

# Clase 7 - Material de lectura (Formato libro)

Sitio: [Centro de E-Learning - UTN.BA](#) Imprimido Ricardo Monla  
Curso: Diplomatura en Inteligencia Artificial por:  
para No Programadores  
Libro: Clase 7 - Material de lectura (Formato Día:  
libro) Sunday, 4 de January de 2026,  
21:34

## Tabla de contenidos

- 1. De generar a gobernar resultados con IA**
- 2. 2) Metodología práctica de iteración con IA (No Code)**
- 3. Plantilla breve de prompt (copiable)**
- 4. 3) Interpretar y ajustar resultados automáticos**
- 5. 4) Limitaciones y responsabilidad del usuario**  
5.1. Human In The Loop práctico (qué significa en la herramienta):
- 6. 5) Prácticas rápidas**
- 7. 6) Asistentes pre-armados**
- 8. 7) Caso de estudio (lectura guiada)**
- 9. Material de lectura**

# 1. De generar a gobernar resultados con IA

## 1) De generar a gobernar resultados con IA

- **Primero:** aprender a pedir bien (prompts y requerimientos claros) y prototipar con IA.
- **Luego:** verificar y gobernar esos resultados: validar, reducir sesgos y encajarlos en procesos reales con Human-in-the-Loop (HITL).

**Idea clave:** No basta con que la IA “genere algo”; necesitamos criterios de calidad y controles simples antes de producir efectos reales (publicar, enviar, facturar, etc.).

## 2. 2) Metodología práctica de iteración con IA (No Code)

### 2) Metodología práctica de iteración con IA (No Code)

#### 1. Enmarcar la tarea y el resultado esperado

- ¿Qué problema resuelve? ¿Para quién y para cuándo?
- Definición de hecho (DoD): cómo luce una salida “aceptable” (formato, campos obligatorios, tono, idioma).

#### 2. Reunir contexto y datos de referencia

- Políticas, glosario, precios/stock vigentes, nombres propios correctos.
- Aclará también lo que NO debe hacer (límites y exclusiones).

#### 3. Diseñar un prompt base con estructura

- Rol/Contexto · Objetivo · Entradas · Salida esperada + formato · Criterios/Restricciones · Estilo/Idioma.
- Pedí 2 variantes para poder comparar.

#### 4. Generar una primera salida (asistente IA) y guardar la versión

- Conservar “antes/después” facilita el aprendizaje.

#### 5. Validar con una checklist mínima

- ¿Están los campos obligatorios? ¿Formato correcto? ¿Sin datos sensibles? ¿Coherencia con el contexto?
- Si impacta afuera (clientes usuarios), previsualizar.

#### 6. Ajustar el prompt

- Acotar alcance, pedir formato estricto (JSON/tablas), agregar ejemplos positivos/negativos, aclarar tono.
- (Si la herramienta lo permite) bajar temperatura y fijar longitud máxima.

#### 7. Añadir controles simples en el flujo

- Paso de revisión humana (HITL) antes de enviar/publicar.
- Mensajes de error claros, reintentos y captura de excepciones.

#### 8. Probar casos límite

- Datos faltantes/duplicados, nombres poco comunes, ciudades/regiones distintas, textos muy largos/cortos.

#### 9. Registrar el aprendizaje

- Qué falló, cómo lo corregiste, ejemplo de prompt “antes/después”.
- Dejarlo en un doc/Sheet compartido para reuso del equipo.

#### 10. Desplegar con seguridad

- Versionado del flujo, responsables, cómo revertir si algo sale mal.
- Métricas simples: tasa de correcciones humanas, errores evitados, tiempo ahorrado

### 3. Plantilla breve de prompt (copiable)

#### Plantilla breve de prompt (copiable)

- **Rol/Contexto:** "Actuá como... en el ámbito ..."
- **Objetivo:** "Tu objetivo es..." (DoD en 1 línea)
- **Entradas:** "Usá SOLO estos datos: ..."
- **Salida/Formato:** "Devolvé ... en formato ... con campos: ..."
- **Criterios/Restricciones:** "No inventes datos; si falta X, indicá 'faltante'."
- **Estilo/Idioma:** "Tono formal/breve · Español (AR)"
- **Variantes:** "Dame 2 opciones distintas y marcá pros/contras en una línea."

## 4. 3) Interpretar y ajustar resultados automáticos

### 3) Interpretar y ajustar resultados automáticos

**Errores típicos (y cómo atenderlos de forma simple):**

Error	Ejemplo simple	Acción sugerida
<b>Confianza ciega en resultados</b>	Publicar un texto sin revisarlo y se cuela un dato erróneo	Previsualizar + Confirmación: siempre ver y aceptar manualmente antes de publicar/enviar
<b>Falta de contexto</b>	Resumen genérico que omite datos de tu empresa/ciudad	Contexto explícito: mencionar empresa, región, campos obligatorios
<b>Aceptar la primera respuesta</b>	Quedarte con el primer borrador aunque sea flojo	Iterar: pedir 2 variantes y elegir/combinar
<b>Datos/ejemplos sesgados</b>	Recomendaciones que ignoran población/realidad local	Acotar y equilibrar: pedir diversidad de ejemplos y límites claros
<b>Sin gestión de errores</b>	La IA no responde y el flujo "muere"	Plan B: mensaje claro al usuario + opción de reintentar

## 5. 4) Limitaciones y responsabilidad del usuario

### 4) Limitaciones y responsabilidad del usuario

- Los modelos pueden alucinar o estar sesgados por datos históricos.
- La responsabilidad es compartida: quien diseña el flujo debe prever controles y puntos de intervención humana.

En contextos sensibles (salud, finanzas, justicia) los controles deben ser más estrictos (doble validación, registro de decisiones, criterios de aceptación).

## 5.1. Human In The Loop práctico (qué significa en la herramienta):

### **Human In The Loop práctico (qué significa en la herramienta):**

- Paso de aprobación manual antes de enviar o publicar.
- Checklist de revisión visible para el equipo.
- Logs/historial de cambios y decisiones.

## 6. 5) Prácticas rápidas

### 5) Prácticas rápidas

#### 1. Ajuste de prompt:

- Toma un prompt de la Clase 6.
- Detecta un problema (ej., campo faltante, tono incorrecto).
- Agrega una línea de contexto/criterios y vuelve a ejecutar.
- Documenta en una frase: "Qué cambié y por qué mejoró la salida".

#### 2. Control humano mínimo:

- En tu flujo, añade un paso de confirmación antes de cualquier acción externa (correo, post, documento compartido).

## 7. 6) Asistentes pre-armados

### 6) Asistentes pre-armados

- **Asistente Gemini:** <https://gemini.google.com/gem/1w5TaYJB49jkAOV-INAAiPAI-XHUmuyC0?usp=sharing>
- **Asistente GPT – Desarrollo y Debug (Make):** <https://chatgpt.com/g/g-68d7df705d08819184f4ea5aaa99a695-asistente-de-desarrollo-y-debug-make>

Usá cualquiera de los dos para ejecutar los ejercicios de arriba.

## 8. 7) Caso de estudio (lectura guiada)

### 7) Caso de estudio (lectura guiada)

#### Sesgo en algoritmo de salud (EE. UU., 2019)

- **Qué pasó:** Un algoritmo comercial, usado para asignar ayuda adicional a pacientes con necesidades complejas, mostraba sesgo racial: a igual score de “riesgo”, pacientes negros estaban más enfermos que pacientes blancos.
- **Causa de fondo:** el algoritmo usaba costos de salud como proxy de “necesidad”. Como históricamente se gasta menos en poblaciones con peores barreras de acceso, ese proxy introducía sesgo.
- **Por qué importa:** elegir mal el objetivo o el proxy puede amplificar desigualdades.
- **Qué aprender:** define objetivos/labels que representen la realidad que querés mejorar (salud/ enfermedad), no sus consecuencias económicas.

#### Preguntas para pensar

- ¿Dónde pondrías checkpoints humanos para evitar daños?
- ¿Qué métrica de éxito revisarías (además de precisión) para vigilar sesgos?

## 9. Material de lectura

### Material de lectura

Te compartimos un [archivo descargable](#) con todo el contenido visto en la Clase 7.