

# System Resource Monitoring Dashboard

How to monitor system resources and performance



# PROYECTO INTEGRADO

## Dashboard de Monitoreo de Recursos del Sistema

**Autor:** [Ismael Mariscal Santos]

**Tutor:** [[Castillo García, Juan Pedro](#)]

**I.E.S. Francisco Romero Vargas (Jerez de la Frontera)**

**Administración de Sistemas Informáticos en Red**

**Curso: 2023/2024**

---

### Tabla de contenido

- Introducción
  - Finalidad
  - Objetivos
  - Medios necesarios
  - Planificación
  - Realización del Proyecto
    - Trabajos realizados
    - Problemas encontrados
    - Modificaciones sobre el proyecto planteado inicialmente
    - Posibles mejoras al proyecto
  - Medios Necesarios
  - Bibliografía
- 

### Introducción

#### Introducción

El presente proyecto tiene como objetivo crear un dashboard para monitorear en tiempo real los recursos del sistema, tales como la CPU, la RAM y el disco duro. Utilizando tecnologías web, este sistema permitirá visualizar gráficamente el uso de estos recursos y proporcionar alertas cuando ciertos umbrales sean superados.

#### Finalidad

La finalidad de este proyecto es facilitar el monitoreo continuo de los recursos del sistema, permitiendo a los administradores de sistemas identificar rápidamente cualquier problema de rendimiento. Al tener una visualización en tiempo real, se pueden tomar

decisiones informadas para optimizar el uso de los recursos y mantener el sistema funcionando eficientemente.

## **Objetivos**

Los objetivos de este proyecto son:

- Desarrollar un dashboard interactivo que muestre el uso de la CPU, RAM y disco duro.
- Implementar gráficos de tipo "doughnut" y de líneas utilizando Chart.js para la visualización de datos.
- Integrar el dashboard con una API (Netdata) para obtener datos en tiempo real de los recursos del sistema.
- Proporcionar alertas visuales y de texto cuando los recursos superen ciertos umbrales.
- Implementar un sistema de cambio de tema (oscuro y claro) para mejorar la usabilidad del dashboard.

## **Medios necesarios**

Para la realización de este proyecto se necesita lo siguiente:

### **Hardware:**

- Un ordenador personal de configuración media.
- Conexión a Internet para acceder a la API de Netdata.

### **Software:**

- Editor de código (Visual Studio Code).
  - Biblioteca Chart.js para la creación de gráficos.
  - API de Netdata para la obtención de datos de los recursos del sistema.
  - Lenguajes: HTML, CSS, y JavaScript.
- 

## **Planificación**

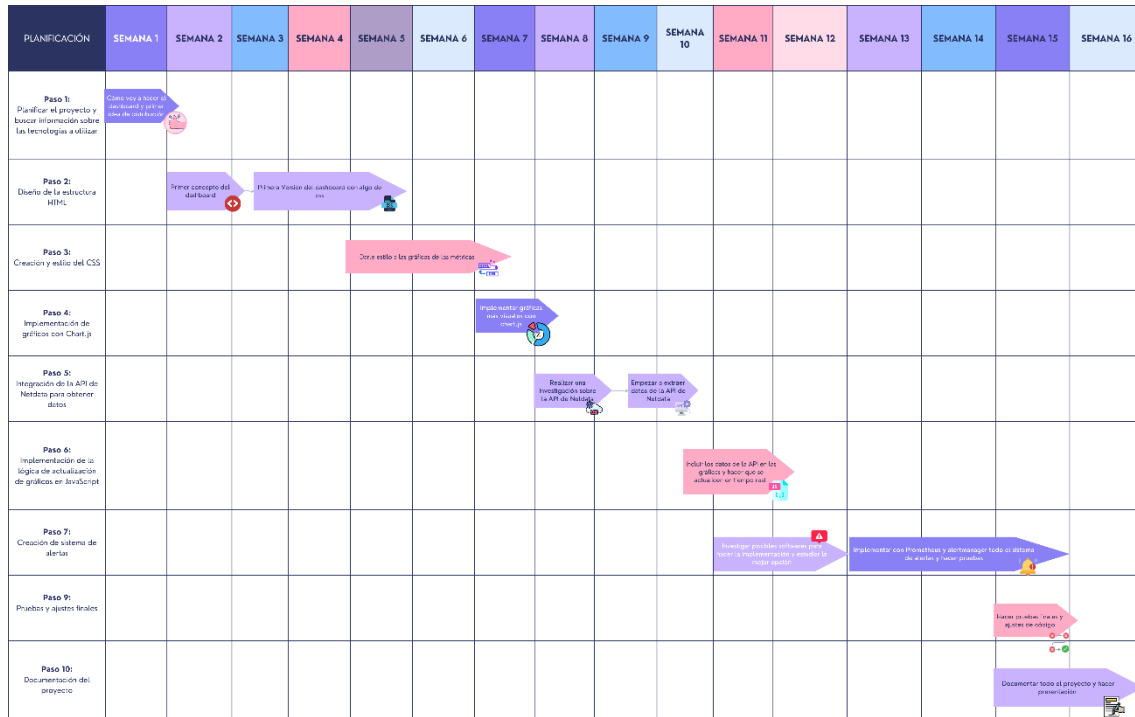
### **Estimación del tiempo necesario para cada tarea:**

- Planificación del proyecto: 20 Horas
- Diseño de la estructura HTML: 40 horas.
- Creación y estilo del CSS: 30 horas.
- Implementación de gráficos con Chart.js: 20 horas.
- Integración de la API de Netdata para obtener datos: 30 horas.
- Implementación de la lógica de actualización de gráficos en JavaScript: 10 horas.
- Creación de sistema de alertas: 50 horas.
- Implementación del cambio de tema y css responsive: 5 horas.
- Pruebas y ajustes finales: 10 horas.

- Documentación del proyecto: 20 horas.

**Total estimado: 235 horas.**

## Diagrama de Gantt



Ubicación: Proyecto/PDFs/Gantt.png

## Realización del Proyecto

### Trabajos realizados

#### 1. Estructura HTML:

- Se creó una página HTML con elementos contenedores para los gráficos de CPU, RAM y disco.
- Se agregaron elementos para mostrar la instancia del sistema y los botones para alternar temas y mostrar/ocultar el menú de targets.

#### 2. Estilo CSS:

- Se definieron los estilos CSS para el dashboard, incluyendo la disposición de los gráficos, etiquetas y alertas.
- Se establecieron estilos para el cambio de tema entre claro y oscuro.
- Se implementaron estilos responsivos para dispositivos móviles.

#### 3. Gráficos con Chart.js:

- Se utilizó la biblioteca Chart.js para implementar gráficos tipo "doughnut" para CPU, RAM y disco.
- Se agregaron gráficos de líneas para mostrar el uso del disco en lectura y escritura.

#### **4. Integración con API de Netdata:**

- Se desarrollaron funciones en JavaScript para obtener datos en tiempo real de la API de Netdata.
- Estos datos se utilizaron para actualizar los gráficos con la información más reciente sobre el rendimiento del sistema.

#### **5. Actualización de gráficos:**

- Se implementó la lógica necesaria para actualizar dinámicamente los gráficos cada segundo utilizando los datos obtenidos de la API de Netdata.

#### **6. Sistema de alertas:**

- Se añadieron alertas visuales para indicar cuando el uso de los recursos (CPU, RAM, disco) supera ciertos umbrales predefinidos.
- Las alertas se muestran con diferentes colores según la gravedad de la situación (crítica, advertencia, información, éxito).

#### **7. Cambio de tema:**

- Se implementó una función en JavaScript para alternar entre temas claro y oscuro.
- Al hacer clic en el botón correspondiente, se cambia el tema de la página y se ajustan los estilos CSS en consecuencia.

Tabla.

PASOS	TAREA	DESCRIPCIÓN	SEMANAS	PERFIL	COSTE POR HORA	HORAS	COSTE TOTAL
1	Planificar el proyecto y buscar información sobre las tecnologías a utilizar	Identificar tecnologías y realizar planificación inicial	1	Gestor de Proyecto	€30	20	€600
2	Estructura HTML	Crear una página HTML con contenedores para los gráficos de CPU, RAM y disco	2	Desarrollador Front-end	€30	40	€1200
3	Estilo CSS	Definir estilos para el dashboard, incluyendo la disposición de gráficos y cambio de tema	3 - 4	Desarrollador Front-end	€30	30	€900
4	Gráficos con Chart.js	Implementar gráficos "donut" para CPU, RAM y disco, y gráficos de líneas para uso del disco	6 - 7	Desarrollador Front-end	€30	20	€600
5	Integración con API de Netdata	Desarrollar funciones en JavaScript para obtener datos en tiempo real de la API de Netdata	8 - 9 - 10	Desarrollador Back-end	€50	30	€1500
6	Actualización de gráficos	Implementar lógica para actualizar gráficos cada segundo con datos de la API	11 - 12	Desarrollador Full Stack	€50	10	€500
7	Sistema de alertas	Añadir alertas visuales cuando uso de recursos supera ciertos umbrales	13 - 14	Desarrollador Full Stack	€50	30	€1500
8	Cambio de tema	Implementar función para alternar entre temas claro y oscuro	14 - 15	Desarrollador Front-end	€50	5	€250
9	Pruebas y ajustes finales	Realizar pruebas finales y ajustes de código	15 - 16	QA (Quality Assurance)	€30	10	€300
10	Documentación del proyecto	Documentar todo el proyecto y preparar presentación	16	Gestor de Proyecto	€30	20	€600

Ubicación: Proyecto/PDFs/tabla.png

## Problemas encontrados

- **Retrasos en la obtención de datos:**
  - Se encontraron problemas con la latencia en la respuesta de la API, lo cual fue solucionado mediante la implementación de caché local.
- **Compatibilidad de Chart.js:**
  - Hubo problemas iniciales con la configuración de Chart.js para gráficos de líneas, que fueron solucionados consultando la documentación oficial.
- **Sistema de alertas:**



- Problemas para enlazar las métricas de Netdata con prometheus, problemas para enlazar prometheus con alertmanager, problemas al implementar las alertas por correo.
- Problemas para crear un histórico de alertas.

### **Modificaciones sobre el proyecto planteado inicialmente**

- Se añadió un sistema de caché para reducir la latencia en la obtención de datos de la API.
- Se optimizó el CSS para mejorar la responsividad del dashboard en dispositivos móviles.
- Se añadió un sistema de alertas en el dashboard además de un sistema que manda un correo electrónico para las alertas.

### **Posibles mejoras al proyecto**

- **Ampliar el monitoreo a otros recursos:**
    - Añadir gráficos para monitorear la temperatura del sistema y el uso de la red.
  - **Implementar notificaciones push:**
    - Integrar notificaciones push para alertar a los administradores sobre altos usos de recursos.
  - **Mejorar la interfaz de usuario:**
    - Incluir más opciones de personalización y mejoras visuales en la interfaz del dashboard.
- 

### **Medios Necesarios**

Librería de chart.js, markdown, API netdata, HTML, CSS, JAVASCRIPT.

Prometheus, Alertmanager, Canva

---

### **Bibliografía**

- **Chart.js Documentation:**
  - <https://www.chartjs.org/docs/latest/>
- **Netdata API Documentation:**
  - <https://learn.netdata.cloud/docs/agent/web/api>
- **Prometheus Documentation:**
  - <https://prometheus.io/docs/introduction/overview/>
- **Alertmanager Documentation**
  - <https://prometheus.io/docs/alerting/latest/alertmanager/>

---

## Fuentes del Proyecto

- **HTML, CSS y JavaScript:**
  - Los archivos `index.html`, `style.css`, y `script.js` están incluidos en el repositorio del proyecto.

## Anexos

- **Capturas de pantalla:**
  - Se incluyen capturas de pantalla del dashboard en funcionamiento en la carpeta `screenshots`.