**1. Identificación y Priorización de Drivers**

**Drivers Funcionales**

1. Monitoreo en tiempo real del consumo energético.
2. Automatización del consumo según hábitos y clima.
3. Integración con sensores y dispositivos IoT.
4. Gestión remota desde app móvil o web.
5. Alertas y recomendaciones inteligentes.

**Drivers No Funcionales**

1. Escalabilidad para múltiples dispositivos.
2. Tiempo de respuesta < 2 segundos.
3. Disponibilidad del 99% del tiempo.
4. Seguridad en la comunicación y acceso.
5. Mantenibilidad y actualizaciones sin interrupciones.

**Matriz Impacto-Esfuerzo-Riesgo y Justificación**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Driver | Impacto | Esfuerzo | Riesgo | Prioridad | Justificación |
| |  | | --- | | Monitoreo en tiempo real |  |  | | --- | |  | | 5 | 3 | 3 | 11 | |  | | --- | | Clave para la eficiencia, moderado esfuerzo con IoT, riesgo medio. |  |  | | --- | |  | |
| |  | | --- | | Automatización del consumo |  |  | | --- | |  | | 5 | 4 | 4 | 13 | |  | | --- | | Es el centro del ahorro de electricidad, requiere lógica adaptativa compleja. |  |  | | --- | |  | |
| |  | | --- | | Gestión remota |  |  | | --- | |  | | 4 | 3 | 2 | 9 | |  | | --- | | Mejora la usabilidad, esfuerzo medio y bajo riesgo. |  |  | | --- | |  | |
| Escalabilidad | 5 | 2 | 4 | 11 | |  | | --- | | Imprescindible para hogares con muchos dispositivos, bajo esfuerzo inicial. |  |  | | --- | |  | |
| |  | | --- | | Tiempo de respuesta < 2 seg. |  |  | | --- | |  | | 4 | 4 | 3 | 11 | Importante para la experiencia, requiere optimización en el sistema para evitar retrasos. |

**2. Selección y Fundamentación de Patrón Arquitectónico**

**Patrón elegido: Microservicios**

El patrón de microservicios permite escalar independientemente los módulos de monitoreo, automatización y gestión remota, alineándose con los drivers de escalabilidad, tiempo de respuesta y mantenibilidad. Además, facilita actualizaciones sin interrumpir el sistema y permite distribuir la carga entre servicios, lo cual favorece la disponibilidad.

**3. Diseño del Componente Adaptativo**

El componente adaptativo (implementado como un nuevo microservicio llamado gestion\_dinamica) responde a tres de los drivers más importantes que se deben satisfacer: monitoreo en tiempo real, automatización del consumo y escalabilidad.

Este módulo funciona como un componente autónomo pero integrado, capaz de conectarse con sensores o dispositivos eléctricos, interpretar los datos de consumo a medida que se generan y tomar decisiones automáticas en función de umbrales previamente definidos.

**Ejemplo del componente adaptativo en pseudocodigo (simplificado para el ejercicio):**

// Módulo Adaptativo para Gestión Eléctrica

// Inicialización del sistema

Iniciar módulo\_adaptativo

Conectar a base de datos

Conectar a broker MQTT para recibir datos de sensores

// Bucle principal del monitoreo

Mientras sistema esté activo:

// MONITOREO EN TIEMPO REAL

mensaje ← recibir\_datos\_MQTT() // por ejemplo: {id\_dispositivo, consumo\_actual, timestamp}

guardar\_dato\_en\_BD(mensaje)

// AUTOMATIZACIÓN DEL CONSUMO

Si consumo\_actual > umbral\_dispositivo(id\_dispositivo):

enviar\_comando\_apagar(id\_dispositivo)

registrar\_evento("Dispositivo apagado por sobreconsumo")

Si consumo\_actual < mínimo\_operativo(id\_dispositivo):

enviar\_comando\_encender(id\_dispositivo)

registrar\_evento("Dispositivo encendido por eficiencia")

// ESCALABILIDAD

verificar\_dispositivos\_nuevos()

Si hay\_dispositivos\_nuevos:

registrar\_dispositivos(dispositivos\_nuevos)

esperar(1 segundo)

Finalizar módulo\_adaptativo  
  
**Análisis:** El pseudocódigo es coherente con el proyecto, puede implementarse como un nuevo microservicio con Django REST Framework, con vistas y modelos propios.

El pseudocódigo recibe continuamente el consumo energético por dispositivo. Se almacena y se puede visualizar fácilmente desde un frontend por lo que cumple con el monitoreo en tiempo real. El sistema actúa automáticamente sobre los dispositivos según reglas definidas (umbrales de consumo), optimizando el uso energético por lo que también cumple con la automatización del consumo, por ultimo el driver de escalabilidad cumple de forma parcial ya que aunque detecta dispositivos nuevos y los registra (la propuesta de escalabilidad dada), esta seria para crecer el sistema de gestión eléctrica de forma local, aun no contempla la división por usuarios, hogares o zonas y deja muy a la ligera como seria los posibles vínculos con otros microservicios.

**4. Validación Técnica**Para la validación técnica se utilizaron pruebas de carga simuladas con Locust, configurando escenarios con 100, 500 y 1000 usuarios concurrentes, generando aproximadamente 200 solicitudes por segundo (RPS) de forma sostenida durante varias horas. Además, se monitoreó el uso de la CPU y la memoria RAM del servidor local que ejecutaba la prueba.

Las métricas recolectadas incluyen:

* Tiempo de respuesta (ms)
* Total de solicitudes por segundo
* Fallos por segundo
* Uso de CPU (%)
* Uso de memoria RAM (GB)

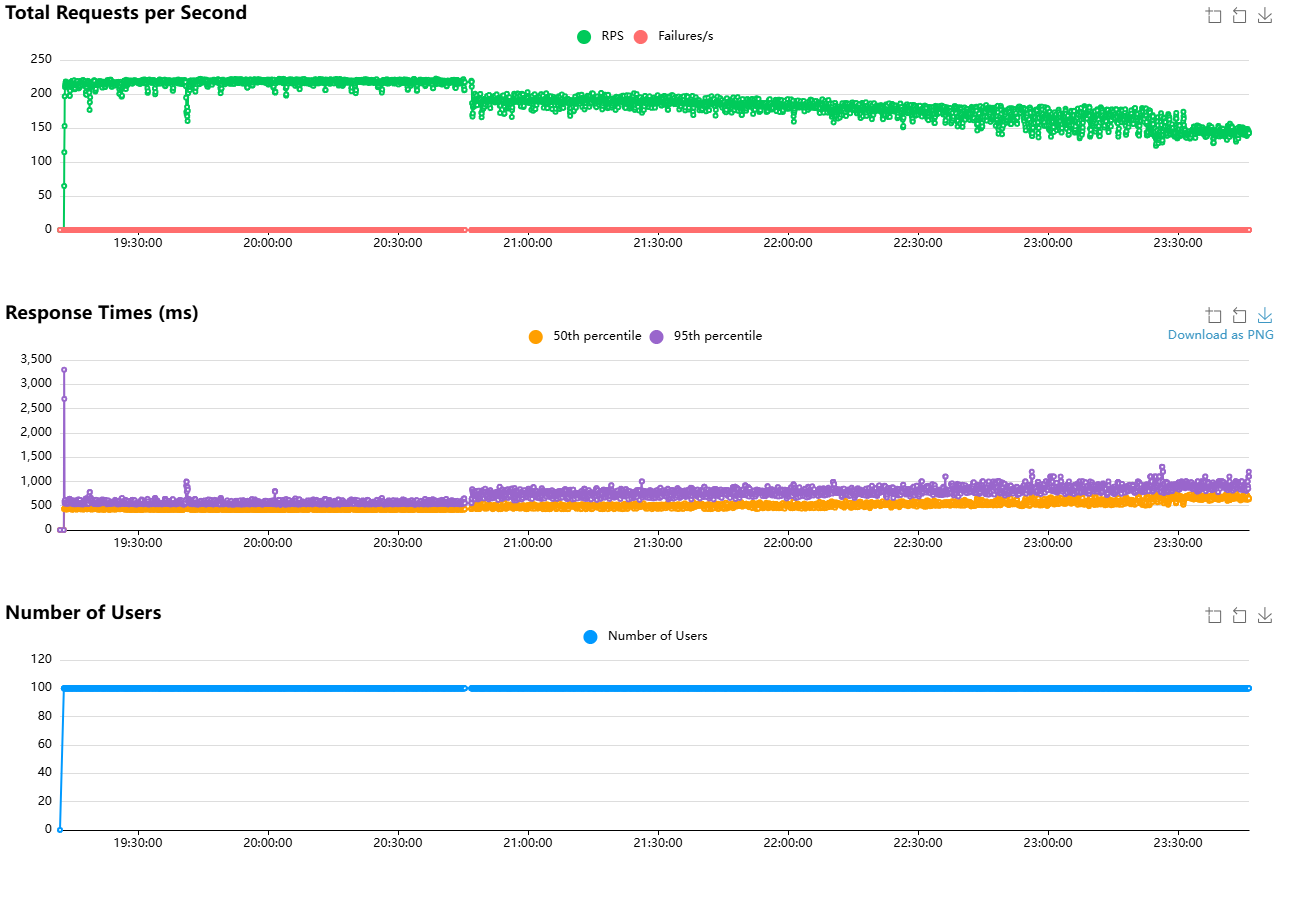
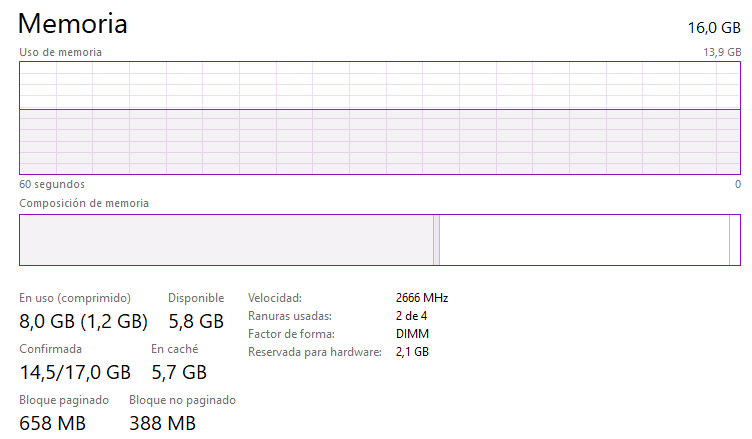
**Primer escenario: 100 usuarios  
 Análisis:** Tubo una duración de mas de 4 horas, las solicitudes por segundo fueron 200 RPS, aquí se puede observar:  
  
El **tiempo de respuesta** medio se mantiene debajo de 500 ms, lo que está dentro de los parámetros aceptables para la aplicación web. Los picos son ocasionales, lo cual es común bajo carga.  
  
La **tasa de error** fue nula ya que no se registraron errores durante el tiempo de ejecución, lo cual indica una alta estabilidad del sistema bajo la carga actual.

Imagen que contiene Gráfico de líneas

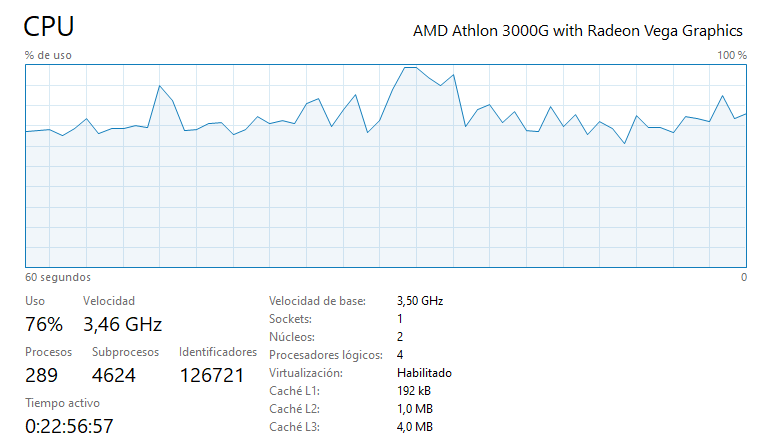
El contenido generado por IA puede ser incorrecto. El **procesador** mantuvo una carga alta pero no crítica es decir no alcanzó el 100% de uso, el sistema tenía aún cierta capacidad de procesamiento libre, lo que permite prever un comportamiento estable bajo escenarios similares.

****

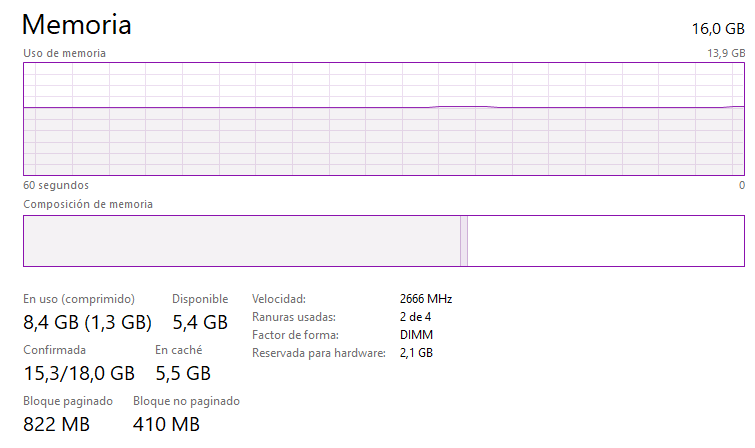
El sistema gestionó eficientemente la **memoria** disponible, sin alcanzar niveles de saturación. Esto indica que la aplicación no presentó un consumo excesivo bajo carga.

**Segundo escenario: 500 usuarios**  
Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto. **Análisis:** Tubo una duración de aproximadamente una hora, las solicitudes por segundo fueron 200 RPS, aquí se puede observar:  
  
El **tiempo de respuesta** medio fue de 2300ms, aunque el sistema se mantuvo operativo, los tiempos de respuesta son altos. El número de usuarios simulados fue de 500 durante toda la prueba, con **cero errores**, es decir sin caídas ni desconexiones. Esto demuestra estabilidad en la concurrencia aunque ya falla con el driver de rendimiento propuesto.



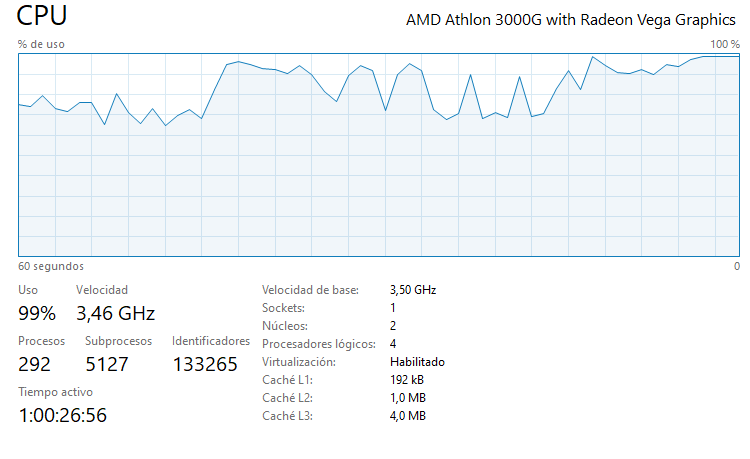
La **CPU** estuvo bajo carga significativa. Aunque no se saturó completamente, operó muy cerca del límite, lo que indica que el sistema podría no escalar mucho más sin degradación del servicio.



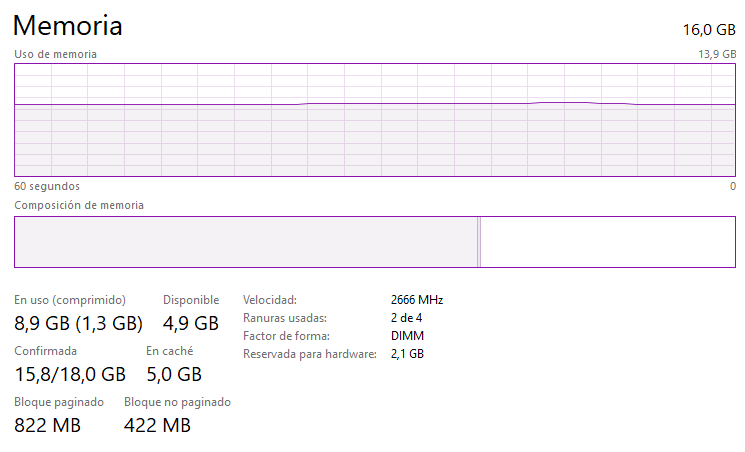
El **consumo de memoria** fue moderado. La RAM no fue un cuello de botella en esta prueba. El sistema tiene suficiente capacidad para tolerar más carga en cuanto a memoria se refiere.

**Tercer escenario: 1000 usuarios**Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto. **Análisis:** Tubo una duración de aproximadamente una hora, las solicitudes por segundo fueron 200 RPS, aquí se puede observar:  
  
El **tiempo de respuesta** medio fue de 6000ms, este valor es extremadamente alto para aplicaciones web estándar, incluso bajo carga y procesamiento como fue el caso. Indica que el sistema no logró procesar solicitudes en tiempo razonable (fue 3 veces mayor al driver de rendimiento propuesto), lo que afectaría drásticamente la experiencia del usuario en un escenario real. Aunque el sistema no colapsó (0 fallos/desconexiones), el alto tiempo de respuesta no cumplió con requisitos de usabilidad.

****

La **CPU** estuvo bajo carga alta, saturándose completamente por momentos, esto indica que los 1000 usuarios esta más allá del limite de lo que el servidor puede soportar. Aunque también hay que recordar que todas las pruebas fueron con un servidor local.

****

El **consumo de memoria** fue moderado. La RAM no fue un cuello de botella en esta prueba tampoco. El sistema tiene suficiente capacidad para tolerar más carga en cuanto a memoria se refiere.