



La Vela Mágica de Cumpleaños

¡Pide un deseo y sopla! Aprende el ciclo completo de la interactividad combinando eventos, variables de estado y condicionales para crear una vela digital que puedes apagar.

Cursos

- Grados 6-12

Materiales

- Celular, tableta o computador
- Conexión a Internet
- Cartón, tijeras y un par de elásticos para sujetar el celular al cartón

Descripción

Esta es una actividad de síntesis crucial que combina los conceptos de las lecciones anteriores: eventos, variables de estado y condicionales. Los estudiantes construirán una vela digital interactiva, lo que les permitirá ver en acción el ciclo completo de "Acción del Usuario -> Actualización de Estado -> Reacción del Programa".

Objetivos Educativos

- Sintetizar el uso de eventos, variables y condicionales en un solo programa.
- Comprender cómo estos tres componentes trabajan juntos para crear programas interactivos con estado.
- Implementar una funcionalidad de "interruptor" o "toggle" (encender/apagar) usando lógica booleana.
- Reforzar la arquitectura de un programa que separa la detección de eventos de la lógica de renderizado.

Inicio (10 minutos) - El Motor de la Interacción

1. Da la bienvenida a la clase: **"Hoy vamos a ensamblar el motor completo de la programación interactiva. Construiremos una vela de cumpleaños mágica que podemos soplar para apagarla y encenderla."**
2. Desglosa el ciclo: **"Pensemos en cómo funciona. ¿Cuál es la 'acción' que hacemos?"** (Soplar). **"Eso es nuestro evento. ¿Qué 'estado' de**

la vela cambia?" (De encendida a apagada). **"Esa es nuestra variable. ¿Y cuál es la 'decisión' que toma el programa?"** (Mostrar la llama o no). **"Ese es nuestro condicional. Hoy, veremos cómo estas tres piezas trabajan en perfecta armonía."**

El Ciclo Interactivo Completo

La mayoría de las aplicaciones que usas, desde videojuegos hasta redes sociales, se basan en un ciclo fundamental de tres pasos: 1. **Acción del Usuario (Evento):** Haces algo (un clic, un toque, un soplo). El programa lo detecta como un evento. 2. **Actualización de Estado (Variable):** En respuesta al evento, el programa actualiza su memoria (una variable) para recordar lo que sucedió. 3. **Reacción del Programa (Condicional):** El programa, que se está ejecutando constantemente, lee esa memoria y decide cómo cambiar lo que ves o escuchas. ¡Hoy, tú construirás este ciclo exacto!

La Lógica del Interruptor (Toggle)

Nuestra vela no solo se apaga, ¡también puede volver a encenderse con otro soplo! Esto se llama lógica de **"interruptor"** o **"toggle"**. En lugar de simplemente establecer la variable `velaEncendida` en FALSO, la establecemos en `(no velaEncendida)`. Este bloque especial invierte el valor booleano actual: * Si `velaEncendida` era VERDADERO, se convierte en FALSO. * Si `velaEncendida` era FALSO, se convierte en VERDADERO. ¡Con un solo bloque, podemos encender y apagar la vela con la misma acción!

La Arquitectura de Nuestro Programa

Nuestro programa tiene dos partes principales que trabajan en equipo: * **El Oyente de Eventos:** Está siempre "escuchando" el micrófono en segundo plano. Su única tarea es invertir el estado de la variable `velaEncendida` cuando detecta un soplo. No le importa la animación. * **El Motor de Animación (Bucle Principal):** Está siempre "dibujando" en la pantalla. Su única tarea es mirar la variable `velaEncendida` en cada ciclo y decidir: ¿dibujo la llama o dejo la pantalla en negro? La rama verdadera alterna **dos fotogramas** —la punta de la llama al azar y la punta al centro— con 200 ms entre ellos. Antes de cada punta se vuelve a dibujar la vela: ese redibujo es lo

que **apaga la punta anterior**. Si lo quitas para simplificar, las puntas se acumulan y la llama deja de titilar. No le importa de dónde vino el cambio.

Desarrollo (20-30 minutos) - Programando el Deseo

1. Ahora que los estudiantes tienen el mapa conceptual del ciclo interactivo, es hora de construirlo.
2. Guíalos a través de **las instrucciones para crear la vela y programar la interacción completa**, como se detalla a continuación. Es importante que noten cómo el bloque de evento y el bucle principal son independientes pero se comunican a través de la variable.

Cierre (5-10 minutos) - Siguiendo el Flujo de Datos

1. Una vez que todos puedan soplar sus velas, es hora de analizar el flujo de información.
2. Inicia la discusión: **"Describan el viaje de la información en nuestro programa. ¿Qué inicia el cambio? ¿Dónde se guarda ese cambio? ¿Quién lee ese cambio para actuar en consecuencia?"** (Evento -> Variable -> Condicional). Enfatiza que este patrón es la base de casi todas las aplicaciones que usan.

Reflexiona

Describe el "ciclo interactivo" en este proyecto. ¿Cuál es el evento, cuál es la variable de estado y cuál es el condicional que reacciona?

¿Qué hace exactamente el bloque `establecer velaEncendida en (no velaEncendida)`? ¿Por qué es más flexible que simplemente establecer la variable en FALSO?

Si quisieras que la vela solo se pudiera apagar una vez (y no volver a encenderse), ¿qué cambiarías en el bloque de evento?