

A man with dark hair and a beard is smiling while looking at a computer monitor. He is wearing a grey sweater. The room is dimly lit, with the primary light source being the computer screen. The text is overlaid in white on the image.

Compositor de Software: Creando a partir de una imagen



01 Componer software desde una imagen

De imagen estática a software funcional

Partir de una imagen permite describir interfaces y comportamientos visualmente, dejando que la IA infiera gran parte de la lógica necesaria.

La imagen como punto de partida

En lugar de detallar cada requisito con texto, se puede usar una **captura de pantalla** o una **fotografía** de algo que ya exista: un juego, una web o un formulario. La IA analiza la imagen y deduce elementos como botones, bloques, barras, campos de texto o distribución general.

En el caso de un juego tipo Arkanoid, basta con mostrar la pantalla de juego para que la IA identifique la **barra inferior**, la **pelota** y los **bloques a destruir**, así como la **dinámica básica de rebotes**.

Reducir la complejidad del prompt

Al aportar una imagen, el **prompt textual** puede ser mucho más breve. Una instrucción del estilo «Programa un juego como el de la imagen» resulta suficiente para que la IA entienda el tipo de juego, sin explicar que es un Arkanoid ni detallar todas sus reglas.

Lo esencial es complementar esa frase con algunos **requisitos técnicos transversales**, por ejemplo: **diseño atractivo**, uso de Tailwind CSS cargado desde un CDN, código completo en un solo archivo y pequeñas instrucciones de uso.

Generalizar más allá de los videojuegos

Este enfoque no se limita a juegos clásicos. Cualquier **interfaz visual** puede servir como especificación: un panel de control, una página de aterrizaje o un formulario complejo. La IA traduce componentes visibles en **HTML, CSS y JavaScript**.

De este modo, las ideas pueden empezar como bocetos, capturas de herramientas existentes o fotos de pizarras, y transformarse rápidamente en prototipos ejecutables.

Pasos para crear un juego tipo Arkanoid

Este proceso muestra cómo pasar de una imagen de referencia a un prototipo funcional de juego usando un asistente de IA.

- 1 Elegir una referencia visual:** Se selecciona una imagen del juego deseado, por ejemplo, buscando “Arkanoid” en un buscador de imágenes y eligiendo una captura de pantalla representativa.
- 2 Capturar la imagen:** Se realiza una captura de pantalla o se guarda la imagen elegida. También puede ser una foto tomada con el móvil de un boceto en papel.
- 3 Adjuntar la imagen en el asistente de IA:** Se crea una nueva conversación con la IA, se pega la captura o se sube la fotografía, de forma que el modelo pueda analizarla visualmente.
- 4 Redactar un prompt breve y técnico:** Junto a la imagen, se indica algo del tipo «Programa un juego como el de la imagen» y se añaden requisitos: diseño atractivo, uso de **Tailwind CSS desde CDN**, todo el código en un único archivo HTML, CSS y JavaScript integrados, e inclusión de instrucciones básicas de uso.
- 5 Probar el prototipo en un editor web:** Se copia el código generado en un editor HTML, se guarda como un archivo (por ejemplo, *juego-version-1.html*) y se abre en el navegador para comprobar que el juego funciona y que la barra y la pelota responden correctamente.

6

Anotar oportunidades de mejora: A partir de la prueba inicial, se observan posibles mejoras estéticas o funcionales (puntuación, número de vidas, corrección de rebotes, colores, etc.) para las siguientes iteraciones con la IA.

Usos clave de la generación desde imágenes

Partir de imágenes abre una serie de posibilidades prácticas en el diseño y desarrollo de productos digitales con apoyo de IA.

Recrear interfaces existentes

Permite reproducir la interfaz de un juego clásico o una web conocida, obteniendo un código base sobre el que iterar y personalizar.

Digitalizar bocetos en papel

Convierte dibujos hechos a mano de formularios, paneles o maquetas en prototipos HTML interactivos sin necesidad de maquetación manual inicial.

Explorar ideas con baja fricción

Facilita experimentar con conceptos nuevos sin escribir largos documentos de requisitos, reduciendo el tiempo entre la idea y el primer prototipo.

Mejorar la comunicación técnica

Sirve como puente entre perfiles de negocio y técnicos, al utilizar imágenes como lenguaje común para describir el resultado esperado.



La IA no solo reconoce la estética de la imagen; también infiere comportamientos, como el movimiento de la barra o el rebote de la pelota, ahorrando descripciones textuales exhaustivas.



02 Iterar, depurar y mejorar con IA

Mejora iterativa del juego generado

Una vez obtenido el primer prototipo funcional, la clave está en iterar: probar, detectar mejoras y trasladarlas a la IA con instrucciones precisas.

Feedback funcional: más allá de la apariencia

El primer prototipo puede funcionar, pero presentar pequeños problemas de jugabilidad. Un ejemplo típico es que, al rebotar, la pelota quede momentáneamente por debajo de la barra, generando sensaciones extrañas para quien juega.

Proporcionar a la IA un **feedback claro y concreto** sobre este tipo de fallos —explicando qué ocurre exactamente y qué debería ocurrir— permite que el modelo ajuste la lógica del juego sin necesidad de depurar manualmente el código.

Profundizar en la estética y las mecánicas

En iteraciones posteriores, se pueden especificar mejoras visuales y de diseño de juego: barra más **redondeada y detallada**, paleta de colores más rica, bloques y fondo con estilos más elaborados usando Tailwind CSS.

Es posible pedir también **elementos de gameplay** adicionales, como un contador de puntuación, un sistema de vidas (por ejemplo, tres vidas iniciales) o mensajes de “fin del juego”. Si el modelo lo considera útil, puede incorporar gráficos en formato SVG directamente en el código fuente.

Evolución de las versiones del juego Arkanoid

Observar cómo cambian las versiones ayuda a estructurar el proceso de mejora y a decidir qué conservar o descartar en cada etapa.

Versión	Características principales	Aprendizajes clave
Versión 1	Juego funcional con mecánica básica de Arkanoid: barra móvil, pelota que rebota y bloques que se destruyen, pero estética sencilla y sin puntuación ni vidas.	La IA es capaz de reproducir la lógica general del juego a partir de la imagen, incluso con un prompt muy breve.
Versión 2	Se solicitan mejoras visuales (barra redondeada, más colores, bloques y fondo más elaborados) y elementos de HUD (puntuación y vidas). La estética mejora, pero el juego deja de arrancar por un error en el código.	Introducir cambios complejos puede generar errores. Es necesario integrar la consola del navegador en el flujo de trabajo para detectar y corregir fallos.
Versión 3	Tras copiar a la IA el mensaje de error de la consola, se genera un nuevo código que mantiene la estética mejorada, incorpora puntuación y vidas, y vuelve a funcionar correctamente.	Proporcionar a la IA mensajes de error concretos acelera la corrección de bugs y permite conservar las mejoras logradas en iteraciones anteriores.

Uso de la consola del navegador para depurar

La consola del navegador es una herramienta clave para identificar errores cuando una versión generada por IA deja de funcionar correctamente.

- 1

Abrir las herramientas de desarrollo: En el navegador, se hace clic con el botón derecho sobre la página y se selecciona “Inspeccionar” o una opción equivalente para mostrar la barra de desarrollo.
- 2

Acceder a la pestaña “Consola”: Dentro de las herramientas de desarrollo, se selecciona la pestaña “Consola”, donde aparecen avisos, advertencias y errores de JavaScript.
- 3

Diferenciar avisos de errores reales: Algunos mensajes son simples advertencias, como el aviso de Tailwind CSS indicando que la carga desde CDN está pensada para desarrollo. Lo relevante son los **errores en rojo** que impiden ejecutar el juego.

4

Copiar el mensaje de error: Se selecciona y copia el texto del error principal asociado al fallo del juego. Es importante incluir el mensaje completo para que la IA tenga todo el contexto.

5

Trasladar el error a la IA: En la conversación, se indica que la nueva versión no funciona, se menciona que se ve la estética actualizada y se pega el error de consola. La IA utilizará esa información para localizar y corregir la causa.

6

Probar de nuevo y validar: Se reemplaza el código en el archivo correspondiente, se guarda y se recarga la página. Si el juego funciona, la versión corregida puede mantenerse como base para futuras mejoras.

Gestión de versiones y colaboración eficaz con la IA

Tratar a la IA como un colaborador técnico requiere cierto orden en las versiones y claridad al formular cada petición de cambio.

Organizar versiones de manera segura

Asignar nombres distintos a cada archivo (por ejemplo, *juego-v1.html*, *juego-v2.html*) permite **conservar versiones estables** por si una iteración introduce errores. Cuando se corrige un fallo concreto, puede ser razonable sobrescribir la versión anterior para evitar acumular código roto.

Este enfoque facilita volver rápidamente a un estado funcional si una nueva propuesta de la IA resulta problemática.

Formular cambios con precisión

Cada nueva petición a la IA debería incluir: qué **sí funciona** en la versión actual, qué **no funciona** o se desea mejorar, y qué **condiciones técnicas** deben mantenerse, como el uso de Tailwind CSS desde CDN o la inclusión de todo en un solo archivo.

- Describir el comportamiento actual y el esperado (por ejemplo, cómo debe rebotar la pelota).
- Recordar restricciones técnicas importantes que no deben perderse en la nueva versión.
- Incorporar mensajes de error de la consola cuando existan problemas de ejecución.



Combinar imágenes como especificación visual, la consola del navegador como fuente de errores y la IA como generador de código crea un ciclo de mejora muy rápido para prototipos digitales.



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.