

## Panel de administración

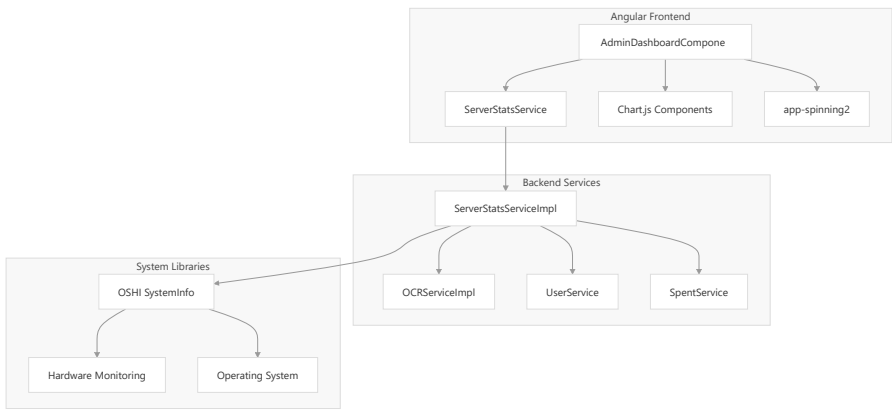
### Propósito y alcance

El Panel de Administración proporciona monitoreo del sistema en tiempo real e información administrativa para la aplicación de gestión de gastos. Muestra métricas del servidor, estadísticas de la aplicación y el estado del servicio mediante gráficos interactivos y paneles informativos. El panel está diseñado para que los administradores de sistemas monitoreen el rendimiento del servidor, rastreen el uso de la aplicación y verifiquen el funcionamiento de servicios críticos como el sistema OCR.

Para obtener información sobre los paneles de control específicos del usuario y las interfaces de gestión de gastos, consulte Panel de usuario e [interfaz de autenticación](#) y gestión de gastos .

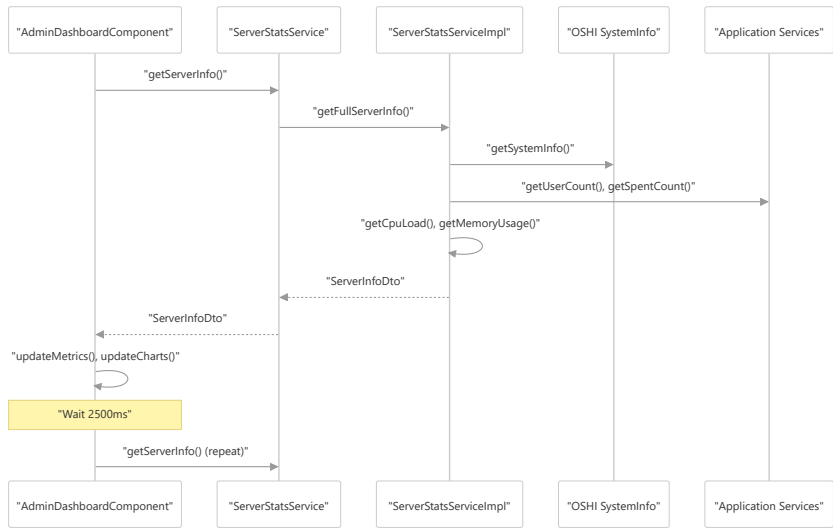
### Arquitectura de componentes

El panel de administración se implementa como un componente independiente de Angular que obtiene estadísticas del servidor en tiempo real y las presenta a través de varios componentes de visualización.



### Flujo de datos y actualizaciones en tiempo real

El panel implementa un mecanismo de actualización en tiempo real basado en sondeo que actualiza las métricas del servidor cada 2,5 segundos.



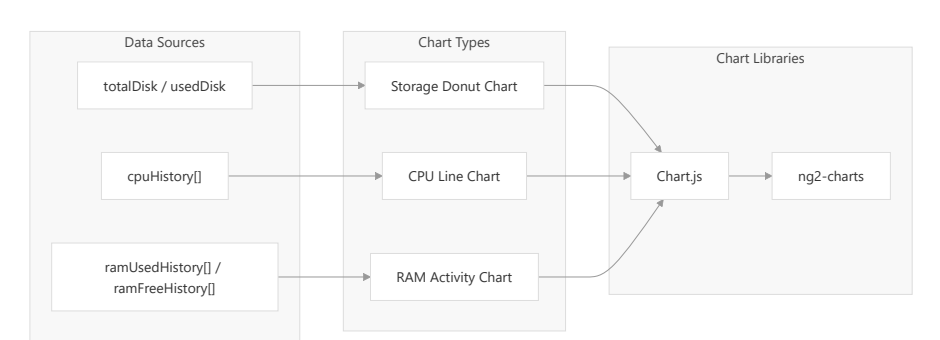
## Métricas centrales y visualización

El panel muestra cuatro categorías principales de métricas a través de tarjetas codificadas por colores y gráficos interactivos:

### Tarjetas de métricas

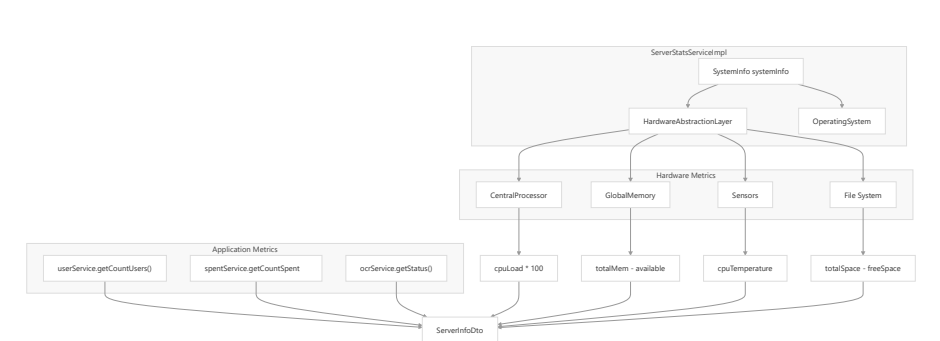
| Métrico       | Fuente de datos                                      | Objetivo                               |
|---------------|--|--|
| Gastos        | <code>spentService.getCountSpents()</code>           | Registro total de gastos en el sistema |
| Usuarios      | <code>userService.getCountUsers()</code>             | Total de usuarios registrados          |
| API de Python | <code>ocrService.getStatus().isStatusServer()</code> | Conectividad del servicio OCR          |
| LOC           | <code>ocrService.getStatus().isDemo()</code>         | Disponibilidad de procesamiento de OCR |

### Visualizaciones de gráficos



## Recopilación de información del sistema

El servicio backend utiliza la biblioteca OSHI para recopilar métricas integrales del sistema:



## Componentes y diseño de la interfaz de usuario

El panel utiliza un diseño Bootstrap responsivo con representación condicional basada en el estado de carga:

### Estado de carga

- Muestra `app-spinning2` el componente con el mensaje de carga
- Controlado por `load` bandera booleana

### Diseño del panel principal

- **Fila de métricas** : cuatro tarjetas de métricas codificadas por colores que muestran estadísticas clave
- **Fila de contenido** : diseño de dos columnas con información de almacenamiento/servidor (izquierda) y gráficos (derecha)

- **Tipos de gráficos** : gráfico de anillos para almacenamiento, gráficos de líneas para actividad de CPU y RAM

### Gestión de datos históricos

El componente mantiene datos históricos continuos para la visualización de gráficos:

```
// From admin-dashboard.component.ts
cpuHistory: number[] = [];
ramUsedHistory: number[] = [];
ramFreeHistory: number[] = [];
timestamps: string[] = [];
maxPoints = 10; // Rolling window size
```

### Transformación de datos y utilidades

El componente incluye funciones de utilidad para la presentación de datos:

#### Conversión de bytes

El `bytesToGB()` método convierte valores de bytes sin procesar a gigabytes para su visualización:

```
bytesToGB(bytes: number): number {
    return Math.round((bytes / (1024 ** 3)) * 100) / 100;
}
```

#### Formato de hora

Una tubería personalizada `SecondsToHmsPipe` formatea los segundos de tiempo de actividad en un formato legible para humanos:

```
transform(value: number): string {
    // Converts seconds to "1h 12m 35s" format
}
```

### Monitoreo del estado del servicio

El panel de control monitorea la disponibilidad de servicios críticos a través del `StatusServerResponse` DTO:

| Propiedad    | Tipo     | Objetivo                                       |
|--------------|----------|--|
| statusServer | booleano | Conectividad del servidor API de OCR           |
| demo         | booleano | Disponibilidad del modo de demostración de OCR |
| ocrLocal     | booleano | Estado del procesamiento de OCR local          |

Estos estados se reflejan en las tarjetas de métricas con fondos codificados por colores ( `bg-success` para activo, `bg-danger` para inactivo).