



## OBJETIVOS

- Familiarizar a los estudiantes con la programación de microcontroladores y el manejo de GPIO.
- Desarrollar habilidades en el diseño y la implementación de circuitos electrónicos básicos.
- Introducir conceptos de tiempo real en sistemas embebidos a través de la medición de tiempos de reacción.
- Fomentar la creatividad en la implementación de soluciones de software y hardware.

## DESCRIPCIÓN

El juego de reacción simple es un proyecto interactivo donde los participantes deben presionar un botón tan rápido como puedan después de que se enciende uno de tres LEDs. El juego mide el tiempo desde que el LED se enciende hasta que el botón es presionado, ofreciendo una manera divertida y simple de introducir a los estudiantes en la programación de microcontroladores, el uso de GPIOs, y los conceptos básicos de la medición de tiempo en sistemas embebidos.

El juego consta de tres LEDs de colores diferentes con tres pulsadores, uno por cada LED, hay un cuarto pulsador que llamaremos botón de START. El juego funciona de la siguiente manera. El sistema arranca en un estado de espera. Un botón de START da inicio al juego y tres LEDs de colores diferentes hacen la siguiente secuencia de luces (111, 011, 001, 000). Cada bit representa un LED, cuando está en uno el LED asociado está encendido y con cero está apagado. De acuerdo con la secuencia, los LEDs se van apagando de izquierda a derecha hasta que todos están apagados. Luego de esto los 3 LEDs permanecerán apagados por un tiempo aleatoria entre 1 y 10 segundos. Transcurrido dicho tiempo, y también de forma aleatoria uno de los 3 LEDs se encenderá y el sistema deberá empezar a registrar el tiempo. El jugador debe reaccionar y presionar el pulsador asociado al LED que se encendió lo mas rápido posible. Al presionar el botón, el sistema registra el tiempo de reacción, apaga el LED, muestra el tiempo transcurrido en 4 displays de 7 segmentos (Uno para los segundos y 3 para las milésimas de segundo) y regresa al estado inicial de espera. Si el jugador no presiona el pulsador en una ventana de 10 segundos, el sistema vuelve al estado inicial de espera sin modificar el último tiempo desplegado en los displays. Presionar otros pulsadores diferentes al asociado al LED encendido genera una penalidad adicional de 1 segundo y en ningún caso detienen el conteo del tiempo.

## PROCEDIMIENTO

1. Revisión de Conceptos:
  - Repase los conceptos básicos de GPIO, incluyendo configuración de pines como entradas y salidas, lectura de entradas digitales, y escritura a salidas digitales.
  - Introducción a la generación de números aleatorios y medidas de tiempo en el microcontrolador.
2. Diseño del Circuito:
  - Conecte los LEDs y los resistores en serie a 3 pines de salida del microcontrolador.
  - Conecte los botones y configure los resistores de pull-down/up en el pin de entrada.
  - Conecte los 4 display de 7 segmentos. Recuerde que dependiendo del tipo de display (ánodo/cátodo común) deberá codificar los segmentos en el software. Controle el terminal común por medio de transistores.
  - Verifique las conexiones para asegurar que el circuito esté correctamente ensamblado.
3. Desarrollo del Software:
  - Programe el microcontrolador para controlar los LED y leer el estado de los botones.
  - Genere la secuencia de encendido de los LEDs
  - Implemente la lógica para generar un retardo aleatorio antes de encender el LED.
  - Mida el tiempo de reacción del usuario y muestre el resultado.
4. Pruebas y Ajustes:
  - Realice pruebas para asegurar que el sistema funciona como se espera.
  - Ajuste el software o el hardware según sea necesario para mejorar la experiencia del juego.
5. Documentación:
  - Documente el código fuente con doxygen, y escriba un reporte en formato IEEE con el proceso de diseño y las pruebas realizadas.
6. Entrega:
  - Suba el reporte y el código fuente en un archivo comprimido a la plataforma Moodle antes de la media noche del viernes 29 de marzo.

## EVALUACIÓN

Criterio	Excelente (90-100%)	Satisfactorio (70-89%)	Insuficiente (0-69%)
Diseño del Circuito (10%)	Circuito correctamente ensamblado y funcional.	Pequeños errores en el circuito que no afectan la funcionalidad principal.	Errores críticos en el diseño del circuito que impiden la funcionalidad.
Calidad del Código (30%)	Código bien organizado, comentado y eficiente.	Código con algunos comentarios y organización adecuada, pero con margen de mejora en eficiencia.	Código desorganizado, escasamente comentado y poco eficiente.
Funcionalidad del Juego (30%)	El juego funciona perfectamente según la descripción.	El juego funciona bien, pero con pequeñas desviaciones de la descripción original.	El juego no funciona o se desvía significativamente de la descripción.
Creatividad y Mejoras (5%)	Implementación de características adicionales o mejoras significativas en el juego.	Algunas mejoras o características adicionales que enriquecen la experiencia del juego.	No hay mejoras ni características adicionales.
Reporte y Sustentación (25%)	Documentación completa, incluyendo diseño, código, y pruebas. Presentación clara y detallada.	Documentación y presentación adecuadas, pero con espacio para detalles adicionales.	Documentación y presentación incompletas o poco claras.