

Explicación de como funcionan los programas

webcam_fotos.html y server.js

Archivo webcam_fotos.html (Interfaz Web con JavaScript)

Este archivo es una página web que accede a la cámara del usuario, captura imágenes automáticamente y las envía a un servidor.

Funcionamiento

1. Acceso a la cámara

- **Usa `navigator.mediaDevices.getUserMedia({ video: true })` para acceder a la cámara del dispositivo.**
- **Muestra la transmisión en vivo en un `<video>` dentro de un contenedor en la esquina superior derecha.**

2. Captura de imágenes

- **Utiliza un `<canvas>` oculto para extraer un fotograma del video y convertirlo en una imagen en formato Base64.**
- **La primera captura se realiza 1 segundo después de cargar la página.**
- **Se configura un `setInterval()` para capturar una nueva imagen cada 30 segundos.**

3. Envío de imágenes al servidor

- **Se usa `fetch()` para enviar la imagen al servidor en una solicitud POST.**
- **Se genera un nombre de archivo basado en la fecha y hora para evitar sobrescribir imágenes anteriores.**

2. Archivo server.js (Servidor en Node.js con Express)

Este archivo configura un servidor para recibir y almacenar las imágenes enviadas desde la página web.

Funcionamiento

1. Carga de dependencias

- Usa express para manejar las solicitudes HTTP.
- Usa fs y path para manipular archivos en el sistema.

2. Configuración del servidor

- Se ejecuta en el puerto 3000 y permite servir archivos estáticos (como el webcam_fotos.html).
- Incluye middleware para manejar JSON y evitar problemas de CORS.

3. Recepción y almacenamiento de imágenes

- Define una ruta /guardar-imagen que acepta solicitudes POST con imágenes en formato Base64.
- Convierte los datos de Base64 en un archivo .jpg y lo guarda en la carpeta images/.
- Devuelve una respuesta JSON confirmando que la imagen se guardó correctamente.

¿Por qué es necesario un servidor con Node.js?

1. Limitaciones del navegador

- Un navegador no puede guardar archivos directamente en el sistema (excepto mediante descargas manuales).
- Se necesita un servidor para recibir la imagen y almacenarla en el disco.

2. Comunicación cliente-servidor

- JavaScript en el navegador solo puede hacer solicitudes HTTP, pero no puede gestionar archivos en el servidor.
- Node.js con Express recibe las imágenes y las guarda en el servidor.

3. Seguridad y CORS

- El servidor incluye configuraciones para permitir el acceso desde el navegador sin restricciones.

Flujo Completo del Proyecto

- 1. El usuario abre la página webcam_fotos.html en su navegador.**
- 2. La cámara se activa y empieza a capturar imágenes automáticamente.**
- 3. Cada imagen se convierte a Base64 y se envía al servidor (server.js).**
- 4. El servidor recibe la imagen y la guarda en la carpeta images/.**
- 5. Cada 30 segundos, se repite el proceso.**

Este sistema es útil para monitoreo, seguridad o aplicaciones que requieren capturas automáticas sin intervención del usuario.