

Clase 2 - Material de lectura

Sitio: [Centro de E-Learning - UTN.BA](#) Imprimido Ricardo Monla
Curso: Diplomatura en Inteligencia Artificial por:
para No Programadores Día: Sunday, 4 de January de 2026,
Libro: Clase 2 - Material de lectura 19:56

Tabla de contenidos

1. Mapeo de Procesos y Priorización de Soluciones con IA (Guía Práctica)

- 1.1. ¿Por Qué Mapear los Procesos Antes de Usar IA?
- 1.2. ¿Qué es un Proceso? Pensarlo de Forma Simple
- 1.3. ¿Cómo Mapear un Proceso Paso a Paso?
- 1.4. ¿Es tu Proceso Escalable? (Escalabilidad del Proceso)
- 1.5. Identificando Cuellos de Botella y Tareas Repetitivas
- 1.6. Matriz Impacto vs. Esfuerzo: ¿Cómo Decidir Prioridades?
- 1.7. Quick Wins: Identificando "Victorias Rápidas" con IA
- 1.8. Automatización con IA y No-Code: ¿Qué se Puede (y qué No)?
- 1.9. Ejemplo Práctico: Clasificar Tareas de un Proceso para Automatización

2. Casos Reales de Mejora de Procesos con IA/Automatización

- 2.1. Caso 1: Chatbot "Boti" en Buenos Aires (Sector Público)
- 2.2. Caso 2: Automatización RPA en Zurich Argentina (Sector Seguros)
- 2.3. Caso 3: IA para Detección de Basurales en Mendoza (Sector Gobierno y Medio Ambiente)
- 2.4. Caso 4: Chatbot Interno en el Departamento de Seguridad Pública de Texas (Sector Internacional, Gobierno)
- 2.5. Patrón entre casos

3. Ejercicio Práctico: Mapear y PriorizAR Tu Propio Proceso (Paso a Paso)

- 3.1. Paso 1: Mapear tu Proceso
- 3.2. Paso 2: Identificar Oportunidades de Mejora
- 3.3. Paso 3: Priorización de Ideas de Mejora

4. Compilando Todo: Tu Tabla de Mapeo y Priorización (Plantilla)

5. Conclusión: de la Identificación a la Acción

6. Material de lectura

1. Mapeo de Procesos y Priorización de Soluciones con IA (Guía Práctica)

Introducción y Contexto

En la Clase 1 de esta diplomatura exploramos los fundamentos de la transformación digital y la inteligencia artificial, sentando las bases conceptuales de cómo la IA puede impulsar cambios en las organizaciones. No repetiremos aquí definiciones ya cubiertas (por ejemplo, qué es transformación digital o qué es IA), sino que partiremos de ese conocimiento previo para pasar a la acción.

En esta Clase 2 nos enfocaremos en cómo identificar oportunidades concretas de mejora en nuestros procesos diarios antes de implementar soluciones de IA. ¿Por qué? Porque los modelos de IA no van a mejorar un proceso que no entendemos bien. Muchas veces los cuellos de botella y oportunidades de automatización están ocultos en las tareas cotidianas. Por eso, mapear y analizar nuestros procesos de trabajo es el paso inicial crítico: lo que no se ve, no se mejora.

A continuación, presentamos una guía práctica y accesible –pensada para profesionales sin perfil técnico– para mapear procesos paso a paso, identificar tareas repetitivas o ineficientes, y priorizar soluciones de automatización (ya sea con IA o con herramientas No-Code). También incluimos ejemplos reales de distintos sectores, ejercicios guiados para aplicar en tu propia experiencia, una matriz de impacto vs. esfuerzo para decidir por dónde empezar (Quick Wins), sugerencias de prompts para usar con ChatGPT u otras IA generativas, y una plantilla en tabla editable para mapear y clasificar tus procesos. ¡Manos a la obra!

1.1. ¿Por Qué Mapear los Procesos Antes de Usar IA?

¿Por Qué Mapear los Procesos Antes de Usar IA?

Implementar IA sin entender primero cómo funciona tu proceso actual es como automatizar a ciegas. Mapear el proceso antes de usar IA te permite:

- **Comprender el flujo de trabajo:** Identificar cada paso y su propósito. Esto revela dónde la IA podría ayudar y dónde no aportaría valor. Como se vio en la clase anterior, la transformación digital efectiva implica revisar procesos, no solo digitalizar por digitalizar.
- **Detectar cuellos de botella ocultos:** Al dibujar el proceso, suelen emerger tareas donde se acumulan demoras, errores o retrabajos. Esos puntos problemáticos son candidatos naturales para mejoras.
- **Encontrar oportunidades de automatización:** Un diagrama claro ayuda a ver qué tareas son repetitivas o con reglas predefinidas (ideales para IA o automatización), versus cuáles dependen de juicio humano. Los cuellos de botella y las oportunidades están escondidos en tareas del día a día, y solo al mapearlos salen a la luz.
- **Evitar implementar IA en procesos rotos:** Si un proceso es ineficiente o no escalable en su diseño actual, meterle IA encima no lo arreglará. Primero optimizamos el proceso (o al menos lo entendemos), luego aplicamos IA para maximizar resultados.

En resumen, mapear antes de automatizar asegura que la IA se aplique donde realmente suma valor y no enmascare problemas de fondo. Es el principio de “primero la claridad, después la tecnología”. Como dice el refrán adaptado: “lo que no se mapea, no se puede mejorar”.

1.2. ¿Qué es un Proceso? Pensarlo de Forma Simple

¿Qué es un Proceso? Pensarlo de Forma Simple

De forma sencilla, un proceso es una secuencia de pasos orientados a lograr un objetivo específico. Cada proceso tiene tres componentes básicos:

- **Entrada (input):** algo que inicia o alimenta el proceso (por ejemplo, un formulario recibido, un pedido del cliente, un conjunto de datos).
- **Acción (actividad):** la tarea o conjunto de tareas que se realizan sobre esa entrada (por ejemplo, validar datos, elaborar un informe, aprobar una solicitud).
- **Salida (output):** el resultado o producto generado (por ejemplo, reporte final enviado, producto entregado, decisión tomada).

En términos muy simples: Entrada → Acción → Salida. Si cualquiera de estas partes falla o se hace inefficientemente, el proceso en su conjunto sufre.

Preguntas guía para definir un proceso: Al describir cualquier proceso de tu organización, conviene responder:

- ¿Cuál es el objetivo del proceso? (¿Qué resultado final se busca y para qué sirve?)
- ¿Cuáles son los pasos clave para lograr ese objetivo? (Secuencia principal de tareas, omitiendo detalles muy menores).
- ¿Quién ejecuta cada paso y con qué herramientas? (Roles o áreas involucradas y sistemas o herramientas usadas en cada actividad).

Esta última pregunta es importante: un proceso no ocurre en el vacío, sino que está a cargo de personas o equipos, apoyados en ciertas tecnologías (desde e-mails y planillas Excel hasta softwares específicos). Mapear quién hace qué y con qué recurso ayuda a detectar dependencias y posibles automatizaciones con tecnología existente.

* **Ejemplo simple:** “Publicar una búsqueda de empleo”. Entrada: necesidad de cubrir un puesto (pedido de gerente de área); Acción: redactar la descripción, cargarla en portales de empleo; Salida: publicación en línea de la búsqueda. Objetivo: atraer candidatos calificados. Pasos: recoger requisitos del puesto, redactar anuncio, publicarlo en portal A y B, difundir en redes internas, etc. Personas: RRHH y jefe del área; Herramientas: documento Word, plataforma de empleos, LinkedIn. – Este ejemplo ilustrativo muestra cómo un proceso se puede desglosar en pasos con entradas/acciones/salidas claramente identificadas.

Nota: En la Clase 1 ya abordamos conceptos de transformación digital y cómo replantear procesos para agregar valor con tecnología. Aquí daremos por sabidos esos conceptos y nos centraremos en la aplicación práctica de IA en procesos específicos, evitando definiciones teóricas ya vistas.

1.3. ¿Cómo Mapear un Proceso Paso a Paso?

¿Cómo Mapear un Proceso Paso a Paso?

Mapear un proceso no requiere software sofisticado ni diagramas UML complejos. Para un mapeo básico y claro, puedes seguir estos pasos:

1. **Definir el alcance:** Selecciona un proceso específico y acotado de tu día a día. Por ejemplo, "Atender una solicitud de reembolso de gastos" o "Procesar la nómina mensual de sueldos". Cuanto más concreto, mejor para empezar.
2. **Listar los pasos en orden:** Escribe cada paso del proceso en secuencia cronológica. Empieza con la entrada inicial (ej: "Recibir email del cliente solicitando X") y termina con la salida final (ej: "Cliente recibe respuesta/resolución"). Incluye entre medio todos los pasos intermedios relevantes.
3. **Detallar cada paso con entrada/acción/salida:** Para cada paso, identifica si dentro de ese paso hay su propia entrada y salida específicas. Por ejemplo: Paso: "Generar informe mensual" podría tener como entrada datos de ventas del mes, acción la consolidación en Excel y salida un reporte PDF.
4. **Anotar responsables y herramientas:** Al lado de cada paso, indica quién lo realiza (un rol, persona o incluso si es actualmente automático) y qué herramienta o medio utiliza (un programa, una hoja de cálculo, un correo electrónico, etc.).
5. **Identificar formatos de los inputs:** Útil para luego pensar en automatización. ¿Los datos llegan en un correo electrónico? ¿En un Excel adjunto? ¿En un formulario web? Saber esto ayuda a evaluar si la entrada puede ser leída por una IA o conectada vía integración.

El resultado de este ejercicio debe ser una lista ordenada de pasos del proceso, cada uno con su entrada, acción, salida, responsables/herramientas y formato de input. Básicamente, un pequeño diagrama o listado estructurado que refleja tu proceso con claridad.

Actualmente existen también herramientas digitales para mapear procesos (diagramas de flujo en Visio, Lucidchart, Miro, etc.), pero puedes comenzar simplemente con lápiz y papel, o incluso pidiéndole ayuda a una IA generativa. Por ejemplo, podrías describir tu proceso a ChatGPT para que te ayude a estructurarlo (más adelante daremos prompts sugeridos para esto).

¿Y qué tan detallado debe ser el mapeo? Lo suficiente para entender las grandes tareas y decisiones. No es necesario incluir cada micro-paso obvio (p.ej., "hacer clic en guardar archivo") a menos que eso sea crítico. Concéntrate en las acciones significativas y en las interacciones entre personas y sistemas.

1.4. ¿Es tu Proceso Escalable? (Escalabilidad del Proceso)

¿Es tu Proceso Escalable? (Escalabilidad del Proceso)

Un aspecto importante al analizar un proceso es preguntarse: ¿qué tan escalable es? La escalabilidad se refiere a la capacidad de un proceso de manejar un aumento considerable de volumen o complejidad sin perder eficiencia ni calidad. En otras palabras, ¿puede este proceso crecer sin romperse o volverse impracticable?

Criterios para detectar si un proceso es escalable:

- **Volumen x10:** Imagina que la cantidad de trabajo del proceso se multiplica por 10 (ej: de procesar 10 solicitudes al día pasas a 100 al día). ¿Seguiría funcionando razonablemente? Si la respuesta es "no, colapsaría", entonces el proceso no es escalable en su forma actual.
- **Dependencia de una sola persona:** ¿Hay tareas clave que solo una persona sabe hacer o puede autorizar? Si esa persona falta, ¿se detiene todo? Un proceso muy dependiente de individuos críticos suele escalar mal (un solo cuello de botella humano limita el flujo).
- **Repetibilidad en otros contextos/clientes:** ¿Este proceso se repite igual para muchos casos (clientes, productos, proyectos) o cada ejecución es muy personalizada? Si cada caso es distinto y requiere mucho "traje a medida", es menos escalable que un proceso más estándar que puedas replicar.
- **Grado de decisiones humanas ad hoc:** ¿El proceso requiere constantes decisiones subjetivas o criterios que cambian caso a caso? Cuanto más manual o artesanal el proceso (mucho juicio humano, poca sistematización), más difícil es escalarlo. En cambio, si la mayoría de pasos siguen reglas claras, es más fácil aumentarlos en volumen.
- **Un proceso no escalable típicamente muestra síntomas:** demoras crecientes al subir el volumen, errores o re-trabajos cuando el personal clave está sobrecargado, calidad inconsistente, etc. Esto no significa que no sea importante (de hecho, puede ser crítico), pero sí indica que hay oportunidad de mejorarlo con automatización o rediseño.

Un proceso no escalable típicamente muestra síntomas: demoras crecientes al subir el volumen, errores o re-trabajos cuando el personal clave está sobrecargado, calidad inconsistente, etc. Esto no significa que no sea importante (de hecho, puede ser crítico), pero sí indica que hay oportunidad de mejorarlo con automatización o rediseño.

Por ejemplo, si un proceso depende de ingresar datos manualmente en un sistema para cada nuevo cliente, con 5 clientes al mes funciona, pero con 50 clientes al mes tu equipo estará saturado y propenso a errores. Aquí la solución podría ser una automatización o una integración que escala el ingreso de datos sin requerir 10 veces más gente.

Para reflexionar: Identifica un proceso en tu trabajo y analiza: ¿qué pasaría si tuvieras el doble de trabajo en ese proceso? ¿y diez veces más? ¿Dónde cruje primero? La respuesta suele apuntarte directamente al cuello de botella principal.

1.5. Identificando Cuellos de Botella y Tareas Repetitivas

Identificando Cuellos de Botella y Tareas Repetitivas

Al mapear el proceso, debemos fijarnos en ciertos tipos de tareas que suelen ser candidatas para automatizar o mejorar, o que representan cuellos de botella (puntos de atasco). En particular, presta atención a las siguientes categorías de tareas:

- **Tareas Repetitivas:** Aquellas que se realizan una y otra vez de la misma forma. Por ejemplo, copiar datos de una planilla a otra todos los días, o responder siempre las mismas consultas frecuentes por email. Las tareas repetitivas no solo son aburridas para las personas, sino que son ideales para delegar a una herramienta de automatización o IA, pues suelen tener reglas claras y poca variación.
- **Tareas Lentas o de Espera (⌚):** Pasos donde el proceso se ralentiza. A veces no es la tarea en sí, sino esperas entre tareas (p.ej., esperar aprobación de un gerente, o esperar 5 días a que llegue información de otra área). Identificar dónde el proceso se “pudre” esperando es clave para acelerar el flujo.
- **Tareas Costosas 💰:** Aquí “costo” puede ser en dinero, en tiempo o en riesgo de errores. Por ejemplo, una tarea que consume 5 horas semanales de un empleado altamente calificado es “costosa en tiempo” y quizás ese talento podría reasignarse a algo de más valor si logramos automatizar dicha tarea. O una actividad manual donde suele haber errores (que luego generan retrabajo) es costosa por mala calidad.
- **Tareas basadas en Reglas Claras 🧠:** Aquellas en las que las decisiones se toman según criterios bien definidos o fórmulas. Por ejemplo, “si el formulario A tiene el campo X vacío, rechazar la solicitud; si está completo, proceder”. Estas tareas son excelentes candidatas para IA o No-Code, porque una máquina puede seguir reglas predefinidas consistentemente. En cambio, tareas Difusas o Subjetivas 🗣️ (como “evaluar si el diseño se ve bonito” o “negociar con un cliente enfadado”) son difíciles de automatizar y suelen requerir un humano.
- **Cuellos de Botella:** Un cuello de botella es cualquier punto del proceso donde el flujo se atasca o reduce su velocidad significativamente. Suele coincidir con tareas lentas, tareas que solo una persona puede hacer, o pasos donde se acumula una cola de trabajo pendiente. Ejemplos: “esperando la firma del director” (si el director se demora, todo el proceso se atrasa), o “esperando consolidar manualmente todos los datos” (mientras no se termine esa consolidación, nada avanza). Al marcar en el mapa dónde se produce acumulación de trabajo pendiente, habrás localizado un cuello de botella.

Ejemplo: Supongamos un proceso de atención al cliente por reclamos. Al mapearlo, identificas pasos: (1) Cliente envía reclamo vía formulario web, (2) agente recibe notificación y recopila info del cliente, (3) agente analiza el caso y pide datos adicionales si faltan, (4) se toma decisión y responde al cliente. Aquí podríamos detectar:

- Paso (2) es repetitivo: para cada reclamo, el agente sigue los mismos pasos iniciales de verificar datos → posible automatizar esa verificación inicial.

- Paso (3) puede ser un cuello de botella si el agente tiene muchos casos acumulados (tarea lenta manualmente, y costosa en tiempo).
- La decisión en (4) quizás tiene reglas (p.ej., si es monto < \$100 se aprueba automáticamente, etc.), si es así, es basada en reglas claras (automatizable); si no, y requiere evaluar caso a caso con subjetividad, es difusa (no trivial de automatizar).
- ¿Qué pasa si llegan 10 veces más reclamos? ¿Dónde se ahoga el proceso? Probablemente en el paso (3) de análisis manual – indicando poca escalabilidad ahí.

Listar estas observaciones al lado del mapa de proceso (marcando cada paso con etiquetas como “repetitivo”, “propenso a error”, “cuello de botella”, “regla clara”, etc.) es muy útil para el siguiente paso: pensar soluciones de IA o automatización.

1.6. Matriz Impacto vs. Esfuerzo: ¿Cómo Decidir Prioridades?

Matriz Impacto vs. Esfuerzo: ¿Cómo Decidir Prioridades?

Cuando ya tienes detectadas varias posibles mejoras o automatizaciones en tu proceso, surge la pregunta: ¿Por dónde empiezo?. No todas las mejoras valen la pena por igual, ni requieren el mismo trabajo implementarlas. Aquí es donde la Matriz Impacto vs Esfuerzo se vuelve una herramienta muy valiosa.

¿Qué es la Matriz Impacto vs. Esfuerzo? Es un esquema simple (a menudo presentado como cuadrante o tabla 2x2) que clasifica iniciativas o ideas según dos criterios:

- **Impacto:** el beneficio o mejora esperada si se implementa la idea (por ejemplo, ahorro de tiempo, reducción de costos, mejora en la calidad o satisfacción del cliente). Se suele pensar en impacto alto vs impacto bajo.
- **Esfuerzo:** el costo o dificultad de implementar la idea (por ejemplo, horas de desarrollo, inversión económica, complejidad técnica, necesidad de cambiar muchos hábitos). Se categoriza en esfuerzo alto vs esfuerzo bajo.

Combinando ambos ejes, obtenemos cuatro cuadrantes principales:

Impacto \ Esfuerzo	Bajo Esfuerzo	Alto Esfuerzo
Alto Impacto	<p>Quick Wins (ganancias rápidas) – Prioridad alta. Son mejoras de gran impacto que requieren poco esfuerzo, ideales para abordar primero.</p>	<p>Proyectos Estratégicos – Valiosos pero complejos; suelen requerir planificación a largo plazo.</p>
Bajo Impacto	<p>Pequeñas Mejoras – Cambios menores, de poco impacto pero fáciles. Útiles si hay tiempo, aunque no urgentes.</p>	<p>Evitar – Iniciativas de bajo beneficio que consumen mucho esfuerzo. No valen la pena.</p>

La lógica es priorizar primero las iniciativas Alto Impacto / Bajo Esfuerzo, conocidas como Quick Wins (victorias rápidas). Estas te dan resultados visibles en poco tiempo, aumentando la confianza del equipo en la transformación y liberando beneficios inmediatos.

Las iniciativas Alto Impacto / Alto Esfuerzo son importantes, pero puedes planificarlas a mediano plazo como proyectos más grandes (no descartarlas, pero tampoco intentar hacerlas primero al mismo tiempo que todo lo demás). Las Bajo Impacto / Bajo Esfuerzo se pueden hacer si ayudan en algo (son pequeñas optimizaciones), pero sin distraer de lo importante. Y cualquier idea Bajo Impacto / Alto Esfuerzo casi seguro debe descartarse o al menos dejarse al final de la lista.

Ejemplos para ilustrar la matriz (en contexto de recursos humanos):

- **Quick Win (Alto impacto / Bajo esfuerzo):** Implementar un formulario web simple para pre-filtrar candidatos antes de entrevistas (ahorra tiempo significativo a RRHH con poco trabajo técnico). Otro Quick Win: usar un chatbot básico para responder preguntas frecuentes de

postulantes automáticamente.

- **Proyecto Estratégico (Alto impacto / Alto esfuerzo):** Desarrollar un sistema integral de gestión de talento o implementar un software robusto tipo SAP SuccessFactors – podría transformar RRHH pero requerirá gran inversión de tiempo y dinero.
- **Pequeña Mejora (Bajo impacto / Bajo esfuerzo):** Actualizar la plantilla de emails de rechazo a candidatos para que sea más cordial – mejora un poco la experiencia, se hace rápido, pero el impacto es limitado.
- **Evitar (Bajo impacto / Alto esfuerzo):** Rehacer todo el portal de empleos de la empresa solo para cambiar el look & feel, sin mejorar realmente el proceso de postulación – un gran esfuerzo que no aportará valor real.

Estos son solo ejemplos en el área de RRHH; en tu sector los ejemplos variarán, pero la matriz se aplica igual. Siempre pregunta: ¿Cuánto impacto tiene esta idea? y ¿Cuánto esfuerzo costaría implementarla? Ubica la idea en el cuadrante correcto y eso te guiará en qué hacer primero.

1.7. Quick Wins: Identificando “Victorias Rápidas” con IA

Quick Wins: Identificando “Victorias Rápidas” con IA

Vale la pena profundizar en el concepto de Quick Wins, ya que será tu brújula para priorizar soluciones de IA o automatización. Un Quick Win es un cambio o mejora relativamente simple que genera resultados rápidos, visibles y de alto impacto. Son ideales para iniciar un proyecto de IA/automatización con buen pie, porque:

- Generan confianza en el equipo y dirección, al mostrar valor tangible rápidamente.
- Tienen impacto rápido en el negocio sin requerir grandes desarrollos ni inversiones.
- Inician sin fricción: suelen involucrar herramientas o pasos conocidos, evitando choques con la operación diaria.
- Demuestran resultados que luego justifican acometer proyectos más ambiciosos.

¿Cómo identificar Quick Wins en la práctica? Podemos seguir un proceso estructurado:

1. Inventario rápido de tareas: Reúne de cada área o de tu propio trabajo 3 tareas frecuentes, repetitivas y con reglas claras. Piensa en esas actividades que haces casi en piloto automático o que ves que consumen tiempo valioso cada semana. Involucra a tu equipo: pídeles que mencionen “¿qué tarea rutinaria te quita más tiempo?” Ejemplos típicos: responder correos similares, armar reportes manuales, cargar siempre los mismos datos en distintos sistemas, atender las mismas preguntas de clientes o empleados (FAQs), etc. Haz una lista de estos candidatos.

2. Define criterios simples de evaluación: Para cada tarea identificada, asígnale un puntaje, digamos del 1 al 5, en al menos estas dimensiones:

- **Impacto:** ¿Cuánto beneficio obtendríamos si la optimizamos/automatizamos? (Ej: horas ahorradas, dinero, satisfacción del cliente o empleado).
- **Urgencia/Dolor:** ¿Qué tan “dolorosa” es actualmente esa tarea? Una tarea puede ser repetitiva pero tal vez no duele tanto; otra puede ser una pesadilla. Considera la urgencia o el clamor por mejorarla.
- **Esfuerzo de implementación:** ¿Cuánta complejidad tendría automatizarla? (¿Existe ya una herramienta? ¿Habrá que desarrollar algo complejo? ¿Requiere cambiar políticas?).
- **Riesgo:** ¿Hay riesgo si se automatiza mal? (Por ejemplo, involucra datos sensibles, posibles errores graves, temas de cumplimiento legal).
- **Dependencias:** ¿Para automatizarla necesitamos apoyo de IT u otra área? ¿Depende de que haya un sistema con API, etc.? Las dependencias y riesgos pueden sumar dificultad.

3. Calcula un “Score Quick Win”: Con los puntajes anteriores, puedes hacer un cálculo sencillo a modo de prioridad. Un enfoque común es:

$$\text{Score Quick Win} = (\text{Impacto} + \text{Urgencia}) - (\text{Esfuerzo} + \text{Riesgo} + \text{Dependencias})$$

Win=(Impacto+Urgencia)–(Esfuerzo+Riesgo+Dependencias)

Esto significa que tendrán score más alto las tareas que tienen alto impacto/urgencia y bajo esfuerzo/riesgo. Aquellas con score alto claramente son candidatas a Quick Win. Las de score muy bajo seguramente sean mejoras pequeñas o descartables por ahora.

4. Filtra y valida: Ordena las tareas por el score obtenido. En especial, quédate con las que cumplan Impacto/Urgencia altos y Esfuerzo/Riesgo bajos. Discútelo con tu equipo: a veces un factor cualitativo puede hacerte ajustar (por ejemplo, quizás una tarea tiene impacto moderado pero consume tanta moral del equipo que deciden priorizarla de todos modos). El objetivo es identificar de 1 a 3 Quick Wins máximos para abordar primero.

Siguiendo estos pasos, habrás generado una lista priorizada de “victorias rápidas” factibles.

Ejemplo de Quick Win típico: supongamos que en una empresa detectan que los empleados de soporte gastan horas en responder siempre las mismas preguntas de clientes. Tarea: “Responder FAQs de clientes manualmente”. Impacto: alto (ahorra muchas horas, clientes más satisfechos con respuesta inmediata), Urgencia: alta (el equipo soporte está saturado), Esfuerzo: bajo (se podría implementar un chatbot con una base de conocimiento existente), Riesgo: bajo (las respuestas ya están definidas, no hay mucha incertidumbre), Dependencias: mínimas (se puede usar una herramienta SaaS de chatbot sin pedir mucho a IT). Resultado: Score Quick Win altísimo → priorizar implementar el chatbot FAQ cuanto antes.

Otra tarea en la lista quizás era “Generar reporte estratégico mensual manualmente en Excel”. Impacto: alto también (informe importante), Urgencia: media (es mensual, molesto pero manejable), Esfuerzo: más alto (habría que conectar bases de datos, quizás desarrollar algo a medida), Riesgo: bajo, Dependencia: alta (depende de IT para acceso a datos). Score sale más bajo. Conclusión: proyecto valioso pero no Quick Win, clasifícalo como “🚧 Proyecto estratégico” para después.

1.8. Automatización con IA y No-Code: ¿Qué se Puede (y qué No)?

Automatización con IA y No-Code: ¿Qué se Puede (y qué No)?

Una vez identificados los pasos o tareas a mejorar, debemos preguntarnos cómo implementarlos: ¿Podemos resolver esto con una herramienta no-code o con un agente de IA existente? ¿O necesitaremos un desarrollo a medida más complejo? No todas las soluciones requieren programar desde cero; de hecho, hoy hay infinidad de herramientas accesibles para automatizar sin código o con bajo código.

Veamos unas pautas para decidir:

¿Se puede automatizar con herramientas No-Code? Sí, cuando se cumplen algunas condiciones claras:

- La tarea tiene reglas bien definidas y se repite de forma idéntica cada vez. Por ejemplo, copiar datos de un email a un Excel es siempre igual, enviar un email de respuesta cuando llega un formulario, etc.
- Involucra herramientas conocidas y accesibles mediante integraciones. Ejemplo: Gmail, Excel, Google Drive, WhatsApp, formularios web, CRM comunes.... Si tu proceso transcurre en estas plataformas populares, es muy probable que existan integraciones ya hechas.
- Existen integraciones o servicios que lo hagan: Plataformas tipo Make, Zapier, Microsoft Power Automate, IFTTT, etc., que conectan aplicaciones entre sí. O incluso funcionalidades nativas (p.ej., Gmail tiene filtros automáticos, Excel macros, etc.). Las herramientas No-Code permiten a cualquier persona automatizar flujos arrastrando y soltando, configurando triggers y acciones sin programar.

¿Cuándo necesitamos desarrollo a medida? Hay casos en que la automatización deseada excede las capacidades de las herramientas no-code y debemos pensar en programar o involucrar a un equipo técnico:

- Si el problema involucra lógica muy compleja o decisiones condicionales múltiples que no se pueden configurar fácilmente en un flujo simple. Ejemplo: un algoritmo personalizado de optimización, o una serie de condiciones intrincadas que cambian con frecuencia.
- Si hay que integrar sistemas cerrados o antiguos sin APIs disponibles. Por ejemplo, un software legado que no se conecta con nada; allí quizás un robot RPA es opción, pero a veces ni eso, y hay que desarrollar un conector específico.
- Si la solución requiere control estricto sobre la infraestructura, seguridad o datos más allá de lo que un servicio externo garantiza. Por políticas corporativas, a veces no se pueden usar servicios cloud no-code para ciertos datos sensibles, o se requiere un desarrollo in-house por compliance.

En resumen, siempre intenta primero pensar en soluciones No-Code o de IA ya disponibles (más

rápidas y baratas). Muchos Quick Wins se pueden lograr combinando APIs o servicios existentes. Por ejemplo, antes de encargar un software a medida para generar informes, podrías armar una automatización con Power Automate que cada mes tome datos de Excel y genere un PDF con una plantilla. Si esa ruta no es viable por las razones anteriores (lógica demasiado compleja, sistemas no integrables, etc.), recién ahí consideras un desarrollo propio o contratar desarrolladores.

Tip: Una categoría interesante de herramientas No-Code son las de RPA (Robotic Process Automation), que básicamente actúan como "robots software" haciendo clics y tareas en pantalla como lo haría una persona. RPA suele usarse cuando no hay APIs y hay que automatizar un sistema "a la vieja usanza" (por ejemplo, un bot que llena campos en un sistema antiguo porque no hay otra forma). Es una alternativa no-code (o low-code) útil en ciertos cuellos de botella.

1.9. Ejemplo Práctico: Clasificar Tareas de un Proceso para Automatización

Ejemplo Práctico: Clasificar Tareas de un Proceso para Automatización

Veamos un ejemplo sencillo para afianzar estos conceptos. Supongamos el proceso: "Enviar informe mensual a clientes". Imagina que cada mes alguien en tu empresa debe mandar un informe personalizado a varios clientes. Al mapearlo, salen estos pasos:

1. Buscar el archivo del informe en Drive (el informe general del mes está guardado en Google Drive).
2. Redactar correo electrónico manualmente para cada cliente, adjuntando el informe y escribiendo un mensaje.
3. Adaptar el informe según cada cliente (por ejemplo, filtrando o resaltando datos relevantes para cada uno).

Ahora, analizamos cada paso con las preguntas: ¿Es repetitivo? ¿Escalable? ¿Automatizable con No-Code?:

- **Paso 1: Buscar archivo en Drive** – ¿Es repetitivo? Sí, todos los meses se hace. ¿Escalable? Sí, si tuvieras 10 clientes o 100, el esfuerzo de buscar el archivo es similar (una búsqueda digital). ¿Automatizable con No-Code? Sí, seguramente. Por ejemplo, Google Drive tiene APIs y usando una herramienta como Make se puede automatizar la búsqueda y obtención del archivo. Clasificación: Repetitivo, Escalable, Se puede automatizar con una integración (Drive + Make).
- **Paso 2: Redactar el mail manualmente** – ¿Repetitivo? Sí, es una tarea rutinaria cada mes por cliente (y propensa a copiar/pegar mucho texto similar). ¿Escalable? Sólo hasta cierto punto – si tu base de clientes crece mucho, redactar individualmente se vuelve pesado (lo marcamos como \triangle escalabilidad limitada). ¿Automatizable? Parcialmente sí: se podría usar una plantilla de email automatizada que inserte el nombre del cliente y adjunte el informe, en lugar de escribir de cero cada vez. Clasificación: Repetitivo, \triangle Escalabilidad limitada, Automatizable (mediante plantilla o envío masivo personalizado).
- **Paso 3: Adaptar el informe según el cliente** – ¿Repetitivo? No del todo; cada cliente puede requerir toques específicos (no es idéntico cada vez, sino personalizado). ¿Escalable? Difícil: si tienes más clientes, requiere más criterio humano para adaptar, y la calidad podría sufrir si intentas hacerlo en masa. ¿Automatizable? No fácilmente: aquí hay juicio humano decidiendo qué destacar a cada cliente, algo difuso para una IA actual a menos que se entrene algo muy específico. Clasificación: No repetitivo (varía por cliente), No muy escalable, No automatizable con no-code (requiere criterio humano).

Este análisis nos muestra que los pasos 1 y 2 son buenas oportunidades de automatización, mientras que el paso 3 probablemente deba seguir siendo manual o repensarse de otra forma. En la práctica, podríamos implementar un flujo que automáticamente busque el informe en Drive y envíe correos personalizados a todos los clientes con ese informe adjunto usando una

plantilla. Así, el encargado solo revisa resultados o interviene en casos especiales, en lugar de hacerlo todo uno por uno. El impacto: enorme ahorro de tiempo mensual.

(En la tabla a continuación se resume este ejemplo de clasificación de tareas del proceso con símbolos de Sí (✓), No (✗) y Parcial (⚠)):

Paso del proceso	¿Repetitivo?	¿Escalable?	¿Automatizable con No-Code?
1. Buscar archivo en Drive	Sí ✓	Sí ✓	Sí ✓ (ej: usar Google Drive + Make)
2. Redactar email manual a cliente	Sí ✓	Limitado ⚠	Sí ✓ (con plantilla automática)
3. Adaptar informe según cliente	No ✗	No ✗	No ✗ (requiere juicio humano)

Este ejemplo muestra cómo podemos etiquetar cada paso de un proceso para decidir acciones: pasos 1 y 2 son candidatos a automatizar (quizá incluso Quick Wins), paso 3 no.

2. Casos Reales de Mejora de Procesos con IA/ Automatización

Casos Reales de Mejora de Procesos con IA/ Automatización

Para ilustrar cómo estos conceptos se aplican en la vida real, veamos algunos casos de distintos sectores, incluyendo ejemplos locales de Argentina y uno internacional. En cada caso, notemos cómo identificaron un cuello de botella o tarea repetitiva, y la solucionaron con IA o automatización, priorizando Quick Wins.

2.1. Caso 1: Chatbot “Boti” en Buenos Aires (Sector Público)

Caso 1: Chatbot “Boti” en Buenos Aires (Sector Público)

Un caso emblemático de Quick Win en gobierno es “**Boti**”, el chatbot de la Ciudad de Buenos Aires. Antes, muchas consultas ciudadanas (estado de trámites, información turística, reclamos) requerían la atención de operadores humanos o búsquedas engorrosas en sitios web. El Gobierno porteño implementó en 2019 un asistente virtual (Boti) para responder automáticamente preguntas frecuentes y brindar servicios vía WhatsApp y web.

Este chatbot fue un éxito rápido: resolvía cientos de dudas al instante, liberando a operadores humanos para casos más complejos. En 2024, Boti se potenció incorporando IA generativa (modelos de lenguaje) para ofrecer una experiencia conversacional aún más sofisticada, pudiendo adaptar respuestas a las preferencias del usuario y ayudar, por ejemplo, a planificar actividades turísticas personalizadas. El resultado fue una interacción ágil, simple y accesible entre el Estado y la ciudadanía, mejorando significativamente la calidad y rapidez de la atención. Este Quick Win público generó confianza en el uso de IA, al punto que otras ciudades empezaron a replicar la idea.

¿Por qué es un Quick Win? Porque la implementación inicial de Boti se pudo hacer relativamente rápido aprovechando tecnologías existentes de chatbot, y el impacto fue alto (miles de interacciones automatizadas por día). Impacto alto, esfuerzo razonable = Quick Win logrado.

2.2. Caso 2: Automatización RPA en Zurich Argentina (Sector Seguros)

Caso 2: Automatización RPA en Zurich Argentina (Sector Seguros)

En el sector privado, **Zurich Argentina (seguros)** detectó cuellos de botella en procesos administrativos críticos, por ejemplo en la actualización mensual de pólizas de seguro que involucraba mucha carga manual de datos. Era un proceso repetitivo y lento: cada mes ~37.000 pólizas debían actualizarse, tarea que tomaba unos 10 días de trabajo parcial de ~10 empleados – claramente poco escalable si la cartera crecía.

Zurich decidió implementar RPA (Robotic Process Automation), una forma de automatización no-code donde “robots” de software realizan tareas repetitivas en las mismas aplicaciones que usa el personal. Tras seleccionar la plataforma adecuada, desarrollaron bots para este proceso. El resultado: el mismo volumen (~37.000 pólizas) pudo ser procesado por ~12 robots de software en solo ~4 días, trabajando casi sin pausa. En otras palabras, redujeron el tiempo del proceso a menos de la mitad (de ~10 días humanos a ~4 días automatizados) manteniendo la calidad.

Este Quick Win tuvo un impacto tangible en productividad y eficiencia operativa. Además, el equipo descubrió lecciones valiosas, como la importancia de involucrar a usuarios clave y hacer pruebas piloto antes de escalar. Tras este éxito, Zurich continuó identificando más tareas automatizables, construyendo un “Centro de Excelencia en Automatización” interno para priorizar nuevas oportunidades.

¿Por qué califica como Quick Win? Porque apuntaron a una tarea administrativa altamente repetitiva (impacto grande en ahorro de horas), y la tecnología RPA permitió implementarlo sin tener que reconstruir sistemas de cero (esfuerzo moderado en comparación a reprogramar todo el core de pólizas). La mejora fue rápida y evidente en un proceso crítico de negocio.

2.3. Caso 3: IA para Detección de Basurales en Mendoza (Sector Gobierno y Medio Ambiente)

Caso 3: IA para Detección de Basurales en Mendoza (Sector Gobierno y Medio Ambiente)

No solo en oficinas se aplican estas ideas: en la Ciudad de Mendoza (Argentina) abordaron un problema urbano con IA. Tenían el desafío de detectar micro-basurales clandestinos (acumulaciones ilegales de residuos) dispersos en la ciudad, algo que tradicionalmente requería patrullas de inspección lentas o denuncias vecinales. Identificaron que este proceso de monitoreo ambiental era poco escalable manualmente.

Como solución, implementaron un sistema de visión artificial: usando drones y cámaras aéreas, toman imágenes de la ciudad y una IA analiza automáticamente esas fotos para identificar basurales a cielo abierto. Cuando el sistema detecta una acumulación sospechosa de basura, alerta en tiempo real a las cuadrillas municipales para intervenir de inmediato.

El impacto fue notable: se pueden cubrir grandes áreas en poco tiempo, priorizar acciones de limpieza donde más se necesita y optimizar el uso de recursos operativos. Además, la respuesta más rápida reduce riesgos sanitarios y mantiene la ciudad más limpia y segura para los vecinos. Este caso muestra cómo la IA puede escalar un proceso (vigilancia ambiental) que manualmente era muy costoso y reactivo.

¿Quick Win o proyecto? Montar drones e IA puede sonar complejo, pero comparado con las alternativas (inspecciones exhaustivas humanas), resultó ser un gran triunfo relativamente rápido: la tecnología de visión ya existía, se adaptó a este uso y los resultados se vieron en mejoras inmediatas en limpieza urbana. Sin duda un impacto alto; y si bien requirió esfuerzo técnico, se justifica como iniciativa estratégica con quick wins parciales (por ej., empezar por zonas críticas primero).

2.4. Caso 4: Chatbot Interno en el Departamento de Seguridad Pública de Texas (Sector Internacional, Gobierno)

Caso 4: Chatbot Interno en el Departamento de Seguridad Pública de Texas (Sector Internacional, Gobierno)

En un ejemplo internacional, el Departamento de Seguridad Pública de Texas (EE.UU.) demostró cómo un Quick Win con IA generativa puede mejorar la productividad interna. El departamento manejaba un enorme repositorio de documentos de propiedad y registros, a los cuales debían acceder más de 10.000 empleados y partes interesadas. Antes, obtener cierta información específica requería consultar manualmente al director o buscar en sistemas complejos – un cuello de botella de conocimiento.

La solución fue implementar un chatbot interno potenciado por IA, conectado directamente a la base de conocimientos institucional. Ahora, los empleados pueden hacer preguntas en lenguaje natural al chatbot y obtener respuestas inmediatas, o extraer el documento relevante, en lugar de esperar ayuda humana.. Por ejemplo, un agente puede preguntar “¿Cuál es el procedimiento para X situación?” y el bot le trae el documento exacto y la respuesta.

El resultado: ahorro de tiempo valioso y mejor integridad de los datos utilizados, ya que todos acceden a la información oficial actualizada de forma ágil. Se eliminó un cuello de botella (consultas centralizadas en el director), fomentando una cultura de autoservicio de información.

¿Por qué es un Quick Win? Integrar un chatbot sobre datos existentes fue un proyecto de pocos meses, apalancando IA generativa ya disponible, y tuvo alto impacto en la productividad de miles de empleados sin requerir reestructurar todo el sistema interno. Demostró rápidamente valor, justificando más inversiones en IA interna.

2.5. Patrón entre casos

Patrón entre casos

Estos casos, de distintos ámbitos, comparten un patrón: identificaron un problema acotado y factible de resolver (bottleneck/tarea repetitiva) y aplicaron una solución de IA/automatización de forma enfocada. En cada uno se evaluó impacto vs esfuerzo, priorizando aquello que daba frutos rápidos. También ilustran que Argentina ya tiene ejemplos concretos (no hace falta irse a Silicon Valley para encontrar éxito con IA) y que internacionalmente la tendencia es similar.

3. Ejercicio Práctico: Mapear y PriorizAR Tu Propio Proceso (Paso a Paso)

Ejercicio Práctico: Mapear y PriorizAR Tu Propio Proceso (Paso a Paso)

Ahora que hemos cubierto teoría y ejemplos, es momento de pasar a la práctica. Te proponemos un ejercicio guiado, paso por paso, para que apliques estos conceptos en un proceso real de tu trabajo o entorno. La idea es que, al final, obtengas un mapa de tu proceso, identifiques oportunidades de mejora con IA/No-Code y priorices un Quick Win para implementar.

Para facilitarte la tarea, puedes incluso aprovechar herramientas de IA como ChatGPT para asistirte en cada paso. Te daremos prompts sugeridos en español que puedes copiar y usar directamente con ChatGPT (u otra IA similar) para que actúe como tu “consultor” en este ejercicio.

3.1. Paso 1: Mapear tu Proceso

Paso 1: Mapear tu Proceso

Objetivo: Obtener una estructura clara de tu proceso, con sus pasos ordenados, entradas, acciones, salidas y herramientas involucradas. En esencia, que la IA te ayude a dibujar el mapa del proceso.

Instrucciones: Piensa en un proceso real de tu día a día que quieras analizar. Idealmente, escoge uno que realmente hagas con frecuencia y donde sospechas que podría haber ineficiencias (por ejemplo, "Proceso de cotizar y aprobar una compra con proveedores" o "Proceso de registro de un nuevo empleado en la empresa"). Una vez elegido:

- **Describe brevemente el proceso:** su nombre, en qué consiste, quiénes participan, cuál es su objetivo y qué herramientas/interacciones implica. No te preocupes si no lo tienes super detallado, puedes hacerlo con lenguaje cotidiano.
- Usa el siguiente prompt con ChatGPT (u otra IA) para que te ayude a estructurarlo. Recuerda reemplazar lo que esté entre corchetes [...] con los datos de tu proceso:

Prompt sugerido (Mapeo de proceso):

Quiero que me ayudes a mapear un proceso que realizo seguido.

El proceso se llama: [nombre del proceso].

Consiste en: [explicación breve de lo que trata].

Las personas involucradas son: [roles o áreas que intervienen].

El objetivo del proceso es: [objetivo o resultado final buscado].

Las herramientas que uso son: [ej: Gmail, Excel, Drive, CRM, etc.].

El input que recibo suele venir en [formato de entrada] (ej: correo, Excel, formulario).

¿Podés ayudarme a estructurarla en pasos simples con entrada, acción y salida para cada paso?

Cópialo tal cual, completa los datos entre corchetes y envíalo. La IA te debería responder con una lista ordenada de pasos, donde cada paso tenga claramente su entrada, la acción realizada y la salida generada. Idealmente también mencionará las herramientas usadas en cada paso y el formato de los inputs si lo incluiste.

Resultado esperado: Un esquema tipo:

1. Paso 1 – Entrada: ..., Acción: ..., Salida: ... (Herramienta: ..., Formato: ...).
2. Paso 2 – Entrada: ..., Acción: ..., Salida: ...
3. (y así sucesivamente)

Revisa la respuesta. ¿Refleja bien tu proceso? Ajusta manualmente cualquier detalle que la IA haya interpretado mal. Este es tu mapa base para trabajar.

 **Tip:** Guarda esta lista de pasos, pues la usarás en los siguientes pasos del ejercicio. Si lo haces en ChatGPT, puedes copiar la lista a un documento propio para referencia.

3.2. Paso 2: Identificar Oportunidades de Mejora

Paso 2: Identificar Oportunidades de Mejora

Objetivo: Marcar en tu proceso qué tareas o pasos podrían mejorarse, optimizarse o automatizarse con IA/No-Code, y de qué manera. Básicamente, encontrar esos cuellos de botella, tareas repetitivas, o propensas a error que podrían ser Quick Wins.

Instrucciones: Toma la lista de pasos que obtuviste en el paso 1 (ya sea la respuesta de la IA o tu propia lista). Vas a pedírselas ahora a la IA para que las analice y te dé sugerencias. Puedes copiar la lista tal cual en el prompt a continuación donde dice [pasos de mi proceso].

Prompt sugerido (Detección de mejoras):

Te paso los pasos de mi proceso. Ayudame a identificar:

- Cuáles son tareas repetitivas o manuales.
- Cuáles consumen mucho tiempo o tienen alto riesgo de error.
- Qué pasos podrían resolverse con IA generativa, un agente GPT o herramientas No-Code.

También decime, para cada punto identificado, si sería mejor automatizarlo con una herramienta como Make/Zapier (integración No-Code) o si conviene armar un agente GPT con una secuencia de prompts personalizados.

Aquí están los pasos de mi proceso:

[lista numerada de los pasos de tu proceso con sus detalles]

Al enviar esto, la IA analizará cada paso y probablemente te devuelva una lista donde etiqueta cada paso con observaciones. Por ejemplo, podría decir: "Paso 3 – es repetitivo y consume tiempo, podría automatizarse con una herramienta No-Code (ej: Zapier)" o "Paso 5 – implica creatividad o juicio humano, conviene dejarlo manual por ahora", etc. Incluso puede sugerirte acciones específicas: "usar un prompt para que ChatGPT redacte X" o "usar un software de RPA para Y".

Resultado esperado: Para cada paso de tu proceso, tener indicadores o etiquetas tales como: "(Repetitiva)", "(Manual propenso a error)", "(Automatizable con IA generativa)", "(Requiere desarrollo a medida)", etc. junto con una recomendación de acción: por ejemplo "Sugerencia: crear un prompt para GPT que realice esta revisión" o "Usar herramienta No-Code para integrar sistema A con B", etc.

Revisa estas sugerencias críticamente: al final, tú conoces tu proceso mejor que la IA, así que decide cuáles tienen sentido. Pero seguramente obtendrás buenas ideas, incluso alguna que no se te había ocurrido. Marca especialmente aquellos pasos donde la IA sugirió automatización fácil (esas son candidatos a Quick Win).

3.3. Paso 3: Priorización de Ideas de Mejora

Paso 3: Priorización de Ideas de Mejora

Objetivo: Decidir por dónde empezar entre las mejoras identificadas, usando la mentalidad de Quick Wins (impacto vs esfuerzo). Aquí vas a hacer una priorización apoyándote en la IA para que te ayude a comparar impacto y dificultad de las soluciones propuestas.

Instrucciones: Con la lista de oportunidades del paso 2, ahora pídele a la IA que las priorice considerando impacto y facilidad. El prompt siguiente le dice que categorice cada idea como Quick Win, proyecto estratégico u otras categorías, similar a nuestra matriz.

Prompt sugerido (Priorización Quick Wins):

Ayudame a priorizar las mejoras que detectamos en el proceso.

Quiero saber:

- Cuáles tendrían más impacto si se implementan.
- Cuáles son más fáciles de implementar (ya sea con IA generativa, herramientas No-Code o GPTs personalizados).

Clasificalas para mí en estas categorías:

Quick Win con IA o No-Code (alto impacto, bajo esfuerzo).

Proyecto estratégico (requiere más desarrollo o validación, alto impacto pero alto esfuerzo).

No vale la pena automatizar (bajo impacto o demasiado esfuerzo/riesgo).

Para cada mejora, indicá la categoría y una breve justificación del porqué.

Al enviar esto, la IA tomará las ideas (según lo que discutieron en paso 2) y las ubicará en Quick Win, Proyecto o Descartable, con una breve explicación. Por ejemplo: "Mejora X: Quick Win – porque ahorra 5 horas semanales y se hace con una herramienta simple. Mejora Y: Proyecto estratégico – alto valor pero requiere integraciones complejas. Mejora Z: No vale la pena – impacto mínimo en ahorro de tiempo."

Resultado esperado: Una clasificación de cada posible mejora en una de las tres categorías (Quick Win, Proyecto, No prioritaria), con su justificación. Esto prácticamente te dará un orden de implementación sugerido: primero abordar los Quick Wins, planificar a mediano plazo los Proyectos estratégicos, y olvidarse de lo que no rinde.

Al llegar aquí, ¡felicitaciones! Ya tienes lo esencial: conoces tu proceso en detalle, identificaste cuellos de botella/tareas automatizables, generaste ideas de solución con IA/No-Code, y las priorizaste lógicamente. Este es exactamente el ejercicio que hacen las organizaciones exitosas antes de lanzarse a implementar IA: alinear la tecnología con oportunidades reales de negocio.

4. Compilando Todo: Tu Tabla de Mapeo y Priorización (Plantilla)

Compilando Todo: Tu Tabla de Mapeo y Priorización (Plantilla)

Como entregable final de este ejercicio (ya sea para ti mismo, para tu equipo, o si estás cursando formalmente la diplomatura), es muy útil resumir todo en una tabla. Esta tabla actúa como “mapa + backlog” de mejoras: resume el proceso paso a paso, identifica el tipo de cada tarea, la posibilidad de automatización, la acción sugerida y la prioridad. Básicamente concentra en un solo lugar todo tu análisis.

A continuación, te presentamos una plantilla de tabla editable que puedes copiar y completar con la información de tu proceso. Incluimos un par de filas de ejemplo para guiarte (puedes quitarlas cuando hagas la tuya propia):

Paso del proceso	Herramienta usada	Tipo de tarea (manual, repetitiva, etc.)	Automatización posible	Acción sugerida (Prompt, GPT, No-Code, Desarrollo)	Prioridad (Alta/Media/Baja)
Ej: Recibir pedido del cliente	Gmail (email)	Manual, repetitiva	Sí (lectura automática del mail)	GPT extractor de datos del correo	Alta (Quick Win)
Ej: Analizar viabilidad del pedido	Análisis humano + Excel	Lógica difusa (requiere juicio)	No (IA puede asistir pero no decidir)	N/A – mantener manual (IA sug. como asistencia)	Media
...

Plantilla en blanco (para que la rellenes):

Paso del proceso	Herramienta usada	Tipo de tarea	Automatización posible	Acción sugerida	Prioridad
... (manual/repetitiva?...)	... (Sí/No/Parcial)	(Prompt, GPT, No-Code?)	... (Alta/Media/Baja)

(Agrega tantas filas como pasos tenga tu proceso. Puedes hacer una tabla similar en Excel o Docs si te resulta más cómodo.)

Una vez llena, esta tabla te servirá de hoja de ruta. En el ejemplo de la tabla, se ve que “Recibir pedido del cliente” se marcó como repetitivo y automatizable (se podría usar IA para leer

automáticamente el email y extraer los datos del pedido), por eso se sugirió una acción (un GPT que lea correos) y se le dio prioridad Alta por el valor que aportaría. En cambio, “Analizar viabilidad” es una tarea de juicio humano complejo, difícil de automatizar completamente; quizás la IA solo ayude con sugerencias, pero seguirá siendo humana la decisión – por eso prioridad media.

5. Conclusión: de la Identificación a la Acción

Conclusión: de la Identificación a la Acción

En esta guía desarrollamos un recorrido completo desde entender qué es un proceso y mapearlo, hasta identificar cuellos de botella y priorizar Quick Wins para aplicar IA o automatización. Si seguiste los pasos con tu propio caso, ahora tienes no solo conocimientos teóricos, sino un plan concreto de mejoras para tu proceso.

Algunas reflexiones finales:

- Siempre comienza por entender y arreglar el proceso antes de meterle IA. La tecnología debe venir a potenciar un buen proceso, no a ocultar uno malo.
- Enfócate en Quick Wins para ganar tracción inicial. Es mejor automatizar 5 tareas pequeñas que ahorran 1 hora cada una (total 5 horas ahorradas) en un mes, que embarcarte en un mega-proyecto de 6 meses que ahorrará 10 horas pero recién se vea después.
- Involucra a tu equipo en estos análisis. Mapear procesos en grupo aporta visiones distintas (quizá alguien conoce un atajo o sufre un problema que otros no ven) y genera compromiso con las soluciones.
- Aprovecha las IA generativas como ChatGPT u otras herramientas en el camino. Como vimos, pueden ser consultores accesibles para estructurar ideas, brindarte listas de verificación, incluso para generar automáticamente parte de la documentación del proceso o hasta para crear prototipos de soluciones (p. ej., escribir un pequeño script, sugerir prompts, etc.).
- Documenta y mide. Una vez que implementes un Quick Win, anota el antes y después (ej: "X tarea tardaba 3 horas a la semana, ahora con la automatización tarda 10 min"). Esto te dará evidencia del impacto, invaluable para justificar futuras inversiones o para iterar mejoras.

La transformación con IA es un camino continuo. Hoy identificaste retos y mejoras; mañana, con esos Quick Wins en marcha, surgirán nuevos desafíos y oportunidades (quizá al liberar tiempo del equipo, pueden enfocarse en tareas de mayor valor agregado que antes no hacían). El hábito de mapear, analizar y priorizar nunca termina, pero se vuelve más natural con la práctica.

¡Enhорabuena por llegar hasta aquí! Te animamos a implementar tu Quick Win priorizado y vivir la experiencia de mejorar tu proceso con IA o No-Code. En la próxima clase, continuaremos avanzando en esta línea, explorando metodologías de innovación (Design Thinking aplicado a IA) para seguir convirtiendo retos en soluciones creativas.

6. Material de lectura

Material de lectura

Te compartimos un [archivo descargable](#) con todo el contenido visto en la Clase 2.