Reporte de usabilidad final

En general, los archivos muestran una base sólida para una buena usabilidad, especialmente en términos de accesibilidad y diseño adaptable (móvil). La estructura del contenido es lógica y fácil de seguir.

Sin embargo, el principal problema de usabilidad es la confusión y redundancia creadas por la existencia de dos calculadoras idénticas: una incrustada en la página principal (index.html) y otra en su propia página (calculadora.html). Esto crea una experiencia de usuario confusa y un mantenimiento innecesario.

## Puntos Fuertes

* Accesibilidad (A11y): Ambos archivos demuestran una excelente atención a la accesibilidad.
  + El uso de atributos aria- en el botón de hamburguesa (aria-label, aria-expanded, aria-controls) es una práctica excelente para lectores de pantalla.
  + El uso correcto de etiquetas <label> para todos los campos de <input> y <textarea> asegura que los usuarios sepan qué información deben ingresar.
* Diseño Adaptable (Móvil): La inclusión de la metaetiqueta viewport y un menú de hamburguesa en index.html indica claramente que el sitio está diseñado para funcionar bien en dispositivos móviles, lo cual es crucial para la usabilidad moderna.
* Estructura Semántica: El uso de etiquetas HTML5 como <header>, <main>, <footer>, <nav>, <section> y <article> (en index.html) organiza el contenido de manera lógica, lo que beneficia tanto a los usuarios como al SEO.
* Retroalimentación al Usuario: El formulario de contacto incluye un div (#mensaje-enviado) para notificar al usuario que su envío fue exitoso. De igual manera, la calculadora tiene un div (#resultado) para mostrar la salida. Esto es fundamental para que el usuario sepa que sus acciones tuvieron efecto.

## Problemas Críticos y Recomendaciones

### 1. Contenido Duplicado (La Calculadora)

* Problema: Existe una calculadora en la sección #3 de index.html y una calculadora *idéntica* en el archivo calculadora.html. El enlace de navegación "Calculadora" en index.html apunta a la sección interna (href="#3"), no a la página separada. Esto significa que el archivo calculadora.html es probablemente "huérfano" (ningún enlace apunta a él) y completamente redundante.
* Impacto: Causa confusión al usuario (si llegara a encontrar ambas) y duplica el esfuerzo de mantenimiento (cualquier cambio en la calculadora debe hacerse en dos lugares).
* Recomendación: Elimine el archivo calculadora.html. Dado que la navegación principal ya enlaza a la calculadora incrustada en la página de inicio, esta es la versión que debe mantenerse. Simplifique el proyecto manteniendo solo una versión de esta herramienta.

### 2. Título de la Página Principal

* Problema: El archivo index.html tiene el título <title>Document</title>.
* Impacto: Este es un título genérico que no describe el contenido. Afecta negativamente al SEO y, lo que es más importante, a la usabilidad. Los usuarios no pueden identificar la pestaña en su navegador ni entender de qué trata la página si la guardan en marcadores.
* Recomendación: Cambie el título por uno descriptivo, como <title>Captaciones de Agua Pluvial - Soluciones Sostenibles</title>, que coincida con el <h1> principal.

### 3. Enlaces de Navegación

* Problema: En index.html, el enlace de navegación " Captaciones" tiene un href="#".
* Impacto: Este es un enlace "muerto". Al hacer clic, simplemente recarga la página o lleva al usuario a la parte superior, lo cual es confuso y no cumple ninguna función.
* Recomendación: Decida qué debe hacer este enlace. ¿Debería ser el enlace "Inicio" y apuntar a index.html? ¿O debería eliminarse si no hay contenido específico de "Captaciones"? Nunca deje enlaces con href="#" en una navegación final.

## Problemas Menores y Mejoras

* autocomplete="off" en el Formulario:
  + Problema: El formulario de contacto (index.html, sección #4) usa autocomplete="off".
  + Impacto: Esto impide que el navegador ayude al usuario a rellenar automáticamente su nombre y correo electrónico, creando una fricción innecesaria. Los usuarios esperan que esta función exista.
  + Recomendación: Elimine autocomplete="off". Los navegadores modernos gestionan esto bien, y solo debe desactivarse en campos muy sensibles (como contraseñas únicas), no en un formulario de contacto.
* JavaScript "Inline" (onclick):
  + Problema: Ambas calculadoras usan onclick="calcularAgua()" directamente en la etiqueta <button>.
  + Impacto: Esta es una práctica anticuada. Mezcla la lógica (JavaScript) con la estructura (HTML), lo que dificulta el mantenimiento del código.
  + Recomendación: Mueva esta lógica al archivo .js (ej. calculadora.js). Use addEventListener para asignar el evento al botón,

FEEDBACK DE 5 USUARIOS

Comentario 1 : "¡Me encantó la calculadora! Es súper fácil de usar. Solo puse los metros cuadrados de mi techo y un aproximado de lluvia, y al instante me dijo cuántos litros podría juntar. Justo lo que necesitaba para darme una idea de qué tamaño de tinaco comprar. Muy práctica, ¡gracias!"

Comentario 2 : "Muy buena información. Vivo en una zona donde sufrimos mucho por el tandeo y las fugas, y la sección de 'Problemas' describe perfectamente la situación. Me parece excelente que no solo expongan el problema, sino que también hablen de 'Soluciones' como la mejora de infraestructura. Se nota que investigaron."

Comentario 3 : "Navegar la página desde mi celular fue muy sencillo. El menú (el de las tres rayitas) funciona bien y todo se ajusta perfecto a la pantalla. Aprecio mucho que el diseño sea limpio y que la calculadora también se pueda usar sin problemas en el teléfono."

Comentario 4 : "Hola, ¡gran iniciativa! La página es muy útil. Solo un pequeño detalle: en el menú principal, cuando le doy clic a ' Captaciones', la página solo se recarga, no me lleva a ningún lado. Me confundió un poco, pensé que me llevaría a otra sección. ¡Fuera de eso, todo genial!"

Comentario 5 : "Me gustó mucho el sitio. Tenía una duda específica sobre los materiales y usé el formulario de contacto. Me pareció muy directo y fácil de llenar, sin pedirme mil datos. Ahora a esperar la respuesta. ¡Buen trabajo con la página!"

Problemas criticos

Haz: CSS inválido con y reglas incompletas

Problema: hay bloques como @keyframes fadeIn y muchas reglas con … que rompen el parseo del CSS — el navegador ignora esas reglas y puede invalidar el resto.

Arreglo: eliminar o reemplazar los por reglas válidas.

IDs que empiezan con números (section id="1", "2", ...)

Problema: los selectores CSS como section#1 no funcionan sin escape y es fuente de fallas en tus media queries.

Arreglo: renombrar IDs (ej.: id="sec1", id="sec2") y actualizar los enlaces/hrefs.

Haz: Anchos/altos fijos en vw/vh para los items del nav (.li { width:19vw; height:14vh; })

Problema: los links ocupan demasiado espacio y se ocultan en varias resoluciones (por eso no veías Contacto).

Arreglo: usar width:auto y paddings en vez de vw/vh; permitir flex-wrap o ajustar gap.

Haz: Uso de body { width:100vw; height:100vh; } y muchos vh en layout

Problema: provoca barras de scroll, recortes y comportamientos extraños en móviles (header y menús posicionados con vh son frágiles).

Arreglo: eliminar width/height en body; usar min-height en secciones y dejar el flujo natural.

Posicionamiento del menú dependiente de header en vh (top: calc(20vh + 8px))

Problema: si cambias header en distintos dispositivos, el dropdown queda mal posicionado.

Arreglo: posicionar el menú relativo al header con top:100% en un contenedor posicionado o calcular con JS. Ejemplo: poner .header-inner { position: relative } y usar #main-nav { position: absolute; top:100%; right:1rem; } para que quede justo debajo.