Propósito y Alcance

Página de configuración general accesible para todos los usuarios autenticados, proporcionando controles personalizables para la experiencia de usuario y monitoreo del sistema.

Arquitectura Detallada

Estructura de Componentes

```
javascript

      // Componentes principales renderizados

      - userInfo: Información básica del usuario (rol, sesión, última
      conexión)

      - visualizacion: Configuración de parámetros de visualización

      - notificaciones: Preferencias de notificaciones y alertas

      - sistemaInfo: Métricas y estado del sistema

- configAvanzadaLink: Enlace condicional para administradores
```

Flujo de Datos

1. Inicialización: Carga configuración existente vía `configService`
2. Renderizado: Construye interfaz con estado actual
3. Persistencia: Guarda cambios localmente y sincroniza con servidor
4. Actualización: Recarga información del sistema periódicamente

API y Dependencias

Importaciones Clave

```
javascript

      import { configService } from "../utils/configService.js";
```

```
import { ConfigAPI } from "../api.js";
import { getState } from "../state/store.js";
```

Métodos Principales

`saveVisualizationConfig()`

```
javascript
// Propósito: Persistir configuración de visualización
// Flujo:
// 1. Recopilar valores de formulario
// 2. Validar y transformar datos
// 3. Guardar mediante configService
// 4. Mostrar feedback al usuario
// 5. Loggear cambios en consola
```

`testNotifications()`

```
javascript
// Propósito: Probar sistema de notificaciones del navegador
// Flujo:
// 1. Verificar soporte del navegador
// 2. Solicitar permisos
// 3. Enviar notificación de prueba
// 4. Manejar estados de error/éxito
```

`loadExistingConfig()`

```
javascript
// Propósito: Cargar configuración persistida
```

```
      // Fuentes:
      // - configService.getVisualizationConfig()
      // - configService.getNotificationConfig()
// - Fallback a valores por defecto
```

`loadSystemInfo()`

```
javascript
      // Propósito: Obtener métricas del sistema
      // Estrategia:
      // - Primero intenta API (ConfigAPI.getGeneralConfig())
      // - Fallback a datos simulados si falla
// - Actualiza UI asíncronamente
```

Manejo de Estado

Estado Local del Componente

```
javascript
      // Configuración actual en formularios
      const currentConfig = {
        temperatureUnit: "celsius",
        chartRefresh: 1000,
        chartPoints: 60,
        browserNotifications: false,
        soundAlerts: true
};
```

Integración con Estado Global

```javascript
const { role } = getState(); // Para control de acceso
```

Interfaz de Usuario

Secciones de Configuración

Visualización

- Selector de unidad de temperatura (°C/°F)
- Intervalo de actualización de gráficos (1s, 5s, 30s)
- Límite de puntos en gráficos (10-1000)

Notificaciones

- Toggle notificaciones del navegador
- Toggle alertas sonoras
- Botón de prueba de notificaciones

Información del Sistema

- Versión de la aplicación
- Estado de conexión (online/offline)
- Contador de registros de datos
- Fecha de última actualización

Manejo de Errores

```javascript
      try {
        const success = configService.setVisualizationConfig(config);
        if (!success) throw new Error("Error guardando configuración");
      } catch (error) {
        console.error("Error guardando configuración:", error);
```

```
        alert("Error guardando la configuración");
}
```