

# IACADEMY

DE FUNDAMENTOS A ARQUITECTURA DE IA

MÓDULO 02

## Prompting que funciona

Fundamentos

iacademy.com — 2026

## MÓDULO 02

# Prompting que funciona

---

**Nivel:** Fundamentos

**Autor:** Ricardo Gutierrez

**Publicación:** Mayo 2026

**Plataforma:** [iacademy.com](https://iacademy.com)

Este material es parte del curso completo de IAcademy.

Uso personal e intransferible. Queda prohibida su redistribución o reproducción sin autorización.

## Los 7 componentes de un prompt profesional

La mayoría de la gente escribe prompts de 1 línea y espera resultados profesionales. No funciona así. Un prompt profesional puede tener hasta 7 componentes. No siempre necesitas los 7, pero saber que existen cambia radicalmente tus resultados.

### 1. Rol

Quien debe ser la IA. Define el nivel de expertise y la perspectiva.

```
Eres un consultor senior de ciberseguridad con 15 años de experiencia en empresas del sector financiero.
```

### 2. Contexto

Información de fondo que la IA necesita para personalizar la respuesta.

```
Trabajo en una pyme de 50 empleados del sector salud. Tenemos un presupuesto de IT de 120.000 EUR anuales. No tenemos equipo de seguridad dedicado.
```

### 3. Tarea

Que quieres que haga. Especifico y medible.

```
Redacta una política de contraseñas para toda la empresa.
```

### 4. Formato

Como quieres el resultado. Estructura, longitud, estilo.

```
En bullet points. Máximo 10 puntos. Sin introducción ni conclusión. Cada punto debe ser accionable (que se pueda implementar directamente).
```

## 5. Restricciones

Que NO debe hacer. Los límites son tan importantes como la tarea.

No menciones soluciones enterprise de más de 10.000 EUR.  
No uses jerga técnica que un empleado de administración no entienda.  
No incluyas requisitos que requieran infraestructura que no tenemos.

## 6. Ejemplos

Muestras del resultado que esperas. Es la técnica más infravalorada (la veremos en detalle en la sección de few-shot).

Ejemplo del estilo que busco:

- "Todas las contraseñas deben tener al menos 12 caracteres, combinando letras, números y símbolos."
- "Cambiar contraseñas cada 90 días. IT enviará recordatorio automático 7 días antes."

## 7. Evaluación

Como sabes si el resultado es bueno. Le das criterios para auto-evaluarse.

El texto final debe ser entendible por alguien sin perfil técnico.  
Si un punto requiere conocimiento de IT para implementarse, no lo incluyas.

### Regla práctica

Para tareas simples: **Tarea + Formato** basta. Para tareas complejas: añade **Rol + Contexto + Restricciones**. Cuando el resultado no sea lo que esperas: añade **Ejemplos + Evaluación**.

## Ejemplo completo con los 7 componentes

[ROL] Eres un analista de ciberseguridad senior especializado en normativa europea (NIS2, DORA, ENS).

[CONTEXTO] Trabajo en una empresa SaaS B2B de 80 empleados que procesa datos de salud. Estamos preparando la certificación ENS nivel alto.

[TAREA] Genera una lista de los 10 controles de seguridad más críticos que debemos implementar antes de la auditoría.

[FORMATO] Tabla con columnas: # | Control | Descripción (1 frase) | Prioridad (Alta/Media/Baja) | Plazo estimado

[RESTRICCIONES] Solo controles que una empresa de 80 personas puede implementar sin consultora externa. No incluir controles que requieran infraestructura on-premise (somos 100% cloud).

[EJEMPLOS]

#	Control	Descripción	Prioridad	Plazo
1	MFA obligatorio	Activar autenticación multifactor en todos los sistemas	Alta	2 semanas

[EVALUACION] Cada control debe ser implementable por nuestro equipo de IT (3 personas) en el plazo indicado. Si un control es demasiado complejo para 3 personas, divídelo en sub-tareas.

[IMG: Captura de pantalla del resultado de Claude con los 10 controles en formato tabla perfecta]

## Los 4 tipos de prompts

No todos los prompts se construyen igual. Hay 4 tipos fundamentales, cada uno para una situación diferente.

### Tipo 1: Zero-shot

Le pides algo sin dar ejemplos previos. Funciona para tareas simples y bien definidas.

Traduce este texto al inglés manteniendo el tono formal:  
[texto en español]

**Cuando usarlo:** tareas estándar donde el formato esperado es obvio.

## Tipo 2: Few-shot

Le das 2-3 ejemplos antes de la tarea real. La IA aprende el patrón de tus ejemplos y lo replica.

Clasifica estos emails de cliente. Aquí tienes 3 ejemplos:

Email: "Necesito la factura del mes pasado"  
Categoría: soporte-facturación | Prioridad: baja

Email: "El sistema se ha caído y no podemos trabajar"  
Categoría: incidencia-crítica | Prioridad: alta

Email: "Queremos ampliar a 20 licencias"  
Categoría: comercial-upsell | Prioridad: media

Ahora clasifica este:

Email: "Tenemos un audit de seguridad la semana que viene y necesitamos el informe de cumplimiento"

**Cuando usarlo:** cuando quieres que siga un formato o estilo específico.

## Tipo 3: Chain-of-thought

Le pides que razone paso a paso antes de dar la respuesta final. Mejora dramáticamente las tareas de lógica y análisis.

Antes de responder, sigue estos pasos:

1. Lee el documento completo e identifica los 3 temas principales.
2. Para cada tema, lista los riesgos asociados.
3. Clasifica los riesgos por probabilidad e impacto.
4. Solo entonces, recomienda las 3 acciones prioritarias.

Piensa paso a paso. Muestra tu razonamiento antes de la conclusión.

**Cuando usarlo:** tareas que requieren analisis, comparacion o razonamiento multi-paso.

## Tipo 4: System prompt

Instrucciones fijas que definen el comportamiento base. No cambia entre peticiones. Es el manual de instrucciones permanente.

```
[System]
Eres el asistente de soporte de Acme Corp. Reglas:
- Responde siempre en español formal.
- Si no sabes la respuesta, di: "Voy a escalar esto a un humano."
- Nunca inventes datos sobre el producto.
- Maximo 150 palabras por respuesta.
- Siempre termina con: "Puedo ayudarte con algo mas?"
```

**Cuando usarlo:** workflows recurrentes, chatbots, agentes con comportamiento consistente.

### Progresion practica

Empieza con **zero-shot**. Si el resultado no es lo que quieres, sube a **few-shot**. Si necesita razonamiento complejo, usa **chain-of-thought**. Para workflows repetitivos, crea un **system prompt**.

## Few-shot learning con ejemplos

El few-shot es la tecnica mas infravalorada del prompting. La mayoría no la usa y es la que mas mejora los resultados.

La idea es simple: en vez de describir lo que quieres, **muestraselo con ejemplos**. La IA aprende por patron, no por descripcion.

## Regla: 3 ejemplos diversos > 1 pagina de instrucciones

Tres ejemplos que cubran casos diferentes (facil, medio, edge case) ensena a la IA mejor que una pagina entera de reglas. Porque los ejemplos muestran implicitamente el formato, el nivel de detalle, las excepciones y el tono.

### Como disenar buenos ejemplos

1. **Variiedad:** tus ejemplos deben cubrir casos diferentes, no 3 veces lo mismo.
2. **Consistencia:** el formato de output debe ser identico en todos los ejemplos.
3. **Edge cases:** incluye al menos un ejemplo dificil o ambiguo para que la IA sepa como manejarlos.

### Ejemplo practico: clasificador de tickets

```

Clasifica tickets de soporte. Formato de salida exacto:
Categoria: [una de: bug, feature-request, pregunta, queja]
Urgencia: [alta, media, baja]
Resumen: [1 frase, max 15 palabras]

---
Ejemplo 1 (caso facil):
Ticket: "No puedo iniciar sesion desde ayer"
Categoria: bug
Urgencia: alta
Resumen: Usuario bloqueado, posible fallo de autenticacion

Ejemplo 2 (caso ambiguo):
Ticket: "Estaria bien poder exportar a Excel"
Categoria: feature-request
Urgencia: baja
Resumen: Solicitud de nueva funcionalidad de exportacion

Ejemplo 3 (caso dificil):
Ticket: "Esto es inaceptable, llevo 3 dias sin respuesta y el
sistema sigue fallando"
Categoria: queja
Urgencia: alta
Resumen: Cliente insatisfecho con tiempos de respuesta y estabilidad

---
Ahora clasifica este ticket:

```

Ticket: "El dashboard tarda 30 segundos en cargar desde la actualización del viernes"

[IMG: Resultado de Claude clasificando correctamente el ticket con el formato exacto de los ejemplos]

## Debug de prompts: por que falla y como arreglarlo

Tu prompt va a fallar. No es si, es cuando. Lo importante es saber diagnosticar por que y arreglarlo sistemáticamente.

### Los 5 fallos mas comunes

#### 1. Resultado demasiado generico

**Diagnostico:** falta contexto. La IA no sabe tu situacion y da una respuesta "para todos".

**Solucion:** anade Contexto y Restricciones. "Para una pyme de 50 empleados del sector salud, no para una multinacional."

#### 2. Formato incorrecto

**Diagnostico:** no especificaste el formato o lo especificaste de forma ambigua.

**Solucion:** da un ejemplo del formato exacto. Mejor aun, pon el esqueleto: "Usa esta estructura: ## Titulo / ### Seccion / - Bullet."

#### 3. Inventa datos (alucinacion)

**Diagnostico:** la IA rellena huecos con informacion inventada.

**Solucion:** anade "Si no estas seguro de un dato, dilo explicitamente. No inventes." Y verifica siempre los datos criticos.

#### 4. Se enrolla demasiado

**Diagnostico:** la IA tiende a dar contexto y explicaciones que no pediste.

**Solucion:** "Maximo 200 palabras. Sin introduccion. Sin conclusion. Ve al grano directamente."

## 5. No sigue instrucciones

**Diagnostico:** la IA presta mas atencion al principio y al final del prompt. Las instrucciones del medio se pierden.

**Solucion:** pon las instrucciones criticas al **principio Y al final** del prompt. Repite lo mas importante.

**Tip de debug avanzado:** Si un prompt falla, antes de reescribirlo entero, pregunta a la IA: "Que entendiste que te pedi? Resume mi peticion en 3 puntos." Si su resumen no coincide con lo que querias, el problema esta en como lo expresaste, no en la IA.

## Crear tu prompt library

Tu prompt library es tu coleccion de prompts que ya funcionan. Es la diferencia entre empezar de cero cada vez y tener un sistema reutilizable.

### Estructura recomendada

```
# Mi Prompt Library

## Categoria: Emails
### Email de seguimiento comercial
- Rol: Director comercial B2B
- Contexto: [variable: empresa, producto, ultima interaccion]
- Tarea: Escribir email de seguimiento post-demo
- Formato: Asunto + cuerpo (max 150 palabras) + CTA claro
- Restricciones: Sin "espero que estes bien", sin adjuntos
- Ultimo uso: 2026-05-10
- Nota: 9/10

## Categoria: Analisis
### Resumen ejecutivo de articulo
- Rol: Analista de inteligencia
- Contexto: [variable: sector, audiencia]
- Tarea: 3 bullets con las conclusiones clave
- Formato: Bullet points, max 100 palabras total
- Restricciones: Sin tecnicismos, incluir cifras
- Ultimo uso: 2026-05-08
- Nota: 8/10
```

```
## Categoría: Código
### Code review automático
- Rol: Senior developer (Python/TypeScript)
- Contexto: Proyecto FastAPI con Supabase
- Tarea: Revisar diff y listar bugs, mejoras, violaciones de estilo
- Formato: Tabla [Archivo | Issue | Severidad | Sugerencia]
- Restricciones: Solo issues reales, no nitpicking
- Último uso: 2026-05-12
- Nota: 9/10
```

Guarda este archivo donde quieras (Notion, Obsidian, un .md en tu repo). Cada vez que un prompt funciona bien, anadelo. En 2 semanas tendrás 10-15 prompts reutilizables.

## Ejercicio práctico

### Ejercicio M02: Construye tu prompt library

1. Crea un archivo `prompt-library.md` con la estructura de arriba.
2. Escribe 5 prompts para tareas que haces regularmente. Usa los 7 componentes.
3. Para cada prompt, pruébalo al menos 2 veces y ajústalo.
4. Clasifica cada prompt: zero-shot, few-shot, chain-of-thought, o system.
5. Para al menos 1 prompt, convierte un zero-shot en few-shot y compara resultados.

**Bonus:** Escribe un prompt que falle a propósito y debuggealo con los 5 métodos de esta sección. Documenta que fallo y como lo arreglaste.

## Conclusiones clave

### Key takeaways del M02

1. Un prompt profesional tiene hasta 7 componentes: Rol, Contexto, Tarea, Formato, Restricciones, Ejemplos, Evaluación.

2. 4 tipos: zero-shot (simple), few-shot (con ejemplos), chain-of-thought (razonamiento), system (comportamiento fijo).
3. 3 ejemplos diversos valen mas que 1 pagina de instrucciones. Few-shot es la tecnica mas infravalorada.
4. Los 5 fallos: generico (falta contexto), mal formato (falta ejemplo), alucinacion (falta restriccion), verboso (falta limite), ignora instrucciones (repetir al inicio y final).
5. Tu prompt library es tu activo mas valioso. Cada prompt que funciona, guardalo.

### **Siguiente: M03 - Tu primer workflow con Claude Code**

Ahora que dominas los prompts, el siguiente paso es darle a la IA acceso a tu sistema. Claude Code, instalacion, configuracion y tu primer agente.

[Ir al Modulo 03](#)

# IACADEMY

[iacedemy.com](https://iacedemy.com)

---

De fundamentos a arquitectura de IA.  
12 módulos prácticos. 24 recursos descargables.  
Quizzes con certificado. Vídeos profesionales.

Empieza gratis en [iacedemy.com/free](https://iacedemy.com/free)

© 2026 IAcademy — Todos los derechos reservados