

UNIVERSIDAD DE COSTA RICA

ESCUELA DE INGENIERÍA ELÉCTRICA



UNIVERSIDAD DE
COSTA RICA

IE-0523 CIRCUITOS DIGITALES II

Tarea 1

Profesor:

Enrique Coen Alfaro

Alumno:

Diego Alfaro Segura C20259

7 de septiembre de 2024

1. Resumen

En el presente diseño se propone una arquitectura y lógica de unidad de control para el portón de un estacionamiento, el cual debe permitir o prohibir el paso de automóviles según una clave ingresada y señales de sensores de entrada/salida. El controlador emplea dos señales de salida distintas para la apertura y cierre. Además, si se ingresa una clave incorrecta tres veces o si se detecta que un automóvil está ingresando mientras otro sale, se activará una señal de alarma correspondiente al tipo de alarma y se bloquea el sistema. Particular a este diseño (y a diferencia de las especificaciones dadas) se adicionó una entrada para indicar el ingreso de la clave al controlador con el propósito de ejemplificar un comportamiento real y facilitar las simulaciones. El controlador se diseñó como una máquina de estados sincrónica por medio del lenguaje de descripción de hardware Verilog y se verificó el funcionamiento de este por medio del visualizador de ondas GTKWave. Los resultados de las simulaciones muestran el funcionamiento correcto del diseño ante cuatro situaciones planteadas: el ingreso inmediato de la clave correcta, el ingreso de la clave correcta en el tercer intento, el ingreso de 3 claves incorrectas y la detección de un carro ingresando antes de que salga el anterior. Del diseño se logró obtener un código conductual válido para el controlador especificado con expectativas de que sea capaz de continuar las fases de desarrollo hasta la implementación en un chip. Finalmente, se reconoce que existen implementaciones potencialmente más simples (como el uso de contadores internos o designar una clave que indique clave no ingresada) que podrían mejorar el diseño o futuros diseños similares.

2. Descripción Arquitectónica

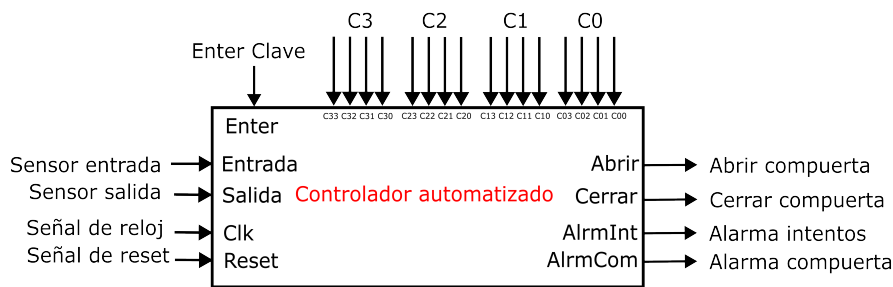


Figura 1: Diagrama de bloques del controlador diseñado.

En la Figura 1 se muestra el diagrama de bloques del controlador, donde se señalan las entradas y salidas del diseño. De las especificaciones se toman las entradas de los sensores y los 16 bits de la clave codificada en BCD. Esta última se muestra dividida en los 4 dígitos de la clave, 4 bits cada uno, pero en la implementación por código se asignan como una misma variable. Además se tienen las señales de reloj y reset necesarias para una máquina de estados. Se adiciona la señal de Enter para reconciliar dos problemas notados durante el diseño del controlador. Los problemas provienen del hecho que en las especificaciones no se indica como se le informa a la unidad cuando se ha ingresado una clave, lo cual causa que (al detectar un vehículo) se deban ingresar las claves a todo momento y cambiarlas entre los ciclos de reloj de la unidad de control (lo cual no es realista para circuitos cuyos ciclos de reloj son miles de veces más rápidos que las reacciones humanas). Por esto se tienen los problemas de que el controlador diseñado tendría operación no apta y las simulaciones serían difíciles de hacer para las transiciones entre estados. Una solución no empleada es la de escoger un código que signifique el estado de “idle”, pero esto reduce la cantidad de claves disponibles y (hasta donde se analizó el caso) no evita el problema una vez se ingresó la clave. En cambio, se consideró el botón de Enter como alternativa. Por último, se tienen las salidas para controlar la apertura, cierre e inicio de alarma.

Para el desarrollo de la lógica del controlador se emplea la lógica mostrada en el diagrama ASM de la Figura 2, donde los círculos definen los estados, los rombos donde se evalúa una entrada y rectángulos las salidas que se activan. Antes de describir la lógica empleada se debe definir el bloque de evaluación llamado BCD, pues este no es una entrada, sino es la comparación entre la clave ingresada y la clave correcta. Este bloque es equivalente a los bloques vistos en la Figura 3. Cabe recalcar que el diagrama de esta figura es para el caso donde la clave válida es 0259 en BCD, este es el caso para las simulaciones hechas pero en caso de cambiar este parámetro cambia la forma del bloque. Se escogió representarlo de esta manera pues el diagrama completo resultaría muy extenso y difícil de leer.

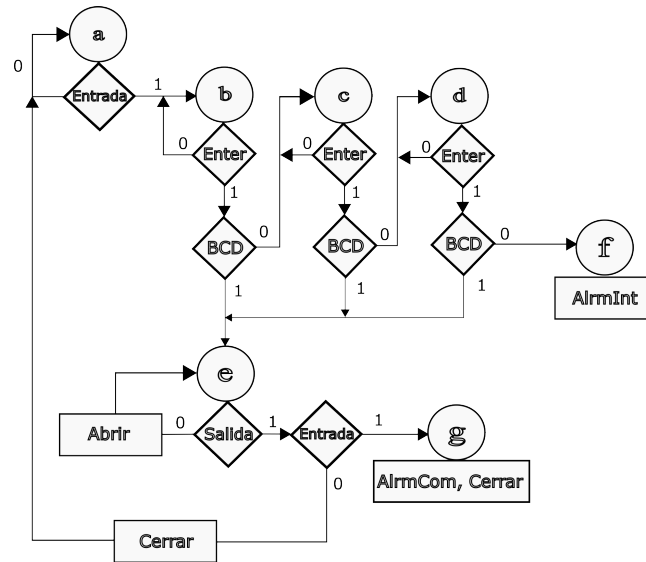


Figura 2: Diagrama ASM de la máquina de estados del controlador.

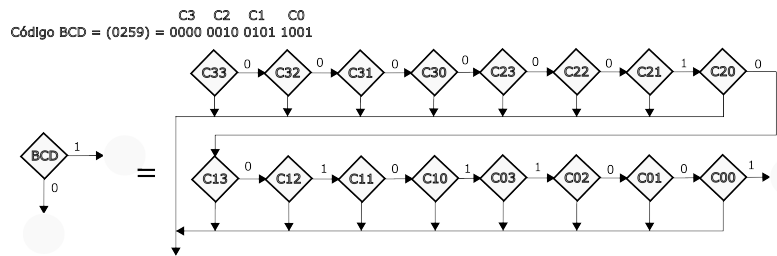


Figura 3: Diagrama ASM del bloque de comparación BCD.

Los estados asignados son los siguientes:

- **a : No hay automóvil ingresando, esperando señal de entrada.** Este es el estado inicial, a este estado se llega desde sí mismo si no han llegado autos, si se da una secuencia exitosa de paso de un auto o si se da la señal de reset.
- **b : Auto en entrada, esperando primer intento de clave.** Este estado inicia cuando se activa la señal de entrada por un auto, se espera señal de Enter (sino se mantiene en el estado) y se evalúa la clave ingresada por medio del bloque BCD. Si la entrada es correcta pasa a estado e, sino pasa a estado c.
- **c : Auto en entrada, esperando segundo intento de clave.** Ocurre cuando se falla el primer intento de la clave, se espera al segundo intento de la misma manera que en el estado b, al igual que si la clave es correcta se pasa al estado e, sino se pasa al estado d.
- **d : Auto en entrada, esperando tercer intento de clave.** Ocurre cuando se falla el segundo intento de la clave, se espera la clave, si es correcta se pasa al estado e, sino se pasa al estado f.
- **e : Clave aceptada, esperando salida del automóvil.** Este último estado se da cuando la clave ingresada es la correcta, se activa la señal para abrir el portón hasta que se detecte que está saliendo por medio de la señal Salida, en cuyo caso se deja de abrir. Si en este punto se activan ambas señales de Salida y Entrada, se cumple las condiciones de alarma y se pasa al estado g. Si solo se activa la señal de Salida se activa la señal de Cerrar para cerrar el portón y se retorna al estado a. Debido a que no se especifica cual debe ser la duración de la señal para cerrar se usó como una salida que se activa momentáneamente, pero se puede cambiar a que se mantenga en el estado a si se debe mantener por una mayor duración. El diseño usado asume que al ser momentánea el proceso de cerrado podrá finalizar aún si se desactiva la señal de cerrado.
- **f : Estado de alarma por intentos.** Sucede cuando se ingresan 3 claves incorrectas, se activa la alarma por cantidad de intentos denominada AlrmInt. Se bloquea el sistema y solo cambia de estado por medio de la señal de reset.

- **g : Estado de alarma por compuerta.** Sucede cuando se detecta la señal de entrada simultáneamente con la señal de salida cuando se está abriendo el portón. Activa la alarma por compuerta (AlrmCom) y se bloquea. Se interpretó el bloqueo de este caso como cerrar el portón, lo cual tiene sentido pues en caso de que un auto intente saltarse la verificación lo más prudente es negarle el paso, por esto se activa la señal Cerrar. La única forma de salir de este estado es por medio de la señal de reset.

Los estados se codificaron de la siguiente manera (Tabla 1), con el objetivo de asignar un flip-flop a cada estado y así reducir la posibilidad de que se den condiciones de carrera durante la operación del chip. Se agrega el equivalente de dicha codificación en hexadecimal para tener mayor facilidad de lectura de los estados en los diagramas de tiempo de la Sección 5.

Cuadro 1: Asignación de estados

Estado	Codificación	Equivalente en hexadecimal
a	0000001	01
b	0000010	02
c	0000100	04
d	0001000	08
e	0010000	10
f	0100000	20
g	1000000	40

La estructura del código describe una máquina de estados que emplea flip-flops para actualizar los estados en el flanco creciente de la señal de reloj, mientras que guarda las entradas en el flanco decreciente. Se tiene memorias de entradas y estados, al igual que secciones combinacionales para la lógica de calculo de próximo estado y lógica de cálculo de salidas. Por otra parte, se configuró la señal de Reset de tal manera que tome acción en su flanco creciente.

3. Plan de pruebas

Para verificar el funcionamiento típico se emplearon cuatro tipos de pruebas que cubren las situaciones generales que pueden darse durante su operación normal. Según el enunciado se asignó la clave correcta como 0259.

3.1. Prueba #1, funcionamiento normal básico

Se comprueba que el controlador abra el portón y lo cierre posterior a que un automóvil entrante ingrese la clave correcta en su primer intento. Primeramente se inicia la señal de Entrada indicando que llegó un automóvil y se ingresa la clave correcta para luego activar la señal de Enter. La señal de Enter se activa por un tiempo consistente para simular el comportamiento de un botón de enter de teclado. Según el comportamiento normal se desactiva la señal de Entrada del automóvil y un tiempo después se activa la señal de Salida del mismo. Los resultados esperados de esta prueba son la transición de estados de forma $a \Rightarrow b \Rightarrow e \Rightarrow a$ y que se de la señal de Abrir el portón hasta que el automóvil llegue a activar la señal de salida, punto en el cual se activa la señal Cerrar momentáneamente.

```
// Entra un carro
#2 Entrada = 1;

// Se ingresa el código correcto
#7 Clave = {4'd0, 4'd2, 4'd5, 4'd9};
#3 Enter = 1; #7 Enter = 0;

// Sale el carro, ya no esta ingresando clave.
#11 Entrada = 0; Clave = {4'd0, 4'd0, 4'd0, 4'd0};
#15 Salida = 1;
```

3.2. Prueba #2, ingreso de pin incorrecto menos de 3 veces

En esta prueba se comprueba que se permita el paso del vehículo si ingresa claves incorrectas menos de 3 veces. Se hace un procedimiento similar al de la prueba anterior, pero se ingresa claves incorrectas en los primeros dos intentos y la correcta en el tercero. Esto significa que se cambian los dígitos de la clave ingresada y se vuelve a enviar la señal de Enter 2 veces. Se espera la transición de estados $a \Rightarrow b \Rightarrow c \Rightarrow d \Rightarrow e \Rightarrow a$, con las mismas salidas que la prueba anterior.

```
// Entra un carro
#10 Entrada = 1; Salida = 0;

// Se ingresa el código correcto en el tercer intento
#4 Clave = {4'd0, 4'd2, 4'd5, 4'd1}; #3 Enter = 1; #7 Enter = 0;
#5 Clave = {4'd0, 4'd6, 4'd5, 4'd5}; #3 Enter = 1; #7 Enter = 0;
#6 Clave = {4'd0, 4'd2, 4'd5, 4'd9}; #3 Enter = 1; #7 Enter = 0;

// Sale el carro
#7 Entrada = 0; Enter = 0; Clave = {4'd0, 4'd0, 4'd0, 4'd0};
#8 Salida = 1;
```

3.3. Prueba #3, ingreso de pin incorrecto 3 veces

En esta prueba se verifica que el controlador entra en bloqueo e inicia la alarma cuando se ingresa una clave incorrecta en los 3 intentos. Se repite el procedimiento de la prueba anterior pero nunca se ingresa la clave correcta. Se espera la transición de estados $a \Rightarrow b \Rightarrow c \Rightarrow d \Rightarrow f$ y se espera que al ingresar la tercera clave incorrecta se active la señal de alarma por intentos. Una vez en este estado se mantendrá en el hasta que se activa la señal de reset, donde se espera que se retorne al estado a.

```
#10 Entrada = 1; Salida = 0;

// No se ingresa el código correcto
#4 Clave = {4'd0, 4'd2, 4'd5, 4'd10}; #3 Enter = 1; #7 Enter = 0;
#5 Clave = {4'd0, 4'd6, 4'd5, 4'd5}; #3 Enter = 1; #7 Enter = 0;
#6 Clave = {4'd0, 4'd3, 4'd5, 4'd9}; #3 Enter = 1; #7 Enter = 0;

// Reset de las alarmas
#14 Reset = 1; Entrada = 0; Clave = {4'd0, 4'd0, 4'd0, 4'd0};
#1 Reset = 0;
```

3.4. Prueba #4, alarma de bloqueo

En esta última prueba se verifica que si se detectan las señales de Entrada y Salida cuando se ingresa la clave correcta, el control active la alarma por compuerta e inicie el bloqueo. Para esto se sigue un procedimiento similar al realizado en la Prueba #1, pero durante la apertura del portón se recibe la señal de Entrada de nuevo. Se espera la transición de estados $a \Rightarrow b \Rightarrow e \Rightarrow g$ y la activación de la señal de apertura desde que se ingresa la clave correcta y hasta que se activa la entrada, punto en el cual se debe activar la señal de alarma por compuerta AlrmCom y la de Cerrar.

```
// Entra un carro
#10 Entrada = 1;

// Se ingresa el código correcto
#5 Clave = {4'd0, 4'd2, 4'd5, 4'd9};
#4 Enter = 1; #7 Enter = 0;

// Carro saliendo y entra otro.
#5 Salida = 1;
```

4. Instrucciones de utilización de la simulación

Para repetir las simulaciones utilizadas se incluye un archivo de Makefile que permite correr los comandos necesarios para realizarlas. Se debe tener instalado las herramientas de software Icarus Verilog (compilador de verilog), GTKWave (visualizador de ondas) y GNU-make o mingw Make (comandos para ejecutar un Makefile). Además, se deben tener los archivos incluidos del diseño (tester.v, testbench.v y Controlador.v) y el archivo Makefile en el mismo directorio que estos archivos. Se incluyen 3 comandos, notar que si se emplea la versión de make de mingw se debe escribir “mingw32-make” en vez de solo “make”.

Comandos incluidos en el Makefile:

- **make tarea** Corre la simulación al compilar el archivo y mostrar los resultados en GTKWave, está configurado para mostrar todas las ondas importantes según se muestra en los resultados.
- **make tareaSinConfig** Corre la simulación al compilar el archivo y mostrar los resultados en GTKWave sin la configuración del comando pasado (usar si el anterior genera error).
- **make clean** Elimina los archivos generados para limpiar el directorio (no elimina los de código ni el Makefile).

5. Ejemplos de los resultados

A través de las simulaciones se obtuvieron los resultados mostrados en los siguientes diagramas de tiempos. Todos estos provienen de una sola simulación, pero se separa en distintas imágenes, una para cada una de las pruebas realizadas. En general los resultados de cada una de las pruebas cumplieron las expectativas descritas en la Sección 3, incluyendo las transiciones de estados y las salidas generadas. Para los cambios de estados es útil consultar la Tabla 1. Debido a esto solo se resaltan los puntos clave de los resultados.

5.1. Resultados prueba #1

En la Figura 4 se muestra el ingreso de la clave correcta una vez se detectó el auto, se nota como se mantiene la señal para Abrir hasta que se da la señal de salida, punto en el cual se activa la señal Cerrar.

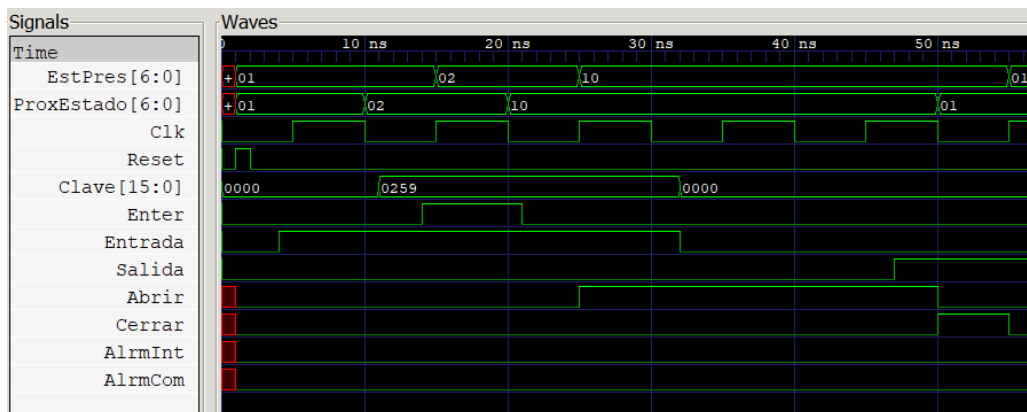


Figura 4: Diagrama de tiempo de prueba #1.

5.2. Resultados prueba #2

Algo a notar es que esta prueba se hizo posterior a la anterior pero sin aplicar la señal de reset, indicando que los intentos permitidos se reinician cuando se retorna al estado inicial. En el diagrama de la Figura 5 se muestra donde se ingresan los 3 intentos, donde el último corresponde a la clave correcta y se genera el comportamiento esperado cuando se ingresa dicha clave sin importar que se haya ingresado incorrectamente en los intentos anteriores.

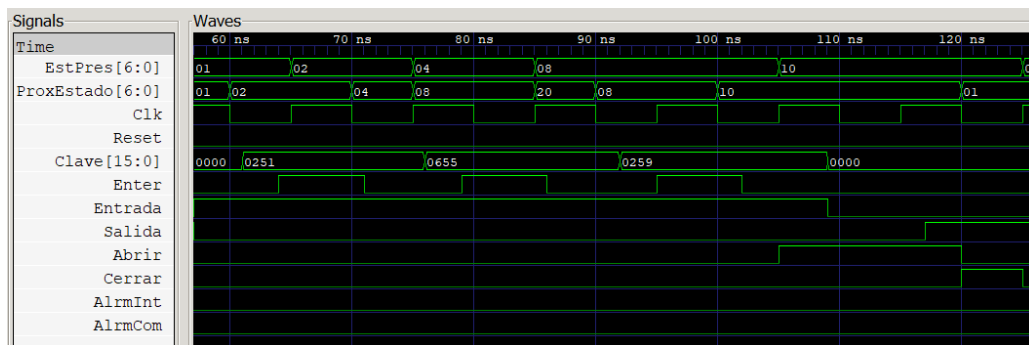


Figura 5: Diagrama de tiempo de prueba #2.

5.3. Resultados prueba #3

En la Figura 6 se muestra el primer comportamiento de bloqueo, donde se repite básicamente la misma secuencia que en la prueba anterior pero nunca se digita la clave correcta. Se nota el hecho que la señal de alarma por cantidad de intentos (AlrmInt) se activa al pasar al estado f (20). Además, estas señales y el estado no cambian sino hasta que se aplica la señal de reset.

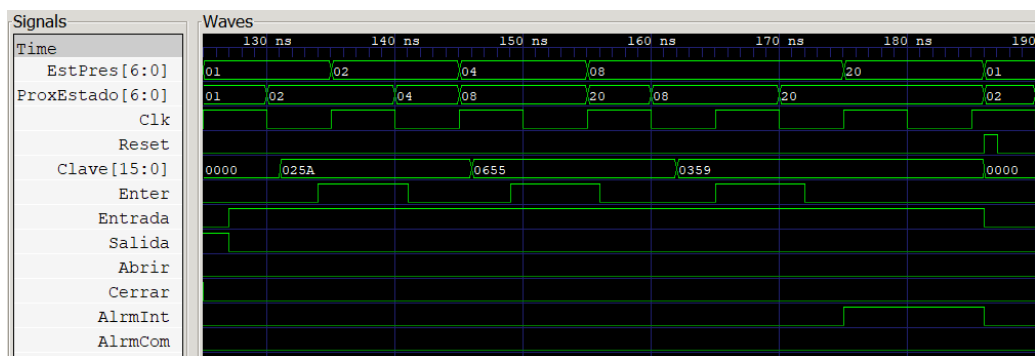


Figura 6: Diagrama de tiempo de prueba #3.

5.4. Resultados prueba #4

En la Figura 7 se muestran los resultados de la última prueba realizada y la otra condición de bloqueo. Se nota que se ingresa la clave correcta, pero durante el estado e (10) se activó Entrada antes de que se desactivara Salida, causando la transición al estado de emergencia g (donde se activan AlrmCom y Cierre). Se nota que como no se aplicó un reset el estado no cambia, cumpliendo el hecho que se debe mantener en estado de bloqueo.

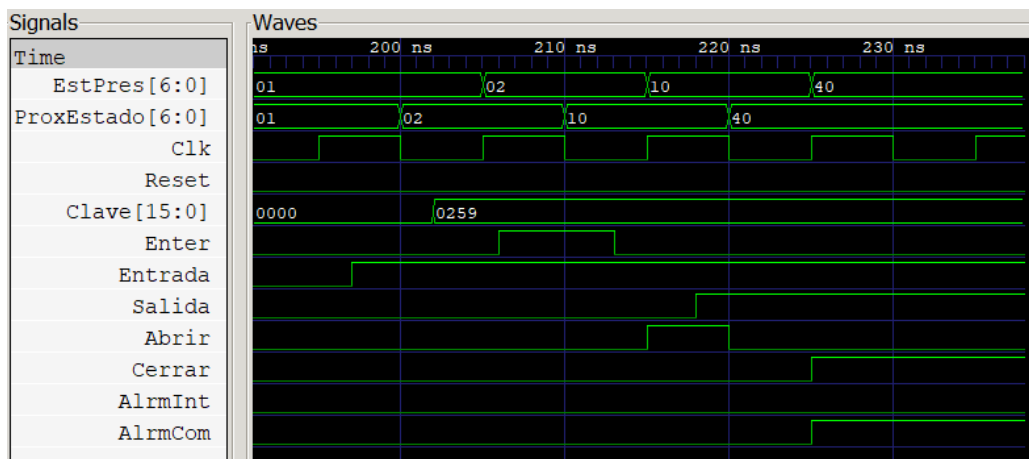


Figura 7: Diagrama de tiempo de prueba #4.

6. Conclusiones y recomendaciones

Durante el proceso de diseño se logró obtener una implementación en Verilog del controlador acorde a las especificaciones dadas y a través de las simulaciones/pruebas se pudo verificar que dicho diseño funciona correctamente. Aún así, se tienen ciertas limitaciones en el diseño, principalmente por la forma en la que interactúa con el exterior. Uno de estos radicó en la necesidad de añadir la señal de Enter, la cual no se indicó en las especificaciones, para reconciliar el ingreso obtuso de la clave. El otro en la ambigüedad de las señales de control, por ejemplo se desconoce si se debe mantener la señal Cerrar para completar su proceso ni su duración. Otro aspecto es el uso de contadores, el cual no se consideró sino hasta tener el diagrama ASM planteado y el proyecto iniciado. El uso de contadores para la cantidad de intentos podría reducir la cantidad de flip-flops usados (debido a que se darían menos estados) y haría un código conductual menos dependiente en el diagrama ASM. Para diseños futuros o en caso de continuar elaborando el controlador, se podrían analizar las diferencias entre el uso del contador y el diseño planteado, para encontrar cual es más apto o si resultan equivalentes.