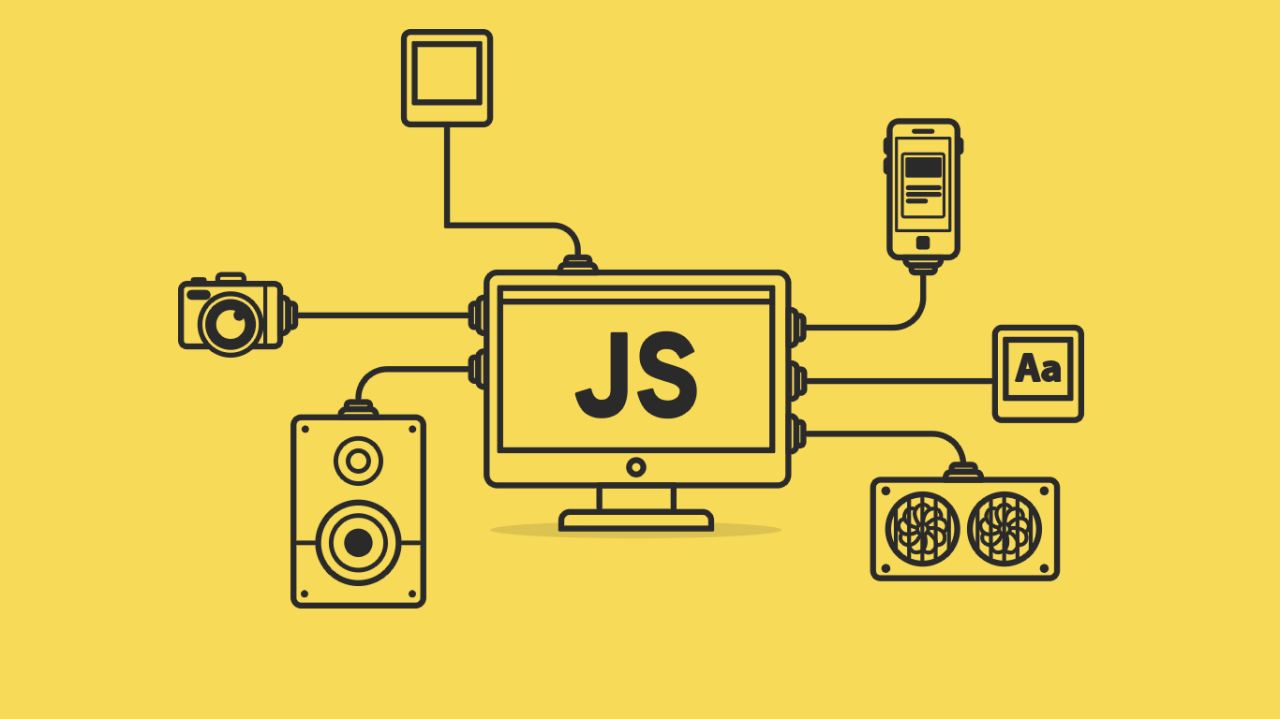
**Documentación del Código JAVASCRIPT**



Ismael Mariscal Santos

# Proyecto de administración de sistemas informáticos en red

Índice

[Gráfico de Donut para CPU 4](#_Toc167312811)

[Gráfico de Donut para RAM 6](#_Toc167312812)

[Gráfico de Donut para Disco 8](#_Toc167312813)

[Gráfico de Líneas para Lectura de Disco 10](#_Toc167312814)

[Gráfico de Líneas para Escritura de Disco 12](#_Toc167312815)

[Gráfico de Líneas para Uso de la Red 14](#_Toc167312816)

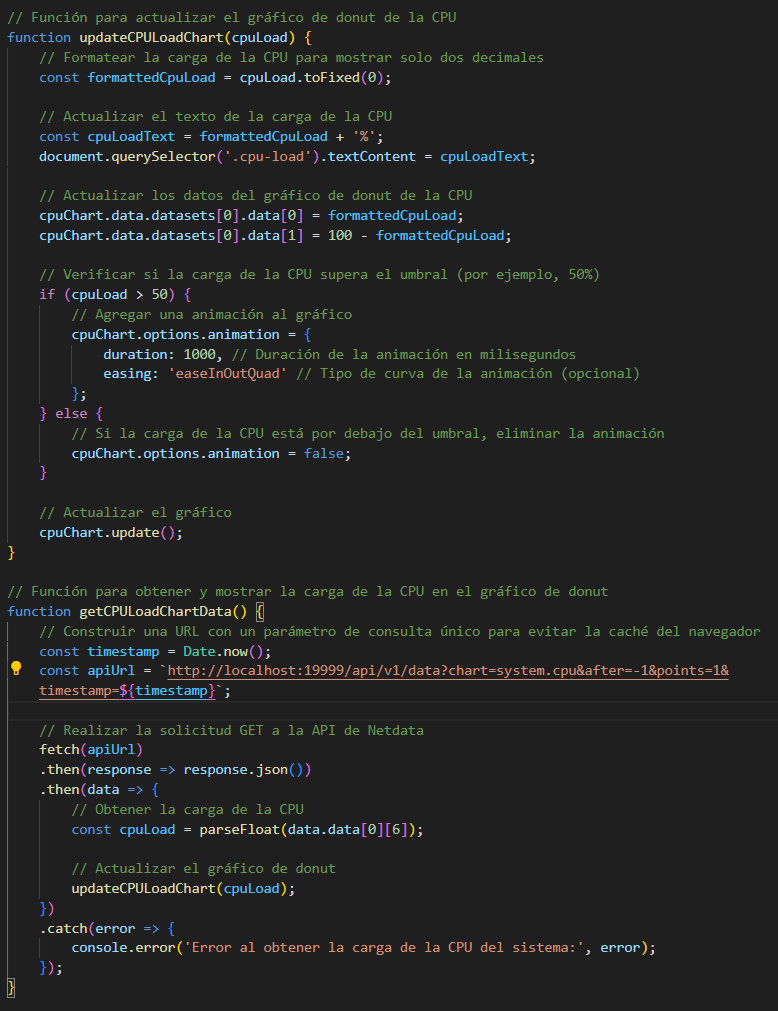
[Alertas y Monitoreo de Objetivos 16](#_Toc167312817)

[Monitoreo de Objetivos 18](#_Toc167312818)

[Interfaz de Usuario 20](#_Toc167312819)

[Utilidades 21](#_Toc167312820)

# Gráfico de Donut para CPU



* **Descripción:** Representa la carga de la CPU en forma de un gráfico de donut.
* **Funciones:**
  + updateCPULoadChart(cpuLoad): Actualiza el gráfico de donut con la carga de la CPU.

**Parámetros:**

* cpuLoad: Carga actual de la CPU en porcentaje.

**Funcionalidad:**

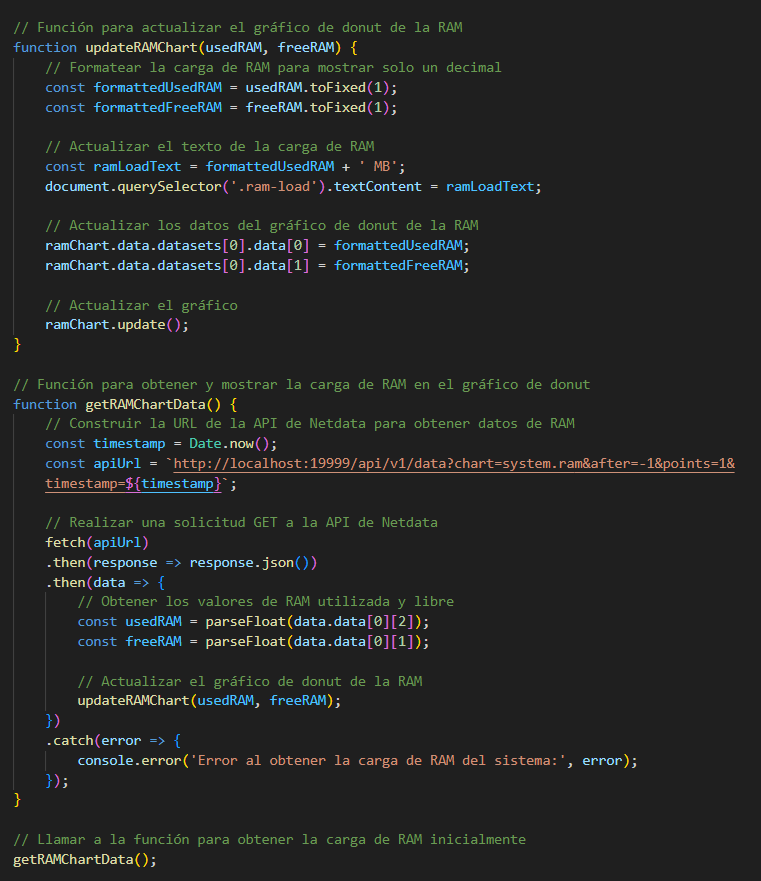
* Formatea la carga de la CPU para mostrar solo dos decimales.
* Actualiza el texto de la carga de la CPU en la interfaz.
* Actualiza los datos del gráfico de donut con la carga de la CPU.
* Agrega una animación al gráfico si la carga de la CPU supera un umbral definido.
* Actualiza el gráfico.
  + getCPULoadChartData(): Obtiene y muestra datos de carga de CPU en el gráfico de donut.

**Descripción:** Obtiene y muestra datos de carga de CPU en el gráfico de donut.

**Funcionalidad:**

* Construye una URL para obtener datos de carga de CPU desde una API.
* Realiza una solicitud GET a la API de Netdata.
* Procesa la respuesta para obtener la carga de la CPU.
* Llama a updateCPULoadChart() para actualizar el gráfico con los nuevos datos.
* Maneja errores en caso de fallo en la solicitud o procesamiento de datos.

# Gráfico de Donut para RAM



* **Descripción:** Representa el uso de la memoria RAM en forma de un gráfico de donut.
* **Funciones:**
  + updateRAMChart(usedRAM, freeRAM): Actualiza el gráfico de donut con el uso de RAM.

**Parámetros:**

* usedRAM: Cantidad de RAM utilizada.
* freeRAM: Cantidad de RAM libre.

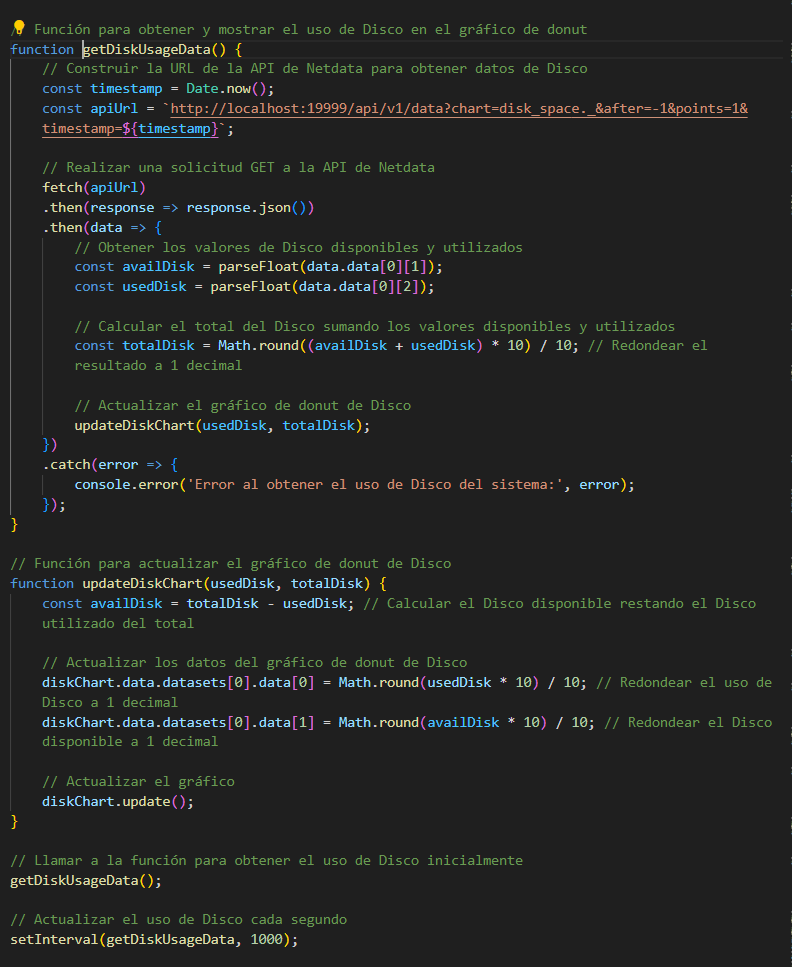
**Funcionalidad:**

* Formatea el uso de RAM para mostrar solo un decimal.
* Actualiza el texto de uso de RAM en la interfaz.
* Actualiza los datos del gráfico de donut con el uso de RAM.
* Actualiza el gráfico.
  + getRAMChartData(): Obtiene y muestra datos de uso de RAM en el gráfico de donut.

**Funcionalidad:**

* Construye una URL para obtener datos de uso de RAM desde una API.
* Realiza una solicitud GET a la API de Netdata.
* Procesa la respuesta para obtener el uso de RAM.
* Llama a updateRAMChart() para actualizar el gráfico con los nuevos datos.
* Maneja errores en caso de fallo en la solicitud o procesamiento de datos.

# Gráfico de Donut para Disco



* **Descripción:** Representa el uso del disco en forma de un gráfico de donut.
* **Funciones:**
  + updateDiskChart(usedDisk, totalDisk): Actualiza el gráfico de donut con el uso de disco.

**Parámetros:**

* usedDisk: Espacio de disco utilizado.
* totalDisk: Espacio total de disco.

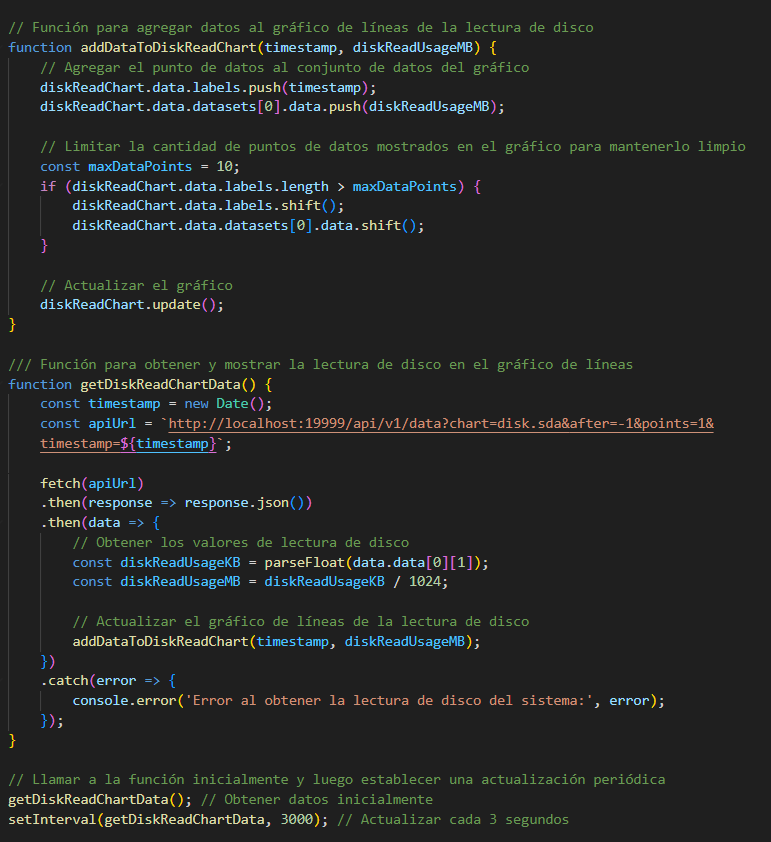
**Funcionalidad:**

* Calcula el espacio de disco disponible.
* Actualiza los datos del gráfico de donut con el uso de disco.
* Actualiza el gráfico.
  + getDiskUsageData(): Obtiene y muestra datos de uso de disco en el gráfico de donut.

**Funcionalidad:**

* Construye una URL para obtener datos de uso de disco desde una API.
* Realiza una solicitud GET a la API de Netdata.
* Procesa la respuesta para obtener el uso de disco.
* Llama a updateDiskChart() para actualizar el gráfico con los nuevos datos.
* Maneja errores en caso de fallo en la solicitud o procesamiento de datos.

# Gráfico de Líneas para Lectura de Disco



* **Descripción:** Representa la velocidad de lectura del disco en forma de un gráfico de líneas.
* **Funciones:**
  + addDataToDiskReadChart(timestamp, diskReadUsageMB): Agrega datos al gráfico de líneas de la lectura de disco.

**Parámetros:**

* timestamp: Marca de tiempo de la lectura.
* diskReadUsageMB: Velocidad de lectura de disco en megabytes por segundo.

**Funcionalidad:**

* Agrega el punto de datos al conjunto de datos del gráfico.
* Limita la cantidad de puntos de datos mostrados en el gráfico.
* Actualiza el gráfico.
  + getDiskReadChartData(): Obtiene y muestra datos de lectura de disco en el gráfico de líneas.

**Funcionalidad:**

* Construye una URL para obtener datos de lectura de disco desde una API.
* Realiza una solicitud GET a la API de Netdata.
* Procesa la respuesta para obtener la velocidad de lectura de disco.
* Llama a addDataToDiskReadChart() para agregar datos al gráfico.
* Maneja errores en caso de fallo en la solicitud o procesamiento de datos.

# Gráfico de Líneas para Escritura de Disco



* **Descripción:** Representa la velocidad de escritura del disco en forma de un gráfico de líneas.
* **Funciones:**
  + addDataToDiskWriteChart(timestamp, diskWriteUsageMB): Agrega datos al gráfico de líneas de la escritura de disco.

**Parámetros:**

* timestamp: Marca de tiempo de la escritura.
* diskWriteUsageMB: Velocidad de escritura de disco en megabytes por segundo.

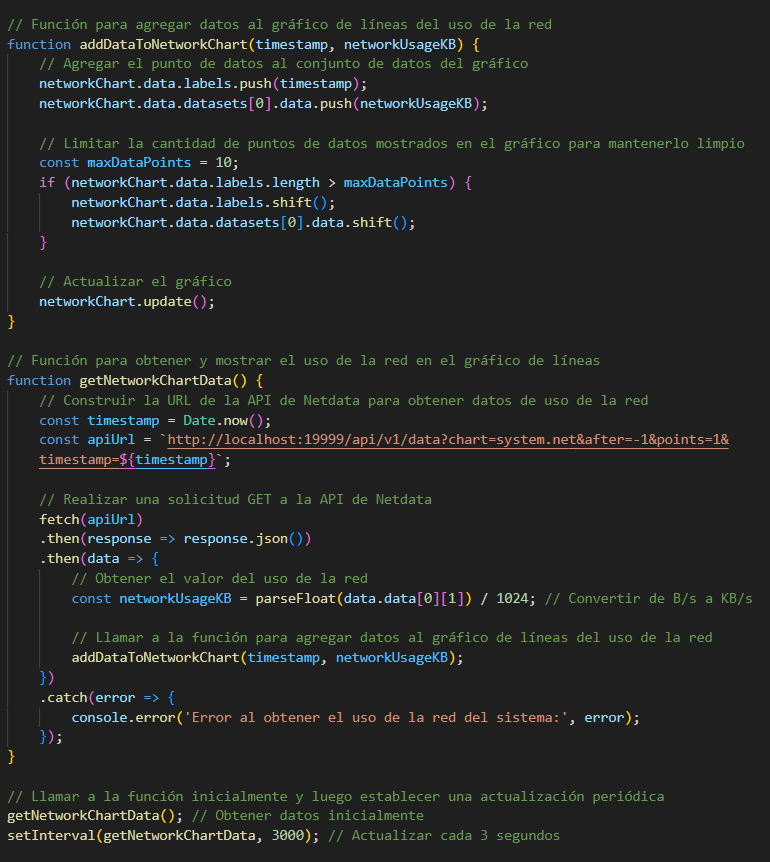
**Funcionalidad:**

* Agrega el punto de datos al conjunto de datos del gráfico.
* Limita la cantidad de puntos de datos mostrados en el gráfico.
* Actualiza el gráfico.
  + getDiskWriteChartData(): Obtiene y muestra datos de escritura de disco en el gráfico de líneas.

**Funcionalidad:**

* Construye una URL para obtener datos de escritura de disco desde una API.
* Realiza una solicitud GET a la API de Netdata.
* Procesa la respuesta para obtener la velocidad de escritura de disco.
* Llama a addDataToDiskWriteChart() para agregar datos al gráfico.
* Maneja errores en caso de fallo en la solicitud o procesamiento de datos.

# Gráfico de Líneas para Uso de la Red



* **Descripción:** Representa el uso de la red en forma de un gráfico de líneas.
* **Funciones:**
  + addDataToNetworkChart(timestamp, networkUsageKB): Agrega datos al gráfico de líneas del uso de la red.

**Parámetros:**

* timestamp: Marca de tiempo del uso de la red.
* networkUsageKB: Uso de la red en kilobytes por segundo.

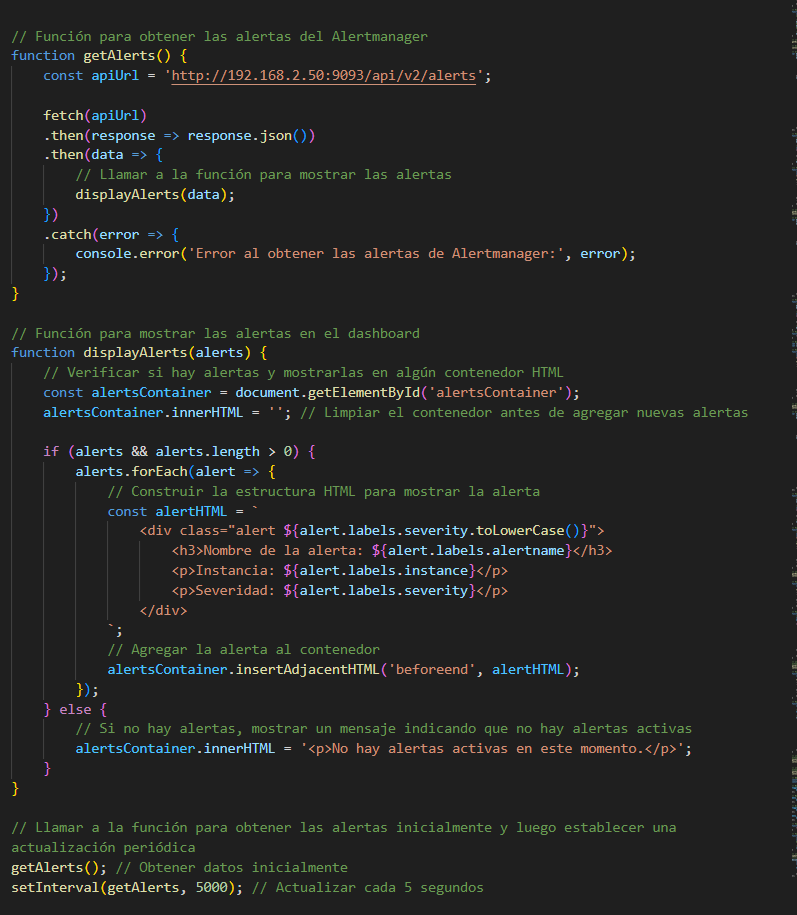
**Funcionalidad:**

* Agrega el punto de datos al conjunto de datos del gráfico.
* Limita la cantidad de puntos de datos mostrados en el gráfico.
* Actualiza el gráfico.
  + getNetworkChartData(): Obtiene y muestra datos de uso de la red en el gráfico de líneas.

**Funcionalidad:**

* Construye una URL para obtener datos de uso de la red desde una API.
* Realiza una solicitud GET a la API de Netdata.
* Procesa la respuesta para obtener el uso de la red.
* Llama a addDataToNetworkChart() para agregar datos al gráfico.
* Maneja errores en caso de fallo en la solicitud o procesamiento de datos.

# Alertas y Monitoreo de Objetivos

****

**Alertas**

* + **Descripción:** Obtiene y muestra alertas del Alertmanager.
  + **Funciones:**
    - getAlerts(): Obtiene alertas del Alertmanager.

**Funcionalidad:**

* Realiza una solicitud GET a la API del Alertmanager para obtener las alertas.
* Procesa la respuesta y llama a displayAlerts() para mostrar las alertas.
* Maneja errores en caso de fallo en la solicitud o procesamiento de datos.
  + - displayAlerts(alerts): Muestra las alertas en el dashboard.

# Monitoreo de Objetivos

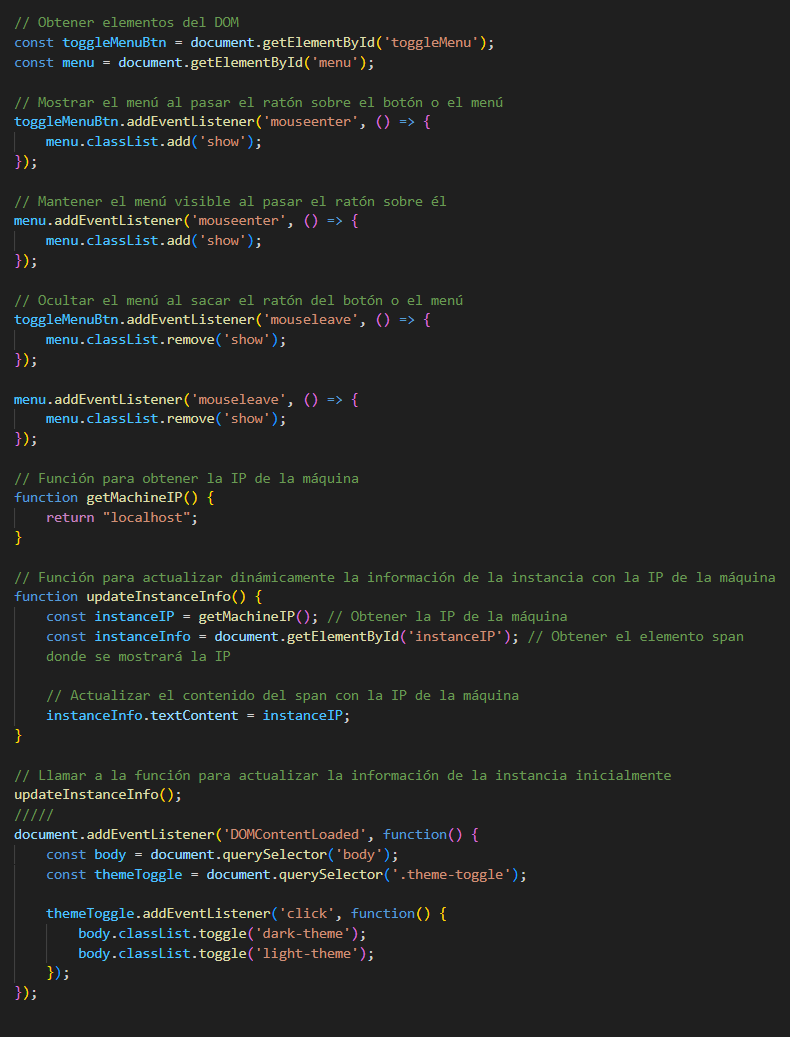


* **Descripción:** Obtiene y muestra el estado de los objetivos de monitoreo.
* **Funciones:**
  + getTargets(): Obtiene el estado de los objetivos de monitoreo.

**Funcionalidad:**

* Realiza una solicitud GET a la API Prometheus para obtener los targets.
* Procesa la respuesta y llama a displayTargets() para mostrar los targets y su estado.
* Maneja errores en caso de fallo en la solicitud o procesamiento de datos.
  + displayTargets(targets): Muestra los objetivos y su estado en el dashboard.

# Interfaz de Usuario



* + **Descripción:** Gestiona la interfaz de usuario y la interacción con el usuario.
  + **Funciones:**
    - Gestiona eventos de clic para abrir enlaces.
    - Actualiza la información de la instancia con la dirección IP.
    - Gestiona el cambio de tema entre claro y oscuro.

# Utilidades

* + **Descripción:** Proporciona funciones de utilidad reutilizables en el código.
  + **Funciones:**
    - eliminarSignoNegativo(valor): Elimina el signo negativo de un valor.

**Parámetros:**

* valor: Valor numérico que puede ser positivo o negativo.

**Funcionalidad:**

* Devuelve el valor absoluto del parámetro valor.

