

Guía Básica para Prompts Efectivos en Cursor

Esta guía describe cómo crear prompts claros y concisos al usar el editor con IA (por ejemplo, Cursor). Además, explica cómo agregar archivos de contexto (como un README.md o un archivo shipments.json) para que la IA entienda mejor el proyecto y genere respuestas más acertadas.

1. Principios Clave

KISS (Keep It Simple, Stupid)

- Di exactamente lo que necesitas
- Un problema = una solución
- Evita explicaciones innecesarias en el prompt

YAGNI (You Aren't Gonna Need It)

- Pide solo lo que necesitas ahora
- No agregues requisitos “por si acaso”
- Enfócate en el problema actual que deseas resolver

SOLID

- Cada componente debe tener un propósito claro
- Pide código fácil de extender y mantener
- Define bien las interfaces y dependencias si necesitas varias partes trabajando juntas

2. ¿Por Qué Funciona?

- **Menos Confusión:** Al ser conciso, la IA no se pierde en detalles irrelevantes
- **Respuestas Más Precisas:** La IA se enfoca en la tarea exacta
- **Código Más Limpio:** Se tiende a generar soluciones más organizadas y mantenibles
- **Ahorro de Tiempo:** Menos revisiones y correcciones posteriores

3. Ejemplos

Ejemplo Malo

“Necesito un sistema que tal vez pueda hacer X, Y, Z y quizás W en el futuro. El sistema debe tener un montón de configuraciones y un panel de control muy avanzado...”

Problema: El prompt es ambiguo, mezcla demasiados requisitos y añade funcionalidades que quizá no son necesarias en este momento.

Ejemplo Bueno

“Necesito una función que haga X. Debe seguir KISS y YAGNI. Solo implementa la funcionalidad actual.”

Ventaja: El prompt es claro y cumple las tres reglas (KISS, YAGNI, SOLID). Se solicita exactamente lo que hace falta, sin mezclar temas.

4. Agregar Archivos de Contexto al Composer

En un proyecto práctico, la IA se beneficia enormemente de tener acceso a archivos que describan el contexto. En Cursor, puedes adjuntar o asociar archivos al “Context” para que la IA los analice.

Archivos Importantes

README.md

- Instrucciones generales del proyecto
- Librerías utilizadas
- Estructura de archivos
- Detalles del ejercicio

shipments.json

- Archivo con datos reales o de prueba
- Estructura de datos (campos order_id, status, region, etc.)

Otros Archivos Relevantes

- Código base
- Configuraciones
- Documentación adicional

Pasos para Agregar un Archivo

1. Ve a la sección de “Context” o “Composer”
2. Haz clic en “Add File” o arrastra el archivo
3. Verifica que aparezca en la lista de contextos
4. Menciona en tu prompt si necesitas que la IA consulte un archivo específico

5. Cómo Crear un Prompt Efectivo

Paso 1: Expón Tu Objetivo (1-2 Frases)

“Necesito generar una interfaz gráfica en React para mostrar envíos 3PL.”

Paso 2: Añade Detalles Clave

“Los datos de envíos están en shipments.json. Cada objeto tiene campos como order_id, date, status, region, etc.”

Paso 3: Indica la Salida Deseada

“Genera el componente principal que cargue los datos y los muestre en una tabla, siguiendo las prácticas KISS y YAGNI.”

Paso 4: Menciona Requisitos Específicos

“Usa únicamente hooks de React, no librerías adicionales. Muestra una opción de filtrado por status y region.”

Paso 5: Revisa y Recorta

- ¿Hay algo que pueda omitirse sin perder claridad?
- ¿Hay requisitos innecesarios en este momento?

6. Integrando Todo en Cursor

Antes de Enviar

1. Verifica tu Prompt: ¿Es claro y corto?
2. Confirma los Archivos de Contexto:
 - README.md
 - shipments.json
3. Pulsa “Enviar Prompt”

Evaluación

- ¿Cumple KISS, YAGNI y SOLID?
- ¿Necesitas hacer preguntas de seguimiento?

7. Conclusión

Seguir estas pautas te ayudará a: - Obtener respuestas más útiles y enfocadas
- Mantener la limpieza y robustez del código - Incorporar eficazmente datos y documentación como contexto

Recuerda: Siempre puedes refinar tus prompts conforme ves los resultados. La clave está en comunicar tus necesidades de forma clara y concisa, y en adjuntar los archivos necesarios para que la IA tenga todo el contexto a mano.