



Universidad  
Nacional  
de Quilmes

## Trabajo Práctico

# Introducción a la Electrónica y Programación de Controladores con Arduino

### **proyecto**

## Implementación “Guitar Hero” con Arduino

Estudiante: Abel Espínola

Materia: Seminario Introducción a la Electrónica  
y Programación de Controladores con Arduino

Docente: José Luís Di Biase

Ciclo lectivo: 2° cuatrimestre 2022

Carrera: Tecnicatura en Programación – CyT

## Introducción

Este proyecto consiste en el armado e implementación usando una placa Arduino de un juego musical basado en [Guitar Hero](#), [Rock Band](#), [Clone Hero](#), etc..

En este tipo de juegos el objetivo del jugador es intentar apretar a tiempo, usando joystick o mando, una serie de botones que el juego le muestra en una pantalla. Mientras más aciertos tenga, más puntos sumará.

En este proyecto se va a armar un mando usando una placa Arduino UNO con un mini-parlante y 6 botones, donde cada uno está acompañado de su propio led. El parlante va a sonar una melodía y los leds se van a encender o apagar para indicar al jugador cuándo tiene que apretar un botón. Si el jugador aprieta a tiempo el botón cuyo led se encendió, en el parlante va a sonar la nota correspondiente; si aprieta un botón incorrecto, sonará una nota incorrecta; si no se aprieta ningún botón ninguna nota va a sonar. Además, al final de la melodía se va a obtener un puntaje basado en la cantidad de aciertos.

### Origen del proyecto y material externo usado

La idea del tema de este proyecto surgió de este video

<https://www.youtube.com/watch?v=Ej18ALR0Dx0>

aunque la implementación del video y este proyecto son distintas y no se usó ni código ni materiales del video.

La melodía y el mapeo de cada nota musical a un tono del speaker se obtuvo de este otro proyecto

[https://www.youtube.com/watch?v=lseVCKs\\_3ik](https://www.youtube.com/watch?v=lseVCKs_3ik)

De ese proyecto se tomó el archivo llamado pitches.h, que es un header de c++ que tiene unas constantes donde cada una tiene el nombre de una nota musical y como valor un nro. entero (o pitch) que Arduino va a usar para enviar al parlante.

### Link video demostración

El siguiente link muestra el proyecto finalizado y funcionando:

<https://youtu.be/NfSm9QN-0lg>

## **Materiales**

- 1 Arduino Uno R3 chip desmontable Atmel + Cable Usb
- 1 Protoboard breadboard 830 Puntos
- 6 diodos led rojo 2.0-2.2V, 5mm, 20PCS
- 6 botón switch tact B3f-4055 push button genérico 12x12x7.3
- 14 cables macho-macho 30cm
- 1 Mini altavoz speaker 8 Ohm 0.25w 27mm
- 1 Resistencia 1/4w 5% 220 ohm

## **Links y precios de materiales**

Links a materiales, fotos y precios de los materiales usados.

Los precios en pesos argentinos y en dolares son del 8 de Julio 2022.

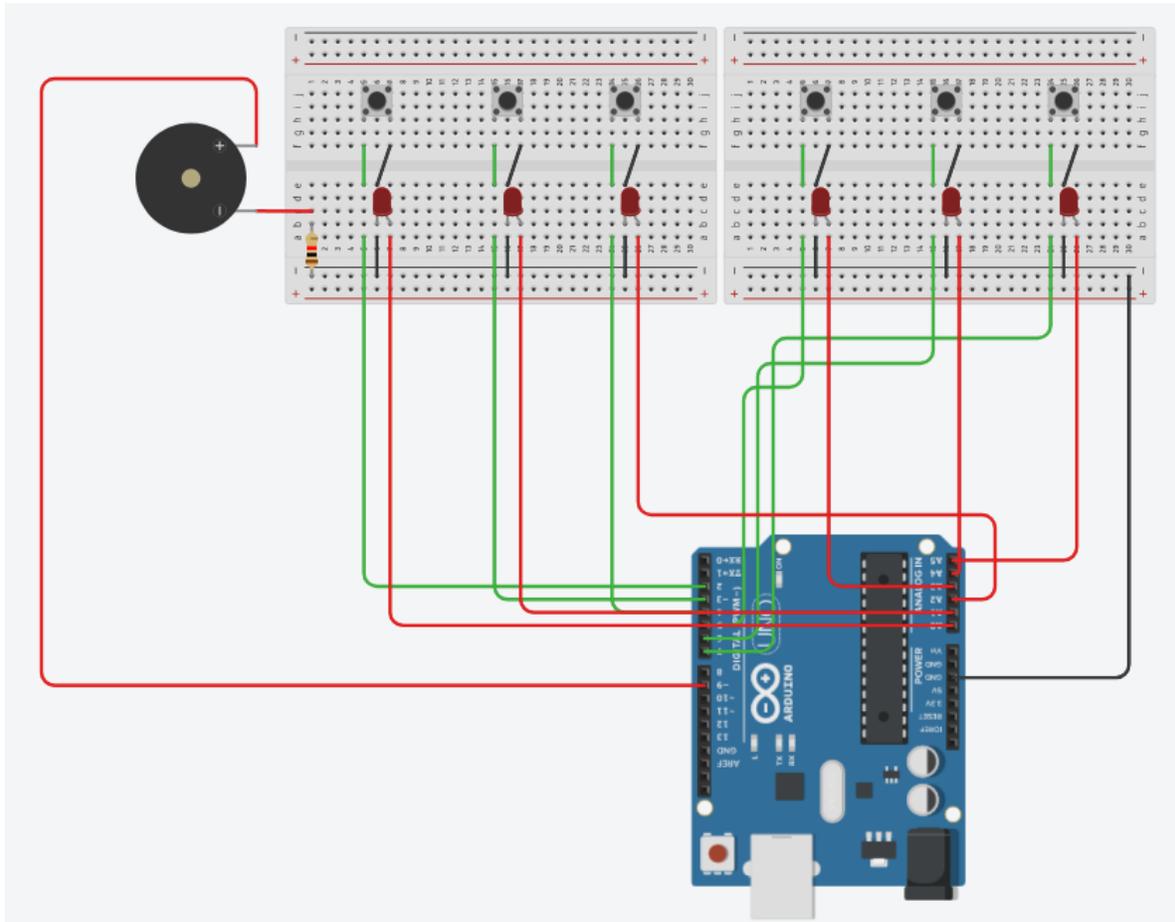
*(cotización dólar informal al 08/07/2022: us\$1/ar\$262)*

Material	Foto de publicación	Precio (ARS USD)
<a href="#">1 x Arduino Uno R3 chip desmontable Atmel + Cable Usb</a>		us\$20,95 ar\$5.488,99
<a href="#">1 x Protoboard breadboard 830 Puntos</a>		us\$3,08 ar\$807,99
<a href="#">6 x Diodos led 2.0-2.2V, 5mm y 3mm</a>		us\$0,18 ar\$48,18
<a href="#">6 x switch tact B3f-4055 push button generico 12x12x7.3</a>		us\$3,36 ar\$887,94
<a href="#">40 x cables macho-macho 30cm</a>		us\$1,96 ar\$513,99
<a href="#">1 x Mini altavoz speaker 8 Ohm 0.25w 27mm</a>		us\$1,02 ar\$267,99
<a href="#">50 x Resistencias Metal 1/4w 5%</a>		us\$1,16 ar\$304,99

Total us\$ 31,71 | Total ar\$ 8.320,07

# Armado

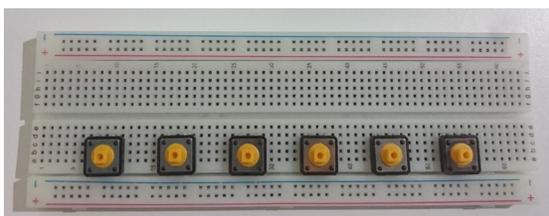
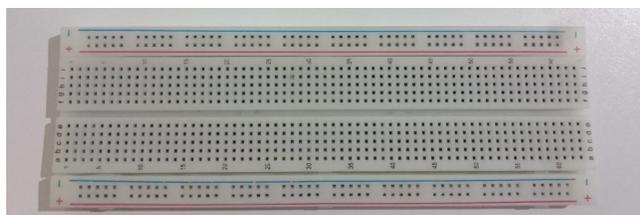
## Circuito



## Pasos

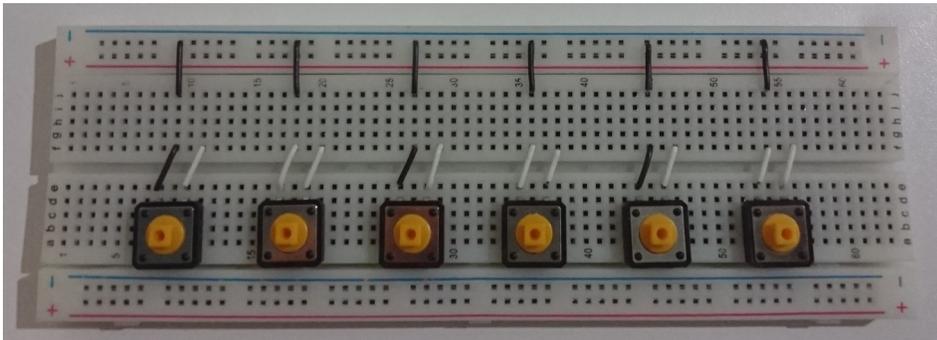
Tener todos los componentes al alcance para empezar con el armado del circuito.

El primer paso es disponer del protoboard sin nada conectado.

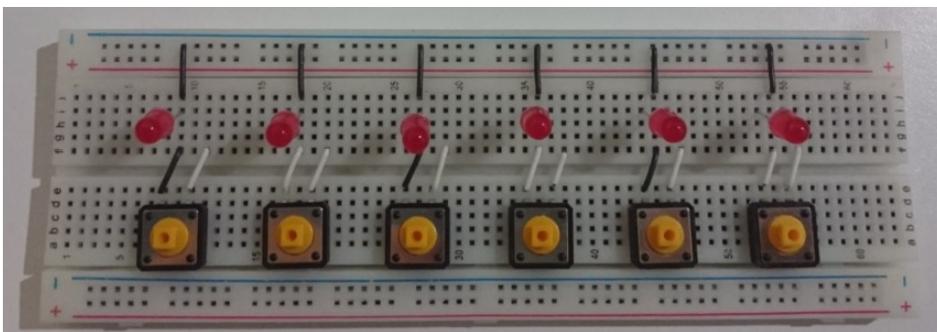


Montar sobre el protoboard los 6 pulsadores, distribuyendolos de igual manera a lo largo de la placa.

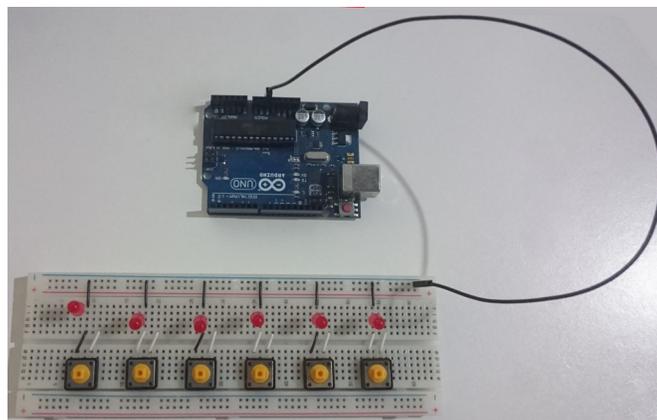
Hacer puentes entre las distintas líneas del protoboard como se ve en las siguientes imágenes. Todos los puentes son iguales para cada botón



Montar los 6 leds al protoboard. Cada led debe tener su terminal negativo en contacto con los cables negros de la parte superior de la placa y su terminal positivo libre.

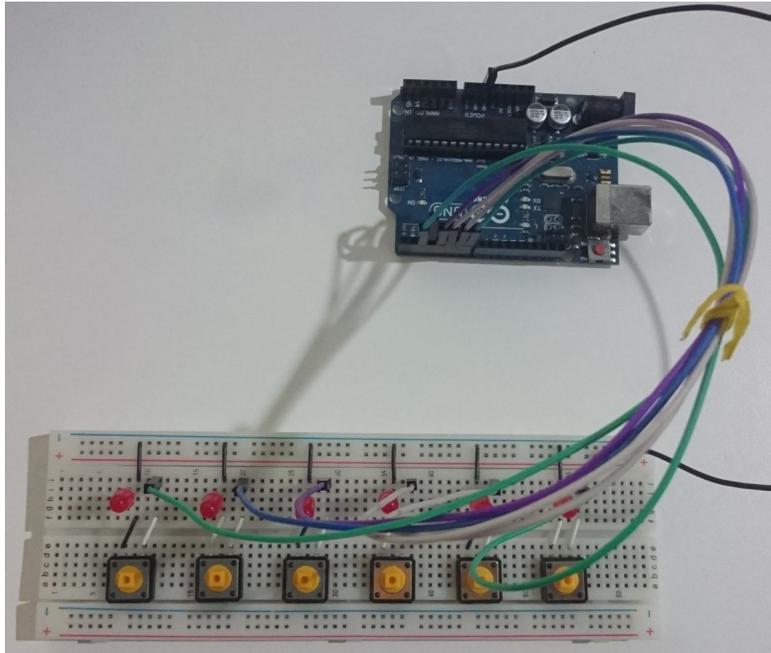


Usar un cable para conectar los terminales negativos del protoboard con un terminal tierra (ground) de la placa Arduino



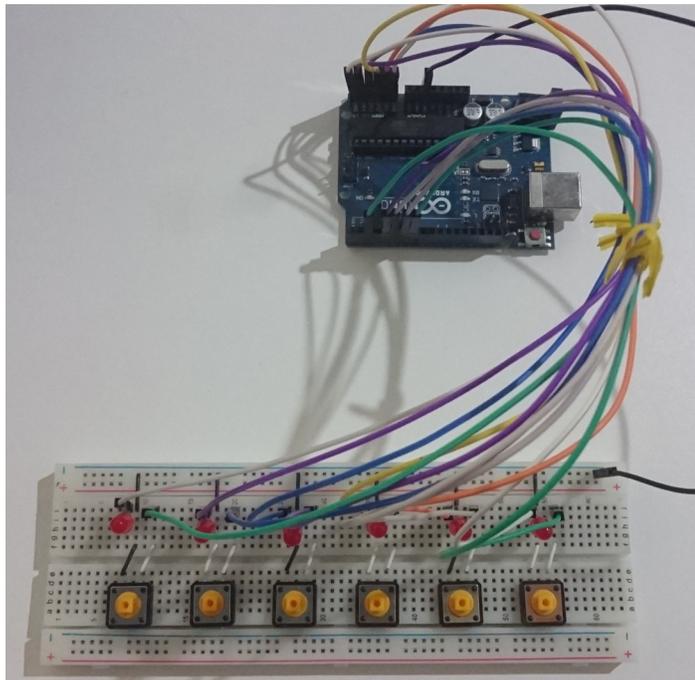
Conectar usando 6 cables cada botón con un puerto digital de la placa arduino:

- btn1 → pin digital 2
- btn2 → pin digital 3
- btn3 → pin digital 4
- btn4 → pin digital 5
- btn5 → pin digital 6
- btn6 → pin digital 7



Conectar usando 6 cables cada pata libre de los leds con un puerto analógico de la placa arduino:

- led1 → pin analógico 5
- led2 → pin analógico 4
- led3 → pin analógico 3
- led4 → pin analógico 2
- led5 → pin analógico 1
- led6 → pin analógico 0

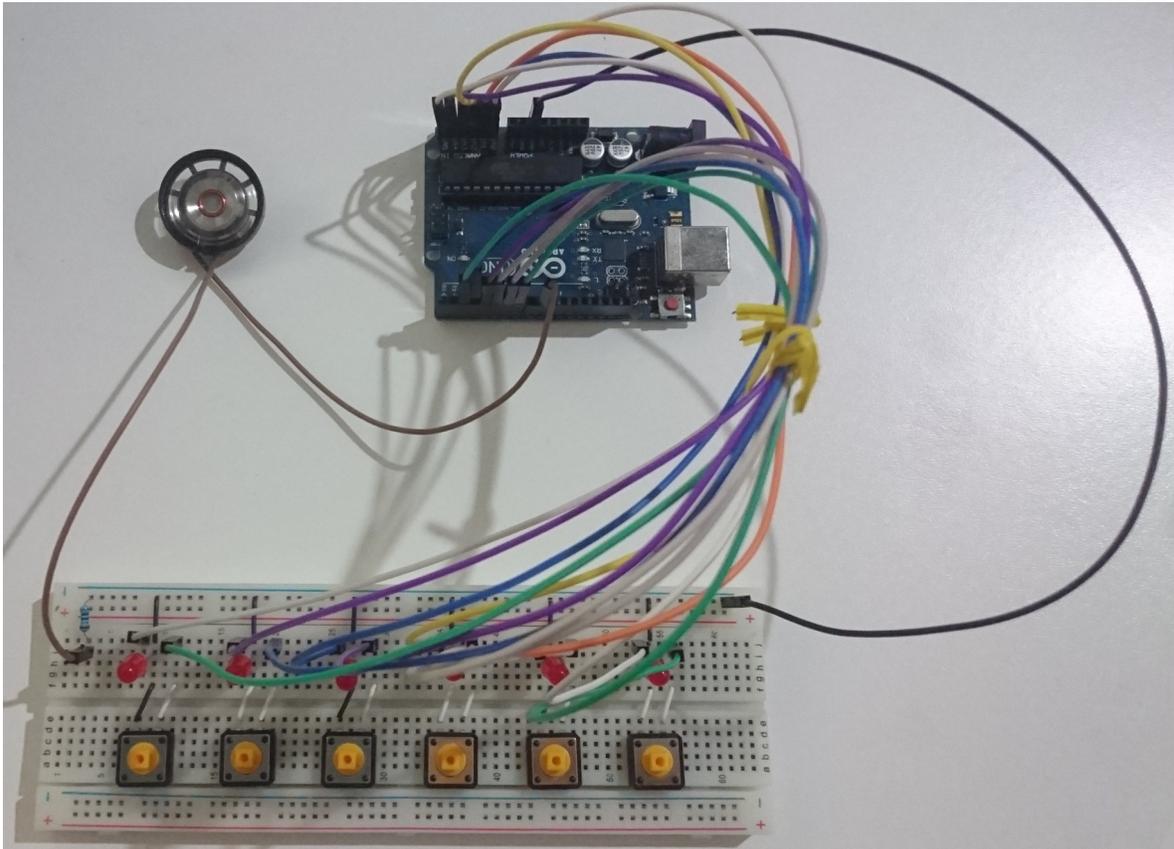


El último paso del armado es conectar el parlante. Para hacer esto el parlante debe tener soldados a sus terminales dos cables, uno para conectarse al

puerto 9 de Arduino y otro cable para conectarse a algún terminal de tierra.

En la siguiente captura se usa el terminal negativo del protoboard como tierra del parlante.

Opcionalmente, para reducir el volumen del parlante puede ponerse una resistencia entre el terminal negativo del parlante y terminal negativo del protoboard.



# Código fuente Arduino

## Repositorio

El código fuente está disponible en Github en el link

<https://github.com/EspinolaAbel/UNQ-ProgramacionMicrocontroladoresArduino>

## IDE

Para modificar el código, compilarlo y subir los binarios al Arduino UNO se puede usar el IDE "Arduino IDE" que cuenta con una interfaz gráfica y que está disponible de forma gratuita en <https://www.arduino.cc/en/software>.

Otra opción orientado a línea de comandos, es "Arduino CLI". Este programa permite compilar, subir binarios a Arduino, etc..

Para el desarrollo de este proyecto se usó Arduino IDE v1.8.19 en una primera etapa del desarrollo y luego se terminó usando Arduino CLI 0.24.0 y Vim 8.1 como editor de código.

## Dificultades encontradas

- El lenguaje de programación usado para programar la placa Arduino (C, C++) fue una dificultad debido a la inexperiencia programando en ese lenguaje.

Adaptarse a este lenguaje fue una complicación que consumió algo de tiempo pero que se pudo superar eventualmente.

- Debido a las limitaciones de espacio en memoria en la placa Arduino UNO hubo complicaciones al cargar la melodía. La melodía consta de alrededor de 100 notas musicales que se representaban como 3 nros. de tipo int.

Ejemplo: 2 39 8, donde:

- 2 es el botón a apretar
- 39 el tono o pitch que Arduino debía enviar al parlante
- 8 la duración en milisegundos de la nota

Cada int ocupa 16bits en la memoria de Arduino UNO, por lo que cada nota ocupaba 48bits y la melodía entera aproximadamente 4.800bits. Al momento de compilar e intentar cargar la aplicación en la placa se lanzaba un error.

Para solucionar esto se optó por representar cada nota de la melodía en binario usando el tipo short en vez de int, lo que permitía ocupar solo 16bits por nota en vez de los 48bit cuando se usaba el tipo int, dando un total de 1.600bits por la melodía entera.

Continuando el ejemplo anterior, la misma nota representada en binario sería 0b0100010011101000 donde:

- 010 representa el botón 2
- 00100111 representa el pitch 39
- 01000 representa la duración de 8 milisegundos