IACADEMY

AGENTIC ENGINEER & SDD

MÓDULO 08

Proyecto Final

Avanzado

iacedemy.com — 2026

MÓDULO 08

Proyecto Final

Nivel: Avanzado

Autor: Alicia Sanz

Publicación: Julio 2026

Plataforma: iacedemy.com

Este material es parte del curso Agentic Engineer & SDD de IAcademy.
Uso personal e intransferible. Queda prohibida su redistribución o reproducción sin autorización.

© 2026 IAcademy — iacedemy.com — Material con licencia personal e intransferible

> “Construye un agente de principio a fin usando todo lo aprendido. Sin atajos.”

8.1 Descripcion del Proyecto

El Reto

Construir un **agente de soporte tecnico completo** usando Spec-Driven Development. El agente debe:

1. Responder preguntas frecuentes usando una base de conocimiento
2. Escalar a humano cuando no puede resolver
3. Resistir ataques de prompt injection
4. Estar evaluado con CI gates
5. Estar instrumentado con observabilidad
6. Estar listo para deploy en produccion

© 2026 IAcademy — iacedemy.com — Material con licencia personal e intransferible

Entregables

```
1. SPEC (30% de la nota)
   - PRD con criterios de exito numericos
   - ADR con decisiónes y alternativas descartadas
   - System Spec YAML completa
   - CHANGELOG.md

2. EVAL DATASET (25% de la nota)
   - >= 30 golden cases (5 por behavior)
   - >= 10 edge cases
   - >= 10 adversarial cases
   - >= 5 regression cases (simulados)
   - Eval runner funcional
   - CI gate configurado

3. IMPLEMENTACION (25% de la nota)
   - Agente funcional con 3+ tools
   - Tool registry con metadata (risk_level, timeout, etc.)
   - Circuit breakers
   - Memory management (session-level minimo)
   - HITL para acciones de alto riesgo

4. OBSERVABILIDAD (10% de la nota)
   - Trazas OpenTelemetry (o equivalente)
   - Metricas exportadas
   - Dashboard basico

5. DOCUMENTACION (10% de la nota)
   - README con instrucciones de setup y ejecucion
   - Runbook para incidentes
   - Plan de rollout
```

8.2 Opcion A: Agente de Soporte Tecnico

Contexto

Empresa SaaS con 50 paginas de documentacion. El 70% de los tickets de soporte son preguntas que estan respondidas en la documentacion. El agente debe resolver esas preguntas automaticamente.

© 2026 IAcademy — iacedemy.com — Material con licencia personal e intransferible

Requisitos Funcionales

```
BEHAVIORS:

1. answer_faq
   - Buscar en KB y responder con fuente
   - Max 3 tool calls
   - Si no encuentra, pedir reformulacion

2. handle_account_query
   - Consultar estado de cuenta del usuario
   - Responder con datos del usuario (nombre, plan, fecha expiracion)
   - NUNCA revelar datos de otros usuarios

3. handle_escalation
   - Si usuario pide humano: escalar SIEMPRE
   - Si 2+ intentos fallidos: escalar automaticamente
   - Contexto recopilado en el summary de escalacion

4. handle_billing
   - Preguntas sobre pricing: responder desde KB
   - Cambios de plan o cancelaciones: escalar SIEMPRE

TOOLS:
  - search_knowledge_base(query, category, top_k)
  - get_user_account(user_id)
  - escalate_to_human(reason, summary, priority)
  - log_interaction(session_id, resolution_type)

GUARDRAILS:
  - Input: max 2000 chars, sanitizar HTML, detectar injection
  - Output: no PII de otros usuarios, no URLs internas, no precios
internos
  - Operational: max 10 iterations, 50k token budget, circuit breaker 5
fallos
```

© 2026 IAcademy — iacedemy.com — Material con licencia personal e intransferible

Base de Conocimiento Simulada

© 2026 IAcademy — iacedemy.com — Material con licencia personal e intransferible

```python
# Para el proyecto, usar una KB simulada con 20-30 articulos.
# No necesitas un vector DB real, una lista de diccionarios basta.

KNOWLEDGE_BASE = [
    {
        "id": "kb-001",
        "title": "Cómo resetear password",
        "category": "account",
        "content": """
        Para resetear tu password:
        1. Ve a Configuracion > Seguridad
        2. Clic en 'Cambiar contrasena'
        3. Introduce tu contrasena actual
        4. Introduce la nueva contrasena (min 8 chars, 1 mayuscula, 1
numero)
        5. Confirma con el codigo enviado a tu email

        Si no recuerdas tu contrasena actual, usa 'Olvide mi contrasena'
en la pantalla de login.
        """,
    },
    {
        "id": "kb-002",
        "title": "Planes y precios",
        "category": "billing",
        "content": """
        Planes disponibles:
        - Free: 100 queries/mes, 1 usuario, soporte email
        - Pro: 10.000 queries/mes, 5 usuarios, soporte prioritario. 29
EUR/mes.
        - Enterprise: ilimitado, usuarios ilimitados, soporte dedicado.
Contactar ventas.

        Todos los planes incluyen 14 dias de prueba gratuita.
        """,
    },
    # ... 20-30 articulos más cubriendo FAQ tipicas
]

def search_kb(query: str, category: str = "all", top_k: int = 5) ->
list[dict]:
    """Busqueda simple por keyword matching. En produccion seria vector
search."""
    results = []
    query_lower = query.lower()

    for article in KNOWLEDGE_BASE:
        if category != "all" and article["category"] != category:
            continue
```

© 2026 IAcademy — iacedemy.com — Material con licencia personal e intransferible

```
        # Score simple basado en keywords compartidos
        content_lower = article["content"].lower() + " " + article["title
"].lower()
        query_words = set(query_lower.split())
        content_words = set(content_lower.split())
        overlap = len(query_words & content_words)
        score = overlap / max(len(query_words), 1)

        if score > 0.1:
            results.append({**article, "score": score})

    results.sort(key=lambda x: x["score"], reverse=True)
    return results[:top_k]
```

8.3 Opcion B: Agente de Triage de Alertas

Contexto

SOC (Security Operations Center) que recibe ~200 alertas diarias. El 60% son falsos positivos. El agente debe clasificar alertas, enriquecer con contexto, y priorizar para los analistas.

© 2026 IAcademy — iacedemy.com — Material con licencia personal e intransferible

Requisitos Funcionales

```
BEHAVIORS:

1. classify_alert
   - Clasificar severidad: info, low, medium, high, critical
   - Detectar falsos positivos conocidos
   - Asignar categoria: malware, phishing, brute_force, data_exfil, other

2. enrich_alert
   - Buscar IOCs (IPs, hashes, dominios) en threat intel
   - Correlacionar con alertas previas del mismo source
   - Anadir contexto del asset afectado

3. prioritize
   - Score de prioridad basado en: severidad + asset value + recurrencia
   - Top 10% alertas = notificar analista inmediatamente
   - Bottom 50% = auto-close si match con patron de falso positivo

4. create_incident
   - Si alert es critical o high + correlacion con otras alertas
   - Crear incident con timeline y IOCs
   - Asignar a analista de guardia

TOOLS:
  - query_siem(alert_id) -> detalles de la alerta
  - search_threat_intel(ioc, type) -> reputacion del IOC
  - get_asset_info(ip_or_hostname) -> valor y owner del asset
  - get_related_alerts(source_ip, timeframe) -> alertas correlacionadas
  - create_incident(title, severity, iocs, timeline)
  - notify_analyst(analyst_id, message, priority)
```

8.4 Opcion C: Tu Propio Agente

Requisitos Minimos

Si prefieres construir tu propio agente (tema libre), debe cumplir:

© 2026 IAcademy — iacedemy.com — Material con licencia personal e intransferible

```
OBLIGATORIO:
  - 3+ tools con risk_levels variados (al menos 1 LOW y 1 MEDIUM+)
  - 3+ behaviors definidos en la spec
  - Eval dataset >= 30 casos
  - CI gate funcional
  - Guardrails (input + output + operational)
  - Circuit breakers

OPCIONAL (extra credit):
  - Multi-agent (coordinator + workers)
  - LLM-as-judge en evals
  - Feature flag para rollout
  - Prompt regression testing
  - Dashboard de observabilidad
```

Ideas

```
- Agente de onboarding (guia nuevos usuarios paso a paso)
- Agente de code review (lint + seguridad + style)
- Agente de scheduling (coordinar disponibilidad y agendar)
- Agente de data analysis (consultar DB, generar insights)
- Agente de content moderation (clasificar y escalar contenido)
- Agente de inventory management (stock, reorder, alertas)
```

© 2026 IAcademy — iacedemy.com — Material con licencia personal e intransferible

8.5 Estructura del Proyecto

```
mi-agente/
├── README.md                    # Setup, ejecucion, arquitectura
├── docs/
│   ├── PRD.md                   # Product Requirements Document
│   ├── ADR-001.md               # Architecture Decision Record
│   └── RUNBOOK.md               # Incident response
├── specs/
│   ├── system-spec-v1.0.yaml    # System Spec completa
│   └── CHANGELOG.md
├── prompts/
│   └── system-prompt-v1.0.txt   # Generado desde spec
├── src/
│   ├── agent.py                 # Agente principal
│   ├── tools/
│   │   ├── registry.py          # Tool registry
│   │   ├── search_kb.py
│   │   ├── get_account.py
│   │   └── escalate.py
│   ├── guardrails.py            # Input/output validation
│   ├── circuit_breaker.py
│   └── telemetry.py             # OpenTelemetry setup
├── tests/
│   ├── evals/
│   │   ├── golden.jsonl
│   │   ├── edge_cases.jsonl
│   │   ├── adversarial.jsonl
│   │   ├── regression.jsonl
│   │   └── runner.py            # Eval runner
│   └── unit/
│       ├── test_tools.py
│       └── test_guardrails.py
├── .github/
│   └── workflows/
│       └── agent-eval.yml       # CI gate
├── docker-compose.yml           # Para ejecutar localmente
└── requirements.txt
```

© 2026 IAcademy — iacedemy.com — Material con licencia personal e intransferible

8.6 Rubrica de Evaluacion

Spec (30 puntos)

Criterio	Puntos	Descripcion
PRD completo	5	Problema, objetivo, usuarios, criterios de exito numericos, fuera de alcance
ADR con trade-offs	5	Decision, alternativas descartadas con razon, consecuencias
System Spec behaviors	10	Cada behavior tiene trigger, steps, constraints. Cubre todos los casos
System Spec tools	5	Cada tool tiene risk_level, timeout, description orientada al LLM
System Spec guardrails	5	Input, output, operational. Concretos y testeables

Eval Dataset (25 puntos)

Criterio	Puntos	Descripcion
Golden cases	8	>= 5 por behavior, formulaciones variadas
Edge cases	5	Inputs vacios, largos, ambiguos, multilingues
Adversarial	7	Injection, data exfil, role confusion. >= 10 cases
Eval runner	3	Ejecuta evals, genera reporte, calcula pass rate
CI gate	2	GitHub Action (o script) que bloquea si < threshold

© 2026 IAcademy — iacedemy.com — Material con licencia personal e intransferible

Implementacion (25 puntos)

Criterio	Puntos	Descripcion
Agente funcional	8	Corre, responde, usa tools correctamente
Tool registry	4	Metadata, timeout, error handling estandar
Circuit breakers	3	En todas las tools externas
Guardrails	5	Input validation, output sanitization, injection detection
HITL	3	Para al menos 1 accion de riesgo
Memory	2	Session-level minimo (no perder contexto entre turnos)

Observabilidad (10 puntos)

Criterio	Puntos	Descripcion
Trazas	4	Spans para LLM calls y tool calls
Metricas	3	Tokens, latencia, error rate
Alertas	3	Al menos 2 reglas de alerta definidas

Documentacion (10 puntos)

Criterio	Puntos	Descripcion
README	4	Setup, ejecucion, arquitectura (diagrama)
Runbook	3	Al menos 2 escenarios de incidente
Rollout plan	3	Fases, criterios go/no-go, rollback

© 2026 IAcademy — iacedemy.com — Material con licencia personal e intransferible

8.7 Proceso Recomendado

Semana 1: Spec + Eval

```
Dia 1: Elegir opcion (A, B, o C) y escribir PRD
Dia 2: Escribir ADR (decisiónes tecnicas)
Dia 3: Escribir System Spec v1.0 completa
Dia 4: Derivar eval dataset (golden + edge + adversarial)
Dia 5: Implementar eval runner basico
```

Semana 2: Build

```
Dia 6:  Implementar tools + tool registry
Dia 7:  Implementar agente (loop basico)
Dia 8:  Implementar guardrails
Dia 9:  Correr evals, iterar prompt/spec
Dia 10: Circuit breakers + HITL
```

Semana 3: Harden + Ship

```
Dia 11: Observabilidad (trazas + metricas)
Dia 12: CI gate (GitHub Action o script)
Dia 13: Adversarial testing, fix vulnerabilidades
Dia 14: Documentacion (README, runbook, rollout plan)
Dia 15: Presentacion y entrega
```

© 2026 IAcademy — iacedemy.com — Material con licencia personal e intransferible

Criterios de Exito Personal

```
MINIMO VIABLE (aprobado):
  - Spec escrita y coherente
  - Agente funciona (responde, usa tools)
  - >= 20 eval cases
  - Guardrails basicos

NOTABLE:
  - Spec completa con todos los campos
  - >= 40 eval cases con runner
  - CI gate funcional
  - Circuit breakers
  - Observabilidad basica

EXCELENTE:
  - Todo lo anterior
  - LLM-as-judge en evals
  - Prompt regression testing
  - Adversarial resistance >= 95%
  - Dashboard de metricas
  - Runbook detallado
```

8.8 FAQ del Proyecto

Que LLM usar?

```
Cualquiera que soporte function calling:
  - OpenAI (GPT-4o-mini para coste bajo)
  - Anthropic (Haiku para rapido, Sonnet para calidad)
  - Ollama local (Qwen2.5 7B, gratis)
  - vLLM self-hosted (si tienes GPU)

Para evals, usa un modelo DIFERENTE como juez.
```

© 2026 IAcademy — iacedemy.com — Material con licencia personal e intransferible

Necesito un vector DB real?

```
NO para el proyecto. Una búsqueda por keywords basta.
El objetivo es demostrar SDD, no infra de RAG.

Si quieres extra credit: usa Qdrant o ChromaDB local (Docker).
```

Puedo usar LangGraph/CrewAI?

```
SI, pero no es obligatorio.
Un agente con loop manual (while + tool calls) es valido.
El framework no suma puntos; la spec y los evals si.
```

Cuanto deberia tardar?

```
Dedicacion estimada: 15-25 horas total.
  - Spec + eval: 6-8h
  - Build: 5-8h
  - Harden: 3-5h
  - Docs: 2-3h

El 60% del tiempo deberia ser spec + evals.
Si inviertes < 40% en spec, probablemente estas haciendo vibe coding.
```

© 2026 IAcademy — iacedemy.com — Material con licencia personal e intransferible

8.9 Ejemplo: Esqueleto del Proyecto

© 2026 IAcademy — iacedemy.com — Material con licencia personal e intransferible

```python
# src/agent.py — Esqueleto minimo para arrancar

import json
from tools.registry import ToolRegistry
from guardrails import check_input, check_output
from circuit_breaker import CircuitBreaker
from telemetry import tracer

class SupportAgent:
    def __init__(self, llm, registry: ToolRegistry, system_prompt: str):
        self.llm = llm
        self.registry = registry
        self.system_prompt = system_prompt
        self.max_iterations = 10
        self.token_budget = 50_000

    async def run(self, user_input: str, session_id: str = None) -> dict:
        with tracer.start_as_current_span("agent.run") as span:
            # Guardrail: input
            input_check = check_input(user_input)
            if not input_check.passed:
                return {"response": input_check.message, "blocked": True}

            messages = [
                {"role": "system", "content": self.system_prompt},
                {"role": "user", "content": user_input},
            ]

            tokens_used = 0

            for iteration in range(self.max_iterations):
                # Budget check
                if tokens_used > self.token_budget:
                    return {"response": "Limite de procesamiento
alcanzado.", "budget_exceeded": True}

                # LLM call
                with tracer.start_as_current_span("llm.call"):
                    response = await self.llm.complete(
                        messages=messages,
                        tools=self.registry.get_schemas()
                    )
                    tokens_used += response.total_tokens

                # Sin tool calls = respuesta final
                if not response.tool_calls:
                    # Guardrail: output
                    output_check = check_output(response.text)
                    if not output_check.passed:
```

© 2026 IAcademy — iacedemy.com — Material con licencia personal e intransferible

```
                        return {"response": "No puedo procesar esa
solicitud.", "guardrail": True}

                    return {"response": response.text, "iterations":
iteration + 1, "tokens": tokens_used}

                # Ejecutar tool calls
                for tc in response.tool_calls:
                    with tracer.start_as_current_span("tool.call") as
tool_span:
                        tool_span.set_attribute("tool.name", tc.name)
                        result = await self.registry.execute(tc.name,
tc.arguments)

                        messages.append({
                            "role": "tool",
                            "tool_call_id": tc.id,
                            "content": result.to_llm_string()
                        })

            return {"response": "No pude resolver tu pregunta.
Transfiriendo a soporte.", "max_iterations": True}
```

© 2026 IAcademy — iacedemy.com — Material con licencia personal e intransferible