



# IA ACADEMY

DE FUNDAMENTOS A ARQUITECTURA DE IA

---

MÓDULO 30

## Certificación IAcademy Pro

Avanzado

iacademy.com — 2026

MÓDULO 30

# Certificación IAcademy Pro

---

**Nivel:** Avanzado

**Autor:** David Moya

**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.

## Que cubre el examen

---

La certificación IAcademy Pro no es un examen tipo test. No hay preguntas de opción múltiple sobre definiciones teóricas. Es un examen 100% práctico que evalúa si puedes usar IA de forma profesional para resolver problemas reales.

El examen cubre los 30 módulos del curso, pero no de forma uniforme. Se enfoca en las habilidades que realmente importan en el mundo profesional:

- **Prompt engineering aplicado:** no solo escribir prompts, sino diseñar sistemas de prompts para producción.
- **Automatización e integración:** conectar IA con herramientas reales, APIs, bases de datos, flujos de trabajo.
- **Construcción de productos:** diseñar, construir y desplegar una solución funcional con IA.

El examen es asíncrono. Tienes 48 horas desde que lo empiezas para entregar los 3 bloques. Puedes usar cualquier herramienta, documentación o IA durante el examen (es open-book). Lo que se evalúa no es si memorizaste algo, sino si puedes ejecutar.

## Los 3 bloques del examen

---

### Bloque 1: Prompt Engineering Aplicado (30%)

Recibes un escenario de negocio real y debes diseñar un sistema de prompts completo. No un solo prompt: un sistema con system prompt, user prompt template, formato de output, manejo de edge cases, y validación.

```
# Ejemplo de lo que se pide en el Bloque 1:  
  
## Escenario  
Una empresa de e-commerce quiere un sistema que analice reseñas  
de productos y genere respuestas personalizadas automáticas.  
  
## Tu entregable:  
1. System prompt del agente de respuesta  
2. Template de user prompt (con variables)  
3. Formato de output (JSON schema)  
4. Manejo de edge cases:  
   - Resena con lenguaje ofensivo  
   - Resena en otro idioma  
   - Resena ambigua (positiva y negativa)  
5. Prompt de evaluación (para testear calidad de respuestas)  
6. 3 ejemplos de input/output que demuestren que funciona
```

## Bloque 2: Automatizacion e Integracion (30%)

Recibes un flujo de trabajo manual y debes automatizarlo usando IA + herramientas. El entregable es código funcional (no pseudocódigo) que conecta al menos 2 sistemas externos.

```
# Ejemplo de lo que se pide en el Bloque 2:

## Flujo manual actual
1. Cada mañana, el equipo revisa emails de soporte (Gmail)
2. Clasifican manualmente por prioridad (alta/media/baja)
3. Asignan al departamento correcto
4. Redactan una respuesta inicial

## Tu entregable:
- Script Python funcional que:
  1. Lee emails no leídos de una bandeja (API de Gmail o mock)
  2. Usa un LLM para clasificar prioridad y departamento
  3. Genera borrador de respuesta
  4. Guarda resultado en Supabase/JSON
- Tests que verifican clasificación correcta
- Documentación de cómo ejecutar
```

## Bloque 3: Build Challenge (40%)

El bloque más pesado. Recibes un brief de producto (similar a un mini-PRD) y tienes que construir un MVP funcional en las horas restantes. No tiene que ser bonito, pero sí funcional y desplegado.

```
# Ejemplo de brief del Bloque 3:

## Producto: Generador de Informes Semanales
Un freelancer quiere una herramienta que:
- Reciba sus notas diarias (texto libre)
- Genere un informe semanal estructurado para enviar al cliente
- Incluya: resumen ejecutivo, tareas completadas, bloqueantes, plan semana siguiente

## Requisitos mínimos:
- Input: formulario web o endpoint API
- Procesamiento: LLM con prompt estructurado
- Output: informe en formato legible (HTML o PDF)
- Persistencia: guardar informes generados
- Deploy: accesible vía URL pública

## Se evalúa:
- Funcionalidad (hace lo que promete)
- Uso efectivo de IA (no es un CRUD disfrazado)
- Calidad de código (legible, con tipos, sin secretos hardcoded)
- Documentación (README mínimo viable)
- Originalidad (tu enfoque al problema)
```

### Umbral de aprobacion

Necesitas **70% en CADA bloque** por separado. No puedes compensar un bloque malo con otro bueno. Si sacas 90% en Build Challenge pero 60% en Prompt Engineering, no apruebas. Esto garantiza que dominas las 3 areas.

## Requisito de portfolio

Ademas del examen, necesitas un portfolio con minimo 3 proyectos. Cada proyecto debe ser de un tipo diferente:

### Tipo A: Contenido o analisis con IA

Un proyecto donde la IA genera, transforma o analiza contenido. Ejemplos: generador de newsletters, analizador de sentimiento, resumidor de documentos, traductor especializado.

### Tipo B: Automatizacion o workflow

Un proyecto que automatiza un proceso real conectando multiples herramientas. Ejemplos: pipeline de procesamiento de datos, bot de Telegram/Slack, sistema de alertas inteligente, ETL con enriquecimiento via LLM.

### Tipo C: Producto o herramienta

Un producto funcional con interfaz de usuario (web, CLI o app). Ejemplos: SaaS minimo, herramienta de productividad, dashboard con insights generados por IA, marketplace de prompts.

### El proyecto del M29 cuenta

Si completaste el ejercicio del modulo 29, ya tienes uno de los 3 proyectos. Dependiendo de lo que construiste, clasificalo como Tipo A, B o C. Solo necesitas dos proyectos mas para completar el portfolio.

## Que debe tener cada proyecto del portfolio

- Repositorio publico en GitHub (o enlace a demo si es privado)
- README con las 5 secciones del M29 (que, para quien, como, demo, instalar)
- Al menos un test automatizado
- Deploy funcional (URL accesible)
- Uso demostrable de IA (no un CRUD con un prompt pegado)

## Criterios de evaluacion

Tanto el examen como el portfolio se evaluan con 5 criterios. Cada uno pesa un 20%:

CRITERIO	QUE SE EVALUA	EJEMPLO DE NOTA ALTA
<b>Funcionalidad</b>	Hace lo que promete, sin errores criticos	El flujo principal funciona end-to-end, maneja errores gracefully
<b>Uso efectivo de IA</b>	La IA aporta valor real, no es decorativa	El LLM resuelve algo que seria muy costoso hacer manual
<b>Calidad de codigo</b>	Legible, tipado, sin secretos, estructura clara	Otro dev puede leer y entender en 5 minutos
<b>Documentacion</b>	README util, instrucciones de instalacion, demo	Alguien puede clonar y ejecutar sin preguntar nada
<b>Originalidad</b>	Enfoque propio, no copia de tutorial	Resolver un problema real de tu contexto, no el tipico "chatbot"

## Rúbrica simplificada

```
# Niveles por criterio:
# 0-49%: No cumple (funcionalidad rota, sin IA real, sin docs)
# 50-69%: Basico (funciona parcialmente, IA generica, docs minimas)
# 70-84%: Competente (funciona bien, IA aporta, codigo limpio)
# 85-100%: Excelente (robusto, IA innovadora, produccion-ready)
```

## Preparacion recomendada

El examen no requiere estudio de memorizacion. Requiere practica. Estas son las formas mas efectivas de prepararte:

### 1. Revisa tus ejercicios de los modulos 1-28

Si hiciste los ejercicios de cada modulo, ya tienes la practica necesaria. Los bloques del examen son versiones mas completas de esos ejercicios. Revisa especialmente: M5 (prompt engineering), M11 (automatizacion), M22 (APIs), M29 (proyecto completo).

## 2. Practica el formato de entrega

```
# Antes del examen, practica entregar algo completo:
# 1. Repo limpio (sin archivos innecesarios)
# 2. README que explica sin ambigüedad
# 3. .env.example con todas las variables (sin valores reales)
# 4. Tests que pasan: pytest -v
# 5. Deploy accesible

# Checklist pre-entrega:
- [ ] git status limpio (no hay cambios uncommitted)
- [ ] README explica que hace en 1 frase
- [ ] .env.example existe y lista todas las variables
- [ ] pip install -r requirements.txt funciona en un entorno limpio
- [ ] pytest pasa sin errores
- [ ] La URL de demo funciona ahora mismo
```

## 3. Gestiona el tiempo

48 horas parece mucho, pero si trabajas los 3 bloques secuencialmente, el tiempo vuela.

Recomendación: Bloque 1 en las primeras 4-6 horas. Bloque 2 en las siguientes 6-8 horas. Bloque 3 con el resto del tiempo (necesita más porque incluye deploy).

## Después de la certificación

### Certificado digital verificable

Al aprobar recibes un certificado digital con código QR único. Cualquier persona puede escanear el QR o visitar la URL para verificar que tu certificado es auténtico, incluyendo la fecha de emisión y tu puntuación por bloques.

```
# Estructura del certificado digital:
{
  "holder": "Nombre Completo",
  "certification": "IAcademy Pro",
  "issued": "2026-06-15",
  "verification_url": "https://iacademy.com/verify/abc123",
  "scores": {
    "prompt_engineering": 82,
    "automation": 78,
    "build_challenge": 91
  },
  "portfolio_projects": 3,
  "valid": true
}
```

## Donde usarlo

- **LinkedIn:** anadirlo en la seccion de certificaciones con el enlace de verificacion.
- **CV:** seccion de certificaciones, con la URL de verificacion.
- **Portfolio personal:** badge en el footer o seccion dedicada.
- **Propuestas a clientes:** como evidencia de competencia tecnica con IA.

## La comunidad IAcademy

---

La certificacion no es el final. Es la puerta de entrada a una comunidad de profesionales que usan IA de forma seria.

### Que incluye la comunidad

- **Foro privado:** espacio para preguntas tecnicas, compartir proyectos, pedir feedback entre peers.
- **Meetups virtuales mensuales:** sesiones de 1 hora donde un miembro presenta un proyecto o tecnica. Formato lightning talk + Q&A.
- **Proyectos colaborativos:** cada trimestre, un proyecto open-source donde los miembros contribuyen. Buen material para portfolio.
- **Code reviews:** puedes pedir que otros miembros certificados revisen tu codigo o arquitectura.
- **Canal de oportunidades:** ofertas de trabajo, freelance y colaboracion compartidas por miembros.

### Reglas de la comunidad

1. Comparte codigo, no solo opiniones. Los mensajes con codigo/ejemplo concreto tienen prioridad.
2. Feedback constructivo. "Esto no funciona" no ayuda. "Esto falla porque X, prueba Y" si ayuda.
3. Respeta el tiempo de los demas. Antes de preguntar, muestra que intentaste resolver por tu cuenta.

## Recursos para seguir aprendiendo

---

El curso termina aqui, pero el aprendizaje no. Estos son los recursos que recomendamos para mantenerte actualizado:

### Documentacion oficial (siempre actualizada)

- **Anthropic docs:** [docs.anthropic.com](https://docs.anthropic.com) (Claude API, prompt engineering, best practices)

- **OpenAI cookbook:** [cookbook.openai.com](https://cookbook.openai.com) (ejemplos practicos, patterns de produccion)
- **Supabase docs:** [supabase.com/docs](https://supabase.com/docs) (auth, RLS, edge functions, realtime)
- **LangChain/LangGraph:** [python.langchain.com](https://python.langchain.com) (orquestracion de agentes, chains)

## Newsletters y blogs

- **Simon Willison's blog:** el mejor blog tecnico sobre LLMs en produccion.
- **The Batch (Andrew Ng):** resumen semanal de noticias de IA con contexto.
- **Latent Space:** podcast y newsletter sobre AI engineering.
- **IAcademy blog:** actualizaciones del curso, tutoriales nuevos, casos de uso.

## Practica continua

- Construye 1 micro-proyecto al mes (scope: 1 fin de semana).
- Contribuye a proyectos open-source que usen IA.
- Documenta lo que aprendes (blog personal, notas publicas).
- Ensenalo a otros: la mejor forma de consolidar conocimiento.

### La regla del 70/20/10

70% de tu aprendizaje viene de construir cosas. 20% de interactuar con otros (comunidad, code reviews, pair programming). 10% de consumir contenido (cursos, libros, newsletters).  
No inviertas esas proporciones.

## Conclusiones clave

---

### Key takeaways del M30

1. El examen tiene 3 bloques practicos (prompt engineering, automatizacion, build challenge). 48 horas, open-book. Necesitas 70% en CADA bloque.
2. Portfolio obligatorio: minimo 3 proyectos de tipos diferentes (contenido/analisis, automatizacion, producto). El proyecto del M29 cuenta como uno.
3. 5 criterios de evaluacion: funcionalidad, uso efectivo de IA, calidad de codigo, documentacion, originalidad. Cada uno pesa 20%.
4. Certificado digital verificable con QR. Usalo en LinkedIn, CV, portfolio y propuestas a clientes.
5. La comunidad IAcademy incluye foro, meetups mensuales, proyectos colaborativos y code reviews entre peers.
6. Sigue aprendiendo: documentacion oficial, newsletters tecnicas, y sobre todo construir 1 micro-proyecto al mes.
7. Regla 70/20/10: 70% construir, 20% comunidad, 10% consumir contenido.