



Desafío inicial

Desafío del generador de frases de Next.js

Objetivo

Cree una aplicación Next.js que genere frases combinando sujetos y predicados mientras aplica lógica condicional. La aplicación debe incluir una interfaz de usuario para mostrar los resultados de forma dinámica.

Instrucciones

1. Configura tu proyecto:
 - a. Cree un nuevo proyecto Next.js usando create-next-app. b. Configure TailwindCSS para aplicar estilos.
2. Defina los datos:
 - a. En el backend, defina dos matrices: i. Sujetos: ["El perro", "La tortuga", "Mi amigo", "Sebastián"] ii. Predicados: ["corre rápido", "es muy sabio", "le encanta programar", "canta mal"]
3. Implemente la lógica condicional: a. Si el sujeto tiene más de 8 caracteres, agregue un adjetivo como "inteligente". b. Si el predicado incluye la palabra "codificación", agregue un signo de exclamación (!) al final.
 - do. Si no se cumple ninguna de las condiciones, simplemente concatene el sujeto y el predicado.
4. Descomponga la lógica: a. Cree un método para seleccionar aleatoriamente un sujeto. b. Cree otro método para seleccionar aleatoriamente un predicado. c. Combine y aplique las reglas para generar una frase.
5. Diseñar el frontend: a. Construir una interfaz de usuario donde los usuarios puedan:
 - i. Haga clic en un botón para generar una nueva frase.
 - ii. Vea la frase generada en tiempo real.
6. Conexión backend:



- a. Utilice una ruta de API de Next.js para manejar la lógica y devolver la frase generada a la Interfaz.

Entregables

1. Repositorio del proyecto:
 - a. Envíe su proyecto completado a un repositorio público de GitHub.
 - b. Incluya un README claro con instrucciones de configuración.
2. Demostración:
 - a. Asegúrese de que la aplicación se ejecute sin errores y genere frases de forma dinámica.
 - b. Incluya capturas de pantalla o un video corto de la aplicación en acción (opcional).

Características esperadas

- Generación dinámica de frases: o La aplicación debe aplicar la lógica condicional descrita anteriormente.
- Interfaz de usuario moderna: o Utilice TailwindCSS para crear una interfaz limpia y visualmente atractiva.
- Separación de preocupaciones: o La lógica debe implementarse en la ruta de la API del backend.
- o El frontend debe centrarse en mostrar el resultado y la interacción del usuario.

Desafíos adicionales (opcional)

- Agregue más sujetos y predicados a las matrices.
- Permita que los usuarios ingresen sujetos o predicados personalizados a través de la interfaz.
- Mejore la interfaz de usuario con animaciones o estilos avanzados utilizando TailwindCSS.

¡Buena suerte y diviértete creando tu generador de frases Next.js!