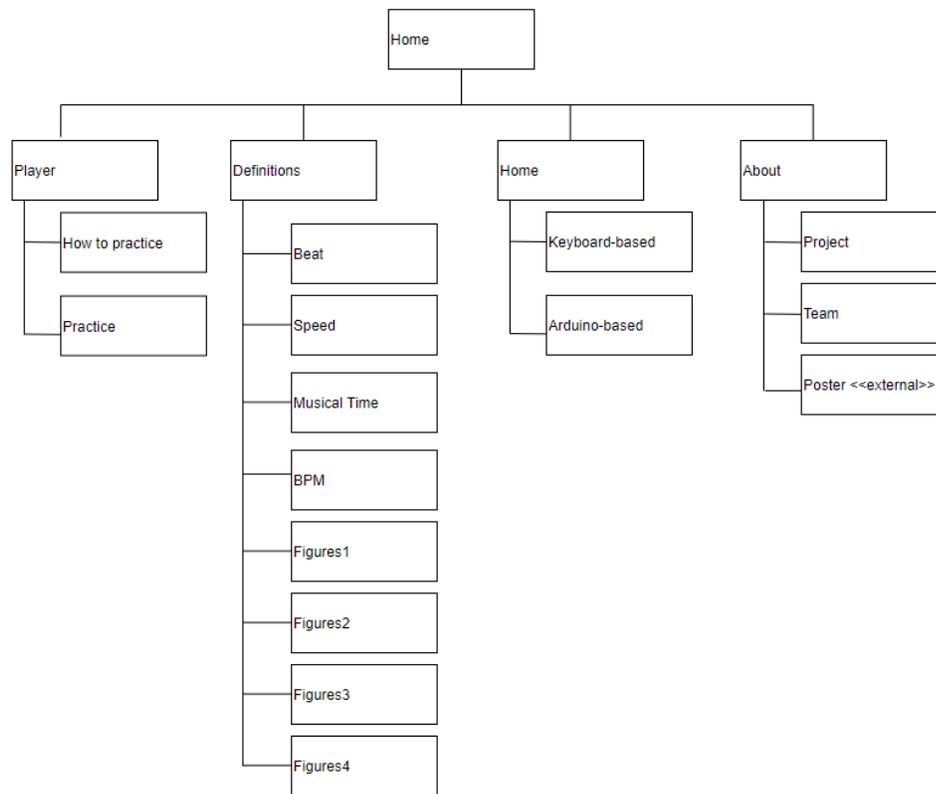


Fuente: desarrollo propio.

## 5.2. Mapa de navegación

En la figura 5, se muestra el diagrama de navegación del sitio, se inicia por la página Home que es la página de inicio de la navegación, usando el menú se pueden ver las otras páginas y los respectivos contenidos como se describe en el árbol. El menú está disponible siempre en el encabezado del sitio web por lo que se puede reiniciar la navegación en cualquier momento.

Figura 5. Diagrama mapa de navegación.



Fuente: desarrollo propio.

## 6. Código Fuente

### 6.1. Obtener Código Fuente

El código fuente del sitio web se encuentra disponible en github.com, el repositorio público se encuentra ubicado en la ruta: <https://github.com/driverco/Rhythme.git>

Una vez obtenido el código fuente desde la carpeta dev es necesario ejecutar el comando “npm install” para que traiga todo lo requerido por el proyecto para la ejecución.

### 6.2. Ramas

El desarrollo del sitio web se realizó empleando varias ramas, a continuación, se describen cada una de ellas:

- **Rama master:** esta es la rama principal, contiene los desarrollos probados y es la que se debe usar para realizar el despliegue en ambiente productivo.
- **Rama DEV:** es la de desarrollo es la que integra los desarrollos para ser probados, allí las demás ramas deben mezclarse y desde esta se hace el pull request para integrar los desarrollos a la rama master.
- **Ramas USXX:** las ramas de historias de usuario se crean desde la rama DEV y son utilizadas para realizar el desarrollo de cada historia de usuario, el nombrado de cada rama se hizo con el número de la historia de usuario, una vez terminadas se realizó la validación de código del sitio con codacy, luego de esto las correcciones que este sitio sugiere para posteriormente hacer el pull request a la rama DEV e integrar el desarrollo en esta.

### **6.3. Compilación**

La compilación se realiza desde la carpeta “client”, con el comando “npm run build”, como resultado obtendremos los archivos que se pueden desplegar en el hosting, estos archivos se ubican en la carpeta build.

### **6.4. Despliegue**

Para el despliegue de esta app se utilizó el servicio de hosting de firebase, que al utilizar la herramienta firebase-cli permite realizar el despliegue con el comando “firebase deploy”, el cual se encuentra configurado en el archivo “firebase.json”, una vez realizado esto encontraremos el sitio en la ruta: <https://rhythme-776fc.web.app/>

## **7. Multipad**

### **7.1. Multipad basado en teclado**

Para el uso del sitio se sugiere que se realice un multipad basado en un teclado de computador, las instrucciones de este multipad y las imágenes del proceso de construcción se encuentran en el sitio web dentro en la ruta:

<https://rhythme-776fc.web.app/keyboardMultipad>

### **7.2. Multipad Arduino**

La herramienta de práctica es más fácil de usar si se usa un multipad, utilizando un Arduino con un microprocesador 32u4 o SAMD podemos crear un multipad que nos permita una interacción con baquetas directamente con la herramienta. Las instrucciones para realizar este multipad y el código fuente requerido se encuentran en la ruta: <https://rhythme-776fc.web.app/arduinoMultipad>

### **7.3. Placas base compatibles**

El multipad Arduino se puede construir con las placas base que tengan el microprocesador 32u4 o SAMD, el listado de las placas compatibles es:

- Arduino micro
- Arduino Leonardo

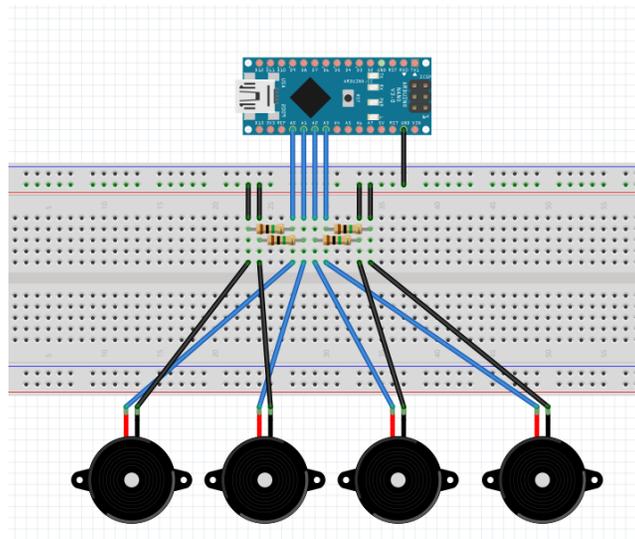
#### 7.4. Conexión del Arduino

La placa base Arduino debe ser conectada al ordenador para realizar la instalación del código fuente, se debe seleccionar el tipo de board correcto y el puerto COM donde se encuentra la board Arduino, con esto ya se puede realizar la instalación del código fuente subiendo a la placa Arduino el archivo que se encuentra en la ruta: <https://rhythme-776fc.web.app/multipad/multipad.ino>

#### 7.5. Diagrama de conexión de componentes electrónicos

La conexión del multipad se realiza utilizando 4 piezoeléctricos, la placa base Arduino, cables de conexión y 4 resistencias de 1kOmhs, el diagrama de conexión se presenta a continuación en la figura 6.

*Figura 6. Diagrama conexión de componentes.*



*Fuente: elaboración propia*

## **8. Resolución de problemas**

Los posibles problemas que se pueden presentar se han dividido en 2, el sitio web y el Arduino.

### **8.1. Sitio web**

- El sitio web ha alcanzado el límite de descarga: este sitio web está alojado usando la capa gratuita de firebase, la cual nos permite una descarga diaria de 350 Mb, una vez alcanzado este límite debemos esperar al día siguiente para que el sitio se vuelva a presentar.

### **8.2. Arduino**

- La librería keyboard.h no se encuentra: este error se presenta cuando se trata de usar una placa base que no es compatible con esta librería, esto debido a que no tiene el microprocesador 32u4, se debe cambiar la placa base Arduino con una compatible.

## Bibliografía

Arduino. (2020). *Arduino*. Obtenido de <https://www.arduino.cc/> el 12 de agosto de 2020

EcuRed. (2020). *EcuRed*. Obtenido de <https://www.ecured.cu/JavaScript> el 06 de octubre

GitHub Docs. (02 de octubre de 2020). Obtenido de <https://docs.github.com/es/free-pro-team@latest/github/collaborating-with-issues-and-pull-requests/about-pull-requests>

Google. (2020). *Firebase*. Obtenido de *Firebase hosting*: <https://firebase.google.com/products/hosting> el 10 de octubre de 2020

Phaser. (2020). *Phaser.io*. Obtenido de <http://phaser.io/> el 06 de octubre de 2020

React. (2020). *React*. Obtenido de <https://es.reactjs.org/> el 06 de octubre de 2020

ReactStrap. (2020). *ReactStrap*. Obtenido de <https://reactstrap.github.io/> el 06 de octubre de 2020

Redux. (2020). *Redux*. Obtenido de <https://es.redux.js.org/docs/introduccion/tres-principios.html> el 06 de octubre de 2020