Inteligencia Artificial Aplicada a la Arquitectura

Transformando el diseño arquitectónico con inteligencia artificial

Día 3:
Agentes IA

La Evolución de la IA: Hacia los Agentes Inteligentes

Del aprendizaje automático básico a sistemas autónomos capaces de percibir, razonar y ejecutar flujos de trabajo complejos.

Cronología de la IA

1956

Se acuña formalmente el término "Inteligencia Artificial" en Dartmouth.

1997

La IA simbólica triunfa: Deep Blue de IBM vence al campeón mundial de ajedrez.

2020

Revolución de los LLMs. La IA generativa alcanza capacidades casi humanas en texto.

2024+

Era de los Agentes. Sistemas que no solo responden, sino que actúan y resuelven problemas.

Crecimiento Exponencial

La capacidad de cómputo y el tamaño de los parámetros han crecido a un ritmo vertiginoso. Lo que en 2020 parecía magia con GPT-3, hoy es el motor base para herramientas orquestadas y sistemas multi-agente avanzados.

La Era de los Agentes

¿Qué es un Agente IA?

No es solo un Chatbot

Un LLM estándar espera tus instrucciones, responde, y se detiene. Es puramente reactivo y depende enteramente de la guía paso a paso del usuario.

Autonomía y Ejecución

Un agente es un sistema impulsado por IA que tiene un objetivo. Puede dividir tareas, buscar información en la web, usar herramientas externas (como ejecutar código), evaluar resultados y autocorregirse.

Componentes Clave

Percepción

El agente interpreta su entorno. Lee documentos, scrapea sitios web, revisa bases de datos corporativas o procesa entradas de sensores en tiempo real.

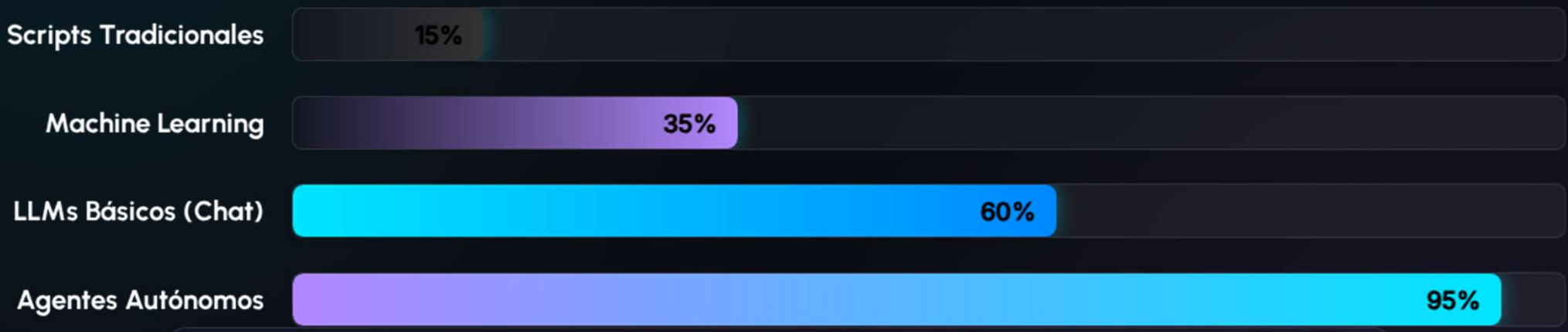
Decisión

El "cerebro" (usualmente un LLM avanzado) razona sobre los datos obtenidos. Planea estrategias, divide la tarea en subtareas y decide qué herramienta usar.

Acción

Ejecuta el plan. Llama a APIs, escribe código en un intérprete Python, envía correos o interactúa con el software operativo para lograr la meta.

Autonomía y Capacidad



El gráfico ilustra cómo pasamos de la automatización rígida (scripts) a una autonomía dinámica. Los Agentes IA actuales pueden manejar un 95% de la carga cognitiva en flujos de trabajo cerrados.

Sistemas Multi-Agente

Arquitecturas Jerárquicas

Para problemas complejos, un solo agente falla. Se requiere un modelo de "Supervisor" y "Trabajadores".

Delegación de Tareas

El Supervisor recibe el objetivo global, elabora un plan y delega tareas a agentes especializados (ej. un agente investigador, un agente programador).

Colaboración en Tiempo Real

Los agentes revisan el trabajo de los demás, comparten hallazgos y corrigen errores colectivamente.

¿Qué es realmente un Agente?

Cerebro (LLM)

El núcleo de razonamiento (Claude, GPT, Llama). Procesa lenguaje y planea la estrategia del diseño general.

Herramientas

Capacidad para usar software: API, scripts de Python, búsqueda .

Memoria

Contexto persistente del proyecto. Recuerda normativas previas y decisiones de diseño del usuario.

El Ciclo de Trabajo Agéntico

1. Objetivo

El docente define una meta: "Optimiza la fachada para reducir carga térmica".

2. Planificación

El agente desglosa pasos: buscar datos de clima, abrir Grasshopper, ejecutar script.

3. Acción

Uso de herramientas reales. El agente ejecuta código y observa el resultado técnico.

4. Refinamiento

Si el resultado no es óptimo, el agente itera y corrige sin intervención humana.

¿Nube o Local? Decisión Estratégica

Criterio	Nube	Local
Inversión	Pago por uso (SaaS). Muy económico para empezar pruebas.	Hardware dedicado (GPUs potentes). Alto coste inicial.
Privacidad	Los datos viajan por la red (Sujeto a las políticas de la API).	Máxima soberanía. Ideal para concursos secretos y academia.
Modelos	Acceso a modelos punteros y pesados (GPT-4o, Claude 3.5).	Modelos abiertos más ligeros (Llama 3, Mistral) optimizados.
Velocidad	Depende de la conexión y carga de red del proveedor.	Determinista, constante y con latencia de red nula.

ejercicio 1:
instalación

Configuración de Agentes

Arquitectura del Agente

La conexión requiere tres pilares fundamentales:

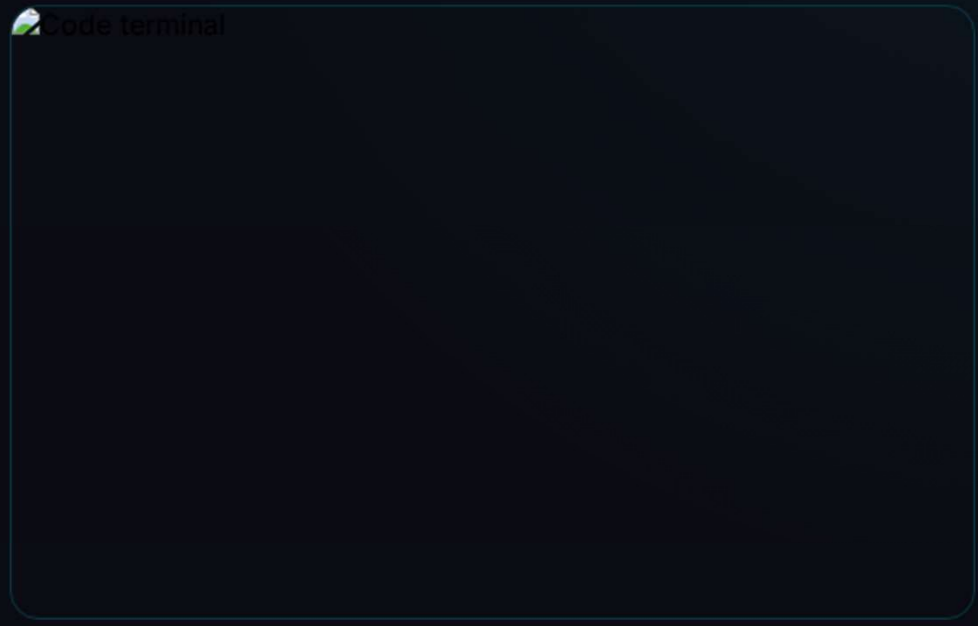
- **Groq:** Motor de inferencia para el razonamiento lógico.
- **Telegram:** Interfaz de control móvil y notificaciones.
- **Gmail:** Gestión documental y comunicación asíncrona.

Paso 1: Configurar Groq

API Key: Regístrate en console.groq.com.

Variable: Exporta tu llave: export GROQ_API_KEY='gsk_...'.

Testing: Realiza una petición cURL básica para confirmar la conexión del modelo (Llama-3).



Paso 2: Bot de Telegram

BotFather: Escribe /newbot en Telegram.

Tokens: Guarda el API Token proporcionado.

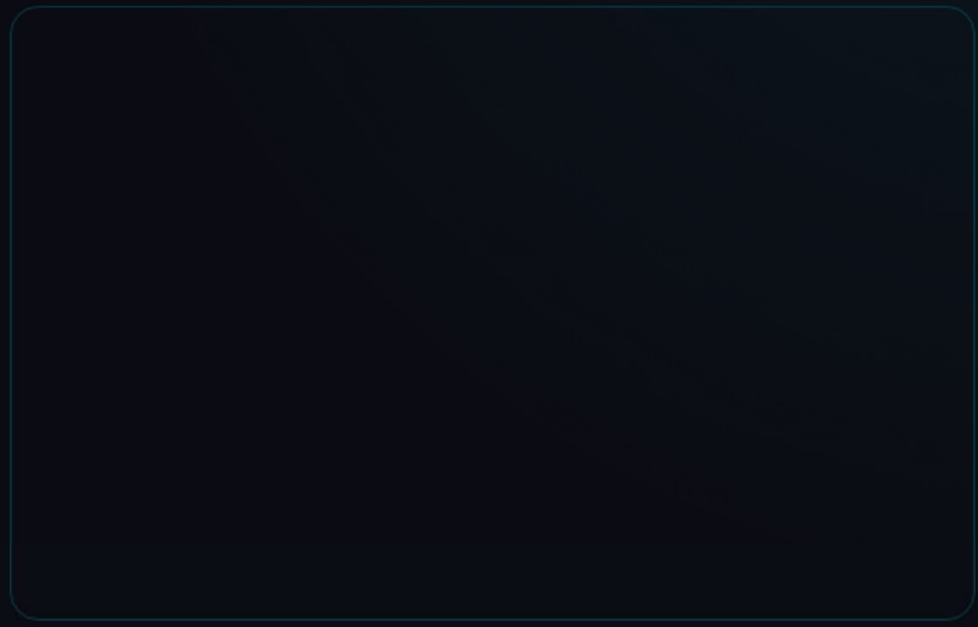
Webhook: Configura el endpoint de escucha en tu servidor local (o usando ngrok para desarrollo).

Paso 3: Gmail API

Cloud Console: Crea un nuevo proyecto y activa la API de Gmail.

Credenciales: Genera un archivo credentials.json (OAuth 2.0).

Scope: Define los permisos (ej. leer correos o enviar borradores).



Resumen de Conectividad

Servicio	Método	Key Principal
Groq	REST API	GROQ_API_KEY
Telegram	Bot API	BOT_TOKEN
Gmail	OAuth 2.0	credentials.json

Instalación de Agentes IA

Guía paso a paso desde la terminal para arquitectos e investigadores.

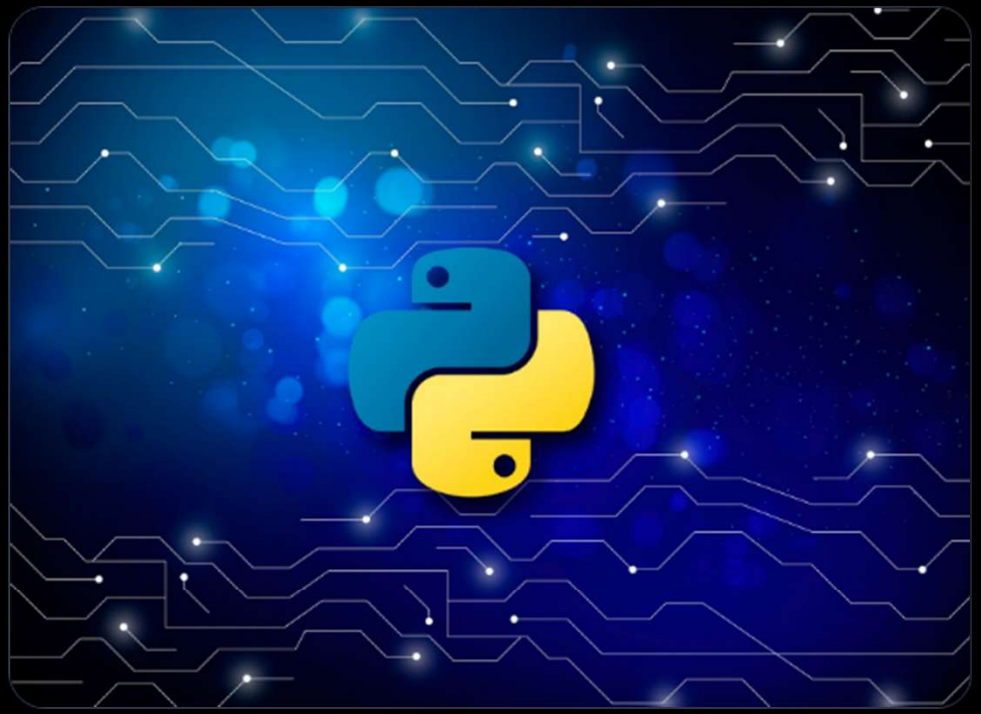
Paso 1: Abrir la Terminal (CMD)

La terminal es el centro de control donde daremos vida a nuestros agentes.

Método A: Pulsa la tecla Windows, escribe **"CMD"** y pulsa Enter.

Método B: Atajo Win + R, escribe **"cmd"** y pulsa Enter.

Paso 2: Instalación de Python



1. Descarga Python desde python.org (versión 3.10+).

¡CRÍTICO!

Durante la instalación, marca la casilla:

> [X] Add Python to PATH

Sin esto, el CMD no sabrá qué es "python".

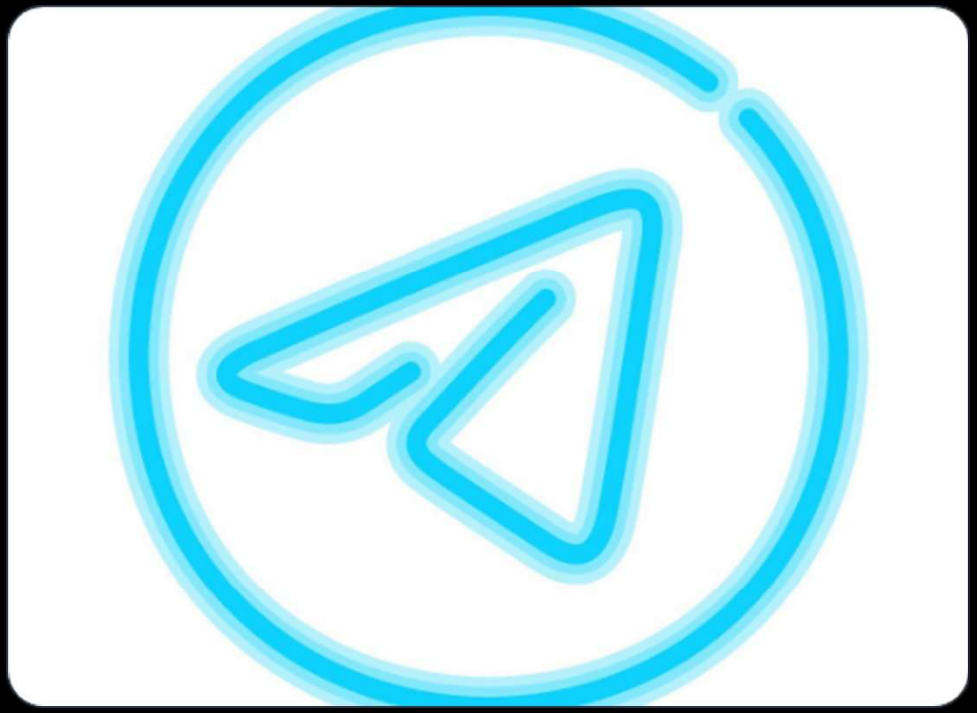
Paso 3: Verificación del Sistema

Abre el CMD y escribe estos comandos para confirmar que todo está listo:

```
C:\  python --version
# Debería devolver: Python
3.10.x
C:\  pip --version
# El gestor de paquetes está
activo
```

Si sale error, reinicia el CMD o revisa el PATH de la instalación anterior.

Paso 5: Conexión con Telegram



Necesitamos la librería python-telegram-bot para que el agente pueda chatear.

```
c:\  pip install python-telegram-bot
>
```

Recuerda: Debes hablar con @BotFather en Telegram para obtener tu **Token de Bot**.

Paso 6: Configurar las Llaves (API)

Para no exponer tus llaves en el código, las guardamos en el sistema:

```
C:\  setx GROQ_API_KEY "tu_llave_de_groq"
S:\  setx TELEGRAM_TOKEN "tu_token_de_telegram"
>
```

⚠ **IMPORTANTE:** Cierra y vuelve a abrir el CMD para que estas llaves se activen.

Código Base: Importación

Crea un archivo llamado agente_arq.py y pega el inicio:

```
impor  os
fro  groq impor  Groq
fro  telegram impor  Update
fro  telegram.ext impor  ApplicationBuilder, CommandHandler, MessageHandler, filters
# Inicializamos
client = Groq(api_key=os.environ.get("GROQ_API_KEY ))
                                     "
```

Código Base: Lógica y Respuesta

```
async def responder(update, context):
    user_msg = update.message.text
    # Consulta a Groq
    completion = client.chat.completions.create(
        model="llama3-70b-8192",
        messages=[{"role": "user", "content": user_msg}]
    )
    respuesta = completion.choices[0].message.content
    await update.message.reply_text(respuesta)
```

Este bloque recibe el mensaje de Telegram, lo envía a Groq y devuelve la respuesta.

Código Base: Main Loop

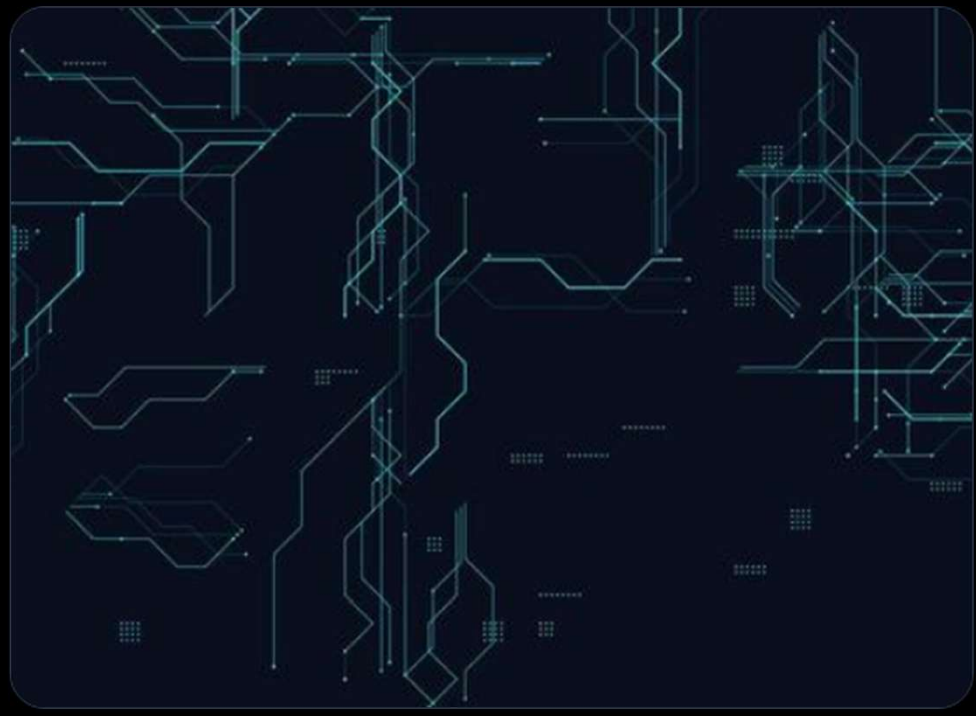
```
i  __name__ == '__main__ :
f   token = os.environ.get("TELEGRAM_TOKEN )
    app = ApplicationBuilder().token(token).build()

    app.add_handler(MessageHandler(filters.TEXT & (~filters.COMMAND), responder))

    prin ("Agente Arquitecto en          )
    app.run_polling()
```

Paso Final: Ejecución

Para poner en marcha tu agente, navega en el CMD hasta la carpeta donde guardaste el archivo y escribe:

```
C:\IA_Proyectos  python agente_arq.py
>
```

¡Listo! Ahora ve a Telegram, escribe a tu bot y Groq responderá al instante.



ejercicio 2:
Normativa CTE

Crea un agente conversacional capaz de responder preguntas sobre el Documento Básico HS Salubridad del CTE a partir de un PDFs subidos. Debe citar siempre el artículo y la página de referencia.

Eres un arquitecto técnico especializado en normativa del CTE.
Tu trabajo es interpretar EXCLUSIVAMENTE el contenido del PDF que acabo de subir.
REGLAS:
1. Nunca inventes artículos ni páginas que no existan en el PDF.
2. Cita siempre el apartado y, si es posible, la página de la referencia.
3. Si la respuesta requiere datos técnicos (áreas, alturas, materiales), usa los valores numéricos del documento.
4. Si una pregunta excede el alcance del documento, indícalo claramente.
5. Responde en español, con tono profesional y directo.

ejercicio 3:
Memoria Proyecto

Diseña un prompt que genere una memoria descriptiva completa de un proyecto

Actúa como redactor técnico de proyectos de arquitectura.
Debes generar una MEMORIA DESCRIPTIVA completa de una edificación con los siguientes datos:

- TIPOLOGÍA: [Vivienda unifamiliar / Bloque de viviendas / Oficinas / Industrial]
- LOCALIZACIÓN: [Ciudad/Comunidad Autónoma]
- SUPERFICIE CONSTRUIDA: [XXX m²]
- NÚMERO DE PLANTAS: [X]
- ALTURA TOTAL: [XX m]
- USO PRINCIPAL: [Residencial / Terciario / Industrial]
- SISTEMA CONSTRUCTIVO PREDOMINANTE: [Placa alveolar + muro de carga / Pórticos metálicos / Muro pantalla + losa plana]
- SISTEMA DE CLIMATIZACIÓN: [Aerotermia + suelo radiante / Gas Natural / Biomasa / Sin definir]
- CERTIFICACIÓN ENERGÉTICA OBJETIVO: [A / B / C]

ESTRUCTURA OBLIGATORIA DE LA RESPUESTA:
1. Antecedentes y objeto del proyecto
2. Descripción general de la edificación
3. Cimentación y estructura
4. Cerramientos y cubiertas (mencionar resistencia térmica según DB-HE)
5. Instalaciones (aguas, saneamiento, energía, climatización siguiendo RITE)
6. Instalaciones de protección contra incendios (si aplica DB-SI)
7. Accesibilidad y supresión de barreras arquitectónicas
8. Sostenibilidad y eficiencia energética
9. Relación de materiales principales

REGLAS DE REDACCIÓN:
- Usa vocabulario técnico del CTE sin simplificar.
- Indica por cada apartado qué DB aplica.
- Menciona valores numéricos concretos cuando sea posible (U-values, espesores).
- Escribe en español neutro, presente descriptivo.
- Extensión aproximada: 1000 palabras.

ejercicio 4:
material para curso

12 temas de "Patología de la Construcción" en un cuatrimestre de 14 semanas, respetando 1 semana de presentación, 1 semana de revisión y 2 semanas de exámenes finales.

Actúa como un profesor para el Grado en Arquitectura.
CONTEXTO:
- Asignatura: Patología de la Construcción.
- Duración: 14 semanas (cuatrimestre).
- Semana 1: presentación del curso (sin nuevo temario).
- Semana 14: examen final (sin nuevo temario).
- Semana 13: revisión y resolución de dudas.
- Total de horas presenciales: 4 h/semana × 14 = 56 h.
- Horas de estudio autónomo recomendado: 4-6 h/semana.
- No más de 2 temas de alta densidad consecutivos.
- Debe haber una sesión de laboratorio de diagnóstico (2h) tras el tema 6.
- Entregable final: informe de patología aplicada a un edificio real (semana 12).

TEMAS:
1. Introducción a la patología constructiva (3h autónomas)
2. Mecanismos de degradación en hormigón (5h)
3. Patologías en estructuras metálicas (4h)
4. Cimentaciones: hundimientos y asientos (6h)
5. Patologías en madera y elementos mixtos (5h)
6. Patologías de cerramientos y envolvente (4h)
7. Humedades y patologías higrotérmicas (5h)
8. Sistemas de impermeabilización: fallos y reparación (4h)
9. Intervención y refuerzo de estructuras (6h)
10. Normativa y procedimientos de inspección (3h)
11. Estudios de caso locales (4h)
12. Técnicas no destructivas de evaluación (5h)

INSTRUCCIÓN:
Distribuye los temas en las 14 semanas de forma coherente.
Devuélvelo en tabla: Semana | Temas | H. presenciales | H. autónomas | Actividad / Entregable.
Indica la carga semanal de autonomía. Si una semana supera 6 horas, redistribuye.

ejercicio 5:
ejercicio interactivo

Haz que el agente cree un ejercicio interactivo: cálculo de la flecha de una viga biapoyada

Calculadora de Flecha - Viga E

```
========================================
  CALCULADORA DE FLECHA VIGA BIAPOYADA
========================================

Verificando Python...

Usando: python

Iniciando aplicacion...

========================================
Iniciando Calculadora de Flecha...
```

Calculadora de Flecha - Viga Biapoyada

CÁLCULO DE FLECHA EN VIGA BIAPOYADA

Datos de entrada

Longitud (L):	5.0	m
Carga distribuida (q):	10.0	kN/m
Módulo E:	210000.0	MPa
Inercia (I):	5000.0	cm⁴

CALCULAR Y VER VIGA

Resultados

Flecha máxima: 0.007750 m = 7.75 mm

Flecha en mm: 7.75 mm

Momento máximo: 31.25 kN·m

Cortante máximo: 25.00 kN

Fórmulas utilizadas

Flecha máxima (en centro): f = (5 · q · L⁴) / (384 · E · I)
Momento máximo (en centro): M = q · L² / 8
Cortante máximo (en apoyos): V = q · L / 2

Guardar imagen de la viga

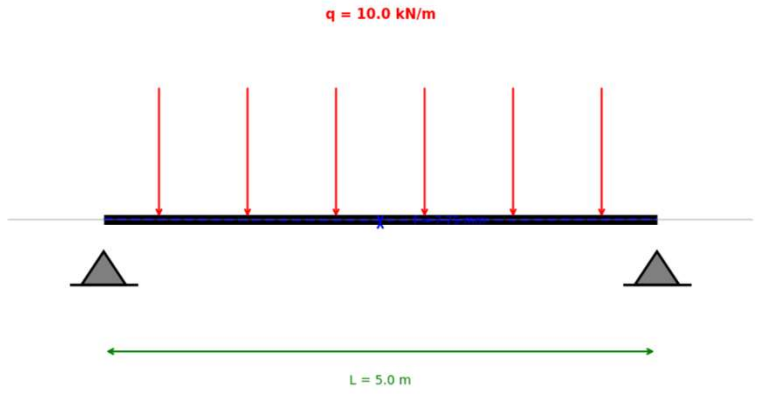
Figure 1

VIGA BIAPOYADA CON CARGA DISTRIBUIDA

RESULTADOS:

Flecha máxima: 7.750 mm
Momento máx: 31.25 kN·m
Cortante máx: 25.00 kN

DATOS:
L = 5.0 m
q = 10.0 kN/m
E = 210000 MPa
I = 5000 cm⁴



ejercicio 6:
IFC

Haz que el agente analice un archivo IFC de un edificio