



Compositor de Software: Introducción y primeros pasos



01 Componer software con IA, sin programar

El rol del compositor de software

La inteligencia artificial generativa permite producir código funcional a partir de descripciones en lenguaje natural. De esta capacidad surge la figura del **compositor de software**, central en el enfoque de Founderz Business School.

Definición y mentalidad

Un **compositor de software** es una persona que diseña aplicaciones sin escribir código manualmente. Su tarea principal consiste en **imaginar con precisión qué quiere construir** y traducir esa idea en instrucciones claras y estructuradas para la IA.

Más que conocimientos técnicos profundos, se requiere una **mentalidad metódica**: definir objetivos, acotar el alcance, revisar los resultados y proponer mejoras de forma iterativa, aprovechando la potencia de los modelos de IA conversacional.

Competencias clave del compositor

Para componer software con eficacia es esencial dominar la **comunicación con la IA** (prompt engineering) y aplicar una forma de trabajar ordenada. El énfasis se desplaza desde escribir código hacia **diseñar soluciones**.

- Precisar casos de uso, entradas, salidas y reglas de funcionamiento de la aplicación.
- Anticipar la experiencia de la persona usuaria y cómo interactuará con la herramienta.

- Aceptar un enfoque de ensayo, error y mejora continua apoyado en la retroalimentación de la IA.

Herramientas básicas para componer software con IA

El ecosistema mínimo para componer aplicaciones web con IA se apoya en pocas herramientas, todas gratuitas y multiplataforma.

Herramienta	Función principal	Comentarios prácticos
Navegador web (Edge, Chrome, etc.)	Acceso a las IAs conversacionales y a la versión web del editor de código.	Cualquier navegador moderno es válido; resulta útil mantener varias pestañas abiertas para cada recurso.
IA conversacional (ChatGPT, Copilot en versión gratuita)	Generar y modificar el código a partir de descripciones en lenguaje natural.	Conviene conservar el mismo hilo de conversación para mantener el contexto de la aplicación creada.
Visual Studio Code Web	Editor de texto en el navegador para pegar, visualizar y guardar el código HTML, CSS y JavaScript.	No requiere instalación; detecta automáticamente el lenguaje y colorea la sintaxis, facilitando la lectura.
Visual Studio Code instalado en local	Alternativa de escritorio, también gratuita y de software libre, disponible para Windows, macOS y Linux.	Ofrece funciones avanzadas y permite trabajar sin conexión, manteniendo una experiencia similar a la versión web.
Navegador de archivos del sistema operativo	Abrir el archivo HTML resultante mediante doble clic para ejecutar la aplicación en el navegador.	Se recomienda agrupar los archivos en una carpeta dedicada a las creaciones como compositor de software.

Ejemplos de aplicaciones sencillas con IA

La IA permite materializar ideas muy diversas en forma de pequeños programas que se ejecutan directamente en el navegador.

Planificador de menú semanal

Una aplicación que propone menús diarios para una familia, permitiendo registrar preferencias, restricciones alimentarias y rotación de platos.

Juego de banderas de países

Un quiz visual para aprender banderas y capitales, con sistema de puntuación y repaso de los fallos más frecuentes.

Mini-juegos educativos

Herramientas lúdicas para practicar contenidos escolares, como las tablas de multiplicar o vocabulario en otros idiomas.

Utilidades de productividad personal

Pequeñas aplicaciones como contadores de hábitos, listas de verificación interactivas o calculadoras especializadas, adaptadas a necesidades muy concretas.



La composición de software se apoya menos en escribir código y más en formular buenas especificaciones funcionales y en revisar con criterio lo que genera la IA. Cuanto más concreta sea la descripción del comportamiento esperado, más sencillo resultará obtener una primera versión útil.



02 Ejemplo práctico: juego de multiplicar

Proceso para generar la primera versión

Mantener una secuencia clara de trabajo simplifica la colaboración con la IA y reduce errores al ejecutar la aplicación en el navegador.

1

Definir el objetivo pedagógico del juego: La aplicación se centra en practicar las tablas de multiplicar. Debe mostrar multiplicaciones aleatorias, por ejemplo del 1 al 9, permitir introducir la respuesta, indicar si es correcta o incorrecta y llevar la cuenta de aciertos y fallos.

2

Redactar un prompt detallado para la IA: El prompt describe con precisión qué debe hacer la aplicación e incluye requisitos de diseño (aspecto atractivo, fuentes grandes, orientado a infancia, diseño responsivo). También especifica que todo el código se escriba en **un único archivo HTML** con HTML, CSS y JavaScript, que se utilice **Tailwind CSS** cargado desde un CDN y que se proporcionen instrucciones claras para ejecutar el archivo, indicando que la persona que lo solicita no es programadora.

3

Obtener el código generado por la IA: Tras enviar el prompt, la IA devuelve el código completo de la página. Es aconsejable pedir que genere el bloque íntegro de código seguido o, si aporta explicaciones, identificar y copiar únicamente la parte de código.

4

Pegar y guardar el archivo HTML: El código se pega en un editor de texto como **Visual Studio Code Web**, que reconoce el lenguaje y colorea la sintaxis. A continuación se guarda el archivo con extensión **.html** dentro de una carpeta del proyecto, utilizando un nombre sin espacios y con mayúsculas internas, por ejemplo *PracticarTablasDeMultiplicar.html*, siguiendo la convención conocida como **camelCase**.

5

Ejecutar la aplicación en el navegador: Una vez guardado el archivo, se localiza desde el explorador de archivos del sistema operativo y se abre con doble clic. El navegador mostrará la interfaz del juego y permitirá practicar las multiplicaciones de forma interactiva.

Tecnologías y diseño de la aplicación web

El juego de tablas de multiplicar es una aplicación web sencilla que combina HTML, CSS, JavaScript y Tailwind CSS en un solo archivo fácilmente compartible.

HTML, CSS y JavaScript en un solo archivo

La aplicación utiliza los tres lenguajes básicos de la web: **HTML** define la estructura (títulos, botones, campos de entrada), **CSS** controla la apariencia (colores, tamaños, alineaciones) y **JavaScript** implementa la lógica (generar operaciones, validar respuestas, actualizar marcadores).

Pedir a la IA que incluya todo en un único archivo simplifica el trabajo para quienes no dominan la programación, ya que se evita gestionar múltiples ficheros y dependencias internas. Basta con guardar un solo documento y abrirlo en el navegador para disponer de la aplicación completa.

Uso de Tailwind CSS y diseño responsivo

Tailwind CSS es una librería de estilos que ofrece clases utilitarias, como *bg-blue-100*, para definir propiedades visuales sin escribir hojas de estilo extensas. Al cargarse desde un servidor de distribución de contenido (CDN), no requiere instalación adicional.

- Permite crear una interfaz que se adapta a móviles, tabletas y ordenadores con diferentes tamaños de pantalla.
- Facilita conseguir un aspecto atractivo y moderno con pocas líneas de código de estilo.
- Asegura una apariencia coherente entre las distintas secciones o pantallas de la aplicación.

Evolución del juego de tablas de multiplicar

Mediante nuevas instrucciones a la IA, la aplicación inicial se va enriqueciendo con elementos de juego, tiempo y feedback visual.

Característica	Versión inicial	Tras mejoras con IA
Gestión del tiempo	No existe límite de tiempo; la persona usuaria responde a cada multiplicación sin prisa.	Se añade una cuenta atrás de unos 10 segundos que introduce presión temporal y dinamismo en la partida.
Sistema de puntuación	Solo se contabilizan aciertos y fallos, sin un marcador global de puntos.	Se incorpora un contador de puntos que parte de un valor alto y va disminuyendo; cuanto más rápida la respuesta, mejor puntuación.
Feedback tras responder	Se muestra un mensaje simple indicando si la respuesta es correcta o incorrecta.	El mensaje se mantiene visible durante unos segundos, mostrando también la solución correcta antes de pasar a la siguiente multiplicación.
Número de ejercicios por partida	La serie de multiplicaciones puede prolongarse indefinidamente.	La aplicación se detiene tras un número concreto de operaciones (por ejemplo, diez) y presenta un resumen final.
Visibilidad del rendimiento	La información se limita a la cuenta simple de aciertos y errores.	El resultado final combina aciertos, fallos y puntuación total, lo que permite valorar tanto la precisión como la velocidad.



Cuando se realizan iteraciones con la IA, resulta prudente guardar cada versión del archivo HTML con un nombre diferente (por ejemplo, “versión 1” y “versión 2”). Así se conserva siempre una copia funcional y, gracias a los comentarios dentro del propio código, es más sencillo entender qué hace cada bloque y aplicar nuevas mejoras con seguridad.



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.