



Compositor de Software: Planificador de Menús Semanales



01 Planificador web de menús semanales

Descripción general del planificador de menús

El planificador de menús semanales es una aplicación web sencilla que permite organizar desayunos, comidas y cenas de forma visual y práctica.

Objetivo de la aplicación

El propósito central es facilitar la **planificación semanal de las comidas**, reduciendo la improvisación diaria y favoreciendo decisiones más saludables. La herramienta resulta útil tanto para familias como para personas que viven solas.

La aplicación se apoya en una **IA generativa** (por ejemplo, ChatGPT o Copilot) y en un editor de texto como Visual Studio Code en su versión web, que permiten generar el código necesario de forma rápida a partir de un buen prompt.

Estructura básica de la interfaz

La interfaz se organiza en una **tabla semanal** con siete columnas (los días de la semana) y tres filas principales: desayuno, comida y cena. Cada celda admite la introducción libre de platos.

- **7 columnas:** lunes a domingo.
- **3 filas:** desayuno, comida y cena.
- **Celdas editables** para escribir los platos deseados.

Además, se incluyen botones para **generar sugerencias aleatorias de platos saludables**, limpiar toda la tabla y **crear una lista de la compra** basada en los platos elegidos.

Tecnologías y diseño empleados

La aplicación se implementa en un único archivo que contiene **HTML, CSS y JavaScript**. El estilo se define principalmente mediante **Tailwind CSS**, cargado desde un CDN, lo que permite un diseño responsivo y moderno.

El diseño busca ser **atractivo y agradable**: se emplean colores, emojis y una distribución clara que facilite la lectura del menú en diferentes dispositivos (ordenador, tableta o móvil).

Pasos para obtener la primera versión funcional

El primer objetivo consiste en conseguir una versión sencilla pero operativa del planificador, generada automáticamente por la IA a partir de un prompt bien definido.

- 1 Preparar el entorno:** Abrir un editor de texto (por ejemplo, Visual Studio Code en versión web) y tener acceso a una herramienta de IA generativa como ChatGPT o Copilot.
- 2 Redactar el prompt base:** Explicar que se desea una aplicación web para planificar el menú semanal, indicando la tabla con siete columnas (días) y tres filas (desayuno, comida y cena).
- 3 Especificar funcionalidades clave:** Pedir que permita introducir platos en cada celda, que exista un botón para generar sugerencias aleatorias de platos saludables y otro para limpiar toda la tabla.
- 4 Incluir la lista de la compra:** Solicitar una sección que genere automáticamente una **lista de la compra** basada en los platos seleccionados en el menú semanal.
- 5 Detallar requisitos técnicos:** Indicar que se desea un único archivo con todo el código HTML, CSS y JavaScript, que Tailwind CSS se cargue desde un CDN y que el diseño sea atractivo y responsivo.



Pedir instrucciones de uso: Incluir en el prompt que la IA proporcione **instrucciones claras para usar y ejecutar la aplicación**, asegurando que funcione al hacer doble clic sobre el archivo HTML resultante.

Componentes principales del prompt inicial

Un prompt bien estructurado aumenta la probabilidad de que la IA genere una aplicación útil desde la primera versión.

Bloque del prompt	Qué se solicita	Ejemplo de formulación
Estructura del menú semanal	Definir la tabla con días y tipos de comida.	«La aplicación debe mostrar una tabla con siete columnas (días) y tres filas: desayuno, comida y cena».
Entrada de platos	Permitir introducir manualmente el contenido de cada celda.	«Permitir a la persona usuaria introducir los platos en cada celda de la tabla».
Sugerencias saludables	Generar menús aleatorios orientados a la salud.	«Introducir un botón para generar sugerencias aleatorias de platos saludables para cada comida».
Lista de la compra	Crear automáticamente los ingredientes necesarios.	«Incluir una sección para añadir una lista de la compra basada en los platos elegidos».
Diseño y tecnología	Aspecto visual y carga de Tailwind CSS.	«Tener un diseño atractivo, usar emojis, colores y ser responsivo utilizando Tailwind CSS cargado desde un CDN».
Código y ejecución	Formato del archivo y forma de ejecución.	«Incluir todo el código HTML, CSS y JavaScript en un solo archivo y asegurar que se pueda ejecutar haciendo doble clic».

Ventajas de digitalizar el menú semanal

Disponer de un planificador web ofrece beneficios prácticos y organizativos en el día a día, especialmente en hogares con varias personas.

Mejor organización doméstica

Contar con el menú semanal visible ayuda a anticipar preparaciones, evitar olvidos y coordinar horarios de todas las personas del hogar.

Impulso a la alimentación saludable

Las sugerencias de **platos saludables** y la visión global de la semana facilitan equilibrar verduras, proteínas, hidratos y procesados.

Ahorro de tiempo y compras eficientes

La **lista de la compra automática** reduce el tiempo dedicado a pensar qué comprar y minimiza viajes repetidos al supermercado.

Adaptabilidad a distintos perfiles

El menú puede adaptarse a familias con niñas y niños, personas con alergias o intolerancias y preferencias concretas, personalizando el prompt.



Cuanto más concreto y detallado es el prompt (estructura, funciones, estilo, contexto familiar), más alineado estará el código generado con las necesidades reales.



02 Iterar y mejorar el planificador con IA

Mejora progresiva del planificador con la IA

Una vez disponible la primera versión funcional, comienza una fase de iteración en la que la IA va refinando el código a partir del feedback recibido.

De una versión simple a menús más completos

En la primera versión, la aplicación suele proponer un único plato por comida. Mediante nuevas instrucciones se puede solicitar que en comida y cena aparezcan dos platos (primero y segundo), con combinaciones equilibradas entre sí.

También es posible indicar el contexto, por ejemplo, «menú semanal para una familia con dos adultos y tres niños», y pedir que la IA evite casos como servir pasta al mediodía y pasta por la noche o dos platos muy similares el mismo día.

Perfeccionamiento de la lista de la compra

Una mejora frecuente consiste en solicitar que la lista de la compra muestre un ítem por línea, sin agrupar varios platos separados por comas. Esto hace la lista más legible y práctica al hacer la compra.

Otra petición útil es añadir un cuadrado junto a cada ítem al imprimir, de forma que se pueda marcar con bolígrafo lo que ya se ha colocado en el carrito.

Optimización de impresión y legibilidad

La funcionalidad de imprimir requiere especial atención. Es importante que, al pulsar el botón de imprimir, se mantenga el contenido de las celdas y no aparezcan campos vacíos en la vista de impresión.

Además, se puede pedir que al cancelar la impresión no se borren los datos introducidos. También es recomendable ampliar el tamaño de los campos de texto de las comidas para que se vea completo el nombre de cada plato.

Ciclo recomendado de mejora iterativa

El trabajo con la IA sigue un ciclo de prueba, observación y ajustes sucesivos hasta alcanzar un comportamiento satisfactorio de la aplicación.

- 1 Probar la versión actual:** Completar un menú de ejemplo, generar sugerencias, crear la lista de la compra e imprimir para detectar comportamientos no deseados.
- 2 Registrar incidencias y oportunidades:** Anotar problemas concretos (por ejemplo, campos vacíos al imprimir) y mejoras deseadas (mejor distribución de platos, campos más grandes, etc.).
- 3 Redactar feedback preciso para la IA:** Explicar qué funciona, qué no y qué se espera, usando ejemplos claros: «si hay sopa al mediodía, no pongas pasta por la noche», «un ítem por línea en la lista».
- 4 Solicitar una nueva versión del código:** Pedir que se apliquen exactamente esos cambios, manteniendo el resto del comportamiento y estructura general de la aplicación.
- 5 Guardar y probar la nueva versión:** Pegar el nuevo código en un archivo distinto, probarlo y confirmar si los problemas se han resuelto o es necesario insistir en algún punto.
- 6 Repetir el proceso según sea necesario:** Aceptar que pueden ser necesarias varias iteraciones hasta que el menú, la lista de la compra y la impresión funcionen de forma estable.

Ejemplos de feedback y su impacto en la aplicación

La calidad del feedback determina la calidad de las mejoras que la IA introduce en cada nueva versión del código.

Problema detectado	Instrucción dada a la IA	Resultado esperado
Platos poco variados el mismo día	«Haz que las combinaciones sean saludables entre sí; no pongas pasta para comer y pasta para cenar».	Menús más equilibrados, evitando repeticiones excesivas del mismo tipo de plato en un día.
Solo un plato en comida y cena	«En comida y cena pon dos platos (primero y segundo) y combina bien entre ellos».	Menús con estructura de primer y segundo plato, adecuados a costumbres de muchas familias.
Varios elementos en la misma línea de la lista	«En la lista de la compra pon un ítem por línea, nunca dos o más separados por comas».	Lista de la compra limpia, en formato tipo checklist, fácil de seguir en el supermercado.
Impresión sin contenido de las celdas	«Al pulsar imprimir debe salir el menú con los platos escritos y la lista de la compra completa».	Documento imprimible que refleja el menú y la lista sin perder información.
Contenido borrado tras cancelar la impresión	«Al pulsar Cancelar en la pantalla de imprimir, no se debe vaciar el contenido de las celdas».	Conservación de los datos del menú aunque se cancele el proceso de impresión.
Campos de texto demasiado pequeños	«Haz que los campos donde se escriben las comidas sean más grandes para ver todo el texto».	Mejor legibilidad de platos largos y menús con descripciones detalladas.

Buenas prácticas al formular feedback a la IA

Ciertos hábitos de comunicación con la IA facilitan que el código generado se acerque más al comportamiento deseado.

Ser específico y contextual

Incluir detalles sobre el tipo de familia, gustos o restricciones alimentarias ayuda a proponer menús más ajustados a la realidad.

Usar ejemplos concretos

Frases como «no pongas paella de primero y risotto de segundo» ilustran claramente qué combinaciones deben evitarse.

Describir qué funciona y qué no

Indicar qué partes ya son correctas y cuáles fallan (por ejemplo, impresión o lista de la compra) orienta mejor las modificaciones.

Aceptar iteraciones sucesivas

Asumir que algunas correcciones requerirán insistir o reformular el mensaje permite alcanzar un resultado más robusto.



La IA entiende instrucciones en lenguaje natural, incluso coloquial o con errores, pero la precisión del mensaje marca la diferencia en el resultado.



03 Gestión de versiones y conversaciones con la IA

Gestión de versiones del archivo HTML

Trabajar con una IA generadora de código implica ir acumulando versiones. Una buena gestión de archivos evita pérdidas de trabajo y retrocesos innecesarios.

Por qué guardar versiones incrementales

Cada vez que la IA entrega una nueva iteración del código, conviene guardar el archivo como una nueva versión (por ejemplo, «versión 1», «versión 2», «versión 3»).

De esta forma, si una modificación introduce errores o resultados indeseados, siempre existe la posibilidad de volver a una versión anterior estable sin tener que rehacer el proceso desde cero.

Organización de carpetas y nombres

Es recomendable crear una carpeta específica, por ejemplo «Planificador de Menús Semanales», y dentro guardar cada iteración con un nombre descriptivo.

- Usar «Guardar como...» en lugar de sobrescribir el archivo existente.
- Incluir el número de versión en el nombre del archivo.
- Mantener solo las versiones relevantes para evitar confusión.

Esta estructura facilita seguir la evolución del proyecto y comparar visualmente cambios de una versión a otra.

Procedimiento sugerido para trabajar con el código generado

Un flujo de trabajo ordenado con los archivos ayuda a aprovechar mejor las capacidades de la IA y mantener el control del proyecto.

- 1 Crear un nuevo archivo de texto:** En el editor, abrir un archivo vacío cada vez que la IA proporcione una nueva versión del código.
- 2 Pegar el código completo:** Copiar todo el contenido generado por la IA y pegarlo íntegramente en el nuevo archivo.
- 3 Guardar con un nombre de versión:** Utilizar «Guardar como...» y asignar un nombre que indique la versión (por ejemplo, «planificador_menus_v2.html»).
- 4 Abrir el archivo en el navegador:** Ejecutar la aplicación haciendo doble clic sobre el archivo HTML guardado y comprobar que se carga correctamente.
- 5 Probar y documentar resultados:** Verificar el comportamiento de los botones, la generación de la lista de la compra y la impresión; anotar cualquier incidencia.
- 6 Decidir la versión de referencia:** Mantener como versión «estable» aquella que funciona mejor, y usarla como base para futuros cambios.

Uso eficaz de nuevas conversaciones con la herramienta de IA

En ocasiones interesa retomar una versión anterior del código o evitar que la IA se confunda con el historial de mensajes acumulado.

Cuándo iniciar una conversación nueva

Si se desea volver a partir de una **versión más sencilla o estable** (por ejemplo, la versión 1 con menos funcionalidades), resulta conveniente abrir una conversación nueva con la herramienta de IA.

Continuar en la misma conversación puede hacer que la IA mezcle instrucciones contradictorias de mensajes anteriores, generando resultados menos predecibles.

Cómo proporcionar el contexto mínimo necesario

En una conversación nueva, se puede **arrastrar y soltar el archivo HTML** de la versión que interesa y añadir un mensaje del tipo: «Este es un software que he creado; realiza los siguientes cambios».

Conviene repetir siempre algunas instrucciones clave:

- Pedir que genere el **código completo** y actualizado.
- Indicar que todo el código HTML, CSS y JavaScript vaya en un **solo archivo**.
- Solicitar que **Tailwind CSS** se cargue desde un CDN.
- Incluir instrucciones sobre **cómo usar y ejecutar la aplicación** haciendo doble clic en el archivo HTML.

Con este enfoque, la IA trabaja directamente sobre la versión seleccionada y se mantiene el control sobre la estructura y los requisitos técnicos del proyecto.



Conservar versiones estables y reiniciar conversaciones con la IA cuando sea necesario son dos pilares para desarrollar aplicaciones con mayor control y seguridad.



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



Última actualización 16 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.