

Trabajo Práctico Final: Diseño e Implementación de una Solución No Code**Autor: Maximiliano Zuin****FASE 1: Relevamiento y análisis del proceso a mejorar**

Actualmente Gerencio un local comercial minorista dedicado a la venta de artículos del hogar, tecnología, muebles y colchones. Dentro de mis actividades semanales y las del equipo de ventas, puedo identificar dos procesos que considero repetitivos, y/o de gran volumen y donde ganaríamos mucho automatizándolos:

- **A) Carga de Facturas de compra / Remitos / Recibos de pago a proveedores en planilla Excel.** Este proceso lo realizo yo. Debería ejecutarlo de forma semanal, pero suelo retrasarme dado que es la carga administrativa más grande que tengo y la hago según necesidad (dado que en el día a día aparecen muchas otras tareas). La realizo en una planilla Excel que utilizo básicamente para “manejar el negocio” donde cargo no solo eso sino toda la info financiera (cheques y mov. Bancarios) para proyectar los flujos de dinero a corto/mediano y largo plazo y las compras a realizar. Los puntos de dolor aquí son el tiempo manual que pierdo al cargar cada documento en el Excel (lamentablemente no puedo costear un sueldo de un administrativo hoy por hoy), siendo una tarea super repetitiva y de mucho volumen, y quitándome tiempo para otras cuestiones más estratégicas del negocio. También el riesgo de error al pasar cada uno de los datos de forma manual desde los distintos documentos.

- **B) Actualización de precios en Excel.** Manejamos una lista en Excel que en la medida que van llegando las listas de los proveedores (llegan al mail), el equipo de venta (3 vendedores) en sus tiempos muertos y para aprovechar el tiempo, actualizan los costos en la mismas, y yo los mkups de venta y comisiones de tarjeta (manejamos aprox. 3000 – 4000 productos). La misma se actualiza según llegan las listas (hace unos años era permanente y diariamente, hoy un poco más espaciado, mensual). La misma se encuentra subida en la nube (DROPBOX) y todos tienen acceso en las diferentes PC`s. En esta tarea también hay una gran carga manual y repetitiva y no pueden modificar dos personas al mismo tiempo (dado que se pisa con la otra). Los puntos de dolor aquí son el tiempo que pierde cada uno de los vendedores que actualizan los costos en la misma (pudiendo dedicar ese tiempo a vender por ejemplo), los errores de carga de costos al transcribir los mismos, el solapamiento de los mismos (dado que la planilla se encuentra en DROPBOX)

Al analizar estos dos para priorizarlos:

- A) IMPACTO: **Alto** (me ahorraría entre un 10% - 20% del tiempo que destino a cargar esto) - COMPLEJIDAD: **Baja** (si logro que los datos del documento digitalizado pasen a una hoja tipo excel me ahorraría toda la carga manual) – VOLUMEN: **Alto**
- B) IMPACTO: **Alto** (También lograría reducir una gran carga administrativa y errores al pasar de las lista al Excel) – COMPLEJIDAD: **Alta** (considero muy complejo que logre identificar todos los productos de cada lista de precio de cada proveedor (más de 30 proveedores y cada una tiene un formato diferente, suelen aparecer códigos nuevos de productos) y lo considero más como un proyecto de mínimo 3 meses de desarrollo. – VOLUMEN: **Medio**.

	IMPACTO	COMPLEJIDAD	VOLUMEN
Carga de Documentos en Excel	ALTO	BAJA	ALTO
Actualización de lista de precios en Excel	ALTO	ALTA	ALTO

Con esta priorización considero el proceso A es un verdadero Quick Win que puede resolverse con una solución no code. El otro (que también ejecutaré en un futuro mediano plazo) representa más un proyecto de medio plazo y considero otro tipo de solución más compleja.

Al diseñar el Flujo de la solución del proceso A, tenemos:

- **Entrada:** el proceso es disparado por el ingreso del mail a Gmail con el documento adjunto (Factura, Pedido, Recibo y/o Remito).
- **Proceso:** Aquellos mails que cumplen con los requerimientos establecidos (adjuntos con la palabra “Factura”, “Pedido”, “Recibo” o “Remito”) se baja el documento y se guarda en una carpeta ubicada en Google Drive para tal fin, luego mediante el uso de alguna IA se toma el archivo y procesa los datos del documento (17 campos en total donde se encuentra “Fecha”, “Tipo de documento”, “CUIT”, “Nombres del proveedor”, “Detalle”, etc) y se actualiza en un hoja de cálculo ubicada en la nube.
- **Salida:** El documento guardado en la carpeta de la nube y la fila en sheet con los datos preestablecidos.
- **Herramientas involucradas:** Gmail, Gemini, Google Drive, Google Sheets.

FASE 2: Diseño e implementación de la solución No Code

- 1) Diseño Técnico

- **Trigger:** Cualquier email que ingrese a la casilla de Gmail y contenga adjunto un archivo cuyo nombre contenga las Palabras “Fac”, “Fc”, “Pedido”, “Recibo”, “Remito”.

- **Frecuencia:** Se programó para que el proceso se ejecute una sola vez al día (9:00).

- Nodos involucrados y descripción del Flujo:

- **Nodo 1: Gmail (Watch email):** Este es el nodo de inicio (trigger) donde comienza observando la casilla de email y mediante un filtro permite seleccionar solo mail con adjuntos para que en el nodo siguiente listarlos y filtrarlo (only email with attachments and media)
- **Nodo 2: Gmail (List email attachment and media):** En este nodo listamos aquellos mails con adjuntos y, al agregarle un filtro sobre la línea que une este nodo y el que sigue (**Google drive**) podemos seleccionar todos aquellos que cumplan con el criterio de búsqueda de documentos (factura, recibo, remito, pedido).
- **Nodo 3: Google Drive (Upload a file):** En este filtramos el filename que proviene del nodo anterior y le pedimos que lo suba en la carpeta que creamos para tal fin (“FACTURAS REMITOS RECIBOS”).
- **Nodo 4: Google Gemini Ai (Upload a file):** Se introduce este módulo para poder subirle a la IA de Google (solicitando previamente la API Key correspondiente y conectando el nodo) el archivo desde Gmail – List email attachments and media.
- **Nodo 5: Google Gemini Ai (Generate a response):** Aquí mediante el prompt correspondiente solicitamos actúe como experto contable y analizando el archivo, extraiga los datos de los 17 campos listados, y siga algunos parámetros (por ejemplo si son varios elementos que los separe con punto y coma) y que devuelva un archivo en formato JSON.
- **Nodo 6: JSON (Parse JSON):** Aquí toma el texto Json del nodo anterior y lo lleva a variables separadas para que aparezcan en columnas separadas luego en Sheets.
- **Nodo 7: Google Sheets (Add a Flow):** Aquí mapeamos cada uno de los 17 campos que surgen del Nodo anterior a cada una de las columnas del archivo de Google Sheet cuyo nombre y ubicación tenemos definidos (FACTURAS REMITOS RECIBOS/FACTURAS REMITOS Y RECIBOS).
- **Nodo 8:** Se agregó un Error handler unido al Nodo de Gmail en caso de algún error al filtrar los adjuntos en los mails, enviando un correo a otra casilla de correo.

- 2) Implementación Técnica

