CENTRO DE E-LEARNING UTN BA
Medrano 951 CABA, Buenos Aires Argentina
(1179) // tel +54 11 7078 – 8073 / fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

Documento Clase 7: IA para generar código sin conocimiento previo II

Material de lectura — Diplomatura IA para No Programadores

---

1. De generar a **gobernar** resultados con IA

En la primera etapa de trabajar con IA, nuestro enfoque suele estar en **generar** resultados: aprendemos a pedirle cosas de forma clara mediante *prompts* bien formulados y a construir pequeños prototipos basados en IA. Sin embargo, a medida que avanzamos es crucial pasar de simplemente generar a **gobernar** esos resultados. ¿Qué significa esto? Significa **verificar, validar y controlar** las salidas que nos da la IA antes de utilizarlas en el mundo real.

Idea clave: No basta con que la IA "genere algo"; necesitamos **criterios de calidad** y **controles simples** antes de producir efectos reales (por ejemplo, antes de publicar un texto generado, enviar un código a producción, mandar un correo automático, emitir una factura, etc.)*[1]*. En términos sencillos, debemos actuar como "filtro" y "gobernador" de la IA:

- **Primero:** hay que **aprender a pedir bien** (prompts claros, requerimientos precisos) y quizás crear un prototipo rápido con la IA para ver por dónde van los resultados*[2]*.
- **Después: verificar y gobernar** esos resultados, lo que implica validarlos, **reducir sesgos** si los hubiera, y **encajarlos en procesos reales** donde correspondan*[2]*. En otras palabras, integrar la salida de la IA a nuestros flujos de trabajo con **supervisión humana** (lo que se conoce como enfoque *Human-in-the-Loop* o HITL) y con reglas de negocio.

💡 **Analogía:** Pensemos en la IA como un **asistente nuevo en la oficina**. Al principio, le decimos claramente qué necesitamos (por ejemplo, un informe), y el asistente nos entrega un borrador. No lo enviamos directamente al cliente sin mirarlo; **lo revisamos, corregimos errores y verificamos que cumple con nuestros estándares**. Solo entonces lo damos por finalizado. Del mismo modo, con la IA debemos revisar lo que genera antes de usarlo externamente.

En este proceso de gobierno de resultados con IA, se vuelven cruciales conceptos como la **validación** (¿es correcto y útil lo que produjo?), la **reducción de sesgos** (¿hay alguna

tendencia injusta o error sistemático en la salida?) y la **integración con humanos en el ciclo** (*Human-in-the-Loop*). Esto último quiere decir que siempre haya un punto donde una persona revise o apruebe lo que hizo la IA, especialmente en tareas sensibles. Más adelante profundizaremos en *Human-in-the-Loop*, pero adelantemos que es un componente esencial de la **IA responsable**: implica que los humanos supervisan y pueden intervenir en puntos críticos del proceso de IA para guiar o corregir los resultados*[3]*.

¿Por qué es importante este cambio de enfoque? Porque si solo nos quedamos en "que la IA genere cosas" corremos el riesgo de confiar ciegamente y difundir errores o contenido inapropiado. En entornos profesionales, **la responsabilidad final recae en nosotros, los usuarios o implementadores de la IA**, no en la máquina. Debemos garantizar que lo que la IA entrega cumple con las expectativas y no cause problemas.

Por ejemplo, si la IA genera código, hay que **probar ese código**; si genera un informe, hay que **revisar que los datos sean correctos**; si redacta un email para un cliente, hay que **leerlo antes de enviarlo**. A esto nos referimos con "gobernar resultados": ejercer control de calidad y responsabilidad sobre lo que produce la inteligencia artificial antes de liberarlo al mundo real.

En resumen, pasamos de ser simplemente **solicitantes** (pedir y recibir contenido de la IA) a ser también **guardianes o gobernadores** de ese contenido, asegurándonos de que sea adecuado. En los siguientes apartados veremos metodologías prácticas para lograr esto, desde cómo iterar con IA para mejorar resultados hasta cómo implementar chequeos y *checkpoints* humanos en un flujo de trabajo impulsado por IA.

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

Trabajar efectivamente con IA sin saber programar requiere un enfoque metódico. A continuación, presentamos una **metodología paso a paso** para **iterar con una IA** y refinar sus resultados. Esta metodología es especialmente útil cuando usamos herramientas *No Code* (sin programación) con asistentes de IA, y nos ayudará a obtener salidas de mayor calidad y confiabilidad. Mantendremos la estructura planteada en la clase, ampliando cada punto con explicaciones, ejemplos y consejos prácticos.

2.1 Enmarcar la tarea y el resultado esperado

Antes de siquiera escribir un prompt, es fundamental **enmarcar bien la tarea** que le vamos a pedir a la IA y tener claro **qué resultado esperamos**. Esto significa responder preguntas básicas como: **¿Qué problema queremos resolver? ¿Para quién es la solución y para cuándo la necesitamos?***[4]*.

- **Propósito de la tarea:** Identifica cuál es la necesidad o problema. Por ejemplo, "*Necesito generar un resumen de noticias diarias para el equipo de ventas.*" O "*Quiero obtener código SQL para hacer cierta consulta a mi base de datos.*"
- **Destinatario y contexto:** ¿Quién utilizará este resultado y en qué contexto? No es lo mismo un informe para ejecutivos (que debe ser formal y breve) que un mensaje para clientes jóvenes en redes sociales (que puede ser más informal). Ejemplo: "*El resumen*

es para el equipo de ventas de mi empresa, que no tiene conocimientos técnicos, y se usará en la reunión de cada mañana."

- **Plazo o frecuencia:** ¿Es algo urgente, es algo que se hará regularmente? Por ejemplo, "*Este resumen se necesita cada día antes de las 9 AM.*" Saber esto ayuda a calibrar cuánto tiempo podemos invertir en refinar la salida y si requerimos automatizarlo.

En esta etapa, básicamente estamos definiendo el **objetivo** de la interacción con la IA en términos humanos. Esto se parece mucho a definir requisitos en cualquier proyecto. Un buen ejercicio es tratar de **explicar en una o dos frases qué se busca**, por ejemplo: "*Quiero que la IA genere un borrador de código en Python que procese un archivo CSV de ventas y calcule métricas básicas, para yo luego revisarlo.*"

💡 **Analogía simple:** Enmarcar la tarea es como encargar un trabajo a un **freelancer** o a un **proveedor externo**. Antes de encargarlo, uno define qué necesita, para cuándo y bajo qué condiciones. No le diríamos simplemente "haz algo interesante", sino que detallaríamos qué queremos lograr. Con la IA es igual: cuanto más claro tengamos el *qué, quién, para quién y para cuándo*, mejor podremos guiarla.

2.2 Definir la "Definición de Hecho" (DoD) o criterio de éxito

Una vez sabemos qué queremos en términos generales, el siguiente paso es establecer cómo luce una **salida aceptable**. En desarrollo de software se usa el término "Definition of Done" (DoD) para describir los criterios que debe cumplir una tarea para considerarse completa. Aquí podemos traducirlo como **Definición de Hecho** (o definición de terminado). Básicamente preguntarnos: *¿Cómo sé que la respuesta de la IA es lo que necesito?*

Al definir esto, considera los siguientes aspectos[5]:

- **Formato esperado:** ¿Debe la respuesta ser un texto libre, una lista, un código en un lenguaje específico, un JSON, una tabla...? Ejemplos: "*Quiero la salida en formato de tabla con columnas X, Y, Z*" o "*Necesito que la respuesta sea un párrafo de no más de 5 oraciones.*"
- **Campos o datos obligatorios:** Si se trata de generar texto estructurado (un informe, un email, una respuesta a cliente), ¿qué elementos sí o sí deben aparecer? Ejemplo: "*El resumen debe incluir la fecha, los 3 titulares principales y una recomendación final.*" Si es código, ¿qué variables o entradas debe manejar? "*La función debe recibir el nombre del archivo CSV como parámetro.*"
- **Tono e idioma:** ¿Debe ser formal o informal? ¿Está en español neutro, español de Argentina, u otro idioma? Ejemplo: "*El tono debe ser profesional pero accesible, en español rioplatense.*"
- **Nivel de detalle:** ¿Queremos algo breve o algo extenso y detallado? "*La respuesta debe ser breve, tipo bullet points*" vs "*que incluya explicaciones detalladas para cada punto.*"

- **Restricciones específicas:** Por ejemplo, "*No debe incluir datos confidenciales,*" o "*No debe hacer suposiciones no indicadas.*" También puede ser relevante establecer *qué no queremos*.

Escribir estos criterios ayuda mucho. Por ejemplo, podríamos anotar: "*Salida aceptable = Código Python bien formateado, que se ejecute sin errores básicos, con comentarios en español y que siga nuestras convenciones de estilo.*" O bien: "*Salida aceptable = Resumen de 200 palabras, mencionando X e Y, en tono formal, sin datos inventados.*"

Tener esta "definición de éxito" antes de interactuar con la IA nos sirve como **guía para revisar el resultado** después. Si la IA nos da algo que no encaja con lo que definimos, sabremos exactamente en qué falló (quizá faltó un campo, o el tono no fue el correcto) y podremos iterar.

⚠️ **Importante:** Comunica estos criterios a la IA en el prompt si es posible. Si tú sabes qué formato y contenido necesitas, **díselo a la IA desde el inicio**. Así aumentan las probabilidades de que te dé lo que esperas. Por ejemplo: "*Tu respuesta debe estar en español, tono casual, y consistir en una lista de viñetas que cubra A, B y C.*".

2.3 Reunir contexto y datos de referencia

La IA trabaja en base a la información que le proporcionamos (más su conocimiento entrenado). Un error común es olvidar darle contexto relevante y luego sorprendernos de que la respuesta sea demasiado genérica o imprecisa. Para evitar esto, **reúne y prepara los datos o referencias que la IA necesite conocer** para la tarea*[6]*.

¿Qué tipo de contexto puede ser útil?:

- **Información de tu organización o problema específico:** Por ejemplo, políticas internas, un glosario de términos de la empresa, precios actualizados de tus productos, stock disponible, nombres propios (de personas, productos, proyectos) y sus descripciones correctas. *Si la IA va a generar un reporte de ventas, facilítale los datos clave: resultados del mes, nombres de los equipos, metas, etc.*
- **Ejemplos relevantes:** Si quieres un cierto estilo de output, considera mostrarle a la IA un ejemplo (si la herramienta lo permite) o describirlo. Ejemplo: "*Por ejemplo, nuestro tono suele ser: 'Estimado cliente: ... Saludos cordiales.' No queremos expresiones demasiado coloquiales.*"
- **Limitaciones explícitas:** Indica también **lo que no debe hacer o usar**. Por ejemplo: "*No incluyas información anterior a 2020*" si es relevante, o "*No mencionés X producto, porque está descontinuado.*".
- **Datos externos confiables:** Si necesitas que la IA use datos específicos (por ejemplo, la cantidad de habitantes de una ciudad, o un hecho histórico preciso), proporciónaselos en el prompt o asegúrate que los tenga. Cuanto menos tenga que adivinar la IA, mejor será el resultado.

Recopilar este contexto es parecido a como cuando le damos a un consultor humano documentos o briefings antes de que empiece a trabajar. No queremos que "reinvente la rueda" ni que cometa errores que podríamos evitar con la información adecuada.
Ejemplo: Supongamos que la tarea es redactar una descripción de producto para la página web de una tienda. Contexto útil sería: características del producto, precio, stock, público objetivo, tono de la marca. Si no damos eso, la IA hará una descripción genérica que tal vez no encaje con la realidad de nuestro inventario o estilo de comunicación.
💡 **Tip:** A veces es útil hacer una breve **lista de bullet points** en el prompt con todo el contexto y datos clave. Por ejemplo: - Nuestra empresa: ACME Inc, sector financiero en Argentina. - Objetivo: email de saludo a nuevo cliente que abrió una cuenta. - Debe incluir: nombre del cliente, beneficios principales (X, Y, Z). - No mencionar: precios específicos, evitar jerga técnica. - Tono: cercano pero respetuoso, español (AR).
Con un contexto así, la IA tendrá mejores "mimbres" para tejer su respuesta.
2.4 Aclarar lo que **NO** debe hacer (límites y exclusiones)
Este punto es tan importante como decirle lo que *sí* queremos. Muchas veces, la IA puede desviarse o incluir elementos no deseados porque no sabe que son inapropiados. Por eso, en el prompt podemos (y debemos cuando corresponda) **aclarar explícitamente lo que NO queremos***[6]*.
Algunos ejemplos de límites/exclusiones a indicar:

- **Datos sensibles o privados:** "*No incluyas direcciones de email reales ni datos personales de nadie.*"
- **Suposiciones o invenciones:** "*Si falta información, no inventes datos. En ese caso, indica 'dato no disponible' en lugar de inventar.*" Esto es crucial para evitar las llamadas *alucinaciones* de la IA (cuando la IA se "inventa" algo plausible pero falso, lo que explicaremos más adelante).
- **Ámbitos fuera de alcance:** "*Limítate al contexto dado. No hables de otros productos que no mencioné.*" Por ejemplo, si la IA a veces tiende a divagar, podemos restringirlo.
- **Estilo inadecuado:** "*No uses lenguaje vulgar ni informal*" (si quisiéramos evitarlo) o "*No uses emoticones ni emojis.*"
- **Cualquier otro riesgo identificado:** Por ejemplo, "*No ofrezcas consejos médicos, solo explica las generalidades.*" en caso de estar redactando algo sobre salud (porque solo un profesional debería dar consejos personales).

Es útil imaginar qué podría salir mal o fuera de lugar y advertirlo. Esto funciona como **"vallas de contención"** para la IA. Muchas IAs son obedientes a instrucciones negativas si las formulamos claramente. Por ejemplo: "*No mencionar ningún contenido político*" si eso no viene al caso y queremos evitar roces.
Ejemplo práctico: Estás usando IA para generar respuestas en un chatbot de servicio al cliente. Podrías añadir: "*No des información de contacto personal, no confirmes ni niegues si el*

cliente tiene deuda (porque es información sensible que no debe compartirse por chat), no hagas promesas que no estén en nuestras políticas." De esta manera, la IA sabrá que hay líneas rojas que no debe cruzar.

Recuerda que **la IA no tiene sentido común ni conoce las normas de tu empresa a menos que se las indiques**. Por eso, dejar claros los **NO** es parte de un prompt bien diseñado.

2.5 Diseñar un **prompt base** con estructura clara

Ahora sí, con el objetivo definido, criterios de éxito claros, contexto recopilado y límites establecidos, estamos listos para **redactar el prompt**. Un **prompt base estructurado** aumentará mucho las chances de obtener lo que buscamos en el primer intento.

Una estructura útil (sugerida en la clase) para organizar el prompt es incluir secciones como*[7]*:

- **Rol/Contexto:** Indica a la IA qué rol tomar o en qué contexto trabajar. Ej: "*Actúa como un analista de datos experto en marketing digital...*" o "*Eres un asistente de programación experto en Python...*". Esto pone a la IA en el modo correcto.
- **Objetivo:** Describe brevemente la tarea. "*Tu objetivo es generar un reporte de ventas semanales...*" o "*Tu objetivo es escribir el código para X...*". Aquí puedes resumir también la Definición de Hecho en una línea: qué sería una buena salida.
- **Entradas (Input):** Si hay datos específicos que le das, menciónalos. "*Usa SOLO estos datos: [listado de datos clave o resumen].*" Si no hay datos numéricos específicos pero le diste contexto, podrías omitir este punto o decir "*Basate únicamente en la información proporcionada arriba*".
- **Salida esperada + Formato:** Aquí detalla qué debe entregar. "*Devuelve la respuesta en formato de lista con viñetas, incluyendo A, B, C.*" o "*Entrega el código en Markdown con sintaxis Python.*" O si es JSON, "*Devuelve un JSON válido con campos 'nombre', 'edad', ... etc.*"
- **Criterios/Restricciones:** Las cosas a no hacer o tener cuidado. "*No alucines datos. Si falta información, responde 'dato no disponible'.*" o "*No menciones otras marcas.*" etc. También puedes incluir requisitos de calidad: "*El código debe seguir la PEP8 (en el caso de Python)*", "*El texto no debe superar 2 párrafos.*", "*Debe incluir referencias si se mencionan datos duros.*", etc.
- **Estilo/Idioma:** Reitera el tono y el idioma. "*Usa un tono formal y técnico, en español (Argentina).*" o "*Tono cercano, lenguaje sencillo, en español neutro.*"
- **Variantes (opcional):** Si la herramienta lo permite, puedes pedir más de una variante. "*Genera 2 opciones distintas de respuesta.*" Esto es útil para comparar (lo veremos en el siguiente punto).

Juntar todo esto en un solo mensaje suena extenso, pero vale la pena. A veces es útil hasta formatearlo con viñetas o secciones delimitadas para mayor claridad. Muchos modelos de lenguaje aprecian la estructura en la entrada.

Ejemplo de plantilla de prompt estructurado: (podemos usar la **plantilla breve copiable** dada en el material de clase):
Rol/Contexto: Actuá como un analista de datos en el ámbito de ventas minoristas.
Objetivo: Tu objetivo es generar un resumen semanal de las ventas (Definición de Done: incluye totales por categoría, mejor y peor día de ventas).
Entradas: Usá SOLO estos datos: (1) Ventas diarias de la semana por categoría, (2) Meta semanal de ventas.
Salida/Formato: Devolvé un informe en formato de lista con viñetas, con los campos: "Resumen general", "Mejor día", "Peor día", "Cumplimiento de meta".
Criterios/Restricciones: No inventes datos; si falta alguna categoría, indicá "faltante". No incluyas datos de otras semanas.
Estilo/Idioma: Tono formal pero accesible · Español (AR).
Variantes: Dame 2 opciones distintas de informe y, al final, agregá en una línea una breve nota comparando cuál es más detallada.
Arriba hemos mezclado todo en un solo mensaje estructurado. Observa cómo se cubren todos los aspectos: le decimos *quién es* (rol), *qué tiene que hacer*, *con qué datos*, *en qué formato*, *qué no hacer* y *cómo expresarlo*, e incluso pedimos dos resultados para poder elegir.
💡 **Consejo: Si tu herramienta lo permite, guarda esta plantilla de prompt** como base. Podrás reutilizarla y modificarla según la tarea. Muchas veces, tener una plantilla y llenar los huecos (como quien llena un formulario) te ahorra tiempo y asegura que no olvides nada importante. Por ejemplo, esa plantilla base la podrías reutilizar para distintos departamentos cambiando "ventas" por "recursos humanos" o "marketing", etc.
2.6 Pedir **2 variantes** para comparar (cuando sea posible)
Como mencionamos en la plantilla, solicitar **más de una variante de respuesta** es muy útil. ¿Por qué? Porque la primera respuesta de la IA podría tener aciertos y errores, y tener una segunda (o tercera) opción nos permite **comparar y elegir lo mejor de cada una**. Incluso si ambas no son perfectas, compararlas nos da una idea de posibles enfoques.
Muchas herramientas de IA generativa (como ChatGPT, etc.) permiten o toleran que se pida algo como "*Dame 2 alternativas*". Si la tuya lo soporta, úsalo a tu favor. Si no lo hace de forma automática, siempre puedes simplemente pedir de nuevo con ligeros cambios y obtendrás otra versión.
Ejemplo: Le pedimos a la IA: "*Escribe dos versiones del email de bienvenida al nuevo cliente, una más formal y otra más amistosa.*" Así obtenemos dos estilos y podemos ver cuál encaja mejor o combinar elementos de ambas.
Otra manera de obtener variantes es ajustar la *aleatoriedad* (parámetro de "temperatura" en algunos modelos) o re-generar la respuesta manteniendo el prompt. Sin entrar en detalles técnicos, algunas interfaces tienen un botón de "Regenerar respuesta". Aprovecha eso para no quedarte con lo primero que salga si no estás convencido.

💡 **Tip práctico:** Cuando tengas **dos variantes**, intenta hacer una pequeña comparación: ¿Cuál de las dos cumple mejor tu Definición de Hecho? ¿Se complementan entre sí? A veces es útil incluso **combinar**: tal vez la Variante A está mejor estructurada pero la Variante B tiene un dato útil que A olvidó. Como usuario gobernando el resultado, puedes armar un resultado final superior utilizando lo mejor de cada una. ¡La IA no se ofende si editas o mezclas sus salidas! 😉

Finalmente, **guardar las variantes** y lo que has elegido es buena práctica, porque queda un registro de cómo fue mejorando la salida. Esto nos lleva al siguiente punto.

2.7 Generar una primera salida y **guardar la versión**

Al obtener la primera respuesta del asistente de IA (o las primeras variantes), es recomendable **guardar esa versión inicial** antes de comenzar a hacer cambios o ajustes. ¿Para qué? Para tener un "*antes y después*" que facilite el aprendizaje y la mejora continua[8].

Puedes guardar la salida en un documento, en una nota, o si la herramienta tiene historial, asegúrate de saber cuál fue la primera iteración. ¿Por qué es útil? Imagina que tras varias mejoras, quieres explicar a tu equipo qué tan lejos llegaron desde el primer borrador. Tener la versión original permite **ver el progreso** y también analizar qué partes vinieron mal desde el inicio.

Además, a veces ocurre que en iteraciones posteriores la IA mejora unas cosas pero empeora otras. Si guardaste la versión inicial, puedes rescatar alguna frase o dato que en versiones posteriores desapareció accidentalmente.

Ejemplo real: Supón que la IA generó un código que corría pero sin optimización. En la segunda iteración, le pediste optimizar y comentar el código. El nuevo código sale más rápido pero, por alguna razón, dejó de incluir un comentario importante que sí estaba en el primero. Si guardaste el primero, puedes copiar ese comentario y reintroducirlo. Si no lo guardaste, tendrás que pedírselo de nuevo o reconstruirlo de memoria.

Guardar versiones también facilita pedir ayuda si algo sale mal: puedes mostrar "*Este fue el primer resultado, luego hice X cambio en el prompt y obtuve este segundo resultado...*" para que alguien más entienda el proceso.

💡 **Consejo:** Lleva un pequeño **registro de versiones de prompt y resultados**. Por ejemplo: - Versión 1 prompt: "Describir producto A con tono informal." Resultado... (lo guardas). - Versión 2 prompt: "Describir producto A con tono informal, **añadiendo especificaciones técnicas**." Resultado... etc.

Con este mini-historial, aprendes qué cambios funcionaron. Incluso en equipo, pueden consolidar un "repositorio de prompts mejorados".

En resumen, no trabajes **sobre la única copia**. Siempre es bueno tener el respaldo del resultado anterior antes de sobreescribirlo con uno nuevo.

2.8 Validar con una **checklist mínima**

Ahora tienes una salida inicial de la IA (o un par de variantes). Es tentador a veces decir "*Listo, ya está, suena bien*" y usarla de inmediato. **¡No caigamos en la trampa de la confianza ciega!** Antes de dar por buena la respuesta de la IA, hay que validarla aunque sea con un **chequeo rápido** usando una lista de verificación (*checklist*). La clase sugirió algunos puntos básicos*[9]*:

Pregúntate al revisar la salida:

- **¿Están todos los campos o elementos obligatorios?** Vuelve a tu Definición de Hecho: si pedías que incluya 5 puntos y solo ves 4, hay algo faltante. Ej: la IA debía incluir la fecha en el informe, ¿lo hizo?
- **¿El formato es correcto?** Si pediste una tabla, ¿la estructura es de tabla? Si era JSON, ¿es un JSON válido? Si es código, ¿falta algún import o alguna llave de cierre?
- **¿Hay datos sensibles que no deberían estar?** Revisa que no se haya "colado" información privada o inventada. A veces la IA puede, por ejemplo, en un informe agregar nombres ficticios o suponer datos. Verifica eso.
- **¿Coherencia con el contexto dado?** Comprueba que no haya algo que contradiga el contexto o datos que proporcionaste. Por ejemplo, si le diste un dato de que la empresa tiene 200 empleados y en la respuesta dice "con más de 500 empleados", claramente algo anda mal. O si era un resumen para Argentina y empezó a hablar de datos de otro país, hay falta de alineación con el contexto.
- **¿El tono/estilo corresponde a lo pedido?** Quizá pediste tono formal y lo ves lleno de jerga juvenil, entonces no se cumplió del todo.

Una forma sencilla es armar tu **propia checklist** antes (incluso puede ser mental) y al recibir la respuesta, revisarla punto por punto. Por ejemplo: "*1) Tiene saludo inicial? 2) Menciona el nombre del cliente? 3) Incluye los 3 beneficios? 4) Despedida cordial?*". Si algo falla, lo marcas.

Ejemplo: Si la IA generó un pequeño programa de automatización, tu checklist podría ser: "*¿Se ejecuta sin errores? ¿Cumple el requerimiento principal? ¿Tiene comentarios que expliquen el código? ¿Maneja errores (ej: si falta un archivo)*?". Ejecutas el código, ves resultados, y evalúas. Quizá descubres que "*no maneja el caso de archivo no encontrado*"; anótalo para la próxima iteración.

⚠️ **Nunca omitas esta validación**, por más corta que sea la respuesta. Incluso una simple frase puede tener errores (¿imaginan un correo donde por un error la IA puso "Estimado {nombre}" porque no recibió el nombre correctamente?). El costo de un minuto de revisión es bajísimo comparado con el bochorno o error que podría ocurrir de no detectar algo evidente.

Y si la salida de la IA va a impactar *afuera*, es decir, la van a ver clientes, usuarios o va a publicarse en algún lado, **doble cuidado**. Lo ideal es **previsualizar exactamente cómo se verá**. Vamos a ese punto:

2.9 Previsualizar si impacta externamente

Si el resultado que generaste con la IA será consumido por un usuario final (por ejemplo, un email automatizado que realmente le llegará a un cliente, un post en redes sociales, un reporte que se imprimirá), entonces además de la revisión técnica, conviene **visualizarlo tal cual lo verá la persona** antes de lanzarlo*[10]*.
Esto quiere decir: no solo mires el texto *en la interfaz de la IA*, sino intenta ponerlo en el formato final. Por ejemplo:

- **Email:** Cópialo en un borrador de email, revisa asunto, destinatario, formato, si los links funcionan (si la IA puso algún link).
- **Documento/formato:** Si es un PDF o informe, genera ese PDF y ábrelo. Asegúrate de que los gráficos (si los agrega) se ven bien, que los títulos no quedaron cortados, etc.
- **Post en red social:** Pruébalo en un perfil de prueba o en vista previa, para ver si cabe el texto, si el tono se ve adecuado en contexto.
- **Código en producción:** Si es un script, pruébalo en un entorno controlado antes de pasarlo a producción.

Ejemplo: Supongamos que la IA generó un mensaje de WhatsApp automatizado para clientes. Podrías enviarlo a un número de prueba para ver si la longitud es la correcta (no se corta), si los emojis que puso se ven bien, etc. Quizá en la vista notas que el mensaje es demasiado largo para la plataforma y conviene recortarlo.
Previsualizar también ayuda a captar errores que pasan inadvertidos en texto plano. A veces un texto puede parecer correcto, pero al verlo "*como flyer*" te das cuenta de que una frase suena extraña para el público objetivo, o que hay un detalle fuera de lugar.
Este paso es parte de esa **última milla de calidad**. Así como un diseñador web prueba la página en distintos navegadores, nosotros probamos el *output* de la IA en el "lienzo" final donde vivirá.
2.10 Ajustar el prompt y **reiterar**
Después de la validación, es común encontrar una o varias cosas que se pueden mejorar. ¡Es totalmente normal! El siguiente paso entonces es **ajustar el prompt** con base en lo que observamos y volver a pedir a la IA que genere la salida. Esta es la iteración propiamente dicha: cada ciclo prompt -> respuesta -> revisión -> nuevo prompt ajustado.
¿Qué tipos de ajustes son los más útiles según la situación? El material sugiere varios enfoques*[11]*:

- **Acotar el alcance:** Si la respuesta fue demasiado genérica o larga, limita el terreno. Ej: "*Concéntrate solo en las tres causas principales, no listes todo.*" o "*No entres en detalles de historial, solo habla del estado actual.*"
- **Pedir formato estricto:** Si en el primer intento la IA no respetó formato, sé más exigente. Ej: "*Responde EXACTAMENTE en formato JSON con campos 'nombre', 'edad' y 'ciudad', sin explicaciones adicionales.*" Incluso puedes darle un ejemplo de la estructura deseada en el prompt (un pequeño JSON de ejemplo con valores ficticios).

- **Agregar ejemplos positivos/negativos:** Esto es poderoso. Si la IA no entendió bien el estilo, proporciona un ejemplo: "*Ejemplo de respuesta esperada: [ejemplo]. Ejemplo de respuesta que NO queremos: [ejemplo].*" Así la IA tiene referencias claras.
- **Aclarar el tono o estilo nuevamente:** A veces hay que enfatizar: "*Recuerda: el tono debe ser formal, nada de chistes.*" o "*Usa un lenguaje muy sencillo, como explicando a un niño de 12 años.*" Si la salida anterior fue muy técnica, esto ayuda.
- **Temperatura y longitud (si la herramienta lo permite):** La *temperatura* es un parámetro que controla la aleatoriedad de la IA. Temperatura alta (~1) = más creativo pero más riesgo de incoherencias; temperatura baja (~0) = más preciso/repetitivo. Si tu herramienta permite fijarla, puedes bajarla un poco si la primera respuesta fue demasiado "creativa" o errática. Por ejemplo, de 1.0 a 0.7. También, algunas herramientas permiten decir "*máximo 200 palabras*" o un número de tokens, para evitar salidas larguísimas. Aprovecha eso si la respuesta fue excesiva en longitud*[12]*.

En esta fase, es útil **cambiar una cosa a la vez y observar**. Por ejemplo, primero intentas agregando un ejemplo bueno y malo, y vuelves a correr. Si mejora pero aún hay un detalle, luego intentas agregar otra restricción.

Ejemplo práctico de iteración: Queremos que la IA genere código SQL para una consulta. Versión 1 prompt: "*Escribe un SQL para obtener ventas totales por mes.*" La salida funciona pero notamos que no filtró por año. Ajuste: Versión 2 prompt: "*... para obtener ventas totales por mes del año 2023.*" Sale bien pero los nombres de columnas no coinciden con nuestra base. Ajuste: Versión 3 prompt: "*Usa la tabla Ventas (col: fecha, monto) y obtén ventas totales por mes de 2023. Las columnas en el resultado deben ser 'mes' y 'total_ventas'.*" Ahora ya clava el SQL perfecto. Observa cómo fuimos afinando detalles con cada iteración.

🔄 **Iterar es normal:** No esperes la perfección en el primer intento. Incluso desarrolladores experimentados con IA hacen varias pasadas hasta quedar satisfechos. La clave es **aprender de cada iteración**: si en la primera el fallo fue X, añade algo para corregir X en el prompt; si en la segunda el fallo es Y, aborda Y, y así sucesivamente.

Con la práctica, notarás que vas anticipando esos ajustes desde el inicio (es decir, tus prompts iniciales serán más completos) y el número de iteraciones necesarias disminuye. Pero siempre mantén una actitud de "*ok, probemos, veamos el resultado, y ajustemos*". La IA es flexible y no se cansa de que le pidas refinamientos (¡a diferencia de un humano! 😅).

2.11 Añadir **controles simples** en el flujo (paso de revisión, errores, etc.)

Hasta ahora hemos hablado del proceso de iteración interactiva con la IA para obtener una *buena salida*. Supongamos que ya la tenemos tras varias iteraciones. ¿Terminó el trabajo? Si vas a **integrar esta salida en un flujo automatizado** (por ejemplo, una secuencia en una herramienta *no-code* tipo Zapier, Make, Power Automate, etc.), conviene añadir unos pasos extra de **control de calidad dentro del flujo** para mayor seguridad*[13]*.

Dos controles simples pero poderosos son:

- **Paso de revisión humana (Human-in-the-Loop) antes de una acción final:** Esto significa que antes de que el flujo haga algo irreversible o público (enviar el correo al cliente, publicar la respuesta en web, ejecutar la transacción, etc.), insertes un paso donde un humano da el visto bueno. Por ejemplo, en un proceso *no-code*, podrías tener un paso "Enviar a aprobación" donde tú o alguien del equipo lee el texto generado y marca un checkbox si todo está bien, solo entonces el flujo continúa. Es literalmente implementar ese *HITL práctico* del que hablamos: una aprobación manual antes de enviar/publicar*[14][15]*. En muchas plataformas, puede ser mandar un email interno con el contenido para que alguien lo revise, o pausar la ejecución hasta confirmación.
- **Manejo de errores y excepciones:** Piensa qué pasa si la IA *no responde* o responde mal. El flujo no debería simplemente "morir" sin dejar rastro. Agrega un **mensaje de error claro** al usuario si algo falla, e intenta un **reintento** automático quizás. Por ejemplo, si la IA no devolvió nada (posiblemente por un fallo de API), tu flujo podría reintentar una vez más, y si aun así falla, notificar "Lo sentimos, no se pudo generar la respuesta en este momento. Intenta más tarde."*[16]*. También captura excepciones: si esperabas un JSON y vino texto, quizás añade un paso que valide el JSON; si es inválido, que envíe una alerta a un humano en vez de procesar datos erróneos.

Estos controles se parecen a los **"guardarraíles"** de un auto con piloto automático: el auto (IA) maneja, pero tienes barandas y alguien listo para tomar el volante si el auto se desvía.
Ejemplo: Tienes un flujo que atiende preguntas de clientes automáticamente con IA. Implementas: - Si la confianza de la IA en la respuesta es baja (algunas herramientas devuelven un score, pero si no, podrías basarte en ciertas palabras clave de incertidumbre), en lugar de enviar directo al cliente, escalas a un agente humano. - Pones un paso que si la respuesta generada contiene la frase "Lo siento, no puedo..." (indicando posiblemente que la IA no supo responder), no se envía tal cual, sino que deriva la consulta. - Cada 10 respuestas enviadas, un supervisor las revisa por muestreo (ese es un HITL a nivel de monitoreo).
⚠️ **Nunca olvides los errores:** Todo sistema robusto contempla qué hacer si algo sale mal. En IA, "*que algo salga mal*" puede ser tanto un fallo técnico como un contenido inadecuado. Los pasos de control simples reducen riesgos. Un mensaje de error claro al usuario final siempre es mejor que dejarlo frente a un silencio o un texto raro. Y si tu flujo puede reintentar o tomar camino alternativo, mejor experiencia.
En resumen, cuando pases de la fase de diseño a la fase de *deployment* (despliegue del flujo con IA), **no lo dejes 100% en piloto automático**. Pon semáforos en amarillo donde haga falta: una confirmación humana aquí, un manejo de excepción allá. Esto hará tus soluciones de IA mucho más confiables y profesionales.
2.12 Probar **casos límite**

¿El flujo ya está listo y controlado? Excelente, pero antes de cantar victoria, es recomendable **probar casos límite o escenarios extremos** para ver cómo se comporta la IA y el sistema*[17]*. Los casos límite son situaciones no tan comunes pero posibles, que podrían descolocar a la IA o al flujo:
Algunos casos límite que podrías probar:

- **Datos faltantes:** ¿Qué pasa si la entrada viene incompleta? Ejemplo: la IA debe generar una carta con el nombre del cliente, pero ¿y si el nombre viene vacío o nulo? ¿El resultado que hace? ¿Pone "Estimado [Nombre]"? Deberíamos probar eso y ver si en nuestro prompt o sistema hay que ajustar (tal vez poner un valor por defecto o instruir "si no hay nombre, decir 'cliente'").
- **Datos duplicados o inconsistentes:** Si la IA resume una lista, ¿qué hace con duplicados? O si la entrada tiene dos valores conflictivos (ej: un registro dice hombre y otro dice mujer para la misma persona), ¿la IA cómo reacciona?
- **Nombres poco comunes o variaciones regionales:** Si la IA genera texto con nombres propios (personas, ciudades), probar con alguno fuera de lo común. Por ejemplo: un cliente cuyo nombre es muy extraño, ¿la IA lo respeta bien o lo recorta? O una ciudad con tilde (Córdoba) a ver si la pone correctamente.
- **Longitudes extremas:** Probar con un texto de entrada **muy largo** (¿se trunca la respuesta? ¿la IA empieza a divagar?) y con uno **muy corto** (¿la IA inventa relleno para completar?). Por ejemplo, si la IA traduce descripciones de productos, probar con una descripción de 3 páginas y con una de 2 palabras.
- **Idioma inesperado:** Si usualmente es español, ¿y si entra algo en inglés o spanglish? ¿La IA intentará responder en inglés o mezclado? ¿Tenemos que forzar siempre el idioma de salida?
- **Caracteres especiales o formato raro:** Por ejemplo, meter algún emoji, o símbolo raro en la entrada a ver si rompe algo, o si la IA lo copia tal cual.

Hacer estas pruebas manuales *antes* de que sucedan en vivo te permite pulir detalles. Quizá descubres que en un caso límite la IA produce algo no deseado, entonces ajustas tu prompt o pones un filtro.
Ejemplo: Estás haciendo un bot que responde preguntas frecuentes. Pruebas la pregunta: "*¿Qué pasa si... (algo muy específico y extraño)?*". La IA podría alucinar o dar una respuesta insegura. Ahí decides: "Ok, para preguntas fuera de cierto scope, mejor contesto con un '*Te contactaremos con un representante*' en lugar de dejar que la IA improvise".
💡 **Regla general:** "*Esperemos lo mejor, pero preparemos el sistema para lo peor.*" Probar casos extremos es una manera de inyectar resiliencia. Piensa en ello como pruebas de estrés o "crash tests" de tu flujo IA. Más vale que choque en el ambiente de pruebas y no con un usuario real delante.

Documenta también qué hiciste ante cada caso límite, especialmente si trabajas en equipo, para que todos sepan "*se probó input con 10,000 caracteres y el sistema respondió con X, así que implementamos un recorte a 1000 caracteres*".

2.13 Registrar el **aprendizaje** (lecciones y mejoras)

A medida que iteramos y probamos, vamos acumulando aprendizajes muy valiosos: descubrimos qué prompts funcionan mejor, qué errores comunes se presentaron y cómo los solucionamos, etc. Es fundamental **registrar estas lecciones** de alguna forma*[18]*.

¿Por qué registrar? Porque así construimos una especie de **guía de buenas prácticas interna** y la próxima vez no partimos de cero. También para compartir con colegas y que todo el equipo eleve su conocimiento.

¿Qué conviene anotar? Algunas ideas:

- **Qué falló y cómo se corrigió:** Por ejemplo, "*La IA solía olvidar el campo 'fecha'. Se solucionó añadiendo 'incluye siempre la fecha' al prompt.*"
- **Prompt antes/después:** Tener ejemplos concretos de cómo era un prompt inicial y cómo quedó tras refinarlo es oro puro. Podrían guardarlo en un documento compartido, algo así:
- Prompt v1: "*Haz un resumen del reporte.*"
- Prompt v2 (mejorado): "*Haz un resumen ejecutivo de máximo 3 párrafos del reporte adjunto, enfatizando los hallazgos clave e incluyendo cifras importantes. No menciones detalles menores.*" (Se mejoró para incluir longitud, enfoque y exclusión de detalles menores).
- **Buenas ideas que surgieron:** Tal vez en el proceso se te ocurrió un truco, por ejemplo, "*Pedir la respuesta también en inglés y luego traducir, mejoró la calidad de la traducción.*" (hipotético). Anótalo.
- **Cosas que no funcionaron:** Igualmente útil, saber "*Probamos indicarle a la IA 'sé creativo', pero eso hizo que se inventara cosas, así que no lo usamos más.*"

Lo ideal es dejar esto en un lugar accesible: puede ser un documento de Google Docs, un Wiki interno, un Google Sheet con columnas de "Caso / Problema / Solución / Ejemplo". Lo importante es que **no se pierda el conocimiento**. De esta manera, si otro miembro del equipo enfrenta un problema similar, podrá recurrir a las experiencias previas.

Ejemplo: Piensa en un *equipo de contenido* que utiliza IA para escribir artículos. Tras meses de uso, podrían tener un doc compartido con secciones: "Manejo de alucinaciones", "Tono conversacional vs formal", "Listas en Markdown", etc., donde van acumulando hallazgos. Cuando alguien nuevo se suma, lee eso y en una hora aprende lo que al resto le tomó iteraciones descubrir.

Además, documentar ayuda a justificar mejoras frente a jefes o stakeholders: puedes mostrar "*Miren, originalmente la IA tardaba 1 hora en darnos algo usable; tras afinar prompts y proceso, lo hace en 15 minutos y con 90% menos correcciones.*" con evidencias.

💡 **Tip de colaboración:** si trabajas en equipo, hagan **retrospectivas** ocasionales. Reúnanse y repasen "*¿Qué aprendimos en las últimas 2 semanas usando la IA?*". Cada quien comparte un par de tips o trampas que vio, y actualizan el documento. Esto crea una cultura de mejora continua en el uso de IA.

2.14 Desplegar con seguridad (versionado, roles y métricas)

Finalmente, al **poner en producción** o en uso real el flujo con IA, tomemos algunas medidas de seguridad y seguimiento*[19]*:

- **Versionado del flujo:** Si construiste un flujo automatizado, guarda una copia de la versión estable antes de hacerle cambios a futuro. Así, si tocas algo que rompe el sistema, puedas revertir a la versión anterior fácilmente. Anota qué versión está activa.
- **Asignar responsables:** Define quién del equipo es el "dueño" de este flujo o componente de IA. En caso de un problema, ¿a quién se le contacta? ¿Quién decide futuras mejoras? Esto evita la situación de "esto se cae y nadie sabe qué hacer".
- **Plan de reversión:** Relacionado al versionado, pero implica tener un plan en caso de que la IA empiece a dar resultados malos de repente (porque el modelo cambió, por ejemplo, o los datos de entrada cambiaron). Quizá eso significa "*desactivar temporalmente la función de IA y volver al proceso manual*" hasta solucionar el problema. Tener ese plan escrito ahorra tiempo en crisis.
- **Métricas simples de desempeño:** Es muy recomendable medir el impacto de la solución con IA. No tiene que ser complicado; unos pocos indicadores clave bastan. Por ejemplo:
- *Tasa de correcciones humanas:* de cada 10 outputs de la IA, ¿en cuántos tuvo que intervenir un humano para corregir? Si al inicio era 5/10 y luego bajó a 1/10, es un gran avance medible.
- *Errores evitados:* si antes sin IA se cometían ciertos errores y con el nuevo proceso se redujeron, anótalo. Ej: "*Desde que implementamos la revisión por IA, las faltas de ortografía en informes disminuyeron 90%.*"
- *Tiempo ahorrado:* estimen cuánto más rápido hacen la tarea ahora. Ej: "*Generar el reporte mensual nos tomaba 4 horas, ahora con IA+revisión nos toma 1 hora.*" Ese ahorro de tiempo es importante para valorar la iniciativa.
- *Satisfacción del usuario final:* aunque cualitativa, pueden recoger feedback. Ej: "*Clientes respondieron con menor frecuencia pidiendo aclaraciones desde que los emails se personalizan con IA (indicador de mejor comprensión).*"

Ejemplo de métricas en acción: Imagina un departamento legal que usa IA para bosquejar contratos. Podrían medir "*número de cláusulas revisadas por humano por contrato*". Si antes tenían que reescribir 10 cláusulas y ahora solo 2 gracias a que la IA aprendió sus plantillas, es un éxito demostrable. O registrar "*ningún contrato con error grave desde implementación de IA, comparado con 2 errores el trimestre anterior*".

Llevar métricas también facilita **justificar la continuación o ampliación del proyecto de IA**. Puedes mostrar a dirección que la diplomatura IA para No Programadores valió la pena porque concretamente "*aumentamos X% la eficiencia en tal proceso gracias a la IA*".
Y con eso, completamos nuestra metodología de iteración práctica con IA, de punta a punta: desde planificar la tarea hasta medir los resultados en producción. En la siguiente sección, profundizaremos en cómo **interpretar y ajustar** los resultados automáticos de la IA, tratando casos típicos de error y sesgos, pero ya dentro de esta mentalidad de usuario responsable.
3. Interpretar y **ajustar resultados automáticos**
Incluso siguiendo todos los pasos anteriores, las salidas de la IA pueden venir con detalles inesperados o errores. Esta sección aborda **errores típicos** que ocurren al usar IA y **cómo gestionarlos de forma simple**, es decir, cómo interpretarlos y ajustar nuestro enfoque para corregirlos. Presentaremos cada error común con un ejemplo simple y la acción sugerida para solucionarlo, tal como se vio en clase pero extendido con explicación.
A continuación, un **cuadro comparativo** de errores frecuentes vs. soluciones:

Error Típico	**Ejemplo** (qué podría pasar)	**Cómo Solucionarlo (Acción sugerida)**
Confianza ciega en resultados	Publicar o usar un texto generado sin revisarlo, y se cuela un dato erróneo (ej: una fecha incorrecta)*[20]*.	**Previsualización + Confirmación Humana:** Nunca publiques/uses directamente. Siempre revisa manualmente antes de enviar/publicar, aunque la IA suene convincente*[21]*. En otras palabras, ten un paso de confirmación (visual o formal) para todo output crítico.
Falta de contexto en la respuesta	La IA da un resumen genérico que omite datos importantes de *tu* empresa o ciudad (porque no se los mencionaste)*[22]*.	**Agregar contexto explícito en el prompt:** Asegúrate de incluir detalles relevantes en la solicitud (ej: menciona el nombre de tu empresa, región, cifras específicas). Si salió muy genérico, significa que la IA no tenía suficiente contexto, dáselo en la siguiente iteración*[23]*.
Aceptar la primera respuesta sin iterar	Te quedas con el primer borrador generado aunque sea flojo o mejorable*[24]*.	**Iterar y comparar variantes:** Pide 2 opciones y elige o combina lo mejor de cada una*[25]*. Anima a la experimentación: si la primera no convence, reajusta el prompt (como vimos en la metodología) en lugar de conformarte.

Error Típico	Ejemplo (qué podría pasar)	Cómo Solucionarlo (Acción sugerida)
Datos o ejemplos sesgados	La IA te da recomendaciones o ejemplos que no aplican a tu realidad local, e.g., sugiere acciones pensando en EE.UU. ignorando la realidad de Argentina[26].	**Acotar y equilibrar los ejemplos:** Oriéntala para incluir diversidad. Por ejemplo, especifica "*incluye ejemplos latinoamericanos*" o "*considera el contexto local*". Y deja claros los límites: "*Evita suposiciones culturales fuera de X alcance*"[27]. Si sospechas sesgo, proporciónale contraejemplos o info balanceada.
Sin gestión de errores en el flujo	La IA a veces no responde (falla la API, etc.) y tu flujo de trabajo se queda colgado sin respuesta[28].	**Implementar un Plan B en el sistema:** Mensaje claro al usuario si la IA no responde ("Lo sentimos, intente más tarde"), más opción de reintentar[28]. Y en lo posible, manejar excepciones automáticamente: si la IA no entrega nada útil, que el sistema derive a un humano o a un flujo alternativo en vez de simplemente detenerse.

Como vemos, **muchos errores se solucionan con pasos adicionales de nuestra parte**: revisar, brindar más contexto, no conformarse con el primer intento, y codificar manejos de error. Estos principios ya los fuimos mencionando en la metodología, pero vale recalcarlos:

- La **IA a veces suena muy segura aunque esté equivocada**. Esto se debe a que los modelos de lenguaje *pueden alucinar*, es decir, generar afirmaciones que parecen ciertas pero no lo son[29]. Por eso **no debemos confiar ciegamente** en la salida, por muy bien redactada que esté.
- Si la IA omitió algo importante, normalmente es fallo de contexto: **lo que no se le da, no lo puede inventar correctamente** (o si lo inventa, mete la pata). La lección es: **alimentar bien de contexto**, y si aun así omite, recalcarlo en el prompt.
- La primera respuesta rara vez es la mejor. La verdadera potencia viene de **iterar**. Un usuario perezoso que se queda con lo primero pierde gran parte del valor.
- Los **sesgos en IA** suelen venir de sus datos de entrenamiento, que tal vez tengan predominancia de ciertas perspectivas (por ejemplo, anglosajonas). Tenemos que ser quienes **ajustemos la brújula cultural o de pertinencia**. Si notamos un sesgo (como en el caso del algoritmo de salud más adelante), reconocerlo y corregir la dirección es parte de nuestra responsabilidad.

- Los sistemas de IA necesitan un **plan de contingencia**. Nada de suponer que "como es inteligente siempre responderá". Los errores técnicos o silencios ocurrirán; prepara el sistema para ello (lo repetimos porque es crucial).

Historia ilustrativa (internacional): En 2016, Microsoft lanzó un bot de IA en Twitter llamado *Tay* que aprendía de las interacciones con usuarios. En menos de 24 horas, usuarios malintencionados lo bombardearon con comentarios tóxicos y *Tay* empezó a tuitear frases ofensivas y racistas, aprendidas de ese input. Microsoft tuvo que apagarlo avergonzado. ¿Qué pasó? *Falta de controles e instrucciones claras sobre qué NO decir*. Este es un caso extremo de "confianza ciega" en que la IA se autorregule. La lección: **nunca dejar un modelo abierto sin límites ni supervisión, porque puede irse a lugares indeseados**. Siempre deben existir filtros de contenido y la posibilidad de intervención humana ante salidas indebidas.

Historia ilustrativa (caso local): Más adelante veremos en detalle un caso argentino de error con IA (el sistema de reconocimiento facial de la Ciudad de Buenos Aires que dio un falso positivo). Adelantemos la idea general: el error fue confiar en la identificación automática sin una verificación humana adecuada, lo que llevó a detener a una persona equivocada*[30]*. Es literalmente "confianza ciega en resultados" llevada a la vida real, con consecuencias serias. ¿Solución que debió existir? Un *Human-in-the-loop*: un oficial confirmando la identidad con documentación antes de proceder, en lugar de arrestar solo porque la máquina dijo. También, como veremos, había sesgo en los datos (errores de carga de nombres) que no fue gestionado. Este caso nos recuerda que **cualquier sistema automático, más si usa IA, debe tener revisión humana en puntos críticos**.

En resumen de este apartado: debemos **anticipar y reconocer los errores típicos** al usar IA, y tener estrategias para enmendarlos. Gran parte de "*gobernar resultados con IA*" es justamente ser un usuario atento que no delega el juicio totalmente a la máquina, sino que la usa como herramienta poderosa pero **conduce el proceso**.

Con estos errores típicos y sus mitigaciones en mente, pasemos a examinar más de cerca el tema de las **limitaciones de los modelos de IA** y la **responsabilidad** que tenemos los usuarios al implementarlos.

4. Limitaciones de la IA y **responsabilidad del usuario**

Los modelos de IA actuales, por impresionantes que sean, tienen limitaciones inherentes y pueden cometer errores o sesgos. En esta sección discutiremos cuáles son esas limitaciones (como las famosas *alucinaciones* y los sesgos), qué significa *responsabilidad compartida* al usar IA, la importancia del contexto sensible, y profundizaremos en el concepto de *Human-in-the-Loop* y nuestra responsabilidad al desplegar sistemas con IA. Es un apartado más reflexivo, pero crucial para un uso **ético y eficaz** de la inteligencia artificial.

4.1 Limitaciones técnicas: alucinaciones y sesgos del modelo

Modelos pueden alucinar: Como mencionamos, *alucinación* en IA se refiere a cuando el modelo genera contenido **falso o sin sentido pero con apariencia de verdad**. Por ejemplo,

podría inventar una cita, dar una estadística que suena precisa pero es ficticia, o traducir un nombre propio de forma incorrecta sin avisar. Esto no es un "bug" en el código sino una consecuencia de cómo funcionan los modelos de lenguaje (prediciendo la palabra más probable). Los modelos aprenden de patrones en datos masivos, pero **no tienen una base de conocimientos verificada ni la capacidad innata de decir "no sé"** a menos que se les entrene para ello. Por eso, a veces completan la información con una conjetura.
La definición desde OpenAI lo resume bien: "*Las alucinaciones son declaraciones verosímiles pero falsas, generadas por los modelos de lenguaje.*"*[29]*. Pueden aparecer incluso en respuestas sencillas. Un chatbot podría dar tres versiones distintas de la biografía de una persona, todas incorrectas, pero redactadas con seguridad*[29]*.
¿Por qué ocurre esto? Investigaciones recientes indican que en el entrenamiento se premia mucho que el modelo *no se quede callado* y trate de responder, lo que incentiva que adivine si no está seguro*[31][32]*. Además, el modelo no tiene noción intrínseca de qué es verdad factual; solo sabe qué secuencia de palabras suele seguir a otra en sus datos. Así que puede mezclar datos reales con inventados sin darse cuenta.
Modelos pueden estar sesgados: Los sesgos en IA vienen de los datos con que se entrenan. Si la mayoría del texto de entrenamiento tiene ciertos prejuicios o perspectivas limitadas, el modelo tenderá a replicarlos. Por ejemplo, se ha visto sesgo de género (asociando "enfermera" con femenino y "doctor" con masculino) o sesgos raciales (asociando nombres de ciertas etnias con noticias de crimen). También sesgos culturales, dando por supuestas cosas de EE.UU. que no aplican a otros países.
El modelo *no es malicioso*, solo refleja patrones. Pero eso puede **amplificar desigualdades** si no se corrige. Un caso ilustrativo (que ya se mencionó en clase) fue un algoritmo de salud en EE.UU. que parecía neutro pero resultó discriminatorio: sin usar raza explícitamente, perjudicaba a pacientes negros porque usaba **costos de salud como proxy de necesidad**. Como históricamente se gastaba menos en pacientes negros (por barreras de acceso), el algoritmo *creía* que tenían menos riesgo, cuando en realidad estaban más enfermos*[33][34]*. Es decir, aprendió un **sesgo histórico** incrustado en los datos de costos. Según el estudio en Science, a igual score de riesgo, los pacientes negros tenían en promedio 26% más enfermedades crónicas que los blancos*[35]*, una clara señal de sesgo. ¿La causa? El modelo predecía gastos, no salud*[36]*, y como *los pacientes negros generaban en promedio USD 1800 menos gastos que los blancos a igual nivel de salud*, los clasificaba de menor riesgo erróneamente*[36]*.
Esto muestra un límite: **un modelo de IA no entiende la justicia ni el contexto social**, solo sigue la métrica que le definieron. Si le definimos mal (costos en vez de salud), optimizará eso aunque sea injusto. Y no nos va a advertir "*oye, esto podría ser discriminatorio*".

¿Qué significa para nosotros como usuarios? Que **debemos estar conscientes** de que la IA: - **Puede estar equivocada** aunque no lo parezca (alucinar). - **Puede traer sesgos ocultos** que debemos detectar y mitigar.
Por lo tanto, nuestra responsabilidad es: - **Verificar hechos importantes** por nuestra cuenta. Si la IA da un dato crítico (ej: "el ingreso neto fue X millones"), conviene confirmarlo con una fuente oficial si es posible. - **Usar filtros y herramientas antialucinación:** Algunas implementaciones permiten restringir que la IA no responda si no está segura. Otras dan un score de confiabilidad. Podemos aprovechar eso. También formular los prompts de manera que la IA se sienta con permiso de decir "No tengo suficiente información para responder" en lugar de inventar (diciéndole explícitamente "*Si no estás seguro, indica que no sabes*"). - **Revisar lenguaje y contenido para sesgos:** Si generamos, por ejemplo, descripciones de candidatos a un puesto con IA, revisemos que no esté introduciendo sesgos de género, etc. Si hallamos uno, corregir el prompt o el proceso. - **Elegir bien los objetivos y datos** cuando diseñamos una solución con IA. Como en el caso de salud: la lección fue "*define objetivos que representen la realidad que quieres mejorar, no un proxy que puede estar contaminado*"*[34]*. En términos de usuario no técnico: pensar *¿qué está optimizando realmente mi IA?* Si entrenas una IA de selección de CVs con datos del pasado y en el pasado había discriminación, la IA optimizará según ese patrón. Entonces *no delegar ciegamente decisiones sensibles a la IA* sin auditar estas cuestiones.
4.2 Responsabilidad compartida: usuario + IA
Muchas veces se dice "*la IA se equivocó*", "*la IA discriminó*", etc. Pero en última instancia, **la responsabilidad es compartida** entre la herramienta y **quien la usa o diseña el flujo***[37]*. La IA no es un ente autónomo (por más autónoma que parezca); es un producto de nuestras indicaciones y de cómo la integramos.
En la Diplomatura para No Programadores, esto es fundamental: **nosotros, como profesionales implementando IA, somos responsables de su comportamiento final**.
¿Qué implica responsabilidad compartida?: - Si un email automatizado ofende a un cliente, no es culpa del modelo de lenguaje en abstracto, sino de **quien lo envió sin revisar** (o sea, nosotros). Igual que un becario que manda algo mal, es responsabilidad del supervisor. - Si un modelo decide mal un caso (ej: rechaza un préstamo injustamente), la organización que lo implementó es responsable de ese daño. No pueden culpar al algoritmo como si fuera un empleado independiente; al final alguien decidió usarlo así. - Por lo tanto, **debemos prever controles** y puntos de intervención humana al diseñar cualquier flujo con IA*[37]*. Esto lo hablamos en control de flujo: poner aprobaciones humanas, logs, etc. Somos responsables de insertar esas salvaguardas. - También significa que **debemos ser transparentes** cuando corresponda: si usamos IA para generar algo, asumir la responsabilidad. Ej: hay medios que publican artículos escritos con IA; la responsabilidad de verificar datos sigue siendo del medio, y deberían idealmente aclarar que hubo asistencia de IA.

En cierto modo, hay que ver a la IA como una **herramienta o colaborador** bajo nuestra supervisión. Al igual que un gerente es responsable de lo que hace su equipo, nosotros lo somos de lo que "hace" nuestra IA en nuestro proceso.
Regla de oro: Si no estaría bien que un humano hiciera X sin supervisión, tampoco está bien dejar a la IA hacerlo sin supervisión. Por ejemplo, en salud, un residente novel no da un diagnóstico final sin que lo firme un médico senior. Pues una IA tampoco debería dar recomendaciones médicas al paciente sin que un médico valide.
4.3 Contextos sensibles requieren **más rigor**
Hay áreas donde un error de la IA no solo causa molestia, sino que puede causar **daño real, legal o incluso riesgo de vida**. La presentación menciona específicamente **salud, finanzas, justicia** como ejemplos de contextos sensibles*[38]*, y seguramente podríamos agregar educación (imaginemos sesgos enseñando mal a niños), recursos humanos (decidiendo despidos/contrataciones), infraestructura crítica, etc.
En esos casos, las prácticas de control deben ser **más estrictas aún**: - **Doble validación humana:** Por ejemplo, en un diagnóstico médico asistido por IA, quizás se requiere que dos profesionales independientes revisen la recomendación antes de aplicarla. En un análisis financiero, un segundo par de ojos de un analista verifica las conclusiones de la IA. - **Registro detallado de decisiones:** Llevar un log de qué propuso la IA, qué decidió el humano, por qué se aprobó o rechazó. Esto es importante para auditoría y aprendizaje. En justicia, por ejemplo, si se usa IA para sugerir penas o identificar riesgos de reincidencia, cada caso debería documentarse cómo se llegó a la decisión final, y la IA ahí solo debería ser un *input* no vinculante. - **Criterios de aceptación muy claros:** Definir explícitamente qué consideramos un resultado aceptable y qué no. Por ejemplo, en salud, quizás no se permite que la IA sugiera un tratamiento invasivo, solo recomendaciones generales; lo inaceptable se descarta automáticamente. - **Limitación de autonomía:** En contextos críticos, la IA raramente debe tomar acciones por sí sola. Más bien su papel es *asistir*. Un juez robot que dicte sentencias es inaceptable; un sistema que sugiere jurisprudencia relevante al juez, eso sí puede ser. - **Evaluación ética y legal previa:** Antes de implementar IA en estos campos, debe haber consultas con comités éticos, cumplimiento de regulaciones (p.ej., en la UE se está legislando fuerte sobre IA en estos ámbitos). Un buen diseño considerará desde inicio "¿qué pasa si mi IA se equivoca en tal contexto? ¿a quién afecta y cómo mitigamos ese impacto?"
Para aterrizarlo: - **En salud:** Un error podría significar un tratamiento inadecuado. Así que la IA nunca debería ser la fuente única de diagnóstico. Máxima precaución: quizás la IA resume la historia clínica, prioriza casos, pero un médico revisa todo antes de actuar. - **En finanzas:** Un algoritmo que decide créditos mal puede endeudar injustamente a personas o discriminar. Se debe testear exhaustivamente con datos históricos y reales, y mantener siempre posibilidad de apelación humana a las decisiones automáticas. - **En justicia:** Hay casos ya de IA para predecir riesgo de reincidencia (COMPAS en EE.UU., muy controversial por sesgos raciales).

Imagina el peligro de tomarlo como verdad absoluta. Lo responsable es usarlo, si es que se usa, solo como una referencia, y con funcionarios conscientes de sus fallos.
En resumen, a mayor sensibilidad del dominio, **mayor nivel de intervención humana y controles**. Como dijo la clase: en estos contextos, no solo es recomendable, sino **obligatorio** tener doble validación, registro de decisiones, etc., para no delegar ciegamente en la IA*[38]*.
4.4 *Human-in-the-Loop* (HITL): integrando la supervisión humana
Hemos mencionado varias veces este término, pero definámoslo claramente y veamos cómo se implementa. **Human-in-the-Loop** significa literalmente "humano en el bucle" o ciclo de procesamiento de la IA. Es un enfoque donde **las personas forman parte activa del proceso de IA en puntos críticos**, en lugar de tener un sistema 100% automático. Según una guía empresarial, HITL en IA describe sistemas donde **las personas intervienen en puntos clave como la preparación de datos, la validación de resultados, la aprobación de decisiones y la gestión de excepciones**, permitiendo a los humanos **guiar, corregir o anular** las acciones del sistema automatizado*[3]*.
¿Por qué es tan importante? Porque combina lo mejor de dos mundos: la velocidad y consistencia de la máquina con la **intuición, ética y sentido común** humanos. La investigación muestra que esta participación humana hace los resultados más precisos, adaptables y éticos*[39]*. Piensa en HITL como los rieles de seguridad en una montaña rusa: los carros (IA) van rápido, pero los rieles (humanos) aseguran que no se descarrilen.
Ejemplos de HITL práctico: - Un sistema de moderación de contenido en redes sociales que usa IA para marcar comentarios tóxicos, pero un equipo humano revisa los marcados antes de borrarlos definitivamente. La IA pre-filtra, el humano decide. - Un flujo de *email marketing* donde la IA genera correos personalizados, pero antes de enviar, se colocan en una bandeja de revisión donde un humano los lee por encima (o lee una muestra representativa) y aprueba el envío masivo. - Una cadena de producción con visión artificial para control de calidad: la IA detecta posibles piezas defectuosas, las separa, y luego un operario verifica esas piezas antes de desecharlas. Así no tiene que checar todas, solo las dudosas.
En la clase se mencionó "*HITL práctico (qué significa en la herramienta)*" con algunas implementaciones concretas*[15]*: - **Paso de aprobación manual antes de enviar/publicar:** (Ya lo describimos en controles de flujo) – esto es HITL al final del proceso. - **Checklist de revisión visible para el equipo:** Podría ser un formulario donde el humano tic ea "sí, cumple tono, sí, cumple datos, etc." antes de dar OK. Eso integra a la persona formalmente en el ciclo. - **Logs/historial de cambios y decisiones:** Cada vez que un humano interviene (corrige un prompt, modifica una salida), queda registrado. Así, el *loop* queda documentado y se puede auditar o aprender de él.
El concepto de **usuario responsable** al desplegar IA se fundamenta en aplicar *Human-in-the-Loop*. Por ejemplo, supongamos que hacemos un asistente legal que redacta borradores de contratos. Un usuario responsable diría "*Siempre un abogado revisará el*

borrador antes de enviarlo al cliente." Un usuario irresponsable tal vez enviaría el contrato generado sin revisión. Lo primero es HITL (bueno), lo segundo es delegar totalmente (riesgoso).
Equilibrio: ¿Significa esto que la IA no sirve para automatizar? Para nada. HITL bien implementado no anula la ganancia de eficiencia; simplemente la **canaliza de forma segura**. Puedes automatizar el 90% del trabajo y poner un control en el 10% final. Eso sigue ahorrando muchísimo tiempo comparado con 0% IA. Por eso, no hay que ver la revisión humana como un cuello de botella insalvable, sino como un punto de calidad.
Con el tiempo, a veces el rol del humano en el loop puede disminuir, pero idealmente **nunca desaparecer del todo**, o al menos, siempre debe haber monitoreo. Un buen dicho en IA es "*La inteligencia artificial es amplificación de la inteligencia humana, no sustitución.*"
4.5 Reflexión: responsabilidad del usuario al desplegar IA
Ya cubriendo todos estos puntos, vale hacer una **pausa reflexiva**. Como profesionales que no necesariamente programamos, pero sí usamos y desplegamos soluciones con IA, **tenemos una gran responsabilidad**.
Cuando desplegamos un flujo o sistema con IA, debemos asumir que: - **Somos responsables ante las consecuencias**: buenas o malas. Si algo sale mal por la IA, no podemos lavarnos las manos. Hay que anticipar y responder. - **Debemos conocer la herramienta**: así como uno no conduciría un coche autónomo sin saber cómo frenarlo manualmente, no implementemos IA sin entender sus limitaciones. Formarse (como lo estamos haciendo en esta diplomatura) es parte de esa responsabilidad. - **La ética y el impacto importan**: Más allá de lo técnico, pensemos en el efecto en personas. Ej: Si una IA rechaza a un postulante por sesgo, hay un ser humano afectado. Como usuarios/desarrolladores de flujos, debemos valorar la *justicia*, *transparencia* y *no maleficencia* de lo que hacemos. Preguntarnos "*¿A quién podría perjudicar esto? ¿Cómo lo evito?*". - **Comunicación y transparencia**: Al desplegar IA de cara al público, es responsable decirlo o por lo menos no engañar. Por ejemplo, si un chatbot es IA y no humano, conviene que el usuario lo sepa, para calibrar expectativas. Y si la IA se equivoca, admitir el error, corregirlo, aprender de ello (no encubrirlo). - **Mejora continua**: La responsabilidad no termina en el lanzamiento. Debemos monitorear el funcionamiento de la IA en vivo, recolectar feedback de usuarios, y pulir. La IA puede degradar su performance si cambian condiciones (por ej, modas de lenguaje que no conoce, nuevos nombres propios, etc.). Estar atentos para actualizar prompts o sistemas según la necesidad.
Pensemos en casos extremos de irresponsabilidad: - Lanzar un detector de mentiras basado en IA al público diciendo que es infalible, cuando en verdad tiene 70% de acierto. Es irresponsable porque puede acusar inocentes. - Usar IA para generar contenido masivo sin revisar y publicarlo como noticia. Riesgo de desinformación. - Implementar un sistema de video vigilancia con reconocimiento facial sin evaluar falsos positivos (como el caso porteño). Resultado: gente inocente detenida, invasión de privacidad injustificada.

Nosotros queremos estar en las antípodas de esos casos. Y la buena noticia es que **podemos**. Con lo aprendido: - Sabemos que siempre revisaremos outputs importantes (no confianza ciega). - Sabemos poner al humano donde importa (HITL). - Sabemos afinar para que la IA sea lo más acertada posible, pero aun así la supervisaremos. - Sabemos documentar y aprender de errores para no repetirlos. - Y sabemos decir "*esta tarea tal vez no es apta para IA aún*" y reservar ciertas decisiones solo para humanos, si la tecnología no da garantías suficientes.
Al desplegar flujos con IA, asumamos la actitud de "*Soy el responsable de que esto funcione bien y de minimizar daños*". Esto eleva la calidad de nuestras soluciones y es un signo de madurez profesional. En últimas, la IA es poderosa, pero "*un gran poder conlleva una gran responsabilidad*" – no solo del que crea la IA (OpenAI, Google, etc.), sino de quien la usa en campo.
Para cerrar esta sección: la responsabilidad del usuario en IA es como la de un **capitán con un nuevo tipo de navegante automático**. El capitán sigue al timón, puede delegar ciertas rutinas al piloto automático (IA), pero debe estar listo para tomar control en tormentas, y es él quien rinde cuentas si el barco encalla. Usemos la IA, sí, pero **conduciendo nosotros el barco** en dirección segura.
5. **Prácticas rápidas** – Ejercicios de aplicación (20 minutos cada uno)
Para consolidar lo aprendido, proponemos algunos **ejercicios prácticos cortos** que pueden realizarse en ~20 minutos. El objetivo es que, de forma guiada, apliquen la iteración con IA, la validación de resultados y los ajustes de prompt. Estos ejercicios también reforzarán la necesidad de la intervención humana y la mejora progresiva.
Ejercicio 1: Ajuste de prompt y mejora de salida
Este ejercicio simula el proceso de iteración para mejorar la respuesta de la IA.

- **Paso 1:** Toma un prompt que hayas usado anteriormente (por ejemplo, uno de la Clase 6 o cualquier consulta que hayas hecho a una IA) cuya respuesta inicial **no haya sido ideal**. Si no tienes uno a mano, aquí tienes uno de ejemplo: "*Resume el siguiente texto sobre nuestra empresa.*" (sin más contexto).
- **Paso 2: Identifica un problema** en la salida que obtuviste. Puede ser: faltó información importante, el tono no era el correcto, la respuesta estaba desordenada, se inventó algo, etc. Escribe brevemente cuál es el defecto principal. Ej: "La IA omitió mencionar el producto principal de la empresa en el resumen" o "El tono resultó demasiado informal".
- **Paso 3:** Ahora, **ajusta el prompt** agregando o modificando **una o dos líneas** para corregir ese problema. Siguiendo los ejemplos: podrías añadir "*Asegúrate de mencionar nuestro producto estrella, XYZ.*" O "*Usa un tono formal y corporativo.*" Dependiendo de cuál era la falla identificada.
- **Paso 4:** Ejecuta de nuevo la petición con el prompt ajustado y observa la nueva salida.

- **Paso 5: Documenta en una frase o dos** qué cambiaste en el prompt y por qué mejoró (o si no mejoró, qué harías distinto). Por ejemplo: "*Agregué al prompt la línea 'No olvides incluir datos de ventas recientes', y la nueva respuesta efectivamente incluyó las cifras del último trimestre, haciendo el resumen más completo.*".

Objetivo: Este ejercicio te hace pensar de forma crítica sobre cómo los cambios en el prompt afectan la respuesta de la IA. Refuerza el hábito de no conformarse con la primera respuesta y de usar iteraciones planificadas para llegar a una versión mejor. Además, la documentación final de "qué cambié y por qué mejoró" es una mini práctica de registrar aprendizajes.

Ejercicio 2: Implementando un control humano mínimo en un flujo

Este ejercicio te lleva a imaginar (o configurar si tienes acceso a una plataforma) un flujo automatizado e identificar dónde colocarías un punto de revisión humana.

- **Paso 1:** Piensa en un flujo sencillo en tu ámbito profesional donde podrías incorporar IA. Por ejemplo: "*Responder automáticamente consultas frecuentes de clientes por email*" o "*Generar reportes semanales a partir de datos de ventas*".
- **Paso 2:** Esboza los pasos de ese flujo. Ejemplo para respuestas a clientes: (1) Llega un email con consulta del cliente. (2) IA genera un borrador de respuesta. (3) Se envía respuesta al cliente.
- **Paso 3:** Ahora, **inserta un paso de confirmación humana** antes de cualquier acción externa. En el ejemplo, colocaríamos: (3) **Humano revisa/edita el borrador de la IA y aprueba**. (4) Se envía respuesta al cliente.
- **Paso 4:** Reflexiona brevemente: ¿qué beneficios aporta ese paso humano? ¿qué podría salir mal si no estuviera? Escribe 2 o 3 ideas. Siguiendo el ejemplo: "*El humano corregirá posibles errores de la IA (nombre mal escrito, tono inadecuado) y asegurará que la respuesta sea 100% correcta antes de enviarla.* Sin ese paso, cabría la posibilidad de que el cliente reciba una respuesta errónea o confusa."
- **(Opcional - si cuentas con herramientas no-code):** Si tienes acceso a herramientas tipo Zapier, Make (Integromat) u otras, intenta realmente armar el flujo con un módulo de "aprobar". Por ejemplo, en Make existe un módulo de "*approve*" donde puedes pausar la ejecución hasta que alguien haga clic en aprobar. No es obligatorio hacerlo funcional, pero puede ser interesante ver cómo se implementa técnicamente un HITL.

Objetivo: Este ejercicio te hace concretar el concepto de *Human-in-the-Loop* en un caso real. Al explicitar dónde y por qué poner una revisión humana, interiorizas la importancia de ese control. Te darás cuenta de que no es difícil añadir ese paso, y mentalmente te entrenas para no pasar por alto la supervisión en futuros diseños.

Ejercicio 3: Detectando y manejando una alucinación de la IA

En este ejercicio experimentarás cómo identificar una posible "alucinación" en la respuesta de la IA y cómo reaccionar.

- **Paso 1:** Fórmula una pregunta o prompt a una IA sobre un dato específico que la IA *podría no saber con certeza*. Por ejemplo: "*¿Cuál es el salario promedio de un desarrollador en Argentina en 2025?*" (Un dato que depende de fuentes actualizadas, y la IA podría inventar un número si no lo sabe).
- **Paso 2:** Observa la respuesta. Supongamos que la IA responde: "*El salario promedio es de 120.000 ARS mensuales en 2025.*"
- **Paso 3: Verifica la información**: busca rápidamente en internet fuentes confiables (ministerio de trabajo, encuestas) o utiliza tu conocimiento. Si no coincide o no encuentras esa cifra exacta, sospecha que puede ser inventada o desactualizada.
- **Paso 4:** Haz una **pregunta de seguimiento a la IA para comprobar su certeza**, por ejemplo: "*¿En qué fuente te basaste para ese dato?*". Muchas IAs generativas no citan fuentes a menos que se les haya entrenado para ello; si tartamudea o inventa una fuente poco confiable, es señal de alucinación.
- **Paso 5: Corrige el curso:** Indícale a la IA algo como "*Si no tienes el dato exacto, por favor indica que no estás seguro.*" y vuelve a preguntar. Alternativamente, formula un prompt diferente más seguro, ej: "*Proporciona un rango estimado o explica qué factores afectan el salario de un desarrollador en Argentina.*"
- **Paso 6:** Toma nota de la experiencia: *¿La IA inicialmente pudo haber alucinado? ¿Cómo lo detectaste? ¿Qué cambio hiciste en tu interacción para obtener una respuesta más confiable?*.

Objetivo: Que vivas de primera mano la posibilidad de una alucinación de IA y practiques formas de manejarla: mediante preguntas de verificación o reorientando el prompt. Después de esto, en situaciones reales sabrás no aceptar cualquier dato sin cuestionar, y cómo pedir a la IA que sea más humilde cuando corresponde.

Estos ejercicios son breves, pero encierran muchos de los principios clave: iterar y mejorar, no confiar ciegamente, incluir humanos en el circuito, y estar atento a la veracidad de lo generado.

Recomendación: Intenta realizarlos y discutir los resultados con compañeros o en el foro del curso. Cada uno puede obtener resultados distintos, y es enriquecedor ver cómo otros ajustaron sus prompts o qué fallos detectaron.

6. **Asistentes pre-armados** y recursos útiles en español

En esta sección destacamos algunas **herramientas y recursos** que pueden facilitarte la práctica y la implementación de IA para generar código o texto, especialmente pensados para usuarios no programadores. En la clase se compartieron un par de asistentes ya configurados; aquí los mencionamos y agregamos otros recursos adicionales en español que pueden ser de ayuda en tu aprendizaje continuo.

- **Asistente Gemini (Google):** Un asistente pre-armado disponible en la plataforma Gemini. Link proporcionado: *[40]*https://gemini.google.com/gem/1w5TaYJB49jkAOV-INAaiPAI-XHUmuyC0?usp=shar

ing. Este asistente está diseñado para guiarte en la iteración con IA dentro de ciertos entornos de Google. Por ejemplo, podrías usarlo para experimentar con prompts dentro de Google Docs u otras herramientas de Google integradas con IA. Al ser *pre-armado*, significa que ya tiene ciertas configuraciones o scripts listos para ayudarte (podría incluir plantillas de prompt u opciones para debugging). *¿Cómo usarlo?* Haz clic en el enlace, sigue las instrucciones de autenticación si pide (Gemini es una iniciativa de Google, puede requerir cuenta), y verás una interfaz conversacional. Puedes pegar tus prompts allí, o incluso conectar tus datos. Este asistente te será útil para los ejercicios: por ejemplo, puede guiarte paso a paso en ajustar un prompt o en añadir un paso de revisión en un flujo de Google Apps Script.

- **Asistente GPT – Desarrollo y Debug (Make):** Link proporcionado: *[40]*https://chatgpt.com/g/g-68d7df705d08819184f4ea5aaa99a695-asistente-de-desarrollo-y-debug-make. Este parece ser un asistente de ChatGPT orientado a integrarse con la plataforma **Make** (antes Integromat) para desarrollo y depuración de escenarios (flujos) *no-code*. En otras palabras, es como un "ayudante" cuando estás creando flujos en Make, que puede sugerirte cómo usar ciertas operaciones, cómo corregir un error en tu escenario, etc. *¿Cómo usarlo?* Siguiendo el link, probablemente se abra ChatGPT con un contexto especial (a juzgar por la URL). Puede que tengas que estar logueado en ChatGPT. Una vez dentro, podrías describir tu flujo Make y preguntar cosas como "*¿Cómo implemento un iterador en Make para procesar una lista?*" o "*Mi escenario está dando error X, ¿cómo lo corrijo?*". Este asistente está programado con foco en debugging, así que aprovechalo cuando estés atascado configurando tu automatización con IA en Make.

(Nota: Ambos asistentes mencionados arriba fueron compartidos en la clase, por lo que los enlaces son confiables. Úsalos para practicar. Por ejemplo, tras hacer los ejercicios de la sección 5, podrías consultar con estos asistentes tus soluciones o pedirles sugerencias. El Asistente GPT para Make, en particular, puede ser valioso si decides realmente armar flujos no-code integrando IA y necesitas ayuda técnica.)

- **Recursos adicionales en español:**
- **Guía de Ingeniería de Prompts (en español):** Existe un recurso abierto llamado *Prompt Engineering Guide* de DAIR.AI, traducido al español*[41]*, que recopila técnicas y ejemplos para diseñar mejores prompts. Es muy completo si deseas profundizar en cómo interactuar con modelos de lenguaje de forma óptima. Te enseña desde lo básico hasta casos avanzados, y lo bueno es que muchos ejemplos están en nuestro idioma.
- **Comunidad y tutoriales en español:** Plataformas como *prompt.org.es* ofrecen tutoriales y artículos en español para aprender a sacar provecho a la IA generativa*[42]*. También blogs de tecnología locales (Argentina, España, etc.) suelen publicar artículos

explicando conceptos como LLMs, sesgos, etc., en español y con contextos regionales.

- **Cursos en línea localizados:** Por ejemplo, Coursera tiene un curso de la Universidad de Palermo (Argentina) llamado "Inteligencia Artificial (IA): Interacciones y prompts"*[43]*, que aborda qué es un prompt, su estructura lingüística, cómo refinarlos para distintos formatos. Es gratuito y en español; puede servirte para reforzar lo visto aquí con otra perspectiva académica.
- **Blog del BID sobre IA y prompts:** El Banco Interamericano de Desarrollo (BID) tiene un blog *Abierto al Público* donde publicaron "10 recursos prácticos para fortalecer tus habilidades de prompt engineering"*[44][45]*. Allí destacan la importancia de validar las respuestas de la IA porque pueden contener alucinaciones, y listan recursos (en su mayoría en inglés, pero comentados en español) para seguir aprendiendo. Es un buen punto de partida para quien quiera más material curado sobre el tema.
- **Herramientas de código con IA en español:** Por ejemplo, **GitHub Copilot** tiene documentación en español sobre cómo usarlo dentro de VSCode para generar y depurar código*[46]*. Si bien Copilot es más para programadores, algún alumno curioso podría integrarlo con lo aprendido. También hay asistentes gratuitos como *YesChat Code Debugger*[47] que prometen ayudar a depurar código en español.

Recuerda: la tecnología avanza rápido. Te animamos a **mantenerte actualizado**. Por fortuna, cada vez más contenido aparece en español, ya sea por traducciones oficiales (como el blog de OpenAI en español que citamos sobre alucinaciones) o por creadores locales. Un profesional informado podrá usar la IA con más confianza y seguridad.

Termina la diplomatura, pero con estos asistentes pre-armados y recursos adicionales tendrás apoyo para continuar tu viaje de aprendizaje. ¡No dudes en explorarlos y compartir con la comunidad tus hallazgos!

7. Casos de estudio: **lecciones de implementaciones de IA** (Argentina e internacional)

Para cerrar con profundidad, analizaremos brevemente dos casos reales donde la implementación de IA tuvo problemas significativos: uno en Argentina y otro internacional. Estos casos involucran errores, sesgos o malas implementaciones de IA, y extraeremos de ellos enseñanzas sobre qué salió mal, cuál fue el impacto y qué se podría haber hecho mejor para evitarlos. Son una invitación a reflexionar críticamente como futuros responsables de proyectos con IA.

7.1 Caso internacional: Sesgo en un algoritmo de salud (EE. UU., 2019)

¿Qué pasó?

Un algoritmo comercial ampliamente utilizado en Estados Unidos para gestionar la salud de poblaciones (es decir, para asignar apoyo adicional a pacientes con mayores necesidades) fue descubierto teniendo un **sesgo racial grave***[33]*. En concreto, al comparar pacientes con el mismo "score" de riesgo según el algoritmo, los pacientes negros resultaron estar mucho más

enfermos que los pacientes blancos. En otras palabras, el algoritmo subestimaba el riesgo de los pacientes negros sistemáticamente*[33]*.

Causa de fondo:

El sesgo provenía de la **variable objetivo que el algoritmo intentaba predecir**. En lugar de predecir directamente la salud o gravedad de los pacientes, el modelo usaba **costos de atención médica** como un proxy de la "necesidad" de cuidados*[48]*. Esto parece razonable en principio (quien está más enfermo, suele costar más al sistema), pero ignoraba un hecho social crítico: históricamente en EE.UU., se invierte menos dinero en la atención de minorías raciales, debido a desigualdades en el acceso y calidad de la atención. Por lo tanto, a igual nivel de salud, los pacientes negros **incurrían en menores costos que los blancos** (unos \$1.800 menos de gasto médico al año en promedio a igual condición)*[36]*. El algoritmo, al no conocer la raza pero sí ver los costos, "asumía" que si costaron menos es que estaban más sanos, lo cual era falso, era un sesgo estructural. En resumen: **eligieron mal la métrica (costos) y metieron sesgo sin darse cuenta**.

Impacto:

Este algoritmo se utilizaba con millones de pacientes cada año. El sesgo implicaba que **miles de pacientes negros no estaban recibiendo la intervención extra que necesitaban**, porque el sistema no los identificaba como de alto riesgo*[49]*. Potencialmente, personas más enfermas quedaban fuera de programas de cuidado intensivo, lo que tiene consecuencias en su salud e incluso supervivencia. Además, amplificaba desigualdades: los que ya estaban desfavorecidos (menos acceso, menos gasto histórico) eran *aún más descuidados* por la herramienta supuestamente "objetiva".

Qué se hizo y lecciones:

Cuando los investigadores (Obermeyer et al.) detectaron esto, colaboraron con la empresa que desarrolló el algoritmo para buscar soluciones*[50]*. La principal fue **cambiar la variable objetivo**: en vez de predecir solo costos, modificar el algoritmo para que intentara predecir directamente indicadores de salud (cantidad de enfermedades crónicas, etc.). Con ese cambio, el sesgo se redujo en un 84%*[51]*, lo que muestra que gran parte del sesgo venía de la variable errónea. La lección principal es: **definir correctamente qué queremos optimizar/importante**. Si optimizamos la métrica equivocada, podemos perpetuar o agravar injusticias*[34]*.

Otra lección: **transparencia y auditoría**. Aquí la empresa permitió analizar su algoritmo y se encontró el sesgo. No todas lo hacen. Es vital abogar porque los algoritmos de alto impacto sean auditables. Y como usuarios/profesionales, insistir en evaluar no solo la "precisión global" de un modelo, sino su desempeño en subgrupos (¿trata distinto a poblaciones diferentes?).

En este caso, un *Human-in-the-Loop* podría haber sido: médicos o gestores revisando casos de pacientes negros en altos porcentajes por muestreo para ver si coincidía con la experiencia clínica. Si algo suena contraintuitivo ("¿cómo que este paciente grave no salió prioritario?"), ahí

un humano curioso puede detectar la anomalía. Confiar ciegamente en el número llevó un tiempo a que el problema saliera a la luz.

Qué se podría haber hecho mejor desde el inicio:

Idealmente, haber elegido una métrica combinada desde un principio (por ejemplo, incluir número de condiciones médicas junto con costos, no solo costos). O entrenar el modelo con etiquetas de "quién se enfermó más" en vez de "quién gastó más". También haber realizado pruebas discriminadas por raza antes de implementarlo a escala: si se hubiera visto que para pacientes negros el score correlacionaba distinto con resultados de salud, se habría identificado la discrepancia temprano.

Este caso nos alerta de que **los sesgos históricos en los datos pueden colarse furtivamente**. Y a veces la variable proxy que parece disponible y fácil (como costos) no es la adecuada éticamente. Como futuros gestores de IA, debemos preguntarnos "*¿Estoy midiendo realmente lo que quiero? Si no, ¿qué sesgo puede introducirse?*".

7.2 Caso argentino: Error en sistema de reconocimiento facial (Buenos Aires, 2019)

¿Qué pasó?

En la Ciudad de Buenos Aires se implementó en 2019 un sistema de **reconocimiento facial** de personas prófugas, utilizando IA para analizar las imágenes de cámaras de seguridad (unas 300 cámaras) en tiempo real. El sistema comparaba las caras captadas con una base de datos de personas con pedido de captura (el registro CONARC)*[30]*. En teoría, cuando había un *match*, la policía actuaba.

El caso puntual fue la **detención errónea de un hombre inocente, Guillermo Ibarrola, durante seis días** a raíz de este sistema*[52]*. El sistema marcó su rostro en la estación de Retiro como si fuera una persona buscada por un delito en Bahía Blanca. La policía lo detuvo ahí mismo. A pesar de que él insistió que jamás había estado en Bahía Blanca ni tenía antecedentes, estuvo casi una semana preso y a punto de ser trasladado a un penal, hasta que finalmente comprobaron que se trató de un error de identidad*[52][53]*.

Causa de fondo:

Se identificaron dos problemas principales: 1. **Error o inconsistencia en los datos de la base de prófugos (CONARC):** Resulta que la orden de captura era para otra persona con nombre similar (Guillermo *Walter* Ibarrola), y aparentemente hubo una confusión con el número de DNI en la carga de datos*[53][54]*. Es decir, la base de datos tenía datos mal cargados, lo que provocó un falso positivo: el sistema reconoció el rostro de Guillermo (Federico) Ibarrola y lo asoció erróneamente a la orden de captura de Guillermo (Walter) Ibarrola. 2. **Margen de error del sistema y falta de verificación humana rigurosa:** Si bien las autoridades defendían que el sistema tenía solo un "3% de margen de error"*[55]*, eso sigue siendo 3 de cada 100 identificaciones potencialmente falsas. En un padrón de millones, eso puede ser bastante gente inocente señalada. En este caso, la policía actuó *automáticamente* ante la alerta, sin hacer verificaciones más allá de pedir el DNI en el momento (que coincidía en número con el

del buscado, porque justo ese dato estaba mal en la base). No hubo quizás un protocolo de doble chequeo, como contactar al juzgado de Bahía Blanca antes de detenerlo, o hacer una revisión más detallada de la foto contra el DNI. Básicamente, **confiaron en "la máquina" como si fuera infalible**. Los oficiales decían "fue un error de la máquina"*[56]*, reflejando esa confianza ciega en la tecnología.

Impacto:

Para el individuo, fue muy grave: sufrió 6 días de privación de la libertad injustamente, con todo el estrés y daño que eso conlleva (él declaró "me podían haber arruinado la vida"*[57]*). Se iba a trasladar a un penal con presos condenados, exponiéndolo a peligro. Socialmente, el caso generó **desconfianza en el sistema de reconocimiento facial** y cuestionamientos sobre vigilancia masiva. Si bien las autoridades minimizaban diciendo que el error fue de datos y no del algoritmo en sí*[54]*, la situación evidenció que **un error administrativo combinado con una IA sin supervisión estricta puede terminar en violación de derechos**. También se abrió debate sobre **falsos positivos**: el gobierno dijo "3% de error", pero organizaciones de derechos humanos (CELS) sostuvieron que esos sistemas suelen fallar más y que además el concepto mismo de vigilancia tan amplia es problemático*[58][59]*.

¿Qué se podría haber hecho mejor?

Varias cosas: - **Mejor calidad de datos y validación cruzada:** Antes de poner en marcha, auditar la base de datos de prófugos. Aquí hubo un error de carga de DNI. Un control de consistencia (ej: dos personas distintas con DNI similares) pudo haber saltado. O al menos, al sonar la alerta, verificar manualmente la foto y datos completos de la orden de captura vs la persona detenida. Si se hubiera hecho, habrían notado que buscaban a *Guillermo W. Ibarrola, DNI X*, y el detenido era *Guillermo F. Ibarrola, DNI Y*, evitando prolongar la detención. - **Human-in-the-Loop robusto:** El sistema debió usarse como **herramienta de apoyo**, no como juez. Es decir, ante un match de la IA, **un oficial especialista debería revisar la imagen, los datos, quizás entrevistar a la persona en el lugar, verificar huellas dactilares** si fuera necesario, antes de proceder a encarcelar. La IA debió ser el inicio de la investigación, no la confirmación total. Tener un protocolo: "Alerta facial => detención preventiva de minutos => verificación intensiva => si se confirma, entonces detención formal; si hay dudas, se libera". - **Transparencia sobre error y mejora continua:** Después del error, las autoridades no deberían solo culpar al "error de carga" ajeno, sino revisar todo el sistema. Por ejemplo, mejorar la integración de datos entre Poder Judicial y Policía para evitar incongruencias. También recalibrar el algoritmo si fuera necesario. Y comunicar claramente al público cómo van a evitar que esto pase de nuevo. - **Límites a la tecnología según su precisión real:** Un 3% de falsos positivos puede sonar bajo, pero en aplicaciones de seguridad es alto (imagina 3 de cada 100 transeúntes siendo erróneamente detenidos si tomamos las cifras crudas). Si la tecnología no da para más, se debería tal vez restringir su uso a casos muy acotados (por ejemplo, buscar solo terroristas de altísimo riesgo en eventos masivos, etc., y aun así con

muchas garantías). O esperar a que madure antes de aplicarla a toda la ciudad. La prisa por innovar sin delimitar puede llevar a casos como este.

Lecciones extraíbles:

Este caso nos enseña la importancia de: - **No externalizar la responsabilidad en la "IA"**: Los funcionarios inicialmente dijeron "fue un error de carga, no del sistema", como quitándole peso al tema. Pero desde el punto de vista ciudadano, *el sistema en su conjunto* falló. Las personas a cargo deben asumir responsabilidad y corregir. - **Diseñar sistemas centrados en el ser humano:** Sobre todo en gobierno y justicia. Esto significa que las herramientas de IA se usen de forma complementaria y supervisada. La vida, libertad y reputación de las personas no pueden quedar en manos de una coincidencia algorítmica sin revisión. - **Datos fiables alimentan IA fiables:** "Garbage in, garbage out". Si la base tiene errores, la IA cometerá errores. Invertir en calidad de datos es tan importante como en el algoritmo en sí. - **Considerar el factor de escala:** Un 3% de error, como decíamos, puede significar docenas de casos si se escanean miles de caras al día. Entonces la probabilidad de *algún* inocente detenido injustamente se acerca a certeza con el tiempo. Hay que evaluar no solo "%", sino el absoluto de personas afectadas posible. Si ese número es inaceptable (y uno detenido injustamente ya es problemático), hay que replantear la política.

Este es un ejemplo de **malas implementaciones de IA en seguridad** que repercutió en Argentina. Afortunadamente, salió en medios y generó debate (Página/12 y otros periódicos lo cubrieron extensamente). Desde entonces, hay mayor escrutinio sobre estos sistemas en CABA y otras jurisdicciones.

Impacto general y cómo haberlo hecho mejor (conclusión de los casos):

Tanto en el caso internacional como en el argentino, vemos que la introducción de IA sin suficientes salvaguardas puede **amplificar errores** (en vez de reducirlos). El impacto va desde injusticias en asignación de recursos de salud hasta la privación de libertad de una persona. Ambos casos pudieron mejorarse con: - **Mejor diseño de objetivos y métricas** (en salud, no usar costos; en reconocimiento, asegurar datos correctos). - **Supervisión humana activa** (médicos y especialistas validando o al menos monitoreando recomendaciones del algoritmo de salud; policías y peritos verificando identificaciones faciales antes de apresar a alguien). - **Pruebas piloto y graduales antes de full deployment:** Quizá probar el reconocimiento facial en un entorno controlado (ej: en entradas a edificios gubernamentales) antes de ciudad entera, para calibrar. O en salud, probar el algoritmo en un hospital y compararlo con criterio clínico antes de expandirlo. - **Consideración ética y de derechos:** Preguntarse "¿Qué pasa si se equivoca con esta persona?" nos lleva a poner garantías. - **Feedback loop para la IA:** En ambos casos, si se detecta un error, retroalimentar al sistema. En salud lo hicieron al ajustar el algoritmo. En reconocimiento facial, implicaría corregir la base y quizá ajustar el modelo para ser más cauteloso cuando la coincidencia es parcial.

La moraleja es que **la IA no es infalible ni neutral per se**; todo depende de cómo la usemos. Como profesionales informados, debemos llevar estas lecciones a nuestras organizaciones para implementar IA de forma **responsable, transparente y centrada en las personas**.

---

8. Conclusiones y cierre

Resumen de puntos clave:

Hemos recorrido un largo camino desde la simple idea de "*usar IA para generar algo*" hasta todo lo que implica **gobernar esos resultados** de forma efectiva y responsable. Empezamos destacando la importancia de solicitar bien (buenos prompts) pero más aún de **validar y controlar** la salida de la IA antes de usarla en el mundo real. Desarrollamos una metodología práctica de iteración con IA, paso a paso: enmarcar la tarea, definir criterios de éxito, dar contexto, estructurar prompts detallados, generar variantes, revisar con checklist, ajustar y repetir, agregar controles humanos, probar casos límite, documentar aprendizajes y desplegar con métricas de seguimiento.

Luego, en la sección 3, identificamos **errores típicos** (confiar ciegamente, falta de contexto, no iterar, sesgos, falta de gestión de errores) y cómo abordarlos con soluciones concretas (revisar manualmente siempre, proveer más contexto, iterar varias veces, guiar la IA para diversidad, implementar planes B en flujos).

En la sección 4, profundizamos en las **limitaciones de la IA**: aprendimos que los modelos pueden *alucinar* (inventar datos)*[29]* y replicar sesgos históricos. Por eso, la **responsabilidad del usuario** que implementa IA es enorme: debemos prever controles y puntos de intervención humana*[37]*, especialmente en contextos sensibles como salud o justicia*[38]* donde exigimos doble validación y transparencia. Explicamos *Human-in-the-Loop* (HITL) como la piedra angular de la IA responsable, manteniendo a los humanos en el circuito para guiar y corregir*[3]*. En resumen, **somos los pilotos y la IA el copiloto**, no al revés.

En la sección 5 propusimos **ejercicios prácticos** para que consolides habilidades: cómo ajustar prompts iterativamente, cómo insertar un control humano en un flujo, y cómo detectar/mitigar alucinaciones de la IA. Estos ejercicios, aunque simples, encapsulan la esencia de iterar, validar y corregir.

La sección 6 listó herramientas y recursos para apoyarte: desde los asistentes pre-configurados (Gemini, GPT Debug) que puedes usar ya mismo, hasta guías y cursos en español para seguir mejorando tus prompts y conocimientos. La tecnología de IA evoluciona rápido, pero con esos recursos tendrás con qué mantenerte actualizado y resolver dudas en tu idioma.

Finalmente, analizamos dos **casos de estudio** en la sección 7 – uno internacional sobre sesgo racial en un algoritmo de salud*[33]*, y otro argentino sobre un error de reconocimiento facial*[30]*. Ambos nos enseñaron que sin el debido cuidado, la IA puede causar daños reales, pero también nos mostraron qué se pudo hacer mejor: elegir bien qué optimizar, mantener la

revisión humana, asegurar la calidad de datos y no confiar ciegamente en la "inteligencia" de estos sistemas. En el fondo, recalcaron que la *ética, la responsabilidad y la supervisión* humana no son opcionales, sino requisitos para desplegar IA sin lamentar consecuencias.

Reflexión final sobre la responsabilidad del usuario:

A medida que las herramientas de IA se vuelven más accesibles (¡hoy sin saber programar ya podemos hacer cosas increíbles!), es tentador dejarnos deslumbrar y automatizar todo. Pero debemos equilibrar la **innovación con la prudencia**. Ser un usuario responsable de IA implica ser consciente de que: - La IA es poderosa pero no perfecta. Debemos actuar como su socio crítico, no como un subordinado. - Cada output de la IA que usemos públicamente es **nuestra responsabilidad**, con lo bueno y lo malo. Si sale bien, genial: fue nuestra guía lo que lo logró. Si sale mal, toca dar la cara y corregir, no hay un robot a quién culpar. - Tenemos el deber de **entrenar al resto del equipo** en estas buenas prácticas. Quizá tras esta diplomatura tú seas el referente de IA en tu área; comparte este enfoque iterativo y cuidadoso con tus colegas. Que no se lancen a usar ChatGPT sin entender riesgos; tú podrás orientarlos, mostrarles cómo validar, etc. - La **ética** no es una capa adicional, debe ser intrínseca al diseño de soluciones con IA. Pensar en equidad, privacidad, honestidad de cara al usuario, etc., desde el minuto cero. - **Aprender de los errores**: tanto propios (ajustando prompts y guardando lecciones) como ajenos (casos de estudio) nos hará mejores implementando IA. Equivocarse puede pasar, pero la respuesta responsable es reconocerlo rápido, analizarlo y mejorar el proceso.

Mirando hacia adelante:

La IA generativa y otras formas de IA seguirán avanzando. Nuevas versiones de modelos (GPT-4, GPT-5, etc.), nuevas plataformas integradas en herramientas de oficina, asistentes en nuestro idioma más afinados... Todo esto vendrá. Con lo aprendido, estás en posición de aprovechar esas novedades **de forma segura y efectiva**.

Imagínate dentro de un año: quizás estés liderando un proyecto piloto de IA en tu empresa. Gracias a este curso, sabrás cómo plantearlo: empezar con un POC (prueba de concepto) pequeña*[60]*, definir qué es MVP (producto mínimo viable) y qué no*[61][62]*, iterar en corto ciclos con feedback humano, medir resultados, demostrar valor con métricas, y escalar progresivamente. Y siempre con la mentalidad de "*el humano controla la IA, no la IA al humano*".

En conclusión, **la IA para generar código (o texto, o análisis) sin conocimiento previo** es una realidad fantástica que nos empodera como profesionales no técnicos. Podemos hacer más con menos. Pero ese poder conlleva **responsabilidad**: la de aprender a usarla bien, a iterar, a no dejarla desbocada, y a estar ahí para direccionarla. Con metodología, sentido común, y atención a la ética, podremos implementar soluciones de IA que no solo ahorren tiempo y recursos, sino que lo hagan manteniendo la confianza, la calidad y el respeto por quienes afectan.

¡Felicitaciones por llegar al final de este extenso documento! Esperamos que se convierta en una **guía de estudio** valiosa en tu día a día. Sigue experimentando, sigue aprendiendo, y sobre todo, aplica estas buenas prácticas. La Inteligencia Artificial, bien gobernada, será una gran aliada en tu carrera profesional.

Resumen final: Pide bien, valida mejor, ajusta las veces que haga falta, involucra a los humanos correctos en el proceso, y nunca dejes de lado la responsabilidad. Así, la IA será una herramienta transformadora y no un riesgo. ¡Adelante con confianza y cuidado!

---

[1] [2] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [33] [34] [37] [38] [40] [48] NP.CLASE 7.Documento de clase.docx
file://file_00000000bbc8622f86d43cc5c422da39
[3] [39] Human-in-the-Loop – Una guía para líderes empresariales sobre IA responsable | FlowHunt
https://www.flowhunt.io/es/blog/human-in-the-loop-a-business-leaders-guide-to-responsible-ai/
[29] [31] [32] Por qué los modelos de lenguaje alucinan | OpenAI
https://openai.com/es-419/index/why-language-models-hallucinate/
[30] [52] [53] [54] [55] [56] [57] [58] [59] Seis días arrestado por un error del sistema de reconocimiento facial | La pesadilla de Guillermo Ibarrola, víctima del Gran Hermano porteño | Página|12
https://www.pagina12.com.ar/209910-seis-dias-arrestado-por-un-error-del-sistema-de-reconocimien
[35] [36] [49] [50] [51] El algoritmo que discrimina a los pacientes negros sin conocer su raza | Ciencia | EL PAÍS
https://elpais.com/elpais/2019/10/24/ciencia/1571909798_596622.html
[41] [43] [44] [45] 10 Recursos prácticos para fortalecer tus habilidades de prompt engineering - Abierto al Público
https://blogs.iadb.org/conocimiento-abierto/es/recursos-practicos-para-fortalecer-tus-habilidades-de-prompt-engineering/
[42] Inicio | Prompt.org.es - Aprende Ingeniería de Prompts en Español
https://www.prompt.org.es/
[46] Depurar con GitHub Copilot - Visual Studio - Microsoft Learn
https://learn.microsoft.com/es-es/visualstudio/debugger/debug-with-copilot?view=vs-2022
[47] Code Debugger-Asistente de código IA gratuito y versátil - YesChat.ai
https://www.yeschat.ai/es/gpts-ZxX7eBNY-Code-Debugger
[60] [61] [62] MVP vs POC | Descubre qué significan, sus diferencias, beneficios...Plain Concepts
https://www.plainconcepts.com/es/que-es-diferencias-mvp-poc/

CENTRO DE E-LEARNING UTN BA
Medrano 951 CABA, Buenos Aires Argentina
(1179) // tel +54 11 7078 – 8073 / fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning