

# Benchmarking de Estrategias de Control Basado en Eventos para Entornos Industriales con Recursos Compartidos

Cristhian Joel Apaza Flores

Facultad de Ingeniería, Universidad Mayor de San Andrés

Email: cjpaza11@umsa.bo

**Abstract**—El Control Basado en Eventos (CBE) ha demostrado ser una estrategia eficiente en la optimización del uso de recursos en sistemas de control. Sin embargo, la falta de un procedimiento estandarizado para evaluar y comparar diferentes estrategias de CBE dificulta su adopción en entornos industriales con recursos compartidos o limitados. Este artículo presenta un procedimiento de benchmarking para analizar estrategias de CBE en comparación con el control periódico. Se comparan estrategias como Send-on-Delta, Lyapunov y Retroalimentación de Estado. Los resultados muestran que las estrategias de CBE mejoran la eficiencia en el uso de recursos y ofrecen mayor adaptabilidad frente a eventos críticos en comparación con el control periódico.

**Index Terms**—Control Basado en Eventos, Benchmarking, Automatización Industrial, Recursos Compartidos

## I. INTRODUCCIÓN

En los sistemas de control industrial tradicionales, los controladores operan en intervalos de muestreo fijos, lo que conlleva un uso excesivo de recursos de procesamiento y comunicación. El CBE permite la activación del control solo cuando se detectan eventos relevantes en el proceso [1], [2]. Sin embargo, la falta de metodologías comparativas dificulta la selección y su implementación efectiva en entornos industriales con recursos compartidos.

Este artículo propone un procedimiento de benchmarking para evaluar estrategias de CBE, abordando criterios como eficiencia de activación, precisión y consumo de recursos. Además, se discuten los desafíos prácticos asociados con la implementación de estas estrategias en sistemas industriales reales.

## II. FUNDAMENTOS TEÓRICOS

### A. Control Basado en Eventos (CBE)

El CBE se basa en la ejecución del control cuando se cumplen ciertas condiciones predefinidas en el sistema [3]. Esto contrasta con el control periódico, donde las acciones se toman a intervalos regulares sin considerar el estado actual del sistema como se aprecia en la Figura 1.

### B. Estrategias de CBE Evaluadas

Se han identificado múltiples estrategias de CBE en la literatura [4]. Se seleccionaron las más representativas según dos criterios: desde el punto de vista académico [5], donde predominan enfoques basados en espacio de estados, y desde

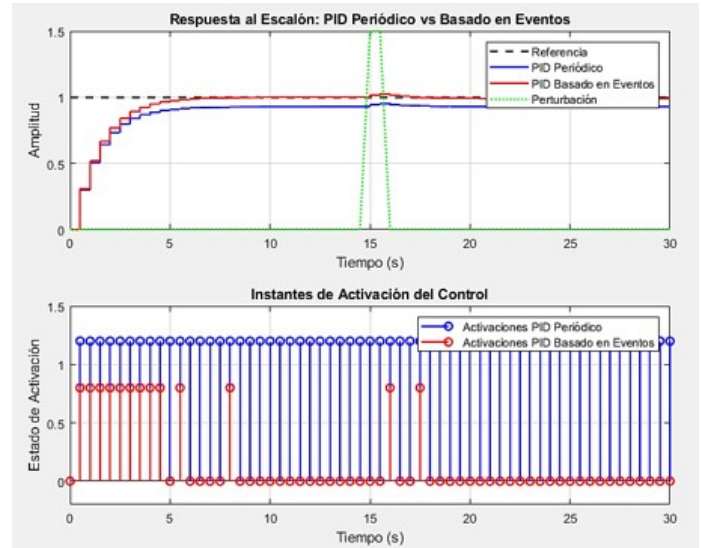


Fig. 1. Diferencia entre el control periódico y el basado en eventos, mostrando su respuesta temporal y los instantes de activación del control.

el punto de vista industrial, donde más del 90% de las aplicaciones utilizan controladores PID. Las estrategias seleccionadas son:

- **Send-on-Delta (SoD):** Este método activa el controlador solo cuando la variación de la señal de error supera un umbral predefinido  $\Delta$  [6]. Esta estrategia permite reducir la frecuencia de activación del controlador y optimizar el uso de recursos computacionales y de comunicación.
- **Sin Condición de Seguridad (WOS):** Esta estrategia es una variante del SoD en la que se omite la condición de seguridad, es decir, el controlador solo se activa cuando la señal de error excede un umbral determinado, sin considerar restricciones adicionales [7]. Su simplicidad la hace atractiva para aplicaciones en las que se busca minimizar activaciones innecesarias sin comprometer significativamente la estabilidad del sistema.
- **Retroalimentación de Estado:** En este enfoque, la activación del controlador depende de las mediciones del estado del sistema en lugar de únicamente la señal de error [8]. A través del uso de observadores de estado o estimaciones dinámicas, el sistema decide cuándo inter-

venir, optimizando la respuesta y evitando activaciones innecesarias. Esta estrategia se aplica en sistemas con restricciones de comunicación y procesamiento, como redes de control distribuido [3].

- **Control basado en Lyapunov:** Utiliza funciones de Lyapunov para definir condiciones de disparo óptimas que garantizan estabilidad y eficiencia [9]. Este método permite que el sistema opere en un equilibrio dinámico, reduciendo la frecuencia de activación sin comprometer la estabilidad del controlador. Se ha aplicado en sistemas con alta incertidumbre y variabilidad, como el control de robots y sistemas no lineales [4].

### III. METODOLOGÍA

Para realizar el benchmarking, se definieron métricas de evaluación como el consumo de CPU, el número de activaciones del controlador y el tiempo de respuesta. Se estableció el siguiente procedimiento basado en iteraciones y validaciones experimentales:

#### A. Procedimiento de Benchmarking

- 1) **Revisión bibliográfica:** Se analizan los antecedentes y trabajos previos sobre CBE, identificando estrategias y métricas relevantes para la evaluación.
- 2) **Selección de estrategias de CBE y escenario de trabajo:** Se eligen estrategias representativas de CBE y se define el entorno en el que se realizarán las pruebas.
- 3) **Extracción o generación de algoritmos:** Se obtienen o desarrollan algoritmos para la implementación de las estrategias seleccionadas.
- 4) **Codiseño de parámetros de control y condición de evento:** Se optimizan los parámetros para garantizar el mejor desempeño de cada estrategia.
- 5) **Ejecución de simulaciones controladas:** Se realizan pruebas en simuladores para observar el comportamiento de las estrategias en un entorno virtual.
- 6) **Implementación en sistemas físicos:** Se aplican las estrategias en un sistema físico real para validar los resultados obtenidos en simulación.
- 7) **Recopilación y análisis de datos:** Se registran los resultados obtenidos en cada prueba y se analizan con base en los criterios definidos.
- 8) **Evaluación y comparación:** Se comparan las estrategias en términos de eficiencia, consumo de recursos y calidad del control.
- 9) **Conclusiones y recomendaciones:** Se sintetizan los hallazgos obtenidos y se formulan recomendaciones para futuras investigaciones.

Se implementó este procedimiento a las estrategias seleccionadas sobre sistemas representativos como un sistema de nivel y un proceso de manufactura validamos el procedimiento.

1) *Identificación de los Sistemas:* Los modelos se obtuvieron experimentalmente mediante análisis de respuesta al escalón:

- **Sistema de Nivel:** Relación entre frecuencia del variador (Hz) y nivel (cm):

$$G(s) = \frac{0.923}{260s + 1} \quad [\text{cm/Hz}] \quad (1)$$

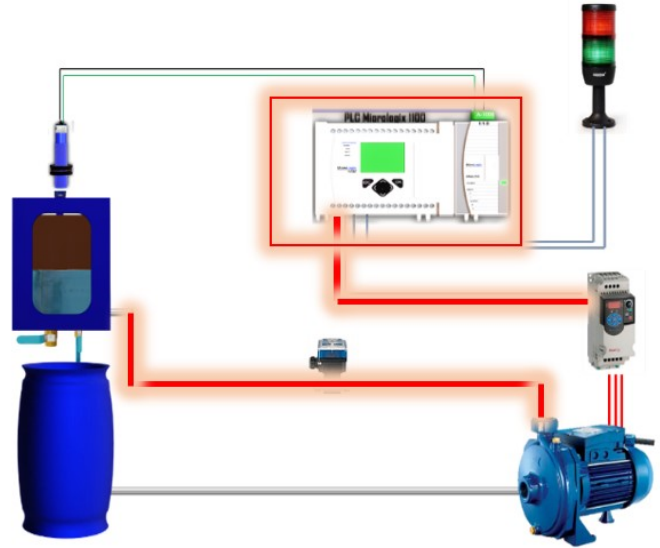


Fig. 2. Esquema del sistema de nivel de líquido.

- **Banda Transportadora:**

$$G(s) = \frac{48.8}{s^3 + 5.163s^2 + 15.58s} \quad [\text{cm/V}] \quad (2)$$

#### B. Unidad de Selección

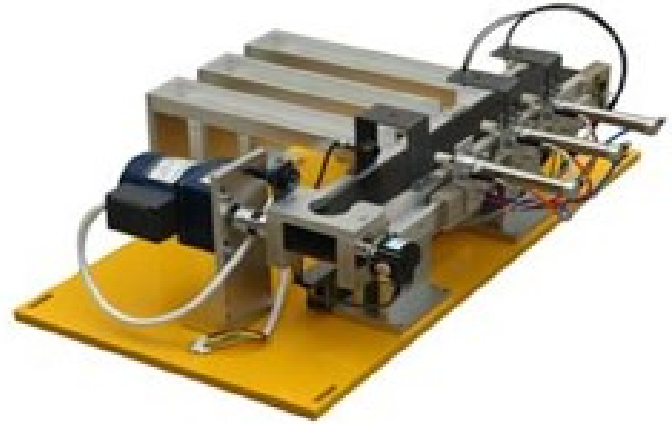


Fig. 3. Esquema de la unidad de selección.

### IV. APLICACION DEL PROCEDIMIENTO

Se asume la existencia previa de un controlador periódico en el sistema de prueba y no se aborda el despliegue del controlador, sino solo la comparación de estrategias. Según el procedimiento establecido, los primeros puntos como la revisión, el análisis y selección son comunes en este tipo

de estudios. Los puntos como la generación del algoritmo representa un reto, puesto que en base a un entendimiento de la estrategia se abstraen el algoritmo de la estrategia.

El tema de codiseño se realiza mediante métodos computacionales de optimización encontrando los parámetros de evento que minimicen el error y la activación de evento, es decir los parámetros de evento seleccionados son los que hacen mínimo el error y menor número de activaciones.

El algoritmo de optimización implementado en MATLAB realiza una búsqueda exhaustiva (grid search) sobre los rangos de  $\Delta$  (0.1 a 1.0 en pasos de 0.1) y  $h_{\max}$  (0.5 a 5.0 en pasos de 0.5). Este enfoque garantiza que los parámetros seleccionados representen el mejor compromiso entre reducción de activaciones y precisión del control.

Este enfoque garantiza que los parámetros seleccionados ( $\Delta_{\text{opt}}$ ,  $h_{\max, \text{opt}}$ ) representen el mejor compromiso entre reducción de activaciones y precisión del control.

La búsqueda en cuadrícula se eligió por su transparencia y facilidad de implementación, aunque requiere mayor costo computacional que métodos basados en gradientes.

Con el algoritmo extraído y los parámetros de control y evento seleccionados se procede con la simulación, implementando los algoritmos en un software de simulación, en la figura 4 podemos ver la simulación de SoD. Uno de los

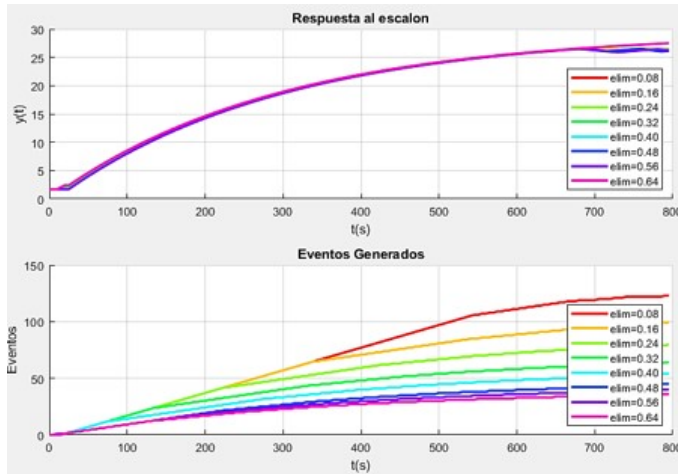


Fig. 4. La figura muestra la respuesta temporal y el número de eventos acumulados en el tiempo para el caso de SoD manteniendo el parámetro de evento  $h_{\max}$  constante y haciendo variar el parámetro  $elim$ .

aspectos más relevantes es la implementación del algoritmo en el controlador. Se sabe que la mayoría de las unidades de producción usan PLCs los cuales usan lenguaje Ladder para su programación, en base a los bloques aritméticos, lógicos, etc. se implementan los algoritmos de CBE como se aprecia en la figura 5 tanto la condición de evento como el algoritmo de control se pueden implementar con los bloques disponibles del lenguaje Ladder. Una vez implementado el algoritmo en el controlador, se realizan las pruebas, la adquisición y el procesamiento de los datos para obtener las respuestas temporales como se aprecia en la figura 6

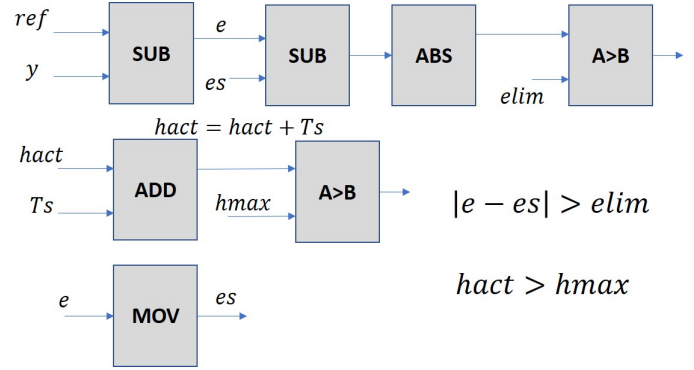


Fig. 5. Implementación de la condición de evento para la estrategia SoD utilizando bloques en lenguaje Ladder.

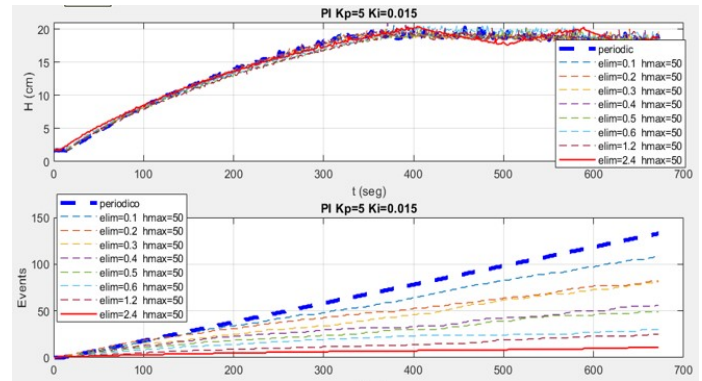


Fig. 6. La figura muestra la respuesta temporal de SoD manteniendo el parámetro  $h_{\max}$  constante y haciendo variar  $elim$  de la misma forma que en la simulación.

Tanto la figura 4 como la figura 6 nos muestran la relación entre el parámetro  $elim$  y el número de eventos activados, notando que a medida que el parámetro aumenta el número de eventos disminuye, esto es evidente pues el parámetro  $elim$  define la sensibilidad a la ocurrencia de un evento en este caso para la estrategia SoD.

En base al tipo de bloque de instrucción y el número de estos utilizados se hace el cálculo de cuánto tiempo le lleva el controlador la ejecución del programa implementado por cada periodo de muestreo para el caso periódico y por cada instante de activación y no activación para el caso por eventos, para el caso de SoD y usando el PLC S7-200 de Siemens se aprecia en la Tabla I.

Los cálculos en el PLC S7-200 se realizaron con aritmética de punto fijo (16 bits), introduciendo un error máximo de  $\pm 0.4\%$  en las condiciones de disparo.

Se repite el mismo cálculo para las demás estrategias, el resumen se aprecia en la Tabla II. Para poder realizar la comparación también se hace el cálculo de tiempo de ejecución para el caso periódico como se aprecia en la Tabla II.

En base a las tablas II III y aplicando diferencias porcentuales se llega a la tabla final de comparación Tabla IV.

TABLE I  
TIEMPOS DE EJECUCIÓN DE INSTRUCCIONES

Instrucción	Tiempo máximo (μs)	Nro. de instrucciones
ADD	13.34	1
SUB	13.46	2
ABS	9.71	1
AND	13.06	1
LES	9.09	2
IMP	1.15	1
RET	1.68	1
<b>Total</b>	<b>84.04</b>	

TABLE II  
NUMERO DE ACTIVACIONES Y TIEMPOS DE EJECUCION DE  
ESTRATEGIAS DE CBE

Estrategia	# Act.	# Per. sin Act.	Tiempo por Act. (μs)	Tiempo sin Act. (μs)
SOD	27	93	591.38	84.04
WOS	24	96	773.95	84.04
Lyapunov	31	89	537.7	76.86
State-feedback	100	20	664	627.5

## V. RESULTADOS Y DISCUSIÓN

En esta sección se presentan los resultados obtenidos de la evaluación y comparación de las estrategias de control basado en eventos (CBE) frente al control periódico (CP). Los resultados obtenidos se resumen en la Tabla IV, donde se comparan las estrategias de CBE en términos de reducción de activaciones y eficiencia en el uso de recursos.

Los resultados obtenidos, sintetizados en la Tabla IV, indican que las estrategias CBE logran una reducción significativa en el número de activaciones del actuador en comparación con el CP, sin comprometer el rendimiento del sistema. En particular, la estrategia Sin Condición de Seguridad mostró una mejora del 89% en ejecución de instrucciones en comparación con CP, evidenciando su ventaja en escenarios donde la optimización del consumo es prioritaria.

En términos de respuesta dinámica, las estrategias CBE lograron mantener tiempos de establecimiento y sobrepicos similares a los del CP, como se muestra en la figura 6, validando su viabilidad en aplicaciones industriales que requieren estabilidad y precisión. La Figura 1 ilustra la comparación de los instantes de activación de control entre el CP y el CBE mostrando que el CBE solo se activa ante la ocurrencia de un evento pero con el mismo nivel de funcionamiento que el control periódico.

Se han identificado diferencias entre simulación e implementación que inicialmente dificultaron la validación de resultados. Sin embargo, la incorporación de modelos de saturación y zona muerta permitió que los resultados físicos y simulados converjan de manera más efectiva.

Los resultados confirman que las estrategias de CBE pueden reducir significativamente el consumo de recursos sin comprometer la estabilidad del sistema. Como parte del procedimiento de benchmarking desarrollado, se realizó una comparación sistemática de las estrategias CBE mediante la evaluación de métricas clave como el tiempo sin activación y el incremento en tiempo con activación. La Tabla IV muestra un resumen comparativo entre distintas estrategias evaluadas, indicando el

TABLE III  
TIEMPO DE ACTIVACIÓN ESTRATEGIAS DE CONTROL PERIODICAS

Estrategia	# Activaciones	Tiempo por activación (μs)
PID	120	427.9
Realimentación de estados	120	426.1
Lyapunov	120	396.2

TABLE IV  
COMPARACIÓN DE ESTRATEGIAS DE CONTROL BASADO EN EVENTOS

Estrategia	Mejora en Tiempo sin Activación (%)	Incremento con Activación (%)
SoD	86.30	38.47
WOS	89.14	81.21
Lyapunov	85.71	35.71
State-feedback	5.50	55.83

porcentaje de mejora en tiempo sin activación y el incremento en tiempo con activación comprobando de esta manera la utilidad del procedimiento desarrollado.

Un desafío encontrado en este estudio fue la implementación de estrategias como Lyapunov, donde el algoritmo no estaba desarrollado previamente y fue necesario generarlo desde su base teórica. Además, la adaptación de los algoritmos en los PLCs presentó limitaciones debido a la falta de funciones matemáticas como la exponencial, lo que requirió aproximaciones mediante series de Taylor para su implementación.

## VI. CONCLUSIONES

La aplicación del procedimiento de benchmarking desarrollado demuestra que las estrategias de CBE representan una alternativa viable y eficiente frente al control periódico en entornos industriales con recursos compartidos. La reducción en la cantidad de activaciones del actuador permite extender la vida útil de los dispositivos y mejorar la eficiencia al reducir los cálculos computacionales sin afectar significativamente el rendimiento del sistema.

Se concluye que la estrategia SoD es particularmente adecuada para aplicaciones donde la optimización del consumo de recursos computacionales es prioritaria, manteniendo un desempeño comparable al CP. Sin embargo, su implementación requiere un ajuste adecuado de los parámetros de control y de la condición de evento para garantizar su efectividad en diferentes escenarios operativos.

Estudios en actuadores indican que reducir las activaciones en un 40-50% puede aumentar su vida útil en 2-3 años en operación continua. Nuestros resultados con SoD y WOS (45-52% de reducción) sugieren beneficios significativos en mantenimiento preventivo.

A pesar de los resultados positivos, este estudio presenta algunas limitaciones.

Se determinó que el rendimiento de los CBE depende no solo del algoritmo de control, sino también de los recursos computacionales disponibles en la plataforma de implementación. En particular, la ausencia de ciertas funciones matemáticas en los PLCs exigió el desarrollo de soluciones alternativas, como la aproximación de la función exponencial mediante series de Taylor. Además, la discrepancia inicial

entre las simulaciones y la implementación física resaltó la importancia de modelar correctamente fenómenos como la zona muerta y la saturación de actuadores. La corrección de estos aspectos en la simulación permitió obtener resultados más representativos del comportamiento real del sistema, validando así la efectividad del procedimiento de benchmarking.

La implementación de estrategias de CBE en entornos industriales puede requerir ajustes adicionales para adaptarse a distintos procesos. Además, el análisis se realizó en escenarios específicos y podría ser ampliado a sistemas más complejos.

La aplicación del CBE en la industria tiene un gran potencial para optimizar el consumo de recursos computacionales y extender la vida útil de los actuadores. Este procedimiento de benchmarking proporciona una metodología para evaluar la viabilidad de estas estrategias en distintos entornos industriales.

Para futuras investigaciones, se recomienda ampliar la validación a sistemas industriales adicionales, incorporando nuevas estrategias de CBE con enfoques híbridos y técnicas de aprendizaje automático para mejorar la adaptabilidad de los controladores. Asimismo, se sugiere explorar algoritmos de optimización automática para el codiseño de los parámetros de control en tiempo real, con el objetivo de maximizar la eficiencia de los sistemas bajo control basado en eventos.

## REFERENCES

- [1] K.-E. Årzén, "A simple event-based PID controller," in *IFAC Proceedings Volumes*, 1999.
- [2] K. J. Åström and B. Bernhardsson, "Comparison of periodic and event based sampling for first-order stochastic systems," in *IFAC*, 2008.
- [3] J. Lunze and D. Lehmann, "A state-feedback approach to event-based control," *Automatica*, 2010.
- [4] W. Heemels, K. H. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," *IEEE Control Systems Magazine*, 2012.
- [5] E. Aranda-Escolastico et al., "Event-Based Control: A Bibliometric Analysis of Twenty Years of Research," *IEEE Access*, vol. 8, pp. 47188–47208, 2020, doi: 10.1109/ACCESS.2020.2978174.
- [6] M. Miśkiewicz, "Send-on-Delta Concept: An Event-Based Data Reporting Strategy," *Sensors*, vol. 6, no. 1, pp. 49–63, 2006, doi: 10.3390/s6010049.
- [7] S. Durand and A. Sename, "Event-triggered Control Without Security Condition for Networked Systems," *IFAC-PapersOnLine*, vol. 51, no. 3, pp. 237–242, 2018, doi: 10.1016/j.ifacol.2018.06.024.
- [8] D. Lehmann and J. Lunze, "Event-Based Control Using State-Feedback Estimates," *IEEE Transactions on Automatic Control*, vol. 56, no. 2, pp. 370–375, 2011, doi: 10.1109/TAC.2010.2089532.
- [9] P. Tabuada, "Event-triggered real-time scheduling of stabilizing control tasks," *IEEE Transactions on Automatic Control*, vol. 52, no. 9, pp. 1680–1685, 2007, doi: 10.1109/TAC.2007.904277.
- [10] X.-M. Zhang et al., "Sampled-data control systems with non-uniform sampling: A survey of methods and trends," *Annual Reviews in Control*, vol. 55, pp. 70–91, Jan. 2023, doi: 10.1016/j.arcontrol.2023.03.004.

## APPENDIX

```
function calcularValoresOptimos(obj, controlador)
% Recorre el rango de elim y hmax para encontrar los valores óptimos
for elim = obj.Elim_range
    for hmax = obj.Hmax_range
        controlador.elim = elim;
        controlador.hmax = hmax;
        controlador.simular(); % Simula con los nuevos valores

        Nro_eventos = sum(controlador.count);

        % Almacenar los valores para graficar
        obj.elim_values = [obj.elim_values, elim];
```

```
        obj.hmax_values = [obj.hmax_values, hmax];
        obj.events_values = [obj.events_values, Nro_eventos];

% Actualizar los valores óptimos si hay menos eventos
if Nro_eventos < obj.min_events
    obj.min_events = Nro_eventos;
    obj.best_elim = elim;
    obj.best_hmax = hmax;
end
end
end
end
```