



Compositor de Software: Generador de passwords

Founderz



Microsoft



01 Diseñar un generador de contraseñas

Objetivo y requisitos del generador seguro

El proyecto consiste en crear, junto con una inteligencia artificial, una aplicación web sencilla para generar contraseñas seguras y fáciles de usar.

Funcionalidad básica de la aplicación

El núcleo del proyecto es una herramienta de generación de contraseñas seguras que funcione en cualquier navegador sin instalaciones adicionales.

La aplicación debe permitir ajustar la longitud de la contraseña, decidir qué tipos de caracteres incluir y ofrecer una manera rápida de copiar el resultado al portapapeles. Además, es importante que la persona usuaria entienda si la contraseña generada es fuerte, media o débil mediante un indicador claro de fortaleza de la contraseña.

Requisitos funcionales mínimos

El comportamiento esperado de la aplicación se resume en una serie de funciones imprescindibles:

- Permitir especificar la longitud de la contraseña dentro de un rango seguro.
- Ofrecer casillas para incluir o excluir mayúsculas, minúsculas, números y símbolos especiales.
- Generar una contraseña aleatoria basada en las opciones seleccionadas.
- Mostrar de forma visual y textual la fortaleza estimada de la contraseña.

- Incorporar un botón para copiar la contraseña al portapapeles con un solo clic.

Requisitos técnicos del código

Desde el punto de vista técnico se busca un enfoque muy simple: un único archivo HTML que contenga el marcado, el estilo y el JavaScript necesario para funcionar.

El diseño visual se apoya en **Tailwind CSS** cargado desde internet mediante CDN, evitando procesos de compilación o configuración compleja. El archivo debe poder abrirse con doble clic en el navegador, sin dependencias adicionales ni servidores locales.

Extensión: contraseñas basadas en palabras reales

Además de la contraseña puramente aleatoria, la aplicación incorpora una opción para generar contraseñas basadas en palabras reales aleatorias, tomadas de un diccionario.

Estas contraseñas tipo frase (passphrase) son, en general, más fáciles de recordar, aunque algo menos seguras que una cadena completamente aleatoria. Por ello, se requiere un número mínimo de palabras que garantice una longitud y complejidad suficientes.

Características clave del generador de contraseñas

El generador combina seguridad, usabilidad y un diseño atractivo en una sola página web de fácil ejecución.

Control de longitud flexible

La longitud de la contraseña se define mediante un control numérico o un deslizador, con límites mínimos y máximos predefinidos para evitar opciones inseguras.

Selección de tipos de caracteres

Casillas independientes permiten incluir mayúsculas, minúsculas, números y símbolos especiales, combinando seguridad y preferencias personales.

Indicador visual de fortaleza

El sistema evalúa la contraseña generada y muestra su fortaleza con colores y etiquetas claras: débil, media o fuerte.

Copiado rápido al portapapeles

Un botón dedicado copia la contraseña directamente al portapapeles, evitando errores de escritura y agilizando el uso.

Modo frase con palabras reales

Una opción adicional genera contraseñas basadas en palabras de un diccionario, combinando memorabilidad con una longitud considerable.

Diseño atractivo con Tailwind CSS

La interfaz se construye con Tailwind cargado por CDN, permitiendo un diseño moderno sin necesidad de configurar entornos de compilación.

Secuencia para definir el prompt del proyecto

Una buena definición del prompt permite que la inteligencia artificial genere una aplicación completa, usable y fácil de mantener.

1

Precisar el objetivo de la aplicación: indicar que se necesita una aplicación web de una sola página para generar contraseñas seguras, ejecutable con doble clic en el navegador.

2

Describir con detalle las funcionalidades: incluir longitud configurable, selección de tipos de caracteres, generación aleatoria, indicador de fortaleza y botón de copiado al portapapeles.

3

Definir requisitos de diseño y tecnología: pedir un diseño atractivo, uso de Tailwind CSS cargado desde internet y todo el código contenido en un único archivo HTML.

4

Añadir características opcionales: solicitar la opción de generar contraseñas basadas en palabras reales aleatorias y exigir un número mínimo de palabras para mantener la seguridad.

5

Solicitar instrucciones de uso: pedir que se expliquen claramente los pasos para guardar el archivo, ejecutarlo y entender el funcionamiento básico de la interfaz.

6

Introducir retos creativos: invitar a la IA a sorprender con una paleta de colores inusual o fuentes poco frecuentes, fomentando resultados de diseño menos convencionales.

i

Las contraseñas basadas en varias palabras reales largas pueden ofrecer un buen equilibrio entre seguridad y memorabilidad, siempre que mantengan una longitud considerable.



Mejoras, versiones y diseño guiado por IA

Una vez creada la primera versión funcional, el verdadero valor aparece al iterar sobre el resultado inicial y depurarlo.

Refinamiento funcional de la primera versión

Tras validar que la aplicación funciona, es habitual detectar oportunidades de mejora. En este proyecto se ajustaron varios aspectos clave, empezando por el **control de longitud**, sustituyendo el campo numérico por un **slider** con valores entre 6 y 90 caracteres.

También se redefinió la lógica de la **fortaleza mínima**: cuando la longitud es menor de 10 caracteres, la interfaz debe mostrar un nivel de seguridad muy bajo, resaltado en rojo. Además, se amplió el **diccionario de palabras** utilizado para las contraseñas tipo frase, incorporando términos de categorías muy distintas para mejorar la aleatoriedad.

Mejoras de diseño y experiencia de uso

La IA puede recibir indicaciones específicas sobre estilo visual. En este caso se pidió una **paleta de colores** más cuidada y el uso de **tipografías** poco habituales para crear una estética más distintiva.

- Uso de combinaciones de color coherentes con la temática de seguridad.
- Tipografías que mantengan la legibilidad de caracteres similares (O/0, l/1).
- Jerarquía visual clara para longitud, opciones de caracteres y resultado.

Gestión de versiones del archivo

Para controlar la evolución del proyecto se guardaron varias versiones del archivo HTML, con nombres que reflejan su estado, como “**versión 1**”, “**versión 2**” o incluso “**versión rota**” cuando se introdujeron errores a propósito.

Esta práctica permite conservar siempre una **versión estable** que funciona, mientras se experimenta con nuevas ideas en copias independientes. Si una iteración falla o la corrección no es satisfactoria, basta con volver al archivo anterior sin perder el trabajo previo.

Manejo de respuestas largas de la IA

En ocasiones, la IA interrumpe la generación del código por límite de longitud. Cuando esto sucede, puede continuarse la petición con mensajes como “**continúa**”, obteniendo el resto del código en una segunda respuesta.

Al copiar el resultado al editor (por ejemplo, Visual Studio Code en su versión web), conviene pegar **las partes en orden** y evitar líneas extra que corten estructuras. Los indicadores de error en rojo ayudan a confirmar que el código vuelve a ser sintácticamente correcto tras pegar todas las secciones.

Configuraciones de contraseña y su impacto en la seguridad

Los ajustes de longitud y complejidad modifican significativamente la fortaleza de la contraseña generada por la aplicación.

Configuración	Ventajas	Riesgos y recomendaciones
6–9 caracteres con todos los tipos de caracteres	Fácil de introducir rápidamente, adecuada para pruebas o contraseñas temporales de muy corto plazo.	Considerada muy débil ; el indicador debería mostrarse en rojo y evitarse para cuentas importantes.
10–15 caracteres con variedad de caracteres	Equilibrio razonable entre seguridad y comodidad, especialmente si se incluyen símbolos y números.	Puede ser vulnerable para servicios críticos; conviene combinar con verificación en dos pasos.
16+ caracteres aleatorios con alta complejidad	Contraseñas muy fuertes , difíciles de adivinar incluso con ataques automatizados.	Más difíciles de recordar; recomendable el uso de gestores de contraseñas para almacenarlas.
Frases de 4+ palabras reales variadas	Más memorables, especialmente si las palabras no tienen relación evidente entre sí.	Se requiere un número suficiente de palabras y categorías distintas para que la seguridad sea adecuada.

Proceso para depurar el código con ayuda de la IA

Cuando el archivo HTML deja de funcionar, la inteligencia artificial puede colaborar en la detección y corrección de errores.

- 1** **Detectar el fallo en el editor o navegador:** al guardar el archivo, el editor puede marcar líneas en rojo y la página no se comporta como se esperaba en el navegador.
- 2** **Conservar una copia de la versión defectuosa:** guardar el archivo con un nombre que indique que está roto ayuda a documentar el estado del error y facilita comparaciones futuras.
- 3** **Compartir el código con la IA:** arrastrar el archivo o copiar y pegar su contenido para que la IA pueda leerlo completo y analizar el contexto del fallo.
- 4** **Formular una petición de revisión precisa:** explicar que el editor marca errores y solicitar que se indiquen las **líneas concretas** que deben cambiarse y el código correcto a sustituir.
- 5** **Aplicar y probar las correcciones:** introducir los cambios sugeridos, guardar el archivo y recargar la página. Si persisten errores, se puede volver a enviar la nueva versión para un análisis adicional.
- 6** **Generar, si es necesario, un archivo completo corregido:** en ocasiones la IA propone todo el código reescrito; al pegarlo en un archivo nuevo, puede corregir errores introducidos manualmente sin mencionarlos explícitamente.



La inteligencia artificial actúa como un poderoso compositor de software, pero la responsabilidad de revisar, probar y validar cada versión sigue siendo 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.