| Video | Audio |
| --- | --- |
| FADE IN: | Un *prompt* es la instrucción que das al asistente de inteligencia artificial. Puede ser una solicitud de explicación sobre un código o una pregunta sobre un concepto de programación.  El asistente analiza el contexto y el historial del chat para generar una respuesta. |
| INS.WIPE: Diseña peticiones  INS.NUMS:   1. Describe el propósito   INS.TXT: Crea una función en lenguaje python para validar contraseñas. Debe verificar que tengan al menos ocho caracteres, incluya mayúsculas, minúsculas, números y símbolos.   1. Da ejemplos   INS.TXT: Contraseña: ClaveSegura1!  INS.TXT: Salida: Contraseña Segura  INS.TXT: Contraseña: abc123!  INS.TXT: Salida: Contraseña Insegura   1. Divide tareas   INS.TXT: Crea una función en lenguaje python para validar contraseñas.  Primero debes verificar que la longitud mínima sea de ocho caracteres.  Después revisa el uso de mayúsculas y minúsculas.  Finalmente confirma que incluya números y símbolos.   1. Indica convenciones   INS.TXT: El código debe tener nombres descriptivos para las variables y funciones, estar documentado con comentarios y sin errores de sintaxis.  [*~~Ver video~~*](https://drive.google.com/file/d/1L0k5yL6Q6X_A-8q8MHvaKtJivTX4XfJV/view?usp=drive_link)   1. Sé específico   INS.TXT: ¿Cómo puedo modificar la función para que acepte solo contraseñas con números y símbolos?  [*~~Ver video~~*](https://drive.google.com/file/d/1BEwLxcH0pxnOrMn5BrbpjJb0ongIaLqi/view?usp=drive_link) | Para que esta respuesta sea precisa, diseña tus *prompts* siguiendo estos pasos:   1. Explica de forma clara lo que quieres lograr. Por ejemplo: Crea una función en lenguaje ***python*** para validar contraseñas. Debe verificar que tengan al menos ocho caracteres, incluya mayúsculas, minúsculas, números y símbolos. 2. Proporciona entradas y salidas esperadas para que el asistente comprenda el resultado que buscas. 3. Separa las peticiones complejas en pasos simples para que el asistente los resuelva uno a uno.   Por ejemplo:  Primero pide que valide la longitud, después que revise el uso de mayúsculas y minúsculas, y al final que valide números y símbolos.   1. Indica que el código debe   usar nombres descriptivos para variables y funciones, incluir comentarios, tener un estilo consistente y sin errores de sintaxis.   1. Formula preguntas directas para evitar ambigüedades cuando solicites explicaciones. Por ejemplo: ¿Cómo puedo modificar la función para que acepte solo contraseñas con números y símbolos?. |
| INS.WIPE: Seguimiento de la conversación  INS.NUMS:   1. Edita tu petición 2. Crea un subhilo 3. Navega y compara las respuestas 4. Califica las respuestas 5. Crea un chat para cada tarea   [*~~Ver video~~*](https://drive.google.com/file/d/1Gh7SLMIC62NDWNds6ZlQ-0hYMBUQzRgG/view?usp=drive_link) | Si el resultado no es lo que esperabas, realiza lo siguiente:   1. Coloca el cursor sobre tu petición, haz clic en el icono de lápiz, añade los detalles necesarios y envíala nuevamente. 2. Utiliza el botón **reintentar** para que el asistente genere otra respuesta, o **reintentar con modelo** para obtener la respuesta desde un modelo distinto. La nueva respuesta aparecerá en un subhilo. 3. Usa las flechas de navegación para moverte entre las respuestas y compararlas. 4. Califica una respuesta como buena o mala para que el asistente la considere como contexto positivo o negativo. 5. Inicia una conversación nueva cuando trabajes en otra tarea, así evitarás mezclar contexto y mantendrás el enfoque en un solo objetivo. |
| FADE OUT | Sigue estos pasos para diseñar tus *prompts*, revisa los resultadosy ajusta lo necesario. Cada ajuste te acercará a obtener exactamente lo que buscas. |

**Fuentes:**

[**https://docs.github.com/es/copilot/get-started/quickstart?tool=webui**](https://docs.github.com/es/copilot/get-started/quickstart?tool=webui)

[**https://docs.github.com/en/copilot/concepts/prompt-engineering**](https://docs.github.com/en/copilot/concepts/prompt-engineering)

[**https://docs.github.com/en/copilot/reference/cheat-sheet**](https://docs.github.com/en/copilot/reference/cheat-sheet)

[**https://github.blog/developer-skills/github/how-to-write-better-prompts-for-github-copilot/**](https://github.blog/developer-skills/github/how-to-write-better-prompts-for-github-copilot/)

[**https://learn.microsoft.com/es-es/visualstudio/ide/visual-studio-github-copilot-chat?view=vs-2022**](https://learn.microsoft.com/es-es/visualstudio/ide/visual-studio-github-copilot-chat?view=vs-2022)