

SEM CARTAGO


**Secretaría de Educación Municipal
Cartago, Valle del Cauca**


Laboratorio de Innovación Pedagógica con IA

12 Proyectos Interactivos para Docentes Creadores

**Integración de Prompts, LaTeX, Python
y Desarrollo Web**

 **Autor:** MSc. Néstor Fabio Montoya

 **Cargo:** Líder de Innovación Educativa

 **Fecha:** 7 de noviembre de 2025

 **Web:** profenestor.com/edulab

Índice

1	Introducción: El Docente como Creador Digital	1
1.1	¿Por Qué Este Documento Es Diferente?	1
1.2	Los 12 Proyectos: Tu Arsenal Pedagógico	1
2	Laboratorio STEM: 12 Proyectos Interactivos con Prompts Maestros	3
2.1	Proyecto 1: Simulador de Procesos Abstractos	4
2.2	Proyecto 2: Juego de Gamificación (Ruleta/Ahorcado/Memoria)	8
2.3	Proyecto 3: Calculadora Estándar con Diseño Glassmorphism	12
2.4	Proyecto 4: Simulador de Laboratorio de Física	16
2.5	Proyecto 5: Visualizador 3D Interactivo	21
2.6	Proyecto 6: Línea de Tiempo Interactiva con Modales	26
2.7	Proyecto 7: Infografía Interactiva con Diseño Bento	32
2.8	Proyecto 8: Instrumento Musical Virtual	36
2.9	Proyecto 9: Aventura Narrativa Interactiva	39
2.10	Proyecto 10: Dashboard de Datos Educativos	43
2.11	Proyecto 11: Guía de Taller Académico (LaTeX)	47
2.12	Proyecto 12: Mapa Mental Interactivo	50
3	Conclusión: El Siguiente Paso	54
A	Anexo J: Referencias Bibliográficas (Perfeccionadas)	55

1 Introducción: El Docente como Creador Digital

Bienvenido al Futuro de la Educación

Este documento es más que un manual: es tu pasaporte a una nueva dimensión pedagógica. Aquí no solo aprenderás sobre Inteligencia Artificial; te convertirás en un **arquitecto de experiencias de aprendizaje** capaz de crear recursos digitales de nivel profesional.

La capacitación tradicional te enseñaba a *usar* herramientas. Esta propuesta te enseña a *crear* tus propias herramientas usando la IA como tu asistente de desarrollo.

1.1 ¿Por Qué Este Documento Es Diferente?

A diferencia de otras capacitaciones que solo muestran ejemplos, este documento te proporciona:

1. **Prompts Probados y Funcionales:** Cada uno ha sido refinado y validado para generar resultados reales.
2. **Aplicación Transversal:** Los 12 proyectos están diseñados para adaptarse a cualquier área del conocimiento.
3. **Autonomía Creativa:** No dependes de plataformas cerradas; todo lo que crees es tuyo.
4. **Alineación Curricular:** Todo está contextualizado con los DBA y lineamientos del MEN Colombia.


1.2 Los 12 Proyectos: Tu Arsenal Pedagógico

Los proyectos están organizados en 4 categorías:

 Simuladores (1-5)

 Visualizaciones (6-8)

 Juegos (9-10)

 Documentos (11-12)

 Consejo Clave

Estrategia de Aprendizaje: No intentes dominar los 12 proyectos en una semana. Elige 2-3 que sean más relevantes para tu área, domínalos, y luego expande tu repertorio. La calidad sobre la cantidad siempre gana.

2 Laboratorio STEM: 12 Proyectos Interactivos con Prompts Maestros

✂ Antes de Comenzar: Prerrequisitos

Para trabajar con estos proyectos necesitas:

- **Acceso a una IA:** ChatGPT, Gemini, o Claude (versión gratuita es suficiente)
- **Editor de texto:** Notepad++, VS Code, o similar
- **Navegador web:** Chrome, Firefox, o Edge
- **(Opcional) Overleaf:** Para proyectos LaTeX

Tiempo estimado por proyecto: 30-45 minutos (primera vez)

2.1 Proyecto 1: Simulador de Procesos Abstractos

Simulador de Procesos Abstractos

🎯 Objetivo Pedagógico

Hacer visible lo invisible. Este proyecto permite crear simulaciones visuales de procesos que son demasiado abstractos, microscópicos, lentos o rápidos para ser observados directamente en el aula. Fomenta la comprensión conceptual a través de la interacción en tiempo real.

Habilidades desarrolladas en el estudiante:

- Pensamiento sistémico (identificar variables y relaciones)
- Modelamiento científico (hipótesis → simulación → análisis)
- Pensamiento computacional (causa-efecto algorítmico)

⚙️ Metodología de Creación

La IA generará una aplicación web con tres componentes clave:

1. **Canvas HTML5:** Actúa como lienzo digital donde se dibuja la animación frame por frame usando JavaScript.
2. **Bucle de Animación:** Usa `requestAnimationFrame()` para actualizar la simulación 60 veces por segundo, creando movimiento fluido.
3. **Panel de Control:** Sliders (Tailwind CSS) que permiten ajustar parámetros en tiempo real (velocidad, cantidad, tamaño).

Stack Técnico:

- HTML5 (estructura)
- Tailwind CSS vía CDN (diseño responsivo)
- JavaScript ES6+ (lógica de simulación)

💡 Aplicaciones por Área Curricular

⚙️ Biología:

- Simulación de mitosis y meiosis con fases diferenciadas por color
- Difusión osmótica (moléculas moviéndose de alta a baja concentración)
- Redes tróficas (cadenas alimenticias animadas)

🧪 Química:

- Equilibrio químico (reactivos \rightleftharpoons productos)
- Ley de gases ideales ($PV = nRT$) con partículas colisionando
- Velocidad de reacción (temperatura vs. colisiones)

🌊 Ciencias Naturales:

- Ciclo del agua (evaporación \rightarrow condensación \rightarrow precipitación)
- Placas tectónicas y movimiento continental
- Fases lunares y eclipses

📈 Matemáticas:

- Visualización de funciones matemáticas en tiempo real
- Teorema de Pitágoras geométrico
- Fractales (conjunto de Mandelbrot simplificado)

</> Proyecto 1: Prompt Maestro

```

1 [MODO] Actúa como un programador experto en desarrollo front-end,
2 especializado en simulaciones educativas interactivas con canvas HTML5.
3
4 [AUDIENCIA] El resultado será usado por un docente de secundaria de
5 [ÁREA: ej. Biología] para explicar [CONCEPTO: ej. la Mitosis Celular]
6 a estudiantes de grado [GRADO: ej. 8°].
7
8 [TAREA] Crea un simulador interactivo de [PROCESO ESPECÍFICO] en un
9 único archivo HTML auto-contenido.
10
11 [TECNOLOGÍAS OBLIGATORIAS]
12 1. Archivo Único: TODO el código (HTML, CSS, JavaScript) debe
13 estar en un solo archivo `.html`.
14 2. Estilos: Usa Tailwind CSS cargado desde CDN. NO uses etiquetas
15 <style> personalizadas.
16 3. Estructura de Interfaz (dos columnas):
17 - Columna Izquierda (30%): Panel de controles con:
18   * Botón "Iniciar/Pausar" (toggle)
19   * Botón "Reiniciar"
20   * Slider de "Velocidad de Simulación" (0.1x a 3x)
21   * [CONTROLES ESPECÍFICOS SEGÚN EL PROCESO]
22 - Columna Derecha (70%): Canvas de 800x600px para la simulación.
23 4. Lógica de Simulación:
24 - Usa JavaScript puro (ES6+)
25 - La lógica debe implementar: [DESCRIBIR REGLAS DEL PROCESO]
26 - Dibuja en el canvas usando el contexto 2D
27 - Usa `requestAnimationFrame()` para animación fluida
28 5. Accesibilidad:
29 - Alto contraste de colores
30 - Fuente legible (mínimo 14px)
31 - Todos los botones accesibles por teclado
32
33 [REGLAS DEL PROCESO A SIMULAR]
34 [AQUÍ EL DOCENTE DESCRIBE LA LÓGICA ESPECÍFICA, POR EJEMPLO:]
35
36 Para una simulación de MITOSIS:
37 - Iniciar con 1 célula madre en el centro
38 - Fase 1 (Profase): El núcleo se hace visible
39 - Fase 2 (Metafase): Cromosomas se alinean en el centro
40 - Fase 3 (Anafase): Cromosomas se separan a los polos
41 - Fase 4 (Telofase): Se forman 2 células hijas
42 - Cada fase debe durar 3 segundos (ajustable con el slider)
43 - Usa colores: verde para citoplasma, azul para núcleo, rojo para
44 cromosomas
45
46 [EJEMPLO DE SALIDA ESPERADA]
47 - El docente debe poder pausar en cualquier fase para explicar
48 - Al reiniciar, todo vuelve a la célula madre
49 - El slider permite ralentizar para análisis detallado

```

```
50  
51 [RESTRICCIONES]  
52 - NO uses librerías externas más allá de Tailwind CSS  
53 - El código debe estar bien comentado en español  
54 - NO incluyas explicaciones antes/después del código, solo el HTML  
55  
56 Comienza el código con `<!DOCTYPE html>`.
```

Consejo Clave

Consejo de Implementación: Después de generar el código, prueba el simulador y haz ajustes iterativos. Prompts comunes de refinamiento:

- “Haz que las células sean más grandes y visibles”
- “Añade etiquetas de texto que indiquen la fase actual”
- “Cambia el color del fondo a un tono más claro”

2.2 Proyecto 2: Juego de Gamificación (Ruleta/Ahorcado/Memoria)

Juego de Gamificación

Objetivo Pedagógico

Transformar la evaluación formativa y el repaso de conceptos en una experiencia lúdica y competitiva. Los juegos de gamificación aumentan la motivación intrínseca del estudiante mediante:

- **Recompensas inmediatas** (puntos, niveles, sonidos de éxito)
- **Reducción de ansiedad** (el error es parte del juego, no una “falla”)
- **Repetición espaciada** (querer jugar de nuevo consolida el aprendizaje)

Metodología de Creación

La IA generará un juego completo con tres subsistemas:

1. **Banco de Preguntas en JSON:** Estructura clara y fácil de editar por el docente:

```
1 {
2   "preguntas": [
3     {
4       "pregunta": "¿Cuál es la capital de Francia?",
5       "opciones": ["Londres", "Paris", "Berlin", "Madrid"],
6       "correcta": 1,
7       "explicacion": "Paris es la capital de Francia desde el siglo X."
8     }
9   ]
10 }
```

2. **Motor de Juego:** Lógica que gestiona el estado (puntuación, vidas, progreso) usando JavaScript.
3. **Sistema de Sonido:** Integración con `Tone.js` para efectos auditivos (acierto, error, victoria).

Variantes de Juego:

- **Ruleta de la Fortuna:** Gira para seleccionar pregunta aleatoria
- **Ahorcado:** Adivina el término científico letra por letra
- **Juego de Memoria:** Encuentra pares de concepto/definición

💡 Aplicaciones por Área Curricular

⚙️ Química:

- Ruleta de nomenclatura química (fórmulas → nombres)
- Ahorcado de elementos de la tabla periódica
- Memoria de pares: reacción ↔ tipo de reacción

📖 Lenguaje:

- Ruleta de figuras literarias con ejemplos
- Ahorcado de vocabulario (palabras del libro de lectura)
- Memoria de sinónimos/antónimos

🌐 Sociales:

- Ruleta de geografía (capitales, banderas, datos)
- Ahorcado de personajes históricos
- Memoria de: evento histórico ↔ año

📐 Matemáticas:

- Ruleta de propiedades ("Una propiedad de los números primos es...")
- Memoria de: fórmula ↔ nombre (ej. $a^2 + b^2 = c^2$ = Pitágoras)

</> Proyecto 2: Prompt Maestro (Ruleta)

```
1 [MODO] Actúa como un desarrollador de juegos HTML5 experto en
2 gamificación educativa.
3
4 [AUDIENCIA] Docente de [ÁREA: ej. Química] que necesita un juego de
5 repaso para estudiantes de grado [GRADO: ej. 10°] sobre [TEMA: ej.
6 Nomenclatura de Compuestos Inorgánicos].
7
8 [TAREA] Crea un juego de "Ruleta de la Suerte" educativa en un único
9 archivo HTML.
10
11 [TECNOLOGÍAS]
12 1. Archivo Único: HTML + CSS + JavaScript.
13 2. Estilos: Tailwind CSS desde CDN. La paleta debe ser vibrante:
14 [COLOR PRIMARIO: ej. Azul eléctrico] con acentos en [COLOR
15 SECUNDARIO: ej. Naranja].
16 3. Estructura de Interfaz:
17 - Arriba: Título del juego y puntaje actual (grande y visible)
18 - Centro: Una "ruleta" (puede ser visual simple o solo un botón
19 "GIRAR")
20 - Abajo: Área de pregunta con 4 botones de opción múltiple
21 - Al final: Botón "Reiniciar Juego"
22
23 [BANCO DE PREGUNTAS]
24 El banco debe estar en un objeto JSON al inicio del script con esta
25 estructura EXACTA:
26
27 const preguntasDB = [
28   {
29     "id": 1,
30     "pregunta": "¿Cómo se llama el compuesto H2O?",
31     "opciones": ["Óxido de Hidrógeno", "Agua", "Dihidrógeno",
32                 "Ácido Hidráulico"],
33     "correcta": 1, // índice (comienza en 0)
34     "explicacion": "H2O es el agua, un compuesto molecular formado
35                   por 2 hidrógenos y 1 oxígeno."
36   },
37   // ... (El docente debe incluir 10-15 preguntas como ejemplo)
38 ];
39
40 [MECÁNICA DEL JUEGO]
41 1. Al hacer clic en "GIRAR", se selecciona una pregunta aleatoria del
42 banco.
43 2. Se muestran las 4 opciones como botones.
44 3. Al seleccionar una opción:
45 - Si es correcta: +10 puntos, sonido de éxito, mostrar explicación
46   con fondo verde
47 - Si es incorrecta: -5 puntos, sonido de error, mostrar explicación
48   con fondo rojo
49 4. Después de 3 segundos, el botón "GIRAR" se reactiva.
```

```
50 5. El juego termina después de 10 preguntas, mostrando puntaje final
51     y mensaje motivacional.
52
53 [SONIDO - MUY IMPORTANTE]
54 - Incluye la librería Tone.js desde CDN.
55 - Sintetiza sonidos simples:
56     * Acierto: Frecuencia 800 Hz, duración 200ms
57     * Error: Frecuencia 200 Hz, duración 400ms
58     * Giro: Frecuencia aleatoria entre 400-600 Hz, duración 100ms
59
60 [RENDERIZADO DE FÓRMULAS]
61 Si el tema incluye fórmulas químicas (como H2O, NaCl), incluye
62 MathJax desde CDN para renderizarlas correctamente. Usa la notación:
63 \(\text{H}_2\text{O}\) para subíndices, \(\text{Fe}^{3+}\) para superíndices.
64
65 [ACCESIBILIDAD]
66 - Contraste alto en los botones
67 - Tamaño de fuente mínimo 16px
68 - Todos los botones accesibles con Tab + Enter
69
70 [RESTRICCIONES]
71 - NO uses imágenes externas, solo emojis o íconos de texto
72 - El código debe ser auto-explicativo con comentarios en español
73 - NO incluyas texto antes/después del código
74
75 Comienza con `<!DOCTYPE html>`.
```

Consejo Clave

Personalización Rápida: Una vez generado el juego, el docente solo necesita editar el array `preguntasDB` en el código para cambiar todo el contenido. No requiere entender JavaScript profundamente.

2.3 Proyecto 3: Calculadora Estándar con Diseño Glassmorphism

Calculadora Estándar

Objetivo Pedagógico

Más allá de la funcionalidad básica (que cualquier calculadora tiene), este proyecto se enfoca en:

- **Diseño UI/UX:** Enseñar principios de interfaz de usuario moderna
- **Estética profesional:** Aplicar tendencias de diseño (Glassmorphism, Neumorphism)
- **Fundamentos de lógica:** Entender evaluación de expresiones matemáticas

Es ideal para cursos de ****Tecnología e Informática**** o como proyecto de introducción a desarrollo web.

Metodología de Creación

La IA creará una calculadora con énfasis en el diseño visual:

1. **Glassmorphism:** Efecto de vidrio esmerilado usando:
 - `backdrop-filter: blur(10px)` para el desenfoque
 - Transparencias con `rgba()` para colores
 - Sombras suaves con `box-shadow`
2. **Grid CSS:** Distribución perfecta de los botones en cuadrícula 4x5
3. **Animaciones:** Transiciones suaves en hover y al hacer clic

Lógica de Cálculo: Usa `eval()` de JavaScript (con validación) para evaluar expresiones matemáticas.

Aplicaciones por Área Curricular

Matemáticas:

- Herramienta básica para operaciones en clase

- Ejemplo didáctico de orden de operaciones (PEMDAS)

Tecnología e Informática:

- Proyecto de introducción a HTML/CSS/JavaScript
- Estudio de caso de diseño UI/UX
- Base para crear calculadoras especializadas (científica, conversión)

Artes:

- Análisis de teoría del color en interfaces
- Principios de diseño visual (balance, contraste, jerarquía)

Proyecto 3: Prompt Maestro

```

1  [MODULO] Actúa como un diseñador UI/UX experto y desarrollador front-end
2  especializado en interfaces modernas.
3
4  [TAREA] Crea una calculadora estándar (solo operaciones básicas: +, -,
5  *, /) con una estética ultra-moderna de tipo Glassmorphism.
6
7  [TECNOLOGÍAS]
8  1. Archivo Único: HTML + CSS + JavaScript en un solo .html.
9  2. Estilos: Tailwind CSS desde CDN + clases CSS personalizadas para
10 el efecto de vidrio.
11 3. Diseño Glassmorphism:
12   - Fondo: Gradiente vibrante (ej. de azul a púrpura)
13   - Calculadora: Contenedor con efecto de vidrio esmerilado
14     * backdrop-filter: blur(16px) saturate(180%)`
15     * background-color: rgba(255, 255, 255, 0.15)`
16     * border: 1px solid rgba(255, 255, 255, 0.3)`
17     * border-radius: 20px`
18     * box-shadow: 0 8px 32px 0 rgba(0, 0, 0, 0.37)`
19
20 [ESTRUCTURA DE LA INTERFAZ]
21 La calculadora debe tener:
22 1. Pantalla de Display (arriba):
23   - Fuente grande (36px) y monoespaciada
24   - Fondo semi-transparente oscuro
25   - Alineación del texto a la derecha
26   - Muestra la operación actual
27 2. Grid de Botones (Grid de 4 columnas):
28   - Fila 1: C (Clear), /, *, -
29   - Fila 2: 7, 8, 9, + (el '+' ocupa 2 filas de alto)
30   - Fila 3: 4, 5, 6
31   - Fila 4: 1, 2, 3, = (el '=' ocupa 2 filas de alto)

```

```
32 - Fila 5: 0 (ocupa 2 columnas de ancho), . (decimal)
33 3. **Estilo de Botones**:
34 - Fondo translúcido blanco con leve opacidad
35 - Borde suave redondeado (border-radius: 10px)
36 - Sombra interna sutil
37 - Transición suave al hacer hover (scale ligeramente)
38 - Al hacer clic, efecto de "presionar" (scale down 0.95)
39
40 [FUNCIONALIDAD]
41 - **Operaciones**: +, -, *, / (usando eval() con validación)
42 - **Clear (C)**: Borra toda la pantalla
43 - **Decimal**: Permite un solo punto decimal por número
44 - **Prevención de errores**:
45   * No permite múltiples operadores seguidos
46   * No permite división por cero (mostrar "Error")
47   * Limita el display a 12 caracteres
48
49 [ANIMACIONES CSS]
50 Todos los botones deben tener:
51 ```css
52 transition: all 0.15s ease-in-out;
53 ```
54 - Hover: `transform: scale(1.05); opacity: 0.9;`
55 - Active: `transform: scale(0.95);`
56
57 [PALETA DE COLORES SUGERIDA]
58 - Fondo: Gradiente de #667eea a #764ba2
59 - Botones numéricos: rgba(255, 255, 255, 0.2)
60 - Botones de operador: rgba(236, 72, 153, 0.3) (rosa/fucsia)
61 - Botón igual: rgba(16, 185, 129, 0.4) (verde)
62 - Botón clear: rgba(239, 68, 68, 0.3) (rojo)
63 - Texto: `color: white`
64
65 [ACCESIBILIDAD]
66 - Contraste suficiente en todos los textos
67 - Botones accesibles por teclado (Tab + Enter)
68 - Mapeo del teclado numérico físico a los botones
69
70 [RESTRICCIONES]
71 - NO uses librerías externas más allá de Tailwind
72 - El código debe ser limpio y estar comentado
73 - NO incluyas texto antes/después del código
74
75 Comienza con `<!DOCTYPE html>`.
```

 **Consejo Clave**

Consejo de Implementación: Cuando generes el código, asegúrate de que el efecto `backdrop-filter` esté en un contenedor que permita que el fondo se vea, logrando el efecto de cristal.

2.4 Proyecto 4: Simulador de Laboratorio de Física

Simulador de Física

🎯 Objetivo Pedagógico

Democratizar el acceso al laboratorio de física. Permite a los estudiantes realizar experimentos que serían:

- **Costosos:** Equipo especializado de miles de dólares
- **Peligrosos:** Altas velocidades, voltajes o temperaturas
- **Imposibles:** Condiciones de vacío, gravedad cero, etc.

El objetivo es la **manipulación de variables** y la **observación inmediata de resultados**, permitiendo el método científico de hipótesis-experimentación-análisis en segundos.

⚙️ Metodología de Creación

Similar al Proyecto 1, pero más específico para física:

1. Panel de Variables de Entrada:

- Sliders para parámetros físicos (velocidad inicial, ángulo, masa, fricción)
- Valores numéricos visibles en tiempo real

2. Canvas de Simulación:

- Animación fluida con `requestAnimationFrame`
- Dibujo de trayectorias, vectores de fuerza, velocidad
- Sistema de coordenadas visible

3. Panel de Resultados:

- Cálculo automático de: alcance máximo, altura máxima, tiempo de vuelo
- Mostrar las ecuaciones utilizadas con MathJax

Ecuaciones Físicas: La IA implementará las fórmulas correctas (cinemática, dinámica, energía) en JavaScript.

💡 Aplicaciones Concretas

👉 Cinemática:

- **Movimiento Parabólico:** Lanzamiento de proyectiles con gravedad
- **MRU y MRUA:** Movimiento rectilíneo uniforme y uniformemente acelerado
- **Caída Libre:** Con y sin resistencia del aire

⌚ Dinámica:

- **Leyes de Newton:** Aplicación de fuerzas y aceleración resultante
- **Plano Inclinado:** Descomposición de fuerzas, fricción
- **Movimiento Circular:** Fuerza centrípeta, velocidad angular

⚡ Electricidad:

- **Ley de Ohm:** Circuito simple con voltaje, resistencia, corriente
- **Ley de Coulomb:** Atracción/repulsión de cargas eléctricas

🔥 Termodinámica:

- **Dilatación Térmica:** Cambio de volumen con temperatura
- **Ley de Gases Ideales:** Relación P-V-T

</> Proyecto 4: Prompt Maestro (Movimiento Parabólico)

```

1 [MODO] Actúa como un físico educador experto en simulaciones
2 computacionales de laboratorio.
3
4 [AUDIENCIA] Docente de Física para estudiantes de grado [GRADO: ej.
5 10°] que estudian [TEMA: ej. Cinemática - Movimiento Parabólico].
6
7 [TAREA] Crea un simulador de laboratorio virtual de Lanzamiento de
8 Proyectiles (movimiento parabólico) en un único archivo HTML.
9
10 [TECNOLOGÍAS]
11 1. Archivo Único: HTML + CSS + JavaScript
12 2. Estilos: Tailwind CSS desde CDN
13 3. Matemáticas: MathJax desde CDN para renderizar ecuaciones
14
15 [ESTRUCTURA DE LA INTERFAZ] (3 columnas)
16
17 Columna Izquierda (25%): Panel de Control
18 - Título: "Variables de Entrada"
19 - Velocidad Inicial (v): Slider de 5 a 50 m/s (paso: 1)
20   * Mostrar valor actual: "v = 25 m/s"
21 - Ángulo de Lanzamiento ( ): Slider de 0° a 90° (paso: 5°)
22   * Mostrar valor: " = 45°"
23 - Gravedad (g): Selector desplegable:
24   * Tierra (9.8 m/s2) [por defecto]
25   * Luna (1.62 m/s2)
26   * Marte (3.71 m/s2)
27 - Botones:
28   * "Lanzar" (verde)
29   * "Reiniciar" (rojo)
30   * "Modo Trazado" (checkbox para mostrar trayectoria completa)
31
32 Columna Central (50%): Canvas de Simulación
33 - Canvas de 600x400px
34 - Fondo: cielo azul claro arriba (70%), tierra verde abajo (30%)
35 - Sistema de coordenadas visible (ejes X e Y con marcas cada 10m)
36 - Proyectil: círculo rojo de 10px de radio
37 - Vectores dibujados en tiempo real:
38   * Velocidad (flecha azul proporcional a |v|)
39   * Velocidad horizontal v (flecha verde)
40   * Velocidad vertical v (flecha naranja)
41
42 Columna Derecha (25%): Panel de Resultados
43 - Título: "Datos Calculados"
44 - Mostrar en tiempo real:
45   * Altura actual (y): "h = 12.5 m"
46   * Alcance actual (x): "x = 35.2 m"
47   * Velocidad total (v): "v = 22.3 m/s"
48 - Al finalizar el vuelo, mostrar:
49   * Altura Máxima: "H_max = 15.9 m"

```

```

50 * **Alcance Horizontal**: "R = 63.8 m"
51 * **Tiempo de Vuelo**: "t = 4.5 s"
52 - Sección "Ecuaciones Usadas" (con MathJax):
53 \[ x(t) = v_0 \cos(\theta) \cdot t \]
54 \[ y(t) = v_0 \sin(\theta) \cdot t - \frac{1}{2}gt^2 \]
55 \[ H_{\max} = \frac{v_0^2 \sin^2(\theta)}{2g} \]
56 \[ R = \frac{v_0^2 \sin(2\theta)}{g} \]
57
58 [LÓGICA DE SIMULACIÓN]
59 1. Al hacer clic en "Lanzar":
60   - Calcular componentes de velocidad:
61     * v = v * cos()
62     * v = v * sin()
63   - Iniciar bucle de animación con `requestAnimationFrame`
64 2. En cada frame (60 fps):
65   - Incrementar tiempo: t += deltaTime (ej. 0.016s)
66   - Calcular posición:
67     * x(t) = v * t
68     * y(t) = v * t - 0.5 * g * t2
69   - Actualizar velocidad:
70     * v permanece constante
71     * v = v - g * t
72   - Dibujar proyectil en (x, y)
73   - Dibujar vectores de velocidad
74   - Si "Modo Trazado" está activo, dejar rastro de la trayectoria
75 3. Detener cuando y = 0 (toca el suelo)
76 4. Calcular y mostrar estadísticas finales
77
78 [FÍSICA CORRECTA - MUY IMPORTANTE]
79 - Usar g en m/s2 según la selección
80 - Convertir ángulo de grados a radianes: _rad = * / 180
81 - Escalar el canvas: 1 unidad de canvas = 1 metro (ajustar si es
82 necesario)
83 - Los vectores deben ser proporcionales: longitud = magnitud * escala
84
85 [CARACTERÍSTICAS ADICIONALES]
86 - **Zoom Automático**: Si el proyectil sale del canvas, ajustar escala
87 - **Sonido de Lanzamiento**: Usar Tone.js para un "whoosh" al lanzar
88 - **Colisión**: Al tocar el suelo, mostrar pequeña "explosión" visual
89
90 [ACCESIBILIDAD]
91 - Todos los sliders accesibles por teclado (flechas para ajustar)
92 - Alto contraste en textos
93 - Fuente mínima 14px
94
95 [RESTRICCIONES]
96 - La física debe ser precisa (validar con casos conocidos: 45° da
97 máximo alcance)
98 - El código debe incluir comentarios explicando las ecuaciones
99 - NO incluyas texto antes/después del código

```

100
101

```
Comienza con `<!DOCTYPE html>`.
```

Consejo Clave

Validación Pedagógica: Después de generar el simulador, prueba casos límite:

- Ángulo 0° : Movimiento horizontal puro
- Ángulo 90° : Movimiento vertical (caída libre simétrica)
- Ángulo 45° : Debe dar el máximo alcance horizontal

Si los resultados no coinciden con la teoría, refina el prompt especificando las ecuaciones exactas.

2.5 Proyecto 5: Visualizador 3D Interactivo

Visualizador 3D

🎯 Objetivo Pedagógico

Desarrollar la **inteligencia espacial** y permitir la exploración tridimensional de conceptos que son inherentemente 3D. Es fundamental para:

- **Geometría:** Sólidos platónicos, poliedros, intersecciones
- **Astronomía:** Sistema solar, fases lunares, constelaciones
- **Biología Molecular:** Estructura de proteínas, ADN, células
- **Cristalografía:** Estructuras cristalinas, redes de Bravais

Una representación 2D de estos conceptos es *insuficiente*; la interactividad 3D permite rotar, hacer zoom y explorar desde cualquier ángulo.

⚙️ Metodología de Creación

Este es un prompt **avanzado** que requiere Three.js:

1. **Three.js desde CDN:** La librería estándar de la industria para 3D en la web
2. **Escena 3D:**
 - Scene: Contenedor de todos los objetos 3D
 - Camera: Perspectiva del observador
 - Renderer: Motor que dibuja en el canvas
3. **Controles de Órbita:**
 - Rotar: Arrastrar con el mouse
 - Zoom: Rueda del mouse o pinch en móvil
 - Paneo: Click derecho + arrastrar
4. **Geometrías y Materiales:**
 - Three.js tiene geometrías predefinidas (esfera, cubo, tetraedro)
 - Materiales con iluminación realista (Phong, Lambert)

💡 Aplicaciones por Área

🏠 Geometría:

- **Sólidos Platónicos:** Los 5 poliedros regulares (tetraedro, cubo, octaedro, dodecaedro, icosaedro)
- **Prismas y Pirámides:** Variación de bases (triangular, pentagonal, hexagonal)
- **Secciones Cónicas:** Visualización 3D de parábola, elipse, hipérbola

🌙 Astronomía:

- **Sistema Solar a Escala:** Planetas con órbitas, tamaños relativos
- **Comparación de Modelos:** Geocéntrico vs. Heliocéntrico (animado)
- **Fases Lunares:** Tierra-Luna-Sol con iluminación dinámica

🧬 Biología:

- **Doble Hélice de ADN:** Visualización de la estructura con nucleótidos
- **Estructuras de Proteínas:** Alfa-hélice, hoja beta plegada
- **Anatomía 3D:** Corazón, cerebro, sistema esquelético

💎 Química/Cristalografía:

- **Redes Cristalinas:** Cúbica simple, BCC, FCC, hexagonal
- **Moléculas Orgánicas:** Metano, etanol, glucosa (modelo ball-and-stick)

</> Proyecto 5: Prompt Maestro (Sólidos Platónicos)

```

1 [MODO] Actúa como un programador experto en gráficos 3D por computadora
2 usando Three.js.
3
4 [AUDIENCIA] Docente de [ÁREA: ej. Matemáticas - Geometría] para
5 estudiantes de grado [GRADO: ej. 9°] estudiando [TEMA: ej. Poliedros
6 Regulares].
7
8 [TAREA] Crea un visualizador 3D interactivo de los 5 Sólidos
9 Platónicos en un único archivo HTML.
10
11 [TECNOLOGÍAS]
12 1. Archivo Único: HTML + CSS + JavaScript
13 2. Estilos: Tailwind CSS desde CDN
14 3. Librería 3D: Three.js cargado desde CDN de jsDelivr o UNPKG:
15   https://cdn.jsdelivr.net/npm/three@0.150.0/build/three.min.js
16 4. IMPORTANTE: Para controles, usa OrbitControls. Importa desde:
17   import { OrbitControls } from
18     ↪ 'https://cdn.jsdelivr.net/npm/three@0.150.0/examples/jsm/controls/OrbitControls.js';
19   Esto requiere que tu script principal sea `
```

```

49 [FUNCIONALIDAD 3D]
50
51 **Geometrías de Three.js**:
52 - Tetraedro: `TetrahedronGeometry(1, 0)`
53 - Cubo: `BoxGeometry(1.5, 1.5, 1.5)`
54 - Octaedro: `OctahedronGeometry(1, 0)`
55 - Dodecaedro: `DodecahedronGeometry(1, 0)`
56 - Icosaedro: `IcosahedronGeometry(1, 0)`
57
58 **Material**:
59 - Usar `MeshPhongMaterial` para iluminación realista
60 - Color: Vibrante (ej. cian, magenta, naranja según el sólido)
61 - Wireframe: Opcional, checkbox para activar/desactivar
62
63 **Animación**:
64 - El sólido debe **rotar automáticamente** en el eje Y:
65   `mesh.rotation.y += 0.005;` // En el bucle de animación
66 - Velocidad de rotación ajustable con un slider (0x a 3x)
67
68 **Controles del Usuario** (OrbitControls):
69 - **Rotar**: Arrastrar con mouse (botón izquierdo)
70 - **Zoom**: Rueda del mouse
71 - **Panear**: Arrastrar con mouse (botón derecho)
72 - Habilitar `autoRotate` en los controles y `enableDamping`
73
74 **Transición entre Sólidos**:
75 - Al hacer clic en otro botón, hacer fade-out del sólido actual,
76   cambiar geometría, fade-in del nuevo
77
78 [DATOS DE LOS SÓLIDOS] (para el panel de información)
79 | Sólido      | Caras (F) | Vértices (V) | Aristas (E) | V-E+F |
80 |-----|-----|-----|-----|-----|
81 | Tetraedro  | 4         | 4           | 6           | 2      |
82 | Cubo       | 6         | 8           | 12          | 2      |
83 | Octaedro   | 8         | 6           | 12          | 2      |
84 | Dodecaedro| 12        | 20          | 30          | 2      |
85 | Icosaedro  | 20        | 12          | 30          | 2      |
86
87 [OPTIMIZACIÓN]
88 - El bucle de animación debe usar `requestAnimationFrame`
89 - Actualizar solo lo necesario (no recrear toda la escena cada frame)
90 - Liberar memoria al cambiar de sólido: `scene.remove(mesh);`
91   `mesh.geometry.dispose();`
92
93 [ACCESIBILIDAD]
94 - Botones accesibles por teclado (Tab + Enter)
95 - Indicador visual claro del sólido activo
96 - Instrucciones de uso visibles: "Arrastra para rotar, rueda del mouse
97   para zoom"
98

```

```
99 [RESTRICCIONES]
100 - El `<script>` principal DEBE ser `type="module"` para importar
101   OrbitControls.
102 - El código debe incluir comentarios explicando la configuración de
103   Three.js
104
105 Comienza con `<!DOCTYPE html>`.
```

Consejo Clave

Error Común (script type="module"): Three.js moderno usa módulos ES6. Si el código no funciona, asegúrate de que el script principal sea '`<script type="module">`' y que OrbitControls se importe correctamente ('import OrbitControls ...'). Si la IA falla, pídele: "Corrige el código para usar Three.js y OrbitControls como un módulo ES6 (`type="module"`)."

2.6 Proyecto 6: Línea de Tiempo Interactiva con Modales

Línea de Tiempo Interactiva

Objetivo Pedagógico

Visualizar la **cronología de eventos** de manera no lineal y exploratoria. Las líneas de tiempo tradicionales (en papel) son estáticas y limitadas. Una versión digital permite:

- **Exploración activa:** Hacer clic en eventos para más detalles
- **Contexto visual:** Imágenes, colores por época/tipo
- **Escala flexible:** Zoom temporal para analizar períodos específicos
- **Conexiones:** Líneas que conectan eventos relacionados

Metodología de Creación

La IA generará un juego completo con tres subsistemas:

1. Estructura de Datos JSON:

```
1  [
2    {
3      "fecha": "4600 millones de años atrás",
4      "titulo": "Formación de la Tierra",
5      "descripcion_corta": "Acreción del disco protoplanetario",
6      "descripcion_larga": "La Tierra se formó...",
7      "imagen_url": "https://placeholder.co/400x200/8B4513/white?text=Precámbrico",
8      "categoria": "Precámbrico"
9    }
10 ]
```

2. Layout Horizontal con Flexbox/Grid: Los eventos se disponen en una "pista" horizontal con scroll

3. Modal (Ventana Emergente): Al hacer clic en un evento, se abre un modal con:

- Título y fecha grande
- Imagen (si está disponible)
- Descripción detallada
- Botón de cerrar (X)

💡 Aplicaciones por Área

🌐 Historia:

- **Historia Universal:** De la Prehistoria al siglo XXI
- **Historia de Colombia:** Independencia → La Violencia → Constitución del 91
- **Guerras Mundiales:** Eventos clave de 1914-1945

⚠️ Geología:

- **Eras Geológicas:** Precámbrico → Paleozoico → Mesozoico → Cenozoico
- **Evolución de la Vida:** Primeras células → Dinosaurios → Mamíferos → Humanos

📖 Literatura:

- **Movimientos Literarios:** Clasicismo → Romanticismo → Realismo → Modernismo
- **Vida de un Autor:** Biografía cronológica de García Márquez, Cervantes
- **Cronología de una Novela:** Eventos de "Cien Años de Soledad" ordenados

💻 Tecnología:

- **Historia de la Computación:** Ábaco → ENIAC → Internet → IA
- **Evolución de los Lenguajes:** FORTRAN → C → Python → Rust

🎨 Artes:

- **Historia del Arte:** Arte Rupestre → Renacimiento → Impresionismo → Arte Digital

</> Proyecto 6: Prompt Maestro (Eras Geológicas)

```
1 [MODO] Actúa como un desarrollador front-end experto en visualización
2 de datos históricos y cronológicos.
3
4 [AUDIENCIA] Docente de [ÁREA: ej. Ciencias Naturales - Geología] para
5 estudiantes de grado [GRADO: ej. 7º] estudiando [TEMA: ej. Eras
6 Geológicas y Evolución de la Vida].
7
8 [TAREA] Crea una Línea de Tiempo Interactiva Horizontal sobre las
9 Eras Geológicas de la Tierra en un único archivo HTML.
10
11 [TECNOLOGÍAS]
12 1. Archivo Único: HTML + CSS + JavaScript
13 2. Estilos: Tailwind CSS desde CDN
14 3. NO uses librerías de gráficos (como Chart.js); todo debe ser
15 HTML/CSS/JS puro
16
17 [ESTRUCTURA DE LA INTERFAZ]
18
19 Encabezado (Top):
20 - Título: "Línea de Tiempo: Eras Geológicas de la Tierra"
21 - Subtítulo: "4600 millones de años de historia"
22 - Instrucciones: "Haz clic en cada era para ver más detalles"
23
24 Línea de Tiempo (Centro):
25 - Layout: Horizontal con scroll lateral (si es necesario)
26 - Estructura visual:
27   * Una "pista" o "línea" horizontal (barra gruesa de 10px de alto,
28     color gris)
29   * Sobre la línea, "hitos" o "marcadores" (círculos de 20px de
30     diámetro)
31   * Cada marcador tiene un color según la era (ver datos abajo)
32   * Debajo de cada marcador: el nombre de la era y la fecha
33 - Espaciado: Proporcional al tiempo transcurrido (ej. Precámbrico
34   ocupa 70% del ancho)
35
36 Modal (Ventana Emergente):
37 Al hacer clic en un marcador, se abre un modal centrado en la pantalla:
38 - Fondo oscuro semi-transparente (rgba(0,0,0,0.7)) que cubre toda la
39   pantalla
40 - Contenedor del modal: Caja blanca redondeada, centrada, ancho máx
41   600px
42 - Contenido:
43   * Título grande (nombre de la era)
44   * Fecha/Duración (ej. "4600 - 541 millones de años atrás")
45   * Imagen representativa (si está disponible en el JSON, sino usar un
46     placeholder)
47   * Descripción detallada (3-4 párrafos)
48   * Eventos clave (lista con viñetas)
49   * Botón "Cerrar" (X) en la esquina superior derecha
```

```
50 - Cerrar modal: Al hacer clic en la X o fuera del modal
51
52 [DATOS DE LAS ERAS GEOLÓGICAS] (Estructura JSON)
53
54 const erasGeologicas = [
55   {
56     "id": 1,
57     "nombre": "Precámbrico",
58     "inicio": "4600 millones de años",
59     "fin": "541 millones de años",
60     "duracion": "4059 millones de años",
61     "color": "#8B4513", // Marrón
62     "descripcion": "El Precámbrico abarca aproximadamente el 88% de la
63                   historia de la Tierra...",
64     "eventos_clave": [
65       "Formación de la Tierra (4600 Ma)",
66       "Primeras formas de vida (3800 Ma)",
67       "Gran Oxidación (2400 Ma)",
68       "Primeros organismos multicelulares (600 Ma)"
69     ],
70     "imagen_url": "https://placeholder.co/600x300/8B4513/FFFFFF?text=Precámbrico"
71   },
72   {
73     "id": 2,
74     "nombre": "Paleozoico",
75     "inicio": "541 millones de años",
76     "fin": "252 millones de años",
77     "duracion": "289 millones de años",
78     "color": "#4682B4", // Azul acero
79     "descripcion": "La era Paleozoica vio la explosión de vida
80                   compleja...",
81     "eventos_clave": [
82       "Explosión Cámbrica (541 Ma)",
83       "Primeras plantas terrestres (470 Ma)",
84       "Primeros anfibios (365 Ma)",
85       "Extinción masiva Pérmico-Triásico (252 Ma)"
86     ],
87     "imagen_url": "https://placeholder.co/600x300/4682B4/FFFFFF?text=Paleozoico"
88   },
89   {
90     "id": 3,
91     "nombre": "Mesozoico",
92     "inicio": "252 millones de años",
93     "fin": "66 millones de años",
94     "duracion": "186 millones de años",
95     "color": "#228B22", // Verde bosque
96     "descripcion": "Conocida como la Era de los Dinosaurios...",
97     "eventos_clave": [
98       "Aparición de los dinosaurios (230 Ma)",
99       "Primeras aves (150 Ma - Archaeopteryx)",
```

```

100     "Primeras plantas con flores (140 Ma)",
101     "Extinción K-T - Fin de los dinosaurios (66 Ma)"
102   ],
103   "imagen_url": "https://placeholder.co/600x300/228B22/FFFFFF?text=Mesozoico"
104 },
105 {
106   "id": 4,
107   "nombre": "Cenozoico",
108   "inicio": "66 millones de años",
109   "fin": "Presente",
110   "duracion": "66 millones de años",
111   "color": "#FFD700", // Dorado
112   "descripcion": "La Era de los Mamíferos...",
113   "eventos_clave": [
114     "Diversificación de mamíferos (66-55 Ma)",
115     "Primeros primates (55 Ma)",
116     "Aparición del Homo sapiens (300,000 años)",
117     "Era del Hielo (2.6 Ma - 11,700 años)"
118   ],
119   "imagen_url": "https://placeholder.co/600x300/FFD700/000000?text=Cenozoico"
120 }
121 ];
122
123 [CÁLCULO DE ESPACIADO PROPORCIONAL]
124 - Total de tiempo: 4600 millones de años
125 - Precámbrico (4059 Ma): 88.2% del ancho
126 - Paleozoico (289 Ma): 6.3% del ancho
127 - Mesozoico (186 Ma): 4% del ancho
128 - Cenozoico (66 Ma): 1.4% del ancho
129
130 **Implementación sugerida**:
131 - Contenedor de la línea: `display: flex`
132 - Cada era: `flex-basis: X%` (según el cálculo)
133
134 [INTERACTIVIDAD]
135 1. **Hover sobre marcador**: Efecto de escala (scale 1.2) y mostrar
136   tooltip con nombre
137 2. **Click en marcador**: Abrir modal con animación fade-in
138 3. **Cerrar modal**: Fade-out y ocultar
139 4. **Navegación por teclado**: Flechas izq/der para moverse entre eras
140
141 [DISEÑO RESPONSIVO]
142 - En pantallas pequeñas (<768px):
143   * La línea debe tener scroll horizontal
144   * El modal ocupa 95% del ancho de pantalla
145   * Los marcadores se ajustan en tamaño
146
147 [ACCESIBILIDAD]
148 - Cada marcador debe tener `role="button"` y `tabindex="0"`
149 - El modal debe capturar el foco (trap focus)

```

```
150 - Alto contraste en textos sobre fondos
151 - Descripción alternativa para imágenes (alt text)
152
153 [RESTRICCIONES]
154 - NO uses Chart.js, D3.js u otra librería de gráficos
155 - Todo debe ser HTML/CSS/JS puro con Tailwind
156 - El código debe estar limpio y comentado
157
158 Comienza con `<!DOCTYPE html>`.
```

Consejo Clave

Consejo de Implementación: Cuando generes el código, recuerda que el espaciado proporcional requiere un contenedor de línea muy amplio para que quepa todo (ej. 'w-[4600px]' o un valor similar escalado) para permitir el **scroll** horizontal.

2.7 Proyecto 7: Infografía Interactiva con Diseño Bento

Infografía Bento UI

🎯 Objetivo Pedagógico

Desarrollar la habilidad de **sintetizar y comunicar información compleja** de forma visual, clara y atractiva. Las infografías son:

- **Cognitivamente eficientes:** El cerebro procesa imágenes 60,000 veces más rápido que texto
- **Memorables:** La retención aumenta un 65% cuando se combina texto + visual
- **Compartibles:** Formato ideal para presentaciones y redes sociales educativas

Es un ejercicio de **diseño de información** (Information Design), fundamental en el siglo XXI.

⚙️ Metodología de Creación

El diseño ****Bento UI**** (inspirado en las cajas de comida japonesa Bento) organiza información en una cuadrícula modular:

1. **CSS Grid Layout:** La base técnica
 - Definir un grid de 12 columnas x N filas
 - Cada "widget" ocupa un espacio específico (ej. 3 columnas x 2 filas)
2. **Tipos de Widgets:**
 - **Header:** Título principal + subtítulo
 - **Stat Card:** Un dato grande + descripción (ej. "8 Planetas")
 - **Text Block:** Definición o explicación corta
 - **Icon + Text:** Ícono grande + texto explicativo
 - **Image Block:** Imagen con caption
 - **List:** Lista de items con checkmarks o números
3. **Paleta de Colores Cohesiva:** Máximo 3 colores + neutros
4. **Jerarquía Visual:** Tamaños de fuente, peso y contraste para guiar el ojo.

💡 Aplicaciones por Área

🌿 Ciencias Naturales:

- "El Ciclo del Carbono": Widgets mostrando fotosíntesis, respiración, combustión
- "La Célula Eucariota": Cada organelo en su widget con función
- "Biomás del Mundo": Grid con tundra, bosque, desierto, etc.

🌍 Sociales:

- "Culturas Indígenas de América": Mayas, Aztecas, Incas, Muisca
- "La Revolución Industrial": Inventos clave, cambios sociales, consecuencias
- "Democracia vs. Dictadura": Comparación visual de características

📐 Matemáticas:

- "Números Importantes": π , e , ϕ , cada uno con su historia y aplicación
- "Tipos de Triángulos": Clasificación por lados y ángulos con diagramas

💻 Tecnología:

- "Partes del Computador": CPU, RAM, Disco Duro, etc. con especificaciones
- "Internet: ¿Cómo Funciona?": Capas del modelo OSI simplificado

📖 Lenguaje:

- "Figuras Literarias": Metáfora, hipérbole, símil con ejemplos literarios
- "Conjugación Verbal": Tiempos del verbo "amar" en grid

</> Proyecto 7: Prompt Maestro (Civilizaciones Antiguas)

```
1 [MODULO] Actúa como un diseñador gráfico profesional especializado en
2 infografías educativas y diseño de información.
3
4 [AUDIENCIA] Docente de [ÁREA: ej. Ciencias Sociales - Historia] para
5 estudiantes de grado [GRADO: ej. 6°] estudiando [TEMA: ej. Principales
6 Civilizaciones Antiguas].
7
8 [TAREA] Crea una infografía interactiva de una sola página con diseño
9 **Bento UI** sobre las "4 Grandes Civilizaciones Antiguas: Egipto,
10 Mesopotamia, India y China" en un único archivo HTML.
11
12 [TECNOLOGÍAS]
13 1. **Archivo Único**: HTML + CSS + JavaScript (mínimo)
14 2. **Estilos**: Tailwind CSS desde CDN
15 3. **Íconos**: Usa SVG inline desde Heroicons (ej. `**Header** (col-span-12): Título + Subtítulo
23 2. **Egipto** (col-span-6 row-span-2): Descripción + datos
24 3. **Mesopotamia** (col-span-6 row-span-2): Descripción + datos
25 4. **India** (col-span-6 row-span-2): Descripción + datos
26 5. **China** (col-span-6 row-span-2): Descripción + datos
27 6. **Línea de Tiempo** (col-span-12): Visual simple
28 7. **Logros Comunes** (col-span-12): Lista
29
30 [CONTENIDO DE CADA CIVILIZACIÓN]
31 - **Nombre** (Título)
32 - **Período** (Subtítulo)
33 - **Ubicación** (Icono de mapa + texto)
34 - **Logros** (3 items con checkmarks):
35   * Egipto: Pirámides, Jeroglíficos, Momificación
36   * Mesopotamia: Rueda, Escritura Cuneiforme, Código Hammurabi
37   * India: Sistema Decimal, Yoga, Ayurveda
38   * China: Papel, Pólvora, Gran Muralla
39 - **Dato Curioso** (caja resaltada)
40
41 [PALETA DE COLORES TEMÁTICA]
42 - **Egipto**: Dorado (#FFD700) + Arena (#F4E5C2)
43 - **Mesopotamia**: Azul Río (#4682B4) + Terracota (#E27B58)
44 - **India**: Azafrán (#FF9933) + Verde (#138808)
45 - **China**: Rojo Imperial (#DC143C) + Jade (#00A86B)
46 - **Fondo General**: `bg-gray-100`
47 - **Texto**: `text-gray-800`
48
49 [DISEÑO DE WIDGETS]
```

```
50 - Estilo base: `bg-white rounded-2xl p-6 shadow-lg`
51 - Borde superior grueso del color temático
52 - Hover: `hover:shadow-xl transition-all duration-300`
53
54 [TIPOGRAFÍA]
55 - **Títulos**: Fuente serif (ej. Playfair Display desde Google Fonts)
56 - **Cuerpo**: Sans-serif (ej. Inter o Noto Sans)
57
58 [ACCESIBILIDAD]
59 - **Contraste WCAG AA**
60 - **Texto Alternativo** para íconos
61 - **Responsivo**: En pantallas <768px, el grid se convierte en `grid-cols-1`
62
63 [DATOS A INCLUIR] (Ejemplo para Egipto)
64 **Egipto Antiguo**
65 - Período: 3100 a.C. - 30 a.C.
66 - Ubicación: Valle del Río Nilo
67 - Logros: Pirámides, Jeroglíficos, Momificación
68 - Dato Curioso: Inventaron el calendario solar de 365 días.
69
70 [RESTRICCIONES]
71 - NO uses librerías de gráficos
72 - Los íconos deben ser SVG inline
73 - El diseño debe ser limpio y moderno
74 - Código comentado
75
76 Comienza con `<!DOCTYPE html>`.
```

Consejo Clave

Refinamiento Visual: Una vez generada la infografía, pide ajustes específicos:

- “Aumenta el espacio entre widgets a ‘gap-6’ para más aire visual”
- “Cambia la fuente del título a una más elegante como Playfair Display”
- “Añade sombras más pronunciadas para mayor profundidad”

2.8 Proyecto 8: Instrumento Musical Virtual

🎵 Instrumento Musical Virtual

🎯 Objetivo Pedagógico

Fomentar la creatividad y conectar la tecnología con las artes (el pilar "A" de STEAM). Permite a los estudiantes:

- **Explorar la teoría musical:** Escalas, acordes, notas.
- **Entender la física del sonido:** Frecuencia, tono, duración.
- **Componer digitalmente:** Crear melodías sin un instrumento físico.

⚙️ Metodología de Creación

La IA generará una aplicación que **sintetiza sonido** en tiempo real:

1. **Tone.js:** Es la librería clave (cargada desde CDN). Prohíbe explícitamente el uso de archivos MP3/WAV.
2. **Sintetizador:** Se configura un `Tone.PolySynth()` que permite tocar múltiples notas a la vez.
3. **Doble Interactividad:**
 - **Clic/Táctil:** Las teclas del piano en pantalla son clickeables.
 - **Teclado Físico:** Se mapean las teclas del computador (ej. 'A' es Do, 'W' es Do#, 'S' es Re) a las notas.

💡 Aplicaciones por Área

🎵 Música:

- Piano virtual para practicar escalas y acordes.
- Herramienta de composición y experimentación melódica.

🔬 Física:

- Demostración de frecuencias (ej. La nota A4 es 440 Hz).

- Visualización de ondas sonoras (si se combina con un canvas).

Tecnología:

- Proyecto de manejo de eventos (mousedown, keydown) y librerías externas.

Proyecto 8: Prompt Maestro (Piano Virtual)

```

1  [MODULO] Actúa como un ingeniero de audio digital y desarrollador front-end.
2
3  [TAREA] Crea un Piano Virtual interactivo de una octava en un
4  único archivo HTML.
5
6  [TECNOLOGÍAS OBLIGATORIAS]
7  1. Archivo Único: HTML + CSS + JavaScript.
8  2. Estilos: Tailwind CSS desde CDN.
9  3. Sonido: Tone.js cargado desde CDN.
10     https://cdnjs.cloudflare.com/ajax/libs/tone/14.8.49/Tone.js
11  4. NO USAR ARCHIVOS MP3/WAV: Todo el sonido debe ser sintetizado
12     en tiempo real con Tone.js.
13
14  [ESTRUCTURA DE LA INTERFAZ]
15  1. Título: "Piano Virtual Interactivo"
16  2. Piano:
17     - 8 Teclas Blancas (Do, Re, Mi, Fa, Sol, La, Si, Do5)
18     - 5 Teclas Negras (Do\#, Re\#, Fa\#, Sol\#, La\#)
19     - Las teclas negras deben superponerse visualmente a las blancas
20       usando CSS (absolute positioning).
21  3. Controles:
22     - Un selector de tipo de onda (sine, square, triangle, sawtooth).
23  4. Instrucciones: "Toca con el mouse o usa tu teclado: A, S, D, F,
24     G, H, J (blancas) y W, E, T, Y, U (negras)"
25
26  [LÓGICA DE SONIDO (Tone.js)]
27  1. Inicialización: El audio debe iniciarse por una acción del
28     usuario (ej. un botón "Activar Audio") debido a las políticas del
29     navegador. `await Tone.start()`
30  2. Sintetizador: Usa `const synth = new
31     ↪ Tone.PolySynth(Tone.Synth).toDestination();`
32     para permitir múltiples notas.
33  3. Función de Tocar: `synth.triggerAttackRelease(nota, duracion);`
34     - Ej: `synth.triggerAttackRelease("C4", "8n");`
35
36  [MAPEO DE TECLADO Y CLICS]
37  - Cada tecla HTML (blanca y negra) debe tener un evento `mousedown`
38     que llame a la función de tocar (ej. `tocarNota('C4')`).
39  - Añadir un listener global `keydown` para el teclado físico:
40     - 'A' -> 'C4'
41     - 'W' -> 'C#4'

```

```
41 - 'S' -> 'D4'  
42 - 'E' -> 'D#4'  
43 - 'D' -> 'E4'  
44 - 'F' -> 'F4'  
45 - 'T' -> 'F#4'  
46 - 'G' -> 'G4'  
47 - 'Y' -> 'G#4'  
48 - 'H' -> 'A4'  
49 - 'U' -> 'A#4'  
50 - 'J' -> 'B4'  
51 - 'K' -> 'C5'  
52  
53 [ESTILO VISUAL]  
54 - Teclas blancas: `bg-white border-2 border-black h-48 w-16`  
55 - Teclas negras: `bg-black h-32 w-10 z-10 -ml-5`  
56 - Efecto "presionado": `active:bg-gray-300`  
57  
58 [ACCESIBILIDAD]  
59 - Botones accesibles por Tab.  
60 - Roles ARIA para las teclas.  
61  
62 [RESTRICCIONES]  
63 - Código limpio y comentado en español.  
64 - NO incluyas explicaciones antes/después del código.  
65  
66 Comienza con `<!DOCTYPE html>`.
```

2.9 Proyecto 9: Aventura Narrativa Interactiva

Aventura Narrativa

Objetivo Pedagógico

Enseñar habilidades del siglo XXI: **pensamiento crítico, toma de decisiones y comprensión de las consecuencias**. El estudiante se convierte en el protagonista de una historia y sus elecciones determinan el resultado.

- Fomenta la empatía al explorar diferentes puntos de vista.
- Desarrolla la comprensión lectora de una manera activa, no pasiva.
- Ideal para temas de ética, civismo, historia y literatura.

Metodología de Creación

La IA creará un motor de juego narrativo basado en un grafo:

1. **Historia como JSON:** La narrativa se define como un objeto JSON donde cada "nodo" tiene un ID, un texto y opciones que apuntan a otros IDs.

```

1  const historia = {
2    "inicio": {
3      "texto": "Te despiertas en un bosque oscuro. Ves un sendero a la izquierda y una cueva a la derecha.",
4      "opciones": [
5        { "texto": "Tomar el sendero", "destino": "sendero" },
6        { "texto": "Explorar la cueva", "destino": "cueva" }
7      ]
8    },
9    "sendero": { ... },
10   "cueva": { ... }
11 }

```

2. **Motor de JS:** Una función que lee el nodo actual, muestra el texto y genera los botones de opciones.
3. **Efecto "Máquina de Escribir":** Para mejorar la inmersión, el texto narrativo aparece letra por letra.

Aplicaciones por Área

Ética y Civismo:

- Dilemas de ciberseguridad (¿Qué hacer ante el ciberacoso?).

- Simulaciones de ciudadanía digital y uso responsable de IA.

Historia:

- "Vive la Revolución Francesa": Decisiones como ciudadano, noble o guardia.
- "La Ruta de Colón": Decisiones de navegación y gestión de la tripulación.

Literatura:

- Crear finales alternativos para novelas estudiadas en clase.
- Adaptar un cuento corto (ej. de Poe) a formato interactivo.

Ciencias:

- "Aventura Ecológica": Decisiones que afectan un ecosistema (contaminación, reforestación).

Proyecto 9: Prompt Maestro (Aventura Narrativa)

```

1 [MODO] Actúa como un diseñador narrativo experto y desarrollador de
2 juegos interactivos (estilo "Elige tu propia aventura").
3
4 [TAREA] Crea un motor de juego de aventura narrativa en un único
5 archivo HTML sobre [TEMA: ej. un dilema de ciberseguridad para
6 adolescentes].
7
8 [TECNOLOGÍAS]
9 1. Archivo Único: HTML + CSS + JavaScript.
10 2. Estilos: Tailwind CSS desde CDN.
11 3. Diseño: Estilo "terminal" o "libro antiguo", con una fuente
12 monoespaciada (ej. `font-mono`) o serif (ej. `font-serif`). Fondo
13 oscuro (`bg-gray-900`) y texto claro (`text-green-400` o
14 `text-gray-200`).
15
16 [ESTRUCTURA DE LA HISTORIA (JSON)]
17 La historia DEBE estar en un objeto JSON al inicio del script,
18 siguiendo esta estructura de grafo:
19
20 const historia = {
21   "inicio": {
22     "texto": "Recibes un mensaje en redes sociales de un desconocido
23             con un enlace extraño. Parece ser de un compañero, pero
24             la forma de escribir es rara.",
25     "imagen_url":
26       ↪ "https://placeholder.co/600x300/1E1B2E/FFFFFF?text=Mensaje+Sospechoso",
27     "opciones": [
28       { "texto": "Hacer clic en el enlace", "destino": "clic_enlace" },

```

```

28     { "texto": "Ignorar el mensaje", "destino": "ignorar_mensaje" },
29     { "texto": "Preguntar al compañero por otro medio", "destino":
      ↪ "preguntar_amigo" }
30   ]
31 },
32 "clic_enlace": {
33   "texto": "Haces clic. Tu pantalla parpadea y tu cuenta se cierra.
34     Has sido víctima de phishing. FIN MALO.",
35   "imagen_url": "https://placeholder.co/600x300/DC143C/FFFFFF?text=Phishing",
36   "opciones": [
37     { "texto": "Volver a empezar", "destino": "inicio" }
38   ]
39 },
40 "ignorar_mensaje": {
41   "texto": "Ignoras el mensaje. Al día siguiente, tu compañero te
42     dice que su cuenta fue hackeada. Te salvaste, pero no
43     ayudaste. FIN NEUTRO.",
44   "opciones": [
45     { "texto": "Volver a empezar", "destino": "inicio" }
46   ]
47 },
48 "preguntar_amigo": {
49   "texto": "Le escribes a tu compañero por WhatsApp. Te dice que él
50     no envió nada y que su cuenta fue hackeada. Adviertes a
51     otros. ¡Bien hecho! FIN BUENO.",
52   "imagen_url": "https://placeholder.co/600x300/10B981/FFFFFF?text=¡Salvado!",
53   "opciones": [
54     { "texto": "Volver a empezar", "destino": "inicio" }
55   ]
56 }
57 // ... (El docente puede expandir esto con más nodos)
58 };
59
60 [LÓGICA DEL JUEGO]
61 1. Una función `mostrarNodo(idNodo)` que:
62   - Limpie los contenedores de texto y opciones.
63   - Obtenga los datos del nodo (ej. `historia[idNodo]`).
64   - Muestre el texto narrativo con efecto de máquina de escribir.
65   - Muestre la imagen.
66   - Cree dinámicamente los botones para cada opción.
67   - Asigne un `click` listener a cada botón que llame a
68     `mostrarNodo(opcion.destino)`.
69 2. Iniciar el juego llamando a `mostrarNodo('inicio')`.
70
71 [EFECTO MÁQUINA DE ESCRIBIR]
72 - El texto debe aparecer letra por letra (usando `setTimeout` o
73   `requestAnimationFrame`).
74
75 [ACCESIBILIDAD]
76 - Los botones de opción deben ser enfocables por teclado.

```

```
77 - Alto contraste de texto.  
78 - Texto alternativo para las imágenes.  
79  
80 [RESTRICCIONES]  
81 - NO usar librerías externas (aparte de Tailwind).  
82 - El código debe estar muy bien comentado para que el docente pueda  
83   editar el JSON de la historia fácilmente.  
84  
85 Comienza con `<!DOCTYPE html>`.
```

Consejo Clave

Consejo de Inmersión: El efecto de máquina de escribir (*typewriter effect*) es un detalle pequeño de JS que aumenta enormemente la inmersión del estudiante en la aventura.

2.10 Proyecto 10: Dashboard de Datos Educativos

Dashboard de Datos

Objetivo Pedagógico

Desarrollar la **Alfabetización de Datos (Data Literacy)**. Enseña a docentes y estudiantes a:

- **Interpretar visualizaciones** de datos (barras, tortas, líneas).
- **Identificar patrones** y tendencias (ej. qué tema tiene más fallas).
- **Tomar decisiones pedagógicas** informadas basadas en evidencia.

Transforma una hoja de cálculo estática (ej. notas) en un panel de control dinámico.

Metodología de Creación

La IA generará una página con una cuadrícula de "widgets" usando:

1. **Chart.js**: La librería de gráficos más popular para web (cargada desde CDN).
2. **CSS Grid**: Para organizar los widgets del dashboard.
3. **JSON de Datos**: Los datos se cargan desde un objeto JSON simple al inicio del script.
4. **Tipos de Widgets**:
 - **KPI (Indicador Clave)**: Tarjetas con un número grande (ej. Promedio General).
 - **Gráfico de Barras**: (ej. Distribución de notas A, B, C, D, F).
 - **Gráfico de Torta (Pie)**: (ej. Tasa de aprobación vs. reprobación).
 - **Gráfico de Línea**: (ej. Progreso a lo largo del tiempo).

Aplicaciones por Área

Gestión Académica:

- Dashboard de rendimiento del curso (notas, asistencia).

- Análisis de resultados de Pruebas SABER (por competencia).

🌐 Sociales / Geografía:

- Dashboard demográfico (población, PIB, alfabetización por país).
- Visualización de resultados electorales.

🧪 Ciencias:

- Presentación de resultados de un experimento de laboratorio.
- Dashboard de datos climáticos (temperatura, precipitación).

</> Proyecto 10: Prompt Maestro (Dashboard de Notas)

```

1  [MODULO] Actúa como un analista de datos experto y desarrollador
2  front-end, especializado en dashboards educativos.
3
4  [TAREA] Crea un Dashboard de Rendimiento Académico para un curso,
5  en un único archivo HTML.
6
7  [TECNOLOGÍAS OBLIGATORIAS]
8  1. Archivo Único: HTML + CSS + JavaScript.
9  2. Estilos: Tailwind CSS desde CDN.
10 3. Gráficos: Chart.js cargado desde CDN.
11   https://cdn.jsdelivr.net/npm/chart.js
12
13 [DATOS DE EJEMPLO (JSON)]
14 Usa estos datos al inicio del script:
15
16 const datosCurso = {
17   "nombreCurso": "Matemáticas Grado 10",
18   "totalEstudiantes": 35,
19   "notas": [5.0, 3.5, 4.2, 2.1, 1.8, 3.9, 4.8, 2.5, 3.0, 4.5, 4.1, 3.3,
20             2.9, 1.5, 5.0, 4.0, 3.8, 3.7, 4.3, 2.2, 3.1, 3.6, 4.0, 4.4,
21             2.8, 3.0, 3.0, 4.9, 2.0, 3.4, 4.6, 1.9, 5.0, 3.2, 4.7],
22   "progresoSemanal": [2.5, 2.8, 3.0, 3.4, 3.8, 4.1] // Promedio por semana
23 };
24
25 [ESTRUCTURA BENTO UI (CSS Grid)]
26 - Contenedor principal: `grid grid-cols-4 grid-rows-2 gap-6 p-6
27   bg-gray-100`
28 - Cada widget: `bg-white rounded-xl shadow-md p-4`
29
30 [DISTRIBUCIÓN DE WIDGETS]
31
32 1. KPI: Promedio General (col-span-1, row-span-1)
33   - Título: "Promedio del Curso"

```

```
34 - Valor: (Calcular el promedio de `datosCurso.notas`)
35 - Estilo: Número muy grande (ej. `text-5xl font-bold
36   text-blue-600`)
37
38 2. KPI: Tasa de Aprobación (col-span-1, row-span-1)
39 - Título: "Aprobados (>= 3.0)"
40 - Valor: (Calcular % de notas >= 3.0)
41 - Estilo: Número grande (ej. `text-5xl font-bold
42   text-green-600`)
43
44 3. Gráfico de Línea: Progreso (col-span-2, row-span-1)
45 - Título: "Progreso del Promedio Semanal"
46 - Canvas para Chart.js
47 - Tipo: `line`
48 - Datos: `datosCurso.progresoSemanal`
49 - Eje X: Etiquetas "Sem 1", "Sem 2", ..., "Sem 6"
50
51 4. Gráfico de Barras: Distribución (col-span-3, row-span-1)
52 - Título: "Distribución de Notas"
53 - Canvas para Chart.js
54 - Tipo: `bar`
55 - Datos: Contar notas en rangos (0-1.9, 2-2.9, 3-3.9, 4-4.9, 5.0)
56 - Eje X: "Bajo", "Básico", "Alto", "Superior", "Excelente"
57
58 5. Gráfico de Torta: Aprob/Reprob (col-span-1, row-span-1)
59 - Título: "Estado de Estudiantes"
60 - Canvas para Chart.js
61 - Tipo: `pie` (o `doughnut`)
62 - Datos: [ % Aprobados, % Reprobados ]
63 - Colores: Verde y Rojo
64
65 [LÓGICA DE JAVASCRIPT]
66 1. Escribir funciones para procesar los datos JSON:
67 - `calcularPromedio(notas)`
68 - `calcularTasaAprobacion(notas, umbral=3.0)`
69 - `contarDistribucion(notas)`
70 2. Usar los resultados para poblar los widgets KPI.
71 3. Usar los resultados para configurar y crear las instancias de
72   `new Chart()` para cada gráfico.
73 4. Asegurarse de que los gráficos se rendericen después de que la
74   página cargue (`DOMContentLoaded`).
75
76 [ACCESIBILIDAD]
77 - Colores amigables para daltónicos en los gráficos.
78 - Títulos claros para cada widget.
79 - Roles ARIA para los gráficos.
80
81 [RESTRICCIONES]
82 - NO usar librerías externas (aparte de Tailwind y Chart.js).
83 - El código debe ser limpio y comentado.
```

```
84 - El dashboard debe ser responsivo (en móvil, los widgets se
85   apilan: `grid-cols-1`).
86
87 Comienza con `<!DOCTYPE html>`.
```

Consejo Clave

Consejo de Datos: Para el widget de Distribución de Notas (Gráfico de Barras), asegúrate de que el código JS clasifique correctamente las notas en los rangos definidos (Bajo, Básico, Alto, Superior, Excelente).

2.11 Proyecto 11: Guía de Taller Académico (LaTeX)

Guía de Taller Académico (LaTeX)

🎯 Objetivo Pedagógico

Crear documentos pedagógicos (guías, talleres, exámenes) con una **calidad tipográfica profesional** impecable, listos para imprimir o compartir en PDF.

- **Rigurosidad Académica:** Manejo perfecto de fórmulas matemáticas ($f(x) = \int x^2 dx$), tablas y bibliografía.
- **Alineación Curricular:** Incorporar fácilmente los lineamientos del Ministerio de Educación de Colombia (MEN), como los DBA.
- **Estandarización:** Crear plantillas institucionales reutilizables.

⚙️ Metodología de Creación

Este prompt es diferente: **no genera HTML, sino código LaTeX (.tex)**.

1. **Preamble Robusto:** La IA configurará un preámbulo con paquetes esenciales:
 - `fontspec` y `babel [spanish]`: Para tildes, eñes y fuentes modernas.
 - `amsmath`: Para matemáticas avanzadas.
 - `tcolorbox`: Para crear cajas de "Definición", "Ejercicio" y "DBA".
 - `enumitem`: Para listas personalizadas.
2. **Entornos Personalizados:** Se le pide a la IA que defina `\newtcolorbox` para crear entornos semánticos.
3. **Compilación:** El docente copia el código `.tex` generado y lo pega en **Overleaf**, seleccionando el compilador `LuaLaTeX`.

💡 Aplicaciones por Área

Matemáticas / Física / Química:

- Exámenes con fórmulas complejas, matrices y reacciones químicas.
- Guías de laboratorio con tablas de datos y gráficos.

Ciencias Sociales / Humanidades:

- Artículos de investigación con citas y bibliografía (formato APA).
- Talleres de lectura crítica con cajas de texto resaltadas.

Todas las Áreas:

- **Planes de Área y Mallas Curriculares:** Documentos institucionales estandarizados.
- **Guías de Taller:** El proyecto principal de este prompt.

Proyecto 11: Prompt Maestro (Guía LaTeX con DBA)

```

1 [MODULO] Actúa como un profesor experto en diseño instruccional y un
2 editor profesional de LaTeX, con conocimiento de los lineamientos
3 del Ministerio de Educación de Colombia (MEN).
4
5 [TAREA] Crea el código fuente completo de un documento .tex para una
6 Guía de Taller sobre [TEMA: ej. Introducción a las Ecuaciones
7 Lineales] para [GRADO: ej. Grado 8°].
8
9 [REGLAS OBLIGATORIAS]
10 1. Compilador: El código debe estar diseñado para compilar con
11 LuaLaTeX.
12 2. Preamble Completo:
13 - Clase: \documentclass[12pt, letterpaper]{article}
14 - Fuentes y Lenguaje: Carga fontspec y babel[spanish]. Usa
15 \babelfont{rm}{Noto Serif} y \babelfont{sf}{Noto Sans}.
16 - Márgenes: \usepackage[margin=1in]{geometry}
17 - Paquetes Esenciales: amsmath (matemáticas), tcolorbox
18 (cajas), enumitem (listas), hyperref (links), booktabs
19 (tablas).
20 3. Cajas Personalizadas (tcolorbox):
21 - Define un objetivobox (color primaryPurple!10)
22 - Define un dbabox (color successGreen!10) para los DBA.
23 - Define un conceptobox (color warningAmber!10) para
24 definiciones.
25 - Define un ejerciciobox (color accentFuchsia!10) para
26 actividades.
27 4. Estructura del Documento:
28 - Título, Autor (Docente), Fecha (\maketitle).
29 - Sección 1: Objetivos de Aprendizaje (usar objetivobox).
30 - Sección 2: Estándares y DBA (MEN Colombia) (usar dbabox).
31 - Sección 3: Marco Teórico / Conceptos Clave (usar conceptobox).
32 - Sección 4: Actividades Prácticas (usar ejerciciobox).
33 - Sección 5: Evaluación.

```

```

34
35 [CONTENIDO DE EJEMPLO] (Para Ecuaciones Lineales)
36
37 - **Título**: "Guía de Taller: El Mundo de las Ecuaciones Lineales"
38 - **Objetivos**:
39   1. Identificar los componentes de una ecuación lineal ( $y=mx+b$ ).
40   2. Resolver ecuaciones lineales para 'x'.
41   3. Graficar una ecuación lineal en el plano cartesiano.
42 - **DBA (Ejemplo)**: "DBA 2: Propone, compara y usa procedimientos
43   inductivos y lenguaje algebraico para formular y poner a prueba
44   conjeturas en diversas situaciones."
45 - **Conceptos**:
46   - ¿Qué es una variable? (en \conceptobox)
47   - ¿Qué es una ecuación lineal? (en \conceptobox)
48   - La pendiente (m) y el intercepto (b) (en \conceptobox con
49     fórmula  $y = mx + b$ )
50 - **Ejercicios**:
51   1. Resuelve para 'x':  $3x + 5 = 14$  (en \ejerciciobox)
52   2. Grafica la línea:  $y = 2x - 1$  (en \ejerciciobox)
53   3. Problema de aplicación: "Un taxi cobra... (en \ejerciciobox)
54
55 [RESTRICCIONES]
56 - El código debe ser 100% completo y compilable en Overleaf.
57 - Incluir ejemplos de cómo usar \amsmath (fórmulas) y \enumitem
58   (listas).
59 - NO incluyas explicaciones antes/después del código.
60
61 Comienza con \documentclass[12pt, letterpaper]{article}.

```

Consejo Clave

Compilación: Copia el código generado, pégalo en un nuevo proyecto en [Overleaf](#), ve al "Menú" (arriba a la izquierda), baja a "Compiler" y asegúrate de seleccionar **LuaLaTeX**. Presiona "Recompile".

2.12 Proyecto 12: Mapa Mental Interactivo

Mapa Mental Interactivo

🎯 Objetivo Pedagógico

Fomentar el **pensamiento radial** y la organización jerárquica de ideas. A diferencia de las listas, los mapas mentales:

- **Muestran conexiones** y relaciones entre conceptos.
- **Facilitan la lluvia de ideas** (Brainstorming) y la síntesis.
- **Mejoran la memoria** al asociar ideas con ubicaciones espaciales.

Una versión interactiva permite ocultar/mostrar ramas para evitar la sobrecarga cognitiva.

⚙️ Metodología de Creación

Se le pedirá a la IA que cree el mapa mental **sin librerías externas** (como D3.js, que es compleja).

1. **Estructura de Datos JSON:** La jerarquía se define en un árbol JSON.

```

1  const datosMapa = {
2    "nombre": "Idea Central",
3    "hijos": [
4      { "nombre": "Rama 1", "hijos": [...] },
5      { "nombre": "Rama 2", "hijos": [...] }
6    ]
7  }

```

2. **Renderizado SVG:** JavaScript recorrerá el JSON y generará elementos **SVG** (<rect>, <text>, <line>) dinámicamente para dibujar el mapa.
3. **Interactividad (JS):** Se añadirán eventos `click` a los nodos. Al hacer clic, se expanden o colapsan las ramas "hijas" (cambiando su visibilidad).

💡 Aplicaciones por Área

📖 Todas las Áreas:

- **Resumen de un tema:** (ej. "Revolución Francesa": Causas, Fases, Consecuencias).
- **Lluvia de ideas** para un proyecto o ensayo.

- **Planificación:** Estructura de un plan de clase o un artículo.

Ciencias:

- "Clasificación de los Seres Vivos" (Reinos, Filos, Clases...).
- "Tipos de Energía" (Potencial, Cinética, Térmica...).

Lenguaje:

- "Análisis de una Oración" (Sujeto, Predicado, Modificadores...).
- "Personajes de una Novela" y sus relaciones.

Proyecto 12: Prompt Maestro (Mapa Mental SVG)

```

1  [MODO] Actúa como un desarrollador front-end experto en visualización
2  de datos con SVG y JavaScript puro.
3
4  [TAREA] Crea un Mapa Mental Interactivo (expandible y colapsable)
5  en un único archivo HTML.
6
7  [TECNOLOGÍAS OBLIGATORIAS]
8  1. Archivo Único: HTML + CSS + JavaScript.
9  2. Estilos: Tailwind CSS desde CDN.
10 3. Gráficos: SVG Inline. NO USAR CANVAS.
114. SIN LIBRÍAS EXTERNAS: NO USAR D3.js, Chart.js, o
12   cualquier librería de mapas mentales. Todo debe ser JS puro.
13
14 [DATOS DE EJEMPLO (JSON)]
15 Usa esta estructura de árbol recursiva:
16
17 const datosMapa = {
18   "nombre": "El Sistema Solar",
19   "hijos": [
20     { "nombre": "Sol (Estrella)", "hijos": [] },
21     { "nombre": "Planetas Interiores", "hijos": [
22       { "nombre": "Mercurio", "hijos": [] },
23       { "nombre": "Venus", "hijos": [] },
24       { "nombre": "Tierra", "hijos": [
25         { "nombre": "Luna", "hijos": [] }
26       ]},
27       { "nombre": "Marte", "hijos": [] }
28     ]},
29     { "nombre": "Planetas Exteriores", "hijos": [
30       { "nombre": "Júpiter", "hijos": [] },
31       { "nombre": "Saturno", "hijos": [] },
32       { "nombre": "Urano", "hijos": [] },
33       { "nombre": "Neptuno", "hijos": [] }

```

```
34     }},
35     { "nombre": "Cinturón de Asteroides", "hijos": [] }
36   ]
37 };
38
39 [ESTRUCTURA DE LA INTERFAZ]
40 1. Contenedor SVG: Un único elemento <svg> de 1000px de ancho y
41     800px de alto, con un fondo (bg-gray-50).
42 2. Layout: El mapa debe ser un árbol horizontal (Nodo raíz a la
43     izquierda, ramas hacia la derecha).
44
45 [LÓGICA DE JAVASCRIPT]
46 1. Función de Renderizado Recursiva:
47     - function renderNodo(nodo, x, y, svgContenedor)
48     - Esta función debe:
49         a. Dibujar el nodo actual:
50             - Un <rect> (caja) en la posición (x, y)
51             - Un <text> con nodo.nombre dentro del rect
52         b. Si el nodo tiene hijos:
53             - Calcular las posiciones Y para cada hijo.
54             - Dibujar una <line> (línea) desde el nodo padre al nodo
55               hijo.
56             - Llamarse a sí misma: renderNodo(hijo, x + 200, y_hijo, ...)
57 2. Cálculo de Layout: La lógica debe calcular las posiciones 'y'
58     para que los nodos no se superpongan (layout de árbol simple).
59 3. Interactividad (Click):
60     - Añadir un evento click a cada <rect> (nodo).
61     - Al hacer clic en un nodo:
62         a. Ocultar o mostrar todos sus descendientes (líneas e hijos).
63         b. Cambiar el color del nodo padre (ej. a rojo si está
64            colapsado, verde si está expandido).
65         c. (Opcional) Animar la aparición/desaparición.
66
67 [ESTILO VISUAL (SVG)]
68 - Nodos (rect):
69     - fill: white
70     - stroke: blue (ej. stroke-blue-500)
71     - rx: 8 (bordes redondeados)
72 - Texto (text):
73     - fill: black
74     - font-family: Inter, sans-serif
75     - font-size: 14px
76 - Líneas (line):
77     - stroke: gray (ej. stroke-gray-400)
78     - stroke-width: 2
79
80 [RESTRICCIONES]
81 - El código debe ser 100% JS puro + SVG + Tailwind.
82 - El mapa debe renderizarse correctamente al cargar la página.
83 - La interactividad de colapsar/expandir debe funcionar.
```

```
84 - Código bien comentado.  
85  
86 Comienza con `<!DOCTYPE html>`.
```

Consejo Clave

Compilación: Copia el código generado, pégalo en un nuevo proyecto en [Overleaf](#), ve al "Menú" (arriba a la izquierda), baja a "Compiler" y asegúrate de seleccionar **LuaLaTeX**. Presiona "Recompile".

3 Conclusión: El Siguiete Paso

Has Completado el Laboratorio

Este documento te ha entregado las **plantillas maestras** para 12 tipos distintos de recursos educativos digitales. Has visto cómo, con un prompt bien estructurado, puedes dirigir a la IA para que actúe como tu programador, diseñador gráfico y asistente pedagógico.

Tu misión ahora es iterar. Ningún prompt es perfecto en el primer intento. Toma estos 12 proyectos y adáptalos a tu área específica:

- Cambia los datos JSON en los juegos y mapas mentales.
- Ajusta las ecuaciones en los simuladores de física.
- Modifica los temas en las guías de LaTeX.
- Refina los colores y estilos para que coincidan con tu visión.

El poder no está en el código generado; está en tu habilidad para **dirigir, refinar y contextualizar** la salida de la IA.

¡Bienvenido a la era del docente creador!

A Anexo J: Referencias Bibliográficas (Perfeccionadas)

Referencias

- [1] Ministerio de Educación Nacional de Colombia (2024). *Directiva Ministerial No. 003 del 27 de junio de 2024*. Lineamientos para integración de tecnologías digitales e IA en educación.
- [2] Congreso de Colombia (2012). *Ley 1581 de 2012*. Régimen General de Protección de Datos Personales (Habeas Data).
- [3] UNESCO (2023). *Guidance for Generative AI in Education and Research*. París: UNESCO.
- [4] Mollick, E. (2024). *Co-Inteligencia: Vivir y trabajar con IA*. Wharton School Press.
- [5] Comisión Económica para América Latina y el Caribe (CEPAL). (2023). *La inteligencia artificial en la educación en América Latina: Oportunidades y desafíos*. Santiago: CEPAL.
- [6] Bustamante, C. A. (2023). IA Generativa en la Educación Superior: Retos y Oportunidades para Docentes en América Latina. *Revista de Educación a Distancia (RED)*, 23(74).
- [7] Russell, S., & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson.
- [8] Montoya Palacios, N. F. (2024). *EduLab: Ecosistema de Recursos Educativos Interactivos*. profenestor.com/edulab
- [9] Harari, Y. N. (2018). *21 Lessons for the 21st Century*. Spiegel & Grau.
- [10] Vaswani, A. et al. (2017). *Attention Is All You Need*. Documento de investigación de Google Brain (NIPS).
- [11] Wolfram, S. (2023). *What Is ChatGPT Doing... and Why Does It Work?*. Wolfram Media.