

PROMPTING EN INTELIGENCIA ARTIFICIAL GENERATIVA

Módulo 6: Crear aplicaciones con LLMs

Instructor: Daniel M. Casas

Edición: Empresas

[Danimcasas.com](https://danimcasas.com) y Nironet formacio.

Máster en Inteligencia Artificial y Formador Tecnológico
danimcasas@inextalent.com

Actualizado: 2025



Módulo 6: Crear aplicaciones con LLMs

1. ¿Qué significa “crear aplicaciones con LLMs”?

Crear aplicaciones con **Large Language Models (LLMs)** significa integrar un modelo como **ChatGPT, Gemini, Claude** o **Llama 3** dentro de un flujo de trabajo o interfaz para resolver una tarea concreta.

Estas aplicaciones pueden ser tan simples como un generador de informes o tan complejas como un asistente administrativo virtual.

En esencia, cada aplicación se basa en tres pilares:

Pilar	Descripción	Ejemplo
Prompt bien diseñado	Define la tarea, el contexto y la salida esperada.	“Redacta un informe de 300 palabras sobre...”
Interfaz o entorno	Puede ser una hoja de cálculo, formulario web, chatbot o <i>script</i> .	Google Sheets, Streamlit, ChatGPT Canvas.
Automatización / API	Conecta el modelo con datos externos o automatiza respuestas.	OpenAI API, Make.com, Zapier, Apps Script.

Cómo funcionan las aplicaciones basadas en LLMs

El flujo general de una aplicación de IA generativa suele seguir este ciclo:

1. **Entrada del usuario** (texto, formulario, voz, datos).
2. **Prompt dinámico** que da contexto al modelo.
3. **Ejecución del modelo (LLM)** genera la respuesta.
4. **Postprocesado** (validación, resumen o formato).
5. **Salida estructurada o acción automática.**

Ejemplo real:

Un empleado introduce los datos de un expediente y el sistema genera automáticamente un borrador de informe formal, con conclusiones y formato oficial.

Tipos de aplicaciones posibles

Tipo de App	Descripción	Ejemplo práctico
Chatbots	Responden a preguntas en lenguaje natural.	Asistente virtual del ayuntamiento.
Generadores de documentos	Crean informes, actas o plantillas automáticas.	Informe de incidencia en vía pública.
Analizadores de texto	Extraen información o resumen de grandes textos.	Analizador de actas de reuniones.
Traductores contextuales	Mantienen tono y terminología administrativa.	Traductor automático gallego-castellano.
Asistentes de datos	Transforman datos estructurados en texto o viceversa.	Informe mensual de presupuesto a partir de Excel.

Herramientas y entornos más comunes

- **ChatGPT (Canvas y GPTs personalizados)** ideal para crear asistentes internos sin código.
- **Google Gemini** integrado en herramientas como Docs, Gmail o Sheets.

- **Microsoft Copilot** incrustado en Office 365, con funciones IA contextuales.
- **Make.com / Zapier** permiten automatizar procesos conectando IA con formularios, bases de datos o correos.
- **Streamlit / Gradio** *frameworks* Python sencillos para crear interfaces visuales.
- **WordPress + Plugins AI (ej. AI Power)** útil para crear asistentes web sin necesidad de programar.

Elementos clave para diseñar una aplicación basada en LLMs

1. **Claridad de propósito:** define qué problema soluciona la *app*.
2. **Diseño de prompts reutilizables:** usar plantillas y variables.
3. **Validación y control:** asegurar que las respuestas sean verificables.
4. **Privacidad:** evitar incluir datos personales o confidenciales.
5. **Iteración:** probar, ajustar, mejorar.

2. Ejemplos prácticos

Ejemplo 1 – Generador de informes automáticos

- **Objetivo:** crear una *app* que transforme datos en texto formal.

Prompt base:

Actúa como técnico administrativo.

Genera un informe de 300 palabras con los siguientes datos:

{datos}.

Usa formato con subtítulos, tono formal y evita repeticiones.

- **Implementación posible:**
 - **Entrada:** formulario web o hoja Excel.
 - **Procesamiento:** API de OpenAI o Google Gemini.
 - **Salida:** documento DOCX o PDF.

Ejemplo 2 – Chatbot de atención ciudadana

- **Prompt base (JSON):**
- JSON

```
{  
  
  "rol": "Asistente de atención ciudadana",  
  
  "instrucción": "Responde consultas sobre trámites municipales",  
  
  "condiciones": {  
  
    "tono": "cercano y respetuoso",  
  
    "fuente": "guía de trámites del ayuntamiento"  
  
  }  
}
```

- 👉 Puede implementarse con herramientas *no-code* (Make, Chatbase, GPTs personalizados).

Ejemplo 3 – Resumen automático de actas

Prompt base:

Resume el siguiente texto en formato tabla.

Columnas: "Tema tratado", "Conclusiones", "Acciones propuestas".

Usa un tono formal y no omitas ningún punto relevante.

- **Uso:** importar el acta desde un PDF o documento y exportar la tabla a Excel.

Ejemplo 4 – Asistente para generar contenidos en web municipal

Prompt base:

Crea una noticia breve para la web municipal.

Tema: {tema}.

Público: ciudadanía general.

Extensión: 150 palabras.

Incluye titular y subtítulo.

- 👉 **Integración posible:** API WordPress publicación automática.

Ejemplo 5 – Aplicación de análisis comparativo

Prompt base:

Compara estos dos informes de proyectos municipales y genera una tabla con: similitudes, diferencias, indicadores y conclusiones.

- **Resultado esperado:**
Una tabla con criterios objetivos para facilitar la toma de decisiones.

3. Resumen del módulo

- Crear *apps* con LLMs significa convertir **prompts en sistemas prácticos**.
- Toda *app* parte de **entrada + prompt + salida**.
- Los entornos **sin código** (Canvas, GPTs, Make, Zapier) permiten implementarlas fácilmente.
- Los LLMs pueden **leer, analizar, escribir y estructurar datos** en entornos administrativos.
- La clave está en usar **prompts estructurados, confiables y verificados**.

4. Práctica ampliada

Actividad Módulo 6 – Entregable: “Diseña tu propia app basada en LLMs”

El alumno deberá diseñar una mini aplicación funcional o simulada (no necesita programación) basada en *prompting*.

Parte 1 – Diseño conceptual

Describe la idea general de tu aplicación:

- Nombre o propósito.
- Problema administrativo que resuelve.
- Usuarios a los que se dirige.

(Ejemplo: “Asistente de generación de informes de mantenimiento urbano”).

Parte 2 – Arquitectura de prompts

Crea los *prompts* principales de tu aplicación:

1. **Prompt de inicio** o bienvenida.
2. **Prompt de procesamiento** (razonamiento o resumen).
3. **Prompt de salida** (resultado final o formato).

Incluye ejemplos de **entradas** y **salidas** esperadas.

Parte 3 – Estructura técnica

Explica cómo implementarías tu aplicación usando alguna de estas opciones:

- ChatGPT + Canvas o GPT personalizado.
- Gemini en Google Workspace.
- Microsoft Copilot o Power Automate.
- Plataforma *no-code* (Make, Zapier, Voiceflow, etc.).

Puedes incluir diagramas, esquemas o pseudocódigo.

Parte 4 – Validación y fiabilidad

Añade un *prompt* o sistema para verificar la calidad del resultado.

- **Ejemplo:** “Evalúa el texto anterior y marca incoherencias o datos sin verificar.”

Describe cómo lo incorporarías a la *app*.

Parte 5 – Entrega final

Entrega un documento (Word o PDF) con:

- Descripción completa.
- Prompts utilizados.
- Ejemplo de ejecución.
- **Reflexión final:** ¿qué limitaciones encontraste y cómo las resolverías?

5. Fuentes para repasar.

Fuente / Institución	Título / Descripción	URL
OpenAI	Building Systems with the ChatGPT API (curso práctico)	https://www.youtube.com/watch?v=BunESRhYhec
DeepLearning.AI	ChatGPT Prompt Engineering for Developers	https://www.youtube.com/watch?v=tRvcAdqsJWo
MIT	MIT 6.S087 – Generative AI: Applications and Interfaces	https://www.youtube.com/watch?v=yIfGlAECNFM
Harvard (HILT)	AI Tools for Productivity and Learning	https://www.youtube.com/watch?v=6lzn8Q6IK08
Microsoft Learn	Create intelligent apps with Azure OpenAI Service	https://learn.microsoft.com/en-us/azure/ai-services/openai/how-to/working-with-models

OpenAI Cookbook	Build your first app with the OpenAI API	https://cookbook.openai.com/examples/hello_world_app
Google Developers	Gemini for Apps – Integrate Generative AI into your workflow	https://developers.google.com/ai/gemini/docs/get-started/apps