

A smiling man with a beard and headphones is working at a desk. He is looking at a laptop screen. A Pomodoro timer is visible on the desk. The background is blurred, showing a modern office environment.

Compositor de Software: Pomodoro Timer

Founderz

 Microsoft

A woman with dark curly hair is sitting at a desk, looking at a laptop. A large, semi-transparent Pomodoro timer is overlaid on the laptop screen. The timer has a circular progress bar with red, green, and yellow segments. The text '01' is on the left, and 'Diseño de un temporizador Pomodoro' is on the right.

01 Diseño de un temporizador Pomodoro

Fundamentos de la técnica Pomodoro

El temporizador Pomodoro combina una técnica de gestión del tiempo con una pequeña aplicación web centrada en la simplicidad y la productividad.

La estructura del ciclo Pomodoro

La técnica se basa en bloques de trabajo concentrado de 25 minutos seguidos de pausas cortas de 5 minutos. Estos bloques se denominan pomodoros.

Tras completar cuatro pomodoros, se recomienda realizar un descanso más largo, lo que ayuda a mantener la concentración, reducir la fatiga mental y organizar las tareas en unidades manejables.

Objetivos de la aplicación web

El objetivo principal es disponer de un **temporizador sencillo, atractivo y funcional**, sin anuncios, que apoye el uso de la técnica Pomodoro en el trabajo diario.

- Gestionar de forma automática los periodos de trabajo y descanso.
- Ofrecer controles claros para iniciar, pausar y reiniciar el temporizador.
- Mostrar información comprensible sobre el número de pomodoros completados y el estado actual del ciclo.

Funciones clave del temporizador Pomodoro

La definición precisa de las funcionalidades permite a la IA generar una aplicación completa en un solo archivo HTML.

Alternancia automática

El temporizador cambia de forma automática entre 25 minutos de trabajo y 5 minutos de descanso, sin intervención manual entre fases.

Controles básicos del tiempo

Los botones de **iniciar**, **pausar** y **reiniciar** permiten gestionar el flujo del pomodoro de manera clara e intuitiva.

Notificación visual

La interfaz muestra cambios visuales cuando termina un periodo, ayudando a identificar de un vistazo si se está en trabajo o descanso.

Alarma sonora

Un sonido de alarma resalta el fin de cada fase, reforzando la notificación visual y facilitando el cambio de tarea o de descanso.

Contador de pomodoros

La aplicación lleva la cuenta de los ciclos completados, lo que permite estimar la carga de trabajo y planificar descansos largos.

Diseño responsivo con Tailwind CSS

El uso de **Tailwind CSS** desde un CDN facilita un diseño atractivo, adaptable a móviles y con posibilidad de usar emojis de gran tamaño.

Elementos técnicos y beneficios del temporizador

Cada componente de la aplicación cumple una función técnica concreta que se traduce en ventajas directas para la productividad personal.

Elemento de la aplicación	Implementación técnica	Beneficio práctico
Temporizador principal	Cuenta atrás en segundos para trabajo y descanso, gestionada con JavaScript en un único archivo HTML.	Estructura el tiempo en bloques claros, evitando la multitarea constante y las interrupciones continuas.
Controles de reproducción	Botones asociados a funciones de iniciar, pausar y reiniciar que controlan el temporizador y su estado interno.	Facilita detenerse momentáneamente sin perder el progreso del pomodoro o comenzar un nuevo ciclo cuando convenga.
Notificación visual	Cambios de texto, colores o iconos en la interfaz cuando se completa un periodo de tiempo.	Permite identificar rápidamente la fase actual incluso sin escuchar la alarma sonora.
Alarma de sonido	Objeto <code>Audio</code> en JavaScript (por ejemplo, <code>alarmSound.play()</code>) con una URL a un archivo de sonido.	Marca con claridad el final de cada fase, útil cuando la pantalla no está a la vista en todo momento.
Contador de pomodoros	Variable que incrementa en uno cada vez que se completa un ciclo de trabajo y descanso.	Ofrece una medida sencilla del trabajo realizado y ayuda a decidir cuándo realizar un descanso largo.
Carga de Tailwind desde CDN	Inclusión de la hoja de estilos mediante una URL pública, sin necesidad de compilación ni herramientas adicionales.	Acelera la creación de interfaces modernas y permite centrarse en la lógica del temporizador.



Un prompt eficaz para la IA describe con precisión funcionalidades, diseño, librerías necesarias (como Tailwind CSS desde un CDN) y formato del archivo final ejecutable.



02 Del código generado a la app funcional

Flujo de trabajo con IA y editor web

La combinación de una IA generativa con un editor como Visual Studio Code en versión web permite crear rápidamente un temporizador Pomodoro funcional.

1

Definir el prompt de la aplicación: Se describen los tiempos de trabajo y descanso, los controles necesarios, las notificaciones visuales y sonoras, el uso de Tailwind CSS y el deseo de obtener todo el código en un solo archivo HTML.

2

Generar y revisar el código: La IA devuelve el archivo HTML completo con la estructura, los estilos desde CDN y el JavaScript embebido. Es recomendable leerlo por encima para localizar las principales constantes y funciones.

3

Copiar el código en el editor web: En Visual Studio Code versión web se crea un archivo nuevo, se pega el código y se observa cómo el editor resalta la sintaxis de HTML, CSS y JavaScript.

4

Guardar la aplicación en una carpeta: Se guarda el archivo con un nombre descriptivo dentro de una carpeta específica, por ejemplo “pomodoro-app”. El uso o no de acentos en los nombres de archivo queda a criterio de la persona que desarrolla.

5

Ejecutar con doble clic en el navegador: Al abrir el archivo HTML en un navegador moderno, la aplicación se carga sin necesidad de instalaciones adicionales ni compilación.

6

Probar y ajustar detalles: Se comprueba el funcionamiento de los ciclos, las notificaciones y el sonido, realizando pequeños cambios en el código para adaptarlo a las necesidades reales.

Ajustes rápidos para pruebas y depuración

Una vez que el temporizador está en marcha, resulta muy útil modificar algunos parámetros para validar el comportamiento sin esperar 25 minutos en cada ciclo.

Reducir tiempos para validar el flujo

En el código suelen aparecer constantes como **25 minutos en segundos** y **5 minutos en segundos**, a menudo expresadas como `25 * 60` y `5 * 60`.

Para hacer pruebas rápidas, basta con sustituir esas expresiones por valores mucho menores, por ejemplo **10 segundos** para el periodo de trabajo y **3 o 7 segundos** para los descansos. De esta forma se puede observar en pocos segundos cómo el temporizador cambia de fase, incrementa el contador de pomodoros y reinicia el siguiente ciclo.

Diagnóstico del sonido de alarma

Si al finalizar la cuenta atrás no se escucha ningún sonido, conviene revisar la parte del código donde se declara la variable de audio, por ejemplo `const alarmSound = new Audio("URL-del-sonido")`, y la llamada a `alarmSound.play()`.

- Verificar si la URL de audio es accesible abriéndola directamente en el navegador.
- En caso de recibir un error de acceso, sustituirla por otra dirección de un archivo de sonido disponible en línea o alojado por la propia persona.
- Si se desea, se puede pedir a la IA una nueva versión del código que incluya la URL de sonido correcta y los tiempos de prueba deseados.

Buenas prácticas al experimentar con el código

Pequeñas decisiones técnicas facilitan la iteración sobre el temporizador Pomodoro y reducen errores durante las pruebas.

Trabajar con copias

Antes de modificar el código generado por la IA, resulta conveniente guardar una copia original para poder volver atrás si algo falla.

Usar constantes descriptivas

Variables como `WORK_DURATION` o `BREAK_DURATION` ayudan a localizar rápidamente los tiempos que se desean modificar para las pruebas.

Probar con tiempos muy cortos

Reducir los minutos a segundos permite validar el comportamiento completo del ciclo (trabajo, descanso y contador) en pocos instantes.

Revisar la consola del navegador

En caso de que el sonido o el contador no funcionen, la consola de desarrollador del navegador suele mostrar errores que orientan sobre el problema.

Cuidar el diseño desde el inicio

Indicar a la IA que utilice un diseño atractivo y responsivo, incluso con emojis, mejora la experiencia sin añadir complejidad técnica significativa.

Iterar con la IA

Se pueden solicitar variantes del código que ajusten tiempos, colores, tipografías o sonidos, manteniendo la misma lógica general de la aplicación.



La IA actúa como compositora de software: genera una base completa, pero la validación, las pruebas y los pequeños ajustes siguen siendo responsabilidad humana.



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.