

LABORATORIO # 2

PRESENTADO POR:

- **Cristian Mora**
- **Daniel Avila**
- **Yeison Lara**

Presentado a :

- **Camilo Camacho**

1. INTRODUCCIÓN

En este laboratorio se desarrolla un sistema de generación y verificación de un TOKEN numérico de un solo uso en dos entornos que operan de forma sincronizada:

1. **PC:** Se ejecuta un programa en C que, al iniciar, muestra un recuadro (cuadrado) en pantalla y pone en marcha un temporizador. Este programa permite que el usuario ingrese el TOKEN generado y, mediante la operación XOR con una clave fija, verifica la validez del mismo.
2. **Microcontrolador (STM32):** De manera autónoma, el microcontrolador inicia su temporizador al detectar el recuadro impreso en la pantalla (mediante un sensor digital o señal de sincronización externa) y, cada 30 segundos, actualiza el TOKEN realizando la misma operación XOR con la clave. El valor menos significativo del TOKEN se muestra continuamente en 8 LEDs.

Ambos procesos utilizan un temporizador que cuenta intervalos de 30 segundos para modificar el TOKEN mediante una operación XOR entre una clave fija y un operando calculado a partir del tiempo transcurrido. La sincronización se efectúa inicialmente cuando el usuario presiona ENTER en el programa del PC, lo que imprime un recuadro en la pantalla; dicho recuadro sirve para iniciar el conteo en ambos sistemas. Aunque ambos entornos ejecutan la misma operación y comparten el mismo criterio de cambio de clave, no existe comunicación directa entre ellos durante la operación. Cada sistema calcula y actualiza su TOKEN de manera independiente, lo que permite al usuario validar manualmente el TOKEN ingresado en el PC frente al valor mostrado en los LEDs del microcontrolador.

2. DESARROLLO DEL PROCESO

2.1. Proceso en PC

El programa en C desarrollado para el PC realiza las siguientes tareas:

1. **Sincronización e Inicialización:**
 - El programa solicita al usuario que pulse **ENTER** para sincronizar ambos sistemas.

- Al presionar ENTER, se ejecuta la función `imprimeCuadrado()`, la cual dibuja en pantalla un recuadro formado por caracteres (código 219). Este recuadro es el estímulo visual que, mediante un mecanismo externo (por ejemplo, un sensor digital colocado frente al monitor), sirve para iniciar el temporizador en el microcontrolador.
- Se utiliza `gettimeofday()` para capturar el instante de sincronización, almacenándolo en la variable `start`.

2. Cálculo del TOKEN:

- El programa mantiene un temporizador que se va actualizando con `gettimeofday()` en cada iteración del bucle principal.
- Se solicita al usuario ingresar el TOKEN (mediante `scanf`), lo que simula el ingreso manual de la clave.
- Se calcula el tiempo transcurrido desde la sincronización (en milisegundos) y, a partir de él, se determina el número de intervalos de 30 segundos transcurridos:
- Si el TOKEN ingresado coincide con el TOKEN calculado (mediante la comparación aritmética en el programa), se concede el acceso; de lo contrario, se rechaza.

3. Verificación y Retroalimentación:

- Se muestra en pantalla el tiempo transcurrido, el intervalo actual, la clave fija y el TOKEN generado.
- Se utiliza `system("pause")` para permitir al usuario visualizar el resultado antes de reiniciar el ciclo.

2.2. Proceso en el Microcontrolador (STM32)

El microcontrolador, de manera independiente, ejecuta un proceso similar basado en temporizadores:

1. Sincronización:

- El microcontrolador inicia su temporizador en el mismo instante en que se imprime el recuadro en la pantalla (evento disparado externamente, por ejemplo, mediante un sensor digital que detecta el recuadro).
- Se almacena el instante de sincronización para el cálculo del tiempo transcurrido.

2. Cálculo y Actualización del TOKEN:

- Al igual que en el programa del PC, el STM32 cuenta los intervalos de 30 segundos mediante un temporizador (por ejemplo, `SysTick` configurado para generar interrupciones periódicas, acumulando 30000 ms).

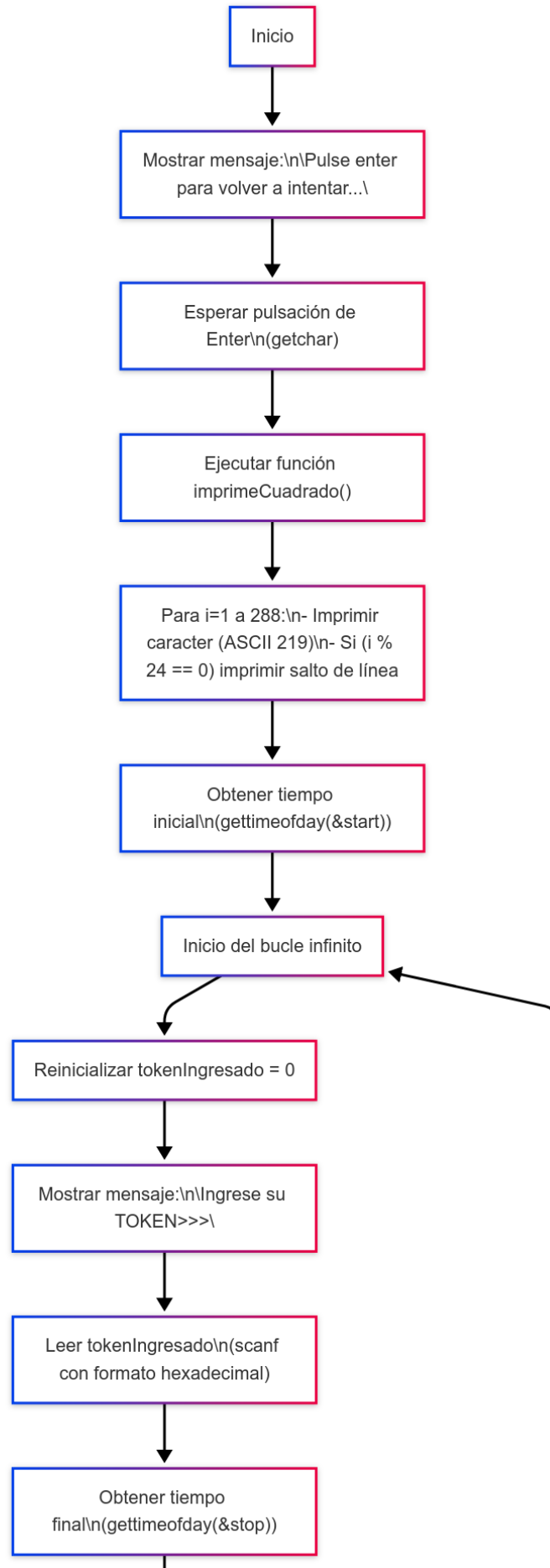
- Cada 30 segundos, se actualiza el TOKEN realizando la operación XOR entre la **clave fija** (misma que en el PC) y un operando calculado a partir del tiempo transcurrido
- En el primer intervalo (cuando el operando es 0), el TOKEN es igual a la clave fija.

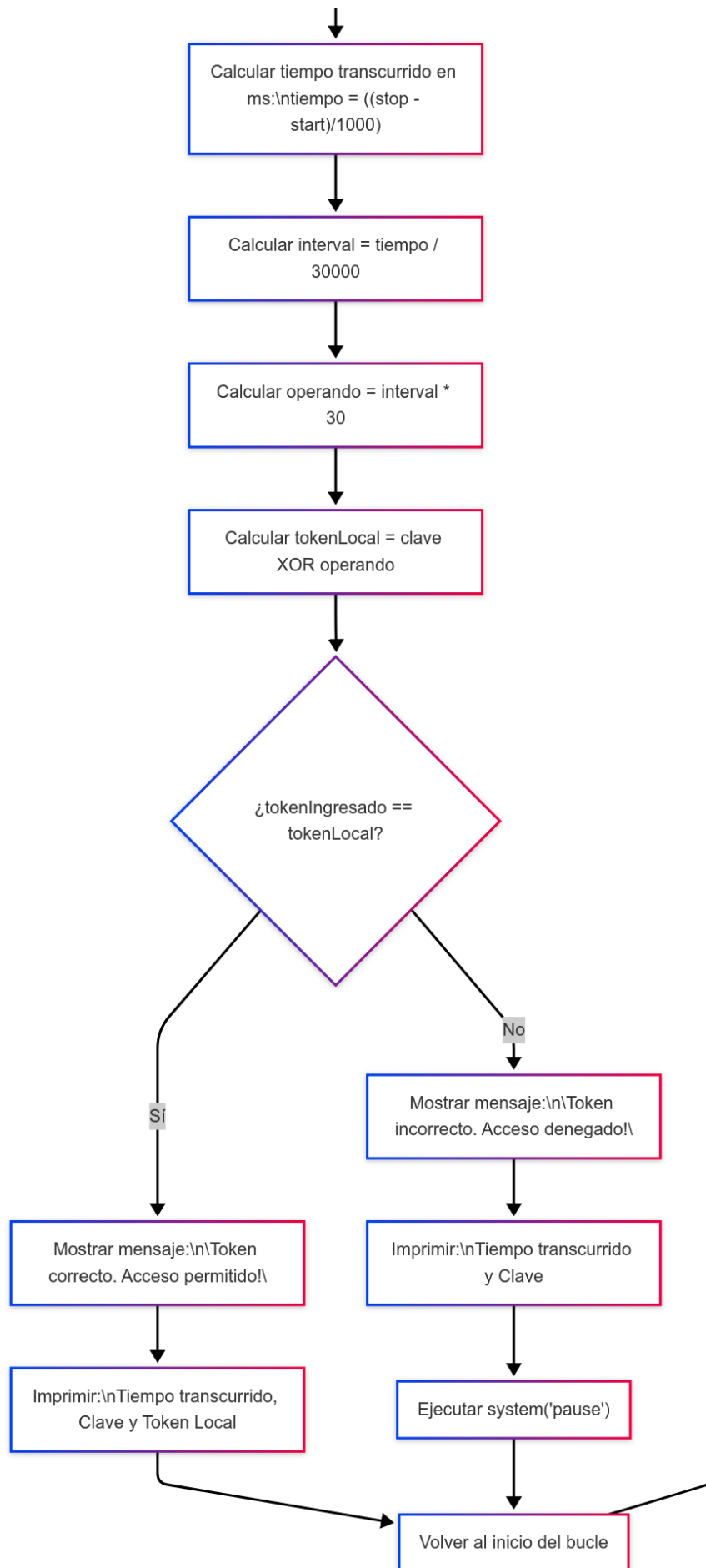
3. Visualización en LEDs:

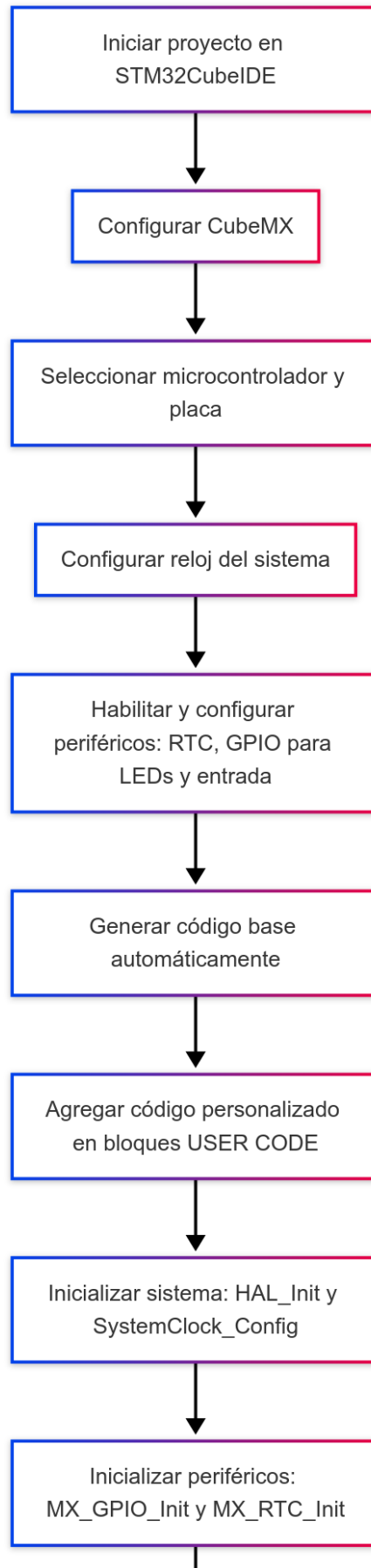
- El microcontrolador extrae el **byte menos significativo** del TOKEN (LSB) y lo muestra continuamente en 8 LEDs, asignando cada bit a un LED.
- Esta visualización se actualiza automáticamente cada vez que se genera un nuevo TOKEN, lo que ocurre cada 30 segundos.

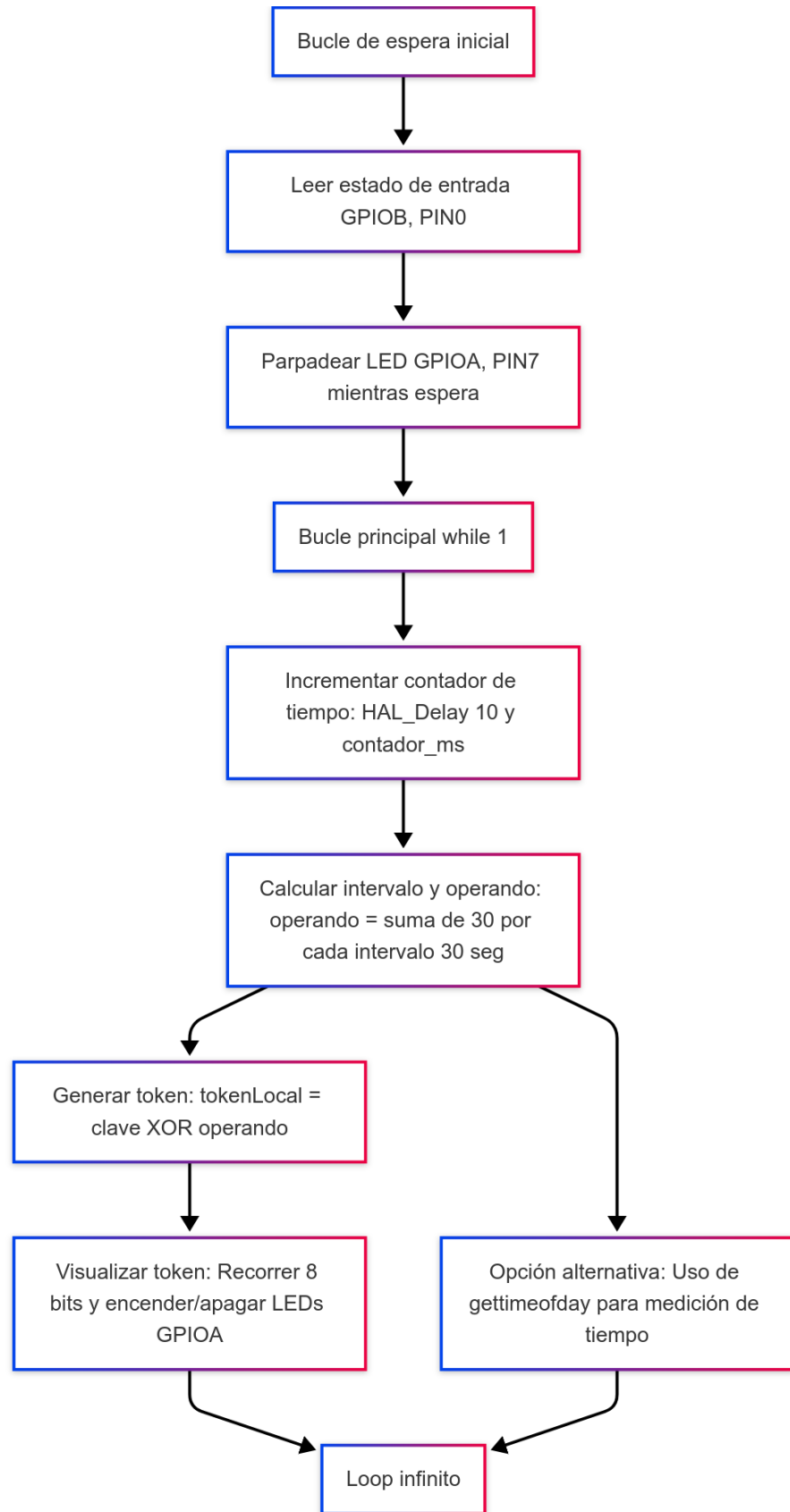
3. DIAGRAMA DE FLUJO

Por medio de este diagrama podemos ver flujo de ambos sistemas implementados tanto en stm32 como en un programa en C de PC (DEV C++)









4. RESULTADOS

- **Sincronización Conjunta:**
Ambos sistemas (PC y microcontrolador) se inician al mismo tiempo mediante la pulsación de ENTER en el PC y la impresión del recuadro, que sirve como señal de sincronización para el microcontrolador.
- **Temporización Precisa:**
Tanto el programa en C como el firmware del STM32 utilizan temporizadores que permiten actualizar el TOKEN de manera precisa cada 30 segundos.
- **Operación XOR Consistente:**
La misma operación XOR se aplica en ambos entornos para generar el TOKEN a partir de la clave fija y el operando derivado del tiempo transcurrido. Esto garantiza que, en condiciones ideales, ambos sistemas calculen valores equivalentes.
- **Validación del TOKEN:**
El programa en C solicita al usuario ingresar el TOKEN y lo compara con el TOKEN generado localmente. La verificación se realiza mostrando mensajes de "Acceso permitido" o "Acceso denegado", según corresponda.
- **Visualización en el Microcontrolador:**
El STM32 muestra constantemente el byte menos significativo del TOKEN en 8 LEDs, permitiendo una verificación visual continua y sincronizada con el tiempo.

5. CONCLUSIONES

1. **Sincronización Independiente pero Consistente:**
La estrategia de sincronizar ambos sistemas a través de un estímulo visual (el recuadro impreso) y el inicio del temporizador permite que tanto el PC como el microcontrolador actualicen el TOKEN de forma paralela sin necesidad de comunicación directa.

2. **Operación XOR para Claves de un Solo Uso:**

La implementación de la operación XOR entre una clave fija y un operando derivado del tiempo asegura que el TOKEN varíe cada 30 segundos, ofreciendo un método simple y efectivo para generar claves de un solo uso.

3. **Implementación Paralela en Dos Entornos:**

El desarrollo del programa en C en el PC y el firmware en el STM32, a pesar de operar de forma independiente, utilizan la misma lógica de temporización y cálculo del TOKEN, lo que facilita la validación manual del sistema.

4. **Visualización y Retroalimentación Inmediata:**

La visualización continua del byte menos significativo del TOKEN en los LEDs del microcontrolador y la retroalimentación textual en el PC permiten una verificación inmediata del funcionamiento del sistema.