



Compositor de Software: Proyectos más complejos: Gestor Kanban

Founderz

 Microsoft



01 Diseñar un gestor Kanban web

Objetivo del mini gestor de proyectos

El proyecto consiste en crear una aplicación web sencilla pero completa para gestionar proyectos mediante un tablero Kanban, usando únicamente tecnologías del lado del cliente e inteligencia artificial generativa como apoyo.

Visión general de la aplicación

El resultado buscado es un **gestor de proyectos visual** que organice el trabajo en columnas al estilo Kanban, permitiendo seguir el estado de cada proyecto de forma clara.

La interfaz se compone de varias columnas (por ejemplo, **Por hacer**, **En progreso** y **Completado**) donde se muestran tarjetas de proyectos. Cada tarjeta representa un proyecto y puede abrir una **vista de detalle en un modal**, desde la que se gestionan sus tareas asociadas.

Requisitos funcionales principales

El prototipo busca integrar en un único archivo HTML todas las funcionalidades esenciales, manteniendo un código legible y fácil de iterar con ayuda de la IA.

- **Gestión de proyectos:** crear, editar y eliminar proyectos desde la interfaz.
- **Tablero Kanban:** mostrar los proyectos en columnas según su estado.
- **Arrastrar y soltar:** mover proyectos entre columnas mediante drag & drop.
- **Vista de detalle en modal:** abrir un proyecto en una ventana emergente dentro de la página.

- **Gestión de tareas:** añadir, editar y eliminar tareas asociadas a cada proyecto.
- **Estados vacíos cuidados:** mostrar un diseño tipo placeholder cuando no haya proyectos o tareas, que anime a empezar.
- **Persistencia en localStorage:** guardar todos los estados para recuperarlos al recargar la página.
- **Diseño responsive con Tailwind CSS:** usar Tailwind desde CDN, con una paleta profesional de grises y azules oscuros y tipografías de Google Fonts.
- **Funcionamiento 100 % en cliente:** sin backend, todo el comportamiento implementado con JavaScript en el navegador.

Dimensiones clave del diseño del gestor Kanban

El proyecto combina aspectos funcionales, técnicos y de experiencia de usuario que deben alinearse para que el tablero Kanban resulte realmente útil.

Dimensión	Descripción	Aplicación en el prototipo
Funcionalidad básica	Capacidades mínimas que permiten gestionar proyectos y tareas de forma coherente.	Crear, editar y eliminar proyectos; añadir y gestionar tareas; mover proyectos entre columnas.
Experiencia de usuario	Claridad, fluidez y facilidad de uso de la interfaz.	Modal de detalle de proyecto, textos claros, placeholders que orientan cuando no hay datos, botones visibles.
Persistencia de datos	Capacidad de mantener la información entre recargas de página.	Uso de localStorage para guardar el estado de proyectos y tareas en el navegador.
Arquitectura técnica	Organización del código y tecnologías empleadas.	Un único archivo HTML con JavaScript incrustado, Tailwind CSS desde CDN y ejecución sólo en cliente.
Diseño visual	Coherencia estética y legibilidad.	Paleta de grises y azules oscuros, tipografías de Google Fonts y componentes estilizados con Tailwind.
Escalabilidad	Facilidad para añadir nuevas funcionalidades sin romper lo existente.	Estructura modular y apoyo de la IA para incorporar, de forma iterativa, nuevas mejoras y correcciones.



Los estados “vacíos” (sin proyectos o tareas) son parte esencial del diseño: un buen placeholder inspira acción y evita sensación de error.

Flujo básico de uso dentro del mini gestor

Comprender el flujo esperado dentro de la aplicación ayuda a validar que el código generado por la IA cubre el comportamiento deseado.

- 1** **Carga inicial y recuperación de datos:** al abrir la página, el script lee el contenido de **localStorage**. Si hay datos previos, reconstruye el tablero; si no, muestra un diseño placeholder que invita a crear el primer proyecto.
- 2** **Creación de un proyecto:** desde un botón principal (por ejemplo, "Nuevo proyecto"), se abre un **modal** que solicita el nombre del proyecto. Al confirmarlo, se añade una tarjeta en la columna inicial, normalmente "Por hacer".
- 3** **Apertura de la vista de detalle:** al hacer clic sobre la tarjeta de un proyecto, se abre un modal de detalle. Esta vista muestra el título del proyecto y un área específica para sus tareas.
- 4** **Gestión de tareas:** desde el modal se pueden **añadir, editar y eliminar tareas** asociadas al proyecto. Cada cambio actualiza inmediatamente la representación interna que se guardará en **localStorage**.
- 5** **Cambio de estado de un proyecto:** los proyectos se pueden **arrastrar y soltar** entre columnas para reflejar su progreso. El cambio de columna se interpreta como un cambio de estado.
- 6** **Eliminación y guardado automático:** tanto proyectos como tareas pueden eliminarse desde la interfaz. Cada acción relevante desencadena una actualización en **localStorage**, de modo que el estado se conserva al recargar la página.

Elementos que hacen funcional el prototipo

Varios componentes concretos transforman un simple HTML en un gestor Kanban útil para organizar trabajo y experimentos.

Tablero Kanban por columnas

Las columnas segmentan el flujo de trabajo en estados claros, facilitando una visualización rápida del avance de cada proyecto.

Interacción drag & drop

El arrastre y suelta de tarjetas permite cambiar estados de forma intuitiva, evitando formularios adicionales o menús complejos.

Modal de detalle del proyecto

El modal concentra la información y las tareas de un proyecto concreto, habilitando una gestión focalizada sin abandonar el tablero.

Persistencia con localStorage

La información permanece disponible tras recargar la página, lo que convierte el prototipo en una herramienta realmente utilizable.

Estilos con Tailwind CSS

Tailwind agiliza la creación de una interfaz limpia y responsive, con una paleta profesional de grises y azules oscuros.



02 IA como compositora de software

Uso de ChatGPT para generar y mejorar el código

La IA actúa como una compositora de software capaz de generar, analizar y refinar el código de una aplicación web a partir de instrucciones en lenguaje natural.

Definición del prompt inicial

El punto de partida es un **prompt detallado** donde se describe la aplicación deseada: un mini gestor de proyectos con vista Kanban, modales de detalle, gestión de tareas y persistencia en `localStorage`.

En el prompt se especifica, entre otros aspectos, que el código debe: incluirse en un **único archivo HTML**, cargar **Tailwind CSS desde CDN**, funcionar únicamente del lado del cliente y proporcionar **instrucciones de uso**. Cuanto más clara sea esta descripción, más alineado estará el código inicial con las expectativas.

Limitaciones del primer resultado

Aun con un buen prompt, el primer código generado suele ser una **versión simplificada** del sistema imaginado. En este caso pueden aparecer varios problemas:

- Funcionalidades ausentes respecto a la especificación (por ejemplo, borrar proyectos o arrastrar tareas entre columnas).
- Detalles de interfaz mejorables, como modales que aparecen abiertos al cargar la página.
- Complejidad limitada, porque una IA generalista tiende a generar soluciones manejables en pocas centenas de líneas de código.

Estas limitaciones no son un fallo, sino un punto de partida para construir un **proceso iterativo de mejora** apoyado en la propia IA.

Ventajas del enfoque asistido por IA

Trabajar con la IA como compositora de software aporta varios beneficios concretos en el desarrollo de prototipos web.

Prototipos rápidos

Permite obtener en minutos una primera versión funcional del gestor Kanban, lista para ser probada en el navegador.

Análisis automático del código

La propia IA puede leer el archivo HTML generado y **detectar qué funcionalidades faltan** respecto a una lista de requisitos.

Corrección guiada de errores

Errores visuales o de comportamiento, como modales abiertos por defecto, pueden describirse en lenguaje natural para que la IA proponga correcciones.

Iteración infinita

Es posible encadenar versiones, añadiendo funcionalidades una a una y refinando la interfaz sin necesidad de herramientas avanzadas.

Aprovechamiento de planes gratuitos

El proceso puede realizarse con la versión gratuita de ChatGPT, combinando chats nuevos cuando se llega a los límites de uso.

Ciclo iterativo de mejora del código con IA

El valor real surge al usar la IA no solo para generar código, sino para supervisarlo, diagnosticar carencias y evolucionarlo paso a paso.

- 1 Generar la primera versión:** se redacta un prompt detallado con todos los requisitos (funcionalidad, diseño, tecnologías) y se solicita el código completo en un único archivo HTML.
- 2 Probar la aplicación en el navegador:** se copia el código en un editor, se guarda el archivo (por ejemplo, *gestor-version-1.html*) y se abre en el navegador para detectar problemas concretos.
- 3 Pedir a la IA que analice su propio código:** se arrastra el archivo HTML al chat y se solicita algo como “lee este archivo y lista qué funcionalidades de esta especificación aún no están implementadas, sin crearlas todavía”.
- 4 Iniciar un nuevo chat si es necesario:** si el análisis consume muchos recursos en un plan gratuito, se puede empezar un chat nuevo, volver a adjuntar el archivo y reutilizar la lista de funcionalidades pendientes.
- 5 Solicitar cambios específicos:** se pide a la IA que haga modificaciones concretas (por ejemplo, corregir modales, añadir eliminación de proyectos y tareas, mejorar la vista de detalle) manteniendo las restricciones iniciales: un único archivo, uso de Tailwind, persistencia en localStorage.
- 6 Versionar y validar de nuevo:** se guarda el nuevo código como una versión adicional (por ejemplo, *gestor-version-2.html*) y se prueba otra vez en el navegador. A partir de ahí, se puede repetir el ciclo para seguir refinando funcionalidad y diseño.

Opciones de herramientas para proyectos de código con IA

El enfoque del mini gestor Kanban se basa en herramientas generales y gratuitas, pero existen alternativas con distintos niveles de especialización.

Enfoque	Ventajas principales	Situaciones recomendadas
ChatGPT gratuito + editor de texto	Accesible, suficiente para prototipos, permite aprender el flujo básico de generación e iteración de código.	Proyectos pequeños o educativos, pruebas de concepto y exploración de ideas sin coste adicional.
Modelos de pago o más potentes	Mayor contexto, mejor manejo de archivos extensos y respuestas más ricas, especialmente en proyectos complejos.	Aplicaciones con más funcionalidades, múltiples archivos o necesidad de análisis de código más profundo.
Entornos especializados tipo "AI coding assistant"	Integración directa con el editor, seguimiento de varios archivos, refactorización y navegación de código más avanzados.	Desarrollo de productos de software a medio y largo plazo, con bases de código crecientes.
Combinación de varios agentes de IA	Permite separar tareas: un agente centrado en código, otro en diseño de interfaz, otro en revisión de requisitos.	Proyectos donde conviene dividir el trabajo conceptual en partes y coordinar varias perspectivas automatizadas.



Una única llamada a la IA raramente produce una aplicación compleja completa; el éxito depende de la iteración controlada y del testeo frecuente.

Buenas prácticas al trabajar con la IA como compositora de software

Establecer una rutina clara de interacción con la IA permite aprovechar al máximo su capacidad generativa sin perder el control del proyecto.

Mantener una lista viva de requisitos

Resulta muy útil conservar, en un documento aparte, una **lista estructurada de funcionalidades** deseadas. Esa lista actúa como referencia para:

1) redactar el prompt inicial; 2) comprobar qué se ha implementado realmente; y 3) indicar a la IA qué elementos siguen faltando o deben mejorarse.

Cuidar las instrucciones sobre el código

Conviene indicar explícitamente a la IA aspectos como:

- "Incluir todo el código en un solo archivo HTML".
- "Cargar Tailwind CSS desde CDN".

- “Guardar el estado completo en localStorage”.
- “No eliminar funcionalidades ya implementadas que funcionen correctamente”.

Estas instrucciones reducen el riesgo de que una nueva versión rompa algo que ya estaba funcionando.

Versionar y probar con frecuencia

Guardar cada iteración en un archivo distinto (por ejemplo, *gestor-version-1.html*, *gestor-version-2.html*) facilita volver atrás si alguna modificación introduce errores.

Tras cada cambio generado por la IA, es recomendable realizar pruebas manuales: crear proyectos, moverlos entre columnas, añadir y eliminar tareas, recargar la página y comprobar que localStorage mantiene el estado esperado.



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.