

A background image showing two young people, a man and a woman, sitting at a desk. They are both smiling and looking down at their open notebooks, appearing to be working or studying together. The man is on the left, and the woman is on the right. They are both holding pens. The image is slightly blurred, giving it a soft, warm feel.

Compositor de Software: Guardando y leyendo datos: Diario Personal



01 Diario personal digital con IA

Fundamentos de un diario personal digital persistente

El objetivo es pasar de aplicaciones que se reinician al recargarse a una herramienta capaz de recordar el contenido escrito previamente.

De aplicaciones efímeras a datos que permanecen

En aplicaciones simples, el estado suele perderse al cerrar la pestaña o el navegador. Un diario digital necesita justo lo contrario: que las **entradas se conserven** aunque se cierre el navegador o incluso se apague el ordenador.

Para lograrlo se recurre a **mecanismos de persistencia en el navegador**. En este caso, se utiliza **localStorage**, que permite almacenar información de forma sencilla sin instalar servidores ni bases de datos externas.

Características clave del diario digital

La especificación de la aplicación se centra en reproducir la experiencia de escribir en un cuaderno, pero en formato web:

- **Entradas con título, fecha y contenido**, imitando el funcionamiento de un diario tradicional.
- **Interfaz a pantalla completa**, similar al editor de entradas de WordPress, para facilitar la concentración al escribir.

- **Diseño limpio y minimalista** que emula una libreta física, implementado con Tailwind CSS cargado desde un CDN.
- **Texto de placeholder inspirador** cuando todavía no existen entradas, evitando una pantalla completamente vacía.
- **Listado de entradas anteriores** que permite visualizarlas o eliminarlas, pero no editarlas, respetando la idea de “no reescribir el pasado”.

Comparación de mecanismos de almacenamiento en la web

El diario digital utiliza localStorage, pero existen otras opciones. La tabla resume las diferencias básicas entre varios enfoques.

Mecanismo	Características clave	Cuándo utilizarlo
Sin persistencia (solo memoria)	Los datos se pierden al recargar o cerrar la página. Implementación muy simple, sin almacenamiento.	Prototipos rápidos, demos y juegos donde no importa conservar el progreso entre sesiones.
Cookies	Pequeñas piezas de información que viajan en cada petición al servidor. Suelen tener fecha de caducidad.	Gestión de sesiones, preferencias ligeras o identificación entre cliente y servidor.
localStorage	Almacena datos en el navegador sin fecha de caducidad por defecto. No se borra al cerrar la ventana.	Aplicaciones cliente ligeras, como diarios personales, donde los datos pueden permanecer en un solo dispositivo.
Servidores y bases de datos externas	Requieren infraestructura adicional, autenticación y sincronización entre múltiples dispositivos y usuarios.	Aplicaciones multiusuario, servicios en la nube y soluciones que deban sincronizar datos entre varios dispositivos.

Comportamiento del diario dentro del navegador

El diario combina una interfaz sencilla con un uso básico pero eficaz de localStorage para guardar, listar y eliminar contenido.

Guardado de entradas y actualización de la vista

Al escribir una nueva entrada y guardarla, el código **serializa la información** (título, fecha y texto) y la almacena en **localStorage**. A continuación, se vuelve a generar el listado visible de entradas, haciendo que la nueva aparezca inmediatamente.

El **placeholder inicial** —el mensaje que inspira a comenzar a escribir— desaparece en el momento en que existe al menos una entrada guardada, reforzando visualmente que el diario ya está en uso.

Recuperación automática y eliminación de entradas

Cuando se vuelve a abrir el archivo HTML o se recarga la página, el script comprueba primero qué datos existen ya en `localStorage` y los utiliza para **reconstruir el listado de entradas**.

- Si hay entradas guardadas, se muestran todas siguiendo el diseño definido.
- Si no hay ninguna, se mantiene el diseño de placeholder para invitar a escribir la primera entrada.
- Al eliminar una entrada, se actualiza tanto el almacenamiento como la interfaz, manteniendo la consistencia entre ambos.

Pasos para que la IA componga el diario

Un buen resultado depende de formular un encargo claro a la IA compositora de software, detallando funcionalidad, diseño y almacenamiento.

1

Definir el objetivo de la aplicación: se especifica que se desea una aplicación web para crear un diario personal digital con entradas fechadas, título y contenido.

2

Indicar el uso de `localStorage`: se incluye explícitamente la instrucción de “guardar las entradas en `localStorage`” para que la IA incorpore la persistencia en el navegador.

3

Describir el diseño de la interfaz: se pide un editor a pantalla completa, estilo similar al creador de entradas de WordPress y un diseño minimalista que recuerde a una libreta física.

4

Concretar la tecnología a utilizar: se solicita que el resultado sea un único archivo con HTML, CSS y JavaScript, que cargue Tailwind desde un CDN y funcione completamente del lado del cliente.

5

Probar el funcionamiento: se abre el archivo con doble clic, se crean varias entradas, se eliminan algunas y se comprueba que los datos se mantienen al cerrar y volver a abrir el navegador.

i

localStorage ofrece una forma extremadamente sencilla de guardar y leer datos en aplicaciones web, pero solo mantiene la información en el mismo navegador y dispositivo; no sustituye a un sistema de copias de seguridad ni a un almacenamiento centralizado.



02 Exportar e importar el diario

Exportación e importación de entradas del diario

Aunque `localStorage` conserva los datos en el navegador, resulta útil poder sacar una copia externa o trasladar el diario a otro ordenador.

Exportar: crear una copia externa de seguridad

Se añade un botón de **exportación** que toma todas las entradas almacenadas en `localStorage` y las empaqueta en un archivo descargable. Ese archivo puede guardarse en un **pendrive**, en una carpeta local o en algún espacio en la nube.

El formato elegido es un tipo de archivo de texto estructurado, adecuado para volver a importar los datos más adelante y reconstruir el diario con fidelidad.

Importar: restaurar o migrar el contenido

La función de **importación** permite cargar un archivo previamente exportado y utilizarlo para rellenar de nuevo el diario. En la implementación descrita, el contenido importado **sobrescribe completamente** las entradas actuales.

- Antes de importar, se muestra un mensaje que explica que el diario se reescribirá con los datos del archivo seleccionado.
- Si la persona acepta, se sustituyen las entradas actuales por las del archivo y se actualiza `localStorage`.

- Alternativamente, podría pedirse a la IA que implemente una importación que **añada** las nuevas entradas en lugar de reemplazarlas.

Fortalezas de una app 100 % cliente

El diario personal digital ejemplifica las ventajas de una aplicación que funciona por completo en el navegador sin infraestructura adicional.

Sin servidores ni instalación

Basta con un único archivo HTML que puede abrirse con doble clic. No requiere despliegues complejos ni configuración de servidores.

Persistencia sencilla con localStorage

Las entradas se mantienen al cerrar o reabrir el navegador, sin necesidad de bases de datos externas ni autenticación.

Privacidad y control de datos

El contenido del diario permanece en el dispositivo de la persona autora, salvo que decida exportarlo a otro soporte.

Portabilidad mediante copias

Los botones de exportar e importar permiten trasladar el diario entre ordenadores o crear copias de seguridad periódicas.

Flujo típico de exportación e importación

El siguiente recorrido resume cómo se combinan localStorage y los archivos exportados para garantizar continuidad y respaldo del diario.

1

Escritura y guardado en el primer navegador: se crean varias entradas en el diario; cada una se almacena en localStorage y se muestra en el listado.

2

Exportación y descarga del archivo: al pulsar el botón de exportar, la aplicación genera un archivo con todas las entradas y el navegador ofrece guardarlo.

3

Traslado a otro entorno: el archivo descargado puede copiarse a un pendrive, a otra carpeta o a un servicio de almacenamiento en la nube, y desde allí llevarse a otro ordenador.

4

Importación y reconstrucción del diario: en el nuevo navegador se abre la aplicación, se pulsa en importar, se selecciona el archivo, se acepta el aviso de sobrescritura y el diario queda restaurado con las entradas exportadas.



En cualquier funcionalidad que reescriba datos —como la importación del diario— resulta fundamental mostrar avisos claros, explicar las consecuencias y ofrecer siempre la posibilidad de cancelar la acción antes de perder información existente.



Creado por Victoria, AI Founderz Fellow, y aprobado por el equipo de Founderz.



Última actualización 5 de diciembre de 2025



Este documento fue originalmente generado por la IA y revisado por nuestro equipo humano. En Founderz, utilizamos la IA de forma responsable y transparente.