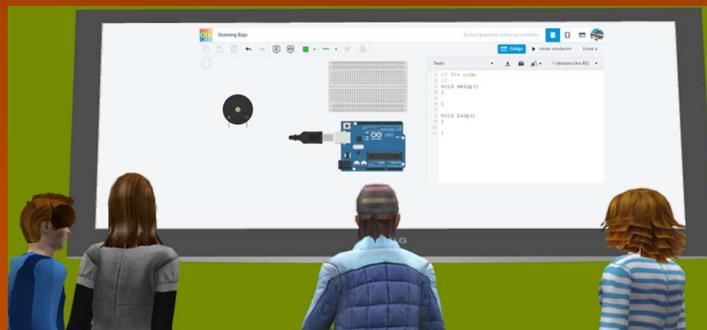


Introducción a la Programación



Secuencias de trabajo para Taller de Ciclo Básico

“Programación de Buzzer...Sonamos”



Secuencias de trabajo para Taller de Ciclo Básico

Secuencia de Trabajo Nº 2 (Para estudiantes de primer ciclo de ETP)

“Programación de un Buzzer...sonamos”

Esta secuencia de trabajo estará organizada con las siguientes actividades:

1. **Explicativa:** Realización del circuito y la programación de un zumbador utilizando “tone”.
2. **Explicativa:** Realización del circuito y la programación de un zumbador utilizando “noTone”.
3. **Problemática circuital y de programación.** (Método de Pólya)
4. **Problemática circuital y de programación.** (Método de Pólya)
5. **Problemática circuital y de programación.** (Método de Pólya)
6. **Problemática circuital y de programación.** (Método de Pólya)
7. **Situación Problema Tecnológica.** “Problemática Abierta” (Metodología Proyectual). *

*En esta última actividad se pretende que las y los estudiantes de taller del primer ciclo (Ciclo Básico), resuelvan una problemática articulando lo distintos saberes desarrollados durante su formación, a través de la metodología proyectual y que este proceso de como resultado la fabricación de un prototipo maquetzado y automatizado.

1 Actividad Explicativa

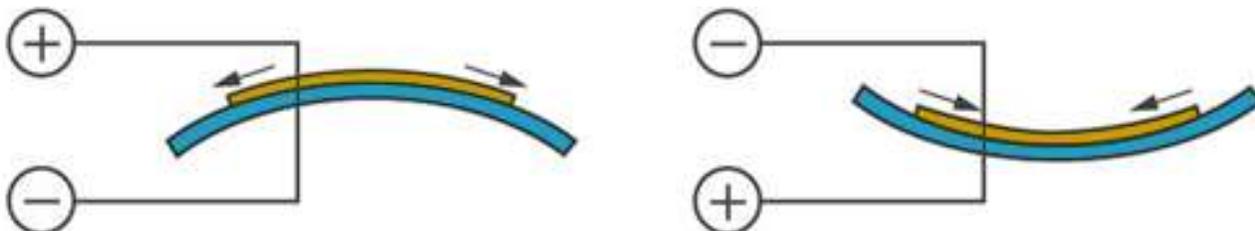
Objetivos

- Emitir un tono de una frecuencia determinada con el zumbador
- Controlar el tiempo de emisión de sonido
- Emitir notas musicales con el zumbador

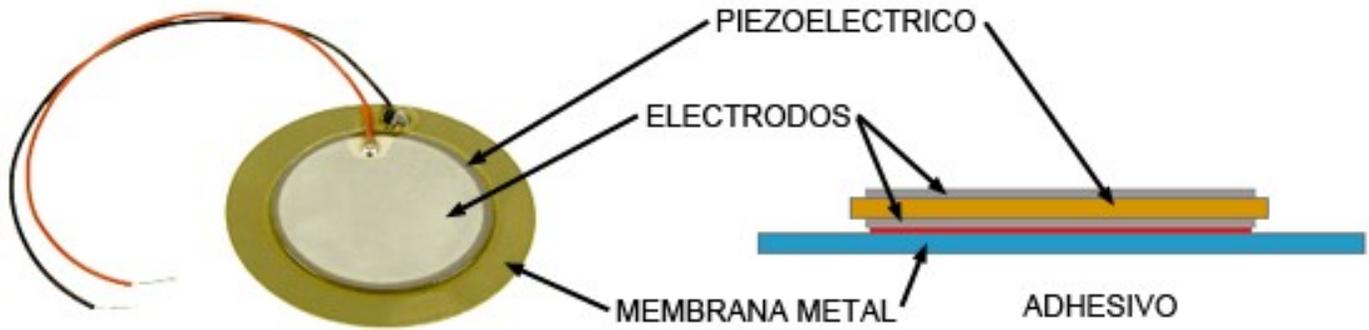
Zumbador o Buzzer emisor de sonidos

El zumbador (en inglés Buzzer) es un pequeño altavoz con forma cilíndrica y de color negro. Su función es hacer señales acústicas para llamar la atención, por esa razón tiene un sonido especialmente agudo y penetrante. La calidad del sonido emitido es pobre.

Los buzzer son transductores piezoeléctricos. Los materiales piezoeléctricos tiene la propiedad especial de variar su volumen al ser atravesados por corrientes eléctricas.



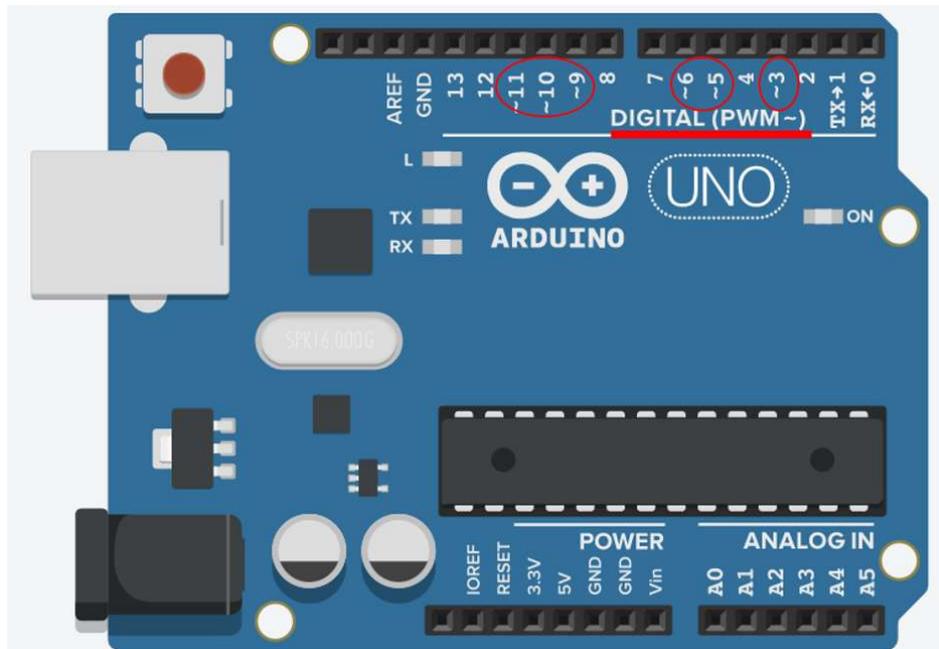
Un buzzer aprovecha este fenómeno para hacer vibrar una membrana al atravesar el material piezoeléctrico con una señal eléctrica.



Los buzzer son dispositivos pequeños y compactos, con alta durabilidad, y bajo consumo eléctrico, cuya calidad de sonido es reducida.

El buzzer o zumbador no tiene control sobre la intensidad del sonido ni sobre su timbre (sonoridad). Por otro lado, el zumbador sí que tiene la capacidad de reproducir diferentes notas musicales controlando la frecuencia (tono) y la duración (figura) del sonido emitido. Esto le permite reproducir partituras de una forma simple.

Si desde un pin de Arduino conseguimos suministrar una tensión que varíe con el tiempo como una señal **PWM**, podremos reproducir diferentes sonidos.



Las unidades utilizadas para medir frecuencia serán los hercios, o su símbolo Hz. Un hercio equivale a una oscilación por segundo. Otra unidad común es el kilohercio o kHz que equivale a 1000 Hz o mil oscilaciones por segundo. Las unidades utilizadas para medir tiempo serán los milisegundos o su símbolo ms. Mil milisegundos equivalen a un segundo.

El oído humano es más sensible al rango de frecuencias que va desde 500 Hz (graves) hasta los 2000 Hz (agudos). Este rango lo cubre aproximadamente las octavas 5ª y 6ª (octavas musicales segunda y tercera)

La siguiente tabla representa las constantes, el valor equivalente y las notas que representan para la primera octava musical (cuarta octava en la notación científica).

Constante	Valor	Frecuencia	Nota Científica	Nota
Do4	49	261 Hz	C ₄	Do
Do_4	50	277 Hz	C# ₄	Do sostenido (Do#)
Re4	51	294 Hz	D ₄	Re
Re_4	52	311 Hz	D# ₄	Re sostenido (Re#)
Mi4	53	330 Hz	E ₄	Mi
Fa4	54	349 Hz	F ₄	Fa
Fa_4	55	370 Hz	F# ₄	Fa sostenido (Fa#)
Sol4	56	392 Hz	G ₄	Sol
Sol_4	57	415 Hz	G# ₄	Sol sostenido (Sol#)
La4	58	440 Hz	A ₄	La
La_4	59	466 Hz	A# ₄	La sostenido (La#)
Si4	60	494 Hz	B ₄	Si

El resto de octavas tienen la misma denominación para las notas, cambiando solo el número final para designar la octava. Para cambiar una octava, se puede añadir o restar el número 12:

$$\text{Do4} + 12 = \text{Do5}$$

$$\text{Do4} - 12 = \text{Do3}$$

La siguiente tabla muestra el valor y la frecuencia de la nota Do de cada una de las octavas:

Constante	Valor	Frecuencia	Nota Científica	Nota
Silence	0	0 Hz		Sin sonido (silencio)
Do0	1	16,35 Hz	C ₀	Do Sub contra octava
Do1	13	32,70 Hz	C ₁	Do Contra octava
Do2	25	65,41 Hz	C ₂	Do Gran octava
Do3	37	130,8 Hz	C ₃	Do Pequeña octava
Do4	49	261,6 Hz	C ₄	Do Octava prima
Do5	61	523,2 Hz	C ₅	Do Octava segunda
Do6	73	1046 Hz	C ₆	Do Octava tercera
Do7	85	2093 Hz	C ₇	Do Octava cuarta
Do8	97	4186 Hz	C ₈	Do Octava quinta
Do9	109	8372 Hz	C ₉	Do Octava sexta
Do10	121	16744 Hz	C ₁₀	Do Octava séptima
Fa10	127	23679 Hz	F ₁₀	Fa Octava séptima

Situación a resolver:

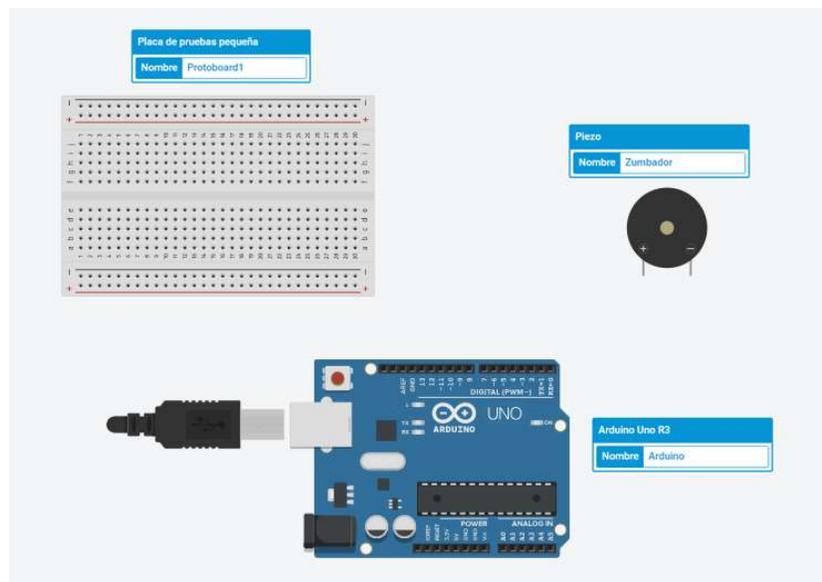
Hacer sonar un BUZZER durante 3 segundos y apagarlo durante 3 segundos de forma indefinida con una frecuencia de 261 Hz (Nota Do). También esta situación se podría plantear “Hacer sonar un buzzer en nota DO con intervalos de 3 segundos de forma permanente”. El o la docente diseñara el enunciado del problema de la forma que considere.

Componentes de conexión del sistema:

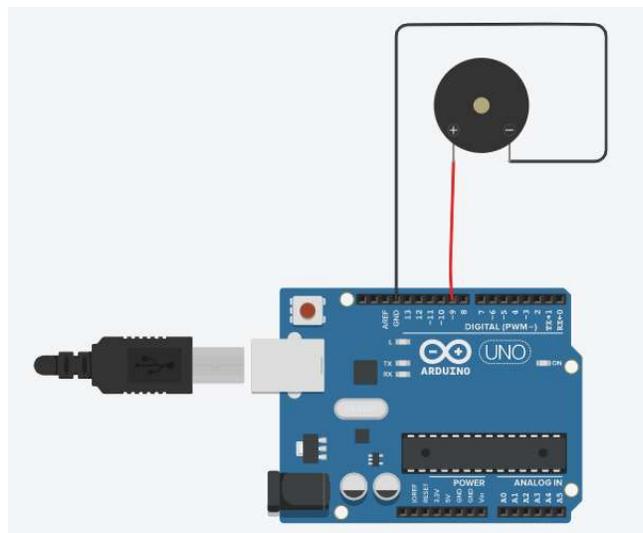
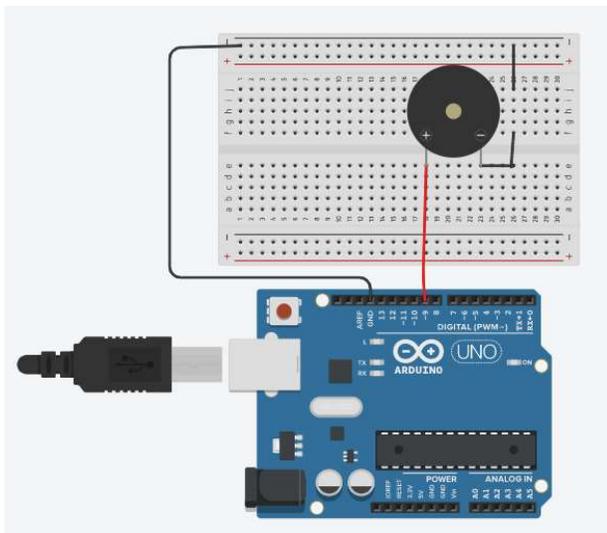
Arduino (Pin seleccionado 8 como salida) - Protoboard – Buzzer.

Procedimiento:

Seleccionar la placa Arduino Uno R3, El protoboard y el zumbador de los componentes de TINKERCAD.

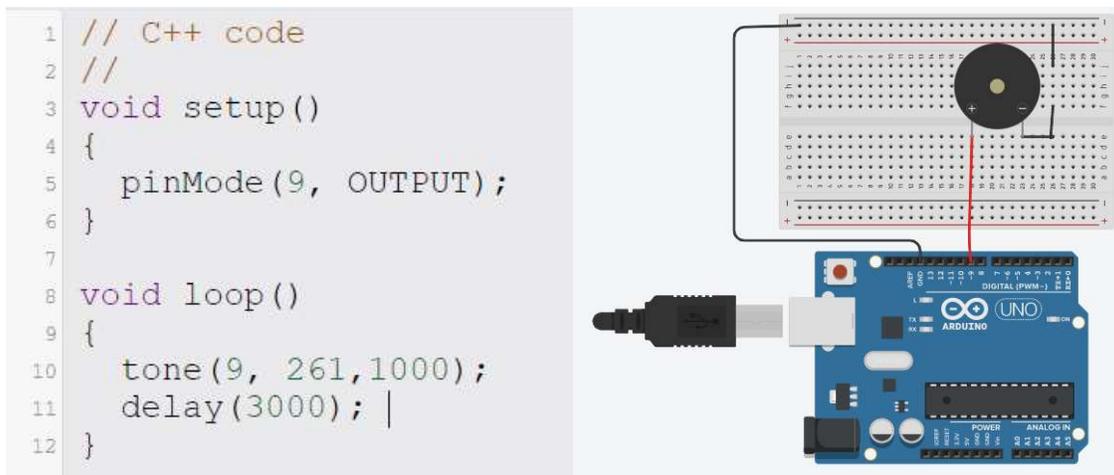


Realizamos el conexionado: En la imagen se puede ver el mismo conexionado con el protoboard y sin él. El funcionamiento será el mismo en ambos casos.



Programación del Sistema

Analicemos la siguiente programación:



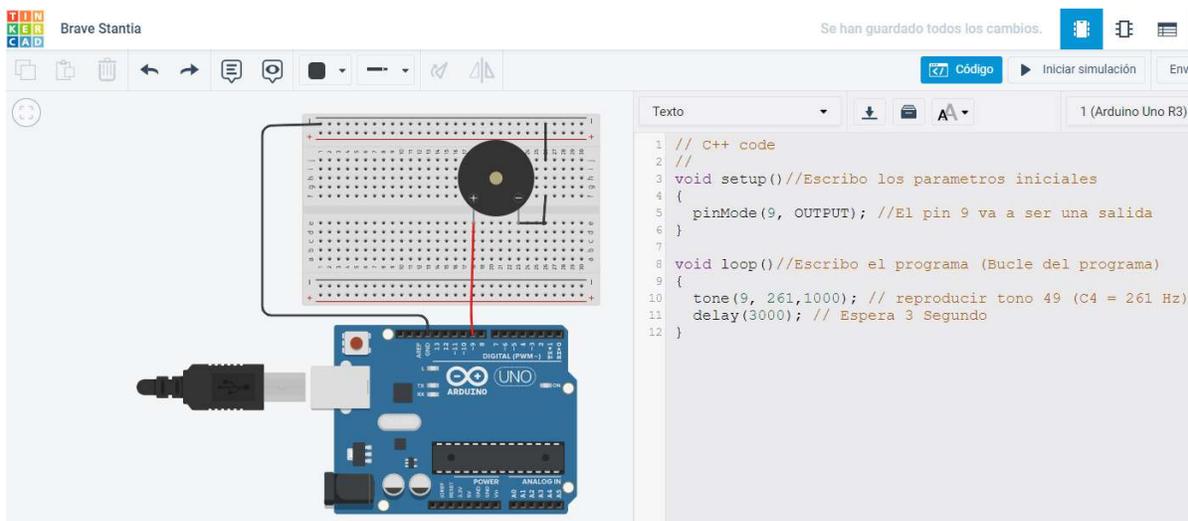
“**tone ()**” genera una onda cuadrada de una frecuencia específica.

“**void setup ()**”: Es donde escribimos los parámetros iniciales. En nuestro caso el parámetro indica que el “**pin 9**” será una **salida**.

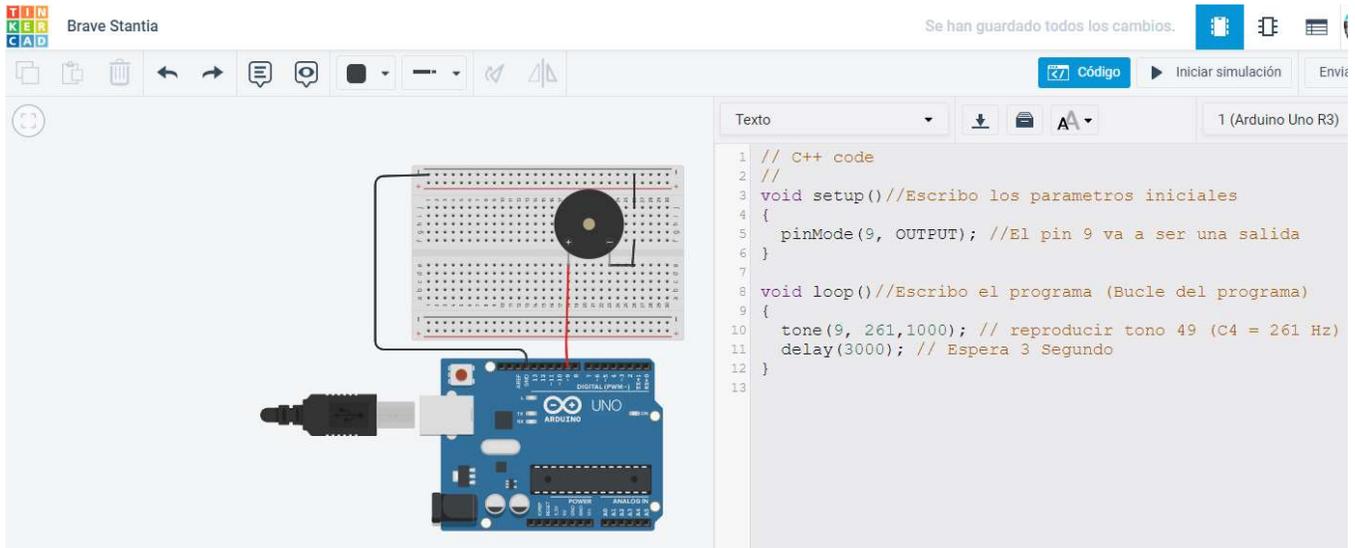
“**void loop ()**”: Es donde escribiremos el programa (lo que queremos hacer) en nuestro caso que deseamos que el buzzer suene durante 3 segundos con una frecuencia de 261 Hz reproduciendo la nota “Do” y luego que se mantenga apagado durante 3 segundos, y así sucesivamente de forma indefinida:

“**tone (9, 261,1000);**” Esto significa reproducir tono valor 49 cuya nota científica es C_4 , a la que corresponde una frecuencia de 261 Hz.

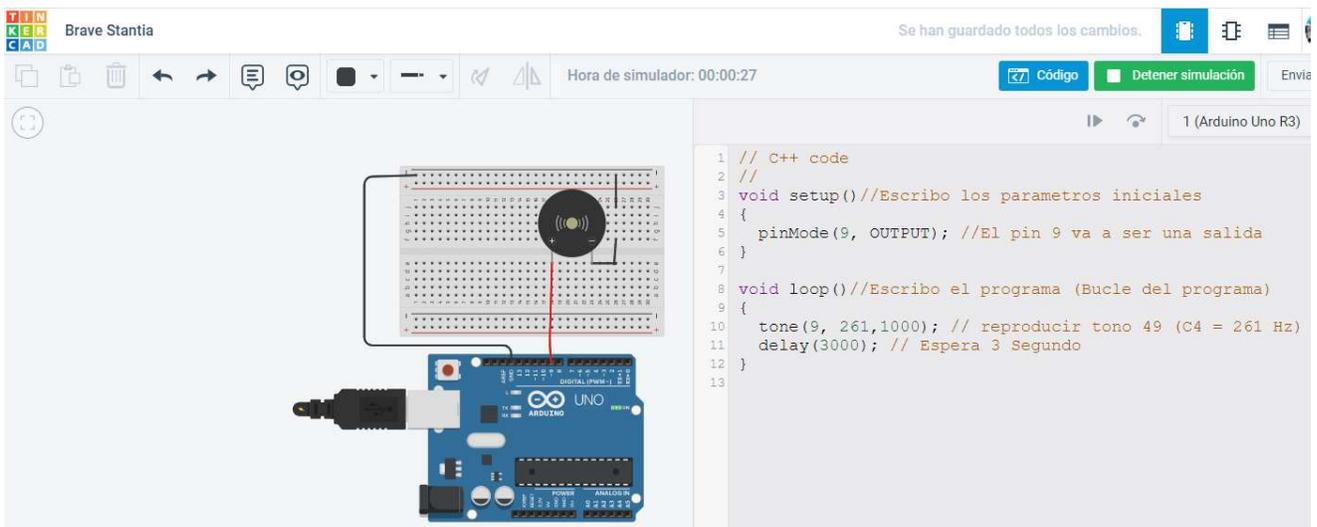
“**delay (3000);**” // Espera 3 Segundo



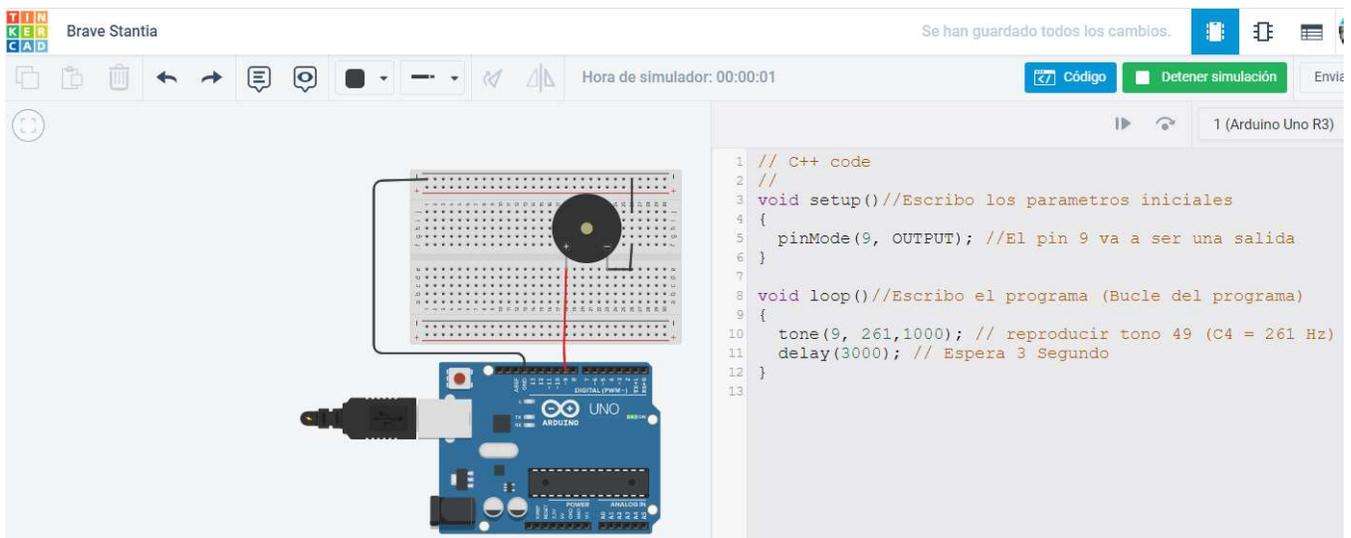
Una vez escrito nuestro código iniciamos la simulación del sistema y verificamos su funcionamiento:



El buzzer conectado al “pin 9”, emite el sonido de la nota Do durante 3 segundos.



El buzzer deja de emitir el sonido de la nota Do durante 3 segundos.



Este programa se realiza correctamente. Por lo tanto la programación es adecuada. Pero no es la única forma:

2 Actividad Explicativa

Objetivos

- Emitir un tono de una frecuencia determinada con el zumbador
- Controlar el tiempo de emisión de sonido
- Emitir notas musicales con el zumbador
- Utilizar la función noTone

Situación a resolver:

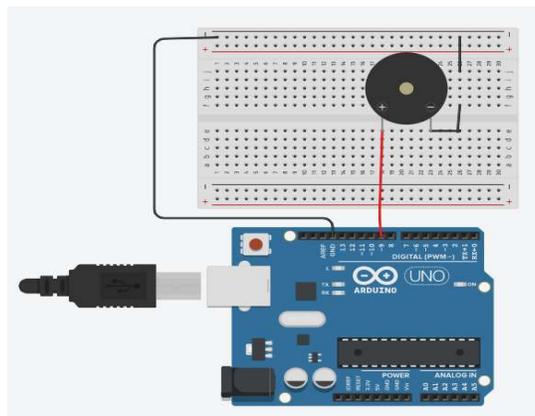
Hacer sonar un BUZZER durante 3 segundos y apagarlo durante 3 segundos de forma indefinida con una frecuencia de 261 Hz (Nota Do), utilizando la función “noTone”. También esta situación se podría plantear “Hacer sonar un buzzer en nota DO con intervalos de 3 segundos de forma permanente, utilizando la función “noTone”. El o la docente diseñara el enunciado del problema de la forma que considere.

Componentes de conexión del sistema:

Arduino (Pin seleccionado 8 como salida) - Protoboard – Buzzer.

Procedimiento:

Seleccionar la placa Arduino Uno R3, El protoboard y el zumbador de los componentes de TINKERCAD y realizar las conexiones.



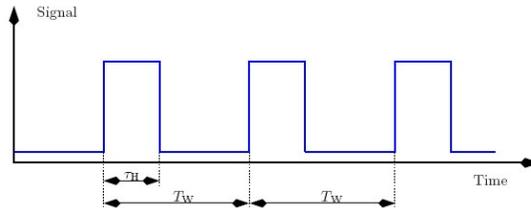
Programación del sistema:

La siguiente programación con la utilización de la función “noTone”, nos permitirá producir el mismo efecto, con la salvedad de que si deseamos reproducir tonos diferentes en varios pines, es necesario llamar noTone () en un pin antes de llamar a tone () en el siguiente pin.

Tone () genera una onda cuadrada de una frecuencia específica y con un 50% de DUTY CYCLE en el pin especificado. La duración del tono puede ser especificado o en caso contrario continúa hasta llamar a la función **noTone ()**.

DUTY CYCLE: En electrónica, el **ciclo de trabajo, ciclo útil o régimen de trabajo** es la relación que existe entre el tiempo en que la señal se encuentra en estado activo y el periodo de la misma. Su valor se encuentra comprendido entre 0 y 1, y viene dado por la siguiente expresión:

Por ejemplo en la siguiente imagen, **TH** es el tiempo en estado activo y **TW** es el periodo de la señal, para calcular el **DUTY CYCLE** tenemos que hacer lo siguiente.



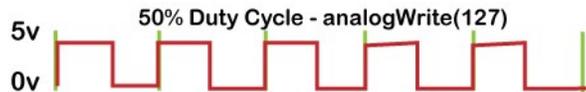
$$D = 100 \frac{Th}{Tw}$$

El 100 es solamente para expresarlo en porcentajes, existen algunos casos particulares que se pueden dar al trabajar este tipo de señales que es bueno comentar.

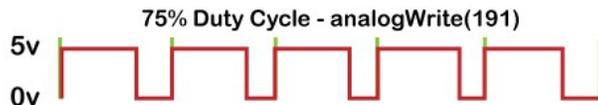
0% Duty: Cuando el Duty Cycle es igual cero significa que en ningún momento la señal se encuentra en estado alto, significa que la señal no entrega potencia.



50% Duty: Al colocar el Duty Cycle en 50%, significa que la mitad del periodo la señal se encuentra en estado alto, lo que significa que si utilizamos esta señal para prender un led, lo estaríamos haciendo con la mitad de la potencia que tenemos disponible, para el caso del led se notaría que la luz se enciende con menor intensidad.



75% Duty: Este caso significa que estamos entregando una potencia de salida del 75% de la máxima posible.



100% Duty: Estamos entregando la totalidad de la potencia posible, lo que significa que la señal nunca se encuentra en estado bajo.



Función “noTone ()”: La función “noTone ()” deja de generar la onda cuadrada que en principio estaba emitiéndose por una ejecución previa de “tone ()”, a través del pin especificado. Si no hay ninguna señal emitiéndose en ese pin, esta función no hace nada.

```
1 // C++ code
2 //
3 void setup()//Escribo los parametros iniciales
4 {
5   pinMode(9, OUTPUT); //El pin 9 va a ser una salida
6 }
7
8 void loop()//Escribo el programa (Bucle del programa)
9 {
10  tone(9, 261,1000); // reproducir tono 49 (C4 = 261 Hz)
11  delay(2000); // Espera 2 Segundo
12  noTone(9);//deja de generar la onda cuadrada que estaba emitiéndose por una ejecución de tone()
13  delay(1000); // Espera 1 Segundo
14 }
```

“noTone ()”: La función “noTone ()” deja de generar la onda cuadrada que en principio estaba emitiéndose por una ejecución previa de “tone ()”

“void setup ()”: Es donde escribimos los parámetros iniciales. En nuestro caso el parámetro indica que el “pin 9” será una salida.

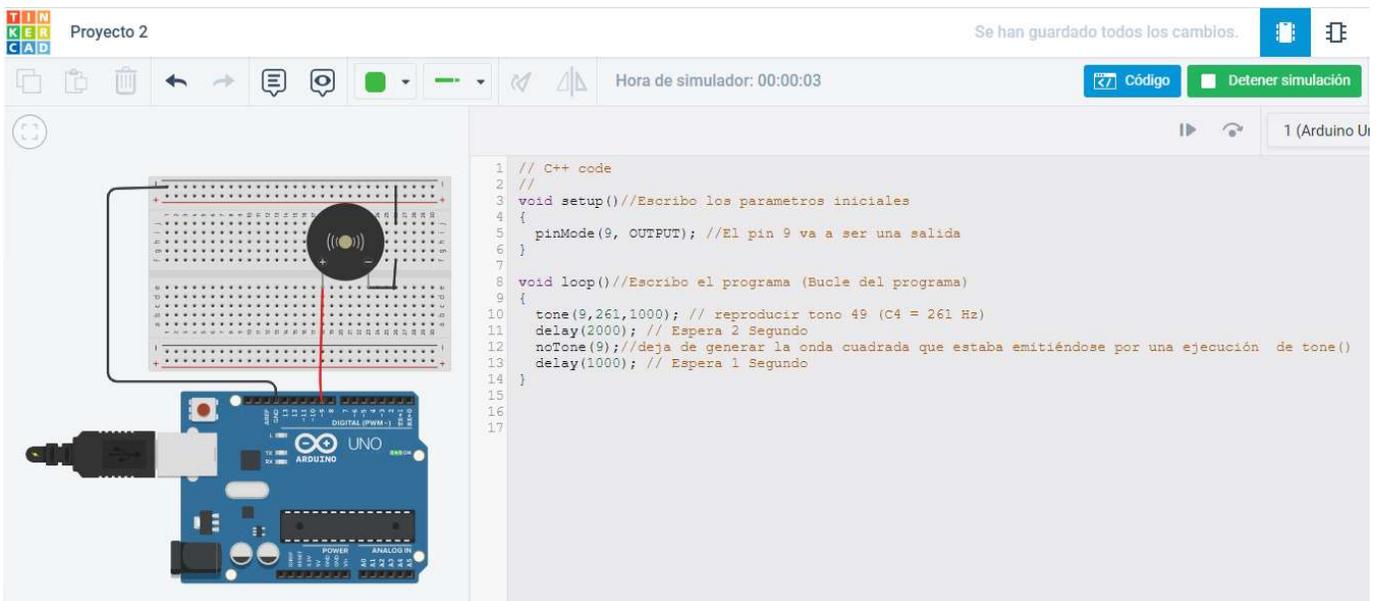
“void loop (): Es donde escribiremos el programa (lo que queremos hacer) en nuestro caso que deseamos que el buzzer suene durante 3 segundos con una frecuencia de 261 Hz reproduciendo la nota “Do” y luego que se mantenga apagado durante 3 segundos, y así sucesivamente de forma indefinida:

“tone(9, 261, 1000);” Esto significa reproducir tono 49 cuya nota científica es C₄, a la que corresponde una frecuencia de 261 Hz.

“delay(2000);” Esto significa espera 2 Segundos.

“noTone(9);” Esto significa que el pin 9 deja de generar la onda cuadrada que estaba emitiéndose por una ejecución de tone()

“delay(1000);” Esto significa espera 1 Segundo.

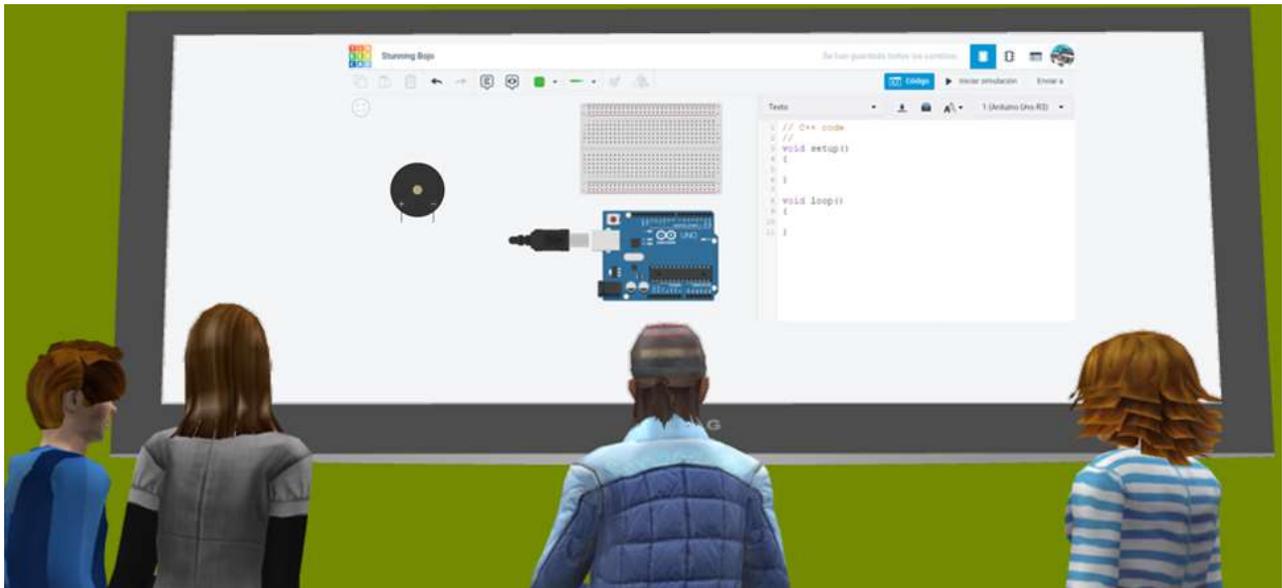


The screenshot shows the Arduino IDE interface. On the left, a breadboard is connected to an Arduino Uno. A buzzer is connected to pin 9 of the Arduino. The code editor on the right displays the following C++ code:

```
1 // C++ code
2 //
3 void setup()//Escribo los parametros iniciales
4 {
5   pinMode(9, OUTPUT); //El pin 9 va a ser una salida
6 }
7
8 void loop()//Escribo el programa (Bucle del programa)
9 {
10  tone(9,261,1000); // reproducir tono 49 (C4 = 261 Hz)
11  delay(2000); // Espera 2 Segundo
12  noTone(9);//deja de generar la onda cuadrada que estaba emitiéndose por una ejecución de tone()
13  delay(1000); // Espera 1 Segundo
14 }
```

Situación Problema

A continuación se plantearán una serie de situaciones problemas para poder trabajar con los estudiantes. A cada una de ellas corresponderá una solución alternativa. Para su solución te invitamos a utilizar el método de Pólya en cual consiste en las siguientes etapas:



Etapas del método Pólya de Resolución de Problemas

1) Comprender el problema. Reconocer que se pregunta, identificar lo que hay que resolver y las condiciones asociadas.

2) Elaborar un plan. Se trata de establecer la vinculación entre los datos presentes y el problema a resolver, determinar los recursos que se utilizarán, verificar la similitud con otros problemas previamente resueltos y también la posibilidad de utilizar teorías o modelos útiles, todo esto en función de buscar una manera de resolver el problema.

3) Ejecutar el plan. Desarrollar el resultado de la respuesta, a partir de ejecutar el plan, avanzando y verificando cada paso.

4) Revisar y verificar. Controlar que hace y que dice el resultado, con vistas a considerar la posibilidad de transferir la solución a otros problemas.

Actividades

3. Problemática circuital y de programación. (Método de Pólya).

Hacer sonar un buzzer cuya conexión será a través de una salida PWM, reproduciendo el tono 51 que equivale a la nota “Re” durante 3 segundos, para luego mantenerse en silencio durante 2 segundos, y después reproducir el tono 60 que equivale a la nota “Si” durante 3 segundos, para luego silenciarse durante 2 segundos. Esta secuencia deberá repetirse de forma indefinida.

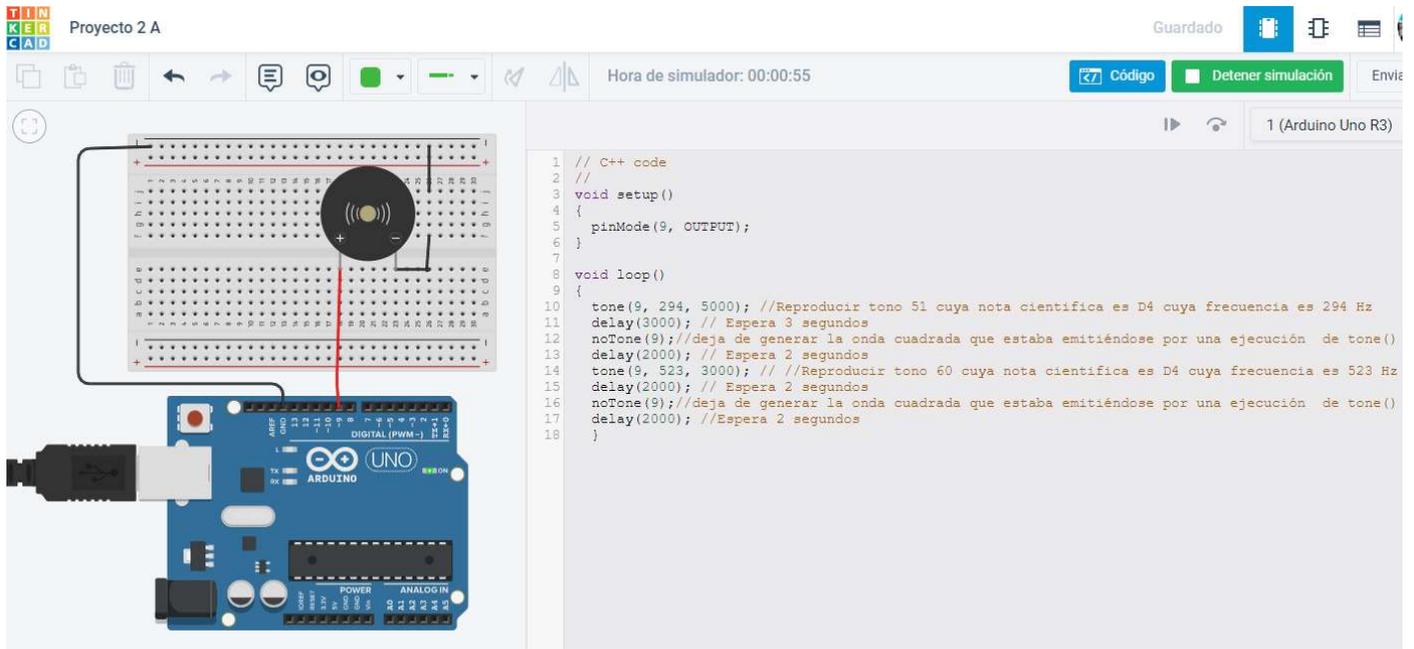
También podrá plantearse “Hacer sonar un buzzer alternando la nota “Re” y la nota “Si” cada 2 segundos. La reproducción de cada tono durara 3 segundos. Esta secuencia deberá repetirse de forma indefinida y para su conexión se utilizara una salida PWM. El o la docente diseñara el enunciado del problema de la forma que considere.

Te invitamos a proponer una solución a esta situación utilizando la metodología de Pólya para resolver problemas y realizar el circuito y la programación que dé respuesta a la problemática:



Etapas del Método Pólya		Acciones
1	Comprender el Problema <i>Lo expresan con sus propias palabras</i>	Analizo y comprendo el problema Debemos hacer sonar un buzzer alternando las notas “Re” y “Si” en un lapso 2 segundos. Cada reproducción de debe durar 3 segundos. Esta secuencia deberá repetirse de forma indefinida y para su conexión debo utilizar una salida PWM.
2	Elaborar un Plan <i>Diseñan un plan para resolver el problema</i>	Diseño un plan para resolver el problema Debemos elegir los componentes diseñar el circuito, seleccionar un pin PWM, realizar la conexión, investigar los Hz que corresponden a cada tono seleccionado y realizar la posible programación.
3	Ejecutar el Plan <i>Ejecutan el plan diseñado</i>	Realizo el circuito y la programación del plan previamente diseñado Utilizando TINKERCAD realizamos el circuito, seleccionamos el pin de salida PWM y la programación diseñada previamente.
4	Revisar y Verificar la Solución <i>Verifican el funcionamiento del sistema</i>	Reviso y verifico que la solución diseñada sea la adecuada Ejecutamos la simulación del sistema y verificamos que funcione correctamente en relación a la problemática a resolver. Analizamos que utilidad le podemos dar a este saber en otro contexto o situación.

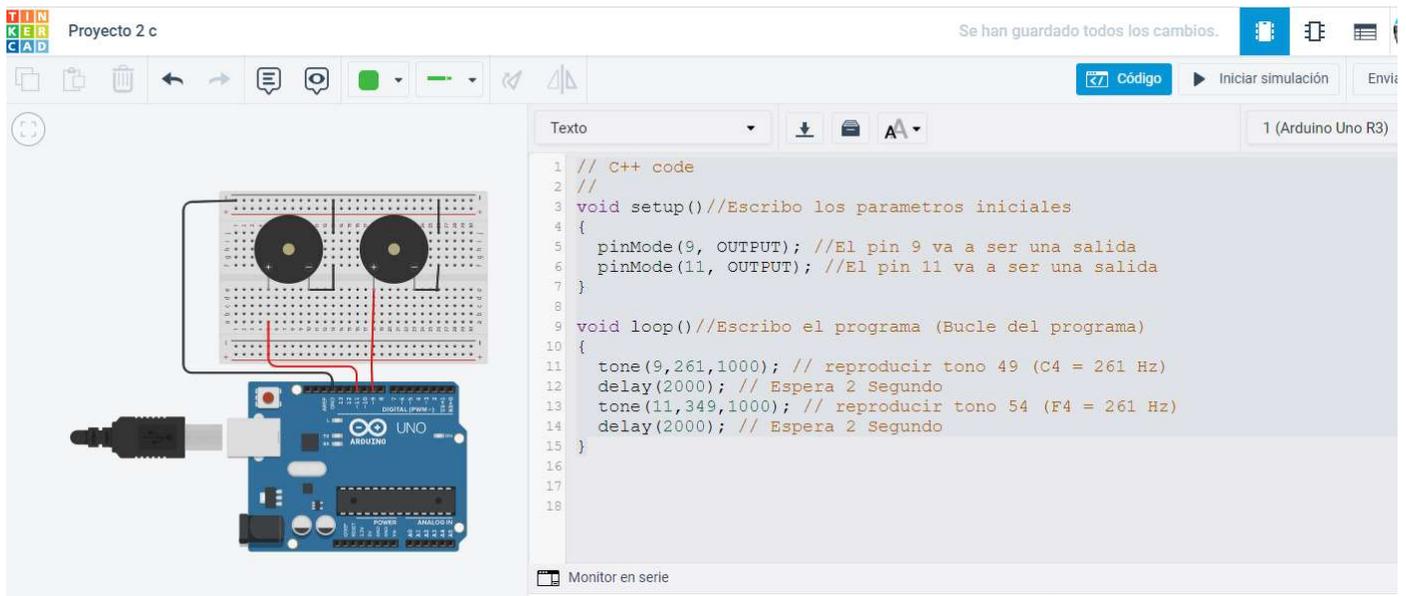
Alternativa posible:



4. Problemática circuital y de programación. (Método de Pólya).

Hacer sonar dos buzzer de forma alternada conectados a dos salidas PWM, reproduciendo el tono 49 que equivale a la nota “Do” de la escala musical en un buzzer y en el otro el tono 54 que equivale a la nota “Fa” de la escala musical, repitiendo el bucle de forma permanente.

Etapas del Método Pólya		Acciones
1	<p>Comprender el Problema</p> <p>Lo expresan con sus propias palabras</p>	<p>Analizo y comprendo el problema</p> <p>Debemos realizar un circuito y la programación de un sistema que contenga y haga sonar dos buzzer de forma alternada. En uno deberá sonar la nota “Do” y en otro la nota Fa de la escala musical, repitiendo el bucle de forma indefinida.</p>
2	<p>Elaborar un Plan</p> <p>Diseñan un plan para resolver el problema</p>	<p>Diseño un plan para resolver el problema</p> <p>Debemos elegir los componentes diseñar el circuito, seleccionar los pines PWM, realizar la conexión, investigar los Hz que corresponden a cada tono seleccionado y realizar la posible programación.</p>
3	<p>Ejecutar el Plan</p> <p>Ejecutan el plan diseñado</p>	<p>Realizo el circuito y la programación del plan previamente diseñado</p> <p>Utilizando TINKERCAD realizamos el circuito, seleccionando los dos pines de salida PWM para luego ejecutar la programación diseñada previamente.</p>
4	<p>Revisar y Verificar la Solución</p> <p>Verifican el funcionamiento del sistema</p>	<p>Reviso y verifico que la solución diseñada sea la adecuada</p> <p>Ejecutamos la simulación del sistema y verificamos que los buzzer emita la nota Do y el otro la nota Fa de forma alternada y por tiempo indefinido. Analizamos que utilidad le podemos dar a este saber en otro contexto o situación.</p>

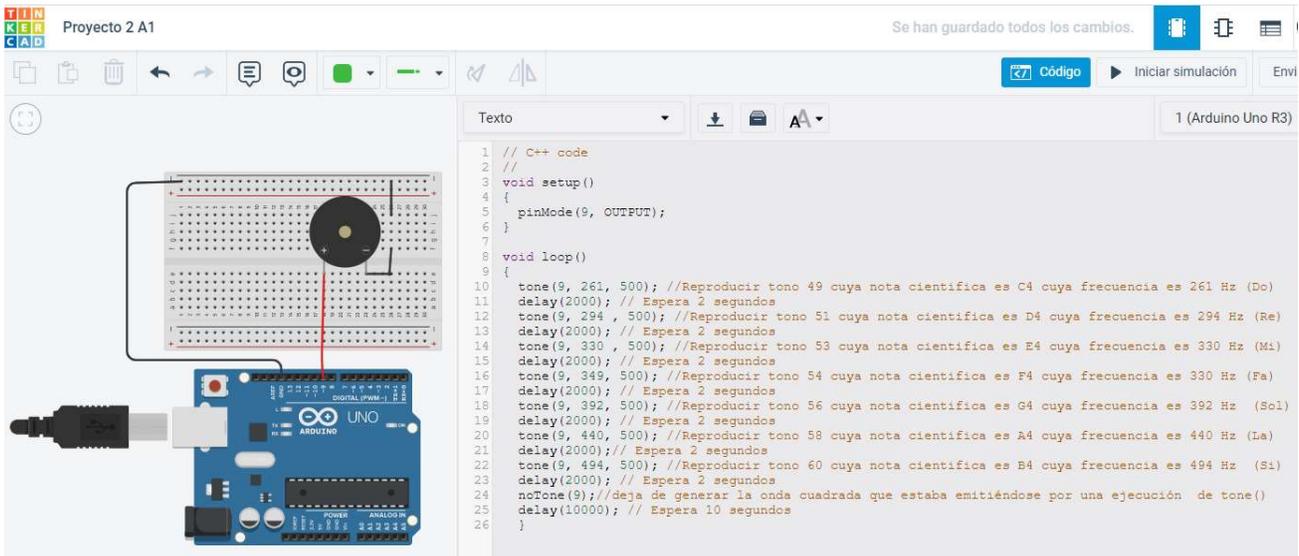


5. Problemática circuital y de programación. (Método de Pólya).

Realizar el circuito y la programación para hacer sonar en un buzzer la escala musical, al finalizar deberá mantenerse en silencio durante 10 segundos para volver repetir el bucle de forma indefinida.

Etapas del Método Pólya		Acciones
1	Comprender el Problema Lo expresan con sus propias palabras	Analizo y comprendo el problema Debemos realizar un circuito y la programación de un sistema que haga sonar en un buzzer la escala musical y al terminar silenciarse durante 10 segundos para volver a repetir la acción indefinidamente.
2	Elaborar un Plan Diseñan un plan para resolver el problema	Diseño un plan para resolver el problema Debemos elegir los componentes diseñar el circuito, seleccionar el pin PWM, realizar la conexión, investigar los Hz que corresponden a cada tono seleccionado y realizar la posible programación.
3	Ejecutar el Plan Ejecutan el plan diseñado	Realizo el circuito y la programación del plan previamente diseñado Utilizando TINKERCAD realizamos el circuito, seleccionamos el pin de salida PWM y ejecutamos la programación diseñada previamente.
4	Revisar y Verificar la Solución Verifican el funcionamiento del sistema	Reviso y verifico que la solución diseñada sea la adecuada Ejecuto la simulación del sistema y verifico que suene la escala musical, se silencie 10 segundos y vuelva a iniciarse el ciclo indefinidamente. Analizamos que utilidad le podemos dar a este saber en otro contexto o situación.

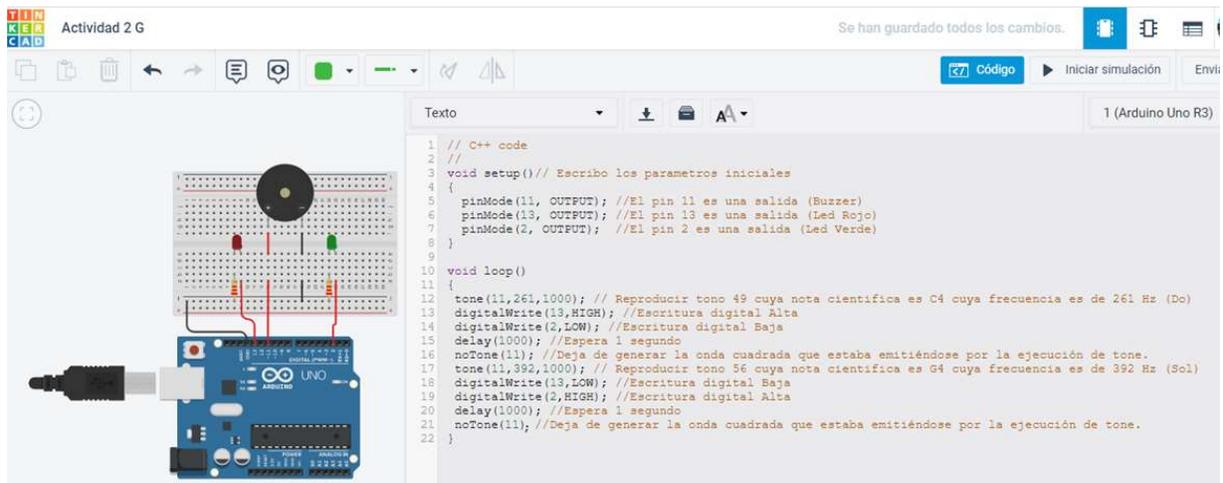
Alternativa posible:



6. Problemática circuital y de programación. (Método de Pólya).

Realizar el circuito y la programación para hacer sonar en un buzzer dos notas musicales (Do, y Sol) de forma alternada, cada vez y durante la nota “Do” se emita, tendrá que encender un led rojo y cada vez y durante que la nota “Sol” se emita se encenderá un led verde. Esta secuencia se realizara de forma permanente. Para el circuito te proponemos utilizar una salida digital PWM para el buzzer, una salida para digital para el led rojo y una salida digital para el led verde.

Etapas del Método Pólya		Acciones
1	Comprender el Problema Lo expresan con sus propias palabras	Analizo y comprendo el problema Debemos realizar un circuito y la programación de un sistema que haga sonar en un buzzer las notas musicales “Do” y “Sol”, teniendo en cuenta que durante el tiempo que suene la nota “Do” se encenderá un led rojo y durante el tiempo que suene la nota “Sol” se encenderá un led verde. Esta secuencia se repetirá de forma permanente e indefinida.
2	Elaborar un Plan Diseñan un plan para resolver el problema	Diseño un plan para resolver el problema Debemos elegir los componentes diseñar el circuito, seleccionar el pin PWM para el buzzer y los pines para los led realizar la conexión, investigar los Hz que corresponden a cada tono seleccionado y realizar la posible programación.
3	Ejecutar el Plan Ejecutan el plan diseñado	Realizo el circuito y la programación del plan previamente diseñado Utilizando TINKERCAD realizamos el circuito, seleccionamos el pin de salida PWM y los pines de salida para los led, y luego ejecuto la programación diseñada previamente.
4	Revisar y Verificar la Solución Verifican el funcionamiento del sistema	Reviso y verifico que la solución diseñada sea la adecuada Ejecutamos la simulación del sistema y verificamos que el circuito y la programación resuelvan la situación planteada. Analizamos que utilidad le podemos dar a este saber en otro contexto o situación.



7. Problema Tecnológico. “Problemática abierta” (Metodología Projectual).

En la situación problema de la secuencia de trabajo N° 1 estaba referida a una iniciativa de un municipio para la semaforización de las puertas de calle de las instituciones educativas a fin de prevenir posibles accidentes.

En este caso el municipio propone también instalar semáforos peatonales que incluyan la emisión de un sonido constante para indicar la vía libre para el cruce y alternado llegando el momento de detenerse, a fin de permitir el cruce peatonal seguro de personas no videntes.

Te invitamos a colaborar con esta iniciativa (con los saberes que poseas), diseñando un prototipo maquetizado de semáforo peatonal, que contenga el circuito y la automatización utilizando una salida PWM para el zumbador que emitirá el sonido para el cruce de personas no videntes y dos salidas digitales diferentes, una que alimentara un led rojo (Alto) y una que un led verde (Avanzar) para su señalización.



Te invitamos a proponer una solución a esta situación utilizando la metodología projectual para resolver problemas y de este modo realizar un prototipo maquetizado que dé respuesta a la problemática:

La tecnología adopta un esquema de trabajo basado en el método projectual, el cual posee diferentes etapas o fases para su concreción.

- **Percepción del problema**
- **Búsqueda de alternativas de solución**
- **Selección de la solución adecuada**
- **Diseño de la solución**
- **Organización del trabajo**
- **Construcción de modelos**
- **Evaluación y perfeccionamiento**

Por fines educativos y teniendo en cuenta las características de las y los estudiantes del ciclo básico, esta metodología proyectual, tomara otra forma. La fase o etapa denominada “Percepción del problema”, tomara una rol más orientativo definido por los contenidos, las expectativas de logro y las capacidades que las y los jóvenes deben desarrollar. Esta fase se redefinirá en dos fases simultáneas “Diseño de la situación Problemática” y “Planteo de la problemática”.

Método proyectual para estudiantes de Ciclo Básico

- Percepción del problema
 - Búsqueda de alternativas de solución
 - Selección de la solución adecuada
 - Diseño de la solución
 - Organización del trabajo
 - Construcción de modelos
 - Evaluación y perfeccionamiento
- [
- **Diseño de la situación problema**
 - **Planteo de la problemática**

Posible Procedimiento

- **Diseño de la problemática**

La acción comienza pensando una situación problema para las y los estudiantes. En el caso de este ejemplo, le dará respuesta a una iniciativa municipal de semaforización peatonal con un dispositivo sonoro para personas no videntes.



Las y los docentes involucrados se reúnen para pensar una situación problema para plantearle a sus estudiantes. Para su diseño deben tener en cuenta los ejes estructurantes, los contenidos y las capacidades que sus estudiantes deben desarrollar. Esta es una instancia en donde se articulan los espacios o módulos que conforman el taller. Para su planteo deben pensar un enunciado que contextualice una situación real que las y los jóvenes consideren útil para la vida.

- **Planteo de la Problemática**

Una vez diseñada la problemática, las o los docentes deben plantearse las a sus estudiantes de la forma más abierta posible. La descripción del problema será realizada de modo tal que pueda ser considerada desde diferentes perspectivas. Esto se refiere a que los y las estudiantes tengan una percepción global en relación al problema como etapa previa a la búsqueda de una solución.



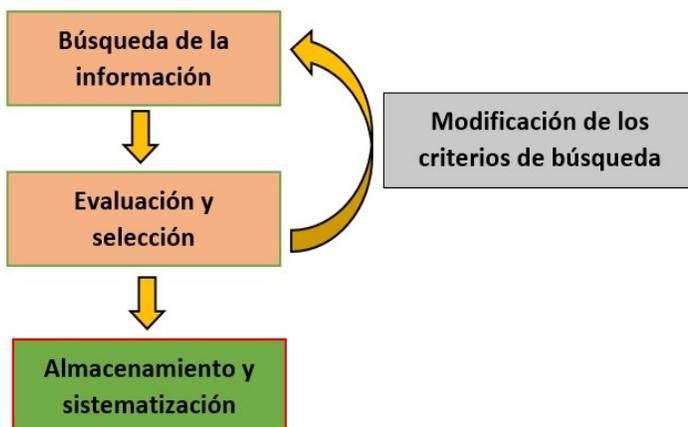
Este planteo de la situación a resolver, puede estar acompañado por documentos, imágenes, artículos periodísticos, videos y todo tipo de información que sirva para contextualizar la situación problema que deberán resolver.

- **Búsqueda de alternativas de solución**

Si bien los docentes deben hacer un andamiaje durante todo el proceso. En esta etapa será de suma importancia sus estrategias de mediación para lograr que los jóvenes puedan consensuar en una propuesta de solución lo más colectiva posible y que este dentro de su alcance.

Esta etapa del método proyectual implica la necesidad, que los estudiantes adquieran saberes relacionados con la búsqueda y análisis de información. Los procesos de búsqueda de información son complejos y cíclicos, e implican una serie de actividades tales como:

- a) Búsqueda, evaluación y selección de la información.
- b) Almacenamiento de resultados parciales.
- c) Comparación y análisis de la información obtenida.
- d) Modificación de los criterios de búsqueda: ampliar, especificar o redefinir los criterios.



- **Selección de la solución adecuada**

En esta instancia los estudiantes tendrán que seleccionar la solución que mejor cumpla con la problemática planteada por el o los docentes.

En esta fase de la metodología proyectual debe predominar la ubicuidad. Estamos hablando de la búsqueda de la solución que pueden apropiarse. La solución debe ser posible de realizar con los recursos que tenemos o que podemos llegar a tener.



En la selección de la solución, los docentes deberán realizar formulaciones a través de diálogos, realizando preguntas al grupo en relación a las características del prototipo que van a fabricar para dar respuesta al problema planteado.

- **La etapa de diseño**

El diseño es la parte creativa del proyecto tecnológico, porque es aquí donde, a partir de la información que las y los jóvenes han consultado y del conocimiento de la situación en la realidad, realizaran su propia respuesta al problema.

Involucra aprendizajes personales y colectivos y requiere de métodos tales como: croquis, bocetos y planos; diseñar la programación, cálculos y estimaciones (de costos, de magnitudes físicas, etc.); la elaboración de planes de acción, etc.



- **Organización del trabajo**

Una vez que las y los estudiantes han optado por una alternativa de solución y realizado el diseño, y antes de comenzar a trabajar en la fabricación, es conveniente que dediquen un tiempo a pensar en cómo organizar y sistematizar las tareas que les permitirán llevar adelante la solución seleccionada.



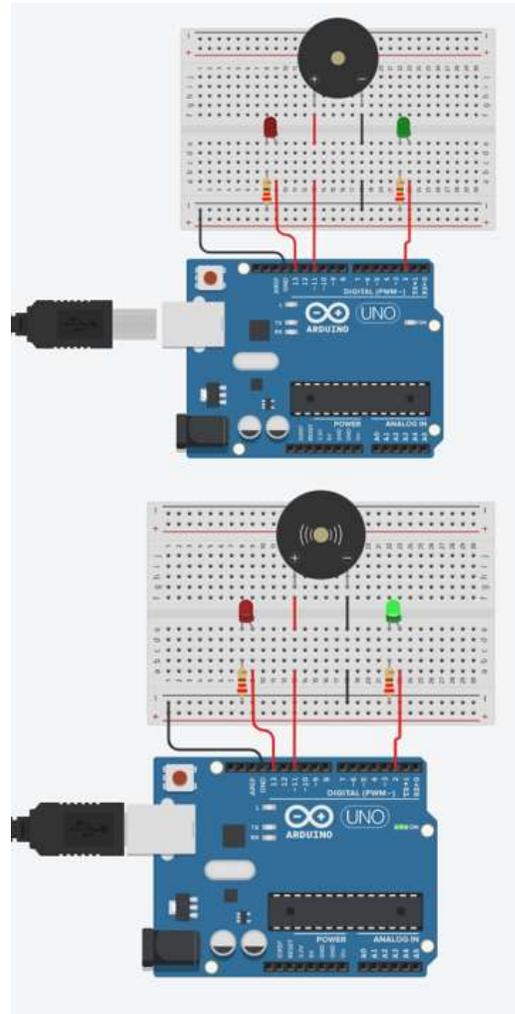
Esto quiere decir, planificar el modo que se llevara a cabo la fabricación del prototipo, detallando todas las tareas a realizar, su secuencia, el tiempo estimado, previendo los recursos necesarios para la ejecución del proyecto y la distribución y asignación de estas tareas entre los integrantes del equipo de trabajo.

- **Construcción de modelos**

Durante esta etapa las y los estudiantes deberán seleccionar y utilizar los materiales, herramientas, aplicar los sistemas involucrados, utilizar máquinas, instrumentos y sus distintos procedimientos, planificando su aplicación eficiente. Explicar a terceros cómo se desarrolla el trabajo. Aplicar técnicas manuales y técnicas digitales de fabricación, aplicaran los sistemas necesarios. Seleccionar caminos alternativos cuando aparezcan dificultades y solicitar ayuda cuando la necesiten.



```
1 void setup()// Escribe los parametros iniciales
2
3 pinMode(11, OUTPUT); //El pin 11 es una salida (Buzer)
4 pinMode(10, OUTPUT); //El pin 10 es una salida (Led Rojo)
5 pinMode(2, OUTPUT); //El pin 2 es una salida (Led Verde)
6
7 void loop()
8
9 digitalWrite(11,HIGH); //Escritura digital Alta (Rojo Prendido)
10 digitalWrite(2,LOW); //Escritura digital Baja (Verde Apagado)
11 delay(1000); //Espera 1 segundo
12 tone(11,200,1000); //Reproducir tono cuya frecuencia es de 200 Hz
13 digitalWrite(11,LOW); //Escritura digital Baja (Rojo Apagado)
14 digitalWrite(2,HIGH); //Escritura digital Alta (Verde Prendido)
15 delay(1000); //Espera 1 segundo
16 noTone(11); //Deja de generar la onda cuadrada que estaba emitiéndose por la ejecución de tone.
17 delay(100); //Espera 0,1 Segundos
18 tone(11,200,1000); //Reproducir tono cuya frecuencia es de 200 Hz
19 digitalWrite(11,LOW); //Escritura digital Baja (Rojo Apagado)
20 digitalWrite(2,HIGH); //Escritura digital Alta (Verde Prendido)
21 delay(1000); //Espera 1 segundo
22 noTone(11); //Deja de generar la onda cuadrada que estaba emitiéndose por la ejecución de tone.
23 delay(100); //Espera 0,1 Segundos
24 tone(11,200,1000); //Reproducir tono cuya frecuencia es de 200 Hz
25 digitalWrite(11,LOW); //Escritura digital Baja (Rojo Apagado)
26 digitalWrite(2,HIGH); //Escritura digital Alta (Verde Prendido)
27 delay(1000); //Espera 1 segundo
28 noTone(11); //Deja de generar la onda cuadrada que estaba emitiéndose por la ejecución de tone.
29 delay(100); //Espera 0,1 Segundos
30 tone(11,200,1000); //Reproducir tono cuya frecuencia es de 200 Hz
31 digitalWrite(11,LOW); //Escritura digital Baja (Rojo Apagado)
32 digitalWrite(2,HIGH); //Escritura digital Alta (Verde Prendido)
33 delay(1000); //Espera 1 segundo
34 noTone(11); //Deja de generar la onda cuadrada que estaba emitiéndose por la ejecución de tone.
35 delay(100); //Espera 0,1 Segundos
36 tone(11,200,1000); //Reproducir tono cuya frecuencia es de 200 Hz
37 digitalWrite(11,LOW); //Escritura digital Baja (Rojo Apagado)
38 digitalWrite(2,HIGH); //Escritura digital Alta (Verde Prendido)
39 delay(1000); //Espera 1 segundo
40 noTone(11); //Deja de generar la onda cuadrada que estaba emitiéndose por la ejecución de tone.
41 delay(100); //Espera 0,1 Segundos
42 tone(11,200,1000); //Reproducir tono cuya frecuencia es de 200 Hz
43 digitalWrite(11,LOW); //Escritura digital Baja (Rojo Apagado)
44 digitalWrite(2,HIGH); //Escritura digital Alta (Verde Prendido)
45 delay(1000); //Espera 1 segundo
46 noTone(11); //Deja de generar la onda cuadrada que estaba emitiéndose por la ejecución de tone.
47 delay(100); //Espera 0,1 Segundos
48 tone(11,200,1000); //Reproducir tono cuya frecuencia es de 200 Hz
49 digitalWrite(11,LOW); //Escritura digital Baja (Rojo Apagado)
50 digitalWrite(2,HIGH); //Escritura digital Alta (Verde Prendido)
51 delay(1000); //Espera 1 segundo
52 noTone(11); //Deja de generar la onda cuadrada que estaba emitiéndose por la ejecución de tone.
53 delay(100); //Espera 0,1 Segundos
54 tone(11,200,1000); //Reproducir tono cuya frecuencia es de 200 Hz
55 digitalWrite(11,LOW); //Escritura digital Baja (Rojo Apagado)
56 digitalWrite(2,HIGH); //Escritura digital Alta (Verde Prendido)
57 delay(1000); //Espera 1 segundo
58 noTone(11); //Deja de generar la onda cuadrada que estaba emitiéndose por la ejecución de tone.
59 delay(100); //Espera 0,1 Segundos
60 tone(11,200,1000); //Reproducir tono cuya frecuencia es de 200 Hz
61 digitalWrite(11,LOW); //Escritura digital Baja (Rojo Apagado)
62 digitalWrite(2,HIGH); //Escritura digital Alta (Verde Prendido)
63 delay(1000); //Espera 1 segundo
64 noTone(11); //Deja de generar la onda cuadrada que estaba emitiéndose por la ejecución de tone.
65 delay(100); //Espera 0,1 Segundos
66 tone(11,200,1000); //Reproducir tono cuya frecuencia es de 200 Hz
67 digitalWrite(11,LOW); //Escritura digital Baja (Rojo Apagado)
68 digitalWrite(2,HIGH); //Escritura digital Alta (Verde Prendido)
69 delay(1000); //Espera 1 segundo
70 noTone(11); //Deja de generar la onda cuadrada que estaba emitiéndose por la ejecución de tone.
71 delay(100); //Espera 0,1 Segundos
72 digitalWrite(11,LOW); //Escritura digital Baja (Rojo Apagado)
73 digitalWrite(2,HIGH); //Escritura digital Alta (Verde Prendido)
74 delay(1000); //Espera 3 Segundos
75 digitalWrite(11,LOW); //Escritura digital Baja (Rojo Apagado)
76 digitalWrite(2,HIGH); //Escritura digital Alta (Verde Prendido)
77 delay(1000); //Espera 3 Segundos
78 digitalWrite(11,LOW); //Escritura digital Baja (Rojo Apagado)
79 digitalWrite(2,HIGH); //Escritura digital Alta (Verde Prendido)
80 delay(1000); //Espera 3 Segundos
81 digitalWrite(11,LOW); //Escritura digital Baja (Rojo Apagado)
82 digitalWrite(2,LOW); //Escritura digital Baja (Verde Apagado)
83 delay(1000); //Espera 3 Segundos
84 }
```



• La etapa de evaluación y perfeccionamiento

En esta etapa, se revisara todo el proceso que llevo a la fabricación del prototipo, se compararan el resultado obtenido con los objetivos iniciales. Probaran su funcionamiento e incluso podrán sugerir cambios y mejoras en cada una de las fases anteriores. Analizaran las consecuencias deseadas y las no deseadas.

