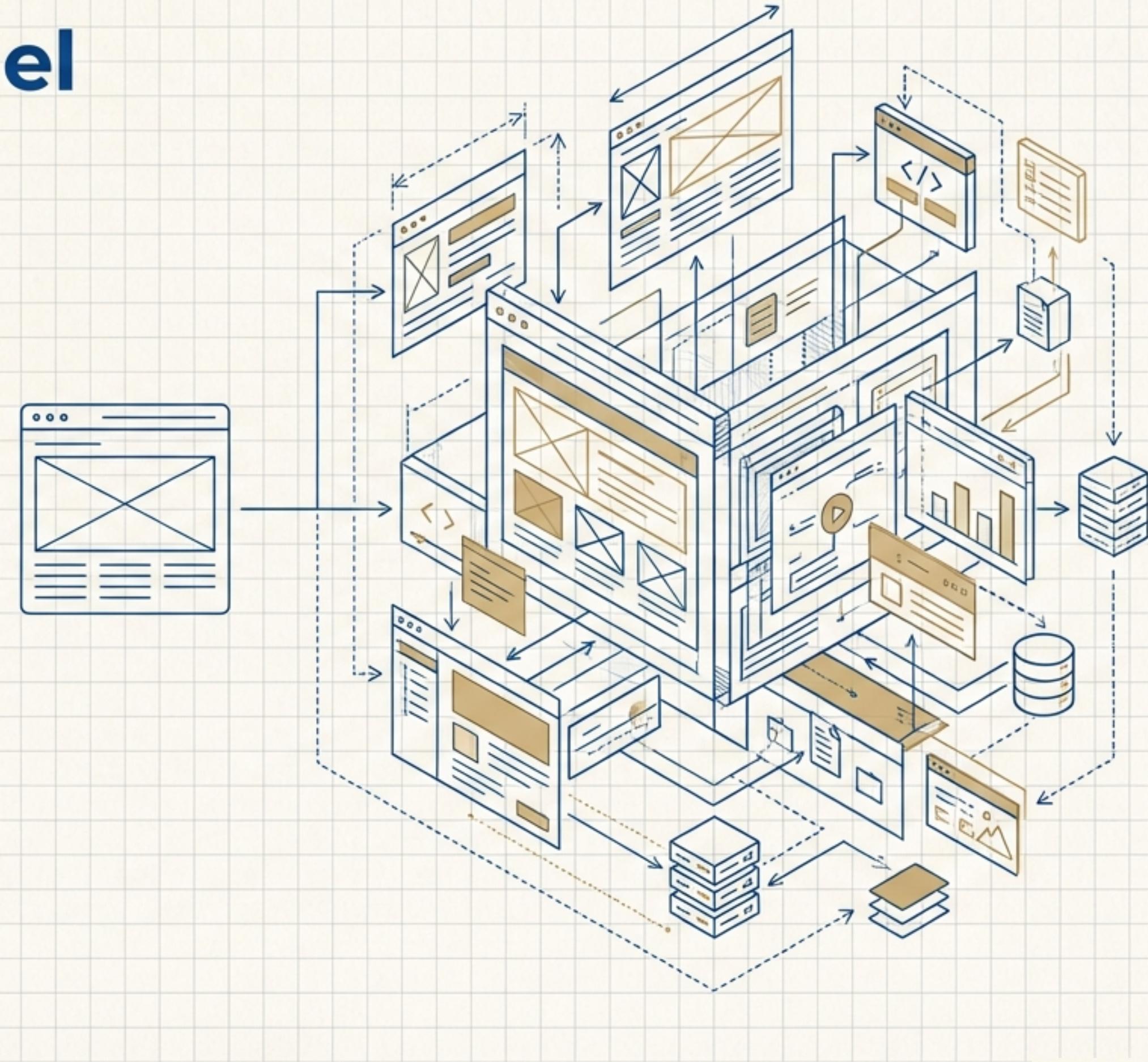


El Navegador es el Nuevo Sistema Operativo

Tus aplicaciones web ya no tienen que ser simples documentos. El navegador es ahora una plataforma con superpoderes.

En este módulo, te entregaremos las llaves para desbloquear estas capacidades. Aprenderás a construir experiencias interactivas, inteligentes y persistentes que antes imposibles, transformando la manera en que los usuarios interactúan con tus creaciones. Prepárate para el viaje.



Nuestro Plan de Vuelo: El Kit de Herramientas del Desarrollador Moderno

HABILIDADES A DESBLOQUEAR (Las APIs)



Almacenamiento Web: 'LocalStorage' y 'SessionStorage' para dar memoria a tus apps.



Geolocalización: Para que tus apps sepan dónde están en el mundo real.



Drag and Drop: Para interacciones físicas e intuitivas.



Canvas: Tu lienzo en blanco para dibujar, graficar y animar.



File API: Para interactuar de forma segura con los archivos del usuario.



Fetch API: La puerta de entrada a cualquier dato en internet.

TALLERES PRÁCTICOS (Los Ejercicios)



'Todo App' con persistencia de datos.



'App de Clima' con datos en tiempo real.



'Ordenador de Tareas' con Drag and Drop.

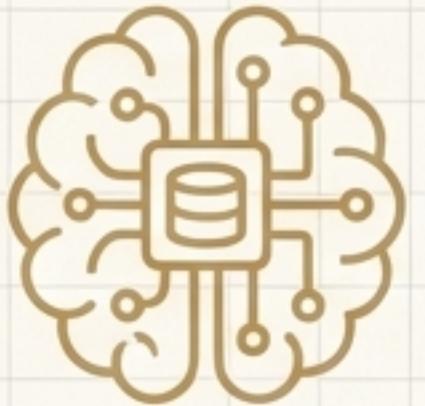


'Editor de Dibujo' con Canvas.



'Galería de Imágenes' interactiva.

Superpoder #1: La Memoria Persistente



API Spotlight

Web Storage (LocalStorage & SessionStorage)

"Dale a tus aplicaciones la capacidad de recordar preferencias, estados o datos, incluso después de recargar o cerrar el navegador."

Conceptos Clave

LocalStorage

Almacena datos sin fecha de expiración. Persiste hasta que el usuario los borra. Perfecto para configuraciones o carritos de compra.

SessionStorage

Almacena datos solo para la sesión actual. Los datos se borran cuando se cierra la pestaña. Ideal para datos temporales de un formulario multi-paso.

Código Esencial

Guardar un dato

```
// Guarda el nombre de usuario  
localStorage.setItem('usuario', 'Ana');
```

Recuperar un dato

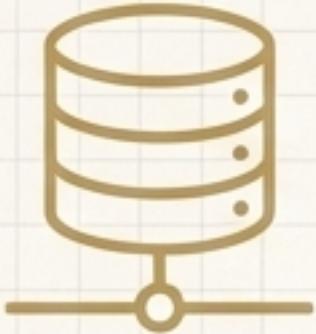
```
// Saluda al usuario al volver  
const usuario = localStorage.getItem('usuario');  
console.log(`¡Hola de nuevo, ${usuario}!`);
```

Taller Práctico: Construye una To-Do App que no olvida nada

El Desafío

Crear una lista de tareas simple donde las tareas persistan aunque el usuario cierre el navegador.

Herramientas Requeridas



Tip Profesional

localStorage solo guarda strings. Usa siempre `JSON.stringify()` para guardar objetos o arrays y `JSON.parse()` para recuperarlos.

Plan de Acción (Paso a Paso)

- 1. Maquetar la Interfaz**
Diseña el HTML con un `input` para la nueva tarea, un botón para agregar y una lista `` para mostrar las tareas.
- 2. Capturar y Guardar**
Al agregar una tarea, obtén el valor del `input` y guárdalo en un array. Convierte este array a una cadena JSON (`JSON.stringify`) y almacénalo en `localStorage`.
- 3. Leer y Renderizar**
Al cargar la página, revisa si existen tareas en `localStorage`. Si es así, obtén la cadena (`getItem`), conviértela de nuevo a un array (`JSON.parse`) y crea los elementos `` para mostrarlos en la lista.
- 4. Sincronizar la UI**
Asegúrate de que cada vez que agregas o eliminas una tarea, actualizas tanto la lista en la pantalla como el array en `localStorage`.

Superpoder #2: Conciencia del Entorno y Conexión Global



Geolocation API

Permite que tu app sepa dónde se encuentra el usuario en el mundo real para ofrecer experiencias personalizadas y contextuales.

```
navigator.geolocation.getCurrentPosition(  
  (posicion) => {  
    const { latitude, longitude } = posicion.coords;  
    console.log(`Ubicación: ${latitude}, ${longitude}`);  
  },  
  (error) => console.error('Error al obtener ubicación.')  
);
```

+

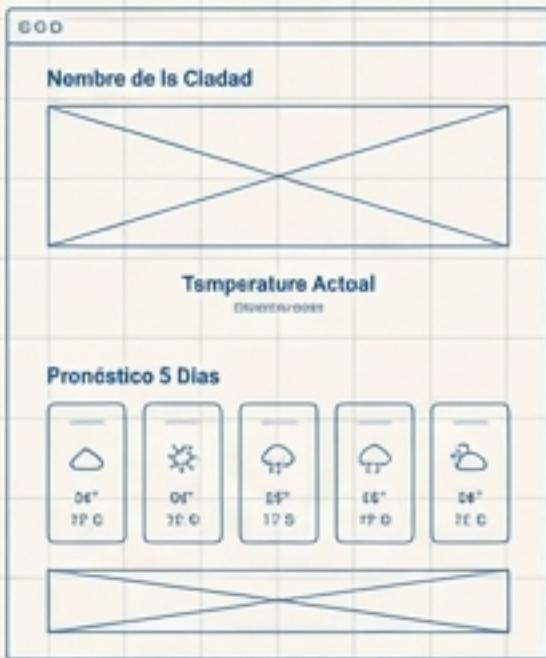


Fetch API

Conecta tu app con cualquier servicio o fuente de datos en internet de forma moderna y flexible.

```
async function obtenerDatos() {  
  const respuesta = await fetch('https://api.example.com/data');  
  const datos = await respuesta.json();  
  console.log(datos);  
}
```

Taller Práctico: Crea una App del Clima en Tiempo Real



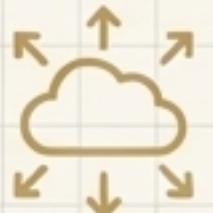
El Desafío

Construir una aplicación que detecte la ubicación del usuario y muestre el clima actual y el pronóstico, permitiendo guardar ciudades favoritas.

Herramientas Requeridas



Geolocation API



Fetch API



Web Storage

Plan de Acción (Paso a Paso)

1. Diseñar la Interfaz

Maqueta el HTML/CSS para mostrar la ciudad, temperatura actual, pronóstico de 5 días y un campo de búsqueda.

2. Obtener Coordenadas

Implementa `navigator.geolocation.getCurrentPosition()` para obtener la latitud y longitud del usuario al cargar la página.

3. Llamar a la API del Clima

Usa ``fetch`` con ``async/await`` para enviar las coordenadas a una API de clima (como *OpenWeatherMap*). Recuerda manejar tu API Key de forma segura.

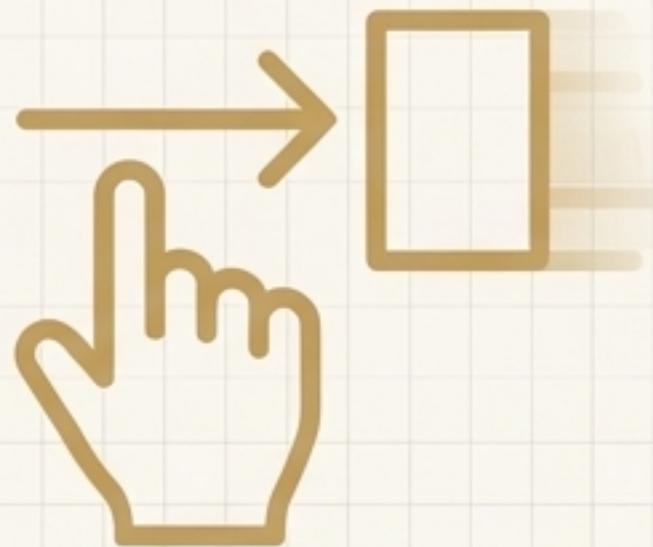
4. Visualizar los Datos

Parsea la respuesta JSON de la API y usa los datos para actualizar dinámicamente el DOM con la información del clima.

5. Añadir Superpoderes

Implementa la búsqueda por ciudad y usa ``LocalStorage`` para guardar y mostrar una lista de ciudades favoritas.

Superpoder #3: Interacción Física y Natural



API Spotlight

Drag and Drop API

Permite a tus usuarios manipular los elementos de la interfaz de forma intuitiva, arrastrándolos y soltándolos como si fueran objetos físicos.

Conceptos Clave

El Ciclo de Vida del Arrastre

`'dragstart'`

Se dispara en el elemento que comienza a ser arrastrado. Aquí se define qué datos se transfieren.

`'dragover'`

Se dispara continuamente sobre un elemento mientras otro es arrastrado sobre él. Debes prevenir el comportamiento por defecto (`event.preventDefault()`) para permitir el `'drop'`.

`'drop'`

Se dispara en el elemento 'receptor' cuando el elemento arrastrado es soltado sobre él. Aquí se realiza la lógica final (ej. mover el elemento).

El Conductor

El objeto `event.dataTransfer` es clave. Úsallo con `setData()` en `dragstart` para guardar información y con `getData()` en `drop` para recuperarla.

Taller Práctico: Un Ordenador de Tareas Interactivo

El Desafío

Crear una lista de elementos que el usuario pueda reordenar libremente mediante drag and drop, y que el orden se guarde.

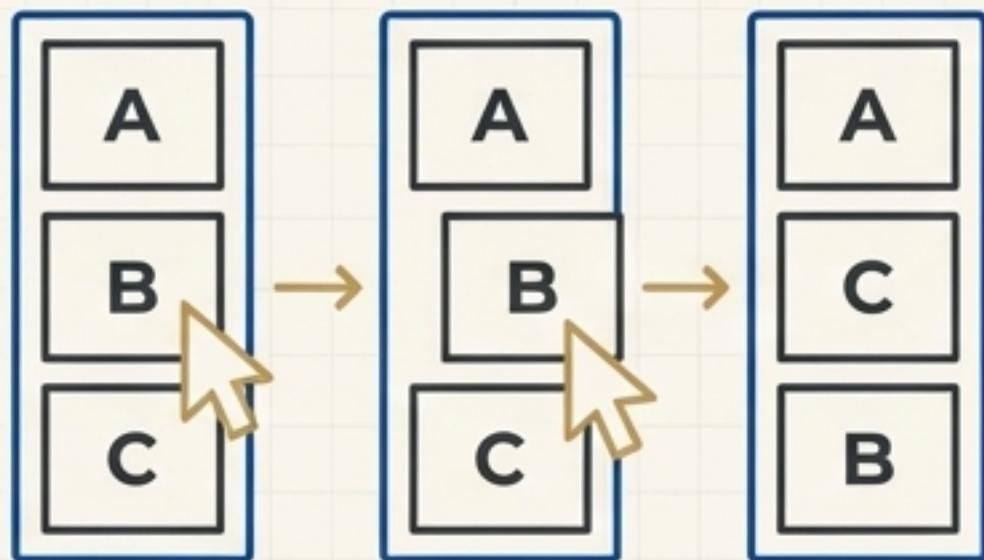
Herramientas Requeridas



Drag and Drop



Web Storage



Plan de Acción (Paso a Paso)

1 Crear Elementos Arrastrables

En tu HTML, crea una lista (``) con elementos (``) y asígnales el atributo `draggable="true"`.

2 Iniciar el Arrastre

Añade un event listener para `dragstart` a cada elemento. Dentro, usa `event.dataTransfer.setData('text/plain', element.id)` para registrar qué elemento se está moviendo.

3 Permitir Soltar

Añade un listener para `dragover` al contenedor (la lista ``). Dentro, llama a `event.preventDefault()` para indicar que es una zona de drop válida.

4 Ejecutar la Lógica

Añade un listener para `drop`. Recupera el ID del elemento con `getData()` y realiza la manipulación del DOM para reinsertar el elemento en su nueva posición.

5 Guardar el Orden

Después de un `drop` exitoso, guarda el nuevo orden de los IDs de los elementos en `localStorage`.

Superpoder #4: El Lienzo del Artista Digital

API Spotlight

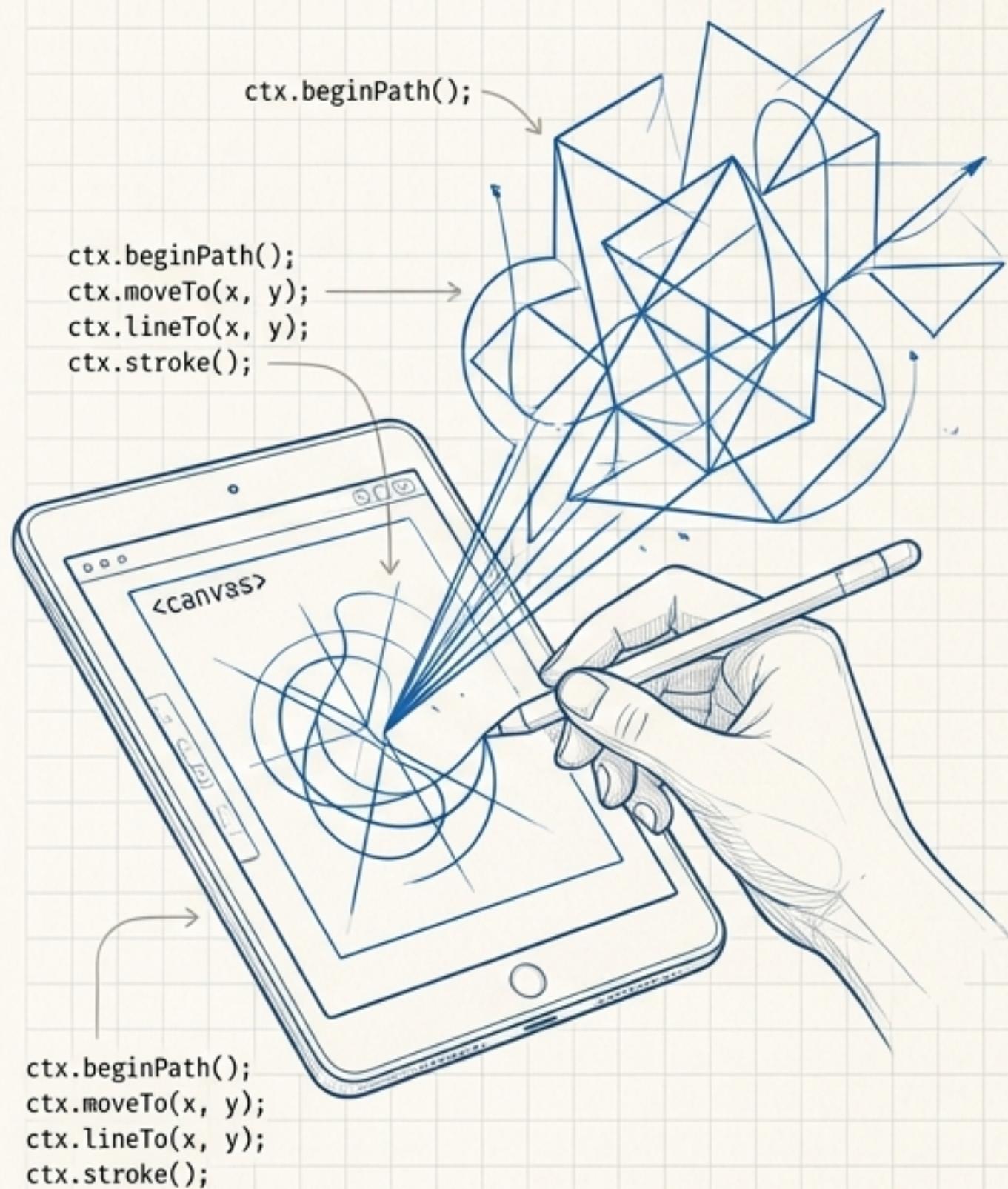


Canvas API

La Promesa: Transforma una sección de tu página en un lienzo dinámico para dibujar, crear gráficos, manipular imágenes o construir animaciones complejas.

Conceptos Clave

- **El Elemento:** Todo comienza con la etiqueta `<canvas id="miLienzo" width="500" height="500"></canvas>`.
- **El Contexto:** La magia ocurre en el 'contexto de renderizado'. Lo obtienes con JavaScript: `const ctx = canvas.getContext('2d');`. Este objeto `ctx` contiene todas las herramientas de dibujo.
- **Las Herramientas:** El contexto tiene métodos para todo: `fillRect()` para rectángulos, `arc()` para círculos, y `beginPath()`, `moveTo()`, `lineTo()`, `stroke()` para crear líneas y formas personalizadas.
- **La Interacción:** Se combina con eventos de mouse (`mousedown`, `mousemove`, `mouseup`) para crear experiencias de dibujo interactivas.



Taller Práctico: Tu Propio Editor de Dibujo

El Desafío

Crear una aplicación de dibujo interactiva con herramientas básicas como lápiz, selector de color y grosor de línea.

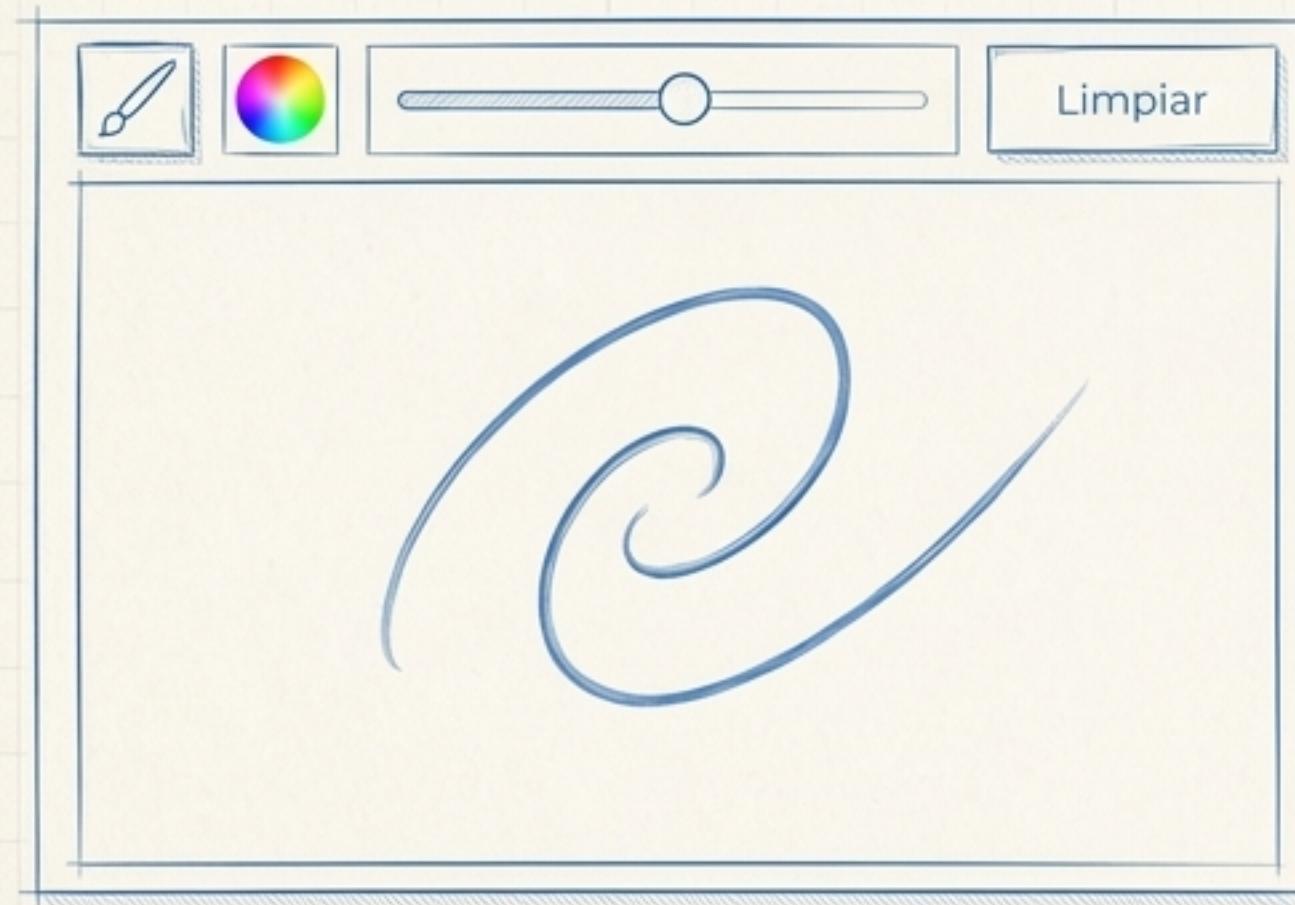
Herramientas Requeridas



Canvas API



File API



Plan de Acción (Paso a Paso)

1 Configurar el Lienzo

Prepara tu HTML con el elemento `<canvas>` y obtén su contexto 2D en JavaScript.

2 Implementar el Lápiz

Usa una variable `isDrawing` para controlar el estado.

- `mousedown`: Fija `isDrawing = true` y comienza un nuevo path con `ctx.beginPath()` y `ctx.moveTo(x, y)`.
- `mousemove`: Si `isDrawing` es `true`, dibuja una línea hasta la nueva posición con `ctx.lineTo(x, y)` y `ctx.stroke()`.
- `mouseup`: Fija `isDrawing = false`.

3 Añadir Color y Grosor

Crea `inputs` en tu HTML (ej. `<input type="color">`) y, cuando cambien, actualiza las propiedades `ctx.strokeStyle` y `ctx.lineWidth`.

4 Implementar "Limpiar"

Crea un botón que llame a `ctx.clearRect(0, 0, canvas.width, canvas.height)` para borrar todo el lienzo.

5 (Avanzado) Exportar

Usa `canvas.toDataURL('image/png')` para obtener una imagen del dibujo que se puede descargar.

Superpoder #5: Un Puente Seguro a los Archivos Locales



API Spotlight

File API

La Promesa

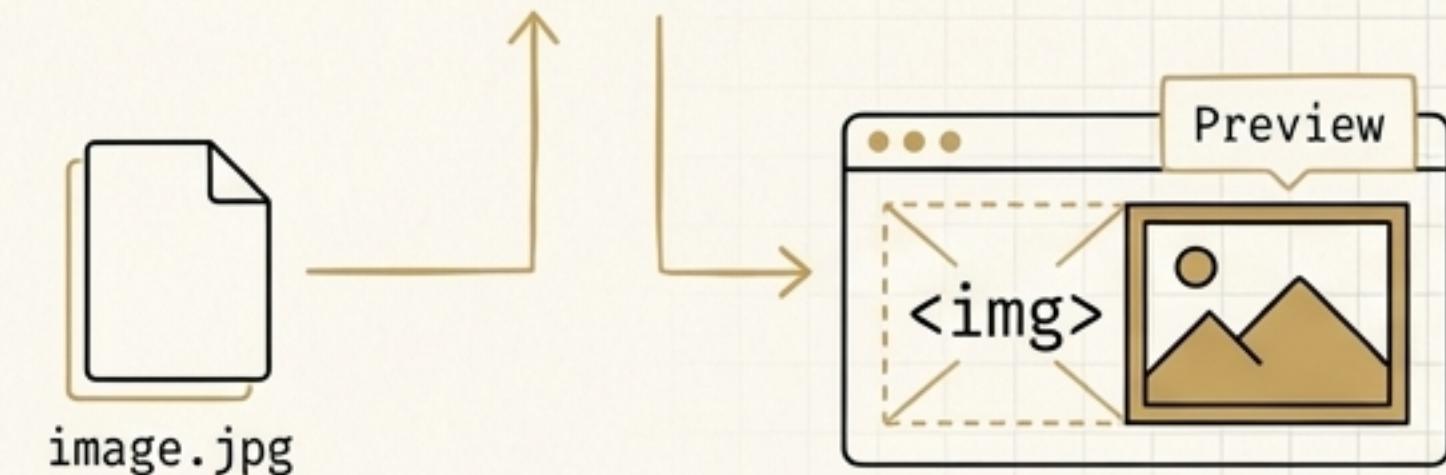
Permite que tu aplicación web lea el contenido de los archivos del usuario (imágenes, texto, etc.) de forma segura y asíncrona, directamente en el navegador.

Conceptos Clave

- **La Entrada:** El usuario selecciona archivos a través de un `<input type="file">` o arrastrándolos a una zona de drop.
- **El Objeto File:** Accedes a los archivos seleccionados a través del evento `'change'` del input. Cada archivo es un objeto con propiedades como `'name'`, `'size'`, y `'type'`.
- **El Lector (FileReader):** Este es el objeto que lee el contenido del archivo. Creas una instancia (`new FileReader()`) y usas sus métodos.

Previsualizar una imagen

```
const reader = new FileReader();
reader.onload = (e) => {
  const img = document.createElement('img');
  img.src = e.target.result; // Contiene la imagen en base64
  document.body.appendChild(img);
};
// Inicia la lectura del archivo seleccionado
reader.readAsDataURL(input.files[0]);
```



Taller Práctico: Galería de Imágenes con Carga Inteligente

El Desafío

Crear una galería que permita al usuario cargar múltiples imágenes desde su computadora, ya sea seleccionándolas o arrastrándolas, y mostrar una previsualización.

Herramientas Requeridas



File API



Drag and Drop



Plan de Acción (Paso a Paso)

1. Preparar la Interfaz

Crea un `<input type="file" multiple>` y una zona de drop visualmente identificada en tu HTML.

2. Manejar la Selección

Escucha el evento `change` en el input. Itera sobre el objeto `event.target.files` para procesar cada archivo.

3. Implementar Drag and Drop de Archivos

Añade listeners de `dragover` y `drop` a tu zona designada. En el `drop`, accede a los archivos a través de `event.dataTransfer.files`.

4. Generar Previsualizaciones

Para cada archivo, crea una instancia de `FileReader` y usa `readAsDataURL()`. En el evento `onload` del lector, crea un elemento ``, asigna el resultado a su `src`, y añádelo a tu galería en el DOM.

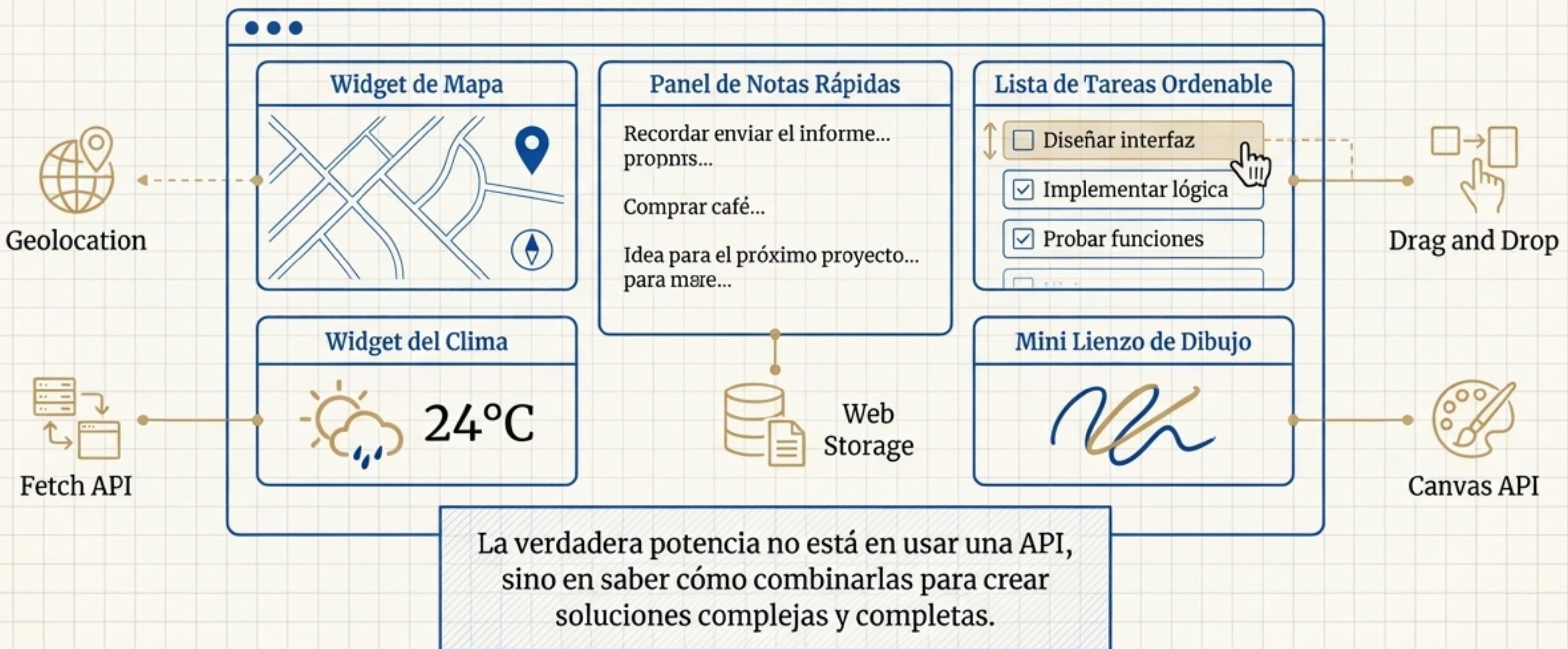
5. Mostrar Información

Junto a cada imagen, muestra datos útiles del objeto `File`, como el `name` y el `size`.

La Fusión: Cuando los Superpoderes Trabajan Juntos

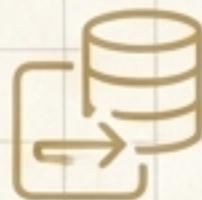
Proyecto Integrador: Explorador de APIs HTML5

Un dashboard donde cada componente es una demostración en vivo de una API diferente, todas funcionando en una sola página.



Has Completado el Viaje. Tu Kit de Herramientas Está Listo.

Nuevas Habilidades Desbloqueadas:



Persistencia de Datos

Sabes cómo usar `LocalStorage` y `SessionStorage` para dar memoria a tus aplicaciones.



Conciencia Contextual

Puedes implementar `Geolocation` para crear experiencias basadas en la ubicación.



Interacción Intuitiva

Dominas el `Drag and Drop` para crear interfaces fluidas y naturales.



Creatividad Gráfica

Tienes el poder de dibujar, animar y visualizar datos con `Canvas`.



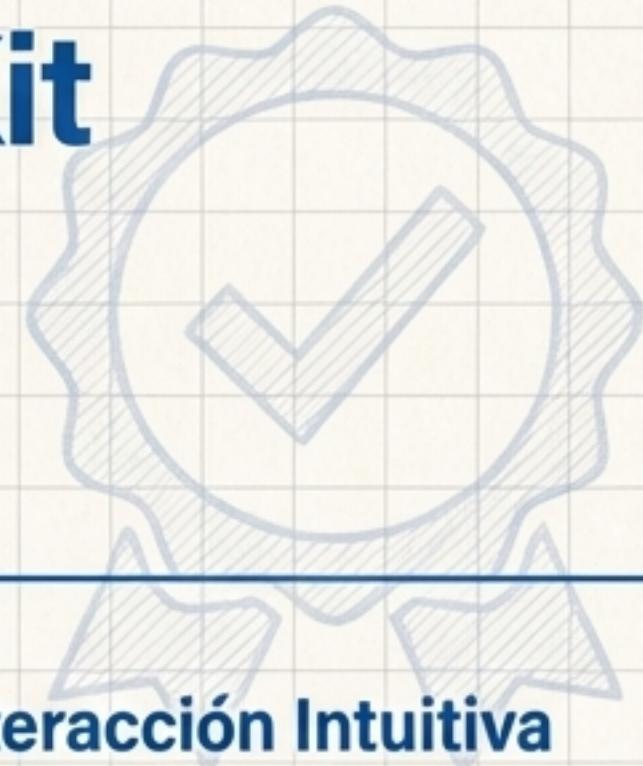
Manejo de Archivos

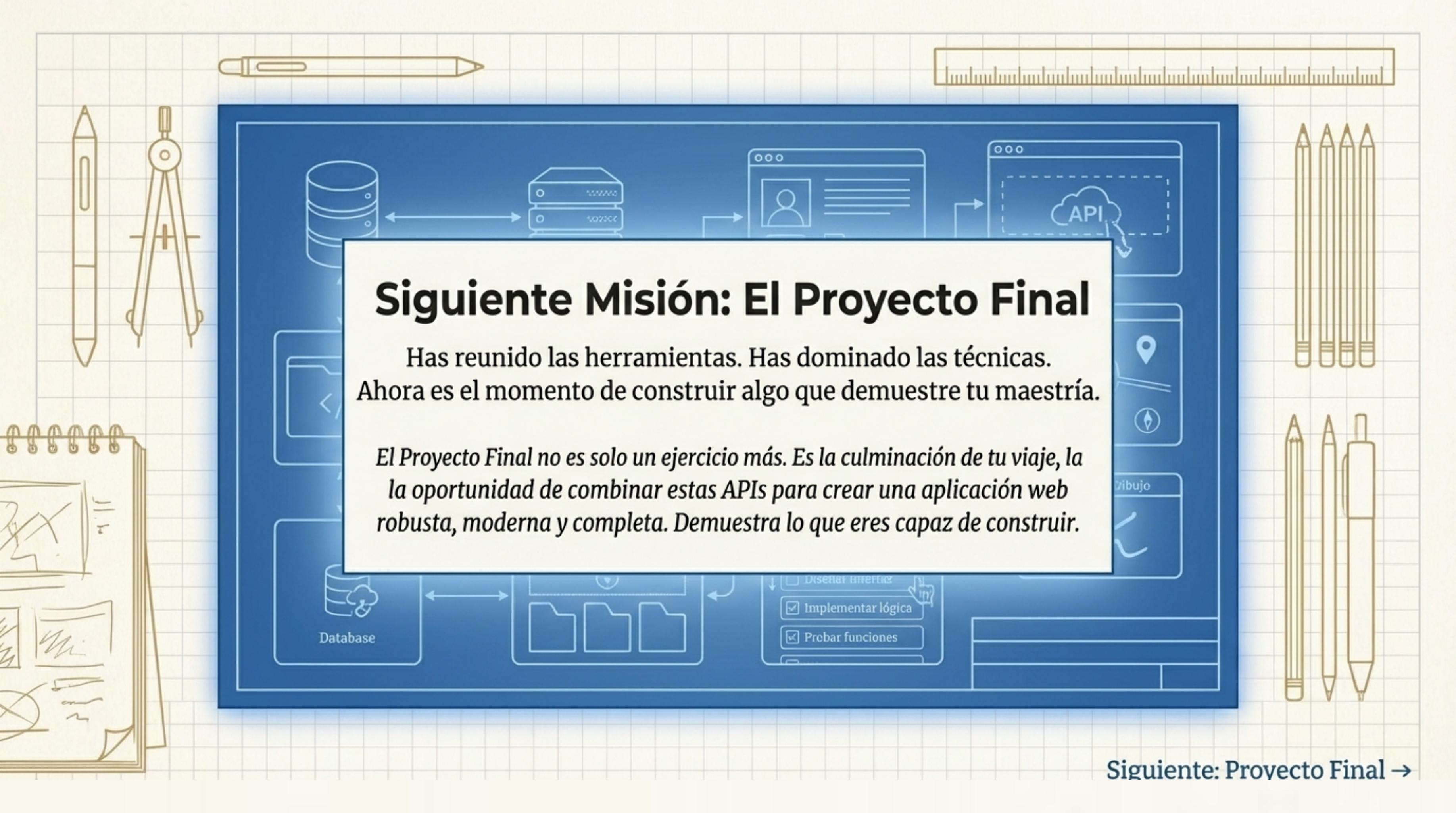
Eres capaz de interactuar de forma segura con los archivos del usuario usando la `File API`.



Conectividad Total

Sabes cómo usar `Fetch` para consumir cualquier API y traer datos del mundo exterior a tu aplicación.





Siguiente Misión: El Proyecto Final

Has reunido las herramientas. Has dominado las técnicas.
Ahora es el momento de construir algo que demuestre tu maestría.

El Proyecto Final no es solo un ejercicio más. Es la culminación de tu viaje, la oportunidad de combinar estas APIs para crear una aplicación web robusta, moderna y completa. Demuestra lo que eres capaz de construir.

Siguiente: Proyecto Final →