

¿Cómo domar el voltaje?

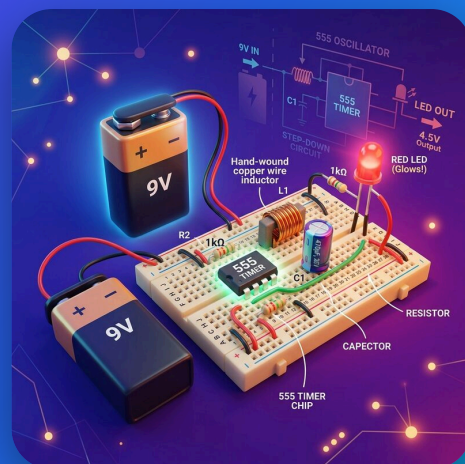


Bajar 9 V a 4 V sin desperdiciar energía en calor. Vamos a intentarlo a mano, después programando, y al final vamos a **ver** lo que pasa con un osciloscopio.

🎓 Secundaria

🕒 80-100 min

🔌 Electrónica + Programación





La gran idea

Tenés un depósito de agua (9 V) y querés llenar un vasito (un LED que necesita 2 V).



La forma antigua

Tapas la salida de agua haciendo mucha fuerza para dejar salir solo un poquito. Conclusión, **desperdiciaste energía**, que seguramente te hizo transpirar 🥵.

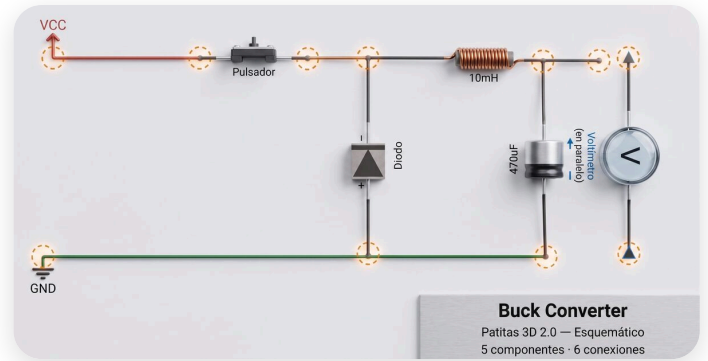


La forma moderna

Un mecanismo para abrir y cerrar una llave de paso **rapidísimo**. El vasito se llena de a gotitas y no se desperdicia casi nada de energía.



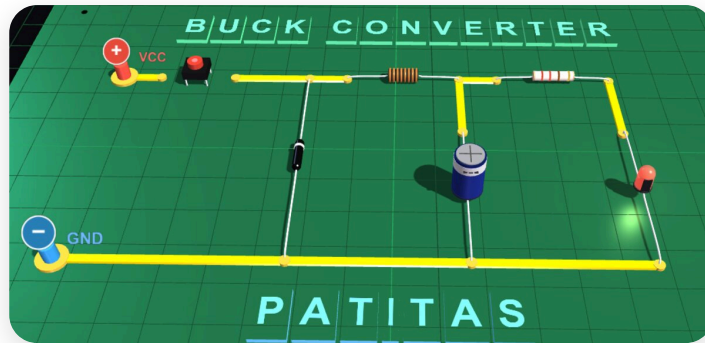
El secreto: si la llave está abierta la **mitad** del tiempo, sale la **mitad** del voltaje. A eso se lo llama **ciclo de trabajo (Duty Cycle)**. La cuenta es $V_{salida} = \text{ciclo} \times V_{entrada}$.



El circuito: pulsador, bobina, diodo y capacitor.

1

Los personajes del circuito 🔍



El circuito en Patitas 3D.

- **El depósito (la fuente de 9 V):** Suministra la energía del circuito.
- **La llave (el pulsador):** abre y cierra el paso de la corriente.
- **El amortiguador (la bobina):** odia los cambios bruscos. Guarda energía cuando el pulsador conduce y la sigue entregando cuando el pulsador se suelta.
- **El tanque intermedio (el capacitor):** se llena con los pulsos de la bobina y entrega un flujo parejo.
- **El camino de vuelta (el diodo):** cuando la llave se corta, le da a la bobina por dónde seguir circulando. Sin él, la bobina no tiene salida.
- **La carga (resistencia + LED):** lo que queremos alimentar con el voltaje reducido.

Valores que vamos a usar: fuente 9 V · bobina 470 mH · capacitor 22 μ F · resistencia 220 Ω · LED rojo.

2

Primer intento: con el dedo 🖐️

1. Descargar el circuito en la **Tienda de Patitas 3D** (buckConverter.json)
2. Cargar buckConverter.json en el simulador de Patitas 3D.
3. **Iniciar simulación** y mantener presionado el pulsador (tecla 1).

¿Qué pasó? El LED se prendió fuerte... y **se quemó**. Con el pulsador apretado todo el tiempo le llegan los 9 V enteros: por él pasan unos **40 mA (9V/220Ω)** cuando el LED solo aguanta 20.

Cambiá el LED y probá apretar y soltar lo más rápido que puedas.

Para pensar: ¿cuántas veces por segundo pudiste apretar? ¿Diez? ¿Quince? Un buck converter de verdad abre y cierra **miles** de veces por segundo. ¿Con el dedo se puede?

3

Segundo intento: programando el dedo

Si la mano es lenta, que lo haga la computadora. Seguramente pensemos que la solución es algo como este programa con bloques:

por siempre

● Mantener pulsador en J19 presionado

⌚ Esperar 0.0001 segundos

● Soltar pulsador en J19

⌚ Esperar 0.0001 segundos

Baja el tiempo de espera todo lo que quieras: 0,0001 ... 0,000001 ... **El voltaje no baja.**

Tampoco funciona: no es que el programa sea lento, es que **el bloque repetir funciona como en Scratch sin turbo: unas 30 veces por segundo.**

El bucle de programa hace que el simulador resuelva el circuito **unas 30 veces por segundo**. Entre vuelta y vuelta del ciclo **no se atienden los tiempos menores (Ej:0.000001) de pulsar y soltar**. Si el programa lo ignora, para el circuito es como si nunca hubiera pasado.



Es como si al filmar la velocidad de la cámara captara solo 30 fotos por segundo. Los movimientos intermedios se pierden, no se registran. Lo que hace entonces la cámara se llama **muestreo**, y es lo que le pasa acá al pulsador.

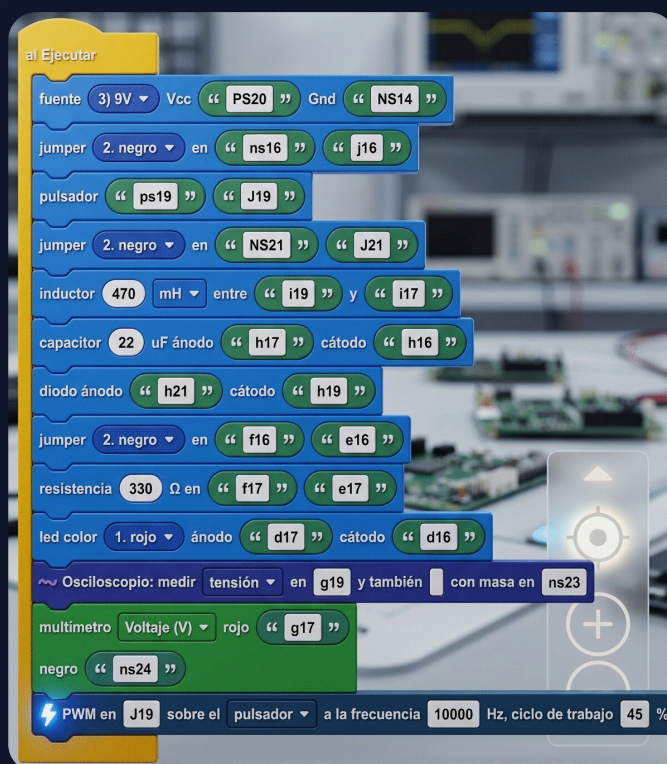
Para pensar: ¿por qué el multímetro tampoco sirve para medir esto? Pista: también toma muestras, y es bastante lento.

4

La solución: el bloque PWM (o repetir con turbo)✂

Si el problema es que el bloque de programa obliga a ralentizar al simulador, la solución es **un bloque que le pida directamente al simulador**. Eso hace el bloque propio de Patitas 3D PWM: no aprieta el pulsador, le **encarga** al motor de simulación que lo conmute él, en su propio reloj, muchísimo más rápido que el bloque repetir.

✂ PWM en J19 sobre el pulsador a la frecuencia 1000 Hz, ciclo de trabajo 50 %



Descargá el programa `buckConverter.xml` de Programación en Patitas 3D.

¡Ahora sí! El LED se prende con brillo normal y **no se quema**: por él pasan unos 10 mA en vez de 32. El multímetro sobre el LED marca alrededor de **1,9 V**, y sobre el capacitor unos **4,1 V**.

⚠ **Ojo:** el bloque PWM **no va adentro de un bucle (se repetiría el error del paso 3)**. Se pone una sola vez y sigue funcionando solo hasta que lo pares. ¡¡Cómo un pin pwm del **Arduino**!!

Probá distintos ciclos de trabajo y completá la tabla:

Ciclo de trabajo

Cuenta: ciclo $\times$ 9 V

Lo que mide el multímetro

25 %	2,25 V	
50 %	4,50 V	
75 %	6,75 V	

Para pensar: te va a dar un poco **menos** que la cuenta. ¿De dónde sale la diferencia? Del diodo. Cada vez que conduce, se "come" 0,7 V. La fórmula completa es $V_{sal} = ciclo \times V_{ent} - (1 - ciclo) \times 0,7 \text{ V}$.

5

La frutilla del postre: ver lo que pasa ~

El multímetro nos da números. Pero un número no cuenta toda la historia: adentro del circuito hay una onda rebotando mil veces por segundo. Para verla hace falta un **osciloscopio**.

1. En Patitas 3D la **perilla del multímetro** en la posición **OSC** hace aparecer el osciloscopio.
2. También tenemos bloques de programa que hacen este trabajo:

✗ Osciloscopio medir (corriente) en G19 y también () con masa en G17

- **CH1 (verde)** → **G19** — donde está la llave
 - **GND (negra)** → **NS18** — el riel negativo
3. Volvé la perilla a **V**: el osciloscopio **se queda** y el multímetro vuelve a medir. Ahora tenés los dos instrumentos a tu disposición.

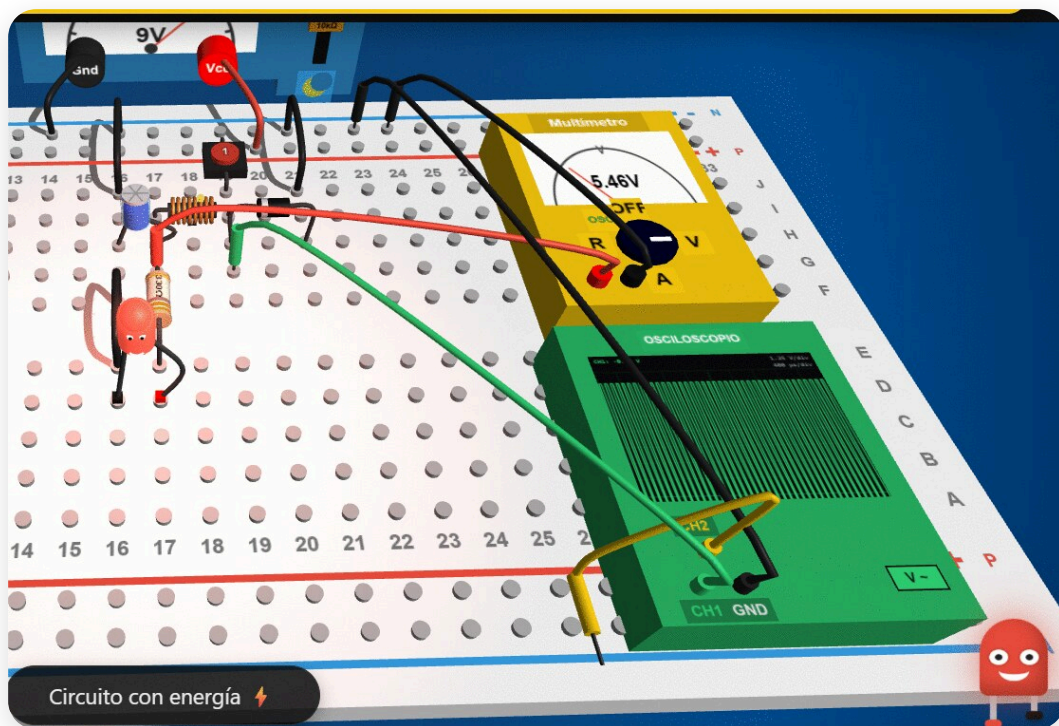
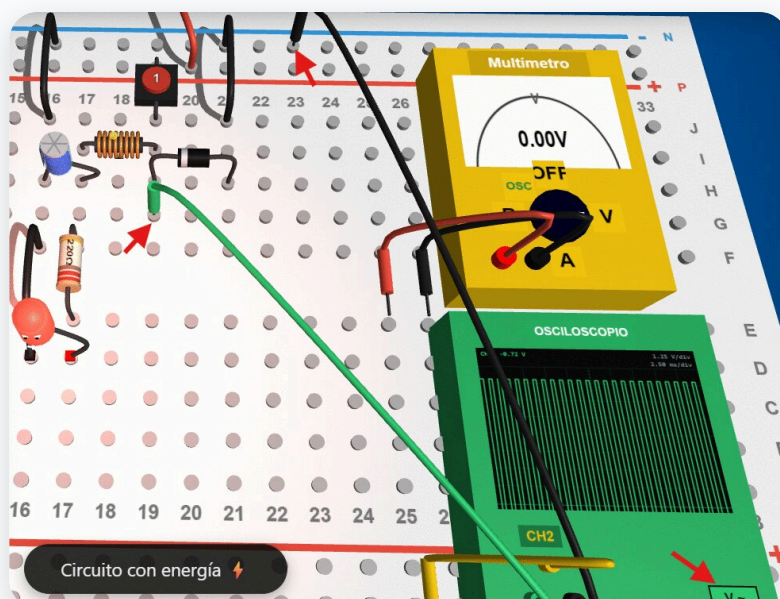


Figura 1: El osciloscopio graficando el comportamiento de la corriente en la bobina.

Qué conectar

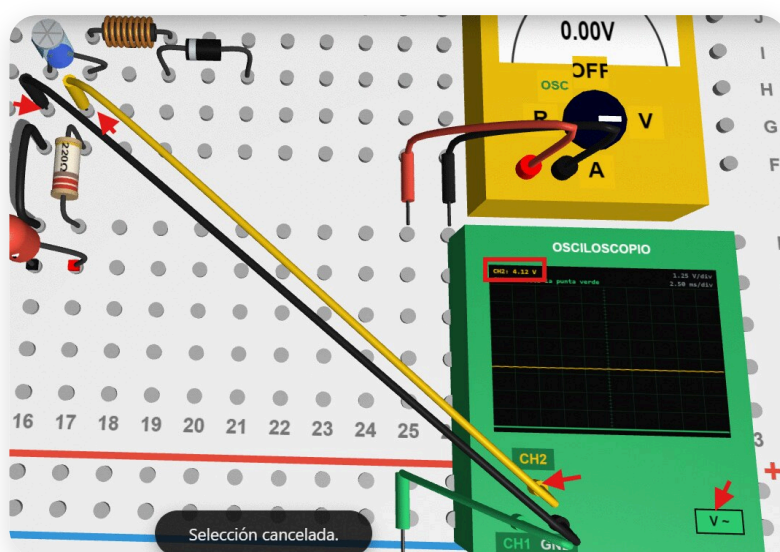
Qué vas a ver

CH1 a la salida del pulsador - GND
a Gnd de la placa - Selector del
Osciloscopio en V



Una **onda cuadrada**: 9 V cuando conduce el pulsador.

CH2 pata positiva del capacitor -
GND pata negativa - Selector del
Osciloscopio en V



Una línea **casi plana** en 4,1 V, con un temblorcito: el
rizado

 **Aprendé a leer la pantalla.** La retícula tiene 10 divisiones a lo ancho y 8 a lo alto. Arriba a la derecha dice cuánto vale cada cuadradito (V/div y ms/div). Contá cuadraditos y multiplicá: eso es leer un osciloscopio.

🌀 Y ahora, la corriente

Un osciloscopio **siempre mide tensión**. Pero el simulador tiene un modo pinza: apretá el botón **V ~ / A ~** del chasis y pasa a **corriente**. En ese modo las dos puntas **abrazan** el componente en vez de medir contra masa.

1. Poné **CH1** en **G19** y la **punta negra** en **G17**: las dos patas de la bobina.
2. Apretá el botón para pasar a modo **A**.

Aquí está el diente de sierra (Figura 1). La corriente de la bobina sube en rampa mientras la llave conduce y baja en rampa cuando conduce el diodo. **Eso** es la bobina guardando y devolviendo energía.

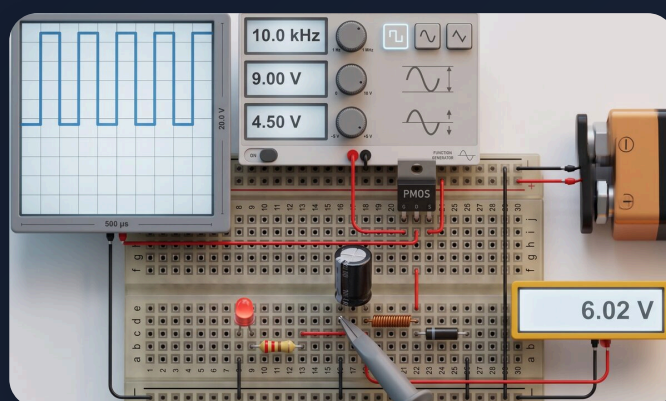
Para pensar: sobre la **misma** bobina, en modo V viste una onda cuadrada y en modo A un diente de sierra. ¿Por qué? ¿Qué es lo que realmente hace la bobina?

Probá cambiar...	¿Qué le pasa a la traza?
El ciclo de trabajo (25 % → 75 %)	
El capacitor (22 μF → 100 μF)	
La bobina (470 mH → 10 mH)	



Notas para docentes

- **Los dos primeros intentos fallidos son parte del proceso, no fracasos.** La actividad está diseñada para que las etapas 2 y 3 *sean parte del aprendizaje*. La 2 muestra que el problema es físico (la mano es lenta); la 3, conocer los límites de la herramienta misma (bloque repetir). Recién cuando se experimenta el problema, el bloque PWM se siente como una solución y no como un paso más de una receta. El próximo escalón puede ser Arduino y sus pines PWM.
- **El concepto que también se llevan de la etapa 3 es el concepto de muestreo.** Es el mismo que explica por qué un multímetro común miente frente a una señal PWM. Vale la pena nombrarlo.
- **Por qué la bobina es tan grande.** Un buck real usa 10–100 μH conmutando a 100 kHz. Acá usamos 470 mH a 1 kHz. Es una **exageración deliberada por las limitaciones propias del simulador**: la física es la misma, la escala de tiempo no. Si un alumno pone una bobina muy chica, el simulador ahora **avisa** que la salida no va a seguir al ciclo de trabajo.



Veamos el circuito equivalente en **Tinkercad** usando PMOS y Generador de funciones (PWM). En Patitas 3D, al no usar mosfet sino pulsador, hay que aumentar la resistencia de protección del LED para cuidarlo del pico de voltaje al comienzo de la simulación.

- **El diodo se nota.** La diferencia entre la cuenta y la medición no es un error del simulador: es la caída del diodo. En los convertidores buenos se lo reemplaza por un segundo transistor — se llama **rectificación síncrona**, y es un buen puente hacia el nivel siguiente.
- **Reciclaje:** desarmar un cargador USB de auto de 12 V. Es un buck converter puro: van a encontrar la bobina, el chip controlador y los capacitores.
- **Siguiente clase — la crisis:** cambien el LED por un motorcito. La salida se cae, porque nada vigila el voltaje. Ahí nace la necesidad de la **realimentación**, y del NE555 con un transistor haciendo el trabajo pesado.

Circuito listo para descargar en la Tienda: `buckConverter.json` (el armado en la protoboard para probar a mano el pulsador).

Circuito listo para descargar en Programación: `buckConverter.xml` (el programa de bloques con pwm y osciloscopio).

Hecho con  para el aula · Simulador **Patitas 3D**